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.

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.

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.

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
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, \( \beta \) and \( \alpha \) (Fig. 1). The accumulator is parameterized by the pair \((\beta, \alpha)\) 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 \( i = 1, 2, \ldots, N \), be the feature vectors from the region of interest of a depth image, then the Covariance Descriptor of this region \( C \in S_{++}^p \) is defined as:

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

(1)

where \( \mu_X \) is the mean feature vector and \( S_{++}^p \) 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:

\[
d_{GD}(X,Y) = \sqrt{\sum_{i=1}^{n} \log^2(\lambda_i(X,Y))}
\]  

(2)

for \( X, Y \in S_{++}^n \) 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:

\[
d_{LE}(X,Y) = \| \log(X) - \log(Y) \|_F
\]  

(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.
The computation of these features is as follows:

\[
\begin{align*}
\alpha &= \|pp' - (pp' \cdot n) n\|_2 \\
\beta &= pp' \cdot n \\
\theta &= \arccos \left( \frac{pp' \cdot n}{\|pp'\|_2} \right) \\
\rho &= \|pp'\|_2 \\
\psi &= \arccos (n \cdot n') \\
n' &= [n'_x, n'_y, n'_z]^T
\end{align*}
\]

The intuition behind using \(\alpha\) and \(\beta\) in the feature vector is inspired by the success of spin images, while the angle \(\theta\) is used to capture the association of the point with the surface normal. The \(\rho\) 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 \(\alpha\) and \(\beta\), 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:

\[
\begin{align*}
F_4 &= [\alpha, \beta, \theta, \rho]^T \\
F_5 &= [\alpha, \beta, \theta, \rho, \psi]^T \\
F_6 &= [\alpha, \beta, \theta, \rho, \psi, n'_x, n'_y, n'_z]^T \\
F_3 &= [n'_x, n'_y, n'_z]^T
\end{align*}
\]

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 \(\frac{n(n+1)}{2}\) 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 arbitary 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
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.

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

<table>
<thead>
<tr>
<th>Object</th>
<th>Bottle</th>
<th>Box</th>
<th>Chair</th>
<th>Loudspeaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Points (×10^3)</td>
<td>27</td>
<td>213</td>
<td>316</td>
<td>38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Person</th>
<th>Robot</th>
<th>Soccerball</th>
<th>Tennisball</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Points (×10^3)</td>
<td>149</td>
<td>463</td>
<td>157</td>
<td>16</td>
</tr>
</tbody>
</table>

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.

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
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 \( \alpha \) and \( \beta \), 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 number of samples per class.

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


