 DOI: 10.3217/978-3-85125-524-9-04

Visual Localization System for Agricultural Vehicles in GPS-Obstructed Environments*

Stefan Gadringer[1], Christoph Stöger[1] and Florian Hammer[2]

***Abstract*— Accurate outdoor localization and orientation determination using the Global Positioning System (GPS) usually works well as long as the GPS antenna receives signals from a sufficient number of satellites. Especially in agricultural applications, the respective lines of sight are frequently obstructed due to the presence of trees. In this paper, we investigate the applicability of an alternative method for position and orientation estimation that is based on a stereo-camera system and Visual Odometry (VO). We have experimentally validated our approach in a logging road scenario. Based on the results of the position and orientation estimation, we discuss challenges of VO in such a non-trivial environment.**

I. Introduction

Localization of a vehicle is a very important task and hence a research topic for decades. In general, localization is possible with sensors like GPS, rotary encoder, IMU (Inertial Measurement Unit), laser scanner or a camera. Of course, there exist even more sensors and each one has its own pros and cons in terms of accuracy, drift, price, etc. The area of application highly depends on these properties. In this paper, we focus on outdoor localization in natural terrain. This is an important topic for precision farming [4], for example. Hereby, the question is always the same: Which sensors are suitable for the application?

As discussed in [25], a GPS antenna always needs intervisibility to several satellites to guarantee an accurate position estimation. This is sometimes impossible in areas like in a forest where trees occlude the satellites. The usage of wheel odometry via rotary encoders is not suitable as well due to problems with inaccuracies of the wheel geometry and slipping situations. In comparison, an IMU allows a good estimation of the orientation but not for the position because the double integration of the acceleration results in a high drift over time. A laser scanner has a very high position accuracy on the one hand but it is very expensive and not so well proofed for high vibrations on the other hand. Thus, just the camera remains of the sensors mentioned above. This sensor is relatively cheap but a position and orientation estimation via VO is normally linked with high computing demand and continuous growth of the drift per number of used images. Furthermore, overexposed images and other problems like branches that occlude cameras need a robust implementation of a VO to be able to get a valid pose estimation. However, this paper shall show the applicability of Visual Odometry to estimate position and orientation in different wooden environments with ambiguous natural structures.

This paper is structured as follows. Section II gives an overview of related work. Visual Odometry and all its components are explained in Section III. Finally, the experiments are shown in Section IV. Last but not least, Section V contains the conclusion as well as some remarks about future work.

II. Related work

Visual Odometry (VO) is the incremental estimation of the pose (position & orientation) via examination of the changes on images due to motion induction [24]. The research on VO already started in the early 1980s and one if its advantage is that no prior knowledge about the environment is necessary. A good example is the implementation of Cheng et al. [6], [21], which was used in the rover of the NASA Mars exploration program. Since then VO was continuously under research, which means that the literature about Visual Odometry is huge. Therefore, this section just contains an overview about relevant literature of VO for the localization of a vehicle in an outdoor environment.

Nister et al. [22] proposed one of the first real-time VO which was capable of a robust pose estimation over a long track. They use a stereo-camera system and detect Harris corner features [15] in the images. 3D points are estimated through triangulation of the corresponding features in a stereo pair. In a next step Nister et al. use these 3D points and the features of a following image to estimate the pose via a 3D-to-2D algorithm as described in [24]. RANSAC (Random Sample Consensus) [12] removes outliers in the motion estimation step. Regarding to Scaramuzza et al. [24], this VO procedure was a high improvement to previous implementations and is still used by many researcher.

Comport et al. [7] use a similar procedure but estimate the motion using 2D-to-2D instead of 3D-to-2D feature correspondences. With reference to Scaramuzza et al. this results in a more accurate pose because triangulation is not needed.

In [26], [17] or [27] bundle adjustment is applied to further reduce the drift of the Visual Odometry. Bundle adjustment optimizes the latest estimated poses using features over more than just two stereo pairs. Konolige et al. [17] show that this step reduces the final position error about a factor of two to five.

*Parts of this work have been supported by the Austrian COMET-K2 programme of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government, and the federal state of Upper Austria.

