TOWARDS LOCATION RECOGNITION USING RANGE IMAGES

A. Al-Nuaimi[1]*, R. Huitl*[1]*, S. Taifour*[1]*, S. Sarin*[2]*, X. Song*[2]*, Y. X. Gu*[2]*, E. Steinbach*[1]*, M. Fahrmair*[2]

[1]Institute for Media Technology, TUM, Munich, Germany {anas.alnuaimi,huitl,sinan.taifour,eckehard.steinbach@tum.de}
[2]Docomo Eurolabs, Munich, Germany {sarin,y.gu,song,fahrmair@docomolab-euro.com}

ABSTRACT

Retrieving the location of a mobile device by matching a query image to a database of geo-tagged imagery is one popular application of *content-based image retrieval* (CBIR). Standard CBIR-based approaches exploit appearance features of the environment for the matching process. Many locations, however, are characterized by distinct structural (geometric) features. We investigate whether a standard appearance-based CBIR pipeline can be adapted to perform location retrieval using a range image-based representation of the environment. The contributions are three-fold: We design a rigorous experimental setup using an extensive and challenging indoor dataset. Secondly, we compare the state-of-the-art feature algorithm specifically designed for range images, the *Normal Aligned Radial Feature* (NARF) [1], against some of the most established appearance-based features. Thirdly, we combine the high key point detection rate of NARF, with the robustness of the Speeded-Up Robust Feature for range-image based location recognition. This detector-descriptor combination, which we coin NURF, leads to 15% improvement in absolute location recognition performance compared to simple NARF in our experimental setup.

Index Terms— NARF, SURF, CBIR, Range Image

1. INTRODUCTION

Content-based image retrieval (CBIR) is concerned with developing methods for robust matching of similarly looking images. One of its applications is location recognition (also called place recognition or location retrieval in this paper). The basic idea is that a database of photos captured at different locations is collected. Every image is associated with a *geo-tag*, meta data representing the location in which it was captured. Once a photo is captured at any location, it is queried against the database. The location(s) of the database photo(s) identified as being most similar define(s) the location of the query image. Place recognition systems based on CBIR have been presented in [2, 3, 4, 5]. Huge improvements have been achieved in this area due to the development of robust texture-based feature detectors and descriptors as well as the effective use of the concept of *Bag-of-Features* (BoF) [6], in particular *Vocabulary Trees* (VT) [7], for efficient matching.

Today's CBIR-based approaches for location recognition typically use geo-tagged photos for their databases. The underlying assumption is that the visual appearance of the scene is distinctive for the location. However, some locations are better described by their geometry. In recent years, algorithms and sensors have been developed that facilitate retrieving information about the structure of the environment. Not only have low-cost 3D sensors, such as the Microsoft *Kinect* sensor, been developed, but also algorithms that can reconstruct entire city neighborhoods in 3D as demonstrated by Agarwal et al. [8]. Also, 3D lasers are sometimes used to obtain accurate scans of the surrounding. With these advances, attempts to perform location recognition using the methods of 3D shape matching were made as demonstrated in [9].

The area of 3D shape matching is drawing increasing attention mainly from the object recognition community [10]. Performing location recognition using 3D shape matching has not been extensively studied. Therefore, it is interesting to investigate whether it is possible to exploit the leaps in the area of CBIR for the purpose of location recognition using 3D data. For that, a representation of the 3D data in the form of images is required. *Range images*, serve this particular purpose. The intensity value of each pixel in the range image is directly related to the depth of the scene along the ray extending from the camera center through the pixel. Accordingly, researchers have started considering using range images for structure-based localization as demonstrated in the work of Steder et al. [11] who present a novel detector and descriptor coined NARF. NARF is not compared to standard appearance-based features in [11]. Since it operates on range images, it would be interesting to perform such a comparison. Particularly, we want to find out whether standard feature algorithms could perform better than NARF for certain types of range queries and propose improvements accordingly.

Focusing on the application of location retrieval using range images, we build an evaluation setup using an extensive indoor dataset. We then compare different feature algorithms with NARF in different view change scenarios. As a third contribution, we propose a combination of the high key point detection rate of NARF's detector with the SURF descriptor's robustness to create what we call the *NURF* feature.

