Seminarium WEiI Politechniki Koszalińskiej, 11 grudnia 2018

USING B-SPLINES FOR CURVE AND SURFACE APPROXIMATION

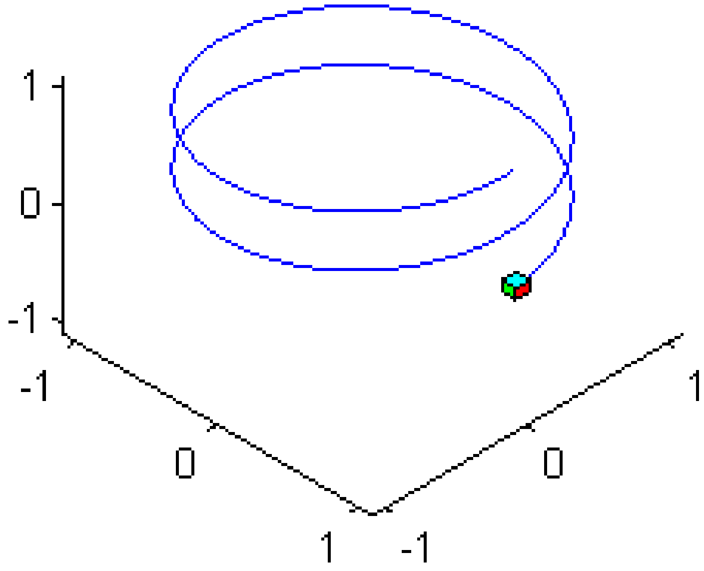
dr inż. Robert Świta
Zakład Systemów Multimedialnych i Sztucznej Inteligencji
robert.swita@wp.pl, p.216A

Source materials

- **R.Świta, Z.Suszyński, „**Image approximation using B-spline surfaces", 21st European Microelectronics and Packaging Conference, Warsaw University of Technology, 10-13 September 2017 POLAND
- **R.Świta, Z.Suszyński, „**Thermal Image Approximation Using B-Spline Surfaces", International Journal of Thermophysics (2018) 39(11):127
- Udział wykonawczy w projekcie realizowanym przez Politechnikę Krakowską „Bezwykopowy przestrzenny system monitoringu i detekcji przecieków, erozji i przemieszczeń" w Programie Operacyjnym Inteligentny Rozwój 2014-2020 nr: POIR.01.01-00-0501/16-00

$$x(t) = r\cos(t)$$
$$y(t) = r\sin(t)$$
$$z(t) = ht$$

Parametric curves



$$C(t) = C[x(t), y(t), z(t), \ldots]$$

$$C(t) = \sum_{i=0}^{b-1} B_i(t) P_i$$

$$\boldsymbol{C} = \boldsymbol{B}\boldsymbol{P}$$

$$\boldsymbol{B}^T\boldsymbol{C} = \boldsymbol{B}^T\boldsymbol{B}\boldsymbol{P}$$

$$\boldsymbol{P} = \left(\boldsymbol{B}^T\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{C} = \boldsymbol{B}^{\dagger}\boldsymbol{C}$$

$$\begin{bmatrix} C_0.x & C_0.y & \cdots \\ \ldots & \ldots & \cdots \\ C_{n-1}.x & C_{n-1}.y & \cdots \end{bmatrix} = \begin{bmatrix} B_0(t_0) & \ldots & B_{b-1}(t_0) \\ \ldots & \ldots & \ldots \\ B_0(t_{n-1}) & \ldots & B_{b-1}(t_{n-1}) \end{bmatrix} \begin{bmatrix} P_0.x & P_0.y & \cdots \\ \ldots & \ldots & \cdots \\ P_{b-1}.x & P_{b-1}.y & \cdots \end{bmatrix}$$

