Egomotion Estimation by Point-Cloud Back-Mapping

Haokun Geng, Radu Nicolescu, and Reinhard Klette

Department of Computer Science, University of Auckland, New Zealand
hgen001@aucklanduni.ac.nz

Abstract. We consider egomotion estimation in the context of driver-assistance systems. In order to estimate the actual vehicle movement we only apply stereo cameras (and not any additional sensor). The paper proposes a visual odometry method by back-mapping clouds of reconstructed 3D points. Our method, called *stereo-vision point-cloud back mapping method* (sPBM), aims at minimizing 3D back-projection errors. We report about extensive experiments for sPBM. At this stage we consider accuracy as being the first priority; optimizing run-time performance will need to be considered later. Accurately estimated motion among subsequent frames of a recorded video sequence can then be used, for example, for 3D roadside reconstruction.

1 Introduction and Related Work

Computer vision techniques are widely used for solving problems that require extensive and precise geometric calculations. For example, driver-assistance systems (DAS) in the automotive industry require solutions of such problems. Computers are trained to listen, to see, and to sense the road geometry and the dynamic traffic environment. DAS are designed to provide comfort with safety, to assist drivers to follow traffic instructions, and to deal with road incidents. For instance, DAS should avoid that a sudden turn results in a pedestrian accident, or a braking manoeuvre in a collision. Demands for computer vision involved in DAS are increasing in future towards holistic scene understanding.

Motion data can be obtained by multiple types of sensors, including inertial measurement units (IMU), global positioning system (GPS) units, radar sensors, cameras, or laser range-scanners. Stereo cameras offer in principle economy and robustness, but require more advances in vision methodologies. Camera-based ego-motion estimation (also known as *visual odometry*) is the method we use to determine the trajectory of the *ego-vehicle* (i.e. the vehicle where the cameras are operating in). It is the first step of a whole pipeline of processes for understanding the road environment. Computationally it is an expensive task that requires massive observations and calculations. It estimates positional and directional data from input image pairs recorded at speeds of 25 Hz or more.

Nister et al. [13] firstly introduced visual odometry in 2004; it estimates odometry information based on recorded stereo image pairs only, not using other

data as, e.g., the vehicles' yaw rate or speed. An advantage of visual odometry is that it avoids the influence of motion estimation errors in other sensors, e.g. the influence of mechanical issues such as wheel slips, or the still existing inaccuracy of (cost-efficient) GPS or IMU sensors. Scaramuzza et al. [16] suggest that visual odometry methods usually lead to a smaller relative position error (in a range between 0.1% to 2% of actual motion), compared to traditional wheel odometry methods. [10] presents a vision-based ego-motion application used in Mars exploration rovers; it demonstrated the great capability of this technology on another planet for the first time. With these advantages and examples, vision-based ego-motion analysis proves itself as being a valuable navigation technology, and a potential feature of mobile computer vision applications.

This paper presents a novel vision-based ego-motion estimation method. 3D point clouds are calculated based on generated disparity maps. In this paper we apply an *iterative semi-global stereo matching* (iSGM) method as introduced in [7] in 2012; see also [8] for a description of the iSGM algorithm.

Due to the types of used video input data, existing visual odometry methods can be divided into three main categories: monocular, stereo, and omnidirectional. Each of the three types is designed for particular problem domains. For solving our problem, we choose to focus on stereo-vision methods. A solid mathematical foundation of for navigation using stereo-vision methods has been published in [11, 12].

Following those basic contributions, various studies have been carried out, from 2D-to-2D matching to 2D-to-3D registration, and finally to 3D-to-3D motion estimation. A method for solving a 3D-to-3D point registration problem was presented in [6]. The given algorithm estimates the stereo cameras' 3D rigid transformation by directly taking the disparity images as the input. [14] defined concepts of two-frame motion field estimation; the paper demonstrated a novel approach for calculating 3-dimensional scene flow data using a Kalman filter and parallel implementations.

Visual odometry depends on feature detection. A good feature detector is a critical factor for improving the estimation results. [17] carefully evaluated a number of popular 2D feature detectors; the results showed that the *oriented BRIEF* (ORB) detector (see [15] for its implementation and, e.g., [8] for a definition) gave the best results among all available (in mid 2013) detectors in OpenCV when also requesting time efficiency. Rotation invariance and noise resistance of the ORB detector comes close to SIFT performance, but at a much better time efficiency of ORB compared to SIFT.

