Dimensional Induced Clustering for Surface Recognition

Massimiliano B. Porcu
Dip.to Matematica e Informatica
Università di Cagliari
Via Ospedale, 72
I-09124, Cagliari, Italy
massi@dsf.unica.it

Riccardo Scateni
Dip.to Matematica e Informatica
Università di Cagliari
Via Ospedale, 72
I-09124, Cagliari, Italy
riccardo@unica.it

Abstract

Understanding when a cloud of points in three-dimensional space can be, semantically, interpreted as a surface, and then being able to describe the surface, is an interesting problem in itself and an important task to tackle in several application fields. Finding a possible solution to the problem implies to answer to many typical questions about surface acquisition and mesh reconstruction: how one can build a metric telling whether a point in space belongs to the surface? Given data from 3D scanning devices, how can we tell apart (and eventually discard) points representing noise from signal? Can the reached insight be used to align point clouds coming from different acquisitions?

Inside this framework, the present paper investigates the features of a new *dimensional clustering algorithm*. Unless standard clustering methods, the peculiarity of this algorithm is, using the *local fractal dimension*, to select subsets of lower dimensionality inside the global of dimension N.

When applied to the study of discrete surfaces embedded in three dimensional space, the algorithm results to be robust and able to discriminate the surface as a subset of fractal dimension two, differentiating it from the background, even in the presence of an intense noise. The preliminary tests we performed, on points clouds generated from known surfaces, show that the recognition error is lower than 3 percent and does not affect the visual quality of the final result.

Keywords: Surface representations; Point-based representation; Geometric algorithms.

1 INTRODUCTION

In the last few years, the diffusion of 3D data acquisition devices and scanning systems, boosted the use of huge volumes of point samples to model and represent real world objects [18]. At the same time, a big effort has been done, from the computational point of view, to improve algorithms and analytic techniques for point clouds processing, surface reconstruction, and semantic interpretation [11, 17].

In this context, the contribution of this paper regards the relations between a surface and the points cloud defining it. Particularly, we investigate on the possibility that there is some intrinsic feature of a points cloud, allowing us to discriminate a dataset defining a surface from the acquisition noise or any other uncorrelated point subset. We define *intrinsic feature* something that is independent from any particular reconstruction we can arrange, and can be evaluated directly on the points set. Some important topics are strictly related to this problem: given a points cloud defining a surface, can we figure out whether a new point belongs to the same set of the others or not (is a point on the surface or not)? Can we select, and discard, mismatched points (error outliers)? How many points can we remove from a surface, without damaging its intrinsic existence (decimation)?

As a research starting point, we can observe that the essential peculiarity of a surface is to be a subset of intrinsic dimension two. Several clustering methodologies has been proposed lately for the detection, inside a higher-dimensional dataset, of subsets of points whit lower intrinsic dimensionality [10, 4, 16]. These techniques are then capable to discover, into the global set, points arranged along lines (1D subsets), over surfaces (2D subsets) or hypersurfaces (nD subsets).

A recently proposed algorithm, the *Dimensional Induced Clustering* algorithm (DIC), appears to be very effective at doing this [10]. Based upon the evaluation of the intrinsic fractal dimension and the local point density, it is able to discriminate between subsets of different dimensions, and also of the same dimension but different density.

 ISBN 978-80-86943-98-5

In this paper, we investigate on the use of DIC for the discrimination of points clouds in three dimensional space, and on its consequences. The rest of this work is organized as follows: in Section 2 we briefly go over the previous work done in points clouds processing and clustering techniques; in Section 3 we illustrate the Discrete Induced Clustering Method; then, we describe the use of DIC for the study of surface points clouds, introducing the general investigation framework and going into details in Section 4; in Section 5 we show the results obtained using the algorithm on several meshes widely used in literature; finally in Section 6 we draw our conclusions and describe the future evolutions of this work.

2 PREVIOUS WORK

Several techniques have been proposed for the reconstruction of surfaces from oriented point samples.

The problem of reconstructing a surface obtained from 3D scans is mainly reduced to the registration of the set of regular maps (the points are placed on the vertices of a regular grid) since the relative positions of the single maps is known by the acquisition planning [9]; under normal conditions there is low digital noise and it is interesting to understand where the map superpositions are.

