

2D Visual Odometry method for Global Positioning Measurement

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

Department of Electronics. Escuela Politécnica Superior. University of Alcalá. Alcalá de Henares, Madrid, Spain.

Contact Author E-mail: miguel.sotelo@uah.es

Abstract – The goal of this paper is to develop a method for estimating the 2D trajectory of a road vehicle using 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 computed using Harris detector. After that, features are matched between pairs of frames and linked into 2D trajectories. A photogrametric approach is proposed to solve the non-linear equations using a least-squared approximation. The purpose is to merge trajectory information provided by the visual odometry system with information provided by other sensors, such as GPS, in order to produce really accurate measurements of vehicle position. Providing assistance to drivers is among the prime applications of the proposed method. Nonetheless, other applications such as autonomous robot or vehicle navigation are also considered. The proposed 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.

Keywords – 2D visual odometry, egomotion estimation, global position measurement, RANSAC, non-linear least squares.

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 them 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.

In this paper, we propose a method for ego-motion computing based on a non-linear, photogrametric approach using RANSAC and 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] [3] and [4]. A common

factor of these works is the use of robust estimation and outliers rejection using RANSAC. In [2] a so-called firewall mechanism is implemented in order to reset the system to remove cumulative error. Both monocular and stereo-based versions of visual odometry were developed in [2], although the monocular version needs additional improvements to run in real time, and the stereo version is limited to a frame rate of 13 images per second. In [5] a stereo system composed of two wide Field of View cameras was installed on a mobile robot together with a GPS receiver and classical encoders. The system was tested in outdoor scenarios on different runs under 150 meters. In [6], trajectory estimation is carried out using visual cues for the sake of autonomously driving a car in inner-city conditions.

The rest of the paper is organized as follows: section II provides a description of the proposed non-linear method for estimating vehicle ego-motion and the simplified 2D vehicle trajectory estimation; implementation and results are provided in section III; finally, section IV is devoted to conclusions and discussion about how to improve the current system performance in the future.

II. VISUAL ODOMETRY USING NON-LINEAR ESTIMATION

In each frame, Harris corners [7] are detected, since this type of point feature has been found to yield detections that are relatively stable under small to moderate image distortions [8]. 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 right 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 7×7 pixels 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 [9] because of its robustness.

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 1.

$$R = \begin{pmatrix} cycz & sxsycz + xsz & -xsycz + xsz \\ -cysz & -sxsysz + xcz & cxsysz + sxz \\ sy & -sxy & cxy \end{pmatrix} \quad (1)$$

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 [10] 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:

$$\begin{cases} f_1(x_1, x_2, \dots, x_p) = b_1 \\ f_2(x_1, x_2, \dots, x_p) = b_2 \\ \vdots \\ f_n(x_1, x_2, \dots, x_p) = b_n \end{cases} \quad (2)$$

where f_i , for $i = 1, \dots, n$, is a differentiable function from \mathbb{R}^p to \mathbb{R} . In general, it can be stated that:

1. if $n < p$, the system solution is a $(p - n)$ -dimensional subspace of \mathbb{R}^p .
2. if $n = p$, there exists a finite set of solutions.
3. if $n > p$, 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.

$$E(\mathbf{x}) \triangleq \sum_{i=1}^N (f_i(\mathbf{x}) - b_i)^2 \quad (3)$$

The error function $E : \mathbb{R}^p \rightarrow \mathbb{R}$ can exhibit several local minima, although in general there is a single global minimum. Unfortunately, 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 4.

$$f_i(\mathbf{x} + \delta\mathbf{x}) = f_i(\mathbf{x}) + \delta x_1 \frac{\partial f_i}{\partial x_1}(\mathbf{x}) + \dots + \delta x_p \frac{\partial f_i}{\partial x_p}(\mathbf{x}) + O(|\delta\mathbf{x}|)^2 \approx f_i(\mathbf{x}) + \nabla f_i(\mathbf{x}) \cdot \delta\mathbf{x} \quad (4)$$

where $\nabla f_i(\mathbf{x}) = (\frac{\partial f_i}{\partial x_1}, \dots, \frac{\partial f_i}{\partial x_p})^t$ 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 an iterative process. Substituting (4) in (3) yields:

$$E(\mathbf{x} + \delta\mathbf{x}) = \sum_{i=1}^N (f_i(\mathbf{x} + \delta\mathbf{x}) - b_i)^2 \approx \sum_{i=1}^N (f_i(\mathbf{x}) + \nabla f_i(\mathbf{x}) \cdot \delta\mathbf{x} - b_i)^2 = |\mathbf{J}\delta\mathbf{x} - \mathbf{C}|^2 \quad (5)$$

where

$$\mathbf{J} = \begin{pmatrix} \nabla f_1(\mathbf{x})^t \\ \dots \\ \nabla f_n(\mathbf{x})^t \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial f_1}{\partial x_p}(\mathbf{x}) \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial f_n}{\partial x_p}(\mathbf{x}) \end{pmatrix} \quad (6)$$

and

$$\mathbf{C} = \begin{pmatrix} b_1 \\ \dots \\ b_n \end{pmatrix} - \begin{pmatrix} f_1(\mathbf{x}) \\ \dots \\ f_n(\mathbf{x}) \end{pmatrix} \quad (7)$$

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

$$\mathbf{J}\delta\mathbf{x} = \mathbf{C}, \quad (8)$$

System given by equation 8 can be solved using least squares, yielding:

$$\delta\mathbf{x} = (\mathbf{J}^t \mathbf{J})^{-1} \mathbf{J}^t \mathbf{C} = \mathbf{J}^\dagger \mathbf{C}. \quad (9)$$

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 $(E(\mathbf{x}_i) > \epsilon_{min})$ and $i < i_{max}$

$$\begin{aligned} & _ \delta\mathbf{x}_i = \mathbf{J}(\mathbf{x}_i)^\dagger \mathbf{C}(\mathbf{x}_i) \\ & _ \mathbf{x}_{i+1} = \mathbf{x}_i + \delta\mathbf{x}_i \\ & _ E(\mathbf{x}_{i+1}) = E(\mathbf{x}_i + \delta\mathbf{x}_i) = |\mathbf{J}(\mathbf{x}_i)\delta\mathbf{x}_i - \mathbf{C}(\mathbf{x}_i)|^2 \end{aligned}$$

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

B. 2D Approximation

Under the assumption that only 2D representations of the global trajectory are needed, like in a bird-eye view, the system can be dramatically simplified by considering that the vehicle can only turn around the y axis (strictly true for planar roads). It implies that angles θ_x and θ_z are set to 0, being θ_y estimated at each iteration.

Solving for the rotation matrix in equation (1) yields:

$$\mathbf{R} = \begin{pmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{pmatrix}. \quad (10)$$

In the following, the rotation matrix obtained in equation (1) is used as the approximate rotation matrix in the mathematical development explained in the previous section for 3D motion estimation. Given (10), the new equations system is:

$$\begin{cases} {}^1X_i = \cos\theta_y \cdot {}^0X_i - \sin\theta_y \cdot {}^0Z_i + t_x \\ {}^1Y_i = {}^0Y_i + t_y \\ {}^1Z_i = \sin\theta_y \cdot {}^0X_i + \cos\theta_y \cdot {}^0Z_i + t_z \end{cases}; \quad i = 1, 2, \dots, N$$

As observed, a non-linear equation with four unknown variables $\mathbf{w} = [\theta_y, t_x, t_y, t_z]^t$ is obtained. Thus, at least 2 points are needed to solve the system (or more than 2 points to solve the system using non-linear least squares). For each iteration k of the regression method, the following linear system has to be solved:

$$\mathbf{J}(\mathbf{w}_k)\delta\mathbf{w}_k = \mathbf{C}(\mathbf{w}_k). \quad (11)$$

where:

$$\mathbf{J}(\mathbf{w}_k) = \begin{pmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_N \end{pmatrix} = \begin{pmatrix} J_{1,11} & J_{1,12} & J_{1,13} & J_{1,14} \\ J_{1,21} & J_{1,22} & J_{1,23} & J_{1,24} \\ J_{1,31} & J_{1,32} & J_{1,33} & J_{1,34} \\ J_{2,11} & J_{2,12} & J_{2,13} & J_{2,14} \\ J_{2,21} & J_{2,22} & J_{2,23} & J_{2,24} \\ J_{2,31} & J_{2,32} & J_{2,33} & J_{2,34} \\ \vdots & \vdots & \vdots & \vdots \\ J_{N,11} & J_{N,12} & J_{N,13} & J_{N,14} \\ J_{N,21} & J_{N,22} & J_{N,23} & J_{N,24} \\ J_{N,31} & J_{N,32} & J_{N,33} & J_{N,34} \end{pmatrix},$$

