Communications in Applied and Industrial Mathematics, DOI: 10.1685/journal.caim.461
ISSN 2038-0909, e-461

Constrained reconstruction of 3D curves and surfaces using integral spline operators

Franca Caliò[1], Edie Miglio[2]

[1] *Dipartimento di Matematica, Politecnico di Milano, Italy*

[2] *MOX - Dipartimento di Matematica, Politecnico di Milano, Italy*

Communicated by Roberto Natalini

Abstract

In the context of direct/reverse engineering processes one of the main problem is the reconstruction of curves and surfaces starting from a cloud of points. Most of the times the (re)constructed curves and surfaces have to satisfy some particular geometric constraints and functional properties related to the desired shapes.

In this paper, referring to 3D curves and surfaces, we propose an algorithm based on an interpolatory variation diminishing integral spline operator characterized by the presence of shape parameters. In order to choose the best value for the shape parameters different functionals can be adopted.

Some test cases are presented in order to show the effectiveness of the proposed algorithm: both academic and real world test cases are considered.

Keywords: surface and curve approximation, spline approximation, functional minimization.

AMS subject classification: 65D10, 65D07, 65D17.

1. Introduction.

In the context of reverse modeling and/or creative design one of the main problem is to (re)construct a virtual model of an object starting from a cloud of points. This amounts to determine a suitable parametrization, only depending on one or two parameters, of the required shape which follows the given data and is "optimal" with respect to specific requirements given by the designer. To this aim it is necessary to develop geometric models and algorithms that automatically create shapes close as much as possible to the given data and assuring at the same time a good representation of the geometric and functional properties of the required virtual model.

In literature some methods for spatial curves have been proposed (see for example [1–4]): these methods are based on the use of free form along with a least square approach. The resulting (re)constructed curves are in general shape preserving (see for example [5,6]), but not sufficiently flexible

Received on July 22nd, 2013. Accepted on December 11th, 2013. Published on December 12th, 2013.

to ensure desired properties. Similar approaches based on tensor models are used for surfaces (see for example [7–9]).

The alternative idea, that we present in this paper, is to reconstruct a free form curve controlled by a set of points whose position is chosen in order to preserve the shape and to satisfy geometrical properties of (re)constructed curves imposed by designer. In [10] a similar approach has been used for planar curve fairing. This idea is extended to surfaces as well.

The main ingredients of the proposed algorithm are:

- the observed points are considered as interpolation points;
- starting from the interpolation points the corresponding control points are generated;
- a parameter λ is inserted in the definition of the control points;
- a suitable discrete functional is minimized in order to obtain the optimal value of the parameter (λ_{opt});
- λ_{opt} is used in the new optimal control point set.

This scheme is easily extended to surface as well. Finally the "optimal" free form is defined and the curve or the surface is reconstructed: this reconstructed surface satisfy the required properties.

Hence the key idea of the advocated method is to stick to an interpolating approach but allowing to move the control points by means of the "proper" determination of the value of the shape parameters. This proper determination is linked to the choice of the functional to be minimized in order to obtain the optimal value of the shape parameters. In the case of curve most of the time the requirement is to obtain a smooth curve in terms of curvature and torsion, with different weights of these requirements on the shape of curve; as for surfaces a suitable curvature control is usually required.

The paper is organized as follows: the first section summarizes the form and properties of the particular spline operator with shape parameter used in the free form (re)construction; in the second section the tensor extension of this operator is presented; the third section describes some possible functionals both for curves and surfaces; the fourth section describes the used algorithms and the fifth section presents some numerical examples.

2. Integral spline operator.

2.1. *3D curves.*

In this section we briefly recall the main concepts about the so called Variational Diminishing Integral Spline operator with shape parameter

DOI: 10.1685/journal.caim.461

(λ-VDIS). For details see [11] and for an application in the reverse engineering field see for example [10].

Given a set of *control points* $\mathbf{CP}$, an integer number k and a knot vector $\mathbf{t} = (t_i)_{i=-k}^{m+1}$, $0 = t_{-k} = ... = t_0 < t_1 < ... < t_{p+1} = ... = t_{m+1} = 1, \quad p = m - k$ the operator λ-VDIS is defined as follows:

$$(T_m^{\lambda}\mathbf{CP})(u) = (1-\lambda)(S_m\mathbf{CP})(u) + \lambda(T_m\mathbf{CP})(u) \tag{1}$$

where

- λ is a real number (shape parameter) such that $0 \leq \lambda \leq 1$;
- $(S_m\mathbf{CP})(u) = \mathbf{b}_{mk}^T(u)\mathbf{CP} \quad 0 \leq u \leq 1$ is the so called a k-degree Variation Diminishing Spline (VDS) operator, where:

  $$\mathbf{b}_{mk} = [B_0^k(u), B_1^k(u), ..., B_m^k(u)]^T$$

  and the basis functions $B_i^k(u)$ $(i = 0, 1, \ldots m)$ are the classical *B-spline* functions of degree k, on the knot vector $\mathbf{t}$;
- $(T_m\mathbf{CP})(u) = \mathbf{b}_{mk}^T(u)(\mathcal{M}\mathbf{CP}) \qquad 0 \leq u \leq 1$ is called a k-degree Variation Diminishing Integral Spline (see [11]) where

$$\mathcal{M} = \begin{bmatrix} \beta_0 & \gamma_0 & 0 & ... & 0 \\ \alpha_1 & \beta_1 & \gamma_1 & ... & 0 \\ 0 & \alpha_2 & \beta_2 & ... & 0 \\ 0 & ... & ... & & \gamma_{m-1} \\ 0 & ... & ... & \alpha_m & \beta_m \end{bmatrix} \tag{2}$$

