Stereo Visual Odometry in Urban Environments Based on Detecting Ground Features

Arturo de la Escalera[a], Ebroul Izquierdo[b], David Martín[a], Basam Musleh[a], Fernando García[a], Jose María Armingol[a]

[a] *Universidad Carlos III de Madrid, Leganés, Spain*
[b] *Queen Mary University London, London, UK*

Abstract

Autonomous vehicles rely on the accurate estimation of their pose, speed and direction of travel to perform basic navigation tasks. Although GPSs are very useful, they have some drawbacks in urban applications that affect their accuracy. Visual odometry is an alternative or complementary method because provides the ego motion of the vehicle with enough accuracy and uses a sensor already available in some vehicles for other tasks, so no extra sensor is needed. In this paper, a new method is proposed that detects and tracks features available on the surface of the ground, due to the texture of the road or street and road markings. This way it is assured only static points are taking into account in order to obtain the relative movement between images. A Kalman filter improves the estimations and the Ackermann steering restriction is applied so the vehicle follows a constrained trajectory, which improves the camera displacement estimation obtain from a PnP algorithm. Some results in real urban environments are shown in order to demonstrate the good performance of the algorithm. They show the method is able to estimate the linear and angular speeds of the vehicle with high accuracy as well as its ability to follow the real trajectory drove by the vehicle along long paths within a minimum error.

Keywords: Autonomous Vehicles, Visual Odometry, Kalman Filter

1. Introduction

Vehicle localization is a fundamental task in autonomous vehicle navigation. It relies on accurate estimation of pose, speed and direction of travel

to achieve basic tasks including mapping, obstacles avoidance and path following. Nowadays, many autonomous vehicles rely on GPS-based systems for estimating their ego motion. Although GPSs are very useful, they have some drawbacks. The price of the equipment is still high for the centimeter accuracy needed for autonomous applications. Moreover, above all for urban applications, the shortcomings of the GPSs are clearer because there are some situations that affect their accuracy. For example, there may not be a direct line of sight to one or several satellites because of the presence of a building or a tree canopy. The urban canyon effect is very frequent within cities due to building heights. Finally, the vehicle has not available the GPS signal for an important task as driving along tunnels. Other sensors available are low-cost IMUs; however, although they are fast, they have a measurement bias and therefore need frequent corrections. Several solutions can be proposed to solve this problem, such as the use of maps or odometry provided by the vehicle wheels. The first one needs a continuous updating of the maps to be useful and the second lacks enough precision for several applications. That is why another sensor is needed and here is where digital cameras can play an important role. On one hand because, as it will be shown, they are useful for obtaining the vehicle's ego motion and, on the other hand, because nowadays they are already used for other tasks such as pedestrian, traffic sign or road lane detection [1], accordingly it is a sensor that can be applied for multiple assignments. Visual Odometry (VO) estimates the ego motion of a camera or a set of cameras mounted on a vehicle using only the visual information provided by it or them. The term is related to the wheel odometry used in robotics and was formulated in 2004 by Nister [2]. Usually, VO algorithms have three steps:

1. Detect features or points of interest (POI) in every image and match the ones found in two consecutive ones.
2. Find and remove the wrong matches.
3. Estimate the relative movement of the cameras.

This can be done using monocular or stereo cameras and assuming planar or non-planar motion models. A tutorial on VO can be found in [3] [4]. In [5] a stereo system is presented, where it estimates the rigid body motion that best describes the transformation among the sets of 3D points acquired in consecutive frames. Optical flow and stereo disparity are computed to minimize the re-projection error of tracked feature points. Instead of performing this task using only consecutive frames, they use the whole

history of the tracked features to compute the motion of the camera. The camera motion is estimated in [6] using a quaternion and RANSAC [7] for outlier removal and a Two-stage Local Binocular Bundle Adjustment for optimizing the results. In [8] the rotation and translation between consecutive poses are obtained, minimizing the distance of the correspondent point projections. They take into account that farther 3D points have higher uncertainty, RANSAC for outliers and they constrain pose estimation taking temporal flow into account. A persistent map containing 3D landmarks localized in a global frame is presented in [9]. They automatically distinguish some frames, used to update the landmark map, which serves for ego-localization. The other frames are used to track the landmarks and to localize the camera with respect to the map. In [10], they apply some monocular techniques to stereo visual odometry system. The features are detected using FAST [11], described with BRIEF [12] and tracked during the image sequence. A P3P algorithm is used for the pose estimation and local bundle adjustment is used for result refinement. Other sensors, like lasers, has been used in [13][14][15]. In [16], the authors combine visual and lidar odometry. Visual odometry is useful to estimate the ego-motion and as a help to register point clouds from a scanning lidar, which refines the motion estimation.

Urban environments are highly dynamic, so the case of a static scene cannot be assumed. Moreover, these surroundings are highly cluttered with frequent occlusions. Consequently, there are some specific difficulties any method has to face:

- The detected POI can belong to moving objects and, as a consequence, the camera motion estimation would be erroneous if they are used for obtaining the camera displacement.

- Due to ego motion and occlusions, some detected POI in an image are not detected in the next one, and vice versa, but this can lead to an erroneous matching and, again, to an erroneous motion estimation.

