UNIVERSITÀ DEGLI STUDI FIRENZE

DOTTORATO DI RICERCA
IN

INFORMATICA, SISTEMI e TELECOMUNICAZIONI

Ingegneria Informatica, Multimedialitá e Telecomunicazioni

Ciclo XXVII
Coordinatore Prof. Chisci Luigi

Motion Estimation and 3D Reconstruction from Video Sequences

Stime di Moto e Ricostruzione 3D da Sequenze Video

Settore Scientifico Disciplinare ING-INF/05

Dottorando
Dott. Fanfani Marco

Tutors
Prof. Colombo Carlo

Prof. Allotta Benedetto

Coordinatore
Prof. Chisci Luigi

Anni 2012/2014

Marco Fanfani

Motion Estimation and 3D Reconstruction from Video Sequences

Ph.D. Thesis

December, 2014

Universitá degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione
Computational Vision Group

Abstract

The ability to obtain three-dimensional relations using only bi-dimensional projections of the surrounding environment is a basic key-element for any autonomous robotic system that wants to emulate the human vision to actively interact with the world. This thesis focuses on the geometrical estimation of motion and three-dimensional structure using as only input an image sequence.

In the first part two Simultaneous Localization and Mapping (SLAM) algorithms are presented. The SLAM softwares are designed with the aim to provide accurate estimation in real-time without the need of back-corrections or global optimizations. In the second part an hybrid 3D reconstruction method is described: Using an active laser projector to retrieve the three-dimensional structure of the scene, and a passive motion estimation to track the device movements, accurate 3D model of desktop-size objects are built.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The ability to estimate self position and a rough representation of the surrounding environment is an high desirable feature for any autonomous system. Without any prior knowlegde neither about the world nor its position, a robot have to estimate its pose and the map landmarks simultaneously. Techniques named as Simultaneous Localization and Mapping (SLAM) address this problem (Durrant-Whyte and Bailey, 2006).

First works on the SLAM theory — also know as Concurrent Mapping and Localization — appeared in the robotic community as a fundamental step to build a fully autonomous robot able to move in unknown environment. Initially, statistical foundation to solve the probabilistic mapping process was addressed (Smith and Cheeseman, 1986, Durrant-Whyte, 1988). Shortly after it was clear that to solve the problem a joint state of pose and landmark positions had to be incrementally estimated based on new observations (Leonard and Durrant-Whyte, 1991). In fact to solve the localization problem a reference map of the scene must be known; On the other hand, to update and augment the map with new or refined landmarks, the actual position have to be already estimated. The structure of the SLAM problem—and its acronym—was finally defined during the 1995 International Symposium on Robotic Research (Durrant-Whyte et al., 1996). Then, with the beginning of the new century, an increasing interest was put on the SLAM problem with the organization of focused workshops and summer schools.

Various kind of sensors were soon employed to perceive the world, such as laser range finder (Chatila and Laumond, 1985), sonar-based (Crowley, 1989) and vision-based (Ayache and Faugeras, 1988) systems, just to name the most relevant.

Among the available sensors, cameras are probably the most versatile and economic choice. They are passive sensor, able to produce high resolution images. However the extraction of useful information from images is difficult and prone to error: For example, visual features needed to correlate multiple pictures are hard to track and three-dimensional landmarks are only estimable from at least two independent measurements.

1.1 Visual SLAM

With Visual SLAM (vSLAM) we refer to specific SLAM solutions that estimate pose and map using only information extracted from images. Typical inputs for this kind of problem are video sequences recorded with a known device, i.e. all calibration parameters are supposed to be known.

Stable and accurate feature points have to be extracted from the images and successively tracked among the frames of the sequence so to establish initial constraints on the input data. Then, after an initialization step—particularly hard when a single camera is employed—a 3D map representation begin to be available.

The three-dimensional data are represented in form of a sparse point cloud; Even if this representation is insufficient to build a detailed and complete model of the environment, is suitable to guarantee an accurate localization, and with some post processing is sufficient to give to a robot a rough estimation of the scene. Moreover, sparse representation are more manageable in term of both computational time and memory occupancy.

Once obtained correspondences between measured image features and 3D points in the map, the position is estimated minimizing the re-projection error and the map is updated with new and refined data (see Fig. 1.1).

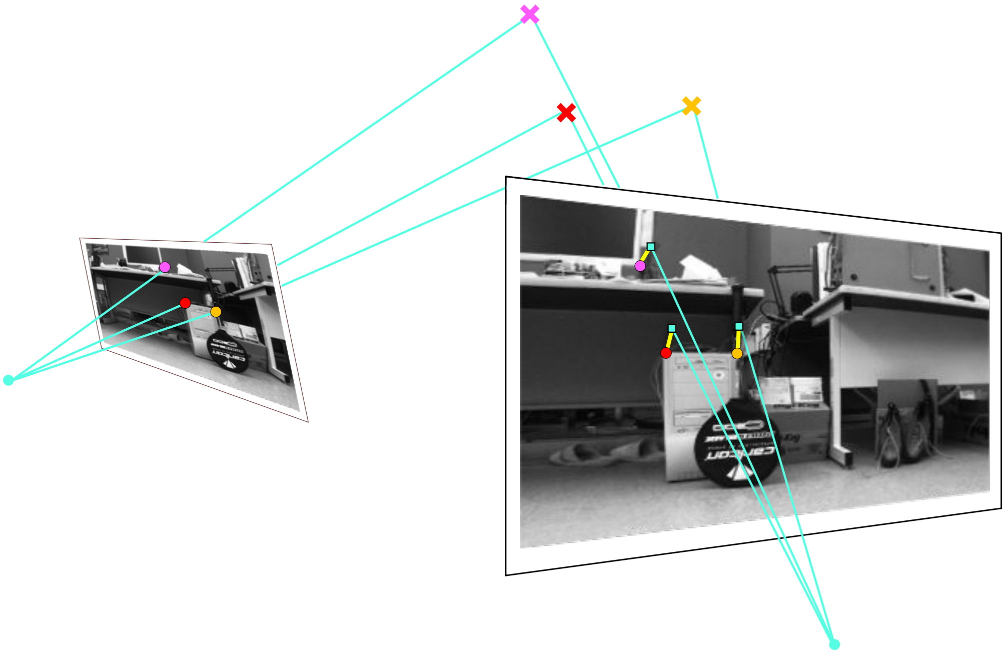


FIGURE 1.1: (Best viewed in color) To estimate the position of a camera the re-projection error (yellow segments) between 2D feature correspondences (red, blue, orange and purple points) and the projections of the relative 3D points (cyan square dots) must be minimized.

First works on vSLAM employed solutions based on recursive Bayesian filtering techniques (Chen, 2003), such as the Extended Kalman Filter (EKF), where the pose estimation and the map refinement were tackled at the same time. After few years different approaches were proposed, based on the Structure from Motion (SfM) theory (Szeliski, 2010, Chapter 7)—a widely known topic in the computer vision community. In this case the localization and the mapping task could be accomplished with an interleaving approach. In any case, both formulations assume a mainly static scene; While vSLAM system are more or less able to deal with occasionally moving objects in the scene, they fail to operate in an high changing environment given that not enough stable landmarks could be found.

In the following we will refer to the former method as Bayesian vSLAM and as Keyframe vSLAM for the latter. In the description a monocular setup will be assumed. Differences between the mono and stereo case will be addressed later.

1.1.1 Bayesian vSLAM

In Bayesian vSLAM, Extended Kalman Filters (EKF) are typically used to simultaneously estimate the joint state that defines the pose and the map, alternating the prediction and update phases.

During the prediction step the device position estimates—expressed with a 6 degree of freedom (DoF) vector describing a three-dimensional roto-translation—are moved forward with a velocity model and, consequentially, the covariance matrix is modified increasing the uncertainty of the system. When a new measurement (i.e. a new image) is available, matches between features in the previous and current frame are established and with a non-linear update equation (the pin-hole projection model) the updated full state (position plus landmarks) is computed.

Using only one camera, the exact position of a 3D point could not be estimated until it is viewed again in a second image. To overcome this problem and avoid a delayed initialization, a particle filter created over the optical ray (Davison, 2003) or an inverse depth parametrization (Montiel et al., 2006) could be used.

During the years many works have been proposed exploiting a similar approach with positive results. Nevertheless to deal with real-time algorithms a constant computation time must be respected. For this reason only few 3D points could be used so to keep the filter state dimension quite constant and avoid a computational time increase.

As stated in (Strasdat et al., 2012) this is the main drawback of Bayesian vSLAM approaches: Using a strongly limited number of 3D correspondences the estimation accuracy is inferior w.r.t. Keyframe-based methods.

1.1.2 Keyframe vSLAM

In the computer vision community camera localization and map building are tasks typically accomplished through the Structure from Motion (SfM) theory. Exploiting the projective geometry relationship (Hartley and Zisserman, 2004) between the image space and the scene 3D space, SfM softwares (Farenzena et al., 2009, Agarwal et al., 2009) starting from a sparse collection of images are able to compute a point cloud representation of the scene and, as byproduct, the localization of each image. It's easy to see a strong correlation with the SLAM problem. The main differences are the order by which to acquire information and the fulfillment of the real time constraint: In particular images must be used in temporal order and the localization and map output have to be provided at regular time intervals.

In this case, a standard pipeline include the following steps: For each frame, features have to be detected and matched across past frames, then during the initialization phase the Essential matrix decomposition algorithm is used to retrieve the poses for the first couple of frames and compute a 3D map by triangulation (Hartley and Sturm, 1997). This approach eases the initialization w.r.t. the Bayesian case, avoiding the need to include partial observation—feature points seen for the first time can be used at most to constraint the rotation motion only. After the initial phase an exterior orientation problem (Hartley and Zisserman, 2004, Chapther 7) is solved, exploiting the known correspondences between 3D landmarks and 2D features. As for the Bayesian framework, the pose is estimated minimizing the re-projection error with non linear equations; Nevertheless with this approach robust routines, such as RANSAC (Fischler and Bolles, 1981), could be included during the estimation process, enabling the system to deal with strong noisy measurements (i.e. outliers).

It's worth noting that, the camera pose and the 3D map, even if highly correlated, are estimated in different times even at different frequencies: Using this observation it's possible to split the frames into two separate classes: frames that only need to be localized and frames that have also to update the map. The latter are called keyframes. In this sense, keyframes are images sampled along the sequence for which triangulation and re-projection could be accomplished with high accuracy. A careful criteria for keyframe selection is an important factor for vSLAM systems.

Finally to reduce the estimation error, Bundle Adjustment (BA) (Triggs et al., 2000) is used to improve the accuracy of both pose and map. The introduction of this refinement process is time consuming, so it has to be executed rarely and on few data. This has lead to implementations of local BA, working only on a limited set of keyframes and the connected 3D representation (only 3D points visible in the keyframe subset are included in the optimization).

1.1.3 Loop closure

SLAM approaches, by updating incrementally poses and map, are prone to accumulate errors during time. This will eventually lead to strong erroneous outputs, manifested as a drift w.r.t. the ground truth trajectory. Particularly affected by this kind of problem are monocular setups that have to deal with the scale factor ambiguity.

The main solution that seems to overcome or at least mitigate the drift error is the Loop Closure (LC) (Williams et al., 2009). When an already visited place is seen again new constraints among actual and past measurements can be established. Adding such information in an optimization routine could lead to an high increasing of accuracy. Loop closure therefore include two distinct step: The loop closure detection and the optimization of poses and map.

In literature various approaches to solve the detection and refinement problems appeared during the years. In most of the cases the detection is carried out evaluating the similarity between images—inspired by content based image retrieval—with careful implementation that can operate fast and avoid false positive detection. In fact, by adding an erroneous correlation the estimates will be highly corrupted. To globally optimize the detected loopy trajectory a topological representation of the pose and map is used: dividing the SLAM problem in many sub-problem represented in independent reference frame, i.e. sub-map, and knowing the relative transformation that link subsequent sub-map the system could be expressed with a graph that include in the nodes the local maps and on the edge the relative transformation. Then by imposing that a complete loop will produce an identity transformation a graph optimization algorithm—such as the g^{2}o framework (Kümmerle et al., 2011)—could increase the final SLAM output accuracy.

Although loop closure demonstrated good results to correct drift errors, in our opinion its practical use is strongly limited for two main reason. SLAM should deliver an accurate localization and a map representation output in real time, given that these estimates are likely to be used immediately; instead with the loop closure method only past, and probably already used, estimates could be corrected. Moreover to detect a loop closure, a loopy trajectory is needed, very frequent in odometry and SLAM dataset, but less plausible in a real application scenario.

1.1.4 Monocular and Stereo configurations

During this introduction the vSLAM problem has been presented mostly keeping in mind the monocular setup, where only a single moving camera can provide measurements of the environment. Another widely employed configuration is the stereo setup: The SLAM sensor is equipped with two synchronized cameras kept in relative fixed position and fully calibrated—not only intrinsic parameters of each camera, but also extrinsic values describing the roto-translation between the two cameras are known a-priori.

While the monocular setup is highly versatile—a single camera can deal with close and far scene without particular efforts—less expensive and widely available on consumer mobile devices, it is affected by additional issues such as the delayed 3D computation and the scale factor uncertainty.

On the other hand stereo setups ease these problems providing a 3D representation at each time, obtained by triangulation of stereo pair correspondences, and the metric scale factor of the scene using the extrinsic calibration parameters. In this way more accurate estimates could be computed, at the cost of a minor flexibility of the system: The pair baseline has to be related to the environment (shorter for close scenes, longer for wide areas), a careful calibration must be performed, and few stereo devices are available off-the-shelf.

Finally, it's worth noting that other setups have been proposed in the literature, such as omni-directional camera (Rituerto et al., 2010) or multiple camera configuration (Zou and Tan, 2013).

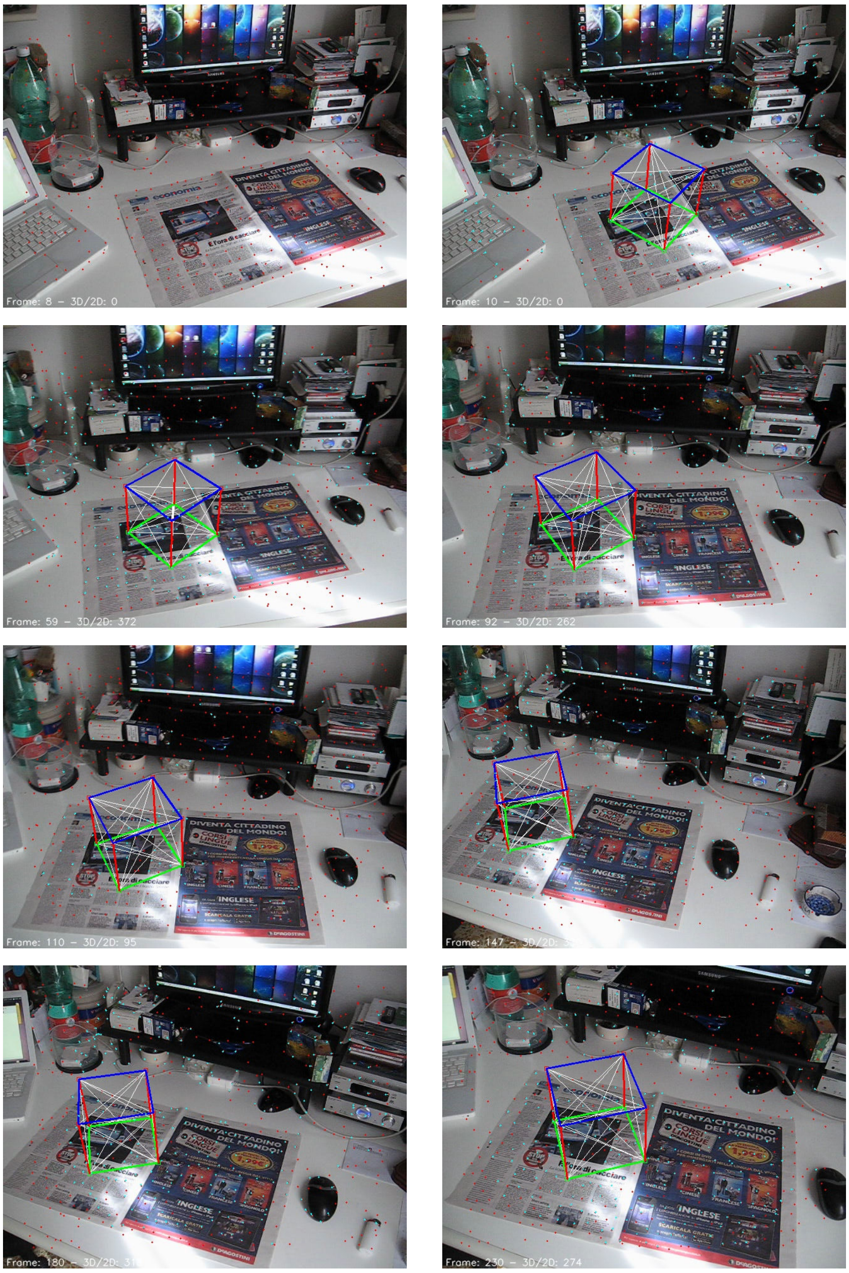


Figure 1.2: Frames of an augmented reality sequence. Over the real images is drawn a virtual wire-frame cube that is subject to proper perspective distortions.

1.1.5 Application

As already said, vSLAM can estimate incrementally both the location of the device and a representation of the environment. This two output are already sufficient to justify the use of SLAM module for autonomous robot and the interest of the scientific community aiming to develop ever more accurate and reliable solutions.

In literature it's possible to find applications in indoord (Davison, 2003, Klein and Murray, 2007), outdoor (Lim et al., 2011, Mei et al., 2011), aerial (Nemra and

Aouf, 2009) and even underwater (Mahon et al., 2008) environment. Anyhow it's important to understand that in most application fields vSLAM is mainly used together with other sensory devices, such as Global Positioning System (GPS), Inertial Measurement Unit (IMU), Doppler Velocity Log (DVL), ect. In the ARROWS project [1], that see the Computational Vision Group of the University of Florence as a partner, vSLAM will be included in a distributed framework that will fuse different measurements with the aim to provide a sort of awareness for underwater autonomous vehicle (AUV).

In addition vSLAM could be used also as a fundamental step to realize different applications. The localization output could be easily exploited to develop Augmented Reality (AR) applications. In AR softwares a virtual object is drawn over the acquired images at runtime. While the device is moved, the object must be subjected to correct projective deformations to give to the user the feeling to observe a real object. In Fig. 1.2 a simple test object (a wire-frame cube) is projected onto the scene and transformed accordingly with the camera motion. In our experiment we see that to satisfy the user perception no particularly accurate estimates are needed, so even simple monocular vSLAM software could be used for this task.

In a similar way visual motion estimates could be included in an active reconstruction software to design an hybrid device with a camera and a laser fan projector to build dense 3D models (see Ch. 3). Differently form the AR application, hybrid 3D reconstruction needs more accurate estimates in order to avoid strong noises in the final model.

Finally also 3D map could be used to extract rough geometric information about the scene. For example, in an autonomous driving system, the three-dimensional point cloud could be used to find the road plane simply by solving a linear system in a robust estimation framework. If denser reconstructions are needed the vSLAM map could be augmented with a feature expansion approach (Furukawa and Ponce, 2010) or using multiple stereo depth-maps fused with the localization information and finally refined (Bradley et al., 2008) (see Fig. 1.3).

[1] The ARROWS project is supported by the European Commission under the Environment Theme of the 7th Framework Programme for Research and Technological Development (http://www.arrowsproject.eu/).

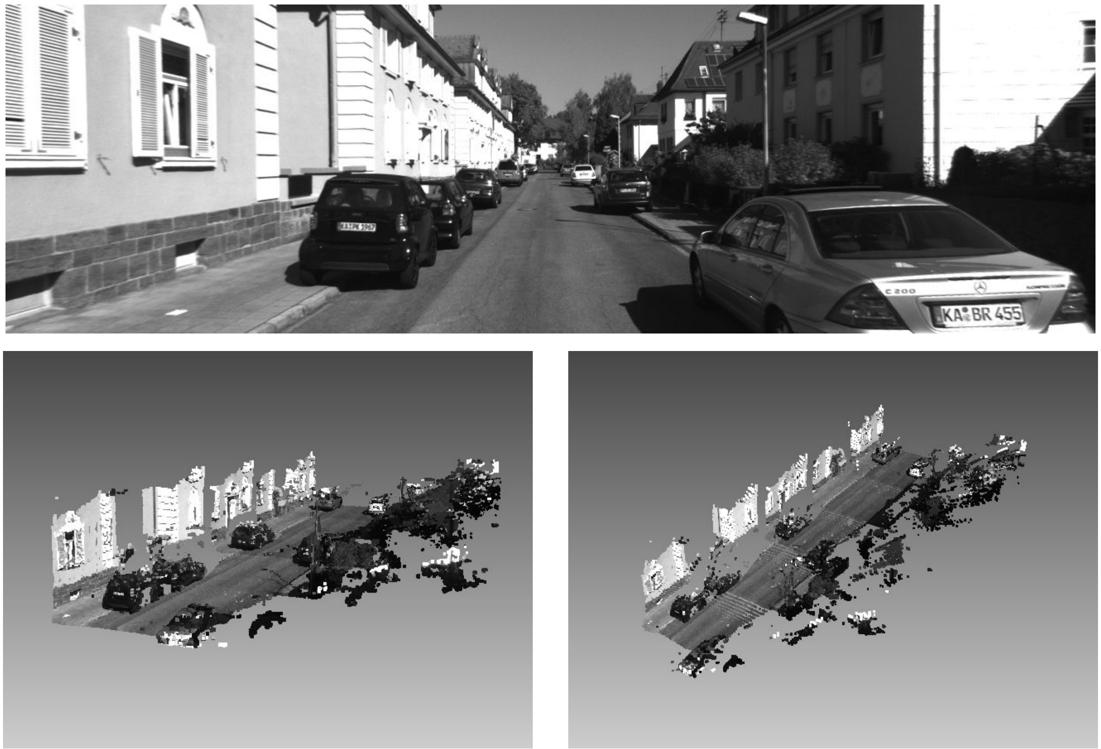


Figure 1.3: Dense 3D reconstruction of a road scene. Top: one frame of the original video sequence. Bottom: two views of the 3D model obtained by fusion of local dense point clouds (computed with the OpenCV SGBM algoritm) exploiting vSLAM motion estimates.

1.2 Contributions

The aim of the work in this thesis is to study and develop robust visual Simultaneous Localization and Mapping algorithms for both the monocular and the stereo setup. In Ch. 2 the mono and stereo SLAM softwares are presented and evaluated. Both solutions employ the keyframe-based approach but avoid the use of global optimization techniques and back-correction of the estimates (i.e. neither Bundle Adjustment nor Loop Closure are employed). Instead is performed a careful selection of information using an accurate feature matching scheme, robust estimations to detect and reject outliers and a keyframe selection strategy that guaranties sufficient baseline among successive keyframes.

The proposed monocular setup, named Simulated Annealing Monocular SLAM (SAMSLAM, see Sect. 2.2), is based on an optimization scheme that iteratively solve an absolute orientation and a perspective n-point problem in a RANSAC framework, decreasing the threshold for the maximum allowable errors at each repetition. This approach, inspired by the Simulated Annealing algorithm, provides more accurate solutions at each iteration.

The stereo approach, Selective Stereo SLAM (SSLAM, see Sect. 2.3), exploits the stereo pair configuration to obtain accurate estimates of the 3D environment and makes a careful selection of keyframes based on the feature temporal flow on the images. Working with local information the system is able to correctly estimate long trajectories with a very low drift error, without requiring any global optimization.

As a particular application of the vSLAM estimates, in Ch. 3 two hybrid 3D reconstruction approaches are presented and compared: the first solution employs the monocular motion estimates while the second one exploits a stereo camera setup. Both solutions achieve good accuracy results providing a simple and economic device for 3D acquisitions.

