Introduction

The digital representation of geometric objects belonging to the real world is of fundamental importance in many scientific disciplines and related applications areas. For example, they can be used for simulations, which involve numerical computations directly on the model or its components, as well as for visualization and direct measurements during the design or manufacturing phases.

The input information typically comes in the form of *discrete* data, usually collected as 2D/3D point clouds in out setting, hence a *continuous* geometric model needs to be (re-)constructed. Continuous representations facilitate the shaping and manipulation procedures since the dimensionality and complexity of the considered geometric objects are reduced.

The final *goal* of this Thesis consists in developing *automatic*, *robust*, *highly-accurate*, and *efficient* methods for the representations of the input discrete data via a continuous geometric model. To this aim, we properly combine free-form geometric models with data-driven schemes, hence merging *Computer Aided Geometric Design (CAGD)* [76, 44] methods with *Deep Learning (DL)* [64, 19] technologies.

0.1. Computer Aided Geometric Design

CAGD provides the mathematical and computational methods as well as suitable modelling and/or approximation schemes for the design, construction, and analysis of geometric objects, i. e. free-form curves, surfaces, and volumes. Examples of its applications are Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM), Geometric Modelling, Robotics, Computer Vision, Computer Graphics, Pattern Recognition, Image Processing, and Scientific Visualization. Key computational tools in this settings are splines, meshes, and subdivision schemes [76, 45, 44], among others.

Splines are piecewise polynomial functions characterized by a certain regularity. The current CAD standard for splines representations relies on B-splines [11] and their non-uniform rational extension, i. e. Non Uniform Rational B-Spline (NURBS) [128, 136]. The multivariate case for standard CAD spline representations relies on the *tensor-product* structure. This construction poses several limitations when we want to properly include *local* properties in the geometric model. More specifically, tensor-product

Sofia Imperatore, Eindhoven University of Technology, the Netherlands, s.imperatore@tue.nl, 0009-0003-9116-9978

Referee List (DOI 10.36253/fup_referee_list)
FUP Best Practice in Scholarly Publishing (DOI 10.36253/fup_best_practice)

Sofia Imperatore, *Introduction*, © Author(s), CC BY 4.0, DOI 10.36253/979-12-215-1002-7.02, in Sofia Imperatore, *Adaptive spline approximation: data-driven parameterization and CAD model (re-)construction*, pp. 3-11, 2026, published by Firenze University Press, ISBN 979-12-215-1002-7, DOI 10.36253/979-12-215-1002-7

Book References DOI 10.36253/979-12-215-1002-7.references

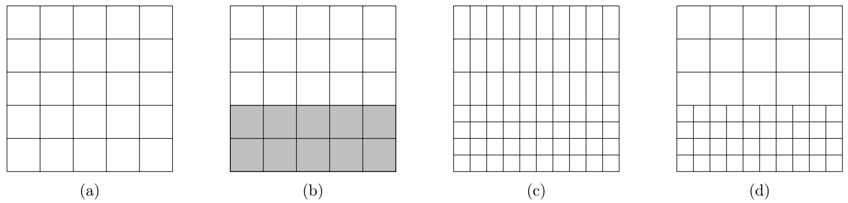

Figura 1. Given an initial tensor-product grid (a) and a region of interest for refinement (b), the refined tensor-product grid (c) propagates the refinement beyond the marked areas, while the adaptive model with T-junctions enables local refinement capabilities (d).

constructions allow only *global* refinement. Hence, unnecessary degrees of freedom are added in regions far away from the one of interest, and this also usually increases the computational costs, see Chapter 1 (a–b–c). The need to overcome the limits of tensor-product spline constructions, the achieve enhanced flexibility and efficient computation of high-quality approximations, led to the development of new *adaptive* spline spaces defined on extensions of tensor-product constructions on T-meshes, which also include T-junctions and enable local refinement capabilities, see Chapter 1 (a–b, d).

