The Self-Referenced DLR 3D-Modeler

K. H. Strobl,* E. Mair,§ T. Bodenmüller,* S. Kielhöfer,* W. Sepp,*
M. Suppa,* D. Burschka,§* and G. Hirzinger*§

*Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
D-82230 Wessling, Germany
Klaus.Strobl@dlr.de

§Department of Informatics
Technische Universität München
D-85748 Garching, Germany
Elmar.Mair@cs.tum.edu

***Abstract*— In the context of 3-D scene modeling, this work aims at the accurate estimation of the pose of a close-range 3-D modeling device, in real-time and passively from its own images. This novel development makes it possible to abandon using inconvenient, expensive external positioning systems. The approach comprises an ego-motion algorithm tracking natural, distinctive features, concurrently with customary 3-D modeling of the scene. The use of stereo vision, an inertial measurement unit, and robust cost functions for pose estimation further increases performance. The method is validated by demonstrations and abundant video material.**

I. Introduction

Visual perception is the process by which visual sensory information about the environment is received and interpreted, and it is key for the achievement of truly autonomous robots. Visual perception does not necessarily reveal a geometric 3-D model of the scene; 3-D modeling on the other hand focuses precisely on generating these surface models and leaves interpretation aside. However, it is believed that it is only through the formation of 3-D models that a considerable number of the remaining challenges on visual perception will be eventually solved; in fact, 3-D vision already took on applications previously treated in 2-D. A number of areas outside robotics also demand solutions on this point, e.g. computer graphics, industrial inspection and recognition, security, cultural heritage, or medical and scientific imaging—key findings in paleontology using 3-D scanners were recently reported in Ref. [1].

In this work we focus on 3-D modeling systems for close-range applications. These systems are twofold motivated: on the one hand for close-range applications themselves, and on the other as a complementary device for large-scale, extensive 3-D modeling applications which rarely fulfill a task using a sole device. Both motivations require *flexible* platforms and this work is a significant step in this direction.

In general, it is not possible for a 3-D modeling system to acquire a model at one single measurement step due to its limited field of operation, object self-occlusion, or object size—this particularly holds for close-range systems. The 3-D geometrical information gathered from a single vantage point is limited, so that multiple views (or multiple sensors) are required in order to subsequently merge data to a single 3-D model.[1] In this regard, the most widespread approach is to measure the position and orientation (pose) of the sensor while sensing, thereby registering multiple views in real-time. For this purpose tracking systems, robotic manipulators, passive arms, turntables, CMMs or electromagnetic devices are commonly used—some of them allow for convenient, hand-held operation of the sensor. These options are extremely limiting for three reasons: *firstly*, they limit the user on mobility—even external tracking systems restrict movements, especially in two rotational degrees of freedom; *secondly*, the final performance strongly depends on an accurate synchronization and an external hand-eye calibration, which are cumbersome, error-prone processes [2]—and what is more the hand-eye attachment cannot be rearranged; *lastly*, it turns out that every external positioning system usually represents the largest and most expensive part of the modeling system.

Digital cameras are widely used in robotics because they are light, affordable, consume less energy, allow for a very accurate parametrization of its (fairly simple) operating model, and still they gather a large amount of information (both radiometric an geometric) within a single, rapid measuring cycle. In addition, cameras are already present in most close-range 3-D modeling systems.[2] From these strengths concerning both lightness and accuracy it is difficult to overlook the potential of digital cameras for estimating the external reference of the device by themselves. Further benefits exist: cameras are non-contact sensors, thus free-floating, and they are passive since they do not need to project or exert action on the environment. In addition, the positioning estimation becomes inherently synchronized with further image-based sensing. Especially in robotics, where 3-D sensors are meant to increase robotic autonomy, doing without spatially limiting external positioning systems by the use of digital cameras should be a matter of priority. Unfortunately, this point currently only applies within the SLAM community.

Even though the frame rate of regular cameras seems sufficient and the images carry plenty of information, it is still a difficult task to yield real-time estimations from images because geometric information becomes entangled in radiometric and perspective geometry issues. The contribution

[1] Projective reconstruction approaches, or even metric but unscaled reconstruction often is not an option e.g. for the tasks mentioned above.

[2] Unfortunately, they are often severely handicapped in order to facilitate their joint operation with other sensors, e.g. for laser light triangulation where the cameras may be optically filtered to the laser wavelength. Our contribution in Ref. [3] brings forward novel ideas to avoid doing so.

of this work is precisely on a real-time capable, highly accurate extension for hand-held 3-D modeling systems by ego-motion estimation based on regular camera images.

In order to alleviate difficulties, inertial sensors are being extensively implemented in vision systems lately. This is because inertial measurement units (IMUs) perfectly complement off-the-shelf cameras both in measuring rate and in temporal precision: On the one hand, regular cameras take, say 25 images per second and estimations from their images are, in principle, equally accurate all the time. On the other hand, IMUs yield data at kHz rates and their readings are far more accurate when they lie close together in time than when they lie far apart. Thus the following tactics is used: IMU data that are close in time are meant to support image-based estimations (in the images where they coincide in time). The opposite may also apply. This can be done either by fusing final pose outcomes from both sensors (either stochastically or just in time), or directly for one sensor to support pose estimation *within* the other sensor's estimation process. The latter approach is employed in this work where IMU data will support the prediction step of the location of the image projections of world features.

The remainder of this article is as follows: Section II is a survey on related systems. In Section III we present our own 3-D modeling system, the DLR 3D-Modeler, which now implements the ego-motion algorithm of Section IV. This is the central section of the article and is illustrated by the diagram in Fig. 2. The experimental validation in Section V will surely help to comprehend the algorithm. We conclude in Section VI and reveal present research directions.

