Available online at www.sciencedirect.com

SCIENCE DIRECT®

Image and Vision Computing 24 (2006) 153–165

image AND vision COMPUTING

www.elsevier.com/locate/imavis

Inverse perspective mapping and optic flow: A calibration method and a quantitative analysis

Sovira Tan *, Jason Dale, Andrew Anderson, Alan Johnston

Department of Psychology, University College London, Gower Street, London WC1E 6BT, UK

Received 15 July 2004; received in revised form 19 September 2005; accepted 24 September 2005

Abstract

In most imaging devices (including the human eye), the mechanisms that govern the projection of a 3D scene onto a 2D surface render the extraction of useful 3D information difficult. We investigate the effects of perspective on optic flow in a driver assistance application in which a camera is mounted on the wing mirror in order to observe a driver's so called 'blind spot'. A car travelling toward the camera appears to increase in speed and size on the projected image although its real speed and size are constant. We show that the inverse perspective mapping, previously used for obstacle detection, can also help in the problem of extracting real world speed from 2D optic flow data. We provide a quantitative analysis that shows precisely to what degree speed uniformity in the 3D world can be recovered by the mapping. To determine some mapping parameters, we devised a calibration method adapted to our specific situation that can be performed on-line and unsupervised. Its simplicity lends itself to fast software or hardware implementation.

Keywords: Inverse perspective mapping; Optic flow; Calibration methods

1. Introduction

Obtaining information about the external environment is a crucial task for biological organisms as well as artificial systems designed for autonomous navigation or driver assistance applications. While sonar and radar are sometimes used in real world applications, vision seems to have been evolved in most diurnal animals because it provides a rich and detailed source of information. Artificial systems can emulate animal vision by acquiring images through cameras that essentially perform the same task as eyes. Inside the camera or the eye, however, on the image plane where the 3D scene is projected, the effects of perspective will complicate most high-level information retrieval problems.

In the present work, we are concerned with the effects of perspective on optic flow. Optic flow algorithms strive to assign a correct velocity vector to every point in the image, thus providing crucial information about ego-motion and the speed of surrounding objects [1–3]. We investigate a driver assistance application in which a camera observes the driver's 'blind spot'. From the resulting video sequence, optic flow is computed and analysed to provide useful feedback to the driver about the presence and speed of other nearby road users.

We can consider the motion of the car as rigid and purely translational. There are two special cases. If the car is travelling on a line orthogonal to the axis of the camera the effects of perspective are minimal. The image of the car does not expand in size and its image speed is uniform if its real speed is uniform. If the car is travelling along the camera's axis, that is, straight at the camera, the effects of perspective are maximal. We will not consider this particular case, which is the domain of application of time-to-collision techniques [4,5].

In this paper, we are addressing the more general case of a car travelling towards the camera although not straight at it. The artefacts, due to perspective, that render the extraction of useful information difficult, are manifold. As the car approaches the camera, its image size increases, which adds a component of expansion to the translational motion. Its projected speed on the image will appear larger as it gets closer to the camera. As a further complication, large pixel displacements make speed measures less reliable in most gradient-based optic flow algorithms. We give a thorough analysis of the problems posed by perspective projection on the recovery of real 3D motion in Section 2.

The strategy we adopted to compensate for the effects of perspective is to remap the image inverting the projection

* Corresponding author. Tel.: +44 20 7679 5386; fax: +44 20 7436 4276.
E-mail address: sovira.tan@ucl.ac.uk (S. Tan).

0262-8856/$ - see front matter © 2005 Published by Elsevier B.V.
doi:10.1016/j.imavis.2005.09.023

equations so that parallel lines and equal distances in the real scene are remapped to parallel lines and equal distances in the reconstructed image. The technique, called Inverse Perspective Mapping (IPM), was used previously by Mallot et al. [6] and Bertozzi et al. [7,8] to obtain a top view of the road. Their main motivation was obstacle detection. In addition, Bertozzi et al. also found IPM useful for lane detection. A detailed derivation of the IPM equations is given in Section 3.

Those equations, however, have parameters that depend on the camera's position and its viewing angle. Mallot et al. assumed those to be known a priori. Bertozzi et al. used a calibration grid with known geometry painted on the ground [8]. We want to propose an approach to parameter calibration that does not assume a priori knowledge and that can be performed on-line without any specifically designed calibration pattern. For our application we need to be as flexible as possible. Our algorithm must be able to adapt if the viewing angle is changed. The main idea behind the algorithm is to detect the angle between the road markings and work out the parameters that will make them parallel. One of the main advantages of our method is its simplicity. It dispenses with time-consuming and computationally expensive optimisation techniques. The other important feature of the method is that, despite its simplicity, it can be proven to remap equal distances in the real scene onto equal distances in the reconstructed image plane, which is a requirement to guarantee that uniform speed in the real world is correctly recovered. This proof is provided in Section 4 alongside a detailed exposition of the calibration algorithm.

In Section 5, we provide a thorough quantitative analysis of the effect of IPM on optic flow velocity measures using a manual segmentation of the car. Mallot et al. only provided a qualitative assessment while Bertozzi et al. did not use optic flow at all for their application. We compare the optic flow measures obtained from the original sequence with those from the remapped sequence. The tests are performed on a real-life sequence of a car travelling at constant speed. We show that in the remapped sequence the acceleration of the image of the car due to perspective projection is dramatically reduced. Since image speed is now relatively constant over time, it provides a scaled measure of the real world closing speed of the overtaking car.

In Section 6, we introduce a simple automatic optic flow segmentation of the overtaking car that is made possible by the IPM. We compare this automatic segmentation to the manual one. We also process an additional sequence. In both sequences we show that speed constancy has been recovered to an acceptable degree. Section 7 discusses the possible difficulties that the algorithm might encounter. We show how the detection of lines fares in bad weather conditions. We also examine how robust the IPM is to calibration errors.

2. The perspective problem

The projection process shown in Fig. 1 can be described by the following equations [9]:

$$x = c\frac{X}{Z} \qquad y = c\frac{Y}{Z} \tag{1}$$

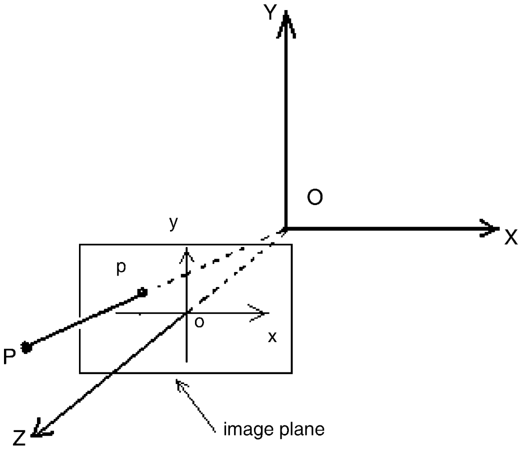

Fig. 1. Projection of a point P onto point p in the image plane.

where X, Y and Z are the coordinates of point P in the scene and x and y the coordinates of p, the projection of P on the image plane. The parameter c is the distance of the image plane to the centre of projection O. The Z-axis is the camera's optical axis.

We make the simplifying assumption that the car moves in a straight line, that is, in one dimension. From Fig. 2(a) it can be seen that if the car travels in a plane perpendicular to the optical axis, equal distances will project on equal distances and speed uniformity is preserved. This configuration is called fronto-parallel. However, in the case of our application the car is travelling towards the camera and thus its speed vector has an important component on the Z-axis. In that case, Fig. 2(b) shows that equal distances will not project onto equal distances but onto increasing distances, which explains the apparent acceleration of the car as it approaches the camera.

In the first case (Fig. 2(a)) the distance Z is constant. Taking the time derivative of Eq. (1) yields:

$$u = \frac{c}{Z}U \qquad v = \frac{c}{Z}V \tag{2}$$

where U and V are, respectively, the X-axis and Y-axis components of the speed of point P in the real scene and u and v their projected counterparts on the image. It can be seen that there is only a multiplicative constant between the real speed of the car and its speed as seen on the projected image. In our application the camera is always looking for cars in the adjacent lane. Since the centre of the adjacent lane will approximately be at a fixed lateral distance from the camera we can calibrate the algorithm with a test car travelling at a known speed and roughly in the middle of the lane. The real speed of cars travelling in that lane can then be recovered.

