Fast and Stable Evaluation of Box-Splines via the BB-Form

Minho Kim and Jörg Peters

`http://www.cise.ufl.edu/research/SurfLab`

University of Florida

10th SIAM Conference on Geometric Design & Computing

Motivation

Motivation

- Box-splines for generating continuous field from discrete data.

Motivation

- Box-splines for generating continuous field from discrete data.
- Recursive evaluation (de Boor ’93 and Kobbelt ’97) can evaluate arbitrary box-splines

Motivation

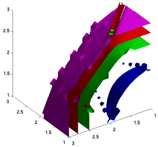
- Box-splines for generating continuous field from discrete data.
- Recursive evaluation (de Boor ’93 and Kobbelt ’97) can evaluate arbitrary box-splines but are too slow.

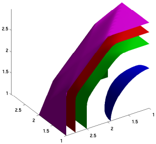
Motivation

- Box-splines for generating continuous field from discrete data.
- Recursive evaluation (de Boor '93 and Kobbelt '97) can evaluate arbitrary box-splines but are too slow.
- Error in tri-variate box-splines.

Motivation

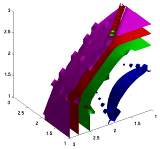
- Box-splines for generating continuous field from discrete data.
- Recursive evaluation (de Boor ’93 and Kobbelt ’97) can evaluate arbitrary box-splines but are too slow.
- Error in tri-variate box-splines.

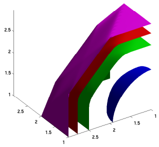




Motivation

- Box-splines for generating continuous field from discrete data.
- Recursive evaluation (de Boor '93 and Kobbelt '97) can evaluate arbitrary box-splines but are too slow.
- Error in tri-variate box-splines.





- Existing methods via BB-form (Chui et al. '91 and Casciola et al. '06) evaluate only specific box-splines.

Box-spline

$$M_{\Xi}$$

Box-spline

$M_{\boxed{\Xi}}$

direction matrix

Example

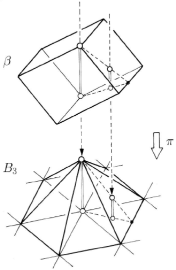
$$\Xi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_{\Xi}$$

Example

$$M_{\Xi}$$

$$\Xi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

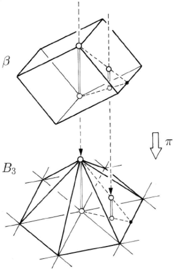


(image courtesy of Prautzsch et al.)

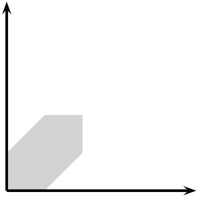
Support

$$\Xi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

M_{Ξ}



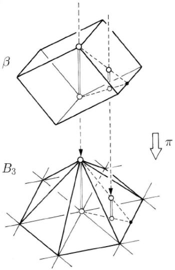
(image courtesy of Prautzsch et al.)



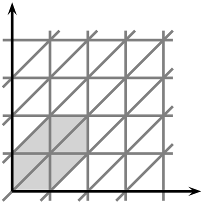
Knot planes

$$\Xi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_{\Xi}$$



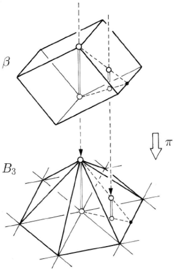
(image courtesy of Prautzsch et al.)



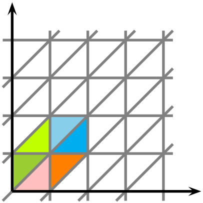
Piecewise polynomial

$$\Xi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_\Xi$$



(image courtesy of Prautzsch et al.)



