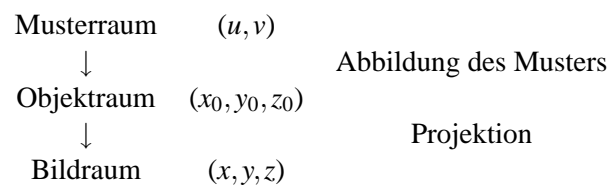


Kapitel 19

Texturing

Unter dem Begriff *Texturing* werden neben dem *Texture Mapping* auch alle anderen Verfahren zusammengefasst, bei denen das Aussehen einer Fläche an jedem Punkt mit Hilfe einer BitMap, einer Funktion oder sonstigen Daten verändert wird.

Zur realistischen Gestaltung von Oberflächen verwendet man ein zweidimensionales Musterfeld (*texture map*), bestehend aus *Texeln*, aus dem für jedes Pixel die zugehörige Farbe ermittelt werden kann. Zugrunde gelegt sind zwei Abbildungen



Die Verknüpfung der zugehörigen inversen Transformationen liefert die zum Einfärben eines Pixels benötigte Information.

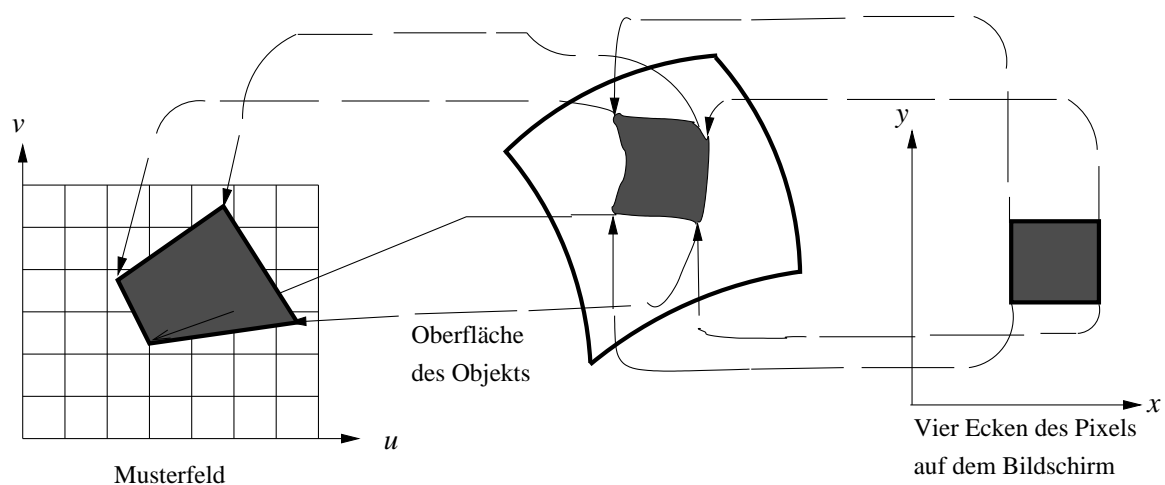


Abbildung 19.1: Transformationskette beim Texture Mapping

Wie Abbildung 19.1 zeigt, müsste eigentlich zum Einfärben eines Pixels im Bildraum die gewichtete Durchschnittsintensität aller überdeckten Pixel im Musterraum benutzt werden. Aus Effizienzgründen begnügt man sich jedoch mit dem Zugriff auf das Texel in der Mitte des zuständigen Texturbereichs.

Damit stellt sich der zugrunde liegende Prozess wie folgt dar:

- Raumkoordinaten des Flächenpunktes berechnen,
- Projektion in den Parameterraum durchführen,
- Texturkoordinaten berechnen
- Texturwerte ggf. anpassen
- Erscheinungsbild mit dem Texturwert modifizieren.

Nachdem zunächst das Texturing anhand der häufigsten Anwendung - dem *Texture Mapping* - beschrieben wird, werden eine Reihe weiterer Texturing-Techniken vorgestellt.

19.1 Texture Mapping

Beispiel: Eine Fläche der Szene soll den Eindruck erwecken als handle es sich um eine Mauer aus Steinen und Fugen. Dazu wird ein Foto von einer solchen Mauer auf die Fläche "geklebt". Immer, wenn die Fläche gerastert wird, werden die oben genannten Schritte für jedes Bildschirmpixel der Fläche durchgeführt.

