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

3D Visual Odometry for GPS Navigation Assistance

R. G. García-García, M. A. Sotelo, I. Parra, D. Fernández, M. Gavilán

Abstract— **This paper describes a method for estimating the vehicle global position in a network of roads by means of visual odometry. To do so, the ego-motion of the vehicle relative to the road is computed using a stereo-vision system mounted next to the rear view mirror. Feature points are matched between pairs of frames and linked into 3D trajectories. The resolution of the equations of the system at each frame is carried out under the non-linear, photogrametric aproach using RANSAC. This iterative technique enables the formulation of a robust method that can ignore large numbers of outliers as encountered in real traffic scenes. The resulting method is defined as visual odometry and can be used in conjunction with other sensors, such as GPS, to produce accurate estimates of the vehicle global position. The obvious application of the method is to provide on-board driver assistance in navigation tasks, or to provide a means for autonomously navigating a vehicle. The method has been tested in real traffic conditions without using prior knowledge about the scene nor the vehicle motion. We provide examples of estimated vehicle trajectories using the proposed method and discuss the key issues for further improvement.**

I. INTRODUCTION

The use of video sensors for vehicle navigation has become a research goal in the field of Intelligent Transportation Systems and Intelligent Vehicles in the last years. Accurate estimation of the vehicle global position is a key issue, not only for developing useful driver assistance systems, but also for achieving autonomous driving. Using stereo-vision for computing the position of obstacles or estimating road lane markers is a usual technique in intelligent vehicle applications. The challenge now is to extend stereo-vision capabilities to also provide accurate estimation of the vehicle ego-motion with regard to the road, and thus to compute the vehicle global position. This is becoming more and more tractable to implement on standard PC-based systems nowadays. However, there are still open issues that constitute a challenge in achieving highly robust ego-motion estimation in real traffic conditions. These are discussed in the following lines.

1) There must exist stationary reference objects that can be seen from the cameras position. Besides, the reference objects must have clearly distinguishable features that make possible to unambiguously perform matching between two frames. Accordingly, the selection of features becomes a critical issue.
2) Information contained on road scenes can be divided into road feature points and background feature points. On the one hand, roads have very few feature points, most of then corresponding to lane markings, or even no points in the case of unmarked roads. On the other hand, information corresponding to the background of road scenes may contain too many feature points. Robust matching techniques are then needed to avoid false matching.
3) Typical road scenes may contain a large amount of outlier information. This includes non-stationary objects such as moving vehicles and pedestrians, car wipers. All these artifacts contribute to false measurements for ego-motion estimation. Possible solutions to overcome this problem are two fold: to deploy some outlier rejection strategy; to estimate feature points motion using probabilistic models in order to compensate for it in the estimation process.

The authors are with the Department of Electronics. Escuela Politécnica Superior. University of Alcalá. Alcalá de Henares, Madrid, Spain sotelo,parra,llorca@depeca.uah.es

In this paper, we propose a method for ego-motion computing based on stereo-vision. The use of stereo-vision has the advantage of disambiguating the 3D position of detected features in the scene at a given frame. Based on that, feature points are matched between pairs of frames and linked into 3D trajectories. The idea of estimating displacements from two 3-D frames using stereo vision has been previously used in [1] [2] and [3]. The resolution of the equations of the system at each frame is carried out under the non-linear, photogrametric aproach using RANSAC. This iterative technique enables the formulation of a robust method that can ignore large numbers of outliers as encountered in real traffic scenes. The resulting method is defined as visual odometry and can be used in conjunction with other sensors, such as GPS, to produce accurate estimates of the vehicle global position. The obvious application of the method is to provide on-board driver assistance in navigation tasks, or to provide a means for autonomously navigating a vehicle. The method has been tested in real traffic conditions without using prior knowledge about the scene nor the vehicle motion. We provide examples of estimated vehicle trajectories using the proposed method and discuss the key issues for further improvement.

The rest of the paper is organized as follows: in section II the feature detection and matching technique is presented; section III provides a description of the proposed non-linear method for estimating vehicle ego-motion and the 3D vehicle trajectory; implementation and results are provided in section IV; finally, section V is devoted to conclusions and discussion about how to improve the current system performance in the future.

