When to Use What Feature? SIFT, SURF, ORB, or A-KAZE Features for Monocular Visual Odometry

Hsiang-Jen Chien[1], Chen-Chi Chuang[2], Chia-Yen Chen[2], and Reinhard Klette[1]

[1] School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology, Auckland, New Zealand
Email: jchien@aut.ac.nz
[2] Dept. Computer Science and Information Engineering
National University of Kaohsiung, Taiwan

***Abstract*—Image feature-based ego-motion estimation has been dominating the development of visual odometry (VO), visual simultaneously localisation and mapping (V-SLAM), and structure-from-motion (SfM) for several years. The detection, extraction, or representation of image features play crucial roles when solving camera pose estimation problems, in terms of accuracy and computational cost. In this paper we review three popular classes of image features, namely SIFT, SURF, and ORB, as well as the recently proposed A-KAZE features. These image features are evaluated using the KITTI benchmark dataset to conclude about reasons for deciding about the selection of a particular feature when implementing monocular visual odometry.**

I. Introduction

Visual odometry (VO) has been extensively studied since the early 1980's. Extensive work for its development has led to a separation into either appearance-based or feature-based techniques [1]. Appearance-based methods make direct use of pixel intensities to establish dense inter-frame correspondences, leading to a higher demand of computation power (e.g. [2]–[4]). Feature-based methods deploy image abstraction strategies which identify sparse but distinguishing key points from image intensities and extract their vector representations (e.g. [5]–[8]); in the feature space, a matching process is performed to establish a set of inter-frame point correspondences which are used in turn to solve the ego-motion estimation problem.

An image feature identification algorithm comes with two stages - keypoint detection first and then descriptor extraction. In the keypoint-detection stage, local operators are applied to the image at a single scale, or in multiple scales. These operators are crafted to give strong responses once applied to patches containing rich structural information. Centres of regions with high responses are located as keypoints. To further improve the uniqueness of the image feature, a non-maximum suppression strategy is optionally carried out to eliminate redundant keypoints. Vector representations encoding local characteristics (the *descriptors*) of the detected keypoints are then extracted in the second stage.

978-1-5090-2748-4/16/$31.00

One might ask "what type of image feature is the best?" when first time stepping into the ego-motion estimation problem [18]. SIFT [9] and SURF [10] features are two popular choices when solving the pose-estimation problem. A more recent alternative, ORB features [11], are also becoming widely adopted, especially by embedded robotics systems and for real-time applications (e.g. [7]), due to greatly reduced computational requirements. Two even more recent options, KAZE [16] and A-KAZE [17], have also been found to provide comparative performance with better computational efficiency.

SIFT, SURF, and ORB (and more) had been evaluated with respect to generic invariance properties in [15]. In this paper we review and evaluate SIFT, SURF, ORB, and A-KAZE image features with respect to process outcomes within a monocular visual-odometry framework, for answering the question stated above.

The rest of this paper is organised as follows. In Section II we describe the monocular ego-motion estimation problem and our implementation. In Section III we briefly recall the used four types of image features, which are then tested in Section IV. We conclude this paper in Section V.

II. Monocular Visual Odometry

In this section we present the fundamentals and an implementation of a monocular visual odometry framework which is used to evaluate SIFT, SURF, ORB, and A-KAZE features. The stages of the framework are depicted in Fig. 1.

A. Motion Recovery

Let $\mathbf{X} = (x, y, z)^\top$ be a 3D point and $\mathbf{x} = (u, v)^\top$ its projection in the image plane. The central projection model is as follows:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_u & 0 & u_c & 0 \\ 0 & f_v & v_c & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (1)$$

where the upper 3×3 triangular matrix $\mathbf{K}$ is the *camera matrix* modelled by the intrinsic parameters of the camera including focal lengths f_u and f_v, and the image centre or principle point (u_c, v_c), and $\sim$ denotes equality up to an unknown scale.