The novelty of the proposed algorithm is related to the previous difficulties. This paper is an extended version of the one presented at the Second Iberian Robotics Conference, Robot2015, in Lisbon, Portugal [17]. Here, a more detail explanation of the algorithm is presented as well as the experiments, which number has been increased. The overall of the algorithm can be seen in Fig.1. Due to the two difficulties for urban environments mentioned before, in this proposal, points of interests belonging to the road are going

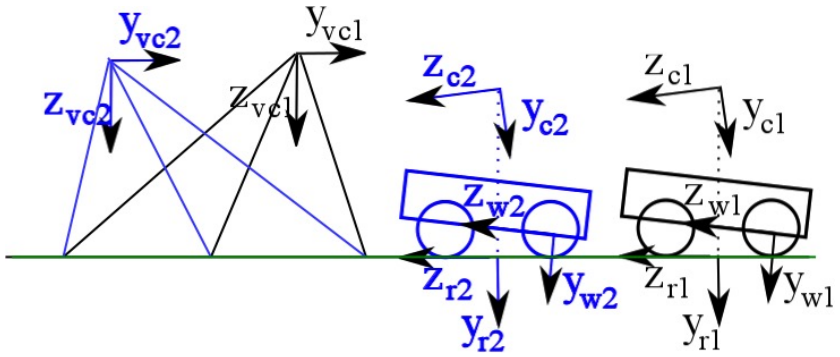


Figure 1: Different reference axes used in the algorithm (best view in color)

to be detected and matched, as they belong to the static part of the scene. In order to do so, first, the road ahead of the vehicle, which is assumed flat up to 20 meters, is obtained. Unlike other approaches that assume that only the yaw angle changes, estimations of the camera roll, pitch and height are obtained for every image. This way the extrinsic parameters of the stereo camera for every image are found. As the road is flat and the pose and orientation of the cameras are known, any virtual image of the road can be obtained. In this case, a virtual bird-view image, perpendicular to the road, where the features are going to be detected, is created. The POI are detected due to the texture of the road or street and to the presence of road markings available on the surface of the ground. Matching the features of two consecutive images, the relative movement of the vehicles is found. A Kalman filter improves the estimations and the Ackermann steering restriction is applied so the vehicle follows a constrained trajectory.

The rest of the paper describes the algorithm. The features are going to be detected in a virtual bird-view image. In order to do this, section 2 explains how the extrinsic parameters of the stereo camera are obtained for every image. Section 3 explains how the features are matched and the relative movement of consecutive images is found. The Kalman filter is explained in section 4 and the results in real driving situations are shown in section 5. Finally, the conclusions are presented.

The results are based on sequences of the KITTI Vision Benchmark Suite [18][19]. The stereo cameras for this benchmark were placed on the vehicle roof and parallel to the ground. Because of the camera placement, the minimum distance that the cameras capture is a bit far for this method, 6 meters. Because the presented algorithm looks for features on the road, the cameras

are not placed on the best place, on the vehicle's wind-shield and looking at the road. Another limitation is the resolution, 1344 by 391 pixels, so the images are a bit narrow. But as the sequences have a very good ground truth obtained from a centimeter GPS, they are very useful to show if the method is valid or not. Others specification of the cameras are the stereo baseline is 60 cm and the frame rate is 10 Hz.

2. Continuous Extrinsic Parameters Estimation

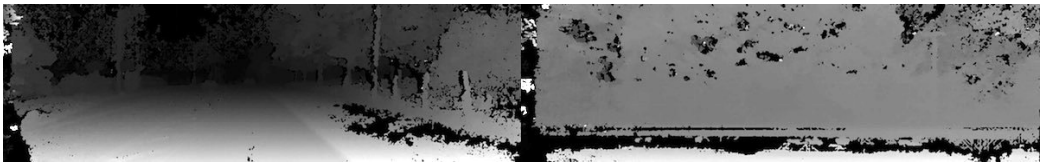
The first step of the algorithm is to find the extrinsic parameters of the stereo camera. Other approaches find the initial position of the cameras and assume that only the yaw angle changes. Although this is valid for several domains, it is not practical in urban applications due to the change in the extrinsic parameters because of the vehicle movements, the effect of the shock absorbers and the presence of uneven road surfaces. The road is assumed to be flat up to a near distance, 20 m., and the plane of the road is found. From the plane coefficients, the values of the pitch and roll angles and the height of the camera are obtained. Besides the application for visual odometry, finding the plane is also useful for other tasks of the vehicle as obstacle and driveable area detection.

2.1. Obtaining the 3D Point Cloud

As shown in Fig.2, the changes in illumination inside the images, the lack of texture in many objects and the presence of repetitive patterns in others are the three main problems in order to obtain 3D points from stereo images taken in urban environments. Accordingly, stereo local methods, although being fast, are not reliable enough and at least a semi-global method has to be used. A popular one in vehicle applications, which is used here, is [20]. Not all the provided 3D points are needed, as the information is going to be used to detect the plane of the road in front of the vehicle. So, from all the points in the 3D cloud, only those points between a minimum, 6 m., a maximum distance, 20 m., and within a certain width, 12 m., are taken into account. Moreover, they are normalized within a grid. So, a regular grid in the 3D space is created over the input 3D point cloud data and for each voxel all the points in it are approximated with their centroid. This way, although there are much more points per square meter in the nearest distances, there is not a bias towards them when the road plane is obtained.

