2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Improved Local Shape Feature Stability Through Dense Model Tracking

Daniel R. Canelhas, Todor Stoyanov, Achim J. Lilienthal
Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden

Abstract— **In this work we propose a method to effectively remove noise from depth images obtained with a commodity structured light sensor. The proposed approach fuses data into a consistent frame of reference over time, thus utilizing prior depth measurements and viewpoint information in the noise removal process. The effectiveness of the approach is compared to two state of the art, single-frame denoising methods in the context of feature descriptor matching and keypoint detection stability. To make more general statements about the effect of noise removal in these applications, we extend a method for evaluating local image gradient feature descriptors to the domain of 3D shape descriptors. We perform a comparative study of three classes of such descriptors: Normal Aligned Radial Features, Fast Point Feature Histograms and Depth Kernel Descriptors; and evaluate their performance on a real-world industrial application data set. We demonstrate that noise removal enabled by the dense map representation results in major improvements in matching across all classes of descriptors as well as having a substantial positive impact on keypoint detection reliability.**

I. Introduction

With the recent availability of affordable 3D range sensors, a number of algorithms for simultaneous sensor tracking and environment reconstruction using dense models have been proposed. Several of these approaches [1], [2] use Signed Distance Functions (SDF) to achieve highly detailed 3D models of small-scale environments by consistently fusing depth information from multiple camera viewpoints. While this class of approaches has produced impressive tracking and modeling performance, the obtained models have so far only been used in augmented reality and mapping applications. As suggested also in [1], however, consistent SDF environment models have a number of potential applications in robotics. In this article we discuss one such application of particular relevance to robotics — improving the performance of local shape features.

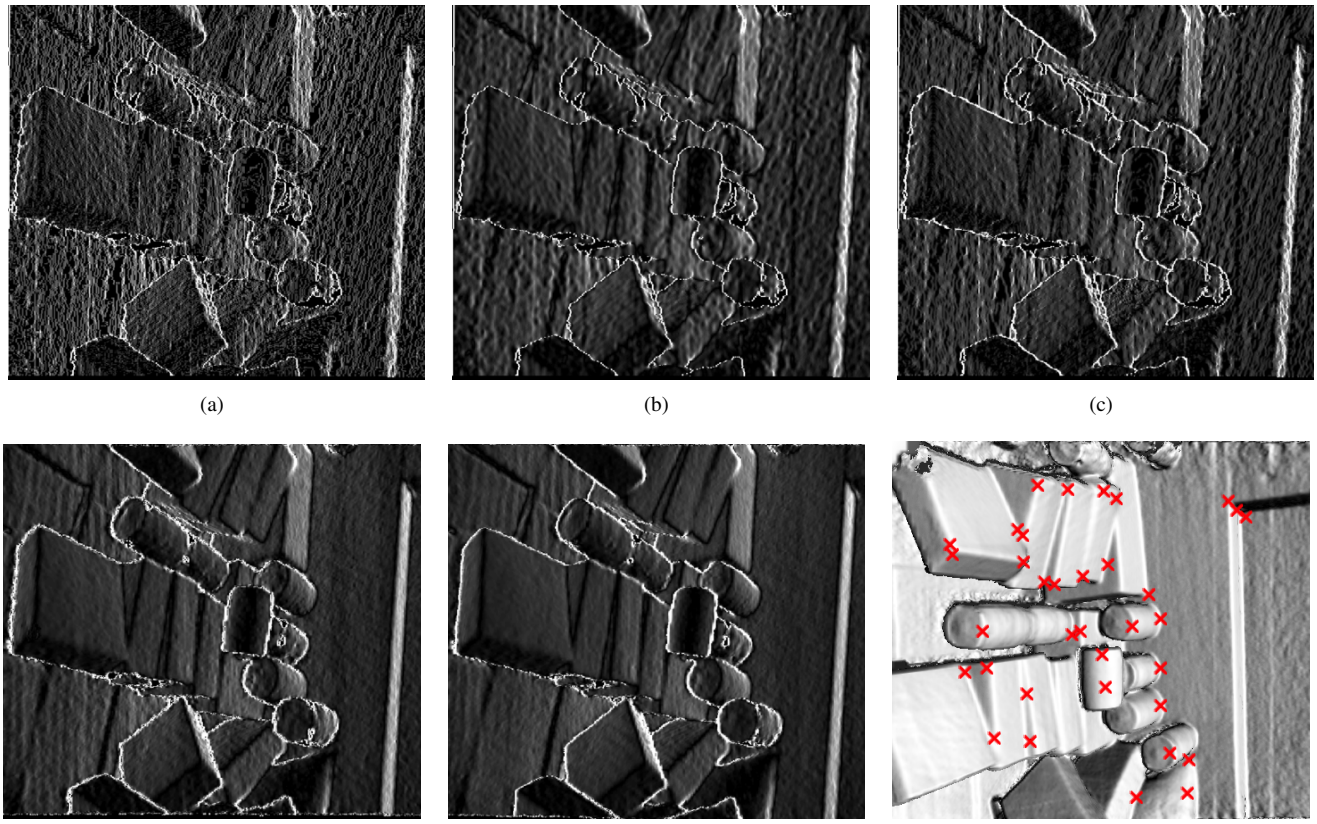
Recent advances in robot perception algorithms show that salient keypoints and local feature descriptors continue to play a big role in many important tasks. When performing object recognition, scan alignment or localization, many methods rely on detecting informative regions in the available sensor data. The sensor data at such keypoint locations can then be encoded, using a feature descriptor and matched to other descriptors, computed from previous observations or loaded from a database. While this process is well understood for local visual feature approaches, algorithms that operate on depth data are a recent development that has been given much less attention.

