Computers in Industry 99 (2018) 53–68

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Portable 3-D modeling using visual pose tracking

Klaus H. Strobl*, Elmar Mair[1], Tim Bodenmüller, Simon Kielhöfer[2], Tilo Wüsthoff, Michael Suppa[3]

Robotics and Mechatronics Center, German Aerospace Center (DLR), D-82230 Wessling, Germany

ARTICLE INFO

Article history:
Received 17 August 2017
Received in revised form 8 March 2018
Accepted 15 March 2018
Available online xxx

Keywords:
3-D modeling
Pose tracking
SLAM
Visual odometry

ABSTRACT

This work deals with the passive tracking of the pose of a close-range 3-D modeling device using its own high-rate images in realtime, concurrently with customary 3-D modeling of the scene. This novel development makes it possible to abandon using inconvenient, expensive external trackers, achieving a portable and inexpensive solution. The approach comprises efficient tracking of natural features following the Active Matching paradigm, a frugal use of interleaved feature-based stereo triangulation, visual odometry using the robustified V-GPS algorithm, graph optimization by local bundle adjustment, appearance-based relocalization using a bank of parallel three-point-perspective pose solvers on SURF features, and online reconstruction of the scene in the form of textured triangle meshes to provide visual feedback to the user. Ideally, objects are completely digitized by browsing around the scene; in the event of closing the motion loop, a hybrid graph optimization takes place, which delivers highly accurate motion history to refine the whole 3-D model within a second. The method has been implemented on the DLR 3D-Modeler; demonstrations and abundant video material validate the approach. These types of low-cost systems have the potential to enhance traditional 3-D modeling and conquer new markets owing to their mobility, passivity, and accuracy.

1. Introduction

3-D modeling already assumed a central role in areas like industrial inspection and recognition, reverse engineering, cultural heritage, medical imaging, computer graphics, and robotics. Other areas like leisure gaming, human–computer interaction, robotics, forensics, agriculture, and construction show a less direct requirement for 3-D modeling, but are increasingly taking advantage of it as a means to solve the visual perception problem. Visual perception is the process by which visual sensory information about the environment is received and interpreted; it is believed that it is through the explicit formation of 3-D models that a considerable number of the challenges on visual perception will be eventually solved. This is, of course, subject to the performance, flexibility, and cost of 3-D modeling devices.

Several factors like object self-occlusion, object size, or limited field of view make it impossible for a 3-D modeling system to acquire a complete model in a single measurement step, especially in close-range. Multiple views (or multiple sensors) are required to merge data to a single model. The prevalent approach is to measure the position and orientation (pose) of the sensor while acquiring range data, thereby registering multiple views into the same frame of reference. A range of tracking systems, robotic manipulators, passive arms, turntables, CMMs, or electromagnetic devices are deployed for this purpose. These options are inconvenient for three reasons: *First*, they limit mobility; *second*, they require accurate synchronization and extrinsic calibration (and cannot be rearranged); *third*, they usually represent the largest and most expensive part of the 3-D modeling system.

In this work we present an overview on the state of the art of close-range 3-D modeling systems regarding their data registration concept. We then make the case for data registration by visual pose tracking in realtime and go on describing their adaption for close-range 3-D modeling devices, using the video captured by their own cameras. Cameras are preferred sensors in many areas because they are light, affordable, consume less energy, allow for a very accurate parametrization of its operating model, and still they gather a plethora of information (both radiometric and geometric) within a single, rapid measurement. Further benefits exist:

* Corresponding author.
E-mail address: klaus.strobl@dlr.de (K.H. Strobl).
URL: http://rmc.dlr.de (K.H. Strobl)

[1] Present address: X Development LLC, 1600 Ampitheatre Pkwy, Mountain View, CA 94043, U.S.

[2] Present address: Ringstraße 38, D-86911 Dießen am Ammersee, Germany.

[3] Present address: Roboception GmbH, Kaflerstr. 2, D-81241 Munich, Germany.

https://doi.org/10.1016/j.compind.2018.03.009
0166-3615/© 2018 Elsevier B.V. All rights reserved.

cameras are non-contact sensors, thus free-floating, and passive since they do not need to project or exert action on the environment. In addition, visual pose tracking becomes inherently calibrated and synchronized with further image-based sensing.

Visual pose tracking is a hard problem because geometric information becomes entangled in radiometric and perspective geometric issues. Following distinct regions of interest in the images in realtime (**feature-based tracking**) is a popular technique to overcome this problem. This is especially demanding in close-range because features move faster than in medium- or long-range because they are also affected by camera translation. We proposed two novel schemes for efficient feature tracking on this type of devices: either leveraging an inertial measurement unit (IMU) [1] or adopting the Active Matching paradigm in Ref. [2] for more efficient tracking [3,4].

In the present case of cameras mounted on close-range scanning devices, highest accuracy in visual pose tracking is necessary as cameras feature small angular fields of view, which call for the concatenation of relative measurements (dead reckoning) so that errors readily accumulate. We propose **graph-based, nonlinear optimization** (keyframe-based bundle adjustment) on relative pose transformations and measurements, **parallel computing** of front-end, back-end and other sub-tasks, feature-based stereo vision, as well as **loop closure detection** for error compensation. Even in the case that everything else fails, **appearance-based recognition** of older features is provided so that pose tracking can be resumed. These contributions have been described in detail in Refs. [5,6].

Finally, since manual 3-D scanning requires visual feedback to the user, a **streaming surface reconstruction** method is presented that delivers realistic 3-D models *in-the-loop* during scanning as well as refined models promptly after loop-closing corrections.

We implement these methods on the DLR 3D-Modeler [7], creating the first 3-D scanner for close-range applications that localizes itself passively from its own images in realtime, at a high data rate. Systems of this type deliver more accurate results than depth sensors using coded infrared light (e.g., Kinect, Xtion) by an order of magnitude [8]. The DLR 3D-Modeler is a low-cost, hand-held device for accurate geometric and radiometric reconstruction of close-range objects in realtime that was originally tracked by robotic manipulators or external infrared light trackers (Fig. 1).

The remainder of this article is as follows: An extended survey on related 3-D modeling devices, their pose tracking techniques, and more specifically visual pose tracking is delivered in Section 2. In Section 3 we present the visual pose tracking algorithms implemented in the DLR 3D-Modeler. We validate the approach with experiments in Section 4 and supplementary videos.

2. State of the art

In this section we review 3-D modeling work with regard to their 3-D data registration concept—provided the system meets our requirements, i.e., is non-contact and light-weight. We focus on mature, commercial systems and only mention research work in the areas where commercial systems are missing. Lastly, we elaborate on the real-time variants of visual pose tracking for online 3-D data registration.

Fig. 1. The portable DLR 3D-Modeler used for cultural heritage preservation.

2.1. Data registration by scan alignment

Dense depth sensors that provide 2-D range images (i.e., 2.5-D images) yield rich surfaces that allow for data registration by 3-D matching, without the necessity for explicitly estimating sensor motion. This is not possible, however, in the case of 1-D range images (e.g., laser stripe triangulation).

3-D matching is computationally demanding because correspondence search is on higher dimensionality compared to traditional 2-D image registration. Additionally, data overlapping is required, which has to be detected in advance out of raw depth data and perhaps some motion priors. For these reasons, scan alignment is often being performed off-line, in an interactive way. The estimation involves an optimization in the form of the minimization of a distance metric between scans (e.g., ICP [9]). Different metrics and ICP modifications have been proposed for improved robustness against noise and efficiency [10]. With the recent advent of general-purpose computing on GPUs, real-time implementations of ICP have been presented (e.g., sequential multi-scale ICP on RGB-D data [11]). Other authors opt for bootstrapping ICP by feature-based visual pose tracking, see Ref. [12] and Section 2.4. Indeed, Coudrin et al. for the company Noomeo SAS use visual pose tracking for initial estimation for subsequent ICP optimization [10]. They are unable to use if for online data registration because they use densely projected patterns, which preclude concurrent visual tracking. They use interleaved stereo frames where the projected pattern is switched off, so that 3-D modeling and pose tracking are innerly desynchronized. In the end, half of the images serve 3-D modeling whereas the other serve as an initialization step for ICP.

2.2. Data registration by external pose tracking

3-D pointcloud registration is an over-determined problem with as few as 6 degrees of freedom (DoF). It is common practice to take data subsets to simplify the estimation problem. In addition, its convergence is subject to a high degree of unpredictability as it is strictly dependent on the particular surface geometry. We would benefit from a registration method that is independent of the 3-D data. It is well known that the sensor motion estimation problem (6 DoF) yields that same solution, although represented in the camera reference frame instead of in the object reference frame.

The use of traditional absolute positioning systems attached to a 3-D sensor is arguably the most straightforward approach for solving this problem. Due to their robustness and accuracy, the systems listed below became widespread and are the dominant (commercial) 3-D modeling devices in close-range:

- *External, optical tracking systems* are used by Northern Digital Inc., Metris NV, and Steinbichler Optotechnik GmbH. These systems detect and track artificial (e.g., infrared-reflecting) markers attached to the 3-D sensor. They seem convenient to hand-held operation due to the absence of a rigid contact to the tracking sensor. On second sight, however, the user feels strongly limited because of their small tolerance to sensor rotation owing to visibility constraints. Furthermore, since the spatial distribution of the markers is limited, the accuracy of orientation estimation is generally poor.
- *Passive arms* are used by FARO Technologies Inc., KREON Technologies, RSI GmbH, Metris NV, and ShapeGrabber Inc. Passive arms, or even robotic manipulators, are inconvenient for

manual operation. They are, however, the most accurate option for pose tracking—subject to their accurate synchronization and extrinsic calibration w.r.t. the sensor. Price and size are prohibitive in many applications.
- *Electromagnetic tracking systems* are chosen by Polhemus Inc. These devices resemble optical tracking in operation, but now it is not required for the sensor to maintain a free line of sight to any marker. Accuracy is dependent on the distance to the electromagnetic emitter and its signal can be affected by metallic structures.
- *Turntables* are used by Cyberware Inc. and Polygon Technology GmbH. These allow for inexpensive systems, but are limited to small, light objects and rarely allow for the generation of complete models.

The above absolute positioning systems have in common that they represent the *bulkiest and most expensive part of the eventual 3-D modeling systems*. Furthermore, they limit the system in mobility and flexibility, and are subject to accurate external calibration and synchronization. These strong limitations apply especially in the realm of robotics, where sensors are precisely meant to promote autonomy without imposing additional constraints.

2.3. Data registration by visual pose tracking

Since digital video cameras are already present in most close-range 3-D modeling systems, the estimation of the sensor motion from its own video footage is highly desirable to avoid using the above-mentioned systems. Motion estimation is feasible because, on a static scene, the camera motion is the only factor that accounts for varying perspective projection of the 3-D scene onto 2-D images. In addition, since visual pose tracking is in the camera frame, an external calibration step of the tracking system w.r.t. the camera is no longer required. Similarly, estimations become inherently synchronized with further visual sensing. From this idea two variants emerged:

- *Low-rate visual pose tracking* is used by Noomeo SAS in the Optinum™ scanners as an initialization stage for the alignment of dense range images.
- *High-rate visual pose tracking* is achieved by the Handyscan 3D scanners of Creaform Inc. (also marketed as ZScanner® by Z Corporation) and recently by the rc_visard sensor of Roboception GmbH.

The latter lie close to our goal of high-rate pose tracking from a video stream. In the case of the Handyscan, however, the necessity to adhere reflective markers to the objects is inconvenient. In fact, in a number of applications it is prohibited or impossible. Being one of the main motivations for using cameras the fact that they are non-contact, free-floating sensors, i.e., effectively passive to the scene, it is counterproductive to rely on adhesive markers. Furthermore, their dependency on active infrared illumination entails limitations. In 2017, the rc_visard 3-D sensor was introduced by Roboception GmbH. It features high-rate visual pose tracking similar to our approach in Ref. [1], leveraging low-rate depth estimation by SGM stereo vision together with high-rate feature tracking and IMU data. The sensor is designed to perceive the environment of a robot in 3-D, which usually is at a lower level of detail than 3-D modeling of single objects in close-range. To generate depth data, the sensor uses the SGM stereo vision algorithm, which delivers precise, relatively low-resolution (640×480 pixels) data at low-rate (3 Hz) [13].

The DAVID-Laserscanner is a commercially available, very simple scanner that works without an external tracking system [14]. The pose *of the laser projector* is estimated from images of a static camera that, at the same time, estimates projections depths by triangulation. The approach is fundamentally limited to a single view with potential, subsequent scan alignment.

For the remainder we concentrate on research work.

In Refs. [15,16] a self-referenced, hand-held cross-hair laser stripe profiler was presented. Its stereo camera makes use of fixed marker points, actively projected onto the scene, and localizes itself continuously by stereo triangulation w.r.t. these points. Actively projecting marker points onto a scene is inconvenient and, furthermore, limits flexibility since the cameras must see the markers the entire time. In addition, both laser profiler operation and texturing are influenced by active illumination. The algorithm seems to lack robustness, and efficiency considerations are not reported. Similarly, in Ref. [17] a pattern projector is used for dense multi-view stereo achieving high reconstruction precision of untextured models. On this occasion, the projector is rigidly attached to a stereo camera to achieve dense depth images of untextured surfaces from correlation-based stereo including a joint, multi-view optimization at low-rate. By algorithm design choice, however, the projector precludes texturing and field operation in sunlight.

