Visual Odometry by Multi-frame Feature Integration

Hernán Badino
hbadino@cs.carnegie.edu

Akihiro Yamamoto*
akihiroy@cs.cmu.edu

Takeo Kanade
takeo.kanade@cs.cmu.edu

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA

Abstract

This paper presents a novel stereo-based visual odometry approach that provides state-of-the-art results in real time, both indoors and outdoors. Our proposed method follows the procedure of computing optical flow and stereo disparity to minimize the re-projection error of tracked feature points. However, instead of following the traditional approach of performing this task using only consecutive frames, we propose a novel and computationally inexpensive technique that uses the whole history of the tracked feature points to compute the motion of the camera. In our technique, which we call multi-frame feature integration, the features measured and tracked over all past frames are integrated into a single, improved estimate. An augmented feature set, composed of the improved estimates, is added to the optimization algorithm, improving the accuracy of the computed motion and reducing ego-motion drift. Experimental results show that the proposed approach reduces pose error by up to 65% with a negligible additional computational cost of 3.8%. Furthermore, our algorithm outperforms all other known methods on the KITTI Vision Benchmark data set.

1. Introduction

Accurate estimation of the motion of a mobile platform is an essential component of many robotic and automotive systems, and visual odometry is one of the most accurate ways of obtaining it. Visual odometry has been gaining increasing popularity over the last decade, as evidenced by the large number of publications on the topic [24] as well as the release of open data sets made available for objective performance comparison [8, 18]. In this paper, we present a novel stereo-based visual odometry method that provides state-of-the-art results in real time, both indoors and outdoors.

Our proposed method follows the procedure of computing optical flow and stereo disparity to minimize the re-projection error of tracked feature points. However, instead of performing this task using only consecutive frames (e.g., [3, 21]), we propose a novel and computationally simple technique that uses the whole history of the tracked features to compute the motion of the camera. In our technique, the features measured and tracked over all past frames are integrated into a single improved estimate of the real 3D feature projection. In this context, the term *multi-frame feature integration* refers to this estimation and noise reduction technique.

This paper presents three main contributions:

- A statistical analysis of the feature tracking error that helps us identify its properties and design an appropriate model to reduce tracking drift (Section 4.1).
- A feature propagation technique that reduces the ego-motion drift over time while maintaining a high inter-frame motion accuracy (Section 4.2).
- A predictor/corrector technique to detect and correct tracking errors (Section 4.4).

Our experimental results in Section 5 show that our new proposed approach reduces pose error by up to 65% with a negligible additional computational cost of 3.8% over the baseline algorithm. Furthermore, our algorithm shows an average translational error of 1.62% of the traveled distance and 0.0062 deg/m rotational error on the KITTI Vision Benchmark data set [8], outperforming all other known visual odometry methods[1].

2. Related Literature

The literature on visual odometry computation is huge, and we will review only the most representative publications. A complete and detailed review can be found in [24].

*Currently at Renesas Electronics Corporation, Japan.

[1] http://www.cvlibs.net/datasets/kitti/eval_odometry.php

International Workshop on Computer Vision for Autonomous Driving (CVAD 13)
In conjunction with ICCV 2013, Sydney, Australia

The first to approach the estimation of camera motion from visual data was Moravec [20] establishing the pipeline of the structure from motion problem. Since then, a variety of different approaches have been proposed in the literature. Methods relying on inter-frame point correspondences typically use detectors/descriptors such as SIFT [5], SURF [15], FAST [11], Harris [13, 21], and even custom designed detectors [9]. Optical flow methods such as KLT [23] and dense scene flow [2] are also typically used. Strategies such as the bucketing technique [28] further help distribute the features more uniformly on the image space to improve the conditioning of the optimization problem [15]. Other techniques involve segmenting features based on distance to solve rotational and translational components independently [13], or even avoiding explicit triangulation using quadrifocal image constraints [6].

Ego-motion drift reduction is an important property of visual odometry approaches. The most popular ways of reducing drift are Simultaneous Localization and Mapping (SLAM) and Bundle Adjustment (BA). SLAM methods [7, 16] can reduce the drift by detecting loop-closure in cases where the same scene is visited more than once. BA methods optimize only camera poses and the position of features over a number of recent frames [27]. Since the computational cost rapidly increases with the number of frames [24], a small number of frames is usually used for real-time applications [26].

The fusion of visual odometry with other positioning or motion sensors such as GPS [23], absolute orientation sensor [22], or IMU [12, 17, 26] can improve the positioning accuracy.