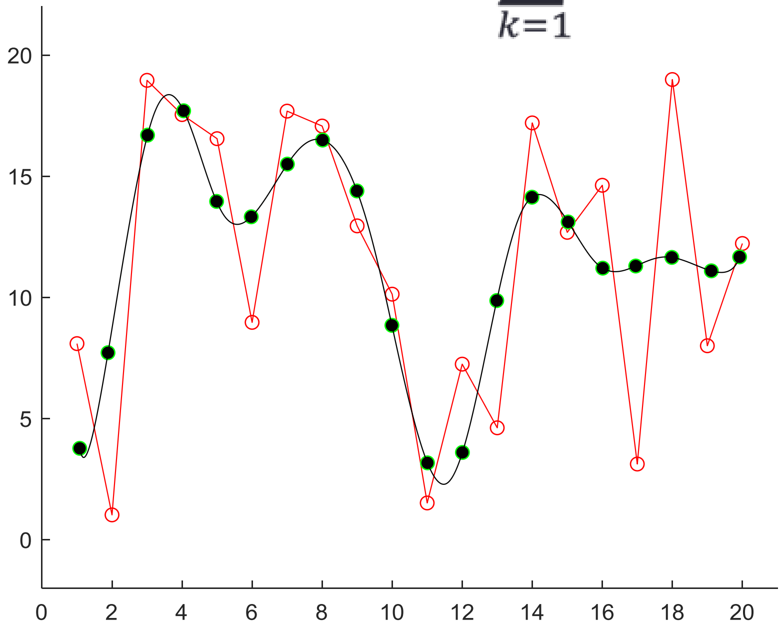
Parameterization

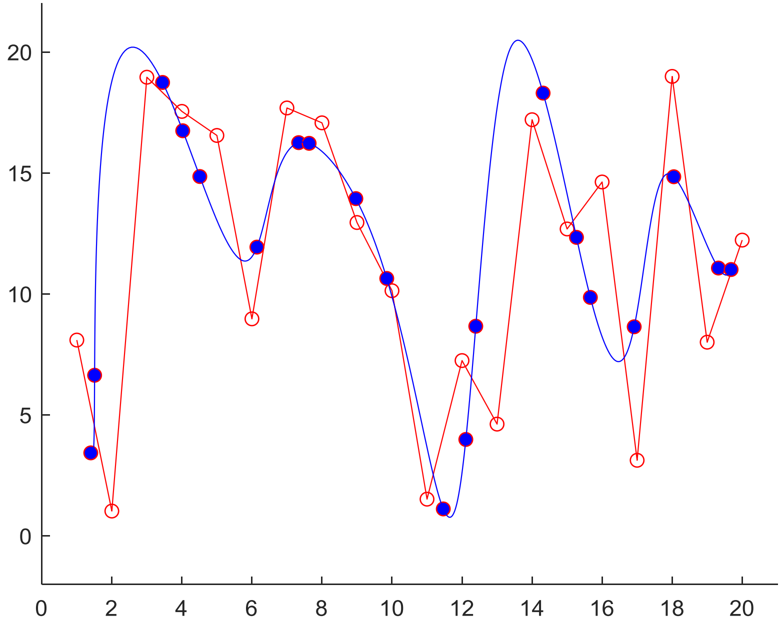
- uniform – parameters are equidistant and taking values of indexes:

$$t_i = i$$

- chordal – parameters are proportional to the length of the curve fragment. In practice one takes sum of the line segments:

$$t_i = \sum_{k=1}^{i} \|C_k - C_{k-1}\| = t_{i-1} + \|C_i - C_{i-1}\|$$





B-splines

The B-spline curves are a parametrically described linear combination of *b* base functions N_i^d, which are piecewise polynomials of degree d:

$$C(t) = \sum_{i=0}^{b-1} P_i \mathrm{N}_i^d(t) \equiv \boldsymbol{C} = \boldsymbol{NP}$$

Parameters t^*, called knots, define connection points of the polynomials forming spline. Base functions have compact and limited suport, which determines local influence of the position of control point P_i on a curve trajectory in the interval $[t_i^*, t_{i+d+1}^*)$. It also means that position of the curve point depends only on $d+1$ control points.

$$\mathrm{N}_i^d(t) \neq 0, t \in [t_i^*, t_{i+d+1}^*)$$

$$\sum_{i=0}^{b-1} \mathrm{N}_i^d(t) = 1$$

Basis functions

Since base functions are also splines, their definition is also recursive:

$$\mathrm{N}_i^d = \alpha_i \mathrm{N}_i^{d-1} + (1-\alpha_{i+1})\mathrm{N}_{i+1}^{d-1}$$

$$\alpha_i = \frac{t - t_i^*}{t_{i+d}^* - t_i^*}$$

We can account for only non-zero terms directly from suport limitation:

$$\mathrm{N}_i^{d-1} \neq 0 \;\Rightarrow\; t \in [t_i^*, t_{i+d}^*)$$
$$\mathrm{N}_{i+1}^{d-1} \neq 0 \;\Rightarrow\; t \in [t_{i+1}^*, t_{i+1+d}^*)$$

Recursive calculation of basis function $\mathbf{N}_i^d$

$\texttt{if}\ (d == 0 \;\&\&\; t_i^* \le t < t_{i+1}^*)\quad \texttt{return}\ \mathrm{N}_i^d = 1$

$\texttt{if}\ (i == b-1 \;\&\&\; t == t_{b+d}^*])\ \texttt{return}\ \mathrm{N}_i^d = 1$

$\mathrm{N}_i^d = 0$

$\texttt{if}\ (t_i^* < t < t_{i+d}^*)\qquad \mathrm{N}_i^d \mathrel{+}= \alpha_i \mathrm{N}_i^{d-1}(t)$

$\texttt{if}\ (t_{i+1}^* \le t < t_{i+1+d}^*)\ \ \mathrm{N}_i^d \mathrel{+}= (1-\alpha_{i+1})\mathrm{N}_{i+1}^{d-1}$

