Nonlinear geometric models

PH.D. DISSERTATION

written by Gábor Valasek

Supervisor: János Vida, C.Sc.



Department of Algorithms and Their Applications Faculty of Informatics Eötvös Loránd University

Doctoral School of Informatics Eötvös Loránd University Head: Erzsébet Csuhaj Varjú, D.Sc. Doctoral Training Program: Information Systems Head: András Benczúr, D.Sc.

Budapest, 2015

Contents

1	Intr	oducti	on	1				
	1.1	1 Background						
	1.2	Overview						
	1.3	Notati	on	6				
2	Geo	ometrio	c Hermite interpolation of curves	7				
	2.1	Overv	iew	7				
	2.2	Deriva	tives in the Frenet frame $\ldots \ldots \ldots$	0				
		2.2.1	Arc-length parameterized curves	0				
		2.2.2	Arbitrary parameterized curves	4				
			2.2.2.1 Application of Frenet coordinate theorems 19	9				
			2.2.2.2 Computation of geometric invariants 19	9				
			2.2.2.3 Conditions of G^n continuity $\ldots \ldots \ldots \ldots 20$	0				
			2.2.2.4 β -splines	2				
	2.3	Forma	lization of geometric Hermite interpolation 23	3				
	2.4	Solvin	g the reconstruction equations $\ldots \ldots \ldots \ldots \ldots \ldots \ldots 2^{\prime}$	7				
		2.4.1	Exact solutions	8				
2.4.2 Approximate solutions								
			2.4.2.1 Norm functionals $\ldots \ldots \ldots \ldots \ldots \ldots 33$	3				
			2.4.2.2 Taylor expansion based functionals	5				
		2.4.3	Symmetric interpolation constraints	6				
		2.4.4	General GH interpolation	9				
	2.5	Paran	netrization optimization $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 4$	1				
	2.6	Adapt	ive curve fitting $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 44$	5				
	2.7	Algori	thms of GH interpolation	1				
		2.7.1	Direct methods	2				
		2.7.2	Indirect methods	2				

3	Geo	ometric	c Hermite interpolation of Surfaces	66				
	3.1 Overview							
	3.2 Second order geometric Hermite interpolation of surfaces \therefore							
		3.2.1	Introduction	67				
		3.2.2	Conditions for second order GH interpolation \ldots	67				
		3.2.3	Quadrilateral Bézier patches	74				
			3.2.3.1 Four corner GH interpolation $\ldots \ldots \ldots$	74				
			3.2.3.2 Bi-quintic Bézier patch	79				
			3.2.3.3 Bi-quartic Bézier patch	80				
			3.2.3.4 Bi-cubic Bézier patch	86				
	3.2.4 Triangular Bézier patches							
	3.3	Geome	etric Hermite surface interpolation $\ldots \ldots \ldots \ldots \ldots$	90				
		3.3.1	Lines of curvature	91				
		3.3.2	Derivatives in the Darboux-frame	92				
		3.3.3	Geometric continuity and lines of curvature \ldots	102				
		3.3.4	Formalization of general geometric Hermite interpolation	107				
	3.4	Solutio	on of GH surface reconstruction	112				
		3.4.1	Exact solutions	113				
		3.4.2	Approximate solutions	114				
		3.4.3	Parametrization optimization	116				
	3.5 Algorithms of GH surface interpolation							
		3.5.1	Blending-based methods	119				
		3.5.2	Paraboloid-based methods	122				
			3.5.2.1 Second order GH base-paraboloids \ldots .	122				
			3.5.2.2 Continuous connection along boundaries	125				
			3.5.2.3 Extension to higher order GH interpolation \therefore	127				
1	Sun	nmary		198				
T	Ack	nowleda	rement	130				
	1100	nowneug	50110110	100				
A	ppen	dices		131				
\mathbf{A}	Pro	of of I	Theorems and Lemmas of curves	132				
	A.1	Equiva	alence of Frenet x coordinates and derivatives of arc-length					
		functio	ons	132				
	A.2	Expres	ssion of geometric invariants with Frenet coordinates \ldots	134				

	A.3 Conditions of geometric continuity in terms of Frenet coordinates 133							
в	Pro	of of p	arabolic GH interpolant reconstruction	137				
С	Seco	ond or	der GH interpolation with triangular patches	145				
	C.1	Three	corner GH interpolation	145				
		C.1.1	Quintic triangular Bézier patch	147				
		C.1.2	Quartic triangular Bézier patch	149				
		C.1.3	Cubic triangular Bézier patch	149				
D	Diff	erentia	al geometry of lines of curvature	152				
\mathbf{E}	Par	aboloi	d solutions to second and third order GH interpo	ola-				
	tion	L		156				
	E.1	Second	d order paraboloids	156				
		E.1.1	$Principal \ curvature \ reconstruction \ . \ . \ . \ . \ . \ . \ .$	156				
		E.1.2	Principal and geodesic curvature reconstruction \ldots	157				
	E.2	Third	order paraboloids	163				
		E.2.1	Normal projection reconstruction	164				

Chapter 1

Introduction

1.1 Background

Representation of shapes is of primary importance in every geometric problem. The answer to a geometric query may be trivial in one representation, and might require a more elaborate mathematical process in the other.

For example, given a point in space, it is trivial to determine if this point is on a surface that is defined by an implicit equation. The same is not true if the surface is represented parametrically.

In computer-aided design (CAD), and in many differential geometric topics, parametric representation of shapes has become the most widespread formulation, which defines the shape as the image of a mapping.

In the CAD domain, the designer is mostly concerned with the *geometry* of the shape, that is, the image of the parametric function. The actual *mapping* between the parametric domain and the shape may not necessarily be specified, and as long as the image stays the same, this mapping may be changed.

Still, in many applications the mapping itself is a crucial component as well. For instance, tool paths can be represented by parametric curves in NC machining [38]. Here, the parametrization determines how fast the tool travels along the path. This motion, however, is subject to physical constraints, such as that the tool cannot move at arbitrary speeds, and the parametrization has to conform to these restrictions. Numerical reparametrizations are often used to yield a feasible or near-optimal tool path speed [13].

The most widespread parametric representations in use, such as Bézier curves, B-splines, NURBS, etc. do not provide means to trivially separate the geometric properties of the shape from parametrization.

One of the goals of this thesis is to present a basis-independent formulation to carry out this separation.

In the case of curves, this leads us to the investigation of how derivatives of a curve change in the Frenet-frame. Arc-length derivatives depend on purely geometric properties. I show, by a recurrence formula, how parametrization alters the Frenet coordinates of derivatives in relation to the arc-length case. Even though this result is a direct consequences of elementary differential geometric theorems, there is no mention of such a formula to the author's knowledge in the classic and more recent literature on differential geometry [9], [45], [27], [43], [44], [39].

Quantitative separation of parametrization from geometry requires a greater effort in the case of surfaces. By using parameter lines that are both lines of curvatures and arc-length, I show that, at non-umbilical points, the geometry of the surface can be defined via the geometry of lines of curvature. After investigating the properties of this *natural parametrization*, I show how partial derivatives of an aribtrary parametrization use the differential geometric properties of the lines of curvature. Again, an appropriate frame of reference has to be fixed to carry out this study, for which the Darboux-frame is the most natural choice.

Once geometric constraints are separated from the degrees of freedom of parametrization, the thesis discusses the reconstruction of prescribed geometric quantities at knots, that is, the problem of geometric Hermite (GH) interpolation.

Classical Hermite interpolation creates curves that reconstruct prescribed positional and derivative vector data at parametric endpoints. For instance, cubic Hermite interpolation yields a cubic polynomial curve with given endpoint positions and first derivative vectors.

The geometric Hermite analogue of this problem is the reconstruction of endpoint position and tangent *directions*, leaving the actual length of the first derivative vectors unspecified (but non-zero). This means two new scalar degrees of freedom, which can be used to improve the quality of the curve or even to decrease the algebraic degree of a polynomial solution.

The first published industrial application of this method was a second order GH interpolation problem of reconstructing position, tangent direction, normal vector and curvature values at endpoints. In [26], Klass created a *cubic* integral polynomial curve to approximate the offset of a given curve by reconstructing the position, tangent, normal and curvatures of the offset. Klass found via experiments that the second order GH interpolant provided a high accuracy approximation to the offset and implemented this method in an automobile CAD system in 1983.

The exact accuracy of this approximation was not verified algebraically, however, until 1987. That year – independently of Klass –, de Boor, Höllig, and Sabin investigated the same problem of reconstructing second order geometric invariants at knots with cubic splines [7], also referred to as BHS splines. They found that such an approximation is sixth order accurate, that is, if one doubles the density of sampling, the error drops by $\frac{1}{2^6}$, which coincides with the accuracy of *quintic* Hermite polynomials. They also addressed the issue of existence, and proved that if the progenitor curve – from which the second order GH data were sampled – is smooth enough, its curvature does not vanish, and sample points are close, there is always a cubic solution to this problem, see [7] for details. This paper also introduced the term geometric Hermite interpolation to address these types of problems.

Both of the above papers handled GH interpolation as a tool of curve approximation, thus the existence conditions were formulated in accordance with that framework. However, these are not tangible concepts from a design point of view: there is no underlying curve in that setting, instead, second order GH data are specified by the user either directly or indirectly.

To alleviate this, in [41] Schaback gave purely geometric constraints on when a given pair of second order GH data can be reconstructed by cubic polynomials. He also presented necessary and sufficient conditions for the existence of quartic interpolants, and shown when only a quintic polynomial can achieve reconstruction. This has also shown that the interpolation problem investigated by Klass and de Boor et. al. are analogous to the quintic Hermite interpolation, the reconstruction of positional, first, and second derivative data.

Indeed, it was Mørken who pointed out that this cubic second order GH interpolant can be considered as a reparametrization of a quintic polynomial such that the coefficients of the quintic and quartic terms become zero. He provided a mathematical analysis of when such a degree reducing parametrization is possible in [35]. He did preliminary work on generalizing his parametric



Figure 1.1: Design and refinement process using control circles to define cubic Bézier curves. From left to right: control circle hierarchy, Bézier segments connected, application of curve-parameter dependent brush.

approximation framework to surfaces as well [34].

The main goal of this thesis is to provide a general formalization of geometric Hermite interpolation of curves and surfaces in arbitrary bases. The formulation presented here allows us to unify the design and approximationtheoretical view on the GH interpolation problem. Incorporation of degree reducing reparametrizations, multiple point, and mixed-order GH reconstruction also becomes trivial with this formulation.

It is important to emphasize that the geometric model presented here is not a substitute for the existing representations, already implemented and dominating in CAD systems. Instead, it acts as an *interface* to separate the geometric and the parametrization components of the control data of an arbitrary underlying representation, whenever possible. As an example, Figure 1.1 illustrates how purely geometric input can be used to provide point, tangent, curvature input for a design process that yields cubic Bézier curves [22], [23].

1.2 Overview

The thesis consists of two main chapters, one for the discussion of GH interpolation and related topics for curves and surfaces, respectively.

The theoretical results presented in the thesis are followed by examples that are included to illustrate the usage of propositions, and to help the validation of these results in relation to already established concepts in differential geometry of CAGD. Proofs of Lemmas and Theorems that are technical in nature are usually disclosed in the Appendix. The main text of the thesis only contains the statement of these and the reference to the proof.

Chapter 2 focuses on GH interpolation of curves. In order to do that, I derive a recurrence formula that describes how parametrization affects the Frenet coordinates of derivatives. To illustrate the flexibility of this recurrence, I also show how it can be used to compute geometric invariants of curves and to validate an arbitrary order geometric continuity of joining segments.

In Section 2.4, I show that once parametrization is set, point-wise geometric reconstruction becomes a linear problem. I formulate the general GH interpolation problem of curves in Section 2.3. It is followed by the investigation of solvability of the geometric reconstruction, for which I derive a general existence theorem, and also provide worst-case degree bounds for polynomial solutions. For cases the conditions of the existence theorem fail, I show how a least squares sense best approximation to the problem can be computed, as well as investigate other norms and functionals.

If parametrization is not fixed, one can formulate quality measures on parametrizations that yield a non-linear optimization problem which is presented in Section 2.5. This way, a best parametrization – in the sense of some real valued functional – can be found that still satisfies the prescribed geometric constraints.

The extension of these results to surfaces is presented in Chapter 3. Due to the lack of a throughout geometric characterization of higher order GH problems of surfaces, the chapter begins with the investigation of second order GH interpolation, where point-wise position, surface normal, principal directions and principal curvatures are to be reconstructed.

Section 3.3.3 derives the formulation of general GH interpolation of surfaces by connecting the differential geometric properties of lines of curvature with the geometry of surfaces, and translating the conditions of geometric continuity to properties of lines of curvature.

The geometric reconstruction is once again a linear problem, if the parametrization is fixed, and the chapter continues with establishing existence conditions analogously to the case of curves. Algorithms to construct geometric Hermite interpolants close the chapter.

Finally, Chapter 3 iterates over the main contributions of this thesis.

1.3 Notation

The following table summarizes the notations used throughout this thesis:

$\mathbb{R}, \mathbb{R}^+_0, \mathbb{R}^0$	set of all, non-negative, and non-positive real numbers
$f \circ g$	the $f(g(\cdot))$ composition of functions f and g
\mathbb{E}^n	the n -dimensional Euclidean space
$\mathbf{a},\mathbf{b}\in\mathbb{E}^n$	a point in the n -dimensional Euclidean space
$\Delta \mathbf{b}_j = \mathbf{b}_{j+1} - \mathbf{b}_j$	
$\Delta^{i+1}\mathbf{b}_j = \Delta^i \mathbf{b}_{j+1} - \Delta^i \mathbf{b}_j$	forward differences on a \mathbf{b}_i sequence of points
$\mathbf{x},\mathbf{y}\in\mathbb{R}^n$	a vector in the <i>n</i> -dimensional vector space over \mathbb{R}
$ \mathbf{x} , \mathbf{x} _2$	length of vector \mathbf{x}
$e,f \subset \mathbb{E}^2, g,h \subset \mathbb{E}^3$	a line in plane or space
$\mathbf{E},\mathbf{F}\subset\mathbb{E}^{3}$	a plane in the Euclidean space
$\mathbf{x} \cdot \mathbf{y}, \ \mathbf{xy}$	dot product of two <i>n</i> -dimensional vectors
$\mathbf{x} \times \mathbf{y}$	cross product of two n -dimensional vectors
$(\mathbf{i},\mathbf{j}),(\mathbf{i},\mathbf{j},\mathbf{k})$	a basis in \mathbb{R}^2 and \mathbb{R}^3 , respectively
$(\mathbf{p}; \mathbf{i}, \mathbf{j}), (\mathbf{p}; \mathbf{i}, \mathbf{j}, \mathbf{k})$	a coordinate system with origin ${\bf p}$ and axes ${\bf i}, {\bf j}(, {\bf k})$
$\hat{\mathbf{r}}(t): [0, L] \to \mathbb{E}^n, \ n = 2, 3$	an arc-length parametrized curve
$\mathbf{r}(t):[a,b] \to \mathbb{E}^n, \ n=2,3$	an arbitrary parametrized regular curve
$s(t):[a,b] \to [0,L]$	arc-length function of a curve
$L, L_{\mathbf{r}(t)} \in \mathbb{R}_0^+$	the arc-length of a curve
$\mathbf{r}'(t), \frac{d^n}{dt^n}\mathbf{r}(t)$	differentiation with respect to the curve parameter
$\hat{\mathbf{r}}'(s), rac{d^n}{ds^n} \hat{\mathbf{r}}(s)$	differentiation with respect to arc-length
$\hat{\mathbf{r}}(s,t), \mathbf{r}(u,v)$	the natural and an arbitrary parametric surface
$\hat{\mathbf{r}}_s, \mathbf{r}_t, \partial_{s^k} \hat{\mathbf{r}}, \mathbf{r}_u, \mathbf{r}_v, \partial_{u^k} \mathbf{r}$	partial derivatives
$\hat{\mathbf{t}}(s), \hat{\mathbf{n}}(s), \hat{\mathbf{b}}(s) : [0, L] \to \mathbb{R}^3$	Frenet frame vectors of an arc-length parametrized curve
$\mathbf{t}(t), \mathbf{n}(t), \mathbf{b}(t) : [a, b] \to \mathbb{R}^3$	Frenet frame vectors of an arbitrary parametrized curve
$\kappa(t), \tau(t): [a, b] \to \mathbb{R}$	curvature and torsion function of a parametric curve
$\hat{\kappa}(t), \tau(t) : [0, L] \to \mathbb{R}$	arc-length parametrized curvature and torsion functions
$[x, y, z]_F^T$	a vector with coordinates (x, y, z) in $F = (\mathbf{i}, \mathbf{j}, \mathbf{k})$ basis
$[x,y,z]_F^T$	a point with coordinates (x, \overline{y}, z) in $F = (\mathbf{p}; \mathbf{i}, \mathbf{j}, \mathbf{k})$

Chapter 2

Geometric Hermite interpolation of curves

2.1 Overview

Hermite interpolation is a classic method in numerical analysis, which has been used in many areas, ranging from curve approximations to numerical integration quadratures (the Gauss-Hermite quadrature), computer animation design [37], and even parametric surface constructions, such as the Ferguson patches [12].

Hermite interpolation yields polynomial curves such that they interpolate prescribed position and derivative data at given parameter values, or in other words knots. Let us consider the simplest case of this, the two endpoint interpolation problem, that is, let there be given an

$$\mathbf{H}_j = (\mathbf{p}_j \in \mathbb{E}^3; \mathbf{m}'_j, \mathbf{m}''_j, ..., \mathbf{m}_j^{(m)} \in \mathbb{R}^3) , j = 0, 1$$

pair of data tuples, and find a $\mathbf{p}(t): [0,1] \to \mathbb{E}^3$ polynomial curve such that

$$\mathbf{p}(j) = \mathbf{p}_j$$

 $\mathbf{p}^{(i)}(j) = \mathbf{m}_j^{(i)}, \ i = 1, 2, .., m$

holds, j = 0, 1, where $\mathbf{p}^{(i)}(t)$ denotes the *i*-th derivative at *t*.

Hermite interpolation guarantees the existence of a degree 2(m + 1) - 1polynomial solution to the above problem and it also constructs it. If input position and derivative data \mathbf{H}_j originates from the sampling of a smooth curve, the resulting Hermite polynomial interpolant has an approximation order of (m+1).

The input data of Hermite interpolation consists of points and vectors, that is, both representation - in this case, parametrization - dependent and independent quantities. As a result, when used for approximation, Hermite interpolation yields a curve that is affected by both the *parametrization* and the *geometry* of the original curve.

In certain application this is a desired property, such as in animation design, where not only the path of a given object is of importance, but also the speed at which it is traversed. In these instances, dependence on parametrization is a deesirable attribute.

In other areas, however, this is not the case. Shape representations are to be dependent only on the geometry of a given objects, not on any of its particular parametrizations.

Geometric Hermite (GH) interpolation is an expression coined by de Boer et. al. [7], which refers to a special generalization of Hermite interpolation, where only parametrization independent quantities are reconstructed. In the case covered by their paper, it meant that instead of interpolating

$$\mathbf{H}_j = (\mathbf{p}_j \in \mathbb{E}^3, \mathbf{m}'_j, \mathbf{m}''_j \in \mathbb{R}^3) , \ j = 0, 1$$
(2.1)

position, first, and second derivative data at endpoints, they created an integral polynomial curve that reconstructed prescribed

$$\mathbf{H}_j = (\mathbf{p}_j \in \mathbb{E}^3, \mathbf{t}_j \in \mathbb{R}^3, \kappa_j \in \mathbb{R}) , \ j = 0, 1$$

data tuples consisting of position, unit tangent direction, and curvature value. They obtained the normal vectors by a 90 degrees rotation of \mathbf{t}_j . Compared to classical Hermite interpolation, \mathbf{t}_j is the substitute of \mathbf{m}'_j and \mathbf{n}_j , κ_j together replace \mathbf{m}''_j as input data.

It turns out that their cubic piecewise polynomial curve, hereafter referred to as the BHS spline, is an even more special case of Hermite interpolation. The approximation order of a BHS spline is six, which could have only been achieved by a quintic polynomial with classic Hermite interpolation.

Indeed, it was shown by Schaback [41], that in general, the solution to

the second order geometric Hermite interpolation problem of reconstructing an arbitrary pair of

$$\mathbf{D}_{j} = (\mathbf{p}_{j}; \mathbf{t}_{j}, \mathbf{b}_{j}; \kappa_{j}) , |\mathbf{t}_{j}| = |\mathbf{b}_{j}| = 1, j = 0, 1$$
 (2.2)

geometric Hermite data tuples at endpoints requires a polynomial curve of degree 5. The existence of a cubic interpolant requires strict geometric constraints on the input \mathbf{D}_{j} tuples.

Obviously, a degree 5 solution always exists to reconstruct (2.2), but instead of using quintics, de Boor et. al. created a cubic interpolant, which retained the approximation properties of a degree 5 polynomial. Such a cubic solution may not exist for arbitrary input - the algebraic conditions for the existence of a cubic solution were also given by de Boor et. al. in [7].

Equivalent, geometric existence conditions were given by Schaback in [41], who formulated intuitive, geometric constraints on the input data so that a degree 3, and also, a degree 4, solution can be created.

These differing approaches may also be explained by the fact that while de Boor et. al. used BHS splines in the context of curve approximations – thus stipulating restrictions on the *input curve* and its sampling –, while Schaback considered geometric Hermite interpolation as a design tool, where the *input* data relies only on the user, and not on any underlying curve.

It is important to note that by replacing (2.1) with (2.2), two scalar degrees of freedom are introduced per endpoint, or in general, per knot. That is, unlike the classic Hermite quintic interpolant, the geometric Hermite quintic interpolant is not uniquely defined. These new degrees of freedom may be used to optimize the curve, for example by minimizing certain energy functionals, or, even to *reduce* the degree of the interpolant.

The latter allows us to think of the BHS spline as a *reparametrization* of a quintic geometric Hermite interpolant, such that the coefficients of the degree 5 and 4 are zeroed out.

This reparametrization-based approach was taken by Mørken and Sherer in [35]. Within the domain of approximation theory, they gave algebraic conditions that an input curve should satisfy so that a degree-reducing reparametrization is guaranteed to exist in the neighborhood of a single point. They proved that in the neighborhood of a point of non-vanishing curvature, any planar curve can be approximated with sixth-order accuracy by integral cubics. These results were also extended to approximation of curve segments, and to higher orders.

In section 2.3, I present a formalization of general geometric Hermite interpolation, which depends only on geometric, parametrization-independent attributes. However, in order to do that, first we have to examine how parametrization affects the geometry of a curve, more precisely, how the derivatives of a given curve are affected by parametrization.

2.2 Derivatives in the Frenet frame

As a reminder, a parametric curve is defined by an $\mathbf{r}(t) : [a, b] \to \mathbb{E}^3$ mapping of a real interval [a, b] to the Euclidean space. Quantitatively separating the effects of the *mapping* from the properties of the actual geometry - the *image* of the mapping - is important in the discussion of geometric reconstruction.

In this section, I derive a recurrence formula that describes the geometry of derivative vectors in the Frenet-frame, such that the effects of parametrization can be studied separately from the geometric invariants. This formula is a direct consequence of the Frenet-Serret formula.

2.2.1 Arc-length parameterized curves

Let $\hat{\mathbf{r}}(s) : [0, L] \to \mathbb{E}^3$ be an arc-length parameterized curve, where L > 0 denotes the arc-length of $\hat{\mathbf{r}}(s)$. Let us also assume that $\hat{\mathbf{r}}(s)$ is biregular, that is, $\hat{\mathbf{r}}'(s)$ and $\hat{\mathbf{r}}''(s)$ are linearly independent for all $s \in [0, L]$.

The Frenet-frame $F = (\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{b}})$ of $\hat{\mathbf{r}}(s)$ consists of the orthonormal vectors

$$\hat{\mathbf{t}}(s) = \hat{\mathbf{r}}'(s) , \hat{\mathbf{n}}(s) = \frac{\hat{\mathbf{r}}''(s)}{|\hat{\mathbf{r}}''(s)|} , \hat{\mathbf{b}}(s) = \hat{\mathbf{t}}(s) \times \hat{\mathbf{n}}(s) ,$$

forming a right handed orthonormal basis of \mathbb{R}^3 , $s \in [0, L]$.

Hereafter, arc-length parameterized curves are denoted by a hat, and prime always denotes differentiation with respect to the actual parametrization of the argument, i.e. $\hat{\mathbf{r}}'$ denotes the derivative of $\hat{\mathbf{r}}$ with respect to arc-length. For the sake of simplicity, let us omit the parameters from the formulae.

The Frenet-Serret formula relates the change of Frenet-frame to the curvature and torsion of the curve as [9]

$$\begin{aligned} \hat{\mathbf{t}}' &= & \hat{\kappa} \hat{\mathbf{n}} \\ \hat{\mathbf{n}}' &= & -\hat{\kappa} \hat{\mathbf{t}} & +\hat{\tau} \hat{\mathbf{b}} \\ \hat{\mathbf{b}}' &= & -\hat{\tau} \hat{\mathbf{n}} \end{aligned}$$

Using the Frenet-Serret formula, I prove below that the Frenet coordinates of successive arc-length derivatives are subject to a simple recurrence formula.

To show this, consider three scalar functions, $\hat{a}(s), \hat{b}(s), \hat{c}(s) : [0, L] \to \mathbb{R}$. Using the rule of differentiation of products and the Frenet formula, one gets

$$(\hat{a} \cdot \hat{\mathbf{t}})' = \hat{a}'\hat{\mathbf{t}} + \hat{a}\hat{\kappa}\hat{\mathbf{n}}$$
$$(\hat{b} \cdot \hat{\mathbf{n}})' = \hat{b}'\hat{\mathbf{n}} - \hat{b}\hat{\kappa}\hat{\mathbf{t}} + \hat{b}\hat{\tau}\hat{\mathbf{b}}$$
$$(\hat{c} \cdot \hat{\mathbf{b}})' = \hat{c}'\hat{\mathbf{b}} - \hat{c}\hat{\tau}\hat{\mathbf{n}}$$

By introducing the notation $[a, b, c]_F^T = a \cdot \hat{\mathbf{t}} + b \cdot \hat{\mathbf{n}} + c \cdot \hat{\mathbf{b}}$ for the Frenet coordinates of $\hat{\mathbf{r}}^{(n)}$, the above can be written as

$$\begin{pmatrix} \begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix}_{F} \end{pmatrix}' = \begin{bmatrix} \hat{a}' \\ \hat{b}' \\ \hat{c}' \end{bmatrix}_{F} + \begin{bmatrix} 0 & -\hat{\kappa} & 0 \\ \hat{\kappa} & 0 & -\hat{\tau} \\ 0 & \hat{\tau} & 0 \end{bmatrix} \begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix}_{F}$$

$$= \begin{bmatrix} \hat{a}' \\ \hat{b}' \\ \hat{c}' \end{bmatrix}_{F} + \begin{bmatrix} \hat{\tau} \\ 0 \\ \hat{\kappa} \end{bmatrix} \times \begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix}_{F}$$

$$(2.3)$$

using the matrix form of the cross product:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \times \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}.$$

Let $\hat{x}_i, \hat{y}_i, \hat{z}_i \in \mathbb{R}$ denote the coordinates of $\hat{\mathbf{r}}^{(i)}$ in the Frenet frame. By applying (2.3) to $\hat{x}_1 = 1, \hat{y}_1 = 0, \hat{z}_1 = 0$ recursively, and recalling that all these coordinates depend on the arc-length parameter, the following is proved:

Theorem 1 Let $\hat{\mathbf{r}}(s) : [0, L] \to \mathbb{E}^3$ be an arc-length parameterized bi-regular curve. For all $s \in [0, L]$, the coordinates of $\hat{\mathbf{r}}^{(n+1)}(s)$ are

$$\hat{\mathbf{r}}^{(n+1)} := \begin{bmatrix} \hat{x}_{n+1} \\ \hat{y}_{n+1} \\ \hat{z}_{n+1} \end{bmatrix}_F = \begin{bmatrix} \hat{x}'_n \\ \hat{y}'_n \\ \hat{z}'_n \end{bmatrix}_F + \begin{bmatrix} 0 & -\hat{\kappa} & 0 \\ \hat{\kappa} & 0 & -\hat{\tau} \\ 0 & \hat{\tau} & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_n \\ \hat{y}_n \\ \hat{z}_n \end{bmatrix}_F , \qquad (2.4)$$

where $n = 1, 2, ..., and \hat{x}_1 = 1, \hat{y}_1 = 0, \hat{z}_1 = 0, and F$ denotes the Frenet-frame at s.

It is also important to note the flow of the coordinates: with the appropriate multiplicative coefficients, the \hat{y}_n coordinate of $\hat{\mathbf{r}}^{(n)}$ is added to \hat{x}_{n+1} and \hat{z}_{n+1} of $\hat{\mathbf{r}}^{(n+1)}$, while \hat{x}_n and \hat{z}_n are added to \hat{y}_{n+1} . More precisely, by expanding the matrix-vector multiplication in (2.4), one gets

$$\begin{aligned} \hat{x}_{n+1} &= \hat{x}'_n & -\hat{\kappa} \cdot \hat{y}_n \\ \hat{y}_{n+1} &= \hat{y}'_n &+ \hat{\kappa} \cdot \hat{x}_n & -\hat{\tau} \cdot \hat{z}_n \\ \hat{z}_{n+1} &= \hat{z}'_n & + \hat{\tau} \cdot \hat{y}_n \end{aligned}$$

The Fundamental Theorem of space curves [45] states that the curvature and torsion functions uniquely define a curve, up to a rigid body motion. Theorem 1 shows quantitatively how they define the geometry of the derivatives at a point.

It is important to note that the highest derivative of curvature in the Frenet coordinates always appears along the normal first, while the first occurrence of the highest derivative of torsion is along the binormal.

More precisely, the *j*-th derivative of the curvature w.r.t. arc-length, $\hat{\kappa}^{(j)}$ first appears in \hat{y}_{j+2} , j = 0, 1, The torsion derivative $\hat{\tau}^{(j)}$ is formally introduced into the derivative coordinate formula in \hat{z}_{j+3} , however, the actual value of the $\hat{\tau}^{(j)}$ torsion may not be computable from $\hat{\mathbf{r}}^{(j+3)}$, because $\hat{\tau}^{(j)}$ appears multiplied by $\hat{\kappa}$, that is, if the curvature vanishes, the value of $\hat{\tau}^{(j)}$ is masked in \hat{z}_{j+3} .

In particular, it was noted by Ye and Maekawa in [56], that the formula to compute the torsion in the case of $\hat{\kappa} = \hat{\kappa}' = ... = \hat{\kappa}^{j-1} = 0$ and $\hat{\kappa}^j \neq 0$ is

$$\hat{\tau} = \frac{\hat{\mathbf{r}}^{(j+3)} \cdot \hat{\mathbf{b}}}{(j+1)\hat{\kappa}^{(j)}}.$$

Example: in order to illustrate the computational use of Theorem 1, let us consider the first three derivatives of $\hat{\mathbf{r}}(s)$.

Using the Frenet formula and the chain rule of differentiation, the first, second, and third derivatives of $\hat{\mathbf{r}}(s)$ are

$$\hat{\mathbf{r}}' = \hat{\mathbf{t}} \tag{2.5}$$

$$\hat{\mathbf{r}}'' = (\hat{\mathbf{t}})' = \hat{\kappa}\hat{\mathbf{n}} \tag{2.6}$$

$$\hat{\mathbf{r}}^{\prime\prime\prime} = (\hat{\kappa}\hat{\mathbf{n}})^{\prime} = \hat{\kappa}^{\prime}\hat{\mathbf{n}} - \hat{\kappa}^{2}\hat{\mathbf{t}} + \hat{\kappa}\hat{\tau}\hat{\mathbf{b}}$$
(2.7)

or, with their Frenet coordinates,

$$\hat{\mathbf{r}}' = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{F}, \ \hat{\mathbf{r}}'' = \begin{bmatrix} 0\\\hat{\kappa}\\0 \end{bmatrix}_{F}, \ \hat{\mathbf{r}}''' = \begin{bmatrix} -\hat{\kappa}^{2}\\\hat{\kappa}'\\\hat{\kappa}\hat{\tau} \end{bmatrix}_{F}.$$
(2.8)

According to Theorem 1, the first derivative is

$$\hat{\mathbf{r}}' = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_F ,$$

the second derivative is computed as

$$\hat{\mathbf{r}}'' = \begin{bmatrix} \hat{x}_1' \\ \hat{y}_1' \\ \hat{z}_1' \end{bmatrix}_F^{+} \begin{bmatrix} 0 & -\hat{\kappa} & 0 \\ \hat{\kappa} & 0 & -\hat{\tau} \\ 0 & \hat{\tau} & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{z}_1 \end{bmatrix}_F^{+} \\ = \begin{bmatrix} 1' \\ 0' \\ 0' \end{bmatrix}_F^{+} \begin{bmatrix} 0 & -\hat{\kappa} & 0 \\ \hat{\kappa} & 0 & -\hat{\tau} \\ 0 & \hat{\tau} & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_F^{+} \begin{bmatrix} 0 \\ \hat{\kappa} \\ 0 \end{bmatrix}_F^{+} ,$$

and the third derivative is

$$\hat{\mathbf{r}}^{\prime\prime\prime} = \begin{bmatrix} 0'\\ \hat{\kappa}'\\ 0' \end{bmatrix}_{F}^{} + \begin{bmatrix} 0 & -\hat{\kappa} & 0\\ \hat{\kappa} & 0 & -\hat{\tau}\\ 0 & \hat{\tau} & 0 \end{bmatrix} \begin{bmatrix} 0\\ \hat{\kappa}\\ 0 \end{bmatrix}_{F}^{}$$
$$= \begin{bmatrix} 0\\ \hat{\kappa}'\\ 0 \end{bmatrix}_{F}^{} + \begin{bmatrix} -\hat{\kappa}^{2}\\ 0\\ \hat{\kappa}\hat{\tau} \end{bmatrix}_{F}^{} = \begin{bmatrix} -\hat{\kappa}^{2}\\ \hat{\kappa}'\\ \hat{\kappa}\hat{\tau} \end{bmatrix}_{F}^{}$$

These are equal to (2.5)-(2.7).

2.2.2 Arbitrary parameterized curves

Without loss of generality, let us assume that the arbitrary parameterized curve $\mathbf{r}(t)$ is a biregular reparametrization of an arc-length parameterized curve $\hat{\mathbf{r}}(s) : [0, L] \to \mathbb{E}^3$, that is, $\mathbf{r}(t) : [0, 1] \to \mathbb{E}^3$ is such that

$$\mathbf{r}(t) = \hat{\mathbf{r}}(s(t)) \; ,$$

where $s(t): [0,1] \to [0,L]$ is the arc-length function of $\mathbf{r}(t)$, i.e.

$$s(t) = \int_0^t |\mathbf{r}'(x)| dx \; .$$

Let us also assume that the reparametrized curve remains biregular, that is $\mathbf{r}'(t) \neq 0$, and $\mathbf{r}'(t), \mathbf{r}''(t)$ are linearly independent.

The Frenet-frame of an arbitrary parametrized curve is computed as

$$\begin{split} \mathbf{t} &= \frac{\mathbf{r}'}{|\mathbf{r}'|} \ ,\\ \mathbf{n} &= \mathbf{b} \times \mathbf{t} \ ,\\ \mathbf{b} &= \frac{\mathbf{r}' \times \mathbf{r}''}{|\mathbf{r}' \times \mathbf{r}''|} \end{split}$$

•

The Frenet-Serret formulae for arbitrary speed curves are

$$\begin{bmatrix} \mathbf{t}' \\ \mathbf{n}' \\ \mathbf{b}' \end{bmatrix} = |\mathbf{r}'| \cdot \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \mathbf{n} \\ \mathbf{b} \end{bmatrix} ,$$

where $\mathbf{t} = \hat{\mathbf{t}} \circ s$, $\mathbf{n} = \hat{\mathbf{n}} \circ s$, $\mathbf{b} = \hat{\mathbf{b}} \circ s$ and $\kappa = \hat{\kappa} \circ s$, $\tau = \hat{\tau} \circ s$, see [9].

A general, recursive formula for the Frenet frame coordinates of the derivatives are obtained similarly to the arc-length case: let $a(t), b(t), c(t) : [0, 1] \to \mathbb{R}$ be scalar functions. It follows from the arbitrary speed Frenet formula that

$$(a \cdot \mathbf{t})' = a'\mathbf{t} + as'\kappa\mathbf{n}$$
$$(b \cdot \mathbf{n})' = b'\mathbf{n} - bs'\kappa\mathbf{t} + bs'\tau\mathbf{b}$$
$$(c \cdot \mathbf{b})' = c'\mathbf{b} + cs'\tau\mathbf{n}$$

Denoting the Frenet-frame coordinates of $\mathbf{r}^{(i)}$ by $x_i, y_i, z_i \in \mathbb{R}, i = 1, 2...,$ the following is proved:

Theorem 2 Let $\mathbf{r}(t) : [0,1] \to \mathbb{E}^3$ be a biregular parametric curve with arclength function $s(t) : [0,1] \to [0,L]$. Frenet frame coordinates of the derivative vectors of $\mathbf{r}(t)$ are subject to

$$\mathbf{r}^{(n+1)} := \begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix}_F = \begin{bmatrix} x'_n \\ y'_n \\ z'_n \end{bmatrix}_F + s' \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}_F,$$

where $n = 1, 2, ..., and x_1 = s', y_1 = 0, z_1 = 0.$

Note that, since $s' = |\mathbf{r}'| = x_1$, the above formula is equivalent to

$$\mathbf{r}^{(n+1)} := \begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix}_F = \begin{bmatrix} x'_n \\ y'_n \\ z'_n \end{bmatrix}_F + x_1 \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}_F.$$
(2.9)

By expanding the matrix-vector multiplication, the flow of Frenet coordi-

nates above evolved as

$$\begin{aligned} x_{n+1} &= x'_n & -s' \cdot \kappa \cdot y_n \\ y_{n+1} &= y'_n &+ s' \cdot \kappa \cdot x_n & -s' \cdot \tau \cdot z_n \\ z_{n+1} &= z'_n & + s' \cdot \tau \cdot y_n \end{aligned}$$

that is, $\hat{\kappa}^{(n-2)}$ is passed from y_n to x_{n+1} and z_{n+1} . Torsion $\hat{\tau}^{(n-3)}$ spreads only to y_{n+1} in $\mathbf{r}^{(n+1)}$, and it is transferred to x_{n+2} later.

Because the derivatives depend on the curve parameter $t \in [0, 1]$, care has to be taken when the derivatives of the coordinates are to be computed. For example, because $\kappa = \hat{\kappa} \circ s$, it follows that $\kappa' = \hat{\kappa}' \circ s \cdot s'$ and so on.

An important result of (2.9) is that, compared to the arc-length parametrized case, arbitrary parametrization alters the geometry of the derivative vectors along the tangent first, and a new degree of freedom is presented by parametrization at each derivative along the tangent, by the presence of $s^{(n)}$ in x_n . Once these s', s'', ... tangential degrees of freedom are set, the tangential $x_1, x_2, ...,$ normal $y_2, y_3, ...$, and binormal $z_3, z_4, ...$ coordinates are uniquely determined, if the geometric invariants of the curve – the values and derivatives of curvature and torsion – are known.

Example: As an illustration of Theorem 2, let us compute the Frenet coordinates of the second and third derivatives of $\mathbf{r} : [0, 1] \to \mathbb{E}^3$.

Using the above formulation, the derivatives of $\mathbf{r}(t)$ are expressed as combinations of the derivatives of the arc-length parameterized derivatives $\hat{\mathbf{r}}^{(i)}(s)$ as follows:

$$\mathbf{r}' = (\hat{\mathbf{r}} \circ s)' \tag{2.10}$$
$$= \hat{\mathbf{r}}' \circ s \cdot s'$$

$$\mathbf{r}'' = (\hat{\mathbf{r}}' \circ s \cdot s')'$$

$$= \hat{\mathbf{r}}'' \circ s \cdot (s')^2 + \hat{\mathbf{r}}' \circ s \cdot s''$$
(2.11)

$$\mathbf{r}^{\prime\prime\prime} = (\hat{\mathbf{r}}^{\prime\prime} \circ s \cdot (s')^2 + \hat{\mathbf{r}}^{\prime} \circ s \cdot s'')'$$

$$= \hat{\mathbf{r}}^{\prime\prime\prime} \circ s \cdot (s')^3 + 3 \cdot \hat{\mathbf{r}}^{\prime\prime} \circ s \cdot s' s'' + \hat{\mathbf{r}}^{\prime} \circ s \cdot s'''$$
(2.12)

where all functions are evaluated at $t \in [0, 1]$, i.e. $\mathbf{r}' = \mathbf{r}'(t), s = s(t)$, etc.

Substituting (2.5)-(2.7) into (2.10)-(2.12), and using $s' = |\mathbf{r}'|$, one gets

$$\mathbf{r}' = \hat{\mathbf{t}} \circ s \cdot s' \tag{2.13}$$

$$\mathbf{r}'' = \hat{\kappa} \circ s \cdot \hat{\mathbf{n}} \circ s \cdot (s')^2 + \hat{\mathbf{t}} \circ s \cdot s''$$
(2.14)

$$\mathbf{r}''' = (\hat{\kappa}' \circ s \cdot \hat{\mathbf{n}} \circ s - \hat{\kappa}^2 \circ s \cdot \hat{\mathbf{t}} \circ s + \hat{\kappa} \circ s \cdot \hat{\tau} \circ s \cdot \hat{\mathbf{b}} \circ s) \cdot (s')^3 \qquad (2.15)$$
$$+ 3 \cdot \hat{\kappa} \circ s \cdot \hat{\mathbf{n}} \circ s \cdot s' s''$$
$$+ \hat{\mathbf{t}} \circ s \cdot s'''$$

The above equations show that the coordinates of the first, second, and third derivatives in the Frenet-frame are

$$\mathbf{r}' = \begin{bmatrix} s' \\ 0 \\ 0 \end{bmatrix}_{F}, \ \mathbf{r}'' = \begin{bmatrix} s'' \\ (s')^{2}\kappa \\ 0 \end{bmatrix}_{F}, \ \mathbf{r}''' = \begin{bmatrix} s''' - (s')^{3}\kappa^{2} \\ 3s's''\kappa + (s')^{3} \cdot \hat{\kappa}' \circ s \\ (s')^{3}\kappa\tau \end{bmatrix}_{F}$$
(2.16)

Using Theorem 2, the first derivative is $\mathbf{r}' = [s', 0, 0]_F^T$ by definition. The second derivative is

$$\mathbf{r}'' = \begin{bmatrix} (s')' \\ 0' \\ 0' \end{bmatrix}_{F} + s' \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \hat{\tau} & 0 \end{bmatrix} \begin{bmatrix} s' \\ 0 \\ 0 \end{bmatrix}_{F} \\ = \begin{bmatrix} s'' \\ 0 \\ 0 \end{bmatrix}_{F} + s' \begin{bmatrix} 0 \\ \kappa s' \\ 0 \end{bmatrix}_{F} = \begin{bmatrix} s'' \\ (s')^{2}\kappa \\ 0 \end{bmatrix}_{F}$$

and the third derivative becomes

$$\mathbf{r}''' = \begin{bmatrix} (s'')'\\ ((s')^{2}\kappa)'\\ 0' \end{bmatrix}_{F}^{+} s' \begin{bmatrix} 0 & -\kappa & 0\\ \kappa & 0 & -\tau\\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} s''\\ (s')^{2}\kappa\\ 0 \end{bmatrix}_{F}^{+}$$
$$= \begin{bmatrix} s'''\\ 2s's''\kappa + (s')^{2} \cdot (\hat{\kappa} \circ s)'\\ 0 \end{bmatrix}_{F}^{+} s' \begin{bmatrix} -(s')^{2}\kappa^{2}\\ s''\kappa\\ (s')^{2}\kappa\tau \end{bmatrix}_{F}^{+}$$
$$= \begin{bmatrix} s''' - (s')^{3}\kappa^{2}\\ 3s's''\kappa + (s')^{3} \cdot \hat{\kappa}' \circ s\\ (s')^{3}\kappa\tau \end{bmatrix}_{F}^{-}$$

These are equal to (2.16).

The first appearance of curvature is in y_2 , while the torsion is introduced in z_3 . The highest derivative of curvature is always in the y Frenet coordinate: $\hat{\kappa}^{(n-2)}$ in y_n . The highest derivative of torsion, $\hat{\tau}^{(n-3)}$, is in z_n .

Derivatives of κ and τ formally present in the various Frenet coordinates are summarized in the table below:

Frenet coordinate	κ up to	au up to
x_n	$\hat{\kappa}^{(n-3)}$	$\hat{\tau}^{(n-5)}$
y_n	$\hat{\kappa}^{(n-2)}$	$\hat{\tau}^{(n-4)}$
z_n	$\hat{\kappa}^{(n-3)}$	$\hat{ au}^{(n-3)}$

Please note that in any given derivative vector, the highest order derivatives of torsion originate from the differentiation of $z_3 = (s')^3 \kappa \tau$. So in any coordinate of $\mathbf{r}^{(n)}$, torsion derivative $\hat{\tau}^{(j)}, j \leq n-3$ is multiplied by an expression that contains $\hat{\kappa}^{(n-3-j)}$.

This, just like in the arc-length case, means that, while formally $\hat{\tau}^{(n-3)}$ is present in z_n , as soon as $\kappa = 0$, the actual value of $\hat{\tau}^{(n-3)}$ does not appear in z_n . No such masking occurs in the case of the highest derivatives of curvature, because they result from the differentiation of $y_2 = (s')^2 \kappa$.

Theorem 2 allows the reformulation of classic results as well. The following section demonstrates this by three examples.

2.2.2.1 Application of Frenet coordinate theorems

2.2.2.2 Computation of geometric invariants

New formulae, which rely only on the Frenet coordinates of derivatives, can be obtained from Theorem 2 to compute the geometric invariants, and their derivatives w.r.t. arc-length. To illustrate this, let us consider the computation of $\kappa, \tau, \hat{\kappa}'$.

This means, from (2.16), that the

$$s'(t) = |\mathbf{r}'(t)| = (\mathbf{r}'(t) \cdot \mathbf{r}'(t))^{\frac{1}{2}}$$
(2.17)

$$s''(t) = ((\mathbf{r}'(t) \cdot \mathbf{r}'(t))^{\frac{1}{2}})' = \frac{\mathbf{r}'(t) \cdot \mathbf{r}''(t)}{|\mathbf{r}'(t)|}$$
(2.18)

$$s'''(t) = \left(\frac{\mathbf{r}'(t) \cdot \mathbf{r}''(t)}{|\mathbf{r}'(t)|}\right)' = \frac{\mathbf{r}''(t) \cdot \mathbf{r}''(t) + \mathbf{r}'(t) \cdot \mathbf{r}'''(t)}{|\mathbf{r}'(t)|} - \frac{(\mathbf{r}'(t) \cdot \mathbf{r}''(t))^2}{|\mathbf{r}'(t)|^3} \quad (2.19)$$

derivatives of the arc-length function have to be expressed in terms of Frenet coordinates.

Lemma 3 The (2.17)-(2.19) derivatives of the arc-length function can be expressed with the Frenet-frame coordinates of the derivatives as

$$s'(t) = x_1$$
 (2.20)

$$s''(t) = x_2$$
 (2.21)

$$s'''(t) = \frac{y_2^2}{x_1} + x_3$$
(2.22)
= $\kappa^2 x_1^3 + x_3$

Proof. The proof can be found in Appendix A.1. ■

Frenet coordinates uniquely determine the geometric invariants of a curve, and provide simple means to compute them.

Lemma 4 Geometric invariants of a curve, up to order three, can be obtained

using Frenet coordinates as

$$\kappa = \frac{y_2}{x_1^2} \tag{2.23}$$

$$\hat{\kappa}' \circ s = \frac{y_3 - 3x_1 x_2 \kappa}{x_1^3} \tag{2.24}$$

$$=\frac{y_3x_1 - 3x_2y_2}{x_1^4} \tag{2.25}$$

$$\tau = \frac{z_3}{\kappa x_1^3} \tag{2.26}$$

$$=rac{z_3}{x_1y_2}$$
 (2.27)

Proof. The proof can be found in Appendix A.2.

Lemma 3 can be also used to reformulate (2.16) with geometric invariants and tangential coordinates only, by direct substitution:

$$\mathbf{r}' = \begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix}_F, \quad \mathbf{r}'' = \begin{bmatrix} x_2 \\ x_1^2 \kappa \\ 0 \end{bmatrix}_F, \quad \mathbf{r}''' = \begin{bmatrix} x_3 \\ 3x_1 x_2 \kappa + x_1^3 \cdot \hat{\kappa}' \circ s \\ x_1^3 \kappa \tau \end{bmatrix}_F \quad (2.28)$$

2.2.2.3 Conditions of G^n continuity

One of the most widespread definition of geometric continuity is due to [Farin 1992][Pottmann 1988]:

Definition 1 Two curves are $G^n, n \ge 1$ at a common point \mathbf{x} iff there exists a regular parametrization with respect to which they are C^n at \mathbf{x} .

An equivalent, and more geometric, definition for G^1 and G^2 connections have been recognized early on: G^1 is equivalent to the coincidence of tangent lines at **x**. G^2 is equivalent to the coincidence of osculating planes and signed curvatures of the curves at **x**, or, in other words, the coincidence of osculating circles.

In the literature, a dominant restriction of the above is to define G^1 as the coincidence of the *normalized tangent vectors* instead of the tangent lines, and define G^2 as the coincidence of the *Frenet-frames* of the curves and the (unsigned) *curvatures* at **x**. Compared to Definition 1, this means that only orientation preserving reparametrizations are allowed. **Definition 2** Two curves meeting at a point are G^n , $n \ge 1$ at a common point \mathbf{x} iff there exists an orientation preserving parameter transformation for both curves, such that they are C^n at \mathbf{x} after reparametrization.

By noting that, if two curves are C^r by some reparametrization, they are also C^r by arc-length parametrization, one can easily find that Definition 2 of geometric continuity is equivalent to the following:

Definition 3 Two curves are G^1 at a common point \mathbf{x} iff their tangent directions coincide. Two curves are $G^n, n \geq 2$ at a common point \mathbf{x} iff their Frenet-frames coincide at \mathbf{x} and the first n-2 derivatives of the curvature and the first n-3 derivatives of the torsion, all taken with respect to arc-length, are equal at \mathbf{x} , where $\kappa^{(-1)} = \tau^{(-1)} = \tau^{(-2)} = 0$.

The equivalence of definitions 3 and 2 follows from the fact that the Frenetframe coordinates of $\hat{\mathbf{r}}^{(n)}(s)$ are expressions in terms of the curvature and torsion functions, and their derivatives with respect to arc-length, up to order n-2 and n-3, respectively, as follows from Theorem 1. The Frenet-frame coincidence for $G^n, n \geq 2$ is imposed by the restriction to orientation preserving parameter transformations in Definition 2.

In this thesis, we use Definition 2 and 3 for the geometric continuity of curves.

A useful application of Theorem 2 is to cast the conditions of geometric continuity into a form, such that it only uses the Frenet coordinates of derivatives.

Let there be given two biregular parametric curves $\mathbf{r}(t), \mathbf{s}(t) : [0, 1] \to \mathbb{E}^3$, sharing a point $\mathbf{r}(1) = \mathbf{s}(0)$. If their unit tangent directions coincide, they form a G^1 join.

