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).

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 applications 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. Another 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 [LZH07], 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. Clas-
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 $G = \{G, E\}$, where $G$ is the set of vertices (sites) of the mesh and $E$ is the set of edges of the mesh. Our objectiv 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 $\Lambda$. $C = |\Lambda|$ 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 variables. 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

$$P(X = x) > 0 \quad \forall x \in \Omega$$

and

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

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 $X_r$ 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

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

where $Z = \sum e^{-U(x)/T}$ is a normalization constant, $T$ is a temperature factor which can be assumed to be 1 for simplicity, $U(x) = \sum_{c \in C} V_c(x)$ is a user defined energy function defined on local interactions in the cliques, $C$ 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:

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

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

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-
where the defined as: logistic model \[Li01\] whose energy potential functions are
cuous regions. For this purpose we modified the multi-level construction of these models.
The next section details 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 homogenous regions. For this purpose we modified the multi-level logistic model \[Li01\] whose energy potential functions are defined as:

\[
U(x) = \sum_{x \in G} \sum_{c \in C} \alpha_i \delta_{i,x} + \sum_{c \in C_3} \beta \gamma(c)
\]

where the \( \alpha_i \) \((i = 1 \ldots C)\) and \( \beta \) are parameters, \( C_3 \) 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

\[
\delta_{i,j} = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{else} 
\end{cases}
\]

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

\[
\gamma(c) = - \sum_{(i,x) \in C \times x, \neq x} \delta_{i,x}
\]

Each parameter \( \alpha_i \) controls the prior probability of a given label \( i \) whereas the parameter \( \beta \) 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:

\[
p(y|x) = \prod_{v \in G} p(y_v|x_v) = \prod_{v \in G} \mathcal{N}(y_v; \mu_v, \Sigma_v)
\]

where \( \mu_v \) and \( \Sigma_v \) are, respectively, the mean vector and the covariance matrix of class \( x_v \). 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 \( X_i \) and \( Y_i \) with the following potential functions:

\[
U(x_i, y_i) \propto \ln[p(y_i|x_i)]
\]

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

\[
q = e^{-\Delta/T}
\]
where $\Delta$ is the difference of the posterior energy potentials $U(x_s, x_{N_s})$ of site $s$ before and after the change of $x_s$:

$$U(x_s, x_{N_s}, y_s) = U(x_s, x_{N_s}) + U(x_s, y_s)$$

$$= \sum_{l \in L} \alpha_l \delta_{x_s} + \sum_{e \in C_l \times e} \beta_e \gamma(e)$$

$$+ (y_s - \mu_{0_s})^T \sum_{x_s} (y_s - \mu_{0_s}) \quad (12)$$

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 median filtered label fields created with an initial k-Means clustering presented on Figure 5 are:

$$\text{Algorithm 1 sums up the MRF clustering algorithm for 3D meshes. The start temperature } T^{(1)} \text{ and speed } K \text{ are high enough to ensure convergence of the sampling algorithm. Obviously, it depends on the specific form of the model, in particular the length of the dependency chains in the dependency graph. To our knowledge, no work exists which is able to learn this number from training data, therefore we determined the necessary number of iterations empirically.}$$
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), imax (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 \( i_{\text{max}} \) 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 objects 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 curvature (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

Table 1: Processing time of our MRF clustering algorithm.

<table>
<thead>
<tr>
<th>3D Model</th>
<th>Cluster nb</th>
<th>Proc. time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dyno-5 (5K vertices)</td>
<td>2</td>
<td>3.2</td>
</tr>
<tr>
<td>Dyno-10 (10K vertices)</td>
<td>5</td>
<td>7.1</td>
</tr>
<tr>
<td>Lion Head (39K vertices)</td>
<td>2</td>
<td>25.2</td>
</tr>
<tr>
<td>Blade (40K vertices)</td>
<td>3</td>
<td>28.9</td>
</tr>
</tbody>
</table>
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.

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 Future 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 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


© The Eurographics Association 2008.


