HAL
open science

# Jacobian-based repair method for finite element meshes after registration

Marek Bucki, Claudio Lobos, Yohan Payan, Nancy Hitschfeld

## To cite this version:

Marek Bucki, Claudio Lobos, Yohan Payan, Nancy Hitschfeld. Jacobian-based repair method for finite element meshes after registration. Engineering with Computers, 2011, 27, pp.285-297. 10.1007/s00366-010-0198-2 . hal-00602162

**HAL Id: hal-00602162**

**https://hal.science/hal-00602162v1**

Submitted on 21 Jun 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Engineering With Computers manuscript No.**
(will be inserted by the editor)

# Jacobian–based Repair Method for Finite Element Meshes after Registration.

**Marek Bucki · Claudio Lobos · Yohan Payan · Nancy Hitschfeld**

Received: date / Accepted: date

**Abstract** Registration methods are used in the meshing field to "adapt" a given mesh to a target domain. Finite Element Method (FEM) is applied to the resulting mesh to compute an approximate solution to the system of Partial Differential Equations (PDE) representing the physical phenomena under study.

Prior to FE analysis the Jacobian matrix determinant must be checked for all mesh elements. The value of this Jacobian depends on the configuration of the element nodes. If it is negative for a given node, the element is invalid and therefore the FE analysis cannot be carried out. Similarly, some elements, although valid, can present poor quality regarding Jacobian–based indicator values, such as the Jacobian Ratio. Mesh registration procedures are likely to produce invalid and/or poor quality elements if the Jacobian parameter is ignored.

M. Bucki
TIMC-IMAG UMR CNRS 5525, Pavillon Taillefer, Faculté de Médecine, Domaine de la Merci, 38706, LA TRONCHE CEDEX, France
Tel.: +33-4-56520005
Fax: +33-4-56520055
E-mail: marek.bucki@imag.fr

C. Lobos
Departamento de Informática, Universidad Técnica Federico Santa María, Av. Vicuña Mackenna 3939, Zip: 8940897, San Joaquín, Santiago, Chile
Tel.: +56-2-432 66 56
E-mail: clobos@inf.utfsm.cl

Y. Payan
TIMC-IMAG UMR CNRS 5525, Pavillon Taillefer, Faculté de Médecine, Domaine de la Merci, 38706, LA TRONCHE CEDEX, France
Tel.: +33-4-56520001
Fax: +33-4-56520055
E-mail: yohan.payan@imag.fr

N. Hitschfeld
Departamento de Ciencias de la Computación, Universidad de Chile, Blanco Encalada 2120, 4to piso, Zip: 837-0459, Santiago, Chile
Tel.: +56-2-9784876
Fax: +56-2-6895531
E-mail: nancy@dcc.uchile.cl

To repair invalid and poor quality elements after mesh registration, we propose a relaxation procedure driven by specific validity and quality energy formulations derived from the Jacobian value. The algorithm first recovers mesh validity and further improves elements quality, focusing primarily on nodes that make the elements invalid or of poor quality.

Our novel approach has been developed in the context of non–rigid mesh registration and validated on a data set of 60 clinical cases in the context of orthopaedic and orthognathic hard and soft tissues modelling studies. The proposed repair method achieves a valid state of the mesh and also raises the quality of the elements to a level suitable for commercial FE solvers.

**Keywords** Meshing · Mesh Repair · Registration · Finite Element Method · Quality Improvement.

## 1 Introduction

The Finite Element Method (FEM) [30,12] is a numerical technique computing an approximate solution to a set of Partial Differential Equations (PDEs) and therefore, it is widely used to simulate mechanical behaviors.

The FEM relies on a subdivision of the continuum domain $\Omega$ to be simulated. This subdivision, called **mesh**, is the partition of $\Omega$ into simpler geometrical bodies known as **elements**. The FEM provides an approximation of the solution to the system of PDEs at each node of the mesh. This approximate solution can be further computed at any point $\boldsymbol{p} \in \Omega$ by interpolation of the values found at the nodes of the element containing $\boldsymbol{p}$.

The context of our work is biomechanical modeling of human organs for biomedical applications. In this field, the most common way to produce a mesh is to perform the following steps:

- To obtain volumetric medical images of the patient's organ using Magnetic Resonance Imaging (MRI), Ultrasound (US), Computed Tomography scanner (CT) or other imaging techniques.
- To achieve image segmentation in order to produce a cloud of points or an initial surface mesh defining the geometry of the modelled domain.
- To add internal nodes to produce the 3D elements that will conform the final volumetric mesh.

Some examples of such mesh generators are [21,11,1] for tetrahedral meshes. Others like [29,24] are interested in hexahedral meshes and finally other works such as [18,17] focus on the production of mixed–element meshes.

Even though those techniques are suited for most of the cases, in the medical field it is sometimes preferred to work with "registration" techniques. Consider for instance the concept of image registration which is the process of transforming different sets of data into one coordinate system. In the case of mesh registration, the different sets of data come from a generic model, named the Atlas, and patient data. The Atlas is normally a predefined volume mesh conserving all the relevant data for the simulation. The patient data is normally a set of points or a surface mesh. The first step is to produce rigid registration, which is, as in the case of imaging, putting both in the same coordinate system. The second step is to produce elastic registration, which is to

deform the Atlas in order to represent patient data. The overall process is then called "mesh registration".

There exist several mesh registration techniques, but to continue with the explanation, the so called **Mesh-Matching** (M–M) algorithm introduced by Couteau et al. in [6] will be used. The statement is that building a mesh of an organ is a complex task. To achieve accurate surface representation and good quality while limiting the number of elements in the mesh is somehow, a difficult compromise. Therefore in many cases, meshes are built by hand. As this task is time consuming and usually requires expert user interaction, it cannot be easily put in practice for a great number of patients. Therefore an interesting alternative is to use a pre–defined generic FE mesh (the *Atlas* mesh) of the target organ in order to achieve accurate representation of its structure.

FE mesh construction relies on patient data obtained from segmented medical images. The resulting data can either be a cloud of points or a surface mesh of the organ. Using this representation a registration process *fits* the surface nodes of the Atlas onto the target organ by applying a transformation function $T$. In other words, an elastic registration function is an application $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that minimizes a given registration error, or better said 'registration energy', between a given *source* points set $S$ and a *destination* points set $D$. The registration energy is a 'similarity' measure between the transformed source points set $T(S)$ and the target destination points set $D$. If the registration energy and function search space are not carefully defined, finding $T$ is an ill-posed problem, i.e. the existence and uniqueness of the solution are not guaranteed.

The registration energy can be freely defined in order to reflect the nature of the registration problem. The optimal registration is achieved when this energy reaches zero i.e. when each transformed source point has found a counterpart within the destination set. Given the sparse and uneven nature of the data, the search for the elastic transformation must be constrained in order to avoid erratic registrations. Various approaches to produce a registration have been proposed in the literature [6, 7, 27, 22], the discussion of which falls out of the scope of this article.