Lemma 5 Let us assume that the Frenet frames of $\mathbf{r}(t)$ and $\mathbf{s}(t)$ coincide at $\mathbf{r}(1) = \mathbf{s}(0)$, and let $[x_i, y_i, z_i]_F^T$ denote the Frenet coordinates of $\mathbf{r}^{(i)}(1)$, and $[\widetilde{x}_i, \widetilde{y}_i, \widetilde{z}_i]_F^T$ that of $\mathbf{s}^{(i)}(0)$. The two curves are G^2 iff

$$\frac{y_2}{\widetilde{y}_2} = \left(\frac{x_1}{\widetilde{x}_1}\right)^2$$

Furthermore, two G^2 curves are G^3 iff

$$\frac{\kappa\tau x_1^3}{\widetilde{\kappa}\widetilde{\tau}\widetilde{x}_1^3} = \left(\frac{x_1}{\widetilde{x}_1}\right)^3 , \quad \frac{y_3 - 3x_1x_2\kappa}{\widetilde{y}_3 - 3\widetilde{x}_1\widetilde{x}_2\widetilde{\kappa}} = \left(\frac{x_1}{\widetilde{x}_1}\right)^3$$

both hold.

Proof. The proof can be found in Appendix A.3.

With this formulation, deciding geometric continuity boils down to checking ratios of Frenet coordinate expressions, instead of finding reparametrizations.

2.2.2.4 β -splines

Checking geometric continuity of two joining curves by Definition 2 has been extensively studied in the literature. Using the chain rule of differentiation and formulating it in terms of *connection matrices* [Veltkamp] leads to β -splines [BarskyDeRose]. The β -conditions for two curves $\mathbf{s}(t), \mathbf{r}(t) : [0, 1] \rightarrow \mathbb{E}^3$ having a G^3 join at a common point $\mathbf{s}(0) = \mathbf{r}(1)$ are as follows

$$s(0) = r(1)$$

$$s'(0) = \beta_1 r'(1)$$

$$s''(0) = \beta_1^2 r''(1) + \beta_2 r'(1)$$

$$s'''(0) = \beta_1^3 r'''(1) + 3\beta_1 \beta_2 r''(1) + \beta_3 r'(1)$$

where the $\beta_i \in \mathbb{R}$ coefficients are the derivatives of an unknown reparametrization of $\mathbf{r}(t)$, from which $\mathbf{s}(t)$ originates. In other words, these β_i coefficients are degrees of freedom in connecting two curves geometrically continuously.

For each new derivative, a new degree of freedom is introduced along the tangent directions, as it is evident in the recurrence formula of Theorem 2.

To illustrate the connection between the β -conditions and the recurrence formula of Theorem 2, let $\mathbf{r}(t)$ and $\mathbf{s}(t)$ have the same Frenet frame and let $\mathbf{r}^{(i)}(1) = [x_i, y_i, z_i]_F^T$ and $\mathbf{s}^{(i)}(0) = [\widetilde{x}_i, \widetilde{y}_i, \widetilde{z}_i]_F^T$, i > 0. If the β conditions are fulfilled, these coordinates have the relation of

$$\mathbf{s}' = \begin{bmatrix} \widetilde{x}_1 \\ 0 \\ 0 \end{bmatrix}_F = \beta_1 \begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix}_F = \begin{bmatrix} \beta_1 x_1 \\ 0 \\ 0 \end{bmatrix}_F$$
$$\mathbf{s}'' = \begin{bmatrix} \widetilde{x}_2 \\ \widetilde{y}_2 \\ 0 \end{bmatrix}_F = \beta_1^2 \begin{bmatrix} x_2 \\ y_2 \\ 0 \end{bmatrix}_F + \beta_2 \begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix}_F = \begin{bmatrix} \beta_1^2 x_2 + \beta_2 x_1 \\ \beta_1^2 y_2 \\ 0 \end{bmatrix}_F$$
$$\mathbf{s}''' = \begin{bmatrix} \widetilde{x}_3 \\ \widetilde{y}_3 \\ \widetilde{z}_3 \end{bmatrix}_F = \beta_1^3 \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}_F + 3\beta_1\beta_2 \begin{bmatrix} x_2 \\ y_2 \\ 0 \end{bmatrix}_F + \beta_3 \begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix}_F = \begin{bmatrix} \beta_1^3 x_3 + 3\beta_1\beta_2 x_2 + \beta_3 x_1 \\ \beta_1^3 y_3 + 3\beta_1\beta_2 y_2 \\ \beta_1^3 z_3 \end{bmatrix}_F$$

It follows from (2.16) that $y_2 = \kappa x_1^2$ and $\tilde{y}_2 = \kappa \tilde{x}_1^2$. Due to the β -conditions, $\tilde{x}_1 = \beta_1 x_1$ holds, so $\tilde{y}_2 = \kappa \tilde{x}_1^2 = \kappa \beta_1^2 x_1^2 = \beta_1^2 y_2$.

Similarly, $z_3 = \kappa \tau x_1^3$ and $\tilde{z}_3 = \kappa \tau \tilde{x}_1^3$ and due to the β -conditions, $\tilde{z}_3 = \beta_1^3 z_3 = \beta_1^3 \kappa \tau x_1^3$.

In the case of y_3 and \tilde{y}_3 it is enough to note that if $\tilde{x}_1 = \beta_1 x_1$, $\tilde{x}_2 = \beta_1^2 x_2 + \beta_2 x_1$ hold, then from (2.16) $\tilde{y}_3 = 3\tilde{x}_1\tilde{x}_2\kappa + \tilde{x}_1^3 \cdot \hat{\kappa}' \circ \hat{s} = 3\beta_1 x_1(\beta_1^2 x_2 + \beta_2 x_1)\kappa + \beta_1^3 x_1^3 \cdot \hat{\kappa}' \circ \hat{s}$ should follow, which equals to the constraint from the β -conditions: $\tilde{y}_3 = \beta_1^3 y_3 + 3\beta_1\beta_2 y_2 = \beta_1^3(3x_1x_2\kappa + x_1^3 \cdot \hat{\kappa}' \circ \hat{s}) + 3\beta_1\beta_2 x_1^2\kappa$.

2.3 Formalization of geometric Hermite interpolation

I propose a general formulation of higher order geometric Hermite interpolation in this section. For the sake of simplicity, we first focus on symmetric cases, i.e. when prescribed geometric data are of the same type and order at each knot.

Let us represent these geometric data by tuples

$$\mathbf{D}_i = (\mathbf{p}_i; \mathbf{t}_i; \kappa_i, \mathbf{n}_i, \mathbf{b}_i; \tau_i, \hat{\kappa}'_i; ..).$$

where $\hat{\kappa}'_i$ denotes the value of the derivative of curvature with respect to arc-

length at an unspecified parameter value. This value is to be reconstructed by a curve. From now on, this notation will be used to denote values of derivatives of curvature and torsion with respect to arc-length as well.

An *n*th order GH data tuple consists of position $\mathbf{p} \in \mathbb{E}^3$, vectors of the Frenet frame (**t**, **n**, **b**), curvature κ_i , torsion τ_i , and derivatives of κ_i, τ_i up to n-2 and n-3 w.r.t. arc-length. The following table summarizes these data, where order *n* refers to the order of GH interpolation:

order (n)01234..ndata (
$$\mathbf{D}_i$$
) \mathbf{p} \mathbf{t} \mathbf{n}, κ $\hat{\kappa}', \tau$ $\hat{\kappa}'', \hat{\tau}'$.. $\hat{\kappa}^{(n-2)}, \hat{\tau}^{(n-3)}$

Note that if $n \ge 2$, by $\mathbf{b} = \mathbf{t} \times \mathbf{n}$, the Frenet frame is specified. Furthermore, each new order constrains two scalar degrees of freedom.

Let there be given two *n*th order GH data tuples $\mathbf{D}_0, \mathbf{D}_1$. Let $\mathbf{q}(t) : [0, 1] \rightarrow \mathbb{E}^3$ be a parametric curve, defined by control data $\mathbf{q}_i \in \mathbb{R}^3 \vee \mathbb{E}^3$, i = 0, ..., m in some, not necessarily polynomial, basis $F_0(t), ..., F_m(t) : [0, 1] \rightarrow \mathbb{R}$ such that

$$\mathbf{q}(t) = \sum_{j=0}^{m} \mathbf{q}_j F_j(t) \; .$$

Please note that not all \mathbf{q}_j need to be Euclidean points. For example, in the case of classic Hermite interpolation, apart from the two endpoints, all data are derivatives, i.e. vectors in \mathbb{R}^3 . We refer to the $F_j(t), j = 0, .., n$ functions as the order (m + 1) basis functions.

First, let us set up the system of equations that a parametric curve has to satisfy, so that it reconstructs a pair of given GH data tuples.

Let us suppose, without loss of generality, that GH data are to be reconstructed at parametric endpoints t = 0 and t = 1. The derivatives of the curve at these points are

$$\begin{bmatrix} \mathbf{q}(0) \\ \mathbf{q}'(0) \\ \mathbf{q}''(0) \\ \cdots \\ \mathbf{q}(1) \\ \mathbf{q}''(1) \\ \mathbf{q}''(1) \\ \cdots \end{bmatrix} = \begin{bmatrix} F_0(0) & F_1(0) & \dots & F_m(0) \\ F_0'(0) & F_1'(0) & \dots & F_m'(0) \\ \cdots & \cdots & \cdots & \cdots \\ F_0(1) & F_1(1) & \dots & F_m(1) \\ F_0'(1) & F_1'(1) & \dots & F_m'(1) \\ F_0''(1) & F_1''(1) & \dots & F_m''(1) \\ \cdots & \cdots & \cdots & \cdots \\ F \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \\ \cdots \\ \mathbf{q}_m \end{bmatrix}}_{\mathbf{q}}$$

Please note that the vectors in the above equation are actually matrices, composed of the row vectors of $\mathbf{q}(0)^T, ..., \mathbf{q}_0^T$ The transposition symbols are omitted from the equations, for the sake of simplicity.

If left-hand side Frenet coordinates satisfy the conditions of Theorem 2, that is, using (2.28),

$\mathbf{q}(0)$		1	0	0	0	0	0	0	0	\mathbf{p}_0
$\mathbf{q}'(0)$		0	$x_1(0)$	0	0	0	0	0	0	\mathbf{t}_0
$\mathbf{q}''(0)$		0	$x_2(0)$	$\kappa_0 x_1^2(0)$	0	0	0	0	0	$ \mathbf{n}_0 $
	_									\mathbf{b}_0
$\mathbf{q}(1)$		0	0	0	0	1	0	0	0	\mathbf{p}_1
$\mathbf{q}'(1)$		0	0	0	0	0	$x_1(1)$	0	0	\mathbf{t}_1
$\mathbf{q}''(1)$		0	0	0	0	0	$x_2(1)$	$\kappa_1 x_1^2(1)$	0	$ \mathbf{n}_1 $
L		[]	$\begin{bmatrix} \mathbf{b}_1 \end{bmatrix}$
$F \cdot \mathbf{q}$ $G(x_1(0),,x_n(0),x_1(1),,x_n(1))$						$f \longrightarrow_{f}$				

or in short

 $F \cdot \mathbf{q} = G(x_1(0), ..., x_n(0), x_1(1), ..., x_n(1)) \cdot \mathbf{f}$ (2.29)

holds, then the curve defined by control data $\mathbf{q}_0, .., \mathbf{q}_m$ reconstructs the GH data tuples $\mathbf{D}_0, \mathbf{D}_1$.

Here, the Frenet x coordinate functions $x_i(t)$ of $\mathbf{q}^{(i)}(t)$ are evaluated at t = 0 and t = 1. These functions are not subject to geometric reconstruction constraints, and they can take on arbitrary values, as long as $x_1(0), x_1(1) > 0$ holds, which ensures the regularity of parametrization at endpoints.

For the sake of simplicity – and to emphasize their independence –, let us

omit the evaluation parameters from the notation and let the particular values of $x_i(0)$ and $x_i(1)$ on the right hand side of (2.29) be denoted by \overrightarrow{x}_i and \overleftarrow{x}_i , respectively.

The left-hand side of (2.29) consists of endpoint positions and derivatives of $\mathbf{q}(t)$. The right-hand side is made up of points and derivative vectors of an unknown curve that reconstructs \mathbf{D}_0 and \mathbf{D}_1 exactly. This curve has a particular parametrization, specified by $\overrightarrow{x}_i, \overleftarrow{x}_i$, which also influences the overall shape of the curve. Our aim is to match the right-hand side data with $\mathbf{q}(t)$.

Let us now investigate how many derivatives need to be constrained in (2.29) so that the given geometric data \mathbf{D}_0 , \mathbf{D}_1 are reconstructed.

If $\kappa_0, \kappa_1 \neq 0$, constraining the position and the first *n* derivatives is all that is required for reconstruction. In this case, κ_i, τ_i , and their (n-2) and (n-3)derivatives w.r.t. arc-length appear with non-zero coefficients in the first *n* derivatives of $\mathbf{q}(t)$.

If $\kappa_0 = 0$, the reconstruction of torsion and its derivatives requires more than *n* derivatives, as it was noted at the end of the previous section. To illustrate this, let us consider the problem of third order GH interpolation with $\kappa_0 = 0$. According to (2.16), the third derivative is then

$$\mathbf{r}^{\prime\prime\prime} = \begin{bmatrix} s^{\prime\prime\prime} \\ (s^{\prime})^3 \hat{\kappa}^{\prime} \\ 0 \end{bmatrix}_F ,$$

which is clearly in the osculating plane, and the torsion of the curve cannot be computed from it. If, additionally, $\hat{\kappa}' \neq 0$, Frenet coordinates of the fourth derivative are

$$\mathbf{r}^{(4)} = \begin{bmatrix} s^{(4)} \\ 6(s')^2 s'' \hat{\kappa}' + (s')^4 \hat{\kappa}'' \\ 2(s')^4 \hat{\kappa}' \tau \end{bmatrix}_F,$$

from which the torsion can be expressed. So the reconstruction of third order GH data constrains the first four derivatives of the curve at t = 0 if $\kappa_0 = 0$ and $\hat{\kappa}'_0 \neq 0$.

More generally, as a consequence of the discussion at the end of the previous section, the following can be stated:

Lemma 6 If $\kappa_i = ... = \hat{\kappa}_i^{(h_i-1)} = 0, \hat{\kappa}_i^{(h_i)} \neq 0, i = 0, 1$, the reconstruction of

prescribed nth order GH data constrains $n + h_0$ derivatives at t = 0 and $n + h_1$ derivatives at t = 1.

If $\kappa_0, \kappa_1 \neq 0$, the dimensions of the matrices in (2.29) are $F \in \mathbb{R}^{2(n+1)\times(m+1)}$, $G \in \mathbb{R}^{2(n+1)\times 8}, \mathbf{q} \in \mathbb{R}^{(m+1)\times 3}, \mathbf{f} \in \mathbb{R}^{8\times 3}$.

In general, the sizes of G, \mathbf{q} , and \mathbf{f} depend on the dimension of the space, as well. In a *d*-dimensional space there are d-1 scalar curvatures [51], and a local coordinate system of the curve consists of *d* curvature vectors at a position, that is $F \in \mathbb{R}^{2(n+1)\times(m+1)}$, $G \in \mathbb{R}^{2(n+1)\times 2(d+1)}$, $\mathbf{q} \in \mathbb{R}^{(m+1)\times d}$, and $\mathbf{f} \in \mathbb{R}^{2(d+1)\times d}$.

If one or more curvatures vanish, and the GH data tuples $\mathbf{D}_0, \mathbf{D}_1$ satisfy Lemma 6, the matrices are of dimensions $F \in \mathbb{R}^{(2n+2+h_0+h_1)\times(m+1)}, G \in \mathbb{R}^{(2n+2+h_0+h_1)\times 8}, \mathbf{q} \in \mathbb{R}^{(m+1)\times 3}, \mathbf{f} \in \mathbb{R}^{8\times 3}$ in the Euclidean space, and of dimensions $F \in \mathbb{R}^{(2n+2+h_0+h_1)\times(m+1)}, G \in \mathbb{R}^{(2n+2+h_0+h_1)\times 2(d+1)}, \mathbf{q} \in \mathbb{R}^{(m+1)\times d}$, and $\mathbf{f} \in \mathbb{R}^{2(d+1)\times d}$ in a *d*-dimensional space.

The tangential coordinates $\overrightarrow{x}_i, \overleftarrow{x}_i$ in (2.29) are degrees of freedom in GH reconstruction. They determine the *parametrization* of the curve, while equations of (2.29) specify the *geometry* of $\mathbf{q}(t)$.

2.4 Solving the reconstruction equations

This section focuses on solving geometric Hermite interpolation via (2.29) for a fixed parametrization at knots. In this case, geometric reconstruction constraints form a system of linear equations.

I present new existence conditions for exact reconstruction and show how approximate solutions are computed in various norms, all these relying on the formulation presented in the previous section.

It is important to note that by setting the scalar degrees of freedom of parametrization, one carries out a kind of Hermite interpolation of the $s'(t) = |\mathbf{r}'(t)|$ derivative of the arc-length function in the sense that every exact solution has the set parametric speed magnitude, acceleration, and higher derivatives at knots.

From this point of view, geometric Hermite interpolation allows us to independently reconstruct geometric properties and specify the parametric speed via an Hermite-like interpolation at knots.

2.4.1 Exact solutions

For now, let us assume that the right-hand side tangential parameters \vec{x}_i, \dot{x}_i in (2.29) are fixed, i.e. the exact derivative vectors are given up to the necessary order, specified by Lemma 6. Let us denote the total number of equations by $\bar{k} = 2n + 2 + h_0 + h_1$ and let us consider the computation of control data of curve $\mathbf{q}(t)$ from equation (2.29).

If k < m+1, (2.29) is an underdetermined linear system that has infinitely many solutions. The remaining degrees of freedom, not constrained by GH interpolation, are available for the optimization of the curve, for example, by minimizing an energy or other functionals.

If F is square and of full rank, the control points are computed as

$$\mathbf{q} = F^{-1} \cdot G \cdot \mathbf{f}.$$

In this case, $m = \overline{k} - 1$ should hold, and the row vectors of F – i.e. values of basis functions and their derivatives at the endpoints – should be linearly independent. The latter is the case for all 'reasonable' basis functions of appropriate order, for example, power and Bernstein basis polynomials of degree at least $\overline{k} - 1$. In general, i.e. including overdetermined cases, we assume that F is full rank. We refer to these as *linearly independent bases*.

In the case of polynomial bases, the next statement is the consequence of Lemma 6 and Theorem 2.

Theorem 7 If $\kappa_i = ... = \hat{\kappa}_i^{(h_i-1)} = 0$, $\hat{\kappa}_i^{(h_i)} \neq 0$, i = 0, 1, there is always a degree $(2n+1+h_0+h_1)$ polynomial solution to the nth order GH interpolation problem.

Proof. All derivatives of curvature and torsion listed in \mathbf{D}_0 and \mathbf{D}_1 appear in the coordinates of the derivatives of $\mathbf{q}(t)$ with either non-zero or unspecified coefficients.

The unspecified coefficients consist of curvature derivatives that are not listed in \mathbf{D}_0 and \mathbf{D}_1 , and of powers of $s' \neq 0$. By setting them to arbitrary non-zero values, the resulting equation constrains all derivatives of curvature and torsion specified in \mathbf{D}_0 and \mathbf{D}_1 .

Since the degree of the polynomial is $(2n + 1 + h_0 + h_1)$, matrix F is invertible, and control data of the curve are computed as $\mathbf{q} = F^{-1} \cdot G \cdot \mathbf{f}$.

In the worst case, if all the specified curvatures vanish at both endpoints, the degree of a polynomial solution to symmetric *n*-th order GH interpolation might be as high as 2n + 1 + 2(n - 1) = 4n - 1. On the contrary, there is always a degree 2n + 1 polynomial solution if $\kappa_0 \neq 0$ and $\kappa_1 \neq 0$.

Moreover, a lower degree polynomial solution may exist for appropriate input data and selection of $\overrightarrow{x}_i, \overleftarrow{x}_i$. We discuss this in more detail next.

Let us suppose that F cannot be inverted, and investigate the conditions for the existence of an exact solution. Let us assume that $rank(F) \ge \min\{m + 1, \overline{k}\}.$

The overdetermined case $\overline{k} > m + 1$ usually does not provide an exact solution for fixed $\overrightarrow{x}_i, \overleftarrow{x}_i$. However, if these tangential parameters may change, the additional $\overline{k} - 2$ scalar degrees of freedom in (2.29) may help us to carry out exact reconstruction.

Mørken studied the conditions of such a reconstruction algebraically in [35], while Schaback investigated this problem from a geometric point of view in [41]. My formulation, presented here, allows us to utilize nonlinear optimization techniques to compute a lower-degree solution to a GH interpolation problem.

In general, that is, for arbitrary bases and orders, by the Rouché-Capelli theorem, an exact solution to (2.29) exists if and only if every column of $G \cdot \mathbf{f}$ is in the column space, C(F), of F, i.e. if $rank(F) = rank([F, G \cdot \mathbf{f}])$, where $[A, \mathbf{b}]$ denotes the augmented matrix of A by appending the column vector \mathbf{b} to A.

If $\overrightarrow{x}_i, \overleftarrow{x}_i$ may vary, an overdetermined exact solution exists if and only if all d columns of $G(\overrightarrow{x}_1, ..., \overleftarrow{x}_1, ...) \cdot \mathbf{f}$ are in C(F) for some $\overrightarrow{x}_1, \overleftarrow{x}_1 > 0, \overrightarrow{x}_2, \overleftarrow{x}_2, ... \in \mathbb{R}$.

Let $\mathbf{C} := \times_{i=1}^{d} C(F) \subset \mathbb{R}^{\overline{k} \times d}$. On the one hand, C(F) and \mathbf{C} are linear subspaces of $\mathbb{R}^{\overline{k}}$ and $\mathbb{R}^{\overline{k} \times d}$. On the other hand, $G(\overrightarrow{x}_1, ..., \overleftarrow{x}_1, ...) \cdot \mathbf{f}$ forms a – generally nonlinear – subset of $\mathbb{R}^{\overline{k} \times d}$ as the $\overrightarrow{x}_i, \overleftarrow{x}_i$ vary (for each d coordinate of the unknown control points). Let \mathbf{G} denote all points of this nonlinear subspace, that is, let

$$\mathbf{G} := \{ \mathbf{x} \in \mathbb{R}^{\overline{k} \times d} | \exists \overrightarrow{x}_1, \overleftarrow{x}_1 > 0, \overrightarrow{x}_2, .., \overleftarrow{x}_2, .. \in \mathbb{R} : \mathbf{x} = G(\overrightarrow{x}_1, .., \overleftarrow{x}_1, ..) \cdot \mathbf{f} \} .$$

Then, using the above notations, the existence conditions are formulated in accordance with the following **Theorem 8** A solution to the n-th order GH interpolation problem with (m + 1) control data exists in a linearly independent basis if and only if $\mathbf{C} \cap \mathbf{G} \neq \emptyset$.

Let A[i] denote the *i*-th column of the argument matrix. The above condition can be interpreted as the requirement of the existence of a $\mathbf{g} \in \mathbf{G}$ such that

$$\forall i \in [0, d-1] : C(F) \cap \mathbf{g}[i] \neq \emptyset$$

If this condition of Theorem 8 is satisfied, any point of the intersection $\mathbf{C} \cap \mathbf{G}$ results in exact reconstruction of GH data tuples $\mathbf{D}_0, \mathbf{D}_1$, as well as the reconstruction of parametric speed defined by $\overrightarrow{x}_i, \overleftarrow{x}_i$ at endpoints.

Any $\mathbf{g} \in \mathbf{G}$ defines a choice of $\overrightarrow{x}_1, \overleftarrow{x}_1 > 0, \overrightarrow{x}_2, ..., \overleftarrow{x}_2, ... \in \mathbb{R}$ values for the parametrization degrees of freedom.

Example: To illustrate the use of Theorem 8, let us consider the first order GH interpolation problem of reconstructing

$$\mathbf{D}_i = (\mathbf{p}_i; \mathbf{t}_i) , i = 0, 1$$

with a parabola

$$\mathbf{b}(t) = (1-t)^2 \mathbf{b}_0 + 2t(1-t)\mathbf{b}_1 + t^2 \mathbf{b}_2$$

in Bézier form.

By choosing $\mathbf{b}_0 = \mathbf{p}_0$, $\mathbf{b}_2 = \mathbf{p}_1$, the end position data are reconstructed. If additionally, the intersection of $\mathbf{p}_0 + t\mathbf{t}_0$ and $\mathbf{p}_1 - s\mathbf{t}_1$, t, s > 0 exist, by choosing it as \mathbf{b}_1 , the reconstruction of tangents is carried out too.

On the other hand, the reconstruction equations take the form of

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & 2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \overrightarrow{x}_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \overleftarrow{x}_1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{t}_0 \\ \mathbf{p}_1 \\ \mathbf{t}_1 \end{bmatrix}$$
(2.30)

from which it follows that

$$C(F) = \left\{ \alpha \begin{bmatrix} 1\\-2\\0\\0\\0 \end{bmatrix} + \beta \begin{bmatrix} 0\\2\\0\\-2 \end{bmatrix} + \gamma \begin{bmatrix} 0\\0\\1\\2\\2 \end{bmatrix} = \begin{bmatrix} \alpha\\-2\alpha+2\beta\\\gamma\\-2\beta+2\gamma \end{bmatrix} \middle| \alpha, \beta, \gamma \in \mathbb{R} \right\}$$

Given actual input data tuples $\mathbf{D}_0, \mathbf{D}_1$, all there is left to do is to check all *i* columns of the right hand side **g** of (2.30) so that there exist $\alpha, \beta, \gamma \in \mathbb{R}$, for each column, independently, such that $\mathbf{g}[i] \in C(F)$.

Let us now consider a numerical example:

$$\mathbf{p}_0 = \begin{bmatrix} 0\\1 \end{bmatrix} , \ \mathbf{t}_0 = \begin{bmatrix} 1\\0 \end{bmatrix} ,$$
$$\mathbf{p}_1 = \begin{bmatrix} 1\\0 \end{bmatrix} , \ \mathbf{t}_1 = \begin{bmatrix} 0\\-1 \end{bmatrix} .$$

The right hand side of (2.30) is

$$\mathbf{g} = \begin{bmatrix} 0 & 1 \\ \overrightarrow{x}_1 & 0 \\ 1 & 0 \\ 0 & -\overleftarrow{x}_1 \end{bmatrix}$$

and for $\mathbf{g}[0] \in C(F)$ the following should hold, for some $\alpha, \beta, \gamma \in \mathbb{R}$:

$$\begin{bmatrix} \alpha \\ -2\alpha + 2\beta \\ \gamma \\ -2\beta + 2\gamma \end{bmatrix} = \begin{bmatrix} 0 \\ \overrightarrow{x}_1 \\ 1 \\ 0 \end{bmatrix}$$

that is, the x coordinates of the $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ control points are

$$\alpha = 0 \ , \ \beta = 1 \ , \ \gamma = 1$$

and so

 $\overrightarrow{x}_1 = 2$.
Similarly, for $\mathbf{g}[1] \in C(F)$ to hold, $\alpha, \beta, \gamma \in \mathbb{R}$ should be such that

$$\begin{bmatrix} \alpha \\ -2\alpha + 2\beta \\ \gamma \\ -2\beta + 2\gamma \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -\overleftarrow{x}_1 \end{bmatrix}$$

that is, the y coordinates of the $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ control points are

$$\alpha = 1$$
 , $\beta = 1$, $\gamma = 0$

 \mathbf{SO}

 $\overleftarrow{x}_1 = 2$.

From this, the $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ control points are

$$\mathbf{b}_0 = \begin{bmatrix} 0\\1 \end{bmatrix} , \ \mathbf{b}_1 = \begin{bmatrix} 1\\1 \end{bmatrix} , \ \mathbf{b}_2 = \begin{bmatrix} 1\\0 \end{bmatrix} .$$

2.4.2 Approximate solutions

If conditions of Theorem 8 do not hold, a best approximate solution has to be found. In general, measuring how good an approximation a given $\mathbf{q}(t)$ is to the input tuples $\mathbf{D}_0, \mathbf{D}_1$ can be formulated as minimizing a functional $f(\mathbf{q}, \mathbf{D}_0, \mathbf{D}_1) \in \mathbb{R}$, where $\mathbf{q} = [\mathbf{q}_0, ..., \mathbf{q}_m]^T$ depends on $\overrightarrow{x}_i, \overleftarrow{x}_i$. Two types of these functionals are considered here.

Norm functionals are used to find \mathbf{q} such that $||F \cdot \mathbf{q} - G \cdot \mathbf{f}||$ is minimal in some norm $|| \cdot ||$. These are computationally less demanding, and the subject of approximations in $||.||_p$ norms has been established and extensively studied in the literature.

Taylor expansion based functionals are used to find a curve $\mathbf{q}(t)$ such that its Taylor expansion at endpoints deviates the least from the two polynomials having derivatives corresponding to the right-hand side of (2.29), for given $\overrightarrow{x}_{i}, \overleftarrow{x}_{i}$.

This can be done such that the parameterization of the end-point polynomials (and hence $\overrightarrow{x}_i, \overleftarrow{x}_i$) are also taken into account, or the approximation can be restricted to the geometric invariants only by using 'geometric' Taylor expansions (in other words, degree n parabolas, as Boehm referred to these in [4]).

For the sake of simplicity, let us assume that $\kappa_0, \kappa_1 \neq 0$.

2.4.2.1 Norm functionals

If $\overrightarrow{x}_i, \overleftarrow{x}_i \in \mathbb{R}, \ \overrightarrow{x}_1, \overleftarrow{x}_1 > 0$ are fixed, the minimization problem

$$\min_{\mathbf{q}_{0},\dots,\mathbf{q}_{m}} ||G \cdot \mathbf{f} - F \cdot \mathbf{q}||$$
(2.31)

is a convex optimization problem for any norm, since it is linear and the objective function is convex [5]. The latter ensures that any local minimum is also a global minimum, i.e. any algorithm that is guaranteed to converge to a local extrema will also find the global one.

The expression $||\mathbf{x}||_p = (x_1^p + x_2^p + ... + x_d^p)^{\frac{1}{p}}$ is a norm for $p \ge 1$ and $p = \infty$. In practice, the three most commonly employed $||\cdot||_p$ norms are the $p = 1, 2, \infty$ norms.

The p = 2 norm is a particularly attractive choice for computational reasons. In this case, a concise expression of the approximate solution, which is a curve having derivatives closest to the right-hand side vectors in the least squares (LSQ) sense, is available.

It is known from linear algebra, that the LSQ solution to the overdetermined system of linear equations $A \cdot \mathbf{x} = \mathbf{b}$ is $\mathbf{x} \approx A^+ \cdot \mathbf{b}$, where A^+ denotes the Moore-Penrose pseudo-inverse of matrix A. The pseudo-inverse is expressed in closed form as $A^+ = (A^T \cdot A)^{-1}A^T$. Even if $(A^T \cdot A)$ is singular, A^+ can be computed by singular value decomposition (SVD) of $A^T \cdot A$.

The LSQ solution minimizes $||F \cdot \mathbf{q} - G \cdot \mathbf{f}||_2^2$, and the control data is found by

$$\mathbf{q} \approx (F^T \cdot F)^{-1} \cdot F^T \cdot G \cdot \mathbf{f}$$

assuming that rank(F) = m + 1.

This **q** minimizes the sum of squared Euclidean distances of the constraint positions and derivatives from the positions and derivatives of curve $\mathbf{q}(t)$, i.e. the right-hand side and left-hand side of (2.29), respectively.

Then a weighting can be readily incorporated into the solution by

$$\mathbf{q} \approx (F^T \cdot W^T \cdot W \cdot F)^{-1} \cdot F^T \cdot W^T \cdot W \cdot G \cdot \mathbf{f} ,$$

where $W = diag(w_{0,0}, ..., w_{0,n}, w_{1,0}, ..., w_{1,n})$ is a diagonal matrix of dimensions $(2n+2) \times (2n+2)$. Each $w_{i,j} > 0$ can be considered as the relative importance of the precise reconstruction of the correct *j*-th derivative at t = i, i = 0, 1, j = 0, 1, ..., n.

The advantage of the L^2 norm is that the solution can be expressed in closed form, however, for certain applications other norms are preferred.

If a small number of input data is expected to be unreliable, the p = 1 norm is often chosen, which amounts to the minimization of the sum of error magnitudes. It provides a more robust, less sensitive solution to error in input [3]. Minimizing the $|| \cdot ||_1$ norm of $F \cdot \mathbf{q} - G \cdot \mathbf{f}$ is an unconstrained, convex optimization problem in the form of

$$\min_{\mathbf{q}_{0},\dots,\mathbf{q}_{m}} ||G \cdot \mathbf{f} - F \cdot \mathbf{q}||_{1} .$$
(2.32)

Transforming (2.32) into a canonical linear programming (LP) form allows us to use the simplex method to find the global optimum. To do that, let $\mathbf{y}^+ - \mathbf{y}^- = G \cdot \mathbf{f} - F \cdot \mathbf{q}$ and $\mathbf{q}^+ - \mathbf{q}^- = \mathbf{q}$ such that all components of the unknown vectors are non-negative: $\mathbf{q}^+, \mathbf{q}^-, \mathbf{y}^+, \mathbf{y}^- \ge \mathbf{0}$.

Then the following LP problem is equivalent to (2.32):

$$\min \mathbf{y}^{+} + \mathbf{y}^{-}$$

$$F \cdot \mathbf{q}^{+} - F \cdot \mathbf{q}^{-} + \mathbf{y}^{+} - \mathbf{y}^{-} = G \cdot \mathbf{f}$$

$$\mathbf{y}^{+}, \mathbf{y}^{-}, \mathbf{q}^{+}, \mathbf{q}^{-} \ge 0,$$

$$i = 0, ..., 2n + 1, \ j = 0, ..., m$$

$$(2.33)$$

The simplex method can be applied to (2.33) to find the optimal 2(m+1) + 2(2n+1) unknowns of (2.33) for each *d* coordinates. Coordinates of control points corresponding to an optimal solution are computed by $\mathbf{q} = \mathbf{q}^+ - \mathbf{q}^-$.

It has been shown in [55], that for larger problems, the dual simplex method performs better. Other, more efficient modifications of the primal and dual simplex methods have been also proposed for minimizing the L^1 error [47], [1]. Cadzow also proposed an algorithm in [5] to minimize the sum of error magnitudes and to find an approximate solution with minimal largest error magnitude, i.e. a best approximation in $|| \cdot ||_{\infty}$.

2.4.2.2 Taylor expansion based functionals

Let us partition (2.29) such that

$$G = \begin{bmatrix} G_0 & 0 \\ 0 & G_1 \end{bmatrix}, \ \mathbf{f} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{f}_1 \end{bmatrix}, \ F = \begin{bmatrix} F_0 \\ F_1 \end{bmatrix},$$

where F_i correspond to basis function value and derivative evaluations at t = i, matrices G_i contain the reconstruction equations, and $\mathbf{f}_i = [\mathbf{p}_i, \mathbf{t}_i, \mathbf{n}_i, \mathbf{b}_i]^T$, i = 0, 1.

Let $T_k(\mathbf{x})(t)$ denote the degree k Taylor expansion operator

$$T_k\begin{pmatrix} \begin{bmatrix} \mathbf{x}_0 \\ .. \\ \mathbf{x}_n \end{bmatrix} (t) = \sum_{i=0}^k \frac{\mathbf{x}_i}{i!} t^i, \ t \in \mathbb{R}, k \le n.$$

Taylor expansions of $\mathbf{q}(t)$ at t = i, i = 0, 1 are then

$$T_k(F_i \cdot \mathbf{q})(t)$$
,

while the right-hand side derivatives in (2.29) can be expanded as a polynomial by

$$T_k(G_i \cdot \mathbf{f}_i)(t)$$
,

which is the expansion of an unknown curve that reconstructs both the geometry specified by $\mathbf{D}_0, \mathbf{D}_1$ and the parametrization determined by $\overrightarrow{x}_i, \overleftarrow{x}_i$.

Let the difference of these two polynomials be denoted by

$$\mathbf{e}_{i}(t) = T_{n}(F_{i} \cdot \mathbf{q})(t) - T_{n}(G_{i} \cdot \mathbf{f}_{i})(t)$$
$$= T_{n}(F_{i} \cdot \mathbf{q} - G_{i} \cdot \mathbf{f}_{i})(t),$$

and consider the functionals

$$e_i(\mathbf{q}_0,..,\mathbf{q}_m) = \int_0^1 ||\mathbf{e}_i(t)|| dt , i = 0,1 ,$$

in some norm $|| \cdot ||$.

Finding control data such that

$$\min_{\mathbf{q}_0,\dots,\mathbf{q}_m} e_0(t) + e_1(t)$$

accounts for the reconstruction of both the geometric invariants and the parametrization of the unknown curve. The latter depends on the choice of $\overrightarrow{x}_i, \overleftarrow{x}_i$ via matrix G.

Detaching parametrization from the functional – so that the optimality of the approximation depends only on the geometry defined by $\mathbf{D}_0, \mathbf{D}_1$ – can be done by using a Taylor-like polynomial expansion of the curves.

Let a degree n geometric Taylor polynomial be such that it has a contact of order n + 1 at t = 0 with a curve and all of its regular reparametrizations.

For example, a second and third degree geometric Taylor polynomial at t = i, i = 0, 1 can be written as

$$\mathbf{p}_{2}(t) = \mathbf{p}_{i} + \begin{bmatrix} t\\ \frac{\kappa}{2}t^{2}\\ 0 \end{bmatrix}_{\mathbf{f}_{i}}, \ \mathbf{p}_{3}(t) = \mathbf{p}_{i} + \begin{bmatrix} t\\ \frac{\kappa}{2}t^{2} + \frac{\hat{\kappa}'}{6}t^{3}\\ \frac{\kappa\tau}{6}t^{3} \end{bmatrix}_{\mathbf{f}_{i}}$$

Let us use the error functional

$$\widetilde{e}(t) = \widetilde{e_0}(t) + \widetilde{e_1}(t) ,$$

using

$$\widetilde{e}_i(t) = \int_0^1 ||(\mathbf{p}_i - \mathbf{q}(i)) + \left(\begin{bmatrix} 1\\0\\0 \end{bmatrix}_{\mathbf{f}_i} - \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{F^{\mathbf{q}(i)}} \right) \cdot t + \left(\begin{bmatrix} 0\\\kappa_i\\0 \end{bmatrix}_{\mathbf{f}_i} - \begin{bmatrix} 0\\\frac{y_2}{x_1^2}\\0 \end{bmatrix}_{F^{\mathbf{q}(i)}} \right) \cdot \frac{t^2}{2} + \dots ||dt|,$$

where $F^{\mathbf{q}(i)}$ denotes the Frenet frame of $\mathbf{q}(t)$ at t = i. The computation of invariants of $\mathbf{q}(t)$ are done using (2.23)-(2.26).

2.4.3 Symmetric interpolation constraints

A drawback of the approximation methods shown previously is that they cannot guarantee exact reconstruction of any of the input geometric quantities. For example, not even weighted LSQ can be used to guarantee position reconstruction.

Imposing partial exact reconstruction conditions, i.e. exact reconstruction of certain quantities of the input geometric data, transforms the convex optimization problem of (2.29) into a constrained nonlinear optimization problem, where the constraints will not necessarily form a convex subset of the problem space - the solution to these usually require more computationally intensive and complex algorithms.

Instead of using these, an alternative approach can be taken if we are only interested in incorporating symmetric partial reconstruction. To illustrate this, let us consider the problem of stipulating exact position reconstruction at both endpoints.

For the sake of simplicity, let us also assume that a best approximation is to be found with respect to an arbitrary norm. Formally, this is equivalent to

$$\min_{\mathbf{q}_0,..,\mathbf{q}_n} ||F \cdot \mathbf{q} - G \cdot \mathbf{f}||$$

$$F_0(0)\mathbf{q}_0 + .. + F_n(0)\mathbf{q}_n = \mathbf{p}_0$$

$$F_0(1)\mathbf{q}_0 + .. + F_n(1)\mathbf{q}_n = \mathbf{p}_1$$

Note that all zero order (positional) constraints of (2.29) are automatically satisfied for any feasible solution, that is, these constraints can be removed from (2.29), yielding

$$\begin{bmatrix} F'_{0}(0) & \dots & F'_{n}(0) \\ F''_{0}(0) & \dots & F''_{n}(0) \\ \dots & \dots & \dots \\ F'_{0}(1) & \dots & F'_{n}(1) \\ F''_{0}(1) & \dots & F''_{n}(1) \\ \dots & \dots & \dots \end{bmatrix} \cdot \mathbf{q} = \begin{bmatrix} \overrightarrow{x}_{1} & 0 & 0 & 0 & 0 & 0 \\ \overrightarrow{x}_{2} & \kappa_{0} \overrightarrow{x}_{1}^{2} & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \overleftarrow{x}_{1} & 0 & 0 \\ 0 & 0 & 0 & \overleftarrow{x}_{2} & \kappa_{1} \overleftarrow{x}_{1}^{2} & 0 \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \cdot \mathbf{f}$$
(2.34)

which can be considered as restating the GH reconstruction constraints of order 1 to n, on the derivative curve, or in other words, the *hodograph* of the unknown curve.

Let $E_0(t), ..., E_{n-1}(t) : [0, 1] \to \mathbb{R}$ be the order *n* basis functions in the initial basis of the problem that span the space of hodographs - for example, in the

case of Bernstein polynomials $F_i(t) = B_i^n(t)$ and $E_i(t) = B_i^{n-1}(t)$.

Let us express the $\mathbf{q}'(0), ..., \mathbf{q}'(1), ...$ derivative vectors in the $E_i(t)$ basis and let us denote their coordinates with $\Delta \mathbf{q}_0, ..., \Delta \mathbf{q}_{n-1}$. Then (2.34) becomes

$$\underbrace{\begin{bmatrix} E_0(0) & \dots & E_{n-1}(0) \\ E'_0(0) & \dots & E'_{n-1}(0) \\ \dots & \dots & \dots \\ E_0(1) & \dots & E_{n-1}(1) \\ E'_0(1) & \dots & E'_{n-1}(1) \\ \dots & \dots & \dots \\ E \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \Delta \mathbf{q}_0 \\ \dots \\ \Delta \mathbf{q}_{n-1} \end{bmatrix}}_{\mathbf{\Delta q}} = \begin{bmatrix} \overrightarrow{x}_1 & 0 & 0 & 0 & 0 & 0 \\ \overrightarrow{x}_2 & \kappa_0 \overrightarrow{x}_1^2 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \overleftarrow{x}_1 & 0 & 0 \\ 0 & 0 & 0 & \overleftarrow{x}_2 & \kappa_1 \overleftarrow{x}_1^2 & 0 \\ \dots & \dots & \dots & \dots \\ \end{bmatrix} \cdot \mathbf{f}$$

Solving this, using any of the methods shown previously, results in an $\mathbf{h}(t): [0,1] \to \mathbb{R}^3$ hodograph, which is a best approximation to the second and higher order geometric invariants in $\mathbf{D}_0, \mathbf{D}_1$.

Integrating this hodograph yields a $\mathbf{q}(t)$ curve with the same geometric invariant approximation properties, and additionally, the integration constant can be used to reconstruct one of the endpoint position data. Let us construct the endpoint interpolating integrated curves for both \mathbf{p}_0 and \mathbf{p}_1 in the $F_0(t), ..., F_n(t)$ basis, i.e. let

$$\mathbf{q}_i(t) = \mathbf{p}_i + \int_0^1 \mathbf{h}(x) dx \ , i = 0, 1$$

and let $\mathbf{q}_j^{<i>}$ denote the control data of $\mathbf{q}_i(t)$:

$$\mathbf{q}_i(t) = \sum_{j=0}^n \mathbf{q}_j^{} F_i(t) \; .$$

By using the notation

$$NZ(\{F_j(t)\}_{j=0}^n, x) = \left\{ i \in \{0, 1, .., n\} | F_i(x) \neq 0 \right\},\$$

a sufficient condition for position reconstruction is

$$\forall i \in NZ(\{F_j(t)\}_{j=0}^n, 0) \cap NZ(\{F_j(t)\}_{j=0}^n, 1) : \mathbf{q}_i^{<0>} = \mathbf{q}_i^{<1>} .$$



Figure 2.1: Cubic, endpoint reconstructing GH interpolant. The red portions of the curve denote non-constant speed regions. The red arrows at endpoints denote the desired tangent directions, green arrows are normal vectors, the gray parabolas represent the desired curvature at endpoints. As it can be seen, a single cubic polynomial was insufficient to yield a good approximation to these and satisfy exact position reconstruction at the same time.

Depending on the actual base of reconstruction, this may or may not be a necessary condition of reconstruction.

The remaining control data can be computed by another optimization, or simply by taking the average of the two sets of control data. In the latter case, the control data of the final curve $\mathbf{q}(t)$ are

$$\forall i \in NZ(\{F_j(t)\}_{j=0}^n, 0) : \mathbf{q}_i = \mathbf{q}_i^{<0>}$$

$$\forall i \in NZ(\{F_j(t)\}_{j=0}^n, 1) : \mathbf{q}_i = \mathbf{q}_i^{<1>}$$

$$\forall i \notin NZ(\{F_j(t)\}_{j=0}^n, 0) \cap NZ(\{F_j(t)\}_{j=0}^n, 1) : \mathbf{q}_i = \frac{1}{2}\mathbf{q}_i^{<0>} + \frac{1}{2}\mathbf{q}_i^{<1>}$$

It is important to take into account, that this method reduces the degrees of freedom available for higher order geometric invariant reconstruction. Figure 2.1 shows an example of a cubic Bézier curve that was used to reconstruct second order GH data with position constrains. The figure illustrates that with the enforced reconstruction constraints, a cubic is not able to deal with approximate reconstruction of higher order invariants simultaneously.

2.4.4 General GH interpolation

So far, our discussion focused on two-point, symmetric GH interpolation, but the formulation of (2.29) allows us to investigate the interpolation of given $\mathbf{D}_i, i = 0, ..., k$ order- n_i GH data tuples at prescribed $t_0, ..., t_k \in [0, 1]$ parameter values, analogously.

In this case, **f** consists of all $\mathbf{f}_i = (\mathbf{p}_i, \mathbf{t}_i, \mathbf{n}_i, \mathbf{b}_i)$ data, and the matrix G is of a block-diagonal structure

$$G = \begin{bmatrix} G_0 & 0 & \dots & 0 \\ 0 & G_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_k \end{bmatrix}, \ \mathbf{f} = \begin{bmatrix} \mathbf{f}_0 \\ \dots \\ \mathbf{f}_k \end{bmatrix},$$

where $G_i \in \mathbb{R}^{(n_i+1)\times(d+1)}$ denotes the reconstruction matrix of order- n_i :

$$G_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x_1^{} & 0 & 0 \\ 0 & x_2^{} & \kappa_i (x_1^{})^2 & 0 \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

On the left-hand side, F contains the basis function and its required derivative evaluations at parameter values $t_0, ..., t_k$. All results shown previously regarding solvability, approximate solutions, and partial exact reconstruction apply to this case as well.

Geometric Hermite interpolation of this kind has been studied by Krajnc, and Zagar et. al. They have shown that a single planar cubic segment, reconstructing position and tangent data at three prescribed parameter values, has an approximation order of six, just like the BHS spline. [28]

Example: In this context, finding the Bézier control points of the parabola that passes through 3 points can be considered as the interpolation of

$$D_0 = (a), \ D_1 = (b), \ D_2 = (c)$$

 at

$$t_0 = 0, \ t_1 = \frac{1}{2}, \ t_2 = 1,$$

that is, as

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix}$$

which gives us the control points as

$$q_0 = \mathbf{a}$$

$$q_1 = 2\mathbf{b} - \frac{1}{2}\mathbf{a} - \frac{1}{2}\mathbf{c}$$

$$q_2 = \mathbf{c}$$

2.5 Parametrization optimization

The right hand side of geometric reconstruction equation (2.29) depends on the $\overrightarrow{x}_i, \overleftarrow{x}_i$ tangential coordinates of the derivatives. These are not fixed by the geometric constraints of GH interpolation, in other words, they are degrees of freedom of parametrization.

Setting them to particular values, and taking into account that the geometric invariants of the curve are fixed, amounts to a form of Hermite interpolation of the hodograph of the arc-length function, as it was discussed earlier. This is the consequence of Theorem 2, from which it can be seen immediately, that the highest derivative of the arc-length function, $s^{(n)}$, is in x_n .

Let us now consider how we could use the $\overrightarrow{x}_i, \overleftarrow{x}_i$ tangential degrees of freedom to optimize the parametrization of the curve.

As a result of Theorem 8, these tangential parameters can be used to find a lower degree solution than that of Theorem 7. Such a curve is found by solving the following nonlinear optimization problem (NLP):

$$\begin{array}{c} \min \\ \overrightarrow{x}_{1},...,\overrightarrow{x}_{n} \in \mathbb{R} \\ \overleftarrow{x}_{1},...,\overleftarrow{x}_{n} \in \mathbb{R} \\ \overrightarrow{x}_{1},...,\overleftarrow{x}_{n} \in \mathbb{R} \\ \overrightarrow{x}_{1} > 0 \\ \overleftarrow{x}_{1} > 0 \end{array} \tag{2.35}$$

where computation of control data $\mathbf{q}_0, ..., \mathbf{q}_m$ reduces to the problem of finding $\mathbf{q}_0, ..., \mathbf{q}_m$ for fixed $\overrightarrow{x}_i, \overleftarrow{x}_i$, which was discussed in the previous section.

Since the difference of curve and objective derivatives is zero for an exact



Figure 2.2: A convergence map of Newton iteration used to solve the GH interpolation problem of reconstructing $\mathbf{p}_0 = [100, 100]^T$, $\mathbf{t}_0 = [1, 0]^T$ and $\mathbf{p}_1 = [300, 300]^T$, $\mathbf{t}_1 = [0, -1]^T$ (on the right) with a parabola. The degrees of freedom are the $\vec{x}_1, \vec{x}_1 > 0$ first derivative lengths. The plane on the right has axes of \vec{x}_1, \vec{x}_1 . Each point corresponds to an initial guess for the start of Newton iteration. Red points indicate initial values of derivative lengths that did not result in a parabola that reconstructed the prescribed point-tangent data within error threshold. Green points that correspond to initial \vec{x}_1, \vec{x}_1 values that resulted in a parabola within error threshold.

solution, an arbitrary norm can be chosen in (2.35). Even if conditions of Theorem 8 are not met, by changing the $\overrightarrow{x}_i, \overleftarrow{x}_i$ degrees of freedom, the deviation from the prescribed geometric quantities can be minimized.

Note, however, that finding the degree reduced solution to a problem is sensitive to the choice of optimization method used to solve (2.35). Figure 2.2 shows an example of this.

By separating the task of control data computation from minimizing a given functional, *parametrization optimization* can be defined by the NLP problem of

$$\min_{\overrightarrow{x}_{1},...,\overrightarrow{x}_{n} \in \mathbb{R}} f(F, \mathbf{D}_{0}, \mathbf{D}_{1}; \overrightarrow{x}_{1}, ..., \overleftarrow{x}_{1}, ...)$$
(2.36)
$$\overleftarrow{x}_{1}, ..., \overleftarrow{x}_{n} \in \mathbb{R}$$

$$\overrightarrow{x}_{1} > 0$$

$$\overleftarrow{x}_{1} > 0$$

where f is a functional incorporating desirable properties, such as minimum

strain energy, minimal curvature variation, etc.

