Image-Based Pose Estimation for 3-D Modeling in Rapid, Hand-Held Motion

Klaus H. Strobl,[§] Elmar Mair,[‡] and Gerd Hirzinger[§‡]

[§]Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
D-82234 Wessling, Germany
Klaus.Strobl@dlr.de

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

***Abstract*— This work aims at accurate estimation of the pose of a close-range 3-D modeling device in real-time, at high-rate, and solely from its own images. In doing so, we replace external positioning systems that constrain the system in size, mobility, accuracy, and cost. At close range, accurate pose tracking from image features is hard because feature projections do not only drift in the face of rotation but also in the face of translation. Large, unknown feature drifts may impede real-time feature tracking and subsequent pose estimation—especially with concurrent operation of other 3-D sensors on the same computer. The problem is solved in Ref. [1] by the partial integration of readings from a backing inertial measurement unit (IMU). In this work we avoid using an IMU by improved feature matching: full utilization of the current state estimation (including structure) *during* feature matching enables decisive modifications of the matching parameters for more efficient tracking—we hereby follow the Active Matching paradigm.**

I. Introduction

In Ref. [1] we presented the self-referenced DLR 3D-Modeler, which is a 3-D modeling device for close-range applications combining complementary sensors in a compact, generic way. A single 3-D modeling device is however rarely capable of creating complete 3-D models of a scene since the geometrical information gathered from a single vantage point is naturally limited. Multiple views (or multiple sensors) are usually required to subsequently merge data to a single, complete 3-D model. The prevalent approach is to move a single sensor around the scene while concurrently measuring its position and orientation (pose), thereby registering multiple views—possibly in real-time. A wide range of external reference systems are commonly deployed for this purpose. These traditional options are however extremely limiting since they constrain the system in size, mobility, accuracy, and cost. In the aforementioned publication we demonstrated for the first time a hand-held 3-D modeling device for close-range applications *that localizes itself passively from its own images* in real-time and at high-rate.

The latter development required however support from an inertial measuring unit (IMU), synchronized and rigidly attached to the device, in order to still keep track of pose in the case of highly dynamic motion (e.g. hand-held). This is because, at close range, projections do not only drift in the face of rotation but they also drift in the face of translation—at far range translational effects are negligible. Moreover, in our particular application both effects are potentially of similar size and may add up to drifts beyond the capabilities of well-established tracking algorithms based on feature matching.[1] In Ref. [1] we bring forward a novel, *hybrid* feature drift *prediction* method that combines translational motion propagation with the rotational readings of an IMU. The method is easy to implement and allows for robust tracking with very high motion bandwidth. In this sequel work we aim at a tracking algorithm capable of handling larger projection drifts at high-rate *without* the use of an IMU, while concurrently digitizing the scene on the same computer, in real-time.

A. Active Vision

An active vision system is defined by Aloimonos in Ref. [2] as "*a system able to manipulate its visual parameters in a controlled manner in order to extract useful data about the scene in space and time that would allow it to best perform a set of tasks*." This is in contrast to general vision systems that aim at "*a* complete *and accurate representation of the scene*" [3] as active vision calls for *partial* perception—subject to the task at hand. Note that this subordination already suggests a top-down, feedback approach to vision. The classic example for active vision systems is humans and animals that evolved to active vision to narrow down perception to significant parts of the scene, which clearly yields an *advantage in the face of limited resources*.[2] Furthermore the purposive (or animated) vision paradigm includes the decision-taking process on *how* to manipulate the visual parameters, e.g. "*where to look next?*" [2], [4].

Active vision systems in robotics were predominantly developed concerning view direction control (gaze control) of cameras e.g. mounted on mobile robots [5]–[7]. These were primarily motivated by navigational tasks like self-localization or obstacle avoidance rather than by real-time computational constraints. The variable visual parameters comprise camera orientation and potentially the robot's own motion.

* This work was performed during the first author's research stay in Dr. Andrew Davison's group at Imperial College London. We are deeply grateful to him as well as to his students Dr. Margarita Chli and Mr. Ankur Handa.

[1] Image features are salient areas of the image that are assumed to correspond to locally planar, textured patches in 3-D. Since they are measurable projections of the state of the system, they can enable pose tracking. Features are an effective tool for tracking *if* search regions are kept small.

[2] Some persons with mental disorders (e.g. savant syndrome by Stephen Wiltshire MBE) do feature the talent of virtually limitless visual memory, but this is invariably in combination with cognitive deficits.

In the context of vision-based sequential localization and mapping (visual SLAM) [8], Davison in Ref. [9] noted that SLAM is an intrinsically passive problem separate from camera motion. He however claimed that we still can benefit of active, purposive vision for improved efficiency of visual SLAM by extending the scope of the aforementioned visual parameters to image processing itself—instead of considering it a detached, self-contained task. His method is coined Active Matching (AM) and basically puts image processing *into the loop* of SLAM in a statistically optimal way [9]–[11]. In this work we present a variation to AM tailored to our 3-D modeling application that aims at particularly efficient tracking with very high motion bandwidth.