1.3 Publication

The work presented in this thesis resulted in the following publication:

- M. Fanfani and C. Colombo. Hybrid 3D Reconstruction: SLAM vs Homography-Based Motion Estimation. In *Proc. of 6th Convegno del Gruppo Italiano Ricercatori in Pattern Recognition, GIRPR 2012*, 2012

- M. Fanfani and C. Colombo. LaserGun: A Tool for Hybrid 3D Reconstruction. In *Proc. of 9th International Conference on Computer Vision Systems, ICVS 2013*, pages 274–283, 2013

- M. Fanfani, F. Bellavia, F. Pazzaglia, and C. Colombo. SAMSLAM: Simulated annealing monocular SLAM. In *Proc. of 15th International Conference on Computer Analysis of Images and Patterns, CAIP 2013*, pages 515–522, 2013

- F. Bellavia, M. Fanfani, F. Pazzaglia, and C. Colombo. Robust selective stereo SLAM without loop closure and bundle adjustment. In *Proc. of 17th International Conference on Image Analysis and Processing, ICIAP 2013*, pages 462–471, 2013

1.4 Thesis Structure

After this introduction in Chapter 2 the SLAM softwares are presented. More specifically in Sect. 2.2 and 2.3 are respectively described and evaluated the monocular SAMSLAM method and the stereo SSLAM approach.

Chapter 3 is devoted to the 3D hybrid reconstruction system that employs the SLAM motion estimates and the active triangulation approach exploiting the deformation of a laser stripe.

Finally in Chapter 4 conclusions are drawn.

CHAPTER 2

Visual Simultaneous Localization and Mapping

In this Chapter both the monocular and the stereo vSLAM algorithms developed during my PhD studies will be described in detail and experimental results will be presented for each system. Before, a brief review of works related to the vSLAM is given hereafter.

2.1 Related Works

Ayache and Faugeras (1988) studied possible solutions for the estimation of three-dimensional representations of the environment fusing multiple estimations with an EKF to reduce uncertainties; Points, lines and planes are used as features to register successive 3D maps. In (Harris and Pike, 1987) the authors present the DROID system in which 3D maps are represented as point clouds where for each point a position estimate and the related uncertainty is given.

Broida et al. (1990) describe a system able to track camera motion starting from a batch initialization. In addition they address the problem of feature matching and propose the idea of exploiting the estimate uncertainties to reduce the search area in the new image. In (Chiuso et al., 2002) a EKF based viual SLAM approach, named by the author as causal Structure from Motion, is described with a minimal realization of the problem and particular attention to feature occlusion problems.

A successful work on monocular SLAM was *MonoSLAM* (Davison, 2003). In this work an EKF state representation similar to that used by Chiuso et al. (2002) is used to realize a fully operational SLAM system able to track the motion of an hand-held camera. The state vector include the positions of both the camera and the 3D landmarks, plus a linear and angular velocity. At each time the velocities are perturbed with acceleration (modeled as Gaussian distribution), and then used to predict the new camera location; To update the state with new measurements correspondences between 3D landmarks and image features are used. To ease the matching 3D points uncertainties are used to limit the search area. Previously unseen landmarks are initialized using a particle filter over the optical ray passing through the 2D features; By observing again the landmark as time goes by the depth estimate converge to a peak and than the point is inserted in the EKF state. In a successive version of the software this problem was resolved using an inverse depth parametrization (Montiel et al., 2006)—able to encode also infinite depth—that initializes the landmark with a wide depth uncertainty.

Instead of an EKF, Eade and Drummond (2006) use a Rao-Blackwel-lized particle filter—inspired by the *FastSLAM* software (Montemerlo et al., 2002)—to estimate a monocular camera trajectory by efficiently tracking hundreds of landmarks, hard using the EKF implementations.

While approaches EKF based were developed, other research groups proposed different solutions to the SLAM problem. In (Nistér et al., 2004) a Visual Odometry system for mono and stereo configuration is described. Once established the correspondences between images, a robust 3-point algorithm (Haralick et al., 1994) is implemented to estimate the camera position. To mitigate trajectory drift, the authors insert *firewalls* at particular times to split the estimation into sub-problems, limiting the error growth. Anyhow no loop closure techniques are used. Mouragnon et al. (2006) present a monocular SLAM system similar to the previous cited one, but including also a local Bundle Adjustment—ran when a new keyframe is detected—to obtain better estimates without excessive increase in computational times.

Klein and Murray (2007) developed the *PTAM* system, a monocular SLAM designed for augmented reality application in close environment. Apart from the pipeline used for the pose estimation—that includes quite standard steps—the main contribution of the paper is to exploit two separate and parallel estimation procedures for the localization and mapping tasks. This idea came from the observation that while camera pose must be computed with high frequency (relative to the real camera speed) the 3D map doesn't need to be updated frequently—given a static environment. A slower mapping strategy enables the system to use time consuming optimization: A global Bundle Adjustment is performed over the whole set of keyframes and the 3D map, achieving impressive results.

In the following years an increasing interest toward keyframe-based vSLAM software was notable. Mei et al. (2009) describe a stereo vSLAM system that employs an adaptive sub-map selection—named *active region*—to both localize the estimation and, in case of detected loop closure, correct the trajectory without an explicit use of Bundle Adjustment.

In Strasdat et al. (2010) the authors ask the question "Why filter?", and propose a comparison between Bayesian and keyframe based SLAM evaluating the trade-off between accuracy and computational cost. They come to the conclusion that keyframe based approaches, implementing a Bundle Adjustment optimization, show an increased accuracy with respect to the filtering solutions. This result is motivated by the observation that using a lot of landmark measurements and tracking only a sparse set of frames (keyframes) produces better estimates. Following this hint, more published solutions propose systems similar to *PTAM* but including pose-graph optimizations to deal with larger workspaces.

In (Strasdat et al., 2010) a monocular keyframe-based approach is presented to deal with large scale environments: Using a double thread solution similar to *PTAM*, the authors define an improved back-end for map optimization. Local Bundle Adjustment is carried out on keyframes and 3D points included in a sliding window. To achieve scale-drift free estimations, loop closure is used. When detected a previously seen scene, the loop is closed and a 7 DoF (rotation, translation plus scale) optimization is performed on the pose-graph. Then Strasdat et al. (2011) propose a framework to handle both mono and stereo setup. They employ a *double window optimization* where in the small inner windows Bundle Adjustment optimization is carried out using hard pose-to-point constraints while in the outer windows soft pose-to-pose constraints are used to solve a pose-graph optimization.

A similar approach is presented in (Lim et al., 2011); While Strasdat et al. optimize the inner and outer window simultaneously, Lim et al. interleave between a local and global Bundle Adjustment exploiting a *divide-and-conquer* strategy—a two level hierarchical optimization where local maps are seen as rigid object in the global optimization.

Apart from the main division in Bayesian and keyframe based approaches, vSLAM can also be characterized by the feature matching strategies adopted. Many works use computational non expensive solutions to find feature correspondences, such as in (Davison, 2003) or in (Klein and Murray, 2007) where after a quick corner detection matches are computed with an *active search* approach and fast similarity measures (i.e. normalized sum-of-squared distances). SIFT features are used by Se et al. (2001) to build a trinocular vSLAM system while in (Beall et al., 2010) their robustness is exploited to work in underwater environment. Also SURF features have been used (Zhang and Kang, 2008). In (Klein and Murray, 2008) edge features are added to the map to improve tracking robustness to motion blur and fast movements.

2.2 Simulated Annealing Monocular SLAM

In this section a novel monocular visual SLAM is presented, named SAMSLAM. The main contribution is the development of a robust optimization scheme—inspired by Simulated Annealing strategy—in order to replace more typical refinement approaches like local Bundle Adjustment.

SAMSLAM works locally on overlapping triplets of successive keyframes with the aim to guarantee scale and 3D structure coherence. Given a calibrated image sequence $S = \{I_t\}$, with radial distortion corrected, SAMSLAM proceeds by detecting successive triplets $T_i = \{I_{k_{i-1}}, I_{k_i}, I_{k_{i+1}}\}$ of image keyframes $\{I_{k_i}\} \subseteq S$, $k_0 = 0, k_i < k_{i+1}$—see Fig. 2.1.

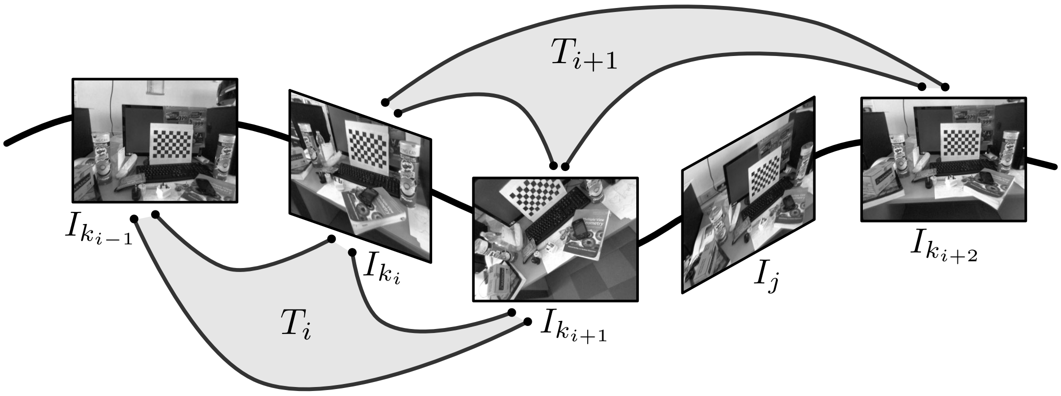


Figure 2.1: Overview of the SAMSLAM approach. Keyframe triplets T_i and T_{i+1} are used to estimate successive overlapping local 3D maps, that are then employed to retrieve the pose of a generic image frame I_j.

A local 3D map $\mathcal{M}_i$ and the keyframe poses, P_{k_{i-1},k_i} and $\mathrm{P}_{k_{i-1},k_{i+1}}$ are obtained for the current keyframe triplet T_i using the proposed simulated annealing scheme. The optimization iterates by interleaving between the three-dimensional registration of maps obtained from each image pairs $(\mathrm{I}_{k_{i-1}}, \mathrm{I}_{k_i})$, $(\mathrm{I}_{k_i}, \mathrm{I}_{k_{i+1}})$ and $(\mathrm{I}_{k_{i-1}}, \mathrm{I}_{k_{i+1}})$ and the computation of the camera poses expressed in a local reference frame.

Both estimations are wrapped in RANSAC frameworks to eliminate possible outlier measurements. While the iterations proceed, thresholds on the re-projection error (used to evaluate both the RANSAC hypothesis and the updated results) are progressively reduced. In other words, starting from a rough solution for the poses and map, in successive steps the algorithm recomputes refined outputs.

As the keyframe triplet is updated from T_i to T_{i+1}, the first keyframe is dropped and a new one is queued, so that the 3D maps $\mathcal{M}_i$ and $\mathcal{M}_{i+1}$ overlap and the consistency of scale and 3D structure is guaranteed. The positions of images that are not included into the keyframe set, are computed exploiting 2D/3D correspondences between matched features and points of last optimized 3D map using the EPnP algorithm (Lepetit et al., 2009).

In Subsec. 2.2.1 the implemented feature matching strategy is described. Then in Subsec. 2.2.2 details of the simulated annealing optimization are given. Subsection 2.2.3 concludes presenting results obtained with SAMSLAM.

2.2.1 Feature Matching and Keyframe Selection

Image alignment for the generic pair (I_{t_1}, I_{t_2}) is based on keypoint matching. For each image, keypoints are extracted using the HarrisZ detector (Bellavia et al., 2011), and matches are found using the sGLOH descriptor (Bellavia et al., 2010) with a Nearest Neighbor scheme.

HarrisZ Detector

The HarrisZ is an extension of the Harris-affine detector that avoid the need of tuning the thresholds used to compute and select a corner.

As for the standard Harris detector, for each point $\mathbf{p} = (x, y)$ the autocorrelation matrix $\mu(\mathbf{p})$ is computed using the partial derivatives of the image $I_x(\mathbf{p})$ and $I_y(\mathbf{p})$ integrated over a small window N centered on $\mathbf{p}$, as

$$\mu(\mathbf{p}) = \begin{bmatrix} \sum\limits_{\mathbf{q}\in N} I_x^2(\mathbf{q}) & \sum\limits_{\mathbf{q}\in N} I_x(\mathbf{q})I_y(\mathbf{q}) \\ \sum\limits_{\mathbf{q}\in N} I_x(\mathbf{q})I_y(\mathbf{q}) & \sum\limits_{\mathbf{q}\in N} I_y^2(\mathbf{q}) \end{bmatrix} \tag{2.1}$$

Then exploiting the relation between eigenvalues of $\mu(\mathbf{p})$ and its determinant and trace, corner are extracted if $H_z(\mathbf{p}) > \tau$ where

$$H_z(\mathbf{p}) = Z(\det(\mu(\mathbf{p}))) - Z(\mathrm{tr}^2(\mu(\mathbf{p}))) \tag{2.2}$$

and

$$Z(x) = \frac{x - \mathrm{mean}(x)}{\mathrm{std}(x)} \tag{2.3}$$

is the z-score function (DeGroot and Schervish, 2001). This normalization avoids the introduction of both the scalar factor κ used in the standard Harris method (Harris and Stephens, 1988) to balance the determinant and trace contributions, and the threshold τ used to select a corner that can be fixed to $\tau = 0$—given that the mean value of determinant and trace (equal to zero after normalization) can be related to flat regions. To improve robustness the image derivatives are filtered with a smoothed edge map in order to enhance the gradient values near discontinuities.

The method is used in a scale-space framework to guarantee scale-invariance. Corner detection at different scales is obtained repeating the method by convolving the image and a Gaussian filter with an increasing standard deviation. It's worth noting that no pyramidal representation is used: This choice increases the corner detection accuracy even at higher scales at the expense of more computational time to compute the convolutions with bigger Gaussian kernels. ■

sGLOH Descriptor

sGLOH is a histogram-based descriptor able to handle efficiently rotation invariance. In classical histogram-based descriptor—like SIFT (Lowe, 2004)—for each feature a signature is defined by concatenation of orientation histograms computed in blocks of a feature-centered grid. Typically, invariance w.r.t. rotation is obtained using a dominant orientation given by the higher bin of the gradient orientation histogram computed on the feature neighborhood.

Instead of using a square grid (as in the SIFT), a polar grid is used, following the GLOH descriptor (Mikolajczyk and Schmid, 2005), with N circular rings each equally split into M region, defining NM regions $R_{n,m}$ with $n = \{1, \ldots, N\}$ and $m = \{0, \ldots, M-1\}$. The central area could be defined as a single region $R_{0,0}$ or divided in M orientation $R_{0,m}$. In Fig. 2.2 example grids are shown.

For each region $R_{n,m}$ an orientation histogram is computed. Each bin $h^i_{n,m}$ of the histogram is computed with a kernel density estimation by Gaussian window as

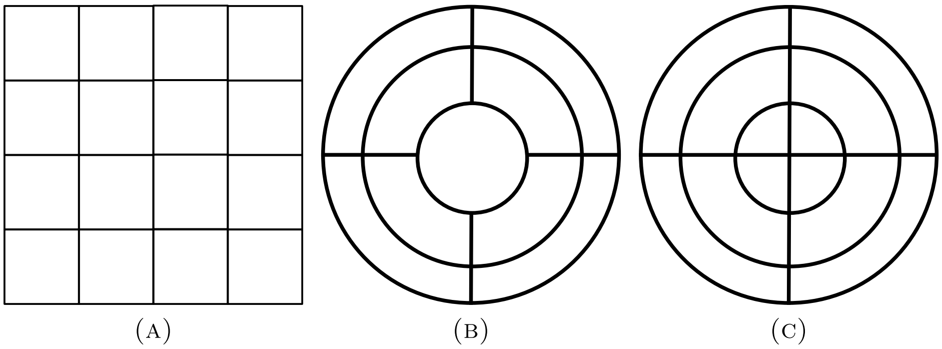


FIGURE 2.2: Descriptor grid examples: (A) the SIFT grid. (B,C) sGLOH grids respectively with single and divided central region. (Image taken from (Bellavia et al., 2010))

$$h^i_{n,m} = \frac{1}{\sqrt{2\pi}\sigma} \sum_{\mathbf{p}\in R_{n,m}} \nabla_{\mathrm{M}}(\mathbf{p}) \exp\left(-\frac{(\Phi_{2\pi}(\nabla_\theta(\mathbf{p}) - \theta_i))^2}{2\sigma^2}\right) \tag{2.4}$$

where ∇_{M} and ∇_θ are respectively the gradient magnitude and orientation, θ_i is the mean orientation of the i-th bin and σ the bin amplitude. $\Phi_{2\pi}(x)$ is a function that takes into account 2π periodicity. More intuitively, each bin accumulates scores based on the bin-point orientation difference weighted with the point gradient magnitude, so that more importance is given to points in which gradient computation is more robust.

Then the histogram $H_{n,m}$ related to the region $R_{n,m}$ is obtained by concatenation of $h^i_{n,m}$ for $i = \{0, \ldots, M-1\}$. The first bin in these local histogram represents the direction that points outward w.r.t. the grid center, then the remaining bins are appended in order. Finally, the full descriptor is obtained by concatenating the local histograms. If a rotation transform αk, with $\alpha = 2\pi/m$ and $k = 0, \ldots, M-1$, is applied the relative descriptor is modified by a cyclic shift of the local region histograms for each ring without the need of recomputing the whole descriptor (see Fig. 2.3).

A rotation-invariant similarity measure has to be defined in order to exploit the full descriptive power of the sGLOH. Given two descriptors $H^{(a)}$ and $H^{(b)}$, their distance is the minimum w.r.t. each possible rotation angle

$$\hat{D}(H^{(a)}, H^{(b)}) = \min_{k=0,\ldots,M-1} D(H^{(a)}, H^{(b)}_{\alpha k}) \tag{2.5}$$

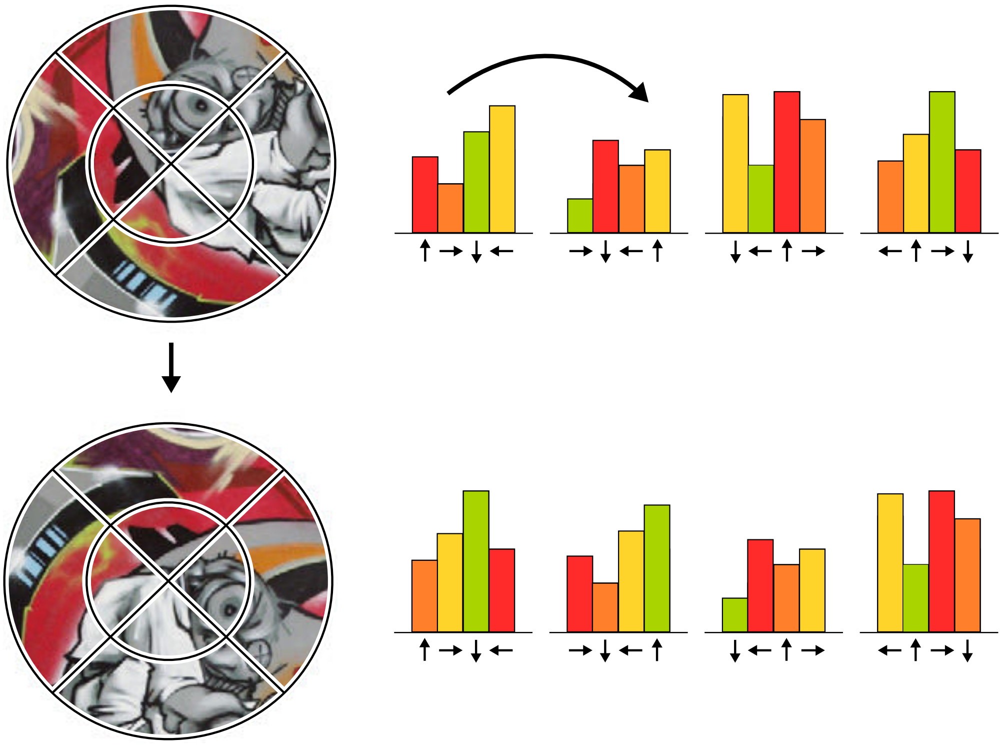


FIGURE 2.3: A rotation of the same visual patch is expressed as a cyclic shift of the local region histograms of the sGLOH descriptor. (Image taken from (Bellavia et al., 2014), best viewed in color)

It's worth noting that to achieve better matching accuracy in a general framework, the authors propose to use the sGOr approach (Bellavia et al., 2014). After a first matching computation, for each best match obtained comparing a point $\mathbf{x}^{(a)} \in I^{(a)}$ with any point in $I^{(b)}$, the chosen orientation is used to vote for the global relative orientation k_g, roughly representing the image rotation—under the assumption that all scene points undergo a similar rotation. Then matches are recomputed limiting the possible orientations to $k = \{k_g - 1, k_g, k_g + 1\}$ in order to correct wrong matches. ■

Once obtained the correspondences between image pairs, these are refined on a temporal constraint basis. Given that frames are acquired in sequence and feature points move slowly in the image domain, it's useful to limit the correspondences search space in a neighborhood of the previous location of the feature. More formally, let $\mathbf{x}_{t_1} \in I_{t_1}$ be a generic keypoint of image I_{t_1}, the corresponding feature $\mathbf{x}_{t_2} \in I_{t_2}$ must lie in a circular window of radius δ_r (that define the maximum allowable flow displacement), i.e.

$$\| \mathbf{x}_{t_1} - \mathbf{x}_{t_2} \|_2 < \delta_r \tag{2.6}$$

Obtained matches are refined in a RANSAC framework, computing the fundamental matrix, so to discard correspondences that not satisfy the epipolar constraint ($\mathbf{x}_{t_2}^{\mathrm{T}} F_{1,2} \mathbf{x}_{t_1} = 0$). Moreover, when a new keyframe triplet $T_i = \{I_{k_{i-1}}, I_{k_i}, I_{k_{i+1}}\}$ is found only matches which form a *loop chain* $\mathcal{C}_i$ are kept

$$\mathcal{C}_i = \big\{(\mathbf{x}_{k_{i-1}}, \mathbf{x}_{k_i}), (\mathbf{x}_{k_i}, \mathbf{x}_{k_{i+1}}), (\mathbf{x}_{k_{i+1}}, \mathbf{x}_{k_{i-1}})\big\} \tag{2.7}$$

i.e. only features visible in all the three keyframes are used to estimate the 3D map $\mathcal{M}_i$ and the relative keyframe poses P_{k_{i-1},k_i}, $\mathrm{P}_{k_i,k_{i+1}}$ and $\mathrm{P}_{k_{i-1},k_{i+1}}$. Note that, since outliers are dropped out by the simulated annealing scheme, only a fraction of the loop chain matches contribute to 3D points in the final map $\mathcal{M}_i$ (see Subsec. 2.2.2).

To select a frame as a new keyframe, an heuristic based on the feature flow motion is used. Given the latest two keyframe $I_{k_{i-1}}$ and I_{k_i} a new keyframe I_{k_i} is detected if there is a sufficient number of matches that show a strong displacement (also named *unfix* points) w.r.t. both old keyframes, i.e.

$$\frac{\sum\limits_{i \in \{\mathbf{x}_o, \mathbf{x}_n\}} T(\| \mathbf{x}_o^{(i)} - \mathbf{x}_n^{(i)} \|_2 > \delta_f)}{|\{\mathbf{x}_o, \mathbf{x}_n\}|} > \eta \tag{2.8}$$

where $\{\mathbf{x}_o, \mathbf{x}_n\}$ is the matching set between old ($o = \{k_{i-1}, k_i\}$) and new ($n = \{k_{i+1}\}$)) frames, with $|\{\mathbf{x}_o, \mathbf{x}_n\}|$ its cardinality; T is an indicator function that outputs 1 if its predicate is true, 0 otherwise, δ_f is the threshold used to select strong flow matches and η the minimal percentage of *unfix* points. This heuristic comes from the well known observation that correspondences with high flow (that typically represent close scene points) provide better results during triangulation (see also Subsec. 2.3.2 and Fig. 2.11).

2.2.2 Simulated Annealing Optimization

When a new keyframe triplet $T_i = \{I_{k_{i-1}}, I_{k_i}, I_{k_{i+1}}\}$ is detected, chain matches $\mathcal{C}_i$ are computed and a Simulated Annealing optimization is performed in order to estimate both the updated 3D map $\mathcal{M}_i$ and the refined new keyframe pose $P_{k_{i+1}}$, while guaranteeing compatibility with the previous estimations.