Minimizing the deviation of parametrization from arc-length parametrization can be also achieved by minimizing

$$\int |1 - \mathbf{q}'(t) \cdot \mathbf{q}'(t)| dt$$

however, a practically more useful objective is to penalize deviation from constant speed parametrization. By noting that if $\mathbf{q}(t)$ is roughly constant speed then $\mathbf{q}'(t) \cdot \mathbf{q}'(t) \approx c$, for an arbitrary $c \in \mathbb{R}^+$, hence $\mathbf{q}'(t) \cdot \mathbf{q}''(t) \approx 0$, then minimizing

$$\int |\mathbf{q}'(t) \cdot \mathbf{q}''(t)| dt$$

results in a parametrization that minimizes deviation from being constant speed. Figure 2.4 illustrates a GH quintic reconstructing second order GH data that also minimizes the parametric acceleration.

This is an especially attractive parametrization when one considers the use of geometric Newton-Raphson methods for curve interrogation [21] [48], or any other technique that relies on a local re-parametrization of a curve. In geometric Newton-Raphson methods, a nearly constant-speed parametrization decreases the error of the pull-back step, increasing the accuracy of algorithms using higher order geometric proxies.

Another set of problems that benefit from uniform speed is when the curves define tool paths. Farouki proposed in [13] a rational reparametrization of Bézier curve so that the reparametrization of the form $t_{\alpha}(u) = \frac{u(\alpha-1)}{2u\alpha-u-\alpha}$ yields a more constant speed traversal of the curve. Figure 2.3 compares optimal parametrizations. Note that while Farouki's reparametrization turns the curve in question into a rational curve, the optimization I proposed retains its algebraic class, i.e., integral polynomials in our case, at the expense of changing the non-endpoint geometry of the curve.

The NLP problem of (2.36) requires two optimization algorithms at two stages:

- a nonlinear optimization method is required to find the $\vec{x}_i, \overleftarrow{x}_i$ coordinates that minimize $f(\cdot)$,
- another optimization method is used to compute the control data **q** such that the resulting curve approximates the correct derivatives



(c) Nonlinear parametrization optimiza- (d) Farouki's optimal rational tion yields a more uniform speed reparametrization applied to the parametrization. Red points are samples curve. Green points are equal spacing from the curve taken at an equal para- in the rational parametrization, almost metric spacing of [0, 1].

Figure 2.3: Parametrization optimization in GH interpolation. The input second order GH data tuples are represented by the dark red parabolas at endpoints. Green diagrams at the bottom show the normalized magnitude of parametric speed from t = 0 to t = 1 (from left to right). Note how compressed the range of speed became after applying parametrization optimization. In this case, Farouki's rational reparametrization resulted only a slightly more compressed parametrization.

Optimal divergence found = 96.26987 Divergence of current curve = 96.26987



Figure 2.4: A quintic second order GH interpolant curve with arc-length like parametrization at endpoints, up to order 2. The blue regions denote constant-speed areas, red regions denote high acceleration.

Figure 2.5 shows the results of two different search methods to find optimal $\overrightarrow{x}_i, \overleftarrow{x}_i$ coordinates on the same second order GH input data pair, using LSQ control data.

Another level of parametrization optimization arises when one interpolates several \mathbf{D}_i GH data tuples at *unkown* parameter values. All classic methods – like uniform, chordal, centripetal, Piegl's, etc. - can be used to estimate these parameter values and once those are fixed, the method discussed in this section can be used. See [15] for a detailed survey of these methods and how they affect the accuracy of approximation.

2.6 Adaptive curve fitting

It was shown in 2.4.3 how to satisfy symmetric reconstruction constraints in GH approximation. However, the necessity of exact non-symmetric reconstruction arises naturally in applications.

An adaptive curve fitting scheme is introduced in this section to approximate a given set of GH data tuples with a piece-wise degree n Bézier spline curve. For this, an asymmetric, left-endpoint interpolatory GH interpolation technique is used for segment construction. That is, let us consider the follow-



Figure 2.5: Usage of different nonlinear optimization methods. Approximate second order GH interpolant cubic curves, minimizing deviation from constant speed. The red parts of the curve denote points where acceleration is large, blue parts correspond to small acceleration points. In comparison of the two, the curve on the left has a smaller deviation from being completely constant speed.

ing problem:

Given are N GH-data tuples $\mathbf{D}_1, ..., \mathbf{D}_N$, an $\epsilon > 0$ error threshold, $n \in \mathbb{N}^+$.

Find a piece-wise degree n polynomial spline that approximates the prescribed $\mathbf{D}_1, ..., \mathbf{D}_N$ data tuples within piece-wise ϵ error and it has a minimum number of segments.

For the sake of simplicity, the solution to the above problem is sought among G^0 splines. The arguments presented here can be extended to higher order continuity between segments. Let us also assume that the error functional is the $|| \cdot ||_2$ norm of $F \cdot \mathbf{q} - G \cdot \mathbf{f}$.

The spline solution is constructed segment by segment. For each segment, a consecutive sequence of GH data tuples $\mathbf{D}_a, \mathbf{D}_{a+1}, ..., \mathbf{D}_b$ is sought, such that they can be approximated within ϵ error by a single degree n Bézier curve.

First, appropriate t_j parameter values have to be found, where the segment should reconstruct the \mathbf{D}_j data, $j \in \{a, a+1, ..., b\}$. This can be done simply by $t_j = \frac{j-a}{b-a}$ equidistant parameters, but more sophisticated parameter estimation methods can be used. Please refer to Floater et. al.'s survey of these [15].

In the construction of a single segment, G^0 continuity stipulates that seg-

ments should join at endpoints, which is achieved by connecting each new segment to the t = 1 endpoint of its predecessor.

This requires that the first control point of the *i*-th segment, denoted by $\mathbf{q}_0^{\langle i \rangle}$, is fixed, that is, for the $\mathbf{q}_0^{\langle i \rangle}, ..., \mathbf{q}_n^{\langle i \rangle} \in \mathbb{E}^3$ control data of segment *i*

$$\min_{\mathbf{q}_{0}^{},..,\mathbf{q}_{n}^{}} ||F^{} \cdot \mathbf{q}^{} - G^{} \cdot \mathbf{f}^{}||_{2}$$

$$\mathbf{q}_{0}^{} = \mathbf{q}_{n}^{(i-1)}$$
(2.37)

should be solved, assuming that the GH tangential degrees of freedom are set to some feasible values.

Similarly to the symmetric reconstruction case, the equality constraint can be omitted by removing $\mathbf{q}_0^{\langle i \rangle}$ from $\mathbf{q}^{\langle i \rangle}$. In order to do that, let us translate (2.37) to the hodograph of $\mathbf{q}^{\langle i \rangle}(t)$. Once that is done, integrating the hodograph and setting its integration constant to $\mathbf{q}_0^{\langle i \rangle} = \mathbf{q}_n^{(i-1)}$ yields a leftendpoint interpolatory curve.

Transforming the $F^{\langle i \rangle} \cdot \mathbf{q}^{\langle i \rangle} = G^{\langle i \rangle} \cdot \mathbf{f}^{\langle i \rangle}$ system to $\Delta \mathbf{q}_j^{\langle i \rangle}$ hodograph control data is straightforward for derivatives:

$$\begin{bmatrix} nB_0^{n-1}(0) & \dots & nB_{n-1}^{n-1}(0) \\ n(n-1)B_0^{n-1'}(0) & \dots & n(n-1)B_{n-1}^{n-1'}(0) \\ \dots & \dots & \dots & \dots \\ nB_0^{n-1}(1) & \dots & nB_{n-1}^{n-1}(1) \\ n(n-1)B_0^{n-1'}(1) & \dots & n(n-1)B_{n-1}^{n-1'}(1) \\ \dots & \dots & \dots & \dots \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \Delta \mathbf{q}_0 \\ \dots \\ \Delta \mathbf{q}_{n-1} \end{bmatrix}}_{\Delta \mathbf{q}} = \begin{bmatrix} \overrightarrow{x}_1 & 0 & 0 & 0 & 0 & 0 \\ \overrightarrow{x}_2 & \kappa_0 \overrightarrow{x}_1^2 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \overleftarrow{x}_1 & 0 & 0 \\ 0 & 0 & 0 & \overleftarrow{x}_2 & \kappa_1 \overleftarrow{x}_1^2 & 0 \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}} \cdot \mathbf{f}^{\langle i \rangle} , \qquad (2.38)$$

however, the positional constraints on $\mathbf{q}^{\langle i \rangle}(t)$ have to be expressed in the $\mathbf{q}_0^{\langle i \rangle}, \Delta \mathbf{q}_j^{\langle i \rangle}$ basis as well.

So let us find $\alpha_j^{<i>}(t): \mathbb{R} \to \mathbb{R}$ coefficient functions such that

$$\mathbf{q}^{}(t) = \sum_{j=0}^{n} \mathbf{q}_{j}^{} B_{j}^{n}(t) = \mathbf{q}_{0}^{} + \sum_{j=0}^{n-1} \alpha_{j}(t) \Delta \mathbf{q}_{j}^{}$$

holds, that is, let us express the evaluations of the $\mathbf{q}_0^{\langle i \rangle} + \int \sum_{j=0}^{n-1} n \Delta \mathbf{q}_j^{\langle i \rangle} B_j^{n-1}(t) dt$ integral curve with some $\alpha_j^{\langle i \rangle}(t)$ degree *n* polynomial coefficients for $\Delta \mathbf{q}_j^{\langle i \rangle}$.

From the rules of integration and the indefinite integral formula of Bernstein polynomials [10]

$$\int B_i^n(t)dt = \frac{1}{n+1} \sum_{j=i+1}^{n+1} B_j^{n+1}(t) ,$$

it follows that

$$\int \sum_{j=0}^{n-1} n \Delta \mathbf{q}_j^{} B_j^{n-1}(t) dt = \sum_{j=0}^{n-1} n \Delta \mathbf{q}_j^{} \int B_j^{n-1}(t) dt$$
$$= \mathbf{q}_0^{} + \sum_{j=0}^{n-1} n \Delta \mathbf{q}_j^{} \frac{1}{n} \sum_{k=j+1}^n B_k^n(t) ,$$

by setting the integration constant to $\mathbf{q}_0^{\langle i \rangle} = \mathbf{q}_n^{(i-1)}$.

The coefficients are then

$$\alpha_j^{\langle i \rangle}(t) = \sum_{k=j+1}^n B_k^n(t) ,$$

which can be used to re-add the original positional constraints to (2.38) by replacing the

$$\mathbf{q}^{\langle i \rangle}(t_j) = \mathbf{p}_j$$

type of position reconstruction constraints with

$$\sum_{j=0}^{n-1} \alpha_j^{\langle i \rangle}(t) \cdot \Delta \mathbf{q}_j^{\langle i \rangle} = \mathbf{p}_j - \mathbf{q}_n^{\langle i-1 \rangle} .$$

The solution of the thus modified

$$F' \cdot \Delta \mathbf{q}^{\langle i \rangle} = G^{\langle i \rangle} \cdot \mathbf{f}' \tag{2.39}$$

system yields a $\Delta \mathbf{q}^{\langle i \rangle}$, from which the control points of the degree *n* Bézier segment are found as

$$\mathbf{q}_{j}^{} = \mathbf{q}_{j-1}^{} + \Delta \mathbf{q}_{j}^{}, \ j = 1, 2, .., n$$
.

Now, let us formulate an algorithm that computes a degree n LSQ approximant to a consecutive sequence of GH data tuples.

Algorithm	1	Approximation	of a	sequence o	of GH	data t	uples	

1: procedure Approximate $(a, b \in \{1, .., N\}, n \in \mathbb{N}^+)$

2: Parameter setup:

3: Let $t_i \in [0, 1]$, $i \in [a, b]$ be parameter values associated with \mathbf{D}_i

- 4: Setup reconstruction system:
- 5: Let $\mathbf{q} = [\mathbf{q}_0, ..., \mathbf{q}_n]^T$ be the unknown control data, F the matrix of basis function evaluations at $t_i, i \in [a, b]$, G the generalized reconstruction matrix of $\mathbf{D}_a, ..., \mathbf{D}_b$, and \mathbf{f} the Frenet trihedrons of $\mathbf{D}_a, ..., \mathbf{D}_b$
- 6: Optimize parametrization:

7: Solve

$$\min_{x_1^{(a)},\dots,x_1^{(a+1)},\dots,x_1^{(b)},\dots} ||F \cdot \mathbf{q} - G \cdot \mathbf{f}||_2$$

where $x_1^{(a)}, ..., x_1^{(b)} > 0$, with $\mathbf{q} = F^+ \cdot G \cdot \mathbf{f}$

8: *Return*:

9: $\mathbf{q}(t) \text{ and } \epsilon \leftarrow ||F \cdot \mathbf{q} - G \cdot \mathbf{f}||_2$

Algorithm 1 is used to construct the first segment of the final spline. It is expressed in terms of control data $\mathbf{q}_0, ..., \mathbf{q}_n$, since in its case, there is no position reconstruction restrain on the left endpoint.

All subsequent segments are created by Algorithm 2, which makes sure that the resulting $\mathbf{q}^{\langle i \rangle}(t)$ approximant is such that $\mathbf{q}^{\langle i \rangle}(0) = \mathbf{x}$ holds, for an arbitrary $\mathbf{x} \in \mathbb{E}^3$ input point.

The *Parameters setup* is a common step in both algorithms. It is used to find the t_j parameter values of the unknown curve $\mathbf{q}^{\langle i \rangle}(t)$ where it should reconstruct \mathbf{D}_j .

The final algorithm is a sequence of an initial segment creation, by using Algorithm 1, followed by an iterative application of Algorithm 2. The final algorithm is listed in 3. For the sake of simplicity, the trivial error-handling parts are omitted.

The above construction can be generalized to higher order continuity be-

Algorithm 2 Approximation of a sequence of GH data tuples with leftendpoint interpolation

1: procedure ApproximateFromPoint $(a, b \in \{1, .., N\}, n \in \mathbb{N}^+, \mathbf{x} \in \mathbb{E}^3)$

2: Parameter setup:

3: Let $t_i \in [0, 1]$, $i \in [a, b]$ be parameter values associated with \mathbf{D}_i

- 4: Setup reconstruction system:
- 5: Let $F', \Delta \mathbf{q}, G, \mathbf{f}'$ be as specified in (2.39)
- 6: Optimize parametrization:
- 7: Solve

$$\min_{x_1^{(a)}, \dots, x_1^{(a+1)}, \dots, x_1^{(b)}, \dots} ||F' \cdot \Delta \mathbf{q} - G \cdot \mathbf{f}||_2$$

where $x_1^{(a)}, ..., x_1^{(b)} > 0$, with $\mathbf{q} = F^+ \cdot G \cdot \mathbf{f}$

8: Compute control points:

9: Let
$$\mathbf{q}_j = \mathbf{x} + \Delta \mathbf{q}_j, \ j = 1, 2, .., n$$
.

10: *Return*:

11: $\mathbf{q}(t)$ and $\epsilon \leftarrow ||F \cdot \mathbf{q} - G \cdot \mathbf{f}||_2$

Algorithm 3 Adaptive curve-fitting

- 1: procedure FITCURVE $(\mathbf{D}_1, .., \mathbf{D}_N, n \in \mathbb{N}^+, \epsilon > 0)$
- 2: Create first segment:
- 3: By binary search, find $b \in \{1, .., n\}$ such that
- 4: **Approximate** $(1, b, n) < \epsilon$
- 5: $a \leftarrow b+1, b \leftarrow n, \mathbf{x} \leftarrow \mathbf{q}(1)$
- 6: Cover the remaining GH tuples: while a < n:
- 7: By binary search, find $b \in \{a + 1, .., n\}$ such that
- 8: **ApproximateFromPoint** $(a, b, n, \mathbf{x}) < \epsilon$
- 9: $a \leftarrow b + 1, b \leftarrow n, \mathbf{x} \leftarrow \mathbf{q}(1)$

tween segments by translating (2.38) to a higher order hodograph of the solution curve. The hodograph found then needs to be integrated, using the derivative and positional data of the last point of the previous segment as integration constants.

2.7 Algorithms of GH interpolation

Let us now consider the problem of constructing parametric curves such that they reconstruct prescribed GH data tuples at knots.

Theorem 8 already provides us means to compute control data of GH interpolants: in the full rank case the control data is computed as $F^{-1} \cdot \mathbf{f}$, while in the non-full rank case the *i*-th coordinate of the control data vector can be any point of the intersection $C(F) \cap \mathbf{g}(i)$

In this section, I focus on interpolation algorithms that do not rely on direct matrix inversion or on locating intersection points of nonlinear sets.

For the sake of simplicity, let us focus on reconstruction with a single curve segment at parametric endpoints t = 0, 1. A G^n composite spline is obtained by constructing these segments for each consecutive pairs of *n*th order GH data tuples.

I propose two types of algorithms for the construction of GH interpolant segments:

- Theorem 8 provides means to devise *direct methods* to compute the control data of a GH inderpolant. If conditions of Theorem 8 are met, the members of the family of GH intrepolants can be obtained by traversing all feasible tangential $\vec{x}_i, \overleftarrow{x}_i$ values. This provides a k parameter family of solutions, for some k.
- Indirect methods blend one-point GH interpolants, or basic curves, to construct GH interpolant segments. Basic curves reconstruct appropriate \mathbf{D}_i GH data tuples at each knot, and these are blended such that their geometric invariants are unchanged up to order n. Figure 2.6 shows an example of this.

To illustrate these two different approaches, examples of each method are presented to solve third order GH interpolation with Bézier curves.



Figure 2.6: Planar second order GH data are represented by position tangent direction, and osculating circles in (a). Portions of the control circles are then blended together to yield the curve in black in (b).

A motivation for the choice of third order GH interpolation is the fact that no throughout characterization of this problem has been presented in the literature to the author's knowledge.

2.7.1 Direct methods

This section provides a geometric characterization of third order GH interpolation. For robust computations, these conditions are important because numerically finding a degree reduced parametrization is sensitive to the choice of optimization method, as it was illustrated by the example in Figure 2.2.

Let there be given a

$$\mathbf{D}_i = (\mathbf{p}_i; \mathbf{t}_i; \kappa_i, \mathbf{n}_i, \mathbf{b}_i; \tau_i, \hat{\kappa}'_i)$$

pair of third order GH data tuples, i = 0, 1, and let us find a

$$\mathbf{c}(t) = \sum_{i=0}^{n} \mathbf{c}_{i} B_{i}^{n}(t), \ t \in [0,1]$$

Bézier curve that reconstructs these quantities at t = 0 and t = 1. $B_i^n(t)$

denotes the *i*-th degree *n* Bernstein polynomial: $B_i^n(t) = {n \choose i} t^i (1-t)^{n-i}$.

Restrictions on the control points have to be formulated such that, by using the unconstrained degrees of freedom, all members of the family of GH interpolant solutions can be generated. Here, our aim is to derive geometric constraints on the input parametrization-independent data, similarly to how Schaback formulated the second order GH reconstruction constraints of Bézier control points in [41].

For the sake of simplicity, let us first consider only the t = 0 endpoint, that is, the reconstruction of \mathbf{D}_0 at t = 0, and let us assume that $\kappa_0 \neq 0$. Let us also omit the subscripts, that is, let $\kappa = \kappa_0$, $\mathbf{p} = \mathbf{p}_0$ and so on. Figure 2.7 illustrates the Frenet trihedron and the planes spanned by each pair of its vectors at a given point of a curve. These planes play an important role in what follows.

Let $\mathbf{S}[m], \mathbf{N}[m], \mathbf{R}[m]$ denote the translates of the osculating, normal, and rectifying planes with respect to their normal vector, that is let

$$\mathbf{S}[m] = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x} \cdot \mathbf{b} = m\}$$
$$\mathbf{N}[m] = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x} \cdot \mathbf{t} = m\}$$
$$\mathbf{R}[m] = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x} \cdot \mathbf{n} = m\}$$

As the order of geometric Hermite reconstruction increases, the following constraints are found on the derivatives of $\mathbf{c}(t)$, assuming that all lower order reconstruction conditions are met:

• tangent direction reconstruction constrains $\mathbf{c}'(0)$ such that

$$\mathbf{c}'(0) = \alpha \cdot \mathbf{t}$$

for some $\alpha > 0$. This can be interpreted geometrically as

$$\mathbf{c}'(0) \in \mathbf{S} \cap \mathbf{R} \wedge \mathbf{c}'(0) \cdot \mathbf{t} > 0$$

that is, the first derivative vector should lie in the intersection of the osculating and rectifying planes, such that it is contained in the tangent ray. In turn, this means that \mathbf{c}_1 should lie on the tangent ray starting from $\mathbf{c}(0) = \mathbf{p}$, i.e. $\mathbf{c}_1 = \mathbf{p} + t \cdot \mathbf{t}$ for some t > 0. The set of all feasible



Figure 2.7: Frenet trihedron vectors $\mathbf{t}, \mathbf{n}, \mathbf{b}$, spanning the osculating (**S**), normal (**N**), and rectifying (**R**) planes.

 \mathbf{c}_1 control points forms a half line, the tangent ray.

• curvature reconstruction constrains the c''(0) second derivative. From (2.16), it follows that

$$\mathbf{c}''(0) = \begin{bmatrix} s''\\(s')^2\kappa\\0 \end{bmatrix}_F$$

should hold for some $s' > 0, s'' \in \mathbb{R}$. That is, the second derivative should lie in the intersection of the osculating plane and a normal-translate of the rectifying plane. More precisely, $\mathbf{c}''(0)$ is such that

$$\mathbf{c}''(0) \in \mathbf{S} \cap \mathbf{R}[(s')^2 \kappa] . \tag{2.40}$$

This intersection forms a line in \mathbb{R}^3 . Any vector of this intersection can be chosen as the $\mathbf{c}''(0)$ derivative, which – once \mathbf{c}_0 and \mathbf{c}_1 are set – determines the \mathbf{c}_2 control point in \mathbb{E}^3 . The resulting Bézier curve reconstructs the prescribed curvature and binormal vector, if \mathbf{c}_0 and \mathbf{c}_1 were chosen such that position and tangent direction conditions are satisfied. The set of all feasible c_2 control points form an open half plane in \mathbb{E}^3 that lies inside the osculating plane, bordered by the tangent line.

torsion reconstruction places restrictions on the c'''(0) derivative vector, but only along the binormal. This can be seen, once again using (2.16), from

$$\mathbf{c}'''(0) = \begin{bmatrix} s''' - (s')^3 \kappa^2 \\ 3s' s'' \kappa + (s')^3 \kappa' \\ (s')^3 \kappa \tau \end{bmatrix}_F$$

This can be interpreted as stipulating that $\mathbf{c}'''(0)$ lies in a binormal translate of the osculating plane:

$$\mathbf{c}^{\prime\prime\prime}(0) \in \mathbf{S}[(s^{\prime})^{3}\kappa\tau] . \tag{2.41}$$

Choosing any vector of this translated osculating plane determines the fourth control point, once $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2$ are fixed.

All feasible $\mathbf{S}[(s')^3 \kappa \tau]$ translates of the osculating plane – for all s' > 0 – form an open half-space in \mathbb{R}^3 .

• curvature derivative reconstruction gives

$$\mathbf{c}'''(0) \in \mathbf{R}[3s's''\kappa + (s')^3\kappa']$$
, (2.42)

similarly to the torsion reconstruction case.

As a result, torsion and curvature derivative reconstruction together restrict $\mathbf{c}'''(0)$ such that

$$\mathbf{c}^{\prime\prime\prime}(0) \in \mathbf{S}[(s')^3 \kappa \tau] \cap \mathbf{R}[3s's''\kappa + (s')^3 \kappa'] ,$$

which can be interpreted geometrically as the intersection of two open halfspaces in \mathbb{R}^3 .

Analogously, all subsequent $\hat{\tau}^{(j-1)}$ derivatives of torsion and $\hat{\kappa}^{(j)}$ derivatives of curvature impose restrictions on the \mathbf{c}_{3+j} control point along the binormal and normal. This allows a straightforward generalization of the results presented here to fourth and higher order GH reconstruction with Bézier curves.

Translating these results more directly to control points can be easily done

by introducing the Frenet coordinates of the first four control points: let $(\overline{x}_n^{(i)}, \overline{y}_n^{(i)}, \overline{z}_n^{(i)}), i = 0, 1$ denote the coordinates of the *n*-th control point in the $\overline{F} = (\mathbf{p}_i; \mathbf{t}_i, \mathbf{n}_i, \mathbf{b}_i)$ coordinate system, that is, let $\overline{x}_n^{(i)}, \overline{y}_n^{(i)}, \overline{z}_n^{(i)} \in \mathbb{R}$ be such that

$$\mathbf{c}_n^{(i)} = \mathbf{p}_i + \overline{x}_n^{(i)} \mathbf{t}_i + \overline{y}_n^{(i)} \mathbf{n}_i + \overline{z}_n^{(i)} \mathbf{b}_i$$
.

Focusing on the i = 0 case, the first three derivatives of $\mathbf{c}(t)$ at t = 0 are defined by control points $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$. Let us now translate the GH reconstruction constraints on their $\overline{x}_i, \overline{y}_i, \overline{z}_i$ coordinates.

Since **position reconstruction** requires that $\mathbf{c}_0 = \mathbf{p}$, by omitting the upper indices and taking into account the higher order geometric reconstruction constraints (2.40)-(2.42), the following should hold for the control points

$$\mathbf{c}_{0} = \begin{bmatrix} 0\\0\\0\\\overline{F} \end{bmatrix}_{\overline{F}}, \ \mathbf{c}_{1} = \begin{bmatrix} \overline{x}_{1}\\0\\0\\\overline{F} \end{bmatrix}_{\overline{F}}, \ \mathbf{c}_{2} = \begin{bmatrix} \overline{x}_{2}\\\overline{y}_{2}\\0\\\overline{F} \end{bmatrix}_{\overline{F}}, \ \mathbf{c}_{3} = \begin{bmatrix} \overline{x}_{3}\\\overline{y}_{3}\\\overline{z}_{3}\\\overline{z}_{3} \end{bmatrix}_{\overline{F}}$$

for some $\overline{x}_1 > 0$, $\overline{x}_2, \overline{y}_2, \overline{x}_3, \overline{y}_3, \overline{z}_3 \in \mathbb{R}$ scalars.

Moreover, by denoting the Frenet-coordinates of $\mathbf{c}^{(i)}$ by x_i, y_i, z_i , the following holds

$$\begin{bmatrix} x_1\\y_1\\z_1 \end{bmatrix}_F = n \begin{bmatrix} \overline{x}_1 - \overline{x}_0\\\overline{y}_1 - \overline{y}_0\\\overline{z}_1 - \overline{z}_0 \end{bmatrix}_F,$$
$$\begin{bmatrix} x_2\\y_2\\z_2 \end{bmatrix}_F = n(n-1) \begin{bmatrix} \overline{x}_2 - 2\overline{x}_1 + \overline{x}_0\\\overline{y}_2 - 2\overline{y}_1 + \overline{y}_0\\\overline{z}_2 - 2\overline{z}_1 + \overline{z}_0 \end{bmatrix}_F,$$
$$\begin{bmatrix} x_3\\y_3\\z_3 \end{bmatrix}_F = n(n-1)(n-2) \begin{bmatrix} \overline{x}_3 - 3\overline{x}_2 + 3\overline{x}_1 - \overline{x}_0\\\overline{y}_3 - 3\overline{y}_2 + 3\overline{y}_1 - \overline{y}_0\\\overline{z}_3 - 3\overline{z}_2 + 3\overline{z}_1 - \overline{z}_0 \end{bmatrix}_F$$

from the $\mathbf{c}^{(i)}(0) = \frac{n!}{(n-i)!} \Delta^i \mathbf{c}_0$ derivative rule of Bézier curves at the t = 0 endpoint.

Substituting the derivative coordinates from (2.16) on the left hand side

,

and the zero Frenet coordinates on the right, the above is expressed as

$$\begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix}_F = n \begin{bmatrix} \overline{x}_1 \\ 0 \\ 0 \end{bmatrix}_F$$
(2.43)

$$\begin{bmatrix} x_2 \\ x_1^2 \kappa \\ 0 \end{bmatrix}_F = n(n-1) \begin{bmatrix} \overline{x}_2 - 2\overline{x}_1 \\ \overline{y}_2 \\ 0 \end{bmatrix}_F$$
(2.44)

$$\begin{bmatrix} x_3\\ 3x_1x_2\kappa + x_1^3\hat{\kappa}'\\ x_1^3\kappa\tau \end{bmatrix}_F = n(n-1)(n-2) \begin{bmatrix} \overline{x}_3 - 3\overline{x}_2 + 3\overline{x}_1\\ \overline{y}_3 - 3\overline{y}_2\\ \overline{z}_3 \end{bmatrix}_F$$
(2.45)

By rearranging these to express $\overline{x}_i, \overline{y}_i, \overline{z}_i$, and substituting the derivative coordinates with the control point coordinates in the formula obtained from (2.16), the $\mathbf{c}_i, i = 0, ..., 3$ control points are expressed in the $(\mathbf{p}; \mathbf{t}, \mathbf{n}, \mathbf{b})$ Frenet coordinate system as

$$\mathbf{c}_{0} = \begin{bmatrix} 0\\0\\0\\F \end{bmatrix}_{\overline{F}}, \ \mathbf{c}_{1} = \begin{bmatrix} \frac{x_{1}}{n}\\0\\0\\F \end{bmatrix}_{\overline{F}}, \ \mathbf{c}_{2} = \begin{bmatrix} \frac{x_{2}}{n(n-1)} + 2\overline{x}_{1}\\\frac{n}{n-1}\kappa\overline{x}_{1}^{2}\\0\\\end{bmatrix}_{\overline{F}}, \\ \mathbf{c}_{3} = \begin{bmatrix} \frac{x_{3}}{n(n-1)(n-2)} + 3\overline{x}_{2} - 3\overline{x}_{1}\\\frac{3n\overline{x}_{1}n(n-1)(\overline{x}_{2}-2\overline{x}_{1})\kappa + n^{3}\overline{x}_{1}^{3}\dot{\kappa}'}{n(n-1)(n-2)} + 3\frac{n}{n-1}\overline{x}_{1}^{2}\kappa\\\frac{n^{2}}{(n-1)(n-2)}\overline{x}_{1}^{3}\kappa\tau \end{bmatrix}_{\overline{F}},$$

depending on the x_i tangential coordinates of the first three derivatives of $\mathbf{c}(t)$ at t = 0. Since all tangential coordinates of the control points are degrees of freedom, the control points can be more concisely expressed as

$$\mathbf{c}_{0} = \begin{bmatrix} 0\\0\\0\\F \end{bmatrix}_{\overline{F}}, \ \mathbf{c}_{1} = \begin{bmatrix} \overline{x}_{1}\\0\\0\\F \end{bmatrix}_{\overline{F}}, \ \mathbf{c}_{2} = \begin{bmatrix} \overline{x}_{2}\\\frac{n}{n-1}\kappa\overline{x}_{1}^{2}\\0\\\end{bmatrix}_{\overline{F}},$$
$$\mathbf{c}_{3} = \begin{bmatrix} \overline{x}_{3}\\\frac{3\overline{x}_{1}n(n-1)(\overline{x}_{2}-2\overline{x}_{1})\kappa+n^{2}\overline{x}_{1}^{3}\hat{\kappa}'}{(n-1)(n-2)} + 3\frac{n}{n-1}\overline{x}_{1}^{2}\kappa\\\frac{n^{2}}{(n-1)(n-2)}\overline{x}_{1}^{3}\kappa\tau \end{bmatrix}_{\overline{F}}.$$

Let $\mathbf{h}_i : \mathbb{R}^i \to \mathbb{E}^3$ be the function generating all feasible locations of control points \mathbf{c}_i , depending on \overline{x}_j , j = 1, ..., i. In the case of the first three \mathbf{h}_i functions, these can be expressed as

$$\mathbf{h}_0 = \mathbf{p} \tag{2.46}$$

$$\mathbf{h}_1(\overline{x}_1) = \mathbf{p} + \overline{x}_1 \mathbf{t} \tag{2.47}$$

$$\mathbf{h}_{2}(\overline{x}_{1},\overline{x}_{2}) = \mathbf{p} + \overline{x}_{2}\mathbf{t} + \frac{n}{n-1}\kappa\overline{x}_{1}^{2}\mathbf{n}$$
(2.48)

$$\mathbf{h}_{3}(\overline{x}_{1}, \overline{x}_{2}, \overline{x}_{3}) = \mathbf{p} + \overline{x}_{3}\mathbf{t}$$

$$+ \frac{3n(n-1)\overline{x}_{1}\overline{x}_{2}\kappa - 3n^{2}\overline{x}_{1}^{2}\kappa + n^{2}\overline{x}_{1}^{3}\kappa'}{(n-1)(n-2)}\mathbf{n}$$

$$+ \frac{n^{2}}{(n-1)(n-2)}\overline{x}_{1}^{3}\kappa\tau\mathbf{b}$$

$$(2.49)$$

where $\overline{x}_1 > 0, \overline{x}_2, \overline{x}_3 \in \mathbb{R}$.

Let us now introduce the set of all possible locations of control point \mathbf{c}_i by \mathbf{H}_i , that is, let

$$\mathbf{H}_0 = \{ \mathbf{p} \mid \mathbf{p} = \mathbf{h}_0 \} \tag{2.50}$$

$$\mathbf{H}_1 = \{ \mathbf{x} \in \mathbb{E}^3 \mid \exists \overline{x}_1 > 0 : \mathbf{x} = \mathbf{h}_1(\overline{x}_1) \}$$
(2.51)

$$\mathbf{H}_2 = \{ \mathbf{x} \in \mathbb{E}^3 \mid \exists \overline{x}_1 > 0, \overline{x}_2 \in \mathbb{R} : \mathbf{h}_2(\overline{x}_1, \overline{x}_2) \}$$
(2.52)

$$\mathbf{H}_3 = \{ \mathbf{x} \in \mathbb{E}^3 \mid \exists \overline{x}_1 > 0, \overline{x}_2, \overline{x}_3 \in \mathbb{R} : \mathbf{x} = \mathbf{h}_3(\overline{x}_1, \overline{x}_2, \overline{x}_3) \}$$
(2.53)

These sets can be derived for both the t = 0 and t = 1 endpoints, and they are denoted by $\mathbf{H}_{i}^{(0)}$ and $\mathbf{H}_{i}^{(1)}$, respectively.

It is important to note, however, that if $\kappa = 0$, the fourth derivative has to be restricted as well, hence \mathbf{c}_4 is also affected by the reconstruction of the t = 0 endpoint GH data tuple. If, additionally, $\hat{\kappa}' = 0$ holds too, the fifth derivative, that is \mathbf{c}_5 , is also required for reconstruction at t = 0. Figure 2.8 illustrates these cases.

Let us now return to handling both endpoints, and let us assume that $\kappa_0, \kappa_1 \neq 0$. Then there is a 6-parameter family of degree 7 Bézier GH interpolants defined by

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{p}_0 \ , \ \ \mathbf{c}_1 &= \mathbf{h}_1^{(0)}(\overline{x}_1^{(0)}) \ , \ \ \mathbf{c}_2 &= \mathbf{h}_2^{(0)}(\overline{x}_1^{(0)}, \overline{x}_2^{(0)}) \ , \ \ \mathbf{c}_3 &= \mathbf{h}_3^{(0)}(\overline{x}_1^{(0)}, \overline{x}_2^{(0)}, \overline{x}_3^{(0)}) \ , \\ \mathbf{c}_7 &= \mathbf{p}_1 \ , \ \ \mathbf{c}_6 &= \mathbf{h}_1^{(1)}(\overline{x}_1^{(1)}) \ , \ \ \mathbf{c}_5 &= \mathbf{h}_2^{(1)}(\overline{x}_1^{(1)}, \overline{x}_2^{(1)}) \ , \ \ \mathbf{c}_4 &= \mathbf{h}_3^{(1)}(\overline{x}_1^{(1)}, \overline{x}_2^{(1)}, \overline{x}_3^{(1)}) \ , \end{aligned}$$



Figure 2.8: A degree 7 GH interpolant to the third order GH interpolation problem. If $\kappa_0 \neq 0$, control points affected by the \mathbf{D}_0 GH data tuple are the ones in blue. The control points affected by \mathbf{D}_0 in the $\kappa = 0 \land \hat{\kappa}' \neq 0$ and $\kappa = \hat{\kappa}' = 0 \land \hat{\kappa}'' \neq 0$ cases are enclosed by the corresponding rounded rectangles. Control points in red are affected by the \mathbf{D}_1 GH data tuple.

where $\overline{x}_1^{(0)}, \overline{x}_1^{(1)} > 0, \overline{x}_2^{(0)}, \overline{x}_3^{(0)}, \overline{x}_2^{(1)}, \overline{x}_3^{(1)} \in \mathbb{R}$, which are the parameters of the 6-parameter family of integral septic GH interpolants reconstructing \mathbf{D}_0 and \mathbf{D}_1 . The control points satisfy

$$egin{aligned} \mathbf{c}_0 &= \mathbf{p}_0, \ \ \mathbf{c}_1 \in \mathbf{H}_1^{(0)}, \ \ \mathbf{c}_2 \in \mathbf{H}_2^{(0)}, \ \ \mathbf{c}_3 \in \mathbf{H}_3^{(0)}, \ \mathbf{c}_7 &= \mathbf{p}_1, \ \ \mathbf{c}_6 \in \mathbf{H}_1^{(1)}, \ \ \mathbf{c}_5 \in \mathbf{H}_2^{(1)}, \ \ \mathbf{c}_4 \in \mathbf{H}_3^{(1)}. \end{aligned}$$

and they are shown in Figure 2.8.

If the solution is sought among the degree 6 polynomials, the control points are subject to

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{p}_0 \ , \ \ \mathbf{c}_1 &= \mathbf{h}_1^{(0)}(\overline{x}_1^{(0)}) \ , \ \ \mathbf{c}_2 &= \mathbf{h}_2^{(0)}(\overline{x}_1^{(0)}, \overline{x}_2^{(0)}) \ , \ \ \mathbf{c}_3 &= \mathbf{h}_3^{(0)}(\overline{x}_1^{(0)}, \overline{x}_2^{(0)}, \overline{x}_3^{(0)}) \ , \\ \mathbf{c}_6 &= \mathbf{p}_1 \ , \ \ \mathbf{c}_5 &= \mathbf{h}_1^{(1)}(\overline{x}_1^{(1)}) \ , \ \ \mathbf{c}_4 &= \mathbf{h}_2^{(1)}(\overline{x}_1^{(1)}, \overline{x}_2^{(1)}) \ , \ \ \mathbf{c}_3 &= \mathbf{h}_3^{(1)}(\overline{x}_1^{(1)}, \overline{x}_2^{(1)}, \overline{x}_3^{(1)}) \ , \end{aligned}$$

however, not all six $\overline{x}_i^{(j)}$, i = 0, 1, j = 0, 1, 2 parameters are independent: $\overline{x}_1^{(0)}, \overline{x}_1^{(1)} > 0, \overline{x}_2^{(0)}, \overline{x}_2^{(1)} \in \mathbb{R}$ should be chosen such that there exist $\overline{x}_3^{(0)}, \overline{x}_3^{(1)} \in \mathbb{R}$ so that $\mathbf{h}_3^{(0)}(\overline{x}_1^{(0)}, \overline{x}_2^{(0)}, \overline{x}_3^{(0)}) = \mathbf{h}_3^{(1)}(\overline{x}_1^{(1)}, \overline{x}_2^{(1)}, \overline{x}_3^{(1)})$. In other words, \mathbf{c}_3 is under the influence of both \mathbf{D}_0 and \mathbf{D}_1 , as shown in Figure 2.9.

Algebraically, this is a system of 3 nonlinear equations with 6 unknowns: the $\overline{x}_i^{(j)}$ parameters of the family of solutions. Instead of trying to formulate algebraic existence conditions on the solution to this system, let us characterize the above geometrically.



Figure 2.9: A degree 6 GH interpolant to the third order GH interpolation problem, $\kappa_0, \kappa_1 \neq 0$ case. Control points affected by the \mathbf{D}_0 GH data tuple are the ones in blue. Control points in red are affected by the \mathbf{D}_1 GH data tuple.

The control point constraints can be written as

$$\begin{split} \mathbf{c}_0 &= \mathbf{p}_0 \ , \ \mathbf{c}_1 \in \mathbf{H}_1^{(0)} \ , \ \mathbf{c}_2 \in \mathbf{H}_2^{(0)} \ , \\ \mathbf{c}_3 \in \mathbf{H}_3^{(0)} \cap \mathbf{H}_3^{(1)} \ , \\ \mathbf{c}_6 &= \mathbf{p}_1 \ , \ \mathbf{c}_5 \in \mathbf{H}_1^{(1)} \ , \ \mathbf{c}_4 \in \mathbf{H}_2^{(1)} \ . \end{split}$$

It turns out that the above formulation also allows us to concisely summarize the necessary and sufficient condition for the existence of a degree 6 Bézier G^3 interpolant as follows:

Theorem 9 If $\kappa_0, \kappa_1 \neq 0$, there is a degree 6 integral polynomial solution to the third order GH interpolation problem $\Leftrightarrow \mathbf{H}_3^{(0)} \cap \mathbf{H}_3^{(1)} \neq 0$.

Proof.

⇒: if a GH interpolant exists, then control point \mathbf{c}_3 , viewed from the t = 0endpoint, satisfies the $\mathbf{c}_3 \in \mathbf{H}_3^{(0)}$ condition. If \mathbf{c}_3 is viewed from the t = 1endpoint, it satisfies $\mathbf{c}_3 \in \mathbf{H}_3^{(1)}$ as well. That is, $\mathbf{c}_3 \in \mathbf{H}_3^{(0)} \cap \mathbf{H}_3^{(1)}$, hence $\mathbf{H}_3^{(0)} \cap \mathbf{H}_3^{(1)} \neq \emptyset$.

 $\Leftarrow:$ let $\mathbf{c}_3 \in \mathbf{H}_3^{(0)} \cap \mathbf{H}_3^{(1)}$ be an arbitrary point.

Because $\mathbf{c}_3 \in \mathbf{H}_3^{(0)}$, due to the definition of $\mathbf{H}_3^{(0)}$, there exist $\overline{x}_1^{(0)} > 0$, $\overline{x}_2^{(0)} \in \mathbb{R}$ coordinates, such that they produce the $\overline{y}_3^{(0)}$ and $\overline{z}_3^{(0)}$ Frenet coordinates of \mathbf{c}_3 in \overline{F}_0 , and $\overline{x}_3^{(0)} = (\mathbf{c}_3 - \mathbf{p}_0) \cdot \mathbf{t}_0$. But these tangential coordinates also uniquely determine \mathbf{c}_1 and \mathbf{c}_2 , since $\mathbf{c}_1 = \mathbf{h}_1^{(0)}(\overline{x}_1^{(0)})$ and $\mathbf{c}_2 = \mathbf{h}_2^{(0)}(\overline{x}_1^{(0)}, \overline{x}_2^{(0)})$.

Similarly, because $\mathbf{c}_3 \in \mathbf{H}_3^{(1)}$, there exist $\overline{x}_1^{(1)} > 0$, $\overline{x}_2^{(1)} \in \mathbb{R}$ coordinates, such that they produce the $\overline{y}_3^{(1)}$ and $\overline{z}_3^{(1)}$ Frenet coordinates of \mathbf{c}_3 in \overline{F}_1 . These tangential coordinates also uniquely determine $\mathbf{c}_4 = \mathbf{h}_2^{(1)}(\overline{x}_1^{(1)}, \overline{x}_2^{(1)})$ and $\mathbf{c}_5 = \mathbf{h}_1^{(1)}(\overline{x}_1^{(1)})$.



Figure 2.10: Quintic integral polynomial solution to third order GH interpolation. Control points affected by the \mathbf{D}_0 GH data tuple are the ones in blue. Control points in red are affected by the \mathbf{D}_1 GH data tuple.

Because $\mathbf{c}_0 = \mathbf{p}_0$ and $\mathbf{c}_6 = \mathbf{p}_1$, all control points of a GH interpolant are found.

A more complicated situation arises if the solution is sought among the degree 5 integral Bézier curves. The inner two control points are constrained by both \mathbf{D}_0 and \mathbf{D}_1 , as illustrated in Figure 2.10.

The controls point of all solutions are subject to

$$\begin{split} \mathbf{c}_0 &= \mathbf{p}_0 \ , \ \ \mathbf{c}_1 = \mathbf{h}_1^{(0)}(\overline{x}_1^{(0)}) \ , \ \ \mathbf{c}_2 = \mathbf{h}_2^{(0)}(\overline{x}_1^{(0)}, \overline{x}_2^{(0)}) \ , \ \ \mathbf{c}_3 = \mathbf{h}_3^{(0)}(\overline{x}_1^{(0)}, \overline{x}_2^{(0)}, \overline{x}_3^{(0)}) \ , \\ \mathbf{c}_5 &= \mathbf{p}_1 \ , \ \ \mathbf{c}_4 = \mathbf{h}_1^{(1)}(\overline{x}_1^{(1)}) \ , \ \ \mathbf{c}_3 = \mathbf{h}_2^{(1)}(\overline{x}_1^{(1)}, \overline{x}_2^{(1)}) \ , \ \ \mathbf{c}_2 = \mathbf{h}_3^{(1)}(\overline{x}_1^{(1)}, \overline{x}_2^{(1)}, \overline{x}_3^{(1)}) \ , \end{split}$$

As a result, from

$$egin{aligned} \mathbf{c}_0 &= \mathbf{p}_0 \;,\; \mathbf{c}_1 \in \mathbf{H}_1^{(0)} \;, \ \mathbf{c}_2 \in \mathbf{H}_2^{(0)} \cap \mathbf{H}_3^{(1)} \;,\; \mathbf{c}_3 \in \mathbf{H}_3^{(0)} \cap \mathbf{H}_2^{(1)} \;, \ \mathbf{c}_5 &= \mathbf{p}_1 \;,\; \mathbf{c}_4 \in \mathbf{H}_1^{(1)} \;, \end{aligned}$$

a necessary condition for the existence of a quintic G^3 interpolant is

$$\mathbf{H}_{2}^{(0)} \cap \mathbf{H}_{3}^{(1)} \neq \emptyset \ \land \ \mathbf{H}_{3}^{(0)} \cap \mathbf{H}_{2}^{(1)} \neq \emptyset$$

The existence problem of the quintic interpolant can be constructively phrased as finding a $\mathbf{x} \in \mathbf{H}_2^{(0)} \cap \mathbf{H}_3^{(1)}$ point on the osculating plane of \mathbf{D}_0 such that there exists a $\mathbf{y} \in \mathbf{H}_3^{(0)} \cap \mathbf{H}_2^{(1)}$ point on the osculating plane of \mathbf{D}_1 , whose $\overline{x}_1^{(1)}, \overline{x}_2^{(1)}$ tangential coordinates reproduce \mathbf{x} , i.e. $\exists \overline{x}_3^{(1)} : \mathbf{x} =$ $\mathbf{h}_3^{(1)}(\overline{x}_1^{(1)}, \overline{x}_2^{(1)}, \overline{x}_3^{(1)})$.

This can be formalized as the fix-point problem of finding $\mathbf{x} \in \mathbf{H}_2^{(0)} \cap \mathbf{H}_3^{(1)}$ such that

$$\mathbf{x} \in \mathbf{h}_3^{(1)} \big(\mathbf{h}_3^{(0)}(\mathbf{x}) \big) \tag{2.54}$$

where

$$\mathbf{h}_{3}^{(i)}(\mathbf{x}) = \{ \mathbf{y} \in \mathbb{E}^{3} | \overline{x}_{1} = \sqrt{\frac{n-1}{n} \frac{(\mathbf{x} - \mathbf{p}^{(i)}) \cdot \mathbf{n}^{(i)}}{\kappa^{(i)}}} \land$$
$$\overline{x}_{2} = (\mathbf{x} - \mathbf{p}^{(i)}) \cdot \mathbf{t}^{(i)} \land$$
$$\exists \overline{x}_{3} \in \mathbb{R} : \mathbf{y} = \mathbf{h}_{3}^{(i)}(\overline{x}_{1}, \overline{x}_{2}, \overline{x}_{3}) \}$$

and

$$\mathbf{h}_3^{(i)}(A) = \{\mathbf{y} | \exists \mathbf{x} \in A : \mathbf{y} = \mathbf{h}_3^{(i)}(\mathbf{x})\} .$$

The containment in (2.54) can be formulated as an equation by

$$\mathbf{x} = \mathbf{h}_3^{(1)} ig(\mathbf{h}_3^{(0)}(\mathbf{x}) ig) \cap \mathbf{S}^{(0)}$$
 .

Both $\mathbf{h}_{3}^{(i)}$ maps are continuous, however, in general, they do not fulfill existence conditions of neither the Brouwer, nor the Schauder fixed-point theorem since the domain of the problem is an open subset of \mathbb{E}^{3} .

Still, fixed-point iterations provide numerical means to compute the control data of a quintic interpolant, since any fixpoint can be chosen as c_2 , which in turn determines c_3 , hence all the control points of the curve.

2.7.2 Indirect methods

These methods consist of two steps:

- 1. For each \mathbf{D}_i : construct a $\mathbf{p}_i(t) : \mathbb{R} \to \mathbb{E}^3$ basic curve that reconstructs \mathbf{D}_i at t = 0
- 2. For each $\mathbf{D}_i, \mathbf{D}_{i+1}$ segment: with an appropriate $f(t) : \mathbb{R} \to \mathbb{R}$ blending function, form the two-endpoint interpolant to \mathbf{D}_i and \mathbf{D}_{i+1} by

$$\mathbf{p}_{i,i+1}(t) = f(t) \cdot \mathbf{p}_i(t) + (1 - f(t)) \cdot \mathbf{p}_{i+1}(t-1)$$

The advantage of these methods is that they can be easily incorporated into design interfaces via the basic curves, see Figure 2.6. In a prior work, we have shown how a control circle hierarchy based design process can be used to create aesthetically pleasing curves [22], [23], illustrated in Figure 1.1. First, let us consider the construction of basic curves. The cubic

$$\mathbf{p}_{i}(t) = \mathbf{p}_{i} + [\mathbf{t}_{i}, \mathbf{n}_{i}, \mathbf{b}_{i}] \begin{bmatrix} t \\ \frac{\kappa_{i}}{2}t^{2} + \frac{\hat{\kappa}'_{i}}{6}t^{3} \\ \frac{\kappa_{i}\tau_{i}}{6}t^{3} \end{bmatrix}, \quad t \in \mathbb{R}$$
(2.55)

reconstructs a given $\mathbf{D}_i = (\mathbf{p}_i; \mathbf{t}_i, \mathbf{n}_i, \mathbf{b}_i; \kappa_i; \hat{\kappa}'_i, \tau_i)$ third order GH data tuple at t = 0, if $\kappa_i \neq 0$. This can be seen immediately by applying Theorem 2 to the derivatives of $\mathbf{p}_i(t)$.

If $\kappa_i = 0$ and $\hat{\kappa}'_i \neq 0$, reconstruction of a $\tau_i \neq 0$ torsion requires a quartic polynomial. Similarly, if $\kappa_i = \hat{\kappa}'_i = 0$, a quintic curve is required to reconstruct \mathbf{D}_i , provided $\hat{\kappa}''_i \neq 0$. Since the latter is not constrained by \mathbf{D}_i , we can always give it a non-zero value.

