# Smooth parametric surfaces retrieval from triangular meshes using RBFs

Antoine Jonquet, Olivier Nocent, Yannick Rémion

# Smooth parametric surfaces retrieval from triangular meshes using RBFs

Antoine Jonquet, Olivier Nocent, Yannick Remion
CReSTIC LERI,
University of Reims,
Reims, France
jonquet@leri.univ-reims.fr, {olivier.nocent, yannick.remion}@univ-reims.fr

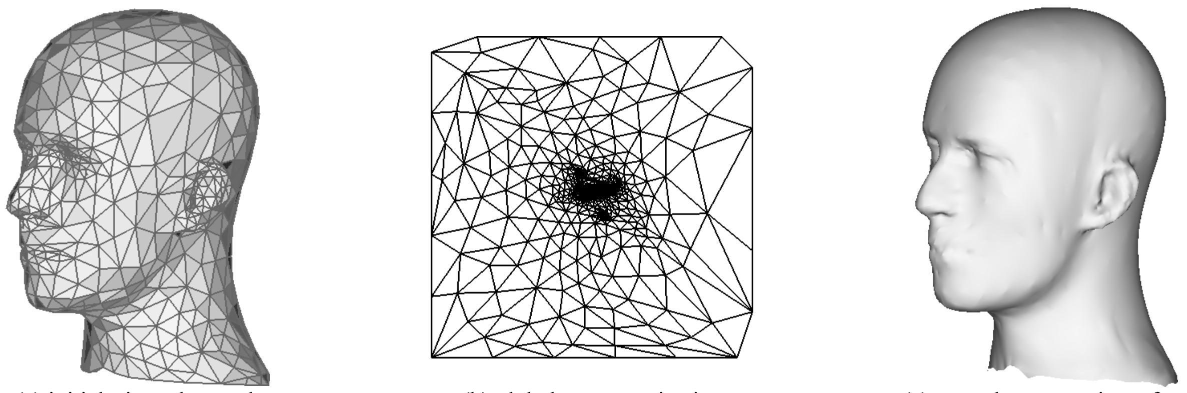

(a) initial triangular mesh
(b) global parameterization
(c) smooth parametric surface

**Figure 1:** The two major stages of our method: from an initial triangular mesh (a), we compute a global parameterization (b) to retrieve a smooth parametric surface (c).

## Abstract

Quality of physically-based simulation mostly relies upon qualities of the geometrical model of the entities involved. Particularly, Lagrangian formalism (which we focus on since several years) claims for continuous and smooth descriptors. Our current project aims a precise Lagrangian knee kinematics model. The studied knee model has been chosen as built up from actual bones geometries and ligament attachments. In this scheme, contact between femur and tibia must be seen as a constraint between two smooth surfaces. But, most of the time, actual geometric data coming from acquisition devices are modeled as discrete quantities. In this paper, we present a method to retrieve a smooth parametric surface from a triangular mesh. The first step calculates a global parameterization of a disk-like mesh. The second step builds a smooth parametric surface that interpolates the actual data using Radial Basis Functions (RBFs). This scheme allows for smoothly expressing any geometric quantity like tangent vectors or curvature at any point on the smooth surface. Out of the biomechanics area, we mention other possible applications of our framework for mesh refinement and non photorealistic rendering.

***Keywords:*** *Geometric modeling, smooth parametric surfaces, global parameterization, surface interpolation.*

## 1. INTRODUCTION

The content of this paper takes place into a research project aiming to propose a realistic model of the knee joint. Although the structure of this articulation is very complex, it has been widely studied, mainly due to the number and variety of its pathologies. Those works fall into two main categories. The first one is based upon rigid body dynamics in order to determine the bones evolution. Collisions are managed according to a discrete representation of the surface [1][2]. The other category relies on the Finite Elements Method (FEM) [3][4]. Even if the FEM allows dealing with soft bodies such as ligaments, it shares the same drawback as the previous one: it relies on a discrete description of bones shapes. The quality of the simulation is then dependent on the refinement of geometrical data. Moreover, polygonal surfaces may produce numerical instability when contacts are taken into account.

By now, our study is limited to the identification of the effective complex degrees of freedom of the knee joint during the flexion/extension movement. Since we focus on physically-based simulation rather than plausible motion simulation [5], it appears to us that bones surfaces are of great interest. To avoid the drawbacks of the previously mentioned works and in accordance to our own experience in simulation, we are willing to deal with the continuous aspect of the phenomenon [6][7]. To remain as close as possible to the actual motion, we reject all methodological simplifications that would reduce the bone contact as a sphere-plane contact for example [8]. It is now well known in the biomechanics area that knee joint can not be depicted by classical joints like ball joints, hinges or sliders. For all these reasons, we propose to manage the bone contact as a contact between two smooth surfaces. Implicit isopotential surfaces have been successfully used for the animation of deformable models [9]. Field functions of each surface are used to detect interpenetration and to generate an appropriate response based on a deformation field function. This method, based on a continuous description of surfaces, is unfortunately restricted to deformable bodies and does not rely on the physical properties of the surfaces involved. For our concern, we neglect deformations and express the contact as a sliding contact constraint between two rigid bodies. The continuous tracking of the contact point, though still marginal, is an increasing centre of interest in the computer graphics community [10].

