International Journal of Applied Mathematics

Volume 31 No. 6 2018, 833-852
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v31i6.12

THE SOLUTION OF FREE-FORM DEFORMATION PROBLEM USING PSEUDOINVERSE

Jana Prochazkova[1 §], Jiri Novak[2]
[1,2]Brno University of Technology
Technicka 2, Brno – 616 69, CZECH REPUBLIC

Abstract: The basis of many commonly used computer graphics algorithms is concealed in well known algebraic structures. Our interest is focused on a method Free Form Deformation (FFD) used in the areas such as surface modeling, image registration or animation. The algebraic core can be described by the undetermined system of equations and pseudoinverse or Lagrange multipliers are perfect tools to solve it. Mutual interconnection is not obvious, so we deduce the fundamental theorems and we present FFD method and its application on general B-spline surfaces.

AMS Subject Classification: 93E24, 65F20
Key Words: free-form deformation, least square method, pseudoinverse, B-spline, NURBS, Lagrange multipliers

1. Introduction

FFD is a method that is primarily designed to direct modification of geometric structures, most often tensor product surfaces. These surfaces (e.g. Bzier, B-spline, Non-Uniform Rational B-Spline – NURBS) are set by a topologically rectangular set of control points, degree and knot vectors (in case of spline). FFD recomputes control points positions depending on the surface point(s) movement to ensure that the resulting surface passes throw new positioned points. The application of FFD is in many graphics areas such as surface modeling [28], image registration [34, 8, 35], animation [21], morphing [18], etc.

The FFD transformation leads to the solution of an undetermined system

Received: September 13, 2018

[§] Correspondence author

of equations (see [31], [1]). We use the pseudoinverse as the computational tool to solve the best results. We also show that this solution corresponds to the Lagrange multipliers (Section 2.3).

Section 2.1 of our article deals with the definition of spline function and spline surfaces. In Section 2.2, we introduce the basic definitions and properties related to the pseudoinverse and the pseudoinverse solution for the linear system of equations. Section 3 covers the main idea of FFD with the proof of the key theorem.

2. Theoretical background

2.1. B-spline

The theory of piecewise polynomial B-spline functions and following B-spline curves and surfaces is well known and precisely described in the literature, e.g. [23], [26]. The main idea is based on the combination of the B-spline basis function with a regular grid of control points and pair of knot vectors. The knot vector is a sequence of numbers provided that each one is greater than or equal to the preceding one. The knot vectors as control points describe the shape of result surface [24]. Knot vectors are generally placed into one of three categories: uniform, open uniform, and non-uniform. The algorithm FFD is independent on the type of knot vector.

In our work, we apply on non-uniform B-spline surfaces. The usage of point weights (B-spline becomes a rational B-spline) only transfers the computational problem to homogenous coordinates in projective space [13]. So that we work without the rational extension. The following section provides the brief overview of B-spline basic definitions and the example of the basis B-spline functions is in Figure 1.

Definition 1. Let $U = (u_0, \dots, u_m)$ be a knot vector ($u_i \leq u_{i+1}$, $i = 0, \dots, m-1$). Then the i-th B-spline basis function of degree p is defined recurrently as:

$$N_{i,0} = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases},$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \tag{1}$$

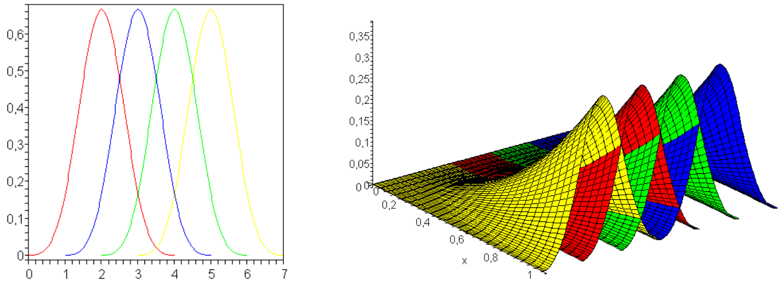

Figure 1: B-spline basis functions $N_{i,3}(u)$ and spline functions $N_{i,3}(u)N_{j,3}(v)$.

Definition 2. Let $\{\mathbf{P}_i\}$ be the control points and $\{w_i\}$, $i = 0, \ldots, n$ be the weights, and the $N_{i,p}(u)$ be $p-$th degree B-spline basis functions. Then the NURBS curve is defined as:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i}. \tag{2}$$

NURBS surface is defined as a tensor product of NURBS curves. Every NURBS topology needs the regular $k + 1 \times l + 1$ control points net P_{ij}, weights w_{ij}, $i = 0, 1, \ldots, k$, $j = 0, 1, \ldots, l$, knot vectors $U = (u_0, \ldots, u_m)$, $V = (v_0, \ldots, v_n)$ and degrees p, q. Then the NURBS surface can be expressed as:

$$S(u,v) = \frac{\sum_{i=0}^{k} \sum_{j=0}^{l} N_{i,p}(u) N_{j,q}(v) w_{ij} \mathbf{P}_{ij}}{\sum_{i=0}^{k} \sum_{j=0}^{l} N_{i,p}(u) N_{j,q}(v) w_{ij}}. \tag{3}$$