The most diffused approaches when dealing with a set of *really*[1] unorganized points are based on topology. They use alpha shapes [12] or three-dimensional Voronoi diagrams of the point set, and then use the facets of the tessellation for the reconstruction of the surface [2]. The latter technique was recently reformulated to be also able to filter out small perturbations (noise) near to the target surface [7].

Density-based clustering methods uses local density information in order to partition a dataset (it could be a point set). In this context, Hierarchical Single Linkage is a widely diffused technique, implemented in common packages as DBSCAN [8], OPTICS [3] and CLIQUE [1].

The intrinsic fractal dimension has been successfully used in numerous database selection problems, as the nearest-neighbor queries [19] and spatial query selective estimation [5]. Recently, also some partitioning and clustering methods based upon this technique have been proposed [4].

3 THE DIC ALGORITHM

Clustering methods search for patterns and regularities in data, to perform partitions in homogeneous sets. The Dimension Induced Clustering algorithm (DIC) [10] carries this out computing the intrinsic dimension and the point density. It is therefore designed to detect, inside a higher-dimensional dataset, subsets of points whit lower intrinsic dimensionality, such as lines (1D subsets) or surfaces (2D subsets). Moreover, the use of local point density information makes the method being able to discriminate between subsets of the same intrinsic dimension.

3.1 Fractal Dimension

Behind the word *dimension* of a set there are actually two different meanings that may not be coincident. The first one is the number of independent coordinates used to locate, in unambiguous way, a point of the set. The second one is the relative growing rate of the point set, that is to say, the *speed* the point set fill the space with. This is in fact the *fractal dimension*. This concept can be explained using the drawing of Figure 1. Consider the seed A and count the number of points of the set inside the ball of radius R: doubling the ball radius, we expect to find twice the number of points, so the fill rate grows linearly with the radius and thus the fractal dimension of the point set around A is one. If we consider, instead, the seed B, we expect to have a quadratic fill rate and thus the fractal dimension of the point set around B is two. We can also have a non-integer fractal dimension: using, for instance, as seed the point C we obtain a fractal dimension of 1.5.

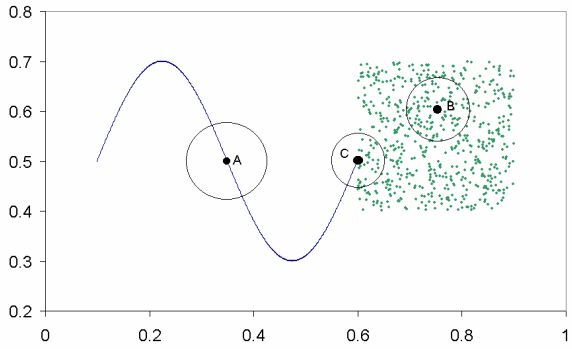

Figure 1: Fractal dimension interpreted as the relative growing rate of a point set.

Consider now the dataset $X \subset \mathbf{R}^m$ and assume that the number of points n of X tends to infinity. Let choose a distance function on X and let $B(x,r)$ be the ball of radius r centered at x; let $|B(x,r)|$ be the number of points inside the ball. We define the *local growth function* for the point x as:

$$G_x(r) = \lim_{n\to\infty} \frac{1}{n}|B(x,r)|$$

The local growth function represents the density of the subset $B \subset X$ for a fixed r; it contains in-

[1] We mean that there is no implicit information on the adjacencies as in 3D scanning.

