Van Hamme *et al. EURASIP Journal on Image and Video Processing* (2015) 2015:10
DOI 10.1186/s13640-015-0065-6

EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Robust monocular visual odometry for road vehicles using uncertain perspective projection

David Van Hamme[1*], Werner Goeman[2], Peter Veelaert[1] and Wilfried Philips[1]

Abstract

Many emerging applications in the field of assisted and autonomous driving rely on accurate position information. Satellite-based positioning is not always sufficiently reliable and accurate for these tasks. Visual odometry can provide a solution to some of these shortcomings. Current systems mainly focus on the use of stereo cameras, which are impractical for large-scale application in consumer vehicles due to their reliance on accurate calibration. Existing monocular solutions on the other hand have significantly lower accuracy. In this paper, we present a novel monocular visual odometry method based on the robust tracking of features in the ground plane. The key concepts behind the method are the modeling of the uncertainty associated with the inverse perspective projection of image features and a parameter space voting scheme to find a consensus on the vehicle state among tracked features. Our approach differs from traditional visual odometry methods by applying 2D scene and motion constraints at the lowest level instead of solving for the 3D pose change. Evaluation both on the public KITTI benchmark and our own dataset show that this is a viable approach for visual odometry which outperforms basic 3D pose estimation due to the exploitation of the largely planar structure of road environments.

Keywords: Visual odometry; Localization; Computer vision; SLAM

1 Introduction

Visual odometry is an increasingly important research domain in the field of intelligent transportation systems. Many emerging and existing applications related to consumer vehicles rely on accurate position estimation. Some examples of these applications are navigation, lane assistance, collision warning, and avoidance. Traditionally, the positioning data for these applications is provided by satellite-based systems such as GPS, GLONASS, or GALILEO, sometimes augmented by closer-range communication (as in DGPS) or additional sensors scanning the local environment (e.g., a lane assist camera). However, the reliance of these applications on satellite navigation is a threat to their full-time availability. Due to the four-dimensional nature of the problem (3D positioning and time synchronization), signals from at least four satellites must be received in order to obtain a positional fix. It is well documented that in certain urban scenarios, large parts of the sky can be obscured by buildings or road infrastructure, making the reception of four satellites unlikely or even impossible for many seconds or even minutes [1]. In these cases, satellite navigation systems cannot provide reliable position estimates.

In this context of assisted or autonomous driving, positioning solutions complementary to the satellite-based systems are needed. One such solution is visual odometry: the measurement of a vehicle's trajectory using vehicle-mounted cameras. Visual odometry is closely related to simultaneous localisation and mapping (SLAM) in the field of robotics, but there are clear distinctions between the two. Whereas SLAM places equal emphasis on constructing a virtual map of the unknown environment as on positioning relative to that environment, visual odometry methods do not need to explicitly map the environment. The two problems remain strongly intertwined, as positioning relies on finding fixed points in the environment of the vehicle, but for visual odometry, the mapping itself

*Correspondence: dvhamme@telin.ugent.be
[1] Ghent University - IPI/iMinds, St-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
Full list of author information is available at the end of the article

is of little interest. In fact, in some cases (especially consumer automotive applications), some information about the environment may already be known (e.g., the local road network layout).

Visual odometry only provides *relative positioning*, i.e., positioning relative to an earlier visited reference point. As a consequence, estimation errors are cumulative, and visual odometry methods are therefore susceptible to *drift*. The greater the distance traveled from the last absolute reference point (e.g., the known GPS coordinates of the starting address), the greater the positional error can become. While this may appear to limit the use of visual odometry to a short distances, the drift error can be bounded by combination with additional passive sensors (e.g., a magnetic compass for dead reckoning) or *a priori* known information (e.g., the local road map) [2,3]. As such, visual odometry is still a prime candidate to supplement satellite navigation even for urban scenarios where signal reception may be unreliable for prolonged distances.

In the classical approach, visual odometry is a pose estimation problem in a calibrated setting. Given a camera with known intrinsic calibration parameters and the images of a scene captured from two unknown viewpoints, what is the relative camera pose between the two viewpoints? In this calibrated setting, visual odometry is achieved by estimating the *essential matrix* that relates the homogeneous image coordinates of the same world point in the two viewpoints up to a scale factor [4]. A computationally efficient solution for this was published in the 1990s by Philip [5] and improved upon by Nistér [6]. In Nistérs solution, a RANSAC algorithm evaluates sets of five correspondence points to find the best estimate for the essential matrix. This type of method is therefore called a *five-point solver*. The RANSAC algorithm is necessary to cope with outliers that will arise from erroneous feature matching and external circumstances (e.g., other traffic). The essential matrix can be decomposed into its rotation and translation components if necessary.

As a generalization of the classical setting, pose estimation can also be performed for uncalibrated cameras. In this case, the matrix that relates the image coordinates of the two viewpoints is called the *fundamental matrix* [7]. It is estimated in a similar way to the essential matrix; however, more point correspondences are necessary. Methods of this type are called *eight-point solvers*. Even in the calibrated setting, there is merit in using an eight-point solver as it yields only one solution, while the five-point methods can produce up to ten valid solutions, requiring additional constraints to be evaluated.

The aforementioned methods for estimating the essential or fundamental matrix are affected by the problem of degenerate configurations. Two distinct cases of degeneracy arise: degeneracy in the motion, where the camera undergoes only rotation and little or no translation, and degeneracy in scene structure, where all or many of the points are coplanar. In both cases, the accuracy of the pose estimation will be severely degraded [8]. This is an important drawback in real-world applications, where vehicles will often make small incremental motions and where the majority of the scene can consist of objects in or close to the ground plane. To remedy the problems of degeneracy, a stereo camera configuration is typically used, which allows for much better triangulation of the feature points even in the cases of motion or scene degeneracy.

Alternatives to fundamental matrix estimation for stereo systems have also been proposed based on triangulation through stereo disparity [9,10]. Typically, this class of algorithms first estimates approximate 3D coordinates from a stereo image pair and then links up feature tracks over multiple pairs to estimate camera motion.

Stereo camera setups however have significant downsides for consumer automotive applications. They are more expensive than a single-camera system and are more difficult to integrate into the car's design. Additionally, they rely on very accurate calibration on account of the long observation distance to baseline width ratio [11]. In the vibration and shock-prone environment of a car, it is generally accepted that long-term calibration stability cannot be guaranteed, and online recalibration methods have been proposed [12,13] in an effort to improve the applicability. Monocular solutions are inherently less susceptible to calibration drift, as fewer assumptions about the capture system's geometry are made.

Monocular visual odometry algorithms that do not employ fundamental matrix estimation and are therefore not impacted by the aforementioned degeneracies have been proposed by Tardiff *et al.* [14] and Scaramuzza [15,16]. However, these methods are only demonstrated using an omnidirectional camera mounted atop the vehicle, which is not practical for application on consumer vehicles. More relevant is the work of Chandraker and Song [17]. In this work, a five-point solver provides an initial triangulation of image points captured over five frames, after which new points are mapped to the known 3D structure and allow for four-point pose estimation. The output of the pose estimation is combined with continuous ground plane estimation in a data fusion framework, providing high accuracy as well as being unaffected by planar scene degeneracy. This proves the merit of combining different visual cues to improve the overall odometry accuracy. We expect this data fusion approach to be applied on other base odometry algorithms as well in the future.

Recently, a different approach to monocular visual odometry has emerged in literature, called *direct* or sometimes *dense* visual odometry. Instead of determining feature correspondences, these methods aim to recover the

camera pose directly from the image data, by reconstructing a surface-based depth map for the image. While this approach is not new, only recently has it become tractable for real-time applications [18-21]. These methods perform very well for structure-rich indoor and outdoor environments, but to the best of our knowledge, their accuracy in sparsely structured open road scenes is yet to be examined.

In this work, we will present and evaluate a monocular visual odometry method that does not depend strongly on accurate camera calibration and does not suffer from degeneracy in case of small incremental motion or planar scene geometry. Furthermore, the method is suitable for any standard camera that views part of the road surface in front of or behind the vehicle. This is compatible with normal camera placement for other currently emerging automotive vision applications such as traffic sign recognition and obstacle detection. The method tracks ground plane features, taking into account the uncertainty of the camera viewing angle with relation to the ground plane. This allows us to exploit the inherently two-dimensional character of vehicle motion while still retaining some of the accuracy benefits of a fully three-dimensional approach. Additionally, the use of uncertainty margins relaxes the requirement of accurate camera calibration.

Two key components of the method provide robustness against the common problem of outliers: a feature matching method constrained by uncertainty zones and a Hough-like parameter space vote. The combination of these two mechanisms eliminates the need for a RANSAC scheme and speeds up computation, while still producing useful odometry for inlier ratios as low as 1:8 in real-world experiments.

This work is a continuation of the concept first introduced in our publication at IV2011 [22] and tested in a real-world scenario at ITSC2012 [23]. The contributions of this paper in addition to the prior work are an extended literature review comparing the different approaches to visual odometry and their relative merits, a proper analysis and justification of the proposed method's underlying assumptions about vehicle dynamics, evaluation on two extended datasets, comparison against a reference method, qualitative assessment of the method's main benefits, calibration sensitivity analysis, and quantification of the effect of non-planarity of the road surface.

The proposed method is shown to produce reliable visual odometry even for longer trajectories of several kilometers, and its accuracy compares favorably to the monocular instance of the eight-point solver of Geiger et al. [24], both on the public KITTI dataset [25] and on a 15-km dataset captured locally with the GrontMij mobile mapping vehicle. This proves that approaching visual odometry as a two-dimensional problem from the bottom up not only offers practical benefits with relation to robustness, execution speed, and calibration but also provides accuracy competitive with the traditional 3D pose estimation approach.

A detailed description of the method is given in Section 2. Details about the calibration procedure and sensitivity simulations are in Section 3. Experimental validation is provided in Section 4, with a discussion of the results in Section 5. Finally, conclusions about the viability of this type of monocular visual odometry are drawn in Section 6.

2 Algorithm description

At the core of the proposed method is the tracking of feature points in the world ground plane surrounding the vehicle. We will perform this tracking not in image coordinates of the perspective camera but in ground plane coordinates. This is advantageous as consistency of motion among features is much easier to assess in the latter. In this respect, the method has similarities to the work of Scaramuzza [15]. An overview of the proposed method is shown in Figure 1.

The general structure of the method bears some resemblance to a Kalman filter in the sense that it uses a prior vehicle state estimate to predict current feature locations and then compares this prediction to current observations to calculate an updated vehicle state. However, because the accuracy of an observed feature depends strongly and nonlinearly on its position, novel strategies for prediction and update are implemented specific to this application.

