UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Electrónica

Visual Odometry

A Thesis submitted for the degree of Doctor of Philosophy

Author

Ignacio Parra Alonso

Supervisor

Dr. D. Miguel Ángel Sotelo Vázquez

2010

To all the mojones on the Earth

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the last years, vision-based systems have become a key element in the research and development of intelligent transportation systems (ITS) and particularly of intelligent vehicles (IV). Moreover, numerous vision-based systems are already in production such as pedestrian detection (PD), lane departure warning (LDW), forward collision warning (FCW), blind spot detection (BSD), intelligent headlight control (IHC), traffic signs recognition (TSR), inventory and quality assessment, road cracks detection and many more to come. The main reasons behind this success are economical and technical. Conventional cameras are cheaper than other commonly used sensors such as LASERs, RADARs or LIDARs and allow for an easy integration between systems (i.e. combining LDW and TSR (OpelEye); FCW, LDW and headway monitoring (Mobileye AWS-4000 (Figure 1.1)) or the new BMW 7 series with Mobileye's LDW, IHC and TSR). As passive sensors, they can be used in as many vehicles as needed, avoiding interferences with other sensors onboard the vehicle or other vehicles on the road. In addition, thanks to the last advances in hardware, it is becoming more and more tractable to process the large amount of data delivered by cameras on standard PC-based systems. The information provided by vision-based systems is extremely rich (shape, texture, color) specially with the last developments on what is known as structure from motion problem (recovering 3D structure from 2D camera images) which is closing the gap with the traditional ranging sensors. This range information is not only used to reconstruct the 3D structure of the scene but to estimate the motion suffered by a camera between two different views.

Figure 1.1: Mobileye AWS-4000 camera based safety solution. It performs forward collision warning, lane departure warning and headway monitoring and warning with a single camera.

Accurate Global Localization has become a key component in vehicle navigation, following the trend of the robotics area, which has seen significant progress in the last decade. Autonomous vehicle guidance interest has increased in the recent years, thanks to events like the Defense Advanced Research Projects Agency (DARPA) Grand Challenge and recently the Urban Challenge. Many ITS applications and services such as route guidance, fleet management, road user charging, accident and emergency response, bus arrival information and other location based services require location information. In the last few years, GPS has become the main positioning technology for providing location data for ITS applications [Quddus 07]. However, due to signal blockage and severe multipath in urban areas, GPS can not satisfy most vehicle navigation requirements. Dead Reckoning systems have been widely used to bridge the gaps of GPS position error, but their drift errors increase rapidly with time and frequent calibration is required [Wu 03]. Vision-based algorithms have proven to be capable of tracking the position of a vehicle over long distances using only the images as inputs and with no a priory knowledge of the environment [Agrawal 06]. Moreover, if combined with map matching algorithms cumulative errors can be corrected and even longer distances could be travelled without the necessity of a correction of the absolute position. The integration of a vision-based localization system with other applications such as PD, LDW or FCW will reduce maintenance and costs.

1.2 Motion estimation in complex urban environments using vision issues

In recent years a lot of research has been carried out on systems to estimate the ego-motion of a vehicle using vision, but very few have addressed the specific problems of complex urban environments. Most of the work is focused on robotics platforms and outdoors environments. Even more, among the few examples of ego-motion estimation in urban environments, many of them were developed for robotic platforms and/or outdoors environments and have been tested on urban environments but not developed for them. Here we will comment some of the specific challenges that have to be solved to success in the task of ego-motion estimation in complex urban environments.

1.2.1 Illumination

As stated before, using cameras instead of other sensors for ego-motion computation reduces maintenance and costs . However, they strongly depend on the illumination conditions and have to be calibrated to get accurate information. Urban environments are very demanding for cameras. Illumination changes in tunnels, glares, saturation or the shadow casted by buildings are difficult to avoid in real conditions (see examples of these situations in Figure 1.2). Adaptive shutter or gain controls can not deal with all the situations and have to be used carefully to avoid blurring the image due to very long exposure times or introducing noise due to high gains. Camera shades to protect the camera lens usually cover part of the field of view. Polarizing filters are expensive and not suitable for mass production. Another problem of conventional cameras is that they can not work at night time conditions unless external illumination is provided.

Therefore, on the one hand, a good camera control of exposure time and gain has to be performed to get the best possible quality of images. On the other hand, the system has to be robust to handle the artifacts that can not be avoided.

(a) Glares on the windshield

(b) Glares on other cars

(c) Underexposed image

(d) Overexposed image at a tunnel exit

Figure 1.2: Examples of problems with illumination conditions.

1.2.2 Complexity

Urban environments are extremely complex and their conditions are variable. Most of the previous work on motion estimation has been carried out in off-road environments where very few features are available and the effort has to be put in finding and tracking features as long as possible. However, urban environments are cluttered and repetitive, superimposed objects appear on the images and some of the objects on the scene are not stationary (see Figure 1.3). In general, too many features are found and the effort has to be put in the selection of the best features for the motion estimation which may not be the same as in a traditional feature detection and tracking system. Non stationary objects such as pedestrians, other cars or buses have to be rejected and the effect of incorrect matchings due to repetitive patterns minimized.

1.2.3 Motion

The trajectory followed by a vehicle in urban roads is very different to the paths in off-road environments. Two main differences are important for the motion estimation. Firstly the speed in off-roads environments is lower. This eases the feature detection because the images are seldom blurred due to the motion. It also eases the tracking, because the features remain longer in the field of view of the cameras. In addition, features that are close to the camera can be tracked if the motion doesn't take them outside of the field of view. As the 3D position accuracy decreases with depth, the closer the features the higher the accuracy in their position, and thus in the motion estimation. Secondly the off-

(a) Bus crossing the scene

(b) Truck crossing the scene

(c) Superimposed cars

(d) Repetitive patterns

Figure 1.3: Examples of the complexity of urban environments.

roads trajectories tend to be devious, not moving forward for long distances, which eases the motion estimation problem. Good optical flow is needed for the motion estimation and a vehicle moving forwards doesn't produce much optical flow for a camera also facing forwards. However, in urban environments this is the kind of motion it will be undertaking most of the time. Moreover, there will be other cars following a parallel trajectory to the ego-vehicle. While a crossing truck or pedestrian can be discarded taking the ego-motion estimation into account, objects with a similar motion will remain longer on the scene than stationary objects and can be repeatedly tracked.

1.3 Objectives

Our final goal is the autonomous vehicle outdoor navigation in large-scale environments and the improvement of current vehicle navigation systems based only on standard GPS. The work proposed is particularly efficient in areas where GPS signal is not reliable or even not fully available (tunnels, urban areas with tall buildings, mountainous forested environments, etc). Our research objective is to develop a robust localization system based on a low-cost stereo camera system and a digital map that assists a standard GPS sensor for vehicle navigation tasks. Then, our work is focused on stereo vision-based real-time localization as the main output of interest. The challenge now is to extend stereo-vision capabilities to also provide accurate estimation of the vehicle's ego-motion with respect to the road, and thus to compute its global position.

(a) Frame 2534 (b) Frame 2598

Figure 1.4: Examples of long straight streets and a car with similar motion

1.4 Document Structure

After the introduction in Chapter 1, Chapter 2 contains a brief review of the most significative published research on motion estimation.

In Chapter 3 the cameras model and the stereo geometry are described. Also the error model for the used 3D reconstruction and the calibration are presented.

The proposed ego-motion estimation system and different feature extractors is explained in Chapter 4. Results for real and simulated experiments are presented and conclusions for each one of the configurations are discussed.

Chapter 5 presents a general overview of the map-matching algorithm and the proposed solution is explained. Results for experiments on real traffic conditions are presented and discussed.

Finally Chapter 6 contains the conclusions and main contributions of this work, and future research lines that may spring from it.

Chapter 2

State of the Art

The problem of recovering relative camera poses and 3-D structure from a set of 2-D camera images has been some of the most active fields of research in computer vision for the last 3 decades. Very impressive results have been obtained over long distances using monocular [Mouragnon 06] [Stein 00], stereo systems [Konolige 07] [Nistér 06] and omni-directional cameras [Scaramuzza 08]. Furthermore, visual odometry has been successfully used by the NASA rovers since early 2004 [Maomone 07], implemented into commercial applications [Shashua 04] and focus of interest for the Defense Advance Research Projects (DARPA) [Nistér 06] [Konolige 07] .

This chapter presents a brief survey of the state of the art in what is known in the computer vision community as "structure from motion" [Hartley 04]. For the sake of clarity, the different approaches presented in this chapter will be divided according to the sensor used in monocular and stereo.

2.1 Monocular systems

The use of a single camera to compute the structure from motion is a challenging problem and it has its main difficulty in the necessity of the estimation of a scale factor to recover the real scale of the scene. On the other hand, it allows for a simple integration with other computer vision systems without the need of calibration between sensors reducing maintenance and cost.

In this line, a monocular system for computing the ego-motion of a vehicle was developed by the company MobilEye [Mobileye 07]. In [Stein 00] they present a method for computing the ego-motion of a vehicle relative to the road using a single camera mounted next to the rear view mirror. Video sequences were acquired at 30fps, 320×240 and with 50° FOV at normal traffic speed (see Figure 2.1). This camera configuration, although not the ideal one, allows to use the same sensor for other applications such as pedestrian detection, lane departure warning, adaptive cruise control or collision mitigation. Two assumptions are made in their system: first, that the roadway is a planar structure and that the measurements will be on the road itself; second, that only 3 parameters are necessary to estimate the ego-motion of the vehicle: forward translation, pitch and yaw. The reason for this unusual simplification of the 2D movement of the vehicle seems to be that the ego-motion system is devised to serve as input to other systems (i.e. pedestrian detection) that will find more useful the pitch of the ego-motion than the lateral translation.

Their motion model assumes that the Z axis of the world coordinate system is aligned with the optical axis of the camera and the X and Y axis are aligned with the image axis

(a) Example of hard conditions with a car and its shadow

(b) Example of night conditions

Figure 2.1: Example images from the system in [Stein 00]

x and y. To ensure this condition they use a calibration process in which they manually correct the extrinsic calibration parameters of the camera. In this calibration procedure they record a video of a pure translational motion (no yaw, no pitch) and estimate the ego-motion. If the camera optical axis is aligned with the direction of motion, a pure translational motion will result on a pure translational estimation. Any drift on the estimation of the yaw means a rotation on the yaw angle of the camera. The same applies for a rotation on the pitch angle. Both angles are manually adjusted until no drift is found.

To avoid the problem of finding correspondences in poor textured roads and of incorrect matches in complex urban environments they propose a direct method [Horn 88] [Stein 97] where each pixel contributes a measurement. These measurements are then combined in a global probability function for the parameters of the ego motion. This probability function takes into account the probability of a patch of being the road and the gradient of information of the patch. Each patch of the image is warped using an initial motion estimation towards the next image and the sum squared difference with the real patch is computed. The best motion for every patch is also computed in a small cube of the possible 3D movements. Combining the probabilities and using a gradient descent minimization the motion is estimated.

As a starting guess, they use either the information from the car speedometer or assume a constant velocity of 40Km/h and wait for the system to converge, which usually happens after 2-3 seconds of motion. The system was tested on rain and night conditions, but results are only presented for a circular loop with no ground truth.

The results indicate an accurate precision on the pitch and yaw angles (0.017° per frame), but no so good in the translation. As mentioned before, the most likely use of this ego-motion information is as an aid to the tracking step of other vision based system where the pitch and yaw angles of the ego-motion are of crucial importance [Llorca 08] [Llorca 09].

A similar approach, using a monocular omnidirectional camera, was presented in [Scaramuzza 08]. In this work, an omnidirectional camera mounted on the roof of a vehicle was used to estimate the ego-motion of a vehicle relative to the road under the assumption of planar motion. To do so, translation and rotation are estimated separately. The translation is estimated using an homography-based tracker that detects and matches robust scale invariant features (SIFT [Lowe 04]) that most likely belong to the ground plane. An

appearance based approach in line with [Labrosse 06], is used to estimate the rotation of the vehicle.

This work exploits the coplanar relation between two views of the same plane [Tsai 81] [Longuet-Higgins 86] [Faugueras 88]. Given two different images of the same world points there is an homography $\mathbf{H}$ that relates the two camera projections of the same plane points. If the camera is calibrated it is possible to recover the rotation and translation $(\mathbf{R}, \mathbf{T})$ between them using the homography matrix $\mathbf{H}$.

At this point they propose two different ways of recovering the rotation and translation from the homography depending on the spatial distribution of the points detected. If the points are spatially uniformly distributed they use a linear method for decomposing $\mathbf{H}$ originally developed by [Wunderlich 82] and later reformulated by [Triggs 98]. This algorithm is based on the singular value decomposition of $\mathbf{H}$. If the image points are too close to a degenerate configuration or they are spatially distributed within one side of the whole omnidirectional image (see Figure 2.2) then they use an Euclidean approximation to a planar motion. In this second solution they solve the system using least-squares and forcing the rotation matrix to be orthonormal through a Hartley's normalization [Hartley 04] and singular value decomposition.

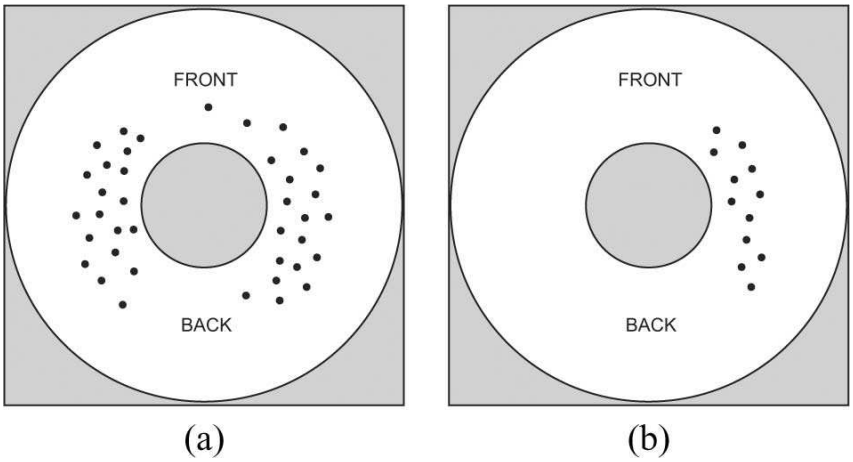


Figure 2.2: (a) Uniform distribution of features. (b) Nonuniform distribution

In their results, they show that the Euclidean method is more robust to image noise when the motion is purely planar and the camera perfectly vertical. Otherwise the more complex Triggs method yields better results. To discard the outliers (bad matches, non static points) they use a Random Sample Consensus paradigm (RANSAC) [Fischler 81].

The previous motion estimation is obtained by linear methods that minimize an algebraic distance, which is not physically meaningful. So, on a further step, they refine the solution by minimizing the maximum likelihood estimate of the re-projection error using the first estimation as starting point. At this step they assume planar motion, accordingly only yaw, forward and lateral translation are estimated. To minimize the re-projection error they use the Levenberg-Marquardt algorithm. The rotation resulting from this method is extremely sensitive to systematic errors so they propose an appearance based method inspired in [Labrosse 06] to recover the rotation undertaken by the camera. The idea is to compare patches of the unwrapped images shifted a certain number of columns. If the motion is purely rotational, and the camera is perfectly vertical the exact rotation angle can be retrieved by finding the best match between a reference image (before rotation) and a column-wise shift of the successive image (after rotation). If the camera undergoes

small displacements or the distance to the objects is big compared to the displacement the pure rotation assumption can still be maintained. According to the last considerations, only two regions of the omnidirectional image are used for the rotation estimation: a small field of view around the front and the back of the camera (Figure 2.3)

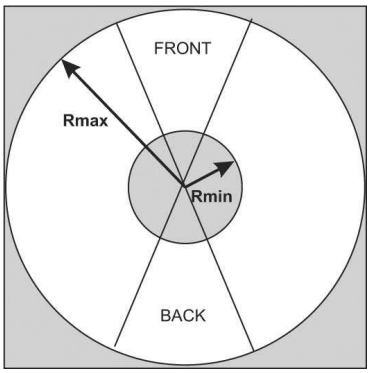


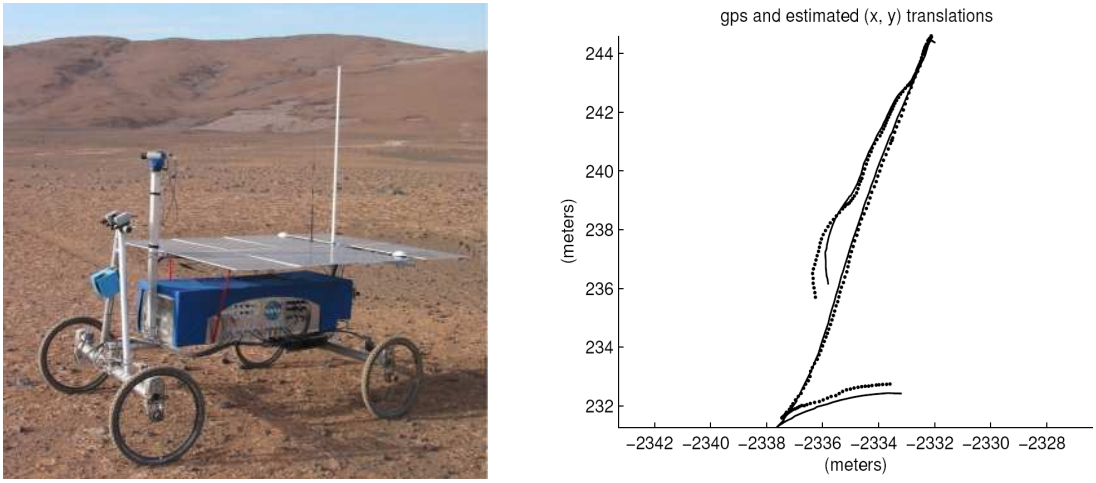
Figure 2.3: Reduced omnidirectional image sections used for the ego-motion computation. For the translation estimation only the image in white is used. For the visual compass only the sectors labeled as FRONT and BACK are used

The algorithm was tested using a 640×480 omnidirectional camera acquiring at 10 Hz on an urban scenario. The length of the path was around 400 m in a closed loop 2.4. The resulting path was overlaid on Google Maps, showing a motion trajectory close to the real one.

Figure 2.4: The estimated path overlaid onto a Google Earth image of the test environment. [Scaramuzza 08]

The main advantage of omnidirectional cameras is that motion in any direction produces good optical flow, while conventional cameras require to be pointed orthogonal to the optical axis [Baker 03]. As can be seen in Figure 2.2 when the system is estimating the translation, feature points are selected at the ”sides” while the rotation is estimated using points at the front and the back (see Figure 2.3). In addition features are retained longer in the wide field of view of an omnidirectional camera. On the contrary feature tracking is more complex due to the strong deformation introduced by the mirror, their price is higher and they are more difficult to integrate with other ADAS applications than conventional cameras.

A closely related work was developed at the Carnegie-Mellon University for a Planetary Rover [Corke 04] . They used an omnidirectional camera mounted on a solar powered rover that recorded color video sequences in the Atacama desert in Chile (see Figure 2.5(a)). The idea was to develop a method that compensated for the odometric error produced in planetary rovers. To do so, Lucas-Kanade [Lucas 81b] method was used to track features through the image sequence as long as they are visible. To recover the structure from motion they used an iterative extended Kalman filter (iEKF) [Strelow 01] with a state vector composed of the 6 camera position parameters and the 3D positions of p 3D points. So the total size of the state is $6+3p$ where p is the number of tracked points in the current image. The results for relatively short runs are good (1% deviation in 29.2m) but the filter shows degeneration for long displacements (after 300 images the filter fails) (see Figure 2.5(b)). The growing uncertainty and the algorithmic complexity discourage the use of this kind of filters for applications with long runs or without loops where the filter can reduce the uncertainty.



(a) Solar powered robot used in the experiments

(b) GPS ground truth (solid) and estimation (dashed)

Figure 2.5: Images from the system in [Corke 04]

Another impressive work using monocular cameras was developed in the LASMEA UMR at France. In [Mouragnon 06], they propose a monocular system based on the tracking of Harris corners [Hariis 88] and the use of the 5-points algorithm [Nistér 03a] and RANSAC to estimate the pose with an optimization of the final poses and 3D points positions using the Levenberg-Marquardt algorithm. In order to be able to maintain consistency in long sequences of images they introduce a *local bundle adjustment*. The central idea is to estimate the relative poses only between what they call "key frames". The "key frames" are frames selected from the video sequences to be as far as possible from each other but with a minimum number of tracked features between them. They want to make sure that the motion between frames is sufficiently large to compute the epipolar geometry. The *local bundle adjustment* is carried out every time a new "key frame" is selected. This *local bundle adjustment* tries to estimate the extrinsic parameters for the last n cameras and the 3D points position taking into account the 2D re-projections in the N (with $N \geq n$) last frames (see Figure 2.6). The solution to the bundle adjustment is carried out taking advantage of the sparse nature of the Jacobian matrix of the error measure vector as described in [Hartley 04].

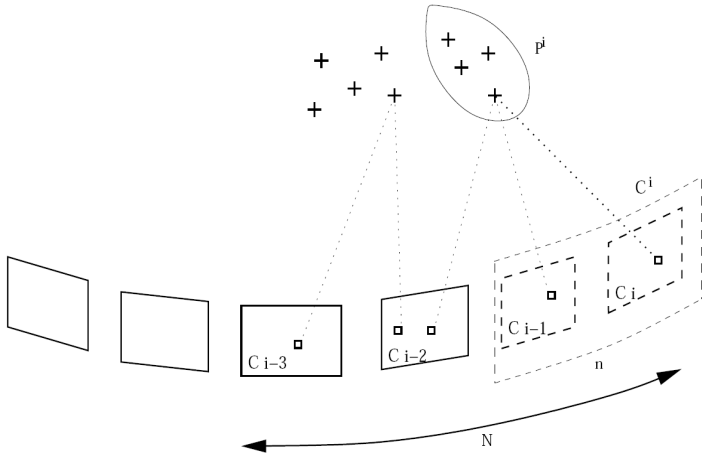


Figure 2.6: Local bundle adjustment when camera C_i is added. Only surrounded points and cameras are optimized. Nevertheless, re-projections in the last N images are taken into account [Mouragnon 06]

This work is inspired on a previous one of Nistér. In [Nistér 04b] they use Harris corners and Zero Mean Normalized Cross Correlation to track features over a certain number of frames. Then, they estimate the relative poses between three of the frames using the 5-point algorithm [Nistér 03a] and a preemptive RANSAC [Nistér 03b] followed by iterative refinement. To get the 3D positions of the points they triangulate the 3D points using [Oliensis 99] with another preemptive RANSAC. Each new camera pose added to the sequence is estimated with respect to the known 3D points using the 3-point algorithm [Haralick 94] and preemptive RANSAC. This last step is repeated until the error accumulation is considered to be high and then the whole algorithm started again in a very similar way as the *local bundle adjustment* keeps a "window" of the last N frames and the last n camera poses. Nister's results for the monocular system were reduced to a reconstructed trajectory for an aerial platform with no ground truth. The camera position in the aerial platform was pitched to the ground showing more optical flow than Mouragnon's system. This makes Mouragnon's results even more impressive.

Moragnon performed experiments using a Real Time Kinematics Differential GPS as ground truth in a 70 m long video sequence. The images used were 512×384 pixels at 7.5 fps. The results are very impressive with a mean deviation from the dGPS position of 27cm for the best configuration of the bundle adjustment parameters [$n = 4$ $N = 11$]. They also show partial results for a loop trajectory of the system in a car in urban environment. There is no dGPS ground truth available for the about 1Km run but the map results show an error in the loop closure in the order of meters (see Figure 2.7).

Closely related to visual odometry is what is known as Simultaneous Localization and Mapping (SLAM). SLAM builds or updates a map of an unknown environment while at the same time keeps track of the current location. SLAM has been traditionally restricted to the use of laser range-finders and short scale 2D maps. But in the last years, several works using standard cameras in outdoor environments has shown encouraging results [Karlsson 05] [Folkesson 05] [Lemaire 07]. A very good example of this kind of systems is [Clemente 07] where they use a Hierarchical Map approach [Estrada 05] and build the independent local maps in real-time using the EKF-SLAM technique and the inverse depth representation proposed in [Montiel 06]. However, the growing uncertainty in the estimations, makes the results for large outdoor environments still less accurate than

pure visual odometry systems and they need to revisit known places to give accurate estimations (see Figure 2.8).

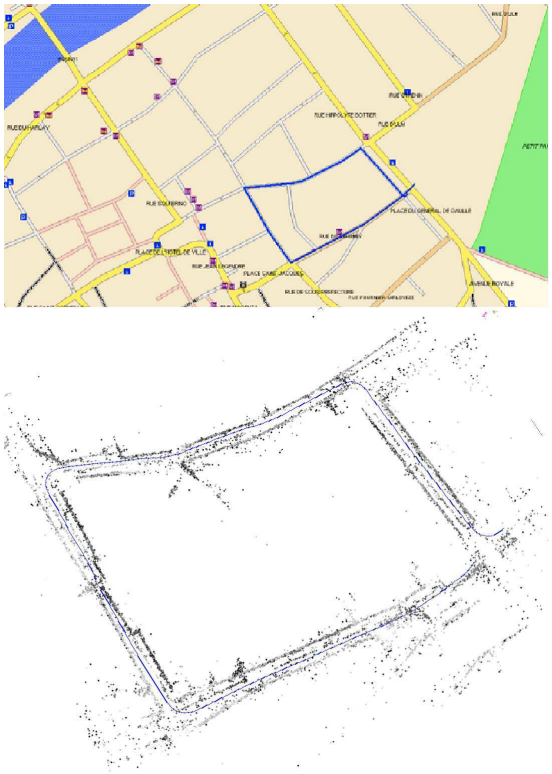


Figure 2.7: Urban sequence. Top map with the trajectory, bottom reconstruction [Mouragnon 06]

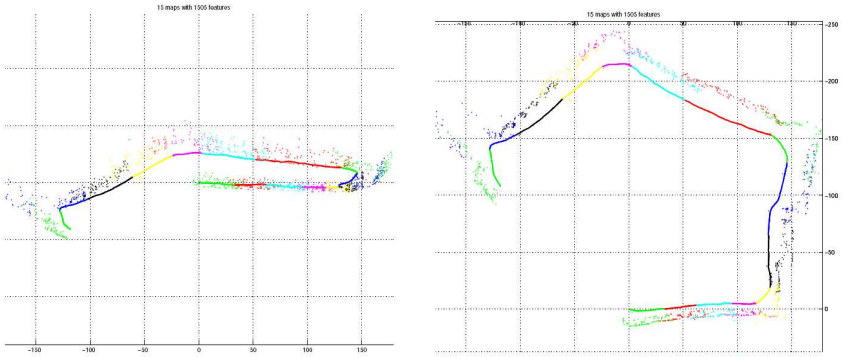


Figure 2.8: Hierarchical map auto-scaled, before loop detection. Side view (left). Top view (right) [Clemente 07]

An overview of the most significative monocular systems is shown on Table 2.1.

Table 2.1: Monocular systems

Approach	Sensor	Features	Motion Model	Outlier Rejection	Error Minimization	Test	Accuracy	Fusion
[Stein 00]	mono 320×240 50 ° FOV	direct method [Stein 97]	Pitch, Yaw, Forward translation	Maximum likelihood	Gradient Descend	Loop 100m	0.017 ° per frame	NO
[Nistér 04b] [Nistér 06]	mono 712×240 50 ° FOV yaw 10 °	Harris corners	6DOF	Preemptive RANSAC [Nistér 03b]	3-point algorithm [Nistér 04a] + iterative refinement	Off-road 600m	2% of the distance	IMU
[Mouragnon 06]	mono 512× 384 7.5 fps	Harris Corners	6DOF	5-point algorithm [Nistér 03a] + RANSAC	Local SBA	Off-road 70m (RTK GPS) Urban no GT	Off-road 27cm. Urban in the order of meters	NO
[Corke 04]	mono omnidirectional color	Lucas-Kanade	6DOF	Reject points with high residual after propagation	IEKF	Off-road 29.2m (GPS)	1% travelled distance. Filter degrades for long runs	NO
[Scaramuzza 08]	mono omnidirectional color 640×480	SIFT [Lowe 04]	3DOF Planar motion	Triggs [Triggs 98] Euclidean + RANSAC	NLLSQ+SVD	Urban 400m no GT	In the order of meters	NO
[Clemente 07]	mono 320×240 90 ° FOV	Shi-Tomassi	6DOF	Joint Compatibility [Neira 01]	EKF-SLAM	Off-road 250m no GT	Before loop closure in the order of tens of meters	NO

2.2 Stereo systems

When compared to monocular video, motion estimation from stereo images is relatively easy and tends to be more stable and well behaved [Nistér 04b]. Typically, the steps of the stereo ego-motion algorithms are:

1. Extract salient feature points in the image.

2. Match feature points between the left and right images of the stereo par and triangulate them to obtain 3D points.

3. Track these 3D points in time and obtain the rigid motion based on these tracked 3D points.

In practice, features correspondences contain mismatches which has to be detected. Outliers rejection methods such as RANSAC are usually employed at this step. The use of 3D point correspondences to obtain the motion suffers from a major drawback - triangulations are much more uncertain in the depth direction. Therefore these 3D points have non isotropic noise, and a 3D alignment between small sets of such 3D points gives poor motion estimates [Agrawal 05]. Different approaches have been proposed to solve this problem: In [Matei 99] the covariance for the 3D points is estimated and used to solve an heterocedastic, multivariate errors in variables regression problem. An alternative approach is to work in the disparity space [Demirdjian 01], a projective space with isotropic noise that can be used for efficiently estimating the motion of a calibrated stereo rig.

The system presented in [Agrawal 06] was based on this last approach. In this work they developed an inexpensive localization system using stereo vision and complementing it with a low-cost GPS and inertial sensors. Harris corners were detected and tracked in the stereo pairs and in time. Then, the motion was estimated using RANSAC-based scheme:

Table 2.2: Loop closure error in percentage [Agrawal 06]

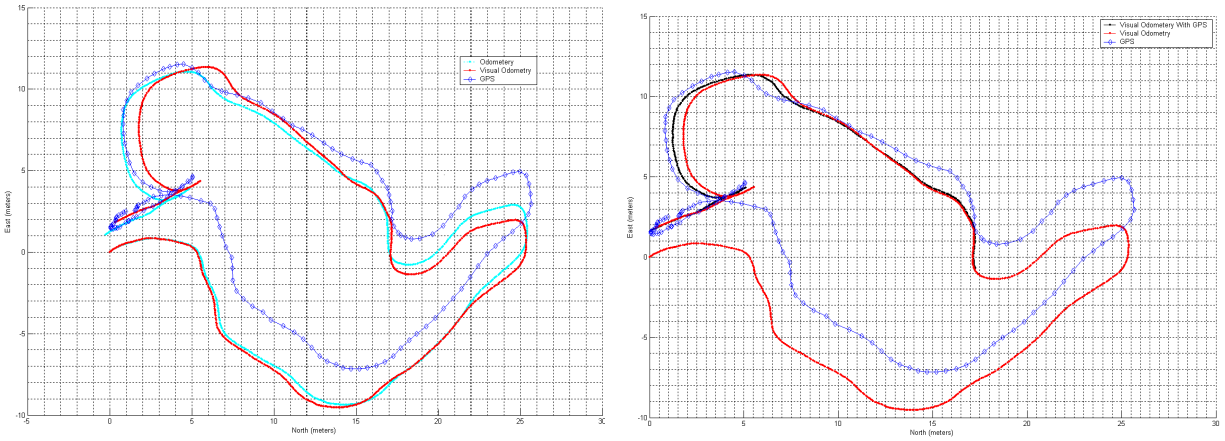
Run Number	1	2	3	4
Distance (meters)	82.4	141.6	55.3	51.0
Method	Percentage error			
Vehicle odometry	1.3	11.4	11.0	31.0
Raw visual odometry	2.2	4.8	5.0	3.9
Visual odometry & GPS	2.0	0.3	1.7	0.9

1. *Hypothesis generation* . Three 3D points were chosen spaced out well in the image to give a good estimation of the motion. Then the estimated rotation and translation is obtained through the singular value decomposition of the homography undergone by these points [Agrawal 05].

