Simultaneous localization and mapping in 3D environments with stereovision

Il Kyun Jung

N° d'ordre : ??? Année 2004

THESE

préparée

au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

en vue de l'obtention

du Doctorat de l'Institut National Polytechnique de Toulouse

Spécialité : Systèmes Informatiques

par

Il-Kyun JUNG

SIMULTANEOUS LOCALIZATION AND MAPPING IN 3D ENVIRONMENTS WITH STEREOVISION

Composition du jury :

MM.			Président
			Examinateurs
	Simon	LACROIX	Directeurs de thèse
	Michel	DEVY	
	Jean-Pierre	COCQUEREZ	Rapporteurs
	Andrew	DAVISON	
	Salah	SUKKARIEH	

Abstract

In robotics, the problems of environment modelling and localisation are intimately tied together, and must therefore be solved in a unified manner. Over the past 15 years, there has been several contributions to this simultaneous localisation and mapping problem (often referred to as "SLAM"), in which landmarks in the environment are both mapped by the robot and used to estimate its position. The achievement of SLAM encompasses various processes: landmark detection and matching from subsequent positions, errors identification of the landmark position estimate from a given position, and finally the use of an estimation theory is required to estimate both the robot and landmarks positions.

In this thesis, we tackle the SLAM problem for robots evolving in 3D in large environments, without any prior knowledge on the environment, and using stereovision. A full implementation of the various processes has been conceived, developed, and experimented in various contexts.

The first part of the thesis deals with the data association problem: it introduces a matching algorithm for invariant image point features, which is robust to image noise and viewpoint changes. To establish points matches, the algorithm mixes the image signal information at a given point with geometric constraints between the considered point and neighbouring points. Experimental results show that the algorithm is versatile, in the sense that it can establish reliable matches in almost any kind of scene.

The second part of the thesis is devoted to the development of a SLAM approach using an extended Kalman filter. Landmarks are the point features detected in the image, their 3D coordinate being computed by stereovision. A full SLAM implementation is depicted, in which the matching algorithm is used for both the motion estimation of the robot (Kalman filter prediction stage) and the observation of the landmarks. The determination of the various errors involved in the whole

process (motion estimation errors, stereovision errors, landmark observation errors) is depicted in details. A way to select the landmarks to be memorised among the numerous point features is presented.

The last part of the thesis presents and analyses results obtained in various contexts: with several hundreds meter long trajectories achieved by a low altitude flying blimp, with an outdoor rover and with a robot evolving indoor. When the trajectory “closes a loop”, a fast backward correction of the robot poses based on the local motion estimates and the landmark map is applied, in order to be able to construct spatially consistent large digital elevation maps of the environment.

Résumé

En robotique, les problèmes de la modélisation d'environnement et de la localisation sont intimement liés, et doivent par conséquent être résolus de manière unifiée. De nombreuses contributions au problème de la cartographie et de la localisation simultanée (souvent noté "SLAM", pour "Simultaneous Localization And Mapping") ont été proposées depuis une quinzaine d'années, dans lesquelles des amers de l'environnement sont à la fois cartographiés et exploités pour localiser le robot. La solution à ce problème passe par le développement de différents processus : détection des amers, appariement entre les amers détectés à partir de positions différentes, identification des différentes erreurs de mesure, et enfin utilisation d'une théorie de l'estimation pour déterminer la position des amers et du robot.

Dans cette thèse, nous abordons le problème SLAM pour des robots évoluant en 3D dans de grands environnements, en utilisant la stéréovision. Une implémentation complète des différentes fonctionnalités nécessaires a été conçue, développée et expérimentée dans différents contextes.

La première partie de la thèse traite du problème d'association des données : elle présente un algorithme de mise en correspondance de points d'intérêt détectés dans les images, qui est robuste par rapport au bruit et aux changements de point de vue. L'algorithme mêle des informations relatives au signal et à la géométrie qui lie les points entre eux. Des résultats montrent que l'algorithme est applicable dans un nombre extrêmement varié de scènes et pour différentes conditions d'acquisition.

La deuxième partie de la thèse est dédiée au développement d'une approche du problème SLAM basée sur le filtrage de Kalman. Les amers sont les points d'intérêt détectés dans les images, dont les coordonnées 3D sont fournies par la stéréovision. Une implémentation complète est présentée, dans laquelle l'algori-

thme de mise en correspondance des points est exploité pour estimer les déplacements du robot (étape de prédiction du filtrage de Kalman) et pour re-observer les amers précédemment cartographiés. La détermination des erreurs intervenant dans les différents processus (estimation du mouvement, stéréovision, observation des amers) est décrite en détail, et un moyen de sélectionner activement les amers à cartographier est présenté.

La dernière partie de la thèse présente et analyse des résultats obtenus dans différents contextes : sur des trajectoires de plusieurs centaines de mètres effectuées par un ballon dirigeable évoluant à faible altitude, avec un robot évoluant en environnement naturel non structuré, et avec un robot évoluant en environnements intérieurs. Lorsque la trajectoire “ferme une boucle”, une méthode rapide de correction de l’estimée des différentes positions par lesquelles le robot est passé permet de reconstruire un modèle numérique du terrain.

Contents

Table of notations

Interest point matching algorithm (chapter 2)

$\mathcal{T}_{s,\theta}$	estimated approximate transformation with scale and rotation change (s, θ) between two local groups of interest points
$G(., \sigma)$	Gaussian Kernel of standard deviation σ
$G_u(., \sigma), G_v(., \sigma)$	first order derivatives of $G(., \sigma)$ along the u and v directions
$M(\mathbf{x}, \sigma, \tilde{\sigma}, s)$	auto-correlation matrix of scale adaptive Harris detector
$\mathcal{S}_{p1}(.,.)$, $\mathcal{S}_{p2}(.,.)$	similarity measures between two interest points
$\mathcal{G}$	local group of interest points
$\mathcal{S}_v(.,.)$	vector similarity between two vectors in local groups
$\mathcal{R}_g(.)$	group repeatability
$\mathcal{S}_g(.)$	group similarity
$S_{.,.}$	matching strength between two local groups
$\gamma_d(.,.)$	discriminancy of a group match
$\gamma_{lc}(.,.)$	local consistency of a group match
γ_{lc}	completeness measure of the propagation
Ω_L, Ω_G	local and global regions of an image
W	local window for search or correlation
$\mathscr{E}(.,.)$	error distribution over a search area
$\mathscr{R}(.,.)$	response distribution

Simultaneous Localization And Mapping (chapter 3)

$\mathbf{f}(.,.)$	nonlinear system model
$\mathbf{h}(.,.)$	nonlinear observation model
$\mathbf{\Theta}$	map of landmarks
$\mathbf{z}$	set of all landmarks observations up to current time
$\mathbf{u}$	set of all control inputs up to current time result (visual motion estimation between two consecutive frames)
$\mathbf{x}_p$	pose of the stereovision bench (6 parameters)
$\mathbf{m}_i$	position of the ith landmark
$\mathbf{R}_m$	error covariance of a new landmark
$\mathbf{R}_u$	error covariance of the visual motion estimation result
$\mathcal{V}_i$	visibility of a landmark
$\mathbf{A}$	intrinsic matrix of pre-calibrated stereo bench
$\mathbf{z}_i$	observation of the ith landmark
$\hat{\mathbf{z}}_i$	expected observation of the i-th landmark
$\mathbf{z}_i - \hat{\mathbf{z}}_i$	observation innovation
$\mathbf{g}(.,.)$	landmark initialization model
$\mathbf{R}_i$	error covariance of ith landmark observation

Chapter 1

Introduction

1.1 Context: autonomous navigation

Autonomous navigation is a fundamental ability for mobile robots, be it for ground rovers, indoor robots, flying or underwater robots. Indeed, all the missions to be achieved by such robots (*e.g.* long range traverses, exploration, surveillance...) require the ability to move autonomously from one place to an other.

In short, autonomous navigation is the ability to reach a distant goal, without any human operator intervention. This ability calls for the integration of a wide variety of processes, from low level actuator control, to higher level strategic decision making, via environment mapping and path planning. Among these various functionalities, self-localization is definitely one of the most important, as it is required at various levels in the whole system, from fine trajectory control to mission supervision:

- The missions to be achieved by the robot are often expressed in localization terms, explicitly (*e.g.* "reach that position", "explore this area"...) or more implicitly (such as in "return to the initial position" when it is out of sight). The knowledge of the robot position (and of the various positions reached during its traverse) is therefore required at the mission control level.

- Autonomous navigation calls for the building of *global maps* of the environment, to find trajectories or paths and to enable mission supervision. The *spatial consistency* of such maps is required to allow an efficient and ro-

bust behavior of the robot: it is the knowledge of the robot position that guarantees this consistency.

- Finally, the proper execution of the geometric trajectories provided by the path planners calls for the precise knowledge of the robot motions. Note that in this latter case, some visual guidance techniques can help to drive the robot without the explicit knowledge of its position. However, the domain of operation of such approaches is essentially restricted to very local trajectories, in a manner similar to object catching tasks with an arm.

Dead reckoning techniques, that integrate over time the data provided by motion estimation sensors, such as wheel encoders for rovers or inertial sensors, provide an estimation of the robot position as it moves. However, these techniques are intrinsically prone to generate position estimates with unbounded error growth, as the position estimates are deduced from the composition of elementary noisy motion estimates. Visual motion estimation techniques that use monocular sequences [Chaudhuri 96, Heeger 92, Vidal 01, Garcia 01] and stereovision sequences [Weng 92, Zhang 92, Mandelbaum 99, Mallet 00, Olson 01] allow to get a quite precise motion estimate between successive data acquisitions, but they are akin to dead reckoning, and their errors also grow with time.

The only solution to guarantee bounded errors on the position estimates is to rely on stable environment features. If a spatially consistent map of the environment whithin which the robot operates is available a priori, map-based position localization can be applied (a number of successful approaches have been reported, *e.g.* in [Kwok 03, Borges 02, Dellaert 99]). Localisation based on radioed beacons, such as GPS, fall in this category of approaches: the beacons, whose position is known, play the role of the a priori map.

On the other hand, it is clear that if the robot position is always perfectly known, the building of an environment map with the perceived data is quite trivial. If the data provide geometric informations properly segmented for the chosen representation to build, the only difficulty is to deal with their uncertainties.

So an accurate map is a prerequisite for absolute localization, and in contrast, an accurate map can be obtained with a correct estimate of the robot motions. But in the absence of an a priori map of the environment, the robot is facing a kind of “chicken and egg problem”: it makes relative observations on the environment

that are corrupted by errors, while the dead reckoning estimates of the positions from which it perceives these data are corrupted with errors that grow with the travelled distance. These errors in the robot's pose have an influence on the estimate of the observed environment feature locations, that are therefore corrupted by both the observation errors and the robot position errors. Similarly, the use of the observations of previously perceived environment features to locate the robot provide position estimates that inherits from both errors: the errors in both the robot's pose and the map feature are correlated.

It has early been understood in the robotic community that the problems of mapping the environment and estimating the robot location are intimately tied together, and that they must therefore be taken into account concurrently and solved in a unified manner [Chatila 85, Smith 87]. The approaches that deal with this problem are now commonly denoted as "Simultaneous Localization and Mapping" (SLAM): they consist in concurrently building up a map of environment features (or landmarks) and localizing the position of the robot using the landmarks, without any prior knowledge on the environment.

1.2 Simultaneous Localization And Mapping

Given a set of ego-motion estimates and observations of environment landmarks, the goal of SLAM is to accurately estimate the robot and landmark positions, using the relative observations on its motions and on environment landmarks. Since the ego-motion estimates and the observations are corrupted by errors, if appropriate error models are available, they can be used as probabilistic constraints. Even though the constraints are initially loose, as landmarks are re-observed, they become gradually tighter, *i.e.* the precision on the robot and landmark positions increases. Figure 1.1 illustrates the various steps involved during a SLAM process.

The implementation of a full SLAM approach encompasses the following functionalities:

- Landmark determination: it consists in detecting in the perceived data the features of the environment that can be used as landmarks,

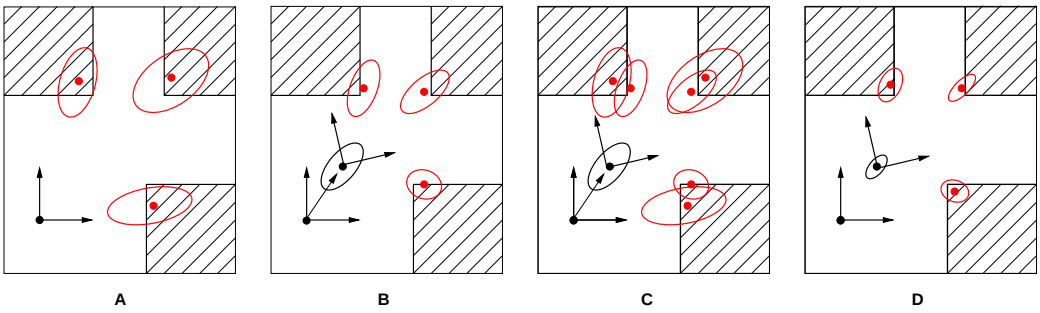


Figure 1.1: *Illustration of the SLAM problem with a robot evolving on a plane: (a), environment features (or landmarks - convex corners here) are perceived and located with an uncertainty (observation). In (b), the robot moves, and knows its position via an external mean and with an associated uncertainty (ego-motion estimation: prediction). The landmarks are re-observed. Thanks to matches of the features perceived from the two positions (c: data association), the uncertainties in both the robot and the features can be reduced (d: fusion). The process goes on, and incrementally estimates the robot and landmarks position errors.*

- Ego-motion estimation: between two observations of the environment landmarks, an estimate of the robot motion has to be provided,
- Data association: it consists in associating the landmarks perceived from different robot positions,
- Estimation: it consists in integrating the various measurements to estimate the positions of the landmarks and the robot in a global reference frame. For that purpose, an estimate on the errors in the landmarks and robot position measurements is required (errors identification),
- Map management: finally, for algorithmic complexity issues and to ensure the best positions estimates as possible, an active way of selecting and managing the various landmarks is necessary.

Landmark determination

Landmarks are features in the environment that are salient, easily observable and whose relative position to the robot can be estimated. The actual definition of a landmark depends on the kind of environment within which the robot evolves and on the sensors the robot is equipped with. The detection of landmarks is a *perception problem*: is consists in finding in the data the features, estimating their

relative position to the robot, and representing them into a specific data structure. For robot evolving on perfect planes (*e.g.* in indoor man made environments), the landmark positions are expressed in 2D coordinates, and most of the times detected on the basis of range data. If the environment exhibits some geometric structures, as straight lines for instance, these structures are ideal landmarks. Note that in the worst cases (as in a perfectly flat desert for instance), no landmarks are available in the environment.

Ego-motion estimation

The estimate of the robot displacement between two data acquisitions can be provided either by some sensors, or thanks to a dynamic model of the robot evolution. The sensors that can provide this information are ususally the dead reckoning sensors the robot is equipped with. In the absence of such sensors and if the dynamic model of the robot is known, the motion control inputs can help to predict the displacement, and in the worst cases, a simple assumption (such as constant velocity) can be used to estimate the motion.

Data association

The perception of landmarks is useful to compute the robot position estimates only if they are perceived from different positions: they must therefore be properly associated (or matched), otherwise the robot position can become totally inconsistant. A first approach to this problem is to reason on the position estimates of the landmarks and the robot to associate the landmarks: the positions of the previously perceived landmarks with respect to the current robot position are predicted on the basis the current robot position estimate, and compared to the measured positions of the landmarks. However, when the errors on some of these positions is big, the associations can become ambiguous: this is the case when the robot re-perceives landmarks after having travelled along a long loop trajectory for instance. This is all the more difficult when the robot is evolving in 3D: the errors in the prediction of the 6 parameters of the robot position (3 angles and 3 translations), rapidly have a drastic influence on the predicted position of the landmarks. An other way to solve the data association process is to *recognize* the landmarks, independently from their position estimate. This requires that the landmarks be

easily distinguishable one from the other, and that they are represented with a sufficient set of attributes that allows to recognize them.

Estimation

Global optimization technique like bundle adjustment used in "structure from motion" in computer vision could be a solution to the SLAM problem [Deans 01], but they require a batch processing, *i.e.* with all observations and ego-motion estimates up to current time. In the case of robot navigation in large environments, such a batch estimation is not adequate, as is does not continuously provide an estimate of the robot and landmark positions. Furthermore, these techniques do not provide an estimate of the uncertainty of the landmark locations, which is useful for robot exploration and path planning in unknown environments.

The online approaches to the SLAM problem consists in estimating a posterior probability distribution over all the robot and landmark positions, with all the available ego-motion estimates and landmark observations up to the current time. The distribution can be written as

$$p(\mathbf{x}_t, \mathbf{\Theta} | \mathbf{z}, \mathbf{u}) \tag{1.1}$$

where $\mathbf{x}_t$ is the current robot's state and $\mathbf{\Theta}$ is the map of landmarks, conditioned on all the landmark observations $\mathbf{z}$ and control inputs $\mathbf{u}$, the control inputs being the results of the ego-motion estimations.

The posterior estimation is more suitable to solve SLAM problem with noisy measurements than a maximum likelihood approach, because it provides the most probable robot and landmarks positions considering and a quantitative estimation of the uncertainty in both the robot and landmark psitions. To compute the posterior, several statistical estimation techniques could be engaged: *e.g.* Kalman filter, information filter and particle filter.

Extended Kalman filter Among the different approaches to solve the SLAM problem, the Kalman filter based approach, or variants like the information filter, is undoubtly the most popular. The state of the Kalman filter is the posterior, approximated by a multivariate Gausian, represented by the vector of expected values (mean) and the corresponding covariances matrix, of which the

off-diagonal elements represent the correlation between landmark locations and between the robot position and landmark locations. In practice, the system and the observation models being non-linear, the Extended Kalman Filter (EKF) is used: a linear Kalman filter formulation is obtained by linearizing the non-linear system and observation models. Kalman Filter based SLAM is theoretically well grounded, and it has been proved that its application to the SLAM problem converges [Dissanayake 01].

Although EKF-based solutions has been mostly used in the past decade, they suffer from some well-known problems: the assumption that the errors follow a Gaussian probability distribution, the sensitivity to incorrect data association, and the cubic time complexity of the filter update.

The uncertainty of the posterior is represented as a covariance matrix that maintains all correlations of the variables in the state vector, and whose size is the square of the state vector. Matrix multiplications are involved to incorporate observations, which leads to a complexity which is cubic in the number of the state variables. In large environments, as new landmarks are incorporated within the state, the process becomes untracktable.

The classical technique to data association in EKF based SLAM is to assign each observation to the landmark that is the most likely to have generated it, the likelyness being determined on the basis of the estimated positions and variances of the landmark. As mentionned above, in case the uncertainties on the robot and landmark locations are high, wrong association between the observations and landmarks might occur, which makes the filter diverge. This problem can be overcomed by the simultaneous consideration of multiple observations, at the cost of higher computations [Neira 01], or by the unambiguous recognition of the landmarks.

Other estimation approaches to the SLAM problem have been proposed, mainly to overcome the assumption that the various error probability distributions are Gaussian, which is required by the Kalman filter. Set membership approaches just need the knowledge of bounds on the errors [Kieffer 00, Marco 01], but they are practically difficult to implement when the number of position parameters exceeds 3, and are somehow sub-optimal. Expectation minimization algorithms (EM) have also been successfully adapted to the SLAM problem [Thrun 98], and an approach that address incremental SLAM in this context can be found in [Thrun 00].

Error Identification The ego-motion and landmarks relative observations are always corrupted by errors. In order to use them as probabilistic constraints, it is necessary to establish appropriate error models and to estimate their quantity, to avoid tedious "filter tuning" steps and to provide a robust solution. In case the observations are directly provided by sensor readings, rigorous studies (*e.g.* statistical analysis) on bias and noise distribution are required. When the observations result from various algorithmic processes, analytical techniques that propagate the errors from the raw sensor readings to the estimation through the various algorithmic stages must be applied.

Map management

One of the main difficulty of the EKF based SLAM solution is the cubic computation complexity caused by keeping the correlations between all the state variables within the sate covariance matrix. To apply EKF based SLAM in large environments, many researchers have paid attention to the fact that an observation of a landmark have a strong effect on adjacent landmarks, and has in contrast relatively weak effect on distant landmarks. Hence, a global map can be splited into local maps, under the assumption that the robot lingers a time long enough in a sub-region to stabilize the sub-region. Delayed incorporation of submaps into the global map is performed after the robot moves out of the submap [Knight 01, Guivant 01]. The approaches based on spliting the global map into submaps lead a sparse description of the correlation between the map elements of overall map provided by SLAM (*e.g.* relative covariance between submaps). Thorough considerations on the relationship between the sparse network of submaps and more efficient computation to incorporate observations have been reported [Bosse 02, Chong 99, Leonard 01].

Recently, Sparse Extended Information Filter (SEIF), which represents the SLAM posterior in the natural parameters of multivariate Gaussians has been proposed [Thrun 02, Thrun 03]. The parameters consist of an information matrix which is the reciprocal of the covariance matrix of the standard kalman filter, and an information vector (from which the expected value of the state can be retrieved with the information matrix). The important property of SEIF based approach is that the information matrix is dominated by a small number of elements, and the

update step therefore can be performed by manipulating only this small number of elements. Furthermore, the update states are additive in the SEIF formulation (to extract the overall landmark map from SEIF, a matrix inversion is still required).

An alternative approach to estimate the posterior has been presented in [Montemerlo 03]. The approach called *FastSLAM* scales logarithmically with the number of landmarks in the overall map and samples over potential robot paths. With motions in 3D, the large number of potential robot's paths still makes the computation expensive.

Application contexts

The SLAM problem has been mostly studied in the case of robots moving on planes, *i.e.* whose position is totally determined by 3 parameters (an historical presentation of the main contributions can be found in the introduction of [Dissanayake 01]). In terms of sensor modality, solutions to the SLAM problem has mainly been experimented with range sensors (sonar sensors [Leonard 91, Wijk 00] or laser range finders in indoor environments [Moutarlier 91, Thrun 00], and more recently millimeter wave radars in outdoor environments [Guivant 01, Dissanayake 01]).

To our knowledge, there are much fewer contributions to the SLAM problem based on vision. In [Se 01], an approach that uses stereovision and visual scale-invariant features transforms (SITF) for a robot evolving on a plane environment is presented, the data association problem being solved a Hough transform hashing. Recently, a real time SLAM approach using a single camera combining particle filter and EKF has been presented [Davison 03]. In the approach, the 3D coordinates of visual landmarks are obtained using a particle filter and once stabilized, they are incorporated into the state vector of an EKF. After initializing the SLAM process with known landmarks, the SLAM proceeds incorporating new landmarks, the ego-motion being predicted using a constant velocity model.

1.3 Our contributions

This thesis presents a solution to the Simultaneous Localization And Mapping problem with stereovision in unknown 3D environments, in which the robot position is *totally 3D, i.e.* determined by 3 translation and 3 orientation parameters,

and that *exclusively* uses stereovision. Vision has the great advantage to allow both a very precise determination of the orientation parameters and the association of stable environment features. Moreover, using a stereovision bench, range estimates of the features are directly available, although much less precise than those provided by a laser range finder. We will however see that thanks to the Kalman filter, it is possible to achieve extremely precise localization of the stereovision bench and build up a spatially coherent map of environment, without the aid of any other positioning sensor.

In our approach, landmarks are *interest points*, *i.e.* visual features that can be matched when perceived from various positions, and whose 3D coordinates are provided by stereovision. The advantage of using such discriminant visual features as landmarks is twofold. First, it does not require any assumption on the nature of the environment: the presence of geometric features or distinctive elements such as trees, rocks or pebbles is not required. Second, interest points are discriminative enough to be matched in images taken from different points of view, regardless of their corresponding 3D position: this allows robust data association. For that purpose, we introduce an interest point matching algorithm that allows both the motion estimation between consecutive stereovision frames and the matching of the current observations to memorized landmarks.

We use an extended Kalman filter (EKF) as a recursive filter: the state vector of the EKF is the concatenation of the stereo bench position (6 parameters) and the landmark's positions (3 parameters for each landmark). The visual motion estimation between consecutive stereovision frames is used to predict the filter state, and is fused with the observations provided by landmark matchings.

The principal contributions of this thesis arise from our robust interest point matching algorithm, which provides an important basis to solve the problems that make the achievement of robust SLAM in 3D environments difficult: stable visual landmark determination, precise ego-motion estimation and robust data association. Thanks to rigorous studies on the errors on the estimation of the 3D landmarks position, on the visual ego-motion estimation, and on the observation, we show shown that a very accurate localization and mapping can be concurrently achieved in various 3D environments.

Landmark detection In the absence of any prior knowledge on the nature of the objects that constitude the environment, salient visual features can be used as stable landmarks, even in an environment where there is no discriminant objects (*e.g.* on a flat lawn). Figure 1.2 shows the results the interest points detected in various scenes: such points have good stability properties with respect to view point changes, thus enabling a robust data association. The use of interest points as landmarks makes the SLAM applicable to almost any kind of 3D environments, except totally non-textured regions (*i.e.* the variations of intensity are too weak to detect any discriminant pixel).

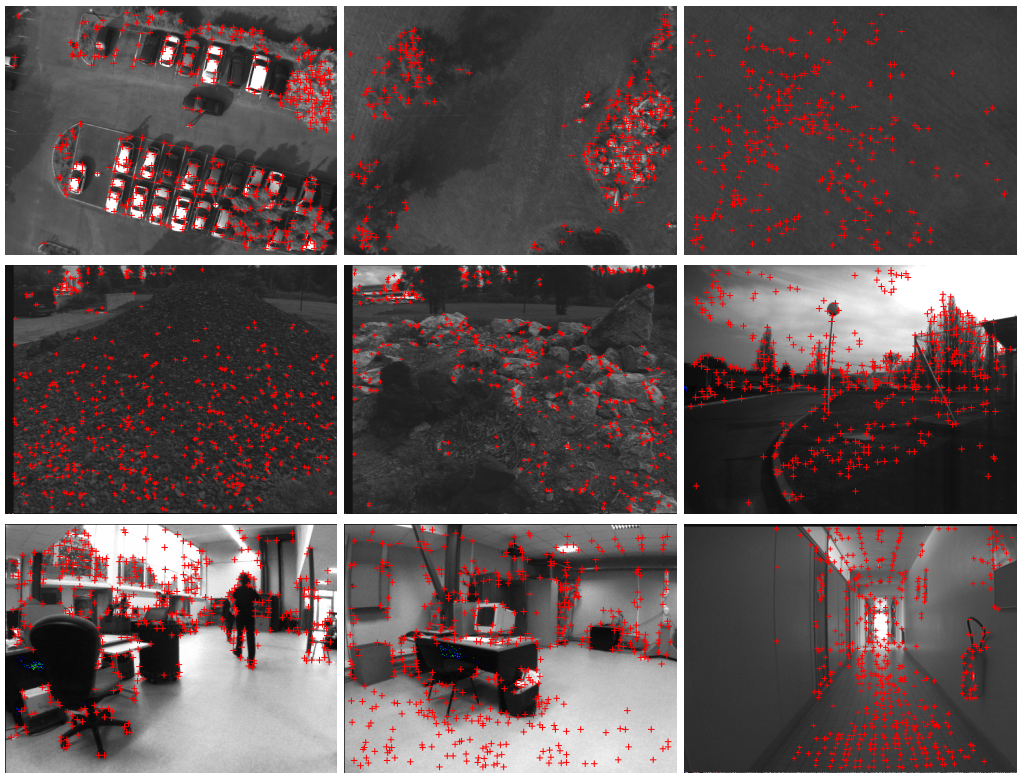


Figure 1.2: *Interest points (red crosses) detected in various environmentsq. Top row: low altitude aerial images; middle row: outdoor images taken by a ground rover; bottom row: indoor scenes. Even in lack of texture, interest points can be detected (see top right image, aerial view of a lawn). For all these images, a detection threshold has been set so that the number of detected points is equal to a fixed value.*

Robust data association We have developped an interest point matching algorithm that enables the matching of the visual landmarks on the images, independently from the uncertainty in both the robot and landmark positions. Outliers that eventually occur can be eliminated by some classical robust techniques like RANSAC (Random Sample Consensus), which roots out most of the wrong matches. A further outlier removal technique is applied to guaranty the rejection of all the remaining wrong matches: this provides reliable data associations, that can be incorporated as observations in the Kalman Filter.

Relative observations in non-planar world To achieve the SLAM, two relative observations have to be achieved: the ego-motion of the robot between consecutive data perceptions, and the estimate of the range to the perceived landmarks. In our approach, stereovision provides the range estimates of the visual landmarks detected in the images, and a visual motion estimation technique that exploits the matched 3D points between two consecutive frames provides a precise ego-motion estimate. We will see that one can achieve SLAM in the absence of any additional sensor.

Error identification Three kinds of error must be identified in a SLAM approach: errors in the landmark relative measurements, errors in the already mapped landmark observations, and errors in the ego-motion estimate. In our approach, these errors result from the processing of noisy data: we will identify them rigorously with statistical and analytic methods, so that the filter setup will not require any “tuning” step.

Map management Although we do not use any recent developpement to deal with the algorithmic complexity of the Kalman filter, we actively select the landmarks to be incorporated in the filter state among all the detected interest points, in order to limit their number, while maximizing their “representativity” of the environment. This is made possible by an active control of the number of interest points detected in each perceived image, and an analysis of their precision and position with respect to already mapped landmarks.

1.3.1 Thesis outline

The remainder of this thesis is organized as follows:

- Chapter 2 depicts in details the detection of interest points, and the proposed algorithm to match them in different images. Various matching results illustrates the developments.
- Chapter 3 presents our approach to the SLAM problem, using exclusively stereovision. The various involved algorithms are presented: stereovision, visual motion estimation, data association using interest point matching algorithm and landmark selection. The Kalman Filter setup is then described, and the determination of the various errors involved in the whole process is presented.
- Chapter 4 presents and discusses results obtained on three very different contexts: with low altitude aerial images gathered by a blimp, with images gathered by a rover in a natural unstructured environment, and with a robot evolving indoor. It also introduces a way to build a spatially consistent digital elevation map, thanks to a determination of all the robots positions once its trajectory closes a loop.

Chapter 2

Interest Points Matching Algorithm

2.1 Introduction

Landmark identification and matching is of course essential in the SLAM process. To implement a vision-based SLAM, one must select features that can be successfully matched under the following conditions:

- Variations on the grey levels of the images, due to noise and illumination conditions,
- Viewpoint changes, that induce variations on the image resolution, on the aspect of the perceived objects (including occlusions), and affect the surface of the overlapping area on the images,
- And finally variations in the perceived scene itself, in which objects may appear or disappear between the image acquisitions.

