Point Cloud Fitting by G1 Smooth Spline Functions

Michelangelo Marsala, Angelos Mantzaflaris, Bernard Mourrain

Point Cloud Fitting by G^1 Smooth Spline Functions

Michelangelo Marsala*, Angelos Mantzaflaris, Bernard Mourrain

Inria, Université Côte d'Azur, France

Abstract

In this work we analyze the space of geometrically smooth biquintic Bézier polynomials defined on a quadrilateral mesh $\mathcal{M}$ generated by the G^1 Approximate Catmull-Clark scheme (G^1ACC) in [27]. With the use of quadratic gluing data functions, an explicit construction for an efficient set of basis functions generating the G^1ACC space is provided as well as a dimension formula. The global structure is therefore applied to solve point cloud data fitting problems defined over multipatch domains whose quality is demonstrated by numerical experiments presented in this paper performed over several data points possessing various features with interest for applied problems.

Keywords: Point cloud fitting, multipatch domain, gluing data, geometric continuity, extraordinary vertices

1. Introduction

A fundamental operation in geometric modeling is the construction of geometric models from measurements or observations. This consists in computing an accurate representation of the surface of an object, from a set of points obtained e.g. from laser scanners, Light Detection And Ranging (LIDAR) scanners or stereo camera devices. With the fast development of accessible acquisition tools, it is nowadays possible to use dense and precise point clouds to compute geometric models, such as meshes. Such reconstruction tools are indeed used in many applications including earth surveying tasks [41, 39, 35], geographic information and navigation systems [30], building model reconstructions [43], cultural heritage information systems [31], reverse engineering [7, 37], metrology [40, 44], medical imaging [33, 34], ...

The computed mesh model has a size similar to that of the input point cloud and often requires pre and post processing steps (mesh simplifications, hole filing, noise removal, remeshing, smoothing, ...) to provide a representation of the shape with good regularity and good approximation properties (see e.g. [2, 18, 25, 3]).

A classical approach employed to smooth data is based on B-spline representations. Given a sequence of points, a B-spline curve of a given regularity, approximating the point set within a given precision, can be computed by repeatly solving a least square distance problem and inserting knots [8]. This results in an accurate, regular and efficient description of a curve. The approach can be extended to rectangular patches of tensor product B-splines for point clouds, which rely on rectangular pieces of surfaces. These techniques are routinely used in Computer Aided Geometric Design. Advance techniques [42] have been investigated to tune the distance function [38] and to associate adequate parameter values to the points to be fit [9]. Local refinement techniques, based on e.g. Truncated Hierarchical B-splines (THB) [10, 20], T-splines [19], Locally Refined splines (LR) [36], or Polynomial splines over Hierarchical T-meshes (PHT) [22] have been developed to reduce the fitting error where needed, while keeping the expected regularity.

However, in many applications, the shape to be described cannot be mapped continuously and bijectively to a single rectangle and collections of quadrangular faces may be needed to properly fit the whole point cloud. This does not prevent the reconstructed B-spline faces to have non-smooth junctions along shared edges. In [23], point wise constraints on the normals of biquintic Bezier patches along the shared edges are

*Corresponding author

Email addresses: michelangelo.marsala@inria.fr (Michelangelo Marsala), angelos.mantzaflaris@inria.fr (Angelos Mantzaflaris), bernard.mourrain@inria.fr (Bernard Mourrain)

imposed to get an approximate regularity along the edges of the quadrangular patches. In [45], a so-called "immune genetic algorithm" is used to solve the fitting problem augmented with G^1 constraints. Theses patch-by-patch approaches require the treatment of G^1 constraints as additional constraints of the squared distance minimization problem and do not guarantee that the computed surface is smooth. In [11, 6], G^1 surfaces of bidegree $(3,3)$ and $(4,4)$ with 4 splits and respectively linear and quadratic gluing data are computed to interpolate a network of curves. Some fitting optimisation is used to approximate a point cloud, once the curve edges of the faces have been fixed. This constraints the resulting surface to have edges aligned with prescribed curves, and the fitting quality of the resulting surface depends directly on these prescribed curves.

To address these issues, we propose a new fitting approach which involves directly global G^1 functions. The resulting surface is, by construction, smooth everywhere with continuous tangent planes along all shared edges. We solve the squared distance minimization problem in the space of G^1 surface parametrisations, associating to each point of the point cloud, well-chosen parameters on the G^1 parametric surface.

The approach requires to know an explicit basis of the space of G^1 functions. The problem of computing dimension and bases of splines spaces is a hot topic in splines theory and isogeometric analysis. Many works have investigated the dimension and basis of spline spaces over planar domains. See e.g. [1, 13, 32, 21, 29] and references therein. Much less results are known on G^r spline spaces [28, 15, 16, 17, 4, 5].

In this work, we consider G^1 splines over quad meshes of arbitrary topology. We use quadratic gluing data, that are smooth along edges across regular vertices. We provide a new and explicit construction of basis functions for the space of G^1 splines functions, which are biquintic on each patch. The basis functions are attached respectively to vertices, edges and faces. Their coefficients in the Bernstein basis of each face depend only on the valence of the vertices and edges. They can be pre-computed independently of the full quad-mesh structure.

Exploiting this efficient basis construction, we experiment the fitting method on complex shapes built from quad meshes, such as Catmull-Clark subdivision surfaces or graphs of functions on spherical domains.

Outline. After having recalled some standard objects we will use throughout the paper in Section 2, in Section 3 we present the definition of G^1 smooth spline on multipatch domain making use of quadratic gluing data functions; moreover, we recall the equations generating the G^1 Approximate Catmull-Clark space (G^1ACC) [27] in which we are interested. Section 4 is devoted to the construction of an explicit set of basis function generating the G^1ACC space: explicit formulas for the computation of the control points governing the Bézier polynomial is given as well as a dimension formula in Section 5. We will test the goodness of our bases in solving point cloud data fitting problems with several examples of different types presented in Section 6, also showing different errors computations and colorplot. Section 7 concludes the work.

2. Preliminaries

Before going into the details of our construction, we need to introduce some definitions that we will use throughout the paper.

Definition 1 (Bézier patch)**.** *Let $\mathbf{b}_{i,j} \in \mathbb{R}^3$ be points in the Euclidean space defined as control (or Bézier) points; a (tensor-product) Bézier patch is a surface defined as*

$$\mathcal{Q}(u,v) = \sum_{i,j=0}^{n,m} \mathbf{b}_{i,j} B_{n,m}^{i,j}(u,v), \quad (u,v) \in [0,1]^2,$$

with $B_{n,m}^{i,j}(u,v) = B_n^i(u) B_m^j(v)$ the bivariate Bernstein polynomials of bidegree (n,m), that is, $B_d^h(u) = \binom{d}{h} u^h (1-u)^{d-h}$.

A quadrilateral *mesh* $\mathcal{M}$ is a collection of elements, i.e. faces, edges and vertices, forming a polyhedral complex whose faces subsists of quadrilaterals. We define the valence N of a vertex as the number of edges (or faces) to which it belongs: in the case of a quad mesh, if $N = 4$ we have a regular vertex (RV), while if $N \neq 4$ we have an extraordinary vertex (EV). Given a mesh $\mathcal{M}$ it is interesting to count the main features it possesses: we will refer to n_V as the number of vertices of the mesh, which can be subdivided in n_{IEV} inner EVs, n_{BEV} boundary EVs and n_{RV} regular vertices. The number of edges is identified with n_E in which we can count more precisely the number of extraordinary and regular edges, respectively n_{EE} and

n_{RE}, namely edges attached to extraordinary and regular vertices, and the number of boundary edges n_{BE}. With n_F we refer to the number of faces and lastly n_C the number of corners of the mesh.

A multipatch surface is defined as the collection of patches $f = (f_\sigma)_{\sigma\in\mathcal{M}}$ where σ is a face of the mesh $\mathcal{M}$ and f_σ is the restriction of f on the face σ. See [27] for more details.

3. G^1 Constrains and G^1ACC space

A surface is said to be G^1 if for every point on it there exists a unique tangent plane. In order to define the G^1 continuity between patches in a mathematical setting, we use *gluing data* functions $a_{N,N'} : e \longrightarrow \mathbb{R}$, $b_{N,N'} : e \longrightarrow \mathbb{R}$, where e is the common edge shared between two adjacent extraordinary patches of the mesh $\mathcal{M}$ and connecting a vertex of valence N with one of valence N'.

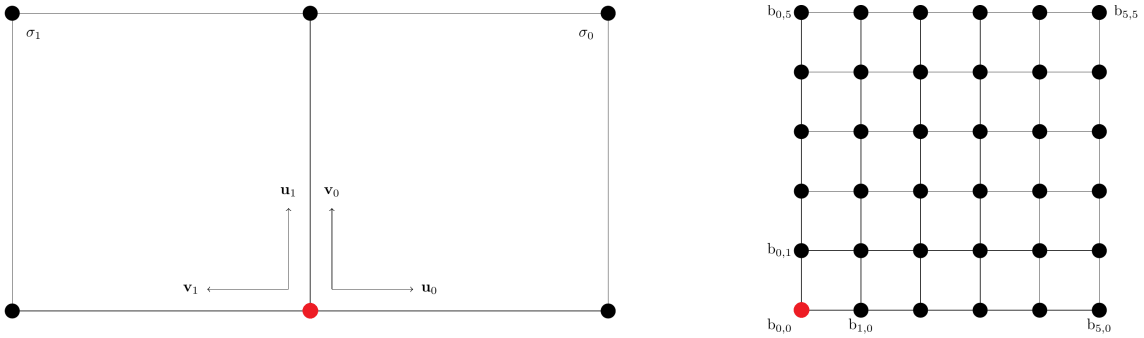


Figure 1: Left: local coordinate systems between two patches. Right: control point labeling in a biquintic patch.

A multipatch function $f = (f_\sigma)_{\sigma\in\mathcal{M}}$ on a mesh $\mathcal{M}$ is said differentiable or G^1 if:

$$\begin{cases} f_1(u_1, 0) = f_0(0, u_1)\,, \\ \dfrac{\partial f_1}{\partial v_1}(u_1, 0) = b_{N,N'}(u_1)\dfrac{\partial f_0}{\partial u_0}(0, u_1) + a_{N,N'}(u_1)\dfrac{\partial f_0}{\partial v_0}(0, u_1)\,, \end{cases} \qquad u_1 \in [0, 1]\,, \tag{1}$$

where f_0, f_1 are the functions defined on faces $\sigma_0, \sigma_1 \in \mathcal{M}$ sharing an edge e (see Fig. 1) and $a_{N,N'}, b_{N,N'}$ are the gluing data defined on the edge e. Since we developed our construction in the parametric domain, we can take $e = [0, 1]$. The functions satisfying eq. (1) for all shared edges belong to the linear space of G^1 B-spline functions on $\mathcal{M}$. In this paper we consider quadratic symmetric gluing data defined as

$$\begin{aligned} &a_{N,N'}(u) = a_0 B_2^0(u) - a_2 B_2^2(u)\,, \quad \text{where} \quad a_0 = 2\cos\left(\frac{2\pi}{N}\right)\,,\ a_2 = 2\cos\left(\frac{2\pi}{N'}\right)\,, \\ &b_{N,N'}(u) = -1\,, \end{aligned} \tag{2}$$

with $B_2^i(u)\,, i = 0, 2$, the standard univariate Bernstein polynomials introduced in Definition 1. We refer to [12, 6, 11, 12, 27] for more details.