with

$$\begin{aligned} &\alpha_0 = 0, \qquad \alpha_i = \frac{(\delta_i^l)^2}{2\Delta_{i-1}^k \Delta_i^{k+1}}, \; i = 1, ...m \\ &\gamma_i = \frac{(\delta_i^r)^2}{2\Delta_i^k \Delta_i^{k+1}}, \quad i = 0, ..., m-1, \quad \gamma_m = 0 \\ &\beta_i = 1 - \alpha_i - \gamma_i, \; i = 0, ..., m \end{aligned} \tag{3}$$

$$\begin{aligned} \Delta_i^k &= \xi_{i+1}^k - \xi_i^k, \\ \delta_i^r &= \xi_{i+1}^{k+1} - \xi_i^k, \; \xi_i^{k+1} < \xi_i^k < \xi_{i+1}^{k+1} \\ \delta_i^l &= \xi_i^k - \xi_i^{k+1}, \end{aligned} \tag{4}$$

and

$$\xi_i^k = \frac{t_{i-k+1} + ... + t_i}{k} \qquad i = 0, \ldots, m. \tag{5}$$

are the so-called Schoenberg or corresponding points (see [12]).

Alternatively (1) can be written as

$$(T_m^{\lambda}\mathbf{CP})(u) = (S_m\mathbf{CQ}^{\lambda})(u) \tag{6}$$

where

$$\mathbf{CQ}^{\lambda} = (1-\lambda)\mathbf{CP} + \lambda\mathcal{M}\mathbf{CP}. \tag{7}$$

This operator shares the same geometrical properties of the VDS operator (see [13]). Precisely some of them:

- affine invariance;
- convex hull property;
- variation diminishing.

2.2. *Surfaces.*

Now we extend the previous concepts about λ-VDIS operator to the field of splines depending on two parameters (u, w).

Let $\mathbf{CP}$ a set of $(m+1)\times(n+1)$ control points, h and k integer numbers and
$\mathbf{t} = (t_i)_{i=-k}^{m+1}$, $0 = t_{-k} = ... = t_0 < t_1 < ... < t_{p+1} = ... = t_{m+1} = 1$, $p = m-k$,
$\mathbf{s} = (s_i)_{i=-h}^{n+1}$, $0 = s_{-h} = ... = s_0 < t_1 < ... < s_{q+1} = ... = s_{n+1} = 1$, $q = n-h$.

The $l-th$ component of bivariate tensor VDS operator can be expressed in matrix form similarly to univariate case (see [13]):

$$\begin{aligned}(T_{mn}^{\lambda\tau}\mathbf{CP})_l(u,w) = {} & (1-\tau)(1-\lambda)(S_{mn}\mathbf{CP})_l + \tau(1-\lambda)(S_{mn}\mathbf{CQ}^{\tau})_l \\ & + \lambda(1-\tau)(S_{mn}\mathbf{CQ}^{\lambda})_l + \lambda\tau(S_{mn}\mathbf{CQ}^{\lambda\tau})_l\end{aligned} \tag{8}$$

with $l = 1, 2, 3$ and where

- λ and τ are real numbers (shape parameters) such that $0 \le \lambda, \tau \le 1$;
- $(S_{mn}\mathbf{CP})(u,w)_l = \mathbf{b}_{mk}^T(u)(\mathbf{CP})_l\mathbf{b}_{nh}(w) \quad 0 \le u \le 1, \quad 0 \le w \le 1$;
- $\mathbf{b}_{mk} = [B_0^k(u), B_1^k(u), ..., B_m^k(u)]^T$ and $\mathbf{b}_{nh} = [B_0^h(w), B_1^h(w), ..., B_n^h(w)]^T$.

The basis functions $B_i^k(u) \quad (i = 0, 1, \ldots, m)$ are the classical *B-spline* functions of degree k built on the set of knots $\mathbf{t}$ and the basis functions $B_j^h(w)$ $(j = 0, 1, \ldots, n)$ are the classical *B-spline* functions of degree h built on the set of knots $\mathbf{s}$

- $(\mathbf{CQ}^{\lambda})_l = \mathcal{M}(\mathbf{CP})_l$;
- $(\mathbf{CQ}^{\tau})_l = \mathcal{N}(\mathbf{CP})_l$;

DOI: 10.1685/journal.caim.461

- $(\mathbf{CQ}^{\lambda\tau})_l = \mathcal{M}(\mathbf{CP})_l\mathcal{N}$;

where $\mathcal{M}$ is the matrix defined in (2) and $\mathcal{N}$ is the analogous matrix built upon the knot vector $\mathbf{s}$.

Alternatively (8) can be written as:

$$T^{\lambda\tau}_{mn}(\mathbf{CP})_l(u,w) = S_{mn}(\mathbf{CP})^{\lambda\tau}_l(u,w) \tag{9}$$

where:

$$\begin{aligned}(\mathbf{CP})^{\lambda\tau}_l = (1-\tau)(1-\lambda)(\mathbf{CP})_l + \tau(1-\lambda)\mathcal{N}(\mathbf{CP})_l \\ + \lambda(1-\tau)\mathcal{M}(\mathbf{CP})_l + \lambda\tau\mathcal{M}(\mathbf{CP})_l\mathcal{N}\end{aligned} \tag{10}$$

Also in this case some interesting geometrical properties hold. We experimentally can show the analogous properties of the curves. We are working about these theoretical results.

3. Adopted functionals.

Before describing in details the approximating procedure we will discuss some functionals that can be used in order to achieve different aims. A discussion on the meaning of different functionals can be found in [14–16].

From the mathematical point of view different results can be obtained using different functionals involving curve properties.

3.1. *3D curve functionals.*

In Table 1 some possible functionals and their properties are presented; κ(t), $\rho(t)$ and $\tau(t)$ are the curvature, the radius of curvature and the torsion and are defined as follows:

$$\kappa(t) = \frac{\|\mathbf{c}' \wedge \mathbf{c}''\|}{\|\mathbf{c}'\|^3}, \tag{11}$$

$$\rho(t) = \kappa(t)^{-1}, \tag{12}$$

and

$$\tau(t) = \frac{(\mathbf{c}' \wedge \mathbf{c}'' \cdot \mathbf{c}''')}{\|\mathbf{c}' \wedge \mathbf{c}''\|^2} \tag{13}$$

where $\mathbf{c}(t)$ is the parametric representation of the curve, the $'$ denotes the derivative with respect to the parameter t and $\wedge$ is the vector product.