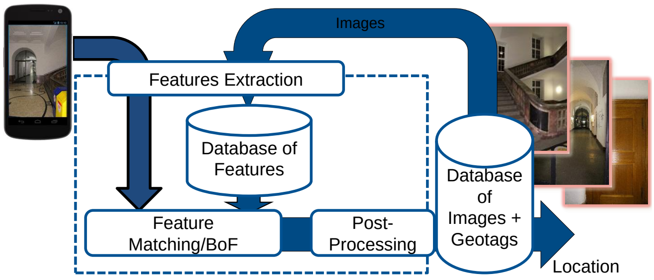


Fig. 1: Standard CBIR location recognition system

2. BACKGROUND AND RELATED WORKS

A standard CBIR pipeline for location retrieval using digital photographs (photos) is depicted in Figure 1. It consists of three main components:

1) A database of geo-tagged images: As in the work of Steder et al. [11] we replace the photometric images with range images. However we use a much more extensive dataset covering a track length of $1169m$.

2) A feature detection and description unit: Image similarity is calculated by first extracting interest points in the image, so-called *key points* using a *feature detector*. In a second step, the image patches around the key points are described in a transformation invariant manner using a *feature descriptor*. The Scale-Invariant Feature Transform (SIFT) [12] and the Speeded-Up Robust Features (SURF) [13] are widely acknowledged in the community for their high performance. Both define their own detector and descriptor. The Maximally Stable Extremal Regions (MSER) [14] key point detector enjoys wide-range popularity due to its elegant way of dealing with affine distortions of image patches. The Hessian-Affine [15] detector is yet another successful affine-invariant keypoint detector that is used as a baseline detector. The reader is referred to the comparison of Tuytelaars et al. [16] for more details. Throughout the paper we refer to a detector-descriptor combination as *feature extractor*, *feature algorithm* or simply *feature*. While some researchers have attempted using such features for retrieval in range images [17], others have attempted developing feature algorithms tailored to range images such as Steder et al.'s NARF [18, 1] and Tipaldi and Arras' FLIRT feature [19]. NARF detects object boundaries by looking for sharp range changes and extracts a descriptor by determining the surface normal at the edge and calculating a descriptor that captures the gradient change under 36 beams of a center-placed star shape.

3) A feature matching unit: The works of Sivic et al. [6] and Nister et al. [7] have demonstrated the advantage of the vocabulary tree concept (VT) for rapid feature-based image matching compared to brute force feature matching. The underlying idea is to quantize the feature space into a visual vocabulary and then to describe each image by a histogram of visual words. The most similar database image is the one with the most similar BoF histogram. Query augmentation methods and intelligent post-processing of the retrieval result have also been developed. The work of Arandjelović [20] provides a comprehensive summary of the important improvements introduced to BoF-enabled CBIR and CBIR in general. A number of VT implementations exist. We use our BoF engine which is described in detail in [3].

