The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Efficient Camera-Based Pose Estimation for Real-Time Applications

Elmar Mair, Klaus H. Strobl, Michael Suppa and Darius Burschka

Abstract— **Accurate online localization is crucial for mobile robotics. In this paper, we describe a real-time image-based localization technique, which is based on a single calibrated camera. This can be supported by a second camera to improve accuracy and to provide the correct translational scale. Our goal is a robust and unbiased pose estimation in highly dynamic scenes on resource-limited systems. The presented approach is characterized through significantly improved robustness of the pose estimation, a novel approach for stereo subpixel accurate landmark initialization, and the speed-up of conventional tracking routines to achieve online capability. Although the algorithm is designed for accurate, online short-range egomotion estimation in hand-held scanning devices, it can be used for any mobile robot application as shown in this paper. Various tests and experimental results with a mobile platform and a hand-held 3D modeler are presented and discussed.**

I. MOTIVATION

Visual localization has become an engaging field in the recent years. Especially in mobile robotics, the advantages of optical sensors are evident. Compact, accurate, noninvasive and low-current cameras replace more and more complex laser or sonar sensors and conventional error-prone odometry. Several other advantages arise from using cameras: not only geometric but also textured maps can be built, and the biological insights about perception can be directly applied.

However, due to photometric effects and the loss of dimensionality in optical projection, it is not trivial to recover the 3D information from images. The 3D reconstruction process is time-consuming and difficult to implement on resource-limited computers like Embedded Systems. Especially, if the acquired scene is close to the camera, e.g. in hand-held scanning, preprocessing becomes more challenging. Tracked landmarks leave rapidly the field of view and stereo correspondences must be found continuously.

For highly dynamic systems, it is essential to obtain an accurate pose in real-time. Processing only the video stream of one camera reduces the computational load allowing online processing on resource-limited systems. There are two possibilities to retrieve 3D information from a monocular image stream: artificial markers or structure from motion (SFM) approach. On one hand, it is cumbersome to add landmarks to the environment. On the other hand, SFM approaches lack a true scaling factor. Using a second camera provides precise 3D landmark estimation by stereo triangulation. However, a continuous stereo matching is time-consuming and hence not online capable on resource-limited systems. A combination of monocular image processing where the tracked features are initialized by a second camera provides the solution matching both requirements: a precise pose estimation in real-time.

In this paper, we present an accurate real-time localization system based on a single, calibrated camera. The speed of single camera processing and the accuracy of subpixel stereo triangulation is achieved by combining monocular localization with a novel stereo-initialization step. No external referencing system is necessary, nor artificial markers are used. Further, the robustness of the pose estimation from tracked image features has been improved compared to the visual GPS (VGPS) approach presented in [1], [2]. An intelligent feature management robustifies the pose estimation additionally. The algorithm is designed for close range applications like hand-held 3D scanning, but allows arbitrary, highly dynamic mobile robots to estimate their motion online without any knowledge about their environment.

Fig. 1 shows two application scenarios where the presented system is used and which are discussed in more detail in Section IV-D and IV-E. In the left picture, the 3D modeling process is illustrated: the precise poses of the sensors have to be determined in order to allow a robust fusion of the acquired data. The right image shows a Pioneer 3-DX navigating around a table. The acquired camera frames are used to build an image-based environment model that requires an accurate pose estimation to merge the images properly.

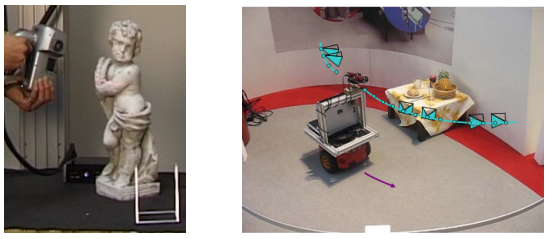


Fig. 1. *Left:* A hand-held 3D modeling system has to be localized globally in order to fuse the acquired data. *Right:* A Pioneer 3-DX watching a scene from different views to build a vision-based environment model.

The remainder of this paper is structured as follows. In the next section, we compare our work in context to related approaches. The algorithm is subdivided into several modules, which are described in Section III. In IV, we show some experimental results for the modules and for the whole algorithm. Furthermore, the two mentioned application sce-

This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems (CoTeSys)* and within the German Aerospace Center (DLR).

E. Mair and D. Burschka are with the Department of Informatics, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany
{elmar.mair,burschka}@cs.tum.edu

K. H. Strobl and M. Suppa are with the Institute for Robotics and Mechatronics, German Aerospace Center (DLR), Münchner Str. 20, 82234 Wessling, Germany
{klaus.strobl,michael.suppa}@dlr.de

narios are discussed in more detail.

II. RELATED WORK

In order to extract some exact 3D information from monocular image sequences, the camera positions have to be known. Several different approaches in this field already exist, which are based on the optical flow between two images. The most popular ones to estimate the sensor position by a calibrated camera are probably the 8- ([3]) and 5-point ([4]) algorithms. Nevertheless, several iterative methods as the 3-point algorithm [4] or vision-based GPS (VGPS) [1], [2] are well known, too.