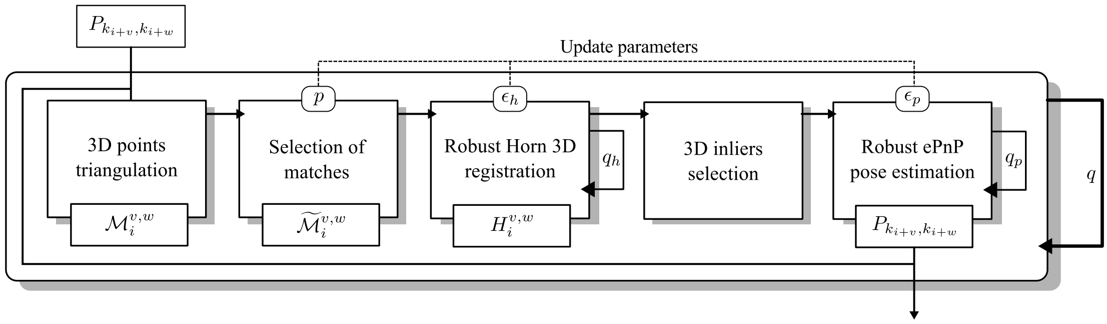


FIGURE 2.4: Block diagram of the SAMSLAM simulated annealing optimization executed on each keyframe triplet T_i.

In Fig. 2.4 a block diagram representing the implemented pipeline is presented. The simulated annealing approach starts by associating to each image pair $(I_{k_{i+v}}, I_{k_{i+w}})$, with $v, w \in \{-1, 0, 1\}$ and $v < w$, an initial 3D map $\mathcal{M}_i^{v,w}$, obtained by triangulation of the matches $(\mathbf{x}_{k_{i+v}}, \mathbf{x}_{k_{i+w}}) \in \mathcal{C}_i$.

To be able to perform triangulation of the 3D map, an initial estimates of the relative position $\mathrm{P}_{k_{i+v},k_{i+w}}$ between each image pair is needed. In the initialization phase, when optimizing the first keyframe triplet $T_1 = \{I_{k_0}, I_{k_1}, I_{k_2}\}$ poses are obtained from the essential matrix decomposition. In fact, given the fundamental matrices F_{k_0,k_1}, F_{k_1,k_2} and F_{k_0,k_2}—already computed to discard outlier matches—exploiting the calibration matrix $\mathcal{K}$ it's possible to obtain the essential matrices E_{k_0,k_1}, E_{k_1,k_2} and E_{k_0,k_2} as

$$E_{k_a,k_b} = \mathcal{K}^{\mathrm{T}} F_{k_a,k_b} \mathcal{K}, \quad a, b \in \{0, 1, 2\} \tag{2.9}$$

Then given the relation $E = [\mathbf{t}]_\times R$ (Hartley and Zisserman, 2004), rotation R and translation $\mathbf{t}$ up to a scale factor ambiguity are extracted for each image pair. On the other hand, during the iterations for triplets T_i with $i \geq 2$, poses are computed w.r.t. the last available 3D map $\mathcal{M}_{i-1}$ exploiting 2D/3D correspondences, solving an exterior orientation problem.

In our Simulated Annealing approach, after initial triangulation, one of the three maps $\mathcal{M}_i^{v,w}$ is used as reference $\mathcal{M}_i^{\mathrm{ref}}$ for all iteration. Inconsistent 3D points with negative depths in any of the three associated stereo configurations $(I_{k_{i+v}}, I_{k_{i+w}})$ are removed as well as points far from any of the corresponding camera centers, since the uncertainty in point localization increases with distance. The proportion p of points discarded by this latter constraint linearly decreases with the iteration

q since a more refined model is obtained as the iterations go on. In our experiments p is made to decrease from 30% to 1%. Remaining points in the resulting sub-map $\widetilde{\mathcal{M}}_i^{v,w}$ are registered to the reference sub-map $\widetilde{\mathcal{M}}_i^{\mathrm{ref}}$ through the Horn method (Horn, 1987), made robust to outliers by RANSAC.

Absolute orientation problem

The *absolute orientation* problem deals with the estimation of a 7 DoF transform—composed by a rotation, a translation and a scale factor—that can map a set of three-dimensional points onto another point set expressed in a different coordinate frame.

Given two set of three-dimensional points S_a and S_b expressed in different coordinate frames, and the relation that links a point $\mathbf{X}_a \in S_a$ with $\mathbf{X}_b \in S_b$, the Horn method is used to find the transformation

$$H = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \tag{2.10}$$

such that

$$\tilde{\mathbf{X}}_b = H\tilde{\mathbf{X}}_a \tag{2.11}$$

where s is a scalar factor, R a rotation matrix, $\mathbf{t}$ a translation vector and $\tilde{\mathbf{X}}$ is the homogeneous representation of the 3D point $\mathbf{X}$. Given that a 7 DoF transform have to be estimated, at least three non collinear 3D points, which rise nine linearly independent constraints, have to be used.

Horn (1987) presented a closed form solution for the absolute orientation problem that is articulated in four main step: (1) computation of the centroids of S_a and S_b; (2) estimation of R through quaternion representation; (3) calculation of the scale factor as ratio of the root-mean-square deviation of the two set w.r.t. their centroid; (4) translation estimation as difference of scaled and rotated centroid.

At first centroids are computed as

$$\bar{\mathbf{X}}_a = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}_{a,i} \quad \text{and} \quad \bar{\mathbf{X}}_b = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}_{b,i} \tag{2.12}$$

where N is the cardinality of S_a and S_b. Then 3D points are expressed w.r.t the centroid in the new coordinate

$$\mathbf{X'}_{a,i} = \mathbf{X}_{a,i} - \bar{\mathbf{X}}_a \quad \text{and} \quad \mathbf{X'}_{b,i} = \mathbf{X}_{b,i} - \bar{\mathbf{X}}_b \tag{2.13}$$

Rotation can be obtained by finding the unit quaternion $\dot{\mathbf{q}}^{\dagger}$ that maximize

$$\dot{\mathbf{q}}^{\dagger} = \arg\max_{\dot{\mathbf{q}}} \sum_{i=1}^{N} (\dot{\mathbf{q}}\dot{\mathbf{X}}'_{a,i}\dot{\mathbf{q}}^{\star}) \cdot \dot{\mathbf{X}}'_{b,i} \tag{2.14}$$

where $\dot{\mathbf{X}} = 0 + ix + jy + kz$ is the purely imaginary quaternion of the vector $\mathbf{X} = [x, y, z]^{\mathrm{T}}$, $\dot{\mathbf{q}}^{\star}$ is the conjugate of $\dot{\mathbf{q}}$ and $\dot{\mathbf{q}}\dot{\mathbf{X}}\dot{\mathbf{q}}^{\star}$ express the rotation of $\dot{\mathbf{X}}$ w.r.t. $\dot{\mathbf{q}}$. The operation $(\cdot)$ is the quaternion dot product. In other words, we want to find the rotation (expressed with an unit quaternion) such that $\dot{\mathbf{X}}'_{b,i}$ and the rotated $\dot{\mathbf{X}}'_{a,i}$ are parallel.

Equation (2.14) could be rewritten as

$$\dot{\mathbf{q}}^{\dagger} = \arg\max_{\dot{\mathbf{q}}} \sum_{i=1}^{N} (\dot{\mathbf{q}}\dot{\mathbf{X}}'_{a,i}) \cdot (\dot{\mathbf{X}}'_{b,i}\dot{\mathbf{q}}) \tag{2.15}$$

where

$$\dot{\mathbf{q}}\dot{\mathbf{X}}'_{a,i} = \begin{bmatrix} 0 & -x'_{a,i} & -y'_{a,i} & -z'_{a,i} \\ x'_{a,i} & 0 & z'_{a,i} & -y'_{a,i} \\ y'_{a,i} & -z'_{a,i} & 0 & x'_{a,i} \\ z'_{a,i} & y'_{a,i} & -x'_{a,i} & 0 \end{bmatrix} \dot{\mathbf{q}} = \mathcal{R}_{a,i}\dot{\mathbf{q}} \tag{2.16}$$

and

$$\dot{\mathbf{X}}'_{b,i}\dot{\mathbf{q}} = \begin{bmatrix} 0 & -x'_{b,i} & -y'_{b,i} & -z'_{b,i} \\ x'_{b,i} & 0 & -z'_{b,i} & y'_{b,i} \\ y'_{b,i} & z'_{b,i} & 0 & -x'_{b,i} \\ z'_{b,i} & -y'_{b,i} & x'_{b,i} & 0 \end{bmatrix} \dot{\mathbf{q}} = \mathcal{R}_{b,i}\dot{\mathbf{q}} \tag{2.17}$$

Then (2.15) could be rewritten as

$$\dot{\mathbf{q}}^{\dagger} = \arg\max_{\dot{\mathbf{q}}}(\dot{\mathbf{q}}^{\mathrm{T}}\mathcal{R}\dot{\mathbf{q}}) \tag{2.18}$$

where $\mathcal{R} = \sum_{i=1}^{N} \mathcal{R}_{a,i}^{\mathrm{T}}\mathcal{R}_{b,i}$. $\dot{\mathbf{q}}^{\dagger}$ is the eigenvector associated to the most positive eigenvalue of $\mathcal{R}$.

The scale factor s is easily computed with the ratio of the root-mean-square distance of the points respect their centroid as

$$s = \sqrt{\frac{\sum\limits_{i=1}^{N} \parallel \mathbf{X'}_{b,i} \parallel^2}{\sum\limits_{i=1}^{N} \parallel \mathbf{X'}_{a,i} \parallel^2}} \tag{2.19}$$

Finally, once obtained the rotation $R(\dot{\mathbf{q}}^{\dagger})$ and the scale factor s, the translation is estimated as

$$\mathbf{t} = \bar{\mathbf{X}}_b - sR(\dot{\mathbf{q}}^{\dagger})\bar{\mathbf{X}}_a \tag{2.20}$$

Note that $R(\dot{\mathbf{q}}^{\dagger})$ express the change of representation from the quaternion $\dot{\mathbf{q}}^{\dagger}$ to the rotation matrix R. ■

At each iteration q_h, a 7 DoF (scale, rotation and translation) transformation $H_i^{v,w}$ is estimated by the Horn method in a RANSAC framework, where the sampling set is a subset of the whole validation set and contains only the 25% of points in $\widetilde{\mathcal{M}}_i^{v,w}$ with maximal flow displacement. This is beneficial to map accuracy, since high disparity matches are characterized by a better localization in 3D space.

To evaluate the inlier set, 3D points of $\widetilde{\mathcal{M}}_i^{\mathrm{ref}}$ are mapped to $\widetilde{\mathcal{M}}_i^{v,w}$ and then projected to the corresponding images $I_{k_{i+v}}$ and $I_{k_{i+w}}$ using the same camera matrices employed for triangulation. The distances between the projected points and the effective matches are evaluated and inliers are selected if an error inferior to ϵ_h is found. Once completed all RANSAC iterations the greatest inlier set for the pair $(I_{k_{i+v}}, I_{k_{i+w}})$ is kept. The Horn registration is executed for each pair independently, so to obtain three set of 3D inliers related to the three maps $\widetilde{\mathcal{M}}_i^{i-1,i}$, $\widetilde{\mathcal{M}}_i^{i,i+1}$ and $\widetilde{\mathcal{M}}_i^{i-1,i+1}$. The final inlier set is composed by the 3D points that are inlier in all three maps.

The inlier threshold value ϵ_h linearly decreases with the iteration q, from 20 to 4 pixels in our experiments. This progressive reduction, used also for the pose estimation (see forward), implements the simulated annealing scheme. Threshold changes smoothly so to refine the results as iterations proceed and, in this way, the system is able to handle even noisy data (e.g. distant 3D points) without fall in wrong estimation.

After the Horn registration step, the 3D inlier set for iteration q and the related 2D feature points on each image are given as input to the pose estimation step, implementing the EPnP algorithm wrapped in a RANSAC scheme.

PnP: Persective n Point problem

The PnP problem deals with the estimation of the pose of a calibrated camera from n 2D-3D correspondences. Lepetit et al. (2009) propose a closed form solution that achieves good results with a computational complexity of $O(n)$.

The main idea is to base the estimation on four control points $\mathbf{c}_j^w$ with $j = 1, \ldots, 4$ obtained from the n input 3D points $\mathbf{X}_i^w$ where $i = 1, \ldots, n$, expressed in the world coordinate frame, such as

$$\mathbf{X}_i^w = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^w, \quad \text{with} \quad \sum_{j=1}^{4} \alpha_{ij} = 1 \tag{2.21}$$

where α_{ij} are homogeneous barycentric coordinate. A similar relation holds if points are expressed in the camera frame as

$$\mathbf{X}_i^c = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^c \tag{2.22}$$

In the proposed solution, instead than estimates rotation R and translation $\mathbf{t}$ minimizing the re-projection error, authors try to obtain the 3D points in the camera coordinate frame and compute R and $\mathbf{t}$ solving an absolute orientation problem.

To achieve such a results, at first α_{ij} values are computed from Eq. 2.21, once defined $\mathbf{c}_1^w$ as the centroid of the 3D data and the other control points in such a way that they form a basis aligned with the principal directions of the data.

Then, given the 2D projections $\mathbf{u}_i$ with $i = 1, \ldots, n$, it's possible to define

$$w_i \begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix} = \mathcal{K}\mathbf{X}_i^c = \mathcal{K} \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^c \tag{2.23}$$

where $\mathcal{K}$ is the known calibration matrix. In Eq. 2.23, the unknowns are the twelve values $\{[x_j^c, y_j^c, z_j^c]\}_{j=1,\ldots,4}$ of the four control point and the n scalar factors w_i. It's easy to figure out that

$$w_i = \sum_{j=1}^{4} \alpha_{ij} z_j^c \tag{2.24}$$

and substituting 2.24 back in 2.23 we can obtain

$$\sum_{j=1}^{4} \alpha_{ij} f_u x_j^c + \alpha_{ij}(u_c - u_i) z_j^c = 0 \tag{2.25}$$

$$\sum_{j=1}^{4} \alpha_{ij} f_v y_j^c + \alpha_{ij}(v_c - v_i) z_j^c = 0 \tag{2.26}$$

where $\mathbf{u}_i = [u_i, v_i]^{\mathrm{T}}$, (f_u, f_v) are the focal lengths and (u_c, v_c) is the camera center offset. Using at least six 2D/3D matches authors obtain twelve constraint in the form of equations 2.25 and 2.26, that can be rewritten as

$$\mathrm{M}\bar{\mathbf{c}} = 0 \tag{2.27}$$

where $\bar{\mathbf{c}} = [\mathbf{c}_1^{c^{\mathrm{T}}}, \mathbf{c}_2^{c^{\mathrm{T}}}, \mathbf{c}_3^{c^{\mathrm{T}}}, \mathbf{c}_4^{c^{\mathrm{T}}}]^{\mathrm{T}}$.

Solving Eq. 2.27 to obtain $\mathbf{c}_j^c$ and then using Eq. 2.22, 3D points $\mathbf{X}_i^c$—expressed in camera coordinate frame—could be retrieved and used as input for an absolute orientation problem to compute the rotation R and translation $\mathbf{t}$. ■

As for the Horn registration step, also in the pose computation the RANSAC sampling set is limited to the 25% of points with greater flow; Moreover the reprojection error threshold ϵ_p—used to define inliers—linearly decreases with the iteration q, from 5 to 3 pixels in the experiments.

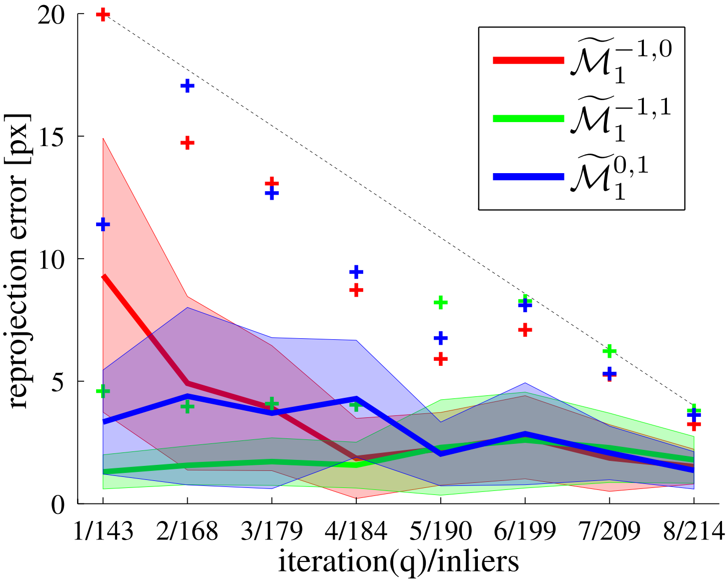


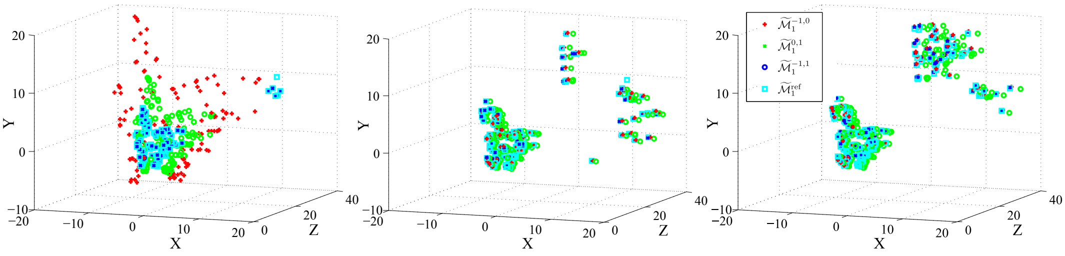
FIGURE 2.5: Plot of the reprojection errors as the iterations proceed for the first keyframe triplet T_1 of the *Monk* video sequence. The reference map is $\widetilde{\mathcal{M}}_1^{\text{ref}} = \widetilde{\mathcal{M}}_1^{-1,1}$. Solid lines indicate the average reprojection errors, while bands show the behaviour of the standard deviation. Marks represents the maximal values and the dashed gray line is the RANSAC linear threshold bound ϵ_h. (Best viewed in color)

Finally relative poses $\mathrm{P}_{k_{i+v},k_{i+w}}$ are recomputed on the whole inlier set of iteration q. These estimates will replace the previous ones for the next iteration.

Figure 2.5 shows an example of the simulated annealing scheme on the first keyframe triplet T_1 of the *Monk* video sequence (see Subsec. 2.2.3). Figure 2.6 shows the corresponding 3D maps $\widetilde{\mathcal{M}}_i^{v,w}$ for different iterations q. The average reprojection error gradually decreases for each image pair $(I_{k_{i+v}}, I_{k_{i+w}})$ to less than 2 pixels, while the number of 3D point inliers increases and the 3D registration improves. Note that the first iteration $q = 1$ of the first keyframe triplet T_1 is the most time consuming in terms of RANSAC iterations with $q_h, q_p \simeq 500$, while in the other cases $q_h, q_p \simeq 50$ since only refinements are required. The RANSAC-based design can be useful to define efficient parallel and multi-threaded implementations of the Simulated Annealing scheme.

2.2.3 Results

In order to evaluate the performance of our monocular SLAM approach, two different experiments have been carried out: A quantitative direct measure of the odometry accuracy, and an indirect evaluation of the 3D reconstruction quality of



(A) (B) (C)

FIGURE 2.6: The 3D maps $\widetilde{\mathcal{M}}_i^{v,w}$ and the reference map $\widetilde{\mathcal{M}}_1^{\mathrm{ref}}$ at iterations $q = 1, 5, 8$ for the keyframe triplet T_1 of the *Monk* video sequence. (Best viewed in color)

an object acquired using a structured-light framework (see Ch. 3 for details on the structured-light reconstruction method).

Three different indoor video sequences with a resolution of 640 × 480 pixels and about 800 frames have been used in the former case — see Table 2.1. The first two sequences (*Desk1* and *Desk2*) explore the same desktop environment as the camera undergoes two different motions, while the last sequence (*Monk*) contains an object scanned by a laser fan projector. This last sequence is also used for the indirect evaluation through 3D reconstruction. A known planar pattern is included in the background of all test sequences to recover accurate ground-truth poses exploiting the homography matrices that exist between the planar pattern and its image through decomposition in a rotation and translation (Fanfani and Colombo, 2013).

TABLE 2.1: Distance error of the camera centres with respect to the ground-truth length.

	 Desk1	 Desk2	 Monk
Mean(%)	1.29	0.93	0.48
Std(%)	0.63	0.30	0.23
Max(%)	3.05	2.39	1.21
Min(%)	0.24	0.29	0.15
Length(cm)	71.31	100.35	74.90

Table 2.1 shows the Euclidean distance error of the camera centers normalized to the ground-truth path length, while corresponding tracks are shown in Fig. 2.7. Since in monocular setup the scale information is lost, camera centers have been registered to the known ground-truth metric scale using the Horn method. SAMSLAM error is about 1% on average, i.e. less than 1 cm for a track length of 100 cm, and trajectories are well aligned.

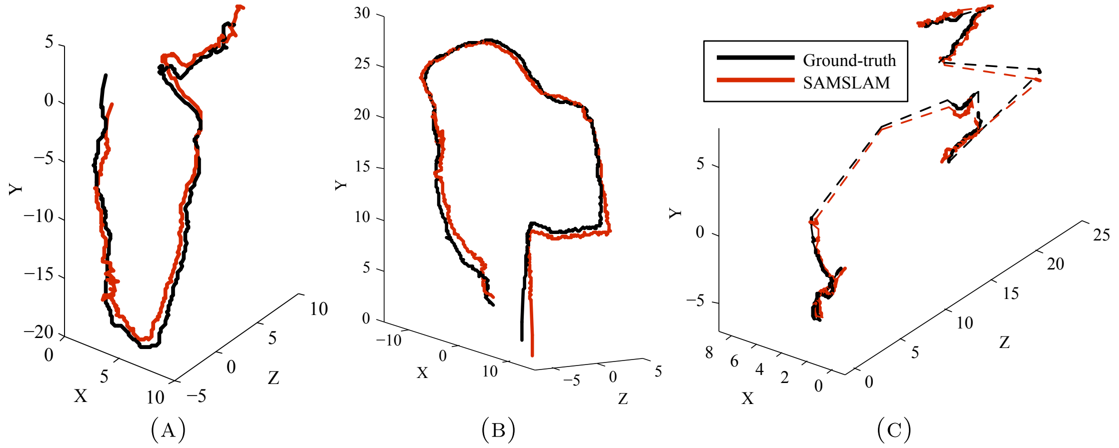


FIGURE 2.7: Track comparison for the video sequences *Desk1* (a), *Desk2* (b) and *Monk* (c). Dashed lines for the *Monk* sequence indicate that no ground-truth has been provided. (Best viewed in color)

For the 3D reconstruction test on the *Monk* sequence, a device equipped with a camera and a laser fan projector kept in fixed relative positions is used in order to get an accurate 3D model.

Figure 2.8c shows the 3D reconstruction obtained with the structured-light method and the SAMSLAM motion estimates. The SAMSLAM reconstruction is qualitatively good and shows a mean euclidean error of $0.105 * 10^{-3}$cm with standard deviation of $0.112 * 10^{-3}$cm and a maximum error of $1.616 * 10^{-3}$cm w.r.t. the ground-truth reconstruction obtained with motion estimation accurately computed exploiting the planar pattern.

SAMSLAM vs Local Bundle Adjustment

In Table 2.2 results of comparison between the proposed simulated annealing optimization and a three-frame version of the standard Bundle Adjustment algorithm (reffered to in the following as Local Bundle Adjustment, LBA) is presented.

Given a keyframe triplet, loop matches among the three images are computed. For the first camera the rotation is set to the identity and the translation to the null

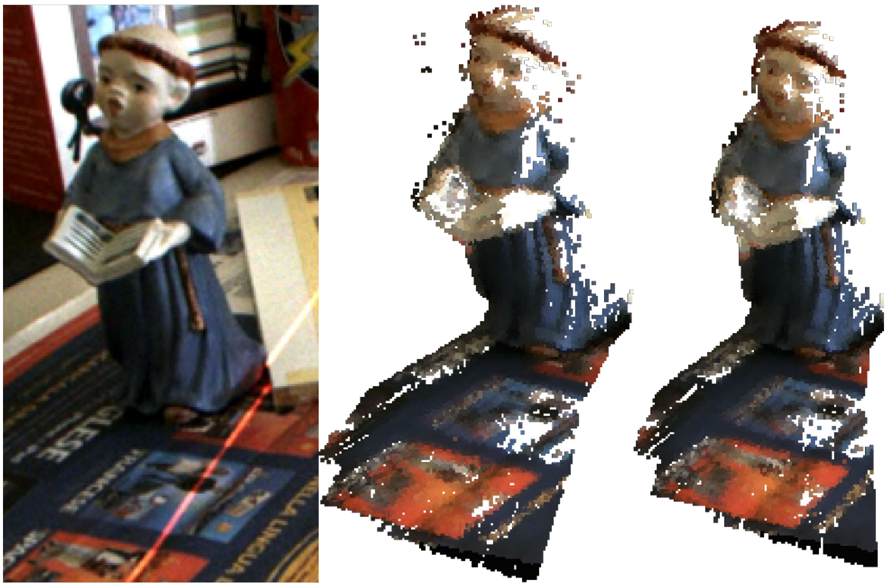


FIGURE 2.8: 3D Reconstructions using the structured-light approach: (a) a frame of the *Monk* sequence zoomed on the object to be reconstruced; (b) ground-truth reconstruction; (c) reconstruction using SAMSLAM motion estimates.

vector. The second camera is obtained by decomposition of the essential matrix relating the first and second views. The third camera is computed in two distinct ways: (i) using the result of decomposition of the essential matrix between the first and third views or (ii) employing PnP on 3D points computed from the first pair and the known 2D correspondences.

