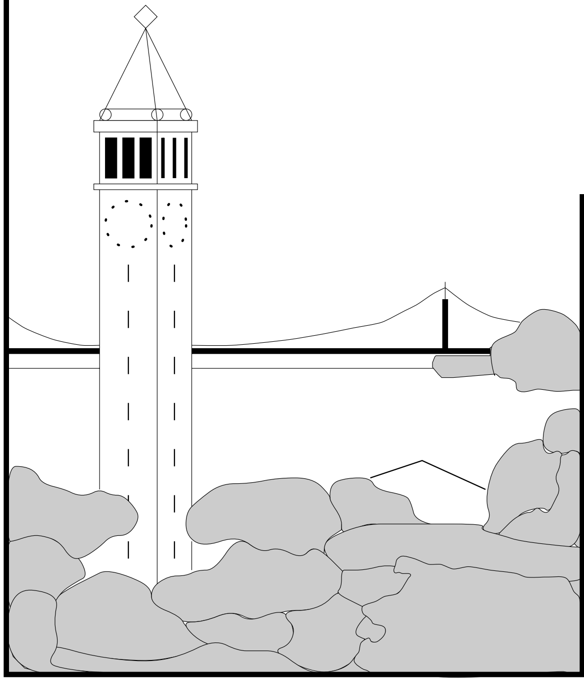
Binocular Stereopsis and Lane Marker Flow for Vehicle Navigation: Lateral and Longitudinal Control

Dieter Koller, Tuan Luong, and Jitendra Malik



Report No. UCB/CSD 94-804

March 24, 1994

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Binocular Stereopsis and Lane Marker Flow for Vehicle Navigation: Lateral and Longitudinal Control[*]

Dieter Koller, Tuan Luong, and Jitendra Malik
University of California, CS Division, Berkeley, CA 97720
{koller,qtluong,malik}@robotics.eecs.berkeley.edu

March 24, 1994

Abstract

We propose a new approach for vision based longitudinal and lateral vehicle control which makes extensive use of binocular stereopsis. Longitudinal control — i.e. maintaining a safe, constant distance from the vehicle in front — is supported by detecting and measuring the distances to leading vehicles using binocular stereo. A known camera geometry with respect to the locally planar road is used to map the images of the road plane in the two camera views into alignment. Any significant residual image disparity then indicates an object not lying in the road plane and hence a potential obstacle. This approach allows us to separate image features into those lying in the road plane, e.g. lane markers, and those due to other objects. The features which lie on the road are stationary in the scene and appear to move only because of the egomotion of the vehicle. Measurements on these features are used for dynamic update of (a) the camera parameters in the presence of camera vibration and changes in road slope (b) the lateral position of the vehicle with respect to the lane markers. In the absence of this separation, image features due to vehicles which happen to lie in the search zone for lane markers would corrupt the estimation of the road boundary contours. This problem has not yet been addressed by any lane marker based vehicle guidance approach, but has to be taken very seriously, since usually one has to cope with crowded traffic scenes where lane markers are often obstructed by vehicles. Lane markers are detected and used for lateral control, i.e. following the road while maintaining a constant lateral distance to the road boundary. For that purpose we model the road and hence the shape of the lane markers as clothoidal curves, the curvatures of which we estimate recursively along the image sequence. These curvature estimates also provides desirable look-ahead information for a smooth ride in the car.

[*] Research funded by PATH grant no. MOU 94

Contents

1 Introduction

We propose an approach and develop a system for vision based longitudinal and lateral vehicle control which makes extensive use of binocular stereopsis. Novel aspects include (a) exploitation of domain constraints to simplify and make robust the search problem in finding binocular correspondences (b) dealing with crowded traffic scenes where substantial segments of the lane boundaries may be occluded. The vision system is designed to interface in a modular fashion with the use of non-visual sensors such as magnetic sensors for lateral position measurement and active range sensors (e.g. Doppler radar) for an integrated approach to vehicle control such as that being investigated in the California PATH project.

Longitudinal control — i.e. maintaining a safe, constant distance from the vehicle in front — is supported by detecting and measuring the distances to leading vehicles using binocular stereopsis. A known camera geometry with respect to the locally planar road is used to map the images of the road plane in the two camera views into alignment. Any significant residual image disparity then indicates an object not lying in the road plane and hence a potential obstacle. This approach allows us to separate image features into those lying in the road plane, e.g. lane markers, and those due to other objects. The features which lie on the road are stationary in the scene and appear to move only because of the egomotion of the vehicle. Measurements on these features are used for dynamic update of (a) the camera parameters in the presence of camera vibration and changes in road slope (b) the lateral position of the vehicle with respect to the lane markers. In the absence of this separation, image features due to vehicles which happen to lie in the search zone for lane markers would corrupt the estimation of the road boundary contours. This problem has not yet been addressed by any lane marker based vehicle guidance approach, but has to be taken very seriously, since usually one has to cope with crowded traffic scenes where lane markers are often obstructed by vehicles. Lane markers are detected and used for lateral control, i.e. following the road while maintaining a constant lateral distance to the road boundary. For that purpose we model the road and hence the shape of the lane markers as clothoidal curves, the curvatures of which we estimate recursively along the image sequence. These curvature estimates also provides desirable look-ahead information for a smooth ride in the car.

1.1 Related Research

A number of different sensing technologies have been proposed for use in an Advanced Vehicle Control System. These include vision, magnetic sensors for lateral position measurement and active range sensors (e.g. Doppler radar).

By far the most important and impressive work on a visually guided AVCS has been done in the group of Prof. E.D. Dickmanns of Universität der Bundeswehr, Munich, Germany. See [Dickmanns & Mysliwetz 92] and the references cited therein. Their work resulted in a demonstration in 1987 of their 5-ton van, the VaMoRs running autonomously on a stretch of the Autobahn at speeds of upto 100 km/h. Vision was used to provide input for both lateral and longitudinal control on free roads. Subsequently they have demonstrated successful operation on cross-country roads (at lower speeds) where the road boundaries are difficult to determine.

Further development of this work has been in collaboration with von Seelen's group in Bochum [Schwartzinger *et al.* 92] and the Daimler Benz VITA project [Ullmer 92]. For collision avoidance with vehicles in one's lane, model-based techniques are used which exploit heuristics such as symmetry of the bounding box of the vehicle, which is based on the fact that rear or the front views of most of the vehicles exhibit a strong vertical symmetry. It seems to us that these techniques are not as reliable as those using binocular stereopsis which enables obstacles to be defined in a very general way. Also, for longitudinal control under platooning conditions such as being studied in the PATH project, one would need a much higher precision for estimating distances to other vehicles.

While Dickmanns's project has resulted in the most impressive demonstration of vision-based vehicle guidance to date, it is only one of the numerous sites where research in this area is being conducted. In the United States, most of the significant projects are supported by the Department of Defense and the desired capability is driving on cross-country terrain. The CMU NavLab project [Thorpe 90] is the key university-based project on this theme with major activity also at sites such as Martin Marietta at Denver. In the Navlab project a number of different lane following algorithms have been used. Currently the favored one seems to be the ALVINN system [Pomerleau 92] based on training a neural network with input a 30x32 low resolution image and output the desired steering command. The best performance cited is that of a 21.2 mile run at speeds of upto 55 miles per hour. The network performs reasonable well if it is trained with similar road conditions. In order to overcome the problem on which network to use, they recently proposed a connectionist superstructure — MANIAC — which incorporates multiple ALVINN networks, each of which is pretrained for a particular road type ([Jochem *et al.* 93]). Their hope is that the superstructure will learn to combine data from each of the ALVINN networks and not simply select the best one.

A more recent project on road following in the US is a collaboration between NIST and Florida Atlantic University[Raviv & Herman 91]. Lateral control is based on sensing the optical flow at a certain tangent point on the lane and steering so as to make it have no horizontal component. There may be difficulties if the tangent point is occluded.

A basic module in any of the visually guided vehicle control algorithms has to be the detection and tracking of lane boundaries. A leading project is the LANELOK system developed at GM Research. Continuing work has resulted in a real-time implementation reported in [Altan *et al.* 92].

In Japan, research is being conducted at a number of industrial and academic research laboratories. These include the Harunobo project at Yamanashi University ongoing since 1982 which has resulted in an autonomous vehicle tested on roads. Industrial sites include Toyota, Nissan, Honda, Matsushita etc. For an extensive survey, we recommend the proceedings of the IEEE Conference on Intelligent Vehicles 1990 [Masaki 92b] and 1992 [Masaki 92a]

The use of binocular stereopsis for vehicle control has been successfully demonstrated by JPL's planetary robotic vehicle [Matthies 92] and Nissan's PVS vehicle

[Ohzora *et al.* 90]. Both systems realize a tradeoff between performance time and density of a depth map. For obstacle detection it is actually not necessary to compute a dense depth map neither is it necessary to perform the depth map computation in video rate. A trinocular stereo system is used by [Ross 93], where the third camera actually serves as a mean to confirm and refine the results obtained from two cameras.

Research closest related to our obstacle detection approach is described in [Zheng *et al.* 90]. They perform first a rectification of the stereo images to achieve zero vertical disparity, before they estimate the disparity by comparing the variances of the difference in a window of the rectified left and right image. The notion SWITCHER of their approach refers to an F-test, which is applied to the variances associated to the disparities, in order to decide whether there is a significant change in disparity which would indicate an obstacle. Although they exploit the full camera geometry, they do not use the knowledge of the ground plane disparity to reduce the search space, neither does their approach apply to lane marker detection. They furthermore admit that their approach is quite computationally expensive.

Other, more classical approaches for obstacle detection are based on motion stereo or optical flow interpretation ([Enkelmann 90; Carlsson & Eklundh 90]). The key idea of these approaches is to predict the optical flow field for a moving observer under constraint motion (e.g. planar motion). Obstacles are then detected by a significant difference between the predicted and the actual observed optical flow field. The major drawbacks of these approaches are (a) computational expense and (b) the lack of reliable and accurate optical flow fields and the associated 3D data (it is well known that structure-from-stereopsis approaches perform better than structure-from-motion approaches).

A combination of stereo and optical flow is suggested in [Chandrashekar *et al.* 91] in order to perform a temporal analysis of stereo image sequences of traffic scenes. They do not explicitly address the problem of obstacle detection in the context of vehicle guidance, but the general problem of object identification. They extract and match contours of significant intensity changes in (a) stereo image pairs for 3D information and (b) subsequent frames to obtain their temporal displacement ([Meygret *et al.* 92; Meygret & Thonnat 90b]). Object descriptions are finally obtained by grouping the Kalman filtered 3D trajectories of these contours using a constant image velocity model. In order to distinguish between obstacles and road boundaries or lane markers, they also exploit some heuristics like horizontally and vertically aligned contour segments as well as 3D information extracted from the stereo data ([Meygret & Thonnat 90a]).

While the most successful lane marker based approaches for lane following perform quite well in uncrowded traffic scenes, where lane markers are clearly visible and not obstructed by other vehicles, we expect them to fail or at least to perform not so well in crowded traffic scenes, where lane markers *are* obstructed by other vehicles. On the other hand we expect our approach to perform reasonable well even in crowded scenes, since we explicitly distinguish and reason about lane markers and obstacles, lying in the search region for lane markers.

The rest of this paper is organized as follows: In Section 2 we outline our system approach for lane following and lane changing. Section 3 is dedicated to the longitudinal vehicle control based on binocular stereopsis. Our lateral control approach based on lane marker flow is introduced in Section 4. Our experimental and computational setup is explained in Section 5, where we also present the camera calibration procedures. Some results are finally shown in Section 6.

2 System configuration

In this section we want to elucidate our basic ideas and the concept of the system design being developed in the course of this project, in order to give the reader an idea of the problem and to bring the whole project in a larger context. The system configuration and the used algorithms have to meet two major goals: robustness and realtime performance. Real time performance in the context of computer vision means that the processing time of a single image frame acquired by the vision sensor is of the order of the videorate, i.e. at least 5-10 Hz. The robustness of the proposed system is based on fusion of data from different sensors: vision, the magnetic sensor for lateral control, the velocity signal derived from the antilock braking system (ABS) and Doppler radar for detecting obstacles and measuring their distances to the car.

Figure 1 shows a flow chart of the proposed system with the interaction of other sensors. This diagram shows the flow of information for keeping the car inside a certain lane. Our computer vision system uses the same output of the multichannel filterbank for the detection of lane markers and the computation of stereopsis. The task of the different components is to provide the vehicle control system with additional information for lateral and longitudinal control. The importance of this — in the first view redundant — information becomes obvious when planning a lane change during which the other sensors are expected to yield uncertain or even wrong or no information for the control variables.

Lane change maneuvers

In the platooning concept being studied in the PATH project a lane change is required in case a car wants to merge into a platoon or to leave a platoon. In either case of a desired lane change, the vehicle control system has to initiate all safety checks and communications with the involved cars before starting a lane change maneuver. This includes a check of neighboring lanes as well as information about cars approaching from behind. After all safety checks have been successfully executed the vehicle control system can change the state from *lane following* into *lane change.* For that purpose a motion plan has to be computed with a sequence of control variables for the steering and throttle actuators. During the lane change maneuver the predicted control values for the steering and throttle actuators are compared with the sensor data in order to compute new updates. This control feedback keeps the control variables in an appropriate range during the maneuver and keeps the vehicle control system updated in case of an unexpected behavior. Figure 2 shows a flow chart of the proposed system for guiding a car during performing a lane change.

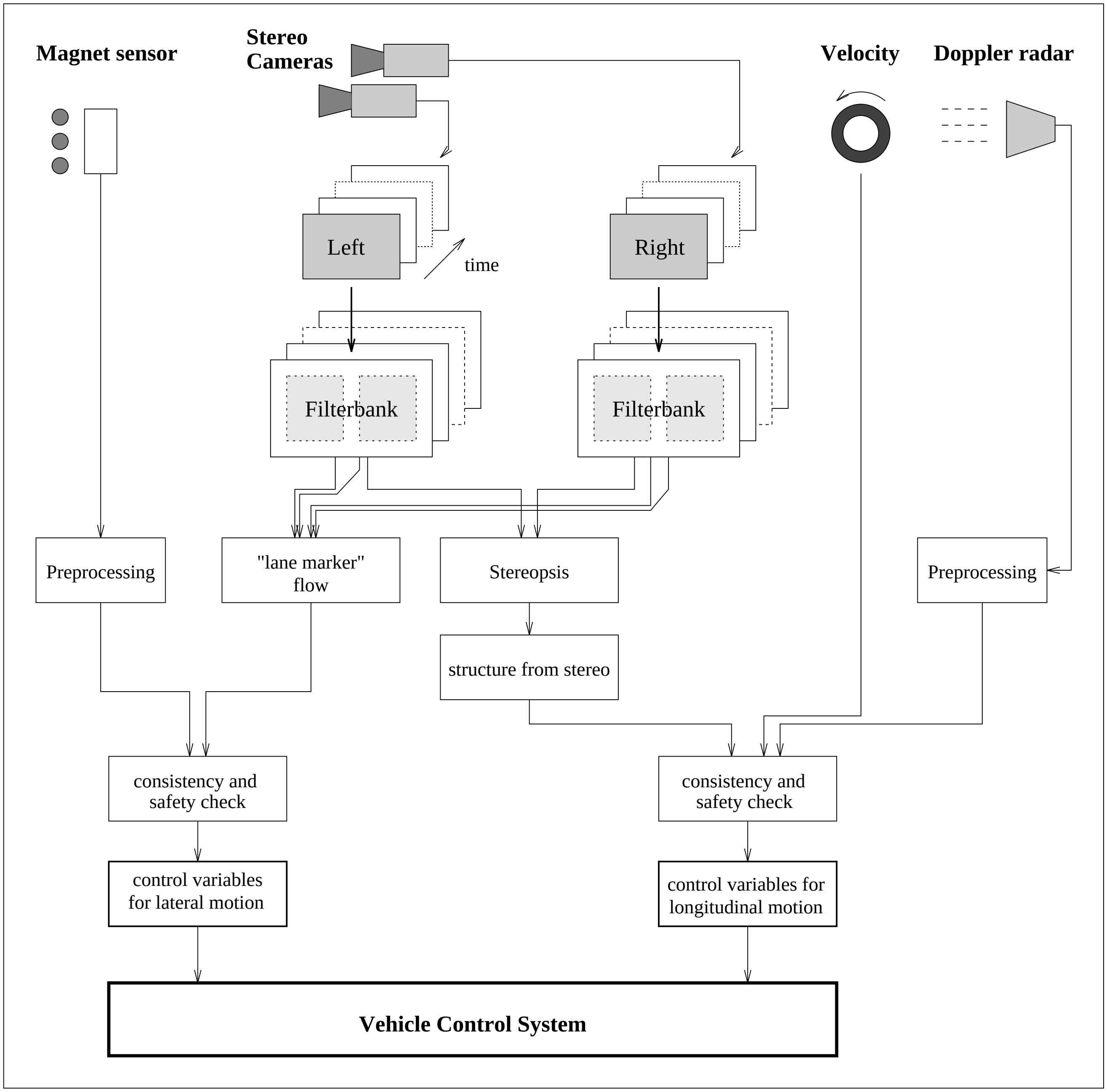


Figure 1: The flow chart of our proposed system for lane following.

