Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries

Bastian Steder Radu Bogdan Rusu Kurt Konolige Wolfram Burgard

***Abstract*— In this paper we address the topic of feature extraction in 3D point cloud data for object recognition and pose identification. We present a novel interest keypoint extraction method that operates on range images generated from arbitrary 3D point clouds, which explicitly considers the borders of the objects identified by transitions from foreground to background. We furthermore present a feature descriptor that takes the same information into account. We have implemented our approach and present rigorous experiments in which we analyze the individual components with respect to their repeatability and matching capabilities and evaluate the usefulness for point feature based object detection methods.**

I. Introduction

In object recognition or mapping applications, the ability to find similar parts in different sets of sensor readings is a highly relevant problem. A popular method is to estimate *features* that best describe a chunk of data in a compressed representation and that can be used to efficiently perform comparisons between different data regions. In 2D or 3D perception, such features are usually *local* around a point in the sense that for a given point in the scene its vicinity is used to determine the corresponding feature. The entire task is typically subdivided into two subtasks, namely the identification of appropriate points, often referred to as *interest point*s or *key point*s, and the way in which the information in the vicinity of that point is encoded in a *descriptor* or *description vector*.

Important advantages of interest points are that they substantially reduce the search space and computation time required for finding correspondences between two scenes and that they furthermore focus the computation on areas that are more likely relevant for the matching process. There has been surprisingly little research for interest point extraction in raw 3D data in the past, compared to vision, where this is a well researched area. Most papers about 3D features target only the descriptor.

In this paper we focus on single range scans, as obtained with 3D laser range finders or stereo cameras, where the data is incomplete and dependent on a viewpoint. We chose range images as the way to represent the data since they reflect this situation and enable us to borrow ideas from the vision sector.

We present the normal aligned radial feature (NARF), a novel interest point extraction method together with a feature descriptor for points in 3D range data. The interest point extraction method has been designed with two specific goals in mind: i) the selected points are supposed to be in positions where the surface is stable (to ensure a robust estimation of the normal) and where there are sufficient changes in the immediate vicinity; ii) since we focus on partial views, we want to make use of object borders, meaning the outer shapes of objects seen from a certain perspective. The outer forms are often rather unique so that their explicit use in the interest point extraction and the descriptor calculation can be expected to make the overall process more robust. For this purpose, we also present a method to extract those borders. To the best of our knowledge, no existing method for feature extraction tackles all of these issues.

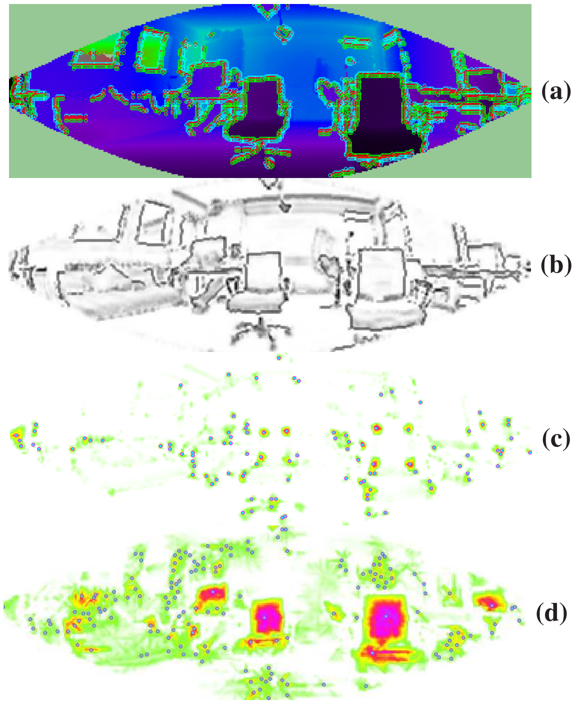

Fig. 1. The interest point extraction procedure. **(a)** Range image of an office scene with two chairs in the front with the extracted borders marked. **(b)** Surface change scores according to borders and principle curvature. **(c)** Interest values with marked interest points for a support size of 20cm. Note how the corners of the chairs are detected as interest points at this scale. **(d)** Interest values with marked interest points for a support size of 1m. Note how, compared to (c), the whole surface of the chair's backrests contain one interest point at this scale.

Figure 1 shows an outline of our interest point extraction procedure. The implementation of our method is available under an open-source license[1].

B. Steder and W. Burgard are with the Dept. of Computer Science of the University of Freiburg, Germany. {steder,burgard}@informatik.uni-freiburg.de; R. Rusu and K. Konolige are with Willow Garage Inc., USA. {rusu,konolige}@willowgarage.com

[1] The complete source code including instructions about running the experiments presented in this paper can be found at http://www.ros.org/wiki/Papers/ICRA2011_Steder

The paper is organized as follows. After discussing related work in Section II, we will introduce our border extraction method in Section III. We then will describe the interest point extraction in Section IV and the NARF-descriptor in Section V. We finally present experimental results in Sections VI and VII.

