UiO : University of Oslo

Application of blending splines in interactive geometric modeling

Supervisor
Prof. Arne Lakså, UiT - Narvik

Co-supervisor
Prof. Knut Mørken, UiO

Jostein Bratlie
UiT The Arctic University of Norway
R&D group Simulations - Narvik

2021-06-11

Outline

- The PhD project
- Blending Spline Constructions
- Applications in interactive geometric modeling
  - Papers I and II
- Contributions to Blending Splines
  - Papers III and V
- Prototyping of Differential Geometry
  - Paper IV

The PhD Project

- Part of the NRC Verdict "Dreamworld Project", (project no. 201511)
  - Initial partners NuC (now UiT Narvik - R&D Simulations) and Funcom
  - UiO through NuC as a doctoral program provider
  - Project goals: social gaming, easy access, seamless world, and **data representation and reduction**

- PhD Research Objectives
  - Explore the **blending spline constructions** and their unique properties,
    - within **Interactivity Geometric Modeling**, and
    - for **Animation** related **purposes**.

Blending Spline Constructions

Blending spline constructions (GERBS : ERBS/LERBS)

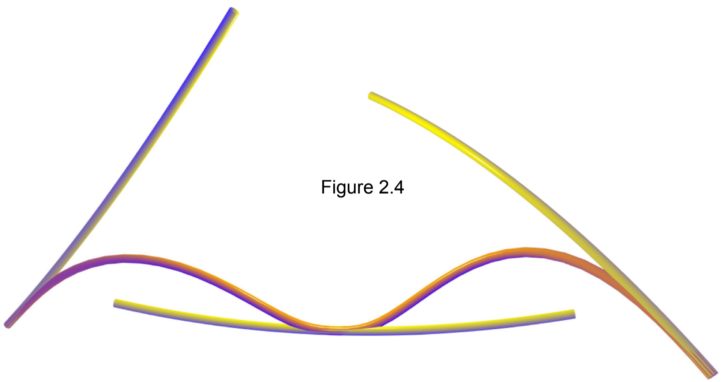
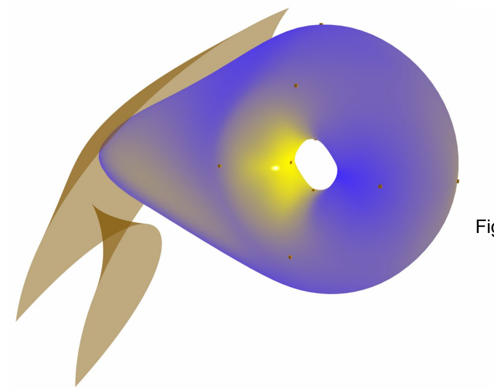


Figure 2.4



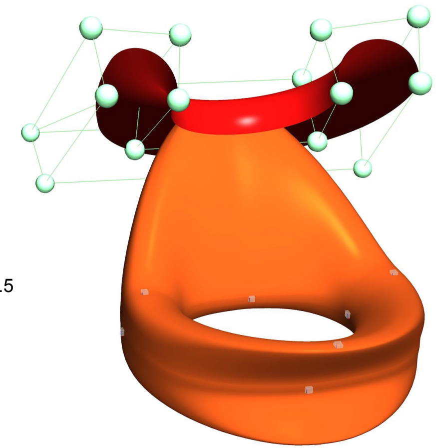


Figure 2.5

$$C(u) = \sum_{i=1}^{n} \bar{C}_i(u)\ B_i(u)$$

$$S(u,v) = \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} \bar{L}_{i,j}(u,v)\ B_j(v)\ B_i(u)$$

[DBL09] "Generalized Expo-Rational B-Splines", Dechevsky, Bang, Lakså, IJPAM, 2009
[LBD05] "Exploring Expo-Rational B-splines for Curves and Surfaces", Lakså, Bang, Dechevsky, 2005

Blending spline constructions - basis function

Blending spline basis

$$B_{d,k}(t) = \mathfrak{B} \circ w_{d,k}(t) B_{d-1,k}(t) + (1 - \mathfrak{B} \circ w_{d,k+1}(t)) B_{d-1,k+1}(t)$$

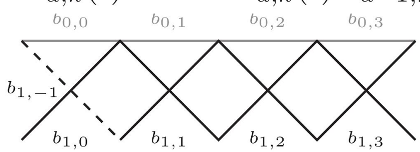
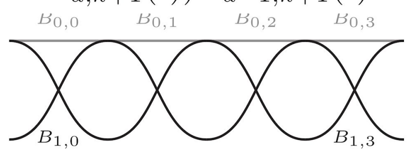


Figure 2.3



B-function

P1: B(t) : [0, 1] => [0, 1],
P2: B(t) has fixed end points,
P3: is continuous and monotone,
P4: is point-symmetric, and
P5: has an Hermite order; S ≥ 0

Basis/spline naming

GERBS: d = 1

ERBS: d = 1, B = ERB

LERBS: d = 1, B = LERB

($\mathcal{F}$)ERBS: d = 1, B = Fab ...

ERB

$$\mathfrak{B}(t) = 1.6571 \int_0^t \exp\left(-\frac{(t-\frac{1}{t})^2}{t(1-t)}\right)\ dt.$$

LERB

$$\mathfrak{B}(t) = \frac{1}{1+\exp\left(\frac{1}{t} - \frac{1}{1-t}\right)}$$

Fab eq. 5

$$\mathfrak{B}(t) = \frac{1}{2} - (\frac{9}{16}\cos(\pi t) - \frac{1}{16}\cos(3\pi t))$$

Fab eq. 21

$$\mathfrak{B}(t) = t - \frac{\sin(2\pi t)}{2\pi}$$

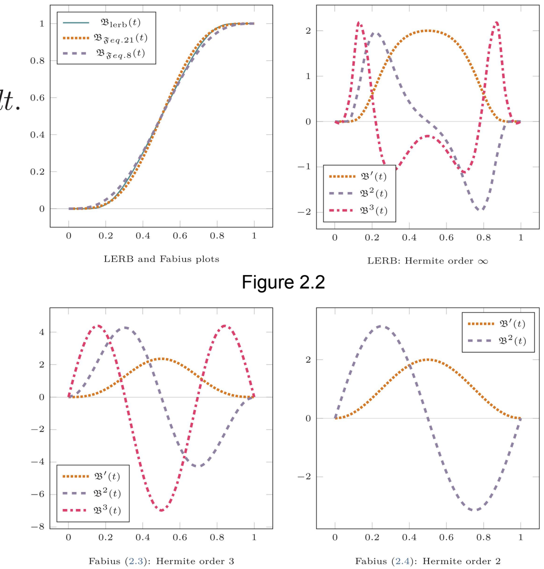


Figure 2.2

[DZ13] "Smooth GERBS, orthogonal systems and energy minimization", Dechevsky, Zanaty, 2013
[Olo19] "Blending functions based on trigonometric and polynomial approximations of the Fabius function", Olofsen, 2019

Blending spline constructions - organization

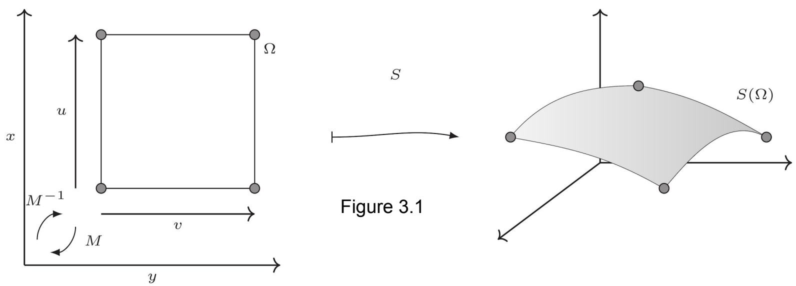


Figure 3.1

Two-function blending, [Lak13]

$$f(t) = \sum_{i=1}^{n} \ell_i(t) B_{d=1,i}(t) \qquad f(t) = (1 - \mathfrak{B}(t))\, \ell_1(t) + \mathfrak{B}(t)\, \ell_2(t)$$

$$= \ell_1(t) + (\ell_2(t) - \ell_1(t))\, \mathfrak{B}(t)$$

$$= \sum_{i=1}^{n} \ell_i(t) B_i(t)$$

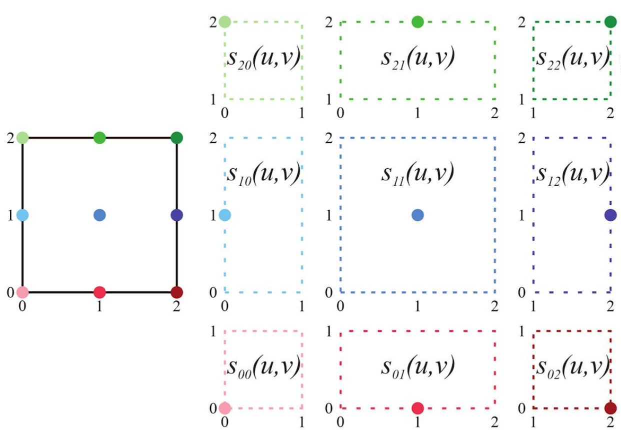
General derivation formulae

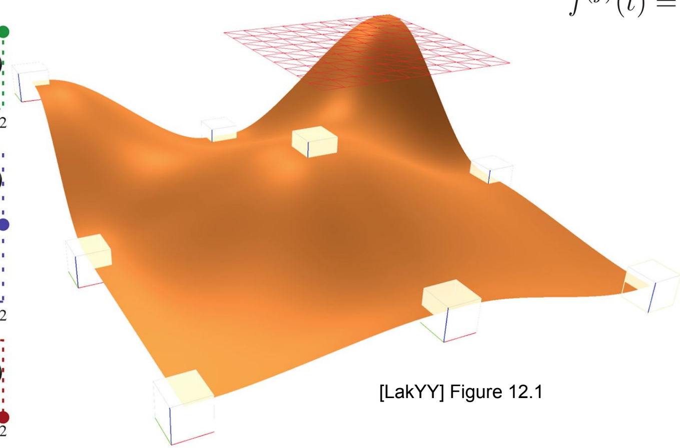
$$f^{(j)}(t) = \ell_1^{(j)}(t) + \sum_{i=0}^{j} \binom{j}{i} \mathfrak{B}^{(i)}(t) (\ell_2(t) - \ell_1(t))^{(j-i)}(t)$$

Vanishing derivatives

$$f^{(j)}(0) = \ell_1^{(j)}(0), \quad j = 0, 1, \dots, S_1$$

$$f^{(j)}(1) = \ell_2^{(j)}(1), \quad j = 0, 1, \dots, S_2$$

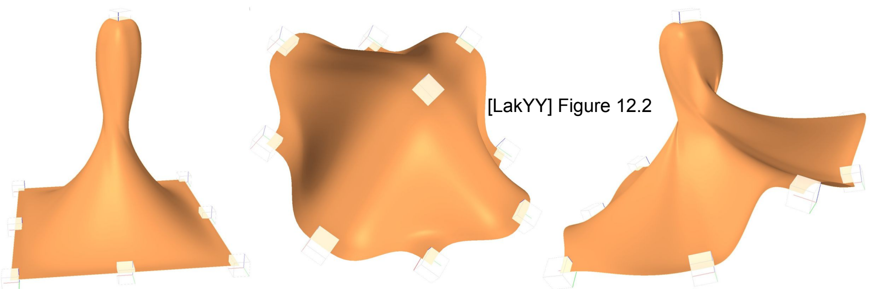




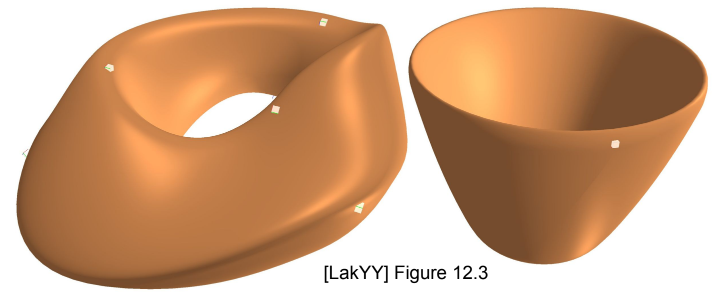
[LakYY] Figure 12.1

[Lak13] “ERBS-surface construction on irregular grids”, Lakså, 2013
[LakYY] “Bending techniques in Curve and Surface constructions”, Arne Lakså, Unpublished.