B. Related Work

Visual SLAM approaches face computational issues either in the long run (e.g. due to map size) or on a frame by frame basis (because of high number of features or rapid motion). The latter case corresponds to our current problem. A few recent (monocular) approaches address it:

- **Active Matching** achieves a lower computational burden using feature projection priors dynamically to guide a feature by feature matching search. In this work we shall adapt this approach (described in Section III) to our particular ego-motion estimation algorithm.
- **Parallel Tracking and Mapping (PTAM)** by Klein and Murray in Refs. [12], [13] is similar to our approach in many aspects. PTAM computationally decouples mapping from tracking and relative pose estimation, which enables lax mapping and exhaustive tracking respectively. In addition, they achieve a remarkable level of robustness towards rapid camera motion by the use of FAST features [14] to identify potential matches and by an extensive pyramidal representation of images.
- **1-Point RANSAC approaches** e.g. Refs. [15], [16] utilize single-point motion hypotheses to constrain expected projection regions of the rest of the features iteratively, in the context of RANSAC. At first this is carried out using constrained motion models; after that assumptions are relaxed.
- **Accelerated, descriptor-based approaches** in Refs. [17]–[19] employ through feature descriptors. These allow for robust tracking and are usually intended to offline operation. The authors manage to condition them to the available information on the state (e.g. deciding on the relevant scales) so that their use just becomes possible in real-time. The use of accelerated hardware computing will surely boost these approaches, which are still computationally very expensive.

In the course of this work these approaches are being individually addressed on their adequacy concerning scarcity of resources by concurrent operation with 3-D modeling.

The remainder of this article is as follows: In Section II we recapitulate the IMU-supported ego-motion algorithm already presented in Ref. [1]. Next, Section III introduces the AM paradigm, which serves our purpose to present our novel development in Section IV as a best case scenario for AM. Section V shows experimental results on the system performance and Section VI summarizes the contribution.

II. Ego-Motion Estimation at the DLR 3D-Modeler

The DLR 3D-Modeler is a multi-purpose 3-D modeling device that can be mounted on a robot or hand-held. 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 [20].

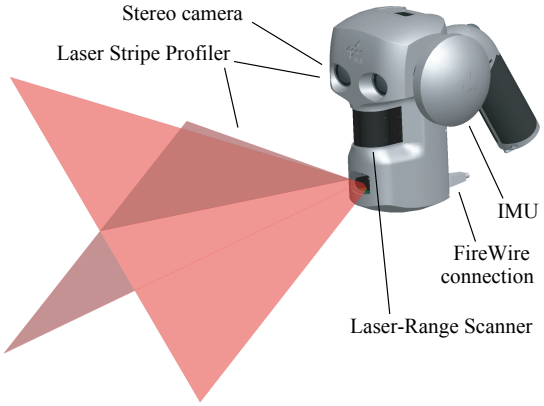

Fig. 1. The DLR 3D-Modeler and its components.

In Ref. [1] we presented an algorithm for efficient and accurate estimation of its motion from its own images. The novel development made it possible to abandon using inconvenient, expensive external positioning systems. It realized the first hand-held 3-D modeling device for close-range applications that localizes itself passively from its own images in real-time and at high-rate. Robust operation in the case of highly dynamic motion required support from a synchronized, on-board IMU.

A. Overview

The algorithm tracks natural, distinctive visual features (following the Shi-Tomasi criterion in Ref. [21]) while concurrently modeling the scene using customary, on-board 3-D sensors. Feature tracking is permanent on a monocular image stream using an extended implementation of the Kanade-Lucas-Tomasi (KLT) feature tracker [22]–[24]. During motion, incoming groups of new features are being initialized in parallel, in 3-D, using the synchronized stereo camera. Close-range initialization performs with sub-millimetric accuracy, which is basis for the non-stochastic nature of the approach. This in turn makes it possible to significantly cut down computing expenses as required for concurrent operation of other 3-D sensors. Rigid 3-D structure together with permanent monocular tracking serve the relative pose estimation algorithm, the robustified Visual-GPS [1], [24], [25], which efficiently supplies accurate pose estimation.

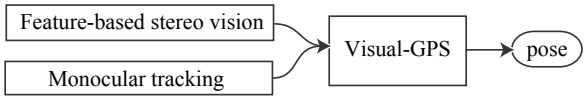

Fig. 2. Ego-motion algorithm: Feature-based stereo vision and monocular tracking serve the Visual-GPS that pays out with camera pose estimations.

Note our accordance with the PTAM paradigm of reducing degrees of freedom (DoF) in high-rate estimation in order to achieve better performance [12]. PTAM reduced them from $6+3\times N$ in general SLAM (N is the number of features) to 6 in PTAM, which estimates the further DoF (mapping) and absolute camera poses in a concurrent thread, at lower rate, from selected keyframes. Mapping in our algorithm also relies on keyframes, but substitutes bundle adjustment by accurate, feature-based stereo vision. The latter is computationally cheaper and it furthermore contributes absolute scaling—a prerequisite in 3-D modeling.

B. Sequential Feature Tracking

Sequential feature tracking is a predictive feature search method that exploits the absolute priors on their expected image projections in order to know where to focus processing resources in each image. These absolute priors depend on the 3-D location of the feature points and on the predicted motion of the camera. The latter motion is being estimated from past measurements and then further predicted using a motion model. 3-D structure, camera past motion estimations as well as its motion model may however differ from reality to some extent, which mirrors in the absolute priors on the expected projections and translates into "gated" image regions where each feature is expected to lie, refer to Fig. 3. The main purpose of the feature tracker is then to seek feature appearance matches within these bounded regions, hereby delivering the image drifts of features—and in doing so keeping track of correct data association as well.

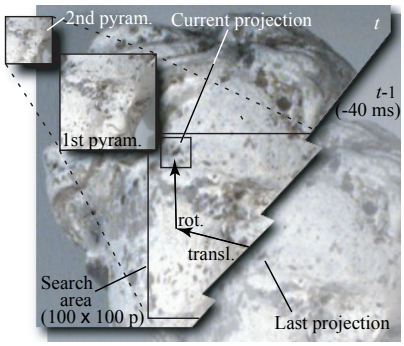

Fig. 3. KLT feature tracker with big search area due to large expected drifts. Two levels of the pyramidal representation of the image are also shown.