 ISBN 978-80-86943-98-5

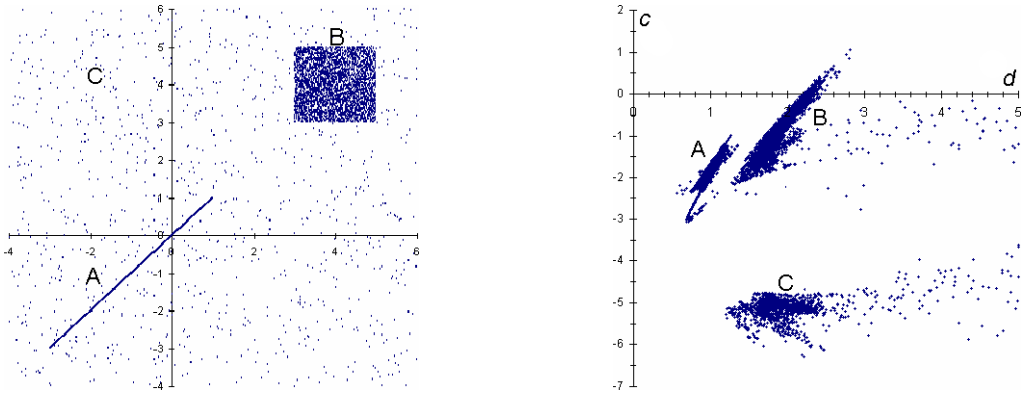

Figure 2: The DIC algorithm is able to discriminate between: (i) datasets of different dimensions; (ii) datasets of the same dimension but different densities.

formation about the growth rate of the number of neighbors of x. The local fractal dimension d_x of a point x is defined as:

$$d_x = \lim_{r,r' \to 0} \frac{\log \frac{G_x(r)}{G_x(r')}}{\log \frac{r}{r'}}$$

Since we deal with finite sets, to evaluate the fractal dimension d_x we define the local growth curve as the fraction of points inside the ball for a chosen radius, $G_x(r) = \frac{|B(x,r)|}{n}$, and compute d_x of a point x as the slope of the curve $G_x(r)$ in log-log scale. Practically we can do that assuming a linear model for the curve, and finding the line that fits it best, using a least squares method. To obtain a correct local fit we need to consider the function $G_x(r)$ in a region not too much distant from the center x (for details see [10]). Using the symbol L_x to denote the curve, our linear model produces a fitting curve of the form:

$$L_x(\log r) = d_x \log r + b_x$$

3.2 Local Density

For any value of x there is a particular value r^* for the radius leading to the definition of the coefficient c_x:

$$c_x = L_x(\log r_x^*) = d_x \log r_x^* + b_x$$

We can take c_x as a measure of the density of the dataset in a ball of radius r^* centered on x and thus the radius r^*, varying from point to point, is the unique value minimizing the correlation between d_x and c_x and therefore maximizing the information contained in the couple of coefficients. It is possible to prove that this condition is obtained choosing:

$$\log r_x^* = \frac{\sum (d_x - \overline{d})(b_x - \overline{b})}{\sum (d_x - \overline{d})}$$

where $\overline{d}$ and $\overline{b}$ are the arithmetic means of the coefficients d_x and b_x computed on the whole dataset.

3.3 Local Representation

Using this procedure, we can thus define a mapping $f : X \to \mathbf{R}^2$ from the m-dimensional dataset X to the plane: we call it the *Local Representation* ($\mathcal{LR}$) of X. This mapping projects each element of the set into the 2D plane using the couple of coefficients (d, c), that are, respectively, the fractal dimension and the local points' density. Once projected the starting dataset X to its $\mathcal{LR}$, the partitioning task becomes definitely easier: we can, in fact, analyze and partition the $\mathcal{LR}$ and then project back the obtained results to the original dataset. The partition on the $\mathcal{LR}$ can be automatic, applying some standard clustering algorithm like Expectation Maximization (EM) [13], or interactive by visual inspection.

3.4 DIC Trials

Using the co-operating descriptors (d, c), DIC appears to be very effective since it is able not only to discriminate between subsets of different dimension, but also to distinguish subset of the same dimension but different densities. An example is given in Figure 2. The starting 2D dataset (left) is a mixture of three different types of regions: a 1D line (region A) a 2D dense cloud (B) and diffuse ground noise (C). B and C have the same dimension while A and B have the same density.

As a result of the application of the DIC algorithm, the set is projected to its $\mathcal{LR}$ (right). Notice the presence of 3 well separated clouds of points: two of them show a fractal dimension $d \approx 2$ while the third shows $d \approx 1$. The back projection shows that the clouds correspond, respectively, to the subsets A, B and C in the original point set. Using only the fractal dimension information, it is not possible to tell B and C apart; instead, using only a measure for the density, subsets A and B are not separable.

