INFORMATION CODING
Linköping University

Splines

or:

Connecting the Dots

Jens Ogniewski
Information Coding Group
Linköping University

Before we start...

- Some parts won't be part of the exam
  - Basically all that is not described in the book.
  - More specific: parameterization schemes, 3-point-spline, change of control points, color interpolation

- Please feel free to ask questions at any time - via mail or just drop by.

Before we start...

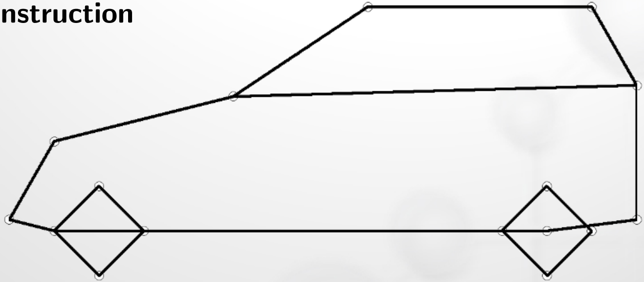
- Take-aways
  - Good interpolation isn't complicated.
  - The best splines are of course the ones that give you the effect you want, with a minimum of resources.

Overview

- Characteristics
- Approximating Splines: Bézier Curves
- Interpolating Splines: Three-Point-Spline, Catmull-Rom
- Parameterization

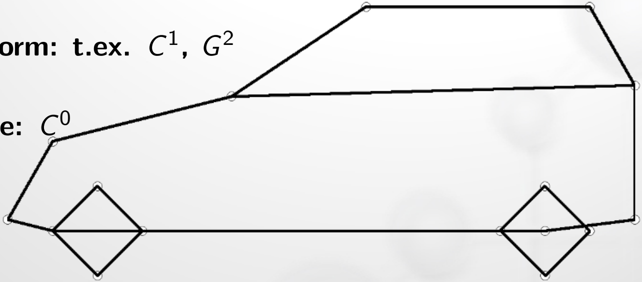
What is a spline

- A curve build from piecewise function
- Defined by control-points
- Historically: a tool to craft smooth curves, used e.g. in ship construction



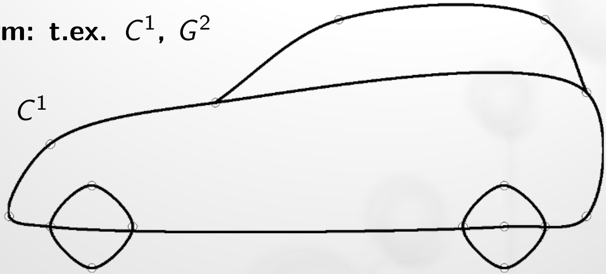
Continuity

- **Which grade?**
  - Which is the highest grade of derivatives where the curves connect?
- **Parametric continuity vs geometric continuity**
- **Short-form: t.ex.** C^1, G^2
- **example:** C^0



Continuity

- **Which grade?**
  - Which is the highest grade of derivatives where the curves connect?
- **Parametric continuity vs geometric continuity**
- **Short-form: t.ex.** C^1, G^2
- **example:** C^1



Continuity

- **Geometric Continuity**
  - Derivative of the functions are vectors
  - If the vector are not the same, but pointing in the same direction, i.e. $\mathbf{P}'_{(u),0} = k\mathbf{P}'_{(u),1}$, $k > 0$, the curve is G continuous

Other characteristics

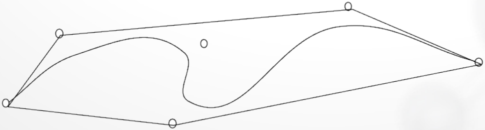
- **Cardinality**
  - Each blending weight is 1 at exactly one control-point, zero at all others, and can be anything in between.
- **Local Control**
  - Each control point only influences a a small, finite part of the overall generated curve.
- **Affine invariant**
  - The sum of all blending weights is always 1.

Approximating vs. Interpolating splines

- **Interpolating: passes all its control points.**
- **Approximating: passes some of its control points.**

Convex hull

- Important property: stay inside convex hull of control-points
- Limits how far the generated curve can stray.