Spline adaptivity in this setting can be achieved with different approaches, and several proposals can be found in the literature. Among others, we mention: multilevel B-splines [96], defined by a coarse-to-fine hierarchy of control lattices to generate a sequence of B-spline functions, whose sum gives the final model; T-splines [146, 145], defined on meshes where T-junctions between axis-aligned segments are allowed, and related modified versions [80, 100, 163, 102]; S-splines [101], a generalization of T-splines that solves the problem of additional control points propagation in T-spline's local refinement algorithms; PHT-splines [35], namely polynomial splines over hierarchical T-meshes; LR-splines [37], locally refined splines for which the refinement is specified by a sequence of mesh-boxes; U-splines [155], splines defined on unstructured meshes that can accommodate local variation in cell size, polynomial degree, and smoothness. Finally, Hierarchical B-splines (Hierarchical B-splines (HB-splines)) [53, 87] are multilevel B-spline extensions where the tensor-product structure is preserved at any level of the hierarchy. Their truncated formulation, i. e. Truncated Hierarchical B-splines (*Truncated Hierarchical B-splines (THB-splines)*) [55, 62], is based on HB-splines and relies on the definition of a certain truncation operator. THB-splines are a more flexible tool compared to HB-splines since they have smaller supports and reproduce some characteristic properties of tensor-product B-splines, such as non-negativity and partition of unity. They also guarantee the preservation of coefficients, i. e. they preserve the coefficients of functions expressed in terms of tensor-product B-splines of a certain hierarchical level [56]. For completeness, we recall that a generalization of THB-splines to allow anisotropic mesh refinement can be achieved in the framework of Patchwork B-splines [39]. In this Thesis, we choose to work

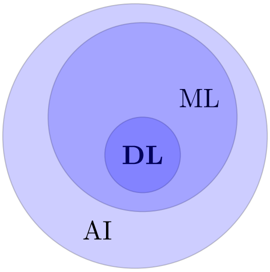

Figura 2. AI, ML and DL framework.

with THB-splines because of their local tensor-product structure, their key properties, and their ease of implementation.

0.2. Deep Learning

DL is a subset of Machine Learning (ML), the field of Artificial Intelligence (AI) consisting of algorithms, based on numerical and statistical methods, which directly *learn* from data, without being explicitly programmed. The core of ML algorithms consists in their ability to process row data and produce new proper representations, suitable for the considered problem, by automatically setting the value of some internal routine parameters. The novelty of DL, with respect to standard ML, consists in the use of multiple layers of abstraction to progressively extract higher-level features from the raw input data [94]. The relationship between the AI, ML and DL frameworks is summarised in Chapter 2. It is very well known that DL methods have improved the state-of-the-art in speech recognition, visual object recognition, object detection, and many other domains. In this Thesis, we also show how DL methods can improve the state-of-the-art for (spline) geometric modelling applications.

Neural Networks (NNs) are the computational framework of deep learning. These architectures are usually composed of several processing layers to learn data representations. Specifically, DL routines perform the data feature extraction by employing backpropagation algorithms to identify *intrinsic* relations in big data sets and determine how to adjust the internal parameters of the NN. Among others, some examples of NNs are the following. Multi Layer Perceptrons (Multi Layer Perceptrons (MLPs)) are the core of deep learning models [64], they consist of at least three fully connected layers, intertwined with non-linear (activation) functions. Convolutional Neural Networks (Convolutional Neural Networks (CNNs)) [91] are NNs that use convolutional operators in place of general matrix multiplication in at least one of their layers. They are a specialized kind of neural network for processing data that has a *known grid-like* topology. RESidual neural networks (RESidual neural networks (RESs)) [71] are an extension of CNNs, characterized by shortcut connections, introduced in [150] and also referred to as skip or residual connections, between their input and output, which allow to efficiently train NNs with many ($\gg 100$)

hidden layers. Skip connections are also used in the Long Short-Term Memory networks [73] and TRAnsformer encoder (TRA) models [36]. All the above-mentioned NNs are feed-forward networks, i. e. the information flows from the input, through the intermediate computations used to define the architecture, and finally to the output. There are no feed-back connections in which the outputs of the model are fed back into itself. When feed-forward neural networks are extended to include feed-back connections, they are called Recurrent Neural Networks (RNNs) [138].

In this Thesis, we initially exploit CNNs, which brought several breakthroughs in processing images, video, speech, and audio, and, in general, they thrive when any kind of data involving spatial-correlated patterns has to be processed. On the other hand, CNNs are not suitable to handle data with a non rectilinear-grid like topology, because of the lack of appropriate structures. Consequently, we also employ methods from geometric deep learning [19] to properly process unstructured or unorganized data configurations.

The translation of standard filter operators to graph operators relies on suitable aggregations of vertex and neighbour features. In particular, Graph Convolutional neural Networks (GCNs) [167] are a generalization of CNNs to non-Euclidean data, characterized by a *graph structure*, e. g., discrete manifolds, graphs, and general point clouds, 3D shapes, chemical molecules, and social/relational network. Hence, GCNs define the convolution operators on graph domains. Among the various graph convolutional operators available, see again [167] and the references therein, we exploit the fast localized spectral filters developed in [33] and the dynamic edge convolution operator proposed in [162].