2. *Hypothesis scoring*. The obtained homography is then applied to the remaining points in the image space and their re-projection error is computed. A correspondence is taken as an inlier to this homography if the infinity norm of the error vector is less than 1.25 pixels. The number of inlier matches to a motion is taken as its score.

3. *Nonlinear minimization*. The steps above are applied for a fixed number of samples and the hypothesis with the best score is used as starting point for a nonlinear optimization (Levenberg-Marquardt for nonlinear least squares). The error function to be minimized is the re-projection error in the image coordinates for the inliers of the best hypothesis.

They performed experiments in outdoor localizations using a stereo rig mounted in a robot 0.5m above the ground, 100° FOV and 12cm of baseline. The experiments length was around 100 meters and the accuracy is estimated as a percentage of the error in the loop closure with respect to the length of the experiment (see Table 2.2).



(a) Raw odometry compared to raw visual odometry and GPS

(b) Visual odometry integrated with GPS

Figure 2.9: Trajectory estimation results from [Agrawal 06]

Using a similar approach to that of their monocular system Nistér presents a stereo visual odometry system in [Nistér 04b] and [Nistér 06]. The stereo system takes advantage of the known scale and perform triangulation followed by pose repeatedly. The whole process is as follows:

1. Match feature points from the stereo par and triangulate to get the 3D Reconstruction.
2. Track the points for a certain number of frames and compute the pose using preemptive RANSAC followed by iterative refinement. The 3-point algorithm [Haralick 94] is used as hypothesis generator. The scoring and iterative refinement are based on re-projection errors in both frames of the stereo pair.
3. Repeat step 2 a certain number of times.
4. Triangulate all new features. Repeat from step 2.
5. Re-triangulate all 3D points to reset error. Repeat from step 2.

Figure 2.10: Autonomous ground vehicle [Nistér 06]

As stated before, the triangulations are much more uncertain in the depth direction than a reconstruction in the disparity space. This is the reason for not using them in step 2 and only get triangulations after some iterations. To overcome this problem they use the 3-point algorithm for single camera pose. However this means that the pose is only based in one camera making any error in the calibration of the camera bias the triangulated positions of the 3D points. The frequency of triangulation of new features is a trade-off between a small error propagation (which requires frequent triangulations) and drift suppression (which requires working in the disparity space as long as possible).

They tested the system on different platforms ranging from autonomous ground vehicles to cars and hand-held or head mounted helmets 2.11. The stereo rig was equipped with a pair of synchronized analog cameras. Each camera had a horizontal FOV of 50°,

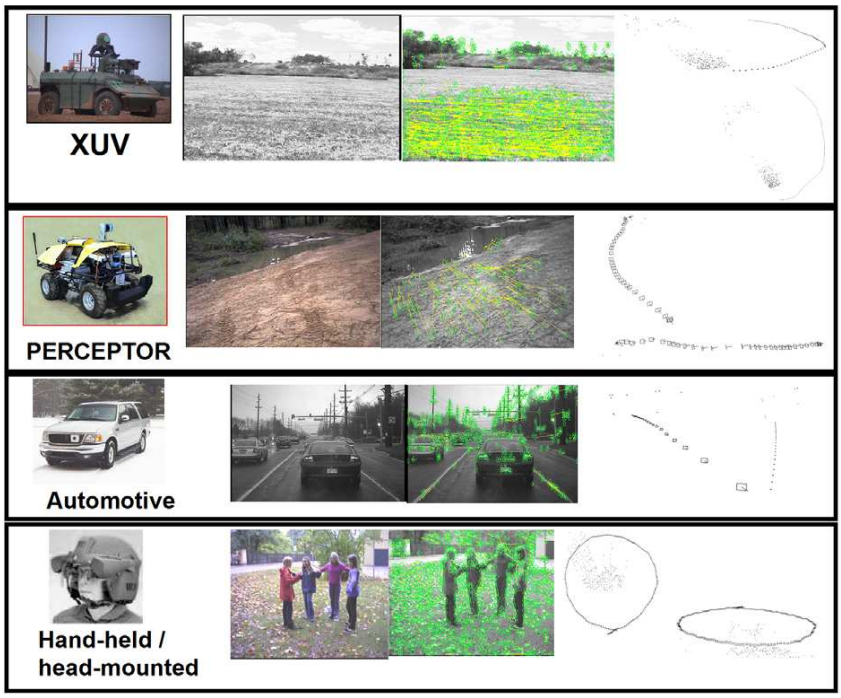


Figure 2.11: Various platforms used to test the visual odometry system in [Nistér 06]

Table 2.3: Metric accuracy of visual odometry estimates [Nistér 06]

Run	Frames	DGPS(m)	VisOdo(m)	% error
Loops	1602	185.88	183.9	1.07
Meadow	2263	266.16	269.77	1.36
Woods	2944	365.96	372.02	1.63

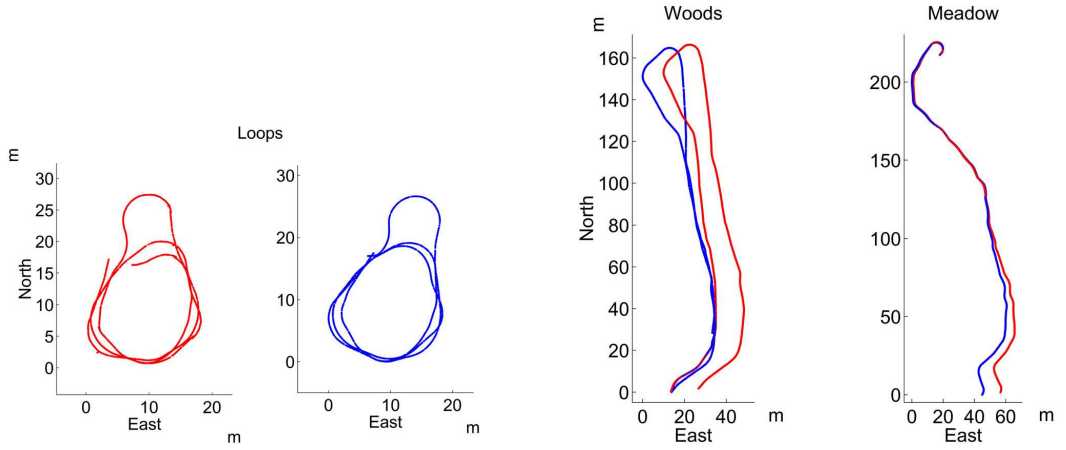
and 720×240 resolution. In the case of the autonomous ground vehicle the camera was tilted to the side of the vehicle about 10° and had a baseline of 28cm. This system was presented to DARPA as part of the program PerceptOR to test the visual odometry capability. The results show a high accuracy in the path lengths estimation of around 1% of the travelled path (see Table 2.3) and errors in the order of meters for loop closures.

It has to be pointed that they are working with very high resolution images and baseline, which increases the accuracy in the reconstructions. Also the features tracking is eased by the low velocity of the autonomous vehicle.

Another work involved in DARPA's project Learning Applied to Ground Robots was [Konolige 07]. The proposed system is similar to the work of [Mouragnon 06] and follows the line previously established by [Sunderhauf 05] for stereo visual odometry systems. Their main contribution is the introduction of a new, more stable feature named CenSurE [Agrawal 08] and the integration of an IMU to maintain global pose consistency. They also present results for a vehicle navigating over several kilometers of rough terrain. The proposed algorithm is as follows:

1. Extract features from the left image.

2. Perform dense stereo to get corresponding positions in the right image.



(a) Vehicle position estimated with visual odometry (left) and DGPS (right)

(b) Vehicle position estimated with visual odometry (left) and DGPS (right)

Figure 2.12: Trajectory estimation results from [Nistér 06]

3. Match to features in previous left image using ZMNCC.
4. Form consensus estimate of motion using RANSAC on three points.
5. Bundle adjust most recent N frames.
6. Fuse result with IMU data.

The tests sequences were recorded using cameras with 35° FOV, a baseline of 50cm and frame rate of 10 Hz (512×384). The length of the runs was 4 Km for the one named Ft Carson and 9 Km for the Little Bit. Results for both tests are shown in Figure 2.13 and compared to a RTK GPS. The high accuracy obtained can be observed, especially when the visual odometry is fused with the IMU.

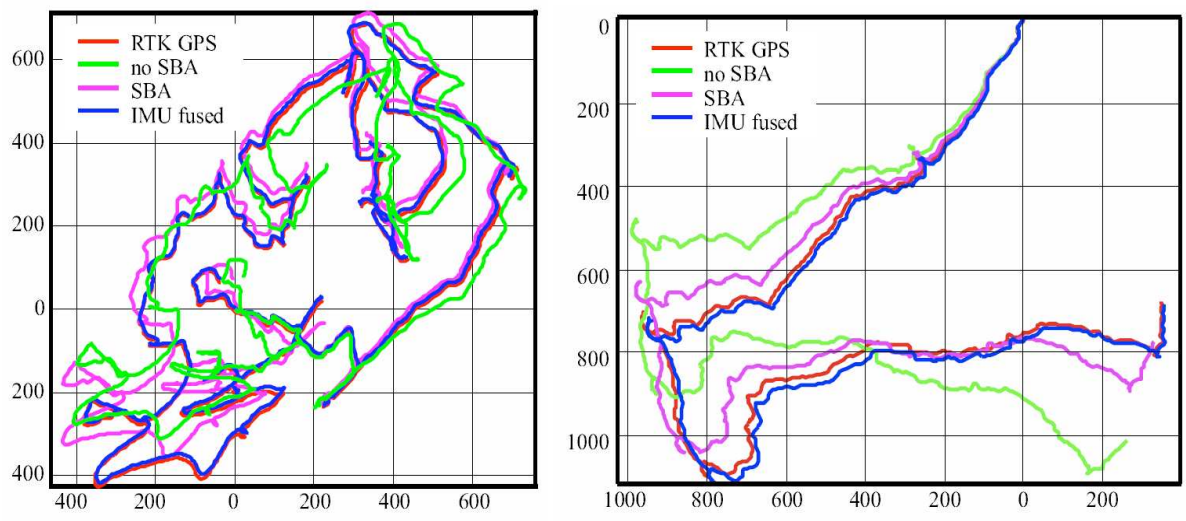


Figure 2.13: Trajectories from Little Bit (left) and Ft Carson (right) datasets [Konolige 07]

A stereo version of the work proposed in [Davison 03] has also been proposed in [Schleicher 09] but these methods based on a Kalman framework have not yet proven

capable of precise trajectory tracking over long distances, because the linearization demanded by the filter can lead to suboptimal estimates for motion. An overview of the most significative stereo systems is shown on Table 2.4

Table 2.4: Stereo systems

Approach	Sensor	Features	Motion Model	Outlier Rejection	Error Minimization	Test	Accuracy	Fusion
[Nistér 04b] [Nistér 06]	stereo 720×240 50 ° FOV	Harris corners	6DOF	3-point pose + Preemptive RANSAC [Nistér 04a]	BA	Off-road 600m	2% distance	Gyro
[Maomone 07]	stereo 640×480 80 ° FOV B=8.4cm	Harris corners	6DOF	Least squares + SVD + RANSAC	Maximum Likelihood + WNLSQ	Short paths	≈2% in position ≈ 5 ° in 29m.	NO
[Konolige 07]	stereo 512×384 35 ° FOV B=50cm	CenSurE [Agrawal 08]	6DOF	3 point + RANSAC	Incremental SBA	Off-road 4km and 9km (RTK GPS)	≈5%	IMU
[Agrawal 06]	stereo 640×480 100 ° FOV B=12cm	Harris corners	6DOF	Triangulation SVD + RANSAC	NLLSQ in disparity space (LM)	Off-road ≈100m (GPS)	2-5% raw VO <2% Fusion	GPS
[Paz 08a]	stereo 320×240 65 ° B=12cm	Dense	6DOF	Join Compatibility	D&C SLAM [Paz 08b]	Off-road 210m no GT	—	NO

2.3 Discussion

Previous sections have introduced a number of published methods for ego-motion estimation using a sequence of images. Several conclusions can be extracted from it:

- The use of a single camera have proven to deliver good accuracy for angular motion [Nistér 04b] and close estimations for the length of the path [Mouragnon 06], but they still have problems when there is no camera motion [Nistér 06].

- Omnidirectional cameras produce good optical flow with motion in any direction and features can be tracked for a higher number of frames [Scaramuzza 08]. Their results are comparable to those of the narrow FOV cameras, but the distortion introduced by the mirror makes the feature matching more complex and they are not as easily integrated with other system as conventional cameras.

- Stereo cameras produce the most accurate results for long runs up to date [Nistér 06] [Konolige 07].

- Triangulations are much more uncertain in the depth direction. A 3D alignment between small sets of such 3D points gives poor motion estimates [Agrawal 05]. Traditionally this has been solved working in the disparity space [Demirdjian 01], a projective space with isotropic noise. Another approach is to model the 3D uncertainty and introduce it in the model [Matei 99] [Montiel 06].

- The camera set up (resolution, orientation, baseline) strongly affects the quality of the estimated trajectory because the precision on the triangulations is also strongly affected by it. No study on the effect of the camera set up to the ego-motion estimation has been previously carried out.

- The algorithms that try to optimize the poses of the cameras and 3D points positions yield the more accurate results up to date [Mouragnon 06] [Konolige 07].
- Although a huge improvement has been shown by EKF-SLAM in the last years [Schleicher 09] [Paz 08a] they still have not reached the level of accuracy of visual odometry systems.
- There is not a general way of performance assessment. Some works use the loop closure distance; other overlay the trajectory on the map; some draw the GPS/RTK GPS trajectory.

2.4 Objectives

After the review of the state of the art, and considering the discussion presented in the introduction, the aims of this thesis are as follows:

1. To study the influence of the stereo configuration parameters (baseline, resolution, position, calibration) in the stereo reconstruction and in the reconstruction of motion trajectory using sequences of images. Find the critical parameters and get an idea of the maximum accuracy for a given set of parameters.
2. To study different feature extractors and its performance in complex urban environments. Find the strengths and weaknesses of different feature extractors and their maximum expectable performance for motion estimation systems.
3. To study the problem of motion estimation using a sequence of images for the specific case of a car. Differences and challenges that make this problem different from the traditional robotic platforms.
4. To develop a robust ego-motion estimation system for urban environments taking into account the previous conclusions. Perform experiments with different feature extractors and configurations to confirm the previous results and to get an idea of the maximum expectable accuracy.
5. To develop a map-matching algorithm for global localization using a digital map. Assess the performance of the proposed system for different configurations.

Chapter 3

Stereo Sensor Modelling and Calibration

The objective of a stereo sensor is to get an accurate tridimensional map of the scene using two simultaneously acquired images. By computing the displacement, or disparity, between two corresponding feature points in the left and right images, the 3D coordinates of the imaged point in the scene can be found. To do so two problems have to be solved. The first one is the *correspondence problem* which consists on determining what elements on the left and the right image are a projection of the same elements in the 3D scene. The second problem is the so-called *reconstruction problem*; the estimation of the 3D position of the matched elements using a previous knowledge of the cameras and the stereo geometry, acquired through an off-line calibration process.

Designing a stereo system involves choosing several parameters: the focal length of the cameras, the baseline distance, the frame rate and the distance of the cameras to the scene. Unfortunately, one must compromise to meet the conflicting requirements of accurate feature matching and accurate range estimation. In order to match feature points accurately and to avoid as much occlusion as possible the product, baseline×focal length, must be small. However, accurate range estimation requires this product to be large [Rodríguez 88].

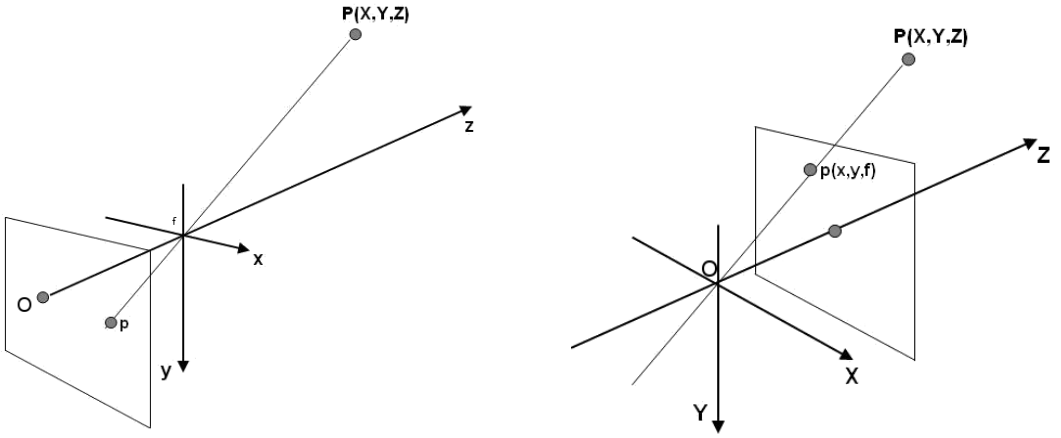
In this chapter the camera and stereo models of the stereo system are described for self-explanatory purposes. A more in-depth explanation of the models can be found on [Llorca 08]. Also, the stereo sensor calibration procedure and the influence of the different parameters on the final performance of the stereo sensor are discussed.

3.1 Camera Modelling

A digital image is a bidimensional array containing depth and intensity information. To model or to calibrate a camera means to find the mathematical relation between the 3D points in the scene and their 2D coordinates in the image plane. Three different types of parameters are involved in the image formation: *optical parameters* (lenses, focal distance, distortion, field of view, etc.), *photometric* (illumination intensity and direction, objects reflectance, etc.) and *geometrical* (projection, camera position and orientation, etc.). In this section a general overview of the problems and parameters of a stereo system is presented along with the minimum mathematical tools necessary to face the next chapters.

3.1.1 Perspective general model without distortion

The most common geometric model used to represent the cameras is the perspective model or *pin-hole* model. In this model all the rays go through a single point called optical centre O. The distance f from the optical centre to the image plane is called *focal distance* (see Figure 3.1) [Dhome 03].



(a) Pin-hole model

(b) Pin-hole model with the image plane before the optical centre

Figure 3.1: Pin-hole model

The relation between the image coordinates of a point (u, v) and its 3D position $[X_w\, Y_w\, Z_w]$ is usually expressed using two matrices, the intrinsic parameters matrix $\mathbf{M}_{int}$ and the external parameters matrix $\mathbf{M}_{ext}$. The general expression for the transformation of a 3D point into image coordinates in homogeneous coordinates is given by [Dhome 03]:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{3.1}$$

where $[u_o\, v_0\, dx\, dy]$ are the camera *intrinsic parameters* and $(r_{(11,\ldots,33)}, T_{(x,y,z)})$ are the *extrinsic parameters*. The two matrices are usually combined in a single matrix $\mathbf{M}_{(3\times4)}$:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \mathbf{M}_{int} \cdot \mathbf{M}_{ext} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \mathbf{M}_{(3\times4)} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{3.2}$$

3.1.2 Perspective general model with distortion

The *pin-hole* camera model assumes ideal lenses, but in reality lenses introduce deformations known as optical distortion. The distortion is produced when the rays going through the lens are deviated and intercept the image plane in positions further away from the ideal ones. This deviation is higher as long as the distance to the optical centre increases. Distortion is generally represented using a radial component do_r and a tangential component do_t. The radial component accounts for the distortion produced in radial lines from

the principal point while the tangential component accounts for the distortion produced in lines perpendicular to the radial lines [Dhome 03] (see Figure 3.2)

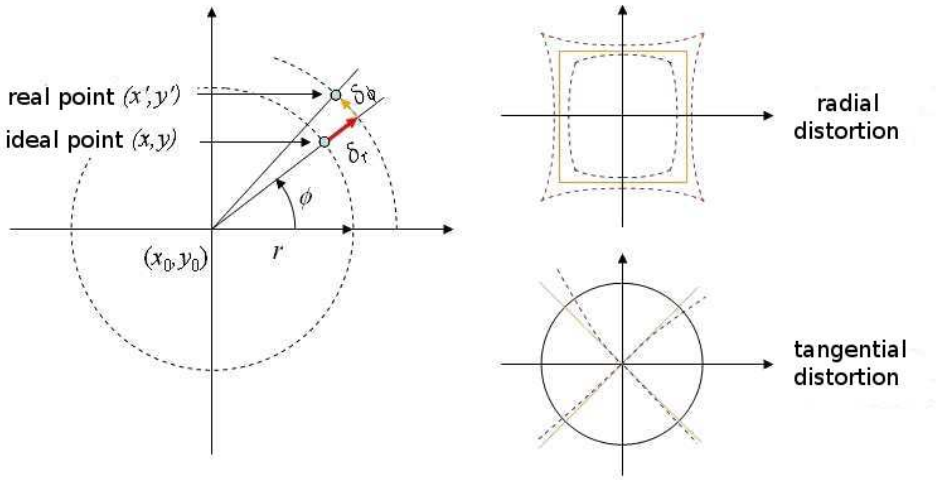


Figure 3.2: Effect of radial and tangential distortion.

The radial and tangential distortion are modelled using even-degree polynomials and introduced into the projection equations. For further details refer to [Dhome 03].

The correction is performed in two steps: rectification of the pixelic coordinates and interpolation of their intensity level. To speed up the process the rectification of the pixelic coordinates is pre-loaded in a look-up table. The correction of the distortion implies a loss of precision due to the interpolation, but it is necessary for the matching process between stereo pairs [Llorca 08] (see Figure 3.3).

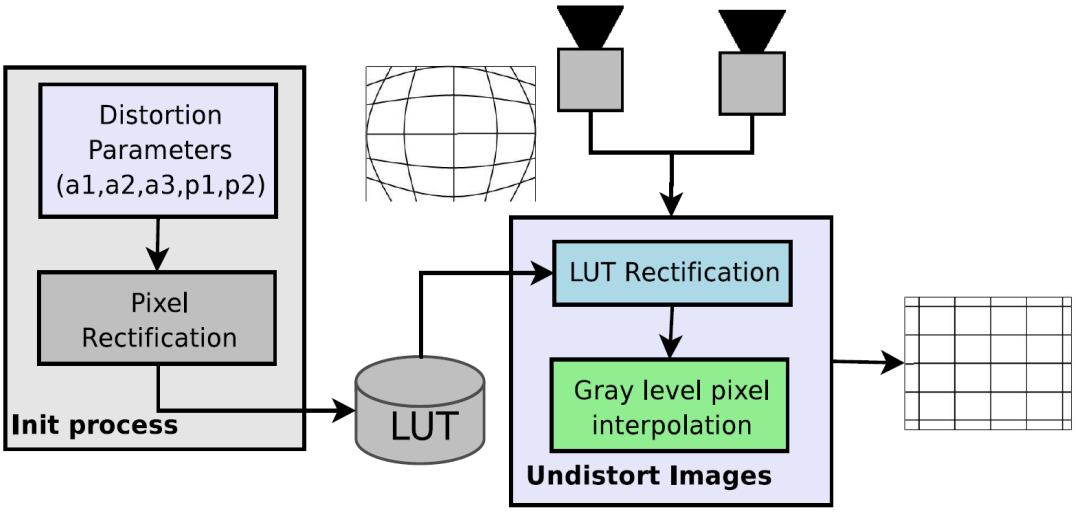
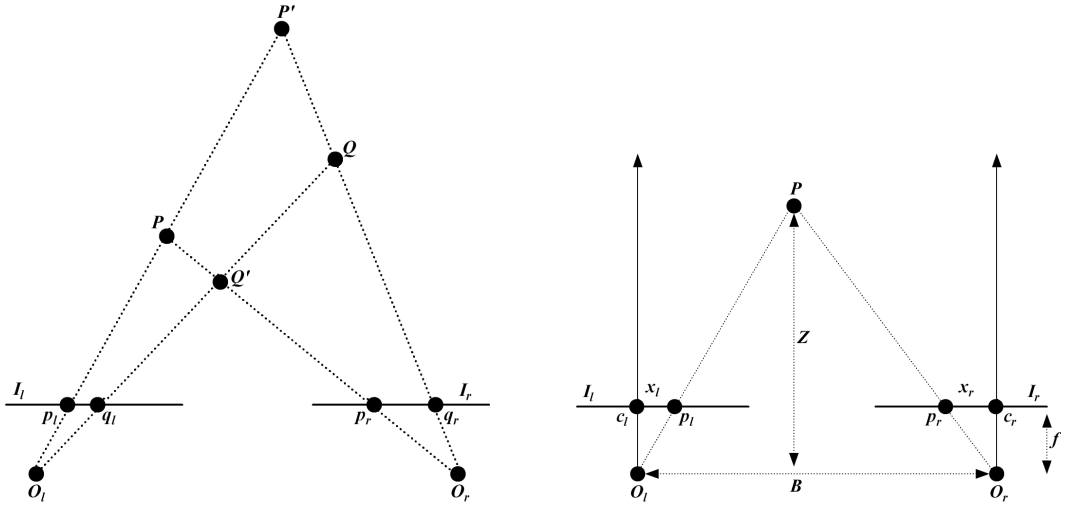


Figure 3.3: Distortion correction process [Llorca 08].

3.2 Stereo Sensor Modelling

3.2.1 Simplified model

In an ideal stereo system the two cameras are perfectly aligned, with their optical axis parallel to each other. The 3D position of a point P is obtained through triangulation [Trucco 98], looking for the intersection of the rays defined by the projection centres, O_l and O_r and the points (p_l, p_r) and (q_l, q_r) (Figure 3.4(a)). Triangulation strongly depends on the matching problem. If q_l and q_r are correctly matched the point Q will be reconstructed, but if the matched pair were incorrectly chosen, p_l and q_r, the reconstructed point would be P'. These makes the matching problem a very important one for an accurate 3D reconstruction of the scene.



(a) Ideal stereo system position estimation

(b) Ideal stereo system depth estimation from disparity

Figure 3.4: 3D position estimation by triangulation

Another important parameter for a stereo system is the separation between cameras or *baseline*. As shown in Figure 3.4(b) the depth of a 3D point can be expressed as:

$$\frac{B + x_l - x_r}{Z - f} = \frac{B}{Z} \rightarrow Z = f \cdot \frac{B}{d} \tag{3.3}$$

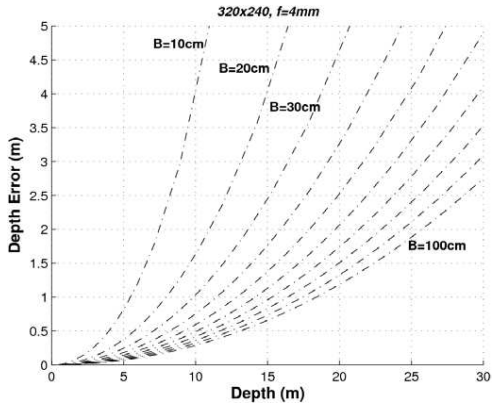
where $d = x_r - x_l$ is the *disparity*. Transforming Equation 3.3 from metric to pixelic coordinates:

$$Z = f \cdot \frac{B}{x_r - x_l} = \cdot \frac{B}{(u_r - u_0)d_x - (u_l - u_0)d_x} = \frac{f}{d_x} \cdot \frac{B}{(u_r - u_l)} = f_x \cdot \frac{B}{d_u} \tag{3.4}$$

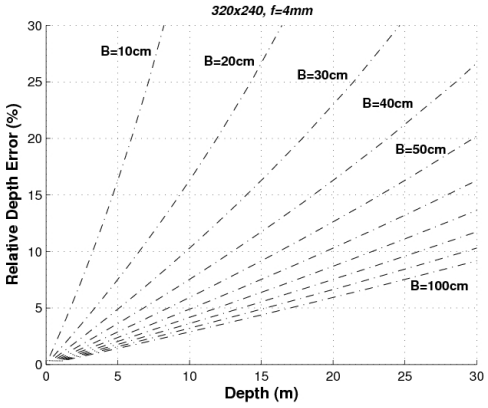
Even though Figure 3.4(b) is a simplified model useful conclusions about the range and precision of the stereo system can be extracted. Given the baseline B, the focal distance for x axis f_x and the resolution of the cameras (W, H) the 3D reconstruction accuracy from the maximum range (minimum disparity $d_u = 1$) to the minimum range (maximum disparity $d_u = W - 1$) can be obtained with the expression [Llorca 10]:

$$\Delta Z_i = Z_i - Z_{i-1} = f_x \cdot B \cdot \left(\frac{1}{d_{ui} - 1} - \frac{1}{d_{ui}} \right) = f_x \cdot B \cdot \frac{1}{d_{ui}^2 - d_{ui}} \tag{3.5}$$

A useful way of representing this information is the relative range error ($\Delta Z/Z$) (Figure 3.5(b)). I.e if we want a system with 320×240 resolution, f=4mm and relative range error $< 10\%$ up to distances of 20m then the baseline should be greater than 60cm.



(a) Depth accuracy for 320×240 images and f=4mm

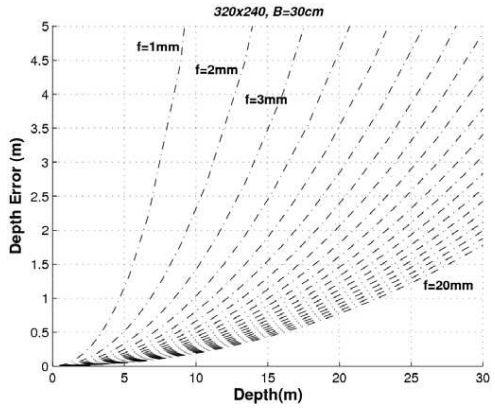


(b) Relative depth error for 320×240 images and f=4mm

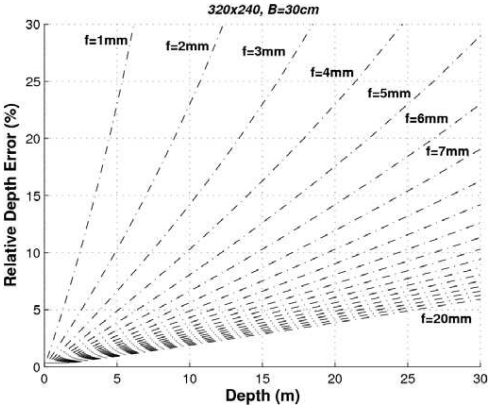
Figure 3.5: Relation between depth and depth accuracy for different baselines

As shown in figure 3.5 the precision in the 3D reconstruction decreases with depth. The higher the baseline the higher the precision in the depth reconstruction. However the higher the baseline the smaller the 3D projective space (the space covered by both cameras) and the higher the computational time for the correspondence process. The final design decision must be a trade-off between the dead zone, the computational time and the depth estimation precision.

In Figure 3.6 the absolute and relative depth error for different focal lengths are depicted.