Data used to set up the simulation is obtained from acquisition devices such as 3D and MRI scanners. As a consequence, this kind of data is usually represented by a 3D point cloud or a 3D triangular mesh. In this article, we suggest a method to generate automatically an interpolating parametric surface from a 3D triangular mesh. This method is composed of two stages (e.g. Figure 1). The first stage calculates a global parameterization of the 3D triangular mesh. This parameterization must be global to guarantee the continuous tracking of the contact point during the flexion/extension movement. The second stage then builds up a smooth parametric surface using Radial Basis Functions (RBFs) that interpolates the 3D mesh. As the surface is explicitly defined, the contact point is naturally identified by its 2D parametric coordinates in global parametric domain. This method settles our problem as it gives us a smooth parametric representation of our data.

In the following section, we describe in details these two steps. We pursue by presenting the application of this framework to our initial problem: the continuous tracking of the contact point between two surfaces. Then, convinced of the usefulness of continuous representation of a 3D mesh, we mention other meaningful applications, like mesh refinement and non-photorealistic rendering, both based on parametric lines and continuous curvature fields.

## 2. PARAMETERIZATION

During the flexion/extension movement, the contact points are localized at bones ends. So, to continuously track these contact points, we just need a global parameterization of these interest regions of bones. That is why we have limited our investigation to techniques that compute a global parameterization of disk-like triangulated surfaces. Of course, it would be interesting to extend our method in order to handle sphere-like surfaces, by using spherical parameterization [11] or even arbitrary meshes by cutting and flattening [12]. But the last technique introduces discontinuities on the borders of the parametric domain that could perturb the continuous tracking of the contact point.

The parameterization stage consists in associating each vertex of a 3D mesh with 2D coordinates $(u_i,v_i)$ belonging to the parametric domain $[0,1]\times[0,1]$, in order to obtain a one-to-one mapping between a parametric domain and the 3D surface. Although convex surfaces could easily be parameterized using a projection on plane, concave surfaces could overlap during the projection. So it is not so easy to find an automatic way to parameterize a 3D triangular mesh.

Planar graph theory gives an interesting formalism to study this kind of problems. Indeed, it is possible under some conditions to express a planar graph from a 3D mesh. It remains to map this graph to the wished parametric domain. Tutte [13] introduces the notion of barycentric mapping to set a planar graph to 2D space. Floater [14] re-uses this concept by introducing convex combinations. The convex combination of vertices is computed in order to minimize ad hoc mesh deformation energy [15]. To describe this last method, we have to remind some elements of graph theory:

A graph $G = G(V,E)$ is composed of a finite set of vertices $V = \{\mathbf{v}_1,\cdots,\mathbf{v}_n\}$ and a set of edges $E = \left\{\{i,j\} : 1 \le i,j \le n, i \ne j\right\}$. The vertex $\mathbf{v}_j$ is a neighbor of $\mathbf{v}_i$ if $\{i,j\} \in E$. By convention, neighbors of $\mathbf{v}_i$ are notated $\mathbf{v}_{ij}$ for $j = 1,\cdots,d_i$. The degree $d_i$ (or valency) of the vertex $\mathbf{v}_i$ is the total number of its neighbors. A graph $G = G(V,E)$ is said to be planar if all its vertices can be projected on distinct points of the plane with no secant edges.

A planar graph subdivides the parametric domain in regions. Parameterization is about determining a graph from a 3D mesh and setting it on the parametric domain. As the parameterization is a one-to-one mapping, it does not allow for overlapping as each point of the parametric domain must define a unique 3D point.

At first, to obtain this result, Floater sets the boundaries vertices of the graph on the border of the parametric domain. The mapping of the 3D border to the parametric one could be done by uniform or chord length distribution in order to take into account edges lengths of the 3D mesh (e.g. Figure 2).

