Loop Closing for Visual Pose Tracking during Close-Range 3-D Modeling

Klaus H. Strobl

Robotics and Mechatronics Center (RMC)
German Aerospace Center (DLR)
D-82234 Wessling, Germany

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 real-time, concurrently with customary 3-D modeling of the scene by laser triangulation. Our former works in Refs. [1,2] successfully implemented visual pose tracking. Accuracy being a central requirement to 3-D modeling, however, here we note that accuracy can be further increased using a graph-based nonlinear optimization of the tracked pose by minimization of reprojection errors. Loop closures e.g. when having scanned all around the objects provide the opportunity to increase pose tracking and 3-D modeling accuracy. The sparse optimization is in the form of a hybrid, keyframe-based bundle adjustment algorithm on stereo keyframes, yielding rapid optimization of the whole trajectory and object mesh model within a second. The optimization is supported by the use of appearance-based SURF descriptors together with a bank of parallel three-point-perspective pose solvers.

1 Introduction

Close-range 3-D modeling is a field that, in our view, is going to rapidly spread in novel areas like human-computer interaction and robotics due to the advent of cheaper and lighter range sensors. It is believed that it is through the explicit formation of 3-D models that a considerable number of the remaining challenges of visual perception will be eventually solved. This is, of course, subject to the performance, cost, and flexibility of application of the 3-D modeling device.

It is often impossible to acquire a complete 3-D model in a single measurement step owing to e.g. object self-occlusion, object size, or limited field of view; this is especially true in close-range. Multiple views (or multiple sensors) are regularly deployed in order to fuse their range images into a registered 3-D model. The prevalent approach is to measure the pose of the sensors while acquiring data, which allows for online registration of data in absolute coordinates. A range of pose tracking systems, robotic manipulators, turntables, or electromagnetic devices are commonly deployed for this purpose. These options are inconvenient for three reasons: First, they limit the mobility of the user; second, they require accurate calibration and synchronization w.r.t. the range sensor; and third, they are (by far) the largest and most expensive part of the 3-D modeling system.

G. Bebis et al. (Eds.): ISVC 2014, Part I, LNCS 8887, pp. 390–401, 2014.

In Refs. [1,2] we presented 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 Ref. [1] pose tracking was optionally supported by an on-board IMU for more efficient feature tracking. In Ref. [2] we present an alternative tracking method that, inspired by the Active Matching paradigm, achieves remarkable tracking resilience without the need for inertial readings.

3-D modeling by browsing an object is largely an exploratory act where loop closing events are rare. Hence real-time pose tracking largely relies on dead reckoning, which is inconvenient as it invariably accumulates drifts. The ultimate goal of 3-D modeling is, however, the complete reconstruction of objects e.g. by scanning all around the object. This event involves (at least) one loop closure that provides the opportunity to greatly increase present and past pose tracking accuracy so that the final 3-D model will excel in accuracy irrespective of prior motion drifts. It is worth noting that visual odometry is still perfectly useful during the browsing period; first, in order to provide live image augmentation and live meshing results; second, in order to support rapid, local loop closing.

In this work we use the DLR 3D-Modeler platform, which is a low-cost, hand-held device for geometric and radiometric reconstruction of close-range objects in realtime. It excels in accuracy compared with e.g. depth sensors based on coded infrared light [3]. In detail, we extend our prior approach presented in Ref. [1], aiming at more accurate pose tracking by graph-based nonlinear optimization of the tracked pose by minimization of residual reprojection errors. We opt for a *hybrid*, keyframe-based bundle adjustment (kBA) algorithm on *stereo* keyframes, because kBA is allegedly the most accurate and efficient option to tackle this problem in the face of a higher number of features and keyframes [4].

2 State of the Art

A straightforward option to register range images is based on their geometry. Depending on the acquired scene, however, this option may be precluded if the surfaces do not show salient 3-D regions, or in the case of 1-D range data e.g. when using slit scanners. A widespread alternative for depth image registration in realtime is to externally track the pose of the modeling device so that range data can be directly represented in a common reference frame, in realtime and irrespective of the range data quality. In Ref. [1] the authors listed the dominant commercial 3-D modeling systems, which either use inconvenient external reference systems, or opt for visual pose tracking relying on active illumination and adhesive markers on the scene. They also listed research work on *passive* visual tracking, which did not, however, run in realtime.

Dense methods (either based on GPGPU or on RGB-D hardware) are presently the preferred choice for inexpensive 3-D modeling. We believe, however, that it is still convenient to follow the method presented in Ref. [1] because *first*, slit scanners still provide higher accuracy [3], *second*, its lower hardware and energy requirements (a GPU is not needed), and *third*, feature-based pose tracking methods provide a higher degree of viewpoint invariance than dense methods.

3 The DLR 3D-Modeler

The DLR 3D-Modeler is a multi-purpose platform for geometric and visual perception. It combines complementary sensors in a compact, generic way. It is low weight and features on-board computation, a FireWire connector, generic mechanical interfaces, and an extensive software suite. Current applications comprise 3-D modeling, visual tracking and servoing, exploration, path planning, and object recognition e.g. as the perception head of the humanoid robot Justin.