[1]Stefan Gadringer and Christoph Stöger are with the Institute of Robotics, Johannes Kepler University, 4040 Linz, Austria {stefan.gadringer,christoph.stoeger}@jku.at

[2]Florian Hammer is with the Linz Center of Mechatronics GmbH, 4040 Linz, Austria florian.hammer@lcm.at

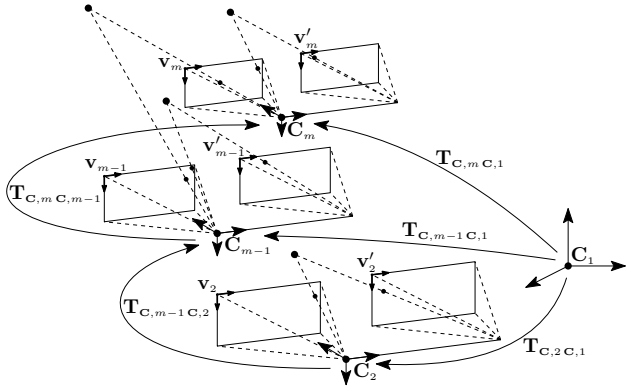


Fig. 1: Illustrated VO problem of a stereo system (relative transformations $\mathbf{T}_{\mathbf{C},m-1\,\mathbf{C},2}$, $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1}$ / absolute transformations $\mathbf{T}_{\mathbf{C},2\,\mathbf{C},1}$, $\mathbf{T}_{\mathbf{C},m-1\,\mathbf{C},1}$, $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},1}$)

Furthermore, the usage of additional sensors like GPS, laser scanner or IMU can improve the pose estimation. For example, in [1], [23], [17] or in [27] the integration of an IMU reduces the error in orientation. In [17] Konolige et al. achieve with their implemented real-time VO a maximum relative position error of just 0.1 % over a 9 km long track. Another good result is shown by Tardif et al. [27] over a 5.6 km long track. This dataset was acquired by a tractor driving next to an orange grove and on a street for the return to the garage.

III. Visual Odometry

As discussed in Section II, Visual Odometry incrementally estimates the pose. Figure 1 shows this for a typical case using a stereo-camera system. The calculation of a relative homogeneous transformation $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1} \in \mathrm{SE}(3)$ of an image pair $\{m-1,m\}$ with camera centers / camera coordinate systems $\mathbf{C}_{m-1}$ and $\mathbf{C}_m$ is done via features in the images. As shown in the figure, the coordinate system of the left camera is the reference point of a transformation $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1}$, which transforms from $\mathbf{C}_{m-1}$ to $\mathbf{C}_m$. The rigid body transformation is given by

$$\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1} = \begin{bmatrix} \mathbf{R}_{\mathbf{C},m\,\mathbf{C},m-1} & {}_{\mathbf{C},m}\mathbf{t}_{\mathbf{C},m\,\mathbf{C},m-1} \\ 0 & 1 \end{bmatrix} \tag{1}$$

where $\mathbf{R}_{\mathbf{C},m\,\mathbf{C},m-1} \in \mathrm{SO}(3)$ is the orthogonal rotation matrix and ${}_{\mathbf{C},m}\mathbf{t}_{\mathbf{C},m\,\mathbf{C},m-1} \in \mathbb{R}^3$ the translation vector, represented in the coordinate system $\mathbf{C}_m$. The concatenation of all relative transformations results in the absolute transformation $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},1} = \mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1}\mathbf{T}_{\mathbf{C},m-1\,\mathbf{C},1}$ from $\mathbf{C}_1$ to $\mathbf{C}_m$.

Therefore, the main task of a VO is to calculate the relative transformations $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1}$ and finally to concatenate them to get the full camera trajectory $\mathbf{T}_{\mathbf{C},m:\mathbf{C},1} = \{\mathbf{T}_{\mathbf{C},2\,\mathbf{C},1},\ldots,\mathbf{T}_{\mathbf{C},m\,\mathbf{C},1}\}$ between the camera centers $\mathbf{C}_1$ and $\mathbf{C}_m$.

