Vision–based Road Detection via On–line Video Registration

Ferran Diego, José M. Álvarez, Joan Serrat and Antonio M. López

***Abstract*— Road segmentation is an essential functionality for supporting advanced driver assistance systems (ADAS) such as road following and vehicle and pedestrian detection. Significant efforts have been made in order to solve this task using vision-based techniques. The major challenge is to deal with lighting variations and the presence of objects on the road surface. In this paper, we propose a new road detection method to infer the areas of the image depicting road surfaces without performing any image segmentation. The idea is to previously segment manually or semi–automatically the road region in a traffic–free reference video record on a first drive. And then to transfer these regions to the frames of a second video sequence acquired later in a second drive through the same road, in an on–line manner. This is possible because we are able to automatically align the two videos in time and space, that is, to synchronize them and warp each frame of the first video to its corresponding frame in the second one. The geometric transform can thus transfer the road region to the present frame on–line. In order to reduce the different lighting conditions which are present in outdoor scenarios, our approach incorporates a shadowless feature space which represents an image in an illuminant–invariant feature space. Furthermore, we propose a dynamic background subtraction algorithm which removes the regions containing vehicles in the observed frames which are within the transferred road region.**

I. Introduction

Road detection, that is, determining the area of free road ahead, is a key component of several driving assistance modules like lane keeping and, vehicle and pedestrian detection. Beyond the evident interest of knowing the forward desirable area, determining the road regions in the images considerably reduces the search for such objects, thus reducing false detections and allowing real–time processing. In this paper we focus on the problem of road detection on images recorded by a single color camera. The main challenges that a road detection method has to face are the continuously changing background, the presence of different objects like vehicles and pedestrian, road types (urban, highways, back–road) and ambient illumination and weather conditions. Common vision–based algorithms for road detection adopt a bottom–up approach whereby low–level pixel or region properties such as color [1], [2] or texture [3], [4] are extracted and grouped according to some similarity measure. However, these algorithms based on low–level visual cues fail under wide lighting variations (camera over– and under–saturation due to large constrast changes) and depend on road structure, road surface homogeneity, simple road shapes, and mild lighting conditions. The performance of these systems is sometimes improved by including constraints such as temporal coherence [5], [6] or road shape restrictions [1] though they cannot solve the problem completely.

This work was supported by the Spanish Ministry of Education and Science under project TRA2007-62526/AUT and research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018) and the FPU MEC grant AP2007-01558.

The authors are with Computer Vision Center & Computer Science Dept., Edifici O, Universitat Autònoma de Barcelona, 08193 Cerdanyola del Vallés, Spain. ferran.diego@cvc.uab.es

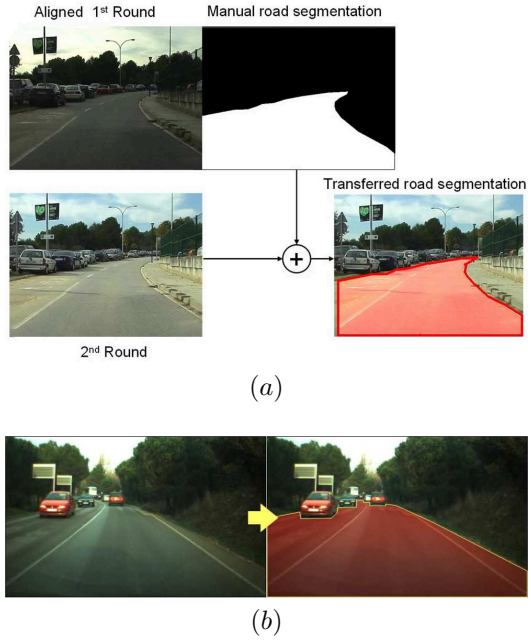

Fig. 1. Road detection algorithm: (a) shows the road detection via on–line video registration where the manually segmented road region is *transferred* from the reference sequence to the 'observed' sequence being acquired whereas (b) shows another examples of the meaning of road detection. Left: input image. Right: road detection results in red.

