Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Stabilization of Inverse Perspective Mapping Images based on Robust Vanishing Point Estimation

Marcos Nieto, Luis Salgado, Fernando Jaureguizar, and Julián Cabrera
Grupo de Tratamiento de Imágenes - E. T. S. Ing. Telecomunicación
Universidad Politécnica de Madrid - Madrid (Spain)
Email: {mnd, L.Salgado, fjn, jcq}@gti.ssr.upm.es

***Abstract*— In this work, a new *Inverse Perspective Mapping* (IPM) technique is proposed based on a robust estimation of the vanishing point, which provide *bird-view* images of the road, so that facilitating the tasks of road modeling and vehicle detection and tracking. This new approach has been design to cope with the instability that cameras mounted on a moving vehicle suffer. The estimation of the vanishing point relies on a novel and efficient feature extraction strategy, which segmentates the lane markings of the images by combining a histogram-based segmentation with temporal and frequency filtering. Then, the vanishing point of each image is stabilized by means of a temporal filtering along the estimates of previous images. In a last step, the IPM image is computed based on the stabilized vanishing point. Tests have been carried out on several long video sequences captured from cameras inside a vehicle being driven along highways and local roads, with different illumination and weather conditions, presence of shadows, occluding vehicles, and slope changes. Results have shown a significant improvement in terms of lane width constancy and parallelism between lane markings over non-stabilized IPM algorithms.**

I. INTRODUCTION

Focusing on the field of driver assistance systems, two major objectives are road modeling and vehicle detection within in-vehicle vision systems. Usually, the road model is firstly computed to obtain a reliable environment description which afterwards allows to accurately detect vehicles. For this purpose, there are typically two main processing stages, *features extraction*, the module which extracts features from images, and *model fitting*, the module that uses those features to obtain the number of lanes, their width or curvature to compose an accurate model of the road. Most works found in literature detect, as features, the lane markings which delimite the road boundaries [1]-[4]. Combined with them, other features, like the color of the road, are also used [5].

Features extraction may deliver erroneous results due to severe illumination changes, occlusions or absence of features (lane markings may be occluded by cars, or even do not exist at all). Model fitting is the processing module that introduces *a priori* knowledge to constrain the problem of obtaining a general model of the road and achieve adequate results. For instance, [6] assumes that the road is flat, without slope changes and with parallel lane markings, whereas other works make use of highly accurate models such as B-snakes [3] or clothoids [7].

These processing modules are applied on images captured from video cameras installed in the vehicles. If the inherent perspective effect is removed, further image analysis is simplified. In this sense, many authors use a single transform, usually denominated *Inverse Perspective Mapping* (IPM) [6], which removes the perspective distortion from the images. This transform delivers a *bird-view* image, which facilitates the tasks of road modeling [2][9][10] and vehicle detection [11][12].

Most of these works assume that the road is flat and the position and rotation of the camera with respect to the road is known, i.e. the camera is initially calibrated. However, these assumptions may be inaccurate taking into account that cameras are mounted on moving vehicles and the road is far from being perfectly flat. This underlying instability may provoque inaccuracies in the transform, obtaining incorrect IPM images, understanding as good IPM images those that show lanes with constant width and parallel lane markings.

To solve the limitations associated with these assumptions, some authors working with IPM consider, independently, better road models [8], or calibration techniques which deliver the relative position and rotation of the camera [9][13].

In this paper a combined stabilization strategy is proposed, which automatically generates correct IPM images taking into account not only the position and rotation of the camera, but also, in the same model, the variations of the slope of the road.

It is based on robust estimations of the vanishing point of the scene through an efficient features extraction module, which segmentates the lane markings of each image within the video sequence. The vanishing point estimation is then used to obtain the value of the angles that define the rotation of the camera with respect to the road. With these parameters the *camera projection matrix* is computed to build the stabilized IPM images for each image of the video sequence. Dynamic adaptability to the road slope variations and global instability is ensured.