Table 1. Different functionals.

Functional	$\mathbb{R}^2$	$\mathbb{R}^3$
$\int [\rho^2\tau^2 + (\rho')^2]^{\frac{1}{2}} ds$	smooths	smooths and rounds
$\int [\kappa^2\tau^2 + (\kappa')^2]^{\frac{1}{2}} ds$	smooths	smooths and flattens
$\int [\kappa^2 + \tau^2]^{\frac{1}{2}} ds$	straightens	flattens
$\int (\tau''(s))^2 + (\kappa''(s))^2 ds$	smooths	smooths and rounds

3.2. *Surface functionals.*

As for the curves we introduce some metrics which are used to measure the smoothness of the surface at interest. These metrics must depend only on invariants such that a reparametrization of the surface does not affect the value of the measure. Let K and H be the Gaussian and the mean curvatures of the surface and $\mathbf{n}$ the normal vector to the surface; the following three derived surfaces are considered as metrics:

1. the *flattening metric* which is the surface area of the derived surface

$$\mathbf{d}(u,w) = K(u,w)\mathbf{n}(u,w). \tag{14}$$

   This metric tends to minimize the magnitude of the Gaussian curvature K and extreme changes in the Gaussian curvature along the lines of curvature: the tendency is to flatten the surface.
2. the *rounding metric* which is the surface area of the derived surface

$$\mathbf{d}(u,w) = \mathbf{r}(u,w) + [H(u,w)/K(u,w)]\mathbf{n}(u,w). \tag{15}$$

   This metric tends to pull the surface towards a sphere.
3. the *rolling metric* which is the surface area of the derived surface

$$\mathbf{d}(u,w) = [K(u,w) + H^2(u,w)]\mathbf{n}(u,w). \tag{16}$$

   This metric tends to make the surface more cylindrical or conical.

The functionals used in the algorithm we are proposing are obtained computing the surface area A of the above mentioned derived surface *i.e.*

$$A = \int_R \left| \frac{\partial \mathbf{d}}{\partial u} \wedge \frac{\partial \mathbf{d}}{\partial w} \right| du dw, \tag{17}$$

where R is the parameter domain of $\mathbf{r}(u,w)$.

DOI: 10.1685/journal.caim.461

3.3. *Discrete curvatures and discrete torsion.*

In the problems we are going to consider the curve will be defined by points hence it is necessary to be able to compute the curvature and the torsion starting from a discrete set of points defining the curve (see for example [17,18]). To this aim let us consider Figure 1.

A simple formula to approximate the curvature is as follows:

$$K(t_i) = \frac{2A_i}{L_i L_{i+1} Q_i} \tag{18}$$

where A_i, L_i, Q_i are defined as follows ($\mathbf{r}_i^h$ denotes the discrete representation of the curve):

$$\begin{aligned} &\mathbf{L}_i = \mathbf{r}_i^h - \mathbf{r}_{i-1}^h \ , \ L_i = \|\mathbf{L}_i\| \ , \\ &\mathbf{Q}_i = \mathbf{r}_{i+1}^h - \mathbf{r}_{i-1}^h \ , \ Q_i = \|\mathbf{Q}_i\| \ , \\ &A_i = \|\mathbf{L}_i \wedge \mathbf{L}_{i+1}\| = \|\mathbf{L}_i \wedge \mathbf{Q}_i\| = \|\mathbf{L}_{i+1} \wedge \mathbf{Q}_i\|. \end{aligned} \tag{19}$$

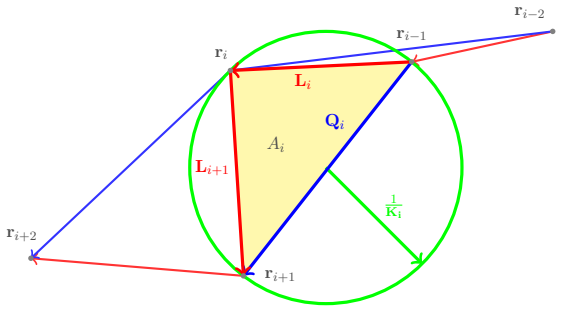

Figure 1. Notations for the computation of the discrete curvature and the discrete torsion.

Similarly the following formula for the discrete torsion can be obtained

$$T(t_i) = \frac{V_i}{B_i B_{i+1}} \tag{20}$$

where:

$$\begin{aligned} &V_i = (\mathbf{L}_{i-1} \wedge \mathbf{L}_i \cdot \mathbf{L}_{i+1}) \\ &\mathbf{B}_i = \mathbf{L}_i \wedge \mathbf{L}_{i+1} \ , \ B_i = \|\mathbf{B}_i\|. \end{aligned} \tag{21}$$

A more useful centered approximation (with respect to t_i) of the discrete torsion can also be derived:

$$T^{sym}(t_i) = \frac{L_i T_{i+1} + L_{i+1} T_i}{L_i + L_{i+1}}. \tag{22}$$

As for the surfaces we have to compute the discrete Gaussian and mean curvatures. Suppose to have a scattered cloud of point in the space defining the surface we want to (re)construct. The first step is to build a 3D Voronoi diagram of the points such that for each point we can define the quantities shown in Figure 2.