In this paper, we describe an original and completely different approach in that it does not perform any image segmentation for road detection. Therefore, it does not need to rely on low–level features like color, texture or intensity whose effectiveness and suitability varies with the ambient lighting and road type. The key idea is to previously record a video sequence along some road track and at each frame manually or semi–automatically segment the road region. On a second round, when the vehicle drives later along this same track, we record a second video sequence which we call the 'observed' sequence. Note that, of course, the

vehicle speed varies and this variation is independent in the two videos, so that at one same location the speed is different, in general. For each current frame of the observed sequence, we will be able to find out the corresponding frame in the reference sequence, that is, the one recorded at the same or the closest camera location in the first ride. In other words, we synchronize the two video sequences on–line, while recording the second one. Once found the corresponding reference frame, we compute the geometric transform which better warps it to the current 'observed' frame. Thus, the manually segmented road region can be *transferred* to it. Note that we perform a temporal and a spatial alignment, and the two must work even under different lighting conditions and slight relative camera translations and rotations, as shown in Fig. 1. As a final step, the regions containing vehicles in the observed frame which are within the transferred road region are removed. This is done by thresholding the difference of observed and warped reference frames, in a sort of dynamic background subtraction since the reference sequence is recorded with the absence of traffic. That is a plausible assumption in certain scenarios.

II. Road Segmentation Framework

The overall road segmentation framework consists of three stages (Fig. 2). Given an observed video sequence, in the first stage, we represent it in an illuminant–invariant feature space with the goal of reducing the influence of different lighting conditions. In the second stage, we synchronize two video sequences in order to find out the corresponding frame in the reference sequence to the current frame in the observed sequence. In order to compare properly the video sequences, all frames of the reference sequence are also represented in a shadowless feature space. Furthermore, we estimate the geometric transform in order to transfer the road segmentation of the reference frame to the current observed frame. Finally, in the third stage, we refine the road segmentation by removing objects which appear on the target frame at the place occupied by the transferred road region. These stages are described below.

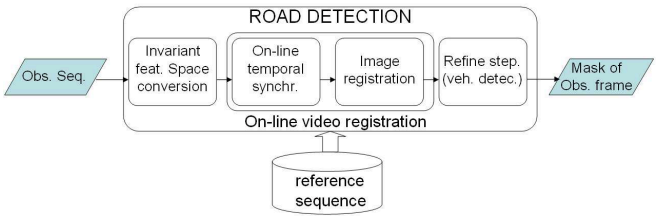

Fig. 2. Scheme of our road detection algorithm.

A. Invariant feature space conversion

The first stage minimizes the influence of lighting variations converting a image onto an illuminant–invariant feature space. Finlayson *et al.* have shown that if *Lambertian* surfaces are imaged by a three delta–function sensor under approximately Planckian light sources it is possible to generate an illuminant-invariant image $\mathfrak{I}$ [7]. Under these assumptions, a log-log plot of two dimensional $\{log(R/G), log(B/G)\}$ values for any surface forms a straight line provided camera sensors are fairly narrow-band. Thus, a lighting change is reduced to a linear transformation along an almost straight line (Fig. 3). This theory holds even for real data with only approximately Planckian lights, *non-Lambertian surfaces* and real cameras having only approximately narrow–band sensors.

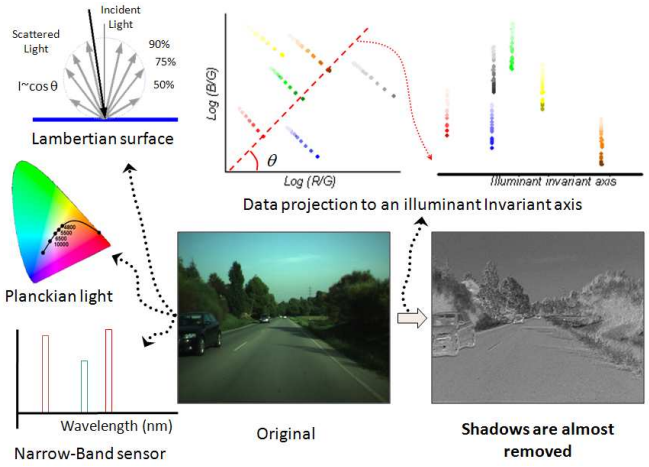

Fig. 3. An illuminant–invariant image is obtained under the assumptions of Planckian light, Lambertian surface and narrow-band sensors. This image is almost shadow free.

In short, $\mathfrak{I}$ is a gray-scale image that is obtained by projecting the $\{log(R/G), log(B/G)\}$ pixel values of the image onto the direction orthogonal to the lighting change lines, *invariant–direction* θ. This direction is device dependent and can be estimated off-line using the calibration procedure of [7].

