Multi-Run: An Approach for Filling in Missing Information of 3D Roadside Reconstruction

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

[1] Department of Computer Science, The University of Auckland, Auckland, New Zealand
hgen001@auklanduni.ac.nz

[2] School of Engineering, Auckland University of Technology, Auckland, New Zealand

Abstract. This paper presents an approach for incrementally adding missing information into a point cloud generated for 3D roadside reconstruction. We use a series of video sequences recorded while driving repeatedly through the road to be reconstructed. The video sequences can also be recorded while driving in opposite directions. We call this a *multi-run* scenario. The only extra input data other than stereo images is the reading from a GPS sensor, which is used as guidance for merging point clouds from different sequences into one. The quality of the 3D roadside reconstruction is in direct relationship to the accuracy of the applied egomotion estimation method. A main part of our motion analysis method is defined by visual odometry following a traditional workflow in this area: first, establish correspondences of tracked features between two subsequent frames; second, use a stereo-matching algorithm to calculate the depth information of the tracked features; then compute the motion data between every two frames using a perspective-n-point solver. Additionally, we propose a technique that uses a Kalman-filter fusion to track the selected feature points, and to filter outliers. Furthermore, we use the GPS data to bound the overall propagation of the positioning errors. Experiments are given with trajectory estimation and 3D scene reconstruction. We evaluate our approach by estimating the recovery of (so far) missing information when analysing data recorded in a subsequent run.

Keywords: Multi-run scenario · Motion analysis · Visual odometry · Kalman filter · GPS data · 3D reconstruction · Multi-sensory integration

1 Introduction

Scene reconstruction plays an important role in many applications, including urban planning, route navigation, entertainment or gaming industry. Accurately reconstructing the 3D scene is still an extremely complex task in computer vision. Scientists and researchers introduced different methods and types of sensors for improving the egomotion (i.e. rotation and translation) estimation. For instance, a mobile terrestrial *light detection and ranging* (LiDAR) system can deliver

F. Huang and A. Sugimoto (Eds.): PSIVT 2015 Workshops, LNCS 9555, pp. 192–205, 2016.
DOI: 10.1007/978-3-319-30285-0_16

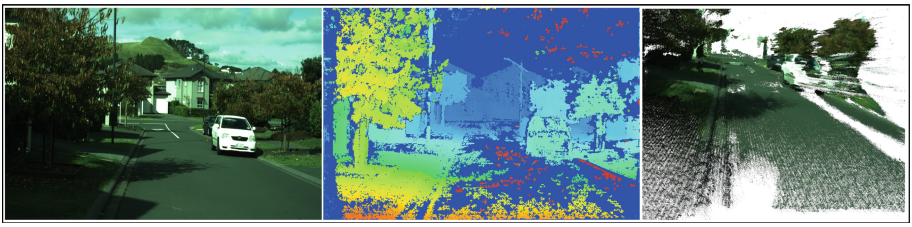

Fig. 1. Example of a disparity map (*middle*) generated from a given stereo image pair. Blank spaces in the point cloud (*right*) suggest occlusions or otherwise missing data for high-confidence stereo matching

highly detailed 3D data, the combination with a *global positioning system* (GPS) or an *inertial measurement unit* (IMU) are other examples of techniques that can provide reasonably good egomotion estimation. The use of optical cameras is the most cost-effective option. A camera-based approach can potentially provide reliability in most situations (e.g. for different weather conditions). The Mars Exploration rovers [13] are successful demonstrations of vision-based odometry.

A major problem for stereo vision is the occurrence of the object occlusions. Figure 1 shows a colour-coded disparity map with occlusions and possibly missing data. By using multi-run image sequences, our proposed approach aims at filling in the missing data, typically represented by 3D point clouds prior to surface triangulations or surface rendering.

Visual odometry (VO) is the key step of a long production pipeline of our multi-run approach. Scaramuzza et al. [20] indicated that VO methods can provide the trajectory estimation with a small error range between 0.1 % to 2 % of the actual motion. Olson et al. [18] suggested that the error of the VO methods is achievable to be less than 1 % of the total distance travelled. However, tiny errors that are caused by noise from the image data can build-up along a sequence quickly. Therefore, we apply Kalman filters to respond to the noise in the input data, in order to improve the overall egomotion estimation. In our research we also use multi-sensory integration to achieve an optimal and reliable solution for visual odometry in complex and dynamic environments.