0.3. Problem presentation

The core problem addressed in this Thesis is the *spline fitting* problem, namely the design of a parametric spline model which approximates a point cloud given as input. The aim is to develop highly-accurate robust and efficient methods for the representation of the discrete input via a continuous spline parametric model.

The data considered in this Thesis are real world data sets or synthetic data that mimic their behaviour. In particular, depending both on the application and the collection method, these data can result in *structured* point grids, meshes, or *scattered* point clouds. We propose new (data-driven) *parameterization* and *fitting* procedures that are able to handle different input data configurations and are suitable for the design of continuous free-from highly accurate and efficient spline models.

The considered data fitting problem can be mathematically described as follows. Given a (noisy) data set of the form,

$$\mathcal{P} = \{\boldsymbol{p}_i \in \mathbb{R}^N \mid i = 1, \dots, m\}, \tag{1}$$

where $N = 2$ for data points laying a plane and $N = 3$ for data points which belong to the three-dimensional space, find a geometric model $\boldsymbol{s} : \Omega \subseteq \mathbb{R}^D \to \mathbb{R}^N$, which approximates the data $\mathcal{P}$ within a certain tolerance $\epsilon \in \mathbb{R}_{>0}$, in the sense that, for each $i = 1, \dots, m$, $\operatorname{dist}(\boldsymbol{s}_i, \boldsymbol{p}_i) \leq \epsilon$, where $\boldsymbol{s}_i$

denotes a point on the geometric model associated with the data point $\boldsymbol{p}_i$, and dist $(\cdot,\cdot)$ is a certain distance metric.

By virtue of their properties, B-splines, and their multivariate hierarchical extension as THB-splines, are a desirable tool for building flexible geometric models. Therefore, the (TH)B-spline fitting problem can be stated in the following way. Given a point cloud as in (1) and an error tolerance $\epsilon > 0$, find a (TH)B-spline model $\boldsymbol{s} : \Omega \subseteq \mathbb{R}^D \to \mathbb{R}^N$, so that

$$\operatorname{dist}\left(\boldsymbol{s}(\boldsymbol{u}_i), \boldsymbol{p}_i\right) \leq \epsilon, \quad \text{with}\, \boldsymbol{u}_i \in \Omega, \quad \text{for each} \quad i = 1, \dots, m, \tag{2}$$

, where $\boldsymbol{u}_i$ denotes a point on the parametric domain Ω associated with the data point $\boldsymbol{p}_i$. Solving the problem in (2) implies the solution of two essential sub-problems: (a) the data parameterization and (b) the definition of the THB-spline approximant $\boldsymbol{s}$. In particular,

(a) Any parametric reconstruction scheme needs to define the data parameterization, namely to define the parametric set

$$\mathcal{U} \coloneqq \{\boldsymbol{u}_i \in \Omega \subset \mathbb{R}^D \,|\, i = 1, \dots, m\}, \tag{3}$$

which assign a parameter value $\boldsymbol{u}_i \in \Omega \subset \mathbb{R}^D$ to each point $\boldsymbol{p}_i \in \mathbb{R}^N$, for $i = 1, \dots, m$. Since the parameters encode intrinsic characteristics of the geometric model representation, estimating a good point cloud parameterization is a fundamental and delicate issue.

(b) The design of a spline model further depends on the characterization of the spline space V, defined by the multivariate degree $\boldsymbol{d} \in \mathbb{N}^D_{>0}$, the *knot* line *placement*, which defines a tessellation of Ω, and, consequently on the computation of the control points.

In this Thesis, we propose solutions to problem (2) for gridded and scattered data, by addressing both point (a) and (b) separately. As concerns (a), we propose *data-driven parameterization methods* based on (geometric) deep learning to address this problem both in the univariate and multivariate cases, considering either structured or unstructured point cloud configurations. As concerns (b), we introduce *novel adaptive fitting schemes* with moving parameterization and THB-spline, based on the optimization of different error metrics. In addition, we propose an adaptive approximation paradigm with THB-splines that addresses problems (a) and (b) *simultaneously*. The workflow of the proposed methods is summarized in Chapter 3.

