



# Adaptive multi-scale analysis for point-based surface editing

Georges Nader, Gael Guennebaud, Nicolas Mellado

## ► To cite this version:

Georges Nader, Gael Guennebaud, Nicolas Mellado. Adaptive multi-scale analysis for point-based surface editing. Computer Graphics Forum, 2014, 33, pp.9. 10.1111/cgf.12485 . hal-01059392

**HAL Id: hal-01059392**

**<https://inria.hal.science/hal-01059392>**

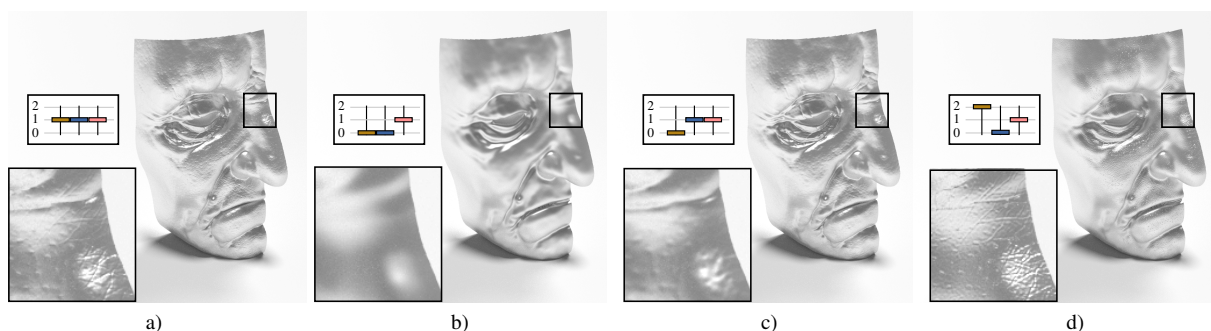
Submitted on 29 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Adaptive multi-scale analysis for point-based surface editing

G. Nader<sup>1</sup> and G. Guennebaud<sup>2</sup> and N. Mellado<sup>3</sup>  
<sup>1</sup>Université de Lyon, LIRIS UMR 5205 CNRS, France  
<sup>2</sup>Inria - Univ. Bordeaux - IOGS - CNRS, France  
<sup>3</sup>University College London, United Kingdom



**Figure 1:** Feature-based editing of a detailed point cloud (1.5 millions points). After a prior analysis of the input model (a) to detect, count and extract pertinent scales, the user can edit the geometry in real-time using a graphic equalizer to, for instance, remove the two first level of details (b), remove only the scratches and skin pores (c), or boost them and remove the wrinkles defined at an intermediate scale (d).

## Abstract

*This paper presents a tool that enables the direct editing of surface features in large point-clouds or meshes. This is made possible by a novel multi-scale analysis of unstructured point-clouds that automatically extracts the number of relevant features together with their respective scale all over the surface. Then, combining this ingredient with an adequate multi-scale decomposition allows us to directly enhance or reduce each feature in an independent manner. Our feature extraction is based on the analysis of the scale-variations of locally fitted surface primitives combined with unsupervised learning techniques. Our tool may be applied either globally or locally, and millions of points are handled in real-time. The resulting system enables users to accurately edit complex geometries with minimal interaction.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

## 1. Introduction

During the last decade, numerous deformation tools have been proposed to ease the manipulation of global shapes (e.g., the pose of a character) of increasingly complex and detailed geometries [BS08, LLCO08, JBPS11, PLG13]. In contrast, the manipulation of medium to fine scale features (e.g., carved symbols) has received very little attention. Sculpting tools, as found in many production software,

are very well suited to the creation of details and they can even be applied to point-clouds [ZPKG02, PKKG03]. This is desirable to avoid issues with changes in topology and to enable the direct editing of data coming from 3D acquisition systems. On the other hand, sculpting tools are time consuming when it comes to exaggerating or smoothing existing geometric features: the user has to manually adjust the size of the brush's footprint, and advanced drawing skills are required to properly follow the features. Moreover, this ap-

proach lacks accuracy as it is often impossible to precisely edit a feature at a given scale without affecting the other ones. Our main objective is to make this possible.

This last observation motivates the needs for a clear separation of the features at different scales. A well established practice for the multi-scale analysis of shapes is to consider the eigenfunction decomposition of the heat kernel or Laplace-Beltrami operator [Lev06, SOG09, ZVKD10]. Such spectral approaches have been recently extended to point-clouds [LPG12, PS13]. The shape can then be manipulated at different frequencies through a re-composition of the eigenfunctions [VL08]. However, this does not permit to directly edit a precise feature at a given scale as such a task requires to determine the respective set of eigenfunctions. Despite a very recent effort in that direction [NVT\*14], this still constitute a challenging problem.

Another approach consists in computing a sequence of surfaces at multiple scales, for instance, through a smoothing operator [KCVS98, GSS99]. Then, to ensure that the changes performed at one level are consistently propagated to higher ones, the levels are stored as the differences between two successive scales in a coarse to fine manner. Then, one can smooth a specific band frequency by weighting the coefficients of the respective levels. As proposed by Pauly et al. [PKG06], this approach can be efficiently applied to point-clouds using Moving-Least-Squares (MLS) [AA03] to define the surface levels, and sub-sampling to speed-up the computation of the largest scales. One of the major limitation is that the levels of the discrete multi-scale representation do not coincide with the scales of details and features which are present in the surface. As a result, manipulating a specific feature is still a tedious task which requires the manual adjustment of several levels in a non uniform manner.

