Visual SLAM for Driverless Cars: A Brief Survey

German Ros*, Angel D. Sappa†, Daniel Ponsa* and Antonio M. Lopez*
gros@cvc.uab.es, asappa@cvc.uab.es, daniel@cvc.uab.es, antonio@cvc.uab.es
*Computer Vision Center and Computer Science Dpt. UAB,
Campus UAB, 08193, Bellaterra, Spain
†Computer Vision Center, Campus UAB, 08193, Bellaterra, Spain

***Abstract*— This paper presents a brief survey of Visual SLAM methods in the context of urban ground vehicles. For this, we have reviewed relevant works, which present interesting ideas applicable to future designs of VSLAM schemes for urban scenarios. Our analysis aims to provide a global picture of state-of-the-art VSLAM systems, using a simple taxonomy based on an identified standard pipeline. This helps to show the global and consistent structure behind the different aspects that form these approaches. In addition, we also bring to the reader a set of useful tools, such as freely available code and datasets, that could help develop and test new VSLAM systems.**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM), was originally conceived as the problem of having an autonomous robot that creates a consistent map of its environment and localizes itself within this map [1]. It has been widely studied by the robotics and the computer vision communities during the last two decades, and as a consequence the original definition has been extended and many special cases have arisen. One of this cases is Visual SLAM (VSLAM), which restricts the used sensors to be passive vision-based, i.e., cameras. The use of cameras allows the development of accurate autonomous systems at the same time that decreases costs and overall energy consumption.

When we take a look into the intelligent vehicles literature, is easy to find many successful approaches that make use of active sensors, such as LIDAR, to acquire the data. Good examples of this are [2], [3] and [4], which describe practical approaches used in international competitions (e.g., the DARPA Grand Challenge, and the European Land Robot Trials), performing the first tests of these ideas in real conditions. These sensors can simplify the underlying estimation and mapping stages while producing remarkably good results. Such simplification is achieved by shifting part of the complexity from the SLAM stage to the acquisition stage, i.e., acquiring dense clouds of 3D points with a laser simplifies the remain stages.

However, developing SLAM approaches based on active sensors might be an important drawback with a view to their future introduction in driverless cars. This kind of sensors are very expensive nowadays, reaching in some cases a cost ten times higher than the vehicle. Even assuming a drastic decrease of the cost of these sensors, they still present a critical problem regarding their excessively high energy consumption. These facts manifest the necessity of considering lower-cost alternatives, as the ones provided by cameras and VSLAM approaches. In this context, we must highlight some notable works as the presented in [5], [6] and [7], where different authors show that using VSLAM for driverless cars — from now on VSLAM-DC — is a feasible task.

After the analysis of these contributions along with many others, we observe that VSLAM systems aiming to be useful in driving assistance tasks typically share a general common anatomy. In the remainder of this paper we discuss about the design decisions and the characteristics of such anatomy, aiming to define a set of standard building blocks.

II. VSLAM-DC, DESIGN FROM NEEDS

Before describing the "anatomy" of VSLAM-DC systems, we consider appropriate to start discussing about two critical aspects that are sometimes forgotten; the assumptions made regarding the environment, which can be seen as our input; and the type of maps generated by the VSLAM method, which is an important part of the output.

Accordingly, the first question to state should be; which model of the environment might produce best results for VSLAM-DC applications. Should we consider it metric, topological, a hybrid combination of both, or something completely different? This question is better addressed when related to the main tasks needed to build intelligent vehicles, which according to our bibliography analysis are:

- Global planning: goal selection, path calculation (shortest path, safest path, etc.).
- Local motion planning: steering control, velocity control, lateral maneuverability, etc.
- Obstacle avoidance: terrain labeling, road detection, object detection.
- Traffic laws enforcement: sign detection and recognition.

By considering these tasks, we are restringing the design of these methods, as they must have the appropriate characteristics to help performing them.

Starting our analysis from the first task, it turns out that it requires the calculation of the current vehicle position related to a set of known areas. However, if we try to model our environment metrically, long trajectories will produce notorious errors. This is, trying to generate long-scale maps, over thousands of kilometers, while keeping drift under acceptable values, is a hard task. A solution for this problem could be the use of topological maps for global level

ISBN: 978-84-695-3472-4

purposes. In this way, places are represented as nodes in a graph, and they can be recognized according to their visual models. An application of this idea can be found in [8].