B. On–line video registration

The aim of video registration is the alignment of two different video sequences in the temporal and spatial dimensions [8]. The video registration algorithm consists of two parts: temporal and spatial alignment. Temporal alignment or synchronization estimates a temporal mapping which relates the frames of the observed sequence to frames of the reference sequence, such that corresponding pairs of frames, show 'similar content'. Once the temporal mapping has been estimated, the corresponding frame in reference sequence can be warped to observed frame in order to compare them pixel–wise. However, registration of two video sequences recorded by an on–board camera from a moving vehicle is a challenging task. It must face (1) varying and independent speed of the cameras in the two sequences which implies a non–linear time correspondences and (2)

slight rotations and translations of the camera location due to dissimilar trajectories. Although several video registration techniques have been proposed [9], [10], [11], only our previous work [8] on video alignment addresses these two specific requirements. Now, we need to add a third important requirement: the temporal correspondence between the observed and the reference sequence must be computed on–line, because we need to obtain the road segmentation right after each frame has been acquired. Therefore, we propose a on–line video registration algorithm by extending [8]. The two parts of the video registration algorithm, temporal and spatial alignment, are described in the following two subsections.

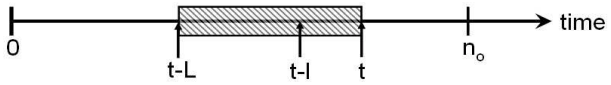

Fig. 4. Temporal meaning of a fixed lag–smoothing of a hidden Markov model where the label x_{t-l} is estimated at time t using L images in the observed sequence, which are from the $(t-L)^{th}$ to the t^{th} frame.

1) On–line synchronization: Let $\mathbf{S}^r$ and $\mathbf{S}^o$ be two video sequences n_r and n_o frames long, respectively. $\mathbf{S}^r$ denotes the reference sequence and $\mathbf{S}^o$ the observed video sequence, that we suppose to be contained entirely within $\mathbf{S}^r$. On–line synchronization consists in estimating a corresponding frame of the current frame of the observed sequence in the reference sequence. This task is formulated as a labelling problem. Each label $x_t \in [1 \dots n_r]$, refers to the frame number in the reference video corresponding to the t^{th} frame in the observed sequence. This task is posed as a maximum a posteriori inference problem of a fixed-lag smoothing hidden Markov model [12] which is defined as

$$x^*_{t-l} = \underset{x_{t-l} \in \Omega_t}{\operatorname{argmax}} p(x_{t-l}|\mathbf{y}_{t-L:t}) \tag{1}$$

where Ω_t is a set of possible labels at time $t-l$, $l \geq 0$ is a lag or delay, $L \geq l$ is the total set of observed frames used to infer the label x_{t-l} and $\mathbf{y}_{t-L:t}$ are the observations from the $(t-L)^{th}$ to the t^{th} in the observed sequence. Note that the estimation x^*_{t-l} requires L frames of the observed sequence. Fig. 4 illustrates the meaning of a fixed–lag smoothing. The range of labels of x_{t-l}, Ω_t, is $[x_{t-L-1}, x_{t-l-1} + \Delta]$, being Δ the maximum label difference between consecutive frames, and x_{t-L-1} and x_{t-l-1} the x^{th}_{t-L-1} and x^{th}_{t-l-1} estimated labels in the reference sequence, respectively. Fixed-lag smoothing infers the label x_{t-l} at time t with a delay of l frames. The max–product inference algorithm is applied in Eq. (1) in order to exactly infer x^*_{t-l} estimating $p(x_{t-l}|\mathbf{y}_{t-L:t})$ as

$$\begin{aligned} p(x_{t-l}|\mathbf{y}_{t-L:t}) &\propto \max_{\mathbf{x}_{t-L:t} \setminus x_{t-l}} p(\mathbf{x}_{t-L:t}|\mathbf{y}_{t-L:t}) \\ &\propto \max_{\mathbf{x}_{t-L:t} \setminus x_{t-l}} p(\mathbf{y}_{t-L:t}|\mathbf{x}_{t-L:t}) p(\mathbf{x}_{t-L:t}) \end{aligned} \tag{2}$$

where $\mathbf{x}_{t-L:t} = [x_{t-L}, \dots, x_t]$ is a list of labels which corresponds the temporal mapping between the reference and observed sequence and, the terms $p(\mathbf{x}_{t-L:t})$ and $p(\mathbf{y}_{t-L:t}|\mathbf{x}_{t-L:t})$ are the *a prior* and the an *observation likelihood*, respectively. The estimation of label x_{t-l} is the argument that maximizes the temporal coherence between two video sequences summarized as

$$x^{MAP}_{t-l} = \underset{x_{t-l} \in \Omega_t}{\operatorname{argmax}} \max_{\mathbf{x}_{t-L:t} \setminus x_{t-l}} p(\mathbf{y}_{t-L:t}|\mathbf{x}_{t-L:t}) p(\mathbf{x}_{t-L:t}) \ . \tag{3}$$