The paper is organized as follows: section II presents the overall strategy; section III explains each of the processing modules composing the features extraction; the vanishing point estimation is described in section IV, while section V depicts the generation of the stabilized IPM image. Finally, sections VI and VII show the obtained results and conclusions.

1-4244-1068-1/07/$25.00 ©2007 IEEE.

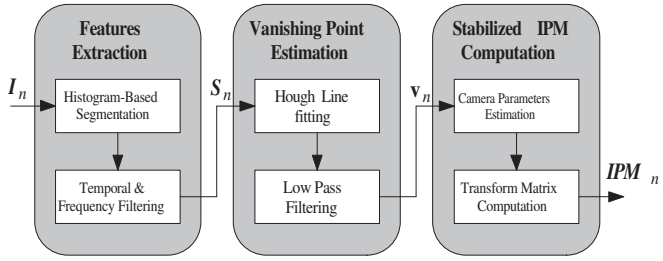


Fig. 1. Proposed Strategy Block Diagram: for each input image I_n a stabilized IPM_n image is generated

II. STRATEGY OVERVIEW

The aim of this work is to provide an automatic, robust and adaptive strategy to generate stabilized IPM images which may be used for road modeling and vehicle detection. The block diagram of the proposed strategy is depicted in Fig. 1. As it is shown, the three main processing modules are: i) *Features Extraction*: this module takes as input each image coming from the camera, I_n, performing a segmentation, S_n that classifies pixels as belonging or not to lane markings; ii) the *Vanishing point estimation* of each image, understood as the point of the image where the lanes seem to converge, $\mathbf{v}_n$, is obtained from S_n as the intersection of the straight lines computed with the Hough transform, followed by a low pass filtering to stabilize the evolution of vanishing point locations in the images; iii) *Stabilized IPM computation* identifies the relative position and rotation of the camera jointly with the slope of the road and generates the stabilized IPM image, IPM_n.

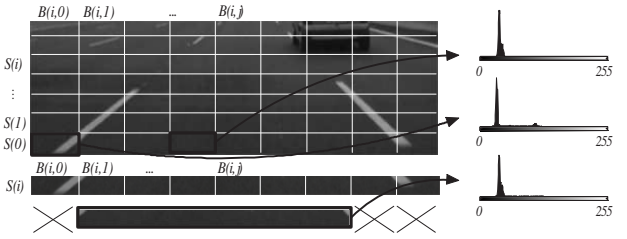
III. FEATURES EXTRACTION

The best way to describe a road is given by the homogeneity of the intensity values of the road and the lane markings that define lanes in almost any well-painted road. In this work, the features to be extracted are these lane markings, which are segmented in S_n.

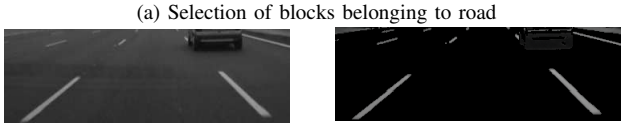
For that purpose, a histogram-based segmentation firstly defines the area of the image relative to the asphalt of the road. By setting to zero this area a new image is obtained where non-zero pixels are lane markings and other elements. Temporal filtering converts discontinuous lane markings into continuous ones while the high-pass frequency filtering followed by an automatically adaptive thresholding stage finally detect the lane markings as those pixels with both high intensity values and high response values to the filter. The details of these steps are described in the following subsections.

A. Histogram-based segmentation

An adaptive histogram-based segmentation is applied on the lower part of I_n whose pixels are supposed to mainly correspond to the road asphalt. This region of interest (ROI) is divided into horizontal slices, $S(i)$, which are sub-divided into blocks, $B(i,j)$. Fig. 2 (a) shows this sub-division strategy, where most of blocks completely belong to the asphalt of the road, while only a few include lane markings or vehicle pixels.