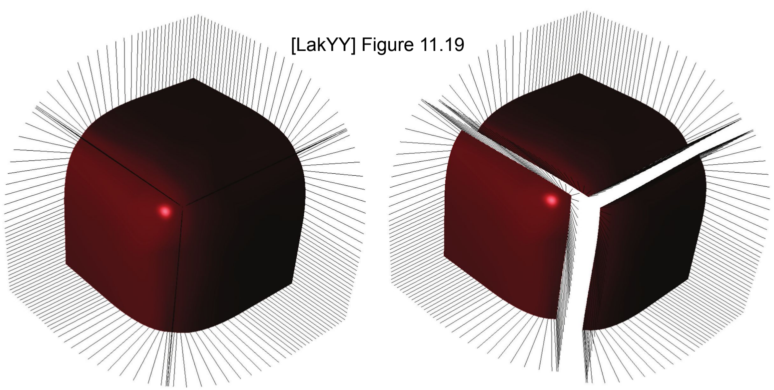
Blending spline constructions - free forming shapes



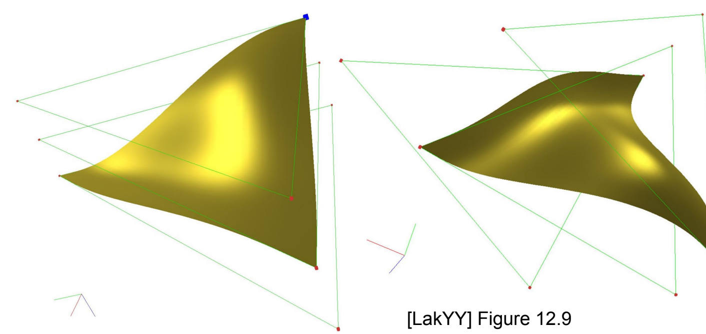
[LakYY] Figure 12.2



[LakYY] Figure 12.3



[LakYY] Figure 11.19



[LakYY] Figure 12.9

[LakYY] "Bending techniques in Curve and Surface constructions", Arne Lakså, Unpublished.

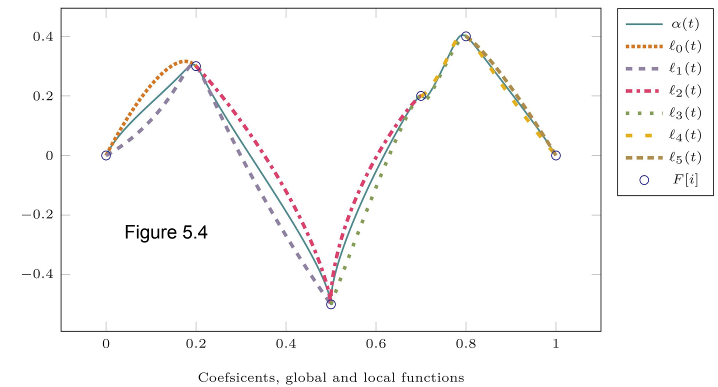
Papers I and II

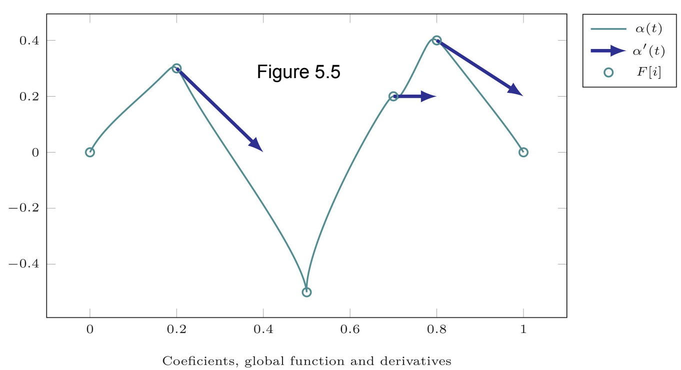
Applications in interactive geometric modeling

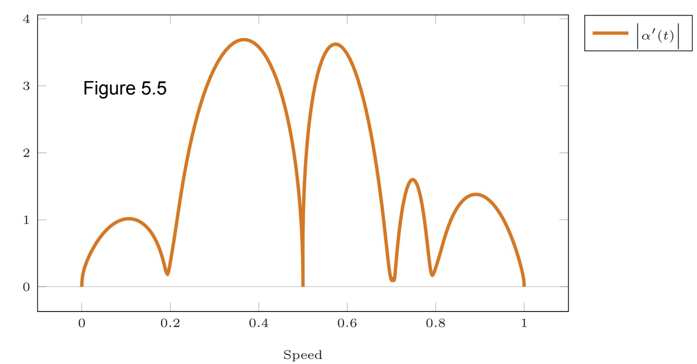
Applications in IGM : keyframing and speed control

Key properties

- Interpolates each local coefficient
- Local has embedding
- Vanishing derivatives
- C^k-smooth while G^0-smooth







Applications in IGM : skinning

Key properties

- Interpolates each local coefficient
- Local has embedding
- Vanishing derivatives
- C^k-smooth while G^0-smooth

Skinning concept

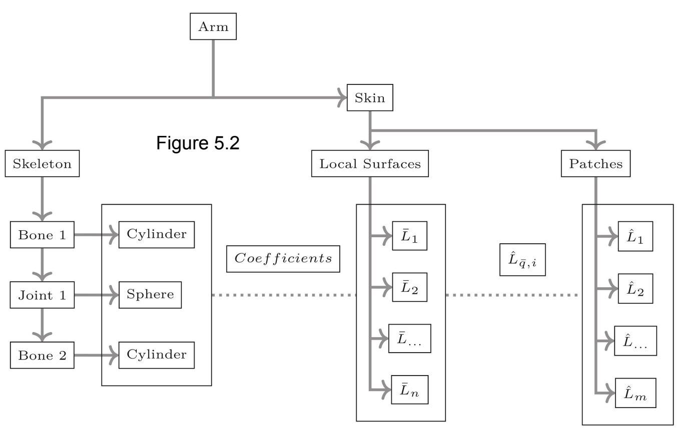


Figure 5.2

Skinning without volumetrics, [HBD14]

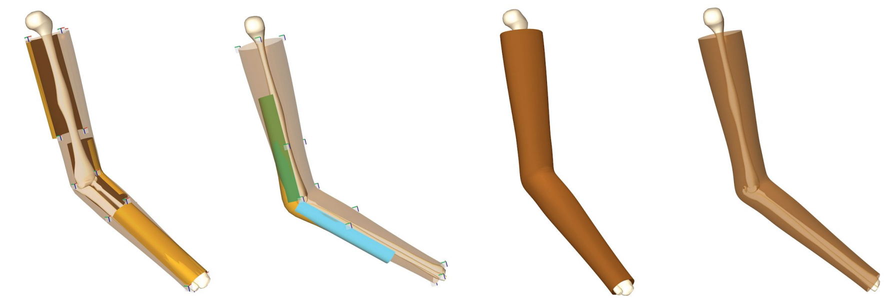
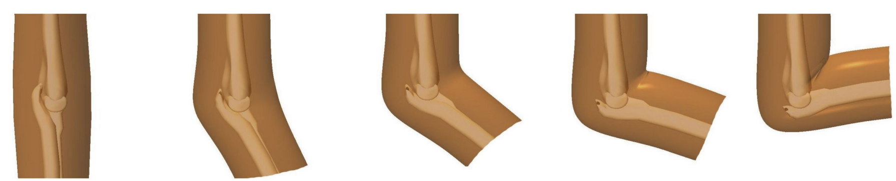


Figure 5.3
&
[BHD14]



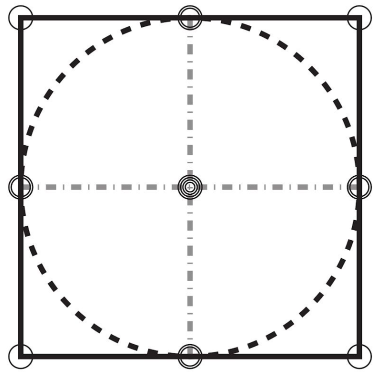
[HBD14] “Surface deformation over flexible joints using spline blending techniques”, Haavardsholm, Bratlie, Dalmo, 2014

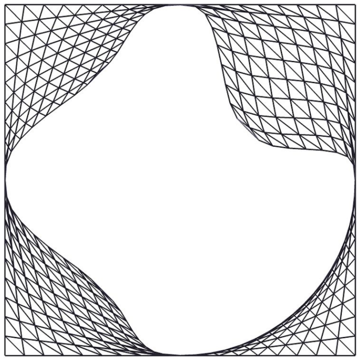
Applications in IGM : warping

Key properties

- Interpolates each local coefficient
- Local has embedding
- Vanishing derivatives
- C^k-smooth while G^0-smooth

Holeshaping using custom locals and double knots, [PBD15]





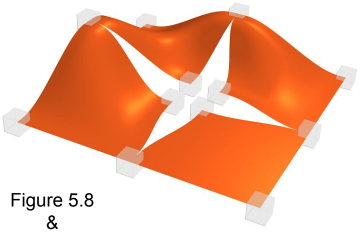


Figure 5.8
&
[PBD15]

[PBD15] “Spline representation of connected surfaces with custom-shaped holes”, Pedersen, Bratlie, Dalmo, 2015

Paper I, [Bra13]

Local refinement of GERBS surfaces

[Bra13] “Local refinement of GERBS surfaces with applications to interactive geometric modeling”, Bratlie, 2013

Local refinement of GERBS surfaces, Paper I, [Bra13]

New local on knot insertion (curve)

$$\hat{\ell}(t) = \begin{cases} \ell_k(t) + (\ell_{k+1}(t) - \ell_k(t))\ \frac{\mathfrak{B}\circ\omega_k(t)}{\mathfrak{B}\circ\omega_1(t)} & \text{if } t \in (t_k, \hat{t}) \\ \ell_k(t) + (\ell_{k+1}(t) - \ell_k(t))\ \frac{\mathfrak{B}\circ\omega_k(t) - \mathfrak{B}\circ\omega_2(t)}{1 - \mathfrak{B}\circ\omega_2(t)} & \text{if } t \in (\hat{t}, t_{k+1}) \end{cases}$$

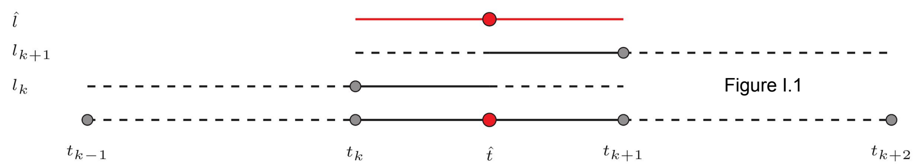


Figure I.1

Motivation

- Knot insertion => rational local
- Preserve geometry on refinement
- Insert modeling friendly local