The structure of our VO approach is similar to the one of Nister et al. [22] and it starts with the feature detection and description but it uses the more distinct features A-KAZE [2] instead of Harris [15]. The next step is to match features between a stereo pair and one consecutive image, either left or right. Then, the triangulated stereo correspondences and the matched $2D$ features are used for the pose estimation. At the end, key frames are selected and windowed bundle adjustment [28] is applied to further optimize the previous calculated poses [27].

A. Feature Detection and Description

Feature detection is one of the most important steps in a feature-based Visual Odometry system. Regarding to Fraundorfer [13], important properties of features are detection repeatability, localization accuracy, robustness against noise as well as computation efficiency. In [8], Cordes et al. compare many different detection algorithms and the detector A-KAZE [2] proofs to be the best candidate in terms of localization accuracy and suitable number of detected features. This detector is implemented in OpenCV [5] and is an extension of the algorithm KAZE [3] to detect blobs. In general, these features are image patterns with different intensity, color and texture compared to its adjacent pixels and they are more distinctive than corners [13]. This is especially important in natural environment with ambiguous structures like branches or leaves. In our case, A-KAZE detects blobs in a nonlinear scale space with four octaves and the same amount of sub-levels.

In addition to the detection algorithm, A-KAZE also provides one for the description of a feature, which is implemented in OpenCV as well. It converts the area around a feature into a binary descriptor which has a length of 486 bit. Every comparison between two areas results in three bit. The description algorithm of A-KAZE is called M-LDB (Modified-Local Difference Binary) and is rotation and scale invariant. According to Alcantarilla et al., A-KAZE allows efficient and successful feature matching, which are mandatory properties of a good descriptor.

B. Feature Matching

The task of this step is to find feature correspondences among images. The easiest way to achieve matching between two images is to compare all feature descriptors of the first image with every other descriptor of the second one. This search is quadratic in the number of features. Fortunately, the usage of epipolar or motion constraints simplifies this task and reduces the computation time drastically. This is necessary to facilitate an online VO system, which could be used on a vehicle like a tractor during its operation in a field or forest.

Our stereo VO relies on rectified images, which are remapped image pairs with horizontal and aligned epipolar lines to each other (see [13]). Thus, epipolar matching just allows a match between features which lie on the same horizontal epipolar line or rather image row.

Descriptors of two consecutive left or right images can be matched via a motion constraint. As proposed in [10], we assume a constant velocity model between two frames. Using the known motion, we can project the $3D$ point of a already matched stereo correspondence into the other image. A constant window of $2\cdot 35 \times 2\cdot 35$ pixel around the projected position defines the allowed area of possible features and therefore reduces the computing time.

The comparison between two binary descriptors itself is done via calculating the Hamming distance [14], which is the number of different bits and a very efficient operation. Normally, the descriptor with the minimum Hamming distance is chosen as the best match. To improve the robustness of the matching, we additionally apply the distance-ratio-test as proposed in [20]. It just accepts a match if the ratio between the two closest neighbors is below a threshold $r_{max} \in \mathbb{R}$ with $0 < r_{max} < 1$. Using binary descriptors, the ratio $r_H \in \mathbb{R}$ between two descriptors is defined as

$$r_H = \frac{d_{H,1}}{d_{H,2}} < r_{max}, \tag{2}$$

where $d_{H,1} \in \mathbb{N}$ and $d_{H,2} \in \mathbb{N}$ are the Hamming distances of the two closest neighbors, respectively. In our case, we use an empirical threshold of $r_{max} = 0.71$ which helps to remove ambiguous matches that can occur at repeatable structures like branches.

C. Motion Estimation and Key Frame Selection

In this step, the calculation of the relative camera motion, i.e. the relative transformation $\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}$ between an image pair $\{m-1,m\}$, takes place. Therefore, we use calibrated stereo-cameras and two sets of corresponding features F_{m-1} and F_m of the images $m-1$ and m, respectively.

