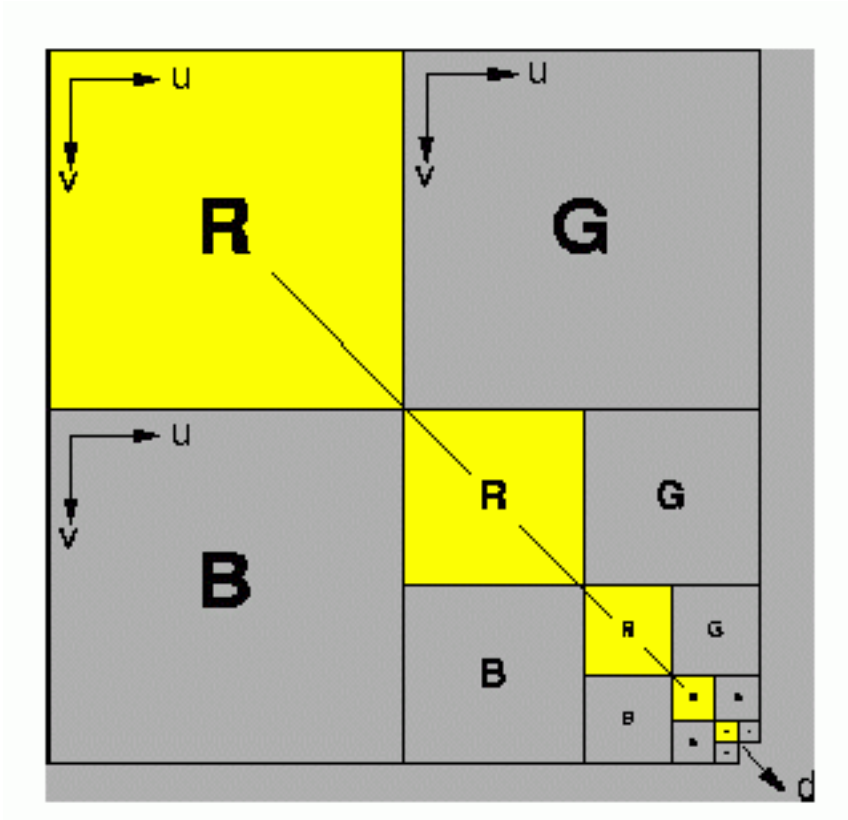
Techniken der Effizienzsteigerung bei 2D-Texturierung:

Mip-Mapping

MIP = "Multum in Parvo" = vieles auf kleinem Raum

- spezielle Texture-Mapping-Erweiterung, häufig bei Echtzeit-anwendungen, z.B. Spielen, verwendet
- LOD (level of detail) in Texturen
- Idee: berechne eine "Pyramide" gefilterter (vergröberter) Bilder im Voraus; Auswahl der Ebene, aus der ein Textur-pixel entnommen wird, abhängig von der Entfernung
- meist speichert eine Mip-Map C_{mip} eine quadratische Textur der Größe $n \times n$, wobei n eine Zweierpotenz ist, in fortwährend halbierten Auflösungsstufen



Auf der Stufe $d=0$ werden die Texturwerte direkt übernommen.

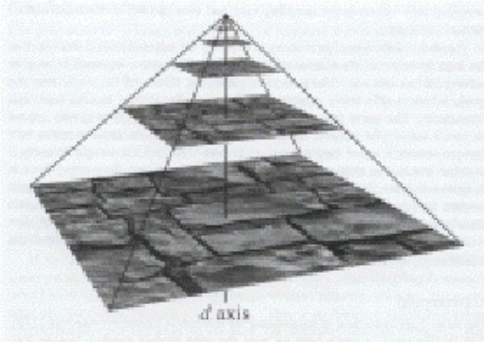
$$C^0_{mip}[i,j] = C[i,j],\ 0 \le i,j < 2^k.$$

Die übrigen Stufen d entstehen durch Filterung der jeweils vorhergehenden Stufe.

$$C^d_{mip}[i,j] = \frac{1}{4}\left(C^{d-1}_{mip}[2i,2j] + C^{d-1}_{mip}[2i+1,2j] + C^{d-1}_{mip}[2i,2j+1] + C^{d-1}_{mip}[2i+1,2j+1]\right)$$

$1 \le d < k$ -1 und $0 \le i, j < 2^{k-d}$.

Auf der Stufe d der Texturhierarchie werden also 2^{2d} Texel der Originaltextur als ein einziges Texel dargestellt.



Diese Vorfilterung erfolgt nur einmal, unabhängig vom Betrachterstandpunkt.
Für die Texturierung der Flächenelemente benötigt man die Kantenlängen des Footprints (= Urbild des Display-Pixels in der Texturebene). Den Gesamttexturwert erhält man durch bilineare Interpolation der MipMap-Werte an den Eckpunkten einer quadratischen Bounding-Region des Footprints.

ausgefeilteres Verfahren: genauere Überdeckung des (näherungsweise als Parallelogramm angenommenen) Footprints durch quadratische Texturfelder (entsprechend MipMap-Zugriffen) = "*Footprint Assembly*" (FPA)

(näheres bei Krömker 2001).