To address this limitation, we propose a novel multi-scale decomposition which automatically and dynamically groups the different frequencies such that every level precisely matches a relevant feature present in a region of interest selected by the user. Each feature may thus be enhanced or smoothed independently by controlling a single weight per feature, for instance via an equalizer interface (Figure 1). This is made possible by a multi-scale analysis technique that automatically determines the ranges of scales corresponding to the most relevant features. From a technical point of view, we introduce a novel method to estimate the pertinence of a scale which extends the notion of *homological persistence* [ELZ00] using a meshless *geometric variation* measure [MGB\*12]. We also show how the extraction problem can be robustly addressed by a clustering algorithm. Consistency between our multi-scale analysis and decomposition is ensured by a joint use of the Algebraic Point Set Surfaces definition [GG07], which robustly defines a smooth implicit surface from a set of points at arbitrary scales through the fitting of algebraic spheres. Finally, after

a short preprocessing time, our approach handles millions of points in real-time.

## 2. Overview

Our editing tool starts with an unstructured set of points  $P = \{\mathbf{p}_i\}$  describing a smooth surface. Using the Algebraic Point Set Surfaces (APSS) method [GG07], we conceptually define a continuous set of scale-varying surfaces  $S$ . At a pre-processing stage this formalism is used to decompose our original point-cloud  $P$  into a discrete multi-scale representation (Figure 3), where each level is incrementally stored in a coarse-to-fine manner, by storing the differences between two successive levels as detailed in section 3.

Given a region of interest (ROI) selected by the user (Figure 2a), our main goal is to automatically generate an adaptive geometric equalizer where each slider controls an interval of scales defining a pertinent feature. We start by computing for each selected point  $\mathbf{p}_i$  its multi-scale geometric variation  $v_i$  (Figure 2b) using the above scale-varying surfaces  $S$  in conjunction with the Growing Least Squares (GLS) technique [MGB\*12]. For each scale the function  $v_i$  measures how much the neighborhood of the point  $\mathbf{p}_i$  changes for a small scale variation. The intuition of our detection method is that pertinent features occur at scales exhibiting small geometric variation, while high geometric variation indicates transition between two features at different scales. This insight is used to determine weighted pertinent scales (Figure 2c) which are accumulated over the ROI.

Assuming the ROI is homogeneous in terms of features and scales, the number and respective scales of these features are extracted using an unsupervised learning procedure (Figure 2d). This extraction process is described in section 4.

Once this local analysis has been carried out, each slider is mapped to a band of the multi-scale representation. The feature corresponding to each band is then either enhanced or reduced independently (Figure 2e), and the surface is reconstructed in real-time as detailed in section 5.

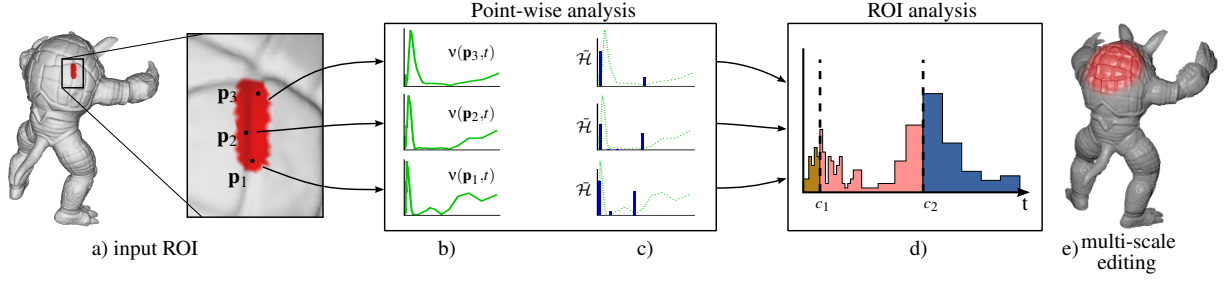
## 3. Multi-scale representation

Our multi-scale representation follows the top-down construction principle of Pauly et al. [PKG06]. Instead of using planar local approximations to define the surface levels, we employ the APSS approach [GG07] which avoids collapsing at large scales, and is more robust to sparse sampling.

Given a set of points  $P = \{\mathbf{p}_i\}$  equipped with normals  $\mathbf{n}_i$ , the surface  $S(t)$  at scale  $t$  is defined as the 0-isosurface of the scale-varying scalar field  $f(\mathbf{x}, t)$ :

$$f(\mathbf{x}, t) = [1 \ \mathbf{x}^T \ \mathbf{x}^T \mathbf{x}] \cdot \mathbf{u}(\mathbf{x}, t), \quad (1)$$

where  $\mathbf{u}(\mathbf{x}, t) = [u_c \ \mathbf{u}_l \ u_q]^T$  is the 5D vector of coefficients defining the algebraic sphere which best approximates the neighborhood of  $\mathbf{x}$ . In practice,  $\mathbf{u}$  is obtained through a fast

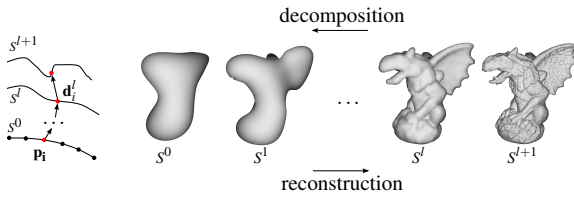


**Figure 2:** Feature aware editing pipeline. Starting with a given Region Of Interest (ROI) of the input point-cloud (a), pertinent structures are learned from the ROI by computing a point-wise multi-scale geometric variation profiles (b) which are analyzed to determine weighted pertinent scales (c). Those weighted scales are combined to extract the pertinent scales of the ROI (d) enabling the manipulation of each pertinent structures independently (e).

weighted least-square minimization [GGG08]. The influence of the neighbors of  $\mathbf{p}_i$  is controlled by the following compactly supported and scale dependent weight function:

$$w_i(\mathbf{x}, t) = \phi\left(\frac{\|\mathbf{x} - \mathbf{p}_i\|}{t}\right), \text{ with } \phi(x) = \left(\min(1, x^2) - 1\right)^2. \quad (2)$$

In this approach, the coarser levels are thus obtained by increasing the support size of the weight function used in the least square minimization. This results in a family of surfaces  $S(t)$  which is continuous both in space and scale, and which is amenable to multi-scale analysis as shown in the Growing Least Square (GLS) framework [MGB\*12].