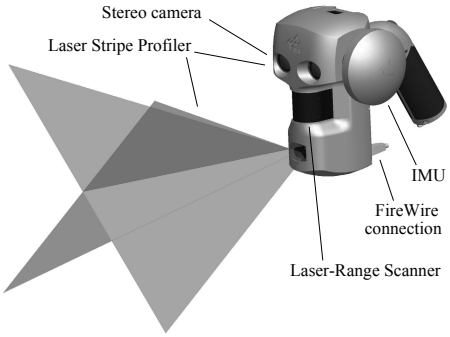

Fig. 1. The DLR 3D-Modeler

4 Visual Pose Tracking at the DLR 3D-Modeler

In the original work in Ref. [1] we provide pose tracking estimations in realtime out of monocular tracking of salient features in the context of an extended KLT feature tracker; the features have been accurately reconstructed in 3-D by one-time stereo vision. The major challenge in this context is tracking features in close-range; unlike in the case of far-range tracking, close-range feature tracking is affected by translation to a similar extent as by rotation—especially in the case of a hand-guided device prone to jerky motion. A novel method was proposed that leverages the rotational readings of an IMU for improved estimation of the displacement of features in between frames. After that in Ref. [2] we presented an alternative method that did without IMU readings by casting the KLT feature tracker unto the Active Matching (AM) paradigm, achieving robust feature tracking at an even higher motion bandwidth. Relative pose estimation is based on an efficient, robustified V-GPS algorithm [5]. In this work we contribute a novel graph-based optimization procedure as well as appearance-based relocalization and loop-closing to eventually boost accuracy.

4.1 Local, Keyframe-Based Bundle Adjustment

It is a peculiarity of 3-D modeling that new areas are continuously being explored and loop-closing events are rare. In this section we focus on optimal motion estimation *without* closing large loops, *i.e.*, by dead reckoning; in Sect. 4.3 we shall present a more complete optimization in the event of a final loop closure e.g. after scanning all around an object.

While robust V-GPS provides a robust, fast motion estimation from monocular footage by dead reckoning in realtime, it is still advisable to perform optimal motion and structure estimation by minimization of reprojection errors at handover stereo keyframes to further increase accuracy. Following e.g. Refs. [6,4]

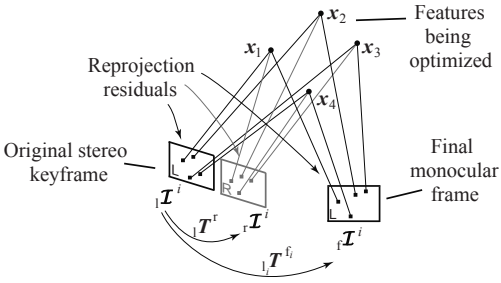

Fig. 2. Data concerned in local, hybrid BA on feature set $\#i$

we opt for an efficient BA optimization disregarding image frames in between selected keyframes (*i.e.*, kBA). Since in our approach all 3-D features are being measured locally, *i.e.*, on a unique static reference frame defined at keyframes, the global optimization of the covered dead reckoning motion can be decomposed into independent sub-optimizations exclusively concerning one reference frame along with its feature set. In detail, the information required for every sub-optimization is confined to the stereo (**l**eft and **r**ight) keyframe ${}_{\mathrm{l}}\mathcal{I}^i \cup {}_{\mathrm{r}}\mathcal{I}^i$ that initialized the ith feature set by stereo triangulation, along with the **f**inal, left monocular image ${}_{\mathrm{f}}\mathcal{I}^i$ that both, tracks the ith feature set last, and coincides with the left frame ${}_{\mathrm{l}}\mathcal{I}^{i+1}$ of the next keyframe (from which the following feature set $\#i{+}1$ will be initialized), cf. Fig. 2. This frugal, hybrid keyframe selection policy does deliver high accuracy as both, initial and last tracking vantage points, are being considered for every feature, maximizing their projected parallax. In addition, the inclusion of stereo images serves to anchor global scale. The novel formulation minimizes the sum of squared reprojection residuals as follows:

$$\widehat{\boldsymbol{\Omega}}^i_\star = \arg\min \sum_{p=1}^{M_i} \Big(\big|\big|{}_{\mathrm{l}}\tilde{\boldsymbol{m}}^i_p - {}_{\mathrm{l}}\hat{\boldsymbol{m}}^i_p({}_{\mathrm{l}}\hat{\boldsymbol{x}}^i_p)\big|\big|^2 + \big|\big|{}_{\mathrm{r}}\tilde{\boldsymbol{m}}^i_p - {}_{\mathrm{r}}\hat{\boldsymbol{m}}^i_p({}_{\mathrm{l}}\boldsymbol{T}^{\mathrm{r}}, {}_{\mathrm{l}}\hat{\boldsymbol{x}}^i_p)\big|\big|^2 + \big|\big|{}_{\mathrm{f}}\tilde{\boldsymbol{m}}^i_p - {}_{\mathrm{f}}\hat{\boldsymbol{m}}^i_p({}_{\mathrm{l}_i}\widehat{\boldsymbol{T}}^{\mathrm{f}_i}, {}_{\mathrm{l}}\hat{\boldsymbol{x}}^i_p)\big|\big|^2 \Big) \tag{1}$$