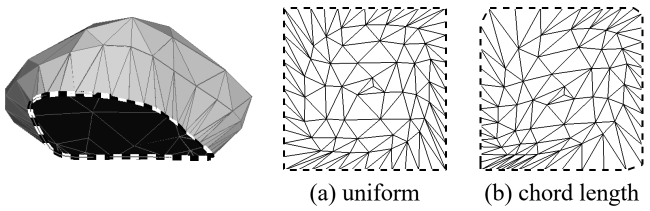


**Figure 2:** Boundary parameterization.

Afterward, Floater expresses the parametric coordinates $(u_i,v_i)$ of inner vertices $\mathbf{v}_i$ of the graph as convex combinations of the parametric coordinates $(u_{ij},v_{ij})$ of their neighbors $\mathbf{v}_{ij}$ (Eq. 2.1). This way, he can initialize a linear system which insures that the solution does not allow any overlapping.

$$u_i = \sum_{j=1}^{d_i} \alpha_{ij} u_{ij} \quad , \quad v_i = \sum_{j=1}^{d_i} \alpha_{ij} v_{ij} \quad \text{with} \quad \sum_{j=1}^{d_i} \alpha_{ij} = 1 \tag{2.1}$$

The coefficients $\alpha_{ij}$ of the convex combination contribute to the deformation of the graph during its mapping to the parametric domain. One solution is to set the same weight to all neighbors which give the barycentric parameterization as illustrated by (Eq. 2.2).

$$\alpha_{ij} = d_i^{-1} \qquad \forall j = 1,\cdots,d_i \tag{2.2}$$

Other solutions are used to choose weights of neighbors of a vertex, such as to take into account the area or shape of the faces. The method that we chose was proposed by Floater. It consists in preserving the shape of the faces of the 3D mesh [14]. This method projects the vertex we are interested in and its neighbors on a local parametric plane, and then determine the best coefficient for each neighbor.

To conclude, this method gives the opportunity to obtain distinct parametric coordinates $(u_i,v_i)$ for each vertex $\mathbf{v}_i$ of the 3D mesh without any overlapping. Furthermore, this method gives some control on the way the 3D mesh is mapped to the parametric domain. In fact, one can choose the shape of the parametric domain border and the way the 3D border is mapped on. Moreover, on can choose the way the $\alpha_{ij}$ are computed to control the mesh deformation.

## 3. INTERPOLATION

After the first step, we have 2D parameters $(u_i, v_i)$ for each vertex $\mathbf{v}_i$ of the initial 3D mesh. Then, the second step consists in the construction of a smooth parametric surface $\mathbf{s}$ that interpolates our discrete data set $\{\mathbf{p}_1, \cdots, \mathbf{p}_n\}$ where $\mathbf{p}_i$ is the 3D position of the vertex $\mathbf{v}_i$ (Eq.3.1).

$$\mathbf{s} : [0,1]^2 \to \mathrm{R}^3 \ , \ \mathbf{s}(u_i, v_i) = \mathbf{p}_i \ \ \forall i = 1, \cdots, n \qquad (3.1)$$

Our application requires an interpolating surface defined from a unique global parameterization to continuously track the sliding contact points all over the bones surfaces. So we have excluded all local interpolation techniques based on patches, such as local parametric surfaces and subdivision surfaces. Moreover, in order to define the contact constraint, we have to obtain a smooth enough expression of the interpolating surface. To insure this continuity, patches based surfaces should impose constraints on patches boundaries, restricting the method to particular mesh topologies. We remind that we do not control the spatial repartition of vertices since they come from acquisition devices.

Radial Basis Functions (RBFs) are widely used to construct implicit surfaces from 3D point sets [16]. This method consists in defining a continuous signed potential field $f : \mathrm{R}^3 \to \mathrm{R}$ in 3D space that must be null for each vertex $f(\mathbf{p}_i) = 0 \ \forall i = 1, \cdots, n$ . To obtain other solution than the trivial $f(\mathbf{p}) = 0 \ \ \forall \mathbf{p} \in \mathrm{R}^3$ , we need to complete the initial point set with inner ( $f(\mathbf{p}) < 0$ ) and outer ( $f(\mathbf{p}) > 0$ ) points.

This method increases considerably the dimension of the linear system to resolve in order to obtain the weight associated with each vertex. Moreover, an implicit surface is not really suitable to identify easily the contact point, so it is not easy to continuously track it during the animation.

Nevertheless, RBFs have interesting properties such as independence according to the geometrical repartition of the vertices. The use of RBFs to construct parametric surfaces was studied for image warping [17][18]. We propose an improvement of the method by using it in the 3D space. Our interpolating surface is defined as follow:

$$\mathbf{s}(u,v) = \sum_{i=1}^{n} \boldsymbol{\lambda}_i \phi_i\left(d_i(u,v)\right) \qquad (3.2)$$

$d_i(u,v)$ is the square of the Euclidian distance between two points of coordinates $(u,v)$ and $(u_i, v_i)$ belonging to $[0,1] \times [0,1]$ (Eq.3.3).

$$d_i(u,v) = (u - u_i)^2 + (v - v_i)^2 \ \ \forall i = 1, \cdots, n \qquad (3.3)$$

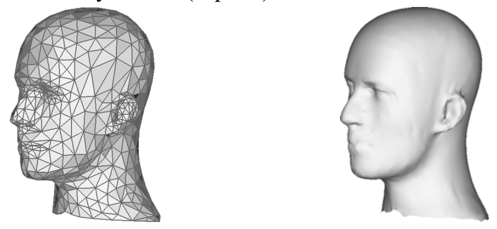
The basis function $\phi_i$ is in our case a multiquadric function that is for *x* being the square of the Euclidian distance:

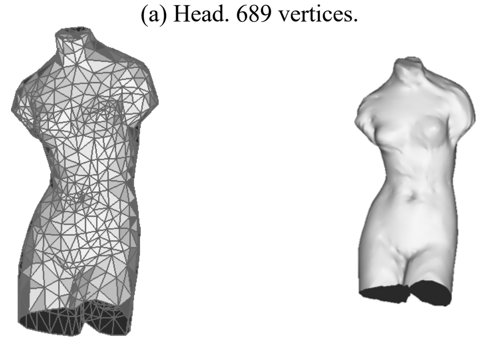
$$\phi_i(x) = \sqrt{x + c_i} \ , \ c_i = \min_j d_i\left(u_{ij}, v_{ij}\right) \qquad (3.4)$$

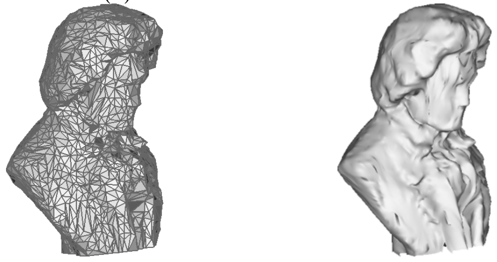
The expression of the interpolating surface depends mainly on the 3D weights $\boldsymbol{\lambda}_i = (\lambda_i^x \ \ \lambda_i^y \ \ \lambda_i^z)^T$ of the vertices $\mathbf{p}_i = (p_i^x \ \ p_i^y \ \ p_i^z)^T$ . From (Eq. 3.2) it is obvious that these weights are solutions of the following linear system:

$$\begin{gathered} \mathbf{A}\left(\lambda_1^\alpha \cdots \lambda_n^\alpha\right)^T = \left(p_1^\alpha \cdots p_n^\alpha\right)^T \qquad \forall \alpha = x, y, z \\ \text{where } \mathbf{A} = \left(a_{ij}\right) \text{ with } a_{ij} = \phi_i\left(d_i\left(u_j, v_j\right)\right) \ \forall i, j = 1,..,n \end{gathered} \qquad (3.5)$$

We must highlight that, because of the characteristics of the chosen $\phi_i(x)$ functions, the matrix $\mathbf{A}$ would be neither symmetric nor sparse, and usually badly conditioned. We use LU decomposition scheme of the GNU Scientific Library (GSL) to solve the linear system of (Eq. 3.5).


(a) Head. 689 vertices.


(b) Venus statue. 711 vertices.


(c) Beethoven chest. 2000 vertices.

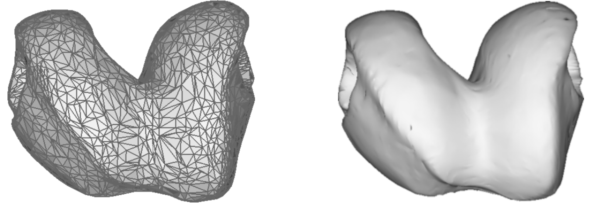

(d) Femur end. 2173 vertices.

**Figure 3:** Sample meshes and their associated smooth parametric surface.

| | Parameterization | | RBF construction | |
|---|---|---|---|---|
| | Mean Time | Standard Error | Mean Time | Standard Error |
| (a) Head | 2 630 | 61 | 4 645 | 116 |
| (b) Venus | 2 767 | 73 | 4 807 | 123 |
| (c) Beethoven | 66 699 | 1 631 | 106 470 | 2 693 |
| (d) Femur end | 80 615 | 2 017 | 130 221 | 3 254 |

**Table 1:** Numerical results (in msecs) extracted from 20 computations (Athlon XP2200+, 512 Mo RAM).

We are aware of the computation cost of this method as well as of the fact that the algorithm we used to solve the linear system restrains the number of vertices. Nevertheless, we plan to use local support surfaces such as unity partition based on RBFs [19] [20]. This method consists in decomposing the main set of vertices in many smaller subsets. The local surfaces are then blended using polynomial weights. By this way, there is not anymore limitation concerning the mesh resolution, and we will be able to reduce significantly the computation time. Moreover, the introduction of polynomials to sum local surfaces would reduce the oscillation phenomena that could appear with global support RBFs.

After pre-computations of the coefficients $\boldsymbol{\lambda}_i$, we have an interpolating parametric surface that is defined all over the parametric domain $[0,1]\times[0,1]$. Furthermore, this surface is smooth enough thanks to the radial functions $\phi_i(x)$ chosen.

The advantage of this explicit approach of the RBFs, is that we didn't have to use an algorithm such as marching cube [16] to obtain a new triangulation of the surface. We just have to choose regular or adaptive subdivisions of the parametric domain to generate a new triangulation if it is necessary.

