

Free Space Detection on Highways using Time Correlation between Stabilized Sub-pixel precision IPM Images

Pietro Cerri and Paolo Grisleri

*Artificial Vision and Intelligent System Laboratory
Dipartimento di Ingegneria dell'Informazione, Università di Parma
Viale delle Scienze 181/A, 43100 Parma, Italy
{cerri,grisleri}@ce.unipr.it*

Abstract—The work described in this paper has been developed in the framework of the Integrated Project APALACI - PRVENT, a European research activity funded by the European Commission to contribute to road safety by developing and demonstrating preventive safety applications and technologies.

This paper describes a new method for detecting free space in front of vehicles driving on highways. The described algorithm is mainly based on time correlation and Inverse Perspective Mapping (IPM). In order to improve the system robustness and reliability IPM has been modified to achieve sub-pixel precision and a new image stabilization method specifically designed to work on structured road is also proposed. The robustness of the method will be shown describing test results. Considerations on computational power requested by the algorithm are provided.

Index Terms—Free space detection, image correlation in time, inverse perspective mapping, sub-pixel accuracy, image stabilization.

I. INTRODUCTION

New automotive applications designed to improve driving safety and comfort, such as obstacle detectors and automatic cruise controls (ACC) need an accurate estimation of the free space in front of the vehicle.

The problem of finding and measuring the free space involves directly many fields connected to perception, especially when dealing with vision systems mounted on a vehicle moving on a real road.

Real world applications need to be robust enough to behave correctly even in critical situations; this typically happens when the information content is poor or noisy.

The luminance changes that can be detected passing under a bridge and the changes in camera orientation due to road bumps, holes and road coarseness belongs to this range of problems.

In this article, a new method for measuring the free space in front of the vehicle is presented. This method takes advantage of several techniques presented in the past, such as Inverse Perspective Mapping (IPM) [5] [6] and time correlation [4]. These techniques have been slightly modified in order to better adapt them to the problem details and have been mixed together, in order to obtain an integrated solution providing satisfactory results.

Another constraint taken into account has been the use of a low cost system, motivated by the consideration that automotive industries prefer to install a single camera

instead of more expensive sensors such as radars, lidars or stereo cameras. A laser range finders could be used as well, but vision system are cheaper and suitable also for other applications such as obstacle classification.

The problem that has been considered is focused to highway environments. This reduces the number of variables involved and allows to make strong assumptions such as flat road and existence of structures (lanes) that can be extracted from images and used for measurement.

Thus the system consists of a single camera mounted on a vehicle. Images are taken from this camera at fixed time intervals and processed on a standard PC mounted inside the vehicle.

Measurements of distances are obtained using the camera but they are affected by camera orientation oscillation which are due to the vehicle movements. Stabilization integrated on camera has a lot of problem: electronic stabilization does not work with low frequency oscillations, mechanical one does not work with high frequency oscillations, thus a low cost camera is used and a stabilization algorithm is developed.

The main idea of the algorithm is to find the best pitch angle estimation choosing it in a precomputed list. This approach is suggested by the fact that in normal driving conditions the camera pitch angle changes at each frame, remaining in a fixed range. This range is determined by many factors such as road coarseness, wind, holes, acceleration and so on.

In the following section the state of the art about free space detection and stabilization will be presented; in section III the algorithm is described in details, subsection III-A describes an extension of the IPM computation using sub-pixel accuracy; subsection III-B describes a stabilization system suitable for highway environments and subsection III-C concludes the algorithm description with the illustration of how to compute the free space in front of the vehicle using temporal correlation. Finally section IV shows some experimental results of the algorithm behavior and lists some considerations about the algorithm strength, weakness and future developments.