II. FEATURES DETECTION AND MATCHING

In each frame, Harris corners [4] are detected, since this type of point feature has been found to yield detections that

are relatively stable under small to moderate image distortions [5]. As stated in [2], distortions between consecutive frames can be regarded as fairly small when using video input [2]. The feature points are matched at each frame, using the left and rights image of the stereo-vision arrangement, and between pairs of frames. Features are detected in all frames and matches are allowed only between features. A feature in one image is matched to every feature within a fixed distance from it in the next frame, called disparity limit. For the sake of real-time performance, matching is computed over a 7x7 window.

Among the wide spectrum of matching techniques that can be used to solve the correspondence problem we implemented the *Zero Mean Normalized Cross Correlation* [6] because of its robustness. The Normalized Cross Correlation between two image windows can be computed as follows.

$$\mathrm{ZMNCC}(p,p') = \frac{\sum_{i=-n}^{n}\sum_{j=-n}^{n} A \cdot B}{\sqrt{\sum_{i=-n}^{n}\sum_{j=-n}^{n} A^2 \sum_{i=-n}^{n}\sum_{j=-n}^{n} B^2}} \quad (1)$$

where A and B are defined by

$$A = \left(I\left(x+i, y+j\right) - \overline{I(x,y)}\right) \quad (2)$$

$$B = \left(I'\left(x'+i, y'+j\right) - \overline{I'(x',y')}\right) \quad (3)$$

where $I(x,y)$ is the intensity level of pixel with coordinates (x,y), and $\overline{I(x,y)}$ is the average intensity of a $(2n+1)$x$(2n+1)$ window centered around that point. As the window size decreases, the discriminatory power of the area-based criterion gets decreased and some local maxima appear in the searching regions. On the contrary, an increase in the window size causes the performance to degrade due to occlusion regions and smoothing of disparity values across boundaries. In consequence, the correspondences yield some outliers. According to the previous statements, a filtering criteria is needed in order to provide outliers rejection. The accepted matches are used both in 3D feature detection (based on stereo images) and in feature tracking (between consecutive frames). Figure 1 depicts an example of features detection and tracking using Harris detector, ZMNCC matching technique, and mutual consistency check.

III. Visual odometry using non-linear estimation

The problem of estimating the trajectory followed by a moving vehicle can be defined as that of determining at frame i the rotation matrix $R_{i-1,i}$ and the translational vector $T_{i-1,i}$ that characterize the relative vehicle movement between two consecutive frames. The use of non-linear methods becomes necessary since the 9 elements of the rotation matrix can not be considered individually (the rotation matrix has to be orthonormal). Indeed, there are only 3 unconstrained, independent parameters, i.e., the three rotation angles θ_x, θ_y and θ_z, respectively. The system's rotation can be expressed by means of the rotation matrix R given by equation 4.

$$R = \begin{pmatrix} cycz & sxsycz + cxsz & -cxsycz + sxsz \\ -cysz & -sxsysz + cxcz & cxsysz + sxcz \\ sy & -sxcy & cxcy \end{pmatrix} \quad (4)$$

where $ci = cos\theta_i$ and $si = sin\theta_i$ for $i = x, y, z$. The estimation of the rotation angles must be undertaken by using an iterative, least squares-based algorithm [7] that yields the solution of the non-linear equations system that must compulsorily be solved in this motion estimation application. Otherwise, the linear approach can lead to a non-realistic solution where the rotation matrix is not orthonormal.

A. Non-linear least squares

Given a system of n non-linear equations containing p variables:

$$\begin{cases} f_1(x_1, x_2, \ldots, x_p) = b_1 \\ f_2(x_1, x_2, \ldots, x_p) = b_2 \\ \vdots \\ f_n(x_1, x_2, \ldots, x_p) = b_n \end{cases} \quad (5)$$

where f_i, for $i = 1, \ldots, n$, is a differentiable function from $\Re^p$ to $\Re$. In general, it can be stated that:

1) if $n < p$, the system solution is a $(p-n)$-dimensional subspace of $\Re^p$.
2) if $n = p$, there exists a finite set of solutions.
3) si $n > p$, there exists no solution.