However, in the second case (Fig. 2(b)), because Z is time-varying, differentiating Eq. (1) yields the Longuet-Higgins equations in absence of rotation [10]:

$$u = \frac{cU - xW}{Z} \qquad v = \frac{cV - yW}{Z} \tag{3}$$

where W is the Z-axis component of the velocity of point P in the real scene. We limit ourselves to the case where the car is travelling at uniform speed so that U, V and W are constants. To recover the real speed of the car we need to determine U, V and

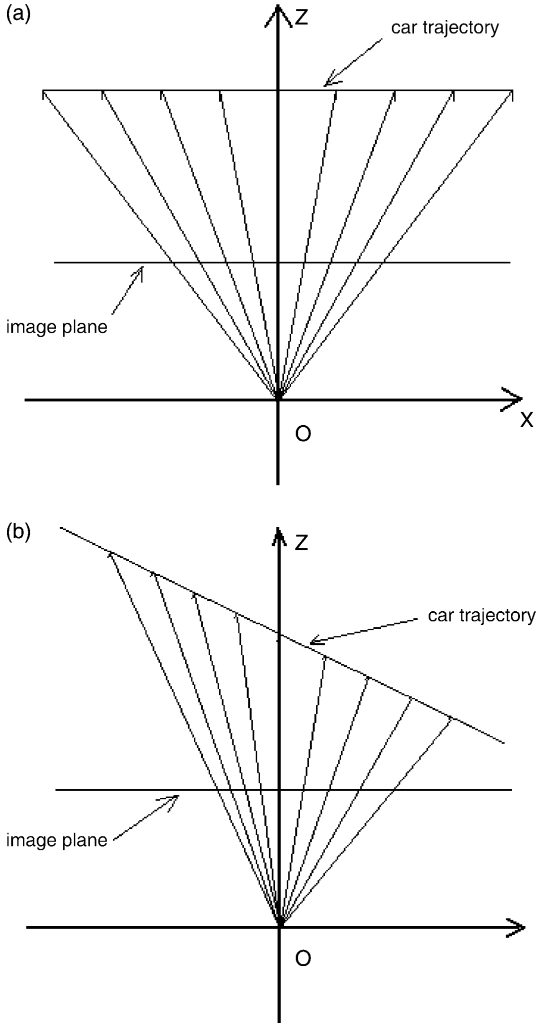

Fig. 2. Projection of the car's trajectory (a) in the fronto-parallel case (b) in the general case.

W from Eq. (3). The distance *Z* is also an unknown and needs to be modelled as a function of *x* and *y*. In the best case, if the camera axis is perpendicular to the visible side of the car, *Z* will not vary with *y*. That nevertheless leaves at least two additional parameters to be introduced to model *Z* as function of *x*. The problem is therefore much more complex than in the first case and would require a numerical algorithm and possibly some optimisation procedures [11,12]. The solution to the problem will depend on the ability to take different points on the car at different depths *Z* together with the speed measures at those points. That means the solution will be very sensitive to the accuracy of optic flow measures at each individual points and it is well known that optic flow results are typically very noisy [13]. By contrast, in the fronto-parallel case, each pixel on the car should have the same velocity vector so that we should be able to extract a robust measure of the car speed by taking the mean.

In Section 3, we show that it is possible the remap an image from the second case to the first (fronto-parallel) case. Remapping, with the computational benefits it brings to speed determination, is all the more attractive in that the complete algorithm including the calibration process is simple, as the following two sections will show.

3. The inverse perspective mapping

Mallot et al. and Bertozzi et al. used the IPM to reconstruct the ground plane as if viewed from a camera placed above with its axis exactly to the vertical. In our application, the plane we want to recover is ideally parallel to the side of the overtaking car (in so doing we assume that the side of the car is roughly planar). Another way to characterise this plane is to describe it as the vertical section of the car containing the line that represents its trajectory. For simplicity we will from now on refer to this plane we want to recover as the 'median plane'. The problem we are facing is therefore that of a mapping between two planes. It is a single view camera problem: given the image plane, we want to reconstitute the 'median plane'.

An important assumption we make is that the camera is mounted vertically with its optical axis parallel to the ground. This assumption has some important consequences:

(1) The depth *Z* of any point on the 'median plane' does not depend on its *Y* (vertical) coordinate but only on *X* (consider Fig. 2(b) with the *Y*-axis orthogonal to the plane of the figure).
(2) The image plane being vertical, if the side of the car were viewed in a fronto-parallel configuration, then the lane markings and the edge of the road should appear to be horizontal. This property will be used to calibrate the IPM algorithm.
(3) This particular camera configuration results in important simplifications in the mapping equations compared to the more general case. More details will be provided in the discussion at the end of this section.

Fig. 2(b) can be considered as a bird's eye view of the projection process in our application. The line representing the trajectory of the car also represents the trace of the 'median plane' on the (OXZ) plane.

Eq. (1) relate the coordinates on the image plane (oxy) to the world coordinate system (OXYZ) as shown in Fig. 1. This world coordinate system is made so that the (OXY) plane and the projection plane (oxy) are parallel. However, because it is the 'median plane' we want to reconstruct, we need a world coordinate frame with a plane parallel to that plane. This is done by rotating the coordinate system (OXYZ) around the (OY) axis by the angle θ_0, which is the angle between the 'median plane' and the image plane and depends on the viewing angle of the camera. In this way, we obtain the new world coordinate system (OX$'$Y$'$Z$'$) shown in Fig. 3, where the plane (OX$'$Y$'$) is parallel to the 'median plane'. To clarify the configuration we have placed the top view of a car and a camera in Fig. 3.

Using the usual rotation equations [9] and the fact that on the whole 'median plane' the coordinate Z$'$ is equal to a

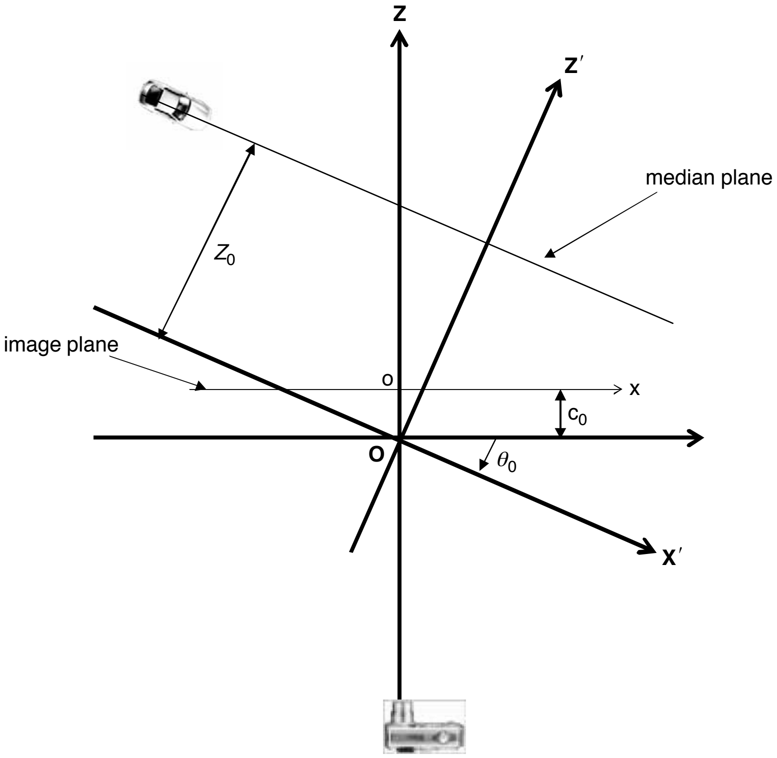

Fig. 3. Relation between world coordinate system and image coordinate system. The (OX′) axis of the world coordinate system is made to be parallel to the car's trajectory.

constant Z_0, we obtain a new form for Eq. (1):

$$x = c_0 \frac{X' - Z_0 \tan\theta_0}{X' \tan\theta_0 + Z_0} \qquad y = \frac{c_0}{\cos\theta_0} \frac{Y'}{X' \tan\theta_0 + Z_0} \tag{4}$$

Those equations relate the coordinates of points in the image plane to their counterparts in the 'median plane' using a world coordinate system linked to that plane.