II. STABILIZATION AND FREE SPACE DETECTION.

Several approaches, available in the literature deal with stabilization. In [1] the image velocities for each small block of pixels, representing features, are extracted

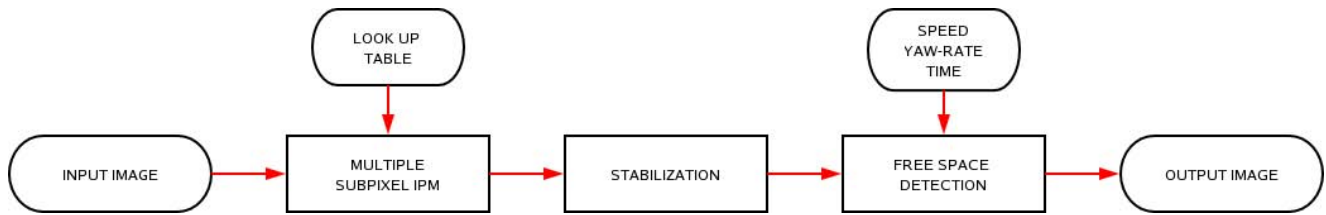


Fig. 1
ALGORITHM FLOW-CHART

by building coarse-to-fine pyramids for two consecutive frames and evaluating their movement using correlation. Then a 2.5-D inter-frame motion model is used to estimate camera movements.

In [2], far features are extracted in a given frame. These features are matched with those extracted from previous frames, estimating the camera motion. Then camera motion is corrected using rotations and translations, thus 2-D model is used to correct the image from the camera motion information. This works well only if the framed objects are concentrated around a fixed distance: corrections for pixels representing far away areas are not suitable for pixels representing near areas.

In [3], after detecting lanes, the vanishing point is computed, and camera motion parameters are estimated for each frame. The advantage of this approach is the avoidance of feature extraction and matching in favor of a robust lane detection. Some problem in the estimation of the vanishing point position may occur when the vehicle is proceeding along a curve.

All the cited works operate on small images (up to 320x240) with displacements up to ± 64 pixels. The stabilized image is usually a sub-window of the acquired image in order to avoid the introduction artifacts in the output. When the stabilization algorithm intervention takes place, the missing information, necessary to compensate camera motion in the stabilized image, is taken from the original image which is larger than the output. Unfortunately this method wastes part of the information supplied from the sensor.

On the other hand this paper contributes to the stabilization techniques panorama with a new method, specific for IPM images. The advantages of this method is that the whole original image is used. In fact the correction is not applied on the image itself but on the camera orientation parameters.

In this camera motion estimation scheme, like in most of the cited works, estimated parameters are filtered in order to obtain a smooth stabilization.

This paper also proposes a method for detecting free space in front of the vehicle using a single stream of frames. In the work proposed in [4] the difference between two consecutive frames acquired from the same camera was used to track multiple moving vehicles. From the difference image it is also possible to detect standing objects. Typical shapes of detected moving and standing object are shown

in figure 8.

The time difference between acquired frames was known, and camera movements between one frame and the subsequent were indirectly estimated from images.

The presented implementation is based on direct measurements of camera speed and yaw rate coming from sensors commonly installed on vehicles. This enforces the robustness of the system providing a better camera motion detection.

III. ALGORITHM

This section describes the structure of the developed algorithm which is depicted in figure 1. The first step is the application of all possible look up table (LUT) computed for each value of the camera pitch angle (θ) to the input image in order to obtain one IPM image for each possible value of θ . The correction of pitch angle has been chosen because is more critical for this kind of system; roll and yaw angles could be adjusted as well, but more computational power and memory would be needed.

IPM images are computed using sub-pixel accuracy. The second step is the selection of the IPM image having a constant road width.

This selection leads to a specific value of θ that is used as an estimation for the pitch angle in a given instant. The only assumption is the parallelism of road markings which is not a strong constraint for the applicability of the system.

The selected IPM image is used as input for the last algorithm step that is the computation of the free space.

The output of the algorithm is a binarized image where black points indicate free space and white points indicate generic obstacles. This image can then be used for successive high level measurements.

Next subsections describe in details the 3 main steps of the algorithm: sub-pixel precision IPM computation, stabilization, and the free space detection procedure.

A. IPM with sub-pixel accuracy

This subsection describes the procedure to compute IPM images from the input image.