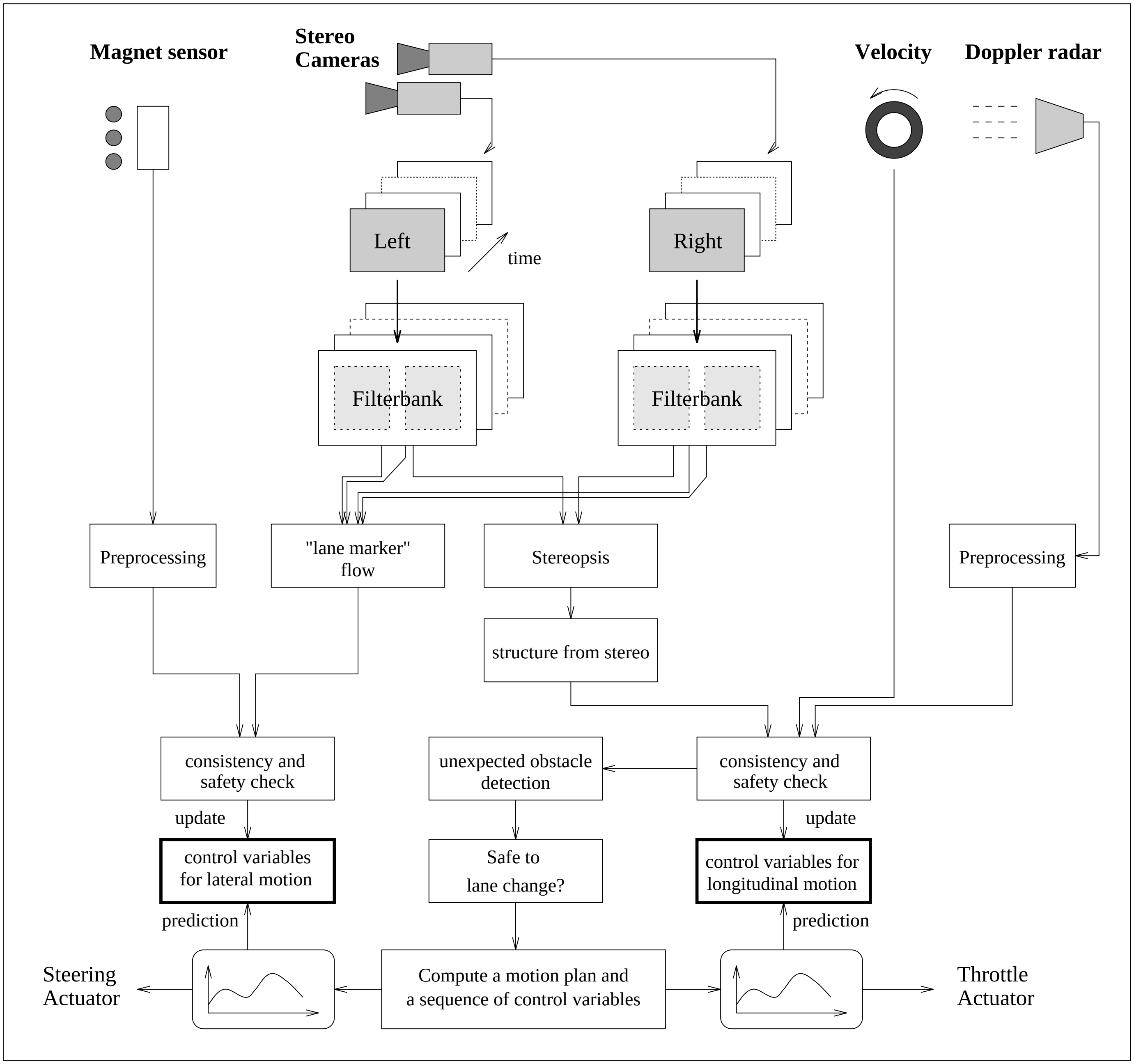


Figure 2: The flow chart of our proposed system for guiding a car during performing a lane change.

3 Longitudinal Control

In order to extract inputs for the computation of longitudinal control variables from images we have to estimate depth and relative velocities of objects appearing in front of the car as well as the velocity of the car itself. In order to obtain robust estimations of the 3D structure of the environment in front of the car we combine approaches based on different combinations of input vision sensor data:

1. Binocular stereopsis using the slight differences of locations of imaged 3D scene structures in the views of two spatially separated cameras.

2. Structure from planar motion using the temporal integration of movements of imaged 3D scene structures in subsequent frames of a single camera — the so-called optical flow.

Our previous work on these problems has been reported in [Jones & Malik 92; Weber & Malik 93; Koller *et al.* 93]. Stereo for vehicle control is used successfully by JPL's planetary robotic vehicle [Matthies 92] and Nissan's PVS vehicle [Ohzora *et al.* 90]. Both systems realize a tradeoff between performance time and density of a depth map. For obstacle detection it is actually not necessary to compute a dense depth map neither it is necessary to perform the depth map computation in video rate. A trinocular stereo system is used by [Ross 93], where the third camera actually serves as a mean to confirm and refine the results obtained from two cameras.

Although proposed algorithms in the literature for computing binocular stereopsis and optical flow are quite computationally expensive we are able to reduce the complexity considerably by using region-of-interest processing and exploitation of domain constraints. The first stage is based on a multichannel filtering approach which is completely parallelizable and hence realtime feasible. The algorithms appear robust and accurate enough to support further computations based on their outputs.

Using the extracted information for the depth and velocity we can formulate obstacle hypotheses which we are going to verify by temporal integration using predict and update formulations with standard Kalman filter techniques (e.g. [Bar-Shalom & Fortmann 88; Scales 85; Koller *et al.* 93]. A detailed spatial description of other vehicles is not necessary since the position and velocity of the vehicles in the nearest neighborhood seems to be sufficient for the task of the longitudinal control as well as for deciding whether a lane change can be performed or not (at the current stage it makes no difference whether the obstacle is a motorcycle, car or truck).

The output of the depth and velocity estimations have to be checked for consistency with the output of the Doppler radar and the velocity from the ABS system. The combination of all of these sensor outputs provide a robust estimation of the control variables for longitudinal motion and detects a failure of one of the inputs. More detailed information — which we are going to expect from the vision sensor because of its capability for looking ahead — can be used to detect unexpected obstacles and signaled to the vehicle control system in order to decide an appropriate behavior.

In the following Subsection 3.1 we briefly state the problem of estimating depth using stereopsis and describe how one can exploit certain constraints and a special camera geometry in order to reduce the computational complexity. It is well known that structure from stereo yield more accurate and robust results than structure from motion using optical flow.

Our results of computing depth from stereo are quite encouraging and good enough for use in longitudinal control (see Subsection 5). Computing robust optical flow from video sequences taken by a camera fixed on a car driving on a highway causes some problems; a) the car usually experiences significant vertical motions due to the suspension, and b) the visual image motion of the ground plane is quite large (in the order of 40 pixels between two subsequent frames). After extensive research in this area we conclude that we will use only stereopsis for depth estimation and longitudinal control. Depth estimation using stereopsis is quite robust against vertical motion and shaking, since only stereo pairs at a certain time instant are used in the computation. A temporal integration can even be used for estimating the vertical motion.

3.1 Stereopsis

A stereo setup for retrieving 3D structure of the environment seems quite obvious, but is not yet used for realtime applications like this because the standard algorithms are computationally expensive. We have developed a very simple and highly efficient algorithm, starting with [Jones & Malik 92] and exploiting domain constraints with region-of-interest processing to gain a major speedup.

In order to retrieve depth information from the images we compute the stereo disparity between the left and right view of the cameras, i.e. we look for the shift of the location of imaged scene features in the two images (c.f. Figure 14a) and b)). The stereo disparity d of an imaged scene feature depends from the camera baseline b (the spacing of the camera center) and the depth Z, i.e. the distance of the scene feature from the camera and the focal length f (see Eqn. 1).

$$d = f\frac{b}{Z} \qquad \begin{array}{l} d : \text{disparity} \\ f : \text{focal length} \\ b : \text{camera baseline} \\ Z : \text{depth (distance camera—object)} \end{array} \qquad (1)$$

From this equation we see that for small δZ we get $\delta d = \frac{fb}{Z^2}\delta Z$ and hence $\delta Z = \frac{\delta d}{fb}Z^2$, which means that the error in depth estimation increases quadratically with the depth. This fits nicely with our requirements, since the nearer the objects the more important they are.

Computing the disparity actually requires solving some correspondences of image features in the left and right view. The process of computing the stereo disparity is tremendously simplified by using an insight due to Helmholtz ([Helmholtz 25]) more than a hundred years ago, but not yet exploited in computer vision. Helmholtz observed that objectively vertical lines in the left and the right view perceptually appear slightly rotated which led him to the hypothesis that the human brain performs a shear of the retinal images in order to map the ground plane to zero disparity. The mathematical reasoning is as follows: Under the viewing geometry of parallel axes, disparity for points on the ground plane has a zero value on the horizon and increases linearly with the image plane coordinate in the vertical direction. By applying a constant shear (the amount is a function of the distance between the eyes and height above the ground plane) one can map all points on the ground plane to have zero disparity. Then, any object above the ground plane will have non-zero disparity. This is very convenient because the human visual system is most sensitive around the operating point of zero disparity.

We can apply the same idea in our context–the advantage is that obstacles get mapped to points of non-zero disparity making them very easy to detect. See Figure 14 where (a) and (b) are a stereo image pair, and (c) and (d) show the points of interest on the sheared left and on the right image. Significant disparities correspond to obstacles. From the amount of the disparity at a certain image location we can simply estimate the distance (cf. Eqn. 1).

This *ground plane disparity removal* can be easily be shown for a special stereo camera set-up, where the optical axis is parallel to the ground plane. From Figure 3a) we obtain:

$$\begin{aligned} x'_l &= f\frac{-b/2 - X}{Z_l} \\ x'_r &= f\frac{b/2 - X}{Z_r} \\ \text{Stereo Disparity} \quad \Delta x' &= x'_r - x'_l = f\frac{b}{Y_w}, \end{aligned} \tag{2}$$

with $Y_w = Z_w = Z_l = Z_r$. The primed coordinates ($'$) are the image coordinates of the point (X_c, Y_c, Z_c) in camera coordinates. Using

$$y' = f\frac{t_z}{Y_w}, \tag{3}$$

with t_z the height of the camera above the ground plane from Figure 3b), the disparity for a constant t_z is a linear function of the y-coordinate in the image:

$$\Delta x' = \frac{b}{t_z}\, y'. \tag{4}$$

We prove a similar result for the more general view with an arbitrary inclination angle α (see Appendix A.1):

$$\Delta x'_0 = \frac{b\ f\ \cos\alpha}{t_z} - \frac{b\ \sin\alpha}{t_z}\, y', \tag{5}$$

which yields a ground plane disparity $\Delta x'$ as a linear function of the y'-coordinate with constant coefficient as long as we use a rigid stereo camera rig with constant base line and height as well as constant inclination angle α.

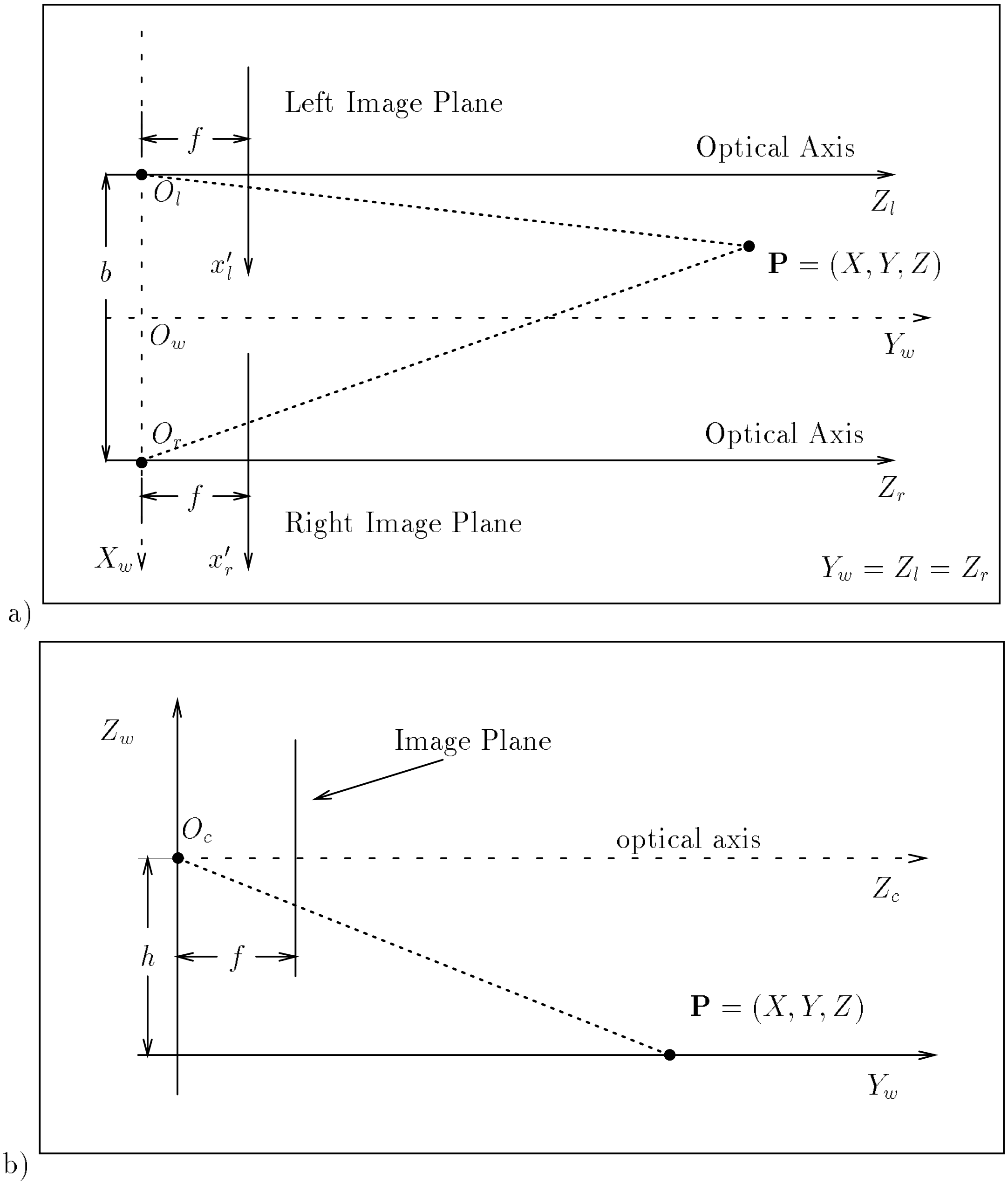


Figure 3: Top view (a) and side view (b) of a simple stereo camera set-up with the optical axis Z_c is parallel to the ground plane $X_w - Y_w$.

