

# A Novel Local Surface Description for Automatic 3D Object Recognition in Low Resolution Cluttered Scenes

Syed Afaq Ali Shah<sup>1</sup>, Mohammed Bennamoun<sup>1</sup>, Farid Boussaid<sup>2</sup>, Amar A. El-Sallam<sup>1</sup>

<sup>1</sup>School of Computer Science and Software Engineering,

<sup>2</sup>School of Electrical, Electronic and Computer Engineering,  
The University of Western Australia

{afaq,bennamoun,elsallam}@csse.uwa.edu.au

## Abstract

*Local surface description is a critical stage for feature matching and recognition tasks. This paper presents a rotation invariant local surface descriptor, called 3D-Div. The proposed descriptor is based on the concept of 3D vector field's divergence, extensively used in electromagnetic theory. To generate a 3D-Div descriptor of a 3D surface, a local surface patch is parameterized around a randomly selected 3D point at a fixed scale. A unique Local Reference Frame (LRF) is then constructed at that 3D point using all the neighboring points forming the patch. A normalized 3D vector field is then computed at each point in the patch and referenced with LRF vectors. The 3D-Div descriptor is finally generated as the divergence of the reoriented 3D vector field. We tested our proposed descriptor on the challenging low resolution Washington RGB-D (Kinect) object dataset, for the task of automatic 3D object recognition. Reported experimental results show that 3D-Div based recognition achieves 93% accuracy as compared to 85% for existing state-of-the-art depth kernel descriptors [2].*

## 1. Introduction

Object recognition in complex real environment scenes in the presence of occlusion, due to multiple objects, and clutter due to the existence of unwanted objects is a challenging task. The core of a robust recognition system is to extract distinctive features from range images (mesh or 3D point clouds) and videos under these conditions. There are two basic types of features that are used to represent a range image, namely global features and local features. Global feature representations are widely used in shape retrieval and object detection techniques, but these features are sensitive to clutter and occlusion [1, 12]. On the other hand, a local surface feature is robust to these conditions as it is more distinctive and descriptive. This is because the

3D surface is represented by a set of 3D keypoints and local surface descriptors. The latter play an important role in 3D object recognition.

A number of local surface descriptors have been proposed in the literature for accomplishing critical task like object recognition. Tombari et al. [20] proposed a descriptor named Signature of Histograms of Orientations (SHOT) for 3D object recognition. They first constructed a local reference frame for a feature point, and divided the neighborhood space into a 3D spherical volumes. They then generated a local histogram for each volume by accumulating the number of points according to the angles between the normal at the feature point and those at the neighborhood points. Knopp et al. [8] proposed 3D SURF descriptor for object retrieval and classification. Their approach is an extension of 2D SURF. They voxelized the 3D object in a volumetric cube using the intersection of faces with the grid-bins. The 3D SURF descriptor is then computed at the maxima of the voxelized grid. Masuda proposed a Log-Polar Height Map (LPHM) descriptor [11], where the neighboring points were projected onto the tangent plane of the keypoint and the depth was stored in the log-polar coordinate frame to get LPHM descriptor. Flint et al. [4] proposed the THRIFT descriptor, which is an extension of SURF and SIFT. In their approach, the deviation of the surface normals is used to create a 1D histogram of descriptors.

Despite significant research efforts, existing local surface descriptors suffer from relatively low descriptiveness because only partial information of the underlying surface is used to generate the descriptors [12, 14]. To address these limitations, we propose the novel use of 3D vector field's divergence, for feature representation in computer vision. Divergence has been widely used for the analysis of multi-dimensional standing waves and flux from electromagnetic surfaces. The divergence of a vector field at a point  $P$  is the net outward flux per unit volume, as the volume shrinks to zero:

$$\text{div}(P) = \lim_{\Delta v \rightarrow 0} \frac{\int_s \langle P, \vec{F} \rangle ds}{\Delta v} \quad (1)$$

where  $v$  is the volume,  $s$  is the surface and  $\vec{F}$  is the outward flux at each point on the surface. The divergence characterizes the volume density of the outward flux of a vector field from a smaller local surface around a given point. It maps vector fields to scalar fields, by measuring how much the former is spreading out at each point of the surface.

