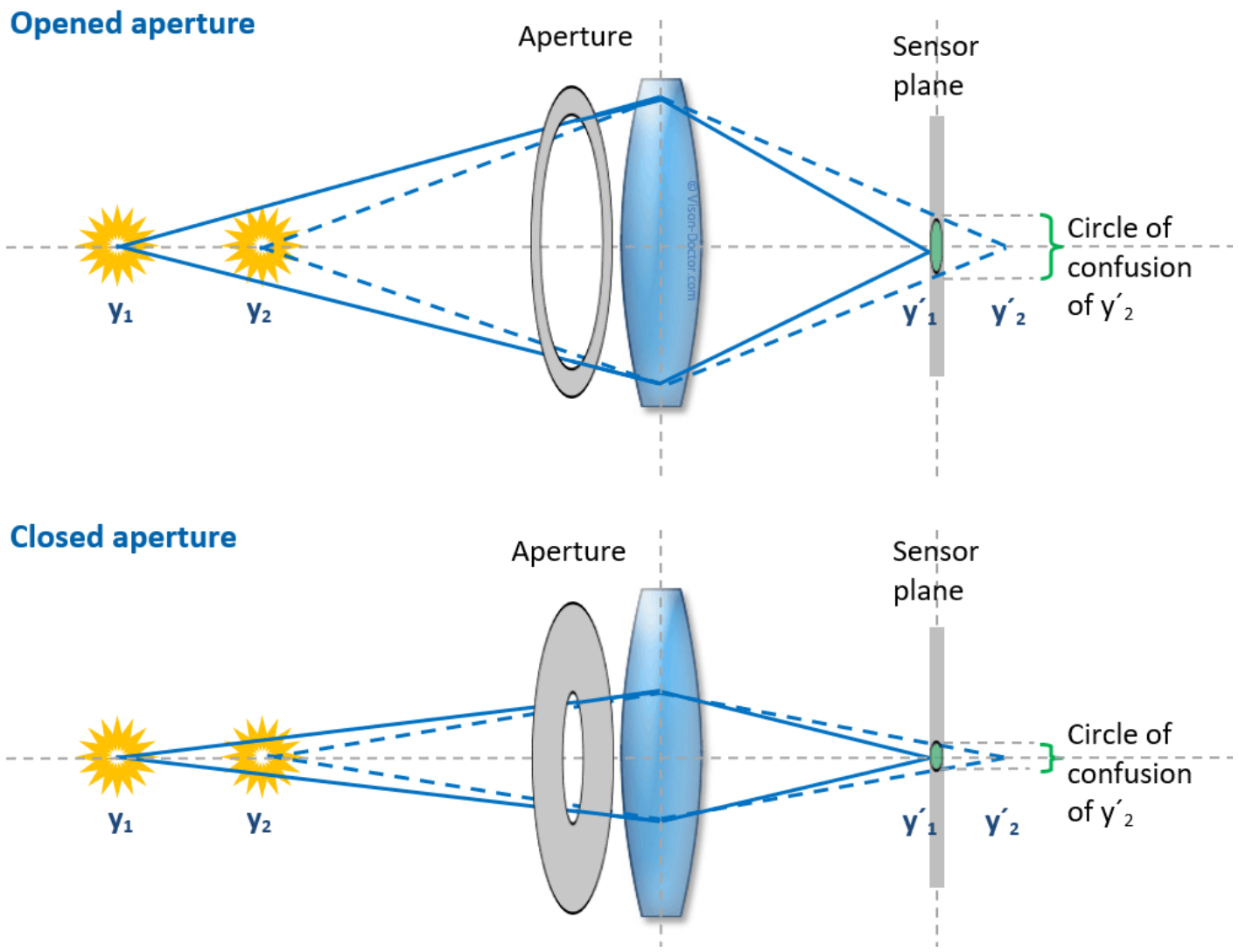
UNIVERSITÄT
LEIPZIG

Texturen

COMPUTERGRAPHIK

TIEFENUNSCHÄRFE



INHALTSVERZEICHNIS

10 Texturen

- 10.1 Texture-Mapping
- 10.2 Bump-Mapping
- 10.3 Environment-Mapping
- 10.4 Weitere Verfahren

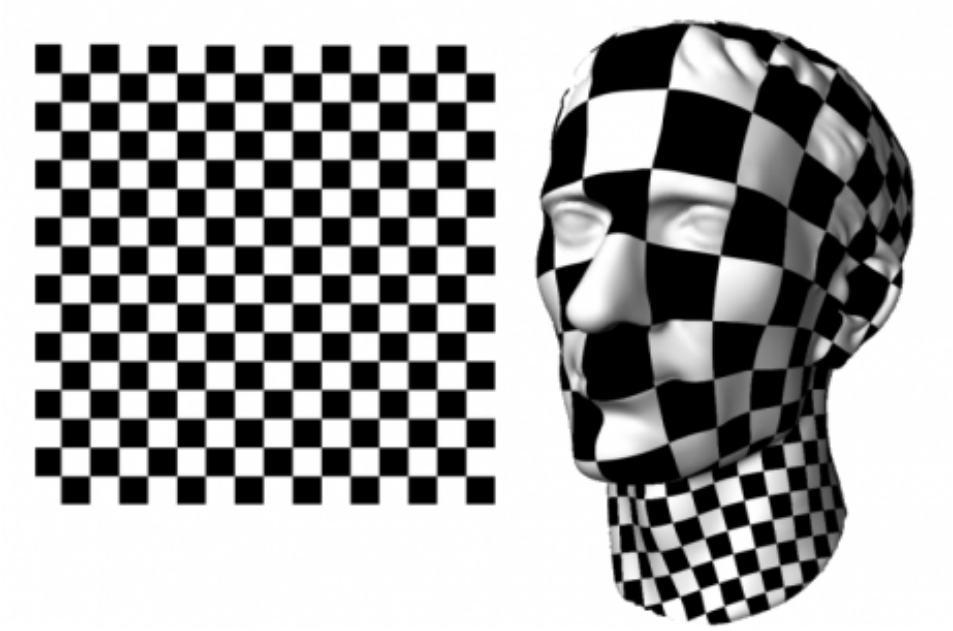
10. ABBILDUNGSVERFAHREN

Motivation

- Reale Umgebung verfügt über ein großes Spektrum geometrischer Formen und physikalischer Materialien
  - Maserungen und Muster von Oberflächen
  - Strukturen unebener Flächen
  - Hintergrund und Spiegelungen mit hohem Detailgrad
- Exakte Nachbildung dieser Objekte ist meist zu aufwändig.

10.1 TEXTURE-MAPPING

- Texture-Mapping
  - Aufbringen von 2D-Texturen auf eine 3D-Oberfläche
  - Ermöglicht komplexe Gestaltung einfacher Objekte



- Beispiele
  - Blick aus dem Fenster
  - Spiegelbild
  - Parkettboden

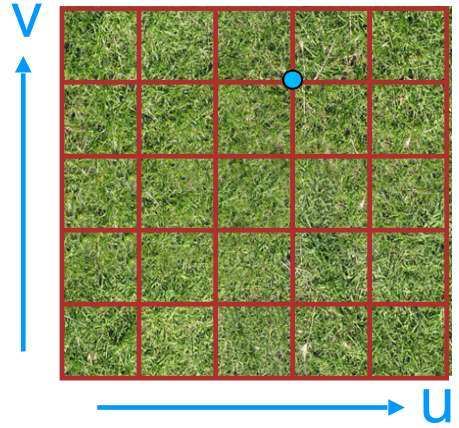
10.1 TEXTURE-MAPPING

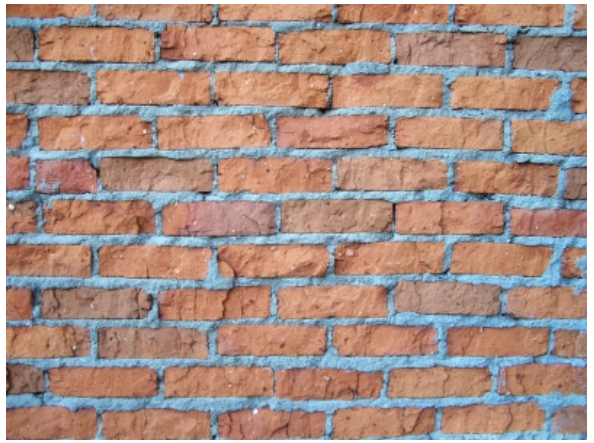
Texturen

- Funktionen, die Punkte des (u,v)-Texturraums auf (r,g,b)-Werte abbilden:

$$(r, g, b) = C_{textur}(u, v)$$

- $(u, v) \in [0; 1]$





10.1 TEXTURE-MAPPING

Diskrete Texturen

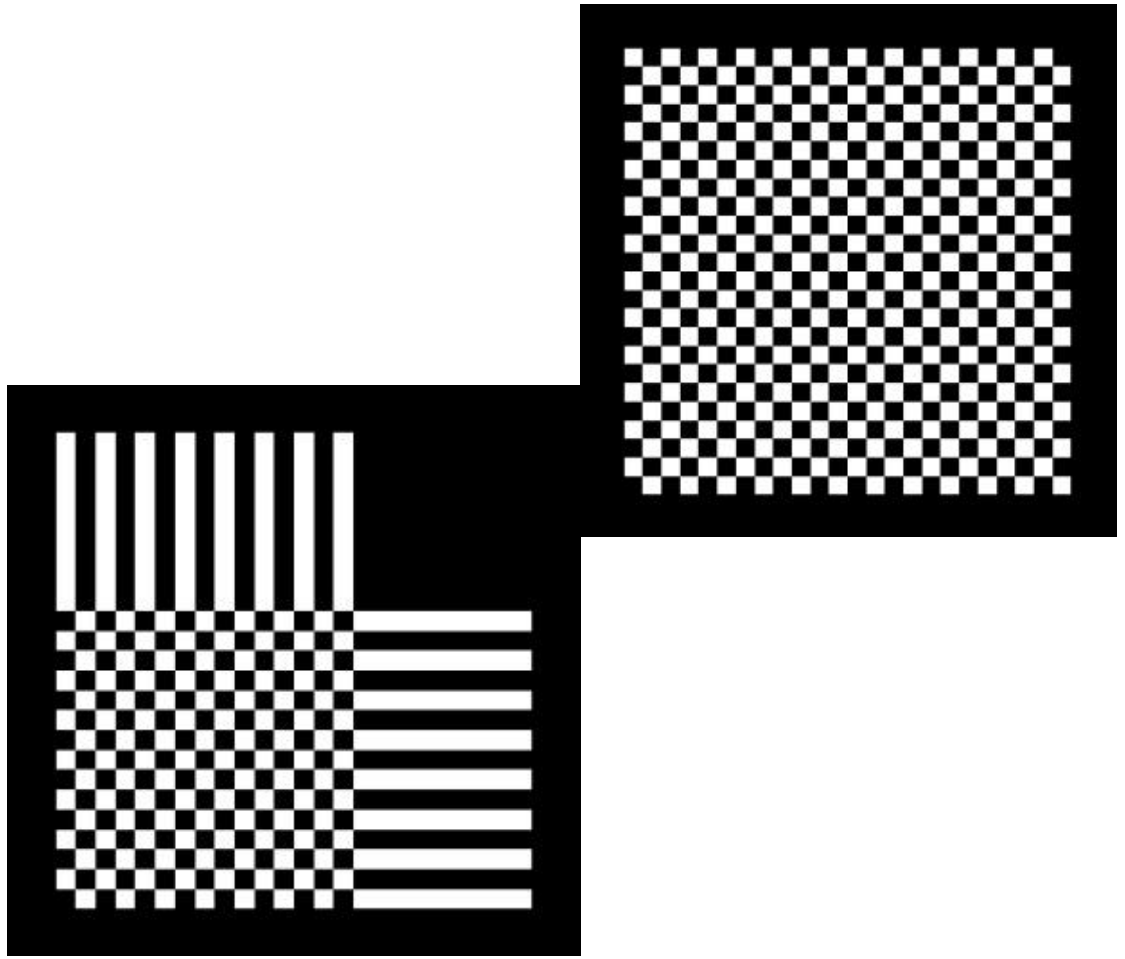
- Diskret
  - Werden als Vektorfelder C abgespeichert
  - Ein Vektor enthält Farbkomponenten und wird als Texel bezeichnet.
- Vorteile
  - Vorrat unerschöpflich
    - Smartphone
    - Fotoapparat
    - Scanner
    - Downloads
  - Photorealismus möglich
- Nachteile
  - Hoher Speicherbedarf
  - Unstimmiger Kontext von Szene und Textur (Schattenwurf…)
  - Anfällig für Artefakte und Aliasing
  - Rekonstruktion der Texturwerte notwendig

10.1 TEXTURE-MAPPING

Diskrete Texturen

- Sonderregeln, falls $(u, v) \notin [0; 1]^2$
- Periodisch (Repeat)
  $C(u) = \left(u - \lfloor u \rfloor\right)$
- Abschneiden (Clamp)

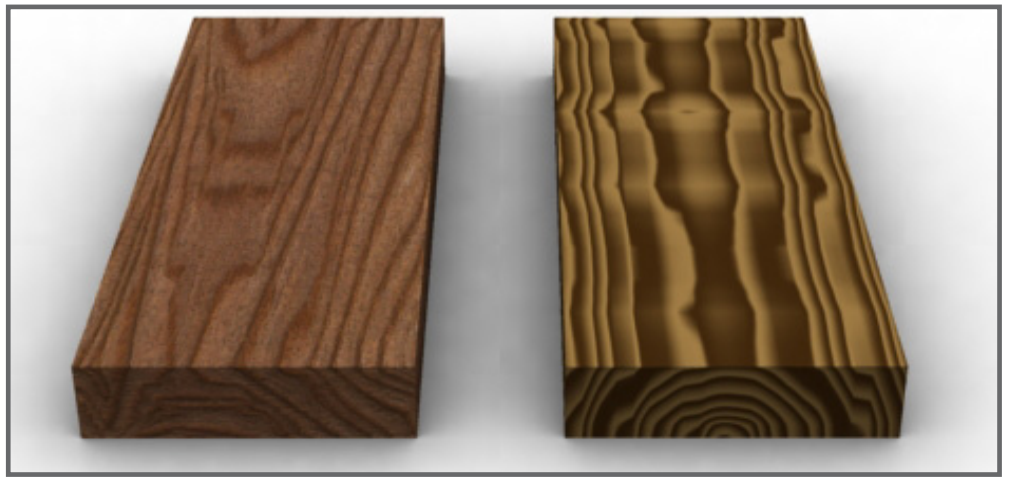
$$C(u) = \begin{cases} u, \ u \in [0; 1] \\ 0, u < 0 \\ 1, u > 1 \end{cases}$$



