ROS2D: Image Feature Detector Using Rank Order Statistics

Yousif, K.; Taguchi, Y.; Ramalingam, S.; Bab-Hadiashar, A.

TR2017-066 May 2017

Abstract
We present a new image feature detection method. Our method selects features based on segmenting points with high local intensity variations across different scales using a robust rank order statistics approach. Our method produces a large number of repeatable features that are invariant to several image transformations such as rotation, scaling, viewpoint, and lighting variations. We show the advantages of our feature in comparison to other existing features using the Oxford dataset. We also show that, when used in monocular and stereo SLAM systems, our feature outperforms SIFT in terms of the pose estimation accuracy using several public datasets including the KITTI dataset.

IEEE International Conference on Robotics and Automation (ICRA)

ROS2D: Image Feature Detector Using Rank Order Statistics

Khalid Yousif[1], Yuichi Taguchi[2], Srikumar Ramalingam[2], and Alireza Bab-Hadiashar[1]

***Abstract*— We present a new image feature detection method. Our method selects features based on segmenting points with high local intensity variations across different scales using a robust rank order statistics approach. Our method produces a large number of repeatable features that are invariant to several image transformations such as rotation, scaling, viewpoint, and lighting variations. We show the advantages of our feature in comparison to other existing features using the Oxford dataset. We also show that, when used in monocular and stereo SLAM systems, our feature outperforms SIFT in terms of the pose estimation accuracy using several public datasets including the KITTI dataset.**

I. Introduction

Extracting salient visual information from local image regions is regarded as one of the most important procedures for many image processing and computer vision applications. These applications include camera calibration, image matching and registration, object recognition and classification, structure from motion/SLAM and many more.

Generally, the main aim of feature extraction is to reduce the amount of resources required to describe an image by sampling it into a subset of points, while still describing the image with sufficient accuracy. For many years, SIFT [1] has been regarded as the golden standard for feature detection and description by the robotics and computer vision communities, due to its repeatability, distinctiveness, and invariance to a variety of image transformations. The number of features extracted by SIFT usually ranges between a few hundred to a few thousand. While this number may be sufficient, several applications benefit from extracting a larger number of features. For instance, feature-based SfM/SLAM systems estimate the camera pose and generate a 3D model using a subset of the matched features (inliers); if the number of the matched features is small, the number of the inliers will likely be small, resulting in inaccurate camera pose estimation and a sparse 3D model.

In this paper, we propose an image feature detection method that is able to extract a large number (ranging from a few thousand to tens and even hundreds of thousand) of highly repeatable features. When paired with robust image descriptors such as SIFT descriptors, the proposed feature is highly invariant to viewpoint, rotation, blurring, lighting, and scale changes. Similar to the 3D feature extraction method presented in [2], our method utilizes a rank order statistics based robust segmentation method (MSSE) to segment the image into regions with uniform intensities and regions containing high intensity variations. Experiments on the Oxford dataset [3] demonstrate that our feature performs favorably compared to existing features in terms of the repeatability and inlier ratio. We also use our feature in monocular and stereo SLAM systems and show that our feature outperforms SIFT in terms of the pose estimation accuracy using several public datasets including the KITTI dataset [4].

II. Related Work

Harris corner detector [5] is one of the earliest and most well-known feature detectors. They defined a corner by a point in which image intensities have a large variation between adjacent regions in all directions. Mikolajczyk and Schmid [6] extended the Harris corner detector to be scale invariant. Rosten and Drummond [7] proposed an efficient corner detector called FAST. FAST corners are found by comparing the neighboring pixels (in an area that includes 16 pixels around the center) to the center pixel. A region is defined as uniform, an edge, or a corner based on the percentage of neighboring pixels with similar intensities to the center pixel. Rublee *et al.* [8] extended FAST by adding an orientation component to the features. BRISK [9] is another feature detector that searches for maxima in both the image plane and the scale-space using the FAST scores as a measure for saliency.

Lowe [1] proposed SIFT, a method that is widely regarded as one of the most robust feature detectors available because of its invariance to scale, rotation, viewpoint, and illumination changes. SIFT features are computed by analyzing the Difference of Gaussian (DoG) between images at different scales. One of the main downsides to SIFT is that it is computationally expensive. Bay *et al.* [10] outlined this issue and proposed SURF, a feature detector that is similar to SIFT in that it is invariant to multiple image transformations, but is faster. As opposed to SIFT which analyzes the DoG, SURF analyzes the determinant of the approximated Hessian matrix in order to find the local maxima across all scales. Lourenco *et al.* [11] outline the problem of matching keypoints extracted from radially distorted images that are acquired by cameras with microlenses or wide field of view and propose a method that improves the repeatability of detection and effectiveness of matching under radial distortion.

