Point Cloud Matching based on 3D Self-Similarity

Jing Huang
University of Southern California
Los Angeles, CA 90089
huang10@usc.edu

Suya You
University of Southern California
Los Angeles, CA 90089
suya.you@usc.edu

Abstract

Point cloud is one of the primitive representations of 3D data nowadays. Despite that much work has been done in 2D image matching, matching 3D points achieved from different perspective or at different time remains to be a challenging problem. This paper proposes a 3D local descriptor based on 3D self-similarities. We not only extend the concept of 2D self-similarity [1] to the 3D space, but also establish the similarity measurement based on the combination of geometric and photometric information. The matching process is fully automatic i.e. needs no manually selected land marks. The results on the LiDAR and model data sets show that our method has robust performance on 3D data under various transformations and noises.

1. Introduction

Matching is a process of establishing precise correspondences between two or more datasets acquired, for example, at different times, from different aspects or even from different sensors or platforms. This paper addresses the challenging problem of finding precise point-to-point correspondences between two 3D point cloud data. This is a key step for many tasks including multi-view scans registration, data fusion, 3D modeling, 3D object recognition and 3D data retrieval.

Figure 1 shows an example of point cloud data that were acquired by airborne LiDAR sensor. The two point clouds represent two LiDAR scans of the same area (downtown Vancouver) acquired at different times and from different viewpoints. The goal is to find precise matches for the points in the overlapping areas of the two point clouds.

Matching of point clouds is challenging in that there are usually enormous number of 3D points and the coordinate system can vary in terms of translation, 3D-rotation and scale. Point positions are generally not coincident; noises and occlusions are common due to incomplete scans; and objects are attached to each other and/or the ground. Furthermore, many data may not contain any photometric information such as intensity other than point positions.

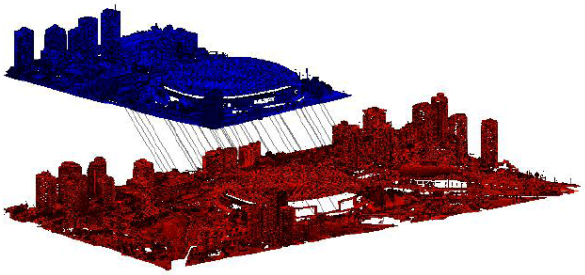


Figure 1. Two point clouds represent two LiDAR scans of the same area captured at different times and from different aspects. The proposed method can find precise matches for the points in the overlapping area of the two point clouds.

Given the problems above, matching methods that solely rely on photometric properties will fail and conventional techniques or simple extensions of 2D methods are no longer feasible. The unique nature of point clouds requires methods and strategies different from those for 2D images.

In real applications, most point clouds are a set of geometric points representing external surfaces or shapes of 3D objects. We therefore treat the geometry as the essential information. We need a powerful descriptor as a way to capture geometric arrangements of points, surfaces and objects. The descriptor should be invariant to translation, scaling and rotation. In addition, the high dimensional structure of 3D points must be collapsed into something manageable.

This paper presents a novel technique specifically designed for matching of 3D point clouds. Particularly, our approach is based on the concept of self-similarity. Self-similarity is an attractive image property that has recently found its way in matching in the form of local self-similarity descriptors [1]. It captures the internal geometric layout of local patterns in a level of abstraction. Locations in image with self-similarity structure of a local pattern are distinguishable from locations in their neighbors, which can greatly facilitate matching process. Several works have demonstrated the value of self-similarity for image matching and related applications [14] [15]. From a totally new perspective, we design a descriptor that can efficiently cap-

ture distinctive geometric signatures embedded in point clouds. The resulting 3D self-similarity descriptor is compact and view/scale-independent, and hence can produce highly efficient feature representation. We apply the developed descriptor to build a complete feature-based matching system for high performance matching between point clouds.

2. Related Work

2.1. 3D Matching

3D data matching has recently been widely addressed in both computer vision and graphic communities. A variety of methods have been proposed, but the approaches based on local feature descriptors demonstrate superior performance in terms of accuracy and robustness [3] [7]. In local feature-based approach, the original data are transformed into a set of distinctive local features, each representing a quasi-independent salient region within the scene. The features are then characterized with robust descriptors containing local surface properties that are supposedly repeatable and distinctive for matching. Finally, registration methods such as the famous Iterative Closest Point (ICP) [5], as well as its variants, could be employed to figure out the global arrangement.

