

Computergraphik

Computergraphik

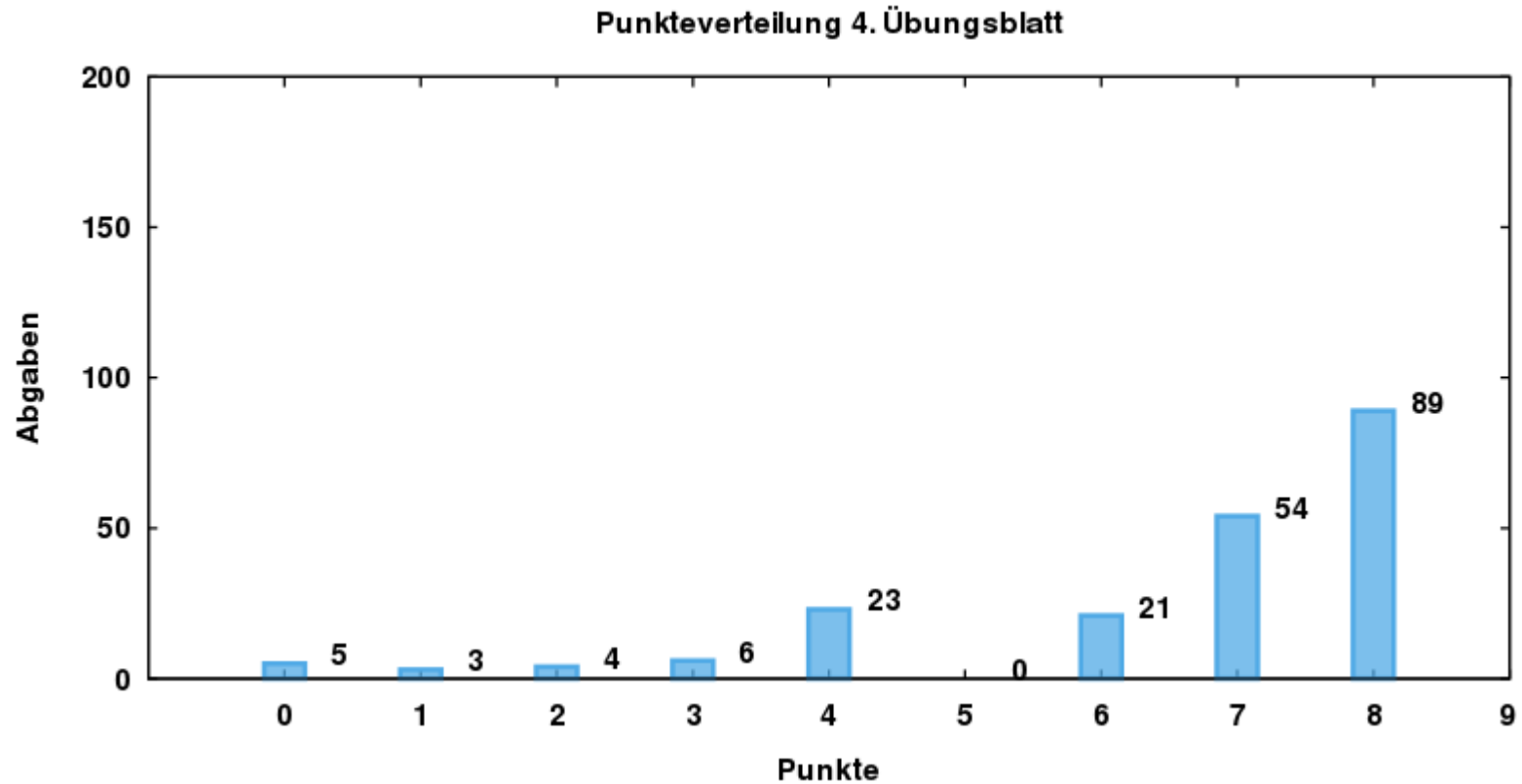
Übung zu Texture Mapping

Prof. Dr.-Ing. Carsten Dachsbacher
Lehrstuhl für Computergrafik
Karlsruher Institut für Technologie



