

# Computergrafik.Online

## Sprechertext

Hochschule Furtwangen University

Fakultät Digitale Medien

Betreut von:

Prof. Jirka Dell'Oro-Friedl

Version: 1.0

Letzte Änderung: 09.11.2018

Autor: Berdan Der, Benedikt Grether , Steven Romanek,  
Davide Russo, Melanie Ratajczak, Lisa Würstle

|   |           |
|---|-----------|
| <b>1 Bits und Bytes</b>   | <b>5</b>  |
| 0101 Einleitung   | 5         |
| 0102 Dezimalsystem  | 5         |
| 0103 Binärsystem  | 6         |
| 0104 Hexadezimalsystem  | 6         |
| 0105 Interaktion: Dezimalsystem - Binärsystem - Hexadezimalsystem | 7         |
| 0106 Bits und Bytes   | 7         |
| 0107 Bezeichnung für größere Datenmengen                          | 7         |
| <b>2 Farbsysteme</b>  | <b>8</b>  |
| 0201 (A) Einleitung   | 8         |
| 0202 (A) Farbsehen  | 8         |
| 0203 (I) Farbsehen  | 9         |
| 0204 (A) Additive Farbmischung                                    | 9         |
| 0205 (A) RGB-Farbsystem   | 9         |
| 0206 (I) RGB-Farbsystem   | 10        |
| 0207 (A) HSB-Farbsystem   | 10        |
| 0208 (I) HSB-Farbsystem   | 10        |
| 0209 (A) YUV- und YCbCr-Farbsystem                                | 10        |
| 0210 (A) Subtraktive Farbmischung (CMY)                           | 11        |
| 0211 (A) CMYK-Farbsystem  | 11        |
| 0212 (A) LAB-Farbsystem   | 11        |
| <b>3 Rastergrafiken</b>   | <b>13</b> |
| 0301 Einleitung   | 13        |
| 0302 Farbtiefe Graustufen   | 13        |
| 0303 Farbtiefe Graustufen – Interaktion                           | 13        |
| 0304 Hexadezimaler Farbcode                                       | 13        |
| 0305 Hexadezimaler Farbcode – Interaktion                         | 14        |
| 0306 Farbtiefe 16/24/30/48 Bit                                    | 14        |
| 0307 Farbtiefe 16/24/30/48 Bit – Interaktion                      | 14        |
| 0308 Indizierte Farben  | 15        |
| 0309 Speicherbedarf   | 15        |
| 0310 Speicherbedarf - Interaktion                                 | 15        |
| <b>4 Farbmischmodi</b>  | <b>16</b> |
| 0401 Farbmischmodi Einleitung                                     | 16        |
| 0402 Unterschiede der Farbmischmodi: Erklärung                    | 16        |
| 0403 Unterschiede der Farbmischmodi – Interaktion                 | 19        |
| 0404 Histogramme – Erklärung                                      | 19        |
| <b>5 Auflösung und Ausgabe</b>                                    | <b>20</b> |
| 0501 Einleitung   | 20        |

|                                    |           |
|------------------------------------|-----------|
| 0502 Auflösung                     | 20        |
| 0503 Pixels per Inch               | 20        |
| 0504 Pixels per Inch               | 21        |
| 0505 Auflösung eines Monitors      | 21        |
| 0506 Auflösung eines Monitors      | 21        |
| 0507 Drucktechnik                  | 21        |
| 0508 Druckraster                   | 22        |
| 0509 Druck                         | 22        |
| 0510 Rasterwinkelung               | 22        |
| 0511 Rasterwinkelung               | 23        |
| <b>6 Bilddatenreduktion</b>        | <b>24</b> |
| 0601 (A) Einleitung                | 24        |
| 0602 (A) Skalieren und Beschneiden | 24        |
| 0603 (A) Farbtiefenreduktion       | 25        |
| 0604 (I) Farbtiefenreduktion       | 25        |
| 0605 (A) RLE                       | 25        |
| 0606 (A) LZW                       | 25        |
| 0607 (A) Huffman-Kodierung         | 26        |
| 0608 (A) JPG                       | 26        |
| 0609 (I) JPG                       | 27        |
| 0610 (A) GIF                       | 27        |
| 0611 (A) PNG                       | 28        |
| 0612 (A) Tipps & Tricks            | 28        |
| <b>7 Vektorgrafik</b>              | <b>30</b> |
| 070001                             | 30        |
| 070101                             | 30        |
| 070103                             | 30        |
| 070201                             | 30        |
| 070202                             | 31        |
| 070301                             | 31        |
| 070303                             | 31        |
| 070401                             | 31        |
| 070501                             | 32        |
| 070503                             | 32        |
| 070601                             | 32        |
| 070602                             | 32        |
| 070701                             | 32        |
| 070702                             | 33        |
| <b>8 Szenenaufbau</b>              | <b>34</b> |
| 0801 Einleitung                    | 34        |
| 0802 Aufbau von Meshes             | 34        |

|  |           |
|--|-----------|
| 0803 Aufbau von Meshes   | 34        |
| 0804 Transformationen  | 34        |
| 0805 Transformationen  | 35        |
| 0806 Hierarchie  | 35        |
| 0807 Hierarchie  | 35        |
| 0808 Farbe und Licht   | 35        |
| 0809 Lichtquellen-Typen  | 36        |
| 0810 Lichtquellen-Typen  | 36        |
| 0811 Kamera  | 36        |
| 0812 Perspektivische Projektion                                | 36        |
| 0813 Parallelprojektion  | 37        |
| 0814 Projektion  | 37        |
| <b>9 Rendering</b>   | <b>37</b> |
| <b>10 Shader</b>   | <b>38</b> |
| 1001 (A) Einleitung  | 38        |
| 1002 (A) Flat-Shading  | 38        |
| 1003 (A) Gouraud-Shading                                       | 39        |
| 1004 (A) Phong-Shading   | 39        |
| 1005 (I) Vergleich zwischen Flat-, Gouraud - und Phong-Shading | 40        |
| 1006 (A) Vertex Shader   | 40        |
| 1007 (I) Vertex Shader   | 40        |
| 1008 (A) Geometry-Shader                                       | 40        |
| 1009 (A) Pixel- / Fragment-Shader                              | 41        |
| 1010 (A) Toon-/Cel-Shading                                     | 41        |
| 1011 (A) Programmierung eines Shaders                          | 42        |
| <b>11 Modellierung</b>   | <b>43</b> |
| 1101 Einführung  | 43        |
| 1102 Grundformen der 3D-Modellierung                           | 43        |
| 110201 Grundformen der 3D-Modellierung – Interaktion           | 43        |
| 1103 High- und Low-Poly-Modelle                                | 44        |
| 110301 High- und Low-Poly-Modelle – Interaktion                | 44        |
| 1104 Extrudieren   | 44        |
| 1105 Lathe/Rotationskörper                                     | 45        |
| 1106 Mirror-Modifier/Operation                                 | 45        |
| 110601 Mirror-Modifier/Operation – Interaktion                 | 46        |
| 1107 Subdivision   | 46        |
| 110701 Subdivision Surface – Interaktion                       | 46        |
| 1108 Sculpting   | 47        |
| 1109 Boolean Modifier/Operation                                | 47        |
| 110901 Boolean Modifier/Operation – Interaktion                | 47        |

|  |           |
|--|-----------|
| <b>12 Texturen</b>                     | <b>48</b> |
| 1201 (A) Einleitung                    | 48        |
| 1202 (I) Einleitung                    | 48        |
| 1203 (A) Texturkoordinaten             | 48        |
| 1204 (A) UV-Mapping                    | 49        |
| 1205 (I) UV-Mapping                    | 49        |
| 1206 (A) Mip-Mapping                   | 49        |
| 1207 (A) Bump-Mapping                  | 50        |
| 1208 (A) Normal-Mapping                | 50        |
| 1209 (A) Displacement-Mapping          | 51        |
| 1210 (I) Vergleich                     | 51        |
| 1211 (A) Environment-Mapping           | 51        |
| 1212 (A) Kubisches Environment-Mapping | 52        |
| <b>13 Animation</b>                    | <b>52</b> |

# 1 Bits und Bytes

## 0101 Einleitung

Die erste urkundlich erwähnte Rechenmaschine wurde 1623 von Wilhelm Schickard in einem Brief an Johannes Kepler knapp beschrieben. Die Maschine habe aus einem Addier- und Subtrahierwerk sowie einer Vorrichtung zum Multiplizieren und Dividieren bestanden.