(a) Selection of blocks belonging to road



(b) Original Image (c) Result image

Fig. 2. Histogram-based segmentation

Fig. 3. Temporal filtered image: (left) k previous images, and (right) T_n

The histogram of a block whose pixels belongs to the asphalt shows one single mode, with a mean value $\mu_{i,j}$ and a standard deviation $\sigma_{i,j}$ describing the gray levels of the asphalt. An example of this single mode is shown in Fig. 2 (a) in the upper histogram. For each horizontal slice, only the blocks having similar $(\mu_{i,j}, \sigma_{i,j})$ are used to compose a global histogram of the slice. The other blocks, likely containing lane markings, are automatically discarded as their histogram show more than one single mode and, therefore, different $(\mu_{i,j}, \sigma_{i,j})$. The middle histogram of Fig. 2 (a) shows this type of distribution. For each slice, a new histogram is computed with the pixels of blocks classified as belonging to the asphalt. The lower histogram shown in Fig. 2 (a) correspond to the example slice, where the single mode has a mean value μ_i and standard deviation σ_i. For each slice, asphalted area is that composed by the pixels with intensity values comprised in $\{\mu_i \pm \sigma_i\}$.

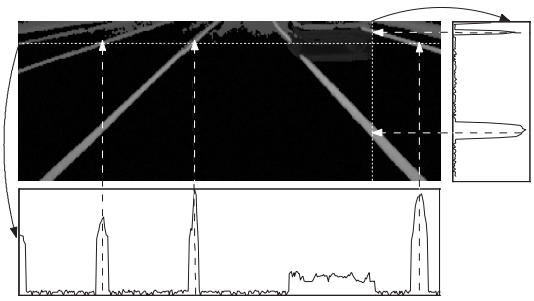
After computing each slice, the pixels belonging to the asphalt are set to zero, obtaining the image shown in Fig. 2 (c). Further filtering processing stages will benefit from this resulting contrast improvement.

B. Temporal filtering

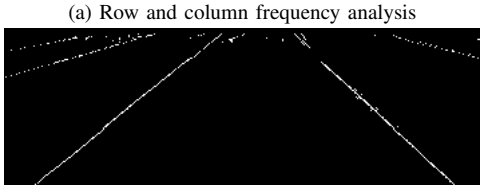
A temporal filtered image, T_n, is generated as in (1)

$$T_n(x,y) = \max_k \{I_{n-k}(x,y), \ldots, I_n(x,y)\} \qquad (1)$$

where (x,y) are pixel coordinates inside the image. As a result, discontinuous lane markings in the images I_n become continuous in T_n and easier to detect by further analysis, as seen in Fig. 3.



(a) Row and column frequency analysis



(b) Result after row and column frequency analysis

Fig. 4. Horizontal and vertical frequency filtering based on an automatic thresholding technique and pattern matching.

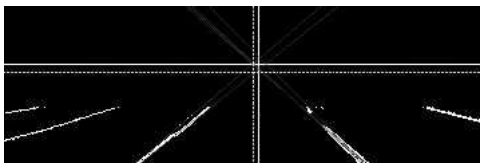


Fig. 5. Vanishing point: computation of $\mathbf{v}_n$ shown in solid white lines, and $\overline{\mathbf{v}}_n$, in dotted lines.

C. Frequency filtering

For each row and each column, the evolution of the gray levels of T_n is analyzed in order to locate lane markings based on the following properties: i) they appear as bursts of high intensity values, and ii) mostly surrounded by zeros. This properties can be seen in Fig. 4 (a), where the intensity values of an example row and column are shown. The row contains three high and narrow peaks corresponding to the lane markings, and other group of values, as a lower and wider peak, belonging to the vehicle. In the example corresponding to the selected column, while the column example shows two peaks surrounded by lower values.

The lane markings are detected by applying one-dimensional horizontal and vertical derivative filters, which enhance steep intensity changes, followed by an adaptive thresholding technique that classify as lane markings only those pixels with high values in T_n. This way, two binary images are obtained, which are simply fused with an OR operation.