Without loss of generality, we assume that $N' = 4$ so that $a_2 = 0$, i.e. all the EVs of the mesh are surrounded by regular vertices.

Let f_1 and f_2 be two Bézier patch of bidegree (d, d). With this assumption, the G^1 relation across an edge defined in (1) specializes as

$$\begin{cases} \displaystyle\sum_{i=0}^{d} \mathbf{b}_{i,0}^{(1)} B_d^i(u) = \sum_{i=0}^{d} \mathbf{b}_{0,i}^{(0)} B_d^i(u)\,, \\ \displaystyle\sum_{i=0}^{d} \left(\mathbf{b}_{i,1}^{(1)} - \mathbf{b}_{i,0}^{(1)} + \mathbf{b}_{1,i}^{(0)} - \mathbf{b}_{0,i}^{(0)}\right) B_d^i(u) = a_0 B_2^0(u) \left(\displaystyle\sum_{i=0}^{d-1} \left(\mathbf{b}_{0,i+1}^{(0)} - \mathbf{b}_{0,i}^{(0)}\right) B_{d-1}^i(u)\right)\,, \end{cases} \tag{3}$$

where we used the gluing functions introduced in (2) and $\mathbf{b}_{i,j}^{(k)}$ refers to the control point $\mathbf{b}_{i,j}$ living in the patch σ_k. In this work we decided to focalize the investigation by using biquintic Bézier patches, i.e. with bidegree $(5,5)$; this choice has also been made in [27], where an explicit construction of G^1ACC spline functions is presented. The G^1ACC surface is defined through masks returning a multipatch Bézier surface with G^1 smoothness around EVs and at least C^1 elsewhere. The coordinate functions of this surface are elements of the space of G^1 spline functions that we study in this paper.

The equations defining the geometric continuity constrains across an extraordinary edge, obtained making explicit the formulas in (3) for $d = 5$ are the following:

$$\begin{aligned}
\mathrm{b}_{0,1}^{(1)} + \mathrm{b}_{1,0}^{(0)} &= \bar{a}_0 \mathrm{b}_{0,0}^{(1)} + a_0 \mathrm{b}_{1,0}^{(1)}, && (4)\\
5(\mathrm{b}_{1,1}^{(1)} + \mathrm{b}_{1,1}^{(0)}) &= a_0 \mathrm{b}_{0,0}^{(1)} + 5\bar{a}_0 \mathrm{b}_{1,0}^{(1)} + 4a_0 \mathrm{b}_{2,0}^{(1)}, && (5)\\
10(\mathrm{b}_{2,1}^{(1)} + \mathrm{b}_{1,2}^{(0)}) &= -a_0 \mathrm{b}_{0,0}^{(1)} + 5a_0 \mathrm{b}_{1,0}^{(1)} + 10\bar{a}_0 \mathrm{b}_{2,0}^{(1)} + 6a_0 \mathrm{b}_{3,0}^{(1)}, && (6)\\
10(\mathrm{b}_{3,1}^{(1)} + \mathrm{b}_{1,3}^{(0)}) &= a_0 \mathrm{b}_{0,0}^{(1)} - 5a_0 \mathrm{b}_{1,0}^{(1)} + 10a_0 \mathrm{b}_{2,0}^{(1)} + 10\bar{a}_0 \mathrm{b}_{3,0}^{(1)} + 4a_0 \mathrm{b}_{4,0}^{(1)}, && (7)\\
\mathrm{b}_{4,1}^{(1)} + \mathrm{b}_{1,4}^{(0)} &= 2\mathrm{b}_{4,0}^{(1)}, && (8)\\
\mathrm{b}_{5,1}^{(1)} + \mathrm{b}_{1,5}^{(0)} &= 2\mathrm{b}_{5,0}^{(1)}, && (9)\\
10(\mathrm{b}_{3,0}^{(1)} - \mathrm{b}_{2,0}^{(1)}) &= \mathrm{b}_{0,0}^{(1)} - 5\mathrm{b}_{1,0}^{(1)} + 5\mathrm{b}_{4,0}^{(1)} - \mathrm{b}_{5,0}^{(1)}, && (10)
\end{aligned}$$

with $\bar{a}_0 = 2 - a_0$.

Starting from the equations (4)-(10), the aim of this paper is to present an explicit construction of a set of basis function generating the G^1ACC space, which will be applied for point cloud fitting problems.

4. Bases extraction

In this section we will present an explicit construction for the space of basis functions generating the G^1ACC space, to which we refer as $\mathcal{B}$. Following the topology of the input mesh $\mathcal{M}$, we can decompose the construction specializing it into space of bases attached to the vertices $\mathcal{B}_V$ (i.e. spanned by bases whose support lies on all the patches sharing the vertex), space of basis functions attached to the edges $\mathcal{B}_E$ (i.e. spanned by bases whose support lies on the patches - or patch- sharing the edge) and space of basis functions attached to the faces $\mathcal{B}_F$ (i.e. spanned by bases whose support lies only on the interior of a single patch). The idea under this construction is to achieve a set of basis function which can be decomposed as

$$\mathcal{B} = \left(\bigoplus_{i=1}^{n_V} \mathcal{B}_{V_i}\right) \oplus \left(\bigoplus_{i=1}^{n_E} \mathcal{B}_{E_i}\right) \oplus \left(\bigoplus_{i=1}^{n_F} \mathcal{B}_{F_i}\right). \tag{11}$$

We assume that we have only isolated EVs. In fact, this justifies the decomposition in (11) we want. Having EVs only attached to regular vertices, which translates into eq. (8)-(9), implies vertices, edges and faces functions with null value and derivatives at the extremes points of their support; thus no basis function belonging to an element of the mesh $\mathcal{M}$ is influenced, or influence, other functions belonging to the neighbouring elements. Chaining the equations in system (4)-(10) circularly around all the edges attached to an EV we can reformulate the G^1 constraints by using a staircase block matrix, which will be useful to better understand the bases extraction and their analysis we will present in the next section. Let $\mathbf{b}_{i,j} = \left(\mathrm{b}_{i,j}^{(k)}\right), k = 1, \dots, N$, be the vector containing all the points $\mathrm{b}_{i,j}^{(k)}$ attached to the neighborhood of the EV we are considering, $\mathbf{u} = (1,1,1,1)^T$, $C \in \mathbb{R}^{N \times N}$ a circular matrix defined as

$$C = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ 1 & 0 & & \cdots & 0 \end{pmatrix} = \mathrm{Circ}(0,1,0,\dots,0), \tag{12}$$

and let $\mathbf{b}_{4,0} = \mathbf{b}_{5,0} = 0$ as a consequence of the isolated EVs hypothesis ; the full system of G^1 relations around an EV can be written as

$$\begin{pmatrix} -\bar{a}_0\mathbf{u} & C_1 & & & & & & & \\ -a_0\mathbf{u} & -5\bar{a}_0 I & -4a_0 I & 5C_2 & & & & & \\ -\mathbf{u} & 5I & -10I & & 10I & & & & \\ a_0\mathbf{u} & -5a_0 I & -10\bar{a}_0 I & & -6a_0 I & 10I & 10I & & \\ -a_0\mathbf{u} & 5a_0 I & -10a_0 I & & 10\bar{a}_0 I & & & 10I & 10I \end{pmatrix} \begin{pmatrix} \mathbf{b}_{0,0} \\ \mathbf{b}_{1,0} \\ \mathbf{b}_{2,0} \\ \mathbf{b}_{1,1} \\ \mathbf{b}_{3,0} \\ \mathbf{b}_{2,1} \\ \widehat{\mathbf{b}}_{1,2} \\ \mathbf{b}_{3,1} \\ \widehat{\mathbf{b}}_{1,3} \end{pmatrix} = \mathbf{0}\,, \tag{13}$$

where $a_0 = 2\cos(2\pi/N)$ is the coefficient of the gluing data function, $\bar{a}_0 = 2 - a_0$, $I \in \mathbb{R}^{N\times N}$ is the identity matrix,

$$C_1 = -a_0 I + C + C^{N-1}\,, \quad C_2 = I + C\,, \tag{14}$$

and $\widehat{\mathbf{b}}_{1,2} = C\,\mathbf{b}_{1,2}$, $\widehat{\mathbf{b}}_{1,3} = C\,\mathbf{b}_{1,3}$, with C in (12). The strategy we exploit to obtain the Bézier coefficients of the basis functions in the Bernstein representation is the following: starting from eq. (4)-(10) or eq. (13) we impose, one by one, each coefficient involved in the G^1 constrains to have value 1. Then, with this initial value we begin to solve the G^1 relations in (4)-(10) or eq. (13), while we gradually set the value for the unconstrained coefficients we encounter during the procedure (i.e. coefficients which lead to overdetermined equation) to be zero.

4.1. The space $\mathcal{B}_V$: basis functions attached to a vertex

Here, we consider basis functions attached to inner and boundary vertices (extraordinary or regular) and corner vertices. We only provide the explicit construction for bases attached to inner extraordinary and inner regular vertices, since the construction for the remaining cases is analogous.

4.1.1. Construction of bases connected to an inner EV

Given an extraordinary vertex of valence N, we have attached to it basis functions concerning their value at the vertex and their value of the first and cross derivatives at the vertex. We will figure out during the construction that these basis are in number N+3.

