

# **VISUAL ODOMETRY FOR ACCURATE VEHICLE LOCALIZATION-AN ASSISTANT FOR GPS BASED NAVIGATION**

**Ignacio Parra, Miguel Ángel Sotelo, David F Llorca, Carlos Fenández**  
RobeSafe Research Group, University of Alcala  
Escuela Politecnica, Campus Universitario  
Ctra. Madrid-Barcelona, Km. 33,600  
Alcalá de Henares (Madrid)  
Phone: +34 91 8856726  
parra,sotelo,llorca,carlos.fernandez@depeca.uah.es

## **ABSTRACT**

This paper describes a new approach for improving the estimation of a vehicle motion trajectory in complex urban environments by means of visual odometry. A new strategy for compensating the heterodasticity in the 3D input data using a weighted non-linear least squares based system is presented. A Matlab simulator is used in order to analyze the error in the estimation and validate the new solution. The obtained results are discussed and compared to the previous system. The final goal is the autonomous vehicle outdoor navigation in large-scale environments and the improvement of current vehicle navigation systems based only on standard GPS. This research is oriented to the development of traffic collective systems aiming vehicle-infrastructure cooperation to improve dynamic traffic management. We provide examples of estimated vehicle trajectories using the proposed method and discuss the key issues for further improvement.

## **INTRODUCTION**

Accurate global localization has become a key component in vehicle navigation, not only for developing useful driver assistance systems, but also for achieving autonomous driving. Autonomous vehicle guidance interest has increased in the recent years, thanks to events like the Defense Advanced Research Projects Agency (DARPA) Grand Challenge and recently the Urban Challenge. Recently, the development of traffic collective systems for vehicle-infrastructure cooperation to improve dynamic traffic management has become a hot topic in ITS research.

Accordingly, our final goal is the autonomous vehicle outdoor navigation in large-scale environments and the improvement of current vehicle navigation systems based only on standard GPS. The work proposed in this paper is particularly efficient in areas where GPS

signal is not reliable or even not fully available (tunnels, urban areas with tall buildings, mountainous forested environments, etc). Our research objective is to develop a robust localization system based on a low-cost stereo camera system that assists a standard GPS sensor for vehicle navigation tasks. Then, our work is focused on stereovision-based real-time localization as the main output of interest.

The idea of estimating displacements from two 3D frames using stereo vision has been previously used in [1, 2] and [3]. A common factor of these works is the use of robust estimation and outliers rejection using RANSAC [4]. 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 m. In [6], trajectory estimation is carried out using visual cues for the sake of autonomously driving a car in inner-city conditions.