In the prediction step, the previously estimated steering angle and velocity of the vehicle are used to define search regions in the ground plane where previously observed

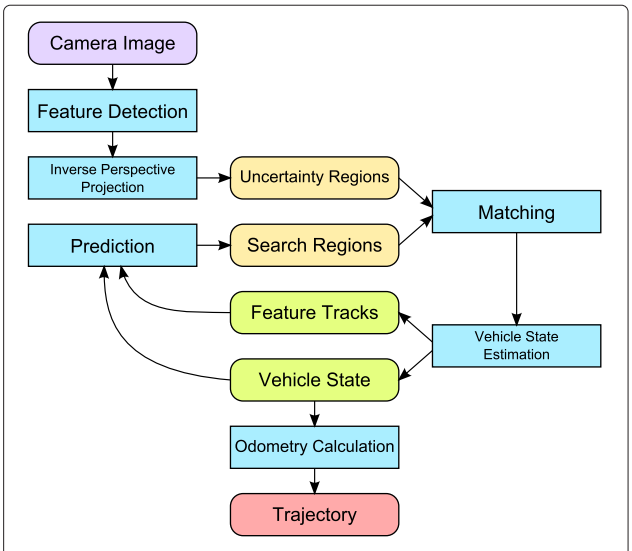

Figure 1 Overview of the proposed method. Rectangles denote procedures, rounded rectangles denote data. The only input is a stream of camera images.

features may be found. The details of this prediction will be explained in Section 2.2.

The observation step is more complex, since we cannot measure ground plane coordinates directly using the perspective camera. In order to relate the image coordinates of features in the camera view to their ground plane positions, inverse perspective projection is performed. This inverse perspective projection is only determined for features originating from a plane with a known orientation relative to the camera. In other words, we can only determine the inverse perspective projection if we know the viewing angle of the camera to the ground plane. This angle, however, is not static. The suspension of the vehicle creates variability in the camera pose, and this translates to uncertainty on the inverse perspective transform. We will take into account this uncertainty and define plausible regions of ground plane coordinates for each feature point. This process will be explained in more detail in Section 2.1.

In the update step, the predictions of feature locations are compared to the observations. In our method, this corresponds to matching the predicted search regions for previously seen features with the uncertainty regions pertaining to the inverse perspective projection of currently seen features. From these matches, a consensus is drawn to update the vehicle state. In this match-and-update step, several mechanisms will ensure robustness to outliers in the input data. This is explained in Section 2.3.

Finally, the vehicle trajectory can be calculated from the consecutive vehicle states.

2.1 Inverse perspective projection

To describe the inverse perspective projection, we first need to define the forward perspective projection that describes the image capturing process. Much of this section follows the standard model for the projective camera as described by Hartley and Zissermann [7]. This model describes a transformation from the 3D world axes to the 2D image axes of the captured video frame. This transformation consists of two steps.

In the first step, the 3D world coordinates are transformed into the 3D camera coordinate system. These axes are defined as shown in Figure 2. The 3D camera coordinate system has its origin in the center of projection of the camera. The 3D world axes are affixed to the vehicle. Let $\mathbf{X} = [X\ Y\ Z\ 1]^T$ denote the homogeneous coordinates of a point in world axes. The corresponding 3D coordinates in the camera axes $\mathbf{X}' = \left[X'\ Y'\ Z'\right]^T$ are then given by:

$$\mathbf{X}' = [\mathbf{R}|\mathbf{t}]\,\mathbf{X} \tag{1}$$

in which $[\mathbf{R}|\mathbf{t}]$ is the rotation matrix $\mathbf{R}$ that aligns the world axes with the camera axes, augmented by the 3D translation vector $\mathbf{t}$ between their origins.

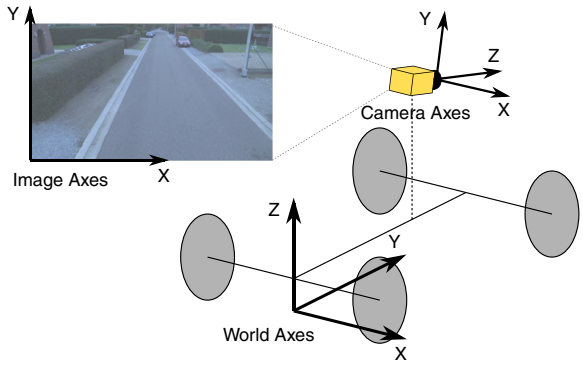

Figure 2 Illustration of camera, world, and image axes. 3D camera axes have the origin in center of projection of the camera. 3D world axes have the origin in the ground plane below the center of the rear axle of the vehicle.

By affixing the 3D world axes to the vehicle, the matrix $[\mathbf{R}|\mathbf{t}]$ is made independent of vehicle position, and $\mathbf{R}$ and $\mathbf{t}$ can be determined by extrinsic calibration (e.g., using Zhang [26] or Miksch et al. [13]). Vehicle motion will now manifest itself as a change in coordinates of the feature points corresponding to static objects in the real world.

The second step in the transformation from 3D world to 2D image coordinates is the camera projection itself. In this step, the 3D camera coordinates are projected through the focal point onto the sensor image plane. For a pinhole camera model, the projection of a point $\mathbf{X}' = \left[X'\ Y'\ Z'\right]^T$ in 3D camera axes onto its homogeneous 2D image coordinates $\mathbf{x} = \left[x\ y\ 1\right]^T$ is given by:

$$w\mathbf{x} = \mathbf{C}\mathbf{X}' \tag{2}$$

in which

$$\mathbf{C} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

is the projection matrix consisting of the horizontal and vertical focal lengths α_x and α_y and the coordinates of the principal point (i.e., the projection of points on the Z axis) (x_0, y_0). These intrinsic parameters can easily be estimated using a standard camera calibration method (e.g., Bouguet [27]). The factor w serves to compress the 3D space onto the sensor plane by scaling the X and Y coordinates by the inverse of the Z coordinate.

To summarize, the complete perspective projection from 3D world coordinates to 2D image coordinates is given by:

$$w\mathbf{x} = \mathbf{C}[\,\mathbf{R}|\mathbf{t}]\,\mathbf{X}. \tag{4}$$

In the special case where all feature points are constricted to the ground plane ($Z = 0$), the transform can be reduced to:

$$w\mathbf{x} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \mathbf{C}\,[\mathbf{R_{XY}}|\mathbf{t}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{5}$$

in which $\mathbf{R_{XY}}$ is the submatrix of $\mathbf{R}$ obtained by omitting the third column. The inverse perspective projection that maps homogeneous image coordinates back onto homogeneous 2D world ground plane coordinates is then the inverse of $\mathbf{C}\,[\mathbf{R_{XY}}|\mathbf{t}]$. Inverse perspective projection is sometimes also referred to as *backprojection*.

An important remark with respect to backprojection is that the calculated transform is only valid for features corresponding to objects in the ground plane. However, there is no easy way to discern the Z coordinate of a feature in the camera view. We therefore have no choice but to apply the backprojection to any features we detect in the camera image and sort out the above-ground features in a higher level reasoning step.

Throughout this discourse, we assumed the matrix $[\mathbf{R}|\mathbf{t}]$ to be known from extrinsic calibration. In practice, however, the camera coordinate system is not rigidly affixed to the axles of the vehicle. Instead, the camera is attached to the body of the vehicle, which has a variable pose with relation to the axles on account of the suspension travel. The matrix $[\mathbf{R}|\mathbf{t}]$ is therefore no longer static but changes somewhat as the vehicle moves along. It is important to take this pose variation into account as it directly affects the estimated ground plane coordinates of all features.

Since we have no reliable way of measuring the attitude of the vehicle, we model the uncertainty on the inverse perspective projection arising from this attitude. To this end, we determine realistic limits on the suspension travel during normal driving and calculate the inverse perspective transforms corresponding to these limits. This yields a region of possible ground plane coordinates for each feature detected in the perspective view.

A typical road vehicle experiences in the range of 100 to 150 mm total suspension travel measured at each wheel. With a track width of 1.4 to 1.5 m, this could theoretically give rise to approximately 10° of lateral roll. Considering a wheelbase of 2.7 m on average, the maximum pitch is approximately 5°. However, these limits would be very hard to achieve in practice even with extremely aggressive driving, as the vehicle will tend to break traction first. In typical town driving, more representative values for maximum roll and pitch are respectively 2° and 1° either side of the level position. For highway driving, the expected angles are even smaller.

The rotation matrix $\mathbf{R}$ can be considered approximately linear in these roll and pitch angles due to the small range of possible angles. This means that the region of possible ground plane coordinates of a feature is also approximately convex, and it is sufficient to evaluate the four extremal backprojections to delimit this region. We will call these regions of possible ground plane coordinates *observation uncertainty regions* as each region represents the limits on the uncertain position of one of the observed features. Examples of observation uncertainty regions are shown in Figure 3. Note that for this typical camera position, the observation uncertainty regions rapidly become more elongated for more distant features, as pitch is the major contributor to the uncertainty at distance. It can therefore be beneficial to exclude features that are too distant from the camera, as their larger associated uncertainty may render them less useful for calculating odometry.

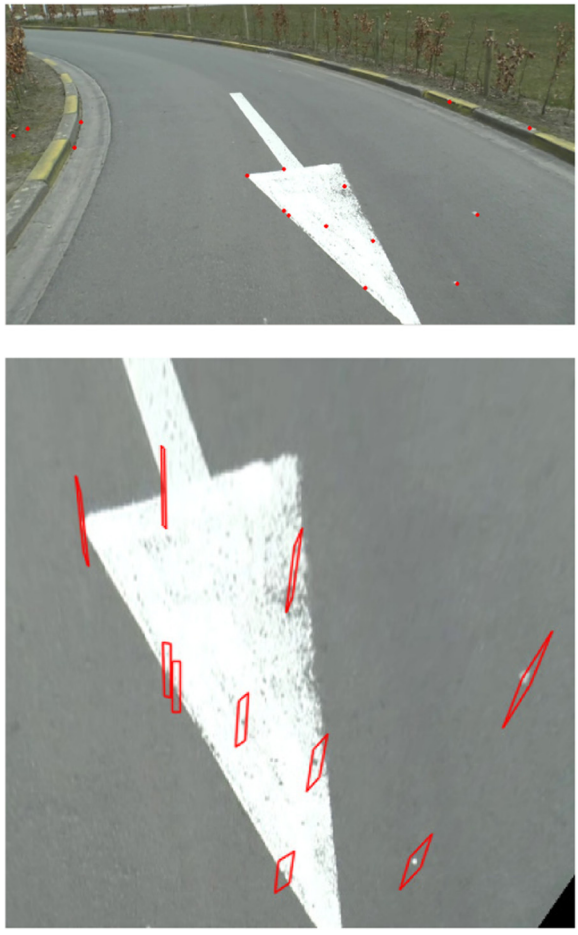

Figure 3 Example of observation uncertainty regions (red quadrilaterals, bottom) for some detected Harris corners in the camera image (red dots, top). The background image in the bottom picture was obtained by applying the inverse perspective transform in the absence of pitch and roll to the entire camera image. It serves as indication of feature context only, as it does not take into account the backprojection uncertainty.

A final remark concerns lens distortion. The above pinhole camera model does not take any distortion into account. In order for this model to be a good approximation, the distortion must either be small or corrected in pre-processing. Especially when using wider angle lenses, it is recommended to estimate radial distortion parameters (e.g., using Bouguet [27]). As these parameters are largely stable for a lens with fixed focal length, this kind of calibration does not need to be recurrent. For consumer vehicles, this means the distortion parameters can be determined at the factory or even specified by the supplier of the optics.

