Interpolation

Interpolation von Grauwerten

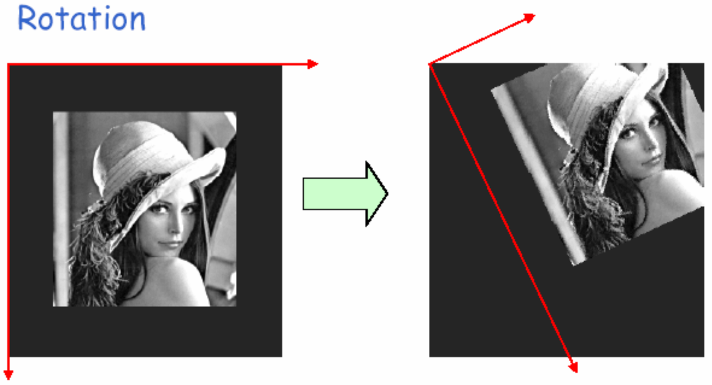
Backfrieder-Hagenberg

Warum Interpolation

- Pixel sind auf regelmäßigem Raster definiert
- Berechnung von Grauwerten zwischen den Pixeln
- Wo?
  - Vergrößern, Verkleinern
  - Rotieren
  - Dehnen und Stauchen

Backfrieder-Hagenberg

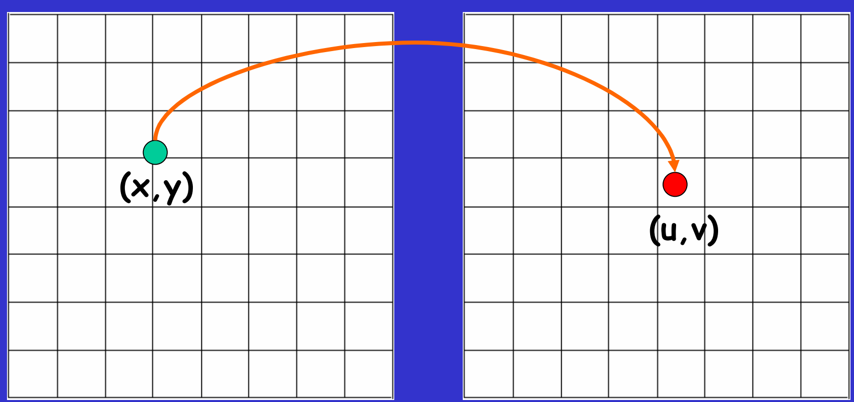
Rotation



$$x = \quad u \cdot \cos\alpha + v \cdot \sin\alpha$$
$$y = -u \cdot \sin\alpha + v \cdot \cos\alpha$$

$$[x, y, 1] = [u, v, 1] \cdot \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformation



Transformierter Punkt (u,v) im Allgemeinen nicht auf einer Rasterposition -> Interpolation