10.1 TEXTURE-MAPPING

Prozedurale Texturen

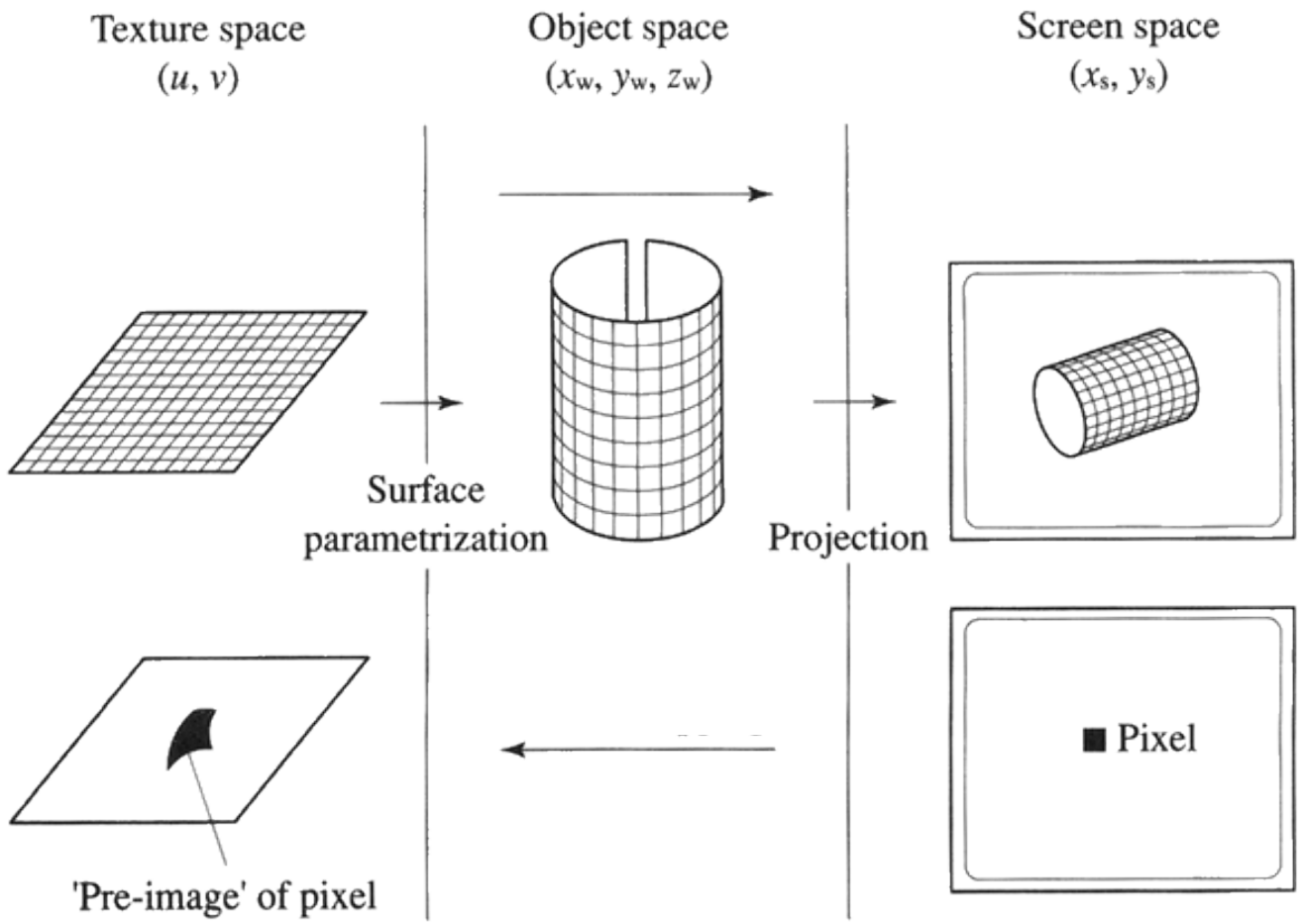
- Prozedural
  - Bei jedem Aufruf von $C_{textur}(u, v)$ wird
    - eine mathematische Formel ausgewertet
    - ein Algorithmus ausgeführt
- Vorteile
  - Minimaler Speicheraufwand
  - Texturen im gesamten Raum definiert
  - Auflösungsunabhängig
- Nachteile
  - Mathematische Beschreibung komplexer Texturen ist schwierig
  - Nicht photorealistisch



diskret prozedural

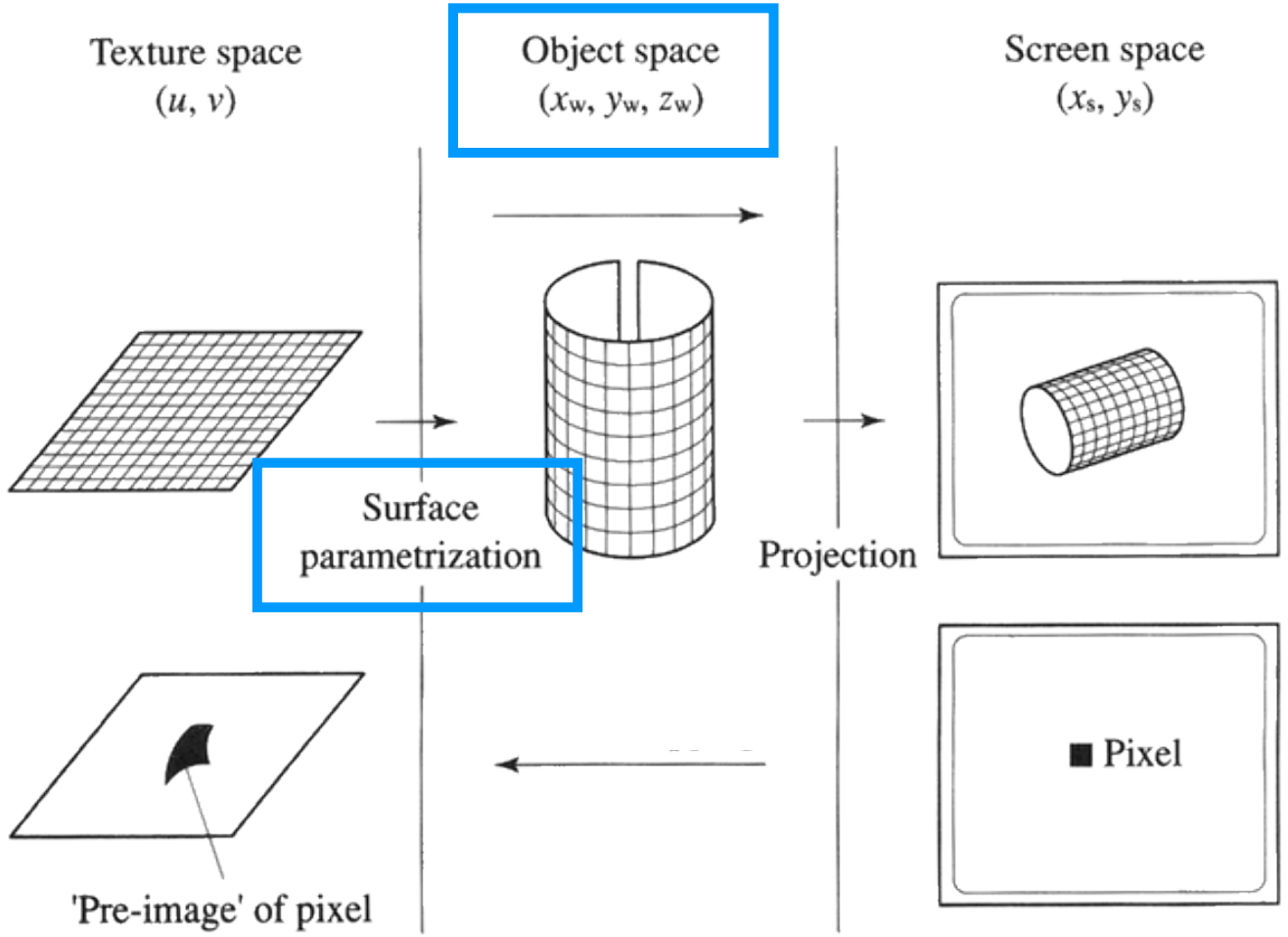
10.1 TEXTURE-MAPPING

Allgemeine Vorgehensweise



10.1 TEXTURE-MAPPING

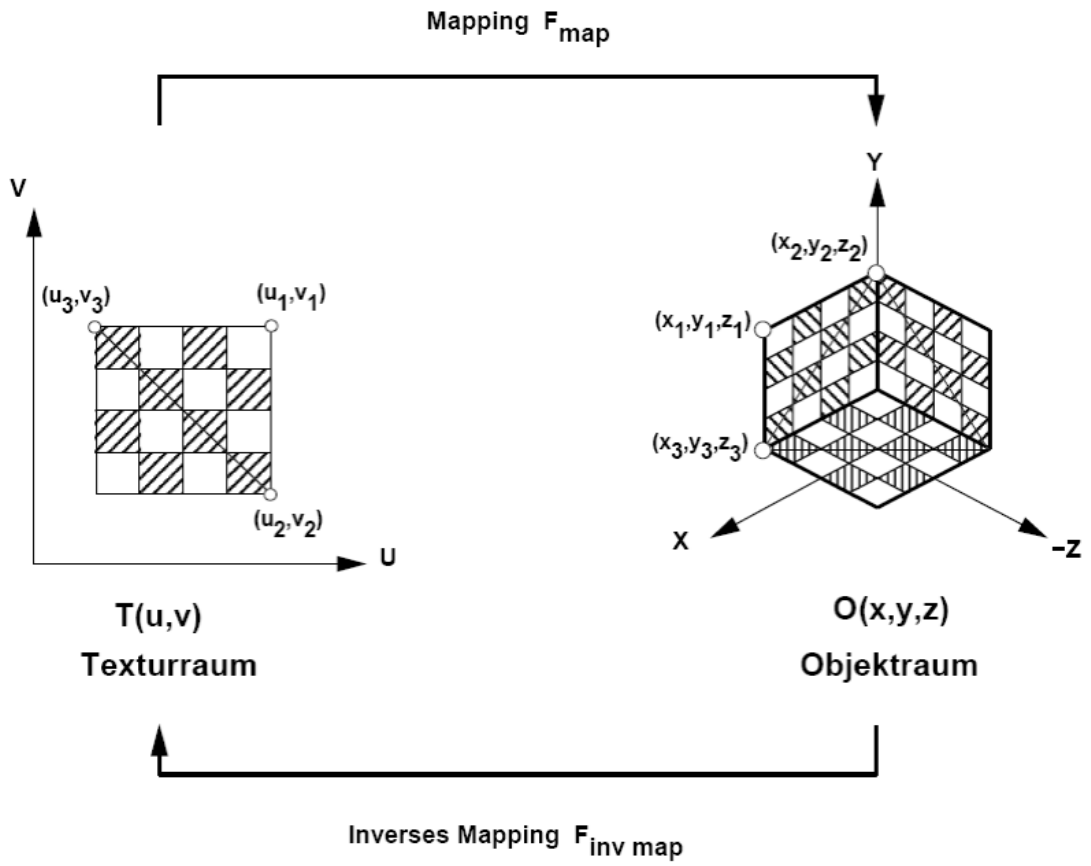
Schritt 1: Abbildung der Textur auf das Objekt



10.1 TEXTURE-MAPPING

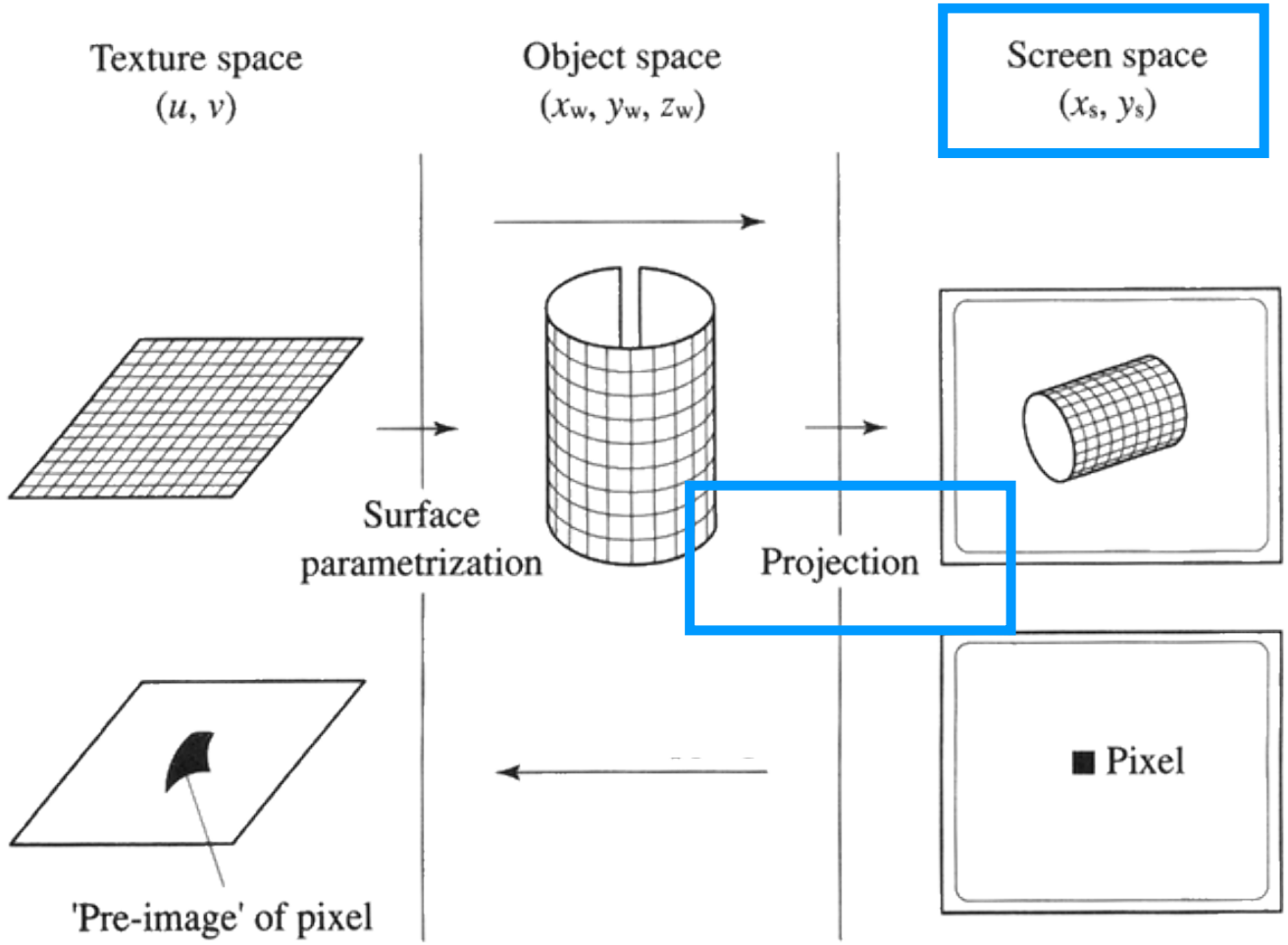
Schritt 1: Abbildung der Textur auf das Objekt

- Forward Mapping
  $$(x, y, z) = F_{map}(u, v)$$
- Zur Visualisierung muss das inverse Mapping Problem gelöst werden:
  $$(u, v) = F_{map}^{-1}(x, y, z)$$
- Texturierung entspricht mathematisch der Hintereinanderausführung von inversem Mapping und Textur-Funktion
  $$(r, g, b) = C_{texture}\left(F_{map}^{-1}(x, y, z)\right)$$



10.1 TEXTURE-MAPPING

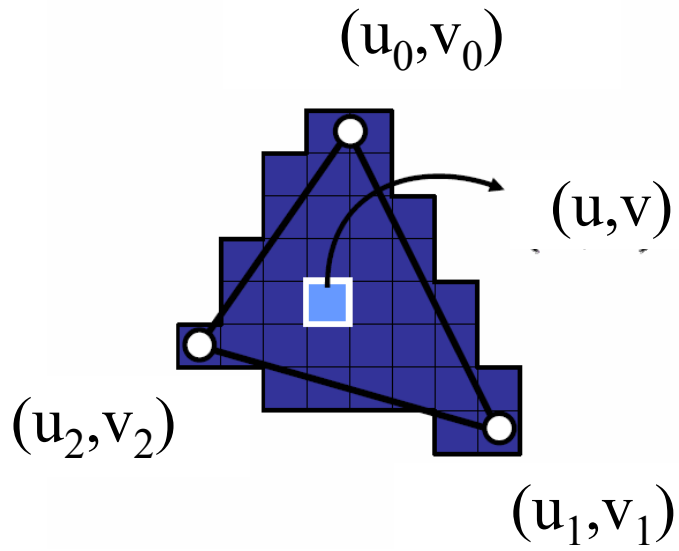
Schritt 2: Transformation in den Bildschirmbereich