Spin image is a well-known feature descriptor for 3D surface representation and matching [3]. One key element of spin image generation is the *oriented point*, or 3D surface point with an associated direction. Once the oriented point is defined, the surrounding cylindrical region is compressed to generate the spin image as the 2D histogram of number of points lying in different distance grids. By using local object-oriented coordinate system, the spin image descriptor is view and scale independent. Several variations of spin image have been suggested. For example, [18] proposed a spherical spin image for 3D object recognition, which can capture the equivalence classes of spin images derived from linear correlation coefficients.

Heat Kernel Signature (HKS) proposed in [7] is a type of shape descriptor targeting for matching objects under non-rigid transformation. The idea of HKS is to make use of the heat diffusion process on the shape to generate intrinsic local geometry descriptor. It is shown that HKS can capture much of the information contained in the heat kernel and characterize the shapes up to isometry. Further, [8] improved HKS to achieve scale-invariant, and developed a HKS local descriptor that can be used in the bag-of-features framework for shape retrieval in the presence of a variety of non-rigid transformations.

Many works also attempt to generalize from 2D to 3D, such as 3D SURF [6] extended from SURF [4] and 3D Shape context [19] extended from 2D Shape Context [2].

A detailed performance evaluation and benchmark on 3D shape matching were reported in [11] that simulates the feature detection, description and matching stages of feature-based matching and recognition algorithms. The benchmark tests the performance of shape feature detectors and descriptors under a wide variety of conditions. We also use the benchmark to test and evaluate our proposed approach.

Recently, the concept of self-similarity has drawn much attention and been successfully applied for image matching and object recognition. Shechtman and Irani [1] proposed the first algorithm that explicitly employs self-similarity to form a descriptor for image matching. They used the intensity correlation computed in local region as resemblance to generate the local descriptor. Later on, several extensions and varieties were proposed. For example, Chatfield *et al*. [14] combined a local self-similarity descriptor with the bag-of-words framework for image retrieval of deformable shape classes; Maver [15] used the local self-similarity measurement for interest point detection; Huang *et al*. [17] proposed a 2D self-similarity descriptor for multimodal image matching, with different definitions of self-similarity evaluated.

This paper extend the self-similarity framework to matching of 3D point clouds. We develop a new 3D local feature descriptor that can efficiently characterizes distinctive signatures of surfaces embedded in point clouds, hence can produce high performance matching. To the best of our knowledge, we are the first one to introduce the self-similarity to the area of point cloud matching.

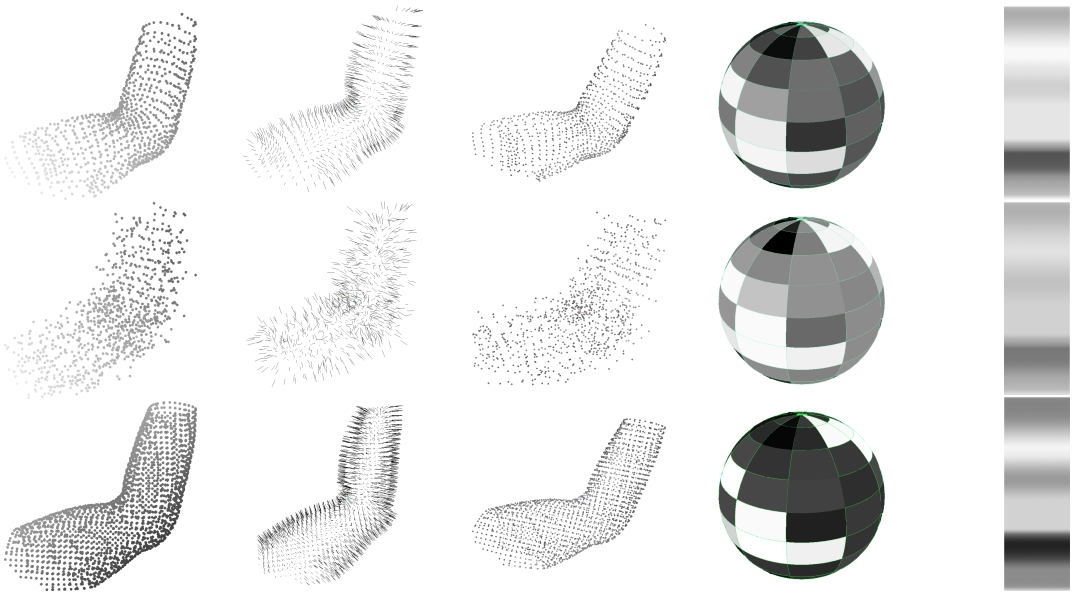
The remainder of the paper describes the details of our proposed approach and implementations. We also present the results of our analysis and experiments.