As we see from equation 1 we obtain zero disparity for the horizon ($Y_w = Z_w \rightarrow 0$). We obtain the horizon line $y' = y'_h$ for the more general view from equation 5 for $\Delta x' \rightarrow 0$:

$$y'_h = \frac{f}{\tan \alpha}. \tag{6}$$

The only drawback of this approach is the sensitivity to the camera adjustment: we assume the stereo cameras are adjusted in such a way that epipolar line (the line of sight in one images which corresponds to the projected point in the other image) appears horizontal in both images. In this way the search for correspondences can be reduced to horizontal scanlines in the images.

3.2 Disparity Computation

We perform this *residual* disparity computation[1] on a reduced set of points-of-interest on horizontal scanlines in the image. These points-of-interest are simply the locations in the image at which the gradient of the image function in horizontal direction is above a certain threshold. For that purpose we shift a correlation window for each point-of-interest along a horizontal scanline and compute the summed-squared-difference between the left and the right image (see Figure 4). The integer disparity estimate at the horizontal image location x is:

$$d(x) = \min_{\{\tau\}} \sum_{u=-W/2}^{+W/2} \|g_1(x+u+\tau) - g_2(x+u), \tag{7}$$

which is the minimum of the summed-squared-difference (SSD) of a window W shifted along a horizontal scanline with at least the expected disparity.

After the integer displacement has been obtained we apply a quadratic interpolation around the minimum in order to obtain sub-pixel accuracy (see Appendix A.2). Some results are given in Figure 14.

3.3 Depth Estimation

For estimating the depth — or more precise the 3D location of a point — we use Equation 7, which relates the image disparity for features on the ground plane to the normal distance t_z. We can use this equation also to compute the normal distance to a virtual plane parallel to the ground plane and which contains the feature which give rise to the observed image disparity. This distance can the be expressed in the height of the feature in world coordinates. From Equation 7 we get:

$$t_z = \frac{b}{\Delta x'_0}(f\cos\alpha - y'\sin\alpha), \tag{8}$$

which is the height of the camera expressed in the expected image disparity $\Delta x'$. Features above the ground plane appearing in the same image location are actually nearer to the camera and originate a larger disparity $\Delta x' = \Delta x'_0 + \delta$, which is:

$$t_z - \tilde{Z}_w = \frac{b}{\Delta x'_0 + \delta}(f\cos\alpha - y'\sin\alpha), \tag{9}$$

where we added a term δ which accounts for the additional image disparity as opposed to the one which we would observe when $Z_w = 0$. If we simply substitute

[1] For the shake of performance we actually we do not compute a sheared image according to the ground plane disparity but use the amount of the shearing as a predisplacement for the search of correspondences.

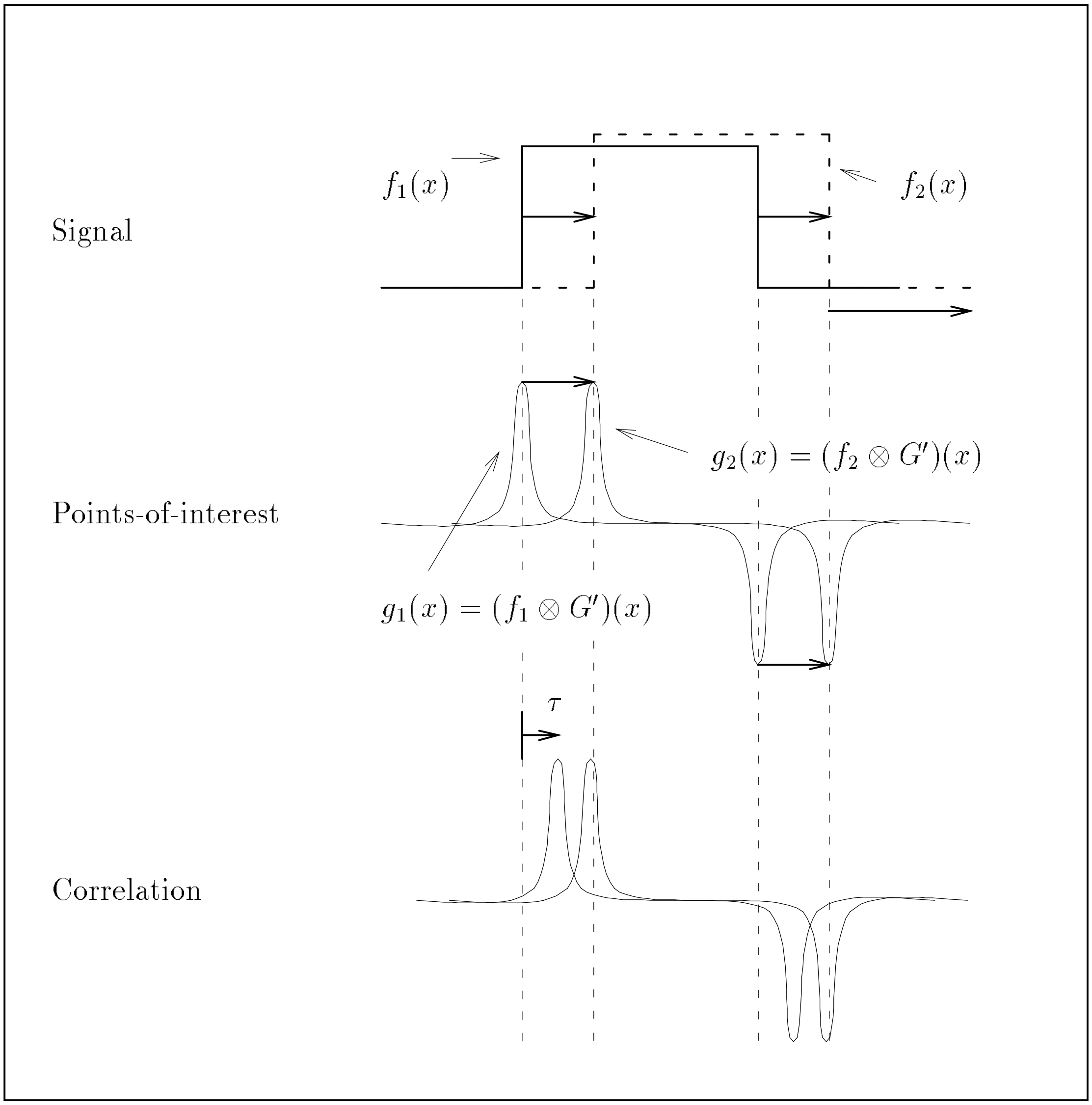

Signal
$f_1(x)$
$f_2(x)$
Points-of-interest
$g_2(x) = (f_2 \otimes G')(x)$
$g_1(x) = (f_1 \otimes G')(x)$
τ
Correlation

Figure 4: The image disparity is computed along horizontal scanlines. Only image locations (points-of-interest) with a first spatial derivative along the horizontal x-axis above a certain threshold are used for disparity computations. We apply a simple summed-squared-difference along a horizontal scanline to obtain an integer disparity estimate τ.

the expected image disparity for features on the ground plane from Equation 7 we finally obtain after some rearrangements:

$$\tilde{Z}_w = \frac{t_z}{1 + \frac{b}{\delta\, t_z}(f \cos\alpha - y' \sin\alpha)}, \tag{10}$$

Since we use $\Delta x'_0$ only for a predisplacement while searching for correspondences we give also the equation for Z_w expressed in the total measured image displacement

$\Delta x'$:

$$\tilde{Z}_w = t_z - \frac{b}{\Delta x'}(f\cos\alpha - y'\sin\alpha), \tag{11}$$

To express the Y-component of the 3D feature point in world coordinates, which we denote by $\tilde{Y}_w$, we use (cf. Figure 17):

$$\tilde{Y}_w = -(t_z - Z_w)\ \tan\gamma. \tag{12}$$

We obtain expressed in the residual disparity:

$$\tilde{Y}_w = -t_z\ \frac{f\sin\alpha + y'\cos\alpha}{\frac{\delta}{b} + f\cos\alpha - y'\sin\alpha)}, \tag{13}$$

and expressed in the total disparity:

$$\tilde{Y}_w = -\frac{b}{\Delta x'}\ (f\sin\alpha + y'\cos\alpha). \tag{14}$$

In order to get the X-component of the 3D feature point in world coordinates we use the perspective projection and solve for the $X_w = X_c$-component:

$$\tilde{X}_w = \tilde{X}_c = \frac{\tilde{Z}_c}{f}\ x'. \tag{15}$$

Using Equation 29 and 30 we obtain:

$$\tilde{X}_w = t_z\ \frac{x'}{f\cos\alpha - y'\sin\alpha}. \tag{16}$$

3.4 Obstacle Detection

Residual disparities — which appear in the image after the ground plane disparity has been mapped to zero (cf. Section 3.1) — indicate objects which appear above the ground plane. This process provides a simple tool to distinguish between features lying on the ground plane (e.g. lane markers or other paintings on the road) and features due to object lying above the ground plane and which may appear as obstacles during the course of the vehicle. For this purpose we set a simple threshold for the residual disparity according to Equation 7. The process for obstacle is sketched in Figure 5.

Image features at image locations with zero or small residual disparity which do not pass the threshold test are assumed to represent features lying on the ground plane. The procedure for how to use these disparities to update the camera parameters is described in the next section.

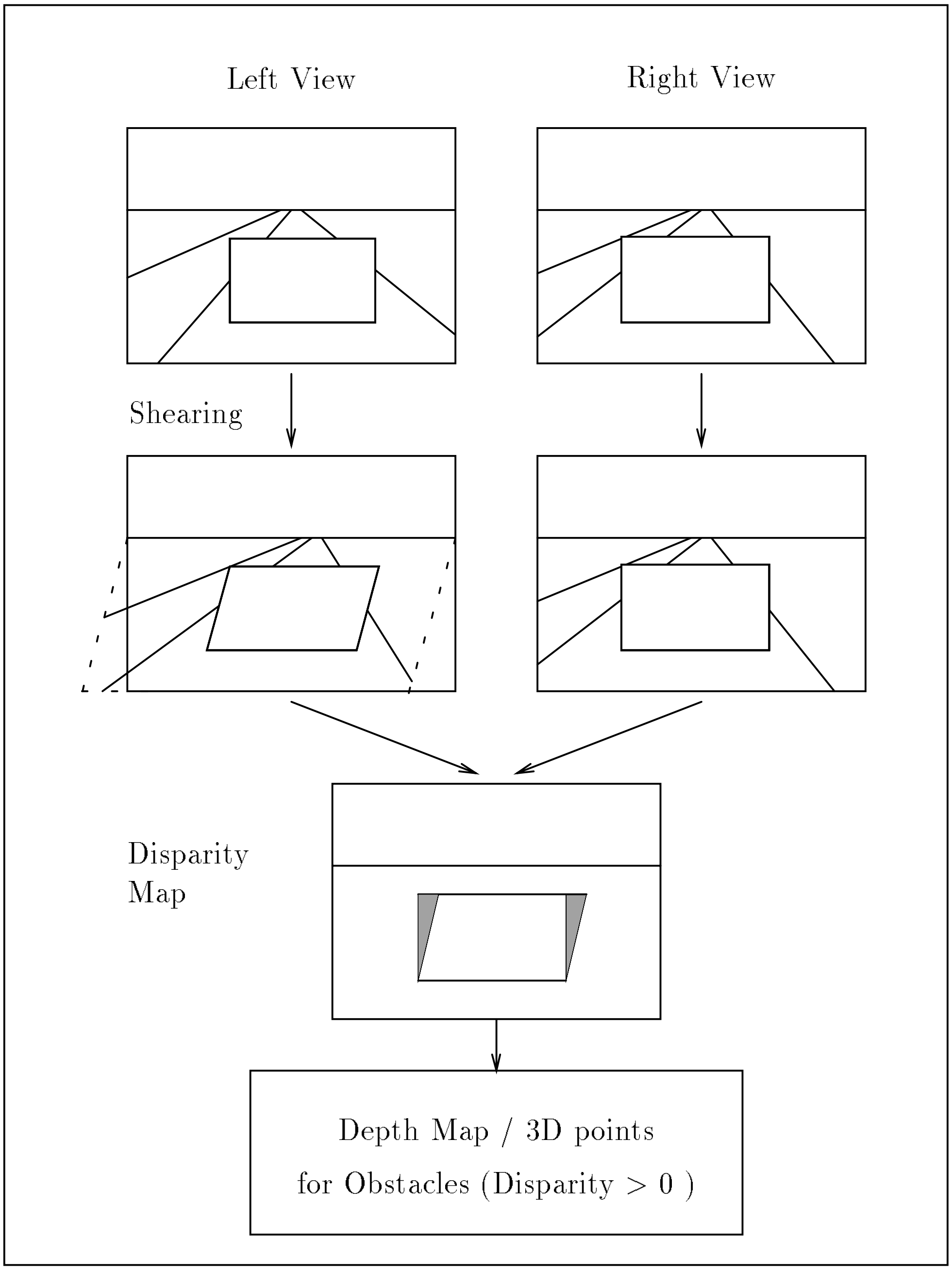


Figure 5: Obstacle detection by detecting residual disparities: to detect obstacles we compare a precomputed disparity according to features on the ground plane and the measured disparity. Any significant differences are due to features lying above the ground plane and are potential candidates for projected features of an obstacle which may preclude the course of the vehicle.

3.5 Dynamical Camera Parameter Update

The disparity of the ground plane is mapped to zero using known camera parameters or more precise the pose of the camera with respect to the ground plane. This camera pose can change with time due to camera shake, movements of the car's suspension and due to changes in the slope of the road. These changes can be incorporated by a camera parameter update using small measured disparities of image features which are assumed to lie on the ground plane. The candidates for these small disparities are the features which do not pass the threshold test used to separate obstacles from ground plane features mentioned in the previous section, and which lie in the search region of lane markers.

The parameter most effected by camera shake and change in vertical road curvature is the inclination angle α. Other parameters are supposed to be constant along the time. For that purpose we include an error $\epsilon(\alpha)$ in the disparity measurement of Equation 7 which accounts for an incorrect α:

$$\Delta x' = \frac{b\, f\ \cos\alpha}{t_z} - \frac{b\ \sin\alpha}{t_z}\, y' + \epsilon(\alpha). \tag{17}$$

We then write an error function $F(\alpha)$ which we want to minimize with respect to α:

$$\begin{aligned} F(\alpha) &= \sum_i \|\epsilon_i(\alpha)\|^2 \\ &= \sum_i \|\Delta x'_i + \frac{b}{t_z}(y'_i \sin\alpha - f\cos\alpha)\|^2. \end{aligned} \tag{18}$$

The summation is carried out over all feature points which yield small disparities and hence are assumed to be on the ground plane.