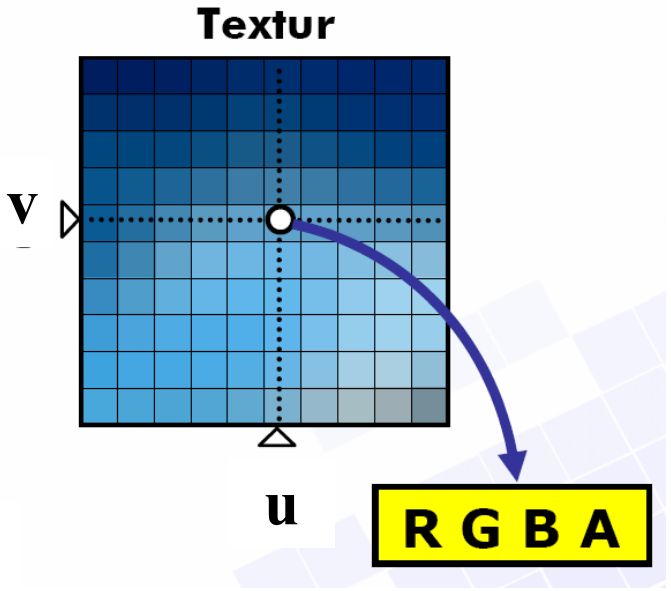


10.1 TEXTURE-MAPPING

Interpolation der Texturkoordinaten

1. Für jedes Pixel:
   - Interpolation der Texturkoordinaten
2. Texture-Lookup:
   - Interpolation der Texturwerte
   - Nearest Neighbour
   - Bilineare Interpolation



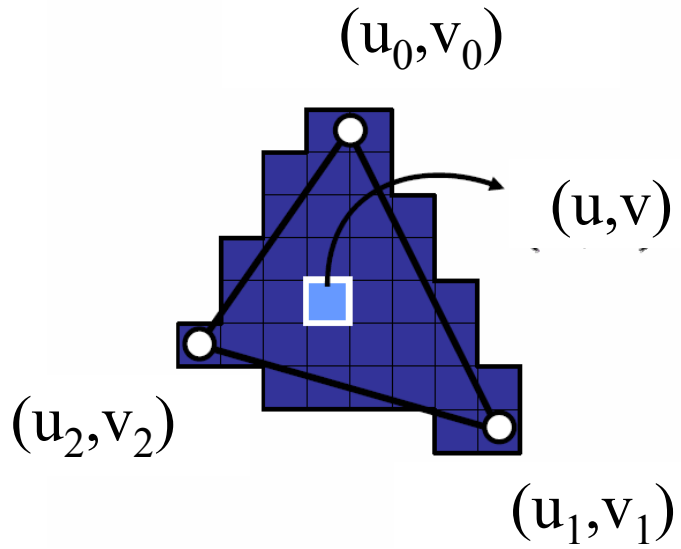


10.1 TEXTURE-MAPPING

Interpolation der Texturkoordinaten

- Abbildung Texturraum auf Objektraum
- Transformation in den Bildraum mit perspektivischer Transformation

- Inverses Mapping erweitert auf das Problem
  - Gegeben: Pixel
  - Gesucht: Punkt auf der Textur



- Information
  - Für jede Polygonecke eine Texturkoordinate

- Lösung:
  - Interpolation der Koordinaten

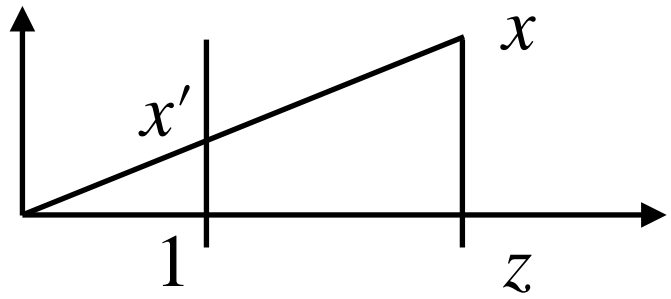
10.1 TEXTURE-MAPPING

Interpolation der Texturkoordinaten

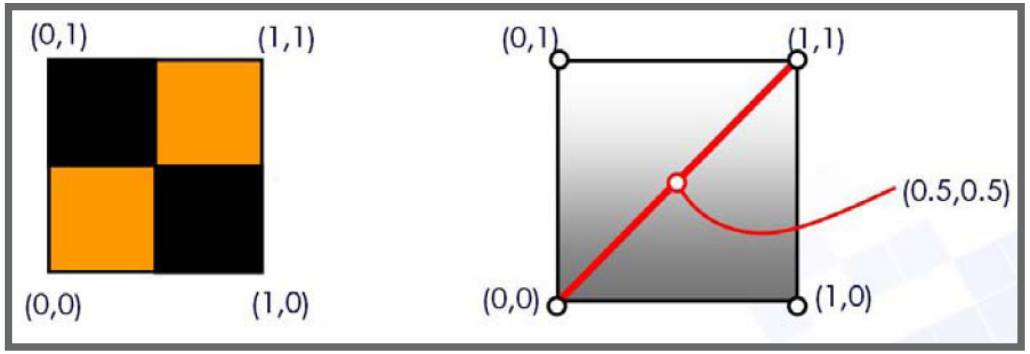
- Erste Idee:
  - Transformiere das Objekt und die Texturzuordnung bleibt dieselbe
  - Stimmt für die Eckpunkte
  - Problem bei der Interpolation, also der Zuordnung der Texturkoordinaten im Inneren eines Polygons bzw. Dreiecks
  - Texturkoordinaten müssen ebenfalls perspektivisch transformiert werden

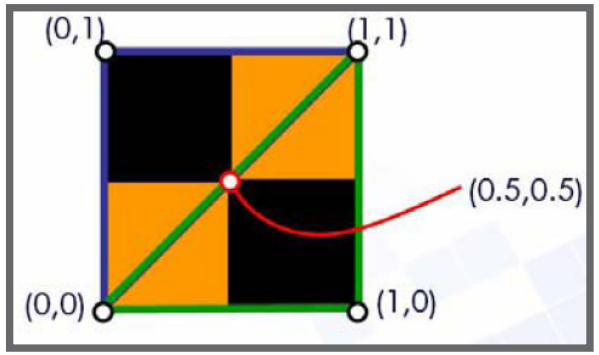
10.1 TEXTURE-MAPPING

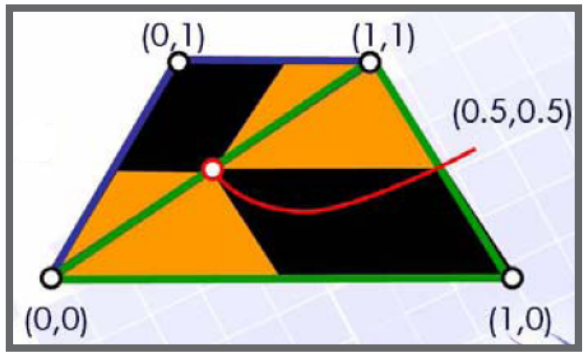
Interpolation der Texturkoordinaten

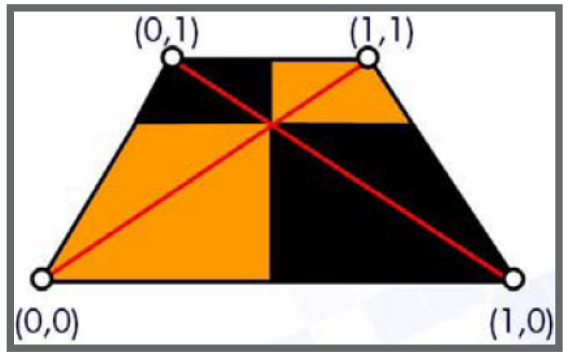


- Perspektivische Texturierung
  - Lineare Interpolation der Texturkoordinaten liefert falsches Ergebnis
  - Texturkoordinaten müssen ebenfalls perspektiv transformiert werden



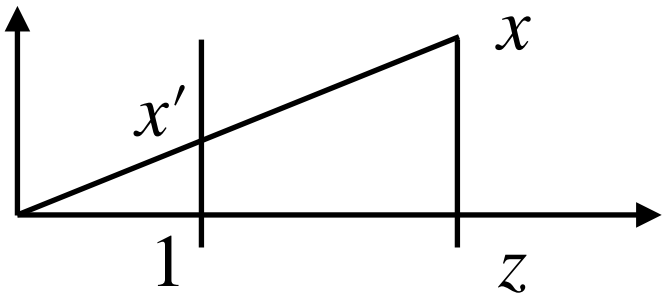






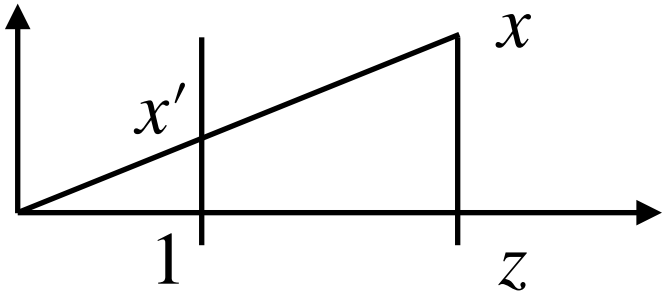
10.1 TEXTURE-MAPPING

Interpolation der Texturkoordinaten

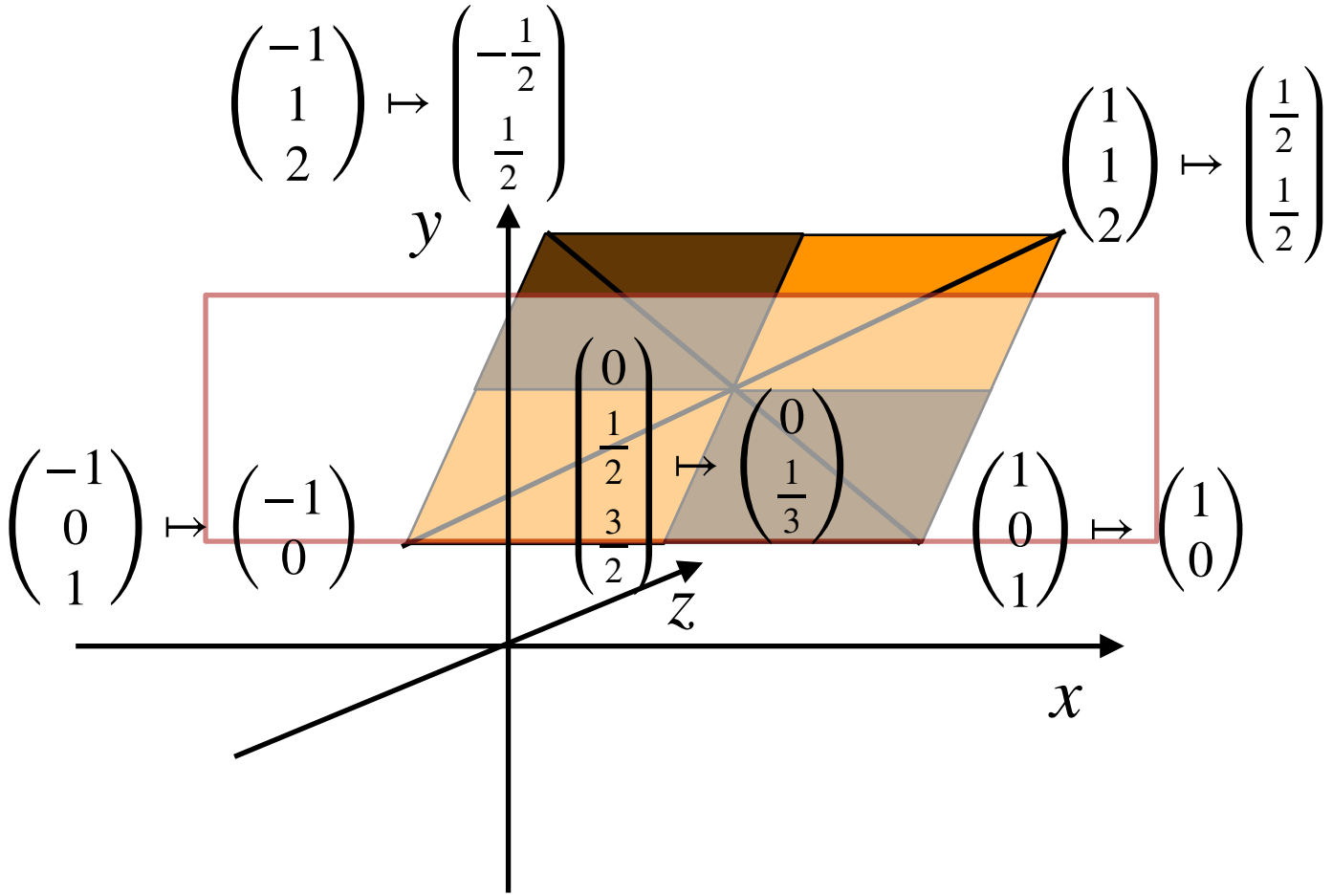


- Perspektivische Texturierung
  - Perspektivische Transformation von Punkt $p = (x, y, z, 1)$ in homogenen Koordinaten ergibt:
    $p' = (x, y, z, z)$
  - Das entspricht dem Punkt im Raum:
    $p' = (\frac{x}{z}, \frac{y}{z}, \frac{z}{z})$
  - Lineare Interpolation der Texturkoordinaten liefert falsches Ergebnis
- Lösung
  - Transformation der Texturkoordinaten $t = (u, v, 1)$ in 2D homogenen Koordinaten ergibt $t = (u, v, z)$
  - Lineare Interpolation der Texturkoordinaten innerhalb des Polygons
  - Textur-Abfrage für jedes Pixel mit rücktransformierten Texturkoordinaten
    $t' = (\frac{u}{z}, \frac{v}{z})$