STOP

2.2 Feature matching

In the previous section, we have described how we can relate the *currently* observed features to regions on the ground plane. In this section, we predict where *previously* observed features may be found on the ground plane. Consider a feature with known ground plane coordinates in the previous frame. The ground plane coordinates of that feature in the current frame depend on two parameters of the vehicle motion: the steering angle and the velocity. When these parameters are known, we can use the properties of Ackermann steering geometry to predict the new coordinates of the feature. As illustrated in Figure 4, the Ackermann principle ensures that the roll axles of all wheels intersect in one point. A vehicle designed as such describes at any moment a circular trajectory around this intersection point. This point is called the *instantaneous center of rotation*. The Ackermann vehicle model does not take sideslip into account; it will therefore give rise to small inaccuracies in the prediction during hard cornering maneuvers, when the instantaneous center of rotation will no longer lie on the extension of the back axle. In normal road driving, however, sideslip angles are typically small and the Ackermann model is a good approximation [28].

Assuming the instantaneous center of rotation remains constant during the interval between two frames, we can use this circularity of trajectory to predict the new ground plane position of a feature with known position in the previous frame, for every combination of steering angle and velocity. In practice, the steering angle and velocity of the vehicle cannot change abruptly over time; at normal video frame rates, the amount of correction that can be applied by the driver between consecutive frames is very small. This means that if we have an estimate of the rotation and velocity at the previous timestep, only a small range of angle-velocity combinations is plausible at the current timestep. By evaluating the circular trajectories corresponding this patch of angle-velocity parameter space, we can delimit a search region in ground plane coordinates for each feature with known ground plane position in the previous frame. These regions will be called *prediction uncertainty regions* as each region represents the limits on the uncertainty on the predicted position of one of the tracked features. This is illustrated in Figure 5.

Our method will use the rotation and velocity estimated in the previous timestep to predict such a search region for all currently tracked features. The prediction uncertainty

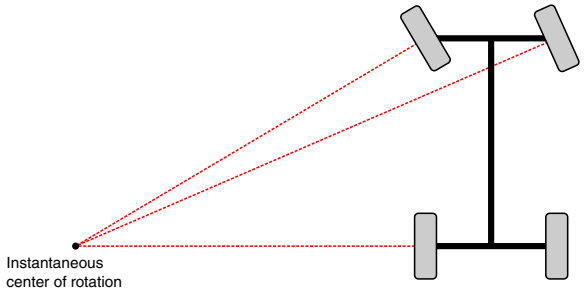

Figure 4 Illustration of the Ackermann steering principle. The vehicle will describe a circular arc around the intersection of the roll axles of the wheels.

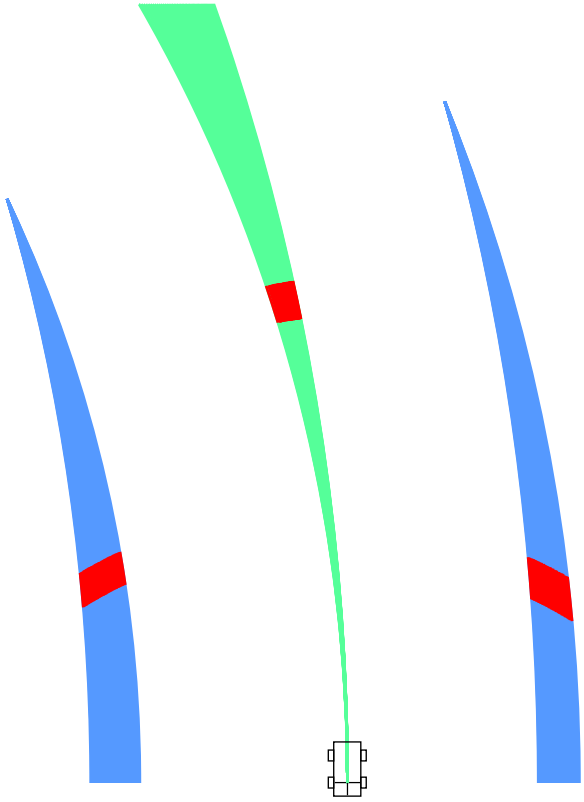

Figure 5 Prediction uncertainty regions for two features for a range of possible vehicle positions. When the vehicle drives forward with a certain initial velocity and steering angle, limited rate of change of these two parameters means the vehicle can only end up in the dark zone (red) in the central band (green). From the vehicle's point of view, the two features that were originally seen at the points of the side bands (blue) will have moved to somewhere in the dark zone on their band (red).

region for each feature is closely approximated by the quadrilateral defined by the predictions corresponding to extremal combinations of steering angle and velocity.

The upper limit on the variability of the rotation angle (i.e., the maximum of the second-order derivative of the vehicle's heading angle) cannot easily be calculated from vehicle specification as it depends on the strength of the driver as well as the power steering of the vehicle. In order to obtain realistic limits for the angular acceleration, we analyzed 22 km of GPS/INS ground truth data. The histogram of the angular acceleration is shown in Figure 6. From this histogram, we observed that 97% of occurring values are between $\pm 20°/s^2$. Of the values, 92% are between $\pm 10°/s^2$. As a trade-off between search region size and odometry accuracy in rapid maneuvres, we will typically choose a limit of $\pm 10°/s^2$.

The theoretical maximum change in vehicle speed corresponds to an emergency stop and is around 10 m/s^2. Again though, typical values during normal driving are much less extreme. Maurya and Bokare [29] measured maximum deceleration for cars in hard braking from motorway speeds to be 1.71 m/s^2. For trucks, this value is reduced to 0.88 m/s^2. On our own data, obtained using a family sedan and a van, we observed maximum deceleration to be under 1.5m/s^2. The maximum rate of acceleration of a normal road vehicle is significantly lower than the maximum rate of deceleration [30]; therefore, we will assume acceleration in normal circumstances to be under 1.5 m/s^2 as well.

Using these limits, the previous estimate of vehicle steering angle and velocity and the previous estimated ground plane positions of each tracked feature, we calculate a set of prediction uncertainty regions in the ground plane for the current frame. An example of these regions is shown in Figure 7. The prediction uncertainty regions are not uniform in shape: closer to the vehicle, they are narrower than further away.

These prediction uncertainty regions will now be used to perform location-based matching with the observation uncertainty regions defined in Section 2.1. Whenever a prediction uncertainty region overlaps with an observation uncertainty region, a potential feature match is generated. An example of the overlap of regions is shown in Figure 8. Any current observation of a ground plane feature which is already being tracked will cause at least one area of overlap between the two types of region unless the previously specified limits on vehicle attitude or maneuverability are exceeded. In case of too few overlapping regions, those limits are extended until a minimum number of matches is reached (typically chosen as 1/8th of the number of features detected).

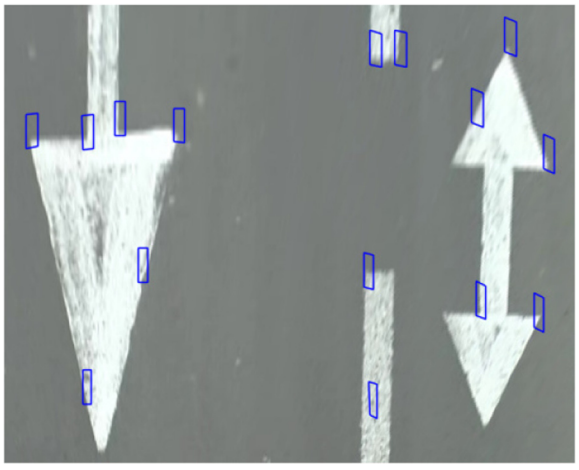

Figure 7 Example of prediction uncertainty regions for current frame based on previously tracked features. The background image was again obtained by applying the inverse perspective transform in the absence of pitch and roll to the entire camera image.

In a road driving context, location-based matching is generally preferable to appearance-based matching, as it is to be expected that many features on the road surface will have the same general appearance and the number of possible matches to be evaluated will therefore be much higher than when using a location-based approach.

Another benefit of location-based matching is that it will produce fewer spurious matches in the event of other moving objects being present in the camera view. Features

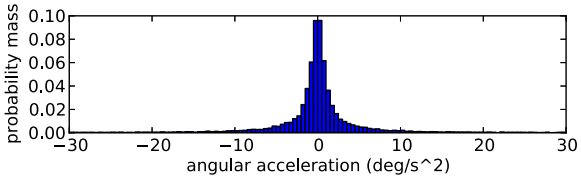

Figure 6 Histogram of angular acceleration of a vehicle during combined urban/suburban/highway driving.

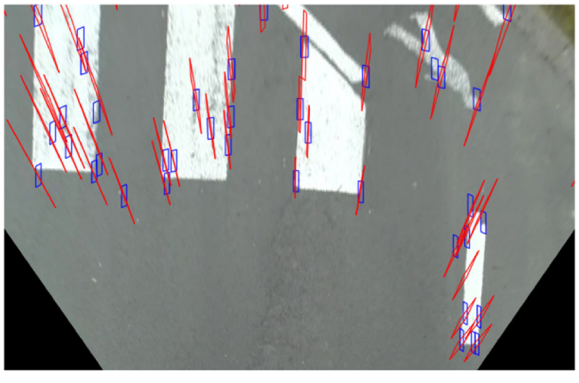

Figure 8 Example of overlap between prediction uncertainty regions (blue) and observation uncertainty regions (red).

detected on this moving object will, in general, not have observation uncertainty regions that consistently overlap with prediction uncertainty regions because the relative motion of the object does not comply to the constraints of the Ackermann model. This is a significant advantage compared to an appearance-based matcher, which will tend to match a large amount of features exhibiting a consistent motion pattern which may be hard to discern from the motion pattern of road surface features. Similarly, our matching principle will not generally produce matches for features that originate from a point significantly above the ground plane, as these features will exhibit exaggerated motion compared to actual ground plane features and therefore fall outside of the prediction uncertainty regions.

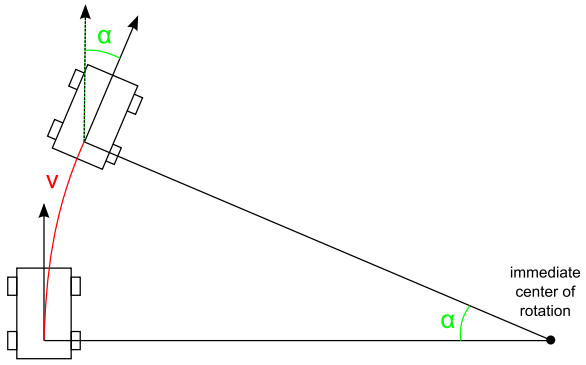

Figure 9 Illustration of the motion parameters that need to be recovered. Velocity v (red) is measured from the length of the circular arc, rotation α (green) from the difference in heading between two consecutive states.

2.3 Odometry estimation

From the feature matches obtained as described in Section 2.2, we now need to remove any remaining outliers and calculate the odometry. The odometry calculation consists of recovering the two parameters that characterize the motion of a vehicle according to the Ackermann principle: rotation angle and velocity. The rotation angle is defined as the difference in heading between two consecutive observations, i.e., the angular change between the start and end of the circle segment. The velocity of the vehicle is measured from the length of the circle segment. These two parameters are linked by the location of the immediate center of rotation (ICR): a higher rotation angle for the same speed means that the ICR is closer to the vehicle. Let α be the difference in heading between two consecutive vehicle states and v the distance traveled between the two states. Since the vehicle trajectory is circular around the ICR, the distance r between the center of the vehicle and the ICR is the radius of the circle segment, and the relation between the three parameters is given by:

$$v = \alpha r. \tag{6}$$

The motion parameters and their relation are illustrated in Figure 9.