Differentiation with respect to α and setting to zero yields the following trigonometric equation:

$$A\ \cos\alpha + B\ \sin\alpha + C\ \sin 2\alpha + D\ \cos 2\alpha = 0, \tag{19}$$

with (N is the number of measurement points):

$$\begin{aligned} A &= \sum_i \Delta x'_i y'_i, \\ B &= \sum_i \Delta x'_i f, \\ C &= \frac{1}{2}\sum_i \frac{b}{t_z}(y'^2_i - f^2) = \frac{b}{2t_z}\sum_i y'^2_i - \frac{b\, f^2}{2t_z}\, N, \\ D &= -\sum_i \frac{f\, b}{t_z} y'_i. \end{aligned}$$

In order to solve this nonlinear equation in α we linearize it using a taylor expansion of the trigonometric functions around an initial value α_0, which we obtain from the static camera calibration (cf. Section 5.2). This is equivalent in solving this nonlinear equation using the first iteration of the Newton-Raphson method. We set:

$$\alpha = \alpha_0 + \delta\alpha \tag{20}$$

and approximate:

$$\begin{aligned}\sin(\alpha) &= \sin(\alpha_0 + \delta\alpha) \approx \sin\alpha_0 + \delta\alpha\cos\alpha_0, \\ \cos(\alpha) &= \cos(\alpha_0 + \delta\alpha) \approx \cos\alpha_0 - \delta\alpha\sin\alpha_0,\end{aligned}$$

and obtain the solution for $\delta\alpha$:

$$\delta\alpha = -\frac{A\cos\alpha_0 + B\sin\alpha_0 + C\sin(2\alpha_0) + D\cos(2\alpha_0)}{A\sin\alpha_0 - B\cos\alpha_0 - C\cos(2\alpha_0) + D\sin(2\alpha_0)}. \tag{21}$$

The inclination angle α is then consistently updated over time using a linear Kalman Filter based on the assumption of a constant α. Small variations are captured by an appropriate process noise. There are essentially two origins for variations in α: a short term variation due to camera vibrations, which requires a large process noise, and a long term variation caused by a change in the slope of the road, which can be captured using a small process noise.

4 Lateral Control

The lateral control variables for automatic guidance of a car in a lane or during a lane change can be extracted by computing the horizontal flow and the angular change of the lane markers. This information is checked for consistency with the lateral information from the magnetic sensors in order to yield the control variables for lateral motion. In the following subsection we want to describe our approach for using lane markers.

4.1 Using line flow

In order to deal with lateral control of the car, we have to obtain an estimate of its motion relative to the road. The traditional approaches based on optical flow are not satisfactory for several reasons:

- in high speed driving, the displacement between two consecutive frames is quite large, thus inducing systematic error due to the first-order approximation in differential methods, and increase the computing cost for any method,
- the unwanted vertical vibration induced by the car suspension are likely to cause large errors on the vertical component.

All these drawbacks could be overcome if, instead of relying on the general optical flow of the road, we consider only the flow of the lane markers, considered to be a global set. The idea is that the specific task of road following doesn't need to be based on as much information about the road as possible, but only on particular visual cues which can be extracted locally along lane markers ([Gordon 66; Dickmanns & Mysliwetz 92; Raviv & Herman 91]). The suggestion of [Raviv & Herman 91] is to use a particular property of the tangent point on the road edge and its optical flow. If the alignment of the moving vehicle is correct, then the radial component of this flow must be zero in a particular coordinate system.

What line flow information to use ? A more general formulation of the visual field obtained from a moving vehicle has already been presented by Gordon [Gordon 66]. When the moving vehicle is aligned with a straight or constant curvature portion of road, each point of one lane marker falls in the position previously occupied by another point of this lane marker. Thus, it is invariant as a set, and therefore the the road has a stationary appearance. If the vehicle is misaligned laterally, the entire lane marker moves, thus no part of it is more essential. This can be contrasted with [Raviv & Herman 91] who make an unnecessary restriction by looking only at tangent points. This restriction reduces the applicability of the idea to curved portions of the road, and also may cause the algorithm to lack robustness. For instance, it will fail if the particular point that is considered is occluded by another car.

As can be seen in Figure 6, the extent of lateral misalignment is indicated by the rate and extent of slewing and sidesliping of the lane markers. This information can be easily obtained by computing — in the coordinate system linked to the car —

- the difference in horizontal offset of the lane marker
- the difference in orientation of the lane marker

We introduce the notion of *line flow* as a more useful model than the standard term *optical flow* (used for denoting the pointwise motion) for this context. Consider the motion of an extended line segment in the plane with its two components of translation and rotation. The component of translation perpendicular to the line segment can be measured using image processing operations, as also its rotation. Because of the *aperture problem*, we expect not to be able to measure the component of the translation along the line. The term *normal flow* has been used before in the optical flow literature; we think the rotational component is also important. We are currently working on differential methods for estimating line flow, both the normal flow as well as the rotational flow.

Motion of a straight line along its direction results in zero line flow while the optical flow is, of course, non-zero. Because of the aperture problem it would be difficult to estimate the pointwise optical flow in this situation. However it will be quite easy to estimate the line flow. Another pleasant feature is that it is expected to be small when one is correctly following a road, so that it will be possible to use an efficient differential method. By modeling the line flow in the case of a lane change, it is expected that the same approach would also work for these planned movements.

line flow perception and control One of the big advantages of this framework is that it enables us to formulate the problem of road following as one of nulling deviation from the steady state by using some visual information which is easy to extract directly from the image. The departure of the previously described quantities from the null value can then be directly used to generate a control command, making a direct perception-action loop, in contrast to traditional approaches for which a conversion to 3D coordinates is needed. These commands can use fairly sophisticated

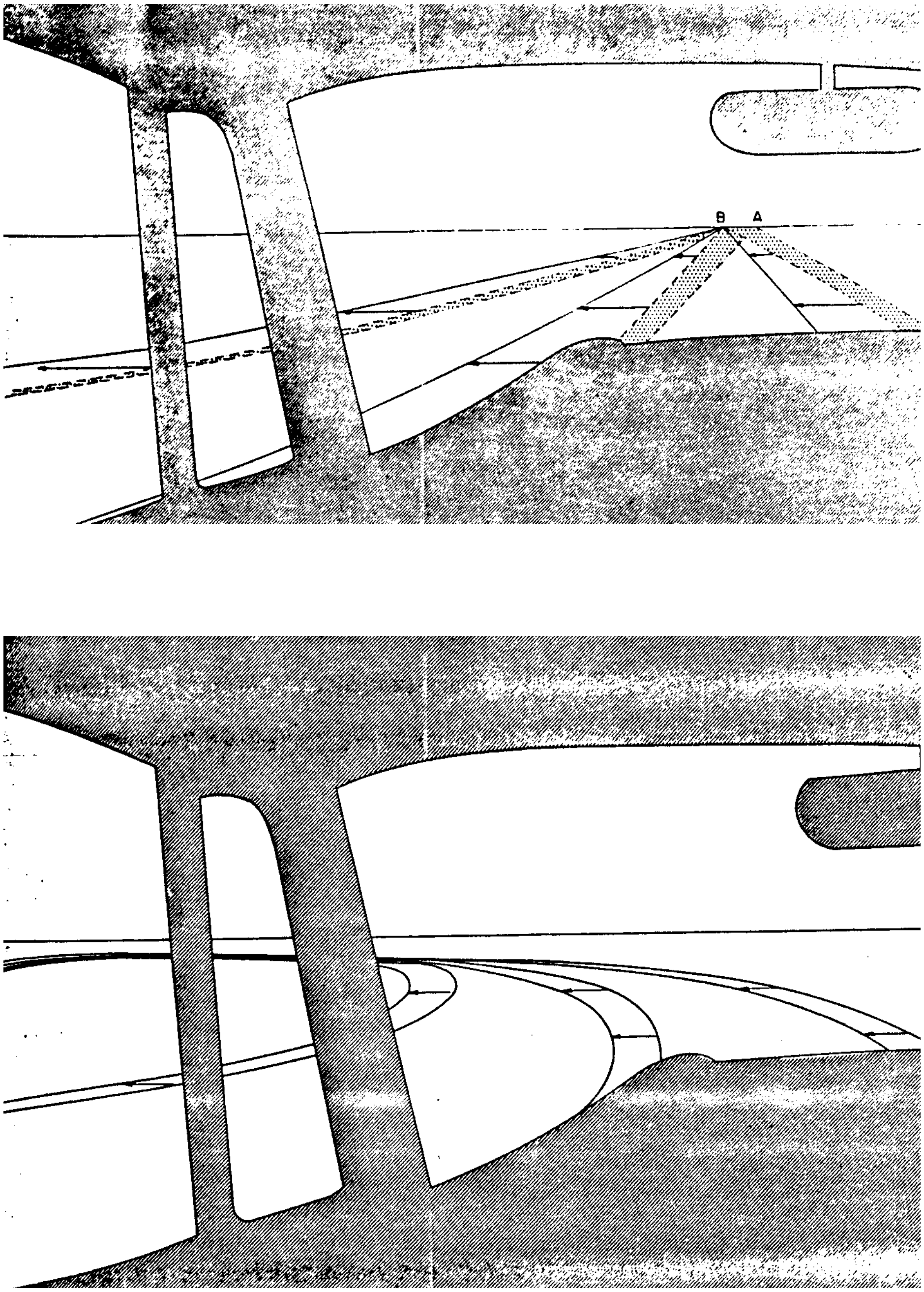


Figure 6: Lateral guidance on straight and curved road. Initial slewing is shown by shaded area and sideslip by arrows [Gordon 66].

geometrical information, and are not limited to a simple actions on the steering direction.

4.2 Lane Marker Detection

The goal of this module is to provide us with the description of the nearest lane markers seen from the car. Since these descriptions will be also used to control lane changes, it is important to be able to track multiple lane markers at various positions.

The Model

Planar surface hypothesis We have chosen to model the road as a planar patch between a closest depth Y_1 and a farthest depth Y_2. Thus there is a one-to-one transformation between the image coordinate and the 3D coordinates of a given point of this plane. This transformation is projective linear, which means there is a 3×3 invertible matrix $\mathbf{H}$, such that the following projective relation holds for each point:

$$\mathbf{m}_{image} = \mathbf{H}\mathbf{M}_{road} \tag{22}$$

We have developed an estimation method to compute this transformation in a simple and accurate way [Robert & Faugeras 93; Luong & Faugeras 93]. By writing that the two proportionality constraints obtained from (22) are satisfied, we obtain two equations which are linear in the coefficients of $\mathbf{H}$. Thus we can obtain a solution provided that we know the road coordinates of at least four points.

For the time being, we have supposed that this transformation is invariant over time, which means that we neglect the departures from planarity. However, the same method can be used to estimate *on-line* the value of the transformation matrix $\mathbf{H}$. In order to deal with non-planar roads –desirable to maintain a very high calibration accuracy[2] in the presence of vertical road curvature – we intend to couple this estimation with the dynamic calibration of inclination angle of the stereo rig which is performed in the stereopsis module.

Reference coordinate All the identification is done in a 3D reference coordinate system attached to the car, rather than a 3D coordinate system attached to the road. This allows us to obtain parameters which are directly relevant in terms of control actions. It is however easy to transpose data to a 3D coordinate system attached to the road, in order to obtain absolute positions of lane markers, and relative position of the car. This could be useful for example to maintain a global description of the road, or to integrate other sensorial inputs.

Model of lane markers The model that we use is based on the the actual road layouts widely used in civil engineering to produce high-speed roads. Each of the lane markers detected is modeled as a plane curve which is characterized by the four

[2] It has been reported [DeMenthon & Davis 90] that a small difference in the assumed and actual camera tilt angle with respect to the ground affects the 3D reconstruction significantly.

parameters, illustrated in Figure 7:

- O: lateral offset of the line in the car's coordinate system
- α_0: angle between the direction of the line at the closest position and the line of sight of the car
- C_1: curvature at the closest position
- C_2: curvature at the farthest position

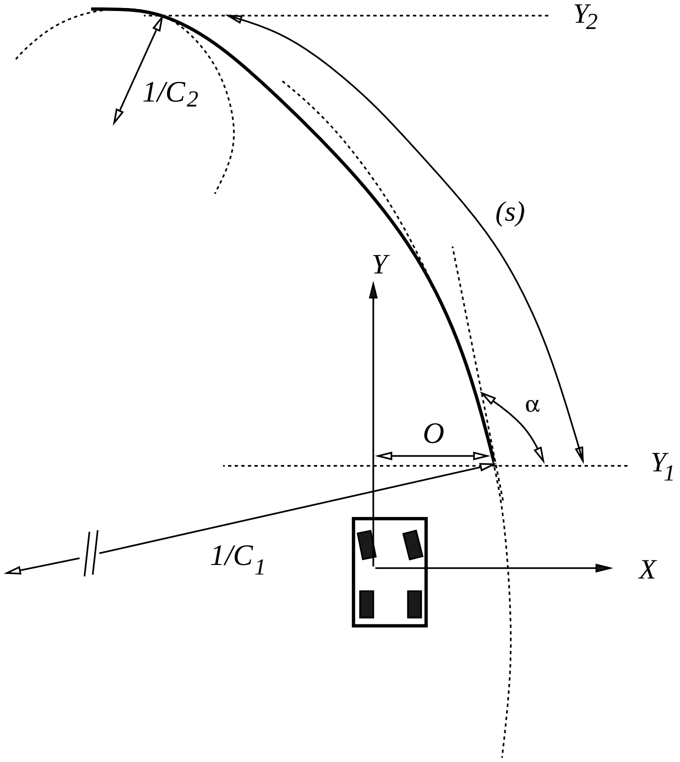


Figure 7: The model of lane markers. In the depth band, each lane marker is represented by an arc of clothoid with a constant factor, and by position and orientation in the car coordinate system.

The clothoid model, which is used in road design, consists of assuming that the curvature along the road is a continuous function of arc length, with a piecewise constant variation. As we look at a time at only portions of the road which do not exceed a distance of 100m, we can further suppose that on the sections that are being considered in a single image, the curvature has a constant variation a, called "clothoid parameter", that is:

$$C(s) = C_1 + as \tag{23}$$

where s is the arc length. The coordinates of the points on the curve are then obtained by:

$$X = O + \int_0^s \cos\alpha(t)dt \ , \ Y = Y_1 + \int_0^s \sin\alpha(t)dt \text{ with } \alpha = \alpha_0 + \int_0^s C(t)dt \qquad (24)$$

Some first order approximations are made to reduce the computational cost attached to the model [Dickmanns & Mysliwetz 92]. However, in spite of the small number of parameters, and of these simplifications, the model is more accurate than assuming just zero curvature [Turk *et al.* 88; Crisman 90; Kenue 89] or parabolic sections [Kluge & Thorpe 92]. It can represent accurately straight lines, arc of circles, and the transitions between them.

The Tracking Scheme

Unlike previous work, ours uses stereo analysis, which we believe will contribute significantly to improve the overall precision and robustness. It allows us to disregard areas that correspond to points that are above the ground plane, thus achieving more robust localization. All the stages of the algorithm make use of this information. We also plan to make use of geometrical constraints that arise from the cooperation of motion tracking and stereo matching [Faugeras *et al.* 90] [Navab *et al.* 90], which allows to recover 3D structure of lines from their 2D optical flow and stereo correspondences.