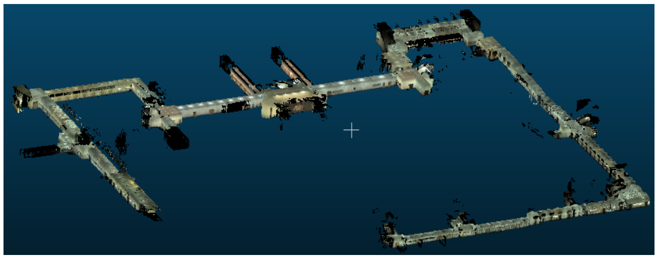


Fig. 2: Visualization of the indoor point cloud dataset

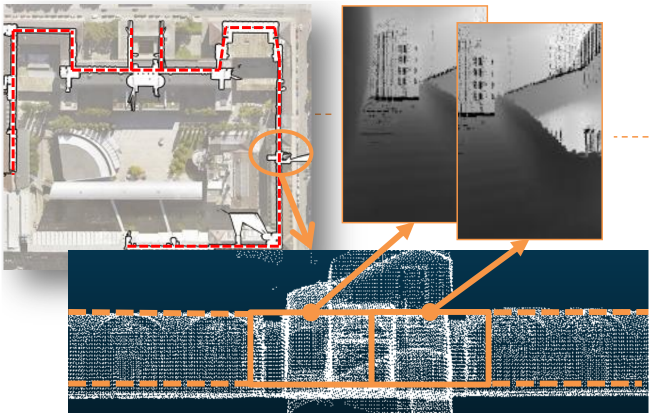
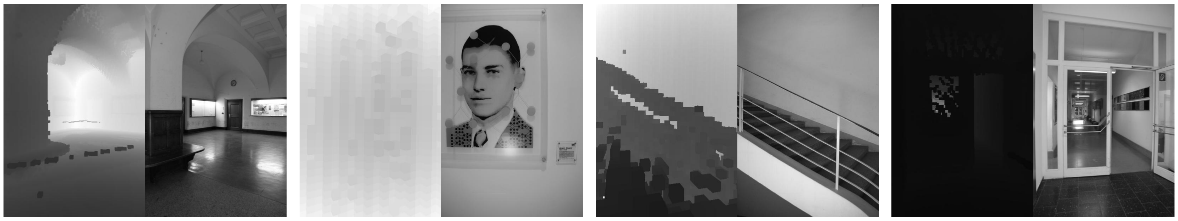


Fig. 3: Generating range images from the point cloud

Performing place recognition using range images has been attempted by Steder et al. [11] and Tipaldi and Arras [19]. The former is particularly interesting because of the results demonstrated for the proposed NARF feature as well as the use of BoF. Moreover, the authors have implemented their algorithm as part of the *Point Cloud Library* (PCL). Unfortunately, the results obtained with NARF are not compared to those obtainable with standard CBIR features. The focus of this paper is to quantify the gains of NARF, if any, when compared to standard CBIR features applied on range images and to understand whether standard CBIR features can be exploited to get better location retrieval performance.

3. DATASET AND RANGE IMAGE RENDERING

Starting with a *point cloud* dataset the range images have to be rendered at specific locations. In this section we shall explain the used point cloud dataset, the rendering method and the poses at which database images and query images are rendered.



(a) Location 2247 (b) Location 7552 (c) Location 2256 (d) Location 39

Fig. 4: Examples of generated range images (left) with the respective photo (right) from the same viewpoint

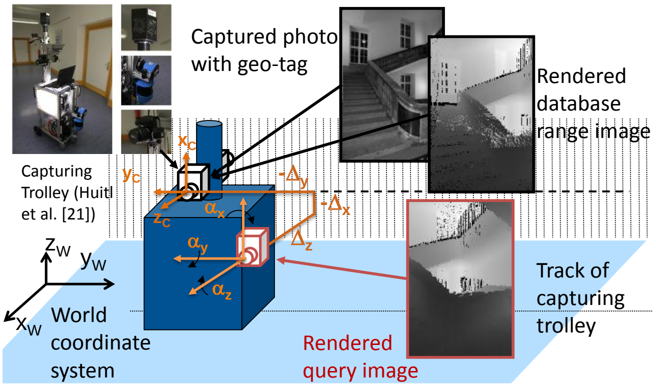


Fig. 5: Database and query images rendering

We use the indoor point cloud dataset captured by Huitl et al. [21] (downloadable at `www.navvis.de/dataset/`). The indoor use case is chosen for two reasons: Firstly, it represents a GPS-devoid localization scenario. Secondly, the structure is typically composed of walls which do not carry many structurally characteristic shapes and hence provide a rather challenging scenario for our application. We use the "1st floor" dataset (visualized in Figure 2) which covers a track length of 1169m (red line in Figure 3). The indoor dataset offered by the NARF authors[1], on the other hand, covers a very limited area (a lab room). The 1st-floor dataset from [21] has around $60M$ points with a resolution of about $1pt/cm$.

The principle idea in our test is to replace the photometric images shown in Figure 1 with range images rendered at different locations from the point cloud as depicted in Figure 3. Ray casting is used to determine the distance between the camera and the scene points. The point cloud is represented using an octree of voxel size $(0.05m)^3$ which facilitates fast ray casting even in huge datasets. For each pixel of the image, a ray is constructed with the camera center as origin. The direction of the ray is determined with the help of a pinhole camera model. The range value for the pixel is then determined by finding the first intersection with an occupied voxel of the octree. The range values are finally normalized to the range $[0, 1]$. Since some range values erroneously take on very large values, we use 40m as the maximum range value in case the maximum rendered range value exceeds this threshold. Figure 4a shows an example with distinct geometric structure. The example in Figure 4b shows how certain locations are better described by their appearance than their geometry. In Figure 4c it can be seen that the rendering process can be of low quality for certain kinds of geometry. The artifacts are due to the voxelization process associated with creating the octree. Sometimes, rendering completely fails as shown in Figure 4d. The latter typically happens when the surface point density is low and the ray extends between two voxels to hit a far away voxel. The normalization by the maximum distance then down-weights all other pixels resulting in a low contrast image.