Refinement by blending

- Two schemes:
  - Multi-knot and multi-level

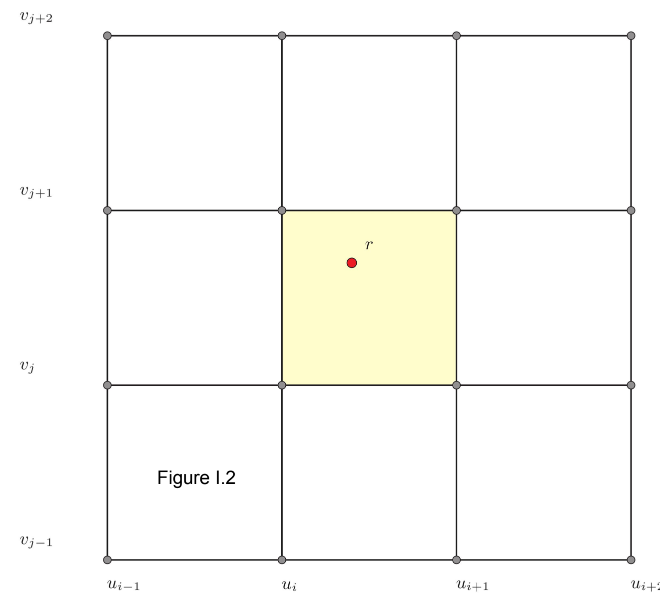


Figure I.2

[Bra13] “Local refinement of GERBS surfaces with applications to interactive geometric modeling”, Bratlie, 2013

Local refinement of GERBS surfaces, Paper I, [Bra13]

Knot based refinement

- Refinement knots have an order

$S_n = S_{n-1} + (S_{r,n} - S_{n-1})\, A \circ \lambda(r_n),$
$S_0 = G + (S_{r,0} - G)\, A \circ \lambda(r_0)$

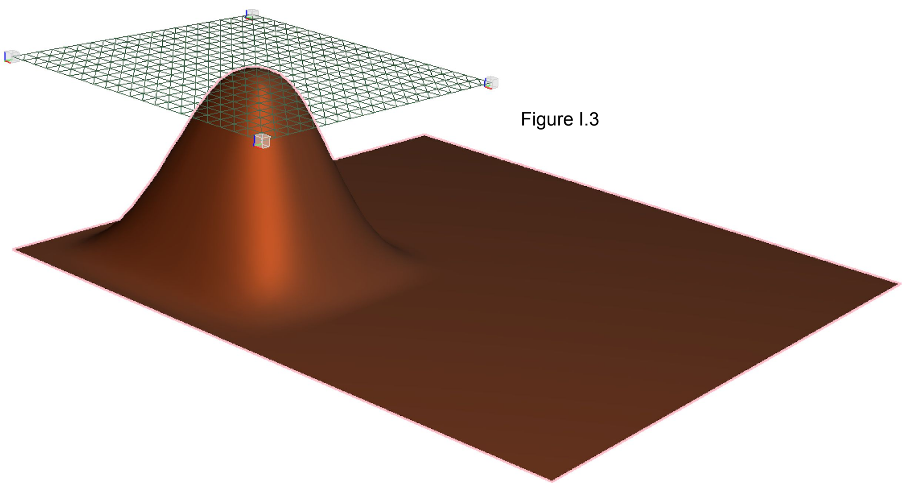


Figure I.3

Level based refinement

- Refinement patch refined

$Sn = G_n + (S_{n-1} - G_n)\, A \circ \lambda(\hat{r}_n),$
$S0 = G_0 + (S_{r,0} - G_0)\, A \circ \lambda(\hat{r}_0)$

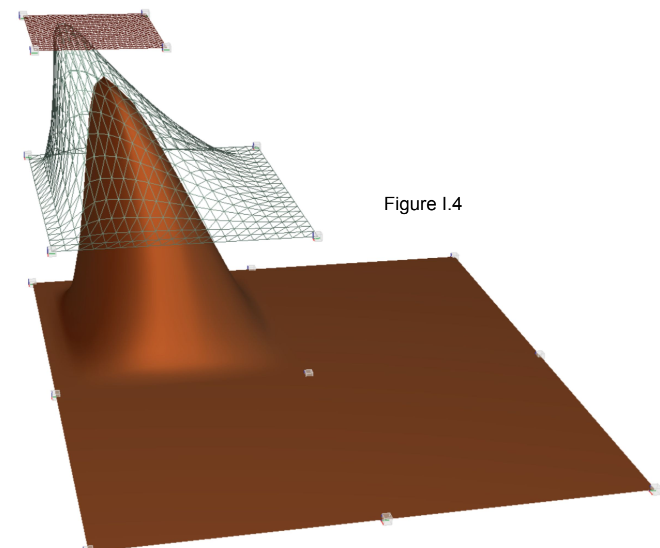


Figure I.4

[Bra13] "Local refinement of GERBS surfaces with applications to interactive geometric modeling", Bratlie, 2013

Paper II, [BDZ14]

Fitting of discrete data with GERBS

[BDZ14] “Fitting of Discrete Data with GERBS”, Bratlie, Dalmo, Zanaty, 2014

Fitting of discrete data with GERBS, Paper II, [BDZ14]

Motivation

- Reuse coefficients in locals coefficients
- Applications to model and animation data

Select feature points

- Curvature base partitioning
- Inflexion base partitioning

Inflexion base partitioning

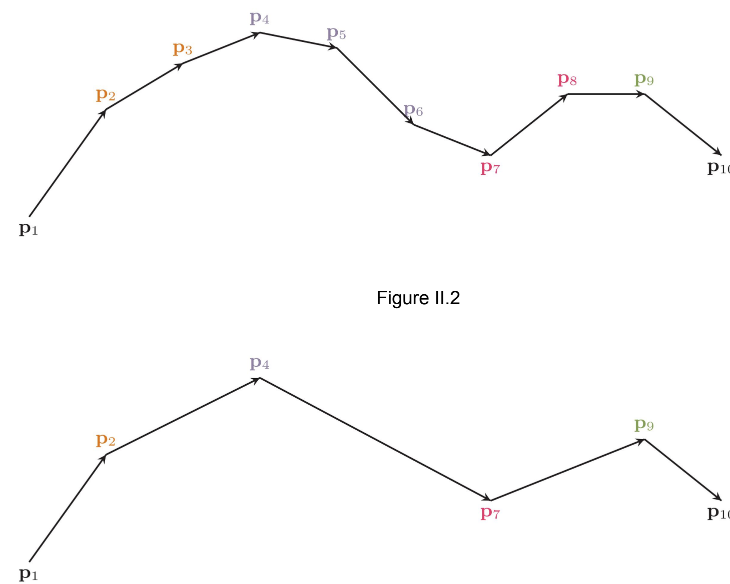


Figure II.2

Benchmark setup

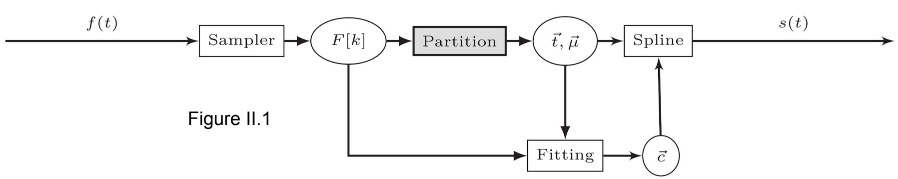


Figure II.1

[BDZ14] "Fitting of Discrete Data with GERBS", Bratlie, Dalmo, Zanaty, 2014

Fitting of discrete data with GERBS, Paper II, [BDZ14]

Smooth benchmark

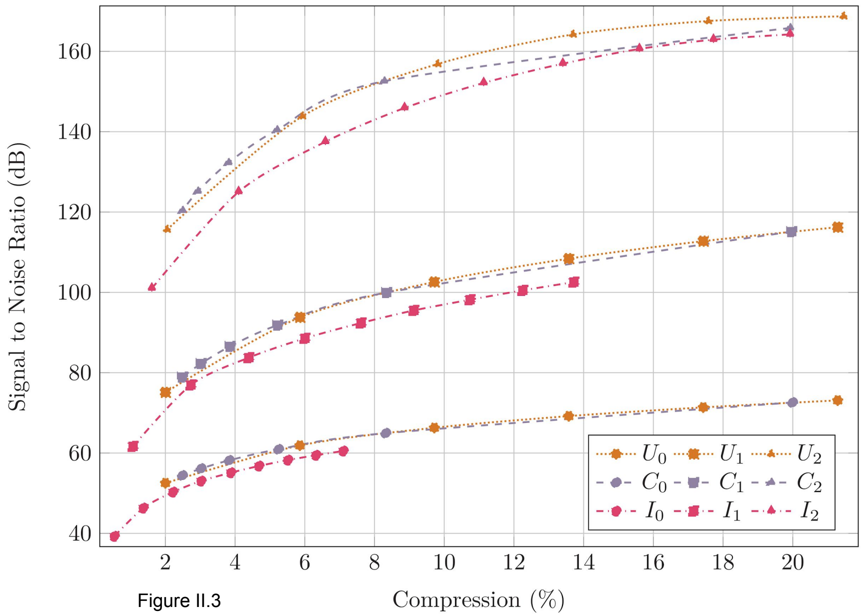


Figure II.3

Oscillating benchmark

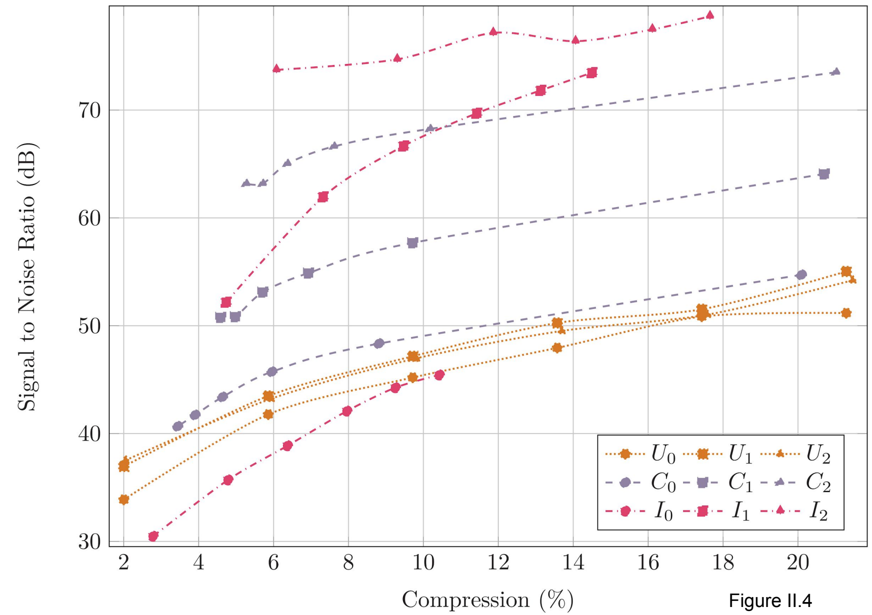


Figure II.4

[BDZ14] “Fitting of Discrete Data with GERBS”, Bratlie, Dalmo, Zanaty, 2014

Papers III and V

Contributions to Blending Splines

Paper III, [BDB15]

GPU evaluation of blending spline surfaces

[BDB15], “Evaluation of smooth spline blending surfaces using GPU”, Bratlie, Dalmo, Bang, 2015

GPU evaluation of blending spline surfaces, Paper III, [BDB15]

Motivation

- Evaluate blending splines on GPU
- Hierarchical, complex, not suited

However

- Patch primitive = BS eval patch

$$\hat{S}(u,v) = \sum_{i=1}^{2}\sum_{i=1}^{2} \bar{L}_{i,j} \circ \omega_{i,j}(u,v)\, \mathfrak{B}_j(v)\, \mathfrak{B}_i(u)$$
$$= \sum_{i=1}^{2}\sum_{i=1}^{2} \hat{L}_{i,j}(u,v)\, \mathfrak{B}_j(v)\, \mathfrak{B}_i(u)$$

