Noname manuscript No.
(will be inserted by the editor)

A survey on non-filter-based monocular Visual SLAM systems

Georges Younes · **Daniel Asmar** · **Elie Shammas**

Received: date / Accepted: date

Abstract Extensive research in the field of Visual SLAM for the past fifteen years has yielded workable systems that found their way into various applications, such as robotics and augmented reality. Although filter-based (*e.g.*, Kalman Filter, Particle Filter) Visual SLAM systems were common at some time, non-filter based (*i.e.*, akin to SfM solutions), which are more efficient, are becoming the de facto methodology for building a Visual SLAM system. This paper presents a survey that covers the various non-filter based Visual SLAM systems in the literature, detailing the various components of their implementation, while critically assessing the specific strategies made by each proposed system in implementing its components.

Keywords Visual SLAM · monocular · non-filter based.

1 Introduction

Localization solutions using a single camera have been gaining considerable popularity in the past fifteen years. Cameras are ubiquitously found in hand-held devices such as phones and tablets and with the recent increase in augmented reality applications, the camera is the natural sensor of choice to localize the user while projecting virtual scenes to him/her from the correct viewpoint. With their low cost and small size, cameras are frequently used in localization applications where weight and power consumption are deciding factors, such as for Unmanned Aerial Vehicles (UAVs). Even though there are still many challenges facing camera-based localization, it is expected that such solutions will eventually offer significant advantages over other types of localization techniques.

Putting aside localization solutions relying on tracking of markers or objects, camera-based localization can be broadly categorized into two approaches. In what is known as Image-Based Localization (IBL), the scene is processed beforehand to yield its 3D structure, scene images and corresponding camera viewpoints. The localization problem then reduces to that of matching new query images to those in the database and choosing the camera position that corresponds to the best-matched image. In the second technique, no prior information of the scene is given; rather, map building and localization are concurrently done. Here we can incrementally estimate camera pose—a technique known as Visual Odometry (VO) (Scaramuzza and Fraundorfer, 2011); or to reduce the considerable drift that is common in VO, we maintain a map and pose estimate of the camera throughout the journey. This is commonly referred to as Visual Simultaneous Localization and Mapping (Visual SLAM). Although all of the above camera-based techniques are equally important, the subject of this paper is related to the topic of Visual SLAM.

Although a number of surveys for the general SLAM problem exist in the literature, only a few exclusively handle Visual SLAM. In 2012, Fuentes-Pacheco et al (2012) published a general survey on Visual SLAM and but did not delve into the details of the solutions put forward by different people in the community. Also, subsequent to date their paper was published, almost thirteen new systems have been proposed, with many of them introducing significant contributions to Visual SLAM. In 2015, Yousif et al (2015)

G. Younes
E-mail: gyy01@aub.edu.lb

D. Asmar
E-mail: da20@aub.edu.lb

E. Shammas
E-mail: es34@aub.edu.lb

Mechanical Engineering department
American University of Beirut
Beirut, Lebanon

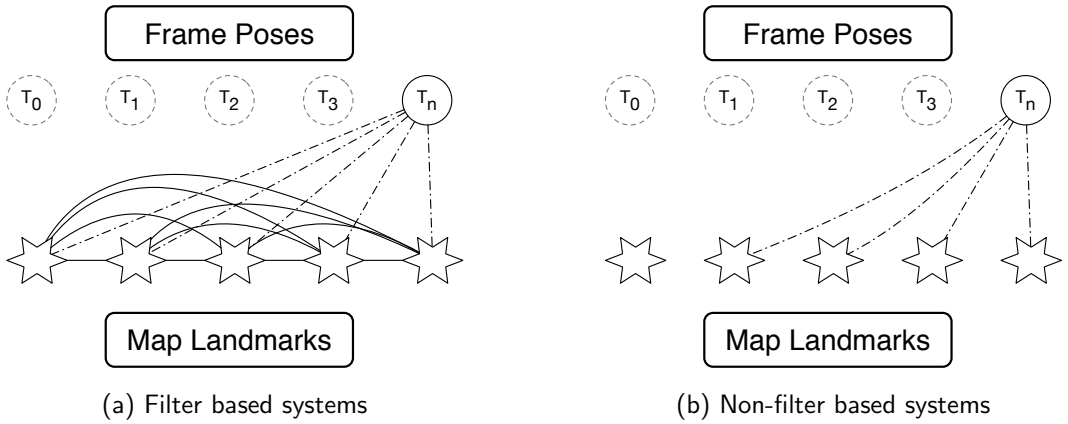


Fig. 1 Data links in filter versus non-filter based Visual SLAM systems. Figure inspired by Strasdat et al (2010b)

also published a general survey on Visual SLAM that includes filter-based, non-filter based, and RGB-D systems. While filter-based Visual SLAM solutions were common before 2010, most solutions thereafter designed their systems around a non-filter-based architecture. The survey of Yousif *et al*. describes a generic Visual SLAM but lacks focus on the details and problems of monocular non-filter based systems. With the above motivations in mind, the purpose of this paper is to survey the state-of-the-art in non-filter-based monocular based Visual SLAM systems. The description of the open-source systems will go beyond the information provided in their papers, but also rely on the understanding and experience we gained modifying and applying their code in real settings. Unfortunately, for the closed source systems we will have to suffice with the information provided in their papers, as well as the insight acquired running their executables (for those who provide them).

A survey as the one proposed in this paper is a valuable tool for any user or researcher in camera-based localization. With the many new proposed systems coming out every day, the information is daunting to the novice and one is often perplexed as to which algorithm he/she should use. Furthermore, this paper should help researchers quickly pinpoint the shortcomings of each of the proposed techniques and accordingly help them focus their effort on alleviating these weaknesses.

The remainder of the paper is structured as follows. Section 2 reviews the historical evolution of Visual SLAM systems, from the time of Mono SLAM (Davison, 2003) to this date. Section 3 describes the fundamental building blocks of a Visual SLAM system and critically evaluates the differences in the proposed open-source solutions; namely in the initialization, measurement and data association, pose estimation, map generation, map maintenance, failure recovery, and loop closure. Section 4 summarizes closed source non-filter based Visual SLAM systems and finally Section 5 concludes the paper.

2 Overview of contributions

Visual SLAM solutions are either filter-based (*e.g*., Kalman filter, Particle filter) or non-filter-based (*i.e*., posing it as an optimization problem). Figure 1a shows the data links between different components of filter-type systems; the camera pose T_n with the entire state of all landmarks in the map are tightly joined and need to be updated at every processed frame. In contrast to non-filter-based systems (shown in Fig. 1b), where the data connections between different components allow the pose estimate of the camera at T_n to be estimated using a subset of the entire map, without the need to update the map's data at every processed frame. As a consequence of these differences, Strasdat *et al*. in 2010 proved that non-filter based methods outperform filter-based ones. It is therefore not surprising that since then, most new releases of Visual SLAM systems are non-filter-based (see Table 1). In this paper we will focus on analyzing only non-filter-based techniques and for filter-based ones we will suffice on listing them.

In 2007, Parallel Tracking and Mapping (Klein and Murray, 2007) was released, and since then many variations and modifications of it have been proposed, such as in Castle et al (2008), Weiss et al (2013), and Klein and Murray (2008). PTAM was the first algorithm to successfully separate tracking and mapping into two parallel computation threads that run simultaneously and share information whenever necessary. This separation made the adaptation of offline Structure from Motion (SfM) methods possible within PTAM in a real-time performance. Its ideas were revolutionary in the monocular visual SLAM community, and the notion of separation between tracking and mapping became the

Table 1 List of different visual SLAM system. Non-filter-based approaches are highlighted in a gray color.

Year	Name	Method	Type	Reference
2003	Real-time simultaneous localization and mapping with a single camera	filter	indirect	Davison (2003)
2004	Simultaneous localization and mapping using multiple view feature descriptors	filter	indirect	Meltzer et al (2004)
2004	Real-Time 3D SLAM with Wide-Angle Vision	filter	indirect	Davison et al (2004)
2005	Real-Time Camera Tracking Using a Particle Filter	filter	indirect	Pupilli and Calway (2005)
2005	CV-SLAM	filter	indirect	Jeong and Lee (2005)
2006	Real-time Localization and 3D Reconstruction	non-filter	indirect	Mouragnon et al (2006)
2006	Scalable Monocular SLAM	filter	indirect	Eade and Drummond (2006)
2006	Real-Time Monocular SLAM with Straight Lines	filter	indirect	Smith et al (2006)
2007	Monocular-vision based SLAM using Line Segments	filter	indirect	Lemaire and Lacroix (2007)
2007	MonoSLAM	filter	indirect	Davison et al (2007)
2007	Parallel Tracking and Mapping (PTAM)	non-filter	indirect	Klein and Murray (2007)
2007	Monocular SLAM as a Graph of Coalesced Observations	filter	indirect	Eade and Drummond (2007)
2007	Mapping Large Loops with a Single Hand-Held Camera	filter	indirect	Clemente et al (2007)
2007	Dimensionless Monocular SLAM	filter	indirect	Civera et al (2007)
2008	A Square Root UKF for visual monoSLAM	filter	indirect	Holmes et al (2008)
2008	An Efficient Direct Approach to Visual SLAM	non-filter	direct	Silveira et al (2008)
2008	Efficient View-Based SLAM Using Visual Loop Closures	filter	indirect	Mahon et al (2008)
2008	Large-Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision	filter	indirect	Pinies and Tardos (2008)
2009	Towards a robust visual SLAM approach: Addressing the challenge of lifelong operation	filter	indirect	Hochdorfer and Schlegel (2009)
2009	Use a Single Camera for Simultaneous Localization And Mapping with Mobile Object Tracking in dynamic environments	filter	indirect	Migliore et al (2009)
2010	On Combining Visual SLAM and Visual Odometry	filter	indirect	Williams and Reid (2010)
2010	Scale Drift-Aware Large Scale Monocular SLAM	non-filter	indirect	Strasdat et al (2010a)
2010	Live dense reconstruction with a single moving camera	non-filter	hybrid	Newcombe and Davison (2010)
2010	Monocular SLAM with locally planar landmarks via geometric rao-blackwellized particle filtering on Lie groups	filter	indirect	Kwon and Lee (2010)
2011	Dense Tracking and Mapping (DTAM)	non-filter	direct	Newcombe et al (2011)
2011	Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation	non-filter	direct	Pretto et al (2011)
2011	Continuous localization and mapping in a dynamic world (CD SLAM)	non-filter	indirect	Pirker et al (2011)
2011	Online environment mapping	non-filter	indirect	Lim et al (2011)
2011	Homography-based planar mapping and tracking for mobile phones	non-filter	indirect	Pirchheim and Reitmayr (2011)
2013	Robust monocular SLAM in Dynamic environments (RD SLAM)	non-filter	indirect	Tan et al (2013)
2013	Handling pure camera rotation in keyframe-based SLAM (Hybrid SLAM)	non-filter	indirect	Pirchheim et al (2013)
2013	Monocular Vision SLAM for Indoor Aerial Vehicles	filter	indirect	Celik and Somani (2013)
2014	Visual SLAM for Handheld Monocular Endoscope	filter	indirect	Grasa et al (2014)
2014	Real-time camera tracking using a particle filter combined with unscented Kalman filters	filter	indirect	Lee (2014)
2014	Semi-direct Visual Odometry (SVO)	non-filter	hybrid	Forster et al (2014)
2014	Large Scale Direct monocular SLAM (LSD SLAM)	non-filter	direct	Engel et al (2014)
2014	Deferred Triangulation SLAM (DT SLAM)	non-filter	indirect	Herrera et al (2014)
2014	Real-Time 6-DOF Monocular Visual SLAM in a Large Scale Environment	non-filter	indirect	Lim et al (2014)
2015	StructSLAM: Visual SLAM With Building Structure Lines	filter	indirect	Zhou et al (2015)
2015	Robust large scale monocular Visual SLAM	non-filter	indirect	Bourmaud and Megret (2015)
2015	ORB SLAM	non-filter	indirect	Mur-Artal et al (2015)
2015	Dense Piecewise Parallel Tracking and Mapping (DPPTAM)	non-filter	direct	Concha and Civera (2015)

standard backbone of almost all visual SLAM algorithms thereafter.

In 2014, SVO (Forster et al, 2014) was published as an open-source implementation of a hybrid system that employs both direct and indirect methods in its proposed solution for solving the Visual SLAM task. Unlike PTAM, SVO requires a high frame rate camera. SVO was designed with the concern of operating on high-end platforms as well as computationally-limited ones such as on-board hardware of a generic Micro Aerial Vehicle (MAV). To achieve such resilience, SVO offers two default configurations, one optimized for speed and the other for accuracy.