The camera matrices initialized as above, the 3D points and the 2D matches are passed to both the SAMSLAM optimization and to the LBA to be refined. Results are finally computed—by evaluating the re-projection error—as reported in Tab. 2.2

Given the obtained results, it's not clear which of the two methods is better. To get a deeper insight into the performances of the methods we compute a normalized score $s_{m,i}$ as

$$s_{m,i} = \frac{|e_{m,i} + \epsilon - \min(e_{S,i}, e_{B,i})|}{|e_{S,i} + \epsilon - \min(e_{S,i}, e_{B,i})| + |e_{B,i} + \epsilon - \min(e_{S,i}, e_{B,i})|} \tag{2.28}$$

where $i = \{0, 1, 2\}$ is the image index, $m = \{S, B\}$ represent the method used, i.e. SAMSLAM or LBA, and ϵ is a small constant value ($\epsilon = 0.001$ in this test) used

Table 2.2: Mean reprojection errors (pxl) after SAMSLAM (SAM) or Local Bundle Adjustment (LBA) keyframe triplet optimizations.

Sequ.	Image 0		Image 1		Image 2		Init.
	SAM	LBA	SAM	LBA	SAM	LBA	
Desk1	**0.117**	0.204	0.318	**0.211**	0.199	**0.195**	Ess
	0.124	0.204	0.345	**0.225**	0.259	**0.189**	PnP
Desk2	**0.190**	0.264	**0.353**	0.609	**0.149**	0.202	Ess
	0.254	0.261	**0.355**	0.683	**0.141**	0.202	PnP
Monk	**0.165**	0.196	**0.261**	0.386	**0.119**	0.176	Ess
	0.209	**0.162**	0.726	**0.282**	0.257	**0.144**	PnP

to avoid strong score differences in the case of very similar errors.

For example, for the 0.117 in the left top corner of Table 2.2 we will obtain

$$s_{S,0} = \frac{|0.117 + \epsilon - \min(0.117, 0.204)|}{|0.117 + \epsilon - \min(0.117, 0.204)| + |0.204 + \epsilon - \min(0.117, 0.204)|} = \dots$$
$$\dots = 0.011236 \quad \text{with} \quad \epsilon = 0.001 \tag{2.29}$$

Note also that $s_{m_A,i} \to 0$ if $e_{m_A,i} \ll e_{m_B,i}$, $s_{m_A,i} \to 1$ if $e_{m_A,i} \gg e_{m_B,i}$, $s_{m_A,i} = 0.5$ if $e_{m_A,i} = e_{m_B,i}$, and obviously $s_{m_A,i} + s_{m_B,i} = 1$. Mean scores of the normalized error values are computed for both methods globally and w.r.t. the chosen initialization (essential matrix decomposition or PnP) and reported in Tab. 2.3.

Table 2.3: Mean values of the normalized error scores obtained from Tab. 2.2 usign Eq. 2.28 for all values and w.r.t. both kind of initialization—essential matrix decomposition (Ess) and PnP.

Init.	SA	LBA
Ess	**0.021397**	0.786029
PnP	0.565420	**0.434580**
Overall	**0.389695**	0.610305

As can be seen in Tab. 2.3, our method obtains better results with initialization of the third camera by essential matrix decomposition, while LBA performs slightly better in the PnP case. Anyway in average, our method produce better estimation.

This behavior could be due to two main aspects: First of all, SAMSLAM optimization is wrapped in a RANSAC framework, so noisy points are excluded from the computation while LBA try to minimize the error over the whole point set. Secondly, in our method, camera poses and the 3D maps are re-computed at each iteration discarding the previous solutions, while LBA proceeds with iterative refinement: this behavior leads to erroneous estimation if the initial solution is not good enough—e.g., using the essential matrix decomposition the translation scale factor is not congruent among all cameras, leading to a poor initialization.

2.3 Selective Stereo SLAM

In this Section a stereo SLAM system named Selective Stereo SLAM (SSLAM) is introduced. This approach employs a stereo camera device to obtain at each time a pair of synchronized images of the same scene. The main idea is to make use of only highly reliable data in the estimation process, which is reflected mainly in the feature matching scheme and the choice of good keyframes.

The feature matching process is the main source of noise in a SLAM system. Wrong matches could lead to erroneous estimation, which can be only partially corrected using robust outlier rejection strategies. To limit as much as possible the introduction of errors in early processing stages, we choose to employ an accurate and computationally heavy matching strategy instead of a less accurate solution. In particular, a robust loop chain matching scheme is adopted, improving upon VISO2-S (Geiger et al., 2011), but using a more robust detector-descriptor pair with respect to this latter system. While an additional computation time is required, the pose estimation gains in accuracy, thus avoiding the introduction of strong noises and the need of further global optimization or loop closure steps. In addition, the robust matching adopted lets the system find correspondences not only between close frames, but also in images with higher disparity, that can be an issue for approaches based on tracking.

The other aspect mainly characterizing SSLAM is the selection of the keyframes used as base references for the measurement of the 3D landmark positions and for the camera trajectory computation. Keyframes are selected only if a strong feature temporal flow is detected. This idea arises from the observation that errors propagate also from the uncertainty of the three-dimensional points, which is higher for distant points corresponding to low temporal flow disparity matches in the images. The proposed strategy can be more stable and effective with respect to using a threshold on the average flow disparity (Lee et al., 2011) or a constant keyframe interleaving (Nistér et al., 2004). Moreover, evaluating 2D measures such as the feature flow leads to a more robust keyframe selection compared to approaches that evaluate in 3D space the distance among frames (Geiger et al., 2011).

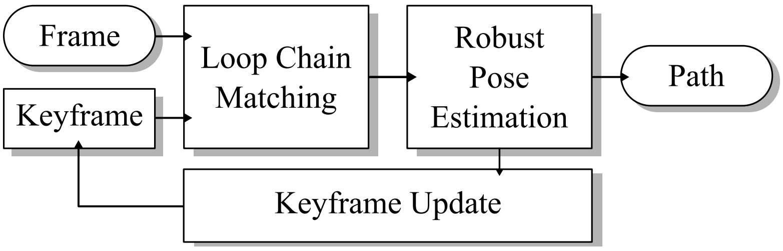


Figure 2.9: Block diagram for SSLAM.

2.3.1 SSLAM Overview

Given a calibrated and rectified stereo sequence $S = \{f_t\}$, where the frame $f_t = (I_t^l, I_t^r)$ is composed by the left I_t^l and right I_t^r input images taken at time $t \in \mathbb{N}$, SSLAM alternates between two main steps (see Fig. 2.9). The former step matches keypoints between the last keyframe f_i and the current frame f_j, while the latter estimates the relative camera pose $\mathrm{P}_{i,j} = [\mathrm{R}_{i,j}|\mathbf{t}_{i,j}] \in \mathbb{R}^{3\times4}$, where $\mathrm{R}_{i,j} \in \mathbb{R}^{3\times3}$ is the rotation matrix and $\mathbf{t}_{i,j} \in \mathbb{R}^3$ is the translation vector. If the new pose is successfully estimated and sufficient temporal flow disparity is detected between f_j and f_i, the frame f_j is updated as the new keyframe.

Assuming that $\mathrm{R}_{i,i} = \mathrm{I}$ and $\mathbf{t}_{i,i} = \mathbf{0}$ (where I and $\mathbf{0}$ are respectively the identity matrix and the null vector) the *absolute pose* at time n is defined as $\mathrm{P}_n = \mathrm{P}_{0,n}$. P_n can be computed by concatenating the poses $\mathrm{P}_{0,0}, \mathrm{P}_{0,k}, \ldots, \mathrm{P}_{i,j}\mathrm{P}_{j,n}$, where time steps $0 < k < \ldots < i < j$ belong to accepted keyframes and $n > j$ is the current frame.

Hereafter the loop chain matching scheme (Subsec. 2.3.2) is described. Subsection 2.3.3 presents the pose estimation algorithm and the keyframe selection strategy. Finally experimental results (Subsec. 2.3.4) are given, based on the KITTI and NewCollege datasets; Moreover examples of estimation in underwater environment are shown.

2.3.2 Loop-chain Matching

The proposed loop chain matching draws inspiration from the *circle match* of VISO2-S (Geiger et al., 2011), as the candidate correspondences should be consistent among the four image pairs (I_i^l, I_i^r), (I_i^l, I_j^r), (I_i^r, I_j^r), (I_j^l, I_j^r). However instead of a fast but less accurate keypoint detector and descriptor based on simple image filters, a robust detector and descriptor pair is used. This also avoids using the two step matching strategy employed by VISO2-S to further refine correspondences, and permits achieving longer and more stable keypoint tracks, crucial for the pose estimation, without re-initialization issues and keypoint losses occurring with tracking strategies such as KLT.

As for the SAMSLAM method (see Sec. 2.2), the HarrisZ detector is used to extract robust and stable corner features in the affine scale-space on the images $I_i^l, I_i^r, I_j^l, I_j^r$; The sGLOH descriptor with a Nearest Neighbour matching on the L_1 distance is used instead to obtain the candidate correspondences between image pairs (I_i^l, I_i^r), (I_i^l, I_j^r), (I_i^r, I_j^r), (I_j^l, I_j^r) after spatial and temporal constraints have been imposed to refine the candidates matches (see hereafter).

Let $\mathbf{x}_s^d = [x_s^d,\, y_s^d]^{\mathrm{T}} \in \mathbb{R}^2$, $d \in \{l, r\}$, $s \in \{i, j\}$ be a point in the image I_s^d. A *spatial* match $(\mathbf{x}_s^l, \mathbf{x}_s^r)$ between the images on the same frame is computed exploiting the stereo epipolar constraints imposed by the calibration

$$|x_s^l - x_s^r| < \delta_x \tag{2.30}$$

$$|y_s^l - y_s^r| < \delta_y \tag{2.31}$$

where δ_y is the error band allowed by epipolar rectification and δ_x is the maximum allowed disparity (i.e. the corresponding stereo point must lie inside a $\delta_x \times \delta_y$ rectangular window) with $\delta_y << \delta_x$.

In the case of a *temporal* match $(\mathbf{x}_i^d, \mathbf{x}_j^d)$ between corresponding images at different times, the flow restriction

$$\parallel \mathbf{x}_i^d - \mathbf{x}_j^d \parallel < \delta_r \tag{2.32}$$

is taken into account, where δ_r is the maximum flow displacement (i.e. the corresponding point in the next frame must lie inside a circular window of radius δ_r).

Only matches that form a *loop chain*

$$\mathcal{C} = \big((\mathbf{x}_i^l, \mathbf{x}_i^r), (\mathbf{x}_i^l, \mathbf{x}_j^l), (\mathbf{x}_j^l, \mathbf{x}_j^r), (\mathbf{x}_i^r, \mathbf{x}_j^r)\big) \tag{2.33}$$

are retained (see Fig. 2.10); however, some outliers can still be present. For this reason, each matching pair of the loop chain $\mathcal{C}$ is further filtered by RANSAC to refine the matches. These four RANSAC runs have an almost immediate convergence due to the high presence of inliers. Only loop chains whose all pair matches survive to the four RANSACs are finally collected into the set $C_{i,j} \subseteq \{\mathcal{C}\}$.

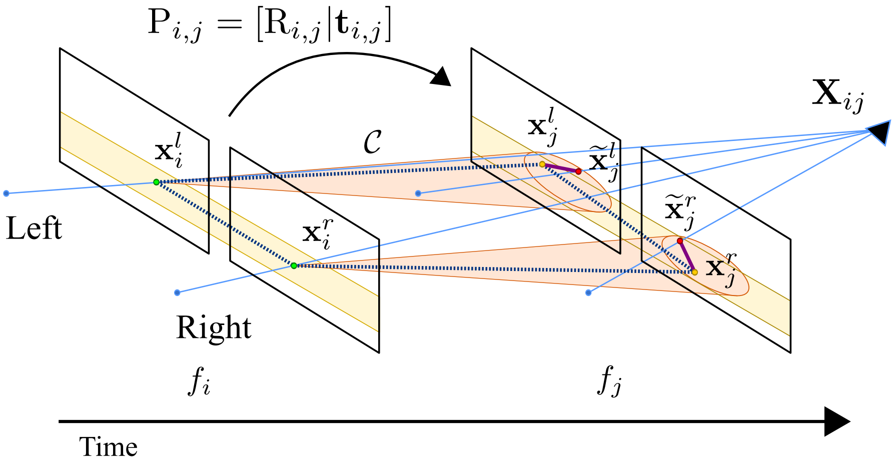


FIGURE 2.10: (Best viewed in color) Keypoint matches between the keyframe f_i and the new frame f_j must satisfy the spatial constraint imposed by the epipolar rectification (yellow band) as well as the temporal flow motion restriction (orange cone). Furthermore, the four matching points must form a loop chain $\mathcal{C}$ (dotted line). In the ideal case, points $\mathbf{x}_j^l$, $\mathbf{x}_j^r$ in frame f_j must coincide with the projections $\widetilde{\mathbf{x}}_j^l$, $\widetilde{\mathbf{x}}_j^r$ of $\mathbf{X}_{i,j}$ obtained by triangulation of $\mathbf{x}_i^l$, $\mathbf{x}_i^r$ in f_i in order for the chain $\mathcal{C}$ to be consistent with the pose $\mathrm{P}_{i,j}$. However, due to data noise, in the real case it is required that the distances $\parallel \widetilde{\mathbf{x}}_j^l - \mathbf{x}_j^l \parallel$ and $\parallel \widetilde{\mathbf{x}}_j^r - \mathbf{x}_j^r \parallel$ are minimal.

2.3.3 Incremental Pose Estimation

The relative pose $\mathrm{P}_{i,j}$ between f_i and f_j is estimated in the second step of the SSLAM approach (see again Fig. 2.10). The 3D point $\mathbf{X}_{i,j}$ corresponding to the match pair $(\mathbf{x}_i^l, \mathbf{x}_i^r)$ in keyframe f_i can be estimated by triangulation (Hartley and Zisserman, 2004), since the intrinsic and extrinsic calibration parameters of the stereo system are known—in particular, we use the iterative linear triangulation method by Hartley and Sturm (1997).

Let $\widetilde{\mathbf{x}}_j^l$ and $\widetilde{\mathbf{x}}_j^r$ be the projections of $\mathbf{X}_{i,j}$ onto frame f_j, according to the estimated relative pose $\mathrm{P}_{i,j} = [\mathrm{R}_{i,j}|\mathbf{t}_{i,j}]$. The distance

$$\sum_{C_{i,j} \subseteq \mathcal{C}, d \in \{l,r\}} \| \widetilde{\mathbf{x}}_j^d - \mathbf{x}_j^d \|^2 \tag{2.34}$$

among the matches of the chain set $C_{i,j}$ must be minimized, in order for the estimated pose $\mathrm{P}_{i,j}$ to be consistent with the data. In other words $\mathrm{P}_{i,j}$ is obtained minimizing the re-projection error on both the left and right images as

$$\mathrm{P}_{i,j} = \underset{\mathrm{R}_{i,j}, \mathbf{t}_{i,j}}{\operatorname{argmin}} \sum_{C_{i,j} \subseteq \mathcal{C}} ||\mathcal{K}(\mathrm{R}_{i,j}\mathbf{X}_{i,j} + \mathbf{t}_{i,j}) - \mathbf{x}_j^l||^2 + ||\mathcal{K}(\mathrm{R}_{i,j}\mathbf{X}_{i,j} + \mathbf{t}_{i,j} + \mathbf{b}) - \mathbf{x}_j^r||^2 \tag{2.35}$$

where $\mathcal{K}$ is the intrinsic calibration matrix and $\mathbf{b}$ is the vector accounting for the baseline between the left and right camera. Equation (2.35) is solved using a Gauss-Newton iterative refinement based on the computation of the Jacobian of the projection functions for both left and right cameras.

Due to the presence of outliers in $C_{i,j}$, a RANSAC test is run, where the number of outliers chain matches over $C_{i,j}$ exceeding a threshold value δ_t is minimized so that pose $\mathrm{P}_{i,j}$ be consistent with data.

At each iteration RANSAC estimates a candidate pose $\mathrm{P}_{i,j}$ using a minimal set of matches, i.e., 3 matches, in order to be robust to outliers (Fischler and Bolles, 1981). The candidate matches used to build the pose model $\mathrm{P}_{i,j}$ are sampled from the set of candidate matches $C_{i,j}$. The pose $\mathrm{P}_{i,j}$ is validated against the whole set of candidate matches $C_{i,j}$ and the best model found so far is retained. The process stops when the probability to get a better model is below some user-defined threshold value, and the final pose $\overline{\mathrm{P}}_{i,j}$ is refined on the whole set of inlier matches.

SSLAM filters the frame sequence according to the following observation. Image resolution provides a lower bound to the uncertainty of the position of the keypoints used in the matching process, although subpixel precision is used. Matches are triangulated to get the corresponding 3D point, and eventually estimate the relative pose between two temporal frames. Close frame matches have a low temporal flow disparity and the associated 3D point positions have an high uncertainty with respect to distant frames, due to the error propagation from the matches on the image planes. Only points with sufficient displacement can give information about the translational and rotational motion, as shown in Fig. 2.11. This idea

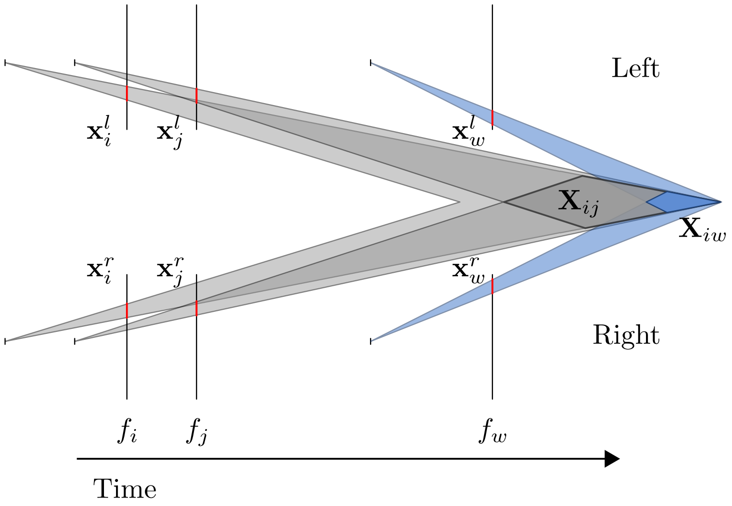


FIGURE 2.11: (Best viewed in color) The uncertainty of matches in the image planes is lower bounded by the image resolution (red) and it is propagated to the 3D points. In order to estimate the 3D point $\mathbf{X}_{i,j}$, by using close frames f_i and f_j, a low temporal disparity flow is present in the image planes, and the 3D point location $\mathbf{X}_{i,j}$ can assume an higher range $\mathbf{X}_{i,j}$ of values (dark gray quadrilateral). In the case of distant frames f_i and f_w, the possible locations $\mathbf{X}_{i,w}$ are more circumscribed (blue quadrilateral), for the same resolution limits.

is a straight generalization of the well-known baseline length issues related to the trade-off between reliable correspondence matching and accurate point triangulation (Hartley and Zisserman, 2004).

Exploiting this idea, SSLAM defines the subset $F_{i,j}$ of the set of chain matches $C_{i,j}$ for f_i and f_j containing points which can be considered fixed with respect to the temporal flow disparity and not effective for a good pose estimation:

$$F_{i,j} = \{\mathcal{C} \in C_{i,j} | T_d(\| \mathbf{x}_i^d - \mathbf{x}_j^d \| \leq \delta_f)\} \quad , \tag{2.36}$$

for a given threshold δ_f. In order for frame f_j to be accepted as new keyframe, the number of non-fixed matches between frames f_i and f_j must be sufficient according to a threshold δ_m:

$$1 - \frac{|F_{i,j}|}{|C_{i,j}|} > \delta_m \quad . \tag{2.37}$$

Indeed, if the estimation fails due to wrong matches or high noisy data, which practically leads to a final small RANSAC consensus set, the frame f_j is discarded and the next frame f_{j+1} is tested. We also tried to verify if the use of only non-fixed matches as input to RANSAC pose estimation can lead to better results but no improvements were found.

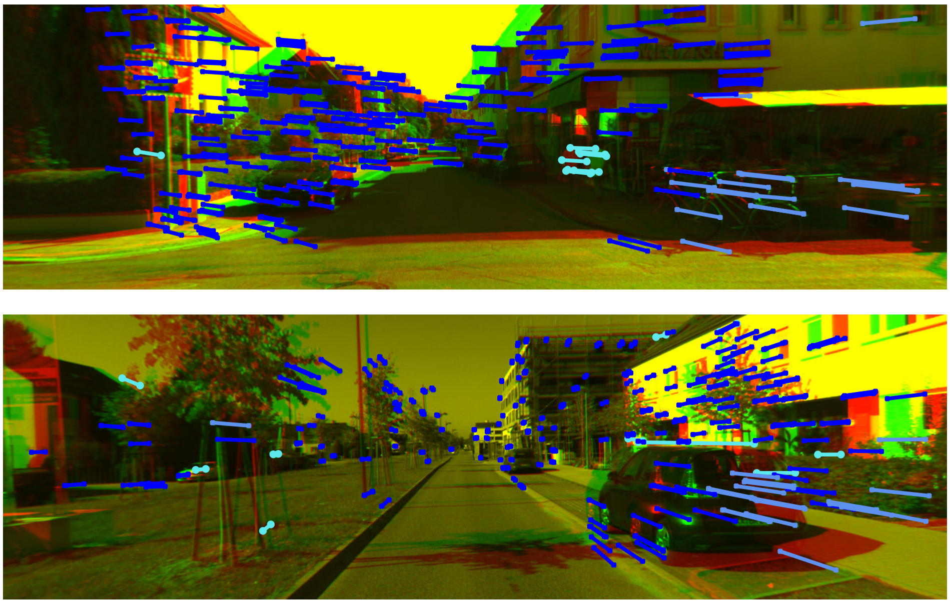


FIGURE 2.12: (Best viewed in color) Examples of successive keyframes retained according to the temporal flow for two different sequences of the KITTI dataset. The two temporal keyframes involved are superimposed as for anaglyphs, only images for the left cameras are shown. Good fixed and unfixed matches are shown in blue and light violet, respectively, while wrong correspondences are reported in cyan.

Examples of fixed point estimations are shown in Fig. 2.12. With respect to the average flow threshold—commonly employed by some SLAM systems— the proposed strategy is more stable and can handle better keyframe drops. As an example, referring to Fig. 2.12, the average flow in the top configuration is considerably higher than that of the bottom one. Lowering the threshold, to accept the bottom frame, would also include very low disparity frames (just consider to replace in the bottom frame the unfixed violet matches by twice the matches with half disparity). In this sense, SSLAM measure is more robust, so that both the frames shown in the figure are retained as keyframes. In analogy, our frame selection resembles RANSAC while the average flow is close to the least-square approach.