where the optimized ($\star$) parameters $\boldsymbol{\Omega}^i_\star$ include the 3-D coordinates ${}_{\mathrm{l}}\boldsymbol{x}^i_p = [{}_{\mathrm{l}}x^i_p, {}_{\mathrm{l}}y^i_p, {}_{\mathrm{l}}z^i_p]^\mathsf{T}$, $\forall p{\in}\mathbb{N}_1$, $i \leq M_i$ of the ith set of M_i features w.r.t. the **l**eft camera at keyframe i, and the inter-keyframe transformation ${}_{\mathrm{l}_i}\boldsymbol{T}^{\mathrm{f}_i}$ of the **l**eft camera frame between keyframes $\#i$ and $\#i{+}1$. The residual is composed of estimated ($\hat{}$) reprojections ${}_{\mathrm{l}}\hat{\boldsymbol{m}}^i_p = [{}_{\mathrm{l}}\hat{u}^i_p\ {}_{\mathrm{l}}\hat{v}^i_p]^\mathsf{T} = \mathrm{proj}({}_{\mathrm{l}}\hat{\boldsymbol{x}}^i_p)$ and ${}_{\mathrm{r}}\hat{\boldsymbol{m}}^i_p = [{}_{\mathrm{r}}\hat{u}^i_p\ {}_{\mathrm{r}}\hat{v}^i_p]^\mathsf{T} = \mathrm{proj}({}_{\mathrm{r}}\boldsymbol{T}^{\mathrm{l}}{}_{\mathrm{l}}\hat{\boldsymbol{x}}^i_p)$ onto the **l**eft and the **r**ight frames at the initial keyframe of feature set $\#i$, respectively, as well as their last, **f**inal feature projections ${}_{\mathrm{f}}\hat{\boldsymbol{m}}^i_p = [{}_{\mathrm{f}}\hat{u}^i_p\ {}_{\mathrm{f}}\hat{v}^i_p]^\mathsf{T} = \mathrm{proj}({}_{\mathrm{f}_i}\widehat{\boldsymbol{T}}^{\mathrm{l}_i}{}_{\mathrm{l}}\hat{\boldsymbol{x}}^i_p)$ at the left frame (remember that ${}_{\mathrm{f}}\mathcal{I}^i \triangleq {}_{\mathrm{l}}\mathcal{I}^{i+1}$). These estimations are being subtracted from the actual measurements ${}_{\mathrm{l}}\tilde{\boldsymbol{m}}^i_p$, ${}_{\mathrm{r}}\tilde{\boldsymbol{m}}^i_p$ and ${}_{\mathrm{f}}\tilde{\boldsymbol{m}}^i_p$ to form the residual reprojection errors. The transformation ${}_{\mathrm{r}}\boldsymbol{T}^{\mathrm{l}}$ stems from the epipolar geometry of the stereo camera by camera calibration. Note that the projection function proj() does not include lens distortion; for efficiency reasons, we opt for minimizing undistorted reprojection errors, undistorting actual projections beforehand.

Note that global scale could also be anchored even if the projections ${}_{\mathrm{r}}\hat{\boldsymbol{m}}_p^i$ had not been included in the residual function, but considered ground truth instead. However, we stress that the nature of the 3-D features $\boldsymbol{x}_p^i$ does not stem from selected, ground truth projections into an image (e.g. ${}_{\mathrm{r}}\tilde{\boldsymbol{m}}_p^i$) in the context of stereo vision, but from their rigid body geometry alone. In this way, by releasing all three key projections the optimal 3-D solution will be solely constrained by the rigid body assumption together with perspective geometry. In addition, it is well known that full BA turns out to be faster than any attempts to eliminate e.g. the structure parameters [7].

The hybrid optimization utilizes the nonlinear least squares optimization function `dlevmar_der()` [8], which implements the Levenberg-Marquardt method. We are providing analytic Jacobians for improved performance. Even though they are always sparse, the small size of the system of equations renders sparse methods unnecessary. It is worth noting that minimal representations are used for unknown rotations, specifically differential perturbations of Euler angles. In addition, the residual function has been robustified in case of outliers.

$$
\begin{aligned}
\frac{\partial(\tilde{\boldsymbol{m}}^i - \hat{\boldsymbol{m}}^i)}{\partial \boldsymbol{\Omega}^i} &= \begin{bmatrix} \vdots \\ \dfrac{\partial \boldsymbol{\Delta} \boldsymbol{m}_p^i}{\partial \boldsymbol{\Omega}^i} \\ \vdots \end{bmatrix}_{6M_i \times (6+3M_i)} = \begin{bmatrix} \vdots \\ \dfrac{\partial \boldsymbol{\Delta}_{\mathrm{l}} \boldsymbol{m}_p^i}{\partial \boldsymbol{\Omega}^i} \\ \dfrac{\partial \boldsymbol{\Delta}_{\mathrm{r}} \boldsymbol{m}_p^i}{\partial \boldsymbol{\Omega}^i} \\ \dfrac{\partial \boldsymbol{\Delta}_{\mathrm{f}} \boldsymbol{m}_p^i}{\partial \boldsymbol{\Omega}^i} \\ \vdots \end{bmatrix} = \\
&= \begin{array}{c} \\ {}_{\mathrm{l}}\boldsymbol{m}_p^i \to \\ {}_{\mathrm{r}}\boldsymbol{m}_p^i \to \\ {}_{\mathrm{f}}\boldsymbol{m}_p^i \to \end{array} \begin{array}{c} \overbrace{\phantom{\boldsymbol{\Delta}\boldsymbol{T}_{\mathrm{l}_{i+1}}}}^{\boldsymbol{\Delta}\boldsymbol{T}_{\mathrm{l}_{i+1}}} \quad \overbrace{\phantom{\boldsymbol{x}_1^i..\boldsymbol{x}_{p-1}^i}}^{\boldsymbol{x}_1^i..\boldsymbol{x}_{p-1}^i} \quad \overbrace{\phantom{\boldsymbol{x}_p^i}}^{\boldsymbol{x}_p^i} \quad \overbrace{\phantom{\boldsymbol{x}_{p+1}^i..\boldsymbol{x}_{M_i}^i}}^{\boldsymbol{x}_{p+1}^i..\boldsymbol{x}_{M_i}^i} \\ \begin{bmatrix} & & \vdots & \\ \mathbf{0}_{2\times 6} & \mathbf{0}_{2\times 3(p-1)} & \dfrac{\partial \boldsymbol{\Delta}_{\mathrm{l}} \boldsymbol{m}_p^i}{\partial \boldsymbol{x}_p^i} & \mathbf{0}_{2\times 3(M_i-p)} \\ \mathbf{0}_{2\times 6} & \mathbf{0}_{2\times 3(p-1)} & \dfrac{\partial \boldsymbol{\Delta}_{\mathrm{r}} \boldsymbol{m}_p^i}{\partial \boldsymbol{x}_p^i} & \mathbf{0}_{2\times 3(M_i-p)} \\ \dfrac{\partial \boldsymbol{\Delta}_{\mathrm{f}} \boldsymbol{m}_p^i}{\partial \boldsymbol{T}} & \mathbf{0}_{2\times 3(p-1)} & \dfrac{\partial \boldsymbol{\Delta}_{\mathrm{f}} \boldsymbol{m}_p^i}{\partial \boldsymbol{x}_p^i} & \mathbf{0}_{2\times 3(M_i-p)} \\ & & \vdots & \end{bmatrix} \end{array} = \\
&= \begin{bmatrix} & & \vdots & \\ \mathbf{0}_{2\times 6} & \mathbf{0}_{2\times 3(p-1)} & \blacksquare\square\blacksquare & \mathbf{0}_{2\times 3(M_i-p)} \\ \mathbf{0}_{2\times 6} & \mathbf{0}_{2\times 3(p-1)} & \square\blacksquare\blacksquare & \mathbf{0}_{2\times 3(M_i-p)} \\ \blacksquare\blacksquare\blacksquare\square\blacksquare\blacksquare & \blacksquare\blacksquare\blacksquare & \blacksquare\blacksquare\blacksquare & \blacksquare\blacksquare\blacksquare \\ \blacksquare\blacksquare\blacksquare\square\blacksquare\blacksquare & \blacksquare\blacksquare\blacksquare & \blacksquare\blacksquare\blacksquare & \blacksquare\blacksquare\blacksquare \\ & & \vdots & \end{bmatrix}
\end{aligned}
\tag{2}
$$

