Compact Covariance Descriptors in 3D Point Clouds for Object Recognition

Duc Fehr, Anoop Cherian, Ravishankar Sivalingam, Sam Nickolay, Vassilios Morellas and
Nikolaos Papanikolopoulos
{fehr, cherian, ravi}@cs.umn.edu, sam011235@yahoo.com, {morellas, npapas}@cs.umn.edu
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455

Abstract— **One of the most important tasks for mobile robots is to sense their environment. Further tasks might include the recognition of objects in the surrounding environment. Three dimensional range finders have become the sensors of choice for mapping the environment of a robot. Recognizing objects in point clouds provided by such sensors is a difficult task. The main contribution of this paper is the introduction of a new covariance based point cloud descriptor for such object recognition. Covariance based descriptors have been very successful in image processing. One of the main advantages of these descriptors is their relatively small size. The comparisons between different covariance matrices can also be made very efficient. Experiments with real world and synthetic data will show the superior performance of the covariance descriptors on point clouds compared to state-of-the-art methods.**

I. Introduction

One of the biggest challenges in mobile robotics is the navigation in an unknown environment. In order to achieve this task, different types of sensors have been and are currently used by various robotic systems. For instance, during the DARPA Grand Challenge and Urban Challenge, laser range finders have been used extensively onboard the different autonomous vehicles in order to build a 3D map of the immediate environment. The recent release of inexpensive range sensors has had a very strong impact on the research community in that now a mobile robot can be equipped with a range measuring device that can map its environment. In certain cases, just seeing and mapping obstacles and trying to avoid them is not enough. In a domain such as domotics [1], that aspires to automate people's homes, it becomes important for a robot to actually be able to recognize what objects are in its environment.

Three dimensional point clouds provide a very important cue for analyzing environments or objects. Other than the mobile robotics area they have been used in topographical mapping [2], in which an airplane or a satellite passes over an area and takes several snapshots with a laser range finder, which is then used to build a height map of the scanned area. Very dense point clouds of models are used for CAD purposes [3], as well as for very accurate reconstructions of historic monuments [4].

In this paper, we introduce a novel three dimensional point cloud descriptor based on covariances for object recognition in a three dimensional point cloud. We will present promising results on real world data as well as synthetic data compared to spin images. In the next section some of the related work on point clouds will be presented. Then we will give a brief description of spin images that are used to compare our results and introduce the covariance based descriptor. After that we will show some of the results and finally we will draw some conclusions.

II. Related Work

The most important area of point cloud processing for mobile robotics is the effective detection of objects in the point cloud. Johnson *et al.* [5] developed a local feature descriptor called spin images. These are local descriptors built at an oriented point, which is a point combined with its surface normal. Each surface of different curvature, has a distinct spin image which is used as a feature to differentiate two surfaces. Carmichael *et al.* [6] extends this work to account for the distribution of points available at different depth. Dinh and Kropac [7] use a multiresolution approach in which each point is characterized by a pyramid of spin images. Assfalg *et al.* [8] use spin image signatures in which the spin images are divided into sectors and the signatures correspond to the histogram of points according to these sectors. Li *et al.* [9] combine the spin image descriptors with a dictionary learning approach in order to quickly compare parts of a CAD model even with only partial information. Frome *et al.* [10] describe a different feature descriptor that extends the idea introduced by Belongie *et al.* [11] for images. A sphere is built around the target point and the number of points in each voxel of the sphere is considered. This sphere with the voxels as accumulators is then the feature descriptor. Their experimental results do not seem to provide any significant improvement over the results using spin images. A survey of other descriptors for 3D point cloud data is available in Tangelder and Veltkamp [12]. More recent work has focused on building descriptors for surface matching. For instance Rusu *et al.* [13] use a histogram based approach to describe local surfaces. Tombari *et al.* [14] go a step further in that they combine histograms with signatures in order to build their descriptor.