The iterative closest point (ICP) method is typically used to find the translation and rotation between two clouds of points by minimising geometric differences between them. This algorithm was firstly implemented in 1992 by Besl and McKay; see [3]. The authors of [5] report an improved extension of the popular ICP algorithm using a least trimmed squares (LTS) approach.

The Kalman filter is a tool for minimising estimation errors caused by white noise of input data; it effectively improves both the robustness and the regularity (smoothness) of calculated trajectories. The authors of [1] designed a robust ego-

motion algorithm with an extended Kalman filter for 3D position and velocity estimation; they continued their work and successfully implemented a head-wearable stereo system with real-time ego-motion estimation capability reported in [2].

In our proposed method, we consider the pair-wise registration task to be a linear problem, since the time interval between every two input frames is relatively small (approximately 0.03 seconds). Our method is different from other existing work with respect to three aspects. First, it uses a motion layer to remove all the moving features between every two frames. Second, it applies a segmentation-cut method to minimise typical errors occurring in stereo matching. Third, it measures a number of local transformations with calculated weights, in order to sum up an accurate global transformation.

2 Ego-motion Estimation Method

A stereo-cameras system is mounted inside the car just behind the rear mirror.[1] Our proposed method consists of the following main components: 2D feature matching, 3D feature triangulation and mapping, motion estimation, and noise minimising with a Kalman filter.

2.1 Disparity Smoothness Issues

Stereo matching applies a data cost and a continuity cost function [8] when calculating the location difference (i.e. the *disparity*) between a pair of corresponding pixels in the left and right image. Our method of choice is iSGM (see [7]) for generating the disparity maps of recorded stereo frames. Calculated disparity maps generally contain noise, miss-match errors, and gaps (due to occlusions and low confidence). In order to improve the availability of dense and accurate disparity data, we implemented an "aggressive" method for smoothing disparity values for any given tracked feature.

For any subsequences of subsequent s frames (we use $s = 5$ in the formula below and experiments), the disparity value of a traced feature is bound by the following rules:

$$d_{(k\cdots k+5)} = \begin{cases} -1 & \text{if } \; d_{\triangle total} < |d_k - d_{k+5}| \\ k \cdot C_d \cdot + d_0 & \text{if } \; d_{\triangle total} \geq |d_k - d_{k+5}| \end{cases} \tag{1}$$

where C_d is the mean value of changes in disparity values among considered subsequences of length s. We consider a linear model for the ego-vehicle's motion.

In order to minimise the errors (or uncertainties) for different estimation scenarios, we propose a feature selection scheme. Features with a smaller disparity are put into a candidate set for estimating rotational measurements. Features with larger disparity values are used to calculate the translational component of the ego-vehicle's motion. The scheme uses a threshold (to create the two candidate sets) which is the median value of disparities of all detected features.

[1] In our test system, the cameras record sequences as 12-bit grey-level images at 30 Hz with a dimension (width times height) of 2046×1080 pixels for both the left and right images.

2.2 From Local to Global Transformations

There is a common observation for any reconstructed cloud of 3D points: the white noise is mainly distributed along the Z-axis. In order to minimise the errors, we segment those clouds of 3D points and smooth disparities. For any set of 3D points on a local (triangulated) segment, we specify two values: (1) The depth value d_{edge} of its nearest edge element, and (2) the smallest depth value $d_{smallest}$ inside the segmented region. Based on those two values, we then calculate a smoothed depth value to replace a measured value as follows:

$$Z_{smooth} = Z_{edge} - (\frac{Z_{edge} - Z_{smallest}}{d_{edge} + d_{smallest}}) \cdot d_{edge} \tag{2}$$

A pair of local segments (between subsequent frames) generates a local transformation. Every local transformation has a weight for finally defining the global transformation. The weight is obtained as follows:

$$\omega(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3}$$

where $\mu = 0$, $\sigma = 150$, and x is the distance to the cameras within an assumed range of $(0, 150]$ meters. The global transformation is then defined by

$$T_{global} = \sum (\omega_x \cdot T_{local}) \; / \sum \omega_x \tag{4}$$

Improved translation and rotation parameters $(\mathbf{R}, \mathbf{t})$ can be obtained by applying least-square optimization with two sorted sets of n matched features from Frame k (feature $m_{k,i}$) to Frame $k+1$ (feature $m_{k+1,i}$):

$$E = \arg \min_{(\mathbf{R},\mathbf{t})} \sum_{i=1}^{n} |P(m_{k+1,i}) - P(\mathbf{R}_{k,i} \cdot m_{k,i} - \mathbf{t}_{k,i})| \tag{5}$$

The projection function $P(m_k)$ is defined as follows:

$$P(m_k) = \frac{f \cdot b}{d} \cdot [u, v, 1]^\top = [X, Y, Z]^\top \tag{6}$$

for a feature m_k detected at (u, v) in the input image. Here, f is the focal length of the corresponding camera, b is the length of the baseline, and d is the disparity value of the given feature.

