
# IGS2023 International Geometry Summit

## Posters' Collection

Genoa, Italy
July 3-7, 2023

**Poster Co-Chairs**
Daniela Cabiddu (*CNR-IMATI, Italy*)
Nicolas Mellado (*CNRS, France*)
Claudio Mancinelli (*University of Genoa, Italy*)
Nelly Villamizar (*Swansea University, Galles*)
Ligang Liu (*University of Science and Technology of China, China*)

Promoted by
Organized by
With the technical support of

# Contents

**A library for robust linear barycentric rational interpolation**
*Chiara Fuda, Kai Hormann* . . . . . 4

**NURBS-based workflow and G-code for precise Additive Manufacturing**
*Javier Sánchez-Reyes, Jesús M. Chacón, Javier Vallejo, Pedro J. Núñez* . . . . 6

**Point Cloud Denoising Using a Generalized Error Metric**
*Qun-Ce Xu, Yong-Liang Yang, Bailin Deng* . . . . . 8

**An Insertability Constraint for Shape Optimization**
*Eric Garner, Jun Wu, Amir Zadpoor* . . . . . 10

**Isotropic Point Cloud Meshing using unit Spheres (IPCMS)**
*Henriette Lipschütz, Ulrich Reitebuch, Konrad Polthier, Martin Skrodzki* . . . . 12

**Geometric deep learning for shape optimization in computational architecture**
*Andrea Favilli, Francesco Laccone, Paolo Cignoni, Luigi Malomo, Daniela Giorgi* 14

**Spatial $n$-Arc Spline Approximation**
*Stephan Alt, Hartmut Prautzsch* . . . . . 16

**Scenic Routes With Weighted Points In 2D**
*Vijayraj Shanmugaraj, Lini Thomas, Kamalakar Karlapalem* . . . . . 18

**Jittor & JNeRF: a novel deep learning framework with meta-operators and unified graph execution, and its NeRF model zoo**
*Guo-Wei Yang, Shi-Min Hu* . . . . . 20

**Multi-patch parameterization method for isogeometric analysis using singular structure of cross-field**
*Ye Ji, Yi Zhang, Chun-Gang Zhu* . . . . . 22

**The Volumetric Integer-Grid Map based Hexahedral Meshing Pipeline**
*Hendrik Brückler, Heng Liu, David Bommes, Marcel Campen* . . . . . 24

**Surfaces from magnetic spheres**
*Henriette Lipschütz, Konrad Polthier* . . . . . 26

**Semantically Enriched 3D Geometric Modeling for Urban Digital Twins**
*Andreas Scalas, Daniela Cabiddu, Michela Mortara, Simone Pittaluga, Michela Spagnuolo* . . . . . 28

**Scenic Routes over Points in 2D Space**
*Loay Rashid, Lini Thomas, Kamal Karlapalem* . . . . . 30

**Soft Robots by Geometric Computing**
*Christopher-Denny Matte, Tsz-Ho Kwok* . . . . . 32

**Numerical Integration for Subdivision IGA**
*Alexander Dietz* . . . . . 34

**An adapted interpolation method for generating smooth surfaces around extraordinary vertices of quadrilateral meshes**
*Byeongseon Jeong, Kyungmi Kim, Hyoseon Yang, Jungho Yoon* . . . . . 36

# Preface

In July 2023, the IGS2023 - International Geometry Summit features 5 major conferences in Computer Graphics and applications

- Shape Modeling International (SMI)
- Symposium on Physical and Solid Modeling (SPM)
- SIAM Conference on Computational Geometric Design (GD)
- EG Symposium on Geometry Processing (SGP)
- Geometric Modeling and Processing (GMP)

IGS2023 posters describe preliminary results, ongoing projects, recent published work, or software and systems (e.g., open source libraries, experimental setup) in several areas related to

- solid and physical modelling
- computational geometric design
- shape modelling and analysis
- geometric modelling and processing

After a review phase, 19 poster presentations were accepted to the program, and they were presented during a joint poster session of IGS2023 that offered interactive discussion between presenters and attendees. attendees.

Special thanks are given to the IGS2023 Chairs, Michela Spagnuolo. Konrad Polthier and Wenping Wang, for their support to the IGS2023 Poster Session.

**Poster Co-Chairs**
Daniela Cabiddu (*CNR-IMATI, Italy*)
Nicolas Mellado (*CNRS, France*)
Claudio Mancinelli (*University of Genoa, Italy*)
Nelly Villamizar (*Swansea University, Galles*)
Ligang Liu (*University of Science and Technology of China, China*)

# A library for robust linear barycentric rational interpolation

Chiara Fuda[a], Kai Hormann[a]

[a] *Università della Svizzera italiana, Via la Santa, Lugano, 6962, Switzerland*

*Keywords:* Interpolation, Stability, Floating-point arithmetic, `C++` library

**Abstract**

Given a set of $n+1$ interpolation *nodes* $X_n = (x_0, x_1, \ldots, x_n)$ with $x_0 < x_1 < \cdots < x_n$ and some associated *data* $Y_n = (y_0, y_1, \ldots, y_n)$, the *barycentric rational interpolant* $r\colon \mathbb{R} \to \mathbb{R}$ that interpolates $y_i$ at $x_i$ can be expressed in *second barycentric form* as

$$r(x) = \frac{\sum_{i=0}^{n} \frac{w_i}{x-x_i} y_i}{\sum_{i=0}^{n} \frac{w_i}{x-x_i}} \tag{1}$$

for some non-zero *weights* $W_n = (w_0, w_1, \ldots, w_n)$ [6]. We focus on *linear* barycentric rational interpolation, meaning that the denominator does not depend on the data $Y_n$, and we use weights $w_i$ that guarantee the absence of poles in the interpolant. Floater and Hormann [3] propose a linear barycentric interpolant by considering a parameter $d \in \{0, 1, \ldots, n\}$ and the *barycentric weights*

$$w_i = \sum_{k=\max(i-d,0)}^{\min(i,n-d)} (-1)^k \prod_{j=k,\, j\neq i}^{k+d} \frac{1}{x_i - x_j}, \qquad i = 0, 1, \ldots, n, \tag{2}$$

which covers both the already known *Berrut*'s case for $d = 0$ [1] and the *polynomial* one for $d = n$ [2]. The barycentric formula (1) is widely used to evaluate the interpolant $r$, as it can be implemented with an efficient $O(n)$ algorithm and the $w_i$ can be rescaled by a common factor to prevent overflow and underflow errors [2]. However, there exists another mathematically equivalent formula to evaluate $r$, namely the *first barycentric form*

$$r(x) = \frac{\sum_{i=0}^{n} \frac{w_i}{x-x_i} f_i}{\sum_{i=0}^{n-d} \lambda_i(x)}, \tag{3}$$

where

$$\lambda_i(x) = \frac{(-1)^i}{(x-x_i)\cdots(x-x_{i+d})}, \qquad i = 0, 1, \ldots, n-d. \tag{4}$$

While this formula is slightly inferior in terms of efficiency, because it requires $O(nd)$ operations to be computed straightforwardly, there exists a more efficient way of computing its denominator in $O(n)$ operations [5].

In this poster, we present the BRI library, which features a new `C++` class template that contains all variables and functions related to linear barycentric rational interpolation. Since there exist several algorithms to evaluate a barycentric rational interpolant based on formulas (1) and (2), the library is designed to autonomously select the best method to be used on a case-by-case basis, as it takes into account the latest results regarding the efficiency and numerical stability of barycentric rational interpolation [5]. Moreover,

*Email addresses:* `chiara.fuda@usi.ch` (Chiara Fuda), `kai.hormann@usi.ch` (Kai Hormann)

we describe a new technique that makes the code robust and less prone to overflow and underflow errors. In addition to the standard `C++` data types, the BRI template variables can also be defined with arbitrary precision, because the BRI library is compatible with the Multiple Precision Floating-Point Reliable (MPFR) library [4].

*Example*

Let us consider $n = 9$, $n + 1$ Chebyshev interpolation nodes in $[a, b] = [0, 10^{-12}]$, i.e.

$$x_i = \frac{a+b}{2} - \frac{b-a}{2} \cos \frac{(2i+1)\pi}{2n+2}, \qquad i = 0, 1, \ldots, n$$

and $d = 2$. We then set $y_i = f(x_i)$ for $i = 0, 1, \ldots, n$, where the test function $f$ is defined as

$$f(x) = \tfrac{3}{4} e^{-\frac{(9x-2)^2}{4}} + \tfrac{3}{4} e^{-\frac{(9x+1)^2}{49}} + \tfrac{1}{2} e^{-\frac{(9x-7)^2}{4}} + \tfrac{1}{5} e^{-(9x-4)^2},$$

and we evaluate the barycentric interpolant $r$ at the midpoint of the interval, $x = 10^{-12}/2$. If we compute $r(x)$ in single precision with the standard implementation of (1) or (2), then we get $r(x) = -\texttt{nan}$ due to the overflow of all terms $w_i y_i/(x - x_i)$, $i = 1, \ldots, n - 1$. For example,

$$\frac{w_1}{x - x_1} y_1 = -6.2335 \times 10^{38},$$

which is not representable as single precision floating-point number, because the smallest such number is $\approx -3.4028 \times 10^{38}$. The BRI library is able to overcome this problem thanks to the new technique that we explain in the poster, thus obtaining the correct result $r(x) = 1.01076$ with both barycentric formulas.

## References


[1] Berrut, J.P., 1988. Rational functions for guaranteed and experimentally well-conditioned global interpolation 15, 1–16. doi:10.1016/0898-1221(88)90067-3.

[2] Berrut, J.P., Trefethen, L.N., 2004. Barycentric Lagrange interpolation 46, 501–517. doi:10.1137/S0036144502417715.

[3] Floater, M.S., Hormann, K., 2007. Barycentric rational interpolation with no poles and high rates of approximation 107, 315–331. doi:10.1007/s00211-007-0093-y.

[4] Fousse, L., Hanrot, G., Lefèvre, V., Pélissier, P., Zimmermann, P., 2007. MPFR: A multiple-precision binary floating-point library with correct rounding. ACM Trans. Math. Software 33. doi:10.1145/1236463.1236468.

[5] Fuda, C., Campagna, R., Hormann, K., 2022. On the numerical stability of linear barycentric rational interpolation 152, 761–786. doi:10.1007/s00211-022-01316-w.

[6] Schneider, C., Werner, W., 1986. Some new aspects of rational interpolation 47, 285–299. doi:10.2307/2008095.

# NURBS-based workflow and G-code for precise Additive Manufacturing

Javier Sánchez-Reyes[a], Jesús M. Chacón[a], Javier Vallejo[a], Pedro J. Núñez[b]

[a] *IMACI, ETS Ingeniería Industrial de Ciudad Real, UCLM Universidad de Castilla-La Mancha (Spain)*
[b] *ETS Ingeniería Industrial de Ciudad Real, UCLM Universidad de Castilla-La Mancha (Spain)*

*Keywords:* Additive manufacturing, NURBS, cubic Bézier curve, G-code, G2/3 command, G5 command

**Abstract**

We put forward [1] a streamlined AM workflow with a seamless transfer from the initial CAD description to the final G-code. Adhering to the NURBS standard at all steps avoids multiple representations, polygonal approximations, and associated errors.

The traditional AM workflow consists of an initial polygonization of the object (e.g., via STL), slicing this approximation, offsetting the polygonal sections, and finally generating G-code made up of polyline trajectories (G1 commands). This workflow does not meet the requirements for truly functional parts regarding quality and precision, especially in large-scale 3D printing, hence wasting the possibilities of existing AM hardware.

Our proposal (Fig. 1) bypasses the polygonal approximation and then proceeds as follows:

(1) Direct slicing of the CAD model in the NURBS environment provided by a NURBS-based CAD system.
(2) Path planning, including offset trajectories, in this NURBS environment.
(3) Accurate G-code generation of NURBS: circular arcs (G2/3), Bézier cubics (G5), and polylines (G1).

Slicing (1) and offsetting, the most complex geometry operation in path planning (2), are already available in any CAD system in a reliable way. Therefore, there is no need to develop ad-hoc procedures. In particular, we employ a NURBS-based commercial CAD system (Rhino3D along with its programming environment Grasshopper) for direct slicing of the model, offset generation, and trimming.

Our main contribution is sticking to the NURBS standard at the last step (3), namely G-code generation, a possibility overlooked in both the literature and commercial applications. To this aim, we exploit the possibilities of exiting firmware controlling 3D printers, such as Marlin, incorporating G2/3 (circular arcs) and G5 (cubic Bézier curves) commands. Since trajectories resulting from offsetting in Rhino3D usually restrict to circles and polynomial (cubic or quadratic) splines, the exact conversion into G2/3 and G5 code is readily performed via standard NURBS geometry processing, such as knot-insertion and degree-elevation.

*Results*

The example of Fig. 1, a rear-view mirror housing designed with SolidWorks and printed with CreatBot PEEK-300, compares the traditional method (via STL file) and our direct NURBS-based approach. It depicts the dimensional deviations $d$ of the outer surface of the printed STL/NURBS model from the theoretical CAD model. The results confirm a considerable improvement in quality thanks to our proposal.

*Acknowledments*

Research supported by grant PID2019-104586RB-I00, funded by MCIN/AEI/10.13039/501100011033; and SBPLY/19/180501/000247 by *Consejería de Educación Cultura y Deportes* (*Junta de Comunidades de Castilla-La Mancha*) and grant 2022-GRIN-34077 by *Universidad de Castilla-La Mancha*, co-financed by the ERDF (European Regional Development Fund).

