OXFORD CDE

Journal of Computational Design and Engineering, 2023, **10**, 1856–1867

DOI: 10.1093/jcde/qwad082

Advance access publication date: 11 August 2023

Research Article

NURBS-based surface generation from 3D images: spectral construction and data-driven model selection

Antoine Perney [1,2,*], Stéphane Bordas[2] and Pierre Kerfriden[1,3,*]

[1]Centre des Matériaux, Mines Paris - PSL University, Evry 91100, France
[2]Institute for Computational Engineering, Faculty of Science, Technology and Communication, University of Luxembourg, Esch-sur-Alzette, L-4364, Luxembourg
[3]Cardiff School of Engineering, Cardiff University, Cardiff, CF10 3AT, UK
*Correspondence: antoine.perney@mines-paris.org (AP); pierre.kerfriden@mines-paristech.fr (PK)

Abstract

In this paper, we present a set of improved algorithms for recovering computer aided design (CAD-type) surface models from three-dimensional (3D) images. The goal of the proposed framework is to generate B-spline or non-uniform rational B-spline (NURBS) surfaces, which are standard mathematical representations of solid objects in digital engineering. To create a NURBS surface, we first compute a control network (a quadrilateral mesh) from a triangular mesh using the Marching Cubes algorithm and Discrete Morse theory. To create a NURBS surface, we first compute a triangular mesh using the Marching Cubes algorithm, then the control network (a quadrilateral mesh) is determined from the triangular mesh by using Discrete Morse theory. Discrete Morse theory uses the critical points of a specific scalar field defined over the triangulation to generate a quad mesh. Such a scalar field is obtained by solving a graph Laplacian eigenproblem over the triangulation. However, the resulting surface is not optimal. We therefore introduce an optimization algorithm to better approximate the geometry of the object. In addition, we propose a statistical method for selecting the most appropriate eigenfunction of the graph Laplacian to generate a control network that is neither too coarse nor too fine, given the precision of the 3D image. To do this, we set up a regression model and use an information criterion to choose the best surface. Finally, we extend our approach by taking into account both model and data uncertainty using probabilistic regression and sampling the posterior distribution with Hamiltonian Markov Chain Monte Carlo.

Keywords: parametric curve and surface models, NURBS surface, sampling

1. Introduction

NURBS surfaces are widely used in CAD software due to their continuity and the ease with which one can interact and adjust them. In practice, people tend to use CAD softwares; however, they do not have efficient capabilities to process triangulation. In addition, NURBS surfaces are also used for numerical simulation using isometric analysis (Hughes *et al.*, 2005; Nguyen *et al.*, 2015). Methods for drawing a three-dimensional (3D) object with NURBS surfaces in CAD software are relatively well established; however, sometimes it is necessary to obtain a NURBS surface representation of a real object such as an organ or a bone, for example. Hence, we expect to reconstruct a NURBS surface from images such as computed tomography (CT) or magnetic resonance imaging (MRI) scans. A naive approach determining such a surface would be to start from an arbitrary surface. Then, an optimization process is employed to gradually minimize the distance between the surface and the data points. However, such a method assumes a priori knowledge of the topology of the object, i.e., whether it resembles a sphere, a torus, a double torus or not, to initialize the process with the correct topology. For example, in Anderson and Crawford-Hines (2000), some organs can be reconstructed but must be homeomorphic to a sphere. Indeed, the method uses a cylinder and then solves a mean squared error problem to fit the surface to the point cloud. In Boujraf *et al.* (2012), they also reconstruct objects with sphere-equivalent topology.

By studying the NURBS surface definition, we observe that the requirement of a control net leads to the requirement of building a quadrangular mesh. Thus, an alternative approach would be to first determine a quadrangulation and then use it to draw NURBS surface. In this paper, we will use this method. To calculate a NURBS surface, a semi-regular quadrangular mesh is required (see Bommes *et al.*, 2013 for the definitions of mesh types). In order to establish a NURBS surface, it is necessary to employ a regular quad mesh, as the NURBS surface is defined by a matrix of control points. However, when generating a CAD surface for a complex object, multiple NURBS surfaces are often required. Therefore, it becomes necessary to compute a coarse (irregular) quad mesh. Subsequently, each quad within this mesh can be subdivided and utilized as the control net for a NURBS surface. Additionally, by ensuring that the new vertices introduced during the division process at the boundary of each quad are identical, we ensure that the distinct NURBS surfaces share the same control point at the boundary. This ensures that the resulting surface is continuous $\mathcal{C}^0$. Methods that produce irregular quad meshes, such as the Dual Marching Cubes, may not be the best option. This is due to the large number of quads they generate. Subdividing these irregular quad meshes to fit a NURBS surface on each introduces a significant number of patches and subsequently a large number of parameters. This can make it more difficult to manipulate them within CAD software. Various techniques have been devel-

Received: January 17, 2023. **Revised:** July 2, 2023. **Accepted:** July 29, 2023

oped to address this issue by utilizing a triangular mesh as input (see e.g., Bommes *et al.*, 2009; Dong *et al.*, 2006; Eck & Hoppe, 1996; Fang *et al.*, 2018; Hormann & Greiner, 2000; Kälberer *et al.*, 2007; Owen *et al.*, 1999; Ray *et al.*, 2006; Tarini *et al.*, 2011). One notable method is the approach proposed by Dong *et al.* (2006); this method demonstrates robustness, as it can handle different types of topologies, and offers parameter adjustments to control the density of quads and subsequently the number of control points in the resulting NURBS surface. Moreover, Tierny *et al.* (2018) provide an open source implementation of this method.

This structure allows us to calculate a NURBS surface on each of the patches. The calculation of such a mesh is based on the Discrete Morse theory, (see Forman, 2002 for a complete introduction). The main idea of this theory is to obtain information about the topology of a manifold by considering a well-chosen function defined on this variety. More precisely, in our case, a scalar field is computed on the triangulation by solving a graph Laplacian eigenproblem. This gives us the 'well-chosen' function and then by calculating the critical points and integral lines, a topological data structure called the Morse-Smale complex is determined, as described in Tierny (2017). This structure reveals a representation in the form of patches. Thus, by subdividing each patch, a quadrangulation is obtained. Then, on each of these patches a NURBS surface can be calculated and finally, by juxtaposing all the NURBS surface a complete representation of the object is obtained.

To generate the Morse-Smale complex and the quadrangular mesh, we use the Topology ToolKit (TTK) library (Tierny *et al.*, 2018). When calculating the quadrangular mesh, the quadrilaterals are adjusted to fit the triangulation; however, the patched-NURBS surface is not interpolated into its control net. We therefore introduced an optimization step using a quasi-Newton method to reduce the distance between the patched-NURBS surface and the triangulation (see Byrd *et al.*, 1995; Nocedal & Wright, 2006). Moreover, the scalar field being determined in an eigenvalue problem, we have at our disposal different scalar fields and thus different quadrangular meshes. Thus, we can ask ourselves how to choose the scalar field. As we want to use the NURBS surface representation in CAD software, the number of control points has to be as small as possible while keeping a surface that accurately represents the data.

Since the triangulation may be inaccurate or too dense compared to the real data, we will compare the patched-NURBS surfaces directly with the images. To do this, we suggest building a regression model generating new images from a given patched-NURBS surface. This regression model allows us to consider the *real* noise into account, i.e., the noise in the data. Then, using a maximum likelihood technique and an information criterion (Akaike, 1998), a model with a minimal number of control points and representing accurately the data is chosen in the set of all possible quad meshes generated with Dong *et al.* (2006), i.e., the set of models generated from a given Laplacian eigenproblem. Once the statistical model generation is established we are going one step further, by not only taking into account the noise in the data but also by encoding our lack of knowledge in the patched-NURBS surface itself via a prior probability density distribution. Hence, we will seek to obtain a surface probability distribution. To do this, the control points will be considered as random variables, as some parameters of the regression model. We therefore will estimate a probability distribution of these regression parameters. More precisely, we seek to determine $\mathcal{P}(\theta|Y)$ where Y is the data and θ the parameters. Using the Bayes theorem, this is equivalent to sampling $\mathcal{P}(Y|\theta)\mathcal{P}(\theta)$. Thus, with a sampling method, here Hamiltonian Markov Chain Monte Carlo (HMCMC), we will obtain the probability distribution of the control points which is effectively a probability distribution of patched-NURBS surfaces.