4.1.1.1. Basis attached to the value at the vertex. To extract the basis concerning the value at the vertex, we start solving the system (4)-(10) with initial value $\mathbf{b}_{0,0} = 1$ and continuing the construction fixing zero values for all the control points which are not constrained by any relation we will encounter during the construction. With this assumption, we can rewrite eq. (4) in the form

$$C_1\mathbf{b}_{1,0} = (2 - a_0)\mathbf{u}\,, \tag{15}$$

with $C_1 \in \mathbb{R}^{N\times N}$ defined in (14). The solution of eq. (15) will return the values concerning the first derivatives of the basis with unit value at the vertex. As shown in [27], the matrix $C_1 = -a_0 I + C + C^{N-1}$ is singular and $\mathrm{corank}(C_1) = 2$; for this reason, in order to obtain a unique solution we need to insert two extra constrains to the system (15). Let $\mathrm{Ker}(C_1) = K = \mathrm{Span}\{\mathbf{k}_1, \mathbf{k}_2\}$ be the kernel of the matrix C_1 generated by the two vectors $\mathbf{k}_1$ and $\mathbf{k}_2$, we can choose our solution to be orthogonal to the space K, i.e. $\langle \mathbf{b}_{1,0}|\mathbf{k}_1\rangle = \langle \mathbf{b}_{1,0}|\mathbf{k}_2\rangle = 0$, where $\langle\cdot|\cdot\rangle$ represents the classical Euclidean scalar product. An explicit formula to compute the kernel K has been provided in [27]. This procedure allow us to achieve a unique solution for this set of Bézier points. Going further in the resolution of the system, using the solution we just obtain for the equation (4) and the circulant matrix C, and taking also into account the constrain along the edge in eq. (10) which becomes

$$\mathbf{b}_{2,0} = \frac{1}{2}\mathbf{b}_{1,0} - \frac{1}{10}\mathbf{u}\,, \tag{16}$$

we can reorder eq. (5) as

$$C_2\mathbf{b}_{1,1} = -\frac{1}{5}a_0\mathbf{u} + 5(2 + a_0)\mathbf{b}_{1,0}\,, \tag{17}$$

with C_2 as in eq. (14). For odd values of the valence N the matrix C_2 is invertible, while for even occurrences we have $\mathrm{corank}(C_2) = 1$. To obtain a unique solution in the singular case we need to fix an extra constrain

which we decide to be the orthogonality of the expected solution $\mathbf{b}_{1,1}$ to $\mathrm{Ker}(C_2)$. For a deeper explanation of the solving strategy of (17) we refer the reader to [27]. Regarding the control points for the higher derivatives $\mathbf{b}_{2,1}$ and $\mathbf{b}_{3,1}$, from equations (6)-(7) and using again eq. (16) we come up with the relations

$$\mathbf{b}_{2,1} + \widehat{\mathbf{b}}_{1,2} = -\frac{1}{5}\mathbf{u} + \mathbf{b}_{1,0} \quad \text{and} \quad \mathbf{b}_{3,1} + \widehat{\mathbf{b}}_{1,3} = \mathbf{0} \tag{18}$$

which can be solved, for instance, imposing the extra relations $\mathbf{b}_{2,1} = \widehat{\mathbf{b}}_{1,2}$ and $\mathbf{b}_{3,1} = \widehat{\mathbf{b}}_{1,3}$. This procedure, as depicted from the construction, will return a unique basis function. In Fig. 2-(a) is presented an example of the coefficients obtained with the previous construction in case of an EV of valence $N = 3$.

4.1.1.2. Basis attached to the first derivatives at the vertex. To proceed with the Bézier extraction of this second set of basis we start again from eq. (4) but imposing this time the value $\mathbf{b}_{0,0} = 0$; this choice leads to the following homogeneous linear system

$$C_1 \mathbf{b}_{1,0} = \mathbf{0}\,, \tag{19}$$

where the matrix C_1 is the same as the previous section and introduced in (14). A solution of eq. (19) is easily given by

$$\mathbf{b}_{1,0} \in \mathrm{Ker}(C_1) = \mathrm{Span}\{\mathbf{k}_1, \mathbf{k}_2\}$$

as for eq. (15); since the kernel of the matrix C_1 is a 2-dimensional space generated by the vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ we will have two admissible solutions for the system (19) which, in fact, will end to two different bases function attached to value of the first derivative at the vertex obtained by solving the other G^1 relations starting with $\mathbf{b}_{1,0} = \mathbf{k}_1$ and $\mathbf{b}_{1,0} = \mathbf{k}_2$. The remaining constrains relating high order derivatives, taking into account the edge constrain obtained from eq. (10)

$$\mathbf{b}_{2,0} = \frac{1}{2}\mathbf{b}_{1,0}\,,$$

are given by the equations

$$\begin{aligned} C_2 \mathbf{b}_{1,1} &= \frac{1}{5}(2 + a_0)\mathbf{b}_{1,0}\,, \\ \mathbf{b}_{2,1} + \widehat{\mathbf{b}}_{1,2} &= \mathbf{b}_{1,0}\,, \\ \mathbf{b}_{3,1} + \widehat{\mathbf{b}}_{1,3} &= \mathbf{0}\,, \end{aligned}$$

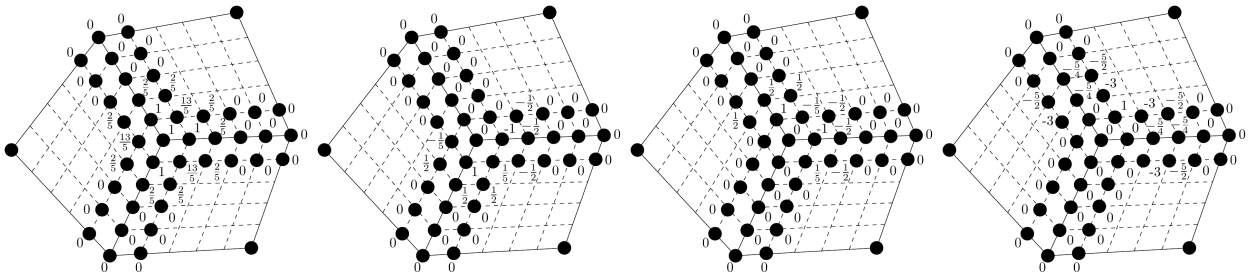
which can be solved in the same way as eq. (17) and eq. (18). Figure 2-(b) and (c) show the result given by the investigation above for an EV with $N = 3$.

4.1.1.3. Basis attached to the cross derivatives at the vertex. To a vertex of valence N correspond N cross derivatives attached to it; this means that we need to compute N bases function related to the value of the cross derivative at the vertex. Let's consider the k-patch belonging to the vertex ring. By setting the value $\mathrm{b}_{1,1}^{(k)} = 1$, from eq. (5) we realize that this point has only influence on values laying in patches $k-1$, k and $k+1$; more precisely, the points affected by this choice are only $\mathrm{b}_{2,0}^{(k)}$ and $\mathrm{b}_{2,0}^{(k+1)}$ regarding the second derivatives, which will have values equal to

$$\mathrm{b}_{2,0}^{(k)} = \mathrm{b}_{2,0}^{(k+1)} = \frac{5}{4a_0}\,,$$

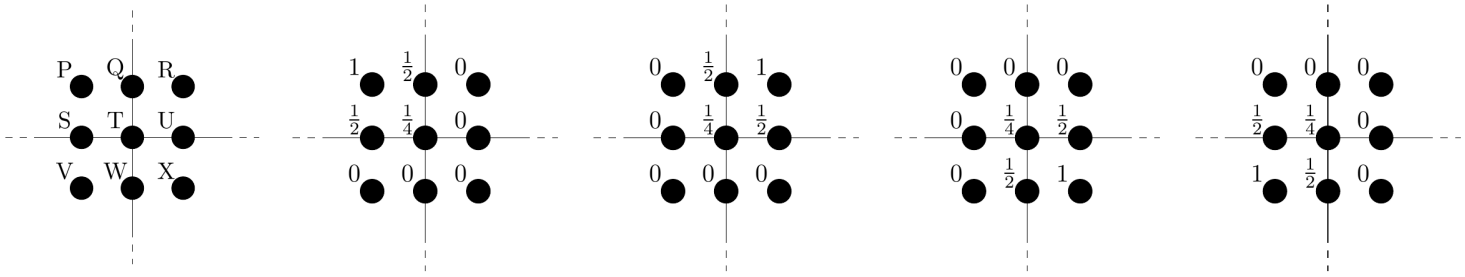
being well defined since $a_0 \neq 0 \iff N \neq 4$, that is the case of an EV we are in fact investigating. Regarding higher orders derivatives we have the points $\mathrm{b}_{3,0}^{(k)}, \mathrm{b}_{3,0}^{(k+1)}, \mathrm{b}_{1,2}^{(k-1)}, \mathrm{b}_{2,1}^{(k)}, \mathrm{b}_{1,2}^{(k)}, \mathrm{b}_{2,1}^{(k+1)}$ and $\mathrm{b}_{1,3}^{(k-1)}, \mathrm{b}_{3,1}^{(k)}, \mathrm{b}_{1,3}^{(k)}, \mathrm{b}_{3,1}^{(k+1)}$ defined by the relations

$$\begin{aligned} \mathrm{b}_{3,0}^{(k)} = \mathrm{b}_{3,0}^{(k+1)} &= \frac{5}{4a_0}\,, \\ \mathrm{b}_{1,2}^{(k-1)} + \mathrm{b}_{2,1}^{(k)} = \mathrm{b}_{2,1}^{(k)} + \mathrm{b}_{1,2}^{(k+1)} &= \frac{1}{2}\left(\frac{5}{a_0} - 1\right)\,, \\ \mathrm{b}_{1,3}^{(k-1)} + \mathrm{b}_{3,1}^{(k)} = \mathrm{b}_{3,1}^{(k)} + \mathrm{b}_{1,3}^{(k+1)} &= \frac{5}{2a_0}\,, \end{aligned}$$



(a) (b) (c) (d)

Figure 2: Coefficients for a basis function attached to the value at the vertex (a), attached to the first derivatives at the vertex (b)-(c) and attached to the cross derivatives at the vertex (d) for an EV of valence $N = 3$.



(a) (b)

Figure 3: (a): labelling of the control points around an inner regular vertex. (b): values for the Bézier points of the four basis functions.

which are obtained making use of eq. (6)-(7) and (10). Similarly to the other situation, the previous equations can be solved as in (18). Repeating the same construction for all the patches in the ring we will come up with the N basis functions attached to the cross derivatives. The result in the case $N = 3$ is shown in Fig.2-(d), while Fig. 5 presents the set of basis functions for an EV of valence $N = 5$.

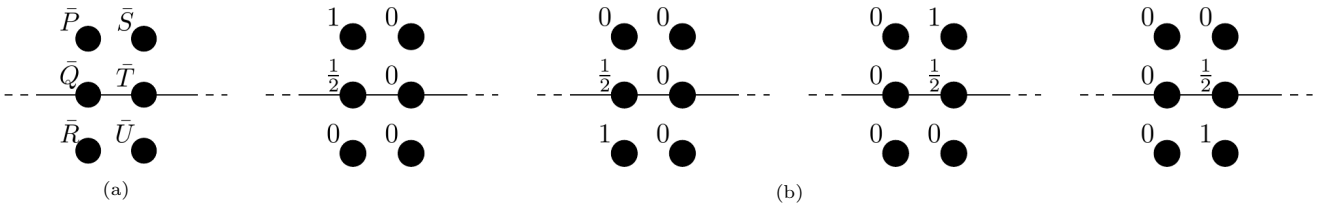
4.1.2. Bases related to an inner regular vertex

In presence of a regular vertex (RV), i.e. a vertex with valence $N = 4$, by construction we expect to have C^1 regularity for these bases; this is in fact what eq. (8)-(9) state.

First we need to investigate how many RV bases we have. To do that we first need to expand cyclically eq. (8)-(9) to all the control points around the vertex; this procedure, using the notation in Fig. 3-(a) leads to the following system:

$$\begin{pmatrix} 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{P} \\ \mathrm{Q} \\ \mathrm{R} \\ \mathrm{S} \\ \mathrm{T} \\ \mathrm{U} \\ \mathrm{V} \\ \mathrm{W} \\ \mathrm{X} \end{pmatrix} = \mathbf{0}\,. \tag{20}$$

Let D be the coefficient matrix in (20), the number of bases attached to the RV is given by $\mathrm{corank}(D) = 9 - \mathrm{rank}(D) = 9 - 5 = 4$. Hence, a choice for the Bézier points returning linearly independent bases (by construction) verifying eq. (20) is given by the coefficients in Fig. 3-(b). In presence of RV connected to an EV is also needed to modify more control points for each extraordinary edge, according with eq. (6) and (7).