Critically, at close range both translations and rotations potentially cause image drifts of similar size (the rotational component is dominant at far range). The drifts may add up to long distances (e.g. search areas of 100×100 pixels) that are beyond the real-time capabilities of the regular KLT feature tracker, even in its pyramidal implementation [23]. The pyramidal implementation applies the original implementation's gradient descent search also to coarser resolutions (higher pyramidal levels) of the original image pair—for conveniency, pyramidal levels differ in size at least by powers of two (octave steps). Matching at lower-resolution images helps in the predominant case where, at original resolution, the search region is bigger than the 'basin of attraction' of the match function minimum—the latter depends on the chosen size of the feature template (typically between 7×7 and 11×11 pixels). The use of similar-sized patches in lower-resolution images implies bigger, virtual 'basins of attraction' at the original resolution, which aim at the size of the original search region to be able to track robustly with broader motion bandwidth. By sequentially searching into lower pyramidal levels (higher-resolution images), absolute convergence is in theory guaranteed, if the matching precision at the higher level is higher than the 'basin of attraction' in the lower level search—for all pyramid levels. The algorithm ideally ends up matching correctly at the lowest level—and matching is finished. Two significant limitations may apply:

1) The bigger the search range, the more pyramidal levels have to be created. This can render tracking computationally too expensive—especially with a high number of features.
2) Features following the Shi-Tomasi criterion are good features to track at the resolution where they were selected in the first place. At lower resolutions this does not necessarily hold anymore, as e.g. distinctive small corners will attenuate and potentially disappear.

In fact, hand-held operation of the DLR 3D-Modeler can be highly dynamic and its motion model will not be able to narrow down feature search areas to admissible sizes, see Fig. 3. Therefore, both limitations apply.

After consideration of several motion models, in Ref. [1] we opted for the insertion of an IMU, rigidly attached to the DLR 3D-Modeler, synchronized with respect to (w.r.t.) its inner measurement cycle, and externally calibrated w.r.t. it (at least in orientation). The higher data rate of the IMU makes it possible to predict feature projections more accurately, which then require much smaller search regions. We observed that feature drifts owing to camera translation can be much better predicted by a motion model than feature drifts owing to camera rotation—even though both are potentially of similar size. This allowed for a very simple and robust novel implementation by predicting the translation of the camera from a motion model, together with using *rotational* updates from the IMU. These informed predictions allow for smaller feature search regions that in turn facilitate rapid feature matching with still very high motion bandwidth.

C. Outlook

In our quest for flexibility and reduced cost, the replacement of an external positioning system by an IMU was already a significant step. Still, being the only purpose of the IMU to *lead* the search for features and not to directly support pose estimation accuracy, in Ref. [1] we stated our intention to implement a more efficient, information-driven tracking step to comply with motion and computational constraints without the need for an IMU. A further advantage is that exclusively image-based motion estimation implies inherent pose synchronization with all other image-based sensors.

In the next section we present the Active Matching (AM) paradigm that precisely aims at more efficient tracking. In Section IV we take it as a basis for our own implementation, which turns out to be a best case scenario for AM.

III. The Active Matching Paradigm

The vast majority of computer vision applications that include feature matching consider it as a separate task—a self-contained, bottom-up operation applied regularly to incoming images, i.e., a purely 2-D process between pairs of images. The results are then fed forward to higher level tasks like full structure and camera pose estimation.[3] In this way however feature matching waives its right both to access and to modify more informative representations of the system (and potentially of the state of the world, i.e., the state model) *during* operation. The AM paradigm in Refs. [9]–[11] breaks this habit of leaving aside feature matching from higher level estimation tasks, see Fig. 4, aiming at higher tracking performance in several aspects:

- **Built-in global consensus**. Instead of hypothesizing on correct data association after a monolytic feature matching process [29], AM can readily walk down the sole correct hypothesis *during* feature matching by alternation of single feature matching and subsequent state update—for all features. In doing so, AM puts image processing *into the loop* of the search for global consensus by not processing areas of the image where features are not really expected in the first place. Feature matching will then be trapped in far less matching ambiguities. In addition, in order to cope with residual mismatches due to unavoidable image ambiguity, in Refs. [10], [11] the authors make use of dynamic Mixture of Gaussians (MoG) representations.
- **Less computation through less image processing**. AM leads feature tracking to process far smaller areas of the image. This is because of the paradigm shift from matching between images to matching between an image and the state, which is a far more informative description of the system history.
- **Less computation through guided search**. A stochastic representation of the system along with the use of information theory makes it possible for AM to quantify potential information gain. Some feature measurements will be more informative than others; by taking their measurements first, the overall, eventual computational cost will be further reduced. Furthermore this allows for anticipated termination of feature matching at a point of diminishing returns.
- **Estimation accuracy**. Real-time algorithms are usually tuned to perform at full system capacity at the expense of e.g. a higher number of features being tracked, or more accurate feature matching or pose estimation results. Therefore, in real-time vision more efficient algorithms generally imply more accurate estimations.

The aforementioned aspects allow for more effective feature matching, but this is not the whole story because they are not for free as two important calculations must be repeatedly performed: first, the system state must be invariably updated after every single attempt of feature matching;[4] second, making guided search decisions on information-theoretic grounds is associated with substantial computational costs. It is appropriate to question whether the alleged information gain really merits these extra calculations involved, finally yielding an overall more efficient algorithm. In general, sensibly including image processing *into the loop* of e.g. global consensus will be more satisfactory than using RANSAC or JCBB after blunt, uninformed blanket feature matching [11].