On the other hand, the later three tasks need to make decisions regarding vehicle environment in a local neighborhood. Thus, a topological representation of the world does not cover the necessities of these tasks. Nevertheless, this problem can be solved having a metric representation of the vehicle surroundings. Accordingly, the most appropriate way of representing the world seems to be a hybrid model that is locally metric and globally topological. This appears to be a trend in modern VSLAM-DC systems, as can be deduced from [5], [6], [9], and others.

A second key aspect to consider is the type of output that these systems should provide. As we have just justified, some of the previous tasks need accurate local information to carry out their mission. We defend the idea that such information has to be encoded in a 3D dense map in order to provide fine details to perform efficient maneuvering and avoid obstacles. The literature shows some works dealing with this issue, as in [10] and [11], where a near real-time algorithm for dense maps creation is proposed.

Even though dense representations are a desired feature, they come with the drawback of a high computational complexity (i.e., to manage raw dense maps is a slow process). In order to solve this problem, some authors have proposed more compact representations that generate simpler maps to reduce the overall computation. This is the case of the Stixel representation [12]; a technique that approximates vertical surfaces of a map with rectangular sticks in order to distinguish between free space and objects.

Open challenges: One of the most important challenges for creating real VSLAM-DC systems remains being the development of topological approaches that allow for building long-life visual maps. Those approaches should try to cover the necessity of sharing maps between vehicles, and reusing previously built maps in a long-term fashion.

III. Anatomy of VSLAM-DC methods

Fig. 1 presents a standard pipeline that shows the different stages of a general VSLAM-DC approach, being these: (*i*) visual cues acquisition, (*ii*) current parameters initialization, (*iii*) information management, (*iv*) loop-closure detection, and (*v*) optimization. Said in simple words, we use visual information to initialize an estimation of the current vehicle position and the environment map. Then, a subset of all the available information is used to perform an optimization process that produces a refined version of our parameters, i.e., vehicle trajectory and map.

For the sake of simplicity, we consider convenient to arrange the posterior bibliographic analysis according to these stages, thereby helping to facilitate the understanding of the fundamental concepts behind VSLAM. Accordingly, a detailed description of all these stages is presented in the following subsections, where we also provide a review of the most relevant literature in the context of each stage.

However, before moving to the core of this paper, we must warn the reader regarding the necessity of including in our discussion techniques that have not been specifically designed for ground vehicles, but they come from the computer vision and robotics fields. This is principally due to the high influence that such communities have on the intelligent vehicles field, specially when talking about VSLAM. Many of the ideas proposed for autonomous vehicles have their origins in robotics or computer vision approaches. For this reason, current techniques arising in other fields are very important for the future of intelligent ground vehicles, and therefore, they have been considered here.

A. Visual Cues Acquisition

This stage deals with the problem of acquiring visual information (a.k.a. observations), needed in further stages, as for instance: establishing relationships between vehicle poses, creating the final map, etc.

The first aspect to be considered here is the number of available views of the scene, namely: one (monocular camera), two (stereo rig), or N (array of cameras). The use of more than one camera has an impact on the later processing time, although such an impact is not always negative in terms of computation time. Sometimes, acquiring information from more than one camera can help simplifying algorithms, as more constraints can be established from the extra information grabbed. An example of this can be seen in [6], where authors take advantage of a stereo camera to speed-up feature extraction.

VSLAM-DC literature is full of approaches that use monocular cameras as main input [13], [14], and [15]. The motivation behind this is simple, these cameras are cheap at the same time that allow for reaching good results. On the other hand, stereo cameras and arrays of cameras have been typically considered as expensive, although nowadays their cost is perfectly affordable. In addition, these cameras help producing better results, since they allow for direct depth measuring. For these reasons, to use multiple cameras seems to be more appropriate for VSLAM systems in the context of ground vehicle assistance. Proof of this are the successful approaches that use this hardware, as in [6], [5], [7].

Other important aspect arises from the nature of the observations. According to this, we can classify features as corners, edges, blobs, or higher level structures (e.g., planes, curves, etc.). The amount of information encoded on these features varies according to their complexity; i.e., planes encodes more information than corners, in exchange for being computationally more expensive.