10.1 TEXTURE-MAPPING

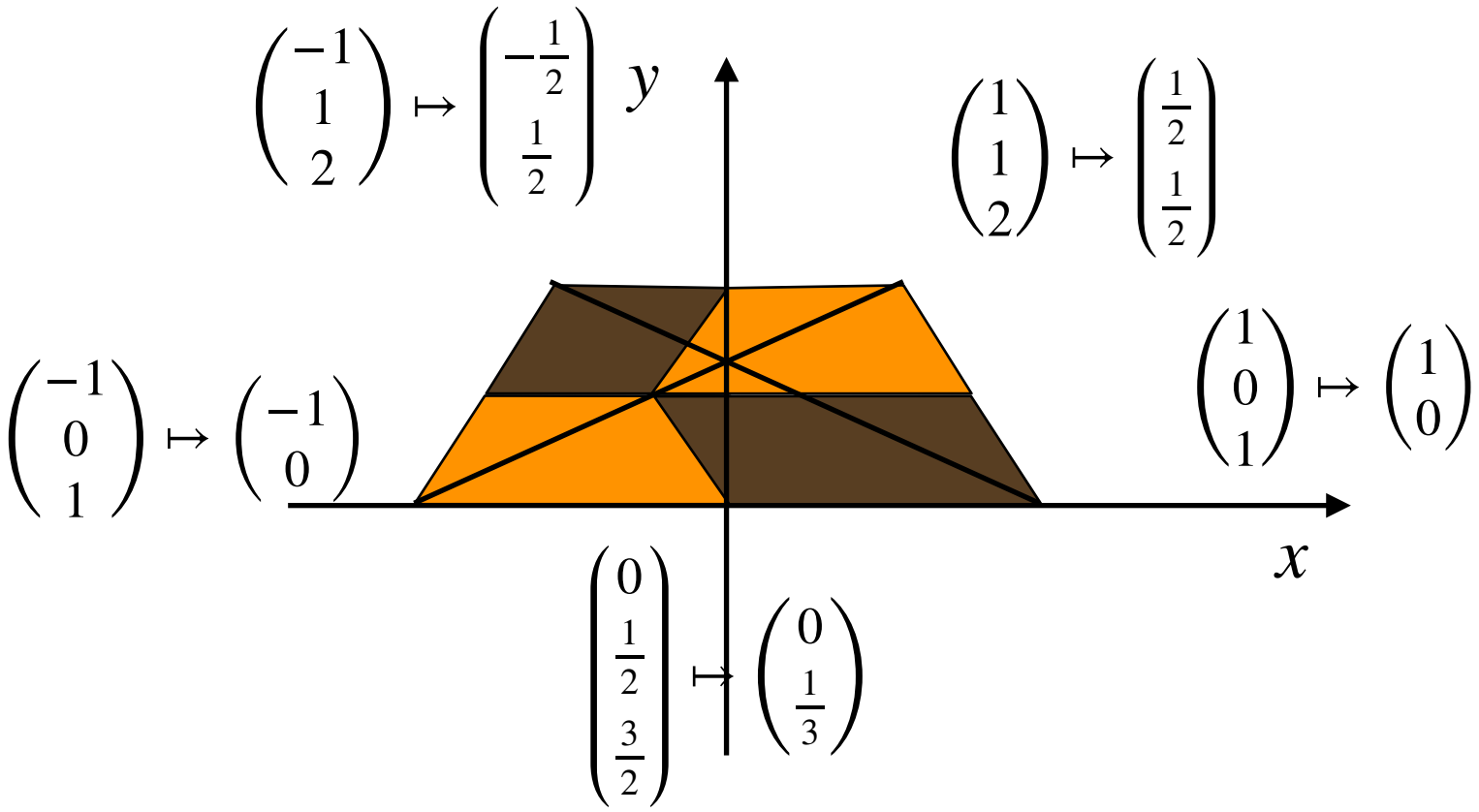


Interpolation der Texturkoordinaten



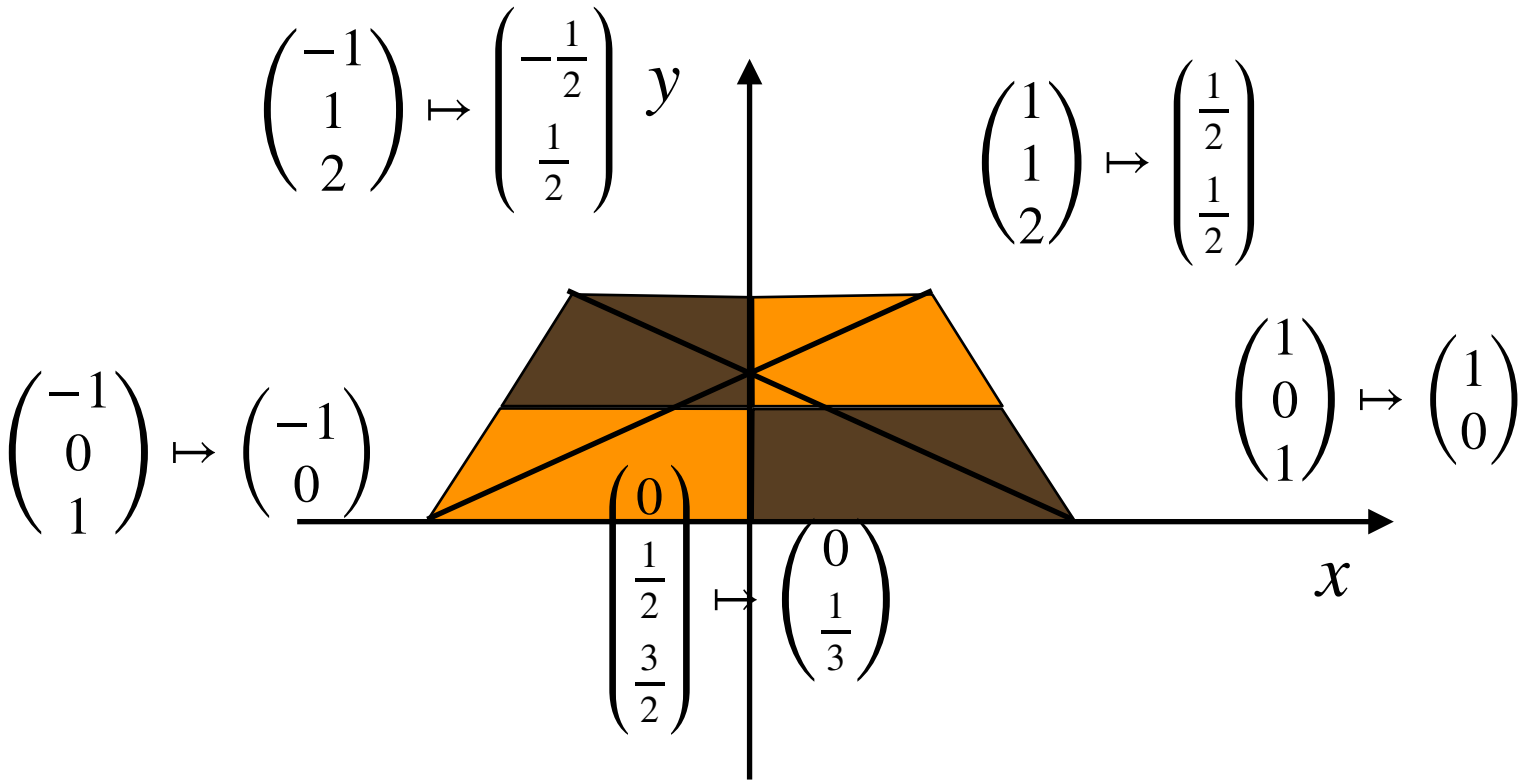
10.1 TEXTURE-MAPPING

Interpolation der Texturkoordinaten: Linear

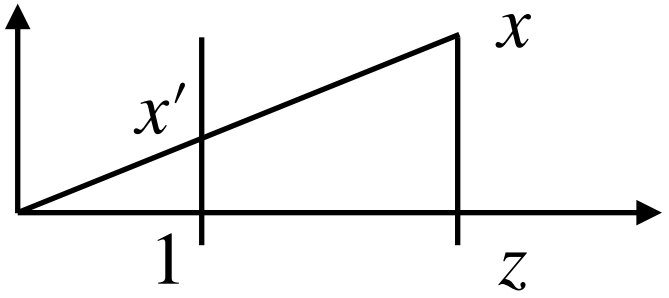


10.1 TEXTURE-MAPPING

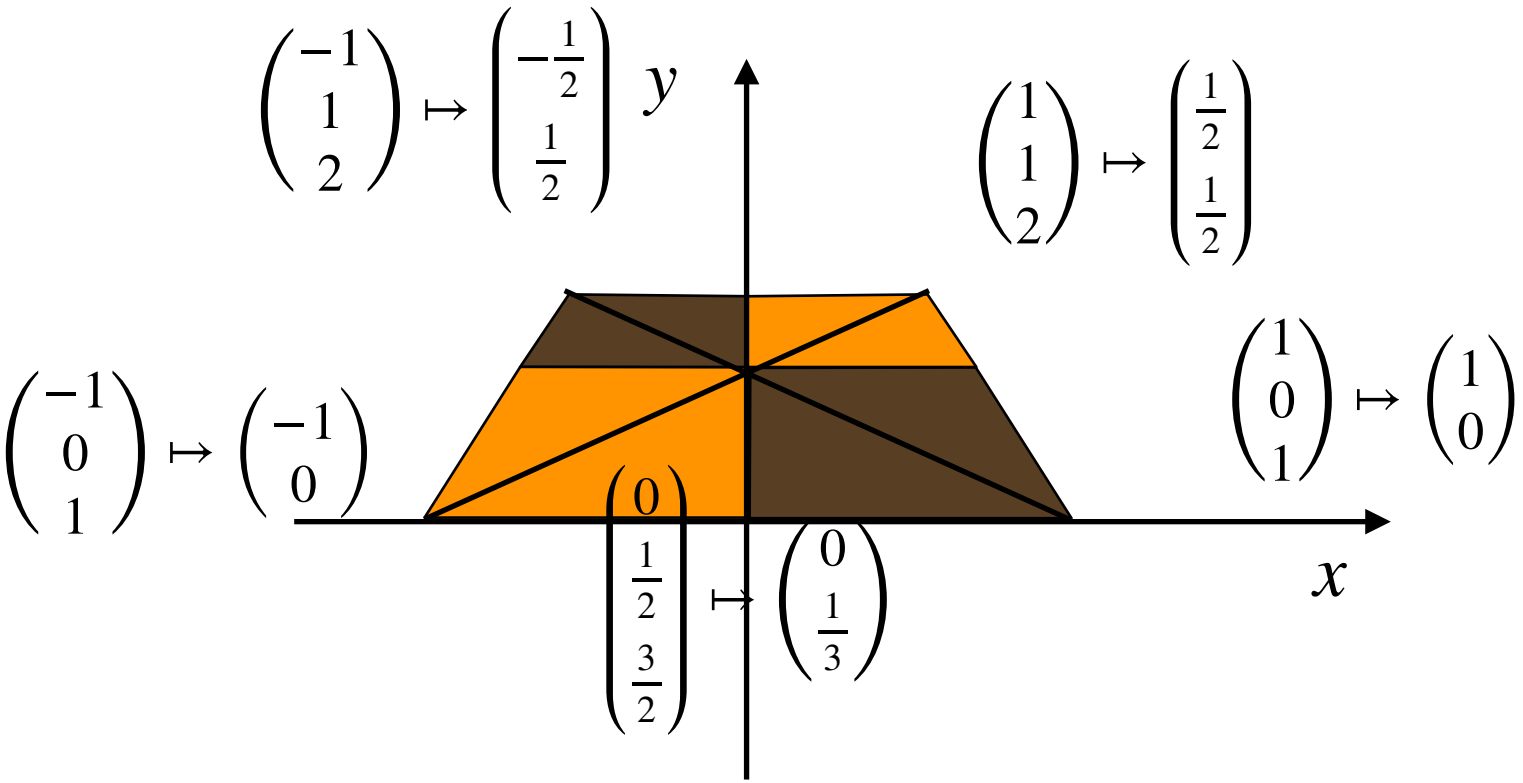
Interpolation der Texturkoordinaten: Bilinear



10.1 TEXTURE-MAPPING



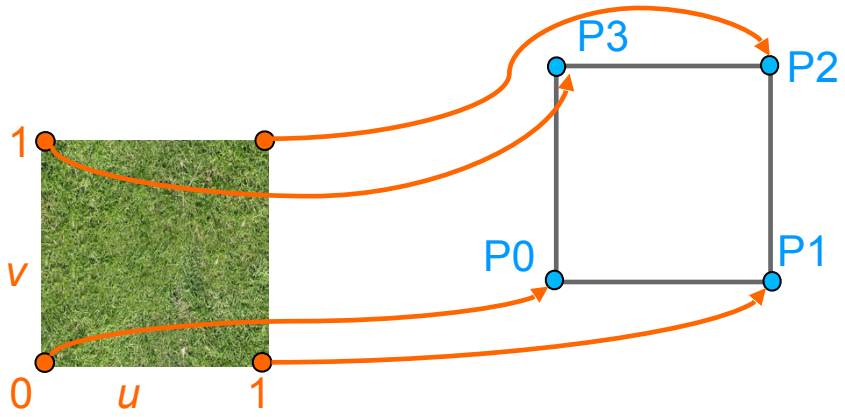
Interpolation der Texturkoordinaten: Perspektivisch



10.1 TEXTURE-MAPPING

Zuordnung von Polygonecken und Texturkoordinaten

- Bei einfachen Objekten u.U. manuell

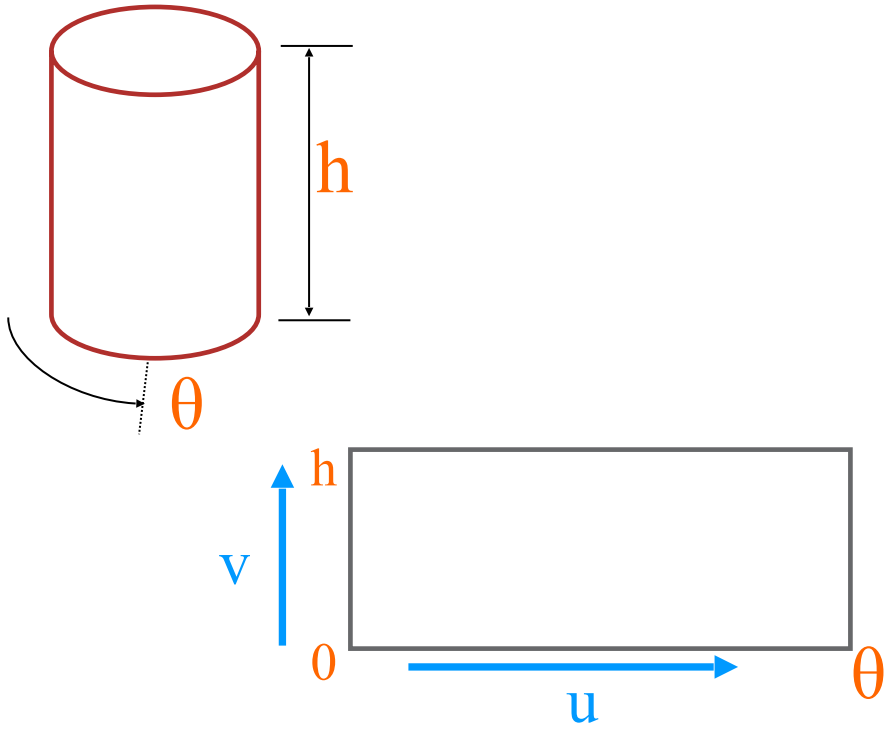


- Bei komplexeren Objekten: Two-Part Mapping
  - S-Mapping (Surface-Mapping) Abbildung der Textur auf eine einfache, virtuelle Zwischenfläche (z.B. Quader, Zylinder, Kugeln)
  - O-Mapping (Object-Mapping) Übertragung von der umhüllenden Zwischenfläche auf das zu texturierende Objekt

10.1 TEXTURE-MAPPING

Zylinder-Mapping

- Geeignet für rotationssymmetrische Objekte
- Diskontinuität an der Naht (parallel zur Achse)
- Parametrisierung $(u, v) \mapsto \left(\Theta, h\right)$



10.1 TEXTURE-MAPPING

Zylinder-Mapping

- Geeignet für rotationssymmetrische Objekte
- Diskontinuität an der Naht (parallel zur Achse)
- Parametrisierung $(u, v) \mapsto \left(\Theta, h\right)$