**Organization of the paper:** The next section reviews related work. Section 3 describes the proposed methodology for generating 3D-Div surface descriptor. Section 4 discusses the use of 3D-Div descriptor for 3D object recognition in a video sequence. Section 5 reports experimental results for 3D object recognition and evaluates performance against state-of-the-art. The paper is concluded in Section 6.

## 2. Related Work

This research focuses on developing novel rotation invariant local surface descriptors for 3D object recognition. The following is a brief survey of related work in the area of 3D object recognition. Mian et al. [12] proposed a 3D-Tensor descriptor for 3D object recognition. They first constructed a local reference frame by selecting a pair of vertices that satisfied certain geometric constraints. They then generated a 3D-tensor descriptor by constructing a local 3D grid over the range image, and summed the surface areas which intersected each bin of the 3D grid. In a method proposed by Tabia et al. [19], a set of feature points was detected first. Then the geodesic distances were computed from each feature point to all the vertices on the 3D surface. Next, a feature descriptor was generated for each feature point by accumulating these geodesic distances into a histogram. These descriptors were subsequently used together with a Bag-of-Feature method for 3D shape categorization. Jain et al. [6] proposed an EVD descriptor for 3D object recognition. In their approach, a geodesic distance based affinity matrix was calculated first for a 3D object. An eigenvalue decomposition was then performed on the affinity matrix and the  $k$  largest eigenvalues were used to form the descriptor. To perform object recognition, the distance between two objects was measured by calculating the  $\chi^2$ -distance between their EVD descriptors. Johnson and Hebert proposed the renowned Spin Image (SI) descriptor [7]. They used the surface normal at a keypoint as the local reference frame and computed a tangent plane at the keypoint by using the neighboring points. A region around the keypoint was then considered in which two distances were computed to determine the spin image. The spin image is robust to clutter and occlusion. However, it is sensitive to variations in mesh resolution and is weakly distinctive [12]. Smeets et al. [17] proposed a geodesic distance matrix and a

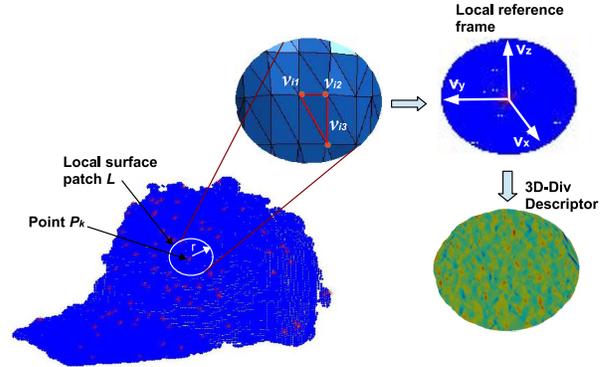


Figure 1. Randomly selected points  $P_k$ , shown in red on the 3D surface. A local surface patch  $L$  is defined using a sphere of radius  $r$  (shown in white) centered at  $P_k$  (shown in red). The local reference frame ( $V_x$ ,  $V_y$  and  $V_z$ ) is constructed by using the triangle of the surface mesh with vertices  $v_{i1}$ ,  $v_{i2}$  and  $v_{i3}$  (shown in orange). 3D-Div descriptor of a local surface patch is formed by combining all the divergence values (which are shown in different colors). (Figure best seen in color).

diffusion distance tensor as descriptors for 3D shape recognition. The empirical results showed that the combination of these two descriptors provide a better object recognition performance as compared to a single descriptor. More recently, kernel descriptors comprising of up to five different cues, extracted from depth map and 3D point cloud, were proposed to provide a way of turning any pixel attribute to patch-level features [2]. A significant increase in accuracy of object recognition can be achieved by combining all the proposed cues.

