Forward Collision Detection using a Stereo Vision System

Sergiu Nedevschi, Andrei Vatavu, Florin Oniga
Technical University of Cluj-Napoca
Sergiu.Nedevschi@cs.utcluj.ro

Abstract

It is known that 93% of accidents occur because of the driver behavioral errors. Many of these can be prevented. Therefore a driver assistance system must include a collision detection component that is capable of generating warnings to prevent an imminent impact and to advise the driver in the traffic scenarios. This paper presents a forward collision warning approach based on a 3D Elevation Map provided by a Dense Stereo Vision System. We detect the obstacle delimiters and taking into account the car parameters evaluate the car trajectory and associate to it a driving tunnel. A warning is generated when the obstacle delimiters intersect the current driving tunnel at an unsafe distance. Our system is robust and works in real-time.

1 Introduction

1.1 Motivation

According to the statistical data of the United Nations Economic Commission for Europe (UNECE) [1], in 2004 there were 3.681.571 road accidents in Europe and 1.899.870 in North America, registered with 146685 killed persons in Europe and 42836 in North America. It is known that 93% of accidents occur because of the driver behavioral errors [2] [3]. Rear-end crash problem, associated with a driver inattention factor occurs in 12.9% of these cases. A study made by Daimler-Benz in 1992 estimates that 60% of all rear-end collisions can be prevented giving a 0.5-second additional warning time to the driver. Moreover, 90% of all rear-end collisions can be prevented having one second in advance (cited in [4]). Therefore a driver assistance system must include a collision detection component, able generating warnings to prevent an imminent impact and to advise the driver in the traffic scenarios.

1.2 Related Work

The Forward Collision Warning (FCW) systems can be categorized based on the used sensors in the collision detection process (RADAR, LASER, Vision Based etc). Most of FCW employ Laser or Radar sensors [5] [6] situated in front of the car to receive the information about the traffic scene. A FCW system that uses radar sensor is VORAD VS-400 by Eaton. VORAD includes a high frequency forward looking radar that warns drivers of potential obstacle up to 137 meters ahead. A different type of FCW technology is vision based using single camera and providing information like image scale change or image position to detect or track vehicles on the road. An example of a system that uses monocular vision is MobilEye-AWS [7]. Most algorithms for FCW are based on determining the relative speed and Time To Collision (TTC) value directly from the position of the object in the image [7][8][9][10]. In [11] is presented a dynamic situation and threat assessment for collision warning.

1.3 Contributions

This paper presents a FCW approach using a dense stereo system. The proposed algorithm is based both on the 3D information provided by the stereo system and by the ego-car sensors such as yaw rate and car speed. We propose a FCW method that employs two different 3D models:

- The Obstacles Delimiters
- The Drivable Tunnel

The Obstacles Delimiters are extracted from the Elevation Map and are given as a set of unstructured polygons. We developed a novel approach for delimiter extraction exploiting the Elevation Map through radial scanning. The Drivable Tunnel model describes a non-convex polytope and has a variable trajectory depending on the yaw rate, steering angle and car speed [12]. The tunnel has a different length at the time *t*, based on an adjustable warning time and the car speed.

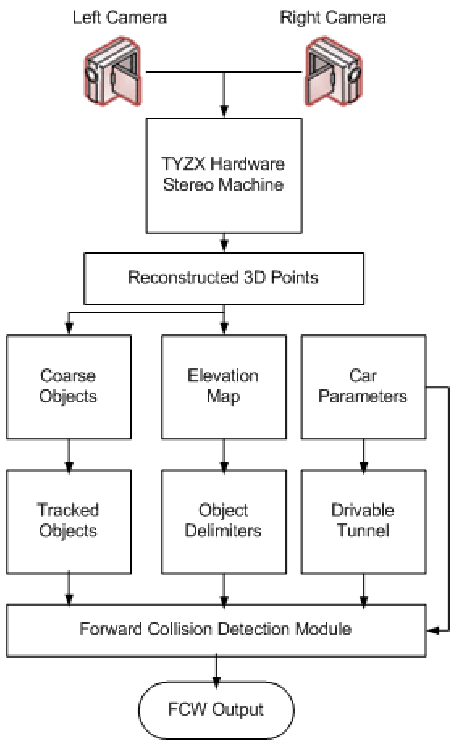

Figure 1. System Architecture

In the next section we describe the proposed FCW architecture. Several ways to extract obstacle delimiters are presented in the section 2. Section 3 shows the proposed Drivable Tunnel Model, whereas collision detection approach is presented in the fourth section. The last two sections show the experimental results and conclusion about the FCW we have developed.