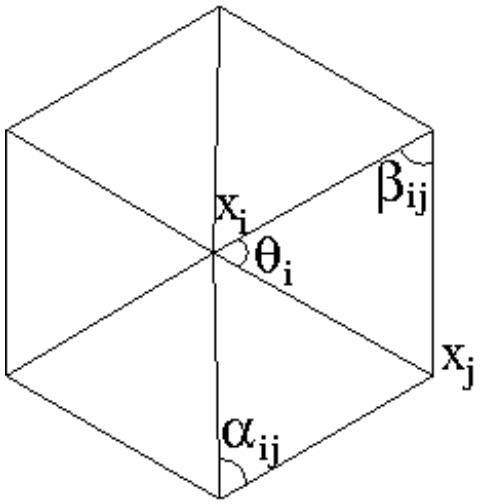

Figure 2. Notations used in the definition of the discrete curvatures for a surface.

Now we can compute the mean curvature normal operator $\mathbf{K}$ (see [19, 20]) using the following expression:

$$\mathbf{K}(\mathbf{x}_i) = \frac{1}{2A_p} \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(\mathbf{x}_i - \mathbf{x}_j), \tag{23}$$

where A_p is a suitable area surrounding point $\mathbf{x}_i$ and can be chosen in different ways (see [20]), $N(i)$ is the set of 1-ring neighbor vertices of vertex i and α_{ij} and β_{ij} are defined in Figure 2. The mean curvature H can be easily computed

$$H(\mathbf{x}_i) = \frac{1}{2} \|\mathbf{K}(\mathbf{x}_i)\|. \tag{24}$$

Moreover the normal vector is computed normalizing $\mathbf{K}(\mathbf{x}_i)$.

As for the Gaussian curvature G the following expression can be used:

$$G(\mathbf{x}_i) = \left(2\pi - \sum_{j=1}^{N_f} \theta_j \right) / A_p \tag{25}$$

where N_f is the number of faces around vertex i.

DOI: 10.1685/journal.caim.461

Finally the principal curvature can be estimated starting from G and H as follows:

$$K_{1,2}(\mathbf{x}_i) = H(\mathbf{x}_i) \pm \sqrt{\Delta(\mathbf{x}_i)}, \tag{26}$$

where $\Delta(\mathbf{x}_i) = H^2(\mathbf{x}_i) - G(\mathbf{x}_i)$.

4. Constrained reconstruction procedure.

4.1. *Algorithm for spatial curves.*

The algorithm we are advocating can be summarized as follows:

1. Given a set of $m+1$ measured points $\mathbf{P}_i$ the algorithm computes a set of $m+1$ control points $\mathbf{CP}_i$ defining an interpolating B-spline curve of degree k. The parametric equation of the curve is:

$$S_m(u) = \sum_{i=0}^{m} B_i^k(u)\mathbf{CP}_i. \tag{27}$$

Precisely once a suitable parametrization (centripetal, uniform, chordal, ...) of the given data is chosen, each u_j corresponds to a data point $\mathbf{P}_j$. In our tests we have used the centripetal parametrization.
The values in the knot vector can be arbitrary chosen or they can be linked to the values of the parameter. In particular in our case we have chosen the following parameter averaging strategy [12]:

$$\begin{aligned} &t_{-k} = \ldots = t_0 = 0, \\ &t_{j+k} = \tfrac{1}{k}\textstyle\sum_{i=j}^{j+k-1} u_i, \qquad j = 1, \ldots, n+1, \\ &t_{n+1} = \ldots = t_{m+1} = 0, \end{aligned} \tag{28}$$

where $n = m - k$.
We can write the interpolation conditions as follows

$$\mathbf{P}_j = S_m(u_j) = \sum_{i=0}^{m} B_i^k(u_j)\mathbf{CP}_i \qquad j = 0, \ldots, m. \tag{29}$$

Introducing the B-spline collocation matrix $\mathcal{B}$, whose elements are $b_{ji} = B_i^k(u_j)$, we can rewrite equation (29) in the following matrix notation

$$\mathbf{P} = \mathcal{B}\mathbf{CP} \tag{30}$$

which is a linear system in the unknown vector $\mathbf{CP}$.

2. Using the knot vector $\mathbf{t}$, the control point vector $\mathbf{CP}$, the Schoenberg point computed according to (5) and the matrix $\mathcal{M}$ computed using formula (2) we can write formally the curve $T_m^\lambda \mathbf{CP} = S_m((1-\lambda)\mathbf{CP} + \lambda\mathcal{M}\mathbf{CP})$. From the practical point of view this amounts to apply equation (7).
3. The chosen (discrete) functional is written in terms of $\mathbf{CQ}^\lambda$; the minimization of this functional gives λ_{opt} (by means of the Matlab command `fminsearch`).
4. Taking into account expression (7) the final curve is constructed using the following control points:

$$\mathbf{CP}^{\lambda_{opt}} = (1-\lambda_{opt})\mathbf{CP} + \lambda_{opt}\mathcal{M}\mathbf{CP} \tag{31}$$

4.2. *Algorithm for surfaces.*

We extend the previous algorithm to the bivariate spline case and in the following we summarize it.

1. Let suppose we have $(m+1)(n+1)$ data points $\mathbf{P}_{ij}$ $i = 0, 1, ..., m$, $j = 0, 1, ..., n$ on the surface, first the algorithm computes a set of $(m+1)(n+1)$ control points $\mathbf{CP}$ defining an interpolating tensor (degrees k, h) B-spline surface. Let us organize the set $\mathbf{CP}$ into $n+1$ sets of $m+1$ elements each, *i.e.* $\mathbf{CP}_{ij}$ $i = 0, 1, ..., m$, $j = 0, 1, ..., n$, where $\mathbf{CP}_{ij}$ are three-dimensional vectors, whose components are the coordinates of the control points. The parametric equation of the curve is:

$$S_{mn}(u, w) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_j^k(w) \mathbf{CP}_{ij} B_i^h(u). \tag{32}$$