The dataset captured in [21] provides photos along side the point cloud data (see right image in each of the four subfigures in Figure 4). The dataset is recorded using a trolley on which two DSLR cameras are side-mounted (see Figure 5). The cameras are triggered roughly every $0.75m$ of covered track length. Accordingly, there is a left-right photo pair at 1573 locations in the covered track (3146 photos). Range images are rendered for the database with the exact poses of the photos. For each rendered database image a number of respective query images is rendered. Each query image is obtained by taking a fixed displacement from the respective database image location described by the triplet $(\Delta x, \Delta y, \Delta z)$ along the local camera coordinate system as shown in Figure 5. Also, a view direction change is incorporated described by the Euler angles rotations $(\alpha_z, \alpha_y, \alpha_x)$ around the z-, y- and x- camera coordinate axes in this particular order. All query images obtained by a particular *relative pose* are stored in one *query set*. These query sets facilitate a systematic performance evaluation since there is a 1:1 correspondence between the database image set and any query image set. In total, we generate 11 query sets representing different view change scenarios shown in Table 1. The displacements in the first three sets result in the scene appearing side-shifted or upward/downward-shifted in the image compared to the database image. The scene is captured from a larger distance in query sets 4 and 5 and hence objects in the scene appear smaller and more is seen of the environment. The dis-

[1] `http://ais.informatik.uni-freiburg.de/projects/datasets/quadrotor079/`

Table 1: Query sets. Displacements in *meter*, angles in *degree*

#	$(\Delta x, \Delta y, \Delta z)(\alpha_z, \alpha_y, \alpha_x)$	scenario (visual effect)
1	$(0.5, 0.0, 0.0)(0, 0, 0)$	shift up
2	$(0.0, -0.5, 0.0)(0, 0, 0)$	shift sideways
3	$(0.0, 0.5, 0.0)(0, 0, 0)$	shift sideways
4	$(0.0, 0.0, -0.5)(0, 0, 0)$	shift behind (scaling)
5	$(0, 0, -1.0)(0, 0, 0)$	shift behind (scaling)
6	$(0.0, 0.0, 0.0)(20, 0, 0)$	in-plane rotation
7	$(0.0, 0.0, 0.0)(0, 20, 0)$	up-down ofpl rot.
8	$(0.0, 0.0, 0.0)(0, 0, 30)$	right-left ofpl rot.
9	$(0.0, 0.0, 0.0)(0, 0, 20)$	right-left ofpl rot.
10	$(0.0, 0.0, 0.0)(0, 0, -20)$	right-left ofpl rot.
11	$(0.0, 0.0, -0.5)(0, 0, 10)$	scaling + ofpl rot.

Table 2: Parameter configurations

	parameter	conf.1	conf.2	conf.3	conf.4
MSER	_max_area	0.05	0.1	0.1	0.25
hesaff	-thres	335	670	1340	2680
SURF	-thres	29750	59500	119000	238000
NARF	support_size	0.2	1	0.01	0.05