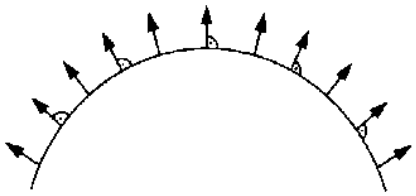
Bump Mapping

Mit den bisher beschriebenen Verfahren können nur Materialparameter variiert und Beleuchtungsergebnisse skaliert werden
⇒ beleuchtete Flächen sehen dadurch variantenreicher und realistischer aus, trotzdem wirken sie häufig noch zu glatt.

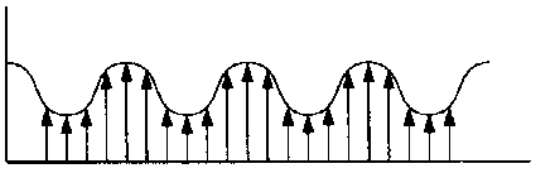
Gesucht: einfache Technik, mit der man Oberflächen rauh, runzlig, zerknittert, gefaltet, gekräuselt darstellen kann.

Exakte Modellierung von Unebenheiten: Aufaddieren eines 2D-Höhenfeldes zu einer ebenen Grundfläche (→ *Offsetfläche*)
⇒ oft zu rechenaufwändig
außerdem: in die Phong-Beleuchtungsgleichung gehen die exakten Positionen gar nicht ein; es kommt auf die *Normalenvektoren* an!

⇒ Idee des *Bump-Mapping* (Blinn 1978):
für kleine Unebenheiten reicht es aus, die Visualisierung mit der Originalgeometrie durchzuführen, bei der Beleuchtungsgleichung aber die Normalen der uneben gemachten Oberfläche zu verwenden.

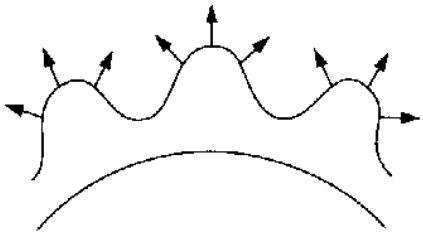


(a) Originalfläche **P**(u) mit Normalen **N**(u)



(b) Bump Map h(u)

(c) Offsetfläche **P'**(u)



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

Resultat: scheinbare Oberflächenverformung im kleinen Maßstab durch Verwendung der abgelenkten Normalen-vektoren

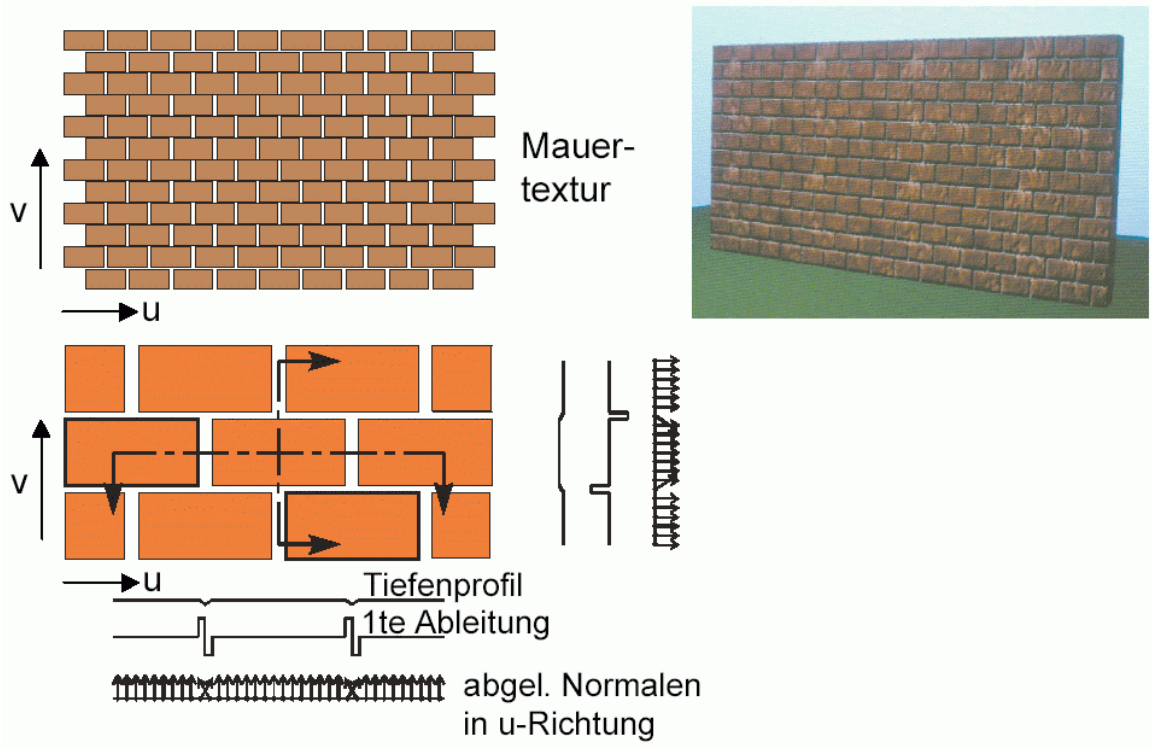
Bump Map = Feld mit Offset-Werten, die genutzt werden, um einen Punkt auf der Oberfläche nach oben oder unten zu versetzen (nur für die Bestimmung der Normalen)