Passive visual pose tracking approaches for 3-D modeling are reported next.

The approach in Ref. [18] uses projective reconstruction jointly with posterior self-calibration to estimate metric—yet unscaled—motion in uncalibrated image sequences. After that, bundle adjustment is used to refine the results. A similar approach in Ref. [19] makes partial use of a previous camera calibration for metric reconstruction. The approach is intended for dense stereo vision applications and is not real-time. Accuracy analyses are missing even though non-stochastic approaches to self-calibration compromise accuracy.

It is worth mentioning the instant Scene Modeler iSM device by MDA Ltd., Space Missions in Ref. [20]. The system produces 3-D models from hand-held stereo vision by the registration of views with scaled poses from visual pose tracking. In contrast with the objectives in this work, the system aims at mid-range operation using dense stereo vision. Stereo is computationally expensive and, therefore, frame-rate is low, which in turn makes pose tracking under unknown motion harder and essentially different from a high-rate variant. The problem is solved using SIFT features—which again are computationally expensive—as well as lower resolution footage.

Strobl et al. presented in Ref. [1] the first hand-held 3-D modeling device for close-range applications that localizes itself passively from its own images in realtime, at a high data rate. In that work, pose tracking was optionally supported by an on-board IMU for more efficient feature tracking. In Ref. [4] the authors present improved feature matching by Active Matching, see Ref. [21], that achieves remarkable tracking resilience without the need for inertial readings.

Finally, we mention a development by Newcombe and Davison on 3-D modeling from dense images by concurrent simultaneous localization and mapping (SLAM), so-termed DSLAM [22,23]. DSLAM aims at considering every single pixel of the video stream for structure estimation and interleaved pose tracking, maximizing information gathering and overall performance. It is hard to explicitly do without distinct features (cf. Section 2.4-I.) as features are, by definition, invariant under several aspects and can be better discriminated. Consequently, the method is limited to confined viewpoint areas and constant lighting conditions as it assumes brightness constancy (surface smoothing priors are introduced to partly relieve of this limitation). Still, viewpoint limitation is certainly unsuitable for full-body 3-D modeling. The current implementation is computationally very costly, leveraging on

GP-GPUs for real-time performance. Despite all that, DSLAM already reached improved performance concerning resilience to erratic camera motion, pose tracking accuracy (albeit unproven in experiments) and, most importantly, concerning its low hardware requirements, namely a single camera and a commodity computer featuring a GPU.

2.4. Visual pose tracking in realtime

Visual pose tracking is a hard problem because, in geometric terms, images merely convey 2-D information that originally stems from a higher dimensional space (e.g., 6 DoF of camera pose, full 3-D geometry of the scene, and intrinsic camera geometry). It is often just one among the latter parameters that we are interested in, yet *still have to infer them all* from 2-D images. This dimensionality reduction renders the problem often unsolvable using a single image. It is by increasing the dimensionality of the gathered data by more measurements that we can draw a distinction between the original, unknown parameters themselves, and infer their respective values. In doing so, we regularly exploit prior knowledge (e.g., on the rigidity of the scene, on Euclidean geometry, and on perspective projection).

In particular, there is a prevalent ambiguity in scene structure and camera pose estimation: it is impossible to discriminate between object size and camera range to that object. It is the chicken-and-egg problem that characterizes research in SLAM: motion estimation (localization) is straightforward on known 3-D geometry, whereas 3-D geometry estimation (mapping) in turn asks for known camera motion. As previously mentioned, tackling the problem of SLAM is solved by integrating data in time, when some parameters vary (e.g., camera motion, i.e., apparent perspective distortion) to differentiate them from others (e.g., static scene geometry).

To make matters worse, many applications require estimations in realtime (e.g., at 30 Hz). On the one hand, it is important to realize that less applications require a complete optimized motion history in realtime, but only a local solution—the full history can be delivered delayed in time. On the other hand, parts of the solution are really being required in realtime and, therefore, efficient methods are in demand. Temporal priors (e.g., on the dynamics of the system) can be used for improved performance. Next we address three key aspects for designing real-time visual pose tracking algorithms:

- *The representation of the structure of the scene.*
- *The storage of the associative visual measurements.*
- *Approximate solutions for real-time performance.*

I. The representation of the structure of the scene: Feature-Based vs. Dense, Direct Tracking. A picture might well be worth a thousand words, but not all visual information is created equal. Depending on the task at hand, some image regions convey more information than others [24–26]. Visual information can be reduced to regions of interest (points or corners, edges) that still allow for highly accurate inference. These features represent the *Merkwelt* necessary for pose tracking. As most of these regions of interest can be described in very concise, parametric ways, methods following this paradigm ought to be more efficient than *dense methods*, which compute pixelwise *directly* from dense, raw image data.[4] Furthermore, these regions are more invariant to viewpoint location (e.g., light reflection) and varying lighting conditions, which allows wide baseline matching for increased accuracy. Lastly, estimation on these separate regions is largely uncorrelated, i.e., statistical independence holds (unlike when using dense methods) and, therefore, optimal estimation using maximum likelihood methods is still justified. Admittedly, the feature-based estimation paradigm entails limitations on its own, like the feature selection, scene understanding, and data association issues. In general, feature-based methods are being preferred when designing visual pose tracking algorithms.

Feature-based methods utilize interest operators to *detect* salient/distinct regions of the images, i.e., fiducial points or features at repeatable, stable locations despite change of viewpoint. Salient regions arise either from texture or from geometry (e.g., object corners). In general, features from (planar) texture are preferred since corner projections are not invariant to viewpoint location due to self-occlusion. Well-known detectors are: Harris–Stephens [27] or Shi–Tomasi [28], the Laplacians LoG, DoG or DoB [29], MSER [30], SUSAN [31], SURF [32], and FAST [33]. Additionally, an operator for invariant *description* of these features is needed to be able to discriminate features against each other. Well-known descriptors are: planar, oriented patches [34], SIFT [35], GLOH [36], HOG [37], SURF [32], CenSurE [38], BRIEF [39], BRISK [40], ORB [41], FREAK [42], and KAZE [43]. In the case of repetitive patterns, context-aware, dense descriptors would be beneficial; this has been recently achieved leveraging convnets in the context of deep learning [44]. We speak of feature tracking when these descriptions are being matched in time, either starting from the anonymous output of a feature detector or based on camera/feature motion priors. In the former case, a current description is compared with a database of past descriptions, whereas in the latter case the current description is compared with a subset of that database (potentially just one description) within a reduced area of the image [45,46,4]. Of course, the matching method is descriptor-specific (e.g., normalized cross-correlation for planar patches or Hamming distances for BRIEF descriptors).

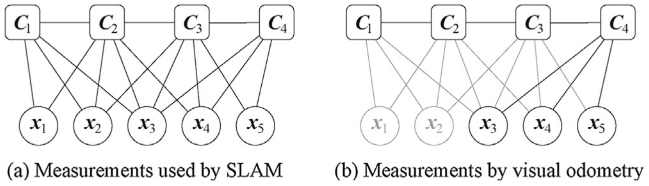
Setting an optimal framework for detection, description and matching of features is subject to trade-offs: a *general* descriptor is expected so that it is invariant to change of viewpoint or illuminance; at the same time, feature descriptions have to be distinctive and, therefore, *specific* to particular features.

Dense methods potentially are more accurate and locally robust than feature-based methods because their representations (whole images) are more informative than just features. They are, however, less invariant to change of viewpoint or illuminance, and therefore are being complemented with simplifying assumptions like brightness constancy [23,47,12]. In any case, the implementation of dense methods on current hardware is demanding both on computational and electric power, which keeps them away from cheaper, mobile implementations.

II. The storage of the associative visual measurements: visual odometry vs. visual SLAM. Both visual odometry (VO) by dead reckoning and visual SLAM (V-SLAM) incrementally estimate camera motion from video streams in realtime. For that purpose VO exclusively uses the last subsequent images—potentially more than two,[5] but then critically the total number of images considered is limited. If an image gets outside this scope, its associated information will not be explicitly used for motion estimation anymore [48–50]. On the other hand, V-SLAM may accumulate *all* information from past images, representing it either in the form of a graph of camera motions and measurements or in the form of a map, continuously updating them using present

[4] This conventional view is much-debated since the introduction of GP-GPUs.

[5] Using two frames for sequential motion estimation is subject to scale drift. It is by using at least three frames of matched features that estimations anchor in the original scale.



(a) Measurements used by SLAM (b) Measurements by visual odometry

Fig. 2. Graph on the measurements used for pose estimation at $\mathbf{C}_4$ by SLAM (a) and VO (b).

visual information (Fig. 2). VO does not maintain these types of representations of the environment.

Considering older information is convenient in two respects: *First*, the relative pose estimation accuracy is essentially improved. Since the graph or the map relates to older camera stations, relative pose estimations w.r.t. that older stations will be more accurate than performing repeated, relative pose estimation over unrelated frames. In addition, virtual parallax is bigger, therefore relative pose estimation more accurate on the assumption of constant image noise level. *Second*, the very existence of a map or a graph makes it possible to find older features again (loop closing), based either on their relative locations w.r.t. the camera or merely on visual descriptions; this is critical to further increase pose estimation accuracy. Indeed, it is only through closing loops that consistent graphs and drift-free camera motion estimation can be achieved in the presence of noise.[6] A downside to maintaining a map also exists: it is computationally expensive, as complex calculations are involved to obtain statistically optimal estimations. In addition, considerable amount of memory is used.

When performing VO the data quantity is limited to recent camera frames, which renders the estimation problem tractable. Still, some tricks are used to boost performance and ensure robustness against outliers. For instance, it is common practice to compute minimal relative motion solutions from either 3 [51–54], 5 [55], 6, 7 or 8 [56] feature points (depending on our knowledge of the structure and the camera), which are rapidly computed in closed-form, to obtain ballpark motion estimates. Coarser resolution can be also used. After that, the best solution may bootstrap a least squares optimizer minimizing reprojection errors (iterative refinement), potentially using more than two images (sliding window optimization yields optimal *motion* estimation [48,57,58]). Scene structure is usually unknown, and consequently feature matching may be erroneous. In order to detect outliers, the latter minimal solutions to the relative motion problem are often within a geometric hypothesize-and-test framework like RANSAC [59,60,50]. The final least squares solution may also concern a robustified residual function.

From an operational point of view, the essential difference between VO and V-SLAM can be summarized as follows: Whereas VO estimates camera motion from feature correspondences between selected images, V-SLAM estimates camera motion from a conceptual matching between current image features and a representation of the accumulated system state, which in turn stems from past feature tracking. It is generally acknowledged that hybrid solutions, running both processes potentially at different rates, are most effective as they complement one another [61–65].

III. Approximate solutions for real-time performance: the back-end of SLAM. The SLAM problem can be divided in two tasks: front-end and back-end. The motivation for this division is the unfeasibility of achieving overall optimal estimation in realtime. Front-end calculations deal with image processing and the arrangement of input data, and should run in realtime. Note that the arrangement of data includes the solution to the data association problem and that local pose tracking (or VO) in realtime may be of necessity to that end. On the other hand, back-end calculations concern the consistent representation of the data arranged by the front-end in the form of a graph of associated measurements or of a map. As the map grows and becomes interconnected, the complexity of this sub-task naturally grows—eventually becoming the bottleneck to optimally solving SLAM. Consequently, back-end methods dominated research on SLAM for the last decade, with methods that compute approximate solutions in realtime being preferred. Recently, however, a pertinent observation led to a different type of algorithms delivering far more accurate results: "*Global geometric representation is rarely being required in realtime*" [61]. More accurate estimations can be readily delivered *at a lower rate*, paving the way to a plethora of methods trading off efficiency against accuracy.

As a consequence of V-SLAM being preceded by SLAM, initial research adapted existing techniques (mainly using 2-D lidars) to visual input data, without actually realizing the two main challenges of V-SLAM: *First*, digital cameras feature a *narrower field of view* than 2-D scanners, which makes triangulation harder and the time window for feature tracking shorter; many noisy local relative estimations will now have to concatenate for dead reckoning motion estimation. *Second*, visual data spreads now in 3-D, stacking up *larger amounts of data* than former SLAM methods in 2-D. In fact, the first, best-known approach to V-SLAM by Davison in Ref. [66] used an Extended Kalman Filter (EKF), which delivered good, fast results if the map size was kept small concerning both, the number of features and the overall number of measurements. Early adopters rapidly noted this limitation, along with inconsistency in the estimations due to linearization errors and potential inadequacy of the Gaussian error models [67]. The preferred measure to ameliorate effects has been the decomposition of maps into submaps that become strictly uncorrelated from one another [68–70], which is at the cost of map accuracy.

The second major method for back-end estimation in V-SLAM is the particle filter (PF) or sequential Monte Carlo method [71]. A PF aims at more accurate and consistent estimations by representing estimation distributions as well as model noise by sets of particles. The size of the map that is manageable is, however, still limited as the number of required particles grows exponentially with the number of features and their dimensions. A variant of the PF was proposed called Rao-Blackwellized PF (e.g., FastSLAM) [72,73]. The authors observe that feature measurements are uncorrelated if they are conditioned to a particular path estimate of the camera. Consequently, feature maps can be efficiently computed using sparse EKFs associated to their respective pose particles. The principal drawback of PFs and its variants is the resampling step, which is introduced to eliminate improbable particles, keeping computational costs low; regrettably, the resampling step causes the lost of essential, small correlation densities (depletion problem) and consequently a loss of accuracy as well as eventual inconsistency.