In this paper, we propose an approach that reconstructs the scene over multi-run sequences. It takes stereo sequences as the major input, and GPS data as a type of supplementary input. The proposed VO method is the first phase of our approach; it focuses on estimating motion data between subsequent frames within a relatively small time interval. We implemented linear Kalman filter fusion to deal with the errors in the tracked features (i.e. one filter for one tracked feature), in order to ensure that only suitable feature candidates are used for motion estimations. In addition, a single extended Kalman filter is also applied for tracking the cameras' motion altogether.

The rest of the paper is structured as follows: Sect. 2 reviews previous work in the problem domain. Section 3 discusses our proposed approach and its mathematical theories. Section 4 shows how the additional GPS data can be used

to bound the build-up error within in a certain range over a long distance. Section 6 explains the design and measures used for the experiments and their evaluations. Section 7 concludes.

2 Related Literature

There are serval ways for estimating motion data in computer vision. We choose to focus on a stereo-vision-based method as the basis of our approach. Matthies et al. [14,15] demonstrated the benefits and tradeoffs of using a stereo-vision system over the use of a monocular vision system for calculating motion data. By computing and modelling the errors from the input, they found that the monocular vision system would contribute more errors in depth information. Stereo-matching algorithms are designed to generate disparity images that provide detailed depth information for the given stereo images. Demirdjian et al. [6] presented a method for motion analysis using disparity images generated from a stereo matching algorithm as the only input.

Stereo-matching algorithms commonly require rectified images, which are the output of a calibration procedure. However, calibration is one of the few approaches in computer vision where errors cannot be easily removed in subsequent processes. Hirschmüller et al. [9] state that even sub-pixel calibration errors can cause serious problems for the accuracy of structure reconstruction; they propose a method that can avoid calibration-error amplifications.

The Kalman filter, also known as linear quadratic estimation (LQE), is the most common way of dealing with unexpected errors or noise, which are introduced at different phases of a processing pipeline. With the grown of the complexity of motion analysis, the error-filtering task becomes a non-linear problem. The extended Kalman filter and the unscented Kalman filter have been developed for working with non-linear systems. Julier et al. [10] developed the unscented Kalman filter in 2004. Franke et al. [7] developed a more complex method; they used a Kalman filter fusion to track a number of image features, so that they can distinguish the features into foreground and background by motion data prediction. Based on the theory of Kalman filter fusion, Badino et al. [1] presented a novel method using a least-squares formulation to minimize the reprojection error of the overall egomotion estimation. Badino et al. [2] continued to improve their work into a real-time application. For further and more recent work, see [3].

The *Random Sample Consensus* (RANSAC) method is another commonly used approach for error filtering in feature matching. Kitt et al. [11] show that a RANSAC-based outlier rejection scheme can effectively improve the result of the motion estimation in a dynamic environment. Since feature tracking is considered to be the first step in most of the current visual odometry algorithms, the quality of feature detection directly affects the final estimation result. Song et al. [23] and [12] provide a comprehensive evaluation of mainstream feature detectors; these references discuss the best and worst scenarios for different types of feature detectors.

Roadside reconstruction clearly plays an important role in many advanced technologies and applications, such as navigation, visual reality, or driver-assistance system. Musialski et al. [17] provided an overview of current urban reconstruction methods. They stated that complex reconstruction problems remain to be unsolved to date, and that there is still a long way to go.

3 Visual Odometry Estimation

For our visual odometry method, we use a trinocular vision system to collect the image data. The recorded images are with 12-bit depth and 2046×1080 resolution for each camera. Figure 2(top) shows an example of recorded data. The sequences are usually recorded at 25 ∼ 27 Hz. In order to gain more overlapped regions in the point clouds for the multi-run scenarios, we decided to use the right image rather than the left image as the reference for the disparity map. The trinocular vision system also enables us to have a *third-eye evaluation approach*, see [5], to measure the consistency of any disparity values among the three cameras. Figure 2(bottom) shows the corresponding disparity map and the *transitivity-error-in-disparity-space* (TED) based-disparity-consistency map. The red pixels show the region with high confidence, whereas the blue pixels show the region with low consistency in disparity values.

The proposed visual odometry method mainly follows the traditional work-flow: (1) Feature matching and tracking. (2) Stereo matching. (3) Remove outliers with a RANSAC-based scheme. (4) Disguising foreground and background features. (5) Use static features to obtain the motion data by solving the *perspective-n-point* (PnP) problem. (6) Correct the trajectory with an unscented Kalman filter. (7) Optical and GPS data (i.e. multi-sensory) integration.