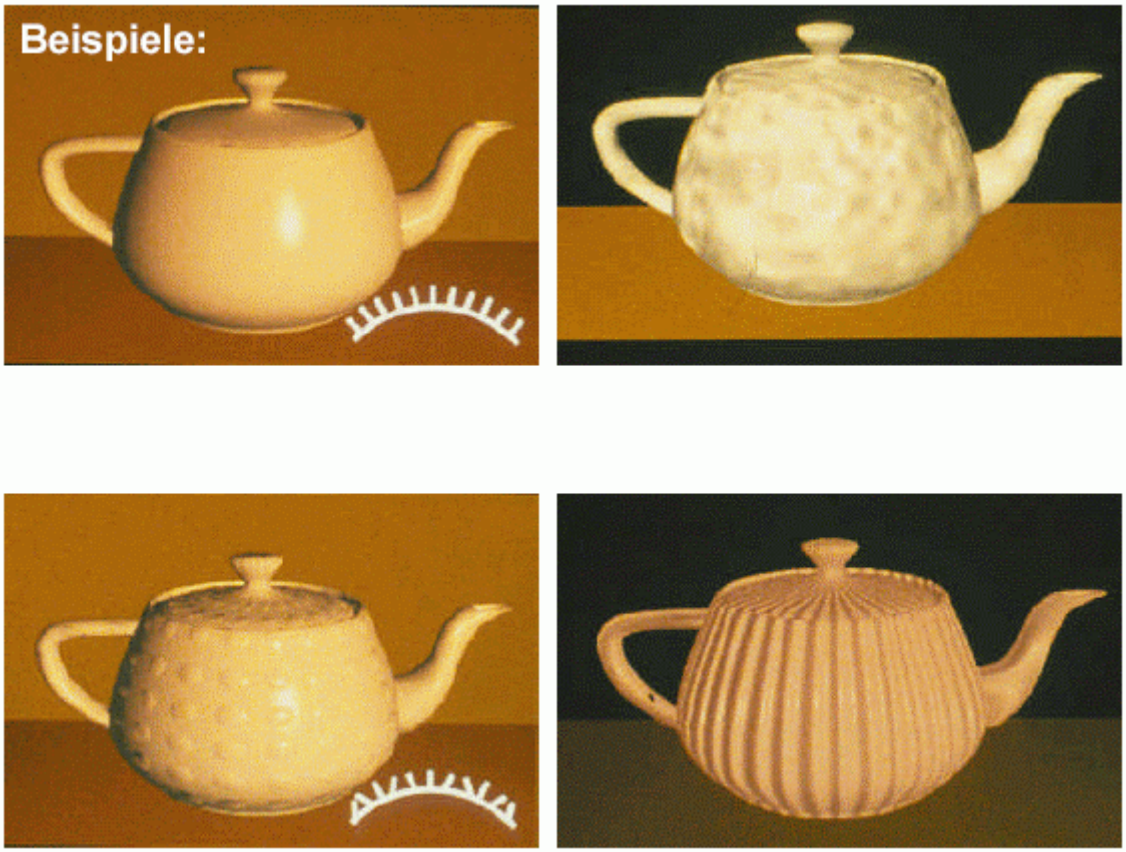
Den gesuchten Normalenvektor bestimmt man als Kreuz-produkt der Richtungsableitungen der Offsetfläche in einem gegebenen Punkt
(Formeln z.B. bei Krömker 2001).