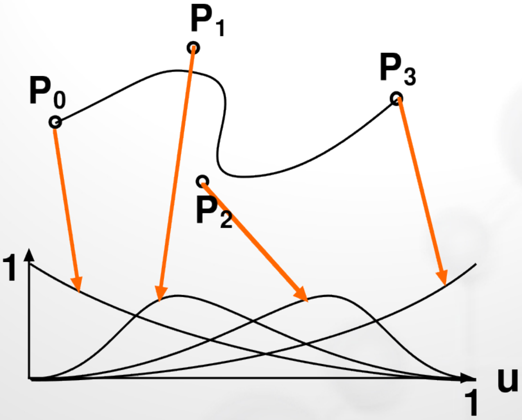


Which spline for what?

- **Surfaces: Approximating, G^1 or G^2.**
- **Movement of objects: interpolating, C^1. Sometimes C^2 needed for acceleration changes.**
- **Movement of camera: interpolating, G^2.**
- **Color: interpolating, C^1.**

Approximating spline: Bézier Curve

- Defined by 4 control points, each using its own blending function.



Approximating spline: Bézier Curve

- **Blending-functions (Bernstein polynomials):**
  $BEZ_{0,3} = (1-u)^3$
  $BEZ_{1,3} = 3u(1-u)^2$
  $BEZ_{2,3} = 3u^2(1-u)$
  $BEZ_{3,3} = u^3$

- $\mathbf{P}_u = \mathbf{P}_0(1-u)^3 + \mathbf{P}_1 3u(1-u)^2 + \mathbf{P}_2 3u^2(1-u) + \mathbf{P}_3 u^3$

Bézier Curve: Characteristics

- **Affine Invariant**
- **Continuity**
  - Two curve segments with points:
    $\mathbf{P}_0,\mathbf{P}_1,\mathbf{P}_2,\mathbf{P}_3$ $\mathbf{P}_3,\mathbf{P}_4,\mathbf{P}_5,\mathbf{P}_6$
  - Derivatives in $\mathbf{P}_3$:
    $3(\mathbf{P}_3 - \mathbf{P}_2)$
    $3(-\mathbf{P}_3 + \mathbf{P}_4)$
    C^1 continuous if $\mathbf{P}_4 = 2\mathbf{P}_3 - \mathbf{P}_2$, i.e. $\mathbf{P}_2$,$\mathbf{P}_3$ and $\mathbf{P}_4$ lie on a line and the distance between $\mathbf{P}_3$ and $\mathbf{P}_4$ is the same than between $\mathbf{P}_2$ and $\mathbf{P}_3$
    If they lie at least on the line (but the distances are different), the curve is G^1 continuous

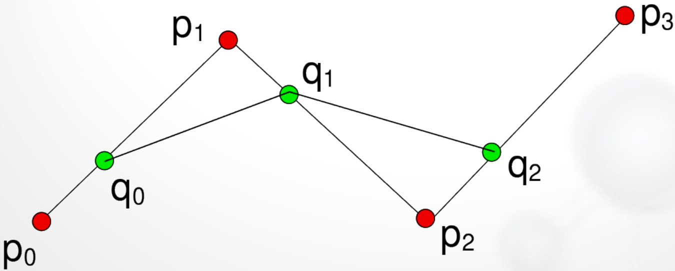
Horner's method

- **Polygon:**
  - $f_{(x)} = ax^3 + bx^2 + cx + d$
  - 3 additions, 6 multiplications

- **Rewrite:**
  - $f_{(x)} = ((ax + b)x + c)x + d$
  - 3 additions, 3 multiplications