Our new proposed stereo-based method differs from previous work through the introduction of an augmented feature set that contains the accumulated information of all tracked features over all frames, allowing the reduction of the drift of the estimated motion. In contrast to BA, the computational cost is not only negligible in absolute terms, but also independent of time and linear with the number of tracked features. Additionally, our proposed drift reduction technique is simple and can be easily incorporated into most visual odometry methods.

3. Visual Odometry Estimation

In this section, we introduce the baseline stereo visual odometry algorithm to define the mathematical symbols that will be used later in the paper. We follow here the same approach as in [3].

3.1. Least Squares Solution

Our algorithm follows the standard dead-reckoning approach of estimating the rigid body motion that best describes the transformation between the sets of 3D points acquired in consecutive frames. The total motion over time is then obtained by the concatenation of the individual motion estimates.

A set of tracked feature points $\boldsymbol{m}_{i,t} = (u, v, d)^T$ for $i = 1, 2, ..., n$ from the current frame and the corresponding set of feature points, $\boldsymbol{m}_{i,t-1} = (u', v', d')^T$ for $i = 1, 2, ..., n$, from the previous frame are obtained, where (u, v) is the detected feature location on the left image and d is the stereo disparity. The goal of the visual odometry problem is to estimate the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ that satisfy the rigid body motion of the tracked points, i.e.,

$$\boldsymbol{g}(\boldsymbol{m}_{i,t}) = \boldsymbol{R}\,\boldsymbol{g}(\boldsymbol{m}_{i,t-1}) + \boldsymbol{t} \tag{1}$$

where $\boldsymbol{g}()$ is the stereo triangulation equation that calculates the 3D point in Euclidean space. Instead of minimizing the residuals in Euclidean space, where error covariances are highly anisotropic [19], a better approach is to work on the image space where the error is similar in all three dimensions (see Figure 1). In order to get the objective function, we first apply the projection equation $\boldsymbol{h}() = \boldsymbol{g}^{-1}()$ to both sides of Equation 1 to obtain

$$\boldsymbol{m}_{i,t} = \boldsymbol{r}(\boldsymbol{m}_{i,t-1}) \tag{2}$$

with

$$\boldsymbol{r}(\boldsymbol{m}_{i,t-1}) = \boldsymbol{h}(\boldsymbol{R}\,\boldsymbol{g}(\boldsymbol{m}_{i,t-1}) + \boldsymbol{t}). \tag{3}$$

In general, Equation 2 will not hold under the presence of noise, for which the weighted residual is

$$e(\boldsymbol{m}_{i,t-1}, \boldsymbol{m}_{i,t}) = w_i ||\boldsymbol{r}(\boldsymbol{m}_{i,t-1}) - \boldsymbol{m}_{i,t}||, \tag{4}$$

with a reprojection error variance of w_i^{-2}. Note that we apply a simple scalar weighting. We found experimentally that using the full inverse covariance matrix has little impact on the motion estimate. The least squares objective function is then

$$\sum_{i=1}^{n} e(\boldsymbol{m}_{i,t-1}, \boldsymbol{m}_{i,t})^2. \tag{5}$$

In order to minimize Equation 5, the rotation matrix $\boldsymbol{R}$ is parametrized by the rotation vector $\boldsymbol{r} = (\omega_x, \omega_y, \omega_z)^T$ and the translation vector as $\boldsymbol{t} = (t_x, t_y, t_z)^T$. Thus, the parameter for minimization is the six-dimensional vector $\boldsymbol{x} = (\omega_x, \omega_y, \omega_z, t_x, t_y, t_z)^T$.

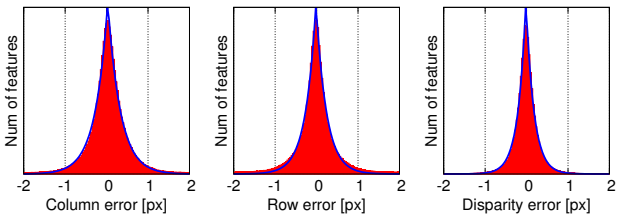


Figure 1. Learned pdf of the feature inter-frame tracking error. Red: measured. Blue: fitted Laplacian pdf.

The least squares solution of Equation 5 is computed using the Newton optimization method. Outliers are handled by an iterative least squares procedure as described in [3].

4. Reducing Drift by Multi-frame Feature Integration

Visual odometry is a dead-reckoning algorithm [14] and, therefore, prone to cumulative errors. The current pose of the platform is obtained from the previous pose by adding the last observed motion, which leads to a super-linear increment of the pose error over time, as shown in [22].