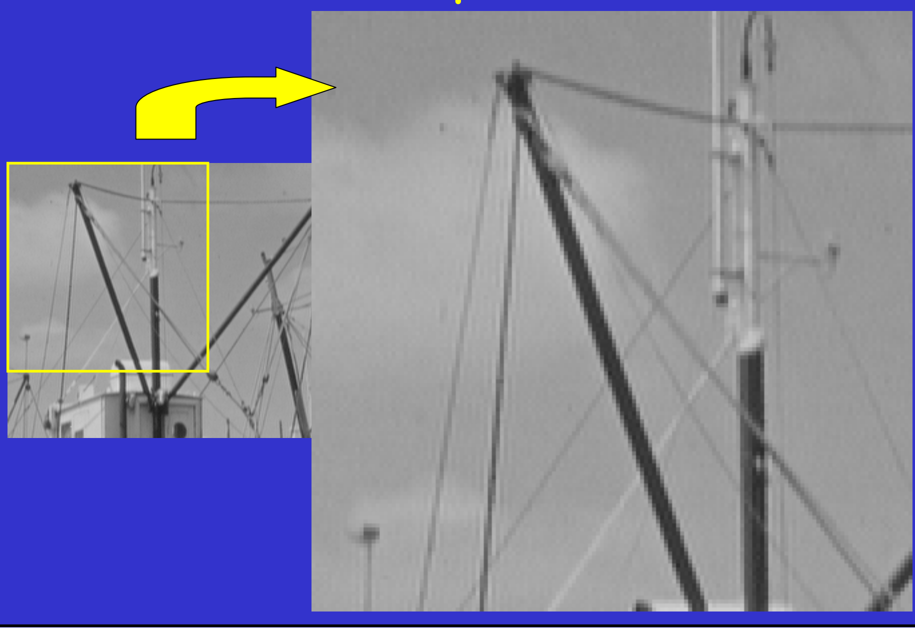
Methoden

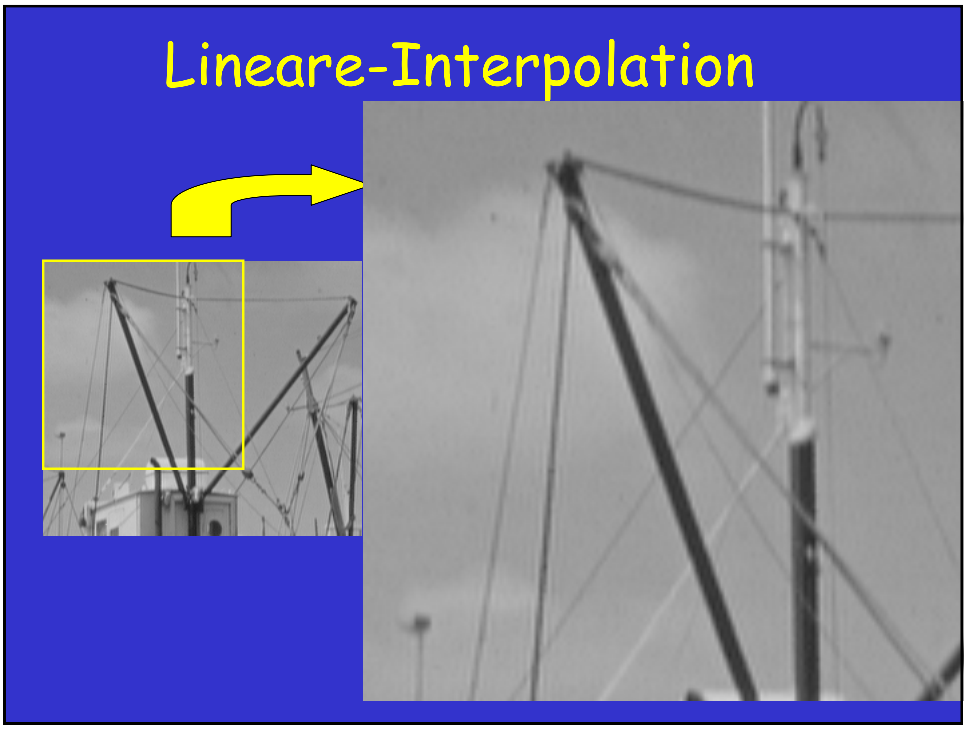
- Nächste Nachbar Interpolation
- Lineare Interpolation
- Kubische Interpolation
  - Faltung, Splines
- Sinc-Interpolation

