A new least-squares approximation of affine mappings for sweep algorithms

Xevi Roca · Josep Sarrate · Antonio Huerta

Abstract This paper presents a new algorithm to generate hexahedral meshes in extrusion geometries. Several algorithms have been devised to generate hexahedral meshes by projecting the cap surfaces along a sweep path. In all of these algorithms the crucial step is the placement of the inner layer of nodes. That is, the projection of the source surface mesh along the sweep path. From the computational point of view, sweep methods based on a least-squares approximation of an affine mapping are the fastest alternative to compute these projections. Several functionals have been introduced to perform the least-squares approximation. However, for very simple and typical geometrical configurations they may generate low-quality projected meshes. For instance, they may induce skewness and flattening effects on the projected discretizations. In addition, for these configurations the minimization of these functionals may lead to a set of normal equations with singular system matrix. In this work we analyze previously defined functionals. Based on this analysis we propose a new functional and show that its minimization overcomes these drawbacks. Finally, we present several examples to assess the properties of the proposed functional.

1 Introduction

The finite element method is currently used to simulate and analyze a wide range of problems in applied sciences and engineering. There are several 3D applications where hexahedral elements are preferred. Hence, the general interest in unstructured hexahedral discretizations has increased. Since an all-hexahedral mesh generation algorithm for any arbitrary geometry is still an unattained goal, research efforts are focused on algorithms that decompose the entire geometry into several simpler volumes. In particular, during the past decade significant progress has been made in developing fast and robust sweeping algorithms [1 4]. Nowadays, the original sweep methods have been modified in order to mesh more complicated geometries allowing multiple source and target geometries [5, 6], and multiple axis geometries [7].

Given an extrusion volume, the common task of all sweeping algorithms is to identify the *source surfaces*, the corresponding *target surfaces*, and the set of surfaces that join them, called *linking sides*. The source surfaces can be meshed using any structured or unstructured quadrilateral surface mesh generator [8 11]. However, the linking sides must be meshed using a structured quadrilateral algorithm, for instance, transfinite interpolation [12]. Then, the source surface meshes are extruded along the sweep direction until

X. Roca · J. Sarrate (✉) · A. Huerta
Laboratori de Càlcul Numèric, ETSE de Camins Canals i Ports de Barcelona, Universitat Politècnica de Catalunya, Edifici C2, Jordi Girona 1 3, 08034 Barcelona, Spain
e mail: jose.sarrate@upc.edu

X. Roca
e mail: xevi.roca@upc.edu

A. Huerta
e mail: antonio.huerta@upc.edu

they reach the target surfaces. Note that the target surfaces may or may not be previously meshed.

In general, the inner nodes are placed, layer by layer, along the sweep direction. Each layer is delimited by loops of nodes that belong to the structured meshes of the linking-sides. Several algorithms have been developed in order to generate the inner layer of nodes. Most of them generate the new nodes by means of a projection of the source surface mesh onto the inner layers. From the computational point of view, sweep methods based on a least-squares approximation of an affine mapping are the fastest alternative to compute these projections [13]. That is, the inner nodes are located using a least-squares approximation of a linear transformation (the homogeneous part of an affine mapping) between the boundary nodes of the source surface and the boundary nodes of the inner layer [1, 5].

From the construction point of view, two main strategies are used to compute the position of the inner nodes. The first one computes, starting from the source surface, the position of the new layer from the previous one in an *advancing front manner* [1, 7] (see Fig. 1a). The second one computes the position of each layer always projecting from the cap surfaces [4] (see Fig. 1b). In addition, the location of the inner nodes can be computed only
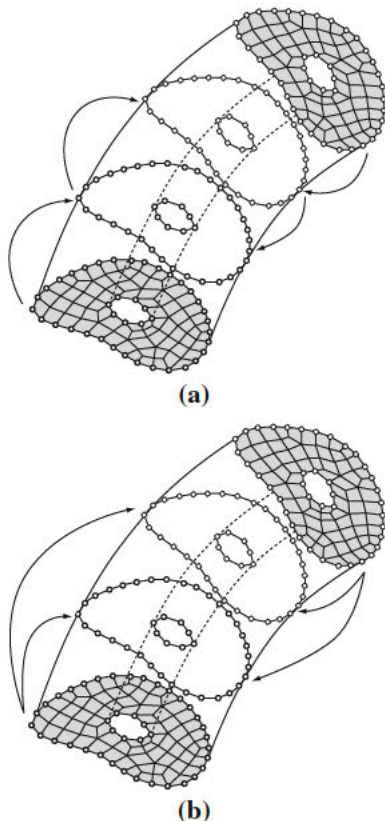

(a)

(b)

Fig. 1 Strategies to generate hexahedral meshes in a sweep procedure: **a** projecting from the previous layer; **b** projecting always from the cap surfaces

projecting in one direction of the sweep path (i.e. from the source surface to the target surface) or averaging the computed projections from both the source and target surfaces [3 5]. The latter is the general choice.

Most of the proposed algorithms to map meshes between surfaces involve an orthogonal projection of nodes onto the target surface [2, 14]. These projections are expensive from a computational point of view since it is necessary to solve as many root finding problems as internal points are on the mesh of the source surface. In order to address this shortcoming, Roca et al. [4] presented a method such that the projection of the source mesh onto the target surface is determined by means of a least-squares approximation of an affine mapping. This affine mapping is defined between the parametric representations of the loops of boundary nodes of the cap surfaces. Once the new mesh is obtained on the parametric space of the target surface, it is mapped up according to the target surface parameterization. It is important to point out that projection algorithms can be used to project meshes both in 2D (parametric space) and 3D (physical space) geometries.

The projection algorithms cited above are based on a least-squares approximation of a linear transformation. To this end, several functionals are defined (see Sects. 2, 3). However, the minimization of these functionals does not always generate an acceptable projected mesh. For several geometric configurations (for instance, source surfaces with a planar boundary, which are common in real applications), the minimization of one of the proposed functionals leads to a set of normal equations with a singular system matrix. Nevertheless, these normal equations may be solved using a singular value decomposition algorithm. In this case, if a mesh over a non-planar surface with a planar boundary is projected to an inner layer defined by a planar set of nodes, then a planar mesh is obtained. It is possible to overcome this shortcoming minimizing an alternative functional. However, the minimization of the new functional may induce a skewness effect on the cross-section of the projected surface mesh. Moreover, its minimization may also lead to a set of normal equations with a singular system matrix (see Sect. 3).

In this paper we propose a new functional that overcomes the drawbacks of the previous formulations. Moreover, we prove that the minimization of this new functional has a unique solution. Finally, we present several examples in order to assess the robustness of the new formulation and compare it with the previously proposed functionals.