One solution to this problem is to compute visual odometry as a bundle adjustment algorithm [27]. BA imposes geometrical constraints over multiple frames, thus providing a global optimal estimate of all camera poses at once. However, the computational cost of BA algorithms increases with the cube of the number of frames used for computation. A common approach, known as local BA, uses a small number of frames to limit the computational complexity [26]. In this section, we present a novel method that uses the whole history of the tracked features with a computational complexity that increases linearly with the number of tracked features and is independent on the number of frames.

In the next sections, we will first analyze the characteristics of the tracking error and then propose a simple method to reduce tracking error and, consequently, ego-motion drift error.

4.1. Characteristics of Measured Features

Tracking features from frame to frame in sequence is also a dead-reckoning process and, therefore, affected by the same accumulation of errors. The process of tracking usually requires detecting a feature in the first image and re-detecting it in the second image. The same process is then repeated between each pair of consecutive images. Every tracking step adds a cumulative error to the feature position, which is propagated to the estimated motion of the camera through the non-linear system that relates feature positions to camera motion (i.e., Eqs. 2-5).

In order to understand this cumulative tracking error, we have performed an experiment using the synthetic New Tsukuba dataset [18]. In this experiment, 4,096 features are detected with the Harris corner detector [10]. FREAK descriptors [1] are computed on the detected keypoints and matched between consecutive frames by brute-force combinatorial search. When a new feature is first detected (i.e., it has no previous associated keypoint), the corresponding original ground truth 3D point is obtained. At each frame, the ground truth projection of the original 3D point is calculated using the ground truth motion. Finally, the ground truth projection is then compared with the measured position of the feature. We have performed two analyses. In the first analysis we found the probability density function of the error, while in the second analysis we evaluated the dependency on time of the tracking error.

Figure 2. RMSE of inter-frame feature position as function of the survival age.

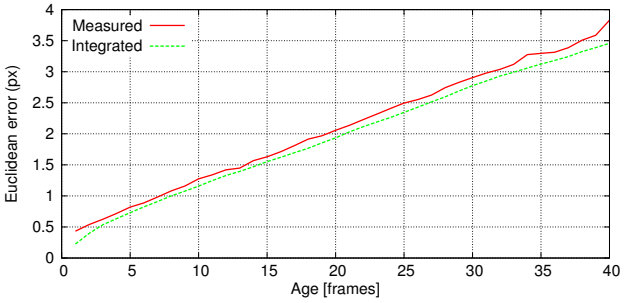


Figure 3. Total reprojection error as a function of the age for features with a survival age of exactly 40 frames.

Probability density function of the error. Figure 1 shows the inter-frame error probability density function for each component of the tracked features for all features tracked in the data set. From these plots, we see not only that all probability density functions have a Laplacian-like distribution (blue fitted curve), but also, and more importantly, that they are zero-mean. This is an important property, since the sample mean of tracked positions provides us with a simple unbiased estimator of the feature position. In order to compute the sample mean, all past tracked positions of a feature must be transformed to the current frame. Section 4.2 addresses this problem.

Tracking error dependency. In order to analyze the accumulated tracking error over several frames, we have computed the distribution of the accumulated Euclidean projection error as a function of the survival age of the feature. As in the previous experiment, in all cases the error has a zero-mean, Laplacian-like distribution, with a standard deviation as shown in the red curve of Figure 2 (the green curve in the figure will be described in the next section). Figure 2 shows that features that die younger, on average, have a larger error that those that can be tracked over several frames. The

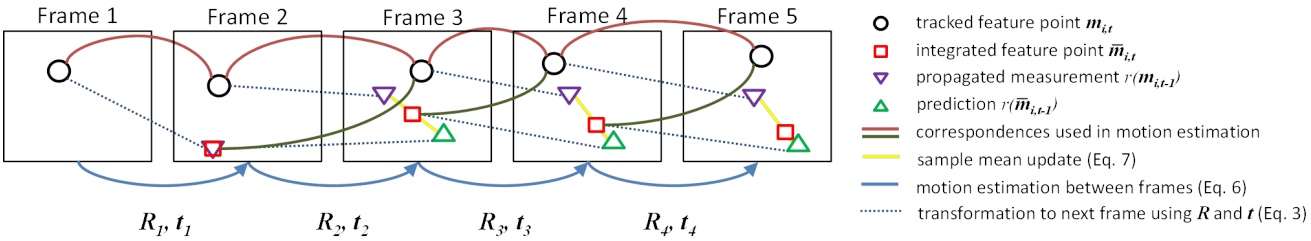


Figure 4. Example of multi-frame feature integration and motion estimation. See end of Section 4.2 for details.

reason for this behavior is that some features are more *trackable* than others, i.e., features that can be tracked longer possess some (unknown) property that makes them easier and more accurate to track. However, within the set of features that survive a specific number of frames, the accumulated reprojection error for a feature tracked n frames is equivalent to n times the inter-frame RMSE, as shown in the red curve of Figure 3 that corresponds to features with a survival age of 40 frames. We have verified the same linear relationship with all features with survival ages smaller than 40 frames. Therefore, error at each tracking step is constant and independent of the previous step, i.e., the inter-frame tracking error is homoscedastic.