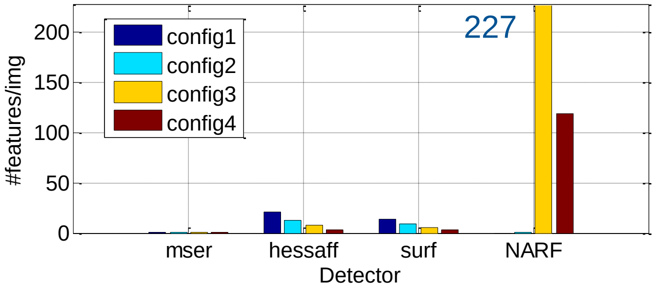


Fig. 6: Average number of features per image

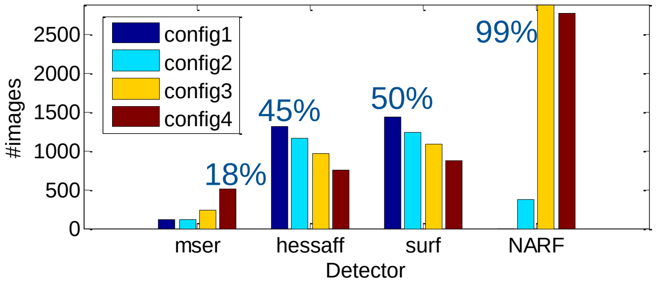


Fig. 7: Number of images with at least one feature

tortion in the sixth query set results in the scene appearing rotated along the camera axis of the database image while the rotation in sets 7-10 results in out-of-plane rotation (ofpl rot.) of the query camera w.r.t. the respective database camera's imaging plane. Finally, set 11 simulates a case where the query is captured from a farther distance from the wall and with an out-of-plane rotation of $10°$. The database image set can, according to this introduced convention, be identified as $((0.0, 0.0, 0.0)(0, 0, 0))$. Due to the rendering process some images have "holes" inside them (unrendered areas). In this experiment a subset of the 3146 image indices is picked as follows: The images carrying the same index in all sets as well as the database set have to have $\leq 10\%$ holes each in order for the index (location) to be added to the subset. The holes are filled using a scan line approach to hinder detectors from firing on these "blind" spots as they typically exhibit sharp intensity changes. Due to this "filtering" operation each set has 2864 images.

4. DETECTOR AND DESCRIPTOR COMPARISON

In a first test standard CBIR detectors are evaluated to check if enough features can be found in the range images. The database image set is used in this test to compare three highly popular detector types (MSER, SURF, Hessian-Affine (hesaff)) against the NARF detector in terms of number of key points found. The reference implementations of SURF and hesaff are used as provided by their authors. The MSER implementation found in OpenCV is used. NARF has been implemented by the authors in the Point Cloud Library (PCL) [22]. For each chosen detector type specific detector parameters are varied choosing four different configurations. The aim is to see whether results can be improved beyond those delivered by the default parameters. The different configurations are summed up in Table 2. The parameters have been named exactly as found in their respective implementation (or binary argument list).

Typically, the more features images have, the more robust is the retrieval. Hence the average number of key points per range image is measured. Figure 6 shows that NARF detects far more key points per image than any of the standard CBIR detectors in any configuration. In fact, the number of detected key points is in the same order of magnitude as that of typical appearance-based CBIR. By investigating these results in detail, it is found that standard CBIR detectors not always find features in a range image in the first place. Figure 7 shows that for MSER no more than 18% of the images have at least one feature. This is very different from standard appearance-based CBIR where typically the number of images without features is very small. The best standard CBIR detector is SURF which - using configuration 1 - detects at least one feature in 50% of all images. NARF's detector, however, manages to find key points in almost every image using configuration 3.

The detector analysis shows that the NARF detector outperforms the MSER, Hessian-Affine and SURF detectors in terms of number of detected key points when applied on range images. However, for CBIR applications it is not enough to detect many key points but also to describe the image regions in an invariant and distinctive manner. This is determined by the descriptor design. Location retrieval on each of the 11

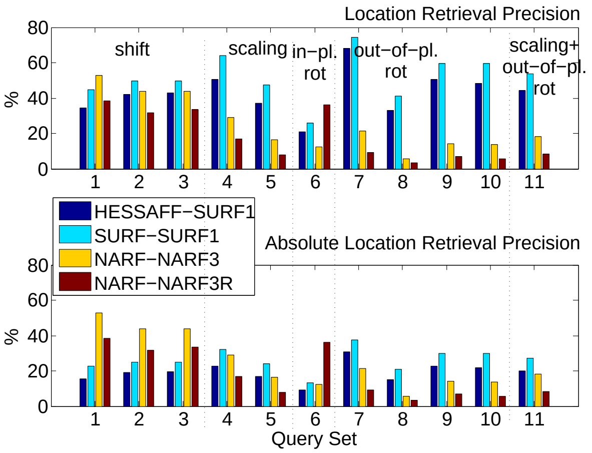


Fig. 8: (Absolute) Location retrieval precision

query sets using our BoF system is performed. The overall retrieval precision is used as the main performance metric to compare overall detector-descriptor performance. This measure for precision represents a rather strict evaluation since only the highest ranked candidate is considered. If the index of this candidate matches that of the query image within a tolerance of ± 1 (equivalent to being the image from one of the two neighboring locations), then it is marked as a successful location retrieval. The precision is the fraction of successful queries out of all 2846 queries.