- Die Koordinaten im DC werden für die Eckpunkte durch die Projektion WC_DC und für alle inneren Punkte durch Interpolation zwischen den Eckpunkten bestimmt. Es ergibt sich ein Tripel (x, y, z) .
- Eine zweidimensionale Textur, wie sie fast immer vorliegt, hat zwei Parameter u und $v \in [0; 1]$, die die Textur parametrisieren. Im zweiten Schritt wird die Projektion der DC-Koordinaten in den (u, v) -Raum berechnet. Im Mauer-Beispiel würde man eine perspektivische Projektion durchführen, um den korrekten perspektivischen Eindruck zu erhalten. Wenn z.B. eine Weltkarte um eine Kugel "gewickelt" werden soll, um den Eindruck eines Globusses zu erwecken, müsste eine sphärische Projektion gewählt werden. Andere mögliche Projektionsarten sind zylindrisch, elliptisch oder kubisch. Alle Projektionen sind Abbildungen; daher das Wort "Mapping" im Titel der diversen Techniken.
- Im darauffolgenden Schritt wird aus den Werten für u und v das Texture Element oder *Texel* für das aktuelle Pixel bestimmt. Diese Abbildung führt also aus dem Parameterraum in den Texturraum. Wenn die Mauer-Textur beispielsweise 256×256 Pixel groß ist, würden die Werte für u und v mit 256 multipliziert und der Nachkommaanteil abgeschnitten, um die ganzzahligen Koordinaten des Texels zu erhalten. Bei der Wahl dieser *Korrespondenzfunktion* (*corresponder function*, siehe Abbildung 19.2) stehen folgende Typen zur Verfügung:

- *wrap, repeat, tile*: Das Bild wiederholt sich auf der Fläche. Dies wird erreicht, indem der Vorkommaanteil der Parameterwerte vernachlässigt wird. Diese Art des Texturing wird gewählt, wenn ein Material die ganze Fläche durch Wiederholung überziehen soll. Im Mauer-Beispiel wäre ein Bild ausreichend, das wenige Steine und Fugen enthält. Allerdings muß darauf geachtet werden, daß die Ränder des Bildes oben und unten bzw. rechts und links identisch sind; sonst sind die Nahtstellen zwischen den Texturen erkennbar.
 - *mirror*: Wie wrap etc. allerdings wird die Textur abwechselnd im Original und in u - bzw. v -Richtung gespiegelt dargestellt. Dadurch läßt sich die Wiederholung schwerer feststellen.
 - *clamp*: Die Ränder der Textur fungieren als Klammer und werden für Werte > 1 wiederholt. Manche APIs erlauben die Einstellung clamp für einen Parameter und wrap für den anderen.
 - *border*: Werte > 1 werden mit einer extra festgelegten Farbe gerendert. Diese Einstellung läßt sich gut verwenden, wenn ein Aufkleber irgendwo auf einer Fläche plaziert werden soll.
- Die Werte, die bei diesen Koordinaten in der Textur hinterlegt sind, können auf verschiedene Weise genutzt werden, um das Erscheinungsbild der Fläche zu ändern. Im Mauer-Beispiel wären RGB-Werte in der Textur enthalten. Falls die Werte in der Textur für diese Fläche angepasst werden müssen, kann dies im nächsten Schritt geschehen. Wenn die RGB-Werte beispielsweise zu dunkel sind, könnten sie mit einem Faktor > 1 multipliziert werden, um so eine Helligkeitssteigerung zu erreichen.
 - Abschließend wird der Texturwert interpretiert. RGB-Werte aus Bildtexturen ersetzen z.B. die diffuse Objektfarbe in der Beleuchtungsgleichung. Neben der harten Ersetzung der Ursprungswerte, können diese auch moduliert werden. Zu welchen Ergebnissen das führt, wird in den folgenden Abschnitten beschrieben.

Wenn in jedem neuen Frame, das gerendert werden soll, ein neues Bild aus einem Film als Textur benutzt wird, erscheint es so, als ob der laufende Film auf die Fläche projiziert wird, wie auf eine Kinoleinwand. Wenn die Texturkoordinaten in jedem Frame neu festgelegt werden, "wandert" die Textur von Frame zu Frame über die Fläche.

