Robotica (2010) volume 28, pp. 441–452. © Cambridge University Press 2009
doi:10.1017/S026357470900575X

Robust visual odometry for vehicle localization in urban environments

I. Parra, M. A. Sotelo*, D. F. Llorca and M. Ocaña

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

(Received in Final Form: April 15, 2009. First published online: May 22, 2009)

SUMMARY

This paper describes a new approach for estimating the vehicle motion trajectory in complex urban environments by means of visual odometry. A new strategy for robust feature extraction and data post-processing is developed and tested on-road. Images from scale-invariant feature transform (SIFT) features are used in order to cope with the complexity of urban environments. The obtained results are discussed and compared to previous works. In the prototype system, the ego-motion of the vehicle 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 distance between estimations is dynamically adapted based on re-projection and estimation errors. Vehicle motion is estimated using the non-linear, photogrametric approach based on RAndom SAmple Consensus (RANSAC). The final goal is to provide on-board driver assistance in navigation tasks, or to provide a means of autonomously navigating a vehicle. The method has been tested in real traffic conditions without using prior knowledge about the scene or the vehicle motion. An example of how to estimate a vehicle's trajectory is provided along with suggestions for possible further improvement of the proposed odometry algorithm.

KEYWORDS: 3D visual odometry; Global localization; Vision; SIFT.

1. Introduction

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. Accurate global localization has become a key component in vehicle navigation, following the trend of the robotics area, which has seen significant progress in the last decade. 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 stereo-vision-based real-time localization as the main output of interest. Accurate estimation of the vehicle global position is a key issue, not only for achieving autonomous driving, but also for developing useful driver assistance systems. Using stereo vision for computing the position of obstacles or estimating road lane markers is a popular technique in intelligent vehicle applications. The challenge now is to extend stereo-vision capabilities to also provide accurate estimation of the vehicle's ego-motion with respect to the road, and thus to compute its global position. This is becoming more and more tractable to implement on standard PC-based systems.

In this paper, a new approach for ego-motion computing based on stereo vision is proposed, as shown in the flow diagram depicted in Fig. 1. 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 3D frames using stereo vision has been previously used in refs. [1–3]. A common feature of these studies is the use of robust estimation and outliers rejection using RAndom SAmple Consensus (RANSAC).[4] In ref. [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 ref. [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 ref. [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 of up to 150 m each. In ref. [6], trajectory estimation is carried out using visual cues for the sake of autonomously driving a car in inner-city conditions.

In the present work, the solution of the non-linear system equations describing the vehicle motion at each frame is computed under the non-linear, photogrametric approach using RANSAC. The use of RANSAC[2] allows for outliers rejection in 2D images corresponding to real traffic scenes, providing a method for carrying out visual odometry on-board a road vehicle.

The rest of the paper is organized as follows: in Section 2 the new feature detection and matching technique is

* Corresponding author. E-mail: sotelo@depeca.uah.es

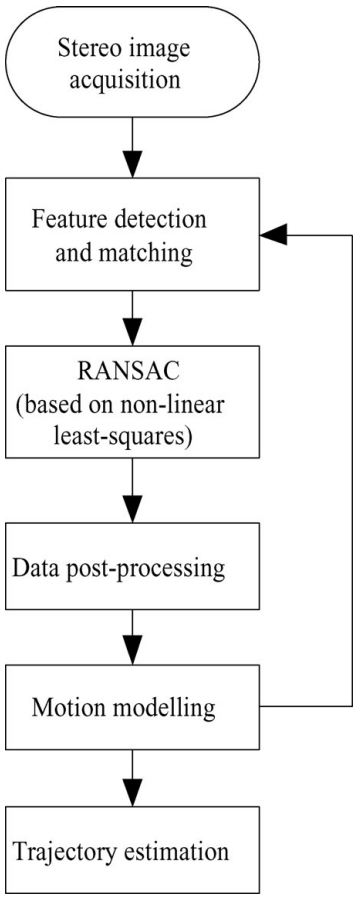


Fig. 1. General layout of the visual odometry method based on RANSAC.

presented; Section 3 provides a description of the proposed non-linear method for estimating the vehicle's ego-motion and the 3D vehicle trajectory; implementation and results are provided in Section 4; finally, Section 5 is devoted to conclusions and discussion on how to improve the current system performance in the future.

2. Features Detection and Matching

In most previous research on visual odometry, features are used for establishing correspondences between consecutive frames in a video sequence. Some of the most common choices are Harris corner detector[7] and the Kanade–Lucas–Tomasi detector (KLT).[8] Harris corners have been found to yield detections that are relatively stable under small to moderate image distortions.[9] As stated in ref. [2], distortions between consecutive frames can be regarded as fairly small when using video input. However, Harris corners are not always the best choice for landmark matching when the environment is cluttered and repetitive superimposed objects appear on the images. This is the situation for urban visual odometry systems. Although Harris corners can yield distinctive features, they are not always the best candidates for stereo and temporal matching. Among the wide spectrum of matching techniques that can be used to solve the correspondence problem, the *zero mean normalized cross correlation* (ZMNCC)[10] is chosen for robustness reasons. The ZMNCC 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

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

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

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 $(2n+1) \times (2n+1)$ window centred 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 order to minimize the number of outliers, a mutual consistency check is usually employed (as described in ref. [2]). Accordingly, only pairs of features that yield mutual matching are accepted as a valid match. The accepted matches are used both in 3D feature detection (based on stereo images) and in feature tracking (between consecutive frames).