Usually IPM computation is a CPU intensive process that can be speeded-up using a LUT. If the input image has resolution $I_w \times I_h$ and the size of the IPM image is $IPM_w \times IPM_h$, then the size of the LUT will be $IPM_w \times IPM_h$; each of its elements contains a pair of numbers x', y' where:

$$(x', y') \in \{(UNDEF, UNDEF), ([0, I_w], [0, I_w])\}$$

and *UNDEF* indicates a special undefined value.

When the size of the IPM image is fixed the LUT size is fixed as well and two effects happens at the same time:

- the input image area representing the road close to the vehicle is sub-sampled;
- input pixels representing far away space are mapped in more than one point in IPM image (oversampling).

This leads to two problems: nearest details are damaged by sub-sampling and artifacts are introduced for far points.

The new proposed LUT element contains weights to be applied to the mapped point and to its 8 neighbors. This weights are double precision floating point numbers and are computed in the following way: Let us define d_{xu} , d_{yu} as the number of the input image pixels that are associated with one pixel of the IPM image (under-sampling factor); these values grows as the distance reduces; let x' , y' be the exact coordinate of the input image that maps the pixel in the IPM image (double precision floating point), x_{ij} , y_{ij} as the center of the pixel i , j of the 3×3 mask applied to the input image (integer value), considered i , j

$$\begin{cases} i \in [-1, 1], & i \in \mathbb{N} \\ j \in [-1, 1], & j \in \mathbb{N} \end{cases}$$

it is possible to compute the distance between each pixel of the mask and x' y' (see figure2):

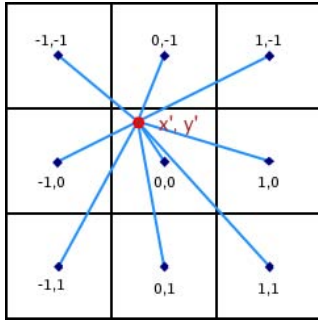


Fig. 2

SUB-PIXEL ACCURACY IPM DISTANCES COMPUTATION.

$$d_{ij}(x, y) = \sqrt{(x' - x_{ij})^2 + (y' - y_{ij})^2}$$

and the weight of each pixel of the mask as:

$$c'_{ij} = e^{-\frac{d_{ij}^2}{var(x,y)}}$$

where $var(x, y)$ is defined as:

$$var(x, y) = \left(\min(d_{xu}, d_{yu}) + \left| \frac{d_{xu} - d_{yu}}{2} \right| \right)^{-1} \quad (1)$$

These values have to be normalized as:

$$c_{ij} = \frac{c'_{ij}}{\sum_{k=-1}^1 \sum_{l=-1}^1 c'_{kl}}$$

The IPM transformation uses as parameters the camera orientation θ , γ , ρ and the camera position x , y , z .

All possible LUTs using different θ between θ_1 and θ_2 with step $\delta\theta$ are computed. The association between the

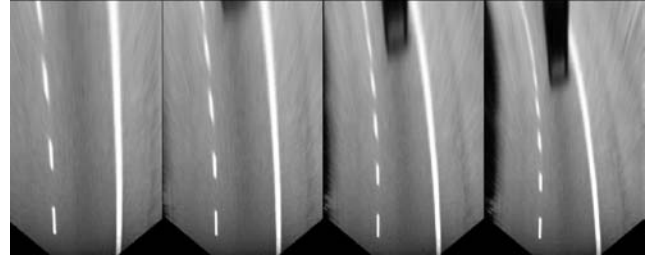


Fig. 3

IPM IMAGES WITH DIFFERENT θ MAKE LANE MARKINGS APPEAR AS CONVERGENT OR DIVERGENT. THE CORRECT VALUE OF THETA GENERATES AN IPM IMAGE IN WHICH LANE MARKINGS ARE PARALLEL (AS EXPLAINED IN SECT. III-B)

LUT and the θ value is maintained for future use. The LUT creation is a CPU intensive operation: it requires 1.4 seconds for on a 3GHz Pentium4 class machine for 400×600 elements size. However this is a one shot operation performed only once at the procedure initialization.

The LUT application to input image requires about 200ms. This calculation is performed N times for each frame where N is the number of LUT used. The more LUT are used, the more accurate is the stabilization output but a higher time is required for the processing of each frame.