Initialization We search for the position of potential lane markers, as straight lines, using the fact that they are roughly parallel. The method works as follow:

- back project the Gaussian derivatives images — obtained by the output of the filter bank — onto the ground plane,
- select by a voting procedure the common orientation of lane markers,
- select by a clustering procedure the offsets of lane markers.

The algorithm is fast and can also be used to reinitialize the model in case of detected inconsistencies. An example of lines detected by the algorithm is shown in Figure 8.

Control structure While the vehicle moves, the following loop continually executes:

- *Predict* new parameters for each lane marker. This is done now by a locally constant velocity model for each of the four parameters. The estimate of the velocity obtained from odometry is used, as well as the estimates of the parameters in the two previous images.
- Define horizontal *search* bands in the image, based on the predicted parameters and their uncertainties, an example of which is shown in Figure 8.

- *Localize* within the search zone the center of lane markers, using a model of the image intensity produced by these lane markers (bright bars). An example of the points found is shown in Figure 8.

- Use these points, and the predicted parameters, to fit a new clothoid model by global optimization over the sample points of each lane marker. The solution relies now on a gradient method. An example of a portion of clothoid found is shown in Figure 9.

- *Update* the model parameters, and modify, if needed, the number of lines being tracked.

The algorithm as been tested in typical road scenes. Although we have not yet introduced any further hypothesis on the lane markers such as parallelism, the results are consistent, as can be seen in 9. The information that we are now able to estimate provides important information for lateral control:

- Change in *lateral orientation and position* can be used for instantaneous control in the way shown in previous section.

- The estimate of *road curvature*, allows to improve the control strategies in the sense that predictions for control variables are available that provide a smooth and safe ride.

All algorithm are based on computations that are fast and a further speed up can be obtained by parallel processing in a very straightforward way. The only time consuming phase is now the fitting procedure. We think that by using a differential approach, we will be able to avoid using global optimization, thus achieving real time performance.

5 Experimental Setup

We now want to describe our experimental setup we built in the this project. The goal was to acquire some image data from some typical driving conditions of a car on a highway and develop and test some key algorithms in order to show the feasibility of the concept.

5.1 The Test Car

For the purpose of acquiring stereo image sequences a stereo camera rig was built and mounted on top of a Lincoln towncar which was provided by PATH as a test vehicle (see Figure 10) The images of the synchronized[3] left and right camera were

[3] Since the car, and thus the cameras, is moving about 30 m/s at a speed of 65 mph, the longitudinal shift of a camera position between two subsequent frames is about 1 m at a framerate of 30 Hz. Unsynchronized cameras would thus give an additional average error of about 0.5 m in the depth estimation using stereopsis.

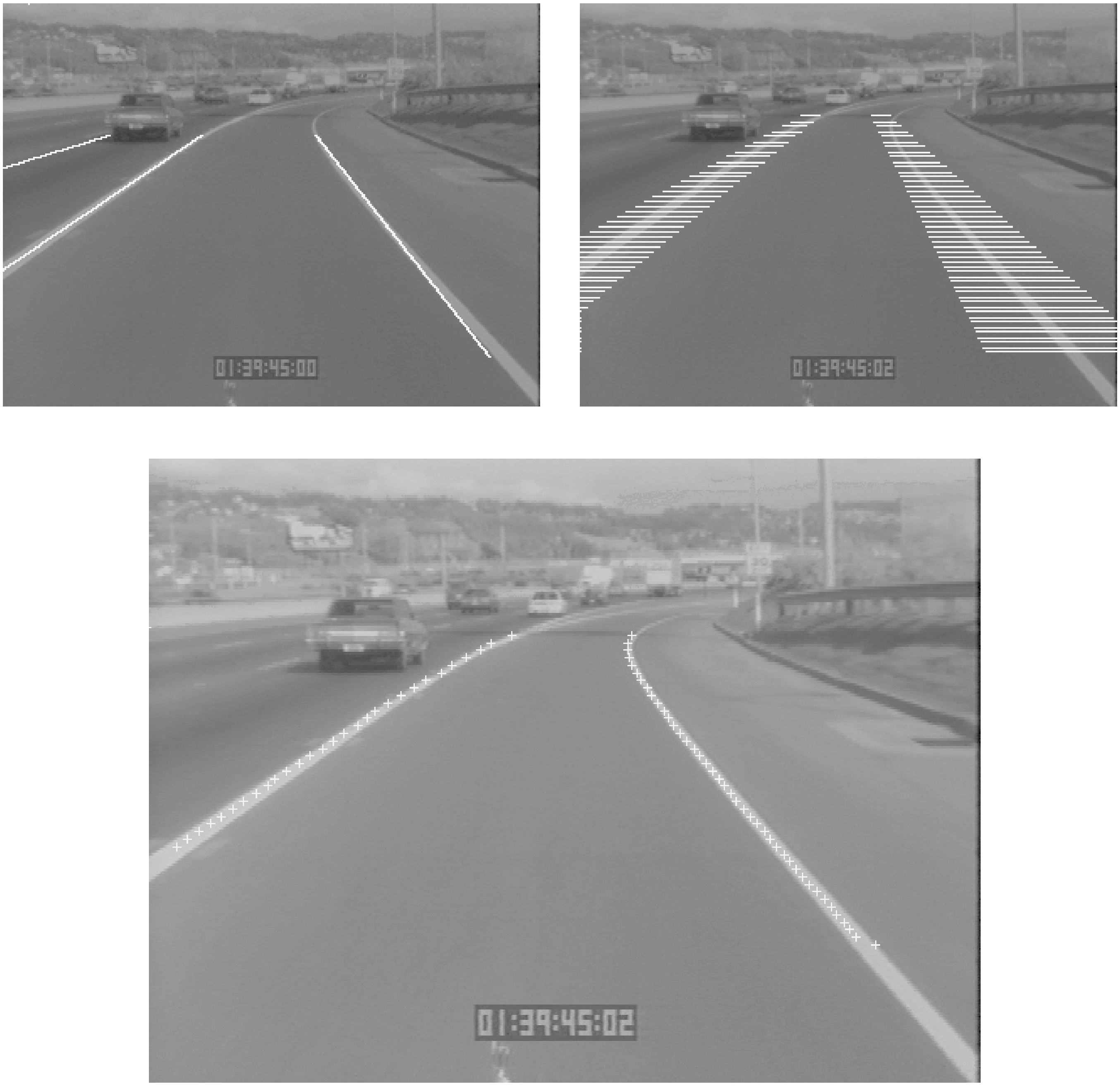


Figure 8: The initialization of the algorithm is done by detection of portions of straight lines of common orientation (top left). Within the search zone predicted using current the current lane markers parameters (top right), a precise model-based localization of lane markers points is performed (bottom).

parameter	left marker	right marker
lateral offset (m)	0.73	4.57
orientation (degree)	90.6	90.0
initial radius of curvature $\frac{1}{C_1}$ (m)	2074	3910
final radius of curvature $\frac{1}{C_2}$ (m)	309	297

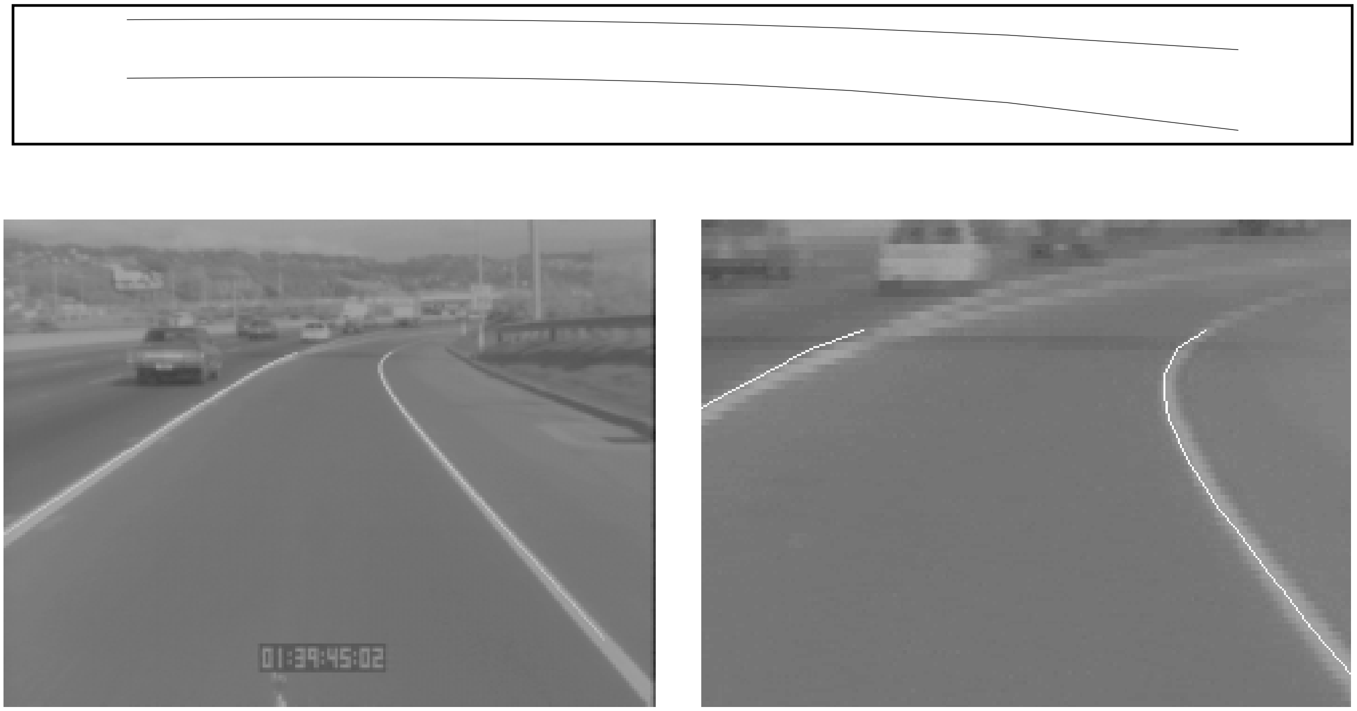


Figure 9: Estimated parameters, top view, and reprojected view of the fitted clothoids. The zoom shows that the fit is quite precise, in spite of the large distance and curvature variation.

recorded on two S-VHS tapes inside the car. The camera spacing (baseline) was about 20 cm. An additional time code generator produces a time code which can be overlaied in the images for an easy association of the left and right view. As a first coupling with other sensors we count the pulses from the ABS system from one of the tires in order to compute the velocity of the car. This velocity is stored together with the frame number in data file by a PC laptop and also overlaied in one of the images (see Figure 14a) and b)). We want to use this velocity information in order to increase the robustness and to check the accuracy of our algorithms.

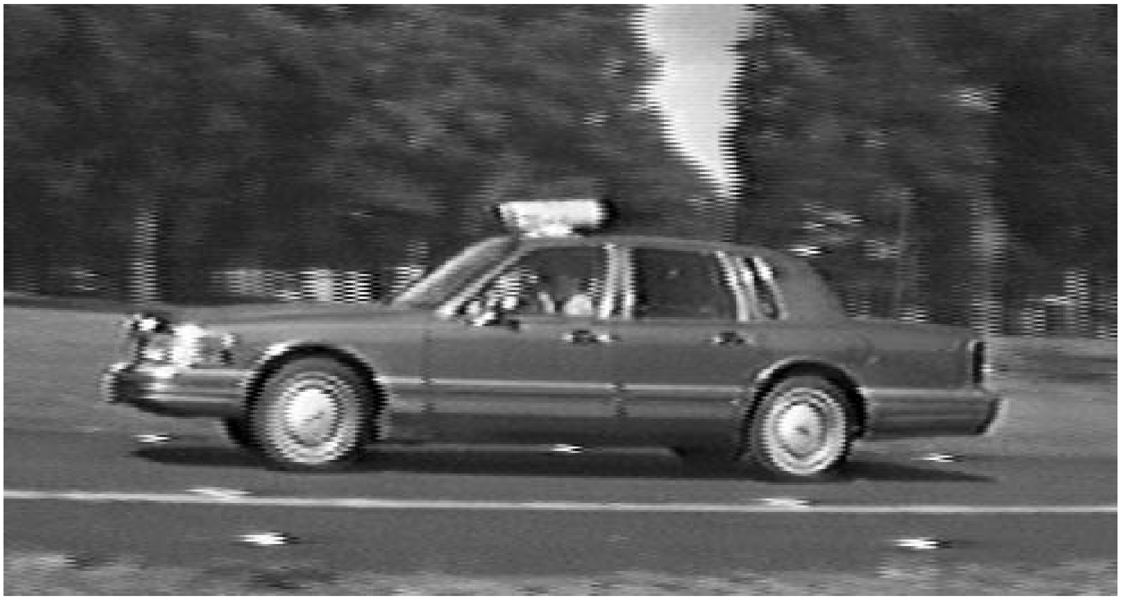


Figure 10: The Lincoln towncar with which we recorded test sequences on highways using the stereo camera rig mounted on top of the roof.

As a first coupling with other sensors we count the pulses from the ABS system from one of the tires in order to compute the velocity of the car. This velocity is stored together with the frame number in data file by a PC laptop and also overlaied in one of the images (see Figure 14a) and b)). We want to use this velocity information in order to increase the robustness and to check the accuracy of our algorithms. A hardware block diagram of the stereo vision setup is shown in Figure 11.

A Sync Generator produces a Sync signal which is amplified by the Video Distribution Amplifier and distributed to either cameras and the Time Code Generator (via the Laptop PC). The Linear Time Code (LTC) is inserted into the video image using the Window Inserter (WG-50) in the left branch and the Time Code Generator (TRG-50) itself in the right branch. Translators (VG-50) are then used to insert a Vertical Interval Time Code (VITS) in the video signal for retrieve with special VCR's. In left camera branch we insert also to digits of the velocity of the car using a Titler (SCT-50). The velocity is computed using the pulses of the Anti Lock Braking System (ABS) and stored together with the time Code in data file of a Laptop PC.