Precisely once a suitable parametrization of the given data is chosen, each u_c and w_d ($c = 0, 1, ..., m$ $d = 0, 1, ..., n$) corresponds to a data point $\mathbf{P}_{ij}$. In our tests we have used the centripetal parametrization for both parameters.
The values in the knot vectors $\mathbf{t}$ and $\mathbf{s}$ can be arbitrary chosen or they can be linked to the values of the parameters. In particular in our case we have chosen the parameters analogously to the univariate case.
Plugging $u = u_c, w = w_d$ in (32) we can write the interpolation conditions as follows:

$$\begin{aligned}(\mathbf{P}_{cd})_l &= (S_{mn}(u_c, w_d))_l \\ &= \sum_{i=0}^{m} (\sum_{j=0}^{n} B_j^k(w_d)(\mathbf{CP}_{ij})_l) B_i^h(u_c) \qquad d = 0, 1, \ldots n;\ c = 0, 1, \ldots, m.\end{aligned} \tag{33}$$

DOI: 10.1685/journal.caim.461

In (32) we can assume the surface swept by a curve spline family whose control points are on a curve spline family.
Introducing the B-spline collocation matrices $\mathcal{B}$ and $\mathcal{D}$, whose elements are respectively $b_{ci} = B_i^k(u_c)$ and $d_{jd} = B_j^h(w_d)$ we can rewrite equation (33) in the following matrix notation

$$(\mathbf{P}_{cd})_l = \mathcal{B}(\mathbf{CP})_l\mathcal{D} \tag{34}$$

Starting from (33):

- we solve $n+1$ linear systems to determine $n+1$ sets of $l-th$ components of $m+1$ points $\mathbf{CQ}_{0d}, \mathbf{CQ}_{1d}, ..., \mathbf{CQ}_{md}$ as control points of spline curves sweeping the approximating surface. Precisely $S_{mn}(u_c, w_d)_l = \sum_{i=0}^{m}(\mathbf{CQ}_{id})_l B_i^h(u_c)$ as $c = 0, 1, ..., m$;
- we solve linear $m+1$ systems to determine $m+1$ set of $l-th$ components of $n+1$ points $\mathbf{CQ}_{c0}, \mathbf{CQ}_{c1}, ..., \mathbf{CQ}_{cn}$, as control points of spline curve giving the control points to sweeping curves.
  As suggested in literature the solution of each systems is carried out by the LU decomposition.

Using the knot vector $\mathbf{t}$, $\mathbf{s}$ the control points $\mathbf{CP}$, the Schoenberg points computed according to (5) respectively on the sets $\mathbf{t}$ and $\mathbf{s}$ and the matrix $\mathcal{M}$ and $\mathcal{N}$ computed adapting formula (2) we can write formally the surface as (8). From the practical point of view this amounts to apply equation (10).

2. The chosen (discrete) functional is written in terms of $\mathbf{CP}_l^{\lambda\tau}$; the minimization of this functional gives λ_{opt} and τ_{opt} (by means of the Matlab command `fminsearch`).
3. Taking into account expression (10) the final surface is constructed using the following control points:

$$\begin{aligned}(\mathbf{CP})_l^{\lambda_{opt}\tau_{opt}} &= (1-\tau_{opt})(1-\lambda_{opt})(\mathbf{CP})_l + \tau_{opt}(1-\lambda_{opt})\mathcal{N}(\mathbf{CP})_l \\ &+ \lambda_{opt}(1-\tau_{opt})\mathcal{M}(\mathbf{CP})_l + \lambda_{opt}\tau_{opt}\mathcal{M}(\mathbf{CP})_l\mathcal{N}\end{aligned} \tag{35}$$

5. Numerical results.

In this section we are going to present some results concerning (academic and real world) test cases in order to asses the effectiveness of the proposed method for (re)constructing curves and surfaces from scattered data.

5.1. *Test 1: analytical curve.*

In the first test case we consider an analytical curve, namely a conical helix whose equation is:

$$\mathbf{c}(t) = \left[\frac{(3\pi + t)\cos(t)}{3\pi}, \frac{(3\pi + t)\sin(t)}{3\pi}, \frac{t}{\pi}\right]^T, \qquad t \in [0, 6\pi]. \tag{36}$$

We took 100 samples from this curve and we have used quintic B-splines for the reconstruction procedure.

The aim of this test is to show the differences obtained in the reconstruction procedure using

- simple interpolation;
- least square approximation;
- constrained interpolation smoothing technique proposed in this paper.

These techniques are evaluated on different data sets characterized by a different number of samples (15 for the small set and 100 for the big one) and different noise. In particular three perturbations have been considered:

- "high noise": gaussian noise with zero average and variance equal to 0.1;
- "low noise": gaussian noise with zero average and variance equal to 0.01;
- "scattered noise": only 10 samples are affected by the noise: these samples are scattered along the curve and the amplitude of the noise is 0.1.

Table 2. Values of $\int(\rho_{ex} - \rho)^2 ds$. LS = Least Squares, CI = Constrained Interpolation, I = Interpolation.

Test	LS	CI	I
Small set - High noise	2.13	1.75	19.32
Big Set - Low noise	1.44	0.91	13.11
Big Set - Scattered noise	0.72	0.55	12.27

According to the results shown in the previous table we can conclude that in general the normal interpolation does not perform very well. More in detail: if the number of sample is small and the sample are characterized by a significant noise then the least squares results are comparable to the ones obtained by the constrained smoothing. The method advocated in this paper is particularly suitable when there are a lot of samples and only a small number of them is affected by the noise.

In this test we have considered the first functional given in Table 1.

DOI: 10.1685/journal.caim.461

5.2. *Test 2: titanium heat data.*

In this second test we consider the classical dataset concerning the titanium heat data introduced by De Boor. In this case we have used the second functional given in Table 1. In Figure 3 the result of the reconstruction along with the original data is reported.