Features can be directly the image signal, or edges, contours, lines, regions detected on the image, up to higher level semantic informations. Aiming at designing a versatile matching algorithm that works in any kind of context precludes the use of structural features such as line segments, that seldom occur in natural environment scenes for instance. Moreover, using lower level features avoids the use of fragile segmentation algorithms: a lot of contributions have therefore focused on the matching problem using directly the image signal as the feature space.

2.1.1 Related work

Matching methods based on local grey value similarity scores are suitable when the transformation between the two images is close to the identity, *e.g.* in stereovision [Faugeras 93] or tracking. Various similarity measure have been used or proposed: the sum of squared difference (SSD) between pixels belonging to a window surrounding the pixel to match is commonly used [Anandan 89, Shi 94, Papanikolopoulos 95]. Common similarity scores have been compared in [Martin 95] and an original one, more robust to signal noise, has been proposed in [Zabih 94]. All these methods are very sensitive to changes in image scale, rotation and viewpoint, because they attempt to match a rigid window between the images: a signal matching approach with adaptive windows has been presented in [Kanade 94] to cope with the aspect changes due to the projective geometry. Still, in order to generate reliable matches, these approaches require to focus the match search, *e.g.* with the epipolar constraint, which is known in calibrated stereovision, or needs to be recovered in the most general case [Zhang 95].

To establish matches when several unknown changes occur in the image, one must consider features that are as much invariant as possible with respect to any image transformation. Texture descriptors extracted with Gabor wavelets have these properties. Gabor wavelets measure the signal energy at a given frequency and direction: texture features are extracted from the energy distribution computed in the 2D frequency/direction space (experiments show that human and mammalian vision perform such a spatial-frequency analysis [Daugman 85]). Such features have been used to track facial feature points [McKenna 97], or to search images in data bases [Wolf 00]. In these approaches, a high resolution of spatial frequency and angular orientation is required, which is computationally expensive.

Point features, often denoted as “interest points”, are salient in images, have good invariant properties, and can be extracted with much less computation: they are therefore excellent landmark candidates for a vision-based SLAM problem. The Harris corner detector locates points in the image where the signal changes in two directions, by analysing the auto-correlation function of the image, estimated with first order derivatives [Harris 88]. Further study in this direction has been done in [Noble 88], and methods which uses the auto-correlation of Har-

ris detector to detect interest points were presented in [Förstner 87, Förstner 94]. In [Wang 95], point features are found using the curvature of the image surface, this curvature being defined by the combination of the curvatures computed normally and perpendicularly to edges. A comparison of various interest points detectors is presented in [Schmid 98]: it introduces a modified version of the Harris detector which uses Gaussian functions to compute the two-dimensional image derivatives, and that gives the best *repeatability* under rotation and scale changes (the repeatability being defined as a quantitative evaluation criteria which is the percentage of repeated interest points between two images). However this repeatability steeply decreases with significant scale changes: in such cases, a scale adaptive version of the Harris detector is required to allow point matching [Dufournaud 00]. But when no information on scale change is available, scale adaptation is not possible: matching features in such contexts becomes quite time consuming, scale being an additional dimension to search through. To avoid this, scale invariant feature detection algorithms have recently been proposed in the context of object recognition and image indexing [Lowe 99, Lindeberg 98, Mikolajczyk 01]. However, these methods generate much less features than the standard or scale adaptive detectors.

To match interest points, Schmid and Mohr have proposed an approach that uses local grayvalue invariants [Schmid 97], evaluated in the context of an image retrieval application. Local description vectors using differential rotation invariants are computed for each detected point, and potential matches are generated by a similarity measure given by a Mahalanobis distance between the vectors. The covariance used to compute the Mahalanobis distance between the invariant vectors must be properly estimated from large set of image samples. Since matches are generated without considering local region characteristic or geometric transformation between the two images, they can be ambiguous, and the initial match set must contain a sufficient number of inliers to allow the application of an outlier removal technique. Coarse-to-fine multi-resolution matching methods have been presented, which generate image pyramids so that only a few prominent features are present at the coarse levels [Rosenfeld 77]. The generated matches at the coarsest level guide matching process gradually up to the finest level(original image). Since small templates at coarse levels contain the signals of a local region at the finest level, the multi-resolution matching process guided by the matches

generated at coarse levels can be performed considering local region characteristic of original image, however these methods are unreliable under the significant viewpoint change (*i.e* small overlapping between images). An algorithm that produces dense matches from un-calibrated image pairs has been presented in [Lhuillier 03]. It is initiated by seed interest point matches established with a cross-correlation measure, which is effective when the viewpoint change between the images is rather small. A propagation of the seed matches is performed using geometric constraints.

2.1.2 Our approach

Principle. Our matching algorithm relies on *interest point group* matching, imposing a combination of geometric and signal similarity constraints, thus being more robust than approaches solely based on point signal characteristics [Jung 01].

An interest point group is a small set of neighbouring interest points, that represent a small region of the image: the matching of two groups in two images allow to compute an estimate of the actual geometric transformation between the corresponding regions. With groups composed of a small number of interest points, the corresponding region is small enough to ensure that a transformation $\mathcal{T}_{s,\theta}$ composed of a scale change s and a rotation change θ approximates fairly well the actual region transformation. The estimation of this geometric transformation is essential in the algorithm, as it is used for the following purposes:

- Group match hypotheses are qualified with a similarity measure based on the steered image derivatives computed in the corresponding points, which can be computed knowing $\mathcal{T}_{s,\theta}$,

- Similarly, the estimate of $\mathcal{T}_{s,\theta}$ enables to compute a correlation coefficient between the points of a group math hypothesis, which is used to validate it,

- Finally, once a group match is assessed as reliable, the corresponding geometric transformation $\mathcal{T}_{s,\theta}$ is propagated in the images to focus the search for new group matches: this reduces the number of match candidates to evaluate and thus enables robust and fast matching.

The algorithm does not require high order derivatives computation, and has proven to be very robust, *i.e.* able to provide numerous good matches while keeping the number of outliers very small, in different kinds of scenes and in a wide variety of conditions, tolerating noticeable scene modifications and changes in viewpoint and illumination. This robustness is due to the fact that the algorithm simultaneously checks point matches and groups matches, enabling consistent evaluation of the hypotheses.

Algorithm and chapter outline. The whole matching procedure is a seed-and-grow algorithm, that is initiated by a reliable interest point group match.

The first step of the algorithm is the detection of the interest points: section 2.2 briefly presents scale adaptive interest points, and introduces the point similarity measure we use.

Once the interest points of the two images to match are detected, local groups of interest points are established. Finding an initial group match must be made very cautiously, as a wrong initial group match would preclude the establishment of further matches and cause the whole algorithm to fail. Section 2.3 is the heart of the chapter: it depicts in details the establishment of a group match, according to a hypothesis generation and confirmation paradigm. A group match hypothesis is generated by a pair of point matches, on the basis of the point similarity measure. The match hypothesis defines an estimate of the geometric transformation between the image regions covered by the group: this transformation allows to compute the steered derivatives and a correlation coefficient between the matched points, and is used to establish further point matches within the considered group. Finally, a discriminancy check is performed to assess that the group match is a reliable one.

The propagation procedure is then invoked (section 2.4): the geometric transformation defined by the initial group match is used to focus the search for new group matches. During the propagation, more consistency checks are performed, in order to avoid problems caused by repetitive patterns or the presence of similar objects in the image: if such cases are detected, the initial group match hypothesis is discarded, and the whole procedure is started over.

Illustrative examples are given throughout the charter, and representative results are presented and discussed in section 2.5. In this last section, we also pro-

pose a way to derive quantitative measures of the spatial precision of the point matches, which is required by the Kalman filter in the SLAM solution (observation error model).

2.2 Interest points

To locate points in the image where the signal changes bi-directionally, the Harris corner detector computes the local moment matrix M of two normal gradients of intensity for each pixel $\mathbf{x} = (u, v)$ in the image [Harris 88]:

$$M(\mathbf{x}, \tilde{\sigma}) = G(\mathbf{x}, \tilde{\sigma}) \otimes \begin{pmatrix} I_u(\mathbf{x})^2 & I_u(\mathbf{x}) I_v(\mathbf{x}) \\ I_u(\mathbf{x}) I_v(\mathbf{x}) & I_v(\mathbf{x})^2 \end{pmatrix}$$

where $G(., \tilde{\sigma})$ is the Gaussian kernel of standard deviation $\tilde{\sigma}$, and $I_u(.)$ and $I_v(.)$ are the first order derivatives of the intensity respectively in the u and v directions. The eigenvalues (λ_1, λ_2) of $M(\mathbf{x}, \tilde{\sigma})$ are the principal curvatures of the auto-correlation function: the pixels for which they are locally maximum are declared as interest points. It has been shown in [Schmid 98] that interest points are more stable when the derivatives are computed by convolving the image with Gaussian derivatives:

$$I_u(\mathbf{x}, \sigma) = G_u(\mathbf{x}, \sigma) * I(\mathbf{x})$$

$$I_v(\mathbf{x}, \sigma) = G_v(\mathbf{x}, \sigma) * I(\mathbf{x})$$

where $G_u(., \sigma), G_v(., \sigma)$ are the first order derivatives of the Gaussian kernel of standard deviation σ along the u and v directions. The auto-correlation matrix is then:

$$M(\mathbf{x}, \sigma, \tilde{\sigma}) = G(\mathbf{x}, \tilde{\sigma}) \otimes \begin{pmatrix} I_u(\mathbf{x}, \sigma)^2 & I_u(\mathbf{x}, \sigma) I_v(\mathbf{x}, \sigma) \\ I_u(\mathbf{x}, \sigma) I_v(\mathbf{x}, \sigma) & I_v(\mathbf{x}, \sigma)^2 \end{pmatrix}$$

2.2.1 Scale Adaptive interest points

To maintain the derivatives stable with respect to the image scale change s, the Gaussian functions are normalised with respect to the scale change [Dufournaud 00]. The auto-correlation matrix of the scale adaptive Harris detector becomes:

$$M(\mathbf{x}, \sigma, \tilde{\sigma}, s) = G(\mathbf{x}, s\tilde{\sigma}) \otimes \begin{pmatrix} I_u(\mathbf{x}, s\sigma)^2 & I_u(\mathbf{x}, s\sigma) I_v(\mathbf{x}, s\sigma) \\ I_u(\mathbf{x}, s\sigma) I_v(\mathbf{x}, s\sigma) & I_v(\mathbf{x}, s\sigma)^2 \end{pmatrix} \quad (2.1)$$

where

$$I_u(\mathbf{x}, s\sigma) = sG_u(\mathbf{x}, s\sigma) * I(\mathbf{x})$$

$$I_v(\mathbf{x}, s\sigma) = sG_v(\mathbf{x}, s\sigma) * I(\mathbf{x})$$

Figure 2.1 compares the repeatability of the scale adaptive detector with the non adaptive one, with a 1.6 scale change.

If a rough estimate of the scale change between two images is known (*e.g.* with an initial guess on the camera motion), interest points are computed accordingly before attempting a matching procedure. When no scale change estimate is available, interest points are detected at different scale levels and matching is performed for each trial level.

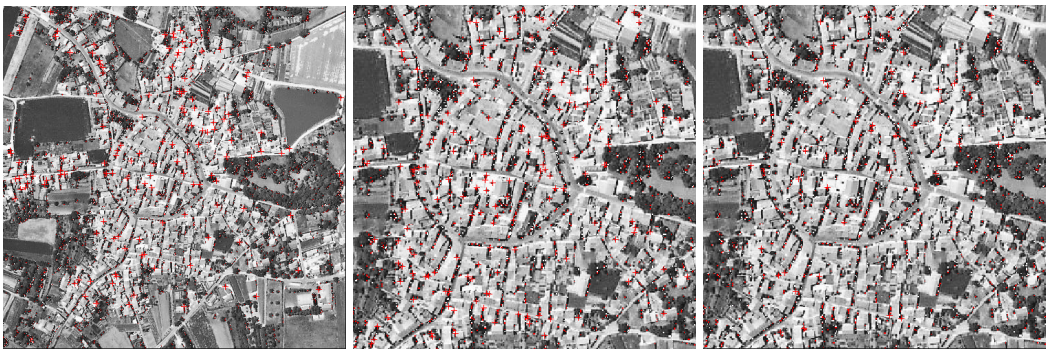


Figure 2.1: *Influence of the scale adaptation. The scale change between the left image and the two others is* 1.6*, interest points are the red crosses (for readability purpose, a high threshold on their detection has been applied to reduce their number). The detected points in the zoomed image without scale change consideration are shown in the middle image: the repeatability is here* 39%*. With the scale adaptive detector, the scale change estimate being set at* 1.5*, the repeatability raises up to* 82% *- rightmost image.*

2.2.2 Point similarity measure

If the geometric transformation $\mathcal{T}$ between two images I and I' is strictly equal to a scale change s and rotation change θ, and if the grayvalues of the images are

properly normalised, the following equality is satisfied for two matching points $(\mathbf{x}, \mathbf{x}')$ in the images:

$$\begin{pmatrix} I_u(\mathbf{x}, \sigma, \theta) \\ I_v(\mathbf{x}, \sigma, \theta) \end{pmatrix} = R(\theta) \begin{pmatrix} I_u(\mathbf{x}, \sigma) \\ I_v(\mathbf{x}, \sigma) \end{pmatrix} = \begin{pmatrix} I'_{u'}(\mathbf{x}', s\sigma) \\ I'_{v'}(\mathbf{x}', s\sigma) \end{pmatrix}$$

where

$$R(\theta) = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix}$$

and $I_u(\mathbf{x}, \sigma, \theta)$ and $I_v(\mathbf{x}, \sigma, \theta)$ are the steered Gaussian derivatives of the image in the direction θ [Freeman 91]. As a consequence, we can write:

$$R(\theta) M(\mathbf{x}, \sigma, \tilde{\sigma}) R(\theta)^T = M(\mathbf{x}', \sigma, \tilde{\sigma}, s)$$

Since

$$M(\mathbf{x}, \sigma, \tilde{\sigma}) = U \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} U^T$$

and

$$M(\mathbf{x}', \sigma, \tilde{\sigma}, s) = U' \begin{pmatrix} \lambda'_1 & 0 \\ 0 & \lambda'_2 \end{pmatrix} U'^T$$

where the columns of U and U' are the eigenvectors. The principal curvatures of the two matched points are therefore equal: $\lambda_1 = \lambda'_1$ and $\lambda_2 = \lambda'_2$.

For two matching points in two images of real 3D scenes, this equality is of course not strictly verified, because of signal noise, and especially because the true transformation of the image at these points is seldom strictly equal to a rotation and scale change. We define the *point similarity* $\mathcal{S}_p$ between two points on the basis of their eigenvalues:

$$\mathcal{S}_{p1}(\mathbf{x}, \mathbf{x}') = \frac{min(\lambda_1, \lambda'_1)}{max(\lambda_1, \lambda'_1)},\ \mathcal{S}_{p2}(\mathbf{x}, \mathbf{x}') = \frac{min(\lambda_2, \lambda'_2)}{max(\lambda_2, \lambda'_2)}$$

The maximum similarity is 1.0. The evolution of $\mathcal{S}_{p1}$ and $\mathcal{S}_{p2}$ with respect to scale and rotation changes for matching points is shown in figure 2.2: the expected value of the measure is always over 0.8, and decreases much less than the repeatability with changes in scale.

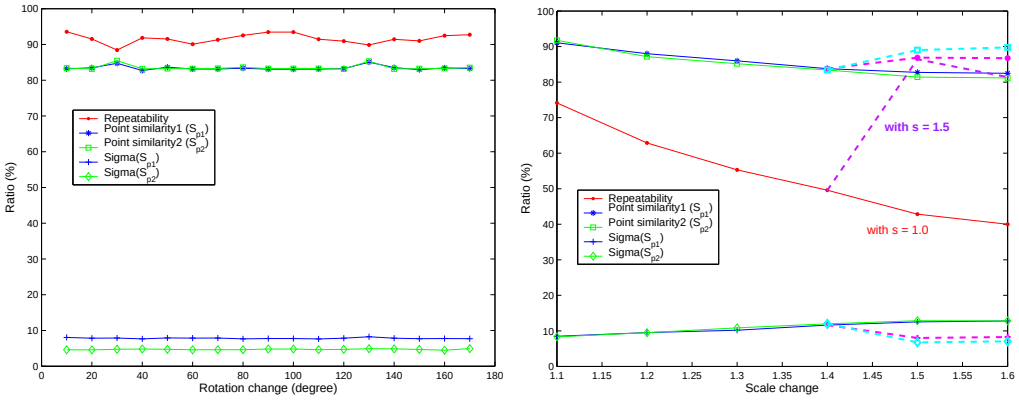


Figure 2.2: *Evolution of the mean and standard deviation of matching points similarity with known rotation and scale changes. On the right curve, the results computed by the scale adaptive detector with a scale change estimate set to* 1.5 *are shown with dashed lines.*

2.3 Interest point group matching

2.3.1 Grouping procedure

Once extracted in the two images I, I', the sets of interest points $\{\mathbf{x}\}, \{\mathbf{x}'\}$ are structured in *local groups*. A local group is made of a pivot point $\mathbf{g}_0$ and its n closest neighbours $\{\mathbf{g}_1, \ldots, \mathbf{g}_n\}$. To ensure that the image region covered by the points of a group is small enough, n is rather small (*e.g.* we use $n = 5$). The groups are generated by studying the neighbourhood of each point following a spiral pattern: the grouping process is stopped if the spiral meets the image border before n neighbours are found. Also, a maximum threshold on the distance between the neighbours points and the pivot is applied, to avoid the formation of groups that would cover a rather big region of the image in low texture areas for instance, where there are scarce interest points (the distance threshold is here set to $3\sqrt{\mathcal{D}}$, where $\mathcal{D}$ is the density of interest points in the image - figure 2.3). This implies that some points do not belong to any group: their matching will be processed later (see section 2.4.3).

After the grouping process, we end up with two group sets G, G':

$$\mathbf{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_N\}, \quad \mathbf{G}' = \{\mathcal{G}'_1, \ldots, \mathcal{G}'_M\}$$

where $\mathcal{G}_i$ denotes the local group $\mathcal{G}_i$ generated with the point $\mathbf{x}_i$ as a pivot:

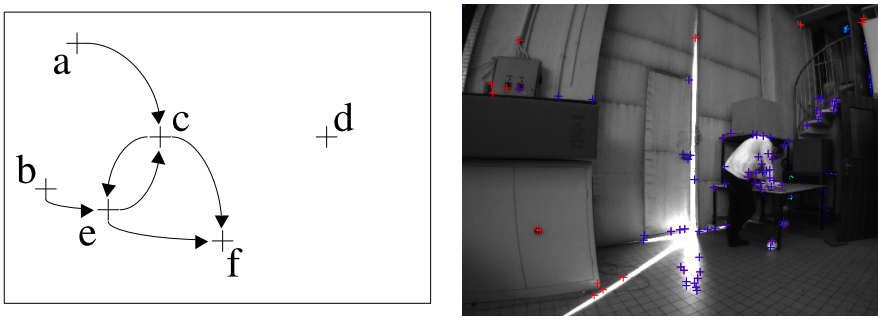


Figure 2.3: *Illustration of the point grouping procedure, with $n = 2$ for readability purposes. Groups have not been generated around points* $\mathbf{a}$ *and* $\mathbf{b}$ *as they are too close to the image border, and neither around* $\mathbf{d}$ *as no neighbour are close enough. Three groups have been generated, with points* $\mathbf{c}$, $\mathbf{e}$ *and* $\mathbf{f}$ *as a pivot (*$\mathbf{b} \rightarrow \mathbf{f}$ *means "*$\mathbf{b}$ *is a neighbour of* $\mathbf{f}$*"). In the right image, where $n = 5$, the blue points are the ones that have been grouped: only the points isolated or near the border have not been grouped.*

$$\mathcal{G}_i = \{\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_n\},\ \mathbf{g}_0 = \mathbf{x}_i$$

and the neighbours $\{\mathbf{g}_1, \ldots, \mathbf{g}_n\}$ are ordered by their distance to the pivot:

$$\| \mathbf{v}_1 \| < \ldots < \| \mathbf{v}_n \|$$

where the vectors $\mathbf{v}_i$ are defined as $\mathbf{v}_i = \mathbf{g}_i - \mathbf{g}_0$ and $\| \cdot \|$ is the norm operator.

2.3.2 Group match hypotheses generation

Given two local groups $(\mathcal{G}_i, \mathcal{G}'_j)$ in the images I, I', a set of hypotheses on the scale and rotation change $\mathcal{T}_{s,\theta}$ between the corresponding image regions is generated by evaluating the possible *vector matches* between the groups. A vector match $(\mathbf{v}_p, \mathbf{v}'_q)$ defines a scale change $s_{p,q}$ and a rotation change $\theta_{p,q}$ computed as follows:

$$s_{p,q} = \frac{\| \mathbf{v}'_q \|}{\| \mathbf{v}_p \|}, \quad \theta_{p,q} = tan^{-1} \frac{\mathbf{u}_p \wedge \mathbf{u}'_q}{\mathbf{u}_p \cdot \mathbf{u}'_q}$$

where $\mathbf{u}_p = \frac{\mathbf{v}_p}{\|\mathbf{v}_p\|}$, $\mathbf{u}'_q = \frac{\mathbf{v}'_q}{\|\mathbf{v}'_q\|}$ and $\wedge$ is the cross product.

The set of all possible vector matches $(\mathbf{v}_p, \mathbf{v}'_q)$ defined is first pruned by applying thresholds on the corresponding point matches similarity. This similarity

measure is then used to select the best match, which is afterward confirmed by a correlation measure.

Hypotheses pruning. The vector match $(\mathbf{v}_p, \mathbf{v}'_q)$ is considered as a potential match if the following constraints are satisfied:

$$\mathcal{S}_{p1}(\mathbf{g}_0, \mathbf{g}'_0) \quad \text{and} \quad \mathcal{S}_{p1}(\mathbf{x}_p, \mathbf{x}'_q) > T_{\mathcal{S}_{p1}}$$

$$\mathcal{S}_{p2}(\mathbf{g}_0, \mathbf{g}'_0) \quad \text{and} \quad \mathcal{S}_{p2}(\mathbf{x}_p, \mathbf{x}'_q) > T_{\mathcal{S}_{p2}}$$

$$|s_{p,q} - s_0| < T_s$$

The value of T_s is defined according to the precision on the initial estimate s_0 of the scale change, and the values on the two thresholds $T_{\mathcal{S}_{p1}}$ and $T_{\mathcal{S}_{p2}}$ are defined considering the evolution of the similarity with respect to scale change presented in section 2.2.2. Empirically, a value 0.6 for these two thresholds ensures that no good match are discarded.

To evaluate the vector matches $(\mathbf{v}_p, \mathbf{v}_q)$ that satisfy these constraints, we define the *vector similarity* $\mathcal{S}_v$ as:

$$\mathcal{S}_v(\mathbf{v}_p, \mathbf{v}'_q) = \frac{2}{(d(\mathbf{g}_0, \mathbf{g}'_0, s_0, \theta_{p,q}) + d(\mathbf{g}_p, \mathbf{g}'_q, s_0, \theta_{p,q}))}$$

where the function d is the sum of the squared differences between the steered derivatives of two potential matching points $(\mathbf{x}, \mathbf{x}')$:

$$d(\mathbf{x}, \mathbf{x}', s_0, \theta) = (I_u(\mathbf{x}, \sigma, \theta) - I'_u(\mathbf{x}, s_0\sigma))^2 + (I_v(\mathbf{x}, \sigma, \theta) - I'_v(\mathbf{x}, s_0\sigma))^2$$

The value of d can be computed thanks to the estimation of $\theta_{p,q}$, which allows to determine the steered derivatives on the basis of a linear combination of the two horizontal and vertical derivatives derivatives computed during the interest points extraction [Freeman 91]. Note that there is no need to recompute these derivatives with the current hypothetic scale change $s_{p,q}$, because the error between the actual scale change and the initial one (s_0) affects similarly all the difference measures of the steered derivatives.

Given a vector $\mathbf{v}_p$ in $\mathcal{G}_i$, the most similar vector $\mathbf{v}'_{\hat{q}}$ in $\mathcal{G}'_j$ among the remaining matches $(\mathbf{v}_p, \mathbf{v}_q)$ (if there exists some) is given by:

$$\hat{q} = arg\max_q \left[\mathcal{S}_v(\mathbf{v}_p, \mathbf{v}'_q)\right], \quad q = 1, \ldots, n$$

This defines a match hypothesis $\mathbf{h}_{p,\hat{q}} = (\mathbf{v}_p, \mathbf{v}_q)$ for the vector v_p, to which is associated the corresponding transformation $\mathcal{T}_{p,\hat{q}} = (s_{p,\hat{q}}, \theta_{p,\hat{q}})$. So a set of matching hypotheses $\mathbf{H}_{i,j}$ is defined for the two groups $(\mathcal{G}_i, \mathcal{G}'_j)$:

$$\mathbf{H}_{i,j} = \{\mathbf{h}_{p,\hat{q}}\}$$

The index p being comprised between 1 and n (the number of neighbours points in a group), a maximum number of n hypotheses are generated for two groups.

Best hypothesis selection. To select the best vector match hypothesis among $\mathbf{H}_{i,j}$, each hypothesis $\mathbf{h}_p$ is *completed* with additional potential point matches that satisfy geometric and steered derivative similarity constraints.

Given an hypothesis $\mathbf{h}_{p,\hat{q}}$, an additional point match $(\mathbf{g}_k, \mathbf{g}'_l)$ between two points of the groups $(\mathcal{G}_i, \mathcal{G}'_j)$ is evaluated. This additional point match defines a vector match $(\mathbf{v}_k, \mathbf{v}'_l)$, to which is associated a scale change $s_{k,l}$ and a rotation change $\theta_{k,l}$, is declared consistent with the hypothesis $\mathbf{h}_{p,\hat{q}}$ if the following three conditions are satisfied:

$$(p < k \quad \text{and} \quad \hat{q} < l) \quad \text{or} \quad (k < p \quad \text{and} \quad l < \hat{q}) \tag{2.2}$$

$$|s_{k,l} - s_{p,\hat{q}}| < T_s, \;\; |\theta_{p,\hat{q}} - \theta_{k,l}| < T_\theta \tag{2.3}$$

$$d(\mathbf{g}_k, \mathbf{g}'_l, s_0, \theta_{k,l}) < T_d \tag{2.4}$$

Condition (2.2) expresses a constraint on the distance of the points to the pivot (the indices of the group points are set according to their distance to the pivot - *cf* section 2.3.1).

Condition (2.3) expresses geometric constraints: the scale $s_{k,l}$ and rotation $\theta_{k,l}$ defined by the vector match $(\mathbf{v}_k, \mathbf{v}'_l)$ must be similar to those of the hypothesis

$\mathbf{h}_{p,\hat{q}}$. The transformation $\mathcal{T}_{p,\hat{q}}$ between the two local groups being an approximation of the actual transformation, we empirically choose a value of $20°$ for T_θ.

The threshold T_d of condition (2.4) is defined as follows:

$$T_d = \frac{1}{2}k(d(\mathbf{g}_0, \mathbf{g}'_0, s_0, \theta_{p,\hat{q}}) + d(\mathbf{g}_p, \mathbf{g}'_{\hat{q}}, s_0, \theta_{p,\hat{q}})) = \frac{k}{\mathcal{S}_v(\mathbf{v}_p, \mathbf{v}'_{\hat{q}})}$$

This last condition ensures that the additional matched points are similar, by comparing their steered derivatives similarity with the steered derivatives similarity of the initial vector match: the value of k is set high enough, to avoid the elimination of a potential good match.

Given a point $\mathbf{g}_k$ in $\mathcal{G}_i$, the best potential match $\mathbf{g}_{\hat{l}}$ in $\mathcal{G}'_i$ consistent with the hypothesis $\mathbf{h}_{p,\hat{q}}$ is the one that minimises the difference of the steered derivatives among the ones that satisfies the conditions (2.2) to (2.4):

$$\hat{l} = arg\min_l [d(\mathbf{g}_k, \mathbf{g}'_l, s_0, \theta_{k,l})] \tag{2.5}$$

Figure 2.4 illustrates an example of the determination of additional potential point matches given a initial vector match.

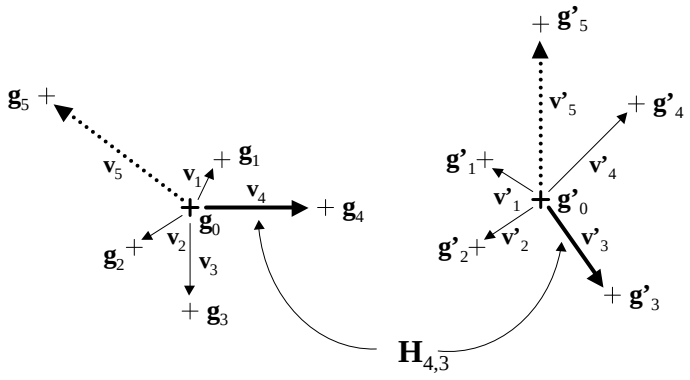


Figure 2.4: *Determination of additional potential point matches. Given an hypothesis* $\mathbf{h}_{4,3}$ *defined by the vector match* $(\mathbf{v}_4, \mathbf{v}'_3)$*, the best potential match of* $\mathbf{g}_5$ *is determined by evaluating geometric and point similarity constraints. In the end, the three point matches consistent with* $\mathbf{h}_{4,3}$ *are obtained:* $(\mathbf{g}_0, \mathbf{g}'_0)$*,* $(\mathbf{g}_4, \mathbf{g}'_3)$ *and* $(\mathbf{g}_5, \mathbf{g}'_5)$

Once all the potential additional point matches corresponding to an hypothesis $\mathbf{h}_{p,\hat{q}}$ are determined, the *group repeatability* $\mathcal{R}_g(\mathbf{h}_{p,\hat{q}})$ and the *group similarity* $\mathcal{S}_g(\mathbf{h}_{p,\hat{q}})$ of the hypothesis are computed as follows:

$$\mathcal{R}_g(\mathbf{h}_{p,\hat{q}}) = \frac{|\mathbf{h}_{p,\hat{q}}|}{n+1}, \quad \mathcal{S}_g(\mathbf{h}_{p,\hat{q}}) = \frac{|\mathbf{h}_{p,\hat{q}}|}{\sum_{i=1}^{|\mathbf{h}_{p,\hat{q}}|} d(\mathbf{g}_i, \mathbf{g}'_i, s_0, \theta_i)}$$

where $|\mathbf{h}_{p,\hat{q}}|$ is the number of matched points in $\mathbf{h}_{p,\hat{q}}$. The best match hypothesis of $\mathbf{H}_{i,j}$ is the one that maximises the group similarity among the ones that maximises the group repeatability.