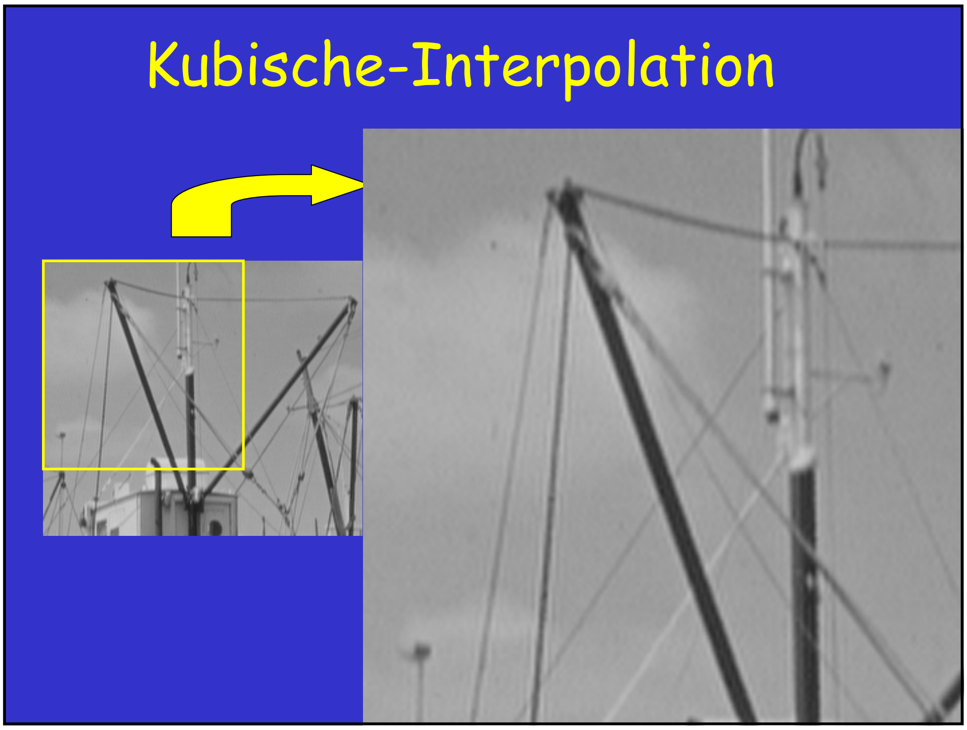
- Eigenschaften: Qualität, Performance

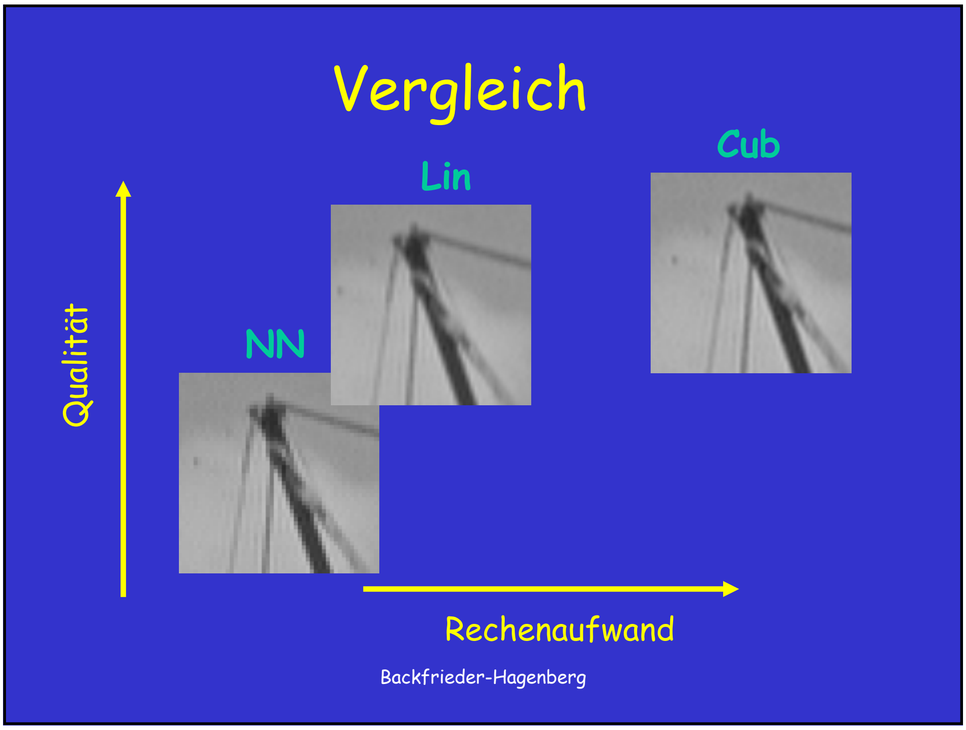
Backfrieder-Hagenberg

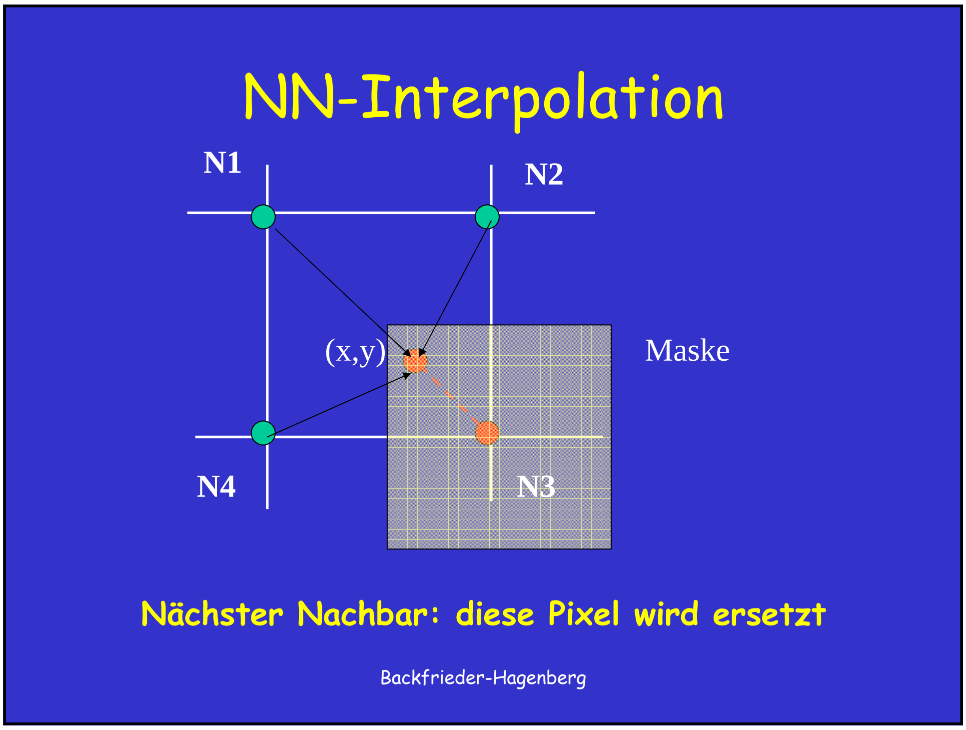
NN-Interpolation




Lineare-Interpolation


Kubische-Interpolation


Vergleich
Cub
Lin
NN
Qualität
Rechenaufwand
