Robuuste egolokalisatie met monoculaire visuele odometrie

Robust ego-localization using monocular visual odometry

David Van Hamme

Promotoren: prof. dr. ir. Wilfried Philips, prof. dr. ir. Peter Veelaert
Proefschrift ingediend tot het behalen van de graad van Doctor
in de Ingenieurswetenschappen

Vakgroep Telecommunicatie en Informatieverwerking
Voorzitter: prof. dr. ir. H. Bruneel
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2016 - 2017

UNIVERSITEIT
GENT

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Telecommunicatie en Informatieverwerking

Promotoren: Prof. Dr. Ir. Wilfried Philips
Prof. Dr. Ir. Peter Veelaert

Voorzitter van de jury: Prof. Dr. Ir. Rik Van de Walle
Leden van de jury: Prof. Dr. Rudi Penne
Prof. Dr. Nico Van de Weghe
Dr. Ir. Hiep Quan Luong
Dr. Ir. Jan Aelterman

Universiteit Gent
Faculteit Ingenieurswetenschappen

Vakgroep Telecommunicatie en Informatieverwerking
St-Pietersnieuwstraat 41, B-9000 Gent, België

Tel.: +32 9 264 34 12
Fax.: +32 9 264 42 95

Dit werk kwam deels tot stand in het kader van het ISYSS (Intelligent Systems for Security and Safety) project gefinancierd door iMinds.

Proefschrift tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Academiejaar 2016-2017

Table of Contents

List of Figures

List of Tables

List of Abbreviations

BRIEF	Binary Robust Invariant Scalable Keypoints
DGPS	Differential Global Positioning System
DOF	Degrees of Freedom (e.g. 3DOF)
DoG	Difference of Gaussian
EKF	Extended Kalman Filter
EOBD	Extended On-Board Diagnostics
FAST	Features from Accelerated Segment Test
GDOP	Geometric Dilution Of Precision
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HDOP	Horizontal Dilution Of Precision
HMM	Hidden Markov Model
ICR	Immediate Center of Rotation
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
KLT	Kanade-Lucas-Tomasi (feature tracker)
LoG	Laplacion of Gaussian
MAP	Maximum A Posteriori (estimation)
MC	Monte Carlo (simulation)
MI	Mutual Information

OBD	On-Board Diagnostics
OBD2	On-Board Diagnostics version 2
OSM	OpenStreetMap
RANSAC	RANdom SAmple Consensus
ROI	Region Of Interest
SAD	Sum of Absolute Differences
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SURF	Speeded Up Robust Features
SUSAN	Smallest Univalue Segment Assimilating Nucleus
SVD	Singular Value Decomposition
VDOP	Vertical Dilution Of Precision

Nederlandse samenvatting –Summary in Dutch–

In de laatste 30 jaar is de innovatie in de automobielsector vooral gedreven door de opmars van elektronica. Die elektronica schuilt achter elk aspect van de moderne wagen, van besturing (bijvoorbeeld drive-by-wire acceleratie in plaats van een mechanische verbinding met het gaspedaal) over emissie (beginnende met elektronische brandstofinjectie) en probleemdiagnose tot veiligheid (bijvoorbeeld airbags en ABS). Nu technologie ons steeds dichter brengt bij de massageproduceerde autonome wagen, is het vooral dat laatste aspect dat de beperkende factor blijkt te zijn. Hoe kunnen we veiligheid garanderen in de complexe, snel bewegende jungle van het wegennetwerk?

Een eerste stap naar autonome voertuigen is actieve veiligheid. Moderne wagens kunnen actief stappen ondernemen om botsingen te vermijden of de schade ervan te minimaliseren. Een voorbeeld is het automatisch aanspannen van de veiligheidsgordel tijdens een botsing om het punt van impact met de airbag te optimaliseren. Een ander voorbeeld is automatisch remmen op basis van radar. Veel meer is echter mogelijk in dit veld, wanneer een voorwaarde vervuld is: perceptie van ruimtelijke context. Een voertuig kan pas autonomie bereiken wanneer het niet alleen weet waar het zich bevindt, maar ook waar alle andere weggebruikers en infrastructuur zijn. Deze dissertatie richt zich op het eerste aspect: hoe kan het voertuig zijn eigen positie bepalen?

Een manier is het gebruik van satellietnavigatie. Dit is een erg aantrekkelijke optie voor niet-kritische toepassingen, wat ook de reden is waarom dergelijke systemen alomtegenwoordig zijn in het huidige wagenpark. Voor applicaties gerelateerd aan veiligheid blijkt satellietnavigatie echter onvoldoende betrouwbaar. In omstandigheden waar onvoldoende satellieten direct zichtbaar zijn, is positionering niet mogelijk. Zulke situaties komen relatief vaak voor, bijvoorbeeld in tunnels of onder bruggen, maar ook in smalle straatjes geflankeerd door hoge gebouwen, of tussen wolkenkrabbers in grootsteden. Gerelateerd aan het zichtbaarheidsprobleem is het probleem van reflectie. Wanneer de satellietsignalen niet rechtstreeks op de ontvanger aankomen, maar ook via indirecte paden (bijvoorbeeld na reflectie op een gevel), kan dit fouten op de positionering veroorzaken gaande van van meters tot kilometers. Ten laatste kunnen ook signaalattenuatie en atmosferische effecten tijdelijk de nauwkeurigheid verminderen. Het mag duidelijk zijn dat een of liefst zelfs meer andere systemen nodig zijn om op terug te vallen wanneer satellietnavigatie het laat afweten.

Radar, infraroodsensoren, laserscanners en andere technologiën kunnen zeker bijdragen tot de perceptie van de omgeving door een bestuurder. Elk van deze sensoren heeft echter ook belangrijke nadelen. Radar heeft lage resolutie en typisch een beperkt gezichtsveld. Infraroodsensoren bieden onvoldoende scherpte, vooral op uniforme oppervlakken zoals het wegdek. Laserscanners zijn nog steeds zeer duur, en moeilijk onopvallend te integreren in een personenwagen.

Het grootste deel van de weginfrastructuur is echter ontworpen voor een andere sensormodaliteit: de ogen van de bestuurder. Omdat visuele aanwijzingen zo alomtegenwoordig zijn, rusten constructeurs steeds vaker hun voertuigen uit met camera's om de bestuurder te helpen zijn omgeving te interpreteren. Deze camera's bieden ook opportuniteiten voor voertuigpositiebepaling. Door te analyseren hoe de omgeving zich door de camerabeelden beweegt, kan het voertuig zijn traject schatten, en bijgevolg ook zijn huidige positie. De schatting van de eigen beweging op basis van camerabeelden wordt *visuele odometrie* genoemd, en zal het onderwerp zijn van dit doctoraat.

Visuele odometrie is geen nieuw onderzoeksdomein; het bestaat al meer dan 25 jaar. Tot op heden hebben de oplossingen uit de literatuur echter nog niet geleid tot een praktisch inzetbaar systeem met een acceptabele prijs voor de autosconstructeurs. De compromissen tussen nauwkeurigheid, complexiteit en uitvoerbaarheid betekenen dat, met uitzondering van prototypes, visuele odometrie geen implementatie vindt in massaproductievoertuigen. Deze thesis legt de basis voor een praktisch haalbaar visueel odometriesysteem gebaseerd op de combinatie van een enkele camera met een offline kaart, dat in goede weersomstandigheden een nauwkeurigheid bereikt competitief met satellietnavigatie.

Om dit mogelijk te maken, wordt het inherent tweedimensionaal karakter van de rijweg benut. Door de wegsituatie te aanzien als een vlak in plaats van een driedimensionale scène, vallen veel van de problemen weg die tot nu toe visuele odometrie ervan weerhouden hebben door te breken als standaardtechnologie. Deze planaire benadering heeft echter zelf twee inherente uitdagingen. De eerste is het probleem van datavervuiling door niet-vlakke structuren. Objecten die boven het grondvlak uitsteken, geven aanleiding tot foutieve datapunten omdat ze niet voldoen aan de veronderstellingen. De tweede uitdaging is nauwkeurigheid: de plaatselijke kromming van het wegdek zal leiden tot fouten op de geschatte positie. Beide uitdagingen worden aangepakt met een nieuw algoritme voor het tracken van kenmerkende punten in het grondvlak, gebaseerd op de predictieve onzekerheid rond de voertuigbeweging en de geometrische onzekerheid rond de kijkhoek van de camera ten opzichte van het grondvlak. Meer specifiek wordt het kinematisch model van het voertuig gebruikt om zoekruimtes af te bakenen in het grondvlak steunend op eerdere observaties, en wordt de tweedimensionale onzekerheid op de rol- en hellingshoek van de camera geprojecteerd tot perspectiefonzekerheidsregio's, ook in het grondvlak. Een stemalgoritme geïnspireerd door de Houghtransformatie verzekert een nauwkeurige schatting van de bewegingsparameters (snelheid en stuurhoek), die dan gebruikt worden om de relatieve beweging van het voertuig te berekenen.

Deze schatting van relatieve beweging is voldoende om op korte termijn het traject van het voertuig te reconstrueren. Wanneer de startpositie en -orientatie gekend zijn, is de huidige positie van het voertuig ook bij benadering bekend. Het proces is echter gevoelig aan foutopstapeling. Opeenvolgende kleine schattingsfouten kunnen grote positie-afwijkingen veroorzaken op de lange termijn. Dit werk gebruikt een extended Kalman filter om de onzekerheid op de globale schatting bij te houden op elk tijdstip. De langetermijnafwijking kan weggewerkt worden door extra informatiebronnen in te koppelen die wel absolute referenties geven. Een eerste mogelijkheid is satellietnavigatie wanneer die voorhanden is. Een interessantere optie is een offline kaart. In dit proefschrift wordt aangetoond dat een hidden Markov model de positionele foutopstapeling volledig kan elimineren en een precieze positie kan bepalen zelfs na zeer lange trajecten.

De evaluatie van de voorgestelde methoden demonstreert duidelijk dat de combinatie van visuele odometrie met een eenvoudige offline kaart een nauwkeurigheid kan bieden hoger dan die van de huidige industriestandaard van satellietnavigatie.

Het ontwikkelde raamwerk laat ook toe om het algoritme voor positiebepaling op basis van visuele odometrie en kaartgegevens eenvoudig te combineren met andere sensoren, bijvoorbeeld satellietnavigatie, een magnetisch kompas of de snelheidssensoren op de wielen van het voertuig, om zo een nog hogere nauwkeurigheid te bekomen.

English summary

For the past 30 years, innovation in road vehicles has been mostly driven by the rise of electronics. Electronics penetrate every aspect of a modern car, from control (e.g. drive-by-wire throttle instead of mechanical linkages) over emissions (starting with electronically controlled fuel injection) and diagnostics to safety (e.g. airbags and ABS). As technology brings us ever closer to the mass-produced autonomous vehicle, this latter aspect is rapidly becoming the limiting factor. How can we guarantee safety in the complex, fast-moving road environment?

A first step towards autonomous vehicles is active safety. Modern vehicles can take active steps to avoid or mitigate accidents. One example is the automatic pretensioning of seatbelts to optimize the impact point with a deploying airbag. Another is automatic braking using radar. Much more is possible in this field however, but is being held back by one limitation: lack of positional awareness. Vehicles can only achieve more autonomy when they know where they are, and where everything else is. This thesis focuses on the first issue: how can the car know where it is? One way for a vehicle to estimate its position is through satellite navigation. This is a very attractive solution for non-critical applications, which is why consumer navigation systems have become so commonplace. For applications related to safety however, satellite navigation is not dependable enough. Without line of sight to a number of satellites, positioning is not possible. Such situations are relatively common: in tunnels or under bridges, but also on narrow streets or between skyscrapers in large cities. Related to the satellite visibility problem is the problem of reflections. When satellite signals reach the receiver indirectly (e.g. after reflecting off a building facade), this can cause positioning errors ranging from meters to kilometers. Finally, signal attenuation and atmospherics can cause temporary increases in position uncertainty. Clearly, one or more backup systems are necessary.

While radar, infrared sensors, laser scanners and other devices can certainly aid the car or its driver in sensing the road environment, each of these technologies also has significant downsides. Radar has insufficient resolution for accurate positioning, and typically a narrow field of view. Infrared sensors lack sharpness, especially in relatively homogeneous surfaces such as the road. Laser scanners are still very expensive, and difficult to integrate inconspicuously in a road car.

Much of the road infrastructure is designed for a different kind of sensor: the driver's eyes. Because the road context is filled with visual clues, car manufacturers are increasingly equipping their models with cameras to interpret the surroundings. Examples include traffic sign recognition systems and park-assist cameras.

These cameras offer opportunities for positioning as well. By looking at how the world moves through the camera images, the vehicle can learn where it has been, and hence know where it is now. The act of estimating one's own motion with a camera is called *visual odometry*, and this will be the topic of this dissertation.

Visual odometry is not a new research field; in fact it has been around for over 25 years. Up to now however, the approaches found in literature have fallen short of being practically feasible at a price suitable for large scale integration by car manufacturers. The trade-offs between accuracy, complexity and practicality mean that short of prototypes, visual odometry is not currently used in mass produced cars. This thesis provides the basis for a practically feasible fair-weather solution based on a single camera, which combined with offline map data can surpass the accuracy provided by satellite navigation.

The way we achieve this is by exploiting the inherently two-dimensional nature of the road surface. By treating the surroundings as a plane instead of a full 3D scene, many of the challenges which have so far prohibited visual odometry from becoming a mainstream technology are avoided altogether. This approach has two problems of its own however. The first problem is the outlier problem: any non-planar structure will give rise to erroneous data points because it violates the assumptions. The second problem is accuracy: local road curvature causes errors on the estimated position. Both problems are dealt with efficiently by a novel ground plane feature tracking algorithm based on predictive uncertainty of the vehicle motion and geometric uncertainty on the viewing angle of the camera with respect to the ground plane. Specifically, the kinematic model of the vehicle is used to project ground plane search regions corresponding to earlier feature observations, and the two-dimensional uncertainty on the roll and pitch of the vehicle is projected to perspective uncertainty regions in the ground plane as well. A parameter space voting algorithm inspired by the Hough transform then ensures an accurate estimation of vehicle motion parameters (velocity and steering angle), which are used to compute the relative motion of the vehicle.

While the relative motion estimate is sufficiently accurate to track vehicle position in the short term (provided its start position and orientation are known), the process is inherently susceptible to drift. The accumulation of errors over time can cause large deviations in calculated position in the long term. We employ an extended Kalman filter to keep track of the expected accuracy of the estimate at each time. The long-term drift problem can be solved by using additional information sources providing absolute world position references. One option is the fusion with satellite navigation when it is available. More interesting however is the integration with an offline map. We show how a hidden Markov model can completely eradicate the positional drift and provide a precise estimate even after very long travel distances.

Evaluation of the proposed methods on realistic data sets clearly demonstrates that in conditions of good visibility, the combination of single-camera visual odometry and simple offline map data can provide accuracy surpassing that of the current industry standard of satellite navigation.

The developed framework allows for easy combination of the positioning algorithm based on visual odometry and map data with other sensors, e.g. satellite navigation, a magnetic compass or the speed sensors on the wheels of the vehicles, to attain even higher accuracy.

1
Introduction

1.1 Problem statement

In Europe, more than one motor vehicle is registered for every two people (ACEA [5], Eurostat [29]). The average Belgian employee spends over 6 hours per week driving or being driven around (Glorieux et al. [36]). No one needs to be convinced of the massive challenges all this motoring presents for infrastructure, well-being and safety. Two approaches are being followed to deal with these challenges. The first approach is reducing the number of vehicles simultaneously on the road. Flexible working hours, working from home, carpooling and public transport are all efforts towards this goal. The second approach is not to reduce motoring, but to *improve* it. Emissions reductions, infrastructure works and vehicle safety are examples of this approach.

It is within this second approach that this thesis is situated, namely in the context of *intelligent vehicles*. Since the inception of the mass produced consumer car in 1901, cars have steadily become more intelligent in the sense that they simplify tasks for the driver, provide the driver with extra information, mitigate the consequences of accidents and, more recently, take active steps to avoid accidents (e.g. radar triggered braking). In the last 10 years, this evolution of cars on the electronics and software front has accelerated sharply, with even entry level models providing several driver aids that were the sole privilege of buyers of flagship models only a couple of years ago. Traffic sign recognition and reverse parking cameras are two examples of this rapid evolution. More important than these convenience features however, are developments in the field of *active safety*. It is no secret that many accidents can be directly attributed to human factors: judgment errors, slow reactions, inattentiveness and poor spatial awareness are all potential causes for accidents or aggravating factors in case of an accident. So far, development in this area has been frustratingly slow. Automatic braking systems fall under this category, as does (arguably) adaptive cruise control, and both have been adopted to some degree, but the reality is that these systems are still far from dependable and do not in their current state make a meaningful contribution to road safety. The next step onwards from active safety is autonomous driving, where the car itself takes over all of the driver's tasks and fully controls itself. The emergence of the first autonomous vehicles on the road (notably Google's self-driving car and Tesla's autopilot function) is testament to the progress which is being made in the field of intelligent vehicles, however the technology used on these prototypes is still a long way from being feasible for mainstream application, and the contexts in which they are able to function remain limited for both legal [2, 25, 73] and technical [38] reasons.

Why is it that it proves prohibitively difficult to make a vehicle take driving decisions by itself? The answer is a lack of *context perception*. A human driver has at any given time a wealth of information supporting his driving decisions: road layout, relative positions of vehicles, road marks, prior knowledge about intersections, weather conditions, even the time of day, all this context informs the driver and influences his decision process. A lot of this information is related to *position*: where am I and where is everything else I need to know about. Any effective active safety system will require accurate and reliable position information if it is to make similarly informed decisions.

The most widely implemented positioning technology in road vehicles is *GPS*. *Global Positioning System* was the first global satellite navigation system (*GNSS*) available to consumers. Based on a swarm of satellites with synchronized clocks circling the earth, GNSS allow users to know their position in absolute world coordinates by comparing the differences in reception delays. At least four different satellites must be received at any time to allow for a positional fix, on account of the four dimensional nature of the problem (three dimensional position plus time). The accuracy of this position measurement varies from one meter to tens of meters depending on environmental factors. Roadside buildings can reflect the satellite signals, leading to timing errors or receiver confusion due to multiple receptions. Foliage cover can reduce signal strength [86]. Atmospheric disturbances cause timing problems [16]. Then there is the factor of immediate satellite constellation. The number of satellites visible to the GNSS receiver depends on location, date and time of day. More satellites generally allows for a more accurate position estimate, but the "spread" of the satellites is also important. Satellites which are observed closer to the horizon contribute more accuracy, but are also more likely to be obscured by buildings or geographic features.

Certain technologies have been developed with the goal of improving the overall accuracy of the GNSS position estimates. High Sensitivity GPS receivers implement more advanced signal processing hardware and software to mitigate the influence of multi-path effects and improve reception of highly attenuated signals. Differential GPS (DGPS) uses ground stations that broadcast the current difference between their known, fixed position and their own GPS estimate to correct the offset of nearby receivers. In theory, this technology reduces the expected position error to within one meter. However, sometimes the environment simply does not allow for a position to be obtained despite this technology, for example when driving through *urban canyons*: streets with high-rise buildings on both sides. In such cases, often too much of the sky is occluded to allow the reception of four satellites [48]. As a consequence, applications depending exclusively on satellite communication for positioning cannot guarantee full-time availability.

Clearly the problems mentioned above mean that GNSS by itself is neither accurate or dependable enough to provide position information for active driver safety technology. Additional sensors are required to complement the GNSS information. Clear proof of this is in the emergence of the first experimental autonomous vehicles: these are laden with tens of thousands of euros worth of sensors to perceive the environment.

An obvious candidate for an extra sensor is *vision*; after all this is the modality through which the vast majority of road infrastructure was designed to be perceived. Indeed consumer vehicles are increasingly being equipped with cameras to aid the driver: lane departure warning, road sign recognition and rear parking are examples of driving tasks supported by cameras. The aim of this thesis is to add absolute positioning to this list. Specifically, we will provide the algorithms to reconstruct the trajectory of the vehicle by analysis of the video stream captured by a front or rear facing camera, and to relate this to a stored map of the road network. Additionally, we will describe a framework to incorporate other positioning sensors (among which GNSS) when they are available.

1.2 Background & Literature

The measuring of a vehicle's trajectory by analysis of video captured by a vehicle mounted camera is called *visual odometry* (from the Greek word for road, *hodos*). It is closely related to what is called Simultaneous Localization 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 salient points in the environment of the vehicle, but for visual odometry the mapping itself 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 short distances, the drift 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) [63, 76]. As such, visual odometry is still a prime candidate to supplement satellite navigation. Chapter 4 will expand on how to use the road network to eradicate drift.

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? The topic of calibration will be covered in Chapter 3. 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 [60]. Note that this means that point correspondences between the two views must be established. Typically this will be done by detecting salient points in both viewpoints, computing a feature descriptor on the local neighborhood of the points, and then matching the descriptors from one view to the other. Typically some spurious matches between descriptors will occur, especially if the camera displacement was significant between the two viewpoints. A computationally efficient solution for essential matrix estimation from slightly polluted feature correspondences was published in the 1990s by Philip [78] and improved upon by Nistér [72]. 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 the 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, or it can be used as-is to compute consequent camera positions for a video sequence.

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* [43]. 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 [98]. 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 [8, 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. Aditionally, they rely on very accurate calibration on account of the long observation distance to baseline width ratio [3]. In the vibration and shock-prone environment of a car, it is generally accepted that long-term calibration stability cannot be guaranteed, and on-line recalibration methods have been proposed [20, 67] 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. Coupled with the lower cost and ease of integration, monocular solutions are far more attractive to vehicle manufacturers than stereo configurations. In addition, monocular cameras are already present on many new cars, so the additional hardware cost is even lower; only the processing hardware needs to be added.

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.* [94] and Scaramuzza [87, 88]. However, these methods are only demonstrated using an omnidirectional camera mounted atop the vehicle, which is again not practical for application on consumer vehicles. More relevant is the work of Chandraker and Song [91]. In this work, a 5-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 4-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 to other base odometry algorithms as well in the future.

Recently, a different approach to monocular visual odometry and SLAM 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 [27, 28, 92, 101]. 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. A few recent methods have attempted to apply the principles of direct visual odometry specifically to the planar road scenario, using image distance metrics such as mutual information to estimate pose change when viewing a planar surface[40, 104]. While this produces satisfactory results in the presented use cases, the applicability to general road driving applications where the view of the road can be obstructed by other road users or roadside objects remains to be proven. Some of the challenges of direct planar visual odometry will be highlighted in Chapter 2.

1.3 Research contributions

In this work, we will present 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 and the corresponding uncertainty on the ground plane coordinates of a feature point detected in the camera's perspective image. 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 and ensures good results are still obtained when the ground plane is not perfectly planar, as will often be the case in practice.

Two key components of the method provide robustness against the common problem of outliers. Firstly a feature matching method constrained by uncertainty zones reduces the likelihood of false matches, and eliminates the need for computation and matching of feature descriptors. Secondly a Hough-like parameter space vote is used to extract a consensus from the matched features while still being tolerant of small feature point inaccuracies (e.g. caused by height variations in the road surface). The combination of these two mechanisms eliminates the need for a RANSAC scheme and speeds up computation, while still producing useful odometry for very low inlier ratios in real-world experiments.

The focus of the presented method is on usability; some accuracy is sacrificed to benefit robustness and computation speed. Nevertheless, we demonstrate that the basic concept of two-dimensional odometry using only the approximate ground plane is capable of outperforming basic fundamental matrix estimation in a real-world scenario, even if the maximum accuracy of the best state-of-the-art methods cannot be matched in scenarios where computation time is unlimited or in the case of perfect lab conditions.

To further pave the way for the implementation of visual odometry for positioning purposes in road vehicles, a framework is presented for sensor fusion based on an Extended Kalman Filter (EKF). This allows for the easy incorporation of various other sensors into the system, including GNSS measurements, magnetic compass, inertial sensors and wheel odometry. Experimental results show a marked increase in accuracy is possible when using visual odometry in addition to GNSS positioning.

Finally, this framework is adapted to also make use of off-line maps of the local road network to eliminate drift even in the total absence of any external communications. To this end, a Hidden Markov Model (HMM) is implemented to entertain and evaluate multiple position hypotheses and at each point in time robustly estimate the most likely map position.

A block diagram of the complete odometry solution is shown in Figure 1.1.

The end result is a complete solution for fair-weather visual odometry for road vehicles. This technology can be used to complement existing positioning technology in cases where reasonable video quality can be obtained. Scenarios that are yet to be thoroughly investigated are harsh weather conditions, night driving and off-road applications.

1.4 Publications

1.4.1 Publications in international journals

The work described in this thesis has been published in two articles in peer-reviewed international journals, one of which as first author.

- **Robust monocular visual odometry for road vehicles using uncertain perspective projection**, Van Hamme, David; Goeman, Werner; Veelaert,

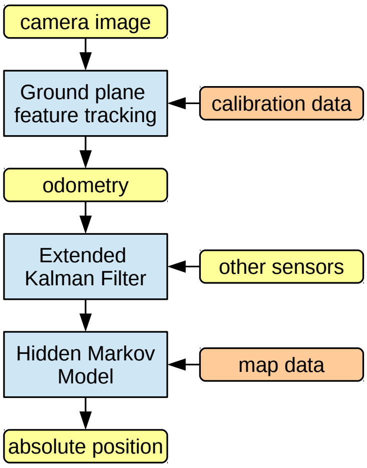


Figure 1.1: *Overview of the proposed odometry solution. Other sensor data is optional.*

Peter; Philips, Wilfried, *EURASIP Journal on Image and Video Processing (2015)*, Vol. 2015:10 pp. 1-23.

- **Cycling around a curve : the effect of cycling speed on steering and gaze behavior**, Vansteenkiste, Pieter; Van Hamme, David; Veelaert, Peter; Philippaerts, Renaat et al., *PLOS One (2014)*, Vol. 9:7 pp.1-11.

1.4.2 Publications in international conferences

This dissertation has resulted in 13 peer-reviewed publications presented at international conferences, seven of which as first author.

- **Robust matching of occupancy maps for odometry in autonomous vehicles**, Dimitrievski, Martin and Van Hamme, David and Veelaert, Peter and Philips, Wilfried, *11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (2016)* Vol. 3 pp. 628-635.

- **Lane identification based on robust visual odometry**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *IEEE International Conference on Intelligent Transportation Systems-ITSC (2013)*, pp. 1179-1183.

- **Parameter-unaware autocalibration for occupancy mapping**, Van Hamme, David; Slembrouck, Maarten; Van Haerenborgh, Dirk; Van Cauwelaert, Dimitri et al., *2013 Seventh international conference on distributed smart cameras (ICDSC) (2013)*, pp. 49-54.

- **Communicationless navigation through robust visual odometry**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *IEEE International Conference on Intelligent Transportation Systems-ITSC (2012)*, pp. 1555-1560.

- **Robust monocular visual odometry by uncertainty voting**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *IEEE Intelligent Vehicles Symposium (2011)*, pp. 643-647.

- **Robust visual odometry using uncertainty models**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *Lecture Notes in Computer Science (2011)*, Vol. 6915 pp. 1-12.

- **Fire detection in color images using Markov random fields**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried; Teelen, Kristof, *Lecture Notes in Computer Science (2010)*, Vol. 6475 pp. 88-97.

- **Foliage recognition based on local edge information**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried; Teelen, Kristof et al., *Lecture Notes in Computer Science (2008)*, Vol. 5259 pp. 838-849.

- **A Visual SLAM system with mobile robot supporting Localization services to visually impaired people**, Nguyen, Quoc Hung; Vu, Hai; Tran, Thanh-Hai; Nguyen, Quang-Hoan; Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *Lecture Notes in Computer Science (2014)*, Vol. 8927 pp. 716-729.

- **Edge based foreground background estimation with interior/exterior classification**, Allebosch, Gianni; Van Hamme, David; Deboeverie, Francis; Veelaert, Peter et al., *Proceedings of the 10th International Conference on Computer Vision Theory and Applications (2015)*, pp. 369-375.

- **Self-learning voxel-based multi-camera occlusion maps for 3D reconstruction**, Slembrouck, Maarten; Van Cauwelaert, Dimitri; Van Hamme, David; Van Haerenborgh, Dirk et al., *International Conference on Computer Vision Theory and Applications, Proceedings (2014)*, pp. 502-509.

- **Time complexity of traditional vision algorithms on a block-based image processor (BLIP)**, Slembrouck, Maarten; Heyvaert, Michaël; Van Cauwelaert, Dimitri; Van Hamme, David et al., *2012 Sixth international conference on distributed smart cameras (ICDSC) (2012)*, pp. 1-6.

- **Combining geometric edge detectors for feature detection**, Heyvaert, Michaël; Van Hamme, David; Coppens, Jonas; Veelaert, Peter, *Lecture Notes in Computer Science (2010)*, Vol. 6474 pp. 221-232.

1.5 Outline

Chapter 2 will explain the ground plane feature tracking method which forms the general concept behind the proposed method. The geometrical relations between the camera images and the ground plane will be further explored in Chapter 3: Calibration. The use of an offline map to obtain drift free absolute positioning will be explained in Chapter 4. Finally, global conclusions will be drawn in Chapter 5.

This dissertation is structured as follows. In Chapter 2, the core visual odometry method that forms the core of this research is presented. The chapter starts with a summary of the traditional approach using epipolar geometry (Section 2.2), which highlights the challenges and motivates our choice for ground plane feature tracking instead. The proposed method is then described in four parts. Feature detection (and a performance comparison of common feature detectors applied to the road tracking context) are explained in Section 2.3.1. The conversion of image coordinates to world ground plane coordinates by means of inverse perspective correction forms the second part (Section 2.3.2). Thirdly, Section 2.3.3 explains how features are matched and tracked over consecutive video frames using a kinematic model of the vehicle. Finally, the calculation of the odometry parameters from the feature tracks forms Section 2.3.4. The method is compared to a traditional 8-point solver from literature in Section 2.4, with a detailed analysis in Section 2.5.

Our visual odometry method requires the camera to be calibrated both intrinsically (relating to its internal properties) and extrinsically (position and orientation). Both aspects are thoroughly investigated in Chapter 3. Section 3.2 briefly summarizes standard theory of intrinsic calibration and investigates its real-world accuracy and repeatability through practical experiments. Section 3.3 describes the extrinsic parameters required for our method, analyzes the sensitivity of our method to calibration errors in these parameters, and presents two algorithms to determine the parameters in practice.

Chapter 4 describes a framework in which the proposed odometry method is combined with other sensors and map data into a complete solution for absolute vehicle positioning on the road network. Section 4.2 shows how the positional error distribution can be estimated using an extended Kalman filter. The combination with other immediate information sources (e.g. satellite navigation) is explained in Section 4.3. Localization to the nearest road is achieved using a hidden Markov model (Section 4.4) and a feedback loop to the extended Kalman filter (Section 4.5) limits the posterior error distribution. The proposed framework is evaluated and analyzed on real data in Sections 4.6 and 4.7.

Finally, Chapter 5 summarizes the main results achieved and conclusions reached in this PhD.

2
Ground plane feature tracking

2.1 Introduction

This chapter will explain the core concept of our visual odometry method: ground plane feature tracking. As we discussed in the introduction, the traditional approach to visual odometry differs from ours in this respect. For decades, methods in literature depended on fundamental or essential matrix estimation, for example Geiger et al. [35], Kitt et al. [53], Longuet-Higgins [60], Nister [72], Philip [78], Song and Chandraker [91], Torr et al. [98]. In Section 2.2 we will give a quick overview of the theory and algorithms behind the classical approach in order to highlight its inherent challenges which our method will avoid. For a more detailed description, Hartley and Zisserman [43] remains the reference work. Section 2.3 will describe the advantages and challenges of limiting computations to the ground plane, and Sections 2.3.1, 2.3.2 and 2.3.3 will explain in detail the feature extraction and tracking, and the calculation of odometry parameters from the feature tracks. This defines the core algorithm, after which Section 2.3.1.1 will compare suitability of the most common feature extraction methods for this algorithm.

In order to understand the classical approach and its problems, let us first analyze the imaging process, i.e. the working principles of the camera. The camera *projects* a part of the 3D world onto a 2D coordinate system. The mechanics through which this happens are fairly straightforward. A light source, for example the sun, illuminates the molecules in the 3D scene. The molecules reflect part of this light, usually scattering it in a wide range of directions. The light rays which are reflected in the direction of the camera lens, pass through this lens and illuminate a part of the image sensor behind. In cameras of a bygone era this was photographic film, in which the chemical properties of the film changed proportional to the amount of light exposure. In modern cameras the film is replaced by electronic devices which convert light into electrical charge; essentially light counters with many individual cells. Both technologies perform the same function: measuring the light which entered the camera from a particular direction. This is identical to the function of the human eye: light coming from different directions is cast onto different parts of the retina, with the photo receptor cells in the retina measuring the light intensity.

The selectivity of direction is paramount to the quality of the image: when the light from multiple directions is cast onto the same spot on the image sensor, the imaging process loses its distinctive properties and becomes less useful. A simple way to ensure only light coming from a specific direction makes it onto the imaging sensor is placing a non-translucent board in front of the sensor, and making a small hole into the board. For each point on the image sensor behind the board, the only rays which land on that point must have gone through the hole. It can easily be seen that the point and the edges of the hole define a cone-like structure in the 3D world delimiting which rays of light project onto this point. In the ideal case, the board has infinitesimally small thickness and size. The sensor must then posses infinitely high sensitivity. This situation is called the *pinhole camera model*. Obviously this is impossible; the attainable sensitivity of an electronic sensor is governed to a large extent by the signal to noise ratio of its readout circuitry. As a result, the hole must be made larger, but this then degrades the orientation selectivity and blurs the image. The solution is the use of a lens: an optical device which will bend parallel rays of light in such a way that they still end up on the same spot on the sensor, while ensuring that non-parallel rays land on different points, like they would if the camera had an infinitesimally small hole instead of the lens. Essentially the lens performs the function of the pinhole but allows much more light through. More details about the compromises made in an optical lens will be given in chapter 3. For now it suffices to note that there is a point, called the *focal point* or *camera center* in Hartley and Zisserman [43], which the rays go through as if it were a pinhole camera.

Although the pinhole camera model is adequate to understand the fundamental imaging process, in practice several imperfections complicate making high quality pictures and video. Light diffraction at the edges of the aperture (i.e. the pinhole or the lens opening) is one problem. This can cause blur for very small apertures. A second consideration is depth of field. The properties of a physical camera lens dictate that only objects at a certain distance (the focal distance) are perfectly sharp, and objects closer or further than this distance will become gradually less

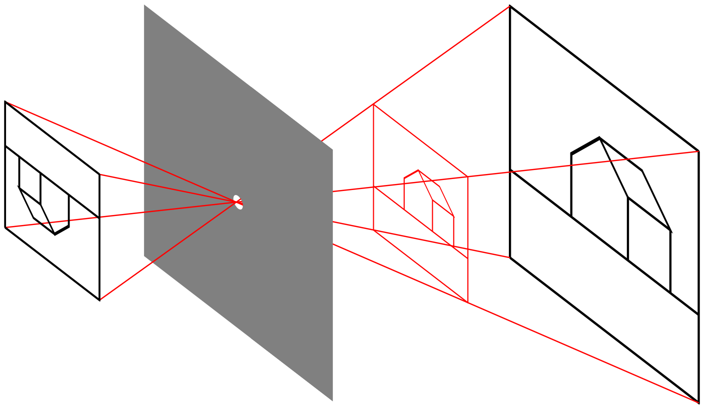


Figure 2.1: Pinhole camera model. The sensor plane (left side, black) is behind the pinhole, but it is often drawn in front of the pinhole (red) as this is mathematically equivalent but more intuitive because the picture is not inverted horizontally and vertically.

sharp. Depth of field is a measure of the span of distances which will still be acceptably sharp. It depends on both lens properties and image sensor size. A final concern is motion blur: if the scene or camera is moving during the time the image sensor is collecting light, then the image will become smeared out in the direction of motion. All three of these concerns will be briefly addressed when discussing the hardware used for our experiments in Chapter 4.

Let us now take a closer look at the properties of the pinhole projection model. The transformation from the 3D world to the 2D image at first sight seems very unconstrained. Neither angles, distances nor ratios or proportions are necessarily preserved through the imaging process. Right angles in the real world (e.g. the corners of a house facade) may become acute or obtuse in the picture depending on how the camera looks at them. Parallel lines in the real world are rarely parallel in the image. Proportions are only preserved between objects at the same distance to the camera. One thing however is preserved: the straightness of lines. Mathematically, camera projection can easily described in homogeneous coordinates. Let $\mathbf{X} = [X\ Y\ Z\ 1]^T$ denote the homogeneous coordinates of a 3D world point (often called *object point*). The camera effects a mapping to a homogeneous 2D *image point* $\mathbf{x} = [x\ y\ 1]^T$ described by

$$w\mathbf{x} = [wx\ wy\ w]^T = \mathbf{P}\mathbf{X}. \tag{2.1}$$

The 3x4 matrix $\mathbf{P}$ is called the projection matrix. $\mathbf{P}$ depends on the camera parameters (e.g. focal distance) as well as the camera location (specifically the 3D world coordinates of the camera center) and orientation relative to the world coordinate axes.

2.2 Epipolar geometry

Suppose we have two different cameras in different places imaging the same object point. We know nothing specific about the cameras, except that they comply to the pinhole model. Hence the only information available to us is that for each camera the camera center, the object point and the point on the sensor where it is imaged are collinear. This in turn means that the camera centers, the object point and the projections on the two sensors are coplanar. This situation is illustrated in Figure 2.2. The plane through these points is called the *epipolar plane*.

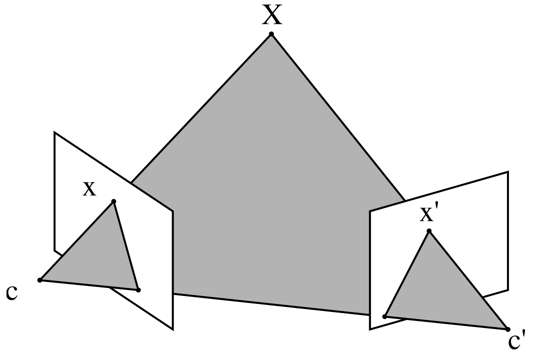


Figure 2.2: *Illustration of the epipolar 2-view geometry. The object point is denoted by* $\mathbf{X}$*, camera centers by* $\mathbf{c}, \mathbf{c}'$ *and projected points by* $\mathbf{x}, \mathbf{x}'$*. All five points lie in the epipolar plane, drawn in gray.*

Moreover, it can easily be seen that all the points on the line through the camera center and the projected point of the first camera project onto a single line in the second camera's image plane; this line is the intersection of the epipolar plane with the image plane and is called an *epipolar line*. For each location of the image point $\mathbf{x}$ there is an epipolar line in the second camera's image plane, and the corresponding image point $\mathbf{x}'$ must lie on this line irrespective of the precise location of the original object point $\mathbf{X}$. This relation is expressed mathematically as

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \tag{2.2}$$

in which $\mathbf{Fx}$ yields the coefficients of the epipolar line (in homogeneous coordinates). Since the 3x3 matrix $\mathbf{F}$ describes a mapping from a point to a line, it must be of rank two and has seven degrees of freedom (since it is determined up to scale only). $\mathbf{F}$ is called the *fundamental matrix*.