Finally, a pose smoothing constraint is added between frames, so that the current relative pose estimation $\mathrm{P}_{i,j}$ cannot abruptly vary from the previous $\mathrm{P}_{z,i}$, $z < i < j$. This is achieved by imposing that the relative rotation around the origin between the two incremental rotations $\mathrm{R}_{z,i}$ and $\mathrm{R}_{i,j}$ is bounded

$$|\mathbf{r}^{k}_{i,j}{}^{\mathrm{T}}\mathbf{r}^{k}_{z,i}| < \delta_{\theta_1} \tag{2.38}$$

where $\mathbf{r}^{k}_{a,b}$ is any k-th column of the rotation matrix $\mathrm{R}_{a,b}$. Optionally, in the case of strong constrained movement, like those of a car, a further constraint on the corresponding translation directions $\mathbf{t}_{z,i}$ and $\mathbf{t}_{i,j}$ can be added

$$\frac{|\mathbf{t}_{i,j}{}^{\mathrm{T}}\mathbf{t}_{z,i}|}{\parallel \mathbf{t}_{i,j} \parallel \parallel \mathbf{t}_{z,i} \parallel} < \delta_{\theta_2} \tag{2.39}$$

This last constraint can also resolve issues in the case of no camera movement or when moving objects crossing the camera path cover the scene.

2.3.4 Results

The KITTI vision benchmark suite (Geiger et al., 2012) and the New College sequence (Smith et al., 2009) were used to evaluate SSLAM.

Recently, the KITTI dataset has become a reference evaluation framework for SLAM systems. The dataset provides sequences recorded from car driving sessions on highways, rural areas and inside cities up to 80 km/h. The benchmark consists of 22 rectified stereo sequences from about 500 m to 5 km, taken at 10 fps with a resolution of 1241×376 pixels. Ground truth trajectories are available to users only for the first 11 sequences to train the parameters of the SLAM methods, while results should be submitted to the authors page for the remaining sequences to get a final detailed ranking. Translation and rotation errors normalized with respect to the path lengths and speeds are computed in order to rank the methods.

The New College dataset is made up by a very long sequence of 2.2 km for more than 50000 stereo rectified frames taken inside the Oxford New College campus using a Segway. Data were recorded at 20 fps with a resolution of 512×384 pixels. Although no (reliable) ground truth is available, the sequence consists of several different loops which can be used to qualitatively compare SLAM methods by visual inspection of their estimated paths. Unlike the KITTI dataset, data are recorded at a lower speed and the camera movements are less constrained, i.e., strong camera shakes are present.

Unless otherwise specified, for SSLAM we set $\delta_f = 55$ px, $\delta_m = 5\%$, $\delta_{\theta_1} = 15^\circ$ (see Subsec. 2.3.3). About the spatial and temporal constraints, the triplet $(\delta_r, \delta_x, \delta_y)$ is set to $(500, 300, 12)$ px in the case of the KITTI dataset and to $(100, 100, 12)$ px for the New College dataset, since videos are taken at different resolutions and baseline. The translation constraint is $\delta_{\theta_2} = 10^\circ$ for the KITTI dataset while it is not used for New College due to high camera shakes.

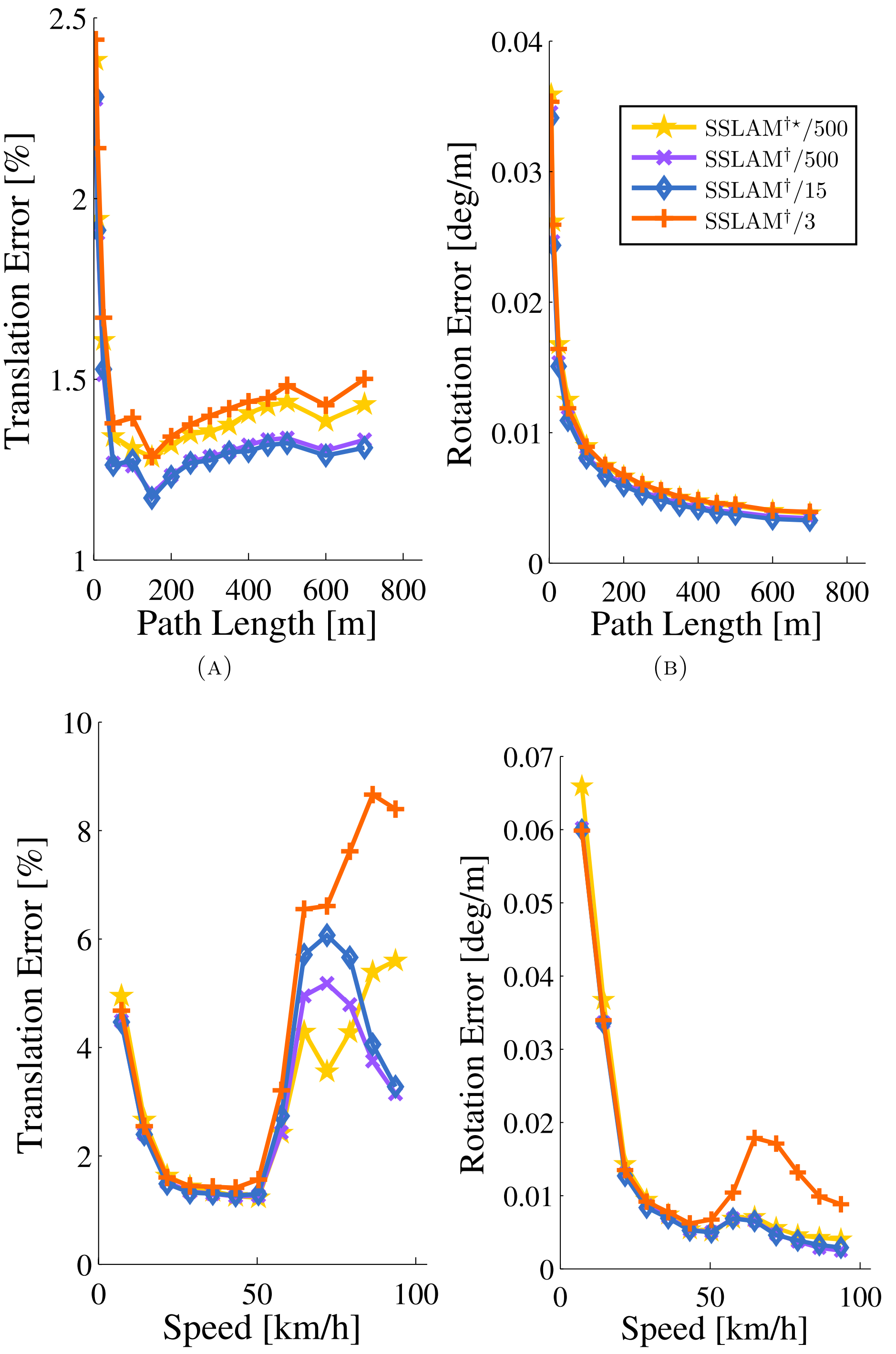
Furthermore, we tested SSLAM using keypoints detected at full and half resolution videos; in the latter case, the notation SSLAM† is used. In the case of SSLAM† less accurate keypoints are found, with bigger (normalized) feature patches, more sensitive to fast camera movements. Note also that more keypoints are found in full resolution SSLAM implementation than with SSLAM†. Nevertheless, different image resolutions do not affect the other parameters of the methods since keypoint positions are rescaled at the full resolution before the constrained matching in both cases.

SSLAM Parameter Analysis

Different versions of SSLAM are compared, corresponding to the successive improvements of the pipeline proposed in Sect. 2.3.3, in particular we analyzed different versions of the more challenging SSLAM†. SSLAM†* indicates the first version which only includes the loop chain matching described in Sect. 2.3.2, while the adaptive keyframe selection is incorporated in the default SSLAM†.

In order to analyze the robustness and the effectiveness of the proposed method, the SSLAM† system was tested with a different number of RANSAC iterations for the pose estimation. In particular, results of SSLAM† with 500, 15 (set as default) and 3 RANSAC iterations, and SSLAM†* with 500 iterations are presented, indicated respectively by SSLAM†/500, SSLAM†/15, SSLAM†/3 and SSLAM†*/500.

Figure 2.13 shows the average translation and rotation errors of the different SSLAM† variants for increasing path length and speed, according to the first 11 sequences of the KITTI dataset—see Fig. 2.14. Similar results hold in the case of full resolution SSLAM.



(A)

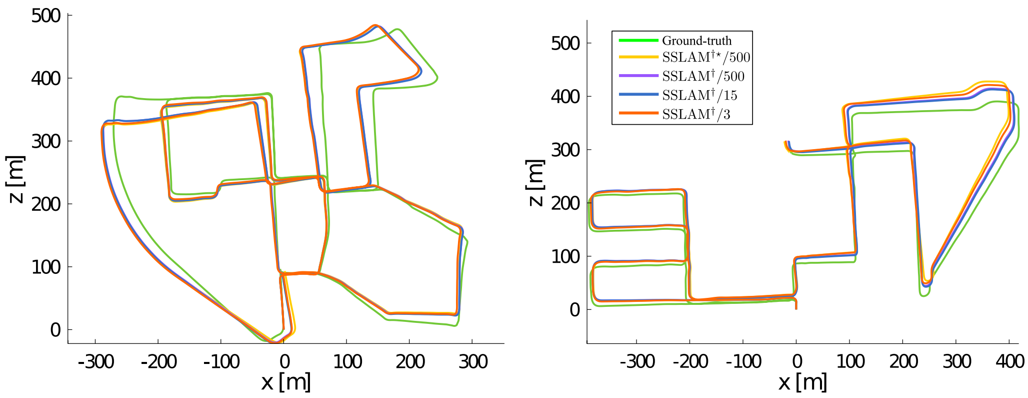
(B)

(C)

(D)

FIGURE 2.13: (Best viewed in color) Average error on the first 11 sequences of the KITTI dataset. Plots (a-b) refer to the average translation and rotation error for increasing path length respectively, while plots (c-d) refer to increasing speed.

The chain loop matching scheme together with the chosen keypoint detector and



(A)

(B)

FIGURE 2.14: (Best viewed in color) An example of the paths computed for Sequence 00 (a) and Sequence 08 (b) of the KITTI dataset.

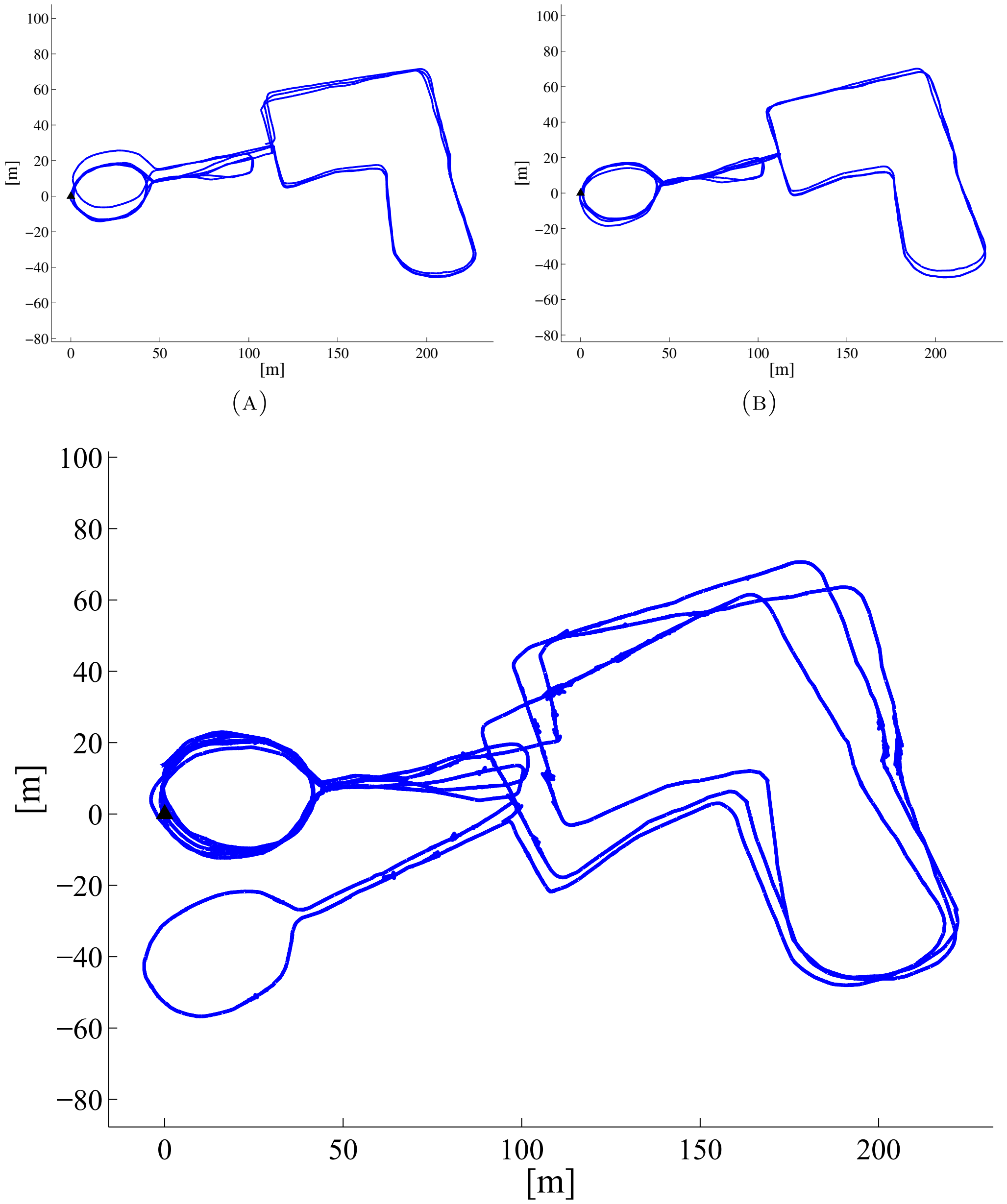
descriptor is robust even for long paths, without bundle adjustment or loop closure. SSLAM† improves on the standard pose estimation without keyframes selection (SSLAM†*), allowing to track longer paths and confirming that the proposed keyframe selection strategy is effective.

Moreover, results for SSLAM†/15 and SSLAM†/500 are equivalent, while SSLAM†/3 obtains inferior results but similar to those obtained by SSLAM†*/500, giving an evidence of the robustness of the proposed matching selection strategy and pose estimation.

A further test aiming at investigating the fixed point threshold δ_f used to accept a frame as keyframe was also done. This is the parameter that mainly affects the results, since selected keyframes decrease as δ_f increases, while we verified that the computation is stable with respect to the choice of the other parameters. In particular, we run SSLAM for different values of $\delta_f = 30$, 50 (default), 80 px on both datasets. In the case of $\delta_f = 30$ slightly inaccurate paths are present with respect to $\delta_f = 50$ on both datasets, while for $\delta_f = 80$ higher pose errors are found.

Figure 2.15 shows the behavior of SSLAM for the different values of δ_f on the New College sequence. Clearly the default set $\delta_f = 50$ px provides better results since even after a long path loops are correctly closed. This results confirm the observation that avoiding close keyframes improves the results, but this choice must be balanced with the tracking capability of the system.

Table 2.4 shows the average number of frames between two consecutive keyframes



(A)

(B)

(C)

Figure 2.15: SSLAM estimated paths for the New College video sequence with $\delta_f = 30$ px (a), $\delta_f = 50$ px (b) and $\delta_f = 80$ px (c).

Table 2.4: Average number of frames between two consecutive keyframes and the corresponding standard deviations for different values of the threshold δ_f.

δ_f	35	55	85	35	55	85
	Average			Std		
KITTI	1	2	3	1	1	2
New College	5	10	32	8	13	39

and the corresponding standard deviations. The values are higher for the New College dataset with respect to the KITTI dataset, according to the different camera speeds. Furthermore, as it can be noted in Fig. 2.16 the keyframe distribution is not uniform but it is denser near camera turns and accelerations.

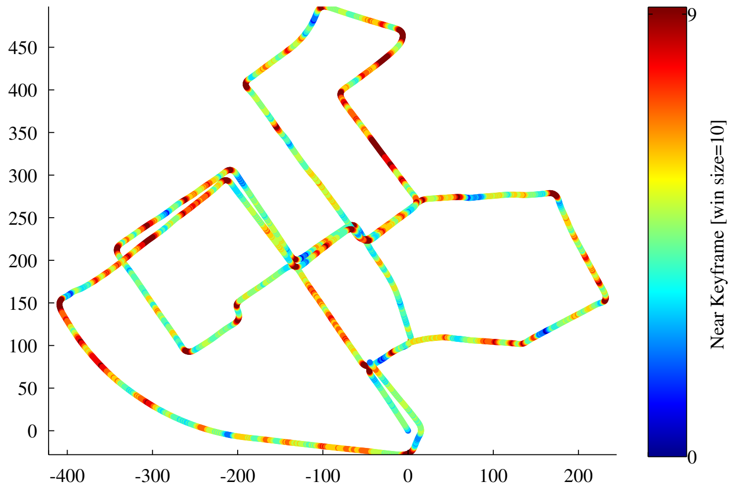


FIGURE 2.16: (Best viewed in color) An example of keyframe distribution along the Sequence 00 of the KITTI dataset for SSLAM (default δ_f=50 px). At each estimated camera position the number of keyframes that fall inside a window of 10 frames centred at the camera location is shown according to the colorbar gradation.

Evaluation on the KITTI dataset

Results on the KITTI odometry benchmark for stereo methods only (more details are available online (Geiger et al., 2012)) are hereafter reported. Methods that employ other sensors than stereo cameras, such as point cloud obtained with laser scanner, are excluded from this discussion.

Table 2.5 reports the evaluation average results, while in Fig. 2.17 shows the average translation and rotation errors of the different methods for increasing path length and speed. At the time of submission SSLAM and SSLAM† was ranked respectively in the 3th and 8th positions, while currently[1] the proposed solutions are ranked in the 6th and 11th positions according to the translation errors.

These rank placements show the robustness of the proposed methodology. Note however that the benchmark provides qualitative results, since these error metrics cannot take into account all the properties of a SLAM system. In particular,

[1] November 2014

referring to Fig. 2.18 where five sample tracks of the KITTI dataset are reported, it can be seen that while eVO and VISO2-S show worse results—and also TLBBA regarding sequence 13 (Fig. 2.18c)—the other methods have similar performance given that all trajectories are well aligned with the ground truth. Moreover it can also be shown that plots of the approaches ranked between the 7^{th} and the 10^{th} positions are less accurate w.r.t. SSLAM†.

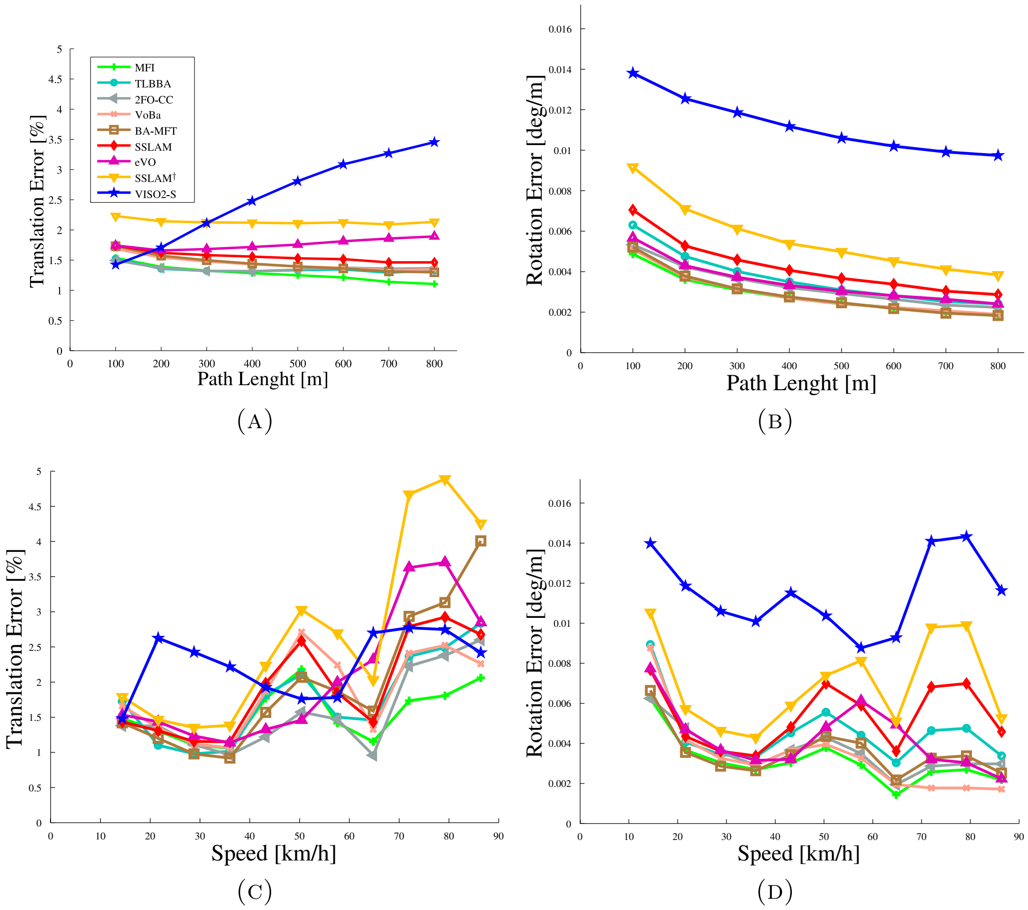
Table 2.5: KITTI benchmark evaluation average results.

Rank	Method	Transl. error(%)	Rot. error (deg/m)
1	MFI (Badino et al., 2013)	1.30	0.0030
2	TLBBA (Lu et al., 2013)	1.36	0.0038
3	2FO-CC (VISAP, 2015)	1.37	0.0035
4	VoBa	1.46	0.0030
5	BA-MFT	1.47	0.0031
6	**SSLAM**	**1.57**	**0.0044**
7	eVO	1.76	0.0036
⋮	⋮	⋮	⋮
11	**SSLAM†**	**2.14**	**0.0059**
12	VISO2-S (Geiger et al., 2011)	2.44	0.0114

Table 2.6 shows the input matches and the found inliers in the RANSAC pose estimation by SSLAM, SSLAM† and VISO2-S. Note that VISO2-S works similarly to SSLAM and its code is freely available. As it can be noted, while SSLAM† outputs a comparable number of initial matches with VISO2-S, only 50% of these are inliers for VISO2-S: This implies that our matching strategy is more robust. Note also that the spatial and temporal flow constraints of VISO2-S are more tight, which would lead theoretically to a higher number of matches since the probability to make an accidental wrong match is higher for SSLAM and SSLAM† (except for the epipolar constraint δ_y, the other thresholds are about equal to the minimal image size). Yet, as it can be seen from Table 2.6, the opposite holds, in favour of the robustness and stability of the propose methodology.

Lastly, according to the little documentation of all the methods, since most of them are anonymous, SSLAM and SSLAM†, MFI and VISO2-S are the only SLAM methods which do not use bundle adjustment, loop closure and work locally taking into account only two consecutive keyframes, thus leading efficiently and light implementations for long sequences.