II. State of the Art

The aim of this section is to review the most significant lightweight, non-contact, close-range 3-D modeling devices, in particular related to the main contribution of this paper: concurrent real-time ego-motion estimation. For a larger review on 3-D modeling devices we refer the reader to Refs. [4] and [5]. Furthermore we shall mainly focus on mature, commercial systems; we mention research works only in the areas where commercial systems are missing.

Close-range 3-D modeling devices are currently being offered on joint account with very diverse referencing systems:

- *External, optical (infrared mostly) tracking systems* by Northern Digital Inc., Metris NV, and Steinbichler Optotechnik GmbH.
- *Passive arms* by FARO Technologies Inc., KREON Technologies, RSI GmbH, Metris NV, and ShapeGrabber Inc.
- *Low-rate, image-based pose estimation* by Noomeo SAS.
- *High-rate, image-based pose estimation* by Creaform Inc.
- *Electromagnetic positioning* by Polhemus Inc.
- *Turntables* by Cyberware Inc. and Polygon Technology GmbH.

From this it appears that the HandyScan 3D of Creaform Inc. (also marketed as ZScanner® by Z Corporation) lies very close by our goal. However, the necessity to adhere reflective, self-adhesive markers to the objects is inconvenient and inflexible. In fact, in a number of applications it is prohibited or impossible. Furthermore the dependency on active infrared illumination may also entail limitations.

Similarly, the DAVID-Laserscanner is commercially available and does without an external positioning system [6]. In fact, the software estimates the pose *of the laser projector* from images from a static camera that, at the same time, estimates the ranges to its projections on the object by triangulation. For this it is necessary to set a rectangular corner behind the object; the pose of the laser projector is easily estimated from the remaining laser projections onto the walls of the corner. A similar principle is used in Ref. [7]. In accordance to the last paragraph, placing a corner behind an object is not always possible, thus inflexible. Furthermore, the user feels very limited since it is not possible to locate the laser projector any nearer to the static camera. Most importantly, the approach is fundamentally limited to a single view and corresponds to *static*, close-range 3-D modeling since the corner, the object, or the camera have to be moved to obtain further range images from a different vantage point, and subsequently registered either by software or using turntables.

In Refs. [8] and [9] a self-referenced, hand-held cross-hair laser stripe profiler system is presented. The stereo camera makes use of fixed points, actively projected onto the scene, and localizes itself continuously by stereo triangulation w.r.t. these points. Actively projecting marker points is a cumbersome process and furthermore limits flexibility since the cameras must see the markers continuously. In addition, both the laser profiler operation and texturing become influenced by active illumination. The algorithm seems to lack of robustness, and efficiency considerations are missing.

Another approach to concurrent localization and 3-D modeling is presented in Ref. [10]; this active approach uses laser-range scanners that operate at mid-range as follows: a horizontal scanner is used for localization whereas a vertical scanner acquires the scene.

Actual image-based, *passive* localization approaches for 3-D modeling do exist:

The approach in Ref. [11] 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. [12] does make partial use of the camera calibration for metric reconstruction. The approach is intended for dense stereo vision applications and is not real-time. Accuracy analyses are also missing, even though non-stochastic approaches to self-calibration potentially compromise it.

Finally it is worth mentioning the recent development by MDA Ltd., Space Missions, in Ref. [13]: the instant Scene Modeler iSM. The system effectively provides 3-D models from hand-held stereo vision by registering views with scaled poses from vision-based ego-motion estimation. In contrast to the objectives in this work the system aims at mid-range operation using dense stereo vision. Stereo is computationally expensive and, therefore, the frame-rate will necessarily be

low, which in turn makes ego-motion estimation more difficult under unknown motion. The problem is solved by the use of SIFT features—which again are computationally expensive, and by the choice of cameras with low resolution.

Note that all these devices use only one sensor type (mostly triangulation-based) assigned to a specific task; incidentally the sensors usually include at least one camera.

The bottom line is as follows: *Neither extensively multi-sensory, nor image-based, passively self-referenced high-rate modeling devices currently exist*. In the next section we shall review our multisensory 3-D modeling device; Section IV will explain how to concurrently localize it from its own images accurately, in real-time, and in a truly passive way; finally, Section V presents the device in operation.

III. The DLR 3D-Modeler

The DLR 3D-Modeler is a multi-purpose platform for geometric and visual perception [14]. It combines multiple sensors in a compact, generic way, conveniently complementing each other. Further highlights of the platform are its low weight and power consumption, the accurate and adaptable synchronization of internal and external sensors, its on-board computational capabilities, its generic mechanical interfaces, the functional communication channel to computers via FireWire, as well as the availability of an extensive, congruent software suite. Current applications comprise 3-D modeling, tracking, visual servoing, exploration, path planning, and object recognition e.g. as the perception system of the humanoid robot Justin [15]. Its main sensory components are:

- The DLR Laser-Range Scanner (LRS) [16] operates by laser light triangulation. A visible, weak laser ray is continuously rotated and its reflection is intelligently recorded by a position sensitive detector. Because of its robust data acquisition capabilities and its small size, the LRS is used as a high definition, short-range sensor. Its wide scan angle has further advantages in robot vision as well.
- The DLR Laser Stripe Profiler (LSP) [3], [14] includes two laser beams that sequentially project stripes on a surface. These stripes are recorded by the cameras and reconstructed by triangulation—this is the dual, cross-hair operation mode; operation with only one laser is still possible. The LSP delivers close- to mid-range geometric information. It is worth noting that the LSP does without optical filters on the cameras, as they would make stereo vision, texturing, and visual ego-motion impossible.
- The stereo camera consists of two AVT Marlin F-046C progressive scan cameras, resolution 780×582, separated $50\,mm$ from each other, and with $6\,mm$ Sony VCL-06S12XM objectives mounted on them. The base distance and the focal length were chosen to cover the remaining desired sensing range from approx. $30\,cm$ up to $2\,m$. The implemented stereo algorithm is detailed in Ref. [17].
- The rigidly attached inertial measurement unit (IMU) is the AscTec AutoPilot from Ascending Technologies GmbH [18]. It features 6-DoF (3 gyros, 3 accelerometers, 3 magnetometers) with attitude estimation at $1\,kHz$, on-board data fusion, and a second $60\,MHz$ ARM processor that remains fully available for potential use by the user. It only weighs $19.6\,g$ and is size $10 \times 50 \times 50\,mm$.

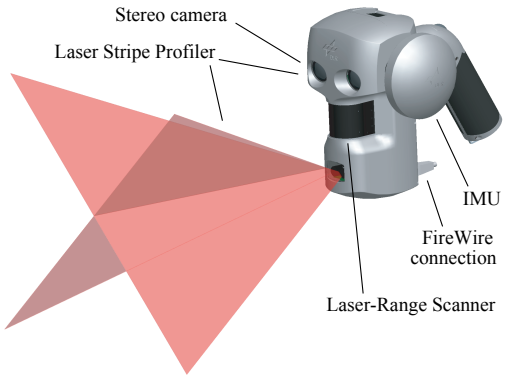

Fig. 1. The DLR 3D-Modeler and its components.

So far, the external pose of the device was provided by either an external tracking system with infrared-reflecting markers mounted on the 3D-Modeler,[3] or by attaching it to a robotic manipulator or a passive arm. These options are very limiting. In the next section a method for image-based pose estimation from the images of the stereo camera is presented.

IV. Concurrent 3-D Modeling and Ego-Motion Estimation

In Section I it was mentioned that the estimation of the pose of the 3D-Modeler from the information of its own sensors would signify a major improvement primarily in flexibility and costs of the system, and secondarily in potential positioning accuracy. In this section we present a method for real-time, accurate pose estimation from images that has been specially tailored to the 3D-Modeler and does not actively affect the scene nor the other sensors, since they will function concurrently. Further, in Section V the feasibility of this method will be demonstrated.

For 3-D modeling with a self-referenced 3D-Modeler **three major requirements** arise: *1) real-time capability* for the method to supply pose information, *2) high positioning accuracy* as required for 3-D modeling in general (compared to robotic manipulators or tracking systems *plus* the corresponding hand-eye transformations),[4] and *3) time-invariant estimations*, which means that repeated scans should provide the same (high) accuracy irrespective of the scanning time.

We next specify **three major consequences** that follow from the preceding requirements. Firstly, real-time capability implies both that all calculations should be regularly performed within e.g. $40\,ms$ ($25\,Hz$) *and* that this should

[3] Active, infrared-*emitting* diode-based markers have been recently also implemented.

[4] Typical accuracies for robotic manipulators are $\sigma_\theta < 0.1^\circ$ and $\sigma_p \approx 0.5\,mm$; for infrared tracking systems $\sigma_\theta \approx 0.25^\circ$ and $\sigma_p > 0.5\,mm$. The accuracy of the tracking system in orientation depends on the constellation of markers and is therefore very limited, which is critical for 3-D modeling. Ego-motion estimation should definitely better this.