These two analyses show that the inter-frame feature tracking error of the features is zero-mean, independent of the age, and identically distributed (the same is true for KLT and SURF features, as we could verify performing the same experiments addressed in this section). From these observations, we conclude that the unweighted sample mean is an unbiased optimal estimator of the feature position. In the next section, we use this important property of the features to improve the motion estimation accuracy.

4.2. Augmented Feature Set

To reduce the drift caused by the tracking error, we introduce an augmented feature set $\bar{\boldsymbol{m}}_{i,t}$ for $i = 1, 2, ..., n$. The new feature $\bar{\boldsymbol{m}}_{i,t}$, which we call an *integrated feature*, is obtained by the sample mean of all previous measured positions of feature $\boldsymbol{m}_i$ transformed into the current frame. With this new augmented set, we establish a new set of feature correspondences to be included in the camera motion optimization. The new objective function finds the optimal rotation and translation by minimizing two sets of feature correspondences:

1. **Measured-to-measured**. The same correspondence $\boldsymbol{m}_{i,t-1} \leftrightarrow \boldsymbol{m}_{i,t}$ defined in Section 3.1.

2. **Integrated-to-measured**. The new correspondence $\bar{\boldsymbol{m}}_{i,t-1} \leftrightarrow \boldsymbol{m}_{i,t}$.

Equation 5 is expanded to include the additional set of correspondences:

$$\alpha \sum_{i=1}^{n} e(\boldsymbol{m}_{i,t-1}, \boldsymbol{m}_{i,t})^2 + \beta \sum_{i=1}^{n} a_i e(\bar{\boldsymbol{m}}_{i,t-1}, \boldsymbol{m}_{i,t})^2 \quad (6)$$

where α and β are weighting factors such that $\alpha + \beta = 1$, and a_i is the age of the feature at time $t-1$. At each frame, after the rigid body motion between current and previous frame has been estimated, we update the integrated feature incorporating the new propagated measurement $\boldsymbol{m}_{i,t-1}$ into the sample mean:

$$\bar{\boldsymbol{m}}_{i,t} = \frac{\boldsymbol{r}(\boldsymbol{m}_{i,t-1}) + a_i\, \boldsymbol{r}(\bar{\boldsymbol{m}}_{i,t-1})}{1 + a_i} \quad (7)$$

Observe that when the feature is first detected, $a_i = 0$ and $\bar{\boldsymbol{m}}_{i,t} = \boldsymbol{r}(\boldsymbol{m}_{i,t-1})$.

The first term in Equation 6 is the same as in Equation 5 and contributes to estimate the motion observed between current and previous frames. The weighting factor w_i, implicit in $e()$ and defined in Equation 4, is obtained from the age of the feature as the inverse (i.e., $1/x$) RMSE of Figure 2. The second term corresponds to the augmented feature set and contributes to reduce the drift produced over multiple frames. The normalized weighting factors α and β can be used to change the balance of the contributions of each term. We have performed experiments using synthetic and real data and found that uniform weighting of both terms (i.e., $\alpha = \beta = 0.5$) provides the best results.

The reduction on the feature tracking error for integrated features can be seen in green curves of Figures 2 and 3.

Figure 4 depicts the proposed multi-frame procedure. At Frame 1, a feature (circle) is first detected and then tracked by the optical flow algorithm in Frame 2 (circle). The camera motion between Frames 1 and 2 is obtained using all available correspondences between both frames (other features are not shown for clarity purposes). At this point, the feature in Frame 1 is propagated to Frame 2 (inverted triangle) using the optimized rotation matrix and translation vector. Since the feature is new, the integrated feature (square in Frame 2) is equivalent to this first observation. The real

measured feature (circle) in Frame 2 is then tracked to the next frame. Now, two correspondences are established between Frames 2 and 3 (red and green lines), both of which are used by the optimization algorithm to obtain the camera motion (i.e., $\boldsymbol{R}_2$ and $\boldsymbol{t}_2$). The same propagation process is now applied to transform both, measured (circle) and integrated (square) features from Frame 2 to Frame 3 (obtaining the two triangles). The sample mean is updated by Equation 7, giving the new integrated feature (the square on Frame 3). The same optimization and propagation process is then repeated until the feature cannot be tracked any more.