The following table summarizes which cubic, quartic, and quintic polynomials reconstruct \mathbf{D}_i , depending on the curvatures:

κ	$\hat{\kappa}'$	τ	polynomial curve
$\neq 0$	$\neq 0$	$\in \mathbb{R}$	$\begin{bmatrix} t\\ \frac{\kappa_i}{2}t^2 + \frac{\hat{\kappa}'_i}{6}t^3\\ \frac{\kappa_i\tau_i}{6}t^3 \end{bmatrix}$
0	$\in \mathbb{R}$	0	$\begin{bmatrix} t \\ \frac{\hat{\kappa}'_i}{6} t^3 \\ 0 \end{bmatrix}$
0	$\neq 0$	$\in \mathbb{R} - \{0\}$	$\begin{bmatrix} t\\ \frac{\hat{\kappa}_i'}{6}t^3 + \frac{\hat{\kappa}_i''}{24}t^4\\ \frac{\hat{\kappa}_i'\tau_i}{12}t^4 \end{bmatrix}$
0	0	$\in \mathbb{R} - \{0\}$	$\begin{bmatrix} t \\ \frac{\hat{\kappa}''_{1}}{24}t^{4} + \frac{\hat{\kappa}'''_{1}}{120}t^{5} \\ \frac{\hat{\kappa}''_{1}\tau_{i}}{40}t^{5} \end{bmatrix}$

The proof of these reconstruction properties can be found in Appendix B.

Next, an appropriate **geometric blending function construction** has to be formulated. In [17], Hartmann proposes the use of the following rational G^n blending function:

$$f_{n,\mu} = \frac{\mu(1-t)^{n+1}}{\mu(1-t)^{n+1} + (1-\mu)t^{n+1}} , t \in [0,1]$$

where $n \ge 0$, and $\mu \in (0, 1)$ is a design parameter called the *balance* of blending function $f_{n,\mu}$.

More generally, any C^n function that satisfies

$$f(0) = 1 , f(1) = 0 ,$$
 (2.56)

$$f^{(k)}(0) = f^{(k)}(1) = 0, k = 1, ..., n$$
(2.57)

is a C^n blending function and as such, $f(t) \cdot \mathbf{p}_i(t) + (1 - f(t)) \cdot \mathbf{p}_{i+1}(t)$ has a C^n contact with $\mathbf{p}_i(t)$ and $\mathbf{p}_{i+1}(t)$ at $\mathbf{p}_{i,i+1}(0)$ and $\mathbf{p}_{i,i+1}(1)$, respectively. In turn, this means that

$$\mathbf{p}_{i,i+1}^{(k)}(0) = \mathbf{p}_{i}^{(k)}(0)$$
$$\mathbf{p}_{i,i+1}^{(k)}(1) = \mathbf{p}_{i+1}^{(k)}(0)$$

k = 0, ...n, that is, $\mathbf{p}_{i,i+1}(t)$ also reconstructs \mathbf{D}_i and \mathbf{D}_{i+1} at t = 0 and t = 1.

From (2.56)-(2.56), an integral quintic blending function can be found for the construction of second order GH interpolants as

$$f(t) = -6t^5 + 15t^4 - 10t^3 + 1 ,$$

while a septic blending function, for third order GH blending, has the form of

$$f(t) = 20t^7 - 70t^6 + 84t^5 - 35t^4 + 1$$

These blending functions are readily found by Hermite interpolation.

If $\kappa_i, \kappa_{i+1} \neq 0$, each GH interpolant segment consists of a rational septic curve using Hartmann's rational blending function. Integral septic Hermite blending functions yield a degree 10 GH interpolant segment.

The corresponding direct solution was a degree 7 integral polynomial, that is, these indirect methods yield either rational or higher degree solutions.

This is an unfortunate disadvantage of indirect methods: even though specifying higher order geometric input can be done intuitively via using basic curves as *control shapes*, the resulting curve might be of higher complexity.

Now, I propose an algorithm for blending polynomial basic curves in such a way that the result matches the degree of the matrix-inversion based direct methods. The algorithm first converts the $\mathbf{p}_i(t)$, $\mathbf{p}_{i+1}(t)$ basic polynomials to Bézier form. The degree of these Bézier basic curves are then elevated to the degree that is guaranteed to offer a solution, specified by Theorem 7. After degree elevation, all control points that are affected by reconstruction of \mathbf{D}_i and \mathbf{D}_{i+1} are independent of each other. Together, they form a degree $n_1+n_2+1+h_0+h_1$ Bézier curve that reconstructs \mathbf{D}_i and \mathbf{D}_{i+1} at t = 0 and t = 1.

Algorithm 4 Parabolic <i>n</i> -th order GH interpolant			
1:	procedure CREATEINTERPOLANT($\mathbf{D}_0, \mathbf{D}_1, \overrightarrow{x}_1, \overleftarrow{x}_1$)		
2:	$\mathbf{p}_0(t) \leftarrow GetParabola(\mathbf{D}_0, \overrightarrow{x}_1)$		
3:	$\mathbf{p}_1(t) \leftarrow GetParabola(\mathbf{D}_1, \overleftarrow{x}_1)$		
4:	$m \leftarrow n_1 + n_2 + 1 + h_0 + h_1$		
5:	Convert:		
6:	$\mathbf{b}_0(t) \leftarrow \mathit{ToBezier}(\mathbf{p}_0(t))$		
7:	$\mathbf{b}_1(t) \leftarrow \mathit{ToBezier}(\mathbf{p}_1(t))$		
8:	Degree Elevate:		
9:	$\mathbf{b}(t) \leftarrow DegreeElevateTo(\mathbf{b}_0(t), m)$		
10:	$\mathbf{c}(t) \leftarrow DegreeElevateTo(\mathbf{b}_1(t), m)$		
11:	Combine:		
12:	$\mathbf{d}_0,,\mathbf{d}_{n_1+h_0} \leftarrow \mathbf{b}_0,,\mathbf{b}_{n_1+h_0}$		
13:	$\mathbf{d}_{n_1+h_0+1},,\mathbf{d}_{n_1+n_2+1+h_0+h_1} \leftarrow \mathbf{c}_{n_1+h_0+1},,\mathbf{c}_{n_1+n_2+1+h_0+h_1}$		
14:	Return:		
15:	$\mathbf{d}(t)$, defined by control points $\mathbf{d}_0,, \mathbf{d}_{2n+1+h_0+h_1}$		

That is, Algorithm 4 results in a degree $n_1 + n_2 + 1 + h_0 + h_1$ Bézier solution to given n_1 -th and n_2 -th order GH data tuples $\mathbf{D}_0, \mathbf{D}_1$, with two degrees of freedom, $\overrightarrow{x}_1, \overleftarrow{x}_1 > 0$. In the case of symmetric third order GH interpolation, if $\kappa_i, \kappa_{i+1} \neq 0$, the solution is of degree 7.

In Algorithm 4, the function *GetParabola* returns a basic curve reconstructing the argument GH data tuple, with the given scalar as the tangential coordinate of its first derivative. This way the user can control the magnitude of the speed at endpoints.

These $\overrightarrow{x}_1, \overleftarrow{x}_1 > 0$ degrees of freedom can be also considered as the design parameters of the algorithm, specifying the relative weights of their respective basic curve in the formulation of the GH interpolant segment.

In general, basic curves provide simple means to specify higher order geometric input. Their simplicity can be taken advantage of even if one is using parametrization optimization methods to compute a GH interpolant.

Chapter 3

Geometric Hermite interpolation of Surfaces

3.1 Overview

Interpolating a set of points with parametric surfaces has been an extensively studied topic in the literature. It is not only prevalent in modeling, it has important applications in reconstruction, design, visualization, and other areas.

Incorporating first order geometric data of surfaces into this problem yields point-normal interpolation, which also gathered attention in the literature [52], [11]. However, references to generalizations to second and higher order geometric data is much less often found, these quantities are usually involved only indirectly via energy minimizing and fairing functionals [33].

In many applications, this absence of attention is justified by the circumstances of data acquisition: if the set of points are obtained by measurements, estimating higher order geometric quantities of the scanned surface is often discouraged by measurement errors. However, in modeling and design, these quantities can be easily made available, even without explicit specification of these data.

Before formalizing the general geometric Hermite surface interpolation problem, I investigate a case study of second order GH surface interpolation in the following section. The exploration of this specific topic is absent from the literature, but it provides important insight into common problems of GH interpolation of surfaces. The contents of Section 3.2 are from our paper [50]. The general formulation follows in Section 3.3, and the analogue of the parabola GH curve interpolant is presented in the final Section of this chapter.

Throughout this chapter all parametric surfaces are assumed to be regular and sufficiently many times continuously differentiable.

3.2 Second order geometric Hermite interpolation of surfaces

3.2.1 Introduction

The problem of second order GH interpolation has been studied for a wide range of curves, including integral and rational Bézier curves ([41]), Pythagorean Hodograph curves ([2]), Pythagorean Hodograph spiral curves ([54], [16]).

Surfaces that interpolate position and normal data, which can be considered as the generalization of first order geometric Hermite curve interpolation, have been studied in the literature, e.g. ([36]), ([42]).

Moreton used scattered data interpolation techniques to interpolate positional, surface normal and principal curvature data by solving nonlinear optimization of fairness and engergy functionals subject to geometric reconstruction constraints [33]. His paper, however, did not give a geometric characterization of the reconstruction problem, instead, it focused on the incorporation of functional minimization into the design process.

In this section, I characterize a generalization of second order geometric Hermite interpolation to surfaces by examining the conditions for reconstructing G^2 surface base point data, that is, position, normal, principal curvature, and principal direction data, at a given point of a regular parametric surface.

3.2.2 Conditions for second order GH interpolation

Let us consider a regular parametric surface $\mathbf{s} : \mathbb{R}^2 \to \mathbb{E}^3$ and a (u_0, v_0) point in its domain, and let there be given a G^2 base point data tuple

$$D = (\mathbf{p}, \mathbf{m}, \mathbf{t}_1, \mathbf{t}_2, \kappa_1, \kappa_2), \qquad (3.1)$$
which is to be reconstructed at $\mathbf{s}(u_0, v_0)$, where $\mathbf{p} \in \mathbb{E}^3$ denotes a point in the Euclidean space, $\mathbf{m} \in \mathbb{R}^3$ is the unit surface normal, $\kappa_1, \kappa_2 \in \mathbb{R}$ are principal curvature values, and $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{R}^3$ are corresponding principal directions, $|\mathbf{m}| = |\mathbf{t}_1| = |\mathbf{t}_2| = 1$, and κ_1 is the minimum and κ_2 is the maximum normal curvature. Without loss of generality, we assume that $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$ form a right-handed orthonormal base of \mathbb{R}^3 , i.e. $\mathbf{m} = \mathbf{t}_1 \times \mathbf{t}_2$.

Let us examine what local constraints the regular parametric surface $\mathbf{s}(u, v)$ should satisfy in order to reconstruct the second order geometric data of \mathbf{D} at parameter (u_0, v_0) .

Analogously to the curve case, our goal is to find constraints on the partial derivatives of the surface, such that when satisfied, they guarantee the reconstruction of the given geometric Hermite data.

• Position reconstruction: interpolation of position **p** given in **D** requires that

$$\mathbf{s}(u_0, v_0) = \mathbf{p} \tag{3.2}$$

should hold at (u_0, v_0) for the surface $\mathbf{s}(u, v)$.

• Normal reconstruction: let $\mathbf{s}_u(u, v)$ and $\mathbf{s}_v(u, v)$ denote the u and v partial derivatives and let

$$\mathbf{m}(u,v) = \frac{\mathbf{s}_u(u,v) \times \mathbf{s}_v(u,v)}{|\mathbf{s}_u(u,v) \times \mathbf{s}_v(u,v)|}$$

be the unit surface normal function of \mathbf{s} at (u, v). Interpolation of the normal $\mathbf{m} \in \mathbf{D}$ of the base point data at (u_0, v_0) means that $\mathbf{m}(u_0, v_0) = \mathbf{m}$ should hold.

To simplify notation, let \mathbf{s}_u and \mathbf{s}_v denote the u and v partial derivatives of $\mathbf{s}(u, v)$ at (u_0, v_0) , i.e. $\mathbf{s}_u = \mathbf{s}_u(u_0, v_0)$ and $\mathbf{s}_v = \mathbf{s}_v(u_0, v_0)$, and let $\mathbf{T}_{\mathbf{D}}$ denote the tangent plane corresponding to the data specified in \mathbf{D} :

$$\mathbf{T}_{\mathbf{D}} = \{ \mathbf{x} \in \mathbb{E}^3 \mid (\mathbf{x} - \mathbf{p}) \cdot \mathbf{m} = 0 \} .$$

To match the surface normal with \mathbf{m} , the partial derivatives \mathbf{s}_u and \mathbf{s}_v should lie in the $\mathbf{T}_{\mathbf{D}}$ plane and their cross product should have the same direction as \mathbf{m} . The latter can be written as prescribing that their dot product should be positive:

$$(\mathbf{s}_u \times \mathbf{s}_v) \cdot \mathbf{m} > 0 \tag{3.3}$$

Proposition 10 A regular parametric surface $\mathbf{s} : \mathbb{R}^2 \to \mathbb{E}^3$ interpolates the normal \mathbf{m} of the G^2 base point data **if and only if** there exist $(x_u, y_u) \in \mathbb{R}^2$ and $(x_v, y_v) \in \mathbb{R}^2$, such that

$$\mathbf{s}_u(u_0, v_0) = x_u \mathbf{t}_1 + y_u \mathbf{t}_2 \tag{3.4}$$

$$\mathbf{s}_v(u_0, v_0) = x_v \mathbf{t}_1 + y_v \mathbf{t}_2 \tag{3.5}$$

$$0 < x_u y_v - x_v y_u \tag{3.6}$$

Proof. (3.4) and (3.5) stipulate that the partial derivatives \mathbf{s}_u and \mathbf{s}_v can be expressed in the $(\mathbf{t}_1, \mathbf{t}_2)$ orthonormal basis of $\mathbf{T}_{\mathbf{D}}$, such that their length is greater than zero and (3.6) is the expression of condition (3.3) in terms of the partial derivatives' coordinates with respect to $(\mathbf{t}_1, \mathbf{t}_2)$, utilising that $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$ form a right-handed orthonormal basis.

Principal curvature value and direction reconstruction: let us find what conditions should hold so that a regular parametric surface s(u, v) has prescribed κ₁, κ₂ principal curvature values and t₁, t₂ principal directions at (u₀, v₀).

Let us assume that the position and normal reconstruction conditions are already satisfied at (u_0, v_0) , and that the principal curvature values and directions at (u_0, v_0) are $\kappa_1, \kappa_2 \in \mathbf{D}$ and $\mathbf{t}_1, \mathbf{t}_2 \in \mathbf{D}$.

Now we show that, by choosing three specific independent tangent directions, the second order partial derivates $\mathbf{s}_{uu} = \mathbf{s}_{uu}(u_0, v_0)$, $\mathbf{s}_{uv} = \mathbf{s}_{uv}(u_0, v_0)$, and $\mathbf{s}_{vv} = \mathbf{s}_{vv}(u_0, v_0)$ are subject to simple geometric constraints.

Given a tangent vector in the tangent plane $\mathbf{T}_{\mathbf{D}}$, with coordinates $(du, dv) \neq \mathbf{0}$ in the $(\mathbf{s}_u, \mathbf{s}_v)$ skew basis of the tangent plane, the normal curvature along that tangent at (u_0, v_0) can be computed as [9]

$$\kappa(du, dv) = \frac{II}{I} = \frac{L \cdot du^2 + 2M \cdot du \cdot dv + N \cdot dv^2}{E \cdot du^2 + 2F \cdot du \cdot dv + G \cdot dv^2}$$
(3.7)

The coefficients of the first and second fundamental forms above are com-

puted at (u_0, v_0) as

$$E = \mathbf{s}_u \cdot \mathbf{s}_u = x_u^2 + y_u^2$$

$$F = \mathbf{s}_u \cdot \mathbf{s}_v = x_u x_v + y_u y_v$$

$$G = \mathbf{s}_v \cdot \mathbf{s}_v = x_v^2 + y_v^2$$

$$L = \mathbf{s}_{uu} \cdot \mathbf{m} = z_{uu}$$

$$M = \mathbf{s}_{uv} \cdot \mathbf{m} = z_{uv}$$

$$N = \mathbf{s}_{vv} \cdot \mathbf{m} = z_{vv}$$

where partial derivatives, with respect to the $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$ Darboux frame are

$$s_u = x_u \mathbf{t}_1 + y_u \mathbf{t}_2$$

$$s_v = x_v \mathbf{t}_1 + y_v \mathbf{t}_2$$

$$s_{uu} = x_{uu} \mathbf{t}_1 + y_{uu} \mathbf{t}_2 + z_{uu} \mathbf{m}$$

$$s_{uv} = x_{uv} \mathbf{t}_1 + y_{uv} \mathbf{t}_2 + z_{uv} \mathbf{m}$$

$$s_{vv} = x_{vv} \mathbf{t}_1 + y_{vv} \mathbf{t}_2 + z_{vv} \mathbf{m}$$

If principal curvatures and corresponding directions are known, the normal curvature along a tangent direction is computed using the angle α between the tangent direction and the principal direction \mathbf{t}_1 by Euler's theorem:

$$\kappa(\alpha) = \kappa_1 \cos^2 \alpha + \kappa_2 \sin^2 \alpha \tag{3.8}$$

By choosing a $\mathbf{d} \in \mathbf{T}_{\mathbf{D}}$ direction in the tangent plane, we can use both (3.7) and (3.8) to compute the normal curvature corresponding to \mathbf{d} , either by using its coordinates in the $(\mathbf{s}_u, \mathbf{s}_v)$ or the $(\mathbf{t}_1, \mathbf{t}_2)$ basis. Both expressions should yield the same, that is

$$\frac{L \cdot du^2 + 2M \cdot du \cdot dv + N \cdot dv^2}{E \cdot du^2 + 2F \cdot du \cdot dv + G \cdot dv^2} = \kappa_1 \cos^2 \alpha + \kappa_2 \sin^2 \alpha$$
(3.9)



Figure 3.1: Normal curvature of a curve on surface with given tangent vector $\mathbf{d} = \cos \alpha \cdot \mathbf{t}_1 + \sin \alpha \cdot \mathbf{t}_2 = du \cdot \mathbf{s}_u + dv \cdot \mathbf{s}_v \in \mathbf{T}_{\mathbf{D}}$ can be computed by (3.7) in the $(\mathbf{t}_1, \mathbf{t}_2)$ orthonormal basis of $\mathbf{T}_{\mathbf{D}}$, and by (3.8) in the $(\mathbf{s}_u, \mathbf{s}_v)$ skew basis of $\mathbf{T}_{\mathbf{D}}$.

should hold, where $\alpha = \tan^{-1} \frac{du \cdot y_u + dv \cdot y_v}{du \cdot x_u + dv \cdot x_v}$ since

$$\mathbf{d} = du \cdot \mathbf{s}_u + dv \cdot \mathbf{s}_v$$

= $du \cdot (x_u \mathbf{t}_1 + y_u \mathbf{t}_2) + dv \cdot (x_v \mathbf{t}_1 + y_v \mathbf{t}_2)$
= $(du \cdot x_u + dv \cdot x_v)\mathbf{t}_1 + (du \cdot y_u + dv \cdot y_v)\mathbf{t}_2$

Let us substitute the trigonometric functions in (3.9) by

$$\cos \alpha = \frac{du \cdot x_u + dv \cdot x_v}{\sqrt{(du \cdot x_u + dv \cdot x_v)^2 + (du \cdot y_u + dv \cdot y_v)^2}}$$
$$\sin \alpha = \frac{du \cdot y_u + dv \cdot y_v}{\sqrt{(du \cdot x_u + dv \cdot x_v)^2 + (du \cdot y_u + dv \cdot y_v)^2}}$$

after which (3.9) can be written as

$$\frac{II}{I} = \frac{Ldu^2 + 2Mdudv + Ndv^2}{Edu^2 + 2Fdudv + Gdv^2} = \frac{\kappa_1(du \cdot x_u + dv \cdot x_v)^2 + \kappa_2(du \cdot y_u + dv \cdot y_v)^2}{(du \cdot x_u + dv \cdot x_v)^2 + (du \cdot y_u + dv \cdot y_v)^2}$$
(3.10)

Note that

$$I = E \cdot du^{2} + 2F \cdot du \cdot dv + G \cdot dv^{2}$$

= $(x_{u}^{2} + y_{u}^{2}) \cdot du^{2} + 2(x_{u}x_{v} + y_{u}y_{v}) \cdot du \cdot dv + (x_{v}^{2} + y_{v}^{2}) \cdot dv^{2}$
= $(du \cdot x_{u} + dv \cdot x_{v})^{2} + (du \cdot y_{u} + dv \cdot y_{v})^{2}$

that is, the denominators are equal, which simplifies (3.10) into

$$II = L \cdot du^2 + 2M \cdot du \cdot dv + N \cdot dv^2 = \kappa_1 (du \cdot x_u + dv \cdot x_v)^2 + \kappa_2 (du \cdot y_u + dv \cdot y_v)^2$$

$$(3.11)$$

By evaluating (3.11) along the \mathbf{s}_u , $\mathbf{s}_u + \mathbf{s}_v$, and \mathbf{s}_v tangents, the following constraints are found on the second order partial derivatives:

• $\mathbf{d} = \mathbf{s}_u$: in this case (du, dv) = (1, 0) and, since $z_{uu} = L$

$$L = z_{uu} = \kappa_1 x_u^2 + \kappa_2 y_u^2$$

that is, for the $\mathbf{s}_{uu} = x_{uu}\mathbf{t}_1 + y_{uu}\mathbf{t}_2 + z_{uu}\mathbf{m}$ partial derivative

$$z_{uu} = \kappa_1 x_u^2 + \kappa_2 y_u^2$$

should hold.

• $\mathbf{d} = \mathbf{s}_{v}$: in this case (du, dv) = (0, 1) and, since $N = z_{vv}$

$$N = z_{vv} = \kappa_1 x_v^2 + \kappa_2 y_v^2 ,$$

that is, for the $\mathbf{s}_{vv} = x_{vv}\mathbf{t}_1 + y_{vv}\mathbf{t}_2 + z_{vv}\mathbf{m}$ partial derivative

$$z_{vv} = \kappa_1 x_v^2 + \kappa_2 y_v^2$$

should hold.

• $\mathbf{d} = \mathbf{s}_u + \mathbf{s}_v$: in this case (du, dv) = (1, 1) giving

$$L + 2M + N = \kappa_1 (x_u + x_v)^2 + \kappa_2 (y_u + y_v)^2$$

and since $M = z_{uv}$,

$$z_{uv} = \frac{1}{2} \left(\kappa_1 (x_u + x_v)^2 + \kappa_2 (y_u + y_v)^2 - L - N \right)$$

= $\frac{1}{2} \left(\kappa_1 (x_u + x_v)^2 + \kappa_2 (y_u + y_v)^2 - (\kappa_1 x_u^2 + \kappa_2 y_u^2) - (\kappa_1 x_v^2 + \kappa_2 y_v^2) \right)$
= $\kappa_1 x_u x_v + \kappa_2 y_u y_v$

should hold.

To formulate the conditions for curvature reconstruction, we utilize the Three Tangents Theorem proven by Pegna and Wolter in [29], which states the following:

Theorem 11 (Three Tangents Theorem) Let there be given two C^2 smooth surfaces $\mathbf{s}_1, \mathbf{s}_2 : \mathbb{R}^2 \to \mathbb{E}^3$ that are tangent at a point $\mathbf{p}_0 \in \mathbb{E}^3$. The two surfaces have the same normal curvatures along any tangent direction at that point *if* **and only if** they have the same normal curvatures along three tangent directions of which any two are linearly independent.

The theorem is used to prove the following

Theorem 12 Let $\mathbf{s} : \mathbb{R}^2 \to \mathbb{E}^3$ be a regular parametric surface that interpolates the position and normal data of the G^2 base point data \mathbf{D} at (u_0, v_0) . The surface interpolates the principal curvature data at (u_0, v_0) as well **if and only if** the following hold for the second order partial derivatives:

$$z_{uu} = \kappa_1 x_u^2 + \kappa_2 y_u^2 \tag{3.12}$$

$$z_{uv} = \kappa_1 x_u x_v + \kappa_2 y_u y_v \tag{3.13}$$

$$z_{vv} = \kappa_1 x_v^2 + \kappa_2 y_v^2 \tag{3.14}$$

Proof. \Rightarrow : this follows directly from the derivation above. \Leftarrow : since $\mathbf{s}(u, v)$ is regular, the tangent directions \mathbf{s}_u , $\mathbf{s}_u + \mathbf{s}_v$, and \mathbf{s}_v are pairwise linearly independent.

If the second partial derivatives of $\mathbf{s}(u, v)$ are such that (3.12)-(3.14) hold, the normal curvatures of $\mathbf{s}(u, v)$ along \mathbf{s}_u , $\mathbf{s}_u + \mathbf{s}_v$, \mathbf{s}_v equal to the normal curvatures specified by Euler's theorem. By the Three Tangents Theorem this means that $\mathbf{s}(u, v)$ has the same normal curvatures along all tangent directions as specified by the Euler theorem, which proves that at (u_0, v_0) surface $\mathbf{s}(u, v)$ has principal curvature values κ_1, κ_2 and corresponding principal directions $\mathbf{t}_1, \mathbf{t}_2$, given in \mathbf{D} .

Intuitively, Theorem 12 has the following geometric interpretation: if the parametrization of the tangent plane is given, that is, the \mathbf{s}_u and \mathbf{s}_v partial derivatives are set, the reconstruction of principal curvature relations poses restrictions on the second partial derivatives \mathbf{s}_{uu} , \mathbf{s}_{uv} , and \mathbf{s}_{vv} along the surface normal \mathbf{m} only, with respect to the local $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$ base. Each of the second partial derivative vectors must lie in a specific offset plane of the tangent plane, and they can be moved parallel to the tangent plane without affecting the principal curvature reconstruction. This is analogues to the fact that geometric continuity in curves realized a scalar degree of freedom along the tangential Frenet coordinate of derivatives.

3.2.3 Quadrilateral Bézier patches

3.2.3.1 Four corner GH interpolation

Let there be given four G^2 base point data tuples

$$\mathbf{D}^{(ij)} = (\mathbf{p}^{(ij)}, \mathbf{m}^{(ij)}, \mathbf{t}_1^{(ij)}, \mathbf{t}_2^{(ij)}, \kappa_1^{(ij)}, \kappa_2^{(ij)}),$$

i, j = 0, 1. Four corner second order geometric Hermite interpolation aims at finding a

$$\mathbf{b}(u,v) = \sum_{j=0}^{m} \sum_{i=0}^{n} \mathbf{b}_{ij} B_i^n(u) B_j^m(v),$$

quadrilateral Bézier surface, $(u, v) \in [0, 1]^2$, $\mathbf{b}_{ij} \in \mathbb{E}^3$, i = 0, ..., n, j = 0, ..., m, $n, m \geq 2$, that reconstructs the $\mathbf{D}^{(ij)}$ base point data at its parametric corners, that is a surface for which

$$\mathbf{b}(i,j) = \mathbf{p}^{(ij)} \tag{3.15}$$

$$\mathbf{m}(i,j) = \mathbf{m}^{(ij)} \tag{3.16}$$

$$\kappa_1(i,j) = \kappa_1^{(ij)} \tag{3.17}$$

$$\kappa_2(i,j) = \kappa_2^{(ij)} \tag{3.18}$$

$$\mathbf{t}_1(i,j) = \mathbf{t}_1^{(ij)} \tag{3.19}$$

$$\mathbf{t}_2(i,j) = \mathbf{t}_2^{(ij)} \tag{3.20}$$

holds for i, j = 0, 1, where $\kappa_1(u, v), \kappa_2(u, v)$ denote the principal curvature values, $\mathbf{t}_1(u, v), \mathbf{t}_2(u, v)$ denote the corresponding principal directions of $\mathbf{b}(u, v)$ at (u, v).

Let us begin the inspection of the problem by considering the parametric corner (u, v) = (0, 0), and let us find the control points necessary for the reconstruction of the second order geometric Hermite data in $\mathbf{D}^{(00)}$. For the sake of simplicity, we omit the upper indices of $\mathbf{D}^{(00)}$ and its members, i.e. $\mathbf{D} = \mathbf{D}^{(00)}, \mathbf{p} = \mathbf{p}^{(00)}$, etc. in what follows.

Position reconstruction (3.15) states that $\mathbf{b}(0,0) = \mathbf{p}$ should hold, imposing the

$$b_{00} = p$$

constraint on control point \mathbf{b}_{00} , which is the only control point required for the position interpolation.

Normal reconstruction (3.16) states that $\mathbf{m}(0,0) = \mathbf{m}$ should hold. The normal at (u, v) = (0, 0) is computed as

$$\mathbf{m}(0,0) = \frac{\Delta^{10}\mathbf{b}_{00} \times \Delta^{01}\mathbf{b}_{00}}{|\Delta^{10}\mathbf{b}_{00} \times \Delta^{01}\mathbf{b}_{00}|}$$

for which control points \mathbf{b}_{00} , \mathbf{b}_{10} , \mathbf{b}_{01} are used.

Fulfilment of principal curvature reconstruction conditions (3.17) - (3.20) require the first and second fundamental forms, which, in the case of the



Figure 3.2: The control points of a Bézier surface that determine the position, tangent plane, and principal curvature relations at (u, v) = (0, 0)

(u, v) = (0, 0) corner, are written as

$$E(u, v) = \mathbf{b}_{u} \cdot \mathbf{b}_{u} = n^{2} \Delta^{10} \mathbf{b}_{00} \cdot \Delta^{10} \mathbf{b}_{00}$$

$$F(u, v) = \mathbf{b}_{u} \cdot \mathbf{b}_{v} = nm \Delta^{10} \mathbf{b}_{00} \cdot \Delta^{01} \mathbf{b}_{00}$$

$$G(u, v) = \mathbf{b}_{v} \cdot \mathbf{b}_{v} = m^{2} \Delta^{01} \mathbf{b}_{00} \cdot \Delta^{01} \mathbf{b}_{00}$$

$$L(u, v) = \mathbf{b}_{uu} \cdot \mathbf{m} = n(n-1) \Delta^{20} \mathbf{b}_{00} \cdot \mathbf{m}$$

$$M(u, v) = \mathbf{b}_{uv} \cdot \mathbf{m} = nm \Delta^{11} \mathbf{b}_{00} \cdot \mathbf{m}$$

$$N(u, v) = \mathbf{b}_{vv} \cdot \mathbf{m} = m(m-1) \Delta^{02} \mathbf{b}_{00} \cdot \mathbf{m}$$

The control points required for a second order geometric Hermite reconstruction at (u, v) = (0, 0) are $\mathbf{b}_{00}, \mathbf{b}_{10}, \mathbf{b}_{01}, \mathbf{b}_{20}, \mathbf{b}_{11}, \mathbf{b}_{02}$, shown in figure 3.2. The reconstruction poses 8 scalar constraints on these six control points.

To make these constraints explicit, let us express the control points in the coordinate system $(\mathbf{p}; \mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$ – that is, the coordinate system centered at \mathbf{p} and having the Darboux-frame as axes – as

$$\mathbf{b}_{ij} = \mathbf{p} + \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \mathbf{m} \end{bmatrix} \begin{bmatrix} \overline{x}_{ij} \\ \overline{y}_{ij} \\ \overline{z}_{ij} \end{bmatrix}, i, j \in \mathbb{N}, i+j \le 2,$$

Using the results of Propositions 10 and Theorem 12, and the expression of partial derivatives with respect to control points, the control points take the form

$$\mathbf{b}_{00} = \mathbf{p} \tag{3.21}$$

$$\mathbf{b}_{10} = \mathbf{p} + \frac{x_u}{n} \mathbf{t}_1 + \frac{y_u}{n} \mathbf{t}_2 \tag{3.22}$$

$$\mathbf{b}_{01} = \mathbf{p} + \frac{x_v}{m} \mathbf{t}_1 + \frac{y_v}{m} \mathbf{t}_2 \tag{3.23}$$

$$\mathbf{b}_{20} = \mathbf{p} + \frac{x_{uu} + 2(n-1)x_u}{n(n-1)}\mathbf{t}_1 + \frac{y_{uu} + 2(n-1)y_u}{n(n-1)}\mathbf{t}_2 + \frac{z_{uu}}{n(n-1)}\mathbf{m}$$
(3.24)

$$\mathbf{b}_{11} = \mathbf{p} + \frac{x_{uv} + mx_u + nx_v}{nm} \mathbf{t}_1 + \frac{y_{uv} + my_u + ny_v}{nm} \mathbf{t}_2 + \frac{z_{uv}}{nm} \mathbf{m}$$
(3.25)

$$\mathbf{b}_{02} = \mathbf{p} + \frac{x_{vv} + 2(m-1)x_v}{m(m-1)}\mathbf{t}_1 + \frac{y_{vv} + 2(m-1)y_v}{m(m-1)}\mathbf{t}_2 + \frac{z_{vv}}{m(m-1)}\mathbf{m} \quad (3.26)$$

where the coordinates (x_u, y_u) , (x_v, y_v) are subject to (3.6), and z_{uu}, z_{uv}, z_{vv} should satisfy (3.12)-(3.14). The coordinates (x_{uu}, y_{uu}) , (x_{uv}, y_{uv}) , and (x_{vv}, y_{vv}) can be chosen freely.

To give a more geometric interpretation of (3.24)-(3.26), let us denote the lifted tangent planes of **D** along the surface normal by

$$\mathbf{M}^{(ij)}(m) = \{ \mathbf{x} \in \mathbb{E}^3 | (\mathbf{x} - \mathbf{p}^{(ij)}) \cdot \mathbf{m}^{(ij)} = m \}.$$

In the case of the (u, v) = (0, 0) corner, (3.24) - (3.26) can be written as

$$\mathbf{b}_{20} \in \mathbf{M}(\overline{z}_{20}) \tag{3.27}$$

$$\mathbf{b}_{11} \in \mathbf{M}(\overline{z}_{11}) \tag{3.28}$$

$$\mathbf{b}_{02} \in \mathbf{M}(\overline{z}_{02}) \tag{3.29}$$

where we define

$$\overline{z}_{20} = \frac{z_{uu}}{n(n-1)} \tag{3.30}$$

$$\overline{z}_{11} = \frac{z_{uv}}{nm} \tag{3.31}$$

$$\bar{z}_{02} = \frac{z_{vv}}{m(m-1)} \tag{3.32}$$

We can now turn to the discussion of all four corners. The control points

required for the reconstruction of G^2 base point data are

$$\mathbf{b}_{i \cdot n + (-1)^{i} k, j \cdot m + (-1)^{j} l} \in \mathbb{E}^{3}$$
(3.33)

i, j = 0, 1, and $k + l \le 2$.

Their coordinates can be formulated similarly as in the case of the (u, v) = (0, 0) corner, however, it must be noted that the computation of the offsets of the lifted tangent planes differ in the diagonal tangent directions.

This follows from the fact that in (3.11), the computation of the $z_{uu}^{(ij)}$, $z_{uv}^{(ij)}$, $z_{vv}^{(ij)}$ coordinates of the second partial derivatives at corner (ij) have to be done by parametric directions $((-1)^i, 0)$, $((-1)^i, (-1)^j)$, $(0, (-1)^j)$, respectively. Since

• $(du, dv) = (1, 0) : L = \kappa_1 x_u^2 + \kappa_2 y_u^2 \to L = z_{uu} = \kappa_1 x_u^2 + \kappa_2 y_u^2$

•
$$(du, dv) = (0, 1) : N = \kappa_1 x_v^2 + \kappa_2 y_v^2 \to N = z_{vv} = \kappa_1 x_v^2 + \kappa_2 y_v^2$$

•
$$(du, dv) = (-1, 0) : L = \kappa_1 x_u^2 + \kappa_2 y_u^2 \to L = \kappa_1 x_u^2 + \kappa_2 y_u^2 = z_{uu}$$

•
$$(du, dv) = (0, -1) : N = \kappa_1 x_u^2 + \kappa_2 y_u^2 \to N = \kappa_1 x_v^2 + \kappa_2 y_v^2 = z_{vv}$$

the offset of the lifted tangent planes of the uu and vv partial derivatives (and by that, the planes of the \mathbf{b}_{20} , \mathbf{b}_{02} , etc. control points) is the same in the (1,0),(-1,0), and (0,1), (0,-1) directions, but in the case of the mixed partial derivatives they differ because

•
$$(du, dv) = (1, 1)$$
:

$$z_{uv}(1,1) = \frac{1}{2} \left(\kappa_1 (x_u + x_v)^2 + \kappa_2 (y_u + y_v)^2 - L - N \right)$$

= $\kappa_1 x_u x_v + \kappa_2 y_u y_v$

• (du, dv) = (-1, 1):

$$z_{uv}(-1,1) = -\frac{1}{2} \left(\kappa_1 (-x_u + x_v)^2 + \kappa_2 (-y_u + y_v)^2 - L - N \right)$$

= $-\kappa_1 x_u x_v - \kappa_2 y_u y_v = -z_{uv}(1,1)$



Figure 3.3: Control net of the bi-quintic Bézier patch. The red, blue, green, and azure regions correspond to the control points that are necessary for the reconstruction of G^2 base point data $\mathbf{D}^{(00)}$, $\mathbf{D}^{(10)}$, $\mathbf{D}^{(11)}$, and $\mathbf{D}^{(01)}$.

•
$$(du, dv) = (-1, -1)$$
:
 $z_{uv}(-1, -1) = \frac{1}{2} \left(\kappa_1 (x_u + x_v)^2 + \kappa_2 (y_u + y_v)^2 - L - N \right)$
 $= z_{uv}(1, 1)$

$$(du, dv) = (1, -1):$$

$$z_{uv}(1, -1) = -\frac{1}{2} \left(\kappa_1 (-x_u + x_v)^2 + \kappa_2 (-y_u + y_v)^2 - L - N \right)$$

$$= -\kappa_1 x_u x_v - \kappa_2 y_u y_v = -z_{uv}(1, 1)$$

3.2.3.2 Bi-quintic Bézier patch

The four corner Bézier second order geometric Hermite interpolation can be always solved by bi-quintic integral Bézier surfaces.

Theorem 13 There is always a quadrilateral bi-quintic integral Bézier surface solution for the four corner second order geometric Hermite interpolation problem. **Proof.** To prove the proposition, we have to choose the coordinates of all control points such that the surface reconstructs the prescribed base point data at its parametric corners.

In the m = n = 5 bi-quintic case, the control points required for the reconstruction of corner base point data do not overlap, that is, each corner's six control points can be determined independently, see figure 3.3.

Let us consider the corner (u, v) = (0, 0). The equations (3.21)-(3.26) take the form

$$\begin{aligned} \mathbf{b}_{00} &= \mathbf{p} \\ \mathbf{b}_{10} &= \mathbf{p} + \frac{x_u}{5} \mathbf{t}_1 + \frac{y_u}{5} \mathbf{t}_2 \\ \mathbf{b}_{01} &= \mathbf{p} + \frac{x_v}{5} \mathbf{t}_1 + \frac{y_v}{5} \mathbf{t}_2 \\ \mathbf{b}_{20} &= \mathbf{p} + \frac{x_{uu} + 10x_u}{20} \mathbf{t}_1 + \frac{y_{uu} + 10y_u}{20} \mathbf{t}_2 + \frac{z_{uu}}{20} \mathbf{m} \\ \mathbf{b}_{11} &= \mathbf{p} + \frac{x_{uv} + 5(x_u + x_v)}{25} \mathbf{t}_1 + \frac{y_{uv} + 5(y_u + y_v)}{25} \mathbf{t}_2 + \frac{z_{uv}}{25} \mathbf{m} \\ \mathbf{b}_{02} &= \mathbf{p} + \frac{x_{vv} + 10x_v}{20} \mathbf{t}_1 + \frac{y_{vv} + 10y_v}{20} \mathbf{t}_2 + \frac{z_{vv}}{20} \mathbf{m} \end{aligned}$$

Position reconstruction is satisfied by the choice $\mathbf{b}_{00} = \mathbf{p}$.

We can always find a u and v partial derivative $x_u \mathbf{t}_1 + y_u \mathbf{t}_2$, $x_v \mathbf{t}_1 + y_v \mathbf{t}_2$ that satisfy (3.6), for example \mathbf{t}_1 and \mathbf{t}_2 , which determines the position of control points \mathbf{b}_{10} , \mathbf{b}_{01} . By proposition 10, normal reconstruction is satisfied.

Any point can be choosen for \mathbf{b}_{20} , \mathbf{b}_{11} , \mathbf{b}_{20} from the $\mathbf{M}(\overline{z}_{20})$, $\mathbf{M}(\overline{z}_{11})$, $\mathbf{M}(\overline{z}_{02})$ lifted tangent planes, respectively. By (3.30)-(3.32) and Theorem 12, the curvature relations are also reconstructed. The other corners are analogous.

The 12 control points in the middle of the control polyhedron, not required for any of the corner's G^2 base point data's reconstruction, can be chosen freely.

3.2.3.3 Bi-quartic Bézier patch

The control points required for the G^2 base point reconstruction are not independent in the bi-quartic case. Each of the \mathbf{b}_{20} , \mathbf{b}_{02} , \mathbf{b}_{24} , \mathbf{b}_{42} control points



Figure 3.4: The control points required for base point data reconstruction in the case of a 4x4-es Bézier surface. The red, green, blue, and azure areas are the control points required for $\mathbf{D}^{(00)}, \mathbf{D}^{(10)}, \mathbf{D}^{(01)}, \mathbf{D}^{(11)}$.

depend on two base point data tuples:

$$\mathbf{b}_{20} \in \mathbf{M}^{(00)}(\overline{z}_{20}^{(00)}) \cap \mathbf{M}^{(10)}(\overline{z}_{20}^{(10)}) \tag{3.34}$$

$$\mathbf{b}_{02} \in \mathbf{M}^{(00)}(\overline{z}_{02}^{(00)}) \cap \mathbf{M}^{(01)}(\overline{z}_{02}^{(01)}) \tag{3.35}$$

$$\mathbf{b}_{24} \in \mathbf{M}^{(01)}(\overline{z}_{20}^{(01)}) \cap \mathbf{M}^{(11)}(\overline{z}_{20}^{(11)})$$
(3.36)

$$\mathbf{b}_{42} \in \mathbf{M}^{(10)}(\overline{z}_{20}^{(10)}) \cap \mathbf{M}^{(11)}(\overline{z}_{20}^{(11)}) \tag{3.37}$$

These *overlapping control points* and the control polyhedron are shown in figure 3.4.

Only the overlapping control points and the ones that are used for the first partial derivatives require further inspection, the rest can be chosen as in the previous subsection.

In the following, we examine the overlapping control points, and whether a boundary curve, satisfying the base point data reconstruction constraints, can be constructed between the corresponding base point data tuples.

Let us consider the case of control point \mathbf{b}_{20} , and the v = 0 boundary curve

of $\mathbf{b}(u, v)$. The other overlapping control points, and their boundary curves are analogous.

Equation (3.34) states that \mathbf{b}_{20} should lie on the intersection of $\mathbf{M}^{(00)}(\overline{z}_{20}^{(00)})$ and $\mathbf{M}^{(10)}(\overline{z}_{20}^{(10)})$.

- If $\mathbf{m}^{(00)}$ is not parallel to $\mathbf{m}^{(10)}$, the above intersection is a line *l*. Any point of *l* is such that the (00) and (10) corner base point reconstruction conditions are satisfied, and thus can be choosen as \mathbf{b}_{20} .
- If m⁽⁰⁰⁾ and m⁽¹⁰⁾ are parallel, the lifted tangent planes M⁽⁰⁰⁾(z₂₀⁽⁰⁰⁾) and M⁽¹⁰⁾(z₂₀⁽¹⁰⁾) have to coincide, or else, no suitable b₂₀ control point can be chosen. If M = M⁽⁰⁰⁾(z₂₀⁽⁰⁰⁾) = M⁽¹⁰⁾(z₂₀⁽¹⁰⁾) can be arranged, any point of M satisfies the second order reconstruction constraints.

Let us examine the conditions necessary to guarantee the $\mathbf{M}^{(00)}(\overline{z}_{20}^{(00)}) = \mathbf{M}^{(10)}(\overline{z}_{20}^{(10)})$ equality!

For the sake of simplicity, let $\mathbf{M}^{(ij)}$ denote $\mathbf{M}^{(ij)}(\overline{z}_{20}^{(ij)}), i, j = 0, 1$, and $\mathbf{p}_{\mathbf{M}}^{(ij)} \in \mathbf{M}^{(ij)}$ the closest point of $\mathbf{M}^{(ij)}$ to $\mathbf{p}^{(ij)}$, that is

$$\mathbf{p}_{\mathbf{M}}^{(00)} = \mathbf{p}^{(00)} + \overline{z}_{20}^{(00)} \mathbf{m}^{(00)}$$
(3.38)

$$\mathbf{p}_{\mathbf{M}}^{(10)} = \mathbf{p}^{(10)} + \overline{z}_{20}^{(10)} \mathbf{m}^{(10)}$$
(3.39)

Equality $\mathbf{M}^{(00)} = \mathbf{M}^{(10)}$ holds if and only if

$$(\mathbf{p}_{\mathbf{M}}^{(00)} - \mathbf{p}_{\mathbf{M}}^{(10)}) \cdot \mathbf{m} = 0, \qquad (3.40)$$

where $\mathbf{m} = \mathbf{m}^{(00)} = \mathbf{m}^{(10)}$. By definition (3.30) of $\overline{z}_{20}^{(ij)}$

$$0 = (\mathbf{p}_{\mathbf{M}}^{(00)} - \mathbf{p}_{\mathbf{M}}^{(10)}) \cdot \mathbf{m}$$

= $(\mathbf{p}^{(00)} + \overline{z}_{20}^{(00)} \mathbf{m}^{(00)} - (\mathbf{p}^{(10)} + \overline{z}_{20}^{(10)} \mathbf{m}^{(10)})) \cdot \mathbf{m}$
= $((\mathbf{p}^{(00)} - \mathbf{p}^{(10)}) + (\overline{z}_{20}^{(00)} \mathbf{m}^{(00)} - \overline{z}_{20}^{(10)} \mathbf{m}^{(10)})) \cdot \mathbf{m}$
= $(\mathbf{p}^{(00)} - \mathbf{p}^{(10)}) \cdot \mathbf{m}$
+ $\frac{1}{12} \left(\kappa_{1}^{(00)} (x_{u}^{(00)})^{2} + \kappa_{2}^{(00)} (y_{u}^{(00)})^{2} \right) \mathbf{m}^{(00)} \cdot \mathbf{m}$
- $\frac{1}{12} \left(\kappa_{1}^{(10)} (x_{u}^{(10)})^{2} + \kappa_{2}^{(10)} (y_{u}^{(10)})^{2} \right) \mathbf{m}^{(10)} \cdot \mathbf{m}$

that is, (3.40) is a quadratic equation with four unknowns $(x_u^{(00)}, y_u^{(00)}), (x_u^{(10)}, y_u^{(10)}), (x_u^{(10$ the coordinates of the $\mathbf{b}_u(0,0)$, $\mathbf{b}_u(1,0)$ partial derivatives in the bases $(\mathbf{t}_1^{(00)}, \mathbf{t}_2^{(00)})$, $(\mathbf{t}_1^{(10)}, \mathbf{t}_2^{(10)}).$ Let $(l_u^{(00)}, \gamma_u^{(00)}), (l_u^{(10)}, \gamma_u^{(10)})$ be such that

$$l_u^{(00)}(\cos\gamma_u^{(00)}, \sin\gamma_u^{(00)}) = (x_u^{(00)}, y_u^{(00)})$$
$$l_u^{(10)}(\cos\gamma_u^{(10)}, \sin\gamma_u^{(10)}) = (x_u^{(10)}, y_u^{(10)}).$$

Using these polar unknowns, (3.40) can be written as

$$0 = (\mathbf{p}^{(00)} - \mathbf{p}^{(10)}) \cdot \mathbf{m} \rangle + \frac{1}{12} l_u^{(00)^2} (\kappa_1^{(00)} \cos^2 \gamma_u^{(00)} + \kappa_2^{(00)} \sin^2 \gamma_u^{(00)}) \mathbf{m}^{(00)} \cdot \mathbf{m} - \frac{1}{12} l_u^{(10)^2} (\kappa_1^{(10)} \cos^2 \gamma_u^{(10)} + \kappa_2^{(10)} \sin^2 \gamma_u^{(10)}) \mathbf{m}^{(10)} \cdot \mathbf{m}$$

Let us introduce the coefficients and variables

$$a = \frac{1}{12} (\kappa_1^{(00)} \cos^2 \gamma_u^{(00)} + \kappa_2^{(00)} \sin^2 \gamma_u^{(00)}) \mathbf{m}^{(00)} \cdot \mathbf{m}$$

$$b = \frac{1}{12} (\kappa_1^{(10)} \cos^2 \gamma_u^{(10)} + \kappa_2^{(10)} \sin^2 \gamma_u^{(10)}) \mathbf{m}^{(10)} \cdot \mathbf{m}$$

$$c = (\mathbf{p}^{(00)} - \mathbf{p}^{(10)}) \cdot \mathbf{m}$$

$$x = l_u^{(00)}$$

$$y = l_u^{(10)}$$

so that (3.40) can be written as

$$ax^2 - by^2 + c = 0. ag{3.41}$$

Let the angles $\gamma_u^{(00)}$, $\gamma_u^{(10)}$ be fixed. Now (3.41) can be interpreted as a quadratic curve.

Any point of the curve (3.41), that satisfies x > 0, y > 0, corresponds to a partial derivative length pair $l_u^{(00)}$, $l_u^{(10)}$, such that $(l_u^{(00)}, \gamma_u^{(00)})$, $(l_u^{(10)}, \gamma_u^{(10)})$ represent $\mathbf{b}_u(0,0)$, $\mathbf{b}_u(1,0)$ partial derivatives, whose $\mathbf{M}^{(00)}$, $\mathbf{M}^{(10)}$ lifted tangent planes coincide. These points of the quadratic curves solve the second order geometric Hermite interpolation problem along their boundary curve in the case of parallel normals.

Table 3.1 details the different types of quadratic curves the above a, b, c coefficients can determine. Curves in bold denote a, b, c coefficient configurations where no solution can be given. Red curves denote coefficient configurations where no real solution can be given, while the curves in blue correspond to curves that do not have any points in the first quadrant x > 0, y > 0.

The coefficient c depends on the input data only, its value and sign cannot be changed. The sign of coefficients a and b, on the other hand, depend on the choice of $\gamma_u^{(00)}, \gamma_u^{(10)}$, and the principal curvature signs. If $\kappa_1^{(00)} < 0$ and $\kappa_2^{(00)} > 0$, the sign of a can be chosen to be negative, positive, or zero by selecting an appropriate $\gamma_u^{(00)}$ angle. This can be utilized to change a specific (3.41) curve providing no solutions, into a real one, having points inside the first quadrant, if the curvature configurations allow it.

We can use this remark to characterize the existence of bi-quartic Bézier solutions for the four corner base point interpolation problem. In order to do this, let $H^{(ij)}$ denote the half-space containing all the possible, \mathbf{b}_u partial derivative dependent, lifted tangent planes, that is let

$$\begin{aligned} H^{(ij)} &= \{ \mathbf{p}^{(ij)} + \alpha \mathbf{t}_1^{(ij)} + \beta \mathbf{t}_2^{(ij)} + \gamma \mathbf{m}^{(ij)} | \\ \alpha, \beta \in \mathbb{R}, \gamma = sx, x > 0, s \in [sgn(\kappa_1^{(ij)}), sgn(\kappa_2^{(ij)})] \}, \end{aligned}$$

where

$$sgn(x) = \begin{cases} 1, & x > 0\\ 0, & x = 0\\ -1, & x < 0 \end{cases}$$

The derivation above proves the following

Theorem 14 The four corner second order geometric Hermite interpolation problem can be solved using quadrilateral bi-quartic integral Bézier surfaces if and only if

$$H^{(ij)} \cap H^{(kl)} \neq \emptyset \tag{3.42}$$

holds for all neighbouring $\mathbf{D}^{(ij)}$, $\mathbf{D}^{(kl)}$ base point data.