Backfrieder-Hagenberg

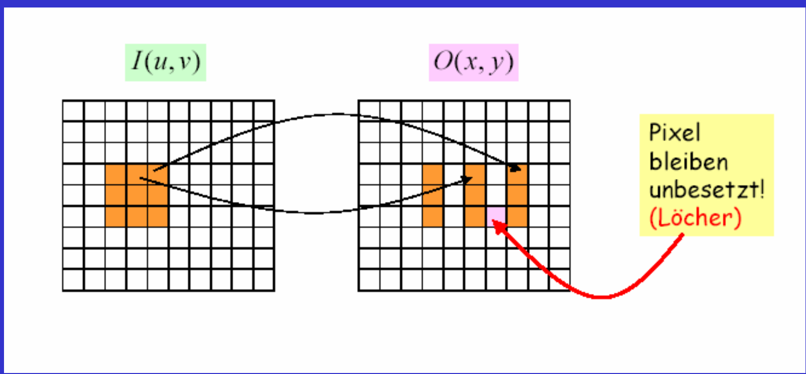

NN-Interpolation
N1
N2
(x,y)
Maske
N4
N3
Nächster Nachbar: diese Pixel wird ersetzt
Backfrieder-Hagenberg

NN: Eigenschaften

- Vorteil: einfache Berechnung (schnell)
- Nachteil: schlechte Qualität
  - Blockbildung
  - Aliasing