4.3. Computational Complexity

The computational complexity of our proposed algorithm is given by the additional time required by the least squares method to find a solution with the augmented feature set, plus a small overhead for calculating the integrated features (i.e., Eq. 7). Therefore, the computational complexity of this multi-frame integration approach is simply $O(n)$ for n features.

4.4. Outlier Rejection and Measurement Correction

In addition to the multi-frame feature integration, we apply two techniques to improve tracking accuracy. The first technique evaluates the innovation of the measurements, providing a measure of the accuracy of the tracked feature. The second technique applies an n-sigma threshold to the tracked feature position to check for tracking errors and correct them.

Tracking accuracy via innovation analysis. Equation 7 updates the mean of the integrated feature incorporating the propagated observation from the previous frame. We define *innovation* as the distance between the prediction $\boldsymbol{r}(\bar{\boldsymbol{m}}_{i,t-1})$ and the observation $\boldsymbol{r}(\boldsymbol{m}_{i,t-1})$ that will be incorporated into it (i.e., the length of the yellow line in Fig. 4). A small innovation value is a good indicator of the consistency of the new observation with the estimation. We keep the mean of the innovations at each step as a measure of the tracking reliability. If the mean of the innovations becomes larger than a predefined threshold, the feature is marked as lost and considered a new detection.

Correction of inaccurate measurements. Tracking mismatches and inaccuracies are common, and their early detection is important to avoid their propagation to future frames, and consequently, to the position estimates. After the integrated feature has been updated by Equation 7, we calculate the distance between the integrated feature and the tracked feature at the current time t (i.e., distance between square and circle within the same frame in Fig. 4). If the distance is larger than a predefined threshold, the tracked feature (i.e., the circle) is discarded and replaced with the integrated feature. A new descriptor must be computed at this point, since the feature has changed position. This adds a small additional computational overhead to the algorithm, as we will show later in the experimental results. If the correction occurs more than a certain number of consecutive times (*e.g.*, 3 frames in a row), the feature is considered unstable, and the feature is marked as lost.

Both thresholds mentioned above were found by parameter optimization using the training data sets of the KITTI Vision Benchmark suite [8].

5. Experimental Results

In this section, we evaluate our proposed approach using two data sets with ground truth and compare it with the traditional method to obtain an estimate of the improvements.

5.1. Experimental Setup

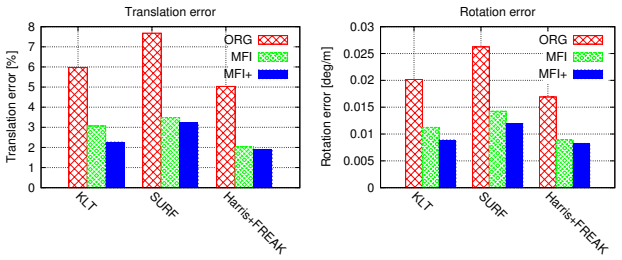
Implementation. We have implemented the proposed method in C++ using OpenMP technology, the Intel Performance Primitives library, and the OpenCV library. For feature tracking, we compare results using KLT [25] and two state-of-the-art descriptors: SURF [4] and FREAK [1]. Since FREAK is just a feature descriptor extractor, we use Harris [10] as keypoint detector. The keypoints are matched between consecutive frames by brute-force combinatorial search. Stereo disparity is measured by local correlation using a rectangular window centered on the feature position. Up to 4,096 features are tracked between consecutive frames.

Methods. In the following sections we compare the following three methods:

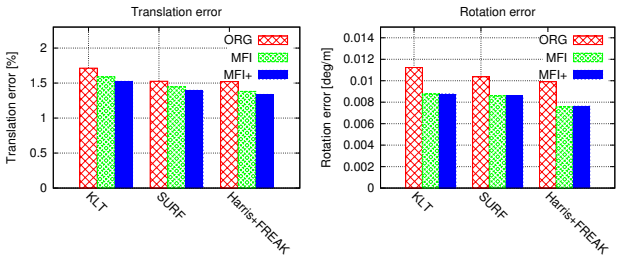
- **ORG.** The algorithm as presented in Section 3 [3].
- **MFI.** The new proposed multi-frame integration algorithm using the augmented feature list as described in Section 4.2.
- **MFI+.** The multi-frame integration algorithm including the techniques addressed in Section 4.4.