- ▶ Vielen Dank für alle Rückmeldungen zum Übungsblatt
 - ▶ Dadurch konnten wir einen Fehler im Framework beheben

- ▶ Zu den Tests
 - ▶ Wir haben andere Tests als die, die wir euch mitgeben
 - ▶ Wir können unmöglich alle eventuellen Fehler abtesten
 - ▶ Löst die Aufgaben, nicht die Tests



▶ 3D - Homogene Koordinaten

▶ Punkte: $(x, y, z) \rightarrow (x, y, z, 1)$

▶ Richtungen: $(x, y, z) \rightarrow (x, y, z, 0)$

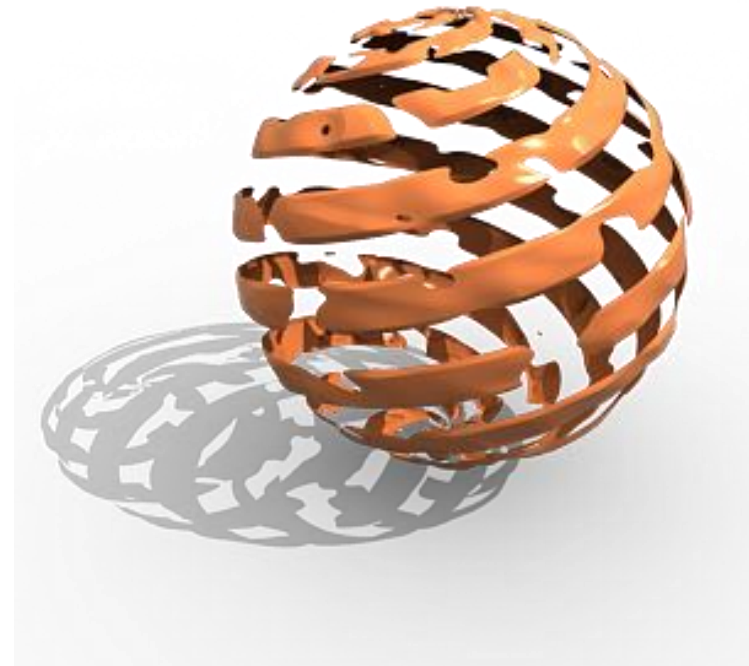
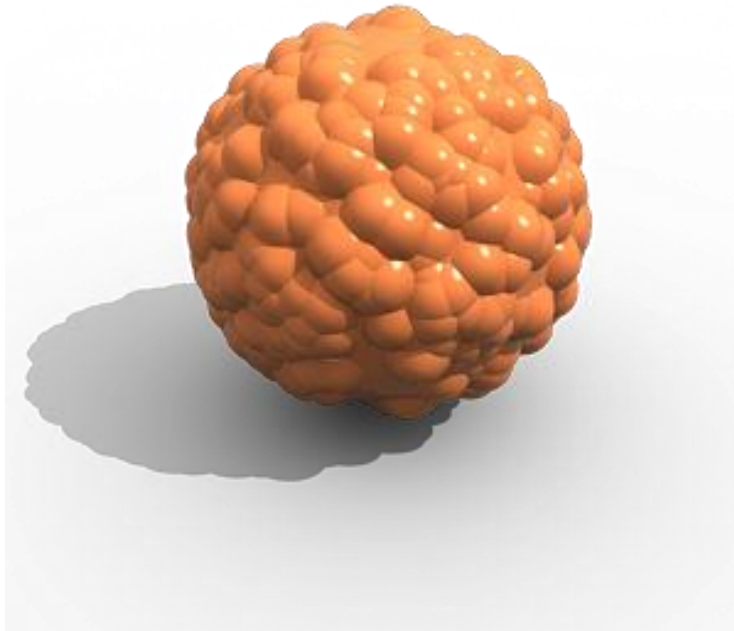
▶ $(w * x, w * y, w * z, w) = (x, y, z, 1)$ für alle $w \neq 0$