(a) (b)

Figure 4: (a): labelling of the control points across a regular inner edge. (b): values for the Bézier points of the four basis functions.

4.1.3. Bases linked to extraordinary and regular boundary vertices and corners

The extraction of these types of bases is analogous to the constructions developed in the previous sections for the inner cases. Naming with κ the valence of an extraordinary boundary vertex, i.e. the number of patches attached to it, imitating the process in Section 4.1.1 we will come up with $\kappa + 3$ bases functions, which is equivalent to $N + 2$ since $\kappa = N - 1$. On the other hand, to obtain the bases bounded to a regular boundary vertex and corner we need to copy the procedure shown in Section 4.1.2: in both cases, following the same strategy leading to eq. (20), we come to get to obtain also in this case four bases functions.

4.2. The space $\mathcal{B}_E$: basis functions attached to an edge

In this second set of functions belong bases attached to inner and boundary edges, either extraordinary and regular ones. In the same way as the vertex bases we will present the explicit construction in the case of inner extraordinary and regular edge functions, whereas the construction for the remaining case comes straightforwardly.

4.2.1. Construction of bases connected to an extraordinary edge

The forming of these functions is obtained starting from eq. (6),(7) and (10). Similarly to the construction in Section 4.1.1, to extract the Bézier coefficients for the bases, we need to set zero values at the free points appearing in the equations; in this construction of bases connected to extraordinary edges the control points we need to nullify are all the points laying on the edge, i.e. $\mathbf{b}_{0,0}$, $\mathbf{b}_{1,0}$, $\mathbf{b}_{2,0}$, $\mathbf{b}_{3,0}$, $\mathbf{b}_{4,0}$ and $\mathbf{b}_{5,0}$. This assumption transforms eq. (6),(7) into

$$\begin{aligned} \mathbf{b}_{2,1} + \widehat{\mathbf{b}}_{1,2} &= \mathbf{0}\,, \\ \mathbf{b}_{3,1} + \widehat{\mathbf{b}}_{1,3} &= \mathbf{0}\,, \end{aligned} \tag{21}$$

which define the two basis functions living on an extraordinary edge. The easiest solution satisfying eq. (21) is to take $\mathbf{b}_{2,1} = \mathbf{b}_{3,1} = 1$ and $\widehat{\mathbf{b}}_{1,2} = \widehat{\mathbf{b}}_{1,3} = -1$, or viceversa.

4.2.2. Bases belonging to an inner regular edge

These bases are obtained with a similar approach as the bases in Section 4.1.2 for an inner regular vertex; the points involved in this construction are the two pairs of layers in the inner part of the edge, i.e. away from the influence of the vertices' equations. In order to determine the number of these bases, applying the C^1 constraints in eq. (8) and eq. (9) to the two layers of control points implicated in this analysis with the notation given by Fig. 4-(a) we have:

$$\begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathrm{P}} \\ \bar{\mathrm{Q}} \\ \bar{\mathrm{R}} \\ \bar{\mathrm{S}} \\ \bar{\mathrm{T}} \\ \bar{\mathrm{U}} \end{pmatrix} = \mathbf{0}\,. \tag{22}$$

If $\bar{D}$ is the matrix in (22), the number of basis functions attached to an inner regular edge is given by $\mathrm{corank}(\bar{D}) = 4$ and a set of possible solutions verifying these constraints returning linearly independent bases is given by the configurations in Fig. 4-(b).

4.2.3. Boundary edge basis functions

Since we are in presence of a boundary edge, in this situation we have no smoothness constraints to impose. Hence, the basis functions in this case are the classical bivariate Bézier polynomials obtained assigning the unit value to the four control points involved in this setting, one at the time, to obtain the four bases we were looking for.

4.3. The space $\mathcal{B}_F$: basis functions attached to a face

To conclude the construction of our space of bases we miss to the define the basis functions belonging uniquely to a single patch. As for the case of boundary edge functions, here we have no regularity conditions to impose; thus the construction is the same as in Section. 4.2.3 returning four Bézier polynomial on each face.

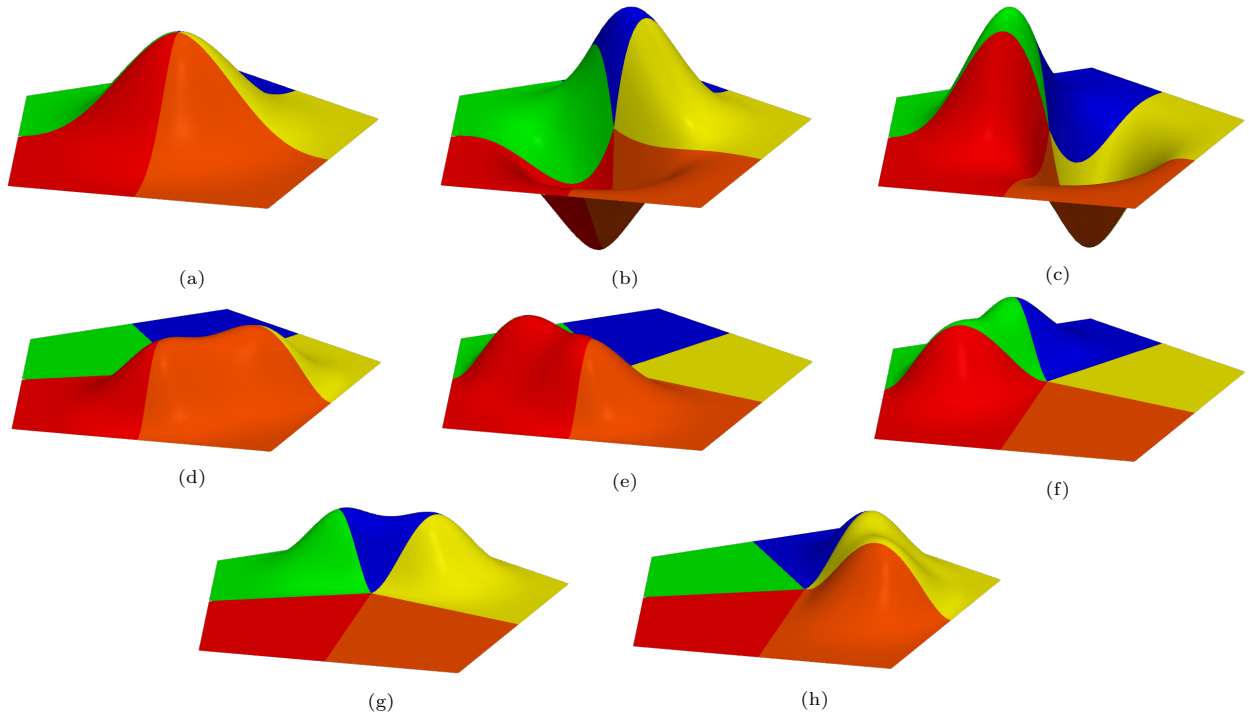


Figure 5: Basis function attached to the value of the vertex (a), value of first derivatives (b)-(c) and value of cross derivatives (d)-(e)-(f)-(g)-(h) for an EV of valence $N = 5$.

5. Analysis of the bases and space dimension

The functions we built in the previous Section 4 actually represent a set of basis functions for the G^1ACC space defined over a mesh $\mathcal{M}$, which we will call $\mathbb{G}^1(\mathcal{M})$; a proof for this statement will be provided as well as a dimension formula for the space.

Theorem 1. *The functions constructed in Section 4 form a set of bases $\mathcal{B}$ for the space $\mathbb{G}^1(\mathcal{M})$ over a generic quad mesh $\mathcal{M}$.*

Proof. In order to prove that the set $\mathcal{B}$ is effectively a basis for our space we need to prove their linearly independence and the property to be a set of generator. Since by construction the functions have different support one to the other, the linearly independence is ensured; we need thus to investigate the generating feature. Let $f \in \mathbb{G}^1(\mathcal{M})$. Since the basis functions $B_i^{F_j}$ attached to a face F_j are C^1 smooth we can define the function

$$f' := f - \sum_{j=1}^{n_F} \sum_{i=1}^{4} c_i^{F_j} B_i^{F_j}, \tag{23}$$

which is still G^1 and such that the inner face coefficients vanish. The same procedure can be applied to the C^1 basis function attached to corners $B_i^{C_j}$, boundary edges $B_i^{BE_j}$ and regular edges $B_i^{RE_j}$ which can be used to define a new function starting from (23) as

$$f^{''} := f^{'} - \sum_{j=1}^{n_C}\sum_{i=1}^{4} c_i^{C_j} B_i^{C_j} - \sum_{j=1}^{n_{BE}}\sum_{i=1}^{4} c_i^{BE_j} B_i^{BE_j} - \sum_{j=1}^{n_{RE}}\sum_{i=1}^{4} c_i^{RE_j} B_i^{RE_j} \in G^1\,. \tag{24}$$

In presence of a regular vertex connected to others regular vertices, the coefficients for the corresponding C^1 bases $B_i^{RV_j}$ are given by the solution of the homogeneous system $A\mathbf{c}^{RV} = \mathbf{0}$, with $A \in \mathbb{R}^{8\times 9}$ as in eq. (20). If a regular vertex is linked to an EV, then the linear system to be solved contains more control points to be computed and it has the form $\widetilde{A}\widetilde{\mathbf{c}}^{RV} = \widetilde{b}$, which can be rearranged to reach an homogeneous system $\bar{A}\bar{\mathbf{c}}^{RV} = \mathbf{0}$, $\bar{A} \in \mathbb{R}^{16\times 33}$ returning the coefficients for the G^1 functions $B_i^{RV_j}$ attached to these vertices. This is possible since the submatrix involving the value of the cross derivatives around a regular vertex is invertible and giving back the null solutions; then, because of the C^1 constraints, having zero values for the cross points leads to zero values for the entire set of control points involved around a regular vertex. A similar construction allow us to obtain the coefficients for the G^1 functions belonging to boundary EVs $B_i^{BEV_j}$ and boundary regular vertices $B_i^{BRV_j}$. Thus, from eq. (24) we can define the function

$$f^{'''} := f^{''} - \sum_{j=1}^{n_{RV}}\sum_{i=1}^{4} c_i^{RV_j} B_i^{RV_j} - \sum_{j=1}^{n_{BRV}}\sum_{i=1}^{4} c_i^{BRV_j} B_i^{BRV_j} - \sum_{j=1}^{n_{BEV}}\sum_{i=1}^{N_{BEV_j}+2} c_i^{BEV_j} B_i^{BEV_j}\,, \tag{25}$$

which is still a G^1 smooth function and such that has zero coefficients in the previously treated regions. Now, the function $f^{'''}$ has only non-zero coefficients around extraordinary vertices and their extraordinary edges. In these regions we can also compute G^1 functions $B_i^{EV_j}$ whose coefficients are obtained following equations (4)-(10) or, equivalently, system (13), by using the extraction procedure we explained in detail in Section 4. Finally, using (25) we can define the new function

$$f^{''''} := f^{'''} - \sum_{j=1}^{n_{EV}}\sum_{i=1}^{3N_{EV_j}+3} c_i^{EV_j} B_i^{EV_j} \equiv 0\,,$$

which concludes the proof.