As mentioned before, the two main drawbacks of exclusively using filtering methods (EKF, PF) for the back-end of V-SLAM are both their computational cost when dealing with a large number of features (map size) as well as their limited potential accuracy and inconsistency. This limitation is inherent to filtering approaches for the following reason: Filtering is about maintaining a compact state-space estimation of currently useful parameters by marginalizing out past estimations (e.g., past camera locations) so that less computations and memory are required. In doing so, artificial correlations between parameters

[6] VO systems may leverage IMU or GPS devices fusing data to overcome this problem.

(e.g., estimated feature positions) have to be produced since their current position estimations depend on common past camera locations that now have been removed from memory. Note that these correlations were non-existent at the moment of measurement, see Fig. 3(b). Even though these correlations can be rapidly processed if the number of features is low, the complexity of the algebra of non-sparse matrices (full of correlations) is cubic in the number of features, which rapidly renders filtering approaches ineffective since cameras gather many more features than 2-D scanners. This could be avoided if the original measuring locations were still being considered, leading to a sparse graph of constraints. It is precisely the algebra of sparse matrices that is fast to solve after all.

From this, a different paradigm for the back-end of V-SLAM arose: graph-based nonlinear optimization in near-realtime. The authors of the seminal work PTAM in Ref. [61] utilize the well-known optimal algorithm for concurrent estimation of scene structure and camera motion called bundle adjustment (BA) [74]. The basic idea was first formulated by Lu and Milios in Ref. [75], by which all motion data and measurements can be represented as a stochastic graph of nodes and edges (in V-SLAM: camera and feature locations and measurements, respectively). The goal is to find an optimal spatial configuration of the nodes that agrees with the constraints provided by the edges, by means of probabilistic inference (e.g., a nonlinear optimization). BA is known to be unsuitable for real-time estimation. However, the novel nature of off-the-shelf hardware featuring multiple cores for parallel computing gives the opportunity to perform BA in a real-time context: By computationally separating front-end and back-end calculations, BA can readily perform at lower rate without affecting local tracking performance at the front-end. It turns out that BA is less affected by both of the limitations of the aforementioned filtering methods. Still, its complexity linearly increases with the number of measurements and is cubic with the number of frames, which can quickly become prohibitive. It has been shown that, in the context of real-time SLAM, gathering many features per frame is preferable to processing many frames with less features, close in time [76]. Therefore, the authors proposed a variant of BA called keyframe-based BA (kBA) [61,58], which selects, in a heuristic way, the most informative frames to consider, see Fig. 3(c). If the number of keyframes is low, its complexity is effectively quadratic in the number of frames.

Regrettably, static, regularly-spaced keyframes do not sort well with the heterogeneous nature of V-SLAM in mobile systems. In the spirit of kBA, more flexible approaches arose that focus resources on different parts of the state space. Since there are many more features than frames, pose-to-pose graph-based optimizations like FrameSLAM perform well *in large-scale* by marginalizing out feature locations [62,77]. Marginalization may come at a cost of lower estimation accuracy if the optimized poses deviate too much from their initial estimations where marginalization took place. By formulating the problem in terms of relative transformations, the authors alleviate these effects. Another successful approach, called RSLAM, avoids computation by sticking with a topological representation of the localization problem [78], leaving metric reconstruction aside. By using a continuous, relative representation of the camera trajectory, BA computation becomes largely sparse (see RBA in Ref. [79]), which is especially efficient when closing large loops. In general, V-SLAM for mobile systems is a broad area where engineers ought to set up a task-oriented, hybrid algorithm combining different methods featuring local metric accuracy in realtime and robust loop closing on a topological representation [65,80,81]. It is worth mentioning that filtering methods are not out of the race as they are believed to have a niche in systems with low resources and smaller map size. They can also take part in hybrid algorithms during Euclidean feature initialization or local tracking within the front-end, where by the way their explicit covariances can be of use to improved feature matching as in Ref. [21].

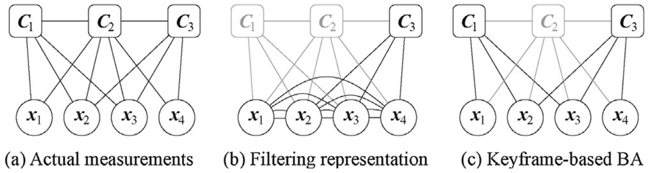


Fig. 3. Filtering approaches (b), motivated by the Markov property, marginalize out past measurements (a) producing artificial correlations; keyframe-based approaches (c) avoid doing so discarding frames with lower information content.

3. Visual pose tracking with the DLR 3D-Modeler

In this section we present novel methods required for visual pose tracking of the DLR 3D-Modeler from its own images, in realtime. By doing this, concurrent 3-D data acquisition and registration is possible without the need for external reference systems, which implies a remarkable improvement in flexibility and cost of the system. Taking the multisensory capabilities of the DLR 3D-Modeler into account, the methods have been specially tailored not to actively affect the scene nor, by implication, other 3-D sensors, cf. Fig. 4. The computational complexity of the algorithms has to be especially low for unrestricted concurrent operation of the other 3-D sensors on the same hardware.

First, in Section 3.1 we motivate the design of our light-weight pose tracking system as in the diagram in Fig. 5. After that, the auxiliary method of feature-based stereo vision is presented in Section 3.2, which increases the efficiency and accuracy of the overall approach and is separate from dense SGM matching (which is still used as a depth sensor, at low rate). Third, we address the front-end of our SLAM system, starting with the efficient tracking of features in real-time in Section 3.3 and the local motion estimation in Section 3.4. Then, appearance-based relocalization is presented in Section 3.5, which enables loop closing as explained in Section 3.6. Last, real-time surface reconstruction out of depth data in the global reference frame, by a textured triangle mesh, will be explained in Section 3.7. Note that the methods in Sections 3.3, 3.4 (in part) and 3.6 have been already published and, therefore, will only be summarized.

3.1. Motivation

Three major requirements have to be satisfied to enable a mobile 3-D modeling device by visual pose tracking as introduced in Section 1: *(1) real-time capability* for the methods to supply

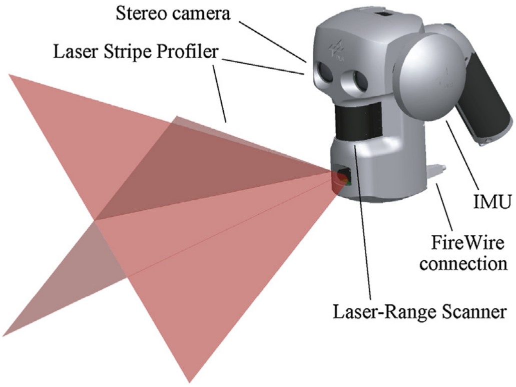


Fig. 4. The DLR 3D-Modeler and its components.

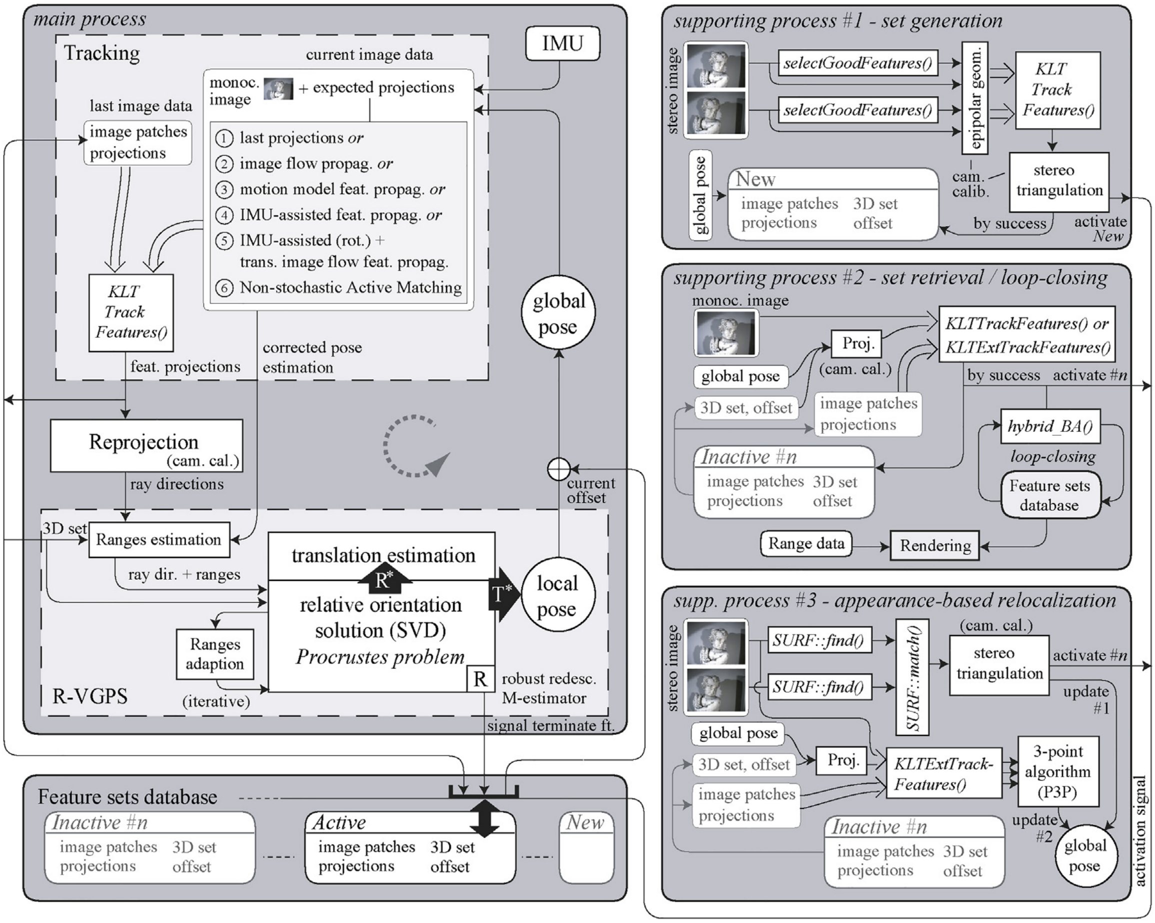


Fig. 5. Block diagram for visual pose tracking using four processes. The feature sets database serves them storing data.

motion estimation, *(2) high positioning accuracy* as required for 3-D modeling (compared to robotic manipulators or tracking systems *plus* their corresponding extrinsic calibrations),[7] and *(3) time-invariant estimations*, i.e., repeated scans of the same area must show constant (high) accuracy irrespective of the scanning time.

In the light of these requirements, three major consequences follow: First, real-time capability implies both, that motion estimations should be regularly performed within 40 ms (25 Hz) *and* that this should hold all the time, i.e., irrespective of the motion history; we support this requirement on efficiency by the choice of a *feature-based approach* on naturally salient, local regions of the images—refer to Section 2.4-I. Furthermore, the requirement on constant efficiency irrespective of motion history merges with the requirement on time-invariant precision, yielding the choice of a *non-filtering approach* for sequential pose tracking as explained in Section 2.4-III. Stochastic filters use system knowledge (e.g., image processing noise or uncertainty in the motion model) to increasing overall precision, which is important. We choose, however, to circumvent this option by using highly accurate 3-D position estimation of features on the scanning area by *feature-based stereo vision*, which enables an accurate non-filtering approach (dense 3-D modeling is left apart for concurrent operation of the other sensors). The hereby achieved efficiency sorts well with the present paradigm of multithreaded, efficient computing.

The above rationale is depicted in Fig. 6 and leads to the development of a feature-based, non-filtering pose tracking algorithm that requires occasional stereo initialization of natural features and monocular tracking of these features over time. Monocular tracking yields the 2-D motion of salient features in the image stream. Since stereo vision already provides the depth of these features, their projected 2-D motion is now solely dependent on perspective projection, i.e., on the 6-DoF camera motion. This is estimated using an especially efficient solution to the relative pose estimation problem: the Visual-GPS method first presented in Ref. [82], refer to Figs. 7 and 12. Lastly, feature initialization, loop closing, and global graph optimization (Section 3.6) are governed by a data management scheme addressed in Section 3.4.2, cf. Fig. 5.

Note our accordance with the latest graph-based optimization paradigm in SLAM of reducing DoF in high-rate pose tracking for better performance, see Section 2.4-III. For instance, PTAM reduced the DoF of the general SLAM problem from 6+3·M (M is the number of features) to 6 in PTAM (for local pose tracking), estimating further DoF (mapping) and the global posegraph in a concurrent

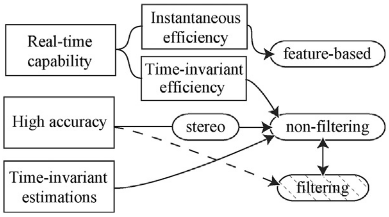


Fig. 6. Requirements (rectangles), implications (arrows), and consequences (rounded rectangles).