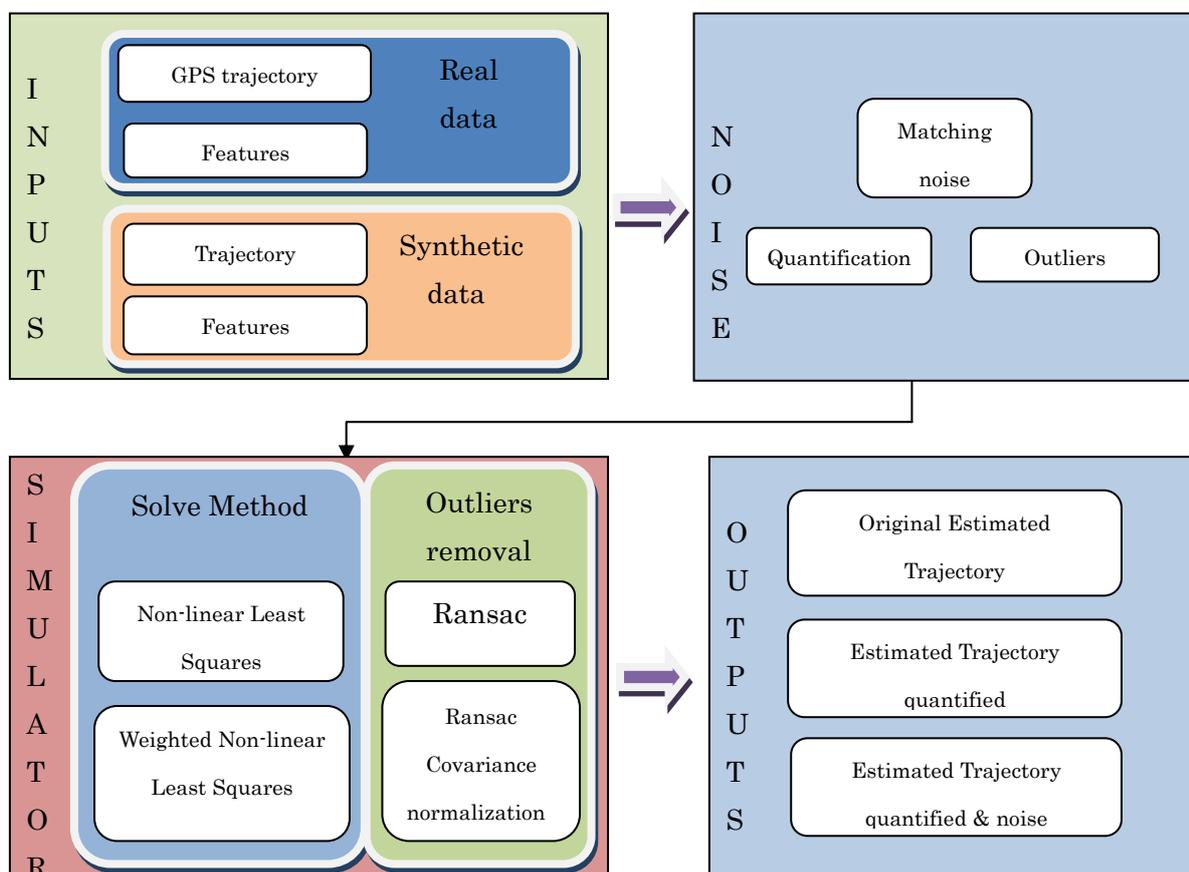
In our previous work [7] the ego-motion of the vehicle relative to the road is computed using a stereo-vision system mounted next to the rear view mirror of the car. Feature points are matched between pairs of frames and linked into 3D trajectories. Vehicle motion is estimated using the non-linear, photogrammetric approach based on RANSAC. In the present work, a comprehensive study of the errors in our previous ego motion estimation system has led to an improvement of the solution to the non-linear system equations describing the vehicle motion. To study the nature of these errors, a simulator was implemented in MatLab where all the variables of the system could be controlled. The results suggest heterogeneous variances of the input data due to the intrinsic error in 3D estimations. The new solution is based in a weighted non-linear least squares algorithm which compensates for the heterodasticity of the system input data. Results show a 20 times improvement in the medium distance to the ground true data and a better fit to the shape of the trajectory.

The rest of the paper is organized as follows: in section *MatLab Simulator* the new simulator is briefly described and the heterodasticity in the input data is shown; section *Non linear least squares* presents the new solution; section *Error in 3D estimation* provides a description of the intrinsic errors in 3D estimation using a stereo system which will be used for the weighting scheme; section *Results* the improvements achieved are compared to the previous system, and finally section *Conclusions and future works* is devoted to conclusions and discussion on how to improve the current system performance in the future.

## MATLAB SIMULATOR

Trajectory estimation and global positioning using ego motion is a difficult task from computer vision perspective. Large variations in environmental conditions (e.g. lighting, moving cars, poor texture scenes, repetitive patterns, etc.) make this problem particularly challenging. Understanding the influence of the different errors in the estimation is crucial to focus the research. In order to improve results a better understanding of the errors underlying our system is needed.

A mathematical study of the different errors present in the ego motion estimation was carried out using MatLab. To do so, the proposed trajectory estimation algorithm was programmed in MatLab assuming ideal conditions, and different errors were added one by one allowing us to measure their effect on the final trajectory estimation. In Figure 1 a flow diagram of the simulator is shown.



**Figure 1. Flow diagram of the simulator implemented in Matlab**

The *inputs* to the simulator are twofold: on the one hand the simulator can be fed with *synthetic data* (synthetic trajectories and synthetic feature points) which will be used to test the influence of parameters we cannot control in real data such as the distribution of feature points in the images or the length and turnings of a trajectory. On the other hand the simulator can process *data recorded* from real runs on the test vehicle. Trajectories are recorded using an RTK-dGPS (Real Time Kinematic- differential GPS) and the detected features are stored to

be used as inputs to the simulator.