□

As consequence of the structure of the bases set $\mathcal{B}$ we have the following Corollary returning a formula for the dimension of our spline space

Corollary 1. *The space $\mathbb{G}^1(\mathcal{M})$ has dimension given by:*

$$\begin{aligned}\dim\left(\mathbb{G}^1(\mathcal{M})\right) &= \sum_{i=1}^{n_V}\dim\left(\mathcal{B}_{V_i}\right) + \sum_{i=1}^{n_E}\dim\left(\mathcal{B}_{E_i}\right) + \sum_{i=1}^{n_F}\dim\left(\mathcal{B}_{F_i}\right)\\ &= \sum_{i=1}^{n_{EV}} N_{EV_i} + 3n_{IEV} + 2(n_{BEV} + n_{EE}) + 4(n_{RE} + n_{BE} + n_{RV} + n_C + n_F)\,.\end{aligned}$$

Proof. The proof is obtained using the decomposition in (11) and summing up all the basis functions for each feature of the mesh shown in Sections 4.1, 4.2 and 4.3. □

6. Numerical experiments

We will present now some numerical experiments in which we apply the basis functions $B_i(u,v)$ we constructed in Section 4 for point cloud fitting problems. The setup of our investigation is the classical least square fitting problem: given a point cloud $\mathcal{P}$, i.e. a set of points $P_i \in \mathbb{R}^3$, $i = 1,\dots,n_P$, with associated

parameters $(u_i, v_i) \in \mathbb{R}^2$, we want to find the coefficients $c_i \in \mathbb{R}^3$ of a Bézier surface $Q(u,v) = \sum_i c_i B_i(u,v)$ such that the quantity

$$F = \sum_{k=1}^{n_P} \|Q(u_k, v_k) - P_k\|_2^2 + \lambda E_{thin}\,, \quad \lambda \geq 0\,, \tag{26}$$

is minimal. We also take into account in our minimization problem (26) an energy term given by the standard thin-plate energy

$$E_{thin} = \iint_{[0,1]^2} \|Q_{uu}\|_2^2 + 2\|Q_{uv}\|_2^2 + \|Q_{vv}\|_2^2 \, du\, dv\,, \tag{27}$$

which is controlled by the smoothing parameter λ. The minimization of the functional in (27) is responsible for a smoothing effect on the final surface Q. Moreover, following the construction presented in [14], we perform some iterations of parameter correction to further reduce the approximating error. The point clouds used in the following numerical experiments are manufactured data, i.e. they are obtained evaluating certain input functions or surfaces over a certain domain identified by a known mesh $\mathcal{M}$ on which we construct our spline space $\mathbb{G}^1(\mathcal{M})$.

After having computed a least square surface Q from a point cloud $\mathcal{P}$ let define the array of errors $\mathbf{err} = \{err_i\}_{i=1}^{nP}$ whose entries are the quantities $err_i = \|Q(u_i, v_i) - P_i\|_2$, i.e. the Euclidean distance (ℓ_2 norm) from each point of the cloud and the correspondent value on the surface evaluated in its parameter. From this we define three quantities which will be use as indicator of accuracy of the fitting, namely the smallest and biggest error and the root mean squared error (RMSE in short); more precisely:

$$\underline{m} := \min_{i=1,\dots,n_P} err_i\,, \quad L^\infty := \max_{i=1,\dots,n_P} err_i\,, \quad \text{RMSE} := \sqrt{\frac{1}{n_P}\sum_{i=1}^{n_P} err_i^2}\,. \tag{28}$$

6.1. Point cloud by analytic function evaluation

We present here experiments which are developed using point cloud data obtained by sampling an input function $f(x,y)$ over a domain identified by a polygonal mesh $\mathcal{M}$. In Example 1 we focus the attention on the goodness of the fitting showing how it improves when the number of basis functions increases, even in presence of a sophisticated point cloud and without the need to use extra smoothness constraints. On the other hand, Example 2 exhibit the power of our construction when smoothing is needed in order to obtain an optimal result, but without increasing the dimension of the spline space.

Example 1. For this test, the point cloud is obtained evaluating the function

$$f_T(x,y) = \frac{y}{2\left(\cos(4(x^2+y-1))\right)}\,, \quad (x,y) \in \mathcal{M}_T \tag{29}$$

over a triangular mesh $\mathcal{M}_T$ formed by 3 patches which presents an EV of valence $N = 3$; our sampling produced a point cloud formed by 150528 points. This example uses no smoothing property, which means we fix $\lambda = 0$. From the result in Figure 6-(a) can be noticed that constructing our spline space over a coarse mesh as $\mathcal{M}_T$ leads to a fitting surface which is not approximating in a proper way our point set, due to its very oscillating behavior. This issue can be solved increasing the number of bases generating the spline space, i.e. increasing the number of patches defining the polygonal domain; in this example the mesh has been refined via Catmull-Clark subdivision. Fig. 6 show how the quality of the fitting increases when more basis functions are involved, while in Tab. 1 and Fig. 7 are respectively listed and plotted the errors computed from (28) for 5 refinement levels.

Example 2. Here, the cloud data is derived sampling the function

$$f_E(x,y) = \sum_{\nu\in\{-3,0,3\}} \frac{2}{3e^{\sqrt{(10x+\nu)^2+(10y+\nu)^2}}}\,, \quad (x,y) \in \mathcal{M}_E\,, \tag{30}$$

where $\mathcal{M}_E$ is an hexagonal mesh composed of 96 patches identifying an EV of valence $N = 6$ in its middle. The point cloud we obtain is formed of 153600 points. Here we show the power of the smoothing property of

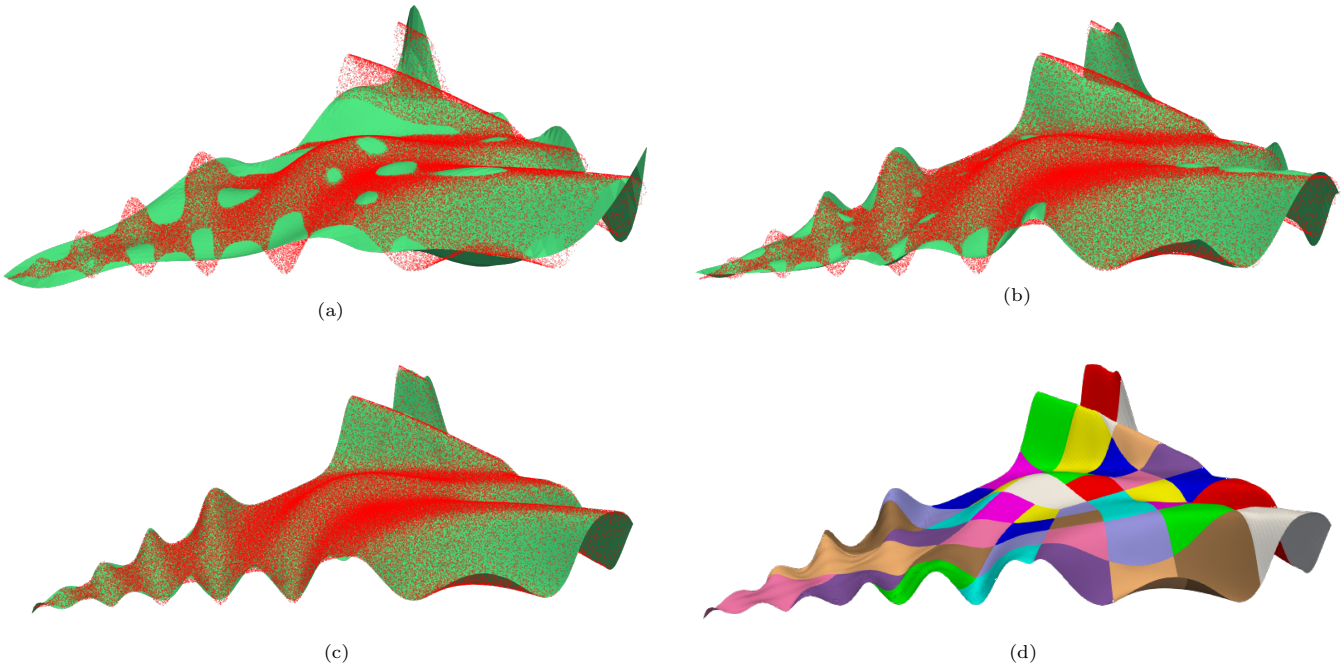


Figure 6: (a)-(c): approximating surfaces (green) obtained from the same point cloud by sampling the function f_T in (29) (red) by using, respectively, 72, 240 and 864 basis function. (d)- multipatch coloring of the surface in (c).

$\mathbf{dim}\left(\mathbb{G}^{\mathbf{1}}(\boldsymbol{\mathcal{M}}_T)\right)$	72	240	864	3264	12672
$\underline{\boldsymbol{m}}$ error	2.331e-05	2.070e-06	2.836e-08	5.328e-11	3.364e-14
$\boldsymbol{L}^{\infty}$ error	0.364e-00	0.122e-00	0.834e-01	0.178e-01	0.790e-02
RMSE	0.491e-01	0.149e-01	0.335e-02	0.353e-03	0.570e-06

Table 1: Minimal error $\underline{m}$, maximal error L^{∞} and RMSE for the surfaces in Example 1 obtained under 5 Catmull-Clark subdivision steps from a point cloud of 150528 elements. The geometry in Fig. 6 is contained in a box whose biggest length is 3.

this construction: differently from Example 1, fixing the number of patches i.e. the number of basis function, which is equal to 1725 in our case, we compute the fitting surface increasing the smoothing parameter λ from 10^{-3} to 10^{-1} with a step of 10^{-1}. From Fig. 8-(a) we noticed that the fitting surface presents several wrinkles around the middle peak; by increasing the smoothing factor λ we recover regularity in the output function which presents no more irregularities, as can be noticed in Fig. 8-(d). As expected, this procedure will produce at every iteration a smoother function than the previous, but on the other side this forced regularity constraint is reflected in an increase of the three errors in (28); this phenomenon is shown in Tab. 2 and graphically in Fig. 9.

$\boldsymbol{\lambda}$	0	10^{-3}	10^{-2}	10^{-1}
$\underline{\boldsymbol{m}}$ error	1.248e-11	1.457e-09	2.308e-09	3.260e-08
$\boldsymbol{L}^{\infty}$ error	0.164e-00	0.165e-00	0.170e-00	0.196e-00
RMSE	0.308e-02	0.310e-02	0.312e-02	0.375e-02

Table 2: Minimal error $\underline{m}$, maximal error L^{∞} and RMSE for the surfaces in Example 1 for the surfaces in Example 2 computed making use of 1725 basis functions and progressively bigger smoothing parameter λ. The biggest length of the box containing the model in Fig. 8 is 2.