## 3. Proposed Methodology

Vector fields are a common data representation in computer vision tasks. For instance, histogram based approaches (Spin Image [7] and 3D-Tensor [12]), signature based techniques (Point's fingerprint [18] and [11]) and recently proposed signature of histogram based approach (SHOT descriptor [20]), all rely on vector fields to construct the Local Reference Frame (LRF), also known as 3D coordinate basis, to generate rotation invariant features. These techniques use either the scalar product, angle between vectors or the vector itself for generating LRF. By using these computationally efficient vectors for only LRF construction, their significance and potential use has not been fully exploited by the computer vision community. In this paper, we propose the use of the 3D vector field to capture the significant geometrical information of the underlying 3D surface for local feature description. We represent each component of the 3D vector field as a third-order tensor ( $k \times k \times k$ ), and employ the divergence of the 3D vector field to generate a novel local surface descriptor.

### 3.1. 3D-Div Representation

For an input 3D point cloud  $V_i$  (where  $V_i$  is an  $n_i \times 3$  matrix representing the 3D coordinates of its vertices and  $i = 1, \dots, M$ ) of the range image, the following steps are performed to extract 3D-Div local surface descriptor.

#### 3.1.1 LRF Construction

Given a set of randomly selected 3D points  $P_k, k=1,2,\dots,K$  on a 3D surface, a local surface patch  $L$  is first defined using a sphere of radius  $r$  centered at each point  $P_k$ :

$$L[r](P_k) = \{P_n \in \mathbb{R}^3 \mid \|P_k - P_n\|_2 \leq r\} \quad (2)$$

where  $P_n$  are the neighboring points in the local surface patch. A Local Reference Frame (LRF) for  $P_k$  is then constructed by using all the points lying on the local surface. Using PCA, the scatter matrix is calculated for the  $i$ th triangle with vertices  $v_{i1}, v_{i2}$  and  $v_{i3}$ :

$$\begin{aligned} Scat_i = & \frac{1}{12} \sum_{l=1}^3 \sum_{m=1}^3 (v_{il} - P_k)(v_{im} - P_k)^T \\ & + \frac{1}{12} \sum_{l=1}^3 (v_{il} - P_k)(v_{il} - P_k)^T \end{aligned} \quad (3)$$

An eigenvalue decomposition is then performed on the scatter matrix to get three orthogonal eigenvectors ( $v_x, v_y$  and  $v_z$ ). These eigenvectors form the basis for the LRF, as shown in Figure 1. To eliminate the sign ambiguity of LRF, each eigenvector should point in the major direction of the scatter vectors (Eq. 3). Therefore, the sign of each eigenvector is found from the sign of the inner product of the eigenvector and the scatter vectors [5].

#### 3.1.2 3D Vector Field

For each 3D point  $P_k$ , local surface patch and LRF vectors, a trilinear interpolation is first performed to get uniformly sampled points [16]. The vector field is then defined as a map  $\vec{F}$ , at each point of the local surface:  $\mathbb{E}^3 \rightarrow \mathbb{R}^3$ , where  $\mathbb{E}^3$  denotes the Euclidean 3D space equipped with a cartesian coordinate system  $(x,y,z)$  and  $\mathbb{R}^3$  is a 3D vector space. The partial derivatives of the 3D vector field  $\vec{F}$  along the x-axis are denoted by  $F_x = (u_x, v_x, w_x)^T$ , where  $u_x(x, y, z) = \frac{d}{dx}u(x, y, z)$ ,  $v_x(x, y, z) = \frac{d}{dx}v(x, y, z)$  and  $w_x(x, y, z) = \frac{d}{dx}w(x, y, z)$ . The same applies to  $F_y$  and  $F_z$ . Finally, the 3D vector field is given by:

$$\vec{F}(x, y, z) = F_x \hat{i} + F_y \hat{j} + F_z \hat{k} \quad (4)$$

where  $F_x, F_y$  and  $F_z$  are the x, y and z components of the 3D vector field respectively. The normalized 3D vector field for a 3D surface is then computed as:

$$\vec{f} = \vec{F}(x, y, z) / \|\vec{F}\| \quad (5)$$

---



---

1. For a given 3D point cloud  $V(x,y,z) \rightarrow \mathbb{R}^3$

2. Select keypoints  $P_k$

for  $k = 1, \dots, N$ , ( $N = \text{Number of keypoints}$ )