Note that those equations are a simpler version of the usual and more general equations found in the literature (for example in Ref. [14]) that are of the form:

$$x = \frac{h_{11}X' + h_{12}Y' + h_{13}}{h_{31}X' + h_{32}Y' + h_{33}} \qquad y = \frac{h_{21}X' + h_{22}Y' + h_{23}}{h_{31}X' + h_{32}Y' + h_{33}} \tag{5}$$

In our method, we have expressed the h_{ij} parameters as functions of θ_0, Z_0 and c_0, that is, as functions of the geometry of our problem. In our equations h_{12}, h_{21}, h_{23} and h_{32} are null. This simplification is due to the way we chose the axis on the two planes (oxy) and (OX′Y′). The axes (oy) and (OY) that represent the vertical to the ground have the same orientation. In a more general case there could also be a rotation around the (OZ) axis between the two planes, in which case the parameters h_{12} and h_{21} would no longer be null. In Eq. (1), taken from Ref. [9], this choice of axis orientation is already adopted implicitly. It can be seen in Fig. 1 that between (OXY) and (oxy) there is only a translation, no rotation. The fact that h_{23} and h_{32} are null comes from the fact that there is no rotation around the (OX) axis between the 'median plane' and the image plane. In fact we have allowed only one rotation (described by the angle θ_0) around the (OY) axis between the two planes. Our choice of axis orientation might seem restrictive but it only involves placing the camera in a sensible manner, that is, vertically with (ox) parallel to the ground and (oy) orthogonal to it. We shall see in Section 5 that our algorithm is tolerant to small violations to our assumptions.

Eq. (4) give us an easy algorithm with which to reconstruct the 'median plane'. We scan the reconstructed image and we fill each pixel (X′,Y′) with the pixel value at the corresponding image point (x, y) given by Eq. (4). The problem is that the three parameters θ_0, Z_0 and c_0 are unknown. Section 4 will show how it is possible to calibrate them on-line with information contained in the original image.

4. The calibration method

The camera calibration problem has been extensively investigated [14,15]. The single view case has however attracted less attention than the 2, 3 or N-view case. Hartley et al. [14] propose several methods to undo perspective from a single view image. Their methods however are aimed at the general case and are fairly sophisticated. Our aim here is to describe a more specific method adapted to our particular application. We also aim to provide a method simple enough to be translated straightforwardly into an algorithm and that can run fast on a software or hardware implementation. Because it is based on the simplifying assumptions stated in Section 3, our method is much simpler than the ones described in Hartley et al. [14]. Another comparative advantage of our method is that it is unsupervised because there is no need to select a set of points or lines manually.

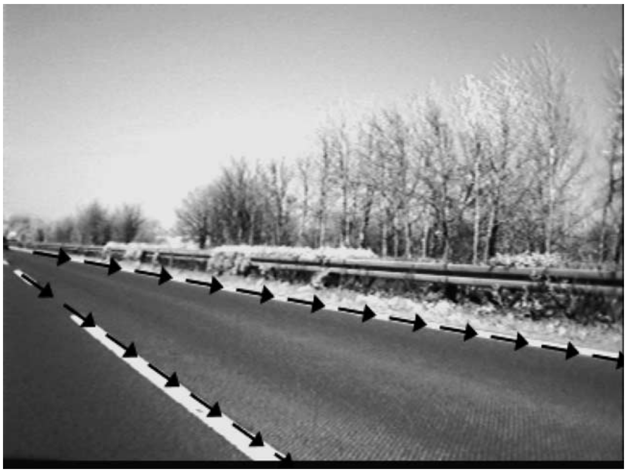

Fig. 4. The two lines (in arrows) that can be detected by a Hough transform.

In our specific case, we only have to deal with three unknown parameters, because of our simplifying assumptions and choice of coordinate systems. However, finding the three parameters from a single view image and without using a specifically designed calibration grid is an impossible task. We will show, however, that for our purpose it is not necessary to determine the three parameters. In fact we will prove that the crucial factor to get right is the ratio between two of those parameters. Our two important requirements are:

(1) The parallelism of lines lost in the projection process must be recovered by the IPM.
(2) Equal distances in the ‘median plane’ must be remapped into equal distances. Achieving this means that a car travelling at constant speed will also appear to have constant speed in the remapped image.

We will show that it is possible to solve the first problem in a way that also satisfies the second condition.

The main idea behind our calibration algorithm is to exploit the information contained in the lines that the road markings constitute. In our specific application we are helped by the fact that the lines are strongly outlined as can be seen in Fig. 4. We have found experimentally that, in most cases, when we applied a Hough transform to images from the sequence, the maximum count in the resulting accumulators corresponded to either of the two lines shown in Fig. 4. We will show that in theory using either of those lines should yield equivalent calibration parameters. This is however not the case in practice but in the next chapter we will present optic flow results obtained by calibrating with those two possible lines and show that the difference although noticeable is not critical.

Let us suppose the Hough transform results give us the equation of one of the two lines, that for convenience we will from now on call Line 1:

$$y = ux + v \tag{6}$$

The calibration problem consists of choosing the projection parameters θ, Z and c not knowing the real parameters θ_0, Z_0 and c_0. If the ‘median plane’ were viewed in a fronto-parallel configuration, the projection of Line 1 would be a horizontal line. Because the projection configuration was not fronto-parallel, the projection of Line 1 is a slanted line in the image plane but we can choose parameters θ and c so that Line 1 is remapped into a horizontal line. To see that let us invert the projection equations (Eq. (4)). We obtain:

$$X' = Z\frac{x - c\tan\theta}{-x\tan\theta + c} \tag{7a}$$

$$Y' = \frac{Z}{\cos\theta}\,\frac{y}{-x\tan\theta + c} \tag{7b}$$

To force all the points of Line 1 to be remapped onto a horizontal line, it suffices to substitute its equation, $y = ux + v$, into Eq. (7b) and to set Y' to a constant K:

$$Y' = \frac{Z}{\cos\theta}\,\frac{ux + v}{-x\tan\theta + c} = K \tag{8}$$

This gives us values for the three projection parameters:

$$\tan\theta = -u \qquad c = v \qquad \frac{Z}{\cos\theta} = K \tag{9}$$

u and v are fixed, given by the Hough transform results. However, K and by extension Z can be chosen arbitrarily and can be seen as a scaling parameter. Users can set this factor at their convenience to control the scaling.

We have therefore determined the three parameters we had to calibrate just by fulfilling the parallelism condition. We now have to investigate the consequences of this calibration for the remapping of distances. Will equal distances in the ‘median plane’ be remapped into equal distances in the reconstructed image?

To check that, let us consider a point in the ‘median plane’. Its horizontal coordinate is X_0. Following Eq. (4) its projection on the image plane will have horizontal coordinate:

$$x_0 = c_0\frac{X_0 - Z_0\tan\theta_0}{X_0\tan\theta_0 + Z_0} \tag{10}$$

Using our calibration method we now remap this point onto the reconstructed world plane. Following Eq. (7a), the new point has horizontal coordinate:

$$X_0' = Z\frac{x_0 - c\tan\theta}{-x_0\tan\theta + c} \tag{11}$$

Substituting the expression of x_0 (Eq. (10)) into Eq. (11) and after straightforward simplification, we obtain:

$$X_0' = Z\frac{X_0(c_0 + c\tan\theta_0\tan\theta) + Z_0(c\tan\theta - c_0\tan\theta_0)}{X_0(c\tan\theta_0 - c_0\tan\theta) + Z_0(c + c_0\tan\theta_0\tan\theta)} \tag{12}$$

Now recall that we can write tanθ and c as functions of the parameters u and v of Line 1 (Eq. (9)). The parameters u and v can themselves be written as functions of θ_0, Z_0, and c_0. According to our assumptions, Line 1 is the projection on the image plane of a horizontal line in the ‘median plane’. Let us write the equation of this horizontal line as:

$$Y' = k \tag{13}$$

Using the projection equations (Eq. (4)), it is easy to obtain:

$$u = -\left(\frac{\cos\theta_0}{Z_0}k\right)\tan\theta_0 \qquad v = \left(\frac{\cos\theta_0}{Z_0}k\right)c_0 \tag{14}$$

The consequence of writing u and v in this manner is that, in the expression of X_0' (Eq. (12)), the term in the denominator depending on X_0 cancels out. We are therefore left with:

$$X_0' = \lambda X_0 + \lambda_0, \quad \text{with} \quad \lambda = \frac{Z}{Z_0}\frac{c_0 + c\tan\theta_0\tan\theta}{c + c_0\tan\theta_0\tan\theta}$$
$$\text{and} \quad \lambda_0 = Z\frac{c\tan\theta - c_0\tan\theta_0}{c + c_0\tan\theta_0\tan\theta} \tag{15}$$

Note that λ and λ_0 are constants. This is sufficient to prove that equal distances in the 'median plane' are remapped into equal distances in the reconstructed world image because a distance $(X_1 - \mathrm{X}_2)$ in the 'median plane' will be remapped into:

$$X_1' - X_2' = \lambda X_1 + \lambda_0 - \lambda X_2 - \lambda_0 = \lambda(X_1 - X_2) \tag{16}$$

which is a proportional distance.