In urban cluttered environments repetitive patterns such as zebra crossings, building windows, fences, etc. can be found. In Fig. 2 the typical correlation response along the epipolar line for a repetitive pattern is shown. Multiple maxima or even higher responses for badly matched points are frequent. Although some of these correlation mistakes can be detected using techniques such as the mutual consistency check or the unique maximum criterion, the input data for the ego-motion estimation will be regularly corrupted by these outliers which will decrease the accuracy of the estimation. Moreover, superimposed objects limit observed

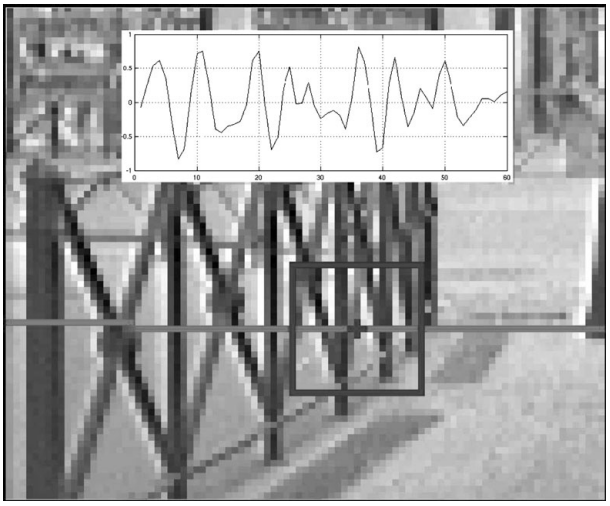
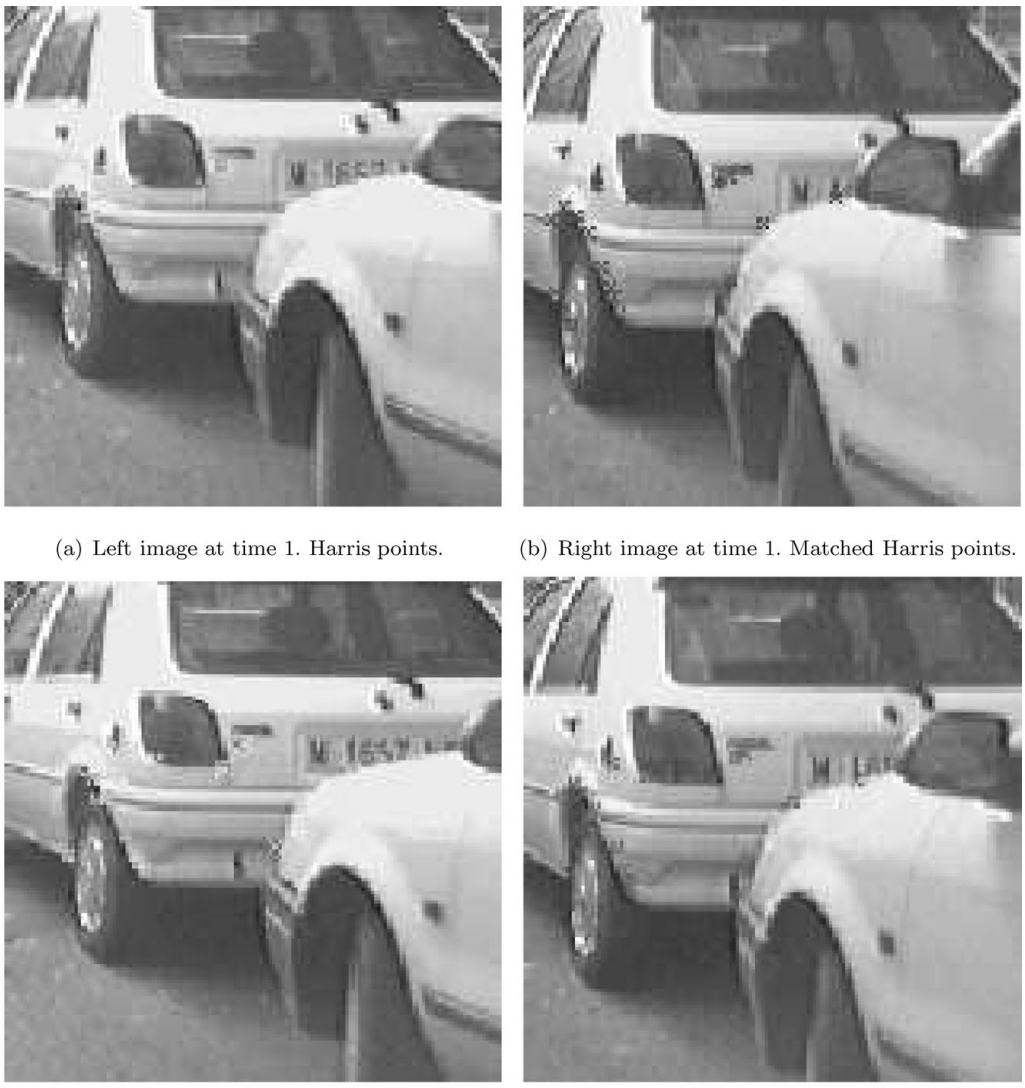


Fig. 2. Correlation response along the epipolar line for a repetitive pattern.



(a) Left image at time 1. Harris points.

(b) Right image at time 1. Matched Harris points.

(c) Left image at time 2. Harris points matched with Harris points from Left image at time 1. Outliers in orange.

(d) Right image at time 2. Harris points matched with Harris points from Left image at time 2.

Fig. 3. Examples of matches for superimposed objects.

from different viewpoints are a source of correlation errors for the system. Figure 3 depicts a typical example of an urban environment in which a car's bonnet is superimposed on the image of the next car's license plate and bumper. As can be seen in Fig. 3(a), the Harris corner extractor chooses, as feature points, the conjuncture in the image between the car's bonnet and the next car's license plate and bumper. In the image plane these are, apparently, good features to track, but the different depths of the superimposed objects will cause a mis-detection due to the different viewpoints. In Fig. 3(b) and 3(c) it can be seen how the conjuncture in the image between the number 1 on the license plate and the bonnet is matched but they do not correspond to the same point in the 3D space. We can see the same kind of mis-detection in the conjuncture between the car's bonnet and the bumper. The error in the 3D reconstruction of these points is not big enough to be rejected by the RANSAC algorithm so they will corrupt the final solution.