3. For each  $P_k$ , extract a local surface patch  $L_k$  at fixed scale  $r$ , using all neighboring points  $P_n$

$$L_k[r](P_k) = \{P_n \in \mathbb{R}^3 \mid \|P_k - P_n\|_2 \leq r\}$$

4. Construct LRF at  $P_k$  to get LRF vectors  $v_x, v_y$  and  $v_z$  (Eq. 2)

5. Compute 3D vector field  $\vec{F}_{k,n}(x, y, z)$

6. Compute normalized 3D vector field  $\vec{f}_{k,n}(x, y, z)$

7. Reference normalized 3D vector field with LRF vectors (obtained in step 4) to get  $\vec{f}'_{k,n}(x, y, z)$

8. Generate 3D-Div descriptor using the reoriented normalized 3D vector field

$$\begin{aligned} div \vec{f}'_{k,n} = & \nabla \cdot \vec{f}'_{k,n} = \frac{\partial f'_x}{\partial x} + \frac{\partial f'_y}{\partial y} + \frac{\partial f'_z}{\partial z} \\ \text{end} \end{aligned}$$


---



---

Table 1. Computation of normalized 3D vector field and 3D-Div descriptor.

#### 3.1.3 3D-Div Descriptor

The normalized 3D vector field  $\vec{f}$  for the local surface is then aligned with LRF vectors while the 3D divergence is computed for the reoriented normalized 3D vector field  $\vec{f}'$ :

$$div \vec{f}' = \nabla \cdot \vec{f}' = \frac{\partial f'_x}{\partial x} + \frac{\partial f'_y}{\partial y} + \frac{\partial f'_z}{\partial z} \quad (6)$$

The  $\nabla \cdot \vec{f}'$  in Eq. 6 evaluates the magnitude of a 3D vector field's sink or source at each point of the local surface patch, as given by Eq. 1. 3D-Div is formed by concatenating the divergence values (Eq. 6) into a vector. As Eq. 6 shows,  $div \vec{f}'$  does not depend on the location of the coordinate axis  $(x, y, z)$ , in Cartesian coordinates. Therefore, the resulting local surface feature is invariant to rotation and translation. Moreover, the reoriented normalized radial field  $\vec{f}'$  with respect to its unique local coordinate system also ensures the rotation invariance of the local feature. Thus, if a patch is rotated by some angle, the local coordinate system along with  $\vec{f}'$  also rotate by the same angle. Figure 1 shows the 3D-Div descriptor for a local surface patch. The normalized 3D vector field acts as a limiter to surface noise and therefore the divergence computed using this vector field is robust to noise. The algorithm for computing the normalized 3D vector field and 3D-Div is reported in Table 1.

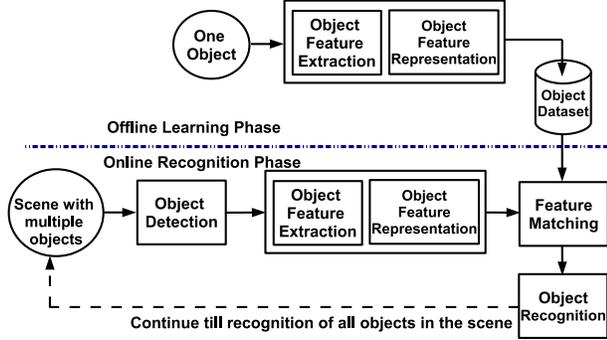


Figure 2. Automatic Object Recognition System. During offline phase, object features are extracted and stored in object database. During online recognition phase, object is detected in the scene, its features extracted for matching with object features in the database to accomplish recognition process.

## 4. 3D Object Recognition

To evaluate the performance of our rotation invariant local surface descriptor, we used 3D-Div to perform 3D object recognition. Figure 2 shows the block diagram of our complete object recognition system, including the offline feature extraction and representation, and online object recognition phases. Our algorithm is fully automatic and requires no user intervention at any stage. In the following, we elaborate the different steps of our algorithm.

### 4.1. Offline Phase

During the offline phase, the 3D-Div descriptors are extracted from the 3D point clouds of the objects. These features are stored in an object database for later use in feature matching and object recognition tasks.