**Figure 3:** Illustration of the multi-scale decomposition.

As illustrated in Figure 3, our multi-scale representation is thus obtained by a discretization of  $S(t)$  both in space and scale as a set of point clouds  $\mathbf{P} = P^0, \dots, P^L$ . Here  $P^L$  corresponds to the original point cloud  $P$ , while the other levels  $P^l$ ,  $l \in [0, L[$  correspond to a sampling of the surface  $S^l = S(t_l)$ . The scales  $t_l$  are increasing for smaller levels  $l$ , thus  $P^0$  corresponds to the smoothest approximation. A sample  $\mathbf{p}_i^l \in P^l$  is obtained by the projection of  $\mathbf{p}_i^{l+1}$  onto the surface  $S^l$ . Even though an orthogonal projection would be possible [AA04], we found it sufficient to follow a classical approach which consists in iteratively projecting  $\mathbf{p}_i^{l+1}$  onto the best locally fitted algebraic sphere until convergence. Finally, the representation stores the coarsest level  $P^0$  and for each level  $l \in [0, L[$ , the set of displacement vectors  $D^l = \{\mathbf{d}_i^l\}$ , with  $\mathbf{d}_i^l = \mathbf{p}_i^{l+1} - \mathbf{p}_i^l$ .

From this decomposition, one can reconstruct a family of point clouds by accumulating the displacement vectors:

$$P(\mathbf{a}) = P^0 + \sum_{l=0}^{L-1} a_l D^l, \quad (3)$$

where  $\mathbf{a} = \{a_l\}$  is the vector of coefficients controlling the influence of each level. For instance, using  $a_l = 1$  for all levels yields the original point cloud  $P$ .

#### 4. Detection of pertinent feature scales

This section describes our analysis and learning pipeline for the automatic detection of pertinent scales in a given region of interest (ROI). Its general principle is depicted in Figure 2. As sketched in the overview, our analysis is based on the GLS measure of the geometric variation which, given a point  $\mathbf{p} \in P$  and scale  $t$  estimates how much the surface varies for increasing scales around  $\mathbf{p}$  (Figure 2b). This measure is recalled and discussed in section 4.1. Then, the number and respective scales of these features are extracted using a two steps procedure:

1. For each point of the ROI, we analyze its geometric variation profile to extract a super-set of the possible pertinent scales which are weighted according to some heuristics (Figure 2c). This step is detailed in section 4.2.
2. The results of the previous point-wise analysis are gathered over the ROI to get a measure of the likelihood of each scale to be a pertinent one. Pertinent scales are finally extracted using an unsupervised clustering algorithm as depicted in Figure 2d and detailed in section 4.3.

##### 4.1. Geometric variation

In order to detect pertinent features with their respective scales, we first need a measure of the variation of the shape along the scale axis. A first option would be to directly consider the magnitude of the displacement vectors  $\mathbf{d}^l$  coming from the multi-scale decomposition. This measure is ex-

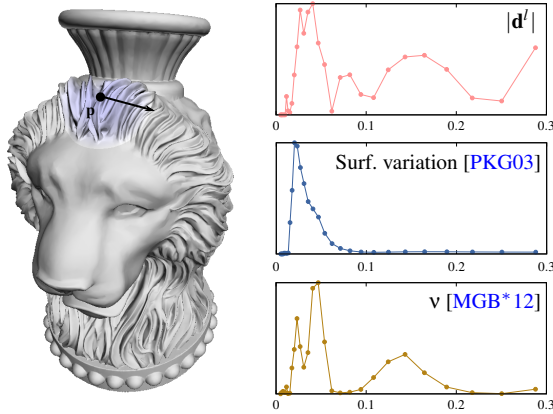


tremely local and significantly favors inflection points which will always exhibit small displacements. Indeed, intuitively other important aspects are the changes in the tangent frame orientation and changes of curvature. These three quantities are well captured by the *geometric variation*  $v$  as defined by the GLS framework [MGB\*12]:

$$v(\mathbf{x}, t) = \left( \frac{\partial \tau}{\partial t} \right)^2 + \left( t \frac{\partial \eta}{\partial t} \right)^2 + \left( t^2 \frac{\partial \kappa}{\partial t} \right)^2. \quad (4)$$

Here,  $\tau$ ,  $\eta$  and  $\kappa$  are directly obtained from the parameter  $\mathbf{u}(\mathbf{x}, t)$  of the best fitted algebraic sphere such that they approximate the distance from  $\mathbf{x}$  to the surface  $S(t)$  as well as its normal and mean curvature respectively. The role of the scale factors is to guarantee a scale invariant measure. We refer to [MGB\*12] for the details on computing  $v$ . By defining the discrete geometric variation of the point  $\mathbf{p}_i^l \in P^l$  as  $v_i^l = v(\mathbf{p}_i, t_l)$ , the scale variation of the offset  $\tau$  is directly correlated to the magnitude of the displacement vector  $\mathbf{d}_i^l$ . An important difference though, is that thanks to the continuous definition of these parameters in scale, their partial derivatives can be computed analytically and thus they do not depend at all on the arbitrary choice of a discretization in scale.