2 Problem statement

Let $X = \{\mathbf{x}^i\}_{i=1,\dots,m} \subset \mathbb{R}^n$ be a set of source points, and $Y = \{\mathbf{y}^i\}_{i=1,\dots,m} \subset \mathbb{R}^n$ be a set of target points with $m \geq n$. Our goal is to find a mapping $\phi : \mathbb{R}^n \to \mathbb{R}^n$ such that

$$\mathbf{y}^i = \boldsymbol{\phi}(\mathbf{x}^i), \quad i = 1, \ldots, m. \tag{1}$$

We approximate $\boldsymbol{\phi}$ by an affine mapping $\boldsymbol{\varphi}$ from $\mathbb{R}^n$ to $\mathbb{R}^n$. This affine mapping is determined by a least-squares fitting of the given data. Thus, we want to find $\boldsymbol{\varphi}$ such that it minimizes the functional

$$E(\boldsymbol{\varphi}) := \sum_{i=1}^m \left\| \mathbf{y}^i - \boldsymbol{\varphi}(\mathbf{x}^i) \right\|^2. \tag{2}$$

The above minimization has a clear geometrical meaning: the optimal affine mapping is such that the sum of the square of distances between the target points and the image of the source points is minimized. We also define

$$\mathbf{c}^X := \frac{1}{m} \sum_{i=1}^m \mathbf{x}^i \quad \text{and} \quad \mathbf{c}^Y := \frac{1}{m} \sum_{i=1}^m \mathbf{y}^i \tag{3}$$

as the geometric centers of the sets X and Y, respectively.

Remark 1 One of the most important practical applications of the least-squares fitting of affine mappings is the projection of a given mesh in a sweeping tool. In these applications X is the set of nodes of the source surface boundary, and Y consists of the loop of nodes that define a given inner layer or the boundary nodes of the target surface discretization.

It is well known that any affine mapping $\boldsymbol{\varphi}$ from $\mathbb{R}^n$ to $\mathbb{R}^n$ can be written as

$$\boldsymbol{\varphi}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}',$$

where $\mathbf{A} \in \mathcal{L}(\mathbb{R}^n)$ is a linear transformation (the homogeneous part), and $\mathbf{b}' \in \mathbb{R}^n$ is the affine part. If we consider $\mathbf{b} := \mathbf{b}' + \mathbf{A}\mathbf{c}^X$, then we can write $\boldsymbol{\varphi}$ as

$$\boldsymbol{\varphi}(\mathbf{x}) = \mathbf{A}(\mathbf{x} - \mathbf{c}^X) + \mathbf{b}. \tag{4}$$

Therefore, without loss of generality, we can write the initial least-squares problem (2) as the minimization of the functional

$$E(\mathbf{A}, \mathbf{b}) := \sum_{i=1}^m \left\| \mathbf{y}^i - \mathbf{A}(\mathbf{x}^i - \mathbf{c}^X) - \mathbf{b} \right\|^2, \tag{5}$$

where $\mathbf{A} \in \mathcal{L}(\mathbb{R}^n)$ and $\mathbf{b} \in \mathbb{R}^n$.

Remark 2 It is straightforward to prove that if $(\mathbf{A}^E, \mathbf{b}^E) \in \mathcal{L}(\mathbb{R}^n) \times \mathbb{R}^n$ is the optimal solution of (5), then the affine part of (5) is the geometric center of Y, i.e. $\mathbf{b}^E = \mathbf{c}^Y$. Therefore, using (4), the optimal solution maps the center $\mathbf{c}^X$ to the center $\mathbf{c}^Y$.

3 Alternative formulations

According to Remark 2, the solution of the minimization of E maps the center $\mathbf{c}^X$ to the center $\mathbf{c}^Y$. This property

induces, as Knupp shows in [1], the definition of the new coordinates $\overline{\mathbf{x}} = \mathbf{x} - \mathbf{c}^X$ and $\overline{\mathbf{y}} = \mathbf{y} - \mathbf{c}^Y$. These new coordinates can be interpreted as translating the sets of points X and Y to the origin (see Fig. 2). Using these new coordinates we have

$$
\begin{aligned}
F(\mathbf{A}) &:= \sum_{i=1}^m \left\| \mathbf{y}^i - \mathbf{c}^Y - \mathbf{A}(\mathbf{x}^i - \mathbf{c}^X) \right\|^2 \\
&= \sum_{i=1}^m \left\| \overline{\mathbf{y}}^i - \mathbf{A}\overline{\mathbf{x}}^i \right\|^2.
\end{aligned} \tag{6}
$$

Therefore, we are looking for a linear mapping $\mathbf{A}$ such that approximately transforms, in the least-squares sense, $\overline{X} = \{\overline{\mathbf{x}}^i\}_{i=1,\ldots,m}$ into $\overline{Y} = \{\overline{\mathbf{y}}^i\}_{i=1,\ldots,m}$. Functional (6) is used in [1] in order to reduce the number of degrees of freedom involved in the minimization of functional (5).

However, functional (6) has an important drawback: if the set of source points determines a plane in 3D geometries, or a straight line in 2D applications, then the matrix of the normal equations corresponding to the minimization of functional (6) is singular. Note that this is a typical situation in practical CAD models (see Fig. 3). In order to formalize the analysis of functional F, we introduce the following definitions, lemmas and propositions. All of them apply to the general case of $\mathbb{R}^n$. Therefore, we will
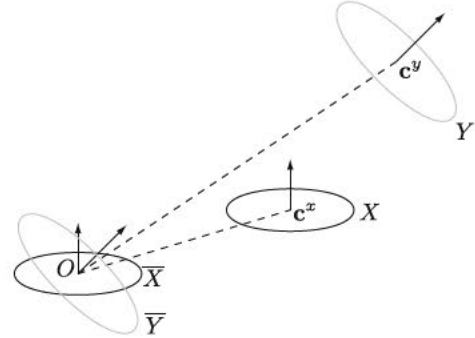

Fig. 2 Geometric representation of the translation of sets X and Y to the origin
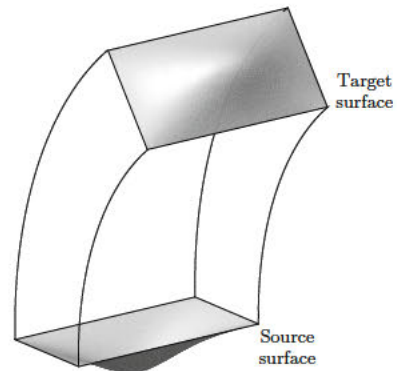

Fig. 3 Example of a geometry with a source surface defined by a planar boundary loop

use the term *hyperplanar* to denote a linear variety of dimension $n - 1$ (a plane for $n = 3$ and a straight line for $n = 2$). From the practical point of view, two cases are important: $n = 2$ used to project sets of points between parametric spaces [4], and $n = 3$ used to project sets of points in the physical space [1, 4].