10.1 TEXTURE-MAPPING

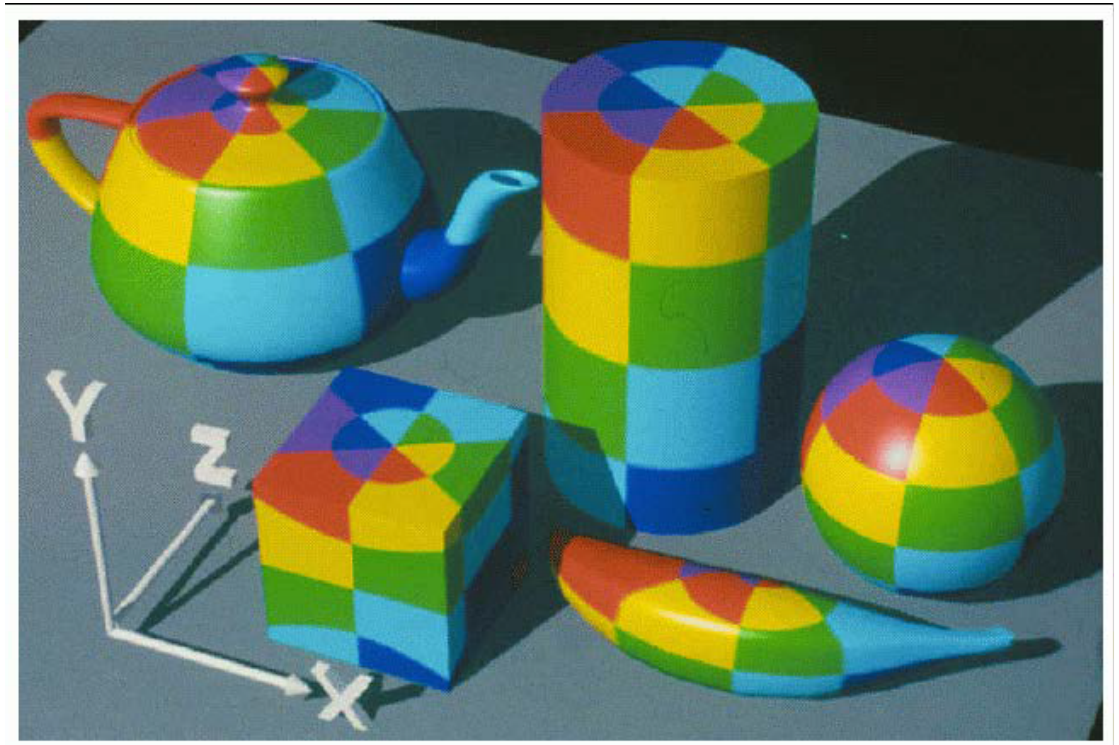
Kugel-Mapping

- Geeignet für Kugel-symmetrische Objekte
- Verzerrung, vor allem an den Polen
- Parametrisierung durch Kugelkoordinaten $(u, v) \mapsto (\Phi, \Theta)$

10.1 TEXTURE-MAPPING

Kugel-Mapping

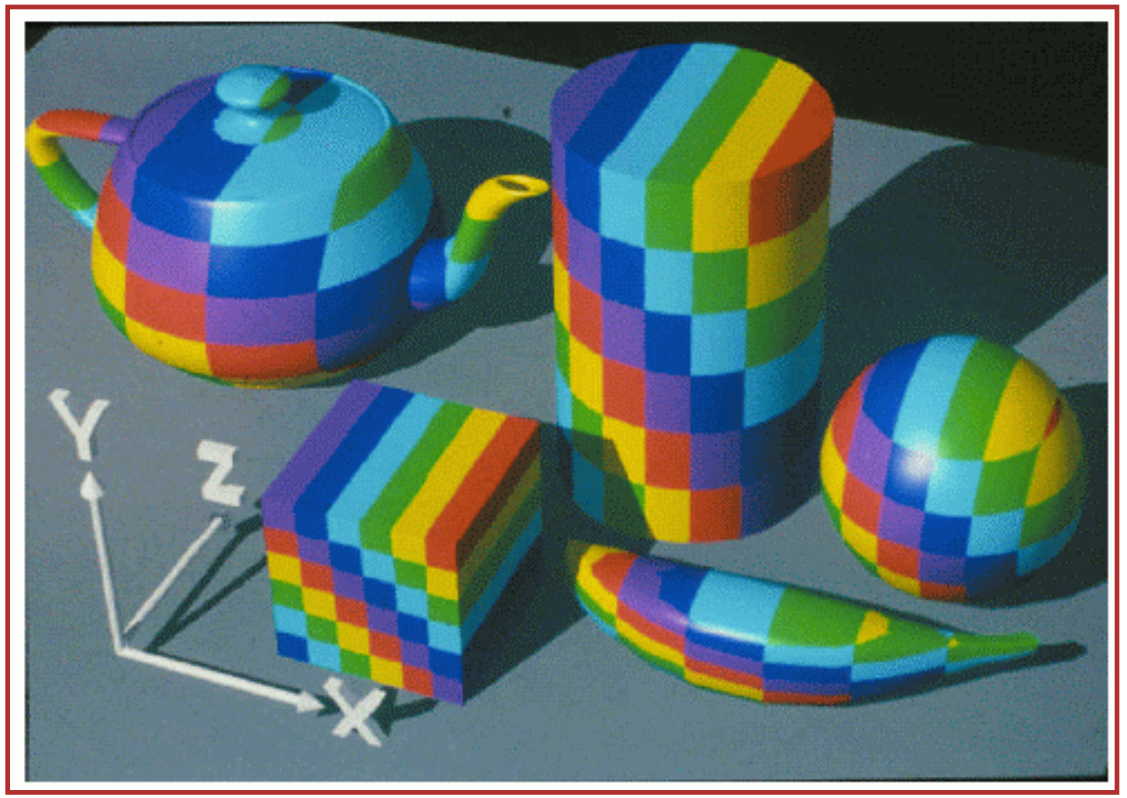
- Geeignet für Kugel-symmetrische Objekte
- Verzerrung, vor allem an den Polen
- Parametrisierung durch Kugelkoordinaten $(u, v) \mapsto (\Phi, \Theta)$



10.1 TEXTURE-MAPPING

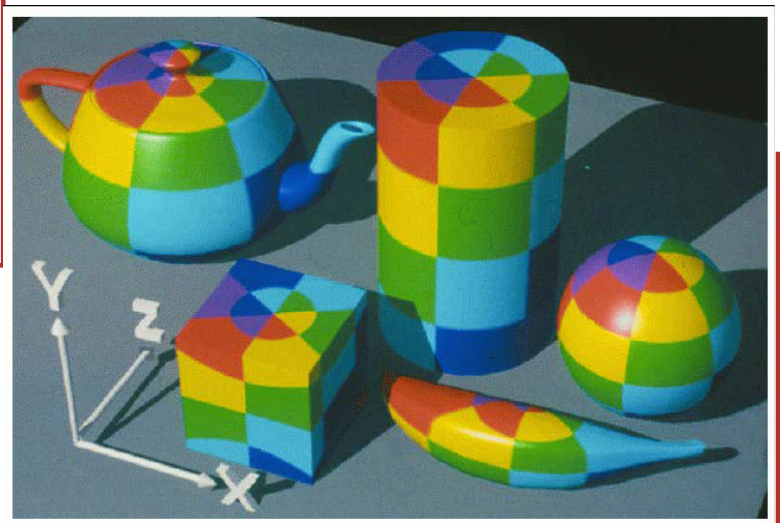
Box-Mapping

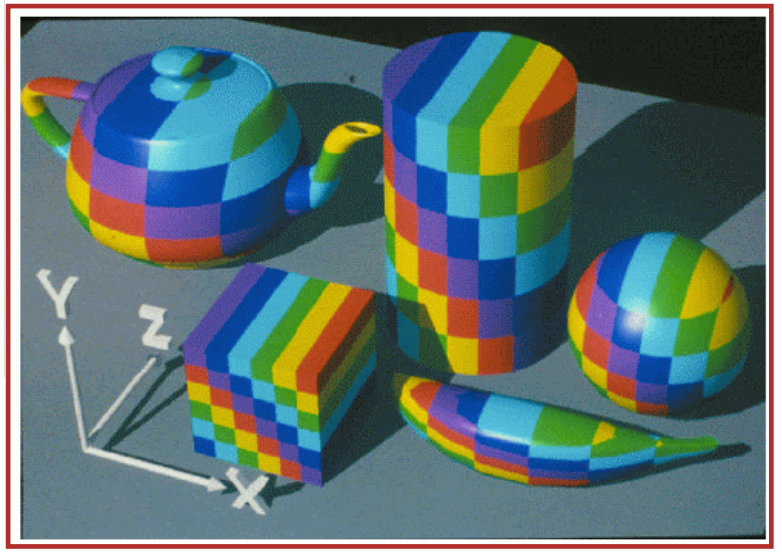
- Quader als umhüllende Fläche
  - meist achsenparallele Bounding-Box des Objekts
- Mögliche Parametrisierung
  - u-Achse:
    Längste Kante des Quaders
  - v-Achse:
    Zweitlängste Kante des Quaders



10.1 TEXTURE-MAPPING

Zylinder – Kugel – Box-Mapping

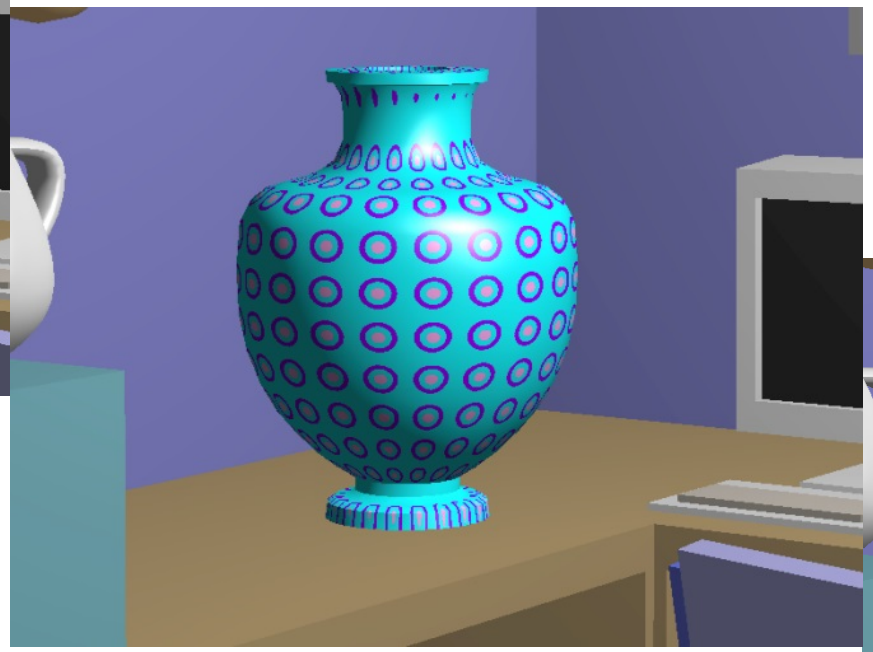




10.1 TEXTURE-MAPPING

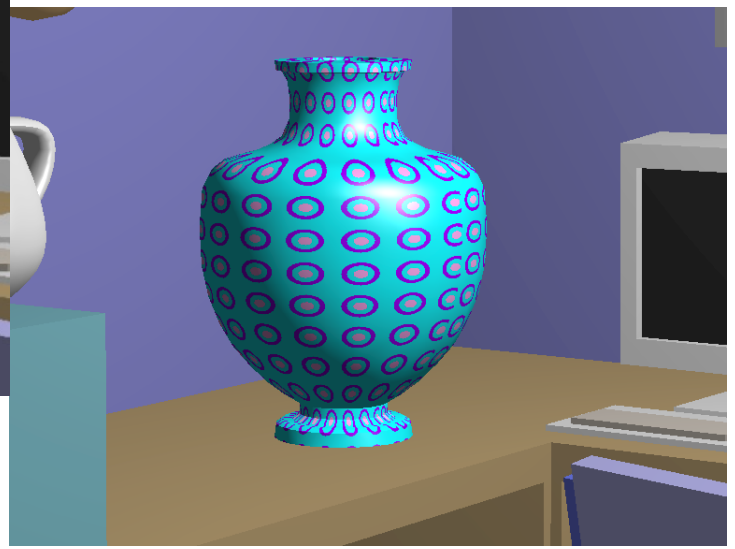
Verwendung verschiedener Zwischenobjekte