For the 3*D*-to-2*D* algorithm, the features of F_{m-1} are defined by 3*D* points in $\mathbf{C}_{m-1}$ and the one of F_m by 2*D* image points [24]. Normally, we use 2*D* features of the left image with coordinate system $\mathbf{v}_m$. Alternatively, if the motion estimation fails due to less feature matches, features of the right image with coordinate system $\mathbf{v}'_m$ can also be used to prevent a failure of the VO. The estimation of the 3*D* points is done via the linear triangulation method of Hartley and Zissermann [16], which is implemented in OpenCV [5]. Using a function $\mathrm{d_E}$ to calculate the Euclidean distance [11], the transformation $\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}$ can be found through minimizing the image reprojection error of all features

$$\min_{\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}} \sum_{i=1}^{n} \mathrm{d_E}\left({}_{\mathbf{v},m}\mathbf{t}_{\mathbf{v},m\mathbf{X},i},\, {}_{\mathbf{v},m}\hat{\mathbf{t}}_{\mathbf{v},m\mathbf{X},i}(\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1})\right)^2. \tag{3}$$

Thereby, ${}_{\mathbf{v},m}\mathbf{t}_{\mathbf{v},m\mathbf{X},i}$ is the 2*D* coordinate vector of the image point $\mathbf{x}_i$ and ${}_{\mathbf{v},m}\hat{\mathbf{t}}_{\mathbf{v},m\mathbf{X},i}$ the image coordinate vector of the 3*D* point $\mathbf{X}_i$, which is observed in $\mathbf{C}_{m-1}$ and projected through $\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}$ and the corresponding camera projection matrix [16] into image m. Equation (3) can be solved using at least three 3*D*-to-2*D* correspondences, is known as P3P (Perspective from three Points) and returns four solutions. Therefore, at least one another point is necessary to get a single and distinct solution. PnP-algorithms (Perspective from n Points) like EPnP (Efficient PnP) [18] use $n \geq 3$ correspondences to solve the problem. Normally, these methods just calculate accurate results if the used correspondences are correct. If this is not guaranteed, the well known procedure RANSAC (Random Sample Consensus) [12] should be used to remove wrong correspondences, so called outliers. In [13], such a robust motion estimation using RANSAC is explained more in detail. Our VO uses EPnP for the pose estimation and a preliminary non-minimal RANSAC with five points to acquire trustworthy results of the outlier removal as suggested by Fraundorfer et al. [13].

If the first motion estimation with the left image fails due to less feature matches, or the motion is implausible (position or orientation is unrealistic), then the estimation is retried with 2*D* features of another image as a backup. The order of these images is the following. Firstly, the right image of the actual stereo frame is used. If the motion estimation with the features of this image is also unsuccessful, then a consecutive still unused left or right image is used until the motion estimation step is successful. This procedure avoids a failure of the VO with high probability.

The selection of key frames is another important component of our VO. In general, the drift of a VO increases with every frame, i.e. every relative motion, which is used for the update of the absolute motion. Therefore, the concatenation of small motions should be avoided to keep the drift as low as possible. This means that the transformation $\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}$ should not be used to update the absolute transformation $\mathbf{T}_{\mathbf{C},m\mathbf{C},1}$ if the motion between the image pair $\{m-1,m\}$ is small or even zero. Instead, we should stay with $\mathbf{T}_{\mathbf{C},m-1\mathbf{C},1}$.

We define a stereo frame m as a key frame $\overline{m}$ if its relative transformation is used for the absolute motion update. Our defined requirement is that the relative change in position is bigger than 2 m or the relative angle of rotation [9] is bigger than 20°.

D. Bundle Adjustment

Windowed bundle adjustment [28] is the last important step in our feature-based VO system. It is used to optimize the relative transformations of the most recent $\overline{M}$ key frames. For simplicity, we assume n 3*D*-points $i \in \{1,\ldots,n\}$, which are seen in a window of $\overline{M} \leq \overline{m}$ key frames $j \in \{\underline{m},\ldots,\overline{m}\}$. Hereby, the index of the oldest stereo frame in the window is defined as $\underline{m} = (\overline{m} - \overline{M} + 1)$. To reduce the computation demand, our VO just uses a window with the most recent $\overline{M} = 2$ key frames, i.e. in total the features of four images are used for the optimization.