Definition 1 (*Hyperplanar set*) A set of points $X = \{\mathbf{x}^i\}_{i=1,\ldots,m}$ is hyperplanar if there exists only one hyperplane through all the points in X.

Remark 3 The definition of hyperplanar set has an interesting geometrical interpretation. It states that there exist n points in X that are linearly independent as affine points. In other words, if we take any point of X, the differences between the rest of points of X and the selected point determine a vectorial subspace of dimension $n - 1$.

Definition 2 (*Unitary normal vector*) Let X be a set of points. A unitary normal vector to X is a vector $\mathbf{n}^X \in \mathbb{R}^n$ with $\|\mathbf{n}^X\| = 1$ such that

$$\langle \mathbf{n}^X, \mathbf{x}^i \rangle = c, \quad i = 1, \ldots, m, \tag{7}$$

for some $c \in \mathbb{R}$.

Definition 3 (*Homogeneous hyperplane*) Let X be a hyperplanar set of points. The homogeneous hyperplane of X is the subspace of vectors

$$\mathbb{H} = \{\mathbf{v} \in \mathbb{R}^n | \langle \mathbf{n}^X, \mathbf{v} \rangle = 0\},$$

where $\mathbf{n}^X \in \mathbb{R}^n$ is a normal vector to X.

Lemma 1 *If X is a hyperplanar set, then $\mathbf{c}^X$ is such that*

$$\langle \mathbf{n}^X, \mathbf{c}^X \rangle = c,$$

where $\mathbf{n}^X$ and c are introduced in Definition 2.

Proof Since X is a hyperplanar set, Eqs. (7) hold. Adding these m equations, and taking into account that $\langle \cdot, \cdot \rangle$ is bilinear, then

$$\left\langle \mathbf{n}^X, \sum_{i=1}^m \mathbf{x}^i \right\rangle = mc.$$

Dividing both terms of the last equation by m, and using the definition of the geometric center $\mathbf{c}^X$ we finally obtain $\langle \mathbf{n}^X, \mathbf{c}^X \rangle = c$. □

Lemma 2 *If X is a hyperplanar set, then*

$$\mathbb{H} = \operatorname{span}(\mathbf{x}^1 - \mathbf{c}^X, \ldots, \mathbf{x}^m - \mathbf{c}^X).$$

Proof The proof of this Lemma is straightforward from Definition 1, Remark 3, and Lemma 1.

Proposition 1 *If X is hyperplanar, then the minimization of functional F is equivalent to solving n uncoupled overdetermined linear systems of rank $n - 1$.*

Proof The minimization of functional F is equivalent to imposing the following m constraints:

$$\mathbf{A}(\mathbf{x}^i - \mathbf{c}^X) = \mathbf{y}^i, \quad i = 1, \ldots, m. \tag{8}$$

Our unknowns are the coefficients of the $n \times n$ matrix $\mathbf{A}$ which we denote as

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & \ldots & a_{1,n} \\ \vdots & & \vdots \\ a_{n,1} & \ldots & a_{n,n} \end{pmatrix}.$$

Defining

$$\mathbf{X} := \begin{pmatrix} x_1^1 - c_1^X & \ldots & x_1^m - c_1^X \\ \vdots & & \vdots \\ x_n^1 - c_n^X & \ldots & x_n^m - c_n^X \end{pmatrix}$$

and

$$\mathbf{Y} := \begin{pmatrix} y_1^1 - c_1^Y & \ldots & y_1^m - c_1^Y \\ \vdots & & \vdots \\ y_n^1 - c_n^Y & \ldots & y_n^m - c_n^Y \end{pmatrix}$$

we can write the m constraints (8) as

$$\mathbf{A}\mathbf{X} = \mathbf{Y}.$$

Hence, the minimization of F is equivalent to solving

$$\mathbf{X}^T \mathbf{A}^T = \mathbf{Y}^T.$$

This equation is equivalent to solving the following n $(m \times n)$-overdetermined linear systems:

$$\mathbf{X}^T \mathbf{a}_k = \mathbf{y}_k, \quad k = 1, \ldots, n,$$

where $\mathbf{a}_k := (a_{k,j})$ for $j = 1, \ldots, n$ and $\mathbf{y}_k = (y_k^l - c_k^Y)$, for $l = 1, \ldots, m$. To conclude, we have to prove that $\mathbf{X}^T$ has rank $n - 1$. By Lemma 2, and taking into account that $\dim \mathbb{H} = n - 1$

$$\operatorname{rank} \mathbf{X}^T = \dim \operatorname{span}(\mathbf{x}^1 - \mathbf{c}^X, \ldots, \mathbf{x}^m - \mathbf{c}^X)$$
$$= \dim \mathbb{H} = n - 1.$$

□

Remark 4 It is well known that solving a rank-deficient overdetermined linear system is equivalent to solving a set of normal equations with singular system matrix [15, 16].

When X is hyperplanar, we have seen that the minimization of F amounts to solving n uncoupled overdetermined linear systems of rank $n - 1$. Thus, we have n extra degrees of freedom which allow us to find a solution of the minimization of F such that it has $\mathbf{c}^Y - \mathbf{c}^X$ as a fixed vector. This idea leads to the change of coordinates $\overline{\overline{\mathbf{x}}} = \mathbf{x} - \mathbf{c}^X + \mathbf{c}^Y - \mathbf{c}^X$ and $\overline{\overline{\mathbf{y}}} = \mathbf{y} - \mathbf{c}^X$ (see [1] for details). These new coordinates have a clear geometric interpretation: the sets of points X and Y are translated to $\mathbf{c}^Y - \mathbf{c}^X$

(see Fig. 4). According to [1], these new coordinates suggest the definition of the following functional:

$$G(\mathbf{A}) := \sum_{i=1}^{m} \left\| \mathbf{y}^i - \mathbf{c}^X - \mathbf{A}(\mathbf{x}^i - \mathbf{c}^X + \mathbf{c}^Y - \mathbf{c}^X) \right\|^2$$

$$= \sum_{i=1}^{m} \left\| \overline{\overline{\mathbf{y}}}^i - \mathbf{A}\overline{\overline{\mathbf{x}}}^i \right\|^2. \tag{9}$$

Therefore, we are looking for a linear mapping $\mathbf{A}$ such that it approximately transforms, in the least-squares sense, $\overline{\overline{X}} = \{\overline{\overline{\mathbf{x}}}^i\}_{i=1,\ldots,m}$ into $\overline{\overline{Y}} = \{\overline{\overline{\mathbf{y}}}^i\}_{i=1,\ldots,m}$.