▶ Beispiel: 2D – Homogene Koordinaten

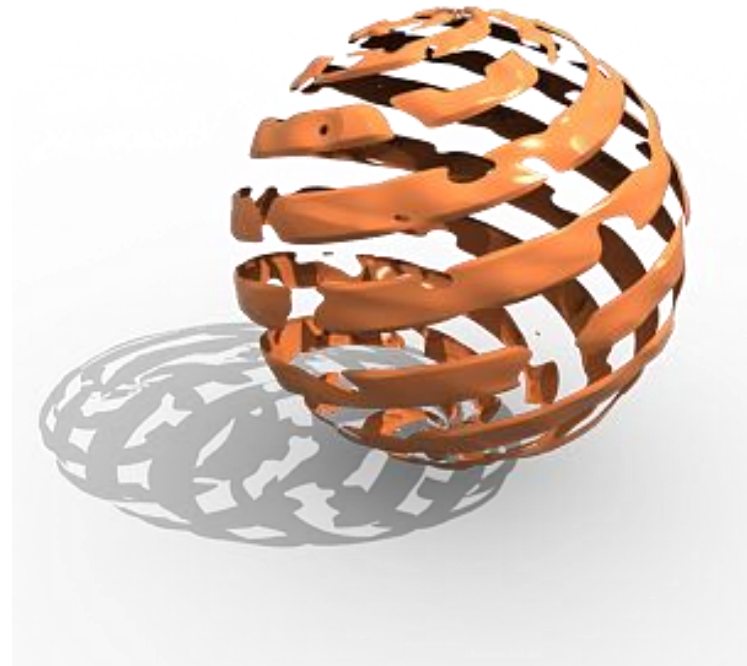
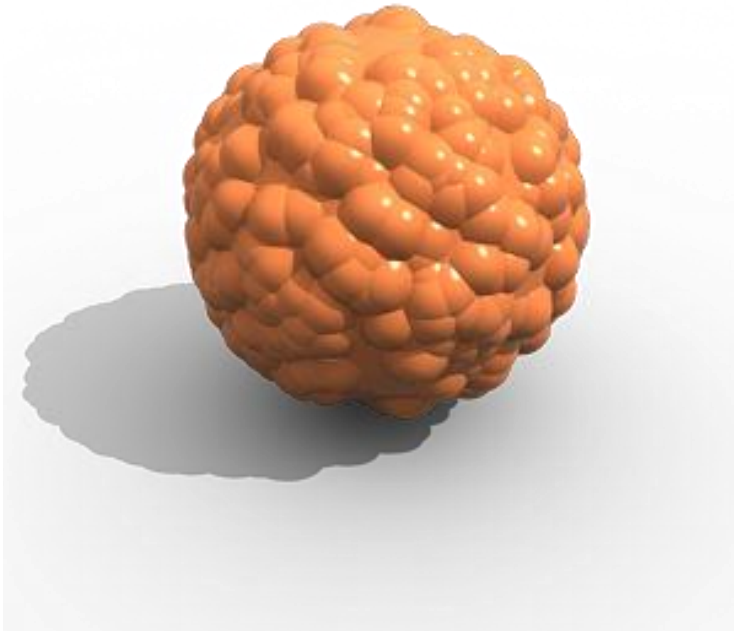
▶
$$\begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}$$

▶
$$\begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

- ▶ Ermöglicht, Objekten viel Oberflächendetail hinzuzufügen
 - ▶ Variation der Materialeigenschaften (Farbe, Spekularität)
 - ▶ Variation der Geometrischen Eigenschaften (Normale, Ableitungen, ...)
- ▶ Vergleichsweise günstig und flexibel
- ▶ Einfach zu filtern

