Multi-frame Feature Integration for Multi-camera Visual Odometry

Hsiang-Jen Chien[1](✉), Haokun Geng[2], Chia-Yen Chen[3], and Reinhard Klette[1]

[1] School of Engineering, Auckland University of Technology,
Auckland, New Zealand
jchien@aut.ac.nz

[2] Department of Computer Science, University of Auckland,
Auckland, New Zealand

[3] Department of Computer Science and Information Engineering,
National University of Kaohsiung, Kaohsiung, Taiwan

Abstract. State-of-the-art ego-motion estimation approaches in the context of visual odometry (VO) rely either on Kalman filters or bundle adjustment. Recently proposed multi-frame feature integration (MFI [1]) techniques aim at finding a compromise between accuracy and computation efficiency. In this paper we generalise an MFI algorithm towards the full use of multi-camera-based visual odometry for achieving more consistent ego-motion estimation in a parallel scalable manner. A series of experiments indicated that the generalised integration technique contributes to an improvement of above 70 % over our direct VO implementation, and further improved the monocular MFI technique by more than 20 %.

Keywords: Visual odometry · Ego-motion estimation · Feature tracking

1 Introduction

The development of visual odometry contributed not only to robotics, it is also of growing importance for self-driving vehicles. The recovery of camera motion and 3D structures from video sequences has been studied since the early 80s [15]. The vision-guided rovers on Mars defined one of the early milestones. They operate by applying the framework of structure from motion (SfM). Since then, an extensive amount of work has been added to theories and practice for solving the ego-motion estimation problem in fields of visual odometry (VO) and simultaneous localisation and mapping (SLAM).

Existing visual odometry algorithms include patch-based and feature-based methods, depending on how inter-frame pixel correspondences are established. Patch-based methods, e.g. by following common optical flow, deploy search windows to track each pixel, while feature-based approaches perform a matching in feature spaces where each feature vector encodes the regional characteristics centering at the tracked pixel [6]. Feature-based approaches dominated the development of visual odometry in the last decade [7].

T. Bräunl et al. (Eds.): PSIVT 2015, LNCS 9431, pp. 27–37, 2016.
DOI: 10.1007/978-3-319-29451-3_3

Although the tracking of features has been a well-studied topic in the field of computer vision, for the use in visual odometry more strict conditions apply, and hence a generic feature tracker can often fail. Some filtering mechanisms and temporal constraints need to be considered in order to remove incorrect inter- frame feature matches which could hazard the ego-motion estimation process [12].

The drift caused by the accumulation of ego-motion estimation is yet another major concern [11]. Bundle adjustment (BA) is considered to be the "golden standard" to solve this issue [5]. BA reduces the accumulated error by a global optimisation process designed to converge to the maximum-likelihood estimate that optimally fits the observed feature locations and their 3D coordinates. However, many BA approaches are implemented only in a local scale since it requires a huge amount of computation involving the solution of large linear systems.

Multi-frame feature integration (MFI), proposed in [1], provides a cost-effective yet comparable solution for drift suppression as well as for improving the tracking process. By integrating multiple measurements of the same feature at different times, the 3D measurement noise is canceled under certain conditions. The integration of feature also implicitly introduces a dependency of the ego-motion estimation between each frame. Such dependency is helpful in the reduction of drift. Figure 1 shows an example of the growth of accumulated drift and the suppression of the drift.

However, we noticed that the original MFI algorithm uses the right camera of a stereo-vision system only for 3D calculation, while the feature detection and tracking is done in a monocular manner. In this work we propose a generalisation of a similar idea, which uses multi-camera data to enhance the robustness of feature tracking, and to further reduce the drift.

The paper is organised as follows. In Sect. 2 we formulate the VO problem. In Sect. 3 the MFI algorithm is described. In Sect. 4 we generalise the MFI algorithm to make a full use of a multi-camera system. Experimental results are discussed in Sect. 5 to study the improvement of the proposed method, while Sect. 6 concludes this paper.

2 Feature-Based Visual Odometry

The pipeline of binocular visual odometry, as shown in Fig. 2, involves several domains in computer vision. The input of the pipeline is a pair of images captured time-synchronised by a left and right camera. It produces the estimated 3D structure of tracked scene points and motion of the vision system relative to the pose where the previous input images were taken. In this work we follow a feature-based framework which derives the motion of the cameras by tracking sparse features instead of coping with dense image patches. The advantage of a binocular framework over a monocular one is that the 3D information can be acquired using stereo matching and triangulation, hence it is preferable in high-precision applications.