## 4. CONTINUOUS CONTACT SIMULATION FOR SMOOTH SURFACES

In order to achieve a realistic mechanical simulation of the knee joint, we need to model the contact between femur and tibia in an efficient and robust way. According to physically-based modeling principles, each bone is considered as a free rigid body whose evolution is governed by six degrees of freedom (DOF) representing both position and orientation. Without any contact, our complete mechanical system is composed of twelve DOF. The contact between two smooth surfaces can be managed in two different ways.

The first approach consists in manually reducing the total number of DOF to decrease the dimension of possible configuration space (position and orientation in this case). Kry et al. [10] present a scheme to express the relative configuration of the first rigid body according to the current configuration of the second one. This process, that deletes redundant parameters, guarantees that the surfaces will stay in contact in spite of the round-off errors due to numerical calculations. But this approach has a main drawback: because the contact is implicitly integrated in the reduced set of DOF, it can't be released. Unfortunately, during a complete flexion of the knee joint, there is no more contact between femur and tibia.

The second approach consists in expressing the contact as a dynamic constraint. The main advantage of this approach, especially in the context of knee joint simulation, is that a constraint can be easily released at any time during the simulation process. This constraint is then integrated in our system as an equation relying on the twelve initial DOF and new unknowns identifying the position of the sliding contact point. In a technical report, Remion [21] proposed a new formalism in order to manage such new unknowns called *free variables* as new virtual DOF. This method allows for expressing new dynamic constraints like curve-curve, curve-surface and surface-surface contact constraints. This technique has been successfully used by Lenoir et al. [22] for surgical simulations.

From the parametric expression of the surface **s**, we can easily compute two tangent vectors at any point on this surface (Eq. 4.1).

$$\partial_\alpha \mathbf{s} = \sum_{i=1}^{n} \boldsymbol{\lambda}_i \phi_i' \circ d_i \cdot \partial_\alpha d_i \text{ with } \alpha = u,v \qquad (4.1)$$

The normal vector **n** at this point (e.g. Figure 4) is obtained by calculating the cross product of these two tangent vectors (Eq. 4.2).

$$\mathbf{n} = \partial_u \mathbf{s} \times \partial_v \mathbf{s} \,/\, \left\| \partial_u \mathbf{s} \times \partial_v \mathbf{s} \right\| \qquad (4.2)$$

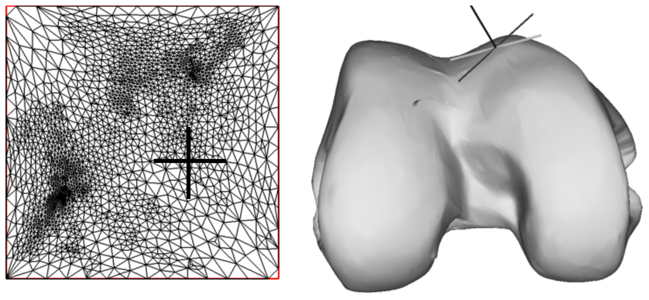

(a) 2 perpendicular vectors (b) 3D local coordinate frame

**Figure 4:** Normal computation based on tangent vectors.

If $\mathbf{s}^1$ and $\mathbf{s}^2$ are two parametric surfaces respectively associated with tibia and femur. The position of the contact point according to $\mathbf{s}^1$ is equal to its position according to $\mathbf{s}^2$. Moreover, the tangent spaces of $\mathbf{s}^1$ and $\mathbf{s}^2$ are aligned at this point (e.g. Figure 5). If $(u^i,v^i)$ are the 2D coordinates of the contact point according to $\mathbf{s}^i$ for $i$=1,2, the contact constraint is expressed as follows:

$$\begin{aligned} \mathbf{s}^1\left(u^1,v^1\right) - \mathbf{s}^2\left(u^2,v^2\right) &= \mathbf{0} \\ \partial_u \mathbf{s}^1\left(u^1,v^1\right) \cdot \mathbf{n}^2\left(u^2,v^2\right) &= 0 \\ \partial_v \mathbf{s}^1\left(u^1,v^1\right) \cdot \mathbf{n}^2\left(u^2,v^2\right) &= 0 \end{aligned} \qquad (4.3)$$