### 4.2. Online Recognition Phase

During the online recognition phase, the object is first of all detected in the scene (depth image). The portion of the scene containing the detected object is extracted and thresholding used to eliminate the background. The depth map is then converted to 3D point clouds by mapping each pixel into its corresponding 3D coordinate vector [2]. The features are then extracted from the object point cloud before being matched with the model features in the database to accomplish the recognition task. This process continues until all the objects in the scene have been recognized. The various stages of the online recognition phase are discussed in more details in the following paragraphs.

#### 4.2.1 3D Object Detection

Object detection is one of the fundamental challenges in computer vision and an initial step towards object recognition. The objective of object detection is to extract an

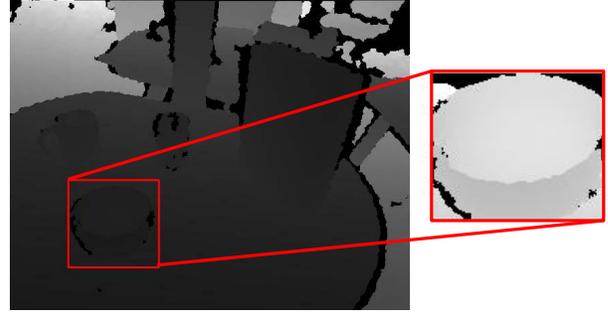


Figure 3. Object detection in depth map. The detected object is extracted from the scene and converted to 3D point cloud for feature extraction and representation.

object from its background prior to recognition. State-of-the-art object detection algorithms [3, 15, 13] rely on a sliding window based model, which scores a large number of windows in a test image using a classifier.

The HOG based sliding window classifier is among the most successful object detection techniques [10]. In this work, we used the 3D implementation of this classifier [9], for the task of object detection.

The HOG features are first extracted from the RGB and depth image to capture the shape and appearance information for each view of the object. The gradient orientations in each 8x8 pixel grid are encoded using two different quantization levels into 18 ( $0^\circ - 360^\circ$ ) and 9 orientations ( $0^\circ - 180^\circ$ ). The resulting 108 ( $4 \times (18+9)$ ) feature vector is projected into 31-dimensional feature vector [10]. The last four dimensions capture the overall gradient energy in four blocks of 2x2 cells. The recursive median filter is then used to fill in missing values in the depth map. Finally, a normalized depth histogram is computed. The depth image is normalized by using the size of the bounding box. A histogram of 20 bins (each bin having range of 0.30m) is then computed over an 8x8 grid [10].

To better capture the spatial extent of detected objects, the probability map of the object is refined by using high scoring bounding boxes found by the detector. For each bounding box having a detector score threshold above -0.6, we set

$$p(x) = p(x_o) \exp(-\beta |x - x_o|^2), x \in \alpha \quad (7)$$

where  $\alpha$  is the set of 3D points in the bounding box,  $x_o$  is the center of the bounding box, and the parameter  $\beta$  controls how quickly the probability decreases with increasing depth differences. The output of object detection stage is shown in Figure 3. The detected object captured by the bounding box is then extracted from the scene, as shown in Figure 3 and a threshold  $t_d$  is applied to eliminate depth values above the threshold. In our case, we set  $t_d$  equal to the mean of the depth value. The depth map after the threshold is then

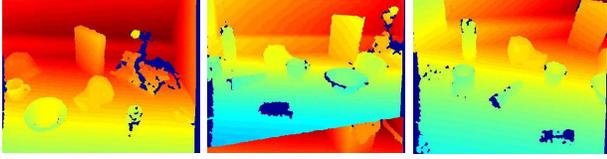


Figure 4. Different depth frames of same video sequence showing significant rotation and translation of objects in the scenes due to change in view point and viewing angle with the horizon.

converted to 3D point clouds by mapping each pixel into its corresponding 3D coordinate vector [2].