A number of remarks need to be made:

(1) The preceding proof holds only for horizontal distances. This is enough for our application because the car does not move in the vertical direction.

(2) In Eq. (12) the term in the denominator depending on X_0 cancels out if:

$$\frac{\tan\theta}{c} = \frac{\tan\theta_0}{c_0} \tag{17}$$

This is enough to guarantee the remapping of distances in the proportional manner expressed in Eq. (16). We do not need to get any of the projection parameters (θ_0, Z_0, c_0) right. Only the ratio between two of them matters.

(3) We said earlier that in theory we could equivalently use the line corresponding to the lane marking or the one corresponding to the edge of the road. That is because in the 'median plane' both lines are horizontal differing only, in their respective equations, in the position parameter k (Eq. (13)). Examining Eq. (14), it is easy to see that the ratio (u/v) and by extension $\tan\theta_0/c_0$ would be the same for both lines.

In fact neither of those two lines belongs to the 'median plane', that is, the car trajectory plane sectioning the car vertically in the middle. However, they belong to planes that are parallel to this plane. That means the projection equations are unchanged except for the parameter Z_0 that represents the distance from the 'median plane' to the image plane. Once again, Eq. (14) tell us that changing Z_0 will not affect the important ratio $\tan\theta_0/c_0$. An additional consequence is that it is also equivalent to perform the calibration using the road railings that are sometimes detected by the Hough transform.

We devised a test to roughly check our calibration method. Fig. 5(a) shows a checkerboard plane viewed in the configuration of Fig. 3. The arrowed line marks the line used to calibrate the IPM. Fig. 5(b) shows the resulting remapped image. To check that the remapping restores equal distances we measure the horizontal length of the reconstructed squares. There are seven complete squares in one row. Between the arrowed lines shown in Fig. 5(b), the distances, in pixels are: 65-66-65-67-67-65-68. Those distances are roughly equivalent.

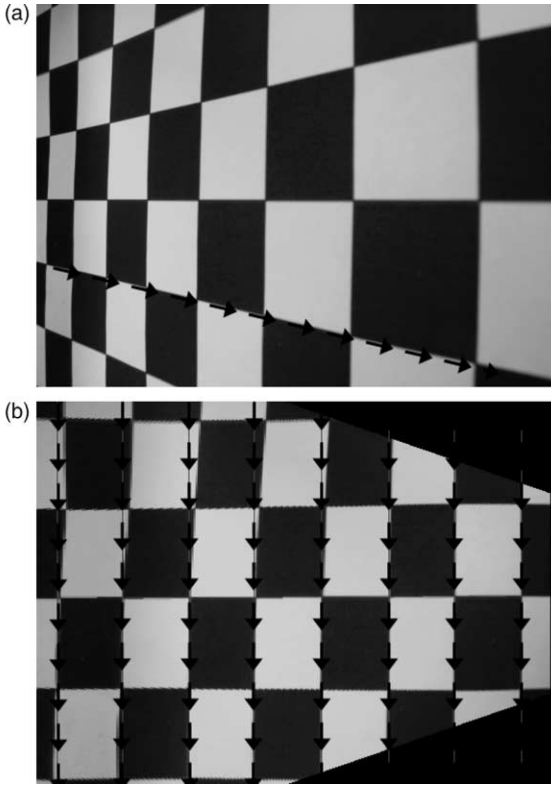

Fig. 5. Checkerboard pattern test for the inverse perspective mapping. (a) Original plane. The arrowed line is used for calibration. (b) Remapped plane. We measure the distance between the arrowed lines to verify that the remapping has restored equal distances.

5. Quantitative analysis of the optic flow

To test the IPM and our calibration method, we ran Johnston et al.'s optic flow algorithm [16] over an overtaking car sequence. The sequence was realised by Hella KG. The camera was placed on a car travelling at the constant speed of 90 km/h in the configuration shown in Fig. 6. In the adjacent lane, the overtaking car, also controlled by Hella KG, is travelling at the constant speed of 110 km/h. The frame rate is 25 frames per second.

To demonstrate the benefits of performing IPM, we want to show that in the remapped sequence the velocity of the car appears to be approximately constant while in the original sequence the same velocity varies greatly.

Fig. 7 shows an example of a remapped image. In Fig. 7(b) and (c) calibration is performed using the lines corresponding respectively to the edge of the road and the lane marking. In Fig. 7(b) the parallelism of the lane marking is not completely restored although according to our analysis either line should yield identical calibration results. This is probably due to small

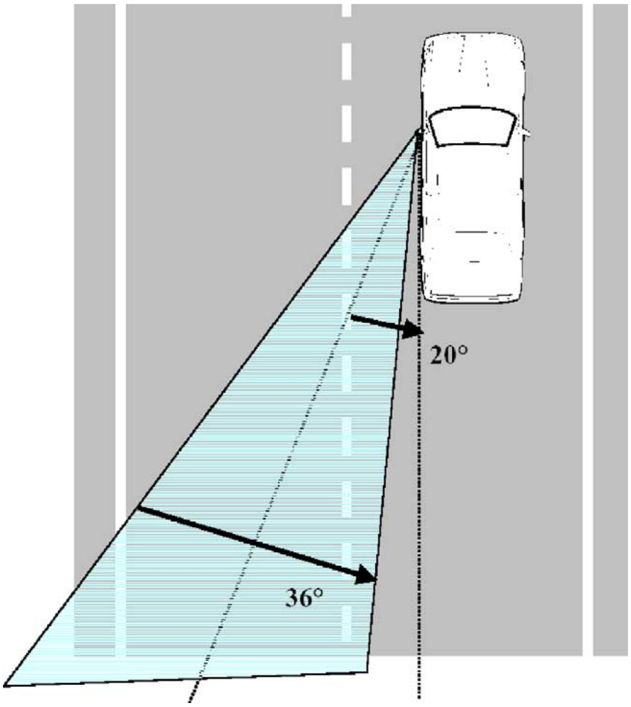

Fig. 6. Configuration for the acquisition of the overtaking car sequence (Courtesy of Hella KG).

violations of our simplifying assumptions. We will show however that the optic flow results are not significantly affected by this.

In our optic flow results, we manually segment the region corresponding to the car. We do that for 20 frames regularly spaced (every 4 frames in the sequence). Fig. 8(a) and (b) show seven of these frames (including the first and the last) for respectively the original sequence and the remapped one. It can be seen that those 20 frames cover a wide range of situations for the overtaking car from far away to close to the camera. In both figures the top row show the temporally filtered images on which spatial derivatives are performed to compute the optic flow [16]. The bottom row shows the segmented car. In both rows each image is 128 pixels in width and 96 in height. Because optic flow algorithms are computationally expensive, sub-sampling input images down to a smaller size is necessary to ensure a high processing rate.

The masks that correspond to the segmented car allow us to only include optic flow measures of the car when computing the mean speed (modulus of the velocity vector) and its standard error. It should be noted that not every point inside the mask provides a velocity measure. Obtaining dense optic flow is a difficult problem [2]. For example, a uniform region with no features at all is likely to yield null time and space derivatives that leads to undefined velocity (in that case human vision is also unable to detect motion).