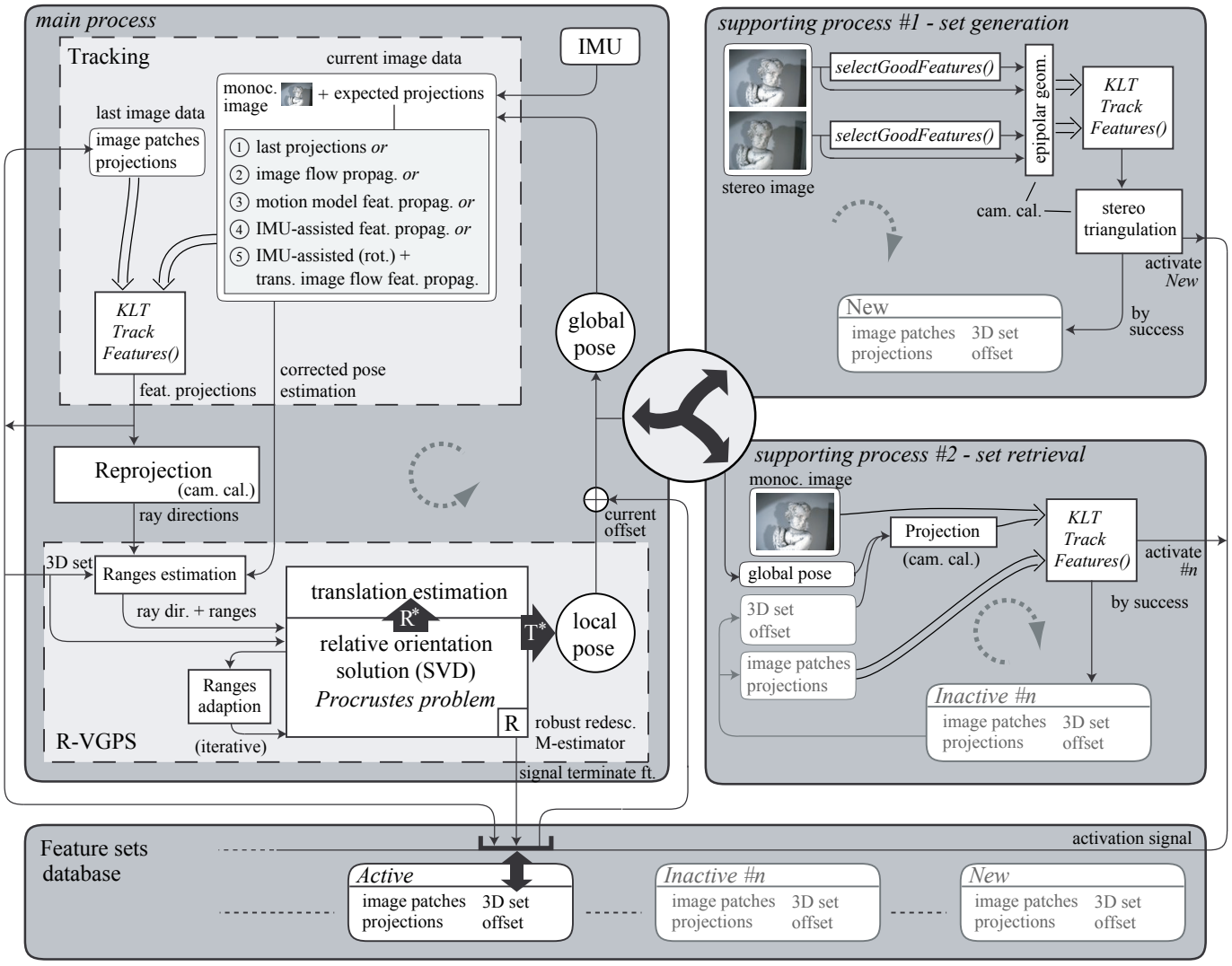

Fig. 2. Block diagram for the ego-motion estimation algorithm. Three main processes are differentiated. The feature sets database serves them storing data.

hold all the time, i.e. irrespective of the motion history; we support this requirement on processing time by the choice of a *feature-based approach* where the algorithm processes naturally salient, local regions of the images, i.e. a distributed representation of them—thus saving computing power compared to approaches processing whole images. Furthermore, the requirement on constant processing time irrespective of the motion history merges with the requirement on time-invariant precision in the estimation mentioned before, and points at the selection of a *non-stochastic approach* for sequential pose estimation. Stochastic approaches use knowledge of the modeling errors (e.g. noise in image processing or uncertainty in the models' parametrization or in the motion model) in order to increase certainty (accuracy) in the estimations.[5]

[5] In doing so, stochastic approaches usually get more computationally intensive (both in required computing power and in memory) as time goes by. This is because of the consideration of uncertain motion models, landmark positions *and* position relations, as well as of the actual motion estimation. All estimations—present and past—become related and it is by the mathematical ability of the developer that these calculations may become lighter, e.g. by carefully decoupling weak stochastic dependencies. Unfortunately, this issue and others usually lead to inconsistency [19].

This feature is most relevant *if* that extra accuracy is required. In fact, requirement #2 states that high accuracy is actually required for this system. However, it turns out that the system is already capable of doing highly accurate 3-D reconstruction of surface features in the scanning area by *feature-based stereo vision*, which provides a highly accurate structure estimation which in turn allows for highly accurate, non-stochastic pose estimation. In addition, by using feature-based stereo vision the algorithm only processes the strictly required 3-D structure information for accurate 6-D localization—extensive 3-D modeling is left for concurrent operation of the modeling capabilities of the 3D-Modeler.

This rationale (cf. Fig. 3) leads to the following development of a feature-based, non-stochastic ego-motion estimation algorithm that requires stereo initialization of natural features and monocular tracking of these features over time.[6] The result of the tracking is the motion of known features in

[6] These features are supposed to be in rigid coupling, thus in general no deformable object or dynamic scene should be treated without further modification of the algorithm. However, in reality, moving objects *are* tolerated as a by-product of the robustified approach that will be presented in Section IV-C.

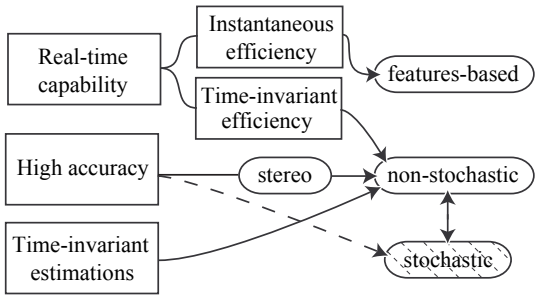

Fig. 3. Requirements, implications, and consequences.

a monocular image stream in relation to both the (static) magnifying characteristics of the camera and its motion (i.e. its perspective characteristics altogether). For extracting the camera motion from this (see Fig. 4), we opt for an efficient solution to the relative pose estimation problem: the Visual-GPS method first presented in Ref. [20]. Further, a data management scheme has to be defined in order to command these subsystems as well as to govern acquired data.

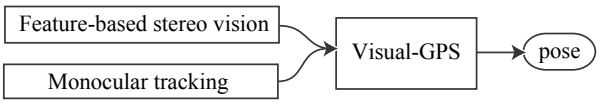

Fig. 4. Ego-motion algorithm: Feature-based stereo vision and monocular tracking serve the Visual-GPS, which pays out with camera pose estimations.