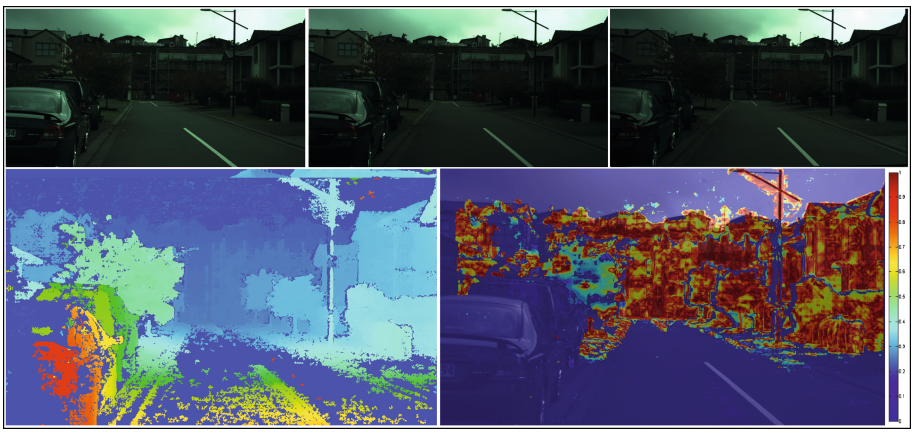

Fig. 2. *Top*: An input example for trinocular vision. *Bottom*: Corresponding color-enhanced disparity image, and its TED consistency map

The roadside reconstruction and the GPS trajectory will be projected into a left-hand local Cartesian coordinate system, where the origin is the shifted GPS position of the first frame. The z-axis is pointing forward into the initial default driving direction. The x-axis represents distance shifts from the origin to the left or right. The y-axis indicates changes in elevation. A positive value means a more elevated position. In the scene reconstruction coordinate system, we assume that the camera set is at a pre-defined position $[x, y, z]^\top = [\text{left}, \text{height}, 0]^\top$ at the beginning.

4 Error Handling and Kalman Filters

Optical cameras can provide fairly robust performance for all situations, but being relatively cheap also comes with some drawbacks (e.g. noise or geometric errors in image data). The ideal environment for egomotion estimation is that all the static features are perfectly detected and matched, and the disparity maps should provide perfect depth information for all the features. In reality, our world is never perfect, every aspects of the approach will bring a certain amount of errors. Noise filtering is an essential step for our algorithm; it should eliminate or at least bound the error propagation within an acceptable range.

In our proposed approach, we use Kalman filter fusion for filtering the errors in the feature matching phase to establish correct correspondences between two feature sets (in the *base* and in the *match image*). In addition, by tracking the features individually with extra depth information, the errors in the disparity and Euclidean space can be minimised. Therefore, the perspective transformation can be estimated more accurately.

We also propose an extended Kalman filter for tracking the rotation (Euler angles) and translation of the vehicle's motion. Julier et al. [10] suggested that the EKF is reliable for solving nonlinear problems that are almost linear. In our proposed method, we use the EKF for tracking changes of the egomotion transformation frame by frame, where the gaps (i.e. the time intervals) between every two frames are relatively small. Therefore, tracking and correcting the camera's trajectory and pose can be considered to be an almost linear problem, solvable with an EKF.

4.1 Linear Kalman Filter Fusion for Tracked Features

We propose local linear Kalman filter fusion for feature tracking: one filter for each of the continuously tracked features. Franke et al. [7] firstly used Kalman filter fusion for tracking image features in the 3D space, in order to remove the outliers by measuring the gap between actual and predicted positions of tracked features, and to classify features into foreground or background. This research shows that error minimisation in feature matching and tracking is the key factor that directly influences the quality of the final egomotion estimation. Therefore, we use a set of linear Kalman filters to keep track of all the selected features, and to detect any unexpected change in the tracked features pool. An unexpected

change can be, for example, a rapid change in depth or direction; the relevant features are then outliers and need to be discarded. We assume that the noise in the input data is white Gaussian noise, such as the noise introduced by the stereo matching algorithms.

State Vector. The state vector is a 12×1 vector; it contains the 3D positional data $[x, y, z]^\top$ and its velocity $[x', y', z']^\top$. Additionally, it contains the direction data $[\varphi_k, \theta_k, \psi_k]^\top$, and the corresponding angular speed $[\varphi'_k, \theta'_k, \psi'_k]^\top$. Thus, the state vector $\mathbf{x}_k$ is formed as follows:

$$\mathbf{x}_k = [x, y, z, x, y, z', \varphi_k, \theta_k, \psi_k, \varphi'_k, \theta'_k, \psi'_k]^\top \tag{1}$$

Process Model. The process model relates to the state vector; it describes the state vector change from the previous moment $k-1$ to the present moment k:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{A}_k & 0 \\ 0 & \mathbf{A}_k \end{bmatrix} \cdot \mathbf{x}_{k-1} + \mathbf{b}_k^\top + \mathbf{n}_k \tag{2}$$

$$\text{where} \quad \mathbf{A}_k = \begin{bmatrix} \mathbf{I}_3 & \Delta t \cdot \mathbf{I}_3 \\ 0_3 & \mathbf{I}_3 \end{bmatrix}$$

$\mathbf{A}_k$ is the state-transition matrix and $\mathbf{b}_k$ is the input-control vector.