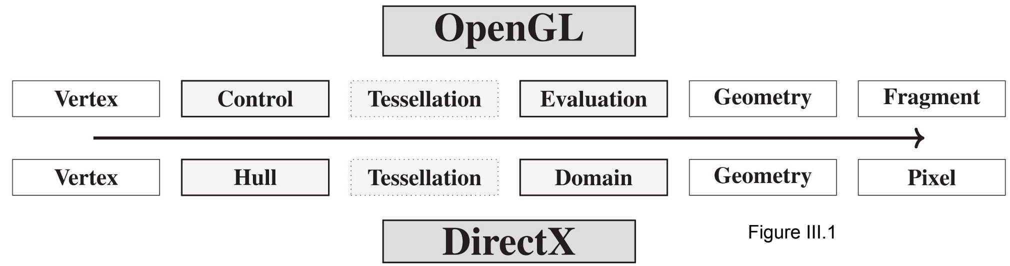


Figure III.1

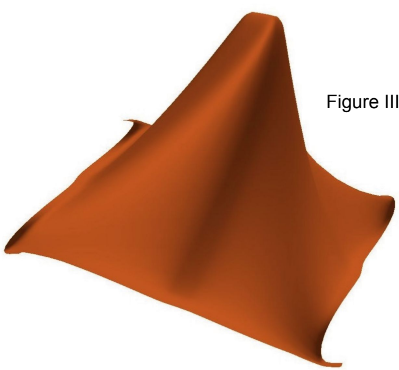


Figure III.4

Edge requirement

- Equal tessellation factor along adjacent the edges

Example

- 3x3 local surface
- 4 evaluation patches

[BDB15], “Evaluation of smooth spline blending surfaces using GPU”, Bratlie, Dalmo, Bang, 2015
[Lak14] “Construction and properties of non-polynomial spline curves”, Lakså, 2014

GPU evaluation of blending spline surfaces, Paper III, [BDB15]

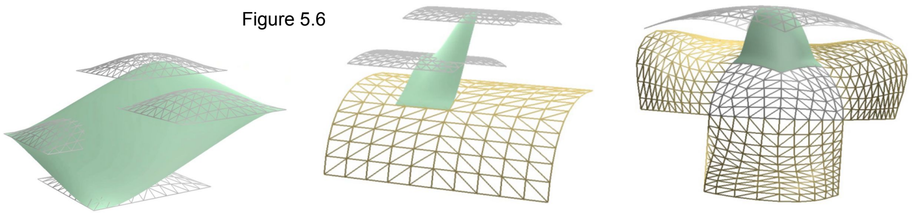


Figure 5.6

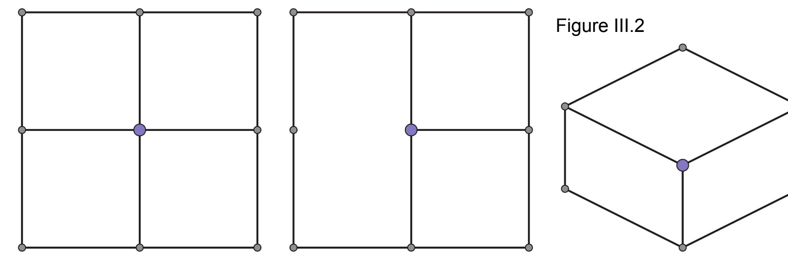


Figure III.2

Structure dictates two strategies

- Direct evaluation
- Local Pre-evaluation

Examples

- Over T- and Star-joints
- Produced for MMCS 2016, Tønsberg

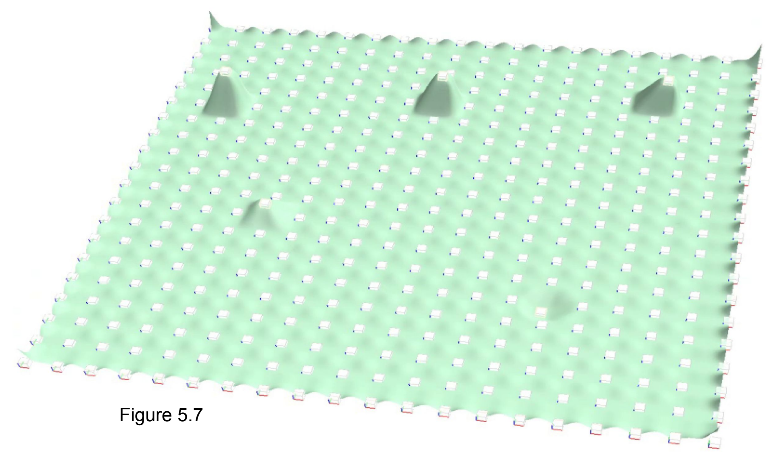


Figure 5.7

[BDB15], “Evaluation of smooth spline blending surfaces using GPU”, Bratlie, Dalmo, Bang, 2015
[Lak14] “Construction and properties of non-polynomial spline curves”, Laksâ, 2014

Paper V, [BD21]

Blending surfaces over polygonal mesh

[BD21] “Blending spline polygon surface over arbitrary poly-mesh topology”, Bratlie, Dalmo, Under revision

Blending surfaces over polygonal mesh, Paper V, [BD21]

Motivation

- Blending spline construction over polygonal mesh
- “Solved” the GPU evaluation issue
- Publication of Enhanced GB Patches, [VSK16]
- Using modeling friendly local geometry

Challenges

- BS construction inherently parametric
  - (control blend parameter direction across a edge)
- Control parameter direction across edge
- Topology
  - Define parametric domains
  - Knot vectors
- Looked good on paper, challenging to prototype

Topological

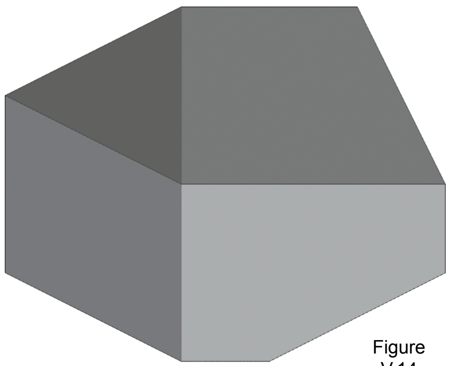


Figure V.14

Polygonal BS Surface

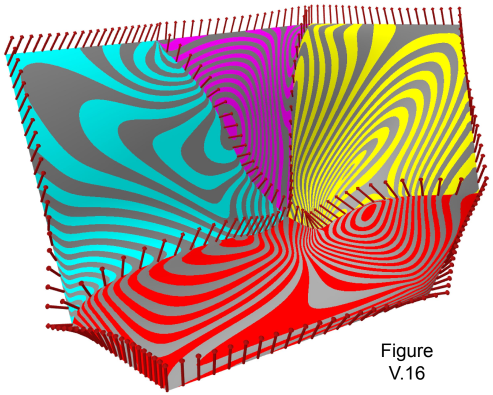


Figure V.16

[BD21] “Blending spline polygon surface over arbitrary poly-mesh topology”, Bratlie, Dalmo, Under revision
[VSK16] “A Multi-sided Bézier Patch with a Simple Control Structure”, Várady, Salvi, Karikó, 2016

Blending surfaces over polygonal mesh, Paper V, [BD21]

Polygonal BS Surface

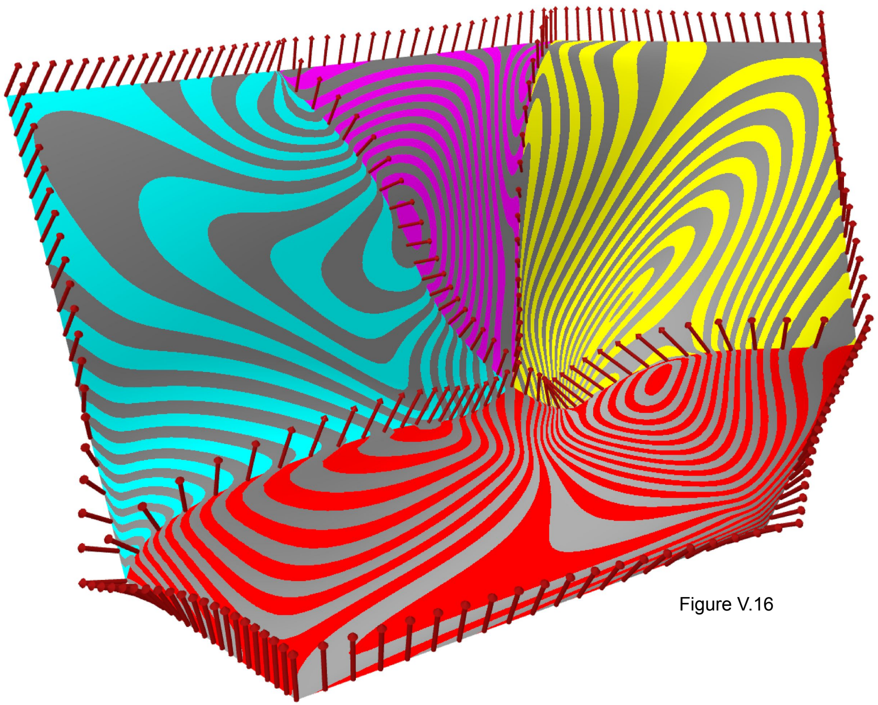


Figure V.16

Local Polygonal Surfaces

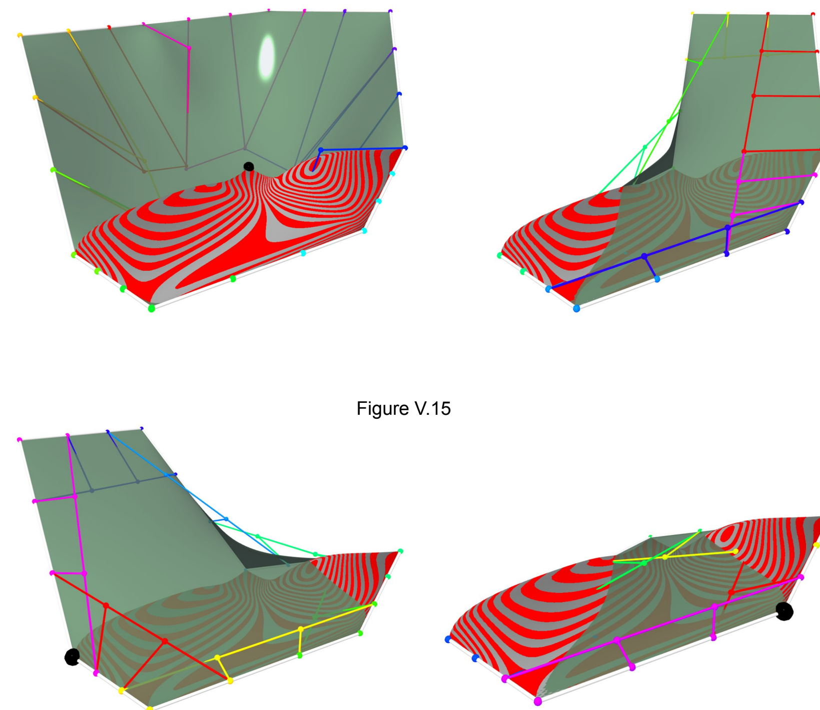
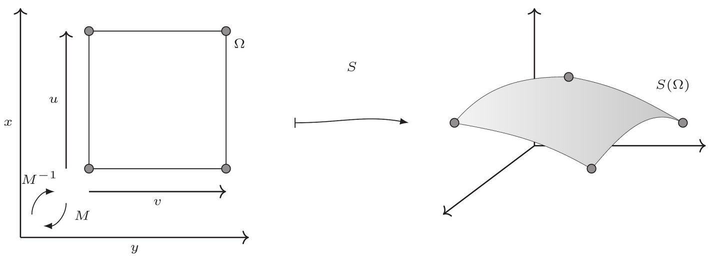