3. Point clouds and self-similarity

Originated from fractals and topological geometry, self-similarity is the property held by those parts of a data or object that resemble themselves in comparison to other parts of the data. The resemblance can be photometric properties, geometric properties or their combinations.

Photometric properties such as color, intensity or texture are useful and necessary to measure the similarity between imagery data, however, they are no longer as reliable on point cloud data. In many situations, the data may only contain point positions without any photometric information. Therefore, geometric properties such as surface normals and curvatures are treated as the essential information for point cloud data.

Particularly, we found that surface normal is the most effective geometric property that enables human visual perception to distinguish local surfaces or shapes in point clouds. Normal similarity has shown sufficiently robust to a wide range of variations that occur within disparate object classes. Furthermore, a point and its normal vector can form a simple local coordinate system that can be used to



(a) (b) (c) (d) (e)

Figure 2. Illustration for self-similarities. Column (a) are three point clouds of the same object and (b) are their normal distributions. There are many noises in the 2nd point cloud, which lead to quite different normals from the other two. However, the 2nd point cloud shares similar intensity distribution as the 1st point cloud, which ensures that their self-similarity surface (c), quantized bins (d) and thus descriptors (e) are similar to each other. On the other hand, while the intensity distribution of the 3rd point cloud is different from the other two, it shares similar normals as the 1st point cloud (3rd row vs. 1st row in column (b)), which again ensures that their self-similarity surface, quantized bins and descriptors are similar to each other.

generate view/scale-independent descriptor.

Curvature is another important geometric property that should be considered in similarity measurement. The curvature illustrates the changing rate of tangents. Curved surfaces always have varying normal, yet many natural shapes such as sphere and cylinder preserve the curvature consistency. Therefore, we incorporate the curvature in similarity measurement to characterize local geometry of surface. Since there are many possible directions of curvature in 3D, we consider the direction in which the curvature is maximized, i.e. the principal curvature, to keep its uniqueness.

We also consider the photometric information in our algorithm development to generalize the problem. We assume the case that both the photometric and geometric information are available in the dataset. We propose to use both the properties as similarity measurements and combine them under a unified framework.

4. 3D Self-similarity descriptor

Given an interest point and its local region, there are two major steps to construct the descriptor: (1) generating the self-similarity surface using the defined similarity measurements, and (2) quantizing the self-similarity surface in a rotation-invariant manner. In this work, we consider similarity measurements on surface normal, curvature, and photometric properties. Once the similarity measurements are defined, the local region is converted to self-similarity surface centered at the interest point, with multiple/united property similarity at each point. We can then construct the 3D local self-similarity descriptors to generate signatures of surfaces embedded in the point cloud.

4.1. Generating self-similarity surface

Assume there are property functions f_1, f_2, $\dots f_n$ defined on a point set X, which map any point $x \in X$ to property vectors $f_1(x)$, $f_2(x)$, $\dots f_n(x)$. For 2D images, the property can be intensities, colors, gradients or textures. In our 3D situation, the property set can further include normals and curvatures, besides intensities/colors.

For each property function f that has definition on two points x and y, we can further induce a pointwise similarity function $s(x, y, f)$. Then, the united similarity can be defined as the combination of the similarity functions of all the properties. Figure 2 gives the intuition of how combined self-similarity would work for different data.

4.1.1 Normal similarity

The normal gives the most direct description of the shape information, especially for the surface model. One of the most significant characteristics of the normal distribution is the continuity, which means the normal similarity is usually positively correlated to the distance between the two points. However, any non-trivial shape could disturb the distribu-

Figure 3. Self-similarity surface of normals. The brighter a point is, the more similar its normal is to the normal at the center point.

tion of normals, which gives the descriptive power of the normal similarity.

We use the method described in [13] to extract the normals. Figure 2(b) are examples of normal distributions.

The property function of normal is a 3D function $f_{normal}(x) = \vec{n}(x)$. Assume that the normals are normalized i.e. $\|\vec{n}(x)\| = 1$, we can define the normal similarity between two points x and y as the angle between the normals, as formula 1 suggests.

$$\begin{aligned} s(x, y, f_{normal}) &= [\pi - \cos^{-1}(f_{normal}(x) \cdot f_{normal}(y))]/\pi \\ &= [\pi - \cos^{-1}(\vec{n}(x) \cdot \vec{n}(y))]/\pi. \end{aligned} \tag{1}$$

It's easy to see that when the angle is 0, the function returns 1; whereas the angle is π, i.e. the normals are opposite to each other, the function returns 0.