 ISBN 978-80-86943-98-5

4 SURFACE RECOGNITION

Now we want to focus our attention on the main contribution of this paper: how to use the DIC algorithm for surface recognition and decimation.

One possible scenario is as follows: consider to have to deal with a 3D points dataset of possible unknown origin sampling a surface in space, that can also be affected by a high experimental error or be completely intermixed with ground noise points. You want to be able, first, to reconstruct the surface, then to capture its saliencies using possibly unsupervised methods. Another alternative scenario can instead be the alignment of several range maps coming from 3D scans. We will see how we can use DIC for both these purposes.

To experiment on the use of the DIC algorithm we first modified well known benchmark datasets adding artificial noise to them. We embedded the datasets in a space region in which is present white noise constituted by a set of points, without any special regularity, randomly distributed with homogeneous density. The noise was produced using a standard random number generator.

To represent a correlate set of points, as for example a surface, a subset of the dataset should have some intrinsic features allowing us to distinguish it from the ground noise. Particularly, using the DIC approach, we want:

- DIC to be able to separate the data set from the ground noise, using dimension and/or density;
- The local fractal dimension of the data set points be correct (for a surface, $d \approx 2$).

This point of view is analogous to a signal-noise ratio evaluation. The starting data set being the signal, that we suppose to be two-dimensional, and the added points being the background noise, three-dimensional by construction.

Let now step over the different tasks performed on the dataset.

4.1 Surface Finding

In Figure 3 we represent a point set embedded in a noisy 3D space region. We constructed the point set using the vertices of a triangle mesh commonly used as a benchmark mesh (Oilpump, on the left). Starting from a surface, we know for sure that the points cloud does define a surface. We then introduced random noise, adding a number of points equal to half the points on the surface, chosen at random inside a box twice the volume of the surface's bounding box (on the right).

The global set of points, (surface + noise), has been processed with DIC, producing its Local Representation $\mathcal{LR}$ (in Figure 4). The possibility to separate the signal from the noise, and the computational techniques used to do it, depends on the features of $\mathcal{LR}$. In the diagram we can notice the presence of two well separated clusters, and of some outliers. The cluster named N, with a lower value of c, is centered around $d \approx 3$; the cluster S, with greater c, is centered around $d \approx 2$. S and N have spheroidal shape, are well separated from each other and overlap just in the outlier region. Under these conditions, it is possible to subdivide the $\mathcal{LR}$ plot in clusters, project them back on the 3D points generating them, and then use the clusters to partition the original 3D dataset. This procedure can be completely unsupervised using a standard partition method. We adopted, in this case, the K-Medoids algorithm [14].

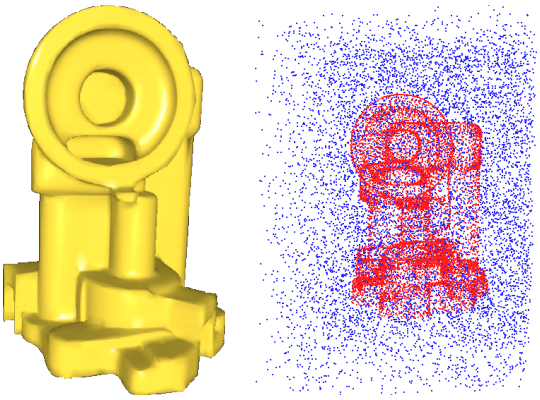

Figure 3: The Oilpump data set. The original surface on the left, the point set with white noise added on the right.

A backtrace check for clusterization procedure is possible, because the membership for the points, surface or noise, is known by construction. It shows that DIC discriminates correctly the noise cloud (N) from the surface point set (S). In fact, N has a mean fractal dimension $d \approx 3$ and lower density c and it is contains almost only noise points. S has dimension $d \approx 2$ and higher density and it is mainly constituted of surface points. In our experiments, as it will be explained in further detail in Section 5, uncorrect assignments are bounded by less than 3 percent for false negatives and by 6 percent for false positives.

4.2 Noise Raising

The surface finding procedure has been applied on datasets with different noise levels, changing the noise density. The back projections show a remarkable capability of DIC to discriminate the surface point set, also in challenging environmental conditions.

