# Creating patient-specific Finite Element Models with a Simple Mesh Morpher

Michał Borysiak, Zuzanna Krawczyk, Jacek Starzyński
Warsaw University of Technology
00-662 Warszawa, ul. 75 Koszykowa, e-mail: jstar@iem.pw.edu.pl

The paper presents a simple 3D finite element mesh morpher aimed at creation of patient-specific models of human body parts. These models are to be used in realistic simulation of magneto- and electrotherapeutic treatment. The presented morpher uses simple algorithm of guided stretching which needs only a few measurements of patients body, but it may deform some finite elements. A public domain code Stellar is used to fix these problems.

## 1. Introduction

Patient specific models of bioengineering phenomena are nowadays becoming more and more popular [2,3,4]. Using them makes possible more precise investigation of bioelectromagnetic phenomena and planning of sophisticated modern therapies and evaluations. However, creation of realistic, precise finite element models of human body is usually based on the exhaustive input data. It is also a time consuming process, which cannot be yet fully automated.

In the classical approach to realistic model creation one starts with fine quality cross-section images of human body. These may be taken from a topographic scans or, in case of base, averaged models, from the available digital images datasets such as the Visible Human Project [4]. Based on such images, segmentation exhibiting the desired tissue distinction is created. Segmented images are then layered together to build a digital, voxelized model of the body. Further smoothing of the model may be necessary if the finite element mesh needs to be obtained.

The whole process takes several hours of work, even for relatively simple parts of body. The most difficult part – segmentation – is not yet fully automated, but extensive research in the picture segmentation and evaluation will probably solve these problems in the near future. However, the input data acquisition process will still remain difficult, costly, and time consuming.

## 2. The method

The authors would like to propose another approach, which should allow to create simplified yet quite realistic models with minor computational efforts and only a few measurements of the patient body. The presented work is a part of the larger project aimed on the creation of software which will be used by medical staff in planning of electrical and magnetic therapeutic treatment. For such applications we need a tool which will quickly create a patient-similar model of body parts.

In our approach we like to minimize the input data to the absolute minimum. Thus we start with an universal, average sized fine model created using the classical procedure with fine quality input data. This fine "standard" model is created only once. Then it is used as foundation of the individualized models which are obtained by transformation of the base one. Simple measurements of external dimensions of the patients body form set of input data, which should allow to morph (shrink and/or stretch) the base model, to fit it to the given patient. This attitude will surely produce a model only roughly compatible the concrete patient but still it will be usable for presenting electric or magnetic field of external stimulator in the model similar to the patient body.

## 3. The implementation

The 2D implementation of the proposed methodology was shown in [1]. Here we shall present the first 3D implementation.

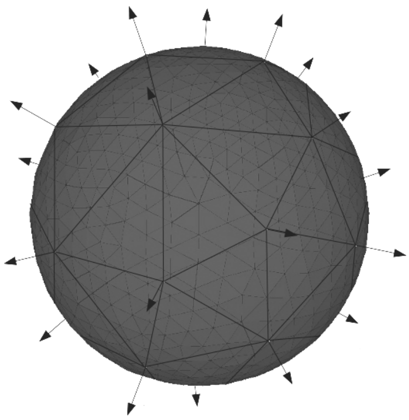

Fig. 1. Set of vectors $V$ with initial points on triangular surface mesh $T$

Let us assume that the base model to which we apply our method has a closed, connected external surface. Further, we assume that morphing is defined by a set of vectors $V$ determining displacements of several characteristic points lying on the model surface to the desired position on the destination surface and the invariance point $CP$ of the transformation. Initial points of vectors from $V$ can be connected to create a triangular surface mesh $T$ (Fig. 1). Each triangle of this mesh can be regarded as a face of a tetrahedron t with the opposite vertex coinciding with $CP$. The set of the tetrahedra defines a connected and comprehensive division $D$ of the space, where the division of the outer space is defined by extending the tetrahedron edges beyond the triangles from $T$. Each of the model mesh nodes belongs to one of the tetrahedron from $D$. The algorithm defines the displacement of a given node $n_i$ as a weighted sum of vectors (from V) corresponding to the vertices of the triangle determining the tetrahedron to which $n_i$ belongs (see Fig. 2).