In einer Abhandlung für die berühmte Pariser Akademie der Wissenschaften (Académie des Sciences) legte Gottfried Wilhelm Leibniz 1703 das nur auf 0 und 1 basierende binäre Zahlensystem dar, das er unabhängig von anderen fand.

In diesem System werden alle Zahlen durch Verbindungen von 0 und 1 dargestellt. Er war der erste, der eine auf dem binären Zahlensystem beruhende Rechenmaschine konzipierte; wenngleich sie nicht realisiert wurde.

1837 veröffentlichte Charles Babbage die ersten Beschreibungen zu seiner Analytical Engine. Bedingt durch finanzielle und technische Probleme, wurden aber nur wenige Komponenten tatsächlich gebaut.

1941 war die Zuse Z3 von Konrad Zuse in Zusammenarbeit mit Helmut Schreyer der erste universell programmierbare Rechner, der tatsächlich gebaut wurde und funktionierte. Außerdem ist sie auch der erste Rechner, der auf dem binären Zahlensystem und der binären Schaltungstechnik basiert.

Die Z3 bestand aus 600 Relais im Rechenwerk und 1600 Relais im Speicher. Die Berechnung wurde mittels einer Lochkarte eingegeben. Leider ist von diesem Rechner nur eine Zeichnung übriggeblieben, da die Z3 bei einem Bombenangriff am 21. Dezember 1943 zerstört wurde.

In der Computergrafik und in allen anderen Computerprogrammen wird weiterhin auf das binäre Zahlensystem gesetzt, welches unter anderem bei der Darstellung von Farbtiefen zum Einsatz kommt.

## 0102 Dezimalsystem

Das Dezimalsystem, welches aufgrund der Basis 10 auch oft Zehnersystem genannt wird, ist ein Stellenwertsystem zur Darstellung von 10er Zahlen.

Hat man im Zehnersystem von 0 bis 9 gezählt und möchte die Zählung fortsetzen, werden die folgenden Zahlen aus den bereits vorhandenen Zahlen zusammengesetzt, dies gilt auch für die nächsten Vorgestellten Zahlensysteme.

So folgt nach der 9 dann die 10, dafür wird nun auf der 10er Stelle eine 1 und auf der 1er Stelle wieder eine 0 gesetzt, somit erhalten wir die Zahl 10.

Wird nun weiter hochgezählt so wird an der 1er Stelle die Zahl 0 mit der Zahl 1 ausgetauscht und daraus resultiert sich die Zahl 11.

Das Dezimalsystem ist heute das weltweit verbreitetste Zahlensystem.

## 0103 Binärsystem

Das Binärsystem, auch Zweiersystem genannt hat die Basis 2, die Zahlen werden nur mit den Ziffern 0 und 1 dargestellt. Das Binärsystem wird folgendermaßen verwendet:

Mit jeder Stelle, die beim Binärsystem hinzugefügt wird, verdoppelt sich der Zahlenwert. Wenn man also mit dem Zahlenwert 1 anfängt, hat die nächste Stelle den Wert 2 und die übernächste den Wert 4.

Das Binärsystem ist in der Digitaltechnik das wichtigste Zahlensystem, da die Zahlen schon mit vergleichsweise einfacher Technik realisiert werden können.

Die Binärzahlen werden durch Schalter repräsentiert, die die Zustände 1 oder 0 haben können. Der Zustand An entspricht einer logischen 1, Aus entspricht einer logischen 0. Durch die Verknüpfung dieser Zustände können verschiedene Berechnungen realisiert werden.

Die Zahl 26 wird im Binärsystem aus den Zahlenwerten 16, 8 und 2 zusammengesetzt

## 0104 Hexadezimalsystem

Im Hexadezimalsystem werden Zahlen zur Basis 16 dargestellt, jede Stelle versechzehnfacht sich.

Dabei können Zahlen von 0 bis 15 dargestellt werden. Also insgesamt 16 Zahlen. Die Zahlen von 0 bis 9 werden mit den Dezimalzahlen dargestellt, und die Zahlenwerte 10-15 mit den Buchstaben a – f.

In der Datenverarbeitung wird das Hexadezimalsystem sehr oft verwendet, um eine komfortablere Darstellung des Binärsystems zu ermöglichen.

Dabei werden 4 Bits zu einem Nibble zusammengefasst. Mit einem Nibble lassen sich im Hexadezimalsystem 16 Werte darstellen und der Wertebereich von 0 bis 15 im binären Zahlensystem.

## 0105 Interaktion: Dezimalsystem - Binärsystem - Hexadezimalsystem

Über den Regler kannst du die Zahl erhöhen und sehen, wie sie in den verschiedenen Zahlensystemen dargestellt wird.

## 0106 Bits und Bytes

Mit einem Bit wird die kleinste Informationseinheit eines Rechners angegeben, diese entspricht dem Zustand 1 oder 0 im Binärcode.

Bei binär kodierten Informationen ist die Anzahl der Bits eine Maßeinheit für die Datenmenge. Das Wort Bit ist eine Wortschöpfung aus dem englischen Ausdruck "binary digit", was mit Binärziffer übersetzt werden kann.

Ein Byte besteht aus insgesamt acht Bits und kann somit  $2^8$ , also 256 Zustände wiedergeben.

Mit der Einheit Bit pro Sekunde wird oft die Datentransferrate angegeben. DSL-Anbieter geben z.B. die Datenübertragungsrate von 16 oder 32 Megabit pro Sekunde an.

Die Einheit Byte wird für die Größe des Speicherplatzes verwendet, z.B. wird die Speicherkapazität eines USB-Sticks mit 8, 16 oder 32 Gigabyte angegeben.

## 0107 Bezeichnung für größere Datenmengen

Zu Beginn des Computer-Zeitalters wurden die Datenmengen und Speichergrößen als Potenzen zur Basis 2 verwendet.

Da dies für Nicht-Mathematiker schwierig zu verstehen war und es noch keine Präfixe für binäre Einheiten gab, bediente man sich an den Dezimalpräfixen.

Diese eigentlich falsche Praxis sorgt bis heute noch für Verwirrung. Man sagte zwar Kilobyte, meinte dabei aber nicht 1000 sondern 1024 Byte. Entsprechend ist ein Megabyte nicht eine Million sondern 1.048.576 Bytes.

Gegen Ende der 1990er Jahre führte die International Electrotechnical Commission deshalb Binärpräfixe ein. Für Angaben auf Basis einer Zweierpotenz sollten nicht Kilobyte, Megabyte und Gigabyte verwendet werden, sondern Kibibyte, Mebibyte und Gibibyte. Die Silbe bi steht für binär.

Als man vorwiegend in der Größenordnung von Kilo- und Megabytes arbeitete, waren die Abweichungen relativ klein und wurden daher meistens vernachlässigt. Doch



heute rechnet man in Giga- und Terabytes und hier werden die Abweichung relevant: beim GB sind es schon 7.4%, beim TB fast 10%.

## 2 Farbsysteme

### 0201 (A) Einleitung

Mithilfe des Binärsystems, welches im vorherigen Kapitel Bits und Bytes behandelt wird, können nicht nur Zahlen, sondern auch Farben definiert werden.

Es wird in additive und subtraktive Farbmischungen unterschieden, welche jeweils auf unterschiedlichen Primärfarben basieren. Additive Farbmischungen basieren auf den Primärfarben Rot, Grün und Blau. Die subtraktive Farbmischung basiert auf den Primärfarben Cyan, Magenta und Gelb. Diese zwei Arten werden in den folgenden Kapiteln genauer beschrieben.

Im Bereich der Computergrafik wird hauptsächlich mit dem RGB-, dem HSB- und dem YUV-Farbsystem gearbeitet.

### 0202 (A) Farbsehen

Bevor die unterschiedlichen Farbmischungen und Farbsysteme genauer betrachtet werden, wird geklärt, wie wir Menschen Farben überhaupt sehen können.