[7] Typical accuracies for robotic manipulators are $\sigma_\theta < 0.1^\circ$ and $\sigma_p \approx 0.5$ mm; for IR tracking systems $\sigma_\theta \approx 0.25^\circ$ and $\sigma_p > 0.5$ mm. The accuracy of the IR tracking system in orientation depends on the constellation of markers and is very limited.

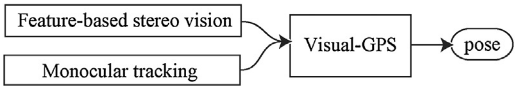


Fig. 7. Feature-based stereo vision and monocular tracking serve Visual-GPS, which in turn yields pose estimations.

thread, at lower rate, from selected keyframes. In our case mapping also relies on keyframes, but substitutes repeated BA by accurate, feature-based stereo vision. The latter is computationally cheaper and contributes absolute scaling—a prerequisite in 3-D modeling. Of course, in the event of loop closures, global structure will still be optimized by graph-based nonlinear optimization techniques, see Section 3.6.

3.2. Accurate structure estimation by stereo vision

The above-mentioned accurate estimation of the sparse scene structure is a grounding pillar of our approach, as it increases both efficiency and the accuracy of local pose tracking. In addition, it anchors the metric scale as required for 3-D modeling, in a passive way (e.g., without the inclusion of artificial landmarks in the scene).

Conventional, sparse feature-based stereo matching relies on computationally expensive Harris affine and DoG feature detectors that deal with affine transformations [29]. In our case of parallel cameras on a short baseline, however, affine distortion can be neglected, which leads to the same assumptions of Shi and Tomasi [28]: *Good features to track* are extracted from the main camera image. Next, a larger number of features are extracted from the second image. Correspondence search is restricted to a few locations within the epipolar band, which is also limited in disparity to obtain useful features on the near scene. Gradient descent optimization yields sub-pixel accurate feature matching; the match with the smallest difference in gradient patches is chosen. Feature triangulation is then performed by linear least squares and subsequently tested for consistency. The expected accuracy levels by stereo vision in our application are detailed in Ref. [1].

Note that it is not a requirement that this feature initialization process be in realtime; we opt for a separate computing thread while concurrently tracking already initialized features in the former thread (in 2-D, monocular) so that local pose tracking is not interrupted. Of course, at the very first initialization step no features are available and 6-DoF pose cannot be delivered in realtime. Here the potential features are monocularly tracked until their stereo correspondences are found and triangulated, which subsequently (seamlessly) bootstraps the regular feature and pose tracking algorithm presented next.

3.3. Efficient monocular tracking of features

Our pose tracking algorithm basically compares an accurately estimated set of 3-D features (result of last section) with their current *monocular* projections—with due regard to correct feature-to-projection correspondences. In order to correctly establish correspondences, two approaches are possible: *global feature tracking* searches for their appearance (e.g., a 2-D descriptor patch) within the whole image, whereas *local, sequential feature tracking* looks for them locally, in particular spots of the image after tracking them ever since their 3-D stereo initialization. In this section we opt for the latter option, which is on the premise that features slightly move in successive images, which holds if the camera motion is moderate.

Sequential feature tracking is a predictive feature search method that exploits probabilistic priors on their expected image projections in order to know where to focus processing resources in each image. These prior distributions ultimately depend on the 3-D location of the features and on the camera motion. Camera motion can be estimated from past measurements and further predicted (e.g., using a motion model). 3-D structure, camera past motion estimation as well as its present motion model may however differ from reality to some extent, translating into “gated” image regions where each feature is expected to lie according to priors, see Fig. 8. The feature tracker seeks feature appearance matches within these bounded regions, hereby delivering temporal image displacements of features—keeping track of correct data association.

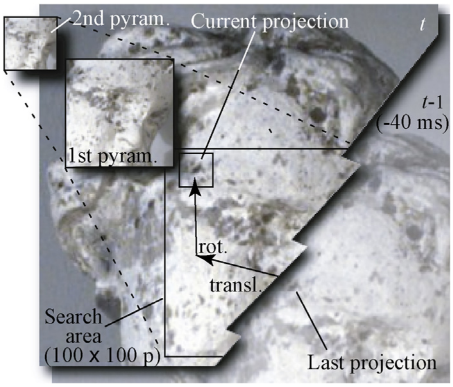


Fig. 8. A KLT feature tracker with big search area due to large expected displacements. Two levels of the pyramidal representation of the image are shown.

Close-range feature tracking is arguably twice as hard as its long-range counterpart. At close range, camera translation and rotation between subsequent camera frames cause feature projection displacements of similar size; at long range, however, the feature projection displacements due to camera translation can be neglected, and only the rotational ones have to be considered (these are independent of the feature distance to the camera). Worst-case interframe camera motions (at 25 Hz) amount to up to 3° and 2 cm (75°/s and 0.5 m/s speeds). At close range, feature displacements may add up to long distances (e.g., search areas of 100 × 100 pixels) that are beyond the real-time capabilities of standard feature trackers.

In Refs. [1,3] we presented an efficient feature tracking algorithm based on the KLT feature tracker [83]. The original implementation was extended in many ways, regarding the efficient processing of local image regions, coarse-to-fine feature matching by a pyramidal implementation, and the use of the latest instructions of modern processors. In addition, several *optical flow prediction schemes* were proposed, implemented, and evaluated. The top-performing method concerns the use of the IMU attached to the DLR 3D-Modeler (accurate calibration and synchronization are needed [84,85]). In detail, the IMU dictates the prediction of the rotational motion of the camera between camera frames, whereas translational motion is extrapolated from the last state estimations, see Fig. 9. The motivation for this approach is the fact that

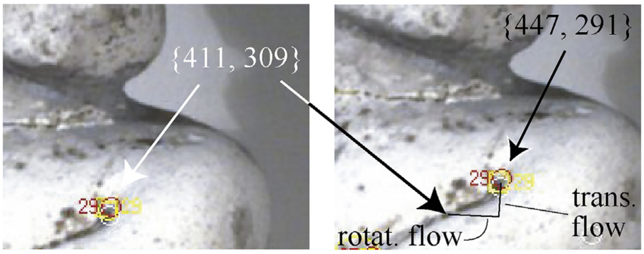


Fig. 9. Feature displacement in two consecutive images. The feature moves from {411, 309} to {447, 291} a distance of 40.2 pixels within 40 ms. 37 pixels are due to rotation, whereas 17 pixels stem from translation; some pixels cancel out.

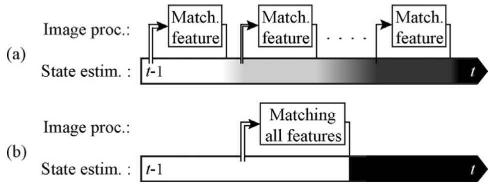


Fig. 10. Traditional methods (b) take priors on feature projections once, where image projections are most uncertain. Active Matching (a) recursively updates the state after single feature matching so that feature projection priors can be more accurately estimated before a matching attempt starts.

translation extrapolation is more accurate than the rotational one due to the mass of the hand-held 3-D modeling device: It is easier for the user to rotate the device with a facile twist movement than to linearly accelerate it. This contribution received the best paper finalist award at the IROS conference in 2009.

In Ref. [4] we presented a more advanced feature tracking algorithm that coped with bigger search areas and, consequently, allowed successful feature tracking in realtime *without the need for an IMU*. The method relies on the Active Matching (AM) paradigm, which follows from the crucial observation that feature matching does not necessarily have to be a monolithic 2-D process, but might as well incur higher level estimations *during* operation, see Refs. [2,21,46] and Fig. 10. In short, AM is putting feature matching *into the loop* of SLAM, performing feature by feature matching search while updating the system state as well as predicting measurement projections after every single feature matching process. The method yields a built-in global consensus for data association, less computation through smaller image processing areas as well as guided search, and, consequently, more accurate estimations.

In detail, we extended the traditional AM method by leveraging the accurate knowledge of the scene structure by feature-based stereo vision explained in Section 3.2. This led to a non-stochastic formulation of AM that is more efficient than the original one. The novel approach allows a very accurate prior rotation estimation from a minimal set of 2 features, see Fig. 11.

3.4. Real-time pose tracking from features flow

Real-time pose tracking of the DLR 3D-Modeler is required both, for online 3-D mesh generation and to support efficient feature tracking in the first place. Real-time estimations can be eventually refined in the case of loop features, refer to Section 3.6. The inputs for real-time pose tracking are the feature tracking results from the last section, together with their accurate 3-D geometry as explained in Section 3.2. Assuming a rigid set of 3-D points and a static camera geometry, the feature projections flow is solely caused by varying perspective projection, i.e., by varying pose of the camera w.r.t. the scene. In this context, pose tracking basically works out valid camera poses that match the measured feature displacements (optical flow), see Fig. 12.

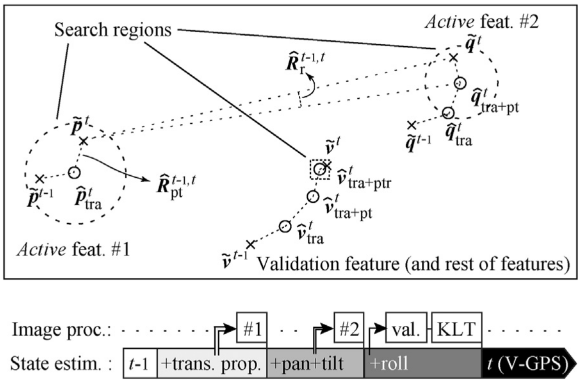


Fig. 11. Top: Pictorial schematic on image projections; two *active* features $\boldsymbol{p}$ and $\boldsymbol{q}$ as well as the subsequent estimation steps on a further feature v are detailed. After exhaustive search of $\boldsymbol{p}$ and $\boldsymbol{q}$, $\boldsymbol{v}$ and the rest of features can be readily tracked in a standard way. Bottom: Evolution of state estimation in time regarding the image processing stages.

3.4.1. The robust visual-GPS

Visual-GPS (V-GPS) is an algorithm that solves the *relative orientation problem* iteratively, but efficiently [82]. After the determination of the orientation, the translation can be also estimated. The method assumes a set of known 3-D points ${}_0\boldsymbol{p}^i$, $\forall\, i \in \mathbb{N}_1,\, i \leq M$ related to the initial camera reference frame S_0, which we obtain by feature-based stereo vision as in Section 3.2. The *exterior orientation problem* for the following *monocular* camera pose S_{T} w.r.t. the reference set of points is now equivalent to the original relative pose estimation problem—provided the correspondences between the points $\boldsymbol{p}_i$ and their projections are known.

In order to solve the exterior orientation problem of S_{T} w.r.t. the set of points ${}_0\boldsymbol{p}^i$, an additional, tentative 3-D model ${}_T\hat{\boldsymbol{p}}_i$ is generated at the current frame S_{T} using both, the current 2-D projections of $\boldsymbol{p}_i$ as well as approximated ranges to that points (from preceding estimations). The problem now reduces to solving the *absolute orientation problem* between these two 3-D sets of points ${}_0\boldsymbol{p}^i$ and ${}_T\hat{\boldsymbol{p}}_i$, which is an approximate, orthogonal Procrustean problem that can be solved in closed form using the singular value decomposition (SVD). As relative translation and rotation are estimated separately, we first set the origins of the sets to their respective centers of mass without modifying their orientations, which yields ${}_0\boldsymbol{p}^{i\prime}$ and ${}_T\hat{\boldsymbol{p}}_i{}'$. The relative rotation between the sets corresponds to the wanted relative rotation between camera reference frames S_0 and S_{T} and can be optimized $(^*)$ by maximizing the trace of the inertia matrix of the matched set:

$$
{}_T\boldsymbol{R}^* = \arg\max_{\boldsymbol{R}} \mathrm{trace}({}_T\boldsymbol{R}^{\mathrm{t}}{}_T\boldsymbol{M}), \quad \boldsymbol{M} = \sum_{i=1}^{M} {}_T\hat{\boldsymbol{p}}_i{}'\, {}_0\boldsymbol{p}_i^{0}{}'{}^{\mathrm{t}}. \tag{1}
$$

Let $(\boldsymbol{U}, \boldsymbol{\sigma}, \boldsymbol{V})$ be the SVD of $\boldsymbol{M}$, that is $\boldsymbol{U}\boldsymbol{\sigma}\boldsymbol{V}^{\mathrm{t}} = \boldsymbol{M}$, then: ${}_T\boldsymbol{R}^* = \boldsymbol{U}\boldsymbol{V}^{\mathrm{t}}$ and the optimal translation ${}_T\boldsymbol{t}^*$ is found by subtracting the center of mass:

$$
{}_T\boldsymbol{t}^* = \frac{1}{M}\sum_{i=1}^{M} {}_T\hat{\boldsymbol{p}}_i - {}_T\boldsymbol{R}^* \frac{1}{M}\sum_{i=1}^{M} {}_0\boldsymbol{p}^i. \tag{2}
$$

Since the tentative 3-D pointset ${}_T\hat{\boldsymbol{p}}_i$ may differ from the actual ${}_0\boldsymbol{p}^i$, the final solution is found iteratively, by concurrently optimizing the ranges of the tentative model. The algorithm normally terminates whenever sufficient consistency with the original set of points ${}_0\boldsymbol{p}^i$ is achieved or, as we choose, by a threshold on absolute orientation correction. The method is sequentially applied to future monocular frames with a sufficient amount of tracked points $\boldsymbol{p}_i$, see Section 3.4.2.