Spline

$$
\begin{tikzcd}
M_{\Xi} \arrow[d] & \\
* \arrow[r] & \displaystyle\sum_{j\in\mathbb{Z}^s} a(j)\, M_{\Xi}(\cdot - j) \\
a \arrow[u] &
\end{tikzcd}
$$

Evaluation methods

approximate exact

M_{Ξ}

Evaluation methods

approximate exact

- subdivision

M_Ξ

Evaluation methods

	approximate	exact
M_{Ξ}	▶ subdivision ▶ sampling & interpolation	

Evaluation methods

	approximate	exact
M_{Ξ}	- subdivision - sampling & interpolation - inverse FFT	

Evaluation methods

	approximate	exact
M_{Ξ}	- subdivision - sampling & interpolation - inverse FFT	de Boor '93

Evaluation methods

	approximate	exact
M_{Ξ}	- subdivision - sampling & interpolation - inverse FFT	de Boor '93 Kobbelt '97

Evaluation methods

M_{Ξ}

	approximate	exact
	- subdivision - sampling & interpolation - inverse FFT	- recursive de Boor ’93 Kobbelt ’97

Evaluation methods

	approximate	exact
M_{Ξ}	▶ subdivision ▶ sampling & interpolation ▶ inverse FFT	▶ recursive de Boor ’93 Kobbelt ’97 ▶ BB-form

Conversion to BB-form

$$M_{\Xi} = \sum c_{\alpha} b_{\alpha} \left(\beta_{\sigma} \left(\cdot \right) \right)$$

Conversion to BB-form

$$M_{\Xi} = \sum c_\alpha b_\alpha \left(\boxed{\beta_\sigma}(\cdot)\right)$$

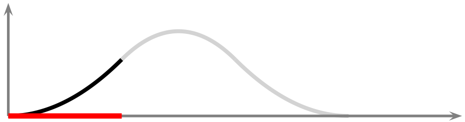
barycentric coordinate function
w.r.t. the domain simplex σ

Conversion to BB-form

$$M_{\Xi} = \sum c_\alpha \boxed{b_\alpha} (\beta_\sigma (\cdot))$$

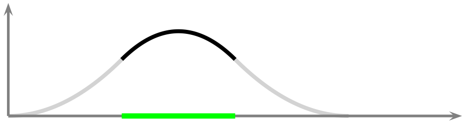
Bernstein basis polynomial

Conversion to BB-form



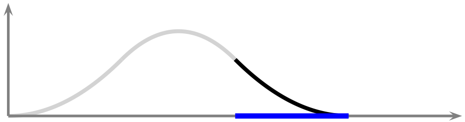
$$M_{\Xi} = \sum c_{\alpha} b_{\alpha} \left(\beta_{\sigma} \left(\cdot \right) \right)$$

Conversion to BB-form



$$M_{\Xi} = \sum c_{\alpha} b_{\alpha} \left(\beta_{\sigma}\left(\cdot\right)\right)$$

Conversion to BB-form



$$M_{\Xi} = \sum c_{\alpha} b_{\alpha} \left(\beta_{\sigma} \left(\cdot \right) \right)$$

Conversion to BB-form

$$M_{\Xi} = \sum \boxed{c_\alpha} b_\alpha\left(\beta_\sigma\left(\cdot\right)\right)$$

How to compute the coefficients?

Conversion to BB-form

$$M_{\Xi} = \sum c_\alpha b_\alpha \left(\beta_{\boxed{\sigma}}(\cdot)\right)$$

How to index a polynomial piece (domain simplex σ)?

Computing coefficients

M_{Ξ} $\qquad\qquad$ c_α

Computing coefficients

$$M_{\Xi}(\boxed{x_i}) = \sum c_\alpha b_\alpha \left(\beta_\sigma \left(\boxed{x_i}\right)\right)$$

sample points

sample points

Computing coefficients

$$M_{\Xi}(x_i) = \sum c_{\alpha} b_{\alpha}\left(\beta_{\sigma}\left(x_i\right)\right)$$

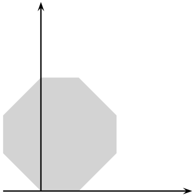
Theorem

Let $\Xi \in \mathbb{Z}^{s \times n}$ *and* $\operatorname{rank}(\Xi) = s$. *Then the polynomial pieces of* M_{Ξ} *can be represented in BB-form with coefficients in* $\mathbb{Q}$.

Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

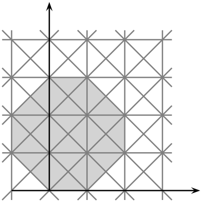
M_{Ξ}



Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

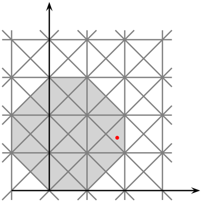
M_Ξ



Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

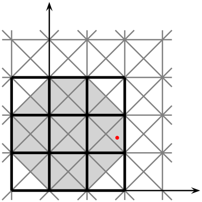
M_Ξ



Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

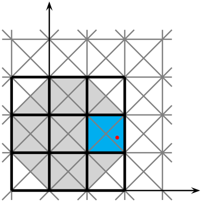
M_Ξ



Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

M_Ξ

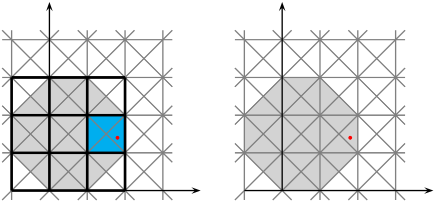


$\mathbf{i}_\square$

Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

M_Ξ

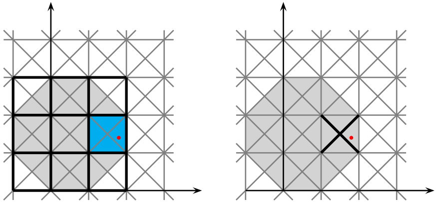


$\mathbf{i}_\square$

Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

M_{Ξ}

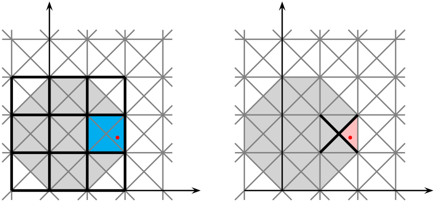


$\mathbf{i}_{\square}$

Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

M_Ξ

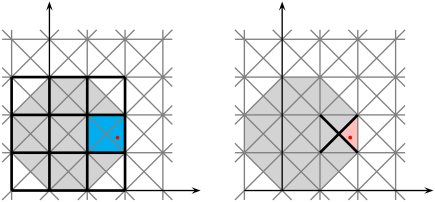


$\mathbf{i}_\square$ $\mathbf{i}_\triangle$

Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

M_Ξ

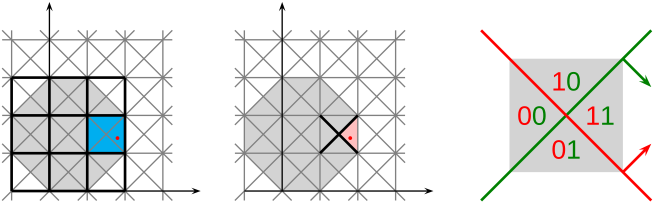


$\mathbf{i}_\square$ $\quad\quad$ $\mathbf{i}_\triangle := U\left(N_\Xi\left(\cdot - \lfloor\cdot\rfloor\right) - \eta_\Xi\right)$

Indexing polynomial piece (domain simplex)

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

M_{Ξ}



$\mathbf{i}_{\square}$

$\mathbf{i}_{\triangle} := U\left(N_{\Xi}\left(\cdot - \lfloor \cdot \rfloor\right) - \eta_{\Xi}\right)$

Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$\text{EVALUATESPLINE}_{\Xi}(a, x)$

$i_{\triangle} \leftarrow U\left(N_{\Xi}(x - \lfloor x \rfloor) - \eta_{\Xi}\right)$

$u \leftarrow \text{COMPUTEBARYCENTRIC}(i_{\triangle}, x - \lfloor x \rfloor)$

$P \leftarrow \sum_{i_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rfloor - i_{\square}) C_{\Xi}(i_{\square}, i_{\triangle})$

return $\text{EVALUATEBÉZIER}(P, u)$

Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$\textsc{EvaluateSpline}_{\Xi}\,(a, x)$

$\mathrm{i}_{\triangle} \leftarrow U\left(N_{\Xi}\left(x - \lfloor x \rfloor\right) - \eta_{\Xi}\right)$

$u \leftarrow \textsc{ComputeBarycentric}(\mathrm{i}_{\triangle}, x - \lfloor x \rfloor)$