We should be careful that a locally stable normal estimation method is needed here to ensure that the directions of normals are consistent with each other, because flipping one normal could lead to the opposite result of the function.

Figure 3 shows the visualization of the self-similarity surface of normals of one key point.

4.1.2 Curvature similarity

The curvature illustrates the changing rate of tangents. Curved surfaces always have varying normals, yet many natural shapes such as sphere and cylinder preserve the curvature consistency. Therefore, it's worthwhile to incorporate the curvature information in the measurement of similarity. Since there are infinite possible directions of curvature in 3D, we only consider the direction in which the curvature is maximized, i.e. the principal curvature.

The principal curvature direction can be approximated as the eigen vector corresponding to the largest eigen value of the covariance matrix C of normals projected on the tangent plane. The property function of curvature is defined as a single-value function

$$f_{curv}(x) = \frac{1}{N} \arg\max(\lambda | det(C - \lambda I) = 0), \tag{2}$$

where N is the number of points (normals) taken into account in the neighborhood so that values are scaled to the range from 0 to 1 (In practice this value is typically less than 0.7). We then define the curvature similarity between two points x and y as the absolute difference between them:

$$s(x, y, f_{curv}) = 1 - |f_{curv}(x) - f_{curv}(y)|. \tag{3}$$

Again, the function returns 1 when the curvature values are similar, and returns 0 when they are different.

4.1.3 Photometric similarity

Photometric information is an important clue for our understanding of the world. For example, when we look at a gray image, we can infer the 3D structure through the observation of changes in intensity. Some point clouds, besides point positions, also contain certain photometric information such as intensity or any reflective values generated by sensors. Such information is a combinational result of geometric structure, material, lighting and even shadows. While they are not as generally reliable as geometric information for point clouds, they can be helpful in specific situations and we also incorporated them in the similarity measurement. We try to use the photometric similarity to model this complicated situation as it is invariant to lighting to some extent, given the similar material properties.

In our current framework, the property function of photometry is a single-value function $f_{photometry}(x) = I(x)$ where $I(x)$ is the intensity function. With the range normalized to $[0, 1]$, we can define the photometric similarity between two points x and y as their absolute difference:

$$\begin{aligned} s(x, y, f_{photometry}) &= 1 - |f_{photometry}(x) - f_{photometry}(y)| \\ &= 1 - |I(x) - I(y)|. \end{aligned} \tag{4}$$

4.1.4 United Similarity

Given a set of properties, we need to combine them to measure the united similarity:

$$s(x, y) = \sum_{p \in PropertySet} w_p \cdot s(x, y, f_p). \tag{5}$$

The weights $w_p \in [0, 1]$ can be experimentally determined according to availability and contribution of each considered property. For example, when dealing with point clouds converted from mesh models, we will let $w_{photometry} = 0$ since there's no intensity information in the data. Another example is when we have known that there are many noise points in the data, which makes the curvature estimation unstable, we can reduce its weight accordingly. In general cases, the equal weights or weights of 2:1:1 (with normals dominating) are good enough without prior knowledge. Learning the best weights from different data sets, however, could be an interesting topic.

Once the similarity measurements are defined, construction of self-similarity surface is straightforward. First, the point cloud is converted to 3D positions with the defined properties. Then, the self-similarity surface is constructed by comparing each point's united similarity to that of surrounding points within a local spherical volume. The radius of the sphere is k times the detected scale at which the principal curvature reaches its local maxima. The choice of the size can determine whether the algorithm is performed at a local region or more like a global region. We found by experiments that the performance is the best when $k \approx 4$.

4.2. Forming the Descriptor

Our approach is trying to make full use of all kinds of geometric information on the point cloud, mainly including the normal and curvature, which can be seen as the first-order and the second-order differential quantities. Since we are facing discrete data, certain approximations are needed for calculation. Such approximations have been provided by open-source libraries such as Point Cloud Library (PCL).

The rotation invariance is achieved by using local reference system (Fig. 4) of each given key point: the origin is placed at the key point; the z-axis is the direction of the normal; the x-axis is the direction of the principal curvature; and the y-axis is the cross product of z and x directions.

In order to reduce the dimension as well as bearing small distortion of the data, we quantize the correlation space into bins. In our experiments we have $\#Bin(r) = 6$ radial bins, $\#Bin(\phi) = 8$ bins in longitude ϕ and $\#Bin(\theta) = 6$ bins in latitude θ, and replace the values in each cell with the average similarity value of all points in the cell, resulting in a descriptor of 6*8*6 = 288 dimensions (Fig. 4).