Features are extracted using the Hessian-Affine and SURF detectors both with config 1 combined with the *upright*-SURF descriptor (HESSAFF-SURF1, SURF-SURF1). Also NARF descriptors with the NARF detector in config 3 (NARF-NARF3) are computed. Additionally, a rotation invariant version of NARF-NARF3 (NARF-NARF3R) is used. The upper part of Figure 8 compares these four different detector-descriptor combinations in terms of location retrieval precision. It can be seen that HESSAFF-SURF1 as well as SURF-SURF1 exhibit a fairly stable performance in excess of 40% precision irrespective of the query set used except in the case of in-plane rotation (query set 6), which is due to the use of upright SURF, and severe out-of-plane rotation (query set 8). Moreover, SURF-SURF1 always outperforms HESSAFF-SURF1. NARF-NARF3R is consistently worse than NARF-NARF3 except in the case of query set 6. This is expected, since endowing descriptors with in-plane rotation invariance typically comes at the expense of descriptor distinctiveness. The curves also show that both NARF-NARF configurations severely degrade in precision when considering scaling and/or out-of-plane rotation scenarios. While the former is justifiable since the NARF detector does not incorporate automatic scale selection [18, 1], the latter indicates that the descriptor invariance of SURF is substantially better.

The results in the upper part of Figure 8 only show precision in the cases where retrieval can be performed (when database images have features). Comparing the actual location retrieval performance requires normalizing the results by the fraction of the database images for which each image has at least one feature (multiplying the bars for HESSAFF-SURF1 and SURF-SURF1 by 45% and 50% respectively). The lower part of Figure 8 shows that the NARF feature algorithm gains in absolute location retrieval precision since it detects key points in almost every range image. This normalization, however, does not compensate for its shortcomings in the scaling and out-of-plane rotation scenarios.

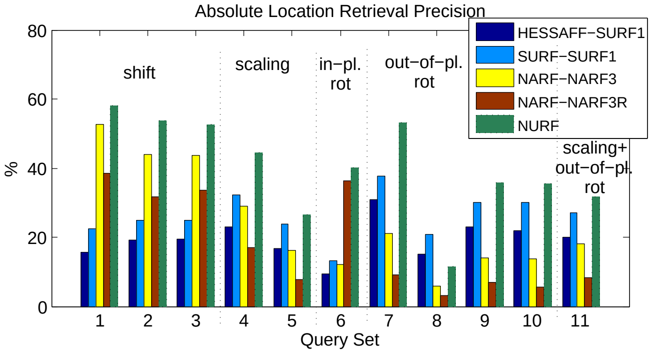


Fig. 9: NURF location retrieval precision

5. FROM NARF TO NURF

The results in Section 4 show that the NARF detector substantially outperforms standard CBIR detectors in terms of number of key points detected in range images. The results of the previous section, however, also show that the SURF descriptor has much higher robustness towards different view-point changes between query and database range images which motivates us to consider combining the high detectability of NARF with the robust descriptivity of SURF to create the NURF detector-descriptor.

Combining the NARF detector with SURF is not straightforward since the NARF detector, as opposed to most standard CBIR detectors, does not have a scale in the image space [1]. NARF's descriptor support is chosen based on the size of the geometry of the scene and specifies a 3D sphere around the key point. This has to be projected into the image plane. This can be done by knowing the intrinsic camera parameters as well as the 3D location of the key point. The radius of the support in the image plane is calculated as:

$$r = R\frac{f}{z}c \qquad (1)$$

where r is the support radius in the image domain (pixels) R is the used feature support radius in 3D space [1], f is the focal length of the virtual camera and z is the depth of the interest point. c is a correction factor accounting for implementation details (the NARF implementation in PCL includes scaling parameters used when computing a point cloud out of the range image). The calculated radius is passed to the SURF descriptor which computes the feature vector. The performance of NURF is benchmarked against HESSAFF-SURF1, SURF-SURF1 and the two previously presented NARF feature configurations. The results in Figure 9 clearly show that the proposed feature algorithm outperforms all other algorithms in all but one scenario (query set 8). This scenario represents a very strong view direction change of 30° which is known in the literature to be the performance limit of the SURF descriptor [13]. Averaging over all query sets, NURF outperforms NARF-NARF3 by more than 15% in absolute location recognition performance.