1.4 Proposed Architecture

We describe an approach of collision detection. Our method has been conceived for an urban driving assistance system based on dense stereovision that we are developing [13]. The FCW system structure includes many separated modules(Fig. 1):

TYZX Hardware Stereo Machine The "TYZX" hardware board performs 3D reconstruction [14].

Coarse Objects The Coarse Objects are extracted from the dense stereo information through the grouping process taking into account 3D density variation with distance [15].

Tracked Objects The coarse object's position is tracked using Kalman filtering. Tracked objects are described by their position, size and speed [16].

Elevation Map The elevation map represents a description of the scene, derived from the raw dense stereo information. The elevation map cells are classified into drivable points (blue points), curb points (colored in yellow) and object points (red points). The Elevation Map result is used in the Obstacle Delimiters detection [17].

Car parameters The car parameters such as wheel speed and yaw rate are collected from the onboard sensors at the acquisition time for each frame in the video sequence.

Object Delimiters The Object Delimiters detection uses the Elevation Map results as the input and generates a set of unstructured polygons approximated with the objects contour. The delimiters are extracted from the Elevation Map through the radial scanning process. We calculate the Obstacle Delimiters for both structured and unstructured objects.

Drivable Tunnel The drivable tunnel represents a virtual area around the ego-car trajectory and is generated by the mechanical and movement characteristics of the car. It can be perceived as a safety zone for the ego-car. The length of the trajectory must be greater or equal to the car braking distance. The 3D model of the tunnel is described by a polyhedron representing a non-convex and structured polytope.

Forward Collision Detection The collision detection process is performed between the Drivable Tunnel and Obstacle Delimiters.

FCW Output A visual warning is generated based on the detected results from the FCW module. The warning magnitude can be different taking into account the type of the classified points (object points or curb points) and the relative velocity between the Ego-Car and the tracked object from the scene.

2 Delimiters Detection

This section presents the implementation of the obstacle delimiters extraction. Several ways to extract obstacle delimiters were analyzed.

2.1 Detection Methods

A set of steps have been identified for the delimiters extraction:

1. **The generation of the Top View projection**. The Top View image is computed from the Elevation Map. We suppose that obstacles are disjoint in the Top View.

2. **Object labeling**. In this step each object from the Elevation Map is labeled with a unique identifier.

3. **The contour extraction**.

4. **The polygonal approximation**. Given a curve *C* we will find a polygon that closely approximates *C* while having as small a number of vertices as possible.

Many approaches for the delimiters extraction have been elaborated. All these methods have in common the 1st, 2nd and 4th step. The 3rd step represents the main difference in each case. We used two main approaches for the contour extraction:

- **The contour tracing for a given object** Once an object point has been identified, a contour tracing is performed starting from this point. The contour tracing stops when all contour points from the object are processed. During the contour traversing, each new point is stored in a list.

- **The border scanning** A radial scanning is performed. It exploits the elevation map from the Ego Car position. The scanning process is implemented with a given radial step, traversing the interest zone and accumulating the delimiters points in the same time. The main difference of this approach is that only the visible part can be scanned. The idea is that the occluded part does not represent relevant information in the delimiters detection process.