(a) Depth accuracy for 320×240 images and different focal lengths



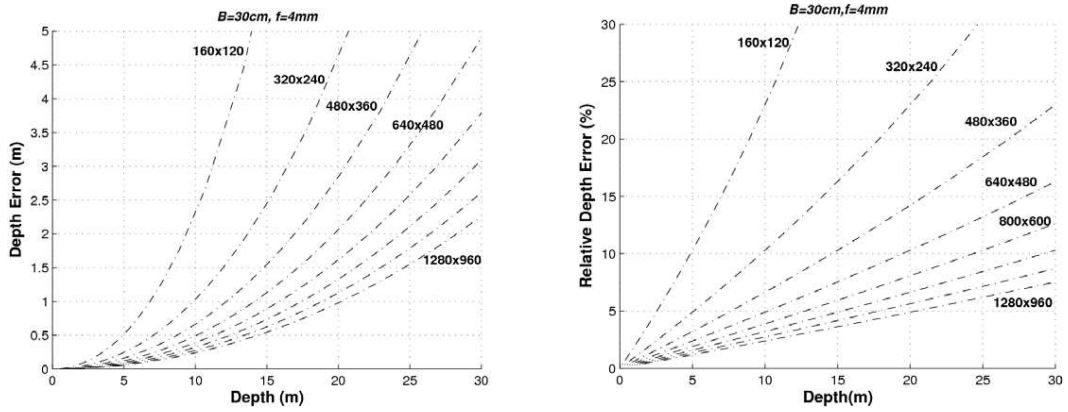
(b) Relative depth error for 320×240 images and different focal lengths

Figure 3.6: Relation between depth and depth accuracy for different focal lengths

The precision in the 3D reconstruction increases with focal length but once again, the

higher the focal length the smaller the 3D projective space.

Finally, Figure 3.8 shows both the absolute and the relative range errors for different image sizes corresponding to a sensor with f=4mm and a baseline of 300mm . As can be observed, the higher the image size the lower the error.



(a) Depth accuracy for B=400mm f=4mm and different image sizes

(b) Relative depth accuracy for B=400mm f= 4mm and different image sizes

Figure 3.7: Relation between depth and depth accuracy for different image resolutions

The previous graphs can be used for determining the system parameters according to the depth error requirements. However, other parameters have to be taken into account when designing a stereo sensor: the computational load (which is defined by the range of the disparity search space) and the size of the frontal blind zone. As long as the the baseline and the focal length increase, both the size of the frontal blind zone and the range of the disparity search space also increase. In addition, the higher the size of the images, the higher the disparity search space and thus, the computational load (see Figure 3.8) .

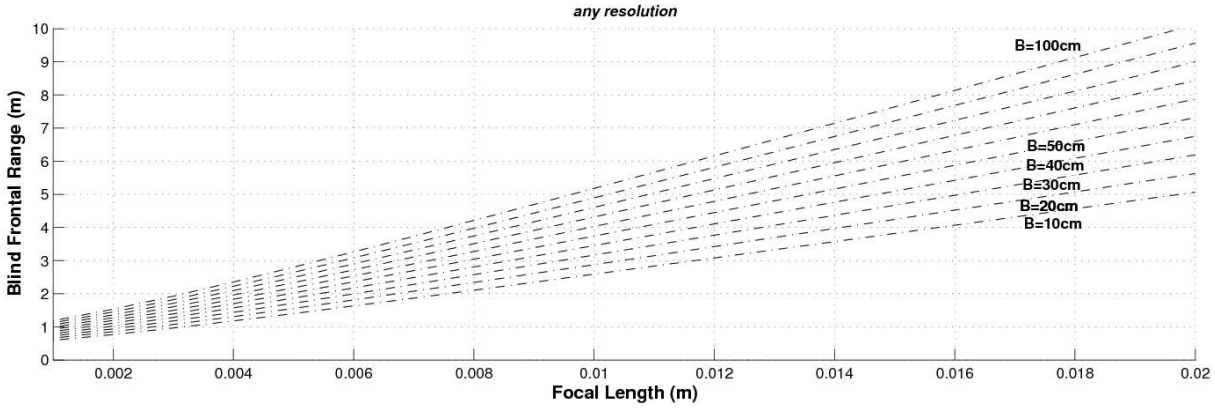
3.2.2 Epipolar Geometry

The epipolar geometry refers to the geometry of stereo vision when two cameras view a 3D scene from two distinct positions (Figure 3.9). There are a number of geometric relations between the 3D points and their projections onto the 2D images that lead to constraints between the image points. These relations are derived based on the assumption that the cameras can be approximated by the pinhole camera model [wikipedia 10a].

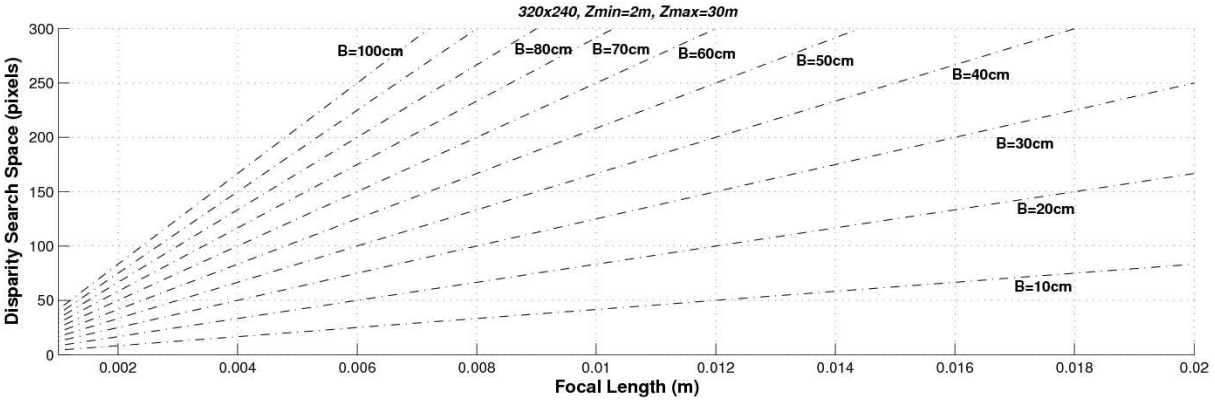
Using a single camera the 3D information from the scene can not be recovered. Another view of the 3D scene and the relation between the two views is needed to solve for the 3D positions of the points in the scene. The rigid transformation between the two views is given by a matrix $\mathbf{M}_{ext}^{S}$ consisting of a rotation and a translation. The transformation between the world system coordinates and the cameras are given by matrices $\mathbf{M}_{ext}$ and $\mathbf{M}'_{ext}$ as depicted in Figure 3.9. The relation between a point in the left camera coordinate system and the right camera coordinate system is given by:

$$\mathbf{P}_r = \mathbf{M}'_{ext}\mathbf{M}_{ext}^{-1}\mathbf{P}_l = \mathbf{M}_{ext}^{S}\mathbf{P}_l \tag{3.6}$$

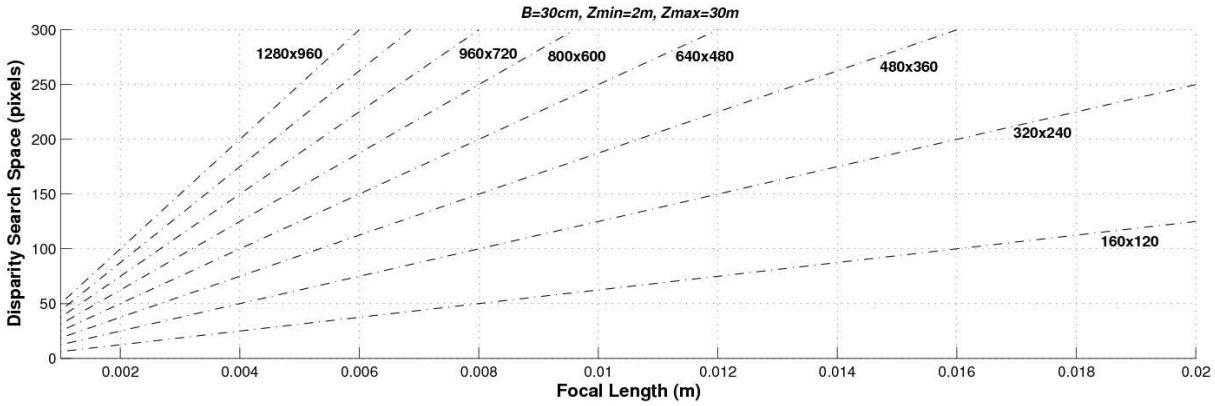
This relation is known as *epipolar geometry* and it implies that the projections $\mathbf{p}_l$ and $\mathbf{p}_r$ of the 3D point $\mathbf{P}$ fall in the lines l_l and l_r on their respective images. This restriction (*the epipolar restriction*) is widely used in stereo reconstruction problems because it constraints the search area in the *correspondence problem* (see Figure 3.10).



(a) Size of the blind frontal zone for different baselines



(b) Size of the disparity search space for different baselines



(c) Size of the disparity search space for different image resolutions

Figure 3.8: Negative effect of the depth accuracy increment on different parameters

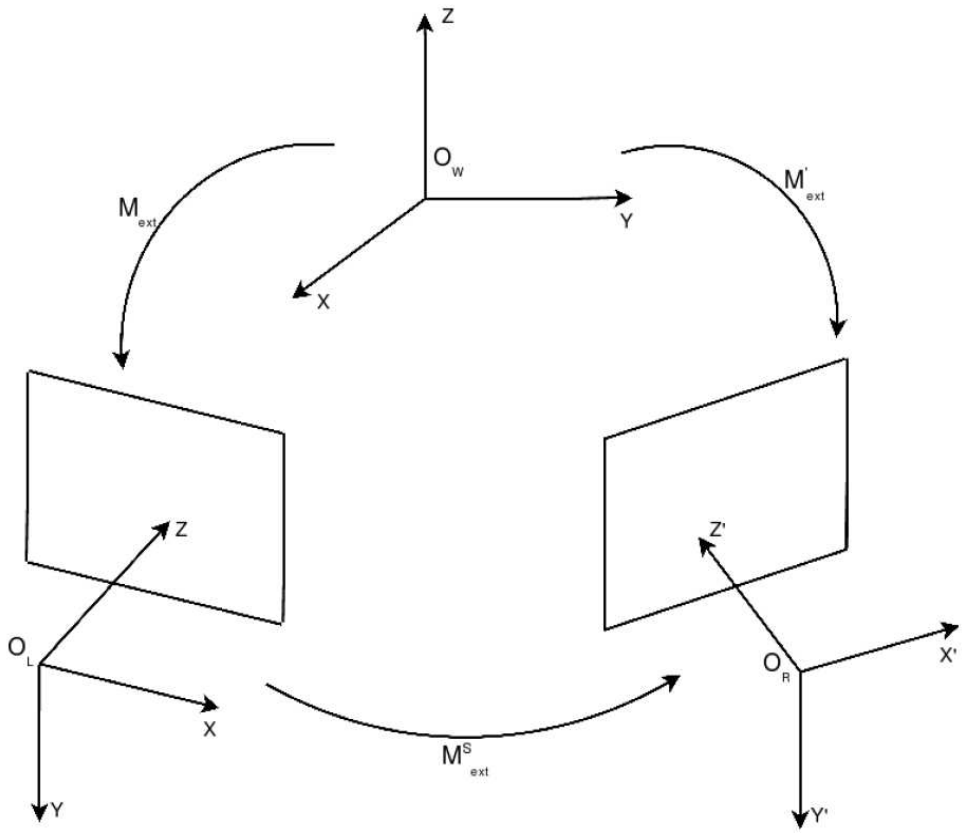


Figure 3.9: Stereo system geometry.

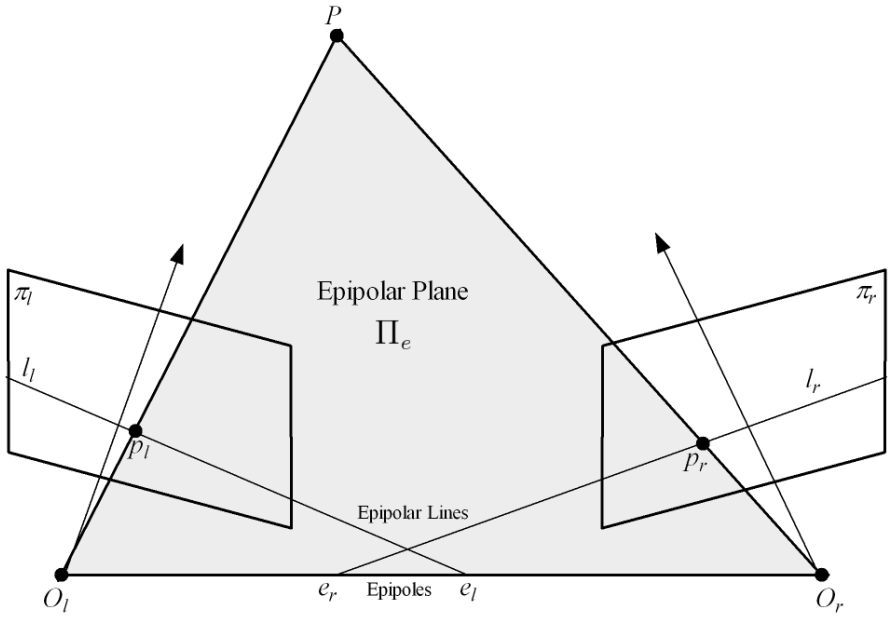


Figure 3.10: Epipolar geometry.

The equation of the line in the right image where the point $\mathbf{p}_l$ should be searched for is given by the *fundamental matrix*:

$$\begin{pmatrix} a'_r \\ b'_r \\ c'_r \end{pmatrix} = \mathbf{F} \begin{pmatrix} u_l \\ v_l \\ 1 \end{pmatrix} \tag{3.7}$$

where $\mathbf{F} = (\mathbf{M}_{ext}^{-1})^T \mathbf{E} \mathbf{M}_{ext}'^{-1}$ and $\mathbf{E}$ is the essential matrix. For further details please refer to [Llorca 08].

3.2.3 Cameras calibration

To calibrate a camera means to find the mathematical relation between the 3D points in the scene and their 2D coordinates in the image plane. As shown in sections 3.2 and 3.1 this relation is described by the intrinsic parameters for each one of the cameras and the extrinsic parameters which describe the rigid transformation between the optical centres of the cameras.

In this thesis an off-line supervised calibration process has been performed as described in [Llorca 08] using the *Camera Calibration Toolbox for MatLab* [MatLab 07].

3.2.4 3D reconstruction

Assuming we have solved the *correspondence problem* and that the geometric relation between cameras is known through a calibration process we can get the 3D position of a point. These algorithms are known as triangulation algorithms. The most used one tries to find the intersecting rays from the 3D points to the optical centres of the cameras. In practice, both rays don't intersect and the problem is solved by obtaining the middle point of the perpendicular segment to both rays [Xu 96] [Trucco 98] [Forsyth 03]. But, as showed by [Hartley 03], it is inappropriate to used this kind of solutions in a projective space because the concepts of distance, perpendicularity, etc. don't apply.

Here we have used the space invariant triangulation method explained in [Llorca 08]. The equation for the estimation of the 3D position is an overdetermined lineal system given by:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} X_l \\ Y_l \\ Z_l \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \rightarrow \mathbf{A} \cdot \mathbf{P}_l = \mathbf{b} \tag{3.8}$$

where

$$\begin{cases} a_{11} = f_{xl} \\ a_{12} = 0 \\ a_{13} = -(u_l - u_{ol}) \\ b_1 = 0 \\ a_{21} = 0 \\ a_{22} = f_{yl} \\ a_{23} = -(v_l - v_{ol}) \\ b_2 = 0 \\ a_{31} = r_{31} \cdot (u_r - u_{or}) - f_{xr} \cdot r_{11} \\ a_{32} = r_{32} \cdot (u_r - u_{or}) - f_{xr} \cdot r_{12} \\ a_{33} = r_{33} \cdot (u_r - u_{or}) - f_{xr} \cdot r_{13} \\ b_3 = f_{xr} \cdot t_x - t_z \cdot (u_r - u_{or}) \\ a_{41} = r_{31} \cdot (v_r - v_{or}) - f_{yr} \cdot r_{21} \\ a_{42} = r_{32} \cdot (v_r - v_{or}) - f_{yr} \cdot r_{22} \\ a_{43} = r_{33} \cdot (v_r - v_{or}) - f_{yr} \cdot r_{23} \\ b_4 = f_{yr} \cdot t_y - t_z \cdot (v_r - v_{or}) \end{cases} \tag{3.9}$$

The system in 3.8 is and overdetermined linear system which is solved using least squares, that is:

$$\mathbf{P}_l = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{3.10}$$

where P_l is the 3D coordinates of point P on the left camera coordinate system.

3.2.5 Uncertainty in 3D estimation

Given the discrete nature of the imaging system, the image coordinates of each pixel can suffer from quantization errors of up to $\pm$ 1/2 pixel. Because of this quantization error, the estimation of the range Z, is also inexact. The estimated values of X and Y suffer from quantization errors as well. However as discussed in [Blostein 87] and [Solina 85] the error in the estimation of Z dominates.

A simplified example for the case of 2D points projecting onto one dimensional images is shown in Figure 3.11. The marks on the image plane denote pixel boundaries, and the radiating lines extend these boundaries into space. Given the projection of point P onto the left and right images the estimated position of P can lie anywhere in the shaded region surrounding the true location [Solina 85]. We want to take this uncertainty into account in any reasoning based on measurements of P.

Different approaches has been used to modeling such uncertainty. For example [Baird 85] used *tolerance regions* in finding the transformation between a two-dimensional set of model points and their measured image positions. Uncertainty was represented with convex polygons surrounding the measured point locations, and the transformed model points were required to lie within these polygons. In [Moravec 80] *scalar weights* which grow with distance are used, so it can be modeled the increase in uncertainty inversely with distance. However, as shown in Figure 3.11, the uncertainty induced by triangulation is not a simple scalar function of distance to the point; it is also skewed and oriented. Nearby points have a fairly compact uncertainty, whereas distant points have a more elongated uncertainty

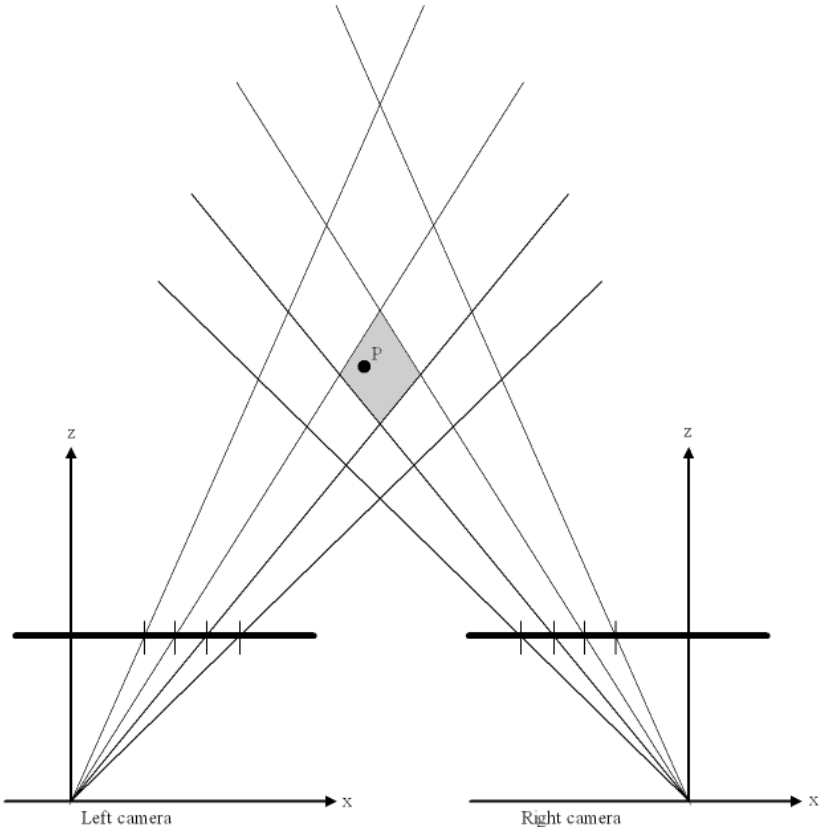


Figure 3.11: Possible positions for a pixel 2D reconstruction.

that is roughly aligned with the line of sight to the point. Scalar errors do not capture these distinction in shape.

Our approach is to assume 2D normally distributed (i.e. Gaussian) error in the measured image coordinates and to derive 3D Gaussian distributions describing the error in the estimated 3D coordinates. Similar approaches has been used in photogrammetry [Slama 80] and in computer vision [Broida 86] [Gennery 80]. For the 3D coordinates, the true distribution will be non-Gaussian, because triangulation is a non-linear operation; we approximate this as a Gaussian for simplicity and because it gives an adequate approximation when the distance to points is not extreme [Mathies 87].

Given a calibrated rig of cameras and a correspondence between two points, one on the left camera (u_l, v_l) and another one on the right (u_r, v_r) the 3D position of a point $\mathbf{P} = [x\, y\, z]$ in the world coordinate system is given by (3.10) where $\mathbf{A}$ is the matrix containing the equations for the 3D to 2D transformation for each one of the cameras and $\mathbf{b}$ the independent term of the same equations. Matrices $\mathbf{A}$ and $\mathbf{b}$ are written as a function of the cameras intrinsic parameters and the image coordinates of the matched feature.

$$\mathbf{A} = \begin{pmatrix} u_l \cdot m_{31}^L - m_{11}^L & u_l \cdot m_{32}^L - m_{12}^L & u_l \cdot m_{33}^L - m_{13}^L \\ v_l \cdot m_{31}^L - m_{21}^L & v_l \cdot m_{32}^L - m_{22}^L & v_l \cdot m_{33}^L - m_{23}^L \\ u_r \cdot m_{31}^R - m_{11}^R & u_r \cdot m_{32}^R - m_{12}^R & u_r \cdot m_{33}^R - m_{13}^R \\ v_r \cdot m_{31}^R - m_{21}^R & v_r \cdot m_{32}^R - m_{22}^R & v_r \cdot m_{33}^R - m_{23}^R \end{pmatrix} \tag{3.11}$$

$$\mathbf{b} = \begin{pmatrix} m_{14}^L - u_l \cdot m_{34}^L \\ m_{24}^L - v_l \cdot m_{34}^L \\ m_{14}^R - u_r \cdot m_{34}^R \\ m_{24}^R - v_r \cdot m_{34}^R \end{pmatrix} \tag{3.12}$$

Each camera intrinsic parameters $[\mathbf{M}^L\mathbf{M}^R]$ are estimated using an off-line calibration process. The intrinsic parameters describe the 3D to 2D transformation for each one of the cameras. In order to compute the uncertainty in the 3D reconstruction in the partial derivatives with respect to $\mathbf{T} = (u_l\, v_l\, u_r\, v_r)$ for equation 3.8 are computed

$$\frac{\partial(\mathbf{A} \cdot \mathbf{P})}{\partial \mathbf{T}} = \frac{\partial \mathbf{b}}{\partial \mathbf{T}} \tag{3.13}$$

Applying the product rule for matrices

$$\mathbf{P}^T \frac{\partial \mathbf{A}^T}{\partial \mathbf{T}} + \mathbf{A}\frac{\partial \mathbf{P}}{\partial \mathbf{T}} = \frac{\partial \mathbf{b}}{\partial \mathbf{T}} \rightarrow \mathbf{A}\frac{\partial \mathbf{P}}{\partial \mathbf{T}} = \frac{\partial \mathbf{b}}{\partial \mathbf{T}} - \mathbf{P}^T \frac{\partial \mathbf{A}^T}{\partial \mathbf{T}} \tag{3.14}$$

the expression for the uncertainty in the 3D position is obtained:

$$\mathbf{A} \cdot \frac{\partial \mathbf{P}}{\partial \mathbf{T}} = \mathbf{C} \rightarrow \frac{\partial \mathbf{P}}{\partial \mathbf{T}} = \left(\mathbf{A}^T \cdot \mathbf{A}\right)^{-1} \cdot \mathbf{A}^T \cdot \mathbf{C} \tag{3.15}$$

where $\mathbf{C}$ is

$$\mathbf{C} = \frac{\partial \mathbf{b}}{\partial \mathbf{T}} - \mathbf{P}^T \frac{\partial \mathbf{A}^T}{\partial \mathbf{T}} \tag{3.16}$$

Solving the partial derivatives for (3.16) we get

$$\mathbf{C} = \frac{\partial \mathbf{b}}{\partial \mathbf{T}} - \mathbf{P}^T \frac{\partial \mathbf{A}^T}{\partial \mathbf{T}} = I_{4\times 4} \cdot \begin{pmatrix} -m_{34}^L - m_{31}^L \cdot X - m_{32}^L \cdot Y - m_{33}^L \cdot Z \\ -m_{34}^L - m_{31}^L \cdot X - m_{32}^L \cdot Y - m_{33}^L \cdot Z \\ -m_{34}^R - m_{31}^R \cdot X - m_{32}^R \cdot Y - m_{33}^R \cdot Z \\ -m_{34}^R - m_{31}^R \cdot X - m_{32}^R \cdot Y - m_{33}^R \cdot Z \end{pmatrix} \tag{3.17}$$

Finally, substituting the intrinsic matrices values we get an expression for $\mathbf{C}$ (note that the intrinsic calibration matrices $\mathbf{M}^L$ and $\mathbf{M}^R$ are sparse and as a consequence $\mathbf{A}$ and $\mathbf{b}$ are also sparse)

$$\mathbf{C} = \begin{pmatrix} -z & 0 & 0 & 0 \\ 0 & -z & 0 & 0 \\ 0 & 0 & -z & 0 \\ 0 & 0 & 0 & -z \end{pmatrix} \tag{3.18}$$

Assuming $\mathbf{T}$ is a normally distributed random variable with mean 0 and variance:

$$\Sigma_T = \begin{pmatrix} \sigma_{u_l}^2 & 0 & 0 & 0 \\ 0 & \sigma_{v_l}^2 & 0 & 0 \\ 0 & 0 & \sigma_{u_r}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v_r}^2 \end{pmatrix} \tag{3.19}$$

where $\sigma^2_{u_l}, \sigma^2_{v_l}, \sigma^2_{u_r}, \sigma^2_{v_r}$ are the uncertainties in pixels on the measure of $\mathbf{T}$, the final expression for the quantization error covariance is (the errors in the images coordinates are assumed to be independent so the covariance matrix is diagonal):

$$cov\left(\frac{\partial \mathbf{P}}{\partial \mathbf{T}} \cdot \mathbf{T}\right) = E\left[\left(\frac{\partial \mathbf{P}}{\partial \mathbf{T}} \cdot \mathbf{T}\right)\left(\frac{\partial \mathbf{P}}{\partial \mathbf{T}} \cdot \mathbf{T}\right)^T\right] = \frac{\partial \mathbf{P}}{\partial \mathbf{T}} \cdot E[\mathbf{T}^2] \cdot \left(\frac{\partial \mathbf{P}}{\partial \mathbf{T}}\right)^T \rightarrow$$
$$cov\left(\frac{\partial \mathbf{P}}{\partial \mathbf{T}} \cdot \mathbf{T}\right) = \frac{\partial \mathbf{P}}{\partial \mathbf{T}} \cdot \Sigma_T \cdot \left(\frac{\partial \mathbf{P}}{\partial \mathbf{T}}\right)^T = \begin{pmatrix} \Delta_x & 0 & 0 \\ 0 & \Delta_y & 0 \\ 0 & 0 & \Delta_z \end{pmatrix} \tag{3.20}$$

In Figure 3.12 a illustration of the geometrical meaning of this uncertainty model is shown. The ellipse represents the contour of the error model and the diamond represents the quantization error of Figure 3.12. For nearby points the contours will be close to spherical; the further the points the more eccentric they become. Where the Gaussian approximation breaks down is in failing to represent the longer tails of the true error distribution. The true distribution is skewed while normal distributions are symmetric. The skew is not significant when points are close, but becomes more pronounced the more distant the points. A possible consequence is biased estimation of point locations, which may lead to biased motion estimates.

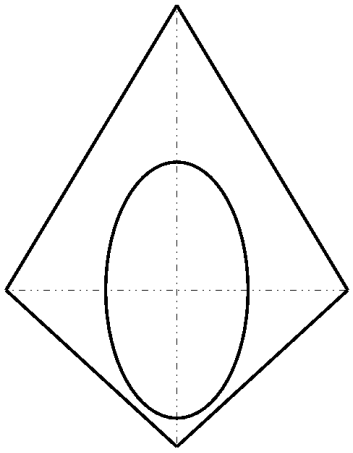


Figure 3.12: Uncertainty in the 2D position of a reconstructed pixel.

3.3 Conclusions

In this chapter the camera model and the stereo geometry have been studied and the influence of the different parameters analyzed. The stereo reconstruction uncertainty has been described and a multivariate Gaussian model has been proposed to describe it. The main conclusions of this chapter are:

- An increase in the resolution, base-line or focal length of the cameras improves the 3D reconstruction accuracy. However this brings other drawbacks as a reduction of the space covered by both cameras, an increment of the blind zone in front of the cameras and a higher amount of data to be processed. The final parameters

must be a trade-off between the dead zone, the computational time and the depth estimation precision. Also the integration with other systems may condition some of these parameters.

- The uncertainty induced by triangulation is not a simple scalar function of distance to the point; it is also skewed and oriented. Nearby points have a fairly compact uncertainty, whereas distant points have a more elongated uncertainty that is roughly aligned with the line of sight to the point. Scalar errors do not capture these distinction in shape.
- A multivariate Gaussian error model has been proposed for the uncertainty in the 3D position. The true distribution will be non-Gaussian, because triangulation is a non-linear operation; we approximate this as a Gaussian for simplicity and because it gives an adequate approximation when the distance to points is not extreme.

Chapter 4

Visual Odometry

Visual odometry consists on determining a camera (or cameras) position and orientation using a sequence of images. It's called *odometry* because of its analogies with the classical encoder sensors in robotics. When the camera (or cameras) are mounted on a vehicle this technique is also known as *ego-motion* estimation because the cameras are moving with the vehicle and the camera's motion is the vehicle's one.

In this chapter, a whole new approach for ego-motion computing in complex urban environments based on stereo-vision is proposed. The specific problems of urban environments and the vehicle dynamics are analyzed and new solutions are proposed. The use of stereo-vision has the advantage of disambiguating the 3D position of detected features in the scene at a given frame. Based on that, feature points are matched between pairs of frames and linked into 3D trajectories. The solution of the non-linear system equations describing the vehicle motion at each frame is computed under the non-linear, photogrametric approach using RANSAC. The use of RANSAC [Nistér 04b] allows for outliers rejection in 2D images corresponding to real traffic scenes, providing a method for carrying out visual odometry on-board a road vehicle. A flow diagram of the proposed method is shown in Figure 4.1.

The rest of the chapter is organized as follows: in section 4.1 the feature detection and matching problem is presented and three different feature extractors are evaluated; section 4.2 provides a description of the proposed non-linear method for estimating the vehicle's ego-motion and the 3D vehicle trajectory; implementation and results are provided in section 4.2.6.