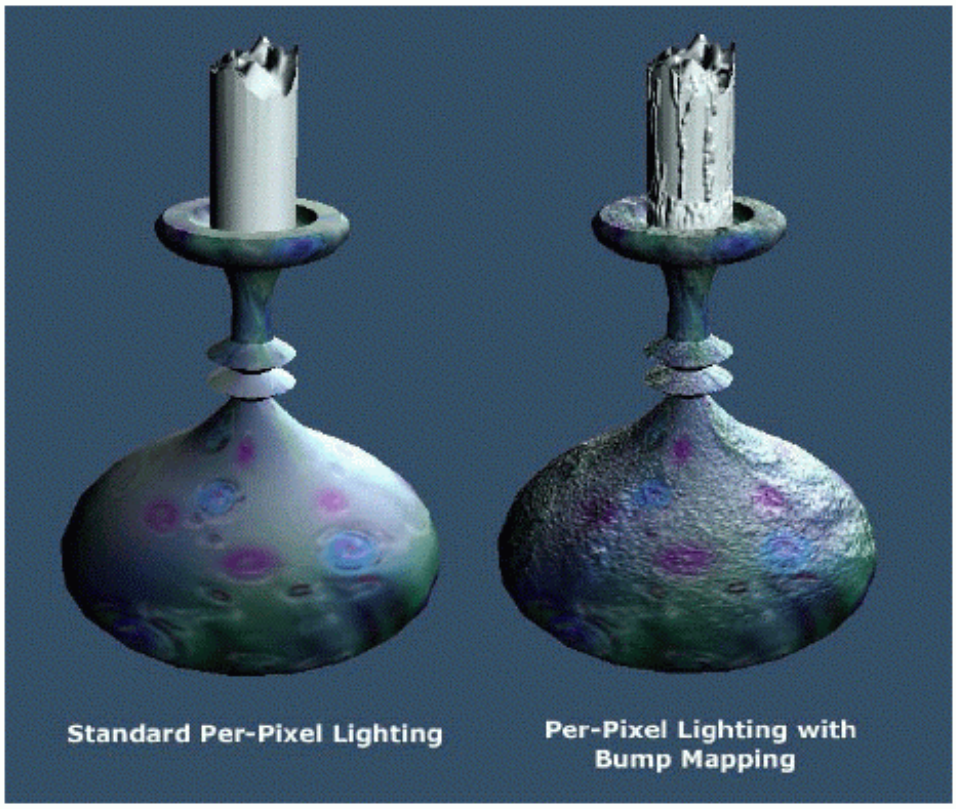
Beachte:

- Bump Mapping kann nur mit denjenigen Beleuchtungs-verfahren kombiniert werden, die eine explizite Beleuchtungsrechnung in jedem Flächenpunkt durchführen (Phong, Raytracing; **nicht** Gouraud).
- Seit 1997 gibt es Hardware-Implementationen des Bump Mapping.

Beispiele:







Bump Mapping

Bei größeren Unebenheiten: tatsächliche Berechnung der Offsetfläche und ihre Benutzung beim Rendering

"Displacement Mapping"

Überlagerung einer parametrischen Oberflächenfunktion

(x,y,z) = *f*(u,v) mit Hilfe eines Höhenfeldes (Displacementfunktion)

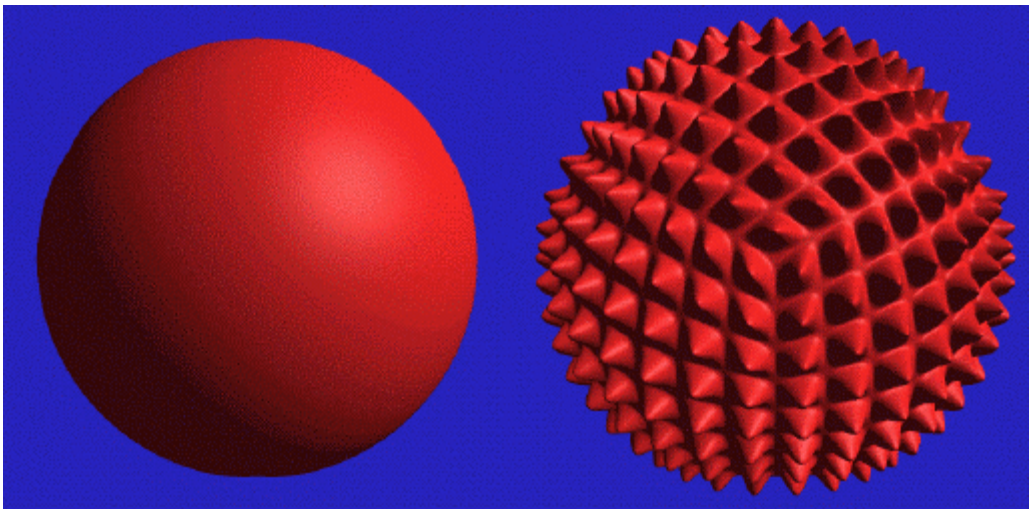
h = *g*(u,v).