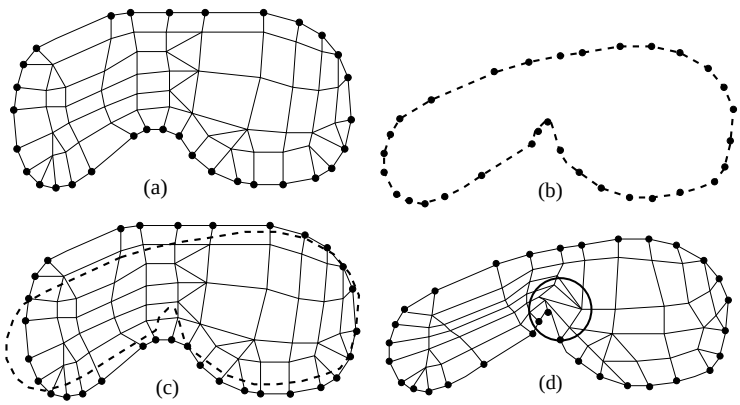


**Fig. 1** The mesh-matching algorithm: (a) The *Atlas* mesh, (b) the target domain or patient data, (c) the *Atlas* and the target surface domain represented by a dashed line and (d) the final mesh of the target organ obtained by applying the transformation function $T$ onto the entire *Atlas* mesh. The circled region shows the location of possible distorted element leading to mesh invalidity or poor quality.

Note that $T$ is computed using surface nodes $S$ (of Atlas volume mesh) and the points set $D$ (patient data). Moreover, $T$ is a 3D transformation function so it can be applied to Atlas internal nodes and so a final 3D mesh of the target organ is produced (patient mesh). Unfortunately it has been reported [19] that, in some cases, the application of $T$ to internal nodes can produce invalid or poor quality elements (to be defined in §2). Fig. 1 shows an example of M–M (here illustrated in 2D for clarity) and suggests how invalid and poor quality elements can be created.

Note that invalid and poor quality elements cannot only be produced by registration methods [5,2,23], but also by meshing techniques such as the whisker-weaving from CUBIT[1] [16].

The goal of the present work is to repair the invalid and poor quality elements that might result from the process of mesh registration. The proposed method is based on the computation of the determinant of the Jacobian matrix and to this end section 2 defines what are invalid and a poor quality elements. Section 3 gives the State of the Art regarding mesh repair and quality improvement in our scope. Section 4 explains in further detail our proposed solution, in particular the Jacobian formulation, the penalty functions that are used, our algorithm and a practical example. Finally sections 5 and 6 provide our conclusions and perspectives.

## 2 Definition of invalid and poor quality elements

This section introduces the concepts of invalid and poor quality elements. It is important to mention that the two definitions proposed here rest upon the Jacobian matrix of the mapping function $F$ relating the parent or reference element frame to its actual coordinates within the mesh. In the remainder of the article this matrix is denoted as $J$. Note that other geometrical functions can be used to define the quality of an element [28,8,17,18].

### 2.1 Invalid element

As mentioned before, the FEM computes for each node of the underlying mesh an approximated solution to the system of PDEs describing the physical phenomenon under study. For this purpose, a perfect element is defined in a reference parent system $(\xi_1, \xi_2, \xi_3)$ where each reference element point is related to its actual counterpart within the element in the modeled domain referential $(x_1, x_2, x_3)$ through the mapping $F$, as shown in Fig. 2. The Jacobian matrix of mapping $F$ considered at parent frame point $\boldsymbol{\xi}$ is defined as:

$$J(\boldsymbol{\xi}) := \frac{\partial F}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi})$$

The FE analysis can be carried out on a mesh as long as the mapping $F$, considered for each mesh element, remains a one–to–one application. This property can be assessed by considering the determinant of Jacobian matrix of $F$, $|J(\boldsymbol{\xi})|$, at a number of element "control points", such as the element nodes or Gaussian quadrature (integration) points, for all the elements forming the mesh. If a Jacobian is found to be negative,

[1] See http://cubit.sandia.gov for details.

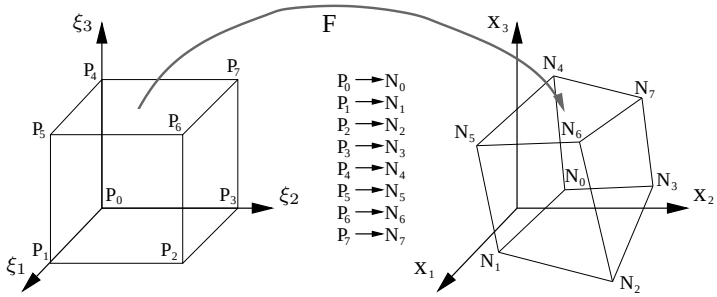


**Fig. 2** The reference–to–actual frame mapping from which the Jacobian matrix is derived. Left: the reference/parent space and right: the normal/actual mesh space.

then the mapping $F$ for the considered element is not a bijection, and the element cannot be used for FE analysis. In this case, the mesh is not not suitable for Finite Element analysis as the singularity of the mapping $F$ leads to modeling inconsistency.

Let's consider this situation from a geometrical point of view as illustrated in Fig. 3. If nodes $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$ are connected to node $\boldsymbol{a}$ then an imaginary plane can be formed by nodes $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$. The normal $\hat{\boldsymbol{n}}$ of this plane is computed as $(\boldsymbol{c}-\boldsymbol{b}) \times (\boldsymbol{d}-\boldsymbol{b})$. After normalizing the vector $\hat{\boldsymbol{n}}$ the signed distance of $\boldsymbol{a}$ to the plane can be computed as: $(\boldsymbol{a}-\boldsymbol{b}) \cdot \hat{\boldsymbol{n}}$. If this distance is positive, then the position of $\boldsymbol{a}$ is valid within this element. If not, then this element is considered invalid for the FEM at node $\boldsymbol{a}$.

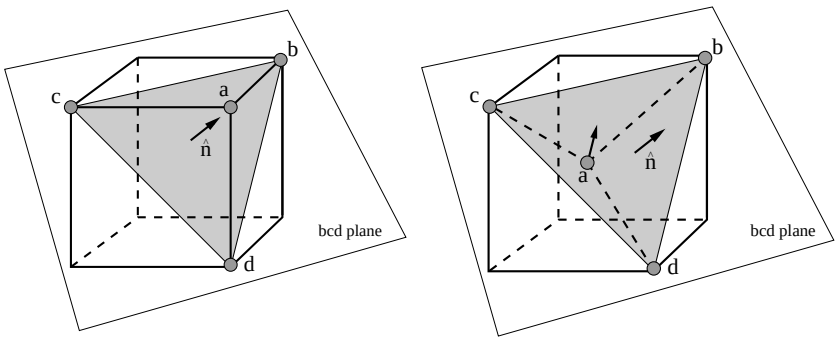


**Fig. 3** If the signed distance of node $\boldsymbol{a}$ to the plane formed by "connected nodes" $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$ is positive, the element is valid (left), in any other case the entire element is said to be invalid (right).

Note that the value of $|J(\boldsymbol{\xi})|$ is computed for each element at each of its nodes. If node $\boldsymbol{a}$ is shared by two elements $e_1$ and $e_2$, it is possible to devise a nodal configuration where the position of $\boldsymbol{a}$ is valid within $e_1$ but invalid within $e_2$.

### 2.2 Poor quality element

The Jacobian value measures the distortion of the actual mesh element w.r.t. its reference configuration, but does not carry overall distortion information. To overcome this issue the quality of an element can be measured at each node $\boldsymbol{n}$ as a ratio between the nodal Jacobian value at $\boldsymbol{n}$, and a global distortion information. The latter is given by the maximal Jacobian value among those measured at all element nodes. This element node quality criterion is called Jacobian Ratio or JR [17].