The index of each dimension can be represented by a triple $(Index(r), Index(\phi), Index(\theta))$, ranging from (0,0,0) to (5,7,5). Each index component can be calculated in the following way:

$$\begin{cases} Index(r) = \lfloor \#Bin(r) \cdot \dfrac{r}{scale} \rfloor \\ Index(\phi) = \lfloor \#Bin(\phi) \cdot \dfrac{\phi}{2\pi} \rfloor \\ Index(\theta) = \lfloor \#Bin(\theta) \cdot \dfrac{\theta}{\pi} \rfloor \end{cases} \tag{6}$$

In the final step, the descriptor is normalized by scaling the dimensions with the maximum value to be 1.

5. Point Cloud Matching

We apply the developed descriptor to build a complete feature-based point cloud matching system. Point clouds often contain hundreds of millions of points, yielding a large high dimensional feature space to search, index and match. So selection of the most distinctive and repeatable features for matching is a necessity.

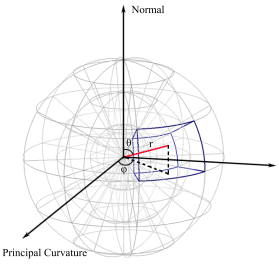


Figure 4. Illustration of the local reference frame and quantization.

5.1. Multi-scale feature detector

Feature, or key point extraction is a necessary step before the calculation of 3D descriptor because (1) 3D data always have too many points to calculate the descriptor on; (2) distinctive and repeatable features will largely enhance the accuracy of matching. There are many feature detection methods evaluated in [9]. Our approach detects salient features with a multi-scale detector, where 3D peaks are detected in both scale-space and spatial-space. Inspired by [10], we propose to extract key points based on the local Maxima of Principle Curvature (MoPC), which provide relatively stable interest regions compared to a range of other interest point detectors. Note that different from [20], where the scale-invariant curvature is measured, we make use of the variation of the curvature to extract the specific scale.

The first step is setting up several layers of different scales. Assume the diameter of the input point cloud is d. We choose one tenth of d as the largest scale, and one sixtieth of d as the minimum scale. The intermediate scales are interpolated so that the ratios between them are constant.

Next, for each point p and scale s, we calculate the principal curvature using points that lie within s units from p. The calculation process is discussed in 4.1.2.

Finally, if the principal curvature value of point p at scale s is larger than the value of the same point p at adjacent scales and the value of all points within one third of s units from p at scale s, meaning that the principal curvature reaches local maxima across both scale and local neighborhood of p, then p is added to the key point set with scale s. Note that the same point could appear in multiple scales.

Figure 5 shows the feature points detected in the model.

5.2. Matching criteria

In case there can be multiple regions (and thus descriptors) that are similar to each other, we follow the Nearest Neighbor Distance Ratio (NNDR) method i.e. matching a key point in cloud X to a key point in cloud Y if and only if

$$\frac{dist(x, y_1)}{dist(x, y_2)} < threshold, \tag{7}$$

where y_1 is the nearest neighbor of x in point cloud Y and y_2 is the 2nd nearest neighbor of x in point cloud Y (in

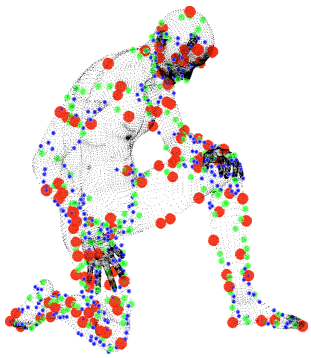


Figure 5. Detected salient features (highlighted) with the proposed multi-scale detector. Different sizes/colors of balls indicate different scales at which the key points are detected. These features turn out to be distinctive, repeatable and compact for matching.

the feature space). Balancing between the number of true positives and false positives, the threshold is typically set to be 0.75 in our experiments.

5.3. Outlier Removal

After a set of local matches are selected, we can perform outlier removal using global constraints. If it's known that there are only rigid body and scale transformations, a 3D RANSAC algorithm is applied to determine the transformation that allow maximum number of matches to fit in. Figure 10 (b) shows the filtered result for Fig. 10 (a). In the future, variants of L_1 norm instead of L_2 norm could be considered as penalty function, which has been proved superior in optical flow methods.

6. Experimental Results and Evaluation

The proposed approach have been extensively tested and evaluated using various datasets including both synthetic data from standard benchmarks and our own datasets, covering a wide variety of objects and conditions. We evaluated the effectiveness of our approach in the terms of distinctiveness, robustness and invariance.