As shown in (1) the variance of the weights grows with the distance in the following way: nearest points are computed using a small component of their neighbors, while far away points are computed with a sensible contribution coming from their neighbors.

This procedure leads to an IPM image more suitable for our purpose (figure 4). The main advantage is that artifacts in the upper portion of the image (the one representing far distances) disappear. In this way, the information related to one pixel of the original image, that in the classic IPM image is constant and distributed over a wide range of pixels, in this image changes smoothly and is provided not only by the re-mapped pixel, but also by his first neighbors. Another advantage is that in the lower part of the image, more details are maintained because of the intrinsic low pass filtering which introduces automatically anti-aliasing.

In the next section a stabilization algorithm that uses these images as input is described.

B. Image stabilization

Once the IPM has been applied, it is necessary to analyze each IPM image in order to find out the one that contains pair of lane markings with a constant relative distance throughout the whole image. The result of this search leads to the estimated pitch value.

The search is performed following this procedure. A rectangular area in the central lower part of the image is scanned in order to compute vertical histograms. The centers of mass of the values around the two highest peaks of this histogram are selected as the starting points for the left and the right lane markings.

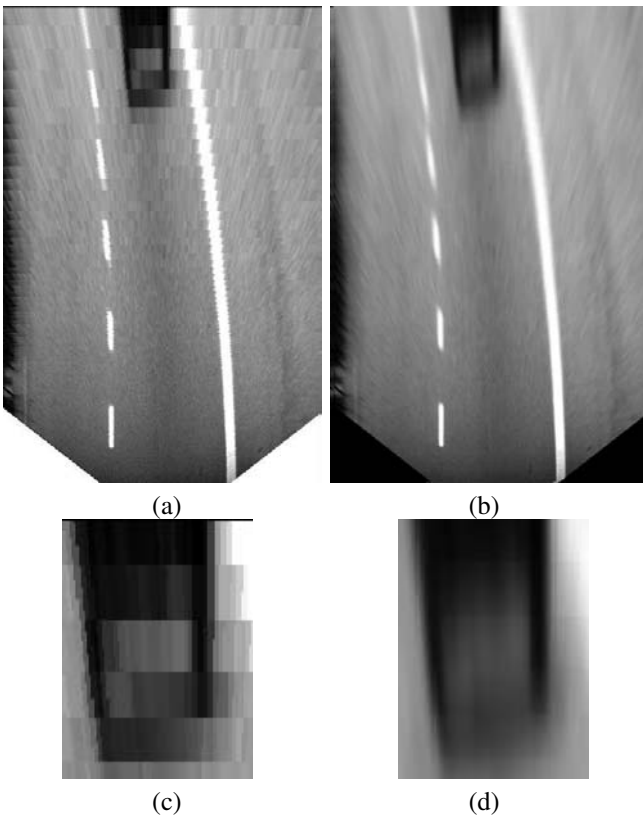


Fig. 4

(a) CLASSIC IPM, (b) IPM WITH SUB-PIXEL ACCURACY, ENLARGEMENTS SHOWING: (c) ARTIFACT FOR FAR DISTANCES IN CLASSIC IPM, (d) RESULT FOR MODIFIED IPM.

Once the starting point for the lane has been computed, a number of areas (depicted in figure 5) are scanned with the same algorithm in order to extract samples of the position of the lane. The height of these areas is constant because in the IPM image each pixels, regardless of its position in the image, represents patch of road with a given size. The areas width grows with distance in order to allow detection of curves and is centered in the position of the starting point found during the scanning of the lower part of the image.

This method has two main advantages; the first one is the reduction of computational time. In fact since the application of the modified LUT is a CPU intensive process, it is useful to reduce as much as possible the number of points to which this transformation is applied.

Using the area optimization the computation time for each LUT application drops down to 80ms for 10 areas like shown in figure 5.