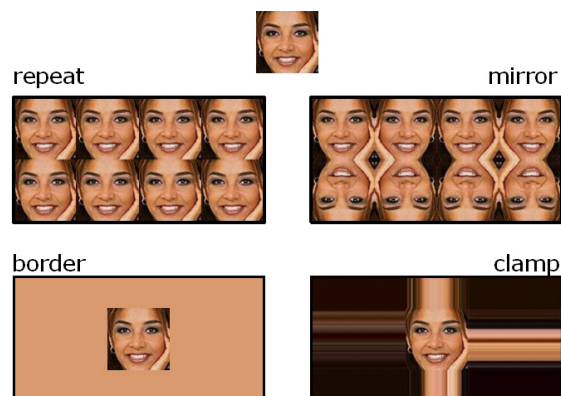


Abbildung 19.2: Auswirkungen der Korrespondenzfunktion

19.2 Mip Mapping

Wenn ein Polygon sich nahe beim Betrachter befindet bzw. weit vom Betrachter entfernt ist, stellt ein Pixel nur einen Teil eines Texels dar bzw. überdeckt ein Pixel mehr als ein Texel. Um den korrekten perspektivischen Eindruck zu erzeugen, muß diese Tatsache durch Antialiasing berücksichtigt werden. Eine Möglichkeit, dies zu tun ist das Mip Mapping (Multum in parvo: Großes im Kleinen). Damit nicht zur Laufzeit für jedes Pixel die Summe über alle überdeckten Texel gebildet werden muß, werden im Speicher schlechter aufgelöste Versionen der Textur vorgehalten. Jede Version hat ein Viertel der Texelzahl der nächstbesser aufgelösten Version (deshalb sind Texturen mit Zweierpotenzen als Kantenlänge ideal). Das Downsampling bei der Berechnung der Texturen kann mit einem Box-Filter - besser mit einem Gauss-Filter - geschehen.

Diese Texturen sind pyramidenförmig angeordnet; die beste Auflösung als Pyramidenboden; die schlechteste Auflösung (1 mal 1 Texel) als Pyramidenspitze. Die Koordinate vom Boden zur Spitze wird als Level of Detail (LOD) bezeichnet. Je nachdem, wie weit das Pixel vom Betrachter entfernt ist, wird es in die entsprechende Version der Textur projiziert, in der es ungefähr ein (oder zwei) Texel überdeckt.

19.3 Light, Gloss und Shadow Mapping

In einer statischen Szene kann die Auswirkung der Beleuchtung durch statische Lichtquellen (z.B. eine Wandlampe, die die Wand beleuchtet) vorab berechnet werden. Für jede Fläche wird eine Lightmap berechnet, die Werte zwischen 0 und 1 enthält. Mit ihr wird später die Beleuchtungsgleichung multipliziert:

$$C_{gesamt,diffus}[x,y] = C_{lighting,diffus}[x,y] * LightMap[u(x,y),v(x,y)]$$

Damit die Szene dabei nicht zuviel an Helligkeit verliert, kann anschließend mit einem Faktor zwischen 1 und 4 multipliziert werden, wobei maximal reines Weiß als Objektfarbe entstehen darf. Der Vorteil durch die Ersparnis an Rechenzeit ist erheblich.

Es besteht auch die Möglichkeit, eine LightMap in Scheinwerferform (weißer Kreis auf schwarzem Grund) zur Laufzeit über eine Fläche wandern zu lassen. Damit wird der Eindruck erzeugt, dass ein Scheinwerfer die Fläche überstreicht.

Gloss Mapping ist Light Mapping für die spekulare Objektfarbe. Z.B. könnten die Ziegel der Mauer glasiert sein. Dann würde die Fläche nur im Bereich der Ziegel spekulare Highlights zeigen.

Auch bei der Berechnung von Schatten kann eine Effizienzsteigerung erreicht werden, wenn die Szene statisch ist. Wie im vorangegangenen Kapitel beschrieben, wird die Szene aus der Sicht jeder Lichtquelle gerendert und die Schatten, die durch andere Flächen auf eine Fläche fallen, werden als Shadow Map bei dieser Fläche gespeichert.

19.4 Alpha Mapping

Wenn eine Fläche Löcher haben soll (z.B. ein Baum, bei dem zwischen den Blättern der Hintergrund zu sehen ist), kann eine Alpha Map darauf plaziert werden. Auch hier sind Werte zwischen 0 (die

Fläche ist bei diesem Texel völlig durchsichtig) und 1 (die Fläche ist völlig opak) abgelegt. Wieder wird die Beleuchtungsgleichung modifiziert:

$$C_{gesamt,alpha}[x,y] = C_{Baum,lighting}[x,y] * AlphaMap[u(x,y),v(x,y)] \\ + C_{Hintergrund,lighting} * (1 - AlphaMap[u(x,y),v(x,y)])$$