6.1. SHREC data

We use the SHREC feature descriptor benchmark [11] [12] and convert the mesh model to point cloud by keeping only the vertices for our test. This benchmark includes shapes with a variety of transformations such as holes, micro holes, scale, local scale, noise, shot noise, topology, sampling and rasterization (Fig. 6).

Table 1 and 2 show the average normalized L_2 error of SS-Normal descriptors and SS-Curvature descriptors at corresponding points detected by MoPC. Note that only the transformations with clear ground truth are compared here i.e. the isometry shape is used as the comparing template. Given the set of detected features $F(X)$ and $F(Y)$, the descriptor quality is the calculated at corresponding points x_k

Transform.	**Strength**				
	1	≤ 2	≤ 3	≤ 4	≤ 5
Holes	0.11	0.22	0.33	0.45	0.52
Local scale	0.40	0.58	0.67	0.77	0.83
Sampling	0.31	0.46	0.57	0.67	0.81
Noise	0.58	0.65	0.70	0.74	0.77
Shot noise	0.13	0.23	0.28	0.32	0.35
Average	0.34	0.45	0.53	0.60	0.67

Table 1. Robustness of Normal Self-Similarity descriptor based on features detected by MoPC (average L_2 distance between descriptors at corresponding points). Average number of points: 518

Transform.	**Strength**				
	1	≤ 2	≤ 3	≤ 4	≤ 5
Holes	0.10	0.21	0.31	0.42	0.49
Local scale	0.38	0.56	0.65	0.75	0.81
Sampling	0.44	0.55	0.63	0.71	0.87
Noise	0.55	0.62	0.67	0.72	0.75
Shot noise	0.13	0.22	0.27	0.31	0.33
Average	0.32	0.43	0.51	0.58	0.65

Table 2. Robustness of Curvature Self-Similarity descriptor based on features detected by MoPC (average L_2 distance between descriptors at corresponding points). Average number of points: 518

(with descriptor f_k ($k = 1, 2, \ldots, |F(Y)|$) and y_j (with descriptor g_j, $j = 1, 2, \ldots, |F(X)|$):

$$d_{kj} = \frac{||f_k - g_j||_2}{\frac{1}{|F(X)|^2 - |F(X)|} \sum_{k,j \neq k} ||f_k - g_j||_2}, \tag{8}$$

and then sum up using (note that $|F(X)| = |F(Y)|$ when we only consider corresponding points):

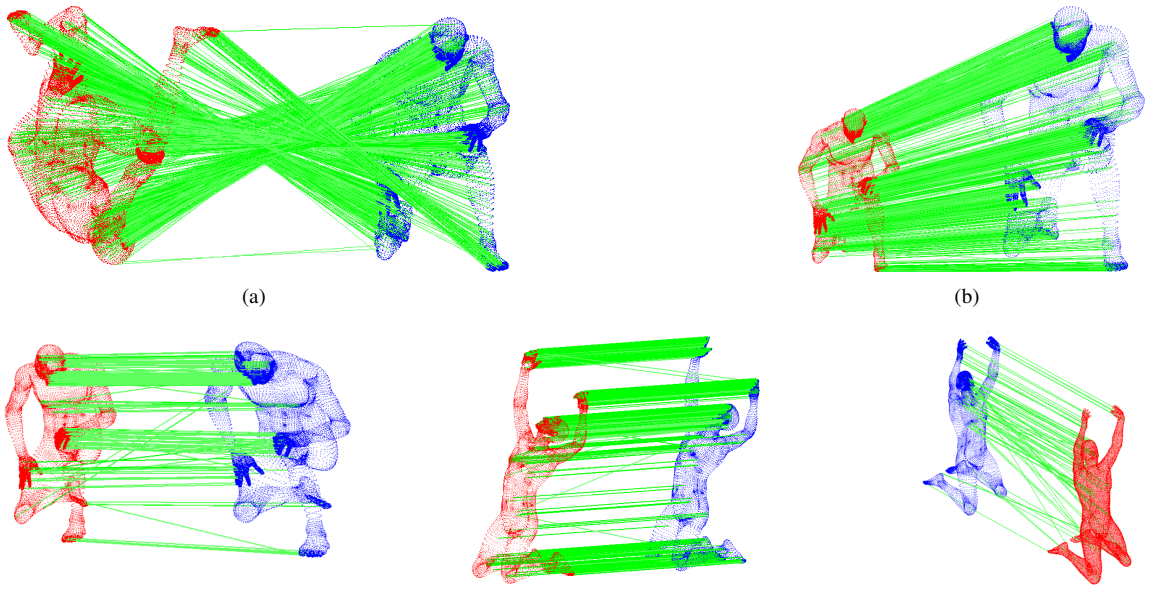
$$Q = \frac{1}{|F(X)|} \sum_{k=1}^{|F(X)|} d_{kj}. \tag{9}$$

The results are competitive to the state-of-the-art methods compared in [11] [12].