A possible disadvantage in the case of the contour tracing approach is that not all the forms of the obstacles can generate a good contour. In some cases, noisy contours can be extracted. There are cases when the same delimiter point can be passed and processed many times. In these cases, segments forming the resulting polygon can include same points. Another problem is when many forms are part of a single object. Therefore, a single obstacle delimiter cannot be extracted through the simple contour tracing method. For that reason, the border scanning approach was chosen as the extraction method in our system

2.2 The Border Scanner algorithm

The Border Scanner algorithm performs a radial scanning with a given radial step. The scanning axis moves in the radial direction, having a fixed center at the Ego Car position. The scanning process is made into the limits of *Q_from* and *Q_to* angles, thus only the interest area are scanned, where the delimiters can be detected (Fig. 2). Having a radial axis with a *Qrad* slope, $Q\_from \leq Qrad \leq Q\_to$, an object situated on this axis will be reached (the nearest point from the Ego Car). In this way, all subsequent

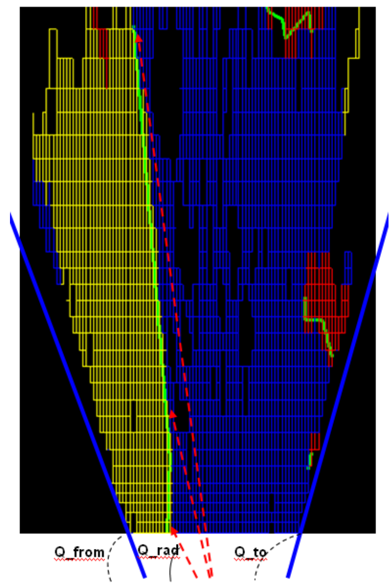

Figure 2. Border Scanner

points will be accumulated into a list named *Contour List*, moving the radial axis in the radial direction and having the *Qrad* slope included into the *Q_from* - *Q_to* limits. At each radial step we'll verify if a new object has been reached. If a new label has been found then the polygonal approximation on the ContourList points is performed. The list will be cleared, and the algorithm will be continued finding a new polygon.

The algorithm consists of a sequence of following steps:

```
1. Qrad = Q_from;
   The ContourList is empty;
2. Find an Object Point
   IF (We have a new object label)
   THEN BEGIN
     Polygonal Approximation;
     Clear the ContourList;
   END;
   ELSE
     Store the new Object Point
     into the ContourList;
3. Qrad = Qrad + Qstep;
4. IF (Qrad < Q_to) THEN
   GO TO step 2.
5. IF (the ContourList is not empty)
   THEN Polygonal Approximation;
6. END.
```

The advantages of border scanning method are:

- The obtained results are more similar to the real obstacle delimiters from the scene.

Table 1. Variable Step Border Scanner and Fixed Step Border Scanner

	Fixed step Border Scanner	**Variable Step Border Scanner**
Number of Frames	203	203
Detected points	4529	5733
The Step (radians)	0.01	0.01
Points per Frames	22	28

- The problem of the compound objects presented previously is eliminated. Therefore more complex patches can be enveloped by a single delimiter. The condition is these patches need to belong to the same object (they have the same label).

2.3 The Border Scanning algorithm using variable step

If we consider the radial step to be constant, then the detected pixel density will decrease with the Z distance. The distance between two consecutive detected pixels is greater at the far Z values. The idea is that some important information about the delimiters can be lost at the far distances. A good solution is to use a scanning method with a variable step, thus the radial step should be adapted at once with the distance. If we have found a pixel at a far distance from the observer, the radial step could be decreased. Therefore the radial step varies with the distance.

In the Table 1 the results from the Variable Step Border Scanner and Fixed Step Border Scanner are shown for the same driving scene. It can be observed that the number of detected points is greater in the case of Variable Step Border Scanner algorithm, thereby 5733 points, which means 28 detected points per frame in comparison with 22 detected points per frame in the case of Fixed Step Border Scanner algorithm.

2.4 The Combined Border Scanner algorithm

Another aspect is that many relevant objects delimiters may be omitted if we take in account only the first nearest point from the car. Many times, if a radial scanning is performed, the first obstacle from the car can be a curb. In this case we are interested not only in the curb delimiters but also in the delimiters above the curb or behind the curb. Also we are not interested in the cases of the curbs occluded by the other obstacles. We have elaborated an improved version of the Border Scanner algorithm which is Combined Border Scanner algorithm. In the Combined Border Scanner algorithm we take in account the obstacle's nature making a decision based on two types of information "What have we found?" and "What we have to find?" The algorithm consists in two passes: one for the Object delimiters detection, and second for the curb delimiters detection.