Measure Model. The measurement is updated by the motion estimated from the last frame. The measurement model observes the current state of the positional and directional data. Thus, the measurement $\mathbf{z}_k$ is given as follows:

$$\mathbf{z}_k = \left[x_k, y_k, z_k, \varphi_k, \theta_k, \psi_k\right]^\top = \begin{bmatrix} \mathbf{H} & 0 \\ 0 & \mathbf{H} \end{bmatrix} \cdot \mathbf{x}_k + \mathbf{n}_k \tag{3}$$

$$\text{where} \quad \mathbf{H} = \begin{bmatrix} \mathbf{I}_3 & 0 \\ 0 & \mathbf{I}_3 \end{bmatrix}$$

$\mathbf{H}$ is the observation model matrix that translates the state vector to the measurement vector. Noise vector $\mathbf{n}_k$ represents white Gaussian noise.

4.2 Extended Kalman Filter for Multi-sensory Integration

The *extended Kalman filter* (EKF) is particularly designed to solve non-linear problems that are "almost" linear. However, the EKF could provide poor results when the prediction and update functions are highly non-linear. In our case, we propose to use the extra GPS data as the major guidance of the positional data of the reconstruction. We still use the continuous image sequences as the major source for motion estimation. The time interval between every two frames is small enough to consider it an 'almost' linear problem. The term 'global' refers to the overall motion transformation into the next state, and it is also refers and compares to the 'local' Kalman filter fusion discussed in Sect. 4.1.

State Vector. The state vector is a 12×1 vector; it contains the 6 elements for positional data and its velocity, and 6 elements for directional information. Thus, the state vector $\mathbf{x}_k$ is formed as above in the local filter, but here with a global meaning:

$$\mathbf{x}_k = [x, y, z, x', y', z', \varphi_k, \theta_k, \psi_k, \varphi'_k, \theta'_k, \psi'_k]^\top \tag{4}$$

Process Model. The process model of the EKF is formed by its common f and h functions. So, the state vector can be described by the EKF function

$$\mathbf{x}_k = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \tag{5}$$

and the measurement vector can be described by

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{6}$$

Extended Process and Measure Model. The state vector is given as

$$\mathbf{x} = [\mathbf{p}\top, \mathbf{p}'\top, \mathbf{w}\top, \mathbf{w}'\top]^\top \tag{7}$$

where vector $\mathbf{p}$ represents the positional and velocity information, vector $\mathbf{w}$ represents the angular data. Thus, the positional information can be extracted by the following equation:

$$\begin{bmatrix} \mathbf{p}_k \\ \mathbf{p}'_k \end{bmatrix} = \begin{bmatrix} I & \Delta t \cdot \mathbf{I}_3 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}'_{k-1} \end{bmatrix} \tag{8}$$

The measurement model needs additional normalization as follows:

$$\mathbf{z}_k = h(\mathbf{x}) = \begin{bmatrix} \mathbf{p} \\ normalize(\mathbf{w}) \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ \frac{\mathbf{w}}{|\mathbf{w}|} \end{bmatrix} \tag{9}$$

Then, the EKF cycle can start to iterate with the two given functions f and h.

5 Multi-run Merging and Integration

The proposed multi-run approach takes one or more independent sequences for the same street block or area, and uses them to find missing information in the total (so far) reconstructed point cloud. Figure 3 shows trajectories of the GPS signals for the four recorded sequences, each independent sequence is marked by a different colour. Theoretically speaking, taking more image sequences around the same block (more dense information) should allow us that the proposed approach achieves higher quality of the overall roadside reconstruction, but this will also come with performance and accuracy issues.

Multi-sensory integration is a solution to accuracy problems. It enriches the dimensions of the input data, so the different types of input data can be used either to evaluate each other's accuracy, or to increase the information density

in the input. For our approach, we use optical cameras (stereo-vision) and a GPS sensor. The image data are the main input of the proposed VO method. For the motion estimation phase, we choose to trust the image data over GPS data. GPS signals usually contain a 0.5 % error, due to a number of approximations, irregularities of the Earth surface, or deviations in GPS readings.