The relationship in Equation 2.2 is stated between image coordinates of points in the two cameras corresponding to the same object point. When a set of at least seven point correspondences is given, the fundamental matrix can generally be computed from Equation 2.2; every correspondence gives rise to one linear equation in the unknown elements f_{ij} of $\mathbf{F}$, which has seven degrees of freedom. With $\mathbf{x} = [x, y, 1]$ and $\mathbf{x}' = [x', y', 1]$ each linear equation is of the form

$$x'xf_{11}+x'yf_{12}+x'f_{13}+y'xf_{21}+y'yf_{22}+y'f_{23}+xf_{31}+yf_{32}+f_{33} = 0. \quad (2.3)$$

For a set of n point matches this can be written as

$$\mathbf{Af} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = 0, \quad (2.4)$$

where $\mathbf{f}$ is the 9-vector formed by the elements of $\mathbf{f}$ in row-major order. If the point correspondences are not in a degenerate configuration, the matrix $\mathbf{A}$ must have rank eight and a solution can be found by linear methods (the solution is the generator of the right null-space of $\mathbf{A}$. In practice, the image points x_i and x'_i will be noisy and discretized, and the rank of $\mathbf{A}$ will be nine. The least-squares solution for $\mathbf{f}$ is then found as the right singular vector corresponding to the smallest singular value of $\mathbf{A}$. Care must be taken about data conditioning of this SVD problem; entries in $\mathbf{A}$ of the form $x'x$ for example can easily be a factor 10^6 larger than its smallest elements. Details about appropriate scaling of the data to ensure numerical stability can be found in Hartley [44].

What we just described is essentially what is called the *normalized 8-point algorithm* or 8-point solver; it was first described in Longuet-Higgins [60] as a way to triangulate points from their projections to two viewpoints.

For the application of visual odometry, the full triangulation of the points is not required; it is only implicitly used to determine the motion of the camera between consecutive observations of the scene. The two cameras we used to construct the fundamental matrix are in fact the same camera at different points in time. Note that in the above discourse, no assumptions of any kind have been made about the cameras, apart from the adoption of the pinhole camera model. The projection matrix $\mathbf{P}$ as defined in Equation 2.1 has not appeared in the mathematics. We will now briefly explain the relation between the projection matrix, the fundamental matrix and the transformation matrix that relates the two viewpoints.

Recall that we mentioned the camera matrix $\mathbf{P}$ depends on both the internal camera parameters as its position and orientation. Specifically, the camera matrix decomposes as

$$\mathbf{P} = \mathbf{C}\left[\mathbf{R}|\mathbf{t}\right], \quad (2.5)$$

in which $[\mathbf{R}|\mathbf{t}]$ is the rotation matrix $\mathbf{R}$ that aligns the world axes with the camera axes, augmented by the translation vector $\mathbf{t}$ between the origin of world axes and the camera center, expressed in camera coordinates. $\mathbf{C}$ determines the scaling and offset of the image coordinates:

$$\mathbf{C} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.6}$$

where α_x and α_y are the horizontal and vertical focal lengths respectively and (x_0, y_0) the coordinates of the *principal point* (i.e. the projection of the ray formed by the Z axis). These values are called the *intrinsic parameters* of the camera, and $\mathbf{C}$ is therefore also called the *intrinsic camera matrix.*

We may then introduce the concept of a *normalized camera* by applying the inverse of $\mathbf{C}$ to the image point $\mathbf{x}$, obtaining the point $\hat{\mathbf{x}} = \mathbf{C}^{-1}\mathbf{x}$ in *normalized image coordinates*. The camera matrix of this normalized camera is then reduced to $\mathbf{C}^{-1}\mathbf{P} = \mathbf{R}|\mathbf{t}$. The fundamental matrix which relates two normalized cameras is called the *essential matrix* and it is governed by the equation

$$\hat{\mathbf{x}}\mathbf{E}\hat{\mathbf{x}}' = 0. \tag{2.7}$$

Its relationship to the fundamental matrix is described by

$$\mathbf{E} = \mathbf{C}'^{\mathbf{T}}\mathbf{F}\mathbf{C}. \tag{2.8}$$

in which $\mathbf{C}$ and $\mathbf{C}'$ are the intrinsic camera matrices of the two cameras. The essential matrix is obviously more constrained than the fundamental matrix; it depends merely on the relative orientation and position of the cameras. It has five degrees of freedom: three for the rotation and three for the translation, diminished by one since it is only determined up to scale. The essential matrix may be determined from the normalized image coordinates by singular value decomposition similar to the solution of Equation 2.4 (as is for example the case in Nister [72]), or it may be derived from the fundamental matrix (as in Geiger et al. [35]).

The essential matrix by itself is often sufficient for visual odometry applications as it completely represents the coordinate transform between two consecutive frames, and the total transform over a set of frames is simply the multiplication of the incremental transforms. If necessary however, it can be decomposed into the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$, by factoring it into a skew-symmetric matrix and a rotation matrix as described by Hartley and Zisserman [43].

While the theory of epipolar geometry was proposed decades ago, its application to solve the problem of monocular visual odometry has remained difficult in practice. Some challenges are inherent to the theory and the algorithms of fundamental or essential matrix estimation. One such challenge is scale ambiguity. The projective transforms are only determined up to scale by the equations. For stereo cameras, the scale ambiguity is easily resolved by taking into account the baseline distance (i.e. the distance between the camera centers of the stereo pair), a measurable fixed value. In the monocular instance, the situation is more difficult. The scale ambiguity for a road may be resolved by measuring a real-world distance which is stable, e.g. the height of the camera center with relation to the ground plane, which can be estimated by triangulating some point or points close to the vehicle (this is the technique used by Geiger et al. [35]). However, the reconstruction of motion paths remains inherently liable to *scale drift* as a result of the multiplication of consecutive transform estimations. This is a well acknowledged problem, and solutions have been proposed based on loop closing and/or map matching as well as continued efforts to minimize inter-frame scale errors (Clemente et al. [18], Eade [26], Pinies and Tardos [79], Williams [102]).

A second weakness of fundamental matrix estimation is that its accuracy depends on distance between the two camera positions. This makes sense intuitively; if the distance between camera centers is very small compared to the distance of the observed object points, the triangulation of the points will be poor. This corresponds to the relationship between baseline and depth resolution of a stereo camera system. The requirement of significant translation means that it is pointless to compute frame-by-frame visual odometry at slow speeds and high frame rates in this fashion; the results will only degrade as many low-quality estimations are multiplied to obtain a trajectory. In practice this is not a big issue for most applications; frames are usually simply discarded until the estimated translation exceeds a threshold. Similar concerns arise when feature correspondences originate from only a small region of the image, in which cases certain components of the inter-frame transform will become poorly determined.

Another problem in fundamental matrix estimation is the case of *motion degeneracy*. This occurs when the motion has fewer degrees of freedom than the 5 degrees of freedom of the essential matrix (e.g. pure translational motion). In the theoretical case, the degeneracy is apparent in the rank of $\mathbf{A}$, but it can become a problem in case of noise and outliers in the data (i.e. spurious point correspondences between views). Normally outliers are filtered by a sampling strategy such as RANSAC, in which a minimal set of random points is chosen to compute the model, and then the validity of that model is verified by evaluating a distance criterion on the other data points. The model that yields the maximum support set (i.e. the highest number of points within the distance threshold from the computed model) is then the best candidate solution and can be refined by for example a least-squares estimation on the entire support set. In the case of motion degeneracy however, the additional degrees of freedom mean that one or more outliers can be added without violating the distance criterion, yielding wrong but mathematically sound solutions with a support set larger than the correct solution. A more thorough explanation of motion degeneracy is given by Decker et al. [22]. Strategies to identify and cope with such situations have been proposed (Decker et al. [22], Goshen and Shimshoni [39], Yang et al. [105]), but they add unwanted complexity to the framework.

A fourth issue which may arise in fundamental matrix estimation is that certain configurations of object points give rise to multiple solutions for $\mathbf{F}$. For a mathematical analysis of which configurations this applies to, we refer the reader to Torr et al. [97]. Although the detection of such degenerate configurations may seem easy since they are known beforehand, in practice the presence of noise on the measured image points makes the detection of *scene degeneracy* significantly more difficult. One form of scene degeneracy which is particularly relevant for the application of visual odometry is outlined by Chum et al. [17]. It describes the case when a dominant plane is present in the scene, from which the majority of the feature correspondences originate. When fundamental matrix estimation is performed on a minimal set of five correct planar point correspondences and two outliers, this gives rise to an epipolar geometry which is wrong but supported by all correspondences originating from the same plane. The dominant plane scenario will regularly occur in a road context, for example on multi-lane highways, especially when the viewing angle of the camera is relatively narrow. Care must be taken regarding the distribution of point correspondences in the scene to avoid such wrong estimations, or filtering steps are required to eliminate inconsistencies in the motion path.

Finally, although not a problem exclusive to fundamental matrix estimation, the dependency on feature matching can also be a weakness. In order to obtain pairs of image points corresponding to the same object point, characteristic points (features) must be identified in both images. Feature descriptors must be computed and matched to each other, and similarity or repetitiveness of scene structures may cause feature confusion and yield the aforementioned outliers. It is especially this property which makes the monocular visual odometry case much more difficult than the stereo case, in which additional epipolar constraints significantly reduce the search space for feature correspondences.

From this list of challenges inherent to fundamental or essential matrix estimation, it should be clear that the search for alternative solutions holds merit. Literature has reflected this, notably with efforts by Scaramuzza and Siegwart [87], Tardif et al. [94] who considered the specific case of planar-only scene geometry. In order to conform the world to this flatness assumption, they used an omnidirectional camera which captures images of the road immediately surrounding the vehicle. While these solutions have been shown to work in practice, they are unattractive from a systems integration standpoint; they require either unsightly contraptions atop the vehicle or cumbersome to calibrate systems with many cameras.

In the last few years, a new family of SLAM-inspired methods have become the state of the art: the so called *direct* methods. The term direct refers to the use of image intensities in the computation rather than feature locations inferred from the image intensities, a concept first introduced for stereo and depth cameras by Comport et al. [19], Kerl et al. [51]. For its application in monocular visual odometry, this concept is applied in conjunction with a prior estimation step to establish some scene geometry, and we speak of *semi-direct* visuawerel odometry. Epipolar geometry lies at the base of the scene prior (Engel et al. [27], Forster et al. [32], Song and Chandraker [91], Stühmer et al. [92], Usenko et al. [99]). Image intensities belonging to patches of known depth are then used directly to accurately compute camera pose changes. As such, we will consider the (semi-)direct methods as an evolution of the fundamental matrix estimation methods, and many of the concerns listed above still apply despite the impressive accuracies obtained in many scenarios (notably in the emerging field of Micro Aerial Vehicles or MAVS, as in Faessler et al. [30]).

2.3 Ground plane feature tracking

In a general road driving context, few assumptions can be made about scene geometry. One constant however is the road itself. Vehicles drive on the road, and for most applications pertaining to intelligent vehicles it is sufficient to be able to tell how the vehicle is moving relative to the section of road directly below the vehicle. Rather than trying to avoid computational problems arising from scenery dominated by the road plane, we will design a method to specifically exploit this dominant plane. In this respect, our work will be similar to that of Tardiff and Scaramuzza [88, 94], who used omni-directional cameras to capture the motion of the ground plane below the vehicle.

Essentially, the working principle of these methods is similar to that of a computer mouse: register the image patch below the mouse correctly and the pointer will track the mouse motion nicely. While this concept is simple enough, some problems immediately arise in the case of road driving. Firstly, we do not have a sensor looking directly down on the road below. Such a sensor would be a high-maintenance item, much like the ball-equipped mice you used to find connected to beige computers of the previous millennium which systematically collected filth off the desk top. Additionally, such a sensor would be specific to the application, whereas ideally we would like to use multi-purpose cameras strategically positioned to capture more than just the road. Many new vehicles already come equipped with a forward facing camera (e.g. for road sign recognition) which could double as a visual odometry camera. This kind of camera orientation presents some challenges of its own however.

Firstly, the orientation of these cameras with relation to the road is only approximately known. As the car rocks around on its suspension, the angle changes. These changes in angle will need to be taken into account to get a good estimate. Methods exist to estimate the pose change between two views of the same plane in an unknown orientation. The transformation between two such views is a com-position of two plane-to-plane transforms (called *homographies*), which is again a plane-to-plane transform between the two image sensors. The homography between the two views can be estimated from a set of at least four point correspondences. The rotation matrix and translation vector are then obtained through a process called *homography decomposition*. Several methods have been proposed. Faugeras and Lustman [31] first described a method based on singular value decomposition (SVD) yielding multiple solutions, followed by the evaluation of additional physical constraints to find the true decomposition. Zhang and Hanson [108] improved this method further by deriving closed form expressions, still based on SVD decomposition, and still producing multiple solutions from which physically impossible configurations must be eliminated. Malis and Vargas [64] proposed an analytical decomposition which does not rely on SVD, but still yields four solutions. Any of these three methods is adequate for finding the appropriate decomposition in $\mathbf{R}$ and $\mathbf{t}$, however all are sensitive to noise (e.g. from poor intrinsic calibration, a problem which will be analyzed in Chapter 3, or from the use of few, noisy control points) and $\mathbf{R}$ may be poorly conditioned as a result. This problem is exacerbated in a road context, where the reference plane (the road) is not a perfect plane. Roads are often constructed with a crown, meaning that the centerline is situated higher than the edges in order to facilitate water draining towards the rain gutters. Homography decomposition is therefore a poor choice for visual odometry, and is outperformed by more constrained algorithms like the ones of Tardif et al. [94] and Scaramuzza [88].

Secondly, short of using epipolar geometry, there is no straightforward way to determine which areas in the image correspond to the ground plane and which do not. Road segmentation in images is not a simple problem, and falls outside the scope of this thesis. Our ground plane tracking method will therefore need to be robust to the presence of features originating from above (or under) the ground plane.

Clearly solving visual odometry problem by through ground plane tracking may avoid the issues inherent to epipolar geometry, but it also brings challenges of its own, each of which will be addressed in the following sections.

2.3.1 Feature extraction

A first choice which needs to be made is whether to use feature based tracking, or direct tracking employing image distance metrics like sum of absolute differences (SAD) as used in Song and Chandraker [91], or mutual information (MI) as used in Douterloigne et al. [24].

Direct tracking using the image data has the benefit that a feature matching step is not required. Feature matching is imperfect and gives rise to outliers; avoiding it means there is no need for computationally expensive coping strategies (e.g.

RANSAC). In theory, direct registration should also provide higher accuracy as it can be regarded as equivalent to feature registration with maximum feature density (i.e. all image data in the registered patch). In practice, we found direct registration to be problematic for the following reasons:

- imaged ground plane regions may be quite noisy and subject to motion blur, reducing accuracy,

- the road surface is generally quite homogeneous, which mostly negates the accuracy advantage due to low signal to noise ratio,

- the image distance function is not monotonic over large search spaces; many local minima occur,

- the dimensionality of the search space is large; suspension motion of the vehicle induces extra dimensions which slows down computation, exacerbates the noise sensitivity and further increases the chance of converging onto a local minimum.

Direct plane tracking will be briefly revisited in Section 2.5.3 with examples of these problems.were

Instead, we will detect and track features in the ground plane. Features are points in the image which stand out from their surroundings in some way; points which have a characteristic appearance which makes them easy to find in consecutive video frames. Many algorithms exist to locate such points. In this work we chose to use the Harris corner detector (Harris and Stephens [42]). The justification for this choice will be given in Section 2.3.1.1.

Ideally, we would only track features which we know to originate from the world ground plane. Our monocular camera provides us with no depth information however, and we do not want to resort to epipolar geometry to create it. We therefore have no straightforward way to determine whether features lie in the ground plane or not. To mitigate this problem, we will only detect features in a region of interest (ROI) of the image in which we normally expect to see the ground plane. For a front or rear facing camera in a road vehicle, looking in the driving direction, the ground plane usually occupies the bottom third of the image. It is also wise to limit the width of the ROI; otherwise in bends the features will often lie on roadside objects. Typically, we will set the ROI to a rectangle spanning 15 meters in front of the vehicle and three meters on either side on the real world plane. The corresponding region in the camera image can easily be calculated with the techniques which will be explained in the next section. An example of the ROI is shown in Figure 2.3. Since we do not want to detect features on the vehicle itself, the hood or any other visible parts of the car must be cut out of the ROI as well.

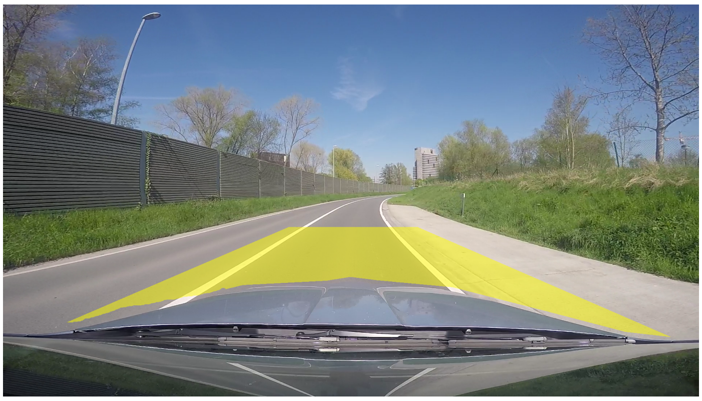


Figure 2.3: *Example of the ROI for feature detection for front facing camera. ROI is highlighted in yellow.*

2.3.1.1 Feature detector comparison

Many different feature extraction methods exist. The main concept underlying all of them is *salience*: which points in the image are remarkable enough that they will be easily recognized in another image of the same scene taken in different circumstances? Such points have the property that they stand out from their surroundings; in their local neighborhoods they maximize or minimize some easily observed property. The simplest example is a point which is higher in intensity than all its neighbors, and indeed some of the most popular feature detection algorithms are based on variations of that criterion.

Several articles have been devoted to comparing feature detectors (and sometimes their associated descriptors) for a variety of tasks, including object classification, clustering of images and tracking (Chao et al. [15], Khalifa et al. [52], Lankinen et al. [54]). Which properties of a feature detector are desirable, depends on the application. For example, if objects need to be classified at varying distances to the camera, scale invariance is required; preferably the same set of feature points would be detected on the object regardless of its apparent size.

The context of visual odometry, and specifically ground plane tracking, comes with specific challenges. While in some case there is clear structure in the road surface (e.g. when driving on pavement), often the surface will have a homogeneous texture, possibly with low contrast. Feature detectors which are better suited to detecting small scale features will perform better as they will pick up more debris or imperfections on a smooth tarmac road. Rotation and scale invariance are only somewhat useful; typically the difference in scale and orientation of an object between consecutive video frames will be small. Another important property is resilience against motion blur. In low light conditions and at high speeds, observations of the part of the road close to the vehicle will be washed out due to their rapid relative motion and the exposure time of the camera.

It is clear that the specific context of our visual odometry method is very different from the conditions in which the relative performance of feature detection algorithms is evaluated in literature. We will therefore conduct our own comparison of a small selection of feature detectors we think may suit our application. The evaluated methods are briefly discussed below, after which the comparison experiment is described.

Harris corners The algorithm by Harris and Stephens [42] is perhaps the most well known and certainly one of the most used feature detectors. It compares patches of the image to shifted versions of themselves to find points which are well localized. If a patch differs strongly from all its shifted versions, it is well localized (e.g. a point on the corner of a structure). If there are directions in which the patch can be shifted without causing a strong difference (e.g. points on a straight edge), the point is not well localized. Partial derivatives of the image intensity function in x- and y-coordinates are used to calculate the weighted sum of squared differences between the patch and its shifted versions.

Laplacian/Difference of Gaussian A class of multi-scale corner detectors uses the Laplacian of Gaussian (LoG) operator, or its faster approximation the Difference of Gaussian (DoG) operator. These operators respond well to points in which the rate of change of the gradient direction is high (which is in fact one of the possible definitions of a corner). The original theory of using derivatives of Gaussian functions for multi-scale feature detection was by Lindeberg [57]. LoG/DoG corner detectors operate on multiple scales. This gives them the potential advantage of also finding stable points inside blobs larger than the Harris patch size. The DoG filter is used in the SIFT (Scale Invariant Feature Transform) algorithm (Lowe [61]), where additional measures are implemented to avoid poorly localized edge responses. It is Lowe's version of the DoG detector which will be evaluated in this test.

Hessian The Hessian feature detector is based on the same scale-space theory as the LoG/DoG methods, but uses the determinant of the Hessian matrix to find the scale-space extrema (Lindeberg [58]). This precludes the need for additional filtering to avoid detections on poorly localized elongated structures. The Hessian method is at the base of the popular SURF feature detector and descriptor algorithm (Bay et al. [9]), in which the Hessian matrix is approximated using box filters for computation speed. It is Bay's version of the Hessian method which will be evaluated in this test.

Features from Accelerated Segment Test FAST (Features from Accelerated Segment Test, Rosten and Drummond [85]) features are computed by comparing the image intensity of a point to the intensities of pixels on a circle around it. A pixel is classified as a corner if there is a set of contiguous pixels on the circle is either brighter or darker than the central pixel. FAST is similar in principle to the SUSAN detector (Smallest Univalue Segment Assimilating Nucleus, Smith and Brady [90]) but the computation is heavily optimized by computing an optimal tree of pixel evaluations which discards non-corners as soon as possible. FAST is generally used for its speed, but its claimed performance for small point features makes it a good candidate for our application.

Binary Robust Invariant Scalable Keypoints BRISK (Binary Robust Invariant Scalable Keypoints, Leutenegger et al. [56]) also detects features as scale-space maxima, but uses a scoring criterion borrowed from FAST as a measure of salience. Like the LoG, DoG and Hessian methods it should respond well to blobs of various sizes as well as speckles.

Test setup To choose the most suitable feature for our visual odometry application, the following experiment was performed. A test bench was created consisting of 50 on-road video frames captured with a GoPro Hero 4 camera, which we deem representative of the quality of compact camera you may expect a car manufacturer to use. The video frames feature a representative variety of road surfaces, with tarmac the most common, but also pavement and concrete. The test set also spans a variety of road contexts and their typical speeds: highway, rural and urban/suburban. Both clear and overcast skies are present in the set, to make sure the motion blur and image noise are realistic for this application. For each of the frames, a region of interest is selected as discussed in Section 2.3.1. The image and the ROI boundaries are then transformed in a way which corresponds to the camera moving one metre forward. This transform is easily obtained as a combination of the inverse perspective transform defined in Section 2.3.2, an simple translation, and the forward perspective transform. An example of a frame and ROI and their transformation is show in Figure 2.4. Bi-cubic interpolation is used to obtain the artificially transformed image.

Features are now detected inside the ROI of the original and transformed images and compared on two criteria: stability and accuracy. Feature stability is measured as the ratio of features in the original image which have a corresponding feature detection in the transformed image. When a feature is found within two pixels from an expected location, we consider it a correspondence. Ideally, all the same real world points would be obtained in the original and transformed images, but since the features are selected on image properties which are non-linearly transformed by the (artificial) motion of the camera, this is not necessarily

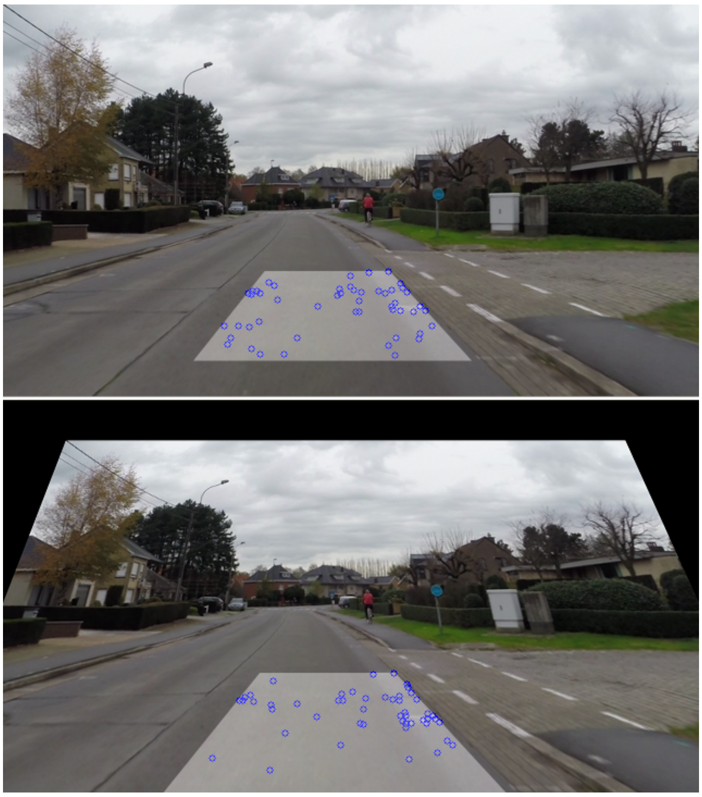


Figure 2.4: ROI for the feature comparison and its transformation corresponding to one metre displacement. Blue circles indicate the 50 highest ranked Harris corners.

the case. Many feature detectors also use non-maximum suppression of some kind, and since the artificial camera transform changes feature spacing, this may cause different features to be suppressed.

Feature accuracy is indicative of how much the re-detection in the transformed image differs from its theoretical location based on the known image transform of the original feature. A feature detector may for example not locate features to sub-pixel accuracy, and discretization noise will then cause a deviation in position. Feature accuracy is measured in average deviation measured in pixels over all feature correspondences.

A property of the feature detectors which is not considered in this test, is computational complexity. While computation speed is a factor in applicability, it is of secondary importance to the other performance criteria for visual odometry. None of the tested methods is prohibitively slow. A straight comparison of speed is also complicated by the fact that some methods (e.g. FAST) do not rank features in descending estimated salience, rather they simply output all points which passed a threshold in no particular order. These methods therefore require either iterative adjustment of the threshold, additional filtering to obtain the best fixed size subset of points, or both.

One criticism of this feature detector comparison is the use of artificial motion to generate image pairs. Ideally we would use a database of pairs of real images with exactly known camera pose between them. This is not straightforward however, even in cases where accurate ground truth for the camera pose change is available. One difficulty arises from the fact that the viewing angle of the ground plane must also be determined accurately, otherwise the expected locations for feature correspondences cannot be calculated. Additionally, imperfections in the road surface (i.e. local non-planarity) will cause deviation from the calculated positions even when feature detector performance is perfect. Finally the available datasets for which accurate IMU pose ground truth is available are generally limited to good light conditions and it is therefore more difficult to obtain a balanced test set. We believe the semi-artificial data used in our comparison is a good test of expected feature detector performance with a high degree of carry-over to real on-road performance.

For each algorithm, we either set the number of detected features to 50 or tuned the detection threshold to get as close to 50 as possible, but not less. For any remaining parameters we took the typical values indicated in the original publication of the feature detection method or the default values in the implementation provided by the authors (if available), or the default values of its OpenCV implementation. Optimizing the parameters of each feature detector for this particular application falls outside the scope of this work.

Results & Conclusion The results of the comparison are shown in Table 2.1. Judged on stability, Harris corners are clearly superior to all other methods. In terms of accuracy, Harris scores only mediocre, with DoG features scoring much better. However, the clear stability advantage means Harris is the preferred method for our visual odometry framework as it will yield fewer outliers compared to the other methods. The BRISK detector scores very poorly; it is simply incapable of finding stable features in low-contrast regions of tarmac. The inferior performance of the DoG, Hessian and BRISK detectors may also indicate that multi-scale approaches offer no benefits for our data, and the majority of image structure is situated at the base scale.

2.3.2 Inverse perspective projection

Now that we have detected salient points in the image which hopefully correspond to ground plane features, tracking these points allows us to estimate the motion of

detector	**stability**	**accuracy**
Harris	0.75	0.95
Hessian	0.66	1.14
DoG	0.61	0.38
FAST	0.58	0.87
BRISK	0.17	0.94

Table 2.1: *Relative feature detector performance comparison with respect to the planar tracking problem. The stability number is the ratio of successful redetections in the transformed image, the accuracy number is the average length of the difference vector between the theoretical and actual displacement vector over all successful redetections.*

the vehicle relative to the ground plane. It is advantageous to perform the tracking in ground plane coordinates rather than image coordinates for the following reason. Tracking relies on matching observed features to predicted locations of earlier observed features. The *proximity* of prediction and observation is better expressed in ground plane coordinates, where we can easily set a distance criterion which will count for all the points. In image coordinates, the formulation of a distance criterion is more complicated: real world points imaged one pixel apart may be centimeters or meters away from each other depending on their location in the image.

In order to track feature points in ground coordinates, we need to determine the transformation from image coordinates to ground plane coordinates. We call this transformation the *inverse perspective projection*. It is sometimes also referred to as *back-projection* in literature.

The transform from homogeneous image coordinates to homogeneous ground plane coordinates is not in itself complicated. With $\mathbf{x} = [x, y, 1]^T$ the image point and $\mathbf{X} = [X, Y, Z, 1]^T$ the world point, Equations 2.1 and 2.5 state that the projection is given by

$$w\mathbf{x} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \mathbf{C}[\mathbf{R}|\mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} . \qquad (2.9)$$

$\mathbf{C}$ is the intrinsic camera matrix containing focal lengths and principal point coordinates, which is fixed and measured through intrinsic calibration (more about this in Chapter 3). $[\mathbf{R}|\mathbf{t}]$ contains the rotation matrix $\mathbf{R}$ which aligns the world axes with the camera's viewing direction and sensor plane and the translation vector $\mathbf{t}$ which is the location of the origin of the world coordinate system in camera coordinates.

The matrix $[\mathbf{R}|\mathbf{t}]$ describes the full transformation of 3D world coordinates to 3D camera coordinates. In order to split this transformation into known (deter-

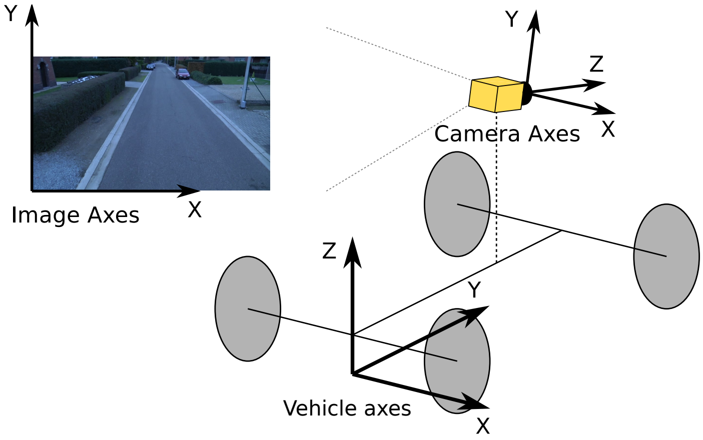


Figure 2.5: Illustration of camera, vehicle and image axes. 3D camera axes have the origin in center of projection of the camera. 3D vehicle axes have the origin in the ground plane below the center of the rear axle of the vehicle.

minable through calibration) and unknown parts, it is useful to consider it as a combination of two separate transformations:

$$[\mathbf{R}|\mathbf{t}] = \left[\begin{array}{c|c} \mathbf{R_{vc}} & \mathbf{t_{vc}} \end{array}\right] \left[\begin{array}{c|c} \mathbf{R_{wv}} & \mathbf{t_{wv}} \\ \hline \mathbf{0} & 1 \end{array}\right] = \mathbf{T_{vc}T_{wv}}.$$

The first applied transformation $\mathbf{T_{wv}}$, defined by $\mathbf{R_{wv}}$ and $\mathbf{t_{wv}}$ is from world to vehicle coordinates. The vehicle coordinate system is defined by the contact patches of the tires with the ground, with the X-axis parallel to the rear axle of the vehicle, the Y-axis equal to the forward driving direction and the Z-axis vertical. The origin of the vehicle coordinate system is chosen as the point on the ground directly below the midpoint of the rear axle. This choice of the vehicle coordinate system reduces $\mathbf{R_{wv}}$ to a single rotation around the vertical, and $\mathbf{t_{wv}}$ to a two-dimensional translation (with third element $z = 0$). Note that the estimation of visual odometry is equivalent to the estimation of this transformation; $\mathbf{T_{wv}}$ completely defines the pose of the vehicle relative to the world axes. Figure 2.5 illustrates our choice of coordinate systems. Note that the world coordinate system can be chosen freely. In many robotics applications, it is chosen equal to the vehicle axes at time step zero, while for navigation relative to a map, a system like Universal Transverse Mercator augmented with elevation may be used.

The second transformation $\mathbf{T_{vc}}$ describes the rotation $\mathbf{R_{vc}}$ and translation $\mathbf{t_{vc}}$ of the camera relative to the vehicle (e.g. is it looking in the driving direction or to the side, how far forward of the rear axle is it). This transformation can be determined through extrinsic calibration, which will be discussed in Chapter 3, although it is affected by the compression and extension of the vehicle suspension, which will be discussed later in this section. For now let us assume it is known and constant.

Under the assumption that all feature points originate from the ground plane, $Z = 0$ in Equation 2.9. Because $\mathbf{T_{wv}}$ is limited to in-plane rotation and translation by choice of the vehicle coordinate system, the vector $\mathbf{T_{wv}X}$ also has third coordinate equal to zero. We may simply omit this third coordinate and the third column of $\mathbf{T_{vc}}$, resulting in a 3×3 matrix $\mathbf{T'_{vc}}$. Multiplication with camera matrix $\mathbf{C}$ yields a 3×3 matrix $\mathbf{H} = \mathbf{CT'_{vc}}$ of rank 3 describing the transformation between homogeneous 2D coordinates relative to the vehicle and homogeneous 2D image coordinates. This plane-to-plane projective transformation H is a homography.

Our strategy to estimate the visual odometry is now as follows:

- project the 2D image coordinates of detected features into 2D ground plane coordinates relative to the vehicle using $\mathbf{H}^{-1}$,

- track the features in ground plane coordinates,

- calculate from the feature tracks the frame by frame rotations and translations which constitute $\mathbf{T_{wv}}$.

An important note concerns the uncertainty on $\mathbf{T_{vc}}$ due to the suspension motion of the vehicle. We may consider $\mathbf{T_{vc}}$ to be the product of two transforms $\mathbf{T_{v'c}T_{vv'}}$ where $\mathbf{T_{v'c}}$ is the calibrated transformation when the vehicle is stationary on level ground, and $\mathbf{T_{vv'}}$ describes the *attitude* of the vehicle, i.e. the rotation and translation caused by the compression and extension of the vehicle's suspension relative to this stationary pose. It is the transformation from "unsprung" vehicle coordinates (defined by the tires) to "sprung" vehicle coordinates. This transformation needs to be taken into account as it affects the viewing angle of the camera relative to the ground plane. It can be estimated directly using inertial sensors (accelerometers and gyroscopes). High-accuracy inertial sensors are expensive however, and while cheap solid-state sensors exist, they are difficult to calibrate and suffer from temperature-related drift, an undesirable property in automotive applications given the wide range of expected ambient temperatures [103]. In this work, we will not measure the attitude of the vehicle, but treat it as unknown within certain constraints. These constraints will result in a set of extremal transformations which delimit for each point in image coordinates a region of possible ground plane coordinates. These will be referred to as *observation uncertainty regions*.

The two main variables in the attitude of the vehicle are *pitch* and *roll*. Under braking or acceleration, inertia causes longitudinal weight transfer, which in turn causes a rotation around an axis parallel to the vehicle X axis (pitch). Left and right turns cause a lateral weight transfer, causing a rotation around an axis parallel to the vehicle Y axis (roll). The exact position of the rotation axes depends on the center of gravity of the vehicle, the relative spring rates front to back, and the geometry of the suspension design. As a general approximation, the point of rotation is assumed to be at the centroid of the quadrilateral formed by the top mounts of the shock absorbers. Other types of suspension loading, e.g. compression at all four corners, occur less frequently and their effects on the inverse perspective projection are much smaller (as will be shown in Chapter 3). We will therefore only consider pitch and roll as contributing factors to $\mathbf{T_{vv'}}$. The suspension geometry of the vehicle limits these angles, and these limits define a set of possible transformations between image and vehicle ground plane coordinates.

A typical road vehicle experiences in the range of 100-150mm total suspension travel (i.e. the difference between full compression and full extension of the shock absorber) measured at each wheel. With a track width of 1.4-1.5m, this could theoretically give rise to approximately 10° of lateral roll. Considering a wheelbase of 2.7m 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. Figure 2.6 illustrates the impact of suspension motion on the inverse perspective projection.

The transformation $\mathbf{T_{vv'}}$ is not strictly a linear function of the pitch and roll angles, as its elements contain trigonometric functions. For the small range of possible pitch and roll angles however, we may assume it to be approximately linear. This means that the region of possible ground plane coordinates for a single feature point is well approximated by the quadrilateral spanned by its four extremal back-projections, i.e. the inverse perspective projection of the feature point evaluated for the four combinations of minimum and maximum pitch and roll. Examples of such regions are shown in Figure 2.7. 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. This supports our decision to vertically limit the ROI for feature detection so distant features are excluded.

An important remark with respect to inverse perspective projection is that the transform is calculated for features with $Z = 0$ and will not be accurate for features not originating from the ground plane. However, there is no easy way to discern whether a feature in the camera view lies on the ground plane or not. We therefore have no choice but to apply the back-projection to any features we detect in the camera image, and sort out the above-ground features in a higher level reasoning step.