0.4. Contributions and Thesis outline

We start in Chapter 1 by providing a selection of the preliminary notions for the contents of this Thesis. In particular, the first part of the Chapter is dedicated to (TH)B-spline constructions: B-splines, tensor-product B-splines, and THB-splines are introduced. The second part of the Chapter focuses on DL and NN architectures. More specifically, we concentrate on CNNs and GCNs, which are the main learning tools employed in the Thesis.

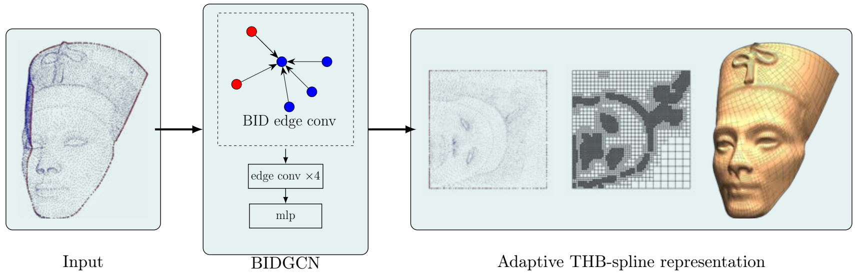

Figura 3. Data-driven point cloud parameterization and adaptive spline fitting. From left to right: an input scattered point cloud with interior (blue) and boundary (red) points is processed by the BIDGCN network proposed in [58] to predict the input point parameterization. The predicted parameters are subsequently employed in an adaptive surface reconstruction scheme to get the final hierarchical geometric model.

Subsequently, the main problem addressed in this Thesis – data fitting with THB-splines – is introduced in Chapter 2. Once an initial parameter and mesh configuration are chosen, any adaptive approximation procedure is characterized by four main steps which are successively repeated until a certain stopping criteria is satisfied, i. e.

$$1.\ \text{SOLVE} \rightarrow 2.\ \text{ESTIMATE} \rightarrow 3.\ \text{MARK} \rightarrow 4.\ \text{REFINE}. \tag{4}$$

SOLVE consists in the computation of the approximation on the current mesh; ESTIMATE computes the error estimation according to a suitable error indicator, whose values are then exploited to define a suitable marking strategy in MARK. Finally, REFINE defines the refinement strategies to suitably identify the adaptive mesh to be used in the next iteration of the adaptive loop. The algorithm performance is usually ruled by several input parameters, which establish the starting setting, the adaptive strategy, and the stopping criteria. Note that, once the initial configuration is chosen, the process is fully automatic, namely no user interaction is required.

We start Chapter 2 by reviewing established data fitting methods, i. e. interpolation and weighted least square schemes for an arbitrary approximation space [7]. Subsequently, we present a generalized formulation for reweighted least squares approximations, as convex combination of certain interpolants. We also provide a general strategy to iteratively update the weights according to the approximation error [57]. We then revisit the global Least Squares (LS) fitting scheme with THB-splines presented in [85], based on ordinary LS, by extending the automatic (error-driven) selection of the weights, within the adaptive procedure. Our reWeighted Least Squares (rWLS) scheme is able to tackle the presence of noisy or corrupted data, as well as data corresponding to fundamental geometric features to be represented in the final spline model, see again [57].

Subsequently, we concentrate on the approximation of scattered data sets with THB-splines, based on the two-stage adaptive Quasi-Interpolation

(QI) method scheme presented in [14] and further developed in [15]. In particular, we propose to modify the the first stage of the existing scheme from polynomial least squares approximations to least squares B-spline approximations, exploiting also a suitable fairness functional to handle data distributions with a locally varying density of points. We then also adjust the adaptive refinement strategy to properly handle the novelties introduced in the first stage. Chapter 2 concludes with a selection of numerical results of industrial complexity.

We address the parameterization problem for (TH)B-spline fitting schemes in Chapter 3. Standard parameterization methods both for *gridded* and *scattered* point clouds rely on a barycentric mapping induced from the local neighbourhoods of the points [128, 49, 51, 47, 50]. The complexity of the problem motivates the employment of DL as a viable option to predict an optimal choice of parameter values.

We develop data-driven parameterization methods for the parameterization problem. By introducing *PARamerization with Convolutional Neural Network (PARCNN)* we tackle the parameterization of planar/spatial point sequences and gridded point clouds, whereas *PARamerization with Convolutional Neural Network (PARCNN)* [32] and *Boundary Informed Dynamic Graph Convolutional neural Network (BIDGCN)* [58] address the parameterization of scattered data. In all these cases, we employ NNs characterized by convolutional operators, defined on the considered domain, for the parameterization learning problem and to assign parameter values at an *arbitrary* number of points for their subsequent use in (TH)B-spline (adaptive) approximation schemes. Differently from the other models, BIDGCN is able to process boundary conditions in addition to the standard vertex features of the discrete surface by a *new* graph convolution operator that contains two separate trainable message functions.