Since the error is irrelevant to travelled distance, so the GPS data can be used to bound the growth of the VO drift error. Once an unusual change is detected, or the growth of the drift error exceeds a tolerance threshold, then the GPS data will be used to correct the overall trajectory by the EKF.

Based on image and GPS data, we propose an approach for the multi-run reconstruction. Figure 4 shows the working pipeline of the proposed multi-run approach. First of all, we use the trinocular vision system to gather the required image data for the VO method, in order to estimate the motion data for every frame, and then create point cloud for 'one run' based on the known motion. Second, we pick the image data recorded in the same street block to construct the point cloud for the 'second run'. Then we use the GPS data of each sequence to roughly guide the merging action of the two point clouds into one 'overall' point cloud. After the rough merging, there are gaps between the two point clouds, so we use an ICP-based method to minimise the transformation between the two given point clouds. However, ICP has its own bottle-neck of performance issues, so we need to reduce the workload to an acceptable level. Thus, we introduce two key interest regions in the 3D reconstruction input: (1) Overlapping regions. (2) Corner regions.

5.1 Determination of Overlapping Regions

In order to reduce the processing time and the calculation workload, we need to determine some overlapping regions in the reconstructed point clouds. Then we take only the overlapped region (i.e. the "joint region") as the input to the ICP method to determine the best transformation between the two independent point clouds. The GPS coordinates are used as guidance for finding candidates of overlapped regions.

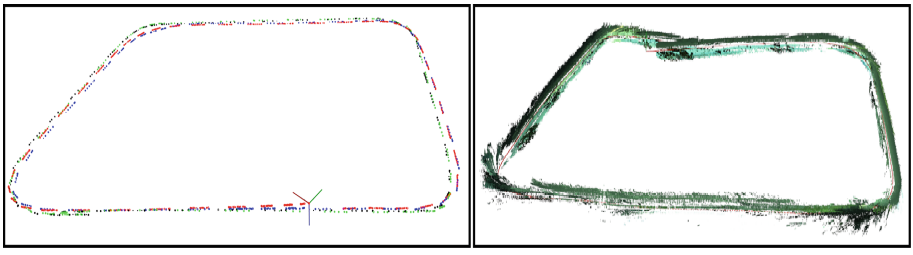

Fig. 3. *Left*: Trajectories of four subsequent runs around the same city block, distinguished by colours. *Right*: Corresponding 3D reconstruction of the city block (shown at very low resolution here, just indicating the reconstruction)

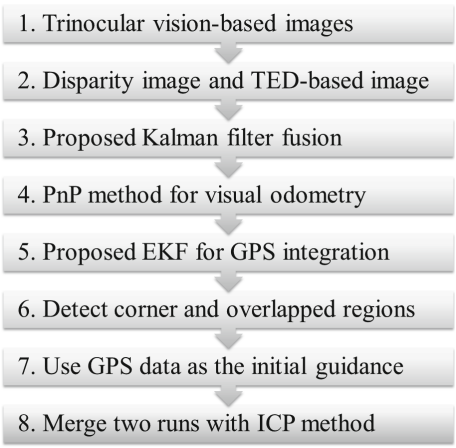

Fig. 4. The pipeline of the multi-run approach

5.2 Corner Region Recognition with GPS

The ICP method is used to find the precise transformation between the two point clouds for the multi-run scenario. However, the ICP method comes with performance limitations. A typical street block usually contains hundreds to thousands frames, which means the size of the point clouds can be too large to process. Due to the limited computation resource, we need to down-sample or reduce the total size of the input data for ICP. Therefore, we choose to use the corners around the block to estimate the transformation. Usually, the corner regions generate point clouds that contain very dense features and textures, because the vehicle needs to slow down at the turning point or corner region of any street block.

Fig. 5. An example of a detected street corner

This action will also gather rich information in all perspective view directions at the corner regions. Figure 5 shows an example of the 3D reconstructed point cloud at one detected corner region. Directly using GPS coordinates is also an efficient and effective way to detect any corners in the trajectory.

6 Results and Evaluation

The following assumptions are applied for our experiments: (1) the rectification process of the trinocular system is assumed to be done perfectly; (2) the recording platform is assumed to basically and only travel forward. The experiments have trajectory estimations and 3D scene reconstruction as generated outputs. We evaluate our approach by measuring the overall merging quality of the independent point clouds, and the percentage of the missing information that was recovered from a subsequent run.