The behavior of these two measures are depicted in Figure 4 which also includes the more standard *surface variation* [PKG03]. This later is defined as the ratio between the smallest and the sum of the eigenvalues of the local covariance matrix. In practice, this ratio only measures the deviation of the neighborhood to a plane and often fails to capture many of the variations.



**Figure 4:** Profiles of the GLS geometric variation  $v$ , norm of the displacement vectors and Pauly et al.'s surface variation for the point  $\mathbf{p}$ . The black arrow represents the second detected scale by the geometric variation ( $\sim 0.08$ ).

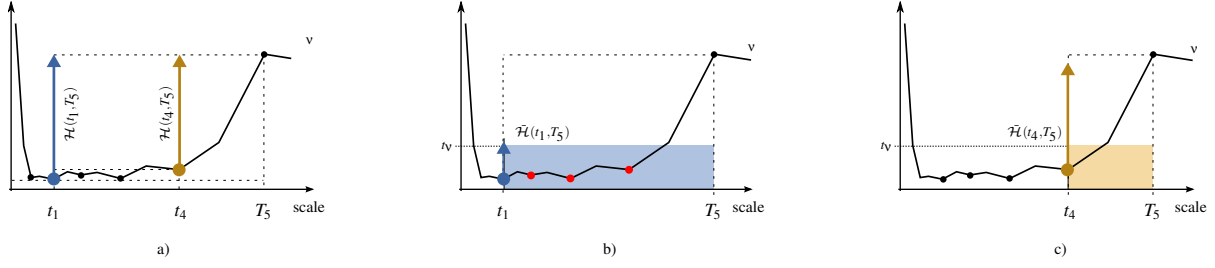
## 4.2. Point-wise pertinent scale weighting

Given a point  $\mathbf{p} \in P$  and its geometric variation  $v(t)$ , our primary goal at this stage of the algorithm is to define a measure  $\beta(\mathbf{p}, t)$  of how well  $\mathbf{p}$  belongs to a pertinent feature at scale  $t$ . A popular approach to measure the importance of features in a signal  $f$  is to consider the *homological persistence*  $\mathcal{H}_f(t_1, t_2) = f(t_2) - f(t_1)$  of a pair of scales  $t_1, t_2$  corresponding to a pair of minimum-maximum in  $f$  [ELZ00, RKG\*11]. The notion of homological persistence builds on the intuitive insight that pertinent features occur at scales exhibiting small variation, while a high variation indicates a transition between two features at different scales. A high persistence  $\mathcal{H}$  characterizes a pair composed of a stable scale  $t_1$  followed by an unstable scale  $t_2$ , which is thus characteristic of the presence of a pertinent feature at  $t_1$ . The homological persistence is usually employed to extract critical points by restricting the study to nested pairs.

Since our objective is different here, we do not want to restrict the set of pairs from topological considerations, but rather consider all possible pairs and defer the precise extraction until the analysis of the entire ROI. Nonetheless, in order to ease this last detection step, it is important to provide both discriminating and consistent weights. We accomplish this by combining  $\mathcal{H}$  with other heuristics penalizing unwanted stable scales. Let  $\{t_i\}$  and  $\{T_j\}$  denote the set of scales corresponding to minima and maxima of  $v$  respectively. We propose to estimate the pertinence of a pair  $(t_i, T_j)$  with  $t_i < T_j$  using the following extended measure:

$$\begin{aligned} \tilde{\mathcal{H}}(t_i, T_j) = & (\min(H_v, v(T_j)) - v(t_i)) \\ & \cdot \phi(v(t_i)/h_v) \\ & \cdot \prod_{k>i, t_k < T_j} (1 - \phi(v(t_k)/h_v)), \end{aligned} \quad (5)$$

where  $\phi$  is a decreasing smooth-step function as defined in equation 2, and  $h_v, H_v$  are two threshold values. The first factor corresponds to the homological persistence value with the difference that  $v$  is clamped to the maximal value  $H_v$ . Indeed,  $v$  sums up differential quantities which can reach arbitrary large values. Such maxima would be given to much importance thus hiding the other relevant features. The second factor smoothly vanishes when  $v(t_i) > h_v$ , resulting in pairs starting from a *too unstable minimum* to get completely discarded. Such pairs can be considered as outliers, and removing them makes further analysis easier. This factor also allows us to give more importance to the magnitude of the minimum, and less to the maximum. Finally, the third factor penalizes pairs containing strong minima in the interval they define. It can be seen as a fuzzy implementation of the topological pair selection done in classical homological persistence. Our approach thus offers more robustness to noise. To understand its rational, let us consider the plot of Figure 5, which exhibits one, very long interval of stable scales. Due to noise, this interval includes many local minima. The minimum of these minima, as selected by classical homological persistence, can appear at very inconsistent scales from

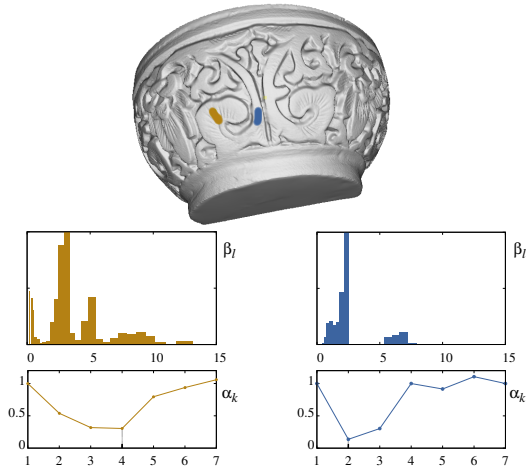


**Figure 5:** (a) Classical homological persistence will pair the global minimum of  $v$  at  $t_1$  with the maximum at  $T_5$ . (b) Our novel measure  $\bar{\mathcal{H}}$  significantly reduces the weight of the pair  $(t_1, T_5)$  because of presence of numerous strong minima lower than  $t_v$  (shown in red) in the respective interval. (c) On the other hand, the weight of the pair formed with the last minimum  $t_4$  remains almost unchanged.

one point on the shape to another. Without this third factor, all these minima would be considered as one pertinent scale, this making further analysis more difficult. In contrast, our heuristics will strongly favor the minimum at the end of the stable interval, leading to both a discriminating and consistent behavior.