If the weights of all points are equal to one, we get a non-uniform B-spline surface and afterwards we can rewrite Eq. (3) as:

$$S(u,v) = \sum_{i=0}^{k} \sum_{j=0}^{l} N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{ij} \tag{4}$$

and Eq. (2) as a non-uniform B-spline curve:

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) \mathbf{P}_i. \tag{5}$$

Let assume the generalization of B-spline surface called T-spline with irregular control point lattice. All the advantages of splines as the local modification scheme or control point insertion remain preserved. The general idea of T-spline was published by Sederberg [29]. Generally, the arbitrary surface point is computed with the combination of the control points and B-spline basis functions, only the knot vectors are not global but connected unambiguously with every control point. The description of this knot vectors is done by T-mesh in (s, t) parameter space. T-mesh often contains T-junctions (similar to letter T). This special type is a vertex which is shared by one s-edge and two t-edges, or by one t-edge and two s-edges.

T-spline surface of degree p is defined by control points $P_i,\ i = 1, \ldots, n$. Every point is connected with two knot vectors of length $2p-1$ derived from T-mesh and we set degree $p = 3$. It is also possible to add weights to the control points to make weighted T-spline surface [19].

The mathematical expression of T-spline is:

$$P(s,t) = \sum_{i=1}^{n} P_i B_i^3(s,t). \tag{6}$$

The basis functions $B_i^3(s,t)$ are given by

$$B_i^3(s,t) = N[s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}](s) N[t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}](t).$$

where $N[s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}](s)$ is B-spline basis function associated with the knot vector

$$s_i = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}]$$

and $N[t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}](t)$ is associated with the knot vector

$$t_i = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}].$$

2.2. Inverse, pseudoinverse

This section gives the definitions and theorems of pseudoinverse that are essential to subsequent parts (more details in [3]).

Definition 3. Let $A \in \mathcal{C}^{m \times n}$ be a general matrix. We say that A is left invertible if there exists an $n \times m$ matrix C such that $CA = I_n$. We call C a left inverse of A. We say that A is right invertible if there exists an $n \times m$ matrix D such that $AD = I_m$. We call D a right inverse of A. We say that A is invertible if A is both left invertible and right invertible.

Remark 1. Let $A \in \mathbb{C}^{m,n}$ be a general matrix and A^* its Hermitian transpose matrix, then:

$$r(A^* \cdot A) = r(A \cdot A^*) = r(A). \tag{7}$$

Note that r is the rank of the matrix [33].

Theorem 4. *Let $A \in \mathbb{C}^{m,n}$. The following claims are valid:*

1. *A is right (left) invertible,*

2. *A is full row (column) rank,*

3. *$A \cdot A^*$ is regular.*

Proof. **(1)-(2):** Let A be a $m \times n$ matrix. Suppose, that (1) holds and B is the right inverse of A. It holds that $A \cdot B = I_m$. Let m be a rank of $A \cdot B$:

$$m = r(A \cdot B) \leq r(A) \leq m.$$

(2)-(3): Let $r(A) = m$ and due to Remark 1, we have

$$r(A \cdot A^*) = r(A) = m$$

and matrix $A \cdot A^*$ is regular.

(3)-(1): If $A \cdot A^*$ is regular, let $B = A^*(A \cdot A^*)^{-1}$. Then $A \cdot B = I_m$ and B is right inverse of matrix A.

Definition 5. Let $A \in \mathbb{C}^{m,n}$. The matrix A^+ is the pseudoinverse of A if:

1. $AA^+A = A$,

2. $A^+AA^+ = A^+$,

3. $(AA^+)^* = AA^+$,

4. $(A^+A)^* = A^+A$.

Furthermore, the pseudoinverse A^+ always exists and is unique. The proof is e.g. in [1], [22].

Theorem 6. *Let $A \in \mathbb{C}^{m,n}$, $m \geq n$ with full row rank. The matrix $A \cdot A^*$ is regular and invertible and the pseudoinverse of A is:*

$$A^+ = A^* \cdot (A^* \cdot A)^{-1} \text{ and } A^+ \cdot A = I_n. \tag{8}$$

Proof. Suppose that the matrix A is $m \times n$, where $m < n$ and A is full row rank. Due to Remark 1, we get $m = r(A) = r(A \cdot A^*)$ and consequently the matrix $A \cdot A^*$ is regular. Let

$$X = A^* \cdot (A \cdot A^*)^{-1}.$$

Then $A \cdot X = A \cdot A^* \cdot (A \cdot A^*)^{-1} = I_m$. The Penrose conditions (1)-(3) holds. The condition (4) can be proved by:

$$(X \cdot A)^* = A^* \cdot X^* = A^* \cdot \left((A \cdot A^*)^{-1}\right)^* \cdot A =$$

$$= A^* \cdot ((A \cdot A^*)^*)^{-1} \cdot A = A^* \cdot (A \cdot A^*)^{-1} \cdot A = X \cdot A.$$

Analogously, this theorem can be formulated for full column rank matrices.

Remark 2. If $A \in \mathbb{R}^{m,n}$, the Theorem 6 can be rewritten as:

$$A^+ = A^T \cdot \left(A^T \cdot A\right)^{-1}. \tag{9}$$

2.2.1. Pseudoinverse solution for linear system of equations

We define Hermitian inner product and the norm of the vector X in this section. Generally, we work in complex domain but we need only real domain for the application in FFD.