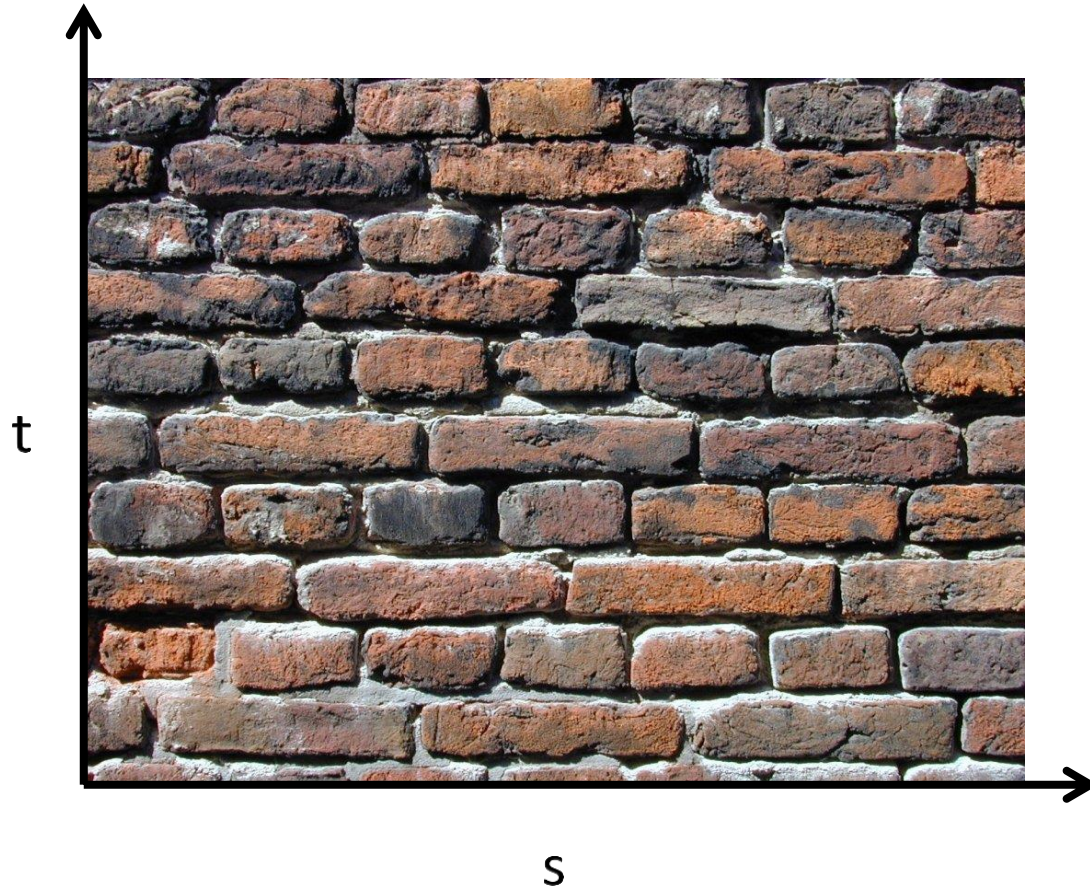


- ▶ Ermöglicht, Objekten viel Oberflächendetail hinzuzufügen
 - ▶ Variation der Materialeigenschaften (Farbe, Spekularität)
 - ▶ Variation der Geometrischen Eigenschaften (Normale, Ableitungen, ...)
- ▶ Vergleichsweise günstig und flexibel
- ▶ Einfach zu filtern