Bundle Adjustment is, like in (3), again the minimization of the image reprojection error and is given by

$$\min_{\mathbf{T}_{\mathbf{C},j\mathbf{C},1},\,{}_{\mathbf{C},1}\mathbf{t}_{\mathbf{C},1\mathbf{X},i}} \sum_{i=1}^{n} \sum_{j=\underline{m}}^{\overline{m}} \mathrm{d_E}\left({}_{\mathbf{v},j}\mathbf{t}_{\mathbf{v},j\mathbf{X},i},\, {}_{\mathbf{v},j}\hat{\mathbf{t}}_{\mathbf{v},j\mathbf{X},i}(\mathbf{T}_{\mathbf{C},j\mathbf{C},1},\, {}_{\mathbf{C},1}\mathbf{t}_{\mathbf{C},1\mathbf{X},i})\right)^2. \tag{4}$$

Thereby, ${}_{\mathbf{v},j}\mathbf{t}_{\mathbf{v},j\mathbf{X},i}$ and ${}_{\mathbf{v},j}\hat{\mathbf{t}}_{\mathbf{v},j\mathbf{X},i}$ are, respectively, the vectors of the observed and estimated 2*D* coordinates of point i in key frame j. Due to the projection of the point $\mathbf{X}_i$ into the image plane, the estimated coordinates are dependent on the absolute transformations $\mathbf{T}_{\mathbf{C},j\mathbf{C},1}$, the 3*D* coordinate vector ${}_{\mathbf{C},1}\mathbf{t}_{\mathbf{C},1\mathbf{X},i}$ and the corresponding camera projection matrices. The camera parameters are assumed as constant and known via a prior calibration. The minimization of (4) is done using the sparse bundle adjustment library of Lourakis et al. [19].

Fig. 2: Vehicle with measurement setup and DGPS-receiver

IV. Experimental validation

Our realistic dataset shows the performance of our VO on a track through a forest. It contains GPS data as well as images during a drive of a truck on a logging road. The vehicle used for the measurement is further discussed in Section IV-A and sample images of the road can be seen in Section IV-B.

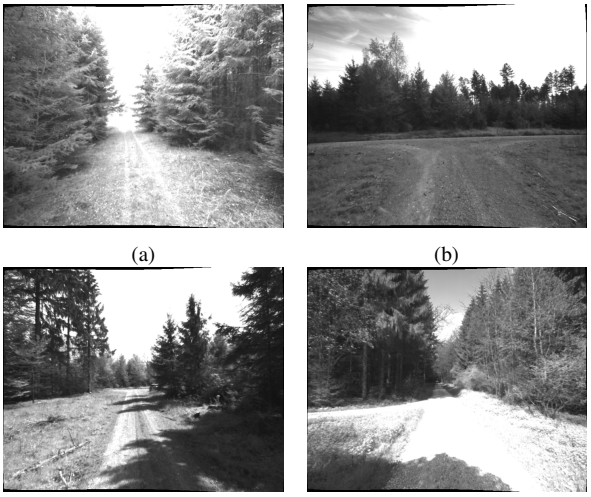
A. Setup

A small truck, equipped with a stereo-camera system, was used for the measurement. The cameras are mounted on the back of the driver's cab via aluminum profiles and magnets. This mounting position guarantees a good viewpoint backward without having unwanted objects within the field of view. In addition to the cameras, a DGPS-unit (Differential Global Positioning System) is used for ground truth although the signal strength lacks inside the dense forest.

The vehicle and its measurement setup is shown in Fig. 2. The cameras are mounted parallel on an aluminum profile at a distance of approximately one meter. The 12 V battery of the truck powers both cameras inside the wired box. Two Gigabit Ethernet cables facilitate the data transfer of the stereo-camera system, which operates at 10 Hz. A higher sample rate of the cameras is unsuitable due to the high computing time of the VO. We used the following sensors and devices:

- 2× JAI GO monochrome-cameras (JAI GO-5000M-PGE) with a maximal sampling rate of 22 Hz with the full resolution of 2560×2048 pixel
- 1× DGPS-system with open sky localization error of ca. 2 cm/0.1°
- 1× Xsens MTi-30 IMU with 400 Hz sampling rate (additional sensor for further experiments)
- Lenovo Thinkpad S540 with Intel Core i7-4510U CPU @ 2.00 GHz and 16 GB RAM
  Windows 7 Professional SP1 - 64 Bit

The JAI GO cameras allow a maximum resolution of 2560×2048 pixel. Due to lots of bumps on the logging road, the long exposure time of the cameras might blur images at darker areas of the forest. Therefore, we use 2×2 pixel binning and a resulting resolution of 1280×1024 pixel to decrease the exposure time. The resolution of the images is further decreased to 640×512 pixel by software to reduce the computing time of feature detection and description. After the decrease of the resolution, a rectification of these images is also done.