The prior $p(\mathbf{x}_{t-L:t})$ can be factorized as

$$p(\mathbf{x}_{t-L:t}) = P(x_{t-L}) \prod_{k=t-L}^{t-1} p(x_{k+1} \mid x_k), \tag{4}$$

under the assumption that the transition probabilities are conditionally independent given the label values. The intended meaning of $P(x_{k+1} \mid x_k)$ is the following: we assume that vehicles do not go backward, that they move always forward or at most stop for some time. Therefore, the labels x_t must increase monotonically. Hence, $P(x_{k+1} \mid x_k)$ is defined as

$$p(x_{k+1} \mid x_k) = \begin{cases} \beta & \text{if } x_{k+1} \geq x_k \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where β is a constant which gives the same importance at the labels satisfying the constraint of Eq. (5). The prior $P(x_{t-L})$ for the first label of the fixed–lag smoothing algorithm gives the same probability to all labels in $\Omega_t \in \{x_{t-L-1}, \dots, x_{t-l-1}, \dots, x_{t-l-1} + \Delta\}$.

The *observation likelihood*, $p(\mathbf{y}_{t-L:t}|\mathbf{x}_{t-L:t})$, measures the similarity of two video sequences given a temporal correspondence $\mathbf{x}_{t-L:t}$. The observation likelihood is assumed to be conditional independent given a label. Hence, it can be written as

$$p(\mathbf{y}_{t-L:t}|\mathbf{x}_{t-L:t}) = \prod_{k=t-L}^{t} p(y_k|x_k) \tag{6}$$

where $p(y_k|x_k)$ describes the similarity between one frame of the reference sequence and another of the observed sequence. We want this similarity to be maximum or at least high, if two frames are corresponding. Before computing this similarity, we compute a simply representation of the frames, which is called image descriptor $\mathbf{d}_*$. The image descriptor is computed from a smoothed image using a Gaussian kernel and downsampled to the $1/16^{th}$ of the original resolution. The image descriptor $\mathbf{d}$ of the illuminant invariant frame described in Sect. II-A is computed as follows. First, partial derivatives (i_x, i_y) are computed and the value at each pixel is set to zero if the gradient magnitude is less than 5% of the maximum. Finally, the descriptor is normalized to unit norm of the vector which are built concatenating the rows of i_x and i_y. We consider the inner product of two image descriptor

as a similarity measure because it measures the coincidence of the gradient orientation in the subsampled image. Hence, our observation probability is defined as

$$P(y_k|x_k) = \max_{\substack{-\Delta_x<i<\Delta_x \\ -\Delta_y<j<\Delta_y}} \mathcal{N}(< \mathbf{d}_{x_k}^{i,j}, \mathbf{d}_k >; 1, \sigma_s^2) \qquad (7)$$

where σ_s^2 controls the likelihood of the similarity measure between two frames. The $\mathbf{d}_{x_k}^{i,j}$ is the image descriptor of the x_t^{th} frame in reference sequence with a translation of i and j pixels over x– and y–directions respectively. The maximum translation of the smoothed downsampled image over x– and y–directions are Δ_x and Δ_y respectively, and set to 2 pixels. These translations are set to increase the robustness of the likelihood against slight camera rotations and translations due to trajectory dissimilarities. In our case, σ_s^2 is to set to 0.5 to give a significant likelihood only to those frames whose image descriptor form an angle less than approximately 5°.