In practice, these errors lead to local minima in the solution space and thus to inaccurate and unstable estimations. A more reliable matching technique is needed in order to cope with the complexity of urban environments. In this system we apply an approach similar to ref. [11], in which images from scale-invariant feature transform (SIFT) features are used for simultaneous localization and map building (SLAMB) in unmodified (no artificial landmarks) dynamic environments. To do so they use a trinocular stereo system[12] to estimate the 3D position of the landmarks and to build a 3D map where the robot can be localized simultaneously. Our approach uses a calibrated stereo rig mounted next to the rear view mirror of a car to compute the ego-motion of the vehicle. In our system, at each frame, SIFT features are extracted from each of the four images (stereo pair at time 1 and stereo pair at time 2), and stereo matched among the stereo pairs (Fig. 4). The resulting matches for the stereo pairs are then, matched again among them. Only the features finding a matching pair in the

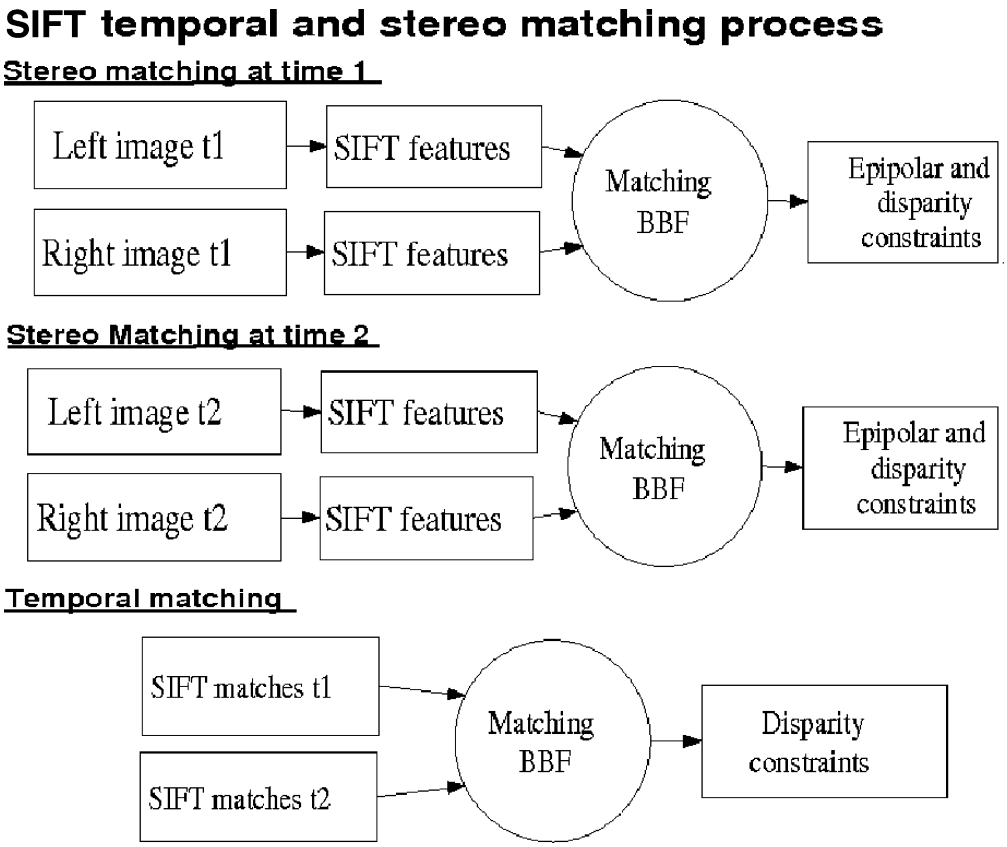


Fig. 4. Diagram of the features extraction method for the proposed system.

three matching processes will be used for the computation of the ego-motion.

SIFT was developed by Lowe[13] for image feature generation in object recognition applications. The features are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or 3D projection. These characteristics make them good feature points for robust visual odometry systems, since when mobile vehicles are moving around in an environment, landmarks are observed over time, but from different angles and distances. As described in ref. [14] the best matching candidate for a SIFT feature is its nearest neighbour, defined as the feature with the minimum Euclidean distance between descriptor vectors. The reliability of the nearest neighbour match can be tested by comparing its Euclidean distance to that of the second nearest neighbour from that image. If these distances are too similar, the nearest neighbour match is discarded as unreliable. This simple method works well in practice, since incorrect matches are much more likely to have close neighbours with similar distances than correct ones, due in part to the high dimensionality of the feature space. The large number of features generated from images, as well as the high dimensionality of their descriptors, make an exhaustive search for closest matches inefficient. Therefore the Best-Bin-First (BBF) algorithm based on a k-d tree search[15] is used. A k-d tree is constructed from all SIFT features which have been extracted from the reference images. The search examines tree leaves, each containing a feature, in the order of their closest distance from the current query location. Search order is determined with a heap-based priority queue. An approximate answer is returned after examining a predetermined number of nearest leaves. This technique finds the closest match with a high probability, and enables feature matching to run in real time. This can give speedup by factor of 1000 while finding the nearest neighbour (of interest) 95 % of the time. For each feature in a reference image, the BBF search finds its nearest and second nearest neighbour pair in each of the remaining images. Putative two-view matches are then selected based on the nearest-to-second-nearest distance ratio. As the SIFT best candidate search is not based on epipolar geometry, the reliability of matches can be improved by applying an epipolar geometry constraint to remove remaining outliers. This is a great advantage with respect to other techniques which rely on epipolar geometry for the best candidate search. For each selected image pair this constraint can be expressed as