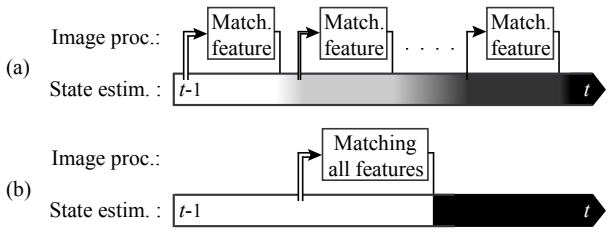

Fig. 4. Traditional methods (b) take absolute priors on feature projections once, where image projections are most uncertain. Active Matching (a) recursively updates (a representation of) the state after single feature matching so that feature projection priors can be more accurately estimated before a matching attempt starts (represented by the thickness of the arrows).

It is worth noting that the aforementioned aspects are potential but not compulsory; it is readily possible to take advantage of some aspects and not of the others. For instance, since a high number of features skyrocket the cost of making exact decisions on guided search, an alternative algorithm called Fast Active Matching (FAM) was recently proposed in Ref. [11]. FAM is cheaper than AM by making approximate, non-optimal decisions. On the other hand, Refs. [30], [31] precisely employ guided search for selective feature matching but do not update the state after single feature matching.

A. Stochastic Active Matching

The original AM formulation in Refs. [9]–[11] features a fully stochastic representation of both system state and measurements using multi-variate Gaussian joint probability distributions. This is in line with traditional SLAM approaches [8] and in addition it is convenient for full utilization of the capabilities of the AM paradigm. In particular, their approach involves an extended Kalman filter (EKF) with occasional inverse-depth parametrization of 3-D features, a dynamic MoG representation within each matching step, as well as information-theoretic guidance of feature search based on regularly updated mutual information scores between features. Guided search aims at selecting the feature that most efficiently reduces the search areas for all remaining fellow features at the next single feature matching step, i.e., it is generally aimed at maximizing tracking efficiency.[5]

[3] This methodology is presently being reinforced both by the development of more robust feature descriptors [26]–[28] as well as by further increase in availability of computing power.

[4] A more efficient representation of the system state, e.g., a *joint* distribution on the expected feature projections may be used instead [10], [11].

[5] This is a very sensible approach but is not unique. For instance, another option is minimizing the search area exclusively of the next, most probably picked feature. A further option is to minimize the eventual search areas of all remaining features taking all potential updates into account. The approach also differs from active vision approaches that aim at the minimization of predicted uncertainty in self-localization or eventual obstacle avoidance [7].

IV. Active Matching with Known Structure: A Best Case Scenario

When using AM to guide a feature by feature matching search, we are essentially modifying visual parameters for better perception of an active vision system—cf. Section I-A. Here the parameters include purposive decisions on which feature to seek for, as well as further parameters required for the search (e.g. search regions). This is a closed-loop problem where the current decision is conditioned to the task at hand (minimizing future search regions), which in turn depends on the results of past decisions.

AM recursively generates updated search regions subject to the accumulated information on the system's state (i.e., robot motion, 3-D scene, and perhaps the tracking quality of individual features). For efficiency reasons, the original AM formulation in Refs. [9]–[11] makes use of the 2-D projection of the current 3-D state estimate instead: the *joint* distribution on feature projections. It includes absolute priors on feature projection locations as well as relative priors on the correlations between these locations.[6] After every measurement they use the correlations to update absolute priors on further feature projections, i.e., future search regions, see Fig. 4.

On a feature by feature decision-taking basis, the commanded feature matching process is expected to maximize information gain so that the updated information on the system's state will minimize future search areas. In SLAM, inferring on the system's state in the light of a single feature drift is however hard, because both, 3-D scene and camera motion, are uncertain and indistinguishable to some extent. Less information can be gathered and eventual projection predictions will remain uncertain. Quite the contrary in our case of *accurate 3-D scene knowledge* (Section II-A) where immediate inference of camera motion from a few feature drifts is possible. Furthermore, all remaining feature drifts will only depend precisely on that newly estimated motion. For this reason we assert that non-stochastic AM with known structure is a *best case scenario for AM* where eventual search regions can be reduced outright. This is basically the same message as in Ref. [10]; the authors observe especially good performance of AM where "*priors on absolute feature locations will be weak but priors on relative locations may still be strong*."

Other authors have performed motion prediction from single feature tracking in the context of SLAM. In Ref. [16] Civera *et al.* update expected projections of features from a single bearing result using the state transition equations of an EKF; this is however under-determined and, as explained above, is in the presence of uncertain structure, which yields large search regions. The authors circumvent the problem by not performing all calculations but data association in the context of random, iterative sampling instead. On the opposite side, Scaramuzza *et al.* in Ref. [15] fully estimate motion from a sole correspondence but they employ a very restrictive motion model. Further, in Ref. [13] Klein and Murray introduce an efficient, inter-frame rotation estimator to aid tracking, using whole low-resolution images. It is on the assumption that the camera either senses at far range or only rotates—otherwise it fails in close-range translations.

Our approach is as follows: We aim at rapid, full (6-D) camera motion *preliminary* estimation using a minimal set of features thanks to our 3-D knowledge of scene features. This estimation will be used to update priors on feature locations yet to be measured. Now very small residuals will allow for extensive feature tracking and *final* accurate pose estimation (e.g. via Visual-GPS) in a highly efficient way—cf. Fig. 2.