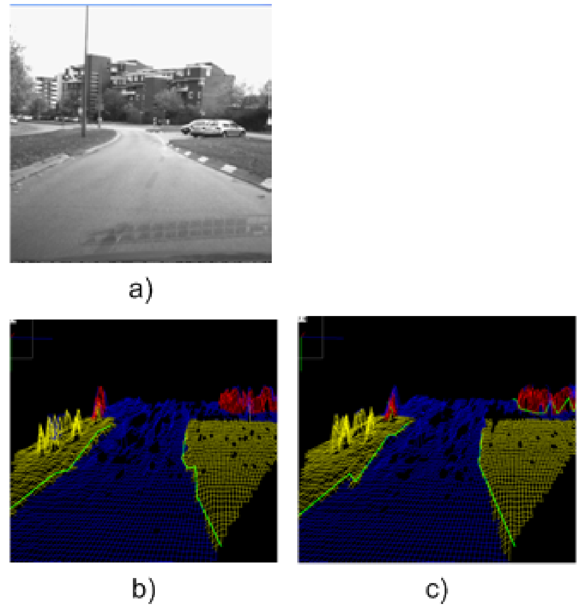

Figure 3. a) Left camera image. b) Simple Border Scanner c) Combined Border Scanner

The Fig. 3 presents the difference between the result of delimiters detection in the case of Simple border Scanner and Combined Border Scanner algorithms. It can be observed that in the case of Combined Border Scanner, the car's delimiter is detected in spite of his position behind the curb.

3 The Drivable Tunnel Model

The drivable tunnel represents a virtual area around the ego-car trajectory and is generated by the mechanical and movement characteristics of the car. It can be perceived as a safety zone for the ego-car. The length of the trajectory must be greater or equal to the car braking distance.

At the moment t, the car is moving on a circular arc, with radius of curvature fR (Fig. 4). The width of the tunnel is tW. The 3D model of the tunnel is described by a polyhedron with the following characteristics:

- It is a non-convex polytop - there exists a line connect-

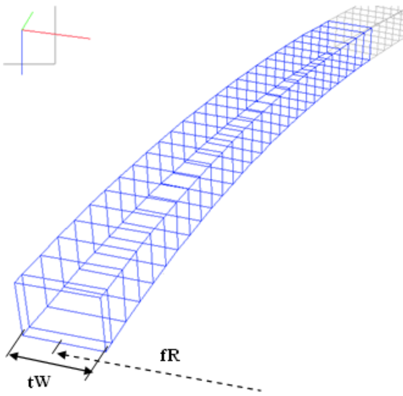

Figure 4. The Drivable Tunnel Model

ing two points inside the polyhedron and does not lie entirely within the polyhedron

- It is a structured model - there is a polygon collection forming a closed topological space for which connectivity information is available. In this model, two faces are intersected by a single edge.

This polyhedron can be decomposed into small cells (Fig. 4). Each cell is a hexahedron with two pairs of parallel faces: $Left$ and $Right$, $Bottom$ and Top. The $Near$ and Far faces are perpendicular to the car displacement trajectory. Any two adjacent cells have a common face. Thus, the Far face of one cell represents the Near face of the next cell.

4 Collision Detection

The collision detection problem in the case of models presented above can be related to the polygon clipping algorithms described in [18], which takes as input the vertices of a polygon and returns one or more polygons. Because the polygon representation is a list of vertices, the polygonal clipping can be done by polygon edge-by-edge passing. According to the Cohen-Sutherland Line Clipping algorithm [18], a line is clipped to an upright rectangular window. In our case, the polygonal clipping problem is extended by using as a clipping region the tunnel projection represented in the figure 5.

The projected tunnel trajectory describes a circular arc with a radius R and the center coordinates $C(x_0, y_0)$. The set of right edges are equidistant from the circle center. The right edges form chords whose endpoints lie on a circle with a R_1 radius. In the same way the left edges form chords whose endpoints lie on a circle with a R_2 radius.