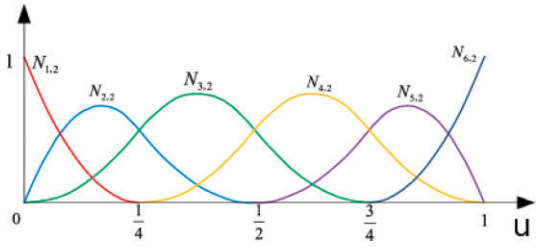

Figure 1: B-spline basis functions.

This paper is organized as follows. In the first section of this paper, we will provide an introduction on using Discrete Morse theory to construct patched-NURBS surfaces. In the second part, we will carry out a model selection based on images. And finally, we will adopt a Bayesian point of view in order to obtain a probability distribution of surfaces.

2. NURBS Surface Generation

As parametric surfaces, NURBS surfaces can be expressed in a 3D space as

$$S(u, v) = (x(u, v), y(u, v), z(u, v))$$

for u and v in a parametric space, generally $[0, 1]^2$.

Moreover, they are constructed over surface basis functions defined by a tensor product of two curve basis functions.

In this section, we give a brief overview of the traditional method for constructing NURBS surfaces. We start by giving the definition of B-spline basis functions and then construct the tensor product basis in order to define NURBS surfaces.

By looking at the definition of NURBS surfaces, we will figure out that control points define a quad mesh. This will lead us to explore Morse theory to construct a quad mesh and then to use this mesh to compute NURBS surfaces.

2.1. A short introduction to NURBS surfaces

We define B-spline basis functions as follows (Piegl & Tiller, 1996):

Definition 1 (Carl De Boor formula) Let $m + 1$ nodes $(t_i)_{i=0}^m$ in $[0, 1]$ such that $0 \leq t_0 \leq t_1 \leq \ldots \leq t_m \leq 1$. The B-spline basis functions of degree n are defined by the recursive formula:

$$N_{i,0}(u) := \begin{cases} 1 & \text{if } \; t_i \leq u \; < \; t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

and for $n \geq 1$

$$N_{i,n}(u) := \frac{u - t_i}{t_{i+n} - t_i} N_{i,n-1}(u) + \frac{t_{i+n+1} - u}{t_{i+n+1} - t_{i+1}} N_{i+1,n-1}(u).$$

The $(t_i)_i$ sequence will be called the knot vector in the NURBS surface definition.

The Fig. 1 shows an example of B-spline basis functions of degree 2.

From an implementation point of view, the recursive aspect of this formula is quite convenient.

Moreover, the $(N_i^p)_i$ basis is used to express NURBS curve: $C(u) = \sum_{i=0}^n w_i P_i N_i^p(u)$.

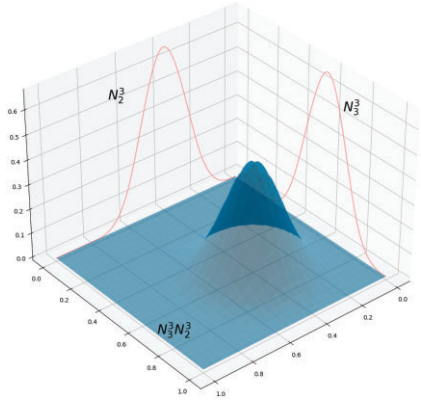

Figure 2: Tensor product surface.

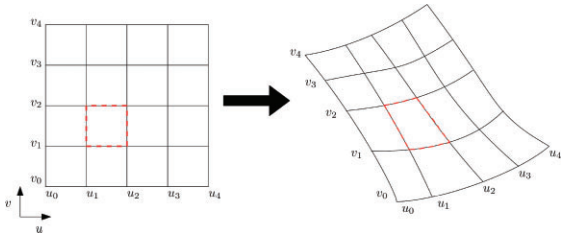

Figure 3: Parametric space.

To construct the surface basis functions, we will use a tensor product of two B-spline basis functions, as in Fig. 2:

Then we can define NURBS surface with other definition.

Definition 2 Let p, q, r, s, n, and m integers such that $r = n + p + 1$ and $s = m + q + 1$.

A NURBS surface of degree p in the u direction and degree q in the v direction is a bivariate vector-valued piecewise rational function, Fig. 3:

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} P_{i,j} R_{i,j}^{p,q}(u,v)$$

(i) $P_{i,j} \in \mathbb{R}^3$ are the control points.
(ii) $R_{i,j}^{p,q}$ are the tensor product NURBS basis functions defined on the knot vectors:

$$U = \{\underbrace{0,\ldots,0}_{p+1}, u_{p+1},\ldots,u_{r-p-1}, \underbrace{1,\ldots,1}_{p+1}\}$$

$$V = \{\underbrace{0,\ldots,0}_{q+1}, u_{q+1},\ldots,u_{s-p-1}, \underbrace{1,\ldots,1}_{q+1}\}$$

$$R_{i,j}^{p,q}(u,v) = \frac{N_i^p(u) N_j^q(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} N_i^p(u) N_j^q(v)} \text{ with } w_{i,j} \in \mathbb{R}.$$

As we can see in Fig. 4, a quad mesh is required (the dashed line representing the control net forms a quad mesh). To have an algorithm that works on arbitrary shapes, we are using the method described in Dong *et al.* (2006) and the algorithm from TTK (Tierny *et al.*, 2018). This method computes a quad mesh over a triangulation by using a topological data structure named the Morse-Smale complex related to Morse theory.

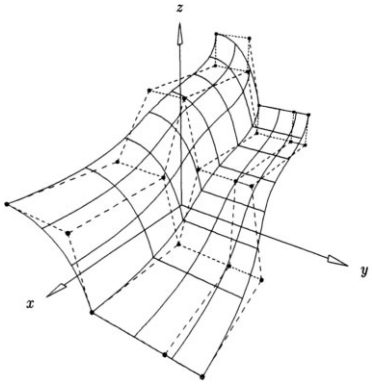

Figure 4: NURBS surface with control net (Piegl & Tiller, 1996).

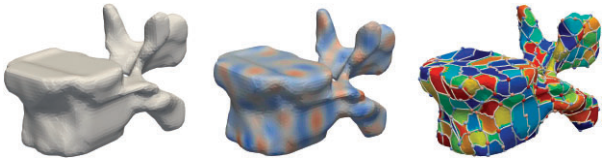

Figure 5: Triangulation (left-hand panel), scalar field (middle panel), and Morse-Smale complex (right-hand panel).

2.2. Morse theory and Morse-Smale complex

The definition of NURBS surface shows that the control net is defined by a quad mesh. To generate such a quad mesh, we will use a topological data structure, the Morse-Smale complex which gives a mesh topologically equivalent to a quad mesh. The Morse-Smale complex is derived from Discrete Morse theory, introduced by Forman (2002), which involves a function f defined from a triangulation T to $\mathbb{R}$ that encodes enough information about T to analyse its topology. More precisely, Discrete Morse theory studies the relationships between the topology of a shape represented by the triangulation and the critical points of a Discrete Morse function defined on it. To generate the Morse-Smale complex in practice, we will use the method of Dong *et al.* (2006) implemented in the TTK (Tierny *et al.*, 2018). However, the generated Morse-Smale complex provides large quads that correspond to an inaccurate quadrangulation, and therefore, each patch is subdivided to form a finer quad mesh. The different steps are represented in Fig. 5.

To obtain a Morse-Smale complex with the most evenly spaced regions over the surface, we can use an eigenfunction of a graph Laplacian over the input mesh. The critical points of such an eigenfunction are indeed well spaced over the mesh. Then, we solve the Laplacian eigenproblem with cotangent weight as suggested in Dong *et al.* (2006):

$$(\Delta f)_i = \sum_{j \in \mathcal{N}(i)} \left(\cot(\alpha_{i,j}) + \cot(\beta_{i,j})\right)\left(f_j - f_i\right)$$

where $\mathcal{N}(i)$ is the set of neighbours of the vertex i.