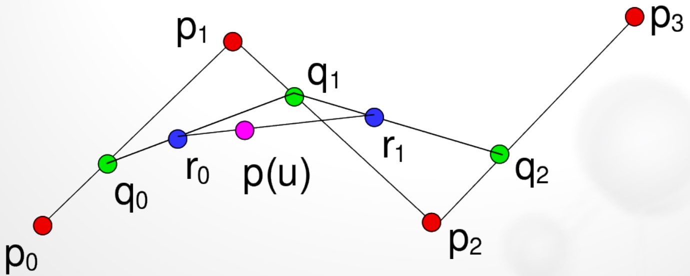
de Casteljau's algorithm

- **Bézier: can be seen as an interpolation of interpolations**



de Casteljau's algorithm

- **Bézier: can be seen as an interpolation of interpolations**

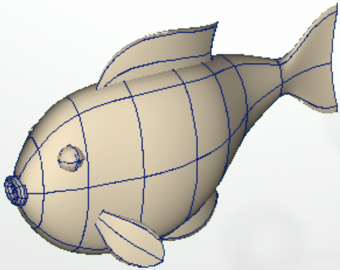


de Casteljau's algorithm

- **Bézier: can be seen as an interpolation of interpolations**
  - Can easily chose which interpolation level we want
  - Note however that this heightens the grade of the polynom!
  - 2 levels: linear
  - 3 levels: cubic (e.g. Bernstein polynomials)
  - More than 4 levels not really useful

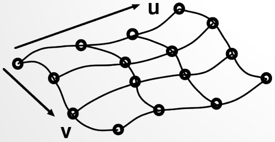
Advanced Approximation Splines

- **B-splines: generalization of Bézier curves**
- **Non-Uniform Rational B-Spline: generalization of B-Spline**
  - Can represent all quadric curves



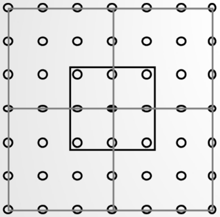
Bézier Surfaces

- Build up from Bézier patches
- Each patch: 4x4 control points



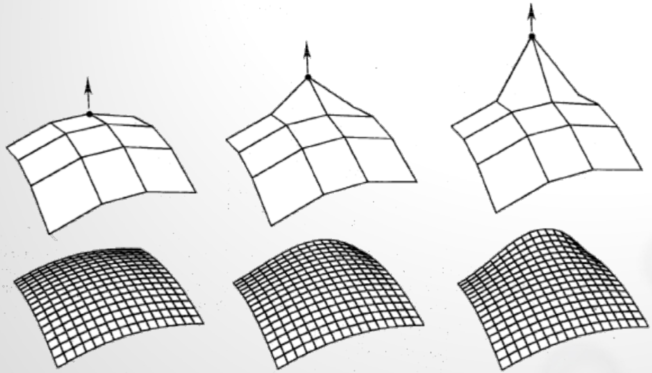
- Blending: calculate a 2-dimensional sum
  $\mathbf{P}_{u,v} = \sum_{j=1}^{4} \sum_{k=1}^{4} BEZ_{j,3(u)} BEZ_{k,3(v)} \mathbf{P}_{jk}$
- Or: first interpolate in x, than in y direction (works in most cases)

Fitting Bézier Patches Together



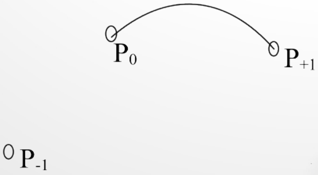
- Start with center point: define point and normal, this spans up a plane
- corner points: have to lie on this plane, can be chosen freely otherwise
- line points: interpolated from corner points

Bézier Surface: Example



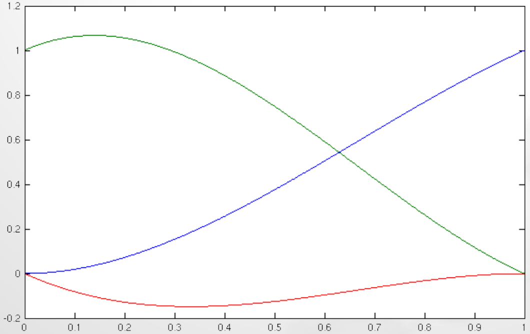
Three-Point-Splines

- Realtime applications: future points not known
- Also: lower complexity as Catmull-Rom
- Calculated from 3 control points, define the curve between the latter two



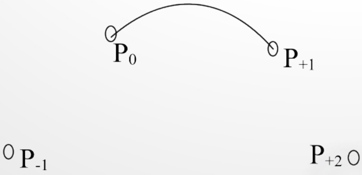
Three-Point-Splines

$(-\alpha u^3 + 2\alpha u^2 - \alpha u)\, P_{-1}$
$(2u^3 + (-3-\alpha)u^2 + \alpha u + 1)\, P_0$
$((\alpha-2)u^3 + (3-\alpha)u^2)\, P_{+1}$