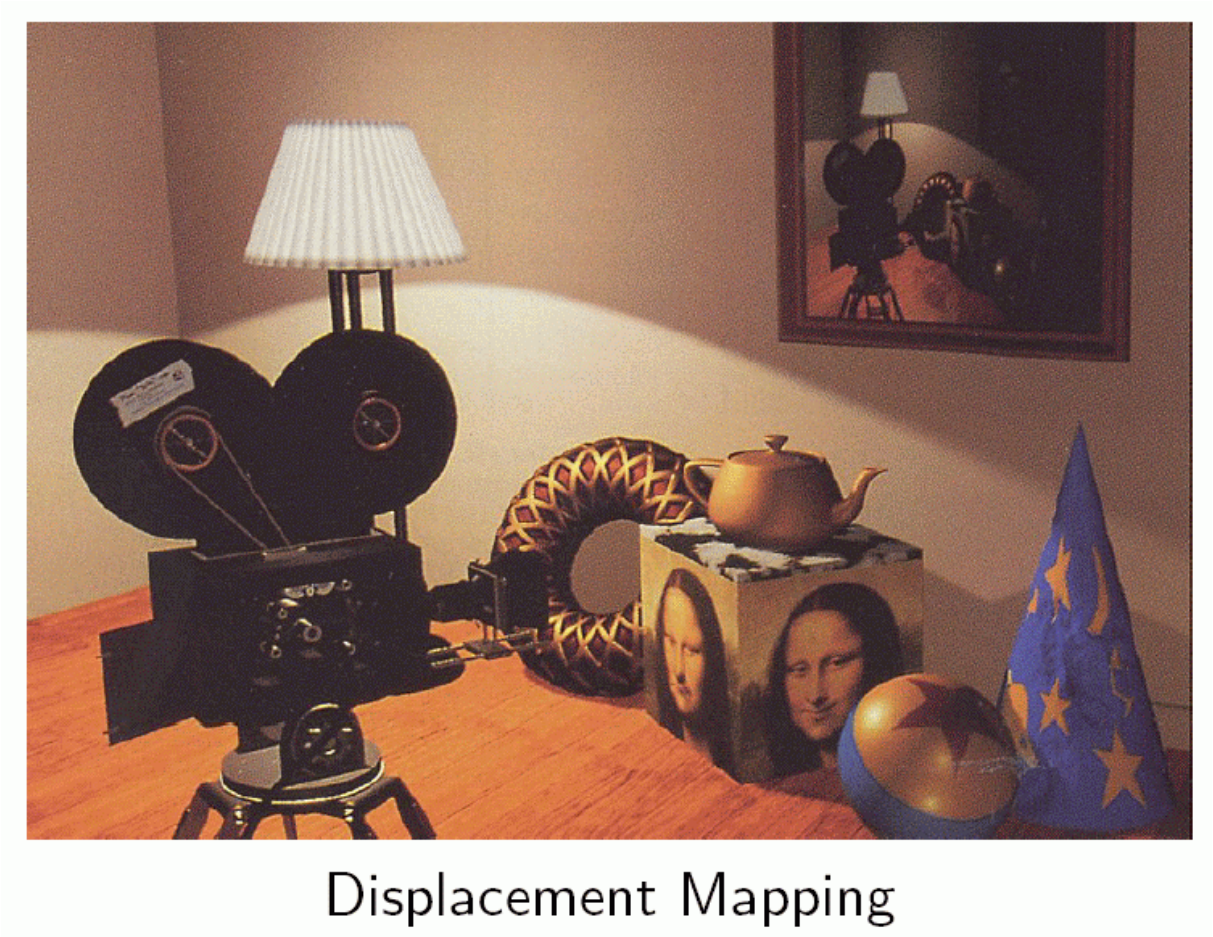
Wir erhalten:

$$f'(u,v) = f(u,v) + g(u,v) \cdot n(u,v)$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + h \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$$

Beispiel:





Displacement Mapping

alle bisher besprochenen Verfahren erzeugen Textur aus 2D-Zahlenfeldern: "diskrete Texturen"

Vorteile diskreter Texturen:

- einfache Erzeugung auch komplexer, realitätsnaher Muster (Foto, Zeichnung... dann einscannen)
- wenige, etablierte Verfahren; z.T. schon hardwaregestützt

Nachteile:

- Texturen mit hoher Auflösung haben hohen Speicherbedarf
- Beim Vergrößern von Bildern treten Artefakte auf
- Fortsetzung (Aneinandersetzen) von Texturen ist oft problematisch
- der der Textur zugrundegelegte Kontext (Sonnenstand, Schattenwurf...) stimmt häufig nicht mit der zu modellierenden Szene überein. Die Suche nach passenden Vorlagen kann aufwändig sein
- beim Mapping auf beliebige Flächen treten Verzerrungen und infolgedessen Abtast-Probleme (Aliasing) auf.

Alternative:

Prozedurale Texturen

(vgl. Kap. 10, prozedurale Modelle)

Bei jedem Aufruf eines Textur-Elements wird eine mathematische Formel bzw. ein Algorithmus ausgewertet.

Beispiel: Fractional Brownian Motion (Kap. 10).

Vorteile:

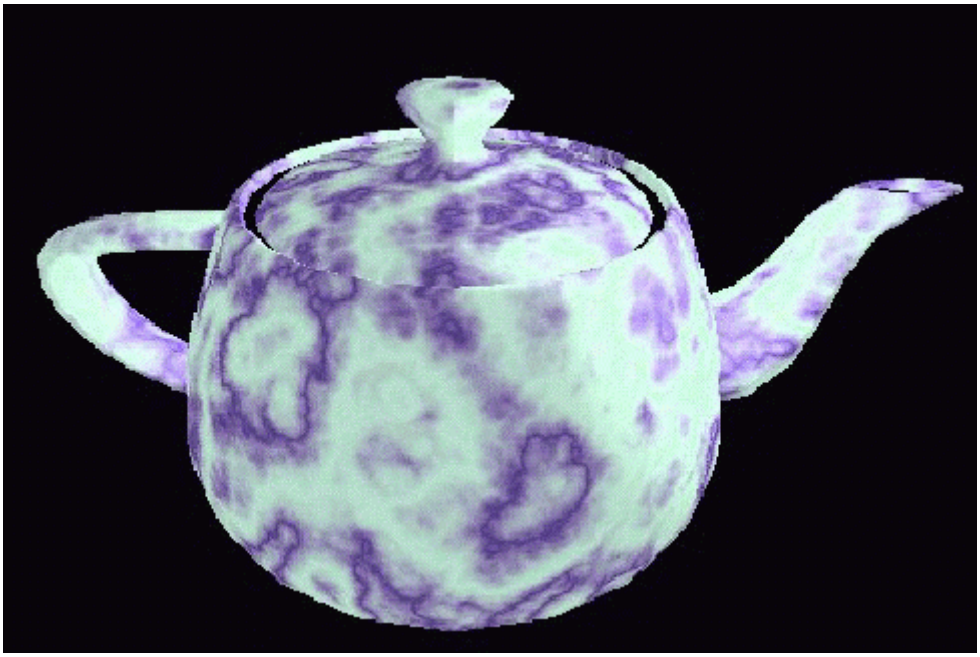
- minimaler Speicheraufwand
- die Texturwerte können an jeder Stelle mit optimaler Genauigkeit berechnet werden
- die Textur ist auf der gesamten Fläche definiert

Nachteile:

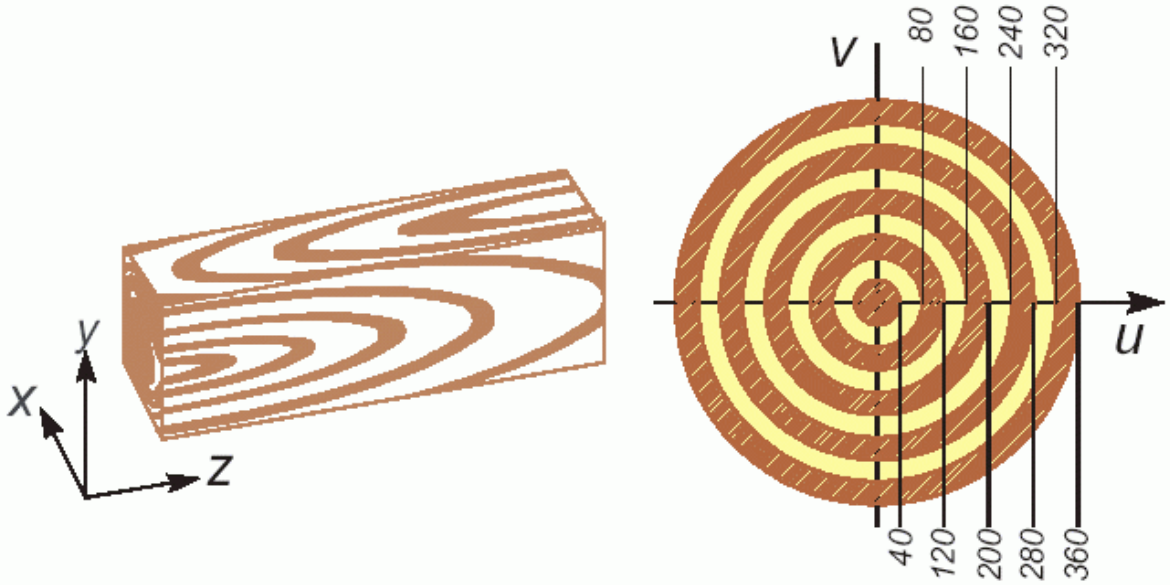
- große Vielfalt an unterschiedlichen Verfahren, Standardisierungen schwierig
- selbst Experten haben Probleme, komplexe Texturen, die sie bildlich vor Augen haben, in mathematische Formeln umzusetzen!

häufig: Kombination des prozeduralen Ansatzes mit *3D-Textur-Ansatz:*

- Texturfunktion wird nicht als $C(u, v)$ in 2 Variablen (flächig), sondern als $C(u, v, w)$ für Volumina definiert.
- Das zu texturierende Objekt wird von diesem Volumen umschlossen
- der Texturwert an einem Objektpunkt ergibt sich aus dem Funktionswert an der jeweiligen Stelle: Objektoberfläche schneidet Textur-Muster aus dem Volumen heraus
- Oft für Maserungen und Muster wie z.B. Holz, Marmor.