Das menschliche Auge verfügt über Rezeptoren, welche Zapfen und Stäbchen genannt werden. Diese Rezeptoren befinden sich an der Retina, der Netzhaut des Auges.

Während die Stäbchen für das Sehen bei Nacht zuständig sind, ermöglichen Zapfen uns das Farbsehen bei Tag.

Es gibt drei verschiedene Zapfentypen, die SML-Zapfen, die jeweils unterschiedlich empfindlich für gewisse Wellenlängenbereiche sind. Der Farbeindruck im Gehirn entsteht durch die verschiedenen Anteile an Licht von Rot, Grün und Blau.

Der Mensch kann ausschließlich Lichtfarben wahrnehmen. Eine Lichtfarbe stammt aus einer selbstleuchtenden Lichtquelle wie zum Beispiel der Sonne. Das menschliche Auge nimmt die Lichtfarbe als weißes Licht wahr, welches sich aber tatsächlich aus gleichen Anteilen von rotem, grünem und blauem Licht zusammensetzt.

Nicht selbstleuchtende Körper haben eine Körperfarbe, die zur Sichtbarmachung eine Lichtquelle benötigt. Körper erscheinen uns farbig, wenn sie einen bestimmten Anteil des weißen Lichts remittieren beziehungsweise zurücksenden und den Rest absorbieren, also schlucken.

## 0203 (I) Farbsehen

Wähle keine, eine, zwei oder alle drei Druckfarben für den Bedruckstoff aus und beobachte, welche der drei Lichtfarben geschluckt, und welche zurückgesendet werden.

## 0204 (A) Additive Farbmischung

Bei einer additiven Farbmischung werden die Farben und die Helligkeit aus den additiven Lichtfarben Rot, Grün und Blau zusammengesetzt. Diese Lichtfarben werden auch Primärfarben genannt.

Addiert man die Primärfarben mit ihrem Maximalwert, ergeben sie immer Weiß. Durch das Addieren von jeweils zwei Lichtfarben ergeben sich die Mischfarben Cyan, Magenta und Gelb. Diese Mischfarben werden auch Sekundärfarben genannt.

Neben dem RGB-Farbsystem gibt es auch noch das HSB- und YUV-Farbsystem, welche Varianten des RGB-Farbsystems sind. Die genannten Farbsysteme beschreiben alle den selben Farbraum.

Die additive Farbmischung bildet die Grundlage für die Farbdarstellung auf Bildschirmen und Monitoren.

## 0205 (A) RGB-Farbsystem

Das RGB-Farbsystem basiert auf den Primärfarben Rot, Grün und Blau.

Das Farbsystem lässt sich mit einem einfachen Würfel veranschaulichen. Dabei bestimmt die y-Achse den Rot-, die x-Achse den Blau- und die z-Achse den Grün-Wert. Bei einer Codierung von einem Byte kann jede der Farben einen Wert zwischen 0 und 255 annehmen, wobei 255 der Maximalwert ist.

Da die drei Farben somit in 256 Abstufungen angegeben werden können, ergeben sich 16,7 Millionen mögliche Farben im RGB-Farbraum und eine Farbtiefe von 24 Bit.

Das RGB-Farbsystem wird für die Farbdarstellung auf Computerbildschirmen und Fernsehgeräten verwendet. Dabei besteht jeder Pixel aus einem roten, einem grünen und einem blauen Pixel.

## 0206 (I) RGB-Farbsystem

Verschiebe die Regler R, G und B und beobachte, an welchem Punkt sich die ausgewählte Farbe im RGB-Farbraum befindet.

## 0207 (A) HSB-Farbsystem

Das HSB-Farbsystem beschreibt und definiert eine Farbe anhand dreier Eigenschaften:

Hue, dem Farbton, Saturation, der Sättigung und Brightness, der Helligkeit. Der Farbton wird dabei in Grad angegeben, die Sättigung und die Helligkeit prozentual.

0 Grad definieren den Farbton Rot, 120 Grad den Farbton Grün und 240 Grad den Farbton Blau.

Varianten, welche sich nur in der Helligkeitsskalierung von dem HSB-Farbsystem unterscheiden, sind das HSI- und das HSL-Farbsystem.

## 0208 (I) HSB-Farbsystem

Verschiebe den Regler H um einen Farbton auszuwählen. Lasse dir anschließend die Abstufungen mittels S, der Sättigung und B, der Helligkeit anzeigen.

## 0209 (A) YUV- und YCbCr-Farbsystem

Das YUV-Farbsystem wurde in den 60er Jahren für das analoge Farbfernsehen entwickelt. Damals wurde ein System geschaffen, welches kompatibel mit dem Schwarz-Weiß-Fernsehen und an das menschliche Auge angepasst ist.

Vor der Übertragung des Farbsignals werden die RGB-Werte in das YUV-System umgerechnet. Dabei wird die Farbauflösung verringert, da das menschliche Auge Farbunterschiede auch in geringerer Auflösung sehr gut unterscheiden kann.

In diesem mathematischen System wird ein RGB-Farbwert in die Helligkeit Y (Luminanz) und zwei Farbkomponenten U und V aufgeteilt. Diese Komponenten enthalten die Farbinformation, welche in den Farbdifferenzsignalen Rot minus Y und Blau minus Y steckt. Die Information zu Grün lässt sich aus dem YUV-Signal berechnen.

Für das Digitalfernsehen wurde das YUV-System zum YCbCr-System weiterentwickelt,

welches ebenfalls für JPEG-Bilder und MPEG-Videos verwendet wird. Das C steht jeweils für Chrominanz und somit für die Buntheit. Cb beschreibt den Bereich zwischen Blau und Gelb, Cr den zwischen Rot und Grün.

Durch die Erkenntnis, dass das menschliche Auge sehr viel weniger empfindlich für Farbunterschiede wie für Helligkeitsunterschiede ist, wird das YUV-System heute noch vielseitig eingesetzt, wie beispielsweise im Bereich der Texturen.

## 0210 (A) Subtraktive Farbmischung (CMY)

Die Primärfarben der subtraktiven Farbmischung sind Cyan, Magenta und Gelb. Um Gelb nicht mit Grün zu verwechseln wird die gelbe Farbkomponente als Yellow und somit mit einem Y bezeichnet.

Das Grundprinzip bei der subtraktiven Farbmischung ist, Farbtöne aus den drei Primärfarben zu mischen und sie auf ein weißes Papier zu drucken. So ergibt beispielsweise das Mischen von Magenta und Gelb in gleicher Intensität den Farbton Rot. Dabei wird das Licht der Farben durch Pigmente absorbiert oder durch Farbfilter subtrahiert. Die resultierende Mischfarbe, die vom menschlichen Auge wahrgenommen wird, ist die Körperfarbe.

Theoretisch sollte sich beim Addieren der drei Grundfarben in voller Intensität die Farbe Schwarz ergeben.

## 0211 (A) CMYK-Farbsystem

Beim CMYK-Farbsystem wird auf ein weißes Blatt Farbe gedruckt und gleichzeitig Helligkeit subtrahiert.

In diesem Farbsystem werden die einzelnen Farbwerte prozentual zwischen 0 und 100% angegeben, wobei 100% der maximalen Intensität der Farbe entspricht. Da es in der Praxis nur schwer gelingt, aus den drei Grundfarben die Farbe Schwarz zu mischen, wird im Druck zusätzlich die Keyfarbe Schwarz hinzugefügt.

Die Bezeichnung Key stammt aus dem Druck, da Schwarz auf einer Key-Plate beziehungsweise einer Schlüssel-Platte gedruckt wird.

Durch das Hinzufügen von Schwarz wird das gedruckte Schriftbild schärfer und Bilder kontrastreicher. Zusätzlich wird Druckfarbe gespart, da statt drei Farben nur eine verwendet wird.

## 0212 (A) LAB-Farbsystem

Das LAB-Farbsystem ist ein theoretisches System und umfasst alle für den Menschen wahrnehmbaren Farben. Technisch ist es jedoch nicht möglich, diese Farben darzustellen. Der große Vorteil dieses Farbsystems ist die Geräteunabhängigkeit.

Das LAB-Farbsystem wird durch drei Faktoren bestimmt: die Luminanz, die Farbkomponente a und die Farbkomponente b.