However, functional (9) also leads to normal equations with singular matrix if the vector $\mathbf{c}^Y - \mathbf{c}^X$ lies in the hyperplane determined by the source points (see Fig. 5). Note that this situation is usual in several practical 3D applications if the inner layers are obtained by means of a direct projection from the source surface mesh [4]. On the other hand, this is typically not the case if the position of the new layer is computed from the previous one in an advancing front manner [1].

Proposition 2 *If X is hyperplanar and $\mathbf{c}^Y - \mathbf{c}^X \in \mathbb{H}$, then the minimization of functional G is equivalent to solving n uncoupled overdetermined linear systems of rank $n - 1$.*
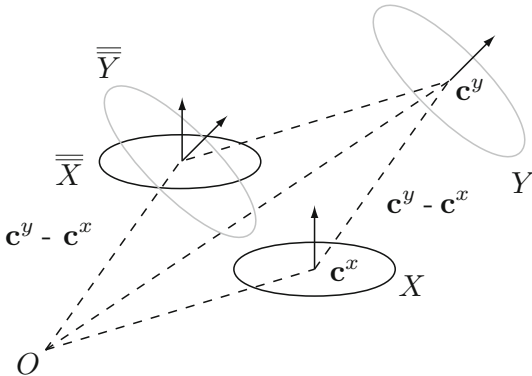
Proof This proof only differs from the proof of Proposition 1 on the definitions of matrices $\mathbf{X}$ and $\mathbf{Y}$. In this case, the correspondent matrices are

$$\mathbf{X} := \begin{pmatrix} x_1^1 - c_1^X + c_1^Y - c_1^X & \ldots & x_1^m - c_1^X + c_1^Y - c_1^X \\ \vdots & & \vdots \\ x_n^1 - c_n^X + c_n^Y - c_n^X & \ldots & x_n^m - c_n^X + c_n^Y - c_n^X \end{pmatrix}$$
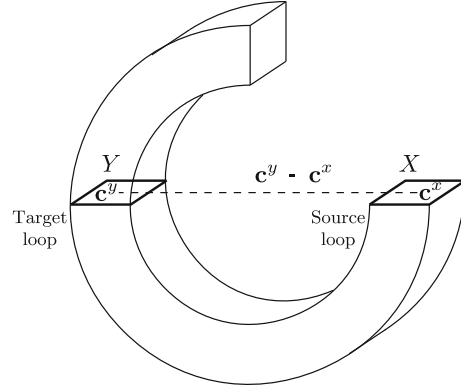
and

$$\mathbf{Y} := \begin{pmatrix} y_1^1 - c_1^X & \ldots & y_1^m - c_1^X \\ \vdots & & \vdots \\ y_n^1 - c_n^X & \ldots & y_n^m - c_n^X \end{pmatrix}.$$

To conclude, we have to show that $\mathbf{X}^T$ has rank $n - 1$. By assumption $\mathbf{c}^Y - \mathbf{c}^X \in \mathbb{H}$, hence

Fig. 4 Geometric representation of the translation of sets X and Y to $\mathbf{c}^Y - \mathbf{c}^X$

Fig. 5 Example of a geometry where $\mathbf{c}^Y - \mathbf{c}^X$ lies in the same plane as the source surface and the boundary of the inner layer

$$\text{span}(\mathbf{x}^1 - \mathbf{c}^X + \mathbf{c}^Y - \mathbf{c}^X, \ldots, \mathbf{x}^m - \mathbf{c}^X + \mathbf{c}^Y - \mathbf{c}^X)$$
$$= \text{span}(\mathbf{x}^1 - \mathbf{c}^X, \ldots, \mathbf{x}^m - \mathbf{c}^X).$$

Finally, using this equation, Lemma 2, and taking into account that $\dim \mathbb{H} = n - 1$, we obtain

$$\text{rank } \mathbf{X}^T$$
$$= \dim \text{span}(\mathbf{x}^1 - \mathbf{c}^X + \mathbf{c}^Y - \mathbf{c}^X, \ldots, \mathbf{x}^m - \mathbf{c}^X + \mathbf{c}^Y - \mathbf{c}^X)$$
$$= \dim \mathbb{H} = n - 1.$$

$\square$

Remark 5 It is also possible to prove that if X is hyperplanar and $\mathbf{c}^Y - \mathbf{c}^X \notin \mathbb{H}$, then the minimization of G is equivalent to solving n uncoupled overdetermined linear systems of rank n.

Remark 6 The minimization of functional G has an additional shortcoming when it is applied to planar sets of points, even in the case of $\mathbf{c}^Y - \mathbf{c}^X \notin \mathbb{H}$. Consider the source surface with a planar boundary and non-planar interior shown in Fig. 6a. Assume that we want to project a source surface mesh to an inner layer (of a sweep volume) defined by a planar boundary, but non-parallel to the source surface. Figure 6b shows a cross-section of the source surface, the correspondent cross-section of the computed projection minimizing functional G, and the desired solution. We know that the optimum affine transformation, $\mathbf{A}^G$, has $\mathbf{c}^Y - \mathbf{c}^X$ as fixed vector. Thus, we can observe that the cross-section obtained with $\mathbf{A}^G$ (gray line in Fig. 6b) does not preserve the shape of the original surface.

4 The new formulation

In order to overcome the drawbacks arising from the minimization of functionals F and G, in this work we propose the following new functional:

(a)

(b)
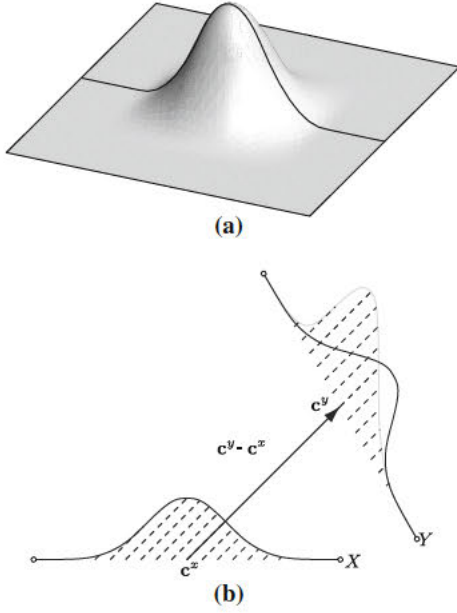
Fig. 6 a A source surface with planar boundary and non planar interior; **b** cross section view of the source surface and its computed image using the minimization of G (*gray line*) and the desired solution (*black line*)

$$H(\mathbf{A}; \mathbf{u}^X, \mathbf{u}^Y) :=$$

$$\sum_{i=1}^{m} \left\| \mathbf{y}^i - \mathbf{c}^Y - \mathbf{A}(\mathbf{x}^i - \mathbf{c}^X) \right\|^2 + \left\| \mathbf{u}^Y - \mathbf{A}\mathbf{u}^X \right\|^2 \tag{10}$$

where $\mathbf{u}^X \in \mathbb{R}^n$, and $\mathbf{u}^Y \in \mathbb{R}^n$. It is important to point out that vectors $\mathbf{u}^X$ and $\mathbf{u}^Y$ in (10) can be properly selected in order to obtain several desired properties of functional H.