For all our experiments, we used  $h_v = 0.2$ , and  $H_v = 2$ . We emphasize that since  $v$  is scale independent, these threshold values can be set once and for all, regardless of the characteristic of the input data sets. Finally, each scale that does not correspond to a local minimum is given a weight of zero, while the others are set to the sum of  $\bar{\mathcal{H}}$ , that is:

$$\beta(\mathbf{p}, t_i) = \sum_{j, T_j > t_i} \bar{\mathcal{H}}(t_i, T_j) .$$



**Figure 6:** Illustration of the automatic extraction of the number of pertinent scales for two different ROI on a vase model (top). Middle-row shows the result  $\beta_l$  of the accumulation of the pertinence weights for each level  $l$ . Bottom-row shows the measure  $\alpha_k$  of the relative improvement of having  $k$  versus  $k-1$  clusters. The estimated ideal number of clusters corresponds the minimum, that is 4 and 2 respectively.

#### 4.3. Region-based pertinent scale extraction

Given a region of interest  $Q \subset P$ , our goal is now to extract the scales of the most prominent features of  $Q$ . To this end, we aggregate the previous point-wise pertinence measures by accumulating them for each scale level  $l \in [0, L]$ . This leads to a discrete set of values:

$$\beta_l = \sum_{\mathbf{p} \in Q} \beta(\mathbf{p}, t_l) , \quad (6)$$

which can be seen as a distribution over the scale axis representing the likelihood of each scale to be a pertinent one in the ROI. An example of this distribution is illustrated in Figure 6 on a vase model.

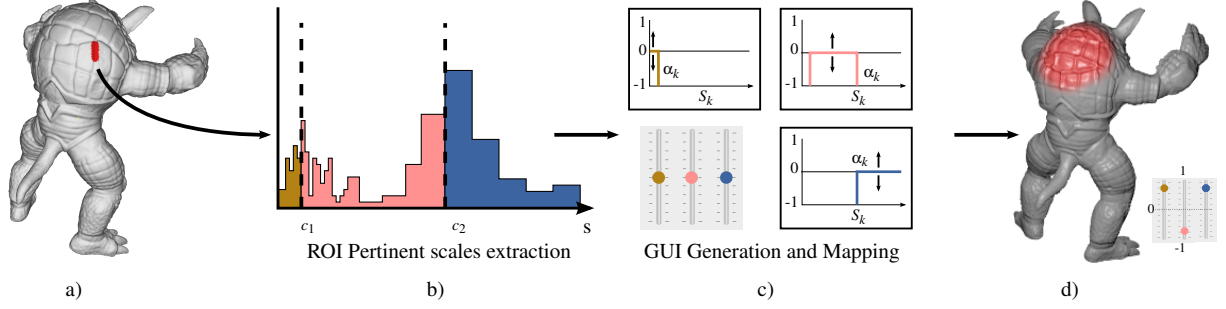
The extraction of the pertinent scales thus boils down to a problem of clustering for which we propose to employ the well known  $k$ -means algorithm. Let us start by defining our objective function  $s_k$  as:

$$s_k = \sum_{j=1}^k \sum_{l=0}^{L-1} \beta_l \cdot (t_l - c_j)^2 , \quad (7)$$

where  $c_j$  is the centroid of the  $j^{\text{th}}$  cluster, and  $k$  is the number of clusters. Minimizing  $s_k$  by the  $k$ -means algorithm implies two major difficulties. Firstly, we have to provide the number of clusters  $k$  which is in our case the unknown number of pertinent scales. Secondly, the convergence of the algorithm and the quality of the result highly depends on the choice of the initial cluster positions.

Regarding the second issue, we adapted the  $k$ -means++ algorithm [AV07] to take into account our weighted data. The centers are computed one after the other using the following procedure:

1. The center  $c_1$  is randomly selected among the discrete scales  $t_l$  with a probability proportional to  $\beta_l$ .
2. For each scale  $t_l$ , compute the distance  $d(t_l)$  between  $t_l$  and the nearest center that has already been computed.
3. The next center is randomly selected among the discrete scales  $t_l$  with a probability proportional to  $\beta_l d(t_l)^2$ .



**Figure 7:** Overview of an interactive editing session: starting from a selected ROI (a), our system automatically extract the pertinent scales interactively (b). A graphic equalizer is generated and mapped to scale intervals in the multi-scale decomposition (c). (d) The user can select any region containing similar feature scales on the model and edit them using the sliders, all the changes being applied in real-time.

Steps 2 and 3 are repeated until all centers have been computed. Since we strived to build the weights  $\beta_l$  to be as discriminating as possible, this initialization of the centers often leads to very good guesses.

Still remains the problem of automatically choosing the number  $k$  of clusters. We accomplish this challenging task using an incremental approach adapted from the work of Pham et al. [PDN05]. The general principle is to perform the above  $k$ -means++ clustering for a predefined range of  $k \in [1, k_{max}]$  and evaluate the relative benefit of an additional cluster. The insight is that the optimal number of clusters corresponds to the case for which adding one cluster yields the best improvement. This is accomplished by computing the ratio  $\alpha_k = \frac{s_k}{s_{k-1}}$ , and the optimal number of clusters is thus the one for which  $\alpha_k$  is minimal. This selection procedure is illustrated in Figure 6. For all our experiments we took  $k_{max} = 7$  which is a rather conservative choice.