$P \leftarrow \sum_{\mathrm{i}_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rfloor - \mathrm{i}_{\square}) C_{\Xi}(\mathrm{i}_{\square}, \mathrm{i}_{\triangle})$

return $\textsc{EvaluateBézier}(P, u)$

Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$\textsc{EvaluateSpline}_{\Xi}\,(a, x)$

$i_{\triangle} \leftarrow U\left(N_{\Xi}\left(x - \lfloor x \rfloor\right) - \eta_{\Xi}\right)$

$u \leftarrow \textsc{ComputeBarycentric}(i_{\triangle}, x - \lfloor x \rfloor)$

$P \leftarrow \sum_{i_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rfloor - i_{\square}) C_{\Xi}(i_{\square}, i_{\triangle})$

return $\textsc{EvaluateBézier}(P, u)$

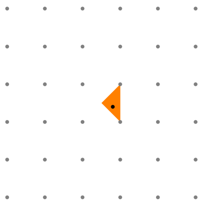
Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

EVALUATESPLINE$_\Xi$ (a, x)

$i_\triangle \leftarrow U\left(N_\Xi\left(x - \lfloor x \rfloor\right) - \eta_\Xi\right)$

$u \leftarrow$ COMPUTEBARYCENTRIC$(i_\triangle, x - \lfloor x \rfloor)$



$P \leftarrow \sum_{i_\square \in \mathbf{I}_\Xi} a(\lfloor x \rfloor - i_\square) C_\Xi(i_\square, i_\triangle)$

return EVALUATEBÉZIER(P, u)

Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

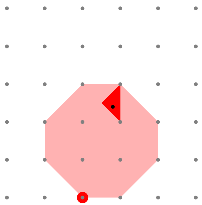
$\textsc{EvaluateSpline}_{\Xi}(a, x)$

$\mathrm{i}_{\triangle} \leftarrow U\left(N_{\Xi}\left(x - \lfloor x \rfloor\right) - \eta_{\Xi}\right)$

$u \leftarrow \textsc{ComputeBarycentric}(\mathrm{i}_{\triangle}, x - \lfloor x \rfloor)$

$P \leftarrow \sum_{\mathrm{i}_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rfloor - \mathrm{i}_{\square}) C_{\Xi}(\mathrm{i}_{\square}, \mathrm{i}_{\triangle})$

return $\textsc{EvaluateBézier}(P, u)$



Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

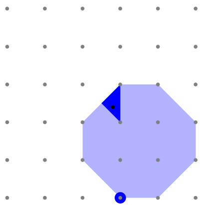
$\text{EVALUATESPLINE}_{\Xi}\,(a,x)$

$\mathrm{i}_{\triangle} \leftarrow U\left(N_{\Xi}\left(x - \lfloor x \rfloor\right) - \eta_{\Xi}\right)$

$u \leftarrow \text{COMPUTEBARYCENTRIC}(\mathrm{i}_{\triangle}, x - \lfloor x \rfloor)$

$P \leftarrow \sum_{\mathrm{i}_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rfloor - \mathrm{i}_{\square}) C_{\Xi}(\mathrm{i}_{\square}, \mathrm{i}_{\triangle})$

return $\text{EVALUATEBÉZIER}(P, u)$



Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

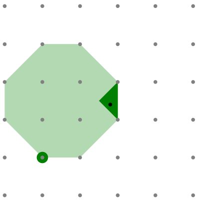
$\textsc{EvaluateSpline}_{\Xi}(a, x)$

$\mathrm{i}_{\triangle} \leftarrow U\left(N_{\Xi}\left(x - \lfloor x \rfloor\right) - \eta_{\Xi}\right)$

$u \leftarrow \textsc{ComputeBarycentric}(\mathrm{i}_{\triangle}, x - \lfloor x \rfloor)$