The second one is the robustness to luminance variations. For example when the shadow of a bridge is projected on the road the following two situations may happens:

- the search area is either completely shadowed or completely out of shadow.
- the bridge shadow border cuts the search area

Since each segment has its own luminance threshold, in the first case correct measurements takes place while in the

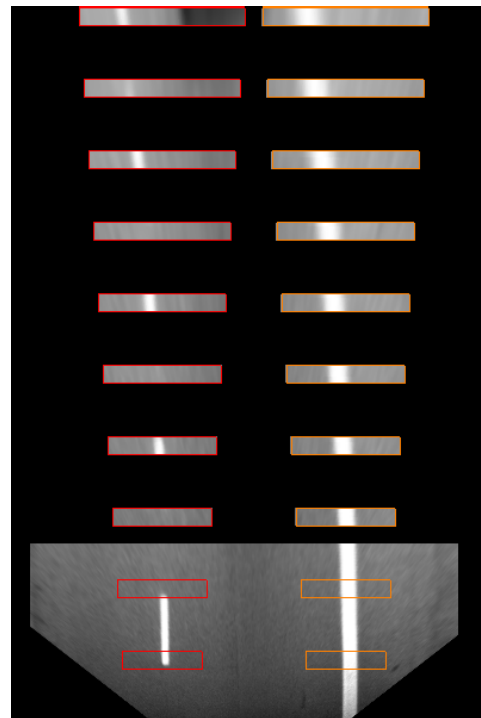


Fig. 5

RESEARCH AREAS

second case the measurement may be lost. This lack can be easily recovered with interpolation as it will be explained later.

This phase produces as output two arrays containing the coordinates, in the IPM image, of the left and right lane centers. For each line, a middle point is computed which represents the road center for a particular pair of left and right lane samples.

The array of centers may have some spurious values. These wrong estimation are eliminated interpolating a sufficient number of values according to the starting values (see figure 7).

If there are not enough values for interpolation, the stabilization cannot work for this IPM image. The road width is computed for each pair of samples in the following way. Two consecutive centers are connected by a segment and the line n orthogonal to this segment is computed. Let l and r be the lines passing through the two consecutive left and right road limits respectively. The distance between the intersections of n with l and r is the road width for that segment (as shown in figures 6 and 7.a).

At this point, for each IPM image an array of road widths has been computed. Each array is analyzed and the angular coefficient of the trend line is extracted.

The lower is the coefficient, the more correct is the IPM because the road is supposed to have a constant width as shown in figure 3.

Selecting the lower angular coefficient between the IPM image leads to select the best estimation of the pitch.

Unfortunately sometimes the lane cannot be correctly

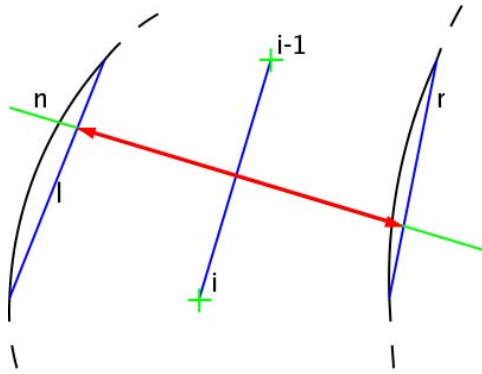


Fig. 6

ROAD WIDTH COMPUTATION

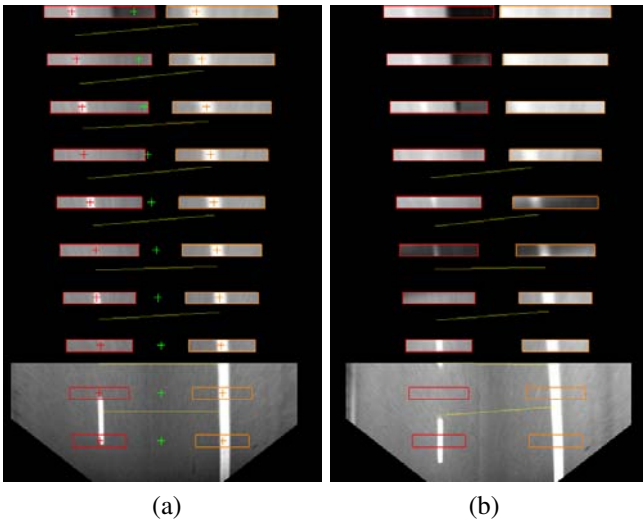


Fig. 7

ROAD WIDTH COMPUTATION (a) WITH ALL MEASUREMENT, (b) WITH A FEW MISSING MEASUREMENT.

detected in the upper region of the image, so only the lower points will be used to compute the angular coefficient (figure 7.b). In order to avoid errors due to a too small number of points considered, the algorithm takes care of the number of points looking for the image that has both maximum number of available points and minimum angular coefficient.