A. Accurate, Stereo-Based Structure Estimation

For accuracy reasons, we choose to produce structure by passive feature-based stereo vision (stereo triangulation).[7] The particular approach used in this work is detailed in Ref. [21]. Fig. 5 shows an accuracy analysis. It is worth mentioning that stereo processing cannot be performed in real-time since it typically requires approx. $0.5\,s$, thus we opt for performing initialization in a separated computing thread while concurrently tracking already initialized features in the former thread so that positioning information is always available. If no initialized data are available (first initialization), the potential features for initialization from one image are tracked until their corresponding features in the other image are found, and subsequently triangulated.

B. Efficient Monocular Tracking of Distinctive Features

The ego-motion estimation algorithm basically compares a known 3-D set of features in the scene as a result of the last section, with their current image projections—provided the correspondences feature-to-projection are known. For the algorithm to correctly relate these, two different approaches can be adopted: either to search for the appearance (2-D patch) of the feature in any whole image, or to look for it locally,

[7] This initialization module could be readily replaced by a monocular one, so that the system only needs one camera. Here further accuracy analyses should be carried out, and the absolute scaling issue should be tackled as well.

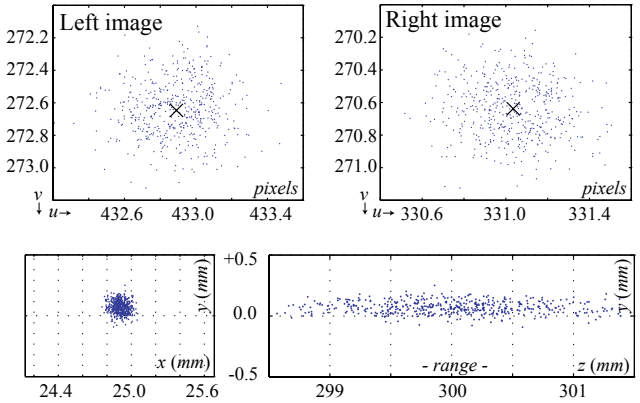

Fig. 5. Monte Carlo analysis on the expected confidence in feature-based stereo vision. A feature located at $\{24.9,\ 0,\ 300\}\,mm$ w.r.t. the left camera (basis distance between cameras $50.1\,mm$) is erroneously detected in both images (top, 1000 samples) with $\sqrt{\sigma_u^2+\sigma_v^2}=0.25\,p.$. The triangulation results show a bias of $+0.004\,mm$ in range (z), and a standard deviation σ_z of $0.6\,mm$. In x and y: $\sigma_x=\sigma_y=0.05\,mm$.

in a particular spot of the image after tracking the feature ever since its 3-D initialization.[8] The latter option is called feature tracking and is based on the premise that features slightly drift in successive images—which usually holds if the camera motion is moderate—and that evaluating these images is opportune, e.g. that the image sequence is already being processed by some other reason. Incidentally, this suits the 3D-Modeler operation.

Both the already presented feature-based stereo vision initialization and this monocular tracking are based on the Kanade-Lucas-Tomasi feature tracker. The available implementation in Ref. [22] was further developed for more efficient and robust operation, refer to [21].

Apart from low-level improvements in image processing, a major contribution to boost performance can be made by feeding in informed estimations of the expected drift of the features. Depending on the available information as well as on its quality, one of the following *image flow prediction schemes* can be used in our system:

1) extrapolation of the last feature projections in time, i.e. features are going to be searched for locally in the image, starting at the locations where they were last found;
2) extrapolation of the (2-D) motion of the last projections, also called optical flow;
3) extrapolation of the last camera motion w.r.t. the scene, usually assuming either constant velocity or acceleration;
4) IMU-assisted camera motion prediction using the last estimated camera motion and the integrated IMU outputs (rotational rate and linear accelerations); or
5) IMU-assisted camera orientation prediction together with optical flow-based translational extrapolation of the projections.

The latter, novel approach was specially developed for this application and will be detailed next, since it provides accuracy and robustness together with a simple implementation.

[8] In Section IV-D we shall see that losing track of individual features is also allowed through uninterrupted pose estimation.

Even though translations and rotations are not commutative in general, their effects on feature projections are clearly differentiated: camera rotations cause drift of feature projections irrespective of their range w.r.t. the camera, whereas camera translations only have a measurable effect at close range. Since the 3D-Modeler operates at close range, both aspects of the motion will affect the drift. In fact, experiments show that (exaggeratedly) jerky hand-guided operation entails maximal speeds of approx. $75\,°/s$ ($3\,°$ between frames at $25\,Hz$) and $0.5\,m/s$ ($2\,cm$ between frames) which correspond, at their worst, to 40 and 50 pixels optical flow between frames (rotational and translational components), at a typical range of $30\,cm$. Thus both components present similar maximal potential drifts and neither of them should be neglected (Fig. 6).

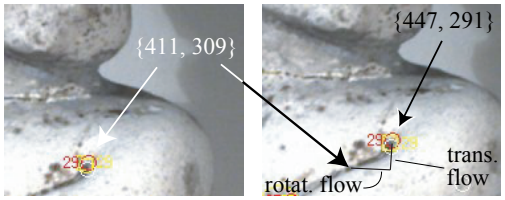

Fig. 6. Image flow in the same area in two consecutive images. The feature drifts from $\{411,\ 309\}$ to $\{447,\ 291\}$ a distance of 40.2 pixels within 40 ms. 37 pixels result from rotation, $17\,p.$ from translation; some pixels cancel out.