Box



Kugel

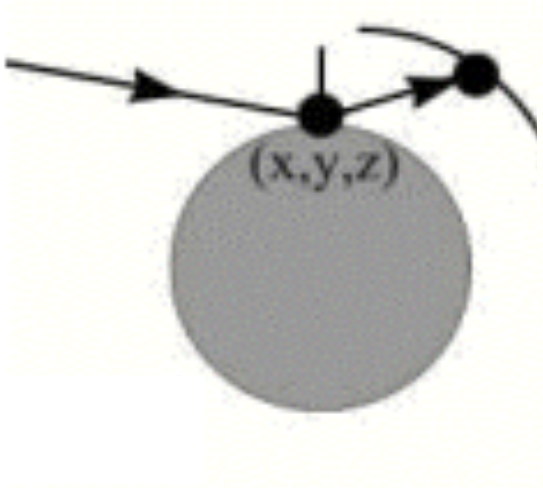
Zylinder



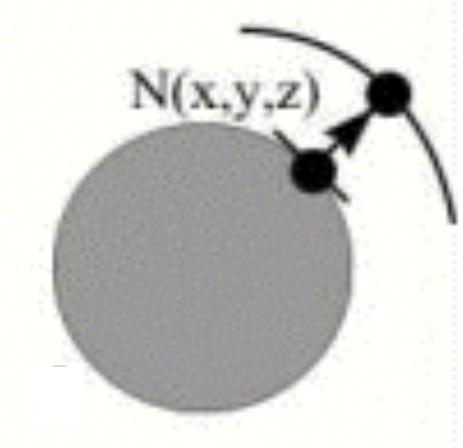
10.1 TEXTURE-MAPPING

Techniken des O-Mapping

Reflexionsstrahl

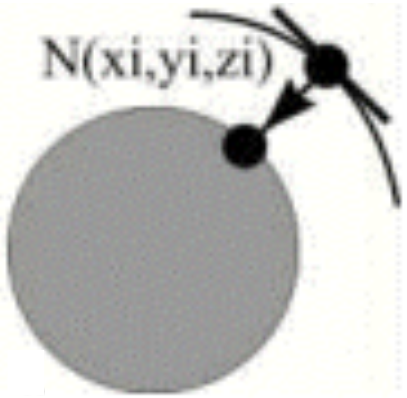


Normalenvektor



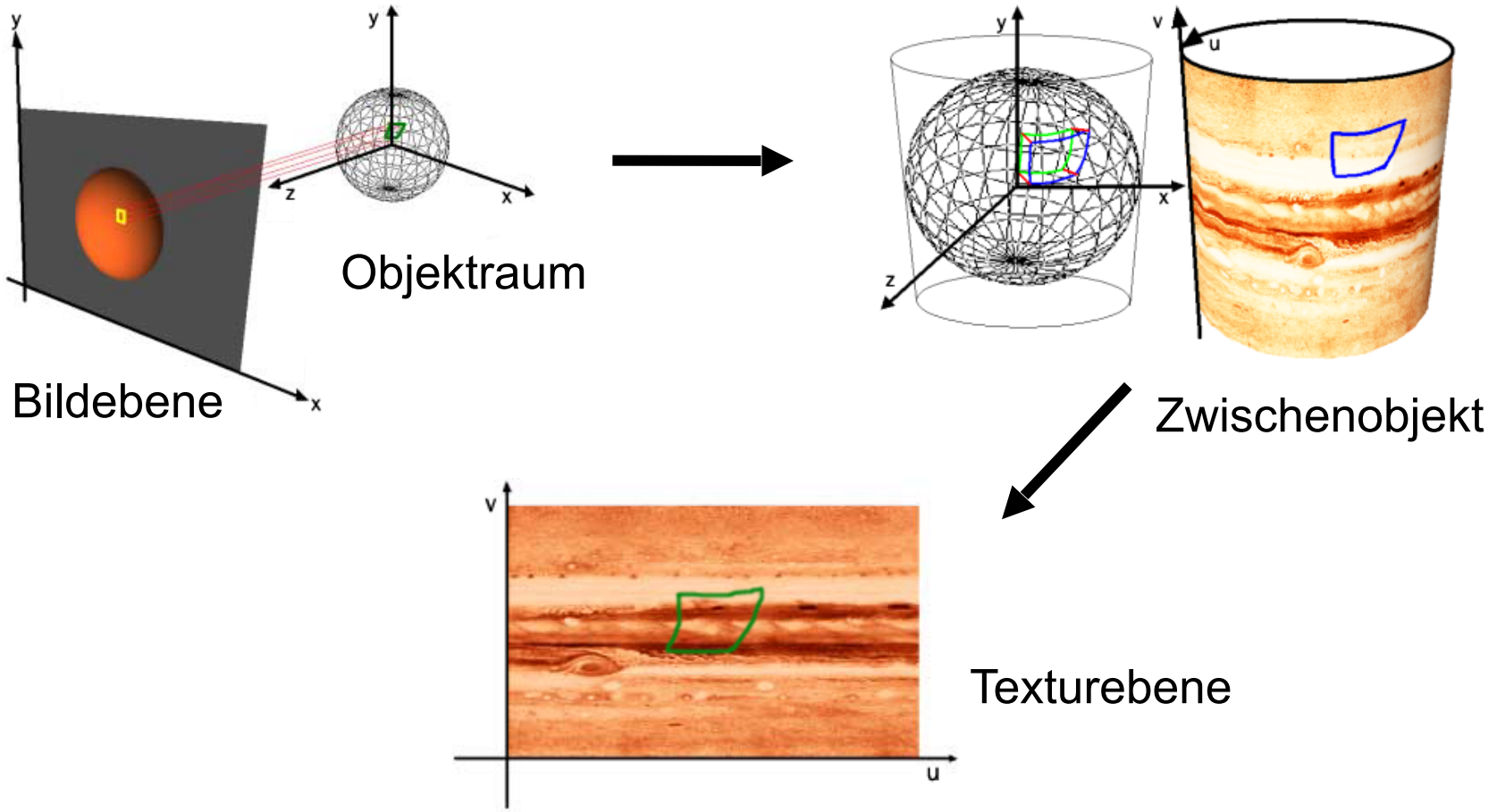
Objektzentrum

Hilfsobjektnormale



10.1 TEXTURE-MAPPING

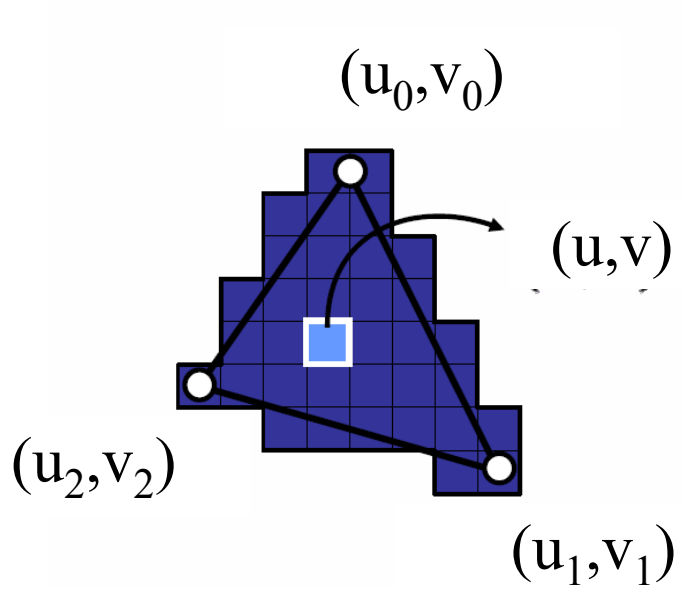
Inverse Abbildung mit Zwischenobjekt



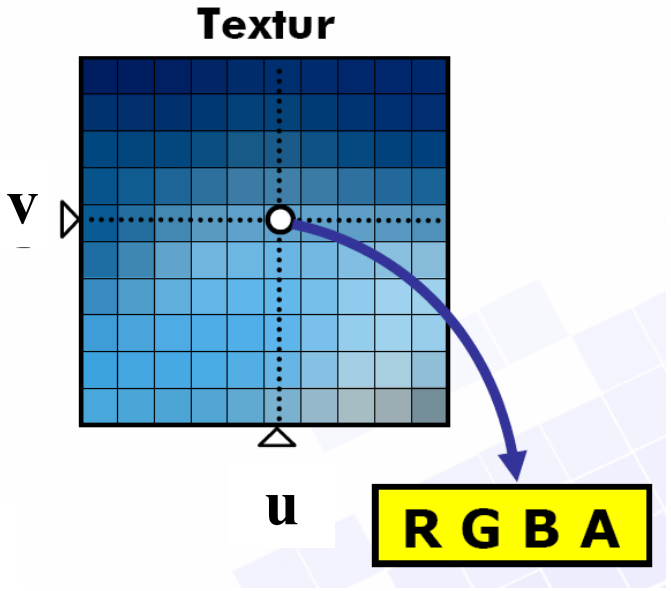
10.1 TEXTURE-MAPPING

Abtastprobleme

- Welche Probleme können bei der Abbildung von Pixel- auf Textur-Raster auftreten?



Pixel-Raster

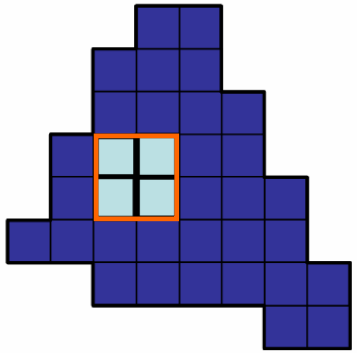


Textur-Raster

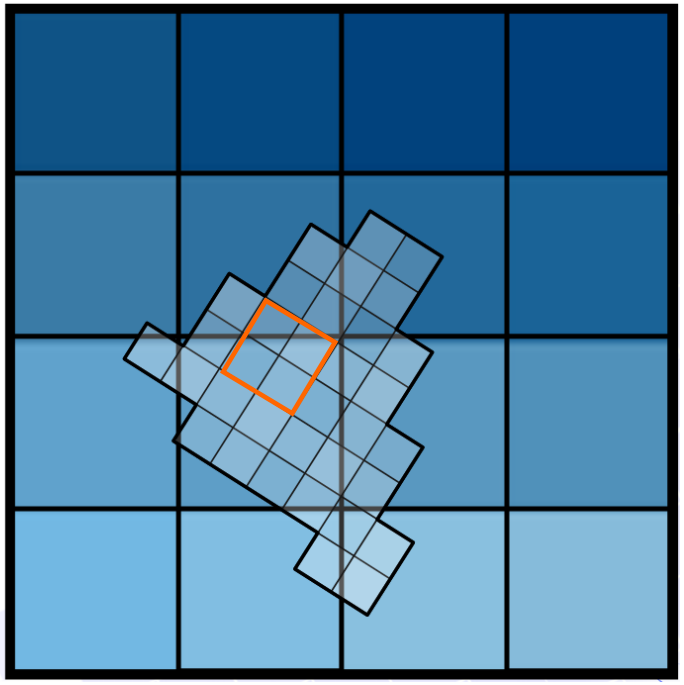
10.1 TEXTURE-MAPPING

Abtastprobleme

- Magnification / Oversampling
  - Pixel-Raster ist feiner als Textur-Raster
  - Ein Texel wird auf mehrere Pixel abgebildet.
  - Texturen erscheinen daher verschwommen.
- Abhilfe
  - Auflösung der Textur erhöhen



Pixel-Raster

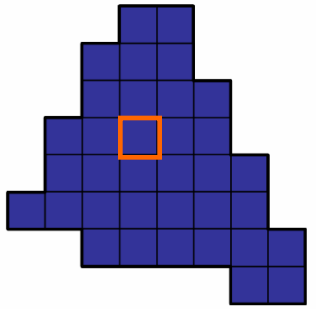


Textur-Raster

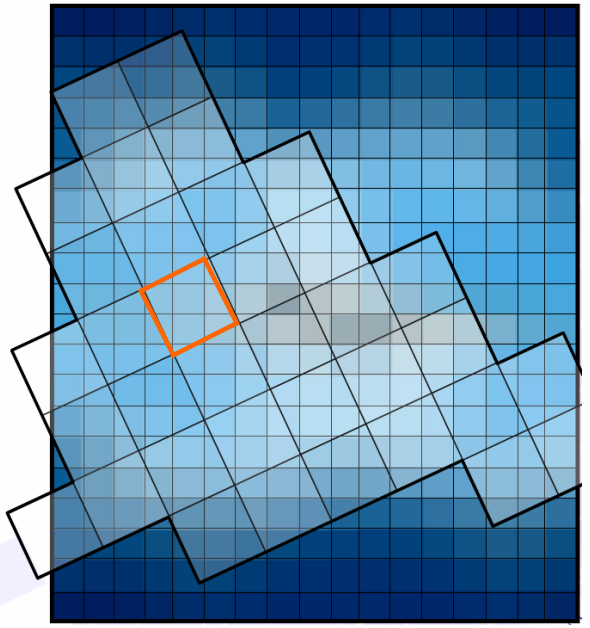
10.1 TEXTURE-MAPPING

Abtastprobleme

- Minification / Undersampling
  - Pixel-Raster ist gröber als Textur-Raster
  - Ein Pixel wird auf mehrere Texel abgebildet
  - Aliasing, da Abtastfrequenz zu gering
- Abhilfe
  - Nach Fläche gewichteten Mittelwert der Texel bestimmen
  - Exakte, aber zu aufwändige Lösung



Pixel-Raster



Textur-Raster

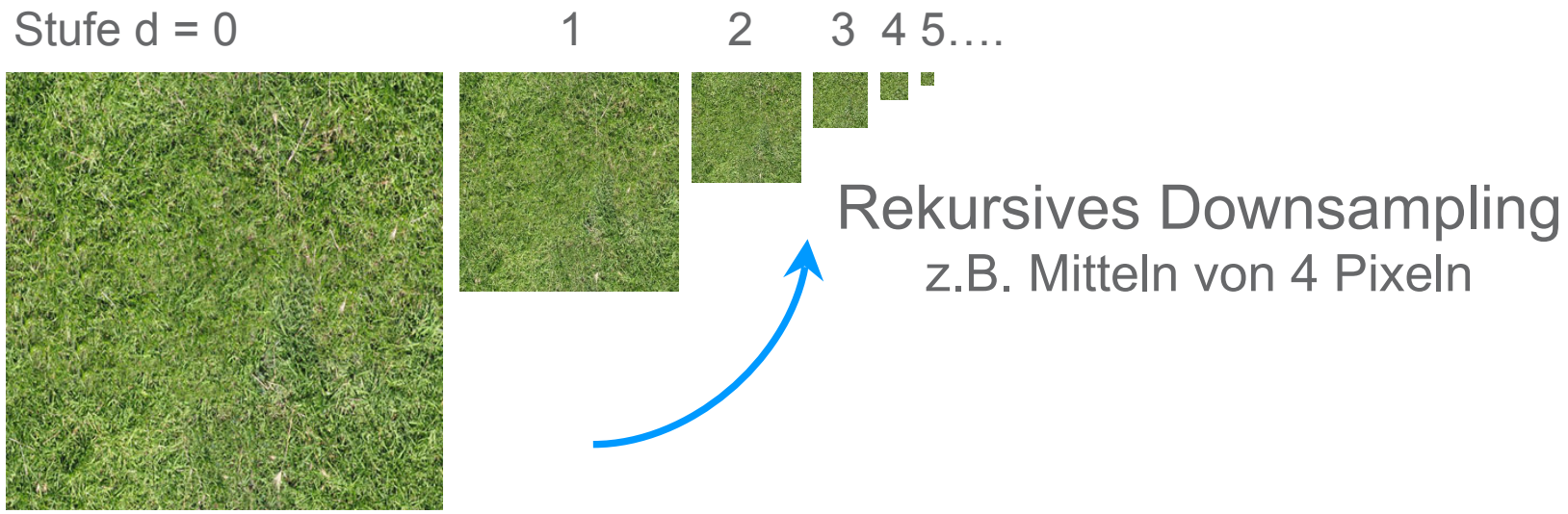
10.1 TEXTURE-MAPPING

Mip-Mapping

- Auflösung der Textur sollte der im Bildraum entsprechen
- Problematisch, wenn Auflösung im Bildraum variiert
- Lösung
  - Textur in verschiedenen Auflösungen bereithalten
  - Quadratische Texturen mit Kantenlänge 2^k (Zweierpotenz)

10.1 TEXTURE-MAPPING

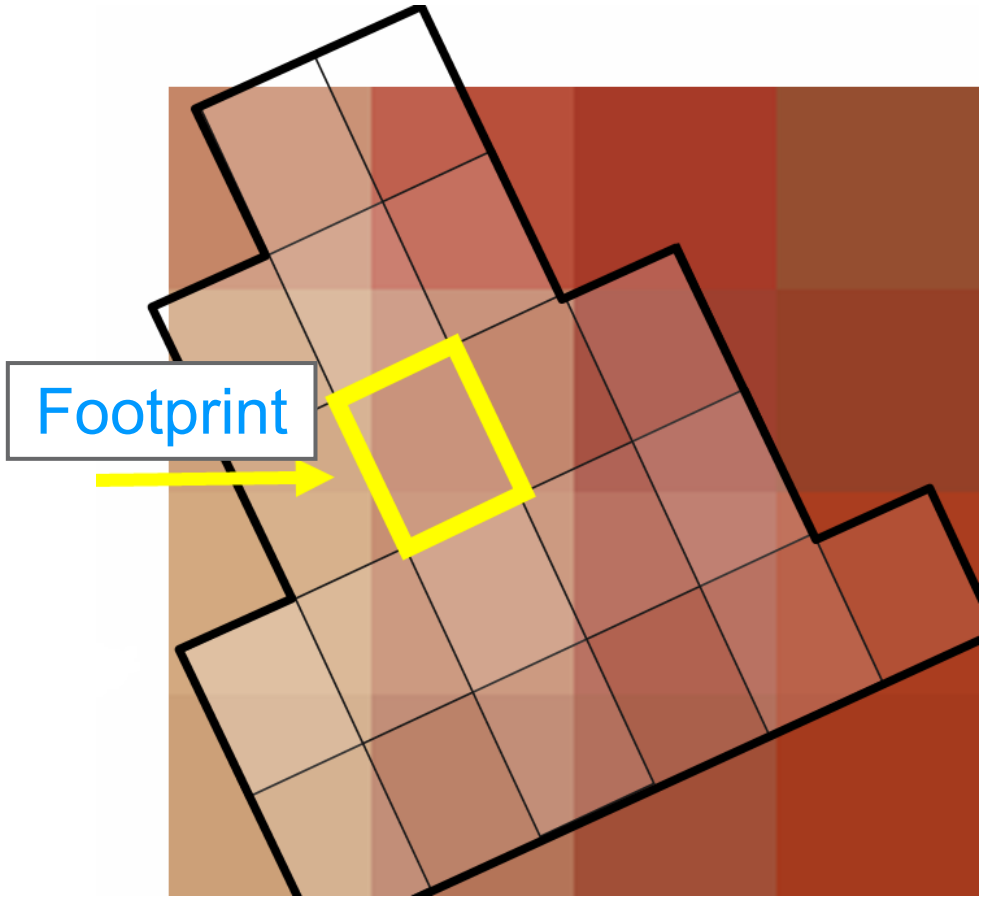
Mip-Mapping



10.1 TEXTURE-MAPPING

Mip-Mapping

- Verfahren
  - Bestimme Footprint des Pixels auf der Textur
  - Längste Kante bestimmt Mipmap-Stufe d
  - Interpoliere Texel aus Textur der Stufe d
- Ergebnis
  - Antialiasing, da der verwendete Texel immer in etwa der Größe des Footprints entspricht