Assume that $\mathbf{X} = (\mathbf{x}, \mathbf{y}, \mathbf{z})^\top$ is a stationary point which remains at the same position as the camera moves to the next frame. Its new projection in the image plane becomes

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \sim \mathbf{K} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{2}$$

where the rotation matrix $\mathbf{R} \in SO(3)$ and the translation vector $\mathbf{t} \in \mathbb{R}^3$ together present the Euclidean transformation introduced by the motion of the camera.

In the context of the motion estimation problem, the transformation $(\mathbf{R} \quad \mathbf{t})$ is the unknown to be solved, given a number of 3D-to-2D correspondences $(x, y, z) \leftrightarrow (u', v')$. This is known as the *perspective-from-n-points* (PnP) problem. Treating the whole 3-by-4 projection matrix $\mathbf{P} = \mathbf{K}\begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix}$ as a black box, the solution can be easily found by solving a homogeneous linear system. Such a strategy is known to be the *direct linear transform* (DLT) method [19]. The rotation matrix, solved in this way, however, might not be a valid element in $SO(3)$ due to over-parametrization. In this work we therefore deploy the *efficient-PnP* (EPnP) algorithm [20] for finding the solution of Eq. (2).

B. Nonlinear Optimisation

Camera motion $(\mathbf{R} \quad \mathbf{t})$, estimated in a linear manner (e.g. DLT, or EPnP), yields a sub-optimal solution only. To achieve the *maximum-likelihood estimation* (MLE), a non-linear adjustment process is required [21].

Assuming that the given measurement noise follows a Gaussian model, the MLE based on N tracked features is achieved by a minimisation of the sum-of-squares of the reprojection error:

$$\phi_{\text{RPE}}(\mathbf{R}, \mathbf{t}) = \sum_{1 \le i \le N} \|\mathbf{x}'_i - \pi_{\mathbf{K}}(\mathbf{R} \cdot \mathbf{X}_i + \mathbf{t})\|^2_{\mathbf{\Sigma}_i} \tag{3}$$

where $\pi_{\mathbf{K}} : \mathbb{R}^3 \to \mathbb{R}^2$ is the projection function that maps a 3D point into the projective space P^2 using the camera matrix $\mathbf{K}$ and converts the resulting homogeneous coordinates into a Cartesian plane. By $\|\cdot\|_\Sigma$ we denote the Mahalanobis distance defined by covariance matrix $\mathbf{\Sigma}$.

To further improve the stability of the adjustment process, we consider also the 2D-to-2D correspondences $(u, v) \leftrightarrow (u', v')$ and introduce an epipolar objective function:

$$\phi_{\text{EPI}}(\mathbf{R}, \mathbf{t}) = \sum_{1 \le i \le N} \delta(\mathbf{x}_i, \mathbf{x}'_i; \mathbf{F}) \tag{4}$$

Here, δ is the Sampson distance [22] between $\mathbf{x}$ and $\mathbf{x}'$ defined by

$$\delta(\mathbf{x}, \mathbf{x}'; \mathbf{F}) = \frac{(\mathbf{x}'^\top \mathbf{F}\mathbf{x})^2}{(\mathbf{F}\mathbf{x})_0^2 + (\mathbf{F}\mathbf{x})_1^2 + (\mathbf{F}^\top\mathbf{x}')_0^2 + (\mathbf{F}^\top\mathbf{x}')_1^2} \tag{5}$$

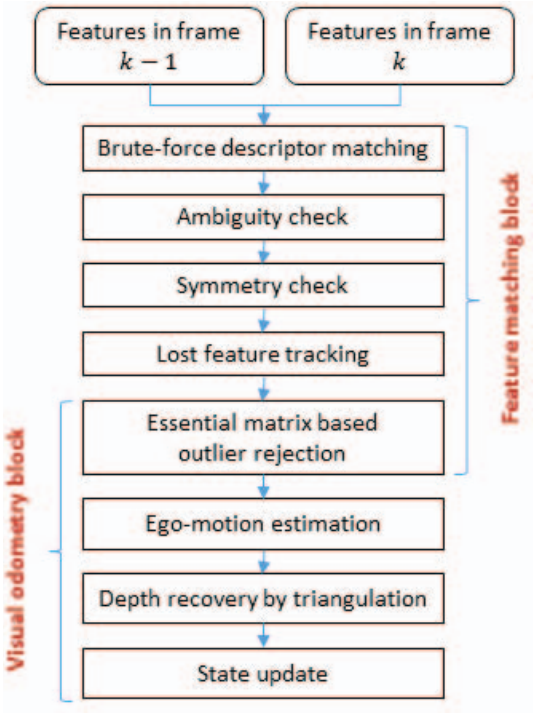

Fig. 1. Stages of feature-based monocular visual odometry as followed in this paper.