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

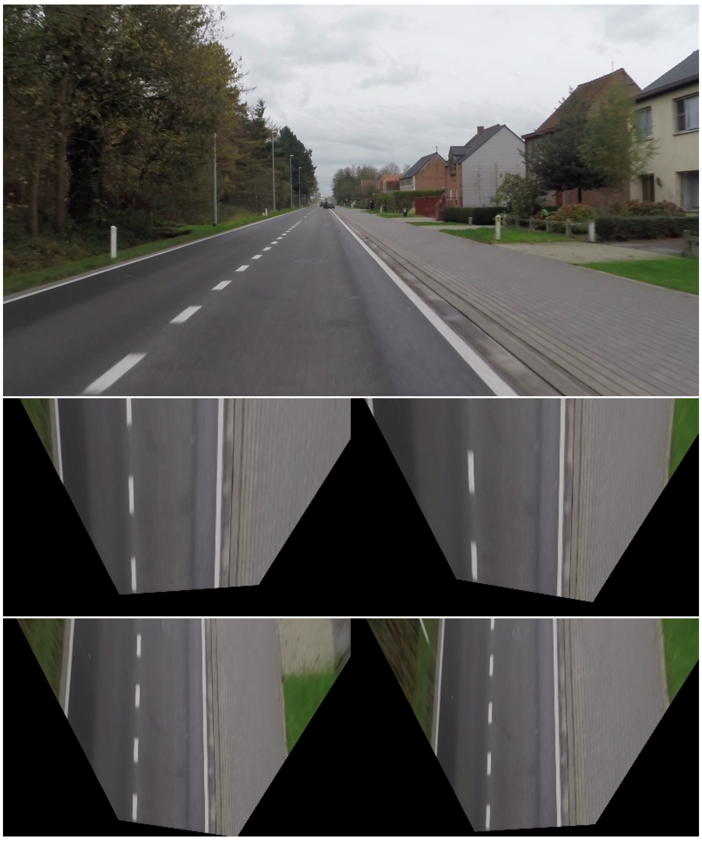


__Figure 2.6:__ Effect of suspension motion on inverse perspective projection. For the original perspective image (top), the four extremal back-projections corresponding to maximum and minimum pitch and roll are shown. The differences between the back-projected images illustrate the uncertainty of feature locations on the ground plane on account of the unknown suspension angles.

approximation, the distortion must either be small, or corrected in pre-processing. More details about distortion correction will be given in Chapter 3.

2.3.3 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, based on the kinematic model of a road car. In robotics, a car-like platform with fixed rear wheels and steerable front axle is called a *nonholonomic robot*. Formally, a non-

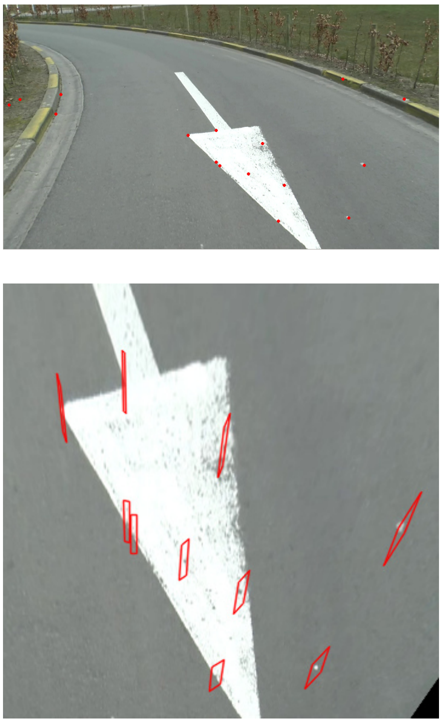


Figure 2.7: *Example of observation uncertainty regions (red quadrilaterals, bottom) for some detected Harris corners in the camera image (red dots, top). The background image in the bottom picture was obtained by setting $\mathbf{T}_{\mathbf{v}\mathbf{v}'}$ to the identity matrix. It represents the static back-projection in the absence of pitch and roll.*

holonomic system is a system in which the state depends on the path leading up to the state. It can easily be seen how the orientation of the vehicle at any time is dependent on the path it took to its current position. The kinematic motion model of a non-holonomic robot is given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \tag{2.10}$$

where x, y denote world position, θ is the heading angle of the vehicle, v is the longitudinal velocity and ω the rotational velocity. For more details concerning the theory of non-holonomic kinematics and control, we refer to Laumond [55]. In practice, road cars are designed to satisfy the Ackermann principle, which states that the roll axles of all wheels must intersect in one point (see Figure 2.8). As such, the vehicle behaves as if the entire front axle were steerable. The non-holonomic kinematic model applies, and the vehicle will describe at any moment

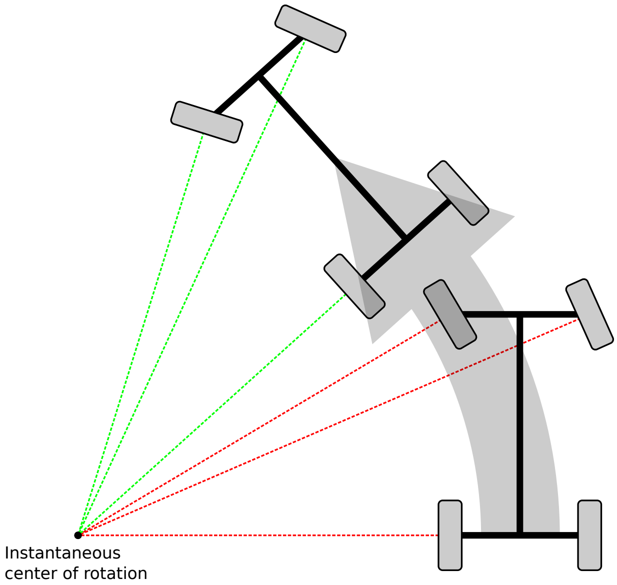


Figure 2.8: *Illustration of the Ackermann steering principle. The vehicle will describe a circular arc around the intersection of the roll axles of the wheels.*

a circular trajectory around the intersection point of the roll axles. This point is called the *instantaneous center of rotation* (ICR) and its location relative to the vehicle depends on the control parameters v and ω.

Consider a feature with known ground plane coordinates in the previous frame. When the parameters v and ω are known, we can use the kinematic model to predict the new coordinates of the feature. Consider the planar vehicle coordinate system, with the Y axis in the direction of travel of the vehicle, the X axis directly under the rear axle of the vehicle and the origin under the center of the rear axle. Over a time interval Δt the vehicle will move along an arc with length $v\Delta t$ and increase its heading by $\omega\Delta t$. The radius of the arc is given by $r = \frac{v\Delta t}{\omega\Delta t} = \frac{v}{\omega}$ and the ICR therefore has coordinates $(\frac{v}{\omega}, 0)$. In the vehicle coordinate system the features will seem to describe arcs around the ICR in the opposite direction. The predicted location of a feature point (x, y) in these axes is therefore given by

$$x' = \cos(-\omega\Delta t)(x - \frac{v}{\omega}) - \sin(-\omega\Delta t)y + \frac{v}{\omega},$$

$$y' = \sin(-\omega\Delta t)(x - \frac{v}{\omega}) + \cos(-\omega\Delta t)y.$$

We can evaluate this for all features to predict their most likely position using the last estimated velocities. The above reasoning has assumed that v and ω remain constant over the time interval Δt. In practice, although the steering angle and velocity of the vehicle are obviously not constant, they cannot change abruptly over a short 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 time step, only a small range of angle-velocity combinations is plausible at the current time step. By applying the kinematic model to this region 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 2.9.

Note that the kinematic model in Equation 2.10 does not take side-slip 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. For normal road driving however, side-slip angles are typically small and the model is a good approximation [21].

Care should be taken when applying this motion model for larger time steps (e.g. one second or more). When the angular velocity changes significantly over the time interval, the motion of the vehicle will not be accurately described by a circle arc and the predicted location and orientation may be inaccurate.

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 22km of GPS/INS ground truth data. The histogram of the angular acceleration is shown in Figure 2.10. From this histogram, we observed that 97% of occurring values are between $\pm 20 \deg/s^2$. 92% of values are between $\pm 10 \deg/s^2$. As a trade-off between search region size and odometry accuracy in rapid maneuvers, we will typically choose a limit of $\pm 10 \deg/s^2$.

The theoretical maximum change in vehicle speed corresponds to an emergency stop and is around $10m/s^2$. Again though, typical values during normal driving are much less extreme. Maurya and Bokare [65] measured maximum deceleration for cars in hard braking from motorway speeds to be $1.71m/s^2$. For trucks, this value is reduced to $0.88m/s^2$. 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 [59], however it is not uncommon for drivers to apply full acceleration even in normal driving situations. Based on comfort and safety recommendations for public transport [75], we assume acceleration in normal circumstances to be under $1.5m/s^2$ as well.

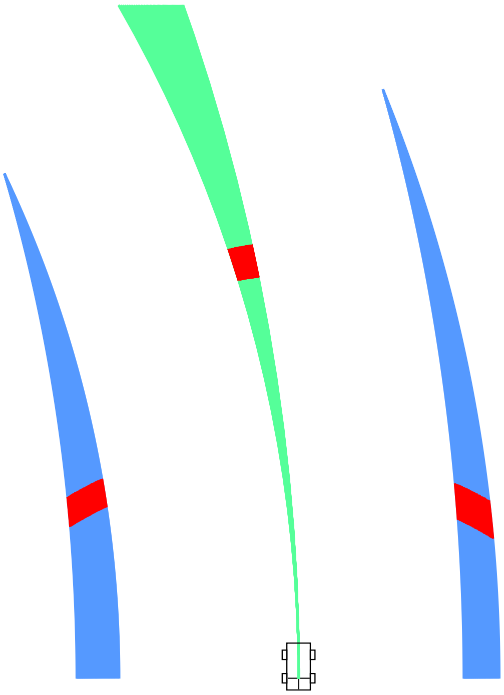


Figure 2.9: *Prediction uncertainty regions for two features for a range of possible vehicle positions. When the vehicle drives forward with a certain initial velocity and steering angle, limited rate of change of these two parameters means the vehicle can only end up in the dark zone (red) in the central 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).*

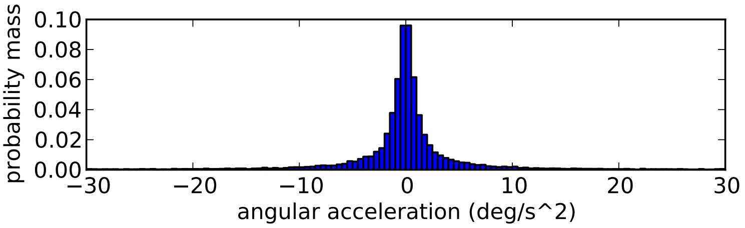


Figure 2.10: *Histogram of angular acceleration of a vehicle during combined urban/suburban/highway driving.*

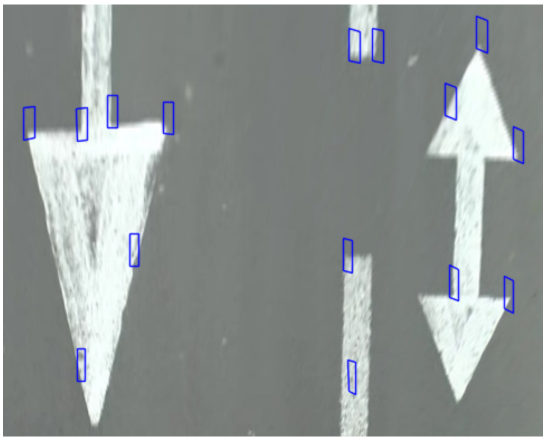


Figure 2.11: *Example of prediction uncertainty regions for current frame based on previously tracked features. The background image was again obtained by applying the inverse perspective transform in the absence of pitch and roll to the entire camera image.*

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 2.11. 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 2.3.2. 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 show in Figure 2.12. 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).

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 detected on this moving object will, in general, not have observa-

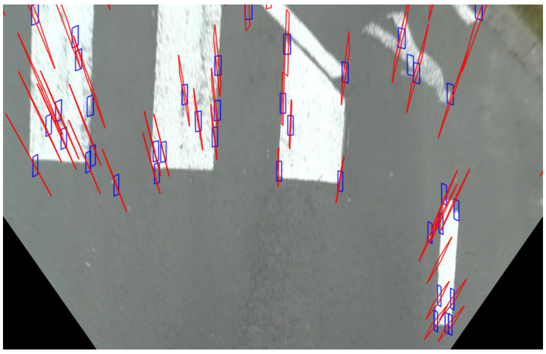


Figure 2.12: Example of overlap between prediction uncertainty regions (blue) and observation uncertainty regions (red).

tion uncertainty regions that consistently overlap with prediction uncertainty regions, because the relative motion of the object does not comply to the constraints imposed by the kinematic model. This is a significant advantage compared to an appearance based matcher, which will tend to match a large number 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.

2.3.4 Odometry calculation

From the feature matches obtained as described in Section 2.3.3, we now need to remove any remaining outliers and calculate the odometry. The current pose of the vehicle (i.e. its heading and position) is calculated by estimating for each inter-frame interval the two parameters that characterize the motion of a vehicle according to the kinematic model: rotational and longitudinal velocity. The rotational velocity ω is defined as the difference in heading between two consecutive observations divided by the time step. The longitudinal velocity v 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 rotational velocity for the same longitudinal velocity means that the ICR is closer to the vehicle. To recap the kinematics explained above, the relation between the three parameters is given by

$$v = \omega r \tag{2.11}$$

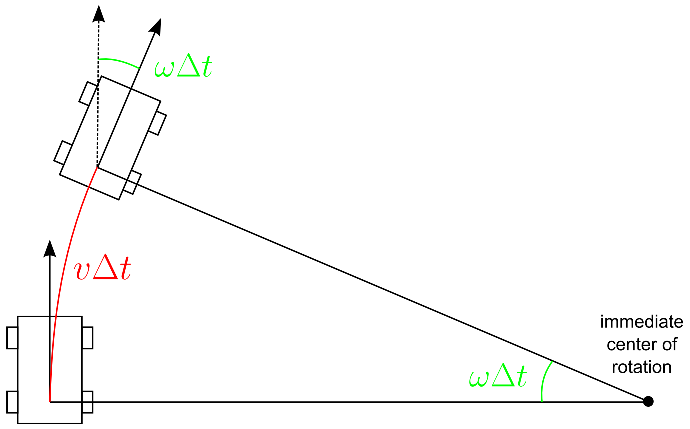


Figure 2.13: *Illustration of the motion parameters that need to be recovered. Longitudinal velocity v (red) is measured from the length of the circular arc, rotational velocity ω (green) from the difference in heading between two consecutive states.*

with r the radius of the circle arc, i.e. the distance of the ICR. The motion parameters and their relation are illustrated in Figure 2.13.

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 Section 2.5.2.

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. In such a scheme, the minimum number of features required to calculate a motion hypothesis is chosen randomly from the detected feature set, and the hypothesis is tested for consistency with the remainder of the set. This is repeated until either a sufficiently supported hypothesis is found, or until a predefined the number of repetitions is reached, after which the best supported hypothesis thus far is chosen as the solution. RANSAC allows to find a consensus among heavily polluted data with a high fraction of outliers. In our case, RANSAC offers few advantages, 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.

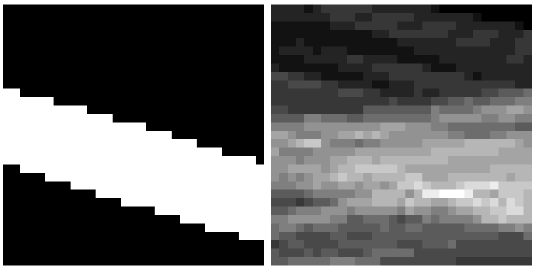


Figure 2.14: *Example of square image representing search region and its overlapped part (left), and sum of many such square images (right).*

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 2.11). The edges of these predicted regions correspond to the limits of change the driver can effect 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 2.14. 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 limited number of features coupled with the binning in the sum image 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 2.3.3.

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 back-projected 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.

A sufficient condition for accurate multi-frame estimation is that for both of the motion parameters the sum of the maximum single frame estimation error and the maximum inter-frame change of this parameter is smaller than the prediction uncertainty for that parameter. When this condition is fulfilled, the next frame's feature consensus will still fall within the predicted uncertainty region and the errors will compensate.

To illustrate this point, consider the simplified example of overestimating the velocity at time t as $0.45m/frame$ while the real velocity is just $0.4m/frame$. This overestimation of the velocity is equivalent to a mis-estimation of the actual feature positions from the fuzzy data by $0.5m$. The prediction for time $t+1$ will assume a constant velocity of $0.45m/frame$ and use the mis-estimated 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 $0.10m$ to the actual feature coordinates. When the actual velocity of the vehicle at time $t+1$ is again $0.4m/frame$, the observation uncertainty regions will then each be centered on a pixel corresponding to $0.10m$ above the center of a prediction uncertainty region. The method will then correct the estimate for the second state to $0.35m/frame$, and the average estimated velocity over two states will be accurate. The robustness to immediate errors afforded by the prediction-correction tracking is a big advantage for a road vehicle application, where camera and vehicle shake are to be expected. This safety mechanism will only mitigate single-frame estimation errors; in case of continuously poor feature matching, errors may still accumulate.

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 time step to be known. Due to the uncertain inverse perspective projection however, these coordinates can not 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 time step. 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 3 and 5) are discarded.

2.4 Results

The proposed method was evaluated on two datasets, and compared to the monocular 8-point method by Geiger *et al.* [33]. The 8-point implementation is provided on-line by the authors. In the KITTI odometry benchmark, several monocular methods currently outperform this standard 8-point method. They are listed below, each with their main concepts and disadvantages. Anonymous submissions are not included.

- MVO: 8-point method, more than 3000 features tracked (claimed 20 fps, no reference paper cited).
- W-SFM: 5-point method with bundle adjustment (10 fps on dual core, no reference paper cited).
- FTMVO: iterative 5-point method, ground plane tracking for scale, almost real-time (9 fps) [69].
- LCMVO: 8-point method combined with optical flow, over 2000 features tracked (claimed 10 fps, no reference paper cited).
- MLM-SFM: 5-point solver optimized by scene structure propagation and ground plane estimation (30 fps on 5 cores) [91].
- RMCPE+GP: 7-point method with ground plane estimation (2 fps) [68].
- VISO2-M+GP: 8-point method we will compare against, but with ground plane estimation (6 fps) [33, 91].

All methods employ either a 5-point, 7-point or 8-point method to perform initial 3D pose estimation. The aim of this research is to prove that our ground plane tracking approach is a viable alternative to traditional essential matrix estimation for visual odometry. We have therefore chosen the basic 8-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 8-point method and our proposed method are compared, is the KITTI odometry benchmark itself [4]. This dataset consists of 22 sequences captured in urban, suburban, rural and highway scenarios spanning approximately 35km. The camera offers a wide 1241x376 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 2.15. The dataset offers the extrinsic and intrinsic camera parameters needed by the 8-point solver. For the proposed method, the translation vector between camera and the center of the rear axle is also required to be known. This translation was approximated using the methods described in Chapter 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 12m in front of the vehicle, as well as lateral cutoffs at 3m left and right of the straight-ahead (to reduce the number of features detected on non-ground plane objects). The cutoffs are illustrated in Figure 2.15. Within the cutoffs, 32 Harris corners were detected on each side of the straight-ahead. Features were considered lost in case of no match for 5 consecutive frames.

The performance evaluation provided by the KITTI benchmark is based on two metrics: translation error and rotation error. Both are calculated relative to the traveled distance to give an indication of expected drift in position and heading for longer travel distances. The two metrics are calculated as follows. Let $\mathbf{P}_i$ denote the ground truth pose corresponding to frame i relative to the starting pose, i.e. the 4×4 matrix describing the rotation and translation that needs to be applied on the initial vehicle coordinate system at the beginning of the sequence to transform it into the vehicle coordinate system at time i. The ground truth pose change $\mathbf{Q}$ between states i and j is given by

$$\mathbf{Q} = \mathbf{P}_i^{-1}\mathbf{P}_j. \tag{2.12}$$

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

$$\mathbf{Q}' = \mathbf{P}'^{-1}_i\mathbf{P}'_j. \tag{2.13}$$

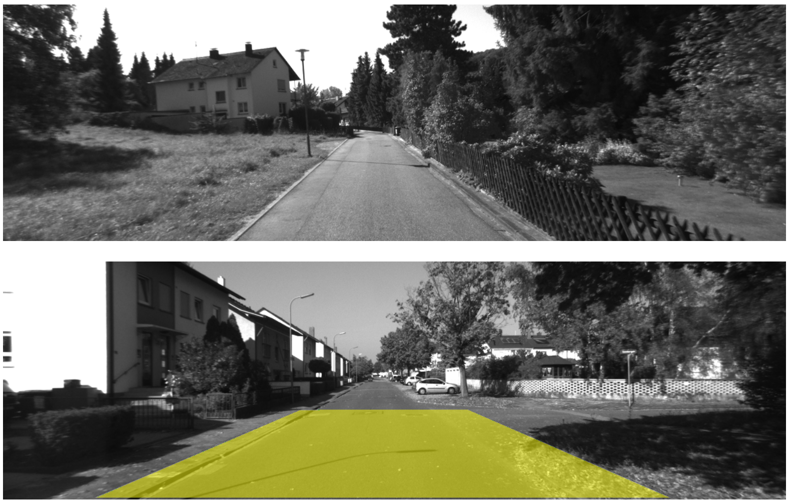


Figure 2.15: *Sample frames out of KITTI dataset [4]. Note the exceptionally wide field of view. Bottom image shows cutoffs for feature detection overlaid on the perspective image. Features will only be detected in the highlighted zone.*

The pose estimation error matrix is

$$\mathbf{E} = \mathbf{Q}'^{-1}\mathbf{Q}. \tag{2.14}$$

This 4x4 matrix describes the residual transform, i.e. the additional pose change which must be applied to the estimated transformation $\mathbf{Q}'$ to obtain the true transformation $\mathbf{Q}$. In this 4x4 matrix, the fourth column contains the residual translation and the translation error is therefore given by

$$\Delta_t = \sqrt{e_{14}^2 + e_{24}^2 + e_{34}^2} \tag{2.15}$$

where e_{uv} denote the elements of $\mathbf{E}$. The rotation error, i.e. the angle between any vector transformed according to $\mathbf{Q}$ and the same vector transformed according to $\mathbf{Q}'$, can be calculated from the trace of the residual transform:

$$\Delta_r = \mathrm{acos}(0.5 * (\mathrm{trace}(\mathbf{E}) - 1)).$$

The translation and rotation errors are calculated on all possible subsegments of the ground truth trajectory of length 100, 200 ... 800 meters (using a sliding window over all ground truth poses). The errors are averaged per segment length. Translation error is expressed as a percentage of segment length, while rotation error is expressed in $^\circ/m$.

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. For the proposed method, the planar poses are expanded to 3D poses with fixed (zero) elevation and zero out-of-plane rotations. 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 2.16. 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 8-point solver performs worse for both low (under 30 km/h) and high (above 60 km/h) vehicle speeds. The poor low speed accuracy can be explained by the requirement of significant translation for epipolar geometry (as explained in Section 2.2), while the high speed errors are possibly due to the increased difficulty of the data association between consecutive frames.

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 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 200m. We may conclude that on the KITTI dataset the proposed method is significantly more accurate than the 8-point solver, with a 24% reduced translation error and 8% reduced rotation error. Some examples of estimated trajectories are shown in Figure 2.17.

Both methods are able to process the data faster than real-time on a desktop computer (Intel Core i5 3.40GHz x4), with the proposed method significantly outperforming the 8-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 4 cores), while the remainder of the processing is single threaded. The method of Geiger *et al.* runs completely single threaded. The high processing speed is clearly an important advantage in the context of intelligent vehicles, and the speed of our method is at least a factor 2.5 faster than the fastest monocular method in the KITTI benchmark despite being only partly parallelized.

The second evaluation dataset consists of 15km 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 2.18. The camera captures video at 50 frames per second and was calibrated intrinsically using a checcurboard pattern and the method of Bouguet [13]. The extrinsic calibration was estimated iteratively as explained in Section 3.

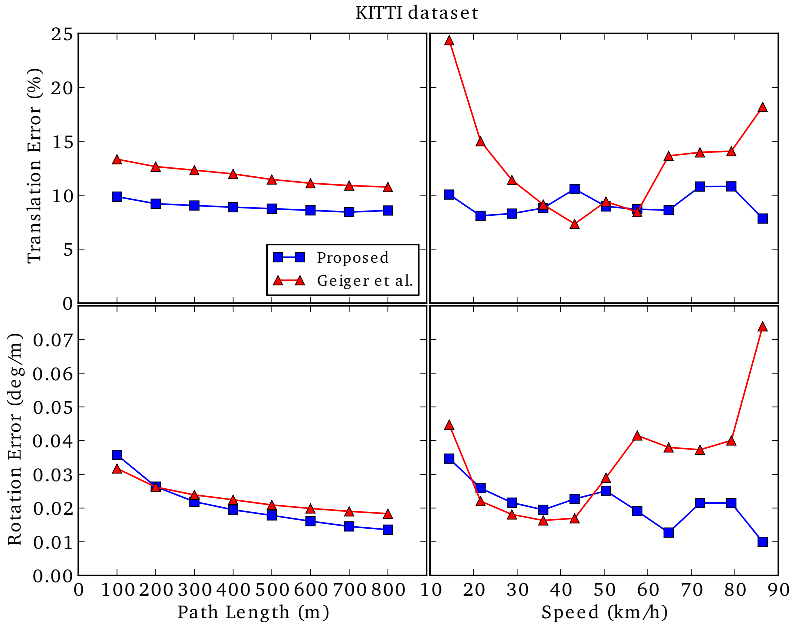


Figure 2.16: *Accuracy evaluated on KITTI dataset. Translation errors are shown in top row, rotation errors in bottom row, both in function of segment length (left column) and speed (right column).*

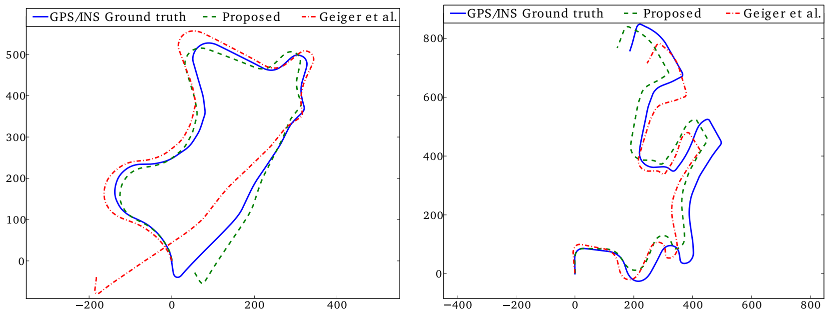


Figure 2.17: *Examples of estimated KITTI trajectory according to proposed method and Geiger et al. compared to ground truth. Axes are in meters. Blue line represents ground truth, green dashed line the proposed method, red dot-dashed line the 8-point solver.*

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. The relative resolutions of the ground plane are shown in Figure 2.19.

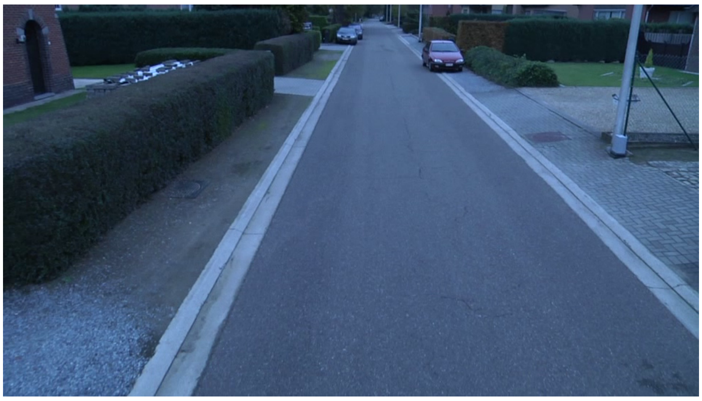


Figure 2.18: *Sample frame of Diepenbeek/Hasselt dataset.*

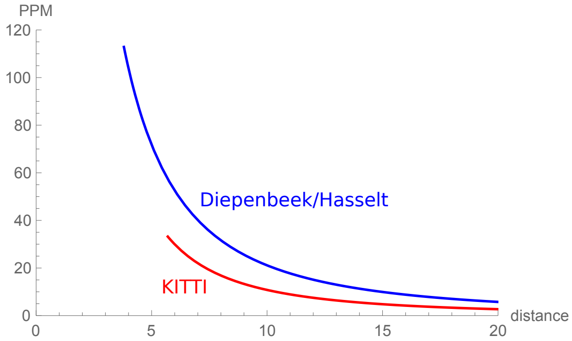


Figure 2.19: *Comparison of ground plane resolution between KITTI and Diepenbeek/Hasselt datasets, expressed in pixels per meter (PPM) on the line straight ahead from the vehicle. The starting point of the curve indicates the nearest distance which is visible at the bottom of the video frame.*

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 wide-screen 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 seen at greater angles from the viewing direction have larger displacements), and therefore the triangulation accuracy is also expected to be slightly lower.

While the camera captured video at 50 frames per second, we discarded 4 out of every 5 frames to attain the same 10 fps frame rate as in the KITTI sequences in order to provide as fair a comparison as possible; the parameters of the 8-point solver are optimized for 10 fps and higher frame rates would exacerbate

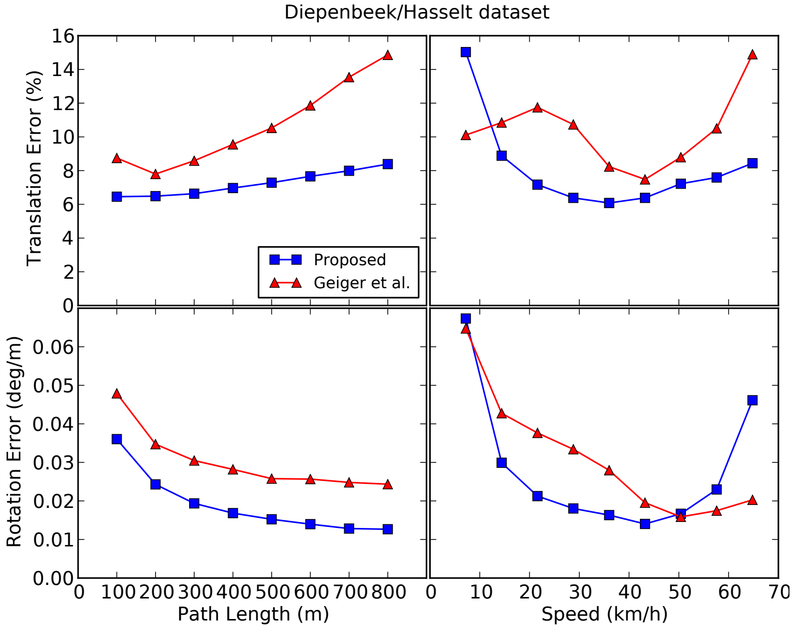


Figure 2.20: *Accuracy evaluated on Diepenbeek/Hasselt dataset. Translation errors are shown in top row, rotation errors in bottom row, both in function of segment length (left column) and speed (right column).*

the errors in case of small inter-frame translations. 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 meters 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 2.20. 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).

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

Some examples of reconstructed trajectories are shown in Figure 2.21. A summary of translation and rotation accuracy on both sets is shown in Table 2.2.

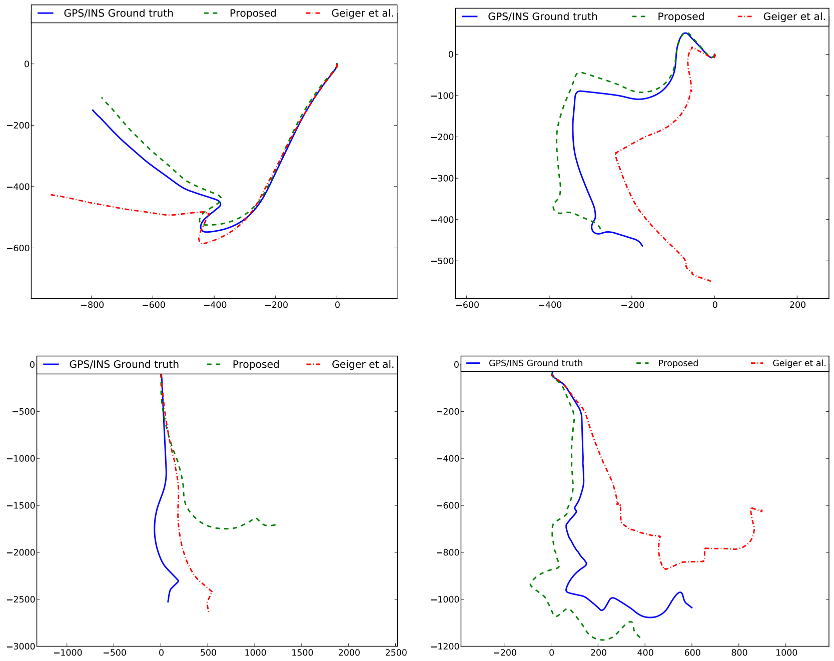


Figure 2.21: *Examples of estimated trajectories on the Diepenbeek/Hasselt dataset. Axes are in meters. Blue line represents ground truth, green dashed line the proposed method, red dot-dashed line the 8-point solver.*

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

Table 2.2: *Summary of mean errors of both methods on both datasets.*

2.5 Analysis

In this section, an analysis of relative strengths and weaknesses of the method is made and the contributing factors to the odometry errors are explored. Additionally, the concept of direct tracking is briefly revisited as a possibility for further improvement, and a qualitative comparison is made to a more standard ground plane feature tracking implementation.

2.5.1 Strengths and weaknesses

The results obtained on both datasets clearly illustrate the main advantage of the proposed method over the 8-point solver, namely better estimation of translation. In several of the sequences, the 8-point solver significantly mis-estimates the length of one or more straight segments (e.g. the final section in the left plot of Figure 2.17). 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.* [53], 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. The proposed method does not suffer from scale drift because it does not need to normalize the computed poses; the ground plane position is directly used as an assumption in the computations instead of computed from the observed points, and one source of scale errors is therefore eliminated.

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 8-point solver can produce an erratic motion estimate, the proposed method assumes steady-state as a fall-back mechanism. This is illustrated in Figure 2.22.

For the proposed method, we observed that meaningful odometry was produced for inlier ratios as low as 1:8, 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, so this 1:8 threshold on inlier ratio was added to the implementation. The results in this section were achieved without this threshold, however the difference is minimal as the inlier ratio is generally much higher.

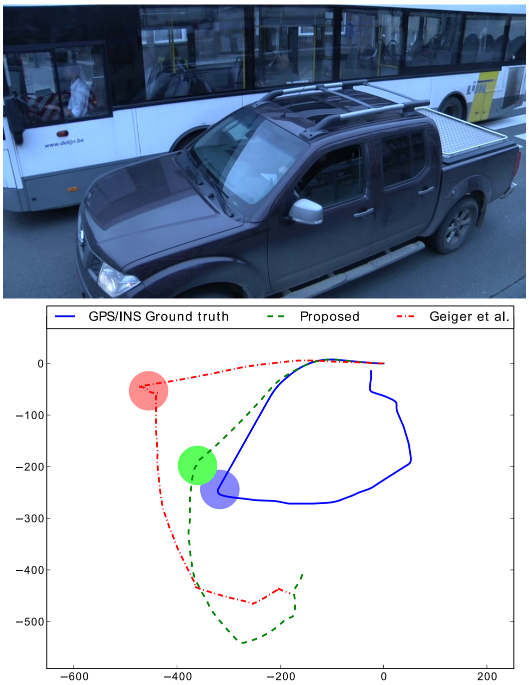


Figure 2.22: *Left turn in dense traffic. Camera image (top) is almost entirely composed of moving vehicles. Trajectory estimation (bottom) of both methods suffers; Geiger et al. produces erratic motion while proposed method assumes steady state during ambiguous period.*

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 example can be seen in the bottom left plot of Figure 2.21. 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 Section 2.5.2. The method of Geiger *et al.* does not suffer from this flaw. It is therefore to be expected that on long, straight roads, the 8-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 observation is that the immediate steering angle and velocity estimates of the proposed method are quite noisy, while their running average tracks the ground truth closely. As an example, the steering angle and velocity estimates for the top right trajectory in Figure 2.21 are shown in Figure 2.23. The power spectra of the errors (plotted in Figure 2.24) resemble that of white noise, with slightly elevated low-frequency components. These graphs indicate that although there are many unpredictable sources of immediate errors (presumably camera and vehicle shake as well as discretization noise), the tracking over several frames is robust and accurate. This is an inherent advantage of the prediction-correction process as explained in Section 2.3.4.

The magnitude of the steering angle and velocity errors is somewhat correlated with the magnitude of the corresponding ground truth values, with Pearson correlation coefficients of 0.23 for steering and 0.25 for velocity. The correlation with steering angle is explained by the fact that the deviation of camera perspective will tend to be larger during cornering. The distribution of feature points and the particular shape of the road surface will decide the direction of the estimation error resulting from the perspective deviation. The correlation with velocity is likely a result of the reduced timespan during which a single feature can be tracked.

Overall, we can observe that the proposed method is often better than the 8-point method at recovering macro-maneuvers present in the trajectory: at intersections and roundabouts, the 8-point solver sometimes fails to accurately estimate the large changes in heading. This may be related to the motion degeneracy problem explained in Section 2.2; in low-speed high-curvature trajectories the depth estimation is generally less accurate. An example can be seen in the top right image of Figure 2.21: the method of Geiger *et al.* misses most of the roundabout. The ability to correctly estimate big maneuvers is especially important for the integration with off-line map data, as they provide more reliable clues for map matching

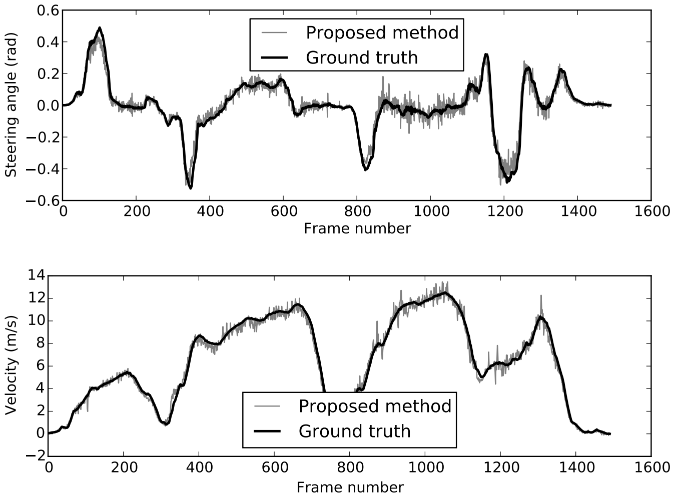


Figure 2.23: *Oscillation of estimated steering angle and velocity (gray) around ground truth values (black), illustrating self-correcting nature of the estimation method.*

than gentle curves and straights. The concept of map matching as a mechanism to eliminate error accumulation will be implemented in Chapter 4.

2.5.2 Degeneracy

In our method, 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 curb. 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 performed a simulation in which artificial video is generated for a vehicle moving along an S-shaped point grid. The trajectory consists of two 30m 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 $5^\circ/s$. The virtual camera was set in a similar configuration to the camera in the KITTI and Diepenbeek datasets, with zero roll and heading and -20° pitch. The simulation trajectory and an artificial video frame are shown in Figure 2.25.