The visual SLAM (V-SLAM) algorithms as localization technique are well known in literature. The solution provided by Davison consists of building a probabilistic 3D map with a sparse set of good landmarks to track [5], [6]. The points are used in an Extended Kalman Filter (EKF) for a repeatable localization with limited drift. However, due to size of the resulting vector, this is only real-time capable in restricted environments - e.g. the amount of landmarks in [6] has been limited to 100. A well known problem of monocular localization is the correlation of localization error and feature initialization error. If no loop-closure can be accomplished (e.g. due to a straight trajectory), the estimated error is accumulated and the accuracy decreases with increasing distance from the starting point. An inaccurate localization leads to a false landmark initialization, which again results in an inaccurate pose estimation.

By efficiently separating the tracking and the mapping routines, Klein was able to achieve even more accurate results as the EKF based approach of Davison [7]. Further, the method scales better and even if it is designed for small workspaces, it is also applicable for larger environments. The drawbacks are the high processing and memory requirements for the keyframes and the large number of features. Therefore, up to now this method is not applicable on resource-limited systems.

Thanh et al. describe a stereo SLAM method with two EKFs [8]. They use a combination of mono- and binocular feature estimation. At the beginning, the stereo-EKF is initialized by the simple odometry and then the stereo-matching is done with help of SLAM. The features are initialized and used for monocular pose estimation. This rather complex stereo algorithm improves the accuracy and robustness of conventional MonoSLAM clearly, especially for long range motion. However, stereo processing is time-consuming and, therefore, the reported framerate is only about 9 Hz.

Mourikis and Roumeliotis present a dual-layer localization architecture [9]. A combination of a Multi-State-Constraint Kalman Filter (MSC-KF) and a Bundle Adjustment provides highly accurate long-term visual pose estimation. In their work, they achieved a processing time of the MSC-KF at about 100 ms. However, the combination with an inertial measurement unit (IMU) leads to a robust long-range localization.

Several other V-SLAM approaches in literature are combined with probabilistic techniques, like e.g. Monte-Carlo localization [10] or particle filters [11]. The application area of such methods is a map based environment, where the map is known in advance. This does not apply to the constraints on our algorithm, which allows 3D modeling without any modifications in the environment.

III. REAL-TIME VISION-BASED LOCALIZATION

In order to provide input for a navigation control-loop of a mobile robot a high data rate is favored. The higher the data rate, the more dynamic can be the controlled system. The localization algorithm should, therefore, not limit the processing framerate. In the following, we call an image-based algorithm real-time or online-capable if its processing time permits the standard camera framerate of 25 Hz. Regarding 3D modeling systems, not only the *speed* but especially the *accuracy* of the estimated pose is crucial. Only this way, the acquired data can be reliably merged in a global model. Thus, it appears that the most challenging part is to bring together these two conflicting objectives.

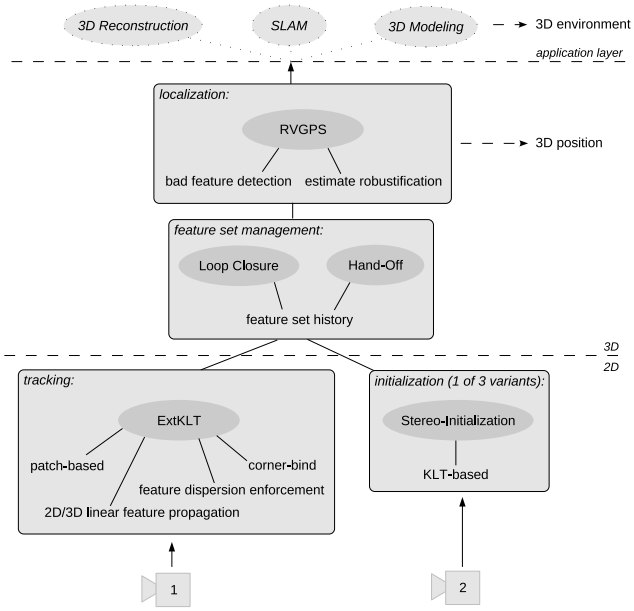


Fig. 2. Several modules are necessary for localization from images. The resulting positions can be used for an arbitrary application as 3D Modeling or SLAM. The 3D structure can be initialized in different ways - in this graphic only the stereo approach is listed.

The complexity of visual localization becomes first apparent in the cumbersome image preprocessing. Landmarks have to be extracted and tracked reliably over time. These image features have to be initialized and managed in an intelligent hand-off and loop closure. Based on the feature locations, the camera pose is estimated. Figure 2 illustrates the modules of the framework.

Dealing with highly dynamic systems, like a human's wrist or a flying robot, makes any time-based filtering (e.g. a Kalman filter) dangerous restrict the dynamics of the measurements. Renouncing such time-based filters prevents any restrictions in dynamics but demands at the same time

measurements with only a small jitter. We do not use any probabilistic methods to smooth the motion estimation over time but try to make the algorithm as reliable and robust as possible. Depending on the application, the results can nevertheless be used as input for any probabilistic framework, as e.g. Kalman or particle filters.

A. Monocular, extended KLT feature tracking

In order to estimate the motion from an image stream, the changes between the images have to be detected. Without any artificial markers and without any knowledge about the environment, this task is known to be not trivial [3]. First, good features to track must be selected and then these features have to be tracked from image to image.