Für die Farbe Weiß ist der Luminanzwert 100, für Schwarz 0. Die beiden Farbkomponenten a und b können Werte zwischen -128 und +127 annehmen.

Ist der Wert der Farbkomponente a gleich 0, können nur Farbtöne im Bereich Blau bis Gelb dargestellt werden.

Ist der Wert der Farbkomponente b gleich 0, können nur Farbtöne im Bereich Grün bis Rot dargestellt werden.

Hier sieht man, welch theoretisch breites Farbspektrum das LAB-Farbsystem im Gegensatz zum RGB- und CMYK-Farbsystem umfasst.

## 3 Rastergrafiken

### 0301 Einleitung

Eine Rastergrafik wird auch Pixelgrafik genannt und ist die Beschreibung eines Bildes in Form von Pixeln. Diese werden in einem Raster angeordnet.

Eine Rastergrafik ist beschreibbar durch ihre Höhe und Breite und ihre Farbtiefe.

Rastergrafiken eignen sich für Bilder, die zu komplex sind, um sie als Vektorgrafiken darzustellen, wie z. B. Fotos.

### 0302 Farbtiefe Graustufen

Bei einer Rastergrafik wird ein Bild in ein Raster unterteilt wobei die Farb- & Helligkeitswerte in den Pixeln gespeichert werden.

Die Farbtiefe ist eine Größe, die beschreibt, wie viele Farb- bzw. Grauwerte ein Pixel darstellen kann. Besitzt ein Bild 1 Bit Farbtiefe können genau zwei Farben dargestellt werden z. B. schwarz und weiß.

Die Anzahl der möglichen Farben lässt sich durch eine Formel einfach berechnen. Es wird die Bitanzahl als Exponent zur Basis zwei genommen.

Je mehr Bit zur Verfügung stehen, desto mehr Farbabstufungen sind möglich und desto besser lässt sich ein Bild darstellen. Das gilt insbesondere für Helligkeits- oder Farbverläufe.

### 0303 Farbtiefe Graustufen – Interaktion

Betrachte, welche Auswirkungen das Verändern der Bit-Anzahl auf das Bild hat.

### 0304 Hexadezimaler Farbcode

Anders als beim Druck werden Farben, am Monitor, durch additive Farbmischung erstellt. Dazu werden die Farbkanäle rot, grün und blau überlagert. Bei den Farbkanälen handelt es sich um die Grundfarben des RGB-Farbraumes.

Um Farben einfacher identifizieren zu können, wurde sich auf eine einheitliche Kodierung geeinigt - den hexadezimalen Farbcode.

Dieser besteht aus drei zweistelligen Hexadezimalzahlen, die die jeweiligen Farbkanäle repräsentieren. Die erste Hexadezimalzahl stellt die Intensität von rot, die zweite von grün und die letzte von blau dar.

Eine Hexadezimalzahl kann alle Werte zwischen 00 und FF annehmen, wobei 00 für einen völlig ausgeschalteten und FF für einen Farbkanal mit maximaler Farbintensität steht. Somit würde #FFFF00 für maximal rot, maximal grün und kein blau stehen, womit die Farbe gelb dargestellt wird.

## 0305 Hexadezimaler Farbcode – Interaktion

Stelle die Farbintensitäten mit Hilfe der Schieberegler ein und betrachte die daraus resultierende Farbe.

## 0306 Farbtiefe 16/24/30/48 Bit

Eine Farbtiefe von 16 Bit wird High Color genannt. Bilder die über 16 Bit Farbtiefe besitzen stehen ca. 65.000 verschiedene Farbabstufungen zur Verfügung. Dem roten und blauen Farbkanal stehen dabei 5 Bit und dem grünen Farbkanal 6 Bit zur Verfügung. Dass dem grünen Farbkanal mehr Bit zur Verfügung stehen hängt mit der menschlichen Farbwahrnehmung zusammen, da das menschliche Auge für Grüntöne besonders empfindlich ist.

Bei einem Bild mit 24 Bit erhalten die Farbkanäle dagegen alle 8 Bit. Dadurch kann jeder Kanal 256 Farben und alle zusammen ca. 16,7 Millionen Farben darstellen. Diese Farbtiefe wird als True Color bezeichnet, da diese natürlich wirken. Des Weiteren können durch True Color komplexere Farbverläufe dargestellt werden.

Neben True und High Color gibt es auch Deep Color, welcher Farbtiefen von 30 bis 48 Bit abdeckt. Hierdurch sind extrem feine Farbabstufungen möglich.

## 0307 Farbtiefe 16/24/30/48 Bit – Interaktion

Wähle eine Farbtiefe und betrachte welche Auswirkungen das auf das Bild hat.

## 0308 Indizierte Farben

Mit indizierten Farben beschreibt man in der Computergrafik ein Verfahren, das zur Speicherung einer Rastergrafik dient.

Hierbei erhalten die Pixel keinen direkten Farbwert, wie es üblicherweise der Fall ist. Stattdessen wird am Anfang eine Farbtabelle erstellt, die zwei Eigenschaften hat: die Farbe und einen Index, der diese Farbe repräsentiert.

Nun werden in den Pixeln Indizes gespeichert, die auf die jeweilige Farbe in der Farbtabelle verweisen.

Bilder, die wenige unterschiedliche Farben enthalten, sparen sich durch dieses Verfahren Speicherplatz. Heutzutage findet man diese Methode z. B. in GIF-Bildern.

## 0309 Speicherbedarf

Den Speicherbedarf eines Bildes kann man leicht berechnen. Dieser ist abhängig von drei Komponenten: der Höhe, der Breite und der Farbtiefe des Bildes.

Nehmen wir als Beispiel ein Bild welches 820px hoch und 388px breit ist. Des Weiteren besitzt es eine Farbtiefe von 24 Bit.

Als ersten Schritt führen wir eine Multiplikation aller Werte durch.

Daraus ergeben sich 318.160 Bit Speicherbedarf. Jetzt wird das Ergebnis in Megabyte umrechnen.

Dazu teilen wir 318.160 Bit durch 8, wodurch wir 39.770 Bytes erhalten.

Teilen wir das Ergebnis nun durch 1024 erhalten wir 38,8 Kibibytes.

Zum Schluss muss das Ergebnis ein zweites Mal durch 1024 geteilt werden, wodurch wir ein Endergebnis von 0,04 Megabytes für den Speicherbedarf erlangen.

## 0310 Speicherbedarf - Interaktion

Bestimme alle Komponenten selbst und schaue wie groß die Datei schlussendlich ist. Wähle dazu eine Farbtiefe aus und skaliere das Bild über den Schieberegler.



# 4 Farbmischmodi

## 0401 Farbmischmodi Einleitung

Der Farbmischmodus bestimmt wie Farbwerte bei der Überlagerung von Pixeln verrechnet werden sollen. Dabei werden für jeden einzelnen Bildpunkt die Farbwerte von Bild eins - und Bild 2 berechnet und als neuer Farbwert dargestellt.

Dies wird bei der Bildbearbeitung verwendet um den Bildern einen gewissen Look zu geben.

Einige der Farbmischmodi sind:

Abdunkeln

Weiches Licht

Hartes Licht

Strahlendes Licht

Multiplizieren

Negativ Multiplizieren

Dunklere Farbe

Hart Mischen

## 0402 Unterschiede der Farbmischmodi: Erklärung

Es gibt unterschiedliche Farbmischmodi. Nun werden ihre Funktionsweisen erklärt.

Abdunkeln:

Wählt anhand der Farbinformationen in den einzelnen Farbkanälen die jeweils dunklere Farbe die später aufgetragen wird als Farbe, die beim Mischen der beiden Farben entsteht. Pixel, die heller als die aufgetragenen Farben sind, werden ersetzt, die Dunkleren, bleiben unverändert.

Weiches Licht:

Je nach zu auftragender Farbe werden die Farben aufgehellt oder verdunkelt. Wenn die aufzutragende Farbe heller als 50-prozentiges Grau ist, wird das Bild heller.

Ist aber das Bild dunkler als 50-prozentiges Grau, dann wird es dunkler.

Durch das Mischen mit reinem Schwarz oder Weiß wird ein deutlich dunklerer oder hellerer Bereich erzeugt, das Ergebnis ist jedoch kein reines Schwarz oder Weiß.