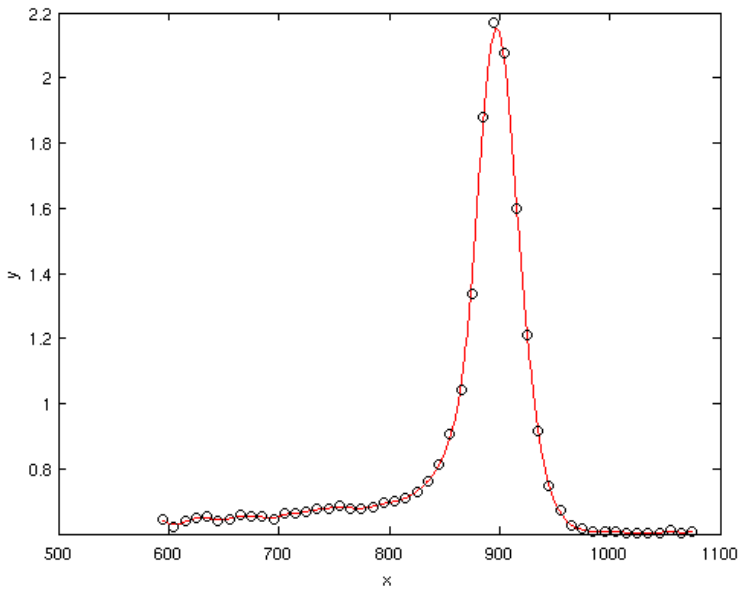

Figure 3. Reconstruction of the titanium heat data.

5.3. *Test 3: real curve.*

In the third test case we considered a data set coming from a measurement of a sole profile collected with the aim of creating a virtual prototype of the shoe in order to easily build models of different shoe's size. Figure 4 shows the 174 measured points obtained using a Coordinate Measuring Machine (CMM).

The advocated method has been applied for the reconstruction of the shoe's profile: in particular quintic B-spline have been adopted and the first functional in Table 1 has been used.

In Figure 5 the control points obtained applying the proposed method are shown: it is possible to see some oscillations in the position of these points inducing unexpected changes in the curvature. The use of different functionals in different parts of the curve (in order to satisfy different shape

constraints) can solve this problem creating a smoother curve. In Figure 6 the curve reconstructed using different functionals is shown.

The choice of the functionals must be done *a-priori* but it can be guided by a first (re)construction of the curve using classical approaches.

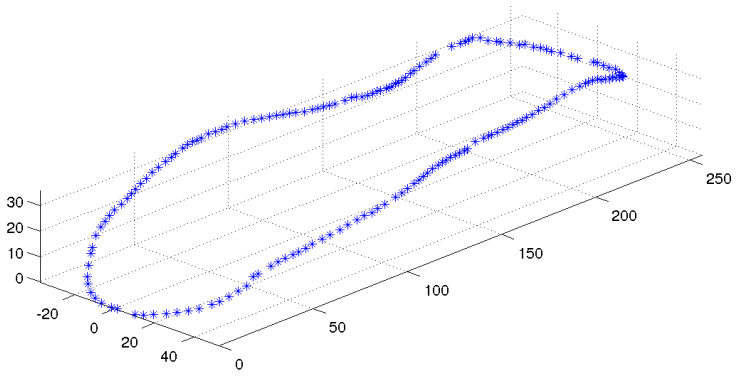

Figure 4. Measured points for the shoe's profile.

5.4. *Test 4: equation of state aluminium data.*

In this case we consider the dataset presented [21] concerning the equation of state of the aluminium. This dataset is quite difficult to approximate. For this case we have considered the rounding metric. Figure 7 shows the original dataset and in Figure 8 the result obtained using the procedure advocated in this paper is reported. The result is satisfactory even if some improvements could be obtained using (as in the previous test case) different functionals in different part of the surface or using more shape parameters to have a stronger control on the resulting shape of the surface (both these approaches are under investigation and they will be the subject of a future paper).

DOI: 10.1685/journal.caim.461

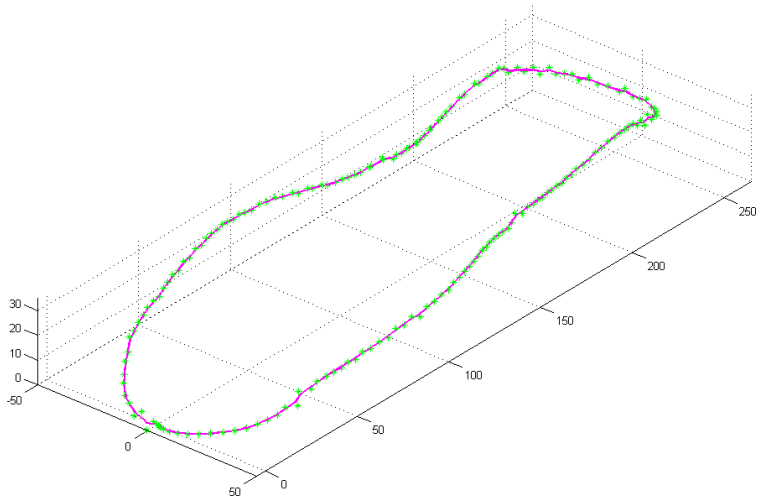

Figure 5. Control points obtained using the proposed approach and the first functional in Table 1.

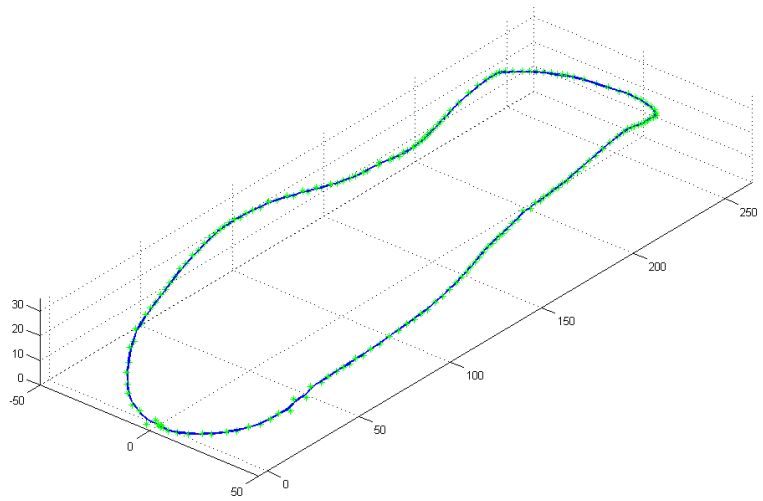

Figure 6. Reconstructed curve using different functionals in different parts.