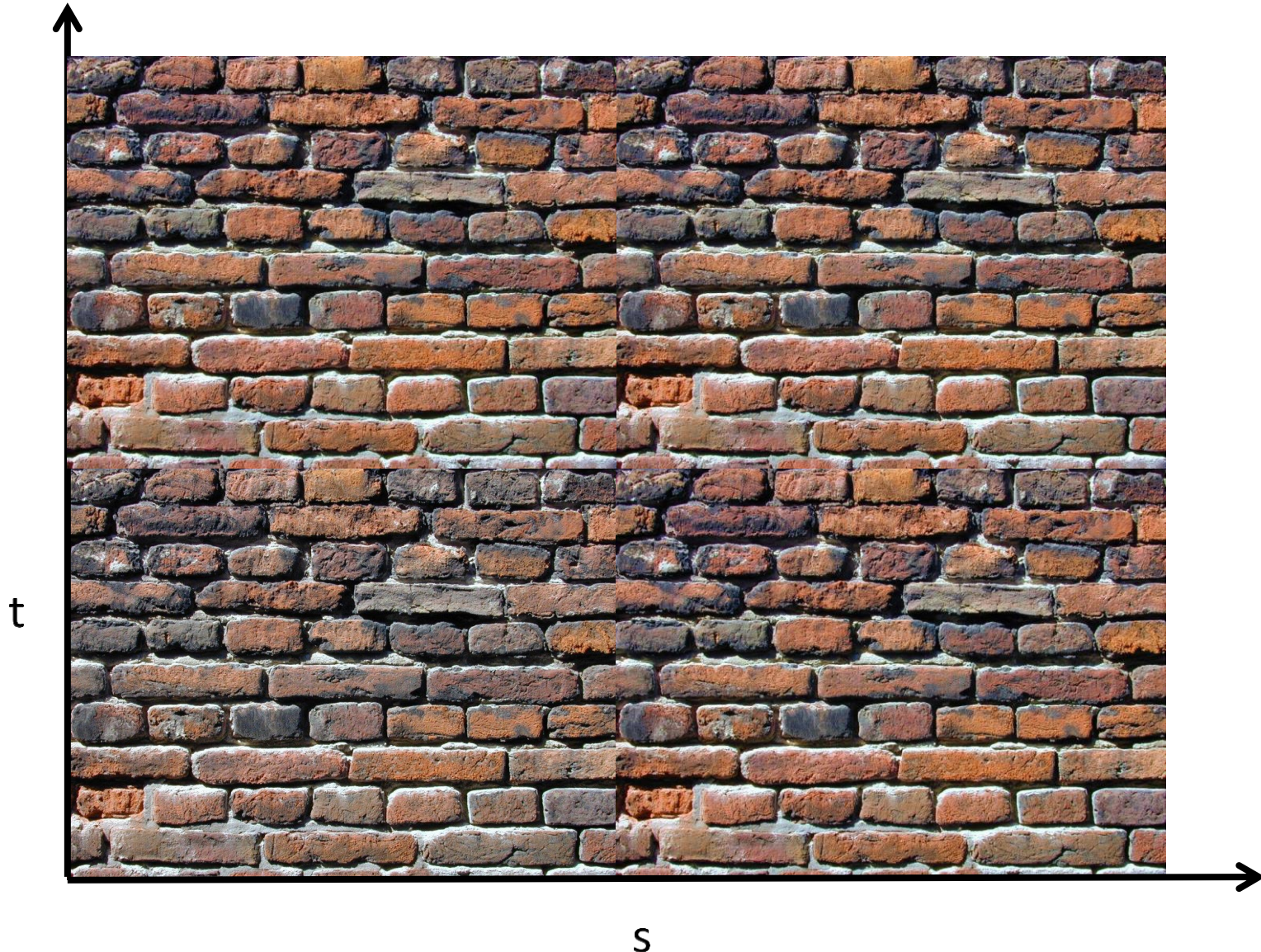
Repeating / Clamping

- ▶ Wie auf Textur zugreifen, wenn Texturkoordinaten nicht in $[0, 1]$?