Multiplizieren:

Multipliziert anhand der Farbinformationen in den einzelnen Kanälen die Originalfarbe im Bild mit der beim Verrechnen der beiden Bilder entstandene Farbe. Dabei ist die entstehende Farbe immer eine dunklere Farbe.

Beim Multiplizieren einer Farbe mit Schwarz entsteht Schwarz.

Beim Multiplizieren mit Weiß bleibt die Farbe unverändert. Werden andere Farben als Weiß oder Schwarz verwendet so ist das zu vergleichen als würde man mit mehreren Textmarkern über ein Bild malen und deren Farben sich dann überlagern: das Bild wird dunkler.

Negativ Multiplizieren:

Multipliziert anhand der Farbinformationen in den einzelnen Kanälen die "Negative" oder die aufzutragende Farbe oder die Originalfarbe im Bild.

Die neu entstehende Farbe nach dem verrechnen der beiden Bilder ist immer eine hellere Farbe.

Bei "Negativ multiplizieren" mit Schwarz bleibt die Farbe unverändert.

Bei "Negativ multiplizieren" mit Weiß entsteht Weiß. Die Wirkung gleicht dem Übereinander projizieren mehrerer Dias.

### Hartes Licht

- Führt eine Multiplikation bzw. eine Negativmultiplikation der Farben durch Abhängigkeiten von der neu aufzutragenden Farbe. Die Wirkung gleicht dem Beleuchten des Bildes mit einem Spot-Strahler mit direktem Licht. Wenn die aufzutragende Farbe (Lichtquelle) heller als 50- prozentiges Grau ist, wird das Bild heller (ähnlich wie "Negativ Multiplizieren").

Diese Option eignet sich daher zum Hinzufügen von Lichtern zu Bildern.

Wenn die aufzutragende Farbe dunkler als 50-prozentiges Grau ist, wird das Bild dunkler (ähnlich dem Multiplizieren). Diese Option eignet sich daher zum Hinzufügen von räumlicher Tiefe in Bildern.

Das Multiplizieren mit reinem Schwarz bzw. Weiß erzeugt reines Schwarz bzw. Weiß.

### Strahlendes Licht

- Die Farben werden je nach der aufzutragenden Farbe durch Erhöhen oder Verringern des Kontrasts aufgehellt oder nachbelichtet.

Wenn die aufzutragende Farbe (Lichtquelle) heller als 50- prozentiges Grau ist, wird das Bild durch Verringern des Kontrasts heller.

Wenn die aufzutragende Farbe dunkler als 50-prozentiges Grau ist, wird das Bild durch Erhöhen des Kontrasts dunkler.

### Dunklere Farbe:

Vergleicht die Summe aller Kanalwerte der aufzutragenden Farbe - oder die Originalfarbe des Bildes und zeigt die Farbe mit dem niedrigeren Wert an. Mit „Dunklere Farbe“ werden die beiden dunkleren Farben nicht zu einer dritten Farbe gemischt, da für die neu entstehende Farbe jeweils die aufzutragende Farbe - oder die Originalfarbe des Bildes mit dem niedrigsten Kanalwert verwendet wird.

Hart Mischen:

Fügt den Wert des Rot-, Grün- und Blaukanals der aufzutragenden Farbe zu den RGB-Werten die Originalfarbe im Bild hinzu. Wenn die Summe eines Kanals 255 oder höher ist, wird der Wert 255 zugewiesen, ist die Summe kleiner als 255, wird der Wert 0 verwendet. Aus diesem Grund haben alle angeglichenen Pixel als Werte für den Rot-, Grün- und Blaukanal 0 oder 254. Dadurch werden alle Pixel in die additiven Primärfarben (Rot, Grün oder Blau), in Weiß oder in Schwarz geändert.

## 0403 Unterschiede der Farbmischmodi – Interaktion

Wähle einen Farbmischmodus aus und betrachte das Ergebnis. Das Testbild kann über den Roboter verschoben werden.

## 0404 Histogramme – Erklärung

Ein Histogramm ist eine grafische Darstellung einer Häufigkeitsverteilung in Bezug auf ein quantitatives Merkmal. In der Bildbearbeitung ist eins der möglichen Merkmale die Verteilung verschiedener Helligkeitsstufen eines Bildes.

Das Histogramm zeigt dabei die Helligkeitsverteilung in den Tiefen im linken Bereich, in den Mitteltönen im mittleren Bereich und in den Lichtern im rechten Teil.

Ein Histogramm bietet auch einen schnellen Überblick über den Tonwertbereich des Bildes, den sogenannten Key-Typ. Bei einem Low-Key-Bild das leicht unterbelichtet ist, konzentrieren sich die Helligkeitsverteilung in den Tiefen, während die Helligkeitsverteilung bei einem High-Key-Bild das leicht überbelichtet ist in den Lichtern anzutreffen sind. Bei einem Bild mit durchschnittlichen Farbwerten sind Details vor allem in den Mitteltönen sichtbar.

# 5 Auflösung und Ausgabe

## 0501 Einleitung

Die kleinste, speicherbare, farbige Informationseinheit eines digitalisierten Bildes oder einer Graphik ist ein Pixel. Dieser Begriff ist zusammengesetzt aus Picture und Element. Jedoch können Pixel unterschiedlich groß sein und erst in Verbindung mit dem Ausgabemedium hat man einen vergleichbaren Wert. In diese Fläche passen 50 Pixel dieser Größe. Wenn sich die Größe der Pixel halbiert passen auf dieselbe Fläche viermal so viele, also 200 Pixel.

## 0502 Auflösung

Die Eingabeauflösung, oder auch örtliche Auflösung, ist die Auflösung der ursprünglichen Bilddatei. Durch sie wird der maximale Detailgrad festgelegt. Sie wird in Megapixeln angegeben und bestimmt, indem man Breite mal Höhe in Pixeln rechnet. Ein Bild mit 1440 mal 900 Pixeln hätte demnach eine Eingabeauflösung von circa 1,3 Megapixeln. Die Ausgabeauflösung wird bestimmt, indem man die Eingabeauflösung mit den Maßen des Ausgabemediums ins Verhältnis setzt. Sie wird in Pixels per Inch angegeben.

## 0503 Pixels per Inch

Zum Druck eines digitalen Bildes bestimmt die Eingabeauflösung die Ausgabeauflösung des Bildes. Zum Beispiel soll ein Bild mit 630 mal 472 Pixeln in einem Bildformat von 5,33 mal 4 cm gedruckt werden. In dem man die Pixelanzahl der Breite durch die Breite des Bildformats teilt, erhält man eine Eingabeauflösung von 118 Pixeln pro Zentimeter. Da standardmäßig mit der Einheit Pixels per Inch, also Pixel pro Zoll, gearbeitet wird, muss das Ergebnis mal 2,54 cm pro Zoll genommen werden. Wir kommen somit ca. auf eine Ausgabeauflösung von 300 Pixels per Inch, abgekürzt PPI.

## 0504 Pixels per Inch

Verändere die Pixelanzahl und die Bildgröße. Lasse dir die Eingabe- und Ausgabeauflösung berechnen.

## 0505 Auflösung eines Monitors

Ein Monitorpixel ist quadratisch und besteht aus drei RGB-Subpixeln, sogenannten Dots, also einem roten, einem grünen und einem blauen Dot, welche die Endfarbe des Pixels additiv zusammenmischen. Als Dot-Pitch bezeichnet man den Abstand zwischen zwei gleichfarbigen Dots, also den Abstand benachbarter Pixel. Die Ausgabeauflösung des Monitors kann in Anzahl Pixel in der Breite mal Anzahl Pixel in der Höhe angegeben werden. Pixel können allerdings von unterschiedlicher Größe sein. Um die tatsächliche Auflösung in Pixels per Inch zu berechnen, kann zum Beispiel eine Beziehung zwischen der Pixelanzahl und der Diagonalen, sprich des Seitenverhältnisses, hergestellt werden. Dazu bildet man ein rechtwinkliges Dreieck aus der Breite, Höhe, und der Diagonalen des Bildschirms. Durch den Satz des Pythagoras erhält man so den Dot-Pitch entlang der Diagonalen. Dieser Wert in Pixeln kann nun durch die Diagonale in Zoll geteilt werden und es ergibt sich die Monitorauslösung in Pixels per Inch.