Our method is closely related to [2], which used a rank order statistics to extract 3D features from a point cloud. Their method computed 3D features on a single metric scale available in the 3D data, while our method computes 2D features using a multi-scale representation to achieve

[1] Khalid Yousif and Alireza Bab-Hadiashar are with school of Engineering, RMIT University, Melbourne, VIC 3083, Australia s3362555@student.rmit.edu.au, abh@rmit.edu.au

[2] Yuichi Taguchi and Srikumar Ramalingam are with Mitsubishi Electric Research Labs (MERL), Cambridge, MA 02139, USA {taguchi,ramalingam}@merl.com

the scale invariance. Moreover, their aim was to obtain as small number of features as possible that are needed to register two RGB-D frames, while we aim to obtain a large number of high quality features to improve the camera pose estimation accuracy. This difference stems mainly from the fact that in RGB-D SLAM systems, 3D point measurements are typically stable and a smaller number of 3D features is sufficient for the pose estimation. On the other hand, in monocular and stereo SLAM systems, 3D points are triangulated from the inlier matches of 2D features; the triangulated 3D points are less stable than the measured 3D points in RGB-D SLAM systems, and the number of the 3D points is smaller because they are triangulated from at least two frames instead of measured in a single RGB-D frame.

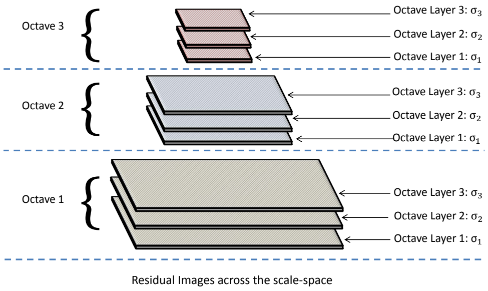

Octave 3
Octave Layer 3: σ_3
Octave Layer 2: σ_2
Octave Layer 1: σ_1
Octave 2
Octave Layer 3: σ_3
Octave Layer 2: σ_2
Octave Layer 1: σ_1
Octave 1
Octave Layer 3: σ_3
Octave Layer 2: σ_2
Octave Layer 1: σ_1
Residual Images across the scale-space

Fig. 1. An example of the constructed residual images across different scales. In this example, there are 3 octaves and 3 octave layers. Note that although the σ values are the same for different octaves, down-sampling the images into half its resolution has the same effect as doubling the scale (but more efficient than using large σ values).

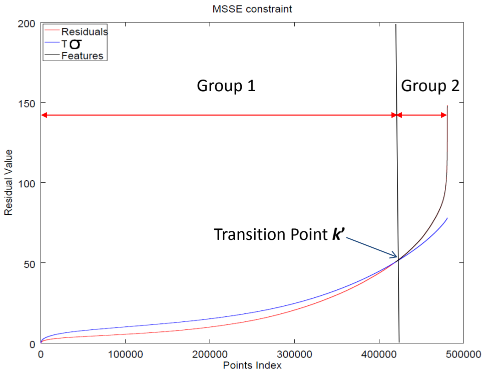


Fig. 2. MSSE segmentation of the residuals across the scale space. ROS2D features are selected as points in the second group associated with high residual values.

III. Rank Order Statistics 2D Features

In this section, we describe the main steps of detecting the Rank Order Statistics 2D features (ROS2D). As a preprocess, we convert the original color image to grayscale, and apply a histogram equalization method to the grayscale image to improve the lighting invariance of the features. The first step of the feature detector involves the construction of a multi-scale representation. This is followed by calculating a saliency measure (we will refer to it as a residual) for each point across both the image and scale dimensions. Using this measure, a robust data segmentation method (MSSE) is employed to find points at different scales with high intensity variations in comparison to their local neighborhood. This is followed by assigning an orientation component to each of the detected features. Finally, a descriptor is computed for each feature.

A. Multi-scale representation and calculating the residuals

Similar to SIFT, we construct a scale-space pyramid consisting of n octaves and m octave layers. Each octave is obtained by sampling the previous one into half its resolution. For each octave layer within an octave, a σ that corresponds to a Gaussian function at that scale is computed using the method described in [1]. Using each σ, a Gaussian kernel (discrete approximation to Gaussian function) is computed using the following equation [12]:

$$G_i = \alpha \exp\left(\frac{-(i - \frac{ksize-1}{2})^2}{2\sigma^2}\right), \tag{1}$$

where G_i is the ith Gaussian coefficient of the one dimensional kernel, α is a scale factor such that $\sum_i G_i = 1$, and $ksize$ corresponds to the size of the kernel. In order to obtain a $ksize \times ksize$ kernel, we simply multiply the one dimensional Gaussian kernel G by its transpose. We note that the size of the kernel is fixed across the scale space, and is calculated based on the σ of the highest octave layer. This is followed by calculating the kernel's Gaussian coefficients for each layer. We also note that the aforementioned kernels are not used as means to blur the images. The Gaussian coefficients at each octave layer are solely used as weights for the calculation of the residuals (saliency measures). The procedure to calculate the residuals is as follows. For each point in the scale-space pyramid, we assign a residual value using the following equation:

$$r = \sum_j w_j (I_c - I_j)^2, \tag{2}$$

where r is the residual at the query point, I_c and I_j are the intensity values of the query point and its jth neighbor within the kernel respectively, and w_j is a weight corresponding to the jth coefficient of the Gaussian kernel at that octave layer. The residual values are stored in residual images as shown in Figure 1. Since we have n octaves and m octave layers, this results in a total of $n \times m$ residual images. We assume that points with small residual values (close to zero) belong to regions with small intensity variations, whereas higher residual values correspond to regions with high intensity variations. Note that a point may be associated with a small residual on one scale (if the discrete Gaussian function falls inside a region), but a large residual on higher scales (if the discrete Gaussian function is larger than a region). In the following section, we will describe how the residuals are segmented and the features are selected.