An analysis of current approaches shows that corners are the most extended features. Corners are the raw material of many successful techniques, such as [16], [17], [6], [5]. However, the specific type of corner detector (and descriptor) varies between different approaches, being the most commonly used Fast-BRIEF [18], Fast-SIFT [6], Fast-SURF [19], and pure SIFT [20].

Edges-like features are not easy to find in the context of urban VSLAM. Probably the most notorious work is

ISBN: 978-84-695-3472-4

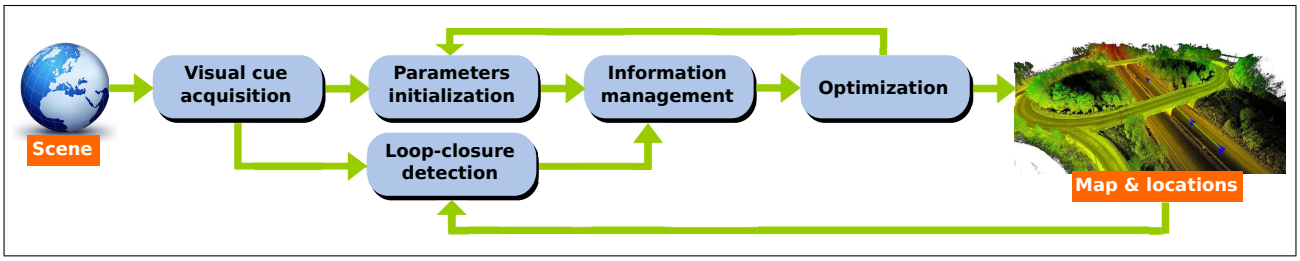

Fig. 1. A standard VSLAM pipeline for driverless cars.

[21], in which authors use parallel lines to recover dominant planes. The absence of more approaches that use edges or lines is due to the extra computational cost associated to their extraction. However, seminal works in the field of indoor VSLAM, as the presented in [13], and [14], point out the possible arrival of these approaches to VSLAM-DC. A system that combines corners and planes as input features has recently been presented in [22]. This work shows a novel parametrization to unify both features while avoiding plane fitting.

An important fact is that planes and other high-level features are formed from more basic entities, like corners or edges. This implies to perform a fitting process in order to create the desired feature. For this reason, although high-level features encode more information —being also more robust—, they are commonly relegated, due to real-time constraints.

Open challenges: There are some proofs in the literature which show that VSLAM systems get better as the amount of information, and its quality, increases [23]. Being able of grabbing more cues and then associate them, results in the creation of more constraints, which help refining the internal state (map and trajectory). Although the benefits of using features are given by quantity and quality, we need to be aware of practical limitations. Firstly, the amount of information we can process is bounded, due to computational constraints. Secondly, our capacity for associating information in posterior steps is not perfect, and depends on consensus methods [5]. Concerning both points, high-level features are more repeatable and easier to associate. Find a way to solve both problems while satisfying limited computational resources, remains as an open issue.

B. Parameters Initialization

This process creates an initial approximation of the current parameters (pose and map), where pose commonly stands for vehicle location and orientation. A new pose is initialized relative to a previous one. To do this, we use the constraints arising from features that are common to the current and previous frames. It is important to mention that pose initialization is critical for the correct performance of the overall system. Then, 3D landmarks take their values according to the initialized pose, since this serves as their reference frame. Thus, the initialization of 3D landmarks is conditioned to the initialization of the current pose.

This general process can be approached in different ways, being the most relevant based on the use of structure from motion (SfM) and optical flow, as pointed out in [24].

SfM approaches track a set of sparse features throughout several frames, and use them to estimate the camera pose [24]. These kind of approaches are more consolidated, and form part of many relevant works such as [25], [26], [17] and [11]. A plethora of variations can be found in the robotics literature, but all of them share the same basis, which are: model generation based on statistical consensus — RANSAC-like methods; and a posterior non-linear refinement.

On the other side, optical flow, although intended for image field estimation, can be also used for camera pose initialization with some extra computations. At present, these methods generate dense representations trough a global energy minimization process. A good example of optical flow methods in the context of camera initialization is [19].

Nonetheless, the line that separates optical flow methods from SfM methods is becoming fuzzy as new methods arise. Comport et al. provide a good example of this, in [26], where they present a stereo technique which behaves in an SfM fashion, but uses dense information.