Outliers may occur, either in the generation of the 3-D set of points or in their 2-D monocular tracking. In order to cancel them, we make novel use of a redescending M-estimator on the residual Euclidean distances between matched 3-D points. We use the biweight function of Tukey because of its continuous derivatives and its handy weights. The contribution of each point to the inertia

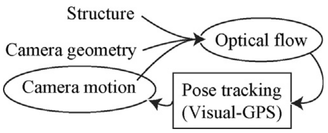


Fig. 12. Structure, camera geometry, and camera motion determine optical flow.

matrix of the matched set of points is now weighted with:

$$\begin{array}{ll} w_i \propto (1 - R_i \cdot R_i)^2 & \text{if } |R_i| < 1 \\ w_i = 0 & \text{if } |R_i| \geq 1 \end{array} \tag{3}$$

where $R_i = ({}_T\boldsymbol{R}\,{}_0\boldsymbol{p}^i - {}_T\hat{\boldsymbol{p}}_i)/s$ is the estimated normalized matching residual for object point i before performing the SVD, and s is the scale of the inlier noise. The robustified inertia matrix

$$\boldsymbol{M}^{\mathrm{R}} = \sum_{i=1}^{M} w_i\, {}_T\hat{\boldsymbol{p}}_i{}'\, {}_0\boldsymbol{p}_i^{\prime\mathrm{t}} \tag{4}$$

substitutes $\boldsymbol{M}$ in Eq. (1). This robustified method (RVGPS) does not only neutralize the effects of outliers, but also signalizes them to be removed from memory to prevent further damage.

3.4.2. Local pose tracking using RVGPS

Following the concept in Section 3.1, for reasons of efficiency we adopt a frugal policy when taking advantage of stereo vision. Consequently, separate sets of 3-D points are used to represent the 3-D structure used for localization. A particular set is triangulated once and then used for local, *monocular* pose tracking thereafter, until a new set of points takes over. It is only when closing loops that we reutilize past sets of points, see Section 3.6. Our approach is similar to VO in Ref. [57], whereas we use AM instead of RANSAC, V-GPS instead of the 3-point algorithm, and more precise feature matching.

Fig. 13 depicts the standard operation of local pose tracking. Note that during handover frames two sets of points are tracked in parallel in order to accumulate the feature displacement information that is needed for successful feature tracking on the most recent set.

Of course, individual feature losses will happen, and features regularly get out of sight and void areas take their place. We treat short- and long-term losses separately: *Short-term losses* are features that are lost by tracking but maintain several fellow points of the 3-D set in track so that camera pose can still be estimated. Monocular tracking will repeatedly try to recover these features with the aid of the current pose—unless RVGPS marked them as invalid. *Long-term losses* are features that are deliberately abandoned because their associated 3-D set of points becomes inadequate to the current vantage point. In this event, either an existing, inactive set of features is retrieved, or a new set of 3-D features is generated.

3.4.3. Local pose tracking using kBA

While RVGPS provides a robust relative motion estimation in realtime, it still seems advisable to perform optimal motion and structure estimation by minimization of reprojection errors (i.e., BA) at handover stereo keyframes to increase dead reckoning accuracy. This is especially so in the case of 3-D modeling where, as a general rule, new areas are explored and loop-closing events are rare. In Section 3.6 we shall present a more complete optimization in the event of final loop closing (e.g., after scanning all around an object).

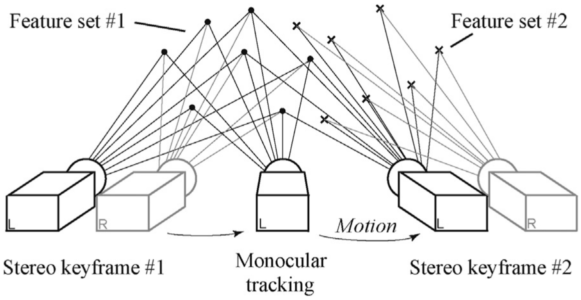


Fig. 13. Local pose tracking: Stereo vision is used in keyframes #1 and #2; monocular tracking elsewhere.

In Ref. [5] we introduced a hybrid kBA formulation in the context of our alternate monocular/stereo pose tracking scheme presented in the last section. The implementation is based on the state-of-the-art approaches in Refs. [61,58,76] but it is extended to use both stereo and monocular frames, which serves to anchor global scale. Since in our approach all 3-D features are measured locally, the global optimization of the covered dead reckoning motion can be exactly decomposed into independent sub-optimizations concerning exclusively one reference frame along with its feature set, which is especially efficient—we achieve 2 to 5 ms with a regular CPU, which is roughly twice as long as RVGPS.

In a nutshell, the novel formulation minimizes the sum of squared reprojection residuals as follows:

$$\begin{aligned} \hat{\boldsymbol{\Omega}}_\star = \arg\ \min \sum_{i=1}^{M} \Big(& \|{}_l\tilde{\boldsymbol{m}}_i - {}_l\hat{\boldsymbol{m}}_i({}_l\hat{\boldsymbol{p}}_i)\|^2 \\ & + \|{}_r\tilde{\boldsymbol{m}}_i - {}_r\hat{\boldsymbol{m}}_i({}_l\boldsymbol{T}^{\mathrm{r}},\, {}_l\hat{\boldsymbol{p}}_i)\|^2 \\ & + \|{}_f\tilde{\boldsymbol{m}}_i - {}_f\hat{\boldsymbol{m}}_i({}_l\hat{\boldsymbol{T}}^{\mathrm{f}},\, {}_l\hat{\boldsymbol{p}}_i)\|^2 \Big) \end{aligned} \tag{5}$$

where the optimized ($\star$) parameters $\boldsymbol{\Omega}_\star$ include the 3-D coordinates ${}_l\boldsymbol{p}_i = [{}_lx_i,\ {}_ly_i,\ {}_lz_i]^{\mathrm{t}}$, $\forall i \in \mathbb{N}_1$, $i \leq M$ of the current set of M features w.r.t. the **l**eft camera at the current keyframe, as well as the inter-keyframe transformation ${}_l\boldsymbol{T}^{\mathrm{f}}$ of the **l**eft camera frame between the current and the upcoming keyframe. The residual is composed of estimated ($\hat{\ }$) reprojections ${}_l\hat{\boldsymbol{m}}_i = \mathrm{proj}({}_l\hat{\boldsymbol{p}}_i)$ and ${}_r\hat{\boldsymbol{m}}_i = \mathrm{proj}({}_r\boldsymbol{T}^{\mathrm{l}}\,{}_l\hat{\boldsymbol{p}}_i)$ onto the **l**eft and the **r**ight frames at the initial keyframe, respectively, as well as their last, **f**inal feature projections ${}_f\hat{\boldsymbol{m}}_i = \mathrm{proj}({}_f\hat{\boldsymbol{T}}^{\mathrm{l}}\,{}_l\hat{\boldsymbol{p}}_i)$ at the left frame, see Fig. 14. These estimations are subtracted from the actual measurements ${}_l\tilde{\boldsymbol{m}}_i$, ${}_r\tilde{\boldsymbol{m}}_i$ and ${}_f\tilde{\boldsymbol{m}}_i$. The transformation ${}_r\boldsymbol{T}^{\mathrm{l}}$ stems from the epipolar geometry of the stereo camera by camera calibration [86].

The hybrid optimization utilizes the nonlinear least squares optimization function `dlevmar_der()` [87], which implements the Levenberg–Marquardt method. We provide all analytic Jacobians for improved performance that include perspective projection, distortion, and rigid-body transformations with differential perturbations of Euler angles for the unknown inter-keyframe transformation ${}_l\boldsymbol{T}^{\mathrm{f}}$, see Ref. [5]. In addition, the residual function has been robustified.

This method yields sub-millimetric corrections w.r.t. RVGPS for every keyframe or feature set. On balance, it seems that this optimal method does not substantially improve the already very accurate dead reckoning motion estimation.

3.5. Appearance-based relocalization

Whenever saccadic motion precludes sequential, seamless tracking, the user browses distant background beyond the reach

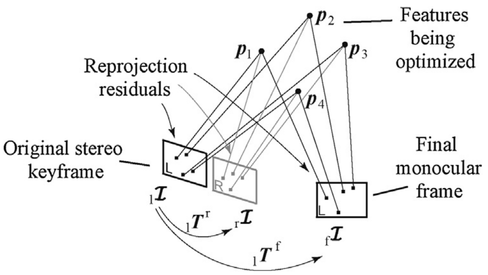


Fig. 14. Local, hybrid bundle adjustment on a particular feature set.

of stereo vision, or the cameras return to a known area that has not been tracked for a long time (loop closing), pose tracking accuracy gets too low for consistent feature tracking to be warranted anymore. Due to the richness of visual data, however, cameras are ideally suited for recognizing similarity to known features; appearance-based relocalization can then help to resume scanning *on the original reference frame*.

As mentioned in Section 2.4-I, there exist operators, called descriptors, that concern about the visual appearance of features, in order to be distinctive between features and invariant to viewpoint pose. In this work, we choose the performant SURF features in its original implementation [32], on stereo images. By using stereo images, the 3-D position of the features w.r.t. the camera ${}_{\text{left}}\boldsymbol{T}^{\text{SURF}}$ can be triangulated *at the same frame* during stereo initialization of the KLT feature set, where we obtained ${}_{\text{left}}\boldsymbol{T}^{\text{KLT}}$, see Section 3.2 and Fig. 15. By doing so, whenever 3 or more of these SURF features (and consequently ${}_{\text{now}}\boldsymbol{T}^{\text{SURF}}$) are found again, the location of the stereo camera w.r.t. the original KLT feature set can be estimated as follows: ${}_{now}\hat{\boldsymbol{T}}^{\text{KLT}} = {}_{\text{now}}\boldsymbol{T}^{\text{SURF}}({}_{\text{left}}\boldsymbol{T}^{\text{SURF}})^{-1}{}_{\text{left}}\boldsymbol{T}^{\text{KLT}}$. This estimation is far less accurate than sequential pose tracking using RVGPS. We opt for using interleaved, *monocular* three point perspective (P3P) pose estimation on the KLT features to increase accuracy [51]. Feature matching is now on extended search regions due to inaccurate SURF-based pose estimation, thus requires exhaustive template matching similar to *active* features in Section 3.3. In the end, regular KLT tracking takes on sequential pose tracking *on the original reference frame*, taking scaling and the affine distortion of the features' templates into account.

3.6. Global graph optimization using kBA on loop closures

Loop closure events occur whenever former scene features are revisited. These events present the opportunity to greatly increase present and past pose tracking accuracy. We distinguish between two types of loop closures: local and global, large-scale. The latter have to be triggered independently from motion estimation precisely because their main goal is to correct inaccurate motion estimation in the first place. Global, large-scale loop closing resorts instead to appearance-based relocalization, see Sections 2.4-I. and 3.5.

In the absence of loop closures, current measurements (projections) only depend on their initial stereo keyframe and on the current relative pose w.r.t. that frame. When closing the loop, however, current projections also depend on the camera motion history ever since their initial stereo keyframe, see Fig. 16.

In Ref. [5] we introduced a novel formulation to optimize all relative poses and points involved in a large-scale loop closure. The formulation concatenates Eq. (5) for the whole skeleton of relative keyframes involved, adding a final loop closure term that relates current projections with the expected projections taking the whole motion history into account:

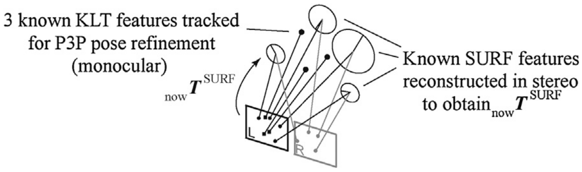

3 known KLT features tracked
for P3P pose refinement
(monocular)
${}_{\text{now}}\boldsymbol{T}^{\text{SURF}}$
Known SURF features
reconstructed in stereo
to obtain ${}_{\text{now}}\boldsymbol{T}^{\text{SURF}}$
L
R

Fig. 15. SURF features are detected in stereo and triangulated to obtain ${}_{now}\boldsymbol{T}^{\text{SURF}}$. This robust estimation is used as an initialization for the more accurate ${}_{now}\boldsymbol{T}^{\text{KLT}}$ estimation using the P3P algorithm; this estimation will eventually support monocular 2-D tracking of known KLT features as in Section 3.3.