5.5. *Test 5: analytical surface.*

The following analytical surface is considered

$$\begin{cases} x(t,s) = \cos(t+s) + \sqrt{|t|} \\ y(t,s) = 2t + 3s + ts + 0.1\epsilon \\ z(t,s) = \sin(t) + s^3 + ts + 0.1\epsilon \end{cases} \tag{37}$$

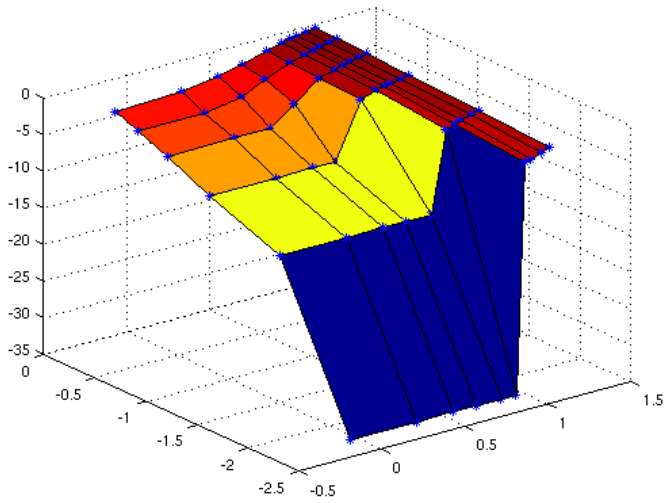

Figure 7. Dataset for the equation of state of the aluminium.

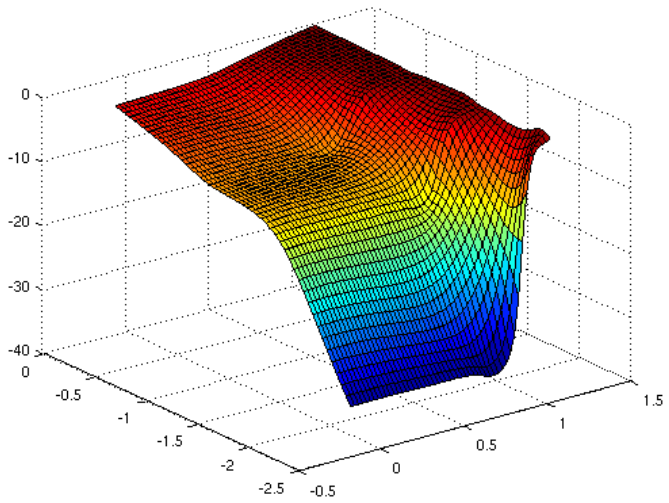

Figure 8. Reconstructed surface for the equation of state of the aluminium.

where ϵ is a uniform random noise. A grid of 20×20 samples is considered. For the reconstruction quintic splines are used for describing both the sweeping curves and the ones giving the control points (to the sweeping curves). The *flattening* metric (14) has been used for defining the functional.

DOI: 10.1685/journal.caim.461

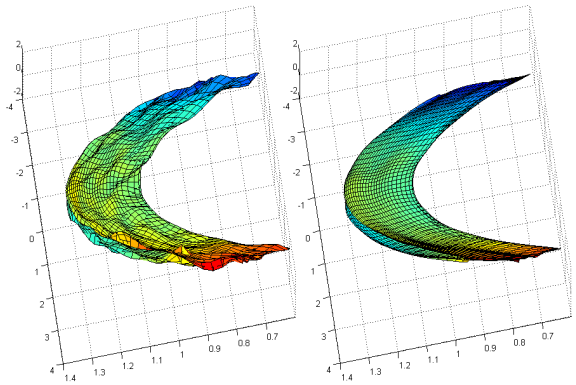

Figure 9. Analytical surface with noise (left) and reconstructed surface (right).

The results are shown in Figure 9.

5.6. *Test 6: real surface.*

The shape of an aeronautical panel must be acquired, prior its submission to a crash test; a coordinate measuring machine (see Figure 10) is used in order to sample the surface: in particular 59×68 $x - y$-grid of points has been acquired (see Figure 11). The panel is a slightly bended cylindrical surface, with evenly-distributed irregularities (rivets): the aim is to reconstruct the surface filtering out the presence of the rivets.

The *flattening* metric (14) has been used for defining the functional. Quintic splines are used for both spline classes in the tensor expression (see equation (33)). In Figure 12 a classical reconstruction is shown while in Figure 13 the result of the proposed method is presented.

6. Conclusions.

In this paper we have presented a method to optimally reconstruct 3D curves and surfaces. The specific features of the method are related to the use of a particular class of integral spline operator characterized by the presence of shape parameters. Moreover in order to satisfy particular requirements (*i.e.* shape constraints) the user can define a properly chosen functionals (related to the desired results) depending on parameters; the minimization of functionals leads to the optimal behavior of curve and surfaces. The approach we have presented is very flexible since it is quite easy and cheap to use splines of high degree if necessary (whereas most of the classical methods uses splines of degree 3).

Some results on academic (taken also from existing literature for comparison) and real-world cases show the effectiveness of the proposed algo-

Figure 10. CMM Measuring Machine.