For each row and column, the filter $D = [-1\ \underline{0}\ 1]$ is applied, obtaining signed values enhancing steep changes. For each pixel, the threshold $t(x, y)$ is in inverse proportion to $T_n(x, y)$, given by (2).

$$t(x, y) = A \cdot e^{-\tau T_n(x,y)} \tag{2}$$

where A is a scale factor (255 for 8-bit gray images), and $\tau = \ln 255/255$ is the decreasing factor of the exponential. The higher the pixel value is, the lower threshold so that pixels with high values (likely belonging to lane markings) easily exceed the threshold, whereas other pixels, with lower intensity values, will obtain higher thresholds. In the abovementioned examples, the threshold ensures that only the peaks relative to the lane markings exceed the threshold, while other intensity changes, like the ones relative to the vehicle in row example in Fig. 4, do not exceed their thresholds.

As a result, each pixel obtain one of these three values: $\{-1, 0, 1\}$. The pattern that describes a lane marking within a row or a column is a 1 followed by an arbitrary number of zeros and ended by a -1. When found, the central pixel of the pattern is set as 1 in the resulting binary image, setting the others to zero.

After computing all rows and columns, the binary image S_n is obtained, as shown in Fig. 4 (b).

IV. VANISHING POINT ESTIMATION

Once the lane markings are clearly identified in S_n, the following step is to fit straight lines to the lane markings, so that the vanishing point, $\mathbf{v}_n$, is computed as their intersection.

A. Line Fitting

The well known Hough Transform [1][14], which is robust against outliers while offering multiple line fitting, is used. The selection of the local maxima of the transform space is performed with the conjugate gradient method [15], initialized with the maxima of the previous image.

The vanishing point, which is, in a perspective image, the point to which parallel lines appear to converge, is consequently obtained as the intersection point of the straight lines that characterize the lane markings. From the Hough transform each line is parameterized with an angle θ and a distance ρ as in (3):

$$y \cdot \cos\theta + x \cdot \sin\theta = \rho \tag{3}$$

However, as there is not a unique intersection point, the vanishing point is selected as the solution of the overdetermined system of equations, shown in (4), build with the equations of each detected line.

$$\left[\ \mathbf{c}\ \middle|\ \mathbf{s}\ \right] \cdot \mathbf{v} = \mathbf{p} \tag{4}$$

where $\mathbf{v} = (y, x)^T$, $\mathbf{c} = (\cos\theta_0, \ldots, \cos\theta_{r-1})^T$, $\mathbf{s} = (\sin\theta_0, \ldots, \sin\theta_{r-1})^T$, and $\mathbf{p} = (\rho_0, \ldots, \rho_{r-1})^T$. This system is solved with singular value decomposition (SVD), giving the least squares error single solution $\mathbf{v}$ to the system.

B. Low pass filtering

The vanishing point of the n-th image is stabilized through a low-pass time filter considering a window composed by the m previous vanishing points as in (5)

$$\overline{\mathbf{v}}_n = \overline{\mathbf{v}}_{n-1} - \frac{1}{m}(\mathbf{v}_{n-m} + \mathbf{v}_m) \tag{5}$$

where $\overline{\mathbf{v}}_k$ and $\mathbf{v}_k$ are, respectively, the vanishing point estimation, and the computed vanishing point, as in (4), for the

instant k. The temporal filtering ensures that outlier vanishing points, due to errors in the features extraction processing module, do not affect significantly the final estimation $\overline{\mathbf{v}}_n$.

Fig. 5 shows the vanishing point estimation for one image. The vertical and horizontal white lines intersect at the computed $\mathbf{v}_n$, while the final estimation, $\overline{\mathbf{v}}_n$ is shown in dotted lines.

V. STABILIZED IPM COMPUTATION

The vanishing point offers very useful information about the position and rotation of the camera with respect to the road. The camera yaw and pitch angles are obtained from the coordinates of the vanishing point, thus allowing to reconstruct the calibration matrix which describes how elements in the real world are projected into the image [16]. This image is then retro-projected into the road plane to obtain the stabilized IPM image.

