

Markov Random Fields for Improving 3D Mesh Analysis and Segmentation

Guillaume Lavoué[†] and Christian Wolf[‡]

LIRIS UMR 5205 CNRS
INSA-Lyon, F-69621, France

Abstract

Mesh analysis and clustering have become important issues in order to improve the efficiency of common processing operations like compression, watermarking or simplification. In this context we present a new method for clustering / labeling a 3D mesh given any field of scalar values associated with its vertices (curvature, density, roughness etc.). Our algorithm is based on Markov Random Fields, graphical probabilistic models. This Bayesian framework allows (1) to integrate both the attributes and the geometry in the clustering, and (2) to obtain an optimal global solution using only local interactions, due to the Markov property of the random field. We have defined new observation and prior models for 3D meshes, adapted from image processing which achieve very good results in terms of spatial coherency of the labeling. All model parameters are estimated, resulting in a fully automatic process (the only required parameter is the number of clusters) which works in reasonable time (several seconds).

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling I.4.6 [Image Processing And Computer Vision]: Segmentation G.3 [Probability And Statistics]: Markov processes

1. Introduction

Technological advances in the fields of telecommunication, graphic hardware and geometry processing during the last decade, have contributed to an evolution of the digital data being manipulated and transmitted over the Internet. Nowadays, static and dynamic three-dimensional meshes constitute the emerging multimedia content. Accordingly, 3D models are subject to a wide variety of processing operations such as compression, simplification, approximation, indexing or watermarking.

A critical issue to improve the efficiency of these processes is to really *understand* the 3D object which is behind the polygonal mesh. To reach that goal the solution is to conduct an in-depth analysis of the shape (in terms of geometric criteria) and/or to provide a structure using some partitioning/segmentation algorithms. This analysis and/or partitioning can then greatly improve the efficiency of the applica-

tions cited above. For instance, some local measures of the shape (like roughness or curvature) can be advantageously used to improve compression or watermarking algorithms by concentrating artifacts on parts of the object which exhibit a high masking degree. An other example is the use of a prior segmentation of the mesh to facilitate remeshing [CSAD04] or approximation [LDB07]. Local analysis can also be used to provide shape signature for partial shape retrieval and indexing [TVD07, GCO06].

This shape analysis leads to the creation of different kinds of attributes, usually associated to the vertices of the mesh (see fig. 1): Curvature [CSM03], roughness [Lav07], saliency [LVJ05], crest or ridge [LZH*07], etc. Moreover, in some specific applications, like in scientific visualization, other external attributes can be associated with the mesh elements, like temperature or density.

In order to be properly used in further processes (compression, indexing, watermarking etc.) or to correctly lead some segmentation/decomposition algorithms, these attributes (intrinsic or external) have to be properly filtered, classified or quantized using clustering algorithms. Clus-

[†] e-mail: glavoue@liris.cnrs.fr

[‡] e-mail: cwolf@liris.cnrs.fr

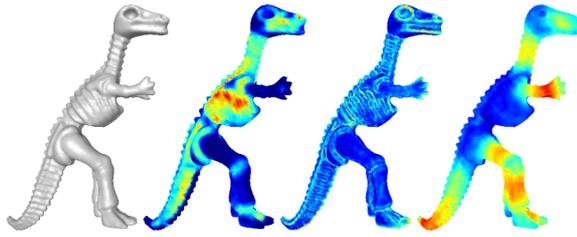


Figure 1: Some examples of vertex attributes, for the Dyno mesh (42K vertices). From left to right: Original model, roughness, local curvature (geodesic radius = 1% of the bounding box max length), global curvature (geodesic radius = 5%). The values are represented from blue (low) to red (high).

tering consists in associating to each vertex an appropriate discrete label (among a given set) according to its attribute value (which might be a scalar or a vector). Two important issues have to be resolved to conduct a good clustering:

- It seems critical to consider, together, both geometry and attribute information in the clustering process. Indeed, a lot of simple techniques, as for instance thresholding or K-Means clustering, exploit information from feature space only to classify each vertex; this may lead to noise or artifacts that can disturb the further processes. For example, in the segmentation algorithm described in [LDB05], vertices are clustered only according to the attribute (curvature) values (using K-Means in the curvature space) then a region growing process uses this attribute-only clustering to create spatial regions. However it produces a noisy over-segmentation that necessitates a further complex merging process. A better clustering taking into account some geometric constraints would have probably greatly improved the result of the region growing.
- An other important issue is to process this clustering, in a *global* way. Indeed, a lot of existing clustering processes are greedy, depend on initial seed positions and thus may result in non-optimal solutions, which are consequently not very robust. A solution is to conduct the clustering in a global way.