Mathematically, the JR of node $\boldsymbol{\xi}_0$ taken within element $e$, $\mathrm{JR}_e^{\boldsymbol{\xi}_0}$, is computed as:

$$\mathrm{JR}_e^{\boldsymbol{\xi}_0} := |J(\boldsymbol{\xi}_0)| \ / \ max_{\boldsymbol{\xi} \in e}\{|J(\boldsymbol{\xi})|\}$$

Considering the case presented in Fig. 3, the value of $|J(\boldsymbol{a})|$, computed at node $\boldsymbol{a}$ within the element, would be very small if $\boldsymbol{a}$ was close to plane $\boldsymbol{bcd}$ and therefore, so would be the value of the $\mathrm{JR}^{\boldsymbol{a}}$. Actually, when $\mathrm{JR}^{\boldsymbol{a}}$ value is positive, but close to 0 then the element presents a poor quality at node $\boldsymbol{a}$. When JR increases, so does the quality of the element until it reaches the ideal value of 1.

As it is impossible to converge to 1 all the JR values within the mesh, a quality acceptance threshold must be defined. To this end, we have chosen to follow the recommendations adopted in the commercial FE analysis software ANSYS (ANSYS Inc., USA), where an element $e$ is said to present poor quality when any node $\boldsymbol{\xi}$ within $e$ has a JR value lower than $0.0\overline{3}$ [13]. A good quality element, thus, has all its JR values within the interval $[0.0\overline{3}, 1]$.

## 3 Overview of mesh repair techniques proposed in the literature

In this work, our goal is to achieve validity and improve the quality of the elements in a mesh after applying a non–rigid registration in the context of mesh generation procedures such as the Mesh–Matching introduced in section 1. To do so, we focus specifically on the Jacobian matrix as it is at the root of many popular mesh rejection criteria implemented in commercial FE analysis softwares.

In [9], Freitag proposed the use of Laplacian–based algorithms to improve the overall quality of meshes. In particular, the Smart–Laplacian algorithm proposed in [9] changes the position of a given node only if it results in overall quality improvement. This filter is implemented in such a way that no invalid element can be generated within the mesh. Following the same idea, Freitag and Plassmann proposed in [10] the combination of Smart–Laplacian with optimization algorithms, having a much higher computational cost than the Laplacian filter alone, in order to achieve mesh untangling. In this approach, invalid mesh configurations are eliminated by mesh nodes reallocation and the applied optimization maximizes the minimum volume of tetrahedra in a local sub–mesh.

Both techniques presented above are designed to repair tetrahedral meshes, and the idea of expanding this concept to hexahedra and wedges was introduced by Knupp in [16, 15]. The main principle here is that the reallocation of a given node will only change the Jacobian values of the nodes connected to it (as showed in subsection 2.1). Moreover, for the tetrahedron, wedge and hexahedron each node is connected to 3 nodes within each element it is part of, therefore for those elements, a Jacobian–based repair method can always be geometrically reduced to an optimization based on tetrahedra.

Paoletti in [25] expanded the work to polyhedral meshes by including the "interpolation tensor" to the optimization.

Another important work was carried out by Kwok and Chen in [17] where the focus was on the quality of hexahedra and wedges. The Jacobian ratio was used to improve the quality in conjunction with other quality functions such as the aspect–ratio, the warping factor and the control number [14].

The main issue with the above mesh untangling techniques is that they are not focused on the strategy of "mesh registration". Lets recall that this type of strategy intend to match a predefined mesh model to a target domain. If the "registration" is accurate, the goal is to displace as less as possible the nodes in order to achieve mesh validity and acceptable quality levels. The above techniques are focused on overall element quality and therefore they have no constraint on reallocating every node, which in our case may cause an accuracy loss in the representation of target domain. Moreover, they do not separate validity recovery and quality improvement procedures, intending to achieve it all at the same time. The problem with this approach is that if invalid node $a$ and poor quality node $b$ are connected, finding a new position to improve the quality of $b$ might decrease the chances of finding a new valid position for $a$.

A completely different approach was proposed by Luboz et al. in [20]. This work was considered as a continuation of the Mesh Matching (M–M) technique, initially presented in [6], as it repairs invalid configurations after registration. First of all, it relies on the determinant of the Jacobian matrix of the mapping between parent and actual element configuration $F$, to detect the nodes that make the elements invalid. Let $\boldsymbol{n}$ be one of these nodes and $\{e_i\}_{i=1,\ldots,k}$ the set of elements for which $\boldsymbol{n}$ is invalid. The direction of displacement of $\boldsymbol{n}$ is deduced from the gradient of $F$ computed from each $e_i$. In other words, the direction of displacement of $\boldsymbol{n}$ is the one that maximizes the sum of the invalid Jacobians associated to node $\boldsymbol{n}$. A "displacement step" is also computed for each node and finally the overall set of invalid nodes is iteratively shifted until the mesh is repaired.

Although the approach of Luboz was successfully used to achieve a valid state of several registered meshes, it didn't consider the JR to improve the quality of the elements. Another important remark is that Luboz computed the direction of displacement of invalid nodes $\boldsymbol{n}$ by considering only elements at which $\boldsymbol{n}$ was invalid, and it should be noted here that if two elements $e_1$ and $e_2$ share node $\boldsymbol{n}$ and $e_1$ is valid but $e_2$ is not, the displacement of $\boldsymbol{n}$ to improve $e_2$ might cause invalidity at $e_1$.

In this article, we propose a new mesh repair technique that only considers the displacement of nodes that make the mesh invalid or of poor quality. To compute the new position of a given node, all the elements linked to it are taken into account. Moreover, cases where invalid configurations arise from the computation of new nodal positions are also managed. Once mesh validity is achieved, the overall quality is assessed and if necessary, our algorithm reallocates mesh nodes exhibiting poor quality. Here, the chosen quality indicator is the JR [17], for which we have chosen a threshold value $\mathrm{JR}_{min} = 0.0\overline{3}$, as mentioned earlier in subsection 2.2. The details of our mesh repair technique are given in the next section.

## 4 The proposed mesh repair procedure

The mesh repair procedure proposed in this article is a twofold sequential process:

- first all the elements in the mesh are inspected and, if necessary, their nodes positions are adapted so as to recover **validity**,
- then a second procedure is carried out on the mesh if the achieved **quality** levels are unacceptable.

The corrections applied to the nodal positions are based on the validity and quality criteria, both derived from the Jacobian of the mapping between the parent and actual element frame, as described above. Section 4.1 gives details on the evaluation of the Jacobian for a given element.

Usually distorted elements are sparsely distributed within the mesh, therefore both validity and quality optimizations can be carried out on "repair regions" defined as connected subsets of the mesh encapsulating distorted elements where displacements may only be applied to "improper nodes", i.e. nodes being either invalid or presenting poor quality, leaving other regions unaffected. This local repair strategy significantly decreases computational complexity as the number of degrees of freedom to be considered in a region is usually small. The local relaxations strategies for both validity recovery and quality improvement are described in section 4.2.

The above mentioned relaxations are carried out as maximizations of validity and quality "energy" terms. The formulation of each energy term ensures that the mesh nodal configuration reached at the end of the maximization procedure meets the desired validity or quality standards. The formulations of both energy terms are discussed in section 4.3.

The complete implementation of the iterative relaxation algorithm, common to both validity and quality optimization phases, is given in section 4.4.

Finally, section 4.5 gives a complete registration and repair example from which a particular repaired region is explained in detail.