Definition 7. Let $X, Y \in \mathbb{C}^n$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

We define the Hermitian inner product as

$$(X, Y) = Y^\star \cdot X = \sum_{i=1}^{n} x_i \cdot \overline{y}_i \tag{10}$$

and the norm of the vector X is defined as:

$$\|X\| = \sqrt{X, X} = \sqrt{\sum_{i=1}^{n} x_i \cdot \overline{x}_i} = \sqrt{\sum_{i=1}^{n} |x_i|^2}. \tag{11}$$

Definition 8. We say that two vectors X, Y are orthogonal if $(X, Y) = 0$. We can write

$$X \perp Y.$$

Theorem 9 (Pythagorean theorem for vectors)**.** *For orthogonal n-dimensional vectors X, Y we have:*

$$\|X + Y\|^2 = \|X\|^2 + \|Y\|^2. \tag{12}$$

The proof is done by simple equivalent operations of expression $\|X + Y\|^2$.

Theorem 10. *Let $A \in \mathbb{C}^{m,n}$, $P = A \cdot A^+$, $Q = A^+ \cdot A$, X, Y are n-dimensional vectors in $\mathbb{C}^n$. It holds:*

(a) $\|A \cdot X + (I_m - P) \cdot Y\|^2 = \|A \cdot X\|^2 + \|(I_m - P) \cdot Y\|^2$,

(b) $\|A \cdot Y + (I_m - Q) \cdot X\|^2 = \|A \cdot Y\|^2 + \|(I_m - Q) \cdot X\|^2$.

Proof. Let $R = A \cdot X$ and $S = (I_m - P) \cdot Y$ are m-dimensional vectors and we can write:

$$(R, S) = (A \cdot X, (I_m - P) \cdot Y) = ((I_m - P) \cdot Y)^\star \cdot A \cdot X$$
$$= Y^\star \cdot (I_m - P) \cdot A \cdot X = Y^\star \cdot A \cdot X - Y^\star \cdot A \cdot A^+ \cdot A \cdot X$$
$$= Y^\star \cdot A \cdot X - Y^\star \cdot A \cdot X = 0.$$

The vectors R, S are orthogonal and the theorem holds due to Pythagorean theorem for vectors. Statement (b) can be proved analogously. □

The system of linear equations is given by

$$A \cdot X = B, \tag{13}$$

where

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ & \vdots & \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

and $X = \begin{pmatrix} x_1 \\ x_n \end{pmatrix}$ $B = \begin{pmatrix} b_1 \\ b_m \end{pmatrix}$, with $a_{ij}, b_j \in \mathbb{C}$.

The following Theorem 11 explains the least squares solution to the system (13) with minimum norm [7]. This solution is X_0, given by the expression:

$$X_0 = A^+ \cdot B. \tag{14}$$

Theorem 11. *Let $X \in \mathbb{C}^n$, $X \neq X_0$ be n-dimensional vector, then we get*

$$\|A \cdot X_0 - B\| \leq \|A \cdot X - B\|. \tag{15}$$

In case $\|A \cdot X_0 - B\| \leq \|A \cdot X - B\|$ it holds

$$\|X_0\| < \|X\|. \tag{16}$$

Proof. The proof is based on rewriting of Theorem 10. A proof based on ortogonality is in [1]. □

2.3. Lagrange problem

Let $A \cdot X = B$ be a general linear system of m equations with n unknowns where $n > m$ (undetermined system) and $a_{ij}, x_j b_j \in \mathbb{R}$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. Generally, according to the Rouch-Capelli theorem, the rank in an undetermined system is necessarily less than the number of unknowns, there are undoubtedly an infinitude of solutions.

Nevertheless, we want to minimize the norm $\|X\|$ under the system of constraints $AX = B$. We introduce a vector λ of Lagrange multipliers and consider the function

$$\mathcal{L} = X^T X - \lambda^T (AX - B). \tag{17}$$

The solution using derivatives for all variables x_i and λ gives the system (18), (19):

$$2X^T - \lambda^T A = \mathbf{o} \tag{18}$$

$$-(AX - \mathbf{B}) = \mathbf{o}. \tag{19}$$

After doing some simplifications of Eq. (18), we get:

$$X = \frac{1}{2} A^T \lambda. \tag{20}$$

Substituting Eq. (20) to Eq. (19) we get:

$$\frac{1}{2} A A^T \lambda = B. \tag{21}$$

Suppose that the matrix A is full row rank so that the matrix AA^T due to Theorem 4 is regular matrix and the inverse exists. The Eq. (21) can be rewritten as follows:

$$\begin{aligned} (AA^T)^{-1}AA^T\lambda &= 2(AA^T)^{-1}B \\ \lambda &= 2(AA^T)^{-1}B. \end{aligned} \tag{22}$$

Substituting Eq. (22) to Eq. (20) and using Remark 2 we can write:

$$\begin{aligned} X &= \frac{1}{2}2\underbrace{A^T(AA^T)^{-1}}_{A^+}B \\ X &= A^+B. \end{aligned} \tag{23}$$

Here we can clearly see the relation between pseudoinverse and the solution based on minimization.