In this paper we investigate the use of probabilistic graphical models to resolve both issues: incorporate the spatial dependencies between the vertices into the clustering process, while providing a globally optimal solution. In particular, we base our framework on Markov Gibbs Random Fields (MRF). The main idea is the following: for a given number of clusters and a 3D mesh associated with any kind of attributes or features (curvature, roughness, saliency etc.) our approach provides, for each vertex, the appropriate cluster label. These labels will be considered to be optimal since they maximize a global probability over the entire mesh, taking into account both attribute values and spatial

relations (geometry). This framework can be easily adapted to any kind of attributes (scalar or vector), located on any kind of mesh element (vertex, edge or face).

The paper is organized as follows: section 2 presents some existing related works, while sections 3, 4 and 5, respectively, detail our adaptation of the Markov paradigm to 3D meshes, our prior and observation models and the global simulated annealing optimization algorithm. Section 6 presents the parameter estimation and finally section 7 illustrates some experiments and results on several meshes with different attributes and different numbers of labels.

2. Related Work

2.1. Mesh clustering and segmentation

In this paper we differentiate *clustering* and *segmentation*. Clustering associates with each mesh element (vertex for instance) an appropriate cluster label by taking into account some attribute values. Typically this process considers only the attribute space and allows to quantize, or filter these values for further use (compression, segmentation etc.). The principal methods are K-Means, uniform quantization or thresholding.

On the contrary, mesh segmentation provides a decomposition into several connected spatial regions: The facets are regrouped in order to obtain regions (usually homeomorphic to a disk) sharing some common properties. Some authors [SSGH01, CSAD04] use planarity criteria to incorporate faces in the regions while others [LPRM02, LDB05] rather consider homogeneous curvature properties. Some higher level algorithms consider feature points [KLT05], skeleton [TVD06], graph [KT03], spectral analysis [LZ04]. A lot of segmentation algorithms exist for 3D meshes, a recent state of the art can be found in [AKM*06].

The main problem of the pure clustering approaches is that they only consider the attribute space, without any geometric constraints. At the opposite, the existing segmentation algorithms decompose the mesh only according to its geometry; no additional attribute data can be introduced in the algorithms to modify the results. Moreover, except some recent algorithms [CSAD04], most of them are greedy and thus can fall into non optimal local minima.

Our MRF based approach allows to cluster a 3D mesh by taking into account both attribute values and geometry, moreover it is a global approach that provides an optimized solution. Besides, several segmentation algorithms are based on *a priori* clustering [LDB05, MDKK06, JM07], hence improving the clustering with geometric constraints should greatly improve the corresponding segmentations. The very recent approach from Shamir et al. [SSCO06] also provides a mixed attribute-geometry clustering framework by adapting the Meanshift algorithm to 3D meshes. They obtain very good results, however, processing time is quite long (several minutes), whereas our method is faster (several seconds).

2.2. Markov Random Fields

Markov Random Fields have a long history, we refer the reader to the seminal work by Geman and Geman [GG84] and to the book written by Li for a large yet profound overview of the theory [Li01]. They have been extensively used in image processing, particularly for segmentation and image restoration, even very recently [SC06, WC07]. In particular this Bayesian framework is employed to combine models of the observation process (i.e. the likelihood of the observation given a label configuration) with models of the spatial interaction (the prior knowledge).

To our knowledge, only two authors have investigated MRF for 3D mesh processing: Willis et al. [WSC04] for surface deformation and Andersen [And07] for mesh smoothing.

3. The Markovian framework

Markov random fields [Li01] are graphical models used to find the optimal labeling of the nodes of a graph — optimal in a sense which shall be defined later. Generally speaking, the graph may be regular or irregular and the labels may be continuous or discrete. Regular graphs are frequently used in image processing [GG84]. In our case, the graph corresponds to the irregular graphical structure of the considered mesh, we therefore consider an undirected graph $\mathcal{G} = \{G, E\}$, where G is the set of vertices (sites) of the mesh and E is the set of edges of the mesh. Our objective is thus to assign the most correct label to each vertex of the mesh (i.e. each site of the graph).