However, the eigenfunctions obtained by solving this problem are not Discrete Morse functions, but we can approximate any scalar field with a Discrete Morse function (see Shivashankar *et al.*, 2012 for details). Moreover, the computation of the discrete gradient, hence of the V-path is done with the method described in Gyulassy *et al.* (2008).

Let us remark that the Laplacian problem gives different scalar fields and hence different Morse-Smale complexes (see Fig. 6). If we select a Discrete Morse function, the ascending and descend-

Figure 6: Different Morse-Smale complexes.

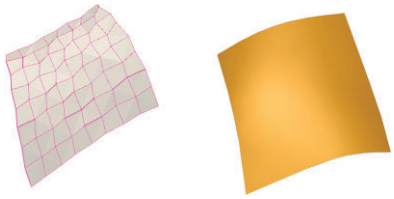

Figure 7: Quad patch (left-hand panel) and NURBS patch (right-hand panel).

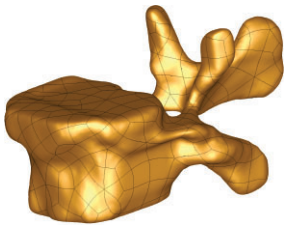

Figure 8: Patched-NURBS surface. The partitions on the patched-NURBS surface correspond to the Mores-Smale complex cell, i.e., NURBS surface patches.

ing manifolds will intersect in a transversal manner, resulting in the Morse-Smale complex. In Section 4, we are going to present a method to choose one of them.

In the next section, we will explain how we can compute a quad mesh from the Morse-Smale complex.

2.3. Quadrangulation and NURBS surface

The patch structure of the Morse-Smale complex is used to generate a quadrangulation. To do so each critical point of the Morse-Smale complex is considered as a vertex of the quadrangulation. Therefore, they are linked by approximating the V-path by a line. This gives a coarse-quad mesh. To improve the quadrangulation, a subdivision step is required. To do so, the authors in the code of Tierny *et al.* (2018) suggest three steps which are subdivision, relaxation, and projection.

We can extract each quad patch and compute a NURBS surface on each one (Fig. 7).

Then by juxtaposing all patches together we obtain a NURBS surface representation of the object (Fig. 8).

In the rest of the paper, the following definitions will be employed:

(i) The term 'NURBS surface' will denote a NURBS surface as defined in Definition 2.
(ii) 'NURBS patch' will denote a NURBS surface that shares boundary control points with another NURBS surface.
(iii) 'Patched-NURBS surface' will refer to the 3D surface representation of an object composed of multiple NURBS patches.

In our experiments, we have considered a uniform knot vector and control point weights equal to 1, therefore, the NURBS surfaces are in fact B-spline.

3. Fitting Optimization According to the Triangle Mesh

3.1. Problem settings

The methodology described in the previous section is limited as the quads are fitted to the triangulation in the **projection step** proposed in TTK code. Therefore, the NURBS surface will not fit the triangles as they pass under or over the quad mesh. We suggest minimizing the sum of squared distances between the surface S and the set of vertices $\mathcal{V}$ according to the control points.

Let $P^k = (P^k_{i,j})_{i,j\in[\![0,n]\!][\![0,m]\!]}$ be the control net of the NURBS patch number k. Each P^k can be rewritten as a 1D vector $(P_l)_{l\in[\![0,nm]\!]}$. Then, let us consider the vector $P = (P^k)_k \in [\![0, N]\!]$, N being the number of patches, ensuring that each element of P is distinct. A table with the corresponding position of each control point in the different patches is created at the same time. P is therefore the set of control points of the patched-NURBS surface S. With each element of P being distinct, we ensure that the optimized surface will be $\mathcal{C}^0$. The dependency of P is written $S(P)$, i.e., the surface S is seen as a function of the control points P. Then we are considering $\mathcal{V}$ the set of triangle vertices, and we are going to adjust the surface according to $\mathcal{V}$. We are trying to minimize the sum of squared distances between the surface S and $\mathcal{V}$, which is equivalent to find $P^\star$ such that

$$P^\star = \arg\min_P \sum_{v\in\mathcal{V}} ||S(P) - v||_2^2.$$

To compute $||S(P) - v||^2$ for a given vertex v, first we determine a set of potential closest NURBS patches by computing the distance from v to each control point $P_{i,j}$ for $i, j \in [\![0, n]\!] \times \in [\![0, m]\!]$ as a vector $(P_k)_{k\in[\![0,nm]\!]}$. Then, for each of these potential closest patches, the distance between these NURBS patches and the vertex v is computed with the algorithm provided by Li *et al.* (2019) and the one with the minimum distance is considered as the closest one.

Let us write $F(P) = \sum_{v\in\mathcal{V}} ||S(P) - v||^2$. To minimize this function, we will use a quasi-Newton method, limited-broyden fletcher goldfarb shanno (L-BFGS), (Liu & Nocedal, 1989), defined by the iteration:

$$P_{n+1} = P_n + \gamma_n$$

where γ_n is the direction of the steepest descent and verify

$$B_n\gamma_n = -\nabla F(P_n)$$

where B_n is an approximation of the Hessian matrix of F at P_n.

3.2. Examples

Here, the example we are considering is the vertebra. A triangulation of the vertebra is shown in Fig. 9.

To demonstrate the effect of the optimization according to the vertices, let us take the scalar field shown in Fig. 10. The generated patched-NURBS surface without optimization is shown in Fig. 11.

Then by using our optimization algorithm, we obtain the patched-NURBS surface in Fig. 12.

The evolution of the distance between the triangle mesh and the patched-NURBS surface is shown in Fig. 13. Figure 14 shows

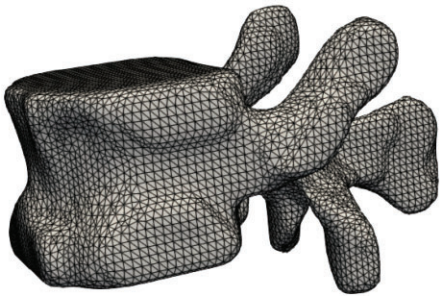

Figure 9: Triangle mesh of a vertebra.

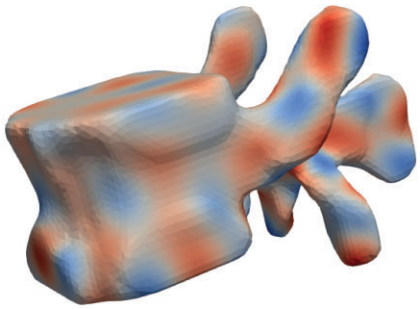

Figure 10: Triangle mesh of a vertebra.

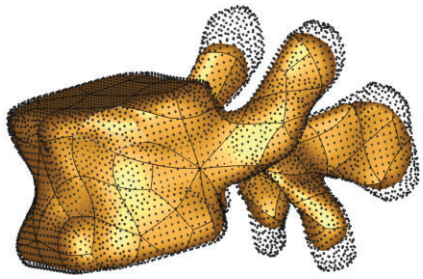

Figure 11: Non-optimized patched-NURBS surface. The non-optimized patched-NURBS surface is in gold and the vertices of the triangulation are represented by the black dots.

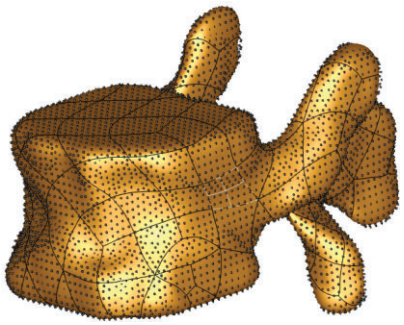

Figure 12: Optimized surface with initial surface shown in Fig. 11.

the evolution of the distance with respect to the iteration with a log-scale.

Now, let us consider a very fine Morse-Smale complex. Figure 15 shows that the L-BFGS does have little effect on the patched-NURBS surface. This is not surprising since the more control

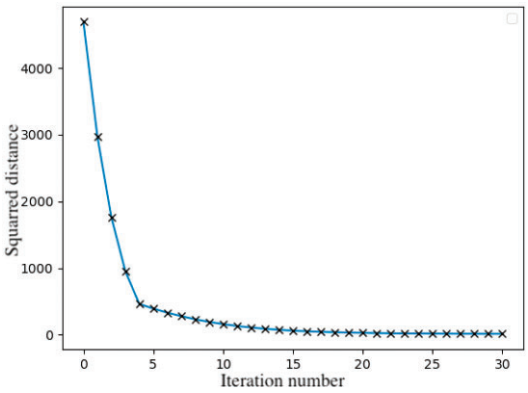

Figure 13: Evolution of the distance with respect to BFGS iteration count.