II. Related Work

Two of the most popular systems for extracting interest points and creating stable descriptors in the area of 2D computer vision are SIFT (Scale Invariant Feature Transform) [7] and SURF (Speeded Up Robust Features) [2]. The interest point detection and the descriptors are based on local gradients, and a unique orientation for the image patch is extracted to achieve rotational invariance. Our approach operates on 3D data instead of monocular camera images. Compared to cameras, 3D sensors provide depth information, and can be less sensitive to lighting conditions (e.g., laser sensors). In addition, scale information is directly available. A disadvantage, however, is that geometry alone is less expressive in terms of object uniqueness. While SIFT and SURF are not directly transferable to 3D scans, many of the general concepts, such as the usage of gradients and the extraction of a unique orientation, are useful there.

One of the most popular descriptors for 3D data is the *Spin-image* presented by Johnson [6], which is a 2D representation of the surface surrounding a 3D point and is computed for every point in the scene. In our previous work [14] we found that range value patches as features showed a better reliability in an object recognition system compared to spin images. The features we propose in this paper build on those range value patches and show an improved matching capability (see Section VII). Spin images also do not explicitly take empty space (e.g., beyond object borders) into account. For example, for a square plane the spin images for points in the center and the corners would be identical, while the feature described in this paper is able to discriminate between such points.

An object detection approach based on silhouettes extracted from range images is presented in [15]. The features proposed are based on a fast Eigen-CSS method and a supervised learning algorithm. This is similar to our work in the sense that the authors also try to make explicit use of border information. By restricting the system to borders only, however, valuable information regarding the structure of the objects is not considered. Additionally, the extraction of a single descriptor for the complete silhouette makes the system less robust to occlusions.

Huang et al. [5] developed a system for automatic reassembly of broken 3D solids. For this purpose the authors also extract boundaries on the 3D structures, in this case to detect sets of faces. The method detects cycles of edges based on surface curvature. The detected surface parts are then matched to find corresponding fragment parts. In our application the border detection finds changes from foreground to background and uses this outer shape in the interest point detection and matching, compared to dividing the 3D structure into faces that are then matched individually.

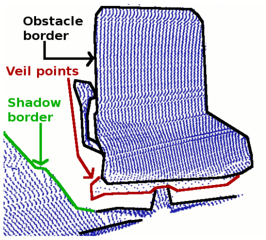

Fig. 2. Different kinds of border points.

Many approaches compute descriptors exhaustively in every data point or use simple sampling methods ([6], [3], [8]), thereby introducing an unnecessary overhead. In [4] the authors present an approach to global registration using Integral Volume Descriptors (IVD) estimated at certain interest points in the data. These interest points are extracted using a self similarity approach in the IVD space, meaning the descriptor of a point is compared to the descriptors of its neighbors to determine areas where there is a significant change. While this method for interest point extraction explicitly takes the descriptor into account, it becomes impractical for more complex descriptors, which are more expensive to extract. Unnikrishnan [16] presented an interest point extraction method with automatic scale detection in unorganized 3D point clouds. This approach, however, does not consider any view-point related information and does not attempt to place interest points in stable positions. In [12] we presented a method for selecting interest points based on their *persistence* in growing point neighborhoods. Given a PFH (Point Feature Histogram) space [11], multiple descriptors are estimated for several different radii, and only the ones similar between r_i and r_{i+1} are kept. The method described in this paper is by several orders of magnitude faster in the estimation of interest points.

In our previous work we also used interest point extraction methods known from the 2D computer vision literature, such as the Harris Detector or Difference of Gaussians, adapted for range images [14], [13]. Whereas these methods turned out to be robust, they have several shortcomings. For example, the estimated keypoints tend to lie directly on the borders of the objects or on other positions that have a significant change in structure. While these areas are indeed interesting parts of the scene, having the interest points directly there can lead to high inaccuracies in the descriptor calculation since these are typically unstable areas, e.g., regarding normal estimation. The goal of our work described here is to find points that are in the vicinity of significant changes and at the same time are on stable parts of the surface.

III. Border extraction

A. Motivation

One important requirement to our feature extraction procedure is the explicit handling of borders in the range data. Borders typically appear as non-continuous traversals from foreground to background. In this context there are mainly three different kinds of points that we are interested in detecting: *object borders*, which are the outermost visible points still belonging to an object, *shadow borders*, which are points in the background that adjoin occlusions, and *veil points*, which are interpolated points between the obstacle border and the shadow border. Veil points are a typical phenomenon in 3D range data obtained by lidars and treating them properly clearly improves matching and classification results. Figure 2 shows an example of the different types of (border) points described above.

B. Overview

There are different indicators that might be useful for the detection of those borders in a range image, like acute impact angles or changes of the normals. In our practical experiments, we found that the most significant indicator, which is also very robust against noise and changes in resolution, is a change in the distance between neighboring points. We will use this feature to classify borders according to the following steps. For every image point we look at its local neighborhood and

- employ a heuristic to find the typical 3D distance to neighboring points that are not across a border,
- use this information to calculate a score for how likely it is that this point is part of a border,
- identify the class to which this border point belongs, and
- perform non-maximum suppression to find the exact border position.

C. The Algorithm in Detail

Please note, that since we are using range images, every point has a 2D position (the position of the pixel in the image) and a 3D position (the measured position in the world coordinate frame). The same applies for distances between points. When we refer to 2D distances we mean the distance between the pixels in the image, whereas 3D distance refers to the Euclidean distance between the 3D points.