Motion estimation can be achieved in either Euclidean space, projective space, or by means of both spaces. Solving the motion in Euclidean space is less

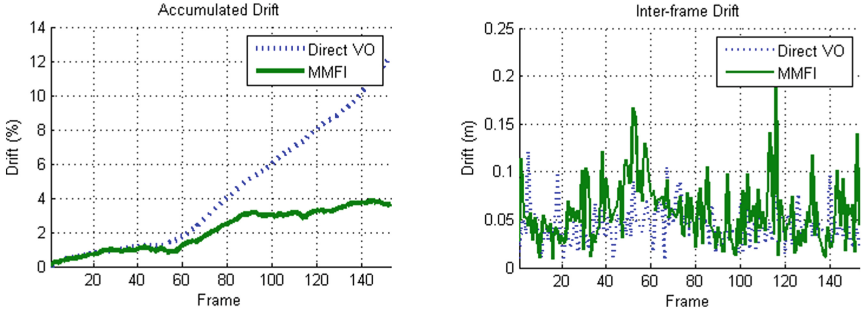

Fig. 1. The accumulation of inter-frame estimation error grows in a super-linear manner when a direct VO method is applied, as can be seen on the left plot, which also shows that the drift is effectively suppressed when the proposed multi-camera multi-frame feature integration (MMFI) method is used. By the introduction of inter-frame dependency the drift slightly increases in a local scale as can be observed on the right.

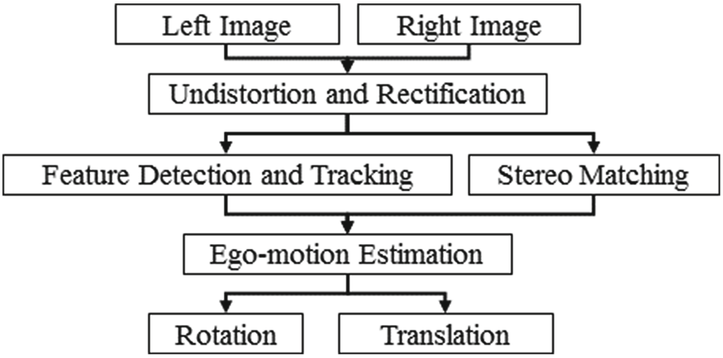

Fig. 2. Pipeline of a stereo visual odometry system using 3D-to-2D correspondences

favourable due to the highly anisotropic error covariances in the case where the 3D structure is measured using a stereo-based disparity value. On the other hand, the projective approach does not provide reliable metric information, hence the 3D-to-2D correspondences are considered as being a better choice for ego-motion estimation [15].

Optimised motion is found by a minimisation of the re-projection error. Given two sets of features F and F', and a mapping $M \subseteq F \times F'$. We also assume defined 3D and 2D measurement functions g and ρ that transform a feature into the Euclidean coordinates in $\mathbb{R}^3$ or into the image coordinates in $\mathbb{R}^2$, respectively. Furthermore, consider the perspective projection function $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$. Optimal motion is defined by the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ which minimise the sum of squares of the re-projection error, formally given by

$$\phi(\mathbf{R}, \mathbf{t}) = \sum_{\forall(\chi, \chi') \in M} \| \pi \left(\mathbf{R} g \left(\chi \right) + \mathbf{t} \right) - \rho \left(\chi' \right) \|^2 \tag{1}$$

Equation (1) can be optimised by nonlinear optimisation. There is a variety of linear closed-form solutions (e.g. efficient perspective-from-n-point (EPnP [14]), or 5-point algorithms) which provide a good initial solution for the optimisation process. In our work we applied the Levenberg-Marquardt algorithm to solve the objective iteratively.

In such a framework, the accuracy entirely relies on two factors - the tracking of features and the stereo matching algorithm, given that the system is well calibrated. Without considering temporal consistency of the recovered motion (i.e. the movements of the system at time slots j and $j+1$ are considered to be independent events), the drift grows in a super-linear manner as the inter-frame ego-motion estimations are chained to derive the global trajectory of the system. A state-of-the-art solution to suppress the drift is to use either Kalman filters or a sliding-window bundle adjustment [16].