3. FFD theory and computation

The FFD method was firstly published by Sederberg and Perry in the year 1986. The principle is based on the insertion of the solid representation to the regular parallelpiped; subsequent transformation recomputes the position of the solid points with respect to the modified points of the lattice [28].

The computation is based on the tensor product trivariate Bernstein polynomial. Every object point $\mathbf{X}$ has (s,t,u) coordinates in the coordinate system on a parallelpiped region:

$$\mathbf{X} = \mathbf{X}_0 + s\mathbf{S} + t\mathbf{T} + u\mathbf{U}. \tag{24}$$

Let $\mathbf{P}_{ijk}$, $i = 0,\dots,l$, $j = 0,\dots,m$, $k = 0,\dots,n$ are control points on a lattice. The deformation is represented by a movement of the control points P_{ijk} from their latticial positions. New position of point $\mathbf{X}_{new}$ is computed by:

$$\mathbf{X}_{new} = \tag{25}$$

$$= \sum_{i=0}^{l}\binom{l}{i}(1-s)^{l-i}s^i\left[\sum_{j=0}^{m}\binom{m}{j}(1-t)^{m-j}t^j\left(\sum_{k=0}^{n}\binom{n}{k}(1-u)^{n-k}u^k\mathbf{P}_{ijk}\right)\right],$$

where (s,t,u) are parameters in Eq. eqrefeq:1.

Sederberg's method was improved by the other methods in the following years. The traditional FFD computes the final shape after translating the control points from their positions. The choice of the lattice points put the base of the other FFD types. The limitation for the rectangular shape of the control lattice is eliminated for example by the usage of Dirichlet surfaces (defined in [9]). The authors in [21] introduce the Dirichlet FFD (or DFFD) that works with so-called Sibson coordinates that describe arbitrary location inside the convex hull of the point set as a linear combination of its Delaunay neighbors in the set. Point-based (PB) FFD [20] uses ellipsoidal radial basis function to define the deformation space. This approach does not need Delaunay triangulation as the DFFD.

Another solution to overcoming the rectangular topology limitation of tensor-product volumes is based on subdivision surfaces. The authors in [10] claim that it is the bridge between the discrete control mesh and the smooth surface.

The pioneering method was proposed in [6]. The CatmullClark subdivision volume [4] is used as the intermediate deformation space and it guarantees that the topology of the control lattice can be arbitrary. It makes the deformation more flexible. Since the subdivision volume is defined by the recursive refinement rules, the refinement procedure convergence is problematic and thus the smoothness of the deformation space is not guaranteed. Furthermore, control lattice generation is not easy and the storage and computational costs are high. The method [10] brings low storage and computational costs and more intuitive and simpler control. It presents the construction of subdivision surface defined by control mesh spans. The object is embedded into space by the nearest point rule. The object shape is changed accordingly to the change of the shape of the control mesh. The computational and storage cost are between $O(n^2)$ and $O(n^3)$.

An extension of subdivision surface usage with a path of deformation and an influence area constraints is described in [17]. The main advantage of this method is that it uses only vertices of an object and satisfies the geometrical constraints provided by the user.

The subdivision scheme uses only regular control lattice that covers the object. There is a possibility to employ irregular subdivision T-lattice that is based on T-splines [29, 19]. The authors in [32] proposed a two-stage algorithm that generates the multiresolution lattices for weighted T-spline volumes and processes the local or global deformation for arbitrary shape by combining weights change and lattice manipulation. However, this method has some limitation that is based on the irregular lattice, e.g. it cannot operate one row of the control lattice in some directions and the computational costs are also higher.

It is obvious that all the developmental methods lead to the method that ensures the global and local deformation ability, storage and computational costs, simplicity, independence of object representation. It is very difficult to find a deformation method to satisfy all of these criteria. It is important to note that from the user point-of-view, it does not matter if the transformed object is 3-dimensional, embedded with subdivision surface or a parametric surface, the user always uses the same process to model a complex object: load an initial object from a library and deform it via FFD methods to follow his demand needs [25].

Direct movement of the surface points is the next level of FFD controlling. The main idea is the direct shift of the surface points and the control net points are recomputed automatically. The users no longer need to understand the tensor-product Bezier surfaces or NURBS to use the FFD for modeling. This method satisfies most of the conditions: it is intuitive, versatile, simple to shape modification, it needs low memory and the computation is fast.

The process of direct manipulation was firstly described in [15]. It employs a pseudo-inverse (generalized inverse) matrix that as [16] mentioned involves complicated calculations. The authors in [16] suggests the approach based on a constrained optimization without the need of pseudoinverse computation. These two approaches are very different at first sight but they describe the same problem. In the following sections, we try to investigate mutual interconnection of these two solutions. We put emphases on the comprehension and detailed description of these two methods to formulate suitable apparatus for the following computer processing.

The problem of FFD is well known but it still offers the up to date applications in many areas, most in car, ships or aircraft manufacturing and also in design. For example, [5, 14, 2] reconnect the main idea of Sederberg's FFD manipulation with suitable apparatus like ICP algorithm or subdivision surfaces for smart ship hull constructions. An interesting design application is a non-realistic modeling where two 3D geometric models are merged to preserve a shape silhouette from different views [27, 30].