$$x_l^T \cdot F \cdot x_r = 0, \tag{4}$$

where F is the fundamental matrix previously computed in an off-line calibration process and x_l^T, x_r are, respectively, the homogeneous image coordinates of the matched features in image *left* transposed and the homogeneous image coordinates of the matched features in image *right*. Also matches are only allowed between two disparity limits. Sub-pixel horizontal disparity is obtained for each match. This will improve the 3D reconstruction accuracy and therefore the ego-motion estimation accuracy. The resulting stereo matches between the first two stereo images are then similarly matched with the stereo matches in the next stereo pair. No

epipolar geometry constraint is applied at this step and an extra vertical disparity constraint is used. If a feature has more than one match satisfying these criteria, it is ambiguous and discarded so that the resulting matching is more consistent and reliable. From the positions of the matches and knowing the cameras' parameters, we can compute the 3D world coordinates (X, Y, Z) relative to the left camera for each feature in this final set. The number of final triple matches (match in the two stereo processes and in time) for each algorithm execution is around 50. Relaxing some of the constraints above does not necessarily increase the number of final matches (matches in the two stereo pairs and in time) because some SIFT features will then have multiple potential matches and therefore be discarded.

From the 3D coordinates of a SIFT landmark and the visual odometry estimation, we can compute the expected 3D relative position and hence the expected image coordinates and disparity in the new view. This information is used to search for the appropriate SIFT feature match within a region in the next frame. Once the matches are obtained, the ego-motion is determined by finding the camera movement that would bring each projected SIFT landmark into the best alignment with its matching observed feature. The good feature matching quality implies very high percentage of inliers, and therefore, outliers are simply eliminated by discarding features with significant residual errors E (currently 3 pixels). Minimization is repeated with the remainder matches to obtain the new correction term.

3. 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. For this purpose a RANSAC based on non linear least-squares method was developed for a previous visual odometry system. A complete description of this method can be found in ref. [16]. Nonetheless, an overview is given in this section for self-containing purpose (also see Fig. 1).

The estimation of the rotation angles must be undertaken by using an iterative, least squares-based algorithm[4] that yields the solution of the non-linear equations system that must 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.

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 Eq. (5):

$$R = \begin{pmatrix} cycz & sxsycz + cxsz & -cxsycz + sxsz \\ -cysz & -sxsysz + cxcz & cxsysz + sxcz \\ sy & -sxcy & cxcy \end{pmatrix}, \tag{5}$$

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[4] 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.

3.1. Non-linear least squares

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

$$\begin{cases} f_1(x_1, x_2, \ldots, x_p) = b_1 \\ f_2(x_1, x_2, \ldots, x_p) = b_2 \\ \qquad\qquad \vdots \\ f_n(x_1, x_2, \ldots, x_p) = b_n \end{cases}, \tag{6}$$

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 sub-space of $\Re^p$,
(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 sub-space 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:

$$E(\mathbf{x}) \stackrel{\triangle}{=} \sum_{i=1}^{N} (f_i(\mathbf{x}) - b_i)^2. \tag{7}$$

The error function $E : \Re^p \to \Re$ 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 Eq. (8):

$$\begin{aligned} f_i(\mathbf{x} + \delta\mathbf{x}) &= f_i(\mathbf{x}) + \delta x_1 \frac{\partial f_i}{\partial x_1}(\mathbf{x}) + \cdots \\ &\quad + \delta x_p \frac{\partial f_i}{\partial x_p}(\mathbf{x}) + O(|\delta\mathbf{x}|)^2 \approx f_i(\mathbf{x}) + \nabla f_i(\mathbf{x}) \cdot \delta\mathbf{x}, \end{aligned} \tag{8}$$

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 Eqs. (8) in (6) yields:

$$\begin{aligned} E(\mathbf{x} + \delta\mathbf{x}) &= \sum_{i=1}^{N} (f_i(\mathbf{x} + \delta\mathbf{x}) - b_i)^2 \\ &\approx \sum_{i=1}^{N} (f_i(\mathbf{x}) + \nabla f_i(\mathbf{x}) \cdot \delta\mathbf{x} - b_i)^2 = |\mathbf{J}\delta\mathbf{x} - \mathbf{C}|^2, \end{aligned} \tag{9}$$

where

$$\mathbf{J} = \begin{pmatrix} \nabla f_1(\mathbf{x})^t \\ \dots \\ \nabla f_n(\mathbf{x})^t \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial f_1}{\partial x_p}(\mathbf{x}) \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial f_n}{\partial x_p}(\mathbf{x}) \end{pmatrix}, \quad (10)$$

and

$$\mathbf{C} = \begin{pmatrix} b_1 \\ \dots \\ b_n \end{pmatrix} - \begin{pmatrix} f_1(\mathbf{x}) \\ \dots \\ f_n(\mathbf{x}) \end{pmatrix}. \quad (11)$$

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

$$\mathbf{J}\delta\mathbf{x} = \mathbf{C}. \quad (12)$$

System given by Eq. (12) can be solved using least squares, yielding

$$\delta\mathbf{x} = (\mathbf{J}^t\mathbf{J})^{-1}\mathbf{J}^t\mathbf{C} = \mathbf{J}^\dagger\mathbf{C}. \quad (13)$$

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.