For each learning model, we provide details on the architecture developed, the training strategy, the model hyper-parameters and the data employed for training, test and validation phases. The performances of each method are illustrated by the related numerical examples.

All the proposed parameterization learning methods are agnostic to the dimension of the input, can generalize to point clouds of different dimensions, are robust to noise, outperform closed-form, heuristic and data-driven choices of the parameterization, and produce high-quality parameterizations for (TH)B-spline reconstruction schemes. In addition, BIDGCN overcomes the failing issues of standard meshless parameterization methods which happen in case of sparse data neighbourhoods. To the best of our knowledge, dimension-independent data-driven models to address the parameterization problem of point sequences and data sets in a general multivariate setting were not previously proposed.

In Chapter 4, we introduce novel adaptive approximation schemes with THB-spline and *moving parameterization.* State-of-the-art (TH)B-spline fitting methods address the parameterization problem, the construction of the spline space and the definition of the control net separately. In particular, given $\mathcal{P}$, a suitable parameterization $\mathcal{U}$ is computed and, on such a *fixed* parameterization, both the spline space, and the approximating spline model is defined. Instead, we follow up on the idea that, as the refinement proceeds in an adaptive setting, not only the geometric model but also the parameter values of each data point should be optimized. In this Thesis we propose two strategies to properly embed the parameterization within the adaptive model construction and to deal with a *moving parameterization*, rather than a fixed one.

The first strategy consists of enriching the adaptive approximating loop with iterative applications of the Parameter Correction (PC) routine [75]. In addition, we propose different adaptive fitting algorithm by exploiting diverse approximation schemes involved in the SOLVE step, in combination with PC routine [59]. In particular, we develop the adaptive *Alternating Point Distance Minimization (A-PDM)* fitting scheme, as the adaptive multivariate extension of the Point Distance Minimization (PDM) method originally presented in [75] for B-spline curve fitting. Moreover, by taking into account within the error term also the normal direction of the current geometric model at each iterative step, we develop the adaptive *Alternating Tangent Distance Minimization (A-TDM)* fitting scheme, based on the Tangent Distance Minimization (TDM) [8, 161, 106]. As suggested in [113, 9], we then combine the PDM and TDM error measures and take into account the discrete curvatures and the point-wise error distribution of the current model, to formulate the adaptive Alternating Hybrid Distance Minimization (A-HDM) multivariate fitting technique [59]. Finally, we exploit the introduction of PC also within an adaptive hierarchical QI methods, based on two-stage approximation scheme with local B-splines [61].

The second strategy to move the parameters consists of addressing the parameterization problem within the SOLVE step of the adaptive loop. The computation of the approximation on the current mesh in this case is performed by the solution a non-linear *Joint Point Distance Minimization (J-PDM)*, which consists in *simultaneously* computing the optimal parameter sites and control points, as proposed in [170] for the case of B-spline curve fitting. Therefore, with this method, we avoid solving a linear system of equations and performing PC at every adaptive iteration.

Our study reveals that using a moving parameterization, instead of a fixed one, can improve the fitting results while also reducing the number of degrees of freedom required to achieve a certain accuracy. It can also lead to earlier termination of the adaptive process, thus providing more compact models with less refinement depth and outperforming state-of-the-art hierarchical spline model reconstruction schemes.

Finally, Chapter 5 concludes the Thesis by summarizing the research results and providing new future research directions and perspectives.

The algorithms related to the learning parameterization models, see Chapter 3, have been implemented in `Python` with the open-source PyTorch and PyTorch Geometric libraries [126, 83, 46]. The novel graph convolution operator introduced for the BIDGCN model, together with the network architecture can be found at the following repository: `https://github.com/felixfeliz/BIDGCN`. The code for PARCNN and PARameterization with Graph Convolutional neural Network (PARGCN) will be made available on request.

The algorithms related to the adaptive fitting schemes with THB-splines, see Chapter 2 and Chapter 4, have been implemented in `C++` with the open-source G+Smo library [78, 112]. The developed code for the proposed algorithms has been integrated and will be available in the next releases.

The original results presented in this Thesis are based on [57, 17] (Chapter 2), [32, 60, 58] (Chapter 3), and [61, 59] (Chapter 4).