Outliers may be caused by accidental matches of moving objects in the scene, by overlapping of uncertainty regions of multiple features with the same search region or vice versa. Also, some of the matches may be inliers but still unreliable for calculating odometry, on account of them not originating from the ground plane. Features on slightly elevated curbs, for example, will generally match, though their uncertainty regions are inaccurate. A more in-depth analysis of the degeneracy that occurs when many features are in a slightly elevated plane is presented in the Appendix.

In a traditional visual odometry framework, the calculation of relative pose change from unreliable feature matches consists of a RANSAC scheme to sort inliers from outliers and find the best supported motion hypothesis. However, RANSAC offers few advantages in our case, as the majority of the outliers have already been eliminated by the location-based matching, and any remaining outliers are difficult to identify due to the uncertainty of the observed feature coordinates associated with both inliers and outliers.

Instead of relying on RANSAC, our method employs a parameter space voting approach. This integrates well with our uncertainty regions and will allow us to easily find a consensus among the matches. Let us revisit the prediction uncertainty regions for each feature (as seen in Figure 7). The edges of these predicted regions correspond to the limits of change the driver can affect on the vehicle state, while the center of the regions corresponds to an unchanged vehicle state. As such, each prediction uncertainty region represents the same patch in rotation-velocity parameter space, centered around the last estimates for rotation and velocity. When an observation uncertainty region of one of the current features overlaps with part of one of the predicted regions, the overlap expresses a vote of this feature on a part of the rotation-velocity parameter space patch. For example, if the observation uncertainty region overlaps with the left side of the prediction uncertainty region, this corresponds to an increased likeliness that the vehicle has turned further to the right or less to the left than in the previous inter-frame interval.

In order to accumulate the votes of all features, we will represent each prediction uncertainty region by a square binary image, where each pixel corresponds to a bin in the rotation-velocity parameter space. In this image, pixels are set to one when they are overlapped by at least one observation uncertainty region and set to zero otherwise. We can now sum these square images to count the votes on each part of the prediction uncertainty regions.

An example of the square images and their sum is shown in Figure 10. As all prediction uncertainty regions represent the same patch of motion parameter space, every pixel in the sum image corresponds to a specific bin in the parameter space. Pixels with high-intensity value in the sum image represent motion parameters supported by many features. This gives us an efficient way to find a consensus among all the feature matches: the best consensus is found at the highest intensity of the image.

In practice, the discretized nature of the sum image and the limited number of features means that the location of the peak intensity is quite sensitive to noise (e.g., a single feature that has shifted by one pixel in the camera image between frames could have a significant impact on the location of the maximum). In order to reduce this noise sensitivity, we will not locate the absolute maximum but the center of gravity of the area of highest intensity. We define this area as the region in which the values exceed a fraction (typically 70%) of the absolute maximum. The center of gravity calculation is essentially an averaging mechanism and therefore reduces noise sensitivity.

The location of the center of gravity can be easily related to its corresponding values in rotation-velocity parameter space, which define the current vehicle state. When the vehicle state is known in every inter-frame interval, the complete estimated trajectory of the vehicle can be reconstructed using the circular motion model described in Section 2.2.

An important remark should be made about the accuracy of this estimation. Due to the uncertain nature of the observations (i.e., the significant size of the backprojected regions) and the limited sampling density in the parameter space, the immediate frame-to-frame estimate is of relatively low accuracy. The uncertainty on the pitch and roll angles prevent us from refining this estimate further through a closed-form calculation (e.g., a least squares solution). However, our method is self-correcting in the sense that an estimation error will result in a prediction for the next frame that is biased in the direction of the error. The observations will then accumulate votes in an area offset in the opposite direction of the prediction bias. As a consequence, the consecutive estimation errors will not accumulate but compensate each other instead. Therefore, the cumulative vehicle state over several frames will prove more accurate than the fuzzy nature of the data suggests. To illustrate this point, consider the simplified example of overestimating the velocity at time t as 0.45 m/frame while the real velocity is just 0.4 m/frame. This overestimation of the velocity is equivalent to a misestimation of the actual feature positions from the fuzzy data by 0.5 m. The prediction for time $t + 1$ will assume a constant velocity of 0.45 m/frame and use the misestimated actual feature coordinates as a starting point. The centers of the prediction uncertainty regions for time $t+1$ will therefore end up at a distance of 0.10 m to the actual feature coordinates. When the actual velocity of the vehicle at time $t + 1$ is again 0.4 m/frame, the observation uncertainty regions will then each be centered on a pixel corresponding to 0.10 m above the center of a prediction uncertainty region. The method will then correct the estimate for the second state to 0.35 m/frame, and the average estimated velocity over two states will be accurate. This safety mechanism will only mitigate single-frame estimation errors; in case of continuously poor feature matching, errors may still accumulate.

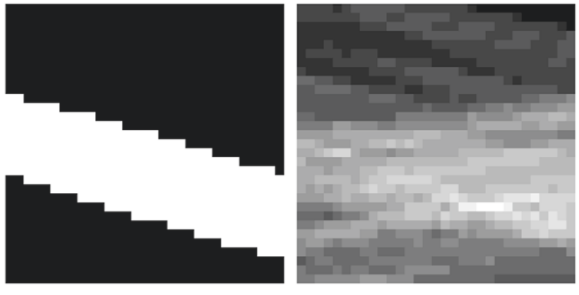

Figure 10 Example of square image representing search region and its overlapped part (left) and sum of many such square images (right).

Another remark should be made about the area of the rotation-velocity parameter space that falls outside of the predicted boundaries. This area is not taken into account for the parameter space vote. By cutting off the parts of the observation uncertainty regions at these boundaries and not taking them into account for the center of gravity calculation, we introduce a slight bias towards the center of the rotation-velocity parameter space patch. This bias is not a problem; as explained above, it is automatically corrected for in the next estimation step as long as the parameter space boundaries are chosen sufficiently wide to accomodate this extra frame-to-frame variability. The estimation bias towards the unchanged vehicle state hypothesis also limits the error caused by low feature quality. In such cases of low feature quality (caused, for example, by excessive camera vibration), the sum image will degrade into noise, and it is beneficial to overall robustness to assume a stable vehicle state in this case that can be corrected when feature quality improves.

As a final step in the odometry method, the feature tracks need to be updated. In the discussion so far, we have assumed the ground plane coordinates of each tracked feature at the previous timestep to be known. Due to the uncertain inverse perspective projection, however, these coordinates cannot be determined exactly. As a best estimate for the ground plane position corresponding to the a feature observation, we will use the centroid of its observation uncertainty region. The estimated vehicle state (rotation and velocity) is used to update this centroid at

each timestep. Additionally, for any feature detected in the camera image that did not match any prediction uncertainty regions, a new track is initiated with the centroid of its observation uncertainty region as starting coordinates. Finally, tracked features which have not matched with any observations for a number of consecutive frames (typically chosen between three and five) are discarded.

3 Calibration

In this section, we will describe how the extrinsic calibration can be determined. The extrinsic calibration is contained in the matrix $[\mathbf{R}_{\mathbf{XY}}|\mathbf{t}]$ in Equation 5. $\mathbf{R}_{\mathbf{XY}}$ is a submatrix of the 3D rotation matrix $\mathbf{R}$ that describes the rotation between the world axes and the camera axes. Homography-based methods can be found in literature to estimate this rotation matrix, notably the work of Miksch et al. [13], who determine the rotation matrix online without using odometry data or known scene geometry, but with known camera height, using inter-frame feature correspondences on the ground plane.

The extrinsic rotation matrix can also be estimated iteratively from a single image using known scene geometry, e.g., the known dimensions of a rectangular parking space. When the rotation matrix $\mathbf{R}$ is known, the translation vector $\mathbf{t}$ can be easily measured in vehicle axes with a tape measure and plumb rule and then rotated into camera axes using $\mathbf{R}^{-1}$.

Regardless of the used calibration method, it is interesting to examine the sensitivity of the proposed method to calibration accuracy. To this end, we performed a simulation in which artifical video is generated for a vehicle moving along an S-shaped point grid. The trajectory consists of two 30-m long straight sections linked by one 180° turn left and one 180° turn right. The turns are modeled as mirrored clothoids with an angular acceleration of 5°/s. The virtual camera was set in a similar configuration to the camera in our real-world dataset which will be discussed in Section 4, with zero roll and heading and −20° pitch. The simulation trajectory and an artifical video frame are shown in Figure 11.

This simulation allows us to easily control the error in each extrinsic calibration parameter and analyze its effect on the global trajectory reconstruction by the proposed method, as well as on the straight sections and bends individually. In our experiments, we first determined the best case scenario using the exact calibration parameters. Then, we performed three tests in which 1° was added to one of the rotation angles and three tests in which 10 cm was added to one of the translation components. The results can be seen in Figures 12 and 13.

From the error graphs and reconstructed trajectories, we can see that three parameters are especially important for translation accuracy. The greatest translation error occurs in the case of misestimated pitch (Figure 13, third from top), with a 1° error resulting in an overestimation of travel distance by 13%. A 10-cm vertical or longitudinal offset both result in a translation error of around 6% (Figure 13, bottom two). Inaccuracies in the other parameters yield much smaller translation errors.

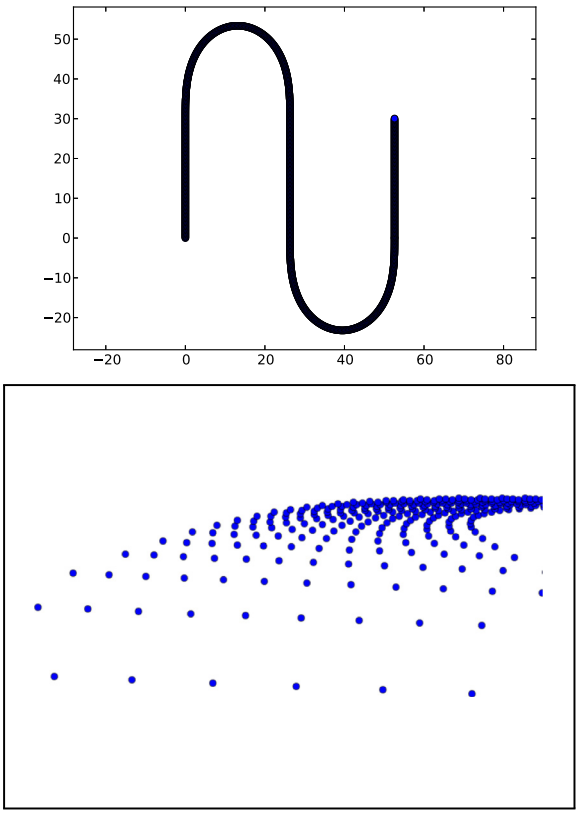

Figure 11 Simulation trajectory composed of two straights and two clothoid turns (left) and sample artifical video frame from start of bend (right).

In terms of rotation error, the single most important parameter is the heading angle (Figure 13, fourth from top). An error of 1° in this parameter results in a rotation error of 0.11°/m. The roll angle is the second most important influence on rotation accuracy (Figure 13, second from top), with a 1° roll misestimation resulting in a 0.04°/m error. The other calibration parameters have smaller effects on rotation error.