As can be observed, there are several differences with regard to the linear case: the solution for $n < p$ does not form a vectorial subspace in general. Its structure depends on the nature of the f_i functions. For $n = p$ a finite set of solutions exists instead of a unique solution as in the linear case. To solve this problem, an underdetermined system is built ($n > p$) in which the error function $E(x)$ must be minimized.

$$E(\mathbf{x}) \triangleq \sum_{i=1}^{N} (f_i(\mathbf{x}) - b_i)^2 \quad (6)$$

The error function $E : \Re^p \rightarrow \Re$ can exhibit several local minima, although in general there is a single global minimum. Unfortunetaly, there is no numerical method that can assure the obtaining of such global minimum, except for the case of polynomial functions. Iterative methods based on the gradient descent can find a global minimum whenever the starting point meets certain conditions. By using non-linear least squares the process is in reality linearized following the tangent linearization approach. Formally, function $f_i(x)$ can be approximated using the first term of Taylor's series expansion, as given by equation 7.

$$\begin{aligned} f_i(\mathbf{x} + \delta\mathbf{x}) = f_i(\mathbf{x}) + \delta x_1 \frac{\partial f_i}{\partial x_1}(\mathbf{x}) + \ldots + \\ + \delta x_p \frac{\partial f_i}{\partial x_p}(\mathbf{x}) + O(|\delta\mathbf{x}|)^2 \approx f_i(\mathbf{x}) + \nabla f_i(\mathbf{x}) \cdot \delta\mathbf{x} \end{aligned} \quad (7)$$

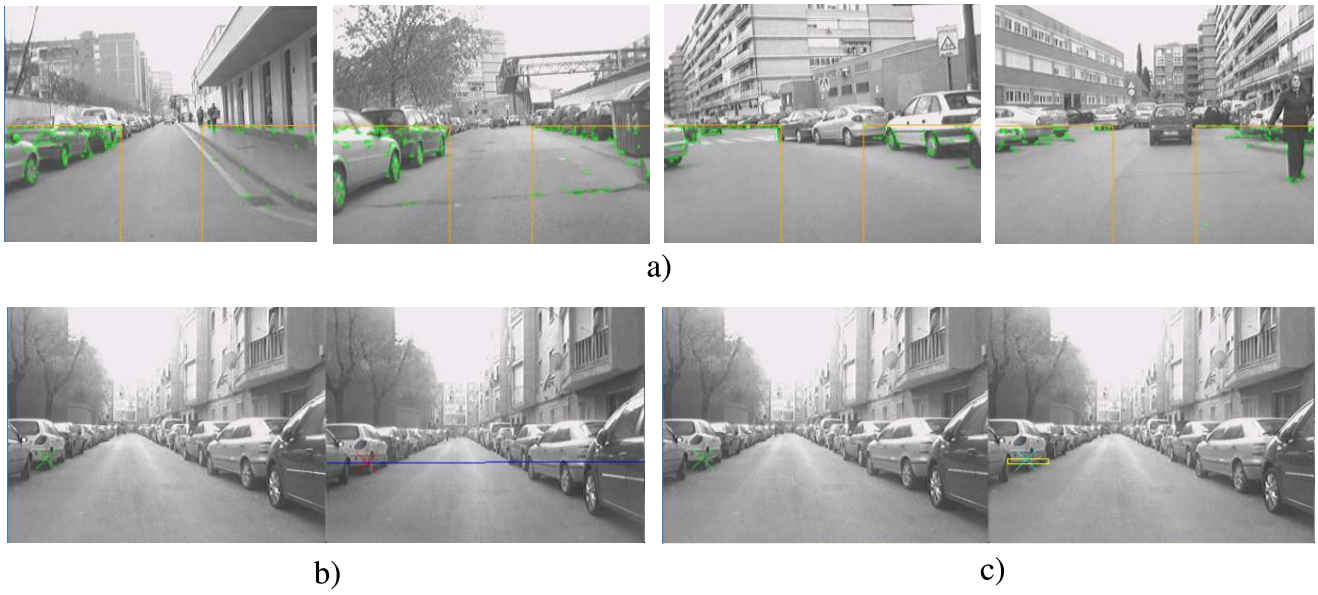


Fig. 1. a) The upper row depicts feature detection results using Harris detector in several images in urban environments. Detection is constrained to a couple of regions of interest located in the lateral areas of the image bellow the horizon line. b) The bottom left image shows an example of features matching in a stereo image. c) The bottom right image depicts an example of feature tracking in two consecutive frames. ZMNCC and mutual consistency check is used both for feature detection and feature tracking.