Backfrieder-Hagenberg

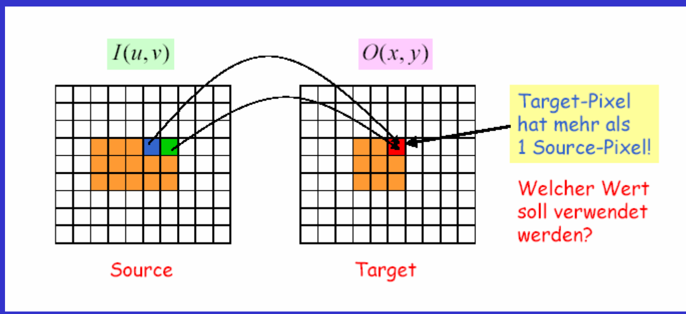
NN: Probleme



Horizonale Skalierung x 2:
„Löcher"

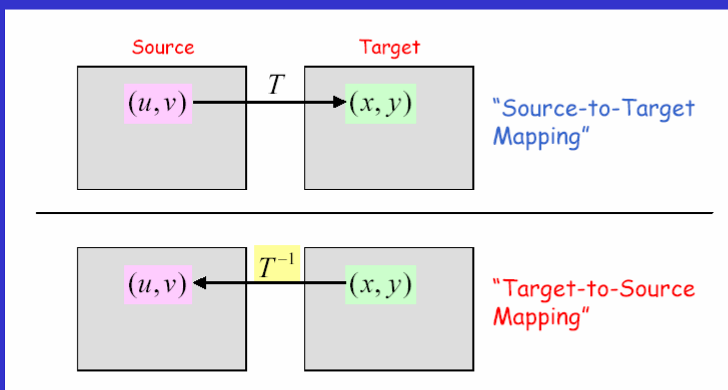
Backfrieder-Hagenberg

NN: Probleme

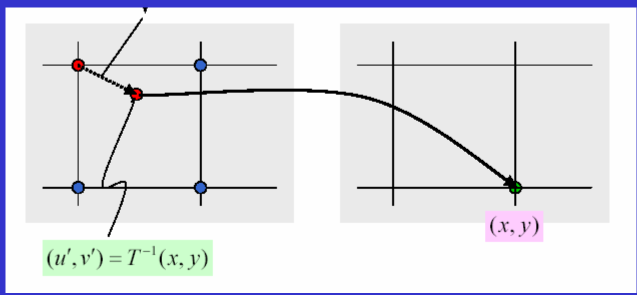


Mehrfachbesetzung bei Verkleinerung

Mapping

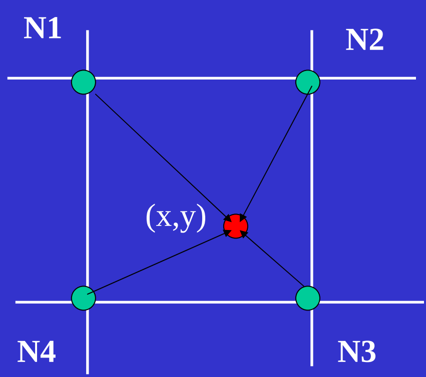


Lösung: backward mapping



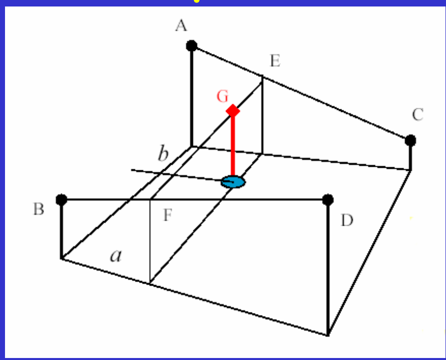
Pixelposition wird im Ausgangsbild bestimmt und dort berechnet