The SSLAM approach can be considered complementary to that proposed by MFI, which averages the estimation of tracked keypoints across previous frames in order



(A) (B)

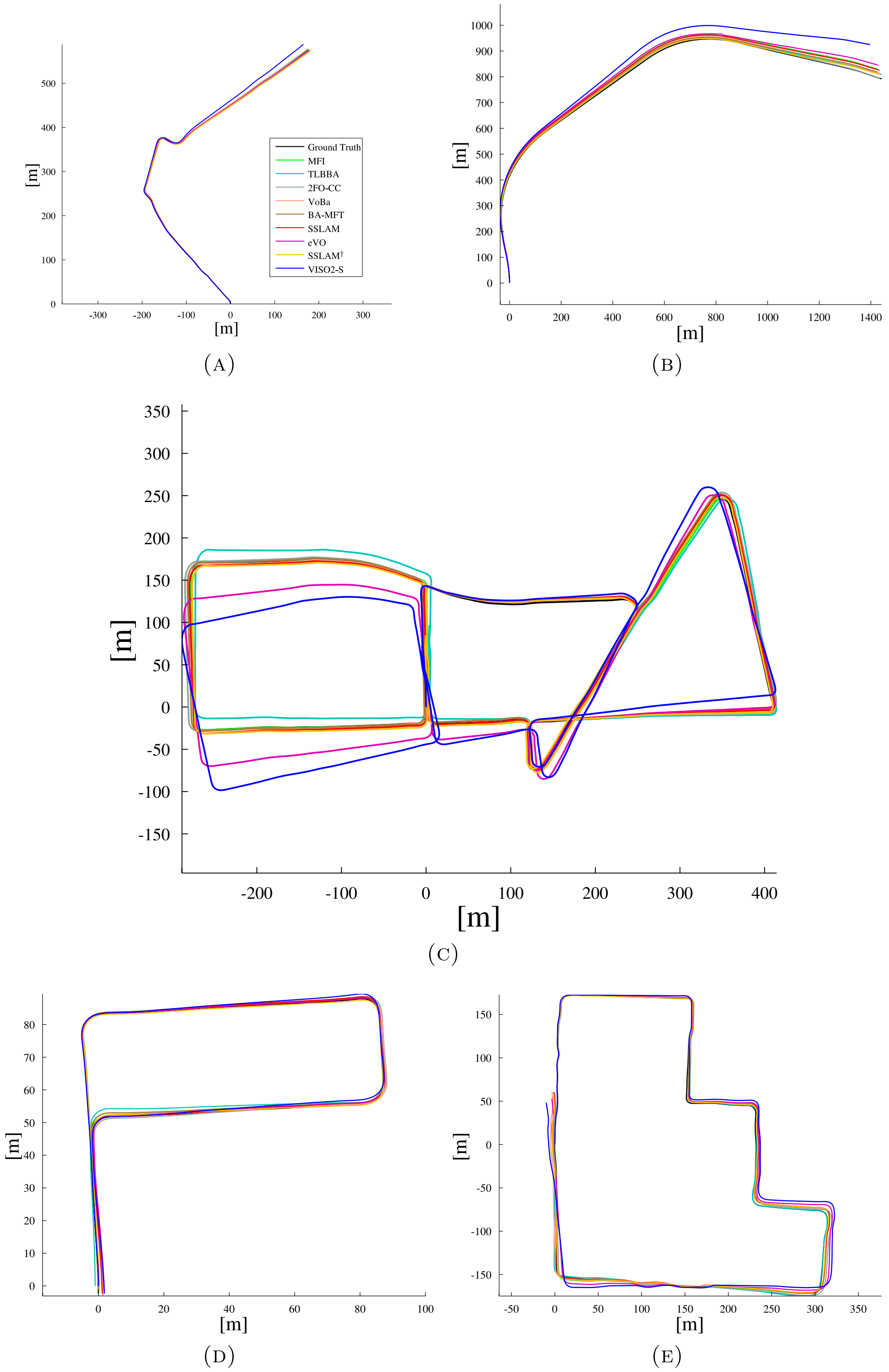
(C) (D)

FIGURE 2.17: (Best viewed in color) Average error on the KITTI benchmark. Plots (a-b) refer to the average translation and rotation error for increasing path length respectively, while plots (c-d) refer to increasing speed.

to lower the error (Badino et al., 2013). This implies that MFI takes advantage from video sequences with long tracks, such as those provided by vanishing points on the KITTI dataset, i.e. exactly the fixed scene points that SSLAM tries to avoid. It is therefore reasonable to expect that MFI results would degrade with respect to SSLAM for high parallax scenes, such as the case of a translation movement perpendicular to the camera, or with a low framerate or fast translations, that make more difficult to track keypoints among more frames.

Evaluation on the New College dataset

We tested SSLAM and SSLAM† versus VISO2-S, the highest rank in the KITTI benchmark for which code is available. For the sake of comparison, the VISO2-S threshold values used during matching were modified from the default ($\delta_r = 200, \delta_x = 200, \delta_y = 3$) px—used in the KITTI evaluation—with ($\delta_r = 100, \delta_x =$



(A)

(B)

(C)

(D)

(E)

Figure 2.18: (Best viewed in color) Trajectories on the sequences 11 (a), 12 (b), 13 (c), 14 (d) and 15 (e) of the KITTI dataset.

$100, \delta_y = 3)$, given that the latter values achieve better estimation results than the default ones in the New College dataset.

SLAM methods are tested not only on the whole sequence but also on the two subsequences corresponding to the small and large loops present in the images. This is done to analyze the behavior of the methods at different starting points. Figure 2.19 shows the obtained tracks. While VISO2-S diverges as the sequence grows, both SSLAM and SSLAM† maintain the correct paths, closing the loops, without the need of bundle adjustment and loop closure techniques. In particular, full resolution SSLAM works slightly better than SSLAM†. This becomes noticeable only at the end of the last part of the video sequence.

The New College video sequence seems more reliable than the KITTI sequences, since as it can be seen from Table 2.6, all methods achieve a higher number of tracked keypoints but also inliers, maybe due to slower camera movements. Anyway, VISO2-S still obtains a lower number of matches and inliers with respect to SSLAM and SSLAM†. Note also that the absence of the optionally translation constraint δ_{θ_2} in this sequence does not affect the quality of the results.

Table 2.6: Average number of input matches before the RANSAC pose estimation and final inlier ratios.

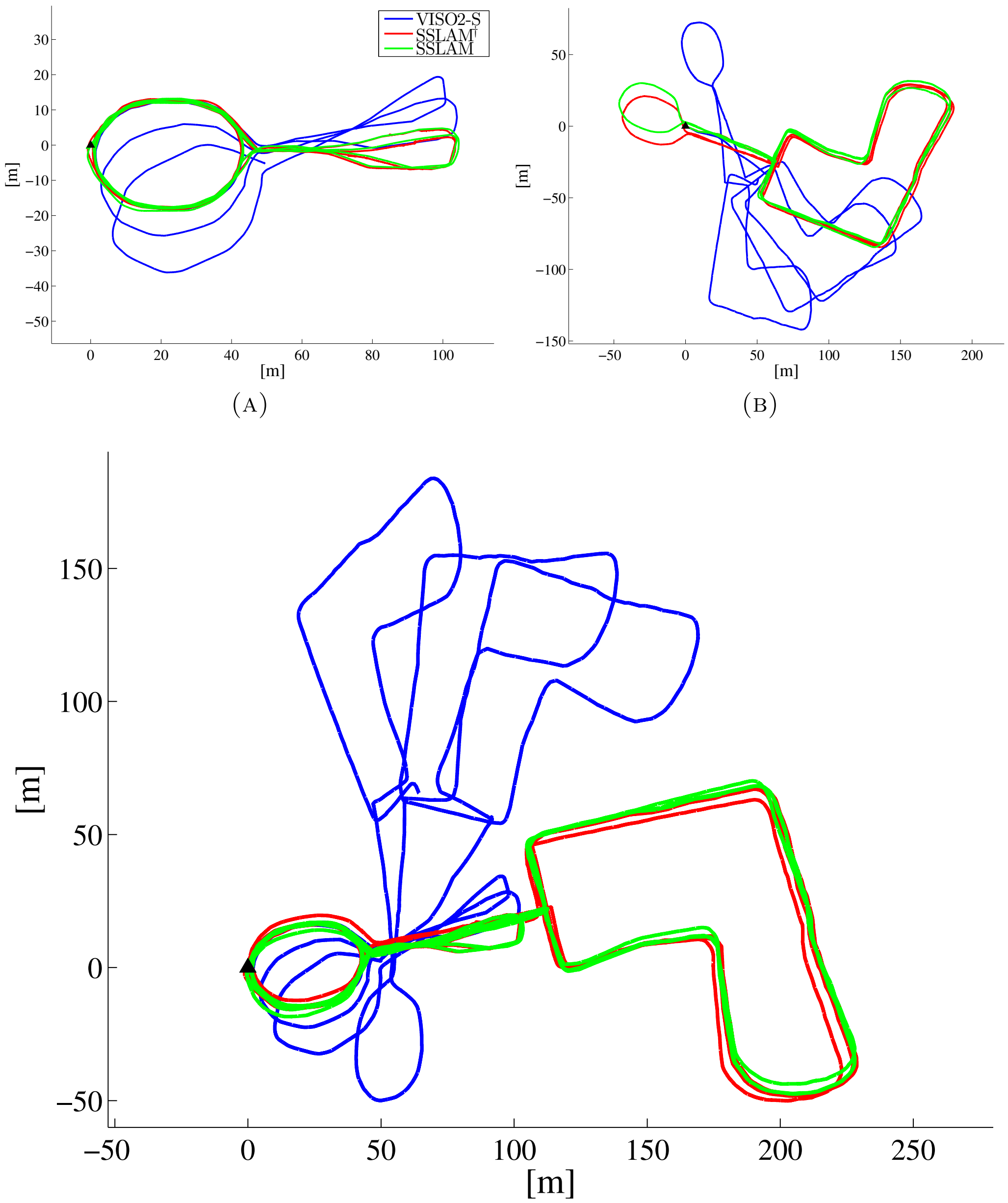
	KITTI		New College	
	pts	inl(%)	pts	inl(%)
SSLAM	766	98	780	99
SSLAM†	222	96	201	97
VISO2-S	245	50	156	84

Underwater experiment

SSLAM has been tested also in underwater environment with the aim to exploit its capabilities in the ARROWS Project[2](Allotta et al., 2013).

ARROWS (ARchaeological RObot systems for the World's Seas) is a European project founded by the European Commission under the Environment Theme of the 7th Framework Programme for Research and Technological Development. Its objective is the realization of low cost underwater autonomous vehicles (AUVs) to be used by archaeological team in exploratory campaigns. SSLAM should be

[2] http://www.arrowsproject.eu/



(A)

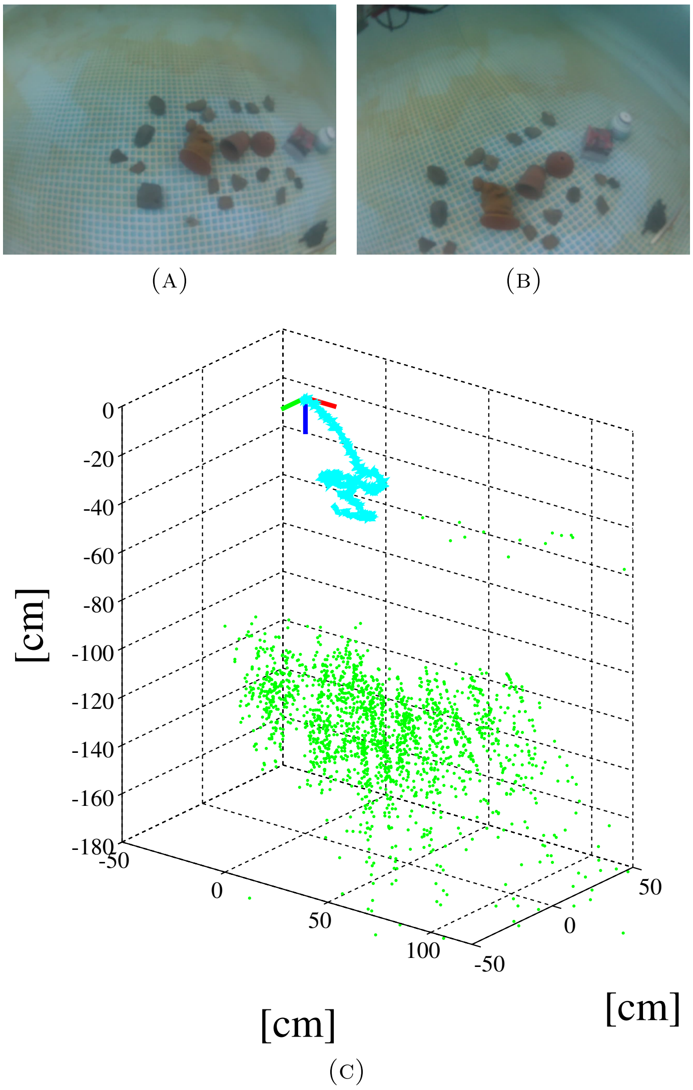
(B)

(C)

Figure 2.19: (Best view in color) Estimated paths for the New College video sequence. The plots (a), (b) and (c) refer respectively to first subsequence (from frame 0 to frame 18400), to the last subsequence (from frame 18400 to frame 52479) and to the whole sequence. Note that to achieve the best top view, each sequence was rotated so that the displayed axes correspond to the major directions of the autocorrelation matrix of the point positions, i.e., to the two greatest eigenvectors.

used in conjunction with other sensors to let the vehicles localize itself in the environment and perform autonomous navigation tasks.

Given the absence of underwater stereo dataset with a provided ground truth,



(a)

(b)

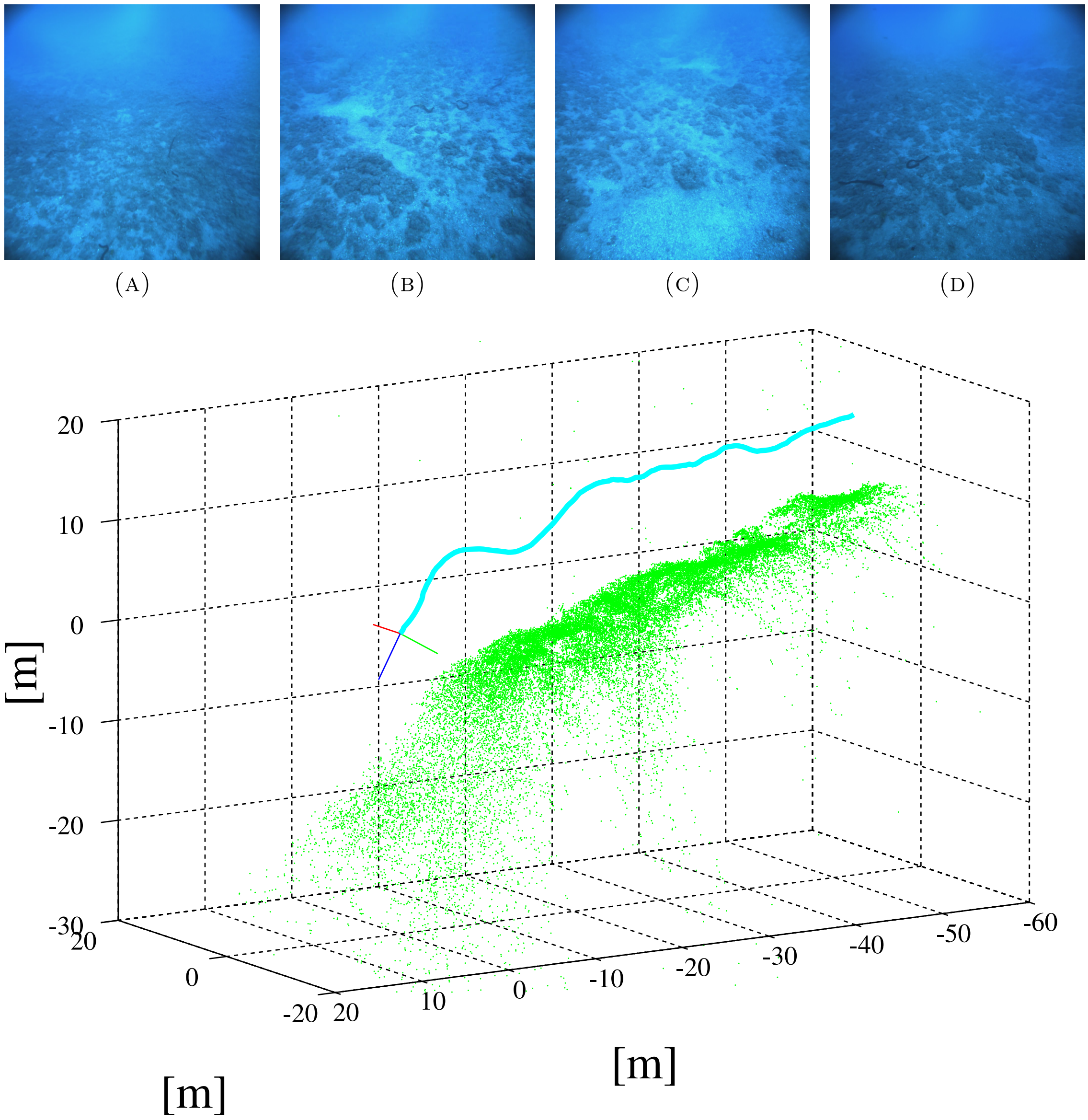
(c)

Figure 2.20: (a) and (b): Images acquired in the swimming pool located at the MDM Laboratory of the University of Florence, using a pair of synchronized GoPRO underwater cameras. (c): Estimated trajectory and 3D map.

some experiment have been carried out both in a swimming pool (Fig. 2.20) and in open sea (Fig. 2.21). No reliable ground truth is available, anyhow in both cases, SSLAM produces plausible estimations respect to the apparent motion visible in the image sequences.

Dense Map Upgrade

Maps created with SSLAM are sparse three-dimensional point clouds. On the one hand, these sparse representations are more than sufficient to be used as reliable landmark in the localization task and offer a simple management with relative



(A) (B) (C) (D)

(E)

FIGURE 2.21: (a) to (d): Example images of a stereo video sequence acquired during an experimental campaign near Akko, Israel. The sequence is recorded with a synchronized pair of Basler ACE cameras, enclosed in a waterproof housing. (e): Three-dimensional plot of the estimated trajectory and structures.

small memory occupancy. On the other hand, by representing the environment in such a way could be insufficient both for robot complex autonomous interactions and for render visually pleasant 3D models of the environment—see for example the green 3D point cloud of the sea-bottom presented in Fig. 2.21. As anticipated in the Introduction chapter (Subsect. 1.1.5) the SLAM output could be used to obtain improved the 3D representations of the scene.

During the ARROWS Project, one of the goal to be achieved is the production of

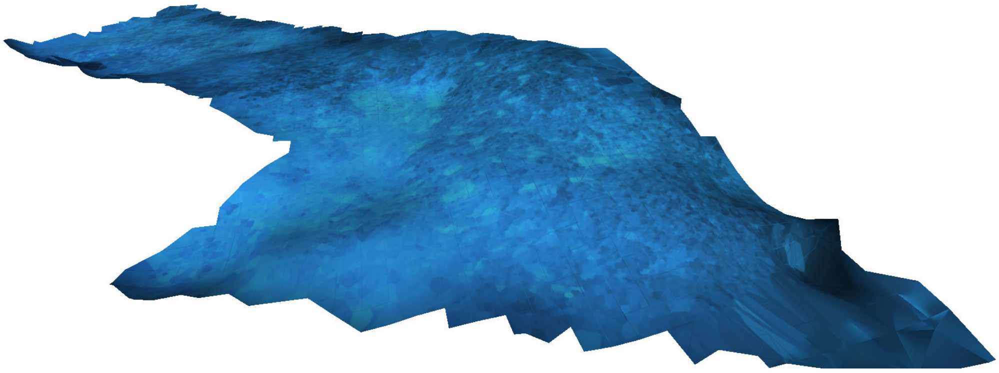


FIGURE 2.22: Dense 3D reconstruction of the sea-floor obtained employing a region growing approach using as input the SSLAM estimate presented in Fig. 2.21

three-dimensional models based on the acquired underwater images, to be used to inspect and document areas of archaeological interest. Clearly SSLAM 3D output is not sufficient.

To obtain more dense reconstructions, two solutions have been tested. Using a dense stereo matching algorithm—more specifically the SemiGlobal Block Matching[3] (SGBM) algorithm implemented into the OpenCV library—is possible to compute local dense maps with sufficient accuracy exploiting a single stereo frame with minimal computational burden. Then with the SSLAM motion estimates, the local maps could be registered into a global model—see Fig. 1.3.

Even if the produced 3D map is not optimal, this solution enriches the 3D scene representation. However stereo matching algorithms typically suffer of two main drawback: (i) in scene with no texture or highly repeated pattern matches are hard to find; (ii) many internal parameters have to be tuned to obtain satisfying results. In fact, in our test, using the same SGBM configuration with underwater images produced noisy disparity maps and a poor quality 3D reconstruction.

The second proposed solution employs a region growing algorithm (Furukawa and Ponce, 2010). Starting from the image correspondences produced by SSLAM together with the localization information, its possible to obtain dense and accurate 3D models—see Fig. 2.22—at the cost of a long computation time.

[3] For more details see `http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#stereosgbm`

Running Times

The SSLAM approach is implemented in C/C++ non-optimized multithreaded code. As it can be seen in Table 2.7, where the average running times for a single frame are reported, SSLAM scales with the resolution. The feature detector is accurate but slow, since it requires large size kernel convolutions. By taking into account that only keyframes are strictly required by SSLAM, real-time performance is achieved when the keyframe computational time is less than f_k/f_v, where f_k is the keyframe rate (see Table 2.4) and f_v is the frame rate of the video sequences ($f_v = 10$, $f_v = 20$ for the KITTI and New College datasets respectively). This implies that the time to estimate a single keyframe must not exceed 0.2 s and 0.5 s respectively for the KITTI and New College sequences. Although only SSLAM† can run almost in real-time, eventually code optimization using GPU acceleration could be used to improve the running times.

Table 2.7: Average computational time for a single frame on a Intel-i7 3.50GHz CPU, 8 cores are used.

	SSLAM	SSLAM†
KITTI	3.85 s	0.55 s
New College	0.95 s	0.20 s

CHAPTER *3*

Hybrid 3D Reconstruction

In this Chapter are described two softwares for hybrid 3D reconstruction that exploit passive and active methods to recover 3D models of desktop-size objects. After a brief presentation of related works, the active triangulation method is described (see Sect. 3.2). This approach used in conjunction with the passive motion estimation obtained with a SLAM software could lead to accurate 3D reconstruction of even texture-less objects. In this thesis both the monocular (Sect. 2.2) and the stereo (Sect. 2.3) SLAM systems are used. In the former case a device with a single camera and a laser stripe projector is used, while in the latter two synchronized cameras are employed together with the laser.

Then will be described steps common to both setups that encompass the laser detection in the images (Subsec. 3.3.1), 3D profiles computation and collation (Subsec. 3.3.2) and finally an optional refinement (Subsec. 3.3.3). Results (Sect. 3.4) with mono and stereo setups conclude the chapter.

3.1 Related Works

Three-dimensional reconstruction from video sequences is a key topic in computer vision. Reconstruction methods can be broadly classified into *passive*—using only information directly extracted from the images—and *active*—employing more sophisticated hardware to ease the acquisition process and improve the results.

Passive methods use only unstructured illumination, and focus on low cost hardware and a sophisticated software, by which a reasonable accuracy and a high flexibility can be obtained. Typical passive approaches encompass multi-view reconstruction from either image collections (Farenzena et al., 2009, Agarwal et al., 2009) or image sequences (Vogiatzis and Hernàndez, 2011), real time stereo (Wang et al., 2006) and shape from shading (Zhang et al., 1999).

On the other hand, active methods recover the 3D structure of the scanned object by exploiting a known light pattern that interacts with the scene while the camera is kept fixed. Accurate models are obtained also for textureless objects, working in structured conditions with sophisticated hardware and relatively simple algorithms.

Rocchini et al. (2001) use a pattern with several light stripes arranged in a regular way; Object shape is obtained through the so called *active triangulation* approach of single image points. While in the classical passive triangulation the intersection between optical rays of corresponding feature points is computed, in the active triangulation a single optical ray is used an its incidence with the light pattern model is estimated to obtain the associated three-dimensional point (see also Sect. 3.2).

An active triangulation approach is also used in (Winkelbach et al., 2006), where a laser plane is projected onto the scene in the presence of a known 3D structure, so as to simultaneously estimate the laser plane equation and reconstruct small-size objects. In (Habbecke and Kobbelt, 2008) a similar approach—used also to reconstruct room-size environments—employs a laser pointer array swept over the scene: Differently form the former method, this latter one need a pre-processing step to calibrate the laser array. Colombo et al. (2011) adopt an *active rectification* approach, where model reconstruction is carried out by warping and collating single laser profiles.

In (Bouguet and Perona, 1999), an active/passive method is presented where cast shadows produced with a wand are used instead of projected light. Another hybrid approach extending standard shape from shading (Zhang et al., 1999) is photometric stereo (Hernàndez et al., 2008), where a collection of photos of the object is taken from a single viewpoint by varying the light source.

3.2 Active triangulation

With *active triangulation* we refer to the technique used to estimate the depth of a point in an image exploiting the geometrical characteristics of a known patter while it interacts with the scene. To better explain such method, we suppose to use a monocular camera in conjunction with a laser fan projector that draws a stripe over the scene.

In Fig. 3.1 the proposed setup is outlined. The laser projects in the environment a fan representable as a three-dimensional plane Λ. When the laser plane hits an object it draws a stripe Γ that is deformed according to the object shape. Once the camera captures the scene, in the image a stripe γ—projection of Γ—is visible. For each image point $\mathbf{x} \in \gamma$ its pre-image $\mathbf{X} \in \Gamma$ can be computed as the intersection of the optical ray passing through the camera center $\mathbf{C}$ and the point $\mathbf{x}$ with the laser plane Λ. In other words, once estimated the laser plane equation w.r.t. the camera coordinate frame, it's possible to recover $\mathbf{X}$ as

$$\mathbf{X} = \alpha \mathtt{K}^{-1}\mathbf{x} \quad \text{with} \quad \alpha = \frac{d_\Lambda}{\mathbf{n}_\Lambda^\top \mathtt{K}^{-1}\mathbf{x}} \tag{3.1}$$

where $\mathbf{n}_\Lambda$ and d_Λ are respectively the laser plane normal and its distance w.r.t. the camera coordinate frame and $\mathtt{K}$ is the camera calibration matrix.