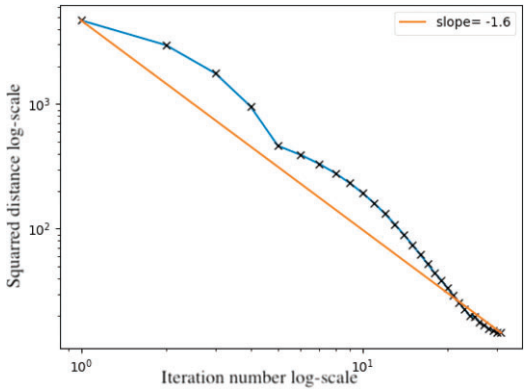

Figure 14: Evolution of the distance with respect to BFGS iteration count (log-scale).

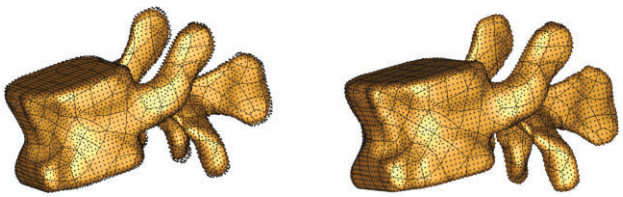

Figure 15: Non-optimized (left-hand panel) and optimized (right-hand panel).

points a NURBS surface has, the more it approximates the quad mesh defined by the control points (Piegl & Tiller, 1996).

We are now able to generate a patched-NURBS surface that corresponds to a specific eigenfunction of the graph Laplacian eigenproblem and optimize the fitting of the patched-NURBS surface according to the triangle mesh. Nevertheless, the selection of the eigenfunction is not based on any specific criteria. Consequently, we will be presenting a model selection algorithm in the next section that will determine the most accurate model for the data, i.e., images, with a minimal number of parameters, i.e., control points. The model selection process is carried out independently of the triangle mesh. The surface triangulation using the Marching Cubes algorithm is performed using the raw 3D image. It produces a deterministic estimate of the 'true' surface. However, when fitting the NURBS surface, one should evaluate the quality of the approximation with respect to the raw data, not with respect to a prior reconstruction that is not equipped with a measure of uncertainty. Seeing from a different perceptive, how could one se-

lect an appropriate threshold for an acceptable level of discrepancy between two reconstructed surfaces? As our statistical generative model is naturally equipped with uncertainty measures, model selection with respect to the raw data may be performed using the Occam's razor: we wish for the simplest surface model that can explain the raw data.

4. Model Selection According to the Images

In this section, we introduce a methodology to determine the surface with the fewest control points that still represents the data (greyscale images) accurately independently of the triangulation.

We will explain how to construct a regression model by assuming that the greyscale data is noisy. Then, by using a maximum-likelihood process we will be able to estimate the regression parameters. Finally, we will show how to perform a statistical model selection using an information criterion.

4.1. Generative model

The basic idea of our model is to assume how far away we are from the patched-NURBS surface utilizing the voxel colour. The voxels are black and as we get close to the surface the voxels become white.

This allows establishing the following regression model, by explaining the greyscale behaviour of a voxel, represented by a random variable Y in $\mathbb{R}$, as a certain function of the voxel position in the space, represented by a random vector X in $\mathbb{R}^3$.

We have at our disposal:

(i) A given patched-NURBS surface S (control points are fixed).
(ii) Observations (x_i, y_i) of the random variables pair (X, Y), where $x_i \in \mathbb{R}^3$ represents the position of the voxel i and $y_i \in \mathbb{R}$ the greyscale value (i.e., the colour of the voxel) of the voxel i.

Then we construct the regression model:

$$Y = \bar{g}_\theta(X) + \varepsilon_\sigma \tag{1}$$

where $\bar{g}_\theta$ is a given function depending on the distance between X and the surface S, and on regression parameters θ. ε_σ is the noise of the regression model, and we have $\varepsilon_\sigma \sim \mathcal{N}(0, \sigma)$. Then we have to estimate the regression parameters $\theta = (\theta_i)_i$ and σ. To do so we are going to use a maximum-likelihood method. This method seeks to find the regression parameters with the highest probability of reproducing the real value from the observed sample. In our case, find the most probable θ and σ such that the model defined by (1) reproduces at best the observed data, i.e., the greyscale value.

4.2. Maximum-likelihood and information criterion

To determine the parameters of the regression above, a maximum likelihood method is used. We wish therefore to maximize the log-likelihood. Let $y = (y_1, \ldots, y_n)$ and $x = (x_1, \ldots, x_n)$ where n is the number of observation and x_i, y_i the ith observation and let $\mathcal{L}_\mathcal{M} = \mathbb{P}(y|x, \theta, \sigma)$ be the likelihood of the model.

Thus, by maximizing $\ln(\mathcal{L}_\mathcal{M})$ or equivalently by minimizing $-\ln(\mathcal{L}_\mathcal{M})$ according to θ, we obtain θ_{ML} and σ_{ML} which are the parameters maximizing $\mathcal{L}_\mathcal{M}$. Therefore, for a given voxel position x we can determine the associate distribution of greyscale value by sampling Y directly from $\mathcal{N}(\bar{g}_{\theta_{ML}}, \sigma^2_{ML})$.

Now, we aim to find the model with the fewest number of parameters, i.e., with the fewest number of control points, but which still represents accurately the data. To do this an information criterion can be used. An information criterion is a measure of the quality of the statistical model. It is based on the fit of the model to the data and the complexity of the model. Here the complexity is the number of control points of the patched-NURBS surface with the parameters of the regression function.

For each model, $\mathcal{M}$, we define the information:

$$IC_\mathcal{M} = \alpha - \ln(\mathcal{L}_\mathcal{M})$$

where α is a penalty term that usually depends on the number of regression parameters, and $\mathcal{L}_\mathcal{M}$ is the maximum-likelihood of the model $\mathcal{M}$.

Then, an information criterion tells us to choose the model with the least information.

4.3. Examples

4.3.1. *MRI: vertebrae*

Since the MRI does not give a uniform greyscale representation for a given object, considering noise becomes problematic, we will use a gradient filter in order to use a simple model for the noise. Usually, the gradient filter will show the contour of the object. Here, the Sobel operator in 3D is used with three filters of size $3 \times 3 \times 3$ (see Sobel, 2014 for the definition).

This operator produces the following results:

As we wish to colour the voxels in function of the distance between their positions and the patched-NURBS surface, we compute all the distances between the patched-NURBS surface and the voxels. To do so we are considering the patched-NURBS surface inside the voxel grid.

Then to compute the distance to each voxel, we are using the vtkDistancePolyDataFilter method in VTK (Schroeder *et al.*, 2006). In order to use the VTK filter, we first tessellate the patched-NURBS surface.

Algorithm 1 Compute signed distance from the patched-NURBS surface to the voxel grid

```
procedure SIGNEDDISTANCEGRID(NURBS, voxelGrid)
    tessellation = tessellate(NURBS)
    distance = computeSignedDistance(tessellation, voxelGrid)
end procedure
```

The details on the signed distance computation can be found in Bærentzen and Aanæs (2005).

As the filter highlights the contour of the objects, we are considering a Gaussian noise around the contours. This allows us to construct the following regression model.

(i) X voxel position
(ii) Y greyscale value

$$Y = \bar{g}_{g_{\max},l}(X) + \varepsilon_\sigma$$

with: $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and $\bar{g}_{g_{\max},l}(x) = g_{\max} e^{-\frac{d(x)^2}{2l^2}}$ where $d(x)$ is the distance of x to the patched-NURBS surface S, where $g_{\max}$ is the greyscale maximum value of the contour, l determines how 'spread' the contour will be, and σ is the image noise.