## 4.1 Jacobian computation

### *4.1.1 Parent-to-actual element mapping*

Mesh relaxation relies on the iterative computation of the Jacobian matrix of the mapping between the element parent (or reference) coordinates system $(\xi_1, \xi_2, \xi_3)$ and the actual element coordinates $(x_1, x_2, x_3)$: $F : \mathbb{R}^3 \to \mathbb{R}^3, \boldsymbol{\xi} \mapsto \boldsymbol{x}$ (see Fig. 2). The Jacobian matrix is computed at a number of "control points" which can be either mesh nodes or element integration points (Gaussian quadrature points).

For a given element type and mesh configuration the Jacobian matrix $\partial F/\partial \boldsymbol{\xi}$ can be assembled and the exact value of its determinant quickly evaluated using pre–computed element shape function derivatives w.r.t. parent coordinates, as described below.

Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$ be an element point in parent coordinates and $\boldsymbol{x} = (x_1, x_2, x_3)$ the same point in actual element coordinates. For an isoparametric element the mapping $F$ can be expressed as a combination of the shape functions $\phi^i : \mathbb{R}^3 \to \mathbb{R}$ used for value interpolation of the PDE solution within the element:

$$\boldsymbol{x}(\boldsymbol{\xi}) := F(\boldsymbol{\xi}) = \sum_{n=1}^{N} \phi^n(\boldsymbol{\xi})\, \boldsymbol{x}^n \tag{1}$$

where $N$ is the number of nodes of the considered element and $\boldsymbol{x}^n = (x_1^n, x_2^n, x_3^n)$ are the coordinates of the $n^{\text{th}}$ element node.

The Jacobian matrix $J$ is defined at parent frame point $\boldsymbol{\xi}$ as:

$$J(\boldsymbol{\xi}) := \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}) = \begin{pmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{pmatrix}$$

where $J_{ij} = (\partial x_i / \partial \xi_j)(\boldsymbol{\xi})$.

The Jacobian matrix elements can be further rewritten using Eq. 1, as:

$$J_{ij} = \frac{\partial}{\partial \xi_j} \sum_{n=1}^{N} \phi^n(\boldsymbol{\xi})\, x_i^n = \sum_{n=1}^{N} \frac{\partial \phi^n}{\partial \xi_j}(\boldsymbol{\xi})\, x_i^n = \sum_{n=1}^{N} P_j^n(\boldsymbol{\xi})\, x_i^n$$

In the above expression all $P_j^n(\boldsymbol{\xi}) := (\partial \phi^n / \partial \xi_j)(\boldsymbol{\xi})$ are independent of the current element configuration $\{\boldsymbol{x}^n\}_{n=1,\ldots,N}$ and can be evaluated beforehand for all points of interest $\boldsymbol{\xi}$.

Finally, the Jacobian matrix of mapping $F$ computed at point $\boldsymbol{\xi}$ is assembled using the $3 \cdot N$ pre–computed shape function derivatives $P_j^n$ as follows:

$$J(\boldsymbol{\xi}) = \sum_{n=1}^{N} \begin{pmatrix} x_1^n \\ x_2^n \\ x_3^n \end{pmatrix} \begin{pmatrix} P_1^n(\boldsymbol{\xi}) & P_2^n(\boldsymbol{\xi}) & P_3^n(\boldsymbol{\xi}) \end{pmatrix} \tag{2}$$

From the computational point of view, the assembly of the matrix $J(\boldsymbol{\xi})$ requires $9 \cdot N$ multiplications and $9 \cdot N$ additions, and the computation of the $3 \times 3$ determinant $|J(\boldsymbol{\xi})|$, 9 multiplications and 5 additions.

If fast computation is mandatory, the technique proposed by Knupp in [15] and based on a geometrical decomposition of hexahedra and wedges into tetrahedra, can be considered.

#### *4.1.2 Normalized Jacobian value*

The Jacobian of the mapping $F$ at parent point $\boldsymbol{\xi}$ represents the ratio between the measure of an infinitesimal volume located around $\boldsymbol{\xi}$ and its image by $F$ in the element actual configuration, as suggested by the notation $\partial \boldsymbol{x} / \partial \boldsymbol{\xi}$.

At element definition stage, parent coordinates magnitudes are usually not related to the physical world of the modeled domain. If the modeled domain has dimensions the order of the millimeter and the mesh nodes coordinates are expressed in meters, then these coordinates are the order of 0.001. The parent coordinates, on the other hand, are usually defined in a standard frame and their value range is in the order of 1.

As a consequence, if we consider a perfectly regular element, the Jacobian computed at one of the nodes already carries the information of the scale change in all 3 dimensions. Its value is thus $0.001 \cdot 0.001 \cdot 0.001 = 10^{-9}$. This small value, although mathematically correct, does not numerically reflect the evenness of the element shape. In this case, the expected Jacobian value, ignoring the scale change, would be 1.

In order to measure element distortion while eliminating the scale factor between parent and actual element referential, we use a "normalized" value of the Jacobian defined as the product of the original Jacobian value defined above and a correction term $K$ computed as follows.

Let's consider the trihedral defined by the Jacobian under study. Let $\overline{\boldsymbol{x}}$ be the mean edge length of the 3 edges forming the trihedral in the actual element frame. Now let

$\overline{\boldsymbol{\xi}}$ be the mean edge length of the corresponding trihedral in parent configuration. We define the correction factor in 3D as:

$$K = \left(\frac{\overline{\boldsymbol{\xi}}}{\overline{\boldsymbol{x}}}\right)^3$$

In 2D the above fraction should be raised to the power of 2. The normalized value of the Jacobian computed at parent point $\boldsymbol{\xi}$ is now defined as:

$$\overline{J}(\boldsymbol{\xi}) := K\, J(\boldsymbol{\xi})$$

If we consider the initial example where a scale factor of 0.001 was applied to all 3 coordinates between parent and actual element frame, the normalized value of the Jacobian for all element points is now $\overline{J} = 1$. This fact can be formally apprehended by considering the following rewriting of the normalized Jacobian and noting that the 3D determinant raises the contents of the Jacobian matrix to the power of 3:

$$\overline{J}(\boldsymbol{\xi}) = \left(\frac{\overline{\boldsymbol{\xi}}}{\overline{\boldsymbol{x}}}\right)^3 \left|\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right| = 1$$

To simplify subsequent computations, we will assume that both mean edge lengths $\overline{\boldsymbol{\xi}}$ and $\overline{\boldsymbol{x}}$ computed beforehand using the input mesh configuration, do not significantly vary during validity or quality relaxations. The $K$ scale factor computed for a given Jacobian is thus considered constant.

In the remainder of the article, for the sake of clarity, we will omit the bar placed over the $J$ symbol, as in the above notation, and refer to the normalized Jacobian simply as "Jacobian".

### *4.1.3 Jacobian differentiation*

As the relaxation procedure relies on variational computation of Jacobian or JR values w.r.t. actual element nodes coordinates, it is interesting to derive from the formulation given in §4.1.1 the required differential expression. To this end, let's consider a Jacobian value estimated at parent point $\boldsymbol{\xi}$ within an element built on the set of nodes $\{\boldsymbol{x}^n\}_{n=1,\ldots,N}$.