Similar to Frome *et al.* [10], the descriptor we are introducing in this paper merges ideas from image processing with the spin image technique. Specifically, we explore the

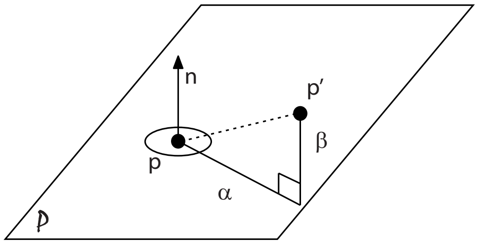


Fig. 1. This figure shows how the different points vote in the accumulator to build the feature descriptor. The normal at each point p is used to find the values α and β, which determine in what bin the point belongs.

possibility of using information fusion from various discriminative image features through a covariance mapping introduced by Tuzel *et al.* [15]. A revised version by Sivalingam *et al.* [16], [17] is used in conjunction with properties of 3D point clouds as used in the construction of spin images, especially properties on the surface curvature, oriented point descriptors, etc.

III. 3D Feature Descriptor

A. Spin Images

Johnson *et al.* introduced spin images in [5], [18], [19]. A spin image, which is essentially an accumulator matrix, is computed at each point of the scan. The normal n to the estimated tangential plane P at each point p is computed. A point-normal pair is called by Johnson an "oriented point." The norm of the projection of the vector pp' onto the normal n and onto the plane P are respectively denoted, β and α (Fig. 1). The accumulator is parameterized by the pair (β, α) and accumulates the votes casted by points neighboring to p, thus forming the spin image.

There are several parameters that need to be chosen for the success of spin images:

1) the number of bins,
2) the size of the bins,
3) the support angle.

These parameters are described by Johnson *et al.* in [19] in detail. The most important parameter is the bin size, because this parameter controls the resolution of the accumulator. If the bin size is too big, too many points will be binned together and the differentiation will be difficult. If it is too small, the spin images are prone to noise. Johnson *et al.* suggest using the average length of the edges from the Delaunay triangulation of the data points as the bin size. The work by Dinh and Kropac [7] circumvents this problem by building spin image pyramids for different resolutions.

A criterion being commonly used to judge the closeness of two spin images is the Pearson's correlation coefficient. Spin images corresponding to the same object surface will be linearly correlated [19] and thus will have a high correlation coefficient. The query spin image from the scene is computed through the same procedure as the database spin images: the normal to the tangential plane at point p is computed and the different points p' around p vote into the accumulator.

B. Covariance Descriptor

Covariance Descriptors are essentially the covariance matrices of feature vectors from the data. These descriptors can be defined as follows:

Definition 1: Let $X_i \in \mathbb{R}^p$, for $i = 1, 2, \cdots, N$, be the feature vectors from the region of interest of a depth image, then the Covariance Descriptor of this region $C \in \mathcal{S}^p_{++}$ is defined as:

$$C = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \mu_X)(X_i - \mu_X)^T \tag{1}$$

where μ_X is the mean feature vector and $\mathcal{S}^p_{++}$ is the space of $p \times p$ Symmetric Positive Definite (SPD) matrices.

The applicability of a covariance matrix for describing object appearances was introduced in [15], [20] for the task of tracking people appearances in surveillance videos. Since covariance descriptors provide a compact low dimensional representation of the data, and at the same time being robust to noise and affine transformations[1], they have found very successful applications in many other areas of computer vision like emotion classification [21], action recognition [22] to name a few.

While the discriminative power of a histogram based descriptor, like the spin images, is dependent on the bin size of its parameters, the representational power of covariance matrices arise from their inherent structure that captures the correlation between the dimensions of the feature vectors. Thus, although, covariances help us get rid of the parameters to be tuned as in spin images, their inherent structure pose significant computational difficulties; these matrices do not adhere to the Euclidean geometry, but span a Riemannian space with an associated Riemannian distance metric. One such metric is the Geodesic distance [23] defined as follows:

$$d_{GD}(X, Y) = \sqrt{\sum_{i=1}^{n} \log^2 \left(\lambda_i(X, Y)\right)} \tag{2}$$

for $X, Y \in \mathcal{S}^n_{++}$ and $\lambda_i(X, Y)$ corresponds to the generalized eigenvalues of X and Y. This metric is affine invariant and has been the most widely used Riemannian metric over covariance descriptors. Unfortunately, it is computationally expensive to use this metric and thus many of the recent papers propose the log-Euclidean Riemannian metric [24] which has the following form:

$$d_{LE}(X, Y) = \|\log(X) - \log(Y)\|_F \tag{3}$$

where $\log$ is the matrix logarithm operator. Since this metric decouples the two matrices involved in the distance computation as in Eq. (2), and since the matrix logarithms can be computed offline, this metric is faster to compute than the geodesic distance. This metric is thus used in this article to determine the distance between two descriptors.