Fig. 9(a) shows the results for the car's mean speed. It can be seen that for the original sequence the mean velocity accelerates while in the two remapped sequence it remains roughly constant. More precisely the ratio between maximum and minimum speed are, respectively, 9.0 for the original sequence, 1.3 for the remapped sequence calibrated with the lane marking, 1.5 for the remapped sequence calibrated with the edge of the road. Using the lane marking seems to work marginally better than using the edge of the road. That is probably because the side of the car facing the camera is closer to the lane marking than to the edge of the road. In our experiments, we found that the Hough transform is more likely to indicate the edge of the road as its maximum than the lane marking. The optic flow results show that although there is a slight disadvantage at using the edge of the road, it does not seem substantial.

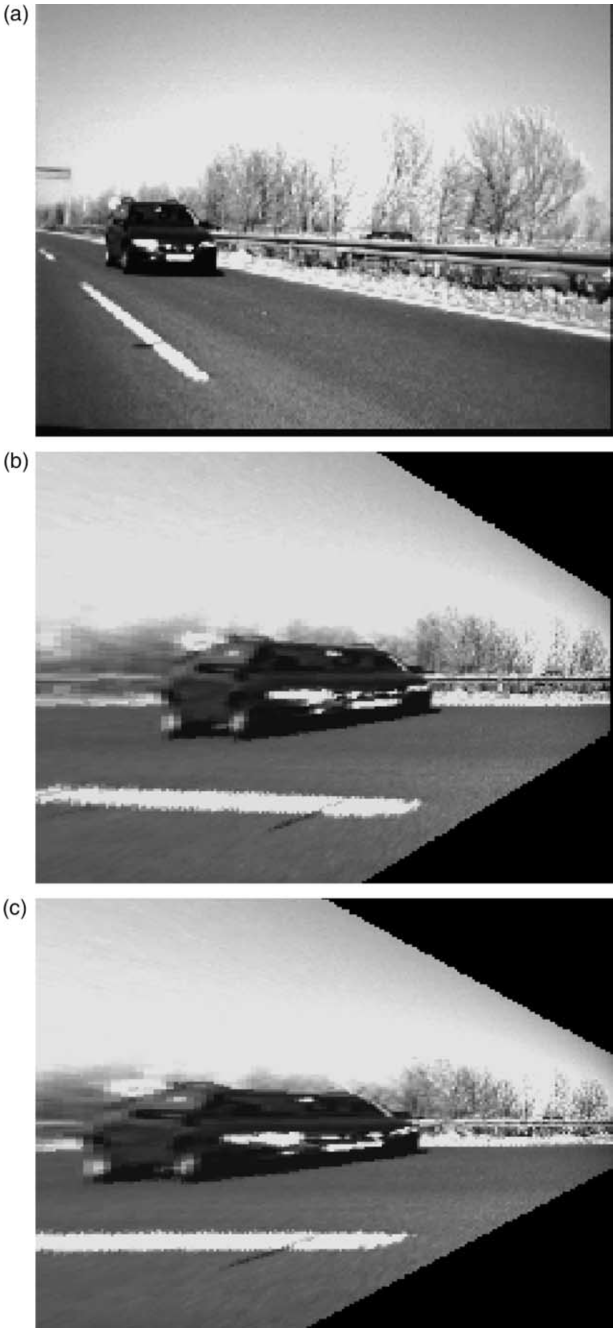

Fig. 7. Example of remapping. (a) Original image. (b) Remapping calibrated with the edge of the road. (c) Remapping calibrated with the lane marking.

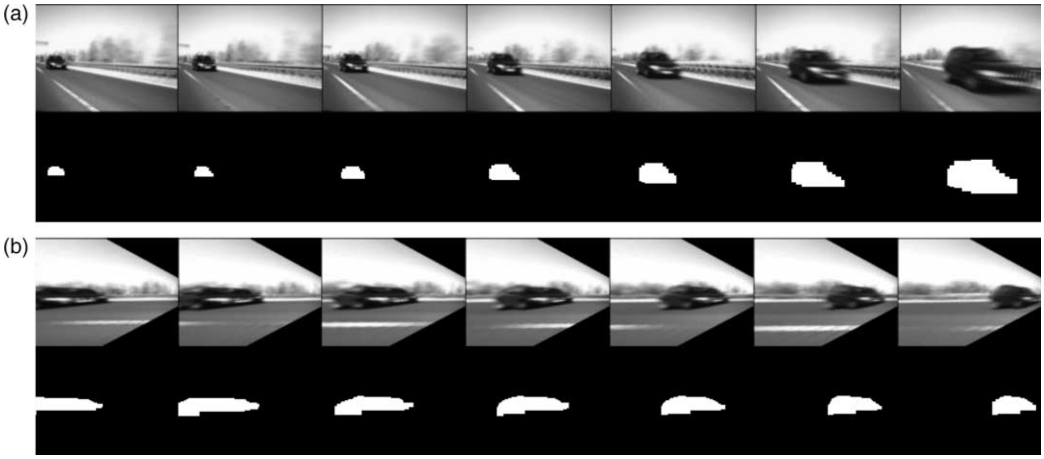

Fig. 8. Manually segmented cars in the (a) original sequence (b) remapped sequence. In both cases, the top row are temporally filtered images from which spatial derivatives are computed for the optic flow algorithm.

Fig. 9(b) shows the standard error associated with speed measures over the car. The standard error is defined by:

$$\Delta = \frac{\sqrt{\sum_{i=1}^{N}(v_i - \bar{v})^2}}{N} \tag{18}$$

where $\bar{v}$ represents the mean speed and N the number of speed measures over the segmented car. To provide a more intuitive error measure we normalise the standard error by the mean so that the error can be expressed as a percentage of the mean speed. We can compute the mean standard error over the 20 selected frames. It is 1.5% for the original sequence, 1.1% for the remapped sequence calibrated with the lane marking and 0.93% for the remapped sequence calibrated with the edge of the road. From Fig. 9(b) it can be seen that for the remapped sequences the standard error is much more stable than for the original sequence. The error is larger for the original sequence at the beginning and at the end of the sequence. At the beginning of the sequence the car is small in the original sequence so that, the statistical population being small, the mean error is more likely affected by a few outliers. In the remapped sequences, the size of the car is more constant, which helps explain the stability of the standard error. At the end of the sequence, the high error for the original sequence is probably due to large pixel displacements (high car image speed), which can present problems to most gradient-based optic flow techniques.

The results show that the camera axes assumptions that simplified our analysis do not represent a strong constraint. In fact, when making the sequence, no special care was taken in placing the camera in an exactly upright position with the base parallel to the ground. The method seems relatively robust to small infringements of our assumptions about camera positioning. However, our method is also based on the assumption that the car is travelling in a straight line and that lane markings are visible. We have not tested our algorithm in the case of a curved road because there were no curves in the sequences provided to us by Hella KG. The application is mainly intended for large, multi-carriageway roads, where changing lanes at high speed can be dangerous.

Using the remapping as a pre-processing step, it is possible to extract the car's real speed by calibrating once with a car

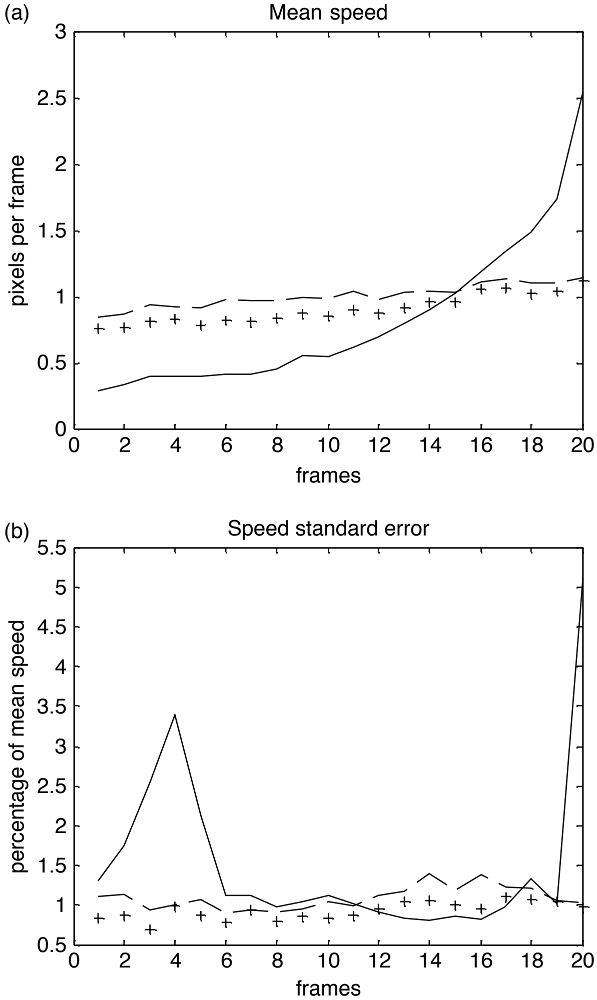

Fig. 9. Optic flow measures over the car: (a) mean speed (b) standard error. The solid, dashed, plus lines respectively represent the original, remapped with lane marking calibration and remapped with edge of the road calibration sequences.