∗ White boxes correspond to zero elements; gray or black boxes to non-zero ones.

By way of illustration, we go into detail about the calculation of the **black** Jacobian element above:

$$\frac{\partial({}_{\mathrm{f}}\tilde{v}_p^i - {}_{\mathrm{f}}\hat{v}_p^i)}{\partial\, {}_{\mathrm{l}}\hat{y}_p^i} = -\frac{\partial\, {}_{\mathrm{f}}\hat{v}_p^i}{\partial\, {}_{\mathrm{f}}\hat{y}_p^i}\,\frac{\partial\, {}_{\mathrm{f}}\hat{y}_p^i}{\partial\, {}_{\mathrm{l}}\hat{y}_p^i} - \frac{\partial\, {}_{\mathrm{f}}\hat{v}_p^i}{\partial\, {}_{\mathrm{f}}\hat{z}_p^i}\,\frac{\partial\, {}_{\mathrm{f}}\hat{z}_p^i}{\partial\, {}_{\mathrm{l}}\hat{y}_p^i} \tag{3}$$

where

$$\begin{cases} {}_{\mathrm{f}}\hat{v}_p^i = (\,\beta\,\dfrac{{}_{\mathrm{f}}\hat{y}_p^i}{{}_{\mathrm{f}}\hat{z}_p^i} + v_0) \\ \begin{bmatrix} {}_{\mathrm{f}}\hat{x}_p^i \\ {}_{\mathrm{f}}\hat{y}_p^i \\ {}_{\mathrm{f}}\hat{z}_p^i \\ 1 \end{bmatrix} = \underbrace{\boldsymbol{\Delta}\widehat{\boldsymbol{T}}_{{}_{\mathrm{l}i+1}}^{-1}}_{\text{perturbation}}\, {}_{\mathrm{f}_i}\boldsymbol{T}^{{}_{\mathrm{l}i}} \begin{bmatrix} {}_{\mathrm{l}}\hat{x}_p^i \\ {}_{\mathrm{l}}\hat{y}_p^i \\ {}_{\mathrm{l}}\hat{z}_p^i \\ 1 \end{bmatrix} \end{cases} ; \tag{4}$$

β and v_0 are part of the intrinsic parameters of the left camera, and $\boldsymbol{\Delta}\widehat{\boldsymbol{T}}_{{}_{\mathrm{l}i+1}}$ represents the estimated rigid body perturbation on the **l**eft camera pose at keyframe $i+1$.

This method yields sub-millimetric corrections w.r.t. V-GPS on 3-D feature locations $\boldsymbol{x}_p^i$ and the relative pose ${}_{\mathrm{l}_i}\boldsymbol{T}^{\mathrm{f}_i}$ for every keyframe or feature set. Millimetric differences may arise on eventual loop closures after many keyframes, e.g. when scanning all around an object. On balance, it turns out that this method does not substantially improve the already accurate dead reckoning motion estimation by V-GPS. On the other hand, its computational cost is still low (2 to 5 ms)—roughly twice as long as V-GPS.

4.2 Appearance-Based Relocalization

Whenever

1. saccadic motion precludes sequential tracking,
2. the user browses outside a proximate scene, or
3. the cameras return to an area used before (loop closing),

pose tracking accuracy gets too low for consistent KLT tracking to be warranted anymore—even in its AM variant. Due to the richness of visual data, cameras are ideally suited for recognizing similarity; appearance-based relocalization can help to resume scanning *on the original reference frame*.

There exist a number of operators, called descriptors, that concern about the visual appearance of features in order to be distinctive and invariant to their viewpoint pose. We choose the performant SURF features in its original implementation, on stereo images. By using stereo images, the 3-D position of SURF features w.r.t. the camera ${}_{\mathrm{left}}\boldsymbol{T}^{\mathrm{SURF}}$ can be triangulated *at the same frame* during stereo initialization of the KLT feature set, where we obtained ${}_{\mathrm{left}}\boldsymbol{T}^{\mathrm{KLT}}$. By doing so, whenever 3 or more SURF features and consequently ${}_{\mathrm{now}}\boldsymbol{T}^{\mathrm{SURF}}$ are found again, ${}_{\mathrm{now}}\widehat{\boldsymbol{T}}^{\mathrm{KLT}}$ can be roughly estimated as follows:

$${}_{\mathrm{now}}\widehat{\boldsymbol{T}}^{\mathrm{KLT}} = {}_{\mathrm{now}}\boldsymbol{T}^{\mathrm{SURF}}\left({}_{\mathrm{left}}\boldsymbol{T}^{\mathrm{SURF}}\right)^{-1}{}_{\mathrm{left}}\boldsymbol{T}^{\mathrm{KLT}}\,. \tag{5}$$

This estimation is far less accurate than sequential pose tracking using V-GPS, compromising seamless transition to KLT tracking. We opt for using interleaved,

monocular three point perspective (P3P) pose estimation on KLT features for increased accuracy. Finally, regular KLT tracking takes on sequential pose tracking *on the original reference frame*—not without prior scaling and affine distortion of the features' templates according to the current relative pose.