The curvature of the road can also be computed using the position of centers; its evaluation is anyway out of the purpose of this work.

The estimated value of θ is then smoothed with the previous history values in order to reduce too abrupt variations.

C. Free Space detection

Once θ has been estimated, the whole IPM transformation is applied using that value and this image is used as input for the free space detection algorithm.

The image is translated and rotated using timestamp, speed and vehicle yaw rate information. The system has been implemented on one of the prototype vehicles avail-

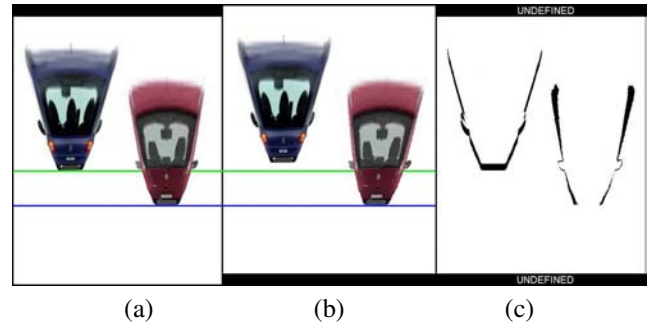


Fig. 8

A SYNTHETIC IMAGE SHOWING (a) PREVIOUS IPM FRAME AT TIME t , (b) CURRENT IPM FRAME AT TIME $t + dt$. (c) DIFFERENCE IMAGE SHOWING A TYPICAL DISTORTED SHAPE OF OBSTACLES: OBSTACLES EDGES ARE VISIBLE, WHILE THE BOTTOM OF EACH DISTORTED FIGURE SHOWS THE RELATIVE SPEED OF THE OBSTACLE. IN THIS CASE THE BLUE VEHICLE IS MOVING AND THE RED ONE IS STILL ON THE EDGE OF THE ROAD.

able at the Artificial Vision And Intelligent System Laboratory in Parma. The vehicle is equipped with a Pentium IV class PC that can grab video from a camera and data such as speed, and yaw rate through a normal CAN bus. Data and images get synchronized by the PC that also processes this information stream extracting the free space as output.

The free space is obtained as follows: two frames are acquired in different instants: t_0 and $t_0 + dt$. At time t the camera is moving with speed \vec{v} with a big component along the x axis and rotates with angular speed $\vec{\omega}$ (yaw). These two motions must be compensated, for the current frame. The two images are subtracted in order to obtain a difference image that is shown in figure 8.

The difference image is undefined in the upper and in the bottom areas because the new image contains information not present in the previous one. The central area is the most interesting part. If something not belonging to the road surface, such as a moving or a still obstacle, is framed in this area, then the two IPMs taken at different times are different. The difference image is relevant in the area surrounding the obstacle and maybe thresholded showing typical shapes, different for still obstacles and moving ones. The shapes are shown in figure 8.

The flowchart of the free space detection algorithm used can be found in figure 9. Once the difference has been computed, a threshold is applied in order to obtain the output image.

The output of this process is a binarized version of the IPM image with white spots in correspondance to obstacles and black spots in correspondance to free space.

IV. CONCLUSIONS AND RESULTS

This paper described a method for detecting the free space in front of a moving vehicle using the temporal correlation between images taken in different time instants by a single camera.