At first, we look at every image point and apply a heuristic to find out what 3D distance a certain point typically has to its 2D neighbors, that belong to the same surface. To detect this we use a concept similar to the idea of Bilateral Filtering [1]. For each point p_i in the range image we select all neighboring points $\{n_1, \cdots, n_{s^2}\}$ that lie in the square of size s with p_i in the middle. Then we calculate their 3D distances $\{d_0, \cdots, d_{s^2}\}$ to p_i and sort this set in increasing order to get $\{d'_0, \cdots, d'_{s^2}\}$. Assuming that at least a certain number M of the points lie on the same surface as p_i, we select $\delta = d'_M$ as a typical distance to p_i-s neighbors, that do not include points beyond a border. In our implementation we selected $s = 5$ and $M = \left(\frac{(s+1)}{2}\right)^2 = 9$, which would be the highest M to still get a correct value for a point lying on the tip of a right angle corner. In the next step we calculate four scores for every image point, describing the probability of having a border on the top, left, right, or bottom. We will only explain the procedure for the direction to the right as the other three are carried out accordingly.

Let $p_{x,y}$ be the point at position x, y in the image. We calculate the average 3D position of some of its neighbors on the right as

$$p_{\text{right}} = \frac{1}{m_p} \sum_{i=1}^{m_p} p_{x+i,y}, \tag{1}$$

where m_p is the number of points used for the average (3 in our implementation). We take this average instead of just the neighbor $p_{x+1,y}$ to account for noise and the possible existence of veil points. Next we calculate the 3D distance $d_{\text{right}} = ||p_{x,y} - p_{\text{right}}||$. We then calculate a score based on the quotient of d_{right} and δ as

$$s_{\text{right}} = \max\left(0, 1 - \frac{\delta}{d_{\text{right}}}\right). \tag{2}$$

This gives us a value in $[0, 1)$, with high values meaning a substantial increase between the typical neighbor distance and the distance to the points on the right, indicating a probable border.

Next, we apply a smoothing operation on the score values to achieve continuous borders and avoid disruptions coming from sensor noise.

To determine if a given point $p_{x,y}$ is in the foreground or in the background, we have to check if the range value (distance to the original sensor position) of $p_{x,y}$ is lower or higher than the range of p_{right}. A lower value indicates an obstacle border, a higher value indicates a shadow border.

For all points $p_{x,y}$ that are potential obstacle borders, we now search for a corresponding shadow border to the right, selecting the one with the highest score in a maximum 2D distance (3 pixels in our implementation). Depending on the score s_{shadow} of this potential shadow border we slightly decrease s_{right} according to

$$s'_{\text{right}} = \max(0.9, 1 - (1 - s_{\text{shadow}})^3) \cdot s_{\text{right}}. \tag{3}$$

In this way, we reduce the score by up to 10% for small values of s_{border}.

In a last step, we check if s'_{right} is above a threshold (0.8 in our implementation) and if it is a maximum regarding $p_{x-1,y}$ and $p_{x+1,y}$. If this is the case, we mark $p_{x,y}$ as an obstacle border, its counterpart from above as a shadow border, and all pixels in between as veil points. Figure 1(a) displays an example of the output of this procedure. In this figure, the different kinds of border points are marked in different colors.

IV. Interest point extraction

A. Motivation

The detection of interest points is an important step to reduce the search space for feature extraction and focus the attention on significant structures. We have the following requirements for our interest point extraction procedure: i) it must take information about borders and the surface structure into account; ii) it must select positions that can be reliably detected even if the object is observed from another perspective; and iii) the points must be on positions that provide stable areas for normal estimation or the descriptor calculation in general.

B. Overview

Stable interest points need significant changes of the surface in a local neighborhood to be robustly detected in the same place even if observed from different perspectives. This typically means, that there are substantially different dominant directions of the surface changes in the area. To capture this, we

- look at the local neighborhood of every image point and determine a score how much the surface changes at this position and a dominant direction for this change, also incorporating the information about borders,
- look at the dominant directions in the surrounding of each image point and calculate an interest value that represents i) how much these directions differ from each other and ii) how much the surface in the point itself changes (meaning how stable it is),
- perform smoothing on the interest values, and
- perform non-maximum suppression to find the final interest points.

The most important parameter of this process is the *support size* σ, which is the diameter of the sphere around the interest point, that includes all points whose dominant directions were used for the calculation of the interest value. This is the same value that will later on be used to determine which points will be considered in the calculation of the descriptor. Choosing the value of σ depends a lot on the size of the structures that we want to find. In general, the higher the value, the more points are used to calculate the feature, which therefore becomes more stable. But in the context of object recognition it should be smaller than the object itself to have some robustness against partial occlusion. We found in our experiments, that 25% of the average object size is a reasonable value. For objects of very different sizes it might be necessary to use multiple scales.

C. The Algorithm in Detail

We start by calculating the directions of the borders we extracted in the previous step. For each point we know if it has a border on its top, right, left, or bottom. Thereby every border pixel already encodes the direction of the border in steps of $45°$. Please note that the estimation of this quantity can be improved by averaging over multiple border pixels in a local neighborhood. Please furthermore note, that we use range images in spherical coordinates, which look distorted when visualized in 2D. If the estimation would be done directly in the range image space, this distortion would influence the calculation of 2D directions in the image itself. To prevent this, we perform the calculations on the 3D points corresponding to the pixels, using local normals. We estimate the normals using PCA on a local 2D neighborhood of the points, where we disregard neighbors with 3D distances above 2δ (see Section III-C).