This simulation allows us to easily control the distribution of the points and analyze its effect on the global trajectory reconstruction by the proposed method,

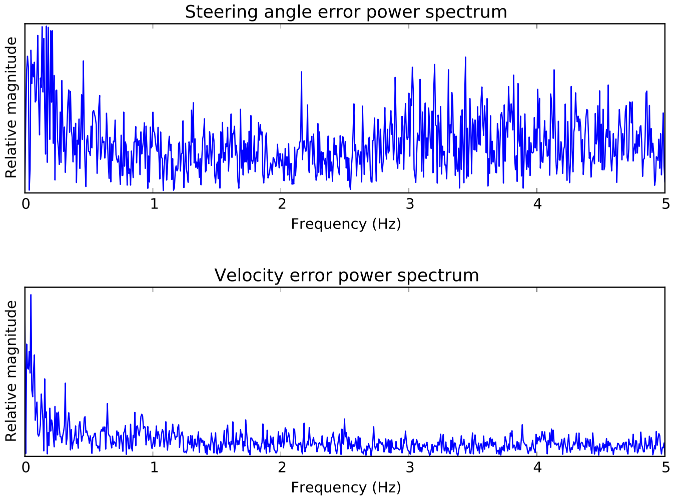


Figure 2.24: *Power spectra of the steering angle and velocity errors, showing nearly uniform distribution across the frequency range.*

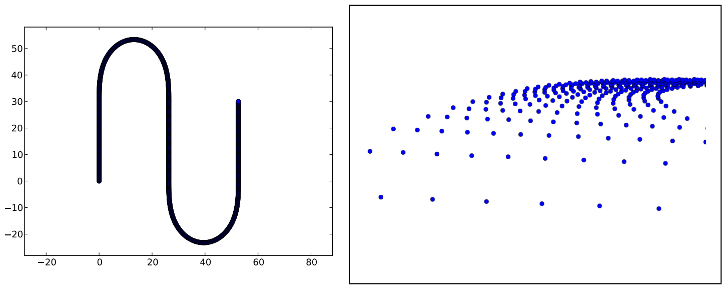


Figure 2.25: *Simulation trajectory composed of two straights and two clothoid turns (left), and sample artificial video frame from start of bend (right).*

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 and perfectly planar, uniformly distributed features. In this perfect configuration, the reconstructed trajectory was accurate within 0.5% and $0.006°/m$ (these small errors are a result of the discretization of point locations to integer pixel coordinates). We then adapted the point grid so that points are elevated on one side of the trajectory. Specifically, points in the zone from 2m to 3m on the right side were raised by 15cm, a typical curb 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 2.26. The resulting odometry errors are shown

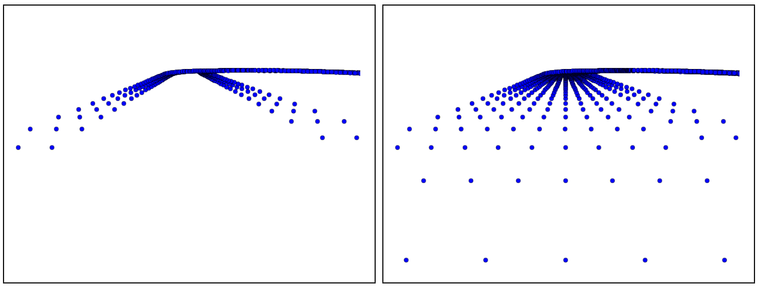


Figure 2.26: *Artificial video frames for the curb simulations without (left) and with (right) center points.*

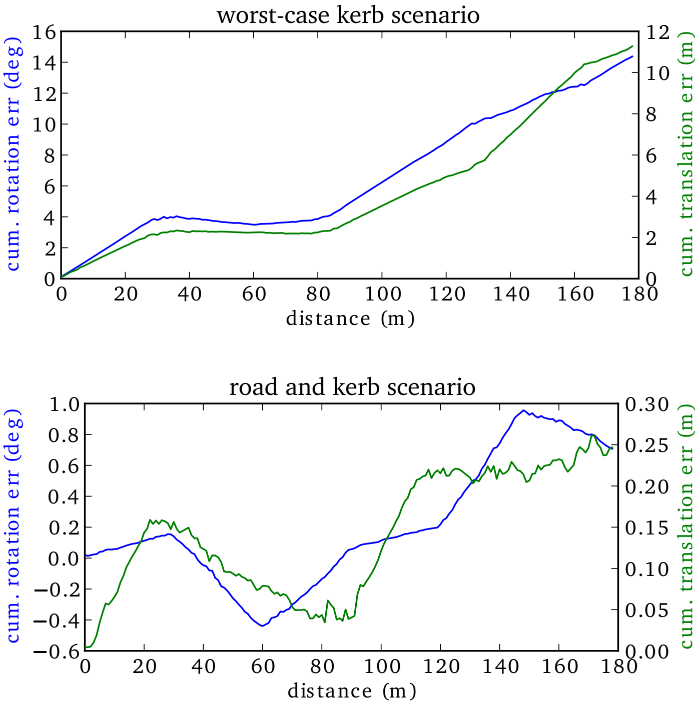


Figure 2.27: *Cumulative errors for curb simulations in worst case scenario without center section points (top) and typical scenario with center section points (bottom).*

in Figure 2.27 (top). The effects in this worst case scenario are quite pronounced: a $0.078°/m$ rotation error and a 6.3% translation error. The height difference between the left and right side gives rise to increased observed translation speed on the high side, giving rise to a clear bias towards turning left.

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 2.26 (right), and the resulting errors in Figure 2.27 (bottom). The errors in this case are insignificant at only $0.004°/m$ for rotation and 0.1% for translation.

The second scenario is that of a road with a crown, i.e. a road of 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 crown-less 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, 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% [84]. The odometry errors were only evaluated on the straight sections, as super-elevation (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 curb 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 2.28. 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.

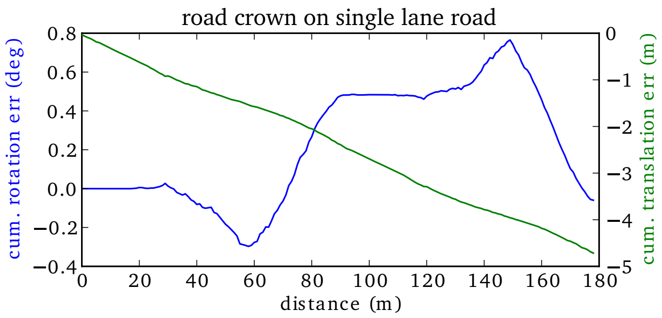


Figure 2.28: *Odometry error in case of single lane road with crown centrally across the lane.*

2.5.3 Direct tracking revisited

We have shown that our method, while robust and computationally efficient, is sometimes lacking in absolute accuracy, especially in the case of slightly polluted feature sets (either because they violate the planarity assumption slightly, or because of poor feature stability on homogeneous road surfaces). In these cases, *direct tracking* could in theory provide a better estimation. To recap, direct tracking refers to the use of the image intensities directly, without extracting features from it. To this end, an image distance function is minimized using an optimization algorithm. For example, Song and Chandraker [91] minimize the sum of absolute differences (SAD) using the Nelder-Mead simplex algorithm. In their work, this is employed to find the orientation and distance of the ground plane. An important consideration is that such optimizations can fail to find the global optimum for non-monotonous functions, converging to a local optimum instead. In SLAM applications, typically a rough estimate is first performed using a different method (e.g. essential matrix estimation in [27, 91, 99]) to ensure the optimization starts relatively close to the global optimum to mitigate this problem, as well as limit the amount of iterations needed to converge.

For our method, we may employ the same principles. Using our Hough-like voting algorithm to obtain a first estimate of rotation and translation, we will attempt to improve this estimate by direct tracking. There are several ways in which direct tracking can be applied. Firstly, we can use direct tracking to estimate the immediate roll and pitch angles. These refined angles can then be used to recompute the odometry using smaller observation uncertainty regions. Such an iterative approach would clearly come at the cost of computation speed however. Another possibility is to optimize the roll and pitch angles and odometry parameters simultaneously. In practice this approach showed poor performance; the four dimensional optimization often failed to converge to the correct optimum. The third and most promising option is to optimize only the odometry parameters themselves. Similar to Song and Chandraker [91], the Nelder-Mead simplex[70] will be used to optimize the SAD distance metric between the prediction and observation of an image patch. The optimization will be done on the perspective camera frames using the at-rest back-projection as a basis. The steps of the algorithm are outlined below:

1. calculate initial θ, v from uncertainty region overlap,
2. extract ground plane ROI from current video frame,
3. transform previous video frame according to $-\theta, -v$,
4. extract ground plane ROI from transformed previous video frame
5. calculate SAD between previous and current ground plane ROI,
6. optimize θ, v using Nelder-Mead simplex on steps 3-5.

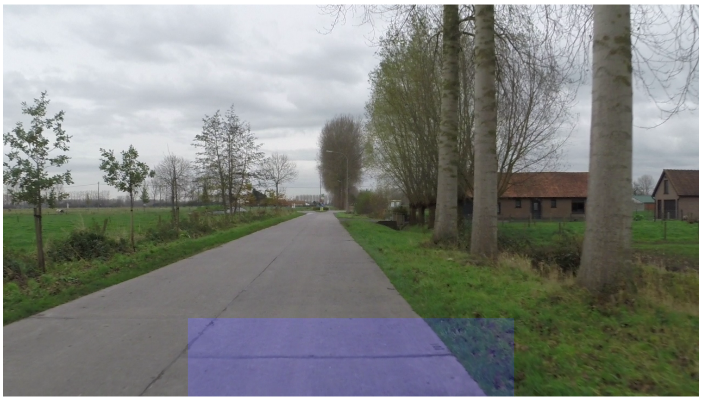


Figure 2.29: *ROI for direct tracking, highlighted in blue.*

The size of the ROI is an important consideration. Firstly, it will have a large impact on computational cost. Secondly, if the ROI is too small, there is a risk that in cases of low ground plane detail, the sum of absolute differences function will be insufficiently steep, leading to increased susceptibility to noise. If the ROI is too large on the other hand, the optimization may be heavily biased towards the alignment of roadside objects because these tend to exhibit much larger contrasts than the road texture itself. Roadside objects have a much lower probability of being in the ground plane than on-road features, and will induce a strong rotation bias on the odometry if they are. Figure 2.29 shows the empirically determined ideal ROI size for our application.

Comparing the trajectories prior to the direct optimization and before shows that in general direct tracking is slightly beneficial for accuracy. Looking at the absolute difference image based on the initial odometry estimate and the absolute difference image after direct optimization shows that this approach generally does converge towards the correct optimum (Figure 2.30). However, direct tracking sometimes fails to converge to the correct odometry parameters at higher speeds due to motion blur, in which cases the optimization becomes sensitive to image noise. Motion blur and image noise are both inversely correlated with light incidence; standard adaptive cameras will try to compensate for low light conditions by increasing gain and/or exposure. In these cases of high speed and low light, tracking is sometimes lost completely; the method does not recover from the erroneous estimation. Gaussian blurring was tried to decrease the noise sensitivity, but this did not solve the problem completely. A detection and recovery mechanism could be conceived to avoid this problem; such a mechanism however would have to rely on a different method altogether and is clearly an unwanted complication.

Figure 2.31 shows the rotation and translation errors with and without direct

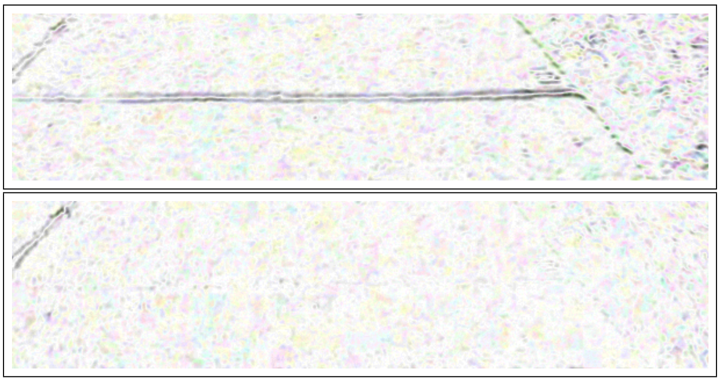


Figure 2.30: *Comparison of absolute difference images between transformed ROIs prior to direct optimization (top) and post-optimization (bottom). Higher intensities indicate bigger absolute differences. In this case correct convergence is achieved and the difference image shows mostly camera noise, which is colored because the difference is computed in the R, G and B channels separately.*

tracking for 18km of test trajectories in a mostly rural and suburban settings in good light conditions with speeds below 25m/s. The evaluation was performed as explained in Section 2.4, with the exception that DGPS was used as ground truth for lack of a more accurate reference. HDOP was less than 1 meter for the entire evaluation trajectory, corresponding to excellent GPS performance. The inaccuracy of the ground truth could introduce an error of the order of 1% for the 100m measurements in the worst case, 0.5% for 200m measurements, etc. It should be noted that the GPS measurements are also Kalman filtered which will improve the accuracy over straight sections. The evaluation against this D-GPS ground truth is therefore meaningful to small fractions of a percent.

We can see that the improvement by direct tracking is only slight. We may conclude that the remaining inaccuracies in the proposed odometry estimation are most likely caused by the ambiguities in viewing angle and the violations of the planarity assumption; therefore the direct tracking as we applied it is of limited value.

Additionally, direct tracking increases the computational complexity more than twofold, partially negating one of the advantages of the proposed method.

In conclusion, direct tracking can be used as a post-processing step and yields a small accuracy benefit, but for real-time applications the computational cost penalty may be prohibitive.

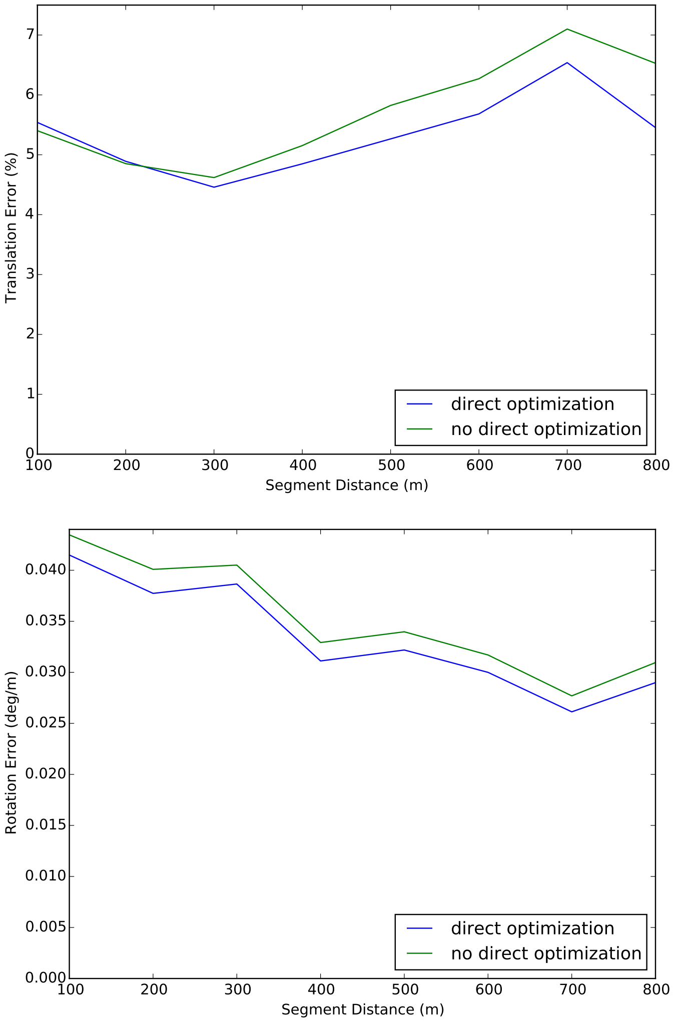


Figure 2.31: *Comparison of translation (top) and rotation (bottom) errors for 18km rural/-suburban test set with and without direct tracking.*

2.5.4 Comparison to naive ground plane feature tracking

The proposed method is novel in two ways. Firstly, odometry calculation is reduced to a 2D problem by considering only the ground plane, which is novel for standard camera placement on road vehicles. Secondly, the overlap voting concept provides robustness to outliers as well as an elegant way of eliminating the unknown suspension angles from the problem. One may wonder to which extent the second concept contributes to the overall performance obtained. We will therefore compare our results against a naive ground plane feature tracker which does not model uncertainty on feature location as a region, but instead matches features between consecutive frames based on appearance, and then searches for consensus among the displacement vectors of the matched features. The basic algorithm is outlined below:

1. extract features from ground plane ROI in the previous frame,
2. find the displacement of these features in the current frame,
3. eliminate outliers based on consistency of their back-projected displacement vectors,
4. calculate the odometry from the outlier-free set.

One of the most popular feature extraction and matching methods for both monocular and stereo visual odometry is the Kanade-Lucas-Tomasi (KLT) feature tracker [62, 89, 96] on account of its reliability and efficiency. Like in many trackers, the feature displacement is calculated by minimizing a residual between the image patch in frame i and the image patches of frame $i + 1$ within a local neighborhood of the original feature location. In the KLT tracker the residual is computed efficiently by solving a linear system of equations defined on the gradient vectors. This efficient computation only takes translation into account, but is robust to small deformations of the image patches caused by small rotations, scale changes or skew.

According to the authors, the robustness of the tracking is largely thanks to the feature selection criterion, which is directly formulated on the data conditioning of the tracking calculations. It relies on calculating a 2x2 matrix from weighted image gradients in the window around the feature, determining the eigenvalues of this matrix, and prioritizing features with the highest second eigenvalue. This ranking by second eigenvalue also has the benefit that an arbitrary number of features can be chosen, unlike some other detectors (e.g. FAST).

For comparison with our uncertainty region overlap voting method, we have modified the code of the KLT tracker as supplied by the authors ([11]) to detect features in the same ground plane ROI as described in Section 2.3.1, spanning

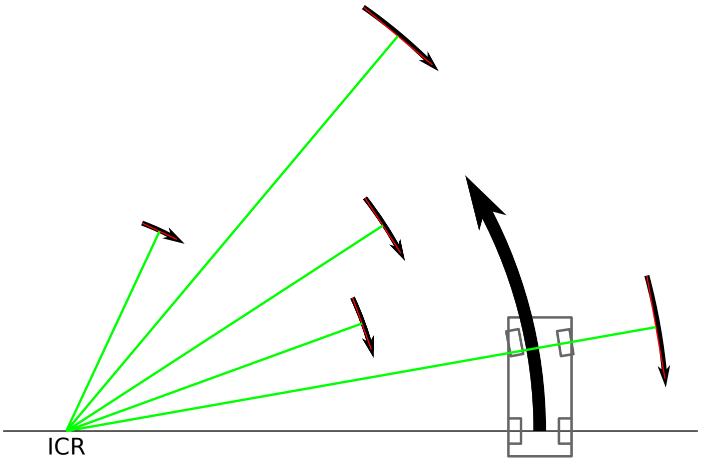


Figure 2.32: Calculation of the ICR based on bisectors of displacement vectors, after back-projection to vehicle axes.

three meters to the side and 15 meters out in front of the vehicle. 50 features are detected and tracked, as in our voting method.

The implementation of the KLT tracker allows for a consistency check among the feature tracks in order to weed out outliers. Three options can be specified: translation consistency, similarity mapping consistency or affine mapping consistency. A maximum deviation threshold can also be specified. In our case, the actual transformation is not affine, but projective. The consistency check is therefore not used, and we will deal with the problem of outliers in a more tailored way. At the heart is a RANSAC-like procedure closely related to the U-RANSAC proposed by [95]. The difference between standard RANSAC and U-RANSAC is in the distance criterion to determine the inlier set for the transformation computed from a set of random samples. Where RANSAC typically uses a fixed threshold, U-RANSAC takes into account the uncertainty on the feature location as well as the properties of the transform to extract maximum information even from data points with high associated uncertainty. Its specific application to the problem of planar odometry is described below.

As explained in Section 2.3.3, the assumption is that the vehicle travels along a circular arc. The odometry parameters are therefore the longitudinal velocity, which dictates the length of the arc, and the angular velocity, which dictates the radius of the arc. In practice this comes down to finding the location of the Instantaneous Center of Rotation, which lies on the extension of the rear axle (actually the perpendicular projection of the axle on the ground plane, since our approach ignores the Z dimension). A straightforward way to find the ICR corresponding to

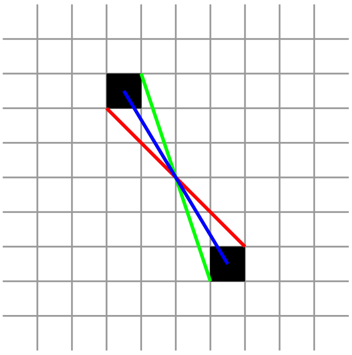


Figure 2.33: *For a given displacement vector (blue) between features discretized to pixel grid (grey), the green and red vectors delimit the uncertainty on the slope of the bisector.*

a feature displacement vector is calculating the intersection between the bisector of the displacement vector (after back-projection from perspective view to ground plane coordinates) and the line through the rear axle, as illustrated in Figure 2.32.

It is clear that the accuracy of the location of the ICR depends on the length of the displacement vector. The KLT feature detection algorithm does not calculate features to sub-pixel level, it therefore suffers from uniformly distributed discretization noise. The shorter the displacement vector of a tracked feature, the higher the uncertainty on the ICR location associated with this discretization noise. In our U-RANSAC like procedure, we will take this into account as follows. The discretization means that a feature may originate from any point within a 1 pixel square centered at the integer feature coordinates, in the image plane. For a given displacement vector, the uncertainty of the bisector is delimited by the bisectors of the sixteen vectors between the four corners of the start pixel and the four corners of the end pixel. It can easily be seen which corners will yield the vectors with the maximum and minimum slope, which will determine the limits on the uncertainty of the ICR (Figure 2.33).

The actual uncertainty distribution of the location of the ICR according to one displacement vector is not uniform between these limits. In order to calculate it, the start and end pixel squares must first be back-projected to the ground plane. Then, for each possible location within the start pixel's quadrilateral, the uniform distribution over the end pixel's quadrilateral must be projected onto the rear axle. To the best of our knowledge there is no straightforward way of describing the final distribution algebraically. Approximating it by sampling would add significant computation time to each RANSAC iteration, and yield little practical benefit since the distribution will be thresholded anyway. We therefore consider the ICR uncertainty distribution uniform within the limits posed by the extremal displacement vectors originating from the pixel uncertainties, i.e. an interval of possible locations on the rear axle line.

We may calculate such an ICR interval for each displacement vector. This essentially means we perform 1-point RANSAC. An inlier is now defined as any feature match for which the ICR uncertainty interval overlaps with the interval of the random sample. In fact, this precludes the selection of random samples altogether; the intervals of all feature tracks can simply be superimposed in a binning process, and the bin with the highest count defines the inlier set. This procedure is very fast and offers all the benefits of a traditional RANSAC program. The most probable location of the ICR is then determined through a least-squares optimization over this inlier set. The longitudinal velocity is computed using least-squares on the displacement vector lengths over the inlier set obtained from the ICR computation.

The results of the KLT ground plane tracker are compared to our proposed method in Figure 2.34. This evaluation is again performed on the 18km test set with D-GPS ground truth mentioned in Section 2.5.3. We can see that although the KLT tracker has slightly better rotation estimation, translation wise its performance is far inferior. The advantage in rotation accuracy is possibly due to the fact that the ICR calculation can take advantage of a finer binning granularity than the consensus image of the proposed image. The much worse translation performance is possibly due to the presence of feedback in the proposed method, which compensates for single-frame estimation errors as explained in Section 2.4. The KLT ground plane tracker has no equivalent mechanism. As a possible improvement in single-frame noise sensitivity, we investigated the performance of the KLT-based method when calculating displacement vectors over 3 and 5 frame intervals instead, but this did not yield higher performance, possibly because fewer features are tracked consistently over this timespan.

It should also be noted that the KLT feature detection and tracking is a factor 2.5 slower than the proposed method.

2.6 Conclusion

We have proposed a novel method for calculating visual odometry which takes advantage of the two-dimensional nature of the motion while efficiently dealing with the challenges of the constrained 6 degrees of freedom of the viewpoint. The method is unaffected by many issues which arise from the traditional solution through epipolar geometry, notably scale drift and small motion inaccuracy.

Evaluation both on the public KITTI dataset and on more challenging recordings of our own show that in terms of accuracy, the proposed method can outperform the basic 8-point solver, if not its more advanced optimizations. The method displays excellent robustness to outliers in difficult urban scenarios, and while a

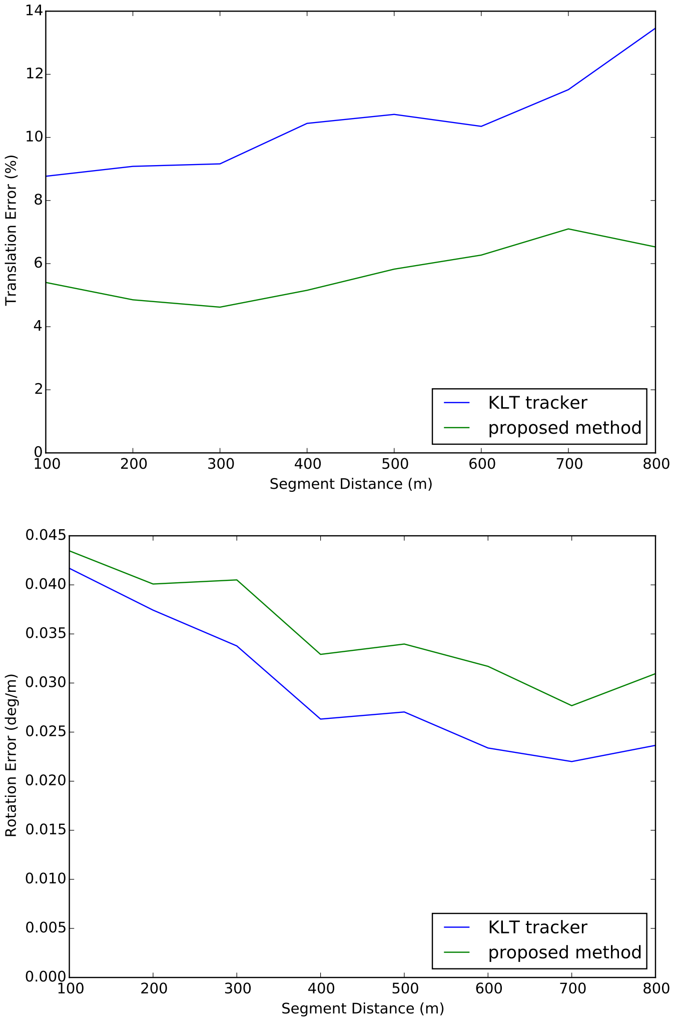


Figure 2.34: *Comparison of translation (top) and rotation (bottom) errors for 18km rural/-suburban test set for KLT tracker and proposed method.*

highly non-planar road surface may introduce a significant bias, the method still tracks macro maneuvers more accurately than the 8-point solver.

The proposed concept of uncertainty region overlap is also shown to outperform ground plane feature tracking without the uncertainty models, and is not significantly improved by adding computationally expensive direct tracking. In terms of computation speed, the method is unmatched by any monocular method ranked in the public KITTI evaluation.

The research contributions made in this chapter are:

- a performance comparison of common feature detectors for the specific task of close-range ground plane tracking on a road vehicle,
- a novel method for modeling the uncertainty on the inverse perspective projection of image feature coordinates to ground plane coordinates,
- a novel method for feature matching and odometry calculation taking into account this uncertainty, which is robust against outliers and deviations from the planar scene model,
- a performance comparison between the proposed method and a reference method,
- an investigation into the merit of direct (non-feature based) tracking as a further optimization,
- a performance comparison between the proposed method and a naive ground plane tracker which does not model uncertainties on the inverse perspective projection.

The work described in this chapter has resulted in two publications in peer-reviewed journals:

- **Robust monocular visual odometry for road vehicles using uncertain perspective projection**, Van Hamme, David; Goeman, Werner; Veelaert, Peter; Philips, Wilfried, *EURASIP Journal on Image and Video Processing (2015)*, Vol. 2015:10 pp. 1-23.
- **Cycling around a curve : the effect of cycling speed on steering and gaze behavior**, Vansteenkiste, Pieter; Van Hamme, David; Veelaert, Peter; Philippaerts, Renaat et al., *PLOS One (2014)*, Vol. 9:7 pp.1-11.

Additionally, parts of the work have been presented at international conferences:

- **Robust monocular visual odometry by uncertainty voting**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *IEEE Intelligent Vehicles Symposium (2011)*, pp. 643-647.

- **Robust visual odometry using uncertainty models**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *Advanced Concepts for Intelligent Vision Systems (2011)*, Vol. 6915 pp. 1-12.

- **Lane identification based on robust visual odometry**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *IEEE International Conference on Intelligent Transportation Systems-ITSC (2013)*, pp. 1179-1183.

- **A Visual SLAM system with mobile robot supporting Localization services to visually impaired people**, Nguyen, Quoc Hung; Vu, Hai; Tran, Thanh-Hai; Nguyen, Quang-Hoan; Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *European Conference on Computer Vision (2014)*, Vol. 8927 pp. 716-729.

3 Calibration

3.1 Introduction

Camera calibration is the estimation of the parameters of the camera. Two kinds of parameters are important: those intrinsic to the camera itself, and those relating to its position and orientation in space. This chapter will cover both *intrinsic calibration* and *extrinsic calibration*. While much of the information in this chapter is relevant to calibration in general, it is not meant to be a reference on this topic. Instead the calibration problem will be explored in the context of our application, and we will only go as deep as is required for this task.

This chapter is structured as follows. Section 3.2 deals with intrinsic calibration. The relevant parameters and their effects are introduced in Section 3.2.1. Section 3.2.2 gives a brief overview of the algorithms in literature that estimate these intrinsic parameters. The most widely used of these algorithms is then analyzed to quantify its typical accuracy. Finally the effects of typical calibration errors on the performance of both the proposed method and the 8-point solver of Geiger et al. [35] are investigated in Section 3.2.3 to compare their tolerance for slight intrinsic calibration errors.

Extrinsic calibration is covered by Section 3.3. Section 3.3.1 gives an overview of how the position and orientation are typically defined. In Section 3.3.2, we analyze the importance of the different extrinsic parameters for the proposed visual odometry method. This analysis is split into two parts. First, the extrinsic calibration sensitivity of the method is approached from a theoretical view, by reasoning for each parameter how a calibration error propagates through the algorithm and predicting the magnitude of its effect. Secondly, this theoretical analysis is validated through a series of simulations using artificial video. Two practical calibration procedures are introduced in Section 3.3.3.

Finally, our conclusions about camera calibration and how it affects visual odometry estimation are presented in Section 3.4.

3.2 Intrinsic calibration

3.2.1 Theory

Let us revisit the mathematics of the pinhole camera model as described in Section 2.1. The camera performs a mapping from homogeneous 3D world points $\mathbf{X} = [X\ Y\ Z\ 1]^T$ to homogeneous 2D image points $\mathbf{x} = [wx\ wy\ w]^T$ described by

$$\mathbf{x} = \mathbf{P}\mathbf{X}. \tag{3.1}$$

The camera matrix $\mathbf{P}$ decomposes as

$$\mathbf{P} = \mathbf{C}\left[\mathbf{R}|\mathbf{t}\right] \tag{3.2}$$

in which $[\mathbf{R}|\mathbf{t}]$ define the rotation and translation between world axes and camera axes and $\mathbf{C}$ is the *intrinsic camera matrix* containing the parameters of the projection:

$$\mathbf{C} = \begin{bmatrix} \alpha_x & s & x_c \\ 0 & \alpha_y & y_c \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.3}$$

α_x and α_y are the horizontal and vertical focal lengths respectively, (x_c, y_c) the coordinates of the *principal point* (i.e. the projection of the ray formed by the Z axis), and s models the skew of the sensor (the degree to which it is a parallelogram instead of a rectangle). The skew factor is often assumed zero as it tends to be very small.

The estimation of these parameters in itself is not overly difficult (a closed form solution is presented by Bouguet [13]). However, in practice the pinhole camera model is a poor representation of a real camera. As we explained in Section 2.1, actual pinhole cameras suffer from either poor light sensitivity due to the tiny dimensions of the hole or visual acuity (i.e. *sharpness*) when the hole is enlarged. The solution is the use of lenses. A lens is a device, usually made out of a curved glass body, which is able to focus beams of light on to a single spot (the focal point) by clever use of the refraction of light on the boundary between two different mediums (Figure 3.1). Using a lens instead of the pinhole allows to capture much more light, while still having high visual acuity.

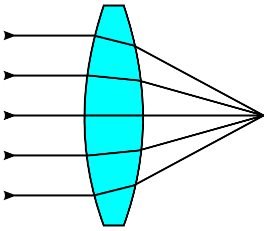


Figure 3.1: *Light refraction in a simple camera lens.*

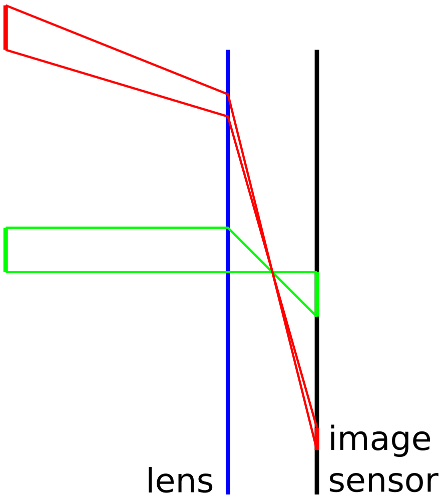


Figure 3.2: *Barrel distortion in a wide angle lens. The red and green objects are the same size, but the red object is imaged much smaller, and closer to the red projection than expected.*

A lens introduces problems of its own however. The refraction angle of the light depends somewhat on the wavelength of the light, causing a single lens to have slightly different focal points for different colors. This is called *chromatic aberration* and can manifest itself as colored fringes around sharp edges in the image.

Another problem is *radial distortion*. This effect causes the scale of the projection of an object to depend on the angle between the light rays coming from the object and the viewing direction of the camera. One consequence is that straight objects (e.g. light poles) do not appear straight in the image.

The most common form of radial distortion is *barrel distortion*, where objects in the center are magnified relative to those closer to the edges. Barrel distortion is commonly seen in wide angle lenses (i.e. lenses with a small focal length). Figure

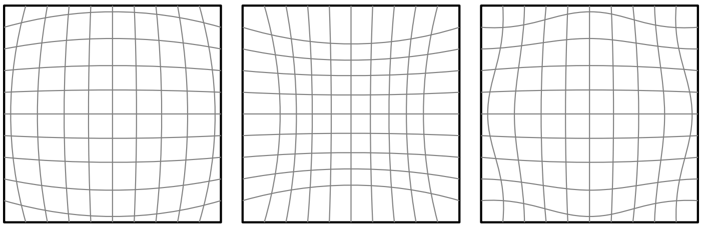


Figure 3.3: *The three types of radial distortion. From left to right: barrel distortion, pincushion distortion, mustache distortion.*

3.2 illustrates this effect. The opposite effect can also occur, called *pincushion distortion*. Sometimes a combination of the two occurs, with barrel distortion in the center and pincushion distortion near the edges. Figure 3.3 shows the different kinds of radial distortion.

Radial distortion can be modelled by an additional nonlinear transformation after the perspective projection. Projected points are moved inwards or outwards on a line through the principal point. Let (x_c, y_c) be the image coordinates of the principal point and (x_i, y_i) the image coordinates of any other point after (distortion-free) perspective projection. The image coordinates under radial distortion (x'_i, y'_i) are then given by

$$x'_i = x_i(1 + K_1 r^2 + K_2 r^4 + ...)$$

$$y'_i = y_i(1 + K_1 r^2 + K_2 r^4 + ...) \tag{3.4}$$

where $r = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$ and $K_1, K_2, ...$ are the radial distortion coefficients[107]. Negative K_1 is characteristic for barrel distortion, positive K_1 for pincushion distortion. Moustache distortion results from a sign change in the K series. In general, radial distortion is well described by only the first two or three terms.

Another type of distortion is *tangential distortion*. This type of distortion occurs when the image sensor and the lens are not mounted parallel to each other in the camera. The effect is that objects projected onto the more distant parts of the sensor are magnified slightly. This effect is typically much smaller than the radial distortion. Under the notation defined above it is given by

$$x'_i = x_i + [2P_1 x_i y_i + P_2(r^2 + 2x_i^2)]$$

$$y'_i = y_i + [2P_2 x_i y_i + P_1(r^2 + 2y_i^2)] \tag{3.5}$$

with P_1 and P_2 the tangential distortion coefficients.

3.2.2 State of the art

Several methods have been proposed to estimate the intrinsic and distortion parameters (e.g. Hartley [45], Heikkila and Silven [47], Pollefeys and Van Gool [80]), usually based on iterative procedures. The most popular to date seems to be the method of Bouguet [13], which draws elements from Zhang [106] as well as Heikkila and Silven [47]. The procedure involves the user waving a regular planar pattern (typically a checkerboard) in front of the camera (Figure 3.4), and an iterative joint estimation of intrinsic parameters and distortion coefficients. The general algorithm is given below.

1. Detect the checkerboard pattern's m crossings in n images.

2. Initialize all distortion coefficients to zero.

3. Calculate initial intrinsic parameters from the first image using the closed form solution of [13].

4. Perform the distortion correction on the image using Equation 3.4 (and Equation 3.5 if required).

5. Calculate the n homographies between the known checkerboard layout and its projections in the n images (using least squares on the overdetermined system).

6. Calculate the expected position of all m points in each of the n images using the estimated homography and distortion coefficients.

7. Calculate the total reprojection error as the sum of the distances between all $m \times n$ observed points and their expected position.

8. Iteratively refine the distortion coefficients and intrinsic parameters using the Levenberg-Marquardt algorithm to minimize the total reprojection error (steps 4-7).

Typically, between 20 and 50 checkerboard frames are used for calibration. Ideally the positions of the pattern in the frame are uniformly distributed across the image. This is primarily important for the distortion coefficients; the higher order model requires to measure distortion near the center as well as the edges.