[1] This is actually a property of the metric used in these descriptors.

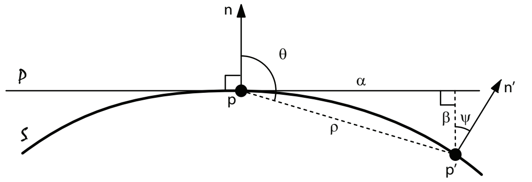


Fig. 2. This figure shows the different features that are used in the covariance descriptor. The features are computed similarly to the spin images by measuring the distances in the plane defined by the normal n at point p and vector pp'. α, β, and θ are defined the same way as previously. ρ is the norm of vector pp', ψ is the angle between the normals. P is the tangent plane at p.

C. Covariances on 3D Point Data

Our motivation for the investigation of covariance descriptors for 3D point data came from their compactness and their flexibility of including multiple features for better discriminative power. The challenge is to define features that are suitable for use with three dimensional point clouds. That is, rather than confining to the elevation distance and the radial distance as in a spin image, covariance matrices can use many other features from the data like the surface normal, angles that neighbors make with the normal, etc. In addition, one can even use the RGB colors associated with the point cloud from a sensor like the Kinect for better discrimination, although we do not explore this idea in the current paper, it is a straightforward extension of our approach.

Inspired by the success of 3D recognition using geometric features as in spin images, in this paper we too approach the feature selection problem from a geometric perspective, deciding on features that could capture the geometric relation between a given point and its neighbors. Towards this end, we started building our feature set with the α and β values as in the spin images, followed by the normed-distance relations between the given point and its neighbors. We also included features corresponding to the surface normals, the motivation being that covariance computed from the surface normals indirectly capture the curvature (hessian) of the surface manifold at that point as suggested in [25]. Curvatures of surfaces hold one of the most reliable cues for object recognition and has been shown to work well for shape recognition in papers such as [11] and [26].

The features that we are using in this paper are described in Fig. 2. Let p be the point for which we compute the covariance and let p' be a point in the neighborhood of p whose features are used in the computation of the covariance. P is the tangent plane at p characterized by normal n. Let us recap the definition of the four initial features. α is the L2 norm of the projection of pp' into the plane P, β is the L2 norm of the projection of pp' onto n, θ defines the angle between vector pp' and the normal n and ρ the L2 norm of vector pp'. These four features get complemented by ψ, the angle between the normals at points p, p' and n' (the normals at points p'). Thus we have access to a total of eight features. The computation of these features is as follows:

$$\alpha = \left\| pp' - (pp' \cdot n)\, n \right\|_2 \tag{4}$$
$$\beta = pp' \cdot n \tag{5}$$
$$\theta = \arccos\left(\frac{pp' \cdot n}{\|pp'\|_2}\right) \tag{6}$$
$$\rho = \|pp'\|_2 \tag{7}$$
$$\psi = \arccos\,(n \cdot n') \tag{8}$$
$$n' = \left[n'_x\, n'_y\, n'_z\right]^T \tag{9}$$

The intuition behind using α and β in the feature vector is inspired by the success of spin images, while the angle θ is used to capture the association of the point with the surface normal. The ρ term complements the angle. Intuitively one set of feature describes Cartesian coordinates while the other set describes cylindrical coordinates. This non-linear relationship, although inherent in the details of α and β, the covariance construction actually computes the distortion from a linearization of this relationship, a cue that we found to provide a good information about the surface structure.

In the following we will present results for different combinations of the features:

$$F_4 = [\alpha, \beta, \theta, \rho]^T \tag{10}$$
$$F_5 = [\alpha, \beta, \theta, \rho, \psi]^T \tag{11}$$
$$F_8 = \left[\alpha, \beta, \theta, \rho, \psi, n'_x, n'_y, n'_z\right]^T \tag{12}$$
$$F_3 = \left[n'_x, n'_y, n'_z\right]^T. \tag{13}$$