*Email addresses:* Javier.SanchezReyes@uclm.es (Javier Sánchez-Reyes), JesusMiguel.Chacon@uclm.es (Jesús M. Chacón), Javier.Vallejo@uclm.es (Javier Vallejo), Pedro.Nunez@uclm.es (Pedro J. Núñez)

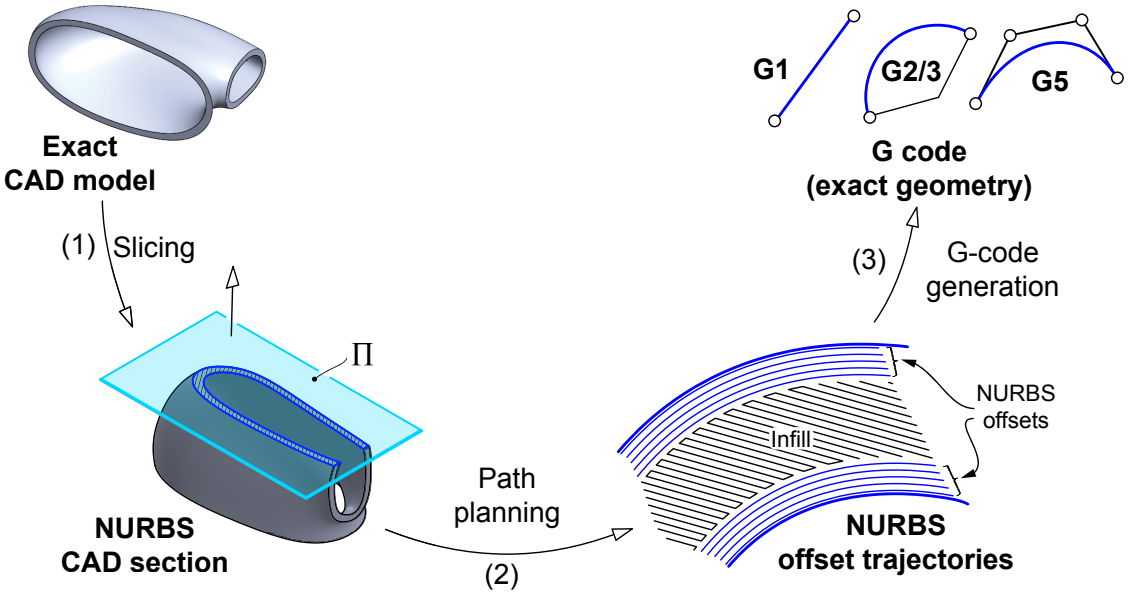



Figure 1: Our approach for generating accurate G-code for AM

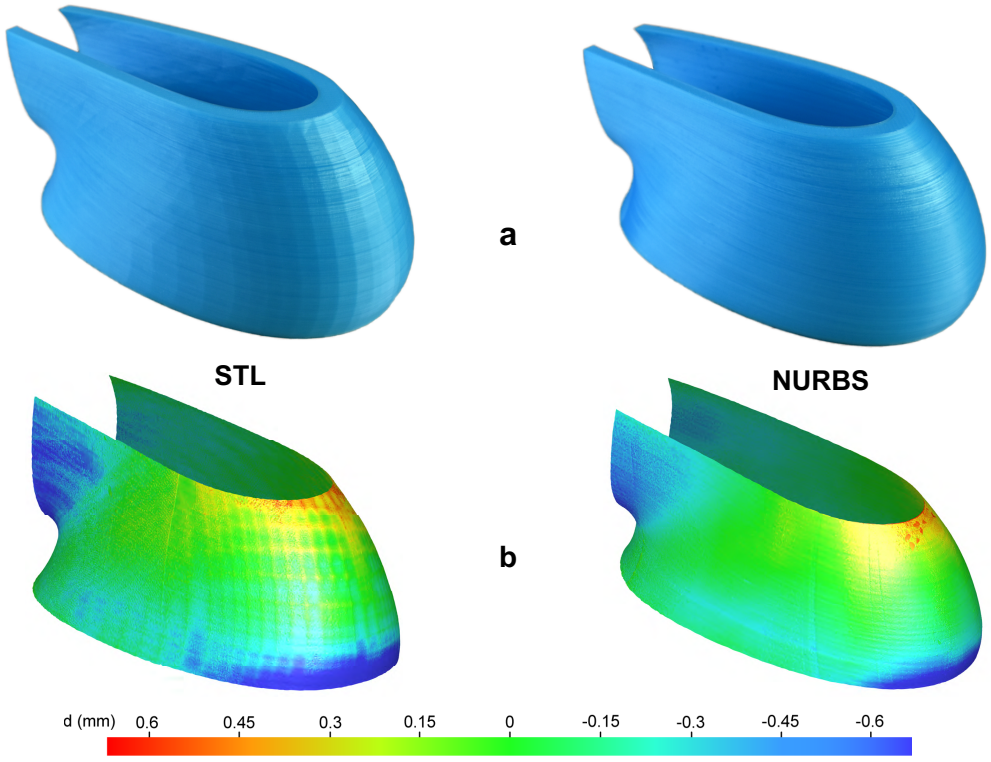



Figure 2: Example comparing traditional AM via STL and our NURBS-based workflow: a) Actual pictures. b) Deviations $d$ from the exact CAD model (outer surface)

## References

[1] Chacón, J.M., Sánchez-Reyes, J., Vallejo, J., Núñez, P.J., 2022. G-code generation in a NURBS workflow for precise Additive Manufacturing. Rapid Prototyp. J. 28(11), 65–76. https://doi.org/10.1108/RPJ-09-2021-0254.

# Point Cloud Denoising Using a Generalized Error Metric

Qun-Ce Xu[a], Yong-Liang Yang[b], Bailin Deng[c]

[a] *Department of Computer Science and Technology, Tsinghua University*
[b] *Department of Computer Science, University of Bath*
[c] *School of Computer Science and Informatics, Cardiff University*

*Keywords:* Geometry Processing, Optimization, Point Cloud Denoising

**Abstract**

We propose a method for feature-preserving point cloud denoising, which jointly optimizes the point positions and normals. We adopt a generalized robust error metric [1] for normal smoothness; by varying a parameter of the error metric, we gradually increase its non-convexity to help retain sharp features. We also develop a numerical solver that guarantees a monotonic decrease of the target function and its convergence.

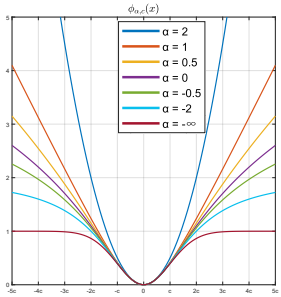
**Method**

Given an input noisy point cloud containing $n$ points, with positions $\{\mathbf{p}_i^0 \in \mathbb{R}^3\}$ and outward normals $\{\mathbf{n}_i^0 \in \mathbb{R}^3\}$ $(i = 1, \ldots, n)$, we would like to compute the positions $\{\mathbf{p}_i\}$ and normals $\{\mathbf{n}_i\}$ for the denoised point cloud. To this end, we optimize $\{\mathbf{p}_i\}$ and $\{\mathbf{n}_i\}$ simultaneously by solving the following problem:

$$\min_{\mathbf{P},\mathbf{N}} \; E_{\text{fid}} + w_{\text{u}} E_{\text{unit}} + w_{\text{d}} E_{\text{disp}} + w_{\text{r}} E_{\text{reg}}. \tag{1}$$

Here $\mathbf{P}, \mathbf{N} \in \mathbb{R}^{3n}$ are new point positions and normals, respectively. $E_{\text{fid}}$ is a fidelity term penalizing changes in positions and normals: $E_{\text{fid}} = w_{\text{p}} \left\|\mathbf{P} - \mathbf{P}^0\right\|^2 + w_{\text{n}} \left\|\mathbf{N} - \mathbf{N}^0\right\|^2$, where $\mathbf{P}^0, \mathbf{N}^0 \in \mathbb{R}^{3n}$ are the initial positions and normals, and $w_{\text{p}}, w_{\text{n}}$ are user-specified weights. $E_{\text{unit}} = \sum_{i=1}^{n} \left\|\mathbf{n}_i - \mathbf{n}_i / \|\mathbf{n}_i\|\right\|^2$ enforces unit lengths for the new normals. $E_{\text{disp}} = \sum_{i=1}^{n} \|\mathbf{n}_i \times (\mathbf{p}_i - \mathbf{p}_i^0)\|^2$ enforces parallel relation between the position change $\mathbf{p}_i - \mathbf{p}_i^0$ and the normal $\mathbf{n}_i$ for each point. $E_{\text{reg}} = \sum_{\substack{i<j \\ (i,j)\in\mathcal{N}}} \phi(h_{ij})$ is a robust regularization term, where $\mathcal{N}$ denotes the index set of neighboring points. Here $h_{ij}$ enforces smoothness between neighboring normals and their consistency with point positions: $h_{ij} = \sqrt{(\mathbf{n}_i \cdot (\mathbf{p}_i - \mathbf{p}_j)/L_{ij})^2 + (\mathbf{n}_j \cdot (\mathbf{p}_i - \mathbf{p}_j)/L_{ij})^2 + \gamma \|\mathbf{n}_i - \mathbf{n}_j\|^2} \Big/ L_{ij}$, where $L_{ij}$ is the initial distance between $\mathbf{p}_i$ and $\mathbf{p}_j$ to account for uneven sampling, and $\gamma$ is a user-specified weight. $\phi$ is a robust error metric proposed in [1] (see inset figure for its graph):

$$\phi_{\alpha,c}(x) = \begin{cases} (x/c)^2/2 & \text{if } \alpha = 2, \\ \log\left((x/c)^2/2 + 1\right) & \text{if } \alpha = 0, \\ 1 - \exp\left(-(x/c)^2/2\right) & \text{if } \alpha = -\infty, \\ \frac{|\alpha-2|}{\alpha}\left(((x/c)^2/|\alpha-2| + 1)^{\alpha/2} - 1\right) & \text{otherwise.} \end{cases} \tag{2}$$




As $\alpha$ decreases to $-\infty$, the function $\phi_{\alpha,c}(x)$ becomes less sensitive to large magnitudes of $x$. In our formulation, this induces robustness against large values of $h_{ij}$ that indicate sharp features. Thus our formulation removes noises in smooth areas by reducing $h_{ij}$, while retaining sharp features by allowing for large values of $h_{ij}$. We solve the optimization problem with $\alpha$ gradually decreased from 2 to $-\infty$.

*Email addresses:* quncexu@mail.tsinghua.edu.cn (Qun-Ce Xu), y.yang@cs.bath.ac.uk (Yong-Liang Yang), DengB3@cardiff.ac.uk (Bailin Deng)

*Numerical Solver.* To solve the optimization problem (1), we adopt an alternating minimization approach: we first fix **P** and optimize **N**, and then fix **N** and optimize **P**, and repeat this process until convergence. For each subproblem, the terms $E_{\text{unit}}$ and $E_{\text{reg}}$ are still non-convex. To further simplify the problem, we replace each of these terms with a convex quadratic surrogate function that has bounds the term from above and has the same function value as the respective term for the current variable values. This reduces both the **N**- and **P**-subproblems to minimizing a convex quadratic target function, which can be solved via a sparse symmetric positive definite linear system. Our approach is an instance of the majorization-minimization (MM) algorithm [3], which repeatedly constructs and minimize a surrogate function for the true target function. It can be shown that the MM algorithm guarantees monzonitic decrease of the target function until convergence [3].

## Results

Figure 1 uses two input point cloud models to compare our denoising results with two existing methods, EAR [2] and RIMLS [4]. It is evident that our method can remove noises while preserving features and produce better results than the other two methods. Figure 2 compares our method with two alternative parameter settings for the error metric in Eq. (2), which fix the value of $\alpha$ to 2 or $-\infty$ instead of gradually decreasing it. The color coding shows the distance to the ground-truth surface. Our result point cloud is notably closer to the ground truth than those with fixed $\alpha$ values, which demonstrates the effectiveness of approach that gradually increases the non-convexity of the error metric.

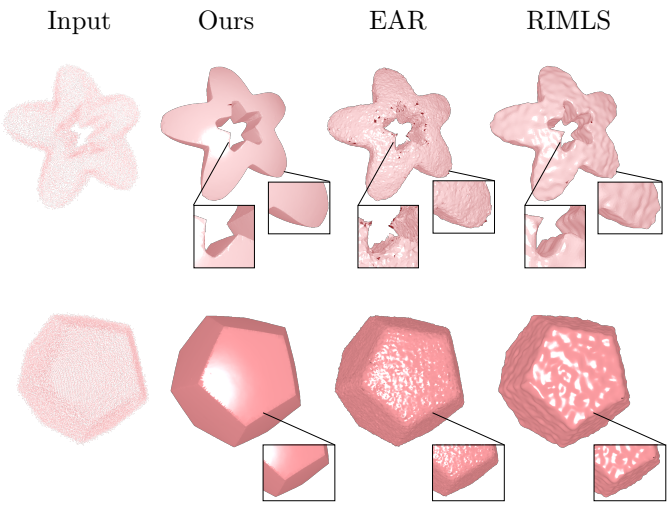



Figure 1: Comparison on sharp feature preservation. The model 'Star' at the top has 24K points with 2.0% Gaussian noise. The 'Polyhedron' in the bottom has 43K points with 1.5% Gaussian noise.

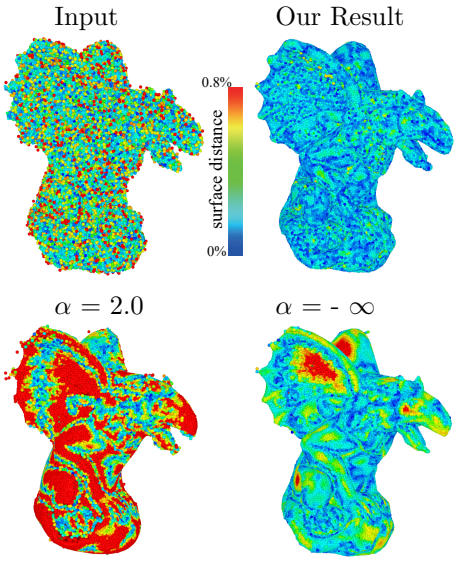



Figure 2: Different settings of $\alpha$ on the Gargoyle model with 20K points and 1.0% Gaussian noise. Our method with a decreasing value of $\alpha$ (top right) performs significantly better than fixed $\alpha$ values (bottom).

## References

[1] Barron, J.T., 2019. A general and adaptive robust loss function, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4331–4339.
[2] Huang, H., Wu, S., Gong, M., Cohen-Or, D., Ascher, U., Zhang, H.R., 2013. Edge-aware point set resampling. ACM Trans. Graph. 32.
[3] Lange, K., 2016. MM optimization algorithms. SIAM.
[4] Öztireli, A.C., Guennebaud, G., Gross, M., 2009. Feature preserving point set surfaces based on non-linear kernel regression. Computer Graphics Forum 28, 493–501.

# An Insertability Constraint for Shape Optimization

Eric Garner[a], Jun Wu[b], Amir Zadpoor[a]

[a] *Department of Biomechanical Engineering, Delft University of Technology, Mekelweg 2, Delft, 2628 CD, the Netherlands*
[b] *Department of Sustainable Design Engineering, Delft University of Technology, Landbergstraat 15, Delft, 2628 CE, the Netherlands*

*Keywords:* patient-specific implants, insertability analysis, computational design, shape optimization, structural optimization, path planning, robot-assisted surgery

**Abstract**

Patient-specific implants, designed according to patients' unique anatomy, offer a host of benefits over their generic counterparts. Nonetheless, the design and optimization of these components present several technical challenges, among them being the need to ensure their insertability into the host bone tissue. This presents a significant challenge due to the tight-fitting nature of the bone-implant interface.

Insertability analysis, which is the problem of assessing whether a rigid body can be inserted into a tight-fitting cavity, is ubiquitous in a wide range of engineering applications. In automated tooling, for example, it is used to plan an interference-free path for inserting robotically-guided parts into mating fixtures [1, 3]. In molding, it is used to ensure that the part can be extracted from the core and cavity without excessive force [2]. And in engineering design, it is used to design tightly-fitting parts and establish tolerances for assemblability [4]. In particular, with respect to the design of orthopaedic implants, where a tight fit with the host bone is usually required, insertability analysis plays a key role in avoiding obstructing geometry and stuck configurations.

This work presents a novel insertability metric designed to efficiently assess whether a rigid body can be inserted into a tight-fitting cavity, without interference. In contrast to existing solutions, the metric is fully differentiable and can be incorporated as a design constraint into shape optimization routines. By exploiting the tight-fitting condition, the problem of planning an interference-free insertion path is reformulated as the search for a single interference-free movement, starting from the inserted configuration. We prove that if there exists any outward movement for which no interference is indicated, then the body can be fully extracted from or, equivalently, inserted into the cavity. This formulation is extremely efficient and highly robust with respect to the complexity of the geometry.

We demonstrate the effectiveness and efficiency of our method by applying it to the optimization of two-dimensional (2D) and three-dimensional (3D) designs for insertability, subject to various design requirements. We then incorporate the proposed metric into the optimization of an acetabular cup used in total hip replacement (THR) surgery where geometric and structural requirements are considered.