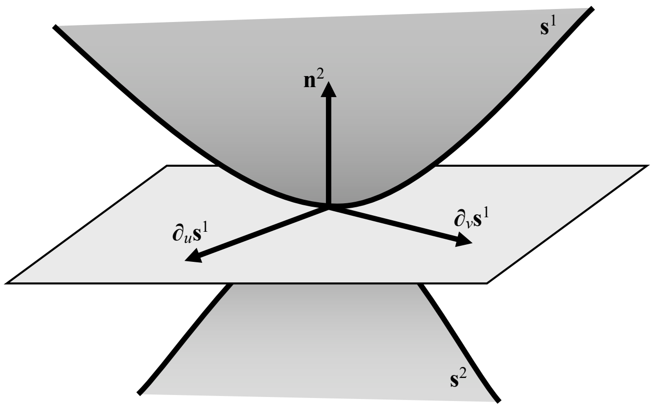


**Figure 5:** Contact point between two smooth surfaces $\mathbf{s}^1$ and $\mathbf{s}^2$.

Starting from a dynamic system composed of 16 DOF (12 initial DOF + 4 *free variables*), the constraints equations cancel 5 DOF (Eq. 4.3). So it only remains 5 DOF if we exclude the global configuration (6 DOF corresponding to position and orientation) of the system.

## 5. PERSPECTIVES

We came to propose this method to retrieve a smooth parametric surface because of our involvement in contact modeling and continuous tracking of contact points. But, out of this context, we consider that the ability to build a smooth parametric surface from a discrete triangular mesh can offer interesting results in other computer graphics applications. We mention two possible applications of our continuous framework.

### 5.1 Mesh refinement

Another advantage of this continuous framework is the ability to express any geometric quantity like the curvature field at any point of the surface. Even if we did not exploit this aspect of our results yet, a continuous formulation of the curvature field seems to be a powerful tool to solve the *mesh refinement* problem. In fact, Gaussian and mean curvature are used as criteria for refinement [23][24]. We briefly remind some essential formulas to calculate those curvatures.

$$\begin{aligned}
\partial_{\alpha\alpha}\mathbf{s} &= \sum_{i=1}^{n} \boldsymbol{\lambda}_i \left\{ \begin{matrix} \phi_i'' \circ d_i \cdot \left(\partial_\alpha d_i\right)^2 + \\ \phi_i' \circ d_i \cdot \partial_{\alpha\alpha} d_i \end{matrix} \right\} \text{ with } \alpha = u,v \\
\partial_{uv}\mathbf{s} &= \sum_{i=1}^{n} \boldsymbol{\lambda}_i \phi_i'' \circ d_i \cdot \partial_u d_i(u,v) \cdot \partial_v d_i(u,v)
\end{aligned} \tag{5.1}$$

Using (Eq. 5.1), we deduce the expression of first and second fundamental forms [25]:

$$\begin{matrix} e = \partial_u\mathbf{s}\cdot\partial_u\mathbf{s} & f = \partial_u\mathbf{s}\cdot\partial_v\mathbf{s} & g = \partial_v\mathbf{s}\cdot\partial_v\mathbf{s} \\ l = \mathbf{n}\cdot\partial_{uu}\mathbf{s} & m = \mathbf{n}\cdot\partial_{uv}\mathbf{s} & n = \mathbf{n}\cdot\partial_{vv}\mathbf{s} \end{matrix} \tag{5.2}$$

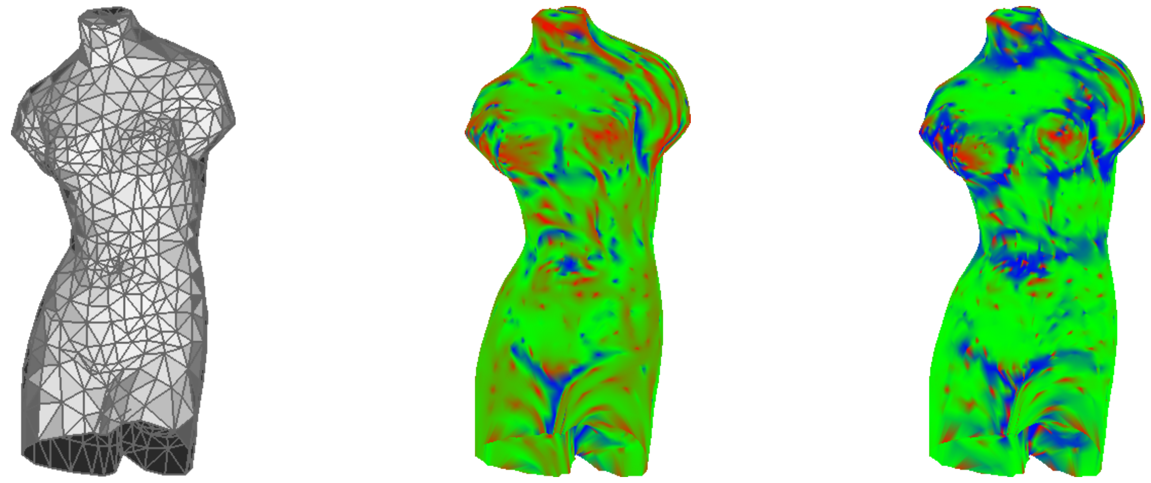
The Gaussian curvature $K$ and the mean curvature $H$ are expressed as follows:

$$K = \frac{ln - m^2}{eg - f^2} \quad H = \frac{ne - 2mf + lg}{2\left(eg - f^2\right)} \tag{5.3}$$

According to these formulas, we are able to retrieve the value of both curvatures at any point of the interpolating surface (e.g. Figure 6).