Cardinal Spline

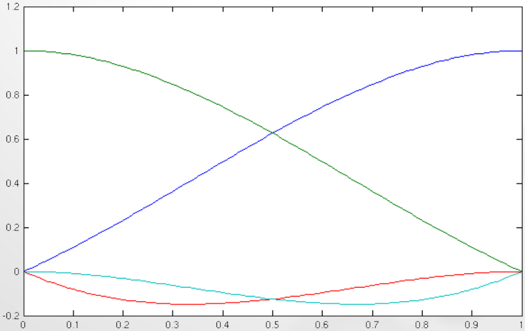
- Specified by control points
- Shape can be varied by a tension parameter t
- Calculated from 4 control points, define the curve between the middle two



- Even called Catmull-Rom (for t=0?)

Cardinal Splines

$(-\alpha u^3 + 2\alpha u^2 - \alpha u)\ P_{-1}$
$((2-\alpha)u^3 + (\alpha-3)u^2 + 1)\ P_0$
$((\alpha-2)u^3 + (3-2\alpha)u^2 + \alpha u)\ P_{+1}$
$(\alpha u^3 - \alpha u^2)\ P_{+2}$



Catmull-Rom Spline ($\alpha = 0.5$)

$$P(u) = [\, u^3 \; u^2 \; u \; 1 \,] \begin{bmatrix} -1/2 & 3/2 & -3/2 & 1/2 \\ 1 & -5/2 & 2 & -1/2 \\ -1/2 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_{k-1} \\ p_k \\ p_{k+1} \\ p_{k+2} \end{bmatrix}$$

$$\begin{aligned} P(u) = \; & p_{k-1} (-u^3/2 + u^2 - u/2) + \\ & p_k (3u^3/2 - 5u^2/2 + 1) + \\ & p_{k+1} (-3u^3/2 + 2u^2 + u/2) + \\ & p_{k+2} (u^3/2 - u^2/2) \end{aligned}$$

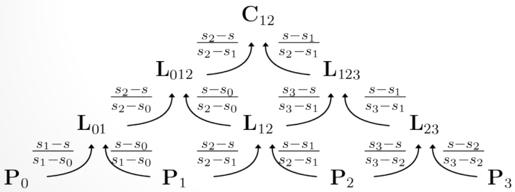
$$= p_{k-1} {}^*CAR_0(u) + p_k {}^*CAR_1(u) + p_{k+1} {}^*CAR_2(u) + p_{k+2} {}^*CAR_3(u)$$

Cardinal Spline: centripetal parameterization

- **Problem: distance and tangent depend on the length of each segment**

- **Can we do something against that?**
  - 1. Find recursive evaluation form
  - 2. Substitute u with $u = \frac{s}{||\mathbf{P_{+1}} - \mathbf{P_0}||^\beta}, 0 <= s <= ||\mathbf{P_{+1}} - \mathbf{P_0}||^\beta$

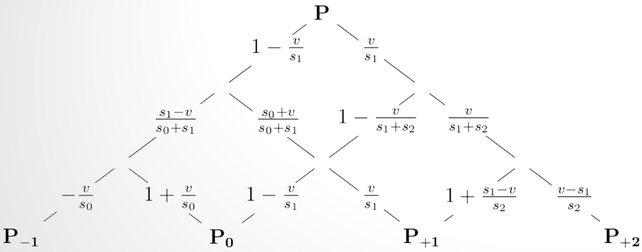
Cardinal Spline: centripetal parameterization



$$S_0 = 0,$$
$$S_{i+1} = |P_i - P_{i+1}|^{0.5} + s_i$$

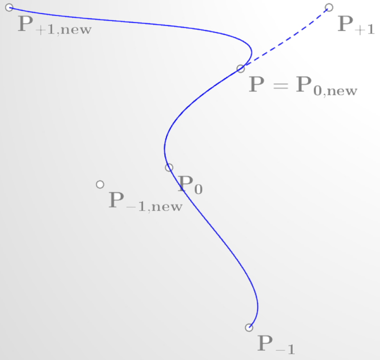
Cem Yuksel, Scott Schaefer, John Keyser: Parameterization and applications of Catmull-Rom curves, 2011

Cardinal Spline: centripetal parameterization



- $s_j = ||P_j - P_{j-1}||^{\beta_j}$
- **slightly less computations, might have higher accuracy**