Recently, the multi-frame integration (MFI) technique has been proposed to achieve drift suppression by introducing a dependency between the recovered ego-motion at j, and feature tracking between frame j and $j+1$, based on the idea of iteratively and alternatively improving the ego-motion estimation and feature tracking along the given video sequence.

3 Multi-frame Feature Integration

The 3D coordinates of a tracked feature, measured in multiple frames, can be averaged to acquire a better estimate of the true position of the feature in the Euclidean space, if the measurement function g follows a Gaussian error. According to this property, the integrated measurement function $\bar{g}$ is defined recursively as follows:

$$\bar{g}(\chi_i^j) = \frac{\alpha(\chi_i^j) \cdot [\mathbf{R}_j \bar{g}(\chi_i^{j-1}) + \mathbf{t}_j] + g(\chi_i^j)}{\alpha(\chi_i^j) + 1} \tag{2}$$

where $(\mathbf{R}_j, \mathbf{t}_j)$ defines the rigid transformation from frame $j-1$ to j, and $\alpha(\chi_i^j)$ denotes the accumulated number of measurements of feature χ_i at moment j. If the feature χ_i is first discovered in frame j_0, it is defined that $\bar{g}(\chi_i^j) = \mathbf{0}^\top$ and $\alpha(\chi_i^j) = 0$ for all $j < j_0$.

Before the states of features are updated by Eq. (2), the optimal ego-motion $(\mathbf{R}_j, \mathbf{t}_j)$ is estimated first by minimising

$$\phi_j(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{n} \eta \cdot \varepsilon(\chi_i^j, g; \mathbf{R}, \mathbf{t}) + (1-\eta) \cdot \alpha(\chi_i^j) \cdot \varepsilon(\chi_i^j, \bar{g}; \mathbf{R}, \mathbf{t}) \tag{3}$$

where $\eta = [0,1]$ controls the significance of the feature integration and ε measures the deviation of the projection of feature χ_i in frame t versus its tracked position:

$$\varepsilon(\chi_i^j, g; \mathbf{R}, \mathbf{t}) = \omega(\chi_i^j) \cdot \left\| \pi \left(\mathbf{R} g(\chi_i^{j-1}) + \mathbf{t} \right) - \rho \left(\chi_i^j \right) \right\|^2 \tag{4}$$

By ω we denote the weighting term of feature χ_i at moment j. If feature χ_i is not discovered at that moment, we have $\omega(\chi_i^j) = 0$ so that the feature is not taken into account for the estimation of $\mathbf{R}_j$ and $\mathbf{t}_j$.

The estimated motion can also be used to improve the tracking of features. The prediction of a feature's location in the current frame is calculated by projecting its previously integrated 3D coordinates into the current frame. The projection is then compared to the image coordinates obtained by feature matching. The deviation between both is then used to denote the reliability of the measurement, and taken into account to adjust the weighting term ω accordingly. The MFI algorithm keeps tracking the mean of such deviations, every time a feature's state is updated. When the cost of an update is higher than a predefined threshold, then the tracking process for a feature is marked "lost", and hence terminated.

Instead of terminating the tracking immediately, an attempt to re-discover the missing feature can be optionally carried out. The feature vector of the pixel, at the projection of the integrated feature, is extracted and compared with the feature being tracked. If a significantly high similarity is found, then the tracking process is resumed.

4 Multi-camera Multi-frame Feature Integration

To maximise the robustness of the feature tracking mechanism, the MFI [1] is extended in our work to use images taken by all cameras at moments j and $j+1$. The four steps of the proposed algorithm, to be walked through in the rest of this section, are as follows:

1. *Feature detection and cross matching.* Image features are detected, extracted and then matched for all camera combinations in consecutive frames.
2. *Ego-motion estimation.* The optimal rotation and translation of the inter-frame motion is calculated by minimising re-projection error *subject to all cameras.*
3. *Update of the integrated features.* The states of all the actively tracked features are updated to take into account the new observations based on the solved ego-motion. Features having significantly different prediction and observation locations are marked as lost. Attempts will be carried out to resume the tracking of these features at a later stage.
4. *Lost feature recovery.* For those features, failed to update their states due to missed matches, a prediction-and-check strategy is performed to re-discover their corresponding image features.