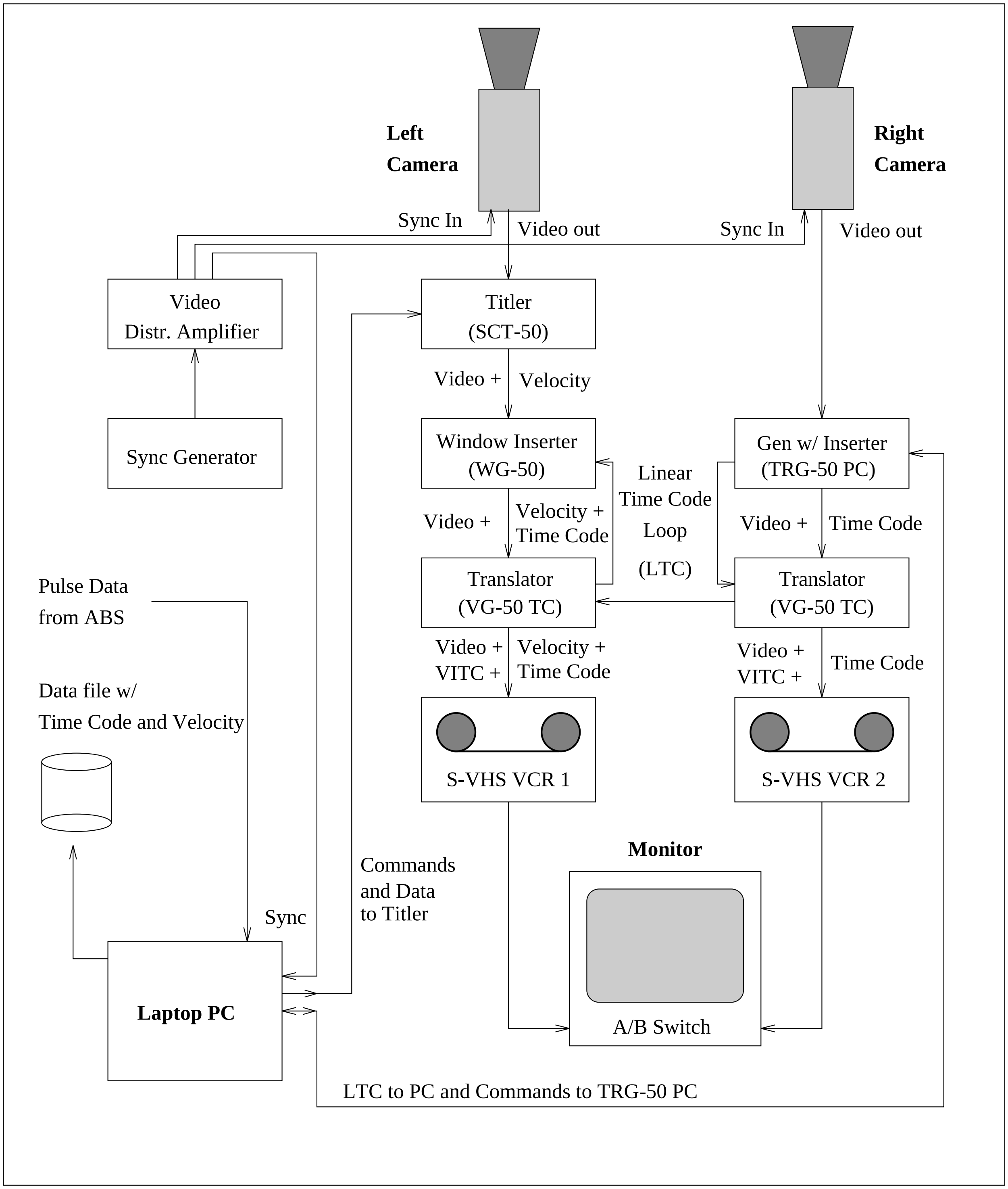


Figure 11: A hardware block diagram of the stereo vision system we used for recording synchronized stereo video sequences, based on suggestions from Randy Woolley, Caltrans.

5.2 Calibration of the Cameras

Since the algorithm we use in our approach for computing the stereo disparity is based on well aligned and calibrated cameras, we had to spend some effort to do this. Our algorithm assumes that the stereo camera baseline is parallel to the road plane and the optical axes of the cameras are parallel, i.e the common fixation point of the cameras is at infinity. This results in a very simple camera geometry in which the so-called epipolar lines (the intersection of the plane containing the two camera centers and a point in the scene with the image plane) are horizontal in both images. This reduces the search for correspondences to compute disparity to a single horizontal line (see subsection 3.1).

Camera calibration means the computation of internal (e.g. focal length) and external (the orientation and location of the camera with respect to some fixed world coordinate system) parameters, that enable the transformation from points on the road plane to points in the image and vice versa. The static calibration is based on correspondences between locations of 3D scene features and their associated 2D locations in the image. The larger the number of correspondences the larger the accuracy of the result, which is obtained by minimizing an error function (e.g. [Tsai 87; Lenz & Tsai 88]). For that purpose we put some markers on a flat road at exactly measured locations and recorded images for the left and right view (see Figure 12a)).

To complement our calibration we used also a calibration technique developed at INRIA [Faugeras & Toscani 86; Faugeras *et al.* 92]. This technique is based on images of a calibration grid and provides very accurate internal camera parameters (see b) of Figure 12).

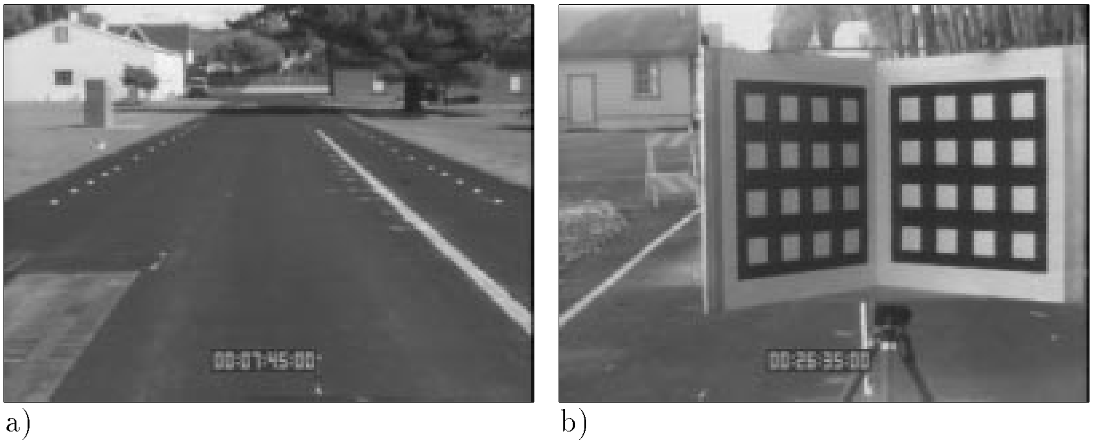


Figure 12: The left image shows a view of the calibration markers from one of the cameras. The right image shows a view of the calibration grid.

With this calibration technique we achieved a precision of about 0.6 pixels in the image which corresponds to an average error of about 0.05m at a distance of 15m (the nearest markers in Figure 12 a) and about 0.30m for a distance of 50m (the farthest markers in Figure 12 a).

5.3 Computational Environment

The computational environment should reflect the realtime aspect we require for the system. From the experience of other groups in building realtime applications we decided to build our final realtime system in three steps:

1. In the first step we use a simulation environment for developing and testing algorithms on standard UNIX workstations.

2. In the second step we plan to implement only the key algorithms on dedicated special hardware using DSP chips. This special hardware uses standard bus technology so we can control and test the algorithms running on the special hardware but using our simulation software implemented in the first step. This environment enables already a realtime test using VCR's and recorded tapes instead of cameras as sensor input.

3. In the final step we want to put the same special hardware in the car and link it with the other sensors and vehicle control system.

5.3.1 Software Simulation

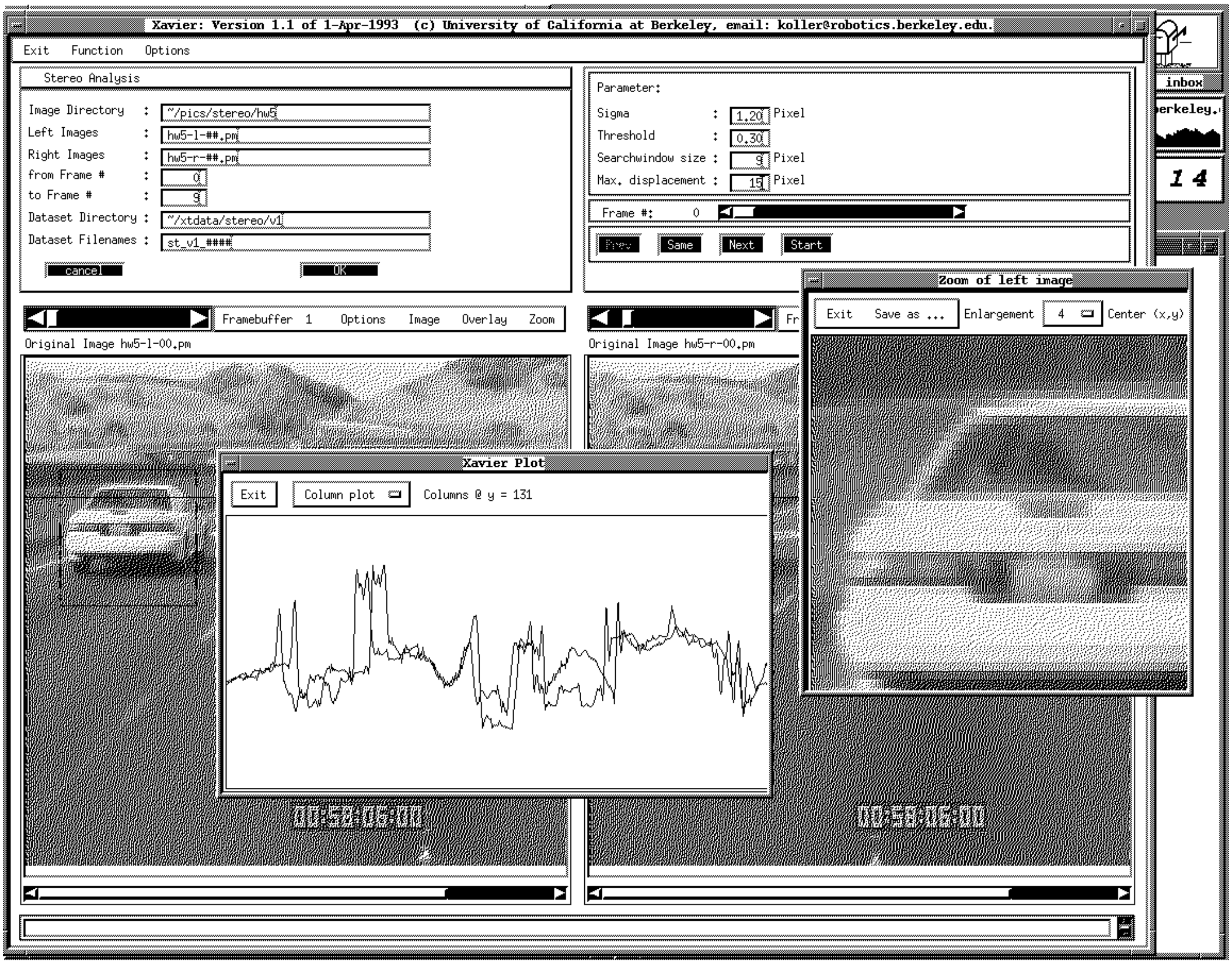
In this research project we developed a software simulation tool (XAVIER[4]) for testing the basic ideas and the key algorithms with image sequences we already recorded with the hardware setup. XAVIER is based on a graphical user interface and enables the selection and processing of single frames as well as an entire image sequence. XAVIER provides also all necessary tools for inspection and assessment of intermediate results in processing the images using graphical outputs (see Figure 13).

5.3.2 Real-Time issues

Concerning the problem of vision-based lateral control, real-time performance has already been achieved for several years, for example by the group of Dickmanns, using relatively simple hardware consisting of standard PC motherboards. However, they used an approach which was entirely integrated, in the sense that control variable and strategies were mixed with the motion parameters, and that the model for line markers was also built into the algorithm. Our approach is slightly more complex, since it has the advantage of providing a separate module for perception, thus allowing an easy integration of multiple sensors and alternative control laws, which is highly desirable from an engineering point of view. However, the tasks that are required are not very computationally expensive. For instance, Blake [Blake *et al.* 93] at Oxford has an implementation of a tracking scheme which may be suitable for lane tracking (it uses splines snakes) and works pretty close to realtime even on a Sun SparcStation.

Concerning the problem of longitudinal control and obstacle detection, although in the past real-time performance was a serious problem, the introduction of fast modern hardware architectures has already allowed a certain number of research

[4] XAVIER = X-based automatic vehicle image evaluation routines.


Xavier: Version 1.1 of 1-Apr-1993 (c) University of California at Berkeley, email: koller@robotics.berkeley.edu.

Stereo Analysis

```
Image Directory   : ~/pics/stereo/hw5
Left Images       : hw5-l-##.pm
Right Images      : hw5-r-##.pm
from Frame #      : 0
to Frame #        : 9
Dataset Directory : ~/xtdata/stereo/v1
Dataset Filenames : st_v1_####
```

```
Parameter:
Sigma             : 1.20 Pixel
Threshold         : 0.30
Searchwindow size : 9 Pixel
Max. displacement : 15 Pixel
Frame #:    0
```

Original Image hw5-l-00.pm

Original Image hw5-r-00.pm

Columns @ y = 131

Enlargement 4 Center (x,y)

Figure 13: A screendump of XAVIER during processing an image.

groups in the world to design near real time stereo algorithms. The best known of them are the real-time correlation algorithms developed at JPL by Matthies [Matthies & Grandjean 93], which currently produces a range image in less than 0.3s, and at INRIA in the group of Faugeras [*et al.* 93], which has similar performance with hardware comparable to the one we plan to use (8 Motorola DSP96002 chips), and is more than ten times faster when implemented in hardware (using DEC Programmable Gate Arrays). It should be noted that the aim of these works is to produce a dense depth map at a relatively high resolution, suitable for instance in the navigation of a rover for future planetary exploration over a rough terrain. In the case of the PATH project, we are mainly interested in detecting the closest obstacles. We do not need a very dense map, thus there is no need for us to perform the correlation over all image points. Also, there are two reasons for which we can work with a limited resolution: we do not need a very precise estimate of the distances, and since the road obstacles which are close enough to be a threat will have a fairly large image size, we are sure not to miss them even at very coarse scales.

In summary, real time performance has already been achieved for problems which we believe require more computational expense than the one we address here. In our research, we will take advantage of the simplifications specific to the PATH project to produce very fast algorithms.

Real-Time Hardware Early vision is dominated by parallel computations performed on small subsections of an image. These computations can be performed by independent, locally connected processors. A serial machine is very inefficient at performing these local computations. In these cases, tremendous computational speed-up can be obtained from using parallel processors. The use of parallel processing for vision-based vehicle guidance is therefore mandatory.

It is essential to carefully plan a strategy for porting programs to special purpose hardware and to check that appropriate programming and debugging tools are available.

We are using an array of interconnected TMS320C40 processors. The TMS320C40 is the latest in the line of Texas Instruments Digital Signal Processing (DSP) chips. The 'C40 runs at 50Mhz and has a peak performance of 50MFLOPS. More importantly, the 'C40 has 6 communications links which connect to other 'C40's providing 20Mb/s of data transfer per link. These communication links make it possible for the processors to communicate the large amounts of information in digitized images at video rate.

Our current system consists of an array of 6 processors residing on two motherboards. These motherboards are standard VME format boards. A link between the Sun S-BUS and the VME backplane provides communication between the host Sun and the processors. A frame grabber digitizes incoming images at video rate and broadcasts the image to the processors. An enclosure with a VME backplane provides the power and cooling for each of the motherboards and the frame grabber. By using a standard VME backplane we can later transfer the hardware to a vehicle. In addition, the host can be replaced by either a separate PC computer or a VME based host.

The software environment is a parallel version of the C language. This is standard C with a number of message passing routines added. In our current hardware configuration, software development is done on the host computer. The compiled code is downloaded to the processor array and executed. Messages can be passed between host and the processors during execution. Thus the graphics and storage facilities of the host computer can be used.

6 Results

Figure 14 shows a result of the disparity computation and a depth estimation using the left and right view of a highway scene. Figure 14a) and b) shows the original image, while Figure 14c) and d) exhibit the extracted points of interest in the already sheared left image and the right image. The result of the disparity computation is displayed in Figure 14e).