3.2. Three-dimensional trajectory estimation

Between instants t_0 and t_1 we have:

$$\begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} = R_{0,1}\begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + T_{0,1}, \quad i = 1, \dots, N, \quad (14)$$

Considering Eq. (5) it yields a linear six-equations system at point i, with six variables $\mathbf{w} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z]^t$:

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

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):

$$\mathbf{J}(\omega)\delta\mathbf{x_k} = \mathbf{C}(\mathbf{x_k}), \quad (15)$$

with

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

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

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

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 3×6 sub-matrix associated to such point. Considering Eq. (10) the elements of the Jacobian Matrix that form sub-matrix $\mathbf{J_i}$ for point i at iteration k are

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

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

3.3. RAndom SAmple Consensus

RANSAC[17,18] 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 sub-set of points S_i for whom the distance to the model s is bellow the threshold t. Sub-set S_i (defined as consensus sub-set) defines the inliers of S.
(4) If the size of sub-set 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 sub-set 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 sub-set 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:

$$\begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} = R_{0,1} \begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + T_{0,1}, \quad i = 1, \ldots, n. \tag{16}$$

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:

$$\begin{cases} \text{Inlier} & e < t \\ \text{Outlier} & e \geq t \end{cases}. \tag{17}$$

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:

$$\begin{pmatrix} {}^1\widetilde{x}_i \\ {}^1\widetilde{y}_i \\ {}^1\widetilde{z}_i \end{pmatrix} = \widetilde{R}_{0,1} \begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + \widetilde{T}_{0,1}, \quad i = 1, \ldots, n. \tag{18}$$

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):

$$\mathbf{e} = \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} = \begin{pmatrix} {}^1\widetilde{x}_i \\ {}^1\widetilde{y}_i \\ {}^1\widetilde{z}_i \end{pmatrix} - \begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix}. \tag{19}$$

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

$$e = |\mathbf{e}|^2 = \mathbf{e}^t \cdot \mathbf{e}. \tag{20}$$

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

3.3.1. 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.

3.3.2. 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 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:

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)}. \tag{21}$$

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

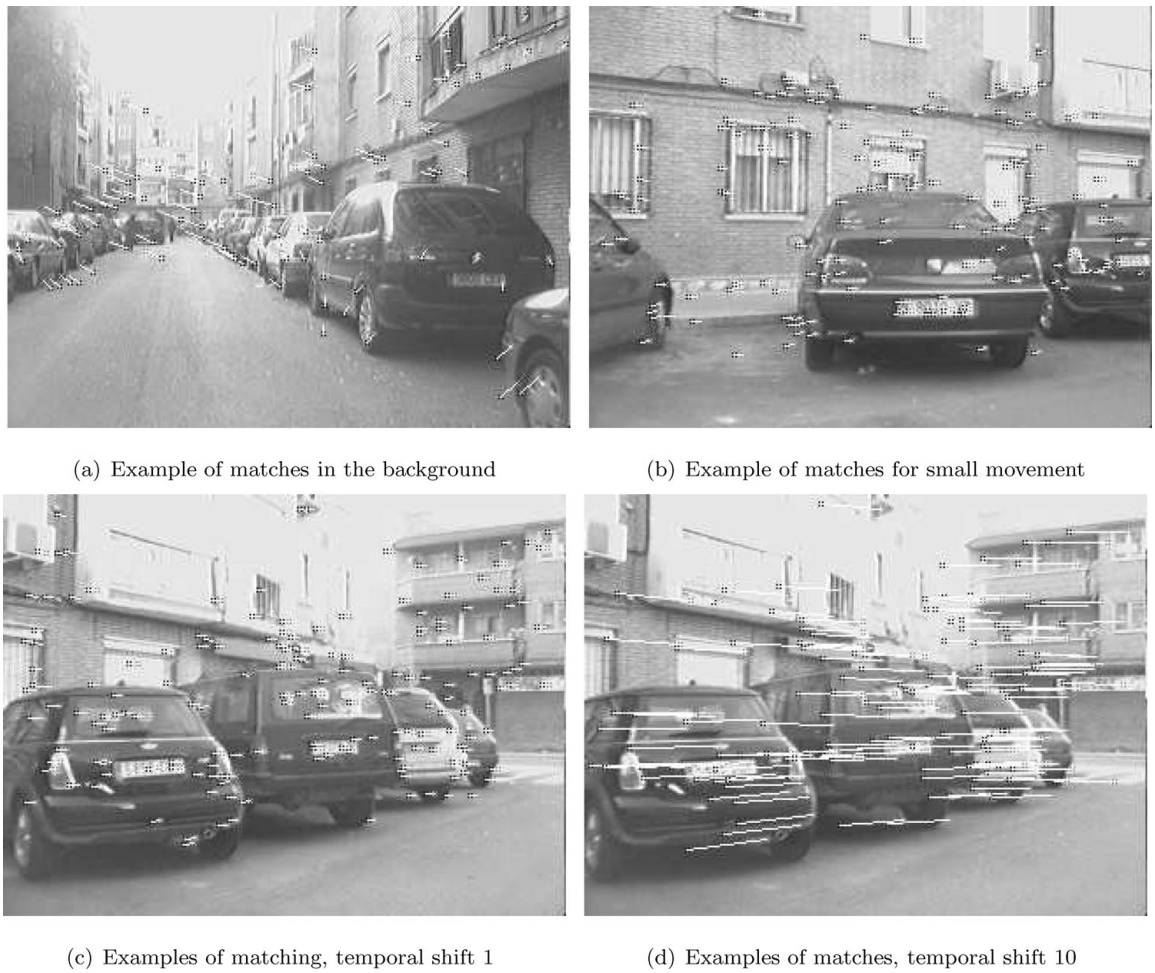
3.3.3. Consensus threshold T. The iterative algorithm ends whenever the size of the consensus set (composed of inliers) is larger than the number of expected inliers T given by ϵ and n:

$$T = (1 - \epsilon)n. \tag{22}$$

3.4. Two-dimensional approximation

Under the assumption that only 2D representations of the global trajectory are needed, like in a bird's-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.