In our analysis, we are addressing the presence of noise in the image which can be modelled as a random variable with a vari-

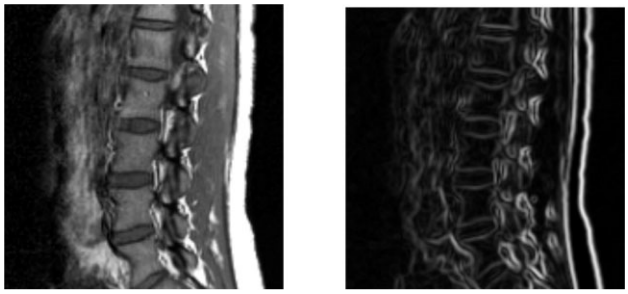

Figure 16: Original (left-hand panel) and filtered (right-hand panel). Images courtesy of Synopsys.

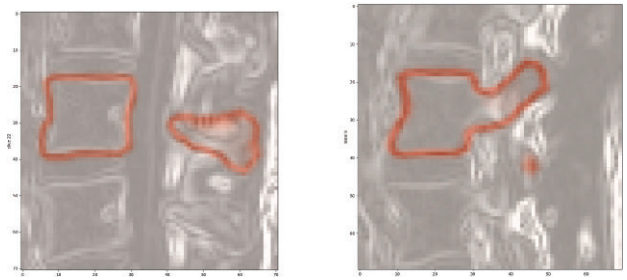

Figure 17: In red the generated images and the original image in the background. The regression parameters : g_{max}, l, and σ are the ones obtained with the maximum likelihood method for a given patched-NURBS surface.

ance of σ. This is particularly relevant due to the non-uniform nature of the vertebrae contour depicted in Fig. 16. By considering the stochastic nature of the noise, we establish a suitable framework that enables us to employ maximum likelihood techniques and the Akaike Information Criterion (AIC) for further analysis and inference.

Then, the greyscale can be expressed as a random variable Y of probability $\mathcal{N}(\bar{g}(X), \sigma^2)$ where X is a given position in the 3D space.

Thus, with the data (x_i, y_i) where x_i is the position of the voxel i and y_i the greyscale value of the voxel i, the log-likelihood is given by

$$\mathcal{L}(g_{max}, l, \sigma | y_i, x_i) = \left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(\bar{g}(x_i, g_{max}, l) - y_i \right)^2 - \frac{n}{2} \log(\sigma^2) \right).$$

By minimizing the log-likelihood according to g_{max}, l, and σ, with a quasi-Newton method, L-BFGS, we can generate new images, see Fig. 17, in order to compare them with the original images to run a model selection.

We will now use the log-likelihood to determine the model that accurately represents the images with the fewest possible parameters.

For each model, we compute this log-likelihood, and we will choose the one with the lowest value, the maximum likelihood estimation. Thus, we have the following graph:

However, the graph in Fig. 18 does not allow us to make a clear decision. The minimum is non-obvious. This is why we are using an information criterion, which will help us to make a clear decision for the model selection.

Let us remark that the different peaks in the graph correspond to poor Morse-Smale complexes, i.e.,: low eigenvalue, but with a larger subdivision step. Thus, the number of control points increases but the Morse-Smale complex fails to capture the geometry of the object correctly.

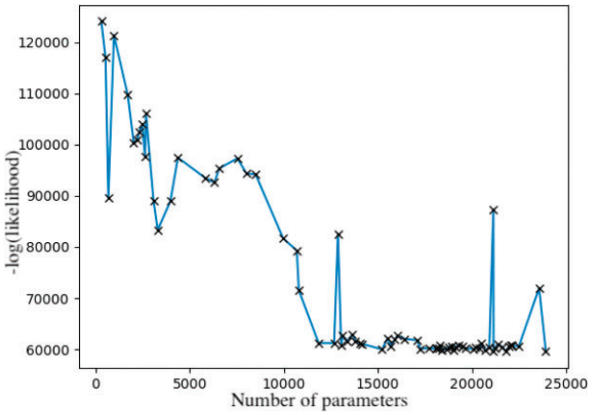

Figure 18: Likelihood according the number of control points. Increasing the number of parameters is done by increasing both the eigen-number and subdivision step.

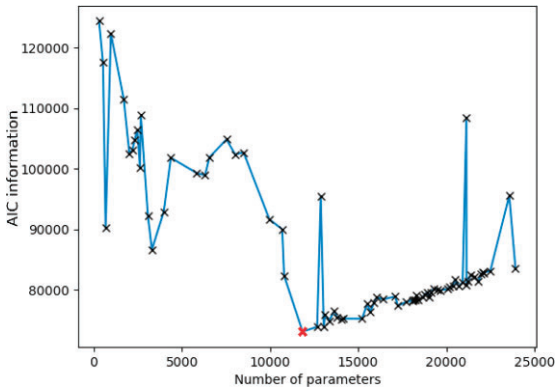

Figure 19: Information criterion for vertebrae model selection.

Here, we are using the Akaike criterion (Akaike, 1998), defined as follows:

(i) For a model $\mathcal{M}$, compute the information:

$$AIC_{\mathcal{M}} = 2k - 2\ln(\hat{\mathcal{L}}_{\mathcal{M}}) = 2k + 2\hat{l}_{\mathcal{M}}$$

where k is the number of parameters of the model, $\hat{\mathcal{L}}_{\mathcal{M}}$ the maximum likelihood value for the model $\mathcal{M}$, and $\hat{l}_{\mathcal{M}} = -\ln(\hat{\mathcal{L}}_{\mathcal{M}})$.

(ii) We choose the model with the minimum information value.

The AIC criterion is based on the likelihood function of a statistical model and considers both the model's goodness of fit and the number of parameters used in the model. Because it balances the trade-off between model complexity and goodness of fit, it is a useful tool for model selection. In our case, models with more parameters tend to fit the data better, but they also tend to be overparametrized, which can lead to poor performance in terms of storage and manoeuvrability in CAD software. Since the AIC criterion penalizes models with more parameters, it is a useful tool for selecting a model that provides a good balance of model complexity and goodness of fit.

Here, the best model in the set of models generated from a given Laplacian eigenproblem is the one with approximately 11 850 regression parameters, marked by the red cross in Fig. 19.

The best model and extreme cases are shown in Fig. 20.

Let us remark, that if the fitting optimization is not used we have the following result, Fig. 21.

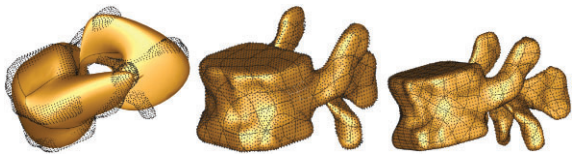

Figure 20: Sub-parametrization, minimum AIC, and overparametrization.

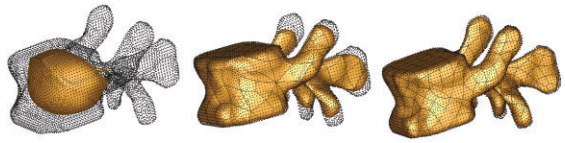

Figure 21: Sub-parametrization, minimum AIC, and overparametrization.

Moreover, we can compute the relative likelihood of a model thanks to the AIC information (Wagenmakers & Farrell, 2004):

$$\mathcal{RL}_{\mathcal{M}_i} = \exp\left(\frac{AIC_i - AIC_{min}}{2}\right).$$

Then we can normalize this value to obtain the Akaike weights:

$$w_i = \frac{\exp\left(\frac{AIC_i - AIC_{min}}{2}\right)}{\sum_{k=0}^{n} \exp\left(\frac{AIC_k - AIC_{min}}{2}\right)}$$

where n is the total number of models. These Akaike weights can be interpreted as the probability that the model $\mathcal{M}_i$ is the best model. For example, the probability that the best model is the one with 10 800 regression parameters is $w_i = 0$. In fact, due to the difference between the AIC value, the only acceptable model is the minimal one with probability 1.