Since we also want to consider the changes on the surfaces that are not related to borders, we calculate the principal curvature directions at each point, which gives us the principal direction and the magnitude λ (the largest eigenvalue) of the curvature. Every point in the image p_i gets an associated main direction v, which is the border direction in every border point and the principal direction in every other point. All of these directions get a weight w that is 1 for every border point and $1-(1-\lambda)^3$ for every other point (this expression scales the magnitude upwards, while keeping it in $[0,1)$). Figure 1(b) shows an example of the values of w.

Until now, all the calculations were done on fixed 2D pixel radius surroundings. From now on, the actual 3D support size σ will be used. As long as enough points are available inside of the sphere with diameter σ, this will make the method invariant to resolution, viewing distance and non-uniform point distributions.

For every image point p we look at all its neighbors $\{n_0, \cdots, n_N\}$ that are inside of the support size (3D distance below $\frac{\sigma}{2}$) and do not have a border in between. Each of those points n_i has a main direction v_{n_i} and a weight w_{n_i}. To get back to 2D direction vectors, which helps us reduce the influence of noise from the normal estimation, we project the directions onto a plane perpendicular to the direction from the sensor to p. This leads us to a one dimensional angle α_{n_i} for each n_i.

Since two opposite directions do not define a unique position and since the principle curvature analysis does not provide a unique direction, we transform the angles in the following way:

$$\alpha' = \begin{cases} 2\cdot(\alpha+180°) & \text{for} \quad \alpha \le -90° \\ 2\cdot\alpha & \text{for} \quad -90° < \alpha \le 90° \\ 2\cdot(\alpha-180°) & \text{for} \quad \alpha > 90° \end{cases} \tag{4}$$

We furthermore smooth all the weights and angles by applying a bounded Gaussian kernel.

We now define the interest value $I(p)$ of point p as follows:

$$\begin{align} I_1(p) &= \min_i \left(1 - w_{n_i} \max(1 - \frac{10\cdot||p-n_i||}{\sigma}, 0)\right) \tag{5} \\ f(n) &= \sqrt{w_n \left(1 - \left|\frac{2\cdot||p-n||}{\sigma} - \frac{1}{2}\right|\right)} \tag{6} \\ I_2(p) &= \max_{i,j}(f(n_i)f(n_j)(1-|\cos(\alpha'_{n_i} - \alpha'_{n_j})|)) \tag{7} \\ I(p) &= I_1(p)\cdot I_2(p) \tag{8} \end{align}$$

The Term I_1 scales the value of I downwards, if p has neighboring points with high weights (strong surface changes) close by, thereby satisfying our desired property to put interest points only on locally stable surface positions. The term I_2 increases the interest value if there is a pair of neighbors with very different and strong main directions in the vicinity. After I is calculated in every image point, we perform an additional smoothing of the values over the image.

In a final step we now select all maxima of I above a threshold as interest points. See Figure 1(c,d) for an example, where the values of I for two different values of σ are visualized and the interest points are marked. Note, how the interest points are in the corners of the chairs for a small support size, whereas they move more to the middle for higher values.

V. The NARF descriptor

A. Motivation

Feature descriptors describe the area around an interest point in a way that makes efficient comparison regarding similarity possible. Our goals in the development for the NARF descriptor were i) that it captures the existence of occupied and free space, so that parts on the surface and also the outer shape of an object can be described, ii) that it is robust against noise on the interest point position, and iii) that it enables us to extract a unique local coordinate frame at the point. Compared to our former work using range value patches, mainly ii) and iii) needed improvement. For the latter, the normal vector at the point can be used, which leaves the rotation around the normal to be determined.

While many feature descriptors in 3D are invariant to the rotation around the normal (like spin images [6]), or even the complete 3D orientation [10], it is helpful to have the information about this orientation available for multiple reasons. For one, it might be desirable to be able to not use a unique orientation, e.g., if we only search for correspondences with a fixed patch orientation, as in the case of a wheeled robot searching for correspondences between its map and the environment. An invariance regarding the robot's roll might unnecessarily increase the size of the search space, since the robot will operate at roll angle zero most of the time. On the other hand, in cases where the unique orientation is used, it enables additional filtering for consistent local coordinate frames between features. The NARF descriptor enables us to extract a unique orientation around the normal. The underlying idea is similar to what is done in SIFT [7] and SURF [2]. Yet, unlike its 2D siblings, this orientation together with the normal defines a complete 6DOF transformation at the position of the interest point.

B. Overview

To compute the NARF descriptor in an interest point, we

- calculate a normal aligned range value patch in the point, which is a small range image with the observer looking at the point along the normal,
- overlay a star pattern onto this patch, where each beam corresponds to a value in the final descriptor, that captures how much the pixels under the beam change,
- extract a unique orientation from the descriptor,
- and shift the descriptor according to this value to make it invariant to the rotation.

The last two steps are optional, as explained above.

Please consider Figure 3(a,b) for a visual example of the process.