B. Robust segmentation of the residuals

The next step involves segmenting the image points at all the scales into two groups based on their residual values. The first group contains points associated with small residuals, corresponding to regions with small intensity variations. The second group contains points with large residual values, corresponding to regions with high intensity variations. Similar to the procedure described in [2], we first sort the residual values in an ascending order, and then iteratively calculate the standard deviation of the sorted data using the first k sorted values using the following equation:

$$\sigma_k^2 = \frac{1}{n-p}\sum_{i=1}^{k} r_i^2, \tag{3}$$

where r_i is ith residual in the sorted square residuals vector, and p is the dimension of the model. Initial value of k corresponds to the assumed minimum percentage of points that could be considered a segment in the application. The transition point k' (k' corresponds to the new kth order that is flexibly found by MSSE) is found by iteratively incrementing k until the following condition is met:

$$|r_{k+1}| > T\sigma_k, \tag{4}$$

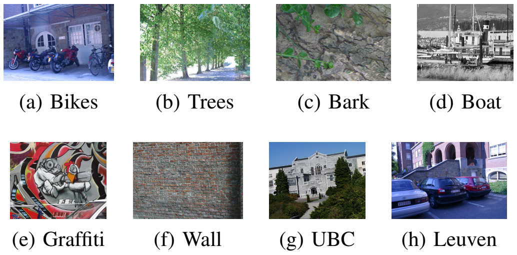
where T is a constant factor and is typically set to 2.5 to include 99% population of a normal distribution [13]. Figure 2 shows an example of the segmentation of the points based on their residual values using the MSSE constraint into two groups. Points associated with residuals in the second group are the selected features.

C. Orientation assignment

Each of the features selected from the above step is assigned an orientation value using the method described in [1]. A histogram consisting of 36 bins is formed around each feature covering 360° using the gradient orientations of points in a region around the feature (the size of this region is directly related to the scale at which the feature is selected). Each of the gradient orientations is weighed using the gradient magnitude at that point. The orientation value corresponding to the maximum value in the histogram is assigned to the feature. In addition, if there are other dominant orientation values within 80% of the maximum value, then new features are created to be identical to the original feature but assigned with a different dominant orientation. To achieve better accuracy, we interpolate the peak position by fitting a parabola to the 3 histogram values closest to each dominant orientation [1]. Assigning an orientation to each feature is crucial to achieve rotation invariance, since a feature descriptor can be computed relative to this orientation.

D. Feature descriptor

ROS2D features could be paired with any image descriptor. In our experiments, we assigned SIFT descriptors to ROS2D features due to its robustness and invariance to a number of image transformations. SIFT descriptors are obtained by dividing the region around the feature into 4×4 subregions. In each subregion, an orientation histogram of 8 bins is constructed. This information is then stored in $4 \times 4 \times 8 = 128$ byte description vector. We computed the descriptors on the histogram-equalized grayscale image, generated in the preprocess of the ROS2D detector. We observed that this slightly improves the lighting invariance of the features.



(a) Bikes (b) Trees (c) Bark (d) Boat

(e) Graffiti (f) Wall (g) UBC (h) Leuven

Fig. 3. Example of the Oxford dataset [3] used for the evaluation of the proposed method: blurring (a and b), rotation and scale (c and d), viewpoint change (e and f), JPEG compression (g), and lighting variation (h).

IV. Feature Performance Evaluation

The proposed ROS2D feature extraction method was implemented in C++ using the OpenCV 3.0 framework. We performed the experiments using a Dell Precision M3800, powered by an Intel i7-4702HQ processor and 16 GB of RAM. Our method was compared with other methods using the Oxford dataset [3] which includes 8 sets of images with different image transformations as shown in Figure 3. For each set (containing 6 images), the transformation level gradually increases (5 levels for each set). The image transformations included in this dataset cover Gaussian blurring (Bikes and Trees), rotation and scaling (Bark and Boat), viewpoint change (Graffiti and Wall), JPEG compression (UBC), and lighting variation (Leuven). The benchmark also provides ground truth information in the form of homographies between the first image and each of the other 5 images in each set.

A. Repeatability evaluation

Repeatability is widely considered as one of the most important attributes of a feature detector. To evaluate the repeatability performance of the proposed method, we use the evaluation method presented in [14]. The repeatability score is calculated as the ratio between the corresponding features and the total number of features that are viewed by both images. Features from two images are considered to be corresponding if the ratio between the overlapped area of their regions (after the projection of the feature's region of the second image into the first, using the ground truth homoghraphy transformation) and the union of the two regions is less than 0.5. In our case, a region is defined as a circle and is directly related to the scale at which the feature was detected. Figure 4 shows the repeatability scores of different feature extraction methods. The compared methods are ROS2D, BRISK [9], FAST [7], ORB [8], SIFT [1], and SURF [10]. The figure shows that the proposed method consistently detects repeatable features under a wide range of image transformations. The results show that the repeatability performance of the proposed method is consistently one of