2) Spatial Registration: On–line temporal alignment estimates a pair of corresponding frame numbers (t, x_t) each time a new frame, $\mathbf{S}_t^o$, is acquired. Ideally, for such a pair, the camera was at the same position and orientation. However, in practice, the camera pose may be different due to trajectory dissimilarities, independent accelerations and brakings, and road surface irregularities. Hence, the coordinates of two corresponding frames are related by a homography $H = KRK^{-1}$, where $K = \text{diag}(f, f, 1)$, being f the focal length of the camera in pixels. The rotation matrix R expresses the relative orientation of the camera for one pair of corresponding frames. This rotation matrix R can be parameterized by the Euler angles $\mathbf{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$ (pitch, yaw and roll respectively). We approximate the homography H by a quadratic motion vector field $\mathcal{W}$,

$$\mathcal{W}(\mathbf{x}; \mathbf{\Omega}) = \begin{bmatrix} -\frac{xy}{f} & f + \frac{x^2}{f} & -y \\ -f - \frac{y^2}{f} & \frac{xy}{f} & x \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}. \qquad (8)$$

This model is quadratic in the x and y coordinates but linear in $\mathbf{\Omega}$ and it holds under the assumption of having small angles and large focal length [13]. We estimate $\mathbf{\Omega}$ using the well-known additive forward implementation of the Lukas–Kanade algorithm [14] which minimizes the sum of squared differences between the target frame (observed frame) $\mathbf{S}_t^o$ and its corresponding transformed reference frame $\mathbf{S}_{x_t}^r$:

$$\mathbf{\Omega}^* = \underset{\mathbf{\Omega}}{\text{argmin}} \big(\sum_{\mathbf{x}} \big[\mathbf{S}_{x_t}^r(\mathbf{x} + \mathcal{W}(\mathbf{x}; \mathbf{\Omega})) - \mathbf{S}_t^o(\mathbf{x}) \big]^2 \big). \qquad (9)$$

In order to deal with large miss–alignments, Ω is iteratively estimated in a coarse–to–fine manner. Furthermore, $\mathbf{S}_{x_t}^r$ is repeatedly warped at each image pyramid level based on previous estimations of $\mathbf{\Omega}$ in order to deal with image non–linearities. For a detailed description we refer the reader to [14]. Let $\mathcal{M}_{x_t}$ and $\mathcal{M}_t$ be the road segmentation of the x_t^{th} frame in the reference sequence and the t^{th} frame in the observed sequence. Once $\mathbf{\Omega}$ is estimated, the road segmentation of the x_t^{th} frame in the reference sequence, $\mathcal{M}_{x_t}(\mathbf{x})$, is warped to the image coordinates of the t^{th} target frame in observed sequence as

$$\mathcal{M}_t = \mathcal{M}_{x_t}(\mathbf{x} + \mathcal{W}(\mathbf{x}; \mathbf{\Omega})) \ . \qquad (10)$$

C. Road transferred region refinement

The refinement algorithm of the road transferred region consists basically in removing regions which contains vehicles in the observed frame and are within the transferred road region. This is done by a dynamic background subtraction. We impose that the reference sequence is recorded in the absence of traffic. This constraint is plausible under certain conditions like back–road, urban scenarios at weekends and etcetera. The dynamic background subtraction proceed as follows. Once a pair of corresponding frames are spatially aligned, we subtract pixel–wise the intensity of such pair of corresponding frames as $\mathbf{S}_{x_t}^{r,w} - \mathbf{S}_t^o$, where w means that the frame $\mathbf{S}_{x_t}^r$ is warped to the image coordinates of $\mathbf{S}_t^o$. These subtraction allow us to spot differences of potential interest points which are the regions which contains the vehicles in the observed sequence. Then, we binarize the absolute value of the pixel–subtraction, $\mid \mathbf{S}_{x_t}^{r,w} - \mathbf{S}_t^o \mid$ by thresholding automatically using the Otsu's method [15]. Finally, a closing operator, which is a mathematical morphology, is applied to fill possible holes in the binary regions detected. Fig. 3 illustrates an example of refinement procedure in order to remove regions on the road surface which contain vehicles. Finally, the refinement algorithm consists directly in removing these regions, $\mathcal{O}_t$, from the transferred road segmentation, $\mathcal{M}_t$, according to a logical XOR operator between them. The logical XOR operator is only applied on the transferred road segmentation using the logical AND operator.