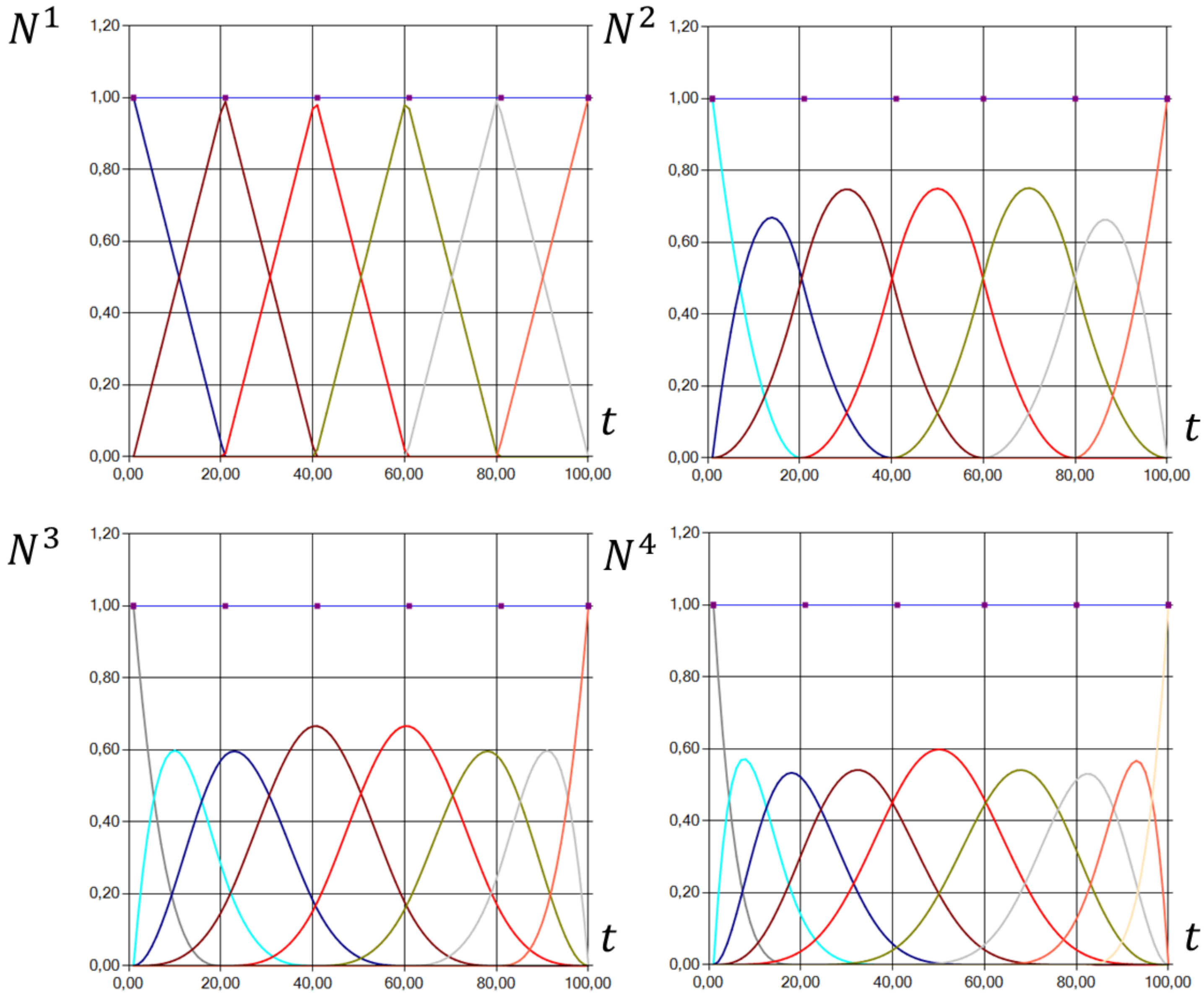
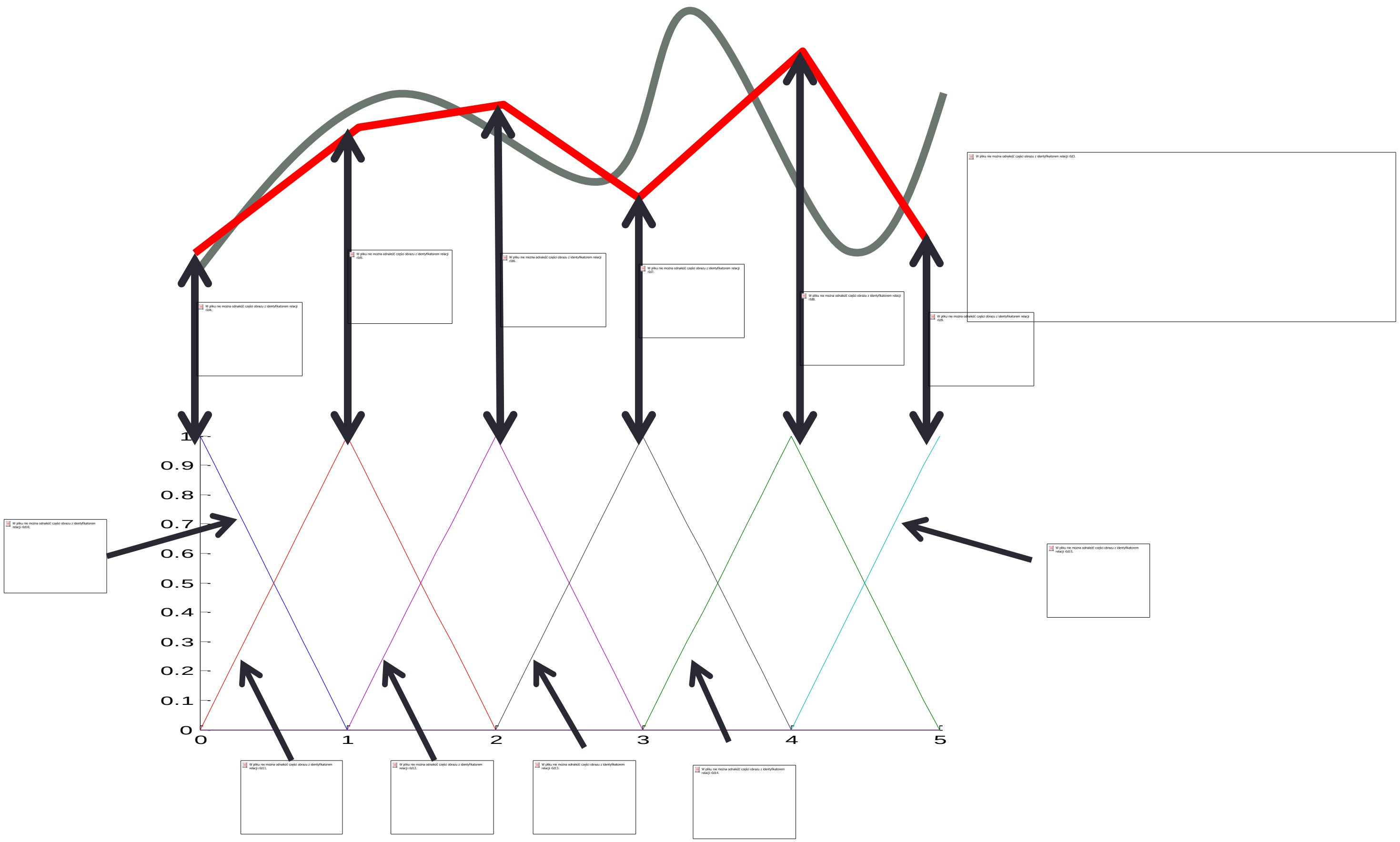
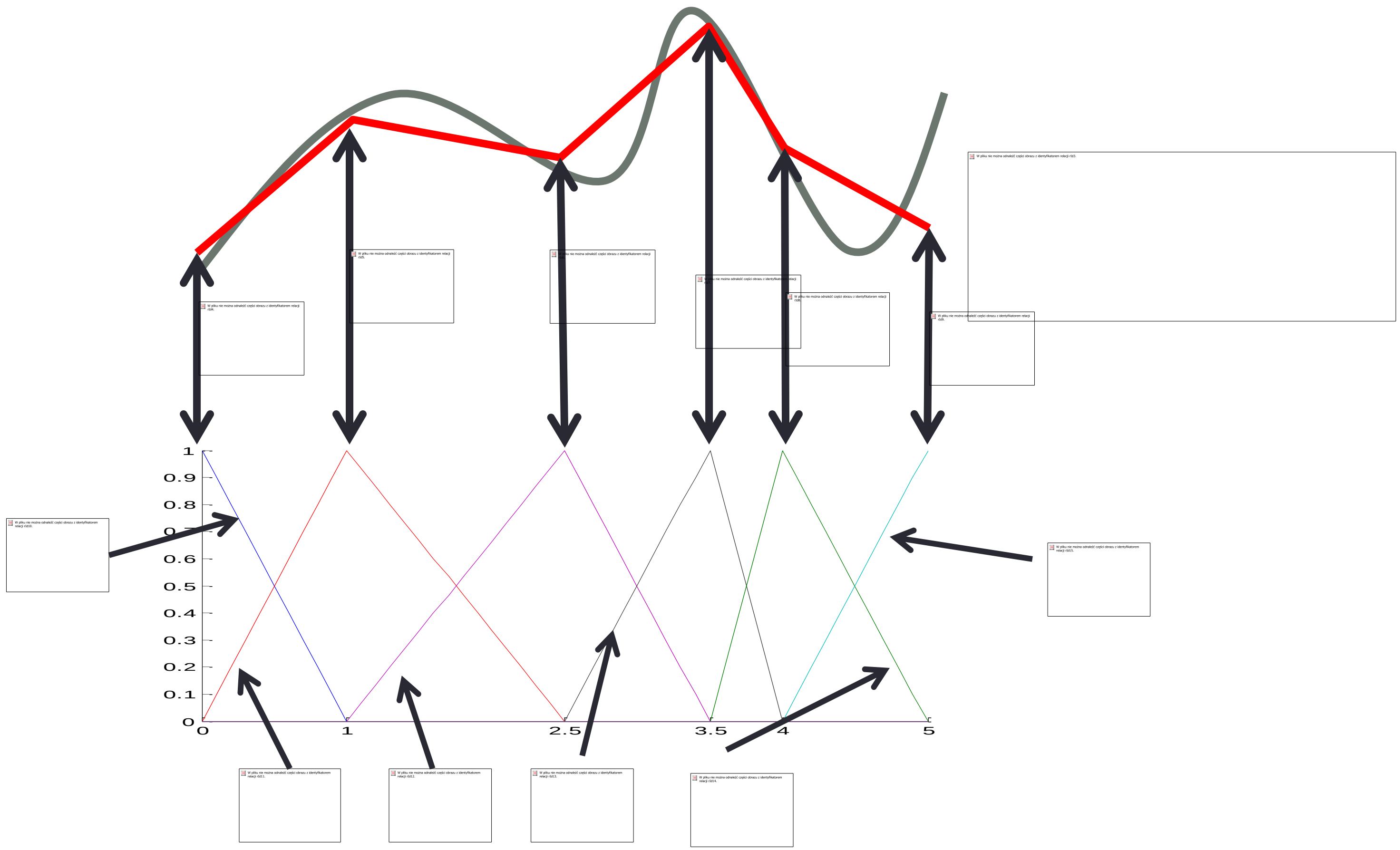