Markov random fields are also probabilistic models, they assign probabilities to the different possible results, i.e. one probability to each possible labeling of the set of vertices. Therefore, each site (i.e. vertex) $s \in G$ is assigned a discrete random variable X_s taking values from the finite set Λ , $C = |\Lambda|$ denoting the number of classes. X_G , or short X denotes the field of random variables of the graph. The space of all possible configurations of the field X is denoted as $\Omega = \Lambda^{|G|}$. As usual, uppercase letters denote random variables or fields of random variables and lower case letters denote realizations of values of random variables or of fields of random values. In particular, $P(X = x)$ will be abbreviated as $P(x)$ when it is convenient.

Probabilistic graphical models take into account the connectivity of the graph. Although a globally optimal solution is searched, i.e. the best global labeling is searched, the probability $P(X = x)$ is defined through local properties, which is reflected by the Markov property of the random field: A field X of random variables is a MRF if and only if

$$P(X=x) > 0 \quad \forall x \in \Omega \text{ and} \quad (1)$$

$$P(X_s=x_s | X_r=x_r, r \neq s) = P(X_s=x_s | X_r=x_r, r \in N_s)$$

where N_s is the neighborhood of the site s . In other words, the variable of a site s is conditionally independent of the

variable of another site r given the neighbor variables of site s . Note, that conditional independence does not mean independence. Two variables X_s and X_r are dependent even when they are separated by a very long minimal path in the graph; however, conditional independence means that the knowledge of x_r does not provide additional information for the inference of x_s if the realizations of x_{N_s} are known.

On a graph, each neighborhood defines a set of cliques, where a clique is fully connected sub graph. For a triangular mesh, there exist 3 types of cliques: vertex (1-site clique), edge (2-site clique) and triangle (3-site clique). According to the Hammersley-Cifford theorem [HC68] [Bes74], the joint probability density functions of MRFs are equivalent to Gibbs distributions defined on the maxima cliques, i.e. are of the form

$$P(x) = \frac{1}{Z} \exp\{-U(x)/T\} \quad (2)$$

where $Z = \sum_x e^{-U(x)/T}$ is a normalization constant, T is a temperature factor which can be assumed to be 1 for simplicity, $U(x) = \sum_{c \in \mathcal{C}} V_c(x)$ is a user defined energy function defined on local interactions in the cliques, \mathcal{C} is the set of all possible cliques of the graph and $V_c(x)$ is the energy potential for the realization x defined on the single clique c .

The probability $P(x)$ encodes the *a priori* knowledge on the result (independent of the actual observations) - it gives us information whether a given solution is probable. The application dependent knowledge is injected through the user defined energy potentials defined for each clique labeling. Commonly used energy potentials favor a classification creating homogeneous regions (see section 4). Concretely that is a way to inject spatial constraints in the labeling.

The segmentation result depends on observations measured at each site, denoted as known random variables Y_s . Concretely, these observations correspond to the values of the attributes at each vertex. We suppose the following widely used statistical assumptions on these variables (If required, these assumptions can be relaxed by posing the problem in the context of the conditional random field framework): each observed variable Y_s is related to a hidden variable X_s and is conditionally independent of the other variables given the realization of the related hidden variable:

$$p(y_s|x) = p(y_s|x_s) \quad \forall s \in G$$

$$p(y|x) = \prod_{s \in G} p(y_s|x_s) \quad (3)$$

Properties 1 and 3 are illustrated in the dependency graph shown in figure 2, where each shaded observed variable is connected to its corresponding non shaded hidden variable only.

The probability $P(x)$ defined by the MRF is independent of the observations Y and can be interpreted as the prior probability in a Bayesian framework, completed by the like-

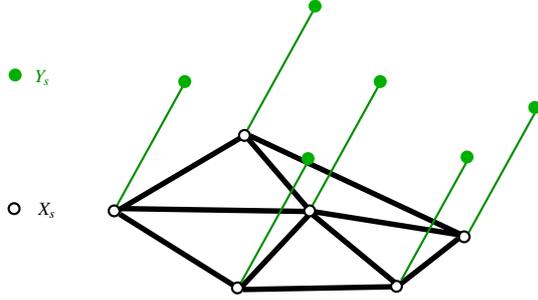


Figure 2: The dependency graph of the markov random field. The shaded nodes are observed, the empty nodes are hidden.

likelihood of the observations given the hidden labels $p(y|x)$ defined in (3). This later probability depends on the observation model, which we will define in section 4. We are interested in inferring the most probable realization of the hidden variables given the observed nodes, which can be obtained using Bayes' rule:

$$P(x|y) = \frac{P(x)p(y|x)}{p(y)} \quad (4)$$

$$\hat{x} = \arg \max_x P(x)p(y|x) \quad (5)$$

This labeling \hat{x} , known as the *maximum a posteriori* or MAP estimate, will be considered as the optimal clustering for our mesh, given its attributes. The result depends on the prior model and on the observation model which respectively drive the weight of the attribute and the weight of the geometry. If we consider only the observation model, we obtain a simple classification in the attribute space (like a K-Means algorithm); the prior model allows to inject some spatial constraints in the clustering. The next section details the construction of these models.