## 5. Feature editing

We now have all the ingredients to design an intuitive tool for the manipulation of surface features. In pre-process, the input point cloud or mesh vertices is augmented by the multi-scale decomposition described in section 3. During the interactive editing session, the user starts by selecting a ROI using a brush (Figure 7a). This ROI is used to learn the scales of the pertinent features as in section 4 (Figure 7b). Therefore, this ROI do not have to correspond to the entire region the user wants to edit: it only has to be a good representative and it can thus be significantly smaller.

This analysis is carried out in real-time from which we build a graphical equalizer as traditionally used to control frequency bands of a signal (Figure 7c). The main difference here is that our analysis allows us to generate an equalizer with the correct number of sliders such that the band controlled by each slider is precisely mapped to a pertinent feature. More precisely, the result of the analysis is a set

of  $k$  pertinent scales  $c_1, \dots, c_k$  ordered such that  $c_j > c_{j+1}$ . We generate  $k + 1$  sliders controlling a partition of bands  $b_0, \dots, b_k$  defined as follow:

$$b_0 = [t_0, c_1], \dots, b_j = [c_j, c_{j+1}], \dots, b_k = [c_k, 0]. \quad (8)$$

The last band controls the finest details or noise. The value of the  $j^{\text{th}}$  slider is thus directly mapped to the subset of coefficients  $\{a_t; t_l \in b_j\} \subset \mathbf{a}$  used to reconstruct the surface from its multi-scale decomposition (equation 3).

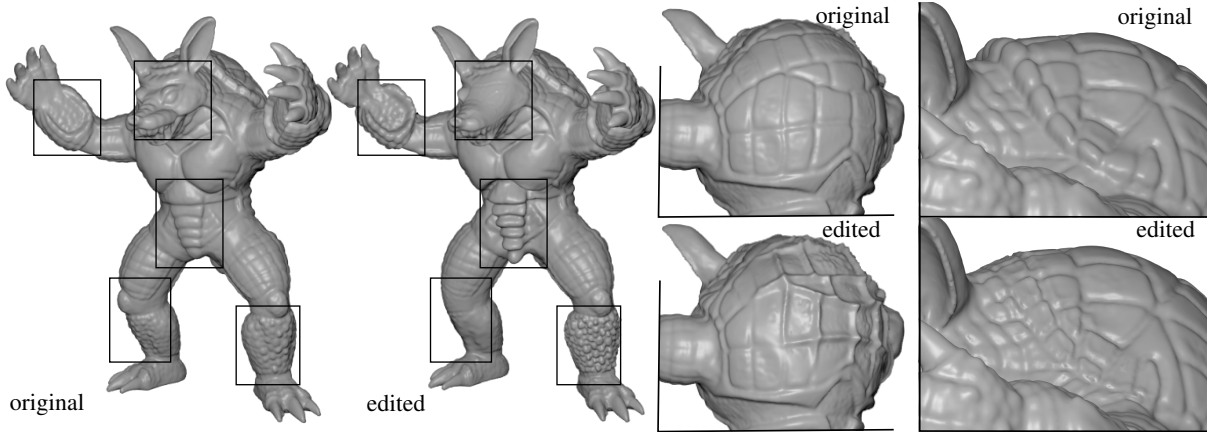
In practice, this editing can be localized by selecting any region having a similar structure as the ROI (Figure 7d). In that case, each point or vertex has to store its own vector of reconstruction coefficients  $\mathbf{a}$ . Moreover, in order to ensure a continuous reconstruction, the effect of the sliders within the region has to smoothly vanish at the boundary of the selected region.

A typical editing session is depicted in Figure 7: a small ROI of the Armadillo's shell is used to quickly learn its feature scales, and then the entire shell is edited by manipulating a single slide to invert the cavities.

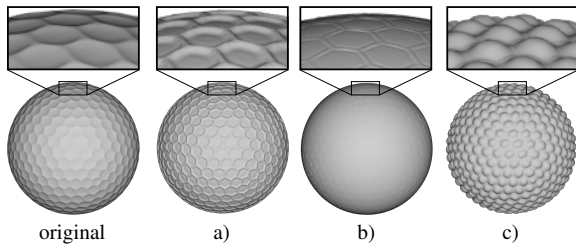
## 6. Results

In our prototype implementation, the multi-scale decomposition is precomputed using a  $kd$ -tree to accelerate the neighbor search. The fitting at scale  $t$  is performed by collecting all the neighbors within the distance  $t$  of the current point. This preprocess takes from several seconds to a few minutes depending on the size of the point clouds, number of scales, and size of the largest scales. We believe that multiple order of magnitude could be saved using both a GPU implementation and by iteratively down-sampling the input point cloud to compute the largest scales [PKG06].

For all our experiments we spread between 10 and 30 discrete levels according to a logarithmic distribution in a scale interval ranging from  $2 \times$  to  $30 \times$  the average point spacing. We report in Table 1 timings measured to process the ROI



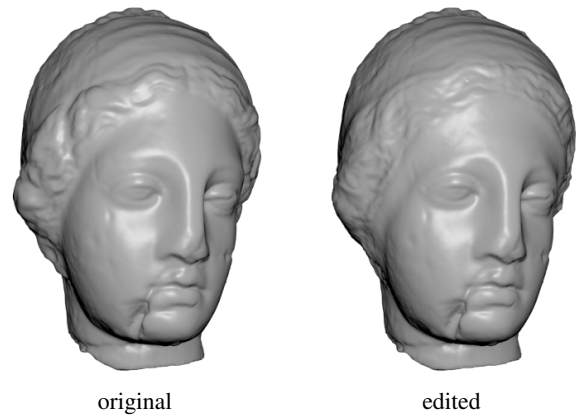
**Figure 8:** Left: Result of an editing session on the Armadillo model. Right: Editing of medium scale details. For each of these edits, only one slider has been manipulated.