We can conclude that the proposed method is most sensitive to pitch and heading angle, followed by roll angle, vertical offset, and longitudinal offset. Generally, the offsets are easy to measure in practice, and an error of 10 cm is not likely. Estimation of the extrinsic rotation angles is more prone to inaccuracies. However, as each of the three angles has a different effect on the evolution of the error on our simulated trajectory, it is easy to identify an error in one angle. In practice, this can be done by driving along a known section of road featuring at least one straight

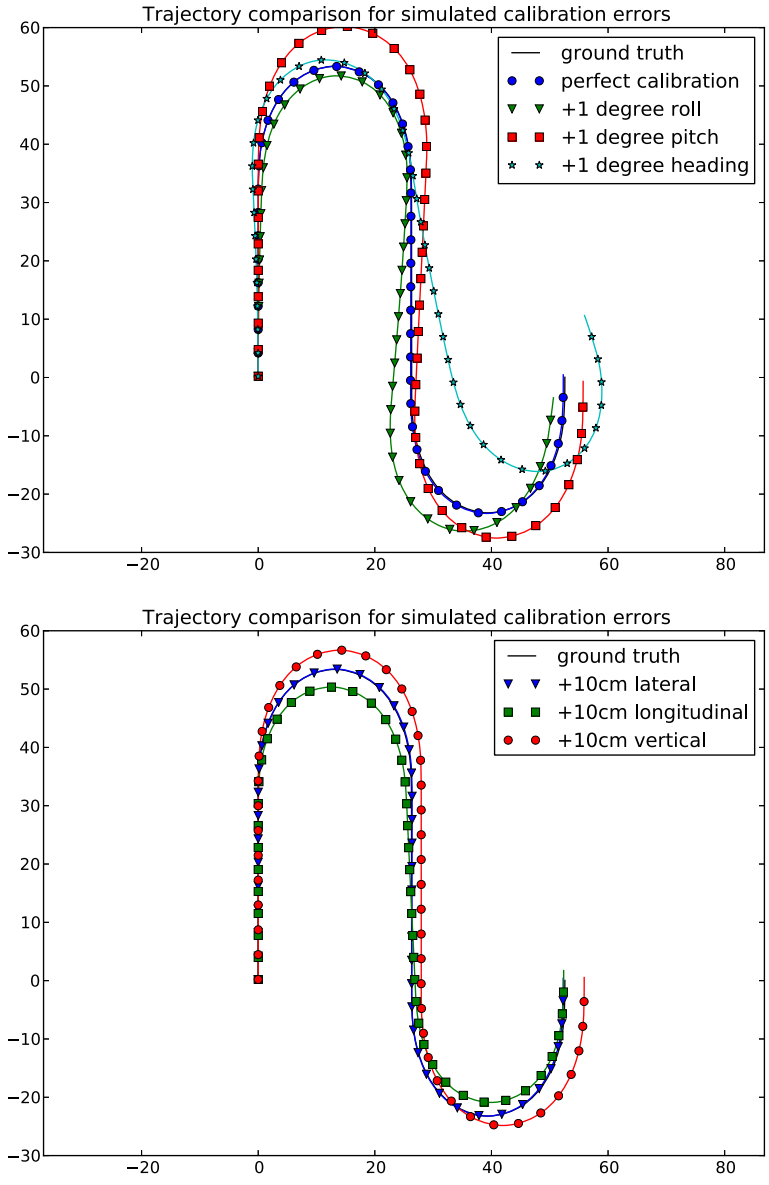

Figure 12 Effect of extrinsinc calibration errors on reconstructed trajectory. Ground truth lines are almost entirely hidden behind the perfect calibration result in the top plot and behind the lateral deviation result in the bottom figure.

section and a bend in each direction and comparing the odometry result to the known ground truth. A roll error causes a significant rotation bias on the straight sections, but not in the bends. This property can be used to refine the roll estimate. A heading error causes a constant bias regardless of road curvature, making it easy to identify and correct as well. Finally, a pitch error results in over- or underestimation of rotation in bends only and a significant constant bias on translation. If the longitudinal offset (which has similar effects) is reliable, the translation error by itself can be used to correct the pitch angle. Although these principles have already been used to manually refine the calibration estimate in some of our experiments, the automation of the process for mixed calibration errors remains future work.

4 Results

The proposed method was evaluated on two datasets and compared to the monocular eight-point solver by Geiger et al. [24]. The implementation is provided online by the authors. In the KITTI odometry benchmark, three monocular methods currently outperform this standard eight-point solver. The highest ranked method, by Chandraker and Song [17], uses a standard five-point

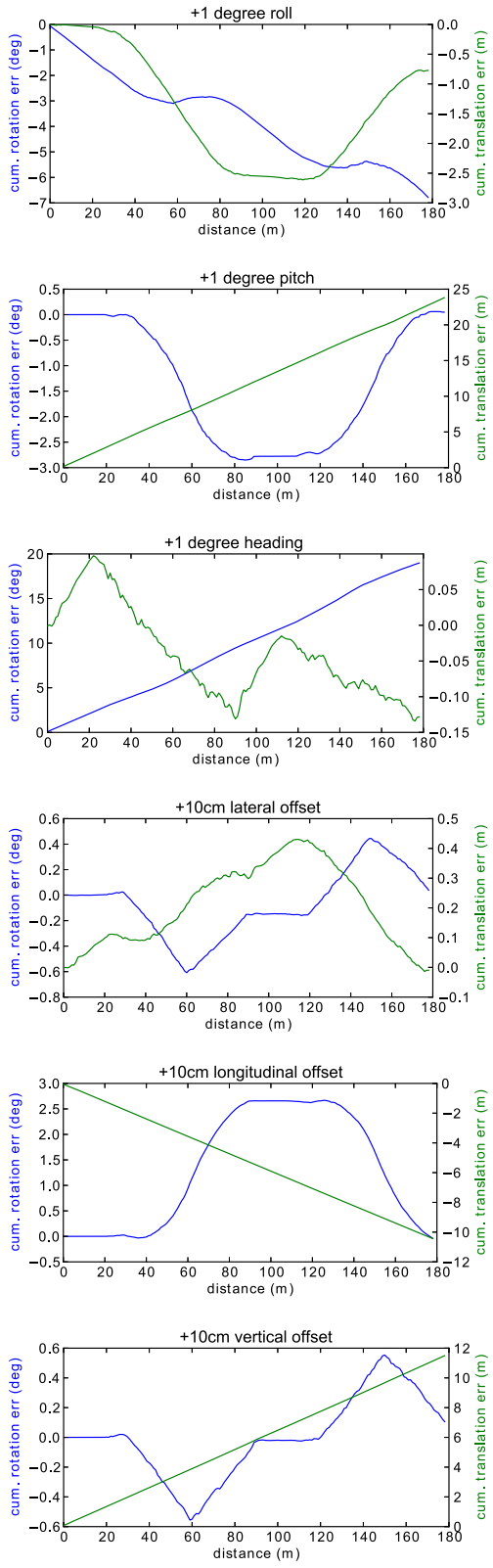

Figure 13 Influence of errors in one of the extrinsic calibration parameters on cumulative rotation and translation error. Straight sections are from 0 to 30 m and from 90 to 120 m. Note that *Y*-axis scale is different per figure.

solver as one of the base components, combined with ground plane estimation and scene structure propagation. The second method, called windowed structure from motion (W-SFM) lists no publication but is described as using a five-point solver and bundle adjustment. The third method is the eight-point solver of Geiger et al. combined with the ground plane estimation proposed by Chandraker and Song. All three methods employ either a five-point or eight-point solver to perform initial 3D pose estimation. The aim of this research is to prove that our 2D approach is a viable alternative to traditional 3D pose estimation for visual odometry. We have therefore chosen the basic eight-point solver as reference method. The potential improvements afforded by bundle adjustment and more precise ground plane estimation for the proposed method are to be explored in future work.

The first dataset on which the eight-point solver and our proposed method are compared is the KITTI odometry benchmark itself [25]. This dataset consists of 22 sequences captured in urban, suburban, rural, and highway scenarios spanning approximately 35 km. The camera offers a wide 1,241 × 376 pixel view straight ahead of the vehicle, with approximately zero pitch and zero roll. A sample frame from the KITTI dataset is shown in Figure 14. The dataset offers the extrinsic and intrinsic camera parameters needed by the eight-point solver. For the proposed method, the translation vector between camera and the center of the rear axle is also used. This was approximated using the methods described in Section 3 (iterative refinement on a short section of ground truth-annotated video). In keeping with our emphasis on ease of calibration, we did not correct for lens distortion. The parameters for the proposed method are as follows. The far cutoff line for feature detection was set at 12 m in front of the vehicle, as well as laterall cutoffs at 3 m left and right of the straightahead (to reduce the number of features detected on non-ground plane objects). The cutoffs are illustrated in Figure 14. Within the cutoffs, 32 Harris corners were detected on each side of the straightahead. Features were considered lost in case of no match for five consecutive frames.

Figure 14 Sample frames out of KITTI dataset [25]. Note the exceptionally wide field of view. Bottom image shows cutoffs for feature detection overlaid on the perspective image. Features will only be detected in the highlighted zone.

The performance evaluation provided by the KITTI benchmark is based on two metrics: translation error and rotation error. These are calculated as follows. Let $\mathbf{P}_i$ and $\mathbf{P}_j$ denote the ground truth poses corresponding to frames i and j relative to the starting pose at the beginning of the sequence. The ground truth pose change $\mathbf{Q}$ between states i and j is then given by:

$$\mathbf{Q} = \mathbf{P}_i^{-1}\mathbf{P}_j.$$

With $\mathbf{P}'_i$ and $\mathbf{P}'_j$ denoting the *estimated* poses for frames i and j relative to the starting pose, the estimated pose change is then given as:

$$\mathbf{Q}' = \mathbf{P}'^{-1}_i\mathbf{P}'_j.$$

The pose estimation error is then calculated as:

$$\mathbf{Q}_{\text{err}} = \mathbf{Q}'^{-1}\mathbf{Q}.$$

From this 4×4 matrix, translation error and rotation error are then calculated as:

$$\Delta_t = \sqrt{\mathbf{Q}_{\text{err}}[1,4]^2 + \mathbf{Q}_{\text{err}}[2,4]^2 + \mathbf{Q}_{\text{err}}[3,4]^2},$$
$$\Delta_r = \text{acos}(0.5 * (\mathbf{Q}_{\text{err}}[1,1] + \mathbf{Q}_{\text{err}}[2,2] + \mathbf{Q}_{\text{err}}[3,3] - 1)).$$

It can be easily verified that Δ_r corresponds to the heading angle difference between $\mathbf{Q}$ and $\mathbf{Q}'$ when the rotation is limited to the Z-axis.

The translation and rotation errors are calculated on all subsegments of the ground truth trajectory of length 100, 200. . . 800 m. The errors is averaged per segment length. Translation error is expressed as a percentage of segment length, while rotation error is expressed in °/m.

The proposed method only estimates rotations along one axis (the normal of the ground plane) and does not measure elevation change, while the evaluation considers full 3D poses and elevation change. The KITTI dataset contains several sequences captured on hilly roads, and we can expect the proposed method to be at a slight disadvantage in this benchmark as a result, while for real-world navigation-related applications, the elevation changes are largely irrelevant due to the planar nature of common map data.