4. Prior and observation model

As mentioned above, the role of the prior model is to regularize the classification decisions, thus favoring homogeneous regions. For this purpose we modified the multi-level logistic model [Li01] whose energy potential functions are defined as:

$$U(x) = \sum_{s \in G} \sum_{l \in \Lambda} \alpha_l \delta_{l, x_s} + \sum_{c \in \mathcal{C}_3} \beta \gamma(c) \quad (6)$$

where the α_i ($i = 1 \dots C$) and β are parameters, \mathcal{C}_3 is the set of 3-site cliques of the graph, i.e. the set of triangles of the mesh, $\delta_{i,j}$ is the Kronecker delta given as

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \quad (7)$$

and $\gamma(c)$ is a function favoring triangles with homogeneous labels, given as:

$$\gamma(c) = - \sum_{(s,s') \in c \times c, s \neq s'} \delta_{x_s, x_{s'}} \quad (8)$$

Each parameter α_i controls the prior probability of a given label i whereas the parameter β controls the amount of smoothness of the result.

The observation model is a (possibly multi-variate) Gaussian one, resulting in the following probability density function:

$$p(y|x) = \prod_{s \in G} p(y_s | x_s) = \prod_{s \in G} \mathcal{N}(y_s ; \mu_{x_s}, \Sigma_{x_s}) \quad (9)$$

where μ_{x_s} and Σ_{x_s} are, respectively, the mean vector and the covariance matrix of class x_s . In our experiments, scalar observations were used, thus a single mean value and a single variance is required for each class.

The combination of MRF prior model $P(x)$ (a distribution) and likelihood $p(y|x)$ (a density) can be seen as a new MRF defining the joint probability density $p(x,y)$ on a new graph: the dependency graph shown in figure 2. The new graph contains the original graph (the mesh) as a subgraph as well as additional sites (the observed variables) and additional 2-site cliques for each pair X_s and Y_s with the following potential functions:

$$U(x_s, y_s) \propto \ln[p(y_s | x_s)] \quad (10)$$

5. Optimization

To calculate the estimate \hat{x} , the maximization in (5) needs to be performed. Unfortunately, the function is not convex and standard gradient descent methods will most likely return a non global solution. Simulated Annealing has been proven to return the global optimum under certain conditions [GG84].

Simulated Annealing received its name from physical processes, which decrease temperatures to allow particles (e.g. atoms in an alloy) to relax into a low energy configuration. Similarly, for the optimization of a non-convex function, the simulated annealing process lowers a (virtual) temperature factor. During the annealing process, the labels of the vertices are changed in order to bring the estimations closer to the model. However, a certain amount of randomness is included in the optimization process, which allows the system to change to more unfavorable estimates at certain times. This amount of randomness depends on the temperature factor: it is set relatively high at the beginning to allow the system to “jump” out of local minima, and is gradually lowered together with the temperature factor.

More precisely, during the annealing process, for each vertex the energy potential is calculated before and after randomly choosing a new state (i.e. a new label). The decision whether to keep the new state or not is based on the value

$$q = e^{-\Delta/T} \quad (11)$$

where Δ is the difference of the posterior energy potentials $U(x_s, x_{N_s}, y_s)$ of site s before and after the change of x_s :

$$\begin{aligned} U(x_s, x_{N_s}, y_s) &= U(x_s, x_{N_s}) + U(x_s, y_s) \\ &= \sum_{l \in \Lambda} \alpha_l \delta_{l, x_s} + \sum_{c \in \mathcal{C}_3: s \in c} \beta \gamma(c) \\ &+ (y_s - \mu_{x_s})^T \Sigma_{x_s}^{-1} (y_s - \mu_{x_s}) \end{aligned} \quad (12)$$

If $q > 1$ then the change is favorable and accepted. If $q < 1$, i.e. if the solution is “worse”, then it is accepted with probability q , which depends on the global temperature factor T . The gradual decrease in temperature assures that this is done less and less frequent. For the cooling schedule we used the suggestions in [DHS00] (page 356), where the temperature T is set to

$$T^{(i)} = T^{(1)} \cdot K^{i-1} \quad (13)$$

where K is a constant controlling the speed of the cooling process and i denotes the current iteration. The start temperature must be sufficiently high to switch to energetically very unfavorable states with a certain probability. It can be calculated as a function of the maximum possible potential differences, as we did in previous work [WD02].