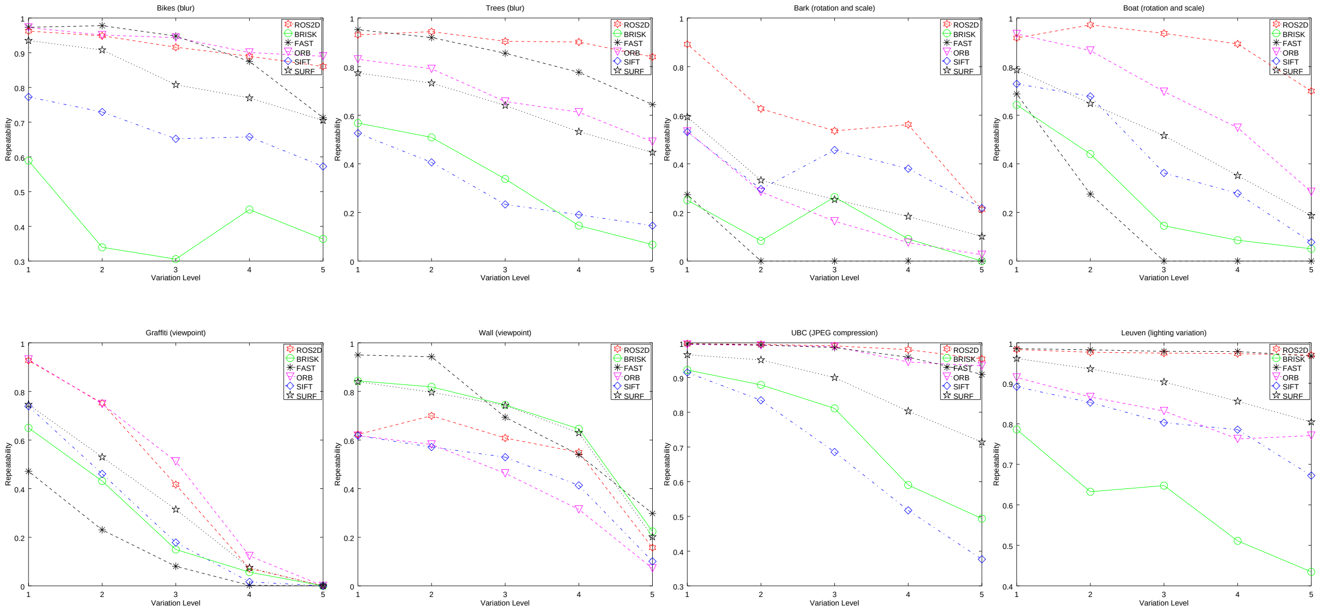


Fig. 4. Repeatability performance of the compared methods using the Oxford image sets with an increasing level of image transformations (1 to 5).

the highest in all datasets when compared with the state-of-the-art methods.

B. Matching performance evaluation

In this experiment, we pair the state-of-the-art feature detectors with descriptors and evaluate their matching performance using the same dataset used in the previous section. We perform a mutual consistency check, by finding the nearest neighbors in the descriptor vector space from the source frame to the target frame and vice versa. Only pairs of corresponding points that were mutually matched to each other were considered as the correspondences. We then compute the inlier ratio as $\frac{Number\_of\_correct\_correspondences}{Total\_number\_of\_correspondences}$.

We compute the number of correct correspondences by transforming the features (from the set of correspondences that were obtained from mutual consistency matching) from the source frame to the target frame using the ground truth homography transformation. A correspondence is assumed to be correct if the L2 distance between the transformed feature and the target feature is less than a predefined threshold (two pixels). The results of this experiment are plotted in Figure 5, which show that the proposed method performs well consistently under various image transformations. The proposed method performs particularly well under the scale and rotation, lighting, and JPEG compression variations. A main advantage of the proposed method is its performance consistency in comparison to the other methods.

Figure 6 shows the average number of inliers obtained by the matching procedure described above using all the image sets. It can be clearly seen that the proposed method is able to provide a large number of repeatable features that are also invariant to various image transformations. Extracting high quantity and quality features is one of the main advantages of the proposed approach.

C. Limitations

In the previous sections, we showed that our method outperforms existing methods in terms of the repeatability and inlier ratio. Here we discuss the limitations of our method.

Precision-recall performance: Figure 7 presents comparisons between different feature extraction methods using the recall $(\frac{number\_of\_correct\_matches}{number\_of\_correspondences})$ vs precision $(\frac{number\_of\_correct\_matches}{total\_number\_of\_matches})$ curve for two image sets of the Oxford dataset. Two features are said to be matched if the distance between their descriptors is lower than a predefined threshold t. The value of t is varied to obtain the aforementioned curve. The results show that SIFT and SURF outperform ROS2D in this test, particularly as we increase t, since the number of false matches are increased at a higher rate for ROS2D. Note that our method still provides good top matches (left most part of the plots), although the entire precision-recall performance is relatively low, which concerns not only the top matches but also the subsequent matches. The top matches are more important when we perform sequential image matching, e.g., in SLAM, because the images have large overlaps and similar appearance.