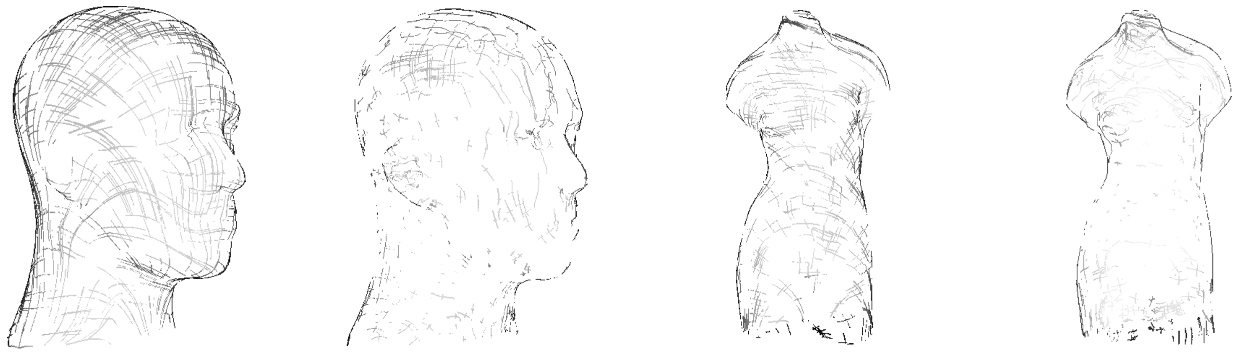
### 5.2 Non-photorealistic rendering

Elber proposes a method based on isoparametric lines drawing [26]. But this method is restricted to parametric surfaces like revolution surfaces. Another advantage of our method is that we are able to build a parametric surface and thus obtain isoparametric lines from a triangular mesh.



(a) Initial triangular mesh (a) Mean curvature (b) Gaussian curvature

**Figure 6:** .Continuous curvature fields (b) (c) extracted from a triangular mesh (a).



(a) 1000 lines (b) 600 lines (c) 1000 lines (d) 600 lines

**Figure 7:** Non-photorealistic rendering using isoparametric lines (a) (c) and principal curvature lines (b) (d).

Others techniques in NPR exploit the curvature field to draw lines following the principal curvature of the surface [27]. The quality of the resulting rendering depends on the geometric definition of the triangular mesh. With our method, since we have a continuous expression of an interpolating surface, we are able to draw high-quality lines of curvature from any low-polygon mesh.

The curvature $k$ of a parametric surface is defined as a function dependant of the rate of change $\lambda = dv/du$ for a displacement ($du$,$dv$) in the parametric domain (Eq. 5.4).

$$k(\lambda) = \frac{l + 2m\lambda + n\lambda^2}{e + 2f\lambda + g\lambda^2} \qquad (5.4)$$

The principal curvatures $k_1$ and $k_2$, that respectively represent the minimal and maximal curvatures, rely on the rates of change $\lambda_1$ and $\lambda_2$, solutions of (Eq. 5.5).

$$\det \begin{bmatrix} \lambda^2 & -\lambda & 1 \\ e & f & g \\ l & m & n \end{bmatrix} = 0 \qquad (5.5)$$

Before the rendering stage, we calculate random seed points in the parametric domain. From these seed points, we generate isoparametric lines (aligned with the $u$,$v$ axis) or curvature lines oriented in the principle curvatures directions (e.g. Figure 7).

## 6. CONCLUSION

In this paper, we presented a new method to retrieve a smooth parametric surface from a discrete triangular mesh. In the biomechanics area, this work is another step further towards a physically-based model of the knee joint. From discrete data coming from acquisition devices, we are now able to build a continuous geometric description of the surface of tibia and femur which is compatible with our simulation framework. The next important stage consists in integrating contact constraints based on *free variables* in our physic engine in order to calculate the evolution of our knee joint model during the flexion movement. First, we want to compare our future results with experimental values to validate our approach. Then, we will try to identify the actual degrees of freedom of this constrained system in order to manage fewer unknowns.

We also mentioned other possible applications of our method. In order to confirm our conviction that this new scheme is an interesting and original tool for mesh refinement, we would like to get further by comparing the classical discrete approach and the continuous one that we suggest. Our first steps in the non-photorealistic area are also encouraging. The first results based on isoparametric and principal curvature lines invite us to explore this field deeper.

## 7. REFERENCES

[1] Keeve E., Kikinis R. *Biomechanics-based simulation of knee dynamics*. Biomedical Engineering and Engineering in Medicine and Biology Society Conference 1, pp.558, 1999.