Depth data is of high relevance in robotics — virtually all autonomous mobile robots are equipped with a range sensing device. It is thus not surprising that shape-based feature descriptors have been proposed and used in a variety of robotic applications: scan alignment [3], place recognition and loop detection [4], [5], and object detection [6], [7], to name but a few. Many of the recent contributions focus on improving the consistency of salient feature detectors, improving descriptors to become robust to viewpoint variations and noise and finding suitable metrics for matching descriptors. In this work we propose a different approach to improving the performance of depth-based local features. Instead of attempting to reduce the sensitivity to noise of either detectors or descriptors, we examine the impact of denoising the sensor data by means of a recently proposed Truncated SDF (TSDF) reconstruction algorithm[1] [2]. We use the TSDF tracking and mapping algorithm to incrementally generate a dense, implicit surface representation that combines the depth images into a common frame of reference. Denoised depth images are then produced by ray-casting the dense surface from the tracked position of the depth sensor and used for feature detection and extraction.

Filtering the depth sensor noise is not in general guaranteed to result in better keypoint detection or feature extraction results, since removing the noise may come at the price of smoothing out important geometric structure. We examine two standard noise removal approaches — bilateral filtering [8] and total variation L_1 norm (TV-L1) filters [9], and compare their effect on feature stability and accuracy to that of the proposed SDF denoising approach. In order to evaluate the effect of these noise removal approaches, we use several recent depth-based feature detection and extraction algorithms. Using a moving camera setup, we evaluate the stability of the Normal Aligned Radial Feature [10] (NARF) interest point detector over increasing baselines in translation and rotation. We also compare the success in matching of NARF, Fast Point Feature Histogram (FPFH) [3] and Depth Kernel [6] descriptors, over the same range of motion.

The influence of these noise removal approaches on keypoint stability and feature matching are evaluated on an industrially relevant dataset from an automated logistics application scenario. The proposed SDF denoising approach is demonstrated to consistently improve the stability and matching performance of local shape features, clearly outperforming the alternative noise filtering approaches. The main contributions of this article are thus two-fold. First, we propose and formulate denoising as a novel application of SDF tracking and modelling and demonstrate that it results

[1] http://www.ros.org/wiki/sdf_tracker



(a) (b) (c)

(d) (e) (f)

Fig. 1. Gradient magnitude for typical depth images, processed with different noise filtering techniques. Fig. 1(a): raw depth image obtained from the sensor, Fig. 1(b): bilateral filter, Fig. 1(c): TV-L1 filter, Fig. 1(d): incremental SDF denoising, Fig. 1(e): full SDF denoising. Finally, Fig. 1(f) shows the reconstructed SDF surfaces along with manually selected keypoint locations.

in substantial improvement of both keypoint stability and feature description. Second, in order to perform a quantitative evaluation, we adapt a method from computer vision [11] to the domain of local shape features, resulting in a comparison of several recently proposed feature descriptors — NARF, FPFH and Depth Kernels. While the computer vision community has produced several comparisons of local visual features [12], [11], [13], and there have been several works evaluating the performance of object classifiers based on local shape features [14], [15], [16], to the best of our knowledge this article presents the first direct comparative evaluation of local shape features.

This article proceeds with an overview of the depth image noise filtering approaches evaluated in this work. Sec. III presents the local shape detectors and descriptors, selected for benchmarking and evaluation. We then present the proposed evaluation methodology in Sec. IV, before reporting and analyzing results in Sec. V. Finally, Sec. VI concludes this paper with a summary of the lessons learned and directions for future investigations.