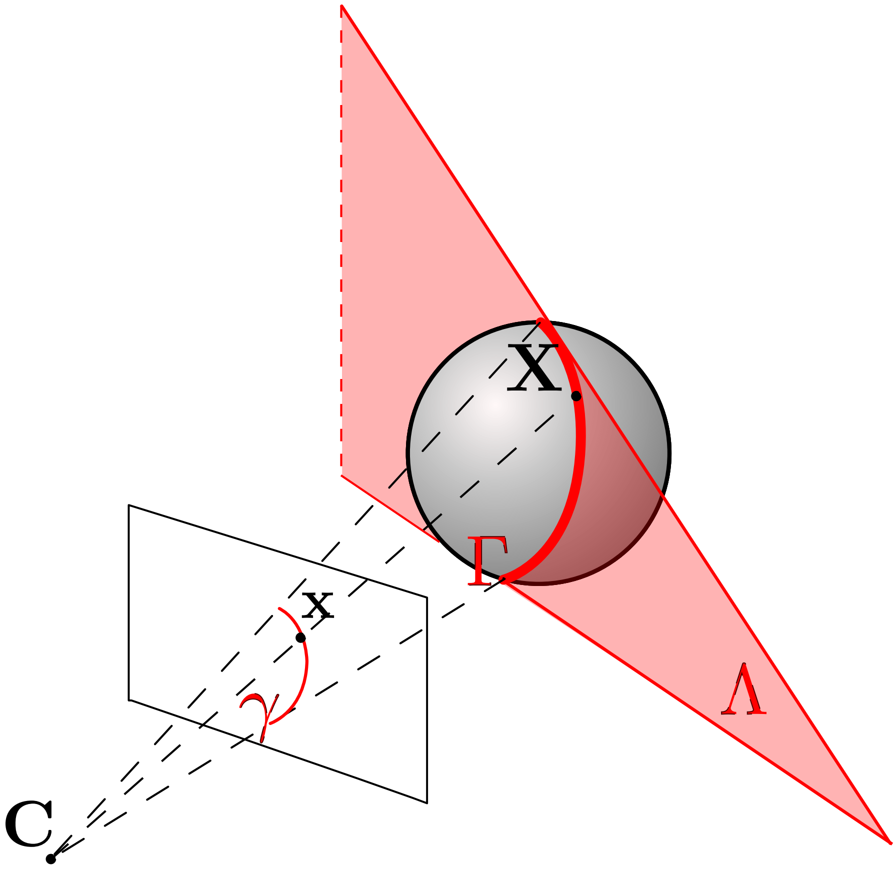


Figure 3.1: Active triangulation scheme. The laser plane Λ interacts with the object drawing a stripe Γ that is projected over the image in γ. For each 2D point $\mathbf{x} \in \gamma$ it's possible to obtain its pre-image $\mathbf{X} \in \Gamma$ computing the intersection between the optical ray passing through the camera center $\mathbf{C}$ and $\mathbf{x}$ and the laser plane Λ.

To obtain the laser plane parameters a calibration step have to be performed (see Fig. 3.2). Using a planar checkerboard pattern moved in front of the device kept fixed, different three-dimensional planes $\pi^{(i)}$ are captured from the camera. Each pattern induces an homography that can be exploited to recover the planar pattern parameters.

In fact, suppose to choose a coordinate frame $\mathcal{P}$ such that the checkerboard plane π has equation ${}^{\mathcal{P}}Z = 0$. A 3D point over the plane will be expressed in homogeneous coordinates as ${}^{\mathcal{P}}\mathbf{X}_\pi = [{}^{\mathcal{P}}X, {}^{\mathcal{P}}Y, 0, 1]^\top$. Then the following relation holds

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathtt{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} {}^{\mathcal{P}}X \\ {}^{\mathcal{P}}Y \\ 0 \\ 1 \end{bmatrix} = \mathtt{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} {}^{\mathcal{P}}X \\ {}^{\mathcal{P}}Y \\ 1 \end{bmatrix} = \mathtt{H}{}^{\mathcal{P}}\mathbf{X}_\pi \tag{3.2}$$

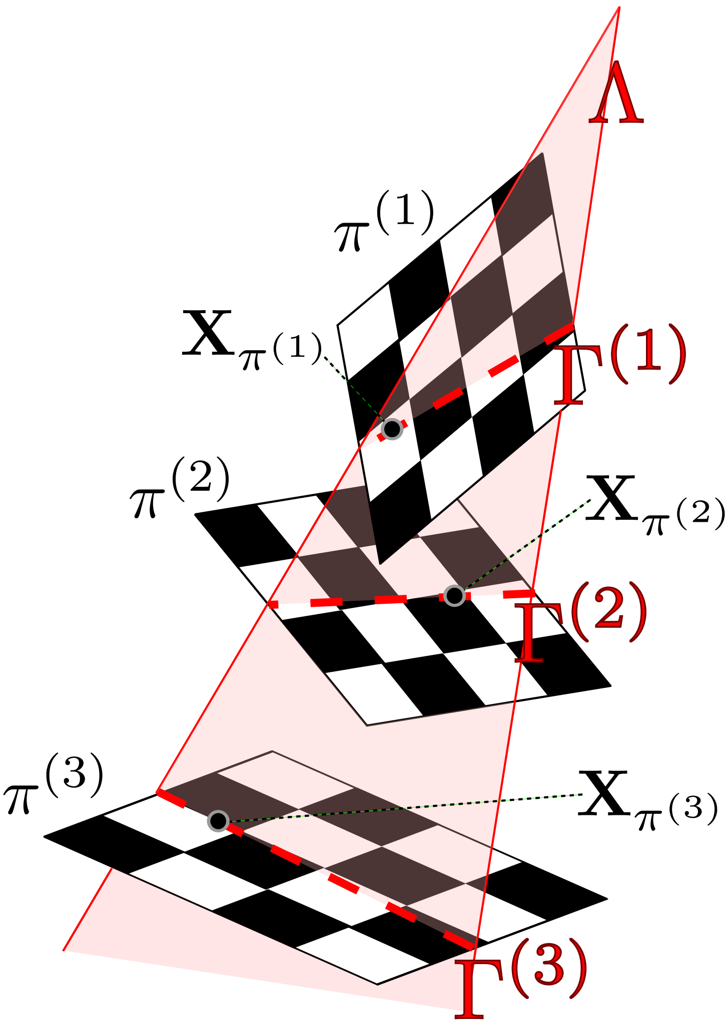


FIGURE 3.2: Laser plane calibration. To obtain the laser plane parameters a calibration routine must be executed. A planar checkerboard pattern is moved in front of the camera-laser device kept in fixed position. Each patter orientation produce a different plane $\pi^{(1)}, \pi^{(2)}, \pi^{(3)}$ w.r.t. the camera coordinate system. Exploiting the homography that exist between each patter $\pi^{(i)}$ and its image, it's possible to recover its parameter $\{\mathbf{n}_{\pi^{(i)}}, d_{\pi^{(i)}}\}$ (see text for details). Then using the active triangulation equation 3D points over the patter $\mathbf{X}_{\pi^{(i)}}$ could be estimated. With at least three non-aligned 3D points the laser plane equation could be estimated solving a linear system.

where $\mathtt{H} = \mathtt{K}[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}]$ is the homography between π and its image, $\mathbf{r}_j$ for $j = 1, 2, 3$ are columns of a rotation matrix and $\mathbf{t}$ is a translation vector. Once estimated the homography $\hat{\mathtt{H}} = [\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \hat{\mathbf{h}}_3]$ up to a scalar factor h_0—i.e. $\hat{\mathtt{H}} = h_0\mathtt{H}$—it's possible to recover $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$ and $\mathbf{t}$ as

$$\mathbf{r}_j = \frac{1}{h_0}\mathtt{K}^{-1}\hat{\mathbf{h}}_j \tag{3.3}$$

for $j = 1, 2$ and

$$\mathbf{t} = \frac{1}{h_0}\mathtt{K}^{-1}\hat{\mathbf{h}}_3 \tag{3.4}$$

while $h_0 = \parallel \mathtt{K}^{-1}\hat{\mathbf{h}}_1 \parallel$, given that $\parallel \mathbf{r}_1 \parallel = 1$, and finally $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$. $\mathtt{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ and $\mathbf{t}$ express the roto-translation that links the camera coordinate frame with that attached on the checkerboard pattern. So it holds

$$\begin{bmatrix} {}^{\mathcal{P}}X \\ {}^{\mathcal{P}}Y \\ 0 \end{bmatrix} = \mathtt{R}^\top \begin{bmatrix} {}^{\mathcal{C}}X \\ {}^{\mathcal{C}}Y \\ {}^{\mathcal{C}}Z \end{bmatrix} - \mathtt{R}^\top \mathbf{t} \tag{3.5}$$

where $[{}^{\mathcal{C}}X, {}^{\mathcal{C}}Y, {}^{\mathcal{C}}Z]^\top$ is the 3D point expressed in the camera frame $\mathcal{C}$; In particular

$$0 = \mathbf{r}_3^\top \begin{bmatrix} {}^{\mathcal{C}}X \\ {}^{\mathcal{C}}Y \\ {}^{\mathcal{C}}Z \end{bmatrix} - \mathbf{r}_3^\top \mathbf{t} \tag{3.6}$$

is the plane π equation expressed in the camera coordinate frame, where $\mathbf{n}_\pi = \mathbf{r}_3$ is its normal and $d_\pi = \mathbf{r}_3^\top \mathbf{t}$ its distance.

Computed the parameters for each checkerboard plane $\pi^{(i)}$, for each 2D point $\mathbf{x}_{\pi^{(i)},\gamma^{(i)}}$—that lies simultaneously on the checkerboard pattern and on the laser stripe—its 3D pre-image $\mathbf{X}_{\pi^{(i)},\Gamma^{(i)}}$ can be obtained using

$$\mathbf{X}_{\pi^{(i)},\Gamma^{(i)}} = \frac{d_{\pi^{(i)}}}{\mathbf{n}_{\pi^{(i)}}^\top \mathtt{K}^{-1}\mathbf{x}_{\pi^{(i)},\gamma^{(i)}}} \mathtt{K}^{-1}\mathbf{x}_{\pi^{(i)},\gamma^{(i)}} \tag{3.7}$$

similar to Eq. 3.1, but with the plane $\pi^{(i)}$ parameters instead of those of the laser plane Λ. With at least three non-aligned 3D points $\mathbf{X}_{\pi^{(i)},\Gamma^{(i)}}$, the laser plane equation could be estimated solving a linear system.

3.3 Model Building

In this section the main steps of the proposed hybrid reconstruction method are presented and discussed. In Figure 3.3 a schematic representation of the implemented pipeline is shown.

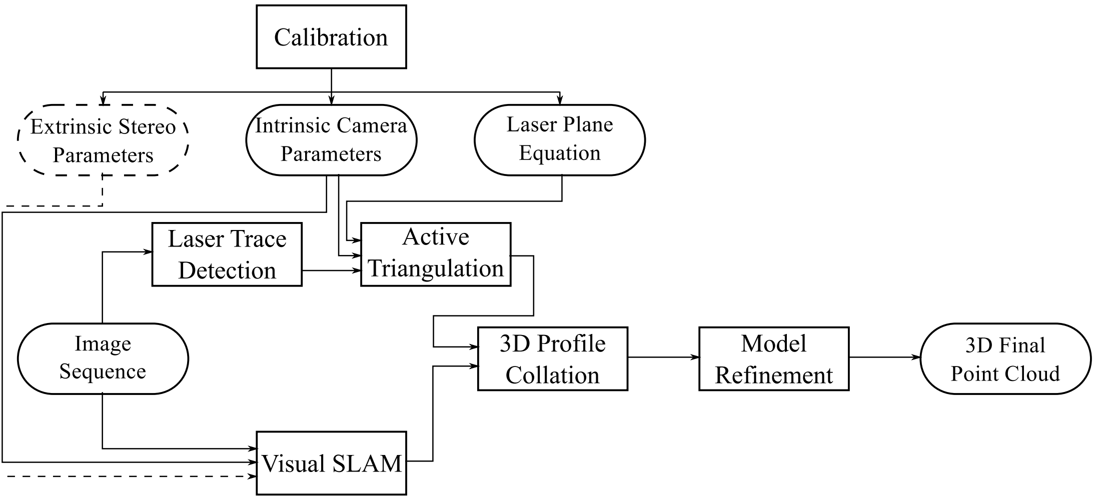


FIGURE 3.3: Block scheme of the proposed hybrid reconstruction method. Dashed lines indicate information used only in the stereo setup.

3.3.1 Laser Detection

To be able to triangulate laser points as described in Sect. 3.2, previously such points must be detected in the image. In more typical active 3D reconstruction framework (Winkelbach et al., 2006, Habbecke and Kobbelt, 2008) the laser trace is detected with *background-subtraction* techniques, given that the camera is kept in fixed position and the imaged scene doesn't change during the acquisition. In our framework, this approach is not feasible.

Given that the laser draws a bright line on the image (see Fig. 3.4a and 3.4b) by isolating the intensity channel corresponding to the laser color—red in our experiments—the laser trace is easily detectable (see Fig. 3.4c). Moreover, assuming to set the laser projector so that its trace crosses diagonally in the image, it's possible to enforce two useful constraints on the laser detection: (i) the laser search could be performed along singular horizontal scan-line and (ii) for each scan-line there can be at most only one laser point.

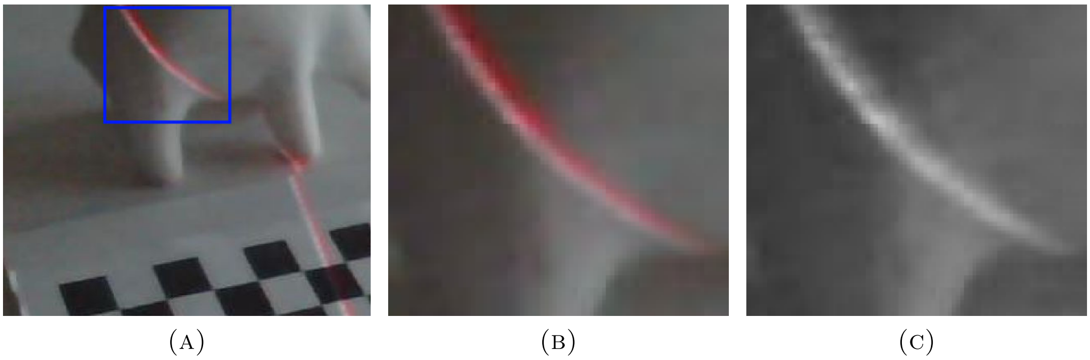

(A) (B) (C)

FIGURE 3.4: Laser stripe appearance on the image. (a) acquired image; (b) zoom of the region included into the blue square; (c) red channel of the image: the pixels of the laser stripe have higher intensity value.

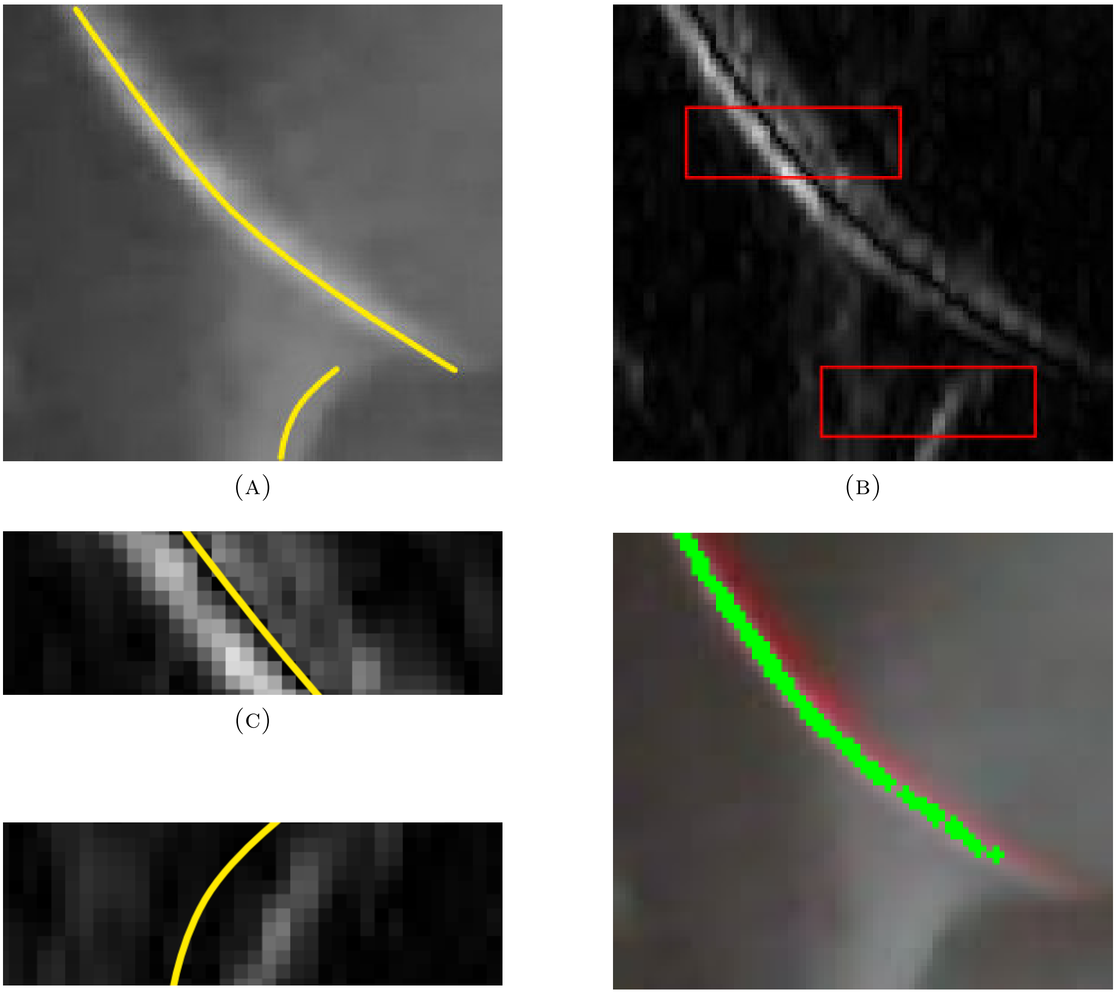
For each scan-line the higher intensity pixels are selected. To avoid losing valid laser points, a low threshold is used: In this case, together with the pixels of the laser trace, also false positive detections are retrieved (see Fig. 3.5a). Observing the image gradient—shown in Figure 3.5b—the laser trace appears surrounded by two strong edge, so only detected points that have edges on both of their left and right sides are kept (see Fig. 3.5c). If in a scan-line more than one candidate satisfies the edge-constraint, the maximum intensity point is chosen. The selected points describe the laser stripe on the image (Fig. 3.5e). Then sub-pixel accuracy is then achieved using a Center of Mass algorithm (Fisher and Naidu, 1996).

Stereo laser detection

When using a stereo setup with a pair of synchronized cameras together with the laser projector, an additional constraint could be enforced during the laser detection—see Fig. 3.6.

Let $\Lambda^l : \{\mathbf{n}^l_\Lambda, d^l_\Lambda\}$ be the laser plane expressed in the left camera coordinate system. A detected point $\mathbf{x}^l$ can be actively triangulated with Λ^l using Eq. 3.1. The obtained 3D point $\mathbf{X}^l$ is then projected into the right camera frame in the point $\mathbf{x}^r$. The correspondence $\{\mathbf{x}^l, \mathbf{x}^r\}$ is then weighted using a Sum of Absolute Differences (SAD) over a square window. This idea comes from the observation that a wrongly detected laser point, if transferred over the right image using the active triangulation, will fall in a very different image region, likely causing SAD to be relatively high. Points with higher weights are therefore discarded. ■

Once laser points are detected on the images, active triangulation (Sect. 3.2) is used to compute 3D laser profile. It's worth remember that each 3D profile is



(a)

(b)

(c)

(d)

(e)

Figure 3.5: To extract image pixels that belong to the laser trace at first the higher intensity pixels are selected (a). Then, to discard false positives the image gradient is exploited (b). Given that the laser trace draws on the image two strong side edges, true positives are found if the previously detected pixels are surrounded by two edges (c) (zoom of the top rectangle in (b)). (d) is an example of a false positive detection, where an edge is missing. Finally points of the laser trace are selected (e).

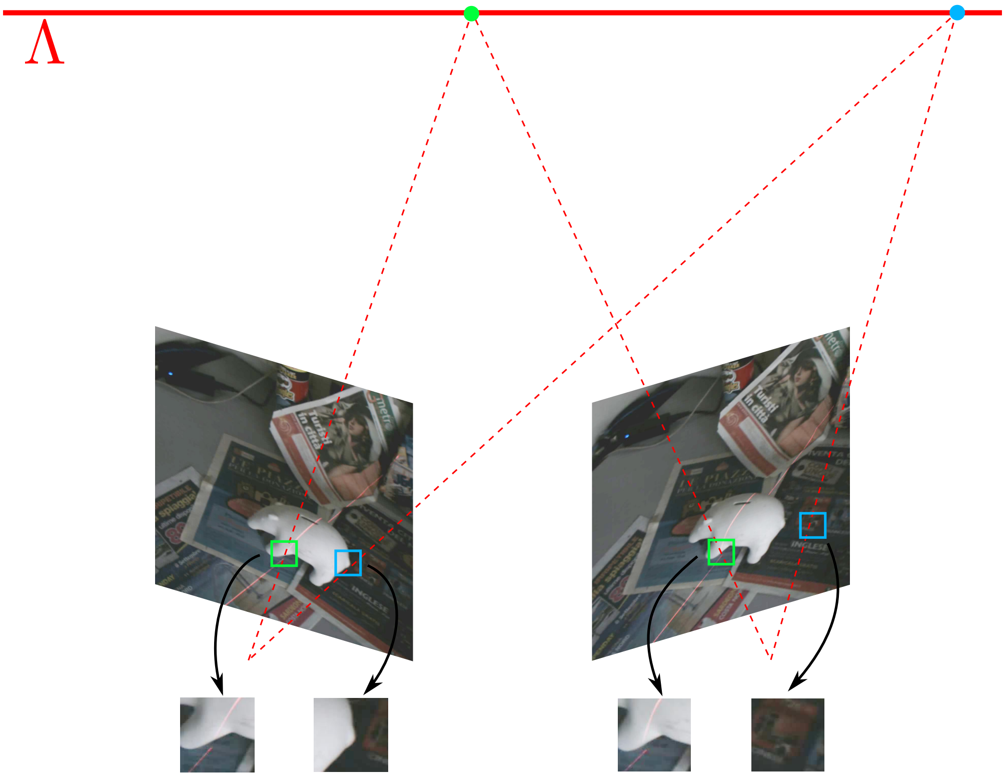


FIGURE 3.6: Stereo laser detection. A laser image point on the left frame and the projection of the triangulated 3D point on the right frame (green boxes) have similar neighborhoods. On the other hand, a wrong detection make that the corresponding point on the right image shows a different appearance (blue boxes).

related to a different camera coordinate frame. A collation step is needed to obtain a complete 3D model of the scanned object.

3.3.2 3D Profile Collation

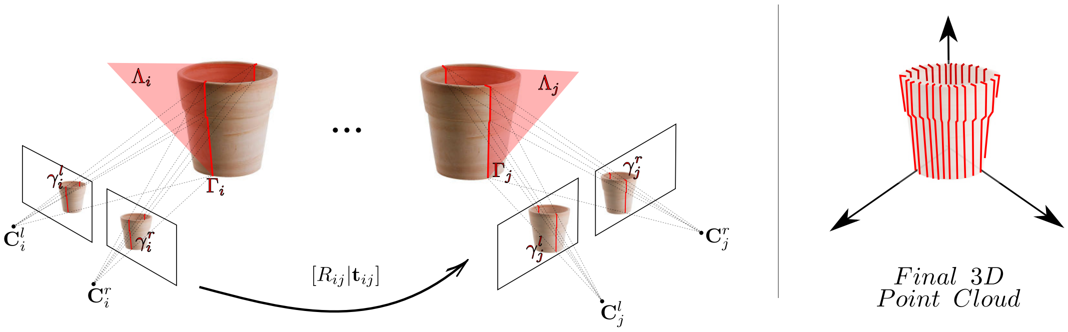


FIGURE 3.7: On the left two examples of 3D profile reconstructed in the i-th and j-th stereo frame. Knowing the motion of the device it's possible to collate all 3D profile in a single three-dimensional model (right).