#### 4.2.2 Feature Extraction and Matching

The local surface features (3D-Div) are then extracted from the scene point cloud by using the procedure described in Section 3. The local surface 3D-Div descriptors from the scene are matched with the model features to accomplish feature matching process. We used the correlation coefficient for matching a pair of 3D-Div descriptors:

$$C_c = \frac{n_v \sum_{i=1}^{n_v} u_i v_i - \sum_{i=1}^{n_v} u_i \sum_{i=1}^{n_v} v_i}{\sqrt{n_v \sum_{i=1}^{n_v} u_i^2 - (\sum_{i=1}^{n_v} u_i)^2} \sqrt{n_v \sum_{i=1}^{n_v} v_i^2 - (\sum_{i=1}^{n_v} v_i)^2}} \quad (8)$$

where,  $n_v$  is the number of pairs of 3D-Div,  $u_i$  and  $v_i$  are the elements of 3D-Div from scene and the model respectively, in their region of overlap. If  $C_c$  is greater than a threshold  $t_c$  (set to 0.5), the 3D-Div of scene and model object are deemed to form a matching pair.

## 5. Experimental Results

We evaluated the proposed 3D-Div descriptor on the challenging low resolution RGB-D (Kinect) Object Dataset [9], which contains 250,000 segmented RGB-D images of 300 everyday objects in 51 categories. The range images have been acquired at viewing angles of  $30^0$ ,  $45^0$  and  $60^0$  with the horizon. The point cloud resolution varies from 350 to 16900 points for different objects. The dataset also includes the RGB-D cluttered scenes, which consists of different video sequences of office and kitchen environments. Each scene contains walls, doors, tables and a variety of objects placed on multiple supporting surfaces [10]. We randomly selected one object from each category, thus making total of 51 objects. To test our rotation invariant local surface descriptor, we only selected range images of objects acquired at viewing angle of  $45^0$  with the horizon. We then extracted 3D-Div descriptors from these objects to form our object features database. We performed evaluation of our 3D object recognition algorithm on three different video sequences. Each video sequence has been acquired at varying angles with the horizon and contains different number of objects from five categories (bowl, cap, cereal box, coffee

mug and soda can) under clutter. Figure 4 shows three different depth frames taken from the same video sequence. The rotation and translation of objects in the scene due to change in viewpoint and viewing angle with the horizon are quite evident. We subsampled each video sequence by taking every fifth frame, resulting in 176 RGB and depth scenes (images). We treated each frame as an individual scene and recognized all the objects in that frame before proceeding to the next frame in the video sequence. We detected objects in the scene and extracted 3D-Div of each object. We then matched scene features with the object features in the database for recognition task. The object that received the maximum number of feature correspondences was declared as recognized. We evaluated the 3D object recognition performance of our algorithm with recently proposed depth kernel descriptors [2], which capture five different cues: gradient, local binary pattern, size kernel, kernel PCA and spin kernel descriptor. Depth kernel descriptors have been extensively tested on Washington RGB-D dataset and have shown good recognition results. In our experiment, the performance evaluation was done on the same three video sequences. In Figure 5, we report precision-recall curves comparing the object recognition performance of our algorithm with [2] for these video sequences. Figure 5 shows that, our proposed technique achieves superior recognition performance as compared to [2] and outperformed the state-of-the-art. The overall accuracy achieved by our algorithm is 93%, while [2] attained an accuracy of 85%. The 8% higher accuracy achieved by our algorithm on the low resolution noisy dataset [9] is a significant improvement. The achieved accuracy illustrates the descriptiveness of the proposed local surface 3D-Div descriptor and its invariance to rotation and translation under clutter.

## 6. Conclusion

In this paper, we presented 3D-Div, a novel rotation invariant local surface descriptor based on the divergence of the normalized 3D vector field. We demonstrated the use of this proposed descriptor for the task of 3D object recognition on a challenging low resolution Washington RGB-D (Kinect) object dataset. Experimental results reveal that 3D-Div achieves superior 3D object recognition performance. We compared our proposed descriptor with the state-of-the-art depth kernel descriptor method [2]. 3D-Div outperformed the latter, by achieving 93% recognition accuracy as compared to 85% accuracy of depth kernel descriptors [2].

## Acknowledgment

The research leading to these results has received funding from Australian Research Council (DP110102166).