2.3.3 Hypothesis confirmation

It is very important that the group match hypothesis $(\mathcal{G}_i, \mathcal{G}'_j)$ used as the search seed is a reliable one, otherwise the propagation procedure would fail to establish further matches: a reliable assessment is required. For that purpose, the zero-mean normalised correlation coefficient (ZNCC) between all the individual points matches $(\mathbf{g}_k, \mathbf{g}'_k)$ defined by the group match is evaluated:

$$z_k = \frac{\sum_{i=1}^{|W_k|} (I(\mathbf{p}_i) - \overline{I}(\mathbf{p}))(I'(\mathbf{p}'_i) - \overline{I}'(\mathbf{p}'))}{\left[\sum_{i=1}^{|W_k|} (I(\mathbf{p}_i) - \overline{I}(\mathbf{p}))^2 \sum_{i=1}^{|W_k|} (I'(\mathbf{p}'_i) - \overline{I}'(\mathbf{p}'))^2\right]^{1/2}}$$

where W_k is the set of pixels $\{\mathbf{p}\}$ defined by a local square correlation window centred on $\mathbf{g}_k$ in I, and on the pixel $\mathbf{p}'$ in I'. The pixels $\mathbf{p}'$ in I' are determined thanks to the geometric transformation $\mathcal{T}_{s_k,\theta_k}$ associated with the point match $(\mathbf{g}_k, \mathbf{g}'_k)$:

$$\mathbf{p}' = s_k R(\theta_k)(\mathbf{p} - \mathbf{g}_k) + \mathbf{g}'_k$$

and the intensity values $I'(p')$ of the corresponding correlation window are computed with bilinear interpolation (figure 2.5).

The point match $(\mathbf{g}_k, \mathbf{g}'_k)$ is considered valid if the corresponding ZNCC score is above a threshold T_z. The *matching strength* of the group match $(\mathcal{G}_i, \mathcal{G}'_j)$ is defined as:

$$S_{i,j} = m + \overline{z}, \quad m \leq n+1, \;\; T_z < \overline{z} \leq 1 \tag{2.6}$$

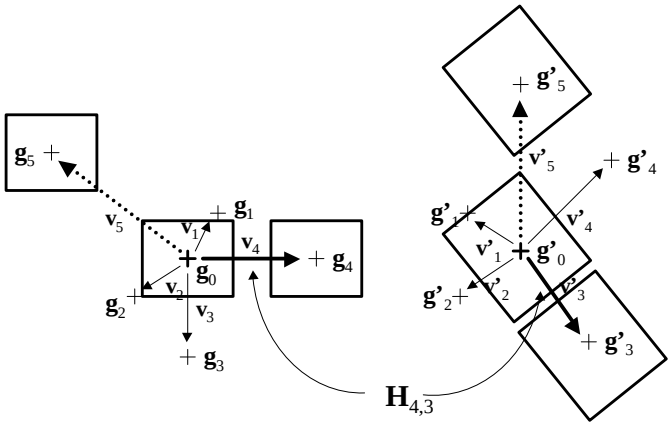


Figure 2.5: *Rotation and scale adaptive correlation on the potential matches generated with the best hypothesis* $\mathbf{h}_{4,3}$ *of the example shown in figure 2.4.*

where m is the number of validated point matches (*i.e.* the group match repeatability) and $\overline{z}$ is the average of the associated ZNCC scores: this kind of awkward sum discriminates the groups matches that have the same repeatability by the mean ZNCC score computed with the point matches.

A group match $(\mathcal{G}_i, \mathcal{G}'_j)$ is finally declared as a valid one if the associated matching strength is above a threshold T_S, and the scale and rotation changes of the corresponding geometric transformation $\mathcal{T}_{i,j}$ are determined as the mean of the scale an rotation changes defined by all the corresponding valid point matches.

At this stage, a last check has to be performed. Indeed, if the image presents repetitive patterns with a smaller space frequency than the groups area size, the neighbours points of a group can be very similar, thus misleading the point match establishment. In such cases, it is very likely that a gross error on the rotation estimate occur: to avoid this, only the groups whose neighbour are distinct enough are considered. This distinctness could be checked with first or higher order derivatives of intensity, but this would cause additional computations. We therefore define the *discriminancy* $\gamma_d(\mathcal{G}_i, \mathcal{G}'_j)$ of a group match $(\mathcal{G}_i, \mathcal{G}'_j)$ on the basis of the principal curvatures of their points:

$$\gamma_d(\mathcal{G}_i, \mathcal{G}'_j) = \frac{1}{2}\left[\left(\frac{\sigma(\lambda_1(\mathcal{G}_i)) + \sigma(\lambda_2(\mathcal{G}_i))}{\overline{\lambda}_1(\mathcal{G}_i)^2 + \overline{\lambda}_2(\mathcal{G}_i)^2}\right)^{1/2} + \left(\frac{\sigma(\lambda_1(\mathcal{G}'_j)) + \sigma(\lambda_2(\mathcal{G}'_j))}{\overline{\lambda}_1(\mathcal{G}'_j)^2 + \overline{\lambda}_2(\mathcal{G}'_j)^2}\right)^{1/2}\right] \tag{2.7}$$

where $\overline{\lambda}(\mathcal{G})$ and $\sigma(\lambda(\mathcal{G}))$ are respectively the mean and variances of the principal curvatures of the matched points in a group. The threshold on the discriminancy T_{γ_d} is set to a value sufficiently low to avoid good group matches removal: empirically, a value of 0.25 proved to be adequate. Figure 2.6 illustrates the effect of considering this discriminancy measure. Note that similar effects can occur with repetitive patterns at higher space scale: other measures to cope with this are proposed in the following section.

2.4 Propagation based matching

Once a reliable group match hypothesis is established, the propagation process searches for new group matches, using the corresponding estimated image transformation to focus the search for candidate groups in a restricted image area. As a consequence, very few group candidates are searched for, thus speeding up the establishment of new matches, and reducing the probability to establish wrong matches. Some consistency criteria are checked during the propagation, in order to detect false matches that may occur with large repetitive patterns in the images, and to detect cases where the geometry of the scene causes the propagation to fail.

2.4.1 Propagation process

Given a group match $(\mathcal{G}_i, \mathcal{G}'_j)$ and the corresponding transformation $\mathcal{T}_{i,j} = (s_{i,j}, \theta_{i,j})$, the closest group $\mathcal{G}_k$ of $\mathcal{G}_i$ in I is found. The position $\hat{\mathbf{x}}'_k$ of the point in the image I' that matches the pivot point $\mathbf{x}_k$ of $\mathcal{G}_k$ is estimated as

$$\hat{\mathbf{x}}'_k = s_{i,j} R(\theta_{i,j})(\mathbf{x}_k - \mathbf{g}_0) + \mathbf{g}'_0 \tag{2.8}$$

The set of candidate groups in I' that potentially match the group $\mathcal{G}_k$ is then

$$\mathbf{G}'_p = \{\mathcal{G}' | \, \mathcal{G}' \cap W(\hat{\mathbf{x}}'_k) \neq 0\},$$

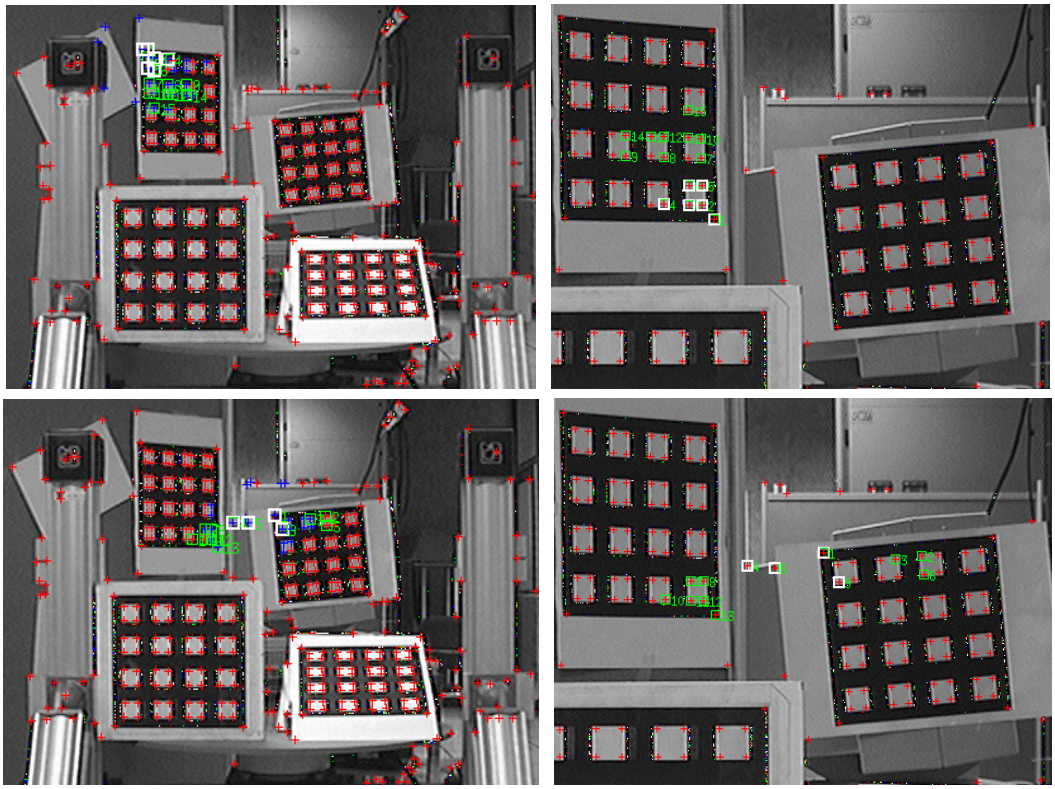


Figure 2.6: *Importance of the group discriminancy measure. The top images show the matches produced (green squares) after the propagation process explained next section, on the basis of the initial group match shown by white squares (the actual scale change is* 2.2*, it has been set as* 2 *to extract the points). Matches are geometrically consistent, although they are all wrong: because of the repetitive patterns, the initial group match procedure has been mislead. The bottom images show the matches produced using an other group match seed: they are good matches. The discriminancy measure for the first group match is* 0.13*, whereas it is equal to* 0.41 *for the second group match.*

where $W(\hat{\mathbf{x}}'_k)$ is a square window around $\hat{\mathbf{x}}'_k$ whose size is defined to cope with errors in the estimation of the transformation, and $\cap$ denotes the intersection of group with a region of the image: this intersection is non-null if a point of a group lies within this region.

Note that when the density of interest points in the image I is high, it can occur that the pivot $\mathbf{x}_k$ of the checked closest group $\mathcal{G}_k$ is a point that belongs to $\mathcal{G}_i$: if this point is a matching point in $(\mathcal{G}_i, \mathcal{G}'_j)$, there is only a single group in I' for which matches with $\mathcal{G}_k$ will be tried.

The group candidates are then evaluated and completed according to the procedure presented in section 2.3.2, with some slight differences at the hypotheses pruning stage: the prior knowledge on scale change s_0 is replaced by the current scale change estimate $s_{i,j}$, and a constraint on rotation change is added:

$$|\theta_{p,q} - \theta_{i,j}| < T_\theta$$

The number of hypotheses to evaluate is of course considerably reduced with respect to the first match group search. For the cases in which two points belonging to a candidate groups have already been matched in the $(\mathcal{G}_i, \mathcal{G}'_j)$, the group match hypotheses generation and pruning step are skipped, as these two points directly constitute a best vector match.

Once a new group match is established, the propagation process is re-iterated with the corresponding transformation estimate, until no more group match can be established.

2.4.2 Monitoring the propagation

There are two problems to consider during propagation: first, repetitive patterns of a size similar to the group size can lead to false matches, although the initial group match has passed the tests described in section 2.3.3. Second, the propagation process can fail when there are big depth discontinuities in the scene and large camera motions. These two problems are handled by checking a *local consistency* measure and a *global consistency* measure during the propagation phase.

Local consistency When similar objects in the scene are perceived in the image, the initial group match can be a wrong one, even if it has a high matching strength and discriminancy. This would not happen if we could match groups covering regions large enough to exceed the size of the repeated objects, but in such cases the assumption that a simple scale and rotation change transformation approximates well the actual image transformation for a group would be broken (and so would be all the group match procedure presented in section 2.3).

The idea to detect when such cases occur is to check whether the propagation process succeeds or not around the considered group match, in a region a few times bigger than the region covered by the group. The local consistency

$\gamma_{lc}(\mathcal{G}_i, \mathcal{G}'_j)$ of a group match $(\mathcal{G}_i, \mathcal{G}'_j)$ is defined as the the ratio between the group match repeatability and the repeatability computed in the expanded region $\Omega(\mathcal{G}_i)$ that surrounds the group $\mathcal{G}_i$:

$$\gamma_{lc}(\mathcal{G}_i, \mathcal{G}'_j) = \frac{(n+1)\,|\{\mathbf{m}_E\}|}{|\mathcal{V}_{i,j}|\,|\{\mathbf{x}_E\}|}, \quad \mathbf{x}_E, \mathbf{m}_E \in \Omega(\mathcal{G}_i)$$

where $n+1$ and $|\mathcal{V}_{i,j}|$ are the number of interest points and matches in the checked group match, $\mathbf{x}_E$ is total the number of interest points in the expanded region $\Omega(\mathcal{G}_i)$, and $\mathbf{m}_E$ is the total number of matched points in this region (these lest numbers include the group points).

If there is no other point match in $\Omega(\mathcal{G}_i)$ than the group matches, *i.e.* if we have $|\mathcal{V}_{i,j}| = |\{\mathbf{m}_E\}|$, the local consistency is not satisfied. Hence the minimum value of the local consistency is $T_{\gamma_{lc}} = \frac{(n+1)}{|\{\mathbf{x}_E\}|}$.

Figure 2.7 illustrates an example with a wrong initial group match caused by the presence of several similar objects in the images.

Global consistency. The local consistency of group matches is evaluated in locally expanded regions: if these regions are not large enough to exceed the size of the repeated patterns of objects, some locally consistent group match could still be wrong matches. In such cases, as propagation based matching proceeds, no group match consistent with the wrong matches are generated outside of the region defined by the repeated pattern. To assess such cases, the distribution of the generated matches during propagation is evaluated within the estimated global *overlap regions* of the images: when matches are not regularly established in these regions, the global consistency is not satisfied. Hence, complete propagation guarantees global consistency of generated matches, and incomplete propagation can be detected by the partiality of the distribution of matches in the overlapping regions.

The propagation completeness (or global consistency) γ_{gc} is defined as:

$$\gamma_{gc} = \frac{|\Omega_L|}{|\Omega_G|}$$

where $|\Omega_L|$ is the size of the local region of the image I which encloses the generated point matches so far, and $|\Omega_G|$ the size of the region of the image I that overlaps with the image I'. But this definition makes only sense if the interest

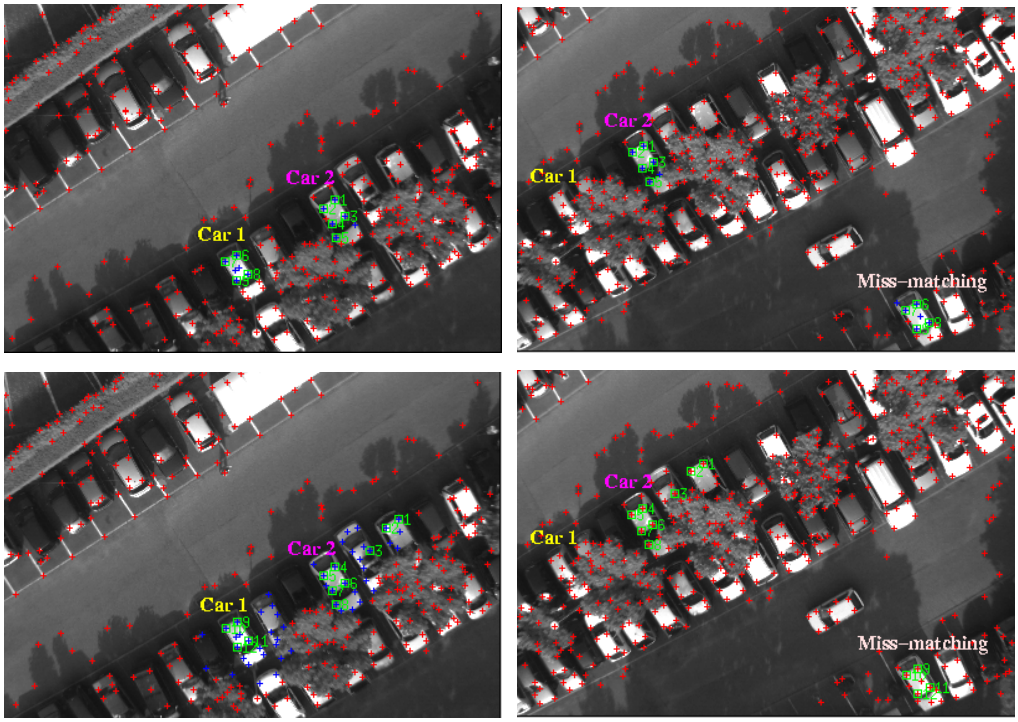


Figure 2.7: *Introduction of the local consistency measure. On the top two images, the blue points are the ones that belong to two groups. The ones that are surrounded by a green square have been matched: the "car 1" group is badly matched, although it passed all the confirmation tests of section 2.3.3. The bottom images show the pixels of the expanded regions in blue: no more additional matches have been determined around the "car 1" group, its local consistency measure is equal to* 0.22*, the minimum possible value for this region. In contrast, the local consistency measure for the "car 2" is equal to* 0.38.

points are regularly distributed in the images, which is seldom the case. Indeed, if most of the points are concentrated within Ω_L, the global consistency value would very low even if all the matches are consistent. Therefore, the number of interest points is used to define the size of the regions, and the global consistency is re-written as:

$$\gamma_{gc} = \frac{|\{\mathbf{x}_L\}|}{|\{\mathbf{x}_G\}|} \tag{2.9}$$

where $\{\mathbf{x}_L\} = \{\mathbf{x}|\mathbf{x} \in \Omega_L\}$ is the set of interest points within Ω_L, and $\{\mathbf{x}_G\}$ is the number of interest points the belong to the region of the image I that overlaps

with the image I': $\{\mathbf{x}_G\} = \{\mathbf{x}|\hat{\mathbf{x}}' \in I'\}$, in which $\hat{\mathbf{x}}'$ is the location of the point corresponding to $\mathbf{x}$ in I', estimated by the equation (2.8) using the transformation of the closest matched group to $\mathbf{x}$.

The maximum value of γ_{gc} is 1.0 when the repeatability of the interest points in the overlapping regions is 100%. Assuming that a repeatability higher than 50% within the overlapping regions is sufficient to generate good matches [Schmid 97, Jung 01], and taking into account the errors in the estimation of the overlapping regions, the threshold on the global consistency $T_{\gamma_{gc}}$ is empirically set to 0.4. Even though this threshold is not high, it is sufficient to distinguish partial distribution of matches in the overlapping regions of the images.

However, the presence of depth discontinuities in the scene can disturb the propagation process: occlusions induced by viewpoint changes can focus the search in wrong areas, and therefore break the generation of new group matches, and the propagation becomes incomplete. In such cases, if a new group match can be determined without focusing the match search (as in section 2.3), and if the corresponding transformation associated to this new match is consistent with the mean transformation associated to the already generated matches, the propagation based matching procedure is continued with the (slightly different) new transformation.

An example of an incomplete propagation caused by a big discontinuity in the scene is shown figure 2.8. The white squares represent matches established so far, the blue crosses indicate the interest points in the local region in which group matches are generated, and the white ones are all the interest points that belong to the overlapping region of the two images. The global consistency γ_{gc} is here equal to 0.28: the propagation process is stopped, new group matches are searched for among the white points.

Figure 2.9 shows all the matches established once a new reliable group match has been determined and a new propagation process based upon its corresponding transformation has been achieved: the completeness of the propagation γ_{gc} is now equal to 0.57, all the matches are declared consistent.

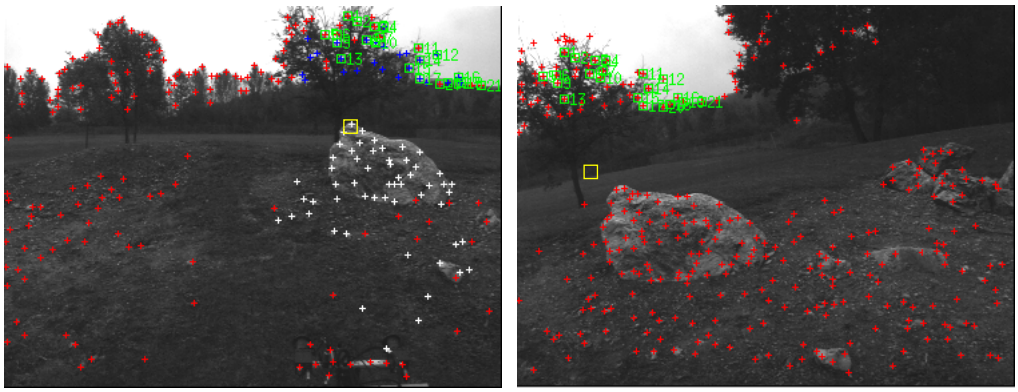


Figure 2.8: *Incomplete propagation in the presence of a big discontinuity of scene. The yellow square in the right image indicates the estimated location of the yellow square in the left image, on the basis of the matches established so far.*

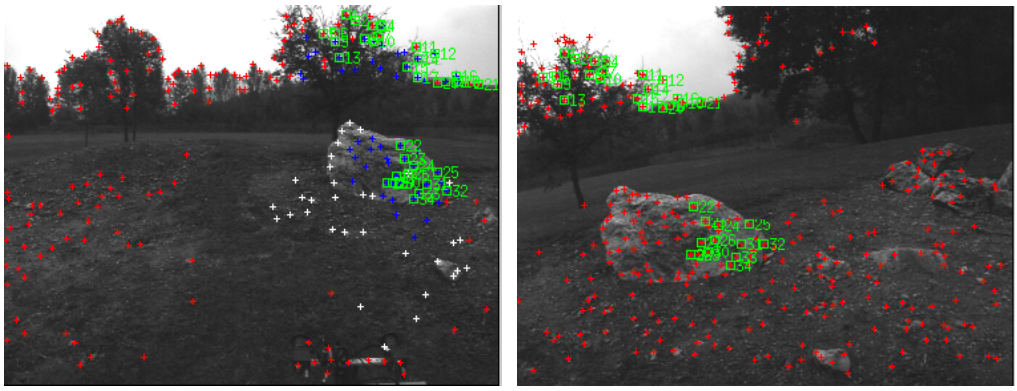


Figure 2.9: *Matches established after a new group match has been determined, and a new propagation procedure achieved.*

2.4.3 Non-grouped interest point matching

As noticed in section 2.3, some points are not associated to groups after the grouping procedure, mainly near the image borders. Hence no matches can be established for a group in the image I that is projected near the border in the image I' for instance. To avoid this, once the propagation procedure is achieved, matches are searched for the set of non grouped points of the image I. For each non grouped point $\mathbf{x}_b$, the set of candidate matches X_c in the image I' is defined as

$$\mathbf{X}_c = \{\mathbf{x} | \mathbf{x} \in W(\hat{\mathbf{x}}'_b)\}$$

where $\hat{\mathbf{x}}'_b$ is the estimated position of $\mathbf{x}_b$ in I', estimated with the image transformation associated to the closest group match of $\mathbf{x}_b$ according to the equation (2.8), and where $W(\hat{\mathbf{x}}'_b)$ is a search window centring $\hat{\mathbf{x}}'_b$.

The points comprised in $\mathbf{X}_c$ are evaluated according to the hypothesis pruning process presented in section 2.3.2: test on the similarity measure of the principal curvatures, and best match hypothesis selection according to the steered derivatives. The threshold on the ZNCC coefficient is then checked for the best hypothesis. Figure 2.10 shows the additional matches provided by this last propagation procedure.

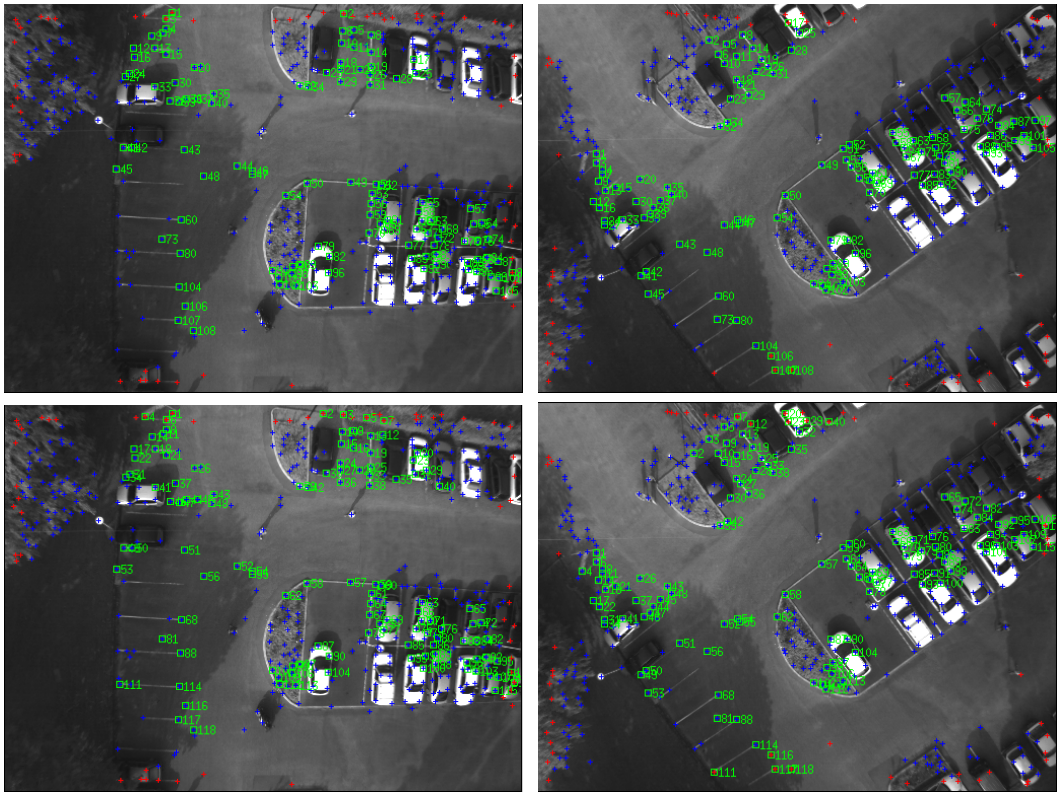


Figure 2.10: *Non-grouped interest point matching. The blue crosses are all the points around which a group has been defined (pivot points), the red ones are the others (either non grouped points, or points around which no group has been found).* 180 *matches have been established in the upper images. After the non-grouped point matching,* 10 *additional matches have been found - lower images, near the borders.*

2.5 Results

The algorithm has been tested with hundreds of image pairs of various 3D scenes, taken from hand-held cameras, various ground robots and a robotized airship, from positions that yield various complex image transformations. This section presents some results on illustrative examples.

2.5.1 Summary of the whole algorithm

Sections 2.2 to 2.4 presented the various processes and tests involved to yield robust matches: the procedure described below presents the sequence along which they are triggered.

1. Interest point computation for I and I'

2. Interest point grouping: generation of $\mathbf{G}$ and $\mathbf{G}'$
   $\mathbf{G}_0 \leftarrow \mathbf{G}$
   $\mathbf{G}'_0 \leftarrow \mathbf{G}'$

3. First group match generation
   while (no *groupMatch*) **and** $\mathbf{G}_0 \neq \emptyset$ **do**

   - randomly select a group $\mathcal{G}_i$ in $\mathbf{G}_0$
   - **while** (no *groupMatch*) **and** $\mathbf{G}'_0 \neq \emptyset$ **do**
     - randomly select a group ${\mathcal{G}'}_j$ in $\mathbf{G}'_0$
     - Generate vector match hypotheses, prune and select the best one
     - **if** $((\mathcal{G}_i, {\mathcal{G}'}_j)$ confirmed) *groupMatch* $\leftarrow (\mathcal{G}_i, {\mathcal{G}'}_j)$
       else remove ${\mathcal{G}'}_j$ from $\mathbf{G}'_0$
   - **if** (no *groupMatch*) remove $\mathcal{G}_i$ from $\mathbf{G}_0$

   if (no *groupMatch*) **end** (no matches found)
   else remove $\mathcal{G}_i$ from $\mathbf{G}$; remove $\mathcal{G}'_j$ from $\mathbf{G}'$; proceed to 4

4. Local consistency check
   if $(\gamma_{lc} < T_{\gamma_{lc}})$ discard group match $(\mathcal{G}_i, {\mathcal{G}'}_j)$; **goto** 3
   else proceed to 5

5. Propagation

6. Global consistency check
   if ($\gamma_{gc} < T_{\gamma_{gc}}$) **goto** 3
      if (new group match consistent) **goto** 5
      else discard all matches ; **goto** 3
   else proceed to 7

7. Non grouped point matching

The time performance of the algorithm mostly depends on the time to establish an initial reliable group match. The propagation procedure is then of course much faster to establish matches, its time being proportional to the number of actual matches in the image. Of course, time performance can be dramatically enhanced if an initial estimate of the image transformation is available (on the base on an information on the camera motion for instance). The time to establish a first reliable group match is indeed greatly reduced for two reasons: only the groups positioned in the first image so that they are in the overlapping region in the second image are checked for, and only a few group match hypotheses on the second image have to be checked.

2.5.2 Algorithm quality assessment

To define the effectiveness of the algorithm, *i.e.* to asses the number of outliers (wrong matches), two evaluation methods are used:

- For planar scenes, a homography $\mathcal{H}$ is determined thanks to a non-linear constrained least square method applied on the matched interest points coordinates. The distance between a point $\mathbf{p}$ and its match $\mathbf{q}$ is defined by

$$d_{\mathcal{H}}(p,q) = \| \mathbf{q} - \tilde{\mathbf{p}} \| \tag{2.10}$$

  where $\tilde{\mathbf{p}} = \mathcal{H}(\mathbf{p})$.