and $\mathbf{F}$ is the fundamental matrix $\mathbf{F} = \mathbf{K}^{-\top}[\mathbf{t}]_\times \mathbf{R}\mathbf{K}^{-1}$ where

$$[\mathbf{t}]_\times = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \tag{6}$$

denotes the skew-symmetric form of $\mathbf{t} = (t_x, t_y, t_z)^\top$. Finally, $(\mathbf{F}\mathbf{x})_\mathbf{i}$ denotes the i-th entry of $\mathbf{F}\mathbf{x}$.

We combine the objectives given by Eqs. (3) and (4) into one regularised energy function:

$$\mathbf{\Phi}(\mathbf{R}, \mathbf{t}) = (1 - \alpha) \cdot \phi_{\text{RPE}}(\mathbf{R}, \mathbf{t}) + \alpha \cdot \phi_{\text{EPI}}(\mathbf{R}, \mathbf{t}) \tag{7}$$

where a chosen damping parameter $\alpha = [0, 1]$ controls the weight of the epipolar constraint.

The value of α is normalised by the number of terms in ϕ_{RPE} and ϕ_{EPI}. Let $\beta = [0, 1]$ be the desired weight, N_{RPE} be the number of 3D-to-2D correspondences, and N_{EPI} the number of 2D-to-2D ones. The normalised damping parameter, applied to Eq. (7), is calculated by

$$\alpha = \left(1 + \frac{N_{\text{EPI}}}{N_{\text{RPE}}} \cdot \frac{1 - \beta}{\beta}\right)^{-1} \tag{8}$$

Equation (7) cannot be solved in any closed form. Thus, one may adopt a non-linear least-square minimiser, say the Levenberg-Marquardt algorithm [23], to minimise the energy function, starting with the linear solution.

C. Feature Matching

Matching of image features has been widely deployed to establish 2D-to-2D pixel correspondences required to populate

terms in Eq. (4). Once the ego-motion of the current frame is solved, these correspondences are used again to obtain 3D-to-2D correspondences as required by Eq. (3), by means of the approach described in the next section.

The features are initially matched in feature space. Let F and F' be the sets of image features identified in two consecutive frames. A feature $\chi \in F$ is mapped to $\chi' \in F'$ where

$$\chi' = \underset{\chi^\star \in F'}{\operatorname{argmin}}\, d(\chi, \chi^\star) \tag{9}$$

and $d : F \times F' \to \mathbb{R}$ is a selected metric to measure the similarity between two features.

In the case of binary feature descriptors, the *Hamming distance* is used; otherwise one might choose the sum of squared distances (SSD) or the sum of absolute distances (SAD) as metric. In our work, a brute-force search is performed over F' to find the mapping $\chi \to \chi'$.

Ambiguous matches are identified by a difference ratio check. Let $\breve{\chi}' \in F'$ be the second best match of χ. The mapping $\chi \to \chi'$ is said to be ambiguous if

$$\frac{d(\chi, \chi')}{d(\chi, \breve{\chi}')} < r \tag{10}$$

where $r = (0, 1]$ is a pre-defined threshold. In [24] it is suggested to set $r = 0.6$. We also remove inconsistent matches by performing a process of backward matching, from F' to F. In particular, a match $\chi \to \chi'$ is considered to be inconsistent if it is found that

$$\underset{\breve{\chi} \in F}{\operatorname{argmin}}\, d(\breve{\chi}, \chi') \neq \chi \tag{11}$$

After enforcing Eqs. (10) and (11), we have a set of unambiguously symmetric matches $\chi \leftrightarrow \chi'$.