Also in 2014, Large Scale Direct monocular SLAM (LSD SLAM) (Engel et al, 2014) was released as an open-source adaptation of the visual odometry method proposed in Engel et al (2013). LSD SLAM employs an efficient probabilistic direct approach to estimate semi-dense maps to be used with an image alignment scheme to solve the SLAM task. In contrast to other methods that use bundle adjustment, LSD SLAM employs a pose graph optimization over $Sim(3)$ as in Kummerle et al (2011), which explicitly represents the scale in the system, allowing for scale drift correction and loop closure detection in real-time. A modified version of LSD SLAM was later showcased running on a mobile platform and another adaptation of the system was presented in Engel et al (2015) for a stereo camera setup. LSD SLAM employs three parallel threads after initialization takes place: tracking, depth map estimation, and map optimization.

In late 2014, short for Deferred triangulation SLAM, DT SLAM (Herrera et al, 2014) was released as an indirect method. Similar to other algorithms, it divides the Visual SLAM task into three parallel threads: tracking, mapping, and bundle adjustment. One of the main contributions in DT SLAM is its ability to estimate the camera pose from 2D and 3D features in a unified framework and suggests a bundle adjustment that incorporates both types of features. This gives DT SLAM robustness against pure rotational movements. Another characteristic of the system is its ability to handle multiple maps with undefined scales and merge them together once a sufficient number of 3D matches are established. In DT SLAM, no explicit initialization procedure is required since it is embedded in the tracking thread; furthermore, it is capable of performing multiple initializations whenever tracking is lost. Since initialization is done automatically whenever the system is lost, data can still be collected and camera tracking functions normally, albeit at a different scale. This ability to re-initialize local sub-maps reduces the need for re-localization procedures. Once a sufficient number of correspondences between keyframes residing in separate sub-maps are found, the sub-maps are fused into a single map with a uniform scale throughout.

In 2015, ORB SLAM (Mur-Artal et al, 2015) was released as an indirect Visual SLAM system. It divides the Visual SLAM problem into three parallel threads, one for tracking, one for mapping, and a third for map optimization. The main contributions of ORB SLAM are the usage of ORB features (Rublee et al, 2011) in real-time, a model based initialization as suggested by Torr et al (1999), re-localization with invariance to viewpoint changes (Mur-Artal and Tardós, 2014), a place recognition module using bags of words to detect loops, and covisibility and Essential graph optimization.

In late 2015, short for Dense Piecewise Parallel tracking and Mapping (DPPTAM) (Concha and Civera, 2015), was released as a semi-dense direct method similar to LSD SLAM, to solve for the Visual SLAM task. A key contribution of DPPTAM's adaptation of LSD SLAM, is the added third parallel thread that performs dense reconstructions using segmented super-pixels from indoor planar scenes.

3 Design of Visual SLAM systems

In an effort to better understand Visual SLAM state-of-the-art implementations, this section provides a look under the hood of the most successful and recent open-source non-filter-based Visual SLAM systems. More specifically, our discussion will be based on information extracted from PTAM, SVO, DT SLAM, LSD SLAM, ORB SLAM, and DPPTAM. A generic non-filter Visual SLAM system is concerned with eight main components (Fig. 2); namely (1) input data type, (2) data association, (3) initialization, (4) pose estimation, (5) map generation, (6) map maintenance, (7) failure recovery, and (8) loop closure.

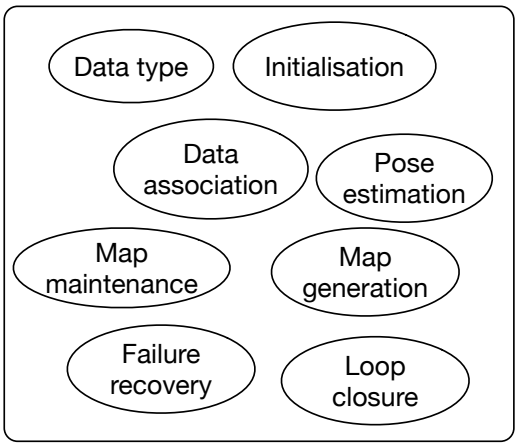


Fig. 2 Eight components of a non-filter-based Visual SLAM system.

In the following sections, we will detail each of these components and critically assess how each Visual SLAM implementation addressed them. It is noteworthy to first

mention that all discussed systems implicitly assume the intrinsic parameters are known based on an off-line calibration step.

3.1 Input data type

Vision SLAM methods are categorized as being direct, indirect, or a hybrid of both. Direct methods—also known as dense or semi-dense methods—exploit the information available at every pixel in the image (brightness values) to estimate the parameters that fully describe the camera pose. On the other hand, indirect methods were introduced to reduce the computational complexity of processing each pixel; this is achieved by using only salient image locations (called features) in the pose estimation calculations (see Fig. 3).

3.1.1 Direct methods

The basic underlying principle for all direct methods is known as the brightness consistency constraint and is best described as:

$$J(x,y) = I(x+u(x,y)+v(x,y)), \tag{1}$$

where x and y are pixel coordinates; u and v denotes displacement functions of the pixel (x,y) between two images I and J of the same scene. Every pixel in the image provides one brightness constraint; however, it adds two unknowns (u and v) and hence the system becomes under-determined with n equations and $2n$ unknowns (where n is the number of pixels in the image). To render (1) solvable, Lucas & Kanade (Lucas and Kanade, 1981) suggested in 1981, in what they referred to as Forward Additive Image Alignment (FAIA), to replace all the individual pixel displacements u and v by a single general motion model, in which the number of parameters is dependent on the implied type of motion. FAIA iteratively minimize the squared pixel-intensity difference between a template and an input image by changing the transformation parameters. Since that time and to reduce computational complexity, other variants of the FAIA were suggested such as FCIA (Forward Compositional Image Alignment), ICIA (Inverse Compositional Image Alignment) and IAIA (Inverse Additive Image Alignment) (Baker and Matthews, 2004).

Direct methods exploit all information available in the image and are therefore more robust than indirect methods in regions with poor texture. Nevertheless, direct methods are susceptible to failure when scene illumination changes occur as the minimization of the photometric error between two frames relies on the underlying assumption of the brightness consistency constraint (1). A second disadvantage is that the calculation of the photometric error at every pixel is computationally intensive; therefore, real-time Visual SLAM applications of direct methods, until recently, were not considered feasible. With the recent advancements in parallelized processing, adaptations of direct methods were integrated within a Visual SLAM context (Concha and Civera, 2015; Engel et al, 2015; Forster et al, 2014).

3.1.2 Indirect methods

Indirect methods rely on features for matching. On one hand, features are expected to be distinctive and invariant to viewpoint and illumination changes, as well as resilient to blur and noise. On the other hand, it is desirable for feature extractors to be computationally efficient and fast. Unfortunately, such objectives are hard to achieve at the same time and a trade-off between computational speed and feature quality is required.

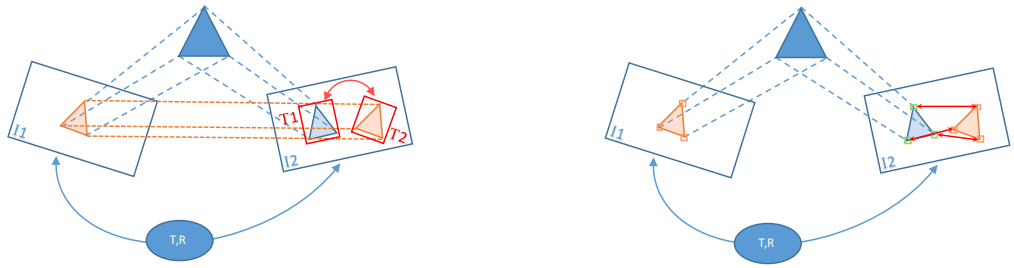
The computer vision community had developed over decades of research many different feature extractors and descriptors, each exhibiting varying performances in terms of rotation and scale invariance, as well as speed. The selection of an appropriate feature detector depends on the platform's computational power, the environment in which the Visual SLAM algorithm is due to operate in, as well as its expected frame rate. Feature detector examples include Hessian corner detector (Beaudet, 1978), Harris detector (Harris and Stephens, 1988), Shi-Tomasi corners (Shi and Tomasi, 1994), Laplacian of Gaussian detector (Lindeberg, 1998), MSER (Matas et al, 2002), Difference of Gaussian (Lowe, 2004) and the accelerated segment test family of detectors (FAST, AGAST, OAST) (Mair et al, 2010).

To minimize computational requirements, most indirect systems use FAST (Rosten and Drummond, 2006) as a feature extractor, coupled with a feature descriptor to be able to perform data association. Feature descriptors include and are not limited to BRIEF (Calonder et al, 2012), BRISK (Leutenegger et al, 2011), SURF (Bay et al, 2008), SIFT (Lowe, 1999), HoG (Dalal and Triggs, 2005), FREAK (Alahi et al, 2012), ORB (Rublee et al, 2011) and a low level local patch of pixels. Further information regarding feature extractors and descriptors are outside the scope of this work, but the reader can refer to Hartmann et al (2013), Moreels and Perona (2007), Rey-Otero et al (2014), or Hietanen et al (2016) for the most recent comparisons.

3.1.3 Hybrid methods

Different from the direct and indirect methods, systems such as SVO are considered hybrids, which use a combination of direct methods to establish feature correspondences and indirect methods to refine the camera pose estimates.

Table 2 summarizes the data types used by the selected Visual SLAM systems. From the list of open-source indirect methods surveyed in this paper, PTAM, SVO and DT



(a) direct methods using all information of the triangle to match to a query image.

(b) indirect methods use the features of the triangle to match to the features of a query image.

Fig. 3 Data types used by a Visual SLAM system.

SLAM use FAST features (Rosten and Drummond, 2006), while ORB SLAM uses ORB features (Rublee et al, 2011).

Table 2 Method used by different Visual SLAM systems. Abbreviations used: indirect (i), direct (d), and hybrid (h)

	PTAM	SVO	DT SLAM	LSD SLAM	ORB SLAM	DPPTAM
Method	I	H	I	D	I	D

3.2 Data association

Data association is defined as the process of establishing measurement correspondences across different images to serve as inputs to other Visual SLAM modules. This step is inherent in systems that employ direct methods and hence the discussion on data association for direct methods will be included in the camera pose section below.

To establish feature correspondences in indirect methods, three types of data association can be distinguished: 2D-2D, 3D-2D and 3D-3D.

3.2.1 2D-2D

When a map is not available, and neither the camera transformation between the two frames nor the scene structure is available, 2D-2D data association is used. To reduce the computation time and avoid the possibility of erroneous data association, the feature's 2D location in the first image is used to define a search window in the second image. Each feature has associated with it a descriptor, which can be used to provide a quantitative measure of similarity to other features. The descriptor distance function varies with the type of descriptors used: for the local patch of pixels descriptor, it is typical to use the sum of squared difference (SSD), or to increase robustness against illumination changes, a Zero-Mean SSD score (ZMSSD) is used (Jérôme Martin, 1995). For higher order feature descriptors such as ORB, SIFT, and SURF, the L1-norm, L2-norm, or Hamming distances may be used; however, establishing matches using these measures is computationally intensive and may degrade real-time operations if not carefully applied. For such a purpose, special implementations that sort and perform feature matching in KD trees or bags of words, are usually employed. Examples include the works of Muja and Lowe (2009), and Galvez-López and Tardos (2012).

3.2.2 3D-2D

Figure 4 represents the 3D-2D data association problem, where the previous camera pose estimate and the 3D structure are known and one seeks to estimate correspondences between 2D features and the projection of 3D landmarks onto a newly acquired frame without the knowledge of the exact motion between the frames. This type of data association is typically used during the regular pose estimation phase of Visual SLAM.

3.2.3 3D-3D

To estimate and correct accumulated drift along loops, 3D-3D data association is required, and for such a purpose, descriptors of 3D landmarks that are observed in both frames are used to establish matches among the landmarks that are then exploited–as explained in (Horn, 1987)–to yield a similarity transform between both frames. Table 3 summarizes the feature types and descriptors employed by various open-source Visual SLAM systems.

PTAM. After a successful initialization of PTAM, a 4 level pyramid representation of every incoming frame is

Table 3 Feature extractors and descriptors. Abbreviations:local patch of pixels (L.P.P.)

	PTAM	SVO	DT SLAM	LSD SLAM	ORB SLAM	DPPTAM
Feature type	FAST	FAST	FAST	None	FAST	None
Feature descriptor	L.P.P.	L.P.P.	L.P.P.	L.P.P.	ORB	L.P.P.