While intrinsic camera calibration is generally considered a solved problem (and the above algorithm is regarded as reliable), in practice it proves to be a nontrivial process, and there can be considerable differences between the results of consecutive calibration efforts due to a number of factors:

- failure of unsupervised algorithms to correctly detect the checkerboard crossings,

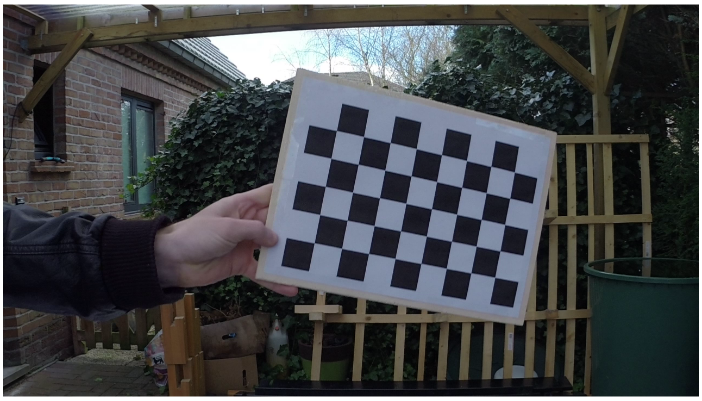


Figure 3.4: *Calibration checkerboard waved in front of camera.*

- insufficient coverage of checkerboard positions across the image,
- motion blur or over/underexposure complicating accurate detection of checkerboard crossings,
- use of insufficient number of images for calibration.

In order to quantify the sensitivity of intrinsic calibration to the size and makeup of the set of calibration images, we have performed the following experiment. A three minute calibration video was made using a 5×8 checkerboard. Unsupervised checkerboard detection was performed (to 0.1 pixel accuracy) after which the detections were manually screened for correctness, leaving 1644 valid calibration frames (238 inaccurate or wrong detections were removed). Random subsets of increasing size were selected from these frames and used to calibrate the camera. 100 random sets were taken of each size, with sizes increasing per 10 from 10 to 100. Figures 3.5 and 3.6 show the variation in focal distance and principal point coordinates for each number of frames. This clearly shows that calibration remains somewhat sensitive to the particular selection of frames even when many are used; the standard deviation of the focal distance for sets of 50 frames is still over 1.1% of its mean value, and 0.62% for 100 frames. The calculation of the principal point is even more sensitive, with a standard deviation of around 3.3% for 50 frames and 2.4% for 100 frames.

3.2.3 Application to visual odometry

Now that we have established that intrinsic calibration is a rather delicate procedure even in a supervised setting where many frames are used, we will analyze

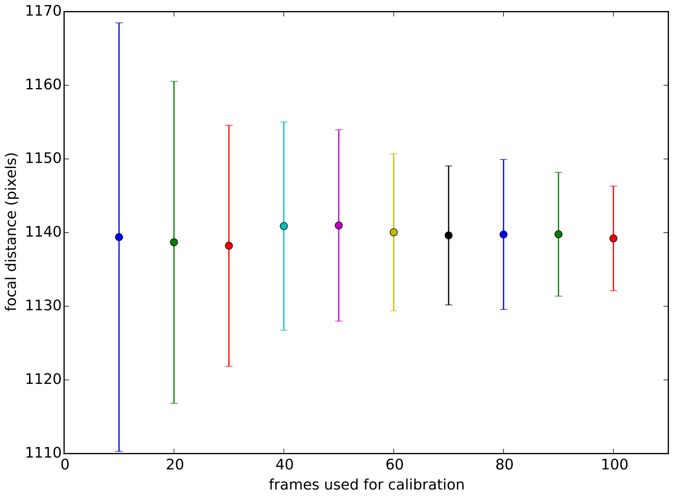


Figure 3.5: *Mean and standard deviation of focal distance vs. number of frames selected for calibration out of 1644 detected checkerboards.*

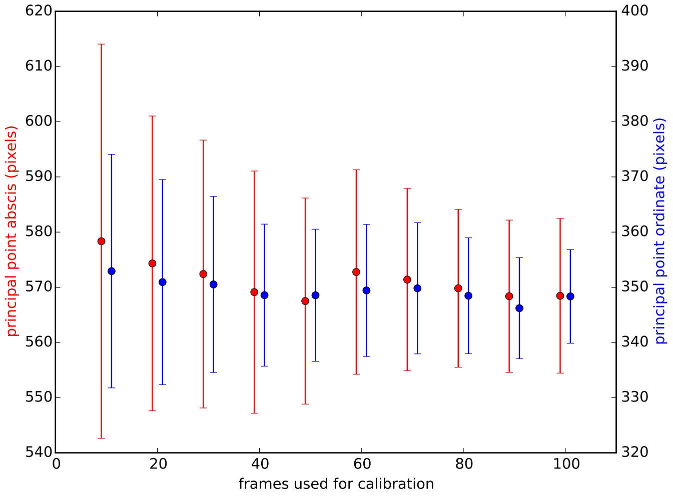


Figure 3.6: *Mean and standard deviation of principal point location vs. number of frames selected for calibration out of 1644 detected checkerboards. X coordinate deviation is plotted in red, Y coordinate in blue.*

how the expected inaccuracy of the intrinsic parameters affects visual odometry calculation with the following experiment.

Using the calibration data provided with the KITTI dataset, we established baseline performance for both our proposed method and the 8-point solver of Geiger et al. [35]. The authors of the KITTI dataset have spent considerable effort calibrating their sensors [34], and we assume the calibration data is of good quality. We will now create random deviations on the three main calibration values (focal distance and principal point coordinates) using normal distributions with the sigmas corresponding to a 20-frame calibration effort as described above and shown in Figure 3.5 and 3.6. Odometry calculation is performed for both methods using these deviating calibration values, and the results are compared to the baseline performance. This gives us an idea of the sensitivity of both methods to calibration inaccuracy. 200 sets of randomly adjusted calibration parameters are evaluated. The increase in error is plotted against *calibration distance*, which we define as the average relative deviation of the three parameters (e.g. if all three randomly drawn parameters differ 1% from the actual values, the calibration distance is 1%). Figure 3.7 shows the evolution of the error against calibration distance. It should be noted that there are complex interactions between the calibration parameters, and as a result different combinations with the same calibration distance may yield different odometry errors. The regression lines clearly show the performance trends however, with the 8-point solver proving much more sensitive to calibration inaccuracy than the proposed method. For the 8-point solver, translation error increases by 1.77% of the traveled distance per 1% of average calibration error. The proposed method is almost impervious to calibration errors of this magnitude, with just a 0.13% increase in translation error per 1% of calibration distance.

An explanation for this much higher resilience to calibration error is in the combination of the margins afforded by the uncertainty region overlap mechanism and the vehicle model imposed on the consensus building. Because feature location is only loosely defined, the vehicle model is able to enforce consistency from the estimation even when the regions are slightly shifted. For example, the effect of an inaccurately calibrated focal distance on the proposed method is that the inverse perspective projections will be non-isotropically scaled; i.e. the vertical axis of the ground plane projection will be stretched slightly. However, the uncertainty on the vehicle's pitch angle means that observation uncertainty regions are increasingly elongated in the vertical direction as the features lie further from the vehicle, so even distant features will still overlap with the prediction uncertainty regions despite this stretching of the vertical axis. Distant features will therefore still consent with the Hough space peak formed by closer (and therefore less displaced) features. The 8-point solver on the other hand employs a least-squares approach to the odometry calculation, and the larger triangulation errors for distant features have a much higher impact in such a scheme.

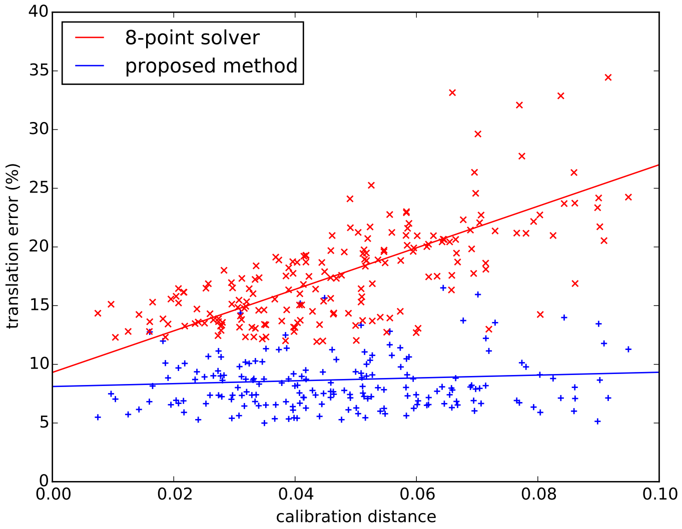


Figure 3.7: *Comparison of sensitivity to calibration errors of 8-point solver by Geiger et al. and proposed method.*

The estimation of distortion parameters proves less problematic. Even though some variation in the distortion coefficients is present in the experiment described earlier, qualitative analysis shows a generally good rectification; even for the wider angle lens of a GoPro camera, straightness of lines is fully restored. Distortion correction is a computationally expensive operation when applied on whole images (it involves a nonlinear map and some form of interpolation), but for feature-based methods it is possible to save on the computation time by correcting after feature extraction, when only the point coordinates need to be transformed and correction becomes relatively cheap.

Since the proposed method already has mechanisms to cope with inaccurate feature extraction, it is worth investigating if correction is at all necessary, or whether we could simply increase the size of the observation uncertainty regions to allow for the displacement caused by radial distortion. An example frame illustrating the degree of distortion in the 18km test set video is shown in Figure 3.8. This video was captured using a GoPro Hero4 camera on its narrowest field of view setting, and we believe this degree of radial distortion is representative for compact optics as would be used in an automotive application. Figure 3.9 shows the difference in odometry error when using distortion correction (undistortion) or not on this 18km test set. The mean translation error for the undistorted case is

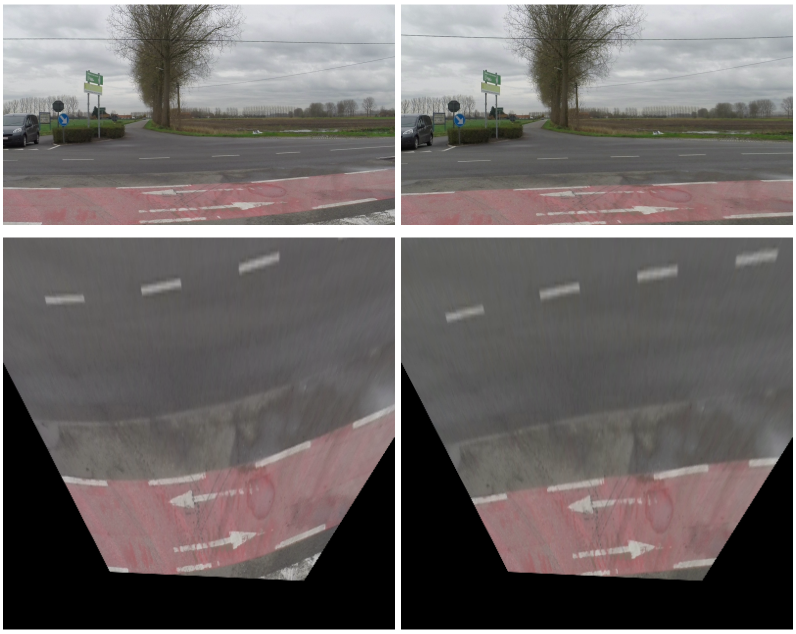


Figure 3.8: *Comparison of video images (top) and their inverse perspective projection (bottom) without (left) and with (right) radial distortion correction. Remaining non-straightness of lines may be due to nonplanarity of the road surface.*

4.96% vs. 5.45% when not using undistortion. The mean rotation error drops from 0.0335 degrees per metre to 0.0298 degrees per metre when enabling undistortion. When using post-feature extraction undistortion, the framerate penalty is less than 5%. We may conclude that distortion correction is a worthwhile improvement.

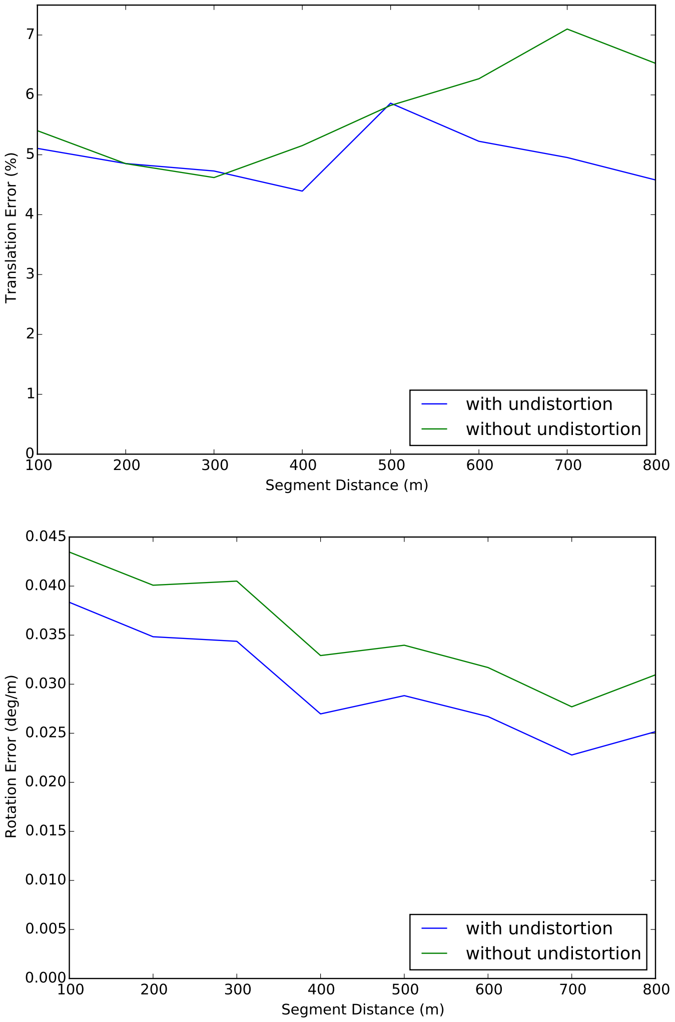


Figure 3.9: *Comparison of translation (top) and rotation (bottom) errors for 18km rural/-suburban test set with and without distortion correction.*

3.3 Extrinsic calibration

3.3.1 General theory

In addition to the intrinsic parameters, which pertain to the properties of the imaging process inside the camera, many applications also require some form of extrinsic calibration, i.e. determining the configuration of the camera with relation to the scene. At most six parameters need to be estimated: the 3D position of the camera and its 3D orientation. Depending on the application, some or all of these parameters may be irrelevant. It is also worth noting that the six degrees of freedom can be expressed in various forms. The position may be expressed as a vector in a frame of reference of choice, or it may be expressed as a set of known distances to objects in the scene. For the description of camera orientation, common conventions include Euler angles or Tait-Bryan angles. Euler angles describe the rotation between the world and camera axes as three successive rotations around two axes (e.g. $z - x - z$), whereas Tait-Bryan angles describe three successive rotations around three axes (e.g. $x - y - z$). Another way to define the camera orientation would be to define a target point, a world point which lies on the principal axis of the camera.

Unlike the intrinsic parameters, extrinsic parameters may be measured directly. Camera position for example, can often be measured relatively simply, especially in indoor situations where perpendicular surfaces are plenty. In less structured environments, extrinsic calibration can still be performed using more advanced measurement techniques from the science of land surveying. Measuring camera orientation using external tools however, is prone to be inaccurate on account of the small size of a camera, and the fact that there is no guarantee the internal camera axes (determined by lens and sensor mounting) are aligned with its housing. In the case of a camera integrated in a vehicle, estimating camera orientation externally may well be impossible.

Instead, extrinsic calibration can be done partly or completely from the camera image itself. When the intrinsic parameters are known, it is easy to see how the observation of known world points carries information about the camera's positioning. However, knowing the camera matrix $\mathbf{C}$ is not strictly required. A distinction is made between *implicit* and *explicit* calibration in this regard. Implicit calibration means that the transformation $\mathbf{P}$ between the world and the camera image is estimated directly from image and real world point correspondences, without calculating any physical camera parameters. For many applications, knowing only $\mathbf{P}$ is sufficient. Explicit calibration aims not just to estimate $\mathbf{P}$, but also at its constituent parts $\mathbf{C}$, $\mathbf{R}$ and $\mathbf{t}$, and the individual single-axis rotation angles which form $\mathbf{R}$. Explicit calibration is often preferred because it allows to separate the distortion model from the pinhole model and allows for more accurate undistortion as a result[47].

Explicit calibration is a more difficult problem than implicit calibration. For implicit calibration, a reliable and simple approach is to to compute the matrix $\mathbf{P}$ which minimizes the reprojection error, as we described in the basic algorithm in Section 3.2. The resulting matrix has 11 degrees of freedom (as it is defined on homogeneous coordinates, up to scale). When considering explicit calibration, even in the distortion-free case, there are 14 unknowns (or 13 when assuming zero skew).

The standard way of performing explicit extrinsic calibration is by using a reference plane. With regard to control points in the reference plane (i.e. real world measurements), the matrix $\mathbf{P}$ is a homography, since it describes a plane to plane transformation. When the camera matrix $\mathbf{C}$ is known from intrinsic calibration, the homography can be decomposed into a rotation matrix and translation vector, using the SVD-based techniques described in Section 2.3. The process typically yields four solutions, two of which can be immediately discarded as they represent situations where the control points are behind the camera. The other two solutions correspond to camera positions mirrored across the plane of the control points, and the correct one can be chosen based on the expected normal vector of the control plane.

The elements of the rotation matrix $\mathbf{R}$ are trigonometric functions of angles. For example, consider the $x - y - z$ variant of Tait-Bryan angles. With α, β and γ denoting the three rotation angles, $\mathbf{R}$ is given by

$$\begin{bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \cos\alpha\sin\gamma + \cos\gamma\sin\alpha\sin\beta & \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\cos\beta\sin\alpha \\ \sin\alpha\sin\gamma - \cos\alpha\cos\gamma\sin\beta & \cos\gamma\sin\alpha + \cos\alpha\sin\beta\sin\gamma & \cos\alpha\cos\beta \end{bmatrix}. \tag{3.6}$$

The trigonometric functions make it easy to extract the individual rotations. They also provide a way to determine the quality of the decomposition result: relations between the elements must hold for $\mathbf{R}$ to be a rotation matrix. In the case of the $x - y - z$ Tait-Bryan angles as in Equation 3.6, one may compute the angles as follows:

$$\begin{aligned} \alpha &= \arctan\frac{-r_{23}}{r_{33}}, \\ \beta &= \arctan\frac{r_{13}}{\sqrt{r_{11}^2 + r_{12}^2}} \\ \gamma &= \arctan\frac{-r_{12}}{r_{11}}. \end{aligned} \tag{3.7}$$

Using ratios of elements ensures the result is unaffected by uniform scaling of the matrix (which may be the case when it is estimated from homogeneous coordinates). Note that using $x - y - z$ Tait-Bryan angles is just a choice. The breakdown of $\mathbf{R}$ into three single-axis rotations is not unique; we may choose any order of three consecutive rotations around two or three different axes.

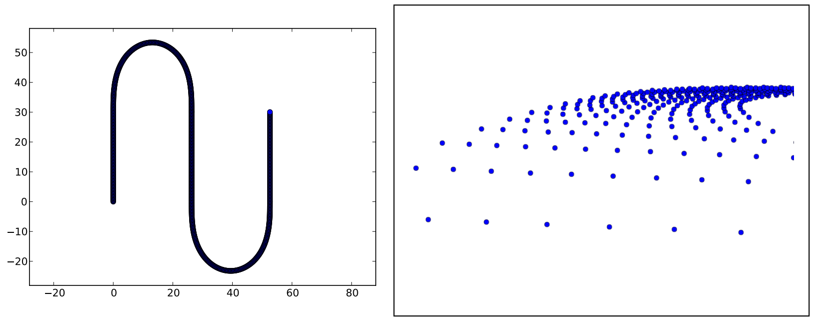


Figure 3.10: *Simulation trajectory composed of two straights and two clothoid turns (left), and sample artifical video frame from start of bend (right).*

3.3.2 Application to the proposed method

In general, visual odometry methods which rely on epipolar geometry do not require full extrinsic calibration. Instead, measuring a single distance is sufficient for the monocular pose estimation algorithms; this serves only to determine the overall scale. However, this calibration must be performed continuously to avoid scale drift.

The proposed method on the other hand requires full extrinsic calibration. For the inverse back-projection, at least the viewing angle (pitch and roll) of the ground plane must be known when the vehicle is resting on a flat and level surface. For the scale of the back-projection, the height of the camera above the ground plane is required, while the implementation of the vehicle model requires the longitudinal distance between the camera and the back axle, the lateral offset to the vehicle centerline, and the heading angle of the camera relative to this centerline.

While slight errors in this calibration will not cause the method to fail (thanks to the robust techniques), they can introduce a bias on the result. We will quantify this bias and determine the sensitivity of the method to errors in each of the extrinsic parameters as it is of high importance for practical applications. For some of the parameters, the sensitivity can be approached analytically. The next section will perform this analysis for some of the parameters. Section 3.3.2.2 will describe a simulation which serves to determine the sensitivity to each of the extrinsic parameters experimentally.

3.3.2.1 Analysis

In this section we will look at how different calibration errors translate into errors on the ground plane coordinates of tracked features, and draw general conclusions about the expected odometry error based on the kinematic model. The analysis described here does not take into account the various robustness mechanisms, which

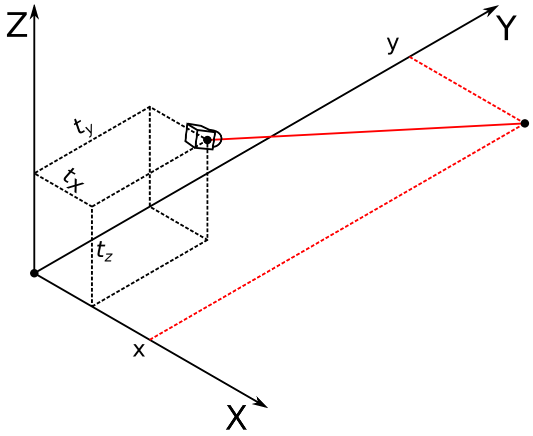


Figure 3.11: *Zero roll, zero heading camera with slightly downward pitch. Red line depicts projection of a ground plane feature onto the camera.*

may in some cases reduce the final error when compared to these theoretical expectations.

Consider the case of a forward facing camera, with zero roll and zero heading and slightly downward pitch angle β, depicted in Figure 3.11. We consider this to be the general configuration for a multi-purpose camera; pitch is the only angle which may deliberately be chosen different from zero to cover more of the ground plane. This is also the configuration used in the Diepenbeek/Hasselt dataset described in Chapter 2. The conclusions which we will draw from this configuration can easily be extended towards the fully axis-aligned case as found in the KITTI dataset by setting $\beta = 0$ in the formulas.

Let (t_x, t_y, t_z) be the true translation of the camera with regard to the world origin, expressed in world coordinates, and $(\overline{t_x}, \overline{t_y}, \overline{t_z})$ the estimated translation vector through extrinsic calibration. With ϵ denoting a calibration error,

$$\begin{aligned} \overline{t_x} &= t_x + \epsilon_x, \\ \overline{t_y} &= t_y + \epsilon_y, \\ \overline{t_z} &= t_z + \epsilon_z. \end{aligned} \tag{3.8}$$

Let the true location of a feature point in ground plane coordinates be (x, y), and its projected image coordinates (x_i, y_i). The proposed method will back-project the points from image coordinates to estimated ground plane coordinates $(\overline{x}, \overline{y})$. The back-projected y-coordinate of the point can be determined from the following relationship (see also Figure 3.12):

$$\tan(\beta + \psi) = \frac{\overline{t_z}}{\overline{y} - \overline{t_y}} \tag{3.9}$$

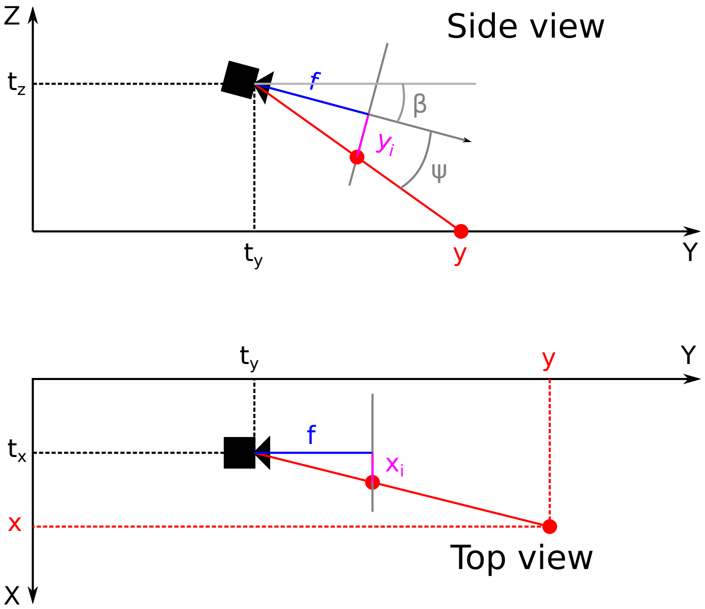


Figure 3.12: *Side and top view of the camera projection of a ground plane feature point.*

in which ψ is the angle between the projection ray through $(0, y_i)$ and the principal ray. This angle is calculated from the vertical image coordinate y_i and focal length f as $\psi = \arctan(y_i/f)$, yielding

$$\overline{y} = \frac{\overline{t_z}}{\tan(\beta + \arctan(\frac{y_i}{f})} + \overline{t_y} \tag{3.10}$$

for the estimated y-coordinate of the point. In the top view in Figure 3.12 can be seen that

$$\frac{\overline{x} - \overline{t_x}}{\overline{y} - \overline{t_y}} = \frac{x_i}{f} \tag{3.11}$$

which after substituting Equation 3.10 gives

$$\overline{x} = \frac{x_i}{f} \frac{\overline{t_z}}{\tan(\beta + \arctan(\frac{y_i}{f}))} + \overline{t_x} \tag{3.12}$$

In expressions 3.10 and 3.12 we can easily see the effect of translation calibration errors on the back-projected point coordinates. An error in lateral camera offset ϵ_x manifests only in a lateral shift of feature point coordinates by the same distance. This will have no effect on straight sections. However, in a bend the estimated ICR will also be shifted. A calibration error on t_x will therefore cause a rotation error in the direction of the calibration error during both left and right turns, but not on straight sections. When a ground truth trajectory is available for a test trajectory, this property can be used to identify a lateral calibration error. Consider a single feature correspondence whose displacement vector has length l in ground plane coordinates and the midpoint of the displacement vector is at a distance y from the rear axle. With r denoting the distance between the feature point and the ICR, the heading change is calculated as $\alpha = \frac{l}{r}$. When the lateral offset is calibrated with a positive error ϵ_x, the rotation during a right turn will be underestimated by a factor $\frac{r}{r+\epsilon_x}$. During a left turn, rotation is overestimated by a factor $\frac{r}{r-\epsilon_x}$. The situation reverses for negative ϵ_x. Considering that the minimum turning circle diameter of an average vehicle is around 10 meters, the rotation error will be small (under 2% for a 10 cm calibration error in a minimum-radius turn).

An error in longitudinal camera offset ϵ_y results only in a longitudinal shift of estimated feature point coordinates. Again, on straight sections the method is unaffected by this. During turns, the estimated ICR can shift significantly however. The ICR will effectively be assumed to lie on a line at a distance ϵ_y behind the vehicle's rear axle. Any turn regardless of direction will therefore be underestimated for positive values of ϵ_y, and overestimated for negative ϵ_y. This is different from a lateral error, where the error is different in sign for left and right turns. Consider again a single feature correspondence with length l in ground plane coordinates, which makes an angle ϕ with the Y-axis (the straight-ahead direction). The distance between the midpoint of the displacement vector and the rear axle is y. From the kinematic model of the vehicle follows that the ICR is estimated to be at a distance $r = \frac{y}{\sin\phi}$ from the feature point. The heading change α can be calculated from this feature correspondence as $\alpha = \frac{l}{r} = \frac{l\sin\phi}{y}$. Under a longitudinal calibration error, the estimated heading change is $\alpha' = \frac{l\sin\phi}{y+\epsilon_y}$. The ratio of estimated and true heading change is therefore inversely proportional to the ratio of the estimated and true Y-coordinate of the feature: $\frac{\alpha'}{\alpha} = \frac{y}{y+\epsilon_y}$. For a typical scenario this amounts to a 1.5% rotation error for a 10cm longitudinal error.

An error ϵ_z in the vertical camera offset results in a uniform scaling of $\overline{x} - t_x$ and $\overline{y} - t_y$ by a factor $\frac{t_z+\epsilon_z}{t_z}$. Important to note is that this is not a pure scaling of the estimated ground plane coordinates on account of the terms t_x and t_y which do not get scaled by the calibration error in t_z. A vertical error will therefore manifest itself on a straight section as an overestimation of velocity, which clearly sets it apart from the longitudinal and lateral errors. To analyze the effect on rotation accuracy, consider again a feature correspondence with length l and angle ϕ with the Y-axis at a distance y to the rear axle. The scaling of $x - t_x$ and $y - t_y$ is uniform, therefore the calibration error will not change the slope of the displacement vector, only its length. If t_y were zero and the camera was on the line through the rear axle, the abscissa of the ICR would be scaled by the same factor $\frac{t_z+\epsilon_z}{t_z}$. However, for nonzero t_y the ICR lies a further distance t_y rearward. Let the true location of the ICR be $(y/\tan\phi, 0)$ and $s = \frac{t_z+\epsilon_z}{t_z}$. The estimated location of the ICR is

$((s(y - t_y) + t_y)/\tan\phi, 0)$. This means that the scaling in arc length according to this feature correspondence is larger than the scaling of the turning radius, and bends will be slightly overestimated for positive ϵ_z. The effect is likely to be small given the typical dimensions of the measurements involved; for a 10% error in t_z, a feature distance of 5m and t_y=1m, the expected error is only 1.8% in a minimum-radius turn.

A final conclusion to be drawn from Equations 3.12 and 3.10 concerns the pitch angle β. A calibration error in the pitch angle gives rise to a uniform scaling in $x - t_x$ and $y - t_y$, but the scaling factor depends on the vertical image coordinate y_i of the feature. Figure 3.13 shows the evolution of the scale factor versus vertical image coordinate for a typical example. The magnitude of the odometry error caused by the pitch mis-estimation depends on the vertical distribution of the tracked features. An overestimation of (downward) pitch causes a reduction in scale of the back-projected coordinates. Traveled distance will be underestimated, and rotation will be slightly underestimated as well on account of the unscaled terms t_x and t_y, similar to the situation for vertical offset error (except the scaling is now in the denominator). This error may therefore be hard to distinguish from a vertical offset error.

In the above formula we have ignored roll and heading as they are normally chosen as close to zero as possible. However, we can still analyze qualitatively the expected effect of calibration errors in these angles. A mis-estimated roll angle means that feature coordinates on one side (the “high” side) will be scaled down, while feature coordinates on the other side will be scaled up. On both straight and curved sections, this will result in a rotation bias; the difference in apparent ground plane feature velocity left to right is similar to feature motion caused by rotation. The magnitude of the error will depend on the width of the ROI, as this affects the mean absolute lateral offset of the features, which in turn determines the mean difference in observed feature velocity between left and right features.

A heading error will cause ground plane features to seemingly “drift” sideways. Consider a feature correspondence exactly in the middle of the region of interest, i.e. the average feature, in a scenario where the heading is mis-estimated by ϕ radians and the vehicle is driving straight ahead. Let y denote the distance from the rear axle to the midpoint of its displacement vector. According to the kinematic model of the vehicle this would place the ICR at a distance $r = y/\sin\phi$. The heading change according to this feature correspondence is then $\alpha = d/r$ with d the traveled distance. Calculating α for a $\phi = 1°$ in a typical configuration yields an expected rotation error of $0.250°/m$.

3.3.2.2 Simulation

In the previous section, we have analyzed the effect of calibration errors on apparent feature motion and drawn general conclusions about the expected odometry

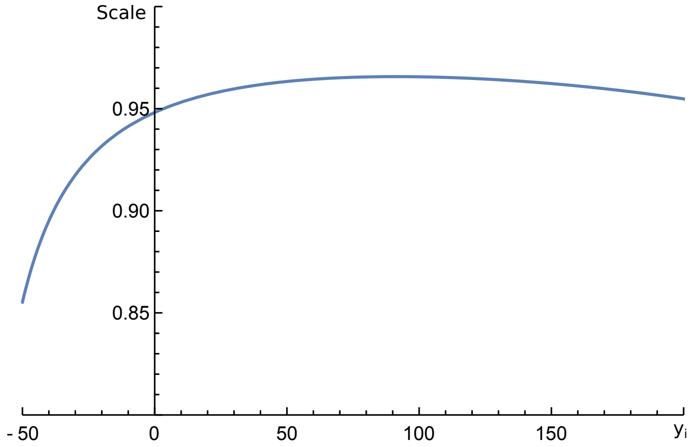


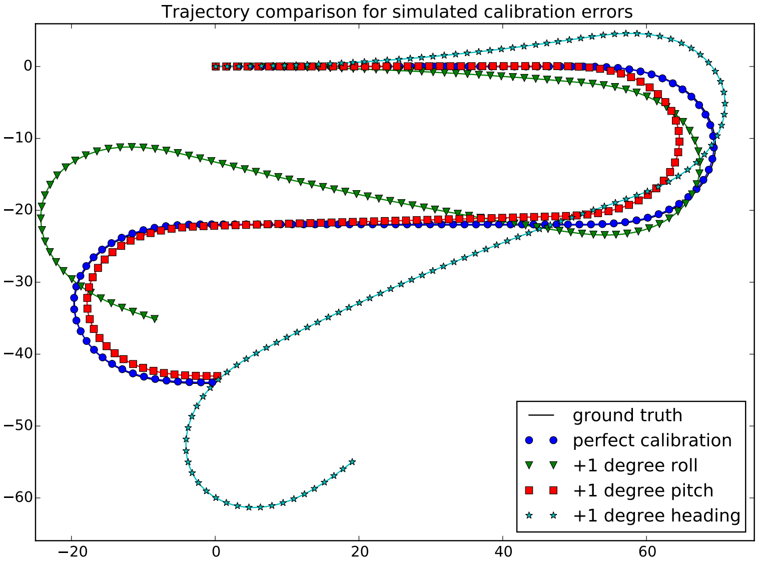
Figure 3.13: *Ground plane coordinate scaling versus vertical image coordinate for a typical example. True pitch in this case was 20 degrees, and estimated pitch 21 degrees. X-axis spans the typical ROI for this application.*

errors. In this section, we will determine the effect of calibration errors empirically. To this end, we performed a simulation in which artificial video is generated for a vehicle moving along an S-shaped point grid, similar to the simulation we described in Section 2.5.2. This experiment serves on one hand to verify the conclusions from Section 3.3.2.1 and on the other hand to quantify the sensitivity of the odometry method in a representative scenario.

The artificial trajectory is two 50m long straight sections linked by one 180° turn left and one 180° turn right. The simulation trajectory and an artificial video frame are shown in Figure 3.10. Points are projected to integer pixel locations; this means the simulation includes discretization noise as it would occur in feature extraction on real video.

The simulation allows us to easily control the error in each extrinsic calibration parameter and analyse 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 10cm was added to one of the translation components. The results can be seen in Figures 3.14 and 3.15.

From the error graphs and reconstructed trajectories, we can see that the most important parameter for translation accuracy is the camera height. A 10cm error in this parameter (on a true height of 1m) results in an overestimation of travel distance of 10%, as predicted by the theory. The translation error is accompanied by



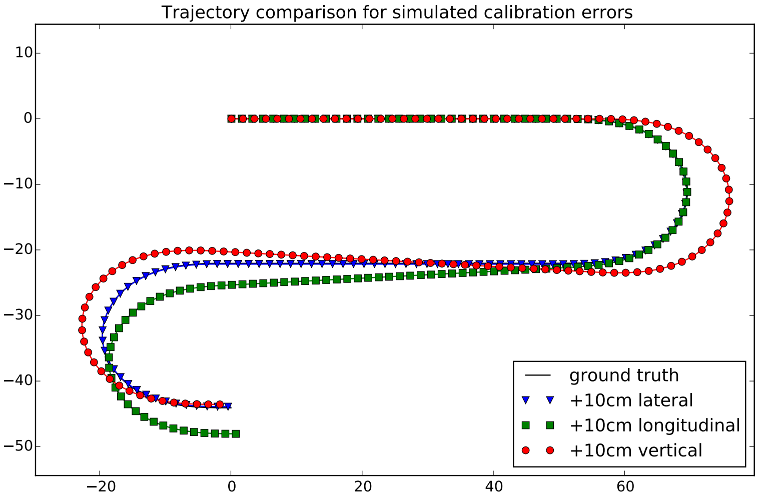


Figure 3.14: *Effect of extrinsinc calibration errors on reconstructed trajectory. Ground truth lines are almost entirely hidden behind the perfect calibration result in the top plot, and behind the lateral deviation result in the bottom figure.*

a small rotation error (1.6%) in each bend, which also corresponds to the estimate presented in Section 3.3.2.1.

The next most important parameter for translation accuracy is the pitch angle. A 1° error gives rise to an overestimation of travel distance by 6.8%. This is accompanied by a 0.9% rotation error in each bend. As expected, these effects are very similar to the ones of mis-estiamted camera height, although the ratio between the translation and rotation error is different and may be used to distinguish between the two.

All other parameters have only very slight consequences for translation accuracy. For rotation accuracy, we see that the heading angle is the primary source of error. A 1 degree heading error results in a rotation error of 0.178°/m, against a prediction of 0.250°/m. The error is mitigated by the consensus voting, as the sideways motion of the features is not consistent with the kinematic model.

The second most important parameter for rotation accuracy is the roll angle. A 1° calibration error causes a 0.087°/m rotation error.

We can conclude that the proposed method is most sensitive to vertical offset, followed by pitch, heading and roll angle. Generally, the offsets are easy to measure in practice, and an error of 10cm 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 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.

3.3.3 Two semi-unsupervised extrinsic calibration algorithms

In the previous sections, we have demonstrated that both intrinsic and extrinsic camera calibration remain difficult tasks. With regard to the proposed visual odometry method however, we may attempt to simplify the process significantly.