The feature matches are then augmented by performing the KLT tracking algorithm [25] on each feature in F that failed to find a match in F'. On the augmented feature set we conduct an outlier rejection process using the *random consensus sampling* (RANSAC) [26] technique. In each iteration, five matches are selected to estimate an essential matrix $\mathbf{E}$ using the five-point algorithm of [27]. The matrix is then used to evaluate the Sampson distances of all the matches. The process is repeated until significantly-many inliers are found to agree with an essential matrix within a tolerable distance, say, of 0.5 pixels.

D. Depth Recovery

To estimate the motion of a camera with a consistent scale, the Euclidean coordinates have to be used. Considering a monocular setup, the absolute scale cannot be projectively determined. The motion from the first to the second frame is usually solved by means of an essential-matrix decomposition, and in turn used to recover features' 3D coordinates. The visual odometry process, bootstrapped in this way, is able to provide ego-motion estimation in a consistent scale through the sequence.

Let (u, v) and (u', v') be the image coordinates of a matched feature $\chi \leftrightarrow \chi'$ in two consecutive frames. Given the camera's motion $(\mathbf{R}\,\mathbf{t})$, the 3D coordinates of χ can be recovered by minimising the following distance with respect to free parameters $k, k' \in \mathbb{R}^+$:

$$\delta_{\text{mid}}(k, k') = \|k\mathbf{a} - (k'\mathbf{a}' + \mathbf{c}')\|^2 \tag{12}$$

where $\mathbf{a} = \mathbf{K}^{-1}(u, v, 1)^\top$ and $\mathbf{a}' = \mathbf{R}^\top\mathbf{K}^{-1}(u', v', 1)^\top$ are the directional vectors of the back-projected rays, and $\mathbf{c}' = -\mathbf{R}^\top\mathbf{t}$ is the new camera centre.

The minimum of Eq. (12) can be found by calculating the least-square solution of the following linear system:

$$\begin{bmatrix}\mathbf{a} & -\mathbf{a}'\end{bmatrix}\begin{bmatrix}k \\ k'\end{bmatrix} = \mathbf{A}\begin{bmatrix}k \\ k'\end{bmatrix} = \mathbf{c}' \tag{13}$$

The resulting values k and k' denote two points on each of the back-projected rays at the shortest mutual distance in 3D space. The midpoint of them equals

$$(x, y, z)^\top = \frac{1}{2}\left(\begin{bmatrix}\mathbf{a} & \mathbf{a}'\end{bmatrix}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top + \mathbf{I}\right)\mathbf{c}' \tag{14}$$

This yields an estimation of the feature's 3D coordinates. This approach is known as *mid-point triangulation*.

E. State Update

To improve the stability of the visual odometry process, we maintain states of the tracked features throughout the sequence. In this work we adopt an effective integration technique which maintains a weighted running average of the state, for each tracked feature. Let $\mathbf{m}_{i,k}$ be an observed state vector of feature i in frame k, and $\omega_{i,k} \in [0, 1]$ the weight denoting how likely the observation is believed to be the true state. The estimate of the true state is calculated as follows:

$$\bar{\mathbf{m}}_{i,k} = \frac{\bar{\omega}_{i,k-1} \cdot f_{k-1,k}(\bar{\mathbf{m}}_{i,k-1}) + \omega_{i,k} \cdot \mathbf{m}_{i,k}}{\bar{\omega}_{i,k}} \tag{15}$$

where

$$\bar{\omega}_{i,k} = \bar{\omega}_{i,k-1} + \omega_{i,k} \tag{16}$$

is the running weight, and $f_{k-1,k}$ is a transition function of a state, from the previous frame $k-1$ to the current frame k.

To temporally improve the depths of the tracked feature points, one may set $\mathbf{m}$ to be the 3D coordinates of a feature, and follow the state update. In this case, $f_{k-1,k}$ is the Euclidean transformation defined by the estimated ego-motion $(\mathbf{R}_k \quad \mathbf{t}_k)$, and the weight is set to be $\omega_{i,k} = \frac{1}{1+\delta_{i,k}}$, where $\delta_{i,k}$ is the estimated error of the triangulation. In the case of mid-point triangulation, we use the sum of the shortest distances from a triangulated point to the two corresponding back-projected rays.