3.1. Free Form Deformation

Without loss of generality we suppose that the degree of the spline curves and surfaces is $p = q = 3$. The generalization for arbitrary degree is possible but the curves and surfaces are often in cubic form.

Let ω_{ij} be the control net of points $\mathbf{P}_{ij}$. Arbitrary point $\mathbf{Q}$ on the spline surface can be expressed as:

$$\mathbf{Q}_{ij}(u,v) = \sum_{i=0}^{k}\sum_{j=0}^{l} N_{i,3}(u)N_{j,3}(v)\mathbf{P}_{ij}. \tag{26}$$

We set $k, l = 3$ because of the local control property of B-spline functions. The movement of one point affects only $(p+1) \times (q+1)$ control points, $p = q = 3$ so that we recompute 4×4 control points (Fig. 2, left). Then, we can rewrite Eq.(26) as:

$$\mathbf{Q}_{ij}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} N_{i,3}(u)N_{j,3}(v)\mathbf{P}_{ij}, \tag{27}$$

or symbolically

$$\mathbf{Q} = \mathbf{NP}, \tag{28}$$

where $\mathbf{N}$ is a row vector (matrix 1×16):

$$\mathbf{N} = (N_k)_{k=0}^{16} \quad \text{where} \quad N_{4i+j+1} = N_i(u)N_j(v),$$

$$i, j = 0, \ldots, 3,$$

and $\mathbf{P}$ is a 16×3 matrix that contains the $x, y, z-$coordinates of the control points,

$$\mathbf{P} = (P_{ij}^{x}, P_{ij}^{y}, P_{ij}^{z},)_{i,j=0}^{3}.$$

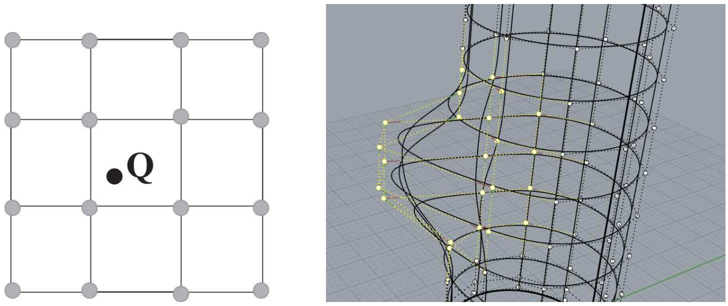

Figure 2: Affected control point net 4×4 in the case of surface point Q movement (left), FFD transformation on the cylinder (right).

The main idea of FFD is how to determine the displacement of the control points to ensure the correct transformation (Figure 2, right). This is solved in Theorem 12.

Theorem 12. *Let $\mathbf{Q}_s$ be a source point and $\mathbf{Q}_t$ be a target point. Similarly, the $\mathbf{P}_s$ denotes the source control net of points and $\mathbf{P}_t$ is the matrix of transformed points. Result displacement $\Delta\mathbf{P}$ of the control points $\mathbf{P}_t$ is equal to:*

$$\Delta\mathbf{P} = \mathbf{N}^+ \cdot \Delta\mathbf{Q} \tag{29}$$

where $\Delta\mathbf{Q} = \mathbf{Q}_t - \mathbf{Q}_s$, $\Delta\mathbf{P} = \mathbf{P}_t - \mathbf{P}_s$.

Proof. We can write the coordinates of the source and target point $\mathbf{Q}_s$, $\mathbf{Q}_t$ as:

$$\mathbf{Q}_s = \mathbf{N}\mathbf{P}_s, \tag{30}$$

$$\mathbf{Q}_t = \mathbf{N}\mathbf{P}_t. \tag{31}$$

We know that $\Delta\mathbf{Q} = \mathbf{Q}_t - \mathbf{Q}_s$ and $\Delta\mathbf{P} = \mathbf{P}_t - \mathbf{P}_s$. The difference Eq. (30) – Eq. (31)can be written as:

$$\Delta\mathbf{Q} = \mathbf{N}\Delta\mathbf{P}. \tag{32}$$

We use the scheme of optimization described in Section 2.3 and we get the solution

$$\Delta\mathbf{P} = \mathbf{N}^+ \cdot \Delta\mathbf{Q}. \tag{33}$$

□

Remark 3. The computation of $\mathbf{N}^+$ is specific so that we mention the detailed construction here. The matrix $\mathbf{N} \in \mathbb{R}^{1,16}$ and its pseudoinverse $\mathbf{N}^+ = \mathbf{N}^T$ is 16×1. It follows that $\mathbf{N}\mathbf{N}^T$ is only a real number and corresponds to the dot product. It is clearly visible that

$$(\mathbf{N}\mathbf{N}^T)^{-1} = \frac{1}{\sum_{k=1}^{16} N_k^2} \tag{34}$$

and the pseudoinverse N^+ can be computed as:

$$N^+ = \frac{1}{\sum_{k=1}^{16} N_k^2} N^T \Delta\mathbf{Q} \tag{35}$$

and the displacement of the control points $\mathbf{P}$ are:

$$\Delta\mathbf{P} = \left(\frac{N_i \Delta\mathbf{Q}}{\sum_{k=1}^{16} N_k^2} \right)_{i=1}^{16} \tag{36}$$

$$= \left(\frac{N_i \Delta\mathbf{Q}^x}{\sum_{k=1}^{16} N_k^2}, \frac{N_i \Delta\mathbf{Q}^y}{\sum_{k=1}^{16} N_k^2}, \frac{N_i \Delta\mathbf{Q}^z}{\sum_{k=1}^{16} N_k^2} \right)_{i=1}^{16}.$$

Remark 4. The Eq. (36) is often used in the programming applications. Figures 3 and 4 show the planar and surface application of FFD technique.