C. The Algorithm in Detail

As previously mentioned, we build on the normal aligned range value patches that we used as a descriptor before. Those can be calculated by creating a local coordinate system with zero in the interest point position, the z-axis facing in the normal direction and y being oriented according to the upright vector in the world coordinate frame. We then transform all points within the support radius $\frac{\sigma}{2}$ (see Section IV-B) into this coordinate frame. The resulting x and y coordinates define the cell of the descriptor in which a point falls, and the minimum over all z values is the value of a cell. A cell where no 3D points fall into gets the maximum value of $\frac{\sigma}{2}$. The normal is calculated using PCA on all points that will be used to calculate the descriptor to maximize it's stability. This size of the image patch should be high enough to keep enough descriptive structure, but low enough to not surpass the typical resolution of the scan. We chose size of 10×10 pixels for our experiments. To prevent problems in areas where the resolution of the original scan is low, interpolation between cells or usage of ray tracing may be necessary. In the next step we put a Gaussian blur onto the patch. Then we project a star shaped pattern with n beams onto it (see Figure 3(b)), where n will be the size of the NARF-descriptor. We chose $n = 36$ for our experiments, i.e., $10°$ between the beams. For each beam b_i, we select the set of cells $\{c_0, \cdots, c_m\}$ that lie under it, with c_0 being the middle of the patch and the rest ordered according to the distance to c_0. The value of the i-th descriptor cell D_i will then be

$$w(c_j) = 2 - \frac{2 \cdot ||c_j - c_0||}{\sigma} \tag{9}$$

$$D'_i = \frac{\sum_{j=0}^{m-1} (w(c_j) \cdot (c_{j+1} - c_j))}{\sum_{j=0}^{m-1} w(c_j)} \tag{10}$$

$$D_i = \frac{\text{atan2}\left(D'_i,\ \frac{\sigma}{2}\right)}{180°} \tag{11}$$

where $w(c_j)$ is a distance-based weighting factor, that weights the middle of the patch with 2 and decreases to 1 towards the outer edges of the patch. The basic intuition for D'_i is: the closer to the center a change in the surface is, and the stronger the change is, the more the beam value will deviate from 0. The step from D'_i to D_i is for normalization purposes and scales every cell to [-0.5, 0.5]. Please consider Figure 3(a,b). At the bottom of (b) the final descriptor is visualized and the arrows mark corresponding beams. Beams that lie on a flat surface have low values, whereas beams going over the border have high values.

Until now, the descriptor was not invariant to the rotation around the normal. We now try to find one or more unique orientations for the descriptor. For this purpose we discretize the possible $360°$ into a number of bins and create a histogram. The value for a histogram cell corresponding to

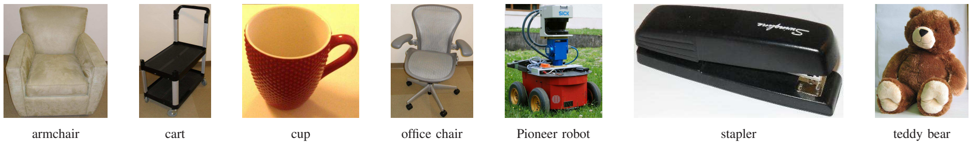

Fig. 4. Objects used for the experiments. For each of these objects we obtained a complete point cloud model.

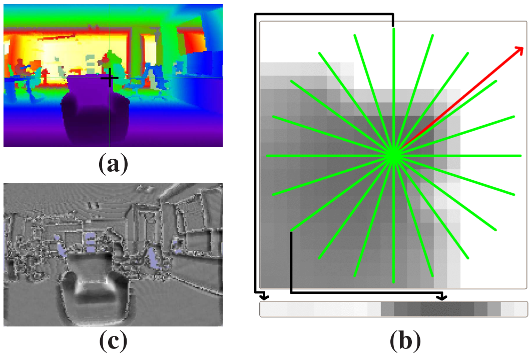

Fig. 3. **(a)**: A range image of an example scene with an armchair in the front. The black cross marks the position of an interest point. **(b)**: Visualization how the descriptor is calculated. The top shows a range value patch of the top right corner of the armchair. The actual descriptor is visualized on the bottom. Each of the 20 cells of the descriptor corresponds to one of the beams (green) visualized in the patch, with two of the correspondences marked with arrows. The additional (red) arrow pointing to the top right shows the extracted dominant orientation. **(c)**: The descriptor distances to every other point in the scene (the brighter the higher the value). Note that mainly top right rectangular corners get low values. *Best viewed in color*

angle β is

$$h(\beta) \;=\; \frac{1}{2}+\frac{1}{n}\sum_{i=1}^{n} D_i\cdot\left(1-\frac{|\beta-\gamma_i|}{180^\circ}\right)^2, \qquad (12)$$

where γ_i is the angle corresponding to the i-th descriptor cell. We select the histogram bin with the maximum as the dominant orientation of the patch. If there is another cell with a value above 80% of the maximum, we create a second feature with this orientation. We can now shift the descriptor to create the rotationally invariant version.

The resulting descriptors can now easily be compared using standard distance functions. We chose the Manhattan distance divided by the number of cells in the descriptor, which normalizes the distance to values in $[0, 1]$. Figure 3(c) visualizes all the descriptor distances to the selected point in the range image.