*Results*

The proposed insertability metric has proven capable of optimizing a range of 2D and 3D systems for insertability. In Fig. 1, several uninsertable body-cavity systems (top row) have been optimized for insertability, while subject to various geometric requirements. In the second row, the body and cavity are both altered. In the third row, the cavity geometry remains unchanged and the body maximally fills the cavity (space-filling). In the fourth row, the body is unchanged and the cavity is grown to ensure insertability (space-making).

*Email addresses:* E.Garner@tudelft.nl (Eric Garner), J.Wu-1@tudelft.nl (Jun Wu), A.A.Zadpoor@tudelft.nl (Amir Zadpoor)

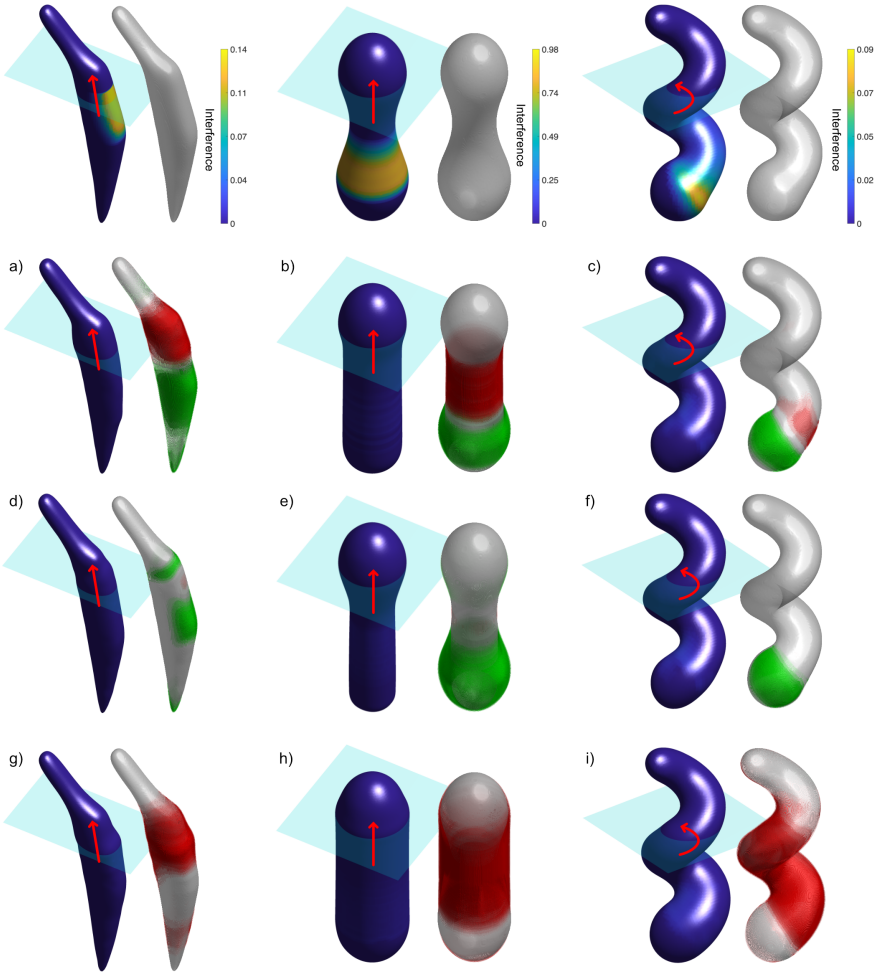



Figure 1: Various 3D body-cavity systems modified for insertability. The original designs (top row) inhibit insertability in the highlighted areas (left). The modified designs (bottom rows) are presented along with the local shape change from the original design (right). Red and green refer to added and removed material, respectively. The second row uses a standard minimum shape change objective. The third and fourth rows use the space-filling and space-making variations of the shape-match functions.

## References

[1] Canny, J., 1988. The complexity of robot motion planning. MIT press.
[2] Carley, J.F., 1993. Whittington's dictionary of plastics. CRC Press.
[3] Oliver, J.H., Huang, H.T., 1994. Automated path planning for integrated assembly design. Computer-aided design 26, 658–666.
[4] Shen, Z., Ameta, G., Shah, J.J., Davidson, J.K., 2005. A Comparative Study Of Tolerance Analysis Methods. Journal of Computing and Information Science in Engineering 5, 247–256.

# Isotropic Point Cloud Meshing using unit Spheres (IPCMS)

Henriette Lipschütz[a], Ulrich Reitebuch[a], Konrad Polthier[a], Martin Skrodzki[b]

[a] *Department of Mathematics and Computer Science, FU Berlin*
[b] *Computer Graphics and Visualization, TU Delft*

*Keywords:* point cloud meshing, surface reconstruction, manifold property, uniform triangle mesh

**Abstract**

We consider point clouds created by scanning real world models, equipped with provided or generated normals, and aim for a reconstruction of the underlying geometry. We strive for high triangle quality, as measured by an edge length close to uniformity across the mesh. The algorithm presented here is based on a sphere-packing algorithm [7]. It is extended such that, if the input and the normals satisfy certain quality assumptions, the output of the algorithm is manifold. Our contributions are (full details to be found in [6]):

- presentation of a meshing algorithm that places touching spheres of uniform radius on the input,
- which creates edge lengths close to uniformity and of a guaranteed minimum length,
- as well as manifold output, provided a suitable input geometry and good enough normals.

*Theory*

Let $\mathcal{M}$ be an orientable, compact $\mathcal{C}^2$-manifold embedded into $\mathbb{R}^3$, which is assumed to be closed and of finite reach $\rho := \inf\{\|a-m\| \mid a \in \mathcal{A}_\mathcal{M} \wedge m \in \mathcal{M}\} \in \mathbb{R}_{>0}$, where $\mathcal{A}_\mathcal{M}$ is the *medial axis* of $\mathcal{M}$ consisting of the points $q \in \mathbb{R}^3$ fulfilling $\min_{p\in\mathcal{M}} |q-p| = |q-\hat{p}| = |q-\tilde{p}|$ for $\hat{p} \neq \tilde{p} \in \mathcal{M}$. Then, we can show (see [6]):

**Lemma 1.** *Let $p \in \mathcal{M}$ be a point and let $N_p$ denote its normal. Then, for $r < \rho$, the image of $B_r(p) \cap \mathcal{M}$ under the projection $\pi$ in direction of $N_p$ to the tangent plane $T_p\mathcal{M}$ is a convex set.*

Assume there is a graph $\mathcal{G}$ embedded on $\mathcal{M}$ and vertices in $\mathcal{G}$ connected by an edge have Euclidean distance $d$. The connected components remaining after removing $\mathcal{G}$ from $\mathcal{M}$ are called *regions*, denoted by $\mathcal{R}$. The set of vertices and edges incident to a region $R \in \mathcal{R}$ is called its *border*, denoted by $\partial R$. Fix one such region $R \in \mathcal{R}$. Lemma 1 implies a choice of points $q_1, \dots, q_k \in \partial R$ is mapped to points $\pi(q_1), \dots, \pi(q_k)$ in the tangent plane $T_p\mathcal{M}$ in cyclic order, for $p \in R$ arbitrarily chosen. Hence, the regions can be extracted correctly (w.r.t. their topology) from the cyclic order in the local projection. Given that the reach criterion is satisfied and given a suitable normal field, we can thus reconstruct a manifold from the input surface.

*Methodology*

The user specifies the input point cloud $\mathcal{P}$, a *target edge length* $d$, and a *splat size* $s$. Here, $d$ defines non-overlapping virtual spheres of radius $d/2$ around the vertices of the resulting mesh, i.e., all mesh edges have a minimum length of $d$. Following Lemma 1, the user has to choose the parameter $d$ with respect to the reach of $\mathcal{P}$. For a point $p \in \mathcal{P}$ and its normal $N_p$, the parameter $s$ defines the radius of a disk $S_p$ with normal $N_p$, centered at $p$, called a *splat*. The algorithm is then given by the following steps:

0. Input: a point cloud $\mathcal{P}$, its normals $\mathcal{N}$, and the user-chosen parameters $d$ and $s$.

*Email addresses:* henriette.lipschuetz@fu-berlin.de (Henriette Lipschütz), ulrich.reteibuch@fu-berlin.de (Ulrich Reitebuch), konrad.polthier@fu-berlin.de (Konrad Polthier), mail@ms-math-computer.science (Martin Skrodzki)

Table 1: Bottle Shampoo (604,903 input points)

| Algorithm | $\lvert\mathcal{T}\rvert$ | $Q_{\text{avg}}$ | $Q_{\text{RMS}}$ |
|---|---|---|---|
| Adv. Front | 1.209,546 | 0.8247 | 16.0 |
| Poisson MG | 150,770 | 0.7204 | 33.7 |
| RIMLS | 1,907,781 | 0.7055 | 35.1 |
| Scale Space | 1,209,093 | 0.8248 | 16.0 |
| Voronoi | 1,209,792 | 0.8241 | 16.1 |
| Ours | 840,453 | **0.9577** | **4.5** |

Table 2: Bowl Chinese (606,320 input points)

| Algorithm | $\lvert\mathcal{T}\rvert$ | $Q_{\text{avg}}$ | $Q_{\text{RMS}}$ |
|---|---|---|---|
| Adv. Front | 1,212,636 | 0.8045 | 18.6 |
| Poisson MG | 503,458 | 0.7062 | 37.1 |
| RIMLS | 6,458,589 | 0.6877 | 39.6 |
| Scale Space | 1,093,339 | 0.8054 | 18.7 |
| Voronoi | 1,212,636 | 0.8042 | 18.7 |
| Ours | 2,137,650 | **0.9485** | **6.2** |

1. The user initializes the algorithm by two starting vertices lying sufficiently close to each other such that a new vertex with distance $d$ to both can be placed on the splats.
2. Iteratively, place a new vertex on a splat such that it has distance $d$ to two currently existing ones, called *parent vertices*, and distance $\geq d$ to all other existing vertices. Stop when no additional sphere of radius $d/2$ can be added without intersecting any $d/2$-spheres around existing mesh vertices. After the disk-growing finished, all edges of the mesh have length $d$ by construction: Each time we add a new vertex $v$, we also insert two edges of length $d$ into the graph $\mathcal{G}$, connecting $v$ to its parent vertices. The edges will build a set of regions $\mathcal{R}$ of an (on average) short border length, representing the surface.
3. Finally, to obtain a triangulation, all regions $R \in \mathcal{R}$ are triangulated.
4. Output: a triangulation $\mathcal{T}$ of $\mathcal{M}$, which is manifold according to the *Theory*.

In order to efficiently implement these operations, we add two components to the underlying sphere-packing algorithm [7]. First, we create a uniform grid data structure to efficiently determine the relationship of introduced vertices and splats on the input points. This data structure also provides estimated normal information and thereby ensures topological correctness of the resulting mesh. Second, we introduce a heuristic to speed up the evaluation process of topological decisions to be made when introducing a new vertex. While it provides topologically correct results in all cases, it reduces the query time from an estimate of $\mathcal{O}(\sqrt{n})$ (where $n$ is the number of vertices currently in the mesh) to a window of constantly many vertices.

*Results*

For our experiments, we turn to 20 scanned objects provided as part of a surface reconstruction benchmark [4]. For a triangle $t \in \mathcal{T}$ with edge lengths $\ell_1(t), \ell_2(t), \ell_3(t)$ and area $A(t)$, we compute the quality $Q(t)$, its average $Q_{\text{avg}}$, and the root mean square deviation in percent $Q_{\text{RMS}}$ as:

$$Q(t) = \frac{4\sqrt{3}A(t)}{\ell_1(t)^2 + \ell_2(t)^2 + \ell_3(t)^2}, \qquad Q_{\text{avg}} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} Q(t), \qquad Q_{\text{RMS}} = \frac{100}{Q_{\text{avg}}}\sqrt{\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}(Q(t) - Q_{\text{avg}})^2}.$$

We compare our algorithm to advancing front [2], multigrid Poisson [5], RIMLS [8], scale space [3], and Voronoi surface reconstruction [1]. See examples in Tables 1 and 2. The results show our method to be able to surpass the performance of the wide-spread comparison algorithms. For this, our algorithm only needs a single sweep over the geometry, while the others run several iterations. See [6] for further data and details.

## References


[1] Boltcheva, D., Lévy, B., 2017. Surface reconstruction by computing restricted voronoi cells in parallel. Computer-Aided Design 90, 123–134.

[2] Cohen-Steiner, D., Da, F., 2004. A greedy delaunay-based surface reconstruction algorithm. The Visual Computer 20, 4–16.

[3] Digne, J., Morel, J.M., Souzani, C.M., Lartigue, C., 2011. Scale space meshing of raw data point sets. Computer Graphics Forum 30, 1630–1642. doi:10.1111/j.1467-8659.2011.01848.x.

[4] Huang, Z., Wen, Y., Wang, Z., Ren, J., Jia, K., 2022. Surface reconstruction from point clouds: A survey and a benchmark. arXiv preprint arXiv:2205.02413 .

[5] Kazhdan, M., Hoppe, H., 2019. An adaptive multi-grid solver for applications in computer graphics. Computer Graphics Forum 38, 138–150. doi:10.1111/cgf.13449.

[6] Lipschütz, H., Reitebuch, U., Polthier, K., Skrodzki, M., 2023. Isotropic point cloud meshing using unit spheres (ipcms). arXiv preprint arXiv:2305.07570 doi:10.48550/arXiv.2305.07570.

[7] Lipschütz, H., Skrodzki, M., Reitebuch, U., Polthier, K., 2021. Single-sized spheres on surfaces (S4). Computer Aided Geometric Design 85, 101971.

[8] Öztireli, C., Guennebaud, G., Gross, M., 2009. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. Computer Graphics Forum 28, 493–501.

# Geometric deep learning for shape optimization in computational architecture

**Andrea Favilli**[a,b], Francesco Laccone[a], Paolo Cignoni[a], Luigi Malomo[a], Daniela Giorgi[a]

[a] *Institute of Information Science and Technologies "A. Faedo" (ISTI), National Research Council of Italy (CNR), via G. Moruzzi 1, Pisa, 56124, Italy*
[b] *University of Pisa (Italy), Lungarno Pacinotti 43, Pisa, 56126, Italy*

*Keywords:* Geometric deep learning, Freeform surface, Shape optimization, Structural design

**Abstract**

Contemporary architecture strongly relies on *freeforms*, complex shapes that are easily displayable through the use of design software tools. Indeed, software-assisted shape exploration satisfies designers, who are driven mainly by aesthetic purposes. However, when it comes to to feasibility freeforms are still difficult to deal with. Transforming a geometric concept in a reliable product at a building scale is a multi-constrained optimization problem. Several hard constraints arise from fabrication requirements (statics, lighting, space prescriptions), as expression of customer (cost, economy of materials) or society demands (environmental impact, pollution). We pursue the ambition of putting all the pieces together. In this poster, we describe a geometric deep learning method to edit freeform shapes in order to minimize the structural compliance and preserve the original design intent. The task is ideal for geometric deep learning [2], since the hidden relationship of different information layers concur to solve a hard problem with a geometrical flavour. In particular, we developed a statics-driven shape optimization tool for *grid shell* structures. Grid shells are 3D networks of straight beams rigidly jointed at the nodes, particularly known for the efficiency on covering large amounts of space with minimum use of materials.