Open challenges: Substantial progresses have been done, over the last decades, for the pose initialization problem. We consider that SfM and optical flow methods have both reached a high level of maturity in the robotics and computer vision communities, and they are nowadays well understood. However, there is still a necessity for finding out which is the most suitable strategy for ground vehicle navigation. In this sense, SfM applications are present on many real projects, but optical flow methods can offer more accurate results. This shows a common trade-off between computation time and precision, although it might change in the near future.

C. Information Management

This stage is intended to present the most important issues associated to the modeling of VSLAM-DC systems. Once that we have acquired the visual cues, and also initialized the current parameters, we must arrange all the information in a consistent framework. There are different strategies for doing this “arrangement”, but all of them draw on the concept of SLAM as a graph of constraints, due to its consistency and simplicity.

In the case of VSLAM-DC, poses and landmarks (atomic units of a map) are represented as nodes; which are connected according to observability relationships, i.e., the observability of landmarks from a given pose. Normally, in this representation, vehicle poses and landmarks are treated as latent nodes;

ISBN: 978-84-695-3472-4

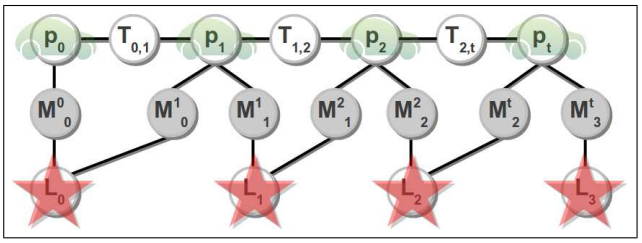

Fig. 2. Graph of constraints for a VSLAM problem. P_i refers to pose ith; $T_{i,j}$ is the transform between P_i and P_j; L_a is the ath landmark; and M_a^b represents the measurement of landmark L_a from pose P_b.

and the concept of measurement is added as the prime source of information. So, measurements — coming from images — provide us with information to constrain and estimate the hidden state of landmarks and poses, i.e., map and trajectory. Fig. 2 shows an example of this kind of graphs.

Such general idea is commonly formulated from two different perspectives, which are: the Bayesian networks (BN), and the more general graphs of algebraic constraints (please refer to [27] and [28] for more details).

The interesting thing is that, standard estimation techniques can be defined according to the way in which they use the information in the graph. For this survey we have identified three broad categories, which are: filtering techniques, global estimation (GE), and sliding window filters (SWF).

Filtering Techniques: within this category the information from past states is used to constraint current states. Once that previous poses haven been estimated $P_{0:t-1}$, filtering methods try to infer the parameters of current pose and landmarks. $P_{0:t-1}$ is just a node that encodes the information of the poses chain $P_0, P_1, \ldots, P_{t-1}$, but after a marginalization process it becomes into P_{t-1}^m. This implies that normally, filtering methods only make use of the previous pose and visual observations to predict new states. In addition the marginalization process is usually approximated (e.g., linearisation), thus altering the information encoded in P_{t-1}^m.

As examples of this category, let us highlight modern works like the one presented in [7], that makes use of filtering along with a RANSAC-based outlier rejection scheme to produce reliable urban localization and mapping. In the same context, in [17] authors propose to combine filtering with the outliers removal process, in a way that the former can guide and ease the task of the latter to achieve fast performance for outdoor navigation.

Global Estimation: This techniques are based on using all the available information in the graph, to estimate the full problem; i.e., the full trajectory and map, from t_0 to t_n. Accordingly, all the latent nodes of the graph are selected to be adjusted; that is, all the constraints are considered to produce the estimation, avoiding any kind of marginalization or reduction.

This family of methods has shown to produce the best results, although its computational complexity, cubic with the number of landmarks and quadratic with the number of poses, might be prohibitive for practical applications.

It is hard to find practical applications of pure global estimation in the context of VSLAM-DC. However, in [29], authors use global estimation in a two-level fashion for the large scale SLAM problem. First they solve part of the full problem with direct methods and then solve the rest with Preconditioned Conjugate Gradient. In [30] a global estimation strategy that use QR factorization is proposed to update the sparse information matrix.

Sliding Window Filters: These methods present an intermediate solution between filtering and global estimation techniques. One of their main characteristics is the selection of a subset of graph nodes (the sliding window), which usually are close to the current pose. Thus, only a part of the available information is used, and therefore, the computational complexity is reduced. Such a reduction favors real-time performance, although this means sacrificing part of the system accuracy.