III. Image Features

This section provides a brief literature review on the four image-features used in the following. A comparison is summarised in Table I.

TABLE I
COMPARISON OF IMAGE FEATURES

	SIFT	SURF	ORB	A-KAZE
Origin: Year and publication	1999 [9]	2006 [10]	2011 [11]	2013 [17]
Scale invariant	Yes	Yes	Yes	Yes
Rotation invariant	Yes	Yes	Yes	Yes
Keypoint measure	DoG	Hessian	FAST	Hessian
Descriptor type	Integer vector	Real vector	Binary string	Binary string
Descriptor length	128 bytes	256 bytes	256 bits	64/256/486 bits

A. SIFT

The SIFT (scale-invariant feature transform) algorithm, proposed by David Lowe in 1999 [9], is perhaps one of the earliest work on providing a comprehensive keypoint detection and descriptor extraction technique.

To formulate a scale invariant representation of image features, SIFT builds a multi-resolution pyramid over the input image and applies *difference of Gaussians* (DoG) operators to locate local extrema in the scale space. The locality is defined by a $3 \times 3 \times 3$ window. These non-edge extrema, presenting high local contrast, are then identified as the keypoints.

The image gradients of a 16×16 window, centred at each keypoint, are then computed and grouped into 4×4 subregions. The direction of gradients within the same subregion are then quantised to conclude an eight-bins histogram. By gathering all the bins from all sixteen histograms in the window, one obtains an 128-element SIFT descriptor vector.

B. SURF

Six years after the advent of SIFT, Herbert Bay et. al. proposed an alternative image feature detection and extraction algorithm known as SURF (speeded-up robust features), which was claimed to be faster and more robust than SIFT [10]. Instead of the DoG operators, the SURF algorithm chooses a box-filter-approximated second-order derivative computation to locate extrema in the scale space. These Haar-like operators can be efficiently implemented over a computed integral image. This leads to a speed-up of SURF compared to SIFT.

The robustness of SURF features comes from the identification of patch direction before extracting its feature vector. The direction of a keypoint is decided based on Gaussian-weighted responses of pixels in a circular neighbourhood with respect to horizontal and vertical Haar wavelets. These responses are transformed into polar coordinates, and partitioned based on a predefined angular resolution. The transformed response vectors in the same partition are then summed up, and the angle of the longest vector sum over all partitions is chosen as the direction of the patch. Based on the direction, the patch is rotated and a Gaussian-weighted discrete sampling on pixel responses to the Haar wavelets is performed to yield a real-vector descriptor.

C. ORB

In 2011, Ethan Rublee et. al. proposed the ORB (oriented FAST and rotated BRIEF) features as an alternative to SIFT and SURF [11]. As its self-explanatory name, the ORB algorithm combines an enhanced FAST (features from accelerated segment test) [12] technique to locate keypoints, with a direction-normalised BRIEF (binary robust independent elementary) [13] descriptor extraction process. To achieve scale invariance, the FAST algorithm is applied repeatedly to each layer of a scale pyramid. The cornerness of detected FAST features is tested using an Harris measurement [14]; non-corner keypoints are excluded.

The BRIEF algorithm produces an n-bit string from local binary tests of n predefined pixel pairs within a patch. This vector representation would be highly unstable against rotation. To address this issue, ORB adopts a rotation-aware variant of BRIEF. For the located keypoints, it first finds patch centroids by means of image moments [15]. The direction of a vector, connecting a keypoint's centre to its patch's centroid, then decides the direction of the keypoint. The binary test pattern is then rotated by the direction of a patch before binary descriptor extraction, allowing a feature to be represented in a rotation-invariant (i.e. isotropic) form.

D. A-KAZE

More recently, Pablo F. Alcantarilla el. al. developed an edge-preserving non-linear filtering strategy to locate image features [16]. The filtering is based on the image diffusion equation:

$$\frac{\partial I}{\partial t}(u, v, t) = \text{div}\left[c(u, v, t) \cdot \nabla I(u, v)\right] \tag{17}$$

where div and ∇ are, respectively, the divergence and the gradient operator; I is an intensity image, $t > 0$ is the scale variable, and c a conductivity function.