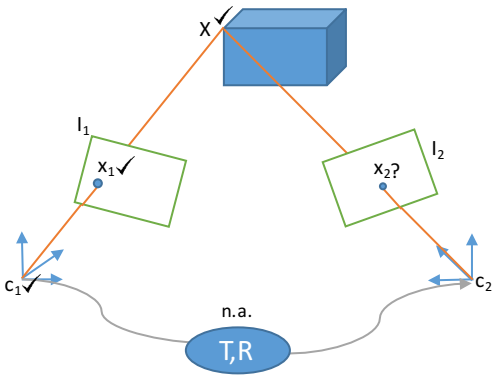


Fig. 4 3D-2D Data association problem (n.a. stands for not available during the task).

generated (*e.g.*, level 1: 640x480, level 2: 320x240). The pyramid levels are used to make features more robust against scale changes and to decrease the convergence radius of the pose estimation module, as shall be described later. FAST features are extracted at each level and a Shi-Tomasi score (Shi and Tomasi, 1994) for each feature is estimated; features having a Shi-Tomasi score below a threshold are removed before non-maximum suppression takes place. This is done to ensure high saliency of the extracted features and limit their numbers–in order to remain computationally tractable.

Each pyramid levels has a different threshold for Shi-Tomasi score selection and non-maximum suppression; thereby giving control over the strength and the number of features to be tracked across the pyramid levels. 3D landmarks are then projected onto the new frame using a pose estimate prior and in a similar manner to the 2D-2D methods, feature correspondences are established within a search window surrounding the projected landmark location. The descriptors used for feature matching of the 3D landmarks are usually extracted from the 2D image from which the 3D landmark was first observed; however some systems propose to update this descriptor as the camera view point observing it changes significantly or, in the case of local patch of pixels are warped to virtually account for view point changes.

DT SLAM. In a similar scheme to PTAM, DT SLAM employs the same mechanism to establish 2D-3D feature matches.

SVO. SVO generates a five level pyramid representation of the incoming frame: data association is first established through an iterative direct image alignment scheme starting from the highest pyramid level up till the third. Preliminary data association from this step is used as a prior to a FAST feature matching procedure, similar to PTAM's warping technique, with a Zero-Mean SSD score.

ORB SLAM. ORB SLAM extracts FAST corners throughout eight pyramid levels. To ensure a homogeneous distribution along the entire image, each pyramid level is divided into cells and the parameters of the FAST detector are tuned on-line to ensure a minimum of five corners are extracted per cell. A 256-bit ORB descriptor is then computed for each extracted feature. The higher order feature descriptor ORB is used to establish correspondences between features. ORB SLAM discretizes and stores the descriptors into bags of words, known as visual vocabulary (Galvez-López and Tardos, 2012), which are used to speed up image and feature matching by constraining those features that belong to the same node in the vocabulary tree.

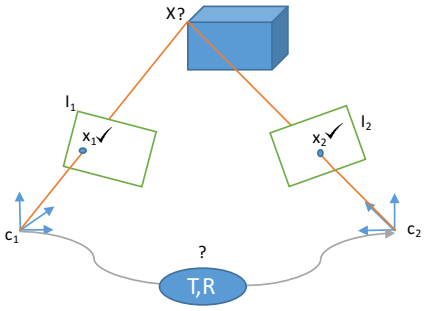


Fig. 5 Initialization required by any Visual SLAM system.

3.3 Initialization

Monocular Visual SLAM systems require an initialization phase, during which both a map of 3D landmarks and the starting camera poses are generated. To do so, the same scene must be observed through at least two viewpoints separated by a baseline. Figure 5 represents the initialization that is required in any Visual SLAM system, where only associated data between two images is known and both the initial camera pose and the scene's structure are unknown.

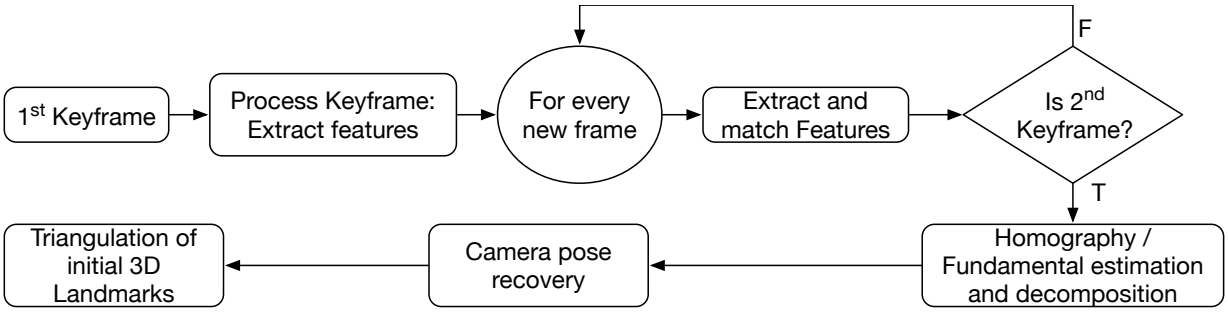


Fig. 6 Generic initialization pipeline.

Different solutions to this problem were proposed by different people.

In early Visual SLAM systems such as in MonoSLAM (Davison et al, 2007), system initialization required the camera to be placed at a known distance from a planar scene composed of four corners of a two dimensional square, and SLAM was initialized with the distance separating the camera from the square keyed in by the operator.

PTAM. Figure 6 shows the flowchart of a generic model-based initialization procedure, such as the one employed in PTAM, SVO and ORB SLAM. To eliminate the obligation of a user's manual input of depth, PTAM's (Klein and Murray, 2007) initial release suggested the usage of the five-point algorithm (Nistér, 2004) to estimate and decompose a Fundamental matrix into an assumed non-planar initial scene. PTAM's initialization was later changed to the usage of a Homography (Faugeras and Lustman, 1988), where the scene is assumed to be composed of 2D planes. PTAM's initialization requires the user's input twice to capture the first two keyframes in the map; furthermore, it requires the user to perform, in between the first and the second keyframe, a slow, smooth and relatively significant translational motion parallel to the observed scene.

FAST Features extracted from the first keyframe are tracked in a 2D-2D data association scheme in each incoming frame until the user flags the insertion of the second keyframe. As the matching procedure takes place through the ZMSSD without warping the features, establishing correct matches is susceptible to both motion blur and significant appearance changes of the features caused by camera rotations; hence the strict requirements on the user's motion during the initialization.

To ensure minimum false matches, the features are searched for twice; once from the current frame to the previous frame and a second time in the opposite direction. If the matches in both directions are not coherent, the feature is discarded. Since PTAM's initialization employs a Homography estimation, the observed scene during the initialization is assumed to be planar. Once the second keyframe is successfully incorporated into the map, a MLESAC (Torr and Zisserman, 2000) loop uses the established matches to generate a Homography relating both keyframes and uses inliers to refine it before decomposing it (as described in (Faugeras and Lustman, 1988)) into eight possible solutions. The correct pair of camera poses is chosen such that all triangulated 3D points do not generate unreal configurations (negative depths in both frames).

The generated initial map is scaled such as the estimated translation between the first two keyframes corresponds to 0.1 units, before a structure-only BA (optimize only the 3D poses of the landmarks) step takes place. The mean of the 3D landmarks is selected to serve as the world coordinate frame, while the positive z-direction is chosen such as the camera poses reside along its positive side.

PTAM's initialization procedure is brittle and remains tricky to perform, especially for inexperienced users. Furthermore, it is subject to degeneracies when the planarity of the initial scene's assumption is violated or when the user's motion is inappropriate, crashing the system, without means of detecting such degeneracies.

SVO. Similarly, Forster et al (2014) adopted in SVO, a Homography for initialization, however, SVO requires no user input and the algorithm uses at startup the first acquired acquired keyframe; it extracts FAST features and tracks them with an implementation of KLT (Tomasi and Kanade, 1991) (variant of direct methods) across incoming frames. To avoid the need for a second input by the user, SVO monitors the median of the baseline of the features tracked between the first keyframe and the current frame; whenever this value reaches a certain threshold, the algorithm assumes enough parallax has been achieved and signals the Homography estimation to start. The Homography is then decomposed; the correct camera poses are then selected and the landmarks corresponding to inlier matches are triangulated and used to estimate an initial scene depth. Bundle Adjustment takes place for the two frames and all their associated landmarks, before the second frame is used as a second keyframe and passed to the map management thread.

As is the case in PTAM, the initialization of SVO requires the same type of motion and is prone to sudden movements as well as to non-planar scenes; furthermore, monitor-

ing the median of the baseline between features is not a good approach to automate the initial keyframe pair selection as it is prone to failure against degenerate cases, with no means of detecting them.

DT SLAM. DT SLAM does not have an explicit initialization phase; rather, it is integrated within its tracking module as an Essential matrix estimation method.

Table 4 summarizes the initialization methods employed by different Visual SLAM systems along with the assumption they make about the configuration of the scene at startup of the system. With the exception of MonoSLAM, all the suggested methods described above suffer from degeneracies when subjected to certain scenes; namely under low-parallax movements of the camera or when the scene's structure assumption for the corresponding method–Fundamental matrix's assumption for general non-planar scenes or the Homography's assumption of planar scenes–is violated.

LSD SLAM. To address this issue, Engel et al (2014) suggested in LSD SLAM, a randomly initialized scene's depth from the first viewpoint, that is later refined through measurements across subsequent frames. LSD SLAM uses an initialization method that does not require two view geometry. Instead of tracking features across two frames, as the other systems do, LSD SLAM initialization procedure takes place on a single frame; pixels of interest (*i.e.*, image locations that have high intensity gradients) are initialized into the system with a random depth distribution and a large variance. Tracking starts directly as image alignment takes place between the first initialized keyframe and preceeding frames. Using the incoming frames and until convergence, the depth measurements of the initialized features are refined using a filter-based scheme. This method does not suffer from the degeneracies of two view geometry methods; however, depth estimation requires a relatively large number of processed frames before convergence takes place, resulting in an intermediate tracking phase where the generated map is not reliable.

DPPTAM. DPPTAM, Concha and Civera (2015) borrows from LSD SLAM its initialization procedure, and therefore it also suffers from the random depth initialization symptoms, where several keyframes must be added to the system before it reaches a stable configuration.

To deal with the limitations arising from all the above methods, Mur-Artal et al (2015) suggested to compute in parallel, both a Fundamental matrix and a Homography (in a RANSAC scheme), while penalizing each model according to its symmetric transfer error (Hartley and Zisserman, 2003), in order to select the appropriate model. Once this is done, appropriate decomposition takes place and both the scene structure and the camera poses are recovered, before a bundle adjustment step optimizes the map. If the chosen model yields poor tracking quality and few feature correspondences in the upcoming frame, the initialization is quickly discarded by the system and it restarts with a different pair of frames. It is noteworthy to mention that the relationship between image coordinates and corresponding 3D point coordinates in all the listed initialization methods, aside that of monoSLAM, can only be determined up to an unknown scale λ.

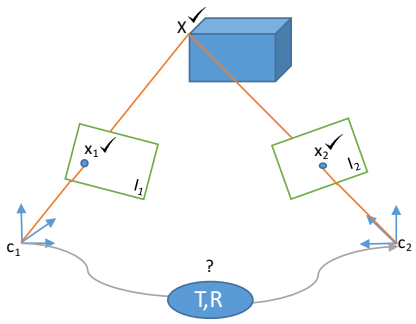


Fig. 7 Pose estimation required by any Visual SLAM system

3.4 Pose estimation

Because data association is computationally heavy, most Visual SLAM systems assume, for the pose of each new frame, a prior, which guides and limits the amount of work required for data association. Estimating this prior is generally the first task in pose estimation. Figure 7 depicts the pose estimation problem; a map and data association between two frames are known and one seeks to estimate the pose of the second frame, given the pose of the first. PTAM, DT SLAM, ORB SLAM, and DPPTAM employ a *constant velocity* motion model that assumes a smooth camera motion and uses the pose changes across the two previously tracked frames to estimate the prior for the current frame. Unfortunately, such a model is prone to failure when sudden change in direction of the camera's motion occurs. LSD SLAM and SVO assume no significant change in the camera pose between consecutive frames (such as the case in high frame rate cameras), and hence they assign the prior for the pose of the current frame to be the same as the previously tracked one.

Figure 8 presents a generic pipeline for pose estimation. The pose of the prior frame is used to guide the data association procedure in several ways. It helps determine a potentially visible set of features from the map in the current frame, thereby reducing the computational expense of blindly projecting the entire map. Furthermore, it helps establish an estimated feature location in the current frame, such that feature matching takes place in small search regions, instead of across the entire image. Finally, it serves as a starting point for the minimization procedure, which refines the camera pose.

Table 4 Initialization. Abbreviations used: homography decomposition (h.d.), Essential decomposition (e.d.), random depth initialization (r.d.), planar (p), non-planar (n.p.), no assumption (n.a.)

	PTAM	SVO	DT SLAM	LSD SLAM	ORB SLAM	DPPTAM
Initialization	h.d.	h.d.	e.d.	r.d.	h.d.+e.d.	r.d.
Initial scene assumption	p	p	n.p.	n.a.	n.a.	n.a.