(a)

(b)



(c)

Figure 2: Stereo images. (a) Left and (b) right images (c) disparity images

2.2. Stereo Extrinsic Parameters

As, can be see in Fig 1, the relationship between road, P_r, and image, P_c, coordinates is defined by a rotation matrix, R_{cr}, and translation vector, T_{cr}:

$$P_r = R_{cr}P_c + T_{cr} \tag{1}$$

If θ is the yaw, ψ the roll and ϕ the pitch of the camera, one of the Tait-Bryan matrices is:

$$R_{cr} = \begin{pmatrix} C\theta C\psi & S\theta S\phi - C\theta S\psi C\phi & S\theta C\phi + C\theta S\psi S\phi \\ S\psi & C\psi C\phi & -C\psi S\phi \\ -S\theta C\psi & C\theta S\phi + S\theta S\psi C\phi & C\theta C\phi - S\theta S\psi S\phi \end{pmatrix} \tag{2}$$

As the pixels belonging to the road have nil height and any yaw angle, equation (2) can be simplified to:

$$\begin{pmatrix} x_r \\ 0 \\ z_r \end{pmatrix} = \begin{pmatrix} C\psi & -S\psi C\phi & S\psi S\phi \\ S\psi & C\psi C\phi & -C\psi S\phi \\ 0 & S\phi & C\phi \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + \begin{pmatrix} 0 \\ h \\ 0 \end{pmatrix} \tag{3}$$

So, the plane equation is:

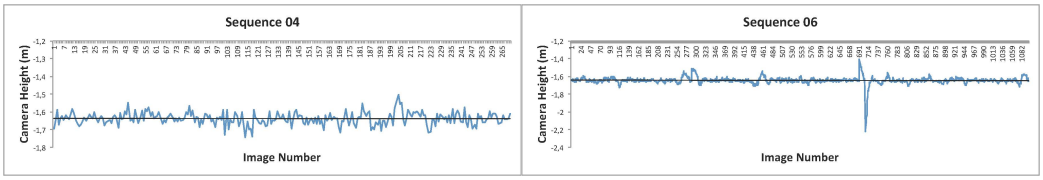
$$S\psi x_c + C\psi C\phi y_c - C\psi S\phi z_c + h = 0 \tag{4}$$

The road in front of the vehicle is assumed to be flat and is defined by the plane:

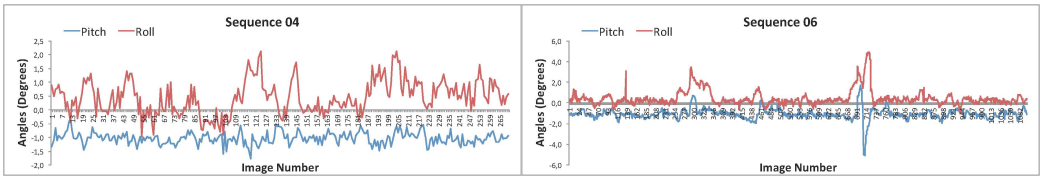
$$ax_c + by_c + cz_c + d = 0 \tag{5}$$

From the point cloud, the plane of the road is obtained with the Sample Consensus Model Perpendicular Plane method so the algorithm detects a plane perpendicular to an axis, in this case the vertical axis, within a maximum specified angular deviation [21]. Thus, the roll, pitch and height of the camera are calculated from the obtained plane:

$$\begin{gathered} \psi = asin(a) \\ \phi = atan(\frac{-c}{b}) \\ h = d \end{gathered} \tag{6}$$



(a)



(b)

Figure 3: Extrinsic parameters for two KITTI sequences (a) camera height (b) pitch and roll angles (best view in color)

The results from two sequences are shown in Fig. 3. In them the effect of the vehicle shock-absorbers is seen as well as some errors like the one in the 714th image of sequence 6. In Table 1, the median values for height, pitch and roll angles are shown. The nominal height of the camera is 1.65 m., which is very close to the median values. Unfortunately, there is no

Table 1: Median values for camera height, pitch and roll angles

Sequence	Height (m)	Pitch (°)	Roll (°)
3	-1.609	-0.418	0.335
4	-1,637	-1.020	0.441
6	-1.647	-1.112	0.258
7	-1.648	-1.091	0.384
10	-1.650	-1.094	-0.011

information about any pitch and roll offsets in the KITTI web page, where no ground truth is provided for these parameters.

2.3. Virtual Camera

A virtual camera with the same intrinsic parameters, K, as the stereo system is used in order to "acquire" the image where the features on the road are going to be detected (Fig. 1). This camera is looking perpendicular to the road and captures a defined area of it: the same area that it has been used for the road plane detection, 14 x 12 m. In order to obtain the homography, which relates both images, the relationship between virtual camera coordinates, p_{vc}, and camera coordinates, p_c, is needed. The pin-hole model is:

$$p_c = KP_c \tag{7}$$

and from equation (1)

$$P_r = R_{cr}K^{-1}p_c + T_{cr} \tag{8}$$

$$p_c = KR_{cr}^{-1}(P_r - T_{cr}) \tag{9}$$

Similarly, the virtual camera,vc:

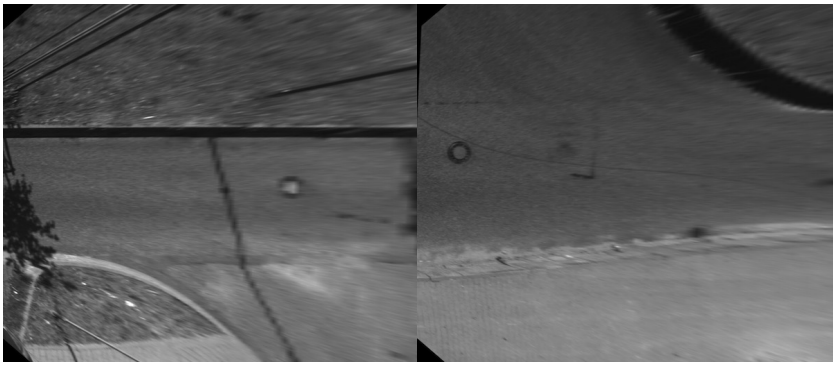
$$p_{vc} = KR_{vcr}^{-1}(P_r - T_{vcr}) \tag{10}$$

Therefore, the homography, H, is obtained using the projection on both images, obtained from equations (9) and (10), of four points on the road. In this application the four corners of the road perceived by the virtual camera.

$$p_{vc} = H p_c \tag{11}$$

Some examples can be seen in Fig. 4, where the camera image and the corresponding bird-view image are shown.