4.3 Global, Hybrid Bundle Adjustment on Loop Closures

Loop closure events occur whenever former scene features that have not been recently tracked are being revisited. These events present the opportunity to greatly increase present and past pose tracking accuracy.

We distinguish between two types of loop closures: local loop closures can still take advantage of metric information for improved performance, whereas global, large-scale loop closure ought to be independent of motion estimation precisely because its main objective is to correct inaccurate motion estimation in the first place. Global, large-scale loop closing may instead concern itself with the projected appearance of features, which are still discriminative in the face of unknown localization, see Sect. 4.2.

Whatever the nature of the loop closure, it is indicated to subsequently optimize structure and motion estimations in the light of the discrepancies between expected and actually matched loop-closing features. 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. In the event of loop closure, however, current projections also depend on the camera motion history, *i.e.*, on all relative transformations and stereo feature triangulations even since the creation of the newly regained features, see Fig. 3.

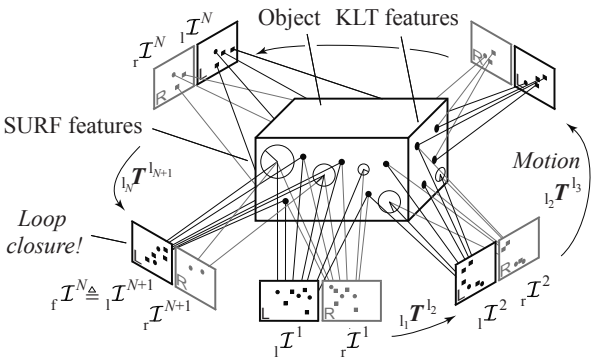

Fig. 3. Skeleton of stereo keyframes 1..N when browsing around an object. During monocular tracking of feature set #N, feature set #1 can be retrieved at images ${}_{\mathrm{l,r}}\mathcal{I}^{N+1}$. Depending on the distance traveled, loop closing occurs either by monocular tracking of KLT features or with the help of stereo SURF features.

As a consequence, the optimal solution by nonlinear optimization consisting in the minimization of squared reprojection residuals presents higher complexity than the local optimization in Eq. (1).

Now:

$$\widehat{\boldsymbol{\Omega}}_\star = \arg\min \sum_{i=j}^{N} \left(\sum_{p=1}^{M_i} \Big(||{}_1\tilde{\boldsymbol{m}}_p^i - {}_1\hat{\boldsymbol{m}}_p^i({}_1\hat{\boldsymbol{x}}_p^i)||^2 + ||{}_\mathrm{r}\tilde{\boldsymbol{m}}_p^i - {}_\mathrm{r}\hat{\boldsymbol{m}}_p^i({}_1\boldsymbol{T}^\mathrm{r}, {}_1\hat{\boldsymbol{x}}_p^i)||^2 + ||{}_\mathrm{f}\tilde{\boldsymbol{m}}_p^i - {}_\mathrm{f}\hat{\boldsymbol{m}}_p^i({}_{1_i}\widehat{\boldsymbol{T}}^{\mathrm{f}_i}, {}_1\hat{\boldsymbol{x}}_p^i)||^2 \Big) + \sum_{p\in\mathcal{R}} ||{}_1\tilde{\boldsymbol{r}}_p^j - {}_1\hat{\boldsymbol{r}}_p^j({}_{1_j}\widehat{\boldsymbol{T}}^{\mathrm{f}_j}, \ldots, {}_{1_N}\widehat{\boldsymbol{T}}^{\mathrm{f}_N}, {}_1\hat{\boldsymbol{x}}_p^j)||^2 \right) \tag{6}$$

where the parameters to be optimized $\boldsymbol{\Omega}_\star = [\boldsymbol{\Omega}^j .. \boldsymbol{\Omega}^N]$ include all history of 3-D features between the older feature set $\#j$ being found again, and the last tracked feature set $\#N$ (*i.e.*, $N-j+1$ feature sets in total), as well as the $N-j$ relative, inter-keyframe transformations between their respectives keyframes and the **f**inal local pose ${}_{1_N}\widehat{\boldsymbol{T}}^{\mathrm{f}_N}$ where the loop was closed (included in $\boldsymbol{\Omega}^N$). In total, this amounts to $\sum_{i=j}^{N}(3\cdot M_i+6)$ parameters, compared to $3\cdot M_i+6$ in Eq. (1). Note that, due to the non-convexity of the regular BA problem, we are optimizing over (differential perturbations of) non-privileged, relative transformations in order to avoid local minima [9]. Consequently, feature locations and camera motions are both locally Euclidean, but globally topological. The global Euclidean representation remains as a separate task, left aside e.g. for the realtime meshing application to consistently visualize it in realtime, perhaps augmenting it with the live image stream.