Repeating / Clamping

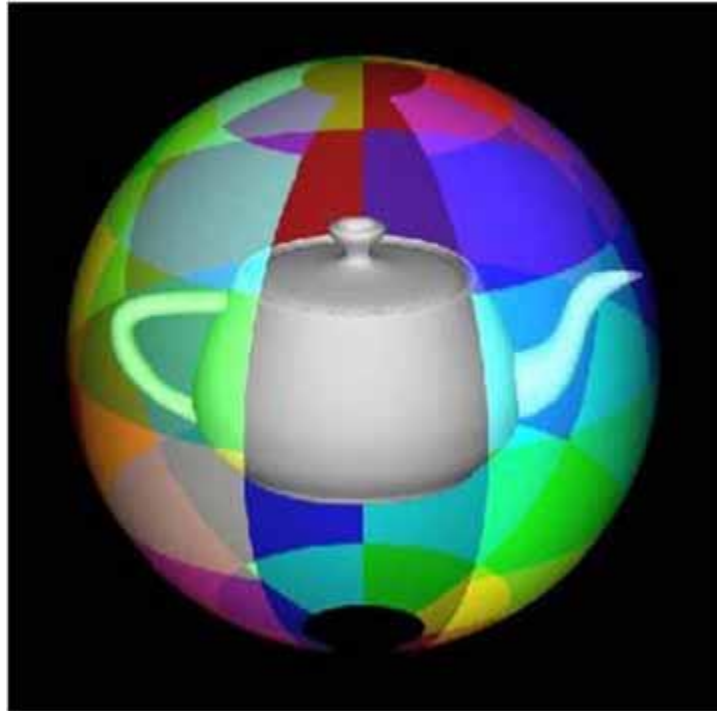
- ▶ Wie auf Textur zugreifen, wenn Texturkoordinaten nicht in $[0, 1]$?



Sphärische Parametrisierung

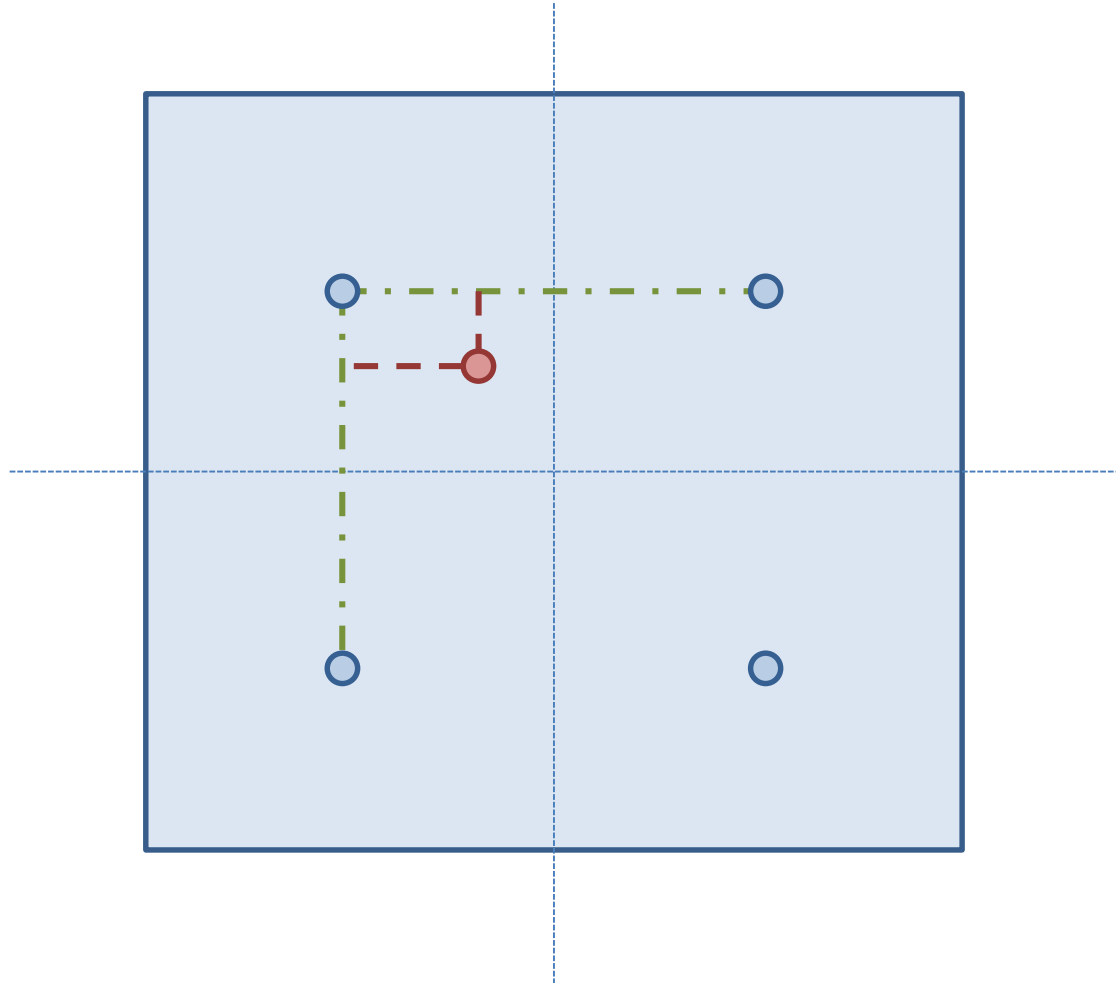
- ▶ Basiert auf Polarkoordinaten φ, θ . Der Radius wird dabei ignoriert.