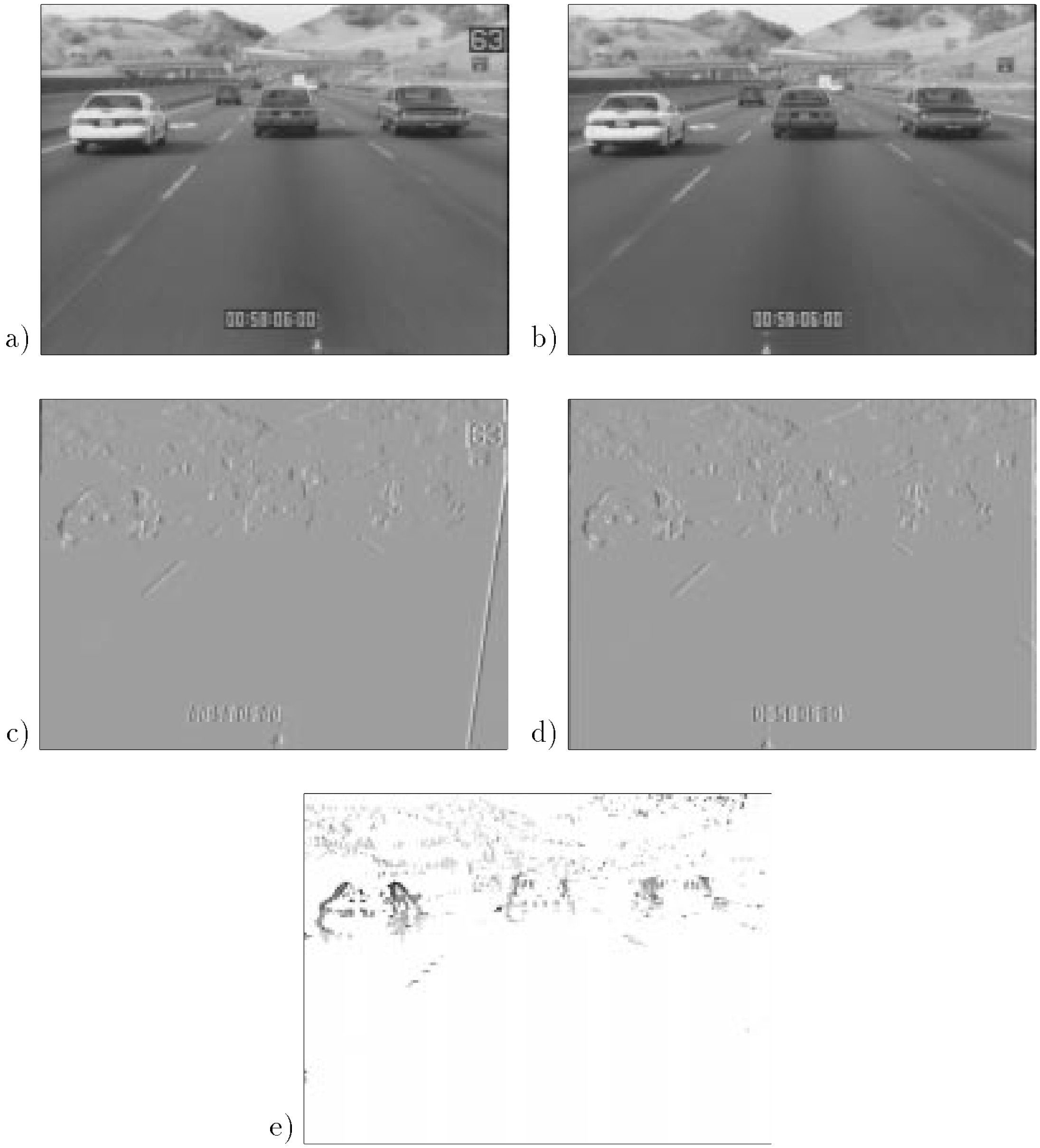


Figure 14: Various steps in the computation of the stereo disparity: a) and b) show the left and the right image. c) and d) show the points of interest in the sheared left image and the right image, respectively. The amount of the *Helmholtz-Shear* that brings the ground plane of left image into the view of the right image can be seen on the right slanted line in image c). The final disparity map is given as greycoded values in e).

Computing the points-of-interest for an 638×478 image takes about 3 Seconds on a Sun SparcStation 10 using a convolution with the first derivative in horizontal direction of a Gaussian function with $\sigma = 1.2$ and a 6×9 filter kernel. Computing the disparity using a correlation between horizontal scanline segments of length 9 pixels and a maximal disparity of 15 pixels takes about 4 Seconds. This includes sub-pixel interpolation by means of a quadratic interpolation. The disparity computation can be speeded up using only a subsampled number of scanlines. Including shearing of one of the images and applying a threshold to the filter outputs the total time for computing the depth from a stereo image pair of size 638×478 is about 12 seconds on a Sun SparcStation 10. This performance is state-of-the-art compared to results obtained by other research groups (e.g. [Matthies 92]). Note that obstacle detection can be performed on largely subsampled copies of the full-scale image, e.g. subsampled to 64×64 pixels, in which case the performance is increased by a factor 100 and is hence only 120 milliseconds. Further speed-up can be achieved using the proposed special purpose hardware.

The result of the depth estimation for the right view is given in Figure 15a), where the depth is displayed as greyscale values. For a better interpretation, a birds eye view is shown in Figure 15b). The three cluster of points correspond to the three cars moving in front of the camera as seen in Figure 14b).

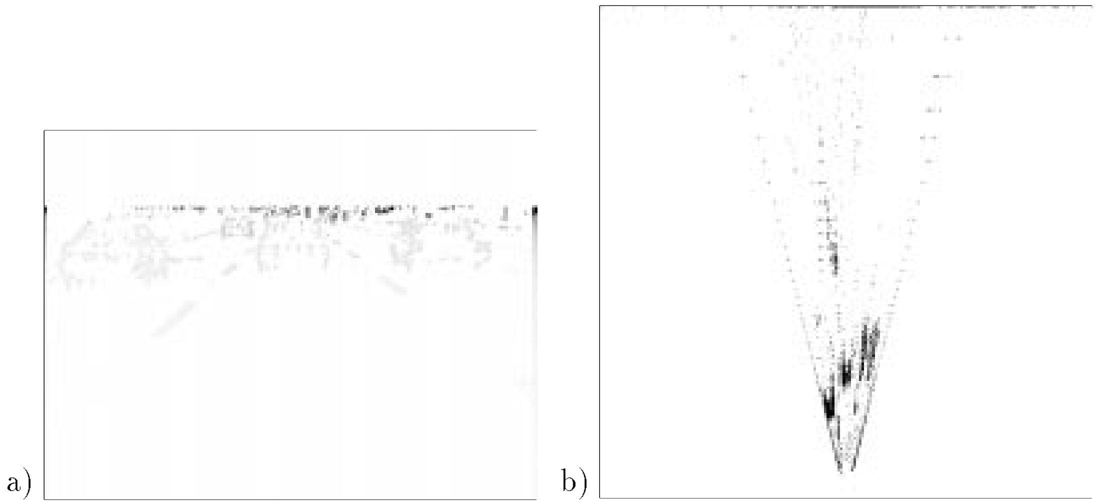


Figure 15: Shows the greycoded depth map of Figure 14 and b) shows a birdseye view of the obstacles and the lane markers.

In order to test the accuracy of the estimated depth from stereopsis we compared these values with the distances calculated by means of perspective inversion using calibration data obtained with a static calibration described in the next section. For that purpose we back projected an appropriate selected rectangle on the road plane beneath each car in order to obtain the minimum and maximum values for the distance to the camera. The minimum and maximum values from the stereopsis are extracted from the clouds of the depth estimations for each car in Figure 15b). The result is shown in the following table:

	estimated depth range in [m] using			
	persp. inversion		stereopsis	
left car	17.5 –	23.5	16.5 –	22.0
middle car	23.8 –	N/A	23.0 –	N/A
right car	21.5 –	31.1	23.4 –	33.8

The reason why the maximum depth is not available for the middle car is that we have only the back view of this car. Deviations of the assumption of an exact aligned camera geometry (see next section) or deviations of the planar road (the road plane in the highway image seems to be slightly more slanted than the calibration image) causes differences between the two estimations. We see also an increasing error with the distance according to the statement in Subsection 3.1. To increase the accuracy and robustness of the depth estimations we plan to exploit temporal integration using prediction and update by means of standard Kalman filter techniques as mentioned in Section 3.

7 Acknowledgment

We would like to acknowledge the very helpful collaboration with Randy Woolley from Caltrans and the group of Wei-Bin Zhang (PATH) for help in building the stereo rig and assembling the video hardware in the test car. We also like to thank Joseph Weber for assistance in acquiring and programming the signal processor hardware. This work was supported by California Department of Transportation through the PATH project grant no. MOU-94.

A Appendix

A.1 Ground Plane Disparity Computation

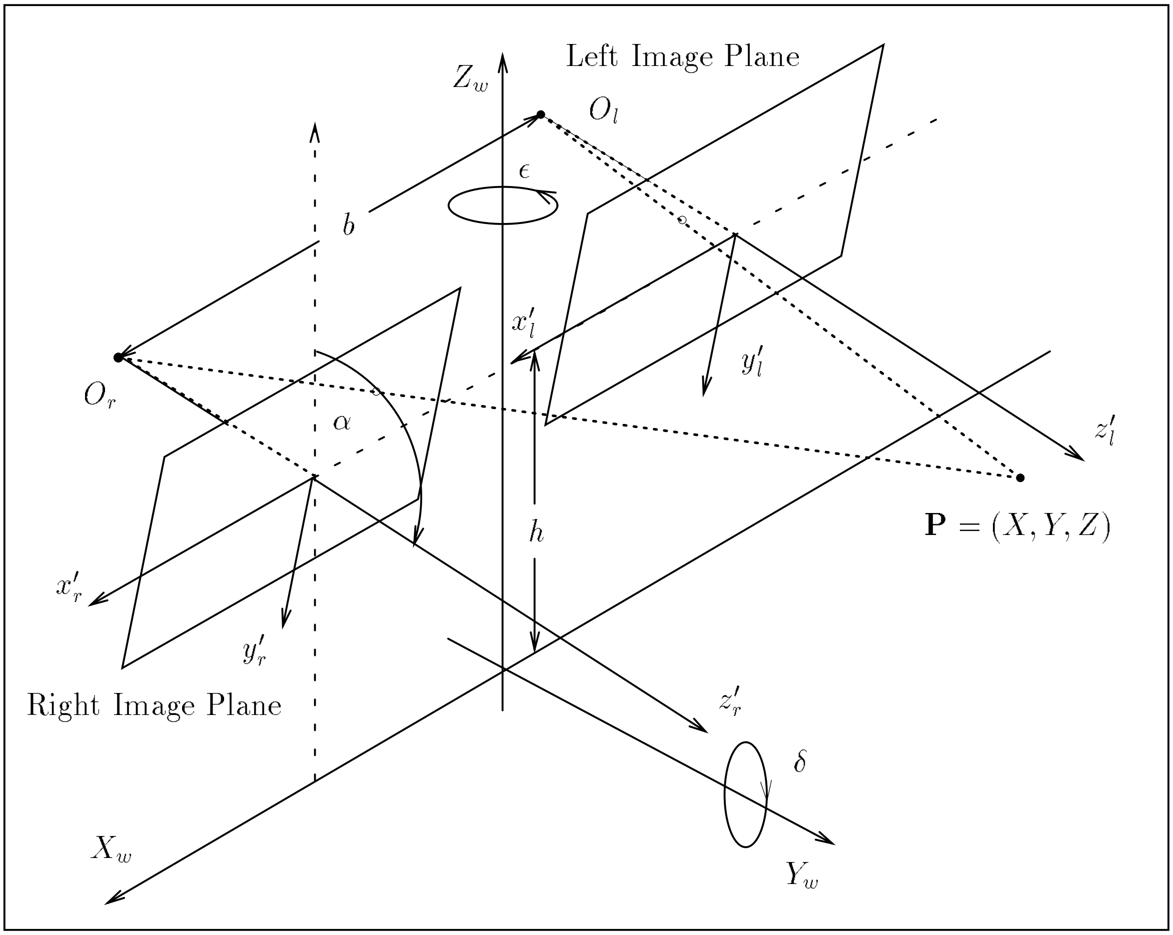


Figure 16: A more general view of a stereo camera set-up where the baseline b is parallel to the ground plane and the optical axes of the cameras are still parallel but with an inclination angle α towards the normal of the ground plane.

Figure 16 depicts a perspective view of the of the camera rig. The cameras have both the same inclination angle α. The cameras are further adjusted in order to make the angles δ and ϵ small and negligible. A transformation from world to camera coordinates is then performed by:

$$\begin{pmatrix} Y_c \\ Z_c \end{pmatrix} = R_x(-\alpha) \begin{pmatrix} Y_w \\ Z_w - t_z \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} Y_w \\ Z_w - t_z \end{pmatrix}, \qquad (25)$$

and hence for a point on the ground plane ($Z_w = 0$):

$$\begin{aligned} Y_c &= Y_w \cos\alpha - t_z \sin\alpha, & (26) \\ Z_c &= -Y_w \sin\alpha - t_z \cos\alpha, & (27) \end{aligned}$$

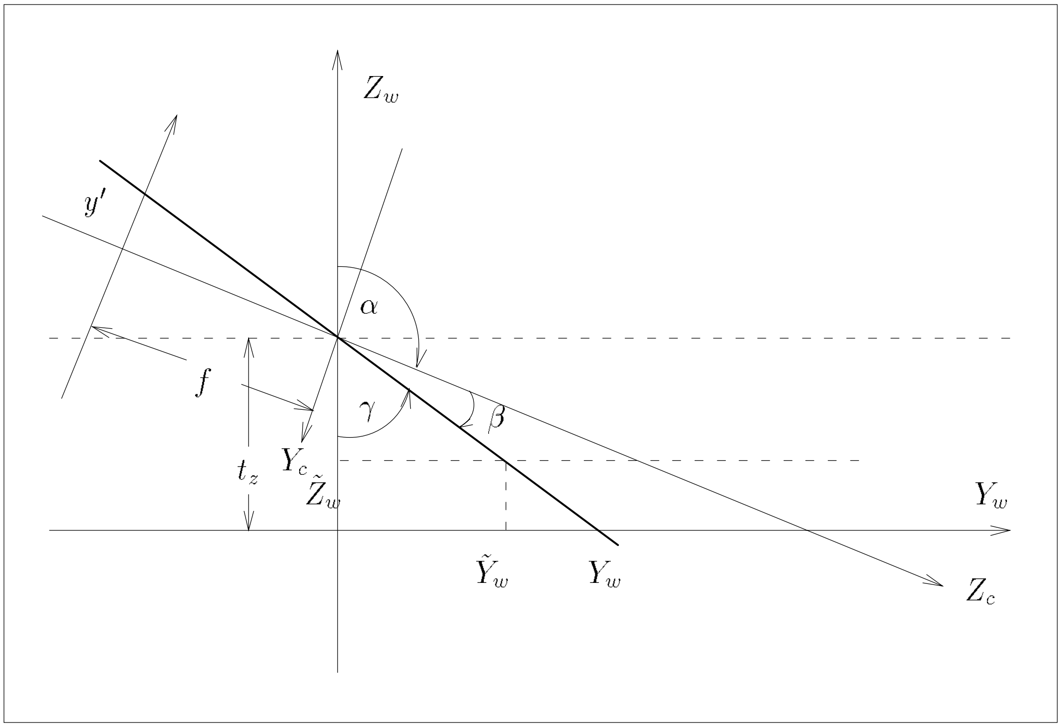


Figure 17: Both cameras are assumed to have an inclination angle α. A point $\mathbf{P} = (X_w, Y_w, Z_w = 0)^T$ on the ground plane is projected onto the image plane at same location (x', y') as a point $(\tilde{X}_w, \tilde{Y}_w, \tilde{Z}_w)^T$.