Figure 1. B-spline basis functions for degrees d = 1..4 (uniform)

Uniform B-spline approximation



Chordal B-spline approximation

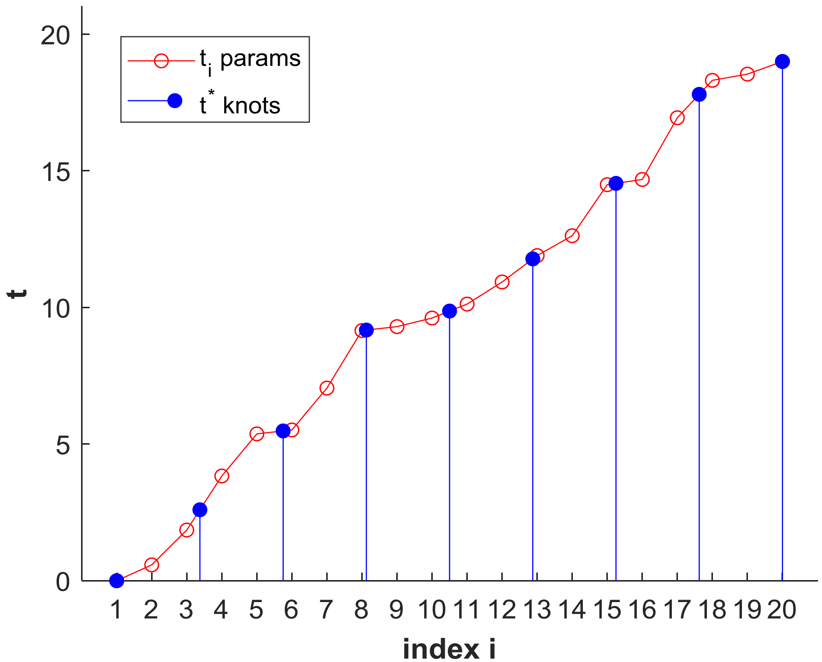


Knots placement

Different strategies are used for the placement of knots in the problem of approximation. The boundary knots are usually d-multiple:

$$t_0^*..t_d^* = t_0$$
$$t_b^*..t_{b+d}^* = t_{n-1}$$

Internal knots t_{d+i}^*, where $i = 1..i_{max}$ and $i_{max} = b - d$ can be distributed uniformly using linear interpolation of parameters for fractional index knot positions (C. de Boor '78)



Natural curves

Approximation of function points **C** with B-spline curves consists of finding best solution to the system of linear equations $\boldsymbol{C} = \boldsymbol{NP}$ and determination of control points $\boldsymbol{P}$.