Proof. Let (a) and (b) denote arbitrary neighbouring base point indices. If $\mathbf{m}^{(a)}$ is not parallel to $\mathbf{m}^{(b)}$, then the intersection (3.42) is not empty, since it contains the intersection line l, on which any point solves the interpolation problem along the corresponding boundary curve.

a	b	с	equation of the quadratic	type of the quadratic
+	-	+	$q^2x^2 + r^2y^2 + t^2 = 0$	ellipse (imaginary)
+	-	-	$q^2x^2 + r^2y^2 - t^2 = 0$	ellipse (real)
+	-	0	$q^2x^2 + r^2y^2 = 0$	intersecting lines (imaginary)
+	+	+	$q^2x^2 - r^2y^2 + t^2 = 0$	hyperbola
+	+	-	$q^2x^2 - r^2y^2 - t^2 = 0$	hyperbola
+	+	0	$q^2 x^2 - r^2 y^2 = 0$	intersecting lines (real)
+	0	+	$q^2x^2 + t^2 = 0$	parallel lines (imaginary)
+	0	-	$q^2x^2 - t^2 = 0$	parallel lines (real) $(x = \pm \frac{t}{q})$
+	0	0	$q^2 x^2 = 0$	coincident line $x = 0$
-	-	+	$-q^2x^2 + r^2y^2 + t^2 = 0$	hyperbola
-	-	-	$-q^2x^2 + r^2y^2 - t^2 = 0$	hyperbola
-	-	0	$-q^2x^2 + r^2y^2 = 0$	intersecting lines (real)
-	+	+	$-q^2x^2 - r^2y^2 + t^2 = 0$	ellipse (real)
-	+	-	$-q^2x^2 - r^2y^2 - t^2 = 0$	ellipse (imaginary)
-	+	0	$-q^2x^2 - r^2y^2 = 0$	intersecting lines (imaginary)
-	0	+	$-q^2x^2 + t^2 = 0$	parallel lines (real) $(x = \pm \frac{t}{q})$
-	0	-	$-q^2x^2 - t^2 = 0$	parallel lines (imaginary)
-	0	0	$-q^2x^2 = 0$	coincident line $x = 0$
0	-	+	$r^2y^2 + t^2 = 0$	parallel lines (imaginary)
0	-	-	$r^2y^2 - t^2 = 0$	parallel lines (real) $(y = \pm \frac{t}{r})$
0	-	0	$r^2y^2 = 0$	coincident line $y = 0$
0	+	+	$-r^2y^2 + t^2 = 0$	parallel lines (real) $(y = \pm \frac{t}{r})$
0	+	-	$-r^2y^2 - t^2 = 0$	parallel lines (imaginary)
0	+	0	$-r^2y^2 = 0$	coincident line $y = 0$
0	0	+	$t^2 = 0$	no solution
0	0	-	$-t^2 = 0$	no solution
0	0	0	0 = 0	the entire plane

Table 3.1: Table of quadratics in coordinates $l_u^{(00)}, l_u^{(10)}$. The first three columns are the signs of the $a, b, c \in \{-, 0, +\}$ coefficients. The fourth column is the equation of the quadratic, determined by the signs of the a, b, c coefficients, q, r, t > 0. The fifth column is the type of the curve.



Figure 3.5: The control points required for base point data reconstruction in the case of a 3x3-es Bézier surface. The red, green, blue, and azure areas are the control points required for $\mathbf{D}^{(00)}, \mathbf{D}^{(10)}, \mathbf{D}^{(01)}, \mathbf{D}^{(11)}$.

If $\mathbf{m}^{(a)}||\mathbf{m}^{(b)}$, the lifted tangent planes $\mathbf{M}^{(a)}$ and $\mathbf{M}^{(b)}$ can be moved such that they coincide. All these planes lie in (3.42).

3.2.3.4 Bi-cubic Bézier patch

The control net of a bi-cubic Bézier patch has double overlaps of G^2 reconstruction control point regions along the boundary curves, as shown in figure 3.5.

Let us consider the case of $\mathbf{D}^{(00)}$ and $\mathbf{D}^{(10)}$, and the v = 0 boundary curve.

Using (3.21)-(3.26), the control points are written as

$$\begin{aligned} \mathbf{b}_{00} &= \mathbf{p}^{(00)} \\ \mathbf{b}_{10} &= \mathbf{p}^{(00)} + \frac{x_u^{(00)}}{3} \mathbf{t}_1^{(00)} + \frac{y_u^{(00)}}{3} \mathbf{t}_2^{(00)} \\ &= \mathbf{p}^{(10)} + \frac{x_{uu}^{(10)} + 2x_u^{(10)}}{6} \mathbf{t}_1^{(10)} + \frac{y_{uu}^{(10)} + 2y_u^{(10)}}{6} \mathbf{t}_2^{(10)} + \frac{z_{uu}^{(10)}}{6} \mathbf{m}^{(10)} \\ \mathbf{b}_{20} &= \mathbf{p}^{(10)} + \frac{x_u^{(10)}}{3} \mathbf{t}_1^{(10)} + \frac{y_u^{(10)}}{3} \mathbf{t}_2^{(10)} \\ &= \mathbf{p}^{(00)} + \frac{x_{uu}^{(00)} + 2x_u^{(00)}}{6} \mathbf{t}_1^{(00)} + \frac{y_{uu}^{(00)} + 2y_u^{(00)}}{6} \mathbf{t}_2^{(00)} + \frac{z_{uu}^{(00)}}{6} \mathbf{m}^{(00)} \\ \mathbf{b}_{30} &= \mathbf{p}^{(10)} \end{aligned}$$

Each inner control point must lie on the intersection of a tangent plane and a lifted tangent plane:

$$\mathbf{b}_{10} \in \mathbf{T}^{(00)} \cap \mathbf{M}^{(10)} \tag{3.43}$$

$$\mathbf{b}_{20} \in \mathbf{T}^{(10)} \cap \mathbf{M}^{(00)} \tag{3.44}$$

where $\mathbf{M}^{(ij)}$ denotes $\mathbf{M}^{(ij)}(m_{20}^{(ij)})$ again, and $\mathbf{T}^{(ij)} = \mathbf{T}_{\mathbf{D}^{(ij)}}, i, j = 0, 1.$

Let us examine when (3.43) and (3.44) can be satisfied. To do that, we have to separate the case of non parallel and parallel normals:

• If $\mathbf{m}^{(00)}$ is not parallel to $\mathbf{m}^{(10)}$, then we have two intersection lines

$$l_0 = \mathbf{T}^{(00)} \cap \mathbf{M}^{(10)}$$
$$l_1 = \mathbf{T}^{(10)} \cap \mathbf{M}^{(00)}$$

if $H^{(00)} \cap \mathbf{T}^{(10)} \neq \emptyset$ and $H^{(10)} \cap \mathbf{T}^{(00)} \neq \emptyset$.

Let us choose a $\mathbf{q} \in \mathbf{T}^{(00)}$ point as the \mathbf{b}_{10} control point. This determines the l_1 intersection line, along which control point \mathbf{b}_{20} may be placed.

The set of control points $\mathbf{x} = \mathbf{p}^{(10)} + x \mathbf{t}_1^{(10)} + y \mathbf{t}_2^{(10)} \in \mathbf{T}^{(10)}$, that create

intersection lines l_0 , such that $\mathbf{b}_1 \in l_0$, are subject to

$$0 = (\mathbf{q} - \mathbf{p}_{M}^{(10)}) \cdot \mathbf{m}^{(10)}$$

= $(\mathbf{q} - \mathbf{p}^{(10)}) \cdot \mathbf{m}^{(10)} - \overline{z}_{20}^{(10)}$
= $(\mathbf{q} - \mathbf{p}^{(10)}) \cdot \mathbf{m}^{(10)} - (\kappa_{1}^{(10)}x^{2} + \kappa_{2}^{(10)}y^{2})$

that is, by setting

$$a = \kappa_1^{(10)}$$
$$b = \kappa_2^{(10)}$$
$$c = (\mathbf{q} - \mathbf{p}^{(10)}) \cdot \mathbf{m}^{(10)}$$

they are the points of the quadratic curve

$$ax^2 + by^2 - c = 0 (3.45)$$

except the origin. Note that not every $\mathbf{q} \in \mathbf{T}^{(00)}$ creates a real quadratic on $\mathbf{T}^{(10)}$.

To solve this, let \mathbf{d}_0 be a unit vector in the $\mathbf{T}^{(00)}$ tangent plane, and let $\mathbf{q}(t) = \mathbf{p}^{(00)} + t\mathbf{d}_0$. The (3.45) quadratic's coefficients, by substituting \mathbf{q} with $\mathbf{q}(t)$, take the form

$$a = \kappa_1^{(10)}$$

$$b = \kappa_2^{(10)}$$

$$c = t\mathbf{d}_0 \cdot \mathbf{m}^{(10)} + (\mathbf{p}^{(00)} - \mathbf{p}^{(10)}) \cdot \mathbf{m}^{(10)}$$

Because $\mathbf{m}^{(00)} \not\parallel \mathbf{m}^{(10)}$, the dot product $\langle \mathbf{d}_0, \mathbf{m}^{(10)} \rangle$ cannot be zero for all directions \mathbf{d}_0 , and it can take any sign, $\{+, -, 0\}$. Using parameter t, we can always guarantee, that $ax^2 + by^2 - c = 0$ is a real quadratic (an ellipse, hyperbola, or an intersecting or parallel pair of lines), provided $H^{(10)} \cap \mathbf{T}^{(00)} \neq \emptyset$.

Placing \mathbf{b}_{20} into either of the intersection points of this quadratic and l_1 satisfies the second order base point reconstruction constraints.

• $\mathbf{m}^{(00)}||\mathbf{m}^{(10)}$: if the surface normals are parallel, the equalities

$$\mathbf{T}^{(00)} = \mathbf{M}^{(10)}$$

 $\mathbf{T}^{(10)} = \mathbf{M}^{(00)}$

should be satisfied by choosing appropriate $\mathbf{b}_u(0,0)$ and $\mathbf{b}_u(1,0)$ partial derivatives.

These can be formulated as

$$(\mathbf{p}_{\mathbf{M}}^{(00)} - \mathbf{p}^{(10)}) \cdot \mathbf{m}^{(10)} = 0$$
$$(\mathbf{p}_{\mathbf{M}}^{(10)} - \mathbf{p}^{(00)}) \cdot \mathbf{m}^{(00)} = 0$$

Using definition (3.30), this can be written as

$$\frac{1}{6} \left(\kappa_1^{(00)} x_u^{(00)^2} + \kappa_2^{(00)} y_u^{(00)^2} \right) \langle \mathbf{m}^{(00)}, \mathbf{m}^{(10)} \rangle + \langle \mathbf{p}^{(00)} - \mathbf{p}^{(10)}, \mathbf{m}^{(10)} \rangle = 0 , \quad (3.46)$$

$$\frac{1}{6} \left(\kappa_1^{(10)} x_u^{(10)^2} + \kappa_2^{(10)} y_u^{(10)^2} \right) \langle \mathbf{m}^{(10)}, \mathbf{m}^{(00)} \rangle + \langle \mathbf{p}^{(10)} - \mathbf{p}^{(00)}, \mathbf{m}^{(00)} \rangle = 0 . \quad (3.47)$$

We examine the solvability of (3.46)-(3.47) by considering them as quadratic curves, in this instance in the $\mathbf{T}^{(00)}$ and $\mathbf{T}^{(10)}$ tangent planes themselves. For this, let

$$a = \frac{1}{6} \kappa_1^{(00)} \mathbf{m}^{(00)} \cdot \mathbf{m}^{(10)}$$
$$b = \frac{1}{6} \kappa_2^{(00)} \mathbf{m}^{(00)} \cdot \mathbf{m}^{(10)}$$
$$c = (\mathbf{p}^{(00)} - \mathbf{p}^{(10)}) \cdot \mathbf{m}^{(10)}$$

so that we can formulate the problem of finding \mathbf{b}_{20} , as finding a real point of the quadratic curve

$$a(x_u^{(00)})^2 + b(y_u^{(00)})^2 + c = 0$$

except the origin. The conditions for the existence of such a point is similar to the discussion in the previous subsection, with the difference that we are looking for solutions in all four quadrants. The case of \mathbf{b}_{10} is analogous.

Let us consider the use of rational cubic patches now. The boundary con-

trol point computation problem can be solved by using osculatory interpolation ([11]), which is concerned with finding weights w_0, w_1, w_2, w_3 for a given Bézier control polygon $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$, such that the rational cubic reconstructs prescribed κ_0, κ_1 curvature values at each end point.

The positions of the $\mathbf{D}^{(ij)}$ base point data are chosen as \mathbf{b}_0 and \mathbf{b}_3 , while the $\mathbf{b}_1, \mathbf{b}_2$ control points can be chosen on the tangent planes freely. The Meusnier theorem can be used to determine the end point curvature data by computing the curvature of skew sections and using the osculatory interpolants as boundary curves. The rest of the control points can be chosen as before.

The following proposition is the consequence of the above derivations:

Theorem 15 The four corner second order geometric Hermite interpolation problem can be solved using quadrilateral bi-cubic Bézier surfaces **if and only if** for each neighbouring $\mathbf{D}^{(ij)}$, $\mathbf{D}^{(kl)}$ base point data

$$H^{(ij)} \cap \mathbf{T}^{(kl)} \neq \emptyset$$
$$H^{(kl)} \cap \mathbf{T}^{(ij)} \neq \emptyset$$

3.2.4 Triangular Bézier patches

For the purpose of this thesis, investigation of GH interpolation with quadrilateral patches provides sufficient insight into higher order GH interpolation: the way reconstruction constraints are to be placed upon partial derivatives of the surface, finding polynomial degree bounds, and how control data overlap increases the complexity of even the geometric characterization of existence.

However, in many applications the need for non-quadrilateral patches arises naturally as well. For the sake of completeness, Appendix C details the application of Theorem 12 to triangular Bézier patches in the same vein as presented for quadrilateral patches.

3.3 Geometric Hermite surface interpolation

In the previous section, a generalization of second order GH interpolation to surfaces was presented. It utilized parametrization independent surface data up to second order to specify the desired geometric properties of the interpolant surface. Similarly to the case of curves, Theorem 10 and 12 posed restrictions upon the partial derivatives of the surface as requirements of GH reconstruction. Unfortunately, these results did not present a "most straightforward" set of attributes to include in a higher order GH data tuples: their derivation relied on Euler's theorem to compute the normal curvature of a normal section curve for a given tangent.

In this section, the properties I propose to prescribe in higher order GH interpolation are based on the differential geometric invariants of lines of curvature, which can be generalized to arbitrary order. The attributes presented for this are chosen such that reconstruction of order n GH surface data tuples ensures G^n geometric continuity of joining surfaces at that point.

The following point reviews the classical results retaining lines of curvatures. It is followed by a discussion of derivative coordinates in a special orthonormal basis, which is an analogue to the examination of curve derivatives in the Frenet frame. The impact of this formulation is investigated next.

The statement of higher order geometric Hermite inteporlation of surfaces closes this section.

3.3.1 Lines of curvature

Lines of curvature are curves on a regular parametric surface $\mathbf{r}(u, v) : \mathbb{R}^2 \to \mathbb{E}^3$, such that at every point, their tangent coincides with one of the principal directions. Throughout this chapter, only regular parametrizations are considered.

A line of curvature can be defined via a $(u(x), v(x)) : \mathbb{R} \to \mathbb{R}^2$ curve in the parameter plane of the surface, such that

$$(L - \kappa_n E)u' + (M - \kappa_n F)v' = 0$$
(3.48)

$$(M - \kappa_n F)u' + (N - \kappa_n G)v' = 0$$
(3.49)

holds, where κ_n denotes one of the principal curvatures, $n \in \{1, 2\}$ and differentiation is carried out with respect to x. The lines of curvature on the surface are then given by

$$\mathbf{c}_i(x) = \mathbf{r}\big(u(x), v(x)\big) \ i = 1, 2$$

Let $dom(\mathbf{r}(u, v))$ denote the domain of $\mathbf{r}(u, v)$, that is, the (u, v) parameter plane. The lines of curvature in $dom(\mathbf{r}(u, v))$ can be also defined as the solution

$$\begin{array}{c|cccc} (u')^2 & -u'v' & (v')^2 \\ E & F & G \\ L & M & N \end{array} = 0$$
 (3.50)

Lines of curvature form an orthogonal family of curves on a surface. The curves of this family cover the surface simply, without gaps in the neighborhood of any point where the first and second fundamental forms are continuous and non-proportional [45].

At umbilics, where the fundamental forms are proportional, all normal curvatures coincide. The examination of the properties of the surface require higher order partial derivatives in these cases. Situations may arise where there are either one or three principal directions, as well as points where there are infinitely many principal directions - for a detailed overview, please refer to [32]. The definition for principal directions, proposed by Rodrigues, gives a well defined principal direction, even at umbilics:

$$\kappa_n \mathbf{dr} + \mathbf{dm} = 0 \; ,$$

where \mathbf{dr} in an infinitesimal displacement along the surface and \mathbf{dm} is the change of surface normal \mathbf{m} as a result of this displacement [32].

Cases where parameter lines are lines of curvatures are of distinct importance in what follows. The necessary and sufficient condition for this to hold [9] is

$$F = 0$$
, $M = 0$. (3.51)

In parametrizations such as this, the curvatures and torsions of the image of parameter lines u = c, v = d $(c, d \in \mathbb{R})$ are that of the lines of curvature.

3.3.2 Derivatives in the Darboux-frame

The unit surface normal can be expressed with the principal directions by $\mathbf{m} = \mathbf{t}_1 \times \mathbf{t}_2$ that is, $D = (\mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$ form an orthonormal system, called the *Darboux frame*. Without loss of generality, we can assume that D is a right-handed basis of \mathbb{R}^3 .

From now on, let $\hat{\mathbf{r}}(s,t)$ denote a parametrization of a surface such that the parameter lines are arc-length parametrized lines of curvature. This can

to

be considered as the analogue of arc-length parametrization of curves. We refer to this parametrization of surfaces as *natural parametrization*. A natural parametrization is also regular, except at umbilics where there is only one principal direction.

The first partial derivatives of the natural parametrization are the principal directions, that is, their Darboux coordinates are

$$\hat{\mathbf{r}}_s = \mathbf{t}_1 = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_D$$
, $\hat{\mathbf{r}}_t = \mathbf{t}_2 = \begin{bmatrix} 0\\1\\0 \end{bmatrix}_D$,

and for the surface normal

$$\mathbf{m} = \hat{\mathbf{r}}_s \times \hat{\mathbf{r}}_t = \begin{bmatrix} 0\\0\\1 \end{bmatrix}_D . \tag{3.52}$$

Every reparamentrization of $\hat{\mathbf{r}}(s,t)$ can be considered as to have the form of

$$\mathbf{r}(u,v) = \hat{\mathbf{r}}(s(u,v),t(u,v))$$

where $s, t : \mathbb{R}^2 \to \mathbb{R}$ encode the deviation from unit speed parametrization, as well as the change of first partial derivative directions from principal directions.

Thus, the partial derivatives of an arbitrary parametrization can be written as a combination of partial derivatives of natural parametrization as

$$\mathbf{r}_{u} = \hat{\mathbf{r}}_{s}s_{u} + \hat{\mathbf{r}}_{t}t_{u}$$

$$\mathbf{r}_{v} = \hat{\mathbf{r}}_{s}s_{v} + \hat{\mathbf{r}}_{t}t_{v}$$

$$\mathbf{r}_{uu} = \hat{\mathbf{r}}_{ss}s_{u}^{2} + 2\hat{\mathbf{r}}_{st}s_{u}t_{u} + \hat{\mathbf{r}}_{tt}t_{u}^{2} + \hat{\mathbf{r}}_{s}s_{uu} + \hat{\mathbf{r}}_{t}t_{uu}$$

$$\mathbf{r}_{uv} = \hat{\mathbf{r}}_{ss}s_{u}s_{v} + \hat{\mathbf{r}}_{st}s_{u}t_{v} + \hat{\mathbf{r}}_{st}s_{v}t_{u} + \hat{\mathbf{r}}_{tt}t_{u}t_{v} + \hat{\mathbf{r}}_{s}s_{uv} + \hat{\mathbf{r}}_{t}t_{uv}$$

$$\mathbf{r}_{vv} = \hat{\mathbf{r}}_{ss}s_{v}^{2} + 2\hat{\mathbf{r}}_{st}s_{v}t_{v} + \hat{\mathbf{r}}_{tt}t_{v}^{2} + \hat{\mathbf{r}}_{s}s_{vv} + \hat{\mathbf{r}}_{t}t_{vv}$$

Darboux coordinates of successive partial derivatives in one direction, that is $\mathbf{r}_s, \mathbf{r}_{ss}, \mathbf{r}_{sss}$, etc., are defined by prescribing the differential geometric invariants of the lines of curvature and the correspondence between the Frenet frame of the given line of curvature and the Darboux frame. We will discuss this in more detail, but prior to that, let us consider the problem of mixed partial derivatives. These are not defined by the data tuples associated with the lines of curvatures, however, the following lemma holds:

Lemma 16 Let $\hat{\mathbf{r}}(s,t)$ be a natural parametrization of a surface. Then

$$\partial_{s^k t^l} \hat{\mathbf{r}} = \mathbf{0}$$

for all k, l > 0.

Proof. We prove the above by induction of total order of differentiation, which is denoted by n = k + l.

For n = 2: since parameter lines are arc-length, they are subject to

$$\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s = 1 , \qquad (3.53)$$

$$\hat{\mathbf{r}}_t \cdot \hat{\mathbf{r}}_t = 1 \ . \tag{3.54}$$

Differentiating these with respect to t and s, in this order, yields

$$2\hat{\mathbf{r}}_{st}\cdot\hat{\mathbf{r}}_s=0 ,$$

$$2\hat{\mathbf{r}}_{st}\cdot\hat{\mathbf{r}}_t=0 ,$$

taking into account that $\hat{\mathbf{r}}_{st} = \hat{\mathbf{r}}_{ts}$. In addition, parameter lines are also lines of curvature, that is, they satisfy

$$M = \hat{\mathbf{r}}_{st} \cdot \mathbf{m} = 0 \ . \tag{3.55}$$

As a result, $\hat{\mathbf{r}}_{st}$ is such that it is perpendicular to $\hat{\mathbf{r}}_s = \mathbf{t}_1$, $\hat{\mathbf{r}}_t = \mathbf{t}_2$, and \mathbf{m} . These three vectors form an orthonormal basis of \mathbb{R}^3 , and as such, only the null vector can be simultaneously orthogonal to all three, that is, $\hat{\mathbf{r}}_{st} = \mathbf{0}$.

Let us suppose that the induction condition holds for some total order n, and let us validate it for a total order of n + 1.

Let n = k + l, k, l > 0. From induction, differentiating (3.53)-(3.54) k and l times by s and t gives

$$\begin{aligned} (\partial_{s^k t^l} \hat{\mathbf{r}}) \cdot \hat{\mathbf{r}}_s &= 0 , \\ (\partial_{s^k t^l} \hat{\mathbf{r}}) \cdot \hat{\mathbf{r}}_t &= 0 . \end{aligned}$$

Differentiating each of the above two equations with respect to both s and t, one gets

$$\partial_s \left((\partial_{s^k t^l} \hat{\mathbf{r}}) \cdot \hat{\mathbf{r}}_s \right) = 0$$
$$\partial_t \left((\partial_{s^k t^l} \hat{\mathbf{r}}) \cdot \hat{\mathbf{r}}_s \right) = 0$$
$$\partial_s \left((\partial_{s^k t^l} \hat{\mathbf{r}}) \cdot \hat{\mathbf{r}}_t \right) = 0$$
$$\partial_t \left((\partial_{s^k t^l} \hat{\mathbf{r}}) \cdot \hat{\mathbf{r}}_t \right) = 0$$

which is evaluated as

$$\begin{aligned} (\partial_{s^{k+1}t^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{s}+(\partial_{s^{k}v^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{ss}&=0\\ (\partial_{s^{k}t^{l+1}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{s}+(\partial_{s^{k}v^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{st}&=0\\ (\partial_{s^{k+1}t^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{t}+(\partial_{s^{k}v^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{ss}&=0\\ (\partial_{s^{k}t^{l+1}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{t}+(\partial_{s^{k}v^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{tt}&=0\end{aligned}$$

that can be written as

$$(\partial_{s^{k+1}t^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{s}=0$$
$$(\partial_{s^{k}t^{l+1}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{s}=0$$
$$(\partial_{s^{k+1}t^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{t}=0$$
$$(\partial_{s^{k}t^{l+1}}\hat{\mathbf{r}})\cdot\hat{\mathbf{r}}_{t}=0$$

using the induction condition $\partial_{s^k v^l} \hat{\mathbf{r}} = \mathbf{0}$. That is, $\partial_{s^{k+1} v^l} \hat{\mathbf{r}}$ and $\partial_{s^{k+1} v^l} \hat{\mathbf{r}}$ are perpendicular to $\hat{\mathbf{r}}_s = \mathbf{t}_1$ and $\hat{\mathbf{r}}_t = \mathbf{t}_2$. Differentiating (3.55) similarly results in

$$\partial_s \left((\partial_{s^k v^l} \hat{\mathbf{r}}) \cdot \hat{\mathbf{m}} \right) = 0$$
$$\partial_t \left((\partial_{s^k v^l} \hat{\mathbf{r}}) \cdot \hat{\mathbf{m}} \right) = 0$$

which, after substitutions becomes

$$(\partial_{s^{k+1}t^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{m}} + (\partial_{s^{k}v^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{m}}_{s} = 0$$
$$(\partial_{s^{k}t^{l+1}}\hat{\mathbf{r}})\cdot\hat{\mathbf{m}} + (\partial_{s^{k}v^{l}}\hat{\mathbf{r}})\cdot\hat{\mathbf{m}}_{t} = 0$$

that is, using induction, both $\partial_{s^{k+1}v^l}\hat{\mathbf{r}}$ and $\partial_{s^kv^{l+1}}\hat{\mathbf{r}}$ are perpendicular to \mathbf{m} as well, that is,

$$\partial_{s^{k+1}v^l} \hat{\mathbf{r}} = \partial_{s^k v^{l+1}} \hat{\mathbf{r}} = \mathbf{0}$$
.

Note that arc-length – more precisely, constant speed – parametrization played a pivotal part in the proof. The above lemma does not extend to surfaces that are parametrized by lines of curvature that are not constant speed. Nevertheless, the lemma holds under weaker assumptions as well: parameter lines need only be conjugate and arc-length, the proof above works analogously.

Example Let us consider the cylinder of radius a > 0 and height h > 0:

$$\mathbf{r}(u,v) = \begin{bmatrix} a \cos u \\ a \sin u \\ vh \end{bmatrix}, u \in [0,2\pi], v \in [0,1]$$

This is known to be a parametrization by lines of curvature, and also, its first partial derivatives are

$$\mathbf{r}_{u} = \begin{bmatrix} -a\sin u\\ a\cos u\\ 0 \end{bmatrix} , \ \mathbf{r}_{v} = \begin{bmatrix} 0\\ 0\\ h \end{bmatrix}$$

that is, this is a parametrization by constant speed. The uv mixed partial derivative, in accordance with Lemma 16, is

$$\mathbf{r}_{uv}=\mathbf{0}$$
 .

On the other hand, the parametrization of a torus of the form

$$\mathbf{p}(u,v) = \begin{bmatrix} (R+r\cos v)\cos u\\ (R+r\cos v)\sin u\\ r\sin v \end{bmatrix}, \ u,v \in [0,2\pi]$$



Figure 3.6: Curve normals $\mathbf{n}_1, \mathbf{n}_2$ of line of curvatures corresponding to the first and second principal direction in the right-handed $(\mathbf{m}, \mathbf{t}_2)$ and $(\mathbf{m}, -\mathbf{t}_1)$ bases, respectively.

with radii R, r > 0, is also parametrized by lines of curvatures. However, its first partial derivatives are

$$\mathbf{p}_{u} = \begin{bmatrix} -(R+r\cos v)\sin u\\ -(R+r\cos v)\cos u\\ 0 \end{bmatrix}, \ \mathbf{p}_{v} = \begin{bmatrix} -r\cos u\sin v\\ -r\sin u\sin v\\ r\cos v \end{bmatrix}$$

where the surface curve image of u = c is not constant speed. In turn, the mixed partial derivative is

$$\mathbf{p}_{uv} = \begin{bmatrix} r \sin u \sin v \\ -r \cos u \sin v \\ 0 \end{bmatrix} \, ,$$

which does not vanish for all $(u, v) \in [0, 2\pi]^2$.

As a result of Lemma 16, partial derivatives of an arbitrary parametrization $\mathbf{r}(u, v)$ only depend on the derivatives of the lines of curvature. The coordinates of these derivatives need to be given in a common basis. Let this common basis be the Darboux frame.

The lines of curvatures at a point can be defined by GH data tuples $\mathbf{D}_1, \mathbf{D}_2$. Theorem 1 allows us to expresses the derivatives of a line of curvature in its $F_i = (\mathbf{t}_i, \mathbf{n}_i, \mathbf{b}_i)$ Frenet frame.

Let α and β denote the angles between the surface normal **m** and the **n**₁, **n**₂ normal vectors of the two lines of curvatures, as shown in Figure 3.6. Then the transformation from the F_i Frenet frame to the *D* Darboux-frame is a simple rotation specified by

$$F_1 \to D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin \alpha & -\cos \alpha \\ 0 & \cos \alpha & \sin \alpha \end{bmatrix}, \quad F_2 \to D = \begin{bmatrix} 0 & -\sin \beta & \cos \beta \\ 1 & 0 & 0 \\ 0 & \cos \beta & \sin \beta \end{bmatrix}$$
(3.56)

since

$$\mathbf{t}_{1} = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{D}, \ \mathbf{n}_{1} = \begin{bmatrix} 0\\\sin\alpha\\\cos\alpha \end{bmatrix}_{D}, \ \mathbf{b}_{1} = \mathbf{t}_{1} \times \mathbf{n}_{1} = \begin{bmatrix} 0\\-\cos\alpha\\\sin\alpha \end{bmatrix}_{D}$$

and

$$\mathbf{t}_{2} = \begin{bmatrix} 0\\1\\0\\D \end{bmatrix}_{D}, \ \mathbf{n}_{2} = \begin{bmatrix} -\sin\beta\\0\\\cos\beta\\D \end{bmatrix}_{D}, \ \mathbf{b}_{2} = \mathbf{t}_{2} \times \mathbf{n}_{2} = \begin{bmatrix} \cos\beta\\0\\\sin\beta\\D \end{bmatrix}_{D}.$$

Let us use notation

$$\kappa_{1g} = \kappa_1 \cdot \sin \alpha , \ \kappa_{1n} = \kappa_1 \cdot \cos \alpha ,$$

$$\kappa_{2g} = \kappa_2 \cdot \sin \beta , \ \kappa_{2n} = \kappa_2 \cdot \cos \beta ,$$

$$\tau_{1g} = \tau_1 \cdot \sin \alpha , \ \tau_{1n} = \tau_1 \cdot \cos \alpha ,$$

$$\tau_{2g} = \tau_2 \cdot \sin \beta , \ \tau_{2n} = \tau_2 \cdot \cos \beta ,$$

to denote the geodesic and normal curvatures and torsions. This notation is used for the derivatives of the curvatures as well, that is, $\hat{\kappa}'_{1g} = \hat{\kappa}'_1 \cdot \sin \alpha$, $\hat{\kappa}''_{1g} = \hat{\kappa}''_1 \cdot \sin \alpha$.

The rotations above need to be applied to the Frenet frame expression of the derivatives. For example, in the case of the first three non-zero partials, one gets

$$\hat{\mathbf{r}}_{s} = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{F_{1}} = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{D}, \quad \hat{\mathbf{r}}_{ss} = \begin{bmatrix} 0\\\kappa_{1}\\0 \end{bmatrix}_{F_{1}} = \begin{bmatrix} 0\\\kappa_{1g}\\\kappa_{1n} \end{bmatrix}_{D}, \quad (3.57)$$

$$\hat{\mathbf{r}}_{sss} = \begin{bmatrix} -\kappa_{1}^{2}\\\hat{\kappa}'_{1}\\\kappa_{1}\tau_{1} \end{bmatrix}_{F_{1}} = \begin{bmatrix} -\kappa_{1}^{2}\\\hat{\kappa}'_{1g} - \kappa_{1n}\tau_{1}\\\hat{\kappa}'_{1n} + \kappa_{1g}\tau_{1} \end{bmatrix}_{D}, \quad (3.58)$$

and

$$\hat{\mathbf{r}}_{t} = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{F_{2}} = \begin{bmatrix} 0\\1\\0 \end{bmatrix}_{D}, \quad \hat{\mathbf{r}}_{tt} = \begin{bmatrix} 0\\\kappa_{2}\\0 \end{bmatrix}_{F_{2}} = \begin{bmatrix} -\kappa_{2g}\\0\\\kappa_{2n} \end{bmatrix}_{D}, \quad (3.59)$$
$$\hat{\mathbf{r}}_{ttt} = \begin{bmatrix} -\kappa_{2}^{2}\\\hat{\kappa}_{2}'\\\kappa_{2}\tau_{2} \end{bmatrix}_{F_{2}} = \begin{bmatrix} -\hat{\kappa}'_{2g} + \kappa_{2n}\tau_{2}\\-\kappa_{2}^{2}\\\hat{\kappa}'_{2n} + \kappa_{2g}\tau_{2} \end{bmatrix}_{D}. \quad (3.60)$$

The following corollary is simply the application of the Bonnet-Kovalevsky formula [32] to the lines of curvature, which provides us the sought computational tool to compute the Darboux coordinates of the derivatives of lines of curvatures:

Corollary 17 Let $\hat{\mathbf{c}}_1(s)$ and $\hat{\mathbf{c}}_2(t)$ denote the two lines of curvatures on a surface through a common point \mathbf{p} . The Darboux coordinates of the derivative vectors of the two lines of curvatures at \mathbf{p} are subject to the recurrence formulas

$$\hat{\mathbf{c}}_{1}^{(n+1)} := \begin{bmatrix} \hat{x}_{n+1} \\ \hat{y}_{n+1} \\ \hat{z}_{n+1} \end{bmatrix}_{D} = \begin{bmatrix} \hat{x}_{n}' \\ \hat{y}_{n}' \\ \hat{z}_{n}' \end{bmatrix}_{D} + \begin{bmatrix} 0 & -\kappa_{1g} & -\kappa_{1n} \\ \kappa_{1g} & 0 & -\tau_{1} \\ \kappa_{1n} & \tau_{1} & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_{n} \\ \hat{y}_{n} \\ \hat{z}_{n} \end{bmatrix}_{D}$$

$$\hat{\mathbf{c}}_{2}^{(n+1)} := \begin{bmatrix} \tilde{x}_{n+1} \\ \tilde{y}_{n+1} \\ \tilde{z}_{n+1} \end{bmatrix}_{D} = \begin{bmatrix} \tilde{x}_{n}' \\ \tilde{y}_{n}' \\ \tilde{z}_{n}' \end{bmatrix}_{D} + \begin{bmatrix} 0 & -\kappa_{2g} & \tau_{2} \\ \kappa_{2g} & 0 & -\kappa_{2n} \\ -\tau_{2} & \kappa_{2n} & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_{n} \\ \tilde{y}_{n} \\ \tilde{z}_{n} \end{bmatrix}_{D}$$
(3.61)
$$(3.62)$$

where n > 1 and

$$\hat{\mathbf{c}}_1' = \begin{bmatrix} 1\\ 0\\ 0 \end{bmatrix}_D , \ \hat{\mathbf{c}}_2' = \begin{bmatrix} 0\\ 1\\ 0 \end{bmatrix}_D$$

Proof. By using the Frenet to Darboux transformation matrices (3.56), and their inverses

$$D \to F_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin \alpha & \cos \alpha \\ 0 & -\cos \alpha & \sin \alpha \end{bmatrix} , \ D \to F_1 = \begin{bmatrix} 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \\ \cos \beta & 0 & \sin \beta \end{bmatrix} ,$$

given a $[x, y, z]_D^T = x \mathbf{t}_1 + y \mathbf{t}_2 + z \mathbf{m}$ Darboux vector, with all coordinates de-

pending on the same variable, differentiation yields

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{D}^{\prime} = \begin{bmatrix} x^{\prime} \\ y^{\prime} \\ z^{\prime} \end{bmatrix}_{D} + [\mathbf{t}_{1}, \mathbf{t}_{2}, \mathbf{m}]^{\prime} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{D},$$

which, using the linearity of differentiation, equals to

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{D}^{\prime} = \begin{bmatrix} x^{\prime} \\ y^{\prime} \\ z^{\prime} \end{bmatrix}_{D} + (F_{1} \to D) \cdot K_{i} \cdot (D \to F_{1}) \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{D}$$

where K denotes the Frenet frame development of either the $\hat{\mathbf{c}}_1(s)$ or the $\hat{\mathbf{c}}_2(t)$ line of curvature:

$$K_{i} = \begin{bmatrix} 0 & -\kappa_{i} & 0\\ \kappa_{i} & 0 & -\tau_{i}\\ 0 & \tau_{i} & 0 \end{bmatrix} , i = 1, 2 .$$

Carrying out the matrix multiplication for i = 1, 2 yields (3.61)-(3.62).

Theorem 3.61 allows us to compute all partial derivatives of a general parameterization, since those are combinations of derivative vectors of lines of curvatures.

Example Let us express the first couple of partial derivatives of an arbitrary parameterized surface, $\mathbf{r}(u, v) = \hat{\mathbf{r}}(s(u, v), t(u, v))$, in the Darboux frame.

Taking into account that the mixed partial derivatives all vanish, these can

be expressed as a combination of natural parametrization partials as

$$\begin{aligned} \mathbf{r}_{u} &= \hat{\mathbf{r}}_{s}s_{u} + \hat{\mathbf{r}}_{t}t_{u} \\ \mathbf{r}_{v} &= \hat{\mathbf{r}}_{s}s_{v} + \hat{\mathbf{r}}_{t}t_{v} \\ \mathbf{r}_{uu} &= \hat{\mathbf{r}}_{ss}s_{u}^{2} + \hat{\mathbf{r}}_{tt}t_{u}^{2} + \hat{\mathbf{r}}_{s}s_{uu} + \hat{\mathbf{r}}_{t}t_{uu} \\ \mathbf{r}_{uv} &= \hat{\mathbf{r}}_{ss}s_{u}s_{v} + \hat{\mathbf{r}}_{tt}t_{u}t_{v} + \hat{\mathbf{r}}_{s}s_{uv} + \hat{\mathbf{r}}_{t}t_{uv} \\ \mathbf{r}_{uv} &= \hat{\mathbf{r}}_{ss}s_{u}s_{v}^{2} + \hat{\mathbf{r}}_{tt}t_{v}^{2} + \hat{\mathbf{r}}_{s}s_{vv} + \hat{\mathbf{r}}_{t}t_{vv} \\ \mathbf{r}_{vv} &= \hat{\mathbf{r}}_{sss}s_{u}^{3} + \hat{\mathbf{r}}_{ttt}t_{u}^{3} + 3\hat{\mathbf{r}}_{ss}s_{u}s_{uu} + 3\hat{\mathbf{r}}_{tt}t_{u}t_{uu} + \hat{\mathbf{r}}_{s}s_{uuu} + \hat{\mathbf{r}}_{t}t_{uuu} \\ \mathbf{r}_{uuv} &= \hat{\mathbf{r}}_{sss}s_{u}^{2}s_{v} + \hat{\mathbf{r}}_{ttt}t_{u}^{2}t_{v} + 2\hat{\mathbf{r}}_{ss}s_{u}s_{uv} + 2\hat{\mathbf{r}}_{tt}t_{u}t_{uv} \\ &+ \hat{\mathbf{r}}_{ss}s_{v}s_{uu} + \hat{\mathbf{r}}_{tt}t_{v}t_{uu} + \hat{\mathbf{r}}_{s}s_{uvv} + \hat{\mathbf{r}}_{t}t_{uvv} \\ \mathbf{r}_{uvv} &= \hat{\mathbf{r}}_{sss}s_{u}s_{v}^{2} + \hat{\mathbf{r}}_{ttt}t_{u}t_{v}^{2} + 2\hat{\mathbf{r}}_{ss}s_{v}s_{uv} + 2\hat{\mathbf{r}}_{tt}t_{v}t_{uv} \\ &+ \hat{\mathbf{r}}_{ss}s_{u}s_{vv} + \hat{\mathbf{r}}_{tt}t_{u}t_{vv} + \hat{\mathbf{r}}_{s}s_{uvv} + \hat{\mathbf{r}}_{t}t_{v}t_{uv} \\ &+ \hat{\mathbf{r}}_{ss}s_{u}s_{vv} + \hat{\mathbf{r}}_{tt}t_{u}t_{vv} + \hat{\mathbf{r}}_{s}s_{uvv} + \hat{\mathbf{r}}_{t}t_{vvv} + \hat{\mathbf{r}}_{s}s_{uvv} + \hat{\mathbf{r}}_{t}t_{v}t_{vv} + \hat{\mathbf{r}}_{s}s_{vvv} + \hat{\mathbf{r}}_{t}t_{v}t_{vv} + \hat{\mathbf{r}}_{s}s_{vvv} + \hat{\mathbf{r}}_{t}t_{v}t_{vv} + \hat{\mathbf{r}}_{s}s_{vvv} + \hat{\mathbf{r}}_{t}t_{vvvv} + \hat{\mathbf{r}}_{s}s_{vvv} + \hat{\mathbf{r}}_{t}t_{vvvv} + \hat{\mathbf{r}}_{s}s_{vvv} + \hat{\mathbf{r}}_{t}t_{vvvv} + \hat{\mathbf{r}}_{v}s_{vvv} + \hat{\mathbf{r}}_{v}s_{vvv} + \hat{\mathbf{r}}_{v}t_{vvv} + \hat{\mathbf{r}}_{v}s_{vvv} + \hat{\mathbf{r}}_{v}t_{vvv} + \hat{\mathbf{r}}_{v}s_{vvv} + \hat{\mathbf{r}}_{v}t_{vvv} + \hat{\mathbf{r}}_{v}s_{vvv} + \hat{\mathbf{r}}_{v}t_{vvv} + \hat{\mathbf{r}}_{v}t_{vvv} + \hat{\mathbf{r}}_{v}s_{vvv} + \hat{\mathbf{r}}_{v}t_{vvv} + \hat{\mathbf{r}}_{v}s_{vvv} + \hat{\mathbf{r}}_{v}t_{vvv} + \hat{\mathbf{r}}_{v}s_{vvv} + \hat{\mathbf{r}}_{v}t_{vvv} + \hat{\mathbf{r}}_{v}t_{vvvv} + \hat{\mathbf{r}}_{v}t_{vvvv} + \hat{\mathbf{r}}_{v}t_{vvvv} + \hat{\mathbf{r}}_{v}t_{vvvv} + \hat{\mathbf{r}}_{v}t_{vvvv} + \hat{\mathbf{r}}_{v}t_{vvvv} + \hat{\mathbf{r}}_{v}t_{vvvv}$$

Using (3.57)-(3.60), the following is derived as the expression of arbitrary parametrization partial derivatives with geometric invariants of lines of curvatures and the s(u, v), t(u, v) reparametrization:

$$\mathbf{r}_{u} = \begin{bmatrix} s_{u} \\ t_{u} \\ 0 \end{bmatrix}_{D}^{}, \ \mathbf{r}_{v} = \begin{bmatrix} s_{v} \\ t_{v} \\ 0 \end{bmatrix}_{D}^{}$$
$$\mathbf{r}_{uv} = \begin{bmatrix} s_{uu} - \kappa_{2g}t_{u}^{2} \\ t_{uu} + \kappa_{1g}s_{u}^{2} \\ \kappa_{2n}t_{u}^{2} + \kappa_{1n}s_{u}^{2} \end{bmatrix}_{D}^{}, \ \mathbf{r}_{uv} = \begin{bmatrix} s_{uv} - \kappa_{2g}t_{u}t_{v} \\ t_{uv} + \kappa_{1g}s_{u}s_{v} \\ \kappa_{2n}t_{u}t_{v} + \kappa_{1n}s_{u}s_{v} \end{bmatrix}_{D}^{}, \ \mathbf{r}_{vv} = \begin{bmatrix} s_{vv} - \kappa_{2g}t_{v}^{2} \\ t_{vv} + \kappa_{1g}s_{v}^{2} \\ \kappa_{2n}t_{v}^{2} + \kappa_{1n}s_{v}s_{v} \end{bmatrix}_{D}^{}$$

and

$$\mathbf{r}_{uuu} = \begin{bmatrix} (\kappa_{2n}\tau_2 - \hat{\kappa}'_{2g})t_u^3 - s_u^3\kappa_1^2 - 3\kappa_{2g}t_ut_{uu} + s_{uuu} \\ (-\kappa_{1n}\tau_1 + \hat{\kappa}'_{1g})s_u^3 - t_u^3\kappa_2^2 + t_{uuu} + 3\kappa_{1g}s_us_{uu} \\ (\kappa_{2g}\tau_2 + \hat{\kappa}'_{2n})t_u^3 + (\kappa_{1g}\tau_1 + \hat{\kappa}'_{1n})s_u^3 + 3\kappa_{2n}t_ut_{uu} + 3\kappa_{1n}s_us_{uu} \end{bmatrix}_D$$

$$\mathbf{r}_{vvv} = \begin{bmatrix} -\kappa_1^2 s_v^3 + (\kappa_{2n}\tau_2 - \hat{\kappa}'_{2g})t_v^3 - 3\kappa_{2g}t_{vv}t_v + s_{vvv} \\ (-\kappa_{1n}\tau_1 + \hat{\kappa}'_{1g})s_v^3 + 3\kappa_{1g}s_{vv}s_v - \kappa_2^2 t_v^3 + t_{vvv} \\ (\kappa_{1g}\tau_1 + \hat{\kappa}'_{1n})s_v^3 + 3\kappa_{1n}s_{vv}s_v + (\kappa_{2g}\tau_2 + \hat{\kappa}'_{2n})t_v^3 + 3\kappa_{2n}t_{vv}t_v \end{bmatrix}_D$$

$$\mathbf{r}_{uuv} = \begin{bmatrix} -\kappa_1^2 s_u^2 s_v + ((\kappa_{2n}\tau_2 - \hat{\kappa}'_{2g})t_u^2 - \kappa_{2g}t_{uu})t_v - 2\kappa_{2g}t_{uv}t_u + s_{uuv} \\ ((-\kappa_{1n}\tau_1 + \hat{\kappa}'_{1g})s_u^2 + \kappa_{1g}s_{uu})s_v - \kappa_2^2 t_u^2 t_v + 2\kappa_{1g}s_{uv}s_u + t_{uuv} \\ ((\kappa_{1g}\tau_1 + \hat{\kappa}'_{1n})s_u^2 + \kappa_{1n}s_{uu})s_v + ((\kappa_{2g}\tau_2 + \hat{\kappa}'_{2n})t_u^2 + \kappa_{2n}t_{uu})t_v + 2\kappa_{1n}s_{uv}s_u + 2\kappa_{2n}t_{uv}t_u \end{bmatrix}_D$$

Please note that κ_{1g} and κ_{1n} are just convenience of notation. The g and n subscripts can be placed anywhere in multiplications, for example $\kappa_{1g}\tau_1 = \kappa_1\tau_1\sin\alpha$.

3.3.3 Geometric continuity and lines of curvature

A straightforward generalization of geometric continuity to surfaces can be stated as [51]

Definition 4 Two parametric surfaces $\mathbf{r}(u, v)$ and $\mathbf{p}(z, w)$ are G^n continuous at (u_0, v_0) and (z_0, w_0) if and only if there exists a reparameterization $u = u(\tilde{u}, \tilde{v})$, and $v = v(\tilde{u}, \tilde{v})$ such that $\tilde{\mathbf{r}}(\tilde{u}, \tilde{v})$ and $\mathbf{p}(z, w)$ are C^n continuous at $\mathbf{r}(t_0, u_0)$ and $\mathbf{p}(z_0, w_0)$.

Once again in practice, by reparameterization the majority of cases mean orientation preserving mappings.

Conditions of G^1 and G^2 continuity can be phrased using purely geometric invariants: two surfaces are G^1 at a point iff their unit normal vectors coincide. Additionally, they are G^2 as well iff their principal directions and curvatures are equal.

Let us investigate the relationship between G^n continuity of lines of curvature and G^n continuity of surfaces. If the lines of curvatures are G^n at a point, then the natural parametrization of the surface is C^n there. As a result, the two surfaces are G^n at the join.

The converse, however, is not true: G^n continuity of two surfaces does not guarantee G^n continuity of lines of curvature.

As an example of this, let us consider the join of $\mathbf{r}(u, v)$ and $\mathbf{p}(z, w)$ and the case of second order geometric continuity at a common point. Because the two surfaces are G^2 , for some $\tilde{\mathbf{r}}(\tilde{u}, \tilde{v})$ reparametrization of $\mathbf{r}(u, v)$, the equities

$$\tilde{\mathbf{r}}_{\tilde{u}} = \begin{bmatrix} s_{\tilde{u}} \\ t_{\tilde{u}} \\ 0 \end{bmatrix}_{D} = \begin{bmatrix} s_{z} \\ t_{z} \\ 0 \end{bmatrix}_{D} , \quad \tilde{\mathbf{r}}_{\tilde{u}\tilde{u}} = \begin{bmatrix} s_{\tilde{u}\tilde{u}} - \kappa_{2g}^{\mathbf{r}}t_{\tilde{u}}^{2} \\ t_{\tilde{u}\tilde{u}} + \kappa_{1g}^{\mathbf{r}}s_{\tilde{u}}^{2} \\ \kappa_{2n}^{\mathbf{r}}t_{\tilde{u}}^{2} + \kappa_{1n}^{\mathbf{r}}s_{\tilde{u}}^{2} \end{bmatrix}_{D} = \begin{bmatrix} s_{zz} - \kappa_{2g}^{\mathbf{s}}t_{z}^{2} \\ t_{zz} + \kappa_{1g}^{\mathbf{s}}s_{z}^{2} \\ \kappa_{2n}^{\mathbf{s}}t_{z}^{2} + \kappa_{1n}^{\mathbf{s}}s_{z}^{2} \end{bmatrix}_{D}$$

hold, where the upper indices of geometric invariants denote the surface, for example $\kappa_{in}^{\mathbf{r}}$ are the principal curvature functions of surface $\mathbf{r}(u, v)$, i = 1, 2. This notation will be used for higher order derivatives of the geometric invariants of lines of curvatures, that is, $\kappa_i^{\mathbf{r}}, \tau_i^{\mathbf{r}}$ are the curvature and torsion of the lines of curvatures of $\mathbf{r}(u, v)$, i = 1, 2.

A G^2 join implicates the equality of principal normal curvatures $\kappa_{in}^{\mathbf{r}} = \kappa_{in}^{\mathbf{s}}$, i = 1, 2. However, the geometric interpretation of G^2 does not stipulate anything on geodesic curvatures. Indeed, since $s_{\tilde{u}\tilde{u}}$ and $t_{\tilde{u}\tilde{u}}$ are degrees of freedom of reparametrization, they can be chosen such that

$$s_{\tilde{u}\tilde{u}} - \kappa_{2g}^{\mathbf{r}} t_{\tilde{u}}^2 = s_{zz} - \kappa_{2g}^{\mathbf{s}} t_z^2$$
$$t_{\tilde{u}\tilde{u}} + \kappa_{1g}^{\mathbf{r}} s_{\tilde{u}}^2 = t_{zz} + \kappa_{1g}^{\mathbf{s}} s_z^2$$

hold without the equality of geodesic curvatures. Because geometric continuity is defined modulo reparametrizations, the s_u , t_u , s_{uu} , t_{uu} coefficients may attain arbitrary values as long as $s_u^2 + t_u^2 \neq 0$ holds.