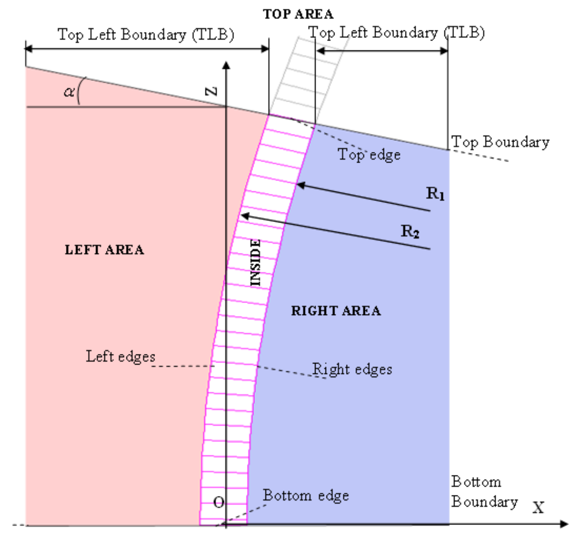

Figure 5. Drivable Tunnel model projected on the xOz plane (top view)

The both arcs, right and left, that describe the drivable tunnel boundaries are comprised between two boundaries: $BottomBoundary$ and $TopBoundary$, respectively. The two circles with the R_1 and R_2 radiuses are concentric circles. The Bottom Boundary represents a line, which is parallel to OX axis. The mathematical relation is:

$$z = Rel2Ego \tag{1}$$

where $Rel2Ego$ is the distance between Ego-Car and $BottomEdge$. The $TopBoundary$ describes a line that forms an angle α with the $BottomBoundary$. Moreover, the angle α is a central angle, which subtends the circular arcs comprised between $BottomBoundary$ and $TopBoundary$. Taking into account the tunnel top view from figure 5, we define four areas, which will help us to collision detection:

1. $RightArea$ - is the area, which is delimited from the tunnel by the right edges. This area is bounded by the tunnel edges on the right side. Having the circle center coordinates $C(x_0, y_0)$, two representative cases are distinguished for the mathematical representation:

   For any $x_0 < 0$ (the radius of curvature is negative), Right Area is given by the following mathematical constraints:

$$\begin{cases} (x - x_0)^2 + (y - y_0)^2 > R_1^2 \\ z < (x - x_0)tan\alpha + z_0 \end{cases} \tag{2}$$

   For any $x_0 > 0$:

$$\begin{cases} (x - x_0)^2 + (y - y_0)^2 < R_1^2 \\ z < (x - x_0)tan\alpha + z_0 \end{cases} \tag{3}$$

2. $LeftArea$ - is the area that is separated from the tunnel by the left edges, and is located at the left of the tunnel. Having the circle coordinates $C(x_0, y_0)$, two cases are distinguished for the mathematical representation, based on the circle centers position that describe the two tunnel arcs (left and right):

   For any $x_0 < 0$ (the radius of curvature is negative):

$$\begin{cases} (x-x_0)^2 + (y-y_0)^2 < R_2^2 \\ z < (x-x_0)tan\alpha + z_0 \end{cases} \tag{4}$$

   For any $x_0 > 0$:

$$\begin{cases} (x-x_0)^2 + (y-y_0)^2 > R_2^2 \\ z < (x-x_0)tan\alpha + z_0 \end{cases} \tag{5}$$

3. $InsideArea$ - is the area from the inside of the tunnel. From the mathematical point of view, for each point situated in this area we have the following conditions:

   For any $x_0 < 0$:

$$R_2^2 < (x-x_0)^2 + (y-y_0)^2 < R_1^2 \tag{6}$$

   For any $x_0 > 0$:

$$R_1^2 < (x-x_0)^2 + (y-y_0)^2 < R_2^2 \tag{7}$$

4. $TopArea$ - is the area located above the Tunnel Top:

$$z > (x-x_0)tan\alpha + z_0 \tag{8}$$

The collision detection process involves determining the position of each delimiter vertex, having defined the four areas (Fig. 5). For each polygonal edge, we assign an edge direction based on the endpoints position. Depending on the edge direction, the intersection can be made with one of the polygonal edges. We assume that the intersection of $BottomEdge$ and $Delimiter$ did not take place, because the Bottom Edge is situated near the $Ego-Car$, where no 3D points are reconstructed. When delimiter edge intersects the right or left side of the tunnel, the result may be a point located on any chords of the tunnel. In this case, we can adopt a linear approach to search the result by intersecting each chord with the current delimiter.