The *outputs* of the simulator are three. Firstly the original (synthetic or real) trajectory is used as ground truth for comparison purposes. The first output is the trajectory obtained processing the raw input data. The second one is the trajectory estimated using the raw input data quantified. The last one is the trajectory estimated using the raw input data quantified and adding the noises and outliers. The difference between the ground truth and the estimated trajectory from raw input data is due to the loss of information in the linearization step.

The *inputs* to the simulator are now briefly described:

1. *Synthetic trajectory module*: This module creates a trajectory described by successive rotation matrices and translational vectors. The trajectory is created by a queue of codified commands such as “move forward 50m at 30km/h” or “turn 45 degrees at 2 degrees/s”. Also the sampling rate can be chosen (number of samples per second).
2. *Synthetic feature points module*: This module creates a cloud of N feature points using an statistical distribution inside programmable limits. For example we can create a cloud of uniformly distributed feature points between 2 and 20m in depth, -5 and 5m wide and 0 to 3m height. This allows us to measure the effect on the trajectory estimation of non uniformly distributed feature points such as areas where features are only detected on the road surface.
3. *Real GPS trajectories*: GPS NMEAS data are recorded with the test vehicle and the onboard standard GPS and RTK-dGPS. These data are converted to WGS-84 northing and easting and then into rotation and translational matrices for the simulator.
4. *Real 3D feature points*: The feature points detected by the real onboard system are stored and can be fed to the simulator. Although these points are not real 3D points in the sense that they will show errors in the 3D reconstruction, its spatial distribution will be more realistic than any random distribution we can create. This helps us to improve the algorithm in the simulator using data the real system will have to face.

Now the added *noises* and sources of error are described:

1. *Quantization in the 3D-2D transformation*: This is the error produced when the 3D points are projected into the image plane and transformed into pixel coordinates. The resolution of the pixel images is limited and, as a result, some information is lost in the projection. The higher the cameras resolution the smaller the error due to quantization. The 3D feature points (synthetic or real) passing through this module will be transformed according to the following projection equations:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} f/dx & 0 & u_0 & 0 \\ 0 & f/dy & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = M \cdot P \quad (1)$$

where  $(u,v)$  are the pixel coordinates for the 3D point  $(X,Y,Z)$  and  $(f,d_x,d_y,u_0,v_0)$  the camera intrinsic parameters. The camera intrinsic parameters are real camera calibration parameters obtained through a calibration process. This error is the cause for the decrease of accuracy in the depth coordinate estimation of stereo systems. The further the points selected for the trajectory estimation the higher the uncertainty there will be in the depth coordinate of these points.

2. *Noise in the matching process:* Given a calibrated rig of cameras and a correspondence between two points, one on the left camera  $(u_l, v_l)$  and another one on the right  $(u_r, v_r)$  the 3D position  $P$  of the point in the world coordinate system is given by :

$$A \cdot P = b \rightarrow P = (A^T \cdot A)^{-1} \cdot A^T \cdot b = (X, Y, Z)^T \quad (2)$$

where  $A$  is the matrix containing the equations for the 3D to 2D transformation for each one of the cameras and  $b$  the independent term of the same equations (see Equations 12 and 13). Matrices  $A$  and  $b$  are written as a function of the cameras intrinsic parameters and the matching points on the left camera  $(u_l, v_l)$  and the right camera  $(u_r, v_r)$ . Random noise is added to the pixel position of the right camera to express the uncertainty on the matching process.

$$p_r = (u_r, v_r) + N(0,1) \quad (3)$$

3. *Outliers from the matching process:* In urban cluttered environments repetitive patterns such as zebra crossings, building windows, fences, etc. can be found. As a consequence the input data for the ego-motion estimation will be regularly corrupted by these bad matches which will decrease the accuracy of the estimation. In this module synthetic outliers (points that don't fit the solution) are added to the initial data. The number and nature -how far they are from the solution- of the outliers can be programmed in the simulator.

Finally the **outputs** of the simulator are described:

1. *Original estimated trajectory:* The trajectory estimated by the simulator using the 3D matched pairs of points. No quantization or noise is added. This is the equivalent to

cameras with infinite resolution and perfect matches. The only error is due to the inaccuracies in the trajectory estimation algorithm, mainly because of the linearization of a non-linear problem.