4.3.2. *CT: femur*

In the case of the CT, the use of a gradient filter is not relevant. Indeed, the interior part of the femur is porous, and therefore the gradient filter does not take into account the porous area. This is why we are using the original images without a pre-processing step. We have at our disposal:

(i) x_i = position of the voxel i
(ii) y_i = greyscale value in voxel i

Then, we can construct the following model:

$$y_i = \bar{g}(x_i) + \boldsymbol{\varepsilon}$$

with: $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, $\bar{g}(x) = g_{max}\frac{1}{1+\exp(ad(x_i)+b)}$, and $d(x_i)$ representing the distance between the voxel i and the patched-NURBS surface.

With a maximization according to the likelihood, we can generate new images, as illustrated in Fig. 22. The evolution of $\hat{l}_{\mathcal{M}} = -\ln(\hat{L}_{\mathcal{M}})$ according to the number of parameters is shown in Fig. 23.

As for the vertebrae, it is difficult to make a clear decision on which model to choose. Let us try the AIC criterion as before to determine which model represents accurately the data with the least number of parameters.

However, we are facing a problem because the AIC graph Fig. 24 looks like the likelihood graph. This is due to the fact that the value of $\hat{l}_{\mathcal{M}}$ is of the order of 10^{11} and the number of regression parameters is of the order of 10^5. Therefore, penalizing $\hat{l}_{\mathcal{M}}$ with the number of regression parameters does not affect its value. Therefore, we will use a slightly different information for this model. Instead of using the log-likelihood, we will use the residual squared sum, RSS, as described in (Miao *et al.*, 2009):

$$RSS = \sum_{i=0}^{n} (y_i - \bar{g}(x_i))^2$$

where $y_0, \ldots y_n$ are the data, i.e., greyscale value, $\bar{g}$ is the function defined above, and $x_0, \ldots, x_n$ the position of the voxels.

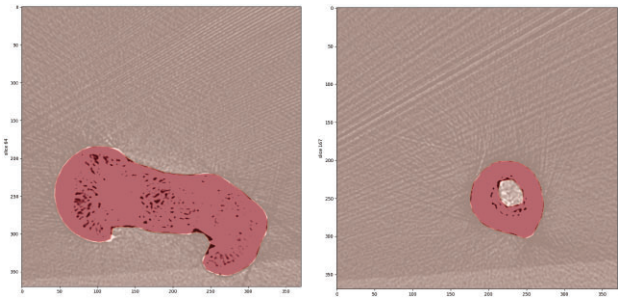

Figure 22: In red the generated images and in the background the original image (the CT scan).

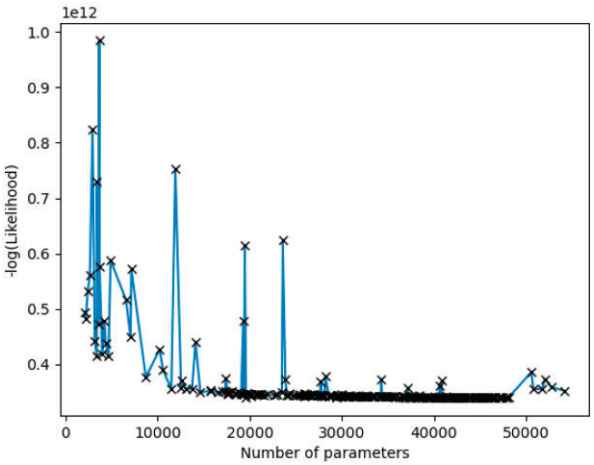

Figure 23: Likelihood femur.

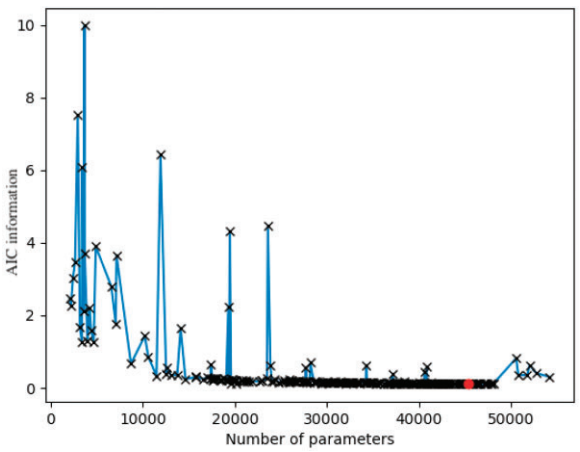

Figure 24: Information criterion for femur model selection.

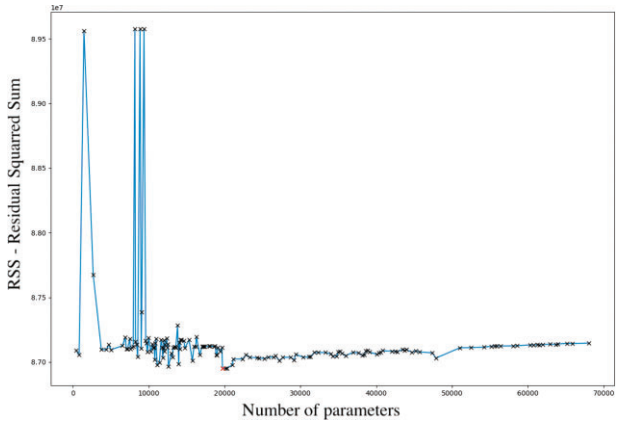

Figure 25: Information criterion with RSS for femur model.

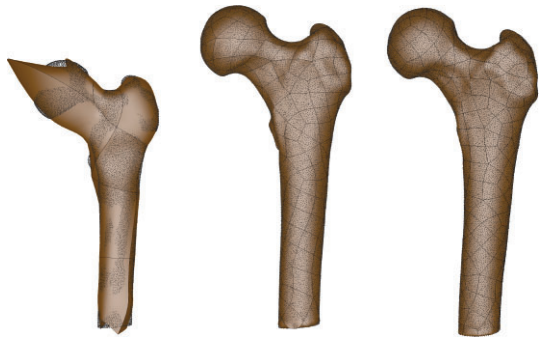

Figure 26: 450 Control points (left-hand panel), 19 600 control points – AIC minimum (middle panel), and 61 965 control points (right-hand panel).

Then the AIC information for a given model $\mathcal{M}$ can be reformulated as follows:

$$AIC_{\mathcal{M}} = n \log\left(\frac{RSS_{\mathcal{M}}}{n}\right) + 2k$$

where n is the number of data points and k the number of parameters of the model $\mathcal{M}$.

The graph of the AIC information with RSS is shown in Fig. 25. The minimum of information is shown with the red cross.

Different examples of the femur are shown in Fig. 26. The AIC minimum is the one in the middle.

We have been able to obtain the best model in the sense of the definition above, now we will try to obtain a surface probability distribution around these fixed optimal control points. Thus, we will use the generative model defined above and then a sampling method based on Markov chains, the HMCMC.

5. Hamiltonian Markov Chain Monte Carlo

Probabilistic modelling is of general interest to computational engineering. It first simplifies model selection, which is largely discussed in chapter 1 of Bishop and Nasrabadi (2006). Secondly, it allows for uncertainties to be propagated when doing inference. For instance, if shape biomarkers are to be extracted from the reconstructed surfaces, confidence intervals can be calculated for these biomarkers, which fully encode the effect of data uncertainty and that of the surface reconstruction process. Confidence intervals could also be obtained if partial differential equations, (e.g., hyperelasticity) are to be solved to compute physics-based quantities of interest.

In our case, the statistical model generation has been established, and further advances have been made by incorporating a prior probability density distribution that encodes the lack of knowledge about the patched-NURBS surface. Utilizing a Bayesian approach, a probability distribution of surfaces will be determined, with the aim of creating an interval of confidence based on both the noise present in the image and the prior knowledge of the regression model weights (Bishop & Nasrabadi, 2006). This interval will provide a measure of certainty in the estimates derived from the scan data. Additionally, an automatic regularization method of the ridge type will be incorporated to address the calibration problem; the weight associated with this regularization being determined basing ourselves on the ratio of knowledge to noise in the data. This will ensure that the model does not overparametrize the data. Therefore, we are using the HMCMC, more precisely the Langevin Monte Carlo. We will briefly present the method and then apply it to the vertebrae example. More details about HMCMC and Langevin Monte Carlo can be found in Girolami and Calderhead (2011), Neal *et al.* (2011), and Betancourt (2018).