**Figure 9:** Edition of a synthetic golfball model. a) Boost of the small scales corresponding to the cell edges. b) Removal of the cell concavities while preserving the sharp edges of the borders. c) Inversion of cell concavities.

of different models, which includes the computation of the geometric variation, the generation of the distribution  $\beta_l$  and the extraction of the pertinent scales. This table also includes times required to update the whole objects when moving sliders, which constitutes a worst-case scenario as one usually edits only a sub-part of the object. In almost all our experiments both the ROI analysis and the geometry editing are performed in real time. When the ROI is too huge, as in the 5<sup>th</sup> row corresponding to a ROI of 453k points, the analysis requires about half a minute. In such a case, we believe that multiple order of magnitude could be saved by iteratively down-sampling the input point cloud to compute the largest scales [PKG06].

Various results of our editing tools are provided in Figure 9 on a synthetic object, in Figure 1 on a large sculpted mesh, and in Figures 8 and 10 on scanned objects. All these results include both smoothing and enhancement of both small and medium scale details. We emphasize that smoothing a medium scale feature while preserving finer details, as for instance in Figures 10 and 8-right, is impossible to



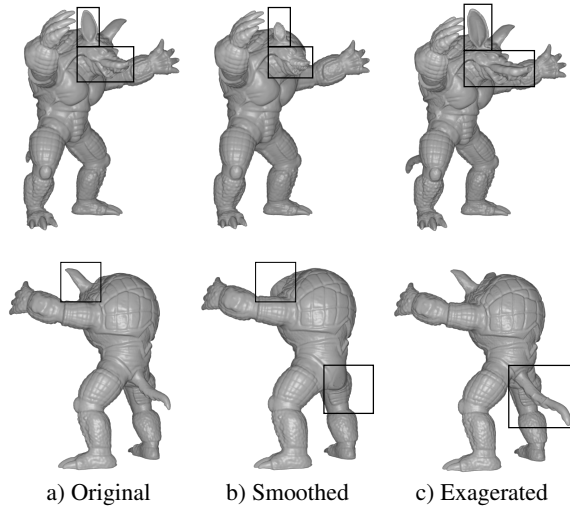
**Figure 10:** Edition of the Igea model headband: a medium-scale structure is flattened while the small details of the hairs are preserved.

achieve with standard smoothing tools. Figure 11 demonstrates the ability of our approach to detect and edit large scale features. In Figure 12 we used our tool on a raw noisy point cloud acquired by a Kinect to remove the levels of the first detected scale, corresponding to the acquisition noise. We believe that such an approach could be automatized to clean noisy point clouds without user supervision.

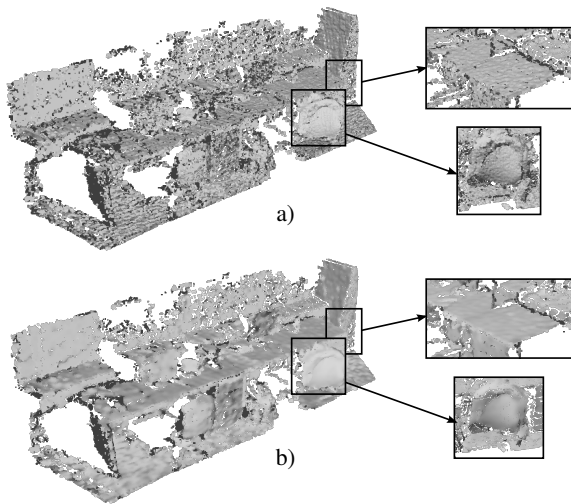
## 7. Conclusion

In this work, we have presented a novel technique for the automatic extraction of the pertinent scales of surface features. Our approach exploits recent advances in multi-scale analysis of point clouds and does not require any connectivity information. We showed how to turn the problem of scale detection into a more general clustering problem thus





**Figure 11:** Editing of large scale features on the Armadillo.



**Figure 12:** Noise removal on a noisy Kinect point-clouds.

allowing us to take advantage of the extended literature on the subject. We demonstrated the robustness and utility of our approach through a novel editing tool of surface features combining our detecting mechanism with a compatible multi-scale point-cloud decomposition.

One of the limitations of multi-scale representations is the relatively high memory overhead to store all intermediate levels. Therefore, it would make sense to study the possibility to exploit meshless multi-resolution structures which are much lighter [WGE\*04]. Our current approach is also limited to features exhibiting a constant scale, and it would be interesting to extend it to track scale-varying features as the headband of Figure 10. Finally, another venue for future

	ROI		Preprocessing time (ms)				Editing	
	#vert.	#scales	GLS	$\beta_l$	clust.	total	#vert.	time
	245	30	20	7	5	32	199k	47
	321	30	26	9	1	36	199k	47
	610	25	66	12	1	79	172k	33
	273	25	22	7	5	34	172k	33
	453k	10	27985	781	2	28768	1225k	76
	9.5k	10	60	22	1	83	1225k	76
	1.4k	25	189	18	3	210	333k	71
	2.8k	25	655	30	3	688	333k	71

**Table 1:** Timings of the different subroutines involved in our approach to analyze a given ROI and edit the object using sliders. The ROI is preprocessed at the selection time, then the geometry of the entire object can be edited in real time even for large objects.

work would be to investigate whether our pertinent scale detection method could be adapted to the automatic selection of eigenfunctions basis in spectral geometry processing.