The minimal set of known features for unconstrained motion prediction (6 DoF) comprises 3 perspective projection correspondences of non-collinear 3-D points—this was first described by Grunert in 1841 [33]. One of our principal findings in Ref. [1] was that, at close range, translational and rotational effects in projection drifts are potentially of similar size, but *translational effects can be accurately predicted using a constant camera velocity model*. From this we now propose reducing the required number of correspondences for full motion estimation from 3 to 2 (1.5 actually) by predicting camera translation from the motion model, so that only rotation remains to be estimated (3 DoF). We expect that potential translation prediction errors will not corrupt this preliminary rotation estimation. Our intention is that the projection estimations of the N–2 remaining features will fall within their respective 'basins of attraction' of regular KLT feature matching, e.g. 5 pixel radius. This directly rules out using expensive pyramidal representations for $N-2$ feature matching processes. Note the potential significance of such an achievement: By using AM, feature projection estimation errors are being reduced, for every remaining $N-2$ features, from potentially more than 50 to less than 5 pixels after two sole feature matching results, see Fig. 5. These last unavoidable residuals are consequences of the approximation concerning translation propagation.

The **dramatic reduction of image processing** involved is the biggest appeal for using AM: Only 2 features have to be extensively searched for (the second one less extensively, see Section IV-B) and the remaining features are easy prey for e.g. the regular KLT tracker. A second major appeal exists: Guided feature search based on information theory is, together with state update, main overhead in potential AM-related calculations [11]. Sensible guidance of feature search is indeed advantageous in SLAM because some 3-D features locations are more correlated than others [34]. In our case of full 3-D map knowledge however, all feature locations are equally (totally) correlated in $SE(3)$, and **fair preliminary motion estimation can be achieved by *any* feature pair used**. Third, we expect small residuals within their 'basins of attraction' for the remaining $N-2$ features. Therefore, **most features are clear of data association issues** since feature matching itself guarantees correct data association. It is only the first two *active* features that account for this issue. These points consolidate our view of non-stochastic AM with known structure as a best case scenario for AM.

[6] In doing so, [9]–[11] are the first to use the *full* stochastic representation of feature projections *during* feature search. It includes the correlations that reflect the stiffness of the structure estimation as well as the common motion. This is a similar problem to neglecting covariances in regular SLAM [32].

A. The KLT Feature Tracker with Larger Search Regions

The KLT feature tracker is able to cope with larger feature search regions by using pyramidal representations of image patches, refer to [23], [24] and Section II-B. This however poses difficulties both in computational cost and in successful feature matching at higher pyramidal levels.

Apart from improvements in image-processing operators and the use of AM itself, we performed two significant modifications to the tracker to further combat these limitations:

1) The height of the pyramidal representation of images is limited, for two reasons: *first*, the generation of images of different resolution in the context of the KLT feature tracker is expensive due to associated filtering and gradient computations; *second*, feature tracking at lower resolutions is prone to errors because features were selected at the original resolution in the first place. We opt for subsampling the original patch only once.
2) At the subsampled level we perform *exhaustive* template search instead of gradient descent search. Otherwise matching would get stuck in local minima because the search region at that resolution is still bigger than the template size. We use similar-sized templates to the ones at original resolution, which correspond to areas four times bigger at original resolution. Sequential, exhaustive template search using bigger templates at lower resolutions is very robust to ambiguities.

We employ this KLT implementation for extensive tracking of a minimal set of two *active* features, cf. Section IV-B.

Along with big search areas and data association issues, motion blur is a third drawback of rapid camera motion. It precludes accurate, point-wise feature tracking. In Refs. [12], [13] Klein and Murray meet with this problem. They ameliorate the damage using pyramidal representations of images, edge features, and even by exploiting its effects on images (directional image flow). In such cases tracking is coarsely warranted but accurate pose estimation is not. In this work we put stress on avoiding motion blur in the first place because highly-dynamic 3-D modeling requires accurate pose estimation *all the time*. Motion blur is minimized by using shorter shutter times (in our case a few milliseconds); adequate imaging can be facilitated by using wide aperture, valuable cameras, as well as proper scene illumination.

B. ***Preliminary*** *Pose Estimation from Two Feature Matches*

In this section we present an algorithm for rotation estimation from two sequentially tracked features. This initial estimation ${}_C\widehat{\boldsymbol{R}}_{\text{ptr}}$, together with translational propagation following a constant velocity motion model, will provide highly tight priors on all other feature projections. These will eventually allow for rapid, robust feature tracking and accurate camera pose estimation in the exact same manner as in Ref. [1].

The algorithm is detailed in Alg. 1 and Fig. 5. The choice of the *active* features $\boldsymbol{p}$ and $\boldsymbol{q}$ is quite immaterial—provided they were sequentially tracked during the last frames $\mathcal{I}^{t-1}$ and $\mathcal{I}^t$, and their templates at lower resolution are distinctive. We choose the two most distant valid features in the image in order to avoid noise in the estimation of ***roll*** rotation ${}_C\widehat{\boldsymbol{R}}_{\text{r}}$. The ***pan+tilt*** rotation ${}_C\widehat{\boldsymbol{R}}_{\text{pt}}$ is estimated ($\widehat{\ }$) from the first *active* feature $\boldsymbol{p}$. Together with ${}_C\widehat{\boldsymbol{R}}_{\text{r}}$ they form ${}_C\widehat{\boldsymbol{R}}_{\text{ptr}}$.

Algorithm 1 Pose est. and correction from a minimal set.