(a)



(b)

Figure 4: Birdview images. (a) Original images captured by the stereo camera (b) Bird-view images (rotated for better view)

3. Feature Detection and Matching

3.1. Road Features Detection

6000 SIFT features [22] are detected on the bird-view images. As a spatial uniform distribution of features is desired, the bird-view image is divided into 8x8 sub-images and an equal number of features is looked for in every sub-image. Some features correspond to textured areas of the road, but others are related to the projection of objects like cars, pedestrians, buildings, etc. Hence, whether the points belong to the ground, or not, has to be checked. From equation (11) the corresponding pixel $(u\, v)$, in the disparity image, of the feature, in the bird-view image, can be obtained:

$$p_c = (u\, v)^t = H^{-1} p_{vc} \tag{12}$$

Knowing the image coordinates, the disparity image, $ImaDisp$, and the intrinsic parameters, it is possible to obtain the real 3D coordinates of that point:

$$disp = ImaDisp(u, v) \tag{13}$$

$$z_c = f\frac{D}{disp} \tag{14}$$

$$x_c = z_c(u - u_0)/f \tag{15}$$

$$y_c = z_c(v - v_0)/f \tag{16}$$

where f is the focal length, D the stereo baseline, and (u_0, v_0) the image center. Finally, the distance to the plane detected before is obtained:

$$dist = |ax_c + by_c + cz_c + d| \tag{17}$$

If it is less than a threshold the feature belongs to a point on the ground and is kept, otherwise it is rejected. In Fig. 5, some examples can be seen. Green points represent features that belong to the road plane, the red ones do not belong to it and the white ones represent features where there is no stereo information available. The images show one of the advantages of detecting the features in the virtual bird-view image instead of using the real image. The features could have been detected in one of the stereo images and, using the disparity information and the plane equation, check if they belong to the ground or not. But they had not been uniformly 3D distributed but 2D, so more features close to the vehicle would have been taking into account and a bias would have been obtained. In Fig. 5 images, it can be seen the features are more grouped at farther distances.

3.2. First outlier removal by a speed filter

The features are detected for every image and a matching between features of two consecutive images is done. Many errors can be expected from the matching as many features are being taken into account and the road surface has a minimum texture to detect them and not much information for their description. Although the algorithm for detecting the vehicles displacement can deal with outliers, it will work better and faster if there are as minimum errors as possible. Therefore a previous filter is done now where:

Figure 5: Road features detection (best view in color)

- Those matched features whose module is higher than a threshold, 130 km/h, are rejected since the vehicle is not allowed to drive over that speed.
- From the rest, the median of their module is obtained and all the matched features with a speed below 75% and higher than 125% of the median are also not taken into account.

3.3. Camera Displacement

With the 3D information of the features belonging to the ground we could try to find the best rotation and translation according to the matches. This approach has the shortcoming that any error, although small, in the homography would produce a big error at the farthest points. That is why the projection of the features in the 2D images are going to be used. So, the camera displacement is obtained using a Perspective-n-Point (PnP) algorithm [23][24]. It estimates the camera displacement given a set of 3D points, P_{c2}, and their corresponding image projections, p_{c1}. It minimizes the re-projection error, that is, the sum of squared distances between the observed projections, p_{c1}, and the projected 3D points, P_{c2}:

$$p_{c1} = K(R|T)P_{c2} \tag{18}$$

RANSAC is used, in the implementation of the algorithm, in order to be robust against outliers. In Fig. 6, all the matched features between consecutive images can be seen. In blue are the ones discarded by the first

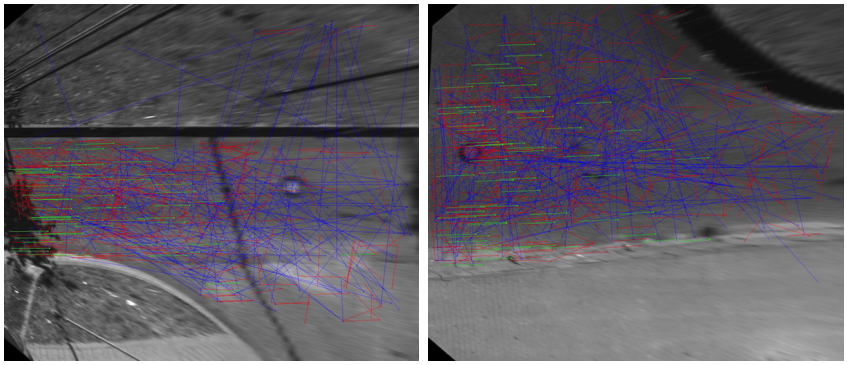


Figure 6: Matched features. In blue are the ones discharged by the first filter and in green the ones whose displacement is given by the algorithm and in red the ones considered as outliers (best view in color)

filter, in green the ones whose displacement is given by the algorithm and in red the ones considered as outliers.