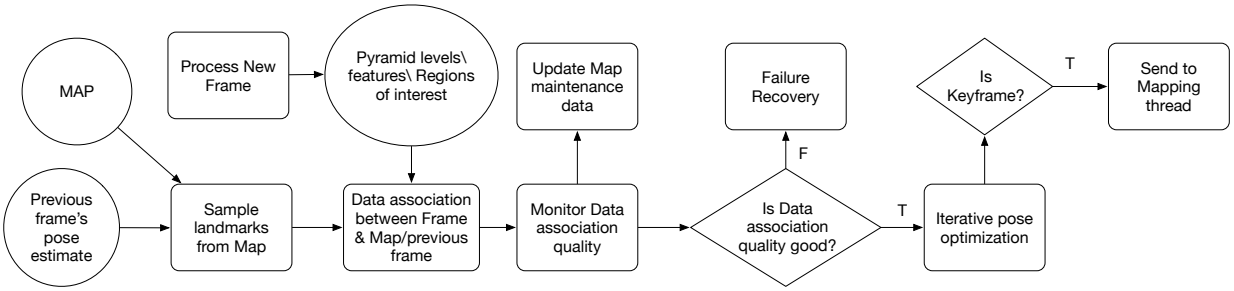


Fig. 8 Generic pose estimation pipeline.

Direct and indirect methods estimate the camera pose by minimizing a measure of error between frames; direct methods measure the photometric error and indirect methods estimate the camera pose by minimizing the re-projection error of landmarks from the map over the frame's prior pose. The re-projection error is formulated as the distance in pixels between a projected 3D landmark onto the frame using the prior pose and its found 2-D position in the image.

Note in Fig. 9 how camera pose estimation takes place. The motion model is used to seed the new frame's pose at C_m, and a list of potentially visible 3D landmarks from the map are projected onto the new frame. Data association takes place in a search window S_w surrounding the location of the projected landmarks. The system then proceeds by minimizing the re-projection error d over the parameters of the rigid body transformation. To gain robustness against outliers (wrongly associated features), the minimization takes place over an objective function that penalizes features with large re-projection errors.

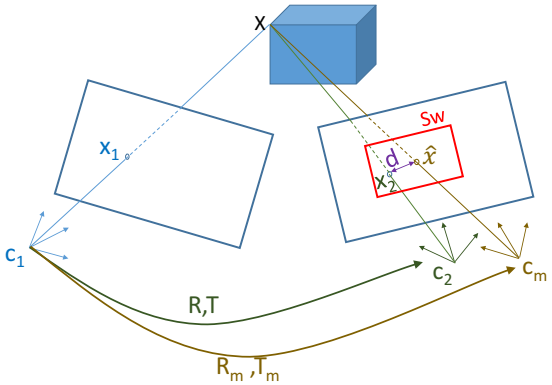


Fig. 9 Generic pose estimation procedure. C_m is the new frame's pose estimated by the motion model and C_2 is the actual camera pose.

PTAM. PTAM represents the camera pose as an $SE(3)$ transformation (Hall, 2015) that can be minimally represented by six parameters. The mapping from the full $SE(3)$ transform to its minimal representation $S\xi(3)$ and vice versa can be done through logarithmic and exponential mapping in Lie algebra. The minimally represented $S\xi(3)$ transform is of great importance as it reduces the number of parameters to optimize from twelve to six, leading to significant speedups in the optimization process.

In PTAM, the pose estimation procedure first starts by estimating a prior to the frame's pose using the constant velocity motion model. The prior is then refined, using a Small Blurry Image (SBI) representation of the frame, by employing an *Efficient Second Order minimization* (Benhimane and Malis, 2007). The velocity of the prior is defined as the change between the current estimate of the pose and the previous camera pose. If the velocity is high, PTAM anticipates a fast motion is taking place and hence the presence of motion blur; to counter failure from motion blur, PTAM restricts tracking to take place only at the highest pyramid levels (most resilient to motion blur) in what is known as a coarse tracking stage only; otherwise the coarse tracking stage is followed by a fine tracking stage. However, when the camera is stationary, the coarse stage may lead to jittering of the camera's pose–hence it is turned off.

The minimally represented initial camera pose prior is then refined by minimizing the tukey-biweight (Moranna et al, 2006) objective function of the re-projection error that down-weights observations with large error. If fine tracking is to take place, features from the lowest pyramid levels are selected and a similar procedure to the above is repeated.

To determine the tracking quality, the pose estimation thread in PTAM monitors the ratio of successfully matched features in the frame against the total number of attempted

Table 5 Pose estimation. Abbreviations are as follows: constant velocity motion model (c.v.m.m), same as previous pose (s.a.p.p.), similarity transform with previous frame (s.t.p.f.), optimization through minimization of features (o.m.f.), optimization through minimization of photometric error (o.m.p.e.), Essential matrix decomposition (E.m.d.), pure rotation estimation from 2 points (p.r.e.), significant pose change (s.p.c.), significant scene appearance change (s.s.a.c)

	PTAM	SVO	DT SLAM	LSD SLAM	ORB SLAM	DPPTAM
Motion prior	c.v.m.m. +ESM	s.a.p.p.	s.t.p.t.	s.a.p.p.	c.v.m.m. or place recogn.	c.v.m.m. or s.a.p.p.
Tracking	o.m.f.	o.m.p.e.	3 modes: 1-e.m.d.; 2-o.m.f.; 3-p.r.e.	o.m.p.e.	o.m.f.	o.m.p.e.
keyframe add criterion	s.p.c.	s.p.c.	s.s.a.c.	s.p.c.	s.s.a.c.	s.p.c.

feature matches. If the tracking quality is questionable, the tracking thread operates normally but no keyframes are accepted by the system. If the tracker's performance is deemed bad for 3 consecutive frames, then the tracker is considered lost and failure recovery is initiated.

Table. 5 summarizes the pose estimation methods used by different Visual SLAM systems.

SVO. SVO uses a sparse model-based image alignment in a pyramidal scheme in order to estimate an initial camera pose estimate. It starts by assuming the camera pose at time t to be the same as at $t-1$ and aims to minimize the photometric error of 2D image locations of known depth in the current frame with respect to their location at $t-1$, by varying the camera transformation relating both frames. The minimization takes places through thirty Gauss Newton iterations of the inverse compositional image alignment method. This however introduces many limitations to SVO since the ICIA requires small displacements between frames ($1 pixel$). This limits the operation of SVO to high frame rate cameras (typically $> 70 fps$) so that the displacement limitation is not exceeded. Furthermore, the ICIA is based on the brightness consistency constraint rendering it vulnerable to any variations in lighting conditions.

SVO does not employ explicit feature matching for every incoming frame; rather, it is achieved implicitly as a byproduct of the image alignment step. Once image alignment takes place, landmarks that are estimated to be visible in the current frame, are projected onto the image. The 2D location of the projected landmarks are fine-tuned by minimizing the photometric error between a patch, extracted from the initial projected location in the current frame, and a warp of the landmark generated from the nearest keyframe observing it. To decrease the computational complexity and to maintain only the strongest features, the frame is divided into a grid and only one projected landmark (the strongest) per grid cell is used. However, This minimization violates the epipolar constraint for the entire frame and further processing in the tracking module is required. Motion-only Bundle Adjustment then takes place, followed by a structure only Bundle Adjustment that refines the 3D location of the landmarks based on the refined camera pose of the previous step.

Finally, a joint (pose and structure) local bundle adjustment fine tunes the reported camera pose estimate. During this pose estimation module, the tracking quality is continuously monitored; if the number of observations in a frame is a below a certain threshold or if the number of features between consecutive frames drops drastically, tracking quality is deemed insufficient and failure recovery methods are initiated.

DT SLAM. DT SLAM maintains a camera pose based on three tracking modes: full pose estimation, Essential matrix estimation, and pure rotation estimation. When a sufficient number of 3D matches exist, a full pose can be estimated; otherwise, if a sufficient number of 2D matches are established that exhibit small translations, an Essential matrix is estimated; and finally, if a pure rotation is exhibited, 2 points are used to estimate the absolute orientation of the matches (Kneip et al, 2012). The pose estimation module finally aims, in an iterative manner, to minimize the error vector of both 3D-2D re-projections and 2D-2D matches. When tracking failure occurs, the system initializes a new map and continues to collect data for tracking in a different map; however, the map making thread continues to look for possible matches between the keyframes of the new map and the old one, and once a match is established, both maps are fused together, thereby allowing the system to handle multiple sub-maps, each at a different scale.

LSD SLAM. The tracking thread in LSD SLAM is responsible for estimating the current frame pose with respect to the currently active keyframe in the map using the previous frame pose as a prior. The required pose is represented by an $SE(3)$ transformation and is found by an iteratively re-weighted Gauss-Newton optimization that minimizes the variance normalized photometric residual error, as described in (Engel et al, 2013), between the current frame and the ac-

tive keyframe in the map. A keyframe is considered active if it is the most recent keyframe accommodated in the map. To minimize outlier effects, measurements with large residuals are down-weighted from one iteration to the other.

ORB SLAM. Pose estimation in ORB SLAM is established through a constant velocity motion model prior, followed by a pose refinement using optimization. As the motion model is expected to be easily violated through abrupt motions, ORB SLAM detects such failures by tracking the number of matched features; if it falls below a certain threshold, map points are projected onto the current frame and a wide range feature search takes place around the projected locations. If tracking fails, ORB SLAM invokes its failure recovery method to establish an initial frame pose via global re-localization.

In an effort to make ORB SLAM operate in large environments, a subset of the global map, known as the local map, is defined by all landmarks corresponding to the set of all keyframes that share edges with the current frame, as well as all neighbors of this set of keyframes from the pose graph (more on that in the following section). The selected landmarks are filtered out to keep only the features that are most likely to be matched in the current frame. Furthermore, if the distance from the camera's center to the landmark is beyond the range of the valid features scales, the landmark is also discarded. The remaining set of landmarks is then searched for and matched in the current frame before a final camera pose refinement step takes place.

DPPTAM. Similar to LSD SLAM, DPPTAM optimizes the photometric error of high gradient pixel locations between two images using the ICIA formulation over the $SE(3)$ transform relating them. The minimization is started using a constant velocity motion model unless the photometric error increases after applying it. If the latter is true, the motion model is disregarded and the pose of the last tracked frame is used. Similar to PTAM the optimization takes place in the tangent space $S\xi(3)$ that minimally parameterizes the rigid body transform by six parameters.

3.5 Map generation

The map generation module represents the world as a dense (for direct) or sparse (for indirect) cloud of points. Figure 10 presents the flowchart of a map generation module. The system proceeds by triangulating the 2D points of interest into 3D landmarks and keeps track of their 3D coordinates and then localizes the camera within what is referred to as a *metric* map. However, as the camera explores large environments, metric maps suffer from the unbounded growth of their size, thereby leading to system failure.

Topological maps were introduced to alleviate this shortcoming, in an effort to minimize the metric information in the map by forfeiting geometric information (scale, distance and direction) in favor for connectivity information. In the context of Visual SLAM, a topological map is an undirected graph of nodes that typically represents keyframes linked together by edges, when shared data associations between the nodes exists.

While topological maps scale well with large scenes, in order to maintain camera pose estimates, metric information is also required; the conversion from a topological to a metric map is not always a trivial task and therefore recent Visual SLAM systems such as (Engel et al, 2014; Lim et al, 2014, 2011; Mur-Artal et al, 2015) employ hybrid maps that are locally metric and globally topological. The implementation of a hybrid map representation permits the system to (1) reason about the world on a high level, which allows for efficient solutions to loop closures and failure recovery using topological information, and (2) to increase efficiency of the metric pose estimate by limiting the scope of the map to a local region surrounding the camera (Fernández-Moral et al, 2015). A hybrid map allows for local optimization of the metric map while maintaining scalability of the optimization over the global topological map (Konolige, 2010).

In a metric map the map making process handles the initialization of new landmarks into the map as well as outlier detection and handling. The 3D structure of the observed scene is sought from a known transformation between two frames, along with the corresponding data associations. Due to noise in data association and pose estimates of the tracked images, projecting rays from two associated features will most probably not intersect in 3D space. Triangulation by optimization as (shown in Fig. 11) aims to estimate a landmark pose corresponding to the associated features, by minimizing its re-projection error e_1 and e_2 onto both frames. To gain resilience against outliers and to obtain better accuracy, some systems employ a similar optimization over features associated across more than two views.

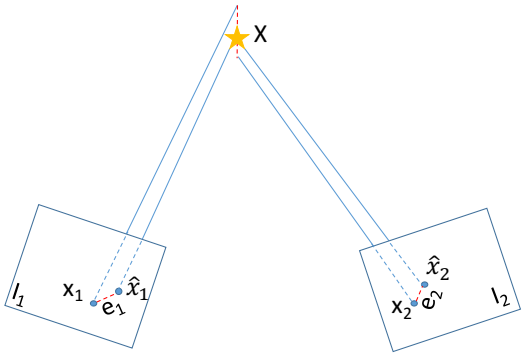


Fig. 11 Landmark triangulation by optimization.