6. Parameter estimation

Since realizations of the label fields X are not available, the parameters of the prior model and the observation model must be estimated from the observed data or from intermediate estimations of the label fields. In this work we chose to estimate the parameters in a supervised manner on the median filtered label fields created with an initial k-Means clustering. Alternatives would be, for instance, iterated conditional estimation [BP93] or the mean field theory [Zha92].

For the prior parameters α_i and β we employ least squares estimation, which was first proposed by Derin et al. [DE87]. The prior part of the potential function for a single site s (12) can be rewritten as

$$U(x_s, x_{N_s}, \theta_p) = \theta_p^T N(x_s, x_{N_s}) \quad (14)$$

where θ_p is the desired prior parameter vector containing the α_i and β , and $N(x_s, x_{N_s})$ is a vector valued function defined as:

$$N(x_s, x_{N_s}) = \begin{bmatrix} \delta_{x_s, 1}, \\ \delta_{x_s, 2}, \\ \dots \\ \delta_{x_s, C}, \\ \sum_{c \in \mathcal{C}_3: s \in c} \gamma(c) \end{bmatrix} \quad (15)$$

From (14) and the basic definition of conditional probabilities on MRFs:

$$P(x_s | x_{N_s}) = \frac{e^{-U(x_s, x_{N_s}, \theta_p)}}{\sum_{x_s \in \mathcal{L}} e^{-U(x_s, x_{N_s}, \theta_p)}} \quad (16)$$

the following relationship can be derived [DE87]:

$$\theta_p^T [N(x'_s, x_{N_s}) - N(x_s, x_{N_s})] = \ln \left(\frac{P(x_s, x_{N_s})}{P(x'_s, x_{N_s})} \right) \quad (17)$$

where x'_s is a label different of x_s having the same neighborhood labels x_{N_s} . The RHS of (17) can be estimated using histogram techniques, counting the number of occurrences of the clique labellings in the label field. Considering all possible combinations of x_s , x'_s and x_{N_s} , (17) represents an over determined system of linear equations which can be rewritten in matrix form as follows:

$$N \theta_p = p \quad (18)$$

where N is a $M \times C + 1$ matrix, the rows of which contain the vectors $[N(x_s, x_{N_s}) - N(x'_s, x_{N_s})]^T$ for different labels x_s and x'_s having the same neighborhood labeling x_{N_s} , and M is the number of data points. The elements of the vector p are the corresponding values from the RHS of (17). System (18) can be solved using standard least squares techniques, as for instance the Moore-Penrose pseudo inverse.

For practical purposes, note that labeling pairs with one or both of the probabilities $P(x_s, x_{N_s})$ and $P(x'_s, x_{N_s})$ equal to zero cannot be used. Furthermore, Derin et al. suggest to discard equations with low labeling counts in order to make the estimation more robust.

For reasons of numerical stability, we set one of the parameters α_i to 1. Although the α_i can be interpreted as logarithms of prior probabilities, it is not necessary that the sum of their exponentials be 1, since the eventual missing or excess probability mass will be absorbed into the partition factor Z in (2).

As an example, the parameters of the Markov model for the clustering presented on Figure 5 are: $\alpha_1 = 1$, $\alpha_2 = 0.979153$ and $\beta = 2.673553$.

The parameters of the observation model are estimated using the classical maximum likelihood estimators: the mean values and the variance values of each class. They are empirically estimated from the result of the median filtered label fields created by the initial k-means clustering.

7. Complete algorithm and results

Algorithm 1 sums up the MRF clustering algorithm for 3D meshes. The start temperature $T^{(1)}$ and speed K have empirically been fixed to, respectively, 4 and 0.97; these values give good results in our experiments. Moreover, small changes of these parameters do not influence the result of the optimization [DHS00]. For the simulated annealing, we have chosen $i_{max} = 50$ iterations; this number must be high enough to ensure convergence of the sampling algorithm. Obviously, it depends on the specific form of the model, in particular the length of the dependency chains in the dependency graph etc. To our knowledge, no work exists which is able to learn this number from training data, therefore we determined the necessary number of iterations empirically.