In [18], authors, propose a SWF technique, which uses two sliding windows with different types of information. Other recent approach [5], uses a breadth-first-search heuristic, which selects the set of nodes to be optimized, based on the variation of their reprojection error.

An important subgroup that arises within SWF schemes is pose-graph approaches. These methods marginalize out landmarks, and attempt for estimating just vehicle poses, producing a skeleton of poses. This strategy is used in [19], where authors attempt to reduce the drift associated to scale, rotation, and translation when revisiting places in large-scale problems.

In a similar fashion, [31] shows an application of pose-graph systems with a good performance recovering trajectories, even when the algorithm is provided with poor initial poses. This is achieved by a stochastic gradient descent method, which demonstrate its results in small areas.

Open challenges: It is important to note that, those techniques using the full original information, without applying a marginalization process, produce better results. However, such trend comes with an elevated computational cost, that makes real urban problems intractable. For these reasons, sliding windows filters and pose-graph strategies have become more suitable alternatives to solve these problems. It is fair to say that methods here presented have reached a high level of maturity over the last years. All the categories have evolved, generating approaches that produce fast and reliable results. Nevertheless, in order to integrate VSLAM in real vehicle navigation systems, we still need to the efficiency of previous methods. We consider that more research should be done on exploiting information in sliding windows filters and pose-graph systems. The main goal should be finding the best way of using available information, at the same time that performing in real-time.

D. Loop-closure Detection

Loop-closing is the action of associating previously seen areas, or features, with the current ones. This association produces a fusion of pose nodes and landmarks that were considered as different entities until then. In this fashion, we can recover from situations that were wrong, and refine previous results.

ISBN: 978-84-695-3472-4

To perform loop-closure detection, first a new pose is initialized and then added as a node of the graph. Landmarks visible from this node are also added to the graph and associated to such pose. After that, if our loop-closure method detects a match between these landmarks and a set of previously seen landmarks (associated to a pose node) it can conclude that both nodes are representing the same pose, and therefore, must be merged in one. If not, the added node is kept in the graph along with its associated landmarks.

When two nodes have to be merged, the previous estimations of some poses and features should be readjusted (drift correction). If a large number of nodes were involved in this process (e.g., think in a long circular chain made up by pose nodes), most of them could require an adjustment. This process can be computationally expensive in cases when vehicles have traveled long paths, although we will show some approaches that avoid this problem.

Before starting the review of the different approaches tackling with this problem, we consider important to bring back the concept of VSLAM-DC as a two level framework, that was introduced in section II. Such an idea is specially important here, since loop-closure methods can operate in both local and global levels, although the formulation of these methods is significantly different, as we show below.

At the local level, loop-closing methods have to detect features based on geometric constraints, such as distances between points; and also similar visual appearance in a low level fashion. This idea is applied in [18], where authors use the reprojection error of 3D points to find correspondences. In [15] authors show a technique which makes use of both relative distances, and visual appearance constraints. In a similar fashion, authors of [9] use CenSuRe features as part of their loop-closure approach for urban scenes. More examples of these methods, together with a more detailed description can be found in [32].

From the viewpoint of the so-called global level, loop-closure strategies need to detect places, in a more abstract way. This is the same as in metric loop-closure methods, but the concept of place should be understood in a loose fashion, as determining where a place begins, and where it ends is a hard task. These definitions lend themselves to be embedded in a topological representation, in where distance information is neglected. Then, in order to be recognized, each place has associated an appearance model.

One of the most successful approaches is presented in [8]. There, the problem is formulated as a Bayesian network, in where each place is a node (here places are defined in a temporal context, i.e., every several frames). This formulation associates visual models, based on bag-of-words, to each place, and then, when a new node arrives, they calculate its posterior based on a maximum a posteriori strategy. From this, the obtained results represent the probability, for the new node, of being a previously observed place. This approach, apart of being general, produces reliable loop-closure detection over distances longer than 1,000 km.

Other authors propose topometric approaches, which cover both local and global levels simultaneously. As an instance, [33] shows a topometric system that makes use of covisibility graphs and dynamic bag of words to close loops without defining places explicitly.