Nevertheless, it is not the amplitude of the optical flow that is critical for successful tracking, but the certainty of its prediction. It is for this reason that in general it holds that informed, IMU-assisted predictions are more accurate than mere (uninformed) motion extrapolations. However, it turns out that hand-held operation allows for much better uninformed *translational* camera motion prediction than *rotational*. This is because of the bulky body of the sensor: It is much easier for the user to rotate it with a facile twist movement than to linearly accelerate the whole sensor in any direction. This easily translates into the following circumstance: Translational drifts may be as big as the rotational ones, but they vary much less in time or, in other words, they allow for more accurate prediction. There exist other factors that discourage the prediction of translational optical flow from IMU data: firstly, translation is given in form of its second derivative, which calls for repeated integration and for maintaining a camera motion model and an IMU model for its own drifts as well; secondly, acceleration values are usually noisier than the rotational rate ones, even after integration; thirdly, the gravity vector has to be estimated and subtracted all the time; lastly, translational camera motions depend both on the rotational rate and on the linear acceleration readings of the IMU, whereas the camera rotation directly corresponds to the integrated rotational rate at the IMU because it is a rigid body motion. What is more, these relationships are defined by an external IMU-to-camera calibration process, which can be very simple for the rotational component but complex and prone to errors for the translational one.

These considerations led us to the development of a *hybrid flow prediction scheme* (⑤ in Fig. 2) where the predicted optical flow $\hat{\boldsymbol{f}}^t$ at time t decomposes into its rotational and translational components: The rotational part $\boldsymbol{f}_{rot}^t$ is an informed prediction from the (integration of the) rotational rate of the IMU, whereas the translational part $\hat{\boldsymbol{f}}_{tra}^t$ is an extrapolation of the last *translational* optical flow $\boldsymbol{f}_{tra}^{t-1}$, for each feature. The latter stems from the subtraction of the last, informed rotational optical flow $\boldsymbol{f}_{rot}^{t-1}$ from the last actual (as from the tracking step) optical flow $\boldsymbol{f}^{t-1}$ as follows: $\hat{\boldsymbol{f}}_{tra}^t \triangleq \boldsymbol{f}_{tra}^{t-1} = \boldsymbol{f}^{t-1} - \boldsymbol{f}_{rot}^{t-1}$.

To round off, and as a by-product, there still exists another major appeal for this approach: The hybrid flow prediction scheme becomes completely independent of any motion model or pose estimation process, but depends only on the IMU rotational rate readings (through both $\boldsymbol{f}_{rot}^{t-1}$ and $\boldsymbol{f}_{rot}^t$) and on the last tracking result (through $\boldsymbol{f}^{t-1}$) — values that are characterized by their low-level noise.

Of course, in the case of features that were not tracked in the past because of occlusions, blur, or limited field of view, this hybrid scheme cannot be applied. In this case, the features projections are predicted from the last pose together with a motion model, until optical flow information for these features exists and the hybrid scheme seamlessly assumes control.

C. Relative Pose Estimation: the Robustified Visual-GPS

In this section a method for relative camera pose estimation is presented. The method requires a known 3-D set of points (Section IV-A), together with their projections onto images (Section IV-B). Assuming a rigid 3-D set of points and constant camera parameters, the projections drift in the images is caused by varying perspective projection, i.e. to the varying pose of the camera w.r.t. the scene, thus the method will simply estimate camera poses that match these perspective drifts.

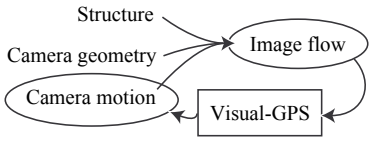

Fig. 7. Structure, camera geometry, and camera motion determine image flow.

Visual-GPS (V-GPS) is an algorithm that provides a solution to the relative orientation problem iteratively, but efficiently. After the determination of the orientation, the translation can be also estimated. The method assumes a known 3-D set of points related to the initial camera reference frame S_0 and solves the exterior orientation problem of the estimation of the following camera poses w.r.t. that reference set. It proceeds as follows: An additional, tentative 3-D set of points is generated in the camera reference frame S_t from the 2-D projections tracked throughout images 0 to t by using approximated ranges only; these ranges are estimated from the preceding pose estimation. In this way, the problem reduces to solving the absolute orientation between these two 3-D sets of points, which can be solved in closed form using the singular value decomposition (SVD).[9] Refer to the original publication in Ref. [20] for more details.

[9] A similar algorithm numerically minimizing reprojection errors has been also implemented without noticeable difference in performance; this is due to the high accuracy of the structure generated by the method in Section IV-A.

A tacit assumption for accurate, unbiased pose estimation is both that the initially known structure is perfectly known (e.g. from Section IV-A), and that monocular tracking accurately follows the *actual* features (Section IV-B). Unfortunately, repetitive background patterns may exist, and fast motion, occlusions, and shadows compromise tracking especially at close range, thus gross errors (outliers) are certainly expected. The modification of algorithms to allow for the detection and elimination of these outliers is called robustification. In order to attain unbiased pose estimation, robustification proceeds as follows: data that do not accurately match the expectations are not taken into account *at all*. Since expectations directly stem from models (including noise models and confidence values), in general it is advisable to perform robustification on the part of the algorithm that makes use of the most extensive models of the system, which in our case is the pose estimation—cf. Fig. 2.[10] On account of this, a novel robust formulation within the V-GPS algorithm is introduced in Ref. [21]; it makes use of a redescending M-estimator on the residual Euclidean distances between matched points. Extensive testing demonstrates the significance of robustification on accurate pose estimation, especially in range. The method not only neutralizes the effects of outliers, but also signalizes them so that they can be immediately removed from memory.

D. Sequential V-GPS: Feature Sets Generation and Retrieval

Since flexibility is one of the principal justifications for this approach, it must be able to tolerate the fact that features 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 the camera pose can be still estimated. Monocular tracking will continuously try to recover these features with the aid of the camera pose unless the robustified V-GPS marks them as invalid.