- For non-planar scenes, results are evaluated by the distance between the epipolar line of a point and its matching point. The fundamental matrix $\mathcal{F}$

that defines the epipolar lines is estimated by a least median square method [Zhang 95]. The distance is then defined as follows:

$$d_{\mathcal{F}}(p,q) = \frac{1}{2}(\| p - \tilde{p} \| + \| q - \tilde{q} \|) \tag{2.11}$$

where $\tilde{\mathbf{p}}, \tilde{\mathbf{q}}$ are the closest points on the epipolar lines respectively defined by $\mathbf{p}$ and $\mathbf{q}$ ($\tilde{\mathbf{q}}\mathcal{F}\mathbf{p} = 0$ and $\tilde{\mathbf{p}}\mathcal{F}^T\mathbf{q} = 0$).

The distance over which a match is declared as an outlier is three times the average distance computed for all the matches, and must never exceed 1.5 pixel. This latter constant threshold is not considered for planar scenes, to take into account the fact that the perceived scenes are not exactly planar. Note that the algorithm is applied on raw images, for which the camera distortion has not been corrected: the distortion is corrected before computing the homography or the fundamental matrix to assess the outliers.

2.5.3 Thresholds and parameters

The whole algorithm involves several thresholds and parameters, that have been empirically determined. The values to which we converged are summarised in table 2.1: for all the results presented here (and during the SLAM experiments), *these thresholds are not modified*.

Note that the threshold on the matching strength during propagation is set to a lower value than during the search for a first group match: for the establishment of a reliable first group match (the seed of the algorithm), its value is set to 3.7, which means at least 50% of interest point repeatability and a minimum value of 0.7 for the correlation score. But during propagation, most of the wrong hypotheses and candidates being eliminated thanks to the search focus, group matches having a lower matching strength are considered as good matches: a value of 2.6 make possible the generation of matches even if the point repeatability is only 33% (2 matches out of 6 members of group).

2.5.4 Various illustrative results

In all the following figures, interest points are represented by blue crosses when they are the pivot of a group, in red crosses otherwise. Good matches are denoted

Threshold on point similarity, $\mathcal{S}_{p1}$ and $\mathcal{S}_{p2}$	0.6
Threshold on scale change, T_s	0.6
Threshold on rotation change, T_θ	20^o
Threshold on ZNCC, T_z	0.6
Threshold on matching strength for 1^{st} match search, $T_{S(g)}$	3.7
Threshold on matching strength in propagation based matching, $T_{S(p)}$	2.6
Threshold on discriminancy, T_{γ_d}	0.25
Threshold on local consistency, $T_{\gamma_{lc}}$	0.25
Threshold on completeness of propagation(global consistency), $T_{\gamma_{gc}}$	0.4
Size of correlation window for ZNCC	9×9
Size of searching window in propagation based matching	21×21

Table 2.1: *Empirical setting of thresholds and parameters for robust matching.*

by green squares, and the matches assessed as wrong ones by red squares. The matching time is measured on a Sun UltraSparc 10 worstation.

Matching with small image transformations Figure 2.11 shows a matching result between two consecutive aerial images taken by an airship. The altitude of the camera being much bigger than the depth variations of the scene, which is rather flat, the wrong matches are estimated using the homography (equation 2.10). In this case, the detected "wrong matches" are not always actual outliers, but rather matches that involves points detected with a poor accuracy, as can be seen on the figure: indeed, the mean distance of rejected matches computed by the equation (2.10) is only 2.66 pixels.

Figure 2.12 shows the proportion of inliers found for a sequence of 64 consecutive images of the same scene, and the distance values $d_{\mathcal{H}}$ of the matches that have been considered as outliers. Again, in all these cases, most of the detected "wrong matches" are not always actual outliers: their average distance is 2.54 pixel.

Matching with complex image transformations A result with significant variations on the illumination conditions and with large camera motion is shown in figure 2.13. The sole wrong match detected is not an actual miss-match: the mean

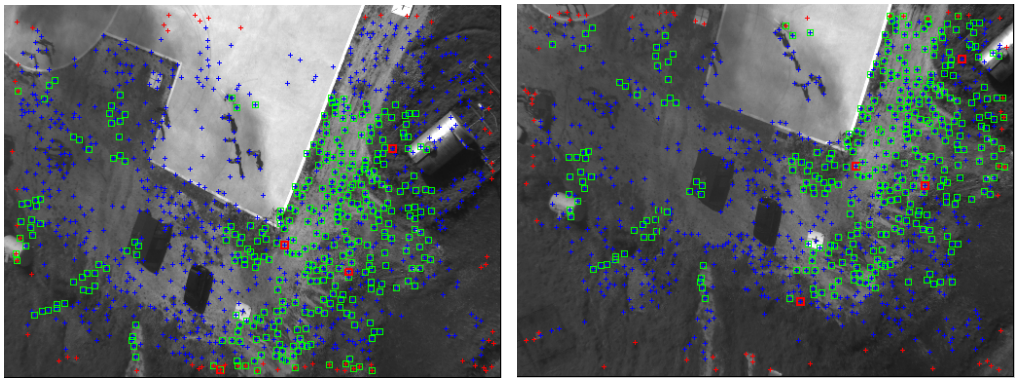


Figure 2.11: *Generated matches between a pair of consecutive aerial images (size* 384×512*), with slight change of translation and rotation: 4 wrong matches out of 342 matches, matching time is* $450ms$

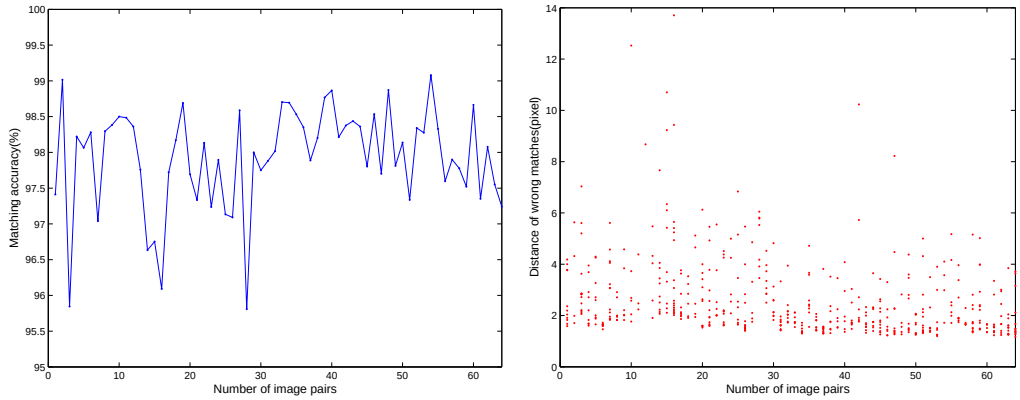


Figure 2.12: *Percentage of inliers matches for the 63 run of the algorithm, and distances of the points that have been considered as outliers.*

of the distances defined by equation (2.11) is here 0.24 pixel, and the distance of the discarded match is 0.84: it is rather an error due a poor accuracy of the interest point location.

Figure2.14 presents the matching results for a two non-consecutive aerial images taken from the sequence in figure2.12.

A matching result on two 480×640 indoor images is shown in figure 2.15: no matches have been established on the people who moved between the acquisitions. The algorithm has been applied on the original distorted images, and the bottom images show the epipolar lines defined by the fundamental matrix computed on

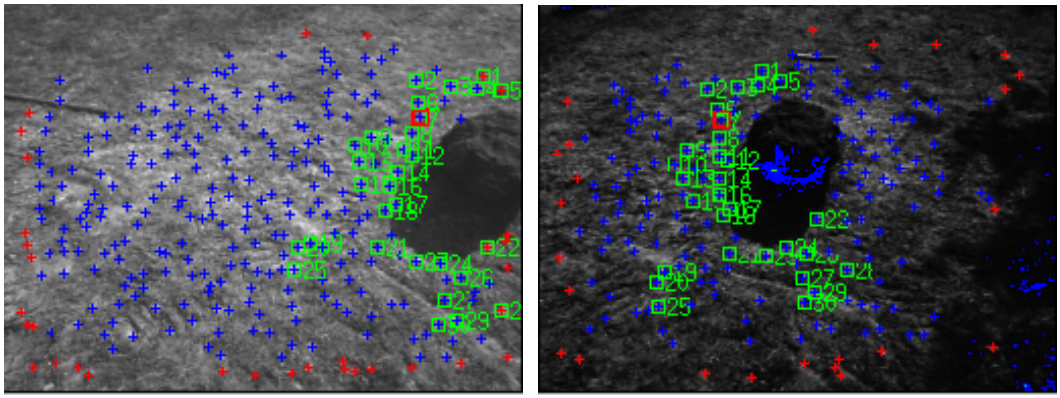


Figure 2.13: *Matching result for a* 192×256 *of an image pair, with significant illumination change, translation and small rotation. Only one wrong match is detected out of 30 matches and matching time,* $80ms$.

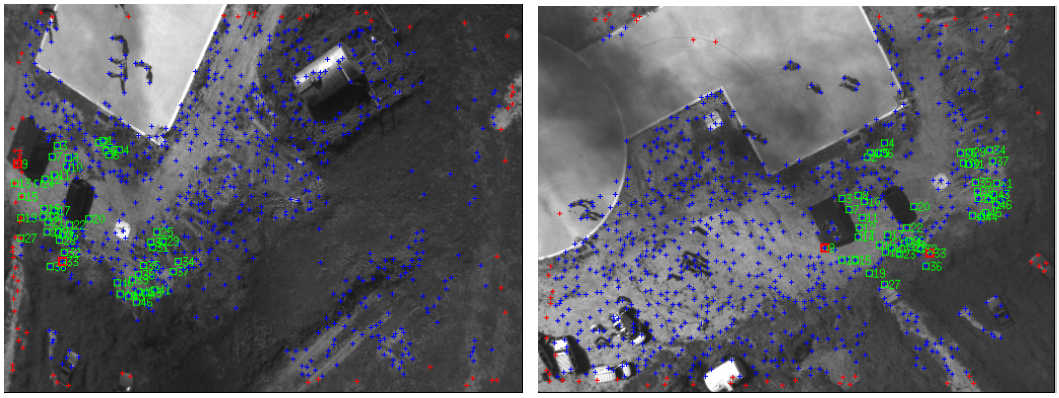


Figure 2.14: *Matching result for two* 384×512 *aerial images, with a big viewpoint change. 2 wrong matches are detected out of 42 matches - the matching time is* $640ms$.

the distortion-corrected images.

Figure 2.16 shows an example in which a large panoramic rotation of the camera makes the overlapping region of the two images really small.

Matching with significant scale changes In case of scale changes greater than 1.5, matching is performed with the scale adaptive interest point detector, with an *a priori* estimate of the scale change. However, for the following examples, 9 different scales have been checked ($s, 1.5s, 2s, \ldots, 5s$ with $s = 1$): each trial scale change is used as the initial scale change estimate.

Figure 2.15: *Matching with viewpoint change and moving elements in the scene. 4 wrong matches out of 204 matches are detected by evaluating with the distance computed using epipolar lines - the matching time is* $140ms$.

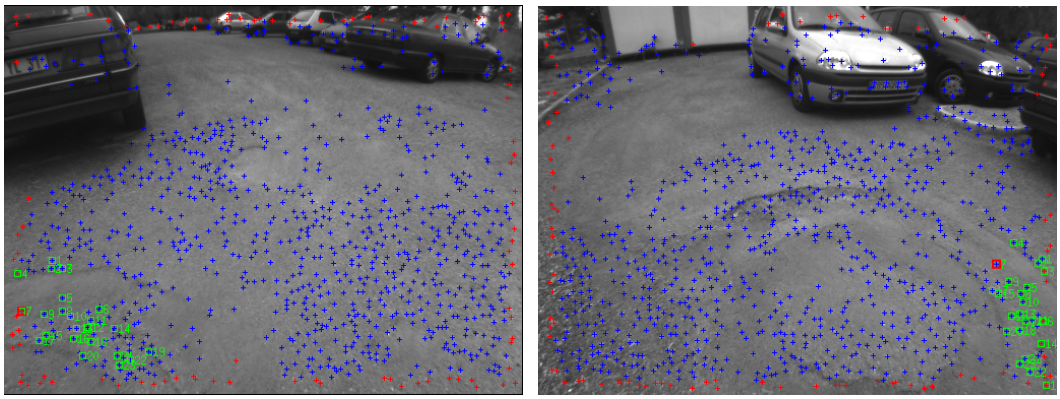


Figure 2.16: *Matching with a very small overlapping region caused by a big panoramic rotation of the camera.*

Figure 2.17 presents the matches found for two 384×512 images of a parking lot. When applying the 9 different scales, matches are generated only for the 1.5 scale change estimate, and not a single match has been generated for all the other scales.

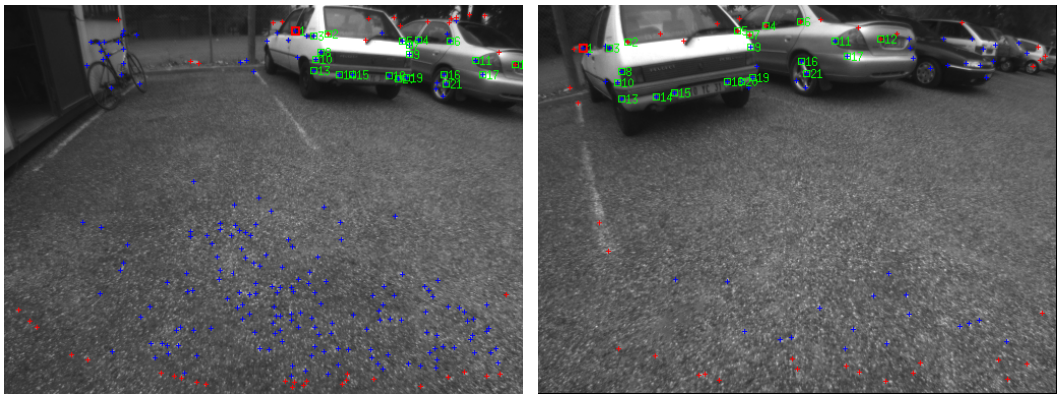


Figure 2.17: *Matching with a small scale change and viewpoint change. One wrong match is detected out of 21 matches, with an initial guess on the scale equal to* 1.5.

Figure 2.6 of section 2.3.3 showed a reliable group match satisfying the discriminancy constraint. Figure 2.18 shows the matching result on these images after the whole propagation, with an initial scale change estimate of 2 (no matches have been found for all the other scale change trials).

Figure 2.18: *Matching with scale change: only one wrong match out of 123 matches, the matching time is* $810ms$.

The images of figure 2.19 have been used as test images in [Dufournaud 00],

to evaluate a matching algorithm based on local grayvalue invariants. Our matching results are similar to the results presented in [Dufournaud 00], but only good matches have been generated with the two initial scale changes estimates 3 and 3.5, no outlier removal is necessary. Again, no matches have been found for all the other tested initial scale changes.

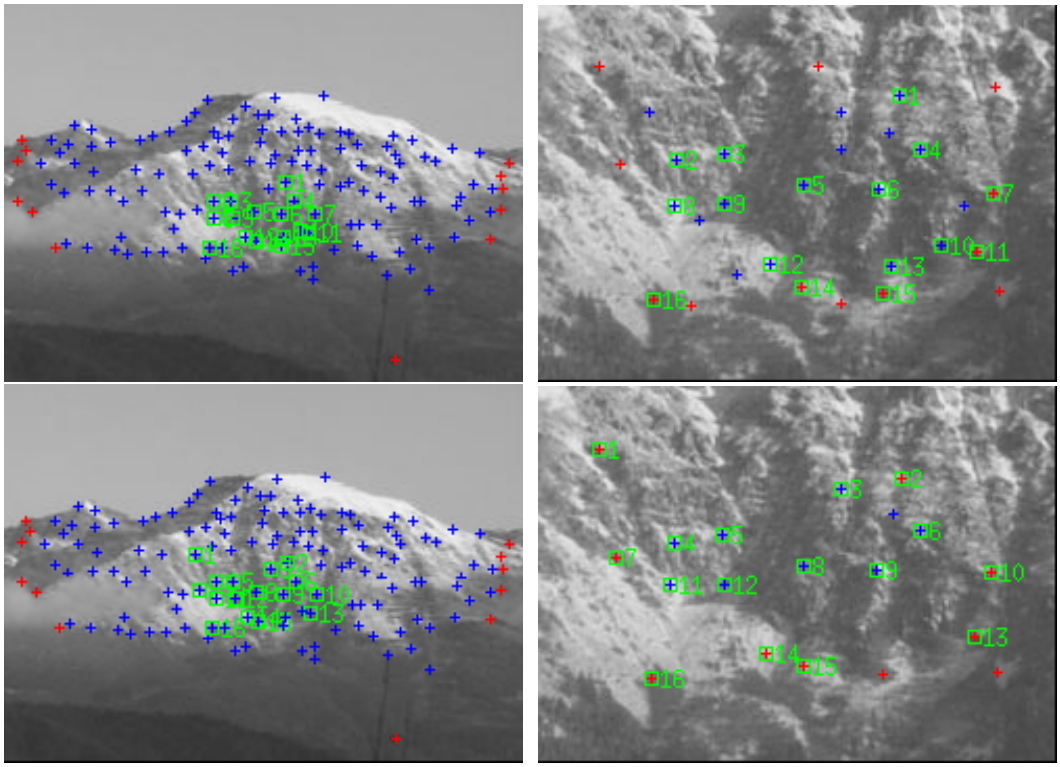


Figure 2.19: *Matching result at different scales (*182×250 *images). The upper images show the result with an initial scale estimate equal to* 3*, and the lowers with an initial estimate equal to* 3.5*. No wrong match are detected in both cases, the matching times is* $20ms$.

Finally, figure 2.20 shows the result with two aerial 384×512 images: matches were only found for a scale estimate equal to 3. All these results with significant scale change prove that an accuracy of 0.5 unity on the scale change is sufficient.

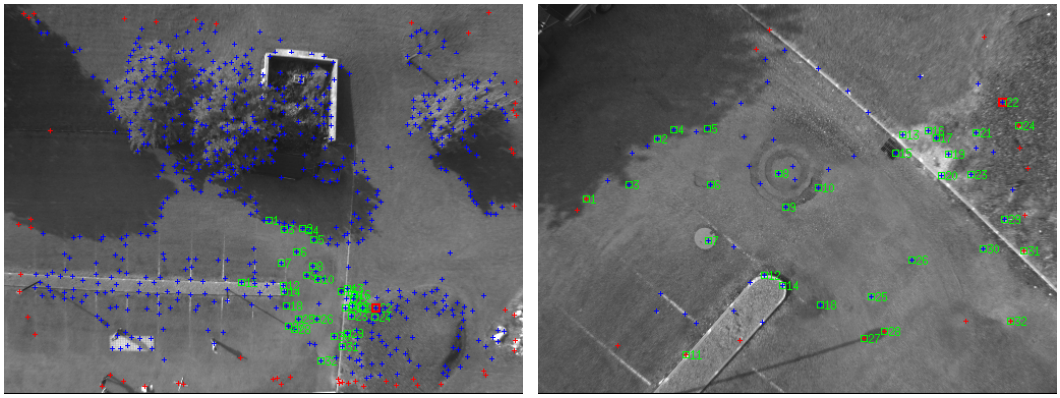


Figure 2.20: *Matching with rotation and scale change. One wrong match is detected out of 32 matches, the matching time is* $110ms$.

2.5.5 Matching accuracy estimation

Outliers can be rejected the way we detected them in the previous results, of thanks to algorithms such as robust fundamental matrix estimation based on RANdom SAmple Consensus (RANSAC). For the remaining inliers, the estimation of the spatial *precision* of the matches is essential in various applications, such as 3D scene reconstruction for instance.

Quantitative measures of the matches spatial uncertainty are estimated by fitting a Gaussian distribution to the correlation surface computed in a search area surrounding the matched points. Since the interest point location error is very small, *i.e.* in general sub-pixellic without significant scale change [Schmid 98], the considered area around the points is also small (5×5 pixels). The values of the pixels in the correlation window surrounding the point in the image I' is computed with the transformation associated to the corresponding group match transformation, using a bilinear interpolation, as presented in section 2.3.3.

As in [Singh 92] and [Nickels 02], we define the *response distribution* $\mathscr{R}(u,v)$ on the pixels of the window surrounding a matched point by :

$$\mathscr{R}(u,v) = e^{-k(1-z(u,v)}$$

where $z(u,v)$ is the score of ZNCC, and k is a normalisation factor chosen so that

$$\sum_{u,v \in W} \mathscr{R}(u,v) \approx 1$$

Is it clear that a pixel within W that yields a low response is less likely to be a good match than a pixel that yields a high response: the response distribution can therefore be interpreted as a probability distribution on the true match location. Under the assumptions of additive, zero mean and independent errors, the covariance matrix $\mathbf{P}_m$ on the location of the matched point in I' is given by

$$\mathbf{P}_m = \begin{pmatrix} \sigma_{u_m}^2 & \sigma_{u_m v_m} \\ \sigma_{u_m v_m} & \sigma_{v_m}^2 \end{pmatrix} \tag{2.12}$$

$$\sigma_{u_m}^2 = \frac{\sum_{u,v \in W} \mathcal{R}(u,v)(u-u_m)^2}{\sum_{u,v \in W} \mathcal{R}(u,v)}$$

$$\sigma_{v_m}^2 = \frac{\sum_{u,v \in W} \mathcal{R}(u,v)(v-v_m)^2}{\sum_{u,v \in W} \mathcal{R}(u,v)}$$

$$\sigma_{u_m v_m} = \frac{\sum_{u,v \in W} \mathcal{R}(u,v)(u-u_m)(v-v_m)}{\sum_{u,v \in W} \mathcal{R}(u,v)}$$

where u_m, v_m are the coordinates of the matched point. The diagonal elements of (2.12) represents the normalised variance in the vertical and horizontal directions and the off-diagonal elements are the normalised covariances.

The distribution of the estimated precision of the matches for four image pairs shown above are presented in figure 2.21, and table 2.2 compares the mean errors computed over all the matches for the same four image pairs: it appears that the smaller the transformation between the matched images is, the better the precision is. The figures indeed show that the estimated σ_u, σ_v of the matches found in figure 2.11 are smaller than those of figure 2.10: with small viewpoint changes, a high repeatability of the interest points is ensured, and their position is more stable than with bigger viewpoint changes.

When there is a significant scale change between the images to match, the location errors of matches are estimated in the high resolution image: the matching precision is therefore degraded, because a pixel in the low resolution image is split into several pixels in the high resolution image. Thus, the location error of interest points is proportional to the scale change.

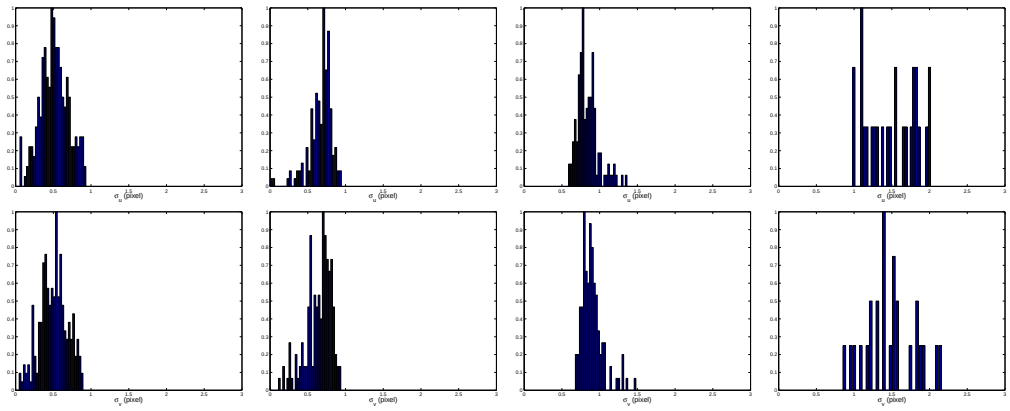


Figure 2.11 Figure 2.10 Figure 2.18 Figure 2.20

Figure 2.21: *Precision estimation: distributions of σ_u (top histograms) and σ_v (bottom histograms) for the set of matches generated by four image pairs shown above (figure references on the bottom line).*

Matches of fig.	2.11	2.10	2.18	2.20
Scale change	1.0	1.0	2.0	3.0
σ_u (pixels)	0.51	0.67	0.84	1.50
σ_v (pixels)	0.51	0.65	0.90	1.48

Table 2.2: *Comparison of the vertical and horizontal mean location error for the set of matches generated by four image pairs*

Chapter 3

Simultaneous Localisation And Mapping with stereovision

3.1 Introduction

To tackle the SLAM problem in real 3D environments, the first prerequisite is the ability to make relative observations in 3D: 3D coordinates of landmarks, and the 6 parameters of ego-motion. The second prerequisite is the ability to determine stable 3D landmarks, and the third is to guarantee robust data association.

The approach we propose here exclusively uses the calibrated binocular stereo bench, is applicable in 3D environments with 3D robot motions, and satisfies these three prerequisites, even without any additional positioning sensor:

- **Range to 3D points:** given a stereo frame taken by the pre-calibrated stereo bench, the interest points are detected on salient 3D objects in the scene projected onto the image frame, and their 3D coordinates are computed by stereovision.

- **Ego-motion:** once the interest points are detected in a stereo frame, only some of them are selected as landmarks, the remaining points are used to estimate the ego-motion of the robot. As shown in the literature [Mallet 00, Olson 01], stereovision-based ego-motion estimation is in general more accurate than dead-reckoning.

- **Stable 3D landmarks:** The number of detected interest points depends on both the detection threshold and the image texture: even in a weak textured image, a high number of interest points can be detected with a low threshold. Hence, a desired number of interest points per image is empirically pre-fixed, and the interest points detection process automatically adjusts the detection threshold according to the desired number. As a result, enough landmarks are always detected, independently from the perceived scene.

- **Robust data association:** The interest point matching algorithm presented in the previous chapter is used to match the current detected points with the landmarks that have already been mapped. Since the matching algorithm is effective with no prior knowledge on image transformation (except a rough scale change estimate), the observed point are correctly matched to existing landmarks without using the current estimated robot pose. Moreover, we will introduce a supplementary outlier removal technique to ensure that no wrong associations can occur.

3.1.1 Principle of the approach

The various algorithmic stages achieved every time a stereovision image pair is acquired are depicted in figure 3.1:

1. Stereovision: the 3D coordinate of interest points is provided by stereovision, along with an estimate of the covariances on the coordinates of the computed 3D interest points.

2. Interest points detection and matching: interest points are detected in one of the acquired images and are matched with the interest points detected in the previous step. If the robot is within an already mapped area, the current interest points are also matched with the image in which landmarks have initially been mapped.

3. Landmark selection: a set of selection criteria are applied to the matched interest points, in order to partition them in three sets: an observed-landmark set, a non-landmark set, and a candidate-landmarks set. The observed-landmarks are the detected points that matches already mapped landmarks,

non-landmarks points will solely be used to estimate the elementary motion between the current and the previous step, and candidate-landmarks are points that may be added to the the filter state, if they pass through the selection criteria during the next steps.

4. Visual motion estimation (VME): the interest points retained as ”non-landmarks” are used to estimate the 6 motion parameters between the previous and current steps, using a least-square minimisation. The associated covariances are also estimated, by propagating stereo and matching errors.

5. Position refinement: this is the update of the Kalman filter state.

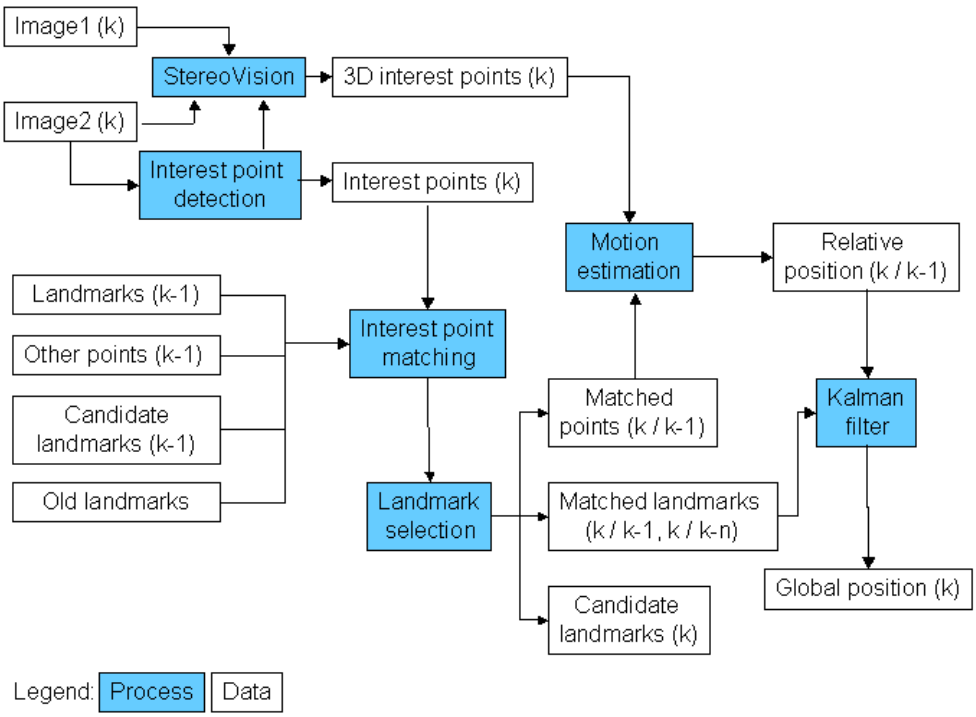


Figure 3.1: *Functional architecture of our approach to the SLAM problem on the sole basis of stereovision*

There is an important point to mention here: indeed, the stereovision bench being the only sensor used in our approach, its data is used both for the prediction stage (visual motion estimation) and the observation stage of the Kalman filter

(reperception of mapped landmarks). The prediction and observation are therefore not fully independent, which violates a necessary condition for the filter to be valid. However, *in the absence of calibration errors* of the stereovision bench, applying the prediction and the observation stages on two separate sets of points does not induce any correlation (which is clear if one considers that the points to estimate the ego-motion and the landmark points are perceived by two different stereovision benches): this is why the interest points are separated in different sets during the landmark selection. Still, the assumption that there is no calibration error is of course never satisfied, and the errors of the prediction and observation stages are therefore correlated. But note that this phenomenon occurs for most of the existing SLAM solutions, in which two different sensors are used for the prediction and observation stages, because the constant 3D transformation between the two sensors mounted on the robot is never perfectly known.