Require: Last tracked features and last camera transl. ${}_C\boldsymbol{t}^{t-1}$.
repeat
  Pick first *active* feature $\boldsymbol{p}$
  Apply **tra**nsl. propagation: $\hat{\boldsymbol{p}}^t_{\text{tra}} = \text{proj}\big({}_C\tilde{\boldsymbol{p}}^{t-1} - {}_C\boldsymbol{t}^{t-1}\big)$
  Exhaustive template match around $\hat{\boldsymbol{p}}^t_{\text{tra}}$ {*wide search*}
until reliable match $\tilde{\boldsymbol{p}}^t$ {*normally* 1×}
Estimate minimal rotation (2 DoF): ${}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{pt}}$ {Eq. (1)}
repeat
  Pick second *active* feature $\boldsymbol{q}$
  Apply transl. propagation: ${}_C\hat{\boldsymbol{q}}^t_{\text{tra}} = {}_C\tilde{\boldsymbol{q}}^{t-1} - {}_C\boldsymbol{t}^{t-1}$
  Apply minimal rotation: $\hat{\boldsymbol{q}}^t_{\text{tra+pt}} = \text{proj}\big({}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{pt}} \cdot {}_C\hat{\boldsymbol{q}}^t_{\text{tra}}\big)$
  Exhaust. templ. match around $\hat{\boldsymbol{q}}^t_{\text{tra+pt}}$ {*narrow search*}
until reliable match $\tilde{\boldsymbol{q}}^t$ {*normally* 1×}
Estimate remaining rotation (1 DoF): ${}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{r}}$ {Eq. (2)}
Pick random *validation* set e.g. ${}^{1..5}\boldsymbol{v}$
Apply translation propagation: ${}^{1..5}_{C}\hat{\boldsymbol{v}}^t_{\text{tra}} = {}^{1..5}_{C}\tilde{\boldsymbol{v}}^{t-1} - {}_C\boldsymbol{t}^{t-1}$
Apply rotation: ${}^{1..5}\hat{\boldsymbol{v}}^t_{\text{tra+ptr}} = \text{proj}\big({}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{pt}} \cdot {}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{r}} \cdot {}^{1..5}_{C}\hat{\boldsymbol{v}}^t_{\text{tra}}\big)$
Valid. by regular KLT tracker on ${}^{1..5}\hat{\boldsymbol{v}}^t_{\text{tra+ptr}}$ {else restart}
Apply transl. propagation to ${}^{i}_{C}\hat{\boldsymbol{f}}^t_{\text{tra}} = {}^{i}_{C}\tilde{\boldsymbol{f}}^{t-1} - {}_C\boldsymbol{t}^{t-1},\ \forall\, {}^i\boldsymbol{f} \in \mathcal{I}^t$
Apply rotation: ${}^i\hat{\boldsymbol{f}}^t_{\text{tra+ptr}} = \text{proj}\big({}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{pt}} \cdot {}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{r}} \cdot {}^{i}_{C}\hat{\boldsymbol{f}}^t_{\text{tra}}\big)$
return updated feat. projections ${}^i\hat{\boldsymbol{f}}^t_{\text{tra+ptr}}$ for regular KLT.

From the discrepancy between the propagated estimation $\hat{\boldsymbol{p}}^t_{\text{tra}}$ and the exhaustive first matching result $\tilde{\boldsymbol{p}}^t$, the minimal camera rotation potentially responsible for it (2 DoF) reads:

$$ {}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{pt}} \begin{cases} \text{axis:} & {}_C\underline{\hat{\boldsymbol{p}}}^t_{\text{tra}} \times {}_C\underline{\tilde{\boldsymbol{p}}}^t \\ \text{angle:} & \pm \arccos\big({}_C\underline{\hat{\boldsymbol{p}}}^t_{\text{tra}} \cdot {}_C\underline{\tilde{\boldsymbol{p}}}^t\big) \end{cases} \tag{1} $$

where ${}_C\underline{\boldsymbol{p}}^t = {}_C\boldsymbol{p}^t / |{}_C\boldsymbol{p}^t|$ and ${}_C\boldsymbol{p}^t$ is the 3-D location of a feature $\boldsymbol{p}$ in the camera reference frame S_C at time t. ${}_C\tilde{\boldsymbol{p}}^t$ is the direction in S_C of the 2-D, actually tracked feature $\tilde{\boldsymbol{p}}^t$.

From the second match $\tilde{\boldsymbol{q}}^t$ the only remaining DoF can be estimated: *roll* rotation ${}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{r}}$ around the axis ${}_C\tilde{\boldsymbol{p}}^t$ that relates the planes containing the estimated projection ${}_C\hat{\boldsymbol{q}}^t_{\text{tra+pt}}$ and the actual one ${}_C\tilde{\boldsymbol{q}}^t$:

$$ {}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{r}} \begin{cases} \text{axis:} & {}_C\underline{\tilde{\boldsymbol{p}}}^t \\ \text{angle:} & \pm \arccos\Big(\big({}_C\underline{\tilde{\boldsymbol{p}}}^t \times {}_C\underline{\hat{\boldsymbol{q}}}^t_{\text{tra+pt}}\big) \cdot \big({}_C\underline{\tilde{\boldsymbol{p}}}^t \times {}_C\underline{\tilde{\boldsymbol{q}}}^t\big)\Big) \end{cases} \tag{2} $$

that jointly with the *pan+tilt* rotation in Eq. (1) yields

$$ {}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{ptr}} = {}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{pt}} \cdot {}_C\widehat{\boldsymbol{R}}^{t-1,\,t}_{\text{r}}, $$

which is good estimate of the camera rotation between $t-1$ and t. Together with the last camera translation ${}_C\boldsymbol{t}^{t-1}$ it can be used to recompute prior beliefs on further feature projections. Note that the *pan+tilt* rotation ${}_C\widehat{\boldsymbol{R}}_{\text{pt}}$ obtained from feature $\boldsymbol{p}$ was also used to better track feature $\boldsymbol{q}$.