Long-term losses are features that are deliberately lost because their associated 3-D set of points becomes inadequate, thus compromising accurate, long-term pose estimation. In this respect two kinds of actions are possible, either to lose the features, or to recover them:

1) Generation of new Features Sets: Whenever too few features of the current 3-D set are being tracked or the 3-D set's center of mass (centroid) drifts outside the central area of the image, we command resetting as in Section IV-A and give up on tracking the current features as soon as the generation successfully concludes. Since this will take some image frames, it proceeds concurrently in a separated computing thread. After that, only the new features are being tracked and the current local relative pose estimation will add to the relative pose between the reference images corresponding to the initialization of the different 3-D sets of points (offset):

$$global\,pose\,\#1 = current\,relative\,pose \oplus \widetilde{offset}\ .$$

[10] For instance, a naive image-based robustification could be implemented, but the algorithm would lack of both the structure and the camera motion information, thus would perform worse in detecting outliers.

2) Retrieval of inactive Features Sets: Whenever the projection of the centroid of an inactive 3-D set of points is more central than the one of the current set, all its features within the current field of view are to be tracked again. The prediction of their projections is solely based on the pose estimation from the features of the current set in the current image, thus no extrapolation is needed, cf. Fig. 2. It is worth mentioning that by referring back to previous sets, potential positioning drifts that have accumulated by leaping onto newer 3-D sets completely disappear:

$$global\,pose\,\#2 = (curr.\ \ relative\,pose \oplus \widetilde{\cancel{offset}}) \ominus \widetilde{\cancel{offset}}\ ,$$

which is another appealing property of this approach compared to dead-reckoning (visual odometry) or even stochastic approaches, whose outcomes often depend on the particular path history, or even become inconsistent [19].

V. Experimental Validation

In this section we demonstrate the robustness of the presented approach for ego-motion estimation by analyzing its operation with an extremely challenging image sequence. Some key frames are reproduced in Fig. 8. We suggest that the reader also retrieves the processed video stream from the Internet.[11]

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

Next, we explain the behavior of the ego-motion algorithm, which sequentially localizes the camera w.r.t. eight different sets of points *in real-time*.[12] The sequence starts with an initialized set of 3-D points *Set*#1. The set is composed of 25 points and this is also the average number of features used by the following sets. Fig. 8 (a) shows *Set*#1. After that the camera moves downwards, see Fig. 8 (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 ego-motion algorithm does not create new sets of points but detects former ones following the policies in Section IV-D, see Fig. 8 (c), and leaps onto them. Fig. 8 (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.

[11] The processed image sequence as well as other demonstrative videos are available online at `http://www.robotic.dlr.de/Klaus.Strobl/iros2009`.

[12] The ego-motion algorithm runs on a "Mini-PC" equipped with an Intel® Core™ Duo T2050 processor and 2GB RAM. The resources are shared with further sensors.

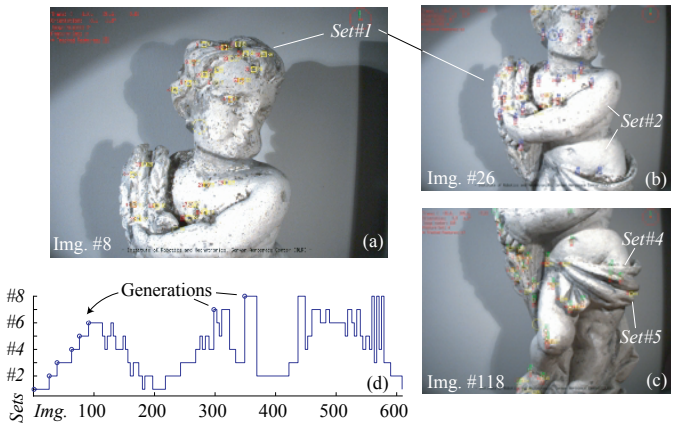

Fig. 8. (a) Image #8 tracking *Set*#1. (b) Image #26 after generation of *Set*#2, changing reference. (c) Image #118 while retrieving *Set*#4. (d) History of the reference sets during the experiment.

The behavior defined by the policies in Section IV-D yielded successful tracking all the time. The algorithm seamlessly leaps from current reference sets onto former ones which directly implies bias-free round-scanning (i.e. the positioning accuracy at the end of the sequence equals the accuracy at the beginning) and furthermore points at high accuracy in general. The accuracy experiment in Ref. [21] obtained *less* than $2\,mm$ and $0.4\,^\circ$ camera pose estimation error after $50\,cm$ of dead-reckoning motion estimation at a longer object range than $50\,cm$. This is due to the accurate set of 3-D points (Section IV-A), to the accurate prediction of the feature locations and subsequent tracking (Section IV-B), to the robust pose estimation (Section IV-C), the subtle management of reference sets (Section IV-D), as well as to the accurate camera calibration [23] and the IMU synchronization.

For a demonstration of the regular operation of the LSP with ego-motion estimation please refer to the attached video—during scanning, hectic movements were intentionally performed to prove the robustness of the system.

VI. Conclusion and Future Work

This work presents the first hand-held 3-D modeling device for close-range applications that references itself passively from its own images in real-time, at high-rate. This is an important contribution both in order to increase flexibility and to be able to do without external positioning systems that constrain the system in size, mobility, and cost.

A comprehensive review of 3-D modeling devices points out the lack of this type of systems. We presented an ego-motion algorithm thoughtfully tailored to the task. Its core characteristics are feature-based image processing, efficient 2-D monocular image tracking, and non-stochastic, robustified relative pose estimation. Stereo-based structure estimation and a novel IMU-supported optical flow prediction scheme make a further contribution to increased performance.