This extends to higher order geometric continuity. As a result, the geometric reconstruction constraints are masked by reparametrization in every coordinate of a partial derivative of total order n where a partial derivative of reparametrization of total order n appears.

In the case of order *n* geometric constraints, these are $\hat{\kappa}_{ig}^{(n-2)}$ and $\hat{\tau}_{in}^{(n-3)}$, i = 1, 2. That is, the Darboux *x* and *y* coordinates of two partial derivatives may coincide without equity of geodesic curvatures and normal torsions - their difference can be compensated by a reparametrization. Figure 3.7 shows two


Figure 3.7: On the left $\kappa_{1n} = -0.25, \kappa_{1g} = -1.125, \kappa_{2n} = 0.25, \kappa_{2g} = 0$. On the right $\kappa_{1n} = -0.25, \kappa_{1g} = -0.0625, \kappa_{2n} = 0.25, \kappa_{2g} = 0.4375$.

surfaces having same principal directions and curvatures and differing geodesic curvatures.

This is not the case for the Darboux z coordinates: even though the partial derivatives of reparametrization s(u, v), t(u, v) appear in those too, their total order of differentiation is strictly smaller than n, that is, their values were set in lower derivatives when ensuring a lower order geometric continuity.

Thus, only the *projections* of the *n*-th derivatives of lines of curvatures onto the surface normal are subject to geometric reconstruction constraints when dealing with geometric continuity. The orthogonal properties are masked by the artifact of reparametrization.

Consequently, reconstruction might seem to have an unmanageable amount of unknowns. For example, let us consider the order 3 partial derivatives of a surface, that is,

$$\mathbf{r}_{\tilde{u}\tilde{u}\tilde{u}} = \begin{bmatrix} (\kappa_{2n}\tau_2 - \hat{\kappa}'_{2g})t_{\tilde{u}}^3 - s_{\tilde{u}}^3\kappa_1^2 - 3\kappa_{2g}t_{\tilde{u}}t_{\tilde{u}\tilde{u}} + s_{\tilde{u}\tilde{u}\tilde{u}} \\ (-\kappa_{1n}\tau_1 + \hat{\kappa}'_{1g})s_{\tilde{u}}^3 - t_{\tilde{u}}^3\kappa_2^2 + t_{\tilde{u}\tilde{u}\tilde{u}} + 3\kappa_{1g}s_{\tilde{u}}s_{\tilde{u}\tilde{u}} \\ (\kappa_{2g}\tau_2 + \hat{\kappa}'_{2n})t_{\tilde{u}}^3 + (\kappa_{1g}\tau_1 + \hat{\kappa}'_{1n})s_{\tilde{u}}^3 + 3\kappa_{2n}t_{\tilde{u}}t_{\tilde{u}\tilde{u}} + 3\kappa_{1n}s_{\tilde{u}}s_{\tilde{u}\tilde{u}} \end{bmatrix}_D$$

and so on.

The z Darboux coordinates define four equations, that formally have eight unknowns ($\hat{\kappa}'_{in}$, $\kappa_{ig}\tau_i$, $s_{\tilde{u}\tilde{u}}$, $t_{\tilde{u}\tilde{u}}$, $s_{\tilde{u}\tilde{v}}$, $\tilde{u}_{\tilde{u}\tilde{v}}$, $s_{\tilde{v}\tilde{v}}$, $t_{\tilde{v}\tilde{v}}$, i = 1, 2). What is even worse, the number of unknowns seems to double for every successive total order of differentiation, because τ_g is tied to κ resulting in two potential unknowns $\hat{\kappa}'\tau_g$, $\kappa_g\hat{\tau}'$. However, the geodesic curvatures are actually already set if we know that the surface forms a G^3 join. This, and a more general statement, is shown in the following **Theorem 18** Let $\mathbf{r}(u, v)$ and $\mathbf{p}(z, w)$ be two surfaces joining at a point, and let us assume that neither principal curvature of the two surfaces is zero.

The surfaces are G^n , n > 1 at the join if and only if their lines of curvatures are G^{n-1} and $(\hat{\kappa}_{in}^{\mathbf{r}})^{(n-2)} + (\hat{\tau}_{ig}^{\mathbf{r}})^{(n-3)}\kappa_i^{\mathbf{r}} = (\hat{\kappa}_{in})^{(n-2)} + (\hat{\tau}_{ig}^{\mathbf{p}})^{(n-3)}\kappa_i^{\mathbf{p}}$.

Proof. \Leftarrow : if lines of curvatures are G^{n-1} , then the natural parametrization of $\mathbf{r}(u, v)$ and $\mathbf{p}(z, w)$ agree up to order (n-1), which implies that there exists a repamaterezitation of $\mathbf{r}(u, v)$, such that all partial derivatives up to order (n-1) agree with those of $\mathbf{p}(z, w)$. This reparametrization is that of $\mathbf{p}(z, w)$ from the – up to order (n-1) – common natural parametrization.

As a result, the $\partial_{\tilde{u}^i\tilde{v}^j}s$ and $\partial_{\tilde{u}^i\tilde{v}^j}t$ reparametrization coefficients of $\mathbf{r}(\tilde{u},\tilde{v})$ are equal to $\partial_{z^iw^j}s$ and $\partial_{z^iw^j}t$ of $\mathbf{s}(z,w)$, i+j < n. Let us denote the common reparametrization terms by

$$s_{ij} = \partial_{\tilde{u}^i \tilde{v}^j} s = \partial_{z^i w^j} s \quad , \quad t_{ij} = \partial_{\tilde{u}^i \tilde{v}^j} t = \partial_{z^i w^j} t$$

The geometric invariants are also equal, that is $\hat{\kappa}_{ig}^{(j)} = (\hat{\kappa}_{ig}^{\mathbf{r}})^{(j)} = (\hat{\kappa}_{ig}^{\mathbf{p}})^{(j)}$ and $\hat{\tau}_{in}^{(j-1)} = (\hat{\tau}_{in}^{\mathbf{r}})^{(j-1)} = (\hat{\tau}_{in}^{\mathbf{p}})^{(j-1)}, \ j = 0, 1, ..., n-3.$

The order *n* partial derivatives of the above \tilde{u}, \tilde{v} reparametrization of **r** are such that each *x* and *y* Darboux coordinate possess a scalar degree of freedom, $\partial_{\tilde{u}^i\tilde{v}^{n-i}s}$ and $\partial_{\tilde{u}^i\tilde{v}^{n-i}t}$, respectively, i = 0, ..., n. They appear along the *x* and *y* coordinates in the sequence of partial derivatives. Hence, these can be used to equate the *x*, *y* Darboux coordinates of $\partial_{\tilde{u}^j\tilde{v}^{n-j}}\mathbf{r}$ to that of $\partial_{z^jw^{n-j}}\mathbf{p}$, regardless of the values of $(\hat{\kappa}_{ig})^{(n-2)}, (\hat{\tau}_{in})^{(n-3)}, (\hat{\kappa}_{ig})^{(n-2)}, (\hat{\tau}_{in})^{(n-3)}, i = 1, 2$.

The z Darboux coordinates of partial derivatives of total order n consist of reparametrization derivatives up to total order (n-1) and lines of curvature and torsion derivatives up to order (n-3) and (n-4) – which are known to be equal –, and of terms of $\hat{\kappa}_{in}^{(n-2)}$, $\hat{\tau}_{ig}^{(n-3)}$, i = 1, 2. The latter pair of two quantities are unknown.

From Corollary 17 and the rules of multivariable derivation of composite functions, it follows that the z coordiante of partial derivatives of total order n can be written as

$$z_{\tilde{u}^{j}\tilde{v}^{n-j}}^{\mathbf{r}} = \left((\hat{\kappa}_{1n}^{\mathbf{r}})^{(n-2)} + (\hat{\tau}_{1g}^{\mathbf{r}})^{(n-3)} \kappa_1 \right) s_{\tilde{u}}^j s_{\tilde{v}}^{n-j} + \left((\hat{\kappa}_{2n}^{\mathbf{r}})^{(n-2)} + (\hat{\tau}_{2g}^{\mathbf{r}})^{(n-3)} \kappa_2 \right) t_{\tilde{u}}^j t_{\tilde{v}}^{n-j} + c_{\tilde{u}^{j}\tilde{v}^{n-j}}^{\mathbf{r}} ,$$

where $c_{\tilde{u}^i\tilde{v}^{n-i}}$ denotes the remaining terms of the z Darboux coordinate, i.e.,

$$c_{\tilde{u}^{j}\tilde{v}^{n-j}}^{\mathbf{r}} = z_{\tilde{u}^{j}\tilde{v}^{n-j}}^{\mathbf{r}} - \left((\hat{\kappa}_{1n}^{\mathbf{r}})^{(n-2)} + (\hat{\tau}_{1g}^{\mathbf{r}})^{(n-3)}\kappa_1 \right) s_{\tilde{u}}^{j} s_{\tilde{v}}^{n-j} - \left((\hat{\kappa}_{2n}^{\mathbf{r}})^{(n-2)} + (\hat{\tau}_{2g}^{\mathbf{r}})^{(n-3)}\kappa_2 \right) t_{\tilde{u}}^{j} t_{\tilde{v}}^{n-j}.$$

Since $c_{\tilde{u}^j\tilde{v}^{n-j}}^{\mathbf{r}}$ and $c_{z^jw^{n-j}}^{\mathbf{p}}$ are made up of terms of reparametrization partial derivatives up to order (n-1) and curvature and torsion derivatives up to order (n-3) and (n-4), they are equal, i.e. $c_{u^jv^{n-j}} = c_{\tilde{u}^j\tilde{v}^{n-j}}^{\mathbf{r}} = c_{z^jw^{n-j}}^{\mathbf{p}}$.

Thus, equality of $z_{\tilde{u}^j\tilde{v}^{n-j}}^{\mathbf{r}}$ and $z_{z^jw^{n-j}}^{\mathbf{p}}$ depends on equality $(\hat{\kappa}_i^{\mathbf{r}}n)^{(n-2)} + (\hat{\tau}_{ig}^{\mathbf{r}})^{(n-3)}\kappa_i^{\mathbf{r}} = (\hat{\kappa}_{in}^{\mathbf{p}})^{(n-2)} + (\hat{\tau}_{ig}^{\mathbf{p}})^{(n-3)}\kappa_i^{\mathbf{p}}$, which holds by assumption of the theorem.

 \Rightarrow : if two surfaces are G^n , then there exists a reparametrization, such that the two surfaces are C^n . Let this reparametrization be $\mathbf{r}(\tilde{u}, \tilde{v})$ and $\mathbf{p}(z, q)$.

If none of the principal curvatures vanish, then the partial derivatives of the surface up to order n uniquely determine the differential geometric invariants of the lines of curvatures up to order n - 1, see Appendix D for additional details. That is, the lines of curvatures are G^{n-1} .

Because lines of curvatures are G^{n-1} , $\hat{\kappa}_{ig}^{(j)} = (\hat{\kappa}_{ig}^{\mathbf{r}})^{(j)} = (\hat{\kappa}_{ig}^{\mathbf{p}})^{(j)}$ and $\hat{\tau}_{in}^{(j-1)} = (\hat{\tau}_{in}^{\mathbf{r}})^{(j-1)} = (\hat{\tau}_{in}^{\mathbf{p}})^{(j-1)}$, j = 0, 1, ..., n-3 hold, which, in turn, yields that $s_{ij} = \partial_{\tilde{u}^i \tilde{v}^j} s = \partial_{z^i w^j} s$, $t_{ij} = \partial_{\tilde{u}^i \tilde{v}^j} t = \partial_{z^i w^j} t$.

All that remains to be seen is that G^n join of $\mathbf{r}(u, v)$ and $\mathbf{p}(z, w)$ implies equality $(\hat{\kappa}_{in}^{\mathbf{r}})^{(n-2)} + (\hat{\tau}_{ig}^{\mathbf{r}})^{(n-3)} \kappa_i^{\mathbf{r}} = (\hat{\kappa}_{in}^{\mathbf{p}})^{(n-2)} + (\hat{\tau}_{ig}^{\mathbf{p}})^{(n-3)} \kappa_i^{\mathbf{p}}$.

Equity conditions $\partial_{\tilde{u}^j\tilde{v}^{n-j}}\mathbf{r} = \partial_{z^jw^{n-j}}\mathbf{p}$ can be written as

$$\begin{bmatrix} s_{10}^{n} & t_{10}^{n} \\ \dots & \dots \\ s_{10}^{n-i} s_{01}^{i} & t_{10}^{n-i} t_{01}^{i} \\ \dots & \dots \\ s_{01}^{n} & t_{01}^{n} \end{bmatrix} \cdot \begin{bmatrix} (\hat{\kappa}_{1n}^{\mathbf{r}})^{(n-2)} + T_{1}^{\mathbf{r},(n-3)} \\ (\hat{\kappa}_{2n}^{\mathbf{r}})^{(n-2)} + T_{2}^{\mathbf{r},(n-3)} \end{bmatrix} + \begin{bmatrix} c_{n0} \\ \dots \\ c_{i,n-i} \\ \dots \\ c_{0n} \end{bmatrix} = (3.63)$$
$$\begin{bmatrix} s_{10}^{n} & t_{10}^{n} \\ \dots & \dots \\ s_{10}^{n-i} s_{01}^{i} & t_{10}^{n-i} t_{01}^{i} \\ \dots & \dots \\ s_{01}^{n} & t_{01}^{n} \end{bmatrix} \cdot \begin{bmatrix} (\hat{\kappa}_{1n}^{\mathbf{p}})^{(n-2)} + T_{1}^{\mathbf{p},(n-3)} \\ (\hat{\kappa}_{2n}^{\mathbf{p}})^{(n-2)} + T_{2}^{\mathbf{p},(n-3)} \end{bmatrix} + \begin{bmatrix} c_{n0} \\ \dots \\ c_{i,n-i} \\ \dots \\ c_{0n} \end{bmatrix}$$

where $c_{j,n-j} = c_{\tilde{u}^j \tilde{v}^{n-j}}^{\mathbf{r}} = c_{z^j w^{n-j}}^{\mathbf{p}}$ and the new unknown $T_i^{\mathbf{x},(n)}$ denotes $T_i^{\mathbf{x},(n)} = (\hat{\tau}_{1g}^{\mathbf{x}})^{(n)} \kappa_i^{\mathbf{x}}, \mathbf{x} \in {\mathbf{r}, \mathbf{p}}.$

Subtracting the right hand side of (3.63) from the equation yields the overdetermined homogeneous system of linear equations

$$\underbrace{\begin{bmatrix} s_{10}^{n} & t_{10}^{n} \\ \dots & \dots \\ s_{10}^{n-i}s_{01}^{i} & t_{10}^{n-i}t_{01}^{i} \\ \dots & \dots \\ s_{01}^{n} & t_{01}^{n} \end{bmatrix}}_{A} \cdot \underbrace{\left(\begin{bmatrix} (\hat{\kappa}_{1n}^{\mathbf{r}})^{(n-2)} + T_{1}^{\mathbf{r},(n-3)} \\ (\hat{\kappa}_{2n}^{\mathbf{r}})^{(n-2)} + T_{2}^{\mathbf{r},(n-3)} \end{bmatrix} - \begin{bmatrix} (\hat{\kappa}_{1n}^{\mathbf{p}})^{(n-2)} + T_{1}^{\mathbf{p},(n-3)} \\ (\hat{\kappa}_{2n}^{\mathbf{p}})^{(n-2)} + T_{2}^{\mathbf{p},(n-3)} \end{bmatrix} \right)}_{\Delta P} = \mathbf{0}$$

Using $P_i^{\mathbf{x},n} = (\hat{\kappa}_{in}^{\mathbf{x}})^{(n-2)} + (\hat{\tau}_{ig}^{\mathbf{x}})^{(n-3)} \kappa_i^{\mathbf{x}}$ and $\Delta P_i^n = P_i^{\mathbf{r},n} - P_i^{\mathbf{p},n}$, the above can be written as

$$A \cdot \begin{bmatrix} \Delta P_1^n \\ \Delta P_2^n \end{bmatrix} = \mathbf{0}$$

for which $[\Delta P_1^n, \Delta P_2^n]^T = \mathbf{0}$ is a unique solution if and only if rank(A) = 2. Because, by regularity of parametrization, $s_u^2 + t_u^2 \neq 0$, $s_v^2 + t_v^2 \neq 0$ and $\nexists \alpha \neq 0$: $\alpha s_u = s_v \wedge \alpha t_u = t_v$, matrix A indeed has a rank of two, thus $(\hat{\kappa}_{in}^{\mathbf{r}})^{(n-2)} + (\hat{\tau}_{iq}^{\mathbf{r}})^{(n-3)} \kappa_i^{\mathbf{r}} = (\hat{\kappa}_{in}^{\mathbf{p}})^{(n-2)} + (\hat{\tau}_{iq})^{(n-3)} \kappa_i^{\mathbf{p}}$ holds.

3.3.4 Formalization of general geometric Hermite interpolation

Let $\mathbf{E} = (\mathbf{D}_1, \mathbf{D}_2)$ denote an order *n* GH surface data tuple, which consists of two $\mathbf{D}_i = (\mathbf{p}_i; \mathbf{t}_i; \mathbf{n}_i, \kappa_i, \mathbf{b}_i; \hat{\kappa}'_i, \tau_i, ...), i = 1, 2$ order *n* GH curve data tuples, specifying the properties of the lines of curvature, with $\mathbf{p} = \mathbf{p}_1 = \mathbf{p}_2$ and $\mathbf{t}_1 \cdot \mathbf{t}_2 = 0$.

Reconstruction of **E** at a point of a regular parametric surface $\mathbf{r}(u, v)$ is formulated by expressing the dependence of the $\partial_{u^k} \partial_{v^l} \mathbf{r}$ partial derivatives on the geometric invariants of the lines of curvatures. The formulations are given in the $D = (\mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$ Darboux frame, where $\mathbf{m} = \mathbf{t}_1 \times \mathbf{t}_2$.

Similarly to the case of curves, these dependencies are expressed a matrix-

vector multiplication as

$$\begin{bmatrix}
\mathbf{r} \\
\mathbf{r}_{u} \\
\mathbf{r}_{v} \\
\mathbf{r}_{vu} \\
\mathbf{r}_{vv} \\
\mathbf{r}_{vv} \\
\dots \\
F \cdot \mathbf{q}
\end{bmatrix} = \underbrace{\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & s_{u} & t_{u} & 0 \\
0 & s_{v} & t_{v} & 0 \\
0 & s_{uv} - \kappa_{2g}t_{u}^{2} & t_{uu} + \kappa_{1g}s_{u}^{2} & \kappa_{1n}s_{u}^{2} + \kappa_{2n}t_{u}^{2} \\
0 & s_{uv} - \kappa_{2g}t_{u}t_{v} & t_{uv} + \kappa_{1g}s_{u}s_{v} & \kappa_{1n}s_{u}s_{v} + \kappa_{2n}t_{u}t_{v} \\
0 & s_{vv} - \kappa_{2g}t_{v}^{2} & t_{vv} + \kappa_{1g}s_{v}^{2} & \kappa_{1n}s_{v}^{2} + \kappa_{2n}t_{v}^{2} \\
\dots & \dots & \dots & \dots \\
\hline
G(s_{u},s_{v},\dots)
\end{bmatrix} \cdot \underbrace{\begin{bmatrix}
\mathbf{p} \\
\mathbf{t}_{1} \\
\mathbf{t}_{2} \\
\mathbf{m} \\
\mathbf{d}
\end{bmatrix}}_{\mathbf{d}}$$
(3.64)

where $s_u^2 + t_u^2 \neq 0$, $s_v^2 + t_v^2 \neq 0$, and $s_u t_v - t_u s_v \neq 0$. The latter expresses the linear independence of the *u* and *v* partial derivatives by stipulating that the cross of product of $[s_u, t_u]^T$, $[s_v, t_v]^T$ shall not be zero.

The entries of geometric reconstruction matrix G are made up of geometric constraints from **E** and degrees of freedom of parametrization, in the form of the $s_u, s_v, t_u, t_v, ...$ derivatives.

If one of the principal curvatures vanish, the order of differentiation needs to be increased until the torsion of the corresponding line of curvature is no longer masked by the zero curvature, similarly to the case of curves. If the *h*-th derivative of a principal curvature is the first non-zero value, then partial derivatives up to total order of n + h need to be added to (3.64).

The system of (3.64) can be cast into a form such that its degrees of freedom are transformed into Darboux coordinates of partial derivatives. Let $\partial_{u^k v^l} \mathbf{r} = [x_{u^k v^l}, y_{u^k v^l}, z_{u^k v^l}]_D^T$, making up the left hand side of (3.64). Then

$$s_{u} = x_{u} , \ s_{v} = x_{v} ,$$

$$s_{uu} = x_{uu} + \kappa_{2g}y_{u}^{2} , \ t_{uu} = y_{uu} - \kappa_{1g}x_{u}^{2} ,$$

$$s_{uv} = x_{uv} + \kappa_{2g}y_{u}y_{v} , \ t_{uv} = y_{uv} - \kappa_{1g}x_{u}x_{v} ,$$

$$s_{vv} = x_{vv} + \kappa_{2g}y_{v}^{2} , \ t_{vv} = y_{vv} - \kappa_{1g}x_{v}^{2} ,$$

so (3.64) becomes

$$\begin{bmatrix} \mathbf{r} \\ \mathbf{r}_{u} \\ \mathbf{r}_{v} \\ \mathbf{r}_{uu} \\ \mathbf{r}_{uv} \\ \mathbf{r}_{vv} \\ \mathbf{r}_{vv} \\ \mathbf{r}_{vv} \\ \mathbf{r}_{vv} \\ \mathbf{r}_{vv} \\ \mathbf{r}_{vv} \\ \dots \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x_{u} & y_{u} & 0 \\ 0 & x_{v} & y_{v} & 0 \\ 0 & x_{uu} & y_{uu} & \kappa_{1n} x_{u}^{2} + \kappa_{2n} y_{u}^{2} \\ 0 & x_{uv} & y_{uv} & \kappa_{1n} x_{u} x_{v} + \kappa_{2n} y_{u} y_{v} \\ 0 & x_{vv} & y_{vv} & \kappa_{1n} x_{v}^{2} + \kappa_{2n} y_{v}^{2} \\ \dots & \dots & \dots & \dots \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p} \\ \mathbf{t}_{1} \\ \mathbf{t}_{2} \\ \mathbf{m} \end{bmatrix}$$
(3.65)

where $x_u^2 + y_u^2 \neq 0$ and $x_v^2 + y_v^2 \neq 0$ and $x_u y_v - y_u x_v \neq 0$.

The formulation of (3.65) emphasizes the masking effect of reparametrization on lines of curvature geometric invariant reconstruction.

Let us now consider a row of (3.65) that corresponds to a total order k+l = n > 2 partial derivative. It is of the form of

$$\begin{bmatrix} \dots \\ \partial_{u^{k}v^{l}}\mathbf{r} \\ \dots \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots & \dots \\ 0 & x_{u^{k}v^{l}} & y_{u^{k}v^{l}} & K^{\mathbf{r},n} + c_{u^{k}v^{l}} \\ \dots & \dots & \dots & \dots \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p} \\ \mathbf{t}_{1} \\ \mathbf{t}_{2} \\ \mathbf{m} \end{bmatrix}$$
(3.66)

-

where $K^{\mathbf{r},n} = P_1^{\mathbf{x},n} x_u^k x_v^l + P_2^{\mathbf{x},n} y_u^k y_v^l$, and $P_i^{\mathbf{x},n} = (\hat{\kappa}_{in}^{\mathbf{r}})^{(n-2)} + (\hat{\tau}_{ig}^{\mathbf{r}})^{(n-3)} \kappa_i^{\mathbf{r}}$, and $c_{u^k v^l}^{\mathbf{r}} = z_{u^k v^l}^{\mathbf{r}} - K^{\mathbf{r},n}$.

In the case of tensor-product surfaces

$$\mathbf{r}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{q}_{ij} F_i^n(u) F_j^m(v)$$

where control data $\mathbf{q}_{ij} \in \mathbb{R}^3 \vee \mathbb{E}^3$, left hand side of (3.64) may be written in a linearized manner as

$$\underbrace{ \begin{bmatrix} F_0^n(u_0)F_0^m(v_0) & F_1^n(u_0)F_0^m(v_0) & \dots & F_n^n(u_0)F_0^m(v_0) & \dots & F_n^n(u_0)F_m^m(v_0) \\ (F_0^n)'(u_0)F_0^m(v_0) & (F_1^n)'(u_0)F_0^m(v_0) & \dots & (F_n^n)'(u_0)F_m^m(v_0) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ F_{u_0,v_0} \end{bmatrix}}_{F_{u_0,v_0}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \mathbf{q}_{10} \\ \dots \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \mathbf{q}_{nm} \end{bmatrix}}_{\mathbf{q}_{nm}} \cdot \underbrace{ \begin{bmatrix} \mathbf{q}_{00} \\ \mathbf{q}_{10} \\ \mathbf$$

If the total number of control data is (n+1)(m+1) in this case, we refer to the type of the unknown interpolant surfaces as an order (n, m) surface. For polynomials, these are the degree n and degree m polynomial surfaces in the two respective parameter lines.

It is important that control data that have zero coefficients in every evaluation are to be omitted from \mathbf{q} . The remaining control points are said to be *affected by geometric reconstruction*. Means of identifying affected control data depends on the choice of surface representation basis and reconstruction parameters. For example, as it was illustrated previously in Figures 3.2 and C.1, computation of position, surface normal, principal curvatures and directions require 6 Bézier control points at corners both over triangular and rectangular domains.

In general GH interpolation of surfaces, we are given (k+1) pieces of (u_i, v_i) parameter values and $\mathbf{E}^{(i)}$ order n_i GH surface data tuples, i = 0, 1, ..., k, and the reconstruction system of equations

$$\begin{bmatrix} F_{u_0,v_0} \\ F_{u_1,v_1} \\ \dots \\ F_{u_m,v_m} \end{bmatrix} \cdot \mathbf{q} = \begin{bmatrix} G_0 & 0 & \dots & 0 \\ 0 & G_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_m \end{bmatrix} \cdot \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \dots \\ \mathbf{d}_m \end{bmatrix}$$
(3.67)

is to be solved, where \mathbf{d}_i denotes the Darboux-frame of GH tuple $\mathbf{E}^{(i)}$, G_i its associated geometric reconstruction matrix and F_{u_i,v_i} contain the basis function evaluations up to the necessary order at (u_i, v_i) . In short, we refer to these type of GH interpolation problems to be of order $(n_0, n_1, ..., n_k)$.

Example Let us consider the four corner point-normal reconstruction problem and its solution with quadrilateral bi-quadratic Bézier patches. For the sake of simplicity, let us use two-dimensional indices and assume that reconstruction is to be carried out at endpoints.

The formulation of the problem is

		-									_						
$\mathbf{r}(0,0)$		1		0	0	0	0	0		0	0						
$\mathbf{r}_u(0,0)$		_	2	2	0	0	0	0		0	0						
${\bf r}_v(0,0)$		_	2	0	0	2	0	0		0	0	$\left[\mathbf{q}_{0}\right]$					
r(1,0)		0		0	1	0	0	0		0	0	$ \mathbf{q}_1 $.0				
${\bf r}_u(1,0)$	-	0		-2	2	0	0	0		0	0	\mathbf{q}_2	:0				
${\bf r}_v(1,0)$		0		0	-2	0	2	0		0	0		1			(2.2.2	
r(0,1)	=			0	0	0	0	1		0	$0 \cdot $		1			(3.68))
$\mathbf{r}_{u}(0,1)$				0	0	0	0	-2	2	2	0	12 0(0				
$r_{u}(0,1)$				0	0	-2	2 0	2		0			2				
r(1, 1)				0	0	0	0	0		0	1	0-11	2				
$\mathbf{r}_{(1,1)}$				0	0	0	0	0		_2	$\frac{1}{2}$	L ₄₇	²]				
$\mathbf{r}_{u}(1,1)$				0	0	0	-2	2 0		0	$\frac{1}{2}$						
	_	L		Ū	0	Ŭ	_			0	-]					Γ (00)]	1
			_												_	$\mathbf{p}^{(00)}$	
			1	0	0	0	0	0	0	0	0	0	0	0		$\mathbf{t}_{1}^{(00)}$	
			0	$x_{u}^{(00)}$	$y_{u}^{(00)}$	0	0	0	0	0	0	0	0	0		$ {f t}_2^{(00)} $	
			0	$x_{v}^{(00)}$	$y_{v}^{(00)}$	0	0	0	0	0	0	0	0	0		$\mathbf{p}^{(10)}$	
			0	0	0	1	0	0	0	0	0	0	0	0		P	
			0	0	0	0	$x_{u}^{(10)}$	$y_{u}^{(10)}$	0	0	0	0	0	0		$\mathbf{U}_{1}^{(10)}$	
	_	_	0	0	0	0	$x_{v}^{(10)}$	$y_{v}^{(10)}$	0	0	0	0	0	0		$ \mathbf{t}_{2}^{(10)} $	
	-	_	0	0	0	0	0	0	1	0	0	0	0	0		$\mathbf{p}^{(01)}$	
			0	0	0	0	0	0	0	$x_{u}^{(00)}$	$y_u^{(00)}$	0	0	0		$t^{(01)}$	
			0	0	0	0	0	0	0	$x_{v}^{(00)}$	$y_{v}^{(00)}$	0	0	0		(01)	
			0	0	0	0	0	0	0	0	0	1	0	0		$\mathbf{t}_2^{(1)}$	
			0	0	0	0	0	0	0	0	0	0	$x_{u}^{(00)}$	$y_{u}^{(00)}$		$ {f p}^{(11)} $	
			0	0	0	0	0	0	0	0	0	0	$x_{v}^{(00)}$	$y_{v}^{(00)}$		$t_{1}^{(11)}$	
			-												-	t ¹ (11)	
																L ^u 2	

Please note that control point \mathbf{q}_{11} and its – all zero – coefficients are omitted from the above. In practice, when computing closed-form solutions via Moore-Penrose pseudo-inverse, this is a necessary step to avoid the singular value decomposition of $F^T \cdot F$.

Example Let us consider the four corner, second order GH interpolation problem. Then a GH surface data tuple at a knot consist of data

$$\mathbf{E}^{(ij)} = \left(\left(\mathbf{p}^{(ij)}; \mathbf{t}_1^{(ij)}; \kappa_1^{(ij)}, \mathbf{n}_1^{(ij)}, \mathbf{b}_1^{(ij)} \right), \left(\mathbf{p}^{(ij)}; \mathbf{t}_2^{(ij)}; \kappa_2^{(ij)}, \mathbf{n}_2^{(ij)}, \mathbf{b}_2^{(ij)} \right) \right), \ i, j = 0, 1$$

According to (3.65) and (3.66), the geometric constraint matrix of a corner is of the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x_u^{(ij)} & y_u^{(ij)} & 0 \\ 0 & x_v^{(ij)} & y_v^{(ij)} & 0 \\ 0 & x_{uu}^{(ij)} & y_{uu}^{(ij)} & \kappa_{1n}^{(ij)} (x_u^{(ij)})^2 + \kappa_{2n}^{(ij)} (y_u^{(ij)})^2 \\ 0 & x_{uv}^{(ij)} & y_{uv}^{(ij)} & \kappa_{1n}^{(ij)} x_u^{(ij)} x_v^{(ij)} + \kappa_{2n}^{(ij)} y_u^{(ij)} y_v^{(ij)} \\ 0 & x_{vv}^{(ij)} & y_{vv}^{(ij)} & \kappa_{1n}^{(ij)} (x_v^{(ij)})^2 + \kappa_{2n}^{(ij)} (y_v^{(ij)})^2 \end{bmatrix}$$

which shows that the second order GH interpolation (3.1) introduced at the beginning of this chapter is indeed the n = 2 case of the general GH interpolation defined here.

3.4 Solution of GH surface reconstruction

Let us study the solvability of system (3.67). For the sake of simplicity, let us assume that the $F_i^n(u), F_j^m(v)$ basis functions are linearly independent in the sense that matrix F is of full rank at all evaluation points.

The above assumption does not hold if there are unused control data listed in \mathbf{q} of (3.64). If that is the case, we obtain a reduced control data sequence \mathbf{q} by removing them. Since these did not affect any of the partial derivatives necessary for geometric reconstruction, we did not alter the problem by their removal.

Singularity of F may arise from improper choice of surface interpolant class as well: for example, all partial derivatives of order N > n vanish for a degree (n, n) integral polynomial surface, thus they cannot satisfy any nonzero geometric constraint. Consequently, matrix F will be rank deficient due to its all-zero rows. Removing these rows from F is equivalent to removing the geometric constraints on all partial derivative vectors of order N > n. This, however, changes the original GH problem and instead, a new choice of interpolant surface class is necessary.

Depending on bases, certain evaluation points may cause degeneracy of F too - these occurrences should be handled on a case by case basis, depending on the representation of the interpolant surfaces.

3.4.1 Exact solutions

The number of rows of an F_{u_i,v_i} matrix of order n_i equals to

$$1 + 2 + \dots + (n_i + 1) = \frac{(n_i + 1)(n_i + 2)}{2}$$

If all control data of an order (n + 1, m + 1) surface (for example degree (n, m) quadrilateral Bézier patches) are required for surface evaluations, the number of columns of F is (n + 1)(m + 1). That is, under the assumptions about the basis functions introduced above, F is invertible if

$$(n+1)(m+1) = \sum_{i=0}^{k} \frac{(n_i+1)(n_i+2)}{2}$$

Affected ontrol data of the interpolant are computed by $\mathbf{q} = F^{-1} \cdot G \cdot \mathbf{d}$.

Polynomial degree conditions, such as Theorem 7, can be also given for surface reconstruction, however, the problem of overlapping control points has to be taken into account. Indeed, as it was demonstrated by the second order GH interpolation section, even though bi-quartic Bézier surfaces have $5 \times 5 =$ 25 control points, possessing 75 scalar degrees of freedom in total, they can not guarantee reconstruction of arbitrary second order GH data tuples at four corners, constraining only $4 \times 8 = 24$ scalar degrees of freedom.

Considering four corner, symmetric GH interpolation of uniformly order n GH data tuples, the following holds

Theorem 19 Let there be given four $\mathbf{E}^{(i)}$ order n GH surface data tuples and let us consider the problem of four corner reconstruction of such data with quadrilateral polynomial surfaces. If $\kappa_1^{(i)}, \kappa_2^{(i)} \neq 0$, there is always a degree (2n+1) by (2n+1) polynomial solution to the problem.

Proof. This is a direct generalization of Theorem 13. Bézier control points required for reconstruction of GH data tuples do not overlap at any corner. Their values can be set by stipulating arbitrary Darboux coordinates in accordance with (3.67).

If principal curvature have h zeros at each corner, the solution in general needs to be found among the degree 2(n + h + 1) - 1 by 2(n + h + 1) - 1polynomial patches, n > 2. The worst case scenario, when all prescribed principal curvatures vanish, lead us to that reconstruction of order n GH surface data tuples may require polyonomial surfaces of degree up to 4n - 1 by 4n - 1, n > 2. In the case of third order GH interpolation of surfaces, this means degree 11×11 solutions. These correspond to the results of Theorem 7 regarding the degree of polynomial curve interpolants.

Now, let us return to the general case where (k + 1) GH data tuples are to be reconstructed and let us allow the right hand side degrees of freedom – the Darboux x and y coordinates of partial derivatives – vary and consider the overdetermined case, that is, when the total number of affected control points is strictly smaller than $\sum_{i=0}^{k} \frac{(n_i+1)(n_i+2)}{2}$.

Again, by the Rouché-Capelli theorem, an exact solution exists if and only if $rank(F) = rank([F, G(x_u, y_u, ...) \cdot \mathbf{d}]).$

Let $\mathbf{C} = \times_{i=1}^{d} C(F)$, where C(F) is the column space of F and d is the dimension of space in which the surface is embedded, i.e. d = 3 in our case, and let **G** denote the set of all possible right hand side of (3.65), that is,

$$\begin{aligned} \mathbf{G} &= \{ \mathbf{x} \in \mathbb{R}^{\overline{k} \times d} | \forall i \in \{0, .., k\} : \exists x_u^{(i)}, y_u^{(i)}, x_v^{(i)}, y_v^{(i)}, x_{uu}^{(i)}, .. \in \mathbb{R} : \\ & (x_u^{(i)})^2 + (y^{(i)})_u^2 \neq 0 \land (x^{(i)})_v^2 + (y^{(i)})_v^2 \neq 0 \land x_u^{(i)} y_v^{(i)} - y_u^{(i)} x_v^{(i)} \neq 0 : \\ & \mathbf{x} = G(x_u^{(0)}, y_u^{(0)}, .., x_u^{(1)}, y_u^{(1)}, .., x_u^{(k)}, y_u^{(k)}, ..) \cdot \mathbf{d} \} \;, \end{aligned}$$

where $\overline{k} = \sum_{i=0}^{k} \frac{(n_i+1)(n_i+2)}{2}$. Then if there is a right hand side that can be reconstructed by the control data, the intersection of **G** and **C** will not be zero, and this intersection determines the control data of the solution.

Theorem 20 A solution to the $(n_0, n_1, ..., n_k)$ order GH interpolation problem with (n+1)(m+1) affected control data exists in a linearly independent basis if and only if $\mathbf{C} \cap \mathbf{G} \neq \emptyset$.

3.4.2 Approximate solutions

Theorem 20 provides means to identify whether an actual GH surface interpolation problem has an exact solution or we need to find approximations.

Analogously to the case of curves, approximate solutions can be obtained by minimizing various real valued functionals. In general, this can be formulated as

$$\min_{\tilde{\mathbf{q}}} f(F, \tilde{\mathbf{q}}, G, \mathbf{d})$$

where $\tilde{\mathbf{q}}$ denotes all control data of the interpolant surface, not just the ones required for GH reconstruction.

The algebraic norm-based methods reviewed in Section 2.4.2.1 work the same way on surfaces as they did on curves, since, formally, the two problems are equivalent.

Thus, the LSQ-sense best interpolant has a closed-form solution, using the Moore-Penrose pseudoinverse as

$$\mathbf{q} = F^+ \cdot G \cdot \mathbf{d}$$

Note that only the reduced control data \mathbf{q} are listed in the equation - all control data that play no part in satisfying the prescribed geometric constraints have to be assigned values independently of GH interpolation. Compared to the curve case, this increases the available degrees of freedom considerably.

For example, four corner second order GH interpolation can be solved by bi-quintic integral polynomial patches for arbitrary input. The total degrees of freedom in bi-quintic polynomials is $6 \times 6 \times 3 = 108$, while the number of scalar constraints from GH interpolation totals at $4 \times 8 = 32$, leaving 76 scalar degrees of freedom available after GH reconstruction.

These degrees of freedom can be used to ensure higher order geometric or parametric continuous join of two GH interpolant patches. I show an example of this in Section 3.5.

Furthermore, degrees of freedom are also available for parametrization optimization of a single patch. The next point discusses these cases as well.

The norm-based functionals minimize the algebraic distance of the left and right hand side of equation (3.64). Euclidean distance based functionals can be obtained by using a geometric Taylor-like expansion of surfaces and considering the distance of these geometric Taylor expansions from that of an ideal, perfectly reconstructing surface.

For example, the paraboloid of

$$\tilde{\mathbf{p}}(u,v) = \begin{bmatrix} u\\v\\\frac{\kappa_1}{2}u^2 + \frac{\kappa_2}{2}v^2 \end{bmatrix}$$

has unit surface normal $\mathbf{m} = [0, 0, 1]^T$, principal curvature κ_1, κ_2 and corre-

sponding principal directions $\mathbf{t}_1 = [1, 0, 0]^T$, $\mathbf{t}_2 = [0, 1, 0]^T$ at (u, v) = (0, 0). This is the osculating paraboloid in the Darboux frame of a regular point of a surface. If given an **E** second order GH data tuple, it is reconstructed by

$$\mathbf{p}(u,v) = \mathbf{p} + [\mathbf{t}_1, \mathbf{t}_2, \mathbf{m}] \begin{bmatrix} u \\ v \\ \frac{\kappa_1}{2}u^2 + \frac{\kappa_2}{2}v^2 \end{bmatrix} .$$

The functional

$$e_i = \int \int ||\mathbf{q}(u,v) - \mathbf{p}(u,v)|| du dv$$

measures the deviation of the parametrization of the $\mathbf{q}(u, v)$ interpolant to $\mathbf{E}^{(i)}$ from that of $\mathbf{p}(u, v)$, as well as the error of reconstruction, up to normal projection of principal curvatures. By using the bi-quadratic geometric Taylor expansion of $\mathbf{q}(u, v)$ about the prescribed reconstruction parameter $(u_i, v_i) = (0, 0)$

$$T_{2,2}(\mathbf{q}(u,v)) = \begin{bmatrix} u \\ v \\ \frac{\kappa_1^{\mathbf{q}}(0,0)}{2}u^2 + \frac{\kappa_2^{\mathbf{q}}(0,0)}{2}v^2 \end{bmatrix}$$

the above can be transformed to take only the difference of reconstruction into account

$$e_i = \int \int ||T_{2,2}(\mathbf{q}(u,v)) - \mathbf{p}(u,v)|| du dv$$

3.4.3 Parametrization optimization

The strategy of separating geometric reconstruction constraints of second order GH reconstruction from the degrees of freedom of the interpolant's parametrization appeared in Moreton's paper [33], who used the latter to optimize the quality of a surface patch.

To achieve this, Moreton applied constrained nonlinear optimization of a real-valued quality functional, which, using our nomenclature here, can be written as

$$\min_{\tilde{\mathbf{q}}} f(\tilde{\mathbf{q}})$$
$$F \cdot \mathbf{q} = G \cdot \mathbf{d}$$

where $\tilde{\mathbf{q}}$ once again denotes all control data that defines the interpolant surface $\mathbf{q}(u, v)$.

If conditions of Theorem 20 hold, an exact reconstruction is guaranteed to exist, so once an enumeration of all feasible solutions is given, only the f() functional that determines its quality needs to be chosen.

A traditional fairness functional in surface design is the strain energy [33], which can be expressed as a surface integral in the form of

$$\int \left(\kappa_1^{\mathbf{q}}\right)^2 + \left(\kappa_2^{\mathbf{q}}\right)^2 dA \; .$$

Moreton argued that due to the fact that the above functional penalizes curvature, minimizing it results in unnecessarily flat surfaces, so instead, he proposed minimizing the variation of curvature by the functional

$$\int \left(\partial_{\mathbf{t}_1^{\mathbf{q}}} \kappa_1^{\mathbf{q}}\right)^2 + \left(\partial_{\mathbf{t}_2^{\mathbf{q}}} \kappa_2^{\mathbf{q}}\right)^2 dA ,$$

using which one can minimize the variation of principal normal curvature along principal directions, by computing the squared sums of directional derivatives of principal normal curvature.

Due to the computational cost of evaluating a surface integral that requires principal curvatures, one could instead try to minimize the squared length of the uu and vv partial derivatives of the surface via

$$\int \mathbf{q}_u^2 + \mathbf{q}_v^2 dA$$

however, this can be a very rough approximation of strain energy.

Alternatively, following our arguments in parametrization optimization of curves, one could try to minimize the deviation of the parametrization of the interpolant surface from natural parametrization.

Natural parametrization is composed of two components: parameter lines

that are lines of curvature, and their arc-length parametrization.

Penalizing the deviation of parameter lines from constant speed parametrization can be formulated as minimizing

$$\int |\mathbf{q}_u \cdot \mathbf{q}_{uu}| + |\mathbf{q}_v \cdot \mathbf{q}_{vv}| dA$$

since if $\mathbf{q}_u \cdot \mathbf{q}_u \approx c$ holds for some $c \in \mathbb{R}^+$, then differentiating by u and dividing by the multiplier of 2, one gets

$$\mathbf{q}_u \cdot \mathbf{q}_{uu} \approx 0$$
.

Minimizing the deviation of parameter lines from lines of curvature can be achieved by maximizing

$$\int \left| \mathbf{t}_{1}^{\mathbf{q}} \cdot \frac{\mathbf{q}_{u}}{|\mathbf{q}_{u}|} \right| + \left| \mathbf{t}_{2}^{\mathbf{q}} \cdot \frac{\mathbf{q}_{v}}{|\mathbf{q}_{v}|} \right| dA$$

Algorithmically, minimizing an f functional that is subject to geometric Hermite reconstruction constraints can be carried out by finding parametrization scalars $s_u, s_v, ...$ (or partial derivative x, y Darboux coordinates $x_u, y_u, ...$) such that the resulting set of affected control data \mathbf{q} amounts to a surface $\mathbf{q}(u, v)$ that is optimal with respect to the functional in question.

If exact reconstruction is not guaranteed, any approximation technique presented in subsection 3.4.2 can be used to find a best approximation to the GH reconstruction using the prescribed $s_u, s_v, ...$ (or $x_u, x_v, ...$) parametrization values.

This, again, amounts to a two-level optimization:

- find a best approximation to the geometric Hermite reconstruction problem for set s_u, s_v, \dots values
- find best s_u, s_v, \dots parametrization degrees of freedom that minimizes the given global functional f

Parametrization optimization can be used to decrease the algebraic degree of the interpolant surface. Morken has shown an example of a biquadratic surface patch that is a fourth order approximation to a sphere in [34] - this biquadratic surface was actually the osculating paraboloid of the sphere.

3.5 Algorithms of GH surface interpolation

This section presents methods to set the degrees of freedom present in GH interpolant surfaces. Introducing heuristics to set the values of these can serve as an initialization for iterative parametrization optimization techniques.

The source of these degrees of freedom is twofold: on the one hand, they arise due to the ability of changing parametrization without affecting geometric reconstruction, while on the other hand, usually there are a number of unaffected control data in the representation of the surface patches that are not subject to any geometric reconstruction constraint, hence they can be assigned to arbitrary values.

Direct methods, in the same vein as curve methods presented in Section 2.7, are not readily available because of the latter type of degrees of freedom: matrix inversion does not specify the entire control data-set of an interpolant. Parametrization optimization methods presented in the previous section need to be used to set the unaffected control data.

For the sake of simplicity, we only consider the problem of four corner GH reconstruction with quadrilateral patches. The arguments presented below can be easily extended to triangular domains as well.

Blending function-based methods presented in the next point allow us to combine a wide-range of one-point GH interpolants to form a solution surface to the four corner problem by blending these surfaces.

The *paraboloid-based* methods I propose next creates integral polynomial Bézier interpolants. They work on a similar principle as the blending-based methods, however, they offer a lower degree solution.

3.5.1 Blending-based methods

Blending-based methods use one-point exact reconstruction interpolants to GH data tuples, then use geometric blending functions to create a solution surface to the four corner reconstruction problem.

Let us suppose that $\mathbf{p}^{(ij)}(u^{(ij)}, v^{(ij)})$ are reconstructing $\mathbf{E}^{(ij)}, i, j = 0, 1$ at $(u^{(ij)}, v^{(ij)}) = (0, 0).$

Using geometric blending functions presented in Section 2.7.2 allow us to



Figure 3.8: Four second order osculating paraboloids being blended by Hartmann's G^2 parametric blending function.

formulate quadrilateral solutions in the form of

$$\mathbf{p}(u,v) := \sum_{i=0}^{1} \sum_{j=0}^{1} f^{(ij)}(u,v) \tilde{\mathbf{p}}^{(ij)}(u,v),$$

where $f^{(ij)}(u, v) : \mathbb{R}^2 \to \mathbb{R}$ are G^n blending functions. Here, the local one-point interpolants $\mathbf{p}^{(ij)}(u^{(ij)}, v^{(ij)})$ are reparameterized such that over the common parameter domain (u, v), one-point interpolant $\tilde{\mathbf{p}}^{(ij)}(u, v)$ carries out reconstruction at (u, v) = (i, j).

One can use single-variable blending functions to create the above $f^{(ij)}$ functions. Let us suppose that $f(t) : \mathbb{R} \to \mathbb{R}$ is a G^n blending function and let

$$f^{(ij)}(u,v) = f(\delta_{i,1} + (-1)^{\delta_{i,1}}u)f(\delta_{j,1} + (-1)^{\delta_{j,1}}v) .$$



Figure 3.9: Blending osculating paraboloids. The size of the subdomain of the blended one-point interpolant is illustrated by the red and blue lines. These lines are determined by projecting the neighboring GH data positions onto the tangent plane of the osculating paraboloid on the bottom-left. On the left these projections were used directly, while on the right the length of these porjections were divided by ten.

These functions satisfy the multivariable blending function properties [17]

$$f^{(ij)}(k,l) = \delta_{ik}\delta_{jl}$$
$$\partial_{u^s v^t} f^{(ij)}(k,l) = 0$$
$$\partial_{v^s v^t} f^{(ij)}(k,l) = 0$$

where i, j, k, l = 0, 1 and m = 1, 2, ..., s + t = m.

If we reparametrize the one-point interpolants such that we map the $[0, 1] \times [0, 1]$ portion of their domain onto the global parametric domain [0, 1] using appropriate flips and mirroring, one gets the following form of a parametric blending-based surface:

$$\mathbf{p}(u,v) = \sum_{i=0}^{1} \sum_{j=0}^{1} f(\delta_{i,1} + (-1)^{\delta_{i,1}} u) f(\delta_{j,1} + (-1)^{\delta_{j,1}} v) \mathbf{p}^{(ij)}(-\delta_{i,1} + u, -\delta_{j,1} + v)$$

Naturally, given a one-point interpolant, an arbitrary portion of its domain can be mapped onto the global parameter domain. The size of this mapped subdomain also affects the overall shape of the blended interpolant, and it can be considered as a form of weighing. Figure 3.9 illustrates this. The choice of blending functions also changes the shape of the blended interpolant surface and also affects the distribution of parameter lines along the surface.

In design, these one point interpolants can surve as design *control shapes* to define the geometry of parametric surfaces.

3.5.2 Paraboloid-based methods

These methods are direct counterparts of the parabola-based methods presented for GH interpolation of curves in Section 2.7.2. Compared to the surfaces presented in the previous section, paraboloid-based methods simply specify the type of one-point intrepolant surfaces, as well as the blending function that combines them, into a single solution surface.

One-point GH interpolant paraboloids are used to achieve exact reconstruction at parametric corners. Portions of these paraboloids are then selected and converted into Bézier form. After degree elevation and combination of the appropriate subsets of their control polyhedra, the control points of the blended interpolant surface are created.

3.5.2.1 Second order GH base-paraboloids

Let us use the following notation for a given G^2 base point data tuple

$$\mathbf{D} = (\mathbf{p}, \mathbf{n}, \mathbf{t}_1, \mathbf{t}_2, \kappa_1, \kappa_2),$$

where $\mathbf{p} \in \mathbb{E}^3$ denotes a point in the Euclidean space, $\mathbf{m} \in \mathbb{R}^3$ is a surface normal, $\kappa_1, \kappa_2 \in \mathbb{R}$ are principal curvature values and $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{R}^3$ are corresponding principal directions, $|\mathbf{m}| = |\mathbf{t}_1| = |\mathbf{t}_2| = 1$, and κ_1 is the minimum and κ_2 is the maximum normal curvature. Without loss of generality, the orthonormal basis $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$ is assumed to be right-handed. We also assume that the normals are oriented consistently.