Open challenges: We consider that loop-closure methods working at local level are robust enough to accomplish their mission. However, this is not true for global level methods, which still need to find out the way of improving their repeatability when the environmental conditions change. In short, works like [8] might be improved, allowing them to deal with drastic changes of illumination, occlusions, etc.; since these situations are common in urban scenes.

E. Optimization

In subsection III-C, we showed different ways of managing the information that defines VSLAM problems. Here we provide a complement that shows some of the most important methods used to estimate the parameters of given models.

One of the most used techniques, in the context of filtering schemes, is the so-called Extended Kalman Filter (EKF). This method is an improved version of the classical Kalman Filter (KF), which attempts to model non-linear systems by means of a linearisation process.

Although, these kind of approaches were proposed in SLAM more than two decades ago, modern approaches are still using EKF. This is the case of [17], where authors propose a robust VSLAM system based on EKF and an improved version of RANSAC.

More general techniques, such as bundle adjustment (BA), attempt for solving problems formulated within a global estimation framework. Bundle adjustment is based on a least-square optimization scheme, which exploits the sparsity patterns arising in some problems to speed-up the process. It has been widely used in the photogrammetry community, producing solutions to complex systems involving thousands or millions of variables. However, in real-time VSLAM systems, BA is applied to solve sub-parts of the full problem, as showed in [25]. Furthermore, the original bundle adjustment has evolved, giving rise to new algorithms that follow the pose-graph and SWF principles. This change makes BA more suitable to real-time performance and VSLAM applications for ground vehicles. Proof of it are successful approaches, such as [5], [6], and [18]; which achieve real-time performances while maintaining fairy precise results.

Open challenges: We note as a future goal, to keep developing optimization methods, that are able to exploit special properties of VSLAM systems for urban navigation.

IV. Available Resources in VSLAM

To know which software and data sets are currently available for the VSLAM community is something very important, as they simplify new developments and stablish a common framework for comparisons. For these reasons, we want to contribute by presenting a collection of resources (software and data sets) that we consider relevant for the problem of VSLAM-DC. However, adding those resources as a part of this paper is not the best approach, since they contain links that change and become outdated very quickly.

ISBN: 978-84-695-3472-4

To solve this problem we present a digital version of these resources in the following web page: `www.cvc.uab.es/adas/projects/slam`, that we will keep updated. In addition, this allows members of the scientific community to contribute to the creation and management of these resources. As an example of such contents we show some relevant data sets in Table I.

TABLE I
DATA SETS FOR VISUAL URBAN SLAM, SHOWING RESOURCE NAME; VISUAL SENSORS (V), I.E. MONOCULAR CAMERA (M), STEREO RIG (S), OMNI CAMERA (O); LASERS (L); INERTIAL MEASUREMENT UNIT (IMU); GPS; AND GROUND TRUTH DISPONIBILITY (GT)

Resource	V	L	IMU	GPS	GT
MIT Darpa Urban Challenge Dataset	M	Y	Y	Y	Y
Ford Campus Vision and Lidar Dataset	O	Y	Y	Y	Y
Karlsruhe Stereo Video Sequences	S		Y	Y	Y
.enpeda. Project datasets	S				Y
Victoria Park dataset	O		Y	Y	Y
The New College Vision and Laser Data Set	S+O	Y			
The Cheddar Gorge data set	S	Y	Y	Y	Y
The New college dataset (FabMap version)	M			Y	Y
Stereo Versailles Round-about Sequence	S				
The Marulan Datasets	M	Y	Y	Y	Y

V. CONCLUSIONS

Throughout this paper we have shown some of the most important aspects of VSLAM-DC techniques. The use of an execution pipeline as the unifying element, allows us to create a clear global picture of the state-of-the-art approaches.

We consider that, in general, VSLAM-DC implementations are beginning to reach good levels of maturity, although there are many aspects to improve in order to produce practical systems. As remarked at the "open challenge" paragraphs, it is necessary to start working on new architectures that combine topometric schemes, with rich 3D metric maps, and novel ways of managing and optimizing the information.

Our forecast is that VSLAM community will remain very active during the upcoming years.

ACKNOWLEDGMENT

This work has been supported by Universitat Autnoma de Barcelona and the Spanish government, by the projects TIN201125606 (SiMeVé); Consolider-Ingenio 2010, MIPRCV (CSD200700018); TRA201129454C0301 (eCo-DRIVERS); and TIN201129494C0302 (FireWATCHER).