A non-linear equation with four unknown variables $\mathbf{w} = [\theta_y, t_x, t_y, t_z]^t$ is obtained where $T = [t_x, t_y, t_z]$ is the translational vector.



(a) Example of matches in the background

(b) Example of matches for small movement

(c) Examples of matching, temporal shift 1

(d) Examples of matches, temporal shift 10

Fig. 5. Examples of SIFT matches. In green SIFT feature at time $t1$ in blue matched feature at time $t2$, in white the movement of the feature.

After an iterative process using all the points obtained from the matching step 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.

This approximation, along with the RANSAC outliers rejection step, allows the system to cope with moving objects such as pedestrians or other cars. On the one hand RANSAC will reject every minimal solution as long as the number of stationary points being tracked is higher than the outliers (pedestrians or other moving cars). On the other hand the 2D approximation adds some information about the car dynamics to the model. Future versions of the system will filter the final trajectory using the vehicle dynamics in a Kalman filter.

3.5. Data post-processing

This is the last stage of the algorithm. In most previous research on visual odometry, features are used for establishing correspondences between consecutive frames in a video sequence. However, it is a good idea to skip the frames yielding physically incorrect estimations or with a high mean square error to get more accurate estimations.

We have found there to be two main sources of errors in the estimation step:

(1) Solutions for small movements (5 cm or less) where the distance between features is also small (one or two pixels), are prone to yield inaccurate solutions due to the discretized resolution of the 3D reconstruction (Fig. 5b).
(2) Solutions for images where the features are in the background of the image (Fig. 5a) are inaccurate for the same reason as previously mentioned: 3D reconstruction resolution decreases as long as depth increases. Although the features extraction algorithm sorts the features depending on its depth and it uses the closest ones, at some frames it is not able to find enough features close to the car.

SIFT features have proven to be robust to pose and illumination changes, so they are good candidates for matching, even if there are some skipped frames between the matching stereo pairs and thus, the appearance of the features has changed (Fig. 5d). Also the fact that they do not rely on the epipolar geometry for the matching process makes its computational time independent on the disparity between features. Using a correlation based matching process it would be necessary to increase the disparity limits in order to find the features which will probably be further away from each other. According to this some ego-motion estimations are discarded using the following criteria:

(a) (b)

Fig. 6. (a) Prototype vehicle, (b) stero-camera platform on-board the vehicle.

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

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 ego-motion is computed again using frames ti and $t(i + 1 + shift)$ where *shift* is an integer which increases by one at every iteration. This process is repeated until an estimation meets the criteria explained above or the maximum temporal shift between frames is reached. The maximum temporal shift has been fixed to 5. By doing so the spatial distance between estimations remains small and thus the estimated trajectory is accurate. Using this maximum temporal shift the maximum spatial distance between estimations will be around 0.5–2.5 m. If the system is not able to get a good estimation after five iterations the estimated vehicle motion is maintained according to motion estimated in the previous correct frame assuming that the actual movement of the vehicle can not change abruptly. The system is working at a video frame rate of 30 fps which allows to skip some frames without losing precision in the trajectory estimation.

4. Implementation and Results

The visual odometry system described in this paper has been implemented on a Core II Duo at 2.16 GHz running Kubuntu GNU/Linux 6.1 with a 2.6.20-16 SMP kernel version. The algorithm is programmed in C using OpenCV libraries (version 0.9.9). A stereo-vision platform based on Fire-i cameras (IEEE1394) was installed on a prototype vehicle, as depicted in Fig. 6. 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 a non-compression algorithm at 30 frames/s with a resolution of 320×240 pixels. All sequences correspond to real traffic conditions in urban environments with pedestrians and other cars in the scene. In the experiments, the vehicle was driven below the maximum allowed velocity in cities, i.e. 50 Km/h. Theoretically, the ego-motion estimation is not affected by the vehicle's speed as long as there are enough linked features between every two frames. In fact, as explained in Section 3.5, a temporal shift is used when the estimation is not good enough. From the system's point of view this has the same effect as if the car were moving faster. The average computation time is 1 frame/sec but no effort has been put on code optimization, and real time is feasible by implementing on hardware some key parts of the algorithm. Glares on the windscreen are the main source of outliers caused by changes on the illumination conditions. It is absolutely necessary to place a sun-shade-like device on the camera lenses to protect them from glares. These types of devices are used in the automotive industry. The intrinsic and extrinsic parameters of the cameras and the distortion parameters are obtained using the method implemented in the *Camera Calibration Toolbox* from Matlab.[19] This method requires the use of a chessboard and is performed in two steps. In the first step, intrinsic parameters are estimated without considering distortion parameters using a linear approximation in closed form. In the second step, a non-linear optimization method is applied based on iterative gradient descent. The list of parameters obtained after calibration is provided bellow:

(a) Left camera: fxl, fyl, (u0l,v0l): 423.908295, 423.838776, (163.685577,113.995888),
(b) Right camera: fxr, fyr, (u0r,v0r): 426.694031,427.079895, (152.329453,120.181602),

where *fxi* and *fyi* represent the focal length in x and y dimensions for the *i* camera (left or right) in pixel/mm, and *u0i* and *v0i* stand for the optical centre coordinates in the *i* camera (left or right).