It's worth to remember that the signal stability, when the level of background noise raises, is

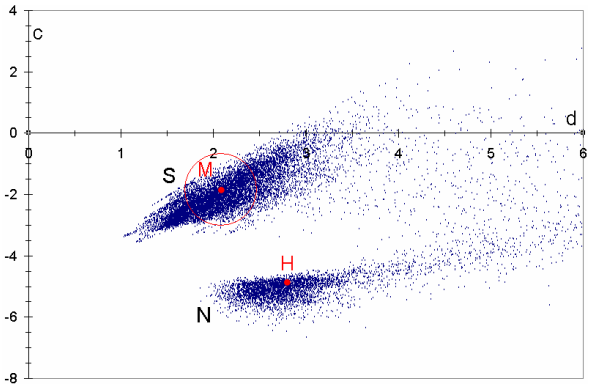

Figure 4: The Local Representation for the Oilpump dataset: in red the medoids M ($d = 2.156, c = -2.038$, signal) and H ($d = 2.764, c = -5.025$, noise).

a measure of the consistency and the reliability of the signal itself. Applied to our problem, it means that the likelihood that the point set defines a surface is directly correlated to the background noise level. We can evaluate this property for every single point of the data set. Embedding the data set in 3D white noise, some points will be mingled with the ground jelly, these points will be easily not classified as belonging to the surface by a reconstruction algorithm. On the contrary, the points that are classified as surface points even for high level of ground noise are the subset of strongly attached points, that is the intrinsic skeleton of the surface.

4.3 Points Decimation

In the last years, modelling and visualization techniques using points as primitives, emerged as interesting alternatives to traditional triangle mesh-based processing [22, 23, 15, 21]. In this frame it becomes important to set up algorithms capable to conveniently reduce the number of point samples while keeping the underlying shape features unchanged. Recently some methods has been proposed, based upon the intrinsic feature of the points cloud, mainly related to sample distribution and density [2, 6, 20].

We investigated the use of the DIC algorithm as a tool to perform a point selection based on intrinsic features like the fractal dimension d and the local density c.

To do this we adopted the following scheme: the clusterization method K-Medoids, used to separate the clouds on $\mathcal{LR}$ plot, works selecting a seed for each cluster, named *medoid*, that is representative of the cluster itself. The method assigns each element to the right medoid, minimizing the expectation value of the distance, with respect to some metric functions [13]. We can then select, for a cluster in the $\mathcal{LR}$ plot, a set of points closer to the relative medoid; for instance, inside a ball of some radius r.

Performing this procedure for the cluster representing the signal, the points we select by construction are:

- Representative of the surface;
- Homogeneous in density;
- Homogeneous in intrinsic dimension;
- Well separated from the noise.

We show an example for the Oilpump dataset in Figures 4 and 5. Using the information from the $\mathcal{LR}$ plot, we select a set of points of the signal by fitting a ball around the medoid M. The point density, as function of the distance from M (in Figure 5), shows that the most part of the surface, over 99%, is contained inside a ball of radius $r = 1.5$.

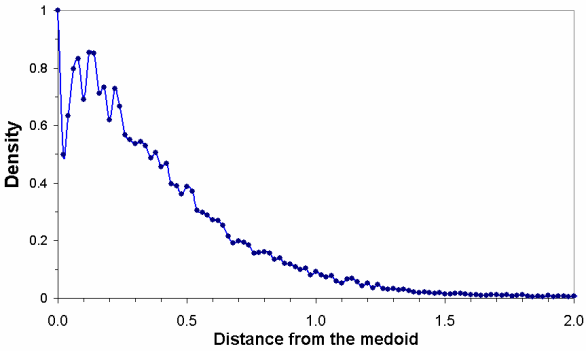

Figure 5: The normalized relative density of the points plot in Figure 4, as function of the distance from M.

In Figure 6 there are four examples of point subsets obtained using balls of different radius to select points from the dataset.

4.4 Superposition and Alignment

While reconstructing surfaces from 3D scans, different views need to be correctly aligned and overlapped in the common borders regions [9]. This task is complicated by the acquisition errors that are a further unknown of the problem.