Due to its speed and its robustness, the Kanade-Lucas-Tomasi (KLT) tracker ([12], [13]) fits best our requirements. Nevertheless, some improvements to the standard implementation of the KLT algorithm were necessary to provide a 25 Hz image processing in resource-limited environments:

- The KLT preprocessing step consists of smoothing the image and calculating its gradients. This means that three convolutions with different kernels need to be done. To speed up tracking, the preprocessing has been restricted to small patches around the tracked points. In this way, not the entire images but only those small patches have to be processed and stored. Therefore, also a processing of high resolution images becomes feasible.
- A linear motion model allows not only for larger feature displacements between the images, but also for smaller search areas. The results are fewer tracking iterations and a reduced size of the image-patches. The motion is modeled based on the 2D feature displacements in the image and provides so a strict separation of the tracking and the pose estimation routines which increases the robustness.
- It is often the case that a scene captured by a camera is built from different structured regions. Patterns with a rich contrast are preferred by local feature trackers like KLT, because they allow a good discrimination during tracking. Therefore, a well known fact for those trackers is that they often take only features within a small region of the image with a high-contrast pattern. Splitting the image into commensurate subimages for feature selection, leads to a better landmark dispersion. A feature set, which covers a wider cone of view allows a better conditioned pose estimation.

B. Subpixel-accurate structure initialization

To ensure real-time capabilities, this visual localization method is based on a monocular video-stream. It is only possible to estimate the translation up to scale Because of the loss dimensionality in the perspective projection without any external reference. However, there are three different ways to initialize the scale from images: by using the dimensions of some known objects (*Structure from Reference*), by moving a camera (*Structure from Motion*) or by using a stereo camera system with stereo triangulation (*Structure from Stereo*) [3]. Any of these three mentioned initialization techniques can be used at this point. The stereo alternative allows the most precise feature initialization and, therefore, this method is chosen. Indeed our approach leads to a subpixel-accurate result.

Conventional stereo matching by corresponding patches on the epipolar line is error-prone and results in pixel-accurate stereo matches. In practice, the epipolar line even grows to a small band due to calibration inaccuracies. Therefore, often feature matching routines like SURF [14] and SIFT [15] are used. These use the computational expensive Harris-Affine and Difference-of-Gaussian point detectors, to be able to deal also with affine transformations. However, in the case, where the cameras are mounted parallel on the stereo rig and the baseline is short, the affine component of the stereo-transformation can be neglected. This leads to the same assumptions of Shi and Tomasi in [13]. According to that, good features correspond to a matrix $\boldsymbol{Z}$ with large eigenvalues, whereas $\boldsymbol{Z}$ is defined as following integral over the window W

$$\begin{aligned} \boldsymbol{Z} &= \int\int_W \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}^T(\boldsymbol{x})w(\boldsymbol{x})d\boldsymbol{x}\,, \\ where \quad \boldsymbol{g} &= \left[\frac{\partial}{\partial x}\left(\frac{I+J}{2}\right) \quad \frac{\partial}{\partial y}\left(\frac{I+J}{2}\right)\right]^T. \end{aligned} \tag{1}$$

Further, $\boldsymbol{x} = [x, y]^T$ are the feature coordinates, I and J the two images and $w(\boldsymbol{x})$ an optional weighting function, e.g. for smoothing.

Good features to track are extracted from the main camera image. Next, even a higher number of features is acquired from the second image. This restricts the correspondence search from all pixels within the epipolar band to a few interesting points in it. These are now used as starting points for the KLT tracker, which aligns to subpixel accuracy. If the tracker finds more than one match, we use the one with the smallest pixel difference of the gradient patches. In case of a match, a subpixel-accurate feature correspondence is found.

Because the stereo matching takes depending on the number of features up to one second, it is not real-time capable. Therefore, the initialization of a new feature structure is done concurrently while the old feature set is continuously tracked.

Restricting the search range for stereo matches to the displacement corresponding to an object's distance, only features in that space are found. With this restriction, we can localize in respect to an object even if it is moved, because we do not refer to landmarks out of the specified range, like e.g. on the scene background. This allows tracking of a moving object and in the 3D modeling scenario the reconstruction of dynamic subjects.

With SURF or SIFT the accuracy is at best one pixel by the nature of the algorithm and they are in addition computationally more expensive. Some results of our technique are presented in section IV-B.

C. Sequential robustified VGPS - RVGPS

Independent of the tracker, its result can contain bad features due to occlusions, recurrence, virtual features and

reflections. Also the stereo initialization can provide false matches resulting an incorrect depth of the respective points. Therefore, a localization algorithm has to be used, which is able to detect bad features and reject them from processing and future tracking.

VGPS (vision-based GPS) is an image-based method for the self-estimation of camera poses [1]. The method assumes a reference image S_0 and thereby solves the relative orientation problem of determining the pose of the camera S_t with respect to S_0. Therefore, it requires an internal 3D model (a set of n points ${}^0P_i,\ i \in \{1..n\}$, in the scene) attached to S_0. This model can be constructed in an arbitrary way (e.g. stereo triangulation, see III-B). The exterior orientation between the current frame and the reference 3D model is computed as follows: an additional tentative 3D model ${}^t\widehat{P}_i$ is generated from the 2D projections in the images by using approximated ranges only. These ranges are estimated from the preceding pose estimation in $t-1$. Thus, the problem of determining the absolute orientation is reduced to finding the relative orientation between these two sets of points ${}^0P_i$ and ${}^t\widehat{P}_i$. This can be solved in closed form using the singular value decomposition (SVD).