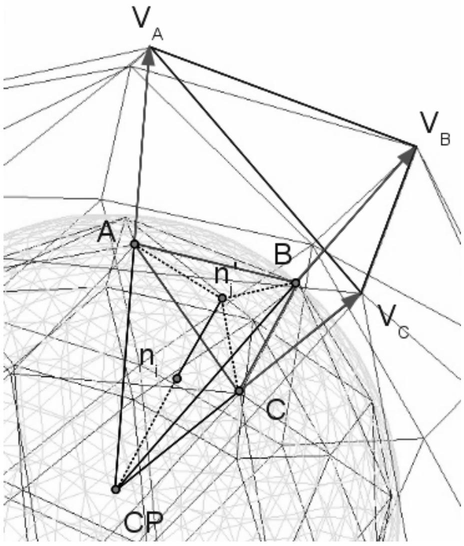


Fig. 2. Details of the morphing algorithm

More precisely, for each node $n_i$ of the base model:

1. we find a triangle $t \in T$ and a point $n'_i$ belonging to t such that $n'_i$ is a projection of $n_i$ along the the ray $CP$ - $n_i$,
2. we define the displacement of $n_i$ as a weighted sum of three vectors from V which have initial points in vertices A, B and C of the found triangle *t:*

$$v_i = a_A r_A V_A + a_B r_B V_B + a_C r_C V_C \tag{1}$$

where weights are calculated as follows ($P$ is the area of the triangle)

$$a_A = \frac{P_{\Delta BCn'_i}}{P_{\Delta ABC}}, \; a_B = \frac{P_{\Delta An'_iC}}{P_{\Delta ABC}}, \; a_C = \frac{P_{\Delta An'_iB}}{P_{\Delta ABC}} \tag{2}$$

$$r_A = \frac{|n_i - CP|}{|A - CP|}, \; r_B = \frac{|n_i - CP|}{|B - CP|}, \; r_C = \frac{|n_i - CP|}{|C - CP|}$$

Practical implementation of the above algorithm requires a prior definition of the displacement vectors $V$ and the invariance point $CP$. For the parts of the body the shape of which is close to spherical, the choice of $CP$ is straightforward: it can be the center of the sphere or any point close to it, like the center of mass. An excellent example of a solid of such characteristic is a model of human head. However, due to the complicated structure of brain, internal structure of a real patient's head can differ significantly from its morphed model, even if they both have similar external shape[2]. Thus, for the first implementation of the algorithm, body parts with simpler internal structure are more suitable. There are many examples of such parts, for instance arm, forearm or leg. They are however elongated and their symmetry is rather cylindrical, not spherical. In case of such elongated shapes we chose the whole axis for model mesh, which remain invariant during the transformation.

The morphing algorithm can be easily adapted to cope with this situation. After selecting the axis we adjust the length of the model mesh to the measured length of patient's body part. It is done by simple scaling the model along its axis. The set $V$ of the displacement vectors is generated from the measured points on the patient's body. These points are the end points of the displacement vectors. The initial points are obtained as the projections of the endpoints on the model surface, perpendicular to its axis. Further, for any node $n_i$ of the original model mesh, we project it on the model axis and regard the projection as a node specific invariance point $CP_i$. Then, we can apply the formulas given by Eq. 1 and 2 without further modifications.

We rely on the input data that is taken in two perpendicular directions in a few planes along the model axis. Only the outer dimensions are measured. To save the measurement time, the number of the points taken is rather limited and not sufficient to precisely define the morphing. Additional points are generated by assuming that each cross section in the plane perpendicular to the axis has an elliptical shape with the diameters determined by the measured points. Still further points can be obtained along the axis as a result of spline interpolation between the points from subsequent cross sections. The simplicity of input data is stressed by the graphical user interface which is shown in Fig. 2.

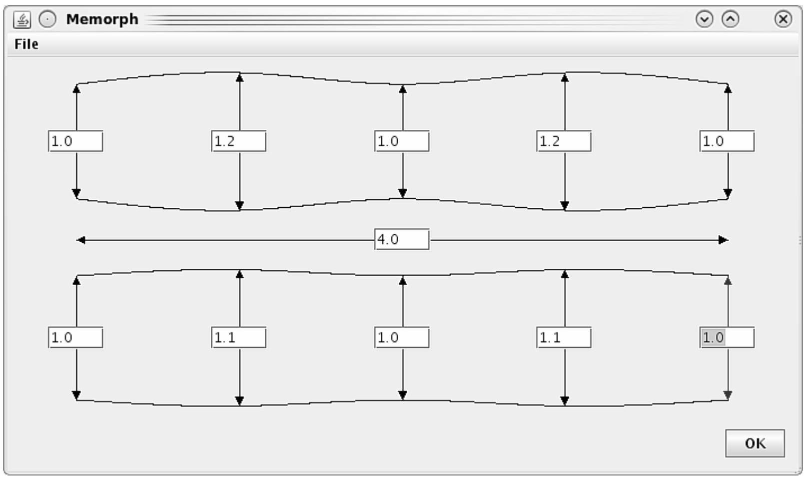


Fig. 3. The essence of the graphical user interface: to setup the morpher one has to measure thickness of an elongated body part into perpendicular dimensions at five planes along the part axis. Total length of the body part is measured separately

For the sake of efficiency the algorithm was implemented entirely in C++ programming language. Only the ISO standard library was used which ensures program portability. The program can read and write a simple text grid format (Diffpack library standard [7]).

