Stereo Visual Odometry Without Temporal Filtering

Joerg Deigmoeller(✉) and Julian Eggert

Carl-Legien-Strasse 30, 63073 Offenbach am Main, Germany
joerg.deigmoeller@honda-ri.de

Abstract. Visual Odometry is one of the key technology for navigating and perceiving the environment of an autonomous vehicle. Within the last ten years, a common sense has been established on how to implement high precision and robust systems. This paper goes one step back by avoiding temporal filtering and relying exclusively on pure measurements that have been carefully selected. The focus here is on estimating the ego-motion rather than a detailed reconstruction of the scene. Different approaches for selecting proper 3D-flows (scene flows) are compared and discussed. The ego-motion is computed by a standard P6P-approach encapsulated in a RANSAC environment. Finally, a slim method is proposed that is within the top ranks of the KITTI benchmark without using any filtering method like bundle adjustment or Kalman filtering.

1 Introduction

For autonomously navigating platforms it is indispensable to use some kind of odometry to travel along a planned path or to relocate itself in a known environment. One possibility is wheel odometry, which is the combination of measured revolutions of a platforms wheel plus steering angle if available. This method suffers from slip and also heavy drift in position which increases over time. Similar problems occur with an IMU (Inertial Measurement Unit) that usually incorporates accelerometers, gyroscopes and sometimes GPS within a strap-down algorithm. This requires an intensive and complex filtering to compensate errors from GPS plus huge drifts from the accelerometers. In the end, such a system can become very expensive and slow in terms of reaction time on sudden changes in movement.

On the other side, visual odometry has become a serious alternative because of decreasing camera costs and the fast developing market of small powerful integrated low cost processors. Still, problems like drift over time remain. Nevertheless, the accuracy is in a much lower range than wheel odometry (also because the slip problem is avoided) and it opens the possibility for many other applications that make use of cameras, like collision avoidance, lane keeping or sign recognition. Also very powerful is the combination of visual odometry with gyroscopes which can have very accurate rotation rates [15].

In this paper, the focus is on visual odometry with a stereo camera. This is the easiest vision set-up to implement and the one that currently leads to

B. Rosenhahn and B. Andres (Eds.): GCPR 2016, LNCS 9796, pp. 166–175, 2016.
DOI: 10.1007/978-3-319-45886-1_14

the most accurate results. In contrast, a mono camera set-up is per se not able to estimate the real 3D-translation (only up to scale) because of the missing real world relation. To overcome this, a ground plane estimation with known distance of the camera above ground could be used to compute the missing scale. This brings in new errors of inaccurate or even false measures. The most straightforward and accurate way is to get the scale factor from a calibrated stereo set-up.

Further, the drift inherent to purely visual odometry system could be reduced by applying VSLAM-techniques (Visual Simultaneous Localization and Mapping) which relocate the moving platform to previously visited places. The main interest of this work is going back to the starting point of a VSLAM-system - the pure visual odometry - and to take maximal advantage of the process to achieve a high precision with a simple as possible approach.

2 Related Work

The term visual odometry appeared for the first time in the publication [14]. A very comprehensive overview of state of the art visual odometry can be found in [7,17]. The key message from their tutorial for a stereo system is to compute the relative rotation and translation by minimizing the re-projection error on the image plane (Perspective from n Points, PnP). In contrast, minimizing the error of 3D-points is inaccurate because of increasing uncertainties with increasing depth. Additionally, a filtering method like bundle adjustment should be applied to reduce the drift over time.

Thanks to publicly available benchmarks like the well-known KITTI benchmark [8], visual odometry methods are now comparable in their precision. Nearly all of the top ranked methods on the KITTI benchmark apply the minimization of the re-projection error. The top ranked visual odometry submission [4] first estimates the rotational motion and subsequently the translation. Features are matched by a combination of SAD and NCC plus geometric constraints. They are tracked over time and pixel positions are refined in a predictor-corrector manner. Long life features are preferred against shortly tracked features. A similar approach of tracking long life features has been used in [1]. Again, they refine tracked features in a prediction-correction framework called “integrated features”. [16] uses standard visual odometry processes like feature tracking and multi-window bundle adjustment in a carefully built system motivated from monocular visual odometry. A different approach is presented by [5] which performs a photometric alignment based on depth maps. Depth is computed along the stereo baseline (Static Stereo) and the movement baseline (Temporal Stereo). Correspondence measures are tracked over time and search ranges are constrained from previous estimates. Camera positions are finally optimized in a pose-graph using key frames and loop closures are done if the cameras are close to a previously visited location.