10.1 TEXTURE-MAPPING

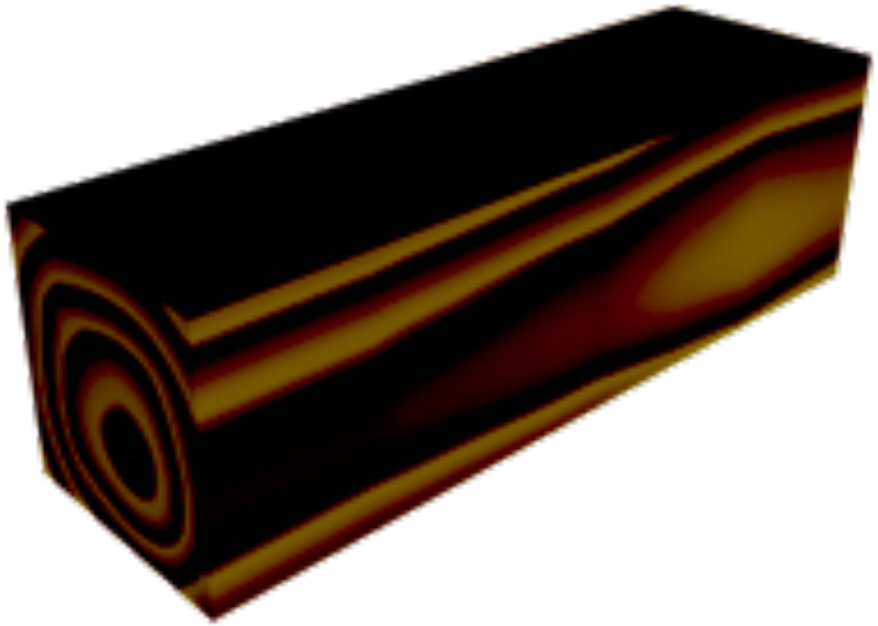
3D-Texturen / Festkörpertexturen

- Handhabung analog zu 2D-Texturen
- (Anschaulicher) Unterschied zu 2D-Texturen
  - Objekt wird aus Textur ausgeschnitten statt damit beklebt
  - Prozedurale Ansätze erlauben wirklichkeitsgetreue 3D-Muster

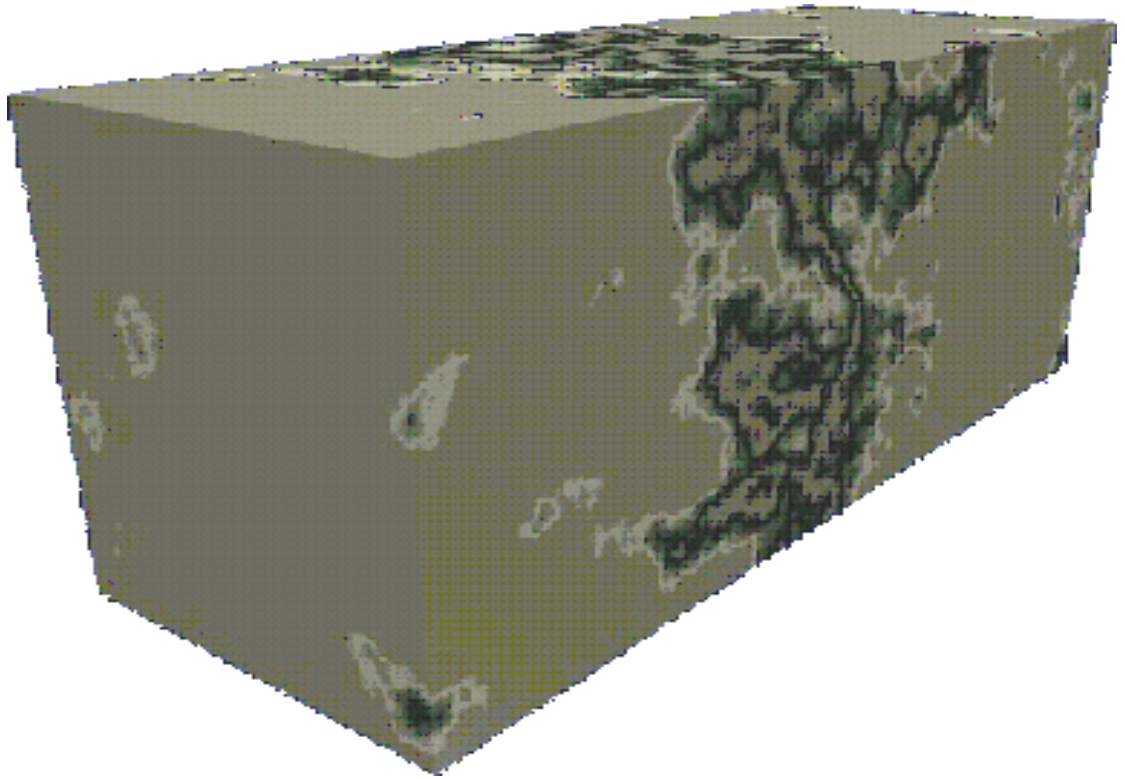
10.1 TEXTURE-MAPPING

3D-Texturen / Festkörpertexturen

- Holzmaserung



- Perlin Marmor

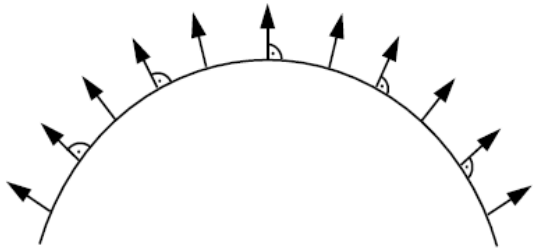


10.2 BUMP-MAPPING

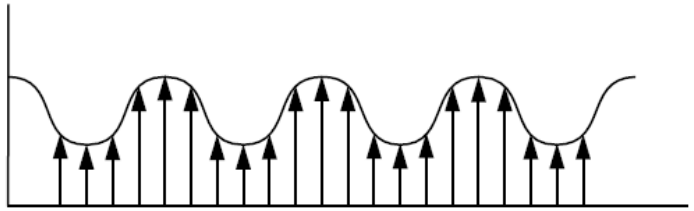
- James Blinn 1978
- Ziel
  - Rauere und plastischere Erscheinung eines Objekts
  - Keine Veränderung der Geometrie
- Simulation von Oberflächenunebenheiten durch
  - Manipulation der Normalenvektoren
  - Struktureffekt nur durch Beleuchtung

10.2 BUMP-MAPPING

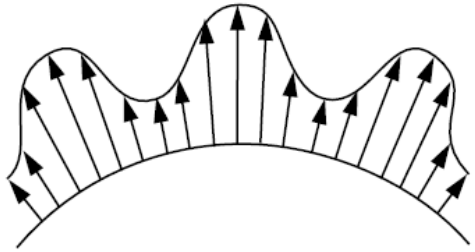
Exakte Modellierung



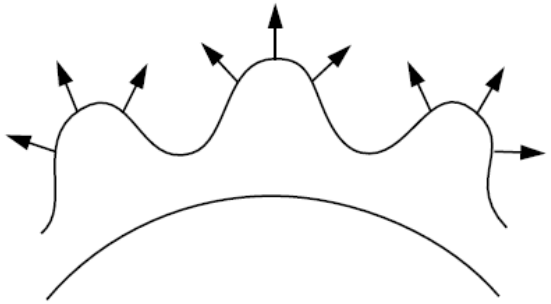
(a) Originalfläche **P**(u)
mit Normalen **N**(u)



(b) Bump Map h(u)



(c) Offsetfläche **P'**(u)



(d) Perturbierte Normalen **N'**(u)

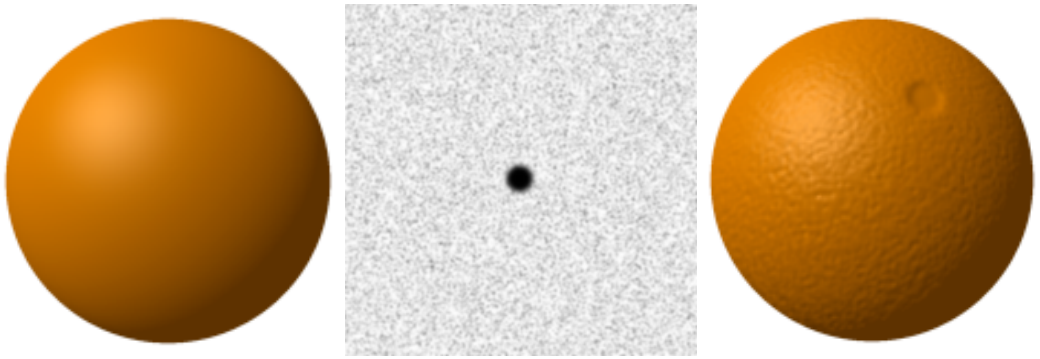
10.2 BUMP-MAPPING

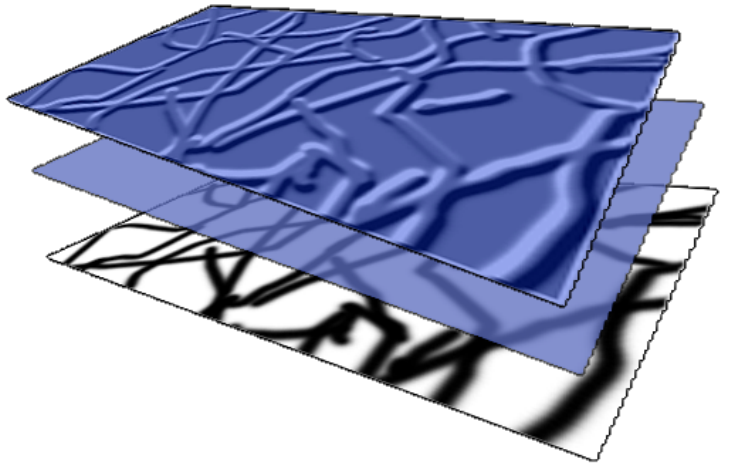
- Prinzip
  - Für die Beleuchtung ist nur die Normale relevant
  - Die Lage des Punktes auf der Fläche spielt keine Rolle
  - Originalgeometrie mit veränderten Normalen kombinieren
- Anmerkungen
  - Bump-Mapping nur mit Verfahren möglich, die Beleuchtung explizit in jedem Flächenpunkt auswerten
    - Phong-Shading
    - Raytracing
  - Nicht
    - Gouraud
  - Schatten und Silhouetten der Objekte bleiben glatt

10.2 BUMP-MAPPING

- Umsetzung
  - Prozedurale Veränderung der Normalen
  - Normalen als RGB-Textur gegeben (Normal Map)
  - Höhenfeld als 2D-Skalarfeld (Grauwerttextur) gegeben
    - Addition des Höhenfeldes auf die Geometrie
    - Berechnung der Normalen über Richtungsableitungen dieser Offsetfläche

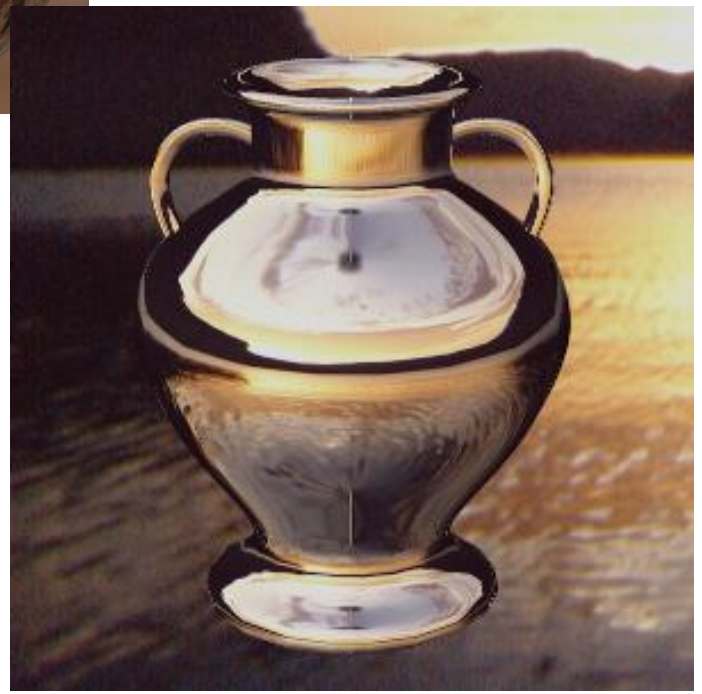
$$N' = P'_u \times P'_v$$





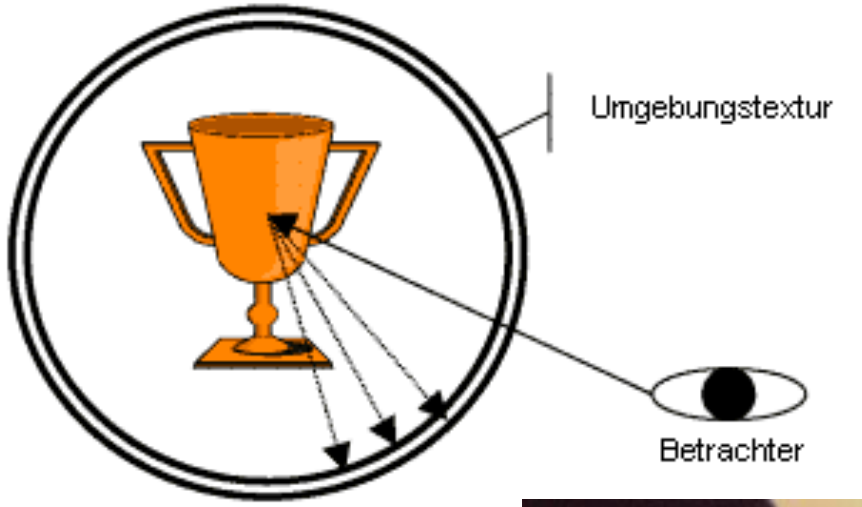
10.3 ENVIRONMENT-MAPPING

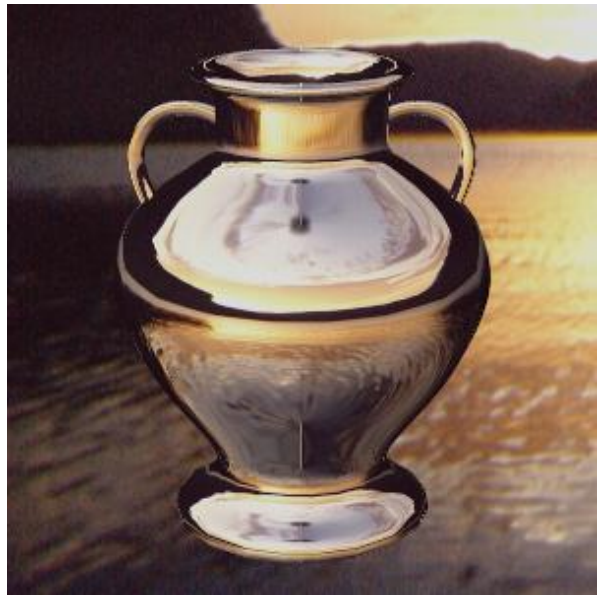
- Modellierung von Interobjekt-Reflexionen
- Abbildung einer komplexen Umgebung auf ein spiegelndes Objekt mit Hilfe von Texturen
- Günstige Alternative zum Raytracing
- Keine Verdeckungsrechnung
- Anwendung im Real-Time Rendering



10.3 ENVIRONMENT-MAPPING

- Verfahren
  - Zwischenobjekt (Würfel, Kugel) umhüllt Objekt
  - Die Umgebungstextur wird auf die Innenseite des Zwischenobjekts aufgetragen
  - Die Texturkoordinaten sind abhängig vom Reflexionsvektor