(a) (b) (c) (d)

Fig. 3: Sample images of the driven logging rode

(a) Left stereo image (b) Right stereo image

Fig. 4: Stereo image pair with branch occlusion

B. Experiments

Our dataset contains two different drives of the presented vehicle on a logging rode and illustrates a realistic performance of our VO system. Figure 3 shows some road sections of our scenarios. Widespread areas and overexposed images may result in an inhomogeneous distribution of features, which is a big challenge for the VO.

The first scenario of our dataset is a 3×75 m long test drive on the part of the logging road, which is shown in Fig. 3a. In this dense forest area, our proposed VO demonstrates its robustness against overexposed images and occluded cameras like shown in Fig. 4, where a branch occludes the left camera entirely. The implementation is robust enough to handle such situations and still estimates a valid pose. The results of our test drive are presented in Fig. 5. The starting point is marked with a circle. Due to the low signal quality of the GPS, the reference position exhibits some inconsistencies. The plotted coordinate system is the one of the GPS with X pointing to East, Y to North and Z upwards.

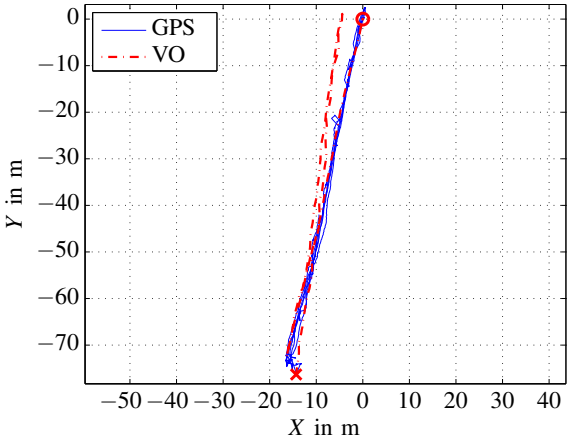


Fig. 5: Scenario 1 – Comparison of the estimated trajectory with GPS

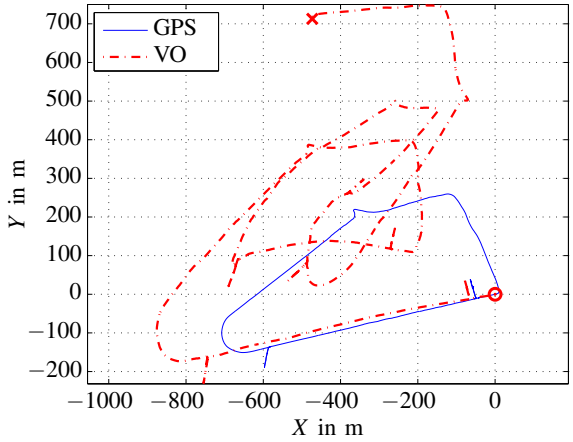


Fig. 6: Scenario 2 – Comparison of the estimated trajectory with GPS

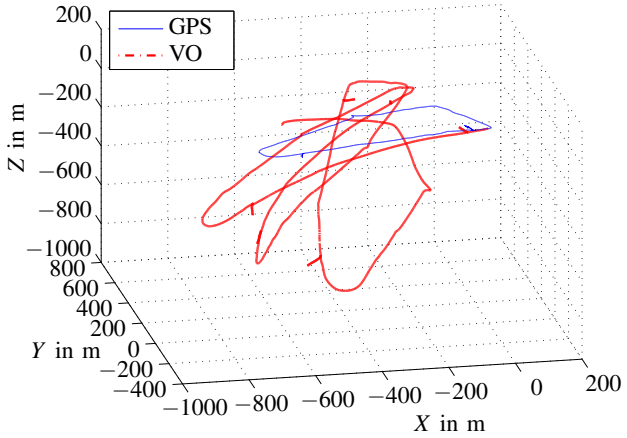


Fig. 7: Scenario 2 – Comparison of the estimated 3*D*-trajectory with GPS

As shown in Fig. 5, the estimated pose of the first 75 m fits very well with ground truth. The estimated trajectory of the return slightly deviates from the GPS reference. The inaccurate estimation of the orientation happens due to occlusions of the left camera like it is shown in Fig. 4. However, our robust VO prevents a total failure and still allows a valid but slightly inaccurate pose estimation via using images of the right camera instead. The third track of the logging road is estimated well as a straight line again.

Using the mentioned laptop, the computation time of our off-line VO of this scenario is about 0.529 s per stereo pair. This time duration is increased due to the occlusion of the left camera, which acquires the additional processing of the right image instead of just the left one. This problem especially happens at the return of the vehicle because the cameras are mounted on the back of the driver‘s cab.