VI. Experiment - Interest Point Stability

Our goal for this experiment was to analyze how stable the position of the interest points is relative to changes in scale (distance to the object) and viewing angle. For this purpose we selected seven object models in the form of complete point clouds. These models represent very different kinds of objects and include furniture-sized objects and tabletop objects (see Figure 4 for a list of the models). To be able to evaluate objects of very different sizes together, we scaled all the models to a bounding sphere with a diameter of 1.0 m. As support size we used 0.25 m, which is large enough to cover most of the significant structure of the models, but low enough to account for partial occlusions. For each of these objects we simulated 50 noiseless views from different angles and distances around the object. Then we simulated 100 different views with additional noise on the point positions (see Figure 5(c) for example views with marked interest point positions). We then compare the interest point positions on the first set of views with each view of the other set.

We calculate the repeatability of the interest point extraction according to the method proposed in [16], which is a 3D extension of a commonly used method in the computer vision literature [9]. For every interest point i_1 in one view of the object, we search for the closest interest point i_2 in another view and calculate the ratio of the intersection between the two spheres with radius $r = \frac{\sigma}{2}$ around them as:

$$s = 1 - \frac{3}{4}\frac{d}{r} + \frac{1}{16}\left(\frac{d}{r}\right)^3, \qquad (13)$$

where d is the 3D Euclidean distance between the two interest points. For this purpose, we only take unoccluded points into account, meaning we check if the point itself is influenced by self occlusion regarding the current pair of views, and reject it if this is the case.

Figure 5(a) shows the result of the cross comparison of the positions of the interest points of all the objects. The scores are shown dependent on the angular viewpoint change and the difference in observing distance (leading to a difference in scale in the range images). While a high change in scale obviously has a negative influence on the interest point stability (see the different plots for different scales), this influence seems minor compared to angular changes, which is why we will omit changes in scale from now on to save space. The dashed black line shows how many samples were available for the calculation of the averages. This value naturally decreases with the angle, since increasingly fewer points are actually visible from both perspectives. The solid black line close to the bottom represents the repeatability value for a random point on the object surface, thereby defining a minimum value below which the results are not meaningful anymore. This plot shows, that the interest point extraction is relatively stable over a wide area of viewpoint changes. For changes below 20° the interest points share about 70% of their support size on average and about 55% for changes below 60°.

Figure 5(b) shows the same analysis, but for every model individually. The cup shows an unusual behavior compared to the rest. This results from the fact that it is rotationally symmetric over a wide range, making the interest point position much more dependent on the viewpoint. The increase in value between 100° and 160° is caused by interest points on the left border of the cup again being extracted when seen from the other side on the right border. The cart, the chair and the pioneer robot models have higher values for the dashed lines. This mainly indicates that a higher number of interest points was extracted on these models, leading to

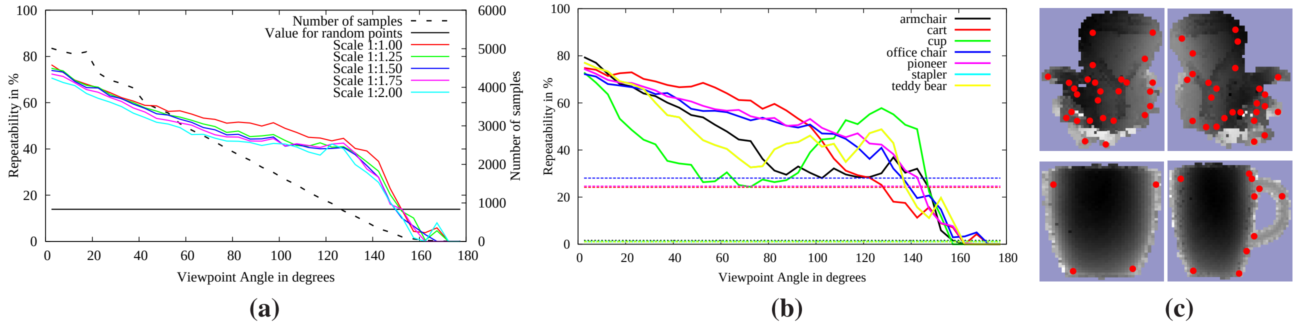

Fig. 5. **(a)**: This graph shows how repeatable the interest point detection is regarding differences in the viewpoint angle (x axes) and differences in the scale between two views of the object (the five solid plots). A scale of 1:1 means the object was seen at the same distance, 1:2 means at double the distance. The dashed line shows how many samples were available for averaging. The number goes down with increased angle since less and less pairs of points are actually visible from both perspectives. The solid black line follows from the average distance of a random point on the models surface to the next interest point, giving a minimum value for the repeatability, below which the value is meaningless. **(b)**: Same as (a), but per object model. The dashed lines with constant value correspond to the solid black line in (a). **(c)**: Examples of the simulated views used to create (a) showing two views of the backside of an office chair and two views of a cup. The interest points extracted on the views are marked in the images. *Best viewed in color*

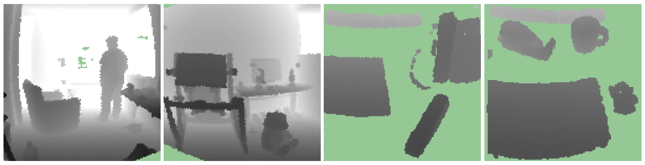

Fig. 6. Examples for the created scenes. From left to right: office scene with armchair, office scene with teddy bear, tabletop scene with stapler, tabletop scene with cup.

the situation that random points on the surface on average are closer to the interest point positions.