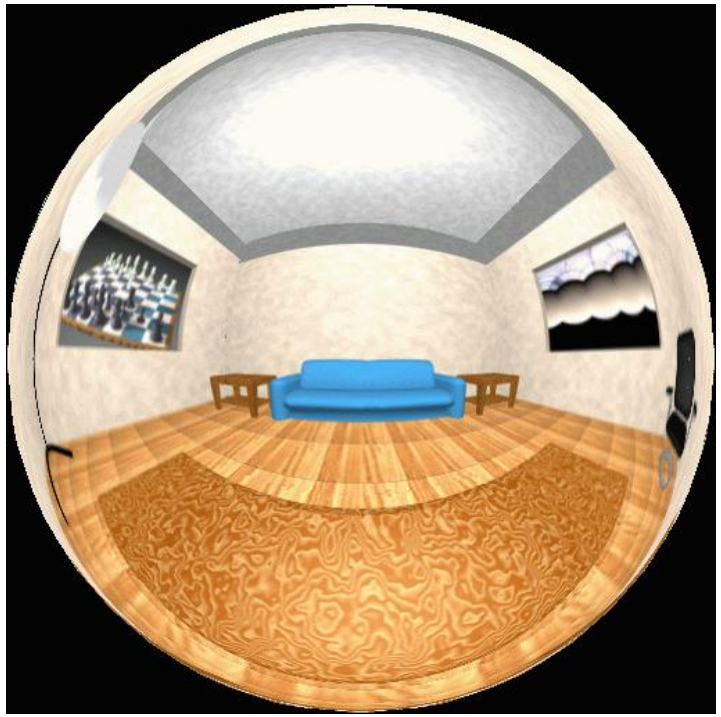




10.3 ENVIRONMENT-MAPPING

Sphere Mapping

- Unterstützt von OpenGL
- Abbildung einer umhüllenden Kugel auf eine 2D-Textur



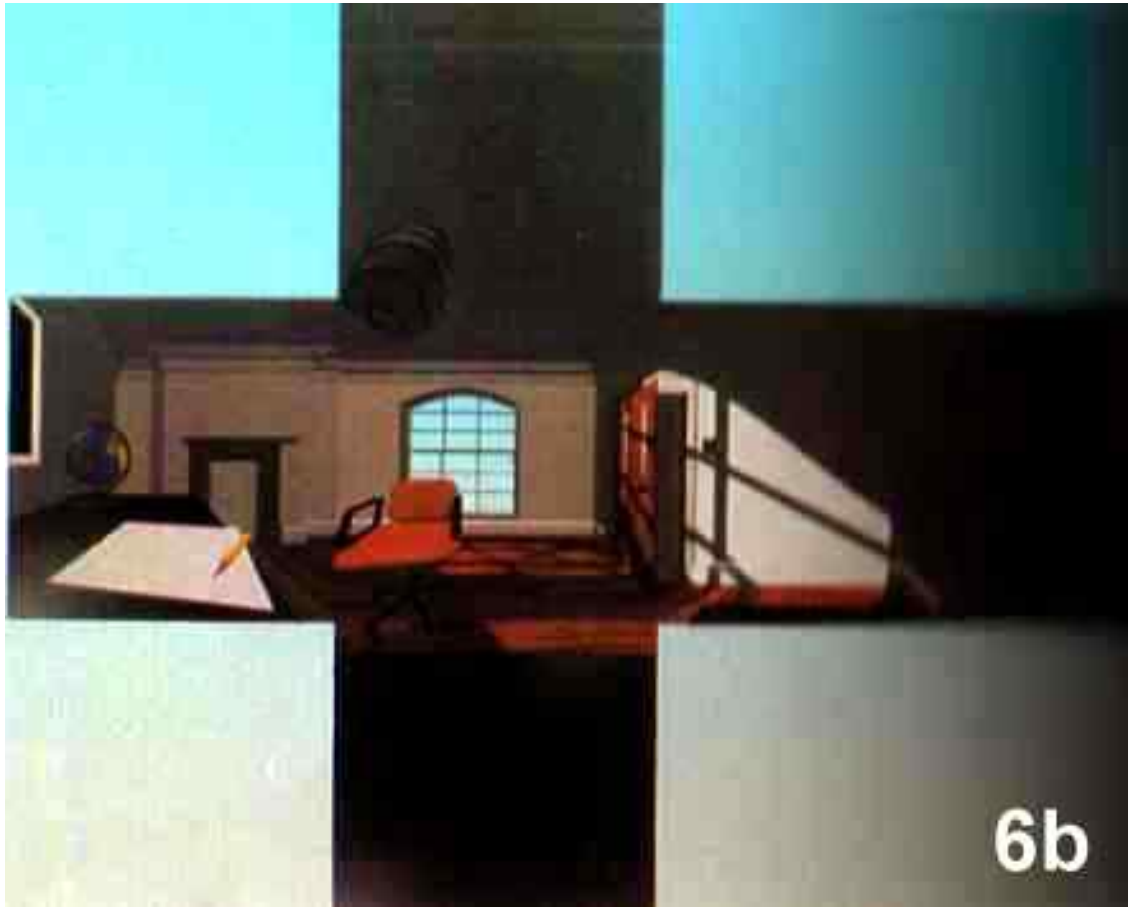
10.3 ENVIRONMENT-MAPPING

Sphere Mapping

- Probleme
  - Ebenfalls schwierige Erstellung
    - Rendering
    - Foto einer spiegelnden Kugel
    - Umrechnung aus ebenen Texturen (Würfel)
  - Abtastrate über Textur nicht regelmäßig
  - Interpolation der Texturkoordinaten führt zu Fehlern, vor allem am Rande der Kugel

10.3 ENVIRONMENT-MAPPING

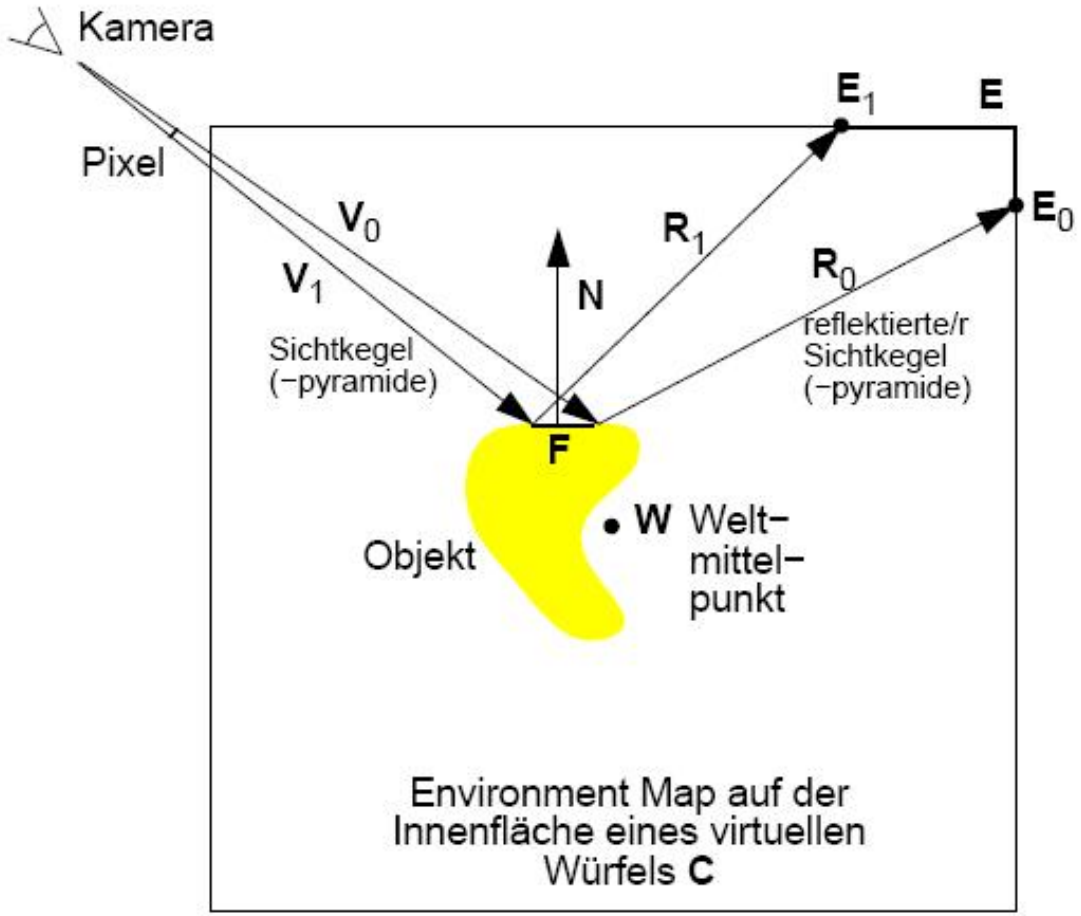
Verwendung eines Würfels



10.3 ENVIRONMENT-MAPPING

Verwendung eines Würfels

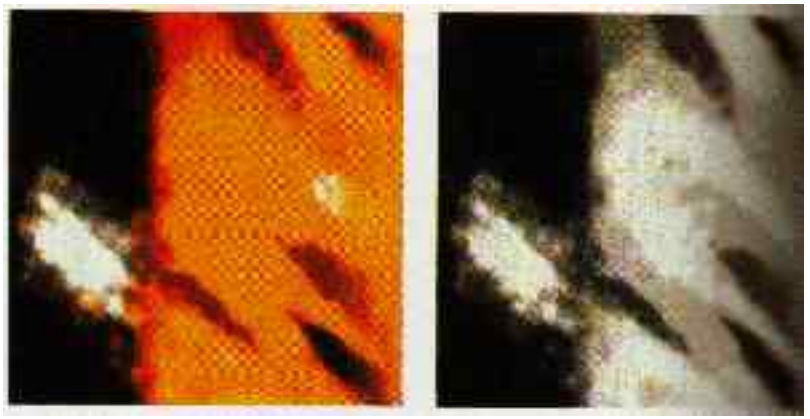
- Einfachere Erstellung der Textur
- Anti-Aliasing durch Sichtpyramide
- Parametrisierung
  - besser als bei einer Kugel
  - immer noch schlecht



10.4 WEITERE VERFAHREN

Chrome-Mapping

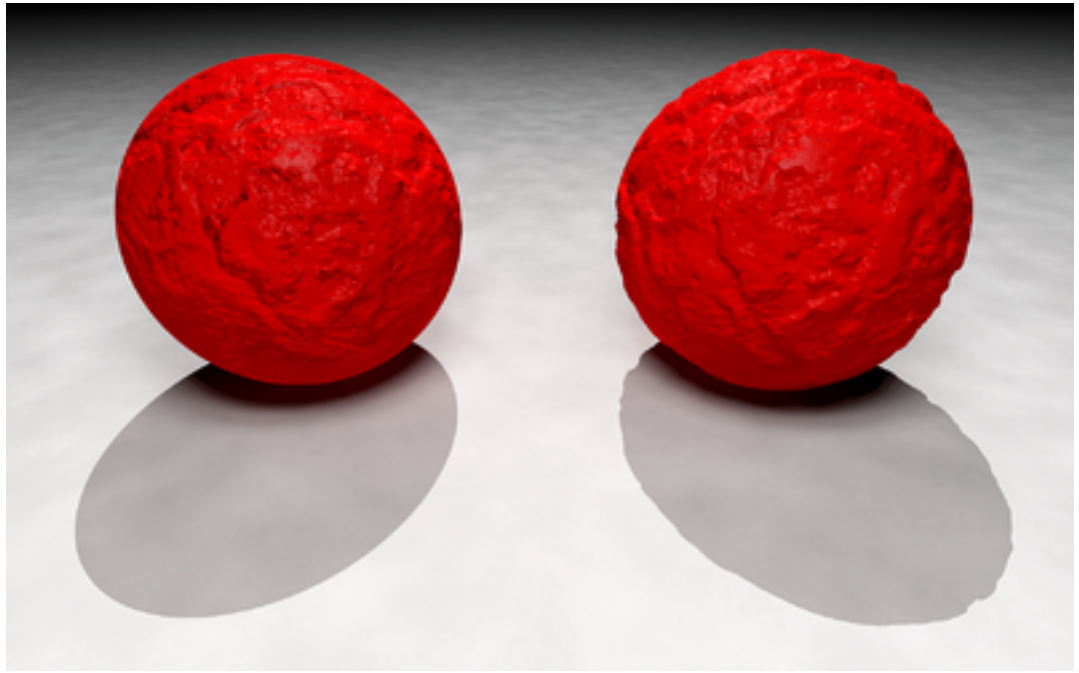
- Abbildung eines willkürlichen Musters (chrome map) auf eine reflektierende Oberfläche
- Künstliche Unschärfe
- Textur fest im Raum (daher geeignet für Animationen)



10.4 WEITERE VERFAHREN

Displacement Mapping

- Ziel
  - Plastischere Erscheinung der Oberfläche
- Unterschied zum Bump-Mapping:
  - Verschieben der Oberflächenpunkte gemäß Höhenfeld entlang der Normalen
  - Meist Kombination mit Textur
- Vorteil
  - Schatten & Silhouette erscheinen nicht mehr glatt



10.4 WEITERE VERFAHREN

Displacement Mapping

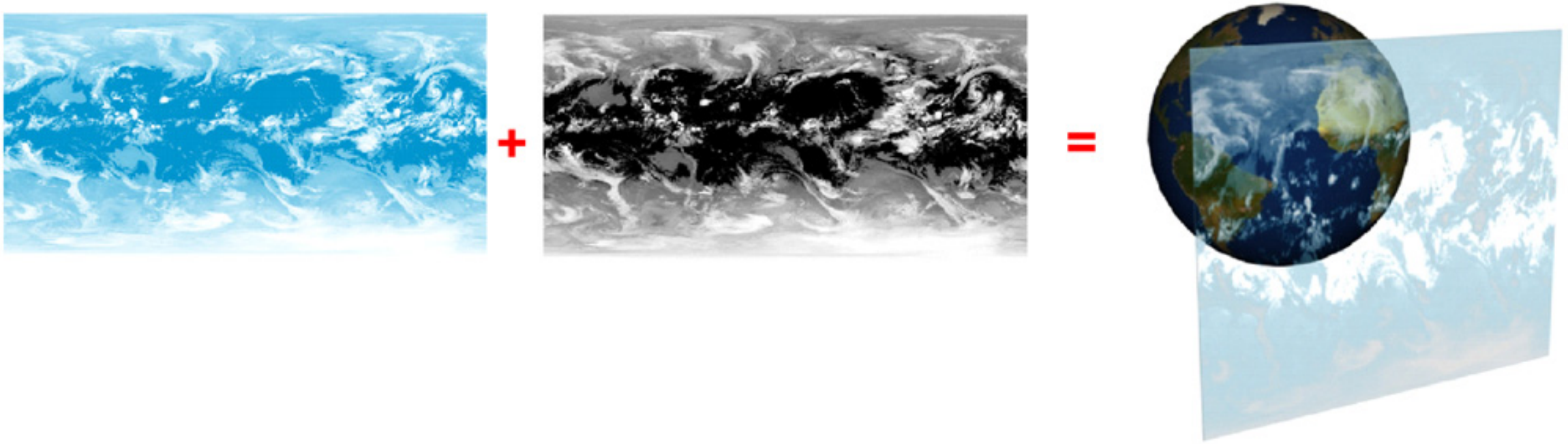
10.4 WEITERE VERFAHREN

Opacity-Mapping / Transparency-Mapping

- Ziel
  - Transparenz von Objekten lokal kontrollieren
- Opacity-Map
  - Grauwerte entsprechen Alpha-Wert
  - Von schwarz (opak) bis weiß (transparent)

10.4 WEITERE VERFAHREN

Opacity-Mapping / Transparency-Mapping



ZUSAMMENFASSUNG

- Erzielte Effekte
  - Veränderte Maserungen und Muster einer glatten Fläche
    - Texture-Mapping
    - Chrome-Mapping
  - Modellierung rauer Oberflächen
    - Bump-Mapping
    - Displacement-Mapping
  - Abbildung komplexer Umgebung auf Objekte
    - Environment-Mapping
- Kombination der Verfahren möglich
- Echtzeit durch Hardwareunterstützung kein Problem

QUELLEN

- Computergraphik, Universität Leipzig
  (Prof. D. Bartz)
- Graphische Datenverarbeitung I, Universität Tübingen
  (Prof. W. Straßer)
- Graphische Datenverarbeitung I, TU Darmstadt
  (Prof. M. Alexa)
- Computergraphik, Uni Siegen (Prof. Klomfaß)