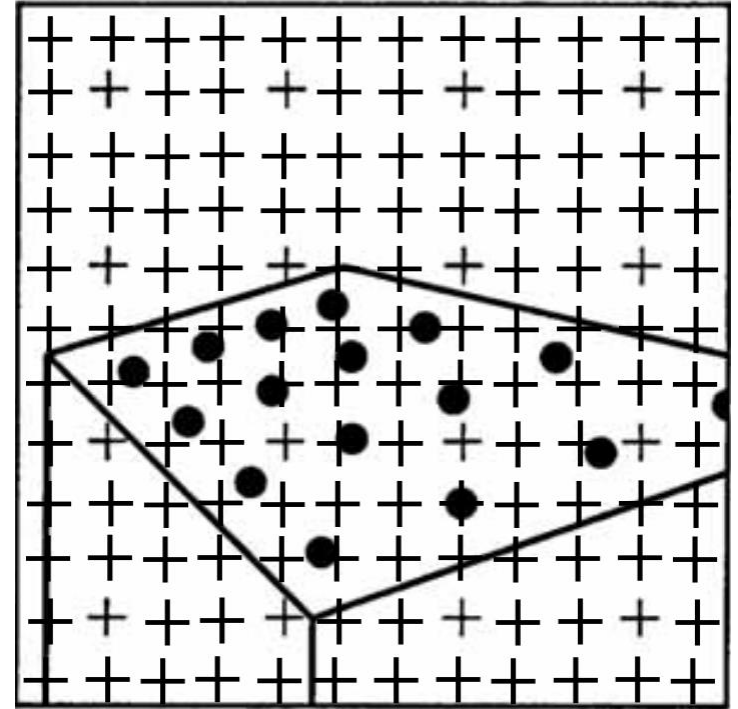
$$\begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} \varphi/(2\pi) \\ \theta/\pi \end{pmatrix}$$