$P \leftarrow \sum_{\mathrm{i}_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rfloor - \mathrm{i}_{\square}) C_{\Xi}(\mathrm{i}_{\square}, \mathrm{i}_{\triangle})$

return $\textsc{EvaluateBézier}(P, u)$



Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

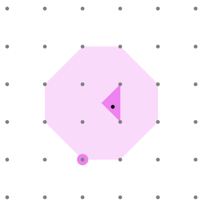
$\textsc{EvaluateSpline}_{\Xi}(a, x)$

$\mathrm{i}_{\triangle} \leftarrow U\left(N_{\Xi}\left(x - \lfloor x \rfloor\right) - \eta_{\Xi}\right)$

$u \leftarrow \textsc{ComputeBarycentric}(\mathrm{i}_{\triangle}, x - \lfloor x \rfloor)$

$P \leftarrow \sum_{\mathrm{i}_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rfloor - \mathrm{i}_{\square}) C_{\Xi}(\mathrm{i}_{\square}, \mathrm{i}_{\triangle})$

return $\textsc{EvaluateBézier}(P, u)$



Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

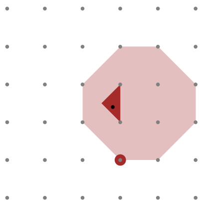
$\text{EVALUATESPLINE}_{\Xi}\,(a, x)$

$\mathrm{i}_{\triangle} \leftarrow U\left(N_{\Xi}\left(x - \lfloor x \rceil\right) - \eta_{\Xi}\right)$

$u \leftarrow \text{COMPUTEBARYCENTRIC}(\mathrm{i}_{\triangle}, x - \lfloor x \rceil)$

$P \leftarrow \sum_{\mathrm{i}_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rceil - \mathrm{i}_{\square}) C_{\Xi}(\mathrm{i}_{\square}, \mathrm{i}_{\triangle})$

return $\text{EVALUATEBÉZIER}(P, u)$



Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

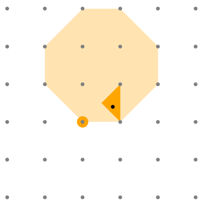
$\textsc{EvaluateSpline}_{\Xi}\,(a, x)$

$\mathrm{i}_{\triangle} \leftarrow U\left(N_{\Xi}\left(x - \lfloor x \rceil\right) - \eta_{\Xi}\right)$

$u \leftarrow \textsc{ComputeBarycentric}(\mathrm{i}_{\triangle}, x - \lfloor x \rceil)$

$P \leftarrow \sum_{\mathrm{i}_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rceil - \mathrm{i}_{\square}) C_{\Xi}(\mathrm{i}_{\square}, \mathrm{i}_{\triangle})$

return $\textsc{EvaluateBézier}(P, u)$



Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

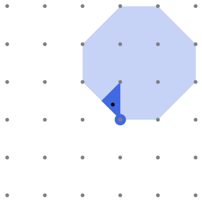
$\textsc{EvaluateSpline}_{\Xi}\,(a,x)$

$i_{\triangle} \leftarrow U\left(N_{\Xi}\left(x - \lfloor x \rfloor\right) - \eta_{\Xi}\right)$

$u \leftarrow \textsc{ComputeBarycentric}(i_{\triangle}, x - \lfloor x \rfloor)$

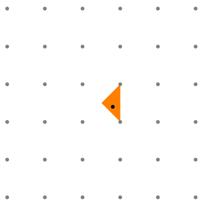
$P \leftarrow \sum_{i_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rfloor - i_{\square}) C_{\Xi}(i_{\square}, i_{\triangle})$

return $\textsc{EvaluateBézier}(P, u)$



Spline evaluation

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$



$\textsc{EvaluateSpline}_{\Xi}(a, x)$

$i_{\triangle} \leftarrow U\left(N_{\Xi}(x - \lfloor x \rfloor) - \eta_{\Xi}\right)$

$u \leftarrow \textsc{ComputeBarycentric}(i_{\triangle}, x - \lfloor x \rfloor)$

$P \leftarrow \sum_{i_{\square} \in \mathbf{I}_{\Xi}} a(\lfloor x \rfloor - i_{\square}) C_{\Xi}(i_{\square}, i_{\triangle})$

return $\textsc{EvaluateBézier}(P, u)$

6-directional tri-variate box-spline M_{Ξ_6} on the FCC lattice

6-directional tri-variate box-spline M_{Ξ_6} on the FCC lattice

- defined by the direction matrix

$$\Xi_6 := \begin{bmatrix} 1 & -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{bmatrix}$$