The residual in Eq. (6) is composed of the accumulation of residuals $\boldsymbol{\Delta m}_p^i$ in Eq. (1), now for every feature set i within the loop, as well as for the subset $\mathcal{R}$ of features contained in feature set j that have been found again in projections ${}_1\tilde{\boldsymbol{r}}_p^j = [{}_{1_{N+1}}\tilde{u}_p^j \; {}_{1_{N+1}}\tilde{v}_p^j]^\mathsf{T}$, see Fig. 3. In matricial form, the number of equations amounts to $\sum_{i=j}^{N}(2\cdot 3\cdot M_i)+2\cdot\mathrm{size}(\mathcal{R})$ compared to $2\cdot 3\cdot M_i$ in the case of local BA for dead reckoning in Eq. (1).

Optimization processes with system equations of this magnitude clearly benefit from sparse optimization methods if their Jacobians are sparse. Indeed, zero elements are pervasive in the Jacobian of this system of equations w.r.t. the abovementioned parameters due to the relative nature of the formulation used:

$$\frac{\partial(\tilde{\boldsymbol{m}} - \hat{\boldsymbol{m}})}{\partial \boldsymbol{\Omega}} = \begin{array}{c} \overbrace{\phantom{\boldsymbol{\Omega}^j}}^{\boldsymbol{\Omega}^j} \cdots \overbrace{\phantom{\boldsymbol{\Omega}^N}}^{\boldsymbol{\Omega}^N} \\ \left[\begin{array}{ccc} \frac{\partial \boldsymbol{\Delta m}^j}{\partial \boldsymbol{\Omega}^j} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \frac{\partial \boldsymbol{\Delta m}^N}{\partial \boldsymbol{\Omega}^N} \\ \hline & \frac{\partial \boldsymbol{\Delta r}^j}{\partial \boldsymbol{\Omega}} & \end{array}\right] \begin{array}{l} \leftarrow \boldsymbol{m}^j \\ \\ \leftarrow \boldsymbol{m}^N \\ \leftarrow \boldsymbol{r}^j \end{array} \end{array} \tag{7}$$

where

$$\frac{\partial \boldsymbol{\Delta r}^j}{\partial \boldsymbol{\Omega}} = \left[\begin{array}{ccccc} & & \vdots & & \\ \frac{\partial \boldsymbol{\Delta r}_p^j}{\partial \boldsymbol{\Omega}^j} & \cdots & \frac{\partial \boldsymbol{\Delta r}_p^j}{\partial \boldsymbol{\Omega}^k} & \cdots & \frac{\partial \boldsymbol{\Delta r}_p^j}{\partial \boldsymbol{\Omega}^N} \\ & & \vdots & & \end{array}\right]$$

$$\frac{\partial \boldsymbol{\Delta r}_p^j}{\partial \boldsymbol{\Omega}^j} = \left[\begin{array}{c|c|c|c} \blacksquare\blacksquare\blacksquare\blacksquare\square\blacksquare \\ \blacksquare\blacksquare\blacksquare\square\blacksquare\blacksquare & \mathbf{0}_{2\times 3(p-1)} & \blacksquare\blacksquare\blacksquare & \mathbf{0}_{2\times 3(M_i-p)} \\ \blacksquare\blacksquare\blacksquare & & \blacksquare\blacksquare\blacksquare & \end{array}\right]$$

$$\frac{\partial \boldsymbol{\Delta r}_p^j}{\partial \boldsymbol{\Omega}^k} = \left[\begin{array}{c} \blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare \\ \blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare \end{array} \middle| \mathbf{0}_{2\times 3M_i} \right]$$
$$\frac{\partial \boldsymbol{\Delta r}_p^j}{\partial \boldsymbol{\Omega}^N} = \left[\begin{array}{c} \blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare \\ \blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare \end{array} \middle| \mathbf{0}_{2\times 3M_i} \right] \tag{8}$$

∗ White boxes correspond to zero elements; gray or black boxes to non-zero ones.

We now go into detail about the calculation of the black Jacobian element highlighted above concerning features within $\mathcal{R}$ that have been tracked again. The estimated reprojection ${}_{\mathrm{l}}\hat{\boldsymbol{r}}_p^j$ of these features from feature set $\#j$ onto the **l**eft camera frame at keyframe $\#N{+}1$ is a function of both, the 3-D structure ${}_{\mathrm{l}}\hat{\boldsymbol{x}}_p^j$ of the original set $\#j$ and the current left camera pose ${}_{\mathrm{l}_j}\widehat{\boldsymbol{T}}^{\mathrm{f}_N}$ that, in turn, is a function of all local transformations by dead reckoning lying between keyframes $\#j$ and $\#N+1$. Here the calculation of its partial derivative w.r.t. the first Euler angle ${}_{\mathrm{l}}\alpha^k$ of the differential perturbation $\Delta\widehat{\boldsymbol{T}}_{\mathrm{l}_k}^{-1}$ at the **l**eft camera frame of keyframe $\#k$ is detailed:

$$\frac{\partial({}_{\mathrm{l}_{N+1}}\tilde{v}_p^j - {}_{\mathrm{l}_{N+1}}\hat{v}_p^j)}{\partial\, {}_{\mathrm{l}}\alpha^k} = -\frac{\partial\, {}_{\mathrm{l}_{N+1}}\hat{v}_p^j}{\partial\, {}_{\mathrm{l}_{N+1}}\hat{y}_p^j}\,\frac{\partial\, {}_{\mathrm{l}_{N+1}}\hat{y}_p^j}{\partial\, {}_{\mathrm{l}}\alpha^k} - \frac{\partial\, {}_{\mathrm{l}_{N+1}}\hat{v}_p^j}{\partial\, {}_{\mathrm{l}_{N+1}}\hat{z}_p^j}\,\frac{\partial\, {}_{\mathrm{l}_{N+1}}\hat{z}_p^j}{\partial\, {}_{\mathrm{l}}\alpha^k} \tag{9}$$