Some work also analyzed the influence of features at different depth on the pose estimation. For example [13] uses features at infinity distance (infinity according to the pixel raster of the image sensor) for rotation estimation and close features for translation. [11] make use of the bucketing technique - known from robust regression methods - to indirectly pick features at different depths for a better pose estimation. For pose optimization, they use an Iterated Sigma Point Kalman Filter (ISPKF).

In contrast to the previously mentioned publications, this work does not use any temporal filtering, neither bundle adjustment nor any predictor-corrector like filtering. The system is built in a way that as few as necessary processing steps are used and poses are concatenated from pure measurements. This reduces the implementation effort drastically as already a proper feature tracking requires indexing and complex managing over time.

The advantage of the proposed system is its fast reaction time which is important for applications like collision avoidance or for drastically changing movements. Still, the approach is competing with the top ranked methods on the KITTI benchmark and currently on rank 8 under the stereo vision methods (see Chap. 6 for more details).

3 System Overview

Assuming that the stereo images are already rectified, the system consists of two parts. First, the 3D-flow (scene flow) computation and second, the pose estimation.

The scene flow computation is a combination of disparity and optical flow computation using standard Harris corner detector [9] with subsequent pyramidal Lucas & Kanade optical flow computation [2].

The pose estimation is a simple P6P-method (Perspective from 6 Points) encapsulated in a RANSAC (Random Sample Consensus) framework.

The idea was to use available standard methods (e.g. from OpenCV) to first extract the crucial points of visual odometry. In a later step - which is not part of this paper - specializations of the core parts are planned.

In the remainder of this paper, the focus is on the comparison of different constraints on the scene flow estimations in Sect. 4. The pose estimation - discussed in Sect. 5 - is not modified and runs with parameters that have been determined in previous optimizations. Finally, experimental results are shown on the KITTI benchmark data.

4 Scene Flow Estimation

The scene flow is always computed by two consecutive stereo image pairs $\{I_i^l, I_i^r\}$ and $\{I_{i+1}^l, I_{i+1}^r\}$. Initially, standard Harris corners are estimated by first computing the partial image derivatives:

$$Q(x) = \sum_W \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{1}$$

and then extracting the corner response $H(x,y)$ by:

$$H(x) = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{2}$$

where λ_1 and λ_2 are the eigenvalues of $Q(x)$. W is the neighbourhood around a pixel position x =(x,y)^T. The parameter k influences the "cornerness" of a feature. W is a 7×7 window, with the size determined by optimization using an extensive parameter grid search. 4000 features are initially computed and sorted by their corner response value.

Additionally, the integer pixel position of the Harris feature is refined to subpixel accuracy. That means, x is recalculated with the help of gradient information in its neighbourhood [3].

After the feature extraction, the Lucas & Kanade optical flow is computed for pairwise combinations of the input images $\{I_i^l, I_i^r, I_{i+1}^l, I_{i+1}^r\}$, where i, $i+1$ denote subsequent images in time and l, r denote left and right images of the stereo set-up. The Lucas & Kanade optical flow is the perfect counterpart to the Harris corner, because its correlation measure results in the same partial image derivatives multiplied by a vector containing the temporal derivatives:

$$\begin{pmatrix} \mathrm{u} \\ \mathrm{v} \end{pmatrix} = \sum_W \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_W I_x I_t \\ -\sum_W I_y I_t \end{bmatrix} \tag{3}$$

where v = (u,v)^T is the optical flow. As v is only valid in a local neighbourhood the pyramidal approach has been used to propagate optical flows from down-sampled images to the highest resolution. 5 pyramid levels have been used and for W a window of 9×9 has been chosen, also determined from previous parameter optimization.

Lucas & Kanade optical flows are very fast to compute but tend to get stuck in local minima. Therefore, further checks are required to limit the measurements to a reliable set. On the other hand, wrong checks can remove features that might be important for the visual odometry. This sensitive issue is tackled more in detail in the following by different experiments on the KITTI training data.

On the KITTI website an evaluation software and ground truth poses for 11 sequences are available [8]. For every sequence, the translation error and rotation error is calculated. All following experiments refer to this error measure.

For scene flow estimation, the disparity between $\{I_{i+1}^l, I_{i+1}^r\}$ and the optical flow between $\{I_i^l, I_{i+1}^l\}$ (see Fig. 1) are computed by the Lucas & Kanade method. This is the minimal processing effort as the poses are optimized on the re-projection error, i.e. 3D positions from $\{I_{i+1}^l, I_{i+1}^r\}$ and 2D correspondences from $\{I_i^l, I_{i+1}^l\}$ are sufficient (see Sect. 5).