3.1.2 Outline of the chapter

This chapter organised as follows :

- The next section presents the Extended Kalman Filter formalism, and specify how we use it to develop a solution to the SLAM problem with stereovision only. In particular, the errors to identify are exhibited.
- Section 3.3 presents the stereovision algorithm, which provides the relative observations to landmarks.
- Section 3.4 presents the visual motion estimation algorithm, which constitutes the prediction stage of the Kalman filter.
- Section 3.5 depicts the strategy used to select the landmarks among the numerous interest points detected.
- Section 3.6 presents how the interest point matching algorithm presented in the previous chapter is used to tackle the data association problem in our context.

Finally a small summary concludes the chapter.

3.2 Kalman Filter Setup

3.2.1 Extended Kalman filter

The EKF is an extension of the standard linear Kalman filter, that linearises the nonlinear prediction and observation models around the predicted state. A general discrete nonlinear system is modelled as

$$x(k+1) = \mathbf{f}(x(k), u(k+1)) + v(k+1) \tag{3.1}$$

where $u(k)$ is a control input, v is a vector of temporally uncorrelated process noise with zero mean and covariance $\mathbf{P}_v(k)$.

The nonlinear observation model of the system is modelled as

$$z(k+1) = \mathbf{h}(x(k+1)) + w(k+1) \tag{3.2}$$

where h maps the state space into the observation space, and w is a vector of temporally uncorrelated observation errors with zero mean and covariance $\mathbf{P}_w(k)$.

In the Kalman filter framework, the state estimation encompasses three stages: prediction, observation and update of the state and covariance estimates.

Prediction. The state and observations are predicted using (3.1) and (3.2), and the state covariance is obtained through the linearisation of (3.1):

$$\hat{x}(k+1 \mid k) = \mathbf{f}(\hat{x}(k), u(k+1)) \tag{3.3}$$

$$\hat{z}(k+1 \mid k) = \mathbf{h}(\hat{x}(k+1 \mid k)) \tag{3.4}$$

$$\mathbf{P}_{\hat{x}}(k+1 \mid k) = \nabla \mathbf{f} \mathbf{P}_{\hat{x}}(k) \nabla \mathbf{f}^T + \mathbf{P}_v(k+1) \tag{3.5}$$

Observation. The true state $x(k+1)$ is observed, yielding the innovation $\nu(k+1)$, the corresponding covariance being obtained by linearising (3.2):

$$\nu(k+1) = z(k+1) - \hat{z}(k+1 \mid k) \tag{3.6}$$

$$\mathbf{S}(k+1) = \nabla\mathbf{h}\mathbf{P}_{\hat{x}}(k+1 \mid k)\nabla\mathbf{h}^T + \mathbf{P}_w(k+1) \tag{3.7}$$

Update. The update stage fuses the prediction and the observation to produce and estimate of the state and its associated covariance, according to the following formulas:

$$\hat{x}(k+1 \mid k+1) = \hat{x}(k+1 \mid k) + \mathbf{K}(k+1)\nu(k+1) \tag{3.8}$$

$$\mathbf{P}_{\hat{x}}(k+1 \mid k+1) = \mathbf{P}_{\hat{x}}(k+1 \mid k) - \mathbf{K}(k+1)\mathbf{S}(k+1)\mathbf{K}^T(k+1) \tag{3.9}$$

in which $\mathbf{K}(k+1) = \mathbf{P}_{\hat{x}}(k+1 \mid k)\nabla\mathbf{h}^T\mathbf{S}^{-1}(k+1)$ is the Kalman filter gain matrix.

3.2.2 Filter setup for SLAM with stereovision

In our approach, the state of the filter is composed of the 6 positioning parameters $\mathbf{x}_p = [\phi, \theta, \psi, t_x, t_y, t_z]$ of the stereovision bench (or the robot - the notations are the same as in the first part of the paper) and of a set of N interest points 3D coordinates $\mathbf{m}_i = [x_i, y_i, z_i]$, $0 < i \leq N$:

$$\mathbf{x}(k) = [\mathbf{x}_p, \mathbf{m}_1 \cdots \mathbf{m}_N] \tag{3.10}$$

The associated state covariance has the following form:

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_{pp}(k) & \mathbf{P}_{pm}(k) \\ \mathbf{P}_{pm}{}^T(k) & \mathbf{P}_{mm}(k) \end{bmatrix}$$

where $\mathbf{P}_{pp}$ represents the robot pose covariance, $\mathbf{P}_{mm}$ the landmark covariance and $\mathbf{P}_{pm}$ the cross-covariance between the robot pose and landmark estimates.

Prediction. Under the assumption that landmarks are stationary, the state prediction is:

$$\hat{\mathbf{x}}(k+1 \mid k) = \mathbf{f}(k+1)(\hat{\mathbf{x}}(k), \mathbf{u}(k+1)) \tag{3.11}$$

where $\mathbf{u}(k+1) = (\Delta\phi, \Delta\theta, \Delta\psi, \Delta t_x, \Delta t_y, \Delta t_z)$ is the visual motion estimation result between k and $k+1$ positions. The predicted state covariance (equation 3.5) is written as:

$$\begin{aligned}\mathbf{P}_{pp}(k+1 \mid k) &= \nabla_p \mathbf{f}(k+1)\mathbf{P}_{pp}(k)\nabla_p \mathbf{f}^T(k+1) + \\ &\quad \nabla_u \mathbf{f}(k+1)\mathbf{R}_u(k)\nabla_u \mathbf{f}^T(k+1) + \mathbf{P}_v(k+1) \quad (3.12)\end{aligned}$$

$$\mathbf{P}_{pm}(k+1 \mid k) = \nabla_p \mathbf{f}(k+1)\mathbf{P}_{pm}(k) \quad (3.13)$$

$$\mathbf{P}_{mm}(k+1 \mid k) = \mathbf{P}_{mm}(k) \quad (3.14)$$

where $\mathbf{R}_u$ represents *the error covariance of the visual motion estimation result*. Note that the covariance of landmarks is not changed in the prediction stage.

Observation. When observing the ith landmark, the observation model and the Jacobian of the observation function are written as:

$$\hat{\mathbf{z}}_i(k+1 \mid k) = \mathbf{h}_i(k+1)(\hat{\mathbf{x}}(k+1 \mid k)) \quad (3.15)$$

$$\nabla \mathbf{h}_i(k) = [\nabla_p \mathbf{h}_i(k), 0 \cdots 0, \nabla_{m_i} \mathbf{h}_i(k), 0 \cdots 0] \quad (3.16)$$

where $\mathbf{h}_i(k+1)(\hat{\mathbf{x}}(k+1 \mid k))$ is a function of the predicted robot state and the ith landmark in the state vector of the filter: it can then be also written as $\mathbf{h}_i(k+1)(\hat{\mathbf{x}}_p(k+1 \mid k), \hat{\mathbf{m}}_i(k+1 \mid k))$. The innovation and the associated covariance is written as:

$$\boldsymbol{\nu}_i(k+1) = \mathbf{z}_i(k+1) - \hat{\mathbf{z}}_i(k+1 \mid k) \quad (3.17)$$

$$\mathbf{S}_i(k+1) = \nabla \mathbf{h}_i(k+1)\mathbf{P}(k+1 \mid k)\nabla {\mathbf{h}_i}^T(k+1) + \mathbf{R}_i(k+1) \quad (3.18)$$

where $\mathbf{R}_i$ represents *the error covariance of ith landmark observation*.

Update. The update stage of the state and associated covariance estimates is made through the applications of equations (3.8) and (3.9), in which the gain matrix $\mathbf{K}$, innovation $\boldsymbol{\nu}$ and associated covariance $\mathbf{S}$ are respectively replaced by $\mathbf{K}_i$, $\boldsymbol{\nu}_i$ and $\mathbf{S}_i$.

If no observation are made (*i.e.* if no already mapped landmarks are re-perceived), the observation and update stages are not activated: the state and its covariance are just updated by the prediction stage.

When detecting a new landmark, it is added to the state vector of the filter, that becomes $\hat{\mathbf{x}}(k) = [\hat{\mathbf{x}}_p(k), \hat{\mathbf{m}}_1(k) \cdots \hat{\mathbf{m}}_N(k), \hat{\mathbf{m}}_{N+1}(k)]$ (its size increased of 3 units). The landmark initialisation model is:

$$\hat{\mathbf{m}}_{N+1}(k) = \mathbf{g}(k)(\hat{\mathbf{x}}_p(k), \mathbf{z}_{\mathbf{N+1}}(k)) \tag{3.19}$$

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_{pp}(k) & \mathbf{P}_{pm}(k) & (\nabla_p \mathbf{g}(k)\mathbf{P}_{pp}(k))^T \\ {\mathbf{P}_{pm}}^T(k) & \mathbf{P}_{mm}(k) & (\nabla_p \mathbf{g}(k)\mathbf{P}_{pm}(k))^T \\ \nabla_p \mathbf{g}(k)\mathbf{P}_{pp}(k) & \nabla_p \mathbf{g}(k)\mathbf{P}_{pm}(k) & {\mathbf{P}'}_{mm}(k) \end{bmatrix} \tag{3.20}$$

$${\mathbf{P}'}_{mm}(k) = \nabla_p \mathbf{g}(k)\mathbf{P}_{pp}(k)\nabla_p \mathbf{g}^T(k) + \nabla_z \mathbf{g}(k)\mathbf{R}_m(k)\nabla_z \mathbf{g}^T(k)$$

where $\mathbf{z}_{\mathbf{N+1}}(k)$ denotes the new landmark, ${\mathbf{P}'}_{mm}(k)$ is the associated covariance of $\mathbf{z}_{\mathbf{N+1}}(k)$,$\mathbf{g}(k)$ represents the initialisation function using the current robot pose estimate and $\mathbf{R}_m$ is *the error covariance of the new landmark*.

Errors to identify

To implement the Kalman filter in our context, the following errors must therefore be estimated:

- the landmark initialisation error (covariance matrix $\mathbf{R}_m$),
- the landmark observation error (covariance matrix $\mathbf{R}_i$ for the observed landmark i),
- and the error of the input control u, which is the visual motion estimation result (covariance matrix $\mathbf{R}_u$).

It is important to obtain a precise determination of these errors, as it will avoid the empirical "filter tuning" step.

Finally, note that in our approach, the lumped process noise v is set to 0, landmarks being stationary and the robot pose prediction being directly computed with the current pose and the ego-motion estimation.

3.3 Landmark relative measurements: stereovision

The 3D relative positions of the landmarks (interest points) with respect to the robot are determined by stereovision. Depending on the application context, dense stereovision might be required during the navigation (*e.g.* to detect the obstacles or to build a digital elevation map): in such a case, the interest points 3D positions are recovered in the 3D point image produced. If dense stereovision is not required, then stereovision needs only to be performed on the detected interest points (sparse stereovision).

When a new landmark is detected, the covariance matrix $\mathbf{R}_m$ on its state coordinates is totally defined by the stereovision error. Once identified, a new landmark is added in the filter state vector according equation (3.19), and its uncertainties are propagated into the state estimate covariance matrix according equation (3.20).

3.3.1 Dense stereovision

Principle of the algorithm. We use a classical pixel-based stereovision algorithm now widely used in robotics, that relies on an off-line calibrated binocular stereovision bench. The various stages of the algorithm are the following:

1. The images are first warped (rectified) so that epipolar lines are horizontal, which allows a dramatic optimisation to compute similarity scores between the pixels [Faugeras 93].

2. The disparities between matching pixels are estimated from the warped image pair thanks to a correlation-based pixel matching algorithm. We use either the ZNCC criteria or the Census matching criteria [Zabih 94]. Once

matches are generated using the correlation criteria, false matches are removed thanks to a reverse correlation.

3. Finally, the 3D coordinates of all the matched pixels are determined, using the relative 3D position between the two cameras of the bench provided by the off-line calibration stage.

Figure 3.2 shows the results of the dense stereovision algorithm, in three different contexts.

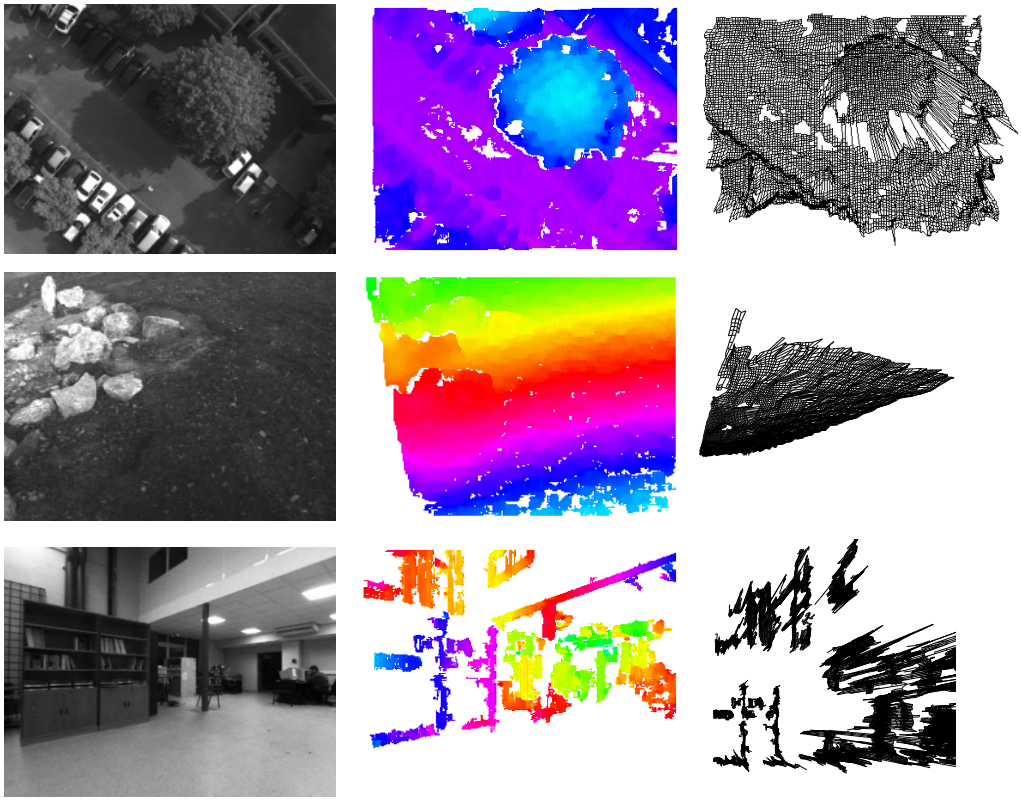


Figure 3.2: *A result of the dense stereovision algorithm, in various contexts. From left to right: one of the original image, disparity map (the disparities are inversely proportional to the depth of the pixels, and are shown here in a blue/close green/far colour scale), and 3D image, rendered as a mesh for readability purposes. Top to bottom: aerial images taken at about* $30\ m$ *altitude, images taken in a natural outdoor environment, and images taken in a indoor environment. Note that in indoor environments, the lack of texture makes the algorithm quite ineffective, and the 3D coordinates of only a few pixels can be recovered.*

An error model of dense stereovision. During the stereo matching phase, disparities are computed for integer values, the matching disparity d_m being the one that maximises the similarity score s. In order to get a finer disparity estimate, a sub-pixellic disparity d'_m is determined by fitting a parabola to the similarity score curve at its peak, the parabola being defined by the similarity scores computed at disparities $d_m - 1$, d_m and $d_m + 1$. The sub-pixellic disparity is the disparity that maximises the found parabola[1]:

$$d'_m = d_m + \frac{s(d_m - 1) - s(d_m + 1)}{2[(s(d_m) - s(d_m - 1) + (s(d_m) - s(d_m + 1)]} \tag{3.21}$$

The sources of errors in the disparity estimates are the image noise, the slight viewpoint change of the two cameras, the spatial sampling of the scene induced by the cameras, the size of the correlation window used, and the interpolation of the similarity score curve. Thorough studies of these phenomena can be found in the vision literature, but they lead to complex algorithms that are not tractable on-line.

In order to have an estimate of the disparity errors, we studied the distribution of the disparities on a set of 100 stereo image pairs acquired from the same position. As in [Matthies 92], it appeared that the distribution of the disparity computed on any given pixel can be well approximated by a Gaussian (figure 3.3). But a much more interesting fact is that there is a *strong correlation* between the shape of the similarity score curve around its peak and the standard deviation on the disparity: the sharper the peak, the more precise the disparity found (figure 3.3). This rather intuitive relation is the basis of our error model: on line, during the stereo matching phase, a standard deviation σ_d is associated to each computed disparity d, using the curvature of the similarity score curve at its peak. This is done at no extra computing cost, as this curvature is the one of the interpolating parabola at its peak.

Once matches are established, the coordinates of the 3D points are computed with the usual triangulation formula:

[1]Note that there does not exist any theoretic ground that justifies the use of a parabolic interpolation. It is only simple to compute, and it shifts the value of the integer disparity towards the neighbour that gives the highest similarity score, which is intuitive.

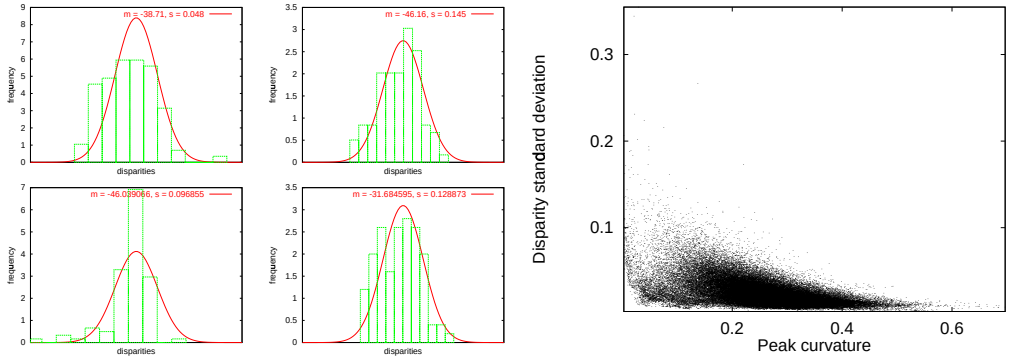


Figure 3.3: *Left: examples of some probability density functions of disparities computed on a set of 100 image pairs, with the corresponding Gaussian fit. Right: Standard deviation of the disparities as a function of the curvature of the similarity score curve at its peak.*

$$z = \frac{b\alpha}{d} \qquad x = \beta_u z \qquad y = \gamma_v z \tag{3.22}$$

where z is the depth, b is the stereo baseline, and α, β_u and γ_v are calibration parameters (the two latter depending on (u, v), the position of the considered pixel in the image). Using a first order approximation, it comes:

$$\sigma_z^2 \simeq (\frac{\partial z}{\partial d})^2 \sigma_d^2 = \frac{(b\alpha)^2}{d^4}\sigma_d^2 \tag{3.23}$$

substituting the definition of z defined in (3.22), we have:

$$\sigma_z = \frac{\sigma_d}{b\alpha} z^2 \tag{3.24}$$

which is a well known property of stereovision, *i.e.* that the errors on the depth grows quadratically with the depth, and are inversely proportional to the stereo baseline. The covariance matrix of the point coordinates is then:

$$\mathbf{R}_m = \begin{bmatrix} 1 & \beta_u & \gamma_v \\ \beta_u & \beta_u^2 & \beta_u\gamma_v \\ \gamma_v & \beta_u\gamma_v & \gamma_v^2 \end{bmatrix} (\frac{\sigma_d}{b\alpha} z^2)^2 \tag{3.25}$$

3.3.2 Sparse stereovision

Dense stereovision is quite time-consuming, and does not provide enough 3D information in weakly textured environments. Either to save time or to recover

enough 3D information, one can use a sparse stereovision algorithm, that only recovers the 3D informations for the interest points.

The 3D informations on the detected interest points are simply provided by applying the interest point matching algorithm on the two images of the stereovision pair. Note that by running the algorithm on rectified images (similarly as in dense stereovision), the matching candidates of a given interest point must lie on the same horizontal line: as a result, the number of candidates is drastically reduced (and so the computation time), and the robustness of the established matches is increased.

Once interest point matches are established, their 3D coordinates and the corresponding covariances are recovered using the same equations as for the dense stereovision algorithm. In the absence of a statistical study, the matching error in the image plane is determined by the the Gaussian fitting technique presented in section 2.5.5.

3.4 Prediction: visual motion estimation

3.4.1 Algorithm description

The interest points matched between consecutive image pairs and their 3D coordinates provided by stereovision are used to estimate the 6 displacement parameters between the images. The principle of the algorithm is simple (figure 3.5): it consists in establishing 3D points matches between two consecutive stereovision frames with the interest point matching algorithm. We then apply the least square minimisation technique presented in [Haralick 89] to estimate the 6 parameters of the corresponding motion.

As shown in figure 3.5, the 3D coordinates of most of the point matches between consecutive are provided by stereovision. Therefore, the 6 displacement parameters are estimated with a set of 3D corresponding points and the 3D-3D pose estimation is used to infer the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{T}$. To simplify the problem, the centroids of each set of points are computed and they are translated in space so that these lie at the origin and $\mathbf{R}$ is then computed. $\mathbf{T}$ is defined using the computed rotation.

The important point here is to get rid of the outliers (wrong matches), as they

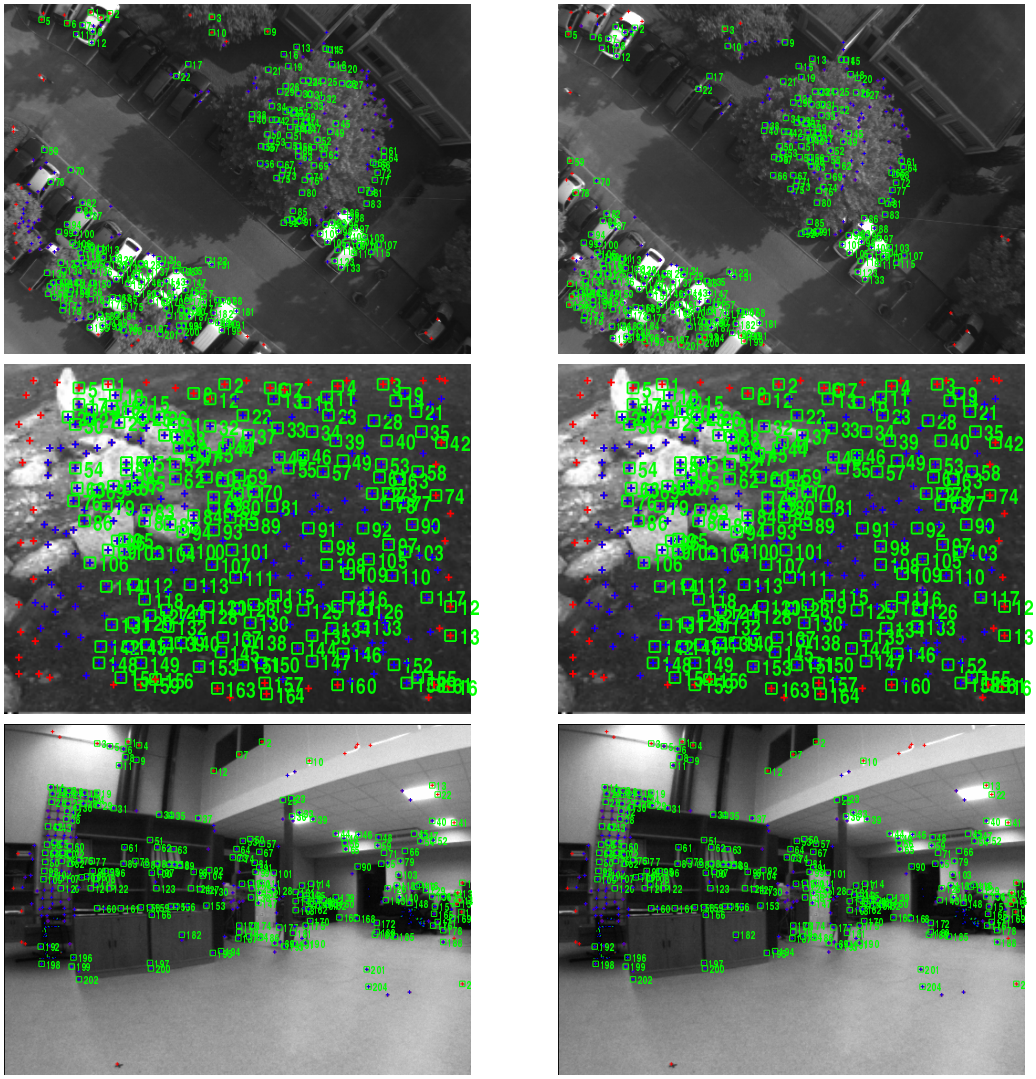
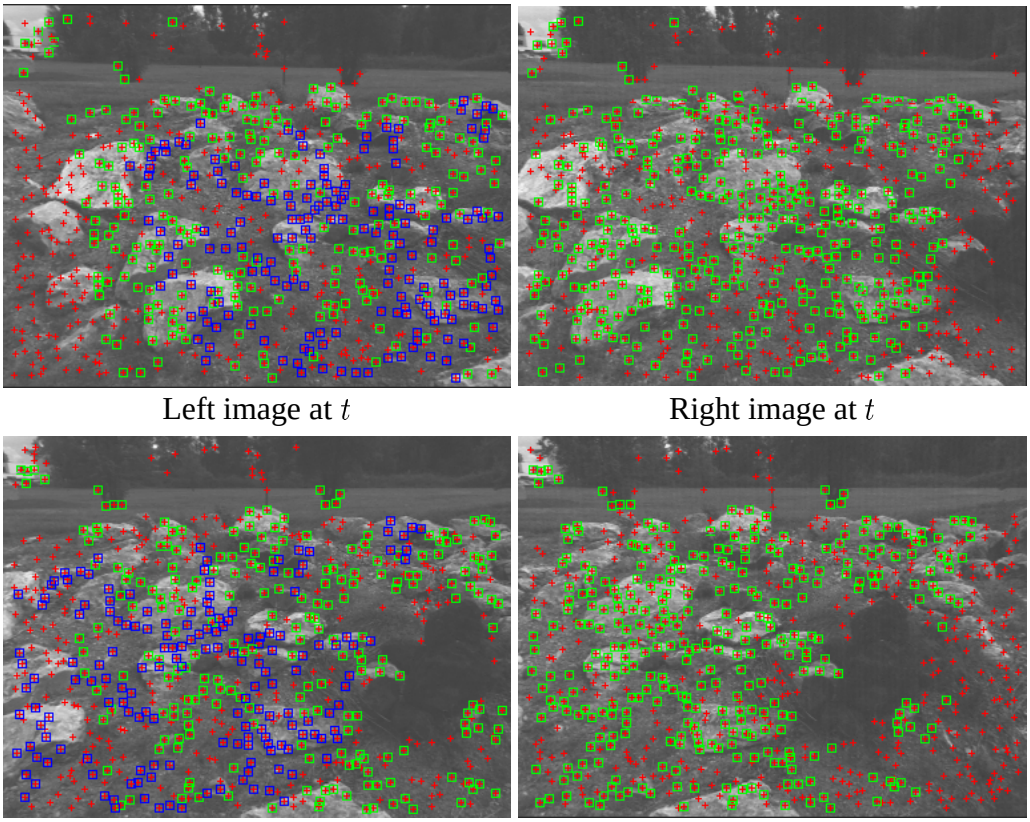


Figure 3.4: *Interest points matched between the two images of a stereovision pair (same images as in figure 3.2). Note that in the indoor images, some interest points have been matched on areas where the dense stereovision algorithm fails to establish matches.*

considerably corrupt the minimisation result. The outliers could be rejected using the epipolar constraint defined by the fundamental matrix computed on the basis of the matches. However, the computation of this matrix is quite sensible to the small errors in the positions of the matched points and to the outliers themselves. Also, such an outlier removal technique will not cope for stereovision errors, such as the ones that occur along depth discontinuities for instance: inliers matches in the image plane might become outliers when considering the corresponding 3D



Left image at t Right image at t

Left image at $t+1$ Right image at $t+1$

Figure 3.5: *Visual motion estimation between consecutive stereo frames, using the sparse stereovision algorithm. A translation to the right occurred here between time t and time $t+1$. The red crosses indicates the interest points detected in the images, the green squares are the matched points between two images of a stereo pair, and the blue ones are the points matched between two consecutive images. The points that have been both matched in the stereo images and the successive left images are used to estimate the 6 parameters of the motion.*

coordinates.

Therefore, we developed a specific outlier rejection method that consider both matching and stereovision errors. It assumes that the first 3D transformation estimates provided by all the matches is close to the true one, which is the case because the matching algorithm generates a large majority of inliers.

First, matches that imply a 3D point whose uncertainties in their 3D coordinate are over a threshold, are discarded (the threshold is empirically determined

by a statistical analysis of stereovision errors). Then, the remaining matches are analysed according to the following procedure:

1. A 3D transformation is determined by the least-square minimisation, and the mean and standard deviation of the residual errors are computed.

2. A threshold defined as k times the residual error standard deviation is defined (k should be at least greater than 3). The 3D matches whose error is over the threshold are eliminated.

3. k is set to $k-1$ and the procedure is re-iterated until $k=3$.

This outlier rejection algorithm guarantees a precise 3D motion estimation (see results in sections 3.4.2 and 4.1), which can then be used during the prediction stage of the Kalman filter (section 3.2.1).