Regarding intrinsic calibration, we have shown that our method is very robust to calibration errors below 10%. This means that we may avoid much of the burden of intrinsic calibration altogether, instead using a predefined calibra-

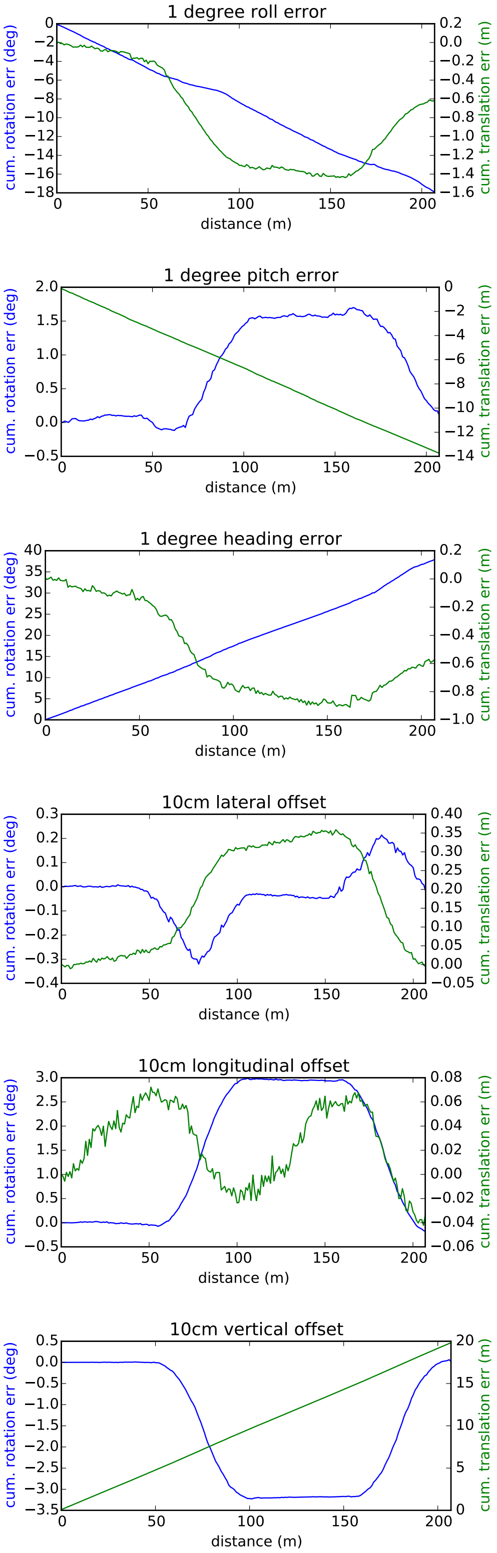


Figure 3.15: *Influence of errors in one of the extrinsic calibration parameters on cumulative rotation and translation error. Straight sections are from 0m to 50m and from 105m to 155m. Note that Y axis scale is different per figure.*

tion profile (e.g. determined as representative for the average camera of the type used). In practice, we have observed that the intrinsic parameters can be carried over between different GoPro Hero 4 cameras without affecting the results. The insensitivity to small intrinsic calibration errors is also a big advantage for potential application in a large fleet of vehicles, as it avoids the requirement of periodic recalibration to counter mechanical drift and the need to spend significant time on calibrating individual cars.

Extrinsic calibration on the other hand, is very important for our method. In the previous section, we have shown that errors in each of the six extrinsic parameters have distinguishable effects on the rotation and translation bias under specific circumstances. We may therefore perform an initial calibration that allows the computation of relatively coarse visual odometry, and then perform online calibration on a trajectory for which ground truth is available. Ideally, the system would be slowly but continuously self-adapting based on sensor fusion (e.g. integration with offline maps and/or satellite positioning). The idea of continuous self-calibration will be briefly revisited at the end of Chapter 4, but its implementation remains future work.

For the initial extrinsic calibration however, two simple procedures are described below. The first is a direct optimization procedure for the rotation angles using the heading-pitch-roll convention to denote the single-axis rotations, using iterative adjustment of single angles and using the inverse perspective projection of the camera image for feedback. It is summarized as follows:

- drive across an empty car park or similar flat, open space with regular markings, making sure to align the driving direction closely with one of the longitudinal markings,
- stop the vehicle behind a rectangular marked area (e.g. a parking space),
- measure the dimensions of the marked area,
- measure the distance from the rear wheel contact point to the marked area,
- measure the lateral distance from the centerline of the vehicle to one of the markings,
- determine the camera rotations as follows:
    1. adjust the pitch angle so the longitudinal markings become parallel in the rectified image,
    2. adjust the roll angle so that longitudinal and lateral markings become perpendicular in the rectified image,
    3. adjust the heading angle so that longitudinal markings become vertical,

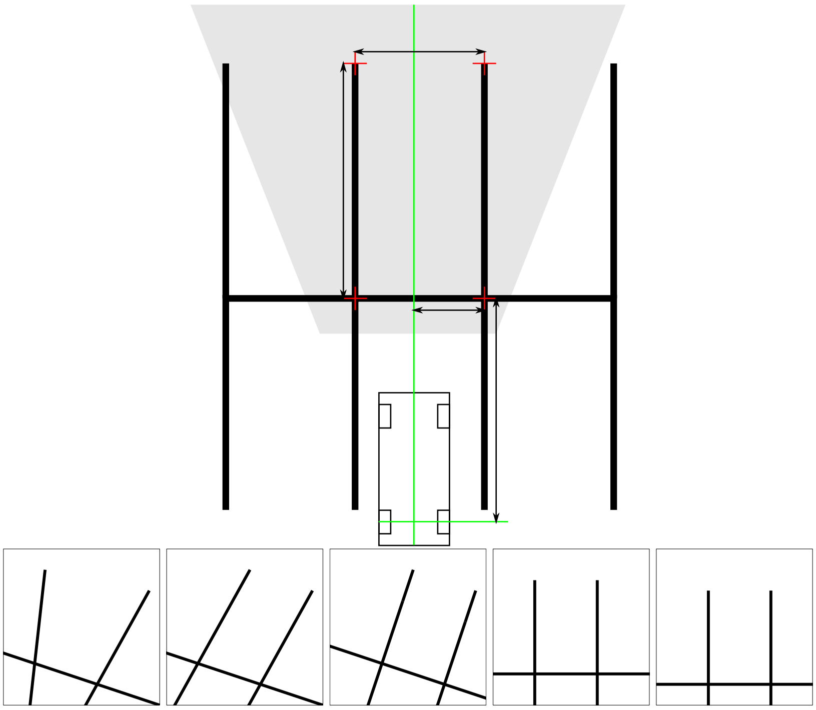


Figure 3.16: *Calibration scenario. Vehicle is lined up with parking grid (top). From left to right: pitch correction, roll correction, heading correction, translation correction.*

4. repeat steps 1-3 if required for accuracy,

- determine camera translation as follows:

  1. adjust the camera height so that the dimensions of the marked area are true to reality (up to a user defined scale),
  2. adjust the camera lateral offset so that the distance from the centerline to the measured marking is true to reality,
  3. adjust the camera longitudinal offset so that the distance from the rear axle to the marked area is true to reality.

The process is clarified in Figure 3.16.

Note that this procedure assumes that images have already been undistorted; in the case of significant distortion, line markings will not appear straight and accurate calibration becomes impossible.

This procedure has the advantage that it can be done quickly once the calibration zone markings have been measured, and it performs explicit calibration in a

Method	Mean repr. err.	Std.
ISAO	3.28	2.63
HD	3.41	2.62

Table 3.1: *Comparison of reprojection error for iterative single angle optimization (ISAO) and homography decomposition (HD).*

step by step progression without requiring operator input beyond clicking on the four marks in the camera image. Additionally, this method optimizes individual rotation angles, ensuring a perfectly conditioned rotation matrix. The resulting calibration is guaranteed to comply with the pinhole camera model.

Alternatively, instead of the iterative calibration described above, the following unsupervised algorithm is also a candidate:

- align the vehicle with the measured markings as described above,
- determine the locations of the corners or endpoints of the markings,
- undistort the corner locations,
- perform homography estimation between the undistorted corner locations and their real-world measurements in the vehicle coordinate frame ($\mathbf{P}$),
- decompose $\mathbf{P}$ into $[\mathbf{R}|\mathbf{t}$ using $\mathbf{C}$ obtained from intrinsic calibration.

This algorithm is potentially less accurate in the individual rotation angles and translation vector, as the constraints on $\mathbf{R}$ are not modelled or enforced in any way. For our odometry method, individual rotation angles must be extracted. The estimated homography is therefore decomposed into a rotation matrix and translation vector using the decomposition method explained in Section 2.3, and the individual rotation angles are extracted from the rotation matrix. The extrinsic calibration matrix is then recomposed using these extracted angles in a similar process to that used in the computation of the observation uncertainty regions (cfr. Section 2.3.2), which ensures a properly conditioned transform.

For comparison, we have measured an empty parking space and run both algorithms. Camera resolution was 1280x720. The user manually indicated the control points in the camera image 10 times to investigate repeatability. The average reprojection error of the two algorithms is then compared, as well as the standard deviation on it. For the method which uses homography estimation, the reprojection error is computed on the recomposed transform, as this is how it will be used in the odometry method. The results are shown in Table 3.1. Figure 3.17 shows the rectified parking space for both algorithms, and the amplified pixel difference between the two.

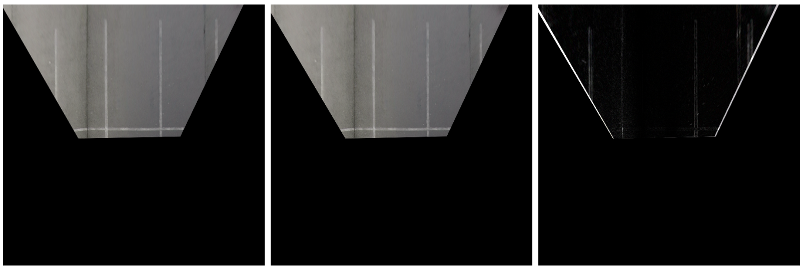


Figure 3.17: Rectified parking space for homography estimation (left), direct angle optimization (center) and pixel difference between the two (amplified by a factor 2 for clarity).

We can conclude that both algorithms yield equal performance, and the main source of error is the clicking on the control points by the user, and not the details of the method. The difference in the acquired calibration angles is below one percent of a degree in all trials, which is too small to make a notable difference on the odometry result.

3.4 Conclusion

In this chapter, we have demonstrated that intrinsic camera calibration is a sensitive process of limited accuracy even when many good calibration frames are used. The proposed odometry method however is insensitive (in fact almost completely impervious) to the residual errors which may be expected from standard checkerboard calibration methods, which is a big advantage over other state-of-the-art methods. Furthermore, distortion correction is shown to yield a worthwhile accuracy improvement.

Extrinsic calibration however, is an important factor in the accuracy of our method. The required extrinsic calibration accuracy calls for fine tuning of the rotation angles after an initial static calibration effort. Two algorithms are provided for the initial calibration, and we described how the individual parameters can be tweaked using a ground truth trajectory.

The main contributions made in this chapter are the following:

- an analysis of the repeatability and accuracy of a standard intrinsic calibration method,
- a comparison of the sensitivity to intrinsic calibration errors of our proposed visual odometry method and a reference method using epipolar geometry,
- an analysis of the specific effects of individual extrinsic calibration errors on the proposed visual odometry method,

- two practical calibration methods to provide initial extrinsic calibration.

Some of the work in this chapter is described in the following journal publication:

- **Robust monocular visual odometry for road vehicles using uncertain perspective projection**, Van Hamme, David; Goeman, Werner; Veelaert, Peter; Philips, Wilfried, *EURASIP Journal on Image and Video Processing (2015)*, Vol. 2015:10 pp. 1-23.

Work related to extrinsic calibration was presented at an international conference:

- **Parameter-unaware autocalibration for occupancy mapping**, Van Hamme, David; Slembrouck, Maarten; Van Haerenborgh, Dirk; Van Cauwelaert, Dimitri et al., *2013 Seventh international conference on distributed smart cameras (ICDSC) (2013)*, pp. 49-54.

4
Sensor fusion & map localization

4.1 Introduction

In the previous chapters we have considered the problem of relative motion estimation, which is an interesting and important topic with many practical applications of its own (e.g. detecting lane changes, aiding in vehicle control, predicting vehicle behavior, assessing accident risk, ...). In this topic we will increase the scope to self-localization, for which relative motion needs to be converted into absolute position.

In literature, self-localization has often been approached from a *recognition* standpoint. Information about the local environment is recorded and stored in a database, and the localization problem is reduced to finding similarities between the live data generated by the vehicle's sensors and the stored data. The most robust solutions thus far have relied on 3D point clouds [23, 41, 74], but as mentioned in Chapter 1 this is an unattractive solution for consumer vehicles due to cost and physical integration constraints. Alternatively, visual features have been used in a similar fashion [6, 7, 46] with good results. Recognition-based localization has several practical problems however, most important of which is the cost of building a rich, densely annotated map of every drivable road and keeping it up-to-date.

Rather than build a densely annotated map, we will perform self-localization using only the sparse, simplified map representation which is readily available commercially as well as through the OpenStreetMap initiative [1]. Rather than use the appearance of the local environment, we will relate the shape of the trajectory to the shape of the local road network. In this respect, our method will be similar to the work of M. Brubaker and Urtasun [63], who use Bayesian inference in a Markovian state-space model to compute the probability distribution over a directed graph of map segments. The probability distribution is formulated as a Gaussian mixture model, which is updated and then simplified after each visual odometry state estimation. While the authors demonstrate the ability to correctly localize the vehicle to within 3.1 meters using stereo visual odometry, in the monocular case the average localization accuracy is only 18.4 meters. Real-time performance is claimed, however this is on a parallelized implementation running on 16 cores and the odometry is subsampled at 1 Hz. Our approach will be broadly similar in principle, but forgoes much of the graph preprocessing and will use multiple hypotheses each modeled by an extended Kalman filter rather than construct a Gaussian mixture model. This greatly reduces the computational cost and allows us to implement map feedback in a straightforward way. Our method achieves positional accuracy within 3 meters for monocular visual odometry, which is comparable to current satellite navigation products available in the consumer market.

This chapter is structured as follow. In Section 4.2 we will analyze the error on position and heading, and demonstrate that the error distribution is well approximated by the posterior uncertainty distribution of an extended Kalman filter.

how to integrate visual odometry with other sensors (Section 4.3), and how to build a real-time, real-world positioning solution on top of it using only offline data sources and vehicle mounted sensors (Section 4.4). This will deliver on the promise we made in the introduction of a complete fair-weather solution for vehicle positioning without any external communications. The algorithms will be evaluated on real data in Section 4.6.

4.2 Error modeling

In the context of autonomous vehicles, in addition to estimating the position of the vehicle it is also important to determine the uncertainty on this estimation. To illustrate this point, let us consider an example on Antwerp's ring road R1, approaching the exit of Antwerp-South. This is a complex junction, with multiple extra lanes in addition to the three main lanes, each intended to filter traffic towards specific destinations. In this situation, a generic navigation instruction (e.g. take the exit, then hold left) can be confusing. However, more specific instructions (e.g. in one hundred meters, move one lane to the right) should only be given when the system has high confidence in its immediate position estimate; it is safer to give generic instructions and rely on the judgment of the driver than to issue a potentially wrong instruction which may confuse the driver.

Many other intelligent vehicle applications benefit similarly from knowing the estimation uncertainty, and in the context of safety it can even be a hard requirement for homologating driver assistance systems.

Visual odometry produces consecutive estimates for the longitudinal velocity v and the angular velocity (turning rate) ω of the vehicle. Integrating these over multiple measurements results in an estimation of the current odometry state

$$\mathbf{s} = [d, \theta]^T$$

in which d is the traveled distance and θ the heading angle relative to the starting position. For navigation or localization purposes however, we need x and y positions in the local world plane rather than the traveled distance, as well as θ:

$$\mathbf{x} = [x, y, \theta]^T.$$

We can analyze the error distributions of the estimates of v and ω, and the uncertainty on the state $\mathbf{s}$ can be calculated as a function of these error distributions and the number of estimates performed since the last exact known state.

The uncertainty on $\mathbf{x}$ however is not only dependent on the error distributions of v and ω, but is also a function of the trajectory itself. Assuming the error distributions of v and ω have zero mean and standard deviations σ_v and σ_ω respectively we may formally write this as

$$\mathbf{x}_k \sim f(x_0, v_0 ... v_k, \omega_0 ... \omega_k, \sigma_v, \sigma_\omega). \tag{4.1}$$

For the remainder of this section, we will abbreviate the probability density function of the vehicle state after k odometry estimates $f(x_0, v_0 ... v_k, \omega_0 ... \omega_k, \sigma_v, \sigma_\omega)$ as f_k. The dependence of this probability distribution on the details of the trajectory is a major obstacle for modeling the *drift* of any odometry estimator.

One possibility to model the possible drift for a given trajectory is using Monte Carlo techniques. Monte Carlo (MC) methods are a class of computational algorithms that rely on repeatedly propagating random samples through a numerical simulation in order to obtain an output distribution. For a more thorough description on MC methods, Bishop [12] is a good reference work. MC simulations are typically used for systems where obtaining a closed form solution is difficult or impossible. In the case of visual odometry, the angular and longitudinal velocities estimated in each frame and their associated error distributions are all inputs of the model, making closed form estimation of f_k intractable. We will therefore perform the following MC simulation.

From a known start distribution f_0 (the prior distribution), a large number of samples is drawn. In practice, the prior distribution could be the posterior distribution of the previous drive, or a starting address (e.g. geotagged location) with its associated uncertainty (in which case the heading component θ would have a uniform distribution). For each sample, a set of simulated odometry parameters $(\hat{v}_1, \hat{\omega}_1$ is generated for time step 1 by taking the exact parameters of a predefined simulation trajectory and adding random noise samples from the typical error distributions on v and ω. These noisy odometry parameters are then used to propagate the start samples through the kinematic model. The result is a sparse distribution of output samples representing the posterior distribution $\hat{f}_1$ of vehicle states which takes into account the uncertainty on the odometry estimation as well as the followed trajectory during the first time step. To obtain a smooth distribution for analysis, kernel density estimation (KDE) can be used. Kernel density estimation (sometimes called Parzen-Rozenblatt windowing) is a technique through which a sparse population of samples can be converted into a smooth distribution [77]. Dirac impulses are placed at the location of the samples, after which a smoothing kernel is convolved with the signal. A suitable bandwidth of the kernel can be computed from the standard deviation of the samples [49], or manually tuned to obtain the desired smoothness.

Through KDE, we obtain the posterior distribution f_1 after one time step (iteration) in the MC simulation. In the next iteration, this distribution serves as the prior distribution from which new samples are drawn, and the process is repeated. When enough samples are used, this Monte Carlo simulation can provide an accurate estimation of the state probability distribution f_k after k estimations for a given trajectory.

Figure 4.1 shows 2000 samples from the final distribution of such a Monte Carlo simulation in the case of a straight 2km trajectory and circular trajectory of the same length, using the error distributions obtained using the KITTI dataset and an exactly known starting state. These results clearly illustrate the dependence of the shape of the posterior distribution on the trajectory.

Using the maximum a posteriori (MAP) principle, this Monte Carlo simulation can also serve to model the distribution of the unknown actual world position when only the starting state and an odometry estimate of the trajectory is known. Maximum a posteriori estimation is a technique in Bayesian statistics where the value of an unknown parameter (or set of parameters) is estimated as the value which corresponds to the mode (i.e. the peak) of a posterior distribution calculated through a closed form, an iterative method such as expectation maximization (EM) or gradient descent or a Monte Carlo simulation. In our context, MAP estimation is used to find the state vector which corresponds to the peak of the posterior distribution estimated through MC simulation.

In the above discourse, we have assumed known error distributions of v and ω. Modeling these input error distributions themselves however is not a straightforward task. Like for most signals which originate for complex systems, the errors are approximately normally distributed when considered on a long enough timespan, but consecutive measurements are not independent. One cause for this is obviously the feedback loop in the odometry estimation; predicted uncertainty regions depend directly on the previous estimation. The proposed method is to

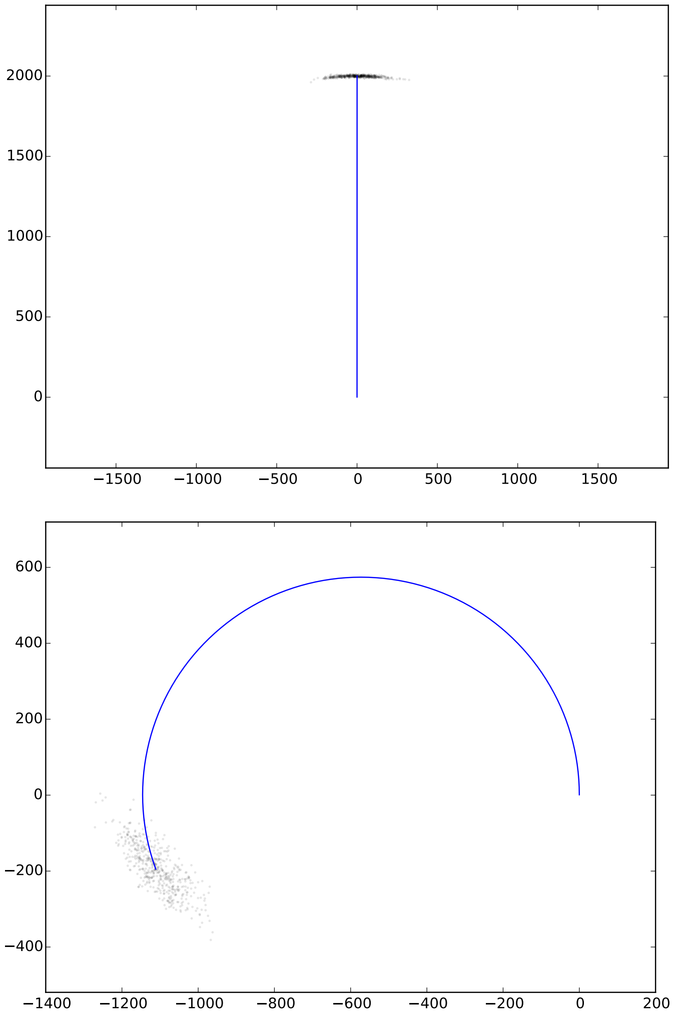


Figure 4.1: *Approximate final distribution of Monte Carlo samples after 2km straight trajectory (top) and 2km circular trajectory (bottom). Note the slight arc shape. For the straight trajectory, lateral deviation is the dominant factor, while for the circular trajectory lateral and longitudinal errors are more balanced.*

an extent self-correcting (as discussed in Section 2.5), which means a large error in an individual frame will typically result in another large (and hopefully compensating) error in the next frame, which will together have little impact on the shape of the posterior distribution. A bigger problem however, is the dependence on road context. The biggest influencing factors on the quality of the odometry estimate are often invisible to the method itself; for example feature pollution by local non-planarity of the road surface will induce a bias on the method which persists over multiple frames, but our attempts to detect this situation by correlating to an easily measurable quality (e.g. the variance among the features which form the consensus, the average lateral and longitudinal feature distance, the number of matching features or the number of features supporting the consensus) have failed. Slight correlations are observed between velocity (v and ω) and the average error, but this effect only enlarges the spread of the distribution and does not induce interdependence between consecutive errors like road context does. In summary, the short-term mean error is correlated with unknown (and unobserved) variables, a problem for which there appears no solution in statistics. These transient road contexts however have little effect on the long term odometry error, which is well represented by a normal distribution.

The Monte Carlo experiment described above, while useful for analysis, is impractical for real-time localization purposes due to the high number of samples necessary for a good approximation, and the computational burden of the kernel density estimation and the resampling. A parametric approximation of $f(\mathbf{x}_{world})$ is far more useful than a sampled distribution in this respect.

A frequently used tool for estimating the probability distribution of an unknown system state from uncertain measurements is a Kalman filter (KF). The Kalman filter is an algorithm that estimates the state of a linear system as well as the uncertainty associated with the estimate, by analysis of noisy measurements. In the case of a linear system with normally distributed noise, the Kalman filter estimates the exact conditional state probability distribution according to Bayesian inference [50].

The algorithm works in two steps: prediction and correction (or update). In the prediction step of each iteration, the next state is predicted using a transition model and the estimate of the previous step. Additionally, the uncertainty of the state (in the form an error covariance matrix) is propagated through this motion model to yield the uncertainty on the state prediction.

In the correction or update step, the prediction is compared to measurements and their associated uncertainty (i.e. the error covariance of the measurements). When the measurements agree reasonably well with the predicted state, the state is adjusted strongly towards the measurements and the uncertainty on the estimate is reduced. When the measurements fall in the fringes of the predicted uncertainty distribution of the state, the state adjustment is minimal and the uncertainty grows.

Mathematically a discrete time Kalman filter is described as follows. Let $\mathbf{x}_{k-1}$ be the real, unknown (hidden) system state, and $\hat{\mathbf{x}}_{k-1|k-1}$ be the state estimate after time step $k-1$, with $\mathbf{P}_{k-1|k-1}$ its associated error covariance estimate. The state at time step k is assumed to be evolved from the state at $k-1$ according to

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \tag{4.2}$$

where $\mathbf{F}$ is the state transition model, $\mathbf{B}$ is the control input model on a known input vector $\mathbf{u}_k$, and $\mathbf{w}_k$ is a process noise term assumed to be normally distributed with known covariance $\mathbf{Q}_k$. The observation vector $\mathbf{z}_k$ at time k is related to the true state by

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{4.3}$$

where $\mathbf{H}$ is the observation model and $\mathbf{v}_k$ the observation noise, normally distributed with known covariance $\mathbf{R}_k$. In the prediction step, a new state estimate $\hat{\mathbf{x}}_{k|k-1}$ is produced as

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_k \tag{4.4}$$

and its predicted estimate covariance as

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q}_k. \tag{4.5}$$

In the update step, first the measurement residual $\mathbf{y}_k$ is computed as

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \tag{4.6}$$

and the residual covariance

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k. \tag{4.7}$$

The Kalman gain, which is an indicator of how well the the measurements fit the prediction, is given by

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1} \tag{4.8}$$

and used to update the state estimate and covariance as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{y}_k, \tag{4.9}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1} \tag{4.10}$$

where $\mathbf{I}$ is the identity matrix.

For the application of visual odometry, the control input vector $\mathbf{u}_k$ is unknown (although measurements related to it may be available, which will be covered in Section 4.3) and the control term is omitted from the equations. The unknown inputs manifest themselves as additional process noise and are therefore modeled in the term $\mathbf{w}_k$ instead.

The standard Kalman filter assumes a linear motion model $\mathbf{F}$. In the case of a wheeled vehicle, the motion model is not linear, instead being described by the kinematic model of the non-holonomic robot with two controls and three degrees of freedom described in Section 2.3.3. The non-linear variant of the Kalman filter is called the extended Kalman filter (EKF) [66]. It applies to systems where the transition model $\mathbf{F}$, observation model $\mathbf{H}$ or both are not linear matrices but instead vector functions $\mathbf{f}(\mathbf{x}, \mathbf{u})$ and/or $\mathbf{h}(\mathbf{x})$ of the state on the condition that a local linearization of these functions is a good approximation within the given time step. Extended Kalman filters are considered the standard solution for road vehicle satellite navigation applications [100]. The prediction steps for an EKF are

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{\mathbf{k-1|k-1}}, \mathbf{u_k}) \tag{4.11}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F_{k-1}} \mathbf{P}_{k-1|k-1} \mathbf{F_{k-1}}^T + \mathbf{Q}_k. \tag{4.12}$$

where $\mathbf{F_{k-1}}$ is the matrix of partial derivatives (Jacobian) of $\mathbf{f}(\mathbf{x}, \mathbf{u})$ to the variables of the state vector, evaluated for the previous state estimate. For the update step, the residual is now computed as

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{\mathbf{k|k-1}}) \tag{4.13}$$

and $\mathbf{H}$ is replaced by the Jacobian matrix $\mathbf{H}_k$ of $\mathbf{h}(\mathbf{x})$ in Equations 4.7 through 4.10.

Let us now fill in the EKF model for our visual odometry application. The prediction of the next position depends not only on the current position x, y, but also on the current heading θ, longitudinal velocity v and turning rate ω, so the Kalman state vector will contain all these elements:

$$\mathbf{x} = [x, y, \theta, v, \omega]^T.$$

Our kinematic model states that the vehicle moves in a circular arc with a length determined by v and a final heading change determined by ω. In the coordinate system of the vehicle's start position, with the X axis parallel to the rear axle and Y axis pointing straight ahead, the displacement vector $(\Delta x_v, \Delta y_v)$ is given by

$$\Delta x_v = \frac{v(1 - \cos(\omega \Delta t))}{\omega},$$

$$\Delta y_v = \frac{v \sin(\omega \Delta t)}{\omega}. \tag{4.14}$$

Note that the case of zero angular velocity results in a division by zero and the implementation will need to consider the case of very small ω separately (by setting $\Delta x_v = 0$ and $\Delta y_v = v\Delta t$).

Transforming this into world coordinates using the current heading θ yields

$$\Delta x = \frac{v}{\omega}(\cos(\theta)(1 - \cos(\omega \Delta t)) - \sin(\theta)\sin(\omega \Delta t)),$$

$$\Delta y = \frac{v}{\omega}(\sin(\theta)(1-\cos(\omega\Delta t)) + \cos(\theta)\sin(\omega\Delta t)), \tag{4.15}$$

The state transition function is then defined as

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_k = \begin{bmatrix} x_{k-1} + \frac{v}{\omega}(\cos(\theta)(1-\cos(\omega\Delta t)) - \sin(\theta)\sin(\omega\Delta t)) \\ y_{k-1} + \frac{v}{\omega}(\sin(\theta)(1-\cos(\omega\Delta t)) + \cos(\theta)\sin(\omega\Delta t)) \\ \theta_{k-1} + \omega\Delta t \\ v_{k-1} \\ \omega_{k-1} \end{bmatrix} + \mathbf{w}_k. \tag{4.16}$$

The distribution of the noise term $\mathbf{w}_k$ needs to accommodate any deviations from the kinematic model, including side slip (which manifests as noise on x and y) as well as the unknown angular and longitudinal accelerations (represented by noise on $\mathbf{v}$ and ω).

The prediction step of the extended Kalman filter requires the evaluation of the Jacobian matrix of this transition function. It is given by

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 0 & \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial v} & \frac{\partial x}{\partial \omega} \\ 0 & 1 & \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial v} & \frac{\partial y}{\partial \omega} \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.17}$$

in which

$$\begin{aligned}
\frac{\partial x}{\partial \theta} &= \frac{v}{\omega}(-\sin(\theta)(1-\cos(\omega\Delta t)) - \cos(\theta)\sin(\omega\Delta t)), \\
\frac{\partial x}{\partial v} &= \frac{1}{\omega}(\cos(\theta)(1-\cos(\omega\Delta t)) - \sin(\theta)\sin(\omega\Delta t)), \\
\frac{\partial x}{\partial \omega} &= \frac{v\Delta t}{\omega}(\cos(\theta)\sin(\omega\Delta t) - \sin(\theta)\cos(\omega\Delta t)) \\
&\quad - \frac{v}{\omega^2}(\cos(\theta)(1-\cos(\omega\Delta t)) - \sin(\theta)\sin(\omega\Delta t)), \\
\frac{\partial y}{\partial \theta} &= \frac{v}{\omega}(\cos(\theta)(1-\cos(\omega\Delta t)) - \sin(\theta)\sin(\omega\Delta t)), \\
\frac{\partial y}{\partial v} &= \frac{1}{\omega}(\sin(\theta)(1-\cos(\omega\Delta t)) + \cos(\theta)\sin(\omega\Delta t)), \\
\frac{\partial y}{\partial \omega} &= \frac{v\Delta t}{\omega}(\sin(\theta)\sin(\omega\Delta t) + \cos(\theta)\cos(\omega\Delta t)) \\
&\quad - \frac{v}{\omega^2}(\sin(\theta)(1-\cos(\omega\Delta t)) + \cos(\theta)\sin(\omega\Delta t)).
\end{aligned}$$

The observation model in our case is linear, as we directly measure v and ω (and none of the other variables).

The EKF is used to compute a parametric posterior error distribution (a multi-dimensional normal distribution in the elements of $\mathbf{x}$ given the measurements for

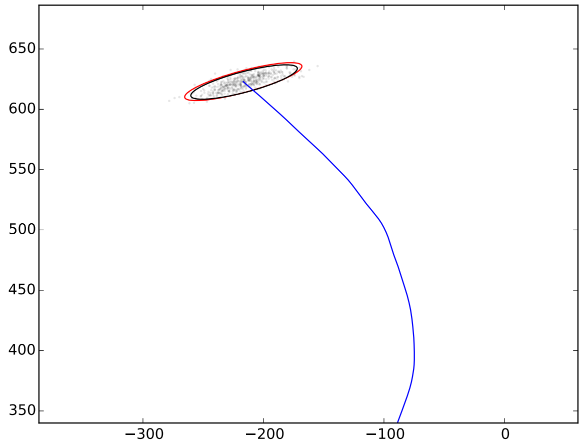


Figure 4.2: Comparison between Monte Carlo (black) and EKF (red) posterior distributions (with 90% confidence ellipses drawn) for a trajectory of the KITTI dataset. The EKF distribution is slightly wider as it takes into account a small amount of process noise, which our Monte Carlo model does not. The MC posterior has a slight boomerang shape which cannot be captured by the EKF, but the spread of the distributions is very similar.

v and ω and their error distributions). While this normal distribution cannot model the exact shape of the true posterior distribution, comparison with the posterior of the Monte Carlo method (which does capture the shape) shows that the differences are minor; for typical driving scenarios the true posterior distribution is well approximated by a multi-dimensional normal distribution (see Figure 4.2).

We may conclude that the EKF is an accurate tool to model medium and long term drift. As mentioned, care should be taken about short term accuracy on account of the interdependence of consecutive estimation errors as a consequence of the influence of road context.

4.3 Sensor fusion

The intent of this thesis is to provide a visual odometry solution which can provide positioning information when satellite navigation systems are unavailable or unreliable. However, even when satellite position is available, visual odometry can be a valuable addition. For example, consider the case of a stationary vehicle. Typically, this situation is hard to detect for a GPS system, as the error distribution on the positional fix means that the estimated position is continuously jumping, possibly over a range of several metres. Commercial GNSS applications attempt to remedy this by filtering the position, e.g. with a Kalman filter. This works well when the vehicle is moving, effectively hiding the uncertainty from the user by smoothing out the trajectory based on estimated velocity and heading, but in stationary situations the system is still prone to wandering in random directions. For the visual odometry system however, standing still is very easy to detect, and the wandering problem is completely absent. Many other scenarios exist where visual odometry can meaningfully augment the satellite positioning, for example on roundabouts with multiple closely spaced exits where an error of several metres is problematic, or on highway exits with filter lanes for specific destinations.

These examples clearly indicate the need for a sensor fusion approach, where visual odometry is combined with GNSS positioning, and any other readily available sensors in a road vehicle, such as a steering angle sensor (as used by the electronic stability control) and velocity signal (as used for the odometer, speedometer, anti-lock brakes, stability control and even the radio). Common sensor fusion approaches are Bayesian networks (also called *belief* networks), Kalman filtering or simpler approaches using the central limit theorem. Since we already use an extended Kalman filter to calculate the posterior distribution of our visual odometry, sensor fusion by Kalman filtering is a logical approach.

Within the EKF framework, integrating GPS measurements is very straightforward. Since the x and y position is part of the Kalman state vector, we can simply adjust the observation model H to include the GPS measurements on these variables. GNSS technology also provides a metric for the error covariance directly from the receiver, namely the *horizontal dilution of precision* (HDOP). This metric corresponds to the estimated variance of the position measurement projected onto the local geoid surface, and is a projection of the *geometric dilution of precision* (GDOP). GDOP is a measure of the sensitivity of the current positional fix to timing inaccuracies, which depends on the number of satellites received and their configuration in the sky relative to the receiver. If the azimuths and altitudes of all visible satellites lie closely together, the GDOP is high; if they are spaced wide apart the GDOP is low. It should be noted that GDOP is an optimistic estimate of the true uncertainty because it does not take into account atmospherical disturbances or multi-path effects (reflections). These effects are local, often transient and dependent on local road context, which makes it impractical to account for them in our model. The first two elements on the diagonal of the EKF measurement error covariance matrix are set to the HDOP value reported by the GPS sensor, as it is a fair estimate in the general case.

Similarly, the velocity and steering angle signals (which are broadcast on the CAN-bus of a modern vehicle) can be captured and added to the measurement vector. Some care should be taken considering scale. The relation between the steering angle sensor reading and the angular velocity ω of the vehicle depends on the vehicle. In most cases the reading is linearly related, but nonlinear steering racks are sometimes found in performance oriented models. Whether the steering signal on the CAN-bus is corrected or not is up to the manufacturer. In case of a nonlinear relationship between the CAN-bus signal and the angular velocity ω, the observation model $\mathbf{H}$ in Equations 4.7 through 4.10 needs to be replaced with an appropriate function vector, and the Jacobian matrix of this function serves as a local linearization.

A magnetic compass is also sometimes found in a car's navigation equipment. This is another direct observation, of the state variable θ (heading). If any of these sensors are biased, calibration will be necessary as the EKF has no provisions to correct for this.

In future work, inertial sensors (accelerometers and gyroscopes) could also be added to the system. Since these devices measure parameters which are not currently in the state vector, their inclusion is less straightforward. A solution could be to extend the EKF model to a higher order, where angular and longitudinal acceleration are added to the state vector. Alternatively, the output of the inertial sensors can be integrated to yield measurements of v and ω, but this requires handling the error distribution of the sensors in some way which may closely resemble the Kalman filter covariance estimation anyway. Inertial sensors will not be discussed further in this work; however the integration with steering angle and wheel speed sensors will be briefly demonstrated in Section 4.6.

4.4 Map localization

In most intelligent vehicle applications, the vehicle's ego-position is not the only source of information. It is the relative position of the vehicle within the local environment which is of primary importance. This section will explain how the position estimate and its associated uncertainty can be translated into a map position. OpenStreetMap (OSM) is used as the example map source; other map sources are very similar in format.

4.4.1 Immediate Localization

4.4.1.1 Unoriented Localization

The EKF provides an estimate of the current vehicle state and its associated uncertainty in the form of the estimated posterior error covariance, using immediate measurements. For road vehicle applications however, these immediate measurements are usually supplemented by prior information about the road network. While the road network is not considered a sensor, it can be an important source of information because it imposes constraints on the domain of possible vehicle states. One way to formulate such constraints could be to impose a maximum distance between the x, y coordinates of the vehicle to the nearest road segment. While this has the potential to strongly limit the spread of the posterior error distribution of the vehicle state, it also has the downside that the shape of the distribution will become very hard to describe parametrically when multiple road segments come together under the EKF error distribution.

Because most applications ultimately require the estimated position to be projected onto the road network, we may look at the problem from this perspective: which road segments are most likely given the EKF vehicle state and error distribution? In order to answer this question, we could integrate the EKF posterior distribution (limited to the first two dimensions) over the area of the road segment. This approach has several drawbacks. Firstly, calculating the double integral of a non-aligned bivariate normal distribution is not a straightforward problem, although good approximation strategies exist in many cases [14]. Secondly, the width of the road segment is ill-defined. Road map metadata rarely includes such information, and even if it does, the relative width of two nearby roads should not greatly influence their relative likelihoods (although one could argue that wider roads are generally constructed that way because they may carry more traffic and are therefore a priori more likely).