Processing time: Our method is computationally expensive due to the fact that the residuals are computed for all the image points at all the scales and that MSSE performs a sort and a linear search. We compared the processing time between ROS2D and SIFT. We extracted features from the Graffiti image set, computed the extraction time, and averaged the results. For a fair comparison, we extracted approximately the same number of features for both methods (around 2600 features). The detection times were 648 ms

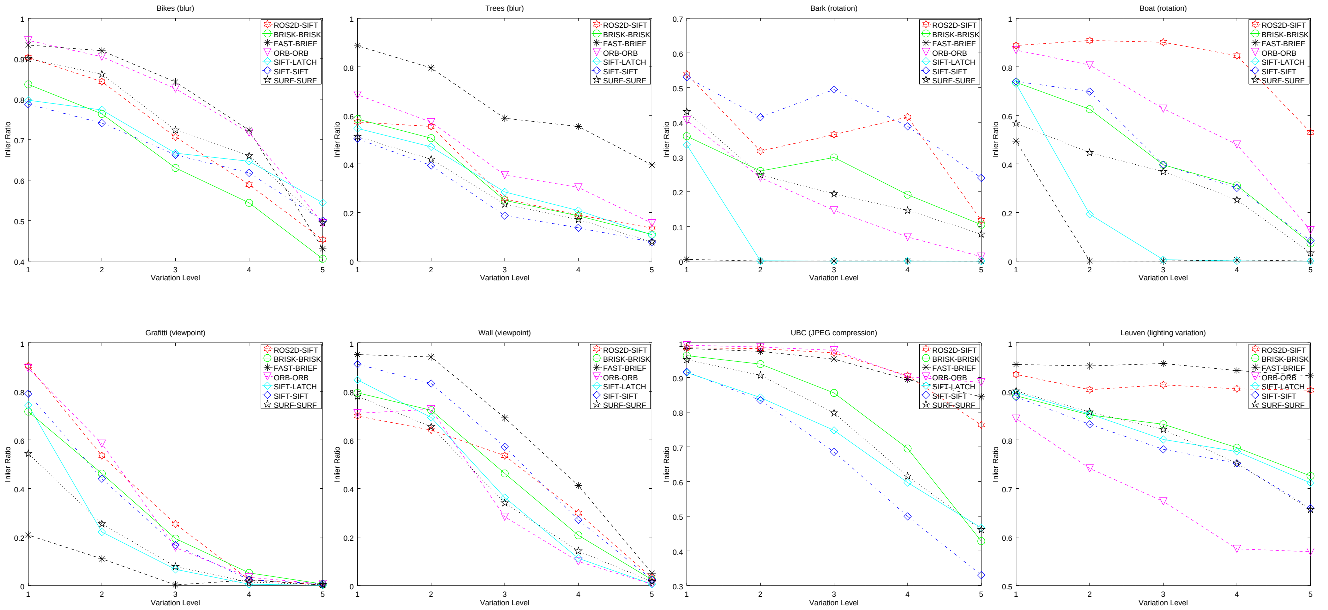


Fig. 5. Inlier detection performance of the compared methods using the Oxford image sets with an increasing level of image transformations (1 to 5).

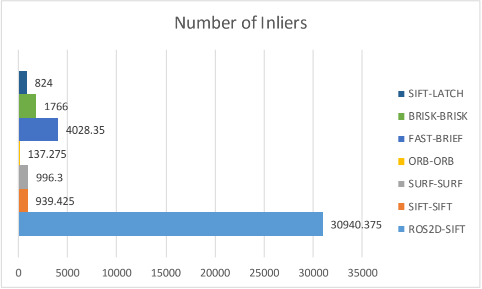


Fig. 6. The average number of inliers obtained with the compared methods using all the Oxford image sets.

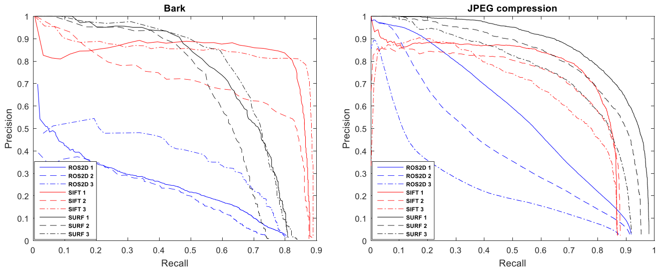


Fig. 7. Precision-recall performance of different feature extraction methods using the Oxford dataset. For each method, 3 curves corresponding to 3 different difficulty levels are plotted.

and 184 ms for ROS2S and SIFT respectively. SIFT feature detection time was faster than the proposed method by a factor of 3.5.

V. Evaluation in SLAM Systems

In this section, we compare the performance between the proposed method (ROS2D detector paired with SIFT descriptor) and SIFT (detector and descriptor) when we use them in stereo and monocular SLAM systems. We show that a larger number of features extracted with our method is helpful to improve the camera pose estimation accuracy in the SLAM systems.

(a) (b) (c)

Fig. 8. Sample images from the KITTI datasets showing different types of environments: (a) urban, (b) country, and (c) highway.

A. Stereo SLAM evaluation

We implemented a stereo SLAM system and used the exact same framework to compare the accuracy of ROS2D with SIFT. The stereo SLAM system first extracts features from the stereo pair, matches them, and obtains their 3D coordinates via triangulation. In the next step, the system matches sequential left images and refines those matches using a RANSAC-MSSE geometric verification step based on the 3D-to-3D correspondences. The aforementioned step results in estimating transformations between sequential images, which are then concatenated up to the current time to obtain the global poses of the frames. To reduce the drift and preserve local consistency, we implemented a sliding window bundle adjustment, in which the measurements obtained in the previous 10 stereo image pairs are used to refine the camera pose estimates as well as the point landmarks.