From equation (18) the rotation and translation between the two camera poses are obtained. But if some dynamic restriction is going to be applied, the rotation and translation of the rear wheel axis is needed. The relationship between wheel, P_w, and image, P_c, coordinates is defined by a rotation matrix, R_{cw}, and a translation vector, T_{cw}:

$$P_w = R_{cw}P_c + T_{cw} \tag{19}$$

At time 2, the displacement of the camera between both positions is obtained and from the previous equation and equation (7) the rotation, R_w, and translation, T_w, between the wheel's center at both times are obtained:

$$P_{w1} = R_w P_{w2} + T_{w1} \tag{20}$$

$$T_w = (I - R_{cw}RR_{cw}^{-1})T_{cw} + R_{cw}T \tag{21}$$

$$R_w = R_{cw}RR_{cw}^{-1} \tag{22}$$

From equation (2) the three angles, pitch, roll and yaw, are:

$$\begin{aligned} \psi &= asin(R_w(1,0)) \\ \phi &= atan\left(\frac{-R_w(1,2)}{R_w(1,1)}\right) \\ \theta &= atan2\left(\frac{-R_w(2,0)}{cos\psi}, \frac{-R_w(0,0))}{cos\psi}\right) \end{aligned} \tag{23}$$

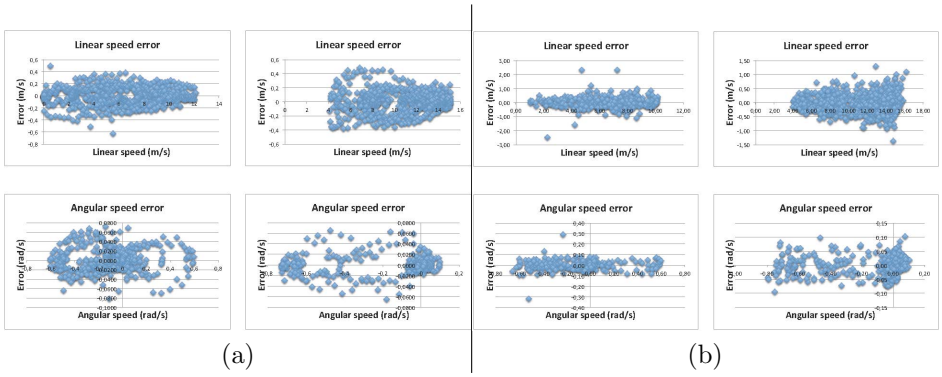


Figure 7: Errors for linear and angular speeds (a) from the ground-truth (b) from the sensor

4. Kalman Filter

4.1. Kalman equations

The state vector is formed by five variables $[v_k\, \omega_k\, \dot{z}_k\, \dot{\psi}_k\, \dot{\phi}_k]$, where v_k and ω_k are the linear and angular speed, $\dot{z}_k$ is the vertical speed and $\dot{\psi}_k$ and $\dot{\phi}_k$ the roll and yaw speed at instant k. The model applied here assume the linear and angular speeds are constant between consecutive images and the pitch, roll and height speeds are nil.

$$\begin{aligned} v_k &= v_{k-1} + r_1 \\ \omega_k &= \omega_{k-1} + r_2 \\ \dot{z}_k &= 0 + r_3 \\ \dot{\psi}_k &= 0 + r_4 \\ \dot{\phi}_k &= 0 + r_5 \end{aligned} \tag{24}$$

where r_1 to r_5 are the white noises whose variances have to be estimated. In order to do so, the information of the ground-truth provided by the KITTI database is used. The error made when the model equations (24) are used instead of the GPS information is obtained and the variances calculated. In order to obtain the sensor noise, a similar process is done using the output of (21) and (23) in some parts of several sequences where the sensor provides good results. The error plot for two sequences can be seen in Fig. 7 where on the left column, the linear and angular speed errors obtain from the ground-truth are shown and on the right column the errors obtain from the sensor are shown.

The measurements are obtained from the values calculated from (21) and (23):

$$\begin{aligned}
\widehat{v_k} &= \frac{\sqrt{T_{x,w} * T_{x,w} + T_{z,w} * T_{z,w}}}{\Delta t} \\
\widehat{\omega_k} &= \frac{\theta}{\Delta t} \\
\widehat{\dot{z}_k} &= \frac{T_{y,w}}{\Delta t} \\
\widehat{\dot{\psi}} &= \frac{\psi}{\Delta t} \\
\widehat{\dot{\phi}} &= \frac{\phi}{\Delta t}
\end{aligned} \tag{25}$$

The prediction is compared with the measurements and if the Normalized Innovation Squared is greater than a threshold, it means the sensor measurements are wrong, so the prediction is used as the new state. If not, the filter is updated.

4.2. Ackerman Constraint

Due to the Ackerman steering, the vehicle trajectory is a circumference (Fig. 8), so some cinematic restrictions can be applied. What is the relationship between the rotation and translation obtained from the camera and the vehicle movement? Applying trigonometric rules, it can be deduced that is:

$$T = \begin{pmatrix} \rho sin(\theta/2) \\ \rho \cos(\theta/2) \end{pmatrix} \tag{26}$$

when ρ is the displacement. So, after applying the Kalman filter, where the linear and angular speeds are calculated, the displacement and yaw increment are obtained:

$$\begin{aligned}
\rho &= v_k \Delta t \\
\theta &= \omega_k \Delta t
\end{aligned} \tag{27}$$

Finally, the camera displacement is obtained from (26).

5. Results

Several results have been obtained in order to test the ideas presented in this paper. In order to check how good the speed estimation is, the

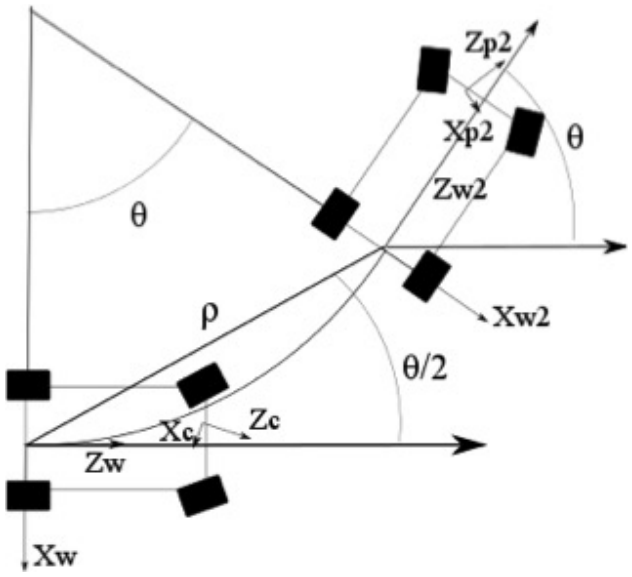


Figure 8: Ackerman constraints

Root Mean Square Error (RMSE) has been obtained for five sequences. The formula is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(f(i) - f_{gt}(i))^2}{n}} \tag{28}$$