The Jacobian matrix depends now on the element nodes positions, hence $J = J(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N, \boldsymbol{\xi})$, and we shall seek the expression of the partial derivative of the determinant $\left|J(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N, \boldsymbol{\xi})\right|$ w.r.t. the $i^{\text{th}}$ coordinate of the $n^{\text{th}}$ node, i.e.:

$$\partial \left|J(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N, \boldsymbol{\xi})\right| / \partial x_i^n$$

For clarity, we denote this quantity as $\partial J_i^n(\boldsymbol{\xi})$. Thus:

$$\partial J_i^n(\boldsymbol{\xi}) := \frac{\partial}{\partial x_i^n} \left|J(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N, \boldsymbol{\xi})\right| = \frac{\partial}{\partial x_i^n} \left|\sum_{n=1}^{N} \begin{pmatrix} x_1^n \\ x_2^n \\ x_3^n \end{pmatrix} \left( P_1^n(\boldsymbol{\xi})\ P_2^n(\boldsymbol{\xi})\ P_3^n(\boldsymbol{\xi}) \right)\right|$$

The analytical expression of the above derivative can be easily obtained for $i = 1, 2, 3$. Let's start by deriving it for $i = 1$. Expression 2 can be rewritten as:

$$J(\boldsymbol{x}^1,\ldots,\boldsymbol{x}^N,\boldsymbol{\xi}) = \begin{pmatrix} \sum_n x_1^n P_1^n(\boldsymbol{\xi}) & \sum_n x_1^n P_2^n(\boldsymbol{\xi}) & \sum_n x_1^n P_3^n(\boldsymbol{\xi}) \\ \sum_n x_2^n P_1^n(\boldsymbol{\xi}) & \sum_n x_2^n P_2^n(\boldsymbol{\xi}) & \sum_n x_2^n P_3^n(\boldsymbol{\xi}) \\ \sum_n x_3^n P_1^n(\boldsymbol{\xi}) & \sum_n x_3^n P_2^n(\boldsymbol{\xi}) & \sum_n x_3^n P_3^n(\boldsymbol{\xi}) \end{pmatrix} \tag{3}$$

Now let's denote $M_{ij}$ the minor of matrix $J$ after removal of row $i$ and column $j$. For example:

$$M_{23} := \begin{vmatrix} J_{11} & J_{12} \\ J_{31} & J_{32} \end{vmatrix} = J_{11}J_{32} - J_{31}J_{12}$$

Using the minor notation, the determinant of the matrix in Eq. 3 can be developed w.r.t. the first row as:

$$\left|J(\boldsymbol{x}^1,\ldots,\boldsymbol{x}^N,\boldsymbol{\xi})\right| = M_{11}\sum_n x_1^n P_1^n(\boldsymbol{\xi}) - M_{12}\sum_n x_1^n P_2^n(\boldsymbol{\xi}) + M_{13}\sum_n x_1^n P_3^n(\boldsymbol{\xi})$$

After deriving the above expression w.r.t. $x_1^n$, it follows that:

$$\partial J_1^n(\boldsymbol{\xi}) = M_{11}P_1^n(\boldsymbol{\xi}) - M_{12}P_2^n(\boldsymbol{\xi}) + M_{13}P_3^n(\boldsymbol{\xi})$$

Finally, the general expression of $\partial J_i^n(\boldsymbol{\xi})$ can be obtained by developing the determinant of the Jacobian matrix w.r.t. the $i^{\text{th}}$ row:

$$\partial J_i^n(\boldsymbol{\xi}) = (-1)^{i+1}\left[M_{i1}P_1^n(\boldsymbol{\xi}) - M_{i2}P_2^n(\boldsymbol{\xi}) + M_{i3}P_3^n(\boldsymbol{\xi})\right]$$

### 4.2 Local relaxations

The goal of the relaxation procedures is to successively optimize the Jacobian and JR values, during validity recovery and quality optimization phases respectively, by moving the "improper nodes" (INs), i.e. the mesh nodes which Jacobian or JR values need improvement. As the INs are being moved by the relaxation procedure, the Jacobian or JR values at the neighboring nodes can also be altered and potentially degraded.

Given the large number of nodes often found in FE meshes, repairing procedures can quickly become computationally prohibitive. This complexity can be greatly reduced by grouping all INs and their neighborhood into "repair regions" defined in such a way that all nodal corrections applied inside a region leave the outside mesh configuration unchanged. This local repair approach makes it possible to perform all relaxation procedures independently on each repair region identified within the mesh. The construction of such regions is described below.

The topological definition of repair region is different between validity and quality optimization phases. This difference stems from the criteria being optimized: Jacobian or JR. During validity recovery, the displacement of a mesh node $\mathbf{n}$ affects the Jacobians of all nodes connected by an edge to $\mathbf{n}$. While during quality optimization phase, the Jacobian at $\mathbf{n}$ can become the maximal Jacobian of any element $\mathbf{n}$ is part of, and thus the JR of any node sharing an element with $\mathbf{n}$ can be affected by its position shift.

Prior to **mesh validity recovery**, the repair regions are extracted as follows. All mesh nodes are examined. If a node **n** has a negative or null Jacobian within an element and is not already part of a region then a new region is grown starting from **n**.

All Jacobians influenced by the newly found improper node **n** are listed. For each of these Jacobians, the 4 nodes forming the Jacobian trihedral are evaluated. If one of these nodes appears to be invalid, it is assigned to the current region and all its related Jacobian trihedrals within the mesh are in turn examined. The repair region growth stops where all nodes under scrutiny have strictly positive Jacobian values within all elements they are part of.

The repair regions have the following properties. An IN is part of one and only one repair region. The same being true of Jacobian trihedrals, the regions are computationally independent. A repair region can comprise many INs along with their associated Jacobian trihedrals.

The repair region extraction performed before **quality optimization** only slightly differs from the above algorithm. All mesh nodes are examined. If a node **n** has a JR below the acceptance threshold $\mathrm{JR}_{min}$, and is not already part of a region then a new region is grown starting from the newly found improper node **n**.

All JR influenced by **n** are listed and for each one, the nodes forming the associated element are evaluated. If one of these nodes appears to have a poor quality level, it is assigned to the current region and all its related JR are in turn examined. The region growth stops where all nodes under scrutiny have satisfactory JR values within all elements they are part of. As before, the resulting repaired regions are computationally independent.

During mesh relaxation phases, it is not uncommon to find a repair region having a nodal configuration for which a solution cannot be found by only moving its INs. In such case, the number of degrees of freedom of the region needs to be extended by allowing other nodes to be displaced by the relaxation procedure.

In our mesh repairing algorithm, we have implemented a straightforward region extension method. A locking situation is detected in a repair region when the attractor of the region relaxation is a nodal configuration having persistent INs. Should this situation arise, the region degrees of freedom are extended by allowing the displacement of a mesh node connected to a region IN by an edge, during mesh validity recovery phase, or by an element, during quality optimization phase.

This new node is tagged as “improper” although its associated Jacobian or JR values are acceptable, as it was not identified as IN in the first place. Validity or quality criteria are used to choose the new IN among all possible neighbors to the region INs. During mesh validity recovery or quality optimization, the node having the lowest Jacobian or JR respectively, is selected.

As the extended region has a wider influence on the surrounding mesh configuration and might overlap with another region, a “fusion” check is performed after a region has been extended. To this end the sets of Jacobians influenced by the INs of all regions are checked against each other. If a common Jacobian is found between two regions, these regions are merged into a single one. This way, the resulting repaired regions layout retains the initial numerical independence that allows all regions to be processed separately and efficiently.