In a nutshell: Relative translation and rotation are estimated separately. We first set the origins of the sets of points to their respective centroids without modifying their orientations which yields the sets ${}^0P'_i$ and ${}^t\widehat{P}'_i$. The relative rotation between these sets of points of the same model corresponds to the relative rotation between camera reference frames and can be calculated by maximizing the trace of the inertia matrix of the matched set:

$$ {}^t\boldsymbol{R}^* = \arg\max_{\boldsymbol{R}} \operatorname{trace}({}^t\boldsymbol{R}^T\, {}^t\boldsymbol{M}) \ ,\ {}^t\boldsymbol{M} = \sum_{i=1}^{n} {}^t\widehat{P}'_i\ {}^0P_i'^T. \quad (2) $$

Let $(\boldsymbol{U},\, \boldsymbol{\sigma},\, \boldsymbol{V})$ be the SVD of ${}^t\boldsymbol{M}$, that is $\boldsymbol{U}^T\boldsymbol{\sigma}\boldsymbol{V} = {}^t\boldsymbol{M}$, then the solution to Eq. (2) is

$$ {}^t\boldsymbol{R}^* = \boldsymbol{V}\boldsymbol{U}^T \quad (3) $$

and the translation can be also found as follows:

$$ {}^tT^* = \frac{1}{n}\sum_{i=1}^{n} {}^t\widehat{P}_i - {}^t\boldsymbol{R}^*\,\frac{1}{n}\sum_{i=1}^{n} {}^0P_i \ . \quad (4) $$

Since the tentative 3D model ${}^t\widehat{P}_i$ may differ from the reference one, the final solution is found iteratively by optimizing the quality (the unknown ranges) of the tentative model at the same time. The algorithm terminates whenever sufficient consistency with the original set of points is achieved.

The specifics of our particular VGPS approach are as follows:

- VGPS is here sequentially applied to different reference frames. This is determined by a higher level decision making process (see section III-D).
- The accurate internal model ${}^0P_i$ used by VGPS is in our implementation obtained by stereo-vision at the correspondent reference frame S_0 (see section III-B). Due to the high accuracy of this initialization the acquired model is not updated anymore. Nevertheless, in the next step weights are applied to each feature to rate their quality and to detect outliers.
- The novel solution to the absolute orientation problem within the VGPS algorithm makes use of a redescending M-estimator on the residual Euclidean distances between matched points. This is done to disregard gross outliers without compromising the estimation convergence because of the naturally noisy nature of the problem. Large outliers may correspond to either a faulty internal 3D model, false matching correspondences, virtual features (e.g. features from occlusions), or, more infrequently, to wrong tentative ranges and are fatal to unbiased pose estimation. Therefore, in case of outliers, the robustified VGPS not only disregards this data but also sends a signal to the features database in order to remove those features. In particular, we use the biweight function of Tukey because of its continuous derivatives and its handy weights. Both, the initial estimation and the chosen scale, are much more influential parameters to global fast convergence than the nature of the employed function [16]. Thus, the modification concerns weighting the contribution of each point to the inertia matrix of the matched set of points with the weight

$$ \begin{array}{lcll} {}^tw_i & \propto & (1 - {}^tS_i \cdot {}^tS_i)^2 & \text{if } |{}^tS_i| < 1 \\ {}^tw_i & = & 0 & \text{if } |{}^tS_i| \geq 1 \end{array} \quad (5) $$

  where ${}^tS_i = ({}^t\boldsymbol{R}\,{}^0P_i - {}^t\widehat{P}_i)/s$ is the estimated normalized matching residual for object point i at instant t before performing the SVD and s is the scale of the inlier noise. In the end:

$$ {}^t\boldsymbol{R}^* = \arg\max_{\boldsymbol{R}} \operatorname{trace}({}^t\boldsymbol{R}^T {}^t\boldsymbol{M}^R) \ ,\ {}^t\boldsymbol{M}^R = \sum_{i=1}^{n} {}^tw_i\, {}^t\widehat{P}'_i\ {}^0P_i'^T. \quad (6) $$

- Finally, we use an efficient termination policy determined by a threshold on the absolute orientation correction over the course of the iterations.

The advantage by using M-estimators and not a simple outlier rejection method like RANSAC is that each measurement can be weighted according to its accuracy contribution. Some test results in section IV-C depict the improvements by using M-estimators and compare the RVGPS to the conventional VGPS algorithm.

D. Feature set management

Whenever the camera is moved, the landmarks can leave its field of view. If not enough features are trackable, we concurrently initialize a new feature set and save its offset to the origin. The motion during the feature initialization is estimated based on the old features and is used to propagate the position of the new ones in the current image. Thus, a wrong initialization, e.g. because of false correspondences or occlusions, leads to a false propagation and, therefore, to a feature's loss - a first bad feature detection is provided. The initialized feature sets are stored in a database.