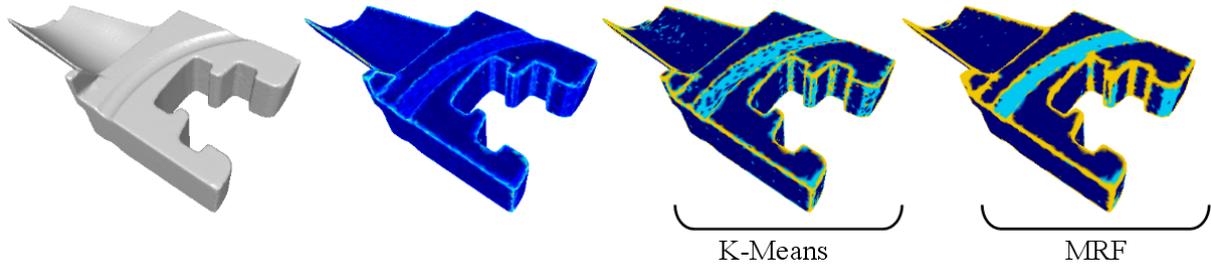


Figure 4: From left to right: *The Blade model (40K vertices), curvature scalar field (geodesic radius = 0.5%), clustering using K-Means (3 clusters), clustering using our MRF algorithm (3 clusters).*

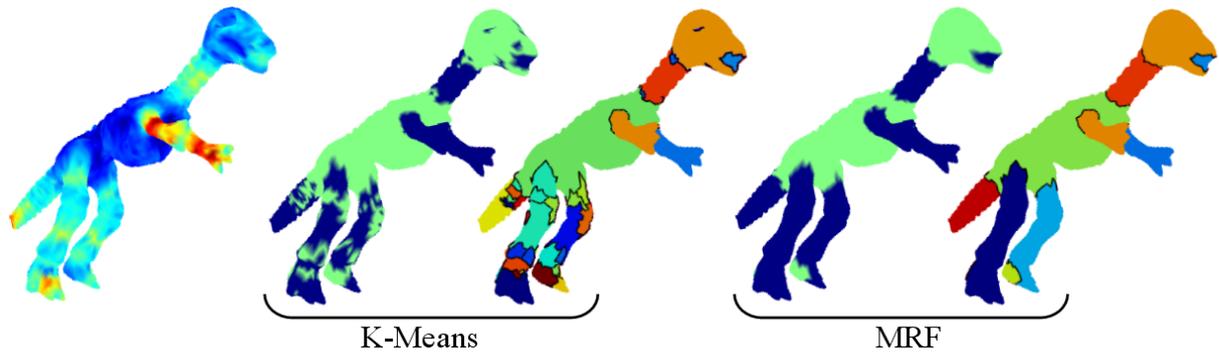


Figure 5: From left to right: *The curvature scalar field (geodesic radius = 6%) of the the Dyno-5 mesh (5K vertices), 2-clustering using K-Means and region growing result, 2-clustering using MRF and region growing result.*

Algorithm 1: Our whole algorithm for 3D mesh clustering

Input: C (number of label), $T^{(1)}$ (start temperature), K (cooling speed), i_{max} (number of iterations)

Output: The estimated label field \hat{x}

- Initialization of the labels x with k-Means clustering of the attribute values y .
 - Median filtering of the labels x
 - Parameter estimation for observation and prior model from x (see section 6).
 - \hat{x} is estimated optimizing (5): simulated annealing with i_{max} iterations, using $T^{(1)}$, K and the parameters (see section 5).
-

In order to demonstrate the efficiency of our algorithm for mesh clustering, we have conducted experiments with different meshes from 5K to 40K vertices and for different numbers of clusters (from 2 to 5). We have particularly focused on the curvature attribute: a scalar value associated with each vertex, but our algorithm works for any other value or combination of values (like roughness in Fig. 6). Table 1 details the processing times for the different ob-

jects which are presented in the figures. For a simple model (<10K vertices) the MRF optimization takes less than 10 seconds. For more complex models (~40K vertices) the processing time is around 30 seconds. We have chosen 50 iterations for the simulated annealing optimization since this value seems enough to reach the convergence in our examples. Figure 3 illustrates the clustering of the maximum cur-

Table 1: Processing time of our MRF clustering algorithm.

3D Model	Cluster nb	Proc. time (s)
Dyno-5 (5K vertices)	2	3.2
Dyno-10 (10K vertices)	5	7.1
Lion Head (39K vertices)	2	25.2
Blade (40K vertices)	3	28.9

vature (geodesic radius = 8%) of the Dyno-10 shape (10K vertices) into 5 clusters. The labeling is very clean without any noise and each region exhibits both attribute and spatial coherency.