We evaluated the accuracy of the stereo SLAM system using the KITTI dataset [4]. The dataset is captured using a stereo camera mounted on top of a vehicle driving around in various environments. The vehicle is also equipped with a high accuracy GPS for retrieving ground truth trajectories. Each image has a 1230 × 370 resolution and a 81° horizontal field of view. The dataset contains 11 sequences provided with the ground truth trajectories. These sequences mainly include three types of scenes: urban with surrounding buildings, country containing small roads with vegetations in the scene, and highway containing wide roads [15]. Examples of these sequences are shown in Figure 8. The accuracy is evaluated using average translational and rotational errors for segments of lengths $100, 200, 300, \ldots, 800$ meters. Translational errors are measured as a percentage of the distance traveled with respect to each of the aforementioned segment lengths, whereas rotational errors are measured in degrees per meter [4].

Evaluation using different numbers of features: In the first experiment, we study the effect of using different numbers of features on the pose estimation accuracy. We evaluated the accuracy of the proposed method and SIFT using the first KITTI sequence (00) which consists of 4541 stereo images captured in an urban environment. We calculated the translational and rotational errors (using the ground truth trajectories) by varying the maximum number of extracted features for the compared methods. For the proposed method, we set the maximum number of features simply by selecting an n number of points after the transition point k' (see Figure 2). To vary the number of extracted SIFT features, we changed the "nfeatures " parameter in the OpenCV SIFT implementation [12], which allows the user to select the number of best features to retain by ranking by their scores (measured as the local contrast).

The results of this evaluation are shown in Figure 9. Note that the maximum number of SIFT features that we were able to extract was around 6800, whereas the proposed method was able to extract up to 33120 features. Figures 9 (a) and (b) show that the rotational and translational errors are correlated, and the most accurate results for SIFT were obtained when using around 4150 features. In general, we observed that more features provide more accuracy, but the improvement saturates after we provide enough features. We found that when using the proposed method, no significant accuracy gain was achieved using more than 8000 features. Overall, the most accurate results achieved by the proposed method outperformed the most accurate results obtained by SIFT by 17.3% for translation and 15.7% for rotation. Figure 9 (c) shows that the inlier ratio decreases when more features are extracted for SIFT, whereas it increases when extracting a larger number of ROS2D features. For SIFT, more features are obtained by changing the parameter explained above, which may reduce their distinctiveness due to the selection of lower ranked features and result in a larger number of false matches. On the other hand, more ROS2D features are obtained by taking more points after the transition point k'. These points are associated with high value residuals, meaning that they belong to regions with high intensity variations and are likely to be highly distinctive. Having said that, we found that only selecting features with the highest residuals (well beyond k') does not necessarily provide the most accurate pose estimation results (although their matching performance is high), since this may result in only selecting features from particular regions in the image (e.g., features obtained from distant objects, which are not adequate for estimating the translation). We found that features with relatively lower residual values (but also beyond k') could provide us with valuable information for accurately registering the frames.

Evaluation using different KITTI sequences: In this evaluation, we compared the accuracy of our method with SIFT using the 11 KITTI sequences. We set both methods' parameters as described above. The results are summarized in Table I and example trajectories are shown in Figure 10. The results show that the proposed method outperformed SIFT in all sequences except for sequences 03 (both methods produced comparable results) and 06 (SIFT produced slightly more accurate results). Note that the trajectories shown in Figure 10 do not show the hidden errors in the vertical position of the camera (y-axis). Also note that despite the proposed method outperforming SIFT for the 01 sequence, both methods produced large errors. This sequence consists of frames captured on a highway, and the bad results may be attributed to outliers that have similar attributes to inliers. Those outliers are not a direct result of completely bad matches or random errors. They belong to structures corresponding to a different (and incorrect) motion. For instance, some features may exhibit no parallax (the angle between the captured rays is the feature's parallax) during camera motion due to them having a very large depth [16]. Such features are only able to constrain the rotational component of the camera motion, thus providing an inaccurate translation. Regardless of the type of features used, identifying such motions is a difficult task, and we plan on resolving this issue in our future work. Another scenario where both methods' accuracy deteriorated was that some sequences contained cars moving at various speeds. The robust estimation method used in this approach was generally able to correctly segment the correspondences associated with the camera motion. However, when another car was driving at a similar speed, correspondences located on this car were mistakenly considered as inliers and used to estimate the camera motion. This results in a very small but continuous drift throughout the sequence [17]. For instance, this scenario occurs on the 04 sequence and an example of this problem is depicted in Figure 11. The figure shows the inlier correspondences that were obtained after performing the robust estimation step. It can be seen that a few (around 5) correspondences were located on a vehicle driving in front of the camera. We will focus on motion segmentation in our future work.