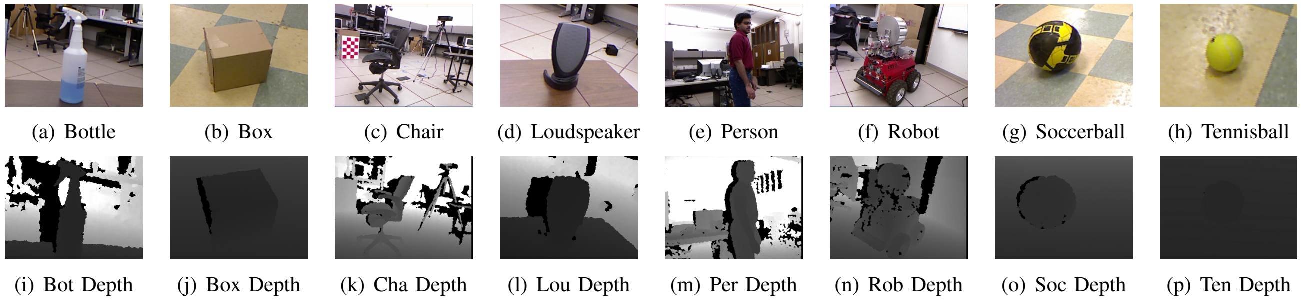
The memory requirements between histogram based descriptors like the spin images and covariance based descriptors compare favorably for our descriptor. In fact, let n be the number of bins in the histogram. Let p be the number of features used in the descriptors. The space complexity is $\frac{p(p+1)}{2}$ for the covariance descriptor and n^p for the spin images. In our case, the vectorized associated covariance matrices are 6, 10, 15, and 36 dimensional vectors. When compared to the high dimensional vector of spin images (400 when the spin image is 20x20), with the same representational power, there is a significant memory gain over the spin images.

IV. Experiments and Results

A. Real world

a) Setup: We are using the Microsoft Kinect to capture point clouds. Our experimental setup is designed towards creating a real world 3D object recognition scenario with arbitrary objects found in our lab. Thus, we collected data from eight different classes of arbitrary objects, the class being: bottle, box, chair, loudspeaker, person, robot, soccerball and tennisball. Sample images and the respective range data for each class are shown in Fig. 3.

b) Data Creation: Depth data is collected for each of these objects using the Kinect sensor. In order to have only points from the actual object in our database, we segment manually the object from each depth map, and then point clouds are generated. The coordinates of the points are given in the camera frame of reference. The result at each step is


(a) Bottle (b) Box (c) Chair (d) Loudspeaker (e) Person (f) Robot (g) Soccerball (h) Tennisball

(i) Bot Depth (j) Box Depth (k) Cha Depth (l) Lou Depth (m) Per Depth (n) Rob Depth (o) Soc Depth (p) Ten Depth

Fig. 3. These figures show different objects we are considering and we are taking depth images of. The top row gives an RGB view of the objects and the bottom row gives the corresponding depth image.

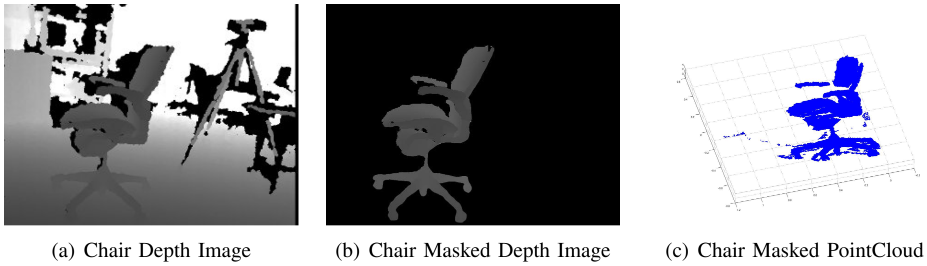

(a) Chair Depth Image (b) Chair Masked Depth Image (c) Chair Masked PointCloud

Fig. 4. These figures show the different steps from the depth image to a point cloud. The reference frame of the point cloud in (c) is off because the points are given in the camera reference frame.

TABLE I

Number of points in object class

Object	Bottle	Box	Chair	Loudspeaker
Number of Points ($\times 10^3$)	27	213	316	38
Object	Person	Robot	Soccerball	Tennisball
Number of Points ($\times 10^3$)	149	463	157	16

given in Fig. 4. The coordinate frame of the point cloud in Fig. 4(c) is oriented differently and hence the point cloud is not nicely aligned with the axes. This is not a problem however since only the relative position of the points to each other is important.