The diffusion of an image formulates an iterative way to find the filtered, yet edge-preserved evolution of the original image, by choosing the conductivity function carefully. A good choice of the conductivity is as follows:

$$c(u, v, t) = \left[1 + \left(\frac{\nabla I_\sigma(u, v, t)}{k}\right)^2\right]^{-1} \tag{18}$$

where I_σ is a Gaussian-smoothed version of image I. In [17], the *fast explicit diffusion* (FED) technique is used to solve the diffusion equation efficiently, contributing to an accelerated variation of the KAZE algorithm, proposed by the authors in 2012. The improved feature detection and extraction technique is known as A-KAZE.

Keypoints are located by finding the extrema of the second-order derivatives of the image over the non-linear multi-scale pyramid built from the principle of image diffusion. A-KAZE deploys a technique similar to SURF to estimate the direction of a patch. A modified *local difference binary* (LDB) representation of the rotation-compensated patch is then extracted as its binary descriptor.

IV. Experiments

We present results for a test sequence selected from the KITTI benchmark datasets [28] to evaluate the image features. The sequence presents a complex street scenario. Moving bicyclists, vehicles, and walking pedestrians are present in the scene. The test vehicle travelled 300 metres and captured 389 frames. We selected gray-level data of the left camera to perform monocular visual odometry. Data of the remaining three cameras are not used.

We use feature detection and extraction subroutines shipped with `OpenCV 3.1`. These subroutines are CPU-only implementations. The default parameters provided by the library are used, with only one exception: We increased the number of detectable features to $3,000$ for ORB, as its default setting to 500 features is insufficient to produce good results, compared to the other tested features. The obtained statistics of the feature extraction process is reported in Table II.

TABLE II
Comparison of Feature Extraction

	SIFT	SURF	ORB	A-KAZE
# Features	765,436	1,140,410	1,017,350	603,246
Time spent	98 sec	50 sec	10 sec	43 sec
Disk storage	391 MB	305 MB	54 MB	49 MB

For each tested feature, we repeat the visual odometry process 48 times on a 4-core Intel Core i7 3.2 GHz laptop. Motion drifts are then evaluated using ground truth available for these KITTI data (derived from GPS/IMU readings). To allow the ego-motion estimation to be evaluated in a consistent scale, the inertial data of the first and the second frame are used to bootstrap the monocular visual odometry process.

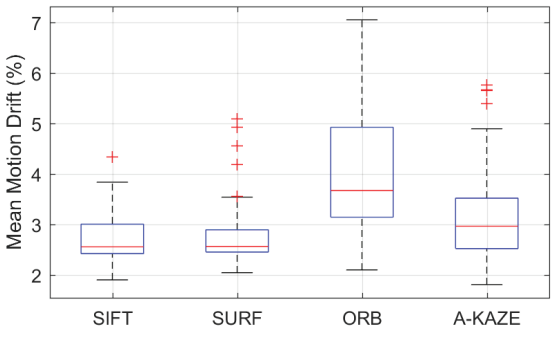

Fig. 2. Box plot of tested features. For each feature, the quartiles are concluded from 48 individual visual-odometry processes. The first and the third quartiles are, respectively, denoted by the bottoms and tops of a box, and the second quartile (median) is marked by the red line in the box.