4.1 Spatial-Temporal Feature Matching

Considering an m-camera visual odometry system, image features are initially detected and extracted from images I_k^j and I_k^{j+1} for each camera $k = 1, 2, .., m$. Let F_k^j and F_k^{j+1} denote these features, respectively. The cross matching is initiated in feature space for each pair of feature sets $(F_k^j, F_{k'}^{j+1})$, where $1 \leq k, k' \leq m$.

The mutual Euclidean distances between feature vectors are calculated, and similar features are associated. For each feature $\chi \in F_k^j$, we have the distances

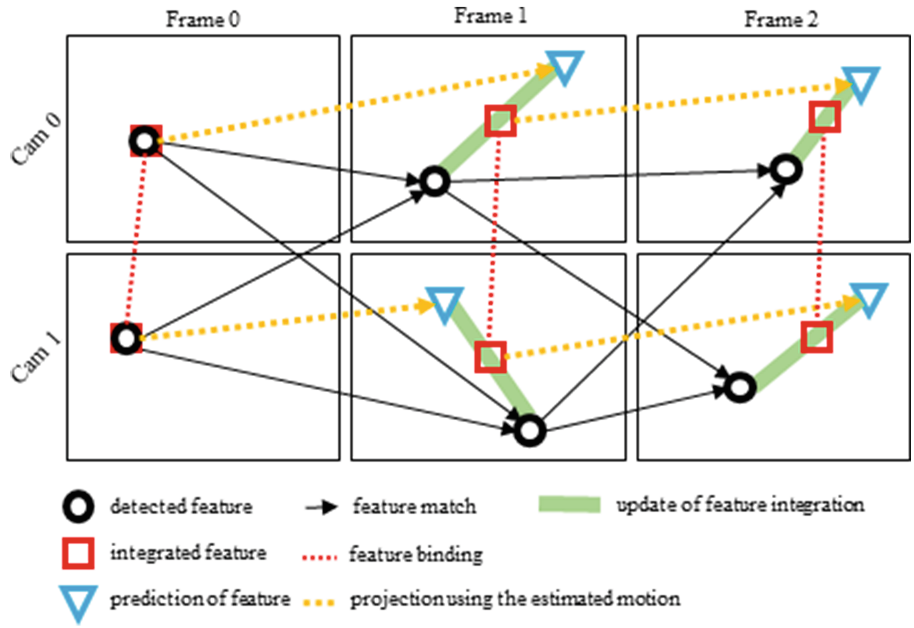

Fig. 3. Illustration of the generalised multi-frame feature integration and ego-motion estimation process

to its best match $\chi'_1 \in F^{j+1}_{k'}$ and its second best match $\chi'_2 \in F^{j+1}_{k'}$. A differential ratio is calculated as

$$\delta(\chi, \chi'_1, \chi'_2) = \frac{\|\nu(\chi) - \nu(\chi'_1)\|}{\|\nu(\chi) - \nu(\chi'_2)\|} \tag{5}$$

where ν transforms a feature to its vector representation in feature space. If the ratio is lower than a defined threshold, say 0.8, then such matching is considered ambiguous, hence rejected at this stage.

The initial matches are then verified in projective space for outlier rejection. The fundamental matrix-based RANSAC strategy is typically carried out at this stage to reject geometrically inconsistent correspondences [9]. In this work, we instead use an LMeD estimator which is considered to be a more strict and stable model for outlier identification [4]. The mislabeled inliers at this stage will still have a chance to be amended later in the lost feature recovery stage. The detected features and the matched correspondences are depicted in Fig. 3 as black circles and black solid lines, respectively.

It is worth a mention that, despite being developed independently, the described cross matching mechanism shares a similar idea of the spatial-temporal network implemented in the open source library LIBVISO2 [2]. In particular, for each frame j the implementation maps features from F^j_1 to F^{j+1}_1. For those mapped features $\chi \in F^{j+1}_1$ the matching is performed again, but this time from

F_1^{j+1} to F_2^{j+1} (i.e. doing a left-right feature matching in j-th frame.) After repeating such process through $F_2^{j+1} \rightarrow F_2^j \rightarrow F_1^j$, in the way that χ finally travels back to F_1^j, it checks if χ is mapped to itself in the end. Feature matches failed to fulfill such circular consistency are rejected by LIBVISO2 to prevent outliers being used in the ego-motion stage.