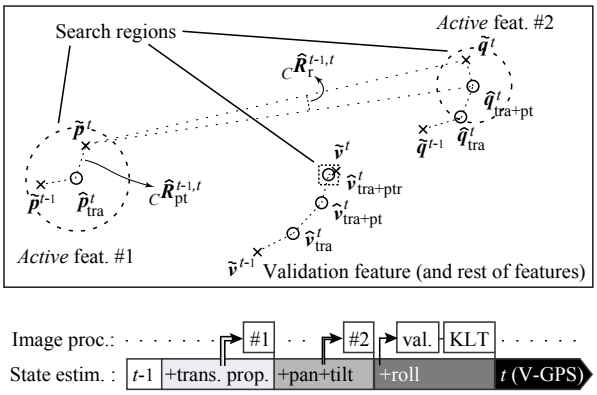

Fig. 5. Top: Pictorial schematic on the 2-D estimations involved. Two *active* features $\boldsymbol{p}$ and $\boldsymbol{q}$ as well as the resulting estimation steps on a further feature $\boldsymbol{v}$ are detailed. The latter is tracked using the regular KLT feature tracker. Bottom: Time evolution of state estimation w.r.t. the image processing steps.

After successful tracking of features $\boldsymbol{p}$ and $\boldsymbol{q}$ we opt for tracking a random subset of five of the remaining features in order to validate the rotation hypothesis in case of mismatches or inaccurate translational motion propagation. The validation features are rapidly tracked using the standard, gradient descent KLT tracker at the original resolution only. Here correct matching demands projection accuracies of half of the template size (e.g. 11×11, thus 5 pixels). Valid hypotheses bring about rapid, extensive matching of all remaining features ${}^{i}\boldsymbol{f}$ (typically 20 to 50), as in the validation step.

Two types of errors may appear when searching for *active* features: *First*, indistinctive matching templates *at lower resolution*, or corrupted projections (e.g. occlusions). The frequency of these errors is minimized thanks to sequential matching. They are best detected during matching itself—are signalized and will not be further used as *active* features. *Second*, image ambiguity may cause incorrect data association (false positives) even though exhaustive search and sequential matching minimize the risk. Consistency checks w.r.t. the state history support correct data association [29], [31]—this is our validation step. Both types of errors are however rare in regular operation. Since hypothesis generation (tracking of $\boldsymbol{p}$ and $\boldsymbol{q}$) is expensive, we opt for rigorous preemption: one sole hypothesis will be generated *unless* the aforementioned errors appear. In exceptional cases of multiple errors at the same image, the computational overhead may exceed the time budget for matching (e.g. 20 ms). These occasional peaks can be filtered out by making use of an image buffer, e.g., of the last two images. This implies a latency of e.g. 80 ms, which is admissible in most applications. Furthermore, occasional tracking losses are still possible e.g. in untextured scenes or unfinished stereo initialization. We implemented a SURF-based relocalization stage to register back to former KLT features [27]. In this case absolute pose estimation is not provided, for several seconds.

The sizes of the search areas for the two *active* features are empirically based on worst case experiments at 25 Hz. They amount to circles of 50 and 25 pixels radii respectively. The second search area is smaller because ${}_{C}\widehat{\boldsymbol{R}}_{\mathrm{pt}}$ is known.[7]

[7] A strap considering both, *roll* orientation uncertainty and translational propagation error, could allow tighter search for the second *active* feature.

Typical matching times on a notebook equipped with an Intel® Core™ 2 Duo P8700 processor are: for *active* feature #1 3.2 ms (50 p. radius) or 2.3 ms (40 p.); for *active* feature #2 1.3 ms (25 p. radius) or 1.0 ms (20 p.); standard matching of 5 validation features takes 0.6 ms; the remaining features require 1.7 ms (15 features) or 2.7 ms (20). Relative pose estimation using V-GPS takes between 1.5 and 2.3 ms. Thus all calculations typically take 10 ms on one thread—acquisition rate is 25 Hz. A second thread is occasionally prompted for stereo initialization. Therefore, a vast amount of resources is left for I/O, visualization, and most importantly for operation of the other 3-D sensors.

It is worth noting that the approach scales well with increasing frame rate since it facilitates tracking through smaller *active* search areas—at constant target motion bandwidth.

V. Experimental Validation

The current contribution addresses feature matching acceleration and its robustness to rapid camera motion. Older publications in Refs. [1], [24] already provided experimental validation of the vision-based ego-motion estimation algorithm and detailed its high estimation accuracy. These aspects remain unchanged here.

The robustness of the approach is demonstrated using the same challenging sequence as in Ref. [1], now *without* using synchronized inertial data. Fig. 6 displays a typical frame that shows both *active* features, the validation set, as well as remaining features. The authors suggest that the reader retrieves the processed video stream from the Internet at `www.robotic.dlr.de/Klaus.Strobl/icra2011`. Robust feature tracking holds during the entire sequence.

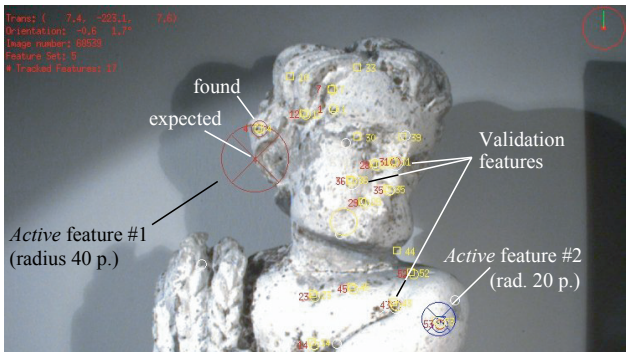

Fig. 6. Detail of a frame including two *active* features, three validation features, and a number of current and past regular features.