19.5 Environment oder Reflection Mapping

Im übernächsten Kapitel wird beschrieben, wie beim Raytracing berechnet wird, welche Objekte sich in anderen spiegeln. Derselbe Effekt kann - etwas weniger akkurat - mit Hilfe von Environment Mapping erzeugt werden. Es wird angenommen, daß die Objekte, die sich in der Fläche spiegeln sollen, weit entfernt sind und daß sich das Objekt, zu dem die Fläche gehört, nicht in sich selber spiegelt. Es wird ein Strahl vom Betrachter zum Pixel der Fläche geschossen und dort reflektiert. Der reflektierte Strahl wird als Verweis in die Environment Map interpretiert und das dort gefundene Texel wird als Modulation der spekularen Objektfarbe benutzt. Die Interpretation des reflektierten Strahls kann z.B. sphärisch, kubisch oder parabolisch geschehen.

19.6 Bump Mapping

Beim Bump Mapping wird durch Veränderung der Flächennormalen anhand einer Bum Map der Eindruck einer rauhen Oberfläche erzeugt. D.h. es wird die Beleuchtungsgleichung modifiziert durch Manipulation der Normalen im Punkt (x,y) der Fläche. Diese Vorgehensweise hat keine physikalische Entsprechung.

Es gibt zwei Möglichkeiten, das Maß der Abweichung in einer Bump Map zu speichern:

- Die Bump Map besteht aus zwei Werten b_u und b_v in jedem Punkt (u,v) , die zwei Vektoren skalieren, die senkrecht zueinander und zur Normalen sind und die zur Normalen addiert werden.
- Die Bump Map besteht aus einem Wert pro Texel und die Werte b_u bzw. b_v werden durch Differenzenbildung mit Werten in der Nachbarspalte bzw. Nachbarzeile errechnet.

Per-Pixel Bump Mapping wirkt sehr echt und ist nicht aufwändig zu berechnen. Allerdings werfen die Bumps keine Schatten und wenn der Betrachter flach über eine Fläche hinwegschaut, sind die durch das Bump Mapping suggerierten Höhendifferenzen nicht zu sehen.

19.7 Multitexturing

Jede der Texturing-Techniken für sich betrachtet, erhöht den Realismus der Darstellung. Noch mehr Realismus läßt sich durch die Kombination mehrerer Techniken erzielen. Beim *Multitexturing* werden dazu mehrere Texturen pro Fläche verknüpft. Die Reihenfolge der Verknüpfung und die Art der mathematischen Operationen spielt dabei eine wesentliche Rolle.

19.8 Displacement Mapping

Das Displacement Mapping fällt etwas aus dem Rahmen der anderen Techniken heraus, denn es modifiziert nicht das Erscheinungsbild der Fläche, sondern ihre Geometrie. Dies geschieht, um die Größe der Datei mit der Szenenbeschreibung klein zu halten. Als Basis dient ein Körper mit geringer Flächenanzahl. Jede Fläche des Körpers wird anhand der Displacement Map in mehrere kleine Flächen unterteilt. Dabei können die neuen Punkte nur entlang der Flächennormalen verschoben werden. Sie werden also entweder aus der Fläche herausgehoben oder in die Fläche hineingeschoben. Die Stärke der Verschiebung ist in der Displacement Map als Grauwert hinterlegt. Diese Technik eignet sich besonders gut, um Landschaften kompakt zu beschreiben (analog zu "Elevation Grid" von X3D) oder um verschiedene Geometrien auf Basis eines Grundkörpers zu definieren.

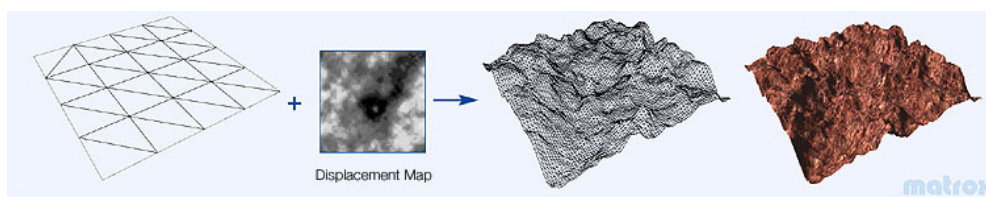


Abbildung 19.3: Displacement Mapping bei der Terraingenerierung (Courtesy by Matrox)