where $\nabla f_i(\mathbf{x}) = (\frac{\partial f_i}{\partial x_1}, \ldots, \frac{\partial f_i}{\partial x_p})^t$ is the gradient of f_i calculated at point $\mathbf{x}$, neglecting high order terms $O(|\delta\mathbf{x}|)^2$. The error function $E(\mathbf{x}+\delta\mathbf{x})$ is minimized with regard to $\delta\mathbf{x}$ given a value of $\mathbf{x}$, by means of a iterative process. Substituting (7) in (5) yields:

$$\begin{array}{c} E(\mathbf{x}+\delta\mathbf{x}) = \sum_{i=1}^{N}(f_i(\mathbf{x}+\delta\mathbf{x}) - b_i)^2 \approx \\ \approx \sum_{i=1}^{N}(f_i(\mathbf{x}) + \nabla f_i(\mathbf{x})\cdot\delta\mathbf{x} - b_i)^2 = |\mathbf{J}\delta\mathbf{x} - \mathbf{C}|^2 \end{array} \quad (8)$$

where

$$\mathbf{J} = \begin{pmatrix} \nabla f_1(\mathbf{x})^t \\ \ldots \\ \nabla f_n(\mathbf{x})^t \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \ldots & \frac{\partial f_1}{\partial x_p}(\mathbf{x}) \\ \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \ldots & \frac{\partial f_n}{\partial x_p}(\mathbf{x}) \end{pmatrix} \quad (9)$$

and

$$\mathbf{C} = \begin{pmatrix} b_1 \\ \ldots \\ b_n \end{pmatrix} - \begin{pmatrix} f_1(\mathbf{x}) \\ \ldots \\ f_n(\mathbf{x}) \end{pmatrix} \quad (10)$$

After linearization, an overdetermined linear system of n equations and p variables has been constructed ($n < p$):

$$\mathbf{J}\delta\mathbf{x} = \mathbf{C}, \quad (11)$$

System given by equation 11 can be solved using least squares, yielding:

$$\delta\mathbf{x} = (\mathbf{J}^t\mathbf{J})^{-1}\mathbf{J}^t\mathbf{C} = \mathbf{J}^\dagger\mathbf{C}. \quad (12)$$

In practice, the system is solved in an iterative process, as described in the following lines:

1) An initial solution $\mathbf{x_0}$ is chosen
2) While ($E(\mathbf{x_i}) > e_{min}$ and $i < i_{max}$)
   - $\delta\mathbf{x_i} = \mathbf{J}(\mathbf{x_i})^\dagger\mathbf{C}(\mathbf{x_i})$
   - $\mathbf{x_{i+1}} = \mathbf{x_i} + \delta\mathbf{x_i}$
   - $E(\mathbf{x_{i+1}}) = E(\mathbf{x_i} + \delta\mathbf{x_i}) = |\mathbf{J}(\mathbf{x_i})\delta\mathbf{x_i} - \mathbf{C}(\mathbf{x_i})|^2$

where the termination condition is given by a minimum value of error or a maximum number of iterations.

B. 3D Trajectory estimation

Between instants t_0 and t_1 we have:

$$\begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} = R_{0,1}\begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + T_{0,1}; \qquad i = 1, \ldots, N \quad (13)$$

Considering (4) it yields a linear six-equations system at point i, with 6 variables $\mathbf{w} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z]^t$:

$$\begin{cases} {}^1x_i = cycz\cdot{}^0x_i + (sxsycz + cxsz)\cdot{}^0y_i + \\ +(-cxsycz + sxsz)\cdot{}^0z_i + t_x \\ {}^1y_i = -cysz\cdot{}^0x_i + (-sxsysz + cxcz)\cdot{}^0y_i + \\ +(cxsysz + sxcz)\cdot{}^0z_i + t_y \\ {}^1z_i = sy\cdot{}^0x_i - sxcy\cdot{}^0y_i + cxcy\cdot{}^0z_i + t_z \end{cases}$$