II. Noise Filtering

In this section we outline relevant noise filtering approaches for depth image denoising. An example of a raw depth image, as well as the filtered outputs produced by the discussed approaches is shown in Fig. 1 and further explained in the next subsections.

A. Bilateral Filter

The bilateral filter is a nonlinear filter that updates a pixel $\boldsymbol{p}$ in an image I with a weighted sum of the values of its neighbours $\boldsymbol{q}$. Unlike a regular Gaussian smoothing filter, which takes into account only the pixel-space distance between $\boldsymbol{p}$ and $\boldsymbol{q}$, the bilateral filter also considers the difference in *intensity* at these pixels. Formally [17],

$$BF[I]_{\boldsymbol{p}} = \frac{1}{W_{\boldsymbol{p}}} \sum_{\boldsymbol{q} \in S} G_{\sigma_s}(\|\boldsymbol{p} - \boldsymbol{q}\|) G_{\sigma_r}(\|I_{\boldsymbol{p}} - I_{\boldsymbol{q}}\|) I_{\boldsymbol{q}} \quad (1)$$

where G are Gaussian PDF's with variances σ_s in pixel-space and σ_r in intensity-space, and $W_{\boldsymbol{p}}$ is a normalization factor. The bilateral filter is an edge-preserving low-pass filter, well suited and previously applied [1] for depth image denoising.

B. Total Variation - L1 Filter

The TV-L1 filter is based on the observation that the noise in an image can be described by the *total variation* of a pixel's intensity value, relative to its neighbours. Minimizing the total variation between pixels, therefore reduces the noise, but may also smooth out important features of the image, such as edges and corners. In order to preserve edges, the filter output values are kept "close" to the original image pixels in I using a regularized penalty, proportional to the norm of the difference. The TV-L1 filter can then be

formulated as an optimization problem:

$$\underset{u \in X}{min.} \|\nabla u\|_1 + \lambda \|u - g\|_1, \quad (2)$$

where u is a vector of filtered pixel values, g is a vector containing the original values from I, λ is a regularization parameter and X is the space of attainable pixel values. Finding efficient means for solving this problem is still an active area of research, though many approaches already exist. In this work we employ a method proposed by Chambolle and Pock [9], results from which are shown in Fig. 1(c).

C. SDF for modelling and depth image denoising

The TSDF is a 3D grid-based map representation, which stores a truncated distance value in each grid cell, measuring proximity to the closest object surface. Cells located on the inside of objects are assigned negative valued distances and cells outside of objects positive. Thus, the precise surface locations can be obtained by detecting the TSDF zero crossing (Fig. 1(f)). In addition, tri-linear interpolation between adjacent grid cell values is used to obtain a continuous function $TSDF(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^3$. Finally, the gradient of the 3D distance field coincides with the surface normal orientation and is computed numerically using a finite differences method.

Distance fields can be generated and used in many ways [18]. Jones et al. [18] note that unless given a closed and oriented surface representation, generating a true signed distance field is not an easy task. To counter this fact, recent work of Newcombe et al. [1] (and originally proposed by [19]) uses projective line-of-sight distance to reconstruct the SDF. Albeit being an approximation, projective distance agrees well with the true signed distance close to surface boundaries and further motivates the truncation of the field at pre-determined positive and negative limits. Other notable uses of distance fields, that favour our application include acceleration of raycasting [20] and Iterative Closest Point (ICP) registration [21].

In this work, we use the recently proposed SDF tracker algorithm [2], which generates and uses a TSDF map for on-line camera pose estimation. Obtaining a denoised map I_D from the virtual camera position $\boldsymbol{c}$ given a TSDF can be accomplished using standard SDF raycasting [20], detailed in Algorithm 1 for completeness. Unlike the previously discussed single frame noise filtering approaches, SDF denoising uses all prior information and incorporates viewpoint knowledge, in order to produce a filtered image. We compute two types of denoised depth images — rendered from the incrementally constructed TSDF in an online fashion (Fig. 1(d)), and obtained using the TSDF of the full data set retroactively (Fig. 1(e)) for offline applications. Both are highly relevant to several recurring tasks in robotics — ranging from object detection to place recognition.