4.1 Features Detection and Matching

In most previous research on visual odometry, features are used for establishing correspondences between consecutive frames in a video sequence. Some of the most common choices are Harris corner detector [Hariis 88] and the Kanade-Lucas-Tomasi detector (KLT)[Lucas 81a]. Harris corners have been found to yield detections that are relatively stable under small to moderate image distortions [Schmid 00]. As stated in [Nistér 04b], distortions between consecutive frames can be regarded as fairly small when using video input. However, Harris corners are not always the best choice for landmark matching when the environment is cluttered and repetitive superimposed objects appear on the images. This is the situation for urban visual odometry systems. Although Harris corners can yield distinctive features, they are not always the best candidates for stereo and temporal matching. Moreover, the changing illumination conditions in urban environment can dra-

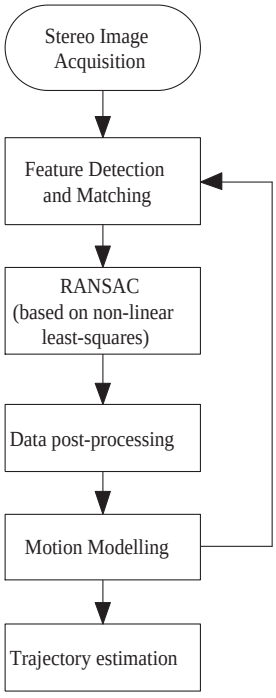


Figure 4.1: General layout of the visual odometry method based on RANSAC.

matically decrease the number of Harris features that can be detected and tracked. The use of Harris corners has yet another disadvantage; the correspondence problem has to be solved and it can introduce new mismatches. Among the wide spectrum of matching techniques that can be used to solve the correspondence problem, the *Zero Mean Normalized Cross Correlation* [Boufama 94] is usually chosen for robustness reasons. The Zero Mean Normalized Cross Correlation between two image windows can be computed as follows

$$\mathrm{ZMNCC}(p,p') = \frac{\sum_{i=-n}^{n}\sum_{j=-n}^{n} A \cdot B}{\sqrt{\sum_{i=-n}^{n}\sum_{j=-n}^{n} A^2 \sum_{i=-n}^{n}\sum_{j=-n}^{n} B^2}} \tag{4.1}$$

where A and B are defined by

$$A = \left(I\left(x+i, y+j\right) - \overline{I(x,y)}\right) \tag{4.2}$$

$$B = \left(I'\left(x'+i, y'+j\right) - \overline{I'(x',y')}\right) \tag{4.3}$$

where $I(x,y)$ is the intensity level of pixel with coordinates (x,y), and $\overline{I(x,y)}$ is the average intensity of a $(2n+1)$x$(2n+1)$ window centered around that point. As the window size decreases, the discriminatory power of the area-based criterion gets decreased and some local maxima appear in the searching regions. On the contrary, an increase in the window size causes the performance to degrade due to occlusion regions and smoothing of

disparity values across boundaries. In order to minimize the number of outliers, a mutual consistency check is usually employed (as described in [Nistér 04b]). Accordingly, only pairs of features that yield mutual matching are accepted as a valid match. The accepted matches are used both in 3D feature detection (based on stereo images) and in feature tracking (between consecutive frames).

In urban cluttered environments repetitive patterns such as zebra crossings, building windows, fences, etc. can be found. In Fig. 4.2 the typical correlation response along the epipolar line for a repetitive pattern is shown. Multiple maxima or even higher responses for badly matched points are frequent. Although some of these correlation mistakes can be detected using the mutual consistency check or the unique maximum criterion [Llorca 08], the input data for the ego-motion estimation will be regularly corrupted by these outliers which will decrease the accuracy of the estimation. Moreover, superimposed objects limit observed from different viewpoints are a source of correlation errors for the system. Fig. 4.3 depicts a typical example of an urban environment in which a car's bonnet is superimposed on the image of the next car's license plate and bumper. As can be seen in Fig. 4.3(a), the Harris corner extractor chooses, as feature points, the conjuncture in the image between the car's bonnet and the next car's license plate and bumper. In the image plane these are, apparently, good features to track, but the different depths of the superimposed objects will cause a misdetection due to the different viewpoints. In Fig. 4.3(b) and 4.3(c) it can be seen how the conjuncture in the image between the number 1 on the license plate and the bonnet is matched but they do not correspond to the same point in the 3D space. We can see the same kind of misdetection in the conjuncture between the car's bonnet and the bumper. The error in the 3D reconstruction of these points is not big enough to be rejected by the RANSAC algorithm so they will corrupt the final solution. In practice, these errors lead to local minima in the solution space and thus to inaccurate and unstable estimations. A more reliable matching technique is needed in order to cope with the complexity of urban environments.

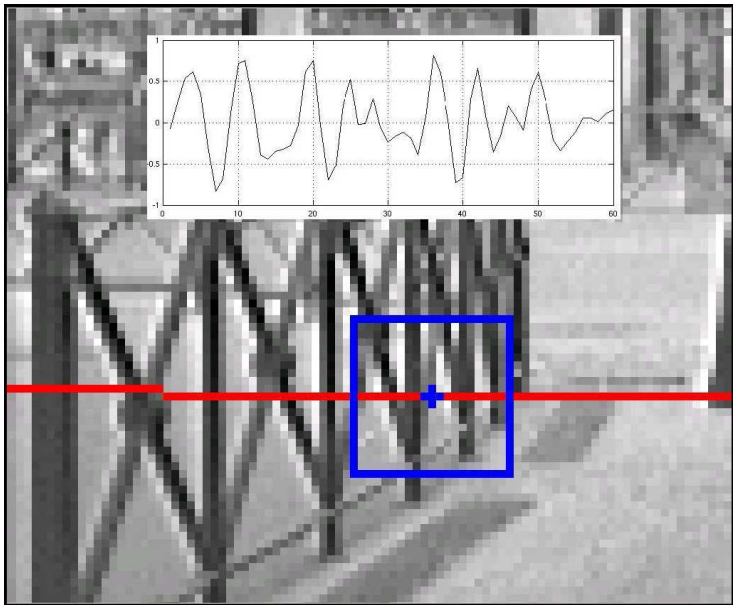
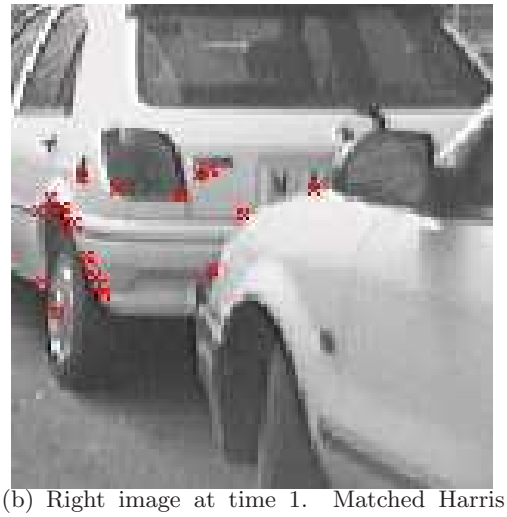


Figure 4.2: Correlation response along the epipolar line for a repetitive pattern.

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



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

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

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

Figure 4.3: Examples of matches for superimposed objects.

4.1.1 SIFT based features detection and tracking

In this system we propose a similar approach to [Se 01], in which scale-invariant image features are used for Simultaneous Localization And Map Building (SLAMB) in unmodified (no artificial landmarks) dynamic environments. To do so, they used a trinocular stereo system [Murray 98] to estimate the 3D position of the landmarks and to build a 3D map where the robot can be localized simultaneously. In our system, at each frame, SIFT (Scale Invariant Feature Transform) features are extracted from each of the four images (stereo pair at time 1 and stereo pair at time 2), and stereo matched among the stereo pairs (Fig. 4.4). The resulting matches for the stereo pairs are then, matched again among them. Only the features finding a matching pair in the three matching processes will be used for the computation of the ego-motion.

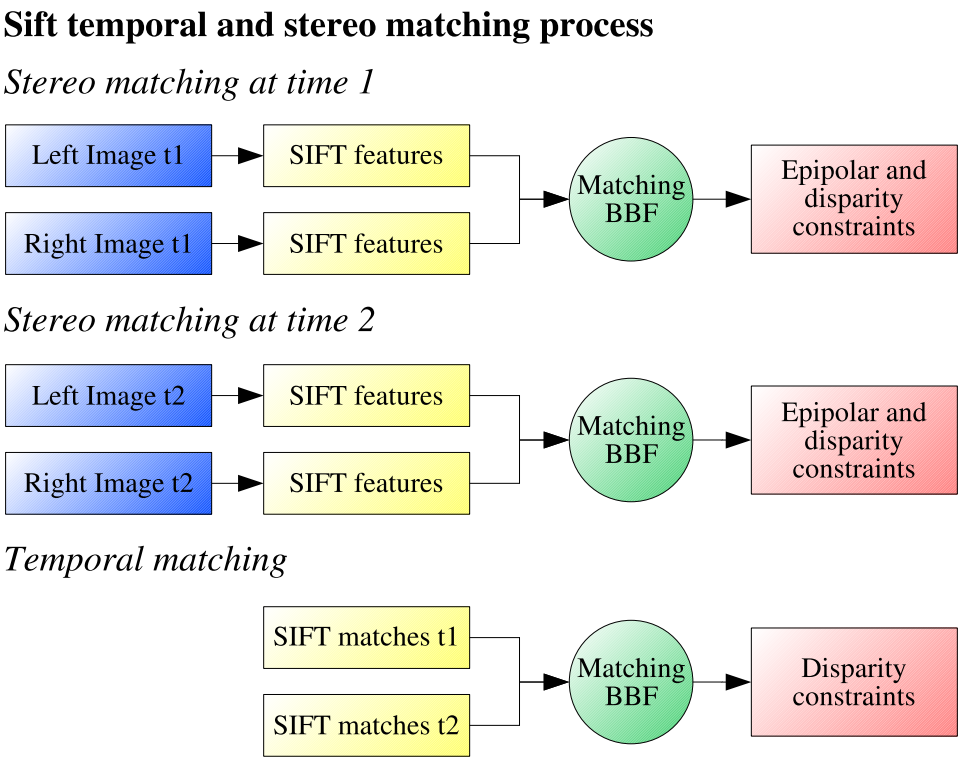


Figure 4.4: Diagram of the proposed feature extraction method

SIFT was developed by Lowe [Lowe 99] for image feature generation in object recognition applications. The features are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or 3D projection. These characteristics make them good feature points for robust visual odometry systems, since when mobile vehicles are moving around in an environment, landmarks are observed over time, but from different angles and distances. As described in [Gordon 06] the best matching candidate for a SIFT feature is its nearest neighbour, defined as the feature with the minimum Euclidean distance between descriptor vectors. The reliability of the nearest neighbour match can be tested by comparing its Euclidean distance to that of the second nearest neighbour from that image. If these distances are too similar, the nearest neighbour match is discarded as unreliable. This simple method works well in practice, since incorrect matches are much more likely to have close neighbours with similar distances than correct ones, due in part to the high dimensionality of the feature space. The

large number of features generated from images, as well as the high dimensionality of their descriptors, make an exhaustive search for closest matches inefficient. Therefore the Best-Bin-First (BBF) algorithm based on a k-d tree search [Beis 97] is used. A k-d tree is constructed from all SIFT features which have been extracted from the reference images. The search examines tree leaves, each containing a feature, in the order of their closest distance from the current query location. Search order is determined with a heap-based priority queue. An approximate answer is returned after examining a predetermined number of nearest leaves. This technique finds the closest match with a high probability, and enables feature matching to run in real time. This can give speedup by factor of 1000 while finding the nearest neighbor (of interest) 95% of the time. For each feature in a reference image, the BBF search finds its nearest and second nearest neighbour pair in each of the remaining images. Putative two-view matches are then selected based on the nearest-to-second-nearest distance ratio. As the SIFT best candidate search is not based on epipolar geometry, the reliability of matches can be improved by applying an epipolar geometry constraint to remove remaining outliers. This is a great advantage with respect to other techniques which rely on epipolar geometry for the best candidate search. For each selected image pair this constraint can be expressed as:

$$\mathbf{x}_l^T \cdot \mathbf{F} \cdot \mathbf{x}_r = 0 \tag{4.4}$$

where $\mathbf{F}$ is the Fundamental matrix previously computed in the off-line calibration process and $\mathbf{x}_l^T$, $\mathbf{x}_r$ are respectively the homogeneous image coordinates of the matched features in the $left$ image transposed and the homogeneous image coordinates of the matched features in the $right$ image . Also matches are only allowed between two disparity limits. Sub-pixel horizontal disparity is obtained for each match. This will improve the 3D reconstruction accuracy and therefore the ego-motion estimation accuracy. The resulting stereo matches between the first two stereo images are then similarly matched with the stereo matches in the next stereo pair. No epipolar geometry constraint is applied at this step and an extra vertical disparity constraint is used. If a feature has more than one match satisfying these criteria, it is ambiguous and discarded so that the resulting matching is more consistent and reliable. From the positions of the matches and knowing the cameras' parameters, we can compute the 3D world coordinates (X, Y, Z) relative to the left camera for each feature in this final set. Relaxing some of the constraints above does not necessarily increase the number of final matches (matches in the two stereo pairs and in time) because some SIFT features will then have multiple potential matches and therefore be discarded.

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

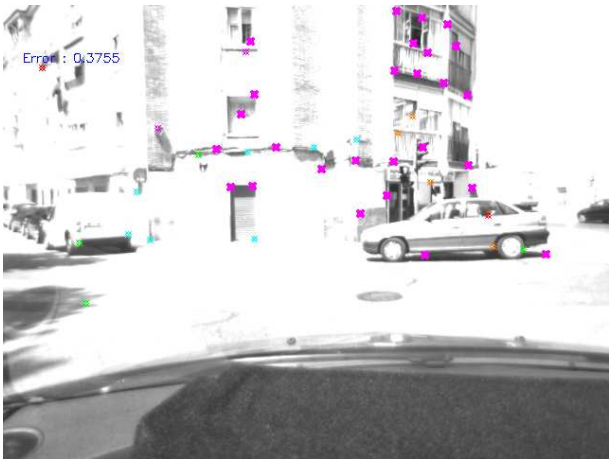
4.1.2 Feature extractors detection and tracking comparison

The methods explained in the previous section have been implemented and tested along with a Speed Up Robust Feature (SURF) extractor in the SIFT scheme. SURF features is a robust image detector and descriptor first presented in [Bay 06]. It is based on sums of approximated 2D Haar wavelet responses and makes an efficient use of integral images. SURF is several times faster than SIFT and the original implementation is claimed to be more robust than SIFT. Here we used the open source implementation from OpenCV.

The performance of the different feature extractors has been estimated using two indirect indicators: the number of inliers/outliers per frame and the error per frame in the motion estimation. Here we assume that the better the features position and its tracking the higher the number of inliers. In the case a feature extraction method yielded very few but very robust features we also evaluate the quality of the inliers looking at the final residual error in the motion estimation (see section 4.2). Finally, a global performance evaluation looking at the accuracy of the estimated trajectory is shown in the next section.

In Figure 4.5 features points for the three explained methods are shown. On the left column an overexposed frame and the extracted features are depicted (from top to bottom, Harris, SURF and SIFT). Harris features color represents the number of frames they have been tracked. For SURF and SIFT the motion of the feature was represented to explain an effect that appears in the SURF extraction method. As can be seen the number of features is higher for SURF and SIFT methods, and they both extract very similar features. However, on the right column another frame of the vehicle undergoing a forward motion is depicted. While SIFT features remain stable SURF matching is least robust to scale changes than SIFT, and delivers more incorrect matches. Even though these bad matches are easily discarded using the epipolar constraints the number of tracked feature points in forward motion decreases when using SURF features. This is an important problem because optical flow is more difficult to detect when the vehicle is undergoing a forward translation and cars move forwards most of the time. In low textured or poorly illuminated environments SURF will deliver less features than SIFT and the estimation accuracy will decrease.

In Table 4.1 the mean inliers (I/F), outliers (O/F) and estimation error per frame for two different videos are shown. Videos 01 and 04 show a path between tall buildings in narrow streets. On video 01 the shutter was selected to avoid overexposing the images. As a consequence some of the images in the narrow streets are underexposed. On the contrary, on video 04 the shutter was selected to get clear images in the narrow streets, getting overexposed images when driving at sunny streets. As can be seen in Table 4.1 SIFT and SURF outperform Harris corners, specially in dark environments. For video 04 the number of Harris features is extremely high, but most of them are outliers and the reconstruction of the trajectory is not accurate for that video using Harris. The error in the estimation is also lower for SIFT, especially in situations where few features are available. On Video 18 a tunnel was crossed. The number of inliers/outliers for SIFT and SURF are quite similar, although reconstruction results are better when using SIFT. On this video Harris fails again to get an accurate motion reconstruction. When there are very few features, and the quality of the image is poor it is very important to get reliable estimations in order to be able to keep the motion estimation. SURF gives a number of inliers similar to SIFT but the quality of the samples is lower as will be shown in the next section by the errors in the estimation.



(a) Harris Corners. Video May 8th 03 frame 181

(b) Harris Corners. Video May 8th 03 frame 270

(c) SURF Features. Video May 8th 03 frame 181

(d) SURF Features. Video May 8th 03 frame 270

(e) SIFT Features. Video May 8th 03 frame 181

(f) SIFT Features. Video May 8th 03 frame 270

Figure 4.5: Examples of extracted features

Table 4.1: Feature extractors performance

		Image Size					
		640 x 480			320 x 240		
Video	**Feature**	**I/F**	**O/F**	**Error**	**I/F**	**O/F**	**Error**
May 8th 01	Harris	28.3	7.28	3.36	41.4	10.16	15.48
	SURF	87.94	46.97	0.44	59.15	36.83	2.37
	SIFT	119.1	43.26	0.4	86.27	29.42	0.48
May 8th 18	Harris	261.95	33.2	0.044	105.58	16.72	0.13
	SURF	99.86	53.97	67.44	58.11	24.27	2.42
	SIFT	106.23	44.98	0.68	66.6	17.93	0.44
May 11st 04	Harris	252.09	30.02	0.08	262.44	19.95	0.008
	SURF	79.98	18.59	0.25	85.54	28.71	0.07
	SIFT	80.02	18.56	0.25	79.32	14.22	0.167

4.2 Visual odometry using non-linear estimation

The problem of estimating the trajectory followed by a moving vehicle can be defined as that of determining at frame i the rotation matrix $R_{i-1,i}$ and the translational vector $T_{i-1,i}$ that characterize the relative vehicle movement between two consecutive frames (see Figure 4.6).

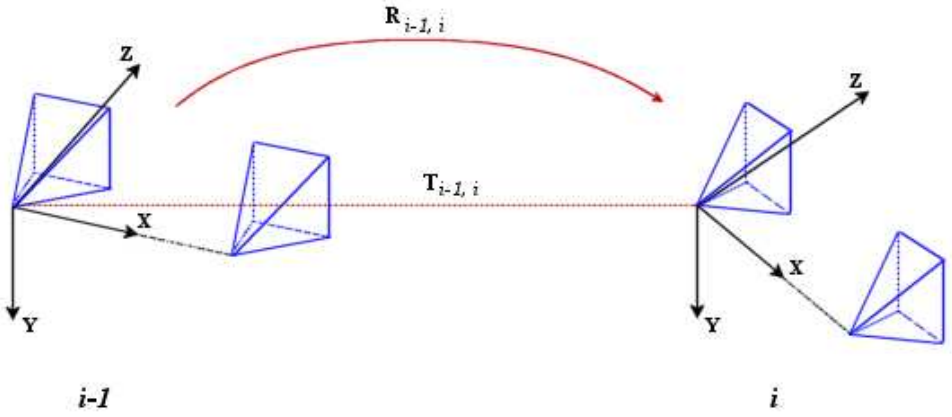


Figure 4.6: Motion estimation problem for a stereo rig

For this purpose a RANSAC based on non linear least-squares method has been developed. The system's rotation can be expressed by means of the rotation matrix R given by equation 4.5. The use of non-linear methods becomes necessary since the 9 elements of the rotation matrix can not be considered individually (the rotation matrix has to be orthonormal). Indeed, there are only 3 unconstrained, independent parameters, i.e., the three rotation angles θ_x, θ_y and θ_z, respectively.

$$\mathbf{R} = \begin{pmatrix} \cos\theta_y \cos\theta_z & \sin\theta_x \sin\theta_y \cos\theta_z + \cos\theta_x \sin\theta_z & -\cos\theta_x \sin\theta_y \cos\theta_z + \sin\theta_x \sin\theta_z \\ -\cos\theta_y \sin\theta_z & -\sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z & \cos\theta_x \sin\theta_y \sin\theta_z + \sin\theta_x \cos\theta_z \\ \sin\theta_y & -\sin\theta_x \cos\theta_y & \cos\theta_x \cos\theta_y \end{pmatrix} \quad (4.5)$$

Using a linear method can lead to a non-realistic solution where the rotation matrix is not orthonormal. However, non-linear least squares is based on the assumption that the errors are uncorrelated with each other and with the independent variables and have equal variance. The Gauss-Markov theorem shows that, when this is so, this is a best linear unbiased estimator (BLUE). If, however, the measurements are uncorrelated but have different uncertainties, a modified approach might be adopted. Aitken showed that when a weighted sum of squared residuals is minimized, the estimation is BLUE if each weight is equal to the reciprocal of the variance of the measurement [Aitken 35].

In our case, the uncertainty in the 3D position of a feature depends heavily on its location as seen in section 3.2.5, making a weighted scheme more adequate to solve the system. This is due to the perspective model in the stereo reconstruction process. A new solution based in a weighted non-linear least squares algorithm has been developed and tested on both synthetic and real data.

4.2.1 Weighted non-linear least squares

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

$$\begin{cases} f_1(x_1, x_2, \ldots, x_p) = b_1 \\ f_2(x_1, x_2, \ldots, x_p) = b_2 \\ \qquad\qquad \vdots \\ f_n(x_1, x_2, \ldots, x_p) = b_n \end{cases} \tag{4.6}$$

where f_i, for $i = 1, \ldots, n$, is a differentiable function from $\Re^p$ to $\Re$. In general, it can be stated that:

1. if $n < p$, the system solution is a $(p - n)$-dimensional subspace of $\Re^p$.
2. if $n = p$, there exists a finite set of solutions.
3. if $n > p$, there exists no solution.

As can be observed, there are several differences with regard to the linear case: the solution for $n < p$ does not form a vectorial subspace in general. Its structure depends on the nature of the f_i functions. For $n = p$ a finite set of solutions exists instead of a unique solution as in the linear case. To solve this problem, an overdetermined system is built ($n > p$) in which the weighted error function $E(x)$ must be minimized.

$$E(\mathbf{x}) = \sum_{i=0}^{N} \mathbf{W}_i \cdot (f_i(\mathbf{x}) - b_i)^2, \tag{4.7}$$

The error function $E : \Re^p \rightarrow \Re$ can exhibit several local minima, although in general there is a single global minimum. Unfortunately, there is no numerical method that can assure the obtaining of such global minimum, except for the case of polynomial functions. Iterative methods based on the gradient descent can find a global minimum whenever the starting point meets certain conditions. By using non-linear least squares the process is in reality linearized following the tangent linearization approach. Formally, function $f_i(x)$ can be approximated using the first term of Taylor's series expansion, as given by equation 4.8.

$$\begin{aligned} f_i(\mathbf{x}+\delta\mathbf{x}) = f_i(\mathbf{x}) + \delta x_1 \cdot \frac{\partial f_i}{\partial x_1}(\mathbf{x}) + \ldots + \\ +\delta x_p \cdot \frac{\partial f_i}{\partial x_p}(\mathbf{x}) + O(|\delta\mathbf{x}|)^2 \approx f_i(\mathbf{x}) + \nabla f_i(\mathbf{x}) \cdot \delta\mathbf{x} \end{aligned} \tag{4.8}$$

where $\nabla f_i(\mathbf{x}) = \left(\frac{\partial f_i}{\partial x_1}, \ldots, \frac{\partial f_i}{\partial x_p}\right)^t$ is the gradient of f_i calculated at point $\mathbf{x}$, neglecting high order terms $O(|\delta\mathbf{x}|)^2$. The error function $E(\mathbf{x}+\delta\mathbf{x})$ is minimized with regard to $\delta\mathbf{x}$ given a value of $\mathbf{x}$, by means of a iterative process. Substituting (4.8) in (4.6) yields:

$$\begin{aligned} E(\mathbf{x}+\delta\mathbf{x}) = \textstyle\sum_{i=1}^{N} \mathbf{W}_i \cdot (f_i(\mathbf{x}+\delta\mathbf{x}) - b_i)^2 \approx \\ \approx \textstyle\sum_{i=1}^{N} \mathbf{W}_i \cdot (f(\mathbf{x}) + \nabla f_i(\mathbf{x}) \cdot \delta\mathbf{x} - b_i)^2 = \\ = \textstyle\sum_{i=1}^{N} (\mathbf{W}_i \cdot (\nabla f_i(\mathbf{x}) \cdot \delta\mathbf{x}) - \mathbf{W}_i \cdot (b_i - f_i(\mathbf{x}))^2 = \\ = |\mathbf{W}_i\mathbf{J}\delta\mathbf{x} - \mathbf{W}_i\mathbf{C}|^2, \end{aligned} \tag{4.9}$$

where

$$\mathbf{J} = \begin{pmatrix} \nabla f_1(\mathbf{x})^t \\ \ldots \\ \nabla f_n(\mathbf{x})^t \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \ldots & \frac{\partial f_1}{\partial x_p}(\mathbf{x}) \\ \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \ldots & \frac{\partial f_n}{\partial x_p}(\mathbf{x}) \end{pmatrix} \tag{4.10}$$

and

$$\mathbf{C} = \begin{pmatrix} b_1 \\ \ldots \\ b_n \end{pmatrix} - \begin{pmatrix} f_1(\mathbf{x}) \\ \ldots \\ f_n(\mathbf{x}) \end{pmatrix} \tag{4.11}$$

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

$$\mathbf{W}_i\mathbf{J}\delta\mathbf{x} = \mathbf{W}_i\mathbf{C} \tag{4.12}$$

System given by equation 4.12 can be solved using least squares, yielding:

$$\delta\mathbf{x} = (\mathbf{J}^T\mathbf{W}_i\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}_i\mathbf{C} \tag{4.13}$$

The covariance estimate of the pose is approximated from the Jacobian $\mathbf{J}$ of the error function as:

$$\mathbf{\Sigma}_x = (\mathbf{J}^t\mathbf{W}_i\mathbf{J})^{-1} \tag{4.14}$$

As stated before $\mathbf{W}_i$ must be equal to the reciprocal of the variance of the measurement:

$$\mathbf{W}_i = \frac{1}{var(\mathbf{x})} = I_{pxp} \begin{pmatrix} \frac{1}{var(x_1)} \\ \frac{1}{var(x_2)} \\ \vdots \\ \frac{1}{var(x_p)} \end{pmatrix} \tag{4.15}$$

The weight matrix $\mathbf{W}$ is the uncertainty in the position of a 3D point and is computed for each point as explained in section 3.2.5.

In practice, the system is solved in an iterative process, as described in the following lines:

1. An initial solution $\mathbf{x_0}$ is chosen
2. While ($E(\mathbf{x_i}) > e_{min}$ and $i < i_{max}$)

   - $\delta\mathbf{x_i} = \mathbf{J_W}(\mathbf{x_i})^\dagger \mathbf{C}(\mathbf{x_i})$
   - $\mathbf{x_{i+1}} = \mathbf{x_i} + \delta\mathbf{x_i}$
   - $E(\mathbf{x_{i+1}}) = E(\mathbf{x_i} + \delta\mathbf{x_i}) = |\mathbf{J_w}(\mathbf{x_i})\delta\mathbf{x_i} - \mathbf{C}(\mathbf{x_i})|^2$

where the termination condition is given by a minimum value of error or a maximum number of iterations.

4.2.2 3D Trajectory estimation

Given a set of N reconstructed 3D points between instants t_0 and t_1 we have:

$$\begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} = R_{0,1} \begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + T_{0,1}; \qquad i = 1, \ldots, N \tag{4.16}$$

it yields a linear six-equations system at point i, with 6 variables $\mathbf{w} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z]^t$:

$$\begin{cases} {}^1x_i &= \cos\theta_y \cos\theta_z \cdot {}^0x_i + (\sin\theta_x \sin\theta_y \cos\theta_z + \cos\theta_x \sin\theta_z) \cdot {}^0y_i + \\ &\quad + (-\cos\theta_x \sin\theta_y \cos\theta_z + \sin\theta_x \sin\theta_z) \cdot {}^0z_i + t_x \\ {}^1y_i &= -\cos\theta_y \sin\theta_z \cdot {}^0x_i + (-\sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z) \cdot {}^0y_i + \\ &\quad + (\cos\theta_x \sin\theta_y \sin\theta_z + \sin\theta_x \cos\theta_z) \cdot {}^0z_i + t_y \\ {}^1z_i &= \sin\theta_y \cdot {}^0x_i - \sin\theta_x \cos\theta_y \cdot {}^0y_i + \cos\theta_x \cos\theta_y \cdot {}^0z_i + t_z \end{cases}$$

At each iteration k of the regression method the following linear equations system is solved (given the 3D coordinates of N points in two consecutive frames):

$$\mathbf{J_W}(\omega)\delta\mathbf{x_k} = \mathbf{C}(\mathbf{x_k}) \tag{4.17}$$

Let us remark that the first index of each Jacobian matrix element represents the point with regard to whom the function is derived, while the other two indexes represent the position in the 3x6 sub-matrix associated to such point. Considering (4.10) the elements of the Jacobian Matrix that form sub-matrix $\mathbf{J_i}$ for point i at iteration k are:

$$
\begin{cases}
J_{i,11} &= \left(\cos\theta_{\theta_{x_k}} \sin\theta_{y_k} \cos\theta_{z_k} - \sin\theta_{x_k} \sin\theta_{z_k}\right) \cdot {}^0y_i + \\
&\quad + \left(\sin\theta_{x_k} \sin\theta_{y_k} \cos\theta_{z_k} + \cos\theta_{x_k} \sin\theta_{z_k}\right) \cdot {}^0z_i \\
J_{i,12} &= -\sin\theta_{y_k} \cos\theta_{z_k} \cdot {}^0x_i + \sin\theta_{x_k} \cos\theta_{y_k} \cos\theta_{z_k} \cdot {}^0y_i - \cos\theta_{x_k} \cos\theta_{y_k} \cos\theta_{z_k} \cdot {}^0z_i \\
J_{i,13} &= -\cos\theta_{y_k} \sin\theta_{z_k} \cdot {}^0x_i + (-\sin\theta_{x_k} \sin\theta_{y_k} \sin\theta_{z_k} + \\
&\quad + \cos\theta_{x_k} \cos\theta_{z_k}) \cdot {}^0y_i + (\cos\theta_{x_k} \sin\theta_{y_k} \sin\theta_{z_k} + \sin\theta_{x_k} \cos\theta_{z_k}) \cdot {}^0z_i \\
J_{i,14} &= 1 \\
J_{i,15} &= 0 \\
J_{i,16} &= 0 \\
J_{i,21} &= -(\cos\theta_{x_k} \sin\theta_{y_k} \sin\theta_{z_k} + \\
&\quad + \sin\theta_{x_k} \cos\theta_{z_k}) \cdot {}^0y_i + (-\sin\theta_{x_k} \sin\theta_{y_k} \sin\theta_{z_k} + \cos\theta_{x_k} \cos\theta_{z_k}) \cdot {}^0z_i \\
J_{i,22} &= \sin\theta_{y_k} \sin\theta_{z_k} \cdot {}^0x_i - \sin\theta_{x_k} \cos\theta_{y_k} \sin\theta_{z_k} \cdot {}^0y_i + \cos\theta_{x_k} \cos\theta_{y_k} \sin\theta_{z_k} \cdot {}^0z_i \\
J_{i,23} &= -\cos\theta_{y_k} \cos\theta_{z_k} \cdot {}^0x_i - (\sin\theta_{x_k} \sin\theta_{y_k} \cos\theta_{z_k} + \\
&\quad + \cos\theta_{x_k} \sin\theta_{z_k}) \cdot {}^0y_i + (\cos\theta_{x_k} \sin\theta_{y_k} \cos\theta_{z_k} - \sin\theta_{x_k} \sin\theta_{z_k}) \cdot {}^0z_i \\
J_{i,24} &= 0 \\
J_{i,25} &= 1 \\
J_{i,26} &= 0 \\
J_{i,31} &= -\cos\theta_{x_k} \cos\theta_{y_k} \cdot {}^0y_i - \sin\theta_{x_k} \cos\theta_{y_k} \cdot {}^0z_i \\
J_{i,32} &= \cos\theta_{y_k} \cdot {}^0x_i + \sin\theta_{x_k} \sin\theta_{y_k} \cdot {}^0y_i - \cos\theta_{x_k} \sin\theta_{y_k} \cdot {}^0z_i \\
J_{i,33} &= 0 \\
J_{i,34} &= 0 \\
J_{i,35} &= 0 \\
J_{i,36} &= 1
\end{cases}
\tag{4.18}
$$

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

4.2.3 RANSAC

RANSAC (RAndom SAmple Consensus) [Fischler 81] [Hartley 04] is an alternative to modifying the generative model to have heavier tails to search the collection of data points S for good points that reject points containing large errors, namely "outliers". The algorithm can be summarized in the following steps:

1. Draw a sample s of n points from the data S uniformly and at random.
2. Fit to that set of n points.
3. Determine the subset of points S_i for whom the distance to the model s is bellow the threshold t. Subset S_i (defined as consensus subset) defines the inliers of S.
4. If the size of subset S_i is larger than threshold T the model is estimated again using all points belonging to S_i. The algorithm ends at this point.
5. Otherwise, if the size of subset S_i is below T, a new random sample is selected and steps 2, 3, and 4 are repeated.