The first consistency check is a forward/backward check. That means, if the optical flow or disparity from the end point back to the starting point deviates more than a threshold t_{fb}, then the feature is rejected. Table 1 shows the translation errors and rotation errors for different t_{fb}. As a threshold of $t_{fb} = 5$ pixels gives the best result, this value is used for further experiments. On the other hand, switching the forward/backward check off leads to a significant drop of the performance.

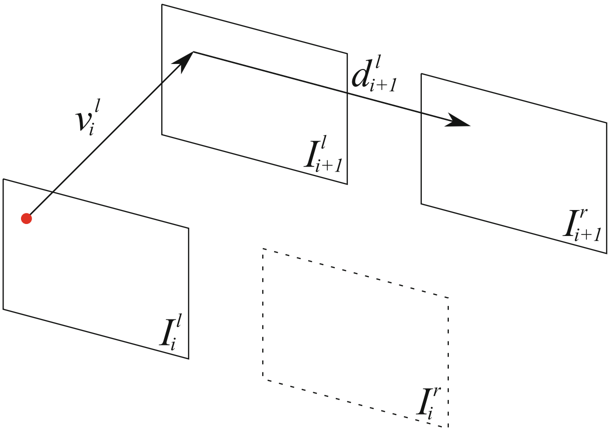

Fig. 1. Computation of the disparity d_{i+1}^l and the optical flow v_i^l for a Harris corner feature initialized in I_i^l.

Table 1. Influence of backwards check on overall performance

	$t_{fb} = 1$	$t_{fb} = 2$	$t_{fb} = 5$	fb off
translation error [%]	1.3426	1.3314	1.3070	2.4815
rotation error [deg/m]	0.0063	0.0065	0.0062	0.0175

The second consistency check is on the disparity measure only. Since for disparity measure a standard optical flow method is used, all vectors that have a y-component greater than zero pixels are theoretically not possible because a perfect rectification aligns both images in a way that the epipolar lines are horizontal. In practice, rectifications are never perfect and hence a threshold to remove disparities with a larger y-component than a threshold t_d is introduced. Additionally, disparities that have a positive x-component are obviously also invalid measurements, because disparities should only have negative signs (measured from the left to the right image). Therefore, positive disparities are also rejected. Based on Table 2, a threshold of $t_d = 1$ is chosen for future tests.

Table 2. Influence of check for y-disparity component on overall performance

	$t_d = 1$	$t_d = 2$	$t_d = 5$
translation error [%]	1.3070	1.3424	1.3663
rotation error [deg/m]	0.0062	0.0066	0.0067

So far, tests were made with a Harris corner response factor k close to zero. This means that as many features as possible are kept to leave the decision of rejection on the subsequent checks. In the next experiment the k value is

Table 3. Influence of k on overall performance

	$k = 0.0$	$k = 0.01$	$k = 0.02$
translation error [%]	1.3070	1.3209	1.3532
rotation error [deg/m]	0.0060	0.0064	0.0064

increased to check the influence of the "cornerness" on the overall performance. A value of $k = 0.0$ means that the features can also be edges and with increasing k features more and more resemble corners.

From Table 3, it can be seen that a value of $k = 0.0$ gives the smallest translation error. This shows that a high corner response is probably not the best indicator for a good feature for visual odometry estimation with combined optical flow and disparity. Definitely, a corner has sufficient structure to allow an optical flow measure and avoid the aperture problem. On the other hand, using stereo images and consecutive images allows for more meaningful outlier rejection checks by using geometric constraints.

The last consistency check is a circle check to identify outlier. This circle check computes the flows between left and right images in time as well as the disparities between first and second image pairs (see Fig. 2). Only if all concatenated pixel measurements end up at the same position in image I^r_{i+1} with an error less than a threshold t_{cc}, the feature is kept.

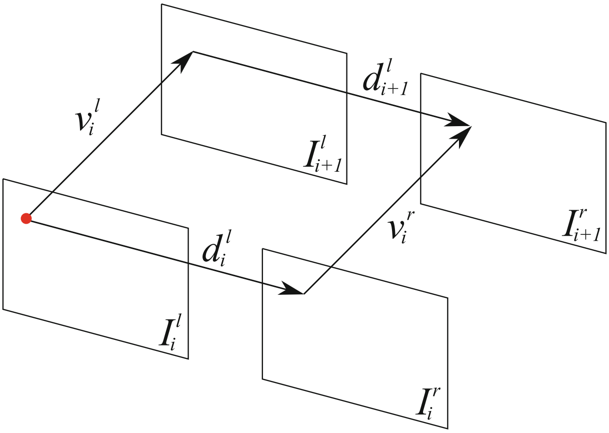

Fig. 2. Circle check: only if all measurements end up at the same position in I^r_{i+1} the feature is kept.