The accuracy comparison for the KITTI dataset is shown in Figure 15. It can be seen that the translation accuracy of the proposed method is markedly better than that of Geiger et al. (8.98% compared to 11.94% average over all segment lengths). The rotational accuracy of the two methods is more similar, with the proposed method slightly better 0.0217°/m vs. 0.0234°/m average over all

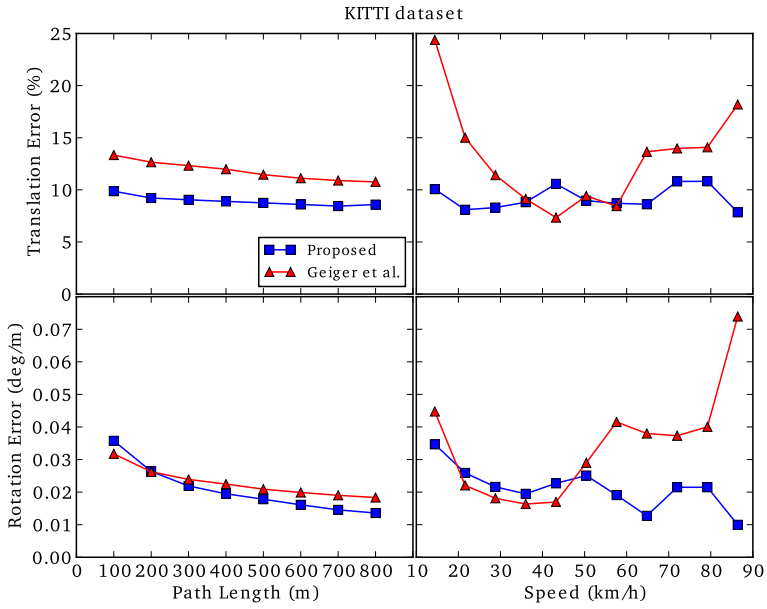

Figure 15 Accuracy evaluated on KITTI dataset. Translation errors are shown in top row, rotation errors in bottom row, both in function of segment length (left column) and speed (right column).

segment lengths. The average translation error of the proposed method is smaller for any segment length, while the average rotation error is smaller for segments longer than 200 m. We may conclude that on the KITTI dataset, the proposed method is significantly more accurate than the eight-point solver, with a 24% reduced translation error and 8% reduced rotation error. Some examples of estimated trajectories are shown in Figure 16.

Both methods were able to process the data faster than real-time on a desktop computer (Intel Core i5 3.40 GHz $\times 4$), with the proposed method significantly outperforming the eight-point method (86.4 vs. 17.2 fps). The feature detection step in the proposed method is implemented to make use of multi-core systems (in this case running on four cores), while the remainder of the processing is single threaded. The method of Geiger et al. runs completely single threaded.

The second evaluation dataset consists of 15 km of video captured in the urban and suburban areas of Hasselt and Diepenbeek, Belgium, using one of the mobile mapping vehicles of Grontmij Belgium. The vehicle uses an Applanix POSLV420 GPS/INS unit for ground truth positions and was equipped with a roof-mounted Panasonic AG-HPX171 960x720 anamorphic HD camera, facing rearwards and pointing slightly down at an angle of approximately 20°. A sample video frame is shown in Figure 17. The camera captures video at 50 fps and was calibrated intrinsically using a checkerboard pattern and the method of Bouguet [27]. The extrinsic calibration was estimated iteratively as explained in Section 3.

The slightly downward pitch of the camera in these video sequences is considered slightly better for the proposed method, as it offers a denser coverage of the nearby road plane. A second difference with the KITTI set is the reduced horizontal angle of view of the camera. In the KITTI set, the aspect ratio is about 3.3, significantly wider than the standard 1.78 widescreen ratio of the HD camera used for the Diepenbeek/Hasselt set. This narrower field of view means that average feature displacement for a given speed is reduced (since features off to the sides have the greatest displacements), and therefore, the triangulation accuracy is also expected to be slightly lower.

While the camera captured the video at 50 fps, we discarded four out of every five frames to attain the same 10 fps frame rate as in the KITTI sequences. The parameters for the proposed method were similar to those for KITTI, with the exception of the far cutoff for feature detection, which was set to 20 m as increasing downward pitch shrinks the perspective uncertainty regions for any given distance. The odometry results for the Diepenbeek/Hasselt dataset were also processed with the evaluation code provided with the KITTI benchmark. The accuracy for the Diepenbeek/Hasselt dataset is shown in Figure 18. The translation accuracy of the proposed method on this data is comparable to that on the KITTI set, with an average error of 7.23% against 10.68%. In terms of rotation accuracy, the advantage of the proposed method increases significantly, with 0.0189°/m vs. 0.0302°/m.

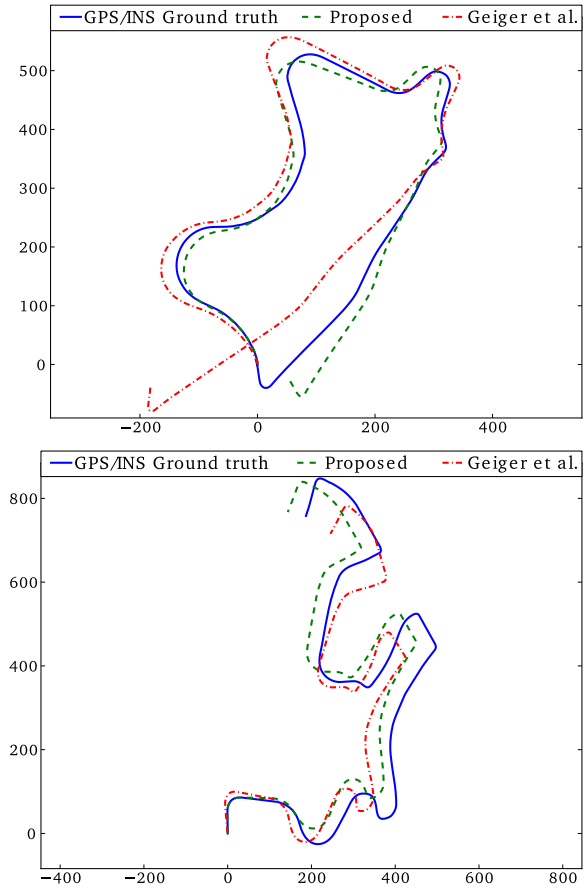

Figure 16 Examples of estimated KITTI trajectory according to proposed method and Geiger et al. compared to ground truth. Axes are in meters. Blue line represents ground truth, green dashed line the proposed method, red dot-dashed line the eight-point solver.

Overall, the proposed method is markedly better than the method of Geiger et al. in both metrics on both datasets.

Some examples of reconstructed trajectories are shown in Figure 19. A summary of translation and rotation accuracy on both sets is shown in Table 1.

Figure 17 Sample frame of Diepenbeek/Hasselt dataset.

5 Analysis

The results obtained on both datasets clearly illustrate the main advantage of the proposed method over the eight-point solver, namely, better recovery of scale. In several of the sequences, the eight-point solver significantly misestimates the length of one or more straight segments (e.g., the final section in the left plot of Figure 16). This is due to an inherent weakness in the monocular pose estimation. Due to the projective nature of the camera, the translation can only be recovered from the fundamental matrix up to a scale factor. As was noted in Kitt et al. [31], this scale factor is susceptible to drift. Scale drift is remedied in Geiger's method by relating the triangulation of points to a known length in the scene, specifically the height of the camera above the ground plane (which is assumed constant). The results both on the KITTI and the Diepenbeek/Hasselt datasets clearly show that this corrected scale is less accurate than the scale obtained by our robust tracking of ground plane features. The fact that Geiger et al. are better able to recover the scale on the Diepenbeek/Hasselt dataset than on the KITTI dataset further corroborates this explanation: in the Diepenbeek/Hasselt set, the camera is placed significantly higher above the ground plane, which means that similar absolute errors in the estimation of the ground plane have a smaller effect when divided by the longer fixed distance.

An important trend can be observed in the results of both methods. Rotational error decreases with increasing segment length. We may conclude from this that there is some noise present on the immediate poses estimated by both methods, which averages to zero over many estimations.

The effect of vehicle speed on the translation and rotation errors is less clear from the plots, as the two datasets show slightly different trends. The high errors of both methods for low speeds on the Diepenbeek/Hasselt dataset can be explained by the fact that the low speeds mostly prevail in the busy city center, where the presence of other traffic degrades the results somewhat. In the KITTI dataset, this correlation between speed and traffic density is not present, and as such, the proposed method does not exhibit significant sensitivity to vehicle speed.

Regarding sensitivity to other traffic, we may conclude that both methods cope reasonably well with the busy urban scenario in the Diepenbeek/Hasselt dataset. Only in cases when an exceptionally large area of the image is occluded by a vehicle (e.g., a street car or truck) is the estimation significantly wrong. The nature of the error, however, is different for both methods. While the eight-point solver can produce an erratic motion estimate, the proposed method assumes steady state as a fallback mechanism. This is illustrated in Figure 20.

For the proposed method, we observed that meaningful vehicle states were produced for inlier ratios as low as 1:8,

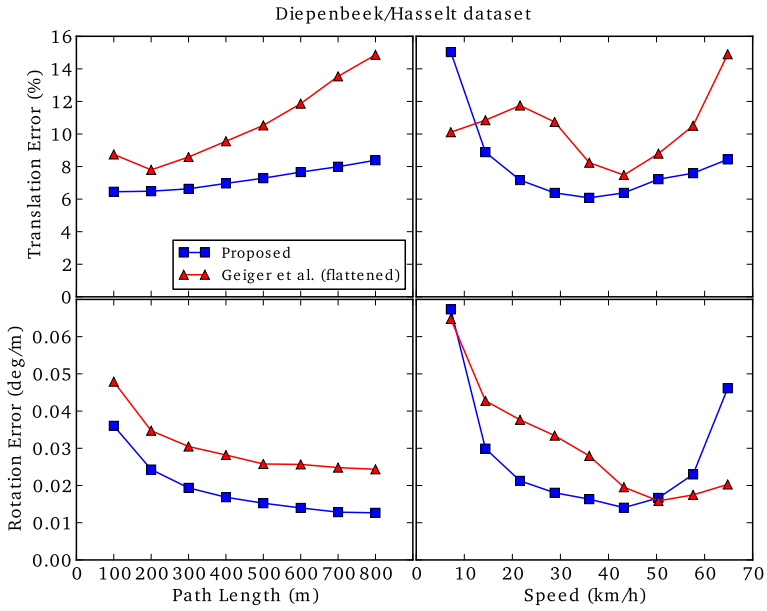

Figure 18 Accuracy evaluated on Diepenbeek/Hasselt dataset. Translation errors are shown in top row, rotation errors in bottom row, both in function of segment length (left column) and speed (right column).

counted as features generating uncertainty region overlap divided by total feature count. This proves the efficacy of the location-based matching and the parameter space voting to extract odometry from noisy and unstable features. Below inlier ratios of 1:8, assuming an unchanged vehicle state proved better than using the state estimation, this 1:8 threshold on inlier ratio was added to the method.