6. After N iterations (maximum number of trials), draw subset S_{ic} yielding the largest consensus (greatest number of "inliers"). The model is finally estimated using all points belonging to S_{ic}.

RANSAC is used in this work to estimate the Rotation Matrix $\mathbf{R}$ and the translational vector $\mathbf{T}$ that characterize the relative movement of a vehicle between two consecutive frames. The input data to the algorithm are the 3D coordinates of the selected points at times t and $t+1$. Notation t_0 and $t_1 = t_0 + 1$ is used to define the previous and current frames, respectively, as in the next equation.

$$\begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} = R_{0,1} \begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + T_{0,1}; \qquad i = 1, \ldots, n \tag{4.19}$$

After drawing samples from three points, in step 1 models $\widetilde{R}_{0,1}$ and $\widetilde{T}_{0,1}$ that best fit to the input data are estimated using non-linear least squares.Then, a distance function is defined to classify the rest of points as inliers or outliers depending on threshold t.

$$\begin{cases} \text{inlier} & e < t \\ \text{outlier} & e \geqslant t \end{cases} \tag{4.20}$$

Generally, the distance function is the square error between the sample and the predicted model. The 3D coordinates of the selected point at time t_1 according to the predicted model are computed as:

$$\begin{pmatrix} {}^1\widetilde{x}_i \\ {}^1\widetilde{y}_i \\ {}^1\widetilde{z}_i \end{pmatrix} = \widetilde{R}_{0,1} \begin{pmatrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{pmatrix} + \widetilde{T}_{0,1}; \qquad i = 1, \ldots, n \tag{4.21}$$

The error vector is computed as the difference between the estimated vector and the original vector containing the 3D coordinates of the selected points (input to the algorithm):

$$\mathbf{e} = \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} = \begin{pmatrix} {}^1\widetilde{x}_i \\ {}^1\widetilde{y}_i \\ {}^1\widetilde{z}_i \end{pmatrix} - \begin{pmatrix} {}^1x_i \\ {}^1y_i \\ {}^1z_i \end{pmatrix} \tag{4.22}$$

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

$$e = |\mathbf{e}|^2 = \mathbf{e}^t \cdot \mathbf{e} \tag{4.23}$$

However, as the individual uncertainties of each reconstructed point depend on their position, the distance function must take into account their individual uncertainty. To do so, the Mahalanobis distance [Mahalanobis 36] is used here to get the support of the sample to the minimal solution. In our system the 3D position of a reconstructed point has been modelled as a multivariate Gaussian distribution (see Section 3.2.5).

Mahalanobis distance was introduced by P. C. Mahalanobis in 1936 and had been widely used in cluster analysis and other classification techniques. Mahalanobis distance has been often used to detect multivariate outliers [Filzmoser 03] [Garrett 89]. It is a useful way of determining similarity of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale invariant. The Mahalanobis distance between 2 random vectors x and y of the same distribution with covariance matrix S is defined as:

$$d_M(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)} \tag{4.24}$$

If the covariance matrix is diagonal, then the resulting distance measure is called the normalized Euclidean distance:

$$d_M(x,y) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2}}, \tag{4.25}$$

where σ_i is the standard deviation of the x_i over the sample set.

In this work the standard deviation for each reconstructed 3D point is computed using (3.20) and the support of a single point to a sample hypotheses is measured as the Mahalanobis distance to its predicted position by the solution for the sample hypotheses.

Let $\widetilde{\mathbf{R}}_{i-1,i}$ and $\widetilde{\mathbf{T}}_{i-1,i}$ be the solution for a subset of the input data. The estimated 3D position of a sample $\widetilde{\mathbf{P}}_i$ is defined

$$\widetilde{\mathbf{P}}_i = \widetilde{\mathbf{R}}_{i-1,i} \cdot \mathbf{P}_{i-1} + \widetilde{\mathbf{T}}_{i-1,i} \tag{4.26}$$

The Mahalanobis distance between the predicted value $\widetilde{\mathbf{P}}_i$ and the measured one $\mathbf{P}_i$ is then

$$d_M(\widetilde{\mathbf{P}}_i, \mathbf{P}_i) = \sqrt{\frac{(\widetilde{x}_i - x_i)^2}{\sigma_x^2} + \frac{(\widetilde{y}_i - y_i)^2}{\sigma_y^2} + \frac{(\widetilde{z}_i - z_i)^2}{\sigma_z^2}} \tag{4.27}$$

where σ_i is the uncertainty on the i coordinate given by (3.20). In this work a value of $t = 0.7$ has been experimentally chosen for the distance threshold.

The use of Mahalanobis distance instead of Euclidean increases the number of points used to solve for the motion. This allows for more robust estimations in complex environments with few points or with many outliers (ie when non-stationary objects are in the scene).

Number of iterations N

Normally, it is unviable or unnecessary to test all the possible combinations. In reality, a sufficiently large value of N is selected in order to assure that at least one of the randomly selected s samples is outlier-free with a probability p. Let ω be the probability of any sample to be an inlier. Consequently, $\epsilon = 1-\omega$ represents the probability of any sample to be an outlier. At least, N samples of s points are required to assure that $(1-\omega^s)^N = 1-p$. Solving for N yields:

$$N = \frac{log(1-p)}{log(1-(1-\epsilon)^s)} \tag{4.28}$$

In this case, using samples of 3 points, assuming $p = 0.99$ and a proportion of outliers $\epsilon = 0.25$ (25%), at least 9 iterations are needed. In practice, the final selected value is $N = 10$.

Consensus threshold T

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

$$T = (1 - \epsilon)n \tag{4.29}$$

MatLab Simulator

Trajectory estimation and global positioning using ego motion is a difficult task from computer vision perspective. Large variations in environmental conditions (e.g. lighting, moving cars, poor texture scenes, repetitive patterns, etc.) make this problem particularly challenging. Understanding the influence of the different errors in the estimation is crucial to focus the research. A mathematical study of the different errors present in the ego-motion estimation was carried out using MatLab. To do so, the proposed trajectory estimation algorithm was programmed in MatLab assuming ideal conditions, and different errors were added one by one allowing us to measure their effect on the final trajectory estimation. The weighted and non weighted solutions were tested on the simulator, as well as the Euclidean and Mahalanobis distance for RANSAC. Results for a synthetic trajectory are shown in Figure 4.7. For further details on the MatLab simulator, please refer to [Parra 10].

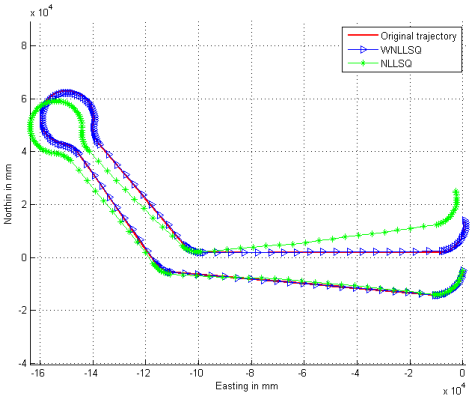
The different solutions were tested using a synthetic trajectory of approximately 406m with several turns and straight stretches. The simulation velocity was 30 km/h and the sampling rate was 6 frames per second. The feature points were generated using a uniform distribution ranging [-3 3] meters wide (x axis), [0 2] meters in height (y axis) and [1 20] m in depth (z axis). The same trajectory was reconstructed using non-linear least squares and weighted non-linear least squares. The results in Table 4.2 show an improvement in the mean distance to the ground truth of about 20 times the previous ones. As expected, all the figures in the table are improved with the weighted solution, but the most significant improvement is the actual shape of the estimated trajectory, which can be seen in Figure 4.7(a). The trajectory with the weighted estimation keeps the shape of the original trajectory while the heterodasticity in the non weighted solution bends the trajectory drifting it away from the real one.

Table 4.2: Results of the MatLab Simulator

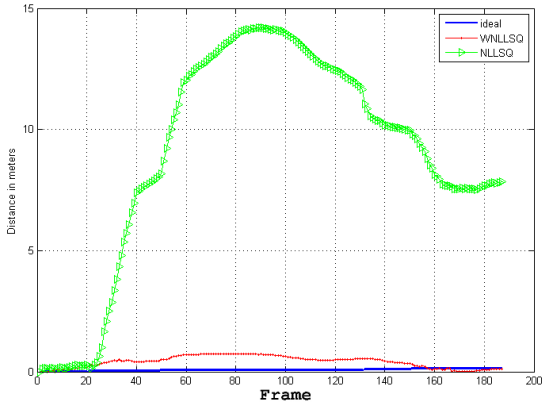
	Mean/max error in tx (mm)	Mean/max error in tz (mm)	Mean/max error in yaw (rad)	Mean/Max distance to real point (m)	Length of the run (m)	Estimated length (m)
Ideal System	0.000001 0.000022	-0.000043 0.000096	0.000000 0.000000	0.074141 0.137823	405.79	405.79
Non-linear LSQ	-0.465969 187.853817	-7.599702 178.345743	-0.000021 0.039577	9.005537 14.190766	405.79	404.48
Non-linear WLSQ	-0.310400 32.527168	-7.599702 155.569126	-0.000021 0.007994	0.413442 0.746929	405.79	404.19

In Figure 4.7(b) the distances to the ground truth of the ideal, the weighted and the non-weighted solutions are depicted. The ideal solution doesn't include the quantization

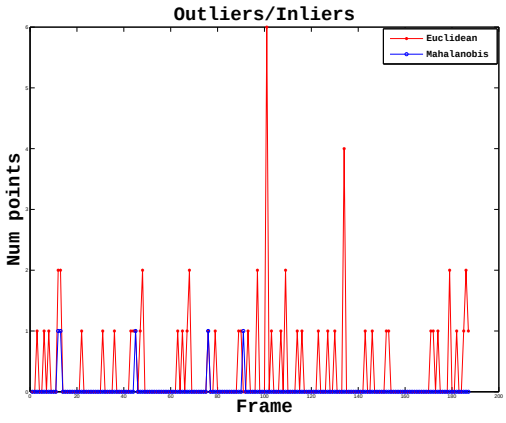
error of the cameras, meaning there is no uncertainty when determining the position of a feature on the image plane. The ideal system was solved using non-linear least squares, because the heterodasticity disappears when there is no uncertainty. The error in the ideal solution is due to the linearization of the non-linear system, and it is equivalent to having infinite precision in the determination of the 2D position of the features. From this point of view it can be considered as the best possible solution using this linearization method. As can be seen, the error introduced by the linearization is small in comparison with the error due to the 3D reconstruction. As a consequence the effort have to be put in getting accurate features. In Figure 4.7(c) the number of outliers detected by RANSAC are depicted. No outliers were introduced in this simulation. When using Euclidean distance, all the detected outliers are due to the heterodasticity of the 3D uncertainty. Far points presenting a higher quantization error are rejected even though they are inliers. If the threshold distance t were adjusted to accept far inliers, outliers close to the car would be accepted as inliers due to the relaxation of the threshold. When using Mahalanobis distance the outliers are due to the fail to represent the longer tails of the further points (see Section 3.2.5). As shown on the Figure, Mahalanobis distance keeps a higher number of inliers thanks to a better representation of their uncertainty, which is very important for a robust estimation when few features are available.



(a) Estimated 2D trajectory for the weighted and non-weighted solutions



(b) Distance to the ideal system per frame



(c) Number of outliers for the Euclidean and Mahalanobis Distance

Figure 4.7: Simulator results for a synthetic trajectory

4.2.4 2D Approximation

In the typical driving scenario, the road forms a planar structure and the motion of the car can be modelled with 3 predominant parameters: forward translation, pitch and yaw. With this simplification it is possible to devise a method more robust to the hard conditions of urban environments. However, in this model, we assume a coordinate frame in which the ground plane is parallel to the XZ plane of our camera coordinate system and that the optical axis is parallel to the Z axis (see Figure 4.2.4). This is not true and requires that the points be rectified prior to computing the ego-motion.

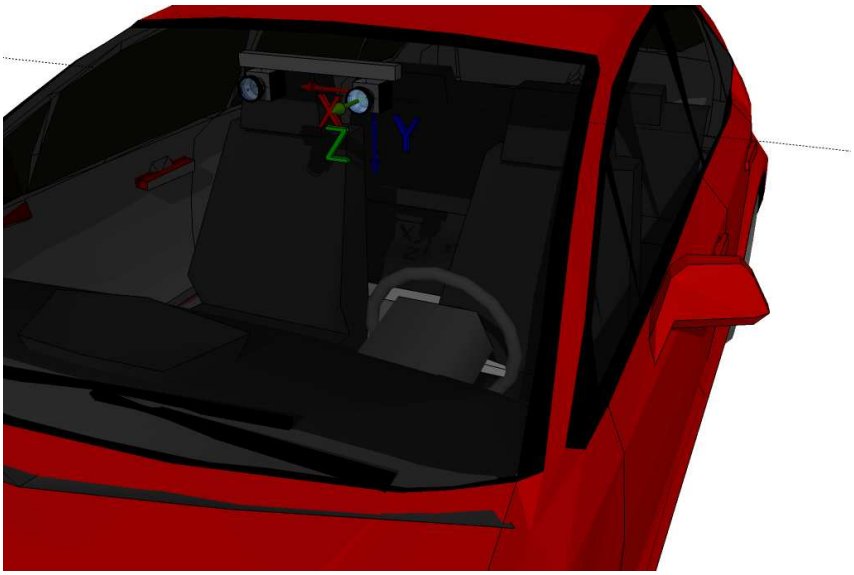
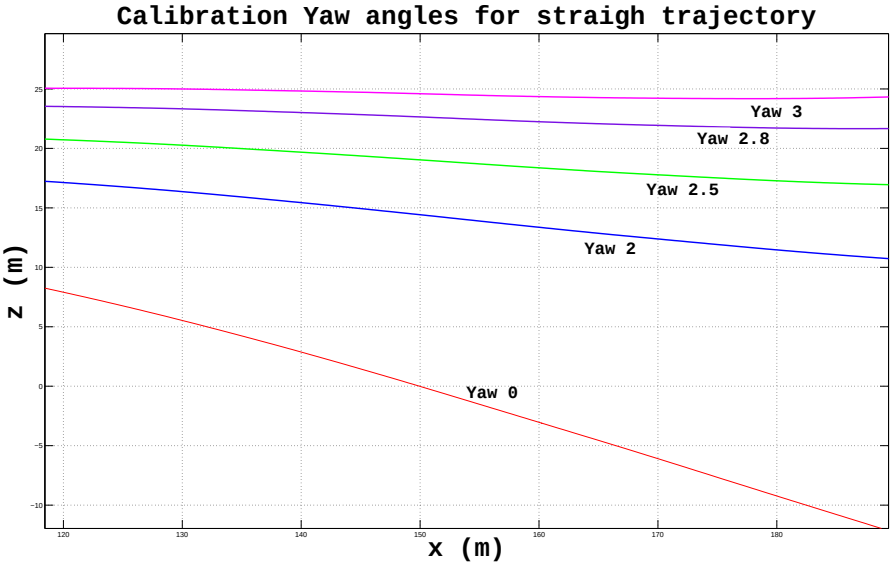


Figure 4.8: Camera coordinate system

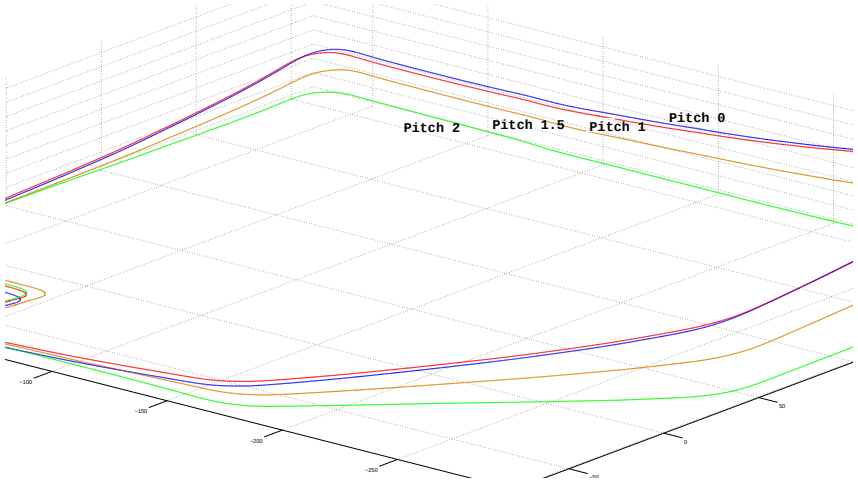
To estimate the pitch and yaw of the cameras rig with respect to the ideal position an off-line calibration procedure is performed as follows:

1. Estimate the ego motion for a video showing a long straight motion on a flat road using only forward translation and yaw.
2. If the rig has some rotation around the y axis the motion model will try to compensate this rotation by bending the trajectory around the y axis.
3. Adjust a yaw value for the rig and repeat from 1 until the depicted trajectory is straight.
4. Estimate the ego motion using pitch, yaw and forward translation.
5. If the rig has some rotation around the x axis the motion model will try to compensate this rotation by bending the trajectory around the x axis.
6. Adjust a pitch value for the rig and repeat from 4 until the depicted trajectory is flat.

This calibration values for the pitch and the yaw are then used for that calibration of the cameras. Prior to the ego-motion estimation the 3D position of the points will be corrected according to this values to comply with the simplified pitch, yaw and forward translation model. This approximation, along with the RANSAC outliers rejection step, allows the system to cope with moving objects such as pedestrians or other cars. On the



(a) Estimated straight trajectory for different Yaw angles(degrees)



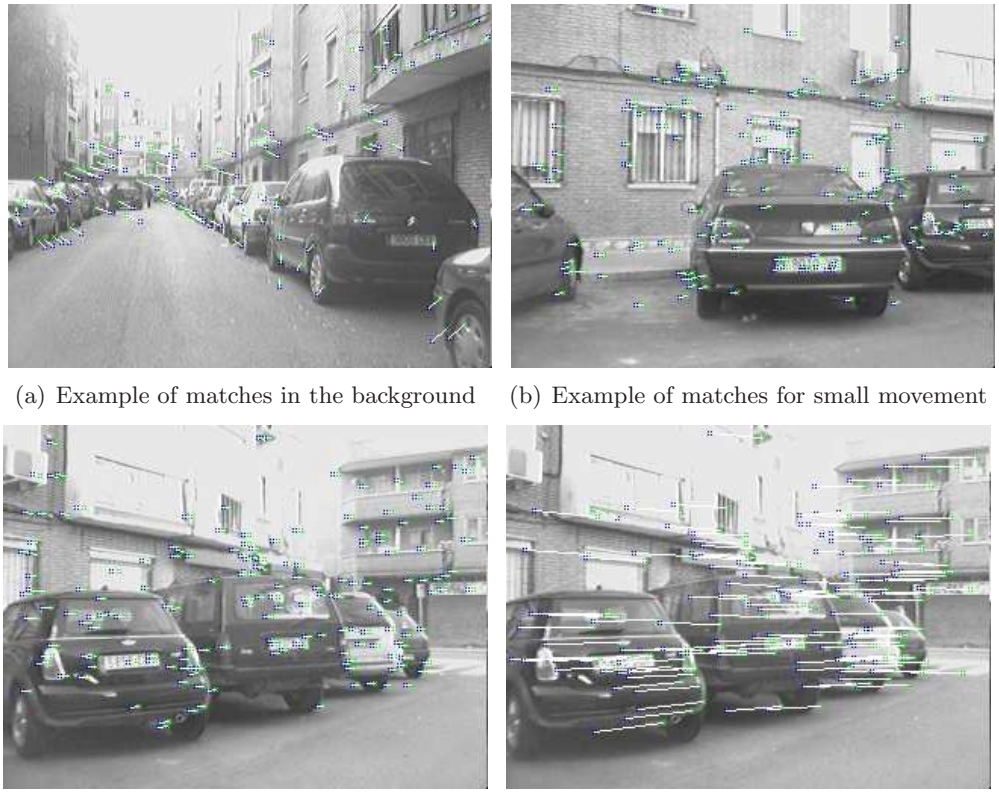
(b) Estimated flat trajectory for different Pitch angles(degrees)

Figure 4.9: Images of the 2D calibration procedure.

one hand RANSAC will reject every minimal solution as long as the number of stationary points being tracked is higher than the outliers (pedestrians or other moving cars). On the other hand the 2D approximation adds some information about the car dynamics to the model.

4.2.5 Data Post-processing

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



(a) Example of matches in the background (b) Example of matches for small movement

(c) Examples of matching, temporal shift 1 (d) Examples of matches, temporal shift 10

Figure 4.10: Examples of SIFT matches. In green SIFT feature at time t1 in blue matched feature at time t2, in white the movement of the feature.

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

1. Solutions for small movements (5 centimeters or less) where the distance between features is also small (one or two pixels), are prone to yield inaccurate solutions due to the discretized resolution of the 3D reconstruction (Fig. 4.10(b)).

2. Solutions for images where the features are in the background of the image (Fig. 4.10(a)) are inaccurate for the same reason as previously mentioned: 3D reconstruction resolution decreases as long as depth increases. Although the features extraction algorithm sorts the features depending on its depth and it uses the closest ones, at some frames it is not able to find enough features close to the car.

SIFT features have proven to be robust to pose and illumination changes, so they are good candidates for matching, even if there are some skipped frames between the matching stereo pairs and thus, the appearance of the features has changed (Fig. 4.10(d)). Also the fact that they do not rely on the epipolar geometry for the matching process makes its computational time independent on the disparity between features. Using a correlation based matching process it would be necessary to increase the disparity limits in order to find the features which will probably be further away from each other. According to this some ego-motion estimations are discarded using the following criteria.

1. High root mean square error e estimations are discarded.

2. Meaningless rotation angles estimations (non physically feasible) are discarded.

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

4.2.6 Experiments and results

The experimental vehicle used in this thesis is a car (Citröen C4) which can be seen in Figure 4.11. It has an on-board computer housing the image processing system, a RTK-DGPS and a low cost GPS connected via USB and a pair of IEEE1394 digital cameras synchronized using an external circuit. A software, specifically developed for this thesis, captures the synchronized camera images and the RTK-GPS, GPS and BUSCAN information from the car. All this information is embedded into 640×480 gray scale images in an overhead along with the capture time stamp and the camera parameters (shutter, gain, exposure, etc.) for each image.

The RTK-GPS receives differential corrections at 5Hz from a base station through the Internet. Although corrections are sent at 5Hz the base station only computes the correction parameters at 1Hz so we can consider that the RTK-GPS corrects at 1Hz but delivers position information at 5Hz. The low cost GPS works at 1Hz.

The stereo sensor uses a baseline of approximately 300mm and a focal length of 4.2mm. Sequences were recorded at different locations in Alcalá de Henares (Madrid). All the sequences correspond to real traffic conditions in urban environments with pedestrians and other cars in the scene. In the experiments, the vehicle was driven around the maximum allowed velocity in cities, i.e., 50 Km/h. More than 3 hours of video has been recorded. Here some significant results are commented.

Figure 4.11: (Top left) Stereo cameras. (Top right) RTK-GPS. (Bottom) Experimental Vehicle

Loop closure

In order to test the accuracy of the ego-motion estimation, a typical experiment is to drive in a closed loop and look at the loop closure error. In this experiment the car was driven in a 1.1Km loop around the University of Alcalá Escuela Politécnica. A representation of the path followed by the car over-imposed in Google Maps is depicted on Figure 4.12.

Figure 4.12: Trajectory for Video 00 May 8th on Google Maps.

The environment is not richly textured, there are no buildings and some repetitive patterns (a fence) can be found at the end of the loop. The sun is high and moderate glares appear on the windshield. Three cars crossed with the ego vehicle travelling on the opposite direction, and one more travelling on the same direction ahead of the ego-vehicle.

At the end, a pedestrian crossed in front of the ego-vehicle. Frames depicting some of the situations explained above can be seen on Figure 4.13.

The estimated 2D trajectory for SIFT and 640×480 images is depicted in Figure 4.14(a) together with the GPS and RTK GPS ground truth. The estimated trajectory is very close to the ground truth, both in shape and length with a loop closure error of 0.2% and a distance estimation error of 0.31% (see Table 4.3). As mentioned before, the robustness of the SIFT feature extractor outperforms SURF, giving better results even for lower resolutions as shown in Table 4.3. The trajectory was correctly reconstructed in the presence of glares and other non-stationary cars and pedestrians. A small over estimation of the motion is produced at very low speeds in the initial frames, due to the lack of features close to the vehicle. The mean distance to the features in these first 500 frames is about 70m where the precision in the 3D position estimation is very low. This effect can be seen clearly in Figure 4.14(b) where the estimated velocity is depicted along with the velocity measured by the GPS.

Table 4.3: Ground truth and estimated lengths for video 00 May 8th

	dGPS (m)	GPS (m)	VO (m)	Loop Error (m)	Disc Fr t_z %
SIFT 640×480	Lost	1098.1	1101.5	2.48	5.07
SIFT 320×240	Lost	1098.1	946.8	31.62	13.21
SURF 640×480	Lost	1098.1	565.53	45.23	7.15

The video sequence was processed using different features and image sizes. Results can be seen in Table 4.3. These results show a tendency to underestimate the motion when the image size decreases. This is due to the fail of the Gaussian approximation of the 3D reconstruction uncertainty. Very distant points have very long tails and the symmetric Gaussian representation underestimate their position (see 3.2.5). Also, SURF shows worse performance than SIFT, especially for poor textured environments like this one.

The velocity estimated by the visual odometry is the mean of 30 samples to filter the information an get the same rate as a GPS. As shown in Figure 4.14(b) the estimated velocity is very close to the velocity measured by the GPS. This gives very useful information about the probability of being turning or simply crossing at intersections for the map matching algorithm. On Figure 4.14(c) the estimated yaw of the vehicle is also represented with high precision. Around frame 1500 the vehicle maneuvers to avoid a car coming and recovers the lane which can be seen on Figure 4.14(c).

(a) Frame 242

(b) Frame 358

(c) Frame 449

(d) Frame 1548

(e) Frame 1551

(f) Frame 1559

(g) Frame 1903

(h) Frame 1912

(i) Frame 1922

(j) Frame 3501

(k) Frame 3569

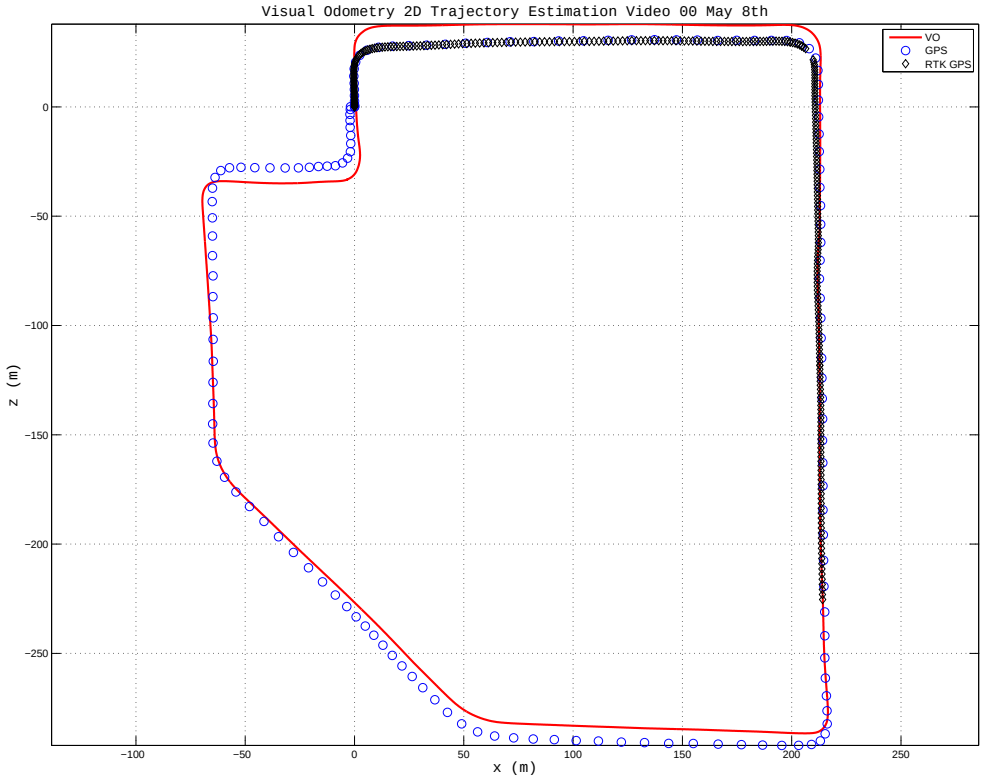
(l) Frame 3700

(m) Frame 4114

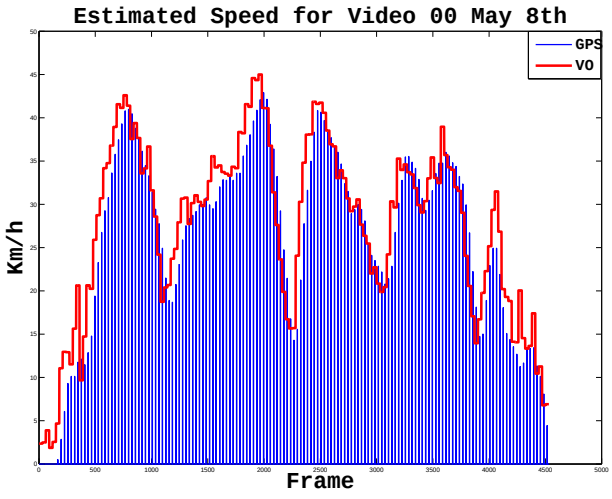
(n) Frame 4139

(o) Frame 4163

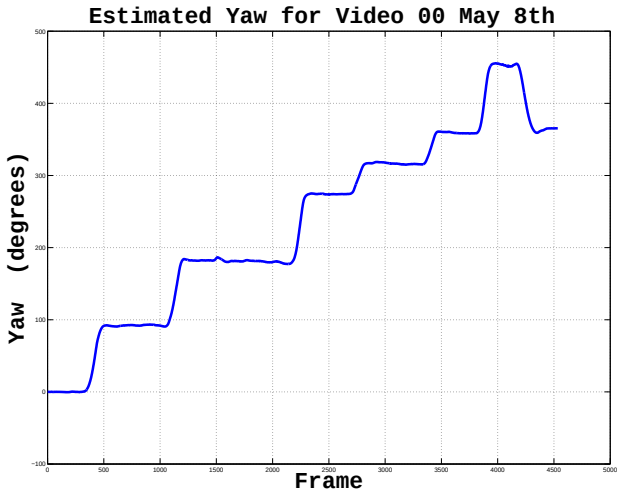
Figure 4.13: Frames from video 00 May 8th



(a) Estimated 2D trajectory for Video 00 May 8th.