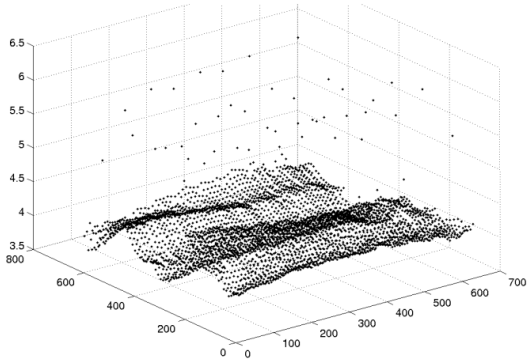

Figure 11. Acquired points.

rithm.

REFERENCES

1. T. Schütze and H. Schwetlick, Constrained approximation by splines with free knots, *BIT Numerical Mathematics*, vol. 37, no. 1, pp. 105–137, 1997.
2. K. Maccallum and J. Zhang, Curve-smoothing techniques using B-splines, *The Computer Journal*, vol. 29, no. 6, pp. 564–571, 1986.
3. R. Goldenthal and M. Bercovier, Spline curve approximation and design by optimal control over the knots, *Computing*, vol. 72, pp. 53–64, 2004.

DOI: 10.1685/journal.caim.461

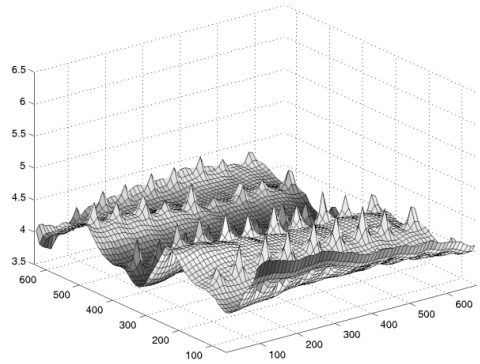

Figure 12. Reconstructed panel with rivets.

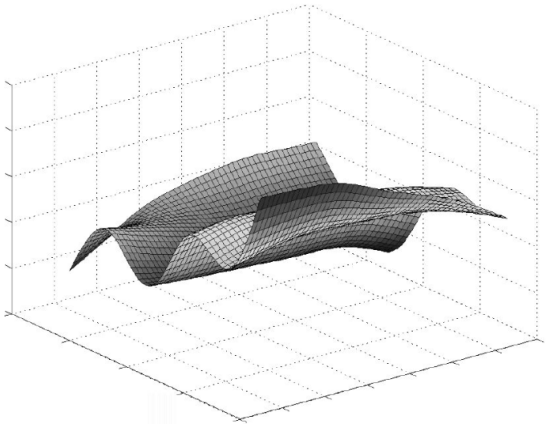

Figure 13. Reconstructed panel without rivets.

4. H. Yang, W. Wang, and J. Sun, Control point adjustment for B-spline curve approximation, *CAD*, vol. 36, pp. 639–652, 2004.
5. P. Costantini and F. Pelosi, Shape-preserving approximation of spatial data, *Advances in Computational Mathematics*, vol. 20, pp. 25–51, 2004.
6. L. Kocić and G. Milovanović, Shape preserving approximations by polynomials and splines, *Computers & Mathematics with Applications*, vol. 33, no. 11, pp. 59–97, 1997.
7. T. Schütze and H. Schwetlick, Bivariate free knot splines, *BIT Numerical Mathematics*, vol. 43, no. 1, pp. 153–178, 2003.
8. P. Costantini, F. Pelosi, and M. Sampoli, Boolean surfaces with shape

constraints, *Computer-Aided Design*, vol. 40, no. 1, pp. 62–75, 2008.

9. P. Costantini and F. Pelosi, Data approximation using shape-preserving parametric surfaces, *SIAM Journal on Numerical Analysis*, vol. 47, p. 20, 2008.
10. F. Caliò, E. Miglio, and M. Rasella, Curve fairing using integral spline operators, *International Journal for Numerical Methods in Biomedical Engineering*, vol. 26, no. 12, pp. 1674–1686, 2010.
11. G. Milovanović and L. Kocić, Integral spline operators in CAGD, *Atti Sem. Mat. Fis. Univ. Modena*, vol. 39, pp. 433–454, 1991.
12. C. De Boor, *A practical guide to splines*, vol. 27. Springer Verlag, 2001.
13. F. Caliò, E. Miglio, G. Moroni, and M. Rasella, Integral $\lambda - \tau$ bivariate spline operators in computer graphics problems, *Studia univ. "Babes-Bolyai", Mathematica*, vol. 49, no. 4, pp. 43–53, 2004.
14. G. Albrecht, Invariante gütekriterien im kurvendesign, in *Effiziente Methoden der geometrischen Modellierung und der wissenschaftlichen Visualisierung*, pp. 134–148, Teubner, 1999.
15. N. Sapidis, *Designing fair curves and surfaces: shape quality in geometric modeling and computer-aided design*. Society for Industrial Mathematics, 1994.
16. G. Greiner, A. Kolb, and A. Riepl, Scattered data interpolation using data dependant optimization techniques, *Graphical Models*, vol. 64, no. 1, pp. 1–18, 2002.
17. E. Karousos, A. Ginnis, and P. Kaklis, Controlling torsion sign, *Computer Aided Geometric Design*, vol. 26, no. 4, pp. 396–411, 2009.
18. A. Ginnis, E. Karousos, and P. Kaklis, A discrete methodology for controlling the sign of curvature and torsion for NURBS, *Computing*, vol. 86, no. 2, pp. 117–129, 2009.
19. R. Martin, Estimation of principal curvatures from range data, *International journal of shape modeling*, vol. 4, no. 3/4, pp. 99–110, 1998.
20. M. Meyer, M. Desbrun, P. Schröder, and A. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, *Visualization and mathematics*, vol. 3, no. 7, pp. 34–57, 2002.
21. R. Carlson and F. Fritsch, Monotone piecewise bicubic interpolation, *SIAM J. Num. Anal.*, vol. 22, no. 2, pp. 386–400, 1985.