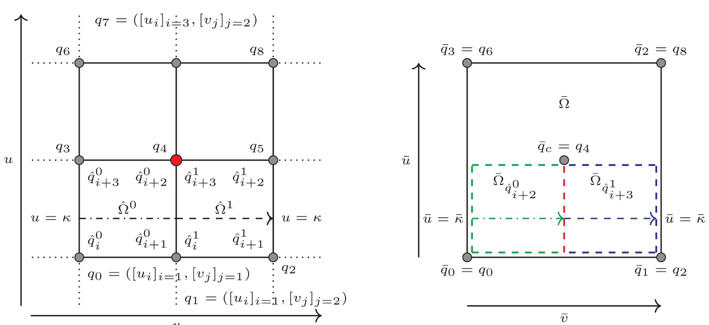
Figure V.15

[BD21] "Blending spline polygon surface over arbitrary poly-mesh topology", Bratlie, Dalmo, Under revision

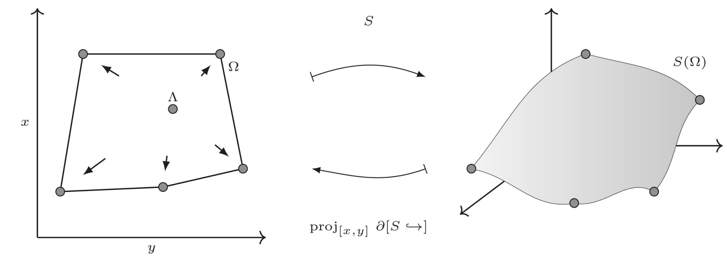
Blending surfaces over polygonal mesh, Paper V, [BD21]

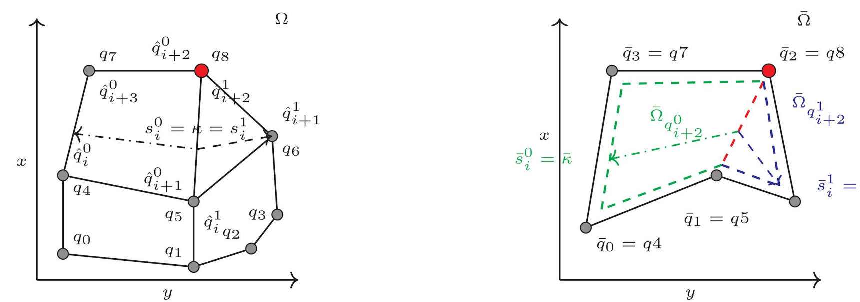
Tensor-product BS Domains





Polygonal BS Domains





[BD21] "Blending spline polygon surface over arbitrary poly-mesh topology", Bratlie, Dalmo, Under revision

Blending surfaces over polygonal mesh, Paper V, [BD21]

Polygonal BS Surface, S, of Patches {Ŝ, …}

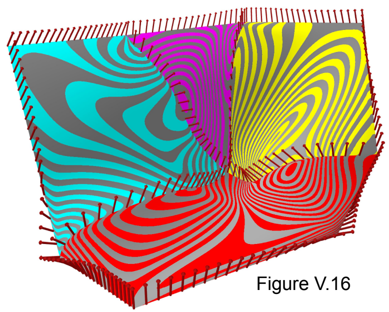


Figure V.16

Local Polygonal Surfaces, L, of Patch, Ŝ

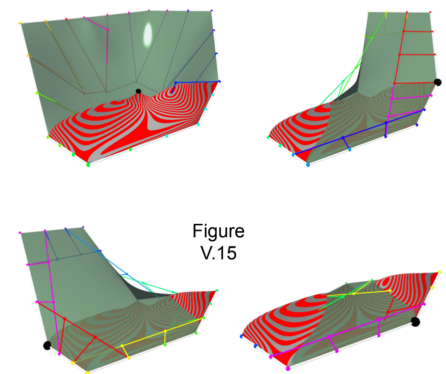
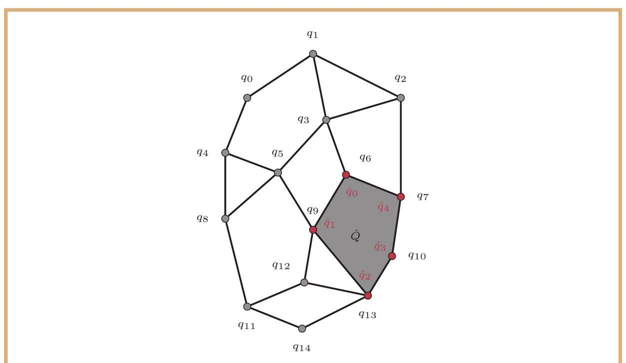
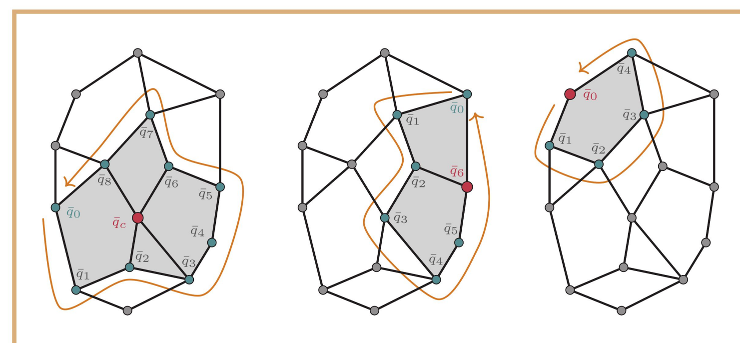


Figure V.15

Evaluation patch cover, Q



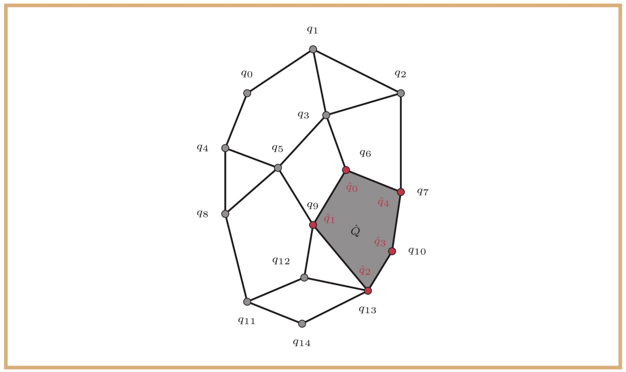
Local surface cover, Q



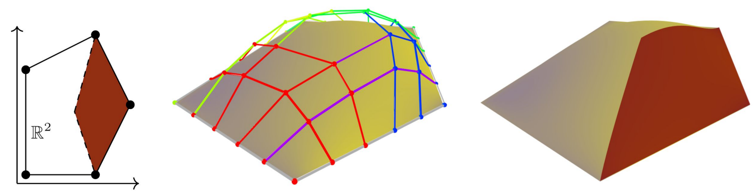
[BD21] "Blending spline polygon surface over arbitrary poly-mesh topology", Bratlie, Dalmo, Under revision

Blending surfaces over polygonal mesh, Paper V, [BD21]

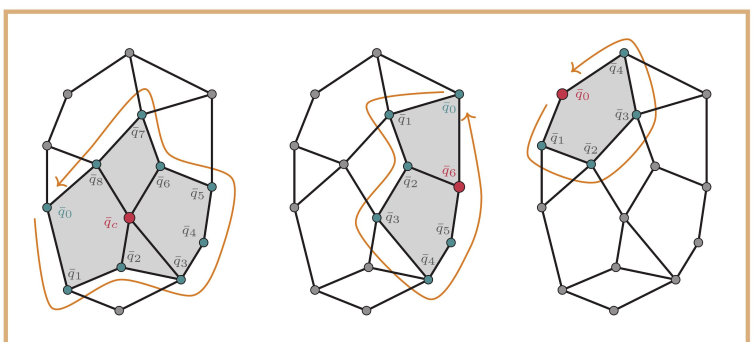
Evaluation patch cover, Q



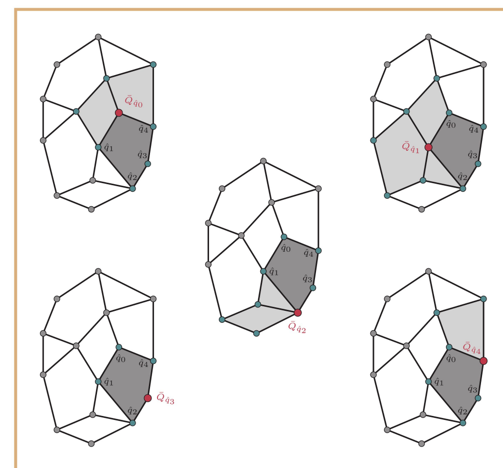
Local polygonal, L, and Sub-polygon, $\hat{L}$



Local surface cover, Q



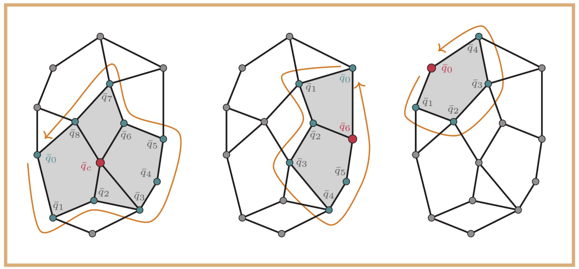
Local sub-surface covers, $Q_{\hat{q}}$



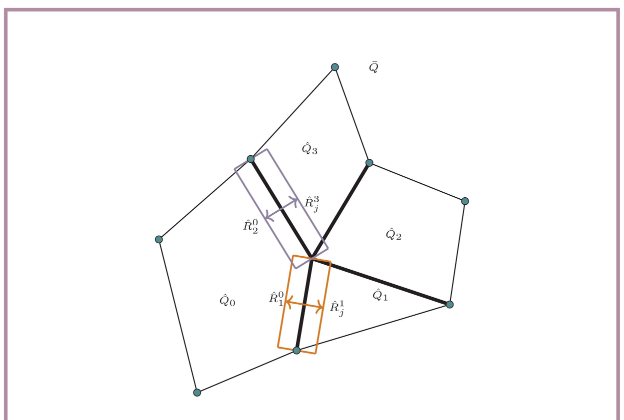
[BD21] "Blending spline polygon surface over arbitrary poly-mesh topology", Bratlie, Dalmo, Under revision

Blending surfaces over polygonal mesh, Paper V, [BD21]

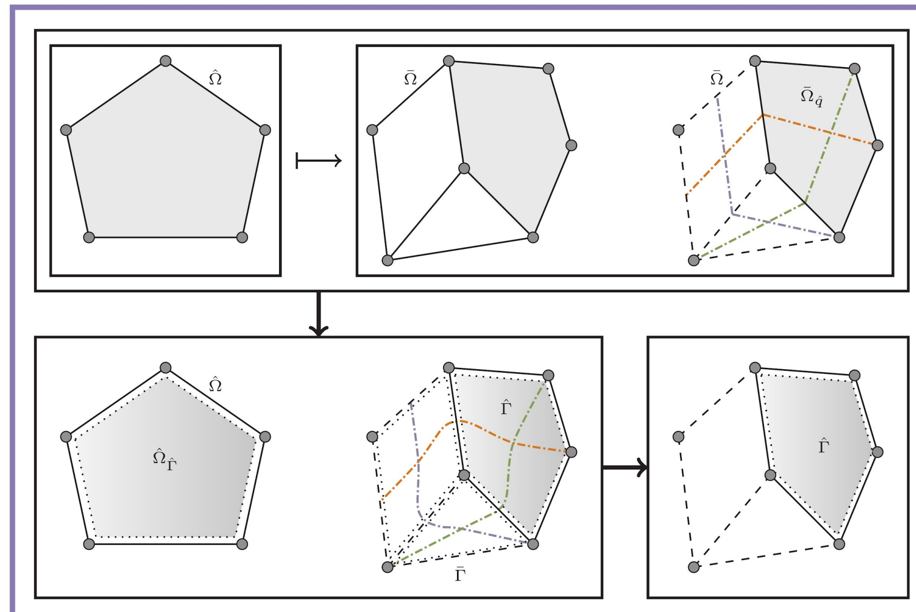
Local surface cover, Q



Local Domain Mapping Patchwork, Γ



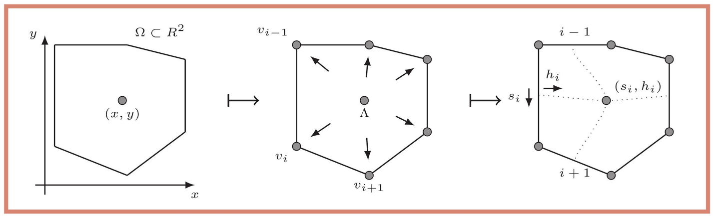
Reparameterization of Q to $Q_{\hat{q}}$ through Γ