(b) GPS and estimated velocity



(c) Estimated Yaw

Figure 4.14: Estimated velocity and Yaw for video 00 May 8th

Tunnel. Pitch estimation

In the typical driving scenario, the road forms a planar structure and the motion of the car can be modelled with 3 predominant parameters: forward translation, pitch and yaw. In this example we will test the accuracy of the pitch estimation and its importance in ego-motion estimation systems. This video shows a 643.96m run in urban scenario (a Google Maps representation of this experiment is depicted in Figure 4.15).

Figure 4.15: Trajectory for Video 05 May 8th on Google Maps.

It was a very bright day, with strong shadows from buildings and overexposed frames. Moderate glaring appears on the windshield. The traffic was not heavy, but several cars crossed the cameras field of view during the experiment. Around the middle of the experiment the vehicle went through a small tunnel and strong changes of illumination at its entrance and exit delivered very poor textured frames. Examples depicting some of the situations explained above can be seen on Figure 4.16.

In Figure 4.17(a) and Table 4.4 the results for the trajectory estimation are shown. The reconstructed trajectory has a small overestimation (around 3%) on the distance of the straight heading to the tunnel. The possible reason for that is the poor quality of the images at the tunnel entrance/exit and also the 2D reconstruction of the GPS, which will slightly underestimate the distance in the height changes. The shape of the trajectory is recovered with high precision even in the presence of other moving cars and over-exposed and under-exposed images.

Table 4.4: Ground truth and estimated lengths for video 05 May 8th

	dGPS (m)	GPS (m)	VO (m)	VO %	Disc Fr t_z %
SIFT 640×480	679.0	643.96	700.63	3.19	8.12
SIFT 320×240	679.0	643.96	598.95	6.83	11.43
SURF 640×480	679.0	643.96	306.68	51.41	23.79
HARRIS 640×480	679.0	643.96	537.28	16.44	8.07

In Figure 4.17(b) the estimated speed indicates that the error in the estimated distance

(a) Frame 617 (b) Frame 659 (c) Frame 734

(d) Frame 869 (e) Frame 1088 (f) Frame 1176

(g) Frame 1291 (h) Frame 1331 (i) Frame 1368

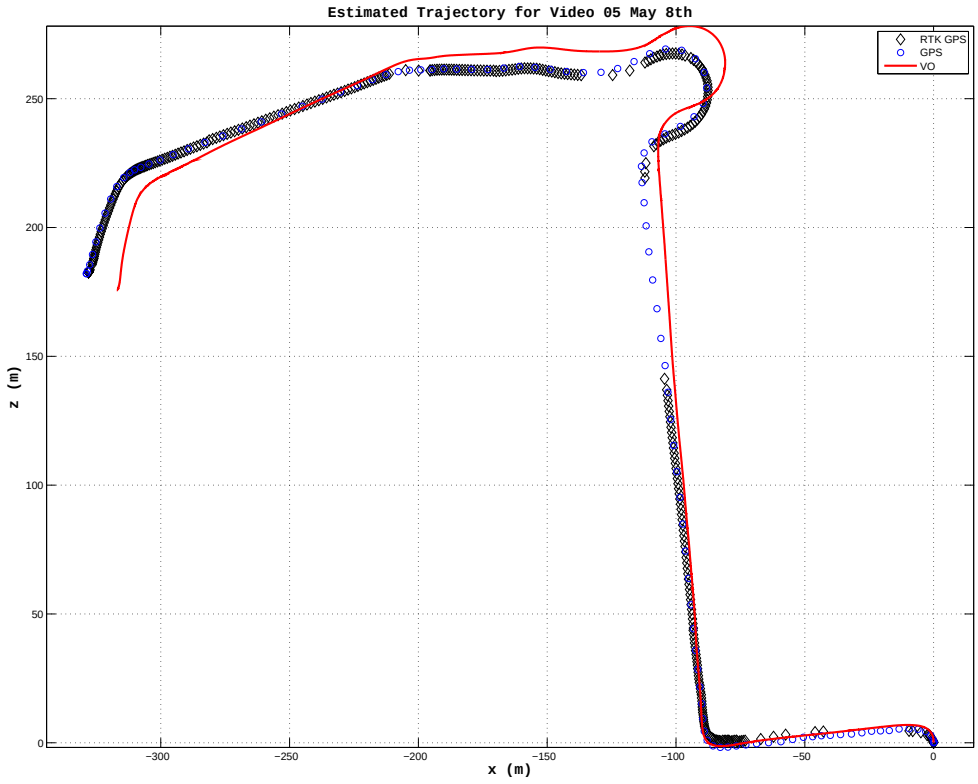
(j) Frame 1440 (k) Frame 2105 (l) Frame 2116

(m) Frame 2128 (n) Frame 2325 (o) Frame 2380

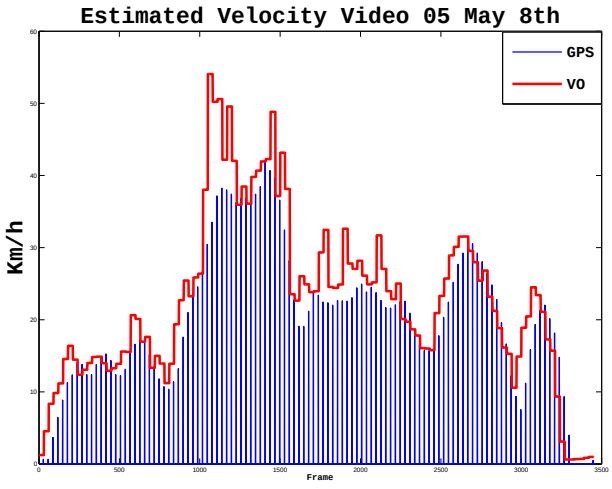
Figure 4.16: Frames from video 05 May 8th

happens at the entrance of the tunnel (frames 1000-1300), when the road is pitching down what supports the hypothesis of lack of precision in the height estimation. Also the difference in the length of the RTK GPS and the GPS suggests that the coverage was not good (see Table4.4).

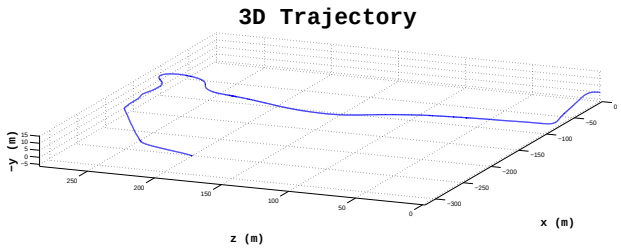
In Figure 4.17(c) the 3D estimated trajectory for the experiment is shown. As ca be seen, the system estimates correctly the slopes of the tunnel. Unfortunately, there is no ground truth information available about the height of the car and the accuracy of the pitch can not be precisely known. The GPS and dGPS height information show a change in height of about 5-7m which is in the range of the change of altitude in the reconstructed trajectory.



(a) Estimated 2D trajectory for Video 05 May 8th.



(b) GPS and estimated velocity



(c) Estimated 3D trajectory

Figure 4.17: Estimated velocity and 3D trajectory for video 05 May 8th

Images synchronization

The synchronization in the capture of the images is very important for a correct motion reconstruction. For robotics platforms, at low velocities, the synchronization is not critical. In urban environments speeds up to 70 Km/h can be reached and at that speeds 30ms of difference between captures means a difference of about 65cm in the point the image is taken. To show the importance of the synchronization of the cameras the results of a desynchronized video are shown. In this video a failure in the hard disk writing led to a desynchronization between frames of 33ms starting on frame 392. A Google Maps representation of this experiment is depicted in Figure 4.18.

Figure 4.18: Trajectory for Video 09 May 8th on Google Maps.

This experiment shows a 90 degrees right turning and a slope to get into a bridge. Several cars crossed the scene in the opposite direction. Frames depicting some of the situations explained above can be seen on Figure 4.19. The results for the 2D and 3D trajectory estimation are shown in Figures 4.20(a) and 4.20(c) respectively. The trajectory bends to the left as a consequence of the desynchronization (the desynchronization point was labeled with a green asterisk), but the slope of the bridge is correctly estimated. The velocity shows some inaccuracies, probably due to the desynchronization (see Figure 4.20(b)).

Table 4.5: Ground truth and estimated lengths for video 09 May 8th

	dGPS (m)	GPS (m)	VO (m)	VO %	Disc Fr t_z %
SIFT 640×480	481.93	480.47	571.85	18.96	3.12

(a) Frame 41 (b) Frame 144 (c) Frame 212

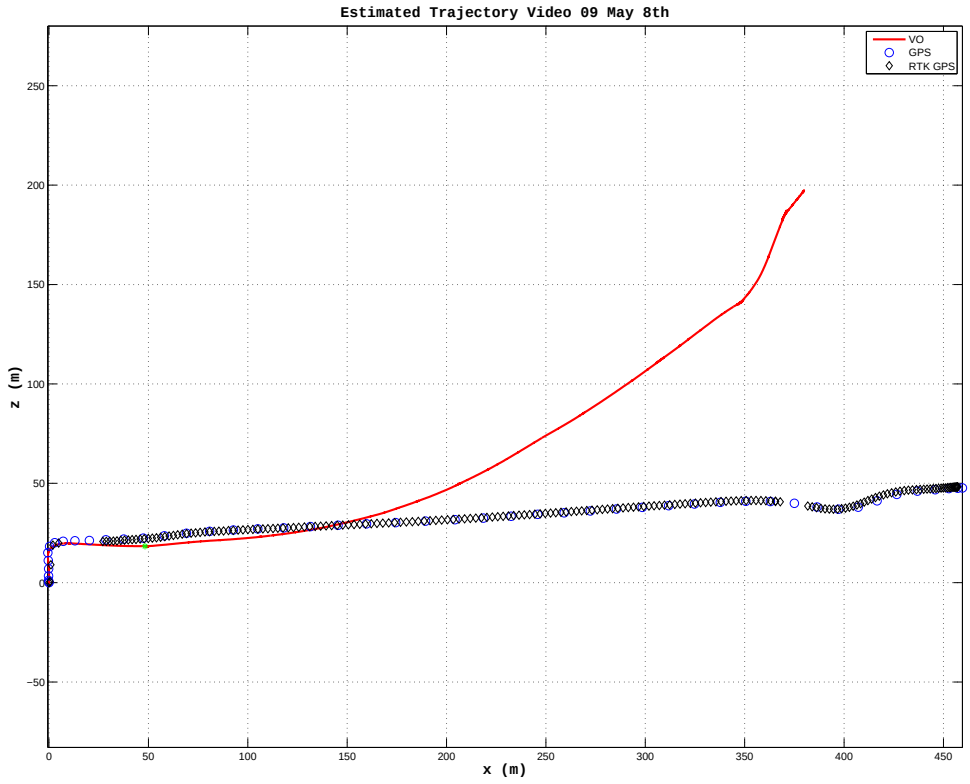
(d) Frame 396 (e) Frame 484 (f) Frame 585

(g) Frame 700 (h) Frame 763 (i) Frame 922

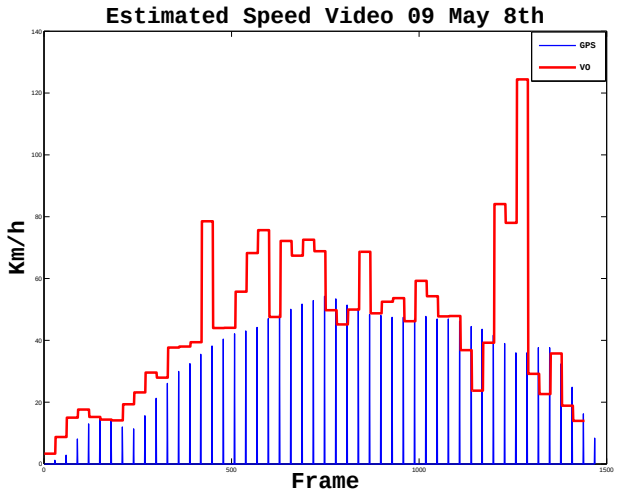
(j) Frame 943 (k) Frame 1079 (l) Frame 1185

(m) Frame 1225 (n) Frame 1326 (o) Frame 1394

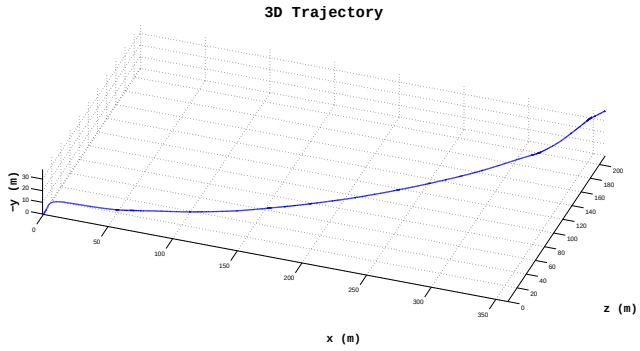
Figure 4.19: Frames from video 09 May 8th



(a) Estimated 2D trajectory for Video 09 May 8th.



(b) GPS and estimated velocity



(c) Estimated 3D trajectory

Figure 4.20: Estimated velocity and 3D trajectory for video 09 May 8th

Urban trajectory

In the following experiment the vehicle followed a path through a urban canyon. The images are underexposed due to the buildings shadows an two cars crossed the cameras field of view: one on the opposite direction and one on the same lane (see Figure 4.22). A Google Maps representation of this experiment is depicted in Figure 4.21.

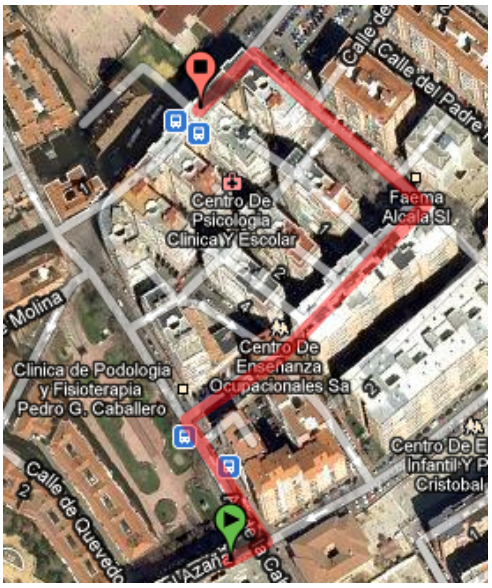


Figure 4.21: Trajectory for Video 15 May 8th on Google Maps.

The estimated distance and trajectory are shown in Table 4.6 and Figure 4.23(a). The closeness of the buildings and the richly textured environment makes the number of tracked features and its distance ideal for the ego-motion estimation. On the contrary also moving cars and pedestrians are closer and can affect the estimation. The accuracy of both the path length and the shape is high with a 0.41% of error in the length estimation.

Table 4.6: Ground truth and estimated lengths for video 15 May 8th

	dGPS (m)	GPS (m)	VO (m)	VO %	Disc Fr t_z %
SIFT 640×480	Lost	418.2	421.14	0.41 %	4.22
SIFT 320×240	Lost	418.2	348.11	16.74 %	10.27

Compared to the previous experiments, the speed is lower as can be seen in Figure 4.23(b) which allows the system to track closer features and increase the estimation accuracy. On Figure 4.23(c) the estimated yaw is depicted. The maneuvers to avoid double-parked cars can be seen between frames 600-1500.

(a) Frame 31

(b) Frame 89

(c) Frame 129

(d) Frame 185

(e) Frame 217

(f) Frame 299

(g) Frame 500

(h) Frame 527

(i) Frame 549

(j) Frame 1163

(k) Frame 1261

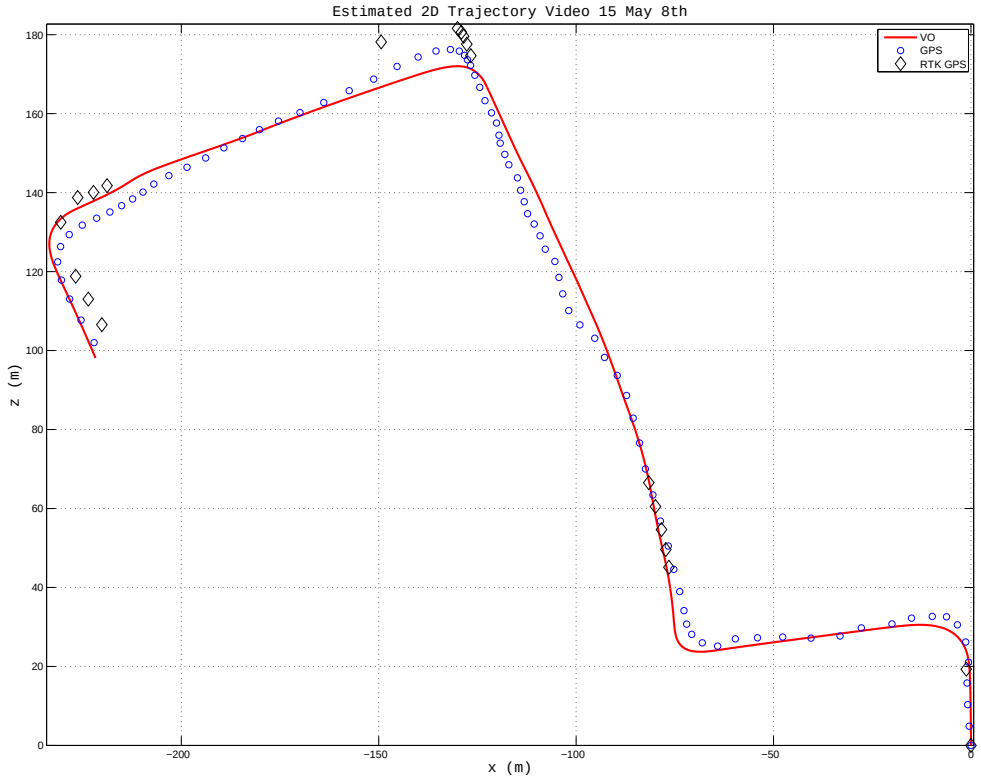
(l) Frame 1410

(m) Frame 2441

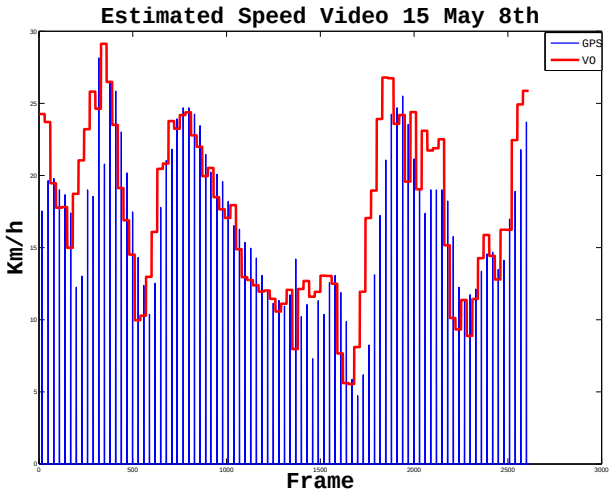
(n) Frame 2529

(o) Frame 2621

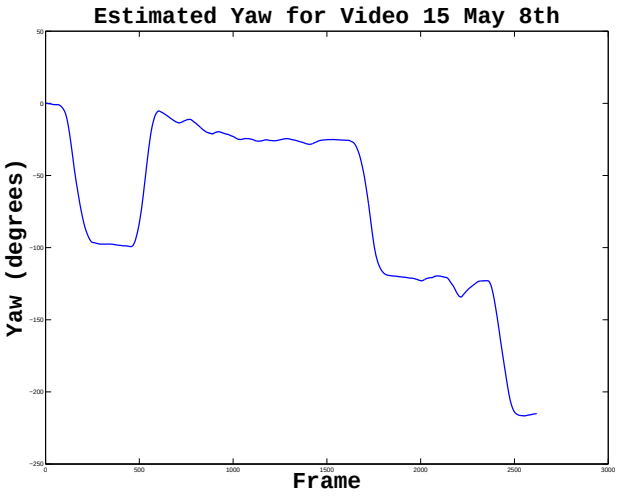
Figure 4.22: Frames from video 15 May 8th



(a) Estimated 2D trajectory for Video 15 May 8th.



(b) GPS and estimated velocity



(c) Estimated Yaw

Figure 4.23: Estimated velocity and yaw for video 15 May 8th

Loop and glares

On the following experiment the vehicle was driven through a narrow and dark boulevard, and then a loop was performed and the vehicle continued for another 300m. Some glares from other cars and a truck moving in front of the ego-vehicle are the main challenges of this video. Frames depicting some of the situations explained above can be seen on Figure 4.25. A Google Maps representation of this experiment is depicted in Figure 4.24.

Figure 4.24: Trajectory for Video 17 May 8th on Google Maps.

The length estimation accuracy is similar to other videos (error of 0.6%) but a small overestimation of the Yaw angle during the loop leads to a drift of the trajectory. The error is probably due to the present of strong glares on the loop. This kind of small but cumulative errors that can lead to mislocalizations will be corrected by the map matching system explained in the next section.

Table 4.7: Ground truth and estimated lengths for video 17 May 8th

	dGPS (m)	GPS (m)	VO (m)	VO %	Disc Fr t_z %
SIFT 640×480	Lost	697.11	693.08	0.6 %	6.62
SIFT 320×240	Lost	697.11	591.12	15.2 %	9.22

As can be seen on Figure 4.26(b) there are 2 points where the system overestimates the motion. One is a 300 frames stretch ($\tilde{6}$00-900) where the trees cover the field of view of the camera and the images are extremely underexposed. This leads to a first deviation of approximately 2 degrees in the yaw. The other one is a 800 frames stretch ($\tilde{1}$200-2000) where glares and a long bush covering most of the scene cold be the reason for another overestimation of the yaw.

(a) Frame 196 (b) Frame 459 (c) Frame 499

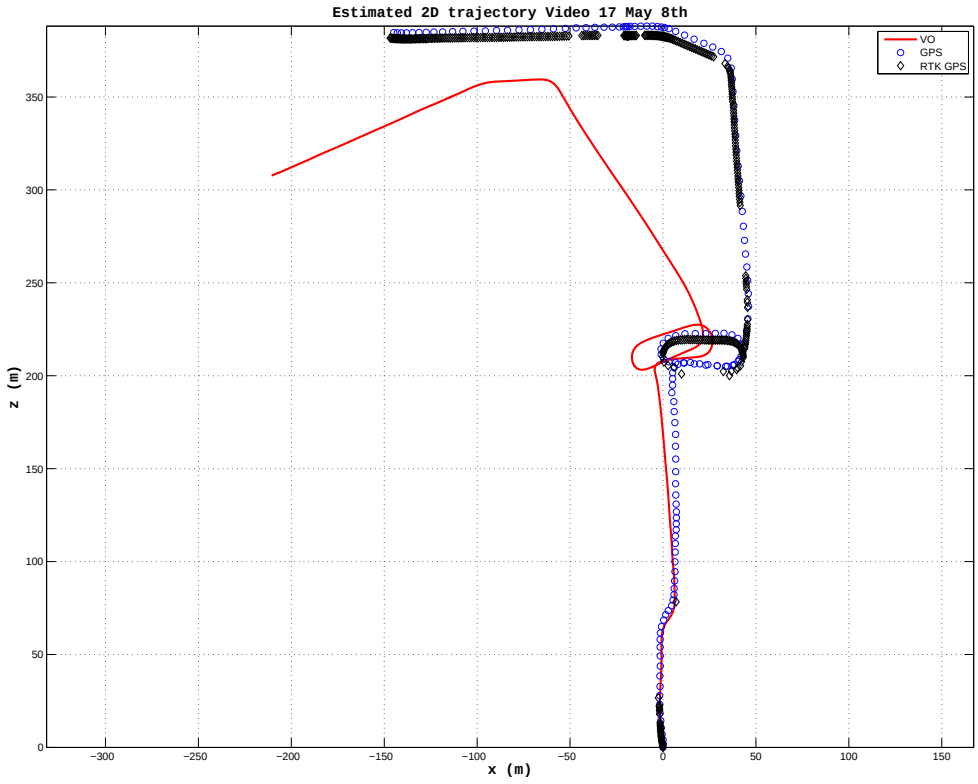
(d) Frame 552 (e) Frame 650 (f) Frame 879

(g) Frame 1312 (h) Frame 1416 (i) Frame 1472

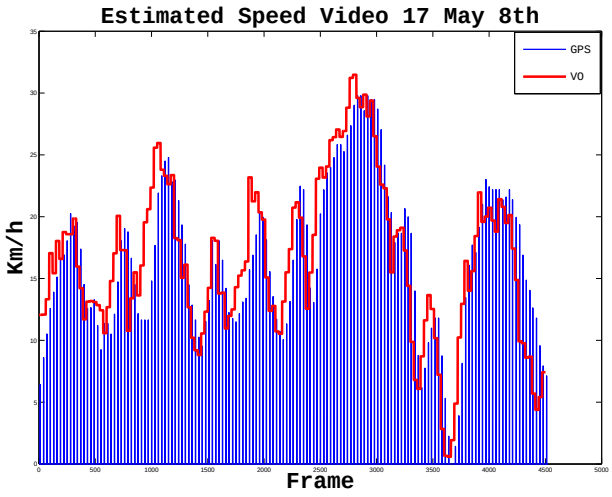
(j) Frame 2606 (k) Frame 2845 (l) Frame 3113

(m) Frame 3333 (n) Frame 3610 (o) Frame 4057

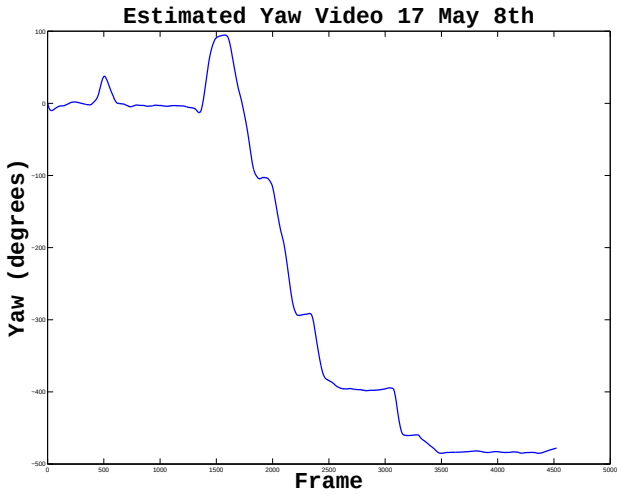
Figure 4.25: Frames from video 17 May 8th



(a) Estimated 2D trajectory for Video 17 May 8th.



(b) GPS and estimated velocity



(c) Estimated Yaw

Figure 4.26: Estimated velocity and yaw for video 17 May 8th

4.3 Conclusions

In this chapter three different feature extractor has been studied and tested for the specific task of feature detection and tracking in complex urban environments. A feature detection and tracking scheme using SIFT has been proposed and tested on real data. Based on the non-linear nature of the motion equations and the heterodasticity of 3D reconstruction a non-linear weighted least squares method has been proposed and tested both on synthetic and real data. The specific nature of a vehicle motion in urban environments has been analyzed and a calibration method has been proposed to estimate the extrinsic parameters of the stereo rig. Results for the global system and different feature extractors and image resolutions are discussed. The main conclusions that can be drawn from this chapter are as follows:

- SIFT outperforms SURF and Harris feature extractor, especially when the illumination conditions are poor. When the image is textured and the illumination is good SURF performance is similar lo SIFT.
- When working with 320×240 images resolution only SIFT sub-pixel accuracy is able to correctly estimate the 3D depth of the features and get an approximate estimation of the real length of the path. Harris and SURF underestimate the depth and the reconstructed motion shows a scaled version of the real one.
- The errors introduced by the linearization are small compared to those introduced by the features detection and triangulation. Thus, the effort has to be put on robust feature detection and 3D reconstruction algorithms.
- The weighted non-linear least squares solution has proven to be several times more accurate than the non-weighted one. The difference is especially noticeable when there are few input points to the algorithm. In this case, the weighted solution is able to use more inliers than the non-weighted one and delivers better motion estimations.
- The heterocedastic nature of the 3D reconstructed position makes the Mahalanobis distance a better way to measure the reconstruction error in the RANSAC step. The uncertainty in the position of closer points is smaller than the one for further points. Mahalanobis distance allows to be stricter with the outliers rejection threshold. Close outliers can be rejected and far inliers accepted at the same time, increasing the amount of inliers for the motion estimation step.
- The car motion can be modelled with 3 predominant parameters: forward translation, pitch and yaw. With this simplification it is possible to devise a method more robust to the hard conditions of urban environments. However, this model, assumes a coordinate frame in which the ground plane is parallel to the XZ plane of our camera coordinate system and that the optical axis is parallel to the Z axis. This is not true and requires that the points be rectified prior to computing the ego-motion. A calibration procedure has been proposed and tested to estimate the stereo rig extrinsic pitch and yaw.
- Results for sequences recorded in real traffic conditions in urban environments with pedestrians and other cars in the scene has been presented. The results show a high level of robustness and accuracy. Very complex scenarios have been presented with non-stationary cars and pedestrians, glares, overexposed and underexposed images,

etc. However the cumulative nature of the errors in visual odometry system makes necessary a correction if global localization for longer runs wants to be achieved.

Chapter 5

GPS assistance using OpenStreetMap

Many ITS applications and services such as route guidance, fleet management, road user charging, accident and emergency response, bus arrival information and other location based services require location information. In the last few years, GPS has become the main positioning technology for providing location data for ITS applications [Quddus 07]. However, due to signal blockage and severe multipath in urban areas, GPS can not satisfy most vehicle navigation requirements. Dead Reckoning systems have been widely used to bridge the gaps of GPS position error, but their drift errors increase rapidly with time and frequent calibration is required [Wu 03]. Visual odometry algorithms have proven to be capable of tracking the position of a vehicle over long distances using only the images as inputs and with no a priory knowledge of the environment [Agrawal 06]. Moreover, if combined with map matching algorithms cumulative errors of the visual odometry will be corrected and even longer distances could be travelled without the necessity of a correction of the absolute position.

Map matching algorithms use inputs generated from positioning technologies and supplement this with data from a high resolution spatial road network map to provide an enhanced positioning output. The general purpose of a map-matching algorithm is to identify the correct road segment on which the vehicle is travelling and to determine the vehicle location on that segment [Greenfeld 02] [Quddus 07]. Map-matching not only enables the physical location of the vehicle to be identified but also improves the positioning accuracy if good spatial road network data are available [Ochieng 04].

Our final goal is the autonomous vehicle outdoor navigation in large-scale environments and the improvement of current vehicle navigation systems based only on standard GPS. In areas where GPS signal is not reliable or even not fully available (tunnels, urban areas with tall buildings, mountainous forested environments, etc) this system will perform the localization during the GPS outrages.