Once we have the point clouds for all the objects we estimate the normals at each point p with the same method that is used in [19]. The 3D surface passing through a set of depth points is computed using a simple least squares approach. Table I lists the number of points collected from each object in the class for different views.

c) Spin Images: The spin images were built with an arbitrary size of 20×20. The support angle was arbitrarily set to $\pi/3$ and the bin size was taken as 0.01. This value was computed as the mean distance of adjacent points. We ran two different sets of experiments in order to check the recall rate on both, the spin images and our covariance descriptor. For memory reasons when we search the k nearest neighbors, instead of saving all the distances for each class, we save only the k best for each class. Once we have the k best for each class we sort them and the k best sorted ones are used for the classification.

d) Results: For this experiment, we take 500 random sample points out of each class and query these against an increasing amount of randomly chosen descriptors from the classes.

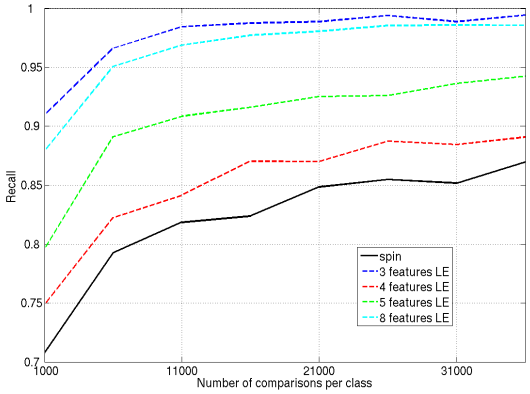


Fig. 5. This figure shows the results for the different feature vectors. The black solid line corresponds to the results of the spin image. The dashed lines represent the results using the log euclidean distance (LE) for different feature vectors.

The results for four different feature vectors are given in Fig. 5. These vectors differ in the number of features used. The vectors F_i , $i \in \{3, 4, 5, 8\}$, introduced in Section III-B, show which features were used to compute the covariances. For comparison, the spin image precision is also included. As it is clear from the figure, the covariance based descriptor performs better than the spin images. The trend in Fig. 5 is for the most part expected. The more features we use, the more discriminative our feature descriptor becomes. What is interesting however is that the results for the covariance when we are only using the normals are better than when we use the full set of 8 features. We will come back to this and

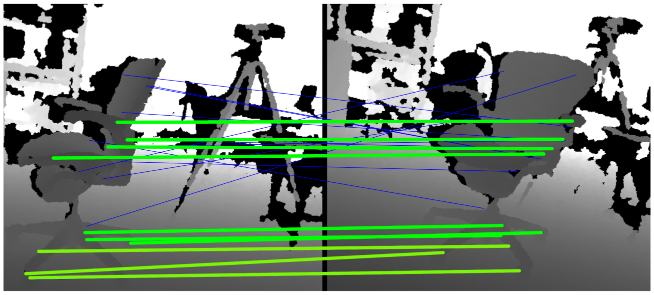


Fig. 6. This figure shows two different images from the same class and the matches between 20 descriptors from the left image with 20 descriptors in the right image. The correct matches are highlighted in green and the incorrect ones in blue.

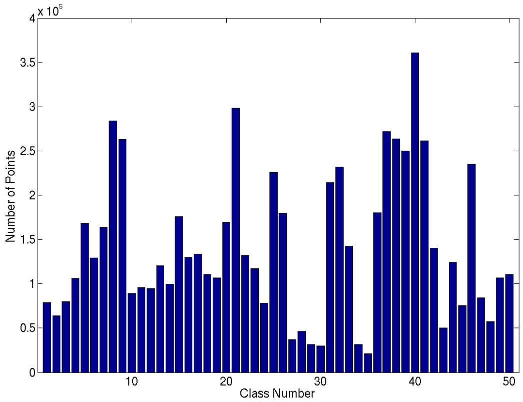


Fig. 7. This figure shows the number of points per class.