III. Results

In this section we present qualitative and quantitative results to validate our approach. The algorithm is tested on two different scenarios. The first one scenario is free of vehicles on both video sequences whereas the second deals with the presence of other vehicles in the observed sequence. The length of the observed and reference sequence in the first and second scenario is 520 and 627, and 1318 and 1459 frames long, respectively. The observed video sequence of each scenario was recorded in a sunny day under the influence of lighting variation while the reference video sequence was recorded in a cloudy day, and hence, it does not contains shadows. The difference in the number of frames is due to differences in the trajectory and the vehicle speed. All sequences were recorded at the same frame rate, 25fps, using a SONY DCR-PC330E camcorder. In addition, the average

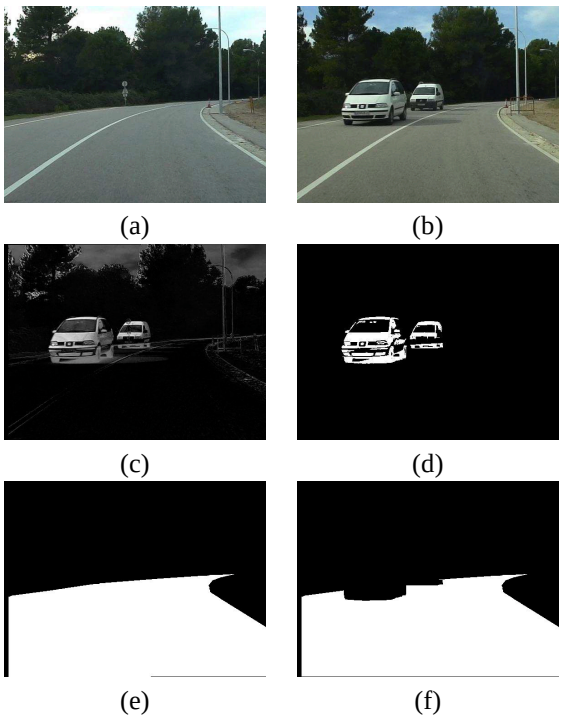

Fig. 5. Road surface refinement. The corresponding frame from the database (a) is aligned with the input frame (b). The difference between them (c) is used to detect foreground objects (d). The known road surface of the corresponding frame is transferred to the input image (e) in order to remove the foreground vehicles and finally obtain the output road segmentation (f).

speed of the vehicle is approximately 50 kph. Example results of our on–board road segmentation are shown in Fig. 6. More results, now in video format, can be viewed at *http://www.cvc.uab.es/∼fdiego/ITSC2010/RS.htm*. All these results suggest that road region is correctly transferred from the reference sequence to observed sequence. Furthermore, the refinement step handles correctly the presence of vehicles cropping properly them from the transferred road segmentation. As shown Fig. 6d, errors (red and green) are mainly located at the road and vehicle boundaries. However, a fraction of these errors is due to the manual road boundary delimitation of the reference sequence. Another source of errors is due to the inaccurate estimation of the motion field $\mathcal{W}$ due to synchronization errors.

Quantitative assessment is provided using four pixel–wise error measures are defined as:

$$\begin{array}{lcll} \text{Accuracy} & : & \text{ACC} & = \frac{TP+TN}{TP+FP+FN+TN} \\ \text{Sensitivity} & : & \text{TPR} & = \frac{TP}{TP+FN} \\ \text{Specificity} & : & \text{SPC} & = \frac{TN}{FP+TN} \\ \text{Quality} & : & \hat{g} & = \frac{TP}{TP+FP+FN} \end{array}$$

where TP is the number of correctly labelled road pixels, TN is the number of non-road pixels detected, FP is the number of non-road pixels classified as road pixels and FN is the number of road pixels erroneously marked as non-road. Each of these measures provides different insight of the results. Accuracy provides information about the fraction of classifications that are correct. Specificity measures the proportion of true negatives (i.e. background pixels) which are correctly identified as such. Sensitivity, or recall, is the ratio of detecting true positives. Quality is related to the completeness of the extracted data as well as its correctness. All these measures range from 0 to 1, from worst to perfect.

To properly assess the quality of the results, we annotated manually the road regions of the reference and observed sequence. Two different evaluations are possible depending on the output step of our on–board road detection, which are before and after the refinement step. This comparison allows to know the contribution of the refinement step. Furthermore, the evaluation is evaluated for each video sequence separately and the average of all sequences together. The averaged performance over all the frames is shown in Table. I. The contribution of the refinement step is negligible on Seq. 1 because the observed sequence is free of traffic. However, the refinement step is crucial on Seq. 2 to remove regions which contains vehicles even if Table. I shows a slight increase of performance of the four error measures. The regions that contain vehicles in Seq. 2 are basically moving vehicles and parked vehicles on the hard shoulder. The highest performance is achieved when the refinement step is used.