where $f(i)$ is the value of the linear or angular speed obtained from the Kalman Filter, $f_{gt}(i)$ is the same information from the ground truth and n the number of images in the sequences. The results can be seen in Table 2. The errors are between 15 and 34 cm/s for the linear speed and between 9 and 38 10^{-3} rad/s for the angular speed. Graphical results can be seen in Fig. 9a and 9b where the ground-truth, the output of the Kalman filter and the result of the PnP algorithm are shown. It shows how the filter follows the real speed and yaw of the vehicle despite the occasional errors of the measurements.

In order to check if the proposed method is able to follow the real trajectory of the vehicle, in Table 3 the final errors for five sequences are shown. There are some trajectories, like sequence 3 and 4 formed by a few hundred images and around half a kilometer length. Others have more than one thousand images and the vehicle drove more than one kilometer like during sequence 6. The percentage errors are below 1% in all the sequences except

Table 2: RMSE errors for linear and angular speeds

Sequence	Num. Ima.	Linear speed (m/s)	Angular speed (rad/s)
3	800	0.181	0.024
4	270	0.151	0.009
6	1100	0.301	0.027
7	1100	0.337	0.038
10	1200	0.271	0.015

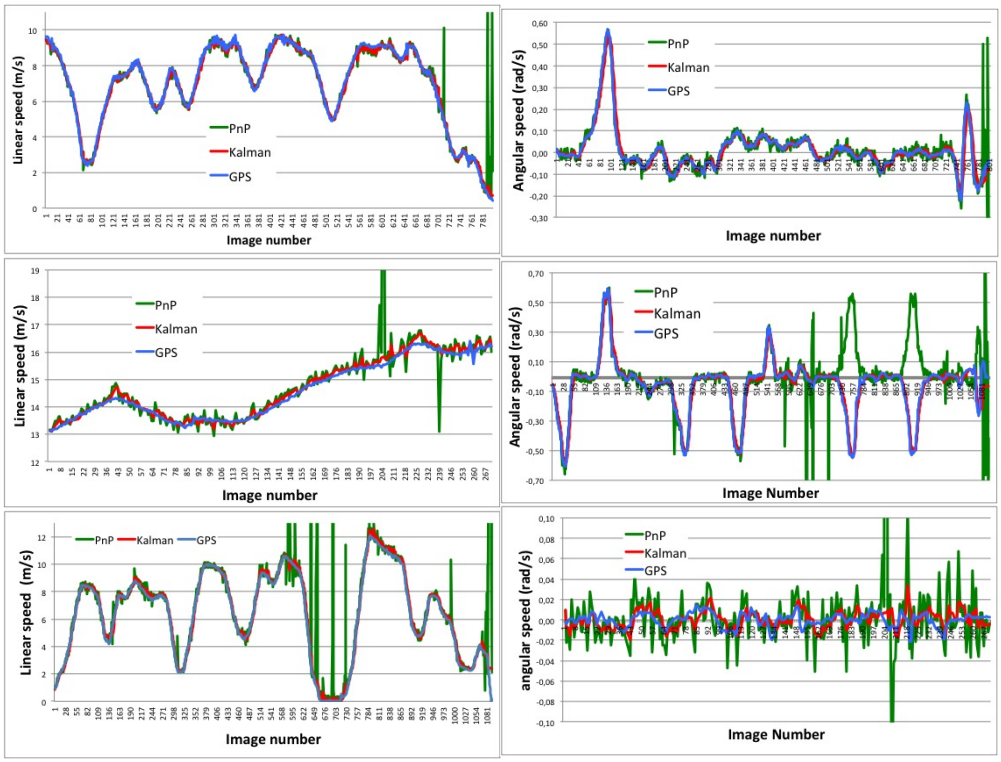
Table 3: Trajectory errors

Seq.	Dist. (m)	Error (m)	Error (%)	Error (deg)	Error (deg/m)
3	558.9	3.5	0.6	1.58	0.0028
4	393.6	3.2	0.8	0.61	0,0016
6	1229.4	8.7	0.7	2.0	0,0016
7	694.4	8.8	1.3	9,16	0.0074
10	917.8	7.7	0.8	0.74	0.0008

sequence 7 where is 1.3%. Good results are obtained for the angular error too. The worst is, again, sequence 7 with an error of 0.0074 deg/m but the others are much lower. The plot of the five trajectories is shown in Fig. 10. It shows the system estimates the real path followed by the vehicle with good accuracy.

6. Conclusions

A new method for VO using stereo vision is proposed, which detects and tracks features available on the surface of the ground. This way, it is assured only static points are taking into account in order to obtain the relative movement between images. The use of a a virtual bird-image assure an uniform 3D distribution of the features. A Kalman filter improves the estimations and the Ackermann steering restrictions is applied so the vehicle follows a constrained trajectory. The results in real urban environments show the algorithm is able to estimate the linear and angular speeds of the vehicle with high accuracy. Although VO is different than SLAM (Simultaneous Localization And Mapping), the results show its ability to follow the real



(a) Linear Speed profile

(b) Angular speed profile

Figure 9: Angular speed profile (best view in color)

trajectory drove by the vehicle along long paths with a minimum linear and angular error.