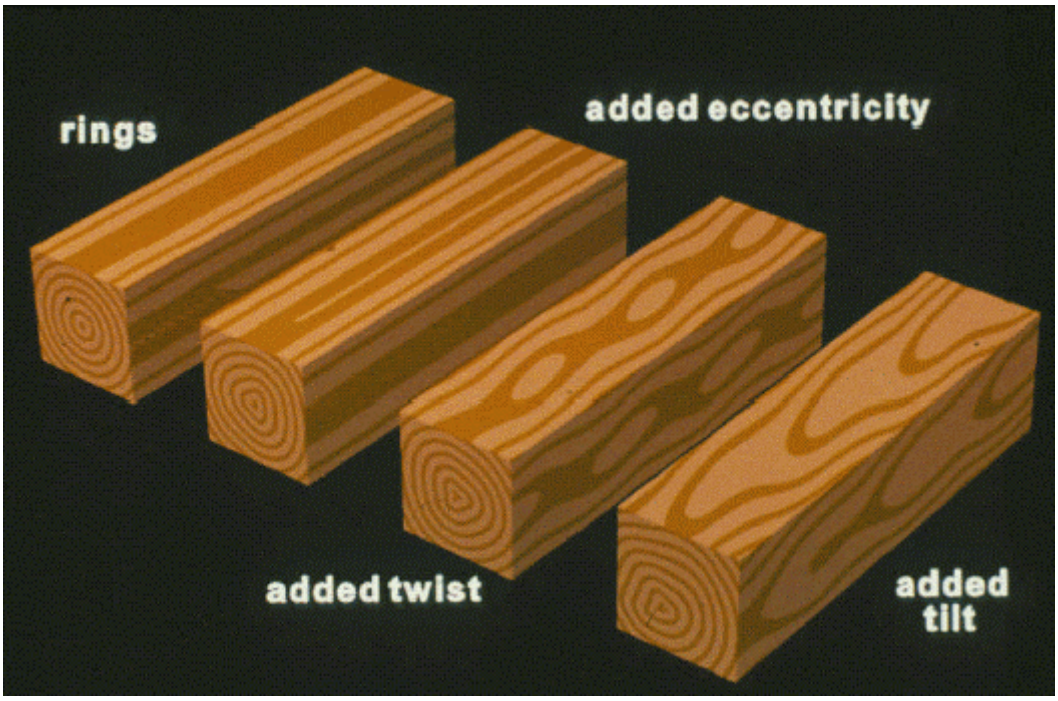
Beispiel: Erzeugung einer Holztextur mit Hilfe einer Prozedur



Beispiel: Erzeugung einer Holztextur mit Hilfe einer Prozedur (Pseudokode)

```
procedure wood(var x,y: real; var r,b,g: real);
var radius, angle: real; grain: integer;
begin
   radius := sqrt(x*x+y*y)
   grain:= round(radius) mod 80
   if grain < 40 then
        begin r:=r_light; g:=g_light; b:=b_light
        end else
        begin r:=r_dark; g:=g_ dark; b:=b_dark
        end
    end

(r,g,b) := wood(x,y)
```



Bis jetzt wurde angenommen, dass der Textur-Wert an einer Stelle lediglich die *Farbintensitäten* an diesem Punkt beeinflusst (Ausnahme: Bump Mapping, Displacement-Mapping).

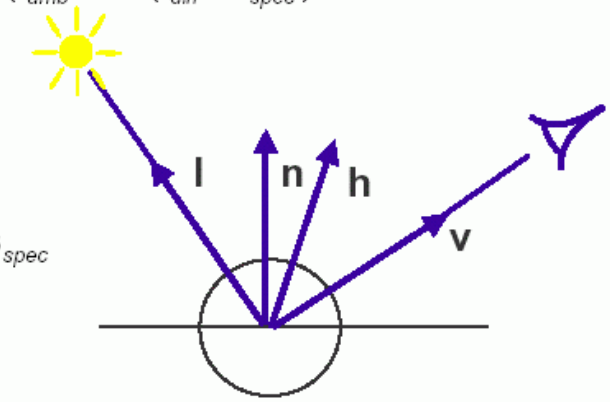
Es gibt wesentlich mehr Möglichkeiten, die Beleuchtungs-rechnung durch Texturen zu beeinflussen, wodurch sich weitere Spezialeffekte erzeugen lassen.
Basis für die folg. Betrachtungen (nach Krömker 2001) ist ein erweitertes Blinn-Phong-Modell mit mehreren Lichtquellen (⊗ bezeichnet komponentenweises Produkt bei Vektoren):

$$\mathbf{i}_{tot} = \mathbf{a}_{glob} \otimes \mathbf{m}_{amb} + \mathbf{m}_{emi} + \sum_k c^k_{spot}(\mathbf{i}^k_{amb} + d^k(\mathbf{i}^k_{diff} + \mathbf{i}^k_{spec})$$

$$\mathbf{i}_{amb} = \mathbf{m}_{amb} \otimes \mathbf{s}_{amb}$$

$$\mathbf{i}_{diff} = \max((\mathbf{n} \cdot \mathbf{l}),0)\mathbf{m}_{diff} \otimes \mathbf{s}_{diff}$$

$$\mathbf{i}_{spec} = \max((\mathbf{h} \cdot \mathbf{n})^{m_{shi}},0) \cdot \mathbf{m}_{spec} \otimes \mathbf{s}_{spec}$$