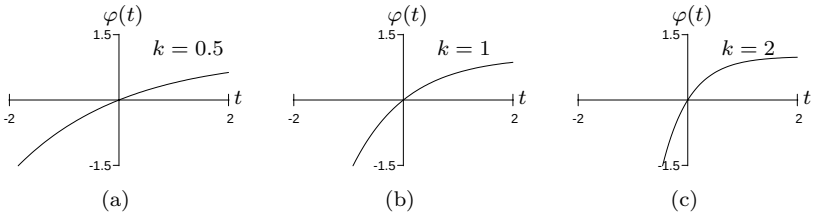


**Fig. 4** Penalty function $\varphi_k$ plot, at the vicinity of 0, for k=0.5 (a), k=1 (b) and k=2 (c).

### 4.3 Validity and quality energy formulation

The goal of validity recovery for a mesh region $R$ is to find a configuration where all the Jacobians[2] in $R$, $\{J_j\}_{j\in R}$, are positive. This nodal relaxation can be formulated as a maximization procedure driven by a "validity energy" $E_V$, expressed as the sum of all Jacobians within $R$.

As the maximization of a sum of terms does not guarantee that all terms end up being positive, all the Jacobian values within the sum are affected by a penalty function $\varphi_k$, which strength is controlled by an index $k$, giving the expression of validity energy:

$$E_V = \sum_{j\in R} \varphi_k(J_j) \quad \text{, where} \quad \varphi_k(t) = 1 - e^{-kt}$$

As the parameter $k$ is increased during the maximization process, the influence of negative values overbalances the positive ones thus favoring a solution where all Jacobians in the sum are positive.

Fig. 4 shows the plot of $\varphi_k$ for three different values of $k$. As can be seen in the figure, $\varphi_k(t) \leq 1$ for all values of $t$, the slope at 0, which is the acceptance threshold for the region Jacobians, is $k$ and the function decreases rapidly as $t$ reaches lower negative values.

The aim of the quality optimization procedure, on the other hand, is to find a nodal configuration where all the JR in a region $R$, $\{\mathrm{JR}_j\}_{j\in R}$, are above a predefined value $\mathrm{JR}_{min}$ which, in our implementation, was set to $0.0\overline{3}$ in accordance with the quality standard requested by the commercial FE analysis software ANSYS Workbench (ANSYS Inc., USA) [13].

We define the "quality energy" $E_Q$ associated to this relaxation problem as the sum of the JR within $R$ weighted by a different penalty function $\psi_l$, thus:

$$E_Q = \sum_{j\in R} \psi_l(\mathrm{JR}_j) \quad \text{, where} \quad \psi_l(t) = 1 - e^{l(\mathrm{JR}_{min} - t)}$$

Similarly to mesh validity recovery, a penalty parameter $l$ is used to find a solution where all the JR contributing to the sum $E_Q$ are above the quality threshold $\mathrm{JR}_{min}$. The penalty functions $\psi_l$ behave similarly to $\varphi_k$: all values are lower than 1, the slope at the JR acceptance threshold $\mathrm{JR}_{min}$ is $l$ and all values below this limit are increasingly penalized as the value of $l$ is raised during relaxation.

[2] Note that in the remainder of the article, all considered Jacobian values are assumed to be normalized as described in 4.1.2

Prior to region relaxation at validity recovery or quality optimization stage, the initial values of the parameter for the corresponding penalty function $\varphi_k$ or $\psi_l$ must be carefully chosen. Indeed, given their formulation, an excessive penalization of unwanted Jacobian or JR values might induce strong numerical instabilities in the optimization process. To avoid this issue, the starting value of $k$ or $l$ is determined by computing the slope of the considered penalty function at the most penalized energy term (minimal Jacobian during validity recovery; minimal JR during quality optimization) and ensuring that it does not exceed a predefined slope threshold.

As the optimization advances, the penalty level $k$ or $l$ is steadily risen to force all energy term values to converge towards the desired interval: $]0, 1]$ during validity recovery and $]\mathrm{JR}_{min}, 1]$ during quality optimization phase.

Both optimizations are carried out by gradient ascent, with a limited number of ascent steps, each having a limited amplitude. These restrictions ensure that the nodal configuration of the relaxed mesh retains a certain similarity with the input mesh. The detailed relaxation algorithm is discussed in §4.4.

### 4.4 Validity and quality energy maximization

Both relaxation procedures, i.e. the maximizations of the energies $E_V$ and $E_Q$ defined above, are performed for each region by gradient ascent. The optimal nodal correction directions are given by the gradient of the energy with respect to the coordinates of the INs in the considered region. These vectors are computed at each step using the derivatives of the penalty functions $d\varphi_k/dt$ or $d\psi_l/dt$ along with the expression of the Jacobian differential given in §4.1.3.

The procedure successfully stops once an acceptable nodal configuration has been found. Otherwise, if locking configurations are detected, successive region extensions are performed and the relaxation is repeated with a higher number of degrees of freedom in the region under study. The relaxation procedure fails if no suitable nodal configuration can be found.

The mesh repair technique presented here was developed as part of the Mesh-Match-and-Repair [3] algorithm which relies on a non–rigid deformation of a generic mesh towards a specific geometry. In this context, the quality of the domain surface representation yielded by the non–rigid deformation procedure must be preserved as much as possible by the mesh relaxation.

In the biomedical field, the dimensions of the modeled organs are the order of the centimeter, and usually a sub–millimetric mean surface representation accuracy is required to ensure the relevance of the Finite Elements analysis carried upon. To control the alteration of surface representation produced by non–rigid registration, our implementation of mesh repair limits the number of relaxation steps to 50, and the maximal amplitude of nodal corrections at each step to 0.1 mm. The maximal allowed nodal displacement is thus 5 mm, a situation which seldom occurs as it requires the same node to be moved in a constant direction throughout the relaxation process.

The common structure for both relaxation procedures is given by Alg. 1 and the main steps (identified by a number) are commented below.

Critical steps within the above relaxation algorithm are commented below.

- [1] : To ensure numerical stability of the relaxation, find the maximal value of $k$ or $l$, depending on the current relaxation phase, which complies with the penalty function slope limit, as discussed in §4.3.

**Algorithm 1** Relaxation algorithm for mesh validity recovery and quality optimization

```
Identify repair regions within mesh : R_r , r = 1, ..., R
for r = 1 to R do {Regions loop}
  repeat {Region extension loop}
    Find initial penalty parameter for region R_r : k or l [1]
    for penalty = 1 to 5 do {Penalty loop}
      for step = 1 to 10 do {Energy maximization loop}
        Compute energy value and gradient at current region configuration
        Find optimal nodal correction in gradient direction within 0.1 mm range [2]
        Apply corrections to region nodes
        if R_r is repaired then
          Leave energy maximization loop
        end if
      end for
      if R_r is repaired then
        Leave penalty loop
      else
        Increase penalty level k or l [3]
      end if
    end for
    if R_r is repaired then
      Leave region extension loop
    else
      Extend region R_r [4]
    end if
  until R_r cannot be extended [5]
  if R_r is repaired then
    Region repair was successful
  else
    Region repair failed
  end if
end for
```

- [2] : The considered relaxation energy gradient vector being formed by the derivatives of the energy w.r.t. each 3D coordinate of the region improper nodes, the coordinates of each nodal correction appear in the gradient vector on the corresponding row. In our specific implementation, the magnitudes of the nodal corrections are clamped, at each step, to 0.1 mm by applying a scale factor to the overall gradient vector. The value of the maximal nodal correction amplitude used here is not universal and must be determined for each application field according to the dimensions of the modeled domain and maximal tolerance on the representation of its geometry. The local maximum line search in the gradient direction is performed using the golden section technique [26] between the current nodal configuration and the configuration obtained after applying the maximal allowed amplitude correction.
- [3] : Raise the value of $k$ or $l$ so as the slope at the most penalized term reaches its maximal allowed value.
- [4] : Increase the region number of degrees of freedom by including new nodes, edge or element-connected to the region improper nodes, as described in §4.2.
- [5] : The region extension can be carried out to the point where the region overlaps the entire mesh and hence all other regions. Yet it appears unnecessary to put into practice this exhaustive and computationally prohibitive solution as in most cases, acceptable region configurations can be found by including in the relaxation process the immediate neighborhood of the originally improper region nodes. Thus,

the iterations stop when all mesh nodes connected to the initial improper nodes in the considered region have been included.