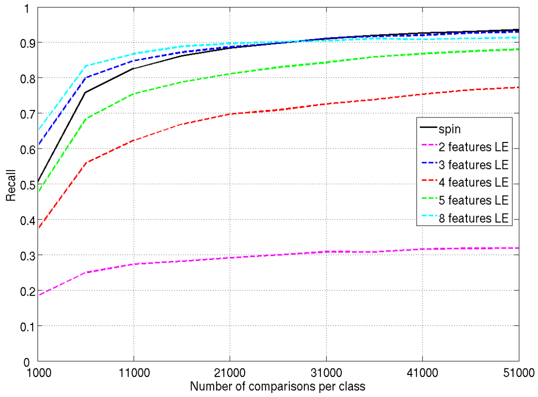


Fig. 8. This figure shows the results for the different feature vectors. The black solid line corresponds to the results of the spin image. The dashed lines represent the results using the log Euclidean distance (LE). 500 queries per class are compared against an increasing number of samples per class.

try to give a qualitative answer to this behavior in Section V.

e) Inter Class Matching: In order to check how invariant our covariance descriptor is to a changing point of view, we compared 20 descriptors from one image matched to all the descriptors of another image in the *chair* class. For this experiment we used as feature vector the $F_4 = [\alpha, \beta, \theta, \rho]$. Qualitative results from this experiment are shown in Fig. 6. As it can be seen, the descriptors produce the correct matches most of the time.

B. Synthetic data setup

a) Setup: In order to validate our finds we ran identical tests on the SHREC 2011 dataset [27]. This dataset is part of the sixth 3D shape retrieval contest and consists of 1000 point clouds classified into 50 classes, 20 clouds per class. These are synthetic and furthermore a surface triangulation is provided which exactly connects the points with a mesh. For this experiment, we are using this mesh to compute the normals at the points. The number of points per class is given in Fig. 7.

b) Results: We are using the same feature vectors given in Eqs. (10)–(13). The result for a two feature vector has also been added in this figure. This vector contains α and β, the two features used in the spin images. In essence the covariance on these features captures the covariance of the spin image. The results are given in Fig. 8. In this figure 500 queries from each class are compared against an increasing datasize.

The colored dashed lines give the recall rates for the covariance descriptors. The solid black line gives the results for the spin images. These perform about the same as the covariance descriptors. Also note that in this case, the three feature (F_3) vector does not outperform the eight feature (F_8) vector, but it is very slightly superior when a large sample size is used for comparison. For all intents and purposes, both feature vectors have the same performance. The three feature covariance provides better results than the four and five feature vector (F_4, F_5) however. We will discuss this in more detail in the Section V. The two feature covariance performs very poorly as expected, since it provides only three values to characterize each point, which contains comparably less information than for instance the ten values represented by the 4×4 covariance matrix and are thus not discriminative enough.

V. Discussion

The experiments conducted produced a very interesting result in that we were able to show that in the case of the real world data captured with the Kinect, the three normal features perform better than the eight features combined (Fig. 5). In the case of the SHREC database however both descriptors perform similarly (Fig. 8). This might be derived from the fact that the Kinect provides noisy data, whereas the SHREC database is clean noiseless data. This suggests that the information carried in the normals is more important than the relative position of the points in the neighborhood. Furthermore it points to the fact that some features are more suitable than others for the differentiation of covariance descriptors.

VI. Conclusion

This work presents a starting point to the exploration of covariance based descriptors for point correspondences in 3D point clouds. We discussed several significant advantages in using covariances against other descriptors, which we summarize here: (i) they are compact and low dimensional, which implies lower computational and storage requirements, (ii) no need to tune any model parameters like the bin sizes as in spin images, or mesh manipulation, (iii) the inherent ability of the covariances to subtract out common features available in the data and consider only the discriminative information (for example, if a region is planar, then the covariance might be rank deficient, which might be useful at times to prune out non-informative regions from the point clouds), and (iv) finally, covariances provide scalability in terms of the various features to be considered for recognition. Going forward, we would like to look at the possibility of improving the recognition using image cues along with the depth data. Furthermore we are working on a theoretical description of the covariances for point clouds which hopefully will give us insight into why the normals, as features, provide good results in the presence of noise.

Acknowledgements

This material is based upon work supported in part by the U. S. Army Research Laboratory, the U. S. Army Research Office under contract number #911NF-08-1-0463 (Proposal 55111-CI) and the National Science Foundation through grants #IIP-0726109, #CNS-0708344, #CNS-0821474, #IIP-0934327, #IIP-1032018, #CNS-1061489, #SMA-1028076, #CNS-1039741, and #CNS-1138020.

References

[1] D. Bonino and F. Corno, "Dogont - ontology modeling for intelligent domotic environments," in *The Semantic Web - ISWC 2008*, ser. Lecture Notes in Computer Science, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, Eds. Springer Berlin / Heidelberg, 2008, vol. 5318, pp. 790–803.