	Seq.	$\hat{g}$	SPC	TPR	ACC
No Ref. step	Seq. 1	0.972	0.992	0.983	0.988
	Seq. 2	0.955	0.984	0.974	0.980
	All	0.960	0.986	0.977	0.982
Ref. step	Seq. 1	0.966	0.987	0.986	0.986
	Seq. 2	0.961	0.985	0.980	0.982
	All	0.962	0.986	0.982	0.983

TABLE I

PERFORMANCE OF OUR ROAD SEGMENTATION ALGORITHM.

An inherent limitation of the method is the delay before obtaining a desired result which is less than 1 second, exactly is $800ms$. However, this is a minor limitation provided a high frame–rate camera. Once the l delay frames are acquired, our algorithm segments the road surface on real–time at the frame–rate of camcorders, which is normally 25 or 30 frames per second.

IV. CONCLUSIONS

In this paper we have proposed an approach for road detection based on an on–line video registration method. The key idea of the algorithm is to infer the areas of the image depicting road surfaces using road segmentations of previous video sequences. Furthermore, we include a pre– and post–processing stage in order to deal with different

Fig. 6. Example results. The frame from the reference sequence (a) is aligned with the input frame (c). The reference road region (b) is used to generate the output road segmentation (d). True positives in yellow, true negatives in black, false positives in read and false negatives in green, with respect to a road/non–road classification.

lighting conditions and the presence of vehicles. The pre–processing stage consists in converting each video sequence onto an illuminant–invariant feature space whereas the post–processing stage refines the road segmentation extracting the foreground objects. The algorithm has been successfully applied to detect the road surface under different lighting conditions and the presence of vehicles. Qualitatively and quantitatively evaluations prove that this method is quite accurate under different metrics.

REFERENCES

[1] M. Sotelo, F. Rodriguez, L. Magdalena, L. Bergasa, and L. Boquete, "A color vision-based lane tracking system for autonomous driving in unmarked roads," *Auton. Robots*, vol. 16, no. 1, 2004.

[2] C. Rotaru, T. Graf, , and J. Zhang, "Color image segmentation in hsi space for automotive applications," *Journal of Real-Time Image Processing*, pp. 1164–1173, 2008.

[3] P. Lombardi, M. Zanin, and S. Messelodi, "Switching models for vision-based on–board road detection," in *Procs. IEEE Intl. Conf. on Intel. Transp. Systems*, 2005.

[4] C. Rasmussen, "Grouping dominant orientations for ill-structured road following." in *CVPR (1)*, 2004, pp. 470–477.

[5] T. Michalke, R. Kastner, M. Herbert, J. Fritsch, and C. Goerick, "Adaptive multi-cue fusion for robust detection of unmarked inner-city streets," June 2009, pp. 1 –8.

[6] C. Tan, T. Hong, T. Chang, and M. Shneier, "Color model-based real-time learning for road following," *Procs. IEEE ITSC*, pp. 939–944, 2006.

[7] G. Finlayson, S. Hordley, C. Lu, and M. Drew, "On the removal of shadows from images," *IEEE Trans. on PAMI*, vol. 28, no. 1, 2006.

[8] F. Diego, D. Ponsa, J. Serrat, and A. López, "Video alignment for difference-spotting," in *ECCV 2008 Workshop on Multi Camera and Multi-modal Sensor Fusion Alg. and Apps.*, 2008. [Online]. Available: http://perception.inrialpes.fr/Publications/2008/ZCIBH08

[9] L. Wolf and A. Zomet, "Wide baseline matching between unsynchronized video sequences," *Int. Journal of Computer Vision*, vol. 68, no. 1, pp. 43–52, 2006.

[10] Y. Caspi and M. Irani, "Spatio–temporal alignment of sequences," *IEEE Trans. Pattern Analisys and Machine Intelligence*, vol. 24, no. 11, pp. 1409–1424, 2002.

[11] C. Lei and Y. Yang, "Trifocal tensor–based multiple video synchronization with subframe optimization," *IEEE Trans. Image Processing*, vol. 15, no. 9, pp. 2473–2480, 2006.

[12] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=773294

[13] J. Serrat, F. Diego, F. Lumbreras, and J. Alvarez, "Alignment of videos recorded from moving vehicles," in *CIAP07*, 2007, pp. 512–517.

[14] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[15] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, January 1979.