The accuracy of the pose estimation is not only based on the amount of landmarks, but also where they are located. Like any image-based pose estimation algorithm, RVGPS is also ill-conditioned if all features lie within a small area of the image and especially if they are all at the same image border. To detect such situations, we calculate the centroid of the 3D features and its projection on the image plane. If it leaves a specified inner area of the image a new feature set initialization is triggered, irrespective of the number of features still trackable.

Further, the image-plane projection of all centroids in the database is used to detect the most central feature set and, for the case it does not correspond to the active one, to trigger a switch-over to the most central landmark set. The accumulated error becomes reduced to the value after initializing that old set. Thus, the feature-set hand-off and short range loop closure policy reads as: A new set is acquired each time the number of trackable features drops below a specific threshold or the projection of the features' centroid is outside the image center. Further, if an old feature set seems to allow a better pose estimation, we attempt to use that one for tracking.

Figure 3 depicts the different steps and branches within the presented framework.

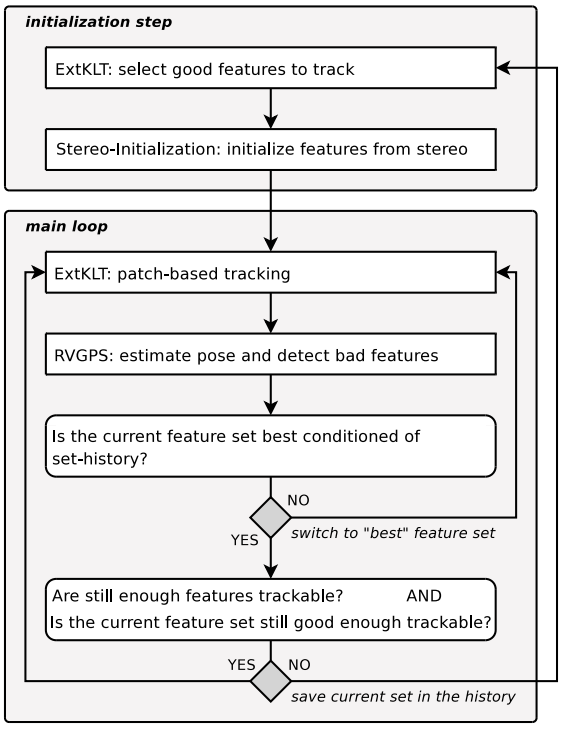


Fig. 3. Simplified flow chart of the image-based localization routine.

IV. EXPERIMENTS

In this section, we present some results of experiments regarding the innovations of the presented algorithm. Further, some application results are discussed.

A. KLT extension

The speed up by the patch based implementation of the KLT tracker carries more weight if less features are tracked within a small search range on a large image. Due to the linear feature propagation this search range can be kept quite small. Fig. 4 shows some experimental results for the standard, the patch-based KLT implementation and a method which uses regions of interest (ROIs) to reduce processing. The latter approach is always less efficient than the whole-image- or patch-based variant and thus it can be neglected. If there are too many features to track, the overlapping areas of the patches produce a larger amount of pixels to be processed and the patch based alternative becomes less efficient. The preferred alternative depends, therefore, on the feature search range in each direction and the number of features. In our applications we use between 6 and 25 features for tracking, which has proven to be sufficient for reliable and robust pose estimation. Therefore, in our case the patch based alternative allows a significant speed up.

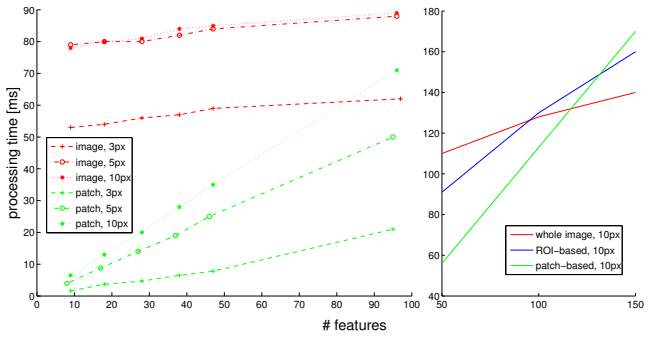
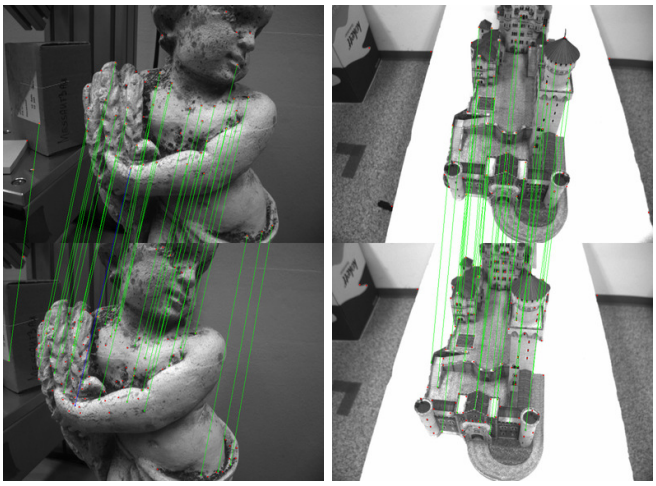


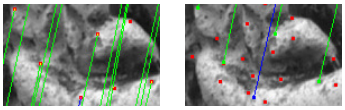
Fig. 4. These figures compare the processing times of the KLT-variants for different number of features to track. The red values are representing the whole-image-based implementation, the blue values the ROI-based version and green are the times for the patch-based variant. On the left, different sizes for the search window are chosen. While the processing time for the whole image almost only depends on the number of pyramid-layers (3px search range results in one, 5px and 10px in two pyramid layers), the patch-based approach is also related to the search range and the number of features. The right image shows the intersection of the lines for a 10px search range.