As shown in Fig. 12, filter-based landmark estimation techniques recover the position of a landmark by populating

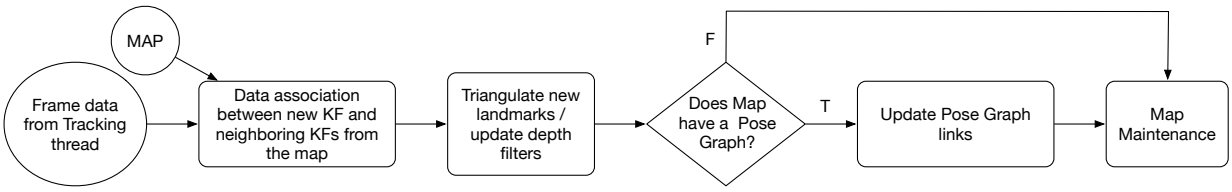


Fig. 10 Generic map generation pipeline.

a particle filter with a uniform distribution (D_1) of landmark position estimates, which are then updated as the landmark is observed across multiple views. This continues until the filter converges from a uniform distribution to a Gaussian with a small variance (D_3). In this type of landmark estimation, outliers are easily flagged as landmarks whose distribution remain approximately uniform after significant observations. Filter-based methods result in a time delay before an observed landmark can be used for pose tracking, in contrast to triangulation by optimization methods that can be used as soon as the landmark is triangulated from two views.

A major limitation in all these methods is that they require a baseline between the images observing the feature, and hence are all prone to failure when the camera's motion is constrained to pure rotations. To counter such a mode of failure, DT SLAM introduced into the map 2D landmarks that can be used for rotation estimation before they are triangulated into 3D landmarks.

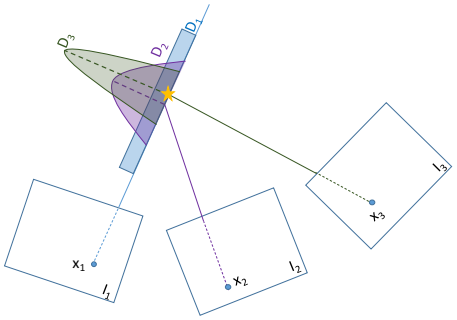


Fig. 12 Landmark estimation using filter based methods.

Table 6 summarizes map generation methods employed by different Visual SLAM systems which can be divided into two main categories: triangulation by optimization (PTAM and ORB SLAM) and filter based landmark estimation (SVO, LSD SLAM and DPPTAM).

PTAM. When a new keyframe is added in PTAM, all bundle adjustment operations are halted, and the new keyframe inherits the pose from the coarse tracking stage. The potentially visible set of landmarks, estimated by the tracker, is then re-projected onto the new keyframe, and feature matches are established. Correctly matched landmarks are marked as seen again; this is done to keep track of the quality of the landmarks and to allow for the map refinement step to remove corrupt data.

New landmarks are generated by establishing and triangulating feature matches between the newly added keyframe and its nearest keyframe (in terms of position) from the map. Already existant landmarks from the map are projected onto both keyframes and feature matches from the current keyframe are searched for along their corresponding epipolar line in the other keyframe at regions that do not contain projected landmarks. The average depth of the projected landmarks is used to constrain the epipolar search from a line to a segment; this limits the computation cost of the search and avoids adding landmarks in regions where nearby landmarks exist. However, this also limits the newly created landmarks to be within the epipolar segment, and hence very large variations in the scene's depth may lead to the negligence of possible landmarks.

SVO. The map generation thread in SVO runs parallel to the tracking thread and is responsible for creating and updating the map. SVO parametrizes 3D landmarks using an inverse depth parameterization model (Civera et al, 2008). Upon insertion of a new keyframe, features possessing the highest Shi-Tomasi scores are chosen to initialize a number of depth filters. These features are labeled as seeds and are initialized to be along a line propagating from the camera center to the 2D location of the seed in the originating keyframe. The only parameter that remains to be solved for is then the depth of the landmark, which is initialized to the mean of the scene's depth, as observed from the keyframe of origin.

During the times when no new keyframe is being processed, the map management thread monitors and updates map seeds by subsequent observations in newly acquired frames. The seed is searched for in new frames along an epipolar search line, which is limited by the uncertainty of the seed and the mean depth distribution observed in the current frame. As the filter converges, its uncertainty decreases and the epipolar search range decreases. If seeds fail to match frequently, if they diverge to infinity, or if a long time has passed since their initialization, they are considered bad seeds and removed from the map.

The filter converges when the distribution of the depth estimate of a seed transitions from the initially assumed uniform distribution into a Gaussian one. The seed in then

Table 6 Map generation. Abbreviations: 2 view triangulation (2.v.t.), particle filter with inverse depth parametrization (p.f.), 2D landmarks triangulated to 3D landmarks (2D.l.t.), depth map propagation from previous frame (p.f.p.f.), depth map refined through small baseline observations (s.b.o.), multiple hypotheses photometric error minimization (m.h.p.m.)

	PTAM	SVO	DT SLAM	LSD SLAM	ORB SLAM	DPPTAM
Map generation	2.v.t.	p.f.	2D.l.t.	p.f.p.f or s.b.o.	2.v.t.	m.h.p.m
Map type	metric	metric	metric	hybrid	hybrid	metric

added into the map, with the mean of the Gaussian distribution as its depth. This process however limits SVO to operate in environments of relatively uniform depth distributions. Since the initialization of landmarks in SVO relies on many observations in order for the features to be triangulated, the map contains few if any outliers and hence no outlier deletion method is required. However, this comes at the expense of a delayed time before the features are initialized as landmarks and added to the map.

DT SLAM. DT SLAM aims to add keyframes when enough visual change has occurred; the three criteria for keyframe addition are (1) for the frame to contain a sufficient number of new 2D features that can be created from areas not covered by the map, or (2) a minimum number of 2D features can be triangulated into 3D landmarks, or (3) a given number of already existing 3D landmarks have been observed from a significantly different angle. The map contains both 2D features and 3D landmarks, where the triangulation of 2D features into 3D landmarks is done through two view triangulation and is deferred until enough parallax between keyframes is observed–hence the name of the algorithm.

LSD SLAM. LSD SLAM's map generation module functions can be divided into two main categories, depending on whether the current frame is a keyframe or not; if it is, depth map creation takes place by keyframe accommodation; if not, depth map refinement is done on regular frames. To maintain tracking quality, LSD SLAM requires frequent addition of keyframes into the map as well as relatively high frame rate cameras.

If a frame is labeled a keyframe, the estimated depth map from the previous keyframe is projected onto it and serves as its initial depth map. Spatial regularization then takes place by replacing each projected depth value by the average of its surrounding values and the variance is chosen as the minimal variance value of the neighboring measurements.

In LSD SLAM, outliers are detected by monitoring the probability of the projected depth hypothesis at each pixel to be an outlier or not. To make the outliers detection step possible, LSD SLAM keeps records of all successfully matched pixels during the tracking thread, and accordingly increases or decreases the probability of it being an outlier.

The $Sim(3)$ of a newly added keyframe is then estimated and refined in a direct, scale-drift aware image alignment scheme, which is similar to the one done in the tracking thread, but with respect to other keyframes in the map and over the $7d.o.f.$ $Sim(3)$ transform.

Due to the non-convexity of the direct image alignment method on $Sim(3)$, an accurate initialization to the minimization procedure is required; for such purpose, ESM (Efficient Second Order minimization) (Benhimane and Malis, 2007) and a coarse to fine pyramidal scheme with very low resolutions proved to increase the convergence radius of the task.

If the map generation module deems the current frame as not being a keyframe, depth map refinement takes place by establishing stereo matches for each pixel in a suitable reference frame. The reference frame for each pixel is determined by the oldest frame the pixel was observed in, where the disparity search range and the observation angle do not exceed a certain threshold. A 1-D search along the epipolar line for each pixel is performed with an SSD metric.

To minimize computational cost and reduce the effect of outliers on the map, not all established stereo matches are used to update the depth map; instead, a subset of pixels is selected for which the accuracy of a disparity search is sufficiently large. The accuracy is determined by three criteria: the photometric disparity error, the geometric disparity error, and the pixel to inverse depth ratio. Further details regarding these criteria are outside the scope of this work, the interested reader is referred to Engel et al (2013). Finally, depth map regularization and outlier handling, similar to the keyframe processing step, take place.

ORB SLAM. ORB SLAM's local mapping thread is responsible for keyframe insertion, map point triangulation, map point culling, keyframe culling and local bundle adjustment. The keyframe insertion step is responsible for updating the co-visibility and essential graphs with the appropriate edges, as well as computing the bag of words representing the newly added keyframe in the map. The co-visibility graph is a pose graph that represents all keyframes in the system by nodes, in contrast to the essential graph that allow every node to have two or less edges, by keeping the strongest two edges only for every node. The map point creation module spawns new landmarks by triangulating ORB

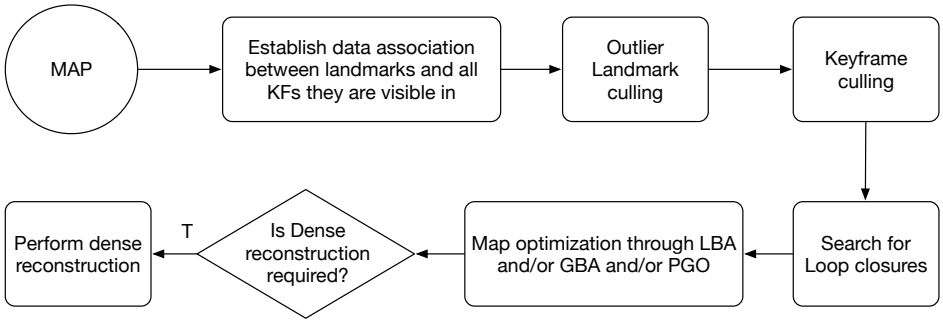


Fig. 13 Generic map maintenance pipeline.

features that appear in two or more views from connected keyframes in the co-visibility graph. Triangulated landmarks are tested for positive depth, re-projection error, and scale consistency in all keyframes they are observed in, in order to accommodate them into the map.

DPPTAM. Landmark triangulation in DPPTAM takes place over several overlapping observations of the scene using inverse depth parametrization; the map maker aims to minimize the photometric error between a high gradient pixel patch in the last added keyframe and the corresponding patch of pixels, found by projecting the feature from the keyframe onto the current frame. The minimization is repeated ten times for all high-gradient pixels when the frame exhibits enough translation; the threshold for translation is increased from an iteration to another to ensure enough baseline between the frames. The end result is ten hypotheses for the depth of each high-gradient pixel. To deduce the final depth estimate from the hypotheses, three consecutive tests are performed, including gradient direction test, temporal consistency, and spatial consistency.

3.6 Map maintenance

Map maintenance takes care of optimizing the map through either bundle adjustment or pose graph optimization (Kummerle et al, 2011). Figure 13 presents the steps required for map maintenance of a generic Visual SLAM. During a map exploration phase, new 3D landmarks are triangulated based on the camera pose estimates. After some time, system drift manifests itself in wrong camera pose measurements, due to accumulated errors in previous camera poses that were used to expand the map. Figure 14 describes the map maintenance effect where the scene's map is refined through outlier removal and error minimizations, to yield a more accurate scene representation.

Bundle adjustment (BA) is inherited from SfM and consists of a nonlinear optimization process for refining a visual reconstruction, to jointly produce an optimal structure and coherent camera pose estimates. Bundle adjustment is computationally involved and intractable if performed on all frames and all poses. The breakthrough that enabled its application in PTAM is the notion of keyframes, where only select frames labeled as keyframes are used in the map creation and passed to the bundle adjustment process in contrast to SfM methods that uses all available frames. Different algorithms apply different criteria for keyframe labeling, as well as different strategies for BA, some use jointly a local (over a local number of keyframes) LBA and global (over the entire map) GBA, while others argue that a local BA only is sufficient to maintain a good quality map. To reduce the computational expenses of bundle adjustment, Strasdat et al (2011) proposed to represent the visual SLAM map by both a Euclidean map for LBA, along with a topological map for pose graph optimization that explicitly distributes the accumulated drift along the entire map.

Pose Graph Optimization (PGO) returns inferior results to those produced by GBA. The reasons is that while PGO optimizes only for the keyframe poses–and accordingly adjusts the 3D structure of landmarks–GBA jointly optimizes for both keyframe poses and 3D structure. The stated advantage of GBA comes at the cost of computational time, with PGO exhibiting significant speed up compared to GBA. However, pose graph optimization requires efficient loop closure detection and may not yield an optimal result as the errors are distributed along the entire map, leading to locally induced inaccuracies in regions that were not originally wrong.