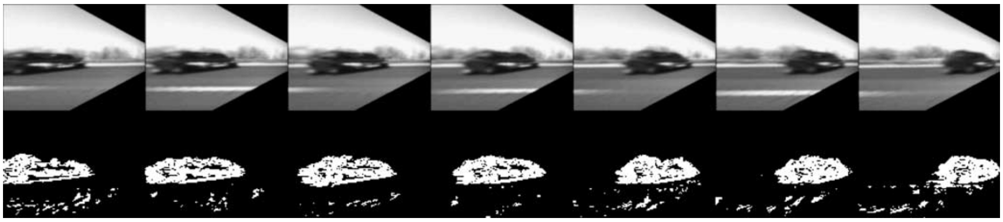

Fig. 10. Automatically segmented cars in the remapped sequence. The top row is temporally filtered images from which spatial derivatives are computed for the optic flow algorithm.

travelling at a known speed. The calibration should provide a proportionality constant between the optic flow speed and the real 3D speed. That calibration constant however will be valid only for a fixed distance between the camera and car trajectory plane but that should be the case in our application since we only need to detect cars in the adjacent lane. Inside a lane, however, there is room for the distance to change, as drivers cannot be expected to occupy the same position inside the lane and to keep that position indefinitely. Testing the sensitivity of the method to distance variation will be the subject of our future work.

6. Automatic optic flow segmentation

Automatic optic flow segmentation is a difficult problem that is still attracting much research [17–19]. Although many methods have been proposed, none can be said to be infallible. Evaluating the validity of IPM on one of them would not constitute a fair test since none of them is guaranteed to segment the car perfectly. For this reason, we resorted to manual segmentation to test the IPM. Although manual segmentation cannot claim to be above all contention, it can at least be said to pass the rough test of visual inspection and should be considered for now as delivering the only available if imperfect ground truth.

Attempts to solve the problem of automatic optic flow segmentation in the most general case invariably produce computationally expensive and time-consuming algorithms that are difficult to implement in real-time. In this section, we show that because the IPM reduces the general case into the special case of a fronto-parallel configuration it can lead to a simple and effective segmentation algorithm.

In our particular application, in the fronto-parallel case, the overtaking car moves rightwards while the background moves in the opposite direction. It would seem that just distinguishing rightward from leftward motion could suffice to segment the overtaking car. We implemented an algorithm in which we select optic flow vectors with directions between -90 and $+90°$. Fig. 10 shows some resulting frames for the car sequence calibrated with the lane markings. It can be seen that, although the segmentation does not seem quite as accurate as in the manual case, the errors appear to be relatively minor. Using this segmentation method we computed the speed of the car in 77 frames encompassing the 20 manually segmented frames used in the previous chapter. Fig. 11 shows the results. As for the manually segmented frames the speed is fairly constant. The ratio between highest and lowest speed is 1.5, a bit up from 1.3 obtained with manual segmentation, but we are using many more frames. It can be seen that the speed measures obtained from automatic segmentation are slightly higher than the ones from manual segmentation. The main reason for the difference is that a threshold is used in the automatic segmentation to cut out noise from the background while in the manually segmented sequence, we take all values greater than 0 no matter how small they might be. The threshold explains the patchy appearance of the car in the automatically segmented sequence. The important result remains that for both segmentation methods the constant speed of the car seems to have been recovered.

An additional advantage of the IPM therefore is that it naturally leads to a simple segmentation method whose results have been shown to be similar to those of our reference manual segmentation. It should be noted that such a left–right segmentation would not work with the original unprocessed sequence. It can be seen from Fig. 12 that in that sequence the noise is non-negligible especially in the right-hand part of the image where speed vectors are larger and less reliable. In some frames of Fig. 12 the area covered by noise is as big as that of the car.

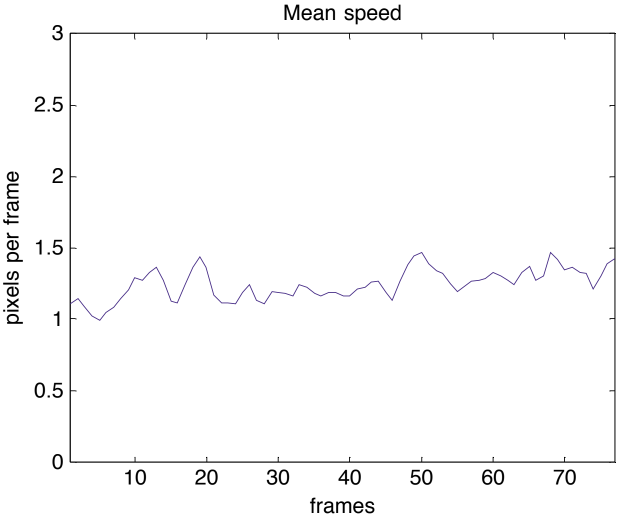

Fig. 11. Mean car speed in 77 frames obtained from the automatically segmented sequence shown in Fig. 10.

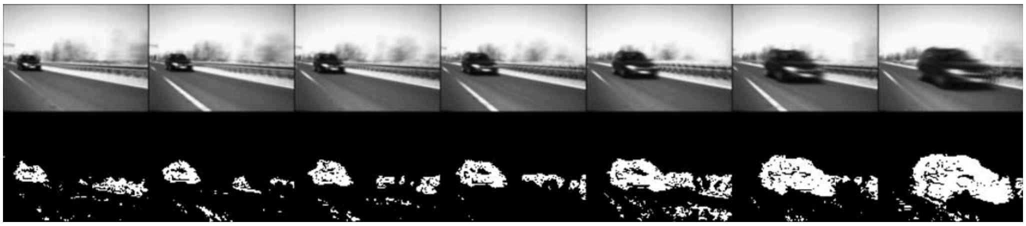

Fig. 12. Automatically segmented cars in the original sequence. The top row is temporally filtered images from which spatial derivatives are computed for the optic flow algorithm.

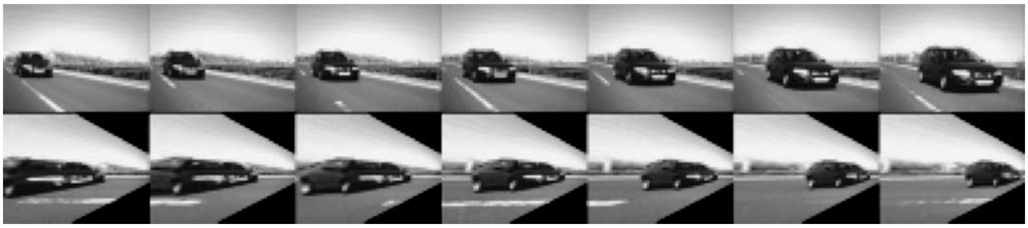

Fig. 13. Additional car sequence: in the top row the original sequence and below the corresponding remapped frames.

We processed 63 frames from another overtaking car sequence provided by Hella. Fig. 13 shows some of the frames. As in the previous sequence the relative speed between the two cars is a constant 20 km/h. Processing the sequence with the IPM, the optic flow and automatic segmentation algorithms we obtain the speed measures shown in Fig. 14. Once again the speed seems fairly constant. The ratio between the highest and lowest speed is 1.3.