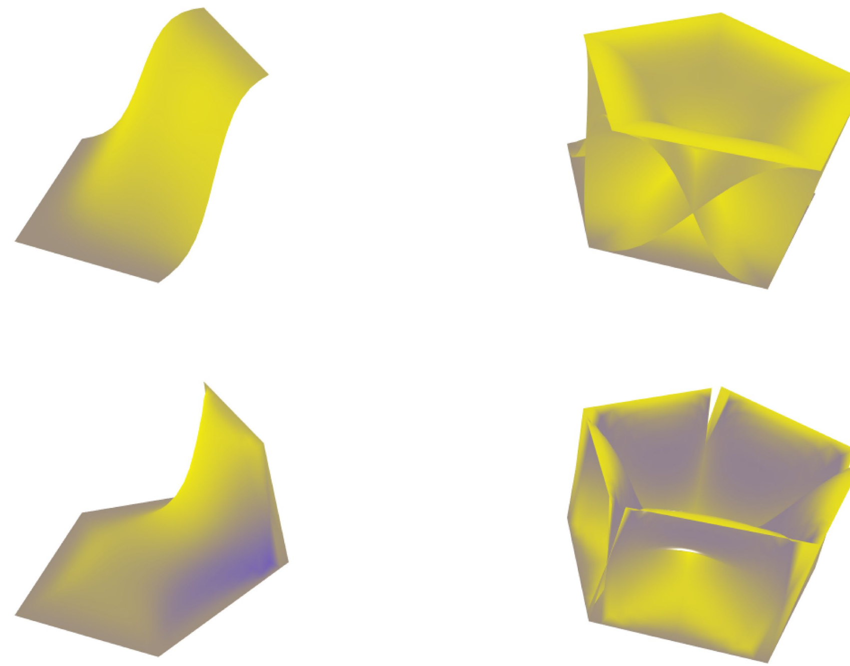
[BD21] “Blending spline polygon surface over arbitrary poly-mesh topology”, Bratlie, Dalmo, Under revision

Blending surfaces over polygonal mesh, Paper V, [BD21]

Side-based parameters, [SVR14]



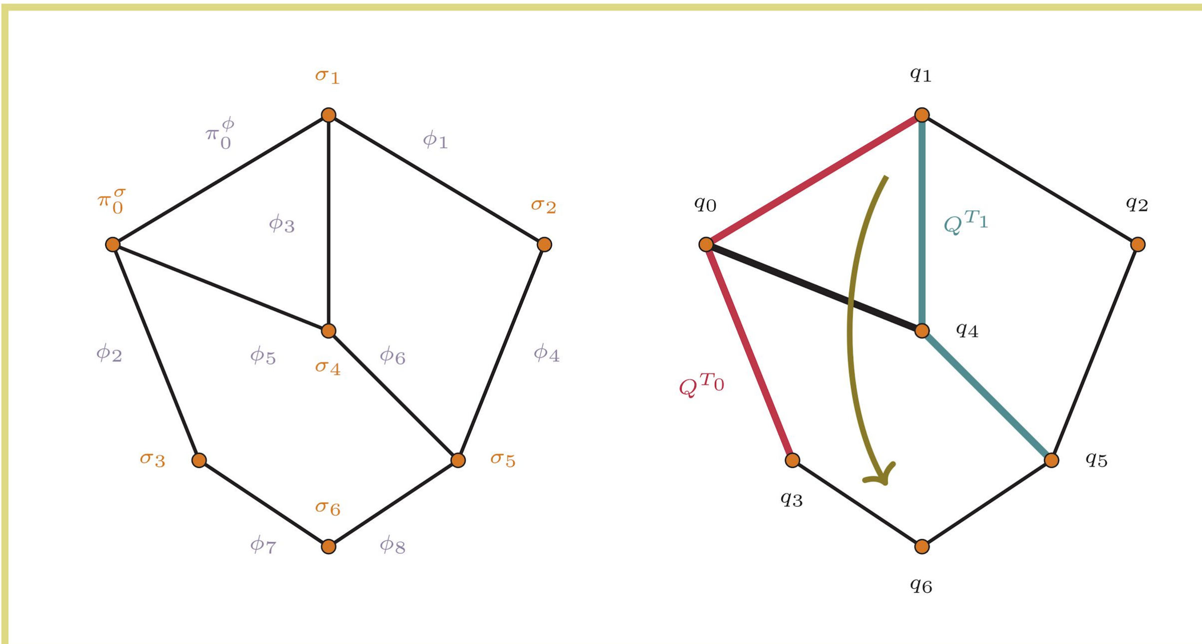
Blending basis functions for s- and h-direction.



[BD21] “Blending spline polygon surface over arbitrary poly-mesh topology”, Bratlie, Dalmo, Under revision
[SVR14] “Ribbon-based transfinite surfaces”, Salvi, Várady, Rockwood, 2014

Blending surfaces over polygonal mesh, Paper V, [BD21]

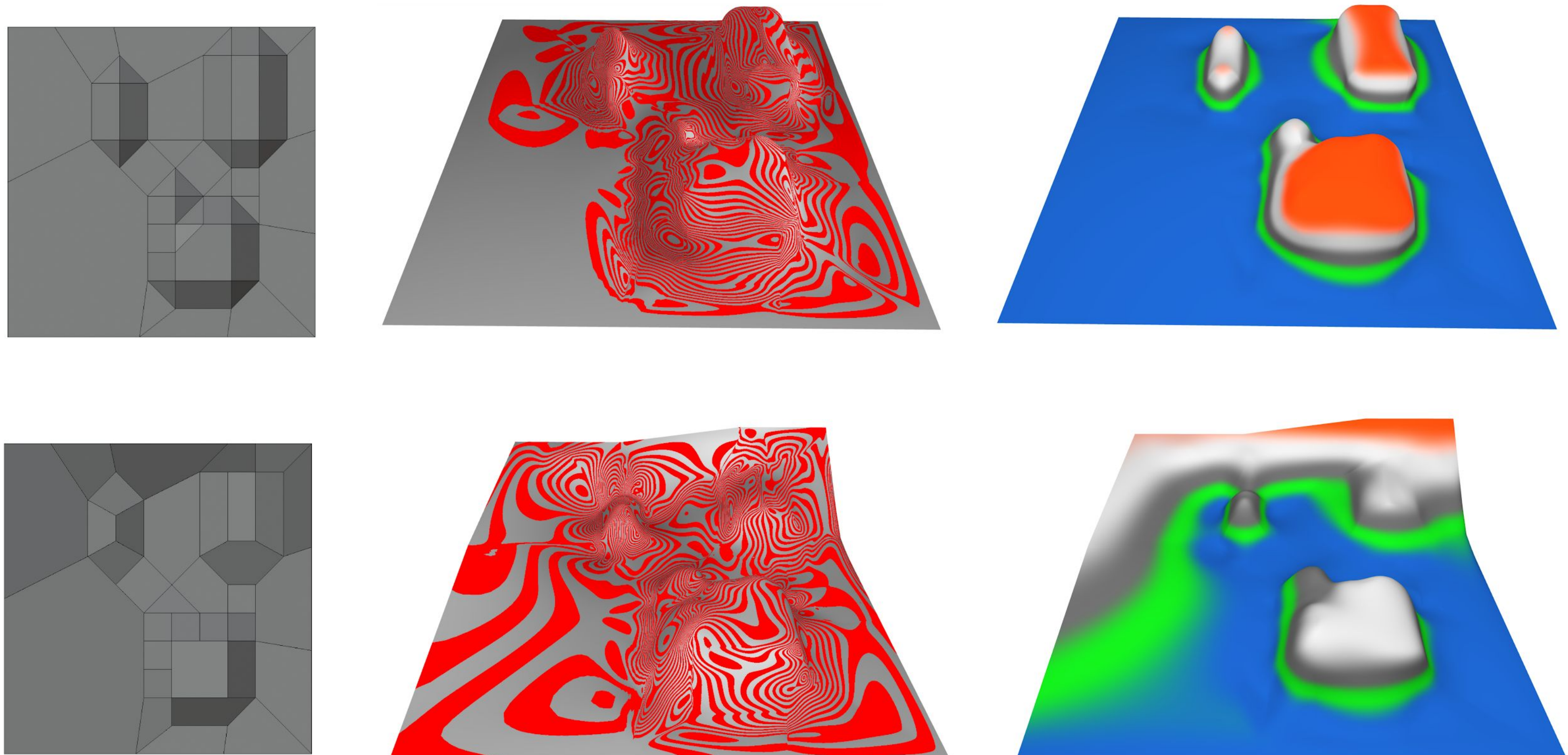
Knot vector from graph, Π - multiplicity π^σ, edge distance, π^φ



[BD21] "Blending spline polygon surface over arbitrary poly-mesh topology", Bratlie, Dalmo, Under revision
[SVR14] "Ribbon-based transfinite surfaces", Salvi, Várady, Rockwood, 2014

Blending surfaces over polygonal mesh, Paper V, [BD21]

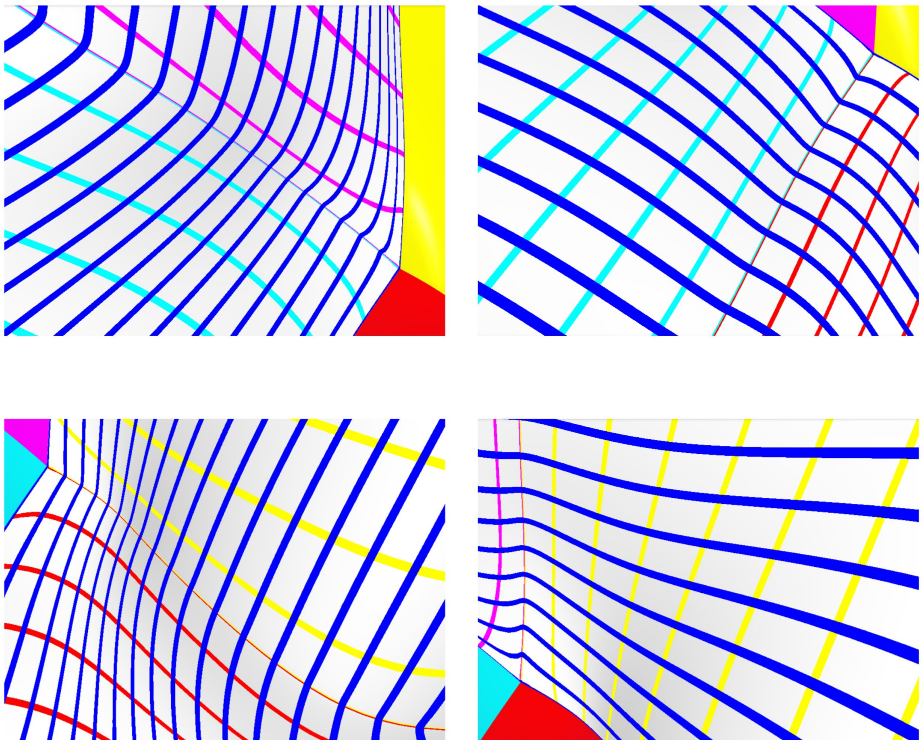
Interpolation of poly-mesh heightmap, with local approximation