Proposition 1 states that if X is hyperplanar, the minimization of F is equivalent to solving n uncoupled overdetermined linear systems of rank $n-1$. Thus, we have n extra degrees of freedom which allow us to find a solution of the minimization of F such that it maps $\mathbf{u}^X$ to $\mathbf{u}^Y$.

Using again the coordinates introduced by Knupp in [1]: $\overline{\mathbf{x}} = \mathbf{x} - \mathbf{c}^X$ and $\overline{\mathbf{y}} = \mathbf{y} - \mathbf{c}^Y$, we can write

$$H(\mathbf{A}) = \sum_{i=1}^{m} \left\| \overline{\mathbf{y}}^i - \mathbf{A}\overline{\mathbf{x}}^i \right\|^2 + \left\| \mathbf{u}^Y - \mathbf{A}\mathbf{u}^X \right\|^2 .$$

Therefore, we are looking for a linear mapping $\mathbf{A}$ such that it approximately transforms, in the least-squares sense, $\overline{X} = \{\overline{\mathbf{x}}^i\}_{i=1,\dots,m}$ into $\overline{Y} = \{\overline{\mathbf{y}}^i\}_{i=1,\dots,m}$, and $\mathbf{u}^X$ to $\mathbf{u}^Y$ (see Fig. 7).

We will prove that the minimization of H, opposite to the minimization of functionals F and G, always leads to a set of normal equations with full rank when X is hyperplanar.

Proposition 3 *If X is a hyperplanar set and $\mathbf{u}^X \notin \mathbb{H}$, then the minimization of functional H is equivalent to solving n uncoupled overdetermined linear systems of rank n .*
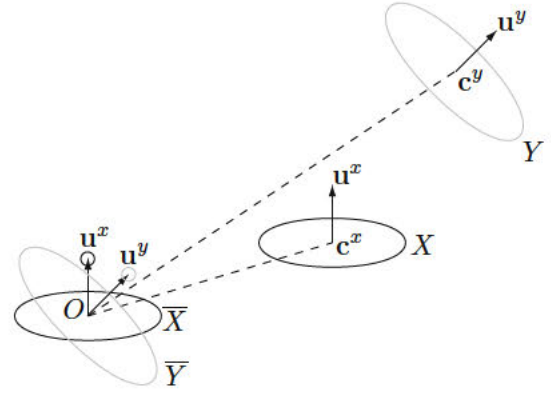
Fig. 7 Geometric representation of the translation of sets X and Y to the origin and the vectors $\mathbf{u}^X$ and $\mathbf{u}^Y$

Proof Similar to the proofs of the previous propositions, the minimization of H leads to n uncoupled overdetermined linear systems. In this case we define

$$\mathbf{X} := \begin{pmatrix} x_1^1 - c_1^X & \cdots & x_1^m - c_1^X & u_1^X \\ \vdots & & \vdots & \vdots \\ x_n^1 - c_n^X & \cdots & x_n^m - c_n^X & u_n^X \end{pmatrix}$$

and

$$\mathbf{Y} := \begin{pmatrix} y_1^1 - c_1^Y & \cdots & y_1^m - c_1^Y & u_1^Y \\ \vdots & & \vdots & \vdots \\ y_n^1 - c_n^Y & \cdots & y_n^m - c_n^Y & u_n^Y \end{pmatrix}.$$

By Lemma 2 $\dim \mathbb{H} = \dim \mathrm{span}(\mathbf{x}^1 - \mathbf{c}^X, \dots, \mathbf{x}^m - \mathbf{c}^X) = n - 1$. Since $\mathbf{u}^X \notin \mathbb{H}$, we conclude that $\mathrm{rank}\, \mathbf{X}^T = \dim \mathrm{span}(\mathbf{x}^1 - \mathbf{c}^X, \dots, \mathbf{x}^m - \mathbf{c}^X, \mathbf{u}^X) = n$. $\qquad\square$

In addition, we will also prove that if X is non-hyperplanar and we set $\mathbf{u}^X = \mathbf{0}$, then the minimization of H is also equivalent to solving n uncoupled and full-rank overdetermined systems.

Proposition 4 *If X generates a linear variety of dimension n (whole $\mathbb{R}^n$), $\mathbf{u}^X = \mathbf{0}$ and $\mathbf{u}^Y \in \mathbb{R}^n$, then the minimization of H is equivalent to solving n uncoupled overdetermined systems of rank n.*

Proof Under these assumptions the minimization of H leads to n uncoupled overdetermined linear systems. In this case we define

$$\mathbf{X} := \begin{pmatrix} x_1^1 - c_1^X & \cdots & x_1^m - c_1^X & u_1^X \\ \vdots & & \vdots & \vdots \\ x_n^1 - c_n^X & \cdots & x_n^m - c_n^X & u_n^X \end{pmatrix}$$

and

$$\mathbf{Y} := \begin{pmatrix} y_1^1 - c_1^Y & \cdots & y_1^m - c_1^Y & u_1^Y \\ \vdots & & \vdots & \vdots \\ y_n^1 - c_n^Y & \cdots & y_n^m - c_n^Y & u_n^Y \end{pmatrix}.$$

By hypothesis X generates a linear variety of dimension n. Thus, rank $\mathbf{X}^T = $ rank $\mathbf{X} = n$. $\qquad\square$

Remark 7 From Propositions 3 and 4, we can conclude the minimization of H has one and only one solution. Note that we do not consider sets of points X that generate linear varieties of dimension less than $n - 1$. For instance, in $\mathbb{R}^3$ we do not consider source surfaces which degenerate to lines or points, because it does not make sense to sweep them in practical applications.

Remark 8 Vectors $\mathbf{u}^X$ and $\mathbf{u}^Y$ are parameters of functional H. In our implementation we have selected them as

- X hyperplanar and Y hyperplanar: $\mathbf{u}^X = \mathbf{n}^X$ and $\mathbf{u}^Y = \mathbf{n}^Y$.
- X hyperplanar and Y non-hyperplanar: $\mathbf{u}^X = \mathbf{n}^X$ and $\mathbf{u}^Y = \tilde{\mathbf{n}}^Y$.
- X non-hyperplanar and Y hyperplanar: $\mathbf{u}^X = \mathbf{0}$ and $\mathbf{u}^Y = \mathbf{0}$.
- X non-hyperplanar and Y non-hyperplanar: $\mathbf{u}^X = \mathbf{0}$ and $\mathbf{u}^Y = \mathbf{0}$,

where $\tilde{\mathbf{n}}^Y$ is a measure of the normal vector to Y. However, further research has to be developed in order to prove the proper selection of vectors $\mathbf{u}^X$ and $\mathbf{u}^Y$.