7. Robustness of the algorithm

We now examine how different weather conditions can affect the algorithm. The crucial factor for the IPM is the visibility of the lines (road edge or lane markings). If those line can be detected the IPM can reliably compensate for the perspective effect. Whether the speed of the car can be accurately recovered is then dependent of the optic flow algorithm. We test the line detector with two images of increasing difficulty. The first image (which can be seen in Fig. 15) was taken from a Hella sequence shot on a very cloudy day. In the second (Fig. 16) fog and low light severely limits the visibility of the lines.

The line detector is based on Matlab's implementation of the radon transform. It should be noted that in our application we approximately know the orientation of the edge of the road and lane markings because the camera is always placed roughly in the same position. This knowledge will greatly help us reduce the range of angles over which to search. A simple inspection of any of the images tells us that the angle of both road edge and lane markings is between 0 and 90° (using the convention that 0° means the horizontal direction and angles are positive in the clockwise direction). The angle of the road edge is usually below 20°. Since the lane markings seem to lead to a better speed constancy recovery we can restrict the search to angles between 20 and 80°.

Fig. 15(b) shows the result of the radon transform for the cloudy image. Two peaks are clearly visible. The highest correspond to the right hand side of the lane marking. The line is plotted in white in Fig. 15(a). The second peak corresponds to the left hand side of the lane marking. Prior to the radon transform, edge detection must be carried out. For this image we found that a simple Sobel edge detector [20] was sufficient.

However, for the more difficult image of Fig. 16 we found that the more sensitive (but also more computationally expensive) Canny edge detector [21] was necessary. Fig. 16(b) shows the results of the radon transform. The two peaks are still visible but are now in a more noisy background. As for the previous image the line corresponding to the highest peak is drawn in white in Fig. 16(a).

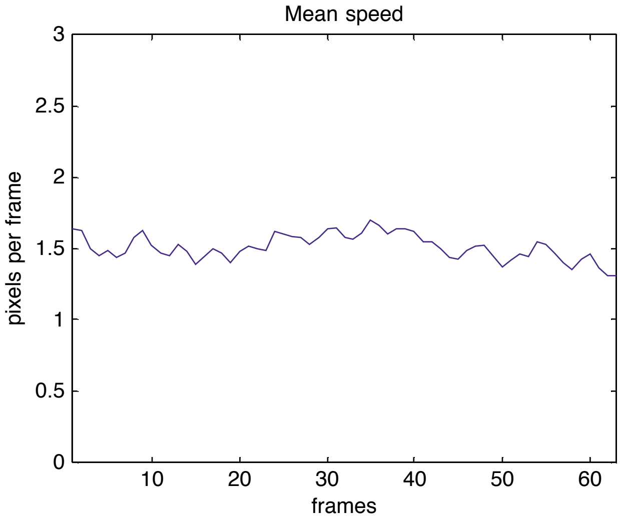

Fig. 14. Mean car speed in 63 frames obtained from the additional sequence shown in Fig. 13. Automatic optic flow segmentation is used.

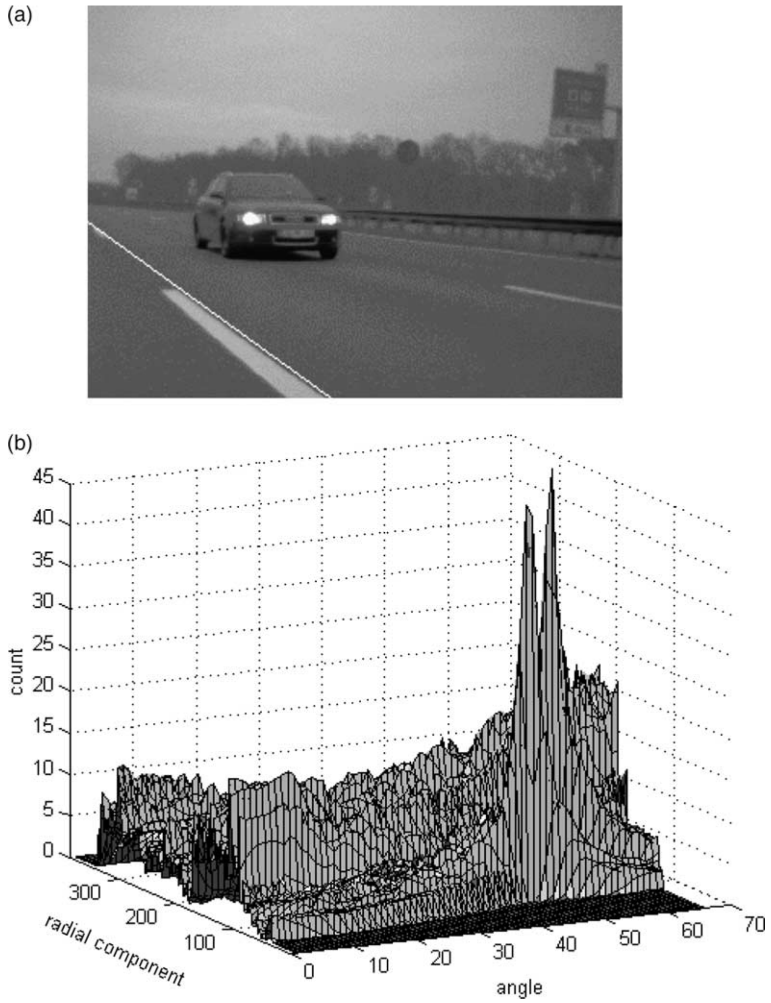

Fig. 15. Line detection in cloudy conditions. (a) The white line on the right of the lane marking corresponds to the highest peak of the radon transform. (b) Results of the radon transform.

Those experiments show that lane markings, that are in general designed to be as visible as possible (white on a dark surface), are indeed detectable by a robust computer algorithm in the most varied weather conditions. In addition, calibration statistics could be retained over time to provide calibration estimates if no data about line markings is available.

However, the possibility of errors in the calibration parameters must still be considered. We now study how robust the IPM is to such errors. We showed in Section 4 that recovering speed constancy is dependent only on the ratio $\tan\theta_0/c_0$. We can therefore simplify our analysis of the IPM's robustness to calibration errors by examining how speed constancy will vary in respect to that single variable, $\tan\theta_0/c_0$. We will consider errors on that ratio in the range of -20 to $+20\%$ by increments of 5%. Fig. 17 shows the consequences of such errors on the remapping of images of the car sequence. The five frames, respectively, correspond to errors of -20, -10, 0, $+10$ and $+20\%$ (we do not show -5, -15%,…). The reference (0% error) sequence is once more the one calibrated with the lane markings.

For each new (and erroneous) set of calibration parameters we remap the entire first overtaking car sequence (the one used in Section 5). The sequence is then run through the optic flow algorithm. We then apply the automatic optic flow segmentation presented in Section 6. Finally, the car speed is computed for 77 frames as in Section 6 and we evaluate the ratio between maximum and minimum speed. How this ratio fluctuates with the calibration error is shown in Fig. 18. It can be seen that the algorithm is fairly robust to errors in the ratio in the positive direction. An error of $+20\%$ will just bring the maximum minimum speed ratio from 1.5 to 1.65. However, errors in the negative direction will cause a more pronounced degradation in the speed constancy. An error of -20% will bring the maximum minimum speed ratio to 2.1. This asymmetry could be explained by the fact that negative errors, as can be seen in Fig. 17, cause the car to be remapped at a smaller scale. Noise from the background could then become significant.