As can be seen in Table 4, the circle check does not significantly improve the performance compared to the previous version (1.3070 % against 1.3048 %). Additionally, it requires double computation effort compared to the version

Table 4. Influence of circle check on overall performance

	$t_{cc} = 1.0$	$t_{cc} = 2.0$	$t_{cc} = 5.0$
translation error [%]	1.3453	1.3124	1.3048
rotation error [deg/m]	0.0064	0.0064	0.0062

without circle check. Therefore, the previous version without circle check is used as final method.

5 Pose Estimation

The pose estimation is a standard P6P-approach minimizing the re-projection error on the image plane as follows:

$$\arg\min_{T_i} \sum_j \left\| \hat{x}^l_{i,j} - \frac{1}{[0\ 0\ 1] T_i \hat{X}^l_{i+1,j}} T_i \hat{X}^l_{i+1,j} \right\|^2 \tag{4}$$

where T_i is the transformation matrix from time step $i+1$ to time step i containing rotation and translation:

$$T_i = \begin{pmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{pmatrix}$$

$\hat{X}^l_{i+1,j}$ is the homogeneous 3D-position in the second image estimated from d^l_{i+1} and the values from rectification for focal length, principal point and baseline. $\hat{x}^l_{i,j}$ are the Harris corner positions converted to homogeneous coordinates (cf. Fig. 1). T_i is computed from $i+1$ to i to directly get the ego-motion of the vehicle. Computing from i to $i+1$ would return the coordinate transformations, which is obviously the inverse ego-motion.

After a first estimation of T_i using Singular Value Decomposition (SVD), R and T are refined by non-linear optimization on the geometric error [12,17].

The computation of T_i is encapsulated in the robust regression framework RANSAC [6,10]. The inlier/outlier-ratio has been set to a conservative value of 0.5 to avoid run-time optimizations at too early stage. The most crucial parameter is the re-projection error that defines the threshold for the census set t_r. The feature points are at such a high precision that the re-projection error can be set below 1 pixel. Table 5 shows the influence of the re-projection threshold. The overall performance drastically increases with decreasing t_r. A threshold of $t_r = 0.1$ pixel finally lead to the best results on training and testing set.

6 Experimental Results

In the previous chapters different methods have been evaluated on the KITTI benchmark training data. In this chapter, the results of the testing data are

Table 5. Influence of the re-projection error threshold on overall performance

	$t_r = 0.5$	$t_r = 0.3$	$t_r = 0.1$
translation error [%]	1.8953	1.6104	1.3070
rotation error [deg/m]	0.0113	0.0090	0.0060

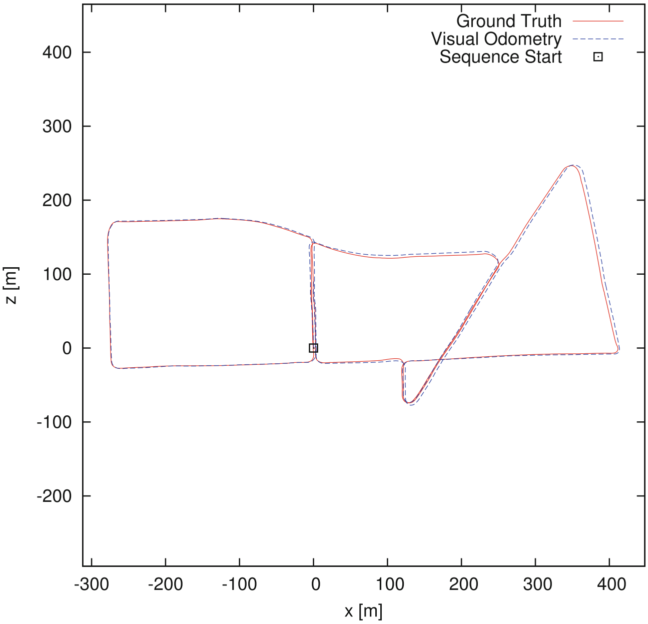

Fig. 3. Reconstructed path of sequence 13 and ground truth path from the KITTI benchmark.