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

$$\mathbf{C}(\mathbf{w}_k) = [c_{1,1}, c_{1,2}, c_{1,3}, \dots, c_{N,1}, c_{N,2}, c_{N,3}]^t.$$

In this case, the Jacobian submatrix \mathbf{J}_i , associated to point i , has a dimension of 3×4 .

The vector of independent terms $\mathbf{c}(\mathbf{w}_k)$ yields:

$$\begin{aligned} c_{i,1} &= {}^1X_i - \cos\theta_{y,k} \cdot {}^0X_i + \sin\theta_{y,k} \cdot {}^0Z_i - t_{x,k}, \\ c_{i,2} &= {}^1Y_i - {}^0Y_i - t_{y,k}, \\ c_{i,3} &= {}^1Z_i - \sin\theta_{y,k} \cdot {}^0X_i - \cos\theta_{y,k} \cdot {}^0Z_i - t_{z,k}. \end{aligned}$$

After computing the coefficients of \mathbf{J}_i at iteration k , system 11 is solved and the iterative process is resumed. On completion of the process (after i_{max} iterations as maximum) the algorithm yields the final solution $\mathbf{w} = [\theta_y, t_x, t_y, t_z]^t$ that describes the relative vehicle movement between two consecutive iterations.

RANSAC (RANdom SAmple Consensus) [11] [12] 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 below 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 belonging 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.

$$\begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} = R_{0,1} \begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + T_{0,1}; \quad i = 1, \dots, n \quad (12)$$

After drawing samples from three points, in step 1 models $\tilde{R}_{0,1}$ and $\tilde{T}_{0,1}$ 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 .

$$\begin{cases} \text{inlier} & e < t \\ \text{outlier} & e \geq t \end{cases} \quad (13)$$

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:

$$\begin{pmatrix} {}^1\tilde{x}_i \\ {}^1\tilde{y}_i \\ {}^1\tilde{z}_i \end{pmatrix} = \tilde{R}_{0,1} \begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + \tilde{T}_{0,1}; \quad i = 1, \dots, n$$

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

$$\mathbf{e} = \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} = \begin{pmatrix} {}^1\tilde{x}_i \\ {}^1\tilde{y}_i \\ {}^1\tilde{z}_i \end{pmatrix} - \begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} \quad (15)$$

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

$$e = |\mathbf{e}|^2 = \mathbf{e}^t \cdot \mathbf{e} \quad (16)$$

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

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.

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 outlier-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:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^s)} \quad (17)$$

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 $N = 10$.

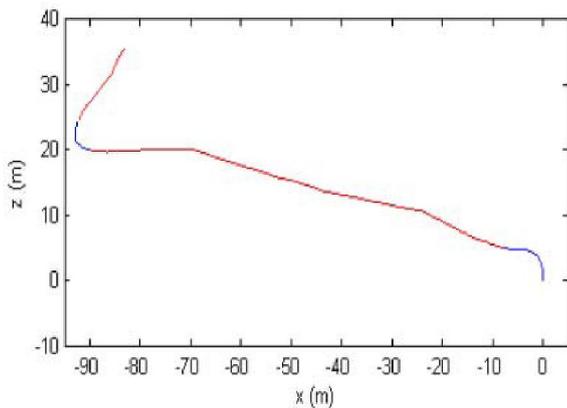
Consensus threshold T

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 :

$$T = (1 - \epsilon)n \quad (18)$$



a)



b)

Fig. 1. A) AERIAL VIEW OF THE AREA OF THE CITY WERE THE EXPERIMENT WAS CONDUCTED. B) ESTIMATED TRAJECTORY USING VISUAL ODOMETRY.

D. 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 maintained according to motion estimated in the previous frame. Removing false rotation estimations is a key aspect in visual odometry systems since false rotation estimations lead to high cumulative errors.