Let there be given four

$$\mathbf{D}^{(ij)} = (\mathbf{p}^{(ij)}, \mathbf{m}^{(ij)}, \mathbf{t}_1^{(ij)}, \mathbf{t}_2^{(ij)}, \kappa_1^{(ij)}, \kappa_2^{(ij)}), \ i, j = 0, 1$$

 G^2 base-point data tuples and let us find a

$$\mathbf{b}(u,v) = \sum_{j=0}^{m} \sum_{i=0}^{n} \mathbf{b}_{ij} B_i^n(u) B_j^m(v),$$

degree (n, m) quadrilateral Bézier surface, $n, m \geq 2$, $(u, v) \in [0, 1]^2$, $\mathbf{b}_{ij} \in \mathbb{E}^3, i = 0, ..., n, j = 0, ..., m$, that reconstructs the $\mathbf{D}^{(ij)}$ base point data at its parametric corners (u, v) = (i, j), i, j = 0, 1.

Now, let us consider the problem of constructing C^2 spline surfaces using bi-quintic four-corner second order GH interpolants, provided the base points constitute of a regular rectangular grid.

First, the remaining degrees of freedom of the control net need to be set. In order to do that, let us assign the *base paraboloids*

$$\mathbf{p}^{(ij)}(u,v) = \mathbf{p}^{(ij)} + [\mathbf{t}_1^{(ij)}, \mathbf{t}_2^{(ij)}, \mathbf{m}^{(ij)}] \begin{bmatrix} u \\ v \\ \frac{\kappa_1^{(ij)}}{2}u^2 + \frac{\kappa_2^{(ij)}}{2}v^2 \end{bmatrix}$$
(3.69)

to each $\mathbf{D}^{(ij)}, i, j = 0, 1.$

It is easily seen that the unit surface normal of a paraboloid in (3.69) at (u, v) = (0, 0) is $\mathbf{m}^{(ij)}$ and its principal curvatures and principal directions are $\kappa_1^{(ij)}, \kappa_2^{(ij)}, \mathbf{t}_1^{(ij)}, \mathbf{t}_2^{(ij)}$. See Appendix E for a formal proof.

Let us consider the corner (i, j) = (0, 0) and let us drop the upper indices, i.e. let $\mathbf{D} = \mathbf{D}^{(00)}, \mathbf{m}^{(00)} = \mathbf{m}, \mathbf{p}(u, v) = \mathbf{p}^{(00)}(u, v)$ and so on.

Let $\mathbf{u} = (a_x, a_y), \mathbf{b} = (b_x, b_y)$ be two points in the domain of the paraboloid $\mathbf{p}(u, v)$, and let $\mathbf{g}_{ij}, i, j = 0, 1, 2$ denote the control points of the bi-quadratic Bézier patch that represents the mapping of the $\mathbf{0}, \mathbf{a}, \mathbf{a} + \mathbf{b}, \mathbf{b}$ quadrilateral of $dom(\mathbf{p}(u, v))$, that is, let

$$\mathbf{g}(u,v) = \mathbf{p}(u \cdot a_x + v \cdot b_x, u \cdot a_y + v \cdot b_y),$$

where $u, v \in [0, 1]$ and $\mathbf{g}(u, v) = \sum_{j=0}^{2} \sum_{i=0}^{2} \mathbf{g}_{ij} B_i^2(u) B_j^2(v)$.

To specify the position of the \mathbf{b}_{ij} , i, j = 0, 1, 2 control points, let us elevate the degree of $\mathbf{g}(u, v)$ to five in both the u and v directions and assign the 3×3 control points of the resulting control net around (u, v) = (0, 0) to $\mathbf{b}_{ij}, i, j = 0, 1, 2$. These 9 control points are expressed explicitly in terms of the power basis coefficients of the given paraboloid portion as follows: let the power basis coefficients be

$$\mathbf{a}_{00} = \mathbf{a}_{21} = \mathbf{a}_{12} = \mathbf{a}_{22} = \mathbf{0},$$
$$\mathbf{a}_{10} = \begin{bmatrix} a_x \\ a_y \\ 0 \end{bmatrix}, \ \mathbf{a}_{01} = \begin{bmatrix} b_x \\ b_y \\ 0 \end{bmatrix},$$
$$\mathbf{a}_{20} = \begin{bmatrix} 0 \\ 0 \\ \frac{\kappa_1}{2}a_x^2 + \frac{\kappa_2}{2}a_y^2 \end{bmatrix}, \ \mathbf{a}_{02} = \begin{bmatrix} 0 \\ 0 \\ \frac{\kappa_1}{2}b_x^2 + \frac{\kappa_2}{2}b_y^2 \end{bmatrix},$$
$$\mathbf{a}_{11} = \begin{bmatrix} 0 \\ 0 \\ \kappa_1a_xb_x + \kappa_2a_yb_y \end{bmatrix}$$

Then the coordinates of the control points in the $(\mathbf{p}; \mathbf{t}_1, \mathbf{t}_2, \mathbf{n})$ basis are computed as

$$\mathbf{b}_{00} = \mathbf{a}_{00} \tag{3.70}$$

$$\mathbf{b}_{10} = \frac{\mathbf{a}_{10}}{5} + \mathbf{a}_{00} \tag{3.71}$$

$$\mathbf{b}_{20} = \frac{\mathbf{a}_{20} + 4\mathbf{a}_{10} + 10\mathbf{a}_{00}}{10} \tag{3.72}$$

$$\mathbf{b}_{01} = \frac{\mathbf{a}_{01} + 5\mathbf{a}_{00}}{5} \tag{3.73}$$

$$\mathbf{b}_{11} = \frac{\mathbf{a}_{11} + 5\mathbf{a}_{10} + 5\mathbf{a}_{01} + 25\mathbf{a}_{00}}{25} \tag{3.74}$$

$$\mathbf{b}_{21} = \frac{5\mathbf{a}_{20} + 4\mathbf{a}_{11} + 20\mathbf{a}_{10} + 10\mathbf{a}_{01} + 50\mathbf{a}_{00}}{50} \tag{3.75}$$

$$\mathbf{b}_{02} = \frac{\mathbf{a}_{02} + 4\mathbf{a}_{01} + 10\mathbf{a}_{00}}{10} \tag{3.76}$$

$$\mathbf{b}_{12} = \frac{4\mathbf{a}_{11} + 10\mathbf{a}_{10} + 5\mathbf{a}_{02} + 20\mathbf{a}_{01} + 50\mathbf{a}_{00}}{50} \tag{3.77}$$

$$\mathbf{b}_{22} = \frac{5\mathbf{a}_{20} + 8\mathbf{a}_{11} + 20\mathbf{a}_{10} + 5\mathbf{a}_{02} + 20\mathbf{a}_{01} + 50\mathbf{a}_{00}}{50} \tag{3.78}$$

If the angle between $a_x \mathbf{t}_1 + a_y \mathbf{t}_2$ and $b_x \mathbf{t}_1 + b_y \mathbf{t}_2$ is smaller than 180 degrees,

direct evaluation of the differences prove that the choice of control points satisfies the conditions of proposition 10, i.e at (u, v) = (0, 0) base point quantities of $\mathbf{D}^{(00)}$ are reconstructed.

After handling the remaining three corners as above, 4×9 control points are generated, which together form the control net of a bi-quintic Bézier patch that satisfies the conditions of the four-corner geometric Hermite reconstruction of $\mathbf{D}^{(ij)}, i, j = 0, 1.$

The vectors $a_x^{(ij)} \mathbf{t}_1^{(ij)} + a_y^{(ij)} \mathbf{t}_2^{(ij)}$ and $b_x^{(ij)} \mathbf{t}_1^{(ij)} + b_y^{(ij)} \mathbf{t}_2^{(ij)}$ will be the tangent vectors of the boundary curves in the u and v parametric directions at (u, v) = (i, j), i, j = 0, 1. Let us refer to these vectors as *base tangents*.

In order to handle all the four corners correctly, we use the following power basis coefficients:

$$\mathbf{a}_{00}^{(ij)} = \mathbf{a}_{21}^{(ij)} = \mathbf{a}_{12}^{(ij)} = \mathbf{a}_{22}^{(ij)} = \mathbf{0},$$
(3.79)

$$\mathbf{a}_{10}^{(ij)} = \begin{bmatrix} (-1)^{i} a_{x} \\ (-1)^{i} a_{y} \\ 0 \end{bmatrix}, \ \mathbf{a}_{01}^{(ij)} = \begin{bmatrix} (-1)^{j} b_{x} \\ (-1)^{j} b_{y} \\ 0 \end{bmatrix},$$
(3.80)

$$\mathbf{a}_{20}^{(ij)} = \begin{bmatrix} 0\\ 0\\ \frac{\kappa_1}{2}a_x^2 + \frac{\kappa_2}{2}a_y^2 \end{bmatrix}, \ \mathbf{a}_{02}^{(ij)} = \begin{bmatrix} 0\\ 0\\ \frac{\kappa_1}{2}b_x^2 + \frac{\kappa_2}{2}b_y^2 \end{bmatrix},$$
(3.81)

$$\mathbf{a}_{11}^{(ij)} = \begin{bmatrix} 0 \\ 0 \\ (-1)^{i+j} (\kappa_1 a_x b_x + \kappa_2 a_y b_y) \end{bmatrix}$$
(3.82)

3.5.2.2 Continuous connection along boundaries

Let us suppose our data are given in a uniform rectangular grid. C^2 continuity at the corners of Bézier patches follows from the construction. Now, let us consider $\mathbf{b}(u, v)$ and $\mathbf{c}(u, v)$, two bi-quintic four-corner GH interpolants of basepoint data tuples $\mathbf{D}^{(ij)}$ and $\mathbf{E}^{(ij)}$, i, j = 0, 1, with control points \mathbf{b}_{ij} , \mathbf{c}_{ij} , i, j = 0, ..., 5 respectively. See figure 3.10.

Let $\mathbf{D}^{(1j)} = \mathbf{E}^{(0j)}$, j = 0, 1 and let us examine how $\mathbf{b}(u, v)$ and $\mathbf{c}(u, v)$ can be connected with second order parametric continuity along the *u* parametric direction.



Figure 3.10: Connecting second order GH interpolants

 C^2 connection along the $\mathbf{b}(1, v) = \mathbf{c}(0, v), v \in [0, 1]$ boundary requires that [11]

$$\Delta^{10}\mathbf{b}_{4j} = \Delta^{10}\mathbf{c}_{0j} \tag{3.83}$$

$$\Delta^{20}\mathbf{b}_{3j} = \Delta^{20}\mathbf{c}_{0j} \tag{3.84}$$

hold for j = 0, ..., 5. This allows us to investigate the continuity condition independently on the two control net portions around the two base point data tuples.

Let us consider the case of $\mathbf{D}^{(10)} = \mathbf{E}^{(00)}$. From (3.70)-(3.78) it follows that $\mathbf{c}_{i,j}, i, j = 0, 1, 2$ depend on the selection of the base tangent vectors $\mathbf{a}^{(00)}, \mathbf{b}^{(00)}$.

Similarly, $\mathbf{b}_{3+i,j}$, i, j = 0, 1, 2 depend on the base tangent vectors $\mathbf{c}^{(10)}$, $\mathbf{d}^{(10)}$, that is, the tangents vectors used for the construction of the \mathbf{b}_{ij} control net.

Since both control net portions share the same base paraboloid, assigned to $\mathbf{D}^{(10)} = \mathbf{E}^{(00)}$, and taking into account the appropriate handling of base tangent directions around this corner, conditions (3.83)-(3.84) can be satisfied by using the same base tangent vectors for the *u* and *v* parametric directions, i.e. if $\mathbf{a}^{(00)} = \mathbf{c}^{(10)} = \mathbf{a}, \mathbf{b}^{(00)} = \mathbf{d}^{(10)} = \mathbf{b}$. See figure 3.10.

The control points around the corner $\mathbf{D}^{(11)} = \mathbf{E}^{(01)}$ are handled analogously. Continuity along the v parametric directions is derived similarly.



Figure 3.11: Connection of two GH bi-quintic interpolants along their common boundary. The length of base tangent vectors are twice as long on the image on the right compared to the ones on the left.

3.5.2.3 Extension to higher order GH interpolation

Using paraboloids that are solutions to third or higher order GH surface interpolation allows us to extend the paraboloid-based approach presented here to create quadrilateral C^n spline surfaces from geometric data. This C^n parametrization then can be used in parametrization optimization as a initial value for the parametrization degrees of freedom.

Appendix E details how a third order GH interpolant paraboloid can be constructed, as well as verifies that the paraboloid used previously indeed does satisfy second order GH interpolation constraints. The techniques shown in this appendix can be extended to higher order GH interpolation.

Chapter 4

Summary

In this thesis I presented a general formulation of geometric Hermite interpolation of curves and surfaces in terms of geometric reconstruction equations. These reconstruction constraints were cast upon the positions and derivatives of curves and surfaces at prescribed parameter values.

This allowed the separation of geometric constraints from degrees of freedom of parametrization in the control data of an arbitrary parametric representation.

In the case of curves, this formulation relied on the expression of curve derivatives in the Frenet frame, using a simple recurrence formula.

I also presented the use of Frenet coordinates to compute geometric invariants – and their derivatives with respect to arc-length –, and I derived formulae to verify G^3 geometric continuity of two joining curves.

Solvability of the reconstruction equations was discussed for arbitrary but fixed parametrization next, a setting in which the problem becomes linear. I derived a general existence theorem for exact reconstruction, irrespective of the basis of the curve. For polynomial curves, I gave upper bounds on the degree of interpolants. I investigated the L^1 and L^2 best approximation to a given reconstruction problem. I showed how both symmetric and one-point partial exact reconstruction constraints can be incorporated into approximation.

By letting the coefficients of parametrization in the reconstruction equations vary, I used nonlinear optimization techniques to find optimal parametrizations, which may be in the sense of some real valued functional or even the degree reduction of a polynomial.

I presented algorithms to carry out geometric Hermite interpolation of

curves and I characterized the problem of third order GH interpolation.

I showed how geometric properties of surfaces at a non-umbilical point can be specified by geometric invariants of lines of curvature. I made a correspondence between the geometric continuity of surfaces and the differential geometric properties of lines of curvature of two joining surfaces. Finally, I used these results to formulate geometric Hermite interpolation of surfaces, which stipulates geometric reconstruction equations on partial derivatives.

Exact and approximate solutions to this problem were discussed in accordance with the case of curves, and analogous results were shown regarding the conditions of exact reconstruction and the upper-bound on the degree of polynomial interpolants. nonlinear parameter optimization methods were presented.

The abundance of degrees of freedom in GH interpolation of surfaces requires heuristics or the aforementioned parametrization optimization techniques to fully define the interpolant patch. Blending based and direct methods were presented to create such patches.

Acknowledgement

I would like to express my gratitude to my advisor, János Vida, for his ideas and countless consultations throughout my research.

I would also like to thank Gábor Renner and Levente Hajder for sharing their opinions and giving me feedback on the final manuscripts of this thesis, and Andrea Szabó for her help during the finishing touches on this dissertation.

Last, but most important of all, I would like to thank my family for all their support throughout these years.

Appendices

Appendix A

Proof of Theorems and Lemmas of curves

A.1 Equivalence of Frenet *x* coordinates and derivatives of arc-length functions

Lemma 21 The derivatives of the arc-length function

$$s'(t) = |\mathbf{r}'(t)| = (\mathbf{r}'(t) \cdot \mathbf{r}'(t))^{\frac{1}{2}}$$

$$s''(t) = ((\mathbf{r}'(t) \cdot \mathbf{r}'(t))^{\frac{1}{2}})' = \frac{\mathbf{r}'(t) \cdot \mathbf{r}''(t)}{|\mathbf{r}'(t)|}$$

$$s'''(t) = \left(\frac{\mathbf{r}'(t) \cdot \mathbf{r}''(t)}{|\mathbf{r}'(t)|}\right)' = \frac{\mathbf{r}''(t) \cdot \mathbf{r}''(t) + \mathbf{r}'(t) \cdot \mathbf{r}'''(t)}{|\mathbf{r}'(t)|} - \frac{(\mathbf{r}'(t) \cdot \mathbf{r}''(t))^2}{|\mathbf{r}'(t)|^3}$$

can be expressed with the Frenet-frame coordinates of the derivatives as

$$s'(t) = x_1 \tag{A.1}$$

$$s''(t) = x_2 \tag{A.2}$$

$$s'''(t) = \frac{y_2^2}{x_1} + x_3$$
(A.3)
= $\kappa^2 x_1^3 + x_3$

Proof.

By noting that $x_1 > 0$, i.e. $|x_1| = x_1$, it follows immediately that

$$s'(t) = |\mathbf{r}'(t)| = (\mathbf{r}'(t) \cdot \mathbf{r}'(t))^{\frac{1}{2}}$$
$$= \left(\begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix} \right)^{\frac{1}{2}} = |x_1|$$
$$= x_1$$

The second derivative of the arc-length function is

$$s''(t) = \left(\left(\mathbf{r}'(t) \cdot \mathbf{r}'(t) \right)^{\frac{1}{2}} \right)' = \frac{\mathbf{r}'(t) \cdot \mathbf{r}''(t)}{|\mathbf{r}'(t)|}$$
$$= \left(\begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix}_F \cdot \begin{bmatrix} x_2 \\ y_2 \\ 0 \end{bmatrix}_F \right) / x_1$$
$$= x_2$$

The third derivative is similarly

$$\begin{split} s'''(t) &= \left(\frac{\mathbf{r}'(t) \cdot \mathbf{r}''(t)}{|\mathbf{r}'(t)|}\right)' \\ &= \frac{(\mathbf{r}''(t) \cdot \mathbf{r}''(t) + \mathbf{r}'(t) \cdot \mathbf{r}'''(t))|\mathbf{r}'| - (\mathbf{r}'(t) \cdot \mathbf{r}''(t))\frac{\mathbf{r}'(t) \cdot \mathbf{r}''(t)}{|\mathbf{r}'(t)|}}{|\mathbf{r}'(t)|^2} \\ &= \frac{\mathbf{r}''(t) \cdot \mathbf{r}''(t) + \mathbf{r}'(t) \cdot \mathbf{r}'''(t)}{|\mathbf{r}'(t)|} - \frac{(\mathbf{r}'(t) \cdot \mathbf{r}''(t))^2}{|\mathbf{r}'(t)|^3} \\ &= \left(\left[\begin{pmatrix} x_2 \\ y_2 \\ 0 \end{pmatrix}_F \cdot \begin{bmatrix} x_2 \\ y_2 \\ 0 \end{bmatrix}_F + \begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix}_F \cdot \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}_F \right) / x_1 - \left(\left[\begin{pmatrix} x_1 \\ 0 \\ 0 \end{bmatrix}_F \cdot \begin{bmatrix} x_2 \\ y_2 \\ 0 \end{bmatrix}_F \right)^2 / x_1^3 \\ &= \frac{x_2^2 + y_2^2 + x_1 x_3}{x_1} - \frac{x_1^2 x_2^2}{x_1^3} = \frac{x_2^2}{x_1} + \frac{y_2^2}{x_1} + x_3 - \frac{x_2^2}{x_1} \\ &= \frac{y_2^2}{x_1} + x_3 \,. \end{split}$$

A.2 Expression of geometric invariants with Frenet coordinates

Lemma 22 Geometric invariants of a curve, up to order three, can be obtained using Frenet coordinates as

$$\kappa = \frac{y_2}{x_1^2} \tag{A.4}$$

$$\hat{\kappa}' \circ s = \frac{y_3 - 3x_1 x_2 \kappa}{x_1^3} \tag{A.5}$$

$$=\frac{y_3x_1 - 3x_2y_2}{x_1^4} \tag{A.6}$$

$$\tau = \frac{z_3}{\kappa x_1^3} \tag{A.7}$$

$$=\frac{z_3}{x_1y_2}\tag{A.8}$$

Proof. It is the direct result from substituting (2.20)-(2.22) into (2.16), and expressing the geometric invariants.

For curvature, this gives from

$$\mathbf{r}'' = \begin{bmatrix} s''\\(s')^2\kappa\\0 \end{bmatrix}_F = \begin{bmatrix} x_2\\x_1^2\kappa\\0 \end{bmatrix}$$

the expression

 $y_2 = x_1^2 \kappa \; ,$

that is

$$\kappa = \frac{y_2}{x_1^2}$$

For torsion and arc-length derivative of curvature, using

$$\begin{bmatrix} s''' - (s')^3 \kappa^2 \\ 3s's''\kappa + (s')^3 \cdot \hat{\kappa}' \circ s \\ (s')^3 \kappa \tau \end{bmatrix}_F = \begin{bmatrix} x_3 - x_1^3 \kappa^2 \\ 3x_1 x_2 \kappa + x_1^3 \cdot \hat{\kappa}' \circ s \\ x_1^3 \kappa \tau \end{bmatrix}_F$$

we get

$$y_3 = 3x_1x_2\kappa + x_1^3 \cdot \hat{\kappa}' \circ s$$
$$z_3 = x_1^3 \kappa \tau$$

that is

$$\begin{aligned} \hat{\kappa}' \circ s &= \frac{y_3 - 3x_1 x_2 \kappa}{x_1^3} = \frac{y_3 x_1 - 3x_2 y_2}{x_1^4} \ ,\\ \tau &= \frac{z_3}{\kappa x_1^3} = \frac{z_3}{x_1 y_2} \ . \end{aligned}$$

A.3 Conditions of geometric continuity in terms of Frenet coordinates

Lemma 23 Let us assume that the Frenet frames of $\mathbf{r}(t)$ and $\mathbf{s}(t)$ coincide at $\mathbf{r}(1) = \mathbf{s}(0)$, and let $[x_i, y_i, z_i]_F^T$ denote the Frenet coordinates of $\mathbf{r}^{(i)}(1)$, and $[\widetilde{x}_i, \widetilde{y}_i, \widetilde{z}_i]_F^T$ that of $\mathbf{s}^{(i)}(0)$. The two curves are G^2 iff

$$\frac{y_2}{\widetilde{y}_2} = \left(\frac{x_1}{\widetilde{x}_1}\right)^2$$

Furthermore, two G^2 curves are G^3 iff

$$\frac{\kappa\tau x_1^3}{\widetilde{\kappa}\widetilde{\tau}\widetilde{x}_1^3} = \left(\frac{x_1}{\widetilde{x}_1}\right)^3$$
$$\frac{y_3 - 3x_1x_2\kappa}{\widetilde{y}_3 - 3\widetilde{x}_1\widetilde{x}_2\widetilde{\kappa}} = \left(\frac{x_1}{\widetilde{x}_1}\right)^3$$

both hold.

Proof. By definition, $\mathbf{r}(t)$ and $\mathbf{s}(t)$ are G^2 iff they are G^1 and the $\kappa, \tilde{\kappa}$ curvatures are equal. It follows from (2.16) that

$$y_2 = x_1^2 \kappa$$
, $\widetilde{y}_2 = \widetilde{x}_1^2 \widetilde{\kappa}$,

and, the ratio of the normal coordinates is

$$\frac{y_2}{\widetilde{y}_2} = \frac{x_1^2 \kappa}{\widetilde{x}_1^2 \widetilde{\kappa}} = \frac{x_1^2}{\widetilde{x}_1^2} \ ,$$

iff $\kappa = \tilde{\kappa}$. That is, the curvatures agree iff

$$\frac{y_2}{\widetilde{y}_2} = \left(\frac{x_1}{\widetilde{x}_1}\right)^2$$

The two curves $\mathbf{r}(t)$ and $\mathbf{s}(t)$ are G^3 iff they are G^1 and G^2 , and both the torsions and arc-length derivatives of curvature agree.

The equality of $\tau = \tilde{\tau}$ is equivalent to

$$\frac{z_3}{\widetilde{z}_3} = \left(\frac{x_1}{\widetilde{x}_1}\right)^3 \;,$$

following a similar reasoning to that of the curvature: it follows from (2.16) that

$$z_3 = \kappa \tau x_1^3, \ \widetilde{z}_3 = \widetilde{\kappa} \widetilde{\tau} \widetilde{x}_1^3,$$

so the ratio of

$$\frac{z_3}{\widetilde{z}_3} = \frac{\kappa \tau x_1^3}{\widetilde{\kappa} \widetilde{\tau} \widetilde{x}_1^3} = \left(\frac{x_1}{\widetilde{x}_1}\right)^3$$

holds iff the torsions agree, since the two curves are already G^2 .

To find a similar ratio-based condition for the equality of arc-length derivatives of the curvature, one has to express $\hat{\kappa}'$ from

$$3x_1x_2\kappa + x_1^3 \cdot \hat{\kappa}' \circ \hat{s} \; ;$$

yielding that, for two G^2 curves, $\hat{\kappa}' \circ \hat{s} = \tilde{\kappa}' \circ \tilde{s}$ holds iff

$$\frac{y_3 - 3x_1x_2\kappa}{\widetilde{y}_3 - 3\widetilde{x}_1\widetilde{x}_2\widetilde{\kappa}} = \left(\frac{x_1}{\widetilde{x}_1}\right)^3 .$$

Appendix B

Proof of parabolic GH interpolant reconstruction

In this appendix we show that given a third order GH data tuple \mathbf{D}_i , the polynomials in Table B.1 reconstruct \mathbf{D}_i at t = 0.

Let us restate the recurrence formula of Theorem 2

$$\mathbf{r}^{(n+1)} = \begin{bmatrix} x'_n \\ y'_n \\ z'_n \end{bmatrix}_F + s' \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}_F , \qquad (B.1)$$

which can be expressed by coordinates as

$$x_{n+1} = x'_n - s' \kappa y_n$$
$$y_{n+1} = y'_n + s' \kappa x_n - \tau z_n$$
$$z_{n+1} = z'_n + s' \tau y_n$$

The first, second, and third derivatives are

$$\mathbf{r}' = s' \begin{bmatrix} 1\\0\\0 \end{bmatrix}_F, \ \mathbf{r}'' = \begin{bmatrix} s''\\(s')^2\kappa\\0 \end{bmatrix}_F, \quad \mathbf{r}''' = \begin{bmatrix} s''' - (s')^3\kappa^2\\3s's''\kappa + (s')^3\hat{\kappa}'\\(s')^3\kappa\tau \end{bmatrix}_F.$$

Let us prove that the parabolas reconstruct \mathbf{D}_i :

• $\kappa \neq 0$: it follows immediately from Theorem 2

κ	$\hat{\kappa}'$	τ	polynomial curve					
≠ 0	$\neq 0$	$\in \mathbb{R}$	$\begin{bmatrix} t \\ \frac{\kappa_i}{2}t^2 + \frac{\hat{\kappa}'_i}{6}t^3 \\ \frac{\kappa_i\tau_i}{6}t^3 \end{bmatrix}$					
0	$\in \mathbb{R}$	0	$\begin{bmatrix} t \\ \frac{\hat{\kappa}'_i}{6} t^3 \\ 0 \end{bmatrix}$					
0	$\neq 0$	$\in \mathbb{R}$	$\begin{bmatrix}t\\\frac{\hat{\kappa}'_i}{6}t^3+\frac{\hat{\kappa}''_i}{24}t^4\\\frac{\hat{\kappa}'_i\tau_i}{12}t^4\end{bmatrix}$					
0	0	$\in \mathbb{R}$	$\begin{bmatrix} t \\ \frac{\hat{\kappa}''_{i}}{24}t^{4} + \frac{\hat{\kappa}''_{i}}{120}t^{5} \\ \frac{\hat{\kappa}''_{i}\tau_{i}}{40}t^{5} \end{bmatrix}$					

Table B.1: Polynomial curves reconstructing a third order GH data tuple at t = 0.

• $\kappa = 0, \hat{\kappa}' \neq 0, \tau \in \mathbb{R}$:

In this case $\mathbf{r}', \mathbf{r}''$ are linearly dependent, that is, the second derivative of the curve does not deviate from the tangent line. As a result, the osculating plane cannot be spanned by $\mathbf{r}', \mathbf{r}''$, thus the Frenet frame is not yet defined, because $\mathbf{r}' || \mathbf{r}'' \to \mathbf{r}' \times \mathbf{r}'' = \mathbf{0}$.

However, since $\hat{\kappa}' \neq 0$, the third derivative of the curve is

$$\mathbf{r}^{\prime\prime\prime} = \begin{bmatrix} s^{\prime\prime\prime} \\ (s^{\prime})^3 \hat{\kappa}^{\prime} \\ 0 \end{bmatrix}_F$$

which – due to the regularity of the curve, that is, because $s' \neq 0$ – can be used to specify the osculating plane, and the Frenet frame as

$$\mathbf{t} = \frac{\mathbf{r}'}{|\mathbf{r}'|}$$
$$\mathbf{n} = \mathbf{b} \times \mathbf{t}$$
$$\mathbf{b} = \frac{\mathbf{r}' \times \mathbf{r}'''}{|\mathbf{r}' \times \mathbf{r}'''|}$$

Note that the torsion is still not present in \mathbf{r}''' , so the fourth derivative has to be computed, which, in general, has the form of

$$\mathbf{r}^{(4)} = \begin{bmatrix} s^{(4)} - 3(s')^2 s'' \kappa^2 - 2(s')^4 \kappa \hat{\kappa}' - 3(s')^2 s'' \kappa^2 - (s')^4 \hat{\kappa}' \kappa \\ 3(s'')^2 \kappa + 3s' s''' \kappa + 6(s')^2 s'' \hat{\kappa}' + (s')^4 \hat{\kappa}'' + \kappa s' s''' - \kappa^3 (s')^4 - \tau^2 \kappa (s')^4 \\ 6(s')^2 s'' \kappa \tau + 2(s')^4 \hat{\kappa}' \tau + (s')^4 \kappa \hat{\tau}' \end{bmatrix}_F$$

Substituting $\kappa = 0$ into this, the fourth derivative becomes

$$\mathbf{r}^{(4)} = \begin{bmatrix} s^{(4)} \\ 6(s')^2 s'' \hat{\kappa}' + (s')^4 \hat{\kappa}'' \\ 2(s')^4 \hat{\kappa}' \tau \end{bmatrix}_F$$

Let us now look for a third order GH interpolant polynomial in the form of

$$\mathbf{p}(t) = \begin{bmatrix} a_1 t \\ \sum_{i=2}^4 b_i t^i \\ \sum_{i=3}^4 c_i t^i \end{bmatrix} = \begin{bmatrix} t \\ b_2 t^2 + b_3 t^3 + b_4 t^4 \\ c_3 t^3 + c_4 t^4 \end{bmatrix}$$

Its derivatives at t = 0 are

$$\mathbf{p}' = \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix}, \ \mathbf{p}'' = \begin{bmatrix} 0 \\ 2b_2 \\ 0 \end{bmatrix}, \ \mathbf{p}''' = \begin{bmatrix} 0 \\ 6b_3 \\ 6c_3 \end{bmatrix}, \ \mathbf{p}'''' = \begin{bmatrix} 0 \\ 24b_4 \\ 24c_4 \end{bmatrix}$$
which should be equal to

$$\mathbf{p}' = \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} s' \\ 0 \\ 0 \end{bmatrix}_F$$
$$\mathbf{p}'' = \begin{bmatrix} 0 \\ 2a_2 \\ 0 \end{bmatrix} = \begin{bmatrix} s'' \\ (s')^2 \kappa \\ 0 \end{bmatrix}_F$$
$$\mathbf{p}''' = \begin{bmatrix} 0 \\ 6b_3 \\ 6c_3 \end{bmatrix} = \begin{bmatrix} s''' - (s')^3 \kappa^2 \\ 3s's''\kappa + (s')^3 \hat{\kappa}' \\ (s')^3 \kappa \tau \end{bmatrix}_F$$
$$\mathbf{p}'''' = \begin{bmatrix} 0 \\ 24b_4 \\ 24c_4 \end{bmatrix} = \begin{bmatrix} s^{(4)} \\ 6(s')^2 s'' \hat{\kappa}' + (s')^4 \hat{\kappa}'' \\ 2(s')^4 \hat{\kappa}' \tau \end{bmatrix}_F$$

because of Theorem 2.

The arc-length function derivatives of $\mathbf{p}(t)$ at t = 0 are

$$s'' = 1$$

$$s'' = \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 2a_2 \\ 0 \end{bmatrix} / |a_1| = 0$$

$$s''' = \frac{\begin{bmatrix} 0 \\ 2a_2 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 2a_2 \\ 0 \end{bmatrix} + \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 6b_3 \\ 6c_3 \end{bmatrix} - \frac{\left(\begin{bmatrix} 0 \\ 2a_2 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix} \right)^2}{|a_1|^3}$$

$$= \frac{4a_2^2}{|a_1|}$$

where the fourth derivative of s(t) is not necessary, since it can be considered as a degree of freedom constrained by the formulation of $\mathbf{p}(t)$.

The coefficients of the polynomial are then

$$a_{1} = 1 = s'$$

$$b_{2} = \frac{\kappa}{2} = \frac{(s')^{2}\kappa}{2}$$

$$b_{3} = \frac{\hat{\kappa}'}{6} = \frac{(s')^{3}\hat{\kappa}'}{6}$$

$$c_{3} = \frac{\kappa\tau}{6} = \frac{(s')^{4}\kappa\tau}{6}$$

$$b_{4} = \frac{\hat{\kappa}''}{24} = \frac{(s')^{4}\hat{\kappa}''}{24}$$

$$c_{4} = \frac{2\hat{\kappa}'\tau}{24} = \frac{(s')^{4}\hat{\kappa}'\tau}{12}$$

which satisfy the arc-length derivative conditions of $s''' = (s')^3 \kappa^2$, etc. as well.

Thus, the following quartic polynomial reconstructs \mathbf{D}_i at t = 0:

$$\mathbf{p}(t) = \begin{bmatrix} t\\ \frac{\hat{\kappa}'}{6}t^3 + \frac{\hat{\kappa}''}{24}t^4\\ \frac{\hat{\kappa}'\tau}{12}t^4 \end{bmatrix}$$
(B.2)

• $\kappa = \hat{\kappa} = 0, \tau \in \mathbb{R}$

In this case the third and fourth derivatives of $\mathbf{r}(t)$ are

$$\mathbf{r}^{\prime\prime\prime} = \begin{bmatrix} s^{\prime\prime\prime} \\ 0 \\ 0 \end{bmatrix}_{F}$$
$$\mathbf{r}^{(4)} = \begin{bmatrix} s^{(4)} \\ (s^{\prime})^{4} \hat{\kappa}^{\prime\prime} \\ 0 \end{bmatrix}_{F}$$

Similarly to the previous case, a higher order derivative is required to

specify the Frenet-frame, which is now $\mathbf{r}^{(4)}$, that is

$$\begin{aligned} \mathbf{t} &= \frac{\mathbf{r}'}{|\mathbf{r}'|} \\ \mathbf{n} &= \mathbf{b} \times \mathbf{t} \\ \mathbf{b} &= \frac{\mathbf{r}' \times \mathbf{r}^{(4)}}{|\mathbf{r}' \times \mathbf{r}^{(4)}|} \end{aligned}$$

In general, the Frenet coordinates of the fifth derivative are

$$\begin{aligned} x_5 &= (s')^5 \kappa^2 \tau^2 - 4 (s')^5 \kappa \hat{\kappa}'' - 3 (s')^5 (\hat{\kappa}')^2 - 30 (s')^3 s'' \kappa \hat{\kappa}' \\ &+ (s')^5 \kappa^4 + \left(-10 (s')^2 s''' - 15 s' s''^2\right) \kappa^2 + s''''' \\ y_5 &= -3 (s')^5 \kappa \tau \hat{\tau}' + \left(-3 (s')^5 \hat{\kappa}' - 10 (s')^3 s'' \kappa\right) \tau^2 + (s')^5 \hat{\kappa}''' + 10 (s')^3 s'' \hat{\kappa}'' \\ &+ \left(-6 (s')^5 \kappa^2 + 10 (s')^2 s''' + 15 s' s''^2\right) \hat{\kappa}' - 10 (s')^3 s'' \kappa^3 + (5 s' s'''' + 10 s'' s''') \kappa \\ z_5 &= (s')^5 \kappa \hat{\tau}'' + \left(3 (s')^5 \hat{\kappa}' + 10 (s')^3 s'' \kappa\right) \hat{\tau}' - (s')^5 \kappa \tau^3 \\ &+ \left(3 (s')^5 \hat{\kappa}'' + 20 (s')^3 s'' \hat{\kappa}' - (s')^5 \kappa^3 + \left(10 (s')^2 s''' + 15 s' s''^2\right) \kappa\right) \tau \end{aligned}$$

Substituting $\kappa = \hat{\kappa}' = 0$ yields

$$\mathbf{r}^{(5)} = \begin{bmatrix} s''''' \\ (s')^5 \ \hat{\kappa}''' + 10 \ (s')^3 \ s'' \ \hat{\kappa}'' \\ 3 \ (s')^5 \ \hat{\kappa}'' \tau \end{bmatrix}_F$$

Let us now look for a third order GH interpolant polynomial in the form of

$$\mathbf{p}(t) = \begin{bmatrix} a_1 t \\ \sum_{i=2}^5 b_i t^i \\ \sum_{i=3}^5 c_i t^i \end{bmatrix} = \begin{bmatrix} t \\ b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \\ c_3 t^3 + c_4 t^4 + c_5 t^5 \end{bmatrix}$$

and let us also assume that $\hat{\kappa}'' \neq 0.$ Then it follows immediately, that

$$\mathbf{p}^{\prime\prime\prime\prime\prime\prime} = \begin{bmatrix} 0\\ 120b_5\\ 120c_5 \end{bmatrix},$$

and the reconstruction conditions are expanded as

$$\mathbf{p}' = \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} s' \\ 0 \\ 0 \end{bmatrix}_F$$
$$\mathbf{p}'' = \begin{bmatrix} 0 \\ 2a_2 \\ 0 \end{bmatrix} = \begin{bmatrix} s'' \\ (s')^2 \kappa \\ 0 \end{bmatrix}_F$$
$$\mathbf{p}''' = \begin{bmatrix} 0 \\ 6b_3 \\ 6c_3 \end{bmatrix} = \begin{bmatrix} s''' - (s')^3 \kappa^2 \\ 3s's''\kappa + (s')^3 \hat{\kappa}' \\ (s')^3 \kappa \tau \end{bmatrix}_F$$
$$\mathbf{p}'''' = \begin{bmatrix} 0 \\ 24b_4 \\ 24c_4 \end{bmatrix} = \begin{bmatrix} s^{(4)} \\ 6(s')^2 s'' \hat{\kappa}' + (s')^4 \hat{\kappa}'' \\ (s')^4 \hat{\kappa}' \tau \end{bmatrix}_F$$
$$\mathbf{p}''''' = \begin{bmatrix} 0 \\ 120b_5 \\ 120c_5 \end{bmatrix} = \begin{bmatrix} s'''' \\ (s')^5 \hat{\kappa}''' + 10 (s')^3 s'' \hat{\kappa}'' \\ 3(s')^5 \hat{\kappa}'' \tau \end{bmatrix}_F$$

which yields

$$a_{1} = 1 = s'$$

$$b_{2} = 0 = \frac{(s')^{2}\kappa}{2}$$

$$b_{3} = 0 = \frac{(s')^{3}\hat{\kappa}'}{6}$$

$$c_{3} = 0 = \frac{(s')^{4}\kappa\tau}{6}$$

$$b_{4} = \frac{\hat{\kappa}''}{24} = \frac{(s')^{4}\hat{\kappa}''}{24}$$

$$c_{4} = 0 = \frac{(s')^{4}\hat{\kappa}'\tau}{24}$$

$$b_{5} = \frac{\hat{\kappa}'''}{120} = \frac{(s')^{5}\hat{\kappa}''' + 10(s')^{3}s''\hat{\kappa}''}{120}$$

$$c_{5} = \frac{3\hat{\kappa}''\tau}{120} = \frac{3(s')^{5}\hat{\kappa}''\tau}{120}$$

after substituting $\hat{\kappa}' = 0$.

κ	$\hat{\kappa}'$	au	polynomial curve
$\neq 0$	$\neq 0$	$\in \mathbb{R}$	$\begin{bmatrix} s \cdot t \\ \frac{s^2}{2} \kappa_i t^2 + \frac{s^3}{6} \hat{\kappa}'_i t^3 \\ \frac{s^3}{6} \kappa_i \tau_i t^3 \end{bmatrix}$
0	$\in \mathbb{R}$	0	$\begin{bmatrix} s \cdot t \\ \frac{s^3}{6} \hat{\kappa}'_i t^3 \\ 0 \end{bmatrix}$
0	$\neq 0$	$\in \mathbb{R}$	$\begin{bmatrix} s \cdot t \\ \frac{s^3}{6} \hat{\kappa}'_i t^3 + \frac{s^4}{24} \hat{\kappa}''_i t^4 \\ \frac{s^4}{12} \hat{\kappa}'_i \tau_i t^4 \end{bmatrix}$
0	0	$\in \mathbb{R}$	$\begin{bmatrix} \frac{s \cdot t}{\frac{s^4}{24} \hat{\kappa}_i'' t^4 + \frac{s^5}{120} \hat{\kappa}_i''' t^5} \\ \frac{\frac{s^5}{40} \hat{\kappa}_i'' \tau_i t^5} \end{bmatrix}$

Table B.2: Polynomial curves with speed parameter s > 0, reconstructing a third order GH data tuple at t = 0.

That is, if $\kappa_i = \hat{\kappa}'_i = 0$, the following polynomial reconstructs \mathbf{D}_i :

$$\mathbf{p}(t) = \begin{bmatrix} t \\ \frac{\hat{\kappa}''}{24}t^4 + \frac{\hat{\kappa}'''}{120}t^5 \\ \frac{\hat{\kappa}''\tau}{40}t^5 \end{bmatrix}$$
(B.3)

Incorporating a speed parameter into these curves is done by equating the tangential coordinate with $s \cdot t$, for some s > 0 value. Then the third order GH interpolant polynomials are of the form shown in Table B.2.

Appendix C

Second order GH interpolation with triangular patches

C.1 Three corner GH interpolation

Let us consider the problem of finding triangular Bézier surfaces that reconstruct the base point data at corners. This Appendix is from [50].

In what follows, the multi-indices denoted by $\mathbf{i} = (i, j, k)$ are always such that $i, j, k \in \mathbb{N}$.

Let there be given three G^2 base point data tuples $\mathbf{D}^{(i)}$, $\mathbf{i} = (0, 0, 1)$, (1, 0, 0), (0, 1, 0).

A degree n triangular Bézier patch is defined as [11]

$$\mathbf{b}(\mathbf{u}) = \sum_{||\mathbf{i}||_1=n} \mathbf{b}_{\mathbf{i}} B_{\mathbf{i}}^n(\mathbf{u})$$

where $\mathbf{u} = (u, v, w), ||\mathbf{u}||_1 = 1, \mathbf{b_i} \in \mathbb{E}^3, ||\mathbf{i}||_1 = n$, and the trinomial coefficients $B_{\mathbf{i}}^n(\mathbf{u})$ are

$$B_{\mathbf{i}}^{n}(\mathbf{u}) = \frac{n!}{i!j!k!}u^{i}v^{j}w^{k}$$

The three corner second order geometric Hermite interpolation problem is concerned with finding triangular Bézier surfaces that reconstruct the prescribed $\mathbf{D}^{(i)}$ base point data at parametric corners (0, 0, 1), (1, 0, 0), (0, 1, 0).

In the triangular setting, we have to make use of directional derivatives instead of parametric u and v derivatives: let \mathbf{d}_1 and \mathbf{d}_2 be two linearly independent directions in the domain. Using the blossom form ([11]), the derivatives



Figure C.1: The control points around a corner required for second order geometric Hermite interpolation.

can be written as

$$D_{\mathbf{d}_1,\mathbf{d}_2}^{r,s}\mathbf{b}(\mathbf{u}) = \frac{n!}{(n-r-s)!}\mathbf{b}[\mathbf{d}_1^{< r>}, \mathbf{d}_2^{< s>}, \mathbf{u}^{< n-r-s>}], \ r+s \le n$$

Let us consider the corner (u, v, w) = (0, 0, 0) and the corresponding, $\mathbf{D} = \mathbf{D}^{(001)}$ base point data, and let $\mathbf{d}_1 = (1, 0, 0) - (0, 0, 1) = (1, 0, -1)$ and $\mathbf{d}_2 = (0, 1, 0) - (0, 0, 1) = (0, 1, -1)$. The directional derivatives in these directions required for the computation of the first and second funademental forms are then

$$D_{\mathbf{d}_1}(\mathbf{0}) = n(\mathbf{b}_{1,0,n-1} - \mathbf{b}_{0,0,n}) \tag{C.1}$$

$$D_{\mathbf{d}_2}(\mathbf{0}) = n(\mathbf{b}_{0,1,n-1} - \mathbf{b}_{0,0,n})$$
(C.2)

$$D_{\mathbf{d}_1}^2(\mathbf{0}) = n(n-1)\left(\left(\mathbf{b}_{2,0,n-2} - \mathbf{b}_{1,0,n-1}\right) - \left(\mathbf{b}_{1,0,n-1} - \mathbf{b}_{0,0,n}\right)\right)$$
(C.3)

$$D_{\mathbf{d}_1,\mathbf{d}_2}(\mathbf{0}) = n(n-1) \left((\mathbf{b}_{1,1,n-2} - \mathbf{b}_{1,0,n-1}) - (\mathbf{b}_{0,1,n-1} - \mathbf{b}_{0,0,n}) \right)$$
(C.4)

$$D_{\mathbf{d}_2}^2(\mathbf{0}) = n(n-1)\left((\mathbf{b}_{0,2,n-2} - \mathbf{b}_{0,1,n-1}) - (\mathbf{b}_{0,1,n-1} - \mathbf{b}_{0,0,n})\right)$$
(C.5)

The control points used in the formulas and for the G^2 reconstruction are shown in figure C.1. Expressing these control points in the $(\mathbf{p}; \mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$ basis we get

$$\mathbf{b}_{0,0,n} = (0,0,0) \tag{C.6}$$

$$\mathbf{b}_{1,0,n-1} = (\overline{x}_{1,0,n-1}, \overline{y}_{1,0,n-1}, \overline{z}_{1,0,n-1}) = (\frac{x_u}{n}, \frac{y_u}{n}, 0) \tag{C.7}$$

$$\mathbf{b}_{0,1,n-1} = (\overline{x}_{0,1,n-1}, \overline{y}_{0,1,n-1}, \overline{z}_{0,1,n-1}) = (\frac{x_v}{n}, \frac{y_v}{n}, 0)$$
(C.8)

$$\mathbf{b}_{2,0,n-2} = (\overline{x}_{2,0,n-2}, \overline{y}_{2,0,n-2}, \overline{z}_{2,0,n-2})$$

$$= (\frac{x_{uu} + 2(n-1)x_u}{n(n-1)}, \frac{y_{uu} + 2(n-1)y_u}{n(n-1)}, \frac{z_{uu}}{n(n-1)})$$
(C.9)

$$\mathbf{b}_{1,1,n-2} = (\overline{x}_{1,1,n-2}, \overline{y}_{1,1,n-2}, \overline{z}_{1,1,n-2}) \tag{C.10}$$
$$= (\frac{x_{uv} + (n-1)(x_u + x_v)}{n(n-1)}, \frac{y_{uv} + (n-1)(x_u + x_v)}{n(n-1)}, \frac{z_{uv}}{n(n-1)})$$
$$\mathbf{b}_{0,2,n-2} = (\overline{x}_{0,2,n-2}, \overline{y}_{0,2,n-2}, \overline{z}_{0,2,n-2}) \tag{C.11}$$

$$\mathbf{b}_{0,2,n-2} = (\overline{x}_{0,2,n-2}, \overline{y}_{0,2,n-2}, \overline{z}_{0,2,n-2})$$

$$= (\frac{x_{vv} + 2(n-1)x_v}{n(n-1)}, \frac{y_{vv} + 2(n-1)y_v}{n(n-1)}, \frac{z_{vv}}{n(n-1)})$$
(C.11)

where (x_u, y_u) and (x_v, y_v) denote the coordinates of $D_{\mathbf{d}_1}\mathbf{b}(\mathbf{0}), D_{\mathbf{d}_2}\mathbf{b}(\mathbf{0})$, and $(x_{uu}, y_{uu}, z_{uu}), (x_{uv}, y_{uv}, z_{uv})$, and (x_{vv}, y_{vv}, z_{vv}) of $D_{\mathbf{d}_1}^2\mathbf{b}(\mathbf{0}), D_{\mathbf{d}_1,\mathbf{d}_2}\mathbf{b}(\mathbf{0})$, and $D_{\mathbf{d}_2}^2\mathbf{b}(\mathbf{0})$, respectively, in the basis of $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$.

The z_{uu}, z_{uv}, z_{vv} tangent plane offset values are computed by expressing the appropriate $D_{\mathbf{d}_1}\mathbf{b}(\mathbf{0}), D_{\mathbf{d}_2}\mathbf{b}(\mathbf{0})$ directional derivatives in the $(\mathbf{t}_1^{(\mathbf{i})}, \mathbf{t}_1^{(\mathbf{i})})$ basis.

The other parametric corners and corresponding control points can be treated similarly.

C.1.1 Quintic triangular Bézier patch

The control net of the quintic triangular Bézier is shown in figure C.2. As in the case of quadrilateral bi-quintic Bézier patches, the base point reconstruction regions at each corner are independent. If they are chosen such that their (C.6)-(C.11) coordinates satisfy (3.4)-(3.6), (3.12)-(3.14), the corner base point data are reconstructed. This proves the following

Theorem 24 There is always a quintic triangular Bézier solution for the three corner second order geometric Hermite interpolation problem.



Figure C.2: The control points of the quintic triangular Bézier patch. The red, blue, and green regions correspond to the control points necessary for the reconstruction of the G^2 base point data $\mathbf{D}^{(001)}$, $\mathbf{D}^{(100)}$, and $\mathbf{D}^{(010)}$, respectively.



Figure C.3: The control points of the quartic triangular Bézier patch. The red, blue, and green regions correspond to the control points necessary for the reconstruction of the G^2 base point data $\mathbf{D}^{(001)}$, $\mathbf{D}^{(100)}$, and $\mathbf{D}^{(010)}$, respectively.

C.1.2 Quartic triangular Bézier patch

The quartic triangular case has three control points, that are under the influence of two different base point data tuples, see figure C.3.

The constraints on the overlapping control points can be written as

$$\mathbf{b}_{2,0,2} \in \mathbf{M}^{(001)}(\overline{z}_{2,0,2}^{(001)}) \cap \mathbf{M}^{(100)}(\overline{z}_{2,0,2}^{(100)}) \tag{C.12}$$

$$\mathbf{b}_{0,2,2} \in \mathbf{M}^{(001)}(\overline{z}_{2,0,2}^{(001)}) \cap \mathbf{M}^{(010)}(\overline{z}_{2,0,2}^{(010)}) \tag{C.13}$$

$$\mathbf{b}_{2,2,0} \in \mathbf{M}^{(010)}(\overline{z}_{2,0,2}^{(010)}) \cap \mathbf{M}^{(100)}(\overline{z}_{2,0,2}^{(100)}) \tag{C.14}$$

Following the derivation of the quadrilateral bi-quartic case, the following can be stated

Theorem 25 The three corner second order geometric Hermite interpolation has a quartic triangular Bézier solution if and only if for each neighbouring D⁽ⁱ⁾, D^(j) base point data

$$H^{(\mathbf{i})} \cap H^{(\mathbf{j})} \neq \emptyset \tag{C.15}$$

C.1.3 Cubic triangular Bézier patch

The control net of the cubic triangular Bézier patch is shown in figure C.4.