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part i the essential algorithms," *IEEE Robot. Automat. Mag.*, vol. 13, no. 2, June 2006.

[2] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, Sep 2008.

[3] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2010.

[4] T. Luettel, M. Himmelsbach, M. Manz, A. Mueller, F. von Hundelshausen, and H.-J. Wuensche, "Combining Multiple Robot Behaviors for Complex Off-Road Missions," in *Proc. IEEE Int. Conf. Intell. Transp. Systems*, Washington, DC, USA, Oct. 2011.

[5] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast-scale outdoor navigation using adaptive relative bundle adjustment," *Int. J. Robot. Res.*, vol. 29, Jul 2010.

[6] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "Rslam: A system for large-scale mapping in constant-time using stereo," *Int. J. Comput. Vision*, vol. 94, Sep 2011.

[7] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *Proc. IEEE Intell. Veh. Symp.*, Jun 2010.

[8] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *Int. J. Robot. Res.*, vol. 30, Aug 2011.

[9] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *IEEE Trans. Robot.*, vol. 24, Oct 2008.

[10] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Proc. IEEE Intell. Veh. Symp.*, Jun 2011.

[11] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Proc. Asian Conf. Comput. Vision*. Berlin, Heidelberg: Springer-Verlag, 2011.

[12] H. Badino, U. Franke, and D. Pfeiffer, "The stixel world - a compact medium level representation of the 3d-world," in *Proc. of the 31st DAGM Symposium on Pattern Recognition*. Berlin, Heidelberg: Springer-Verlag, 2009.

[13] P. Smith, I. Reid, and A. Davison, "Real-Time Monocular SLAM with Straight Lines," in *British Machin Vision Conf.*, vol. 1, Aug 2006.

[14] E. Eade and T. Drummond, "Edge landmarks in monocular slam," *Image and Vision Comput.*, vol. 27, Apr 2009.

[15] L. Clemente, A. Davison, I. Reid, J. Neira, and J. D. Tardós, "Mapping large loops with a single hand-held camera," in *Proc. Robot.: Science and Sys. Conf.*, Jun 2007.

[16] T. Senlet and A. Elgammal, "A framework for global vehicle localization using stereo images and satellite and road maps," in *Proc. Int. Conf. Comput Vision, Workshops*, Nov 2011.

[17] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry," *J. Field Robot.*, vol. 27, Sep 2010.

[18] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual slam," in *Proc. Int. Conf. Comput. Vision*, 2011.

[19] H. Strasdat, J. M. M. Montiel, and A. Davison, "Scale drift-aware large scale monocular slam," in *Proc. Robot.: Science and Sys.*, Zaragoza, Spain, Jun 2010.

[20] A. Irschara, C. Zach, J. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recog.*, Jun 2009.

[21] Y. Cao and J. McDonald, "Improved feature extraction and matching in urban environments based on 3d viewpoint normalization," *Comput. Vision Image Underst.*, vol. 116, Jan 2012.

[22] J. Martínez-Carranza and A. Calway, "Unifying planar and point mapping in monocular slam," in *British Machine Vision Conf.*, 2010.

[23] H. Strasdat, J. Montiel, and A. Davison, "Real-time monocular slam: Why filter?" in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2010.

[24] K. Konolige, M. Agrawal, and J. Solà, "Large scale visual odometry for rough terrain," in *Proc. Int. Symp. Research Robot.*, Nov 2007.

[25] E. Mouragnon, M. Lhuillier, D. M., F. Dekeyser, and P. Sayd, "Generic and real-time structure from motion using local bundle adjustment," *Image and Vision Comput.*, vol. 27, Jul 2009.

[26] A. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3d visual odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr 2007.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[28] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Proc. Int. Conf. Comput. Vision, Workshops*. London, UK: Springer-Verlag, 2000.

[29] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C. Thorpe, "Subgraph-preconditioned conjugate gradients for large scale slam," in *Proc. IEEE Int. Conf. Intell. Robots and Sys.*, Oct 2010.

[30] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, Dec 2008.

[31] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2006.

[32] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular slam," *J. Robot. Autonom. Sys.*, vol. 57, Dec 2009.

[33] C. Mei, G. Sibley, and P. Newman, "Closing loops without places," in *Proc. IEEE Int. Conf. Intell. Robots and Sys.*, Oct 2010.