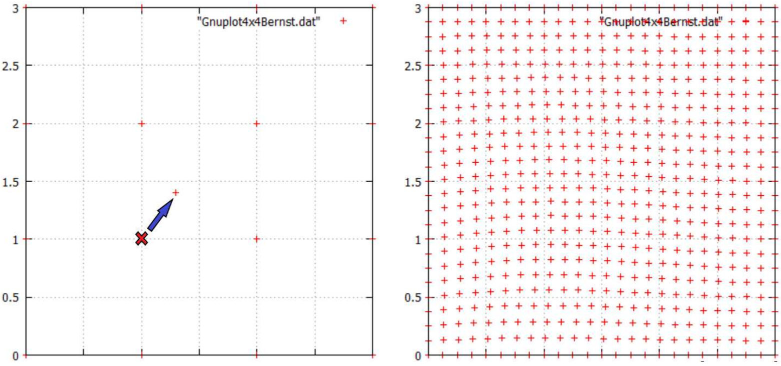

Figure 3: Single transformation of regular control net and resulting surface.

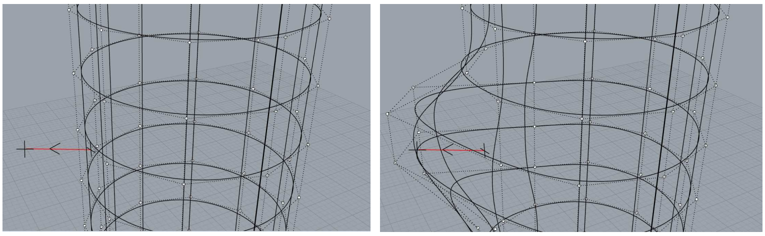

Figure 4: General surface point movement (left) and resulting transformation of control point net (right).

The interesting question is one-to-one property between original and transformed spline surface. Theorem 13 was firstly presented in Goodman [12]. The condition gives the system of $3m(m+1)+2n(n+1)$ linear inequations, so that the computation time increases rapidly with higher number of control points. We have to note that the condition is sufficient not necessary.

Theorem 13. *The function given in* (27) *is one-to-one if* $|\Delta P|_\infty \leq 0.48$.

Proof. The proof is in [18]. □

The condition in the previous theorem gives the condition for suitable behavior of the surface modifications, however, in the case that the points are dragged beyond their volume of the influence, it sometimes generates wild hyperpatch distortion. Using the subsequent small injective deformations [11] prevents the self-intersections. The article presents a set of theoretical conditions based on Jacobian of FFD and its evaluations and weak condition using conic-hull hodograph test.

Remark 5. NURBS surface FFD works with the same apparatus as spline surfaces. We only use the homogeneous coordinates to attach the point weights. Let $\mathbf{Q}^T = \left(\mathbf{Q}_x^T, \mathbf{Q}_y^T, \mathbf{Q}_z^T, \mathbf{Q}_w^T\right)$ be the target point and $\mathbf{Q}^S = \left(\mathbf{Q}_x^S, \mathbf{Q}_y^S, \mathbf{Q}_z^S, \mathbf{Q}_w^S\right)$ be a source point. The difference vector is $\Delta\mathbf{Q} = \mathbf{Q}^T - \mathbf{Q}^S = \left(\mathbf{Q}^X, \mathbf{Q}^Y, \mathbf{Q}^Z, \mathbf{Q}^W\right)$. We can rewrite Eq. (36) as:

$$\Delta\mathbf{P} = \left(\frac{N_i \Delta\mathbf{Q}}{\sum_{k=1}^{16} N_k^2}\right)_{i=1}^{16} \tag{37}$$

$$= \left(\frac{N_i \Delta\mathbf{Q}^x}{\sum_{k=1}^{16} N_k^2}, \frac{N_i \Delta\mathbf{Q}^y}{\sum_{k=1}^{16} N_k^2}, \frac{N_i \Delta\mathbf{Q}^z}{\sum_{k=1}^{16} N_k^2}, \frac{N_i \Delta\mathbf{Q}^w}{\sum_{k=1}^{16} N_k^2}\right)_{i=1}^{16}$$

$$= \left(\frac{\frac{N_i \Delta\mathbf{Q}^x}{\sum_{k=1}^{16} N_k^2}}{\frac{N_i \Delta\mathbf{Q}^w}{\sum_{k=1}^{16} N_k^2}}, \frac{\frac{N_i \Delta\mathbf{Q}^y}{\sum_{k=1}^{16} N_k^2}}{\frac{N_i \Delta\mathbf{Q}^w}{\sum_{k=1}^{16} N_k^2}}, \frac{\frac{N_i \Delta\mathbf{Q}^z}{\sum_{k=1}^{16} N_k^2}}{\frac{N_i \Delta\mathbf{Q}^w}{\sum_{k=1}^{16} N_k^2}}, 1\right)_{i=1}^{16}.$$

3.2. Multiple point constraints

Suppose that we get a set of h point constraints and we search for the final displacement of the control points. The solution is clearly described in following theorem.