Our learning model takes as input grid shells encoded as triangle meshes, exploits different geometric input features and predicts optimal shapes through vertex displacements: Figure 1 presents the pipeline, and Figure 2 shows an example. Hidden layers follow the GATv2 graph attention paradigm [1] on nearest neighbor graphs in the feature space, weights are fixed according to a differentiable structural strain energy employed as a training loss. We are also working on a graphical user interface that allows the user to control some parameters. The user can drive the procedure with a videotape-style set of buttons (*Play*, *Pause*, *Rewind*, *Replay*).

Our research is framed into *Architectural Geometry* [4], a multidisciplinary branch of research that finds geometric solutions to building feasibility problems. In this setting we propose a *fabrication-aware* design paradigm, which takes into account functionality criteria from the initial design phase, usually led by aesthetic purposes only. This shortens the fabrication pipeline, as it prevents intermediate outcomes to be sent back to the designer, saving hence time and improving results.

Future work includes the development of more refined means to control the shape, including both improved user involvement and mathematical instruments. Indeed, the user could enter the execution loop not only by changing parameters, but also by manual shape modifications: adequate tools in this sense are cages and control structures. Moreover, the input design preservation could be controlled also through sophisticated regularizers, based for instance on topological descriptors that analyze the preservation of geometric features [3].

*Email address:* andrea.favilli@isti.cnr.it (**Andrea Favilli**)

The method and the results, even in comparison with some state-of-the-art tools, are the subject of a paper which is currently submitted to a top-rank journal. We reserve the right to add further development in the poster, if available on the date.

*Results*

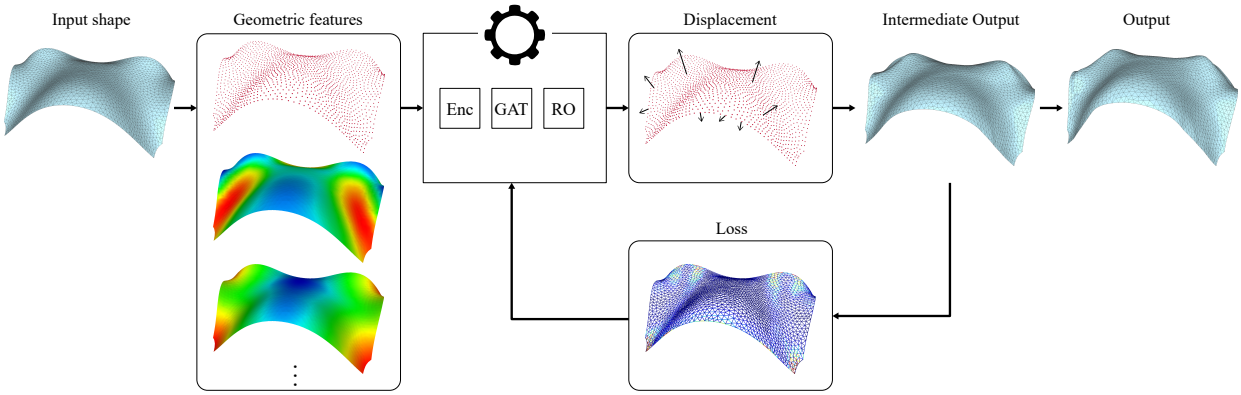
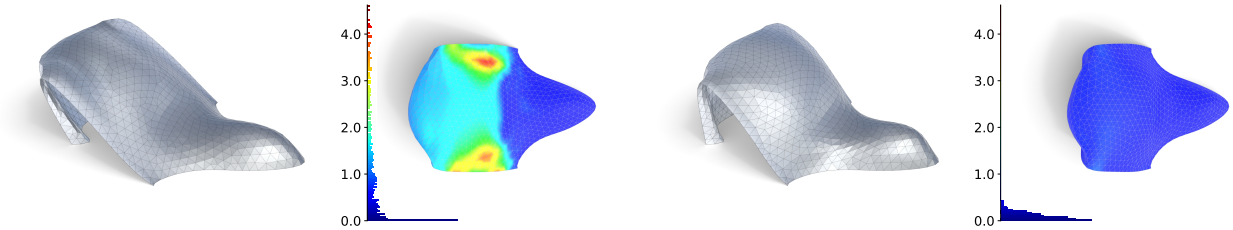



Figure 1: The input mesh, augmented with a set of geometric features (vertex coordinates, normals, curvature, geodesic distances) is fed to a network for 3D deep learning. The network is made up of an encoder (Enc), attention-based graph convolutional layers (GAT), and a MLP readout function (RO) which outputs learned vertex displacements. The network learns the optimal vertex displacements with respect to a differentiable strain energy loss, after a gradient descent procedure on the single mesh itself.




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

Figure 2: (a) A 3D mesh representing the first design of a grid shell, and (b) the deformation it would undergo due to Service Load, in false colors; deformation is expressed in meters. (c) The optimized grid-shell produced by our 3D deep learning model, which preserves the design intent while reducing deformation (d).

## References

[1] Brody, S., Alon, U., Yahav, E., 2022. How attentive are graph attention networks?, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=F72ximsx7C1.

[2] Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P., 2021. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. URL: https://arxiv.org/abs/2104.13478, doi:10.48550/ARXIV.2104.13478.

[3] Hensel, F., Moor, M., Rieck, B., 2021. A survey of topological machine learning methods. Frontiers in Artificial Intelligence 4. URL: https://www.frontiersin.org/articles/10.3389/frai.2021.681108, doi:10.3389/frai.2021.681108.

[4] Pottmann, H., Bentley, D., Asperl, A., Hofer, M., Kilian, A., 2007. Architectural Geometry. Number v. 10 in Architectural geometry, Bentley Institute Press. URL: https://books.google.it/books?id=bIceAQAAIAAJ.

# Spatial $n$-Arc Spline Approximation

Stephan Alt[a], Hartmut Prautzsch[b]

[a] *MACK Rides, Mauermattenstraße 4, 79183 Waldkirch, Germany*
[b] *Karlsruher Institut für Technologie, Am Fasanengarten 5, 76131 Karlsruhe, Germany*

*Keywords:* approximation, biarc, splines, non-linear optimization

**Abstract**

We propose an algorithm that solves the least-squares approximation problem with $n$-arc splines given a fixed $n$ and a discrete, ordered sequence of $m$ points in $\mathbb{R}^3$. At the company MACK Rides, arc splines are used to describe roller coaster rails that are manufactured as bent metal tubes. We approximate spatial freeform curves given by their equidistant sampling, using arc splines with the least number of arcs within some given error tolerance. We achieve this by solving the problem for an increasing number of arcs until the approximation error falls below the given bound.

As Hoschek [1], we represent $n$-arc splines by the intersections of their arc end tangents and non-linear conditions for the arc end points, see Figure 1. Using simple examples in the Euclidean plane, we demonstrate how our method approaches the optimal solution using a two-layered approach, first finding the least-squares approximation for any given fixed segmentation of the input, and then optimizing the segmentations using a generic global optimization scheme. Using several data sets, we show that our method can be used on a wide range of input shapes, as long as each individual arc covers less than a semicircle. Depending on the input data, very biased segmentations can lead to individual arcs turning through nearly 180° such that the control polygon diverges during optimization. We handle this drawback of the representation gracefully in our current implementation.

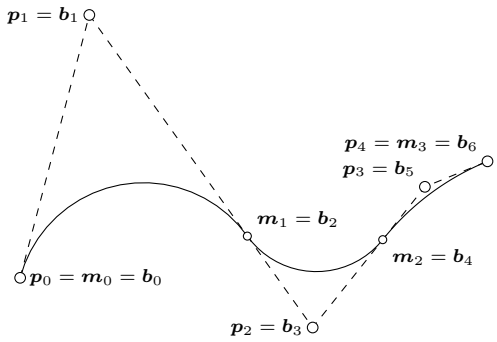



Figure 1: An arc spline with $n = 3$ and its control polygon

*Email addresses:* stephan.alt@mack-rides.com (Stephan Alt), prautzsch@kit.edu (Hartmut Prautzsch)

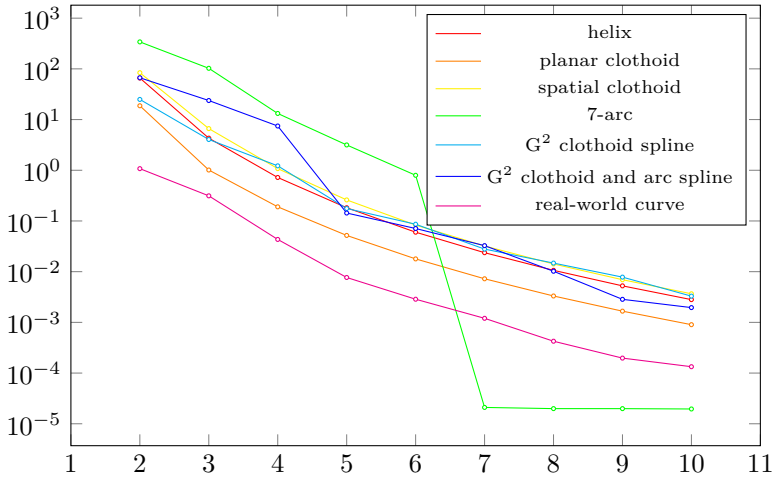



Figure 2: logarithmic plot of the least-squares error for increasing number of arcs $n$

*Results*

Figure 2 shows a logarithmic plot of the approximation error of our $n$-arc approximation algorithm for various exemplary inputs, one of which is shown in Figure 3 together with two $n$-arc spline approximations. For both the *helix* and the *planar clothoid*, our algorithm generates $n$-arcs with even segmentations, which suggests that the *discrete clothoid* approximation by Meek and Walton [2] is optimal. For the *spatial clothoid* with both linearly increasing curvature and torsion, we found that the approximating $n$-arc puts more emphasis on the part of the curve with higher torsion, leading to an uneven segmentation with strictly decreasing arc-lengths of the circular arcs. In case the input itself is an $n$-arc spline and given enough arcs, our algorithm reconstructs the input to within floating point accuracy. The remaining examples demonstrate that our algorithm performs equally well for more involved inputs as well as real-world data, showing similar error convergence for increasing $n$.

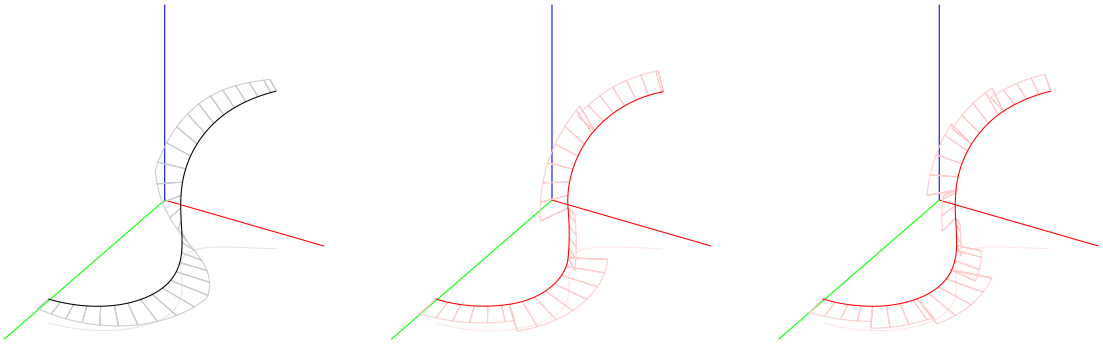


Figure 3: $G^2$ clothoid spline example (left) and its approximating 5-arc (center) and 9-arc (right) and their curvature comb.

## References

[1] Hoschek, J., 1992. Circular splines. Computer-Aided Design 24, 611 – 618. doi:https://doi.org/10.1016/0010-4485(92)90072-I.

[2] Meek, D., Walton, D., 2004. An arc spline approximation to a clothoid. Journal of Computational and Applied Mathematics 170, 59 – 77. doi:https://doi.org/10.1016/j.cam.2003.12.038.

# Scenic Routes With Weighted Points In 2D

Vijayraj Shanmugaraj[a], Lini Thomas[a], Kamalakar Karlapalem[a]

[a]*DSAC, IIIT Hyderabad, Professor CR Rao Rd, Gachibowli, Hyderabad, Telangana 500032*

*Keywords:* Scenic Routes, Graph Traversals, 2D point configurations, Scenic Points, Computational Geometry

**Abstract**

We develop the concept of scenic paths in a 2D space with respect to two points which have weights associated with them. Subsequently, we propose algorithms to generate scenic routes a traveller can take, which cater to certain principles which define the scenic routes. Following are the contributions of this paper: (1) mathematical formulation of a scenic point, (2) introduction of scenic routes formed by such scenic points in two-class point configurations in 2D spaces, and (3) design of scenic route generation algorithms that fulfil certain defined requirements.

## 1. Motivation and proposed solution

In a given 2D space, we can have points that have different levels of importance. In this paper, we indicate relative levels of importance by assigning the points a weight. Consider a red point $P_1$ and a blue point $P_2$, each point having its own weight ($w_1$ and $w_2$ respectively). A scenic point $P$ with respect to a given red point and a given blue point is a point from where the distance of the points from the scenic point is inversely proportional to the weight of the point. Note that this condition (when weights are substituted by heights of landmarks at those points) roughly translates to the apparent (perceived) height of the landmark at $P_1$ being equal to the apparent height of the landmark at $P_2$, where $d_1$ is the distance from $P$ to $P_1$ and $d_2$ is the distance from $P$ to $P_2$.

$$\frac{w_1}{d_1} = \frac{w_2}{d_2} \tag{1}$$

The locus of the point $P$ is a circle. We are given $n$ red points and $m$ blue points as inputs, and the arcs and intersection points of the circles resulting from the scenic point loci for each read blue pair gives rise to a graph, which we attempt to traverse through efficiently via a route satisfying the given set of requirements to the highest possible degree in the below listed order of importance: **1. Completeness:** Travelling on the route provided by the algorithm must allow one to have a view of all red-blue pairs. **2. Only scenic:** The route must consist of only scenic edges. **3. Minimal route length:** The total route length should be as low as possible. **4.Minimal repeated edges:** A route must contain a minimal number of repeated edges (stretches of paths that must be traveled more than once (repeated) to complete the entire route). **5. Minimal number of edges:** A route must not have a large number of edges.

We have three algorithms (All Curve Umbrella (ACU), All Curve Convex Hull (ACCH), and Dense Point Expansion (DPE) algorithms) which generate routes that try to satisfy the aforementioned requirements.