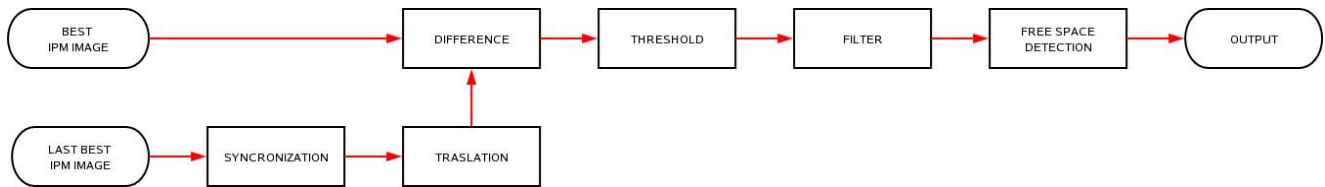


Fig. 9

FREE SPACE DETECTION FLOW-CHART



(a)



(b)



(c)

Fig. 10

(a) ORIGINAL IMAGE, (b) STABILIZED IPM IMAGE, (c) FREESPACE DETECTION.

This work is part of a larger project for driving assistance under development at the Artificial Vision and Intelligent Systems Laboratory, Department of Information Technology of the University of Parma. The testing vehicle is equipped with a daylight, low cost, VGA resolution, analog camera. Video and data are acquired and processed by a 2.8 GHz Pentium IV class PC through an analog frame-grabber based on BT878 chip and a CAN board to access sensors installed on the vehicle.

This article contributes to the integration of different techniques, with the aim of obtaining more robust results.

The algorithm has been tested with 20 LUTs computed for a range of $\theta \in [-0.1, +0.1]rad$ with $\delta\theta = 0.01$ and 20 sample area for each image. Under these conditions the generation of all LUT requires 30.475s at the startup and the processing for a single frame needs 950ms to be executed. It is anyway important to note that the images considered here are high resolution (640x480).

The proposed stabilization system allows to use the whole image acquired from the sensor, extracting the best estimated pitch angle from a set of possible values.

The system has been tested on real world sequences and an example of result is depicted in figure 10.

Stabilization method based on the original image feature matching works in a limited distance range and waste part of the input image. This approach act on the IPM image modifying the pitch angle thus uses all the pixels of the original image.

These are the possible development for the work.

- Actually the biggest algorithm limitation in real application is due to the computational power requested to perform the stabilization process. However a strong code optimization is possible since the slow-level part relies on the multiple LUT application for stabilization. This code could be speeded up using specific SIMD hardware.
- Since the vehicle movement takes normally place along the driving axis, the system tries to stabilize images against pitch oscillations. An extension can be realized correcting roll variations. These oscillations are due to road coarseness and are equally critical, for calibration maintenance, with respect the pitch oscillations.
- Although designed for flat roads, the algorithm should have remarkable performance also on non-flat roads. Tests will be performed soon.

If the camera is moving, the free space detection algorithm is able to detect both fixed and moving objects. If the vehicle is not moving the free space algorithm can only detect moving objects. These features make the algorithm suitable for real application.

REFERENCES

- [1] S. Jesse Jin, Zhigang Zhu, Guangyou Xu, "A stable vision system for moving vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 1, pp. 32-39, March 2000.
- [2] Morimoto C., Chellappa R., "Fast electronic digital image stabilization, ", Proceedings of the 13th International Conference on Pattern Recognition, vol. 3., pp.284-288, 25-29 August 1996
- [3] Yu-Ming Liang, Hsiao-Rong Tyan, Hong-Yuan Mark Liao, Sei-Wang Chen, "Stabilizing image sequences taken by the camcorder mounted on a moving vehicle," Proceedings of IEEE Intelligent Transportation Systems 2003, vol.1., pp. 90-95, 12-15 October 2003
- [4] Hu Z., Uchimura K., "Tracking cycle: a new concept for simultaneous tracking of multiple moving objects in a typical traffic scene," Proceedings of the IEEE Intelligent Vehicles Symposium 2000., pp. 233-239, 3-5 October 2000
- [5] Tsai R., "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," IEEE Journal of Robotics and Automation, vol. 3, no. 4, pp. 323-344, Aug 1987
- [6] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli, "Stereo Inverse Perspective Mapping": Theory and Applications, Image and Vision Computing Journal, 16(8):585-590, June 1998.