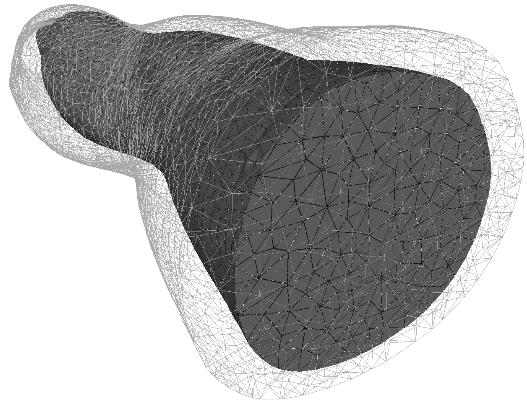

Fig. 4. The initial model (before applying morphing – the external mesh) and the model after morphing (internal mesh - shaded)

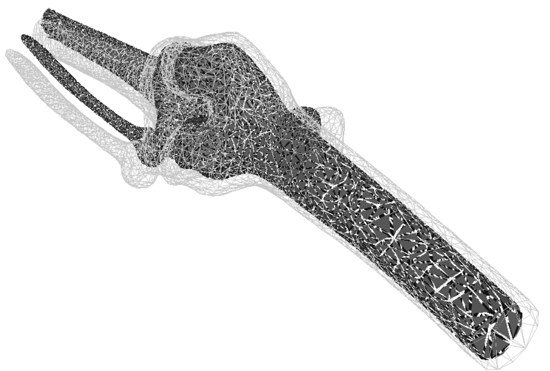

Fig. 5. A part of the mesh representing bones – before applying morphing (external mesh) and after the transformation (shaded internal mesh)

We have applied our method to an exemplary mesh in a Diffpack format representing model of a thigh and shank. The mesh has been build on the basis of scans taken as a part of The Visible Human Project [5]. It consists of two domains representing the bones and the soft tissues. The total number of tetrahedra in the mesh is 1.081.053 while the number of vertices is 186.849. The morphing transformation applied thinned the mesh by 26% along the Y axis and 18% along the Z axis.

As we explain in the next section, the quality of a mesh with respect to the FEM can be estimated by measuring the range of the dihedral angles of the mesh tetrahedra. The applied morphing transformation, as always, has worsen the quality of our model mesh. For the original model, the dihedral angles span a range from 3,116° to 174,6°. After the transformation it changes to: 0,0155° – 179, 97° which means that the resulting mesh contains almost planar, degenerated tetrahedra.

One can wonder how the morphing transformation, based on the measurements of the outer dimensions of the model, deforms its inner structure represented in our case with the subdomain mesh of the bones. The result is encouraging: the bones get thinned as desired. They change they shape but not significantly and one can easily recognize their characteristics shapes and arrangement.

## 4. Improvement of the mesh quality

According to the morphing algorithm – only vertices are moved, and it is not taken into consideration how their displacement affects connections between vertices (tetrahedron edges and faces). Mesh morphing may cause the deterioration of the quality of some tetrahedral mesh elements which in turn may limit their usability for the Finite Element Method analysis. Bad quality elements can be defined as those tetrahedrons the shape of which significantly differs from regular tetrahedra. (Large dihedral angles have negative effect on interpolation error, too small angles cause bad stiffness matrix conditioning). We use a public domain code Stellar [6] in order to improve the quality of morphed meshes. According to the authors of Stellar, the quality of the whole mesh depends not on the average quality of its elements but on the quality of the worst element. Therefore, Stellar concentrates on improving the quality of the worst elements.

Stellar is highly configurable application. The program implements a wide choice of different mesh improvement operations, such as vertex smoothing, different topological operations including vertex insertion. Smoothing operations are moving vertices, but they do not change connections between them, topologiacal operations interfere with internal mesh structure, they may change number of verticies or faces in the mesh. According to [6], the most effective way to improve the mesh consists of applying to mesh all of the above mentioned operations, however, in principle, any set of proposed operations can be chosen. Stellar introduces four quality measures, which can be applied to mesh improvement. Quality measure t is defined as a strictly increasing function q(t), with its maximal value 1 corresponding to a regular tetrahedron. Those measures are: minimum sine – minimum sine of each of six dihedral angles of tetrahedron; biased minimum sine – sines of obtuse angles in tetrahedron are multiplied by a given coefficient, then the minimum sine is chosen; radius ratio - radius of inscribed sphere of the tetrahedron divided by the radius of circumscribed sphere of this tetrahedron, normalized in such a way that the maximum measure value equals 1;

volume-length ratio – volume of the tetrahedron divided by the square root of the sum of squares of tetrahedron edge lengths, with the denominator cubed. The measure is multiplied by such a coefficient that the maximum measure value equals 1.