Let $\mathbf{T}^{(i)}$ denote the tangent plane of $\mathbf{D}^{(i)}$, and let $\mathbf{M}_x^{(i)} = \mathbf{M}^i(z_x^{(i)})$, where $x \in \{uu, uv, vv\}$. Let $\mathbf{M}_i = \mathbf{M}_{i,j,k}$ denote the set of all feasible locations of control point $\mathbf{b}_{i,j,k}$, subject to (C.6)-(C.11).

All control points are under the influence of at least one base point data tuple. The following control points need to satisfy at least two base point reconstruction constraints:

$$\mathbf{b}_{1,0,2} \in \mathbf{T}^{(001)} \cap \mathbf{M}_{2,0,1}^{(100)} \tag{C.16}$$

$$\mathbf{b}_{2,0,1} \in \mathbf{T}^{(100)} \cap \mathbf{M}_{2,0,1}^{(001)} \tag{C.17}$$

$$\mathbf{b}_{0,1,2} \in \mathbf{T}^{(001)} \cap \mathbf{M}_{0,2,1}^{(010)} \tag{C.18}$$

$$\mathbf{b}_{0,2,1} \in \mathbf{T}^{(010)} \cap \mathbf{M}_{0,2,1}^{(001)} \tag{C.19}$$

$$\mathbf{b}_{2,1,0} \in \mathbf{T}^{(100)} \cap \mathbf{M}_{2,0,1}^{(010)} \tag{C.20}$$

$$\mathbf{b}_{1,2,0} \in \mathbf{T}^{(010)} \cap \mathbf{M}_{0,2,1}^{(100)} \tag{C.21}$$

$$\mathbf{b}_{1,1,1} \in \mathbf{M}_{1,1,1}^{(100)} \cap \mathbf{M}_{1,1,1}^{(010)} \cap \mathbf{M}_{1,1,1}^{(001)} \tag{C.22}$$



Figure C.4: The control points of the cubic triangular Bézier patch. The red, blue, and green regions correspond to the control points necessary for the reconstruction of the G^2 base point data $\mathbf{D}^{(001)}$, $\mathbf{D}^{(100)}$, and $\mathbf{D}^{(010)}$, respectively.

Conditions (C.16)-(C.21) can be satisfied as in the case of quadrilateral patches, and the control points affected by these conditions can be chosen as shown in section 3.4.

Condition (C.22) is a constraint specific to triangular patches, which depends on all the boundary directional derivatives, and the base point data of all three corners. It places $\mathbf{b}_{1,1,1}$ on the intersection of planes $\mathbf{M}_{1,1,1}^{(001)}$, $\mathbf{M}_{1,1,1}^{(010)}$, $\mathbf{M}_{1,1,1}^{(100)}$.

If the surface normals $\mathbf{m}^{(001)}$, $\mathbf{m}^{(010)}$, $\mathbf{m}^{(100)}$ are linearly independent, these planes are not parallel, and their intersection is a point. Placing $\mathbf{b}_{1,1,1}$ onto this intersection point satisfies the second order reconstruction constraint of all corners and the result is a cubic triangular patch solution to the three corner reconstruction. However, we cannot state intuitive geometric existence conditions for the parallel normal cases.

Let us consider the case where exactly two surface normals are parallel. Without the loss of generality, let us assume that $\mathbf{m}^{(001)} \parallel \mathbf{m}^{(100)}$. In order for the intersection in (C.22) to exist, we have to guarantee that $\mathbf{M}_{1,1,1}^{(100)} =$ $\mathbf{M}_{1,1,1}^{(001)} = \mathbf{M}$ holds, so that placing $\mathbf{b}_{1,1,1}$ anywhere along the intersection line of $\mathbf{M} \cap \mathbf{M}^{(010)}$, base point reconstruction constraints are satisfied. $\mathbf{M}_{1,1,1}^{(100)} = \mathbf{M}_{1,1,1}^{(001)}$ holds if and only if

$$(\mathbf{p}_M^{(001)} - \mathbf{p}_M^{(100)}) \cdot \mathbf{m} = 0,$$

where $\mathbf{p}_{M}^{(\mathbf{i})} = \mathbf{p}^{(\mathbf{i})} + m_{1,1,1}^{(\mathbf{i})} \mathbf{m}^{(\mathbf{i})}$ and $\mathbf{m} \in {\{\mathbf{m}^{(001)}, \mathbf{m}^{(100)}\}}.$

Finally, consider the case of $\mathbf{m}^{(001)} \parallel \mathbf{m}^{(100)} \parallel \mathbf{m}^{(010)}$. Now all three lifted tangent planes have to coincide, which can be formulated as

$$\left((\mathbf{p}_{M}^{(100)} - \mathbf{p}_{M}^{(001)}) \times (\mathbf{p}_{M}^{(010)} - \mathbf{p}_{M}^{(001)}) \right) \cdot \mathbf{m} > 0$$

 $\mathbf{m} \in {\{\mathbf{m}^{(001)}, \mathbf{m}^{(100)}, \mathbf{m}^{(010)}\}}, \text{ excluding the cases } \mathbf{p}_M^{(100)} = \mathbf{p}_M^{(001)} \text{ and } \mathbf{p}_M^{(010)} = \mathbf{p}_M^{(001)}.$

Appendix D

Differential geometry of lines of curvature

Lines of curvature are curves on a surface having the property that at every point, the direction of their tangent vector coincides with one of the principal directions.

Their $(u(t), v(t)), t \in \mathbb{R}$ curve in the domain of a parametric surface is defined by

$$(L - \kappa_n E)u' + (M - \kappa_n F)v' = 0 \tag{D.1}$$

$$(M - \kappa_n F)u' + (N - \kappa_n G)v' = 0 \tag{D.2}$$

which is result shown in every classical differential geometry textbook, see for example [9], [45]. Lines of curvature form an orthogonal net over the surface – outside the neighborhood of umbilical points –, and we have seen how this orthogonal net can be used to define a natural parametrization of surfaces.

The behavior of lines of curvature around umbilics was of theoretical and practical interest. In these points, every tangent direction is a principal direction due to the coincidence of both principal curvatures. Equations (D.1)-(D.2) cannot provide the tangents of lines of curvature at an umbilic, instead, higher order derivatives of the surface are required.

Although spheres and planes are obvious examples of surfaces that consist of only umbilical points, they also turn out to be the only surfaces that are composed of exclusively umbilical points. Umbilics of a surface are usually of finite number and they are also isolated [38]. Computation of higher order differential geometric properties of lines of curvature was recently investigated by Joo et. al. in [24], both at umbilics and non-umbilical points. The remainder of this appendix details their method. We only consider the case of non-vanishing curvatures - for handling this case, the interested reader is referred to [24].

They have taken advantage of the fact that a line of curvature can be considered as a

$$\mathbf{c}(t) = \mathbf{r}\big(u(t), v(t)\big)$$

curve on the surface, whose derivatives can be expressed in terms of the partial derivatives of the parametric surface $\mathbf{r}(u, v)$, as well as a curve in space, whose Frenet coordinates – in case of arc-length parametrization – are made up of combinations of differential geometric invariants, that is, for example

$$\mathbf{c}' = \mathbf{r}_{u}u' + \mathbf{r}_{v}v' = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{F},$$
$$\mathbf{c}'' = \mathbf{r}_{uu}u' + 2\mathbf{r}_{uv}u'v' + \mathbf{r}_{vv}v' + \mathbf{r}_{u}u'' + \mathbf{r}_{v}v'' = \begin{bmatrix} 0\\\kappa\\0 \end{bmatrix}_{F},$$

if u(t), v(t) are subject to

$$u' = \eta (M - \kappa_p F)$$
$$v' = -\eta (L - \kappa_p E)$$

where

$$\eta = \frac{\pm 1}{\sqrt{E(M - \kappa_p F)^2 + 2F(M - \kappa_p F)(L - \kappa_p E) + G(L - \kappa_p E)^2}},$$

which ensures the unit-speed parametrization of the line of curvature (u(t), v(t)).

Let us consider the two $\mathbf{c}_p(t)$ lines of curvature of a surface at a point, p = 1, 2. The second derivative of $\mathbf{c}_p(t)$ can be written as

$$\mathbf{c}'' = \kappa_{pn}\mathbf{m} + \kappa_{pg}\mathbf{u} = \mathbf{r}_u u'' + \mathbf{r}_v v'' + \mathbf{a}_2 \tag{D.3}$$

where **m** is the unit surface normal, $\mathbf{u}_p = \mathbf{m} \times \mathbf{t}_p$, and κ_{pn}, κ_{pg} denote the normal and geodesic curvature of the line of curvature, p = 1, 2, and

$$\mathbf{a}_2 = \mathbf{r}_{uu}(u')^2 + 2\mathbf{r}_{uv}u'v' + \mathbf{r}_{vv}(v')^2$$
.

By taking the dot product of (D.3) with \mathbf{r}_u , then with \mathbf{r}_v , one gets

$$Eu'' + Fv'' - \kappa_{pg}(\mathbf{u} \cdot \mathbf{r}_u) = -\mathbf{a}_2 \cdot \mathbf{r}_u \tag{D.4}$$

$$Fu'' + Gv'' - \kappa_{pg}(\mathbf{u} \cdot \mathbf{r}_v) = -\mathbf{a}_2 \cdot \mathbf{r}_v \tag{D.5}$$

that is, there are two linear equations for three unknown quantities: u'', v'', κ_{pg} . A third equation can be obtained by differentiating either (D.1) or (D.2). By doing the former, one gets

$$(L - \kappa_{pn}E)u'' + (M - \kappa_{pn}F)v'' = \beta_{1p}$$

where $\beta_{1p} = -(L' - \kappa'_p E - \kappa_p E')u' + (M' - \kappa'_p F - \kappa_p F')v'$. It is important to note that in certain cases, differentiating (D.1) yields an identity of the form 0 = 0. If this occurs, the equation obtained from differentiating (D.2)

$$(M - \kappa_n F)u'' + (N - \kappa_n G)v'' = \gamma_1$$

is to be used instead to complement (D.4) and (D.5), where

$$\gamma_1 = -(M' - \kappa'_n F - \kappa_n F')u' - (N' - \kappa'_n G - \kappa_n G)v' .$$

Therefore, the solution to the system

$$\begin{bmatrix} E & F & -(\mathbf{u}_p \cdot \mathbf{r}_u) \\ F & G & -(\mathbf{u}_p \cdot \mathbf{r}_v) \\ L - \kappa_p E & M - \kappa_p F & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_2 \cdot \mathbf{r}_u \\ -\mathbf{a}_2 \cdot \mathbf{r}_v \\ \beta_1 \end{bmatrix}$$

or, in case of its singularity, to

$$\begin{bmatrix} E & F & -(\mathbf{u}_p \cdot \mathbf{p}_u) \\ F & G & -(\mathbf{u}_p \cdot \mathbf{p}_v) \\ M - \kappa_n F & N - \kappa_n G & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_2 \cdot \mathbf{p}_u \\ -\mathbf{a}_2 \cdot \mathbf{p}_v \\ \gamma_1 \end{bmatrix}$$

produces the u'', v'' parametric second derivatives of the $\mathbf{c}_p(t)$ line of curvature in the parametric domain of $\mathbf{r}(u, v)$, and its κ_{pq} geodesic torsion in space.

Singularity of one of the systems may happen in 'nice' parametrizations too: if the parameter lines are lines of curvature at a point and the first derivatives are of unit length, either (D.4) or (D.5) becomes an identity.

A less articulated attribute of lines of curvature is the fact that their order-n geometric invariants depend on surface derivatives up to order n + 1.

This is an immediate result of the defining differential equation of lines of curvature (D.1)-(D.2), which defines the first order u', v' behavior of lines of curvature via the first and second fundamental form coefficients, that is, due to the presence of E, F, G, L, M, N, via $\mathbf{r}_u, \mathbf{r}_v, \mathbf{r}_{uu}, \mathbf{r}_{vv}$. Similarly, $u^{(i)}, v^{(i)}$ depend on surface derivatives up to order i + 1. This is also evident from the work of Joo et. al. quoted here.

In general, provided that none of the principal curvatures is equal to zero, using the (n + 1)-th order partial derivatives of the surface, the order n geometric invariants of the lines of curvature can be computed by

$$\begin{bmatrix} E & F & -\mathbf{n}_{p} \cdot \mathbf{r}_{u} & -\kappa_{p} \mathbf{b}_{p} \cdot \mathbf{r}_{u} \\ F & G & -\mathbf{n}_{p} \cdot \mathbf{r}_{v} & -\kappa_{p} \mathbf{b}_{p} \cdot \mathbf{r}_{v} \\ \mathbf{n}_{p} \cdot \mathbf{r}_{u} & \mathbf{n}_{p} \cdot \mathbf{r}_{v} & -1 & 0 \\ L - \kappa_{pn} E & M - \kappa_{pn} F & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u^{(m)} \\ v^{(m)} \\ \hat{\kappa}^{(m-2)} \\ \hat{\tau}^{(m-3)} \end{bmatrix} =$$
(D.6)
$$\begin{bmatrix} \mathbf{a}_{m} \cdot \mathbf{r}_{u} + x_{pm} \mathbf{t}_{p} \cdot \mathbf{r}_{u} + (y_{pm} - \hat{\kappa} + p^{(m-2)}) \mathbf{n}_{p} \cdot \mathbf{r}_{u} + (z_{pm} - \kappa_{p} \hat{\tau}_{p}^{(m-3)}) \mathbf{b}_{p} \cdot \mathbf{r}_{u} \\ \mathbf{a}_{m} \cdot \mathbf{r}_{v} + x_{pm} \mathbf{t}_{p} \cdot \mathbf{r}_{v} + (y_{pm} - \hat{\kappa} + p^{(m-2)}) \mathbf{n}_{p} \cdot \mathbf{r}_{v} + (z_{pm} - \kappa_{p} \hat{\tau}_{p}^{(m-3)}) \mathbf{b}_{p} \cdot \mathbf{r}_{v} \\ \beta_{p,m-1} \end{bmatrix}$$

where $\mathbf{t}_p, \mathbf{n}_p, \mathbf{b}_p$ denotes the vectors of the Frenet trihedron of $\mathbf{c}_p(t)$, $\mathbf{c}^{(m)} = [x_{pm}, y_{pm}, z_{pm}]_F^T$, and κ_p , τ_p are the curvature and torsion of the $\mathbf{c}_p(t)$ line of curvature as a space curve, p = 1, 2, and \mathbf{a}_m , β_{pm} are defined analogously as before, using higher order differentiation. For a detailed discussion of this, as well as the case of umbilical points, the interested reader is referred to [24].

Appendix E

Paraboloid solutions to second and third order GH interpolation

E.1 Second order paraboloids

Principal curvature reconstruction E.1.1

Let us consider the paraboloid of the form

$$\mathbf{p}^{(ij)}(u,v) = \mathbf{p}^{(ij)} + [\mathbf{t}_1^{(ij)}, \mathbf{t}_2^{(ij)}, \mathbf{m}^{(ij)}] \begin{bmatrix} u \\ v \\ \frac{\kappa_1^{(ij)}}{2}u^2 + \frac{\kappa_2^{(ij)}}{2}v^2 \end{bmatrix} .$$
 (E.1)

where $\mathbf{t}_1^{(ij)}, \mathbf{t}_2^{(ij)}, \mathbf{m}^{(ij)}$ form a right-handed orthonormal system. Its partial derivatives in the $(\mathbf{t}_1^{(ij)}, \mathbf{t}_2^{(ij)}, \mathbf{n}^{(ij)})$ Darboux frame are

$$\mathbf{p}_{u} = \begin{bmatrix} 1\\0\\\kappa_{1}^{(ij)}u \end{bmatrix}_{D}, \ \mathbf{p}_{u} = \begin{bmatrix} 1\\0\\\kappa_{2}^{(ij)}v \end{bmatrix}_{D}$$
$$\mathbf{p}_{uu} = \begin{bmatrix} 0\\0\\\kappa_{1}^{(ij)} \end{bmatrix}_{D}, \ \mathbf{p}_{uv} = \mathbf{0}, \ \mathbf{p}_{vv} = \begin{bmatrix} 0\\0\\\kappa_{2}^{(ij)} \end{bmatrix}_{D}$$

thus, at (u, v) = (0, 0) we have

$$\mathbf{p}_{u} = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{D}, \ \mathbf{p}_{v} = \begin{bmatrix} 0\\1\\0 \end{bmatrix}_{D}$$
$$\mathbf{p}_{uu} = \begin{bmatrix} 0\\0\\\kappa_{1}^{(ij)} \end{bmatrix}_{D}, \ \mathbf{p}_{uv} = \mathbf{0}, \ \mathbf{p}_{vv} = \begin{bmatrix} 0\\0\\\kappa_{2}^{(ij)} \end{bmatrix}_{D}$$

The principal normal here is $\mathbf{p}_u \times \mathbf{p}_v = \begin{bmatrix} 0\\0\\1 \end{bmatrix}_D$, which equals to $\mathbf{m}^{(ij)}$.

The first and second fundamental forms are

$$E = 1 , F = 0 , G = 0$$
$$L = \kappa_1^{(ij)} , M = 0 , N = \kappa_2^{(ij)}$$

Thus, it follows immediately that the parameter lines are lines of curvature at (0,0) and the principal directions in space are $\mathbf{t}_1^{(ij)}$ and $\mathbf{t}_2^{(ij)}$, and the corresponding principal curvatures are $\kappa_1^{(ij)}$, $\kappa_2^{(ij)}$.

E.1.2 Principal and geodesic curvature reconstruction

If one would like to exert control over the geodesic curvature of lines of curvature, the formulation of (E.1) will not suffice. Instead, quadratic terms have to be introduced along the Darboux x and y coordinates of the paraboloid, that is, they should be of the form

$$\mathbf{p}^{(ij)}(u,v) = \mathbf{p}^{(ij)} + \begin{bmatrix} u + \frac{a}{2}v^2 \\ v + \frac{b}{2}u^2 \\ \frac{c}{2}u^2 + \frac{d}{2}v^2 \end{bmatrix}_D$$

We show in this subsection that the above is enough for the simultaneous reconstruction of normal and geodesic curvature of lines of curvatures.

The partial derivatives of the paraboloid at (u, v) = (0, 0) are then

$$\mathbf{p}_{u}^{(ij)} = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{D}, \ \mathbf{p}_{v}^{(ij)} = \begin{bmatrix} 0\\1\\0 \end{bmatrix}_{d}, \ \mathbf{m} = \begin{bmatrix} 0\\0\\1 \end{bmatrix}_{D}$$
$$\mathbf{p}_{uu}^{(ij)} = \begin{bmatrix} 0\\b\\c \end{bmatrix}_{D}, \ \mathbf{p}_{uv}^{(ij)} = \begin{bmatrix} 0\\0\\0 \end{bmatrix}_{D}, \ \mathbf{p}_{vv}^{(ij)} = \begin{bmatrix} a\\0\\d \end{bmatrix}_{D}$$

from which it follows that

$$\begin{split} E &= 1 \ , \ F = 0 \ , \ G = 1 \ , \\ L &= c \ , \ M = 0 \ , \ N = d \ . \end{split}$$

The principal directions in the domain of $\mathbf{p}(u, v)$ at (u, v) = (0, 0) are again (u', v') = (1, 0) and (u', v') = (0, 1), and the corresponding principal curvatures are

$$\kappa_{1n}^{(ij)}(0,0) = \frac{L}{E} = c ,$$

$$\kappa_{2n}^{(ij)}(0,0) = \frac{N}{G} = d .$$

If $\kappa_1^{(ij)}, \kappa_2^{(ij)}$ are to be the curvatures of the two lines of curvature, the following should hold

$$\kappa_{1n}^{(ij)}(0,0) = \kappa_1^{(ij)} \cos \alpha$$
$$\kappa_{2n}^{(ij)}(0,0) = \kappa_2^{(ij)} \cos \beta$$

that is,

$$c = \kappa_1^{(ij)} \cos \alpha ,$$

$$d = \kappa_2^{(ij)} \cos \beta .$$

Now let us verify that there exist $a, b \in \mathbb{R}$ such that the reconstruction of geodesic curvature is also possible.

Let us consider the line of curvature corresponding to the p = 1 principal

direction first. If this line of curvature, denoted by $\mathbf{c}_1(s)$, is parametrized by arc-length, then if $\kappa_1^{(ij)}$ is its curvature, the second derivative is of the form

$$\mathbf{c}_1''(s) = \kappa_1^{(ij)} \mathbf{n}_1^{(ij)}$$

where $\mathbf{n}_1^{(ij)}$ is the normal vector of $\mathbf{c}_1(s)$ at s = 0. Since $\mathbf{c}_1(s)$ is also expressed as a curve on a surface, its second derivative can be written as

$$\mathbf{c}_{1}''(s) = \mathbf{p}_{u}^{(ij)}u'' + \mathbf{p}_{v}^{(ij)}v'' + \mathbf{p}_{uu}^{(ij)}(u')^{2} + 2\mathbf{p}_{uv}^{(ij)}u'v' + \mathbf{p}_{vv}^{(ij)}(v')^{2}$$

that is

$$\kappa_1^{(ij)} \mathbf{n}_1 = \mathbf{p}_u^{(ij)} u'' + \mathbf{p}_v^{(ij)} v'' + \mathbf{p}_{uu}^{(ij)} (u')^2 + 2\mathbf{p}_{uv}^{(ij)} u'v' + \mathbf{p}_{vv}^{(ij)} (v')^2$$

should hold.

Since the parametric direction vector of the line of curvature corresponding to the p = 1 principal direction is u' = 1, v' = 0, the above simplifies to

$$\kappa_1^{(ij)} \mathbf{n}_1 = \mathbf{p}_u^{(ij)} u'' + \mathbf{p}_v^{(ij)} v'' + \mathbf{p}_{uu}^{(ij)} (u')^2 .$$

Since the curvature of the line of curvature curve can be expressed in the plane perpendicular to the tangent of $\mathbf{c}_1(s)$ as

$$\kappa_1^{(ij)}\mathbf{n}_1 = \kappa_{1n}^{(ij)}\mathbf{m} + \kappa_{1g}^{(ij)}\mathbf{u}$$

rewriting it into Darbaux coordinates one gets

$$\begin{bmatrix} 0\\ \kappa_1^{(ij)} \sin \alpha\\ \kappa_1^{(ij)} \cos \alpha \end{bmatrix}_D$$

that is, the second partial derivative of our unknown paraboloid with respect to u is such that

$$\mathbf{c}''(s) = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{D} u'' + \begin{bmatrix} 0\\1\\0 \end{bmatrix}_{D} v'' + \begin{bmatrix} 0\\b\\c \end{bmatrix}_{D} (u')^{2}$$

from which it follows that b and c are subject to

$$\begin{bmatrix} 0\\ \kappa_1^{(ij)} \sin \alpha\\ \kappa_1^{(ij)} \cos \alpha \end{bmatrix}_D = \begin{bmatrix} u''\\ v''+b\\ c \end{bmatrix}_D$$

.

By expanding this, one gets

$$u'' = 0$$
$$v'' + b = \kappa_1^{(ij)} \sin \alpha$$
$$c = \kappa_1^{(ij)} \cos \alpha$$

that is, the second partial derivative of the unknown paraboloid is to be sought in the form of

$$\mathbf{p}_{uu} = \begin{bmatrix} 0\\ \kappa_1^{(ij)} \sin \alpha - ddv\\ \kappa_1^{(ij)} \cos \alpha \end{bmatrix}_D$$

where the unkown v'' term is substituted by a ddv parameter.

For $\mathbf{c}_2(s)$, that is, the line of curvature corresponding to the second principal direction (u', v') = (0, 1) one similarly gets

$$\kappa_{2,n}^{(ij)}\mathbf{m} + \kappa_{2,g}^{(ij)}\mathbf{u}_{2} = \begin{bmatrix} 1\\0\\0 \end{bmatrix}_{D} u'' + \begin{bmatrix} 0\\1\\0 \end{bmatrix}_{D} v'' + \begin{bmatrix} a\\0\\d \end{bmatrix}_{D} (u')^{2}$$

and by using that $\mathbf{u}_2 = \mathbf{m} \times \mathbf{t}_2 = -\mathbf{t}_1$, this is expressed in the Darboux frame as

$$\begin{bmatrix} -\kappa_2^{(ij)}\sin\beta\\ 0\\ \kappa_2^{(ij)}\cos\beta \end{bmatrix}_D = \begin{bmatrix} u''+a\\ v''\\ d \end{bmatrix}_D$$

that is, the second order partial derivative in the direction of v is

$$\mathbf{p}_{vv} = \begin{bmatrix} -\kappa_2^{(ij)} \sin \beta - ddu \\ 0 \\ \kappa_2^{(ij)} \cos \beta \end{bmatrix}_D$$

.

Then the unknown paraboloid can be written as

$$\mathbf{p}(u,v) = \mathbf{p} + \begin{bmatrix} u - \frac{\kappa_2^{(ij)} \sin \beta + ddu}{2} v^2 \\ v + \frac{\kappa_1^{(ij)} \sin \alpha - ddv}{2} u^2 \\ \kappa_1^{(ij)} \cos \alpha \frac{u^2}{2} + \kappa_2^{(ij)} \cos \beta \frac{v^2}{2} \end{bmatrix}_D$$

Now let us investigave what values ddu, ddv should attain so that the geodesic curvatures of lines of curvature are reconstructed!

The desired geodesic curvatures are

$$\kappa_{1g}^{(ij)} = \kappa_1^{(ij)} \sin \alpha$$
$$\kappa_{2g}^{(ij)} = \kappa_2^{(ij)} \sin \beta$$

Following the method proposed by Joo et al. in [24] to compute the differential geometric properties of lines of curvature, we need to verify if ddu and ddv can be assigned a value to reconstruct these geodesic curvatures.

Let us compute the u'', v'' derivatives and the geodesic curvature κ_{pg} of the $\mathbf{c}_p(s)$ line of curvature, p = 1, 2.

Following the method reviewed in Appendix D, either the

$$\begin{bmatrix} E & F & -(\mathbf{u}_2 \cdot \mathbf{p}_u) \\ F & G & -(\mathbf{u}_2 \cdot \mathbf{p}_v) \\ L - \kappa_n E & M - \kappa_n F & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_2 \cdot \mathbf{p}_u \\ -\mathbf{a}_2 \cdot \mathbf{p}_v \\ \beta_1 \end{bmatrix}$$

or the

$$\begin{bmatrix} E & F & -(\mathbf{u}_p \cdot \mathbf{p}_u) \\ F & G & -(\mathbf{u}_p \cdot \mathbf{p}_v) \\ M - \kappa_n F & N - \kappa_n G & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_2 \cdot \mathbf{p}_u \\ -\mathbf{a}_2 \cdot \mathbf{p}_v \\ \gamma_1 \end{bmatrix}$$

system of linear equations can be used to compute the unknown u'', v'', κ_{pg} values, where

$$\kappa_n = \kappa_{pn}^{(ij)}$$

$$\beta_1 = -(L' - \kappa'_n E - \kappa_n E')u' - (M' - \kappa'_n F - \kappa_n F')v'$$

$$\gamma_1 = -(M' - \kappa'_n F - \kappa_n F')u' - (N' - \kappa'_n G - \kappa_n G)v'$$

In the p = 1 case, the following two systems of linear equations arise:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_2 \cdot \mathbf{p}_u \\ -\mathbf{a}_2 \cdot \mathbf{p}_v \\ \beta_1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & \kappa_{2,n} - \kappa_{1,n} & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_2 \cdot \mathbf{p}_u \\ -\mathbf{a}_2 \cdot \mathbf{p}_v \\ \gamma_1 \end{bmatrix}$$

The former is singular, due the first of the equations

$$(L - \kappa_n E)u' + (M - \kappa_n F)v' = 0$$
$$(M - \kappa_n F)u' + (N - \kappa_n G)v' = 0$$

becoming an identity.

Conversely, in the case of p = 2, the systems of linear equations are

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ \kappa_{1,n} - \kappa_{2,n} & 0 & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_2 \cdot \mathbf{p}_u \\ -\mathbf{a}_2 \cdot \mathbf{p}_v \\ \beta_1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_2 \cdot \mathbf{p}_u \\ -\mathbf{a}_2 \cdot \mathbf{p}_v \\ \gamma_1 \end{bmatrix}$$

that is, here the latter produces a singular system.

Solving the appropriate system for the p = 1 case, one gets

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & \kappa_{2,n} - \kappa_{1,n} & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} 0 \\ ddv - \kappa_1^{(ij)} \sin \alpha \\ \kappa_1^{(ij)} \cos \beta(\kappa_1^{(ij)} \sin \alpha - ddv) \end{bmatrix}$$

from which

$$ddv = \frac{\left(\kappa_1^{(ij)}\right)^2 \cos \alpha \sin \alpha}{\kappa_2^{(ij)} \cos \beta} = \frac{\kappa_{1,n} \kappa_{1,g}}{\kappa_{2,n}}$$

should hold.

Similarly, the solution of the non-singular system corresponding to p = 2

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ \kappa_{1,n} - \kappa_{2,n} & 0 & 0 \end{bmatrix} \begin{bmatrix} u'' \\ v'' \\ \kappa_g \end{bmatrix} = \begin{bmatrix} \kappa_2^{(ij)} \sin \beta + ddu \\ 0 \\ -\kappa_2^{(ij)} \cos \beta(\kappa_2^{(ij)} \sin \beta + ddu) \end{bmatrix}$$

yields

$$ddu = \frac{\left(\kappa_2^{(ij)}\right)^2 \cos\beta\sin\beta}{\kappa_1^{(ij)}\cos\alpha} = \frac{\kappa_{2,n}\kappa_{2,g}}{\kappa_{1,n}}$$

Hence, the paraboloid with normal and geodesic curvature reconstruction is

$$\mathbf{p}(u,v) = \mathbf{p} + \begin{bmatrix} u - \frac{1}{2} \left(\kappa_{2,g} + \frac{\kappa_{2,n}\kappa_{2,g}}{\kappa_{1,n}} \right) v^2 \\ v + \frac{1}{2} \left(\kappa_{1,g} - \frac{\kappa_{1,n}\kappa_{1,g}}{\kappa_{2,n}} \right) u^2 \\ \frac{\kappa_{1,n}}{2} u^2 + \frac{\kappa_{2,n}}{2} v^2 \end{bmatrix}_D$$
(E.2)

For this paraboloid, the lines of curvature in its domain are

$$(u',v') = (1,0) , \ (u'',v'') = (0,\frac{\kappa_{1,n}\kappa_{1,g}}{\kappa_{2,n}})$$

and

$$(u',v') = (0,1)$$
, $(u'',v'') = (-\frac{\kappa_{2,n}\kappa_{2,g}}{\kappa_{1,n}},0)$

Please note that we can not control the second order derivative of the principal direction and the reconstruction of geodesic curvature simultaneously: there are no degrees of freedom left in the system to do this. Instead, cubic polynomial surfaces need to be used.

E.2 Third order paraboloids

Let us find a cubic polynomial surface, that at (u, v) = (0, 0) reconstructs the geometric invariants of lines of curvatures of a prescribed

$$\mathbf{E}^{(i)} = \{\mathbf{p}^{(i)}; \mathbf{t}_1^{(i)}, \mathbf{n}_1^{(i)}, \mathbf{b}_1^{(i)}; \mathbf{t}_2^{(i)}, \mathbf{n}_2^{(i)}, \mathbf{b}_2^{(i)}; \kappa_1^{(i)}, \kappa_2^{(i)}; \tau_1^{(i)}, \hat{\kappa}_1^{(i)}, \tau_2^{(i)}, \hat{\kappa}_2^{(i)}, \hat{\kappa}_2^{(i)}\}$$

GH data tuple, $\mathbf{t}_1, \mathbf{t}_2$ being orthonormal and let $\mathbf{m} = \mathbf{t}_1 \times \mathbf{t}_2$.

E.2.1 Normal projection reconstruction

For the sake of simplicity, let us omit the upper (i) indices and let us add cubic terms to the paraboloid of (E.2) as

$$\mathbf{p}(u,v) = \mathbf{p} + \begin{bmatrix} u - \frac{1}{2} \left(\kappa_{2,g} + \frac{\kappa_{2,n}\kappa_{2,g}}{\kappa_{1,n}} \right) v^2 \\ v + \frac{1}{2} \left(\kappa_{1,g} - \frac{\kappa_{1,n}\kappa_{1,g}}{\kappa_{2,n}} \right) u^2 \\ \frac{\kappa_{1,n}}{2} u^2 + \frac{\kappa_{2,n}}{2} v^2 + \frac{a}{6} u^3 + \frac{b}{6} v^3 \end{bmatrix}_D$$

where $D = (\mathbf{t}_1, \mathbf{t}_2, \mathbf{m})$. The third order partial derivatives at (0, 0) are

$$\mathbf{p}_{uuu} = \begin{bmatrix} 0\\0\\a \end{bmatrix}_{D}, \ \mathbf{p}_{uuv} = \mathbf{p}_{uvv} = \mathbf{0}, \ \mathbf{p}_{vvv} = \begin{bmatrix} 0\\0\\b \end{bmatrix}_{D}$$

Formalization of third order GH reconstruction problem yields the

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{p}_{u} \\ \mathbf{p}_{v} \\ \mathbf{p}_{uu} \\ \mathbf{p}_{vv} \\ \mathbf{p}_{uuu} \\ \mathbf{p}_{vvv} \\ \mathbf{p}_{uuu} \\ \mathbf{p}_{vvv} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & x_{u} & y_{u} & 0 & 0 \\ 0 & x_{v} & y_{v} & 0 & 0 \\ 0 & x_{uu} & y_{uu} & \kappa_{1n}x_{u}^{2} + \kappa_{2n}y_{u}^{2} \\ 0 & x_{vvv} & y_{vv} & \kappa_{1n}x_{v}^{2} + \kappa_{2n}y_{v}^{2} \\ 0 & x_{uuu} & y_{uuu} & K_{1}x_{u}^{3} + K_{2}y_{u}^{3} + 3\kappa_{1n}x_{u}(x_{uu} + \kappa_{2g}y_{u}^{2}) + 3\kappa_{2n}y_{u}(y_{uu} + \kappa_{2g}x_{u}^{2}) \\ 0 & x_{vvv} & y_{vvv} & K_{1}x_{v}^{3} + K_{2}y_{v}^{3} + 3\kappa_{1n}x_{v}(x_{vv} + \kappa_{2g}y_{v}^{2}) + 3\kappa_{2n}y_{v}(y_{vv} + \kappa_{2g}x_{v}^{2}) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p} \\ \mathbf{t}_{1} \\ \mathbf{t}_{2} \\ \mathbf{m} \end{bmatrix} \\ \mathbf{p}_{vvv} \end{bmatrix}$$
(E.3)

system of constraints, where $K_i = (\kappa_i \tau_{ig} + \hat{\kappa}'_{in})$. The mixed-partial derivative lines are omitted from above, taking into account that all mixed-partial derivatives of $\mathbf{p}(u, v)$ vanish and at (0, 0) – because the parameter lines coincide with lines of curvature there – the mixed-partial geometric constraints amount to zero as well. We have also used that

$$s_u = x_u , \ s_v = x_v$$
$$t_u = y_u , \ t_v = y_v$$
$$s_{uu} = x_{uu} + \kappa_{2g} y_u^2 , \ s_{vv} = x_{vv} + \kappa_{2g} y_v^2$$
$$t_{uu} = y_{uu} + \kappa_{2g} x_u^2 , \ t_{vv} = y_{vv} + \kappa_{2g} x_v^2$$

Thus, for the third order partial derivatives one gets

$$\begin{bmatrix} 0\\0\\a \end{bmatrix}_{D} = \begin{bmatrix} x_{uuu}\\y_{uuu}\\K_{1}x_{u}^{3} + K_{2}y_{u}^{3} + 3\kappa_{1n}x_{u}(x_{uu} + \kappa_{2g}y_{u}^{2}) + 3\kappa_{2n}y_{u}(y_{uu} + \kappa_{2g}x_{u}^{2}) \end{bmatrix}$$
$$\begin{bmatrix} 0\\0\\b \end{bmatrix}_{d} = \begin{bmatrix} x_{vvv}\\y_{vvv}\\K_{1}x_{v}^{3} + K_{2}y_{v}^{3} + 3\kappa_{1n}x_{v}(x_{vv} + \kappa_{2g}y_{v}^{2}) + 3\kappa_{2n}y_{v}(y_{vv} + \kappa_{2g}x_{v}^{2}) \end{bmatrix}$$

so the cubic polynomial of

$$\mathbf{p}(u,v) = \mathbf{p} + \begin{bmatrix} u - \frac{1}{2} \left(\kappa_{2,g} + \frac{\kappa_{2,n}\kappa_{2,g}}{\kappa_{1,n}} \right) v^2 \\ v + \frac{1}{2} \left(\kappa_{1,g} - \frac{\kappa_{1,n}\kappa_{1,g}}{\kappa_{2,n}} \right) u^2 \\ \frac{\kappa_{1n}}{2} u^2 + \frac{\kappa_{2n}}{2} v^2 + \frac{a}{6} u^3 + \frac{b}{6} v^3 \end{bmatrix}_D$$

with

$$a = K_1 x_u^3 + K_2 y_u^3 + 3\kappa_{1n} x_u (x_{uu} + \kappa_{2g} y_u^2) + 3\kappa_{2n} y_u (y_{uu} + \kappa_{2g} x_u^2)$$

$$b = K_1 x_v^3 + K_2 y_v^3 + 3\kappa_{1n} x_v (x_{vv} + \kappa_{2g} y_v^2) + 3\kappa_{2n} y_v (y_{vv} + \kappa_{2g} x_v^2)$$

reconstruct the prescribed GH data tuple so that it ensures G^3 connection of surfaces at (0,0). This is true, because as we have already proved, (E.2) ensures G^2 continuity of lines of curvature.

Bibliography

- N. N. Abdelmalek, L₁ solution of overdetermined systems of linear equations. ACM Trans. Math. Software 6, 220-227, 1980
- G. Albrecht, R. T. Farouki, Construction of C² Pythagorean-hodograph interpolating splines by the homotopy method, Adv. Comput. Math. 5 (1996), no. 4, 417-442. MR MR1414289 (97k:65033)
- [3] I. Barrodale and F. D. K. Roberts, An Improved Algorithm for Discrete l₁ Linear Approximation, SIAM Journal on Numerical Analysis, Vol. 10, No. 5 (Oct., 1973) (pp. 839-848)
- [4] W. Boehm, On the definition of geometric continuity, Computer-Aided Design, Volume 20 Issue 7, Sept. 1988, Pages 370 - 372
- [5] J. A. Cadzow, Minimum l₁, l₂, and l_∞ Norm Approximate Solutions to an Overdetermined System of Linear Equations, Digital Signal Processing journal, Volume 12, no. 4 (2002), Pages 524 - 560.
- [6] E. W. Cheney, Introduction to Approximation Theory, 2nd ed., American Mathematical Society. 1982, ISBN 978-0-8218-1374-4.
- [7] C. de Boor, K. Höllig, M. Sabin, High accuracy geometric Hermite interpolation, Computer Aided Geometric Design 1987;4(4):269-78.
- [8] M. Boschiroli, C. Fünfzig, L. Romani, G. Albrecht, G¹ rational blend interpolatory schemes: A comparative study, Graphical Models, Volume 74 Issue 1, January, 2012, Pages 29-49
- [9] M. P. do Carmo, Differential Geometry of Curves and Surfaces, Pearson, 1st edition, 1976

- [10] E. H. Dohaa, A. H. Bhrawyb, M. A. Sakerc, Integrals of Bernstein polynomials: An application for the solution of high even-order differential equations, Applied Mathematics Letters, Volume 24, Issue 4, April 2011, Pages 559–565
- [11] G. E. Farin, Curves and surfaces for CAGD: a practical guide, 5th edition, ISBN 1-55860-737-4, Morgan Kaufmann Publishers Inc. 2002
- [12] G. E. Farin, J. Hoschek, M.-S. Kim, Handbook of Computer Aided Geometric Design, North-Holland, ISBN 978-0-444-51104-1, August 2002
- [13] R. T. Farouki, Optimal parameterizations, Computer Aided Geometric Design, Volume 14, Issue 2, February 1997, Pages 153–168
- [14] R. T. Farouki, T. Sakkalis, Rational space curves are not "unit speed", Computer Aided Geometric Design, Volume 24, Issue 4, May 2007, Pages 238–240
- [15] M. S. Floater, T. Surazhsky, Parameterization for curve interpolation, in Studies in computational mathematics Vol 12, ed. Jetter, Buhmann, Hausmann, Schaback, Stöckler; ISBN:0444518444 Elsevier Science Inc. New York, NY, USA 2005
- [16] T. A. Grandine, T. A. Hogan, A Parametric Quartic Spline Interpolant to Position, Tangent and Curvature, Geometric Modelling, Springer Vienna, 2004, pp 65-78
- [17] E. Hartmann: Parametric G^n blending of curves and surfaces, The Visual Computer, Volume 17, Issue 1, pp 1-13, 2001.
- [18] T. Hermann, G. Lukács, F.-E. Wolter, Geometrical criteria on the higher order smoothness of composite surfaces, Computer Aided Geometric Design, Volume 16, Issue 9, October 1999, Pages 907–911
- [19] G. Jaklič, J. Kozak, M. Krajnc, V. Vitrih, and E. Zagar, *Hermite Geometric Interpolation by Rational Bézier Spatial Curves*, SIAM J. Numer. Anal., 50(5), 2695–2715. (21 pages)

- [20] B. Jüttler, A vegetarian approach to optimal parameterizations, Computer Aided Geometric Design Volume 14, Issue 9, December 1997, Pages 887–890
- [21] M. Kallay, A geometric Newton-Raphson strategy. Computer Aided Geometric Design 18(8):797-803 (2001)
- [22] G. Klár, G. Valasek, Employing Pythagorean Hodograph Curves for Artistic Patterns, Acta Cybernetica, Volume 20, number 1, pages 101-110, 2011
- [23] G. Klár, G. Valasek, A design element creator for smoothly curving patterns, Poster at Computational Aesthetics in Graphics, Visualization, and Imaging 2010, London, 2010.
- [24] H. K. Joo, T. Yazaki, M. Takezawa, T. Maekawa: Differential geometry properties of lines of curvature of parametric surfaces and their visualization, Graphical Models, Volume 76, Issue 4, Pages 224-238, Academic Press, 2014
- [25] J. Kozak, M. Krajnc, Geometric interpolation by planar cubic G¹ splines, BIT Numerical Mathematics, September 2007, Volume 47, Issue 3, pp 547-563
- [26] R. Klass, An offset spline approximation for plane cubic splines, Computer-Aided Design, Volume 15, Issue 5, September 1983, Pages 297–299
- [27] E. Kreyszig, *Differential Geometry*, Dover Books on Mathematics, Dover Publications, 1991, ISBN-10: 0486667219
- [28] M. Krajnc, Geometric Hermite interpolation by cubic G¹ splines, Nonlinear Analysis: Theory, Methods and Applications, Volume 70, Issue 7, 1 April 2009, Pages 2614–2626
- [29] J. Pegna, F.-E. Wolter, Geometrical Criteria to Guarantee Curvature Continuity of Blend Surfaces, J. Mech. Des 114(1), 201-210 (Mar 01, 1992) (10 pages)

- [30] B.R. Piper, Visually smooth interpolation with triangular Bézier patches.
 In: Farin, G. (Ed.), Geometric Modeling: Algorithms and New Trends.
 SIAM, Philadelphia, pp. 221–233., 1987
- [31] G. Renner, Interpolation with G2 Cubic Curves. IMA Conference on the Mathematics of Surfaces 1992: 385-391
- [32] R. R. Martin: Principal patches for computational geometry, PhD. Thesis, Cambridge University, Chichester (1982)
- [33] H. P. Moreton, Functional optimization for fair surface design, ACM SIG-GRAPH Computer Graphics, Volume 26 Issue 2, July 1992, Pages 167 -176
- [34] K. Mørken: On geometric interpolation of parametric surfaces, Computer Aided Geometric Design, Volume 22, Issue 9, December 2005, Pages 838–848
- [35] K. Mørken and K. Scherer: A general framework for high-accuracy parametric interpolation, Math. Comput. 66., no. 217 (1997), Pages 237-260.
- [36] T. Nagata, Simple local interpolation of surfaces using normal vectors Comput. Aided Geom. Des. 22, 4 (May 2005), 327-347.
- [37] R. Parent, Computer Animation: Algorithms and Techniques, Third Edition, Morgan Kaufmann, September 12, 2012, ISBN-10: 0124158420
- [38] N. M. Patrikalakis, T. Maekawa, Shape Interrogation for Computer Aided Design and Manufacturing, ISBN 978-3-642-04074-0, Springer-Verlag Berlin Heidelberg, 2002
- [39] I. R. Porteous, Geometric Differentiation, 2nd ed., Cambridge University Press, p. 350, ISBN 978-0-521-00264-6
- [40] T. Sakkalis, R. T. Farouki, L. Vaserstein, Non-existence of rational arc length parameterizations for curves in ℝ³, Journal of Computational and Applied Mathematics, Volume 228, Issue 1, 1 June 2009, Pages 494–497
- [41] R. Schaback, Optimal Geometric Hermite Interpolation of Curves, Mathematical Methods for Curves and Surfaces II, 1998;1-12

- [42] K.-L. Shi, S. Zhang, H. Zhang, J.-H. Yong, J.-G. Sun, J.-C. Paul, *B-spline interpolation to a closed mesh*, Computer-Aided Design, Volume 43, Issue 2, February 2011, Pages 145-160, ISSN 0010-4485, 10.1016/j.cad.2010.10.004.
- [43] M. Spivak, A Comprehensive Introduction to Differential Geometry, Vol. 1, 3rd Edition, 1999, Publish or Perish, ISBN-10: 0914098705
- [44] M. Spivak, A Comprehensive Introduction to Differential Geometry, Vol. 2, 3rd Edition, 1999, Publish or Perish, ISBN-10: 0914098713
- [45] D. J. Struik, Lectures on Classical Differential Geometry, Reading, Mass., Addison-Wesley, 1961, 2nd edition
- [46] E. Süli, D. Mayers, An Introduction to Numerical Analysis, ISBN: 9780521007948, August 2003
- [47] J. B. Rosen, H. Park, J. G., L. Zhang, Accurate Solution to Overdetermined Linear Equations with Errors Using L1 Norm Minimization, Computational Optimization and Applications, Volume 17, Issue 2-3, pp 329-341, 2000-12-01, 0926-6003
- [48] G. Valasek, J. Horváth, A. Jámbori, L. Sallai: Geometric Newton-Raphson methods for Plane Curves, Acta Cybernetica, Volume 21, Number 1, Pages 191–203, 2013.
- [49] G. Valasek, J. Vida Considerations on offsetting plane curves, Egri-Nagy, Attila (ed.) et al., Proceedings of the 8th international conference on applied informatics (ICAI 2010), Eger, Hungary, January 27–30, 2010. 2 Volumes, Pages 203-210 (2012). ISBN 978-963-9894-72-3
- [50] G. Valasek, J. Vida Second Order Geometric Hermite Surface Interpolation, The Mathematics of Surface XIV, ISBN 978-0-905-091-30-3, Pages 277-308, IMA, Edited by Robert J. Cripps, G. Mullineux and M. A. Sabin
- [51] R. C. Veltkamp, Survey of Continuities of Curves and Surfaces, Computer Graphics forum, volume 11 (1994), Pages 93-112.
- [52] A. Vlachos, J. Peters, C. Boyd, J. L. Mitchell, *Curved PN Triangles*, I3D '01 Proceedings of the 2001 symposium on Interactive 3D graphics, Pages 159-166, ISBN:1-58113-292-1

- [53] Wagner, H. M., Linear Programming Techniques for Regression Analysis. Journal of the American Statistical Association, Vol. 54, No. 285 (Mar., 1959), pp. 206-212
- [54] D.J. Walton, D.S. Meek: A generalisation of the Pythagorean hodograph quintic spiral, Journal of Computational and Applied Mathematics, 2004;172(2):271-287
- [55] G. Williams, Overdetermined Systems of Linear Equations, The American Mathematical Monthly, Vol. 97, No. 6 (Jun. - Jul., 1990), pp. 511-513
- [56] Xiuzi Ye, Takashi Maekawa, Differential geometry of intersection curves of two surfaces, Computer Aided Geometric Design, Volume 16 Issue 8, Sept. 1999, Pages 767 - 788

Összefoglaló

Ertekezésem témája nemlineáris geometriai modellek konstrukciója és vizsgálata. Parametrikus reprezentációk esetén ennek megvalósításához szükségessé válik a parametrikus leképezés képének – a reprezentált alakzat – tulajdonságait elválasztani magától a leképezéstől, vagyis az adott paraméterezéstől.

Egy rekurzív összefüggést adtam arra, hogy miképpen módosítja a paraméterezés egy görbe deriváltjainak geometriáját a Frenet kísérő triéderben az ívhossz szerinti deriváltakhoz képest.

Ezen eredmény felhasználásával formalizáltam az általános geometriai Hermite görbeinterpoláció problémáját általános bázisokban felírt görbékre. Megmutattam, hogy rögzített paraméterezés esetén a geometriai invariánsok rekonstrukciója lineáris probléma. Egzisztencia feltételt adtam geometriai Hermite interpoláns létezésére általános bázisokban. Bemutattam egy általános paraméterezésoptimalizálási algoritmust a paraméterezés szabadságfokainak tetszőleges funkcionál szerinti értelemben vett optimális beállítására. Továbbá algoritmusokat mutattam pontos és közelítő geometriai Hermite interpolánsok számítására.

Geometriailag jellemeztem a másodrendű geometriai Hermite feladat általánosítását parametrikus felületekre. Megmutattam, hogy a főgörbületi vonalak differenciálgeometriai mennyiségeivel miképpen jellemezhető egy felület lokális geometriája. Összefüggést adtam ezen mennyiségek és két felület geometriailag folytonos csatlakozásának feltételei között. Ezek felhasználásával kiterjesztettem a görbéknél látott formalizmust felületek általános rendű geometriai Hermite interpolációjára. A görbékhez hasonló módon beláttam pontos megoldások létezésének feltételeit, általános algoritmikus keretet adtam a fennmaradó paraméterezési szabadságfokok segítségével történő paraméterezés optimalizációra. Algoritmusokat mutattam geometriai Hermite felületek konstrukciójára és ezek paraméterezés szerint is folytonos csatlakoztatására.

Summary

Parametric curves and surfaces represent shapes as the image of a mapping. The topic of this thesis, nonlinear geometric models, relies on the separation of the properties of this mapping, the parametrization, from that of the geometry of the shape.

I showed, by using elementary differential geometric formulae, how the effect of parametrization can be quantified on the derivatives of an arbitrary parametrized curve in relation to the arc-length parametrized case.

Applying these results, I investigated the problem of general geometric Hermite interpolation of curves by formulating a basis independent geometric reconstruction equation. I showed that reconstruction of geometric data is linear for fixed parametrization and provided existence conditions of exact reconstruction, irrespective of the basis of curve representation. I presented a general framework to utilize the degrees of freedom of parametrization to optimize the parametrization of a curve, and also provided algorithms for the construction of exact and approximate geometric Hermite interpolant curves.

I extended these results to surfaces. I presented and characterized a case study of second order geometric Hermite surface interpolation. I derived the connection between the differential geometric properties of lines of curvature and the conditions of a higher order geometric continuous join of two surfaces. Based on these, I presented a generalization of second order geometric Hermite interpolation to surfaces of the same structure as in the case of curves. I extended the existence conditions, approximation and parametrization methods to surfaces. Algorithms were presented to construct geometric Hermite interpolant surface patches that can be controlled by user input, or serve as initial parametrizations for parametrization optimization methods.