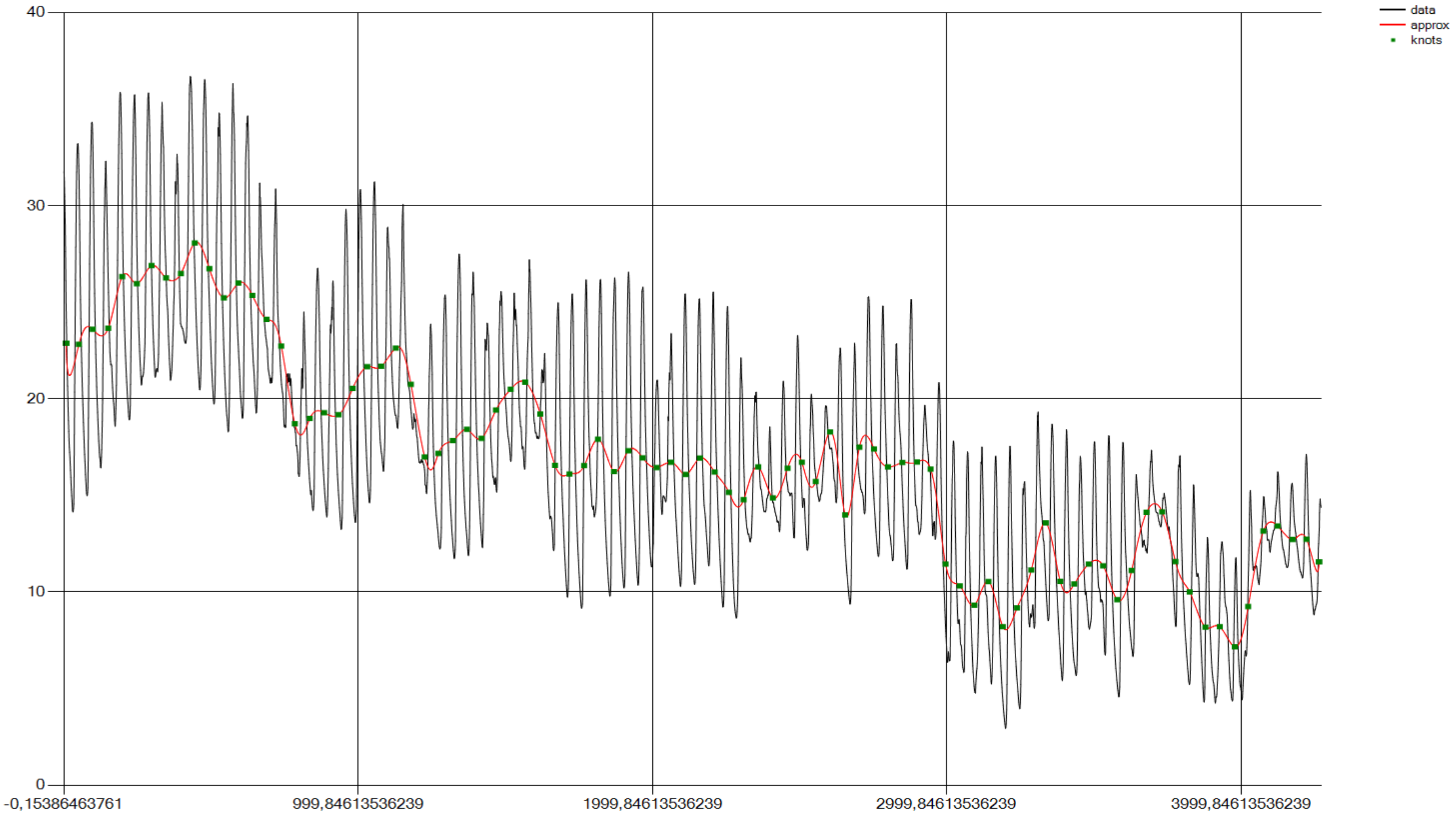
$$\boldsymbol{N}^T\boldsymbol{NP} = \boldsymbol{N}^T\boldsymbol{C}$$

$$\boldsymbol{P} = \left(\boldsymbol{N}^T\boldsymbol{N}\right)^{-1}\boldsymbol{N}^T\boldsymbol{C} = \boldsymbol{N}^{\dagger}\boldsymbol{C}$$

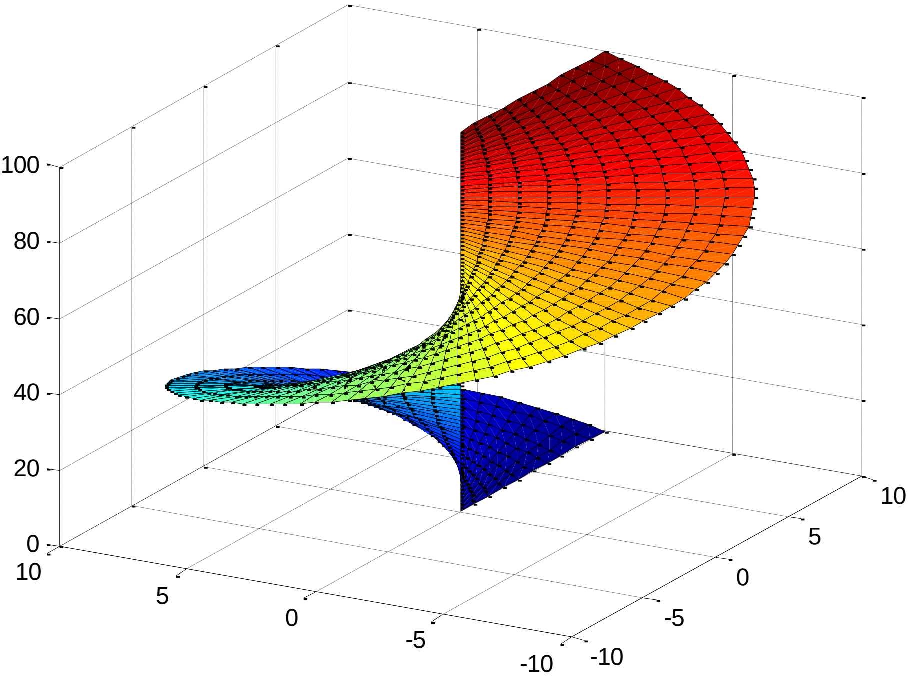
Natural boundary conditions were added to the system of approximation equations, which provided transition of ending curves into straight lines. For most common case, when d=3:

$$\begin{bmatrix} & & \boldsymbol{N} & & \\ -d & d & \dots & 0 & 0 \\ 0 & 0 & \dots & -d & d \end{bmatrix} \boldsymbol{P} = \begin{bmatrix} \boldsymbol{C} \\ C_1 - C_0 \\ C_{n-1} - C_{n-2} \end{bmatrix}$$