## 0506 Auflösung eines Monitors

Vergrößere den Bildausschnitt, um die Dots eines Monitors zu erkennen.

## 0507 Drucktechnik

In der Drucktechnik sprechen wir ebenfalls von Dots per Inch und meinen damit, wie viele Druckpunkte auf einem Inch sind. Eine hohe Anzahl Druckpunkte ist wichtig, wenn Details wie filigrane Linien und Muster dargestellt werden sollen. Echte Halbtöne können nicht gedruckt werden. Um sie zu simulieren wird das Bild in Rasterpunkte zerlegt. Diese variieren je nach Helligkeitswert in ihrer Größe oder Häufigkeit. Diese Rasterung kann nach verschiedenen Verfahren durchgeführt werden.

## 0508 Druckraster

Bei einer Rasterung werden Tonwerte verschiedener Helligkeiten simuliert, indem Druckpunkte variierender Größe oder Häufigkeit erzeugt werden. Es gibt drei Rasterungsgrundtypen: Zum ersten die amplitudenmodulierte Rasterung, außerdem die frequenzmodulierte Rasterung und schließlich eine Mischform, ein sogenanntes Hybrid-Rasterungsverfahren. Bei der amplitudenmodulierten Rasterung werden Tonwerte durch die Größe der Druckpunkte simuliert. Das Bild wird in Rasterzellen aufgeteilt, auf denen Rasterpunkte erzeugt werden. Die Punktform kann quadratisch, elliptisch oder kreisförmig sein. Bei der frequenzmodulierten Rasterung wird die Häufigkeit gleich großer Druckpunkte variiert, um echte Halbtöne zu simulieren. Diese Druckpunkte werden nach dem Zufallsprinzip, ohne periodisch wiederkehrende Struktur verteilt.

## 0509 Druck

Wähle nun zwischen dem Original, der amplitudenmoduliert gerasterten Version oder der frequenzmoduliert gerasterten Version und vergrößere das Bild. Außerdem kannst du die Dot-Anzahl verändern.

## 0510 Rasterwinkelung

Damit die Rasterpunkte der einzelnen Farben sich nicht überlagern, sondern im richtigen Verhältnis nebeneinander gedruckt werden, müssen die Raster unterschiedliche Winkel zueinander besitzen. Werden die Winkel nicht richtig gewählt, entsteht ein Moiré, ein scheinbares Gittermuster. Bei einem Druck mit den vier Farben Cyan, Magenta, Gelb und Schwarz müssen für die Farben verschiedene Winkel gewählt werden, um einen Moiré-Effekt zu verhindern. Handelt es sich bei den Druckpunkten um symmetrisch aufgebaute, wie quadratische oder kreisförmige Druckpunkte, so bekommt die Farbe Cyan den Winkel  $75^\circ$ , Magenta  $15^\circ$ , Gelb  $0^\circ$  und Schwarz den Winkel  $45^\circ$ . Die für den Menschen unauffälligste Farbe bekommt den auffälligsten Winkel: So wird schwarz auf eine  $45^\circ$ -Winkelung gelegt und Gelb, die auffälligste Farbe auf den unauffälligsten Winkel  $0^\circ$ . Wird jedoch mit

unsymmetrischen, elliptischen Druckpunkten gerastert, ist die Winkelverteilung anders: Gelb:  $0^\circ$ , Magenta:  $45^\circ$ , Cyan:  $165^\circ$  und Schwarz:  $105^\circ$ . Das liegt daran, dass statt  $90^\circ$ ,  $180^\circ$  für die Verwinkelung zwischen den Rastern zur Verfügung steht. So können die Raster bis zu  $60^\circ$ -Winkelabstand zueinander haben. Die Rasterung mit elliptischen Druckpunkten gilt heute als Standard

## 0511 Rasterwinkelung

Verändere die Winkel der Rasterwinkelungen und beobachte den Moiré-Effekt.



# 6 Bilddatenreduktion

## 0601 (A) Einleitung

Beim Arbeiten mit digitalen Bildern können sehr schnell große Datenmengen entstehen. Um dennoch eine kompakte Speicherung und schnelle Übertragung zu gewährleisten, ist es häufig sinnvoll, die Bilddaten zu reduzieren.

Zunächst kann das Bild auf einen relevanten Bereich zugeschnitten, skaliert und in der Farbtiefe reduziert werden. Anschließend wird es komprimiert, also platzsparend kodiert, wobei zwischen verlustfreier und verlustbehafteter Kompression gewählt werden kann.

Nun ist das Bild bereit um abgespeichert oder verschickt zu werden. Will man es nun wiederverwenden, so wird es vorher dekomprimiert. Bei der verlustfreien Kompression kann das Bild dabei vollständig wiederhergestellt werden, während dies bei der verlustbehafteten nicht möglich ist.

## 0602 (A) Skalieren und Beschneiden

Bilddatenreduktion muss nicht automatisch Kompression bedeuten. Bilddaten können vorab durch einfache Beschränkungen ebenfalls reduziert werden. Zum Beispiel indem die Pixelanzahl verringert wird. Beschneidet man das Bild, und skaliert es anschließend auf eine geringere Kantenlänge, so wird die Bilddatenmenge vorab erheblich reduziert. Dabei muss aber drauf geachtet werden, welche Skalierungsmethode gewählt ist.

Pixelwiederholung ist eine Methode, die Pixel beim Verkleinern weglässt, was unschöne Effekte auf das Bild haben kann.

Interpolation hat in der Regel ein schöneres optisches Ergebnis zur Folge, erzeugt aber neue Farbwischenstufen. Im Gegensatz zur Pixelwiederholung, hat das Bild nun mehr Farben als vor der Skalierung.

## 0603 (A) Farbtiefenreduktion

Die Farbtiefe ist die Anzahl der Bits die pro Pixel zur Speicherung der Farbinformation zu Verfügung stehen. Üblich sind Farbtiefen wie 8,16, 24 oder 32 Bit.

Manchmal überschreitet die gewählte Farbtiefe allerdings die vom Bild benötigten Farben. Bei 32 Bit Farbtiefe ist das vierte Byte in der Regel für einen unsichtbaren Transparenzkanal, denn sogenannten Alpha Kanal, vorgesehen. Diese Farbtiefe ist nur für bestimmte Aufgaben notwendig. Eine Reduktion auf 24 Bit spart in einem solchen Fall Speicherplatz.

Reduziert man die Farbtiefe weiter, so spart man noch mehr Speicherplatz. Es treten aber oft unschöne Farbverschiebungen auf.

## 0604 (I) Farbtiefenreduktion

Probiere nun selbst die Farbtiefe einzustellen und beobachte dabei die jeweilige Datenmenge.

## 0605 (A) RLE

RLE steht für run-length encoding, also Lauflängenkodierung und ist der einfachste der verlustfreien Kompressionsalgorithmen. Er untersucht die Daten des Bildes und fasst aufeinander folgende gleichfarbige Pixel zusammen. Dabei wird die Farbe einmal abgespeichert sowie ein Zahlenwert angegeben, der beschreibt in wie vielen Pixeln die Farbe hintereinander auftaucht. RLE ist eine sehr einfache Form der Kompression, kann aber bei Bildern im Bitmap-Format äußerst effektiv sein.

## 0606 (A) LZW

LZW ist ein weiterer verlustfreier Kompressionsalgorithmus, der nach seinen Erfindern Lempel, Ziv und Welch benannt ist. Dieser Algorithmus untersucht die Pixel

eines Bildes auf sich wiederholende Muster. Ausgehend von den im Bild vorhandenen Farben, wird ein sogenanntes Codebuch angelegt. Dieses wird erweitert, wenn der Algorithmus das Bild auf Pixelkombinationen untersucht. Findet er beim Codieren eine unbekannte Pixelkette, so speichert er den zuletzt gefunden bekannten Wert. Die unbekannte Pixelkombination wird anschließend im Codebuch abgelegt.

Anschließend, wenn die gleiche Pixelkette noch einmal gefunden wird, verweist LZW nur noch auf den Eintrag im Codebuch, was eine Einsparung an Speicherplatz bedeutet.