At each iteration k of the regression method the following linear equations system is solved (given the 3D coordinates of N points in two consecutive frames):

$$\mathbf{J}(\omega)\delta\mathbf{x_k} = \mathbf{C}(\mathbf{x_k}) \quad (14)$$

with:

$$\mathbf{J}(\omega) = \begin{pmatrix} J_{1,11} & J_{1,12} & J_{1,13} & J_{1,14} & J_{1,15} & J_{1,16} \\ J_{1,21} & J_{1,22} & J_{1,23} & J_{1,24} & J_{1,25} & J_{1,26} \\ J_{1,31} & J_{1,32} & J_{1,33} & J_{1,34} & J_{1,35} & J_{1,36} \\ J_{2,11} & J_{2,12} & J_{2,13} & J_{2,14} & J_{2,15} & J_{2,16} \\ J_{2,21} & J_{2,22} & J_{2,23} & J_{2,24} & J_{2,25} & J_{2,26} \\ J_{2,31} & J_{2,32} & J_{2,33} & J_{2,34} & J_{2,35} & J_{2,36} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ J_{N,11} & J_{N,12} & J_{N,13} & J_{N,14} & J_{N,15} & J_{N,16} \\ J_{N,21} & J_{N,22} & J_{N,23} & J_{N,24} & J_{N,25} & J_{N,26} \\ J_{N,31} & J_{N,32} & J_{N,33} & J_{N,34} & J_{N,35} & J_{N,36} \end{pmatrix}$$

$$\delta\mathbf{x_k} = [\delta\theta_{x,k}, \delta\theta_{y,k}, \delta\theta_{z,k}, \delta t_{x,k}, \delta t_{y,k}, \delta t_{z,k}]^t$$

$$\mathbf{C}(\mathbf{x_k}) = [c_{1,1}, c_{1,2}, c_{1,3}, \ldots, c_{N,1}, c_{N,2}, c_{N,3}]^t$$

Let us remark that the first index of each Jacobian matrix element represents the point with regard to whom the function is derived, while the other two indexes represent the position in the 3x6 sub-matrix associated to such point. Considering (9) the elements of the Jacobian Matrix that form sub-matrix $\mathbf{J_i}$ for point i at iteration k are:

$J_{i,11} = (cx_k sy_k cz_k - sx_k sz_k) \cdot {}^0y_i + (sx_k sy_k cz_k + cx_k sz_k) \cdot {}^0z_i$
$J_{i,12} = -sy_k cz_k \cdot {}^0x_i + sx_k cy_k cz_k \cdot {}^0y_i - cx_k cy_k cz_k \cdot {}^0z_i$
$J_{i,13} = -cy_k sz_k \cdot {}^0x_i + (-sx_k sy_k sz_k + cx_k cz_k) \cdot {}^0y_i + (cx_k sy_k sz_k + sx_k cz_k) \cdot {}^0z_i$
$J_{i,14} = 1$
$J_{i,15} = 0$
$J_{i,16} = 0$
$J_{i,21} = -(cx_k sy_k sz_k + sx_k cz_k) \cdot {}^0y_i + (-sx_k sy_k sz_k + cx_k cz_k) \cdot {}^0z_i$
$J_{i,22} = sy_k sz_k \cdot {}^0x_i - sx_k cy_k sz_k \cdot {}^0y_i + cx_k cy_k sz_k \cdot {}^0z_i$
$J_{i,23} = -cy_k cz_k \cdot {}^0x_i - (sx_k sy_k x_k z_k + cx_k sz_k) \cdot {}^0y_i + (cx_k sy_k cz_k - sx_k sz_k) \cdot {}^0z_i$
$J_{i,24} = 0$
$J_{i,25} = 1$
$J_{i,26} = 0$
$J_{i,31} = -cx_k cy_k \cdot {}^0y_i - sx_k cy_k \cdot {}^0z_i$
$J_{i,32} = cy_k \cdot {}^0x_i + sx_k sy_k \cdot {}^0y_i - cx_k sy_k \cdot {}^0z_i$
$J_{i,33} = 0$
$J_{i,34} = 0$
$J_{i,35} = 0$
$J_{i,36} = 1$

After computing the Jacobian matrix the iterative process is implemented as described in the previous section.