Map maintenance is also responsible for detecting and removing outliers in the map due to noisy and faulty matched features. While the underlying assumption of most Visual SLAM algorithms is that the environment is static, some algorithms such as RD SLAM exploits map maintenance methods to accommodate slowly varying scenes (lighting and structural changes).

PTAM. The map making thread in PTAM runs parallel to the tracking thread and does not operate on a frame by frame basis; instead, it only processes keyframes. When the map making thread is not processing new keyframes, it performs various optimizations and maintenance to the map,

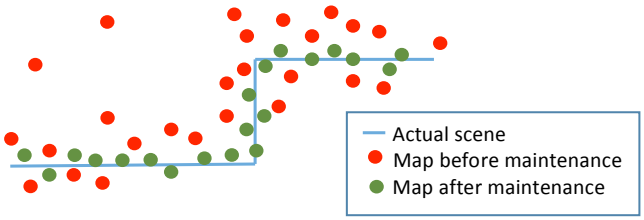


Fig. 14 Map maintenance effects on the map.

such as an LBA for local map convergence and a global adjustment for the map's global convergence. The computational cost in PTAM scales with the map and becomes intractable as the number of keyframes get large; for this reason PTAM is designed to operate in small workspaces. Finally, the optimization thread applies data refinement by first searching and updating landmark observations in all keyframes, and then by removing all landmarks that failed many times to successfully match features.

SVO. For runtime efficiency reasons, SVO's map management maintains only a fixed number of keyframes in the map and removes distant ones when new keyframes are added. This is performed so that the algorithm maintains real-time performance after prolonged periods of operation over large distances.

DT SLAM. Aside the map generation module, DT SLAM employ a third thread that continuously optimize the entire map in the background through a sparse GBA.

LSD SLAM. LSD SLAM runs a third parallel thread that continuously optimizes the map in the background by a generic implementation of a pose graph optimization using the g2o-framework (Kummerle et al, 2011). This however leads to an inferior accuracy when compared to other methods for reasons stated earlier.

ORB SLAM. To maintain a good quality map and counter the effect of frequently adding features from keyframes, ORB SLAM employs rigorous landmark culling to ensure few outliers in the map. A landmark must be correctly matched to twenty five percent of the frames in which it is predicted to be visible. It also must be visible from at least three keyframes after more than one keyframe has been accommodated into the map since it was spawned. Otherwise, the landmark is removed. To maintain lifelong operation and to counter the side effects (computational complexity, outliers, redundancy) of the presence of high number of keyframes in the map, a rigorous keyframe culling procedure takes place as well. Keyframes that have ninety percent of their associated landmarks observed in three other keyframes are deemed redundant and removed. The local mapping thread also performs a local bundle adjustment over all keyframes connected to the latest accommodated keyframe in the co-visibility graph and all other keyframes that observe any landmark present in the current keyframe.

DPPTAM. DPPTAM does not employ optimization tasks to its map as previous systems do, however it produces real-time dense maps by employing a dense mapping thread that exploits planar properties of man-made indoor environments. Keyframes are first segmented into a set of 2D superpixels and all 3D landmarks from the map are projected onto the keyframe and assigned to different superpixels according to the distance of their projections to the appropriate superpixel in the keyframe. 3D points belonging to contours of the superpixels are used to fit 3D planes to each superpixel before three tests determine if the superpixel's plane is to be added into the map. The tests include (1) the normalized residual test, (2) the degenerate case detection, and (3) temporal consistency. Finally a full dense map is reconstructed by estimating depth at every pixel using depth priors of the 3D planes associated with the superpixels.

Table 7 summarizes the map maintenance procedure adopted by different Visual SLAM system.

3.7 Failure recovery

Whether due to wrong user movement (abrupt change in the camera pose and motion blur), or camera observing a featureless region or failure to match enough features, or for any other reason, Visual SLAM methods can eventually fail. A key module essential for the usability of any visual SLAM system is its ability to correctly recover from such failures.

PTAM. Upon detecting failure, PTAM's tracker initiates a recovery procedure, where the SBI (small blurry image) of each incoming frame is compared to the database of SBIs for all keyframes. If the intensity difference between the incoming frame and its nearest looking keyframe is below a certain threshold, the current frame's pose is assumed to be equivalent to that of the corresponding keyframe. ESM tracking takes place (similar to that of the tracking thread) to estimate the rotational change between the keyframe and the current frame. If converged, a PVS of landmarks from the map is projected onto the estimated pose and the tracker attempts to match the landmarks to the features in the frame. If enough features are correctly matched, the tracker resumes normally, otherwise a new frame is acquired and the tracker remains lost. For successful re-localization, this method requires the lost camera's pose to be near the recorded keyframe's pose, and otherwise would fail when there is a large displacement between the two.

SVO. When tracking quality is deemed insufficient, SVO initiates its failure recovery method. The first procedure in the recovery process is to apply image alignment between the incoming frame and the closest keyframe to the last known correctly tracked frame. If more than thirty features are correctly matched during the image alignment step,

Table 7 Map maintenance. Abbreviations used: Local Bundle Adjustment (LBA), Global Bundle Adjustment (GBA), Pose Graph Optimization (PGO),

	PTAM	SVO	DT SLAM	LSD SLAM	ORB SLAM	DPPTAM
Optimization	LBA & GBA	LBA	LBA & GBA	PGO	PGO& LBA	Dense mapping
Scene type	static & small	uniform depth	static & small	static &small or large	static & small or large	static & indoor planar

then the re-localizer considers itself converged and continues tracking regularly; otherwise, it attempts to re-localize using new incoming frames. Such a re-localizer is sensitive to any change in the lighting conditions of the scene, and the lost frame location should be close enough to the queried keyframe for successful re-localization to take place.

LSD SLAM. LSD SLAM's recovery procedure first chooses randomly a keyframe from the map that has more than two neighboring keyframes connected to it in the pose graph. It then attempts to align the currently lost frame to it. If the outlier-to-inlier ratio is large, the keyframe is discarded and replaced by another keyframe at random; otherwise, all neighboring keyframes connected to it in the pose graph are then tested. If the number of neighbors with a large inlier-to-outlier ratio is larger than the number of neighbors with a large outlier-to-inlier ratio, or if there are more than five neighbors with a large inlier-to-outlier ratio, the neighboring keyframe with the largest ratio is set as the active keyframe and regular tracking is accordingly resumed.

ORB SLAM. Triggered by tracking failure, ORB SLAM invokes its global place recognition module. Upon running, the re-localizer transforms the current frame into a bag of words and queries the database of keyframes for all possible keyframes that might be used to re-localize from. The place recognition module implemented in ORB SLAM, used for both loop detection and failure recovery, relies on bags of words as frames observing the same scene share a big number of common visual vocabulary. In contrast to other bag of words methods that return the best queried hypothesis from the database of keyframes, the place recognition module of ORB SLAM returns all possible hypotheses that have a probability of being a match larger than seventy five percent of the best match. The combined added value of the ORB features, along with the bag of words implementation of the place recognition module, manifests itself in a real-time, high recall, and relatively high tolerance to viewpoint changes during re-localization and loop detection. All hypotheses are then tested through a RANSAC implementation of the PnP algorithm (Lepetit et al, 2009) that determines the camera pose from a set of 3D to 2D correspondences. The found camera pose with the most inliers is then used to establish more matches to features associated with the candidate keyframe, before an optimization over the camera's pose using the established matches takes place.

Table 8 summarizes the failure recovery mechanisms used by different Visual SLAM system.

3.8 Loop closure

Since Visual SLAM is an optimization problem, it is prone to drifts in camera pose estimates. Returning to a certain pose after an exploration phase may not yield the same camera pose measurement as it was at the start of the run (See Fig. 15). Such camera pose drift can also manifest itself in a map scale drift that will eventually lead the system to erroneous measurements and fatal failure. To address this issue, some algorithms detect loop closures in an on-line Visual SLAM session and optimize the loops track, in an effort to correct the drift and the error in the camera pose and in all relevant map data that were created during the loop. The loop closure thread attempts to establish loops upon the insertion of a new keyframe in order to correct and minimize any accumulated drift by the system over time.

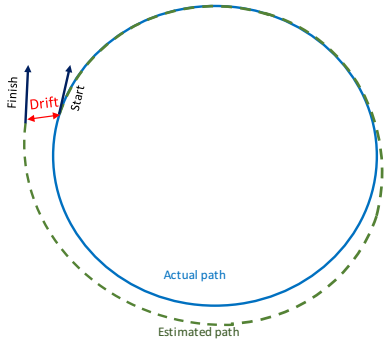


Fig. 15 Drift suffered by the Visual SLAM pose estimate after returning to its starting point.

LSD SLAM. Whenever a keyframe is processed by LSD SLAM, loop closures are searched for within its ten nearest keyframes as well as through the appearance based model of FABMAP (Glover et al, 2012) to establish both ends of a loop. Once a loop edge is detected, a pose graph optimization minimizes the similarity error established at the loops edge by distributing the error over the loops keyframes poses.

ORB SLAM. Loop detection in ORB SLAM takes place via its global place recognition module that returns all hypotheses of keyframes from the database that might correspond to the opposing loop end. To ensure enough distance change has taken place, they compute what they refer to as the *similarity transform* between all connected keyframes to

Table 8 Failure recovery. Abbreviations used: photometric error minimization of SBIs (p.e.m.), image alignment with last correctly tracked keyframe (i.a.l.), image alignment with random keyframe (i.a.r.), bag of words place recognition (b.w.)

	PTAM	SVO	DT SLAM	LSD SLAM	ORB SLAM	DPPTAM
Failure recovery	p.e.m.	i.a.l.	none	i.a.r.	b.w.	i.a.l

the current keyframe in the thresholded co-visibility graph. It the similarity score is less than a threshold, the hypothesis of a loop is removed. If enough inliers support the refined similarity transform, the queried keyframe is considered to be the other end of the loop and loop fusion takes place.

The loop fusion first merges duplicate map points in both keyframes and inserts a new edge in the co-visibility graph that closes the loop by correcting the $Sim(3)$ pose of the current keyframe using the similarity transform. Using the corrected pose, all landmarks associated with the queried keyframe and its neighbors are projected to and searched for in all keyframes associated with the current keyframe in the co-visibility graph. The initial set of inliers, as well as the found matches are used to update the co-visibility and Essential graphs, establishing many edges between the two ends of the loop. Finally, a pose graph optimization over the essential graph takes place similar to that of LSD SLAM, which minimizes and distributes the loop closing error along the loop nodes.

Table 9 summarizes the Loop closure mechanisms used by different Visual SLAM system.

Table 9 Loop closure. Abbreviations used: Bag of Words place recognition (B.W.p.r), sim(3) optimization (s.o.)

	PTAM	SVO	DT SLAM	LSD SLAM	ORB SLAM	DPPTAM
Loop closure	none	none	none	FabMap +s.o.	B.W.p.r. +s.o.	none

4 Closed source systems

We have discussed so far methods presented in open-source Visual SLAM systems; however, plenty of closed source methods exist in the literature. This section aims to provide a quick overview of these systems, which include many interesting ideas for the reader. Table 10 lists in chronological order each of these systems. To avoid repetition, we will not outline the complete details of each system; rather, we will focus on what we feel has additive value to the reader from the information provided in Section 3. In 2006, Mouragnon et al (2006) were the first to introduce in the concept of keyframes in Visual SLAM and employed a local Bundle Adjustment in real-time over a subset of keyframes in the map. To ensure sufficient baseline, the system is initialized by automatic insertion of three keyframes. However the system does not use the 3 views to perform the initialization, instead it solves for the initialization using the 5-point algorithm of Nistér (2004) between the 1^{st} and 3^{rd} keyframes only therefore, it is still susceptible to planar scenes.

Silveira et al (2008) proposed a real-time direct solution by assuming relatively large patches of pixels surrounding regions of high intensity gradients as planar, and performing image alignment by minimizing the photometric error of these patches across incoming frames, in a single optimization step that incorporates Cheirality, geometric and photometric constraints. To gain resilience against lighting changes and outliers, the system employs a photogeometric generative model and monitors the errors in the minimization process to flag outliers.

In 2010, =Strasdat et al (2010a) introduced similarity transforms into Visual SLAM, allowing for scale drift estimation and correction once the system detects loop closure. Feature tracking is performed by a mixture of top-bottom and bottom-up approach, using a dense variational optical flow and a search over a window surrounding the projected landmarks. Landmarks are triangulated by updating information filters and loop detection is performed using a bag of words discretization of SURF features (Bay et al, 2008). The loop is finally closed by applying a pose graph optimization over the similarity transforms relating the keyframes.