The second scenario of our dataset is a 3×2169 m long drive of the presented vehicle on a logging road. This scenario should deliver an answer about the drift behavior of our implemented VO. Figure 3b represents the first image of this sequence. The results are shown in Fig. 6. The estimated trajectory is inconsistent with the ground truth and just the first 2169 m long loop can be identified somehow. Then, the trajectory continuously deviates from the driven track. If we look closely at the start of Fig. 6, it shows that distances are estimated too large in general. The whole trajectory seems to be scaled compared to the original track.

For a better understanding of the results, it is helpful to further investigate the 3*D*-trajectory illustrated in Fig. 7. Referring to the estimated VO path of this figure, from the beginning the truck starts to move downwards and also to twist sideways. This results in a distorted trajectory instead of a more or less planar movement of the truck.

The explanation of the occurrent problem can be found with a closer look at the features, which are used for the pose estimation. Figure 8 shows the detected A-KAZE features of Fig. 3b, and the sweeping area only contains a few key points. Most of them are found at the treetops in the upper half of the image. In the worst-case scenario, for example if all trees have the same height, all features are just on one line instead of being well distributed in the image. The outcome of this is an ill-conditioned pose estimation and hence an inaccurately estimated distance and pitch-angle. The problem of this scenario is that image positions hardly change by a further increase of the distance.

However, as shown in Fig. 9, the yaw angle can be estimated well because a planar rotation definitely changes the image positions of these features. The figure clearly illustrates every turn of the track and the good consensus of the yaw angle for each loop. Just some minor deviations due to different drive behavior and drift can be seen. This means that Fig. 9 shows the potential of our implemented VO for applications which mainly rely on a good estimation

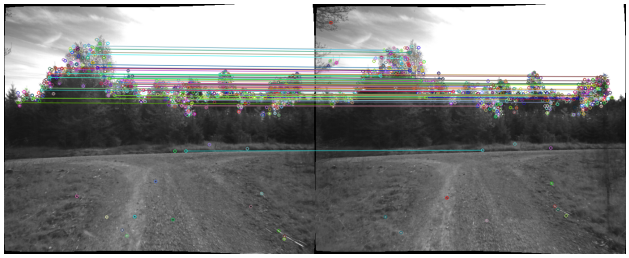


Fig. 8: Features of two left images used for pose estimation

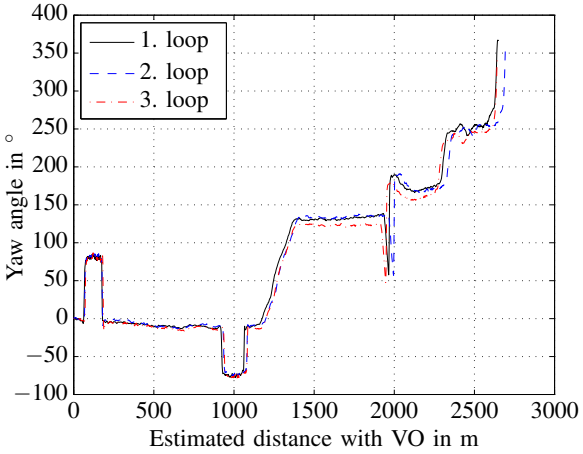


Fig. 9: Scenario 2 – Comparison of estimated yaw angles

of the yaw angle like it's necessary for agricultural vehicles.