*Email addresses:* vijayraj.shanmugaraj@alumni.iiit.ac.in (Vijayraj Shanmugaraj), lini.thomas@email.com (Lini Thomas), kamal@email.com (Kamalakar Karlapalem)

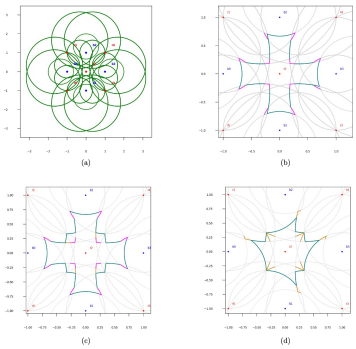


Figure 1: a) Points organised in a grid, with alternating red and blue points, and the resulting scenic graph. b) Route generated by the ACU algorithm on the graph c) Route generated by the ACCH algorithm on the graph d) Route generated by the DPE algorithm on the graph

| Algorithm | RL | NoE | NoRE | RE% |
|---|---|---|---|---|
| ACU | 7.42 | 48 | 4 | 8.33 |
| ACCH | 7.7 | 48 | 12 | 25 |
| DPE | 6.15 | 36 | 16 | 44.44 |

Table 1: Results for experiments on graph in Fig. 1 (Total Path length 558.96, No. of edges: 304)

*Results*

We have performed experiments on multiple graphs, and a subset of the results have been provided in this abstract. The Algorithms perform well on the metrics relevant to the requirements mentioned in the Motivation Section. (**Legend for table headers**: (RL: Total Route Length, NoE: Number of Edges in route, NoRE: Number of Repeated Edges in route, RE: Repeated Edge Percentage))

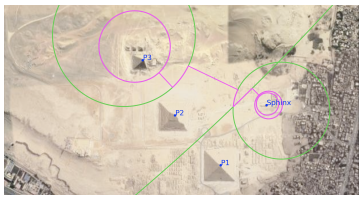



Figure 2: A scenic route on the area of the Great Pyramids, generated using the ACU algorithm (The edges included in the generated route are in pink, and the rest of the scenic paths are in green.)

| Algorithm | RL | NoE | NoRE | RE% |
|---|---|---|---|---|
| DPE | 194.71 | 41.59 | 11.12 | 27.07 |
| ACU | 218.29 | 46.13 | 3.77 | 8.03 |
| ACCH | 209.99 | 43.33 | 10.5 | 24.17 |

Table 2: Random graph experiment results (over 100 graphs, avg. number of nodes: 82.96, avg. number of edges: 176.96, avg. total edge length of graph: 41302.61)

## 2. Conclusion

In this work, we introduce the concepts of scenic points, scenic paths and scenic routes in two-class weighted point configurations in 2D spaces, a characterization of the properties of a scenic route, and an analogy of these conditions in a real world scenario. We present three algorithms to generate scenic routes, and analyze the routes generated by these algorithms. We finally picked some real-world scenarios for which we generate scenic routes. The full manuscript of the work can be found here[3].

## References

[1] Floyd, R.W., 1962. Algorithm 97: shortest path. Communications of the ACM 5, 345.
[2] Jarvis, R., 1977. Computing the shape hull of points in the plane, in: Proceedings of the IEEE Computing Society Conference on Pattern Recognition and Image Processing, New York. pp. 231–241.
[3] Shanmugaraj, V., Thomas, L., Karlapalem, K., 2023. Scenic routes with weighted points in 2d. arXiv:2306.04858.

# Jittor & JNeRF: a novel deep learning framework with meta-operators and unified graph execution, and its NeRF model zoo

Guo-Wei Yang[a], Shi-Min Hu[b]

[a]*Department of Computer Science and Technology, Tsinghua University*
[b]*Department of Computer Science and Technology, Tsinghua University*

*Keywords:* Deep Learning Framework, NeRF, 3D reconstruction

**Abstract**

Jittor [1] is a deep learning framework that leverages just-in-time (JIT) compilation to achieve higher performance and system customization. We introduce the concept of meta operators, which are similar to Numpy operators and can support flexible real-time compilation of operators into high-performance CPU or GPU code. Jittor utilizes unified graph execution, an optimized approach for executing computation graphs that combines the ease of use of dynamic graph execution with the efficiency of static graph execution. Jittor also offers operator fusion, cross-iteration fusion, and unified memory management as additional improvements. The paper also introduces JNeRF [5], a model zoo built upon Jittor that supports efficient NeRF models for novel view synthesis and 3D reconstruction tasks. Results demonstrate the fast training capabilities of JNeRF and showcase its high-quality rendering and reconstruction output.

**Method**

Jittor is a fully just-in-time (JIT) compiled deep learning framework. With JIT compilation, we can achieve higher performance while making systems highly customizable. Jittor provides classes of Numpy-like operators, which we call meta-operators. A deep learning model built upon these meta-operators is compiled into high-performance CPU or GPU code in real-time. To manage meta-operators, Jittor uses a highly optimized way of executing computation graphs, which we call unified graph execution. This approach is as easy to use as dynamic graph execution yet has the efficiency of static graph execution. It also provides other improvements, including operator fusion, cross iteration fusion, and unified memory.

A meta-operator is a general operator, which when specialized gives a class of operators with common properties that make them particularly amenable to optimization. Deep learning frameworks usually provide many operators to make it easy for users to build learning models. In fact, many of them do similar things, and can be expressed as specializations of more general higher-level operators. In particular, reindex is a very useful meta-operator which provides an arbitrary one-to-many mapping between its input and output. Various specialized operators such as broadcast, pad, and slice are particular forms of this operator, and belong to the reindex operator class. Another important meta-operator is reindex-reduce, which provides a many-to-one mapping. Sum and product are particular examples of reindex-reduce operators. The third meta-operator class comprises element-wise operators. Each has one or more matrix inputs, which should all have the same shape; the output matrix also has this shape. Results are computed element by element. Matrix addition is an example of a binary element-wise operator.

Unified graph execution is another major contribution of Jittor. According to the execution method of computational graphs, deep-learning frameworks can be based either on a static graph execution (also called

*Email addresses:* ygw19@mails.tsinghua.edu.cn (Guo-Wei Yang), shimin@tsinghua.edu.cn (Shi-Min Hu)

a define-and-run approach) or a dynamic graph execution (define-by-run, eager execution). Static graph based frameworks are efficient and easy to optimize, and dynamic graph based frameworks are easy-to-use and flexible. As an alternative, we propose our unified graph execution approach. Unified graph execution provides an imperative style interface which has the same flexibility as a dynamic graph. And it is also as efficient as a static graph.

Based on Jittor, we have proposed many model zoos that support different tasks, including an efficient heterogeneous NeRF[2] model zoo, JNeRF[5]. The emergence of NeRF brings a wide range of possibilities for real-world 3D reconstruction and rendering. Still, there are problems to be solved. Previous works have improved NeRF in the sampling technique, the position encoding method, the network structure, etc., but these improvements are difficult to combine conveniently due to different modules is not well decoupled. On the other hand, recent work [3] significantly speed up the core GPU computation of NeRF, leaving the deep learning framework overhead noteworthy. So some of them choose to replace the frameworks with pure CUDA programs, which limits the maintainability and extendability. Therefore, we propose JNeRF[5], a unified, efficient, framework-friendly NeRF model zoo. JNeRF includes many well-known NeRF works, such as Instant-NGP [3], NeuS [4], etc.

*Results*

To show the ability of JNeRF fast training, Fig.1 shows the rendering results of JNeRF Instant-NGP after training for 0.5 seconds, 1 second, and 5 seconds on RTX3090. It can be seen that JNeRF can get a clear result after 5 seconds of training. Fig.2 shows mesh model and colored model results reconstructed by NeuS [4].

Figure 1: Rendering results of Instant-NGP implemented by JNeRF after training for different times on RTX3090. Left: training for 0.5 seconds. Middle: training for 1 second. Right: training for 5 seconds.

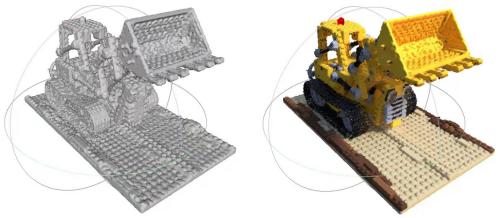


Figure 2: Mesh model reconstructed by NeuS and colored model results.

## References


[1] Hu, S.M., Liang, D., Yang, G.Y., Yang, G.W., Zhou, W.Y., 2020. Jittor: a novel deep learning framework with meta-operators and unified graph execution. Science China Information Sciences 63, 1–21.

[2] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R., 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM 65, 99–106.

[3] Müller, T., Evans, A., Schied, C., Keller, A., 2022. Instant neural graphics primitives with a multiresolution hash encoding. arXiv preprint arXiv:2201.05989 .

[4] Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W., 2021. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 .

[5] Yang, G.W., Liu, Z.N., Li, D.Y., Peng, H.Y., 2023. Jnerf: An efficient heterogeneous nerf model zoo based on jittor. Computational Visual Media 9, 401–404.

# Multi-patch parameterization method for isogeometric analysis using singular structure of cross-field

Ye Ji[a], Yi Zhang[a], Chun-Gang Zhu[a]

[a]*School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China*

*Keywords:* Isogeometric analysis, Analysis-suitable parameterization, Multi-patch configuration, Cross field, Boundary element method, Domain partition

**Abstract**

The cutting-edge numerical methodology of isogeometric analysis [1] offers the potential to seamlessly integrate Computer-Aided Design and Computer-Aided Engineering, effectively bridging the gap between the two domains. Most Computer-Aided Design systems focus exclusively on the boundary representation of models during the design phase, whereas a spline-based mapping is essential in the analysis stage, commonly referred to as domain parameterization. However, generating analysis-suitable parameterizations from existing boundary representations continues to be a considerable challenge in the isogeometric design-through-analysis process, especially for computational domains featuring intricate geometries, such as high-genus cases.

To tackle this challenge, we propose a cross-field-based multi-patch parameterization method for computational domains. Initially, we employ the boundary element method to solve vector field functions across the computational domain. Subsequently, we create a one-to-one mapping between the vector field and the cross-field, thereby obtaining the cross-field. By analyzing the singular structure of the cross-field, we ascertain the position information and topological connection relations of singularities and streamlines. Furthermore, we introduce a simple and effective technique for computing streamlines.

We introduce a novel segmentation strategy for dividing the computational domain into several quadrilateral NURBS sub-patches. After establishing the multi-patch structure, we devise two techniques for generating analysis-suitable multi-patch parameterizations. The first technique expands upon the barrier function-based approach [2], while the second technique yields smoother parameterizations by including the control points at the interfaces of the sub-patches within the optimization model.

Numerical experiments showcase the effectiveness and robustness of the proposed method, highlighting its potential to enhance the isogeometric analysis process.

*Email addresses:* jiye@mail.dlut.edu.cn (Ye Ji), cgzhu@dlut.edu.cn (Chun-Gang Zhu)

*Results*

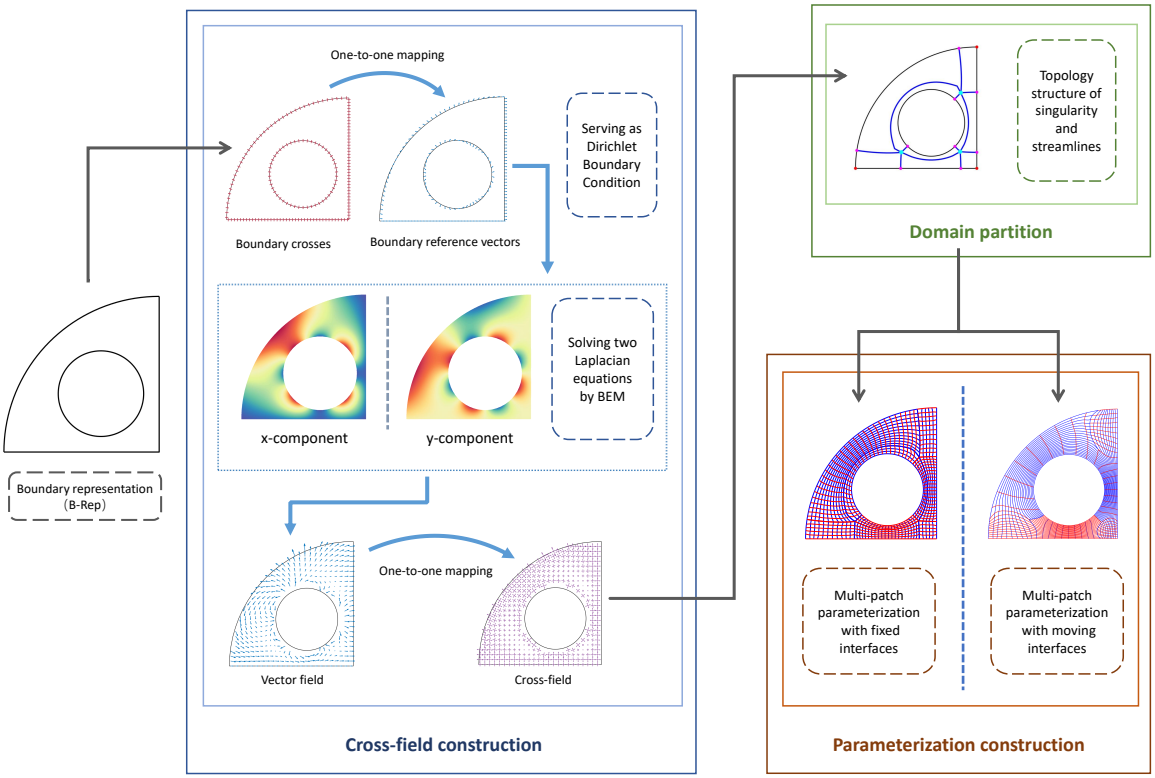



Figure 1: The overall workflow of our method

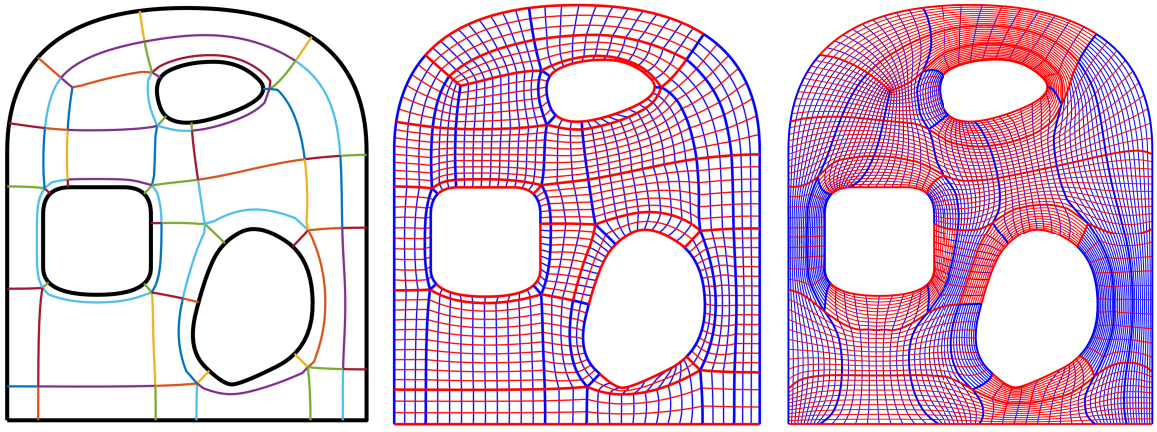


Figure 2: Left: Domain partition; Middle: Fixed-interface method; Right: Moving-interface method

## References

[1] Hughes, T.J., Cottrell, J.A., Bazilevs, Y., 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering 194, 4135–4195.

[2] Ji, Y., Yu, Y.Y., Wang, M.Y., Zhu, C.G., 2021. Constructing high-quality planar NURBS parameterization for isogeometric analysis by adjustment control points and weights. Journal of Computational and Applied Mathematics 396, 113615.