VII. Experiment 2 - Matching Capability

In this experiment we wanted to test the capability of our NARF-descriptors to match features from an object model to the features in a scene containing the model. For this purpose we collected 10 scenes with a 3D laser range finder in a typical cluttered office environment and 10 scenes with a stereo camera system of tabletop scenes. Then we artificially added our models (including noise on the 3D point positions) to these scenes - the armchair, cart, chair, robot, and the teddy bear to the office scenes and the stapler and the cup to the tabletop scenes, thereby creating 70 scenes with one known object in each. In this process, the objects could appear in an arbitrary yaw and x,y positions, while the hight was restricted to the floor/tabletop and roll and pitch were in their natural orientations. The latter were chosen to be able to test our descriptor also without the rotational invariance. For the tabletop scenes, the table plane itself was removed, as it is often done in tabletop object recognition. Figure 6 shows some examples of the created scenes.

To match the objects against the scenes, we sampled poses from our models that differed from the ground truth poses between 0° and 50°.

The resulting numbers of true positives and false positives are summarized in Figure 7(a) as ROC (Relative Operating Characteristic) curves. Please note the logarithmic scales, which show the interesting part of the plot, the bottom left corner, better. The absolute number of false positives is much higher than the absolute number of true positives, which makes areas with a high ratio of false positives less useful. The thicker plots mark areas, where the number of true positives to false positives is lower than 1:10. The plot marked *NARFs all points* is for our NARF feature descriptors extracted at every image point, without using the interest points. The plot marked *NARFs int. points* shows the performance of the NARF features together with the interest points. The plot marked *NARFs(ri) int. points* is for the rotational invariant version of the NARF descriptor. The plot marked *RVPs int. points* is using the range value patches that we used in our former work [14], [13] as feature descriptors. As can be seen, the interest points are a definite improvement compared to random point positions. Additionally, the rotationally invariant version and the rotationally variant version of the NARF descriptor outperform the RVP descriptors with interest points.

To evaluate if the system can be used for object recognition, we used the extracted feature matches to actually calculate object poses. Since every feature encodes a local 6DOF coordinate frame, one feature match is enough to calculate an object position and orientation. We calculated a pose for every match with a descriptor distance below 0.05. Figure 7(b) shows the average number of correct poses versus false poses for one object and one scene. An object pose was classified as correct if its error compared to the true pose was below 0.3 times the object radius in translation and below 15° in rotation. For angular differences below 20° there are typically 2 correct poses versus 10 false poses. It would be up to a spatial verification step to reject those false positives.

Figure 7(c) shows the true poses per used object model. The armchair performs best, since it mostly consists of large rectangles, that do not change much with moderate changes in the viewpoint. The cup model performs worst, which is due to the low number of interest points on the model and additionally the rotational symmetrical structure. Yet, the position of the cup, disregarding the orientation, can still be found, as the plot labeled *cup wrong rot* shows.

We also tested, at which position in the set of matches (ordered by descriptor distance) the first correct object pose typically occurs. Table I gives an overview per object depending on the viewpoint angle difference α (a:armchair, ca:cart, cu:cup, oc:office chair, pr:pioneer robot, s:stapler, tb:teddy bear). Whereas the first number in the table tells the average position in the set of matches (ordered by descriptor

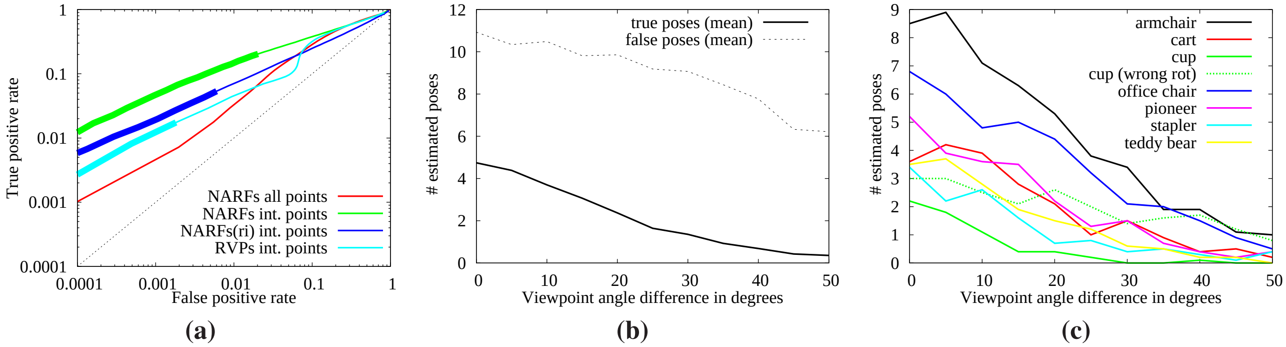

Fig. 7. **(a)**: This graph shows ROC curves for feature matches where the threshold for the descriptor distance increases from the bottom left to the top right. Please note, that the axes are logarithmic. The higher a line in this graph, the better is the performance. The parts of the plots that are printed thicker mark the areas, where the ratio between true positives and false positives, in absolute numbers, is better than 1:10. **(b)**: The average number of true/false poses extracted from the feature matches for all objects. **(c)**: The average number of true poses extracted from the feature matches per object. To account for the symmetry of the cup, the plot marked *cup (wrong rot)* gives the number of poses that are correct regarding (x, y, z)-position, but not necessarily regarding orientation. *Best viewed in color*

TABLE I

POSITION IN THE SET OF MATCHES WHERE THE FIRST CORRECT OBJECT POSE TYPICALLY OCCURS AND THE SUCCESS RATE.