3.4.2 Motion estimation errors

Given a set of 3D matched points $\hat{\boldsymbol{\mathcal{Q}}} = [\mathbf{X}_1,,,\mathbf{X}_N,\mathbf{X}_1,,,\mathbf{X'}_N]$, the function which is minimised to determine the corresponding motion is the following [Haralick 89]:

$$J(\hat{\mathbf{u}},\hat{\boldsymbol{\mathcal{Q}}}) = \sum_{n=1}^{N}(\mathbf{X'}_n - R(\hat{\phi},\hat{\theta},\hat{\psi})\mathbf{X}_n - [\hat{t}_x,\hat{t}_y,\hat{t}_z]^T)^2 \tag{3.26}$$

where $\hat{\mathbf{u}} = (\hat{\phi},\hat{\theta},\hat{\psi},\hat{t}_x,\hat{t}_y,\hat{t}_z)$. $\hat{\mathbf{u}}$ and $\hat{\boldsymbol{\mathcal{Q}}}$ can be written with random perturbations:

$$\hat{\mathbf{u}} = \mathbf{u} + \Delta\mathbf{u}, \qquad \hat{\boldsymbol{\mathcal{Q}}} = \boldsymbol{\mathcal{Q}} + \Delta\boldsymbol{\mathcal{Q}}$$

where the true $\mathbf{u}$ and $\boldsymbol{\mathcal{Q}}$ are not observed. In order to measure the uncertainty of local motion estimation, the uncertainties of 3D matching points set are propagated to the optimal motion estimate $\hat{\mathbf{u}}$. Assuming the optimal motion estimate minimises the cost function, the Jacobian of the cost function is 0, and the uncertainties of landmarks and their observation can be propagated by taking Taylor series expansion of the Jacobian around $\mathbf{u}$ and $\boldsymbol{\mathcal{Q}}$, as shown in [Haralick 94]:

$$\begin{aligned} g(\mathbf{u},\boldsymbol{\mathcal{Q}}) &= g(\mathbf{u}+\Delta\mathbf{u},\boldsymbol{\mathcal{Q}}+\Delta\boldsymbol{\mathcal{Q}}) - \frac{\partial g}{\partial \mathbf{u}}(\mathbf{u}+\Delta\mathbf{u},\boldsymbol{\mathcal{Q}}+\Delta\boldsymbol{\mathcal{Q}})\Delta\mathbf{u} - \\ &\quad \frac{\partial g}{\partial \boldsymbol{\mathcal{Q}}}(\mathbf{u}+\Delta\mathbf{u},\boldsymbol{\mathcal{Q}}+\Delta\boldsymbol{\mathcal{Q}})\Delta\boldsymbol{\mathcal{Q}} \end{aligned} \tag{3.27}$$

where $g = \frac{\partial J}{\partial \hat{\mathbf{u}}}$ is the Jacobian of the cost function , and $\frac{\partial g}{\partial \mathbf{u}}$ is the Hessian of the cost function with respect to $\mathbf{u}$. The Hessian $\frac{\partial g}{\partial \mathbf{u}}$ is positive definite for all $(\mathbf{u}, \boldsymbol{\mathcal{Q}})$ because the relative extremum of the cost function is a relative minimum: this guarantees the existence of the reciprocal of the Hessian. Since $\hat{\mathbf{u}}$ and $\mathbf{u}$ minimises $J(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})$ and $J(\mathbf{u}, \boldsymbol{\mathcal{Q}})$, $g(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})$ and $g(\mathbf{u}, \boldsymbol{\mathcal{Q}})$ are set to 0 in (3.27). The random perturbation $\Delta\mathbf{u}$ and its covariance are then computed as follows:

$$\Delta\mathbf{u} = -(\frac{\partial g}{\partial \mathbf{u}}(\mathbf{u} + \Delta\mathbf{u}, \boldsymbol{\mathcal{Q}} + \Delta\boldsymbol{\mathcal{Q}}))^{-1} \frac{\partial g}{\partial \boldsymbol{\mathcal{Q}}}(\mathbf{u} + \Delta\mathbf{u}, \boldsymbol{\mathcal{Q}} + \Delta\boldsymbol{\mathcal{Q}})\Delta\boldsymbol{\mathcal{Q}} \qquad (3.28)$$

$$\hat{\Sigma}_{\Delta\mathbf{u}} = (\frac{\partial g}{\partial \mathbf{u}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}}))^{-1} \frac{\partial g}{\partial \boldsymbol{\mathcal{Q}}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}}) \Sigma_{\Delta\boldsymbol{\mathcal{Q}}} \frac{\partial g}{\partial \boldsymbol{\mathcal{Q}}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})^T (\frac{\partial g}{\partial \mathbf{u}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}}))^{-1} \qquad (3.29)$$

Considered that $\mathbf{X}_n$ and $\mathbf{X'}_n$ are not correlated, the covariance estimate $\mathbf{P}_{\hat{\mathbf{u}}}$ can be also written as:

$$\mathbf{P}_{\hat{\mathbf{u}}} = (\frac{\partial g}{\partial \mathbf{u}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}}))^{-1} (\Lambda_{\mathbf{X}} + \Lambda_{\mathbf{X}'}) (\frac{\partial g}{\partial \mathbf{u}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}}))^{-1} \qquad (3.30)$$

where

$$\Lambda_{\mathbf{X}} = \sum_{n=1}^{N} \frac{\partial g}{\partial \mathbf{X}_n}(\hat{\mathbf{u}}, \mathbf{X}_n) \mathbf{P}_{\mathbf{X}_n} (\frac{\partial g}{\partial \mathbf{X}_n}(\hat{\mathbf{u}}, \mathbf{X}_n))^T$$

$$\Lambda_{\mathbf{X}'} = \sum_{n=1}^{N} \frac{\partial g}{\partial \mathbf{X'}_n}(\hat{\mathbf{u}}, \mathbf{X'}_n) \mathbf{P}_{\mathbf{X'}_n} (\frac{\partial g}{\partial \mathbf{X'}_n}(\hat{\mathbf{u}}, \mathbf{X'}_n))^T$$

Results of the visual motion estimations and the corresponding error estimates are presented in figure 3.6, along a sequence of 90 low altitude aerial images. The mean variances computed on the 6 motion parameters, and their dispersion is summarised in table 3.1: $\mathbf{E}_V$ denotes the square root of variances of visual motion estimation errors along the sequence. the visual motion estimation measures a few meters translations with a few centimetres accuracy, and measures rotations with a precision of the order of $0.1°$, with a quite good regularity.

The motion estimation covariance $\mathbf{P}_{\hat{\mathbf{u}}}$ is the input covariance matrix $\mathbf{R}_u$ which is used in equation (3.12) to estimate the state variances during the filter prediction stage.

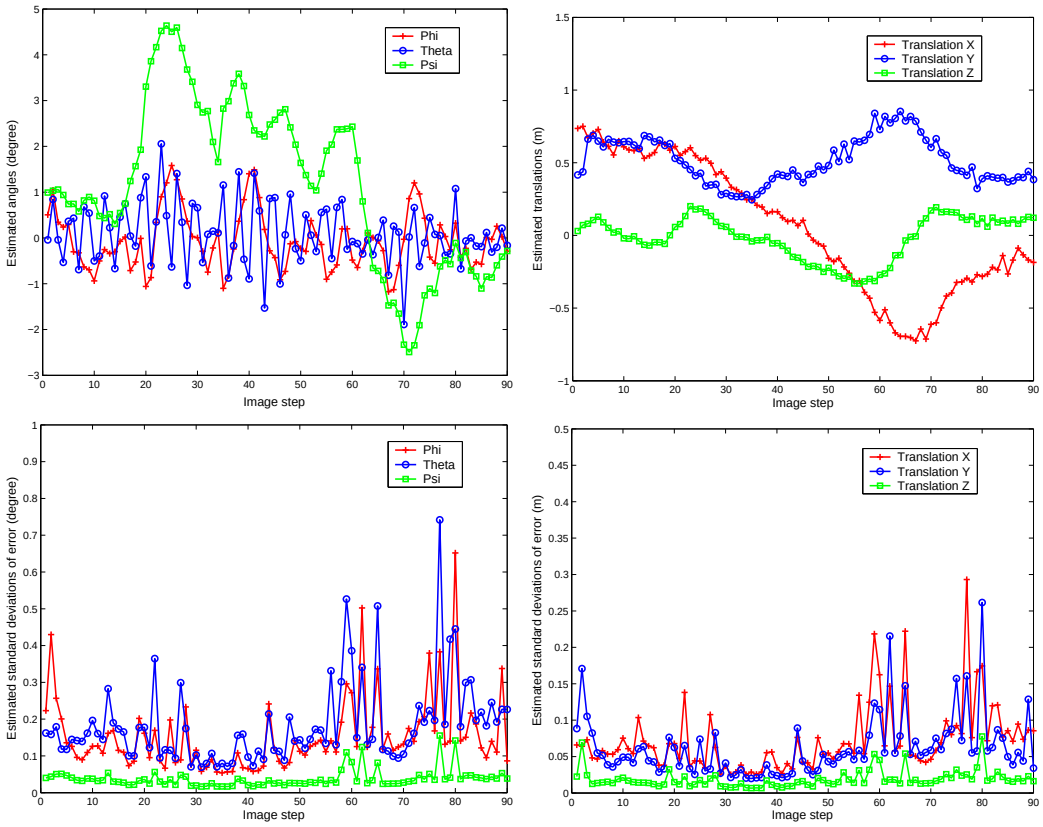


Figure 3.6: *The motion parameters computed by the visual motion estimation algorithm between consecutive frames along a 90 images sequence (top), and corresponding estimated errors (bottom). Note that there are no obvious correlation between the image angular and linear distance and the computed errors.*

3.5 Landmark selection

As explained in the overview of our approach (section 3.1.1), the 3D matches established after the interest point matching step are split into three groups: observed landmarks, non landmarks, and candidate landmarks. The observed-landmarks set is simply the points that corresponds to landmarks already in the state vector of the EKF. New candidate-landmarks should be cautiously added to the filter state, in order to avoid a rapid growth of its dimension and to obtain a regular landmark coverage of the perceived scenes. The candidate-landmarks selection procedure is made according to the following three criteria:

	Φ	Θ	Ψ	t_x	t_y	t_z
Average of $\mathbf{E}_V$	0.15^o	0.18^o	0.038^o	$0.072m$	$0.061m$	$0.0189m$
σ of $\mathbf{E}_V$	0.10^o	0.11^o	0.024^o	$0.046m$	$0.042m$	$0.0122m$

Table 3.1: *Statistics on the estimated errors of the visual motion estimations (VME) of figure 3.6.*

- **Observability.** Good landmarks should be observable in several consecutive frames: it guarantees that they are salient.
- **Stability.** The 3D coordinates of good landmarks must be precisely estimated by stereovision.
- **Representability.** Good landmarks must efficiently represent a 3D scene. The robot state estimation will be more stable if landmarks are regularly dispatched in the perceived scene, and this regularity will avoid a rapid growth of the EKF state vector size.

The number of candidate landmarks that are checked is determined on the basis of the number of new interest point matches (*i.e.* the ones that do not match with an already mapped landmark). This number is a percentage of the new interest points: we actually use 10%, as the visual motion estimation technique requires a lot of matches to yield a precise result.

The landmark selection is performed through the following steps:

1. New landmark candidates are first selected using the observability criterion. The observability of a landmark candidate is evaluated during several frames: when a set of points at time t are observable up to time $t + k$, the points in the set become new landmark candidates at time t. To guarantee high and uniform observability of landmarks, the candidates at time $t + 1$ should be at least observable at time $t + k + 1$. If there is not any candidate observable at time $t + k + 1$, new landmark candidates at time $t + 1$ don't exist.
2. The stability of the candidates that pass through the observability test are then checked. Their errors in their 3D position are memorised, and they are ranked according to the maximum of these errors.

3. Finally, the representability of the candidates is checked. The ranked candidate list is scanned, starting from the most precise one: every time a candidate is located at a distance from the already mapped landmarks and the minimum distance is greater than a given threshold, it is declared as a new landmark.

The observability criterion requires that candidates are evaluated through several successive frames: the EKF is therefore activated *a few image frames later* as the images are gathered.

3.6 Data association

Once a candidate landmark successfully passes the above selection criteria, it is incorporated in the state of the filter. In the following steps, if the landmark is still matched in the acquired images, it yields an observation of the system state. Once it is lost, it becomes an *old landmark*, *i.e.* a landmark that will be perceived again only when the robot travels again nearby.

The data association (re-perception of landmarks) is performed by the interest point matching algorithm, by matching interest points detected in the current image and the previously detected landmarks. For that purpose, the detected interest points for each acquired image must be memorised, and one must reason on the hypothetic visibility of the old landmark to select the image in which they will be matched with the current image.

3.6.1 Selection of the visible old landmarks

The visibility of an old landmark is evaluated by the distance of its projection from the centre of the current image considering the uncertainties on its position and on the current robot position estimate. The coordinates (u_i, v_i) in the image frame $I(k+1)$ of the ith old landmark are computed as follows:

$$(u_i, v_i) = \xi_i(\hat{\mathbf{x}}(k+1|k)) \tag{3.31}$$

where ξ_i is defined with the constant intrinsic matrix $\mathbf{A}$ of the camera and the observation matrix $\mathbf{h}_i$ in (3.15), and $\hat{\mathbf{x}}(k+1|k)$ is the predicted state vector in

(3.11). The associated covariance matrix is estimated by propagating the uncertainty on robot and landmark positions.

$$\mathbf{P}_i = \frac{\partial \xi_i}{\partial \hat{\mathbf{x}}(k+1|k)} \Lambda_i(k+1|k) \frac{\partial \xi_i^T}{\partial \hat{\mathbf{x}}(k+1|k)} \tag{3.32}$$

$$\Lambda_i(k+1|k) = \begin{pmatrix} \mathbf{P}_{pp}(k+1|k) & \mathbf{P}_{pi}(k+1|k) \\ \mathbf{P}_{pi}(k+1|k)^T & \mathbf{P}_{ii}(k+1|k) \end{pmatrix} \tag{3.33}$$

where $\mathbf{P}_{ii}$ is the covariance of ith landmark and $\mathbf{P}_{pi}(k+1|k)$ is the cross covariance between the robot and landmark positions.

Given the coordinates of the projection of an old landmark and its covariance matrix $\mathbf{P}_i$, if three times the covariance ellipsoid defined by $\mathbf{P}_i$ is within the current image, or if the intersection between the image and the ellipsoid exists, the landmark is considered as visible, and its visibility is defined as

$$\mathcal{V}_i = e^{-(u_i', v_i')\mathbf{P}_i(u_i', v_i')^T} \tag{3.34}$$

where $u_i',\ v_i'$ are the normalised coordinates of $u_i,\ v_i$.

$$(u_i', v_i') = \frac{(u_i - u_c, v_i - v_c)}{(u_m^2 + v_m^2)^{1/2}}$$

$(u_c,\ v_c)$ are the centre of image and $(u_m^2 + v_m^2)^{1/2}$, the maximum distance from the centre.

Two things have to be considered here:

- Old landmarks have been perceived in several past images: the past image that will be matched with the current one is the one for which the number of hypothetic visible landmarks is the maximum.

- It might happen that the current robot estimate is so erroneous that the selected image does not contain any landmark currently visible, after a long loop for instance. If such case occur, the matching algorithm is run on all the past images within which old landmarks are visible.

Scale change. When n old landmarks are visible in the current and a past image, the scale change s between the images is defined by the ratio of the two different image resolutions. This scale change can be approximately estimated by the ratio of the distance of the projections of the landmarks from the centre of image. But in fact, the scale change is not only a function of the depth change, but also of the view point change and the projective deformation: we only consider the robot position change along the optical axis of the camera at the current step to evaluate it, which defines two virtual image planes (a current and a past one).

The ith old landmark is projected onto the past image $I(k-t)$ on the coordinates $(u_i(k-t), v_i(k-t))$:

$$(u_i(k-t), v_i(k-t)) = \xi_i(\mathbf{x}(k-t)) \tag{3.35}$$

Its projection on the current virtual image $I'(k)$ defined by applying only a depth change is given by

$$(u_i(k), v_i(k)) = \xi_i(\mathbf{x}(k-t) + \Delta\mathbf{x}) \tag{3.36}$$

with

$$\Delta\mathbf{x} = [0, 0, 0, 0, 0, \Delta t_z(k, k-t), 0, \ldots] \tag{3.37}$$

where $\Delta t_z(k, k-t)$ is the variation of robot position along the optical axis of the camera between $I(k-t)$ and $I'(k)$, which can be defined by the estimated robot position at steps $k-t$ and k.

The scale change $s(k, k-t)$ between the two images is defined using the average ratio:

$$s(k, k-t) = \frac{1}{n} \sum_{i=1}^{n} \frac{\|(u_i(k), v_i(k))\|}{\|(u_i(k-t), v_i(k-t))\|} \tag{3.38}$$

As scale adaptive interest points are generated in the high-resolution image with a pre-defined scale change, if $s(k, k-t)$ is greater than 1, the points are generated in the current image $I(k)$, unless, they are recomputed in the past image $I(k-t)$ with $s(k-t, k)$.

The precise scale change estimation inherits from the error on the estimate $\Delta t_z(k, k-t)$ and on the 3D positions of old landmarks, but we have seen in the

previous chapter that points matches can be established with an approximation of scale change with up to half a unity error. Thus, the estimated scale change is quantised to one of the pre-defined scale changes among $\{1, 1.5, 2, 2.5, \ldots\}$. If the uncertainty in the scale change estimation (obtained with equation (3.32) that defines the uncertainties of the landmark projections) is greater than 0.5, several candidate scale changes close to the estimated change are applied and tested.

3.6.2 Observation errors

Landmark observation being based on interest point matching, matching errors on image plane is the first error source to consider to define the observation error. Two kind of errors can occur: wrong matches (outliers), and interest point location error. Outliers being rejected by the rejection algorithm presented in 3.4, only interest point location errors are considered to determine the matching error and the quantitative measure of the errors can be obtained by our matching uncertainty estimation algorithm in section 2.5.5.

Once the interest point matching error is defined, one must combine it with the errors on the corresponding 3D position estimates to define the observation error. The principle of this combination is illustrated in figure 3.7: the observation error is defined by the reprojection of the matching error in the 3D scene provided by stereovision.

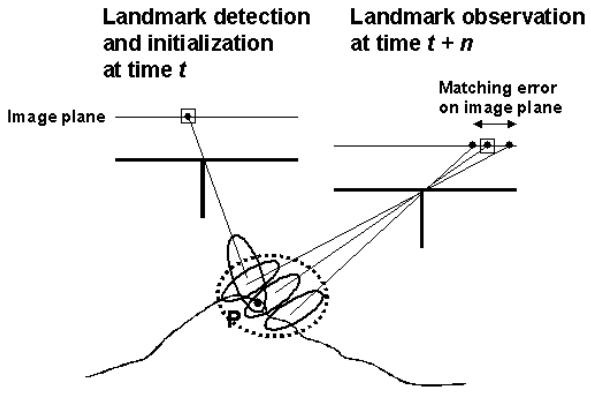


Figure 3.7: *Principle of the combination of the matching and stereovision errors. The points located in the square box are the projection of P on the image plane. Small ellipses indicate stereovision errors, the large dotted ellipsoid is the resulting observation error.*

Given a covariance matrix $\mathbf{P}_0$ associated with a match $\mathbf{x}_0$ on the image plane, the expected 3D coordinate of the match $\mathbf{X}_0$ is computed by stereovision and its associated variance, estimated by its neighbour 3D points of which the projection onto image plane is on the one sigma covariance ellipsoid given by

$$\mathbf{x}{\mathbf{P}_0}^{-1}\mathbf{x}^T = 1, \qquad \mathbf{x} = (u, v) \tag{3.39}$$

where $\mathbf{x}$ is the sub-pixellic projection of a 3D point $\mathbf{X}$. Equation 3.39 is the surface of equal probability density in 2 dimension. Since the 3D coordinates are only computed on integer pixels by stereovision, assuming the stereo error distribution is a zero mean normal one and the 3D surface variation locally linear, the variance of $\mathbf{X}_0$ is computed using its 8 closest neighbours $\mathbf{X}_k$, $k = 1, 2, , , 8$.

$$\boldsymbol{\sigma}^2_{\mathbf{X}_0} = \frac{1}{8}\sum_{k=1}^{8} w_k((\mathbf{X}_0 - \mathbf{X}_k)^2 + \boldsymbol{\sigma}_0^2 + \boldsymbol{\sigma}_k^2) \tag{3.40}$$

$$w_k = (\mathbf{x}_{k'} - \mathbf{x}_0)(\mathbf{x}_{k'} - \mathbf{x}_0)^T, \;\; (\mathbf{x}_{k'} - \mathbf{x}_0){\mathbf{P}_0}^{-1}(\mathbf{x}_{k'} - \mathbf{x}_0)^T = 1 \tag{3.41}$$

where $\boldsymbol{\sigma}_k^2$ is the variance of stereovision in (3.25) of neighbour 3D point $\mathbf{X}_k$ and $\mathbf{x}_{k'}$ is an intersection of the ellipsoid and the line defined by $\mathbf{x}_0$ and $\mathbf{x}_k$ which are the corresponding image coordinates of $\mathbf{X}_0$ and $\mathbf{X}_k$.

These 3D coordinates of matches and the associated variances are used in the equations (3.17) and (3.18).

3.7 Summary

The whole SLAM process based on stereovision, visual motion estimation, landmark selection and data association strategy, is performed according to the following sequence:

1. Given a stereo frame at time t, interest points are detected and their 3D coordinates are obtained by stereovision.

2. The detected interest points are matched with the points detected in the previous frame, outliers are removed by the algorithm presented in section 3.4.

3. The landmark selection procedure is performed, according to the selection criteria. Landmarks that pass the tests are incorporated in the filter state with a delay due to the observability evaluation (*i.e.* at time t, only landmark first detected at time $t - k$ are incorporated in the state).

4. The visual motion estimation is performed, without the observed or candidate landmarks to avoid a correlation between the prediction and the observation, and the prediction stage of the filter is completed.

5. The observations for recently mapped landmarks are defined by the matching results between consecutive frames. For old landmarks, the visibility of all mapped landmarks is checked and if visible old landmarks exist, they are observed by interest point matching between $I(t)$ and $I(t-n)$, applying the estimated scale change between the two images.

6. Incorporating the observations of landmarks, the state vector and covariance of the filter are updated.

Since the interest points matching algorithm requires not only the point coordinates and principal curvatures, but also some pixels in the correlation windows that surrounds them, all the images must be kept in memory to allow data association between a current image and an old one. The RAM of the CPU would be rapidly saturated after a few hundreds of images: therefore, the images are saved on a hard drive as the robot moves. The filter state is kept in live memory, as well as the 2D coordinates of the mapped landmarks in all the images where they have been perceived. When old landmarks are to be re-perceived, the image in which they have been perceived is reloaded in the memory to be matched with the current image.

Chapter 4

Experiments

Our developments have been tested with several thousands of stereovision image pairs in various contexts: aerial images taken at low altitudes with Karma, a robotised airship, outdoor unstructured terrain images taken with the Marsokhod rover Lama, and indoor images taken with the iRobot ATRV rover Dala (figure 4.1). We first present here some localisation and landmark mapping results obtained during various trajectories. Then, we introduce a way to re-estimate all the positions when the robot position is drastically refined by the SLAM after having closed a loop, which allows to build a spatially consistent digital elevation map.

Figure 4.1: *Karma, Lama and Dala: the three robots used to gather stereovision images pairs respectively from low altitudes, on a rough unstructured terrain, and indoor.*

4.1 Positioning errors

4.1.1 Low altitude aerial images

The digital cameras of the $2.2\ m$ wide stereo bench of the blimp Karma are $1/2''$ CCD sensors with 1024×768 pixels, and are equipped with a a $4.8\ mm$ focal length lens ($67° \times 53°$ field of view). The cameras have been calibrated at full resolution, and images are processed after being sub-sampled by a factor of two. The images used in the results below have been taken at altitudes ranging approximately from 20 to $35\ m$.

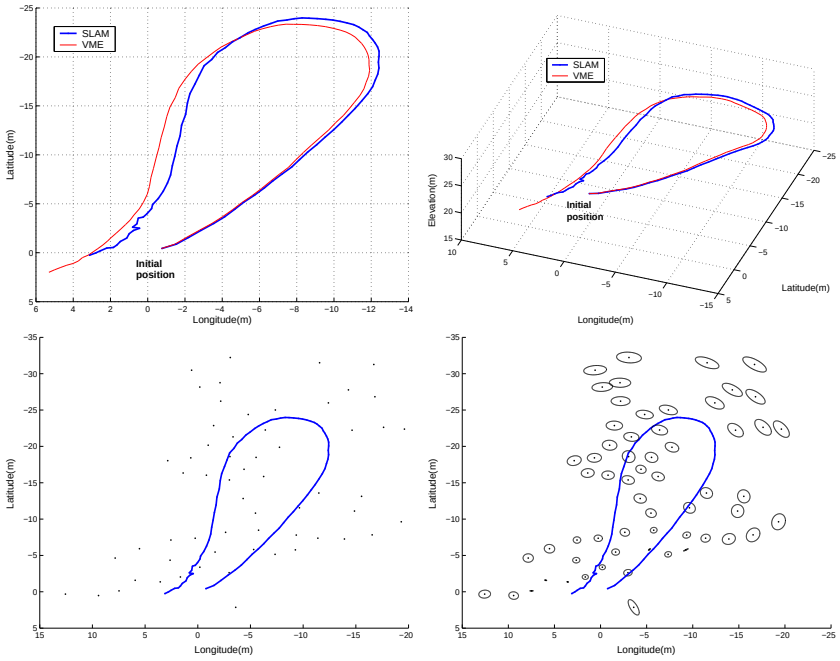


Figure 4.2: *A first result of our SLAM implementation with a sequence of 90 stereovision pair, taken at altitudes ranging from* 20 *to* $25\ m$*. Top image show the reconstructed trajectory in orthogonal projection and in 3D. Bottom images show the 60 landmarks mapped, with* 1σ *uncertainty ellipses (left: real scale, right: magnified by a factor of 40)*

The GPS on-board Karma is a differential code GPS with a 3σ accuracy of $2\ m$: it can not be used as a ground truth reference to validate the position estimates of the stereo bench. However, when Karma flies over an already perceived area (*i.e.* when it "closes a loop"), the visual motion estimate (VME) can provide an estimate of the relative positions between the first and last image of the se-

quence that overlaps. This reference is precise enough, as compared to the cumulation of errors induced with the visual motion estimation applied on consecutive frames.

Figure 4.2 presents a comparison of the reconstructed loop trajectory with a set of 90 images, while figure 4.3 shows the evolution of the standard deviation of the 6 position parameters of the stereo bench when applying the EKF. Two phases can be seen on this latter figure: until image 80, the standard deviation grows, however much more slowly than when propagating only the errors of the VME. A few landmarks detected in the beginning of the sequence are re-perceived in the following images: the standard deviations steeply decreases, and stabilises for the subsequent images where some ”old” landmarks are still observed.

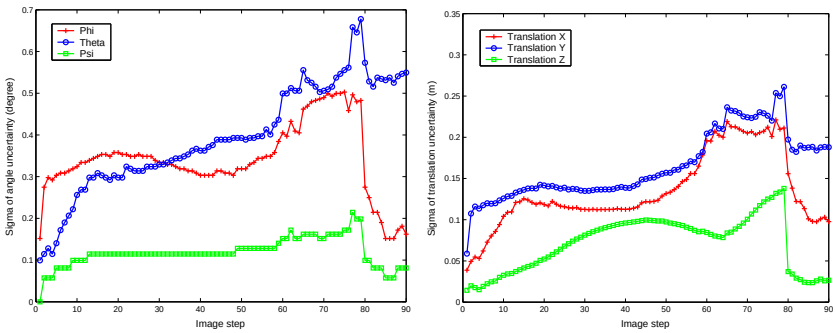


Figure 4.3: *Evolution of the standard deviations of the robot position parameters during the flight shown in figure 4.2.*

The quantitative figures summarised in table 4.1.1 are much more informative: they compare the results of the final position estimate with respect to the reference defined by the VME applied between images 1 and 90: the precision enhancement brought by the EKF is noticeable, and the absolute estimated errors are all bounded by twice the estimated standard deviations. The translation errors are below $0.40\ m$ in the three axes after an about $70\ m$ long trajectory, and angular errors are all below one degree.

Figure 4.4 shows an other 160 m long trajectory reconstructed with a set of 240 images.

On figure 4.5, one can note that the variances start a smooth decrease after the blimps makes a U-turn (around image number 170). The decrease is here much smoother than in the previous trajectory (figure 4.3), because the landmark

	Frame 1/90 Reference	Reference std. dev.	VME result	VME abs. error	SLAM result	SLAM std. dev.	SLAM abs. error
Φ	$6.19°$	$0.18°$	$11.93°$	$5.74°$	$6.01°$	$0.16°$	$0.18°$
Θ	$2.31°$	$0.66°$	$4.00°$	$1.69°$	$1.42°$	$0.55°$	$0.89°$
Ψ	$-105.94°$	$0.06°$	$-105.52°$	$0.41°$	$-106.03°$	$0.08°$	$0.09°$
t_x	$3.17\ m$	$0.26\ m$	$5.31\ m$	$2.14\ m$	$3.13\ m$	$0.09\ m$	$0.04\ m$
t_y	$0.61\ m$	$0.07\ m$	$2.01\ m$	$1.40\ m$	$0.26\ m$	$0.19\ m$	$0.35\ m$
t_z	$-1.52\ m$	$0.04\ m$	$-3.25\ m$	$1.73\ m$	$-1.51\ m$	$0.03\ m$	$0.01\ m$

Table 4.1: *Comparison of the errors made by the propagation of the visual motion estimation alone and with the SLAM EKF approach, using as a reference the VME applied between images 1 and 90*

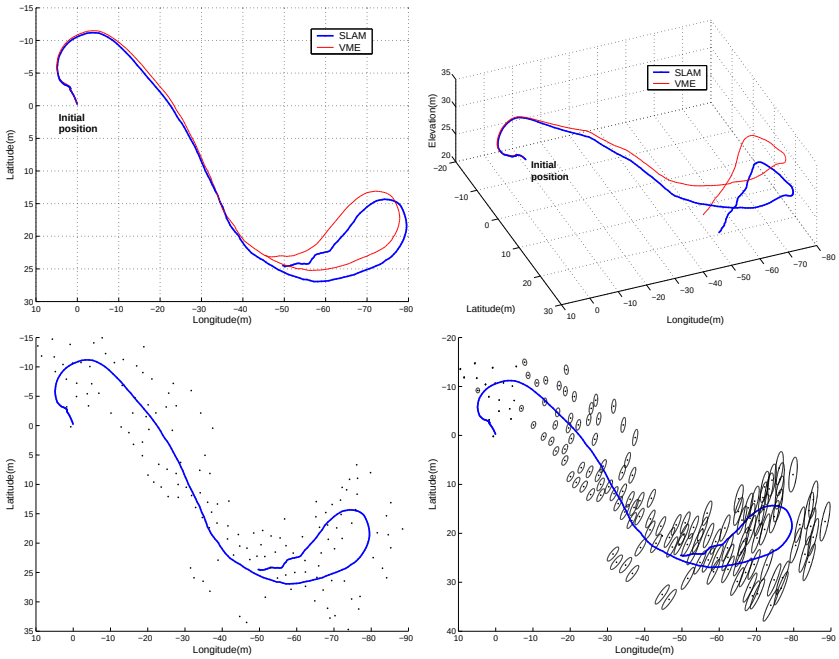


Figure 4.4: *An other trajectory reconstructed with a sequence of 240 images. 152 landmarks have been mapped (the magnification factor of the uncertainty ellipses in the bottom-right image is here 20)*

that are re-perceived were not originally perceived at the very beginning of the trajectory: their position uncertainty as they have been incorporated in the filter inherited from the robot pose uncertainty at that time, which was not very small. On the contrary, the decrease of the variances in figure 4.3 is very steep, because

the re-perceived landmarks have been incorporated in the filter state at the very beginning of the trajectory.