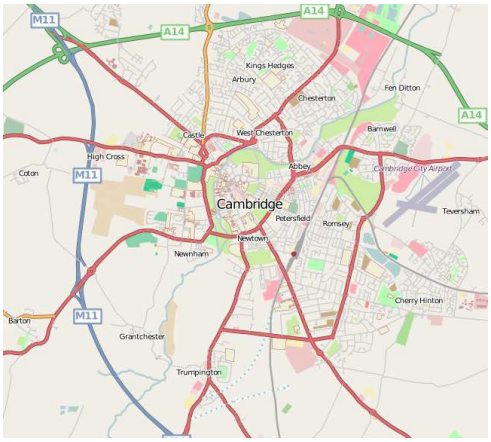
In the next sections the nature of the Geographical Information System (GIS) used, the map-matching algorithm and the geo-localization using the motion trajectory data are explained.

5.1 OpenStreetMap

OpenStreetMap (OSM) is a collaborative project to create a free editable map of the world. The maps are created using data from portable GPS devices, aerial photography,

other free sources or simply from local knowledge. OSM data is published under an open content license, with the intention of promoting free use and re-distribution of the data (both commercial and non-commercial).

Some government agencies have released official data on appropriate licenses. The United States government released Landsat 7 satellite imagery, Prototype Global shorelines (PGS) and Topologically Integrated Geographic Encoding and Referencing (TIGER) data of the United States. UK government have released a subset of their data products with an open source license (OS openData). In December 2006 Yahoo! confirmed that OpenStreetMap was able to make use of their vertical aerial imagery and this photography is now within the editing software as an overlay. Some commercial companies have donated data to the project on suitable licenses (most of them by Automotive Navigation Data AND) [wikipedia 10b].



(a) OpenStreetMap representation of Cambridge

(b) Steve Coast founder of OpenStreetMap

Figure 5.1: Images from [wikipedia 10b]

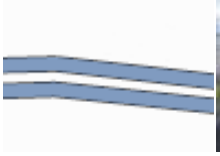
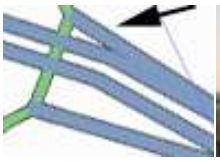
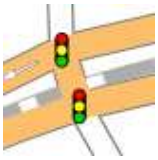
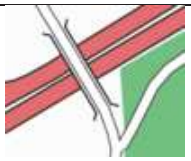
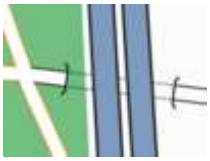
5.1.1 OpenStreetMap data representation

OSM uses a topological data structure along with longitude and latitude information. It uses the WGS 84 latitude/longitude datum exclusively. The amount of information stored in the maps varies from one area to another but at least the basic OSM-elements can be found:

- *Nodes* : Points with a geographic position expressed in latitude and longitude.
- *Ways* : Lists of nodes, representing a polyline or polygon. They can represent buildings, parks, lakes, streets, highways, etc.
- *Relations* : Groups of nodes, ways and other relations which can be assigned certain properties.

The features of the maps are expressed assigning tags to OSM-elements. You can use any tag you like but there are a recommended set of features in order to create, interpret and display a common base map. Tags can be applied to nodes, ways or relations and consist of key=value pairs. Examples of pieces of information stored in these tags are the

Table 5.1: Main tags to express features of the map elements [OpenStreetMap 10]

Key	Value	Element	Comment	Rendering	Photo
			Roads		
highway	motorway		A restricted access major divided highway, normally with 2 or more running lanes plus emergency hard shoulder.		
highway	motorway_link		The link roads leading to/from a motorway to/from a motorway or lower class highway. Normally with the same motorway restrictions		
highway	secondary		Generally linking smaller towns and villages		
			Intersections		
junction	roundabout		The way direction is defined by sequential ordering of nodes within the way.		
highway	traffic_signals		Lights that control the traffic		
			Properties		
bridge	yes		A bridge, use together with the layer tag. Value "yes" is generic, or you can specialize		
tunnel	yes		A tunnel, use together with the layer tag.		
			Restrictions		
maxspeed	speed		Maximum speed		
one-way	yes/no/-1		-1 for traffic direction opposite to the sequence of nodes		

kind of way (highway, secondary, tertiary), orientation (one-way, two-ways), name, speed limit (see Table 5.1).

Hundreds of features can be included into the map elements making the amount of available information huge. All the current raw OpenStreetMap data (nodes, ways, relations and tags) is stored in XML format and can be saved to .osm files. There are different ways in which you can get the maps:

- Download the whole word (http://planet.openstreetmap.org) and cut it into smaller chunks.
- OpenStreetMap (http://www.openstreetmap.org/export) allows to export a bounding box via its web interface.
- The API allows to get the data of a specific bounding box, so download managers can be used:

```
wget -O map.osm http://xapi.openstreetmap.org/api/0.5/map?bbox=11.4,48.7,11.6,48.9
```

To reduce their servers load OSM recommends to download only small areas.

5.1.2 OSM parsing and coordinates conversion

In order to be able to match the vehicle position in the map using the estimated motion trajectory we need to transform the latitude and longitude to Universal Space Rectangular XYZ coordinates and vice-versa. Also the xml map file has to be parsed and converted into Northing-Easting coordinates as a previous step to the map matching. The conversions from and to WGS-84 latitude, longitude and ellipsoid height to and from Universal Space Rectangular XYZ coordinates has been performed using [Laurila 76] ellipsoid approximation by 7 parameters.

The Earth's surface may be closely approximated by a rotational ellipsoid with flattened poles (height deviation from the geoid < 100m). As a result, geometrically defined ellipsoidal systems are frequently used instead of the spatial Cartesian coordinate system. For the determination of points on the physical surface of the Earth with respect to the rotational ellipsoid, the height h above the ellipsoid is introduced in addition to the geographic coordinates ϕ λ; h is measured along the surface normal (see Figure 5.2).

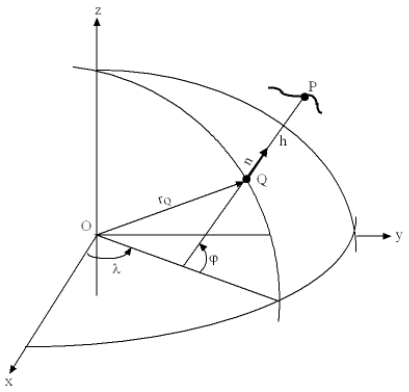


Figure 5.2: Spatial ellipsoidal (geodetic) coordinates.

The spatial ellipsoidal coordinates ϕ, λ, h are designated as geodetic coordinates. The point Q on the ellipsoid is obtained by projecting the surface point P along the ellipsoid normal: Helmert's projection.

The expression for point Q is as follows:

$$\begin{array}{rcl} X_Q & = & (N+h)\cdot\cos(\phi)\cdot\cos(\lambda) \\ Y_Q & = & (N+h)\cdot\cos(\phi)\cdot\sin(\lambda) \\ Z_Q & = & ((1-e^2)\cdot N+h)\cdot\sin(\phi) \end{array} \tag{5.1}$$

where N is the radius of curvature in the prime vertical and e is the first eccentricity.

The inverse problem is solved only by iteration; however, the system of equations converges quickly since $h \ll N$. From 5.1:

$$\begin{array}{rcl} h & = & \sqrt{X_Q^2+Y_Q^2}/\cos(\phi) - N \\ \phi & = & \arctan(Z_Q)/\sqrt{X_Q^2+Y_Q^2} \cdot \left(1-e^2\cdot\dfrac{N}{N+h}\right)^{-1} \\ \lambda & = & \arctan\dfrac{Y_Q}{X_Q} \end{array} \tag{5.2}$$

[Bowring 85] has given solutions for geodetic latitude and longitude that are particularly stable. For further details on these equations please refer to [Torge 91].

This geodetic latitude and longitude are converted to UTM coordinates using an approximation from the US Geological Survey 1532. This conversion equations were written in C by Chuck Gantz:

$$\begin{array}{l} UTMEasting = K0\cdot N\cdot(A+(1-T+C)\cdot\dfrac{A^3}{6}+ \\ \qquad (5-18\cdot T+T^2+72\cdot C-58\cdot ep^2\cdot\dfrac{A^5}{120})+500000 \\ UTMNorthing = K0\cdot\big(M+N\cdot tan(\lambda)\cdot(\dfrac{A^2}{2}+(5-T+9\cdot C+4\cdot C^2)\cdot\dfrac{A^4}{24} \\ \qquad +\big(61-58\cdot T+T^2+600\cdot C-330\cdot ep^2\big)\cdot\dfrac{A^5}{720})\big) \end{array} \tag{5.3}$$

where $K0 = 0.9996$ is the central meridian scale, e is the eccentricity and:

$$\left\{\begin{array}{l} ep = \dfrac{e^2}{1-e^2} \\ N = \dfrac{a}{\sqrt{1-e^2\cdot\sin^2(\lambda)}} \\ T = \tan^2(\lambda) \\ C = ep\cdot\cos(\lambda) \\ A = \cos(\lambda)-\phi \\ M = a\cdot\big((1-e^2/4-3e^4/64-5e^6/256)\cdot\lambda- \\ \qquad (3e^2/8+3e^4/32+45e^6/1024)\cdot\sin(2\lambda)+ \\ \qquad (15e^4/256+45e^6/1024)\cdot\sin(4\lambda)- \\ \qquad (35e^6/3072)\cdot\sin(6\lambda)\big) \end{array}\right. \tag{5.4}$$

In Figure 5.3 a renderized OSM map of the University of Alcalá and its converted UTM representation as used for the map-matching algorithm are depicted.

(a) Original OSM map of the University of Alcalá Campus

(b) UTM coordinates converted map of the University of Alcalá campus

Figure 5.3: Example of coordinates conversion for a map of the University of Alcalá campus

5.2 Visual Odometry and map matching

5.2.1 Introduction to map-matching

Map-matching algorithms integrate the position information with spatial road network data to identify the correct link a vehicle is travelling and to determine the location of a vehicle on a link. Approaches for map-matching can be categorised into four groups:

- *Geometric analysis*: Use the geometric information of the spatial road network data by considering only the shape of the links [Greenfeld 02]. It does not consider the way the links are connected to each other. The easiest to implement and fastest approach is known as *point to point matching* [D. Bernstein 02]. In this approach, each position is matched to the closest node of a road segment. In practice, this approach is very sensitive to the way the points are defined in an arc. Arcs with more shape points are more likely to be matched (see Figure 5.4).

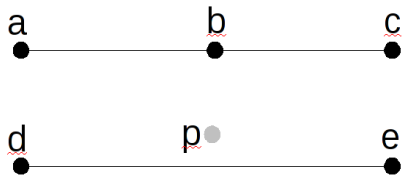
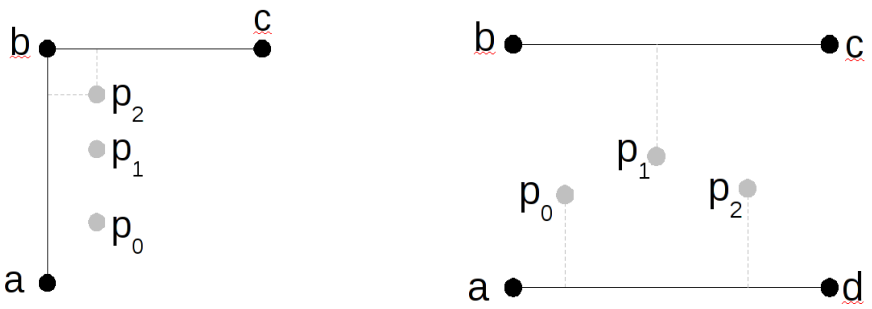


Figure 5.4: *Point to point* map matching problem. Position p will be snapped to node b.

Another geometric approach is to match the position to the closest curve in the network, what is known as *point-to-curve matching* [D. Bernstein 02]. The distance from the position fix to the closest segment of the road is selected as the one on which the vehicle is travelling. Although this approach yields better results than point-to-point matching, it is unstable in urban networks and it does not take into account the historical information (see Figure 5.5).



(a) No historical record problem

(b) Instability problem

Figure 5.5: Examples of problems in *point-to-curve* map-matching

The last geometric approach is to compare the vehicle's trajectory against known roads, known as *curve to curve matching* [D. Bernstein 02] [White 00]. This approach firstly identifies the candidate nodes using point-to-point matching. Then it constructs two curves and determine their distance. This approach is sensitive to outliers and can give unexpected results (see Figure 5.6) [Quddus 07].

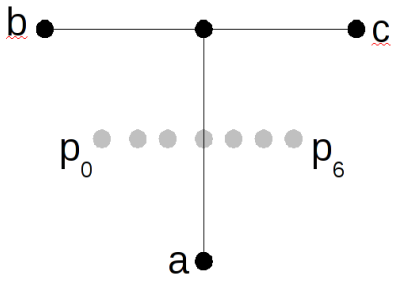


Figure 5.6: *Curve to curve* map-matching problem

- *Topological analysis*: Topology refers to the relationship between elements (nodes, lines and polygons). The relationship can be defined as adjacency, connectivity or containment. Therefore, a map-matching algorithm which makes use of the geometry of the links as well as the connectivity and contiguity of the links is known as topological map-matching algorithm. This algorithms are usually based on features of the road (road turn, road curvature and road connection) and the vehicle trajectory, velocity and heading. The correlation between the trajectory of the vehicle and topological features of the road gives the estimated position of the car on the road [Meng 06]. [Greenfeld 02] proposes a weighted topological algorithm in which different weighting factors are used to control for the importance of each of the criteria.

- *Probabilistic map-matching algorithms*: This technique was first introduced by [Honey 85] in other to match positions from a DR sensor to a map. This kind of map-matching algorithm define an elliptical or rectangular confidence region around a position fix obtained from a navigation sensor. If the error region contains s number of segments, then the evaluation of candidate segments are carried out using heading, connectivity and closeness criteria.

[Ochieng 04] developed an enhanced probabilistic map-matching algorithm in which the elliptical region is only constructed when the vehicle travels through a junction (in contrast to [Zhao 97] in which it was constructed for each position fix). This method is more reliable as the construction of an error region at each step can lead to incorrect link identification if other links are close to the one on which the vehicle is travelling.

- *Advanced map-matching algorithms*: Advanced map-matching algorithms are referred to as those algorithms that use more refined concepts such as a Kalman Filter or an Extended Kalman Filter [Kim 00], Dempster-Shafer's mathematical theory of evidence [Yang 03], particle filters [Gustafsson 02] or Bayesian inference [Pyo 01]. In this line, and related to the current work [Gustafsson 02] developed a map-matching algorithm using a particle filter. One of their applications was to correctly estimate the initial unknown position of the vehicle given an initial estimation in a region of about 2 Km. This initial position could be retrieved from terrestrial wireless communication systems or manually introduced by the user. In this way this method was able to supplement and replace the GPS.

5.2.2 Visual Odometry integration in map-matching

Traditionally GPS and DR has been used as input to map-matching algorithms however the conventional integration does not correct the position after re-location. Given the cumulative nature of errors in visual odometry estimations, the drift will keep increasing without bounding. Moreover, the complex nature of the urban environment and the numerous non-static objects (other cars, pedestrians,..) will make the map-matching process unreliable and eventually loose the vehicle position. If accurate localization is needed for long periods of GPS outrage additional information available in the digital map has to be used to correct the actual vehicle position and reset the cumulative errors from visual odometry. Otherwise, small misestimations due to poor quality of the input images (rain, glares,...) or non-static objects, can quickly lead to mislocalizations.

In our approach, we propose a probabilistic map-matching algorithm constrained to the road which uses map features to control the errors of the visual odometry by feeding back corrections from the map-matching process. Every time the map-matching algorithm correctly matches the vehicle position at one of these features the vehicle position and heading is corrected. This is idea is based on the previous work in [Wu 03] where GPS and DR were fused using a similar approach.

Integration of Visual Odometry and GPS

The signal from each GPS satellite has a level of precision depending on the relative geometry of the satellites. When visible GPS satellites are close together in the sky, the geometry is said to be weak and the dilution of precision (DOP) value is high; when far apart, the geometry is strong and the DOP value is low. In table 5.2 the usual ratings for DOP values are shown.

In our system when the horizontal DOP (HDOP) is greater than 10 the signal is considered not reliable and the position in the map is computed using the visual odometry information (see Figure 5.7).

Table 5.2: Meaning of DOP Values

DOP Value	Rating
1	Ideal
1-2	Excellent
2-5	Good
5-10	Moderate
10-20	Fair
>20	Poor

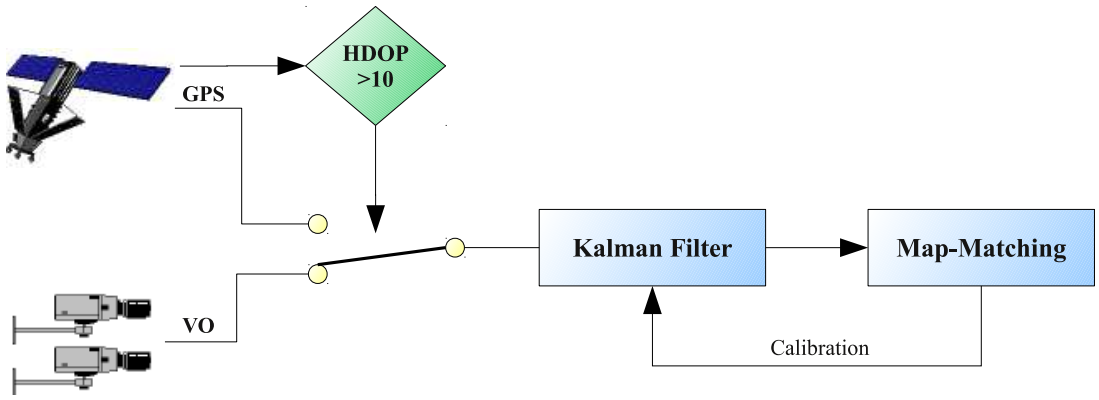


Figure 5.7: Integration of the GPS and VO measures

Identification of the actual link

The most complex element of any map-matching algorithm is to identify the actual link among the candidate links [Greenfeld 02] . In our map-matching algorithm 3 basic assumptions are made:

1. The vehicle travels on the road most of the time.
2. The vehicle can not jump from one place to another one with no connection.
3. The vehicle has to follow certain road rules.

Firstly the initial road segment in which the vehicle is travelling is estimated through an *initialization process*. When the GPS fix is lost the elliptical confidence region of the visual odometry estimation is computed using 4.14 and the last reliable GPS fix. The confidence region is projected into the map and the road segments that are within the confidence region are taken as candidate regions. For simplicity the the elliptical confidence region is approximated to a rectangular one (see Figure 5.8).

If the confidence region contains more than one candidate segment the heading over the last 5 seconds is computed and matched to the segments orientation. If there is only one candidate left after the heading check that is the initial road segment. If not the distance from the motion trajectory to the segments is computed as follows:

1. Compute the starting point using the point-to-curve algorithm for the last GPS fix.
2. Compare the estimated run distance to the distance left of the starting point segment. If it is greater than the distance left discard the starting point segment and go to step 3. Otherwise the starting point segment is the initial segment.

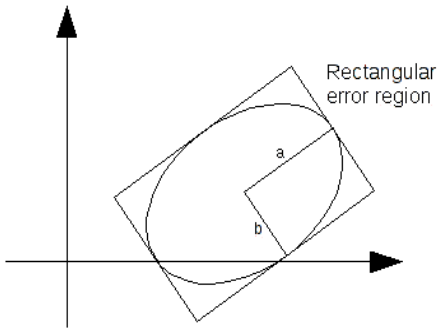


Figure 5.8: Elliptical confidence region and rectangular approximation

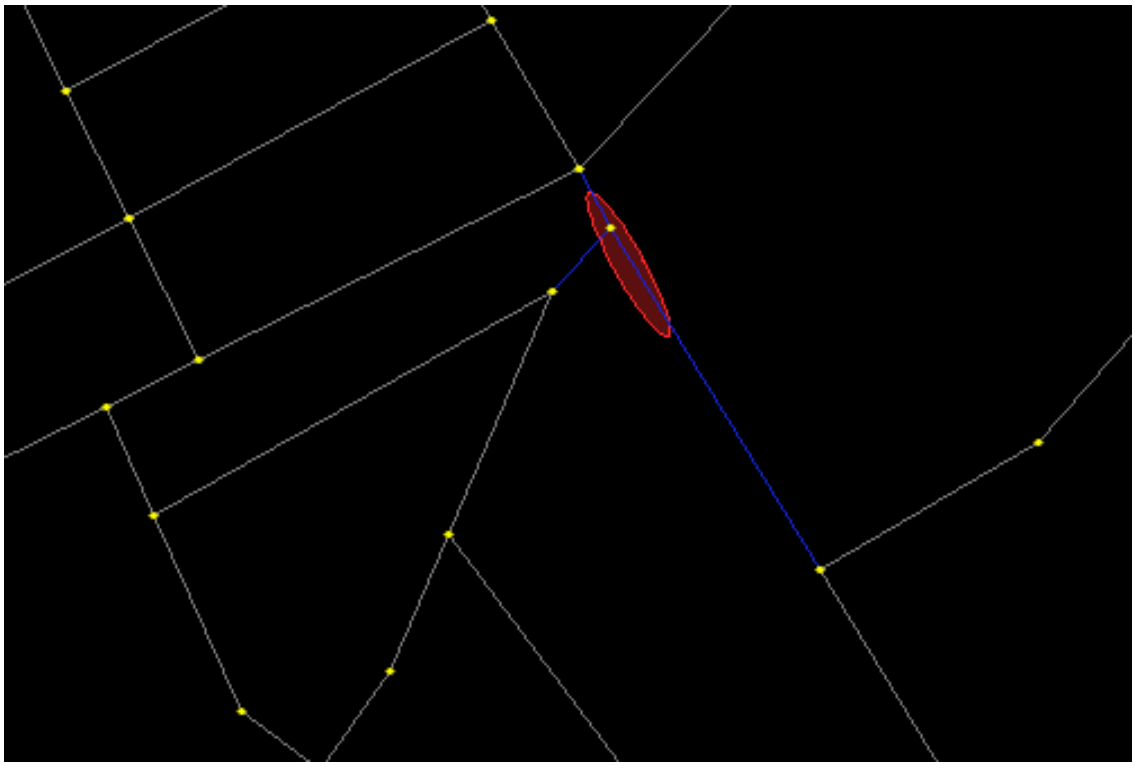


Figure 5.9: Elliptical confidence region (red) and candidate segments (blue) over the converted OSM map of Alcalá de Henares

3. Compute the distance from the motion trajectory estimation to the candidate segments by computing the area under the motion estimation trajectory to a stretch of each one of the candidate segments. This stretch will have the same length as the motion trajectory estimation for all the candidate segments (see Figure 5.10)[D. Bernstein 02].
4. Select the segment closer to the curve as the initial segment.

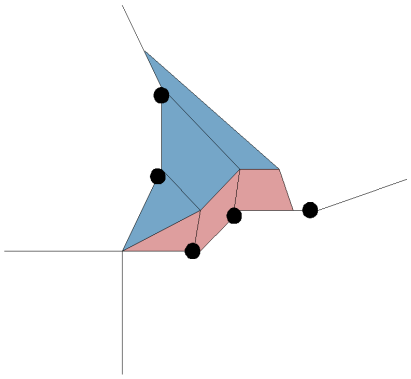


Figure 5.10: Curve-to-curve implemented map-matching algorithm

Tracking of the vehicle position in the map

After setting the initial position of the vehicle in the map subsequent motion estimations from the visual odometry are matched in the map following a different approach. Firstly the vehicle velocity, heading and position uncertainty are used to estimate if the vehicle is turning or driving trough a junction. If so, the identification of the actual link is started. Otherwise a simple tracking of the vehicle position in the map is performed (see Figure 5.11). The steps of this process are:

1. If the difference between the heading of the vehicle and the current road segment is higher than a threshold or there is at least one juncture in the uncertainty region start the identification of the actual link. If this process was triggered by the heading but not by the uncertainty region increase the uncertainty region a 20%. If not continue.
2. Using the vehicle heading and velocity, check the predicted position and the measured one, if close feedback the position to the visual odometry. The position in the road is computed using the *point-to-curve* algorithm and the heading is the road segment orientation.

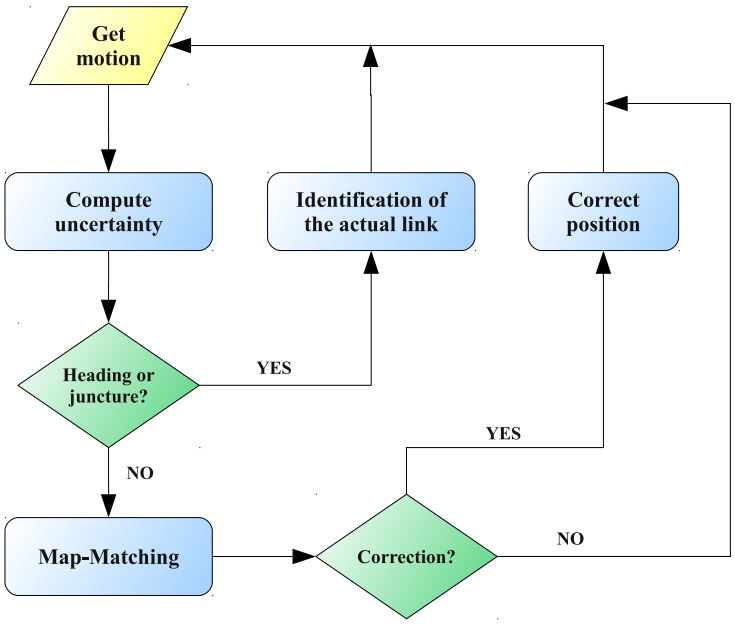


Figure 5.11: Map-matching flow diagram

5.3 Results

In this section the motion estimation results from Chapter 4 are used as input to the described map-matching algorithm. The map-matching algorithm output (latitude and longitude positions) was fed to the Java interface to OSM maps Travelling Salesman [Salesman 10], which performed the map rendering and trajectory representation.

Urban (320×240)

The results shown in this section were recorded using 4.2 mm lenses at 30 fps with a resolution of 320x240 to reduce the computing time. Two experiments of about 200 m each have been selected to display the result of the visual odometry and the map-matching.

On Fig. 5.12 the results for the first experiment are shown. On this experiment the car was driven along a urban cannon in a path of approximately 165m. The distance measured by the visual odometry system was 163.37 m, 99% of the real one. Also the estimation of the turning was very accurate as can be seen on Fig. 5.12. On top the vehicle trajectory over imposed on google maps is displayed. In the middle the vehicle motion trajectory in the OSM map and the GPS information as shown to the user. Below the raw visual odometry information used to get the vehicle position in the map during the GPS outrage. Note that this motion information is not absolute and thus it is represented as a motion in meters from the starting point $(0, 0)$ facing *forward*. The global position reconstruction is accurate and the junction turnings are estimated correctly.

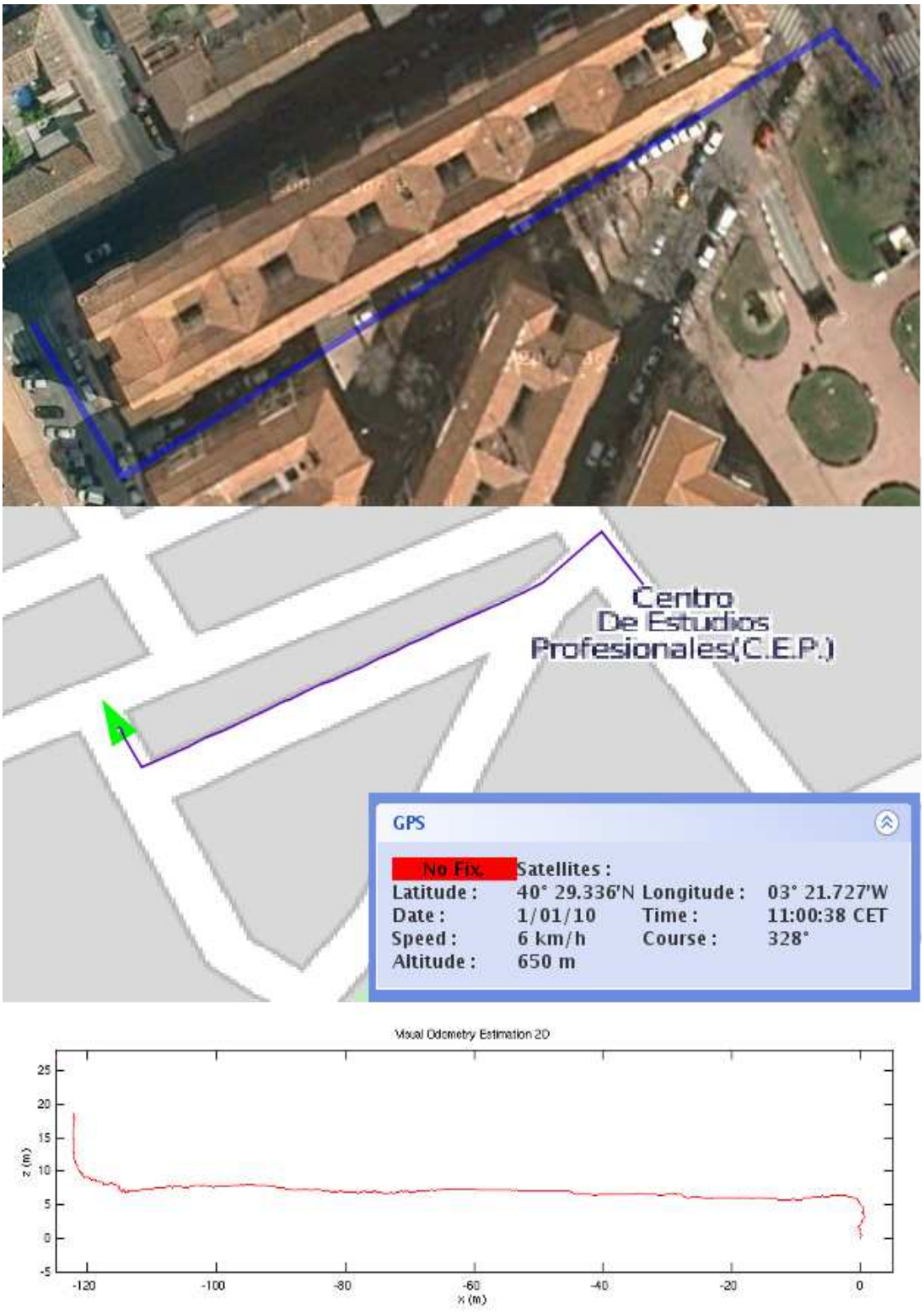


Figure 5.12: On top the vehicle real trajectory displayed in google maps. In the middle the vehicle motion trajectory in the OSM map and the GPS information as shown to the user. Below the raw visual odometry information used to get the vehicle position in the map during the GPS outrage

On Fig. 5.13 the results of the second experiment are shown. In this case the input video images are of poor quality as a result of glares on the windscreen and dazzling of the cameras (see Fig. 5.14). On this experiment the car was driven along a urban cannon for approximately 229m. The distance measured by the visual odometry system for the outrage was 197.31 m, 86.16% of the real one. Due to the poor quality of the input images the number of features tracked was very small on the long straight leading to underestimation of the distance run. However the estimation of the turnings was accurate. As can be seen on Fig. 5.13 the error introduced by the bad illumination conditions is resolved using the topological information of the map resulting on a re-localization at the next turn when the map fusion corrects the, otherwise, cumulative error of the visual odometry. This allows the system to work autonomously for very long distances, correcting the cumulative error using the topological information of the map.