6-directional tri-variate box-spline M_{Ξ_6} on the FCC lattice

- defined by the direction matrix

$$\Xi_6 := \begin{bmatrix} 1 & -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{bmatrix}$$

- piecewise polynomial of degree ≤ 3

6-directional tri-variate box-spline M_{Ξ_6} on the FCC lattice

- defined by the direction matrix

$$\Xi_6 := \begin{bmatrix} 1 & -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{bmatrix}$$

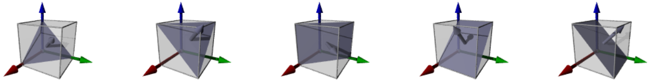
- piecewise polynomial of degree ≤ 3
- equivalent to $M_{\tilde{\Xi}_6}$ on the Cartesian lattice with

$$\tilde{\Xi}_6 := \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

6-directional tri-variate box-spline M_{Ξ_6} on the FCC lattice (cont'd)

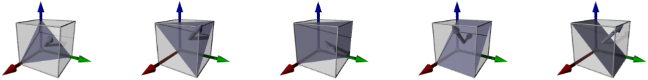
6-directional tri-variate box-spline M_{Ξ_6} on the FCC lattice (cont'd)

- knot planes of $M_{\tilde{\Xi}_6}$ of in $[0..1)^3$

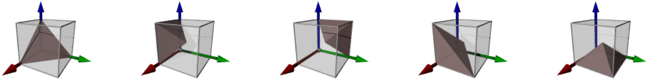


6-directional tri-variate box-spline M_{Ξ_6} on the FCC lattice (cont'd)

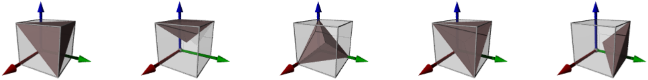
- knot planes of $M_{\tilde{\Xi}_6}$ of in $[0..1)^3$



- polynomial pieces (domain tetrahedra) of $M_{\tilde{\Xi}_6}$ in $[0..1)^3$

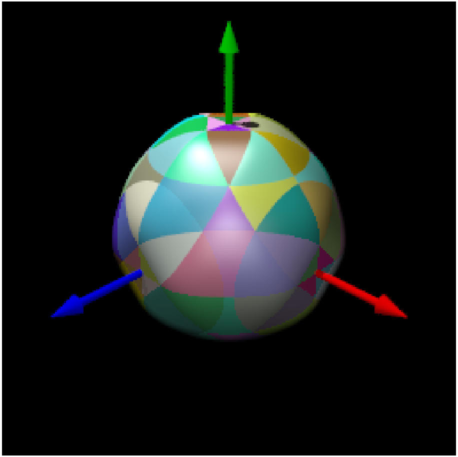


00000 10010 10001 10110 10100



11111 10011 10000 10111 10101

6-directional tri-variate box-spline M_{Ξ_6} on the FCC lattice (cont'd)



6-directional tri-variate box-spline M_{Ξ_6} on the FCC lattice (cont'd)

Loading `6dirfast.mpeg`...

7-directional tri-variate box-spline M_{Ξ_7}

7-directional tri-variate box-spline M_{Ξ_7}

- defined by the direction matrix

$$\Xi_7 := \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 1 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

7-directional tri-variate box-spline M_{Ξ_7}

- defined by the direction matrix

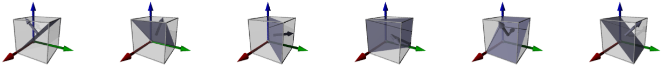
$$\Xi_7 := \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 1 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

- piecewise polynomial of degree ≤ 4

7-directional tri-variate box-spline M_{Ξ_7} (cont'd)

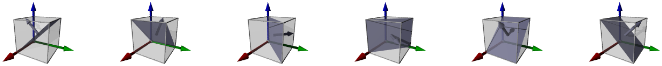
7-directional tri-variate box-spline M_{Ξ_7} (cont'd)