The proposed approach and experiments are implemented in Visual C++ with OpenCV and *Point Cloud Library* (PCL). The feature detector used in our method is 'FAST', and the feature descriptor used is 'BRISK'. The 3D roadside reconstruction of the multi-run approach is considered to be the major experimental result. Its quality can be evaluated both visually and also by understanding whether more gaps (in the point cloud) are filled.

Figure 6 illustrates a 3D roadside reconstruction of a city block. The red dots show the projected positions of the trinocular camera set (i.e. its trajectory) for every recorded frame in the sketched 3D scene. This trajectory is estimated by

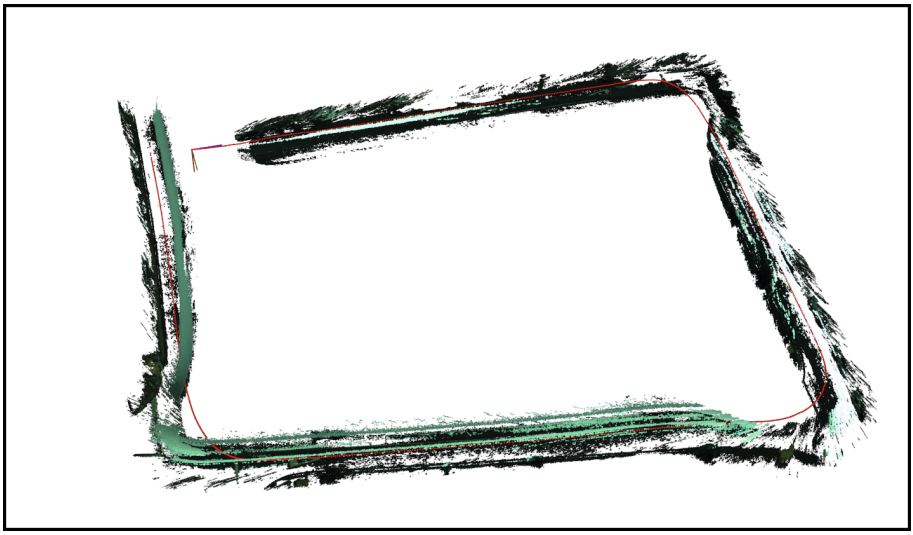

Fig. 6. Result of 3D roadside reconstruction without using GPS data. Red dots indicate the cameras' trajectory in the 3D scene (Color figure online)

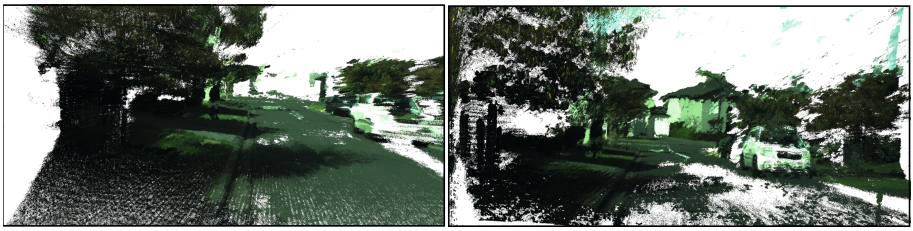

Fig. 7. Initial result for a multi-run approach (i.e. after the first run)

the proposed feature-matching-based VO method. The trajectory is first estimated by the PnP method, and then corrected and predicted by the proposed EKF. From the bird's-eye view of the city block, we can see that the roadside reconstruction, obtained from the sequence recorded in the first run, still contains many gaps and holes in the point cloud. They are inherited from the disparity images, such as from missing matching pixels, or occlusions.

Figure 7 (left) shows a region which was missing in the first-run point cloud. Figure 7 (right) illustrates the result of finding the missing data in the reconstructed point cloud after applying the second run sequences. The independent point cloud from the second run enhanced the overall reconstruction result. It brings in data, that is missing in the first-run point cloud, but also more noise.

Figure 8 shows the 3D reconstruction result after one run of a multi-run sequence set. The GPS data is also used here for bounding the total drift-error growth. However, in our experiments, the drift error is usually not big enough to activate a correction action of the EKF. The red dots are the projected positions of the converted GPS signals. The currently used GPS sensor provides discontinuous data, with time gaps of one second. Therefore, we designed the EKF to be only active when the GPS data is clearly on-line. Once the GPS signals are dropped, the weight of the GPS signals' reliability will be minimised. The red dots show the improved camera's trajectory that is done by the EKF.