Also in 2010, Newcombe and Davison (2010) suggested a hybrid Visual SLAM system that relied on feature-based SLAM (PTAM) to fit a dense surface estimate of the environment that is refined using direct methods. A surface-based model is then computed and polygonized to best fit the triangulated landmarks from the feature-based front end. A parallel process chooses a batch of frames that have a potentially overlapping surface visibility in order to estimate a dense refinement over the base mesh using a GPU accelerated implementation of variational optical flow.

In an update to this work, Newcombe released in 2011 Dense Tracking and Mapping (DTAM) (Newcombe et al, 2011) that removed the need for PTAM as a front-end to the system and generalized the dense reconstruction to fully solve the Visual SLAM pipeline, by performing on-line a dense reconstruction, given camera pose estimates that are found through whole image alignment.

Similar to the work of Newcombe(Newcombe et al, 2011), Pretto et al (2011) modeled the environment as a 3D

Table 10 Closed source Visual SLAM systems

Year	Paper	Additive information beyond open-source techniques
2006	Real Time Localization and 3D Reconstruction Mouragnon et al (2006)	– Introduced keyframes and LBA for real-time Visual SLAM – Utilize 3 keyframes for initialization
2008	An efficient direct approach to Visual SLAM Silveira et al (2008)	– Real-time direct image alignment – Employed a photogeometric generative model – Outlier handling using phometric and geometric errors
2010	Scale-Drift Aware Large Scale Monocular SLAM Strasdat et al (2010a)	– Introduced similarity transforms to visual SLAM – Data association using dense variational optical flow and search windows. – Use information filters to triangulate landmarks – Loop detection and closure techniques using bags of words discretization of SURF features and a pose graph optimization
2010	Live dense reconstruction with a single moving camera Newcombe and Davison (2010)	– Fit a base mesh to a sparse set of landmarks triangulated using PTAM – GPU parallelization of variational optical flow to refine the base mesh
2011	Omnidirectional dense large scale mapping and navigation based on meaningful triangulation Pretto et al (2011)	– Modeled the environment as a 3D piecewise smooth surface. – Meshing using delaunay triangulation – incorporated edgelets into the mesh – interpolated a dense reconstruction using the mesh vertices.
2011	Dense Tracking and Mapping (DTAM) Newcombe et al (2011)	– Removed the need for PTAM as a front-end from Pretto et al (2011)
2011	Continuous Localization and Mapping in a Dynamic world Pirker et al (2011)	– Handle short and long term environmental changes – Use HOC descriptors to gain rotational change immunity – re-localization through FABMAP – Loop closure through pose graph optimization
2013	Robust Monocular SLAM in Dynamic Environments Tan et al (2013)	– Handle occlusions and slowly varying scenes and illumination changes – Prior-based adaptive RANSAC sampling of features – Employed a GPU accelerated SIFT extraction and KD-Tree feature matching – Landmark and keyframe update and culling using color histograms.
2013	Handling pure camera rotation in keyframe based SLAM Pirchheim et al (2013)	– panoramic submaps (2D landmarks) to handle pure camera rotation – phonySIFT feature extraction and matching using hierarchical k-means.
2014	Real-Time 6-DOF Monocular Visual SLAM in a large scale environment Lim et al (2014)	– Hybrid topological and metric map – Tracking, mapping and loop closure all using the same binary descriptor
2015	Robust Large Scale monocular Visual SLAM Bourmaud and Megret (2015)	– Off-line method – Divide the map into submaps stored in a graph – Suggested a loop closure outlier detection mechanism in submaps – Employed a loopy belief propagation algorithm (LS-RSA)

piecewise smooth surface and used a sparse feature based front-end as a base for a Delaunay triangulation to fit a mesh that is used to interpolate a dense reconstruction of the environment.

Pirker et al (2011) released CD SLAM in 2011 with the objectives to handle short- and long-term environmental changes and to handle mixed indoor/outdoor environments. To limit the map size and gain robustness against significant rotational changes, CD SLAM suggests the use of a modified Histogram of Oriented Cameras descriptor (HOC) (Pirker, 2010) with a GPU accelerated descriptor update and a probabilistic weighting scheme to handle outliers. Furthermore, it suggests the use of large-scale nested loop closures with scale drift correction and provide a geometric adaptation to update the feature descriptors after loop closure. Keyframes are organized in an undirected, un-

weighted pose graph. Re-localization is performed using a non-linear least squared minimization initialized, with the pose of the best matching candidate keyframe from the map found through FABMAP; whereas loop closure takes place using pose graph optimization.

In 2013, RD SLAM (Tan et al, 2013) was released with the aim to handle occlusions and slowly varying, dynamic scenes. RD SLAM employs a heavily parallelized GPU accelerated SIFT and stores them in a KD-Tree (Bentley, 1975) that further accelerates feature matching based on the nearest neighbor of the queried feature in the tree. To cope with dynamic objects (moving) and slowly varying scenes, RD SLAM suggests a prior-based adaptive RANSAC scheme, that samples, based on the outlier ratio of features in previous frames, the features in the current frame from which to estimate the camera pose along with a landmark and keyframe culling mechanism, using histograms of colors to detect and update changed image locations, while sparing temporarily occluded landmarks.

Pirchheim et al (2013) dealt with the problem of pure rotations in the camera motion by building local panorama maps, whenever the system explores a new scene with pure rotational motion. The system extracts phonySIFT descriptors as described in Wagner et al (2010)and establish feature correspondences using an accelerated matching method through hierarchical k-means. When insufficient 3D landmarks are observed during pose estimation, the system transitions into a rotation-only estimate mode and starts building a panorama map until the camera observes part of the finite map.

In the work of Lim et al (2014) the sought after objective is to handle tracking, mapping, and loop closure, all using the same binary feature, through a hybrid map representation (topological and metric). Whenever a loop is detected, the map is converted to its metric form where a local Bundle Adjustment take place before returning the map back to its topological form.

In 2015, Bourmaud and Megret (2015) released an offline Visual SLAM system (requiring around two hours and a half for a dataset of 10000 images). The system employs a divide- and-conquer strategy, by segmenting the map into submaps. A similarity transform is estimated between each submap and its ten nearest neighbors. A global similarity transform, relating every submap to a single global reference frame, is computed by a pose graph optimization, where the reference frames are stored in a graph of submaps. The above procedure is susceptible to outliers in the loop detection module and hence the need for an efficient outlier handling mechanism. For such purpose, and to prevent outliers, temporally consecutive similarity measurements are always considered as inliers. The outlier rejection module proceeds then by integrating the similarities over the shortest loop it can find, and monitors the closure error to accept the loop. To cope with a very large number of submaps, a loopy belief propagation algorithm cuts the main graph into subgraphs, before a non-linear optimization takes place.

5 Conclusions

During the course of this work, we have outlined the building blocks of a generic Visual SLAM system; including data type, initialization, data association, pose estimation, map generation, map maintenance, failure recovery and loop closure. We also discussed the details of the latest open-source state of the art systems in Visual SLAM including PTAM, SVO, DT SLAM, LSD SLAM, ORB SLAM and DPPTAM. Finally, we compiled and summarized what added information closed-source non-filter-based monocular Visual SLAM systems have to offer.

Although extensive research has been dedicated to this field, it is our opinion that each of the building blocks discussed above could benefit from improvements. Robust data association against illumination changes, dynamic scenes and occluded environments; an initialization method that can operate without an initial scene assumption nor a large number of processed frames; an accurate camera pose estimate that is not affected by sudden movements, blur, noise, large depth variations nor moving objects; a map making module capable of generating an efficient dense scene representation in regions of little texture, a map maintenance method that improves the map with resilience against dynamic, changing small and large scale environments; a failure recovery procedure capable of reviving the system from significantly large changes in camera view points, are all desired properties that unfortunately most state of the art systems lack and remain challenging topics in the field.

We are currently working on creating a set of experiments, which cater to the requirements of each of the above open-source systems–for initialization, camera frame rate, and depth homogeneity. Upon completion of these experiments, we will benchmark all the open-source Visual SLAM systems to better identify the advantages and disadvantages of each system module. The long-term goal would then be to leverage on all of the advantages to create a superior non-filter-based Visual SLAM.

Acknowledgements This work was funded by the ENPI (European Neighborhood Partnership Instrument) grant # I-A/1.2/113 as well as the Lebanese National Research Council (LNCSR).

References

Alahi A, Ortiz R, Vandergheynst P (2012) Freak: Fast retina keypoint. In: Computer Vision and Pattern Recognition

(CVPR), 2012 IEEE Conference on, pp 510–517, DOI 10.1109/CVPR.2012.6247715

Baker S, Matthews I (2004) Lucas-Kanade 20 Years On: A Unifying Framework. International Journal of Computer Vision 56(3):221–255

Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (surf). Comput Vis Image Underst 110(3):346–359, DOI 10.1016/j.cviu.2007.09.014, URL http://dx.doi.org/10.1016/j.cviu.2007.09.014

Beaudet PR (1978) Rotationally invariant image operators. In: International Conference on Pattern Recognition

Benhimane S, Malis E (2007) Homography-based 2D Visual Tracking and Servoing. International Journal of Robotics Research 26(7):661–676

Bentley JL (1975) Multidimensional Binary Search Trees Used for Associative Searching. Communications ACM 18(9):509–517

Bourmaud G, Megret R (2015) Robust large scale monocular visual slam. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, pp 1638–1647, DOI 10.1109/CVPR.2015.7298772

Calonder M, Lepetit V, Ozuysal M, Trzcinski T, Strecha C, Fua P (2012) Brief: Computing a local binary descriptor very fast. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(7):1281–1298, DOI 10.1109/TPAMI.2011.222

Castle RO, Klein G, Murray DW (2008) Video-rate Localization in Multiple Maps for Wearable Augmented Reality. In: Proc 12th {IEEE} Int Symp on Wearable Computers, pp 15–22

Celik K, Somani AK (2013) Monocular vision slam for indoor aerial vehicles. Journal of Electrical and Computer Engineering 2013:15, URL http://dx.doi.org/10.1155/2013/374165%]374165

Civera J, Davison AJ, Montiel JM (2007) Dimensionless monocular slam. In: Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis, Part II, Springer-Verlag, Berlin, Heidelberg, IbPRIA '07, pp 412–419, DOI 10.1007/978-3-540-72849-8_52, URL http://dx.doi.org/10.1007/978-3-540-72849-8_52

Civera J, Davison A, Montiel J (2008) Inverse Depth Parametrization for Monocular SLAM. IEEE Transactions on Robotics 24(5):932–945

Clemente L, Davison A, Reid I, Neira J, Tardós J (2007) Mapping Large Loops with a Single Hand-Held Camera. Atlanta, GA, USA, URL http://www.roboticsproceedings.org/rss03/p38.html

Concha A, Civera J (2015) Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp 5686–5693, DOI 10.1109/IROS.2015.7354184

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol 1, pp 886–893 vol. 1, DOI 10.1109/CVPR.2005.177

Davison A (2003) Real-time simultaneous localisation and mapping with a single camera. In: Ninth IEEE International Conference on Computer Vision, pp 1403–1410 vol.2

Davison A, Cid YG, Kita N (2004) Real-time 3D SLAM with wide-angle vision. In: Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon

Davison AJ, Reid ID, Molton ND, Stasse O (2007) MonoSLAM: real-time single camera SLAM. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on 29(6):1052–67

Eade E, Drummond T (2006) Scalable monocular slam. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol 1, pp 469–476, DOI 10.1109/CVPR.2006.263

Eade E, Drummond T (2007) Monocular slam as a graph of coalesced observations. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp 1–8, DOI 10.1109/ICCV.2007.4409098

Engel J, Sturm J, Cremers D (2013) Semi-dense Visual Odometry for a Monocular Camera. In: Computer Vision (ICCV), IEEE International Conference on, IEEE, pp 1449–1456

Engel J, Schöps T, Cremers D (2014) LSD-SLAM: Large-Scale Direct Monocular SLAM. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision – ECCV 2014 SE - 54, Lecture Notes in Computer Science, vol 8690, Springer International Publishing, pp 834–849

Engel J, Stuckler J, Cremers D (2015) Large-scale direct slam with stereo cameras. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp 1935–1942, DOI 10.1109/IROS.2015.7353631

Faugeras O, Lustman F (1988) Motion and structure from motion in a piecewise planar environment. International Journal of Pattern Recognition and Artificial Intelligence 02(03):485–508

Fernández-Moral E, Arévalo V, González-Jiménez J (2015) Hybrid Metric-topological Mapping for Large Scale Monocular SLAM . Springer International Publishing, pp 217–232

Forster C, Pizzoli M, Scaramuzza D (2014) SVO : Fast Semi-Direct Monocular Visual Odometry. In: Robotics and Automation (ICRA), IEEE International Conference on

Fuentes-Pacheco J, Ruiz-Ascencio J, Rendón-Mancha JM (2012) Visual simultaneous localization and mapping: a survey. Artificial Intelligence Review 43(1):55–81,