Diese Form der Kompression gibt nur bei bestimmten Bildinhalten ein optimales Ergebnis, schafft aber in der Regel eine bessere Datenreduktion als RLE.

## 0607 (A) Huffman-Kodierung

Die Huffman-Kodierung ist ebenfalls ein verlustfreier Kompressionsalgorithmus. Bei dieser Form der Kodierung nutzt man die Häufigkeit einzelner im Bild auftretender Farben aus.

Der Algorithmus untersucht zuerst das ganze Bild und ermittelt die Häufigkeit jeder einzelnen Farbe. Dann wird jeder Farbe ein Bit-Wert zugewiesen. Häufig auftretende Farben werden mit möglichst wenigen Bits dargestellt, während seltenere Farben mit mehr Bits repräsentiert werden. Die häufigste Farbe kann somit nur mit einem einzigen Bit kodiert werden.

Die hier dargestellten farbigen Pixel erzeugen einen Code von nur 19 Bit Länge.

## 0608 (A) JPG

JPG ist ein verlustbehaftetes Grafikformat, bei dem mehrere Kompressionsvorgänge durchgeführt werden. Verlustbehaftete Kompression bedeutet, dass bei der Kompression Bildteile zusammengefasst oder weggelassen werden. Dies geschieht hinsichtlich der optischen Ansprüche eines Menschen. Dadurch kann eine sehr kleine Datenmenge entstehen, die nach der Dekompression dem Original nur noch ähnlich ist.

Bei JPG im Speziellen läuft es folgendermaßen ab. Zuerst wird das Bild vom RGB in den YUV Farbraum umgerechnet, hierbei findet die erste Reduktion statt, da die Farbinformation U und V mit geringer Auflösung gespeichert werden.

Als Nächstes kommt es zur diskreten Kosinustransformation, oder kurz DCT genannt, die anhand des Y-Kanals beispielhaft gezeigt wird. Zunächst wird das Bild in 8x8 Pixel große Blöcke eingeteilt. Nun wird versucht, die Helligkeits- und Farbstrukturen in jedem Block mit Hilfe von Verläufen mathematisch anzunähern, diese Verläufe basieren auf einfachen vordefinierten Kosinuskurven. Verschiedene solcher Verläufe werden in unterschiedlicher Gewichtung überlagert. Dabei kann die Annäherung an das Originalbild durch die Anzahl der Überlagerungen beeinflusst werden. Zuletzt werden diese nun Zickzack ausgelesen und mit einer Huffman-Kodierung nochmals verlustfrei nachkomprimiert. Als Endergebnis erhält man eine um zehn bis hundertfach verkleinerte Datenmenge, aus der das ursprüngliche Bild wieder betrachtungsfähig konstruiert werden kann.

## 0609 (I) JPG

Probiere nun die JPG Kompression mithilfe des Reglers durchzuführen, beobachte dabei die Bildung von Blockartefakten im Bild.

## 0610 (A) GIF

Das Graphics Interchange Format, kurz GIF, ist ein Grafikformat mit einer verlustfreien Kompression. Das besondere an GIF ist die Möglichkeit der Speicherung von mehreren, auch übereinanderliegenden, Einzelbildern. Dadurch wird die Darstellung als Animation ermöglicht, was auch der Grund ist, warum GIF eine hohe Popularität besitzt.

GIF unterstützt, inklusive Transparenz, nur 256 indizierte Farben oder Graustufen, weshalb komplexe Bilder in ihrer Farbdarstellung reduziert werden müssen.

Außerdem unterstützt das Format auch so genanntes Interlacing, wodurch beim Laden eines GIFs die Auflösung Schritt für Schritt erhöht werden kann. Das war vor allem

früher von Vorteil, da trotz langsamer Internetverbindung schon etwas grob angezeigt wurde. Zur Kompression wird LZW verwendet, welchen wir schon in einem anderen Kapitel kennengelernt haben.

## 0611 (A) PNG

PNG steht für Portable Network Graphics und ist heute das meistverwendete verlustfreie Grafikformat im Internet. Die Entwicklung dieses Grafikformates begann Ende 1994 mit dem Ziel das Grafikformat GIF zu ersetzen.

Das PNG Format unterstützt verschiedene Farbtiefen, üblicherweise 8, 24 und 32 Bit, wobei die 32Bit Variante einen zusätzlichen 8-Bit-Kanal für Transparenz-Informationen enthält.

Die Kompression eines PNG verläuft in drei Schritten. Zuerst kommt es zum Vorfiltern, wo sehr ähnliche Farben auf einen Farbwert gesetzt werden.

Danach kommt es zur Wörterbuch-basierten Kodierung per LZ77 Algorithmus, welcher ein Vorgänger des im LZW verwendeten LZ78 ist.

Das Verlustfreie LZ77 sucht zu Beginn nach sich wiederholende Sequenzen von Daten. Wenn der Algorithmus auf eine Sequenz trifft, welche es schon einmal gab, gibt es nur einen Verweis auf die entsprechende Sequenz, was bei manchen Bildern viel Speicherplatz spart. Enthält ein Bild zum Beispiel zwei identische schwarze Kreise, so verbraucht nur der erste Kreis Speicherplatz.

Im Letzten Schritt werden die bis dahin erzeugten Daten noch mithilfe der Huffman-Kodierung komprimiert.

## 0612 (A) Tipps & Tricks

Um richtig zu komprimieren, sollten im Grafikbereich ein paar Tipps und Tricks beachtet werden. Nicht alle Kompressionsverfahren lassen sich gut auf jede Art von Bildern anwenden. Kompression wird häufig bei Bildern verwendet um diese im Internet zu publizieren, per Mail zu verschicken oder zum Download anzubieten.

Ein Bild, das viele Farbverläufe besitzt, also z.B. ein klassisches Foto, sollte mit JPG komprimiert werden. Dieses Verfahren wurde entwickelt, um Bilder mit Farbverläufen besonders gut zu komprimieren, da diese optisch leicht mit mathematischen Verläufen angenähert werden können.

Ein Bild das harte Farbkanten besitzt, wie zum Beispiel Logos oder Schriftzüge auf einfarbigem Grund, sollte mit GIF oder PNG komprimiert werden, da bei diesen Verfahren die Reduktion nicht durch Zusammenfassung von Bildinhalten erfolgt. Unscharfe Kanten, wie bei JPG üblich, gibt es bei PNG und GIF nicht. Dafür ist speziell GIF auf 256 Farben beschränkt, weshalb man heute in der Regel PNG verwendet.

# 7 Vektorgrafik

## 070001

Vektorgrafiken sind Computergrafiken, die aus geometrischen Formen wie Linien, Kreisen, Polygonen oder allgemeinen Kurven beziehungsweise Splines, zusammengesetzt sind. Sie werden unter anderem bei der Erstellung von einfachen Formen, wie Logos und Schriften verwendet. Welche Eigenschaften Vektorgrafiken besonders machen wirst Du in den folgenden Kapiteln erfahren.

## 070101

Die Bezeichnung „Vektor“ stammt aus dem Lateinischen und bedeutet Fahrer oder Richtung. In der Computergrafik beschreibt ein Vektor eine Länge und Richtung im 2D- oder 3D-Raum. Übertragen auf den Alltag würde ein Vektor einer Wegbeschreibung gleichkommen. In Bezug auf den Ursprung eines Koordinatensystems definiert ein Vektor einen Punkt im Raum.

Verbindet man mehrere Punkte ergeben sich einfache, geometrische Formen. Eine mithilfe von Vektoren beschriebene Form wird Vektorobjekt genannt. Außer den Vertices besitzt das Objekt weitere Eigenschaften wie die Linienstärke und Füllfarbe.

## 070103

Verändere durch die Regler die Eigenschaften der Vektorobjekte.

## 070201

Rastergrafiken bestehen aus einer rasterförmigen Anordnung von Pixel, denen jeweils eine Farbe zugeordnet wird. Sie eignen sich zur Darstellung komplexer Bilder, wie

etwa Fotos. Für einfachere Bilder eignen sich Vektorgrafiken, da sie mathematisch berechnete Darstellungen und dadurch beliebig skalierbar sind. Sie können immer optimal in der Auflösung des Ausgabemediums dargestellt werden.

## 070202

Nutze den Schieberegler, um die Grafiken zu vergrößern und mache dir die Unterschiede klar.