Remark 9 In several applications the source and the target boundaries are not affine. Therefore, it is not possible to obtain an affine transformation that exactly maps X to Y. In these situations an additional smoothing step is required in order to improve the quality of the final mesh. Hence, our goal is also to obtain a good initial inner node location in order to decrease the number of iterations in the smoothing step. We claim that the minimization of functional H provides better node location than the minimization of functionals F and G. Moreover, this projection algorithm may provide an excellent initial guess for morphing procedures [2].

5 Numerical examples

In order to assess the advantages and drawbacks of the proposed functionals used to obtain affine transformations, four examples are presented. These examples are obtained with a sweeping tool that implements the minimizations of functionals F, G, and H. To highlight the behavior of the three functionals, in these examples the inner meshes are obtained projecting directly from the source surface to the inner layers. That is, we have neither used a weighted projection algorithm from both cap surfaces (which we use in practical applications [4]) nor an additional smoothing step to improve the quality of the final mesh. To solve the overdetermined linear systems that do not have full rank, we use a singular value decomposition which supplies the solution with the smallest norm. The set of points X corresponds to the boundary nodes of the source mesh, and the set of points Y corresponds to the boundary nodes of the current inner layer. In all examples, the source surface has a planar boundary, with non-planar interior. Observe that we have selected source surfaces with planar boundaries in order to force a minimization of functional F that leads to a set of normal equations with singular system matrix. Moreover, the minimization of functional G is only used for source surfaces with planar boundaries. However, the target surface may be planar or not. Also, in all the examples, the boundary of the source surface is not parallel to the loops of the inner layers. Note that if they were parallel, the minimization of functional G will not produce the skewness effect presented in Remark 6.

In the first example (see Fig. 8), a C-shaped geometry with circular cross sections is presented. The boundary nodes of the source surface, X, and the boundary nodes of the inner layers, Y, are planar. However, the inner part of the source mesh has curvature. For the minimization of each functional, two views of nine hexahedra layers are provided. The left column is a general view, and the right column is a detail of the fourth, fifth, and sixth layers of hexahedra. When we minimize functional F, by Proposition 1, we know that the overdetermined linear system matrix does not have full rank. This implies that the obtained inner layers become flat, despite the source surface has curvature (see Fig. 8a). The minimization of G generates inner layers that present curvature on the inner part. However, due to the shape of the geometry the skewness effect also appears (see Remark 6). Note that as $\mathbf{c}^Y - \mathbf{c}^X$ tends to the plane defined by X, the skewness effect is more pronounced (see Fig. 8b). In the limit, when the inner layer is on the same plane that the boundary of the source surface, a degenerated projection is obtained (the minimization of G leads to a overdetermined linear system with rank deficient matrix, see Proposition 2). Finally, if we minimize functional H, then the nodes of the inner layers of hexahedra have the desired location and curvature (see Fig. 8c). In this example, we see that the minimization of H provides the best location for inner nodes.

The goal of the second example (see Fig. 9) is to show that the minimization of H provides a better initial inner node location when an additional smoothing step is required. To this end, a square is swept along a semi-circle. The source surface is a planar square with curvature in the inner part, whereas the target surface has a curved boundary. Thus, the inner layers are defined by non-planar

Fig. 8 Projection of a non planar source surface mesh with planar boundary onto planar inner layers. **a** Minimizing F; **b** minimizing G; and **c** minimizing H


(a)

(b)

(c)
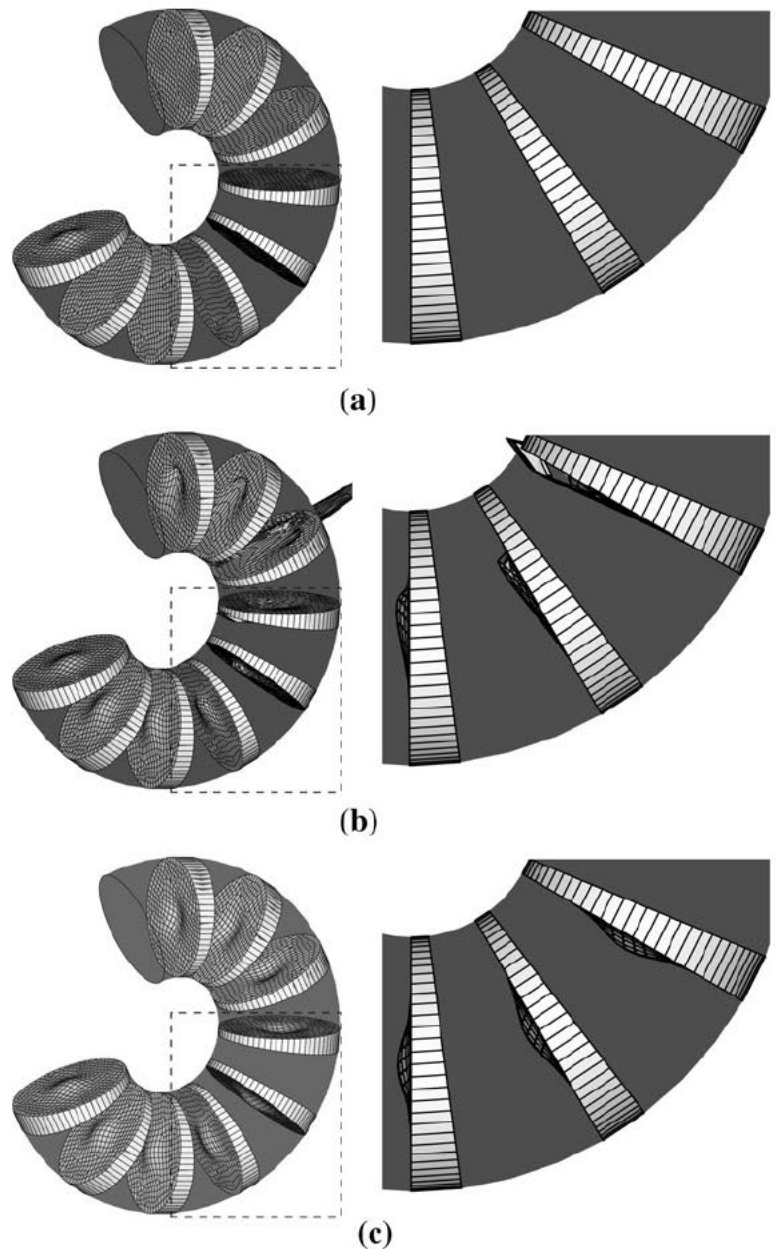
loops of nodes that become more curved near the target surface. Four layers of hexahedra are shown from two different points of view. The left column is a general view, and the right column is a detail of the third and fourth layers of hexahedra. Observe that the source and the target boundaries are not affine. For this reason, the hexahedral elements with nodes on the linking sides present an undesired slope (see Fig. 9). Therefore, an additional smoothing step will be required (see Remark 4). Like in the first example, minimizing F we obtain layers of hexahedra with flat inner part (see Fig. 9a). The minimization of G produces non-flat inner layers of hexahedra, but the skewness effect also appears (see Fig. 9b). Finally, minimizing H we obtain an affine transformation that preserves

the curvature of the source surface. Since the source and the target surfaces are not affine, the computed solution also presents a slope near the boundary (see Fig. 9c). Note that the best initial inner node location is the one provided by the minimization of functional H.