4.2 Ego-Motion Estimation

The direct 3D measurement function g is extended for also using the mean of the measurements from all k cameras:

$$g(\chi_i^j) = \sum_{1 \leq k \leq m} \frac{1}{m} \cdot g_k(\chi_i^j) \tag{6}$$

Here, each component $g_k(\chi_i^j)$ denotes a 3D measurement made by the k-th camera. The definition allows us to develop a generalised multi-camera version of Eq. (3) such that, once minimised, a system-wide consistent solution is found.

4.3 Feature Integration and State Update

Initially as $j = 0$ only the direct measurement $g(\chi_i^0)$ is used in Eq. (3) to find $(\mathbf{R}_1, \mathbf{t}_1)$. After the first ego-motion estimation, $(\mathbf{R}_1, \mathbf{t}_1)$ is taken into account to compute the integration of feature χ_i at moment $j = 1$, which yields

$$\bar{g}(\chi_i^1) = \frac{[\mathbf{R}_1 g(\chi_i^0) + \mathbf{t}_1] + g(\chi_i^1)}{2} \tag{7}$$

according to Eq. (2). Such an update is performed for each tracked feature every time when the ego-motion is solved between frames j and $j + 1$.

The magnitude of the update in Eq. (7) indicates the accuracy of the recovered ego-motion as well as the reliability of the direct measurement, and can be useful to take out unreliable 3D data in further ego-motion estimation. To this purpose we also update the running covariance by

$$\sigma^2(\chi_i^j) = \frac{\alpha(\chi_i^j) \cdot \sigma^2(\chi_i^{j-1}) + [\bar{g}(\chi_i^j) - g(\chi_i^j)]^\top [\bar{g}(\chi_i^j) - g(\chi_i^j)]}{\alpha(\chi_i^j) + 1} \tag{8}$$

and use it to adjust $\omega(\chi_i^j)$ accordingly to decrease the significance of features as more unreliable measurements are integrated.

Projecting the integrated $\bar{g}(\chi_i^j)$ to frame $j+1$ yields prediction of a previously tracked feature χ_i; (depicted as blue triangles in Fig. 3.) The prediction is also helpful to indicate problematic feature tracking. Let $\pi_k(\cdot)$ be the projection function of camera k and $\rho_k(\chi_i^j)$ the observation of feature χ_i by camera k in frame j. The prediction is defined as

$$\bar{\rho}_k(\chi_i^j) = \pi_k \left(\mathbf{R}_{j+1} \bar{g}(\chi_i^j) + \mathbf{t}_{j+1} \right) \tag{9}$$

The deviation $\|\bar{\rho}_k(\chi_i^j) - \rho_k(\chi_i^j)\|$ is checked before the update of an integrated feature is actually performed. If the projection error (depicted as green thick line in Fig. 3) is greater than a predefined threshold, then the feature is marked as lost and the update of $\bar{g}(\chi_i^j)$ will be set on hold for further investigation.

4.4 Lost Feature Recovery

A feature $\chi_i^j \in F_k^j$ is lost in frame $j+1$ between camera k and k' if either there exists no matched feature $\chi' \in F_{k'}^j$, or the difference between its prediction and the observation $\rho_{k'}(\chi')$ is not acceptable.

To resume the tracking of a lost feature, the feature vector is extracted from the predicted location $\bar{\rho}(\chi_i^j)$. Then the Euclidean distance between the extracted descriptor and $\nu_k(\chi_i^j)$ is checked against the statistics of successfully tracked features. If the distance is within 1-σ then the feature is considered recovered.

Figure 4 shows some examples of re-discovery of lost features in the image taken by another camera from frame j and $j+1$. Please note the matched patches are not found by any means of explicit feature matching. Only the solved ego-motion and integrated feature states are used to establish the correspondences in sub-pixel accuracy.

5 Experimental Results

The proposed method is compared with direct VO and MFI approaches. We use the KITTI dataset [8], with the stereo monochrome images as input data,