4.1. Visual odometry results

The results of a first experiment are depicted in Fig. 7. The vehicle starts on 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°of variation in yaw. The upper part of Fig. 7 shows an aerial view of the area of the city (Alcalá de Henares) where the

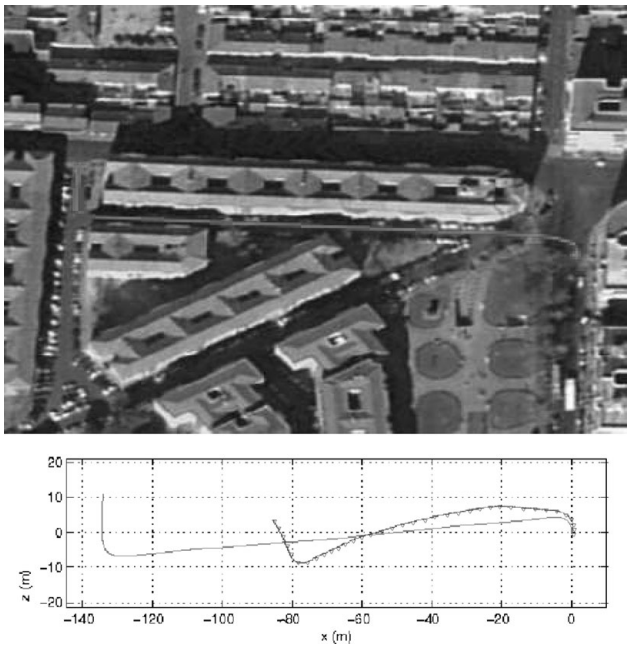


Fig. 7. Above, trajectory in the city for experiment 1. Below estimated trajectory for the previous Harris feature extractor (triangles) and for the new SIFT strategy (no markers).

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 (no marker) and the previous version of the system using Harris corners and ZMNCC (triangles).[16]

As can be observed, the system provides reliable estimations of the path run by the vehicle in all the sections. As a matter of fact, the estimated length run in Fig. 7 is 147.37 m, which is very similar to the ground truth (165.86 m). Compared to the previous system the trajectory is more accurate and closer to the actual length of the run. Taking into account that 13.84 % of the frames were discarded in the post-processing step, the actual length of the run is quite close to the real one.

In a second experiment, the car starts turning left and then runs along an almost straight path for a while. After that, a sharp right turn is executed. Then the vehicle moves straight for some metres and turns slightly right until the end of the street. Figure 8 illustrates the real trajectory described by the vehicle (above) and the trajectory estimated by the visual odometry algorithm (below). The estimated trajectory reflects the exact shape of the real trajectory executed by the vehicle quite well. The system estimated a distance of 197.89 m in a real run of 216.33 m. Similar to the first experiment, 9.51 % of the estimations were discarded by the post-processing step, thus the actual length of the run is again very close to the real one.

4.2. Discussion

After observation of the results provided in the previous section, it can be stated that the 3D visual odometry described in this paper provides approximate trajectory estimations that can be useful for enhancing GPS accuracy, or even for substituting GPS in short outage periods. Nonetheless,

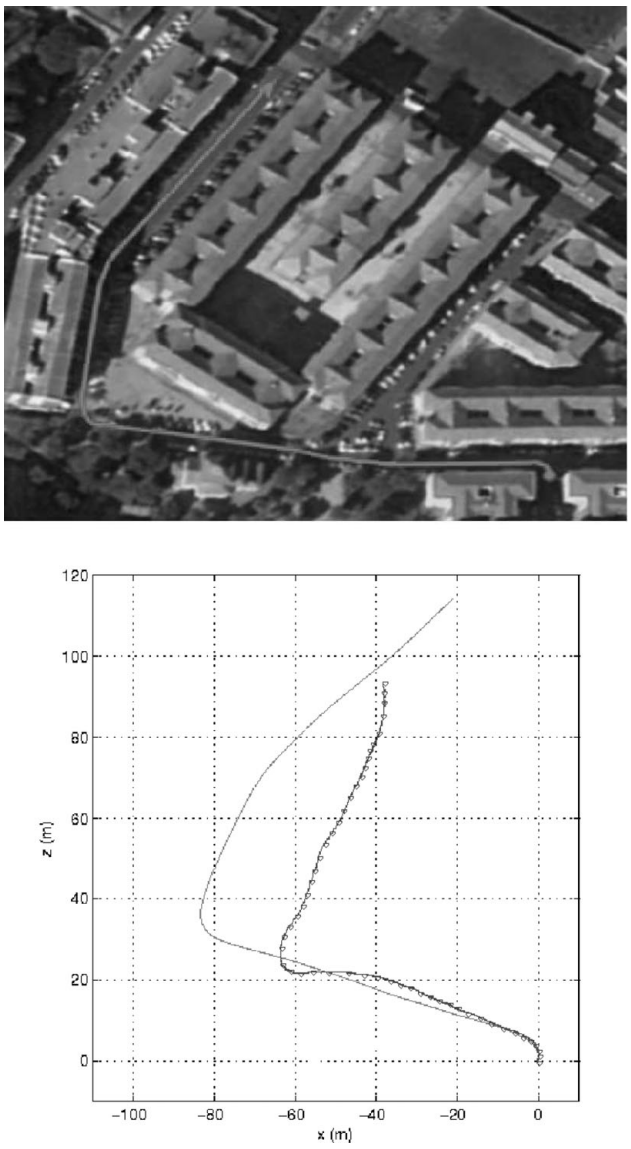


Fig. 8. Above, trajectory in the city for experiment 2. Below estimated trajectory for the previous Harris feature extractor (triangles) and for the new SIFT strategy (no markers).

the system provides estimations that exhibit cumulative errors. Thus, it can not be realistically expected that a 3D visual odometry system be used as a stand alone method for global positioning applications. Apart from this obvious fact, other problems arise especially in altitude estimation. The reason for this stems from the fact that estimations of pitch and roll angles become complex using visual means, since variations of these angles in usual car displacements are really small and difficult to measure in the 2D image plane. These difficulties produce a non-real altitude change in estimated 3D trajectories. Besides, the estimation of pitch and roll angles leads to a decrease in the accuracy of yaw angle estimation with regard to the 2D simplified method. As a consequence of that a greater error in estimated distance occurs. In addition, the 3D visual odometry method needs higher computational requirements to maintain performance at frame rate. Another problem arises when features corresponding to non-stationary objects are detected and used by the system. Non-stationary features

lead to unrealistic motion estimation. This effect is observed with greater magnitude when the car is not moving. So, for instance, if the car is stopped at an intersection or a traffic signal, and other cars or pedestrians appear in the scene, the visual odometry method tends to produce unreal motion estimation in a direction that is contrary to the objects' movements. Though small, this is an upsetting effect that must be removed in future developments.