Temperature approximation of the dam's percolation sensors



Parametric surfaces



$$S(u,v) = x(u,v), y(u,v), z(u,v)$$

$$S(u,v) = \sum_{i=0}^{b-1} \left[\sum_{j=0}^{b-1} P_{i,j} B_j(v) \right] B_i(u)$$

$$S(u,v) = \sum_{i=0}^{b-1} C_i(v) B_i(u) \equiv \boldsymbol{S} = \boldsymbol{B_u C^T}$$

$$C_i(v) = \sum_{j=0}^{b-1} P_{i,j} B_j(v) \equiv \boldsymbol{C} = \boldsymbol{B_v P^T}$$

$$\boldsymbol{S} = \boldsymbol{B_u P B_v}^{\boldsymbol{T}}$$

- Tensor representation enables creating surface based on the curves in the u and v directions
- Curves in the u direction interpolate surface points, and curves in the v direction interpolate control points of these u-curves (or in the reverse order)

Tensor Surface Approximation

For data in rectangular grid form we can utilize parametric curve approximation theory for creating surfaces described as a tensor product of the B-Spline curves. For each row (column) of the grid, the knot vectors are the same, but the weights of basis function are different. Given a grid of $m \times n$ data points $\boldsymbol{S}$, approximation problem can be solved by finding a B-spline surface of degree d, defined by b control points and knot vector in direction u and v that approximates the data using two 1D approximation of curves: first approximation calculates weights $\widehat{\boldsymbol{C}}$ from surface points, second – weights $\widehat{\boldsymbol{P}}$ from weights $\widehat{\boldsymbol{C}}$:

$$[\boldsymbol{S}]_{m\times n} = [\boldsymbol{N_u}]_{m\times b}[\boldsymbol{C}]_{b\times n} \Rightarrow \widehat{\boldsymbol{C}} = \boldsymbol{N_u}^{\dagger}\boldsymbol{S}$$

$$[\boldsymbol{C}^{\boldsymbol{T}}]_{n\times b} = [\boldsymbol{N_v}]_{n\times b}[\boldsymbol{P}]_{b\times b} \Rightarrow \widehat{\boldsymbol{P}} = \boldsymbol{N_v}^{\dagger}\widehat{\boldsymbol{C}}^{\boldsymbol{T}}$$

$$\left[\widehat{\boldsymbol{S}}\right]_{m\times n} = [\boldsymbol{N_u}]_{m\times b}\left[\widehat{\boldsymbol{P}}^{\boldsymbol{T}}\right]_{b\times b}\left[\boldsymbol{N_v}^{\boldsymbol{T}}\right]_{b\times n} = \boldsymbol{N_u}\left(\boldsymbol{N_u}^{\dagger}\boldsymbol{S}\boldsymbol{N_v}^{\dagger\boldsymbol{T}}\right)\boldsymbol{N_v}^{\boldsymbol{T}}$$