A. Camera Parameters estimation

Regarding the computation of IPM images, the most important parameters are those corresponding to the rotation of the camera inside the car [6]: the pitch, yaw and roll angles. The translation parameters, understood as the position of the camera with respect to a world coordinate system, are not considered in this work since they introduce no differences in the computed IPM images. Only the height of the camera with respect to the road plane is taken into account as a scale factor, i.e. it affects how much distance of road will be shown in the IPM images.

The vanishing point contains information that can be used to compute estimations of the pitch and yaw angles as it will be shown in the following paragraphs. The roll angle, as most works found in the literature, is not considered, as the vanishing point remains invariant with respect to the roll angle.

If considering both, the pitch angle of the camera inside the car, θ_c, and the variations of the slope of the road ahead, $\Delta\theta$, it is possible to obtain a corrected pitch angle, θ as $\theta = \theta_c + \Delta\theta$. Fig. 6 shows the effect of the road slope variations in the created IPM images if it is not performed any pitch angle correction, while Fig. 7 (a) shows the proposed model for pitch angle computation. Note that the vanishing point, as it will be shown in next paragraphs, is used to compute directly the θ parameter.

Therefore, from the estimation of the vanishing point $\overline{\mathbf{v}}_n = (\overline{y}, \overline{x})^T$ as described in Section IV-B, and through simple trigonometric computations derived from Fig. 7 it is possible to determine the relationship between $(\overline{y}, \overline{x})^T$ and (θ, γ), which is shown in equations (6) and (7).

$$\overline{y} = \frac{N}{2}\left(1 - \frac{\tan\theta}{\tan\alpha_v}\right) \tag{6}$$

$$\overline{x} = \frac{M}{2}\left(1 + \frac{\tan\gamma}{\tan\alpha_h}\right) \tag{7}$$

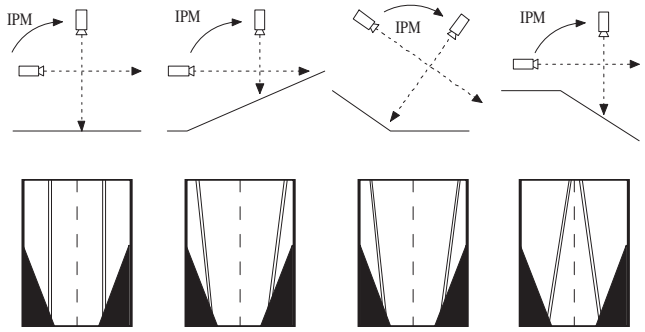


Fig. 6. Effect of the road slope variations in the computed IPM images without pitch angle correction

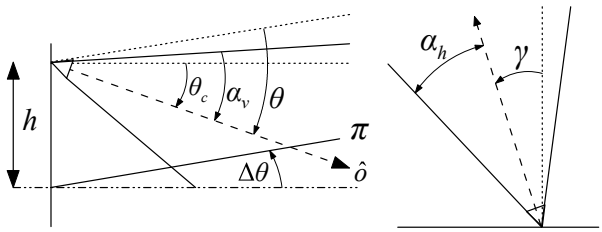


Fig. 7. Relation between road plane π and rotation angles: (a) Pitch angle θ composed by road plane slope $\Delta\theta$ and camera pitch angle θ_c; and (b) Yaw angle γ.

where α_v and α_h are, respectively, the vertical and horizontal angular aperture of the camera, while $M \times N$ are the dimensions of the image in pixels.

Equations (8) and (9) show the expressions to obtain the pitch and yaw angle from the vanishing point coordinates.

$$\theta = \arctan\left[\tan\alpha_v\left(1 - \frac{2\overline{y}}{N}\right)\right] \tag{8}$$

$$\gamma = \arctan\left[\tan\alpha_h\left(\frac{2\overline{x}}{M} - 1\right)\right] \tag{9}$$

B. Transform matrix composition

Once the pitch and yaw angle values are obtained from the estimation of the vanishing point, we can build stabilized IPM images by computing the transform matrix. This matrix describes how points of the real world $\mathbf{X} = (X, Y, Z)^{\mathrm{T}}$ are projected into points in the image, with coordinates, in pixels, $\mathbf{u} = (u, v)^{\mathrm{T}}$ [16].