For synthetic data under rotational and scale transformations, the proposed feature detector and descriptor can achieve nearly fully correct matches, e.g. Fig. 7 (a) and (b). Figure 7 (c), (d) and (e) show the matching results between human point clouds from SHREC'10 [11] dataset under affine transformation, with holes and after rasterization. The feature extraction and descriptor calculation take about 1-2 minutes on typical data with around 50,000 points.

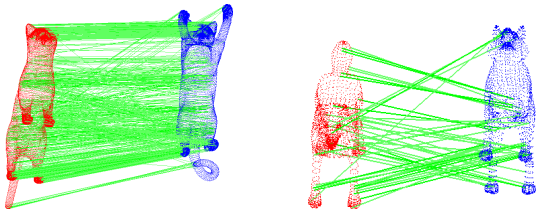
Another set of test data is from TOSCA High-resolution [16]. Figure 8 (a) is a typical matching example using dense self-similarity descriptor and 8 (b) is the matching example using MoPC feature-based self-similarity descriptor. Figure 9 (a) shows the precision-recall curve for the wolf data.

Figure 6. SHREC benchmark dataset. The transformations are (from left to right): isometry, holes, micro holes, scale, local scale, noise, shotnoise, topology and sampling.



(a) (b) (c) (d) (e)

Figure 7. Matching result between human point clouds with rotation, scale, affine transformation, holes and rasterization.



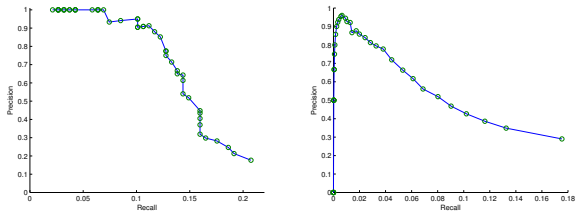
(a) (b)

Figure 8. (a) is the matching result with dense SS descriptor of different poses of a cat. (b) is the matching result with MoPC feature-based SS descriptor of different poses of a wolf.

6.2. Lidar point clouds

Table 3 shows the comparison of different configurations of similarity on 50 pairs of randomly sampled data in randomly chopped 300m * 300m * 50m area of Richard Corridor in Vancouver. The precision is the ratio between the number of correctly identified matches (TP) and all matches (TP + FP). The running time is about 15s per pair.

We also perform experiments on different point clouds of the same region. Figure 9 (b) shows the precision-recall curve for 3D Self-similarity descriptor working on data chopped out from the LiDAR data (600m * 600m * 60m, 100,000 points) of Richard Corridor in Vancouver.

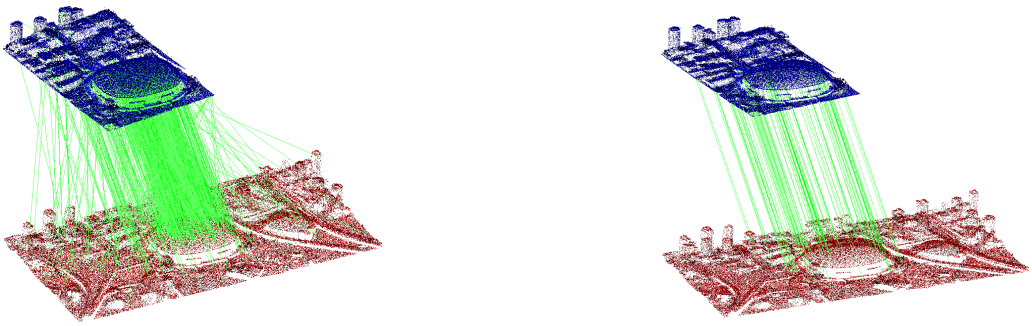


(a) (b)

Figure 9. (a) is the precision-recall curve for the 3D Self-similarity descriptor between two wolf models from TOSCA-HighResolution Data. (b) is the precision-recall curve for the 3D Self-similarity descriptor on Vancouver Richard Corridor data.

Property	Precision
Normal	55%
Curvature	49%
Photometry	49%
Normal+Curvature+Photometry	51%

Table 3. Evaluation of matching results with different configurations (weights) of united self-similarity. Pure normal similarity performs the best overall, but curvature/photometry similarity can do better for specific data.