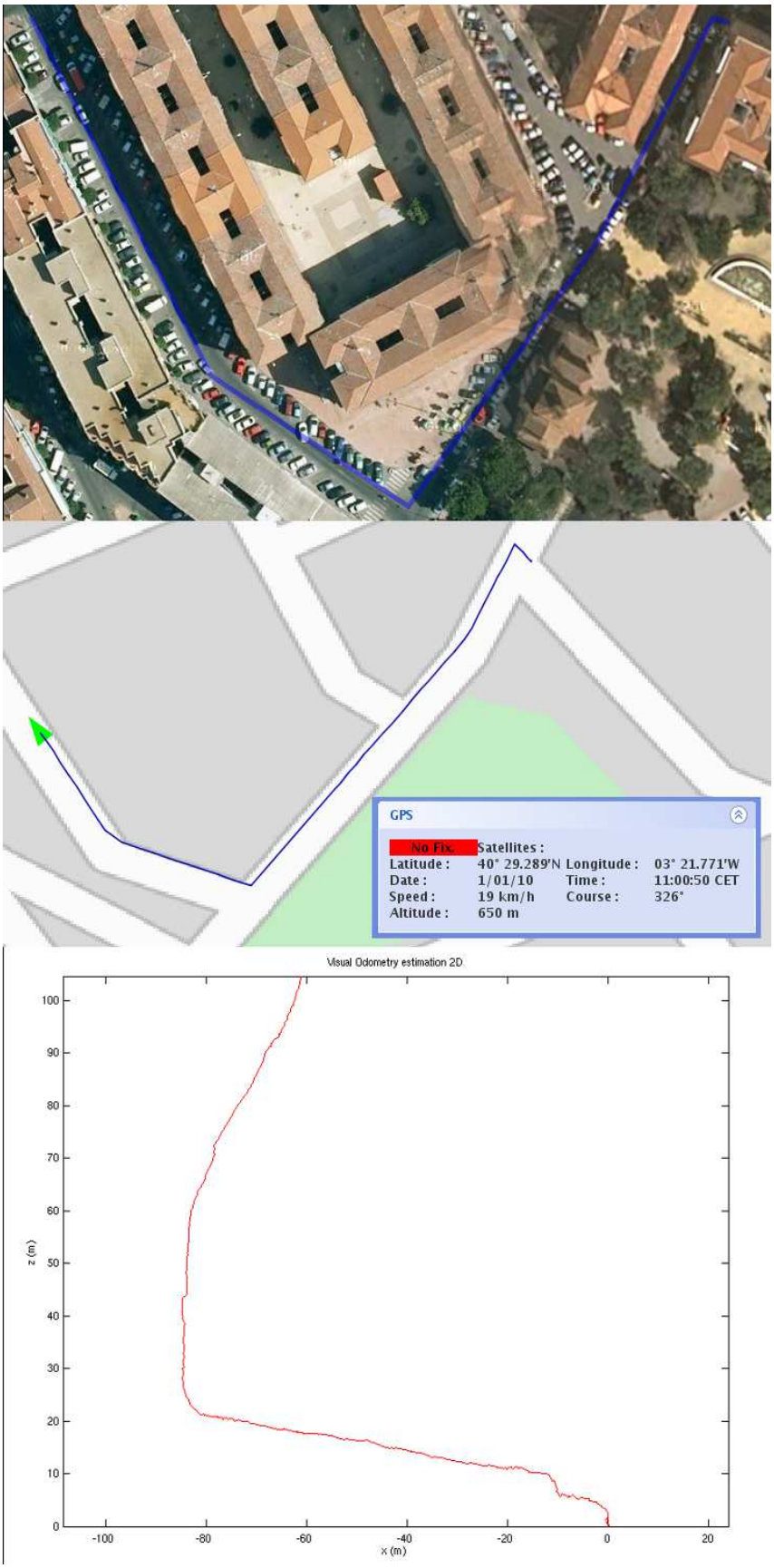


Figure 5.13: On top the vehicle real trajectory displayed in google maps. In the middle the vehicle motion trajectory in the OSM map and the GPS information as shown to the user. Below the raw visual odometry information used to get the vehicle position in the map during the GPS outrage

Figure 5.14: Example of glares and dazzling on the images

Tunnel

On the next experiment we will show the map-matching results for the video 05 of May 8th which visual odometry results have been shown in section 4.2.6. In this experiment the car was driven along an approximately 680m path with other non-stationary cars, and went through a roundabout and a tunnel. The visual odometry results used for the map-matching were obtained using SIFT features and 640×480 images. The estimated motion trajectory was accurate but the length of the tunnel was slightly overestimated. As can be seen on figure 5.15 the global position of the vehicle is tracked with no mistakes and the errors of the visual odometry are corrected by the map matching algorithm. The map-matching algorithm correctly estimates all the turnings and the exit for the roundabout. Other cars and buses present on the video sequenced don't affect the global localization accuracy.

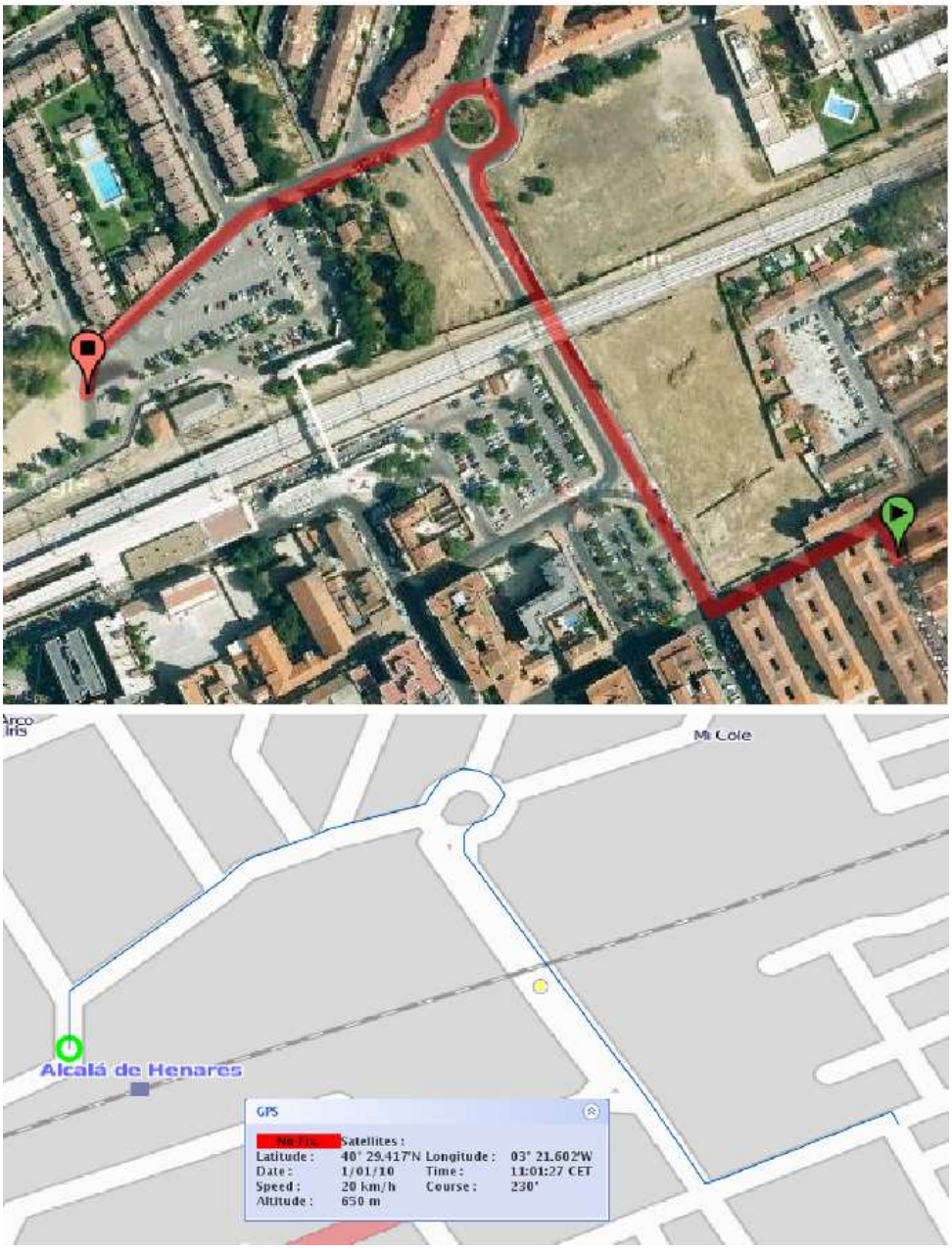


Figure 5.15: On top the vehicle real trajectory displayed in google maps. Bellow the middle the vehicle motion trajectory in the OSM map and the GPS information as shown to the user.

Urban (640×480)

On this experiment we will show the results for the video 15 of May 8th. Its visual odometry results have been shown in section 4.2.6. In this experiment the car was driven through a urban canyon for approximately 418m. The visual odometry results used for the map-matching were obtained using SIFT features and 640×480 images. The estimated motion trajectory and length was accurate (error 0.41%). As can be seen on figure **??** the global position of the vehicle is tracked with no mistakes and all the turnings are correctly estimated.

(a) Google Maps trajectory representation

(b) Travelling Salesman output

Figure 5.16: Map-matching results for video 15 May 8th

5.4 Conclusions

In this chapter a map-matching algorithm was presented which performs robust global localization using a digital map and the output from the visual odometry system. The digital map (OSM) is parsed and converted into Northing-Easting coordinates to perform the map-matching. The map-matching is performed using a probabilistic approach which combines the heading and velocity information of the vehicle with the topological information of the digital map. Map features are used to control the error of the visual odometry by feeding-back corrections from the map matching processes. The main conclusions that can be drawn from this chapter are as follows.

- When working with SIFT 320×240 the accuracy in the motion trajectory reconstruction is enough for the map matching algorithm which corrects the misestimations in the length of the path.
- The map-matching algorithm have shown accurate results for situations in which a GPS may fail such as urban canyons or tunnels.

- The map-matching algorithm has proven capable of tracking the global position of the vehicle using visual odometry. However extensive testing has to be performed for a commercial application. Even though the visual odometry estimation is accurate many situations such as inaccuracies in the maps, missing links, new roads or re-localization after a fail have to be addressed.

Chapter 6

Conclusions

This chapter presents the global conclusions and discuss the main contributions introduced and developed along the chapters of this thesis. Finally, we will draw futures line of research that this thesis leaves open.

6.1 Sensor modeling

The camera model and the stereo geometry were presented and discussed to understand the influence of the different design parameters in the final performance. An exhaustive study of the influence of the different parameters was performed. This study is powerful tool when designing stereo systems and allows for a tuning of the different parameters. Also the 3D reconstruction uncertainty has been explained and a multivariate Gaussian model proposed to describe it. This model have proven to be a good approximation when the distance to the points is not extreme.

6.2 Feature Extractors

Three different feature extractors were studied and tested for the specific task of feature detection and tracking in complex urban environments. A feature detection and tracking scheme using SIFT was proposed and tested on real data. The results show that SIFT outperforms Harris and SURF feature extractors, specially when working with overexposed or underexposed images. When working with 320×240 images only SIFT sub-pixel accuracy is able to correctly estimate the 3D depth of the features and get an approximate estimation of the real length of the path. Harris and SURF underestimate the depth and the reconstructed motion shows a scaled version of the real one. This is due to the failure of the Gaussian model to estimate the longer tails of the 3D uncertainty for distant points.

6.3 Visual Odometry

A MatLab simulator was developed and tested both on real and synthetic data. The results of the simulations showed that the solution of the non-linear system introduces a very small error in the motion estimation and most of the error comes from the inaccuracy in the 3D position estimation. A RANSAC based weighted non-linear least squares solution was proposed and tested both in real and synthetic data. The weighted scheme showed to be a better solution for the motion estimation due to the heterodasticity in the input data. Results showed a 20 times improvement in the mean distance to the ground truth. A

Mahalanobis distance for the RANSAC was introduced and tested to better represent the Gaussian multivariate nature of the 3D input data. A simplified car motion model with 3 parameters (pitch, yaw and forward motion) was presented. A calibration of the cameras rig extrinsic pitch and yaw was proposed and tested to comply with the requisites of this model.

Video sequences were recorded and the algorithms tested on very different situations. Results show that the weighted solution is very accurate in the presence of outliers (moving cars, pedestrians) which is of crucial importance for a urban visual odometry system.

6.4 Map matching

A probabilistic map-matching algorithm using the heading of the vehicle and its velocity was developed and tested. Map features were used to control the error of the visual odometry by feeding back corrections from the map-matching process. The map-matching algorithm have shown accurate results for situations in which a GPS may fail such as urban canyons or tunnels working with SIFT features and 320×240 images. The map-matching algorithm has proven capable of tracking the global position of the vehicle using visual odometry

6.5 Future work

From the results and conclusions of the present work, several lines of work can be proposed:

- With respect to the feature detection and tracking testing new feature extractors such as CenSurE [Agrawal 08] would be of great interest.
- Trying bundle adjustment methods where both the poses of the cameras and the 3D points are optimized, could improve results. Also testing a monocular version of the system and its performance would be very interesting specially for commercial applications.
- Longer experiments, of tenths of kilometers, should be performed to test the robustness of the system. Also to test the system performance at night time would be very interesting.
- Another application for this system is the estimation of the pitch and yaw of the vehicle for other ADAS systems such as pedestrian detection or lane departure warning. Checking the accuracy of the pitch and yaw estimations with an IMU will give an idea of the precision of the instantaneous estimation and its utility for other systems.
- Performing an extensive test of the map-matching algorithm with hours of position estimations. The tests carried out up to date have been very short compared to the usual length of a typical car travel.
- To investigate the possibility of estimating the initial unknown position of the vehicle in the map using SLAM techniques such a particle filter and the inputs form the visual odometry.

Bibliography

[Agrawal 05] Motilal Agrawal, Kurt Konolige & Luca Locchi. *Real-Time Detection of Independent Motion Using Stereo*. In IEEE Workshop on Motion (WACV/MOTION), 2005.

[Agrawal 06] M. Agrawal & K. Konolige. *Real-time localization in outdoor environments using stereo vision and inexpensive gps*. 18th International Conference on Patter Recognition (ICPR06), pages 1063–1068, August 2006.

[Agrawal 08] Motilal Agrawal, Kurt Konolige & Morten Rufus Blas. *CenSurE: Center Surround Extremas for realtime feature detection and matching*. Computer Vision ECCV, vol. 5305, pages 102–115, October 2008.

[Aitken 35] A. C. Aitken. *On Least Squares and Linear Combinations of Observations*. Proceedings of the Royal Society of Edinburgh, vol. 55, pages 42–48, 1935.

[Baird 85] H.S. Baird. Model-based image matching using location. MIT Press, Cambridge, MA, 1985.

[Baker 03] C.F. Patrick Baker, Abhijit S. Ogale & Y. Aloimonos. *New eyes for robotics*. In Proceeding of the Intelligent Robots and Systems (IROS), volume 1, pages 1018–1023, 2003.

[Bay 06] Herbert Bay, Tinne Tuytelaars & Luc Van Gool. *SURF: Speeded Up Robust Features*. In ECCV, pages 404–417, 2006.

[Beis 97] J. S. Beis & David G. Lowe. *Shape indexing using approximate nearest-neightbour search in high-dimensional spaces*. In Proceedings of the IEEE Conference on CVPR. pages 1000-1006, 1997.

[Blostein 87] Steven D. Blostein & Thomas S. Huang. *Error Analysis in Stereo Determination of 3-D Point Positions*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, pages 752–765, November 1987.

[Boufama 94] B. Boufama. *Reconstruction Tridimensionnelle en Vision par Ordinateur: Cas des Cameras Non Etalonnees*. In PhD thesis. INP de Grenoble, France, 1994.

[Bowring 85] B. Bowring. *The accuracy of geodetic latitude and height equations*. Survey review, vol. 28, pages 202–206, 1985.

[Broida 86] T.J. Broida & R. Chellappa. *Estimation of motion parameters in noisy images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, pages 90–99, January 1986.

[Clemente 07] L.A. Clemente, A.J. Davison, I.Reid, J. neira & J.D. Tardós. *Mapping large loops with a single hand-held camera*. Proc. Robotics: Science and Systems Conference, June 2007.

[Corke 04] P.I. Corke, D. Strelow & S. Singh. *Omnidirectional visual odometry for a planetary rover*. Proceedings of Intelligent Robotics and Systems, vol. 4, pages 4007–4012, October 2004.

[D. Bernstein 02] A. Kornhauser D. Bernstein. *An introduction to map matching for personal navigation assistants*, 2002.

[Davison 03] A. Davison. *Real-time simultaneous localisation and mapping with a single camera*. Proceedings of the International Conference on Computer Vision, vol. 2, pages 1403–1410, October 13-16 2003.

[Demirdjian 01] D. Demirdjian & T. Darrell. *Motion estimation from disparity images*. Proceedings of the IEEE International Conference on Computer Vision, pages 213–218, July 2001.

[Dhome 03] M. Dhome, J.T. Lapresté & J.M. Lavest. Calibrage des caméras ccd. LASMEA, Blaise Pascal University of Clermont-Ferrand. France, 2003.

[Estrada 05] C. Estrada, J. Neira & J.D. Tardós. *Hierarchical SLAM: real-time accurate mapping of large environments*. IEEE Transactions on Robotics, vol. 21, no. 4, pages 588–596, August 2005.

[Faugueras 88] O. D. Faugueras & F. Lustman. *Motion and structure from motion in a piecewise planar environment*. International Journal of Pattern Recognition and Artificial Intelligence, vol. 3, pages 485–508, 1988.

[Filzmoser 03] P. Filzmoser, C. Reimann & R.G. Garrett. *Multivariate outlier detection in exploration geochemistry*. Technical report TS 03-5, December 2003.

[Fischler 81] M. A. Fischler & R.C. Bolles. *Random Sample Consensus: A paradigm for model fiting with applications to image analysis and automated cartography*. Communication of the ACM, June 1981.

[Folkesson 05] J. Folkesson, P. Jensfelt & H. Christensen. *Visual SLAM in the measurement subspace*. Proceedings of the IEEE Journal of Robotics and Automation, pages 30–35, 2005.

[Forsyth 03] D. A. Forsyth & J. Ponce. Computer vision. a modern approach. Prentice Hall, Pearson Education International, 2003.

[Garrett 89] R.G. Garrett. *The chi-square plot: A tool for multivariate outlier recognition*. Journal of Geochemical Exploration, vol. 32, pages 319–341, 1989.

[Gennery 80] D.B. Gennery. *Modelling the Environment of an exploring vehicle by means of stereo vision*. PhD thesis, Standford University, Stanford, CA, June 1980.

[Gordon 06] I. Gordon & David G. Lowe. *What and where: 3D Object recognition with accurate pose*. In International Symposium on Mixed and Augmented Reality, 2006.

[Greenfeld 02] Joshua S. Greenfeld. *Matching GPS Observations to Locations on a digital map*. Proceedings of the 81st Annual Meeting of the Transportation Research Board, January 2002.

[Gustafsson 02] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson & P. Nordlund. *Particle filters for positioning, navigation and tracking*. IEEE Transactions on Signal Processing, no. 50, pages 425–435, 2002.

[Haralick 94] R.M. Haralick, C.N. Lee, K. Ottenberg & M. Nolle. *Review and analysis of solutions of the three point perspective pose estimation problem*. In International Journal of Computer Vision, volume 13, pages 331–356, 1994.

[Hariis 88] C. Hariis & M. Stephens. *A Combined Corner and Edge Detector*. Proceedings of the Fourth Alvey Vision Conference, pages 147–151, 1988.

[Hartley 03] R. Hartley & A. Zisserman. Mulitple view geomwtry in computer vision. Cambridge University Press, 2003.

[Hartley 04] R. Hartley & A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, 2004.

[Honey 85] S.K. Honey, W.B. Zavoli, K.A. Milnes, A.C. Phillips, M.S. White & G.E. Loughmiller. *Vehicle Navigation System and Method*, May 1985.

[Horn 88] B.K.P. Horn & Jr E.J. Weldon. *Direct methods for recovering motion*. In International Journal of Computer Vision, volume 2, pages 51–76, 1988.

[Karlsson 05] N. Karlsson, E. di Bernardo, J. Ostrowsky, L. Goncalves, P. Pirjanian & M. Munich. *The vSLAM algorithm for Robust Localization and Mapping*. Proceedings of the IEEE Journal of Robotics and Automation, pages 24–29, 2005.

[Kim 00] W. Kim, G. Jee & J. Lee. *Efficient use of digital road map in various positionings for ITS*. In IEEE Symposium on Position Location and Navigation, San Diego, CA, 2000.

[Konolige 07] Kurt Konolige, Motilal Agrawal & Joan Solà. *Large Scale Visual Odometry for Rough Terrain*. In Proceedings of the International Symposium on Research in Robotics (ISRR), 2007.

[Labrosse 06] F. Labrosse. *The visual compass: performance and limitations of an appearance based method*. Journal of Field Robotics, vol. 23, no. 10, pages 913–941, 2006.

[Laurila 76] H. Simo Laurila & Thomas A. Stansell. Electronic surveying and navigation. John Wiley & Sons, 1976.

[Lemaire 07] Thomas Lemaire & Simon Lacroix. *SLAM with panoramic vision.* Journal of Field Robotics, vol. 24, pages 91–111, 2007.

[Llorca 08] D.F. Llorca. *Sistema de deteción de peatones mediante visión estereoscópica para la asistencia a la conducción.* PhD thesis, University of Alcalá, Alcalá de Henares, September 2008.

[Llorca 09] D.F. Llorca, M.A. Sotelo, I. Parra, J.E. Naranjo, M. Gavilán & S. Álvarez. *An experimental study on Pitch Compensation in Pedestrian-Portection Systems for Collision Avoidance and Mitigation.* IEEE Transactions on Intelligent Transportation systems, vol. 10, no. 3, pages 469–474, September 2009.

[Llorca 10] D.F. Llorca, M.A. Sotelo, I. Parra, M. Oca na & L.M. Bergasa. *Error Analysis in a Stereo Vision-Based Pedestrian Detection Sensor for Collision Avoidance Applications.* Journal Sensors, vol. 10, pages 3741–3758, April 2010.

[Longuet-Higgins 86] H.C. Longuet-Higgins. *The reconstruction of a plain surface from two perspective projections.* Royal Society London, vol. 277, pages 399–410, 1986.

[Lowe 99] David G. Lowe. *Object recognition from local scale-invariant features.* In Proceedings of the Seventh ICCV. pages 1150-1157, 1999.

[Lowe 04] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints.* In International Journal of Computer Vision, volume 60, pages 91–110, 2004.

[Lucas 81a] B.D. Lucas & T. Kanade. *An iterative image registration technique with an aplication to stereo vision.* In Proceedings of the International Joint Conference on Artifial Intelligence. pages 674-679, 1981.

[Lucas 81b] B.D. Lucas & T. Kanade. *An iterative image registration technique with an application to stereo vision.* Seventh International Joint Conference on Artificial Intelligence, vol. 2, pages 674–679, August 1981.

[Mahalanobis 36] P.C. Mahalanobis. *On the generalized distanve in statistics.* Proceedings of the National Institute of Science of India, pages 49–55, December 1936.

[Maomone 07] M. Maomone, Y. Cheng & L. Matthies. *Two years of visual odometry on the Mars exploration rovers: Field reports.* Journal of Field Robotics, vol. 24, no. 3, pages 169–186, 2007.

[Matei 99] Bogdan Matei & Peter Meer. *Optimal Rigid Motion Estimation and Performance Evaluation with Bootstrap.* In Proceedings of the Conference on Computer Vision and Pattern Recognition , Fort Collins Co (CVPR), pages 339–345, 1999.

[Mathies 87] Larry Mathies & Steven A. Shafer. *Error Modeling in Stereo Navigation.* IEEE Journal of Robotics and Automation, vol. RA-3, pages 239–248, June 1987.

[MatLab 07] MatLab. *http://www.vision.caltech.edu/bouguetj/calib_doc/*, 2007.

[Meng 06] Y. Meng. *Improved positioning of land vehicle in ITS using Digital map and other accesory information*. PhD thesis, Deparment of Land Surveying and Goeinformatics, Hong Kong Polytechnic University, 2006.

[Mobileye 07] Mobileye. *http://www.mobileye-vision.com*, 2007.

[Montiel 06] J.M.M. Montiel, J. Civera & A.J. Davison. *Unified inverse depth parametrization for monocular SLAM*. Proceedings of Robotics: Science and systems, August 2006.

[Moravec 80] H.P. Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, Standford University, Stanford, CA, September 1980.

[Mouragnon 06] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser & P. Sayd. *Real time localization and 3d reconstruction*. In Proceedings of the Conference on Computer Vision and Pattern Recognition , (CVPR), pages 363–370, 2006.

[Murray 98] D. Murray & J. Little. *Using real-time stereo vision for mobile robot navigation*. In Proceedings of the IEEE Workshop on Perception for Mobile Agents, 1998.

[Neira 01] José Neira & Juan D. Tardós. *Data association in Stochastic Mapping using the Joint Compatibility Test*. In IEEE Transactions on Robotics and Automation, volume 17, pages 890–897, 2001.

[Nistér 03a] D. Nistér. *An efficient solution to the five-point relative pose problem*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pages 195–202, 2003.

[Nistér 03b] D. Nistér. *Preemptive RANSAC for live Structure and Motion Estimation*. IEEE International Conference on Computer Vision, pages 199–206, 2003.

[Nistér 04a] D. Nistér. *A minimal solution to the Generalised 3-point pose problem*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pages 560–567, 2004.

[Nistér 04b] D. Nistér, O. Narodistsky & J. Beren. *Visual Odometry*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2004.

[Nistér 06] D. Nistér, O. Narodistsky & J. Bergen. *Visual Odometry for ground vehicle applications*. Journal of Field Robotics, vol. 23, pages 3–20, 2006.

[Ochieng 04] W.Y. Ochieng, M. Quddus & R.B. Noland. *Map-matching in complex urban road networks*. Brazilian Journal of Cartography, vol. 55, pages 1–18, 2004.

[Oliensis 99] J. Oliensis & Y. Genc. *New Algorithms for Two-Frame Structure form Motion*. Proceedings of International Conference on Computer Vision, pages 737–744, 1999.

[OpenStreetMap 10] OpenStreetMap. *http://wiki.openstreetmap.org*, 2010.

[Parra 10] Ignacio Parra, Miguel Ángel Sotelo, David F Llorca & Carlos Fernández. *Visual Odometry for accurate vehicle localization- an assistant for GPS based navigation*. Intelligent Transportation Systems World Conference, vol. 28, page (accepted), October 2010.

[Paz 08a] Lina M. Paz, Pedro Piniés, Juan D. Tardós & José Neira. *Large Scale 6DOF SLAM with Stereo-in-Hand*. IEEE Transactions on Robotics, vol. 24, no. 5, October 2008.

[Paz 08b] Lina M. Paz, Juan D. Tardós & José Neira. *Divide and Conquer: EKF SLAM in O(n)*. IEEE Transactions on Robotics, vol. 24, no. 5, October 2008.

[Pyo 01] J. Pyo, D. Shin & T. Sung. *Development of a map-matching method using the multiple hypothesis technique*. IEEE Proceedings on Intelligent Transportation Systems, pages 23–27, 2001.

[Quddus 07] Mohammed A. Quddus, Washintgton Y. Ochieng & Robert B. Noland. *Current map-matching algorithms for transport applications: State-of-the-art and future research directions*. Transportation Research Part C, vol. 15, pages 312–328, 2007.

[Rodríguez 88] Jeffrey J. Rodríguez & J.K. Aggarwal. *Quantization Error in Stereo Imaging*. Proceedings of Computer Vision and Patter Recognition, pages 153–158, 1988.

[Salesman 10] Travelling Salesman. *http://wiki.openstreetmap.org/wiki/Travelling_salesman*, 2010.

[Scaramuzza 08] Davide Scaramuzza & Roland Siegwart. *Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground vehicles*. IEEE Transactions on Robotics, vol. 24, no. 5, October 2008.

[Schleicher 09] David Schleicher, Luis M. Bergasa, Manuel Oca na, Rafael Barea & Elena López. *Real-time Hierarchical GPS aided visual slam on urban evironments*. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA09), pages 4381–4386, 2009.

[Schmid 00] C. Schmid, R. Mohr & C. Bauckhage. *Evaluation of Interest Point Detectors*. In International Journal of Computer Vision. Vol. 37, No. 2, pp. 151-172, 2000.

[Se 01] S. Se, D. Lowe & J. Little. *Vision-based mobile robot localization and mapping using scale-invariant features*. In Proceedings of the IEEE ICRA. pages 2051-2058, 2001.

[Shashua 04] Amnon Shashua, Yoram Gdalyahu & Gaby Hayun. *Pedestrian detection for driving assistance Systems: Single-frame Classification and System level Performance*. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Parma, Italy, 2004.

[Slama 80] C.C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry, 1980.

[Solina 85] Franc Solina. *Errors in Stereo due to Quantization.* Tech. Rep. MS-CIS-85-34, Dept. of Computer and Information Science, Univ. of Pennsylvania, September 1985.

[Stein 97] Gideon P. Stein & Amnon Shashua. *Model based brightness constraints: On direct estimation of structure from motion.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Puerto Rico, June 1997.

[Stein 00] Gideon P. Stein, Ofer Mano & Amnon Shashua. *A robust method for computing vehicle ego-motion.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2000.

[Strelow 01] D. Strelow, J. Mishler, S. Singh & H. Herman. *Extending shape-from-motion to noncentral omnidirectional cameras.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Wailea, Hawai, October 2001.

[Sunderhauf 05] Niko Sunderhauf, Kurt Konolige, Simon Lacroix & Peter Protze. *Visuao odometry using sparse bundle adjustment on an autonomous outdoor vehicle.* In Tagungsband Autonome Mobile Systeme, 2005.

[Torge 91] Wolfgang Torge. Geodesy. Walter de Gruyter, Berlin, 1991.

[Triggs 98] B. Triggs. *Autocalibration from planar scenes.* ECCV, vol. 1046, pages 89–105, 1998.

[Trucco 98] E. Trucco & A. Verri. Introductory techniques for 3-d computer vision. Prentice Hall PTR, 1998.

[Tsai 81] R. Tsai & T. Huang. *Estimating three-dimensional motion parameters of a rigid planar patch.* IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 29, no. 6, pages 1147–1152, 1981.

[White 00] C.E. White, D. Bernstein & A.L. Kornhauser. *Some map-matching algorithms for personal navigation assistants.* Transportation Research Part C, vol. 8, pages 91–118, 2000.

[wikipedia 10a] wikipedia. *http://en.wikipedia.org/wiki/Epipolar_geometry*, 2010.

[wikipedia 10b] wikipedia. *http://en.wikipedia.org/wiki/Openstreetmap*, 2010.

[Wu 03] Chen Wu, Yu meng, Li Zhi-lin, Cheng Yong-qi & J. Chao. *Tight integration of digital map and in-vehicle positioning unit for car navigation in urban areas.* Wuhan University of Natural Sciences, vol. 8, pages 551–556, 2003.

[Wunderlich 82] W. Wunderlich. *Rechnerische Rekonstruktion eines ebenen Objekts aus zwei Photographien.* Mitteilungen der geodaetischen Institute, vol. 40, pages 365–377, 1982.

[Xu 96] G. Xu & Z. Zhang. Epipolar geometry in stereo, motion and object recognition: A unified approach. Kluwer Academic Publisher, Dordrecht, Boston, London, 1st edition, 1996.

[Yang 03] D. Yang, B. Cai & Y. Yuan. *An improved map-matching algorithm used in vehicle navigation systems.* IEEE Proceedings on Intelligent Transportation Systems, vol. 2, pages 1246–1250, 2003.

[Zhao 97] Y. Zhao. *Vehicle localization and Navigation System.* Artech House, Inc. MA, 1997.