Even if the type of error introduced by the acquisition process is not the random white noise that our method is best suited for, the DIC algorithm can still provide a support to existing techniques. Consider the really trivial example shown in Figure 7. Two planar surfaces orthogonal to z, with the same point density, are embedded in a 3D noisy region. Initially the surfaces are partially overlapped on xy, but distant in z (top left). Modifying the position in z of one plane, we generate a clouds overlap (top right). The $\mathcal{LR}$ plots relative to the two different cases are shown in the same

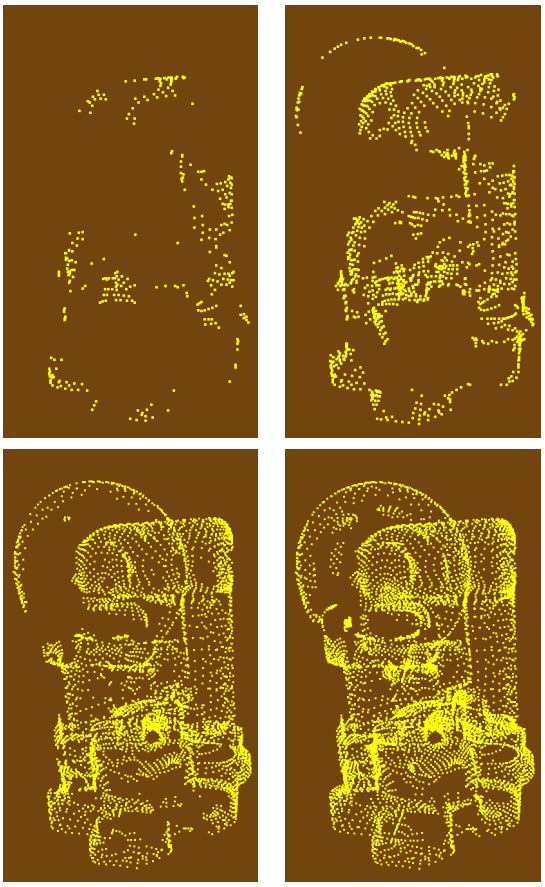

Figure 6: Oilpump points decimation, obtained using the medoid distance technique. From left to right and top to bottom, the subsets are generated using, respectively, $r = 0.3$, $r = 0.5$, $r = 0.8$, and $r = 1.1$.

picture (down left), where red means the overlap case and blue the non-overlap one. When there is no overlap the $\mathcal{LR}$ shows just two clusters since the planes have the same intrinsic dimension and the same density. In this case, DIC can discriminate them from the background noise but cannot create any link from one to another.

In the second case, instead, the points density in the overlapping region is twice the density of each plane. In this situation, DIC is capable to figure out if there is a superposition and also to figure out where the overlapping region is, as one could expect when they represent the same swatch of surface scanned twice. As one can see in the two diagrams, the $\mathcal{LR}$ plot for overlapping surfaces (red points) shows a third cluster, with the same fractal dimension of the blue signal cloud but higher density. As the analysis points out, this cluster is made of points in the superposition region. Being able to separate the signal from the noise, this mechanism correctly characterize the overlapping region taking into account just the points on the surfaces, and discarding the values affected by errors.

In this case the cluster separation is not simple using the K-Medoids method and user intervention is needed. It is, anyway, the only case we found in which the method needs to be supervised.