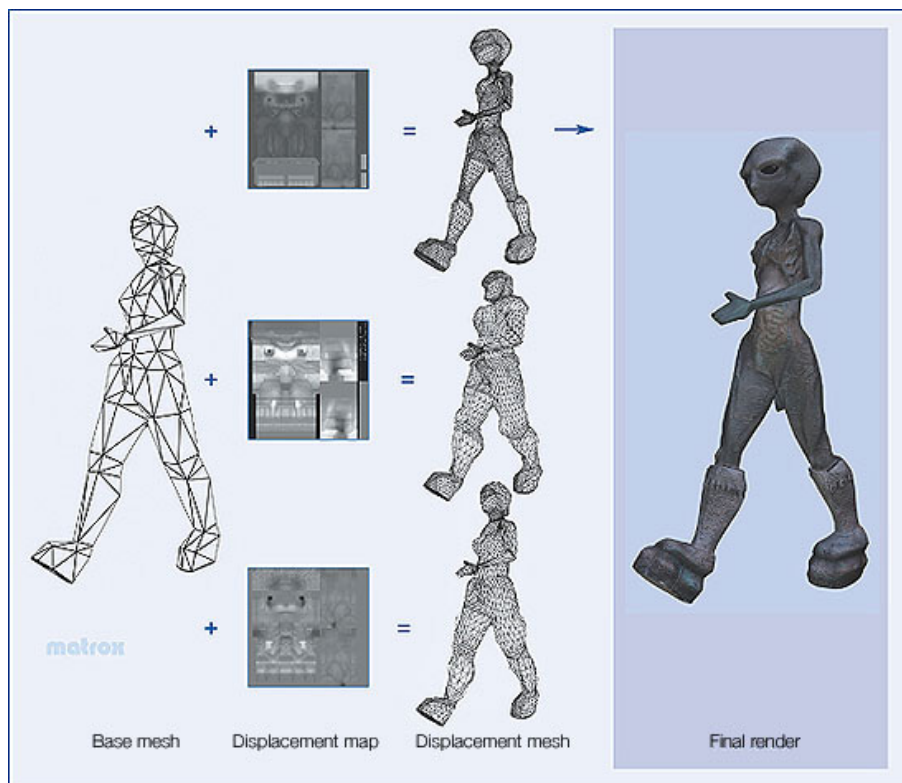


Abbildung 19.4: Generierung von drei Figuren aus einer Geometrie mit Hilfe von Displacement Mapping (Courtesy by Matrox)

19.9 Java-Applet zum Texture-Mapping

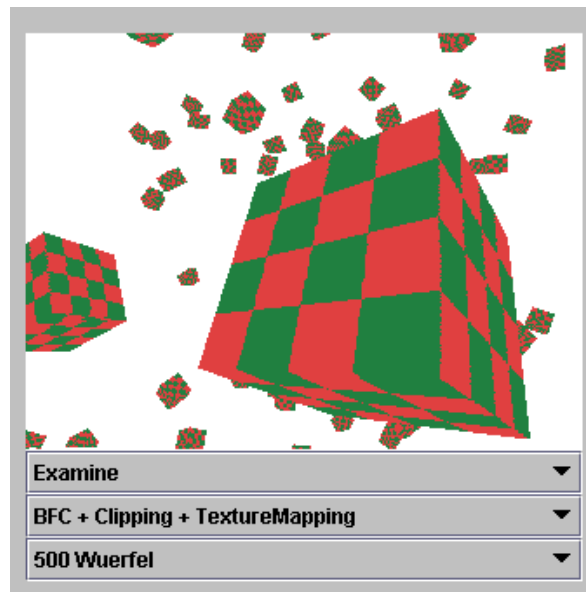


Abbildung 19.5: Screenshot vom Texture-Mapping-Applet

19.10 Java-Applet mit texturiertem Ikosaeder

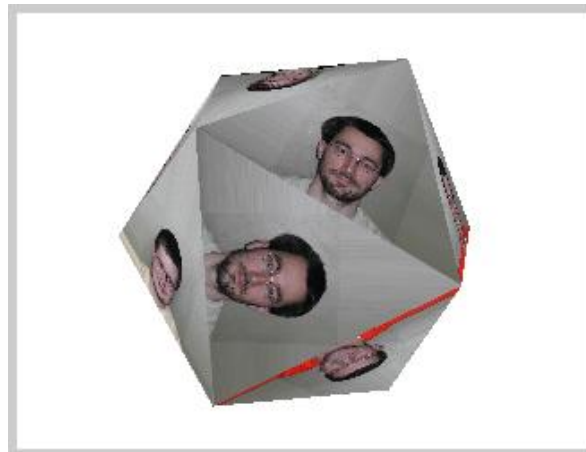


Abbildung 19.6: Screenshot vom Applet mit texturiertem Ikosaeder