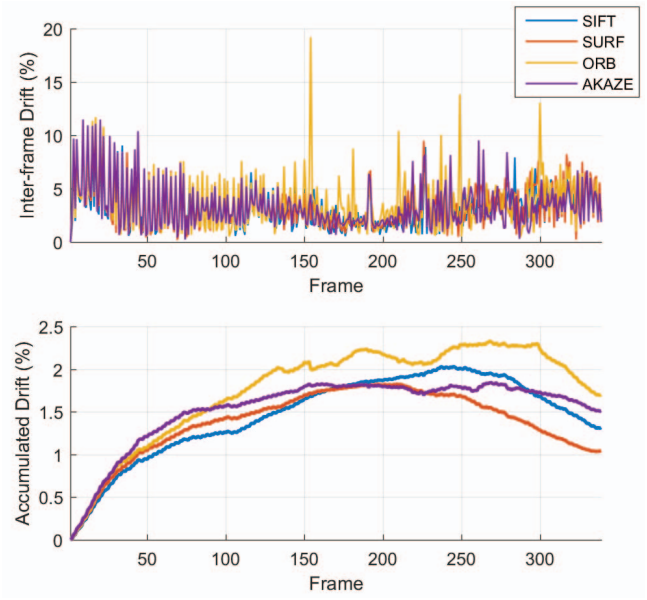

Fig. 3. Inter-frame (top) and accumulated (bottom) drift errors of the best cases of SIFT, SURF, ORB, and A-KAZE features.

The mean drift of each run is calculated individually; the results of all 192 runs are plotted together in Fig. 2. The plot shows that SIFT and SURF features achieve similar levels of accuracy, while the error distribution of the runs using SURF features is slightly less dispersed, compared to SIFT. The ORB features, on the other hand, show inconsistent performance with a wide range of drift errors. The A-KAZE features yield intermediate performance in-between the cases of SURF and ORB.

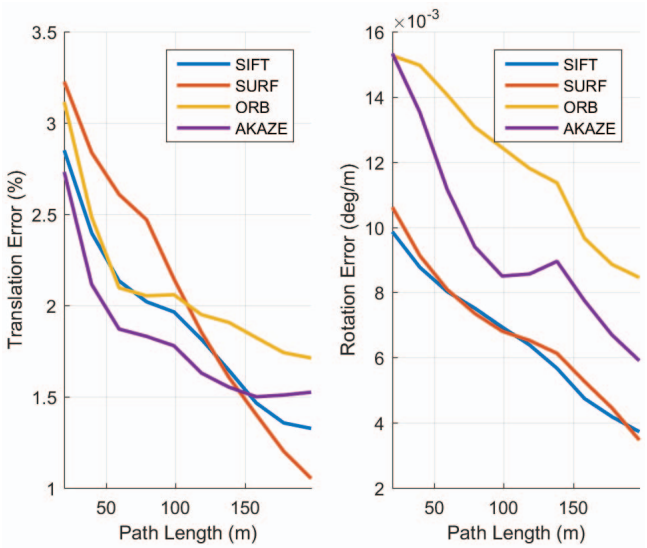

Fig. 4. Segmented motion error analysis on the translational (left) and rotational (right) components.

The best run of each trial set is selected to render a detailed view of how the different image features perform through the visual odometry process. The inter-frame and overall drift errors of these runs are plotted in Fig. 3. Initially, all the tested features shared a similar performance as the vehicle is slowly accelerating. The drift of the ego-motion, estimated from ORB features, however, grows faster than other cases, as the vehicle reaches a moderate speed. At the end of the sequence, the drift error follows the pattern "ORB $>$ A-KAZE $>$ SIFT $>$ SURF". It is also found that ORB-based visual odometry is less stable, as significant inter-frame drift errors occur more frequently.

The segmented motion errors, in terms of translation and rotation components of the estimated ego-motion, are also computed, with respect to various travel distances.

The travel distance is not measured only from the beginning of the sequence; applying the segmented error analysis of KITTI, we consider segments that begin from an arbitrary frame k through to frame $k+n$, where $n > 0$. The actual length of a segment is taken into account when calculating the error at image I being in a temporal interval $[l_p, l_q)$, with temporal order $l_p \leq l < l_q$.

We divide the length of the sequence into 10 equally spaced segments for plotting. The results are depicted in Fig. 4. It shows that, in the translation component, the A-KAZE features perform best for short movement within 150 metres, and the SURF features achieve the lowest overall error of 1% as the vehicle reaches the end of the sequence. In the rotational component, it presents a clear trend that the SIFT and SURF cases converge similarly to the lowest error, while the ORB features achieve two-times worse accuracy, and the A-KAZE features show an intermediate error.