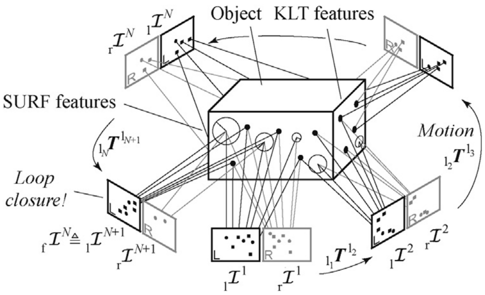

Object
KLT features
${}_{\text{r}}\mathcal{I}^{N}$
${}_{1}\mathcal{I}^{N}$
SURF features
${}_{1_N}\boldsymbol{T}^{1_{N+1}}$
Loop closure!
Motion
${}_{1_2}\boldsymbol{T}^{1_3}$
${}_{\text{f}}\mathcal{I}^{N} \triangleq {}_{1}\mathcal{I}^{N+1}$
${}_{\text{r}}\mathcal{I}^{N+1}$
${}_{1}\mathcal{I}^{1}$
${}_{\text{r}}\mathcal{I}^{1}$
${}_{1_1}\boldsymbol{T}^{1_2}$
${}_{1}\mathcal{I}^{2}$
${}_{\text{r}}\mathcal{I}^{2}$
L
R

Fig. 16. Skeleton of stereo keyframes 1.. *N* when browsing around an object. During monocular tracking of feature set #*N*, feature set #1 can be retrieved at images ${}_{\text{l,r}}I^{N+1}$. Depending on the distance traveled, loop closing occurs either by monocular tracking of KLT features (Section 3.3) or with the help of stereo SURF features (Section 3.5).

$$\begin{aligned}\hat{\boldsymbol{\Omega}}_{\star} = \arg\ \min \sum_{s=c}^{N}\sum_{i=1}^{M_s}\Big(& \|{}_{l}\tilde{\boldsymbol{m}}_i{}^{s} - {}_{l}\hat{\boldsymbol{m}}_i{}^{s}({}_{l}\hat{\boldsymbol{p}}_i{}^{s})\|^2 \\ & + \|{}_{r}\tilde{\boldsymbol{m}}_i{}^{s} - {}_{r}\hat{\boldsymbol{m}}_i{}^{s}({}_{l}\boldsymbol{T}^{\text{r}},\ {}_{l}\hat{\boldsymbol{p}}_i{}^{s})\|^2 + \|{}_{f}\tilde{\boldsymbol{m}}_i{}^{s} - {}_{f}\hat{\boldsymbol{m}}_i{}^{s}({}_{ls}\hat{\boldsymbol{T}}^{1_s}\text{f}_s,\ {}_{l}\hat{\boldsymbol{p}}_i{}^{s})\|^2\Big) \\ & + \sum_{i\in\mathcal{R}} \|{}_{l}\tilde{\boldsymbol{r}}_i{}^{c} - {}_{l}\hat{\boldsymbol{r}}_i{}^{c}({}_{lc}\hat{\boldsymbol{T}}^{1_c}\text{f}_c, \ldots, {}_{lN}\hat{\boldsymbol{T}}^{1_N}\text{f}_N, {}_{l}\hat{\boldsymbol{p}}_i{}^{c})\|^2 \qquad (6)\end{aligned}$$

where the parameters to be optimized $\boldsymbol{\Omega}_{\star} = [\boldsymbol{\Omega}^{c} .. \boldsymbol{\Omega}^{N}]$ include all history of 3-D features between the older feature set # *c* being found again ($\mathcal{R}$ in the subset of actually tracked features), and the last tracked feature set # *N* (i.e., *N*−*c*+1 feature sets in total), as well as the *N*−*c* relative, inter-keyframe transformations between their respectives keyframes and the final local pose. In total, this amounts to $\sum_{s=c}^{N}(3{\cdot}M_s + 6)$ parameters, compared to 3·M_s+6 in Eq. (5).

Again, we are optimizing over differential perturbations of non-privileged, relative transformations to avoid local minima [77]. Consequently, feature locations and camera motions are both locally Euclidean, but globally topological; the global Euclidean representation will be performed at a lower rate, as needed for dense surface reconstruction, see Section 3.7.

In matrix form, the number of equations amounts to $\sum_{s=c}^{N}(2{\cdot}3{\cdot}M_s) + 2{\cdot}\text{size}(\mathcal{R})$, compared to just 2·3·$M_s$ in the case of local kBA for dead reckoning in Eq. (5). Optimization processes with system equations of this magnitude clearly benefit from sparse optimization methods if their Jacobians are sparse. We utilize the nonlinear, least squares sparse optimization function `sparselm_dercrs()` [88], as well as supernodal sparse Cholesky factorization by CHOLMOD [89] and graph partitioning by METIS [90] to observe both primary and secondary sparsity structures of the Jacobian [91]. We provide the full analytic Jacobian in CRS format for improved performance. Common derivative components are being stored instead of recalculated. By way of example, using the sparse variant improves timekeeping from 94 s (standard BA *with* full analytic Jacobian) to between 750 ms and 1.4 s. Not providing analytic Jacobians proves slower by a factor of 2 or 3. Global BA is performed in a separate computing thread not to disrupt concurrent real-time pose tracking and 3-D modeling. In Section 4 we show loop-closing experiments where global BA compensates for substantial dead reckoning errors of several cm to reach consistent topology of the map.

3.7. Streaming surface reconstruction

We implemented a streaming surface reconstruction method that delivers realistic 3-D models *online*, concurrently with range data acquisition and pose tracking, in realtime [92]. Since it is

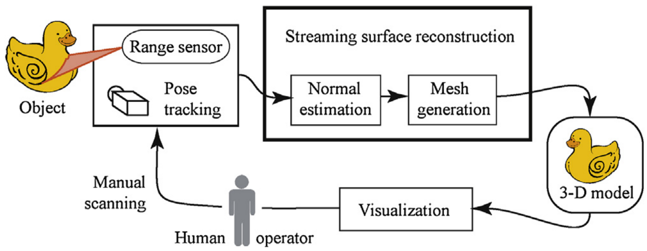


Fig. 17. 3-D modeling pipeline including range and pose data fusion, online surface mesh reconstruction, and 3-D rendering.

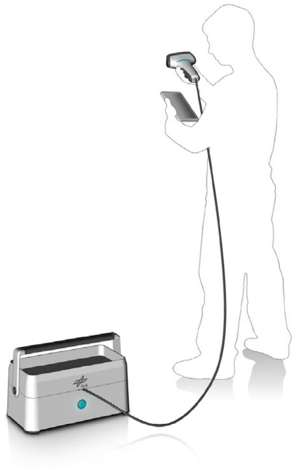


Fig. 18. The DLR 3D-Modeler mobile concept with visual feedback.

online, it serves as a visual feedback for manual scanning by the user, see Figs. 17 and 18 .

In detail, the real-time method iteratively generates a dense and homogeneous triangle mesh in Euclidean space by inserting sample points from data streams and motion readings (e.g., from visual pose tracking). The surface model is refined locally around each new sample. A dynamic spatial data structure using an extendable octree ensures prompt access to growing pointsets as well as continuously updated meshes without restrictions to object size or number of sample points. The generated model can then be accessed at any time (e.g., for visualization or live image stream registration, see Fig. 19).

4. Experimental validation

In this section we first describe in detail the inner operation of the proposed visual pose tracking method on a challenging sequence. Second, the accuracy of the approach is addressed by assessing the consistency of loop closures as well as by predefined motions in concert with a rigidly-attached robotic manipulator that acts as ground truth. Third, the computational efficiency is evaluated. For a descriptive demonstration of the system please refer to former videos in http://goo.gl/PjDeox.

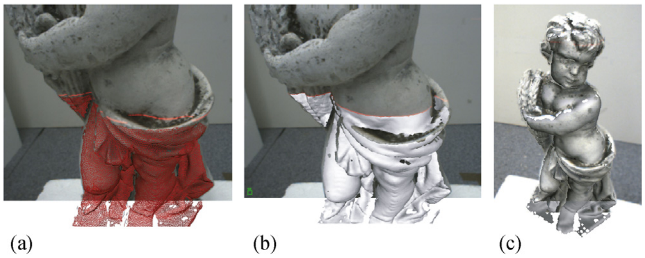


Fig. 19. Online visualization by augmented triangle mesh (a) or surface model (b), leading to a textured 3-D model (c).

4.1. Operation

We illustrate the operation of the proposed methods by following the algorithm's performance on a challenging sequence. The reader can retrieve the processed sequence from http://goo.gl/3n47yj (real time) and http://goo.gl/YC1p4B (slow motion).

The sequence is composed of 625 images acquired at 25 Hz for a period of time of 25 s. The hand-held 3D-Modeler targets a 40 cm tall sculpture at a range of approx. 35 cm, scanning up and down three times. Both the distance to the sculpture and the rough view direction are maintained. During scanning, however, the camera suffers from very strong, saccadic movements, which create an optical flow the size of 40 pixels. The IMU readings state maximal orientation changes of 2.5° and translations of up to 1 cm between images (i.e., 62°/s and 0.25 m/s).

The first feature tracking method presented in Section 3.3 (using the IMU) sequentially localizes the camera w.r.t. eight different sets of points in realtime. The sequence is initialized by a set of 3-D points *Set*#1, which is composed of 25 points and this is also the average number of features in the following sets. Fig. 20(a) shows *Set*#1. The camera moves downwards, see Fig. 20(b), and five further sets of points are initialized, one after another. Then the camera reaches its lowest position and starts moving back to the top. Here the algorithm does not create new sets of points but detects former ones following the policies in Section 3.4.2, see Fig. 20(c), and leaps onto them. Fig. 20(d) traces these changes during the entire sequence; note two additional sets at images number #298 (*Set*#7) and #349 (*Set*#8). In the end, the camera returns to the initial area where the algorithm refers back to *Set*#1.

The behavior defined by the policies in Section 3.4.2 yields successful tracking all the time. It seamlessly leaps from current reference sets onto former ones (local loop closure), which implies bias-free round-scanning, i.e., the positioning accuracy at the end of the sequence equals the accuracy at the beginning.

The second feature tracking method presented in Section 3.3 (non-stochastic active matching) does a similar job *without* the help of an IMU, refer to http://goo.gl/HVnVsr (real time) and http://goo.gl/2rqmeC (slow motion). Fig. 21 displays a typical frame highlighting both active features, the validation set, as well as remaining features.

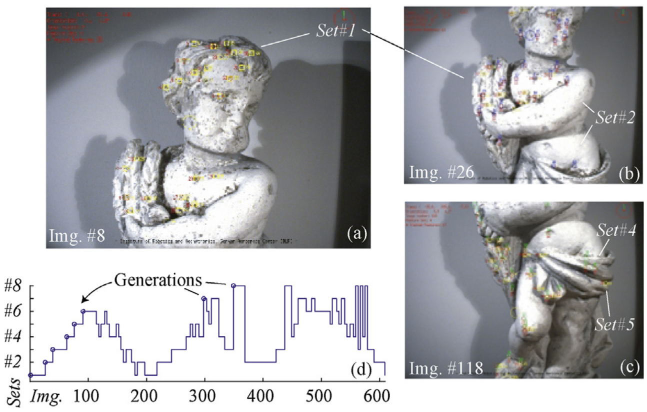


Fig. 20. (a) Image #8 tracking *Set*#1. (b) Image #26 after generation of *Set*#2, changing reference. (c) Image #118 retrieving *Set*#4. (d) Reference sets history in the sequence.

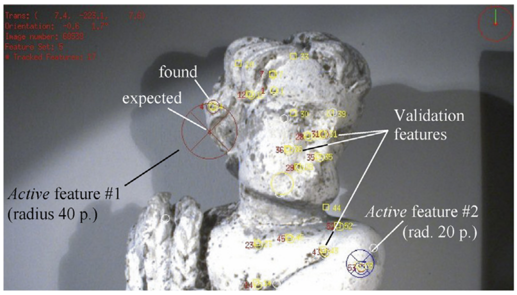


Fig. 21. Image frame including two *active* features, three validation features, and current and past regular features.

4.2. Positioning accuracy

Loop closing is the most natural option for assessing pose tracking accuracy, as pose estimation is possible w.r.t. both, original (say *Set*#1) and present features, immediately after detection of the closure. Subject to the original and the current vantage points w.r.t. the original features (*Set*#1), pose estimation w.r.t. these features is truly very accurate, which virtually acts as ground truth to long-range dead reckoning estimations on the current set. In addition, calibration and synchronization errors w.r.t. an external, ground-truth positioning system are avoided.

Fig. 22 depicts a complete scanning procedure around a 50 cm tall sculpture. A natural browsing procedure asks for prolonged scanning sweeps and is characterized by the absence of loop closure events (neither local nor global). The video at http://goo.gl/tqf4vB shows 4 sweeps featuring a roll angle of 90° between them, a total length of 320 cm and an accumulated rotation of 360°, which certainly bring about dead reckoning errors higher than the tolerated for accurate 3-D modeling. In this event, we close the motion loop as explained in Section 3.6, which corrects current and former pose estimation within a second, and subsequently the whole mesh of the 3-D model as well.