2. *Estimated trajectory quantified*: The trajectory estimated by the simulator using the 3D matched pairs of points after the quantization process. This is the equivalent to cameras with finite resolution -which can be chosen in the simulator parameters- and perfect matches. The main error is due to the loss in accuracy in the quantization.
3. *Estimated trajectory quantified & noise*: The trajectory estimated by the simulator using the 3D matched pairs of points after the quantization process and adding noises. This is the equivalent to cameras with finite resolution, uncertainty in the matches and outliers. This is the most similar equivalent to the real system.

### **Heterodasticity of the input data**

Analysis of the regression residuals, or some transformation of the residuals, is very useful for detecting inadequacies in the model or problems in the data. The true errors in the regression model are assumed to be normally and independently distributed random variables with zero mean and common variance [8,9]. The plot of the residuals against the fitted values of the dependent variable is particularly useful. A random scattering of the points above and below  $e=0$  with nearly all the points being in a band is expected if the assumptions are satisfied. Any pattern in the magnitude of the dispersion about zero associated with changing estimated values ( $\hat{P}_1$ ) suggest heterogeneous variances of the error ( $e_i$ ).

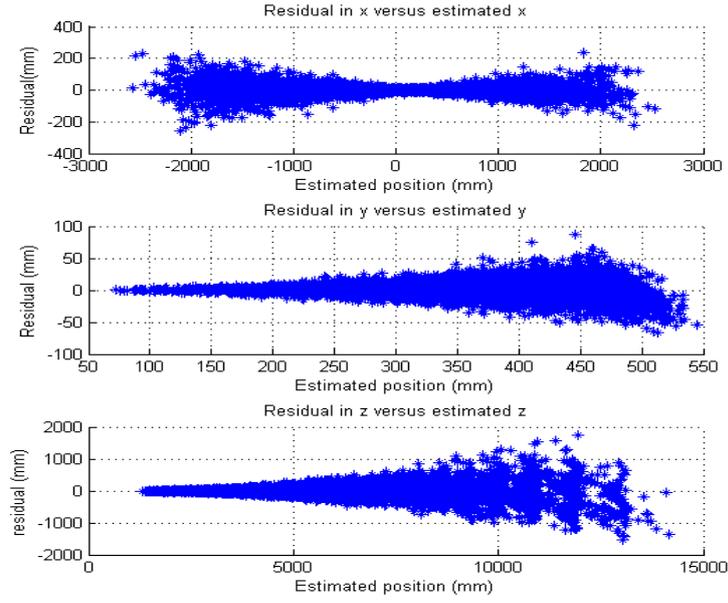
In this system the residuals are the difference between the estimated value and the value used to solve the system. Given two triplets of 3D points matched in times 0 and 1 ( $P_1, P_0$ ) we solve for the rotation matrix  $R$  and translational vector  $T$  that takes  $P_0$  to  $P_1$ :

$$\hat{P}_1 = R \cdot P_0 + T \quad (4)$$

The residual will be defined as:

$$e = \hat{P}_1 - P_1 \quad (5)$$

In Figure 3 the residuals for the estimations versus the estimated values are shown. The fan-shaped pattern in Figure 2 is the typical pattern when the variance increases with the mean of the dependent variable.



**Figure 2. Residuals for the estimated trajectory versus the estimated value**

If the functional relationship between the variance and the mean is known, a transformation exists that will make the variance (approximately) constant.

The linear least squares are based on the implicit assumption that the errors are uncorrelated with each other and with the independent variables and have equal variance. If, however, the measurements are uncorrelated but have different uncertainties, a modified approach might be adopted. When a weighted sum of squared residuals is minimized, the estimation is BLUE (Best Linear Unbiased Estimator) if each weight is equal to the reciprocal of the variance of the measurement [10].

## NON-LINEAR WEIGHTED LEAST SQUARES

Given a calibrated rig of cameras and a correspondence between 2 triplets of 3D reconstructed points at time  $t_0$  and  $t_1$  respectively  $P_0 = [X_0 \ Y_0 \ Z_0]$  and  $P_1 = [X_1 \ Y_1 \ Z_1]$  the system to solve for the movement is given by at least 3 of these triplets in the equation given by:

$$P_1 = P_0 \cdot R + T \quad (6)$$

where R is the rotational matrix for the movement and T is the translational vector. The motion between the two times is represented by vector  $w = [\theta_x \ \theta_y \ \theta_z \ t_x \ t_y \ t_z]$  where  $[\theta_x \ \theta_y \ \theta_z]$  are the pitch yaw and roll angles for the movement and  $[t_x \ t_y \ t_z]$  is the translational vector.