B. Monocular SLAM evaluation

In this section, we show the 3D reconstruction results of the proposed method for monocular sequences obtained from public datasets [18], [19]. Similar to the stereo SLAM system described in Section V-A, we implemented a monocular SLAM system that initially employs a 5-point RANSAC algorithm [20] to estimate the camera motion between the

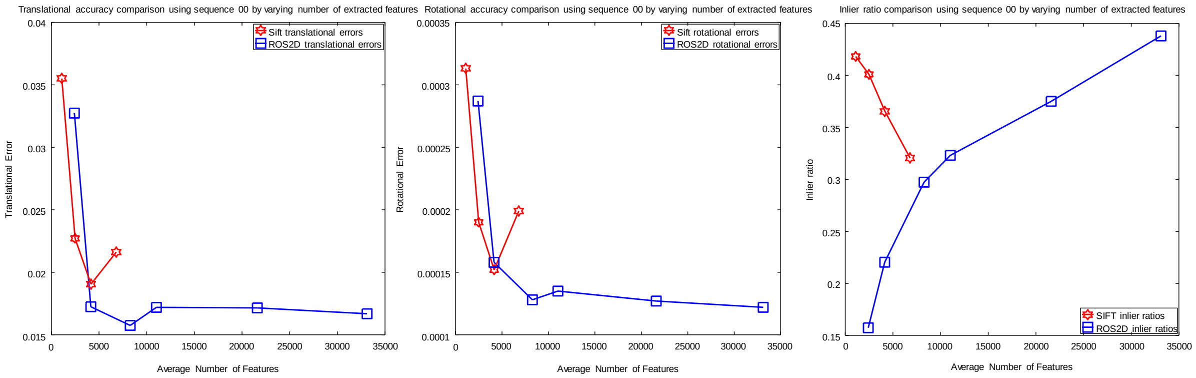


Fig. 9. Comparison of the effect of using a different number of features on the pose estimation accuracy using the 00 sequence: (a) translational errors, (b) rotational errors, and (c) inlier ratio.

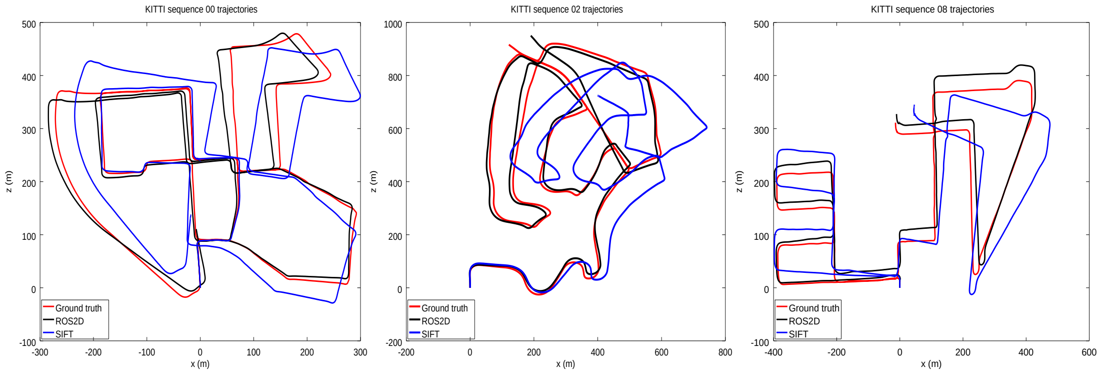


Fig. 10. Stereo SLAM results for three different KITTI sequences.

TABLE I

THE AVERAGE ROTATIONAL AND TRANSLATIONAL ERRORS FOR THE PROPOSED METHOD VS. SIFT USING 11 DIFFERENT KITTI SEQUENCES.

Sequence No.	Environment	Distance	ROS2D-SIFT		SIFT-SIFT	
			Average trans. err. (%)	Average rot. err. (deg/m)	Average trans. err. (%)	Average rot. err. (deg/m)
00	Urban	3714m	1.5749	0.0073	1.9051	0.0087
01	Highway	4268m	26.1425	0.0285	1205.4239	0.0116
02	Urban+Country	5075m	1.9133	0.0091	5.6095	0.0241
03	Country	563m	4.7215	0.0184	4.1427	0.0202
04	Country	397m	4.6817	0.0070	6.0074	0.0077
05	Urban	2223m	3.7162	0.0151	3.978	0.0107
06	Urban	1239m	5.8266	0.0263	4.5559	0.0134
07	Urban	695m	4.8446	0.0277	10.2396	0.0523
08	Urban+Country	3225m	2.6416	0.0083	3.3316	0.0106
09	Urban+Country	1717m	4.0464	0.0108	4.8863	0.0108
10	Urban+Country	919m	2.8794	0.0111	4.0707	0.0191

first two frames and triangulate the 3D points of their correspondences (up to a scale). For the remaining frames, we used a P3P algorithm to estimate the relative poses. We also used a global bundle adjustment algorithm in place of the sliding window bundle adjustment used previously.

Figure 12 shows the 3D reconstruction results of the Temple dataset [18] (only first 13 images were used) and the Sceaux Castle dataset [19] (contains 11 images) using both ROS2D and SIFT features. It can be seen clearly that the proposed method was able to provide much denser models in comparison to SIFT. In addition, Figure 13 shows the 3D reconstruction results obtained by the proposed method using the first 30 images of the Dino dataset [18]. We note that the SLAM failed when using SIFT features for this sequence, as there were not enough inliers detected by the initial 5-point RANSAC step.

VI. CONCLUSIONS

We proposed a 2D feature detector that selects features using a robust rank order statistics segmentation method. The main idea is to segment points with high local intensity variations across different scales. In the experimental evaluation, we showed that our method is able to obtain a large number of high quality and repeatable features. We

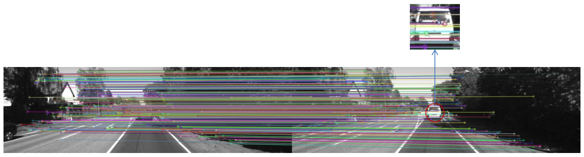


Fig. 11. Inliers obtained by the robust estimation method when matching sequential images. It can be seen that some correspondences were located on a moving object (car), as it was driving at a similar speed to the moving camera.