Using the given laptop, in average the computing time of the pose estimation takes 0.349 s per stereo pair. This equates to approximately three pose estimations per second.

V. Conclusion and Future Work

We developed a Visual Odometry system that is based on a stereo-camera pair and capable of estimating the position and orientation of an agricultural vehicle in GPS-obstructed environments. We deployed our system on a small truck and carried out measurements for two different logging road scenarios. The results show that an insufficient distribution of features can lead to an ill-conditioned pose estimation, and hence to an inaccurately estimated distance and pitch angle. Due to the incremental concatenation of relative motions, this results in an increased error in position. However, our robust VO system is highly capable of estimating the orientation (yaw angle) with acceptable accuracy in unstructured environment. This is especially shown in the first scenario in the dense forest where the signal quality of GPS lacks.

Future work includes the improvement of the distribution of features and hence the pose estimation. Uniformly distributed features could be achieved via using different detector parameters for the upper and lower half of the image.

Another goal is to reduce the computing time of the VO to facilitate an online system. This can mainly be done via the parallelization of repeatable tasks like feature detection and description.

Furthermore, the next steps include the incorporation of the data of an IMU that were recorded simultaneously during our measurements. We plan to use a data fusion algorithm such as a Kalman Filter to improve the overall accuracy by combining the VO with the IMU data.

References

[1] M. Agrawal and K. Konolige, "Rough terrain visual odometry," in *Proceedings of the International Conference on Advanced Robotics (ICAR)*, vol. 1, 2007, pp. 28–30.

[2] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *British Machine Vision Conference (BMVC)*, 2013.

[3] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *Computer Vision–ECCV*. Springer, 2012, pp. 214–227.

[4] S. Blackmore, "Precision farming: An introduction," *Outlook on Agriculture*, vol. 23, no. 4, pp. 275–280, 1994.

[5] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 2005, pp. 903–910.

[7] A. I. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3d visual odometry," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 40–45.

[8] K. Cordes, L. Grundmann, and J. Ostermann, "Feature evaluation with high-resolution images," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2015, pp. 374–386.

[9] E. B. Dam, M. Koch, and M. Lillholm, *Quaternions, Interpolation and Animation*. Datalogisk Institut, Københavns Universitet, 1998.

[10] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 1403–1410.

[11] M. M. Deza and E. Deza, *Encyclopedia of Distances*. Springer, 2009.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] F. Fraundorfer and D. Scaramuzza, "Visual odometry, part II: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[14] R. W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[15] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of 4th Alvey Vision Conference*, 1988, pp. 147–151.

[16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003.

[17] K. Konolige, M. Agrawal, and J. Sola, "Large scale visual odometry for rough terrain," in *Robotics Research*. Springer, 2011, pp. 201–212.

[18] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate $o(n)$ solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[19] M. I. Lourakis and A. A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software (TOMS)*, vol. 36, no. 1, p. 2, 2009.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[22] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2004, pp. I–652.

[23] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[24] D. Scaramuzza and F. Fraundorfer, "Visual odometry, part I: The first 30 years and fundamentals," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[25] T. Strang, F. Schubert, S. Thölert, R. Oberweis, M. Angermann, B. Belabbas, A. Dammann, M. Grimm, T. Jost, S. Kaiser, *et al.*, "Lokalisierungsverfahren," Deutsches Zentrum für Luft-und Raumfahrt (DLR), Tech. Rep., 2008.

[26] N. Sünderhauf, K. Konolige, S. Lacroix, and P. Protzel, "Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle," in *Autonome Mobile Systeme*. Springer, 2006, pp. 157–163.

[27] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, "A new approach to vision-aided inertial navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4161–4168.

[28] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. Lecture Notes in Computer Science (LNCS), B. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer-Verlag, 2000, vol. 1883, pp. 298–372. [Online]. Available: https://hal.inria.fr/inria-00590128