III. 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 320×240 pixels. All sequences correspond to real traffic conditions in urban environments. The simplified 2D method is very useful in practice for trajectory estimation in short runs.

The results of a first experiment are depicted in figure 1. 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 1 shows an aerial view of the area of the city (Alcalá de Henares) where 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.

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 length of the straight section in figure 1.b is 162.37m, which is very similar to the ground truth (165.86m). The estimated vehicle trajectory 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 degrees 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 1.

Sharp bends can be easily identified since they correspond to high values of the estimated yaw angle. Most of the times the number of outliers per frame remains low (below 6). Frames containing a high number of outliers (up to 13) are sporadic and isolated. This means that the feature extraction method is quite effective. The number of discarded frames in this experiment was 147, i.e., 12.25% of the total number of frames in the sequence. This can be considered a reasonable figure since the remaining frames still provide sufficient information for reliable position estimation. A remarkable point is the fact that discarded frames are rarely consecutive in time, allowing for robust interpolation using prediction from previous frames. System performance allows for algorithm execution at frame rate since the whole sequence (40s of duration) was analyzed by the system in 37.56s, including acquisition time.

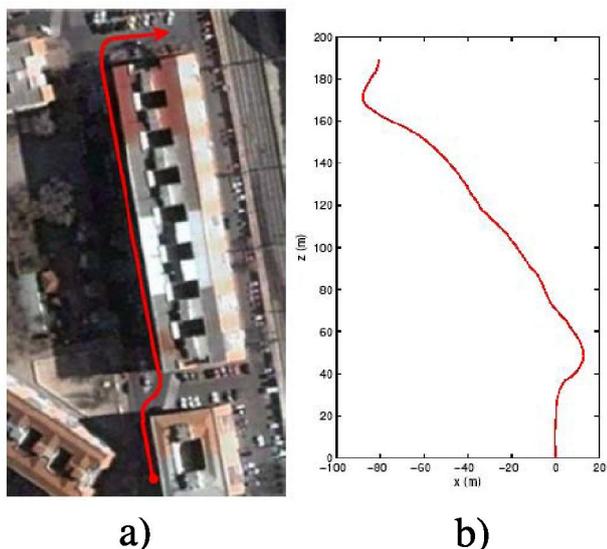


Fig. 2. A) AERIAL VIEW OF THE AREA OF THE CITY WERE THE EXPERIMENT WAS CONDUCTED. B) ESTIMATED TRAJECTORY USING VISUAL ODOMETRY.

In a second experiment, the car started turning slightly 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 2 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. The system estimated a distance of 189.89m in a real run of 194.33m. Video sequences showing the results obtained in several experiments in urban environments can be anonymously retrieved from <ftp://www.depeca.uah.es/pub/vision/visualodometry>.

IV. 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 approach 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.

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 Regional Government of Madrid by means of Research Grant CCG06-UAH/DPI-0411 and 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*, 1992, Vol. 14, No. 12, December.
- [2] D. Nister, O. Naroditsky, and J. Beren, "Visual odometry," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2004, June.
- [3] R. García, M. A. Sotelo, I. Parra, D. Fernández, and M. Gavilán, "3d visual odometry for gps navigation assistance," in *Proc. IEEE Intelligent Vehicles Symposium*, 2007, Istanbul, Turkey, June.
- [4] A. Hagnelius, "Visual odometry," in *Masters Thesis in Computing Science*, 2005, Umea University, April.
- [5] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," in *In 18th International Conference on Pattern Recognition (ICPR06)*, 2006, pp. 1063-1068.
- [6] N. Simond and M. Parent, "Free space in front of an autonomous guided vehicle in inner-city conditions," in *In European Computer Aided Systems Theory Conference (Eurocast 2007)*, 2007, pp. 362-363.
- [7] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conference*, 1988, pp. 147-151.
- [8] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," in *International Journal of Computer Vision*, 2000, Vol. 37, No. 2, pp. 151-172.
- [9] B. Boufama, "Reconstruction tridimensionnelle en vision par ordinateur: Cas des cameras non etalonnees," in *PhD thesis*, 1994, INP de Grenoble, France.
- [10] D. A. Forsyth and J. Ponce, *Computer Vision. A Modern Approach*, Pearson Education International, Prentice Hall, international edition, 2003.
- [11] 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*, 1981, June.
- [12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.