C. RANSAC

RANSAC (RAndom SAmple Consensus) [8] [9] is an alternative to modifying the generative model to have heavier tails to search the collection of data points S for good points that reject points containing large errors, namely "outliers". The algorithm can be summarized in the following steps:

1) Draw a sample s of n points from the data S uniformly and at random.
2) Fit to that set of n points.
3) Determine the subset of points S_i for whom the distance to the model s is bellow the threshold t. Subset S_i (defined as consensus subset) defines the inliers of S.
4) If the size of subset S_i is larger than threshold T the model is estimated again using all points belonging to S_i. The algorithm ends at this point.
5) Otherwise, if the size of subset S_i is below T, a new random sample is selected and steps 2, 3, and 4 are repeated.
6) After N iterations (maximum number of trials), draw subset S_{ic} yielding the largest consensus (greatest number of "inliers"). The model is finally estimated using all points beloging to S_{ic}.

RANSAC is used in this work to estimate the Rotation Matrix R and the translational vector T that characterize the relative movement of a vehicle between two consecutive frames. The input data to the algorithm are the 3D coordinates of the selected points at times t and $t+1$. Notation t_0 and $t_1 = t_0 + 1$ is used to define the previous and current frames, respectively, as in the next equation.

$$\begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} = R_{0,1} \begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + T_{0,1}; \qquad i = 1, \ldots, n \tag{15}$$

After drawing samples from three points, in step 1 models $\widetilde{R}_{0,1}$ and $\widetilde{T}_{0,1}$ that best fit to the input data are estimated using non-linear least squares.Then, a distance function is defined to classify the rest of points as inliers or outliers depending on threshold t.

$$\begin{cases} \text{inlier} & e < t \\ \text{outlier} & e \geqslant t \end{cases} \tag{16}$$

In this case, the distance function is the square error between the sample and the predicted model. The 3D coordinates of the selected point at time t_1 according to the predicted model are computed as:

$$\begin{pmatrix} {}^1\widetilde{x}_i \\ {}^1\widetilde{y}_i \\ {}^1\widetilde{z}_i \end{pmatrix} = \widetilde{R}_{0,1} \begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + \widetilde{T}_{0,1}; \qquad i = 1, \ldots, n \tag{17}$$

The error vector is computed as the difference between the estimated vector and the original vector cointaining the 3D coordinates of the selected points (input to the algorithm):

$$\mathbf{e} = \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} = \begin{pmatrix} {}^1\widetilde{x}_i \\ {}^1\widetilde{y}_i \\ {}^1\widetilde{z}_i \end{pmatrix} - \begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} \tag{18}$$

The mean square error or distance function for sample i is given by:

$$e = |\mathbf{e}|^2 = \mathbf{e}^t \cdot \mathbf{e} \tag{19}$$

In the following subsections, justification is provided for the choice of the different parameters used by the robust estimator.

1) Distance threshold t: According to this threshold samples are classified as "inliers" or "outliers". Prior knowledge about the probability density function of the distance between "inliers" and model d_t^2 is required. If measurement noise can be modelled as a zero-mean Gaussian function with standard deviation σ, d_t^2 can then be modelled as a chi-square distribution. In spite of that, distance threshold is empirically chosen in most practical applications. In this work, a threshold of $t = 0.005$ was chosen.

2) Number of iterations N: Normally, it is inviable or unnecessary to test all the possible combinations. In reality, a sufficiently large value of N is selected in order to assure that at least one of the randomly selected s samples is oulier-free with a probability p. Let ω be the probability of any sample to be an inlier. Consequently, $\epsilon = 1 - \omega$ represents the probability of any sample to be an outlier. At least, N samples of s points are required to assure that $(1 - omega^s)^N = 1 - p$. Solving for N yields:

$$N = \frac{log(1-p)}{log(1-(1-\epsilon)^s)} \tag{20}$$

In this case, using samples of 3 points, assuming $p = 0.99$ and a proportion of outliers $\epsilon = 0.25$ (25%), at least 9 iterations are needed. In practice, the final selected value is $N = 10$.