Three different coordinate systems will be considered: i) the world coordinate system, assumed to be *on the road*, ii) the camera coordinate system, and iii) the image coordinate system. Fig. 8 depicts in detail the relationships between these coordinate systems. A point in the real world, for instance belonging to the road plane, may be expressed as $\mathbf{X} = (X, Y, Z)^{\mathrm{T}}$ with respect to the world coordinate system, or $\mathbf{x} = (x, y, z)^{\mathrm{T}}$ with respect to the camera coordinate system. The transform that links these expressions is shown in equation (10).

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R & -R\widetilde{\mathbf{C}} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{10}$$

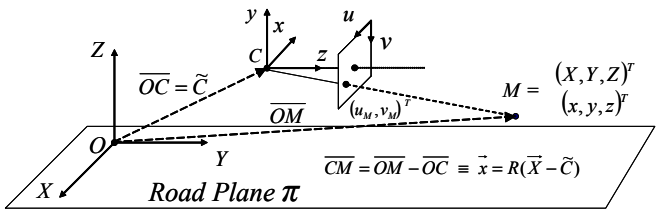


Fig. 8. World coordinate system, O; Camera coordinate system situated in $\widetilde{C}$ with respect to O, and the image coordinate system

where R is the rotation matrix composed by the angles θ and γ, and $\widetilde{C}$ is the translation vector between O and C [16]. The projection of the points in the camera coordinates into image coordinates is given by equation (11).

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} \frac{-M}{\tan\alpha_h} & 0 & u_0 & 0 \\ 0 & \frac{-N}{\tan\alpha_v} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{11}$$

where the symbol $\sim$ means equality except proportion. This latter matrix is denominated K, or the *camera calibration matrix*. The transform between image points and world coordinates is inmediately obtained by multiplying the expressions in (10) and (11), resulting in the equation (12).

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim KR \begin{bmatrix} I_3 & | & -\widetilde{\mathbf{C}} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{12}$$

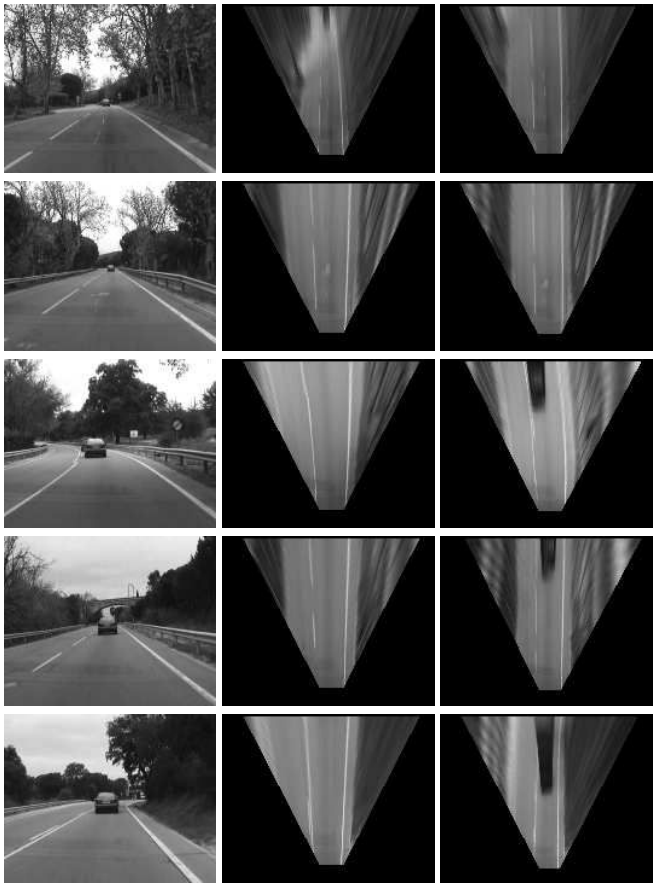
where I_3 is the 3×3 identity matrix. The result is a matrix $P = KR[I_3|-\widetilde{\mathbf{C}}]$ which is usually called *camera projection matrix* [16].