In order to test if the way how Stellar improves meshes is useful for our application, it was applied to a number of different meshes of simple geometrical shapes. *Inter alia*, it has been checked how the program can cope with ellipsoids with various mesh densities and to what extent it will improve meshes which span the same shape but have different quality parameters.

In a similar way as during the morphing, meshes with poor quality were created from the good quality meshes by scaling them along one of the axes with an arbitrary factor. This straightforward procedure allowed us to obtain meshes with small and large dihedral angels.

The results have shown, that the level of improvement of meshes with good quality parameters and those of "bad" meshes is similar. Obviously, improved meshes with the initial bad quality still have worse final quality than improved meshes which have had better quality before improvement. Improvement of meshes with low quality takes Stellar more time than improvement of meshes with better quality.

It turned out that two of Stellar quality measures are more efficient than the other: biased minimum sine and volume-length ratio. Satisfactory results were also achieved by the combination of two quality measures when during the first iteration the mesh was improved with the minimum sine quality measure while during the second iteration the volume-length-ratio measure was applied.

The biased minimum sine measure does not recognize as bad ones the so called spire tetrahedrons (very long, high tetrahedrons) because value of this measure depends only on dihedral angles of the tetrahedron. This is not the case for the volume-length ratio measure. Spire tetrahedrons do not worsen discretization error or stiffness matrix conditioning, although they may cause problems because of precision of calculations in MES which is inversely proportional to the length of the longest edge of the tetrahedron.

A simple example of the application of Stellar to the ellipsoid shape has been depicted in Figure 6.

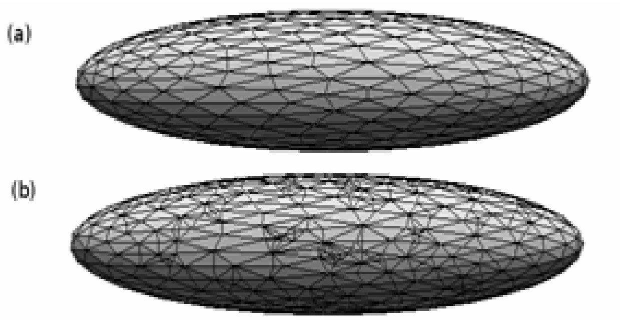


Fig. 6. Ellipsoid mesh before (a) and after (b) improving it with Stellar: the smallest dihedral angle was improved from 9.6° to 27,4° while the largest one changed from 164,7° to 142,63°

Stellar can be used to improve tetrahedral models of parts of human body, although not without problems in some areas. The mesh improvement schedule in the program is designed to achieve the quality of tetrahedron that is as good as possible. Thus, the duration of the improving operation is not the most important factor. Improvement of large meshes can take a considerable long time. Moreover, Stellar does not allow to split a mesh and to form subdomain meshes which is desired in modelling parts of human body. Despite these inconveniences Stellar can be applied to improving Patient-Specific Finite Element Model.

## 5. Conclusion

The simple mesh morphing algorithm combined with the mesh quality improvement program allows to obtain realistic and individually shaped body models with minimal input data.

## References

[1] R. Szmurło, J. Starzyński, Specimen-specific finite element models of human head obtained with mesh-morphing, Przeglad Elektrotechniczny (Electrical Review), R. 85 NR 4/2009, pp. 47-49.

[2] M. Tada, H. Yoshida, M. Mochimaru, Geometric Modeling of Living Tissue for Subject-Specific Finite Element Analysis, Engineering in Medicine and Biology Society, 28th Annual International Conference of the IEEE, 2006, pp. 6639-6642.

[3] I. A. Sigal, M. R. Hardisty, C. M. Whyne, Mesh-morphing algorithms for specimen-specific finite element modeling, J Biomech. 2008, pp. 1381-1389.

[4] N. Inou, M. Koseki, M. Jonishi, K. Maki, Patient-Specific Finite Element Modeling of Human Skull Based on X-ray CRT Images, Proceedings of 1st AOTULE Studens Workshop and 5th KAIST-Tokyo Mechanical Engineering Workshop, Daejeon, Korea, 2007, p. 17.

[5] Visible Human Project: http://www.nlm.nih.gov/research/visible.

[6] Aggressive Tetrahedral Mesh Improvement Bryan Matthew Klingner and Jonathan Richard Shewchuk, Proceedings of the 16th International Meshing Roundtable (Seattle, Washington), pages 3–23, October 2007.

[7] H. P. Langtangen. Computational Partial Differential Equations - Numerical Methods and Diffpack Programming. Texts in Computational Science and Engineering, vol 1. Springer, and edition, 2003.

ACKNOWLEDGEMENT

Research described in this was partially suported by Polish Ministry of Education grant no. N N510 148838.