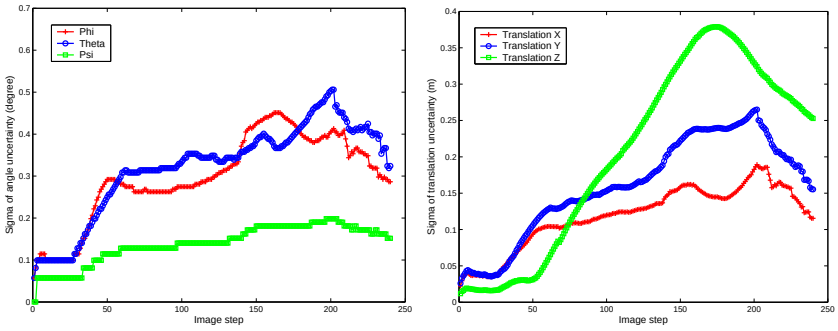


Figure 4.5: *Evolution of the standard deviations of the robot position parameters during the flight shown in figure 4.4.*

Finally, figure 4.6 show results integrating 400 images, the corresponding numerical results being presented in table 4.1.1. In this latter case, the variances start to decrease smoothly as the blimps makes the U-turn (around image number 200), and steeply decrease when the blimp flies again over landmarks perceived at the beginning of the trajectory (after image number 300). Figure 4.7 show some of the established matches during the trajectory.

	Frame 1/400 Reference	Reference std. dev.	VME result	VME abs. error	SLAM result	SLAM std. dev.	SLAM abs. error
Φ	$-0.12°$	$0.87°$	$-0.13°$	$0.01°$	$-3.68°$	$0.38°$	$3.56°$
Θ	$2.87°$	$1.14°$	$-4.99°$	$7.86°$	$5.54°$	$0.40°$	$1.64°$
Ψ	$105.44°$	$0.23°$	$101.82°$	$3.62°$	$104.32°$	$0.19°$	$1.12°$
t_x	$-4.93\ m$	$0.57\ m$	$5.45\ m$	$10.38\ m$	$-3.98\ m$	$0.21\ m$	$0.95\ m$
t_y	$0.14\ m$	$0.46\ m$	$3.04\ m$	$2.90\ m$	$-2.16\ m$	$0.22\ m$	$2.12\ m$
t_z	$3.89\ m$	$0.15\ m$	$19.81\ m$	$15.94\ m$	$3.46\ m$	$0.11\ m$	$0.43\ m$

Table 4.2: *Comparison of the errors made by the propagation of the visual motion estimation alone and with the SLAM EKF approach, using as a reference the VME applied between images 1 and 400*

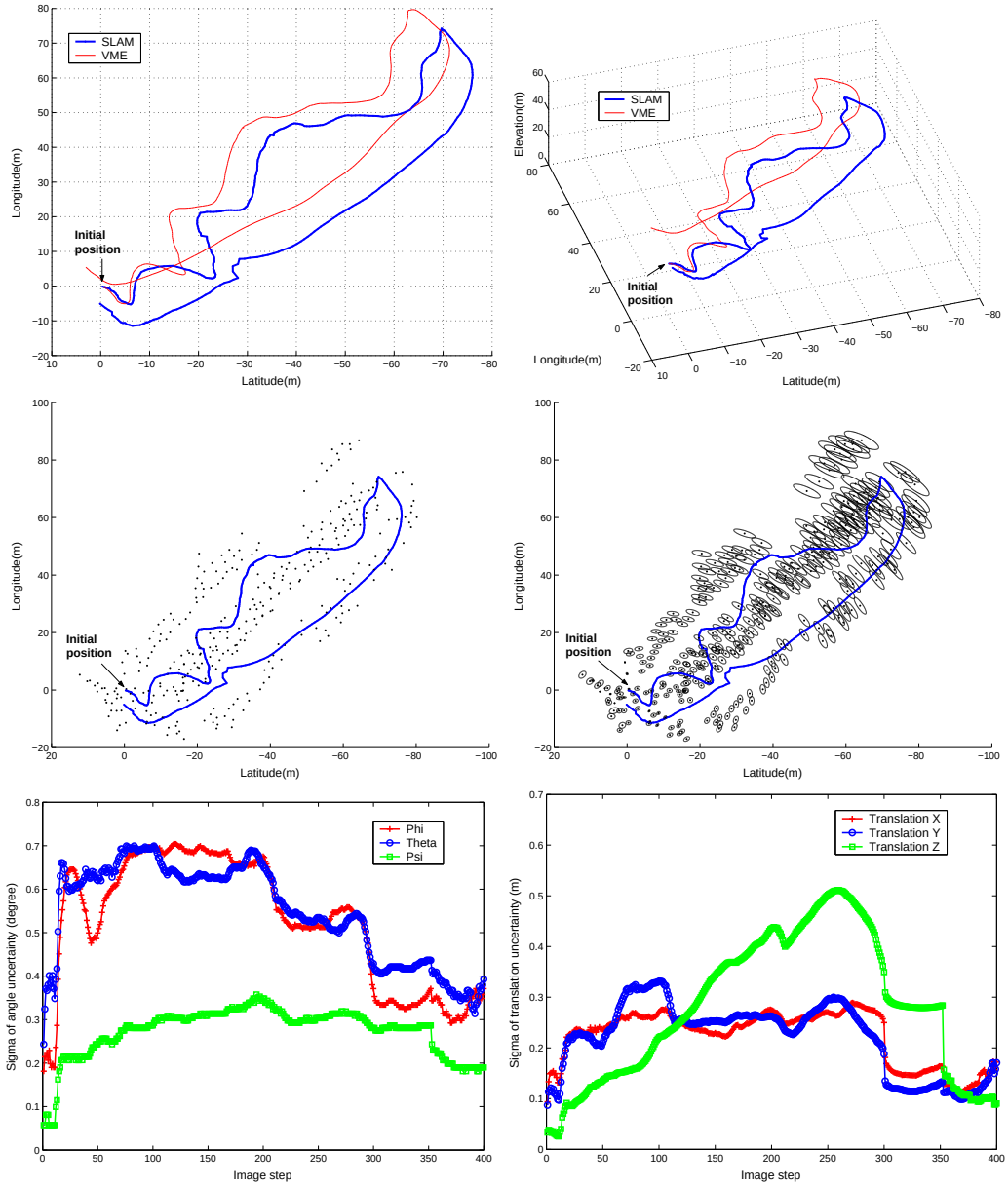


Figure 4.6: *An other result of SLAM implementation with the blimp, with a sequence of 400 stereovision pair. the reconstructed trajectory is about* $270m$ *long. Top images show the reconstructed trajectory in orthogonal projection and in 3D. Middle images show the 355 landmarks mapped, with* 1σ *uncertainty ellipses (left: real scale, right: magnified by a factor of 40), and bottom images show the evolution of the standard deviations of the blimp position parameters.*

4.1.2 Outdoor environment ground images

Figure 4.8 show a positioning result with a set of 111 images taken with the rover Lama, while doing a round shape trajectory on a rough terrain. The 0.4 wide stere-

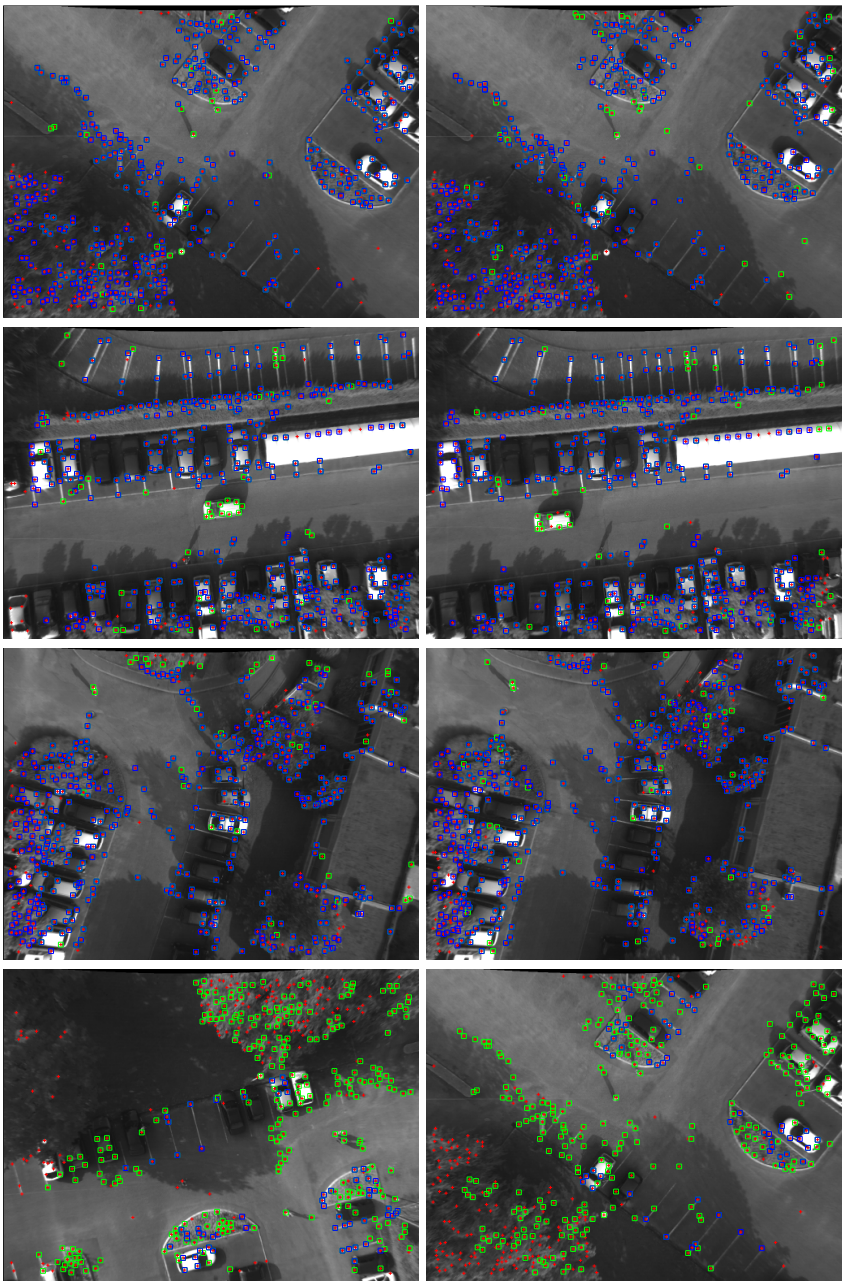


Figure 4.7: *Interest points matched between some consecutive images of the sequence of the trajectory shown in figure 4.6 - the blue squares are the matched point. Each line shows two consecutive images, except the last one that shows the matches established between the image number 390 at the end of the trajectory and the first image (landmark re-perception).*

ovision bench was oriented sidewards, towards the centre of the loop. Quantitative result of the positioning error of the cumulated visual motion estimation and the SLAM approach are shown on table 4.1.4 (as with the aerial images, the reference is obtained by applying the VME between the first and last images of the loop). The SLAM result is astonishingly precise after a $50\,m$ long trajectory on rough terrain. Note also that the cumulated error of VME is much smaller than for the blimp trajectories: the reason why the results are so precise is that the observed points and landmarks are much closer to the stereovision bench than for the blimp: remember that the precision of stereovision is proportional to baseline length and quadratically inversely proportional to the depth: the baseline/depth ratio is much more favourable for Lama than for the blimp.

	Frame 1/111 Reference	Reference std. dev.	VME result	VME abs. error	SLAM result	SLAM std. dev.	SLAM abs. error
Φ	$0.52°$	$0.31°$	$2.75°$	$2.23°$	$0.88°$	$0.98°$	$0.36°$
Θ	$0.36°$	$0.25°$	$-0.11°$	$0.47°$	$0.72°$	$0.74°$	$0.36°$
Ψ	$-0.14°$	$0.16°$	$1.89°$	$2.03°$	$1.24°$	$1.84°$	$1.38°$
t_x	$-0.012\,m$	$0.010\,m$	$0.057\,m$	$0.069\,m$	$-0.077\,m$	$0.069\,m$	$0.065\,m$
$‘t_y$	$-0.243\,m$	$0.019\,m$	$-1.018\,m$	$0.775\,m$	$-0.284\,m$	$0.064\,m$	$0.041\,m$
t_z	$0.019\,m$	$0.015\,m$	$0.144\,m$	$0.125\,m$	$0.018\,m$	$0.019\,m$	$0.001\,m$

Table 4.3: *Comparison of the errors made by the propagation of the visual motion estimation alone and with the SLAM EKF approach for the trajectory of Lama, using as a reference the VME applied between images 1 and 111 (the first and the last one).*

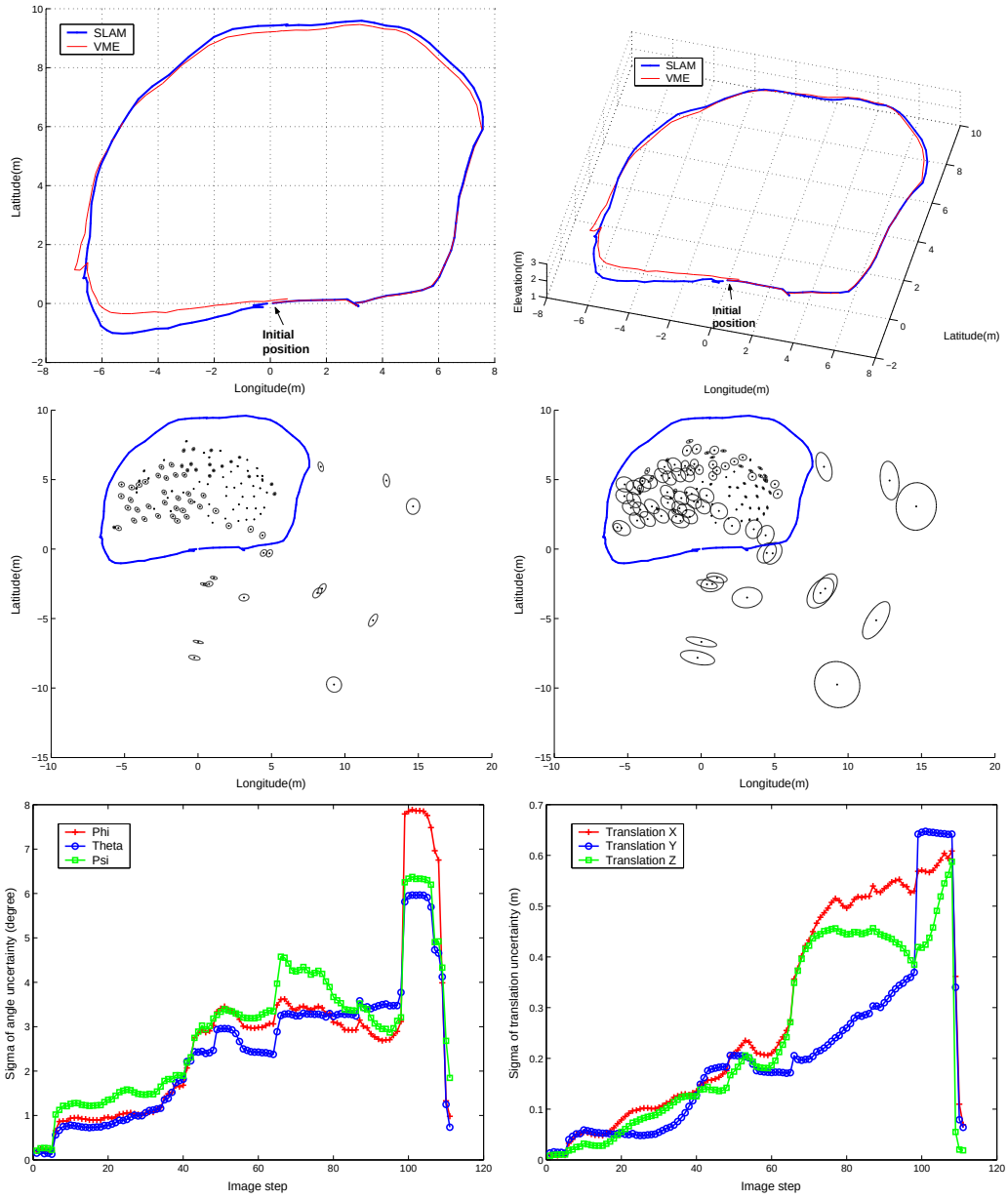


Figure 4.8: *A result of our SLAM implementation with Lama, with a sequence of 110 stereovision pairs. Top images show the reconstructed robot trajectory in orthogonal projection and in 3D. Middle images show the 118 landmarks mapped, with* 1σ *uncertainty ellipses (left: real scale, right: magnified by a factor of 5), and bottom images show the evolution of the standard deviations of the robot position parameters.*

4.1.3 Indoor environment images

We tested our algorithms with the rover Dala, equipped with a $0.35m$ stereovision bench, on trajectories made indoor. Figure 4.10 show the positioning results after

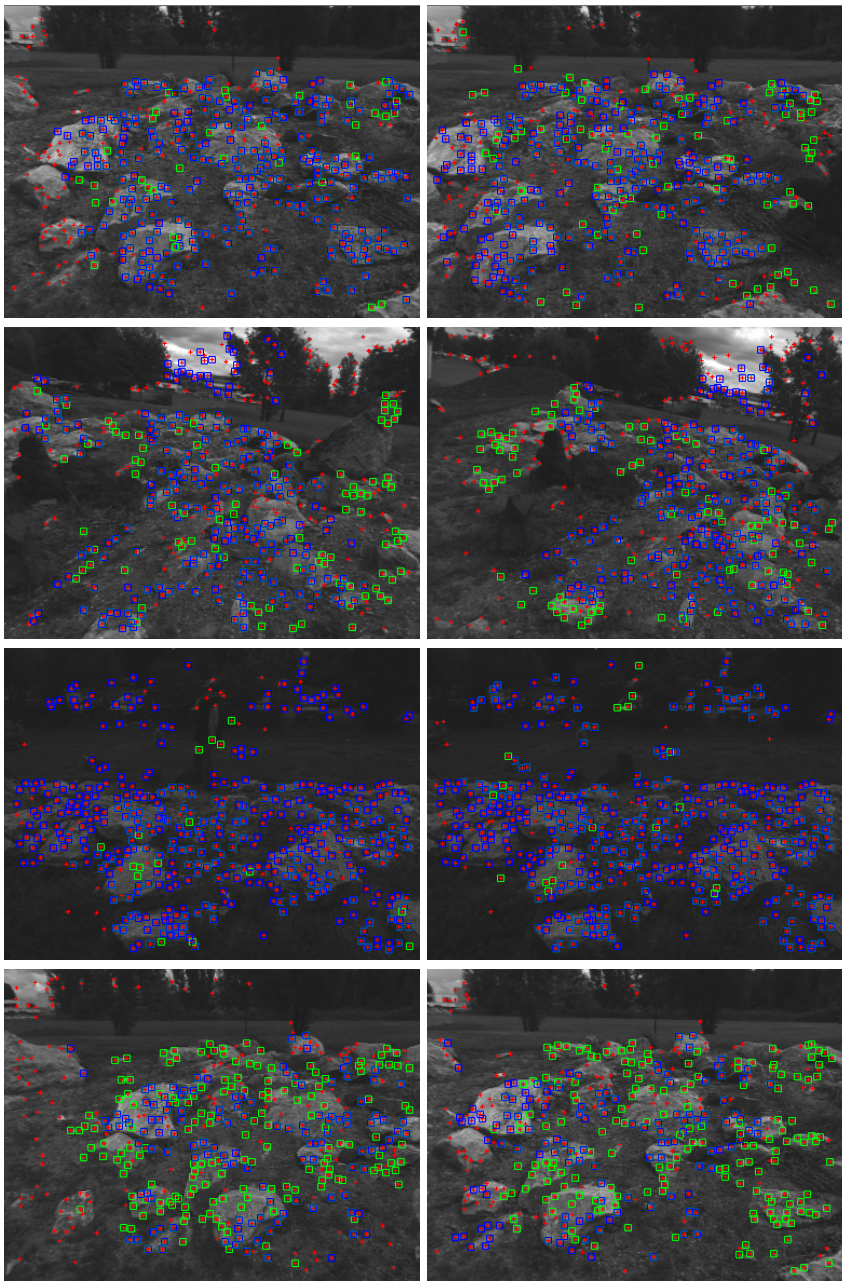


Figure 4.9: *Interest points matched between some consecutive images of the sequence of the trajectory made by the rover Lama in figure 4.8. Each line shows two consecutive images, except the last one that shows the matches established between the image number 111 at the end of the trajectory and the first image (landmark re-perception).*

three consecutive circle loops (the speed command was constant during all the acquisitions), the stereovision bench being oriented sidewards, towards the exterior of the circle. The position correction after the first loop (just after old landmarks a re-perceived for the first time) is noticeable. After that, the robot position is reliably estimated. Note also that the position resulting from the VME only is slowly drifting (and going up !).

The interest of this trajectory is that the robot evolves on a perfect plane, while 6 parameters of displacement are continuously updated by the SLAM algorithm: the reference for the two roll and pitch angles and the elevation is therefore simply 0 (we did not constrained the robot position to lie on a plane on purpose, in order to have this perfect reference). The three curves on the figure 4.11 show the estimation of these variables: they are noticeably stabilised after the first loop.

Finally figure 4.12 show the evolution of some landmark positions during the trajectory, and figure 4.13 show some of the points matches established during the trajectory.

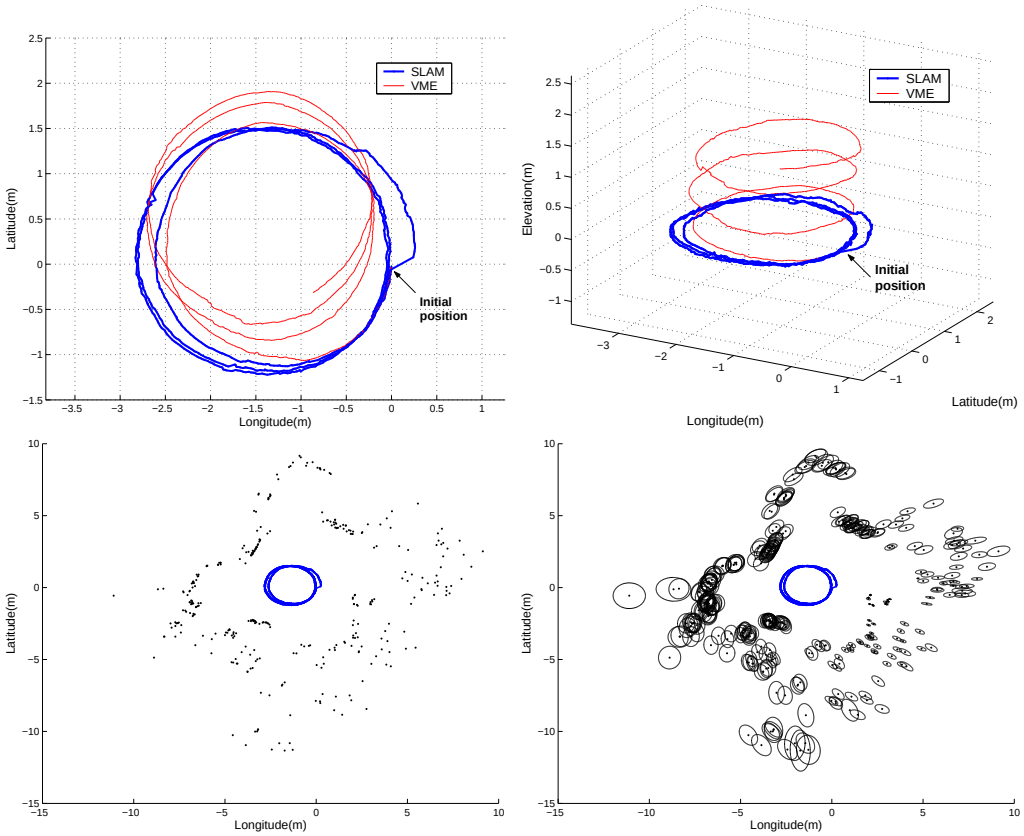


Figure 4.10: *A trajectory reconstructed with a sequence of 1000 images taken by Dala looking to the left side while turning right (25.6m long trajectory). 338 landmarks have been mapped during the whole trajectory (269 during the first turn, 31 during the second and 38 during the third). The uncertainty ellipses of the landmarks in the down-right image are magnified by a factor 10.*

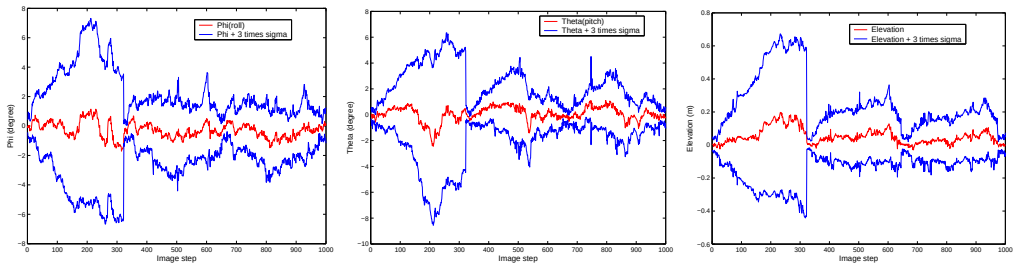


Figure 4.11: *Evolution of the three position variables Φ, θ and z displayed with the corresponding standard deviations estimates.*

Figure 4.14 show a last result obtained with a similar circle trajectory in the same environment (4 loops here), during which the stereovision bench was fac-

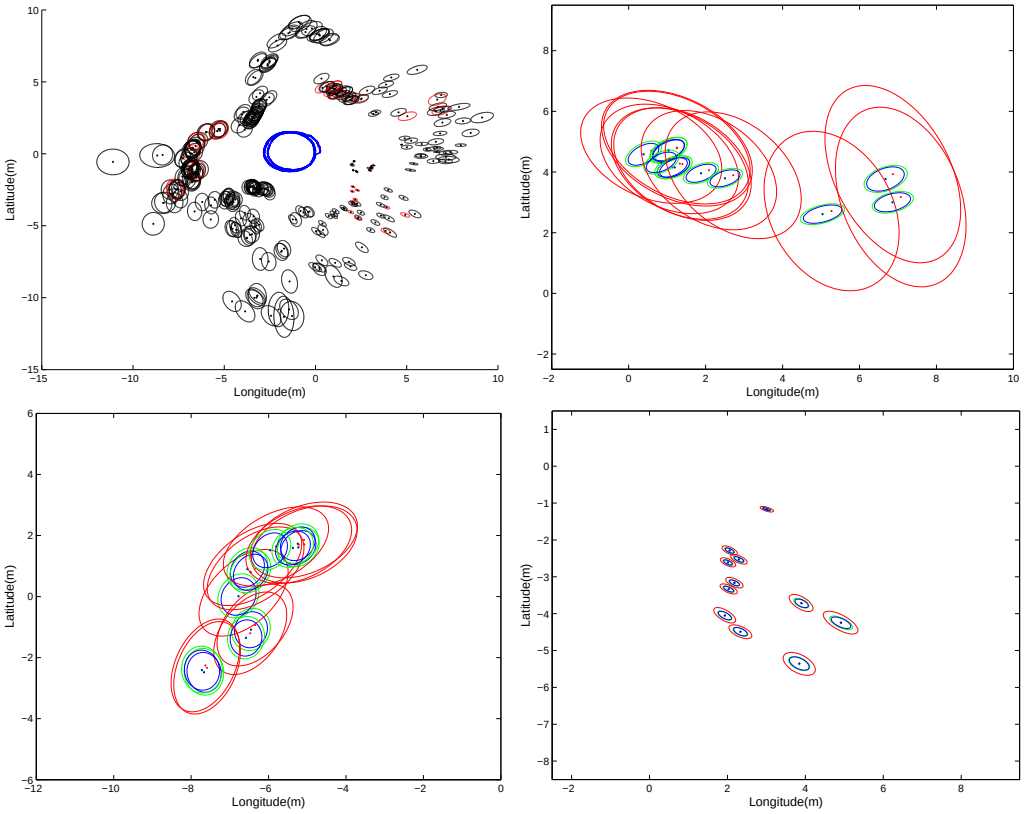


Figure 4.12: *Evolution of the position of three sets of ten landmarks (highlighted in red in the top-left figure that shows all the landmarks). The ellipses size is 10 times the standard deviation: the red ones are the landmarks position estimations at the end of the first loop, the green ones at the end of the second loop, and the blue ones at the end of the third loop.*

ing forward, and figure 4.15 show the evolution of the three position variables that should be stabilised around 0. It is interesting to compare these results with the ones before: first, the cameras being oriented forward and not in the exterior direction of the travelled circle (the walls of the room), more landmarks are perceived, at a bigger distance from the robot. Second, the position estimates by both VME and SLAM are much more noisy (in particular the circle is not perfectly recovered), whereas the same constant speed command was given to the robot. This tends to indicate that looking sidewards give better results than looking forwards, which seems reasonable, when one knows that the errors in stereovision are essentially on the depth estimates. Finally, let's note that on this trajectory, the positions are not stabilised during the second loop, which is probably due to a

Figure 4.13: *Interest points matched between some consecutive images of the sequence of the trajectory shown in figure 4.10. Each line shows two consecutive images, except the last one that shows the matches established between the image number 324 (end of the first loop) and the first image.*

wrong landmark association. However, the large number of landmarks present in the scene prevented the filter from a divergence.