The goal of the third example (see Fig. 10a) is to show that the skewness effect introduced by the minimization of functional G may also appear for very simple sweep paths. In particular, this example presents the discretization of an extrusion volume defined by varying cross-sections along a straight and skewed sweep path. These cross-sections are elliptical-shaped with different size, and only the middle of the extrusion path become circular. The source surface is not flat and has a planar boundary, whereas the target

Fig. 9 Projection of a non planar source surface mesh with planar boundary onto non planar inner layers. **a** Minimizing F; **b** minimizing G; and **c** minimizing H
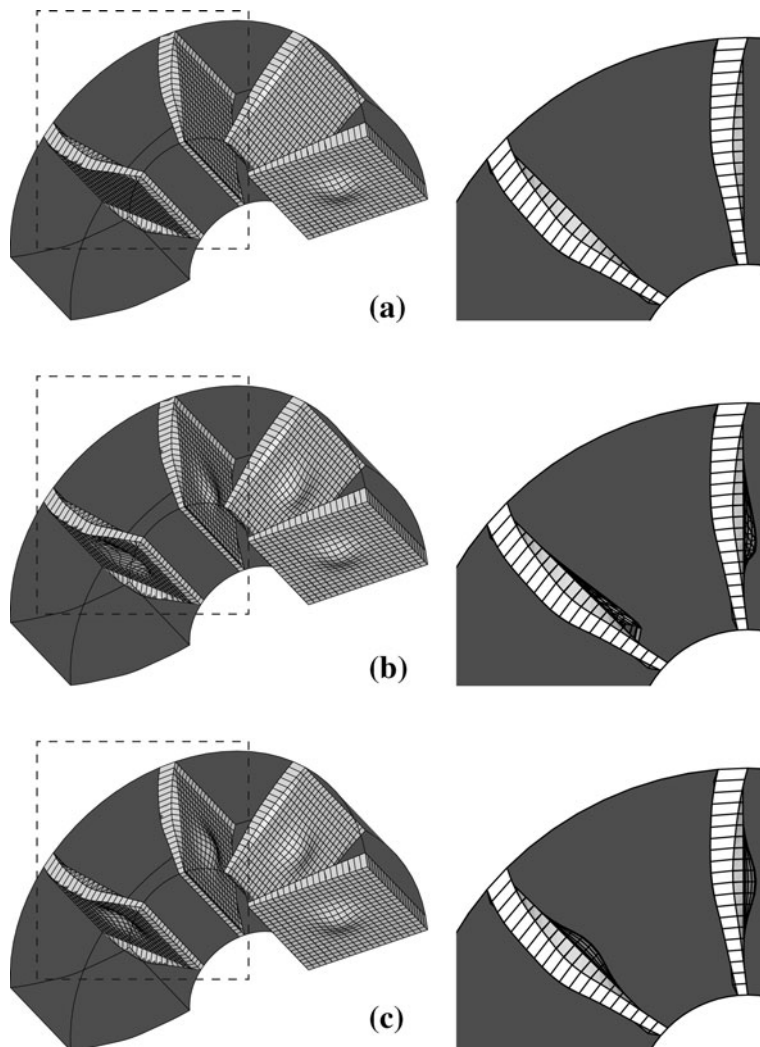
surface is planar. For the minimization of each functional, a view of four inner layers of hexahedra elements are presented. Figure 10b shows that the minimization of F leads to flat inner layer of elements since the boundary of the source surface is planar. As expected, the minimization of G generates non-planar inner layers of hexahedra. However, the shape of the source surface is not well preserved (see the skewness effect in the inner layers presented in Fig. 10c). Note that this effect is more pronounced close to the target surface. As in the previous examples, the minimization of H leads to the desired solution (see Fig. 10d).

The goal of the last example (see Fig. 11a) is to show that if the source and the target surfaces are not affine, the minimization of H provides a better node location than the one obtained with the minimization of F and G (even in the case of geometries simpler than the volume presented in the second example). In this example we discretize an extrusion volume defined by varying cross-sections along a straight and skewed sweep path. The source surface is a planar square with curvature in the inner part. The target surface is planar

and its boundary is defined by four arcs. Hence, both surfaces have planar boundaries but not mutually affine. Moreover, the inner layers are defined by planar loops of nodes that become more curved close to the target surface. Note that source surface boundary is not affine to the inner loops of nodes. Since the source surface is planar, and similar to the previous examples, the minimization of F generates planar inner layers of hexahedral elements (see Fig. 11b). The minimization of G produces skewed layers of elements (see Fig. 11c). Finally, the minimization of H preserves the original shape of the source surface and provides the best initial configuration for the smoothing algorithm (see Fig. 11d).

6 Concluding remarks

In this paper we have presented a comparative analysis of several functionals that have been extensively used to project meshes in sweeping procedures. We first stated that the minimization of functional F leads to a set of normal