7. Biography

[1] D. Martin, F. García, B. Musleh, D. Olmeda, G. Pelaez, P. Marin, A. Ponz, C. Rodriguez, A. Al-Kaff, A. de la Escalera, J. M. Armingol, Ivvi 2.0: An intelligent vehicle based on computational perception, Expert Systems With Applications 41 (2014) 7927–7944.

[2] D. Nister, O. Naroditsky, J. Bergen, Visual odometry, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, pp. 652 – 659.

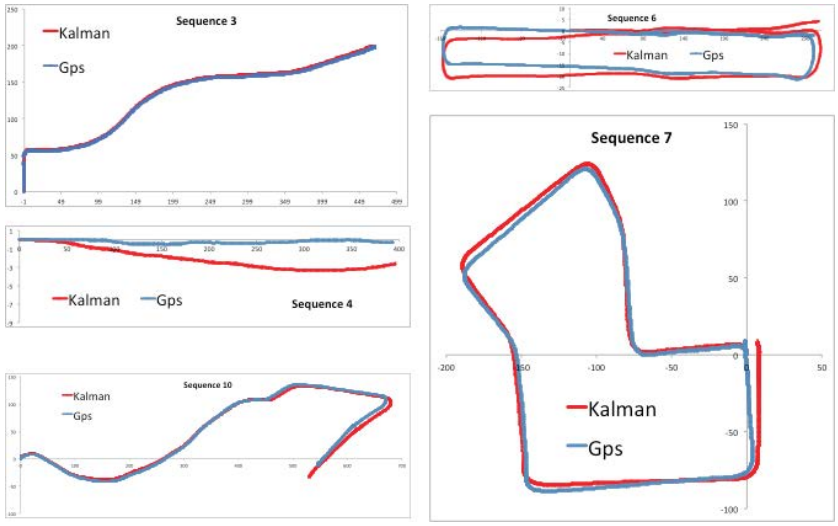


Figure 10: Trajectory followed by the vehicle along six trajectories (best view in color)

[3] D. Scaramuzza, F. Fraundorfer, Visual odometry part i: The first 30 years and fundamentals, IEEE Robotics and Automation Magazine 18 (4) (2011) 80–92.

[4] F. Fraundorfer, D. Scaramuzza, Visual odometry part ii: Matching, robustness, optimization, and applications, IEEE Robotics and Automation Magazine 19 (2) (2012) 78 – 90.

[5] H. Badino, A. Yamamoto, T. Kanade, Visual odometry by multi-frame feature integration, in: IEEE International Conference on Computer Vision Workshops, 2013, pp. 222 –229.

[6] W. Lu, Z. Xiang, J. Liu, High-performance visual odometry with two-stage local binocular high-performance visual odometry with two-stage local binocular ba and gpu, in: IEEE Intelligent Vehicles Symposium, 2013, pp. 1107 – 1112.

[7] M. Fischler, R. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Comm. of the ACM 24 (1981) 381–395.

[8] F. Bellavia, M. Fanfani, F. Pazzaglia, C. Colombo, Robust selective

stereo slam without loop closure and bundle adjustment, in: Image Analysis and Processing – ICIAP, 2013, pp. 462–471.

[9] M. Sanfourche, V. Vittori, G. L. Besnerais, evo: a realtime embedded stereo odometry for mav applications, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 2107 – 2114.

[10] M. Persson, T. Piccini, M. Felsberg, R. Mester, Robust stereo visual odometry from monocular techniques, in: IEEE Intelligent Vehicles Symposium, 2015, pp. 686 – 691.

[11] E. Rosten, T. Drummond, Machine learning for highspeed corner detection, in: European Conference on Computer Vision, 2006, pp. 430–443.

[12] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, P. Fua, Brief: Computing a local binary descriptor very fast, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (7) (2012) 1281–1298.

[13] M. Bosse, R. Zlot, Map matching and data association for large-scale two-dimensional laser scan-based slam, The International Journal of Robotics Research 27 (2008) 667–691.

[14] J. Almeida, V. M. Santos, Real time egomotion of a nonholonomic vehicle using lidar measurements, Journal of Field Robotics 30 (2013) 129–141.

[15] J. Zhang, S. Singh, Loam: Lidar odometry and mapping in real-time, in: Robotics: Science and Systems Conference (RSS), Berkeley, CA, 2014.

[16] J. Zhang, S. Singh., Visual-lidar odometry and mapping: Low- rift, robust, and fast, in: IEEE International Conference on Robotics and Automation(ICRA) 2015, Seattle, WA, 2015.

[17] A. de la Escalera, E. Izquierdo, D. Martin, F. Garcia, J. Armingol, Stereo visual odometry for urban vehicles using ground features, in: ROBOT'2015 - Second Iberian Robotics Conference, 2015.

[18] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3354 – 3361.

[19] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, International Journal of Robotics Research (IJRR) 32 (11) (2013) 1231–1237.

[20] H. Hirschmuller, Stereo processing by semiglobal matching and mutual information, IEEE T on PAMI 30 (2008) 328–341.

[21] R. B. Rusu, S. Cousins, 3d is here: Point cloud library (pcl), in: IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011.
URL `pointclouds.org`

[22] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

[23] J. Hesch, S. Roumeliotis, A direct least-squares (dls) method for pnp, in: IEEE International Conference on Computer Vision, 2011, pp. 383–390.

[24] G. Bradski, The opencv library, Dr. Dobb's Journal of Software Tools 25 (11) (2000) 120–125.

Arturo de la Escalera

Arturo de la Escalera, PhD, graduated from Universidad Politecnica de Madrid (Madrid, Spain) in Automation and Electronics Engineering in 1989, where he also obtained his Ph.D. degree in Robotics in 1995. In 1993, he joined the Department of Systems Engineering and Automation at Universidad Carlos III de Madrid (Madrid, Spain), where he became an Associate Professor in 1997. Since 2005, Arturo de la Escalera is the head of the Intelligent Systems Lab (LSI) and he is an Assistant Director of the Engineering School of Universidad Carlos III de Madrid since May 2012.