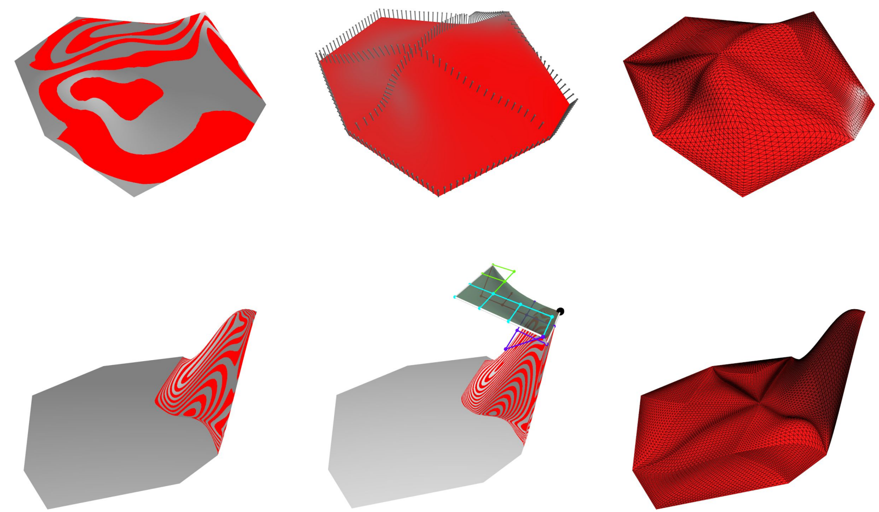
[BD21] “Blending spline polygon surface over arbitrary poly-mesh topology”, Bratlie, Dalmo, Under revision

Blending surfaces over polygonal mesh, Paper V, [BD21]

Isophote smoothness across edges



Isophote, edge normals and tessellation wireframes



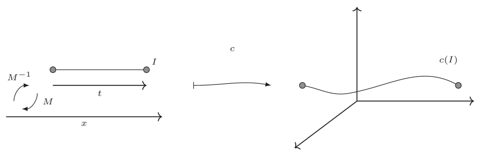
[BD21] “Blending spline polygon surface over arbitrary poly-mesh topology”, Bratlie, Dalmo, Under revision

Paper IV

Differential Geometry Prototyping

Differential Geometry Prototyping

Parametric curve



Parametric tensor-product surface

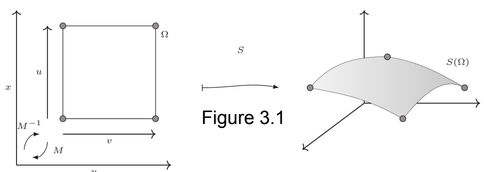


Figure 3.1

Parametric polygonal surface

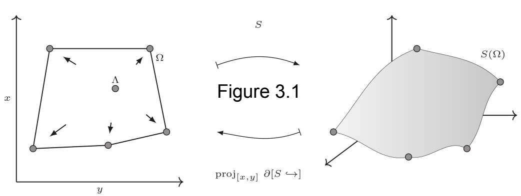


Figure 3.1

Sub-curve in **sub-surface** in a **coons patch**
of **sub-curves** in **curves**

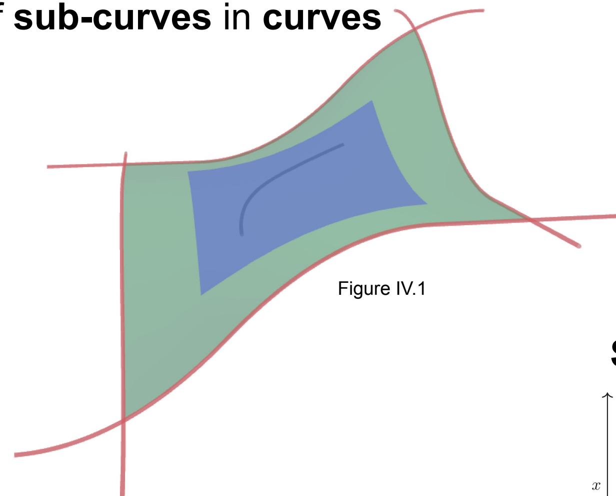


Figure IV.1

Parametric differentiable mapping

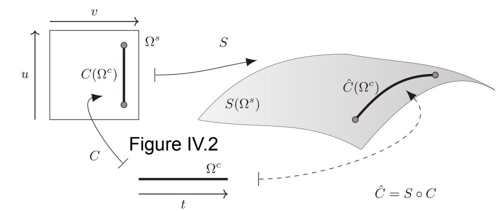


Figure IV.2

Sub-polygonal surface

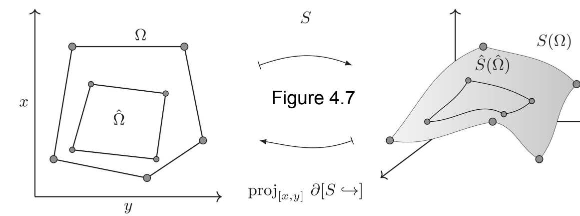


Figure 4.7

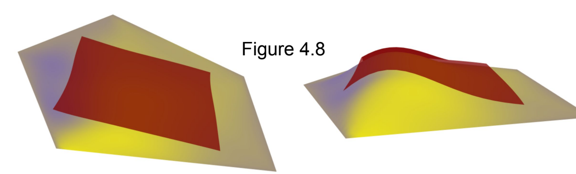


Figure 4.8

Paper IV, [BD19]

Prototyping geometric modeling: C++

[BD19] “Exploring future C++ features within a geometric modeling context”, Bratlie, Dalmo, 2019

Prototyping geometric modeling: C++, Paper IV, [BD19]

Motivation

- Aid Prototyping of Geometric Constructions

C++ (17)

- Generic Programming Mechanism: templates
  - Creates generic code fragment candidates
- Low overhead
  - Unused candidates removed at compile time
- High-Level Language with Low-level features
  - Functional/ OO / OC => … => assembly

Defining a set of idioms or techniques

- Non-intrusive inheritance
- Semantic compile-time (static) polymorphism
- Aggregated properties and transitive constructors

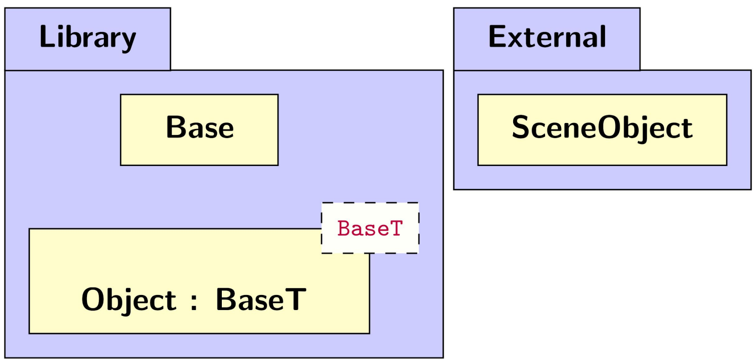
Explore future features (20++ and beyond)

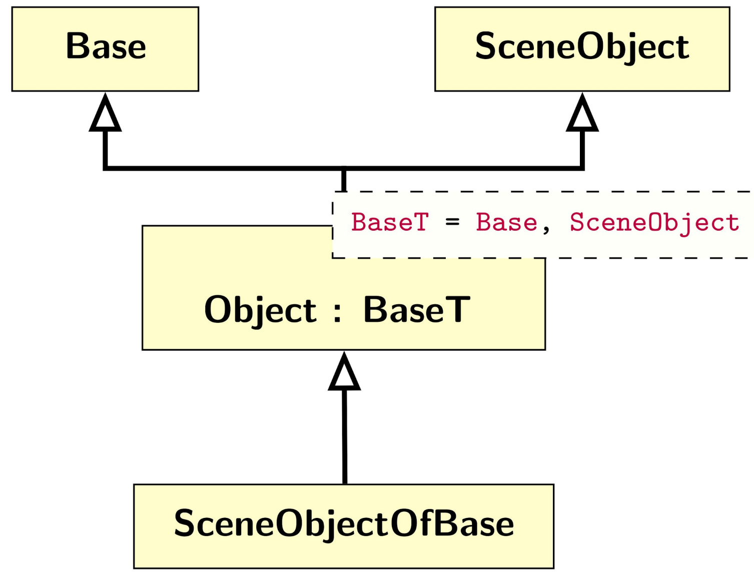
- Concepts, Contracts, Reflection and Meta-Classes

[BD19] “Exploring future C++ features within a geometric modeling context”, Bratlie, Dalmo, 2019

Prototyping geometric modeling: C++, Paper IV, [BD19]

Non-intrusive inheritance

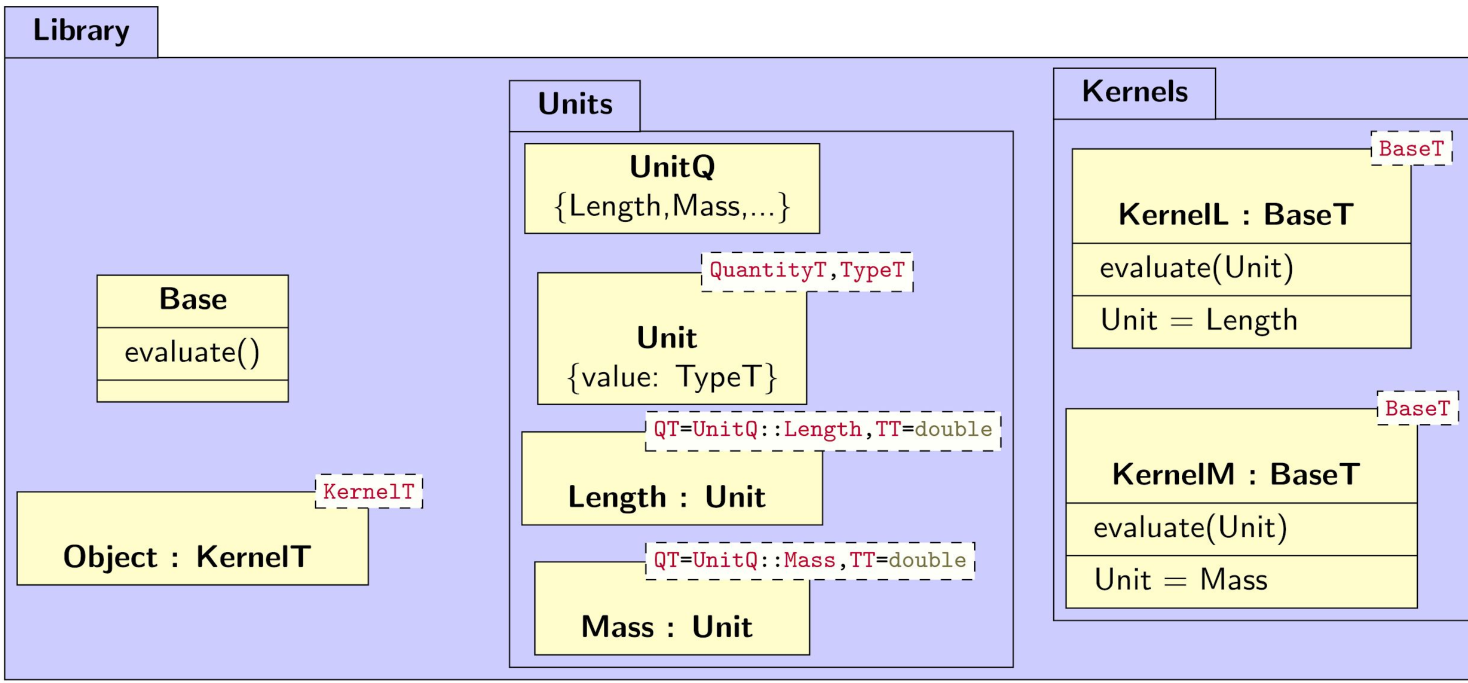




[BD19] “Exploring future C++ features within a geometric modeling context”, Bratlie, Dalmo, 2019