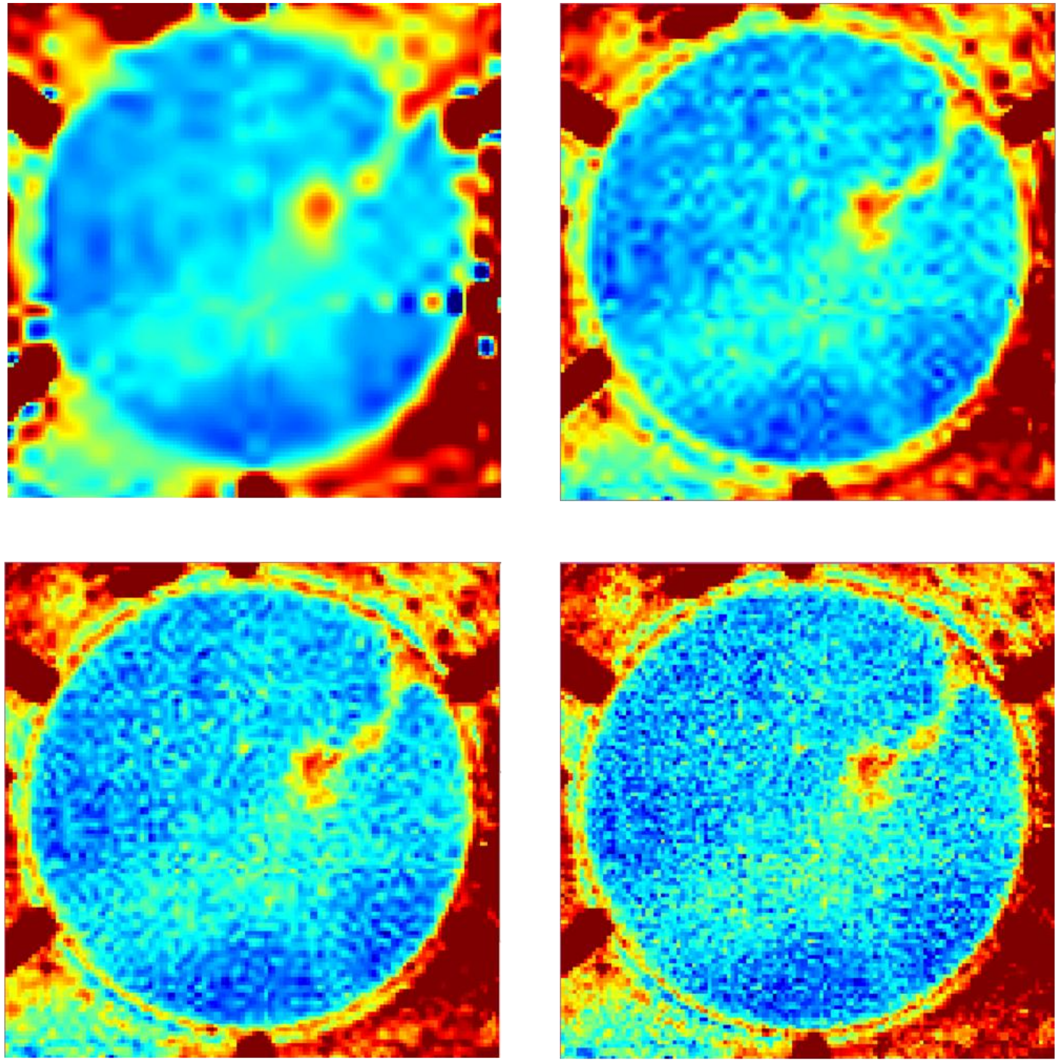


Figure 3. Example of image approximation with surface created by tensor-product of cubic B-splines for image resolution 128x128 and number of basis functions b in u and v directions equal to: a) 20, b) 50, c) 75 d) 100

Rows-Columns and Columns-Rows

The popular method of finding suboptimal solution to least square surface approximation consists of B-spline data approximation in *v* direction, followed by approximation in *u* direction or in reversed order. Changing of approximation order influences resulting surface. Formulas for the final surface are identical as for tensor product, but matrices N_v differ in both cases, because of parameterization. Also, computational cost of tensor surface is lower.

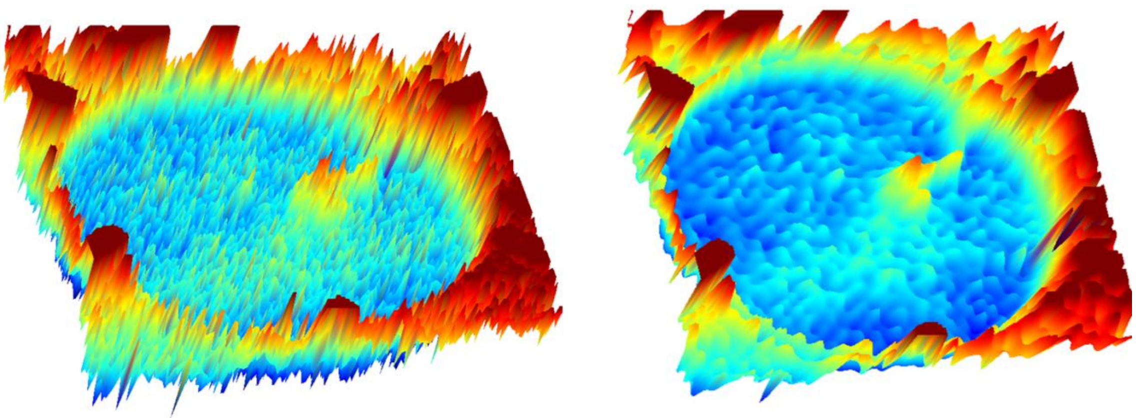


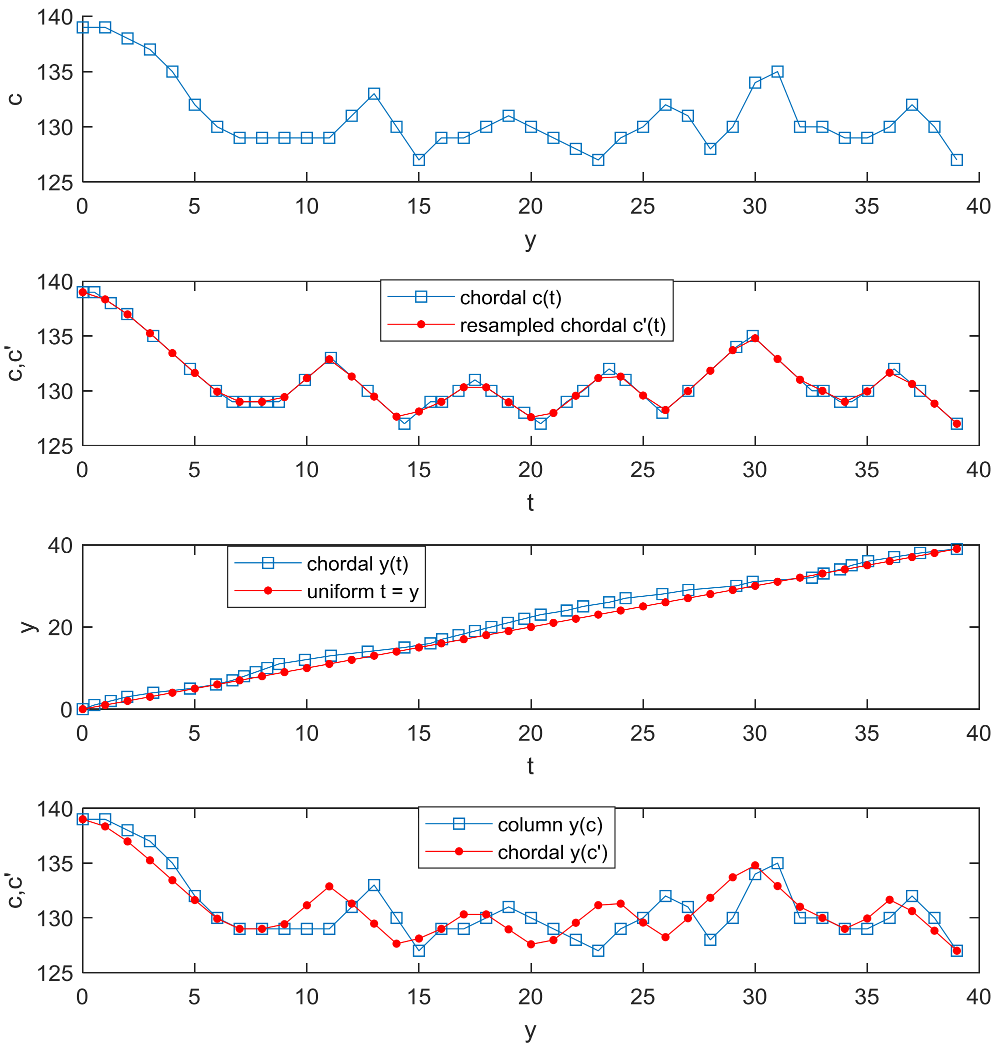
Figure 2. 3D view of processed image a) original, b) approximation for b = 50.