2.3 Extended Kalman Filter

In order to minimise the Gaussian noise in the estimation of the ego-vehicle's motion, we propose to use an iterated extended Kalman filter (IEKF) to solve this problem. As introduced by Julier et al [9], an IEKF is designed for filtering noise in nonlinear calculations, but it is also reliable for solving problems that

are almost linear. Since the time interval between each two subsequent frames is relatively small, which is defining an nonlinear problem but "almost" linear.

Process Model. The process model of our method follows the general form

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{A}_k & 0 \\ 0 & \mathbf{A}_k \end{bmatrix} \cdot \mathbf{x}_{k-1} + \mathbf{B}_k^\top + \mathbf{n}_k \tag{7}$$

of a Kalman filter where the state vector $\mathbf{x}_k$ contains the camera's pose given by world coordinates (X, Y, Z) and Euler angles φ, θ, and ψ, extended by changes (X', Y', Z') and φ', θ', and ψ'. The Euler angles are calculated for pitch, roll, and yaw. For calculating the current state vector $\mathbf{x}_k$, we combine the positional and directional data into a state-transformation matrix as follows:

$$\mathbf{A}_k = \begin{bmatrix} 1 & 0 & 0 & \triangle t & 0 & 0 \\ 0 & 1 & 0 & 0 & \triangle t & 0 \\ 0 & 0 & 1 & 0 & 0 & \triangle t \\ 1 & 0 & 0 & \triangle t & 0 & 0 \\ 0 & 1 & 0 & 0 & \triangle t & 0 \\ 0 & 0 & 1 & 0 & 0 & \triangle t \end{bmatrix} \tag{8}$$

Vector $\mathbf{B}_k$ is the optimal control-input of the Kalman filter.

Measurement Model. A measurement is the observed camera translation and its rotation. As we assumed a linear model, the new translation and rotation is directly obtained by our ICP method. Thus, the relation between the current state vector $\mathbf{x}_k$ and the measurement state vector $\mathbf{z}_k$ is described as follows:

$$\mathbf{z}_k = \mathbf{H} \cdot \mathbf{x}_k + \mathbf{n}_k = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_6 \end{bmatrix} \cdot \mathbf{x}_k + \mathbf{n}_k \tag{9}$$

where $\mathbf{R}_x$, $\mathbf{R}_y$, and $\mathbf{R}_z$ are the rotation matrices calculated from the three relevant Euler angles. The variable $\mathbf{n}_k$ represents the measurement error assumed to be Gaussian noise with zero-mean.

An implemented Kalman filter consists of two major steps in one calculation: prediction and correction. In the prediction step, it predicts the projections of the current state vector to the next state, in order to obtain a so-called *'priori'* estimation. In the correction step, it measures the optimal Kalman gain, then updates the '*posteriori*' state based on the calculated Kalman gain.

Prediction. Between every two frames, it is required to estimate a *priori* state vector $\tilde{\mathbf{x}}_{k+1|k}$:

$$\tilde{\mathbf{x}}_{k+1|k} = \mathbf{A}_k \cdot \tilde{\mathbf{x}}_k + \mathbf{B}_k \tag{10}$$

and a *priori* projection noise covariance matrix $\mathbf{P}_{k+1|k}$:

$$\mathbf{P}_{k+1|k} = \mathbf{A}_k \cdot \mathbf{P}_k \cdot \mathbf{A}_k^\top + \mathbf{Q}_k \tag{11}$$

Variable $\mathbf{Q}_k$ is the covariance of the uncertainty in the measurement.

Correction. First we calculate the Kalman gain for the next state based on Equ. (9).