Future work will concern the examination of different motion models and the implementation of an information-driven tracking process in order to increase performance when not using an IMU. In addition, the implementation of this algorithm with only one camera will be also investigated.

References

[1] M. R. Bennett et al., "Early Hominin Foot Morphology Based on 1.5-Million-Year-Old Footprints from Ileret, Kenya," *Science*, vol. 323, no. 5918, pp. 1197–1201, February 27th, 2009.

[2] K. H. Strobl and G. Hirzinger, "Optimal Hand-Eye Calibration," in *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems IROS*, Beijing, China, October 2006, pp. 4647–4653.

[3] K. H. Strobl, W. Sepp, E. Wahl, T. Bodenmüller, M. Suppa, J. F. Seara, and G. Hirzinger, "The DLR Multisensory Hand-Guided Device: The Laser Stripe Profiler," in *Proc. of the IEEE Int. Conference on Robotics and Automation*, New Orleans, LA, USA, April 2004, pp. 1927–1932.

[4] F. Chen, G. M. Brown, and M. Song, "Overview of Three-Dimensional Shape Measurement using Optical Methods," *Optical Engineering*, vol. 39, no. 1, pp. 10–22, January 2000.

[5] F. Blais, "Review of 20 Years of Range Sensor Development," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 231–240, January 2004.

[6] DAVID-Laserscanner. DAVID Vision Systems GmbH. [Online]. Available: http://www.david-laserscanner.com

[7] J.-Y. Bouguet and P. Perona, "3D Photography Using Shadows in Dual-Space Geometry," *Int. Journal of Computer Vision*, vol. 35, 1999.

[8] P. Hébert, "A Self-Referenced Hand-Held Range Sensor," in *3-D Digital Imaging and Modeling 3DIM*, Quebec City, Que., Canada, May 2001.

[9] R. Khoury, "An Enhanced Positioning Algorithm for a Self-Referencing Hand-Held 3D Sensor," in *3rd Canadian Conference on Computer and Robot Vision CRV*, Quebec City, Que., Canada, June 2006, pp. 44–50.

[10] S. Ono and K. Ikeuchi, "Self-Position Estimation for Virtual 3D City Model Construction with the Use of Horizontal Line Laser Scanning," *Int. Journal of ITS Research*, vol. 2, pp. 67–75, October 2004.

[11] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, "Visual Modeling with a Hand-Held Camera," *Int. Journal of Computer Vision*, vol. 59, no. 3, pp. 207–232, 2004.

[12] G. Roth and A. Whitehead, "Using Projective Vision to Find Camera Positions in an Image Sequence," in *Vision Interface VI'2000*, Montreal, Canada, May 2000, pp. 87–94.

[13] S. Se and P. Jasiobedzki, "Stereo-Vision Based 3D Modeling and Localization for Unmanned Vehicles," *Int. Journal of Intelligent Control and Systems, Special Issue on Field Robotics and Intelligent Systems*, vol. 13, no. 1, pp. 47–58, March 2008.

[14] M. Suppa, S. Kielhöfer, J. Langwald, F. Hacker, K. H. Strobl, and G. Hirzinger, "The 3D-Modeller: A Multi-Purpose Vision Platform," in *Proc. of the IEEE Int. Conference on Robotics and Automation ICRA*, Rome, Italy, April 2007, pp. 781–787.

[15] Ch. Borst, T. Wimböck, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. Robuffo Giordano, R. Konietschke, W. Sepp, S. Fuchs, Ch. Rink, A. Albu-Schäffer, and G. Hirzinger, "Rollin' Justin - Mobile Platform with Variable Base," in *Proc. of the IEEE Int. Conference on Robotics and Automation ICRA*, Kobe, Japan, 2009, video contribution.

[16] F. Hacker, J. Dietrich, and G. Hirzinger, "A Laser-Triangulation Based Miniaturized 2-D Range-Scanner as Integral Part of a Multisensory Robot-Gripper," in *EOS Topical Metting on Optoelectronic Distance/Displacement Measurements and Apps.*, Nantes, France, 1997.

[17] H. Hirschmüller, "Stereo Processing by Semi-Global Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, February 2008.

[18] Ascending Technologies GmbH. [Online]. Available: www.asctec.de

[19] S. J. Julier and J. K. Uhlmann, "A Counter Example to the Theory of Simultaneous Localization and Map Building," in *Proc. of the IEEE Int. Conference on Robotics and Automation ICRA*, Seoul, Korea, May 2001, pp. 4238–4243.

[20] D. Burschka and G. D. Hager, "V-GPS – Image-Based Control for 3D Guidance Systems," in *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems IROS*, Las Vegas, NV, USA, October 2003, pp. 1789–1795.

[21] E. Mair, K. H. Strobl, M. Suppa, and D. Burschka, "Efficient Camera-Based Pose Estimation for Real-Time Applications," in *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems IROS*, St. Louis, MO, USA, October 2009, in press.

[22] S. Birchfield. KLT: Kanade-Lucas-Tomasi Feature Tracker. Dept. of Electrical and Computer Engineering, Clemson University. Clemson, SC, USA. [Online]. Available: http://www.ces.clemson.edu/~stb/klt/

[23] K. H. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. DLR CalDe and DLR CalLab. Institute of Robotics and Mechatronics, German Aerospace Center (DLR). Oberpfaffenhofen, Germany. [Online]. Available: http://www.robotic.dlr.de/callab/