Algorithm 1 Standard depth image raytracing in a TSDF

1: $\alpha \leftarrow 0$
2: **for** $\forall \boldsymbol{u} \in I_D$, over a maximum number of iterations **do**
3: compute the ray $\bar{\boldsymbol{r}}$ through $\boldsymbol{u}$ originating from $\boldsymbol{c}$ using a pinhole camera model
4: $D = TSDF(\boldsymbol{c} + \alpha \bar{\boldsymbol{r}})$
5: **if** $D < 0$ **then**
6: interpolate α based on current and previous D
7: return $I_D(\boldsymbol{u}) = \alpha(\bar{\boldsymbol{r}}_3)$
8: **else**
9: $\alpha = \alpha + D$

III. Depth Features on Noise-Filtered Models

In this section we present a brief overview of the NARF keypoint detector, and several feature descriptors. We chose the following methods because of their simplicity (NARF), reported performance (Kernel descriptors) and apparent popularity in the community (FPFH). Note that a point cloud and a range image can both be computed from a depth image, respectively by a projection of depth pixels to 3D points and by an inverse projection to 2D, storing the length of the rays connecting the camera and points. Note also that we do not consider run-time efficiency in this evaluation. However, our results appear to generalize well over both simple and complex descriptors alike.

A. NARF keypoint detector

The NARF keypoint detector finds regions of interest in a given range image by first locating the boundaries of objects, defined by large differences in range at adjacent pixels. A score is then computed for the local variation around each pixel, compared to the variation of neighbouring regions. Keypoints are determined as the locations that are different from neighbouring regions, but relatively uniform within their immediate surroundings, as this promotes repeatability in detection.

B. NARF feature descriptor

The NARF feature descriptor is computed at a given pixel in a range image, by defining a patch perpendicular to the estimated surface normal at the pixel. The pixel intensities within this patch are then evaluated along predefined directions radiating out from the query pixel. The pixel intensity variation along each radial direction is taken to be one dimension of the descriptor's feature vector. For rotation invariance, a dominant direction is also identified within the patch and the feature vector is shifted relative to this direction.

C. Kernel Descriptors

Kernel descriptors define similarity between patches using kernel functions based on pixel attributes. A machine learning approach is then used to reduce the dimensionality of the kernel functions such that a basis for the most informative dimensions can be stored and used on-line for matching. In this work we evaluate three different kernel descriptors: namely, gradient kernel descriptors which express a difference in surface gradients between patches; local binary patch (LBP) kernels, which encode and compare patterns of local depth variation; and Spin/Normal kernel descriptors which

Fig. 2. The industrial robot and target environment, used as an application scenario for our evaluation.

measure the difference between surface normals around a given point.

D. Fast Point Feature Histogram Descriptors

FPFH has been made popular as part of the point cloud library [22] (PCL), and is a method designed to work on *unstructured* point cloud data, with no assumption regarding adjacency or viewpoint e.g. as implied by a depth image. An FPFH feature vector is computed by first performing a neighbourhood search in 3D, collecting points within a region of pre-defined size. Points are then considered pairwise, along with their estimated surface normals, and a set of angles are computed based on geometrical relationships between the points, their normals and the vector defined between them. The results are weighted based on the distances between the points and binned into a histogram which represents the descriptor's feature vector for that particular point.

IV. Evaluation Methodology

In this section, we outline the procedure used for evaluating the performance of the geometrical feature detectors and descriptors discussed in the previous section. The approach is inspired by a recent work which evaluates visual feature detectors and descriptors in the context of object modelling [11]. We use a two step evaluation procedure — we first test the stability of feature keypoint detectors over multiple data frames and then proceed to evaluate the accuracy of feature descriptors. The results from these evaluation procedures are then used to compare the performance of detectors and descriptors both against each other, as well as over varying types of filtered input depth data.

In all of the performed evaluations, we use datasets collected from a moving Asus Xtion Pro depth camera with a resolution of 640x480 depth pixels. The sensor is mounted on a six axle industrial manipulator, designed for offloading of shipping containers (see Fig. 2). During data collection, a container is filled with goods, typically found in containers — cardboard boxes of various sizes, barrels, sacks and tyres (see Fig. 1(f)). These three types of goods account for over 65% [23] of all container loads and are of high interest for automated unloading applications. The manipulator is then instructed to follow a typical unloading pattern while recording depth camera data, and two representative portions of the data set are selected for use in evaluation.

Given a sequence of depth images D_i, we use the SDF tracker algorithm [2] to obtain an estimated camera trajectory. Thus, for every depth image D_i we associate a global camera pose in the form of a homogeneous transformation matrix T_i. As the length of the camera trajectories used in this evaluation are relatively short and only cover a moderately sized environment, the estimates from the SDF tracker algorithm [2] are virtually drift-free. Therefore, for the evaluation data sets used in this work, we treat the estimated camera poses T_i as a ground truth input to the evaluation procedure.