### 4.5 Repairing example

Fig. 5 shows an example of mesh registration and repair for a face model. At the end of the repair process, all the elements were valid and presented acceptable quality (JR $> 0.0\overline{3}$). In particular, Fig. 6 shows a sample validity recovery for a repaired region. The invalid Jacobian trihedrals are drawn in dark and the valid Jacobians in light gray-scale.

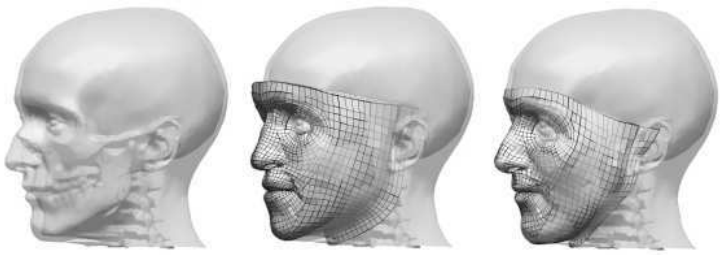

**Fig. 5** Left: input patient data (skull and skin surfaces segmented from medical images). Middle: patient data and Atlas mesh. Right: the final mesh registered onto the patient's skin and skull and repaired.

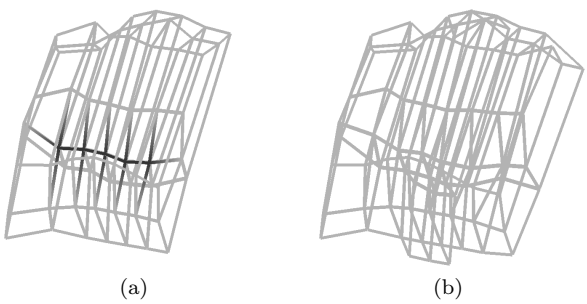


**Fig. 6** (a) Initial invalid region having 5 improper nodes (INs) and 24 elements and (b) the repaired region formed by 32 elements.

The region repair depicted in Fig. 6 took approximately 600 ms. The region had initially 5 invalid nodes and comprised 24 elements. In order to find a suitable nodal configuration the region had to be extended by 2 nodes, therefore including 8 more elements in the region. The mean magnitude of the corrections applied to the 7 displaced mesh nodes was 0.5 mm (max = 0.8 mm, $\sigma = 0.1$ mm).

More examples like this can be found in [3] where 10 femur and 50 face FE meshes were successfully registered and repaired in the context of orthopaedic and orthognathic hard and soft tissues modelling studies.

## 5 Conclusions

A novel mesh repair algorithm, based on a mesh relaxation driven by specific validity and quality energy formulations, has been presented. It first recovers mesh validity and further improves mesh elements quality in order to enhance numerical results reliability. Both procedures rely on the computation of Jacobian values within the mesh elements, which is a mesh acceptance criterion widely used in popular FE analysis softwares. Within both validity and quality energy formulations, the Jacobian measures are weighted by carefully designed penalty functions that lead the nodal corrections towards a configuration where both mesh validity and quality criteria are satisfied. The procedure proposed here was successfully used to produce FE meshes suitable for FE analysis within commercial softwares [3,4], such as ANSYS (Ansys Inc., USA).

The minimized energy function presented in this work is not smooth. To approximate the descent direction despite the slope discontinuities a Centered Finite Differences (CFD) scheme was used. From a numerical point of view, the CFDs act as a smoothing operator over the non–smooth energy landscape yielding a good estimate of the gradient in the smooth domain and managing continuous transitions through the slope discontinuity boundaries.

Our mesh repair procedure only affects invalid and poor quality regions, leaving other mesh areas unchanged. If an acceptable configuration cannot be found by applying corrections to invalid or poor quality nodes, other nodes are progressively included into the repair process, according to their validity and quality levels, following the region extension procedure described above, in §4.2. This strategy preserves valid mesh regions as well as those with higher quality level. In addition to the initially identified improper nodes, only nearly invalid or low quality nodes can be reallocated by our procedure which strongly limits the difference between the repaired and initial input mesh.

The region–based mesh untangling process proposed in the article is able to deal with Jacobian inter–locking situations where a solution can only be found by moving simultaneously a number of nodes involved in shared elements. These situations require the whole set of involved elements to be considered simultaneously and, from our experience, can hardly be solved using a node–based approach as antagonist nodal displacements can quickly lead to an infinite loop in the mesh untangling routine. Moreover, several mesh untangling techniques proposed in the literature do not separate validity recovery and quality improvement procedures, intending to achieve it all at the same time. The problem with this approach is that if invalid node $a$ and poor quality node $b$ are connected, finding a new position to improve the quality of $b$ may decreases the chances of finding a new valid position for $a$. We consider that to focus on mesh validity first, increases the chances of finding a valid position for $a$. Once $a$ and $b$ are of poor quality, both will have the same chances to increase their quality, but the difference is that both start from a valid mesh.

To our knowledge, the only other strategy to perform such a task was introduced by Luboz et al. [20]. The differences with this work and the main contributions of our approach are summarized bellow.

- In order to compute the new position of an invalid node $\boldsymbol{n}$, only the elements that were invalid at $\boldsymbol{n}$ were considered in the work of Luboz et al. Our technique, on the other hand, takes into consideration all the elements attached to $\boldsymbol{n}$ as its displacement can cause previously valid elements to become invalid.
- In our approach, the nodes to be repaired are grouped in independent regions to avoid prohibitive computational costs.
- As opposed to Luboz et al., our algorithm considers mesh quality as well as mesh validity indicators. It is thus capable of producing a mesh that guarantees the stability of the numerical solution yielded by the FE analysis carried out upon.

## 6 Perspectives

In the context of Mesh–Matching, repair strategies should take into account the surface representation constraint, in situations where surface nodes are being moved during relaxation. This can be accomplished by considering the gradient of the energy undergoing maximization at a given improper node not along all space directions, as described above, but tangentially to the represented organ surface. The surface representation constraint should only be implemented for surface mesh nodes, i.e. the ones that need to rest as close as possible to the modeled domain surface during the relaxation, to ensure the accuracy of geometry representation.

Domain surface representation could be further maintained by correcting small registration degradations likely to occur during repairing. To this end, an Atlas–to–surface registration step might be performed after each relaxation step. Nevertheless, as registration and relaxation are independent and not necessarily collaborative procedures, the convergence of this alternating method should be carefully studied so as to avoid infinite loops and eventually reach a repaired mesh configuration.