Theorem 14. *A direct manipulation of FFD with h point constraints can be decomposed into h manipulations with single point constraints.*

Proof. Let $\mathbf{Q}_s^1, \ldots, \mathbf{Q}_s^h$ are source points and $\mathbf{Q}_t^1, \ldots, \mathbf{Q}_t^h$ are corresponding target points. The proof is based on the decomposition of the shift vectors $\Delta\mathbf{Q}_l = \Delta\mathbf{Q}_t^l - \Delta\mathbf{Q}_s^l$ to the sequence: $\Delta^{(l)}\mathbf{Q}_l$, where $l = 1, \ldots, h$. We compute the values of the control point displacement $\Delta^{(l)}\mathbf{P}$ in one step as in Eq. (36):

$$\Delta^{(l)}\mathbf{P} = \left(\frac{N_i \Delta^{(l)}\mathbf{Q}}{\sum_{k=1}^{16} N_k^2}\right)_{i=1}^{16}. \tag{38}$$

The global displacement of the control points **P** is the sum of the previous partial transformation.

$$\Delta \mathbf{P} = \sum_{l=1}^{h} \left(\frac{N_i \Delta^{(l)} \mathbf{Q}}{\sum_{k=1}^{16} N_k^2} \right)_{i=1}^{16} . \tag{39}$$

□

The illustration of FFD transformation with two subsequent point constraints is in Figure 5. We make subsequent transformations of two arbitrary point of spline surface: red graph is initial control net, blue graph is first deformation, green graph is second deformation.

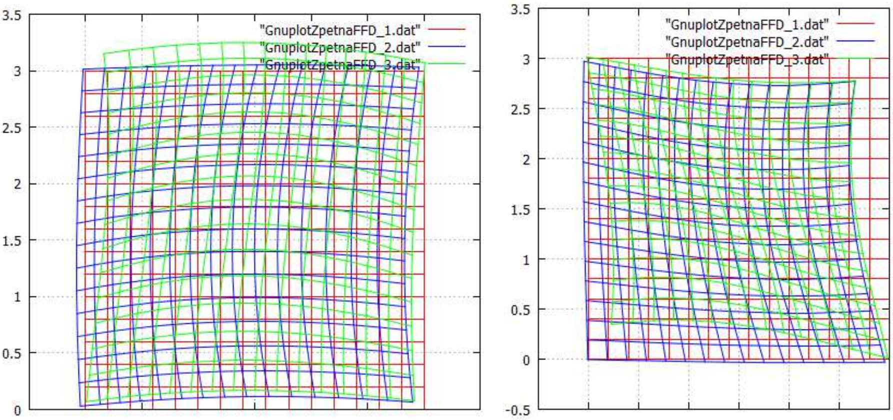

Figure 5: Multiple point constraints: red graph is initial control net, blue graph is first deformation, green graph is second deformation.

4. Conclusion

The main gist of our article is the description of the pseudovinverse interesting application in the solution of FFD problems. We show an elegant way of computation and we solve the problem of direct modification of spline surfaces. In Section 2.3, we show the solution of undetermined system of equations using Lagrange multipliers and the interconnection to pseudoinverse. We also implemented the theoretical results from Section 3.1 on spline surfaces given by regular or irregular control points net.

We describe multiple point constraints problem as the composition of single point constraints in Section 3.2.

It is also possible to work with rational splines – NURBS (Section 2.1) by adding the weights to control points. The weight means the influence of given point to the resulting shape of the surface. The presented FFD theorem holds, only the homogeneous coordinates (in projective space) should be used instead of Euclidean ones.

References

[1] J.C.A. Barata, M.S. Hussein, The Moore–Penrose pseudoinverse: A tutorial review of the theory, *Brazilian Journal of Physics*, **42**, No 1 (2012), 146–165.

[2] R. Bearee, J.Y. Dieulot, P. Rabate, An innovative subdivision-ICP registation method for tool-path correction applied to deformed aircraft parts machining, *International Journal of Advanced Manufacturing Technology*, **53** (2011), 463–471.

[3] A. Ben-Israel, T.N.E. Greville, *Generalized Inverses, Theory and Applications*, Springer-Verlag, New York, Second Edition (2003).

[4] E. Catmull, J. Clark, Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer Aided Design*, **9**, No 6 (1978), 183–188.

[5] A. Coppede, G. Vernengo, D. Villa, A combined approach based on Subdivision Surface and Free Form Deformation for smart ship hull form design and variation, *Ships and Offshore Structures* (2018), 769–778.

[6] R. MacCracken, Kenneth I. Joy, Free-form deformations with lattices of arbitrary topology, In: *Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, ACM, New York (1996), 181–188.

[7] P.J. Davis, *Circulant Matrices*, John Wiley and Sons Inc., New York, Second Edition (1994).