Example 3. In this last example, the point cloud we are going to use is obtained by evaluating a trivariate function defined over the unit sphere $\mathcal{S}^2$. More precisely, the points are achieved from the function

$$f_{\mathcal{S}^2}(x,y,z) = \max\{0, \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)\} + 1\,, \quad (x,y,z) \in \mathcal{S}^2\,. \tag{31}$$

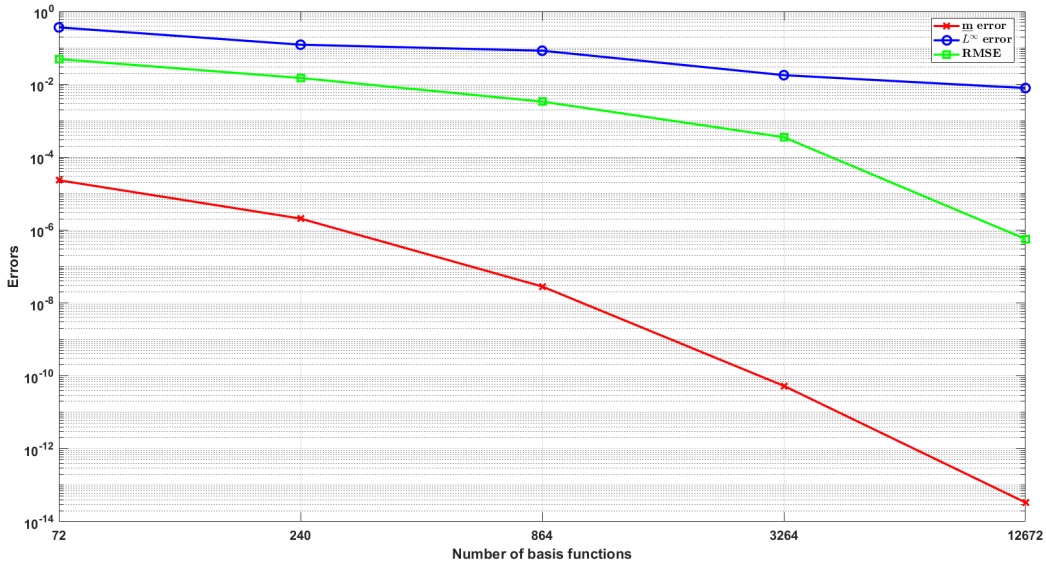


Figure 7: Experimental behavior obtained from the errors represented in Table 1.

The data we get is composed of 540000 points, while the set of basis functions has been built over a quad mesh, $\mathcal{M}_{\mathcal{S}^2}$, approximating the unit sphere composed of 96 faces. We notice from the error colorplot in Fig. 10-(d) that, understandably, the regions in which the error is concentrate involve the peaks present in the point cloud and the lower part of them where there is, in fact, a noticeable slope created by the presence of the max function in (31). Table 3 shows the numerical results for the errors.

	$\dim\left(\mathbb{G}^1(\mathcal{M}_{\mathcal{S}^2})\right)$	$\underline{m}$ error	L^∞ error	RMSE
$f_{\mathcal{S}^2}$	1512	1.028e-05	0.539e-01	0.107e-01

Table 3: Spline space dimension, minimal error $\underline{m}$, maximal error L^∞ and RMSE for the fitting presented in Example 3. The point cloud in Fig. 10 is surrounded by a box with longest size 3.

6.2. Point cloud from ACC surfaces

We provide now fitting example which are obtained from a very big data set. The point clouds utilized in this section are provided by randomly sampling the Approximate Catmull-Clark surfaces obtained from the construction in [24]. Figures from 11 to 16 show the data we use for our investigation: the dimension of the clouds goes from a minimum of 549180 to a maximum of 914628 points. In all the experiment presented here we do not consider any smoothing parameter, i.e. $\lambda = 0$, but some iterations of parameter correction [14] will be performed to optimize the fitting error; the errors in Table 4 are obtained using the formulas in (28).

	bird	dinosaur	hammer	hand	rabbit	venus
$\dim\left(\mathbb{G}^1(\mathcal{M})\right)$, n_P	100386, 549180	44311, 675000	44347, 675000	68034, 557568	80003, 914628	44306, 588000
$\underline{m}$ error	2.139e-07	1.244e-06	2.477e-06	2.390e-06	1.143e-06	1.734e-06
L^∞ error	0.138e-00	0.537e-01	0.110e-00	0.666e-01	0.873e-01	0.993e-01
RMSE	0.325e-02	0.164e-02	0.280e-02	0.308e-02	0.291e-02	0.260e-02
3 × parameter correction						
$\underline{m}$ error	1.371e-10	8.740e-10	5.445e-09	3.427e-08	3.477e-10	1.416e-09
L^∞ error	0.754e-03	0.336e-03	0.451e-01	0.675e-02	7.688e-05	0.455e-03
RMSE	6.482e-06	5.392e-06	0.183e-03	0.213e-03	2.195e-06	1.035e-05

Table 4: Minimal error $\underline{m}$, maximal error L^∞ and RMSE for the fitting examples presented in Fig. 11 to 16 with a description of the spline space and point cloud features. All the above mentioned models are contained in a box whose longest length is 100.

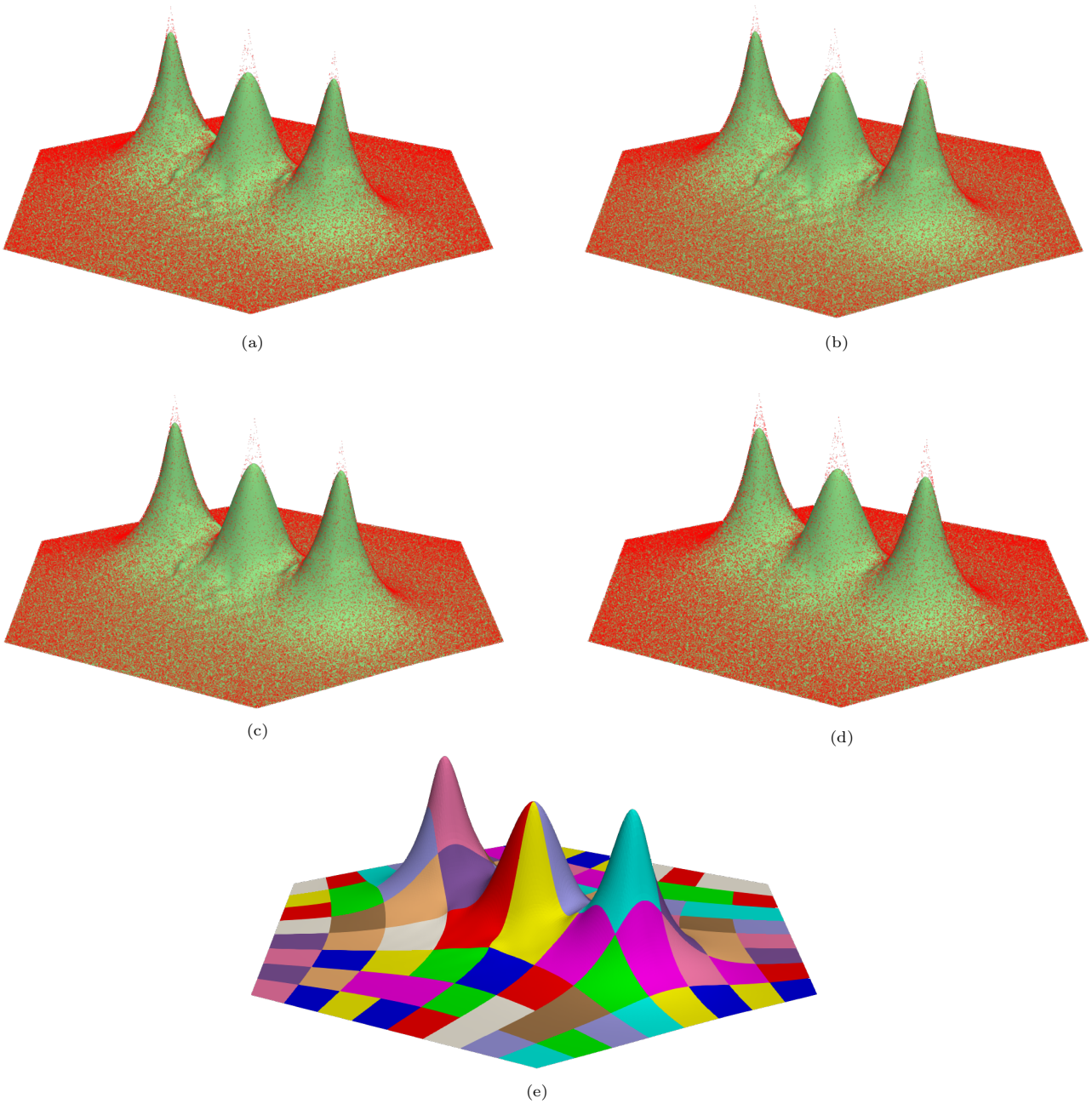


Figure 8: (a)-(d): approximating surfaces (green) obtained by using the same number of basis function and point cloud computed from f_E in (30) (red) with smoothing parameter $\lambda = 0$, $\lambda = 10^{-3}$, $\lambda = 10^{-2}$ and $\lambda = 10^{-1}$, respectively. (e)-multipatch coloring of the surface in (d).

7. Conclusion

In this work we provided an explicit construction of a globally biquintic set of basis function generating the G^1ACC space [27] with G^1 smoothness around extraordinary vertices and at least C^1 elsewhere. The Bézier points defining the multipatch basis functions are explicitly computed starting from the equations defining the space, which are obtained with the use of quadratic gluing data functions defined over the extraordinary edges; afterwards, we provide an analysis of the achieved bases set as well as a dimension formula for the

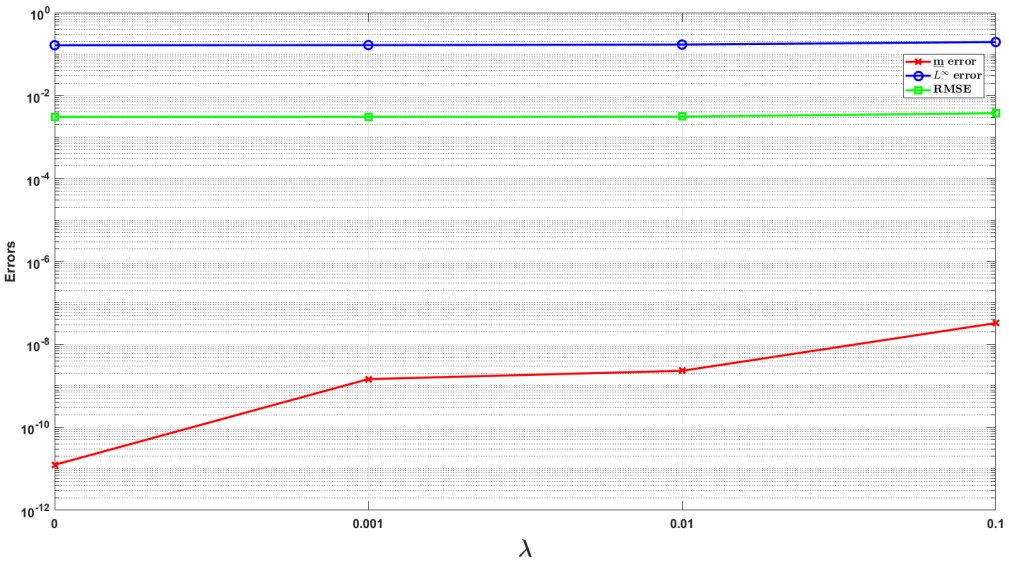
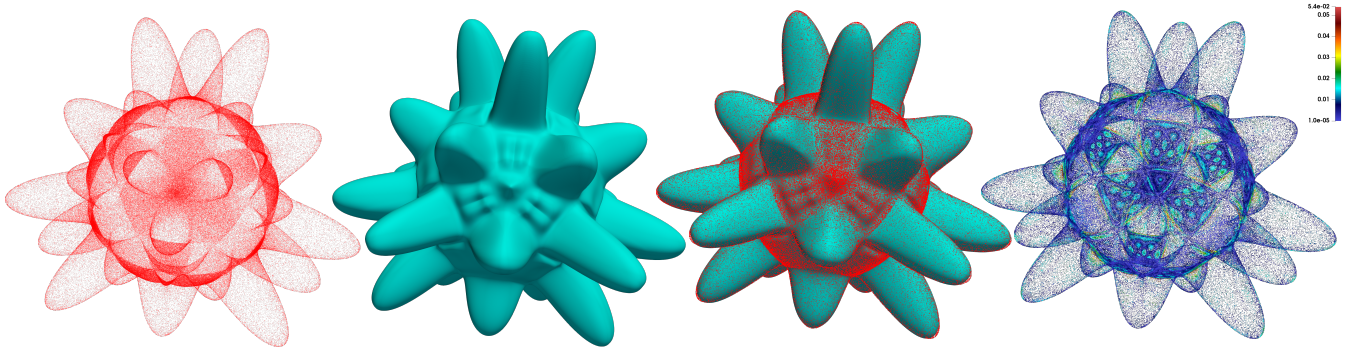


Figure 9: Experimental behavior obtained from the errors represented in Table 2.