Dead reckoning errors accumulate to an extent that precludes seamless KLT tracking when trying to retrieve the two first feature sets (face and chest) based on the expected camera pose at loop closure. This can be seen in the video by the drift of the white circles corresponding to the initial features. 44 feature sets are initialized by feature-based stereo vision in total. Appearance-based relocalization is triggered in the background on a sensible basis (based on the camera pose and the structure of features). It eventually detects loop closing based on SURF features, but the positioning accuracy is insufficient for KLT tracking (even with Active Matching). It is only by the inclusion of the intermediate stage concerning P3P pose estimation on KLT features with larger search regions (Section 3.5) that we achieve the required pose accuracy for seamless KLT tracking of 55 features pertaining to the feature set #1. Note that, since these computations are triggered in parallel threads, local pose tracking is warranted without interruption. After that, pose refinement by global, hybrid BA as explained in Section 3.6 takes place. After successful local pose refinement by P3P pose estimation, the AM implementation of the extended KLT tracker takes over, cf. Fig. 23. These 55 features in turn trigger the global, hybrid BA process explained in Section 3.6 in a separate computing thread, updating all 43 relative transformations ${}_{l_s}\boldsymbol{T}^{l_s}\mathrm{f}_s$, $\forall s \in \mathbb{N}_1$, $i < 44$ along with the 3-D pose of all 1816 features $\boldsymbol{p}_i^s$, $\forall i \in \mathbb{N}_1$, $i \le M_s$. Using a dated notebook equipped with an Intel® Core™ 2 Duo P8700 processor, the robustified nonlinear optimization takes 870 ms. The parameters vector amounts to 5,769 values and the size of the residuals vector is 11,090.

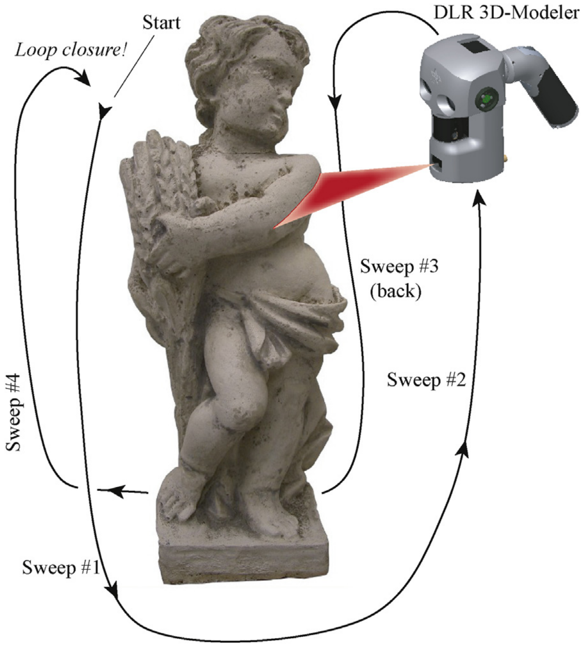


Fig. 22. The hand-guided DLR 3D-Modeler browsing all around the sculpture.

The final pose correction after 320 cm of dead reckoning estimation amounts to 2.5 cm and 6.5°. The appearance-based relocalization stage on SURF features misses the point by 7.5 mm and 1.5°. After KLT relocalization, however, the global localization error is equivalent to local tracking noise (virtually zero). Figs. 24 and 25 show typical corrections of the resulting 3-D pointcloud and mesh after successful loop closure.

Note that the LSP depth sensor is active during the sequence. A second process segments laser stripe projections and subsequently triangulates range data [93]. A third process performs online meshing of 3-D data on live video footage (Section 3.7).

A further experiment has been performed to compare pose tracking accuracy by dead reckoning (RVGPS) with an **external positioning system**: the KR16 KUKA robotic manipulator featuring ~0.1 mm and less than 0.1° accuracy. The DLR 3D-Modeler was attached to the TCP of the manipulator. An extrinsic calibration was

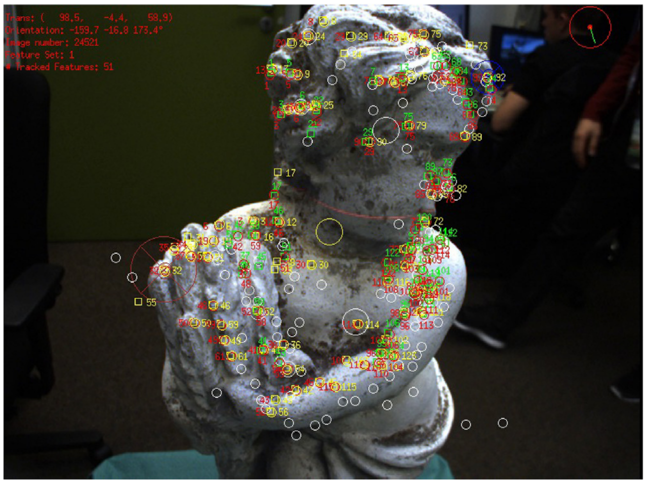


Fig. 23. Parallel tracking of feature set #43 (yellow) and loop-closing set #1 (green) at the loop-closing image frame #24521. Parallel tracking is needed to build up optical flow information. Successful matching is shown in red for both sets. Throughout the whole video, the white features show the two first feature sets as an aid to visualize dead reckoning error drift (note that the feature sets include outliers). **Please find the high-resolution sequence at:http://goo.gl/tqf4vB**.

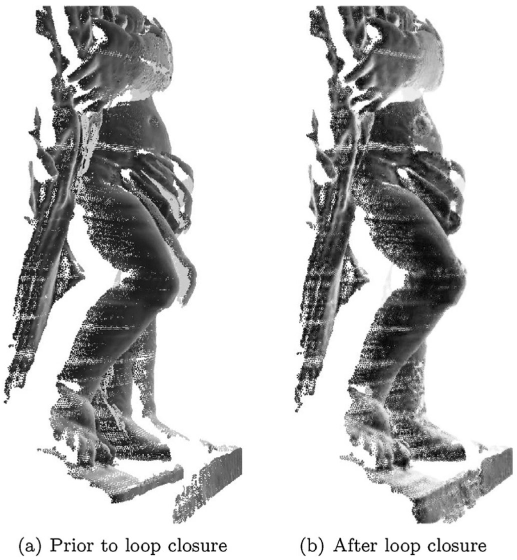
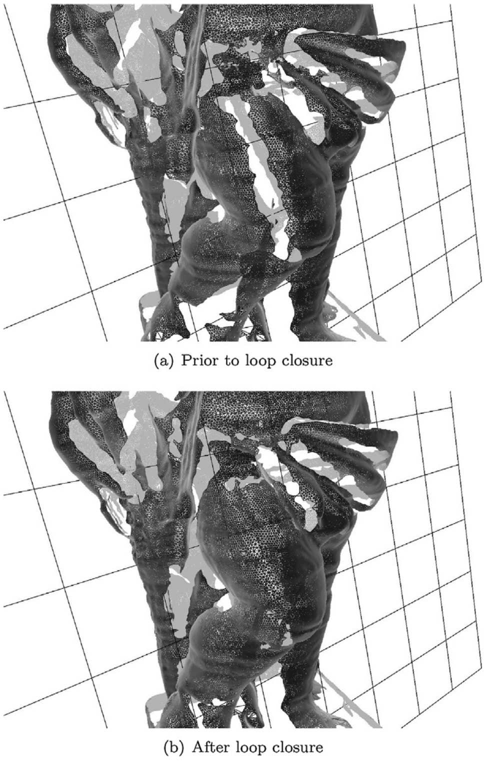

(a) Prior to loop closure (b) After loop closure

Fig. 24. Pointcloud correction after successful loop closure.


(a) Prior to loop closure

(b) After loop closure

Fig. 25. Mesh correction after successful loop closure.

performed [84,86]. As a consequence of potential calibration errors of ground truth data, the accuracy results of this experiment should be considered a worst case.

In detail, a robot motion around an object is performed, total length of 125 cm and 55°, featuring 710 stereo frames. The images are synchronized with the robot's motion [94]. Fig. 26 shows residual errors in translation and rotation. Motion estimation by the original formulation of V-GPS is shown to realize the significance of the robustified variant RVGPS introduced in Section 3.4.1. Pose tracking error by dead reckoning increases up to 3 mm and 0.4° at the turning point; on its way back, the error is removed by retrieval of former sets of points. These results featuring less than 1% distance error match former VO results, cf. Refs. [48,49].

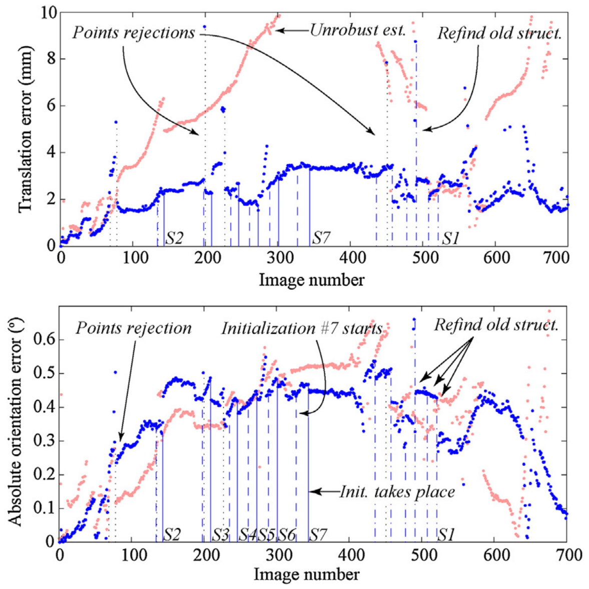


Fig. 26. Translation (upper) and rotation (lower) errors w.r.t. a robotic manipulator using RVGPS (blue) or V-GPS (pink).

4.3. Performance

The typical processing times for visual pose tracking on the DLR 3D-Modeler are listed in Table 1. Note that these computations are in parallel to LSP triangulation [93] and surface reconstruction in Section 3.7.

5. Conclusion

In this work we provide a state-of-the-art overview on static and portable 3-D scanners and describe the algorithms that instantiated the first 3-D modeling device for close-range applications that localizes itself passively from its own images in realtime, at a high data rate. This is an important contribution to increasing the flexibility of these types of devices, by doing without the external positioning systems that constrain existing scanners in terms of size, mobility, and cost, hereby making portable 3-D modeling outdoors possible.

A comprehensive review of 3-D modeling systems points out the lack of devices that are able to passively localize themselves at a high data rate. We implement a visual pose tracking algorithm tailored to 3-D modeling by carefully engineering its key processes: relative motion is delivered at a high data rate from feature tracking on a monocular image stream using a robustified V-GPS algorithm characterized by its efficiency and accuracy; feature tracking is based upon an accelerated KLT feature tracker,

Table 1
Computing times on an Intel Core 2 Duo P8700 processor notebook.

Task	Time (ms)	#feat.
Feature-based stereo triangulation (Section 3.2)	∼300	50
2-D feature tracking using an IMU [1]	∼18	25
2-D feature tracking using AM [4]	12.8 (3+1.2+0.6+8)	50
Robustified V-GPS estimation (Section 3.4.1)	3	50
Local BA (Section 3.4.3)	6	
Appearance-based relocalization (Section 3.5)	∼600	
Global BA (48 stereo keyframes, Section 3.6)	650	2100
Visualization	3	

cast into the Active Matching paradigm for improved performance in close-range (close-range feature tracking is twice as hard as in long range); in order to detach feature set structure estimation from high-rate tracking at the front-end, feature-based stereo vision is frugally triggered at keyframe instants to compute accurate, sparse 3-D geometry and absolute scale; in case of interrupted pose tracking, contingent appearance-based relocalization on known SURF features is provided, together with a rapid pose refinement using a bank of parallel three-point-perspective pose solvers; finally, loop closures are utilized to increase accuracy performing pose-graph optimization in the form of a sparse, keyframe-based bundle adjustment by minimization of the reprojection errors in a hybrid set of stereo and monocular frames. In addition, real-time reconstruction and texturing of the 3-D model's surface provides visual feedback during acquisition. Extended validation experiments with videos are delivered.

References

[1] K.H. Strobl, E. Mair, T. Bodenmüller, S. Kielhöfer, W. Sepp, M. Suppa, D. Burschka, G. Hirzinger, The self-referenced DLR 3D-modeler, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, 2009, pp. 21–28 *best paper finalist*.

[2] A.J. Davison, Active search for real-time vision, Proceedings of the International Conference on Computer Vision (ICCV), Nice, France, 2005, pp. 66–73.

[3] E. Mair, K.H. Strobl, M. Suppa, D. Burschka, Efficient camera-based pose estimation for real-time applications, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, 2009, pp. 2696–2703.

[4] K.H. Strobl, E. Mair, G. Hirzinger, Image-based pose estimation for 3-D modeling in rapid, hand-held motion, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011, pp. 2593–2600.

[5] K.H. Strobl, Loop closing for visual pose tracking during close-range 3-D modeling, ISVC 2014, Part I, Vol. 8887 of Lecture Notes in Computer Science, Springer International Publishing, Switzerland, 2014, pp. 390–401.

[6] K.H. Strobl, A Flexible Approach to Close-Range 3-D Modeling, Dissertation, Institute for Data Processing, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, Munich, Germany (July 2014).

[7] M. Suppa, S. Kielhöfer, J. Langwald, F. Hacker, K.H. Strobl, G. Hirzinger, The 3D-modeller: a multi-purpose vision platform, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy, 2007, pp. 781–787.

[8] S. Meister, S. Izadi, P. Kohli, M. Hämmerle, C. Rother, D. Kondermann, When can we use KinectFusion for ground truth acquisition? Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Color-Depth Camera Fusion in Robotics, Villamoura, Portugal, 2012.

[9] P. Besl, N. McKay, A method for registration of 3-D shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256.

[10] B. Coudrin, M. Devy, J.-J. Orteu, L. Brèthes, An innovative hand-held vision-based digitizing system for 3D modelling, Opt. Lasers Eng. 49 (910) (2011) 1168–1176.