B. Stereo initialization

The results of the fast and subpixel-precise stereo initialization method (see Section III-B) are depicted in Fig. 5. The lower small images are regions of the left scene and show in detail two corresponding parts of the large pictures. Thus, the reader can see, that the right correspondences are found. The red dots in both shots are the extracted points of interest which are used as matching candidates. The green lines link up corresponding features of the left camera (upper image) and the right one (lower image). The green dots at the end of the lines in the lower image are the sub-pixel accurate correspondences. Blue lines are matches, where the KLT tracker could not find a minimum due to the limited iteration number. Nevertheless, experiments have shown, that these correspondences are mostly also acceptable accurate (as it can also be seen in the small figures).



(a) The KLT based stereo initialization.



(b) Details of the putto-scene above.

Fig. 5. The KLT based stereo initialization allows fast subpixel-accurate stereo matching. The lower small images are parts of the putto-scene and show some details of the upper image (left camera) and the lower image (right camera).

C. Accuracy

Next, we apply the method presented in this work to a stereo camera[1] which is mounted on the wrist of a robotic manipulator *KUKA KR16*. The results of our positioning system are then compared to the accurate output of the robot's kinematics. The experiment consists in a motion round the object to be modeled (see Fig. 7). The images and the poses of the manipulator are acquired synchronously [18]. There is a total amount of 710 images (see Fig. 6), the round-trip has a total length of $125\,cm$, and the orientation changes arbitrarily up to $55\,^\circ$.[2] We run the experiment twice, whereas the only difference was that the VGPS robustification has been enabled resp. disabled.

Concerning the accuracy analysis: Figs. 8 and 9 show the residual errors in rotation and translation with respect to the data from the *KUKA* manipulator, which relative positioning accuracy is $0.1\,mm$ and in orientation less than $0.1\,^\circ$. The blue dots show the results using RVGPS, while the pink dots represent the output with conventional VGPS.

On the RVGPS performance: The orientation error increases in the course of the experiment up to $0.5\,^\circ$ at the turning point, $65\,cm$ away from the origin. This is because until then the method performs dead-reckoning and accumulates orientation errors after each initialization moment. The accuracy suffices for several modeling applications.

[1]The intrinsic stereo camera calibration and extrinsic positioning with respect to the manipulator was performed using the camera calibration toolbox *DLR CalDe and DLR CalLab* [17].

[2]The video is available at http://www6.in.tum.de/~maire/videos/visualLocalization_castle.mp4

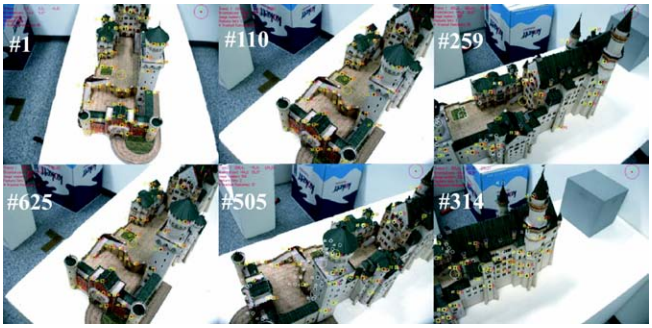


Fig. 6. The motion is from the front (image #1) to the back (image #350) of the castle, and then back to the front in image #710.

After that, on its way back, the field of view heads back to the former structures, being able to find them again and in this way to get rid of the accumulated orientation error. The positioning error is represented in Fig. 9. In the comparison the inaccuracies of our algorithm, due to drifting features and wrong correspondences, add up together with the inaccuracies of the manipulator. At the turning point the mean accuracy surpasses $4\,mm$, but the errors lower after that. In this example the positioning bias is not completely canceled either because of mechanical hysteresis or residual tracking drifts. No Kalman- or particle-filtering is used to prevent any restrictions to the system dynamics.

Comparing both runs: The rotation estimation without using the robustification as described in section III-C is almost the same as with M-estimators. The variation is rather random and would not bear an improvement. However, the difference is visible watching the translation estimations. Here, the pink dots accumulate an error up to more than $20\,mm$. This result can be explained by the characteristics of such an iterative estimation method, like it is used in VGPS: Irrespective of the kind of error, whether it occurred at the feature initialization step or during tracking, the sum of all errors can always be lessened by zooming out the camera. Thus, it is obvious that a gradient descent method follows the gradient on the optical axis of the camera to diminish the error, while disregarding the orientation estimation. Of course, special error combinations could also lead to a wrong rotation estimation. Although, the translational error in the VGPS-run is quite large, the old features are refound, so that also there the error becomes reduced.