5 Experimental Results

We have used a set of 25 scenarios from the urban traffic environment for the test purpose. In the Fig. 6 we present the Top View (Fig. 6 b) and Free Look images (Fig. 6 d) containing two cases of forward collision detection. The delimiters parts being in collision with the Drivable Tunnel are presented with the red color. Figure 7 shows a case

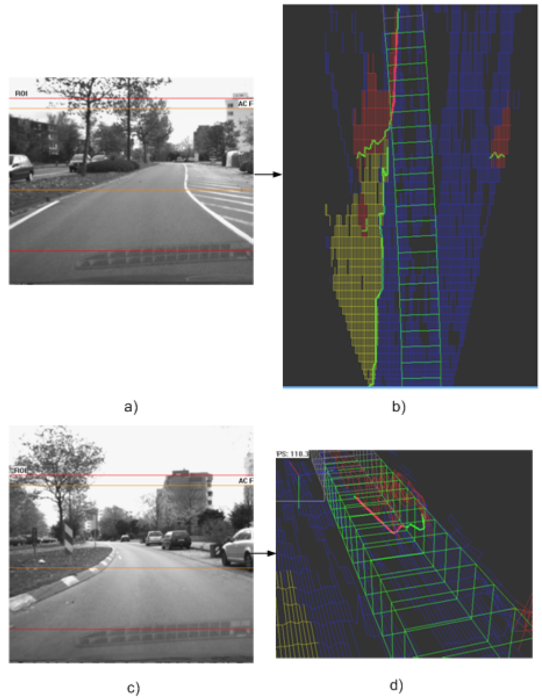

Figure 6. Collision Detection: a) Left camera image. b) Collision detection - Top View. c) Left image. d) Collision detection - Free Look.

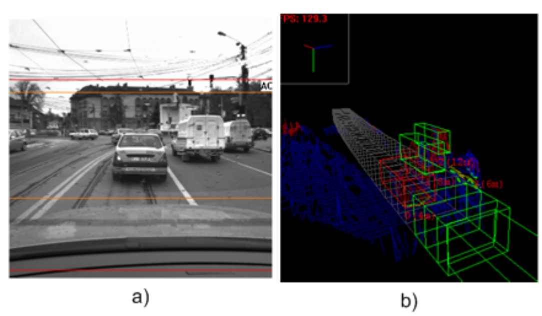

Figure 7. Collision Detection: a) Left camera image. b) The generated tunnel does not include any delimiter

when for the car in front no messages is generated because the relative speed between the two cars is too small and the detected delimiters are outside the generated tunnel.

It must be noted that our FCW algorithm is performed at the higher level in our architecture. Therefore the accuracy of our FCW system depends on the 3D information quality provided by dense stereo, obstacles detection algorithms, and Elevation Map result.

For generating the experimental results we used a 2.66GHz Intel Core 2 Duo Computer with 2GB of RAM.

The average execution time for the FCW module is about 6ms using a radial step of 0.01 radians for the delimiters extraction algorithm. This performance can be improved by further optimizations.

6 Conclusions

We have presented a Forward Collision method which uses 3D information from a Stereo Vision System. The proposed algorithm takes as input the generated information from the Elevation map and ego-car mechanical and moovement parameters such as yaw rate, car speed and steering angle. We have extracted the delimiters of detected objects through the Combined Radial Scanning approach and have built a 3D tunnel which length depends on the car relative velocity. The collision detection was made taking into account these 3D models. Various scenarios from the urban traffic environment have been used.