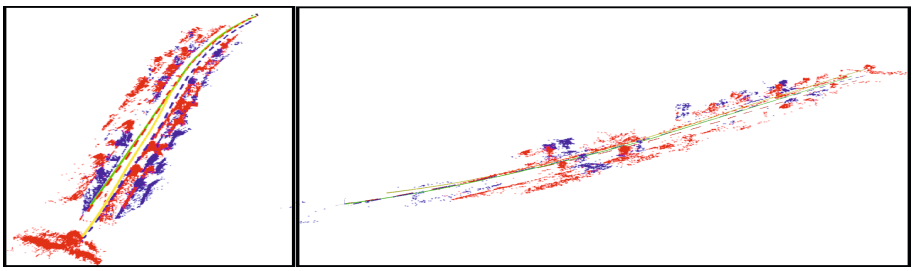

Fig. 8. 3D reconstruction by multi-run approach on the same street. Red and blue point clouds are the features; green and yellow lines are the camera trajectories; red or blue dotted lines are the GPS trajectories of the first or second sequence, respectively (Color figure online)

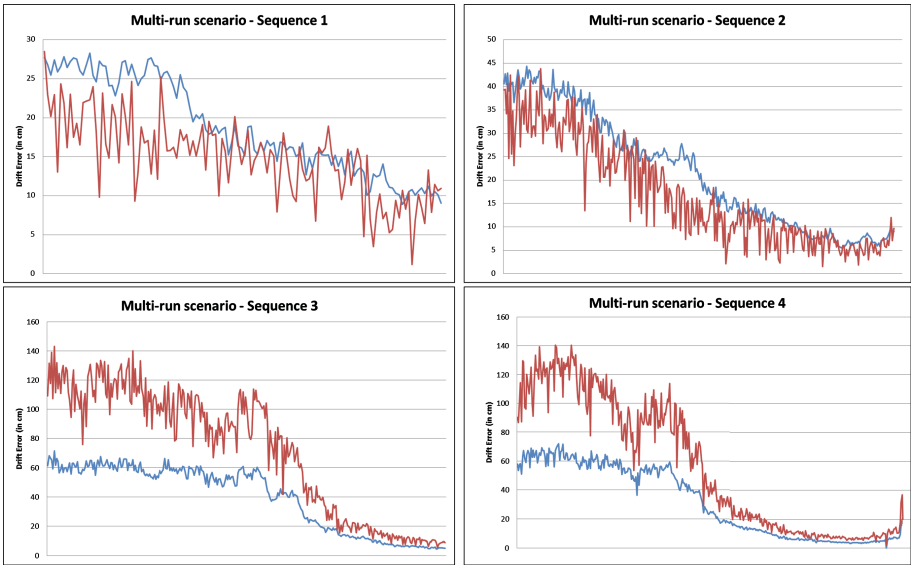

Fig. 9. Comparisons of drift errors for the four input sequences of the multi-run approach. The red and blue curve represents the differences in drift errors before and after the application of the EKF (Color figure online)

Figure 9 shows the detailed comparisons of drift errors before and after applying the EKF for the four input testing sequences. Sequences 1 and 2 are in the same direction, and so are Sequences 3 and 4. The figure demonstrates an effect: the testing sequences recorded in the same direction would more likely lead to similar behaviours of drift error generation and EKF correction. It also shows that the EKF correction will gain more control when the drift error gets larger. Each testing sequence contains around 1,400 stereo image pairs.

Our experimental results show that the multi-sensory method is a promising and valuable addition to any stand-alone VO applications. The extra positioning data can be either used as supplementary input or as evaluation measurements. Generally speaking, a stand-alone VO drift error builds up quickly over distance. For instance, even if there is only a small error occurring in the rotation, it will lead to a large drift error in both rotation and translation estimations. The applied filtering methods (e.g. RANSAC, Kalman filters, or bundle adjustment)can improve results for egomotion estimation, but some errors always remain.

Moreover, multi-sensor integration enriches the input data. The resulting visual odometry supports improved reconstruction; it helps to combine street segments according to the GPS data. Therefore, the 3D scene reconstruction can be done at a larger scale (i.e. in different sections in a large area). This demonstrates an alternative method for solving such a kind of problems, compared to other traditional VO methods. However, it also heavily relies

on the accuracy of the GPS input. If the GPS is accurate enough, our proposed method could use the extra input to bound the growth of the drift error within a relatively small range. It could also lead to a worse motion estimation if the GPS data has low accuracy.