## Acknowledgments

We thank Moos Hueting, Aron Monszpart, Clement Goudard and Gwyneth Bradbury for discussions and proofreading. This work was supported in part by the ERC Starting Grant SmartGeometry (StG-2013-335373), and partially supported by the Arc6.

## References

- [AA03] ADAMSON A., ALEXA M.: Approximating and intersecting surfaces from points. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2003), SGP '03, pp. 230–239. 2
- [AA04] ALEXA M., ADAMSON A.: On normals and projection operators for surfaces defined by point sets. In *In Eurographics Symp. on Point-Based Graphics* (2004), pp. 149–155. 3
- [AV07] ARTHUR D., VASSILVITSKII S.: K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (2007), SODA '07, pp. 1027–1035. 5
- [BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *Visualization and Computer Graphics, IEEE Transactions on* 14, 1 (Jan 2008), 213–230. 1
- [ELZ00] EDELSBRUNNER H., LETSCHER D., ZOMORODIAN A.: Topological persistence and simplification. In *Proceedings of*

- the 41st Annual Symposium on Foundations of Computer Science (2000), FOCS '00, pp. 454–. [2](#), [4](#)
- [GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. *ACM Trans. Graph.* 26, 3 (July 2007). [2](#)
- [GGG08] GUENNEBAUD G., GERMANN M., GROSS M. H.: Dynamic sampling and rendering of algebraic point set surfaces. *Comput. Graph. Forum* 27, 2 (2008), 653–662. [3](#)
- [GSS99] GUSKOV I., SWELDENS W., SCHRÖDER P.: Multiresolution signal processing for meshes. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (1999), SIGGRAPH '99, pp. 325–334. [2](#)
- [JBPS11] JACOBSON A., BARAN I., POPOVIĆ J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 4 (July 2011), 78:1–78:8. [1](#)
- [KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (1998), SIGGRAPH '98, pp. 105–114. [2](#)
- [Lev06] LEVY B.: Laplace-beltrami eigenfunctions: Towards an algorithm that understands geometry. In *IEEE International Conference on Shape Modeling and Applications, invited talk* (2006). [2](#)
- [LLCO08] LIPMAN Y., LEVIN D., COHEN-OR D.: Green coordinates. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 78:1–78:10. [1](#)
- [LPG12] LIU Y., PRABHAKARAN B., GUO X.: Point-based manifold harmonics. *IEEE Transactions on Visualization and Computer Graphics* 18, 10 (2012), 1693–1703. [2](#)
- [MGB\*12] MELLADO N., GUENNEBAUD G., BARLA P., REUTER P., SCHLICK C.: Growing least squares for the analysis of manifolds in scale-space. vol. 31. [2](#), [3](#), [4](#)
- [NVT\*14] NEUMANN T., VARANASI K., THEOBALT C., MAGNOR M., WACKER M.: Compressed manifold modes for mesh processing. [2](#)
- [PDN05] PHAM D. T., DIMOV S. S., NGUYEN C.: Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 219, 1 (2005), 103–119. [6](#)
- [PKG03] PAULY M., KEISER R., GROSS M.: Multi-scale feature extraction on point-sampled surfaces. *Computer Graphics Forum* 22, 3 (2003), 281–289. [4](#)
- [PKG06] PAULY M., KOBBELT L. P., GROSS M.: Point-based multiscale surface representation. *ACM Trans. Graph.* 25, 2 (Apr. 2006), 177–193. [2](#), [6](#), [7](#)
- [PKKG03] PAULY M., KEISER R., KOBBELT L. P., GROSS M.: Shape modeling with point-sampled geometry. *ACM Trans. Graph.* 22, 3 (July 2003), 641–650. [1](#)
- [PLG13] PATANE G., LI X. S., GU D. X.: Surface- and volume-based techniques for shape modeling and analysis. In *SIGGRAPH Asia 2013 Courses* (2013), SA '13, pp. 1–65. [1](#)
- [PS13] PATANÈ G., SPAGNUOLO M.: Heat diffusion kernel and distance on surface meshes and point sets. *Computers & Graphics* 37, 6 (2013), 676–686. [2](#)
- [RKG\*11] REININGHAUS J., KOTAVA N., GUNTHER D., KASTEN J., HAGEN H., HOTZ I.: A scale space based persistence measure for critical points in 2d scalar fields. *Visualization and Computer Graphics, IEEE Transactions on* 17, 12 (Dec 2011), 2045–2052. [4](#)
- [SOG09] SUN J., OVSIANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), SGP '09, pp. 1383–1392. [2](#)
- [VL08] VALLET B., LÉVY B.: Spectral geometry processing with manifold harmonics. *Computer Graphics Forum (Proceedings Eurographics)* (2008). [2](#)
- [WGE\*04] WASCHBÜSCH M., GROSS M., EBERHARD F., LAMBORAY E., WÜRLIN S.: Progressive compression of point-sampled models. In *Proceedings of the First Eurographics Conference on Point-Based Graphics* (2004), SPBG'04, pp. 95–103. [8](#)
- [ZPKG02] ZWICKER M., PAULY M., KNOLL O., GROSS M.: Pointshop 3d: An interactive system for point-based surface editing. *ACM Trans. Graph.* 21, 3 (July 2002), 322–329. [1](#)
- [ZVKD10] ZHANG H., VAN KAICK O., DYER R.: Spectral mesh processing. *Computer Graphics Forum* 29, 6 (2010), 1865–1894. [2](#)