3) Consensus threshold T: The iterative algorithm ends whenever the size of the consensus set (composed of inliers) is larger than the number of expected inliers T given by ϵ and n:

$$T = (1 - \epsilon)n \tag{21}$$

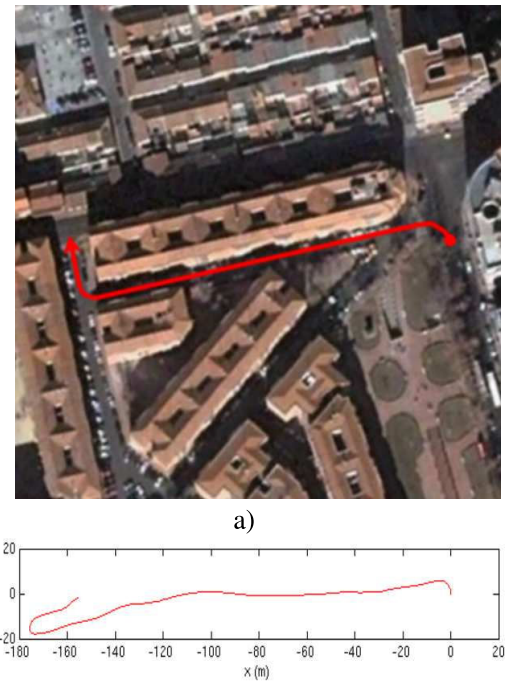
4) Data Post-processing: This is the last stage of the algorithm. Some partial estimations are discarded, in an attempt to remove as many outliers as possible, using the following criteria.

1) High root mean square error e estimations are removed.
2) Meaningless rotation angles estimations (non physically feasible) are discarded.

Accordingly, a maximum value of e has been set to 0.5. Similarly, a maximum rotation angle threshold is used to discard meaningless rotation estimations. In such cases, the estimated vehicle motion is supposed to be $R = I$ y $T = 0$. Removing false rotation estimations is a key aspect in visual odometry systems since false rotation estimations lead to high cumulative errors.

IV. Implementation and Results

The visual odometry system described in this paper has been implemented on a Pentium IV at 1.7 GHz running Linux Knoppix 3.7 with a 2.4.18-6mdf kernel version. The algorithm is programmed in C using OpenCV libraries (version 0.9.7).A stereo vision platform based on Fire-i cameras (IEEE1394) was installed on a prototype vehicle. After calibrating the stereo vision system, several sequences were recorded in different locations including Alcalá de Henares and Arganda del Rey in Madrid (Spain). The stereo sequences were recorded using no compression algorithm at 30 frames/s with a resolution of $320x240$ pixels. All sequences correspond to real traffic conditions in urban environments. The results of a first experiment are depicted in figure 2. The vehicle starts a trajectory in which it first turns slightly to the left. Then, the vehicle runs along a straight street and, finally, it turns right at a strong curve with some 90 degrees of yaw change. The upper part of figure 2 shows an aerial view of the area of the city (Alcalá de Henares) were the experiment was conducted (source: http://maps.google.com). The bottom part of the figure illustrates the 2D trajectory estimated by the visual odometry algorithm presented in this paper.



a)

b)

Fig. 2. a) Aerial view of the area of the city were the experiment was conducted. b) Estimated trajectory using visual odometry.

As can be observed, the system provides reliable estimations of the path run by the vehicle in almost straight sections. As a matter of fact, the estimated lenght of the straight section in figure 2.b is very similar to the ground truth (some 175m). The estimated vehicle trajectoy along the straight street is almost straight, similar to the real trajectory described by the vehicle in the experiment. Nonetheless, there are still some problems to estimate accurate rotation angles in sharp bends (90 degreess or more). Rotation angles estimated by the system at strong curves tend to be higher

than the real rotation experimented by the vehicle. This problem does not arise in the first left curve conducted by the vehicle, where the estimated rotation and the real rotation are very similar, as can be observed in figure 2.

In A second experiment, the car started turning slight right and then left to run along an almost straight path for a while. After that, a sharp right turn is executed. Then the vehicle moves straight for some metres until the end of the street. Figure 3 illustrates the real trajectory described by the vehicle (a) and the estimated trajectory estimated by the visual odometry algorithm (b). In this case, the estimated trajectory reflects quite well the exact shape and length of the real trajectory executed by the vehicle.