Therefore, points of the real world, e.g. belonging to the road, with world coordinates $\mathbf{X}$ are projected into the image at coordinates $\mathbf{u}$ through the projection matrix P as in (13).

$$\mathbf{u} \sim P\mathbf{X} \tag{13}$$

The IPM image, IPM_n, is generated using this matrix and the maximum distance of road to be displayed, Y_{max}. This value is transformed into image coordinates so that defining the region of the image to be retro-projected (only the lower part of the image is transformed). To perform the generation of the IPM images the following algorithm is proposed:

1) Compute the intersection points of the rays projected from C through the corners of the region of interest to be retro-projected and the road plane $Z = 0$.
2) Divide the road plane in squares so that each square correspond with one pixel at the IPM image.



(a) (b) (c)

Fig. 9. Test sequence in a local road, frames, from up to down, 64, 220, 708, 1478 and 2229: (a) Input images, I_n; (b) IPM images with fixed calibration; and (c) Stabilized IPM images using the proposed strategy

3) Apply equation (13) to the center of each square, with coordinates $\mathbf{X} = (X, Y, 0)^{\mathrm{T}}$ to obtain the pixel value with coordinates $\mathbf{u} = (u, v)^{\mathrm{T}}$ of the image to fill the corresponding pixel in the IPM image. Use bilinear interpolation to achieve sub-pixel accuracy.

This algorithm provides a smooth IPM image, with user defined image size in pixels, maximum distance of road to be displayed, Y_{max}, and stabilized (with constant width lanes) thanks to the described computation of the vanishing point.

VI. RESULTS

Tests have been made on images belonging to video sequences captured from cameras inside a car being driven along different types of road, including highways and local roads. The latter are characterized by important slope changes which are of special interest to show the adaptability of the proposed strategy to compute correct IPM images.

Video sequences are captured in uncompressed AVI-format, subsampled to 360×288 pixels per image, with 8-bits per pixels in raw format.

The proposed strategy has been efficiently implemented in a general purpose PC, working in real time for different frame rates according to the size of the images: 30 fps for 180×144 pixels, and 15 fps for 360×288.

Fig. 9 column (a) shows some example images belonging to sequences of a local road (El Pardo, Madrid); in column (b) the computed IPM images considering only an initial calibration of the camera kept fixed along the whole sequence in most of works found in the literature; and in column (c) the stabilized IPM images using the proposed strategy. These examples have been chosen to show that when occurring vanishing point changes, due to instability or road slope variations, the stabilized IPM show much better results than non-stabilized IPM, in terms of lane width constancy and parallelism between lane markings.

The first row show one particular example where the slope of the road ahead the vehicle is negative, so that the IPM image computed with the fixed calibration is incorrect, in the sense that lane markings seem to converge within the image. The stabilized IPM obtains much better results as it is adapted to this slope change, thus displaying a correct *bird-view* image.

Another significative example is shown in the third row, where the images show a curve in the road, with some positive slope variation, that makes the IPM image to fail, displaying divergent lane markings. Again, the stabilized IPM shows the correct *bird-view*, where the curvature of the road is clearly noticeable, so that further curvature detection algorithms could provide high accurate results compared with the results that would be obtained using unstabilized IPM images.

It is very significative than within the stabilized IPM images not only the lane width is almost perfectly constant, but also the distance to be shown is as large as desired. This effect is shown in Fig. 9 (c) where the vehicle appears on the stabilized IPM images in the three lower examples, corresponding to situations where the vehicle is close enough to the camera. As can be seen, in the non-stabilized IPM images, shown in (b), the transform is incorrect and the distance shown is lower, so that the vehicle does not appear on them. This is a significative advantage for further vehicle detection processing modules.