The accompanying video demonstrates real-time operation of the approach, concurrently with scanning and meshing of a scene. *All* calculations (includ. visualization) are performed on a single, Intel® Core™ 2 Duo P8700 processor notebook.

VI. Conclusions

Recently in Ref. [1] we presented the self-referenced DLR 3D-Modeler. It was the first hand-held 3-D modeling device for close-range applications that localizes itself passively from its own images in real-time, at high-rate. Since pose estimation here relies on accurate feature tracking, challenged feature tracking by rapid camera motion naturally compromises accurate pose estimation in real-time.

In Ref. [1] we introduced a novel approach to facilitate feature tracking in the particularly challenging case of rapid camera motion at close range, together with concurrent 3-D modeling. It includes information from an inertial measurement unit (IMU), synchronized and rigidly attached to the DLR 3D-Modeler.

In this work we achieve robust and efficient feature tracking in rapid, close-range motion *without* the use of inertial data. This is an important contribution, both in order to avoid calibration and synchronization issues of the IMU, as well as to further reduce hardware requirements for 3-D modeling. We believe it is precisely through flexible, passive, lighter, smaller, and more affordable sensors that machine perception will eventually enable the breakthrough of service robotics.

The current novel approach for tracking casts the regular KLT feature tracker into the Active Matching paradigm by fully using and updating full state estimations *during* the feature tracking process itself. In particular, a minimal set of features provides preliminary motion estimation that in turn enables fastest operation of the KLT feature tracker on all remaining fellow features. Separate treatment of feature drifts owing to camera translation and rotation makes it possible to use a minimal set of only two features.

Future work will focus on more efficient stereo initialization and image processing in general. In addition, a monocular implementation is being considered for constrained scenes, or scenarios where absolute scaling is not required.

REFERENCES

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

[2] Y. Aloimonos, "Active Vision Revisited," in *Active Perception*, Y. Aloimonos, Ed. Lawrence Erlbaum Associates, 1993, pp. 1–18.

[3] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco, CA, USA: W.H.Freeman & Co Ltd., 1983.

[4] D. H. Ballard, "Animate Vision," *Artificial Intelligence Journal*, no. 48, pp. 57–86, 1991.

[5] J. Schiehlen and E. D. Dickmanns, "Design and Control of a Camera Platform for Machine Vision," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Neubiberg, Germany, September 1994, pp. 2058–2063.

[6] 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.

[7] J. F. Seara, K. H. Strobl, E. Martin, and G. Schmidt, "Task-oriented and Situation-Dependent Gaze Control for Vision Guided Autonomous Walking," in *Proc. of the IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, Munich and Karlsruhe, Germany, Oct. 2003, pp. 1–23.

[8] A. J. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[9] A. J. Davison, "Active Search for Real-Time Vision," in *Proceedings of the International Conference on Computer Vision (ICCV)*, Nice, France, October 2005, pp. 66–73.

[10] M. Chli and A. J. Davison, "Active Matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Marseille, France, October 2008, pp. 72–85.

[11] ——, "Active Matching for Visual Tracking," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1173–1187, 2009.

[12] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Proc. of the Sixth IEEE and ACM Int. Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan, Nov. 2007.

[13] ——, "Improving the Agility of Keyframe-Based SLAM," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Marseille, France, October 2008, pp. 802–815.

[14] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 1, May 2006, pp. 430–443.

[15] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-Time Monocular Visual Odometry for On-Road Vehicles with 1-Point RANSAC," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, pp. 4293–4299.

[16] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-Point RANSAC for EKF-Based Structure from Motion," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 2009, pp. 3498–3504.

[17] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway, "Real-Time and Robust Monocular SLAM Using Predictive Multi-Resolution Descriptors," in *Proceedings of the 2nd International Symposium on Visual Computing (ISVC)*, Nov. 2006, pp. 276–285.

[18] ——, "Robust Real-Time Visual SLAM Using Scale Prediction and Exemplar Based Feature Description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, MN, USA, June 2007, pp. 1–7.

[19] E. Eade and T. Drummond, "Unified Loop Closing and Recovery for Real Time Monocular SLAM," in *Proceedings of the British Machine Vision Conference (BMVC)*, September 2008.

[20] C. Borst *et al.*, "Rollin' Justin - Mobile Platform with Variable Base," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, best video award.

[21] J. Shi and C. Tomasi, "Good Features to Track," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Jerusalem, Israel, October 1994, pp. 593–600.

[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] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker. Description of the Algorithm," Microprocessor Research Labs, Intel Corporation, Santa Clara, CA, USA, Tech. Rep., 2000.

[24] E. Mair, K. H. Strobl, M. Suppa, and D. Burschka, "Efficient Camera-Based Pose Estimation for Real-Time Applications," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 2009, pp. 2696–2703.

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

[26] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[27] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[28] V. Lepetit and P. Fua, "Keypoint Recognition Using Randomized Trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1465–1479, 2006.

[29] J. Neira and J. D. Tardós, "Data Association in Stochastic Mapping Using the Joint Compatibility Test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, December 2001.

[30] J. Solà, "Multi-Camera VSLAM: from Former Information Losses to Self-Calibration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, October 2007, in Workshop on visual SLAM.

[31] M. Kaess and F. Dellaert, "Covariance Recovery from a Square Root Information Matrix for Data Association," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1198–1210, 2009.

[32] J. A. Castellanos, J. D. Tardós, and G. Schmidt, "Building a Global Map of the Environment of a Mobile Robot: The Importance of Correlations," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Albuquerque, New Mexico, USA, April 1997, pp. 1053–1059.

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

[34] M. Chli and A. J. Davison, "Automatically and Efficiently Inferring the Hierarchical Structure of Visual Maps," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, pp. 387–394.