8. Conclusion

Extracting meaningful high-level information from raw optic flow data is a difficult task. We have shown that the complications caused by perspective projection on speed estimation can be addressed by an appropriate remapping. To

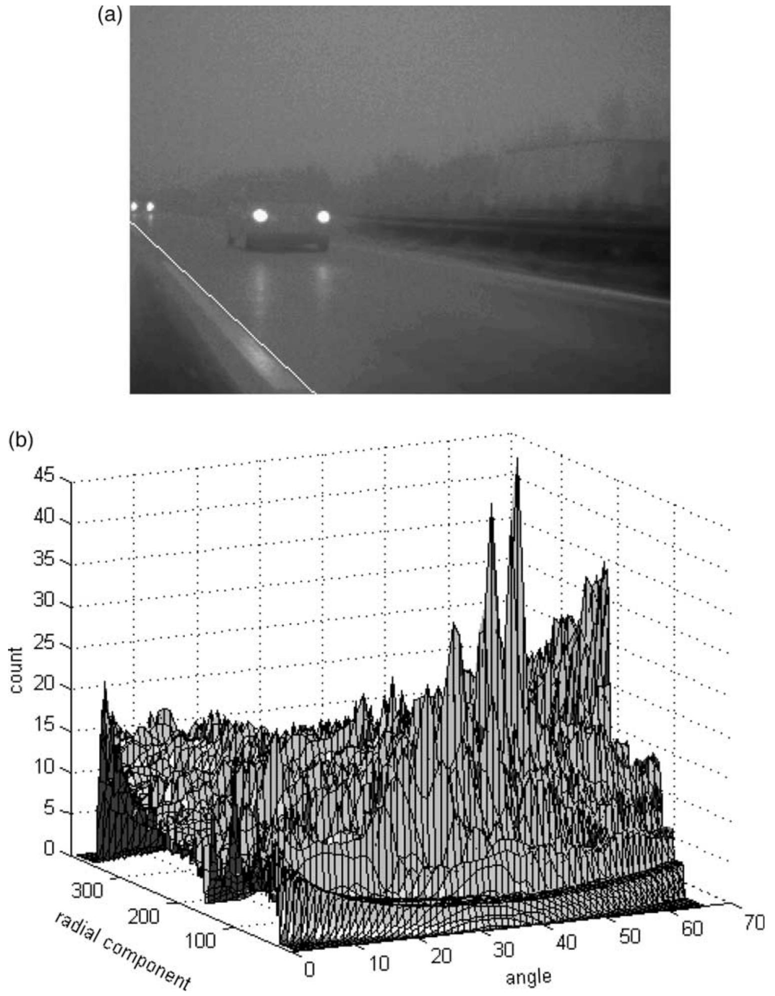

Fig. 16. Line detection in foggy and low light conditions. (a) The white line on the right of the lane marking corresponds to the highest peak of the radon transform. (b) Results of the radon transform.

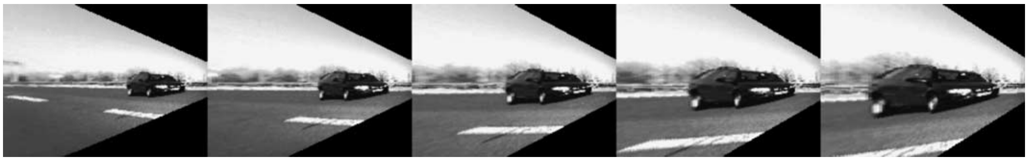

Fig. 17. Consequence of calibration errors on the remapped image. From left to right calibration error of -20, -10, $0, +10$ and $+20\%$.

demonstrate that the IPM can be useful in practical situations we have emphasised quantitative optic flow measures. We have segmented the velocity measures corresponding to the car and computed the mean speed. We have found that performing the IPM dramatically reduces the measured image acceleration due to perspective projection. The ratio between maximum and minimum speed can be reduced from 9.0 for the original sequence to just 1.3 for the remapped sequence. It should be stressed that for all our work, real-life sequences were used.

We have also devised a calibration technique adapted to our application. Its simplicity allows a very fast implementation on either software or hardware. Quite remarkably, although it is only based on restoring parallelism, it can also be shown to remap equal distances in the real scene into equal distances on the reconstructed plane, which is essential to ensure that uniform speed appears uniform. We have found experimentally from real-life sequences that this calibration is possible even in bad weather conditions where visibility is limited.

Besides speed estimation from optic flow, the IPM can also be used for obstacle detection and lane detection. IPM has also recently been shown to improve car tracking significantly [22]. In a similar way we have also shown that the IPM can simplify the optic flow segmentation problem.

Finally, it should be noted that although remapping represents an overhead, its benefits offset the cost. Bertozzi et al. have proven that the IPM can realistically be integrated

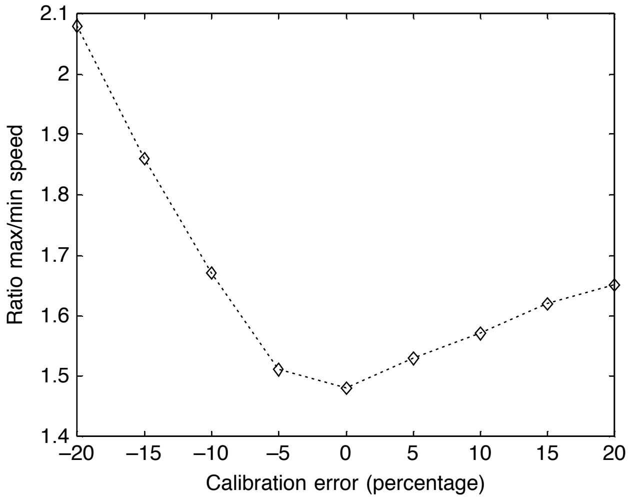

Fig. 18. Maximum minimum speed ratio for the 77 frames sequence remapped with some calibration error.

in a driver assistance application by implementing it in hardware and showing how fast the remapping process can be [7].

Acknowledgements

The research was supported by the project grant ECOVISION, IST-2001-32114. The authors would like to thank Martin Muhlenberg and Alexander Rotter from Hella KG for supplying the car overtaking sequences.

References

[1] J.J. Gibson, The Perception of the Visual World, Houghton Mifflin, Boston, 1950.

[2] J.L. Barron, D.J. Fleet, S.S. Beauchemin, Performance of optical flow techniques, International Journal of Computer Vision 12 (1994) 43–77.

[3] S.S. Beauchemin, J.L. Barron, The computation of optical flow, ACM Computing Surveys 27 (1995) 433–467.

[4] D.N. Lee, A theory of visual control of braking based on information about time-to-collision, Perception 5 (1976) 437–459.

[5] F.G. Meyer, Time-to-collision from first-order models of the motion field, IEEE Transactions on Robotics and Automation 10 (1994) 792–798.

[6] H.A. Mallot, H.H. Bulthoff, J.J. Little, S. Bohrer, Inverse perspective mapping simplifies optical flow computation and obstacle detection, Biological Cybernetics 64 (1991) 177–185.

[7] M. Bertozzi, A. Broggi, GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection, IEEE Transactions on Image Processing 7 (1998) 62–81.

[8] M. Bertozzi, A. Broggi, A. Fascioli, Stereo inverse perspective mapping: theory and applications, Image and Vision Computing 16 (1998) 585–590.

[9] D.A. Forsyth, J. Ponce, Computer Vision a Modern Approach, Prentice-Hall, 2003.

[10] H.C. Longuet-Higgins, K. Prazdny, The interpretation of a moving retinal image, Proceedings of the Royal Society of London, Series B: Biological Science 208 (1980) 385–397.

[11] K. Prazdny, Estimation and relative depth from optical flow, Biological Cybernetics 36 (1980) 87–102.

[12] D.J. Heeger, A.D. Jepson, Subspace methods for recovering rigid motion I: algorithm and implementation, International Journal of Computer Vision 7 (1992) 95–117.

[13] C. Garcia, G. Tziritas, Optimal projection of 2-D displacements for 3-D translational motion estimation, Image and Vision Computing 20 (2002) 793–804.

[14] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.

[15] O.D. Faugeras, Three-Dimensional Computer Vision: A Geometric Viewpoint, MIT Press, 1993.

[16] A. Johnston, P.W. McOwan, C.P. Benton, Robust velocity computation from a biologically motivated model of motion perception, Proceedings of the Royal Society of London, Series B: Biological Science 266 (1999) 509–518.

[17] T. Zhang, C. Tomasi, On the consistency of instantaneous rigid motion estimation, International Journal of Computer Vision 46 (2002) 51–79.

[18] A.S. Ogale, C. Fernmuller, Y. Aloimonos, Motion segmentation using occlusions, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 988–992.

[19] K. Pauwels, M.M. Van Hulle, Robust instantaneous rigid motion estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, 2005.

[20] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, 1993.

[21] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986) 679–698.

[22] S. Mota, E. Ros, J. Díaz, S. Tan, J. Dale, A. Johnston, Detection and tracking of overtaking cars for driving assistance, Early Cognitive Vision Workshop, Isle of Skye, UK, 2004.