# The Volumetric Integer-Grid Map based Hexahedral Meshing Pipeline

Hendrik Brückler[a], Heng Liu[b], David Bommes[b], Marcel Campen[a]

[a] *Osnabrück University, Germany*
[b] *University of Bern, Switzerland*

*Keywords:* Mesh Generation, Meshable Frame Fields, Parametrization Quantization, Motorcycle Complex, Multi-Block Meshes, Bijective Mapping

**Abstract**

The idea of hexahedral mesh generation based on 3D Integer-Grid Maps has been around for more than a decade now [7], generalizing ideas from the 2D quadrilateral mesh setting [5, 2]. This approach is highly promising, because—at least in theory—it is very flexible, versatile, and expressive, conceptually enabling the generation of boundary-aligned, feature-respecting, anisotropic, adaptively sized, semi-regular hexahedral meshes of high quality. It consists of multiple stages and subproblems, such as the generation of a global seamless parametrization of a given object, possibly guided by a 3D frame field, the determination of suitable integer values for discrete degrees of freedom that control mesh resolution and connectivity, the reparametrization to an integer-grid map respecting these values, and the final extraction of a hexahedral mesh from the map.

For more than ten years, however, this concept has been notorious for lacking robust solutions for many of its subproblems. And further, even the non-robust best-effort approaches proposed for these so far, had often not been available in form of open source implementations facilitating further research and exploration.

Recently, in two projects, one hosted at the University of Bern, one hosted at Osnabrück University, we have been able to close multiple of the robustness gaps of this integer-grid map based hexahedral meshing pipeline. These solutions are reported in several recent articles, addressing in particular:

- Singularity Meshability [6],
- Numerical Sanitization [4],
- Volumetric Partitioning [4],
- Volumetric Quantization [3].

Implementations of these solutions are now made available to the community in the form of four open source libraries:

- libAlgoHex (`https://lib.algohex.eu`),
- TrulySeamless3D (`https://github.com/HendrikBrueckler/TrulySeamless3D`),
- MC3D (`https://github.com/HendrikBrueckler/MC3D`),
- QGP3D (`https://github.com/HendrikBrueckler/QGP3D`).

*Email addresses:* `hbrueckler@uos.de` (Hendrik Brückler), `heng.liu@unibe.ch` (Heng Liu), `david.bommes@unibe.ch` (David Bommes), `campen@uos.de` (Marcel Campen)

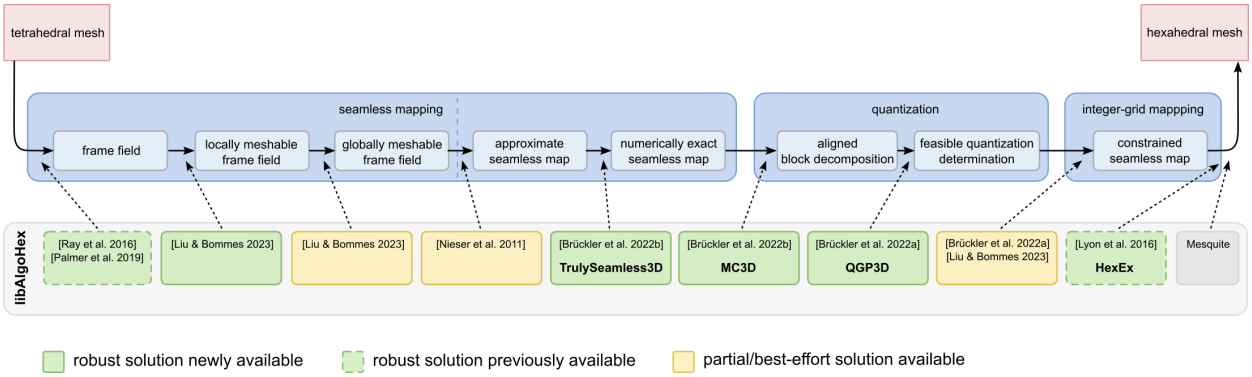



Figure 1: Overview of the algorithmic pipeline, from tetrahedral mesh input to hexahedral mesh output.

In a coordinated effort, these libraries have been integrated in such a manner that they can readily be used to execute the entire pipeline from input tetrahedral mesh to output hexahedral mesh, as illustrated in Figure 1.

We hasten to emphasize that a few subproblems remain that lack a robust solution so far (especially globally meshable singularity structure determination and integer-constrained remapping). For these, methods are included in this collection of integrated libraries as well, of best-effort type, so as to enable the use and exploration of the entire pipeline. These remaining challenges are the topic of ongoing research, with the goal of ultimately establishing a pipeline that is fully reliable end-to-end. We hope that the availability of this implementation can foster further work towards this goal.

*Results*

Using this collection of libraries, significant advances in terms of robustness can be observed. For instance, looking at the challenging HexMe dataset [1] of hexahedral meshing problem instances, now a majority (58%) of the instances can be successfully processed, assuming a sufficiently fine mesh resolution is targeted. While the remaining gap towards full success illustrates the above point that further research is necessary, note that only around 10% were achievable with a previous internal implementation *without* the above mentioned recent advances.

*Acknowledgements*

The underlying projects are funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (AlgoHex, grant agreement No 853343) and the Deutsche Forschungsgemeinschaft (DFG) - 42746936; 456666331.

## References

[1] Beaufort, P.A., Reberol, M., Kalmykov, D., Liu, H., Ledoux, F., Bommes, D., 2022. Hex Me If You Can. Computer Graphics Forum 41, 125–134.
[2] Bommes, D., Zimmer, H., Kobbelt, L., 2009. Mixed-integer quadrangulation. ACM Trans. Graph. 28, 77:1–77:10.
[3] Brückler, H., Bommes, D., Campen, M., 2022a. Volume parametrization quantization for hexahedral meshing. ACM Trans. Graph. 41.
[4] Brückler, H., Gupta, O., Mandad, M., Campen, M., 2022b. The 3D Motorcycle Complex for Structured Volume Decomposition. Computer Graphics Forum 41.
[5] Kälberer, F., Nieser, M., Polthier, K., 2007. QuadCover - surface parameterization using branched coverings. Computer Graphics Forum 26, 375–384.
[6] Liu, H., Bommes, D., 2023. Locally meshable frame fields. ACM Trans. Graph. 42.
[7] Nieser, M., Reitebuch, U., Polthier, K., 2011. CubeCover – Parameterization of 3D volumes. Computer Graphics Forum 30, 1397–1406.

# Surfaces from magnetic spheres

Henriette Lipschütz[a], Konrad Polthier[a]

[a] *Department of Mathematics and Computer Science, FU Berlin*

*Keywords:* physical modelling, surface approximation, magnetic spheres, graph theory

*Abstract*
We consider finite sets of magnetic spheres $\{s_1, \ldots, s_n\}$ made from neodymium. All spheres have the same radius $r > 0$ and are identically magnetized. Placed in $\mathbb{R}^3$, each sphere $s_i$ is characterized by the coordinates of its center $c_i$ and the direction of its magnetization $M_i$, which is pointing along the line of force through inifinity. These spheres can be adjusted into small configurations like pearls on a wire or circles where the spheres' centers form a regular $n$-gon. Especially the circular configurations can be used to build more complicated and highly symmetric configurations in the plane or in space.

*Formulation of minimization problem*
Magnets can be made in various sizes and shapes. Since the outer magnetic field of a (macroscopic) magnet in shape of a sphere coincides with the field of a dipole everywhere, the consideration of magnetic spheres is reasonable, since there is a suitable mathematical model of this kind of magnetic fields. In order to derive a formulation for the interaction energy of the complete set of spheres, first, the magnetic field at position $p$ produced by a point dipole $m$ is formulated as

$$B(m,p) = \frac{\mu_0}{4\pi}\left(\frac{3\langle m,p\rangle}{|p|^5}p - \frac{1}{|p|^3}m\right),$$

where $\mu_0$ denotes the permeability of free space which is constant. This leads to the formulation of the field of some magnetic dipole $m_i$ evaluated at the location of some other dipole $m_j$, where $p_i, p_j$ denote their locations:

$$B_i(p_j) = \frac{\mu_0}{4\pi}\left(\frac{3\langle m,p_j-p_i\rangle}{|p_j-p_i|^5}(p_j-p_i) - \frac{1}{|p_j-p_i|^3}m_i\right).$$

The interaction energy between dipole $m_j$ and and the magnetic field of $m_i$ is given by

$$U_{ij} = -\langle m_j, B_i(p_j)\rangle = \frac{\mu_0}{4\pi}\left(\frac{\langle m_i,m_j\rangle}{|p_j-p_i|^3} - 3\frac{\langle m_i,p_j-p_i\rangle\langle m_j,p_j-p_i\rangle}{|p_j-p_i|^5}\right),$$

which is symmetric in $i$ and $j$, i.e., $U_{ij} = U_{ji}$ for all $i \neq j$, $i,j \in \{1,\ldots,n\}$. Therefore, the total interaction energy of all spheres in the considered set is given by $U = \sum_{i<j} 2U_{ij}$. For further details, see [1].
The spheres have to fulfil two different assumptions:

- Spheres do only touch, never interpenetrate. Hence, for pairwise distinct spheres $i,j$: $|c_j - c_i| \geq 2r$.
- Since the magnetizations are supposed to be of equal magnitude, they are normalized: $|M_i| = 1$.

The optimization problem results in minimizing $U$ with respect to the aforementioned constraints. Such a configuration of spheres is said to be in *stable equilibrium* if the configuration returns to the original placement after small pertubations.

*Email addresses:* henriette.lipschuetz@fu-berlin.de (Henriette Lipschütz), konrad.polthier@fu-berlin.de (Konrad Polthier)

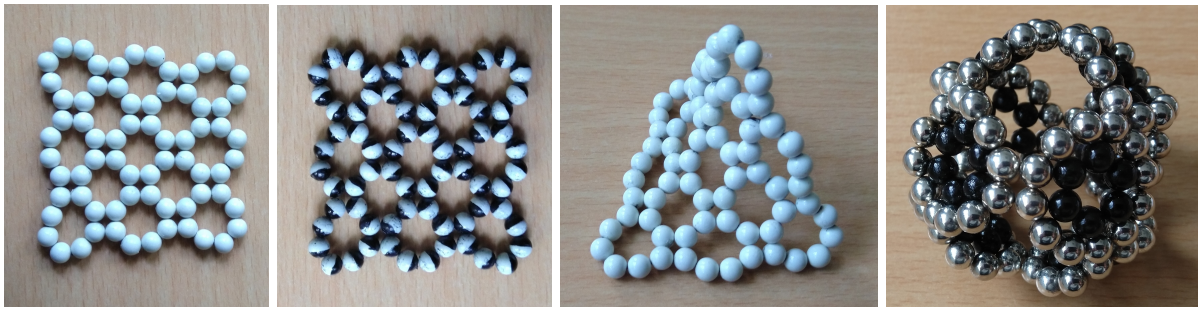


Figure 1: From left to right: Section of Archimedean 4-8-8 tiling; section of Archimedean 4-8-8 tiling, directions of magnetizations indicated by colored spheres; equilibrium in 3-spaces forming a saddle shaped 1-skeleton; approximation of a single building block of the triply periodic Schwarz P surface.

*Approximations of surfaces*

Starting from spheres divided into rings of consisting of different numbers of spheres, they can be assembled in various ways. See [3] for a collection of Archimedean tilings which are an interesting collection of examples since all edges are of the same length and can therefore be replaced by two touching spheres. Disturbing for example a section of the 4-8-8 tiling results in a saddle shaped configuration as depicted in Figure 1, third image from the left which appears to be stable.

Another way to arrange a set of rings is inspired by *triply periodic minimal surfaces*. An approximation of the Schwarz P surface is shown in Figure 1, on the very right hand side. Here, the placement of spheres and the number of spheres per circle is inspired by the symmetry group of the Schwarz P surface.

Since examples like the mentioned ones are stable in the above sense, minimizing the aforementioined energy does not give additional pairs of touching spheres. Hence, in addition to the characterization by the spheres' centers and the directions of magnetization, to each such configuration a graph can be associated. Every vertex represents a sphere and two vertices are connected if and only if the spheres the vertices represent touch. In case of a stable configuration, an embedding of this graph arises from the optimization problem introduced earlier.

*Open questions*

The observations described above give rise to various questions. A slightly deformed circle consisting of $n$ spheres in the plane is represented in terms of the graph introduced by a $n$-gon deformed correspondingly. Considering the directions of magnetization, they appear to be tangential to a (discrete) curve trough the spheres' centers. Hence, a natural question is whether there is a similar interpretation in the 3D case as well or how sufficiently many information can be derived to span at each vertex a fulldimensional tangent space. Since the configurations presented in Figure 1 come from classical smooth surfaces, it is worth to be investigated whether there is a discrete surface that can be derived from the given graph and vector field possibly similar to the ansatz presented in [2].

## References

[1] Edwards, B.F., Riffe, D.M., Ji, J.Y., Booth, W.A., 2017. Interactions between uniformly magnetized spheres. American Journal of Physics 85, 130–134. URL: https://doi.org/10.1119/1.4973409.

[2] Kotani, M., Naito, H., Omori, T., 2017. A discrete surface theory. Computer Aided Geometric Design 58, 24–54. doi:https://doi.org/10.1016/j.cagd.2017.09.002.

[3] Segerman, H., Zwier, R., 2017. Magnetic sphere constructions, in: Swart, D., Séquin, C.H., Fenyvesi, K. (Eds.), Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture, Tessellations Publishing, Phoenix, Arizona. pp. 79–86. URL: http://archive.bridgesmathart.org/2017/bridges2017-79.html.

# Semantically Enriched 3D Geometric Modeling for Urban Digital Twins

Andreas Scalas, Brigida Bonino, Chiara Romanengo, Tommaso Sorgente, Daniela Cabiddu, Michela Mortara, Simone Pittaluga, Michela Spagnuolo

*CNR-IMATI Genova, Italy*

*Keywords:* 3D Models, Mesh Annotation, Urban Intelligence

**Abstract**

An urban digital twin (UDT) is the virtual representation of real assets, processes, systems and subsystems (e.g., transportation, energy distribution, water usage, population, education, health, cultural heritage, etc.) of a city. By using and integrating heterogeneous data, UDTs actually allow monitoring the current status of cities and predicting/anticipating possible complex scenarios [1]. We focus on the design and development of the geometric layer of UDTs, that is the 3D digital representations of urban morphology and physical structures (i.e. buildings), properly enriched with heterogeneous semantic information. Our method aims to build a functional geometric model of the city, exploiting at best the available data, trying to achieve a representation that is able to support geometry-related queries [3].

**Method**

Our approach exploits 2D information from OpenStreetMap (OSM) and the 3D elevation data from Digital Elevation Models (DEM). Both Digital Terrain Models (DTM) and Digital Surface Models (DSM) coming from LiDAR techniques are taken into account to cross-reference both buildings and streets from OSM with elevation data. First, the procedure resolves geometric and topological issues in building boundary polygons; then, such polygons and street segments are added as constraints to triangulate the area between buildings. A 2D constrained Delaunay triangulation is generated and, for each point, its elevation is set according to the DEM: the elevation of points belonging to buildings is an average of the DTM cells in which the building is contained, and the height of each building is set equal to the average difference between corresponding DSM and DTM cells containing the building and extrude the building base accordingly. Finally, the extruded boundary of the buildings are closed by triangulating the corresponding polygon.