Using a non-linear weighted least squares scheme the function to minimize is given by the so called reprojection error and using the Taylor's expansion first term:

$$E(x) = \sum_{i=0}^N W_i \cdot (f_i(x) - b_i)^2 \approx \sum_{i=0}^N W_i \cdot (f_i(x) + \nabla f_i(x) \cdot \delta(x) - b_i)^2 \quad (7)$$

where  $W_i$  are the weights associated to each one of the residuals,  $f_i$  are Equation (6) for each one of the point pairs and  $b_i$  are the 3D coordinates for  $P_i^0$  given by the correspondence matching.

The equation to minimize can be written as:

$$W_i \cdot \nabla f_i(x) \cdot \delta(x) - W_i \cdot (b_i - f_i(x)) = 0 \quad (8)$$

Writing in matricial form:

$$W_i \cdot J \cdot dw = W_i \cdot C \quad (9)$$

where  $J = \nabla f_i(x)$  is the Jacobian matrix and  $C = (b_i - f_i(x))$

Solving using the pseudoinverse:

$$dw = (J^T \cdot W_i \cdot J)^{-1} \cdot J^T \cdot W_i \cdot C \quad (10)$$

The weight matrix for each point will be defined by its covariance matrix:

$$W_i = \frac{1}{\text{cov}(P_i)} \quad (11)$$

## ERROR IN 3D ESTIMATION

To compensate for the heterodasticity in the input data using weighted non-linear least squares we need to calculate an expression for the different variances of the reconstructed 3D points. Here, our approach for computing the quantization error covariance for each point is briefly described.

Given a calibrated rig of cameras and a correspondence between two points, one on the left camera  $(u_l, v_l)$  and another one on the right  $(u_r, v_r)$  the 3D position of the point in the world coordinate system is given by Equation (2) where  $A$  is the matrix containing the equations for the 3D to 2D transformation for each one of the cameras and  $b$  the independent term of the same equations. Matrices  $A$  and  $b$  are written as a function of the

cameras intrinsic parameters:

$$A = \begin{pmatrix} u_l \cdot m_{31}^L - m_{11}^L & u_l \cdot m_{32}^L - m_{12}^L & u_l \cdot m_{33}^L - m_{13}^L \\ v_l \cdot m_{31}^L - m_{21}^L & v_l \cdot m_{32}^L - m_{22}^L & v_l \cdot m_{33}^L - m_{23}^L \\ u_r \cdot m_{31}^R - m_{11}^R & u_r \cdot m_{32}^R - m_{12}^R & u_r \cdot m_{33}^R - m_{13}^R \\ v_r \cdot m_{31}^R - m_{21}^R & v_r \cdot m_{32}^R - m_{22}^R & v_r \cdot m_{33}^R - m_{23}^R \end{pmatrix} \quad (12)$$

$$b = \begin{pmatrix} m_{14}^L - u_l \cdot m_{34}^L \\ m_{24}^L - v_l \cdot m_{34}^L \\ m_{14}^R - u_r \cdot m_{34}^R \\ m_{24}^R - v_r \cdot m_{34}^R \end{pmatrix} \quad (13)$$

Each camera intrinsic parameters  $[M^L \ M^R]$  are estimated using an off-line calibration process. The intrinsic parameters describe the 3D to 2D transformation for each one of the cameras according to the Equation (1). In order to compute how the different errors in quantization in  $T = (u_l \ v_l \ u_r \ v_r)$  affect the 3D position the partial derivatives for Equation (2) are computed.

$$\frac{\partial(A \cdot P)}{\partial T} = \frac{\partial b}{\partial T} \quad (14)$$

Applying the product rule for matrices:

$$P^T \frac{\partial A^T}{\partial T} + A \frac{\partial P}{\partial T} = \frac{\partial b}{\partial T} \rightarrow A \frac{\partial P}{\partial T} = \frac{\partial b}{\partial T} - P^T \frac{\partial A^T}{\partial T} \quad (15)$$

denoting C as the right term of Equation (15) the expression for how the inaccuracies in the pixel position affect the 3D reconstruction is obtained:

$$A \cdot \frac{\partial P}{\partial T} = C \rightarrow \frac{\partial P}{\partial T} = (A^T \cdot A)^{-1} \cdot A^T \cdot C \quad (16)$$