Rather than integrating the posterior likelihood over the physical area of the road, we need to define a comparison metric for the relative likelihoods of the *paths* represented by the road segments. The line integral of the posterior distribution along a map segment can serve as such a likelihood metric. The next paragraphs will explain how this line integral is calculated.

Using the notation from Bishop [12], an n-dimensional multivariate normal distribution centered around $\mathbf{x}_k$ with covariance matrix $\mathbf{\Sigma}_k$ is given by:

$$\mathcal{N}(\mathbf{x}|\mathbf{x_k}, \mathbf{\Sigma_k}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}_k|}} \exp(-0.5(\mathbf{x} - \mathbf{x}_k)^T \mathbf{\Sigma}^{-1}(\mathbf{x} - \mathbf{x}_k)) \qquad (4.18)$$

in which $|\mathbf{\Sigma}_k|$ is the determinant of $\mathbf{\Sigma}_k$. As the map segments do not impose constraints directly on the velocity v or angular velocity ω, we will discard these dimensions. Likewise the heading dimension θ is not strictly constrained due to the possibility of making maneuvers within the road (e.g. driving around parked cars). Thus we will initially consider only the marginal posterior distribution of x and y to be integrated along the map segments. The influence of orientation will be further discussed in Section 4.4.1.2. With S denoting a line segment, our likelihood estimator for a segment given the EKF posterior $\mathcal{N}(\mathbf{x}|\mathbf{x_k}, \mathbf{\Sigma_k})$ is then defined as

$$L(S, \mathbf{x_k}, \mathbf{\Sigma_k}) = \int_S \mathcal{N}(\mathbf{x}|\mathbf{x_k}, \mathbf{\Sigma_k}) ds. \qquad (4.19)$$

Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be the start and end point of the line segment. The parametrization of the line segment is then given by

$$\mathbf{x}(u) = (x(u), y(u)) = \mathbf{x}_1 + u(\mathbf{x}_2 - \mathbf{x}_1), \quad u \in [0, 1]. \qquad (4.20)$$

Substituting this parametrization in Equation 4.19 yields

$$L(S, \mathbf{x_k}, \mathbf{\Sigma_k}) = \int_0^1 \mathcal{N}(\mathbf{x}(u)|\mathbf{x_k}, \mathbf{\Sigma_k}) \sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2} du. \tag{4.21}$$

Based on Equations 4.18 and 4.20 this expands to

$$\begin{aligned} \int_S \mathcal{N}(\mathbf{x}|\mathbf{x_k}, \mathbf{\Sigma_k}) ds = \int_0^1 \Bigg[& \frac{1}{\sqrt{(2\pi)^2|\mathbf{\Sigma}_k|}} \\ & \exp\big(-0.5(\mathbf{x}_1 - \mathbf{x}_k + u(\mathbf{x}_2 - \mathbf{x}_1))\mathbf{\Sigma}^{-1}(\mathbf{x}_1 - \mathbf{x}_k + u(\mathbf{x}_2 - \mathbf{x}_1))\big) \\ & \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} du \Bigg]. \end{aligned} \tag{4.22}$$

This integral is of the form

$$\int_0^1 Ce^{-au^2-bu-c} du$$

which can be evaluated using the error function as

$$\int_0^1 Ce^{-au^2-bu-c} du = \frac{C}{2\sqrt{a}} e^{\frac{b^2}{4a}-c} \sqrt{\pi}\, \mathrm{erf}(\frac{b+2a}{2\sqrt{a}}).$$

Evaluating this line integral along each map segment allows us to compare the relative cumulative likelihoods of the segments under the EKF posterior probability distribution. We then use the maximum likelihood principle to find the map segment which best corresponds to each consecutive EKF state estimation.

Some care should be taken regarding the accuracy of the map segments as a representation of the real road. The map segments are only a polygonal approximation of the road layout. As a consequence, the map segments can deviate from the actual road centerline by up to a few meters, especially near the endpoints of the segment. If the EKF posterior distribution is narrow, this can give rise to significantly underestimated likelihoods. It is unlikely that this will cause errors in the ordering of relative likelihoods of segments however, as in such cases of compact EKF distribution it is probable that either only one segment will have a nonzero likelihood, or two connected segments will be affected equally. In general, for the application of visual odometry the standard deviation of the EKF posterior will quickly grow larger than the maximum map segment deviation.

4.4.1.2 Oriented Localization

In the above discourse, we have ignored the heading θ because the map does not strictly constrain this dimension. However, orientation can still be a powerful clue to the most likely road segment, as road vehicles will *generally* follow the orientation of the map segment. In order to add this orientation selectivity to the map localization process, we will multiply the likelihood function with a factor which takes into account the agreement of the vehicle heading estimated by the EKF and the direction of the map segment.

When comparing vehicle heading to map segment orientation, two considerations need to be made. Firstly, cars can drive around obstacles on the road, and this can cause significant differences between vehicle heading and road direction. Secondly, as mentioned earlier, the map segments are only a polygonal approximation of the road, and the local orientation of the road may therefore differ slightly from its mapped version.

In light of these considerations it makes sense to model the expected driving direction on a road segment as a (one-dimensional) normal distribution. In order to determine the likelihood of a segment given the EKF heading estimate of the vehicle, we can determine the overlap between the normal distribution of driving direction on the map segment and the normal distribution defined by the EKF heading and its estimated variance. Note that this approach ignores the correlation between xy-position and orientation; however it does provide an elegant mechanism for orientation selectivity under the given limitations of map accuracy. We can multiply the segment xy-likelihood (based on the line integral through the first two dimensions of the EKF posterior) and its orientation likelihood (based on the third dimension of the EKF posterior) to obtain a likelihood metric which accounts for both position and orientation of the segment. The resulting likelihoods yield much better results than the likelihood using only x, y. An example showing the merit of incorporating orientation in the likelihood function is shown in Figure 4.3.

An important consideration is the standard deviation of the probability distribution of the expected heading for a map segment. While the distribution is generally very narrow (people spend much more time driving straight along a road than maneuvering inside it), such narrow distributions cause problems at junctions. At a junction, two map segments may join at any angle, giving rise to large discontinuities in the expected vehicle heading according to those two segments. As the vehicle will smoothly transition between one segment and the other, both segments will at some point have a very low likelihood. Figure 4.4 shows a situation in which narrow heading probability distributions cause the wrong segment to attain maximum likelihood. The solution is to either deal with junctions separately, or widen the probability distributions. For reasons of computational complexity, we have chosen the latter approach.

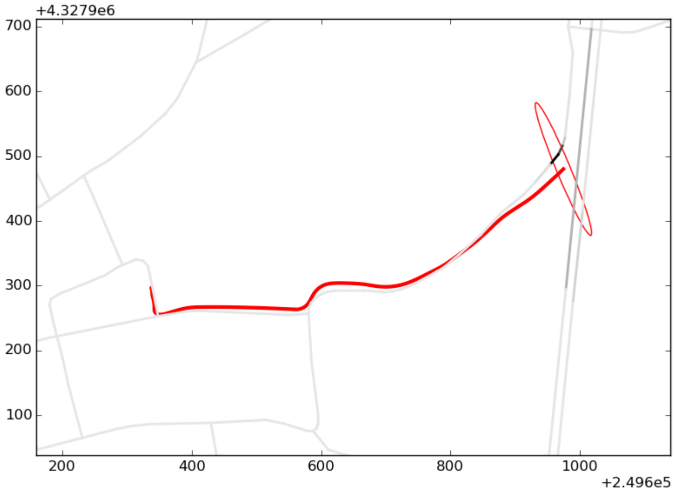


Figure 4.3: *Illustration of the importance of orientation in the segment likelihood function. More likely segments are drawn in a darker grey. The short segment in the bend is approximately at the same distance as the long highway segment, but is more likely due to its orientation.*

4.4.2 Sequential Localization

It is clear from the example in Figure 4.3 that taking into account the full trajectory up to this moment rather than just the current position estimate can further improve mapping accuracy. When we take into account the path that would have to be followed to reach each of the candidate segments, and the likelihood of that path considering the past observations, much better differentiation is possible between segments which are close to each other but only connected through the outskirts of the EKF uncertainty distribution. Such situations are very common around highways, where access roads can follow the highway for miles before reaching an onramp.

Given the discrete nature of the map data and the indirect observations in the form of odometry, a suitable approach for modelling the sequential transitions between map segments is the hidden Markov model (HMM). The suitability of HMMs for solving map matching problems is well documented[37, 71, 82, 83, 93]. A hidden Markov model is a form of Bayesian network which describes a system which transgresses through sequential states, but in which the states are not directly observed (hence *hidden*); instead the likelihood of the states is estimated from output variables which depend on the state. The output variables can be discrete (in which case a probability mass function for the output variables must be defined on

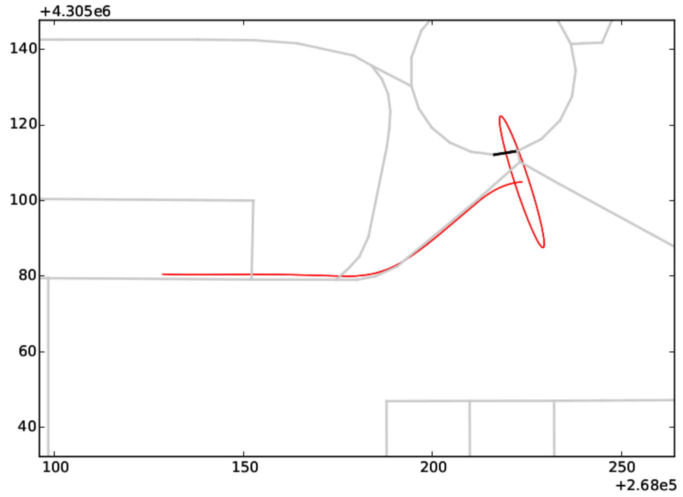


Figure 4.4: *Situation in which narrow probability distributions on the expected heading for each segment cause the wrong segment to have maximum likelihood. The cause is the discontinuity between expected headings at the junction just south of the roundabout.*

each state), or continuous (in which case a probability density function is required). The probability mass or density functions are called *emission* distributions of the states. In addition to the emission distributions, the *transition* distribution must also be known for each state. The transition distribution indicates how likely the system is to go from one state to the other in each time step. The system can be represented by a directed graph; the likelihoods of all edges originating from one state must be known and sum to one (including the self-loop). The system must satisfy the Markov property, i.e. it must be *memoryless*. This means that given the current state, it is irrelevant for the future how this state was reached. More formally, the conditional probability distribution of future states depends only upon the present state. An example of a hidden Markov model in graph representation is shown in Figure 4.5.

It is obvious that our odometry problem satisfies the Markov property: future vehicle states only depend on the current vehicle states, and not on the way in which the current state was reached. The vehicle state is not directly observed, but we have information about the probability distribution of the vehicle states through the EKF modeling of the odometry observations. The EKF posterior distribution can therefore serve as the emission probability distribution in the HMM.

When considering specifically the problem of relating the odometry to the polygonal representation of the road network as given by the map data, it makes

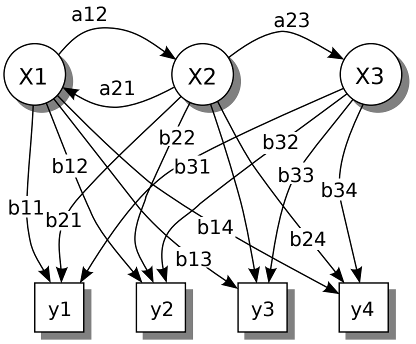


Figure 4.5: *Example of a hidden Markov Model. X_i are the state nodes, y_i the output parameters (in this case discrete), a_{ij} the transition probabilities and b_{ij} the emission probabilities.*

sense to equate the nodes of the Markov graph to the map segments. The edges of the graph indicate the connectedness of the segments. An important consideration in this regard is the definition of connectedness. Depending on the speed of the vehicle, the granularity of the map (i.e. the minimum segment length) and the time step used for the HMM modeling, it may be insufficient to consider only segments which are directly connected through one of their endpoints. Instead it may be necessary to consider all segments which are connected through a path shorter than a threshold length, which may incorporate intermediate segments. An example is given in Figure 4.6. When building the HMM graph dynamically, this threshold length can take into account the current velocity estimate and the maximum acceleration allowed by the vehicle model.

With map segments as HMM states, the emission probability of an odometry measurement is given by the segment likelihood as described in Section 4.4.1. Indeed the goal of this section was to get a closed form solution for the probability that an odometry state originates from any position within a single map segment, so that likelihoods of the vehicle occupying different segments can be compared.

One aspect of the HMM which is not immediately clear is the transition probability between states. The probability of staying on a segment is related to the length of that segment and the speed of the vehicle. However, this dependency on spatial extent and vehicle speed is already modeled in the EKF posterior distribution. Including this data into the HMM prior would enforce an unwanted bias towards longer segments, resulting in “stiction” near the endpoints of such a long segment. M. Brubaker and Urtasun [63] use a Gaussian mixture model

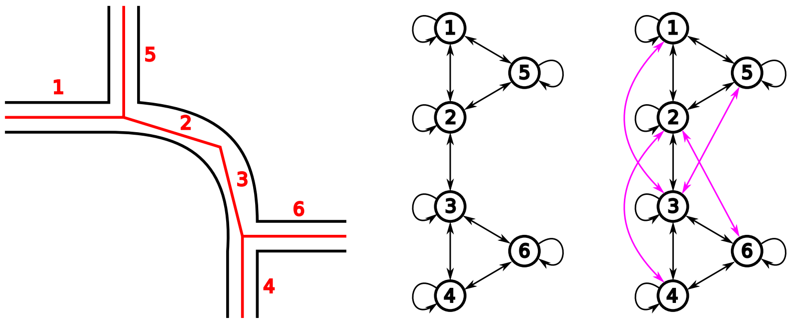


Figure 4.6: Example of road network (left, outline), its mapped representation (left, numbered red segments), the segment connection graph (center) and the HMM connection graph allowing to skip segments based on vehicle speed (right).

to compute transition probabilities, allowing for a different transition probability near the endpoints of a segment than in the middle. However, in our opinion this is largely redundant as it closely mirrors the evolution of segment likelihood along the length of the segment as defined in Section 4.4.1. Instead the transition probabilities from one segment to all connected segments (including itself) will be chosen equal, as the transition probability depends on the projection of the EKF state onto the segment, a property which is not modeled by the HMM. Setting all transition probabilities equal means that the HMM turns into a more specific kind of discrete state-space model and some parts of the calculations become obsolete, however we will continue to use the terminology and mathematics of the standard HMM as they are well-known and well-described in literature, and apply to the equal-transition variant without reservation.

With the mapping problem translated to the HMM, we can now use the forward algorithm to obtain the most likely current map segment [12]. The forward algorithm calculates the belief state: the probability of a state given the history of evidence (output parameters). Formally, the probability of state x at time t considering the sequence of outputs $y_1 ... y_t$ is written as $p(x_t|y_{1:t})$. Note that x now indicates the Markov state, and is distinct from the state vector $\mathbf{x}$ of the EKF discussed earlier, written in bold. Given the Markov property, it is easily seen that the probability of x_t is conditional only on x_{t-1} and can be recursively determined as

$$p(x_t|y_{1:t}) = p(y_t|x_t) \sum_{x_{t-1}} p(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1}). \tag{4.23}$$

At each time, the probability of all states is computed through this equation using the previous state probabilities $p(x_{t-1}|y_{1:t-1})$, the transition probabilities $p(x_t|x_{t-1})$ and the emission probabilities $p(y_t|x_t)$.

The forward algorithm is closely related to the Viterbi algorithm, with the difference that Viterbi also determines the most likely sequence of states leading to each state (it records the state history) [12]. The probability of the most likely sequence of states leading to x_t is denoted as $p(x_{1:t}|y_{1:t})$, and it it recursively determined as

$$p(x_{1:t}|y_{1:t}) = \max_{x_{t-1}} \left(p(y_t|x_t)p(x_t|x_{t-1})p(x_{1:t-1}|y_{1:t-1})\right). \qquad (4.24)$$

The main difference with the forward algorithm is that instead of summing the probabilities of each path leading up to the current state, only the probability of the most likely path is propagated. We will use the Viterbi algorithm as the history it provides can be useful for related applications (e.g. automatic map annotation).

In our visual odometry framework, the HMM is dynamically constructed as follows. At time step 0, the initial EKF state vector and covariance are used to build a starting graph containing nodes for all map segments within five standard deviations of the start state. At each next time step, the current velocity estimate (which is part of the EKF posterior distribution) is used to determine a *horizon* corresponding to the maximum travel distance by the next time step (typically chosen as twice the current velocity estimate). Any map segments which can be reached from any of the segments currently in the graph through a path shorter than this maximum travel distance fall within the horizon and are added to the graph. Transition probabilities for all outgoing edges of each node are set equal (in practice the implementation simply omits them). The Viterbi algorithm is then used to update the state probabilities using the emission probabilities calculated on the EKF (i.e. the segment likelihood metric). The graph is then pruned: nodes with posterior probability below a threshold are removed (typically, the threshold value is set conservatively as 10^{-40} smaller than the maximum state probability in the graph).

For this implementation, it is advantageous to convert the OpenStreetMap data to a format which allows for quicker construction of the HMM graph. The OSM data is available in multiple formats offering various levels of compression, but regardless of the format, it can be thought of as a collection of nodes and a collection of relations specified between these nodes. The nodes represent world points specified by latitude and longitude, and each node has a unique identifying number (ID). An examples of a relation is a way, which is specified as a list of node IDs (with additional metadata indicating the type of road, name of the road etc.). Relations between other relations are also possible (e.g. routes defined as lists of ways).

The split structure between nodes and ways is cumbersome for our application as it increases the number of lookup operations required to extract the properties of road segments required for the HMM (e.g. segment orientation). Therefore we will parse the OSM data in pre-processing and convert it into an array of objects representing segments, with each object having the following fields:

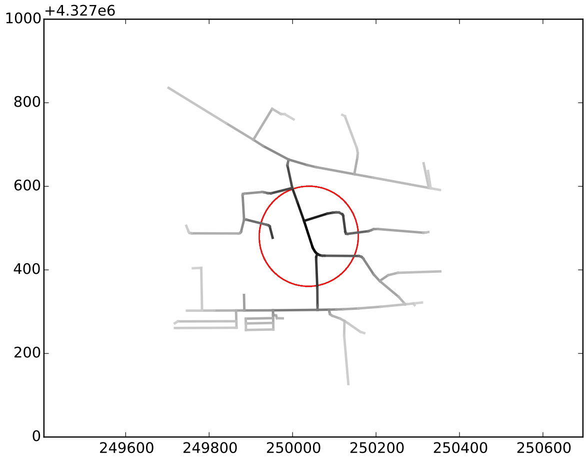


Figure 4.7: Initial plot of the HMM segment likelihoods (top), with starting sigma of 30 meters in x and y, and 30 degrees on heading. Darkness of the line is proportional to logarithm of segment likelihood.

- UTM coordinates (easting and northing in meters) of the start- and endpoint,
- angle of the segment (in $]-\pi, \pi]$ with the east-west direction representing zero angle),
- length of the segment in meters,
- vector of indices of neighboring segments.

This format allows for quick dynamic construction of the HMM graph and avoids redundant computation of angles and lengths. The pre-processing is performed on a 50 by 50 km subregion of the 258MB `belgium.osm` file, but the size of the subregion can be tuned to fit the memory constraints of the platform. The merging of adjacent subregions when the vehicle comes near the boundaries can be performed at relatively low computational cost, by constructing (also in pre-processing) a list of connections across subregion boundaries. Only the vector of neighbors of such crossing segments must be updated in order to add or remove a subregion from the active set. The adding and removing of subregions is not currently implemented however, as the memory requirements are low in any case.

Figures 4.7 through 4.9 show some examples of local map localization through the Viterbi algorithm after various distances driven. It is clear from our experiments that the proposed method can correctly identify the most likely current

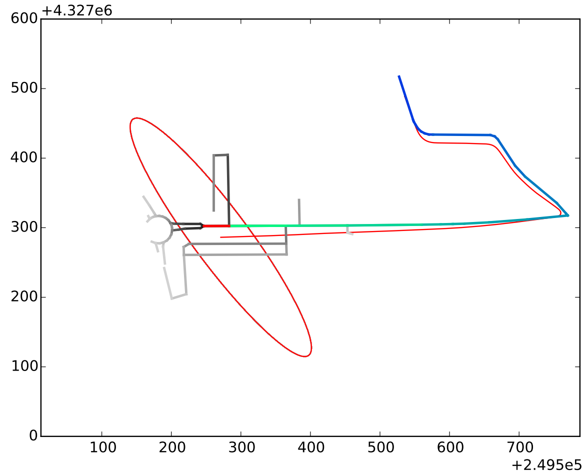


Figure 4.8: *Result of Viterbi algorithm after 800m. Current most likely segment is plotted in red, and most likely path leading up to it in green (recent) to blue (older). Note that even though the parallel lane below the main road is closer to the center of the distribution, it is less likely because the path leading up to it is unlikely.*

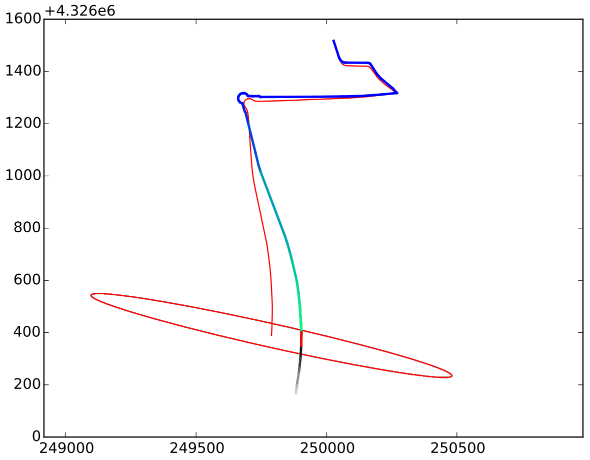


Figure 4.9: *Result of Viterbi algorithm after 2000m.*

map segment in the short and medium term, but as the EKF posterior distribution grows, the discriminative power of the method decreases and the sensitivity of the localization to noise or errors in the odometry increases.

To recap this section, we have defined a method for determining the likelihood that the vehicle is on each map segment given an uncertain starting position and a series of uncertain odometry measurements. The likelihood function incorporates current estimated distance to the map segment, current estimated orientation relative to that of the map segment, and the likelihood of the path to each segment based on earlier estimations. It does not, however, solve the problem of drift, as there is no feedback from the road network back to the state vector. As a consequence, the HMM by itself is unable to correct for medium and long term accumulation of estimation errors; once the EKF posterior distribution spans hundreds of segments, the map matching degrades to the point of being useless. This will be addressed in the next section.

4.5 Map feedback

In order to solve the long term drift issue, we wish to have the road network as an active information source, which influences the EKF state. Such a feedback mechanism makes sense; after all it is very unlikely that the vehicle is driving anywhere but on the road, and the Kalman filter should take this into account by “sticking” to the vicinity of road segments. Section 4.5.1 will describe how we will define a prior probability distribution based on the road network. Section 4.5.2 will explain how this prior interacts with the framework of EKF and HMM we proposed above.

4.5.1 The map prior

The road network can be thought of as a prior distribution of position likelihood. Vehicle states which fall close to a road segment (in position and orientation) are far more likely than positions further away from any roads, or vehicle headings which deviate significantly from the nearby road segment orientations. However, this distribution does not have sharply defined limits; vehicles can make maneuvers on the road, or even in the areas directly adjacent to the road. Additionally, not all private roads may be listed, or the map information may be outdated. However, it is still a fair assumption, if not a hard requirement, that the great majority of the vehicle’s trajectory will follow the road network as described by OpenStreetMap or comparable map sources.

Ideally, we would want to weigh the Kalman posterior distribution with a likelihood function based on distance (in both position and orientation) to the entire local road network. However, such a likelihood function would cause the posterior distribution to have multiple modes, which requires the extended Kalman filter to be adapted, the most common approach for which is a Gaussian mixture model as in Quinlan and Middleton [81] and M. Brubaker and Urtasun [63]. However, instead instead of modelling the entire road network in the posterior distribution, we may approach the map prior from a more local point of view: each state in the hidden Markov model corresponds to the hypothesis of being on one map segment, and we may define a prior distribution on single map segments instead of a distribution for all local roads. In such a localized prior, we wish to incorporate the following assumptions:

- the vehicle heading should be more or less aligned with the segment,
- the vehicle should not be too far from the center line of the map segment,
- the projection of the vehicle's location onto the center line should fall between the endpoints of the segment.

The first two elements are well described by a normal distribution: using ground truth (e.g. from the KITTI database) we can easily determine reasonable standard deviations for heading mismatch and lateral deviation of the vehicle with relation to the road. The third element is more difficult: ideally we want a longitudinal distribution which falls off sharply at the endpoints of the segment, but which is uniform within the segment; any point along the length of the segment should be equally likely. Such a distribution is not described by a single covariance matrix, and therefore it is unclear how its interaction with the Kalman filter should be defined. Again, a Gaussian mixture model with many terms can be used to approximate this behavior but this carries with it a significant increase in computational cost. Instead we will relax the third assumption and define a normal longitudinal distribution of sufficient elongation to be quasi-constant within the endpoints of the segment; in other words the segment's prior distribution will limit lateral position and orientation, but have little influence on longitudinal position. The segment prior is now described by a three-dimensional normal distribution and its application is well described by Kalman filter theory.

From analysis of the KITTI dataset ground truth, representative standard deviations of five meters laterally and eight degrees in orientation are chosen. These values should probably be optimized in future work; they depend to a significant extent on the type of road; on highways there is more lateral deviation but much smaller orientation deviation than on a busy shopping street with 30 kmh speed limit. Longitudinal standard deviation is typically chosen as five times the length of the segment, which causes the longitudinal dimension to be largely ignored.

We have now introduced the road network prior as a collection of 3D normal distributions each defined on a single segment, which we will use to influence the EKF posterior distribution.

4.5.2 Influence of the prior on the HMM

In Section 4.4.2 we have explained how the hidden Markov model is used to calculate the probability of being on each map segment, either using the forward algorithm (which accumulates the probability of all possible paths to that segment), or the Viterbi algorithm (which calculates the probability of the most likely path to that segment only). Since our objective is to have the segment prior influence the posterior distribution, the Viterbi algorithm is clearly preferred. The estimate covariance should take into account the path leading up to the current state, and each hypothesis (i.e. each HMM state with nonzero probability) should have its own set of corrections on the Kalman state corresponding to the most likely path to that state, independent of the other HMM states.

To illustrate the importance of the independence of the Kalman states corresponding to different hypothesis, consider the example of a road fork where one road splits into two roads which start out parallel but then gradually lead apart. One of the roads has a right turn after one kilometer, the other does not. Both hypotheses should be entertained, and each hypothesis should maintain a Kalman posterior distribution influenced by its own road prior, instead of sharing a single posterior which would necessarily have a wide spread. When the vehicle then takes a right turn, one hypothesis is discarded and the narrow EKF posterior belonging to the correct road is propagated to the next section.

In practice this means each node in the graph will be linked to its own Kalman state. The Viterbi algorithm is used to compute the most likely path to each segment, and to inherit the Kalman state associated with the last state of this path.

In Section 4.5.1 we have split the road network prior into a collection of 3D normal distributions each defined on a single segment. Since each HMM node corresponds to the hypothesis of being on one segment, we will use only the prior of the segment corresponding to that node to influence its Kalman state as follows.

If we consider the road segment prior an artificial observation, Equations 4.6 to 4.10 from Section 4.2 describe how this observation should affect the estimated state and covariance through calculation of the residual and Kalman gain. Using the map prior as an observation, while convenient for implementation, raises two questions. What is the frequency of this artificial measurement? How can we avoid violating the Kalman filter assumption of conditionally independent measurements even though our map prior is constant? We will address these concerns in two ways. Firstly, we will trigger the map prior update by distance driven rather than time. This avoids the map prior becoming dominant when the vehicle is stationary or driving slowly, and by choosing a high enough value for this distance trigger we can mitigate the underestimation of the covariance caused by the dependence of consecutive observations. Secondly, we will perform *fractional* updates to the Kalman model by weighing the Kalman gain with a factor (e.g. 0.1 to split the update over 10 steps). The combination of these two mechanisms will allow us to tune the relative importance of the map prior (e.g. one full update per 100 meter driven) and implement it in many small updates rather than a few big “jumps” to keep the trajectory smooth. The next paragraphs describe this process mathematically.

Let $\mathbf{m}$ be the vector containing the xy-coordinates of the midpoint of the segment and its orientation, $\hat{\mathbf{x}}_{k|k}$ the posterior EKF distribution at time k, and $\mathbf{P}_{k|k}$ its covariance, then the *mapping residual* $\mathbf{n}_k$ for that segment is given by

$$\mathbf{n}_k = \mathbf{m} - \mathbf{H}\hat{\mathbf{x}}_{k|k} \tag{4.25}$$

with $\mathbf{H}$ the corresponding observation model (a 5x5 matrix of zeros except for the upper three diagonal elements, which are 1), and the residual covariance is

$$\mathbf{T}_k = \mathbf{H}\mathbf{P}_{k|k}\mathbf{H}^T + \mathbf{U} \tag{4.26}$$

with $\mathbf{U}$ the covariance of the map segment distribution constructed as explained in Section 4.5.1. The Kalman gain is still calculated as

$$\mathbf{K}_k = \mathbf{P}_{k|k}\mathbf{H}^T\mathbf{T}_k^{-1} \tag{4.27}$$

but it is now scaled by a factor w_k proportional to the travel distance between consecutive time steps:

$$w_k = \frac{1}{w_{full}}\sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}.$$

This factor scales the update of the state estimate and covariance:

$$\begin{aligned} \hat{\mathbf{x}}'_{k|k} &= \hat{\mathbf{x}}_{k|k} + w_k\mathbf{K}_k\mathbf{n}_k, \\ \mathbf{P}'_{k|k} &= (\mathbf{I} - Wd_k\mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k}. \end{aligned} \tag{4.28}$$

The constant w_{full} is the distance the vehicle must drive for the map prior to have the weight of a “full” update consistent with a single observation. Typical values for w_{full} are between 10 and 100 (meters).

To summarize, we implemented map influence by means of a separate extended Kalman filter for each node in the hidden Markov model, on which a weighted update is performed consistent with a local road prior constructed as a normal distribution around the segment. In this way, the Kalman posterior distributions are influenced by the map segment hypothesis corresponding to the node, which will reduce the posterior covariance and thus limit positional drift.

One may remark that the inheritance of the Kalman state violates the Markov property; the conditional probability of future states does depend to some extent on the sequence of past states. Specifically, the emission probabilities of future states depend on the current Kalman posterior, which is influenced by the map priors of

Total distance	100.2 km
Total time	114 min
Average speed	52.9 km/h
Slow sections (<30km/h)	5.8 km
Medium speed sections (30-80km/h)	72.0 km
Fast sections (>80km/h)	22.4 km
City/town sections	19.9 km
Rural sections	36.3 km
Highway sections	44.0 km

Table 4.1: *Properties of the evaluation dataset.*

the segments leading up to the current state. In the context of visual odometry however, different paths to the *same* state typically constitute only slightly earlier or later transitions from one segment to the next, and the difference between their Kalman states will be small compared to the difference between the Kalman states of different routes (cfr. the road fork example given earlier).

Figure 4.10 shows the proposed mechanism in action. It is clear that the map feedback effectively eliminates drift; the posterior distributions with map feedback do not grow unbounded as is the case without feedback.

4.6 Results

In this section, the map localisation algorithm will be thoroughly evaluated. The main questions we want to answer are the following.

- How does our mapped visual odometry solution compare to standard GNSS positioning?
- To what extent can sensor fusion between both systems improve on accuracy?
- What are the computational costs of our solution?

4.6.1 Dataset

In order to answer these questions, the visual odometry solution is tested on 100km of test video. This dataset contains city, town, highway and rural driving. Table 4.1 lists the most important properties of the dataset.

The sensor setup for these recordings was as follows:

- GoPro Hero 4 Black camera capturing in 1280×720 resolution (later downsampled to 640×360) at 30 frames per second,

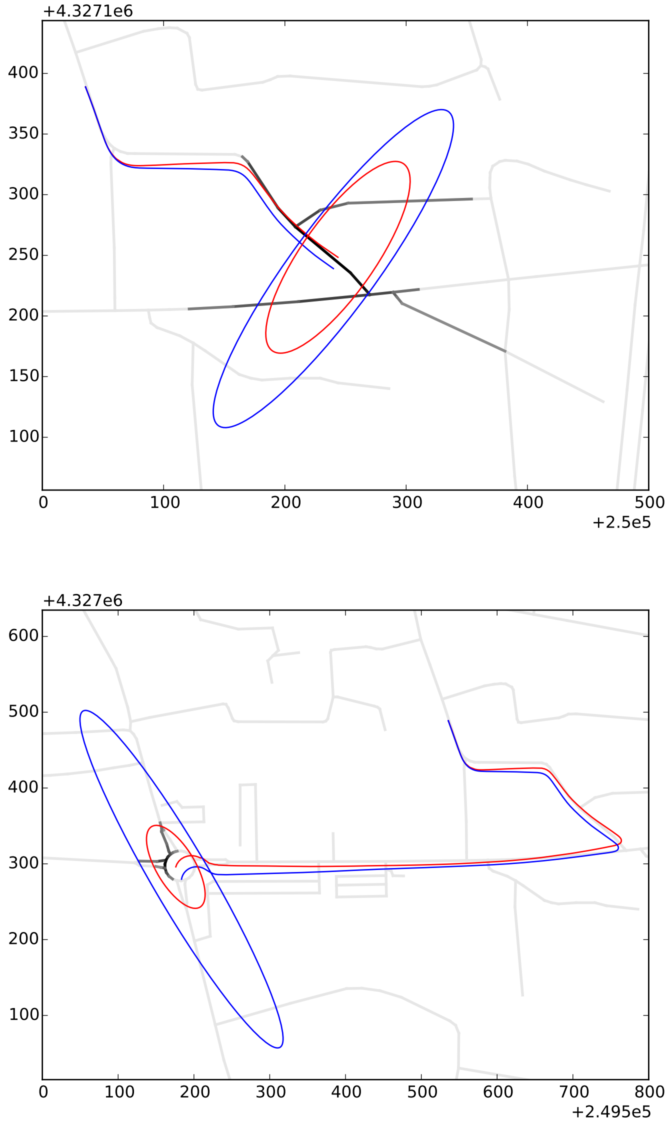


Figure 4.10: *HMM with map feedback. 90% confidence ellipse of the EKF posterior of the most likely node without map feedback is plotted in blue, with feedback in red. Likewise the uncorrected and corrected trajectory so far. Road network is drawn in light gray for reference. Map feedback pulls the position towards the road network and limits the positional uncertainty in the lateral direction.*

Corner detector	Harris
Corners	48
ROI width	6m
ROI length	15m
Undistortion	5th degree
HMM prune threshold	10^{-10}
Map feedback every	10m

Table 4.2: *Configuration parameters for evaluation dataset.*

- OBD2/EOBD data logger capturing vehicle velocity and individual wheel velocities at 10Hz, with 1km/h accuracy,
- GPS+GLONASS satellite positioning at 10Hz through Motorola Moto G smartphone.

Sadly, steering angle measurements were not possible with the EOBD (Extended On-Board Diagnostics) protocol and the test vehicle. Instead, we calculated approximate steering angles from the difference in speed measured at the left and right wheels. The accuracy of these estimated steering angles depends on the turn radius and the speed of the vehicle.

OpenStreetMap was used as map source. Regions of interest spanning approximately 20 by 20 kilometers were extracted from the `belgium.osm` main map to reduce memory requirements.

4.6.2 Algorithm parameters

The configuration parameters for the complete visual odometry, sensor fusion and mapping algorithm are listed in Table 4.2.

Intrinsic calibration was performed as described in Chapter 3 with 60 manually verified calibration checkerboard frames. Extrinsic calibration was performed using the iterative single angle optimization, followed by manual refinement on a 2.8km test loop.

4.6.3 Implementation

The entire framework is implemented in a simulated real-time environment, using C++, OpenCV and the `pthread` library for multi-threading. The extended Kalman filter was implemented using the `Eigen` template library for linear algebra. The main program thread handles the HMM (with an extended Kalman filter in each node) and incorporates the offline map. The map is converted into a dictionary of map segments and a dictionary of segment connections in an offline

	avg. time	max. time	max. framerate
main thread	301	1019	981
VO thread	5920	13785	72

Table 4.3: *Processing time of main and visual odometry threads, in microseconds.*

preprocessing step. This allows for fast extension and computation of the HMM graph without having to parse large map files.

A first auxiliary thread serves video frames to a buffer based on their recording timestamps. A second thread computes visual odometry from the most recent frame in the buffer and outputs odometry measurements to the main thread, then waits for a new video frame if necessary. Another auxiliary thread reads the GNSS and OBD2/EOBD data from a file and sends updates to the main thread at the rate of their recording timestamps. In this manner, the threads simulate the sensors and devices as they would interact in a real-time system. This implementation also allows to speed up the computation in the evaluation experiments: the threads can be configured to run at a multiple of real-time speed, utilizing processor headroom without compromising the validity of the real-time simulation. The evaluations in this section were performed at 2.5 times real-time speed.

4.6.4 Computational requirements

The experiments were performed on an Intel Core i5-3570 quad core processor. Memory usage is very low (typically below 20Mb) and mostly dictated by the size of the offline map. Table 4.3 shows the computation time for the main and visual odometry threads.

Note that the maximum framerate for the visual odometry is not absolute; when the visual odometry takes longer than the main loop iteration, there will simply be fewer measurements and the covariance will grow slightly. As we have demonstrated in Chapter 2 however, even at 10Hz the odometry is fairly accurate. Clearly a normal desktop computer is capable of running the complete odometry solution several times faster than real-time. While implementation on an embedded system falls outside the scope of this thesis, the performance numbers strongly suggest it is possible on a device with a smartphone-grade processor; memory requirements are low and the dominant factor is the map data, with the entire road map of Belgium taking up less than 500MB.

4.6.5 Position accuracy

As we did not have the equipment to record high quality ground truth for our dataset, the most reliable ground truth source is the map itself. The map information is constructed by the OpenStreetMap community of contributors using filtered

GNSS data of multiple passes, and manually verified against satellite images. The authors of the KITTI dataset determined the average accuracy of the OSM map data to be 1.44m [4, 63]. Much of this error stems from intersections, where the map is often an oversimplification of the real road layout. The limited accuracy of the map data means that it is meaningless to consider errors to sub-meter accuracy; instead we will limit our conclusions to errors larger than two standard deviations (2.88m).