A remarkable difference between the results of the two datasets is that on the KITTI sequences, the translation error of both methods is decreasing for increasing segment length, while on the Diepenbeek/Hasselt data, the opposite is true. This can be explained by the fact that the vehicle's trajectory in the KITTI set is in general more compact; many of the sequences contain multiple loops and the starting and ending position are often close to each other. The Diepenbeek trajectories are less circular in nature. It can easily be seen that having loops or u-turns in a segment will reduce the absolute error over this segment compared to a segment of the same length but with a larger offset between start and end position. We consider the Diepenbeek/Hasselt set to be more representative of a typical car journey as it is a 15-km two-way travel from Diepenbeek to Hasselt and back, rather than an artificial data acquisition trajectory with the aim of covering as many streets and turns as possible in a short time and small area.

Looking at the estimated trajectories in more detail, we see that the proposed method has a significant rotational bias on some segments. One examples can be seen in the bottom left plot of Figure 19. This is due to the non-planarity of the road environment in those segments. A more in-depth analysis of these situations is given in the Appendix. The method of Geiger et al. does not suffer from this flaw. It is therefore to be expected that on long, straight roads, the eight-point solver will provide more reliable heading estimation. As both of the evaluated datasets feature many turns in quick succession due to the suburban environment, this is not readily apparent from the performance numbers.

Another interesting observation is that the steering angle and velocity estimates of the proposed method exhibit quick oscillations around the ground truth values, while their running average tracks the ground truth closely. As an example, the steering angle estimate for the top right trajectory in Figure 19 is shown in Figure 21. This corroborates our claims about the self-correcting nature of the estimates as explained in Section 2.3: an error in the immediate state estimate tends to produce an error equal in magnitude but opposite in sign during the next iteration.

Overall, we can observe that the proposed method is often better than the eight-point method at recovering macro-maneuvres present in the trajectory: at intersections and roundabouts, the eight-point solver sometimes fails to accurately estimate the large changes in heading. An example can be seen in the top right image of Figure 19: the method of Geiger et al. misses most of the roundabout. The ability to correctly estimate big maneuvres is especially important for the integration with offline map data, as maneuvres are generally more reliable clues

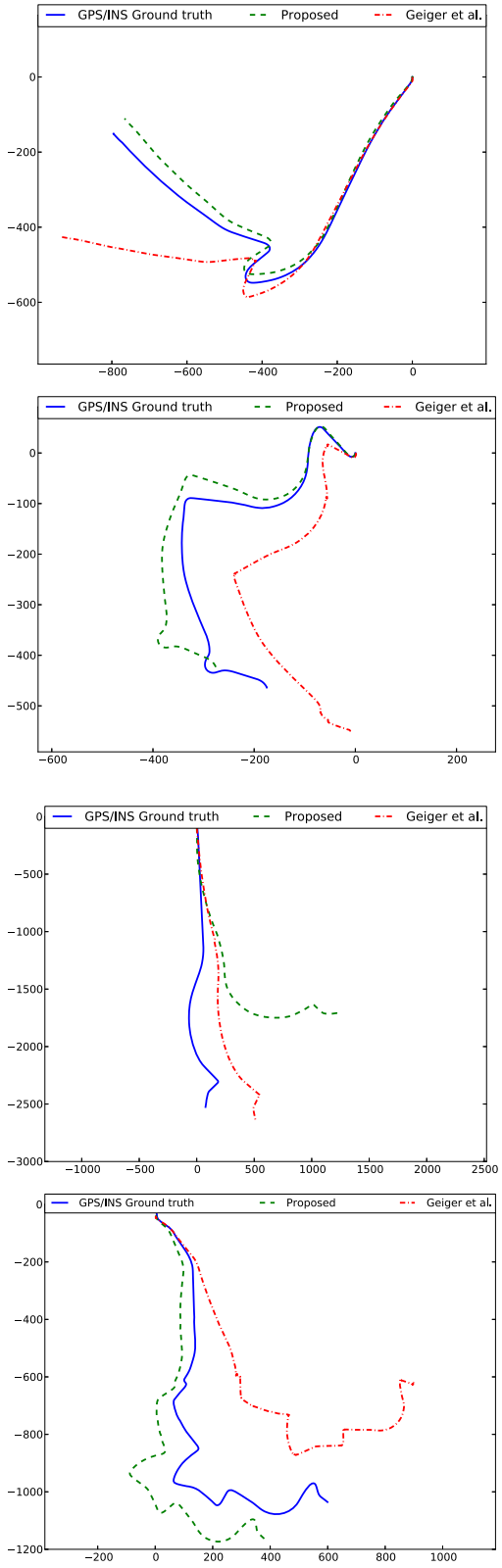

Figure 19 Examples of estimated trajectories on the Diepenbeek/Hasselt dataset. Axes are in meters. Blue line represents ground truth, green dashed line the proposed method, red dot-dashed line the eight-point solver.

Table 1 Summary of mean errors of both methods on both datasets

Dataset	Method	Transl.err. (%)	Rot.err. (° /m)
KITTI	Proposed	8.98	0.0217
	Geiger et al.	11.94	0.0234
Diepenbeek	Proposed	7.23	0.0189
	Geiger et al.	10.68	0.0302

for map matching than gentle curves and straights. The concept of map matching as a mechanism to eliminate error accumulation has already been proven [2,23].

6 Conclusions

We have proposed a monocular visual odometry algorithm that uses planar tracking of features rather than traditional 3D pose estimation. It is demonstrated that in a typical monocular setting, the method has a significant performance advantage over traditional fundamental matrix estimation.

The proposed method is applicable to both forward- and rearward-facing cameras and is proven to work well on camera pitch angles ranging from horizontal (zero pitch) to 20° downwards. We may assume that similar or higher performance will be achieved as long as the camera view covers the ground plane up to a distance of approximately 12 m (this is the nearest cutoff point used for feature detection). The camera height over the two datasets also differs significantly (1.5 and 2.7 m) so we may conclude that our method is applicable to a wide range of vehicles and camera mount points.

The improvement over the eight-point method brings the translation error of the proposed method in a difficult but representative real-world scenario down to around 15 per 200 m traveled on average. Several techniques have already successfully been applied to improve the results of the standard eight-point solver, such as bundle adjustment and ground plane normal estimation [17]. In future work, we expect these techniques to further improve the proposed method as well.

As visual odometry will typically be only one component in a mixed-data system (e.g., coupled with an offline map and magnetic compass), it is our opinion that the performance improvement of the proposed method over the standard eight-point solver is significant and can make a large contribution to a navigation system which does not depend on any outside communication.

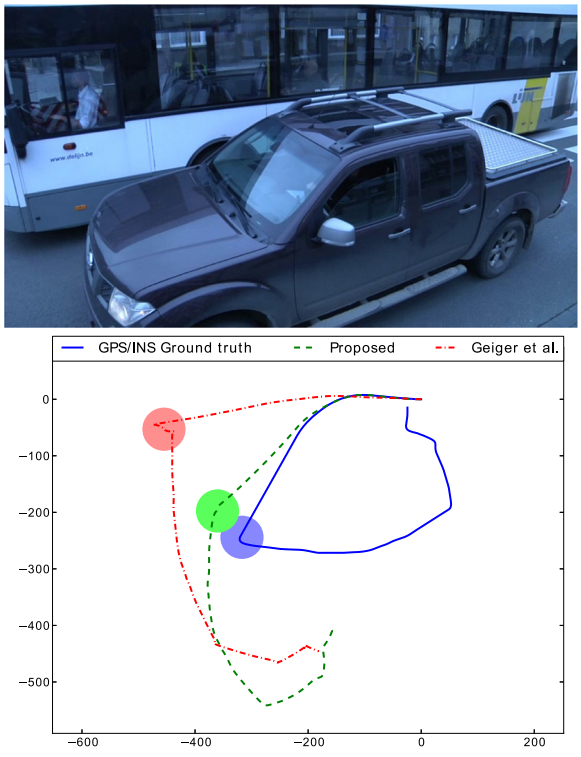

Figure 20 Left turn in dense traffic. Camera image (top) is almost entirely composed of moving vehicles. Trajectory estimation (bottom) of both methods suffers; Geiger et al. produces erratic motion while proposed method assumes steady state during ambiguous period.

Appendix

Degeneracy

In our work, we have made the assumption that the road surface is planar. However, in reality, there are two important scenarios in which this assumption is violated, but in such a way that the outlier removal mechanisms of the proposed method are ineffective. We therefore call these scenarios degenerate configurations for the proposed method.

The first scenario is that of a road with a raised kerb. In urban and suburban environments, this is a common occurrence, and its effects on the odometry estimation must be analyzed and quantified. To this end, we have adapted the simulation from Section 3 so that the point grid is elevated on one side of the trajectory. Specifically, points in the zone from 2 to 3 m on the right side were raised by 15 cm, a typical kerb height. To emulate the worst-case scenario, points on the center section of the road were removed, leaving only the points at the left and right edge for odometry computation. An example of the resulting artificial video is shown on the left of Figure 22. The resulting odometry errors are shown in Figure 23 (top). The effects in this worst-case scenario are quite pronounced: a 0.078°/m rotation error and a 6.3% translation error. This is a logical result: the height difference between the left and right side gives rise to similar errors as an inaccurate estimation of the roll angle.

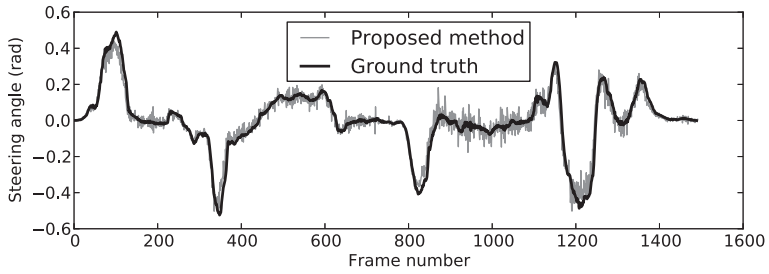

Figure 21 Oscillation of estimated steering angle (gray) around ground truth values (black), illustrating self-correcting nature of the estimation method.

These errors are significantly mitigated when feature points are present in the center section of the road as well. In this case, the consensus is still formed primarily by planar features, and the elevated features have a smaller influence. A simulated video frame for this situation is shown in Figure 22 (right) and the resulting errors in Figure 23 (bottom). The errors in this case are insignificant at only 0.004°/m for rotation and 0.1% for translation.

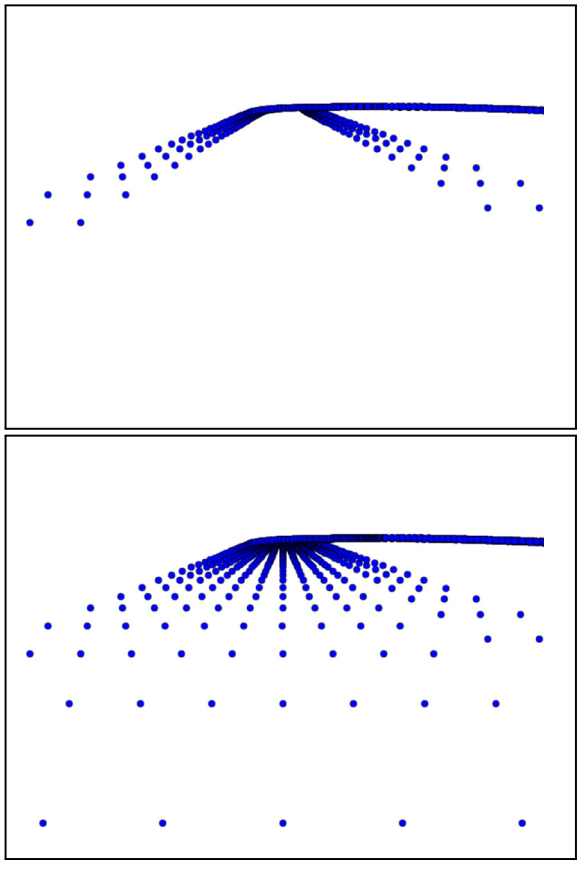

Figure 22 Artificial video frames for the kerb simulations without (left) and with (right) center points.