From Figure 17 we read:

$$Y_w = t_z \ \tan\gamma = -t_z \ \tan(\alpha+\beta) = t_z \frac{f\sin\alpha + y'\cos\alpha}{f\cos\alpha - y'\sin\alpha}, \tag{28}$$

where we made us of the fact that $\tan(\alpha+\beta) = \frac{\tan\alpha + \tan\beta}{1 - \tan\alpha \ \tan\beta}$ with

$$\tan\beta = \frac{y'}{f} = \frac{Y_c}{Z_c} = \frac{\tilde{Y}_c}{\tilde{Z}_c}. \tag{29}$$

Substituting Equation 28 into 26 we obtain:

$$Y_c = t_z \ \frac{y'}{f\cos\alpha - y'\sin\alpha}. \tag{30}$$

Finally by substituting this into Equation 5 we obtain:

$$\Delta x' = \frac{b \ \sin\alpha}{t_z} \ y' - \frac{b \ f \ \cos\alpha}{t_z}. \tag{31}$$

A.2 Quadratic Subpixel Interpolation

The objective is to find the maximum of a function of which we have only discrete samples around the maximum (cf. Figure 18. To find the maximum we perform a quadratic interpolation.

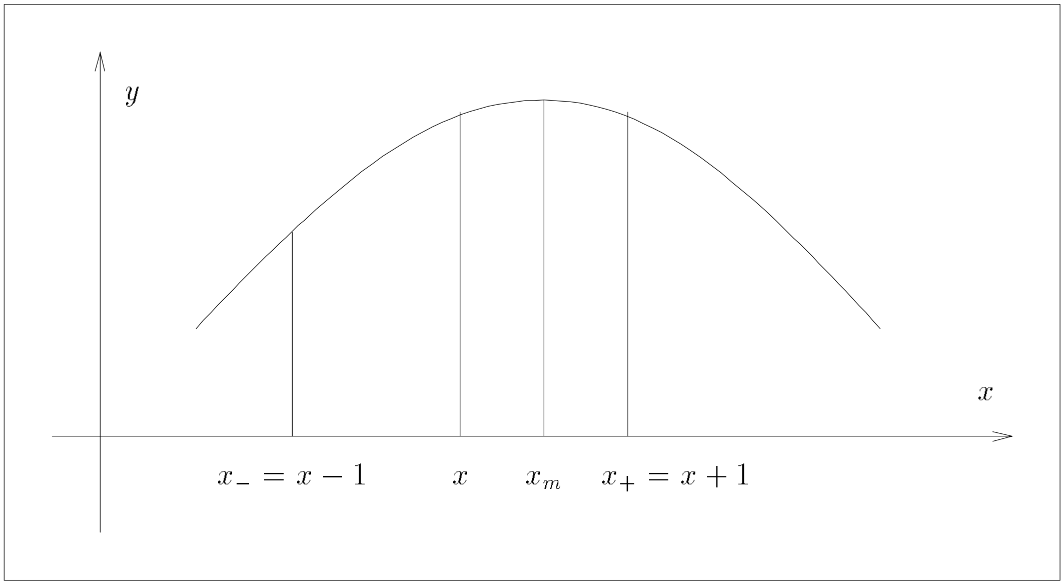


Figure 18: $x_- = x - 1$, x and $x_+ = x + 1$ are three sample of a function around the maximum x_m to be found by quadratic interpolation.

A quadratic function is parameterized by:

$$y = ax^2 + bx + c, \tag{32}$$

with the extremal point at

$$x_m = -\frac{b}{2a}. \tag{33}$$

The values of the function at the sample points are:

$$\begin{aligned} y_- &= ax_-^2 + bx_- + c = ax^2 + 2ax + a + bx + b + c, \\ y &= ax^2 + bx + c, \\ y_+ &= ax_+^2 + bx_+ + c = ax^2 - 2ax + a + bx - b + c, \end{aligned} \tag{34}$$

from which we obtain the parameters:

$$a = \frac{1}{2}(y_+ - 2y + y_-), \tag{35}$$

$$b = \frac{1}{2}(y_+ - y_-) - (y_+ - 2y + y_-)\, x \tag{36}$$

and hence the final position of the maximum:

$$x_m = x - \frac{1}{2} \cdot \frac{y_+ - y_-}{y_+ - 2y + y_-}. \tag{37}$$

References

[Adiv 85] G. Adiv, Determining 3-D motion and structure from optical flow generated by several moving objects, *IEEE Trans. Pattern Analysis and Machine Intelligence* **PAMI-7** (1985) 384–401.

[Altan *et al.* 92] O.D. Altan, H.K. Patnaik, R.P. Roesser, Computer Architecture and Implementation of Vision-based Real-Time Lane Sensing, in *Proc. of the Intelligent Vehicles '92 Symposium*, 1992, pp. 202–206.

[Barron *et al.* 90] J.L. Barron, A.D. Jepson, J.K. Tsotsos, The feasibility of motion and structure from noisy time-varying image velocity information, *International Journal of Computer Vision* **5** (1990) 239–269.

[Bar-Shalom & Fortmann 88] Y. Bar-Shalom, T.E. Fortmann, *Tracking and Data Association*, Academic Press, New York, NY, 1988.

[Blake *et al.* 93] A. Blake, R. Curwen, A. Zisserman, Affine-invariant contour tracking with automatic control of spatiotemporal scale, in *International Conf. on Computer Vision*, Berlin, Germany, May. 11-14, 1993, pp. 66–75.

[Carlsson & Eklundh 90] S. Carlsson, J.-O. Eklundh, Object detection using model based prediction and motion parallax, in *Proc. First European Conference on Computer Vision*, Antibes, France, Apr. 23-26, 1990, O. Faugeras (ed.), Lecture Notes in Computer Science **427**, Springer-Verlag, Berlin, Heidelberg, New York, 1990, pp. 297–306.

[Chandrashekar *et al.* 91] S. Chandrashekar, A. Meygret, M. Thonnat, Temporal Analysis of Stereo Image Sequences of Traffic Scenes, in *Proc. Vehicle Navigation and Information Systems Conference*, 1991, pp. 203–212.

[Crisman 90] J. Crisman, *Color Vision for the Detection of unstructured Roads and Intersection*, PhD thesis, Carnegie-Mellon-University, 1990.

[DeMenthon & Davis 90] D. DeMenthon, L.S. Davis, Reconstruction of a road by local image matches and global 3D optimiztion, in *Proc. International Conf. on Robotics and Automation*, 1990, pp. 1337–1342.

[Dickmanns & Mysliwetz 92] E.D. Dickmanns, B.D. Mysliwetz, Recursive 3-D road and relative ego-state recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14** (1992) 199–213.

[*et al.* 93] O.D. Faugeras *et al.*, Real time correlation-based stereo: algorithm, implementations and applications, 1993. INRIA.

[Enkelmann 90] W. Enkelmann, Obstacle detection by evaluation of optical flow fields from image sequences, in *Proc. First European Conference on Computer Vision*, Antibes, France, Apr. 23-26, 1990, O. Faugeras (ed.), Lecture Notes in Computer Science **427**, Springer-Verlag, Berlin, Heidelberg, New York, 1990, pp. 134–138.

[Faugeras & Toscani 86] O.D. Faugeras, G. Toscani, The calibration problem for stereo, in *Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, June 22–26, 1986, pp. 15–20.

[Faugeras *et al.* 92] O.D. Faugeras, Q.-T. Luong, S.J. Maybank, Camera self-calibration: theory and experiments, in *Proc. Second European Conference on Computer Vision*, S. Margherita, Ligure, Italy, May 18-23, 1992, G. Sandini (ed.), Lecture Notes in Computer Science **588**, Springer-Verlag, Berlin, Heidelberg, New York, 1992, pp. 321–334.

[Faugeras *et al.* 90] Olivier D. Faugeras, Nassir Navab, Rachi Deriche, On the Information contained in the Motion Field of Lines and the cooperation between Motion and Stereo, *International Journal on Imaging Systems and Technology* **2** (1990) 356–370.

[Gordon 66] D.A. Gordon, Perceptual Basis of Vehicular Guidance, *Public Roads* **34**:3 (1966) 53–68.

[Helmholtz 25] H.v. Helmholtz, *Treatise on Physiological Optics* (translated by J.P.C. Southall), Volume 1–3, Dover, NY, 1925.

[Huang 86] T.S. Huang, Determining Three-Dimensional Motion and Structure From Perspective Views, in *Handbook of Pattern Recognition and Image Processing*, 1986, pp. 333–354.

[Jochem *et al.* 93] T.M. Jochem, D.A. Pomerleau, C.E. Thorpe, MANIAC: A next generation neurally based autonomous road follower, in *Image Understanding Workshop*, Washington, DC, April 18-23, 1993, 1993, pp. 473–479.

[Jones & Malik 92] D.G. Jones, J. Malik, A computational framework for determining stereo correspondence from a set of linear spatial filters, in *Proc. Second European Conference on Computer Vision*, S. Margherita, Ligure, Italy, May 18-23, 1992, G. Sandini (ed.), Lecture Notes in Computer Science **588**, Springer-Verlag, Berlin, Heidelberg, New York, 1992, pp. 395–410.

[Kenue 89] S.K. Kenue, LANELOK: Detection of Lane Boundaries and Vehicle Tracking using Image-Processing Techniques: Part I+II, in *SPIE Mobile Robots IV*, 1989.

[Kluge & Thorpe 92] K. Kluge, C. Thorpe, Representation and recovery of road geometry in YARF, in *Proc. Intelligent vehicules symposium*, 1992, pp. 114–119.

[Koller *et al.* 93] D. Koller, K. Daniilidis, H.-H. Nagel, Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes, *International Journal of Computer Vision* **10**:3 (1993) 257–281.

[Lenz & Tsai 88] R.K. Lenz, R.Y. Tsai, Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machin e Vision Metrology, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10** (1988) 713–720.

[Longuet-Higgins & Prazdny 80] H.C. Longuet-Higgins, K. Prazdny, The interpretation of a moving retinal image, *Proc. Royal Society of London* **B208** (1980) 385–397.

[Luong & Faugeras 93] Q.-T. Luong, O.D. Faugeras, Determining the Fundamental matrix with planes: unstability and new algorithms, in *Conference on Computer Vision and Pattern Recognition*, New-York, 1993. To appear.

[Masaki 92a] I. Masaki (ed.), *Proc. of the Intelligent Vehicles '90 Symposium*, IEEE Industrial Electronics Soc., 1992.

[Masaki 92b] I. Masaki (ed.), *Vision-based Vehicle Guidance*, Springer Verlag, 1992.

[Matthies 92] L. Matthies, Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation, *International Journal of Computer Vision* **8** (1992) 71–91.

[Matthies & Grandjean 93] L. Matthies, P. Grandjean, Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors, in *Conference on Computer Vision and Pattern Recognition*, New York City, NY, June 15-17, 1993, pp. 657–658.

[Meygret & Thonnat 90a] A. Meygret, M. Thonnat, Object Detection in Road Scenes usng stereo data, in *Pro-Art Workshop on Vision*, Sophia Antipolis, April 19–20, 1990, 1990.

[Meygret & Thonnat 90b] A. Meygret, M. Thonnat, Segmentation of optical flow and 3D data for the interpretation of mobile objects, in *Proc. International Conference on Computer Vision*, Osaka, Japan, Dec. 4-7, 1990, pp. 238–245.

[Meygret *et al.* 92] A. Meygret, M. Thonnat, M. Berthod, A pyramidal stereovision algorithm based on contour chain points, in *Proc. Second European Conference on Computer Vision*, S. Margherita, Ligure, Italy, May 18-23, 1992, G. Sandini (ed.), Lecture Notes in Computer Science **588**, Springer-Verlag, Berlin, Heidelberg, New York, 1992, pp. 83–88.

[Micheli & Verri 92] Enrico De Micheli, Alessandro Verri, *Planar Passive Navigation: One Dimension is Better than Two*, Technical Report 92-058, International Computer Science Institute, 1992.

[Navab *et al.* 90] Nassir Navab, Rachid Deriche, Olivier D Faugeras, Recovering 3D motion and structure from stereo and 2D token tracking cooperation, in *Proc. the third International Conference on Computer Vsion*, IEEE, Osaka, Japon, December 1990.

[Ohzora *et al.* 90] M. Ohzora, T. Ozaki, S. Sasaki, M. Yoshida, Y. Hiratsuka, Video-Rate Image Processing System for an Autonomous Personal Vehicle System, in *IAPR Workshop on Machine Vision Application*, Tokyo, Japan, Nov. 28–30, 1990, 1990, pp. 389–392.

[Pomerleau 92] D.A. Pomerleau, Progress in Neural Network-based Vision for Autonomous Robot Driving, in *Proc. of the Intelligent Vehicles '92 Symposium*, 1992, pp. 391–396.

[Raviv & Herman 91] D. Raviv, M. Herman, A New Approach to Vision and Control for Road Following, in *Conference on Computer Vision and Pattern Recognition*, Lahaina, Maui, Hawaii, June 3-6, 1991, pp. 217–225.

[Robert & Faugeras 93] L. Robert, O.D. Faugeras, Relative 3D Positioning and 3D Convex Hull Computation from a Weakly Calirated Stereo Pair, in *Proc. International Conference on Computer Vision*, Berlin, Germany, May 1993. To appear.

[Ross 93] Bill Ross, A Practical Stereo Vision System, in *Conference on Computer Vision and Pattern Recognition*, New York City, NY, June 15-17, 1993, pp. 148–153.

[Scales 85] L.E. Scales, *Introduction to Non-Linear Optimization*, Macmillan, London, 1985.

[Schwartzinger *et al.* 92] M. Schwartzinger, T. Zielke, D. Noll, M. Brauckmann, W.v. Seelen, Vision-based Car-Following: Detection, Tracking, and Identification, in *Proc. of the Intelligent Vehicles '92 Symposium*, 1992, pp. 24–29.

[Thorpe 90] C. Thorpe (ed.), *Vision and Navigation: The Carnegie-Mellon Navlab*, Kluwer Academic Publishers, Norwell, Mass, 1990.

[Tsai 87] R. Tsai, A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, *IEEE Trans. Robotics and Automation* **3** (1987) 323–344.

[Turk *et al.* 88] M.A. Turk, D.G. Morgenthaler, K.D. Gremban, M. Marra, VITS — a Vision System for Autonomous Land Vehicle Navigation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10** (1988) 342–361.

[Ullmer 92] B. Ullmer, VITA - An Autonomous Road Vehicle (ARV) for Collision Avoidance in Traffic, in *Proc. of the Intelligent Vehicles '92 Symposium*, 1992, pp. 36–41.

[Waxman *et al.* 87] A.M. Waxman, B. Kamgar-Parsi, M. Subbarao, Closed-form solutions to image flow equations for 3D structure and motion, *International Journal of Computer Vision* **1** (1987) 239–258.

[Weber & Malik 93] J. Weber, J. Malik, Robust computation of optical flow in a multi-scale differential framework, in *Proc. International Conference on Computer Vision*, Berlin, Germany, May. 11-14, 1993, pp. 12–20.

[Weng *et al.* 89] J. Weng, T.S. Huang, N. Ahuja, Motion and structure from two perspective views: algorithms, error analysis, and error estimation, *IEEE Trans. Pattern Analysis and Machine Intelligence* **PAMI-11** (1989) 451–476.

[Zheng *et al.* 90] Y. Zheng, D.G. Jones, S.A. Billings, J.E. W. Mayhew, J.P. Frisby, SWITCHER: a stereo algorithm for ground plane obstacle detection, *Image and Vision Computing* **8** (1990) 57–62.