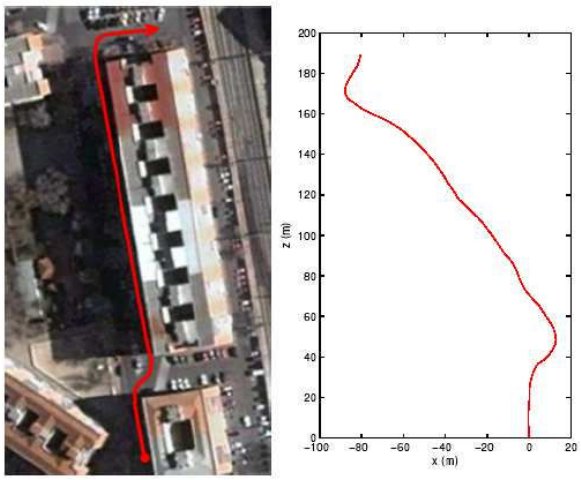


Fig. 3. a) Aerial view of the area of the city were the experiment was conducted. b) Estimated trajectory using visual odometry.

V. Conclusions and Future Work

We have described a method for estimating the vehicle global position in a network of roads by means of visual odometry. To do so, the ego-motion of the vehicle relative to the road is computed using a stereo-vision system mounted next to the rear view mirror of the car. Feature points are matched between pairs of frames and linked into 3D trajectories. The resolution of the equations of the system at each frame is carried out under the non-linear, photogrametric aproach using least squares and RANSAC. This iterative technique enables the formulation of a robust method that can ignore large numbers of outliers as encountered in real traffic scenes. Fine grain outliers rejection methods have been experimented based on the root mean square error of the estimation and the vehicle dynamics. The resulting method is defined as visual odometry and can be used in conjunction with other sensors, such as GPS, to produce accurate estimates of the vehicle global position.

A key aspect of the system is the features selection and tracking stage. For that purpose, a set of 20 points has been extracted using Harris detector. The searching windows have been optimized in order to achieve a trade-off between robustness and execution time. Real experiments have been conducted in urban environments in real traffic conditions with no a priori knowledge of the vehicle movement or the environment structure. We provide examples of estimated vehicle trajectories using the proposed method. Although preliminary, first results are encouraging since it has been demonstrated that the system is capable of providing approximate vehicle motion estimation in non strongly bended trajectories. Nonetheless, further improvements need to be accomplished in order to accurately cope with 90 degrees curves, which are very usual in urban environments.

As part of our future work we envision to develop a method for discriminating stationary points from those which are moving in the scene. Moving points can correspond to pedestrians or other vehicle circulating in the same area. Vehicle motion estimation will mainly rely on stationary points. The system can benefit from other vision-based applications currently under development and refinement in our lab, such as pedestrian detection and ACC (based on vehicle detection). The output of these systems can guide the search for really stationary points in the 3D scene. The obvious application of the method is to provide on-board driver assistance in navigation tasks, or to provide a means for autonomously navigating a vehicle. For this purpose, fusion of GPS and vision data will be accomplished.

Acknowledgements

This work has been supported by the Spanish Ministry of Education and Science by means of Research Grant DPI2005-07980-C03-02.

References

[1] Z. Zhang and O. D. Faugeras, "Estimation of displacements from two 3-d frames obtained from stereo," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 14, No. 12, December, 1992.

[2] D. Nister, O. Narodistsky, and J. Beren, "Visual odometry," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. June, 2004.

[3] A. Hagnelius, "Visual odometry," in *Masters Thesis in Computing Science*. Umea University, April, 2005.

[4] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conference*. pp. 147-151, 1988.

[5] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," in *International Journal of Computer Vision*. Vol. 37, No. 2, pp. 151-172, 2000.

[6] B. Boufama, "Reconstruction tridimensionnelle en vision par ordinateur: Cas des cameras non etalonnees," in *PhD thesis*. INP de Grenoble, France, 1994.

[7] D. A. Forsyth and J. Ponce, *Computer Vision. A Modern Approach*, international ed. Pearson Education International. Prentice Hall, 2003.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," in *Communications of the ACM*. June, 1981.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.