A. Keypoint Detectors

The first type of evaluation performed measures the stability of detected keypoint locations, over a sequence of depth data D_i. Keypoints are commonly used to greatly reduce the number of feature vectors computed in a typical image region of interest. In order to ensure good performance, a keypoint location needs to be distinctive, informative, and crucially it needs to be repeatably detected. Thus, given two depth images D_i and D_j collected at two different camera poses T_i and T_j, we can evaluate the stability of a keypoint detector by calculating the percentage of keypoint locations, detected in both depth frames. We calculate this ratio by first transforming both sets of keypoints from local pixel coordinates to their camera-centric 3D positions k_i and k_j, using the camera projection matrix. Next, we use the known camera positions to obtain the global keypoint coordinates $K_i = T_i k_i$ and $K_j = T_j k_j$. Finally, we calculate the ratio of keypoints from frame i that have close neighbours in frame j, over all keypoints in frame i:

$$s_{i,j}(t_k) = \frac{|dist(K_i, K_j) < t_k|}{|K_i|} \tag{3}$$

where $dist(K_i, K_j)$ is the euclidean distance between each keypoint from K_i and it's closest neighbour in K_j and $|\cdot|$ denotes the set cardinality operator. This score function is parametrized on a keypoint association distance t_k, which signifies the maximum acceptable deviation in a keypoint position. In this evaluation we do not check for occluded keypoint locations or keypoints that do not belong to both fields of view, and thus even a perfect detector will not achieve a score ratio of one. The camera motion between different evaluated frames is however relatively small and occlusion effects are minimal. In addition, in this evaluation we are interested primarily in comparing keypoint stability over differently processed depth images and thus the keypoint ratio is only used as a relative and not an absolute measure. Finally, we compute the feature stability score $s_{i,j}$ over varying translation and rotation between the input depth images D_i and D_j and report stability histograms.

B. Feature Vector Descriptors

The second type of evaluation performed in this work measures the uniqueness of different feature vector descriptors.

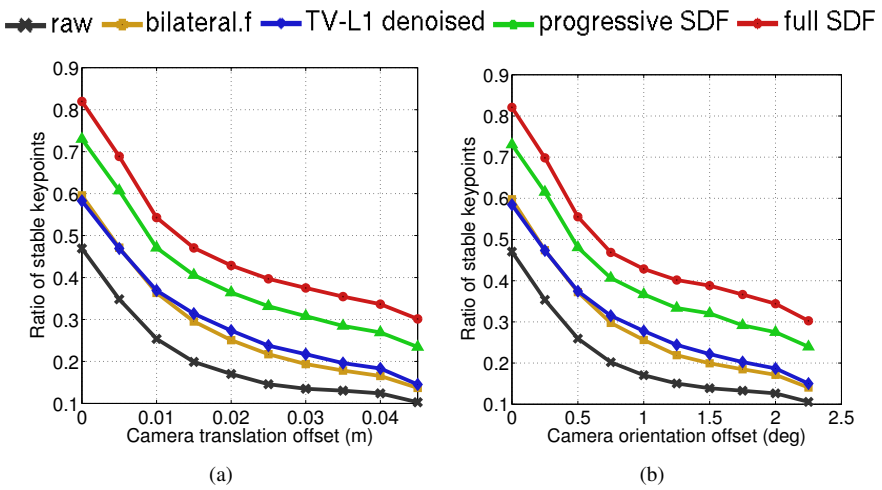


Fig. 3. NARF keypoint detection, for increasing baselines in camera translation and rotation

In order to provide a reliable match between two keypoint locations, their feature vector representations need to be sufficiently similar. In addition, feature vectors extracted at different locations need to be sufficiently different, in order to provide a clear separation between matching and non-matching points. The similarity between two feature vectors v_i and v_j is calculated using a parametric distance $d(v_i, v_j) = ((v_i - v_j)^T S(v_i - v_j))^{0.5}$, where S is a measure matrix that weights the importance of each dimension of the feature vector. Depending on the application scenario, different methods can be used to learn the matrix S and select informative dimensions. In this work we take the standard baseline approach of using an identity S matrix, resulting in a Euclidean distance measure d_e.