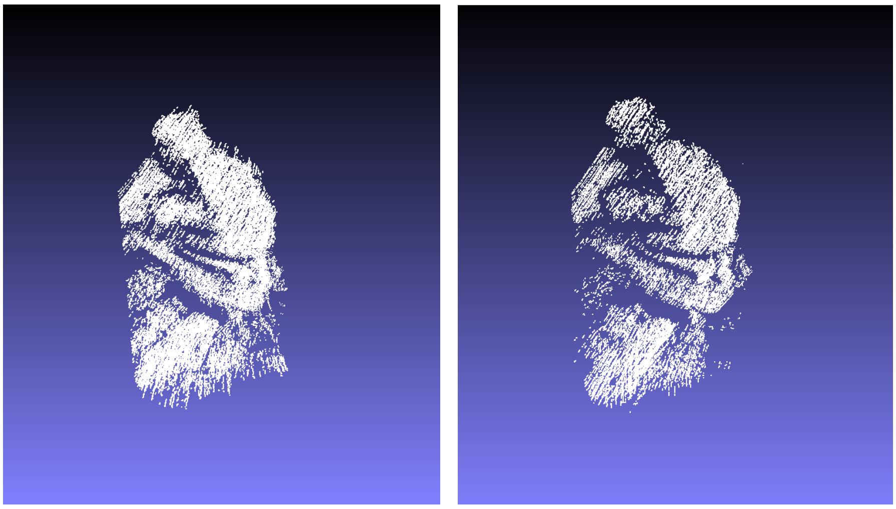
To collate all reconstructed 3D profiles in a global reference frame, the device have to be continuously localized during the acquisition (see Fig. 3.7). In fact, for a generic frame i, once obtained its roto-translation $\{R_i, \mathbf{t}_i\}$ w.r.t. an arbitrary chosen base coordinate frame $\mathcal{B}$, each 3D point $\mathbf{X}_i$ can be remapped to the base frame using

$$\mathbf{X}_{\mathcal{B}} = R_i^{\top}[\mathbf{X}_i - \mathbf{t}_i] \tag{3.8}$$

To estimate the device motion the previously presented visual SLAM system are used. SAMSLAM (Sect. 2.2) is used in the monocular setup, while for the stereo setup SSLAM (Sect. 2.3) is employed. Note also that, as anticipated in the Introduction (Subsect. 1.1.5) accurate motion estimates have to be used to obtain a final model without strong noises. The possibility to use SAMSLAM or SSLAM for this task is another demonstration of the good performances of our SLAM systems.

3.3.3 Refinement

Even if a careful laser detection is performed so to not triangulate wrong points and accurate motion estimates are provided by both the mono and stereo visual SLAM software, errors in the final 3D model are still present—see Fig. 3.8a.



(A)

(B)

Figure 3.8: 3D reconstructed point cloud before (a) and after (b) refinement.

By inspecting the obtained reconstruction once collated all the 3D profiles, the main source of noise—and consequently the unpleasing appearance of the model—is given by slightly misaligned profiles. These effects are caused by local errors during the motion estimation. To improve the final result a refinement step is carried out.

We suppose that during the acquisition the device is moved at constant velocity and images are acquired with a constant frame-rate. Under this assumption its possible to state that each profile should have a similar distance w.r.t. its previous profile. This condition is violated when wrong motion estimate are used. Moreover, given that both SSLAM and SAMSLAM work incrementally on local data, if at time t an erroneous localization is provided, all subsequent estimations will inherit the same error. This behavior leads to a noisy global 3D model composed by coherent sub-models. Exploiting this observation, a refinement technique has been developed.

At first for each 3D profile i its distance D_i w.r.t. the previous profile is computed using

$$D_i = \frac{1}{n_i} \sum_{k=1}^{n_i} dist(\mathbf{X}_i^k, \widehat{\mathbf{X}}_{i-1}^k) \tag{3.9}$$

where n_i is the number of 3D points in the i-th profile and $\widehat{\mathbf{X}}_{i-1}^k$ is the closest point of the previous profile to $\mathbf{X}_i^k$, i.e.

$$\widehat{\mathbf{X}}_{i-1}^k = \underset{h=1,\ldots,n_{i-1}}{\operatorname{argmin}} \mathcal{D}(\mathbf{X}_{i-1}^h) \quad \text{where} \quad \mathcal{D}(\mathbf{X}_{i-1}^h) = dist(\mathbf{X}_i^k, \mathbf{X}_{i-1}^h) \tag{3.10}$$

Once computed all the distances D_i for $i = 2, \ldots, N$, where N is the total number of reconstructed profiles, the mean value μ_D and the standard deviation σ_D are retrieved. A cut in the model is made if

$$D_i > \mu_D + 1.5\sigma_D \tag{3.11}$$

Then sub-models that include few profiles are discarded, while the remaining sub-models are merged again together using the Iterative Closest Point algorithm starting from the bigger ones. This is possible since during acquisition the object is scanned with multiple sweeps so the obtained sub-models are well overlapped. Example of a final cleaned model is shown in Fig. 3.8b.

3.4 Results

Given that, to the best of our knowledge, no datasets are available for this kind of reconstruction approach, to evaluate the accuracy achieved by the proposed hybrid method, measurements were taken on the obtained three-dimensional models and on the real object using an high precision caliber.

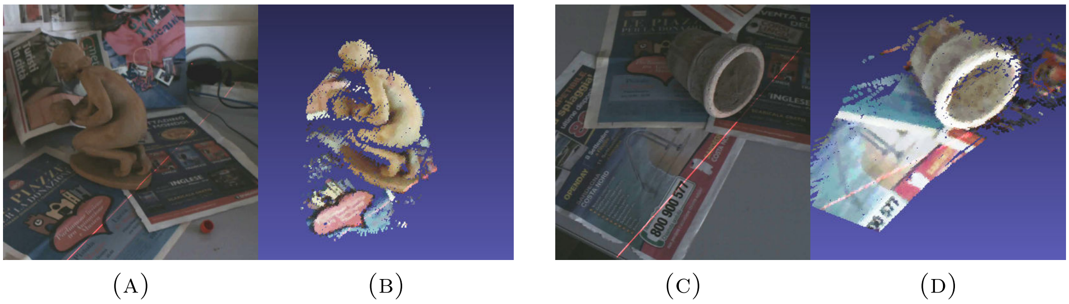


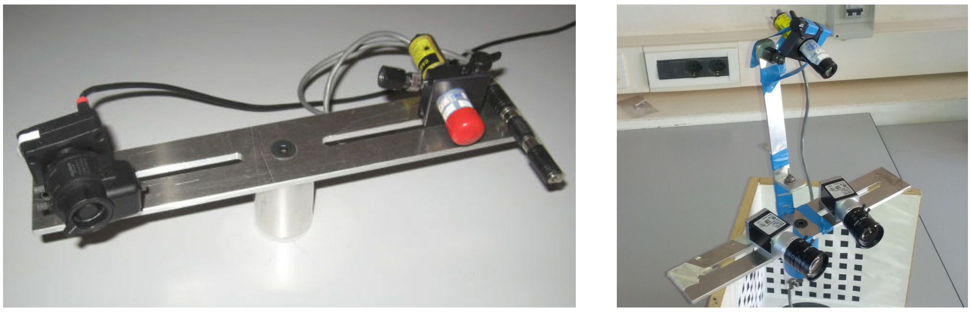
Figure 3.9: Hybrid reconstruction results. Two example reconstruction obtained with the proposed method. See also Fig. 2.8.

Comparing the measurements, both the mono and stereo setups, shows a mean error of 2.5mm. In Fig. 3.9 example of reconstructed objects are shown. To better appreciate the three-dimensional output, texture has been added to the model by sampling the color of the projection of each 3D point into the image sequence.

3.4.1 Monocular *vs* Stereo Setup

As shown, the monocular and stereo setup of the proposed hybrid reconstruction system achieve similar accuracy results. Nevertheless there are differences between the two presented solution: First of all is obvious that the stereo setup requires more efforts in the calibration phase, since in addition to the estimation of the intrinsics camera parameters and the laser plane equation, also an exterior calibration have to be performed so to retrieve the relative geometry between the stereo cameras.

Moreover, considering the cost and the compactness of the device, the monocular setup is clearly preferable respect to the stereo one, given that an additional synchronized camera have to be acquired and mounted—see Fig. 3.10.



(A) (B)

FIGURE 3.10: The monocular (a) and the stereo (b) setups of the prototype device used in our experiment.

TABLE 3.1: Summary of comparison among the presented method and the approaches described in (Winkelbach et al., 2006) and in (Habbecke and Kobbelt, 2008). In the first column the methods are listed. The *Calibration* columns indicate the requirements of *Intrinsics*, *Extrinsic* and *Laser* geometry calibration. The third column shows if the methods allow a variable *point of view* (PoV). In the *Laser Detection* column GB indicate a gradient based detection while BS a background subtraction approach. The *Extra* column shows additional requirements such as simple metric scale reference (SR) or structured 3D pattern present in the scene. Finally we classify the overall flexibility in three classes: Hight, Mid and Low, and in the last column we report the reconstruction errors, where AVG is the average error and RMS is the root mean square error.

Method	Calibration			Movable	Laser	Extra	Overall	Error
	Intr.	Extr.	Laser	PoV	Detection		Flexibility	(mm)
S	X	X	X	Yes	GB	-	High	2.5 (AVG)
M	X	-	X	Yes	GB	SR	Mid	2.5 (AVG)
W	X	X	-	No	BS	3D	Low	0.37 (RMS)
H	X	-	X	No	BS	-	Low	-

S: Proposed stereo hybrid method
M: Proposed mono hybrid method
W: (Winkelbach et al., 2006)
H: (Habbecke and Kobbelt, 2008)

Anyhow its worth noting that in the monocular setup a metric reference has to be included in the scene. In fact using a single camera the 3D scene and the translation magnitude are estimated up to a scale factor; On the contrary, during calibration, the laser plane distance w.r.t. the coordinate frame center is computed with metric precision. This lead to have single laser profiles metrically reconstructed in 3D, but estimated translations with a different scale factor, causing the collation process to fail. To overcome this issue the scale factor in the SLAM system have to be provided exploiting an known reference.

On the other hand employing a fully calibrated stereo pair the scale factor ambiguity is bypassed and there is no need to introduce into the scene a reference object; In this way we could assess that the stereo setup achieve the maximum flexibility of use—also w.r.t. other state-of-the-art active reconstruction method, as reported briefly in Table 3.1.

CHAPTER 4

Conclusion

In this thesis camera localization and three-dimensional structure computation starting from image sequences have been addressed. The main topic focused on the Simultaneous Localization and Mapping (SLAM) problem. SLAM solutions address precisely the task of incrementally estimate the position of a moving camera into an unknown environment—so to compute its trajectory—and, at the same time, obtain a representation of the navigated scene. Starting from approaches emerged into the robotic community in the last years of the 20th century, later many SLAM approaches appear on the computer vision literature—named as Visual SLAM (vSLAM). The work here presented fits into this topic.

In particular two approaches have been presented in Chapter 2: (i) a monocular vSLAM system (SAMSLAM, Sect. 2.2), characterized by an optimization scheme inspired by the Simulated Annealing meta-heuristic and (ii) a stereo vSLAM software (SSLAM, Sect. 2.3) based on an accurate selection of the information to be used in the estimation procedure.

SAMSLAM—Simulated Annealing Monocular SLAM—works incrementally on partially overlapped keyframe triplets. The most innovative idea is the local optimization employed to refine the keyframe estimation: Inspired by the Simulated Annealing scheme, a two steps iterative framework has been developed. In the first phase, 3D maps obtained from each image pair in the triplet are registered with the Horn method made robust using RANSAC. Then inlier 3D points, together with the relative image features, are used as input for the EPnP algorithm—also this included in a RANSAC framework—so to obtain improved camera localization. As the iteration proceeds, threshold used to evaluate the inliers are progressively reduced, with the aim to make the algorithm converges to a minimal error state. Results obtained demonstrates the quality of SAMSLAM for odometry tasks in indoor environment. Moreover, comparing its performances with a local version of the Bundle Adjustment algorithm shows interesting results, obtaining lower or comparable errors thanks to SAMSLAM ability to efficiently deal with poor camera initializations and outlier data.

SSLAM—Selective Stereo SLAM—exploits the stereo camera pair to achieve high accuracy estimations. Employing a robust loop chain matching strategy over two stereo frames—the last detected keyframe and the actual frame—to obtain image correspondences, the pose estimation is based on the minimization of the re-projection error between of a local 3D map computed on the last keyframe and the 2D image features on the current frame. To maintain low the error even for long trajectories, keyframes are carefully chosen by analyzing the temporal feature flow: Only if enough correspondences show an high disparity a new frame is selected as keyframe. Without the need of employ global techniques such as Loop Closure with Bundle Adjustment or graph-based optimization, SSLAM achieves remarkable results in outdoor and even underwater scenarios, reaching the third position in the international KITTI benchmark at the time of submission (actually SSLAM stands in sixth position).

Finally an hybrid 3D reconstruction method has been presented (Ch. 3) employing a prototype device with one or two passive cameras and an active laser fan projector. Exploiting the active triangulation approach together with the motion estimates provided by SAMSLAM or SSLAM (depending on the used setup, monocular or stereo) it's possible to obtain a dense 3D model of a scanned object. Capturing the laser deformations while it interacts with the scene, single three-dimensional profiles can be reconstructed from each video frame. Successively 3D profiles can be collated into a unique model by registering them in a global reference frame using SLAM estimated roto-translations. The reconstructed models show good accuracy and comparing with other active reconstruction methods, the proposed solution achieves the higher flexibility of use, particularly with the stereo setup, since the device can be freely moved by hand and no additional reference object has to be included in the scene.

Bibliography

H. F. Durrant-Whyte and T. Bailey. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. *IEEE ROBOTICS AND AUTOMATION MAGAZINE*, 2:2006, 2006.

R. C. Smith and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainly. *Int. J. Rob. Res.*, 5(4):56–68, December 1986.

H. F. Durrant-Whyte. Uncertain geometry in robotics. *Robotics and Automation, IEEE Journal of*, 4(1):23–31, February 1988.

J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, pages 1442–1447 vol.3+, 1991.

H. F. Durrant-Whyte, D. Rye, and E. Nebot. Localization of Autonomous Guided Vehicles. In *Robotics Research: The 7th International Symposium (ISRR'95)*, pages 613–625, 1996.

R. Chatila and J. Laumond. Position Referencing and Consistent World Modeling for Mobile Robots. In *IEEE International Conference on Robotics and Automation (ICRA).* IEEE Computer Society Press, 1985.

J. L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 674–680 vol.2. IEEE, 1989.

N. Ayache and O. D. Faugeras. Building, registrating, and fusing noisy visual maps. *I. J. Robotic Res.*, 7(6):45–65, 1988.

Z. Chen. Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. Technical report, McMaster University, 2003.

R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 2010.

A. Davison. Real-time simultaneous localization and mapping with a single camera. In *Proc. 9th IEEE International Conference on Computer Vision*, pages 1403–1410, 2003.

J. Montiel, J. Civera, and A. Davison. Unified inverse depth parametrization for monocular SLAM. In *Proc. of Robotics: Science and Systems*. IEEE Press, 2006.

H. Strasdat, J. Montiel, and A. Davison. Visual SLAM: Why filter? *Image and Vision Computing*, 30:65–77, 2012.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

A. M. Farenzena, A. Fusiello, and R. Gherardi. Structure-and-Motion pipeline on a hierarchical cluster tree. In *Proceedings of the IEEE International Workshop on 3-D Digital Imaging and Modeling*, 2009.

S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *Proceedings of the International Conference on Computer Vision*, ICCV ’09, 2009.

R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proc. of the International Workshop on Vision Algorithms: Theory and Practice*, pages 298–372, 2000.

B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós. A comparison of loop closing techniques in monocular slam. *Robot. Auton. Syst.*, 57(12):1188–1197, 2009.

R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A General Framework for Graph Optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

A. Rituerto, L. Puig, and J. J. Guerrero. Visual slam with an omnidirectional camera. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, pages 348–351. IEEE Computer Society, 2010.

D. Zou and P. Tan. CoSLAM: Collaborative visual SLAM in dynamic environments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(2):354–366, Feb. 2013.

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

J. Lim, M. Pollefeys, and J. M. Frahm. Online environment mapping. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. RSLAM: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, 94:198–214, 2011.

A. Nemra and N. Aouf. Robust airborne 3d visual simultaneous localization and mapping with observability and consistency analysis. *J. Intell. Robotics Syst.*, 55(4-5):345–376, August 2009.

I. Mahon, S. Williams, O. Pizarro, and M. Johnson-Roberson. Efficient view-based SLAM using visual loop closures. *IEEE Trans. on Robotics*, 24:1002–1014, 2008.

Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, 2010.

D. Bradley, T. Boubekeur, and W. Heidrich. Accurate multiview reconstruction using robust binocular stereo and surface meshing. In *IN PROC. OF CVPR*, 2008.

M. Fanfani and C. Colombo. Hybrid 3D Reconstruction: SLAM vs Homography-Based Motion Estimation. In *Proc. of 6th Convegno del Gruppo Italiano Ricercatori in Pattern Recognition, GIRPR 2012*, 2012.

M. Fanfani and C. Colombo. LaserGun: A Tool for Hybrid 3D Reconstruction. In *Proc. of 9th International Conference on Computer Vision Systems, ICVS 2013*, pages 274–283, 2013.

M. Fanfani, F. Bellavia, F. Pazzaglia, and C. Colombo. SAMSLAM: Simulated annealing monocular SLAM. In *Proc. of 15th International Conference on Computer Analysis of Images and Patterns, CAIP 2013*, pages 515–522, 2013.

F. Bellavia, M. Fanfani, F. Pazzaglia, and C. Colombo. Robust selective stereo SLAM without loop closure and bundle adjustment. In *Proc. of 17th International Conference on Image Analysis and Processing, ICIAP 2013*, pages 462–471, 2013.

C. G. Harris and J. M. Pike. 3D Positional Integration from Image Sequences. In *In Proc. Alvey Vision Conference, Cambridge.England*, 1987.

Y. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26:639–656, 1990.

A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):523–535, 2002.

E. Eade and T. Drummond. Scalable Monocular SLAM. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 469–476, 2006.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Eighteenth National Conference on Artificial Intelligence*, pages 593–598, 2002.

D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '04, pages 652–659, 2004.

R. M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem. *Int. J. Comput. Vision*, 13(3):331–356, 1994.

E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real Time Localization and 3D Reconstruction. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 363–370, 2006.

C. Mei, G. Sibley, M. Cummins, P. M. Newman, and I. D. Reid. A constant-time efficient stereo slam system. In *Proceedings of the British Machine Vision Conference (BMVC)*. British Machine Vision Association, 2009.

H. Strasdat, J. M. M. Montiel, and A. J. Davison. Scale drift-aware large scale monocular SLAM. In *Proc. of Robotics: Science and Systems*, 2010.

H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige. Double Window Optimisation for Constant Time Visual SLAM. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2352–2359, 2011.

S. Se, D. Lowe, and J. Little. Vision-based Mobile Robot Localization And Mapping using Scale-Invariant Features. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA*, pages 2051–2058, 2001.

C. Beall, B.J. Lawrence, V. Ila, and F. Dellaert. 3D reconstruction of underwater structures. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2010*, pages 4418–4423, 2010.

Z. Zhang and Y. Huang ; C. Li ; Y. Kang. Monocular vision simultaneous localization and mapping using SURF. In *Proceedings of the 7th World Congress on Intelligent Control and Automation, WCICA 2008*, pages 1651 – 1656, 2008.

G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *In Proceedings of the European Conference on Computer Vision, ECCV 2008*, 2008.

V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An Accurate O(N) Solution to the PnP Problem. *Int. J. Comput. Vision*, 81(2):155–166, 2009.

F. Bellavia, D. Tegolo, and C. Valenti. Improving Harris corner selection strategy. *IET Computer Vision*, 5(2), 2011.

F. Bellavia, D. Tegolo, and E. Trucco. Improving SIFT-based descriptors stability to rotations. In *Proc. of International Conference on Pattern Recognition*, 2010.

M.H. DeGroot and M.J Schervish. *Probability and statistics.* Addison Wesley, 3rd edition, 2001.

C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

F. Bellavia, D. Tegolo, and C. Valenti. Keypoint Descriptor Matching with Context-Based Orientation Estimation. *Image and Vision Computing*, 32(9): 559–567, 2014.

B. K. P. Horn. Closed-form Solution of Absolute Orientation Using Unit Quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3D reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, 2011.

G. H. Lee, F. Fraundorfer, and M. Pollefeys. RS-SLAM: RANSAC sampling for visual FastSLAM. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1655–1660, 2011.

D. Nistér, O. Naroditsky, and J. R. Bergen. Visual odometry. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–659, 2004.

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. of Computer Vision and Pattern Recognition*, 2012. URL http://www.cvlibs.net/datasets/kitti/eval_odometry.php.

M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. *The International Journal of Robotics Research*, 28(5):595–599, 2009. ISSN 0278-3649. URL http://www.robots.ox.ac.uk/NewCollegeData/.

H. Badino, A. Yamamoto, and T. Kanade. Visual odometry by multi-frame feature integration. In *Proc of the International Workshop on Computer Vision for Autonomous Driving at ICCV*, 2013.

W. Lu, Z. Xiang, and J. Liu. High-performance visual odometry with two-stage local binocular BA and GPU. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1107–1112. IEEE, 2013.

Submission VISAP. Improving the Egomotion Estimation by Correcting the Calibration Bias. In *VISAPP*, 2015.

B. Allotta, C. Colombo, R. Costanzi, M. Fanfani, A. Ridolfi, M. Reggiannini, O. Salvetti, M. Kruusmaa, T. Salumae, D. M. Lane, G. Frost, N. Tsiogkas, M. Cocco, F. Giardi, L. Gualdesi, G. La Cava, D. Roig Broman, C. Dede, E. L. Tekdemir, H. Tolasa Gundoğdu, S. Baines, S. Tusa, U. Dresen, P. Lätti, and T. Saar. Teams of Robots for Underwater Archaeology: the ARROWS project. In *6th International Congress "Science and Technology for the Safeguard of Cultural Heritage in the Mediterranean Basin"*. IEEE, 2013.

G. Vogiatzis and C. Hernàndez. Video-based, real-time multi view stereo. *Image and Vision Computing*, 2011.

L. Wang, M. Liao, M. Gong, R. Yang, and D. Nistèr. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In *3rd Int. Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 798–805. Springer-Verlag, 2006.

R. Zhang, P. Tsai, J. E. Cryer, and M. Shah. Shape from Shading: A Survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. A low cost 3D scanner based on structured light. *Computer Graphics Forum*, 20:299–308, 2001.

S. Winkelbach, S. Molkenstruck, and F. M. Wahl. Low-Cost Laser Range Scanner and Fast Surface Registration Approach. In *Proceedings of the 28th conference on Pattern Recognition*, DAGM'06, pages 718–728, Berlin, Heidelberg, 2006. Springer-Verlag.

M. Habbecke and L. Kobbelt. LaserBrush: A Flexible Device for 3D Reconstruction of Indoor Scenes. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, pages 231–239, New York, NY, USA, 2008. ACM.

C. Colombo, D. Comanducci, and A. Del Bimbo. Shape reconstruction and texture sampling by active rectification and virtual view synthesis. *Computer Vision and Image Understanding*, 115:161–176, 2011.

J. Y. Bouguet and P. Perona. 3D photography using shadows in dual-space geometry. *International Journal of Computer Vision (IJCV)*, 35:129–149, 1999.

C. Hernàndez, G. Vogiatzis, and R. Cipolla. Multi-view photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 2008.

R. B. Fisher and D. K. Naidu. A comparison of algorithms for subpixel peak detection. In *Image Technology, Advances in Image Processing, Multimedia and Machine Vision*, pages 385–404, 1996.