The second scenario is that of a road with a crown, i.e., a road with a crown in which the center line is higher than the edges to improve water drain properties. On bidirectional single-lane or two-lane roads, this is a common property. On highways or unidirectional roads, a crownless sloped design is the norm. In the latter case, the planarity assumption holds from the point of view of the vehicle, as the axles of the vehicle remain parallel to the entire span of road surface. In the case of a crowned road, however, the two sides of the road are in different planes and this will cause the inverse perspective transform to be inaccurate for part of the features when the vehicle is driving on one side of the center line or for all of the features when the vehicle is driving over the center line.

To quantify the deterioration of the odometry result in these two cases, two simulations were performed similar to those mentioned in Section 3. In the first simulation,

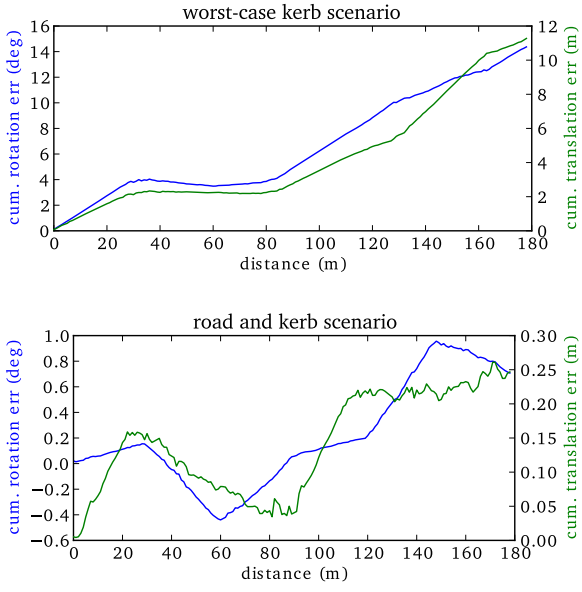

Figure 23 Errors for kerb simulations in worst-case scenario without center section points (top) and typical scenario with center section points (bottom).

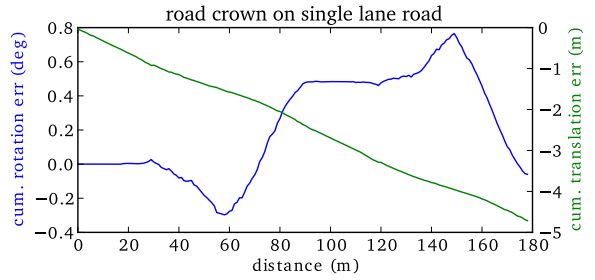

Figure 24 Odometry error in case of single lane road with crown centrally across the lane.

points beyond the left side of the vehicle were sloped downwards with a 4% slope. This corresponds to what can be expected when a vehicle drives on the right lane of a two-lane road crowned at the typical recommended slope of 2% [32]. The odometry errors were only evaluated on the straight sections, as superelevation (i.e., a single-slope, banked turn) is generally used in bends instead of a crowned design. In the worst-case scenario, with no feature points in the center section, the rotation error was 0.013°/m and the translation error −0.5%. These errors are an order of magnitude smaller than those caused by the kerb scenario or in the calibration experiments. We may conclude that for a typical two-lane road, the crown does not cause significant errors in the odometry estimation.

For the simulation of the single-lane road, where the road cross section slopes down on both sides of the vehicle at a rate of 2%, the errors are shown in Figure 24. As a result of the features on average being below the plane defined by the wheels of the vehicle, a translation error of −2.6% is observed, similar to the effect of an underestimated vertical offset. This type of road is uncommon in urban and suburban settings but is often found in rural regions across Europe.

We may conclude that while the outlier removal mechanisms in the proposed method cannot completely avoid errors caused by non-planarity of the road, the impact of these errors in typically occurring road geometries is low. In the worst-case scenarios, performance is still acceptable, although the non-planarity may become the dominant error source.

Competing interests
The authors declare that they have no competing interests.

Acknowledgement
This research was made possible through iMinds, an interdisciplinary research institute founded by the Flemish Government.

Author details
[1]Ghent University - IPI/iMinds, St-Pietersnieuwstraat 41, B-9000 Ghent, Belgium. [2]Grontmij Belgium, Arenbergstraat 13/1, B-1000 Brussels, Belgium.

Received: 28 January 2015 Accepted: 1 April 2015
Published online: 26 April 2015

References

1. P Huang, Y Pi, Urban environment solutions to GPS signal near-far effect. IEEE Aerosp. Electron. Syst. Mag. **26**(5), 18–27 (2011). doi:10.1109/MAES.2011.5871387
2. M Brubaker, A Geiger, R Urtasun, in *Conf. Computer Vision and Pattern Recognition*. Lost! Leveraging the crowd for probabilistic visual self-localization (IEEE Piscataway, NJ, USA, 2013)
3. I Parra Alonso, D Fernandez Llorca, M Gavilan, S Alvarez Pardo, MA Garcia-Garrido, L Vlacic, MA Sotelo, Accurate global localization using visual odometry and digital maps on urban environments. IEEE Trans. Intell. Transp. Syst. **13**(4), 1535–1545 (2012). doi:10.1109/TITS.2012.2193569
4. HC Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections. Nature. **293**, 133–135 (1981). doi:10.1038/293133a0
5. J Philip, A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. Photogrammetric Record. **15**(88), 589–599 (1996)
6. D Nister, An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**(6), 756–770 (2004). doi:10.1109/TPAMI.2004.17
7. R Hartley, A Zisserman, *Computation of the fundamental matrix F, Mult. View Geometry Compu. Vis.* (Cambridge Univ. Press, Cambridge, UK, 2004), pp. 279–309
8. PHS Torr, AW Fitzgibbon, A Zisserman, The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. Int. J. Comput. Vis. **32**(1), 27–44 (1999)
9. F Bellavia, M Fanfani, F Pazzaglia, C Colombo, in *Image Analysis and Processing ICIAP 2013*. Lecture Notes in Computer Science, ed. by A Petrosino. Robust selective stereo slam without loop closure and bundle adjustment, vol. 8156 (Springer New York, NY, USA, 2013), pp. 462–471
10. H Badino, A Yamamoto, T Kanade, in *International Workshop on Computer Vision for Autonomous Driving @ ICCV*. Visual odometry by multi-frame feature integration (IEEE Piscataway, NJ, USA, 2013), pp. 222–229
11. TD A. Bodis-Szomoru, Z Fazekas, in *Stereo Vision*. Calibration and sensitivity analysis of a stereo vision-based driver assistance system (Intech Rijeka, Croatia, 2008), pp. 1–27
12. T Dang, C Hoffmann, C Stiller, Continuous stereo self-calibration by camera parameter tracking. IEEE Trans. Image Process. **18**(7), 1536–1550 (2009). doi:10.1109/TIP.2009.2017824
13. M Miksch, B Yang, K Zimmermann, in *Intelligent Vehicles Symposium (IV), 2010 IEEE*. Automatic extrinsic camera self-calibration based on homography and epipolar geometry (IEEE Piscataway, NJ, USA, 2010), pp. 832–839. doi:10.1109/IVS.2010.5548048
14. J-P Tardif, Y Pavlidis, K Daniilidis, in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference On*. Monocular visual odometry in urban environments using an omnidirectional camera (IEEE Piscataway, NJ, USA, 2008), pp. 2531–2538. doi:10.1109/IROS.2008.4651205
15. D Scaramuzza, Performance evaluation of 1-point-ransac visual odometry. J. Field Robot. **28**(5), 792–811 (2011). doi:10.1002/rob.20411
16. D Scaramuzza, R Siegwart, Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. IEEE Trans. Robot. **24**(5), 1015–1026 (2008). doi:10.1109/TRO.2008.2004490
17. S Song, M Chandraker, in *CVPR, Columbus, Ohio, USA*. Robust scale estimation in real-time monocular SFM for autonomous driving (IEEE Piscataway, NJ, USA, 2014)
18. J Stühmer, S Gumhold, D Cremers. Real-time dense geometry from a handheld camera, Darmstadt, Germany (Springer New York, NY, USA, 2010), pp. 11–20
19. A Wendel, M Maurer, G Graber, T Pock, H Bischof, in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference On*. Dense reconstruction on-the-fly (IEEE Piscataway, NJ, USA, 2012), pp. 1450–1457. doi:10.1109/CVPR.2012.6247833
20. J Engel, J Sturm, D Cremers, in *Computer Vision (ICCV), 2013 IEEE International Conference On*. Semi-dense visual odometry for a monocular camera (IEEE Piscataway, NJ, USA, 2013), pp. 1449–1456. doi:10.1109/ICCV.2013.183
21. J Engel, T Schops, D Cremers, in *Computer Vision ECCV 2014*, Lecture Notes in Computer Science, ed. by D Fleet, T Pajdla, B Schiele, and T Tuytelaars.

LSD-SLAM: large-scale direct monocular slam, vol. 8690 (Springer, 2014), pp. 834–849
22. D Van Hamme, P Veelaert, W Philips, in *Intelligent Vehicles Symposium (IV), 2011*. Robust monocular visual odometry by uncertainty voting (IEEE Piscataway, NJ, USA, 2011), pp. 643–647. doi:10.1109/IVS.2011.5940453
23. D Van Hamme, P Veelaert, W Philips, in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference On*. Communicationless navigation through robust visual odometry (IEEE Piscataway, NJ, USA, 2012), pp. 1555–1560. doi:10.1109/ITSC.2012.6338668
24. A Geiger, LIBVISO2: C++ Library for Visual Odometry 2. http://www.cvlibs.net/software/libviso. Accessed 21 May 2014
25. A Geiger, CSP Lenz, R Urtasun, The KITTI Vision Benchmark Suite. http://www.cvlibs.net/datasets/kitti/eval_odometry.php. Accessed 10 Jan 2015
26. Z Zhang, A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000). doi:10.1109/34.888718
27. J Bouguet, *Visual methods for three-dimensional modeling*. (PhD thesis, California Institute of Technology, 1999)
28. E Davin, *Parameter identification of a linear single track vehicle model*. (Technical report, Technical University of Eindhoven, 2011)
29. AK Maurya, PS Bokare, Study of deceleration behaviour of different vehicle types. Int. J. Traffic Transp. Eng. **2**(3), 253–270 (2012)
30. G Long, *Acceleration characteristics of starting vehicles*. (Technical report, Transportation Research Board, 2000)
31. BM Kitt, J Rehder, AD Chambers, M Schonbein, H Lategahn, S Singh, in *Proc. European Conference on Mobile Robots*. Monocular visual odometry using a planar road model to solve scale ambiguity (IEEE Piscataway, NJ, USA, 2011)
32. JA Rosenow, in *MnDOT Road Design Manual*. Cross sections, (2012)