Identifying matching feature vectors can be achieved by simply setting a threshold on the maximum allowed Euclidean distance, but as suggested by previous works [11], this approach is suboptimal. Much better performance can be achieved by using a relative feature distance. Given two sets of feature vectors $V_i = \{v_i^p\}_{p=1...|V_i|}$ and $V_j = \{v_j^s\}_{s=1...|V_j|}$, the relative distance d_w between a vector v_i^p and it's closest neighbour $v_j^{s1} \in V_j$ is calculated as:

$$d_w(v_i^p, v_j^{s1}) = \frac{d_e(v_i^p, v_j^{s1})}{d_e(v_i^p, v_j^{s2})}, \tag{4}$$

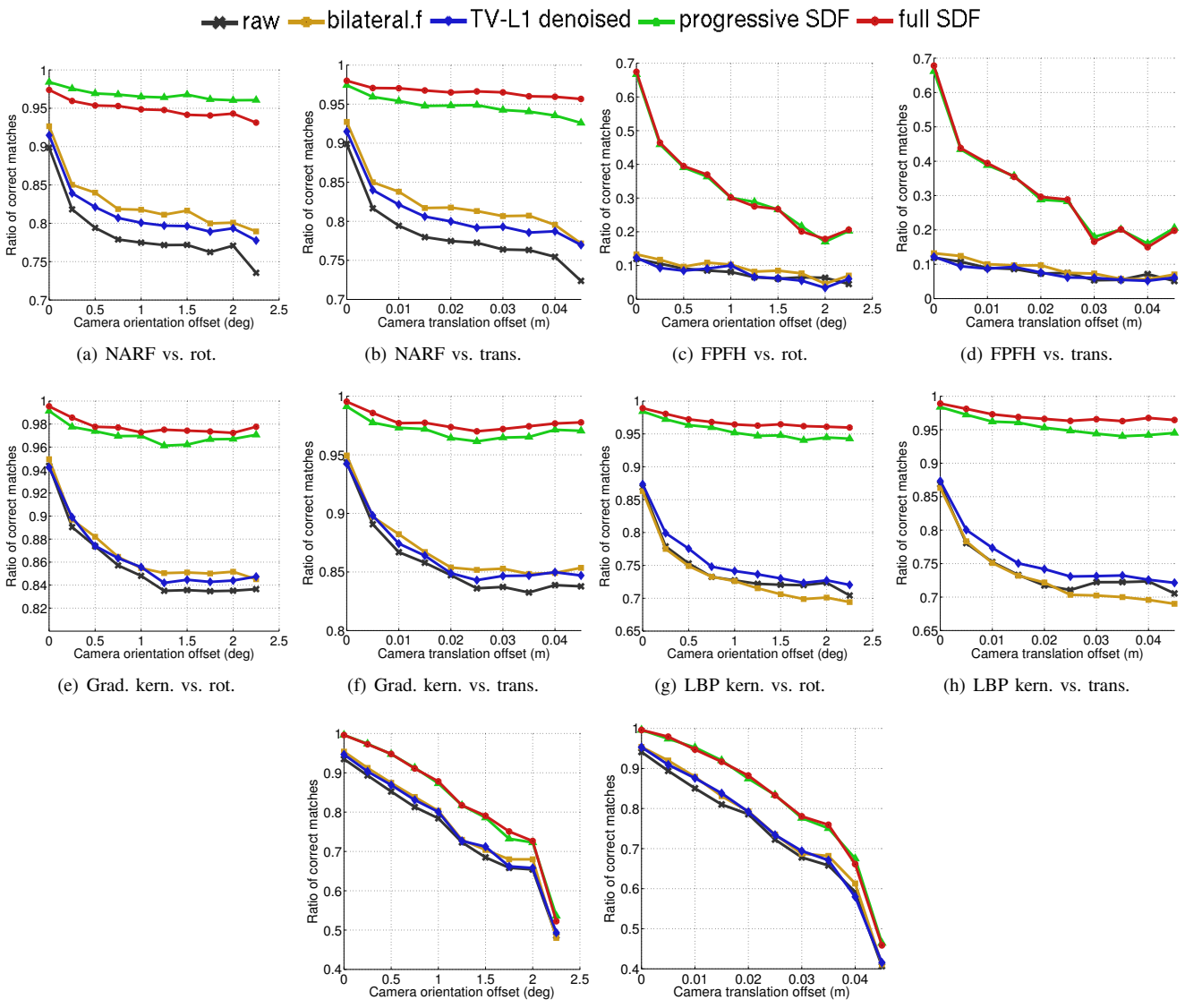
where v_j^{s2} is the second closest feature vector from V_j. The relative distance has several desirable properties — it is normalized in the range between 0 and 1, and more importantly it attains low values only if $d_e(v_i^p, v_j^{s1}) \ll d_e(v_i^p, v_j^{s2})$, i.e. only if v_i^p is much closer to its match than to any other feature vector in V_j.