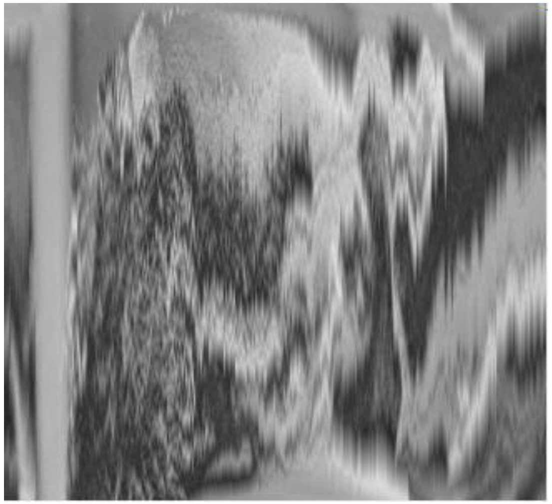
$$\widehat{S}_{cols} = N_u N_u^{\dagger} S$$

$$\widehat{S}^T = N_v N_v^{\dagger} \widehat{S}_{cols}^{\,T}$$

$$\widehat{S} = \left[N_u N_u^{\dagger}\right]_{m \times m} [S]_{m \times n} \left[N_v N_v^{\dagger}\right]_{nxn}^{T}$$

Lofting


c
y
chordal c(t)
resampled chordal c'(t)
c,c'
t
chordal y(t)
uniform t = y
y
t
column y(c)
chordal y(c')
c,c'
y



Approximation methods comparison

b = 15

b = 30

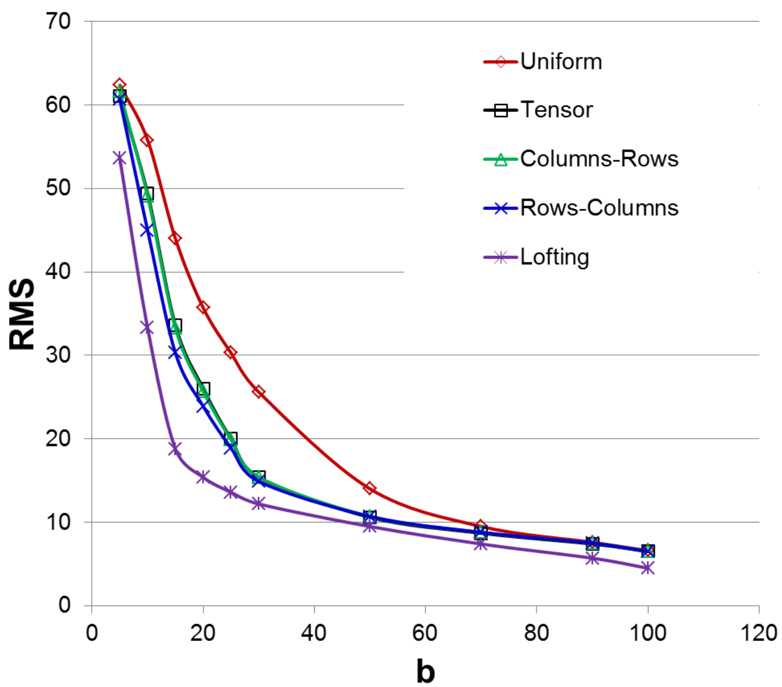
b = 50

Uniform

Chordal tensor

Lofting

Experiment results



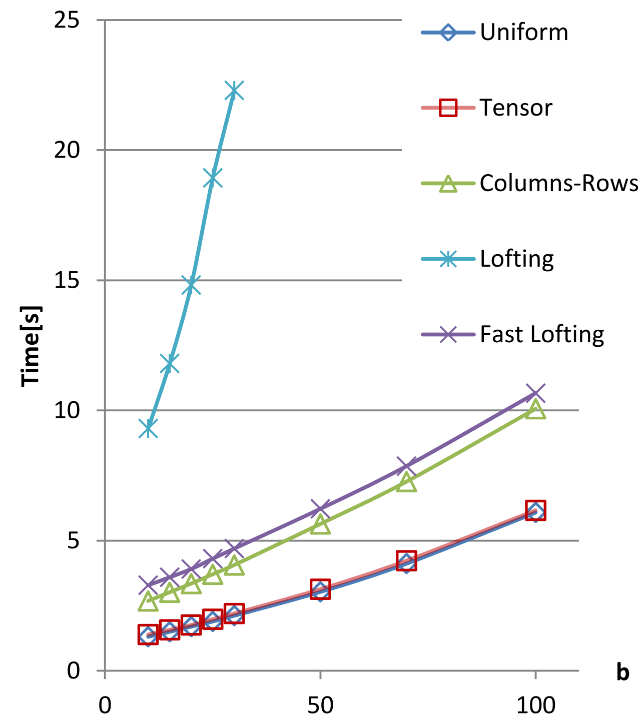


Figure 4. RMS errors of different approximation methods in function of basis number.

Conclusions

- Methods of image approximation using B-spline surfaces differ in knot placement when using non-uniform parameterization
- Creating surface from tensor product of B-splines proved to be the fastest method, approximation columns-rows is the easiest in implementation, but fast lofting with uniform resampling of chordal parameterized columns and rows gives the smallest errors with very little overhead on computations. Resampling of chordal parameterized image lines can also be used to accelerate approximation with tensor-build surfaces with slightly bigger errors, but faster execution times than lofting.
- Because of uniform parameters distribution, calculations can also be performed using FFT fast convolution of resampled image with B-Spline kernel function.
- B-spline approximation methods with their properties of excellent quality and speed have a great potential in data compression

THANK YOU

For Your Attention