Solving the partial derivatives for Equation (15) and using equations (12) and (13) and substituting values from equation (1):

$$C = \frac{\partial b}{\partial T} - P^T \frac{\partial A^T}{\partial T} = I_{4 \times 4} \cdot \begin{pmatrix} -m_{34}^L - m_{31}^L \cdot X - m_{32}^L \cdot Y - m_{33}^L \cdot Z \\ -m_{34}^L - m_{31}^L \cdot X - m_{32}^L \cdot Y - m_{33}^L \cdot Z \\ -m_{34}^R - m_{31}^R \cdot X - m_{32}^R \cdot Y - m_{33}^R \cdot Z \\ -m_{34}^R - m_{31}^R \cdot X - m_{32}^R \cdot Y - m_{33}^R \cdot Z \end{pmatrix} = \begin{pmatrix} -Z & 0 & 0 & 0 \\ 0 & -Z & 0 & 0 \\ 0 & 0 & -Z & 0 \\ 0 & 0 & 0 & -Z \end{pmatrix} \quad (67)$$

Assuming T is a normally distributed random variable with mean 0 and variance:

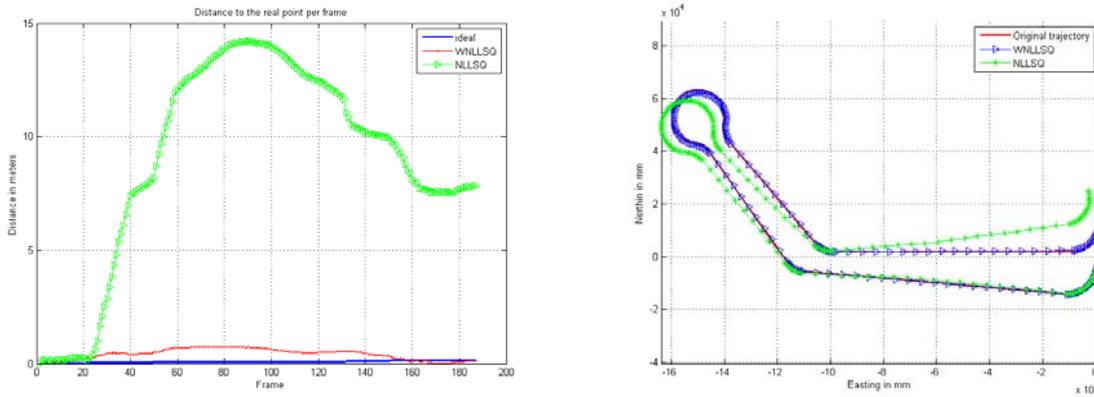
$$\sigma_T^2 = \begin{pmatrix} \sigma_{u_l}^2 & 0 & 0 & 0 \\ 0 & \sigma_{v_l}^2 & 0 & 0 \\ 0 & 0 & \sigma_{u_r}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v_r}^2 \end{pmatrix} \quad (18)$$

where  $\sigma_{u_l}^2, \sigma_{v_l}^2, \sigma_{u_r}^2, \sigma_{v_r}^2$  are the uncertainties in pixels on the measure of  $T$ , the final expression for the quantization error covariance is:

$$\text{cov}\left(\frac{\partial P}{\partial T} \cdot T\right) = E\left[\left(\frac{\partial P}{\partial T} \cdot T\right) \cdot \left(\frac{\partial P}{\partial T} \cdot T\right)^T\right] = \frac{\partial P}{\partial T} \cdot E[T^2] \cdot \left(\frac{\partial P}{\partial T}\right)^T = \begin{pmatrix} \Delta_{xx} & 0 & 0 \\ 0 & \Delta_{yy} & 0 \\ 0 & 0 & \Delta_{zz} \end{pmatrix} \quad (19)$$

## RESULTS

The proposed weighted non-linear least squares solution was tested in the simulator and compared to the standard non-linear least squares. The different algorithms were tested using a synthetic trajectory of approximately 406 m. with several turns and straight stretches. The simulation velocity was 30 km/h and the sampling rate was 6 frames per second. The feature points were generated using a uniform distribution ranging [-3 3] meters wide (x axis), [0 2] meters in height (y axis) and [1 20] m in depth (z axis). The same trajectory was reconstructed using the ideal system, non-linear least squares and the previously explained non-linear weighted least squares.