Using the relative distance d_w, we can identify the matches between two sets of feature vectors by applying a distance threshold t_f. Before describing the details of the evaluation procedure however, we need to extract feature vectors from the input sequence of depth images. In a typical system, the keypoints detected in the previous step will be used for determining the locations at which we extract feature vectors. In order to obtain results independent of the quality of the keypoint detector, we need to extract vectors at precisely the same physical locations over the depth image sequence. Therefore, we manually choose and track a set of informative keypoint locations and thereby decouple the keypoint detection and feature extraction evaluations. An example view from the manual keypoint definition tool is shown in Figure 1(f). The black squares represent user defined keypoint locations, which are tracked throughout the depth image sequence and reprojected in each frame, with field of view and occlusion checks. In this manner, we also obtain reliable ground truth matching data — every manually selected keypoint and all feature vectors extracted at the same physical location are globally identified.

Knowing the ground truth association between features extracted at different frames, we proceed similarly to [11]. For any two depth images D_i and D_j we calculate the numbers of correctly and wrongly matched features, depending on the feature association distance threshold t_f. By varying t_f in the range $[0, 1]$, we obtain different values for the correctly (true positive) and wrongly (false positive) associated features. In [11] the cut-off threshold is manually set to a single value for all evaluated feature extraction techniques, which can introduce some bias in the subsequent results. We use instead a standard approach for binary classifier tuning and set the threshold to a value that achieves equal precision and recall values. Essentially, at this value the number of wrongly matched features is roughly equal to the number of matches that are not reported. We calculate the percentage of correctly detected matching features, over all ground truth matches between the two frames. This value is then accumulated into two histograms, over the translation and rotation difference between the two frames. In the next section, we report our results for keypoint stability and correct match rates of different feature detectors and descriptors.

V. Results

The results for NARF keypoint stability over translation and rotation viewpoint changes are shown in Fig. 3(a) and 3(b). The probability of reliably reobserved keypoints from different viewpoints increases by roughly twenty percent when the raw data is filtered using the bilateral or TV-



(a) NARF vs. rot. (b) NARF vs. trans. (c) FPFH vs. rot. (d) FPFH vs. trans.

(e) Grad. kern. vs. rot. (f) Grad. kern. vs. trans. (g) LBP kern. vs. rot. (h) LBP kern. vs. trans.

(i) Spin Kern. vs. rot. (j) Spin Kern. vs. trans.

Fig. 4. Performances for matching the different types of descriptors. The performance, on the vertical axis, is measured as the ratio between correctly matched descriptors and the total number of possible correct matches. Each plot shows the performance of one descriptor, subject to increasing rotation or translation, on two different data sets. The individual curves in each plot relate to one of the four denoising strategies, and raw data.

L1 methods. The increase in performance when using the SDF denoised images accounts for roughly another 20%, compared to the two noise filters. These results strongly suggest that the benefit of using past observations for data denoising for keypoint stability is significant.

Results of the evaluation of the five types of feature descriptors chosen are shown in Figure 4 (please note the difference in scale of the ordinates). The ratio of correctly identified matches is measured as a function of both translation and orientation offset in the camera pose and shown on separate plots. The impact of reducing the noise, in terms of feature matching is vast, in some cases, such as for the FPFH descriptor leading to over four times more matches compared to the raw data. Across all the feature descriptors tested, the SDF denoised depth maps show by far the largest boost in performance. It is encouraging to see that there is little difference between the progressively denoised model, based on only past observations, and the model which incorporates the entire data-set at once. Interestingly, the bilateral filter improves the matching rate on NARF features, but has an adverse effect on the matching of LBP kernel features, whereas the TV-L1 denoising yields a consistent improvement for matching with the LBP descriptor, but performs slightly worse than the bilateral filter in other cases. When using gradient kernel features neither of the single-frame denoising methods produce noticeable improvements over the raw data.

Beyond quantifying the improvement that denoising incurs in descriptor matching, we can also make a brief comparison between the different descriptors for this particular application. We note, for instance that the rate of matching for FPFH is comparatively low, possibly because it is designed to work with unstructured point cloud data. The Gradient and LBP kernel features show the best overall performance, on both raw and denoised data. The NARF features, deceivingly simple as they may be to compute, show a remarkably good

performance. The spin kernel appears to be the most sensitive to viewpoint variation, though it has among the highest matching rates for short baselines.