V. Conclusions

In this paper we reviewed four popular image features, namely SIFT, SURF, ORB, and A-KAZE. These features are evaluated using a monocular visual-odometry framework, which is outlined in the paper. In our experimental results we found that the SURF-based visual odometry shows the best accuracy, and the SIFT-based implementation achieved similar results. The ORB features provide computationally the cheapest solution, however with a sacrifice of matching accuracy. The newly proposed A-KAZE features, on the other hand, demonstrated a trade-off between motion estimation accuracy and computation efficiency.

References

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I - The first 30 years and fundamentals." IEEE Robotics Automation Magazine, vol. 18, pp. 80–92, 2011.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry." In Proc. IEEE Int. Conf. Robotics Automation, pp. 15–22, 2014.

[3] M. J. Milford and G. Wyeth, "Single camera vision-only SLAM on a suburban road network." In Proc. IEEE Int. Conf. Robotics and Automation, pp. 3684–3689, 2008.

[4] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles." In IEEE Trans. Robot. (Special Issue on Visual SLAM), vol. 24, pp. 1015–1026, 2008.

[5] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers: Field reports." J. Field Robot, vol. 24, pp. 169–186, 2007.

[6] H., Badino, A. Yamamoto, and T. Kanade, "Visual odometry by multi-frame feature integration." Int. ICCV Workshop Computer Vision Autonomous Driving, 2013.

[7] S. Song, M. Chandraker and C. C. Guest, "Parallel, real-time monocular visual odometry." In Proc. IEEE Int. Conf. Robotics and Automation, pp. 4698–4705, 2013.

[8] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System." IEEE Trans. Robotics, vol. 31, pp. 1147–1163, 2015.

[9] D. G. Lowe, "Object recognition from local scale-invariant features." In Proc. ICCV, vol.2, pp.1150–1157, 1999.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features." In Proc. European Conf. Computer Vision, 2006.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF." In Proc. Int. Conf. Computer Vision, pp. 2564–2571, 2011.

[12] E. Rosten and T. Drummond, "Machine learning for high speed corner detection." In Proc. European Conf. Computer Vision, vol. 1, pp. 430–443, 2006.

[13] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features." In Proc. European Conf. Computer Vision, pp. 778–792, 2010.

[14] C. Harris and M. Stephens, "A combined edge and corner detector." In Proc. Alvey Vision Conference, pp.147–151, 1988.

[15] Klette, R.: "Concise Computer Vision." Springer, London, 2014

[16] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features." In Proc. European Conf. Computer Vision, 2012.

[17] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces." In Proc. British Machine Vision Conf., 2013.

[18] H. E. Benseddik, O. Djekoune, and M. Belhocine, "SIFT and SURF performance evaluation for mobile robot-monocular visual odometry." J. Image Graphics, vol. 2, pp. 70–76, 2014.

[19] R. I. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision", second edition. Cambridge University Press, Cambridge, 2004.

[20] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem." Int. J. Computer Vision, vol. 81, pp. 155–166, 2009.

[21] C. Engels, H. Stewenius, and D. Nister, "Bundle adjustment rules." In Proc. Photogrammetric Computer Vision, 2006.

[22] P. D. Sampson, "Fitting conic sections to 'very scattered' data: An iterative refinement of the Bookstein algorithm." Computer Graphics Image Processing, vol. 18, pp. 97–108, 1982.

[23] K. A. Levenberg, "Method for the solution of certain non-linear problems in least squares." The Quarterly Applied Math., vol. 2, pp. 164–168, 1944.

[24] D. Lowe, "Distinctive image features from scale-invariant keypoints." Int. J. Computer Vision, vol. 20, pp. 91–110, 2003.

[25] C. Tomasi and T. Kanade, "Detection and tracking of point features." Carnegie Mellon University Technical Report, CMU-CS-91-132, 1991.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography." Comm. ACM, vol. 24, pp. 381–395, 1981.

[27] D. Nister, "An efficient solution to the five point relative pose problem." IEEE Trans. Pattern Analysis Machine Intelligence, vol. 26, pp. 756–770, 2004.

[28] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, Vision meets robotics: The KITTI dataset. Int. J. Robotics Research, vol. 32, pp. 1231–1237, 2013.