Data sets. We evaluate our proposed methods with two datasets. The first one is the synthetic New Tsukuba Stereo Dataset [18]. In this dataset, a stereo camera is flown around an office environment. The data set is provided with four illumination settings, from which we chose the "fluorescent" version. The second data set is the KITTI Vision Benchmark Suite [8]. These data were captured by driving different traffic scenarios in and around the city of Karlsruhe, Germany. The performance of our proposed approach is evaluated on both data sets in the following two sections.

Evaluation criteria. We use the evaluation method recommended by KITTI dataset: we compute translational

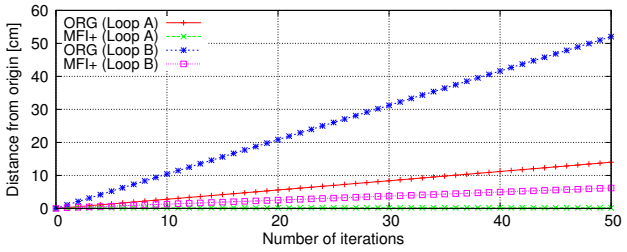


(a) New Tsukuba

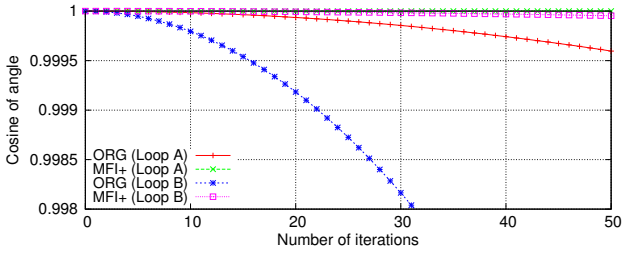


(b) KITTI

Figure 5. Average errors for New Tsukuba and KITTI (00 - 10) data sets



(a) Distance from the origin



(b) Cosine from the original angle

Figure 6. Results on loop sequences

.

and rotational errors for all possible subsequences of length (5,10,50,100,150,...,400) meters and take the average of them. Since the traveled distance in the New Tsukuba data set is smaller, we compute the error using the same path sub-lengths but using centimeters as the unit measure.

5.2. New Tsukuba Stereo Dataset

Figure 5(a) shows the translation and rotation error of the three evaluated methods on the New Tsukuba Stereo dataset. Table 1 shows the corresponding improvements over the original method. Our new multi-frame feature integration method reduces the translation error by up to 63% and the rotation error by up to 53% with respect to the traditional approach. The techniques introduced in Section 4.4 further extend the improvements by an additional 2%. Harris+FREAK is consistently the best tracker in this data set.

To further demonstrate the contribution of our multi-frame method on the drift reduction, we prepared two special data sets that plays forward a part of the New Tsukuba Stereo dataset, and then plays it backward to the first frame. The idea is to evaluate the drift of the camera at the end of each loop. If there is no drift at all, the camera position and angle should revert to zero.

We prepared two types of loop sequences for this purpose. In the first loop sequence there is a large portion of the scene that remains within the field of view of the camera so features can be tracked continuously on this region. We use frames 300 to 323 of New Tsukuba dataset for this loop sequence. The data set contains 50 full loops (i.e., 2,400 frames), and we check the position and angle error at the end of every loop (i.e., every 48 frames). We call this data set "Loop A".

In the second loop sequence, which we call "Loop B", the field of view completely changes between the first and last images (before playing backwards), so no feature can be tracked continuously throughout a loop. We use the frames 850 to 885 of New Tsukuba data set. The complete data set contains 50 loops (i.e., 3,600 frames), and we evaluate the position and angle error at the end of every loop (i.e., every 72 frames). We use Harris+FREAK as the feature tracking method.

Figures 6(a) and 6(b) show the results on the loop sequences, where it can be seen that a significant improvement is obtained by the multi-frame integration. The results show that less drift occurs on data set Loop A, where features can be tracked over long periods of time. When the feature survival is limited, as in Loop B, the improvements are still substantial.

Figure 7 shows the reconstructed paths using Harris+FREAK, in which one can see the improved alignment of our new proposed method over the baseline algorithm.