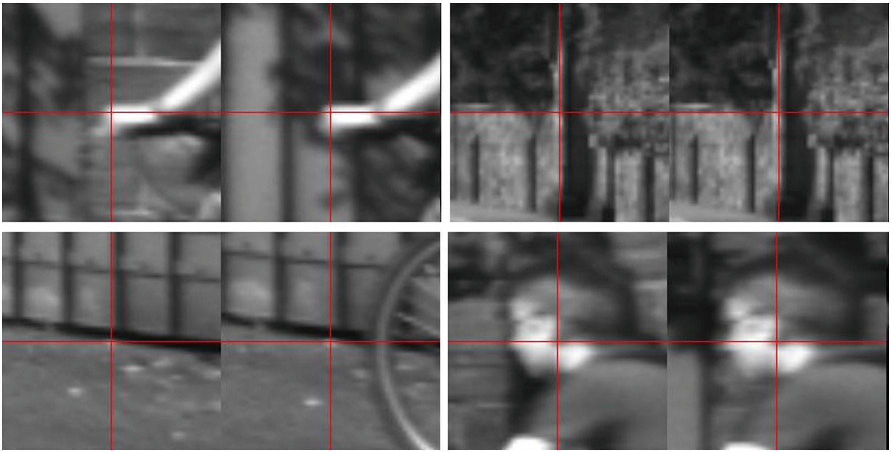

Fig. 4. Four lost features (always in the *left* image) recovered in another frame taken by another camera (see the *right* image of each pair) using the multi-camera multi-frame integration technique. The image patches are enlarged 5 times to illustrate the sub-pixel accuracy of the feature recovery algorithm

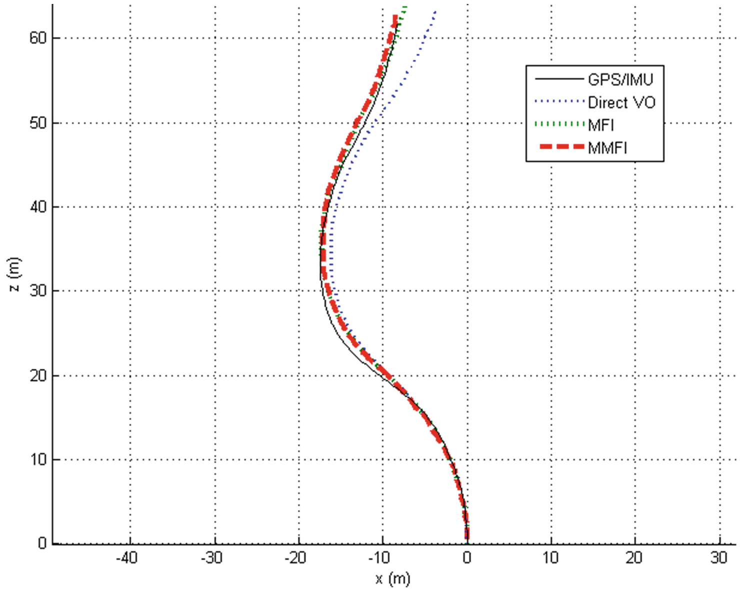

Fig. 5. Visual odometry of KITTI sequence 0005

Table 1. Ego-motion estimation errors and improvements

Sequence	Component	Unit	Direct	MFI	MMFI
0005	Rotation	deg/m	0.22	0.07 (68 %)	0.06 (72 %)
	Translation	%	9.42	6.10 (35 %)	4.49 (52 %)
0027	Rotation	deg/m	0.06	0.02 (67 %)	0.01 (83 %)
	Translation	%	13.2	5.79 (59 %)	2.74 (79 %)

and the GPS/IMU data as ground truth. The direct VO implementation uses only inter-frame feature matching of the left camera to solve ego-motion. The MFI method uses the same camera while the tracked features are integrated over time. The MMFI makes a full use of images taken from both cameras.

Table 2. Averaged time profile per frame

	MFI	MMFI
Feature matching	8.04 ms	38.61 ms
Feature integration	3.70 ms	11.49 ms
Ego-motion estimation	28.12 ms	26.34 ms
Lost feature recovery	5.83 ms	41.52 ms

The SURF features and descriptors (see [3], or presentation in [13]) are used for establishing inter-frame correspondences. The 3D data is acquired by means of a semi-global matching algorithm (SGM, see [10], or presentation in [13]), and stereo triangulation as implemented by OpenCV.

We illustrate results for the `2011_09_26_drive_0005` and `2011_09_26_drive_0027` sequences from the *City* and *Road* categories on KITTI. Each of these sequences includes traffic signs, trees, and moving objects. The cameras traveled about 70 and 380 metres, respectively, in those two test sequences. The estimated trajectories are drawn in Figs. 5 and 6. The ground truth is derived from the GPS and IMU sensors.