- knot planes in $[0..1)^3$

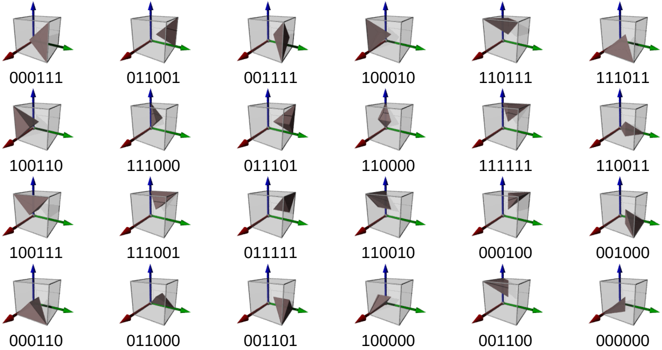


7-directional tri-variate box-spline M_{Ξ_7} (cont'd)

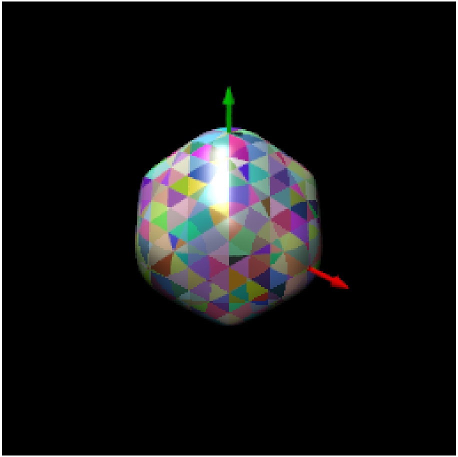
- knot planes in $[0..1)^3$



- polynomial pieces (domain tetrahedra) in $[0..1)^3$



7-directional tri-variate box-spline M_{Ξ_7} (cont'd)



7-directional tri-variate box-spline M_{Ξ_7} (cont'd)

Loading `7dirfast.mpeg`...

Performance

algorithm	spline	resolution 21^3	31^3	41^3
de Boor	M_{Ξ_7}	20.273238 ×**144**	75.297004 ×**154**	187.711522 ×**153**
	M_{Ξ_6}	1.860688 ×**34**	7.087524 ×**39**	18.147211 ×**41**
Kobbelt	M_{Ξ_7}	52.727976 ×**375**	207.840594 ×**424**	550.422698 ×**450**
	M_{Ξ_6}	3.644995 ×**66**	14.034635 ×**78**	37.232097 ×**84**
via BB-form	M_{Ξ_7}	0.140722	0.489674	1.223360
	M_{Ξ_6}	0.055346	0.180976	0.444804

(evaluation of vectorized input by MATLAB®)

- time measured in secs
- BB-form method is ×ratio times faster

High-quality image generation using ray-tracer

Loading `quasi.mpeg`...

Thank you!

- The MATLAB® package can be downloaded at `http://www.cise.ufl.edu/research/SurfLab/tribox`

References

- Carl de Boor, *On the evaluation of box splines*, Numerical Algorithms **5** (1993), no. 1–4, 5–23.
- Gerald Farin, Josef Hoschek, and Myung-Soo Kim (eds.), *The handbook of computer aided geometric design*, 3rd ed., ch. Box Splines, pp. 255–282, Elsevier, Amsterdam, 2002.
- Minho Kim and Jörg Peters, *Fast and stable evaluation of box-splines via the Bézier form*, Tech. Report REP-2007-422, University of Florida, 2007.
- Leif Kobbelt, *Stable evaluation of box-splines*, Numerical Algorithms **14** (1997), no. 4, 377–382.

Spline on non-Cartesian lattice

A spline can also be generated on the non-Cartesian lattice $X^{-1}\mathbb{Z}^s$ spanned by M_Ξ with the coefficients $b : X^{-1}\mathbb{Z}^s \to \mathbb{R}$ (change of variables):

$$\sum_{j \in X^{-1}\mathbb{Z}^s} M_\Xi(\cdot - j)\, b(j) = |\det X| \sum_{j \in \mathbb{Z}^s} M_{X\Xi}(X \cdot - j)\, b(X^{-1} j).$$