Finally, considering the possibility of a future commercial implementation of a visual odometry system for GPS enhancement, the simplified 2D estimation method described in this paper is a realistic, viable option that can help increase conventional GPS accuracy or even support GPS in short outage periods. 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. The videos show a compound image in which the original input image and the estimated car trajectory image are synchronized and depicted together for illustrative purpose.

5. 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. To do so, SIFT feature points are extracted and matched along 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 with, based on the root mean square error of the estimation and the vehicle dynamics. An adaptive temporal shift which tries to avoid bad estimations has also been developed. 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.

Real experiments have been conducted in urban environments in real traffic conditions with no prior knowledge of the vehicle movement or the environment structure. We provide examples of estimated vehicle trajectories using the proposed method. Although preliminary, the first results are encouraging since it has been demonstrated that the system is capable of providing approximate vehicle motion estimation.

As part of our future work we envision the development of a method for discriminating stationary points from those which are moving in the scene. Moving points can correspond to pedestrians or other vehicles 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[20] and ACC (based on vehicle detection). The output of these systems can guide the search for stationary points in the 3D scene. Also a tracking of the features has to be addressed using the information of the movement estimations and a Kalman filter which will estimate the feature's next position. This information will be used to determine a region of interest for the feature extraction algorithm and also to compute the features' probability of being stationary points. This will allow to better deal with pedestrians, cars and other moving objects in the scene. This probability will be used for the resolution of the system using a weighted non-linear least squares method in which every point in the system will be weighted by its probability of being a stationary point.

The obvious application of the method is to provide a means for autonomously navigating a vehicle or to provide on-board driver assistance in navigation tasks. 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 and the Regional Government of Madrid by means of Research Grant CCG06-UAH/DPI-0411.

References

1. Z. Zhang and O. D. Faugeras, "Estimation of displacements from two 3-d frames obtained from stereo," *IEEE Trans. Pattern Analysis and Mach. Intell*. **14**(12) (Dec. 1992).
2. D. Nister, O. Narodistsky and J. Beren, "Visual Odometry," *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA (June, 2004). Vol. 1, pp. 652–659.
3. A. Hagnelius, Visual Odometry *Masters Thesis* (Umea, Sweden: Umea University, Apr. 2005).
4. D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach* (Prentice Hall, 2003).
5. M. Agrawal and K. Konolige, "Real-Time Localization in Outdoor Environments Using Stereo Vision and Inexpensive GPS," *Eighteenth International Conference on Pattern Recognition (ICPR06)*, Hong Kong, China (2006) pp. 1063–1068.
6. N. Simond and M. Parent, "Free Space in Front of an Autonomous Guided Vehicle in Inner-City Conditions," *European Computer Aided Systems Theory Conference (Eurocast 2007)*. Las Palmas de Gran Canaria, Spain (2007) pp. 362–363.
7. C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Proceedings of the Fourth Alvey Vision Conference*. Manchester, UK (1988) pp. 147–151.
8. B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Aplication to Stereo Vision," *Proceedings of the International Joint Conference on Artifial Intelligence*. Vancouver, Canada (1981) pp. 674–679.
9. C. Schmid, R. Mohr and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vis*. **37**(2) 151–172 (2000).
10. B. Boufama, Reconstruction tridimensionnelle en vision par ordinateur: Cas des cameras non etalonnees *PhD Thesis* (France, INP de Grenoble, 1994).
11. S. Se, D. Lowe and J. Little, "Vision-Based Mobile Robot Localization and Mapping Using Scale-Invariant Features," *Proceedings of the IEEE ICRA*. Seoul, Korea (2001) pp. 2051–2058.
12. D. Murray and J. Little, "Using Real-Time Stereo Vision for Mobile Robot Navigation," *Proceedings of the IEEE Workshop on Perception for Mobile Agents*. Santa Barbara, CA, USA (1998).
13. D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proceedings of the Seventh ICCV*. Kerkyra, Greece (1999) pp. 1150–1157.

14. I. Gordon and D. G. Lowe, "What and Where: 3D Object Recognition with Accurate Pose," *International Symposium on Mixed and Augmented Reality*. Santa Barbara, CA, USA (2006). pp. 67–82.
15. J. S. Beis and D. G. Lowe, "Shape Indexing Using Approximate Nearest-Neightbour Search in High-Dimensional Spaces," *Proceedings of the IEEE Conference on CVPR*. San Juan, Puerto Rico (1997) pp. 1000–1006.
16. R. García-García, M. A. Sotelo, I. Parra, D. Fernández, J. E. Naranjo and M. Gavilán, "3D visual odometry for road vehicles," *J. Intell. Robot. Syst*. **51**, 113–134 (2008).
17. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*. **24**(6), 381–395 (June, 1981).
18. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. (Cambridge University Press, 2004).
19. Matlab, "Camera calibration toolbox for matlab," (2007), http://www.vision.caltech.edu/bouguetj/calib_doc/.
20. I. Parra, D. Fernández, M. A. Sotelo, L. M. Bergasa, P. Revenga, J. Nuevo, M. Ocana and M. A. García, "Combination of feature extraction methods for svm pedestrian detection," *IEEE Trans. Intell. Transp. Syst*. **8**(2), (June, 2007).