The motion errors are measured by dividing the trajectory into all possible subsequences of 10 %, 20 %, .., 100 % of the length of the sequence. The mean errors among all the subsequences are calculated and tabulated in Table 1. The errors are lower in the sequence **0005**, which is shorter than **0027**. The MFI achieved improvements of 45 % and 62 % over the direct VO implementation, respectively, in the sequences. The proposed MMFI method, on the other hand, improved the MFI method further by about 20 % in both cases.

The runtime of MFI and MMFI are profiled to study the impact on the efficiency when all the cameras are involved. Table 2 verifies that for most tasks the computation time increased by a factor of four, as three more inter- frame feature integrations are introduced by the MMFI. However, the computation time of the nonlinear optimisation for ego-motion estimation remains at the same level. This is a desired property as the stages of the original MFI, generalised in our work, can be easily parallelised for optimising the procedure. It is therefore possible to further improve the efficiency of MMFI by parallel implementation to match the original MFI method.

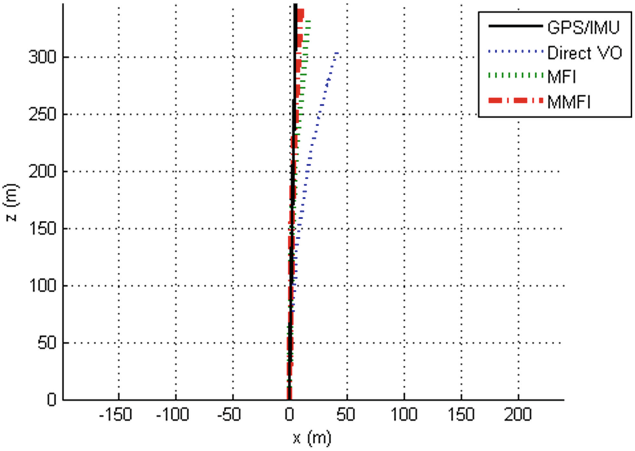

Fig. 6. Visual odometry of KITTI sequence 0027

6 Conclusions

In this paper we generalised the recently proposed multi-frame integration technique to make a full use of a multi-camera visual odometry system. The proposed approach enhances the robustness of the feature integration algorithm and achieves a better ego-motion estimation by taking into account multiple observations.

References

1. Badino, H., Yamamoto, A., Kanade, T.: Visual odometry by multi-frame feature integration. In: International Workshop on Computer Vision Autonomous Driving (ICCV) (2013)
2. Geiger, A., Ziegler, J., Stiller, C.: StereoScan: dense 3D reconstruction in real-time. In: Intelligent Vehicles Symposium (IV) (2011)
3. Bay, H., Van Gool, L., Tuytelaars, T.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
4. Choi, S., Kim, T., Yu W.: Performance evaluation of RANSAC family. In: Proceedings of the British Machine Vision Conference (2009)
5. Engels, C., Stewenius, H., Nister, D.: Bundle adjustment rules. In: Proceedings of Photogrammetric Computer Vision (2006)
6. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: Fast semi-direct monocular visual odometry. In: Proceedings of the IEEE International Conference on Robotics Automation, pp. 15–22 (2014)
7. Fraundorfer, F., Scaramuzza, D.: Visual odometry: part II - matching, robustness, and applications. IEEE Robot. Autom. Mag. **19**, 78–90 (2012)
8. Geiger A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of the Conference on Computer Vision Pattern Recognition (2012)
9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
10. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Proceedings of the Conference Computer Vision Pattern Recognition, vol. 2, pp. 807–814 (2005)
11. Jiang, R., Wang, S., Klette, R.: Statistical modeling of long-range drift in visual odometry. In: Koch, R., Huang, F. (eds.) ACCV 2010 Workshops, Part II. LNCS, vol. 6469, pp. 214–224. Springer, Heidelberg (2011)
12. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 486–492 (2010)
13. Klette, R.: Concise Computer Vision. Springer, London (2014)
14. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: an accurate O(n) solution to the PnP problem. Int. J. Comput. Vis. **81**, 155–166 (2009)
15. Scaramuzza, D., Fraundorfer, F.: Visual odometry: part I - the first 30 years and fundamentals. IEEE Robot. Autom. Mag. **18**, 80–92 (2011)
16. Zhang, Z., Shan Y.: Incremental motion estimation through local bundle adjustment. Technical report MSR-TR-01-54, Microsoft (2001)