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^{\top} \cdot (\mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^{\top} + \mathbf{R}_k)^{-1} \tag{12}$$

Variable $\mathbf{R}_k$ is the covariance of white noise $\mathbf{n}_k$ in the observational data, the residual covariance $\tilde{\mathbf{n}}_{k+1}$ can be measured as follows:

$$\tilde{\mathbf{n}}_{k+1} = \mathbf{z}_{k+1} - \mathbf{H}_{k+1} \cdot \tilde{\mathbf{x}}_{k+1|k} \tag{13}$$

Finally, we update the '*posteriori*' state vector:

$$\tilde{\mathbf{x}}_{k+1|k+1} = \tilde{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \cdot \tilde{\mathbf{n}}_{k+1} \tag{14}$$

and the '*posteriori*' projection noise covariance matrix:

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{P}_{k+1|k} \tag{15}$$

3 Results and Evaluation

The evaluation is carried out following camera calibration, which also provides the cameras intrinsic parameters, and time-synchronised stereo-pair video recording. In this paper we report about a sequence of 1,246 stereo frames recorded at 30 Hz; see Fig. 1. The calibrated focal length is 2908.86 pixels, the baseline length is 0.338903 m. Frame dimensions are 2046×1080 pixels. The proposed method is implemented in C++ with OpenCV and PCL technology on a standard laptop computer. The tracked features are estimated using the Lucas-Kanade algorithm; see, e.g. [4, 8].

Figure 2 presents a dense 3D reconstruction of street segments with the shown camera trajectory. Figure 3 and Fig. 4 show the tracked features' incremental registration in the scene. We measure the re-projection error, frame by frame, in order to evaluate the accuracy of the ego-motion estimation. The result shows that the mean of the re-projection errors for these two subsequences are all under 2 pixels, which is acceptable for driver-assistance purposes, but possibly not yet ideal for the very challenging road-side reconstruction as pioneered in [18].

Due to currently (still) non-availability of sources for previously published ego-motion algorithms (e.g. for [2]) it remains a future project to compare results.

Fig. 1. *Left*: Subsequence when driving straight. *Right*: Subsequence while driving around a roundabout.

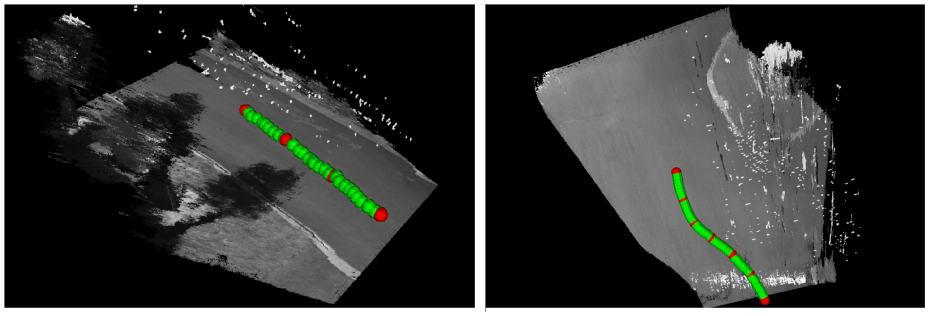

Fig. 2. Estimated camera trajectory with corresponding 3D road reconstruction. *Left*: Driving-straight subsequence. *Right*: Roundabout subsequence.

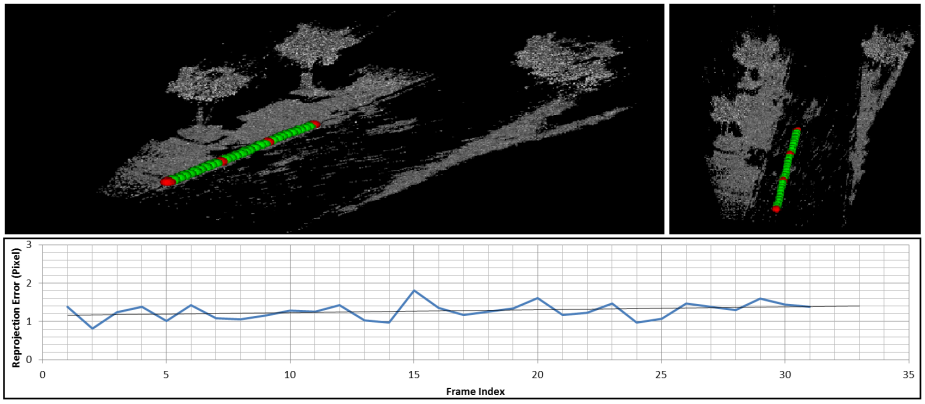

Fig. 3. Tracked features in a 3D representation for the driving-straight subsequence.