D. Application as a hand-held 3D modeler

The *DLR 3D-Modeler* in Fig. IV-D is a multi-sensory compact device for 3D modeling [19]. A strong requirement for modeling is the necessity for accurate pose measurement of the sensor in a fixed common reference frame. Currently, this has been achieved by either (see Fig. 10)

- an external tracking system, e.g. an infrared light-emitting stereo camera rig which tracks the reflecting markers on the modeler platform, or by
- a robotic manipulator, either active or passive.

By using our image-based navigation algorithm, the operation space of the modeler is neither restricted by the robot

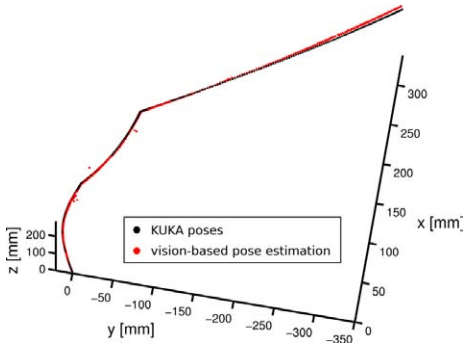


Fig. 7. The two trajectories are compared to each other: black the result of the inverse kinematics of the KUKA robot, red the poses estimated by the presented vision-based algorithm.

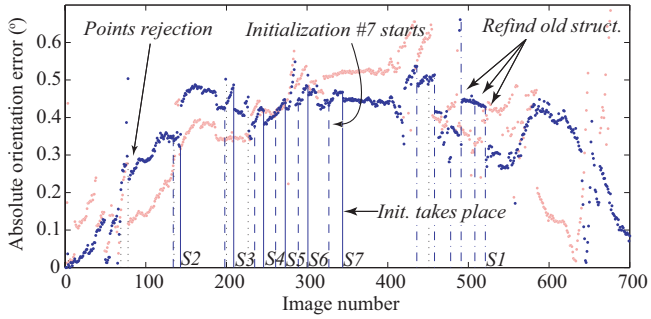


Fig. 8. Residual orientation error with respect to the *KUKA* manipulator. Blue: using RVGPS. Pink: using VGPS.

workspace nor by an external tracking device (see Fig. 11). A further advantage is that the error of the visual navigation algorithm and the laser-stripe profiler correlate, because both methods are based on the same image data.

Fig. 12(a) and 12(b) show modeling results acquired by a two-turn sweep of the DLR 3D-Modeler. Because the device is manually guided, the points are not uniformly distributed on the objects surface.[3] More details to the functionality and usability of this system are depicted in [20]

[3]An example of the operation is available at http://www6.in.tum.de/~maire/ videos/3dMo_putto.mp4

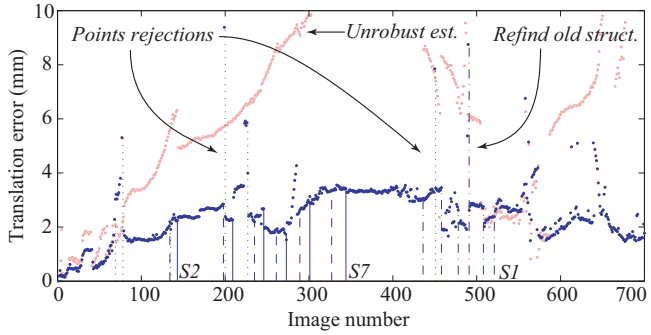


Fig. 9. Residual translation error with respect to the *KUKA* manipulator. Blue: using RVGPS. Pink: using VGPS - the error rises up to more than 20 mm, which is not visible to preserve an adequate scale.

Fig. 10. The upper left picture shows the DLR 3D-Modeler and the PC running the software (an Intel® core duo T2050 with 1.6GHz and 2GB RAM). In the past two external reference systems have been used to estimate the pose of the scanner: a robot arm (lower left), where the joint positions are used to calculate the modeler pose (robot kinematics), and an external tracking system (right), where six spheres attached to the modeler are tracked by stereo cameras.

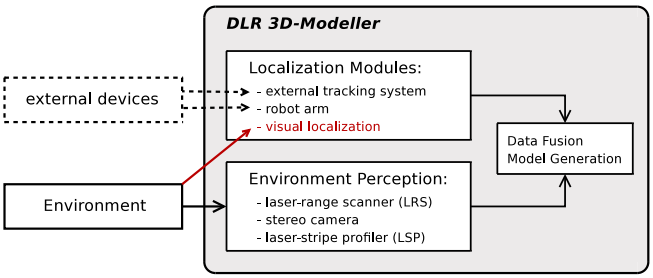


Fig. 11. This figure illustrates the modules provided within the DLR 3D-Modeler. Due to the visual localization module, there is no external device necessary anymore.

E. Application on a Pioneer 3-DX

Like on most mobile platforms the odometry of the Pioneer 3-DX is erroneous. It is usually not sufficient to build a model of the environment from acquired data. We use the raw results of the described localization system without any Kalman- or particle-filter to register the camera images in real-time. The achieved accuracy allows to render a virtual camera image from the exactly registered images of the database. For that application it was not possible to determine the ground truth to compare our results with. Nevertheless, the fact, that we are able to merge the images

(a) 3D point-cloud: statue in front of a box. (b) 3D point-cloud of a putto.