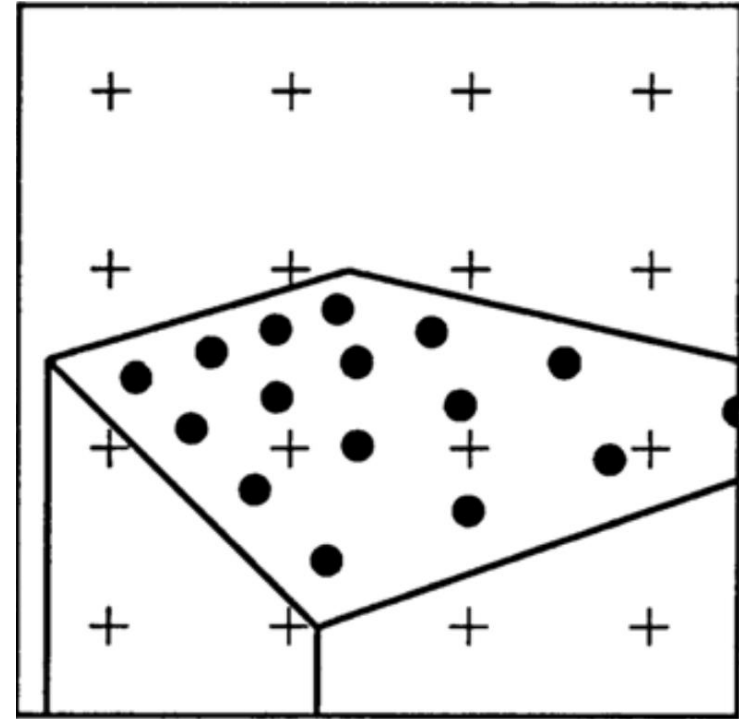
Bilineare Filterung

► Funktioniert wie bilinear Interpolation!





- ▶ Unterabtastung führt zum Verlust von Information
- ▶ Versucht man durch Filterung wieder zu Rekonstruieren

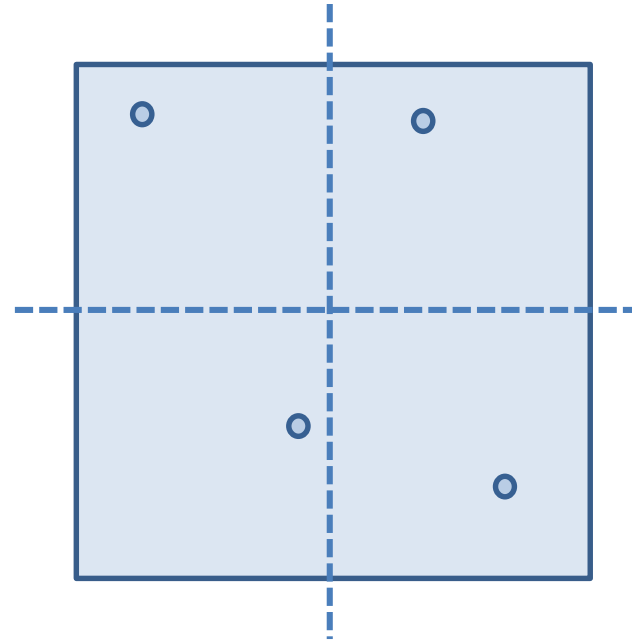
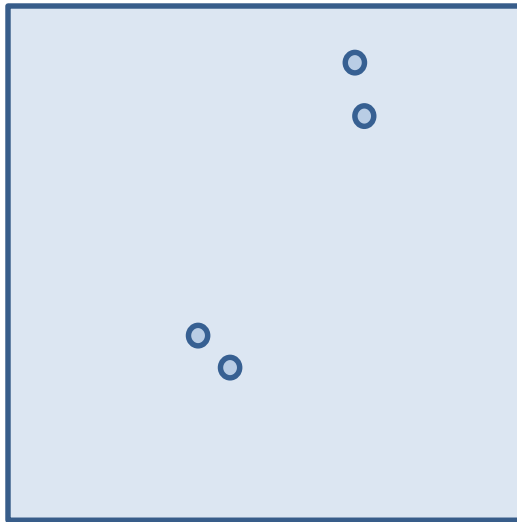


- ▶ Kann verringert werden durch
 - ▶ Überabtastung (Supersampling)
 - ▶ Vorfilterung (falls möglich, MipMapping)

Adaptive Stratified Supersampling

► „Stratified“ ?

- Strahlen durch zufällige Subpixelpositionen, aber
- Möglichst gleichmäßig verteilt (keine „Klumpen“)



► „Adaptive“ ?

- Mehr Primärstrahlen verschießen, wenn viel Aliasing zu erwarten ist.
- Zum Beispiel:
 - N Strahlen verschießen, Pixelfarbe I_0 als Mittelwert
 - N+1 Strahlen verschießen, Pixelfarbe I_1 als neuer Mittelwert
 - N+2 Strahlen, falls I_0 und I_1 zu unterschiedlich sind
 - usw...