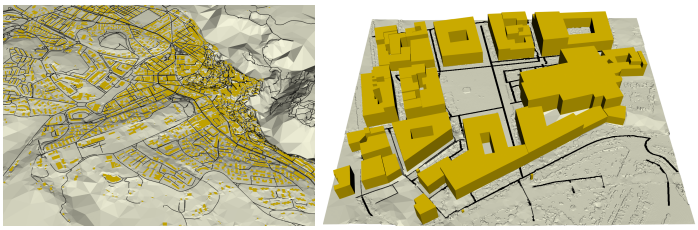
When the 3D triangulated surface is ready, we semantically enrich the model by importing elements from OpenStreetMap. Street edges and building triangles are labelled with the corresponding OpenStreetMap indexes. This creates a biunivocal association between portions of the geometry and OpenStreetMap entities and their attributes to be further interrogated and analyzed.

**Results**

The potential of our approach is shown by generating the geometric layer for the digital twin of the Italian city of Matera and Catania (see Figure 1). Such models are used to answer specific objectives posed by the

*Email addresses:* andreas.scalas@ge.imati.cnr.it (Andreas Scalas), brigida.bonino@ge.imati.cnr.it (Brigida Bonino), chiara.romanengo@ge.imati.cnr.it (Chiara Romanengo), tommaso.sorgente@ge.imati.cnr.it (Tommaso Sorgente), daniela.cabiddu@ge.imati.cnr.it (Daniela Cabiddu), michela.mortara@ge.imati.cnr.it ( Michela Mortara), simone.pittaluga@ge.imati.cnr.it (Simone Pittaluga), michela.spagnuolo@ge.imati.cnr.it (Michela Spagnuolo)

two municipalities, such as computing the in light/shadow areas given date and time, querying morphology-related information about streets, and supporting processes of monitoring the level of occupancy of some points of interest [3].

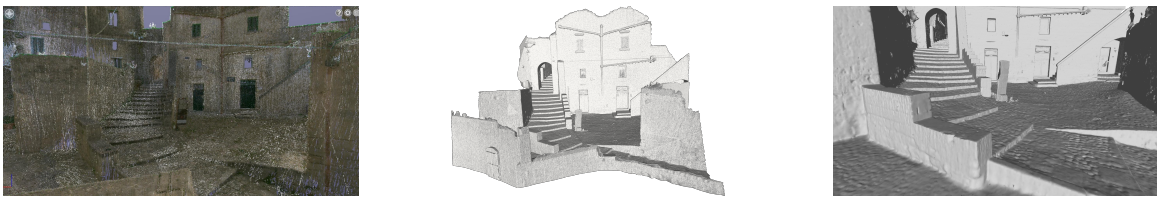


(a) Matera (b) Catania

Figure 1: Urban 3D models generated by our method for the two pilot cases of Matera and Catania, in Southern Italy. Yellow elements are labeled as "building", while black segments are labeled as "street".

### Ongoing and future works

We are currently working on including further salient entities, such as squares, green areas, public and commercial services, urban furniture and facilities, and others. Moreover, we are going to improve our modeling approach to be suitable for simulation of environmental parameters, such as air pollution prediction or rainfall runoff. Finally, additional efforts are currently put on designing new methods to generate top quality 3D models of urban areas starting from laser scanner acquisitions [2] (see Figure 2).



(a) (b) (c)

Figure 2: From laser scanning acquisition to 3D model. An example for the pilot case in Matera. (a) Point cloud from mobile laser scanning acquisition. (b) 3D triangulated mesh. (c) Detail of the triangulated mesh.

### Acknowledgements

The authors would link to thank the Matera and Catania projects. They are funded by the Italian Ministry of Economic Development ("Casa delle Tecnologie Emergenti" project) and by the Italian Agency for Territorial Cohesion ("Programma Operativo Complementare Città Metropolitane 2014-20") respectively.

### References

[1] Castelli, G., Cesta, A., Diez, M., Padula, M., Ravazzani, P., Rinaldi, G., Savazzi, S., Spagnuolo, M., Strambini, L., Tognola, G., Campana, E.F., 2019. Urban intelligence: a modular, fully integrated, and evolving model for cities digital twinning, in: 2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT IoT and AI (HONET-ICT), pp. 033–037. doi:10.1109/HONET.2019.8907962.

[2] Scalas, A., Cabiddu, D., Mortara, M., Pittaluga, S., Spagnuolo, M., 2022a. Mobile laser scanning of challenging urban sites: a case study in matera, in: Ponchio, F., Pintus, R. (Eds.), GCH 2022 - Eurographics Workshop on Graphics and Cultural Heritage, Delft, The Netherlands, 28-30 September 2022, Eurographics Association. pp. 19–22. doi:10.2312/gch.20221218.

[3] Scalas, A., Cabiddu, D., Mortara, M., Spagnuolo, M., 2022b. Potential of the geometric layer in urban digital twins. ISPRS International Journal of Geo-Information 11. doi:10.3390/ijgi11060343.

# Scenic Routes over Points in 2D Space

Loay Rashid, Lini Thomas, Kamal Karlapalem

[a] *Data Sciences and Analytics Center, IIIT Hyderabad*

*Keywords:* Scenic Routes, Graph Traversals, 2D point configurations, Scenic Points, Equidistant

**Abstract**

Consider a 2D coordinate space with a set of red and a set of blue points. We define a scenic point as a point that is equidistant to a red point and a blue point. The set of contiguous scenic points form a scenic path. The perpendicular bisectors to the line joining a red point and a blue point forms a scenic path between the red point and the blue point. A scenic route is a traversal made from scenic paths. In this paper, we address this novel problem by (i) designing algorithms for scenic route generation, (ii) studying the algorithms' different properties and (iii) analysing the routes generated by these algorithms. Scenic routes have applications in geo-spatial visualizations and visual analytics.

## Introduction

### *Our Motivation*

Humans have an implicit awareness of scenic beauty when traveling over various routes. The question arises: How do we transfer our scenic awareness over to two-dimensional data? Here, we define scenic beauty to be a uniform and balanced view of points of interest. A scenic point provides us with an equidistant view of at least two points of interest with the purpose of achieving balanced views.

The onus in this work is to show the nature of scenic routes in a two-dimensional environment. We consider points of interest to be either red or blue colored, with scenic beauty being an equidistant view of a red and a blue point. In particular, we introduce scenic routes to pairs of (red, blue) points and give illustrative situations where such routes are viable in the real world.

Consider the Giza Necropolis as shown in Fig. 1, specifically the pyramid of Khufu and the pyramid of Khafre. Treating the tops of the two pyramids as points of interest (blue), we can draw a scenic route (pink) through the necropolis. Highlighted in green are two city roads on the scenic route, such points are accessible to the viewer and give a scenic view of the pyramids.

### *Scenic Route Problem*

Consider a rectangular region with $N$ red and $M$ blue points. We define a **scenic point** as a point that is equidistant to a red point and a blue point in a pair of red-blue points. Each point on the perpendicular bisector of a line joining a pair of red-blue points is scenic. Therefore, the entire perpendicular bisector is a scenic path. Scenic routes are generated by putting together segments of these bisectors/scenic paths. The intersection points of multiple bisectors offer multiple scenic views as they produce an equidistant view to multiple pairs of points. Moreover, these intersection points offer opportunities to change directions, i.e., move from one bisector to another while maintaining a scenic view.

*Email addresses:* loay.rashid@alumni.iiit.ac.in (Loay Rashid), lini.thomas@iiit.ac.in (Lini Thomas), kamal@iiit.ac.in (Kamal Karlapalem)

Figure 1: The Giza Necropolis

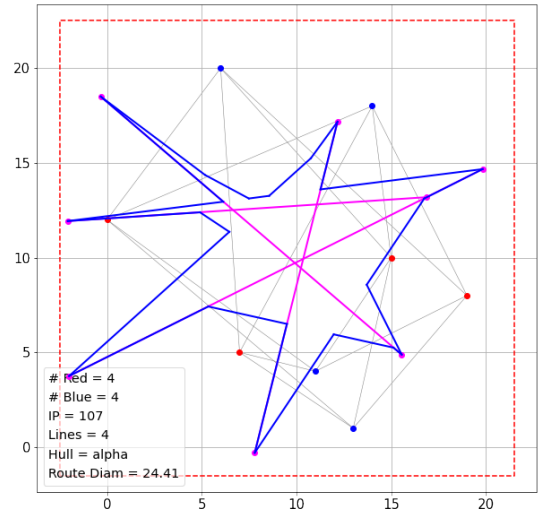



Figure 2: Red/Blue points, bisector intersection points, and the corresponding scenic graph (in pink and blue).

This work primarily deals with methods to find scenic routes in two-dimensional space. Since there can be multiple ways to traverse over these scenic paths, we define the following set of requirements for scenic routes, in decreasing order of importance: **1. Only Scenic**: The route must consist of scenic paths. **2. Completeness**: Travelling on the route must allow one to have a view of a large number of (preferably all) red-blue pairs. **3. Minimal Edges**: A route must not have a large number of edges. Traveling on a route must allow for long, uninterrupted stretches of scenic points. **4. Minimal Repeated Edges**: A route must minimize the number of repeated edges. Repeated edges are defined as stretches of bisectors that must be traveled multiple times (repeated) to complete the entire route.

We present two algorithms, the Min-Max Hull algorithm and the Densest Line algorithm. The two algorithms perform differently on each of the requirements, with densest line providing routes that have long, view-dense stretches, while the Min-Max Hull algorithm is better if one wants to have a shorter route that gives a large number of views. We run our algorithms on multiple graphs and compare the performance of the generated routes against the scenic requirements. We provide an example of a route (in pink and blue) generated using the densest line algorithm in Figure 2.

## Conclusion

In this paper, we introduced the concept of scenic routes in two-class point configurations in 2D spaces and a characterization of the properties of a scenic route. We present two scenic route generation algorithms and analyze the routes generated by these algorithms. Finally, we generate scenic routes on the Capitol Hill area. Ours is a preliminary work on this problem that opens up exciting theory and practical implementation challenges to design and analyze scenic route algorithms. The full manuscript of the work can be found here [4].

## References

[1] Barber, C.B., Dobkin, D.P., Huhdanpaa, H., 1996. The quickhull algorithm for convex hulls. ACM Trans. Math. Softw. 22, 469–483. URL: https://doi.org/10.1145/235815.235821, doi:10.1145/235815.235821.

[2] Edelsbrunner, H., Kirkpatrick, D., Seidel, R., 1983. On the shape of a set of points in the plane. IEEE Transactions on Information Theory 29, 551–559. doi:10.1109/TIT.1983.1056714.

[3] Guha, S., Rastogi, R., Shim, K., 1998. Cure: An efficient clustering algorithm for large databases. SIGMOD Rec. 27, 73–84. URL: https://doi.org/10.1145/276305.276312, doi:10.1145/276305.276312.

[4] Rashid, L., Thomas, L., Karlapalem, K., . Scenic routes over points in 2d space (full paper) url: https://drive.google.com/file/d/1g7I4noUjygJvLYMV6Vlde5RKp5e3_6we/view?usp=sharing .

# Soft Robots by Geometric Computing

Christopher-Denny Matte, Tsz-Ho Kwok

*Concordia University, Montreal, QC H3G 1M8, Canada*

*Keywords:* soft robotics, deformation prediction, geometric computing, hyperelasticity

**Abstract**

The growing interest in soft robotics has underscored the need for novel simulation techniques. Soft robots are gaining popularity because of their inherent compliance, safe human-robot interaction, and adaptability. However, simulating these robots poses numerous challenges due to their infinite degrees of freedom, large deformations, non-linear material, and non-intuitive inputs for traditional simulation methods. Conventional simulation approaches have struggled to handle complex geometries and nonlinear deformations. In response, this research aims to bring the accuracy of traditional Finite Element Methods to geometry-based simulation frameworks which focus more on surface representation and processing. Geometry-based frameworks have demonstrated their speed, robustness, and intuitive nature in modelling interactions and actuation, particularly for soft robotics. Nevertheless, current methods either handle only linear materials or oversimplify the models to maintain speed at the expense of accuracy. This research investigates a novel method for implementing non-linear material capabilities with negligible computational overhead for geometry-based methods. A local-global approach is used, where the behaviour of an element is determined locally, and stitched back with other elements in a least-square global step. Locally an element follows a blending step where a multi-objective function determines its desired shape which can be modified in real-time without touching the global stiffness matrix. This function can consider desired behaviours such as expansion for pneumatic actuation or contraction for shape memory alloys, and stiffness of an element. Material curves are fitted with a polynomial expression, which can determine the tangent modulus, or stiffness, of a material based on strain energy. The moduli of all elements are compared to determine the relative shape factors used to establish an element's blended shape. This process is done dynamically to update a material's stiffness in real-time, for any number of materials, regardless of linear or non-linear material curves.

*Results*

The geometry-based formulation allows for setting a target shape $\mathbf{V_i^B}$ for each element. When there are two different materials in consideration, an element's stiffness is formulated based on how much it cares about its original shape compared to simply conserving its volume. As such, if one element cares about its original shape more than another, it will better maintain its shape during the energy minimization process thus making it act stiffer. Fig. 1 a), depicts how the shape factor $\omega$ is used for materials of different stiffness and the mathematical equivalent is found in Equation 1.

$$\mathbf{V_i^B} = \omega_i * \mathbf{V_i^R} + (1 - \omega_i)\mathbf{V_i^v}. \tag{1}$$

The stiffness of an element is done within the blended shape by selecting a rigidity shape factor $\omega_i$ between a rigid shape $\mathbf{V_i^R}$ and a volume conserving shape $\mathbf{V_i^v}$. The mechanical stiffness is typically reflected

*Email addresses:* christophermatte@outlook.com (Christopher-Denny Matte), tszho.kwok@concordia.ca (Tsz-Ho Kwok)

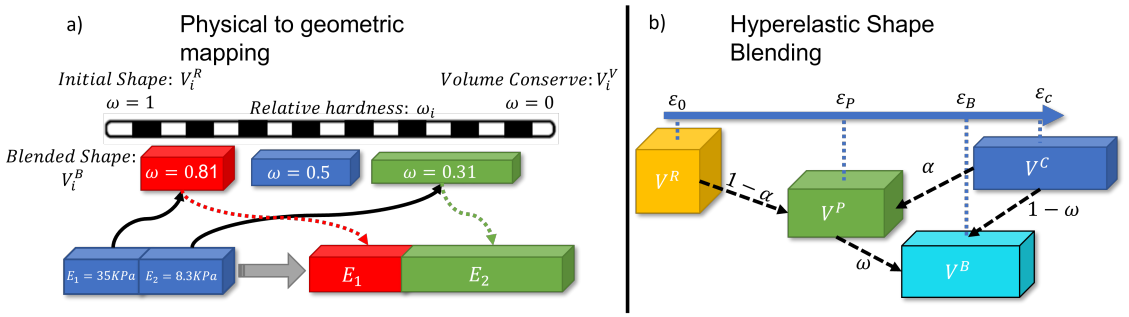



Figure 1: Visualization of physical properties with relation to shape factor. A larger shape factor results in less deformation, similar to having a larger Young's Modulus.