Fig. 10 Projection of a non planar source surface mesh with planar boundary onto varying shape planar inner layers.
a Extrusion volume to mesh; **b** minimizing F; **c** minimizing G; and **d** minimizing H
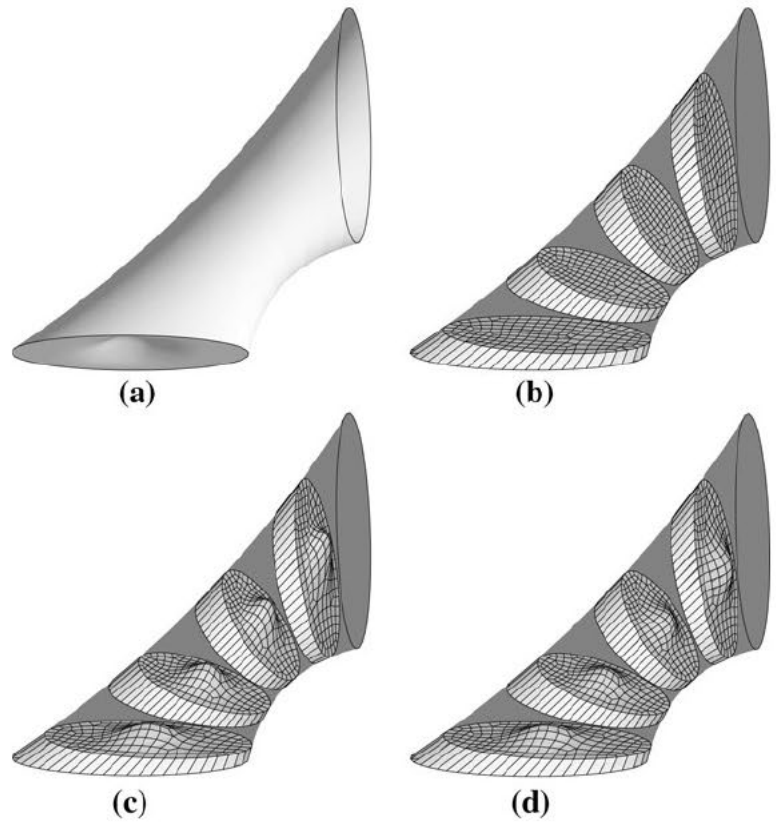
Fig. 11 Projection of a non planar source surface mesh with planar boundary onto varying shape planar inner layers. The inner loops are not affine to the source surface boundary.
a Extrusion volume to mesh; **b** minimizing F; **c** minimizing G; and **d** minimizing H
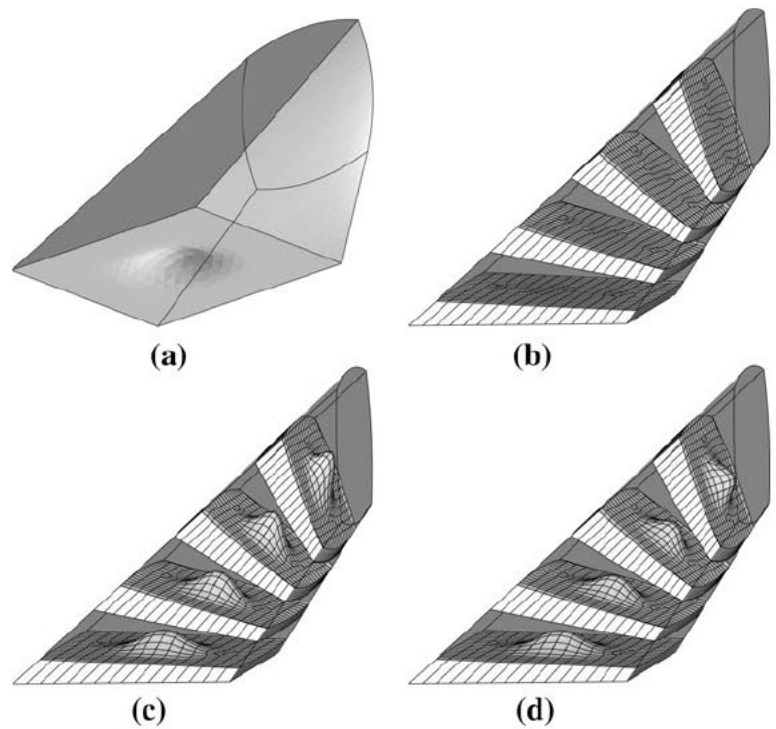
equations with singular system matrix if the source set of points are hyperplanar. We have also shown that the minimization of G leads to normal equations with singular system matrix if, in addition, $\mathbf{c}^Y - \mathbf{c}^X$ lies in the same hyperplane that X. Moreover, we have seen that the minimization of G also introduces a skewness effect when the set of points X is hyperplanar, and the sets X and Y are not parallel.

Finally, in order to overcome the previous drawbacks we have proposed the functional H. We have also proved that the minimization of H has one and only one solution. Thus, sets of normal equations with singular system matrix are avoided. Furthermore, if $\mathbf{u}^X$ and $\mathbf{u}^Y$ are properly selected, the minimization of functional H is preferable since it is not affected by the skewness introduced by the minimization of functional G and tends to preserve the shape of the original source surface. Therefore, it provides suitable node location for the inner layers. In addition, it supplies an excellent initial guess for the position of the inner nodes if an additional smoothing step is required.

References

1. Knupp PM (1998) Next generation sweep tool: a method for generating all hex meshes on two and one half dimensional geometries. In: 7th International meshing roundtable, pp 505 513
2. Knupp PM (1999) Applications of mesh smoothing: copy, morph, and sweep on unstructured quadrilateral meshes. Int J Numer Meth Eng 45:37 45
3. Staten ML, Canann SA, Owen SJ (1999) BMSweep: locating interior nodes during sweeping. Eng Comput 15:212 218
4. Roca X, Sarrate J, Huerta A (2004) Surface mesh projection for hexahedral mesh generation by sweeping. In: 13th International meshing roundtable, pp 169 179
5. Blacker T (1996) The Cooper tool. In: 7th International meshing roundtable, pp 13 30
6. Mingwu L, Benzley SE (1996) A multiple source and target sweeping method for generating all hexahedral finite element meshes. In: 5th International meshing roundtable, pp 217 225
7. Miyoshi K, Blacker T (2000) Hexahedral mesh generation using multi axis Cooper algorithm. In: 9th International meshin groundtable, pp 89 97
8. Blacker TD, Stephenson MB (1991) Paving a new approach to automated quadrilateral mesh generation. Int J Numer Meth Eng 32:811 847
9. Cass RJ, Benzley SE, Meyers RJ, Blacker TD (1996) Generalized 3 D paving: an automated quadrilateral surface mesh generation algorithm. Int J Numer Meth Eng 39:1475 1489
10. Sarrate J, Huerta A (2000) Efficient unstructured quadrilateral mesh generation. Int J Numer Meth Eng 49:1327 1350
11. Sarrate J, Huerta A (2000) Automatic mesh generation of non structured quadrilateral meshes over curved surfaces in $\mathbb{R}^3$. In: 3th ECCOMAS conference
12. Thompson JF, Soni B, Weatherill N (1999) Handbook of grid generation. CRC Press, Boca Raton
13. Scott MA, Earp MA, Benzley SE, Stephenson MB (2004) Adaptive sweeping techniques. In: 7th International meshing roundtable, pp 417 432
14. Goodrich D (1997) Generation of all quadrilateral surface mes hers by mesh morphing. MA thesis, Brigham Young University, Utah
15. Gill PE, Murray W, Wright MH (1991) Numerical linear algebra and optimization. Addison Wesley, Edwood City
16. Lawson C, Hanson R (1974) Solving least squares problems. Prentice Hall, Englewood Cliffs