where

$$\begin{cases} {}_{\mathrm{l}_{N+1}}\hat{v}_p^j = (\,\beta\,\dfrac{{}_{\mathrm{l}_{N+1}}\hat{y}_p^j}{{}_{\mathrm{l}_{N+1}}\hat{z}_p^j} + v_0) \\ \begin{bmatrix} {}_{\mathrm{l}_{N+1}}\hat{x}_p^j \\ {}_{\mathrm{l}_{N+1}}\hat{y}_p^j \\ {}_{\mathrm{l}_{N+1}}\hat{z}_p^j \\ 1 \end{bmatrix} = {}_{\mathrm{f}_N}\widehat{\boldsymbol{T}}^{\mathrm{l}_{k+1}} \underbrace{\Delta\widehat{\boldsymbol{T}}_{\mathrm{l}_{k+1}}^{-1}}_{\text{perturbation}} {}_{\mathrm{f}_k}\widehat{\boldsymbol{T}}^{\mathrm{l}_j} \begin{bmatrix} {}_{\mathrm{l}_j}\hat{x}_p^j \\ {}_{\mathrm{l}_j}\hat{y}_p^j \\ {}_{\mathrm{l}_j}\hat{z}_p^j \\ 1 \end{bmatrix} \end{cases} . \tag{10}$$

These few features are of extreme importance, as they produce the only residuals bringing about loop-closing information—else global optimization equals repeated local optimization by dead reckoning in Eq. (1).

In reality, the formulation explained above corresponds to the ideal case where all features tracked at loop closure have also been tracked at their triangulation frame, *i.e.*, ${}_{\mathrm{f}}\boldsymbol{m}_p^j$ exists and is included in both Eqs. (6) and (7); however, features that were not succesfully tracked until keyframe $\#j{+}1$ can readily be found again when closing the loop. In that case (approx. 15% of the detected features), the residual Eq. (6), the optimization parameters $\boldsymbol{\Omega}$, as well as the Jacobian in Eq. (7) have to be extended to include their initial projections ${}_{\mathrm{l}}\boldsymbol{m}_p^j$ and ${}_{\mathrm{r}}\boldsymbol{m}_p^j$ as well as their 3-D locations.

Our hybrid optimization utilizes the nonlinear, least squares sparse optimization function `sparselm_dercrs()` detailed in Ref. [10], as well as supernodal sparse Cholesky factorization by CHOLMOD and graph partitioning by METIS to observe both primary and secondary sparsity structures of the Jacobian in Eq. (7) [11]. We provide the abovementioned, full analytic Jacobian in CRS format for improved performance. Of course, common derivative terms are being stored

instead of recalculated. By way of example, timekeeping improves from 94 sec (standard BA *with* full analytic Jacobian) to between 750 ms and 1.4 sec using the sparse variant. Not providing analytic Jacobians proves slower by a factor of 2 or 3. Global BA is performed in a separate computing thread in order not to disrupt concurrent real-time pose tracking and 3-D modeling. In Sect. 6 we show extended loop closure experiments, where global BA compensates for substantial dead reckoning errors of several cm in the course of obtaining consistent topology of the map.

Apart from the novel, hybrid nature of our approach to anchor global scale by stereo vision in selected frames (which incidentally deskills local pose tracking), our work differs from similar relative implementations in the SLAM literature ([12,13,14,9,15,16,17]) since accurate motion tracking is here required globally, for the whole motion history, whereas in SLAM metric accuracy is encouraged only locally, as global topological integrity suffices [13].

5 Experimental Validation

We suggest the interested reader to retrieve the videos of the original visual pose tracking in Refs. [1,2] at `http://goo.gl/8ZaJ52` and `http://goo.gl/PjDeox`. In this section we describe the image sequence attached as supplementary material to this paper, see `http://goo.gl/tqf4vB`. It shows an extended scan around a 50 cm tall sculpture. A natural browsing procedure asks for prolonged sweeps and is characterized by the absence of loop closure events (neither local nor global), *i.e.*, only dead reckoning estimation is possible. Tracking starts in frame #23033, featuring 4 sweeps in 90° relative yaw angles, prior to loop closing in frame #24521 for a total motion length of 320 cm. During the whole trayectory 44 feature sets are initialized by feature-based stereo vision.

As can be seen by the drift of the white circles corresponding to the features of the two first datasets, dead reckoning errors accumulate to an extent that precludes seamless KLT tracking when trying to retrieve these sets based on their expected relative pose to the camera—even in its AM implementation. Appearance-based relocalization on stereo images (triggered on a sensible basis based on the rough pose of the camera) may detect older SURF features, but their positioning accuracy by stereo vision is still insufficient. It is only by the inclusion of the intermediate stage concerning P3P pose estimation on KLT features with larger search regions that we achieve the required pose accuracy for seamless KLT tracking of 55 features pertaining to the feature set #1. After that, pose refinement by global, hybrid BA as explained in Sect. 4.3 takes place. Of course, relocalization on SURF features and subsequent P3P pose estimation happened at an older image frame because these prior stages run in a separate computing thread. After successful pose refinement by P3P pose estimation, the AM implementation of the extended KLT tracker takes over [2], tracking as many features of the original feature set #1 as possible. These 55 features in turn trigger the global, hybrid BA process explained in Sect. 4.3 in a separate computing thread. It is only by image frame #24535 that pose refinement on the

whole pose graph is complete, updating all 43 relative transformations ${}_{l_i}\boldsymbol{T}^{\mathrm{f}_i}$, $\forall i \in \mathbb{N}_1$, $i \leq 44$ along with the 3-D pose of all 1816 features $\boldsymbol{x}_p^i$, $\forall p \in \mathbb{N}_1$, $p \leq M_i$.

Using a dated notebook equipped with an Intel® Core™ 2 Duo P8700 processor, the robustified nonlinear optimization takes 870 ms. The parameters vector contains all features and relative poses involved in the loop closure (size 5769). The size of the residuals vector is 11090 including past and current *hybrid* residuals on stereo and monocular images.