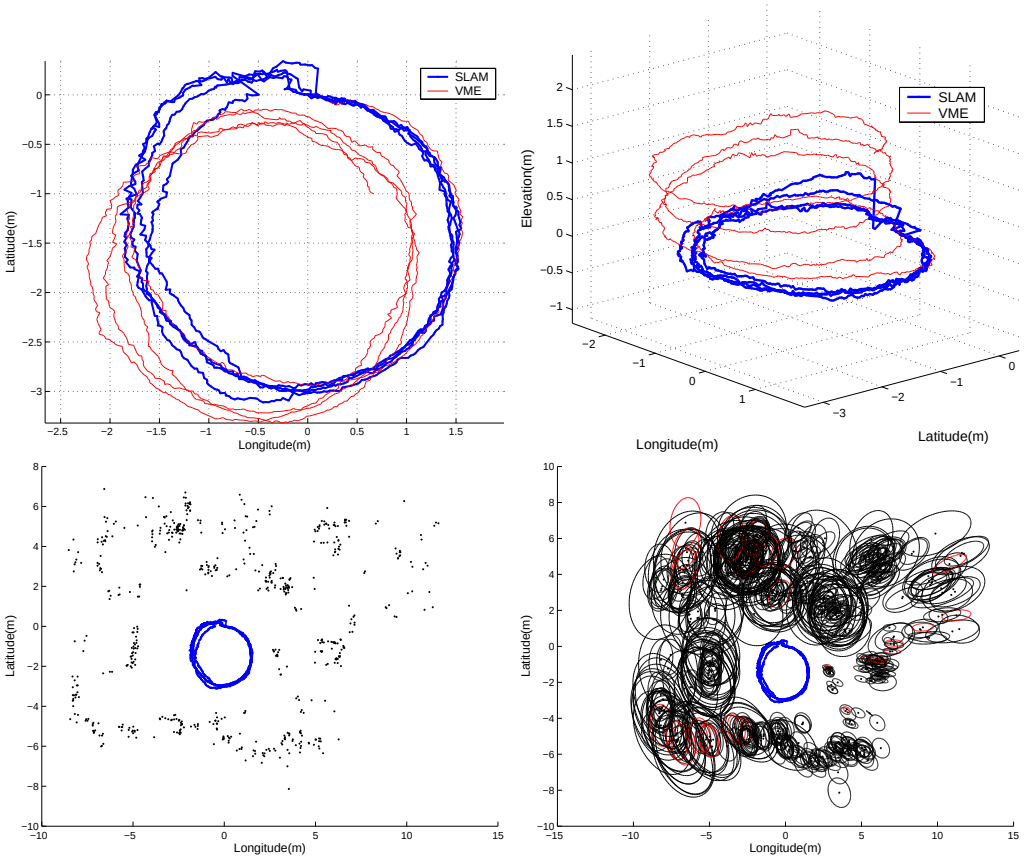


Figure 4.14: *Another trajectory reconstructed with a sequence of 1300 images taken by Dala, with the stereovision bench facing forward (47.3m long trajectory). 628 landmarks have been mapped during 4 loops.*

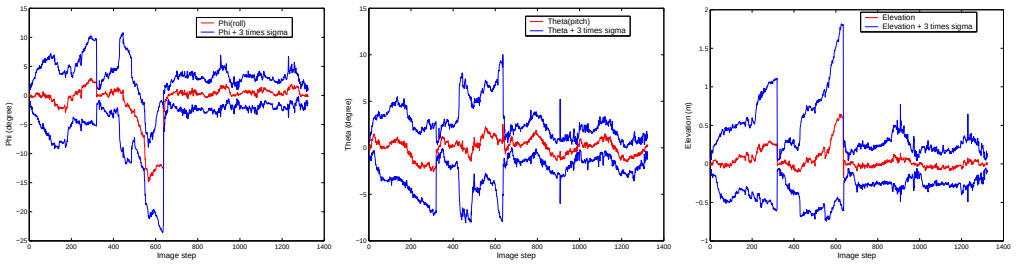


Figure 4.15: *Evolution of ϕ, θ and z during the trajectory of figure 4.14.*

4.1.4 Time performance

The time performance of our SLAM implementation depends on the size of image, interest point detection and matching time and the number of landmarks. Given a stereo frame(384×512) at time T, the computation times of each stages in our SLAM process are measured on sparc SUN Workstation fire v240.

- Pre-processing (distortion correction and rectification): 300ms (constant)
- Interest point detection: 200ms (average)
- Interest point matching (matching for sparse stereo and between two consecutive frames): 100ms (average)
- Extended Kalman Filter(prediction, update and new landmark detection): depending on the number of landmarks

Landmarks	100	200	300	400	...
$Computationtime(average)$	$20ms$	$50ms$	$100ms$	$220ms$	...

- Old landmarks observation (matching between T and T-n): 200ms (average)

Totally, in case there is no visible old landmarks at current step, the computation time is less than $700ms$ keeping up to 300 landmarks. If some old landmarks are visible, the computation become more expensive($700ms + 200ms$) for old landmark observation) to re-observe them.

4.2 Digital Elevation Maps

Thanks to the precise positioning estimation, the processed stereovision images can be fused after every update of the EKF into a *digital elevation map* (DEM), that describes the environment as a function $z = f(x, y)$, determined on every cell (x_i, y_i) of a regular Cartesian grid.

Our algorithm to build a DEM simply computes the elevation of each cell by averaging the elevations of the 3D points obtained by dense stereovision that are vertically projected on the elementary surface it defines. The standard deviation on the cell elevation is also straightforwardly computed, and since a luminance value is associated to each 3D point produced by stereovision, it is also possible to compute a mean luminance value for each map cell.

Figure 4.16 shows the digital elevation built from the 240 images during the trajectory shown in figure 4.4: the resolution of the grid is here $0.1\ m$, and no map discrepancies can be detected in the corresponding ortho-image, which is the orthogonal projection of the luminance informations encoded in the DEM grid.

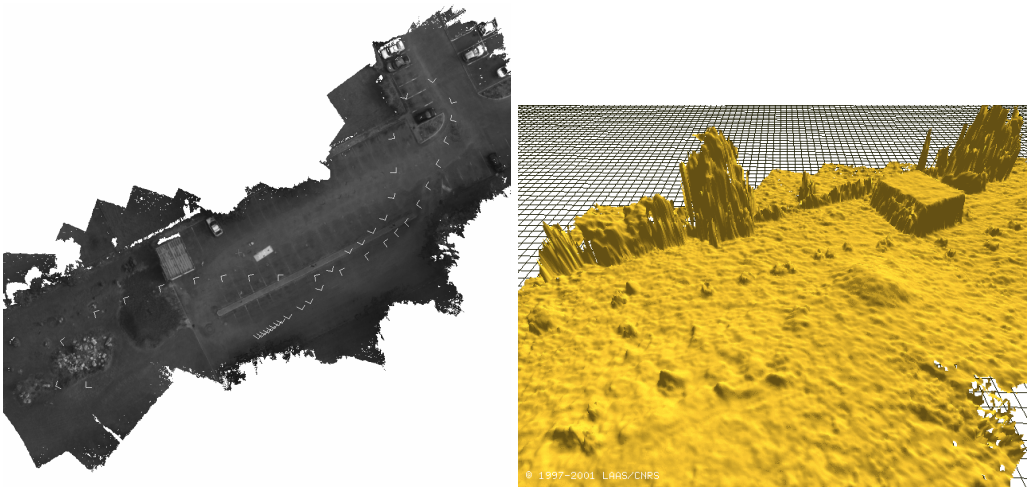


Figure 4.16: *The DEM computed with 240 images, positioned according to the trajectory of figure 4.4: ortho-image and 3D view of the bottom-left area. The map covers an area of about* $3500\ m^2$.

When closing a loop after having travelled over a long distance, a big discontinuity in the robot trajectory is likely to occur, as we saw in the positioning results above. Therefore, to construct a spatially coherent environment map, the trajec-

tory of the robot should be corrected (in other words, one need to re-estimate all the past positions of the robot on the basis of the correction that occurred when old landmarks are re-perceived).

Under the assumption that the local motion estimation error between two consecutive frames is small enough to be negligible, the trajectory can be estimated by a global optimisation method with all observations of landmarks and local motion estimates gathered during navigation, as in [Deans 01] for instance. But this kind of batch process limited by the number of observations and motion steps. An optimal pose estimation approach from a network of relations have been proposed [Lu 97, Gutmann 99], which consistently maintains the history of robot poses and does not use environment features: all the local frames of data as well as the relative spatial relationships between local frames are maintained. Even though the approach is computationally less expensive than the method using all observations of landmark and local motion estimates, since all the history of the robot position are maintained, it is difficult to apply over long ranges, in large environments. Moreover, the nonlinear complex observation model of local motion (6 parameters) requires an iterative nonlinear optimisation, whose computation becomes expensive as the number of considered positions increases.

We propose here a fast backward correction method which locally maintains the observation of landmarks and a local motion estimate per step, using the constructed landmarks map. This local optimisation method proceeds from the most accurate estimate of robot's pose. As long as errors in the estimate exist, even very small, the method is suboptimal because the errors influences the local optimisation in subsequent steps. Using the robot state at current step k, the state at step $k-1$ minimises the cost function with n observation given by

$$\begin{aligned} E(\mathbf{x}_p(k-1)) &= \frac{1}{2}(\mathbf{x}_p(k) - \hat{\mathbf{x}}_p(k))^T \mathbf{R}_u^{-1} (\mathbf{x}_p(k) - \hat{\mathbf{x}}_p(k)) + \\ & \frac{1}{2n} \sum_i^n (\mathbf{z}_i(k-1) - \hat{\mathbf{z}}_i(k-1))^T \mathbf{R}_i^{-1} (\mathbf{z}_i(k-1) - \hat{\mathbf{z}}_i(k-1)) \end{aligned} \tag{4.1}$$

where $\mathbf{x}_p$ is the vector of 6 position parameters, $\mathbf{u}$ the visual motion estimation result (control input), $\mathbf{z}_i$ the observation, $\hat{\mathbf{z}}_i$ the estimated observation by $\mathbf{x}_p$, and $\mathbf{R}_u, \mathbf{R}_i$ are respectively the covariances of $\mathbf{u}, \mathbf{z}_i$ (see section 3.2.2), and the estimated 6 parameters vector $\hat{\mathbf{x}}_p(k)$ is defined using the VME result and the robot

state at the previous step.

$$\hat{\mathbf{x}}_p(k) = \mathbf{f}(\mathbf{x}_p(k-1), \mathbf{u}(k))$$

At first glance, it is easily shown that the first term of (4.2) is constrained by prediction and the second, by observations of landmarks. Using the minimisation result $\mathbf{x}_p(k-1)$, the robot state at step $k-2$ is estimated by the same method above. In this suboptimal method, even though the position error at current step is propagated to all past steps, since the cost function minimises observation and positioning error using built landmark map and local motion estimations between two consecutive frames at the same time, if the landmark map is correct, the minimisation of observation error at each position guarantees a bounded error growth during backward correction.

Figure 4.17 shows the comparison of the result of DEM building with and without backward correction of the trajectory of figure 4.6: in the presence of a significant discontinuity in the trajectory of robot, a coherent DEM can be built with the trajectory corrected by backward correction.

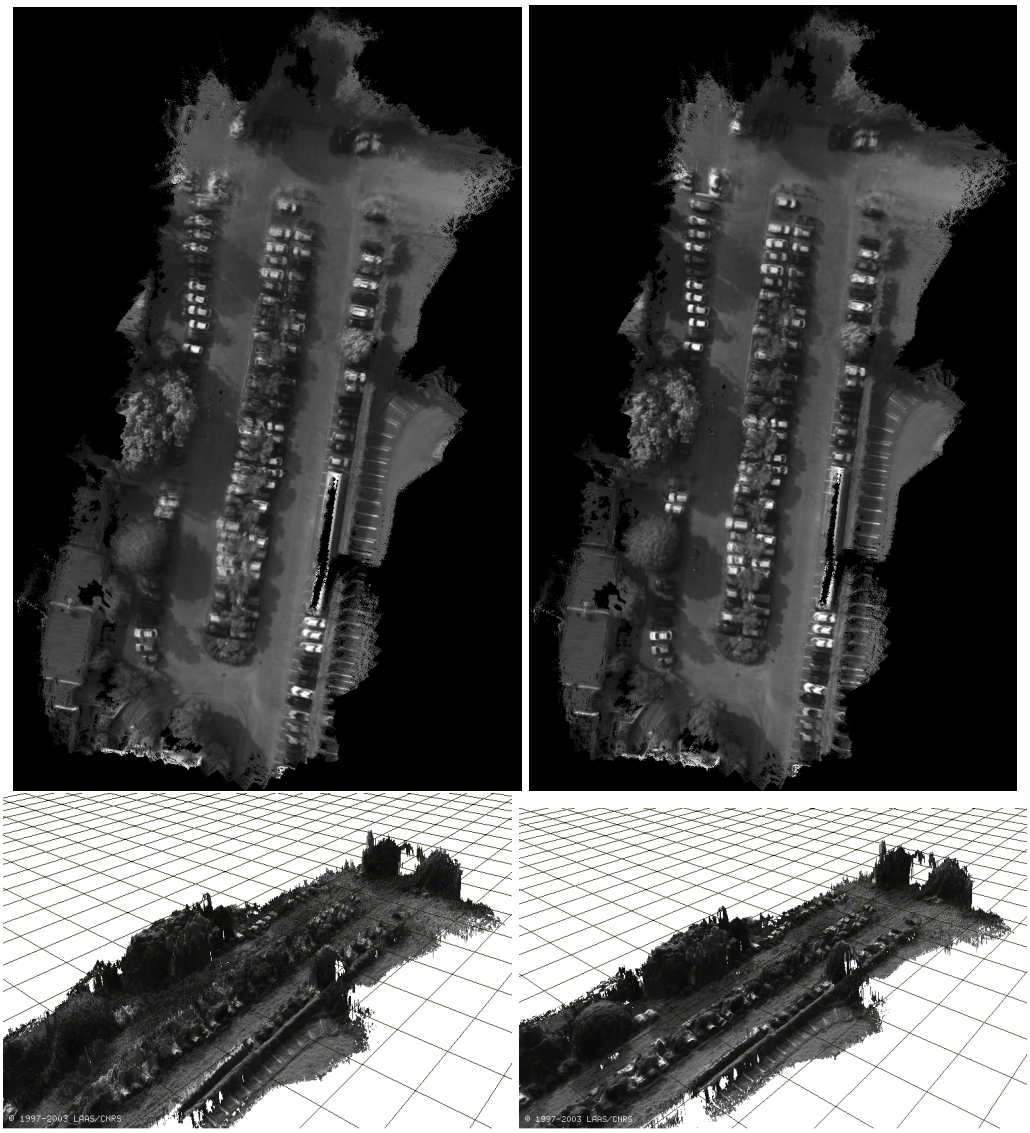


Figure 4.17: *Left: the DEM computed with 400 images, positioned according to the trajectory recovered by SLAM in figure 4.6. Right: DEM computed on the basis of the corrected trajectory, after applying the backward correction. The difference is noticeable on top of the ortho-images (upper images), which corresponds to the areas perceived at the start and end of the trajectory. The map covers an area of about* $6000\ m^2$.

Chapter 5

Discussion

We have presented a solution to the SLAM problem that exclusively uses stereovision. The main advantage of our approach relies on the use of interest points as landmarks. Such points are indeed very numerous in any kind of environment: no ”obvious” landmarks are required for the algorithms to operate successfully, and this allows an active selection of the landmarks to map. The reliability of our interest point matching algorithm is of course a key point here, as it allows robust data associations. Also, the use of the visual estimation technique as a mean to achieve the prediction stage of the filter is very efficient: its estimates are precise enough to yield a fast convergence, keeping the filter linearisations as fair approximations. Finally, a thorough study and identification of the various errors estimates involved in the filter allows to set it up properly, without any empirical tuning stage, which would have been very tedious.

The results presented show that thanks to the application of an EKF on the *sole* basis of stereovision, it is possible to achieve a precise positioning of a robot after navigation over long distances in various kinds of 3D environments. Our approach is especially effective with aerial images, where the visual motion estimation provides an estimate at least as precise than the off-the-shelf cheap positioning sensors (namely GPS and low-cost inertial sensors), and the application of the Kalman filter could reduce the position errors down to a few tens of centimetres after several hundreds meters navigation. Of course, we do not mean that one should only use stereovision to tackle the SLAM problem: any additional sensor is of course welcomed to solve the complex localisation problem, and our

algorithms for the prediction stage (VME) or the landmark mapping could easily be integrated with any other positioning sensor in a Kalman filter framework. Our point is to prove that stereovision is an interesting sensor to consider for the SLAM problem. And since it is a sensor also required by other functionalities (*e.g* obstacle detection, people tracking, or object and place recognition) in many application contexts, one should also benefit from its possibility to localise the robot and to map discriminant landmarks.

Still, several improvements should be made to have a stereovision based SLAM solution running on-board a robot. Besides the consideration of SLAM optimisation techniques such as the consideration of sub-maps or fastSlam approaches (numerous landmarks are required by our algorithm to operate correctly - and the update stage of the Kalman filter with 400 landmarks requires around 1.0s of a powerful CPU), the necessity to keep in memory the whole set of the perceived images is a big drawback. The number of images could be reduced by a factor of several unities, by keeping only "key images" in which numerous landmarks are visible. Finally, stereovision is intrinsically limited by the baseline/depth ratio: the 3D informations provided by a configuration in which this ratio is smaller than about $1/30^{th}$ almost do not make sense. This is not a big problem for ground robots, that hardly need to perceive useful informations further than a few tens of meter, but it is a strong limitation for aerial robots: for that purpose, the consideration of SLAM with monocular images sequences is necessary, which is a very interesting and challenging problem.

Bibliography

[Anandan 89] P. Anandan. *A computational framework and an algorithm for the measurement of visual motion*. International Journal of Computer Vision, vol. 2, no. 3, pages 283–310, 1989.

[Borges 02] G.A. Borges & M-J. Aldon. *Optimal mobile robot pose estimation using geometrical maps*. IEEE Transactions on Robotics and Automation, vol. 18, no. 1, pages 87–94, 2002.

[Bosse 02] M. Bosse, P. Noewman, J. Leonard & S. Teller. *An atlas framework for scalable mapping*. In MIT Marine Robotics Laboratory Technical memorandum, 2002.

[Chatila 85] R. Chatila & J-P. Laumond. *Position Referencing and Consistent World Modeling for Mobile Robots*. In IEEE International Conference on Robotics and Automation, St Louis (USA), pages 138–145, 1985.

[Chaudhuri 96] S. Chaudhuri, S. Sharma & S. Chatterjee. *Recursive estimation of motion parameters*. Computer Vision and Image Understanding, vol. 64, no. 3, pages 434–442, November 1996.

[Chong 99] K. Chong & L. Kleeman. *Feature based mapping in real, large scale environments using an ultrasonic array*. International Journal of Robotics Research, vol. 18, no. 1, pages 3–19, 1999.

[Daugman 85] J.G. Daugman. *Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters*. Journal of the Optical Society of America, vol. A, no. 2, pages 1160–1169, 1985.

[Davison 03] A. J. Davison. *Real-Time Simultaneous Localisation and Mapping with a Single Camera*. In 9th International Conference on Computer Vision, Nice (France), pages 1403–1410, October 2003.

[Deans 01] M. Deans & M. Herbert. *Experimental Comparison of Techniques for Localization and Mapping using a Bearings Only Sensor*. In S. Singh, editeur, Experimental Robotics VII, Lecture Notes in Computer Science. Springer-Verlag, 2001.

[Dellaert 99] Frank Dellaert, Dieter Fox, Wolfram Burgard & Sebastian Thrun. *Monte Carlo Localization for Mobile Robots*. In IEEE International Conference on Robotics and Automation (ICRA99), May 1999.

[Dissanayake 01] G. Dissanayake, P. M. Newman, H-F. Durrant-Whyte, S. Clark & M. Csorba. *A solution to the simultaneous localization and map building (SLAM) problem*. IEEE Transaction on Robotic and Automation, vol. 17, no. 3, pages 229–241, May 2001.

[Dufournaud 00] Y. Dufournaud, C. Schmid & R. Horaud. *Matching Images with Different Resolutions*. In International Conference on Computer Vision and Pattern Recognition, Hilton Head Island, SC (USA), pages 612–618, June 2000.

[Faugeras 93] O. Faugeras, T. Vieville, E. Theron, J. Vuillemin, B. Hotz, Z. Zhang, L. Moll, P. Bertin, H. Mathieu, P. Fua, G. Berry & C. Proy. *Real-time correlation-based stereo : algo-*

rithm, implementations and application. Rapport technique RR 2013, INRIA, August 1993.

[Freeman 91] W. T. Freeman & E. H. Adelson. *The design and use of steerable filters*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no. 9, pages 891–906, 1991.

[Förstner 87] W. Förstner & E. Gülch. *A fast operator for detection and precise location of distinct points, corners and centres of circular features*. In In ISPRS Intercommission Workshop, Interlaken, pages 149–155, June 1987.

[Förstner 94] W. Förstner. *A framework for low level feature extraction*. In European Conference on Computer Vision, Stockholm (Sweden), pages 383,394, 1994.

[Garcia 01] R. Garcia, X. Cufi & M. Carreras. *Estimating the motion of an underwater robot from a monocular image sequence*. In International Conference on Intelligent Robots and Systems (IROS), Hawaii, pages 1682–1687, October-November 2001.

[Guivant 01] J. Guivant & E. Nebot. *Optimization of the Simultaneous Localization and map Building Algorithm for Real Time Implementation*. IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pages 242–257, June 2001.

[Gutmann 99] J.S. Gutmann & K. Konolige. *Incremental Mapping of Large Cyclic Environments, Monterey, California*. In IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA (USA), pages 318–325, 1999.

[Haralick 89] R. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V.G. Vaidya & M.B. Kim. *Pose Estimation from Corresponding Point Data*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, no. 6, pages 1426–1446, Nov/Dec 1989.

[Haralick 94] R. Haralick. *Propagating covariances in computer vision*. In International Conference on Pattern Recognition, Jerusalem (Israel), pages 493–498, Sept. 1994.

[Harris 88] C. Harris & M. Stephens. *A Combined corner and edge Detector*. In 4th Alvey Vision Conference, pages 147–151, 1988.

[Heeger 92] D.J. Heeger & A.D.Jepson. *Subspace methods for recognition rigid motion I: Algorithm and implementation*. International Journal of Computer Vision, vol. 7, no. 2, pages 95–117, 1992.

[Jung 01] I-K. Jung & S. Lacroix. *A robust Interest Point Matching Algorithm*. In 8th International Conference on Computer Vision, Vancouver (Canada), July 2001.

[Kanade 94] T. Kanade & M. Okutomi. *A stereo matching algorithm with an adaptive window: Theory and experiment*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 9, pages 920–932, 1994.

[Kieffer 00] M. Kieffer, L. Jaulin, E. Walter & D. Meizel. *Robust Autonomous Robot Localization Using Interval Analysis*. Reliable Computing, vol. 6, no. 3, pages 337–362, Aug. 2000.

[Knight 01] J. G. H. Knight, A. J. Davison & I. D. Reid. *Constant Time SLAM using Postponement*. In Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems, 2001.

[Kwok 03] C. Kwok, D. Fox & M. Meila. *Adaptive Real-Time Particle Filters for Robot Localization*. In Proc. IEEE International Conference on Robotics and Automation, 2003.

[Leonard 91] J.J. Leonard & H.F. Durrant-Whyte. *Simultaneous map building and localization for an autonomous mobile*

robot. In IEEE/RSJ International Workshop on Intelligent Robots and Systems, Osaka (Japan), 1991.

[Leonard 01] J. J. Leonard & H. J. S. Feder. *Decoupled Stochastic Mapping*. IEEE Journal of Oceanic Engineering, pages 561–571, 2001.

[Lhuillier 03] M. Lhuillier & L. Quan. *Match propagaiton for image-based modeling and rendering*. IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 8, pages 1140–1146, 2003.

[Lindeberg 98] T. Lindeberg. *Feature detection with automatic scale selection*. International Journal on Computer Vision, vol. 30, no. 2, pages 79–116, 1998.

[Lowe 99] D.G. Lowe. *Object recognition from local scale-invariant features*. In 7th International Conference on Computer Vision, Kerkyra, Corfu (Greece), pages 1150–1157, 1999.

[Lu 97] F. Lu & E. Milios. *Globally consistent range scan alignment for environment mapping*. Autonomous Robots, vol. 4, no. 4, pages 333–349, 1997.

[Mallet 00] A. Mallet, S. Lacroix & L. Gallo. *Position Estimation in Outdoor Environments using Pixel Tracking and Stereovision*. In IEEE International Conference on Robotics and Automation, San Francisco, Ca (USA), pages 3519–3524, April 2000.

[Mandelbaum 99] R. Mandelbaum, G. Salgian & H. Sawhney. *Correlation based estimation of ego-motion and structure from motion and stereo*. In Proceedings of the International Conference on Computer Vision, pages 544–550, September 1999.

[Marco 01] M. Di Marco, A. Garulli, S. Lacroix & A. Vicino. *Set Membership Localization and Mapping for Autonomous*

Navigation. International Journal of Robust and Nonlinear Control, vol. 11, no. 7, pages 709–734, 2001.

[Martin 95] J. Martin & J. Crowley. *Comparison of correlation techniques*. In International Conference on Intelligent Autonmous Systems, Karlsruhe (Germany), pages 86–93, March 1995.

[Matthies 92] L. Matthies. *Toward Stochastic Modeling of Obstacle Detectability in Passive Stereo Range Imagery*. In IEEE International Conference on Computer Vision and Pattern Recognition, Champaign, Illinois (USA), pages 765–768, 1992.

[McKenna 97] S.J. McKenna, S. Gong, R.P. Würtz, J. Tanner & D. Banin. *Tracking Facial Feature Points with Gabor Wavelets and Shape Models*. In J. Bigün, G. Chollet & G. Borgefors, editeurs, Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication Crans-Montana, Switzerland, March 1997, volume 1206, pages 35–42. Springer Verlag, 1997.

[Mikolajczyk 01] K. Mikolajczyk & C. Schmid. *Indexing based on scale invariant interest points*. In 8th International Conference on Computer Vision, Vancouver (Canada), pages 525–531, 2001.

[Montemerlo 03] Michael Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.

[Moutarlier 91] P. Moutarlier & R. Chatila. *Incremental free-space modelling from uncertain data by an autonomous mobile robot*. In International Workshop on Intelligent Robots

and Systems, Osaka (Japon), pages 1052–1058, Nov. 1991.

[Neira 01] J. Neira & J. Tardos. *Data association in stochastic mapping using the joint compatibility test*. IEEE Transactions on Robotics and Automation, vol. 17, no. 6, pages 890–897, 2001.

[Nickels 02] K. Nickels & S. Hutchinson. *Estimating Uncertainty in SSD-Based Feature Tracking*. Image and Vision Computing, vol. 20, no. 2, pages 47–48, 2002.

[Noble 88] J. A. Noble. *Finding Corners*. Image and Vision Computing, vol. 6, no. 2, pages 121–128, 1988.

[Olson 01] C. Olson, L. Matthies, M. Schoppers & M. Maimone. *Stereo ego-motion improvements for robust rover navigation*. In IEEE International Conference on Robotics and Automation, pages 1099–1104, May 2001.

[Papanikolopoulos 95] N. Papanikolopoulos. *Selection of Features and Evaluation of Visual Measurements During Robotic Visual Serving Tasks*. Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 13, no. 3, pages 279–304, 1995.

[Rosenfeld 77] A. Rosenfeld & G.J.Vanderbrug. *Coarse-fine template matching*. IEEE Transaction on systems man and cybernetics, vol. 2, 1977.

[Schmid 97] C. Schmid & Roger Mohr. *Local greyvalue invariants for image retrieval*. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 19, no. 5, pages 530–535, May 1997.

[Schmid 98] C. Schmid, R. Mohr & C. Bauckhage. *Comparing and evaluating interest points*. In Proceeding of the 6th Inter-

national Conference on Computer Vision, Bombay (India), pages 230–235, January 1998.

[Se 01] S. Se, D. Lowe & J. Little. *Local and Global Localization for Mobile Robots using Visual Landmarks*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii (USA), pages 414–420, October 2001.

[Shi 94] J. Shi & C. Tomasi. *Good features to track*. In IEEE International Conference on Computer Vision and Pattern Recognition, Seattle (USA), pages 593–600, 1994.

[Singh 92] A. Singh & P. Allen. *Image flow computation: An estimation-theoretic framework and a unified perspective*. Image Understanding, vol. 56, no. 2, pages 152–177, 1992.

[Smith 87] R. Smith, M. Self & P. Cheeseman. *A Stochastic Map for Uncertain Spatial Relationships*. In Robotics Research: The Fourth International Symposium, Santa Cruz (USA), pages 468–474, 1987.

[Thrun 98] S. Thrun, D. Fox & W. Burgard. *A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots*. Autonomous Robots, vol. 5, pages 253–271, 1998.

[Thrun 00] S. Thrun, W. Burgard & D. Fox. *A Real-Time Algorithm for Mobile Robot With Applications to Multi-Robot and 3D Mapping*. In IEEE International Conference on Robotics and Automation, San Francisco, CA (USA), 2000.

[Thrun 02] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte & Ng. A.Y. *Simultaneous Mapping and Localization With Sparse Extended Information Filters*. In Proceedings of

the Fifth International Workshop on Algorithmic Foundations of Robotics, Nice, France, 2002.

[Thrun 03] S. Thrun & Y. Liu. *Multi-Robot SLAM With Sparse Extended Information Filters*. In 11th International Symposium of Robotics Research, Siena (Italy), October 2003.

[Vidal 01] R. Vidal, Y. Ma, , S. Hsu & S. Sastry. *Optimal motion estimation from multiview normalized epipolar constraint*. In 8th International Conference on Computer Vision, Vancouver (Canada), pages 34–41, July 2001.

[Wang 95] H. Wang & M. Brady. *Real-time Coner Detection Algorithm for motion estimation*. Image and Vision Computing, vol. 13, no. 9, pages 695–703, 1995.

[Weng 92] J. Weng, P. Cohen & N. Rebibo. *Motion and structure estimation from stereo image sequences*. IEEE Transactions on Robotics and Automation, vol. 8, no. 3, pages 362–382, June 1992.

[Wijk 00] O. Wijk & H.I. Christensen. *Triangulation Based Fusion of Sonar Data for Robust Robot Pose Tracking*. IEEE Transactions on Robotics and Automation, vol. 16, no. 6, pages 740–752, 2000.

[Wolf 00] C. Wolf, J-M. Jolion, W. Kropatsch & H. Bischof. *Content based Image retrieval using interest points and texture features*. In International Conference on Pattern Recognition, Barcelona (Spain), pages 234–237, Sept. 2000.

[Zabih 94] R. Zabih & J. Woodfill. *Non-parametric Local Transforms for Computing Visual Correspondence*. In Third European Conference on Computer Vision, Stockholm (Sweden), May 1994.

[Zhang 92] Z. Zhang & O. Faugeras. *Estimation of displacements from two 3-d frames obtained from stereo*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 12, pages 1141–1156, December 1992.

[Zhang 95] Z. Zhang, R. Deriche, O. Faugeras & Q-T. Luong. *A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry*. Artificial Intelligence Journal, vol. 78, pages 87–119, Oct 1995.