It remains to be said that this approach for localization is limited by the quality of obtainable range data. Obtaining a real range query on a portable device is yet another challenge to be overcome.

6. CONCLUSION AND OUTLOOK

In this paper we investigate the suitability of standard appearance-based CBIR feature algorithms for the application of location retrieval using range image datasets. We found out that the SURF-detector, MSER-detector and Hessian Affine-detector deliver substantially less features per range image than the NARF detector, and that in many images they are unable to find any key points at all. In a second investigation we found out that for range images the SURF descriptor exhibits higher robustness towards different view point and direction changes than the NARF descriptor. In a final step we integrated the NARF detector with the SURF descriptor, creating the NURF feature algorithm, which demonstrates superior performance and achieves a 15% improvement in absolute location recognition performance compared to pure NARF in our experimental setup. In the future we plan to investigate joint appearance/structure location recognition by exploiting the developed range image-based framework and the standard CBIR pipeline.

7. REFERENCES

[1] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, “NARF: 3D range image features for object recognition,” in *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ IROS 2010*, 2010.

[2] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys, “Leveraging 3d city models for rotation invariant place-of-interest recognition,” *IJCV*, vol. 96, no. 3, pp. 315–334, Feb. 2012.

[3] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach, “Mobile visual location recognition,” *IEEE SPM, Special Issue on Mobile Media Search*, vol. Vol. 28, No. 4, pp. 77-89, 2011.

[4] G. Schroth, A. Al-Nuaimi, R. Huitl, F. Schweiger, and E. Steinbach, “Rapid image retrieval for mobile location recognition,” in *IEEE ICASSP*, May 2011.

[5] D. M. Chen, G. Baatz, K. Koser, S. S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, Xin Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk, “City-scale landmark identification on mobile devices,” in *IEEE CVPR*, Washington, DC, USA, 2011, pp. 737–744.

[6] J. Sivic and A. Zisserman, “Video google: a text retrieval approach to object matching in videos,” in *IEEE CVPR*, 2003, pp. 1470 –1477 vol.2.

[7] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in *IEEE CVPR*, 2006.

[8] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, “Building rome in a day,” in *IEEE CVPR*, 2009, pp. 72–79.

[9] M. Bosse and R. Zlot, “Place recognition using regional point descriptors for 3d mapping,” in *FSR*, vol. 62 of *Springer Tracts in Advanced Robotics*, pp. 195–204. Springer Berlin Heidelberg, 2010.

[10] J. W. Tangelder and R. C. Veltkamp, “A survey of content based 3d shape retrieval methods,” *Multimedia Tools Appl.*, vol. 39, no. 3, pp. 441–471, Sept. 2008.

[11] D. 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 *IROS*, 2011, pp. 1249–1255.

[12] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[13] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *In ECCV*, 2006, pp. 404–417.

[14] J. Matas, O. Chum, U. Martin, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions,” in *BMVC*, 2002, vol. 1, pp. 384–393.

[15] K. Mikolajczyk and C. Schmid, “An affine invariant interest point detector,” in *ECCV*, London, UK, UK, 2002, pp. 128–142, Springer-Verlag.

[16] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: a survey,” *Found. Trends. Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280, July 2008.

[17] K. Sfikas, I. Pratikakis, and T. Theoharis, “3d object retrieval via range image queries based on sift descriptors on panoramic views,” in *5th Eurographics conference on 3D Object Retrieval*, Aire-la-Ville, Switzerland, Switzerland, 2012, pp. 9–15, Eurographics Association.

[18] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, “Point feature extraction on 3d range scans taking into account object boundaries,” in *IEEE ICRA*, 2011, pp. 2601 –2608.

[19] G. D. Tipaldi and K. O. Arras, “Flirt - interest regions for 2d range data,” in *IEEE ICRA*, 2010, pp. 3616–3622.

[20] R. Arandjelović and A. Zisserman, “Three things everyone should know to improve object retrieval,” in *IEEE CVPR*, 2012.

[21] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, “TUMindoor: an extensive image and point cloud dataset for visual indoor localization and mapping,” in *IEEE ICIP*, 2012.

[22] R.B. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *IEEE ICRA*, 2011, pp. 1–4.