Lineare-Interpolation



Lineare Interpolation: alle Nachbarn tragen zum Pixel bei

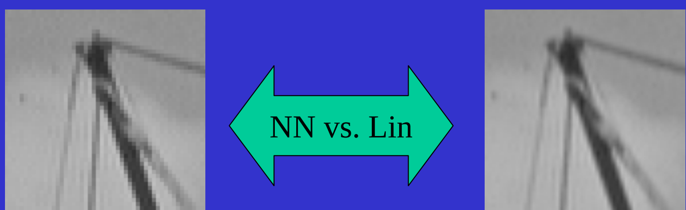
Prinzip der Bi-Linearen-Interpolation



Pixelwert G wird in Abhängigkeit der vier Nachbarpixel berechnet

Bi-Lineare Interpolation

- Schnell zu berechnen
- „gute“ Bildqualität
- sehr häufig verwendetes Verfahren

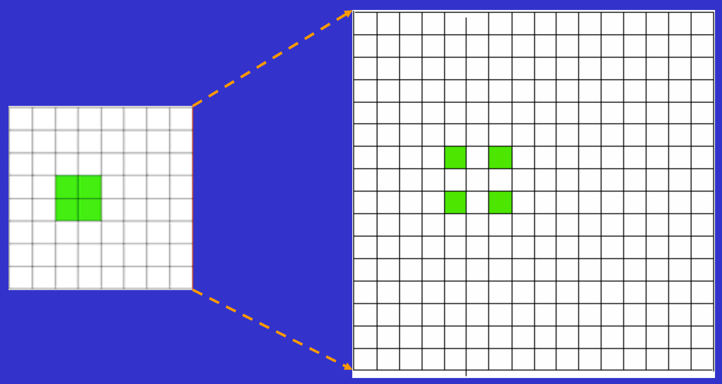


Interpolation als Maskenoperation

- Erweitern der Maske
- Zwischenstellen Null setzen
- Maske berechnen
  - linear: Dreiecks-Funktion
- Maske über das Bild schieben.

Backfrieder-Hagenberg

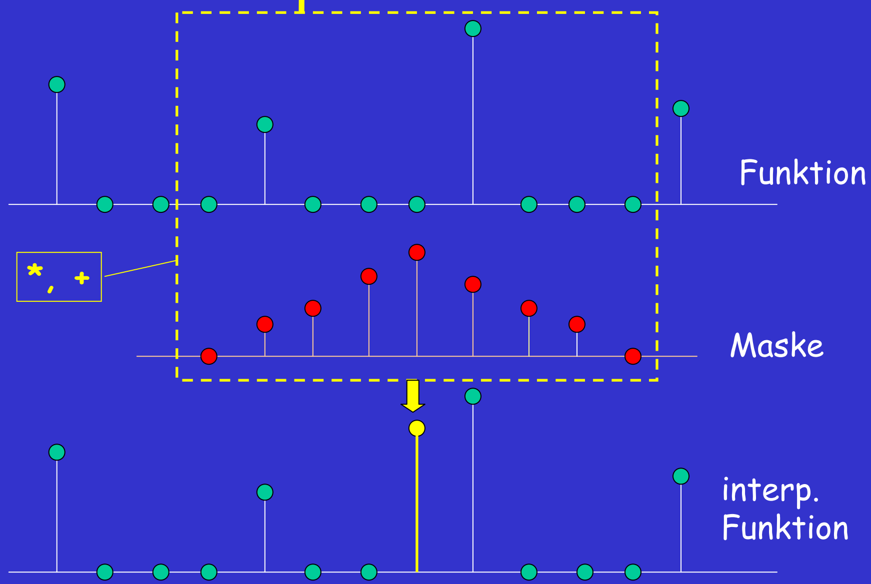
Vergrößern der Matrix



Füllen der Zwischenräume mit Nullen, Berechnung der neuen Positionen mittels Maskenoperation

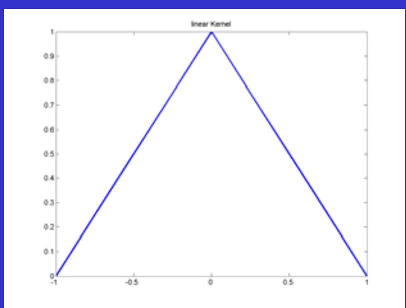
Backfrieder-Hagenberg

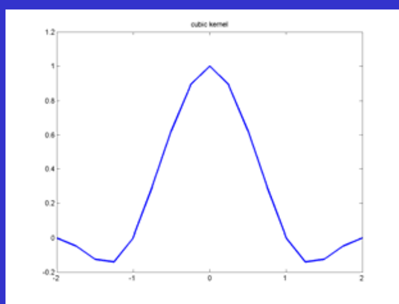
Interpolation-Masken



Backfrieder-Hagenberg

Interpolations-Kerne





Lineare Interpolation:
Dreiecksfunktion

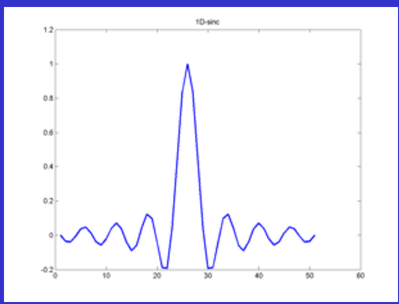
Kubische Interpolation:

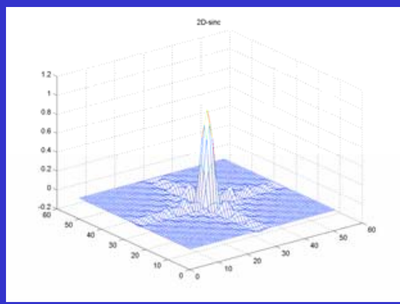
$$k(x) = \begin{cases} 0 \,\forall\, |x| > 2 \\ a|x|^3 - 5ax^2 + 8a|x| - 4a \,\forall\, 1 < |x| \le 2 \\ (a+2)|x|^3 - (a+3)x^2 + 1 \,\forall\, |x| \le 1 \end{cases}$$

Backfrieder-Hagenberg

Sinc-Interpolation

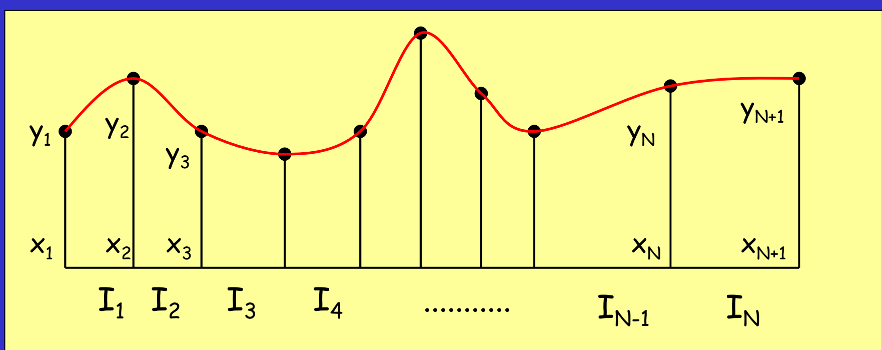
- Funktion: sinc(x)=sin(Pi*x)/(Pi*x)
- Ideale Interpolation
- Glatt, kein Aliasing, langsam
- Reduktion der Maskengröße: Cubic ist abgeschnittener sinc





Splineinterpolation-Problemstellung

- Polynome definiert auf Teilintervalle I_n
- Polynome stetig
- N Teilintervalle, N+1 unregelmäßige Stützstellen



Spline-Interpolation

- Polynome 3.Grades
- stückweise zwischen den Stützstellen
- stetig in den Stützstellen
- erste und zweite Ableitung stetig in den Stützstellen
- Formel: $p_n(x) = a_n x^3 + b_n x^2 + c_n x + d_n$

Backfrieder-Hagenberg

Bedingungen

- N+1 = Anzahl der Stützstellen
- N Polynome
- 4*N unbekannte Parameter

$$p_n(x_n) = f(x_n)$$
$$p_n(x_{n+1}) = f(x_{n+1})$$
$$p_n'(x_{n+1}) = p_{n+1}'(x_{n+1})$$
$$p_n''(x_{n+1}) = p_{n+1}''(x_{n+1})$$

4 Bedingungen für jedes Intervall, außer Randintervalle!!!

Backfrieder-Hagenberg

Randbedingungen

- 4*N Unbekannte
- reguläre Intervalle -> 4*(N-2) Gleichungen
- Randintervalle 2*3 Gleichungen
- 2 Gleichungen fehlen

- Natürliche Randbedingungen
  $p''_1(x_1)=0$
  $p''_N(x_{N+1})=0$

- Zyklische Randbedingungen
  $p'_1(x_1)= p'_N(x_{N+1})$

Eine der beiden Bedingungen muß erfüllt sein!