VII. CONCLUSIONS

In this paper an innovative IPM stabilization strategy based on robust estimations of the vanishing point has been presented. Each image of the video sequence is analyzed so as to obtain a segmentation which identify pixels belonging to the lane markings of the road. Straight lines are fitted to these pixels to obtain intersection points which are temporally filtered to estimate the position of the vanishing point for each image. This filter ensures robustness against outlier vanishing points, which may appear for punctual errors of the algorithm due to severe occlusions or sudden illumination changes.

The vanishing point jointly with a model of the road which considers slope variations, contains enough information to extract the rotation parameters of the camera required to stabilize the perspective transform.

Excellent results have been obtained, in terms of stability of IPM images, showing lanes with constant width along their path during the whole duration of the sequences (tested up to 3000 frames). In these images lane markings are parallel, and the image display road enough to detect vehicles at a distance, so that dramatically facilitating further processing modules, such as vehicle detection and curvature detection.

VIII. ANKNOWLEDGEMENTS

This work has been partially supported by the European Commission 6th Framework Program under project IST-2004-027195 (I-WAY). This work is also supported by the Comunidad de Madrid under project P-TIC-0223-0505 (PRO-MULTIDIS).

REFERENCES

[1] D. Schreiber, B. Alefs and M. Clabian, "Single camera lane detection and tracking", *in IEEE Proc. Intelligent Transportation Systems*, pp. 302-307, Sept. 2005.

[2] J.C. McCall and M.M. Trivedi, Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation, *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, No. 1, March 2006, pp. 20-37.

[3] Y. Wang, E. K. Teoh and D. Shen, Lane detection and tracking using B-snakes, *Image and Vision Computing*, vol. 22, pp. 269-289, 2004.

[4] Y.-M. Liang et al, Video Stabilization for a Camcorder Mounted on a Moving Vehicle, *IEEE Transactions on Vehicular Technology*, vol. 53, No. 6, November 2004.

[5] J. P. González, and U. Ozguner, "Lane detection using histogram-based segmentation and decision tree", *in IEEE Proc. Intelligent Transportation Systems*, 2000, pp. 346-351.

[6] M. Bertozzi and A. Broggi, GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection, *IEEE Transactions on Image Processing*, 1998, vol. 7, No. 1, pp. 62-81.

[7] C. Corridori and M. Zanin, "High Curvature two-clothoid road model estimation", *in IEEE Proc. Intelligent Transportation Systems*, Washington, D.C., USA, October 3-6, 2004, pp.630-636.

[8] M. Kotb and S. Beauchemin, "Generalizing inverse perspective", *in Proc. 2nd Canadian Conf. Computer and Robot Vision*, 2005, pp. 521-527.

[9] P. Cerri and P. Grisleri, "Free Space Detection on Highways using Time Correlation between Stabilized Sub-pixel precision IPM Images", *in IEEE Proc. Int. Conf. on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 2223-2228.

[10] G.Y. Jiang et al, "Lane and obstacle detection based on fast inverse perspective mapping algorithm", *IEEE Int. Conf. on Systems Man. and Cybernetics*, vol. 4, 2000.

[11] T. Bücher, "Measurement of Distance and Height in Images based on easy attainable Calibration Parameters", *in IEEE Proc. Intelligent Vehicles Symposium*, Dearborn(MI), USA, October 2000, pp. 314-319.

[12] M. Bertozzi, A. Broggi, and A. Fascioli, Stereo Inverse Perspective Mapping: Theory and Applications, *Image and Vision Computing Journal*, vol. 8, no. 16, pp. 585-590, 1998.

[13] A. Broggi, M. Bertozzi and A. Fascioli, "Self-calibration of a stereo vision system for automotive applications", *in IEEE Proc. Int. Conf. on Robotics and Automation*, vol. 4, pp. 3698-3703, 2001.

[14] K. Macek, B. Williams, S. Kolski and R. Siegwart, "A Lane Detection Vision Module for Driver Assistance", *in IEEE/APS Proc. Conference on Mechatronics and Robotics*, Germany 2004.

[15] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge Press, 1991.

[16] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*, Cambridge University Press, 2001.