[8] T. Eckhard, J. Eckhard, E. M. Valero, J.L. Nieves, Nonrigid registration with free-form deformation model of multilevel uniform cubic B-splines: application to image registration and distortion correction of spectral image cubes, *Appl. Opt.*, **53** (2014), 3764–3772.

[9] G. Farin, Surface over Dirichlet tessellations, *Geometric Aided Design*, **7** (1990), 281–292.

[10] J. Feng, J. Shao, X.Jin, Q. Peng, A.R. Forrest, Multiresolution free-form deformation with subdivision surface of arbitrary topology, *Visual Computer*, **22**, No 1 (2006), 28–42.

[11] J.E. Gain, N.A. Dodgson, Preventing self-intersection under free-form deformation, *IEEE Transactions on Visualization and Computer Graphics*, **7**, No 4 (2001), 289–298.

[12] T.N.T. Goodman, K. Unsworth, Injective bivariate maps, *Annals of Numerical Mathematics*, **3** (1996), 91-104.

[13] A. Holme, *Projective Space, Geometry: Our Cultural Heritage*, Springer, Berlin Heidelberg (2002), 221–232.

[14] J. Hradil, *Adaptive Parameterization for Aerodynamic Shape Optimization in Aeronautical Applications*, Brno University of Technology, Faculty of Mechanical Engineering, Institute of Aerospace Engineering, Brno (2015).

[15] W.M. Hsu, J.F. Hughes, H. Kaufman, Direct manipulation of Free-Form Deformations, In: *ACM SIGGRAPH Computer Graphics*, **26**, No 2 (1994), 177–184.

[16] S.M. Hu, H. Zhang, C.L Tai, et al., Direct manipulation of FFD: efficient explicit solutions and decomposible multiple point constraints, *Visual Computer*, **17**, No 6 (2001), 370–379.

[17] S. Lanquetin, R. Raffin, M. Neveu, Curvilinear constraints for free form deformations on subdivision surfaces, *Mathematical and Computer Modeling*, **51**, No 34 (2010), 189–199.

[18] S. Lee, G. Woberg, K.-Y. Chwa, S.Y. Shin, Image metamorphosis with scattered feature constraints, *IEEE Transactions on Visualization and Computer Graphics*, **2**, No 4 (1996), 337–354.

[19] L. Liu, Y.J. Zhang, X. Wei, Weighted T-splines with application in reparameterizing trimmed NURBS surfaces, *Computer Methods in Applied Mechanics and Engineering*, **295** (2015), 108–126.

[20] K.T. McDonnell, H. Qin, A Point-based technique for Free-Form deformation, *J. Graphics Tools*, **12**, No 3 (2007), 25–41.

[21] L. Moccozet, N.M. Thalmann, Dirichlet Free-Form deformations and their application to hand simulation. In: *Proc. of the Computer Animation (CA 97)*, IEEE Computer Society, Washington, DC (1997), 93–102.

[22] R. Penrose, A generalized inverse for matrices, *Mathematical Proc. of the Cambridge Philosophical Society*, **51**, No 3 (1955), 406–413.

[23] L. Piegl, W. Tiller, *NURBS Book*, Springer-Verlag, Berlin Heidelberg (1997).

[24] J. Prochazkova, D. Prochazka, *An Influence of Knot Vectors on the Shape of NURBS Surfaces*, VUT Brno (2007), 95–96.

[25] R. Raffin, Free form deformations or deformations non-constrained by geometries or topologies, *Lecture Notes in Computational Vision and Biomechanics*, **7** (2013), 49–74.

[26] D. Rogers, *An Introduction to NURBS*, Morgan-Kaufmann, Burlington (2000).

[27] J. Snchez-Reyes, J.M. Chacn, Anamorphic Free-Form deformation, *Computer Aided Geom. Des.*, **46** (2016), 30–42.

[28] T.W. Sederberg, S.R. Parry, Free-form deformation of solid geometric models. In: *Proc. of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*, ACM, New York (1986), 151–160.

[29] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T–NURCCS, *ACM Transactions on Graphics*, **22**, No 3 (2003), 477–483.

[30] G. Sela, G. Elber, Generation of view dependent models using free form deformation, *The Visual Computer*, **23**, No 3 (2007), 219–229.

[31] I.R. Shafarevich, A. Remizov, *Linear Algebra and Geometry*, Springer-Verlag, Berlin Heidelberg (2012).

[32] W. Song, X. Yang, Free-form deformation with weighted T-spline, *Vis. Comput.*, **21** (2005), 139–151.

[33] G. Strang, *Introduction to Linear Algebra*, 4th ed., Cambridge Press, Wellesley (2009).

[34] N.J. Tustison, B.B. Avants, J.C. Gee, Directly manipulated Free-Form deformation image registration, In: *IEEE Transactions on Image Processing*, **18**, No 3 (2009), 624–635.

[35] R. Van De Walle, H.H. Barrett, K.J. Myers, M.I. Aitbach, B. Desplanques, A.F. Gmitro, J. Cornelis, I. Lemahieu, Reconstruction of MR images from data acquired on a general nonregular grid by pseudoinverse calculation, *IEEE Trans. Med. Imag.* **19** (2000), 1160-1167.