DOI 10.1007/s10462-012-9365-8, URL http://dx.doi.org/10.1007/s10462-012-9365-8

Galvez-López D, Tardos JD (2012) Bags of Binary Words for Fast Place Recognition in Image Sequences. Robotics, IEEE Transactions on 28(5):1188–1197, DOI 10.1109/TRO.2012.2197158

Glover A, Maddern W, Warren M, Reid S, Milford M, Wyeth G (2012) OpenFABMAP: An open source toolbox for appearance-based loop closure detection. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 4730–4735

Grasa O, Bernal E, Casado S, Gil I, Montiel J (2014) Visual slam for handheld monocular endoscope. Medical Imaging, IEEE Transactions on 33(1):135–146, DOI 10.1109/TMI.2013.2282997

Hall BC (2015) Lie Groups, Lie Algebras, and Representations, vol 222, number 102 edn. Springer- Verlag

Harris C, Stephens M (1988) A combined corner and edge detector. In: In Proc. of Fourth Alvey Vision Conference, pp 147–151

Hartley R, Zisserman A (2003) Multiple View Geometry in Computer Vision. Cambridge University Press

Hartmann J, Klussendorff JH, Maehle E (2013) A comparison of feature descriptors for visual SLAM. In: Mobile Robots (ECMR), 2013 European Conference on, pp 56–61, DOI 10.1109/ECMR.2013.6698820

Herrera D, Kannala J, Pulli K, Heikkila J (2014) DT-SLAM: Deferred Triangulation for Robust SLAM. In: 3D Vision, 2nd International Conference on, IEEE, vol 1, pp 609–616

Hietanen A, Lankinen J, Kämäräinen JK, Buch AG, Krüger N (2016) A comparison of feature detectors and descriptors for object class matching. Neurocomputing

Hochdorfer S, Schlegel C (2009) Towards a robust visual slam approach: Addressing the challenge of life-long operation. In: Advanced Robotics, 2009. ICAR 2009. International Conference on, pp 1–6

Holmes SA, Klein G, Murray DW (2008) A square root unscented kalman filter for visual monoslam. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(7):1251–1263, DOI http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.189

Horn BKP (1987) Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A 4(4):629–642

Jeong W, Lee KM (2005) Cv-slam: a new ceiling vision-based slam technique. In: Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on, pp 3195–3200, DOI 10.1109/IROS.2005.1545443

Jérôme Martin JLC (1995) Experimental Comparison of Correlation Techniques. In: IAS-4, International Conference on Intelligent Autonomous Systems

Klein G, Murray D (2007) Parallel Tracking and Mapping for Small AR Workspaces. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality pp 1–10

Klein G, Murray D (2008) Improving the Agility of Keyframe-Based {SLAM}. In: Proc. 10th European Conference on Computer Vision (ECCV), Marseille, pp 802–815

Kneip L, Siegwart R, Pollefeys M (2012) Finding the Exact Rotation between Two Images Independently of the Translation, Springer Berlin Heidelberg, Berlin, Heidelberg, chap Finding th, pp 696–709. DOI 10.1007/978-3-642-33783-3{_}50, URL http://dx.doi.org/10.1007/978-3-642-33783-3{_}50

Konolige K (2010) Sparse sparse bundle adjustment. In: Proceedings of the British Machine Vision Conference, BMVA Press, pp 102.1–102.11, doi:10.5244/C.24.102

Kummerle R, Grisetti G, Strasdat H, Konolige K, Burgard W (2011) G2o: A general framework for graph optimization. In: Robotics and Automation (ICRA), IEEE International Conference on, IEEE, pp 3607–3613

Kwon J, Lee KM (2010) Monocular slam with locally planar landmarks via geometric rao-blackwellized particle filtering on lie groups. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp 1522–1529, DOI 10.1109/CVPR.2010.5539789

Lee SH (2014) Real-time camera tracking using a particle filter combined with unscented kalman filters. Journal of Electronic Imaging 23(1):013,029, DOI 10.1117/1.JEI.23.1.013029, URL http://dx.doi.org/10.1117/1.JEI.23.1.013029

Lemaire T, Lacroix S (2007) Monocular-vision based slam using line segments. In: Robotics and Automation, 2007 IEEE International Conference on, pp 2791–2796, DOI 10.1109/ROBOT.2007.363894

Lepetit V, Moreno-Noguer F, Fua P (2009) EPnP: An Accurate O(n) Solution to the PnP Problem. International Journal of Computer Vision 81(2):155–166

Leutenegger S, Chli M, Siegwart RY (2011) Brisk: Binary robust invariant scalable keypoints. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp 2548–2555, DOI 10.1109/ICCV.2011.6126542

Lim H, Lim J, Kim HJ (2014) Real-time 6-DOF monocular visual SLAM in a large-scale environment. In: Robotics and Automation (ICRA), IEEE International Conference on, pp 1532–1539

Lim J, Frahm JM, Pollefeys M (2011) Online environment mapping. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp 3489–3496, DOI 10.1109/CVPR.2011.5995511

Lindeberg T (1998) Feature detection with automatic scale selection. Int J Comput Vision 30(2):79–116, DOI 10.1023/A:1008045108935, URL http://dx.doi.org/

10.1023/A:1008045108935

Lowe D (1999) Object recognition from local scale-invariant features. In: International Conference on Computer Vision (ICCV), the seventh IEEE, IEEE, vol 2, pp 1150–1157 vol.2

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60(2):91–110, DOI 10.1023/B:VISI.0000029664.99615.94, URL http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

Lucas BD, Kanade T (1981) An Iterative Image Registration Technique with an Application to Stereo Vision. In: International Joint Conference on Artificial Intelligence - Volume 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'81, pp 674–679

Mahon I, Williams SB, Pizarro O, Johnson-Roberson M (2008) Efficient view-based slam using visual loop closures. IEEE Transactions on Robotics 24(5):1002–1014, DOI 10.1109/TRO.2008.2004888

Mair E, Hager GD, Burschka D, Suppa M, Hirzinger G (2010) Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In: Proceedings of the European Conference on Computer Vision (ECCV'10), DOI 10.1007/978-3-642-15552-9{_}14

Matas J, Chum O, Urban M, Pajdla T (2002) Robust wide baseline stereo from maximally stable extremal regions. In: Proc. BMVC, pp 36.1–36.10, doi:10.5244/C.16.36

Meltzer J, Gupta R, Yang MH, Soatto S (2004) Simultaneous localization and mapping using multiple view feature descriptors. In: Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol 2, pp 1550–1555 vol.2, DOI 10.1109/IROS.2004.1389616

Migliore D, Rigamonti R, Marzorati D, Matteucci M, Sorrenti DG (2009) Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments

Moranna Ra, Martin RD, Yohai VJ (2006) Robust Statistics. Wiley

Moreels P, Perona P (2007) Evaluation of features detectors and descriptors based on 3d objects. Int J Comput Vision 73(3):263–284, DOI 10.1007/s11263-006-9967-1, URL http://dx.doi.org/10.1007/s11263-006-9967-1

Mouragnon E, Lhuillier M, Dhome M, Dekeyser F, Sayd P (2006) Real time localization and 3d reconstruction. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol 1, pp 363–370, DOI 10.1109/CVPR.2006.236

Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. In: In VISAPP International Conference on Computer Vision Theory and Applications, pp 331–340

Mur-Artal R, Tardós JD (2014) Fast relocalisation and loop closing in keyframe-based slam. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on, pp 846–853, DOI 10.1109/ICRA.2014.6906953

Mur-Artal R, Montiel JMM, Tardos JD (2015) ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics PP(99):1–17

Newcombe RA, Davison AJ (2010) Live dense reconstruction with a single moving camera. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, pp 1498–1505

Newcombe RA, Lovegrove SJ, Davison AJ (2011) Dtam: Dense tracking and mapping in real-time. In: Proceedings of the 2011 International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, ICCV '11, pp 2320–2327, DOI 10.1109/ICCV.2011.6126513, URL http://dx.doi.org/10.1109/ICCV.2011.6126513

Nistér D (2004) An efficient solution to the five-point relative pose problem. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on 26(6):756–77

Pinies P, Tardos J (2008) Large-scale slam building conditionally independent local maps: Application to monocular vision. Robotics, IEEE Transactions on 24(5):1094–1106, DOI 10.1109/TRO.2008.2004636

Pirchheim C, Reitmayr G (2011) Homography-based planar mapping and tracking for mobile phones. In: Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on, pp 27–36, DOI 10.1109/ISMAR.2011.6092367

Pirchheim C, Schmalstieg D, Reitmayr G (2013) Handling pure camera rotation in keyframe-based slam. In: Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on, pp 229–238, DOI 10.1109/ISMAR.2013.6671783

Pirker K (2010) Histogram of oriented cameras - a new descriptor for visual slam in dynamic environments. In: Proceedings of the British Machine Vision Conference, BMVA Press, pp 76.1–76.12, doi:10.5244/C.24.76

Pirker K, Rüther M, Bischof H (2011) CD SLAM - Continuous localization and mapping in a dynamic world. In: IEEE International Conference on Intelligent Robots Systems (IROS), IEEE, pp 3990–3997

Pretto A, Menegatti E, Pagello E (2011) Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp 3289–3296, DOI 10.1109/ICRA.2011.5980206

Pupilli M, Calway A (2005) Real-time camera tracking using a particle filter. In: In Proc. British Machine Vision Conference, pp 519–528

Rey-Otero I, Delbracio M, Morel J (2014) Comparing feature detectors: A bias in the repeatability criteria, and

how to correct it. CoRR abs/1409.2465, URL http://arxiv.org/abs/1409.2465

Rosten E, Drummond T (2006) Machine Learning for High-speed Corner Detection. In: 9th European Conference on Computer Vision - Volume Part I, Proceedings of the, Springer-Verlag, Berlin, Heidelberg, ECCV'06, pp 430–443

Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: An efficient alternative to SIFT or SURF. In: International Conference on Computer Vision (ICCV), pp 2564–2571

Scaramuzza D, Fraundorfer F (2011) Visual odometry [tutorial]. IEEE Robotics Automation Magazine 18(4):80–92, DOI 10.1109/MRA.2011.943233

Shi J, Tomasi C (1994) Good features to track. In: Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on, pp 593–600

Silveira G, Malis E, Rives P (2008) An efficient direct approach to visual slam. Robotics, IEEE Transactions on 24(5):969–979, DOI 10.1109/TRO.2008.2004829

Smith P, Reid I, Davison A (2006) Real-time monocular slam with straight lines. pp 17–26, URL http://hdl.handle.net/10044/1/5648

Strasdat H, Montiel J, Davison A (2010a) Scale drift-aware large scale monocular slam. The MIT Press, URL http://www.roboticsproceedings.org/rss06/

Strasdat H, Montiel JMM, Davison AJ (2010b) Real-time monocular SLAM: Why filter? In: Robotics and Automation (ICRA), IEEE International Conference on, pp 2657–2664

Strasdat H, Davison AJ, Montiel JMM, Konolige K (2011) Double Window Optimisation for Constant Time Visual SLAM. In: International Conference on Computer Vision, Proceedings of the, IEEE Computer Society, Washington, DC, USA, ICCV '11, pp 2352–2359

Tan W, Liu H, Dong Z, Zhang G, Bao H (2013) Robust monocular SLAM in dynamic environments. 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) pp 209–218

Tomasi C, Kanade T (1991) Detection and Tracking of Point Features. Tech. rep., International Journal of Computer Vision

Torr P, Fitzgibbon A, Zisserman A (1999) The Problem of Degeneracy in Structure and Motion Recovery from Uncalibrated Image Sequences. International Journal of Computer Vision 32(1):27–44

Torr PHS, Zisserman A (2000) MLESAC. Computer Vision and Image Understanding 78(1):138–156

Wagner D, Reitmayr G, Mulloni A, Drummond T, Schmalstieg D (2010) Real-time detection and tracking for augmented reality on mobile phones. Visualization and Computer Graphics, IEEE Transactions on 16(3):355–368, DOI 10.1109/TVCG.2009.99

Weiss S, Achtelik MW, Lynen S, Achtelik MC, Kneip L, Chli M, Siegwart R (2013) Monocular vision for long-term micro aerial vehicle state estimation: A compendium. Journal of Field Robotics 30(5):803–831, DOI 10.1002/rob.21466, URL http://dx.doi.org/10.1002/rob.21466

Williams B, Reid I (2010) On combining visual slam and visual odometry. In: Proc. International Conference on Robotics and Automation

Yousif K, Bab-Hadiashar A, Hoseinnezhad R (2015) An overview to visual odometry and visual slam: Applications to mobile robotics. Intelligent Industrial Systems 1(4):289–311

Zhou H, Zou D, Pei L, Ying R, Liu P, Yu W (2015) Structslam: Visual slam with building structure lines. Vehicular Technology, IEEE Transactions on 64(4):1364–1375, DOI 10.1109/TVT.2015.2388780