[2] Piazza S.J., Delp S.L. *Three-dimensional dynamic simulation of total knee replacement motion during a step-up task*. Journal of Biomechanical Engineering. 123, No 6, pp.599-606, 2001.

[3] McCarthy A.D., Wilkinson I.D., Hose D.R., Barber D.C., Wood S., Darwent G.D., Chan D., Bickerstaff D.R. *Musculo-Skeletal Simulation: Finite Element Meshes Derived From Magnetic Resonance Volumes*. Proceedings of the International Society for Magnetic Resonance in Medicine, 2002.

[4] Gardiner C., Weiss J.A. *Subject-Specific Finite Element Analysis of the Human Medial Collateral Ligament During Valgus Knee Loading*. Journal of Orthopaedic Research 21, pp.1098-1106, 2003.

[5] Barzel R., Hughes J.F., Wood D.N. *Plausible motion simulation for computer graphics animation*. Eurographics Workshop in Computer Animation and Simulation '96, pp.183-197, 1996.

[6] Remion Y., Nourrit J.M., Nocent O. *D-dimensional parametric models for dynamic animation of deformable objects*. The Visual Computer Journal 17, No 3, pp.167-178, 2001.

[7] Nocent O., Remion, Y. *Continuous deformation energy for Dynamic Material Splines subject to finite displacements*. Eurographics Workshop in Computer Animation and Simulation'01, pp.87-97, 2001.

[8] Abdel-Rahman E.M., Hefzy M.S. *Three-Dimensional Dynamic Behavior of the Human Knee Joint under Impact Loading*. Medical Engineering & Physics 20, No 4, pp.276-290, 1998.

[9] Cani-Gascuel M.P., Desbrun M. *Animation of Deformable Models Using Implicit Surfaces*. IEEE Transactions on Visualization and Computer Graphics, Vol. 3, No 1, pp.39-50, 1997.

[10] Kry P.G., Pai D.K. *Continuous contact simulation for smooth surfaces*. ACM Transactions on Graphics. 22, No 1, pp.106-129, 2003.

[11] Praun E., Hoppe H. *Spherical parametrization and remeshing*. ACM Transactions on Graphics 22, No 3, pp.340-349, 2003.

[12] Sheffer A., Hart J.C. *Seamster: Inconspicuous Low-Distortion Texture Seam Layout*. VIS '02, pp.291-298, 2002.

[13] Tutte W. *How to draw a graph*. Proceedings of London Mathematics Society 13, pp.743–768, 1963.

[14] Floater M.S. *Parametrization and smooth approximation of surface triangulations*. Computer Aided Geometric Design 14, No 3, pp.231-250, 1997.

[15] Floater M.S., Hormann K. *Surface parameterization: a tutorial and survey.* Multiresolution in Geometric Modelling, Springer, 2004.

[16] Carr J.C., Beatson R.K., Cherrie J.B., Mitchell T.J., Fright W.R., McCallum B.C., Evans T.R. *Reconstruction and representation of 3D objects with radial basis functions*. SIGGRAPH '01, pp.67-76, 2001

[17] Noh J.Y., Neumann U. *Talking Faces*. IEEE International Conference on Multimedia and Expo (II), pp.627-630, 2000.

[18] Ruprecht D., Muller H.. *Image Warping with Scattered Data Interpolation*. IEEE Computer Graphics and Applications 15, No 2, pp.37-43, 1995.

[19] Reuter P. *Reconstruction and Rendering of Implicit Surfaces from Large Unorganized Point Sets*. PhD thesis, Université Bordeaux 1, France, 2003.

[20] Ohtake Y., Belyaev A., Seidel H.P. *Multi-scale approach to 3D scattered data interpolation with compactly supported basis functions*. Shape Modeling International, pp.153-164, 2003.

[21] Remion Y. *Prise en compte de «contraintes à variables libres»*. Technical Report CReSTIC 03-02-01, pp.1-4, 2003.

[22] Lenoir J., Meseure P., Grisoni L., Chaillou C. *A Suture Model for Surgical Simulation*. 2nd International Symposium on Medical Simulation, pp.105-113, 2004

[23] Kim S.J., Kim C.H., Levin D. *Surface simplification using a discrete curvature norm*. Computers & Graphics. 26, No 5, pp.657-663, 2002.

[24] Dyn N., Hormann K., Kim S.J., Levin D. *Optimizing 3D Triangulations Using Discrete Curvature Analysis*. Mathematical Methods for Curves and Surfaces, pp.135-146. 2000.

[25] Farin G. *Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide*. Academic Press, 1996.

[26] Elber G. *Interactive Line Art Rendering of Freeform Surfaces*. Eurographics 99 18, No 3, pp.1-12, 1999.

[27] Rössl C., Kobbelt L., Seidel H.P. *Line art rendering of triangulated surfaces using discrete lines of curvature*. WSCG'2000, pp.168-175, 2000.