7 Conclusions

In this paper, we propose an effective approach using *multi-run* sequences to fill in the missing data of 3D roadside reconstruction. The point clouds from different directions on the same street are merged together, to solve the missing data problems caused by the occlusion on the disparity images. GPS signals are used as the guidance of the camera set's rough position. It also is an important measurement that could bound the growth of the VO drift errors. The motion data is estimated mainly based on trinocular (stereo vision) image sequences.

In the phase of merging reconstructed point clouds, the ICP-based method appears to be the best option for precisely adjusting the transformation between two independent point clouds. It clearly shows that it can achieve more reliable results when the two point clouds have different perspective angles, compared to the 3D-feature-based method. However, the ICP-based method has performance limitations when the size of the input point clouds is "too large" (obviously a relative measure). Moreover, the ICP-based method can introduce more errors when the overlapped regions are not well-defined; the ICP algorithm will be trapped by the local optimisation problem. It means that the best adjustment description, calculated by the ICP-based method, might not be the *true* best adjustment description.

References

1. Badino, H., Franke, U., Rabe, C., Gehrig, S.: Stereo vision-based detection of moving objects under strong camera motion. Proc. Comput. Vis. Theor. Appl. **2**, 253–260 (2006)
2. Badino, H., Kanade, T.: A head-wearable short-baseline stereo system for the simultaneous estimation of structure and motion. In: Proceedings IAPR Config Machine Vision Applications, pp. 185–189 (2011)
3. Badino, H., Yamamoto, A., Kanade, T.: Visual odometry by multi-frame feature integration. In: Proceedings of the ICCV Workshop Computer Vision Autonomous Driving, pp. 222–229 (2013)
4. Besl, P., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**, 239–256 (1992)
5. Chien, H.J., Geng, H., Klette, R.: Visual odometry based on transitivity error analysis in disparity space - A third-eye approach. In: Proceedings IVCNZ, pp. 72–77 (2014)
6. Demirdjian, D., Darrell, T.: Motion estimation from disparity images. Proc. ICCV **1**, 213–218 (2001)

7. Franke, U., Rabe, C., Badino, H., Gehrig, S.K.: 6D-Vision: Fusion of stereo and motion for robust environment perception. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 216–223. Springer, Heidelberg (2005)
8. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. Int. J. Robot. Res. **32**, 1231–1237 (2013)
9. Hirschmüller, H., Gehrig, S.: Stereo matching in the presence of sub-pixel calibration errors. In: Proceedings CVPR, pp. 437–444 (2009)
10. Julier, S., Uhlmann, J.: Unscented filtering and nonlinear estimation. Proc. IEEE **92**, 401–422 (2004)
11. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In: Proceedings IEEE Intelligent Vehicles Symposium, pp. 486–492 (2010)
12. Klette, R.: Concise Computer Vision. Springer, London (2014)
13. Maimone, M., Cheng, Y., Matthies, L.: Two years of visual odometry on the Mars exploration rovers. J. Field Robot. **24**, 169–186 (2007)
14. Matthies, L.: Dynamic stereo vision. Ph.D. dissertation, Carnegie Mellon University (1989)
15. Matthies, L., Shafer, S.A.: Error modeling in stereo navigation. IEEE J. Rob. Autom. **3**, 239–250 (1987)
16. Milella, A., Siegwart, R.: Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In: Proceedings IEEE International Conference Computer Vision Systems, p. 21 (2006)
17. Musialski, P., Wonka, P., Aliaga, D., Wimmer, M., Gool, L., Purgathofer, W.: A survey of urban reconstruction. J. Comput. Graph. Forum **32**, 146–177 (2013)
18. Olson, C., Matthies, L., Schoppers, M., Maimone, M.: Stereo ego-motion improvements for robust rover navigation. Proc. IEEE Int. Conf. Robot. Autom. **2**, 1099–1104 (2001)
19. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to SIFT or SURF. In: Proceedings of the ICCV, pp. 2564–2571 (2011)
20. Scaramuzza, D., Fraundorfer, F.: Visual odometry tutorial. Robot. Autom. Mag. **18**, 80–92 (2011)
21. Shakernia, O., Vidal, R., Sastry, S.: Omnidirectional egomotion estimation from back-projection flow. Proc. CVPR Workshop **7**, 82 (2003)
22. Sibley, G., Sukhatme, G.S., Matthies, L.: The iterated sigma point Kalman filter with applications to long range stereo. In: Proceedings Robotics Science Systems (2006)
23. Song, Z., Klette, R.: Robustness of point feature detection. In: Wilson, R., Hancock, E., Bors, A., Smith, W. (eds.) CAIP 2013, Part II. LNCS, vol. 8048, pp. 91–99. Springer, Heidelberg (2013)