Table 2 shows the processing time comparing the ORG and MFI+ methods. For this experiment we used an Intel Core i7 2.70 GHz with 4 Cores. The additional processing time required by the new algorithm is just 1.9 ms (3.2% of the total time), of which 1.5 ms correspond to the time is

Table 1. Improvements on the New Tsukuba data set

		KLT	SURF	Harris+FREAK
MFI	Translation	62.8%	54.7%	59.6%
	Rotation	53.0%	45.6%	47.1%
MFI+	Translation	65.0%	58.0%	62.2%
	Rotation	55.5%	54.6%	51.2%

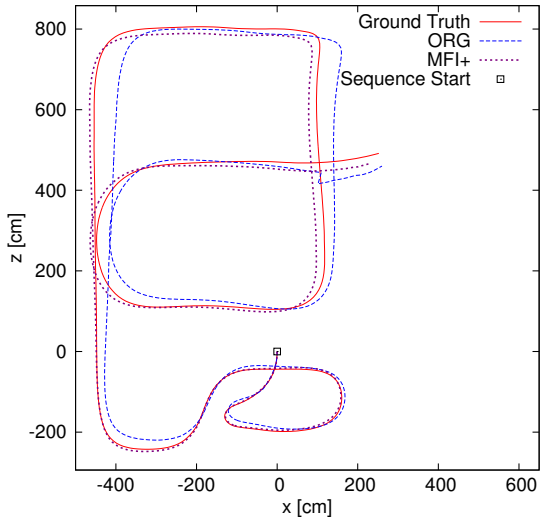


Figure 7. Reconstructed path of New Tsukuba

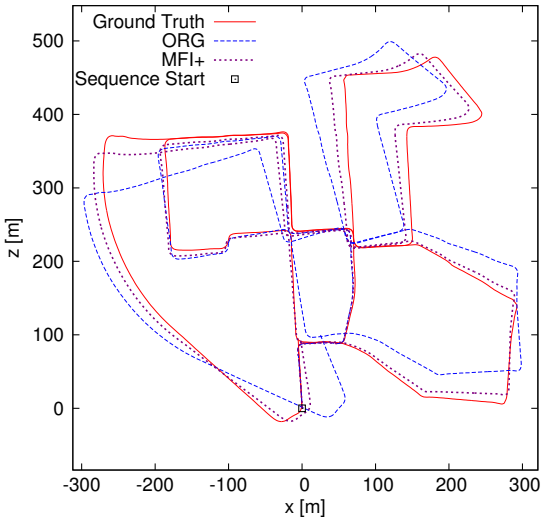


Figure 8. Reconstructed path of KITTI 00

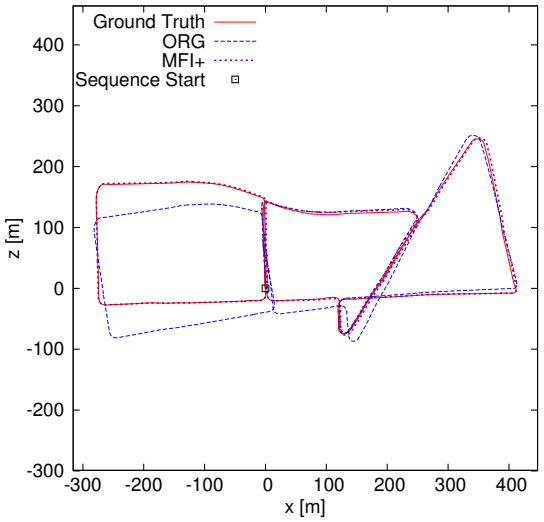


Figure 9. Reconstructed path of KITTI 13

needed to compute new descriptors when corrections occur, as mentioned in Section 4.4.

5.3. The KITTI Vision Benchmark Suite

Figure 5(b) shows the average translation and rotation error for the original and new proposed methods using the 11 training data sets provided by the the KITTI Vision Benchmark Suite. Table 3 shows the relative improvements with respect to the original algorithm.

Our proposed algorithm is up to 12% better in translation and 24% better in rotation than the original algorithm. As with the New Tsukuba data set, the best results were obtained with the Harris+FREAK tracker. On the other hand, SURF performs better than KLT in this data set. This is because of the large illumination changes in the KITTI data sets that the standard KLT can not handle.

The improvements shown in Table 3 are smaller than in the New Tsukuba data set. We think that there are two main reasons for it. First, features can be tracked longer in the New Tsukuba data set, providing more improvement on the motion estimate. Second, all parameters of the algorithm were tuned using the KITTI training data sets, which might not necessarily be optimal for New Tsukuba, leaving less space for improvement on an already optimal configuration.