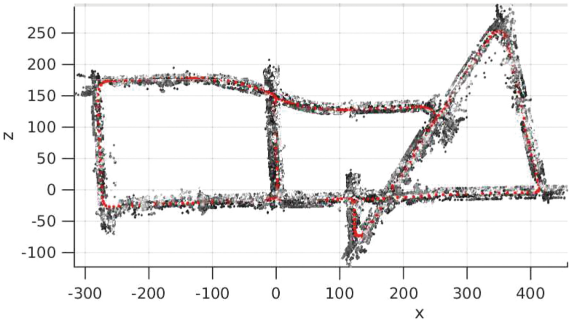

Fig. 4. Reconstructed path of sequence 13 (red dots) and 3D-reconstruction along the driven path (gray dots). (Color figure online)

presented. In summary, the algorithm computes 3D positions from $\{I^l_{i+1}, I^r_{i+1}\}$ and 2D correspondences from $\{I^l_i, I^l_{i+1}\}$ (no circle check). Additionally, a forward/backward check is applied with $t_{fb} = 5$ pixels, a disparity rejection if $t_d > 1$ pixel and a Harris corner response factor of $k = 0.0$. The RANSAC applies a re-projection threshold of $t_r = 0.1$ pixel.

Uploading this version gives a translation error of 1.17 % and a rotation error of 0.0035 [deg/m], which ranks 8th under the vision approaches (NOTF, 5th April 2016) and 11th in overall ranking (including laser approaches).

Figure 3 depicts ground truth poses and the computed poses by the NOTF algorithm. Figure 4 depicts the same path but with the reconstructed features that have been used for the ego-motion estimation.

At the moment, the run-time is comparably high (440 ms on a Core i5-4460, 1 core used at 3.2 GHz) which is due to the fact that the parameters are chosen very conservatively; many more iterations than required for e.g. sub-pixel refinement and random sampling of RANSAC. This has been a deliberate decision so as not to optimize at an too early stage. In a next step, the parameters will be adapted in a way that the performance remains comparable at a lower run-time, which will be expected to be in a range of 100–150 ms on the same machine.

7 Conclusion

A simple and slim visual odometry method has been proposed that is within the top ranks of the KITTI benchmark. In contrast to other methods, no temporal filtering is applied. The results support the conclusion that with a proper outlier rejection, raw and unfiltered optical flow measures can deliver the same precision as current methods applying bundle adjustment or Kalman filtering.

The presented study tackled the problem of outlier rejection by purely varying geometric constraints on the optical flow measure. It has been shown that such constraints have a high influence on the performance. Choosing the right combination is a balancing act between keeping as many accurate features as possible and rejecting imprecise measures.

The pose estimation has not been modified, which is a topic remaining for future work. Probably, there will be a higher precision if the selection of measurements in the RANSAC framework is done with more prior knowledge instead of pure random sampling.

Still unclear is if there will be a performance boost by applying filtering methods. This will also be an open question for future investigations.

Further improvement is expected if a real 1D disparity measure is applied for 3D features instead of standard optical flow with subsequent feature rejection. Probably, a significant number of disparities gained by the optical flow procedure are wrong measures, because a full search is applied.

References

1. Badino, H., Yamamoto, A., Kanade, T.: Visual odometry by multi-frame feature integration. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW) (2013)
2. Bouguet, J.Y.: Pyramidal implementation of the Lucas Kanade feature tracker. Intel Corporation, Microprocessor Research Labs (2000)
3. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library (2008)
4. Cvisic, I., Petrovic, I.: Stereo odometry based on careful feature selection and tracking. In: European Conference on Mobile Robots (ECMR) (2015)
5. Engel, J., Stueckler, J., Cremers, D.: Large-scale direct SLAM with stereo cameras. In: International Conference on Intelligent Robot Systems (IROS) (2015)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**, 381–395 (1981)
7. Fraundorfer, F., Scaramuzza, D.: Visual odometry part II: matching, optimization and applications. IEEE Robot. Autom. Mag. Robustness **19**, 78–90 (2012)
8. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Kitti vision benchmark suite (2015)
9. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of Fourth Alvey Vision Conference (1988)
10. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision (2003)
11. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In: IEEE Intelligent Vehicles Symposium (2010)
12. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: An Invitation to 3-D Vision (2001)
13. Mair, E., Burschka, D.: Z(inf) - monocular localization algorithm with uncertainty analysis for outdoor applications. In: Mobile Robots Navigation (2010)
14. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 1, pp. 652–659. IEEE (2004)
15. Nuetzi, G., Weiss, S., Scaramuzza, D., Siegwart, R.: Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM (2010)
16. Persson, M., Piccini, T., Felsberg, M., Mester, R.: Robust stereo visual odometry from monocular techniques. In: IEEE Intelligent Vehicles Symposium (2015)
17. Scaramuzza, D., Fraundorfer, F.: Visual odometry part I: the first 30 years and fundamentals. IEEE Robot. Autom. Mag. **18**, 80–92 (2011)