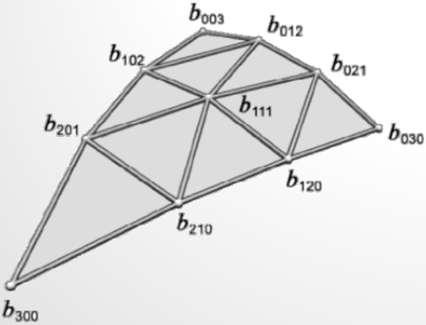
Change Of Control-Points During Ongoing Interpolation



- **Change of the endpoint of an ongoing interpolation in point P:**
  - Using parameterized 3-Point-Spline
  - $\mathbf{P_0} = \mathbf{P}$
  - $\mathbf{P_{+1}} = \mathbf{P_{+1,new}}$, i.e. the new endpoint
  - To maintain C^1 continuity:
  - $\mathbf{P}' = \frac{\mathbf{P}-\mathbf{P_{-1,new}}}{||\mathbf{P}-\mathbf{P_{-1,new}}||^{\beta}}$
  - set $\beta = 0$, i.e. use uniform parameterization for this segment, and $\mathbf{P_{-1,new}} = \mathbf{P} - \mathbf{P}'$

Curved PN Triangles

- **Calculate a set of intermediate points from given vertices and normals**

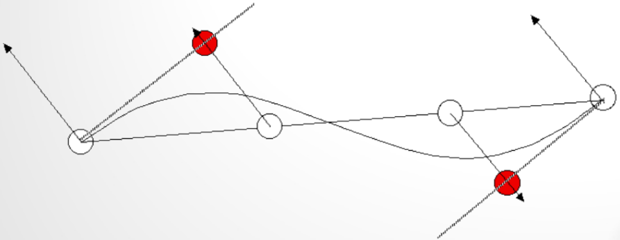


Curved PN Triangles

- **Shape depends on the normals**

Curved PN Triangles

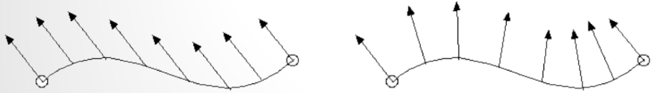
- **Use tangents in equidistant points to project**



INFORMATION CODING
Linköping University

Curved PN Triangles

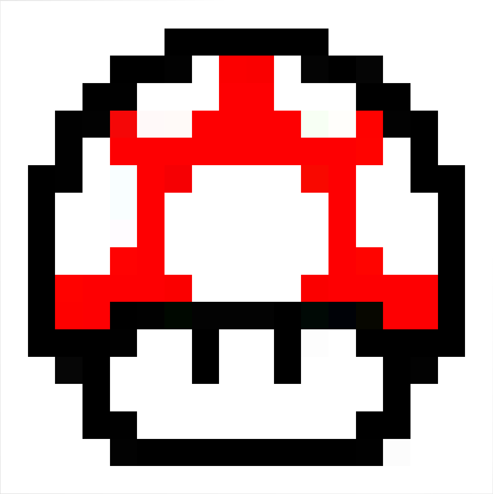
- **Careful: normals can't just be interpolated!**



Color-Interpolation

- **Linear**
  - Used by graphics card for texture interpolation
  - Since it treads direction different: visible lines in x and y direction
- **Cubic**
  - C^1 continuous
  - Since it uses negative blendweights: can lead to clipping artifacts, discoloring
- **Trigonometric**
  - C^1 continuous
  - Based on cosine and since functions
  - Since the GPU has hardware accelerators for that, sine and cosine only take one clock cycle
  - No overshoots or clipping, but ringing artifacts

Color-Interpolation: Input



Color-Interpolation: Linear

Color-Interpolation: Cubic (Catmull-Rom)

Color-Interpolation: Trigonometric

Conclusion

- Splines useful for interpolation of animation, camera parameter, surface generation, color interpolation, ...
- Approximating and Interpolating splines, parameterization (the later will not be included in the exam)
- Take-aways
  - Good interpolation isn't complicated.
  - The best splines are of course the ones that give you the effect you want, with a minimum of resources.

Thank you very much!

www.icg.isy.liu.se