[2] E. Schwalbe, H. Maas, and F. Seidel, "3D building model generation from airborne laser scanner data using 2D gis data and orthogonal point cloud projections," in *Proceedings of the International Society for Photogrammetry and Remote Sensing WG III/3, III/4*, vol. 3, 2005, pp. 12–14.

[3] C. Ip and S. Gupta, "Retrieving matching CAD models by using partial 3D point clouds," *Computer Aided Design and Applications*, vol. 4, no. 5, pp. 629–638, 2007.

[4] P. Allen, I. Stamos, A. Troccoli, B. Smith, M. Leordeanu, and Y. Hsu, "3D modeling of historic sites using range and image data," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA*, vol. 1, 2003, pp. 145–150.

[5] A. Johnson, "Spin-images: A representation for 3D surface matching," Ph.D. dissertation, Carnegie Mellon University, 1997.

[6] O. Carmichael, D. Huber, and M. Hebert, "Large data sets and confusing scenes in 3D surface matching and recognition," in *Proceedings of the Second International Conference on 3D Digital Imaging and Modeling*, Oct 1999, pp. 358 – 367.

[7] H. Q. Dinh and S. Kropac, "Multi-resolution spin-images," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 863–870.

[8] J. Assfalg, M. Bertini, A. Bimbo, and P. Pala, "Content-based retrieval of 3D objects using spin image signatures," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 589–599, 2007.

[9] X. Li, A. Godil, and A. Wagan, "3D part identification based on local shape descriptors," in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, 2008, pp. 162–166.

[10] A. Frome, D. Huber, R. Kolluri, T. Blow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Computer Vision - ECCV 2004*, vol. 3023, 2004, pp. 224–237.

[11] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 509–522, 2002.

[12] J. W. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *International Conference on Shape Modeling and Applications*, vol. 0, pp. 145–156, 2004.

[13] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *Proceedings of the 2009 IEEE international conference on Robotics and Automation (ICRA2009)*, 2009, pp. 1848–1853.

[14] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*. Springer-Verlag, 2010, pp. 356–369.

[15] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Computer Vision ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin / Heidelberg, 2006, vol. 3952, pp. 589–600.

[16] R. Sivalingam, V. Morellas, D. Boley, and N. Papanikolopoulos, "Metric learning for semi-supervised clustering of region covariance descriptors," in *Third ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC 2009.*, 2009, pp. 1–8.

[17] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos, "Tensor sparse coding for region covariances," in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin / Heidelberg, 2010, vol. 6316, pp. 722–735.

[18] A. Johnson and M. Hebert, "Surface matching for object recognition in complex three-dimensional scenes," *Image and Vision Computing*, vol. 16, no. 9, pp. 635–651, 1998.

[19] ——, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 51, no. 5, pp. 433–449, May 1999.

[20] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 728–735.

[21] W. Zheng, H. Tang, Z. Lin, and T. Huang, "Emotion recognition from arbitrary view facial images," in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin / Heidelberg, 2010, vol. 6316, pp. 490–503.

[22] C. Yuan, W. Hu, X. Li, S. Maybank, and G. Luo, "Human action recognition under log-euclidean riemannian metric," in *Computer Vision ACCV 2009*, ser. Lecture Notes in Computer Science, H. Zha, R.-i. Taniguchi, and S. Maybank, Eds. Springer Berlin / Heidelberg, 2010, vol. 5994, pp. 343–353.

[23] W. Forstner and B. Moonen, "A metric for covariance matrices," *Qua vadis geodesia*, pp. 113–128, 1999.

[24] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic resonance in medicine*, vol. 56, no. 2, pp. 411–421, 2006.

[25] R. Wilson and E. Hancock, "Refining surface curvature with relaxation labeling," in *Image Analysis and Processing*, ser. Lecture Notes in Computer Science, A. Del Bimbo, Ed. Springer Berlin / Heidelberg, 1997, vol. 1310, pp. 150–157.

[26] G. Dudek and J. Tsotsos, "Shape representation and recognition from multiscale curvature," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 170–189, 1997.

[27] SHREC, 2011. [Online]. Available: http://www.itl.nist.gov/iad/vug/sharp/contest/2011/RangeScans/