The final pose correction after 320 cm of dead reckoning estimation amounts to 2.5 cm and 6.5°. The appearance-based stage in Sect. 4.2 misses the point by 7.5 mm and 1.5°, which is still adequate for successful tracking by the AM implementation of the KLT tracking in Ref. [2]. Fig. 4 shows a typical correction of the resulting full mesh after successful closure of the loop.

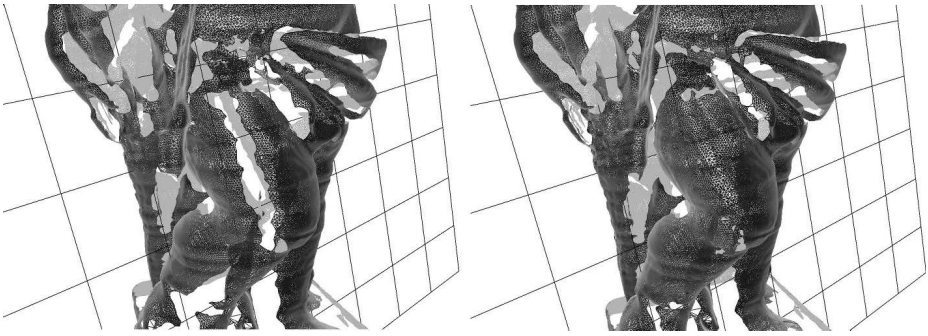

Fig. 4. Full mesh before and after loop closure correction

6 Conclusions

In this contribution we extend the real-time visual pose tracking algorithm originally presented in Refs. [1] and [2] into pose-graph optimization in the form of a hybrid, sparse bundle adjustment (BA) on a set of stereo keyframes and monocular views. This extension is necessary as 3-D modeling by scanning techniques is largely an exploratory act, *i.e.*, dead reckoning motion estimation plays a central role. It is well known that dead reckoning is subject to drifts, which will preclude larger 3-D modeling tasks e.g. scanning around objects.

We learned that BA for dead reckoning estimation hardly improves accuracy compared with V-GPS. In the case of loop closures, however, the use of BA makes large pose corrections possible. This is on the assumption that older features are tracked at loop closures, which is not possible in the presence of motion drifts without further ado. It is by appearance-based relocalization methods, together with a bank of parallel three-point-perspective pose solvers, that seamless feature tracking following Ref. [2] is possible—irrespective of the accumulated motion drift. In addition, we confirm that it is crucial to consider the sparsity of the pose-graph optimization problem for BA to perform in a timely manner.

These types of low-cost systems have the potential to promote 3-D modeling and conquer new markets owing to their passivity and flexibility of use.

References

1. Strobl, K.H., Mair, E., Bodenmüller, T., Kielhöfer, S., Sepp, W., Suppa, M., Burschka, D., Hirzinger, G.: The Self-Referenced DLR 3D-Modeler. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, pp. 21–28 (2009) (best paper finalist)
2. Strobl, K.H., Mair, E., Hirzinger, G.: Image-Based Pose Estimation for 3-D Modeling in Rapid, Hand-Held Motion. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, pp. 2593–2600 (2011)
3. Meister, S., Izadi, S., Kohli, P., Hämmerle, M., Rother, C., Kondermann, D.: When Can We Use KinectFusion for Ground Truth Acquisition? In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Color-Depth Camera Fusion in Robotics, Villamoura, Portugal (2012)
4. Strasdat, H., Montiel, J.M.M., Davison, A.J.: Real-Time Monocular SLAM: Why Filter? In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp. 2657–2664 (2010) (best vision paper award)
5. Burschka, D., Hager, G.D.: V-GPS – Image-Based Control for 3D Guidance Systems. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, pp. 1789–1795 (2003)
6. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: Proc. of the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan (2007)
7. Nistér, D., Naroditsky, O., Bergen, J.: Visual Odometry for Ground Vehicle Applications. Journal of Field Robotics 23 (2006)
8. Lourakis, M.I.A.: levmar: Levenberg-Marquardt Nonlinear Least Squares Algorithms in C/C++ (July 2004)
9. Strasdat, H., Montiel, J.M.M., Davison, A.: Scale Drift-Aware Large Scale Monocular SLAM. In: Proc. of Robotics: Science and Systems, Zaragoza, Spain (2010)
10. Lourakis, M.I.A.: Sparse Non-linear Least Squares Optimization for Geometric Vision. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 43–56. Springer, Heidelberg (2010)
11. Konolige, K.: Sparse Sparse Bundle Adjustment. In: Proc. of the British Machine Vision Conference (BMVC), Aberystwyth, Wales (2010)
12. Mei, C., Sibley, G., Cummins, M., Newman, P., Reid, I.: RSLAM: A System for Large-Scale Mapping in Constant-Time using Stereo. International Journal of Computer Vision 94, 198–214 (2010), Special issue of BMVC
13. Sibley, G., Mei, C., Reid, I., Newman, P.: Adaptive Relative Bundle Adjustment. In: Proc. of Robotics: Science and Systems, Seattle, USA (2009)
14. Konolige, K., Agrawal, M.: FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. IEEE Transactions on Robotics 24, 1066–1077 (2008)
15. Strasdat, H., Davison, A.J., Montiel, J.M.M., Konolige, K.: Double Window Optimisation for Constant Time Visual SLAM. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, pp. 2352–2359 (2011)
16. Clipp, B., Lim, J., Frahm, J.M., Pollefeys, M.: Parallel, Real-Time Visual SLAM. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, pp. 3961–3968 (2010)
17. Lim, J., Frahm, J.M., Pollefeys, M.: Online Environment Mapping. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, pp. 3489–3496 (2011)