His current research interests include Advanced Robotics and Intelligent Transportation Systems; with special emphasis on Vision Sensor Systems and Image Data Processing methods for environment perception and real-time pattern recognition.

He has supervised nine PhDs related with these topics, at which four of them have received the award for Best University PhD Award. He co-authored 33 articles in journals and nearly 80 papers in international conferences. He wrote the book "Visión por Computador: fundamentos y métodos", which was published in 2001 by Pentice Hall, one of the few books about this topic written in Spanish. Up to today, he worked in 19 public funded research projects, at which he lead 7 of them. Besides, he worked on 17 projects in collaboration with private companies. Currently, he is coordinator of the Spanish Computer Vision Group at the Comité Español de Automática. He is member of the Editorial Boards of the International Journal of Advanced Robotic Systems: Robot Sensors (Intech), the International Journal of Information and Communication Technology (InderScience Publisher), the Open Transportation Journal (Bentham Science Publishers), the Scientific World Journal: Computer Science (Hindawi Publishing Corporation) and the Revista Iberoamericana de Automática e Informática Industrial.

He spent a Sabbatical Year researching at Queen Mary University London from October 2014 to February 2015.

Ebroul Izquierdo

Ebroul Izquierdo received the M.Sc., C.Eng., Ph.D., and Dr. Rerum Naturalium (Ph.D.) degrees from Humboldt University, Berlin, Germany. He is currently the Chair of the Multimedia and Computer Vision, and the Head of the Multimedia and Vision Group with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. He has been a Senior Researcher with Heinrich Hertz Institute for Communication Technology, Berlin, and the Department of Electronic Systems Engineering, University of Essex, Colchester, U.K. He has authored over 450 technical papers, including chapters in books, and holds several patents in multimedia signal processing. Prof. Izquierdo is a Chartered Engineer, a Fellow Member of the Institution of Engineering and Technology (IET), a member of the British Machine Vision Association, the Chairman of the IET Professional Network on Information Engineering, a member of the Visual Signal Processing and Communication Technical Committee of the IEEE Circuits and Systems Society, and a member of the Multimedia Signal Processing Technical Committee of the IEEE. He has been an Associated and Guest Editor of several relevant journals, including IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, EURASIP Journal on Image and Video processing, Signal Processing: Image Communication (Elsevier), EURASIP Journal

on Applied Signal Processing, IEEE PROCEEDINGS ON VISION, IMAGE AND SIGNAL PROCESSING, Journal of Multimedia Tools and Applications, and Journal of Multimedia.

David Martín
David Martín, graduated in Industrial Physics (Automation) from the National University of Distance Education (UNED, 2002) and Ph.D. degree in Computer Science from the Spanish Council for Scientific Research (CSIC) and UNED, Spain 2008. He was Ph.D. student at CSIC from 2002 to 2006. He was fellow at the European Organization for Nuclear Research (CERN, Switzerland, 2006-2008) and Post-Doc researcher in Robotics at CSIC (2008-2011). Currently, he is Professor and Post-Doc researcher at Carlos III University of Madrid and member of the Intelligent Systems Lab since 2011. His research interests are Real-time Perception Systems, Computer Vision, Sensor Fusion, Intelligent Transportation Systems, Advanced Driver Assistance Systems, Autonomous Ground Vehicles, Unmanned Aerial Vehicles, and Vehicle Positioning and Navigation. He participates in several industrial research projects, and is reviewer of prestigious Journals, and member of the Spanish Computer Vision Group, among others. In 2014, he was awarded with the VII Barreiros Foundation award to the best research in the automotive field. In 2015, the IEEE Society has awarded Dr. Martín as the best reviewer of the 18th IEEE International Conference on Intelligent Transportation Systems.

Basam Musleh
Basam Musleh obtained the M.Sc. Degree in Industrial Engineering in 2007 and in Robotics and Automation in 2009 from the University Carlos III of Madrid. He received the Ph.D. Degree in Electric, Electronic and Automation from the same University in 2015. He is Assistant Professor in Perception Systems, Control Engineering and Real Time Systems at Department of Systems Engineering and Automation at University Carlos III of Madrid since 2010. His current research interest is in Computer Vision, Intelligent Transportation Systems, Autonomous Vehicles and Machine Learning, in which he has numerous publications.

Fernando Garcia
F.Garcia received the Eng. degree in Telecommunications in 2007, M.S. in Robotics and Automatics in 2007 and, PhD in Robotics in 2012, all in Universidad Carlos III de Madrid.
He is with Intelligent System Lab since 2007. Since 2009, he has been an Assistant Professor with the Systems and Automatics department in Universidad Carlos III de Madrid. His research interests include data fusion, computer vision, intelligent vehicles and human factors in intelligent vehicles.
Dr. Garcia was a recipient of the VII Barreiros Foundation award to the best research in the automotive field, he was also awarded with the 2nd award to the best PhD dissertation 2013-2015, by the Spanish chapter ITSS.

José María Armingol
Professor of Robotics and Automation at Intelligent Systems Laboratory (Carlos III University). José María Armingol received his Ph.D. in Automation from the Universidad Carlos III of Madrid in 1997. His research interest focus on Computer Vision, Image Processing and Real-Time Systems applied to Intelligent

Transportation Systems. He is member of the Editorial Board of the ISNR Robotics, Journal of Physical Agents and Securitas Vialis.


Arturo de la Escalera


Ebroul Izquierdo


David Martín


Basam Musleh


Fernando García

José María Armingol