(a) (b)

Figure 10. Matching result of aerial LiDAR out of two scans of the Richard Corridor area in Vancouver. (b) is the filtered result of (a).

In real applications there are tremendous data that might spread across large scales. Our framework can also deal with large scale data by divide-and-conquer since we only require local information to calculate the descriptors. Figure 10 shows the matching result of aerial LiDAR out of two scans of the Richard Corridor area in Vancouver.

7. Conclusion

In this paper we have extended the 2D self-similarity descriptor to the 3D spatial space. The new feature-based 3D descriptor is invariant to scale and orientation change. The new descriptor achieves competitive results on 3D point cloud data from public dataset TOSCA as well as aerial LiDAR data. Since meshes or surfaces can be sampled and transformed into the point cloud or voxel representations, the method can be easily adapted to the matching models to point clouds, or models to models. We are currently working on the matching propagation and shape retrieval scheme based on our descriptor.

References

[1] E. Shechtman and M. Irani. Matching Local Self-Similarities across Images and Videos. In Proc. CVPR, 2007.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition using Shape Contexts. *Trans. PAMI*, 24(4):509.522, 2002.

[3] A. Johnson and M. Hebert. Object recognition by Matching Oriented Points. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA, pages 684.689, 1997. ICCV 1997.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In Proc. ECCV, pp. 404-417, 2006.

[5] P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *Trans. PAMI*, Vol. 14, No. 2, 1992.

[6] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool. Hough Transform and 3D SURF for Robust Three Dimensional Classification. In: ECCV 2010.

[7] J. Sun, M. Ovsjanikov, and L. Guibas. A Concise and Provably Informative Multi-scale Signature based on Heat Diffusion. In: SGP. 2009.

[8] M. M. Bronstein and I. Kokkinos. Scale-Invariant Heat Kernel Signatures for Non-rigid Shape Recognition. CVPR 2010.

[9] S. Salti, F. Tombari and L. Di Stefano. A Performance Evaluation of 3D Keypoint Detectors. International Conference on 3DIMPVT, 2011.

[10] A. Mian, M. Bennamoun, and R. Owens. On the Repeatability and Quality of Keypoints for Local Feature-based 3D Object Retrieval from Cluttered Scenes. IJCV 2009.

[11] A. M. Bronstein, M. M. Bronstein, B. Bustos, U. Castellani, M. Crisani, B. Falcidieno, L. J. Guibas, I. Kokkinos, V. Murino, M. Ovsjanikov, G. Patan, I. Sipiran, M. Spagnuolo, J. Sun. SHREC 2010: Robust Feature Detection and Description Benchmark. Proc. EUROGRAPHICS Workshop on 3D Object Retrieval (3DOR), 2010.

[12] E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, T. Darom, R. Horaud, I. Hotz, Y. Keller, J. Keustermans, A. Kovnatsky, R. Litman, J. Reininghaus, I. Sipiran, D. Smeets, P. Suetens, D. Vandermeulen, A. Zaharescu, and V. Zobel. SHREC 2011: Robust Feature Detection and Description Benchmark. ArXiv e-prints, February 2011.

[13] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3D Point Cloud Based Object Maps for Household environments. Robotics and Autonomous Systems Journal (Special Issue on Semantic Knowledge), 2008.

[14] K. Chatfield, J. Philbin, and A. Zisserman. Efficient Retrieval of Deformable Shape Classes using Local Self-Similarities. In NORDIA Workshop at ICCV 2009, 2009.

[15] J. Maver. Self-Similarity and Points of Interest. Trans. PAMI, Vol. 32, No. 7, pp. 1211-1226, 2010.

[16] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Numerical geometry of non-rigid shapes. Springer, 2008. ISBN: 978-0-387-73300-5.

[17] J. Huang, S. You, and J. Zhao. Multimodal Image Matching using Self-Similarity. 2011 IEEE Applied Imagery Pattern Recognition Workshop, Washington DC, 2011.

[18] S. Ruiz-Correa, L. G. Shapiro, and M. Meilă. A New Signature-based Method for Efficient 3-D Object Recognition. In Proc. CVPR, 2001.

[19] M. Körtgen, G.-J. Park, M. Novotni, and R. Klein. 3D shape matching with 3D Shape Contexts. In The 7th Central European Seminar on Computer Graphics, April 2003.

[20] J. Rugis, and R. Klette. A Scale Invariant Surface Curvature Estimator. In Proc. PSIVT, pages 138-147, 2006.