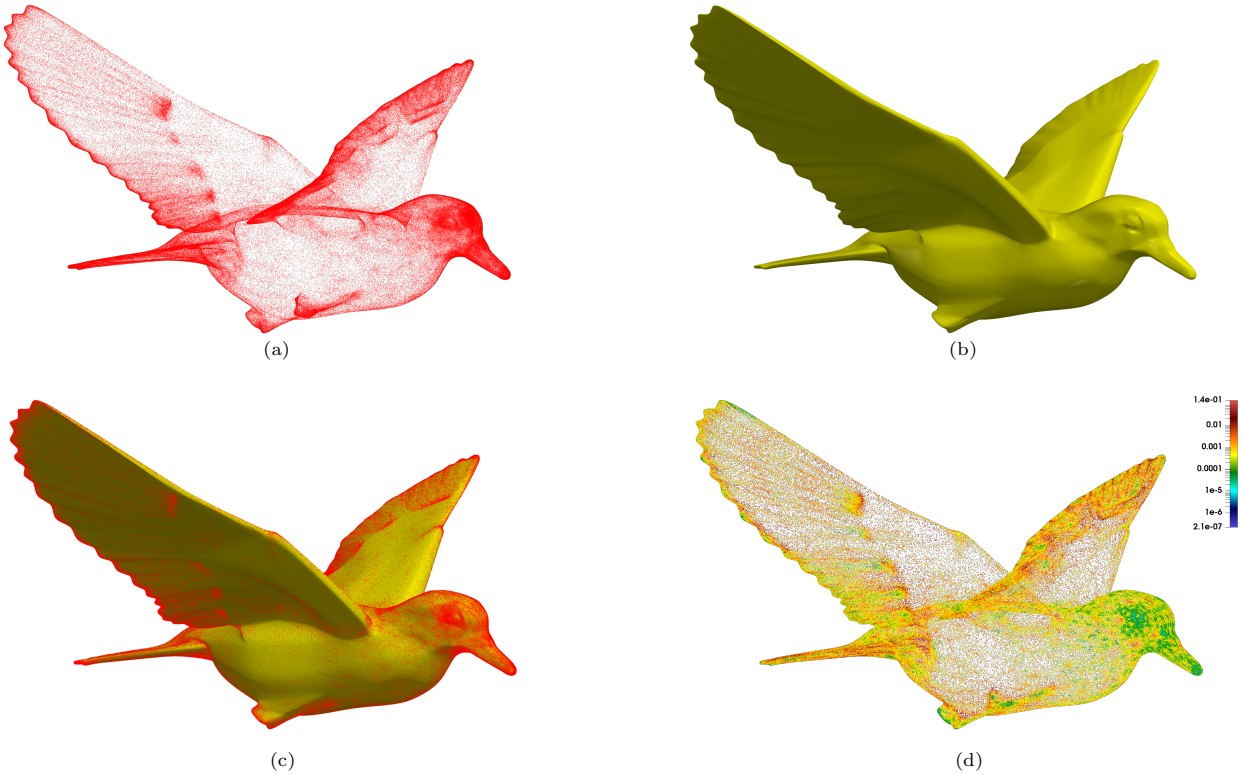
(a) (b) (c) (d)

Figure 10: (a): point cloud obtained sampling the trivariate function defined in (31) over the unit sphere $\mathcal{S}^2$. (b): final least square surface. (c): comparison between point cloud and final surface. (d): colorplot of the ℓ_2 distances contained in **err**.

G^1 spline space. Finally, several exhaustive numerical experiments are provided to demonstrate the quality of the basis functions in point cloud data fitting problems of any type. All the results presented throughout the paper have been obtained by implementing codes using Julia language for the computation of the basis functions and G+Smo library (cf. [26]) for the numerical simulations; it has also been noticed that the whole construction is computationally efficient, even in presence of point clouds with complex geometries and elevated amount of data. An interesting future research we are interested in regards isogeometric analysis environment. More precisely, we would like to investigate and provide an extension of the Bézier bases set presented in this paper to splines patches with inner knots in order to obtain nested spaces of analysis-suitable basis functions to be used in the numerical solving of PDEs.

8. Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 860843.



(a) (b) (c) (d)

Figure 11: Bird. (a): point cloud obtained sampling an ACC surface. (b): final least square surface. (c): comparison between point cloud and final surface. (d): colorplot of the ℓ_2 distances contained in **err** in logarithmic scale.

References

[1] Peter Alfeld and Larry Schumaker. The dimension of bivariate spline spaces of smoothness r for degree $d \geq 4r+1$. *Constr. Approx.*, 3(2):189–197, 1987.

[2] Nina Amenta, Marshall Bern, and Manolis Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98*, pages 415–421. ACM Press, 1998.

[3] Matthew Berger, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Gael Guennebaud, Joshua Levine, Andrei Sharf, and Claudio Silva. A Survey of Surface Reconstruction from Point Clouds. *Computer Graphics Forum*, page 27, 2016.

[4] Ahmed Blidia, Bernard Mourrain, and Nelly Villamizar. G^1-smooth splines on quad meshes with 4-split macro-patch elements. *Computer Aided Geometric Design*, 52-53:106–125, 2017.

[5] Ahmed Blidia, Bernard Mourrain, and Gang Xu. Geometrically smooth spline bases for data fitting and simulation. *Comput. Aided Geom. Design*, 78:101814, 15, 2020.

[6] Georges-Pierre Bonneau and Stefanie Hahmann. Flexible G^1 interpolation of quad meshes. *Graphical Models*, 76(6):669–681, 2014.

[7] Alexandre Durupt, Sébastien Remy, Guillaume Ducellier, and Benoit Eynard. From a 3d point cloud to an engineering CAD model: a knowledge-product-based approach for reverse engineering. *Virtual and Physical Prototyping*, 3(2):51–59, 2008.

[8] Gerald Farin, Josef Hoschek, and Myung-Soo Kim. *Handbook of computer aided geometric design*. Elsevier, 2002.

[9] Michael S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.

[10] David R. Forsey and Richard H. Bartels. Surface fitting with hierarchical splines. *ACM Transactions on Graphics*, 14(2):134–161, 1995.

[11] Stefanie Hahmann, Georges-Pierre Bonneau, and Baptiste Caramiaux. Bicubic G1 interpolation of irregular quad meshes using a 4-split. In *International Conference on Geometric Modeling and Processing*, pages 17–32. Springer, 2008.

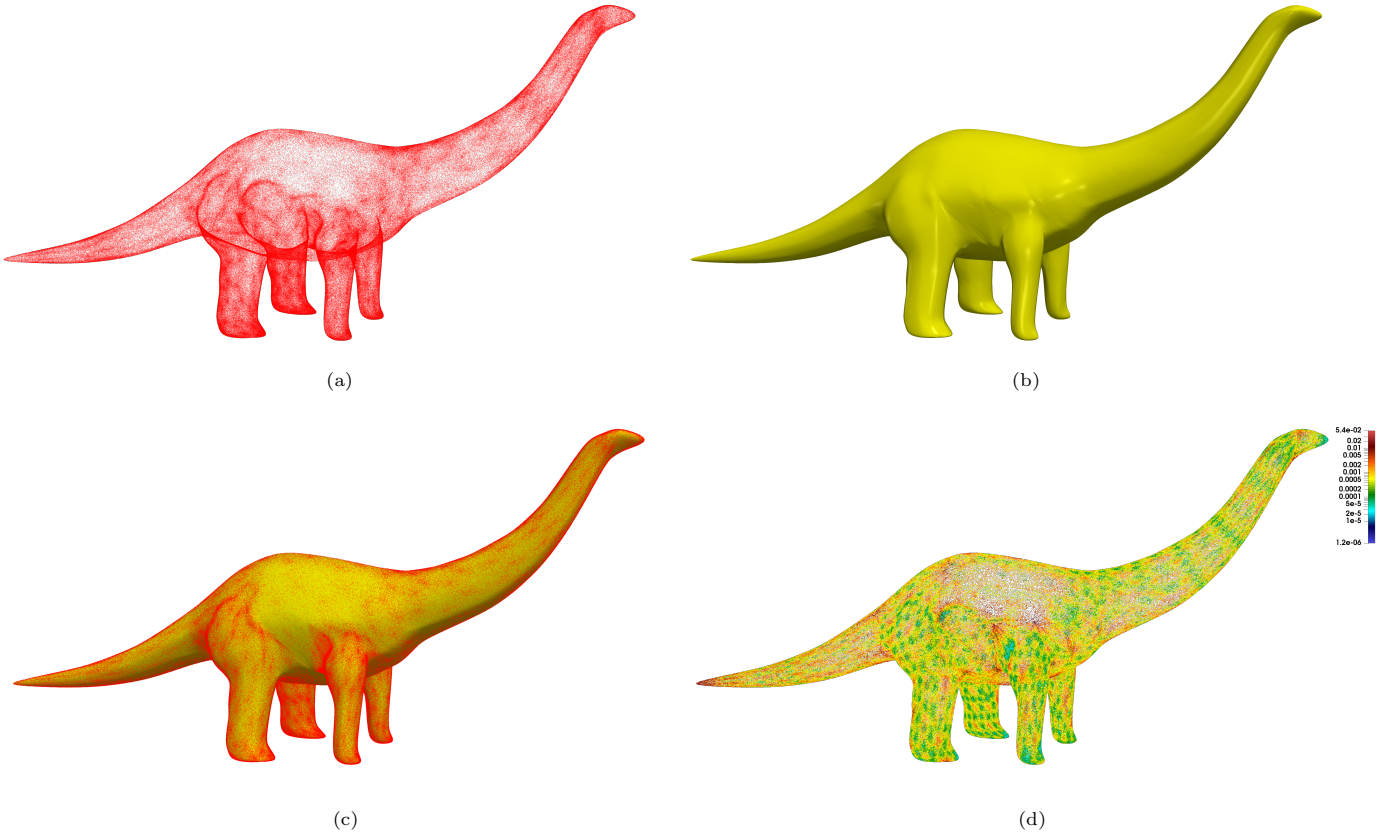
[12] Jörg M. Hahn. Geometric continuous patch complexes. *Computer Aided Geometric Design*, 6(1):55–67, 1989-02-01.