5.1. Vertebrae example

The model we built in Section 4 for the vertebrae, was used in a statistical approach. Now to use the HMCMC method, this model is considered in the Bayesian setting. Therefore, let us consider the control points and the regression parameters as random variables, and let $\theta = (g_{\max}, l, \sigma, P_1, \ldots, P_n)$ be the associated random vector. Thus, we wish to determine the probability destruction $\mathbb{P}(\theta|X, Y)$. Let us write $Z = (X, Y)$. By using the Bayes theorem we have

$$\underbrace{\mathbb{P}(\theta|Z)}_{\text{posterior}} \propto \underbrace{\mathbb{P}(Z|\theta)}_{\text{likelihood}} \underbrace{\mathbb{P}(\theta)}_{\text{prior}}.$$

In order to sample $\mathbb{P}(\theta|Z)$, we will draw a sample from $\mathbb{P}(Z|\theta)\mathbb{P}(\theta)$ by using the HMCMC sampling method.

Let us recall the vertebrae generative model:

(i) X random vector in $\mathbb{R}^3$ representing the position of a voxel.
(ii) Y random variable representing the greyscale value of the voxel in position X.

$$Y = \bar{g}(X) + \boldsymbol{\varepsilon}_\sigma$$

with: $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and $\bar{g}(x) = g_{\max} e^{-\frac{d(x)^2}{2l^2}}$.

In this model the only parameters are the parameters of the regression; however, we would also like to take into account the parameters of the patched-NURBS surface. Thus, let us write the dependency of the optimal patched-NURBS surface S in the way S_P where $P = (P_i)_{i \in [\![0,N]\!]}$ represents the control points of the patched-NURBS surface without redundancy as in Section 2.3. Hence, we can rewrite the model as follows:

$$Y = \bar{g}(X) + \boldsymbol{\varepsilon}_\sigma$$

with: $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and $\bar{g}(x) = g_{\max} \exp(-\frac{\|x - S_P\|^2}{2l^2})$.

Thus, with $x = (x_i)_i$ and $y = (y_i)_i$, the log-likelihood is given by

$$\mathcal{L}(g_{\max}, l, \sigma, P|x, y) = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(\bar{g}(x_i, g_{\max}, l, P) - y_i\right)^2 - \frac{n}{2} \log(\sigma^2).$$

Now, let us write the random vector $\theta = (g_{\max}, l, \sigma, P_1, \ldots, P_n)$ since we let all the parameters be random variables. We aim to sample the distribution $\mathbb{P}(\theta|Z)$ where $Z = (X, Y)$, by using the Bayes theorem as in the previous section, we have

$$\mathbb{P}(\theta|Z) \propto \mathbb{P}(Z|\theta)\mathbb{P}(\theta).$$

$\mathbb{P}(Z|\theta)$ is given by $\mathcal{L}(g_{\max}, l, \sigma, P|Z)$. However, since we do not know $\mathbb{P}(\theta)$, we can make the assumption that the prior probability is Gaussian, therefore $\theta \sim \mathcal{N}(m_\theta, \Sigma_\theta)$. m_θ is taken as the value minimizing the regression model and Σ_θ as a multiple of the identity matrix.

Thus, by introducing the auxiliary variable, ν for the HMCMC method and by denoting the data $z = (x_i, y_i)_i$, we can evaluate the kinetic energy K and the potential energy E:

(i) $K(\upsilon) = \frac{1}{2}\nu\Sigma^{-1}\nu^T$ and
(ii) $E(\theta) = \mathcal{L}(g_{\max}, l, \sigma, P|z) + \frac{1}{2}(\theta - m_\theta)\Sigma_\theta^{-1}(\theta - m_\theta)^T$.

To generate transitions, we draw a sample of ν from the multivariate Gaussian distribution. Then, we wish to use the Hamilton equation to generate a possible new value for θ. To do so, we will use the leapfrog integrator. These two steps can be summarized as follows:

(i) Draw ν_i from $\mathcal{N}(0, \Sigma)$
(ii) Leapfrog integration:

$$\begin{cases} \nu_{i-1/2} = \nu_i - \frac{\varepsilon}{2}\nabla E(\theta_i) \\ \theta_{i+1} = \theta_i + \varepsilon\Sigma\nu_{i-1/2} \\ \nu_{i-3/2} = \nu_{i-1/2} - \frac{\varepsilon}{2}\nabla E(\theta_{i+1}) \end{cases}$$

where ε is a given time step. We drop the dependency in Z because Z represents the data, therefore Z can be replaced by the observed value to evaluate K and E.

Now, we have a new state $(\theta_{i+1}, \nu_{i-3/2})$. We will keep with the sample as a sample of the posterior according to the probability defined by the Metropolis acceptance rate:

$$min(0, H(\theta_i, \nu_i) - H(\theta_{i+1}, \nu_{i-3/2})).$$

If the newly proposed state is rejected, the previous value is used again for the next iteration. By repeating this process numerous times, the all-probability distribution is explored, therefore the distribution $\mathbb{P}(\theta|Y)$ is determined.

We tried to run the HMCMC with a mass matrix equal the identity matrix. However, the size of the gradient in the regression parameters was larger than the one for the control points, therefore the effect of the HMCMC on the control points was negligible. Hence, we tried with the Hessian matrix at the maximum a posteriori, $\Delta E(\theta_{MAP})$ and the inverse of the prior covariance matrix, Σ_θ^{-1}. This leads us to a better representation of the probability distribution.

We plot several samples of patched-NURBS surfaces in Figs. 27 and 28. In fact, the results are close to each other, therefore choosing the Hessian matrix or the inverse of the covariance does not make a difference.

Figure 29 illustrates two different samples for one slice of greyscale images and Figure 30 shows the pre-processed image and a generated greyscale image.

6. Conclusions

We have proposed an automatic method for the generation of patched-NURBS surfaces from images, with a stochastic model selection. First, our method is based on the Discrete Morse theory and more precisely on the generation of the Morse-Smale complex. The generation of such a structure is done using the critical points and integral lines of a scalar field. In practice, we have used the eigenfunctions of the graph Laplacian to obtain such a scalar field. The advantage of using such functions is that their critical points are uniformly spaced on the object, thus allowing to capture the general shape of the object and to preserve its topology. However, the quad mesh obtained with TTK is optimal with respect to the triangulation, therefore it leads to a patched-NURBS surface that is not faithful to the triangulation. Hence, we introduced an optimization process using a quasi-Newton method, L-

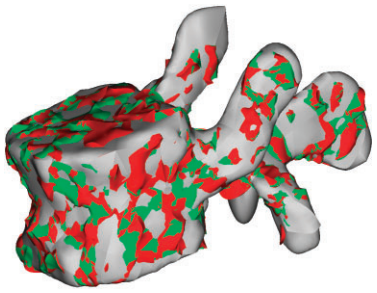

Figure 27: Three patched-NURBS surface samples obtained with the Hessian matrix.

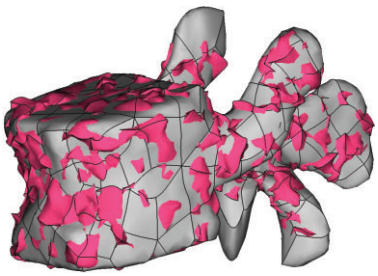

Figure 28: Two patched-NURBS surface samples obtained with the inverse covariance matrix.

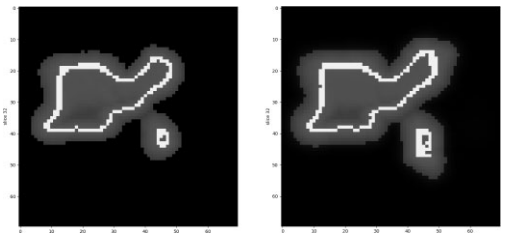

Figure 29: Two greyscale reconstructions from two different samples, the white voxels representing the surfaces.

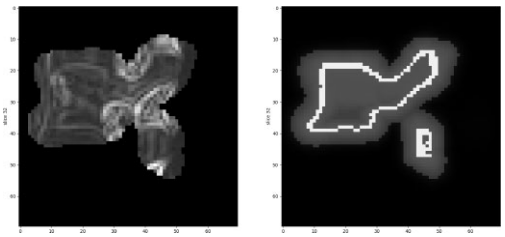

Figure 30: Left-hand panel: Pre-processed image data. Right-hand panel: Generated greyscale image, the white voxels representing the surface.