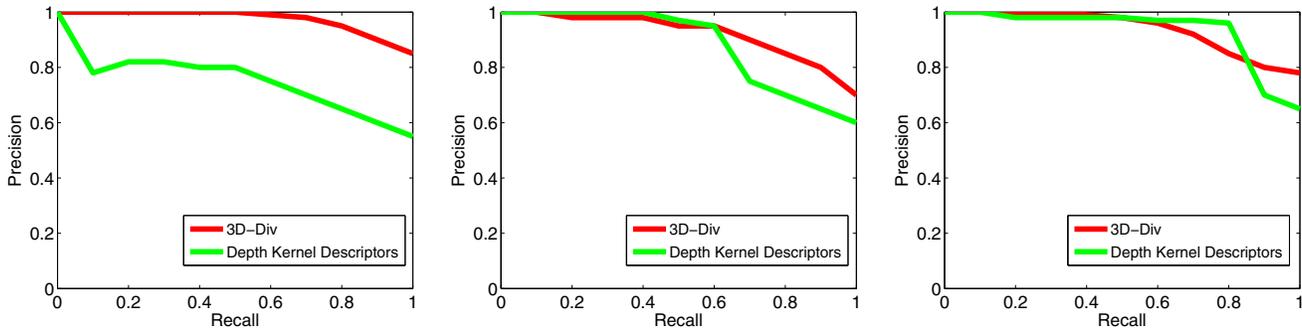


Figure 5. Precision-Recall curves for three video sequences. The performance of 3D-Div based object recognition is shown in red curve, while the green curve shows the performance of depth kernel descriptors [2].

## References

- [1] P. Bariya, J. Novatnack, G. Schwartz, and K. Nishino. 3d geometric scale variability in range images: Features and descriptors. *International Journal of Computer Vision*, 99(2):232–255, 2012.
- [2] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 821–826, 2011.
- [3] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, sept. 2010.
- [4] A. Flint, A. Dick, and A. V. D. Hengel. Thrift: Local 3d structure recognition. In *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on*, pages 182–188, dec. 2007.
- [5] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan. Rotational projection statistics for 3d local surface description and object recognition. *International Journal of Computer Vision*, 105(1):63–86, 2013.
- [6] V. Jain and H. Zhang. A spectral approach to shape-based retrieval of articulated 3d models. *Computer-Aided Design*, 39(5):398–407, 2007.
- [7] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.
- [8] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Gool. Hough transform and 3dsurf for robust three-dimensional classification. In *Computer Vision ECCV 2010*, volume 6316 of *Lecture Notes in Computer Science*, pages 589–602. Springer Berlin Heidelberg, 2010.
- [9] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824, 2011.
- [10] K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3d scenes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1330–1337, 2012.
- [11] T. Masuda. Log-polar height maps for multiple range image registration. *Computer Vision and Image Understanding*, 113(11):1158–1169, 2009.
- [12] A. S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1584–1601, 2006.
- [13] N. Payet and S. Todorovic. From contours to 3d object detection and pose estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 983–990, 2011.
- [14] S. A. A. Shah, M. Bennamoun, F. Boussaid, and A. A. El-Sallam. 3d-div: Novel local surface descriptor for feature matching and pairwise range image registration. In *International Conference on Image Processing (ICIP)*, 2013.
- [15] S. A. A. Shah, M. Bennamoun, F. Boussaid, and A. A. El-Sallam. Automatic object detection using objectness measure. In *1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6, 2013.
- [16] S. A. A. Shah, K. Yahya, G. Mubashar, and A. Bais. Quantification and visualization of mri cartilage of the knee: A simplified approach. In *6th International Conference on Emerging Technologies (ICET)*, pages 175–180, 2010.
- [17] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen, and P. Suetens. Inelastic deformation invariant modal representation for non-rigid 3d object recognition. In F. Perales and R. Fisher, editors, *Articulated Motion and Deformable Objects*, volume 6169 of *Lecture Notes in Computer Science*, pages 162–171. Springer Berlin Heidelberg, 2010.
- [18] Y. Sun and M. Abidi. Surface matching by 3d point’s fingerprint. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 263–269 vol.2, 2001.
- [19] H. Tabia, O. Colot, M. Daoudi, and J.-P. Vandeborre. Non-rigid 3d shape classification using bag-of-feature techniques. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6, 2011.
- [20] F. Tombari, S. Salti, and L. Stefano. Unique signatures of histograms for local surface description. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pages 356–369. Springer Berlin Heidelberg, 2010.