The local repair technique presented in §4.2 has proved to be effective in all cases considered so far [3]. Yet, nodal configurations featuring intricate element inter–locking can be devised so that the approach proposed here fails. To overcome this issue, alternate relaxation paradigms can be considered.

Indeed, we have often observed that during relaxation complex repair regions transit through a configuration with a strongly reduced number of improper nodes. In cases where the relaxation fails, instead of extending the original repaired region and repeating the process from its initial configuration, a more efficient approach would be to start from the region configuration featuring the minimal number of INs encountered during the unsuccessful relaxation.

Should a repair region be restricted to this smaller subset, its influence within the mesh must be recomputed and possible region overlaps checked for. Furthermore, in order to control the nodal correction amplitudes, track must be kept of anterior nodal displacements in order to guarantee that the original surface representation does not get excessively perturbed. Although promising, this approach has not yet been implemented in our software.

Finally, future work should be focused on adding other quality criteria to repair the mesh. The JR is an excellent quality criterion that helps to detect invalid and poor quality elements, but it cannot detect, for instance, elements with small dihedral angles or elements with warped faces. For these reasons, other quality criteria like dihedral angles, aspect–ratio, volume, warping factor, etc. should be also considered to ensure a quality output mesh.

**Acknowledgements** This project has been financially supported by FONDECYT 1061227, ALFA IPECA project, STIC–AmSud PLOMO, and ECOS-Sud C06E04.

## References

1. Alliez, P., Cohen-Steiner, D., Yvinec, M., Desbrun, M.: Variational tetrahedral meshing. ACM Transactions on Graphics **24**(3), 617–625 (2005)
2. Berar, M., Desvignes, M., Bailly, G., Payan, Y.: 3d meshes registration : Application to statistical skull model. In: Image Analysis and Recognition, *Lecture Notes in Computer Science*, vol. 3212, pp. 100–107. Springer Berlin / Heidelberg (2004). DOI 10.1007/b100438
3. Bucki, M., Lobos, C., Payan, Y.: A fast and robust patient specific finite element mesh registration technique: Application to 60 clinical cases. Medical Image Analysis **14**(3), 303–317 (2010)
4. Bucki, M., Nazari, A., Payan, Y.: Finite element speaker-specific face model generation for the study of speech production. In: In proceedings of 1st International Workshop on Dynamic Modeling of the Oral, Pharyngeal and Laryngeal Complex for Biomedical Applications (OPAL-2009), pp. 129–136 (2009)
5. Castellano-Smith, A., Hartkens, T., Schnabel, J., Hose, R., Liu, H., Hall, W., Truwit, C., Hawkes, D., Hill, D.: Constructing patient specific models for correcting intraoperative brain deformation. In: MICCAI, pp. 1091–1098 (2001)
6. Couteau, B., Payan, Y., Lavallee, S.: The mesh-matching algorithm: an automatic 3d mesh generator for finite element structures. Journal of Biomechanics **33**(8), 1005–1009 (2000)
7. Delingette, H.: General object reconstruction based on simplex meshes. International Journal of Computer Vision **32**(2), 111–146 (1999)
8. Field, D.: Qualitative measures for initial meshes. International Journal For Numerical Methods In Engineering **47**, 887–906 (2000)
9. Freitag, L.: On combining laplacian and optimization-based mesh smoothing techniques. In: Trends in Unstructured Mesh Generation, vol. 220, pp. 37–44. ASME Applied Mechanics Division (1997)
10. Freitag, L., Plassmann, P.: Local optimization-based simplicial mesh untangling and improvement. International Journal of Numerical Methods in Engineering **49**(1–2), 109 – 125 (2000)
11. George, P., Borouchaki, H., Laug, P.: An efficient algorithm for 3d adaptive meshing. Advances in Engineering Software **33**, 377–387 (2002)
12. Hughes, T.: The finite element method: Linear static and dynamic finite element analysis. Dover Publications (1987)
13. Kelly, S.: Element shape testing. In: Ansys Theory Reference, chap. 13. Ansys Inc. (1998)
14. Knupp, P.: Matrix norms & the condition number, a general framework to improve mesh quality via node-movement. In: Proceedings, 8th International Meshing Roundtable, pp. 13–22 (1999)
15. Knupp, P.: Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. part ii – a framework for volume mesh optimization and the condition number of the jacobian matrix. International Journal for numerical methods in engineering **48**, 1165–1185 (2000)
16. Knupp, P.: Hexahedral mesh untangling & algebraic mesh quality metrics. In: Proceedings, 9th International Meshing Roundtable, pp. 173–183 (2000)
17. Kwok, W., Chen, Z.: A simple and effective mesh quality metric for hexahedral and wedge elements. In: Proceedings, 9 th International Meshing Roundtable, pp. 325–333 (2000)
18. Lobos, C., Bucki, M., Hitschfeld, N., Payan, Y.: Mixed-element mesh for an intra-operative modeling of the brain tumor extraction. In: Proceedings, 16 th International Meshing Roundtable, pp. 387–404. Sandia National Laboratories, Springer-Verlag (2007)
19. Luboz, V., Chabanas, M., Swider, P., Payan, Y.: Orbital and maxillofacial computer aided surgery: patient-specific finite element models to predict surgical outcomes. Computer methods in biomechanics and biomedical engineering **8**(4), 259–265 (2005)
20. Luboz, V., Payan, Y., Couteau, B.: Automatic 3d finite element mesh generation: an atlas fitted onto patient data. In: Proceedings of the Fifth International Symposium on Computer Methods in Biomechanics and Biomedical Engineering (2001)
21. Molino, N., Bridson, R., Teran, J., Fedkiw, R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In: Proceedings, 12 th International Meshing Roundtable, pp. 103–114. Sandia National Laboratories, Springer-Verlag (2003)

22. Montagnat, J., Delingette, H.: Globally constrained deformable models for 3d object reconstruction. Signal Processing (SP) **71**(2), 173–186 (1998)
23. Montagnat, J., Delingette, H.: 4d deformable models with temporal constraints: application to 4d cardiac image segmentation. Medical Image Analysis (MedIA) **9**(1), 87–100 (2005)
24. Nesme, M.: Milieu mecanique deformable multiresolution pour la simulation interactive. Ph.D. thesis, Universite Joseph Fourier (2008)
25. Paoletti, S.: Polyhedral mesh optimization using the interpolation tensor. In: Proceedings, 11th International Meshing Roundtable, pp. 19–28 (2002)
26. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flanery, B.P.: Numerical Recipes in C. Cambridge University Press (1992)
27. Schnabel, J., Rueckert, D., Quist, M., Blackall, J., Castellano-Smith, A., Hartkens, T., Penney, G., Hall, W., Liu, H., Truwit, C., Gerritsen, F., Hill, D., Hawkes, D.: A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations. In: MICCAI, pp. 573–581 (2001)
28. Shewchuk, J.: What is a good linear element? interpolation, conditioning, and quality measures. In: Proceedings of the 11 International Meshing Roundtable, pp. 115–126. Sandia National Laboratories, Springer-Verlag (2002)
29. Zhang, Y., Bajaj, C.: Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. Computer Methods in Applied Mechanics and Engineering **195**(9–12), 942–960 (2006)
30. Zienkiewicz, O., Taylor, R.: The Finite Element Method. McGraw-Hill (1989)