It should be noted that the map itself is also an input to our algorithm. In our opinion this does not invalidate comparisons between GNSS and visual odometry positioning; they both incorporate the same map information, but combine it with different extra measurements. Both the estimated position and its uncertainty (i.e. posterior covariance) will reflect the degree to which the shape of the measured trajectory agrees with the shape of the road map. The map localization and feedback algorithms are such that the state will slowly collapse onto the road segments laterally, but not longitudinally. The rate at which the position tends towards the road segment is still correlated with the accuracy of individual measurements and in any case the GNSS and visual odometry measurements carry significantly more weight in the algorithm than the map data: at a speed of 50km/h the map only updates the Kalman filter at 1.4Hz, versus 30Hz for visual odometry and 10Hz for GNSS and wheel odometry.

Four different situations are compared, using different combinations of information sources. All configurations use the extended Kalman filter to compute estimated position:

1. GNSS only (representing currently used navigation systems),
2. GNSS and offline map,
3. visual odometry and offline map,
4. visual odometry, GNSS, wheel odometry and offline map.

The contribution of wheel odometry was small; variations of configurations 1-3 with added wheel odometry were left out of the comparison so as not to obfuscate the results. The accuracy of the wheel odometry we extracted through the EOBD protocol was insufficient to provide a meanginful advantage.

Figure 4.11 show a section of the test trajectory as it is reconstructed by the various methods.

Table 4.4 shows the average deviation from the nearest map segment for the four configurations, as well as the average position uncertainty. Figure 4.12 shows the cumulative error histograms. From this comparison, we can draw several conclusions:

- the configuration which uses all information sources has the lowest mean distance to the nearest map segment,

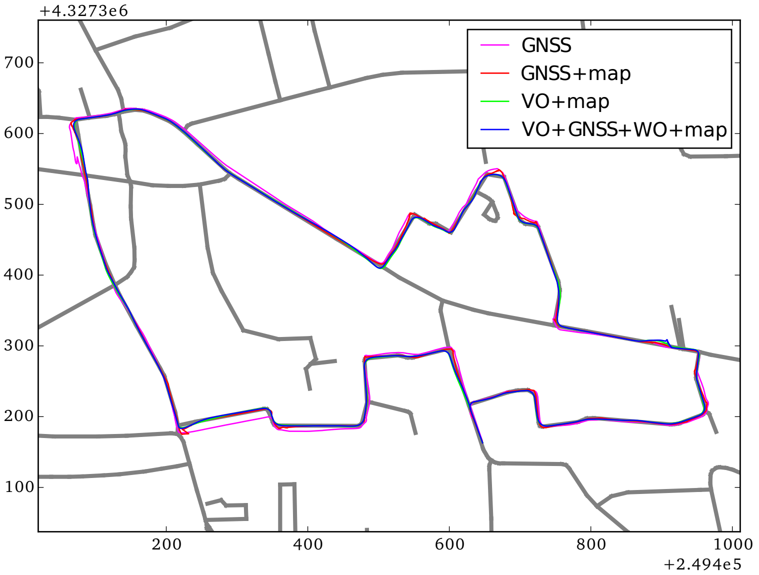


Figure 4.11: *Reconstructed trajectory for a small 2.8km town loop.*

Method	Position error	Uncertainty
GNSS only	3.3	2.1
GNSS and map	3.0	2.1
VO and map	2.7	9.1
VO, GNSS, WO and map	2.3	3.1

Table 4.4: *Average distance to nearest map segment and position uncertainty, per method.*

- the accuracy of visual odometry with map localization exceeds that of currently used GNSS-only navigation systems,
- the proposed map localization technique can improve GNSS accuracy, but not by much,
- fusing GNSS to the proposed visual odometry with map localization reduces larger errors more than smaller errors.

4.7 Qualitative analysis

From the reconstructions, it can be seen that GNSS sometimes exhibits overshoot in sharp corners. This happens when the HDOP is relatively large, and the steady-

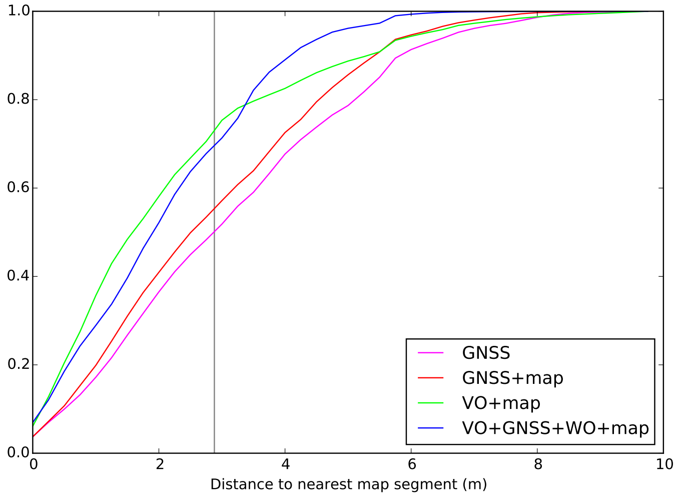


Figure 4.12: *Cumulative position error histogram per method. Vertical line corresponds to 2.88m, which is the two-sigma point for map accuracy (95% of map points are accurate within this distance).*

state assumption therefore holds much weight. Figure 4.13 shows this effect.

HDOP by itself is also not always a reliable indicator of true GNSS uncertainty. Figure 4.14 shows a situation where the HDOP was low, yet the estimated position was offset relative to the true position, possibly due to reflections of a building facade alongside the road.

Another weakness of GNSS navigation is that the applied filtering does not cope well with standing still. Due to random noise, the position tends to wander around. This can be seen in Figure 4.15.

4.8 Conclusion

We have explained how an extended Kalman filter can be used to determine the positional uncertainty inherent to driving longer distances using only relative motion estimation, in casu visual odometry. As an added benefit, this allows for mathematically sound integration with other vehicle sensors. The position uncertainty, in the form of a covariance matrix describing a normal distribution of world position likelihood, can then be combined with an offline map to find the most likely vehicle position, using closed form equations for integrating the likelihood function along a map segment. The use of a hidden Markov model exploits the observation history to have better positioning, as it not only takes into account how close a

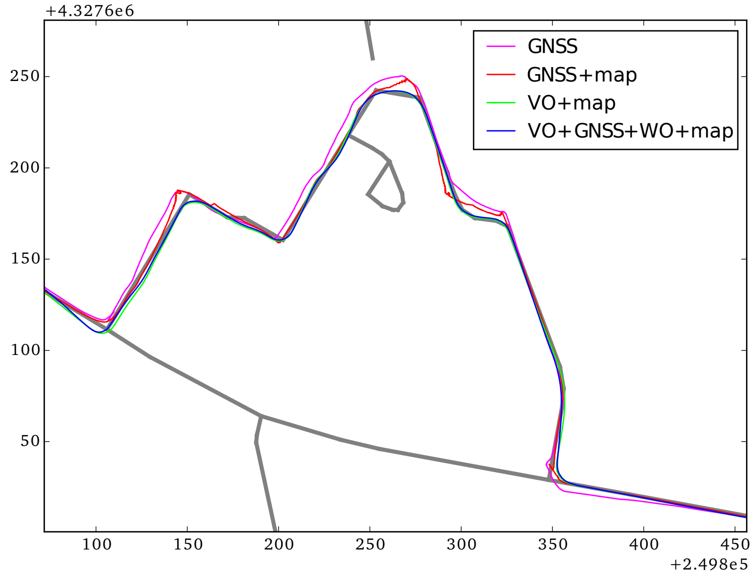


Figure 4.13: *GNSS with poor HDOP exhibits overshoot in corners. HDOP in the top bend in this case was 7 meters.*

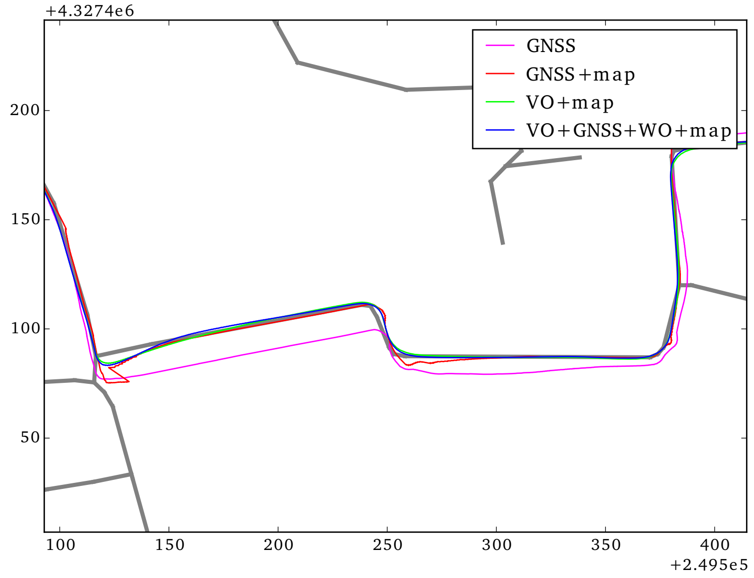


Figure 4.14: *GNSS suffers from constant southern bias, despite low reported HDOP values (HDOP=1).*

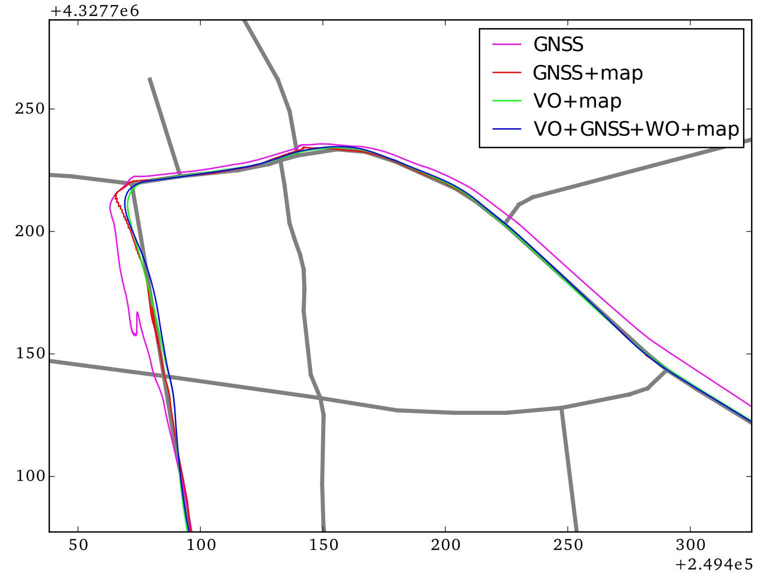


Figure 4.15: *GNSS estimate drifts while standing still (pink kink on left vertical section). Poor HDOP at this point (due to narrow street with 3 story buildings alongside) exacerbates the problem. Inclusion of visual odometry or wheel odometry completely eradicates the wandering.*

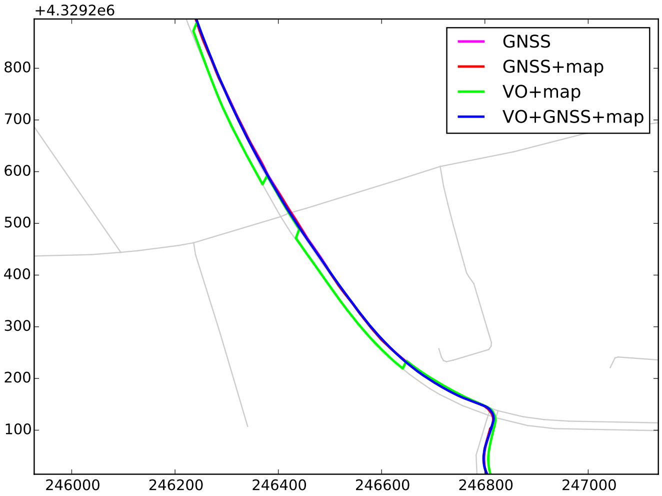


Figure 4.16: *The HMM's most likely hypothesis bounces between the two sections of this highway. The correct hypothesis will be retained after the next turn, but the ambiguity persists for a long period due to parallel nature of the segments. Meta-information could solve this issue as the bottom lane is one-way and runs in the other direction.*

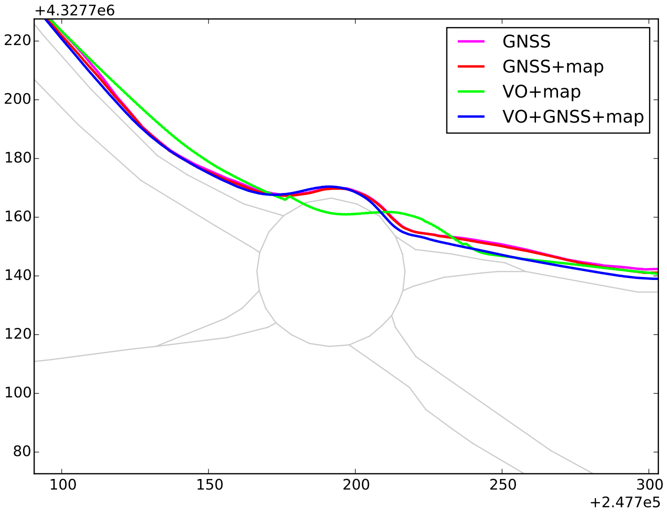


Figure 4.17: *Visual odometry misestimates the length of the approach to this roundabout, causing significant deviation. The HMM correctly estimates the most likely state while the vehicle is on the roundabout, but the state correction is slow due to the low map weight.*

segment is to the likelihood distribution, but also how likely the paths are that lead there. Finally, a feedback mechanism is proposed which suppresses long term drift by influencing the lateral position and orientation of the vehicle with respect to the most likely road segments.

Evaluation shows that the combination of visual odometry and an offline map significantly outperforms the satellite navigation systems currently used in the majority of road vehicles. The developed framework for map feedback based on a hidden Markov model makes GNSS data largely obsolete when visual odometry is available; the incremental improvement in accuracy when including GPS and GLONASS position estimates is very small.

The map matching framework is potentially powerful enough to allow accurate positioning based on wheel odometry; however this remains unproven as we were unable to obtain representative quality wheel odometry from our test setup.

The main scientific contributions made in this chapter are:

- an analysis of the error distribution of the world position, based on the error distributions of the visual odometry estimator,
- a novel framework to perform robust multiple-hypothesis localization using an extended Kalman filter and hidden Markov model,
- a novel method of limiting the position uncertainty using prior map information,
- a comparative evaluation of the proposed framework against commonly used GNSS solutions.

Part of this work has been presented at an international conference:

- **Communicationless navigation through robust visual odometry**, Van Hamme, David; Veelaert, Peter; Philips, Wilfried, *IEEE International Conference on Intelligent Transportation Systems-ITSC (2012)*, pp. 1555-1560.

Additionally, the complete localization solution will be submitted to a peer-reviewed journal.

5 Conclusion

5.1 Global conclusions

At the outset of this research, the goal was to make visual odometry a viable method for real-world vehicle positioning, as a fall-back solution for circumstances in which satellite navigation fails. Throughout this work, we have focused on practicality, meaning the methods must be feasible to integrate in a real mass produced vehicle. It is our opinion that we have achieved this goal.

By opting for a monocular system, we ensured that there are no obstacles to practical implementation. Choosing for two-dimensional ground plane tracking instead of full 3D scene triangulation offered several benefits, mainly computation speed and insensitivity to intrinsic calibration. The downsides normally associated with this monocular, two-dimensional approach, notably reduced accuracy and robustness, were greatly mitigated by the novel algorithm for odometry calculation based on uncertainty regions, to the point where robustness and accuracy exceeded that of the basic epipolar geometry approach which figures extensively in the literature of the last 10 years.

Furthermore, we have described a conceptual framework built around the visual odometry and an offline map which completely eliminates drift from the estimation. This is a significant achievement: it converts relative motion to absolute position. The framework is easily extensible to other sensors; the integration with satellite navigation (GNSS) was already explored. As it turns out, GNSS offers only marginal additional improvements and can be considered obsolete in conditions of good visibility; the visual odometry by itself is then sufficiently accurate to match vehicle position to a road map segment with an accuracy surpassing that of traditional satellite-based navigation.

The visual odometry and mapping framework are fully implemented on a desktop computer, where performance at least twice as fast as real-time is possible. This means that, given further optimization, the methods are suitable for implementation on an embedded platform and integration in a vehicle.

5.2 Contributions

Below is a summary of the main scientific contributions made during this PhD.

Ground plane feature tracking algorithm

In Chapter 2, we proposed a novel algorithm for tracking feature motion in the ground plane, captured from a moving vehicle. The algorithm takes into account the uncertainty of the viewing angle of the camera caused by the suspension motion, the kinematic model (which describes the constraints on the trajectory of the vehicle) and the dynamic model (which constrains acceleration, braking and steering inputs). This algorithm enables reliable feature matching without having to compute descriptors, which saves significant computation time.

Robust odometry estimation

Building on the tracking algorithm described above, we have described how odometry estimation can be performed efficiently and robustly, using a Hough-inspired voting algorithm in which the uncertainty regions cast a vote on a region of parameter space. The algorithm is able to cope with very low inlier rates, as low as 1:8, and provides accuracy surpassing that of a reference 8-point method based on epipolar geometry.

Calibration sensitivity analysis

We have performed experiments to quantify the expected errors of the most widely adopted intrinsic calibration algorithm, and investigated their effect on the estimation accuracy of both the proposed method and a reference method using epipolar geometry. While the reference method is proven to be sensitive to even small calibration errors, the proposed method is much less affected. Because the proposed method does depend on *extrinsic* calibration (the epipolar method does not), we have also quantified the effect of these errors on the estimation accuracy. We have proposed two algorithms to provide initial calibration, and shown how individual extrinsic calibration errors can be identified by comparison with a short section of ground truth, which is important for online calibration refinement in future work.

Map localization by relative motion estimation

We have approximated the error in absolute world position by means of an extended Kalman filter which takes into account the kinematic model of the vehicle, and compared it to the true error distribution obtained by a Monte Carlo simulation. We then described how to determine the most likely current position of the vehicle on a map by evaluating the posterior probability distribution of the EKF as the emission distribution of a hidden Markov model. This approach allows for multiple hypotheses in situations where the combination of odometry and road map is ambiguous.

Drift elimination by mapping

In order to stop the position uncertainty from unbounded growth with increasing travel distance (an inherent problem of relative motion estimators), we have proposed a method in which the prior information contained in the map interacts with the posterior state distributions of the nodes of the HMM. This method effectively eliminates drift.

5.3 Future work

Although this work is complete in the sense that it describes, in detail, a complete system for vehicle positioning using a monocular camera, an offline map and any other sensor input available, there are many opportunities for further improvement. One promising option is the combination with direct image registration. Although this was briefly explored in Chapter 2, the real potential for improvement lies in estimation the instantaneous viewing angle of the camera, and perhaps even of the local curvature function of the road. If successful, this could potentially boost the accuracy to the point where within-lane positioning becomes possible.

A potential weakness in the proposed odometry estimation method is the dependence on accurate extrinsic calibration. While we have provided a thorough analysis of the effects of calibration errors in single angles, it remains future work to use these distinctive effects for automatic calibration refinement, using either a short ground truth trajectory or the road map to correct for long-term biases.

Many of the developed methods could also be applied to other modes of transportation. The odometry estimation has already been successfully applied to bicycles, although the typical accuracy is lower on account of the greater roll angles at which a bicycle turns. As the amount of roll is related to the curvature of the turn, this may be addressed by explicitly modeling this correlation in the predictive model.

The feature matching algorithm based on uncertainty regions has also been demonstrated for a handheld, downward facing camera for pedestrians. This use case is more difficult as the kinetic model is much less constrained as in the vehicle or bicycle case, but meaningful macro-motion estimation was already achieved.

Another very important topic for future research is how to proof the method against adverse weather conditions or other visibility problems. Our performance evaluations were limited to fair weather; preliminary experiments indicated that the method is sensitive to raindrops on the camera, which give rise to strong, but static features which should be excluded from the motion estimation. A possible solution is to continously maintain a feature detection mask where features that have not moved for a significant time are no longer detected, but such a mask would be problematic when the vehicle remains stationary for a while.

Another way to overcome visibility problems would be to use high-quality wheel odometry data. Wheel odometry suffers from several problems, for example sensor bias caused by changes in tire pressure, and inaccuracy due to wheel slip. However, we feel that proper modeling of these effects would make wheel odometry a viable third solution for when both GNSS and visual odometry fail, thanks to the powerful map matching techniques described in Chapter 4.

References

[1] Openstreetmap. URL http://www.openstreetmap.org.

[2] Swisscom reveals the first driverless car on swiss roads. *Swisscom*, May 2015. URL https://www.swisscom.ch/en/about/medien/press-releases/2015/05/20150512-mm-selbstfahrendes-auto.html.

[3] T. Daboczi A. Bodis-Szomoru and Z. Fazekas. *Stereo Vision*, chapter Calibration and Sensitivity Analysis of a Stereo Vision-Based Driver Assistance System. Intech, Rijeka, Croatia, 2008.

[4] C. Stiller A. Geiger, P. Lenz and R. Urtasun. The kitti vision benchmark suite. http://www.cvlibs.net/datasets/kitti/eval_odometry.php. Accessed: 10 Jan. 2015.

[5] ACEA. The automobile industry pocket guide. Technical report, European Automoble Manufacturers Association, 2015.

[6] Georges Baatz, Kevin Köser, David Chen, Radek Grzeszczuk, and Marc Pollefeys. Leveraging 3d city models for rotation invariant place-of-interest recognition. *International Journal of Computer Vision*, 96(3):315–334, 2012. doi: 10.1007/s11263-011-0458-7. URL http://dx.doi.org/10.1007/s11263-011-0458-7.

[7] H. Badino, D. Huber, and T. Kanade. Real-time topometric localization. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1635–1642, May 2012. doi: 10.1109/ICRA.2012.6224716.

[8] Hernan Badino, Akihiro Yamamoto, and Takeo Kanade. Visual odometry by multi-frame feature integration. In *International Workshop on Computer Vision for Autonomous Driving @ ICCV*, December 2013.

[9] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. ISSN 1077-3142. doi: 10.1016/j.cviu.2007.09.014. URL http://dx.doi.org/10.1016/j.cviu.2007.09.014.

[10] Fabio Bellavia, Marco Fanfani, Fabio Pazzaglia, and Carlo Colombo. Robust selective stereo slam without loop closure and bundle adjustment. In Alfredo Petrosino, editor, *Image Analysis and Processing – ICIAP 2013*, volume 8156 of *Lecture Notes in Computer Science*, pages 462–471. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-41180-9.

[11] Stan Birchfield. Klt: An implementation of the kanade-lucas-tomasi feature tracker. URL `https://www.ces.clemson.edu/~stb/klt/`.

[12] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[13] J. Bouguet. *Visual Methods for Three-Dimensional Modeling*. PhD thesis, California Institute of Technology, 1999.

[14] R. Chandramouli and N. Ranganathan. Computing the bivariate gaussian probability integral. *IEEE Signal Processing Letters*, 6(6):129–131, June 1999. ISSN 1070-9908. doi: 10.1109/97.763142.

[15] Jianshu Chao, A. Al-Nuaimi, G. Schroth, and E. Steinbach. Performance comparison of various feature detector-descriptor combinations for content-based image retrieval with jpeg-encoded query images. In *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*, pages 029–034, Sept 2013. doi: 10.1109/MMSP.2013.6659259.

[16] Shilai Cheng, D. Perissin, Fulong Chen, and Hui Lin. Atmospheric delay analysis from gps and insar. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, pages 1650–1653, July 2011. doi: 10.1109/IGARSS.2011.6049549.

[17] O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 772–779 vol. 1, June 2005. doi: 10.1109/CVPR.2005.354.

[18] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardós. Mapping large loops with a single hand-held camera. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[19] A.I. Comport, E. Malis, and P. Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 40–45, April 2007. doi: 10.1109/ROBOT.2007.363762.

[20] T. Dang, C. Hoffmann, and C. Stiller. Continuous stereo self-calibration by camera parameter tracking. *Image Processing, IEEE Transactions on*, 18(7):1536–1550, July 2009. ISSN 1057-7149. doi: 10.1109/TIP.2009.2017824.

[21] E. Davin. Parameter identification of a linear single track vehicle model. Technical report, Technical University of Eindhoven, 2011.

[22] P. Decker, D. Paulus, and T. Feldmann. Dealing with degeneracy in essential matrix estimation. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1964–1967, Oct 2008. doi: 10.1109/ICIP.2008.4712167.

[23] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, page 594 Vol. 2, 1999. doi: 10.1109/CVPR.1999.784976.

[24] K. Douterloigne, S. Gautama, and W. Philips. Registration of vector data and aerial thermal images using modified mutual information. In *International Geoscience & Remote Sensing Symposium (IGARSS)*, pages 570–573, Vancouver, Canada, July 2011.

[25] Pauline Ducamp. Des véhicules autonomes sur route ouverte à bordeaux en octobre 2015. *L'Usine Digitale*, June 2015.

[26] E. Eade. *Monocular Simultaneous Localisation and Mapping*. PhD thesis, University of Cambridge, 2008.

[27] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1449–1456, Dec 2013. doi: 10.1109/ICCV.2013.183.

[28] Jakob Engel, Thomas Schops, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8690 of *Lecture Notes in Computer Science*, pages 834–849. Springer International Publishing, 2014. ISBN 978-3-319-10604-5.

[29] Eurostat. European social statistics. Technical report, Eurostat, 2015. URL `http://ec.europa.eu/eurostat/statistics-explained/index.php/European_social_statistics`.

[30] Matthias Faessler, Flavio Fontana, Christian Forster, Elias Mueggler, Matia Pizzoli, and Davide Scaramuzza. Autonomous, vision-based flight and live

dense 3d mapping with a quadrotor micro aerial vehicle. *Journal of Field Robotics*, pages n/a–n/a, 2015. ISSN 1556-4967. doi: 10.1002/rob.21581. URL `http://dx.doi.org/10.1002/rob.21581`.

[31] Olivier Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. Technical Report RR-0856, INRIA, June 1988. URL `https://hal.inria.fr/inria-00075698`.

[32] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22, May 2014. doi: 10.1109/ICRA.2014.6906584.

[33] A. Geiger. Libviso2: C++ library for visual odometry 2. `http://www.cvlibs.net/software/libviso`. Accessed: 21 May 2014.

[34] Andreas Geiger. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 3354–3361, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-1-4673-1226-4. URL `http://dl.acm.org/citation.cfm?id=2354409.2354978`.

[35] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.

[36] Ignace Glorieux, Suzana Koelet, and Inge Mestdag. *De 24 Uur Van Vlaanderen : Het Dagelijkse Leven Van Minuut Tot Minuut*. LannooCampus, 2006.

[37] C. Y. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet. Online map-matching based on hidden markov model for real-time traffic sensing applications. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 776–781, Sept 2012. doi: 10.1109/ITSC.2012.6338627.

[38] Lee Gomes. Hidden obstacles for google's self-driving cars. Technical report, Massachusetts Institute of Technology, 2014. URL `https://www.technologyreview.com/s/530276/hidden-obstacles-for-googles-self-driving-cars/`.

[39] Liran Goshen and Ilan Shimshoni. Balanced exploration and exploitation model search for efficient epipolar geometry estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1230–1242, 2008. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2007.70768.

[40] Jianjun Gui, Dongbing Gu, and Huosheng Hu. Robust direct visual inertial odometry via entropy-based relative pose estimation. In *Mechatronics and Automation (ICMA), 2015 IEEE International Conference on*, pages 887–892, Aug 2015. doi: 10.1109/ICMA.2015.7237603.

[41] J-S Gutmann, Wolfram Burgard, Dieter Fox, and Kurt Konolige. An experimental comparison of localization methods. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 2, pages 736–743. IEEE, 1998.

[42] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[43] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, chapter Computation of the Fundamental Matrix F, pages 279–309. Cambridge Univ. Press, Cambridge, UK, 2004.

[44] R.I. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, Jun 1997. ISSN 0162-8828. doi: 10.1109/34.601246.

[45] Richard I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, pages 237–256, London, UK, UK, 1994. Springer-Verlag. ISBN 3-540-58240-1. URL `http://dl.acm.org/citation.cfm?id=647302.760233`.

[46] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[47] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112, Jun 1997. doi: 10.1109/CVPR.1997.609468.

[48] Pengda Huang and Yiming Pi. Urban environment solutions to gps signal near-far effect. *Aerospace and Electronic Systems Magazine, IEEE*, 26(5): 18–27, May 2011. ISSN 0885-8985. doi: 10.1109/MAES.2011.5871387.

[49] M. C. Jones, J. S. Marron, and S. J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996. doi: 10.1080/01621459.1996.10476701.

[50] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82 (Series D):35–45, 1960.

[51] C. Kerl, J. Sturm, and D. Cremers. Dense visual slam for rgb-d cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2100–2106, Nov 2013. doi: 10.1109/IROS.2013.6696650.

[52] Faten A. Khalifa, Noura A. Semary, Hatem M. El-Sayed, and Mohiy M. Hadhoud. A comparison of local detectors and descriptors for multi-object applications. In *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*, IPAC '15, pages 23:1–23:5, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3458-7. doi: 10.1145/2816839.2816907. URL `http://doi.acm.org/10.1145/2816839.2816907`.

[53] Bernd Manfred Kitt, Joern Rehder, Andrew D Chambers, Miriam Schonbein, Henning Lategahn, and Sanjiv Singh. Monocular visual odometry using a planar road model to solve scale ambiguity. In *Proc. European Conference on Mobile Robots*, September 2011.

[54] J. Lankinen, V. Kangas, and J.-K. Kamarainen. A comparison of local feature detectors and descriptors for visual object categorization by intra-class repeatability and matching. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 780–783, Nov 2012.

[55] Jean-Paul P. Laumond. *Robot Motion Planning and Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998. ISBN 3540762191.

[56] S. Leutenegger, M. Chli, and R.Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555, Nov 2011. doi: 10.1109/ICCV.2011.6126542.

[57] Tony Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, pages 224–270, 1994.

[58] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30:79–116, 1998.

[59] G. Long. Acceleration characteristics of starting vehicles. Technical report, Transportation Research Board, 2000.

[60] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, Sept. 1981.

[61] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL `http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94`.

[62] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=1623264.1623280`.

[63] A. M. Brubaker and R. Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *Conf. Computer Vision and Pattern Recognition*, 2013.

[64] Ezio Malis and Manuel Vargas. Deeper understanding of the homography decomposition for vision-based control. Research Report RR-6303, INRIA, 2007. URL `https://hal.inria.fr/inria-00174036`.

[65] A.K. Maurya and P.S. Bokare. Study of deceleration behaviour of different vehicle types. *Int. J. Traffic and Transport Engineering*, 2(3):253–270, 2012.

[66] B. A. McElhoe. An assessment of the navigation and course corrections for a manned flyby of mars or venus. *IEEE Transactions on Aerospace and Electronic Systems*, AES-2(4):613–623, July 1966. ISSN 0018-9251. doi: 10.1109/TAES.1966.4501892.

[67] M. Miksch, Bin Yang, and K. Zimmermann. Automatic extrinsic camera self-calibration based on homography and epipolar geometry. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 832–839, June 2010. doi: 10.1109/IVS.2010.5548048.

[68] M Hossein Mirabdollah and Bärbel Mertsching. On the second order statistics of essential matrix elements. In *German Conference on Pattern Recognition*, pages 547–557. Springer, 2014.

[69] M Hossein Mirabdollah and Bärbel Mertsching. Fast techniques for monocular visual odometry. In *German Conference on Pattern Recognition*, pages 297–307. Springer, 2015.

[70] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965. doi: 10.1093/comjnl/7.4.308. URL `http://comjnl.oxfordjournals.org/content/7/4/308.abstract`.

[71] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 336–343, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-649-6. doi: 10.1145/1653771.1653818. URL `http://doi.acm.org/10.1145/1653771.1653818`.

[72] D. Nister. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):756–770, June 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.17.

[73] UK Department of Transportation. Driverless cars in the uk: a regulatory review. Technical report, 2015. URL `https://www.gov.uk/government/publications/driverless-cars-in-the-uk-a-regulatory-review`.

[74] Sang Min Oh, S. Tariq, B. N. Walker, and F. Dellaert. Map-based priors for localization. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2179–2184 vol.3, Sept 2004. doi: 10.1109/IROS.2004.1389732.

[75] Alejandro Palacio, Giuseppe Tamburro, Desmond O'Neill, and Ciaran K. Simms. Non-collision injuries in urban buses—strategies for prevention. *Accident Analysis and Prevention*, 41(1):1 – 9, 2009. ISSN 0001-4575. doi: http://dx.doi.org/10.1016/j.aap.2008.08.016. URL `http://www.sciencedirect.com/science/article/pii/S0001457508001279`.

[76] I. Parra Alonso, D. Fernandez Llorca, M. Gavilan, S. Alvarez Pardo, M.A. Garcia-Garrido, L. Vlacic, and M.A. Sotelo. Accurate global localization using visual odometry and digital maps on urban environments. In *Intelligent Transportation Systems, IEEE Transactions on*, volume 13, pages 1535–1545, Dec 2012.

[77] Emanuel Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076, 09 1962. doi: 10.1214/aoms/1177704472. URL `http://dx.doi.org/10.1214/aoms/1177704472`.

[78] J. Philip. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *Photogrammetric Record*, 15(88):589–599, Oct. 1996.

[79] P. Pinies and J.D. Tardos. Large-scale slam building conditionally independent local maps: Application to monocular vision. *Robotics, IEEE*

Transactions on, 24(5):1094–1106, Oct 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2004636.

[80] M. Pollefeys and L. Van Gool. Stratified self-calibration with the modulus constraint. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):707–724, Aug 1999. ISSN 0162-8828. doi: 10.1109/34.784285.

[81] Michael J. Quinlan and Richard H. Middleton. *Multiple Model Kalman Filters: A Localization Technique for RoboCup Soccer*, pages 276–287. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-11876-0. doi: 10.1007/978-3-642-11876-0_24. URL `http://dx.doi.org/10.1007/978-3-642-11876-0_24`.

[82] R. Raymond, T. Morimura, T. Osogami, and N. Hirosue. Map matching with hidden markov model on sampled road network. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2242–2245, Nov 2012.

[83] Ming Ren and Hassan A. Karimi. A hidden markov model-based map-matching algorithm for wheelchair navigation. *The Journal of Navigation*, 62:383–395, 7 2009. ISSN 1469-7785. doi: 10.1017/S0373463309005347. URL `http://journals.cambridge.org/article_S0373463309005347`.

[84] J.A. Rosenow. *MnDOT Road Design Manual*, chapter Cross Sections. 2012.

[85] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, May 2006. doi: 10.1007/11744023_34. URL `http://www.coxphysics.com/work/rosten_2006_machine.pdf`.

[86] Nick Savage, David Ndzi, Andrew Seville, Enric Vilar, and John Austin. Radio wave propagation through vegetation: Factors influencing signal attenuation. *Radio Science*, 38(5):n/a–n/a, 2003. ISSN 1944-799X. doi: 10.1029/2002RS002758. URL `http://dx.doi.org/10.1029/2002RS002758`. 1088.

[87] D. Scaramuzza and R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *Robotics, IEEE Transactions on*, 24(5):1015–1026, Oct 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2004490.

[88] Davide Scaramuzza. Performance evaluation of 1-point-ransac visual odometry. *Journal of Field Robotics*, 28(5):792–811, 2011. ISSN 1556-4967. doi: 10.1002/rob.20411. URL `http://dx.doi.org/10.1002/rob.20411`.

[89] Jianbo Shi and Carlo Tomasi. Good features to track. pages 593–600, 1994.

[90] S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23:45–78, 1995.

[91] Shiyu Song and Manmohan Chandraker. Robust scale estimation in real-time monocular sfm for autonomous driving. In *CVPR*, Columbus, Ohio, USA, June 2014.

[92] J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *dagm*, pages 11–20, Darmstadt, Germany, September 2010.

[93] Piotr Szwed and Kamil Pekala. *Artificial Intelligence and Soft Computing: 13th International Conference, ICAISC 2014, Zakopane, Poland, June 1-5, 2014, Proceedings, Part II*, chapter An Incremental Map-Matching Algorithm Based on Hidden Markov Model, pages 579–590. Springer International Publishing, Cham, 2014. ISBN 978-3-319-07176-3. doi: 10.1007/978-3-319-07176-3_51. URL `http://dx.doi.org/10.1007/978-3-319-07176-3_51`.

[94] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2531–2538, Sept 2008. doi: 10.1109/IROS.2008.4651205.

[95] Kristof Teelen. *Geometric uncertainty models for correspondence problems in digital image processing*. PhD thesis, Ghent University, 2010.

[96] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.

[97] P. H. S. Torr, A. Zisserman, and S. Maybank. Robust detection of degenerate configurations for the fundamental matrix. *Computer Vision and Image Understanding*, 71(3):312–333, 1998.

[98] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1):27–44, 1999.

[99] V. Usenko, J. Engel, J. Stueckler, and D. Cremers. Reconstructing street-scenes in real-time from a driving car. In *Proc. of the Int. Conference on 3D Vision (3DV)*, October 2015. accepted.

[100] E. Wan. Sigma-point filters: An overview with applications to integrated navigation and vision assisted control. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 201–202, Sept 2006. doi: 10.1109/NSSPW.2006.4378854.

[101] A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof. Dense reconstruction on-the-fly. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1450–1457, June 2012. doi: 10.1109/CVPR.2012.6247833.

[102] Brian P. Williams. *Simultaneous Localisation and Mapping using a single camera*. PhD thesis, University of Oxford, 2009.

[103] Oliver J. Woodman. An introduction to inertial navigation. Technical report, 2007.

[104] Chen Xiao, Xiaorui Zhu, Wei Feng, and Yongsheng Ou. A novel approach to improve the precision of monocular visual odometry. In *Information and Automation, 2015 IEEE International Conference on*, pages 392–397, Aug 2015. doi: 10.1109/ICInfA.2015.7279319.

[105] J. Yang, H. Li, and Y. Jia. Optimal essential matrix estimation via inlier-set maximization. In *Computer Vision - ECCV 2014*, volume 8689 of *Lecture Notes in Computer Science*, pages 111–126, 2014.

[106] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673 vol.1, 1999. doi: 10.1109/ICCV.1999.791289.

[107] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000. ISSN 0162-8828. doi: 10.1109/34.888718.

[108] Zhongfei Zhang and Allen R. Hanson. 3d reconstruction based on homography mapping. In *In ARPA Image Understanding Workshop*, pages 0249–6399, 1996.