Fig. 12. Two 3D point clouds acquired by the DLR 3D-Modeler using the presented visual navigation algorithm.

properly, even without any bundle adjustment, is a proof for accuracy. Fig. 13 illustrates the localization results on an image sequence acquired while performing a quarter circle with 0.9 m diameter (see Fig. 1). For further details about this work please refer to [21].

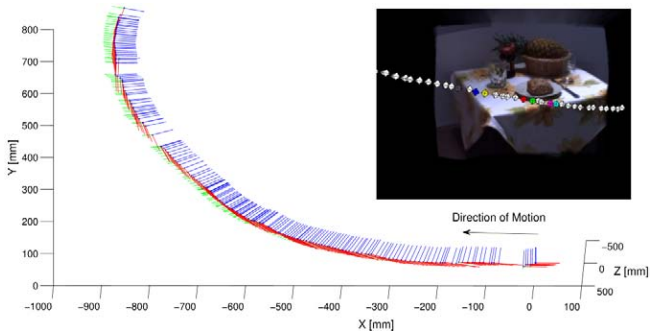


Fig. 13. Localization results of a Pioneer 3-DX performing a quarter circle as illustrated in the small subfigure. The blue lines show the looking direction of the capturing camera. The white gaps in the trajectory are due to swapping on the hard disk. The cameras were slightly tilted to the floor and so the trajectory is not only within the X-Z plane. The small picture in the upper right corner shows the virtually rendered model of the scene (see also Fig. 1).

V. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an accurate and online-capable visual navigation technique for resource-limited systems. The algorithm is based on KLT features tracked in a monocular image sequence. The KLT routines are sped up to allow a 25 Hz camera framerate also with limited computing resources. For stereo-initialization of the landmarks a novel fast and subpixel-accurate approach for stereo matching has been presented. The pose estimation algorithm is based on VGPS, but it has been robustified in order to detect bad features and exclude them from further processing. An intelligent hand-off and short range loop closure provides only a small error accumulation over longer terms. We have shown, that the robustification of VGPS keeps the error drift small enough to allow short range loop closure without Kalman or particle filter. By avoiding such filters the system's dynamics is not restricted which makes the algorithm applicable for high dynamic applications. Experiments provide proof of the framework's properties and illustrate the enhancements to the used algorithms. Also some results with a hand-held 3D modeler are presented. The algorithm allows unrestricted image-based online 3D-modeling without external referencing systems.

A problem of any hand-held modeler are the short, large rotational movements of the humans wrist. Such movements lead to large feature displacements between two consecutive images which prevents proper tracking. To overcome that problem, we currently test a combination with an IMU. Summing up the IMU data should ensure a better feature propagation. First results have proven that tracking can be done also during a wiggly sensor guidance.

VI. ACKNOWLEDGMENTS

We want to acknowledge the great support by the members of the 3D-Modeler project at the German Aerospace Center (DLR): Tim Bodenmüller, Wolfgang Sepp, Simon Kielhöfer and especially Gerd Hirzinger, head of the Institute for Robotics and Mechatronics at the DLR.

Special thanks also to Werner Maier for the virtual rendering results of the Pioneer experiment.

REFERENCES

[1] D. Burschka and G. D. Hager. V-GPS - image-based control for 3d guidance systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1789–1795, 2003.

[2] D. Burschka and G. D. Hager. V-GPS(SLAM): Vision-based inertial system for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 409–415, 2004.

[3] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition, 2003.

[4] D. Nister. A minimal solution to the generalised 3-point pose problem. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[5] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision*, volume 2, pages 1403–1412, 2003.

[6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29:1052–1067, 2007.

[7] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, 2007.

[8] T. N. Thanh, Y. Sakaguchi, H. Nagahara, and M. Yachida. Stereo slam using two estimators. In *IEEE International Conference on Robotics and Biomimetics*, pages 19–24, 2006.

[9] A. I. Mourikis and S. I. Roumeliotis. A dual-layer estimator architecture for long-term localization. In *Workshop on Visual Localization for Mobile Platforms, IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[10] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization by combining an image retrieval system with monte carlo localization. *IEEE Transactions on Robotics*, 21:208–216, 2005.

[11] R. Sim, P. Elinas, M. Griffin, and J. J. Little. Vision-based SLAM using the Rao-Blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 9–16, Edinburgh, Scotland, 2005.

[12] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[13] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[14] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.

[15] D. G. Lowe. Object recognition from local scale-invariant features. pages 1150–1157, 1999.

[16] P. J. Huber. *Robust Statistical Procedures*. SIAM, 2nd edition, 1996.

[17] K. H. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. DLR CalLab and DLR CalDe - http://www.robotic.dlr.de/callab/.

[18] T. Bodenmüller, W. Sepp, M. Suppa, and G. Hirzinger. Tackling multisensory 3d data acquisition and fusion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2180–2185, 2007.

[19] M. Suppa, S. Kielhöfer, J. Langwald, F. Hacker, K. Strobl, and G. Hirzinger. The 3d-modeller: A multi-purpose vision platform. In *IEEE International Conference on Robotics and Automation*, 2007.

[20] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[21] W. Meier, E. Mair, D. Burschka, and E. Eckehard. Visual homing and surprise detection in cognitive mobile robots using image-based environment representations. In *IEEE International Conference on Robotics and Automation*, 2009.