α	a	ca	cu	oc	pr	s	tb
0	1.6/1	1.2/0.9	1.3/1	1.2/0.9	1.9/1	1.2/0.9	2.1/0.8
5	1.1/1	2.6/0.8	1.5/0.8	1.6/0.8	1.6/0.7	2.9/0.9	1.2/1
10	1.6/1	1.3/0.9	2/0.5	1.7/0.9	1.2/0.8	2.8/0.8	2/0.9
15	1.8/1	2.4/0.8	1/0.3	1.1/0.8	2.9/0.8	4/0.8	2.9/0.8
20	1.1/1	3.3/0.9	1.5/0.2	1.2/0.9	4.2/0.6	2.6/0.5	4.6/0.7
25	1.8/0.9	3.5/0.4	2/0.2	3.3/0.6	1.5/0.6	3.8/0.6	2.3/0.6
30	3.4/0.9	6.8/0.6	−/0	2/0.6	5/0.6	4/0.3	1/0.3
35	2/0.8	4.8/0.6	−/0	2.2/0.4	6.4/0.5	3/0.4	3/0.3
40	2.4/0.7	2.3/0.3	6/0.1	3/0.3	2.7/0.3	3/0.2	1/0.1
45	3.6/0.7	5.7/0.3	−/0	1.5/0.2	5.5/0.2	8/0.1	1/0.1
50	5/0.7	5.5/0.2	−/0	2.5/0.2	2.3/0.3	2.7/0.3	−/0

distance) where the first correct object pose occurs, the second is the rate with which a correct position was found. As can be seen, for viewpoint changes below 20°, the first correct pose can typically be found within 3 trials in the 80% where a correct pose could be found.

VIII. Timings

The average timings for range image creation, border detection, interest point extraction and feature descriptor calculation were 18.1 / 22.9 / 27.2 / 2.86 ms respectively for office scenes from point clouds of size 115061 and range image resolution 0.4° with 104 features per scene. The same numbers for the tabletop scenes were 26.4 / 5.41 / 6.85 / 0.989 ms for tabletop scenes from point clouds of size 88395 and range image resolution 0.2° with 48 features per scene. These values were obtained on an Intel I7 quadcore.

IX. Conclusions

In this paper we presented NARF: a novel approach to feature extraction in range images. NARFs are generated at points where the surface is mostly stable but changes significantly in the immediate vicinity. It furthermore makes explicit use of border information. In practical experiments we demonstrated that NARFs give better matching results than the features we used in our earlier work on object recognition, which in turn outperformed the well known spin images. All the software and datasets used for the experiments are available as open source.

X. Acknowledgements

The authors gratefully acknowledge the help of everyone at Willow Garage. This work has partly been supported by the European Commission under contract numbers FP7-231888-EUROPA and FP7-248258-First-MM.

References

[1] A. Ansar, A. Castano, and L. Matthies. Enhanced Real-time Stereo Using Bilateral Filtering. *2nd Int. Symp. on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2004.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *Proc. of the Europ. Conf. on Comp. Vision (ECCV)*, 2006.

[3] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proc. of the Europ. Conf. on Comp. Vision (ECCV)*, 2004.

[4] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Proc. of the third Eurographics symposium on Geometry processing*, 2005.

[5] Q. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. Reassembling fractured objects by geometric matching. *ACM Transactions on Graphics (TOG)*, 25(3):569–578, 2006.

[6] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999.

[7] D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 1999.

[8] B. Matei, Y. Shan, H.S. Sawhney, Y. Tan, R. Kumar, D. Huber, and M. Hebert. Rapid object indexing using locality sensitive hashing and joint 3D-signature space estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):1111 – 1126, July 2006.

[9] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

[10] R.B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Kobe, Japan, May 12-17 2009.

[11] R.B. Rusu, Z.C. Marton, N. Blodow, and M. Beetz. Learning Informative Point Classes for the Acquisition of Object Model Maps. In *Proc. of the 10th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, 2008.

[12] R.B. Rusu, Z.C. Marton, N. Blodow, and M. Beetz. Persistent Point Feature Histograms for 3D Point Clouds. In *Proc. of the 10th Int. Conf. on Intelligent Autonomous Systems (IAS-10)*, 2008.

[13] B. Steder, G. Grisetti, and W. Burgard. Robust place recognition for 3D range data based on point features. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.

[14] B. Steder, G. Grisetti, M. Van Loock, and W. Burgard. Robust online model-based object detection from range images. In *Proc. of the IEEE/RSJ Int. Conf. on Int. Robots and Systems (IROS)*, 2009.

[15] S. Stiene, K. Lingemann, A. Nüchter, and J. Hertzberg. Contour-based object detection in range images. In *Proc. of the Third Int. Symposium on 3D Data Processing, Visualization, and Transmission*, 2006.

[16] R. Unnikrishnan. *Statistical Approaches to Multi-Scale Point Cloud Processing*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2008. Chapter 4.