BFGS. This optimization step allows us to consider valid surfaces with a smaller number of control points. Then, to allow a choice among the models generated using the eigenvalue problem, we used a method comparing the patched-NURBS surface obtained directly to the images, by introducing a regression model to generate new greyscale images. Then, by maximizing the likelihood and using the AIC criterion which penalizes the likelihood with the number of parameters, we can choose the model with the smallest number of parameters but which still accurately represents the data. Furthermore, we used the generative model by letting the control points become random variables in order to obtain a surface probability distribution. Thus, we used a sampling method based on the dynamic Hamiltonian and Metropolis algorithm, HMCMC. Hence, we have access now, not only to one patched-NURBS surface but to a complete probability distribution.

Future work is needed to adapt the patch density according to the local complexity of the 3D object, either by taking into account the density of the triangulation or the curvature of the surface.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Actions Grant agreement No. 764644. This paper only contains the author's views andthe Research Executive Agency and the Commission are not responsible for any use that may be made of the information it contains. S.P.A. BORDAS thanks for the support of the Fonds National de la Recherche Luxembourg FNR grant O17-QCCAAS-11758809. S.P.A. BORDAS received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 811099 TWINNING Project DRIVEN for the University of Luxembourg.

We thank Synopsys Northern Europe Ltd for its support and specifically Dr Viet Bui Xuan for his constructive suggestions during the planning and development of this research work.

Conflict of interest statement

None declared.

References

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike* (pp. 199–213). Springer. https://doi.org/10.1007/978-1-4612-1694-0_15.

Anderson, C. W., & Crawford-Hines, S. (2000). Fast generation of NURBS surfaces from polygonal mesh models of human anatomy. In *Colorado State University Computer Science Technical Report CS-99*(Vol. **101**). Colorado State University.

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. https://arxiv.org/pdf/1701.02434.pdf.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. **4**). Springer.

Bommes, D., Zimmer, H., & Kobbelt, L. (2009). Mixed-integer quadrangulation. *ACM Transactions on Graphics*, **28**, 1–10. https://doi.org/10.1145/1531326.1531383.

Bommes, D., Lévy, B., Pietroni, N., Puppo, E., Silva, C., Tarini, M., & Zorin, D. (2013). Quad-mesh generation and processing: A survey. In *Computer graphics forum* (Vol. **32**, pp. 51–76). Wiley Online Library. https://doi.org/10.1111/cgf.12014.

Boujraf, A., Sbibih, D., Léger, C., & Harba, R. (2012). A spline quasi-interpolant for fitting 3D data on the sphere and applications. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)* (pp. 1841–1844). IEEE.

Bærentzen, A., & Aanæs, H. (2005). Signed distance computation using the angle weighted pseudonormal. *IEEE Transactions on Visualization and Computer Graphics*, **11**, 243–53. https://doi.org/10.1109/TVCG.2005.49.

Byrd, R., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal of Scientific Computing*, **16**, 1190–1208. https://doi.org/10.1137/0916069.

Dong, S., Bremer, P.-T., Garland, M., Pascucci, V., & Hart, J. C. (2006). Spectral surface quadrangulation. *ACM Transactions on Graphics*, **25**, 1057–1066. https://doi.org/10.1145/1141911.1141993.

Eck, M., & Hoppe, H. (1996). Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)* (pp. 325–334). Association for Computing Machinery. https://doi.org/10.1145/237170.237271.

Fang, X., Bao, H., Tong, Y., Desbrun, M., & Huang, J. (2018). Quadrangulation through morse-parameterization hybridization. *ACM Transactions on Graphics*, **37**, 1–15. https://doi.org/10.1145/3197517.3201354.

Forman, R. (2002). A user's guide to discrete morse theory. *Seminaire Lotharingien de Combinatoire*, **48**, 35.

Girolami, M., & Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **73**, 123–214. https://doi.org/10.1111/j.1467-9868.2010.00765.x.

Gyulassy, A., Bremer, P.-T., Hamann, B., & Pascucci, V. (2008). A practical approach to morse-smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, **14**, 1619–1626. https://doi.org/10.1109/TVCG.2008.110.

Hormann, K., & Greiner, G. (2000). Quadrilateral remeshing. In *Proceedings of the International Symposium on Vision, Modeling, and Visualization (2000)*(pp. 153–162).

Hughes, T., Cottrell, J., & Bazilevs, Y. (2005). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, **194**, 4135–4195. https://doi.org/10.1016/j.cma.2004.10.008.

Kälberer, F., Nieser, M., & Polthier, K. (2007). Quadcover-surface parameterization using branched coverings. In *Computer graphics forum* (Vol. **26**, pp. 375–384). Wiley Online Library. https://doi.org/10.1111/j.1467-8659.2007.01060.x.

Li, X., Wu, Z., Pan, F., Liang, J., Zhang, J., & Hou, L. (2019). A geometric strategy algorithm for orthogonal projection onto a parametric surface. *Journal of Computer Science and Technology*, **34**, 1279–1293. https://doi.org/10.1007/s11390-019-1967-z.

Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, **45**, 503–528. https://doi.org/10.1007/BF01589116.

Miao, H., Dykes, C., Demeter, L. M., & Wu, H. (2009). Differential equation modeling of HIV viral fitness experiments: Model identification, model selection, and multimodel inference. *Biometrics*, **65**, 292–300. https://doi.org/10.1111/j.1541-0420.2008.01059.x.

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo* (pp. 113–162). Chapman & Hall/CRC.

Nguyen, V. P., Anitescu, C., Bordas, S. P., & Rabczuk, T. (2015). Isogeometric analysis: An overview and computer implementation aspects. *Mathematics and Computers in Simulation*, **117**, 89–116. https://doi.org/10.1016/j.matcom.2015.05.008.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization*. (2nd ed.). Springer. https://doi.org/10.1007/b98874.

Owen, S. J., Staten, M. L., Canann, S. A., & Saigal, S. (1999). Q-Morph: An indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering*, **44**, 1317–1340. https://doi.org/10.1002/(SICI)1097-0207(19990330)44:9%3C1317::AID-NME532%3E3.0.CO;2-N.

Piegl, L., & Tiller, W. (1996). *The NURBS book*. Springer. https://doi.org/10.1007/978-3-642-59223-2.

Ray, N., Li, W. C., Lévy, B., Sheffer, A., & Alliez, P. (2006). Periodic global parameterization. *ACM Transactions on Graphics*, **25**, 1460–1485. https://doi.org/10.1145/1183287.1183297.

Schroeder, W., Martin, K., & Lorensen, B. (2006). *The visualization toolkit–An object-oriented approach to 3D graphics*. (4th ed.). Kitware, Inc.

Shivashankar, N., M, S., & Natarajan, V. (2012). Parallel computation of 2D morse-smale complexes. *IEEE Transactions on Visualization and Computer Graphics*, **18**, 1757–1770. https://doi.org/10.1109/TVCG.2011.284.

Sobel, I. (2014). An isotropic 3x3 image gradient operator. In *Presentation at Stanford A.I. Project 1968*.

Tarini, M., Puppo, E., Panozzo, D., Pietroni, N., & Cignoni, P. (2011). Simple quad domains for field aligned mesh parametrization. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (pp. 1–12). https://doi.org/10.1145/2024156.2024176.

Tierny, J. (2017). *Topological data analysis for scientific visualization* (Vol. **3**). Springer. https://doi.org/10.1007/978-3-319-71507-0.

Tierny, J., Favelier, G., Levine, J. A., Gueunet, C., & Michaux, M. (2018). The topology toolkit. *IEEE Transactions on Visualization and Computer Graphics*, **24**, 832–842. https://doi.org/10.1109/TVCG.2017.2743938.

Wagenmakers, E.-J., & Farrell, S. (2004). AIC model selection using Akaike weights. *Psychonomic Bulletin & Review*, **11**, 192–196. https://doi.org/10.3758BF03206482.

Received: January 17, 2023. **Revised:** July 2, 2023. **Accepted:** July 29, 2023