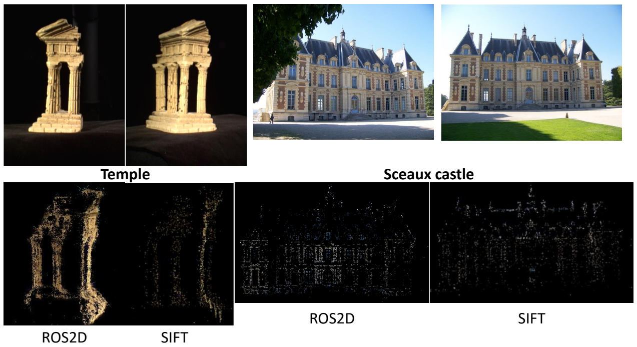


Fig. 12. Top: sample images from the Temple [18] (left) and Sceaux Castle datasets [19] (right). Bottom: 3D reconstruction results using ROS2D and SIFT features.

also showed that our method performs better than state-of-the-art methods in terms of the repeatability and inlier ratio, and that the features obtained by our method are invariant to various image transformations such as rotation, blurring, and lighting variations. In addition, we showed that our method outperformed SIFT when used in SLAM systems in terms of the pose estimation accuracy. We finally showed that using ROS2D features produced denser 3D models than using SIFT. The main limitation of ROS2D is its computational complexity, and in our experiments, SIFT was faster than our method. We note that increasing the number of keypoints leads to increasing the computational burden at description time and therefore the description step could be optimized in the future. In addition, the robust segmentation step consists of storing the residuals and searching for the point that separates features from the rest of the points. This step is computationally expensive, and in the future we plan on using faster search methods such as a binary search, to improve the efficiency of the proposed method.

Acknowledgements

We thank Bharath Sankaran and Ryosuke Sasaki for valuable discussion. This work was supported by MERL.

References

[1] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, “3d slam in texture-less environments using rank order statistics,” *Robotica*, pp. 1–23, 2015.

[3] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[5] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[6] K. Mikolajczyk and C. Schmid, “Indexing based on scale invariant interest points,” in *Proceedings of the 2001 IEEE International Conference on Computer Vision (ICCV 2001)*, vol. 1. IEEE, 2001, pp. 525–531.

[7] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” *Proceedings of the 2006 European Conference on Computer Vision (ECCV 2006)*, pp. 430–443, 2006.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: an efficient alternative to sift or surf,” in *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV 2011)*. IEEE, 2011, pp. 2564–2571.

[9] S. Leutenegger, M. Chli, and R. Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV 2011)*. IEEE, 2011, pp. 2548–2555.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” *Proceedings of the 2006 European Conference on Computer Vision (ECCV 2006)*, pp. 404–417, 2006.

[11] M. Lourenco, J. P. Barreto, and F. Vasconcelos, “srd-sift: Keypoint detection and matching in images with radial distortion,” *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 752–760, 2012.

[12] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[13] A. Bab-Hadiashar and D. Suter, “Robust segmentation of visual data using ranked unbiased scale estimate,” *Robotica*, vol. 17, no. 6, pp. 649–660, 1999.

[14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, “A comparison of affine region detectors,” *International journal of computer vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[15] J. Zhang, M. Kaess, and S. Singh, “Real-time depth enhanced monocular odometry,” in *Proc. IEEE/RSJ Int’l Conf. Intelligent Robots and Systems (IROS)*, Sep. 2014.

[16] J. Montiel, J. Civera, and A. J. Davison, “Unified inverse depth parametrization for monocular slam,” in *Proceedings of the 2006 Robotics: Science and Systems (RSS)*, 2006.

[17] G. Ros, J. Alvarez, and J. Guerrero, “Motion estimation via robust decomposition with constrained rank,” *arXiv preprint arXiv:1410.6126*, 2014.

[18] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” in *Proceedings of the 2006 IEEE Computer Society Conference on Computer vision and pattern recognition (CVPR 2006)*, vol. 1. IEEE, 2006, pp. 519–528.

[19] P. Moulon, P. Monasse, and R. Marlet, “Adaptive structure from motion with a contrario model estimation,” in *Proceedings of the 2012 Asian Conference on Computer Vision (ACCV 2012)*. Springer, 2012, pp. 257–270.

[20] D. Nister, “An efficient solution to the five-point relative pose problem,” in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, vol. 2. IEEE, 2003, pp. II–195.

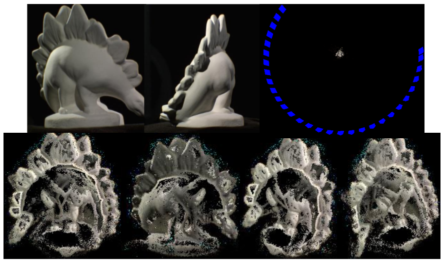


Fig. 13. Top-left: Sample images from the Dino dataset [18]. Top-right: Estimated camera poses. Bottom: 3D reconstruction results using the proposed method.