References

[1] E. C. for Europe, "Statistics of road traffic accidents in europe and north america," 2007. [Online]. Available: http://www.unece.org/trans/main/wp6/pdfdocs/RAS_2007.pdf

[2] J. Treat, "Tri-level study of the causes of traffic accidents: Final report volume 1," May 1979.

[3] Hendricks and Freedman, "The relative frequency of unsafe driving acts in serious traffic crashes," 2001. [Online]. Available: http://www.nhtsa.dot.gov/people/injury/research/UDAshortrpt/UDAsummtechrept.pdf

[4] N. T. S. Board, "Special investigation report - highway vehicle and infrastructure-based technology for the prevention of rear-end collisions," may 2001. [Online]. Available: http://www.ntsb.gov/Publictn/2001/SIR0101.pdf

[5] J. D. Woll, "Vorad collision warning radar," in *Radar Conference, Record of the IEEE 1995 International*, may 1995, pp. 369 – 372.

[6] Z. Lei, W. Jianqiang, and L. Keqiang, "Forward collision warning system based on thasv-ii platform," in *Vehicular Electronics and Safety, 2006. ICVES*, Dec. 2006, pp. 255 – 258.

[7] E. Dagan, O. Mano, G. P. Stein, and A. Shashua, "Forward collision warning with a single camera," in *Intelligent Vehicles Symposium, 2004 IEEE*, June 2004, pp. 37–42.

[8] R. J. Kiefer, "Forward collision warning requirements project final report - task 1," 2003. [Online]. Available: http://ntl.bts.gov/lib/23000/23000/23010/HS809574Report.pdf

[9] D. J. Lee, R. W. Beard, P. C. Merrell, and P. Zhan, "See and avoidance behaviors for autonomous navigation," in *SPIE Optics East, Robotics Technologies and Architectures, Mobile Robot XVII, vol. 5609-05, Philadelphia, PA, USA*, October 25-28 2004.

[10] I. Gat, M. Benady, and A. Shashua, "A monocular vision advance warning system for the automotive aftermarket," in *SAE 2005 World Congress & Exhibition, Detroit, MI, USA*, April 2005.

[11] A. Polychronopoulos, M. Tsogas, A. Amditis, U. Scheunert, L. Andreone, and F. Tango, "Dynamic situation and threat assessment for collision warning systems: The euclide approach," in *in proceedings of the IEEE Intelligent Vehicles Symposium, Parma, Italy*, June 14-17 2004, pp. 636–641.

[12] S. Rhl, M. Grinberg, and D. Willersinn, "Empirical evaluation of motion models for a side-looking driver assistance system," in *Proceedings of the 5th International Workshop on Intelligent Transportation (WIT 2008), Hamburg*, Mar. 2008.

[13] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, and M. A. Obojski, "A sensor for urban driving assistance systems based on dense stereovision," in *Proceedings of Intelligent Vehicles 2007,Istambul*, 13-15 June 2007, pp. 278 – 286.

[14] J. I. Woodfill, G. Gordon, , and R. Buck, "Tyzx deepsea high speed stereo vision system," in *in Proceedings of the IEEE Computer Society Workshop on Real Time 3-D Sensors and Their Use, Conference on Computer Vision and Pattern Recognition*, 2004.

[15] C. Pocol and S. Nedevschi, "Obstacle detection for mobile robots, using dense stereo reconstruction," in *EEE International Conference on Intelligent Computer Communication and Processing (ICCP'07), Cluj-Napoca, Romania*, September 6-8 2007, pp. 127–132.

[16] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, T. Graf, and R. Schmidt, "Driving environment perception using stereovision," in *Procedeeings of IEEE Intelligent Vehicles Symposium, (IV2005), Las Vegas, USA*, June 2005, pp. 331 – 336.

[17] F. Oniga, S. Nedevschi, M. M. Meinecke, and T. B. To, "Road surface and obstacle detection based on elevation maps from dense stereo," in *Proceedings of in IEEE Intelligent Transportation Systems Conference, Seattle, WA*, 2007, pp. 859 – 865.

[18] W. Shoaff, *Clipping*, March 12 2002. [Online]. Available: http://cs.fit.edu/w̃ds/classes/graphics/Clip/clip.pdf