We have also compared our MRF clustering with the K-Means algorithm. Figure 4 illustrates the clustering of the Blade model (40K vertices) according to its curvature, into 3 clusters. The noise introduced by the K-Means classification has been almost entirely removed by using our MRF algorithm. Hence a further segmentation using this spatially

coherent clustering will be much easier.

We have conducted experiments in a segmentation scenario: the curvature values (geodesic radius = 6%) of the Dyno-5 model (5K vertices) have been clustered into 2 clusters using both algorithms (see figure 5). Then we have conducted a spatial segmentation (each facet is affected to a region number) using the region growing from [LDB05]. The result is very clean with our clustering, with about 10 regions corresponding with the significant parts of the Dyno (arms, legs, head etc.). On the contrary, a bad over-segmentation is obtained when considering the simple K-Means classification because of the noise.

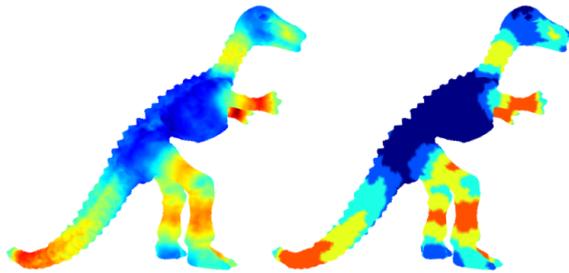


Figure 3: From left to right: *The Dyno-10 mesh (10K vertices), curvature scalar field (geodesic radius = 8%), clustering using our MRF algorithm (5 clusters).*

Lastly, figure 6 illustrates the clustering of the Lion Head model (39K vertices) according to its roughness map (calculated using [Lav07]) into 2 clusters. The roughness represents the amount of local geometric noise on the surface. Indeed, a textured (or *rough*) region is able to hide geometric distortions much better than a smooth one. Hence this measure can be advantageously integrated to watermarking algorithms in order to concentrate the geometric modifications on rough parts of the object. An example of its usage is the classification of the object into 2 clusters: rather rough and rather smooth, so as to watermark the regions accordingly. In order to be properly used in such scenario, the clustering has to be robust to some geometric attacks and rather coherent with the geometry (to keep the watermark invisible). That is the case with the MRF clustering which — due to the global optimization — provides a clean decomposition, while being robust to slight geometric attacks.

8. Conclusion and Futur Work

This paper presents a new Bayesian framework for 3D mesh clustering, based on Markov Random Fields. The approach allows to integrate, in a global optimization process, both attribute values and spatial constraints in the labeling, by using appropriate prior models and observation models. Results demonstrate the efficiency of this framework which thus can be quite useful for 3D mesh analysis or segmentation. In computer vision, hierarchical Markov models have been



Figure 6: From left to right: *The Lion Head model (39K vertices), roughness scalar field, clustering using K-Means (2 clusters), clustering using our MRF algorithm (2 clusters).*

introduced to fasten the optimization of the label field. Bouman and Shapiro were among the first to propose such causal hierarchical models [BS94]. A quad tree models the spatial interactions between the leaf pixel sites through their interactions with neighbors in scale. We plan to introduce this hierarchical Markov modeling for 3D mesh labeling. However contrary to a 2D image, a 3D mesh owns an arbitrary topology and irregular sampling which makes this hierarchical decomposition quite difficult. A solution could consist in using a simplification algorithm [GH97] or a geometry-based decomposition like KD-Trees [GD02].

Acknowledgment

This work is partially supported by the French National Research Agency (ANR) through MADRAS project (ANR-07-MDCO-015).

References

- [AKM*06] ATTENE M., KATZ S., MORTARA M., PATANE G., SPAGNUOLO M., TAL A.: Mesh segmentation: A comparative study. In *Shape Modeling International* (2006).
- [And07] ANDERSEN V.: *Smoothing 3D Meshes using Markov Random Fields*. Master’s thesis, IT University of Copenhagen, 2007.
- [Bes74] BESAG J.: Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B* 36, 2 (1974), 192–236.
- [BP93] BRAATHEN B., PIECZYNSKI W.: Global and Local Methods of Unsupervised Bayesian Segmentation of Images. *Machine Graphics and Vision* 2, 1 (1993), 39–52.
- [BS94] BOUMAN C., SHAPIRO M.: A Multiscale Random Field Model for Bayesian Image Segmentation. *IEEE Transactions on Image Processing* 3, 2 (1994), 162–177.
- [CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. In *ACM Siggraph* (2004), pp. 905–914.