[13] Dong Hong. Spaces of bivariate spline functions over triangulation. *Approx. Theory Appl.*, 7(1):56–75, 1991.

[14] Josef Hoschek. Intrinsic parametrization for approximation. *Computer Aided Geometric Design*, 5(1):27–31, 1988.

[15] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. Dimension and basis construction for analysis-suitable G^1 two-patch parameterizations. *Computer Aided Geometric Design*, 52-53:75–89, 2017.

[16] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. Isogeometric analysis with C^1 functions on planar, unstructured quadrilateral meshes. *The SMAI Journal of computational mathematics*, S5:67–86, 2019.



(a) (b)

(c) (d)

Figure 12: Dinosaur. (a): point cloud obtained sampling an ACC surface. (b): final least square surface. (c): comparison between point cloud and final surface. (d): colorplot of the ℓ_2 distances contained in **err** in logarithmic scale.

[17] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. An isogeometric C^1 subspace on unstructured multi-patch planar domains. *Computer Aided Geometric Design*, 69:55–75, 2019.

[18] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013.

[19] Gaël Kermarrec, Niklas Schild, and Jan Hartmann. Fitting Terrestrial Laser Scanner Point Clouds with T-Splines: Local Refinement Strategy for Rigid Body Motion. *Remote Sensing*, 13(13):2494, 2021.

[20] Gábor Kiss, Carlotta Giannelli, Urška Zore, Bert Jüttler, David Großmann, and Johannes Barner. Adaptive CAD model (re-)construction with THB-splines. *Graphical Models*, 76(5):273–288, 2014.

[21] Ming-Jun Lai and Larry Schumaker. *Spline functions on triangulations*, volume 110 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2007.

[22] Xin Li, Jiansong Deng, and Falai Chen. Surface modeling with polynomial splines over hierarchical T-meshes. *The Visual Computer*, 23(12):1027–1033, 2007.

[23] Hongwei Lin, Wei Chen, and Hujun Bao. Adaptive patch-based mesh fitting for reverse engineering. *Computer-Aided Design*, 39(12):1134–1142, 2007.

[24] Charles Loop and Scott Schaefer. Approximating Catmull-Clark subdivision surfaces with bicubic patches. *ACM Transactions on Graphics*, 27(1):8:1–8:11, 2008.

[25] Chenlei Lv, Weisi Lin, and Baoquan Zhao. Voxel structure-based mesh reconstruction from a 3d point cloud. *IEEE Transactions on Multimedia*, 24:1815–1829, 2022.

[26] Angelos Mantzaflaris. An Overview of Geometry Plus Simulation Modules. In *Mathematical Aspects of Computer and Information Sciences*, Lecture Notes in Computer Science, pages 453–456, Cham, 2020. Springer International Publishing.

[27] Michelangelo Marsala, Angelos Mantzaflaris, and Bernard Mourrain. G1 – smooth biquintic approximation of catmull-clark subdivision surfaces. *Computer Aided Geometric Design*, 99:102158, 2022.

[28] Bernard Mourrain, Raimundas Vidunas, and Nelly Villamizar. Geometrically continuous splines for surfaces of arbitrary topology. *Computer Aided Geometric Design*, 45:108–133, 2016.

[29] Bernard Mourrain and Nelly Villamizar. Homological techniques for the analysis of the dimension of triangular spline spaces. *J. Symbolic Comput.*, 50:564–577, 2013.

[30] Shayan Nikoohemat, Abdoulaye A. Diakité, Sisi Zlatanova, and George Vosselman. Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management. *Automation in Construction*, 113:103109, 2020.

[31] Florent Poux, Roland Billen, Jean-Paul Kasprzyk, Pierre-Henri Lefebvre, and Pierre Hallot. A built heritage information

Figure 13: Hammer. (a): point cloud obtained sampling an ACC surface. (b): final least square surface. (c): comparison between point cloud and final surface. (d): colorplot of the ℓ_2 distances contained in **err** in logarithmic scale.

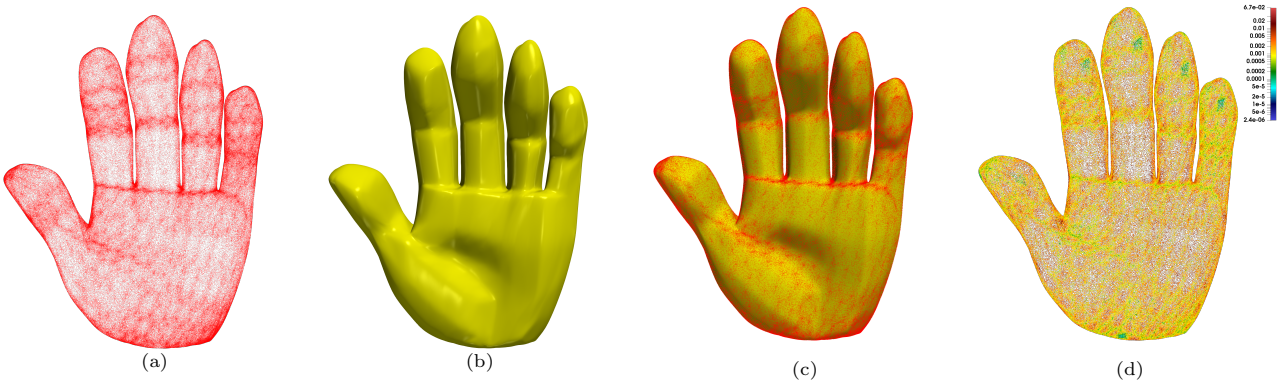


Figure 14: Hand. (a): point cloud obtained sampling an ACC surface. (b): final least square surface. (c): comparison between point cloud and final surface. (d): colorplot of the ℓ_2 distances contained in **err** in logarithmic scale.

system based on point cloud data: HIS-PC. *ISPRS International Journal of Geo-Information*, 9(10):588, 2020.

[32] Larry Schumaker. Bounds on the dimension of spaces of multivariate piecewise polynomials. *Rocky Mountain J. Math.*, 14(1):251–264, 1984.

[33] Martin Sinko, Patrik Kamencay, Robert Hudec, and Miroslav Benco. 3d registration of the point cloud data using ICP algorithm in medical image analysis. In *2018 ELEKTRO*, pages 1–6, 2018-05.

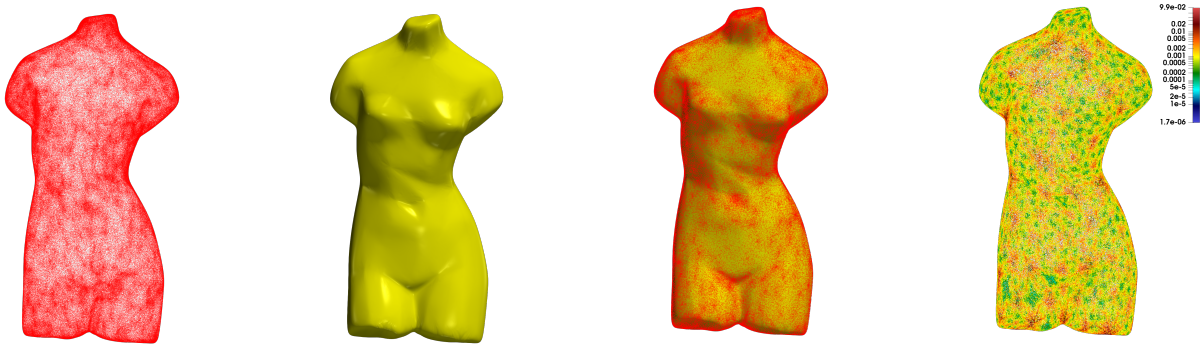
[34] A. Sitek, R.H. Huesman, and G.T. Gullberg. Tomographic reconstruction using an adaptive tetrahedral mesh defined by a point cloud. *IEEE Transactions on Medical Imaging*, 25(9):1172–1179, 2006-09.

[35] George Sithole and George Vosselman. Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(1):85–101, 2004-08-01.

[36] Vibeke Skytt, Gaël Kermarrec, and Tor Dokken. LR B-splines to approximate bathymetry datasets: An improved statistical criterion to judge the goodness of fit. *International Journal of Applied Earth Observation and Geoinformation*, 112:102894, 2022.

(a) (b) (c) (d)

Figure 15: Rabbit. (a): point cloud obtained sampling an ACC surface. (b): final least square surface. (c): comparison between point cloud and final surface. (d): colorplot of the ℓ_2 distances contained in **err** in logarithmic scale.



(a) (b) (c) (d)

Figure 16: Venus. (a): point cloud obtained sampling an ACC surface. (b): final least square surface. (c): comparison between point cloud and final surface. (d): colorplot of the ℓ_2 distances contained in **err** in logarithmic scale.

[37] Qian Wang and Min-Koo Kim. Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics*, 39:306–319, 2019.

[38] Wenping Wang, Helmut Pottmann, and Yang Liu. Fitting b-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. Graph.*, 25:214–238, 2006.

[39] Yan Wang, Zixian Fan, and Yingchun You. Application research of earth volume calculation based on 3d laser point cloud data. *IOP Conference Series: Materials Science and Engineering*, 780(3):032050, 2020-03.

[40] Ziwei Wang, Sijie Yan, Long Wu, Xiaojian Zhang, and BinJiang Chen. Robust point clouds registration with point-to-point lp distance constraints in large-scale metrology. *ISPRS Journal of Photogrammetry and Remote Sensing*, 189:23–35, 2022-07-01.

[41] P.J. Watt and D. N. M. Donoghue. Measuring forest structure with terrestrial laser scanning. *International Journal of Remote Sensing*, 26(7):1437–1446, 2005.

[42] Volker Weiss, Làszló. Andor, Gabor Renner, and Tamas Várady. Advanced surface fitting techniques. *Computer Aided Geometric Design*, 19(1):19–42, 2002.

[43] Linfu Xie, Han Hu, Qing Zhu, Xiaoming Li, Shengjun Tang, You Li, Renzhong Guo, Yeting Zhang, and Weixi Wang. Combined rule-based and hypothesis-based method for building model reconstruction from photogrammetric point clouds. *Remote Sensing*, 13(6):1107, 2021-01.

[44] Jianing Yao, Alexander Anderson, and Jannick P. Rolland. Point-cloud noncontact metrology of freeform optical surfaces. *Optics Express*, 26(8):10242–10265, 2018-04-16.

[45] Xiuyang Zhao, Caiming Zhang, Li Xu, Bo Yang, and Zhiquan Feng. IGA-based point cloud fitting using b-spline surfaces for reverse engineering. *Information Sciences*, 245:276–289, 2013-10-01.