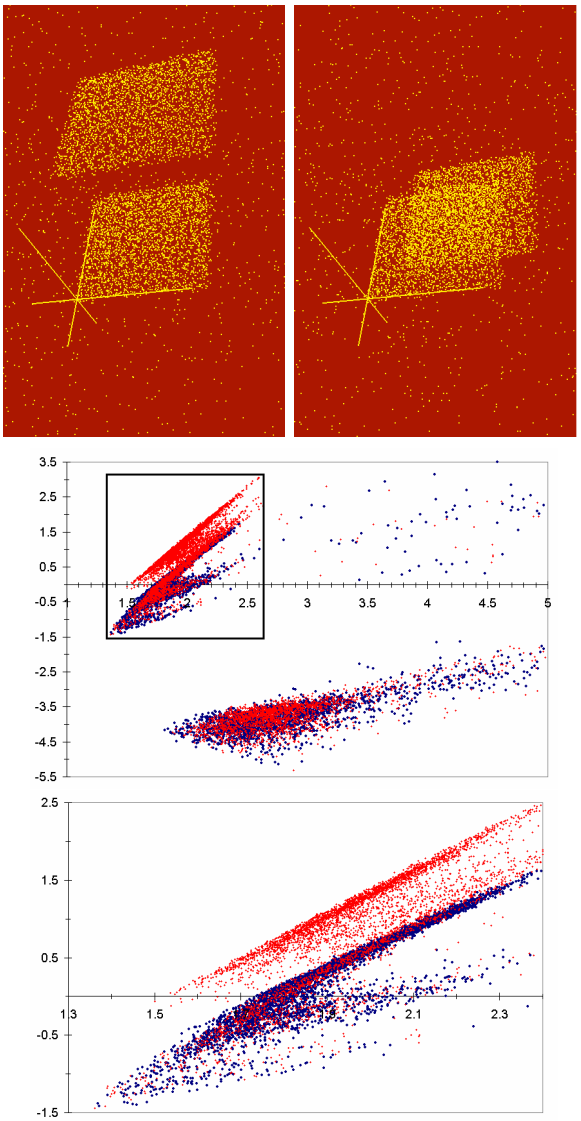

Figure 7: Two planes embedded in background noise. Not overlapped (top left), overlapped (top right). The related $\mathcal{LR}$ (in the middle, red for overlapped and blue for non overlapped) and a detail of the boxed region (bottom).

5 RESULTS AND DISCUSSION

In Table 1 we summarize the results of the discrimination between surface and noise, performed for different meshes. Noise was randomly generated adding points in an amount of 50% the number of points of the original mesh, inside a box eight times the volume of the mesh bounding box. We list, in the order, the name of the mesh, the number of its points and the number of added points. With

Table 1: Noise/Surface discrimination for eight different datasets: in column V there is the number of points from the surface and in column N the number of added points.

Dataset	V	N	α	$\alpha_{\%}$	β	$\beta_{\%}$
Screw	904	452	14	1.5	55	6.0
Bunny	1015	507	2	0.2	155	1.5
Face	2278	1139	0		118	5.1
Ahead	3123	1561	89	2.8	124	3.9
Teapot	4255	2127	56	1.3	163	3.8
Cat	4539	2269	0		198	4.3
Fandisk	6475	3237	0		293	4.5
Oilpump	10274	5137	97	0.9	276	2.6

α we indicate the false negatives (surface points assigned to the noise set) and with β the false positives (noise points assigned to the surface). In columns $\alpha_{\%}$ and $\beta_{\%}$ we list the error percentages.

We can notice that the number of uncorrect assignments is small for both error types. It is very important that the number of false negatives is bounded by 3% and is, on average, much less. This implies that almost all the points to be used for the surface recognition are selected. The false positives are much more, because while dropping points at random in the bounding box the probability to choose spatial locations very close to the surface is not negligible.

In Table 2 we listed $\alpha_{\%}$ and $\beta_{\%}$ as function of the noise points density, for the Oilpump and Teapot datasets. The noise (first column) is expressed in a relative way: we start putting in a box, linearly twice the mesh bounding box, an amount of 50% the points of the mesh; then, we shrink the noise box to increase its density. Assuming 1 as a reference starting density value, the other values represent the relative growth. We can notice that $\alpha_{\%}$ is not related to the noise density; this means that the efficiency of the method is high also in challenging conditions. On the other hand, $\beta_{\%}$ grows more than linearly with the density. This is due to the fact that, reducing the noise box, the probability to generate random points located on, or very close to, the surface increases. In some sense then, a part of $\beta_{\%}$ is not really an error!

6 CONCLUSIONS AND FUTURE WORK

We presented in this work the application of a clusterization method to the recognition of surfaces from point clouds. The algorithm is based on the computation of the fractal dimension and local density for each point in the cloud (its $\mathcal{LR}$ map).

The main advantages of the method are: it can be used in an unsupervised mode using a standard clustering algorithm; it can be used interactively allowing for visual inspection of the characteristics of the dataset; based on the $\mathcal{LR}$ map a point decimation is direct and simple; a straightforward extension of the technique can be used for computing the superposition of different range maps coming from a 3D object scan.