- [CSM03] COHEN-STEINER D., MORVAN J.: Restricted delaunay triangulations and normal cycle. In *19th Annu. ACM Sympos. Comput. Geom.* (2003).
- [DE87] DERIN H., ELLIOTT H.: Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9, 1 (1987), 39–55.
- [DHS00] DUDA R., HART P., STORK D.: *Pattern Classification, 2nd Edition*. Wiley, New York, NY, Nov. 2000.
- [GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.* 25, 1 (2006), 130–150.
- [GD02] GANDOIN P.-M., DEVILLERS O.: Progressive lossless compression of arbitrary simplicial complexes. In *ACM Siggraph* (2002), pp. 372–379.
- [GG84] GEMAN S., GEMAN D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 6 (11 1984), 721–741.
- [GH97] GARLAND M., HECKBERT P.-S.: Surface simplification using quadric error metrics. *ACM Siggraph* (1997), 209–216.
- [HC68] HAMMERSLEY J., CLIFFORD P.: Markov fields on finite graphs and lattices. unpublished manuscript, 1968.
- [JM07] JAGANNATHAN A., MILLER E.: Three-dimensional surface mesh segmentation using curvedness-based region growing approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 12 (2007), 2195–2204.
- [KLT05] KATZ S., LEIFMAN G., TAL A.: Mesh segmentation using feature point and core extraction. *The Visual Computer* 21, 8-10 (2005), 649–658.
- [KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. In *ACM Siggraph* (2003), pp. 954–961.
- [Lav07] LAVOUÉ G.: A roughness measure for 3d mesh visual masking. In *ACM Symposium on Applied Perception in Graphics and Visualization* (2007), pp. 57–60.
- [LDB05] LAVOUÉ G., DUPONT F., BASKURT A.: A new cad mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design* 37, 10 (2005), 975–987.
- [LDB07] LAVOUÉ G., DUPONT F., BASKURT A.: A framework for quad/triangle subdivision surface fitting: Application to mechanical objects. *Computer Graphics Forum* 26, 1 (2007), 1–14.
- [Li01] LI S.: *Markov Random Field Modeling in Image Analysis*. Springer Verlag, 2001.
- [LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.* 21, 3 (2002), 362–371.
- [LVJ05] LEE C., VARSHNEY A., JACOBS D.: Mesh saliency. In *ACM Siggraph* (2005), pp. 659–666.
- [LZ04] LIU R., ZHANG H.: Segmentation of 3d meshes through spectral clustering. In *Pacific Conference on Computer Graphics and Applications* (2004), pp. 298–305.
- [LZH*07] LAI Y.-K., ZHOU Q.-Y., HU S.-M., WALLNER J., POTTMANN H.: Robust feature classification and editing. *IEEE Transactions on Visualization and Computer Graphics* 13, 1 (2007), 34–45.
- [MDKK06] MIZOGUCHI T., DATE H., KANAI S., KISHINAMI T.: Segmentation of scanned mesh into analytic surfaces based on robust curvature estimation and region growing. In *Geometric Modeling and Processing* (2006), pp. 644–654.
- [SC06] SALZENSTEIN F., COLLET C.: Fuzzy markov random fields versus chains for multispectral image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 11 (2006), 1753–1767.
- [SSCO06] SHAMIR A., SHAPIRA L., COHEN-OR D.: Mesh analysis using geodesic mean-shift. *Vis. Comput.* 22, 2 (2006), 99–108.
- [SSGH01] SANDER P. V., SNYDER J., GORTLER S. J., HOPPE H.: Texture mapping progressive meshes. In *ACM Siggraph* (2001), pp. 409–416.
- [TVD06] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: 3d Mesh Skeleton Extraction Using Topological and Geometrical Analyses. In *Pacific Conference on Computer Graphics and Applications* (2006), pp. 85–94.
- [TVD07] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Reeb chart unfolding based 3D shape signatures. In *Eurographics* (2007). short paper.
- [WC07] WU J., CHUNG A.: A segmentation model using compound markov random fields based on a boundary model. *IEEE Transactions on Image Processing* 16, 1 (2007), 241–252.
- [WD02] WOLF C., DOERMANN D.: Binarization of low quality text using a markov random field model. In *International Conference on Pattern Recognition* (2002), vol. 3, pp. 160–163.
- [WSC04] WILLIS A., SPEICHER J., COOPER D. B.: Surface sculpting with stochastic deformable 3d surfaces. In *International Conference on Pattern Recognition* (2004), pp. 249–252.
- [Zha92] ZHANG J.: The mean field theory in em procedures for markov random fields. *IEEE transactions on image processing* 40, 10 (1992), 2570–2583.