**Figure 3. (left) Distance to the real trajectory point per frame for the ideal, non-linear least squares and weighted non-linear least squares methods. (Right) Original, non-linear least squares and weighted non-linear least squares estimated trajectories**

The results in Table 1 show an improvement in the mean distance to the real points of about 20 times the previous ones. All the figures in the table are improved, but the most significant improvement is the actual shape of the estimated trajectory, which can be seen in Figure 3

(right). The trajectory with the weighted estimation keeps the shape of the original trajectory while the heterodasticity in the non weighted solution bends the trajectory drifting it away from the real one.

	Mean/max error in tx (mm)	Mean/max error in tz (mm)	Mean/max error in yaw (rad)	Max distance to real point (m)	Mean distance to real point (m)	Length of the run (m)	Estimated length (m)
<b>Ideal System</b>	0.000001	-0.000043	0.000000	0.137823	0.074141	405.793244	405.793815
	0.000022	0.000096	0.000000				
<b>non-linear LSQ</b>	-0.465969	-7.599702	-0.000021	14.190766	9.005537	405.793244	404.482130
	187.853817	178.345743	0.039577				
<b>Non-linear WLSQ</b>	-0.310400	-7.968557	-0.000036	0.746929	0.413442	405.793244	404.199836
	32.527168	155.569126	0.007994				

**Table 1. Performance of the ideal, non-linear least squares and weighted non-linear least squares estimated trajectories**

## CONCLUSIONS AND FUTURE WORK

We have described a method for improving the estimation of a vehicle's trajectory in urban environments by means of visual odometry. An analysis of the errors in the system has been presented using a simulator developed in MatLab. The results shown, indicate that the error in the 3D reconstruction is heterocedastic, making necessary a weighted non-linear least squares solution. Covariance of the 3D reconstruction has been computed to be used as weights for the weighted system. The resulting method has developed an error around 20 times smaller than the previous one and better fit to the shape of the trajectories.

We are currently fusing this visual odometry information with a commercial inexpensive GPS to produce accurate estimates of the vehicle global position. The final goal is to aid the navigation system in areas where GPS signal is not reliable or even not fully available (tunnels, urban areas with tall buildings, mountainous forested environments, etc). At the same time the ego motion information is being used to navigate, without a GPS receptor inside a map (OpenStreetMaps [11]) with the goal of the development of traffic collective systems aiming vehicle-infrastructure cooperation to improve dynamic traffic management

**Acknowledgements:** This work has been supported by the Spanish Ministry of Education and Science by means of the project GUIADE P9/08

## REFERENCES

- (1) Zhang, Z, Faugeras, O D, “*Estimation of displacements from two 3-D frames obtained from stereo*”. IEEE Trans. Pattern Anal. Mach. Intell. 14(12), 1141–1156, December (1992)
- (2) Nister, D, Naroditsky, O and Beren, J, “*Visual odometry*”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June (2004)
- (3) Hagnelius, A, “*Visual odometry*”. In: Masters Thesis in Computing Science, Umea University, April (2005)
- (4) Forsyth, D A, Ponce, J, “*Computer Vision. A Modern Approach*” (international edn.). Pearson Education International. Prentice Hall (2003).
- (5) Agrawal, M, Konolige, K, “*Real-time localization in outdoor environments using stereo vision and inexpensive gps*” In: 18th International Conference on Pattern Recognition (ICPR06), pp. 1063–1068 (2006)
- (6) Simond, N, Parent, M, “*Free space in front of an autonomous guided vehicle in inner-city conditions*” In: European Computer Aided Systems Theory Conference (Eurocast 2007), pp. 362–363 (2007)
- (7) Parra, I, Fernández, D, Sotelo, M A, Bergasa, L M, Revenga, P, Nuevo, J, Ocana, M, García, M A, “*Combination of feature extraction methods for SVM pedestrian detection*”. IEEE Trans. Intell. Transp. Syst. 8(2), 292–307, June (2007)
- (8) Rawlings J O, “*Applied regression analysis: A research tool*”. The wadsworth & brooks/Cole Statistics/Probability Series (1988).
- (9) Bates D M, Watts D G, “*Nonlinear regression analysis and its applications*”. Wiley series in probability and mathematical statistics (1988).
- (10) Aitken A C, “*On Least Squares and Linear Combinations of Observations*”, Proceedings of the Royal Society of Edinburgh, 55, 42-48. (1935)
- (11) <http://wiki.openstreetmap.org>