Table 2: Variations of $\alpha_{\%}$ and $\beta_{\%}$ when changing the noise density.

Noise density	Oilpump		Teapot	
	$\alpha_{\%}$	$\beta_{\%}$	$\alpha_{\%}$	$\beta_{\%}$
1.00	0.96	1.22	1.39	1.57
1.18	0.98	1.47	1.27	2.11
1.42	0.98	1.94	1.29	2.84
1.72	0.94	2.69	1.32	3.83
2.12	0.94	3.51	1.36	4.61
2.65	0.91	5.53	1.18	6.56

The results, even if preliminary, are very promising, especially those regarding the point decimation aspect. We still need to work on several aspects of the method, the most important being: rework the implementation to be able to deal with larger datasets; make experiments on real sets of unorganized points instead of benchmarking on points clouds derived from meshes.

Nevertheless we are confident that it can already be useful for the purposes exposed.

REFERENCES

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD Conference*, pages 94–105, 1998.

[2] Nina Amenta and Marshall W. Bern. Surface reconstruction by voronoi filtering. *Discrete & Computational Geometry*, 22(4):481–504, 1999.

[3] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. In *SIGMOD Conference*, pages 49–60, 1999.

[4] Daniel Barbarà and Ping Chen. Using the fractal dimension to cluster datasets. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 260–264, New York, NY, USA, 2000. ACM Press.

[5] Alberto Belussi and Christos Faloutsos. Self-spacial join selectivity estimation using fractal concepts. *ACM Trans. Inf. Syst.*, 16(2):161–201, 1998.

[6] Tamal K. Dey, Joachim Giesen, and James Hudson. Decimating samples for mesh simplification. In *CCCG*, pages 85–88, 2001.

[7] Tamal K. Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. *Comput. Geom.*, 35(1-2):124–141, 2006.

[8] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[9] Thomas Franken, Matteo Dellepiane, Fabio Ganovelli, Paolo Cignoni, Claudio Montani, and Roberto Scopigno. Minimizing user intervention in registering 2D images to 3D models. *The Visual Computer*, 21(8-10):619–628, 2005.

[10] Aristides Gionis, Alexander Hinneburg, Spiros Papadimitriou, and Panayiotis Tsaparas. Dimension induced clustering. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 51–60, New York, NY, USA, 2005. ACM Press.

[11] Enrico Gobbetti and Fabio Marton. Layered point clouds: a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models. *Computers & Graphics*, 28(6):815–826, December 2004.

[12] Baining Guo, Jai Menon, and B. Willette. Surface reconstruction using alpha shapes. *Comput. Graph. Forum*, 16(4):177–190, 1997.

[13] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer-Verlag, New York, 2001.

[14] Leonard Kaufman and Peter J. Rousseeuw. *Finding groups in data. an introduction to cluster analysis.* Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990, 1990.

[15] Leif P. Kobbelt and Mario Botsch. A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(6):801–814, December 2004.

[16] Robert Krauthgamer and James R. Lee. The black-box complexity of nearest neighbor search. In *ICALP*, pages 858–869, 2004.

[17] Jens Krüger, Jens Schneider, and Rüdiger Westermann. Compression and rendering of iso-surfaces and point sampled geometry. *The Visual Computer*, 22(8):517–530, 2006.

[18] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 131–144, July 2000.

[19] Bernd-Uwe Pagel, Flip Korn, and Christos Faloutsos. Deflating the dimensionality curse using multiple fractal dimensions. In *ICDE*, pages 589–598, 2000.

[20] Mark Pauly, Markus Gross, and Leif P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Proceedings of IEEE Visualization 2002*, pages 163–170. IEEE Press, 2002.

[21] Mark Pauly, Leif P. Kobbelt, and Markus Gross. Point-based multiscale surface representation. *ACM Transactions on Graphics*, 25(2):177–193, April 2006.

[22] Miguel Sainz and Renato Pajarola. Point-based rendering techniques. *Computers & Graphics*, 28(6):869–879, December 2004.

[23] Ingo Wald and Hans-Peter Seidel. Interactive ray tracing of point-based models. In *Symposium on Point-Based Graphics 2005*, pages 9–16, June 2005.