[11] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, A.W. Fitzgibbon, KinectFusion: real-time dense surface mapping and tracking, 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Basel, Switzerland, 2011, pp. 127–136 *best paper award*.

[12] P. Henry, M. Krainin, E. Herbst, X. Ren, D. Fox, RGB-D mapping: using Kinect-style depth cameras for dense 3D modeling of indoor environments, Int. J. Robot. Res. 31 (5) (2012) 647–663.

[13] rc_visard, Roboception GmbH, 2017. http://roboception.com/en/rc_visard-en/.

[14] DAVID-Laserscanner, DAVID Vision Systems GmbH. http://www.david-laserscanner.com.

[15] P. Hébert, A self-referenced hand-held range sensor, 3-D Digital Imaging and Modeling 3DIM, Quebec City, Que., Canada, 2001, pp. 5–12.

[16] R. Khoury, An enhanced positioning algorithm for a self-referencing hand-held 3D Sensor, 3rd Canadian Conference on Computer and Robot Vision CRV, Quebec City, Que., Canada, 2006, pp. 44–50.

[17] J. Harvent, B. Coudrin, L. Brèthes, J.-J. Orteu, M. Devy, Multi-view dense 3D modelling of untextured objects from a moving projector-cameras system, Mach. Vision Appl. 24 (8) (2013) 1645–1659.

[18] M. Pollefeys, L.V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, Visual modeling with a hand-held camera, Int. J. Comput. Vision 59 (3) (2004) 207–232.

[19] G. Roth, A. Whitehead, Using projective vision to find camera positions in an image sequence, Vision Interface VI'2000, Montreal, Canada, 2000, pp. 87–94.

[20] S. Se, P. Jasiobedzki, Stereo-vision based 3D modeling and localization for unmanned vehicles, Int. J. Intell. Control Syst. Spec. Iss. Field Robot. Intell. Syst. 13 (1) (2008) 47–58.

[21] M. Chli, A.J. Davison, Active matching, Proceedings of the European Conference on Computer Vision (ECCV), Marseille, France, 2008, pp. 72–85.

[22] R.A. Newcombe, A.J. Davison, Live dense reconstruction with a single moving camera, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 2010, pp. 1498–1505.

[23] R.A. Newcombe, S. Lovegrove, A.J. Davison, DTAM: dense tracking and mapping in real-time, Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 2320–2327 *best demo award*.

[24] J.M. Brady, Seeds of perception, Proceedings of the Alvey Vision Conference, Cambridge, UK, 1987, pp. 259–265.

[25] P.H.S. Torr, A. Zisserman, Feature based methods for structure and motion estimation, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Workshop on Vision Algorithms, Corfu, Greece, 1999, pp. 278–294.

[26] M. Irani, P. Anandan, All about direct methods, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Workshop on Vision Algorithms, Corfu, Greece, 1999, pp. 267–277.

[27] C. Harris, M. Stephens, A combined corner and edge detector, Proceedings of the Alvey Vision Conference, Manchester, UK, 1988, pp. 147–151.

[28] J. Shi, C. Tomasi, Good features to track, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jerusalem, Israel, 1994, pp. 593–600.

[29] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, W.H. Freeman & Co Ltd., San Francisco, CA, USA, 1982.

[30] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, Proceedings of the British Machine Vision Conference (BMVC), Cardiff, UK, 2002 *best paper prize*.

[31] S.M. Smith, J.M. Brady, SUSAN – a new approach to low level image processing, Int. J. Comput. Vision 23 (1997) 45–78.

[32] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, SURF: speeded up robust features, Comput. Vision Image Understand. 110 (3) (2008) 346–359.

[33] E. Rosten, T. Drummond, Fusing points and lines for high performance tracking, Proceedings of the IEEE International Conference on Computer Vision (ICCV), vol. 2, Nice, France, 2005, pp. 1508–1511.

[34] A.J. Davison, D.W. Murray, Simultaneous localization and map-building using active vision, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 865–880.

[35] D.G. Lowe, Object recognition from local scale-invariant features, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Corfu, Greece, 1999, pp. 1150–1157.

[36] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1615–1630.

[37] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 2005, pp. 886–893.

[38] M. Agrawal, K. Konolige, M.R. Blas, CenSurE: center surround extremas for realtime feature detection and matching, Proceedings of the European Conference on Computer Vision (ECCV), Marseille, France, 2008, pp. 102–115.

[39] M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: binary robust independent elementary features, Proceedings of the European Conference on Computer Vision (ECCV), Crete, Greece, 2010, pp. 778–792.

[40] S. Leutenegger, M. Chli, R. Siegwart, BRISK: binary robust invariant scalable keypoints, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011.

[41] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient Alternative to SIFT or SURF, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011.

[42] A. Alahi, R. Ortiz, P. Vandergheynst, FREAK: fast retina keypoint, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 2012 *open Source Award Winner*.

[43] P.F. Alcantarilla, A. Bartoli, A.J. Davison, KAZE features, Proceedings of the European Conference on Computer Vision (ECCV), Firenze, Italy, 2012.

[44] T. Schmidt, R. Newcombe, D. Fox, Self-supervised visual descriptor learning for dense correspondence, IEEE Robot. Autom. Lett. 2 (2) (2017) 420–427.

[45] J. Neira, J.D. Tardós, Data association in stochastic mapping using the joint compatibility test, IEEE Trans. Robot. Autom. 17 (6) (2001) 890–897.

[46] M. Chli, A.J. Davison, Active matching for visual tracking, Robot. Auton. Syst. 57 (12) (2009) 1173–1187.

[47] A. Comport, M. Meilland, P. Rives, An asymmetric real-time dense visual localisation and mapping system, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1st International Workshop on Live Dense Reconstruction from Moving Cameras, Barcelona, Spain, 2011.

[48] D. Nistér, O. Naroditsky, J.R. Bergen, Visual Odometry, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 2004, pp. 652–659.

[49] Y. Cheng, M.W. Maimone, L. Matthies, Visual odometry on the mars exploration rovers, IEEE Robot. Autom. Mag. 3 (2) (2006) 54–62.

[50] K. Konolige, M. Agrawal, J. Solà, Large scale visual odometry for rough terrain, Proceedings of the International Symposium on Research in Robotics (ISRR), Hiroshima, Japan, 2007.

[51] J.A. Grunert, Das Pothenotische Problem in erweiterter Gestalt; nebst Bemerkungen über seine Anwendungen in der Geodäsie, Grunerts Archiv für Mathematik und Physik 1 (1841) 238–248.

[52] W. Wolfe, D. Mathis, C.W. Sklair, M. Magee, Three perspective view of three points, IEEE Trans. Pattern Anal. Mach. Intell. 13 (1) (1991) 66–73.

[53] R.M. Haralick, C.-N. Lee, K. Ottenberg, M. Nölle, Review and analysis of solutions of the three point perspective pose estimation problem, Int. J. Comput. Vision 13 (3) (1994) 331–356.

[54] D. Nistér, A minimal solution to the generalised 3-point pose problem, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 2004, pp. 560–567.

[55] D. Nistér, An efficient solution to the five-point relative pose problem, IEEE Trans. Pattern Anal. Mach. Intell. 26 (6) (2004) 756–770.

[56] H. Stewénius, C. Engels, D. Nistér, Recent developments on direct relative orientation, ISPRS J. Photogram. Rem. Sens. 60 (2006) 284–294.

[57] D. Nistér, O. Naroditsky, J. Bergen, Visual odometry for ground vehicle applications, J. Field Robot. 23 (2006).

[58] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, Real time localization and 3D Reconstruction, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 2006, pp. 363–370.

[59] D. Nistér, Preemptive RANSAC for live structure and motion estimation, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Nice, France, 2003, pp. 199–206.

[60] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, Generic and real-time structure from motion using local bundle adjustment, Image Vision Comput. 27 (8) (2009) 1178–1193.

[61] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, Proceedings of the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan, 2007.

[62] K. Konolige, M. Agrawal, FrameSLAM: from bundle adjustment to real-time visual mapping, IEEE Trans. Robot. 24 (5) (2008) 1066–1077.

[63] C. Mei, G. Sibley, M. Cummins, P. Newman, I. Reid, A constant time efficient stereo SLAM system, Proceedings of the British Machine Vision Conference (BMVC), London, UK, 2009.

[64] B. Williams, I. Reid, On combining visual SLAM and visual odometry, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, USA, 2010.

[65] H. Strasdat, A.J. Davison, J.M.M. Montiel, K. Konolige, Double window optimisation for constant time visual SLAM, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 2352–2359.

[66] A.J. Davison, Mobile Robot Navigation Using Active Vision, PhD thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, UK (October 1999).

[67] S.J. Julier, J.K. Uhlmann, A counter example to the theory of simultaneous localization and map building, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea, 2001, pp. 4238–4243.

[68] J.J. Leonard, H.J.S. Feder, A computationally efficient method for large-scale concurrent mapping and localization, International Symposium of Robotics Research, Snowbird, UT, USA, 2000, pp. 316–321.

[69] J. Guivant, E. Nebot, Optimization of the simultaneous localization and map building algorithm for real time implementation, IEEE Trans. Robot. Autom. 17 (3) (2001) 242–257.

[70] E. Eade, T. Drummond, Monocular SLAM as a graph of coalesced observations, Proceedings of the International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, 2007.

[71] G. Qian, R. Chellappa, Structure from motion using sequential Monte Carlo methods, Int. J. Comput. Vision 59 (1) (2004) 5–31.

[72] R. Sim, J. Little, Autonomous vision-based exploration and mapping using hybrid maps and Rao-Blackwellised particle filters, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China, 2006, pp. 2082–2089.

[73] M. Montemerlo, S. Thrun, FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics, Series: Springer Tracts in Advanced Robotics, vol. 27, Springer Berlin Heidelberg, 2007.

[74] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle adjustment – a modern synthesis, Proceedings of the International Conference on Computer Vision (ICCV), Kerkyra, Greece, 1999.

[75] F. Lu, E. Milios, Globally consistent range scan alignment for environment mapping, Autonom. Robots 4 (4) (1997) 333–349.

[76] H. Strasdat, J.M.M. Montiel, A.J. Davison, Real-time monocular SLAM: why filter? Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 2010, pp. 2657–2664 *best vision paper award*.

[77] H. Strasdat, J.M.M. Montiel, A. Davison, Scale drift-aware large scale monocular SLAM, Proceedings of Robotics: Science and Systems, Zaragoza, Spain, 2010.

[78] C. Mei, G. Sibley, M. Cummins, P. Newman, I. Reid, RSLAM: a system for large-scale mapping in constant-time using stereo, Int. J. Comput. Vision 94 (2010) 198–214 special issue of BMVC.

[79] G. Sibley, C. Mei, I. Reid, P. Newman, Adaptive relative bundle adjustment, Proceedings of Robotics: Science and Systems, Seattle, USA, 2009.

[80] B. Clipp, J. Lim, J.-M. Frahm, M. Pollefeys, Parallel, Real-time visual SLAM, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 2010, pp. 3961–3968.

[81] J. Lim, J.-M. Frahm, M. Pollefeys, Online environment mapping, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 2011, pp. 3489–3496.

[82] D. Burschka, G.D. Hager, V-GPS – image-based control for 3D guidance systems, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2003, pp. 1789–1795.

[83] C. Tomasi, T. Kanade, Detection and Tracking of Point Features, Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, 1991.

[84] K.H. Strobl, G. Hirzinger, Optimal hand-eye calibration, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China, 2006, pp. 4647–4653.

[85] M. Fleps, E. Mair, O. Ruepp, M. Suppa, D. Burschka, Optimization Based IMU camera calibration, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 2011, pp. 3297–3304.

[86] K.H. Strobl, W. Sepp, S. Fuchs, C. Paredes, M. Smíšek, K. Arbter, DLR CalDe and DLR CalLab, (2005) . http://www.robotic.dlr.de/callab/.

[87] M.I.A. Lourakis, levmar: Levenberg–Marquardt Nonlinear Least Squares Algorithms in C/C++, (2004) . http://www.ics.forth.gr/lourakis/levmar/.

[88] M.I.A. Lourakis, Sparse non-linear least squares optimization for geometric vision, Proceedings of the European Conference on Computer Vision (ECCV), vol. 2, Crete, Greece, 2010, pp. 43–56.

[89] Y. Chen, T.A. Davis, W.W. Hager, S. Rajamanickam, Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate, ACM Trans. Math. Softw. 35 (3) (2008) 22:1–22:14.

[90] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1) (1999) 359–392.

[91] K. Konolige, Sparse bundle adjustment, Proceedings of the British Machine Vision Conference (BMVC), Aberystwyth, Wales, 2010.

[92] T. Bodenmüller, Streaming Surface Reconstruction from Real Time 3D Measurements (Ph.D. thesis), Institute for Real-Time Computer Systems, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, Munich, Germany, 2009.

[93] K.H. Strobl, W. Sepp, E. Wahl, T. Bodenmüller, M. Suppa, J.F. Seara, G. Hirzinger, The DLR multisensory hand-guided device: the laser stripe profiler, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, USA, 2004, pp. 1927–1932.

[94] T. Bodenmüller, W. Sepp, M. Suppa, G. Hirzinger, Tackling multisensory 3D data acquisition and fusion, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA, 2007, pp. 2180–2185.