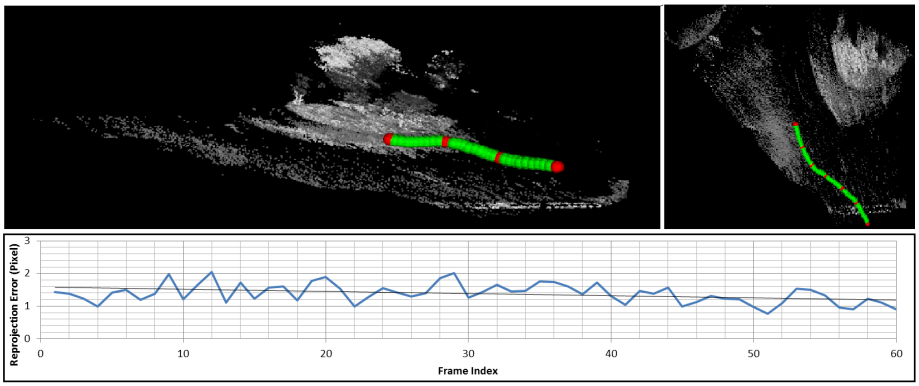

Fig. 4. Tracked features in a 3D representation for the roundabout subsequence.

4 Conclusions

The reason why we choose an ICP-based method is that it has the potential to be extended for adding or removing non-static objects in a traffic scene, and for

mapping changes of road environments over time. As a next step, a 3D feature outlier removal scheme will be designed and implemented for more accurate estimation. As mentioned before, one type of sensors alone (here: stereo cameras) might solve the problem under various conditions, but combining gathered data with those recorded by another sensor (e.g., a GPS unit) is expected to improve the robustness of our proposed method. The ultimate goal is to estimate accurate ego-motion of the vehicle for all circumstances when driving a car, so that it could provide a solid foundation for higher-order applications of the driver assistance system such as holistic traffic scene understanding.

References

1. Badino, H., Franke, U., Rabe, C., Gehrig, S.: Stereo vision-based detection of moving objects under strong camera motion. In Proc. Int. Conf. Computer Vision Theory Applications, 25 – 28 (2006)
2. Badino, H., Kanade, T.: A head-wearable short-baseline stereo system for the simultaneous estimation of structure and motion. In Proc. Conf. Machine Vision Applications (2011)
3. Besl, P. J., McKay, N. D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Analysis Machine Intelligence, 14 (2), 239 – 256, (1992)
4. Bouguet, J. Y.: Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm. Intel Corporation, Microprocessor Research Labs, USA (2000)
5. Chetverikov, D., Svirko, D., Stepanov, D., Krsek, P.: The trimmed iterative closest point algorithm. In Proc. ICPR, 3, 545 – 548 (2002)
6. Demirdjian, D., Darrell, T.: Motion estimation from disparity images. In Proc. ICCV, 1, 213 – 218 (2001)
7. Hermann, S., Klette, R.: Iterative semi-global matching for robust driver assistance systems. In Proc. ACCV (2012)
8. Klette, R.: *Concise Computer Vision*, Springer, London (2014)
9. Julier, S. J., Uhlmann, J. K.: Unscented filtering and nonlinear estimation. Proc. IEEE, 92(3) 401 – 422 (2004)
10. Maimone, M., Cheng, Y., Matthies, L.: Two years of visual odometry on the Mars exploration rovers. J. Field Robotics, 24, 169 – 186 (2007)
11. Matthies, L., Shafer, S.: Error modeling in stereo navigation. IEEE J. Robotics Automation, 3, 239 – 248 (1987)
12. Matthies, L.: Dynamic stereo vision. Ph.D. dissertation, Carnegie Mellon University (1989)
13. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. Proc. ICVPR, 652 – 659 (2004)
14. Rabe, C., Muller, T., Wedel, A., Franke, U.: Dense, robust and accurate 3D motion field estimation from stereo image sequences in real-time. In Proc. ECCV, 5 – 11 (2010)
15. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In Proc. ICCV, 2564 – 2571 (2011)
16. Scaramuzza, D., Fraundorfer, F.: Visual odometry tutorial. Robotics Automation Magazine, vol. 18(4), pp. 80 – 92 (2011)
17. Song, Z., Klette, R.: Robustness of point feature detection. In Proc. CAIP, 91 – 99 (2013)
18. Zeng, Y., Klette, R.: Multi-run 3D streetside reconstruction from a vehicle. In Proc. CAIP, 580–588 (2013)