Prototyping geometric modeling: C++, Paper IV, [BD19]

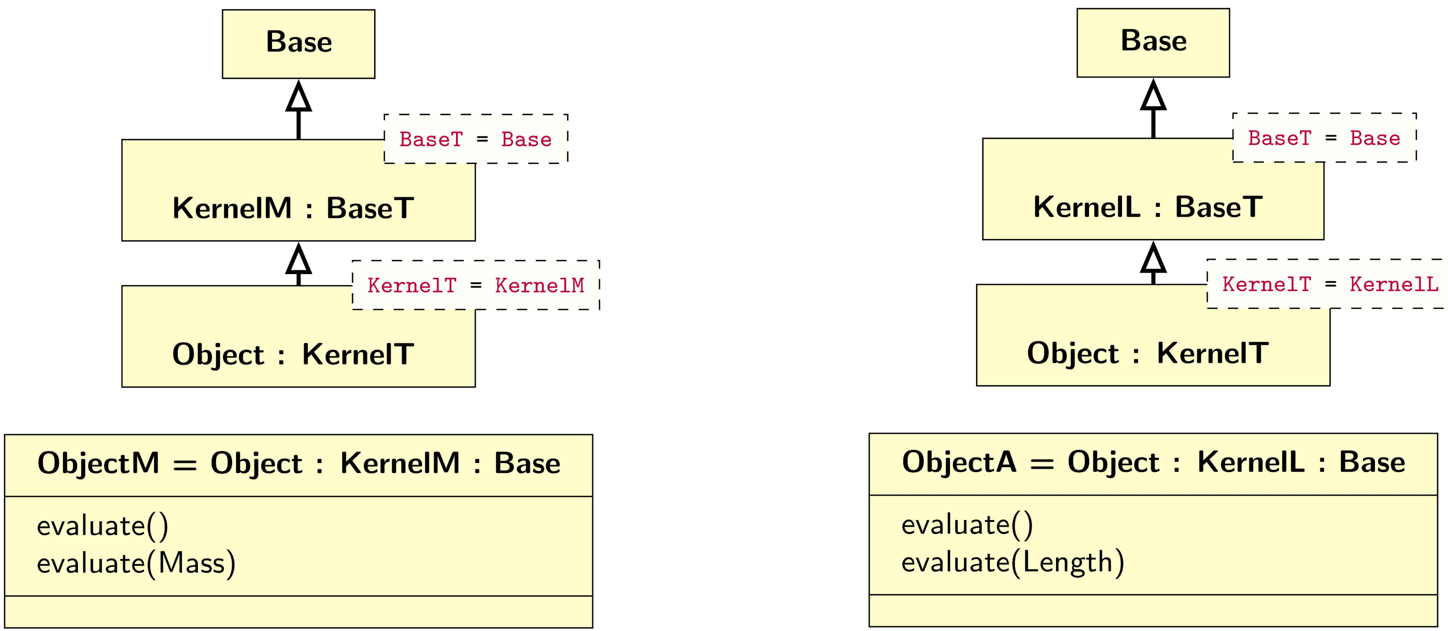
Semantic compile-time (static) polymorphism



[BD19] “Exploring future C++ features within a geometric modeling context”, Bratlie, Dalmo, 2019

Prototyping geometric modeling: C++, Paper IV, [BD19]

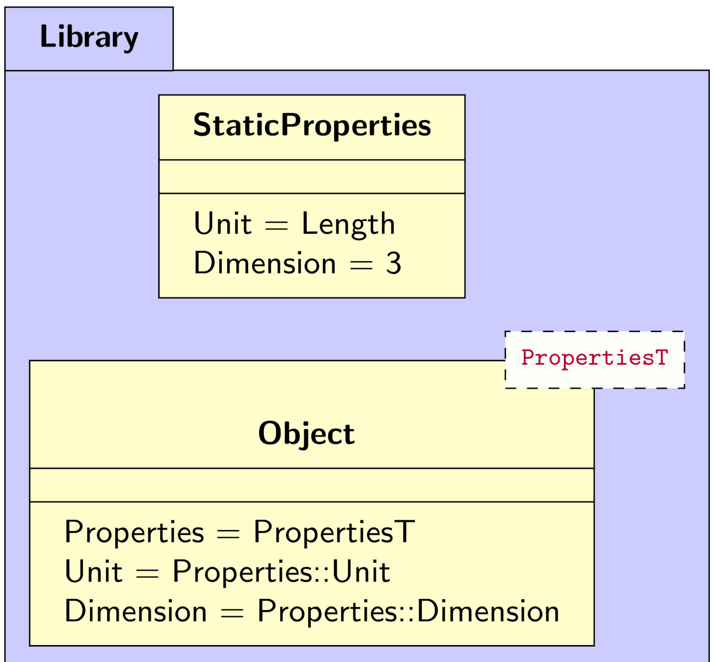
Semantic compile-time (static) polymorphism

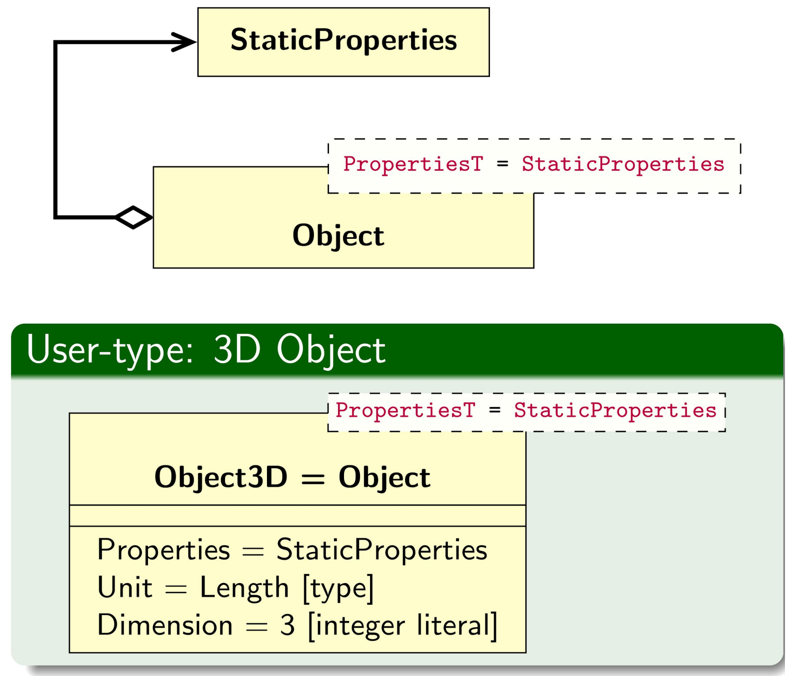


[BD19] “Exploring future C++ features within a geometric modeling context”, Bratlie, Dalmo, 2019

Prototyping geometric modeling: C++, Paper IV, [BD19]

Aggregated properties and Transitive constructors

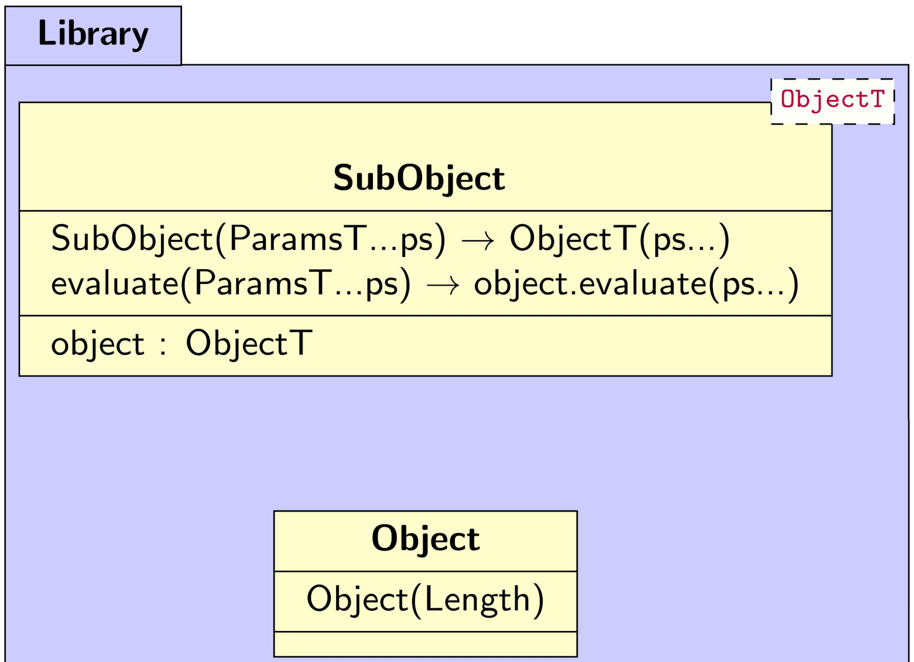


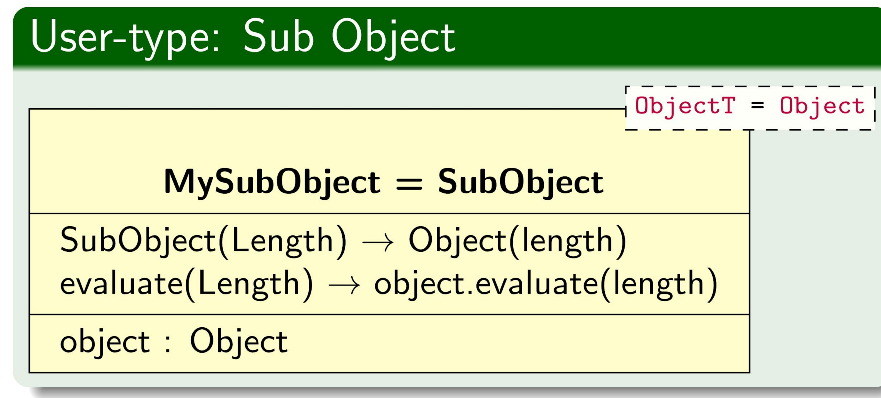


[BD19] "Exploring future C++ features within a geometric modeling context", Bratlie, Dalmo, 2019

Prototyping geometric modeling: C++, Paper IV, [BD19]

Aggregated properties and **Transitive constructors**





[BD19] “Exploring future C++ features within a geometric modeling context”, Bratlie, Dalmo, 2019

Prototyping geometric modeling: C++, Paper IV, [BD19]

Used to solve the following challenges

- Pseudo-reflection
- Dimension-aware initial values in generic members
- Generic sub-domain object deduction
- Static deduction of differential operators

[BD19] “Exploring future C++ features within a geometric modeling context”, Bratlie, Dalmo, 2019

Prototyping geometric modeling: C++, Paper IV, [BD19]

Concepts

- C++ has two types of placeholder-types
  - **auto** : the type of a value {aka. type}
  - **typename** : the name of a type {aka. type name}

- Concepts adds restrictions
  - Curve **auto** v : Type of V must fulfill requirements of Curve
  - template<Sortable T> : Type T must fulfill requirements of sortable

- Helps with
  - Overloadable => fixes SFINAE
  - Abstract on basis of concepts rather than types
  - Better error from compiler: *“container cannot be sorted because”*

[BD19] “Exploring future C++ features within a geometric modeling context”, Bratlie, Dalmo, 2019

Prototyping geometric modeling: C++, Paper IV, [BD19]

Contracts

- Programmatically document **Invariants**
  - Runtime requirements: value of **t** = [0,1]
- Valid on
  - parameters (expects), function return value (ensures), runtime (assert)
- Auditable

Reflection / Meta-classes

- Ask a type what members and types it has (meta-information)
- Extend the **language** as libraries extend a program
  - Create new user-defined types other than struct/class
  - Interface, point, curve, etc.
  - Remove tedious and error prone overhead

[BD19] “Exploring future C++ features within a geometric modeling context”, Bratlie, Dalmo, 2019

Thank you for your attention ^^,