Möglichkeiten der Beeinflussung der Beleuchtungsrechnung

1. Ersetzen der Objektfarbe durch Textur (*replace*)
Bei dieser einfachsten Art der Texturierung wird die Objektfarbe durch die Farbe der Textur ersetzt. Jegliche Beleuchtung des Objekts wird entfernt, außer die Textur selbst enthält Beleuchtungseffekte.

$$\mathbf{c}_{out} = \mathbf{c}_{tex}$$

2. **A posteriori Skalierung des Farbwertes (*modulate*)**
Eine der am häufigsten angewandten Techniken führt die Beleuchtungsrechnung und das Shading mit konstanten Beleuchtungsparametern durch und skaliert den Beleuchtungswert $\mathbf{i}_{tot}$ erst im nachhinein komponentenweise mit dem Texturwert $\mathbf{c}_{tex}$

$$\mathbf{c}_{out} = \mathbf{i}_{tot} \otimes \mathbf{c}_{tex}$$

3. Modulation der Flächenfarbe

Die Flächenfarbe wird im wesentlichen durch den Materialparameter m_{diff}, im geringeren Maße aber auch durch den Parameter m_{amb} bestimmt. Also

$$\mathbf{m}_{diff} = \mathbf{m}_{diff}' \otimes \mathbf{c}_{tex}$$
$$\mathbf{m}_{amb} = \mathbf{m}_{amb}' \otimes \mathbf{c}_{tex}$$

zu setzen.

Der wesentliche Unterschied zu 2. besteht darin, dass die spekularen Reflektionen und Emissionen von der Textur unbeeinflußt bleiben.

4. Modulation der spekularen Reflektion

$$\mathbf{m}_{spec} = \mathbf{m}_{spec} \otimes \mathbf{c}_{tex}$$

Die Highlights können dadurch unregelmäßig gestaltet werden.

5. Modulation der Transparenz
Durch die Modulation des Transparenzparameters

$$\alpha = \alpha_{obj} * \alpha_{tex}$$

können sehr realistisch aussehende Effekte erzielt werden:
α=*1* ➔ Objekte hinter diesem Flächenpixel sind vollständig sichtbar
α=*0* ➔ so wird kein Licht durchgelassen
ansonsten wird mit dem Wert von α gefiltert.

Bei Beschränkung auf die Werte 0 und 1 kann durch die Textur zwischen Sichtbarkeit und Unsichtbarkeit des Objekts "umgeschaltet" werden. Dadurch können auch kompliziert geformte Flächen aus einfachen Flächen "ausgeschnitten" werden.

Mit stochastischen Werten für α können z.B. verschmutzte und milchige Glasscheiben modelliert werden.

6. Perturbation des Normalenvektors
Beim **Bump Mapping** wird mit Hilfe einer skalarwertigen Textur eine Offsetfläche *P'(s,t)* definiert. Die Normalenvektoren der Offsetfläche werden dann als Variationen der Normalenvektoren der Basisfläche interpretiert

7. Modulation von Lichtquellenparametern
Eine weitere Möglichkeit besteht darin, Lichtquellenparameter durch Texturen zu beeinflussen. Besonders anschaulich ist dies bei Projektorlichtquellen. Dabei wird eine zweidimensionale Textur in den Raum projiziert, d.h. die Lichtemission wird in Abhängigkeit von der Lichtrichtung moduliert:

$$L_i \sim C_{tex}(F_{invmap}(\vec{L}_i)).$$

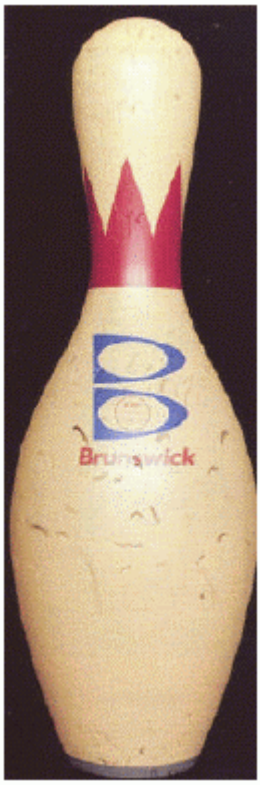
8. Höhenfelder und Offsetflächen
Hierbei wird mit Hilfe von Texturen die tatsächliche Geometrie von Oberflächen verändert. Darauf beruht das sogenannte **Displacement-Mapping**.

9. U.v.a.m. ....

Beachte:
bei Gouraud-interpolierten Dreiecken kann nur die Methode 2, "a posteriori-Skalierung der Farbwerte", angewendet werden.

Die alternative oder kombinierte Anwendung der Texturierungsverfahren 3 bis 8 besteht nur bei Visualisierungsverfahren, die eine explizite Beleuchtungsrechnung an jedem Flächenpunkt durchführen oder zumindest die verschiedenen Anteile der Phong-Beleuchtungsgleichung getrennt durch Gouraud-Interpolation ermitteln.

Beispiele für komplexe optische Effekte durch Kombination mehrerer Texturen und Texturierungstechniken:



- **Modulation des Bildsyntheseprozess**
  - Farbe
  - Fläche
  - Normale
  - Beleuchtung

ILM