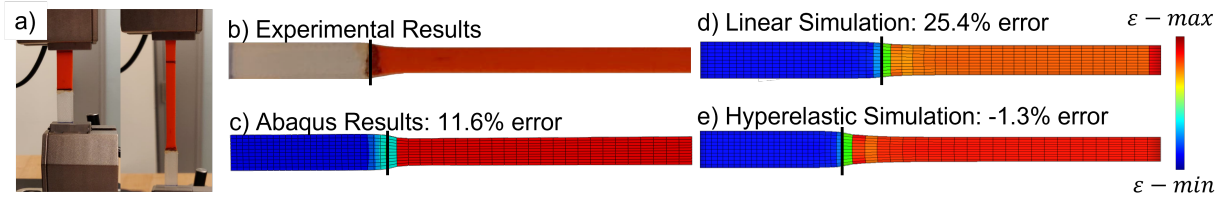



Figure 2: Cheetah TPU(white) and NinjaFlex TPU(orange) filaments in tension at 100% elongation. a) Test set-up b) Experimental results; c) Hyperelastic Abaqus results; d) Linear simulation of our framework; e) Hyperelastic simulation in our framework.

through the Young's Modulus, $E$, of the material. However, the geometric definition of rigidity has a normalized scale from 1 for perfectly rigid, to 0 for an element that solely cares about its volume; requiring a mapping function, $\Gamma(R_M) \mapsto R_G$, to reflect the mechanical implications accurately. The geometric ratio of the two shape factors (i.e., $R_G = \omega_x/\omega_y$) corresponds to the desired ratio of its physical counterpart (i.e., $R_M = E_x/E_y$), where $E_x$ and $E_y$ are the Young's modulus of two theoretical materials. Elements are given their own relative stiffness ratio $\omega_i$ for the current time-step by comparing their current stiffness to the stiffness of other elements. The process assumes if all elements are equally stiff, then $\omega = 0.5$. It is to be noted that the geometric formulation's least square problem has a constant global matrix and does not need to be updated to reflect new material properties. To prevent instability and error accumulation, an intermediate shape $\mathbf{V_i^P}$ is constructed based on the initial shape of an element and the current deformed shape as the new reference point in the shape blending equation. The process is visualized in Fig.1 b). Equation1 is then modified such that $\mathbf{V_i^R}$ is replaced by $\mathbf{V_i^P}$ defined in Eq.2. The updated formulation is Eq.3.

$$\mathbf{V_i^P} = (1-\alpha)\mathbf{V_i^R} + \alpha * \mathbf{Vi^v}, \tag{2}$$

$$\mathbf{V_i^B} = \omega_i * \mathbf{V_i^P} + (1-\omega_i)\mathbf{V_i^v}. \tag{3}$$

Preliminary results of the proposed method can be seen in Fig.2 where a tensile test is performed with two different 3D printed thermoplastic polyurethanes (TPU) filaments and the relative elongation, denoted by the black line, is compared. A simulation with linear material properties and another with hyperelastic properties using the geometric approach are compared to experimental results and a hyperelastic simulation performed in Abaqus. Abaqus took 27.2 seconds for the simulation with an error of 11.6%, while our framework took 18.3 seconds (1.49x faster) with an error of −1.3%. It is to be noted the Abaqus calculation time increased with deformation, while the geometry approach remained constant throughout.

## References

# Numerical Integration for Subdivision IGA

Alexander Dietz[a]

[a]*Faculty of Informatics, USI Lugano*

*Keywords:* Subdivision, Isogeometric Analysis, Integration, Quadrature, Elliptic PDEs, Adaptivity

**Abstract**

Due to their built-in refinability, subdivision algorithms are a promising tool for the construction of function spaces in the context of isogeometric analysis. In this poster we summarise the opportunities and challenges of this approach for the surface and volume cases. In particular, we highlight the challenges in the areas of algorithms, analysis, and implementation. In the area of algorithms, for the 2-dimensional case, it is necessary to (re)discover subdivision algorithms with subdominant eigenvalues $\frac{1}{2}$. For the 3-dimensional case, to the best of our knowledge, there are no subdivision algorithms with such properties, and not even with equal subdominant eigenvalues, which leads to an eigenshell that converges first to a plane and then to a line. In the area of analysis, the 2-dimensional case is largely done, but the 3-dimensional case is completely open as there is no generalisation of the discrete Fourier transform to 3-dimensional objects. Finally, the implementation for quadrature to solve PDEs is non-trivial in the 2-dimensional case and even less so in the 3-dimensional case. This is the focus of the rest of the poster.

Specifically, to construct the Galerkin system, it is necessary to compute integrals on subdivision surfaces and volumes. Since errors in the integration affect the accuracy of the simulation, these integrals should be computed efficiently and with high precision. In this poster, we first discuss integration strategies for the overall integration, including adaptivity and error control, as well as for the integration around irregularities. With our 'finite-capping-strategy' we approximate the infinite fragmented patches around an irregularity by Bézier patches. In doing so we respect the smoothness to the outer patches and use the remaining degree of freedom to interpolate the central point. For Bézier patches of higher order is is also possible to interpolate even more points. We also compare this strategy with other strategies (see Figure 1) like the 'brute-force-strategy' (ignoring the irregular part) and the 'naive-stragegy' (using gaussian quadrature directly on the infinite fragmented patch).

We then show the convergence speed of the simulation based on high-precision integration in a benchmark problem: At first we generate a uniform $n$-gon for $n = 3, \dots 8$ in $\mathbb{R}^2$. On this physical domain we construct an elliptic PDE and approximate it with 7 steps of uniform refinement. For each step we measure the $L_2$-error and the $L_\infty$-error. For $n \neq 4$ we get an approximation speed of $\lambda^{3n}$ for the $L_2$-error and $\lambda^{2n}$ for the $L_\infty$-error, where $\lambda$ is the subdominant eigenvalue of the subdivision algorithm (see Figure 2). For $n = 4$ we get a convergence speed of $\lambda^{4n}$ for both the $L_2$- and the $L_\infty$-error.

Since irregularities slow down the convergence speed of the simulation, our experimental results lead to the assumption that we need the high resolution of the refinement just around the area of the irregularity. Therefore, we present further experimental results for an adaptive (in the sense of basis functions) algorithm using truncated hierarchical B-Splines and show that adaptivity is a useful tool to overcome the lack of convergence speed (see Figure 3).

*Email address:* dietz@mathematik.tu-darmstadt.de (Alexander Dietz)

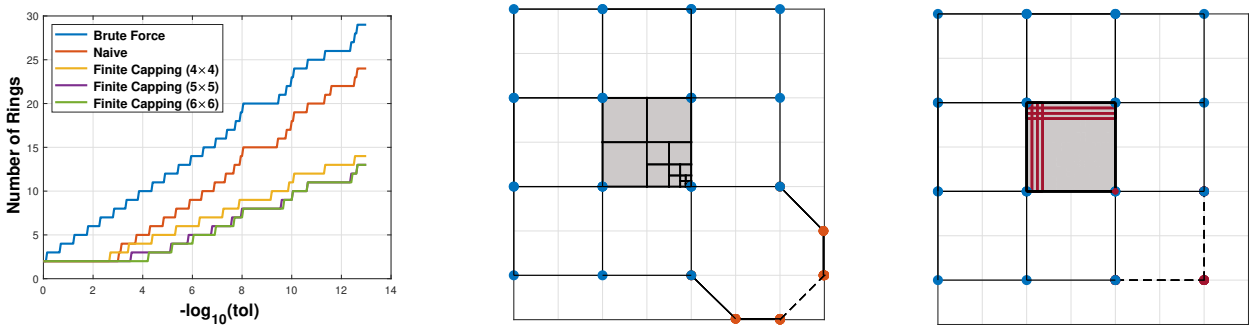



Figure 1: (left) Comparison of different integration methods. On the $x$-axis one can see the tolerance given to the algorithm and on the $y$-axis the amount of rings which means the amount of refinement steps around the irregularity for the given error tolerance. (middle) An infinite fragmented patch and the control points, by which it is influenced. (right) Approximation by the finite-capping-method respecting the smoothness to the outer patches (red bars) and interpolating the central point.

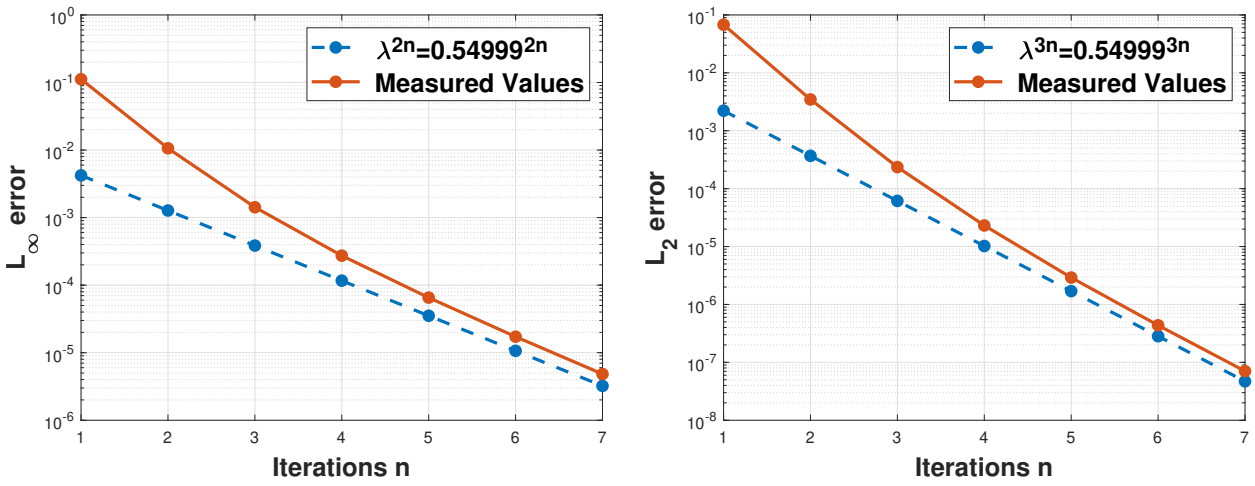



Figure 2: Measured $L_\infty$-error (left) and $L_2$-error (right) of an elliptic PDE on a pentagon in $\mathbb{R}^2$.

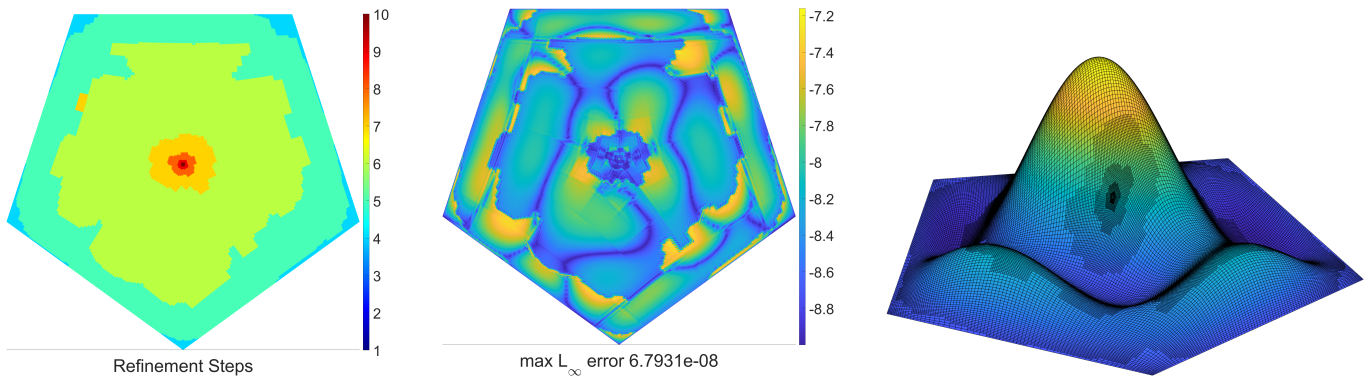



Figure 3: Solution of an elliptic PDE on a pentagon in $\mathbb{R}^2$ with an error tolerance of $10^{-6}$ regarding the $L_\infty$-error using adaptive refinement. (left) Refinement steps on each area. (middle) Measured Ł$_\infty$-error compared with the exact solution. (right) Plotted solution as third coordinate of the parameterization.

# An adapted interpolation method for generating smooth surfaces around extraordinary vertices of quadrilateral meshes

Byeongseon Jeong[a], Kyungmi Kim[b], Hyoseon Yang[c], Jungho Yoon[b]

[a] *Mathematics Major, Faculty of Basic Sciences, Keimyung University, Daegu 42601, S. Korea*
[b] *Department of Mathematics, Ewha Womans University, Seoul 03760, S. Korea*
[c] *Department of Mathematics, Kyung Hee University, Seoul 02447, S. Korea*

**Abstract**

In this study, we introduce a novel method for generating smooth surfaces near extraordinary vertices of initial quadrilateral meshes. To achieve this, we take the following two steps. First, from the initially given control points, we approximate derivatives of orders 1 and 2 at the vertices nearby extraordinary points by modifying the moving least-squares method [4]. We then apply a two-dimensional Hermite interpolatory subdivision rule to the values in the vicinity of the extraordinary vertices to generate smooth surfaces; see Figure 1. To construct our Hermite interpolation, by taking the directional derivatives along the local axis vectors into account, we modify the Hermite subdivision mask in [2] which is derived from the univariate quadratic spline interpolation for $f$, $f_x$, and $f_{xx}$. In particular, the tensor-product scheme of the Deslauriers-Dubuc four-point subdivision [1] is applied to the control points away from extraordinary vertices, resulting in a $C^1$ surface. Some numerical examples are presented to demonstrate the effectiveness of the proposed method.

*Keywords:* Subdivision scheme, extraordinary vertex, Hermite interpolation, four-point scheme

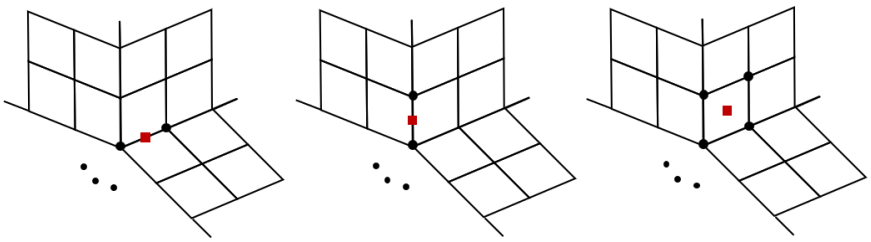



Figure 1: Stencils for new points (red squares) near the extraordinary vertex

## References

[1] G. Deslauriers and S. Dubuc, *Symmetric iterative interpolation processes*, Constr. Approx. **5**(1) (1989) 49–68.
[2] S. Dubuc, B. Han, J.-L. Merrien, Q. Mo, *Dyadic $C^2$ Hermite interpolation on a square mesh*, Comput. Aided Geom. Des. **22** (2005) 727–752.
[3] S. Dubuc and J.-L. Merrien, *Dyadic Hermite interpolation on a rectangular mesh*, Adv. Comput. Math. **10** (1999) 343–365.
[4] D. Levin, *The approximation power of moving least-squares*, Math. Comput. **67**(224) (1998) 1517–1531.
[5] P. Novara and L. Romani, *On extraordinary rules of quad-based interpolatory subdivision schemes*, Comput. Aided Geom. Des. **35-36** (2015) 225–242.

*Email addresses:* jeong@kmu.ac.kr (Byeongseon Jeong), ggoggochoco@ewhain.net (Kyungmi Kim), hyoseon@khu.ac.kr (Hyoseon Yang), yoon@ewha.ac.kr (Jungho Yoon)