Table 2. Processing Time (ms)

	Tracking	Stereo	VO	Other	Total
Original	35.6	3.8	8.4	1.4	49.2
MFI	37.1	3.8	8.8	1.4	51.1

Table 3. Improvements on the KITTI data set

		KLT	SURF	Harris+FREAK
MFI	Translation	5.8%	5.2%	9.2%
	Rotation	20.3%	17.1%	23.7%
MFI+	Translation	9.4%	8.5%	12.1%
	Rotation	20.3%	17.1%	23.1%

Regardless of all this, our proposed approach outperforms all other visual odometry methods known up to the submission date of this paper. The translational error on the evaluation data set is 1.62% of the traveled distance and 0.0062 deg/m rotational error[2].

Figures 8 and 9 show the path reconstructions for the KITTI Vision Benchmark data set Nrs. 0 and 13, from which the drift reduction of the MFI+ method is clearly visible.

6. Summary

In this paper, we have presented a new multi-frame technique that integrates the whole history of the tracked fea-

[2]While this paper was under review, the evaluation criteria on the KITTI Benchmarking Suite changed. The current errors for MFI+ are 1.30% for translation and 0.0028 deg/m for rotation. Under the new error criteria, our proposed method still outperforms all other VO methods.

ture points to reduce ego-motion drift while maintaining a high inter-frame motion accuracy. Our proposed approach is shown to be the best performing algorithm using the challenging KITTI Vision Benchmark data sets. The multi-frame technique relies on two very important properties of the feature tracking noise: the error is zero-mean and homoscedastic. We have verified these properties for a variety of tracking methods including SURF, FREAK and KLT. Based on these findings, we have defined an unbiased optimal estimator of the real feature position and created an augmented feature set with those estimates. The augmented set establishes a new set of correspondences in the least squares objective function. Our proposed algorithm is computationally inexpensive and can be easily adapted into most VO approaches relying on feature tracking.

References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *International Conference on Computer Vision and Pattern Recognition*, pages 510–517, 2012.

[2] P. Alcantarilla, J. Yebes, J. Almazn, and L. Bergasa. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *International Conference on Robotics and Automation*, pages 1290–1297, May 2012.

[3] H. Badino and T. Kanade. A head-wearable short-baseline stereo system for the simultaneous estimation of structure and motion. In *IAPR Conference on Machine Vision Application*, pages 185–189, June 2011.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[5] C. Beall, B. Lawrence, V. Ila, and F. Dellaert. 3d reconstruction of underwater structures. In *International Conference on Intelligent Robots and Systems*, October 2010.

[6] A. I. Comport, E. Malis, and P. Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *International Conference on Robotics and Automation*, April 2007.

[7] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision*, pages 1403–1410, October 2003.

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI Vision Benchmark Suite. In *International Conference on Computer Vision and Pattern Recognition*, 2012.

[9] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium*, pages 963–968, June 2011.

[10] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, 1988.

[11] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[12] E. S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *International Journal of Robotics Research*, 30(4):407–430, April 2011.

[13] M. Kaess, K. Ni, and F. Dellaert. Flow separation for fast and robust stereo odometry. In *International Conference on Robotics and Automation*, pages 973–978, May 2009.

[14] A. Kelly. Linearized error propagation in odometry. *The International Journal of Robotics Research*, 23(2):179 – 218, February 2004.

[15] B. Kitt, A. Geiger, and H. Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *Intelligent Vehicles Symposium*, June 2010.

[16] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality*, pages 1–10, November 2007.

[17] K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In *International Symposium on Research in Robotics*, pages 201–212, November 2007.

[18] M. P. Martorell, A. Maki, S. Martull, Y. Ohkawa, and K. Fukui. Towards a simulation driven stereo vision system. In *International Conference on Pattern Recognition*, pages 1038–1042, November 2012.

[19] L. H. Matthies. *Dynamic stereo vision*. PhD thesis, Carnegie Mellon University Computer Science Department, 1989.

[20] H. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. In *tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University*, September 1980.

[21] D. Nistér, O. Naroditsky, and J. R.Bergen. Visual odometry. In *International Conference on Computer Vision and Pattern Recognition*, pages 652–659, June 2004.

[22] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229, June 2003.

[23] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, July 2008.

[24] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *Robotics & Automation Magazine*, 18(4):80–92, 2011.

[25] J. Shi and C. Tomasi Good Features to Track. In *International Conference on Pattern Recognition*, pages 539-600, June 1994.

[26] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz. A new approach to vision-aided inertial navigation. In *Int. Conference on Intelligent Robots and Systems*, 2010.

[27] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Vision Algorithms: Theory and Practice*, September 1999.

[28] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, October 1995.