VI. Discussion and Future Work

Keypoint detection and feature descriptor matching remain as two cornerstone components of many current algorithms for object detection, tracking, mapping and localization and are therefore highly relevant in the field of robotics. In this work we have presented a methodology for evaluating shape based feature descriptors and provided an empirical analysis on the effects of sensor noise with regards to keypoint stability and robustness of feature descriptor matching. We have demonstrated the benefit of using dense mapping representations to obtain denoised sensor data and compared to state of the art single-frame noise filtering approaches. The proposed noise removal enabled by the dense map representation results in major improvements in matching across all classes of descriptors and keypoint detection stability.

The evaluation performed in this work clearly indicates the benefit of computing local shape descriptors on more consistent and less noisy depth images. In particular, the TSDF-based noise filtering evaluated testifies to the benefits of using previous measurements and viewpoint information. It would be interesting to compare the approach to other noise filtering techniques that use prior information. More in-depth evaluations, filter parameters and noise characteristics influence, as well as evaluations in the context of object recognition in online applications will be further explored as future research directions stemming from this article.

As further extensions to this work, we would like to apply dense map-based denoising methods beyond feature matching and evaluate the benefits of SDF denoising on object recognition. In this context, we can compare the benefits of repeated single-shot classification and candidate fusion, opposed to classification on SDF denoised geometrical models. Finally, we would like to investigate the possibility of defining feature descriptors directly in the dense 3D space, rather than on viewpoint-dependent depth images.

Acknowledgments

This work was partly funded by the 7th European Framework Project, RobLog — Cognitive Robot for Automation of Logistic Processes.

References

[1] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, 2011, pp. 127–136.

[2] D. R. Canelhas, T. Stoyanov, and A. J. Lilienthal, "SDF Tracker: A Parallel Algorithm for On-line Pose Estimation and Scene Reconstruction From Depth Images," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2013, to appear.

[3] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *IEEE Int. Conf. on Robotics and Automation, ICRA 2009*, 2009, pp. 3212 –3217.

[4] B. Steder, M. Ruhnke, S. Grzonka, and W. Burgard, "Place Recognition in 3D Scans Using a Combination of Bag of Words and Point Feature based Relative Pose Estimation," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2011.

[5] M. Magnusson, H. Andreasson, A. Nchter, and A. J. Lilienthal, "Automatic appearance-based loop detection from 3D laser data using the normal distributions transform," *Journal of Field Robotics*, vol. 26, no. 11–12, pp. 892–914, Nov. 2009.

[6] L. Bo, X. Ren, and D. Fox, "Depth Kernel Descriptors for Object Recognition," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*. IEEE, 2011, pp. 821–826.

[7] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D Recognition and Pose using the Viewpoint Feature Histogram," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2010, pp. 2155–2162.

[8] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.

[9] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[10] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "NARF: 3D Range Image Features for Object Recognition," in *Workshop on Realistic Perception in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[11] P. Moreels and P. Perona, "Evaluation of Features Detectors and Descriptors based on 3D Objects," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, 2007.

[12] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A Comparison of Affine Region Detectors," *International journal of computer vision*, vol. 65, no. 1, pp. 43–72, 2005.

[13] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking," *International journal of computer vision*, vol. 94, no. 3, pp. 335–360, 2011.

[14] G. Arbeiter, S. Fuchs, R. Bormann, J. Fischer, and A. Verl, "Evaluation of 3d feature descriptors for classification of surface geometries in point clouds," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*. IEEE, 2012, pp. 1644–1650.

[15] M. Madry, C. H. Ek, R. Detry, K. Hang, and D. Kragic, "Improving generalization for 3d object categorization with global structure histograms," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*. IEEE, 2012, pp. 1379–1386.

[16] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 80–91, 2012.

[17] S. Paris, P. Kornprobst, and J. Tumblin, *Bilateral filtering: Theory and applications*. Now Publishers Inc, 2009, vol. 1.

[18] M. W. Jones, J. A. Baerentzen, and M. Sramek, "3D Distance Fields: A Survey of Techniques and Applications," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 4, pp. 581–599, 2006.

[19] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 303–312.

[20] J. C. Hart, "Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces," *The Visual Computer*, vol. 12, no. 10, pp. 527–545, 1996.

[21] A. W. Fitzgibbon, "Robust Registration of 2D and 3D Point Sets," *Image and Vision Computing*, vol. 21, no. 13, pp. 1145–1153, 2003.

[22] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4.

[23] W. Echelmeyer, A. Kirchheim, A. J. Lilienthal, H. Akbiyik, and M. Bonini, "Performance Indicators for Robotics Systems in Logistics Applications," in *IROS Workshop on Metrics and Methodologies for Autonomous Robot Teams in Logistics (MMARTLOG)*, 2011.