## 070301

Eine Bézierkurve ist eine parametrisch modellierte Kurve, die in der Computergrafik wegen ihrer verhältnismäßig leichten, mathematischen, Handhabbarkeit angewendet wird.

Vereinfacht besteht sie aus zwei Endpunkten und einer Anzahl von Kontrollpunkten. Mehrere hintereinander gesetzte Kurven nennt man in der Mathematik Splines.

Es gibt unterschiedliche Grade von Kurven. Eine Kurve ersten Grades ist nicht gekrümmt. Eine Kurve zweiten Grades wird aus einem einzigen Kontrollpunkt gebildet, eine Kurve dritten Grades aus zwei.

## 070303

Wähle den Grad der Kurve aus, verschiebe danach die Kontroll- und Endpunkte und steuere somit die Kurve.

## 070401

Skalierungen, Rotationen und Scherungen können bei Vektorgrafiken verlustfrei durchgeführt werden, da sie durch mathematische Anweisungen beschrieben werden und somit leicht transformiert werden können.



## 070501

Die Reihenfolge von Transformationen ist bei Verkettung von Transformationen relevant. Im folgenden Beispiel wird die Skalierung in Richtung der Welt-Y-Achse geändert. Sie hat die gleiche Richtung der Objekt-Y-Achse.

Nachdem das Objekt rotiert wird, wird es wieder an der Y-Achse skaliert, diesmal entspricht dies einer Scherung.

## 070503

Wähle eine Transformationsart aus und durchführe diese durch den Schieberegler. Durch die Buttons kannst Du auch die Reihenfolge ändern.

## 070601

Beim Rastern einer Vektorgrafik entsteht ein Treppeneffekt, der sogenannte Aliaseffekt. Mit Antialiasing wird die Abstufung der Pixel verbessert, indem angrenzende Pixel mit Zwischenfarbtönen besetzt werden. Das Bild wird dadurch jedoch unschärfer.

## 070602

Verschiebe die Ankerpunkte und schalte das Antialiasing an und aus.

## 070701

Manchmal ist es sinnvoll, eine Rastergrafik in eine Vektorgrafik umzuwandeln. Oft wird für die Vektorisierung von Rastergrafiken das Tracen, also das computergesteuerte Vektorisieren eines Bildes, verwendet. Leider ist das Ergebnis oft nicht zufriedenstellend und benötigt eine Anpassung des Ergebnisses.

Deswegen Bevorzugen die meisten Grafiker die Rastergrafiken direkt per Hand nachzuzeichnen.

## 070702

Verändere die Qualität des Tracens in dem Du durch die Schieberegler die Genauigkeit und den Farbschwellenwert veränderst.

# 8 Szenenaufbau

## 0801 Einleitung

Eine Szene beschreibt in der 3D-Computergrafik eine Art virtuellen Raum, in dem sich Objekte, Lichtquellen und eine Kamera befinden können. Sich in einer Szene befindliche Objekte, oder auch Meshes, sind aufgebaut aus Polygonen oder Polygonnetzen, also Netzen aus geschlossenen Vielecken.

## 0802 Aufbau von Meshes

In einer Szene können verschiedenförmige Meshes erzeugt werden. In den verschiedenen 3D-Grafik-Modellierungs- und Animationsprogrammen werden einfache Meshes, wie zum Beispiel Würfel, Zylinder und Kugeln, bereits fertig gebaut zur Nutzung angeboten. Kompliziertere Meshes können selbst modelliert werden. Sie bestehen aus Eckpunkten, sogenannten Vertices, Kanten Edges und Flächen, sogenannten Faces.

## 0803 Aufbau von Meshes

Bewege die Vertices, Edges und Faces, um ihr Zusammenspiel zu verstehen. Du kannst auch neue Vertices erstellen und verbinden.

## 0804 Transformationen

Die Transformation eines Objektes beschreibt grundsätzlich seine Position, Rotation und Größe. Diese kann durch Translation, Rotation und Skalierung verändert werden. Diese Veränderungen werden ebenfalls als Transformationen bezeichnet.

## 0805 Transformationen

Verschiebe, skaliere und drehe den Würfel.

## 0806 Hierarchie

Objekte können miteinander hierarchisch verknüpft werden, um Eigenschaften, wie zum Beispiel ihre Transformationen, miteinander zu verketteten. Dadurch beeinflusst das Objekt mit höherer Hierarchie, das so genannte Elternobjekt, alle darunter gestellten Objekte, die Kinder- und Kindeskindern. Diese Hierarchie kann in einem Szenengraph dargestellt werden. Jedes Objekt stellt einen Knoten dar. Die Verbindungen zwischen den Knoten werden als Äste bezeichnet. Oftmals wird daher auch der Begriff des Szenenbaums verwendet. Der „Pivot Point“ ist, ähnlich einem Gelenk, der Punkt um den das Objekt rotiert, das Rotationszentrum. Er ist standardmäßig im Koordinatenursprung des Objektes positioniert. Wenn man Objekte in Beziehung zueinander anordnen und bewegen will, muss man den Pivot Point sinnvoll an den Punkt setzen, um den es rotieren oder skalieren soll.

## 0807 Hierarchie

Setze die Pivotpunkte entweder in die Objektsprünge, oder an die gewünschten Rotationszentren. Spiele dann die Animation ab.

## 0808 Farbe und Licht

Farbe in der 3D-Computergrafik ergibt sich aus vielen Parametern. Sie ist abhängig von der Position des Objektes, der Ausrichtung der Fläche im Raum, also ihrer Flächennormalen, und den Materialeigenschaften, zum Beispiel der Textur. Außerdem wird sie bestimmt durch Farbe, Intensität und Richtung des Lichtes, sowie der Position und Rotation der Kamera.

## 0809 Lichtquellen-Typen

Es werden fünf grundlegende Lichtquellen-Typen unterschieden. Umgebungslicht, Richtungslicht, Punktlicht, Spotlicht und Flächenlichtquellen. Das Umgebungslicht, im Englischen Ambient-Light genannt, ist durch eine Intensität, aber keine bestimmte Richtung definiert. Richtungslicht bezeichnet man auch als „Parallel Light“, darunter versteht man direktionales Licht, das von einem sehr weit oder sogar unendlich weit entfernten Punkt, ähnlich einer Sonne, ausgestrahlt wird. Es besitzt überall in der Szene die gleiche Richtung. Unter Punktlicht versteht man Licht, das von einem Punkt ausgeht und sich bis auf eine bestimmte Distanz in alle Richtungen, und somit radial, im Raum ausbreitet. Das Spotlicht strahlt in einem Kegel von der Kegelspitze aus. Es besitzt eine Position, wie auch ein Punktlicht, strahlt aber nur in einem bestimmten Öffnungswinkel aus. Auf eine gewisse Distanz findet eine Attenuation, eine Dämpfung, statt, die einen Intensitätsabfall mit sich bringt. Flächenlichtquellen erzeugen weiches Licht und bestehen aus Ebenen oder Körpern, die mehrere Lichtquellen enthalten.

## 0810 Lichtquellen-Typen

Wähle verschiedene Lichtquellen.

## 0811 Kamera

Die Kamera projiziert eine dreidimensionale Szene auf ein zweidimensionales Bild. Die 3D-Szene, deren Koordinaten als dreidimensionale Vektoren im Raum vorliegen, wird zu einem Bild mit diskreten, zweidimensionalen Pixeln umgerechnet. Es gibt zwei Projektionsarten, um dies zu realisieren, die Perspektivische Projektion und die Orthografische, bzw. Parallelprojektion. Dafür sind drei Begriffe wichtig. Der Viewpoint beschreibt, wo sich die Kamera im Raum befindet. Das „Center of Interest“ ist der Punkt, auf den die Kamera gerichtet ist. Als „Up Vector“ bezeichnet man den Vektor, der die Oberseite der Kamera markiert und somit ihren Drehwinkel erkennen lässt.

## 0812 Perspektivische Projektion

Die perspektivische Projektion ist der Wahrnehmung des menschlichen Auges sehr nahe. Objekte, die nahe am Betrachter sind werden größer dargestellt, als weit

entfernte. Um scharf sehen zu können, müssen sich die Lichtstrahlen, die von einem Objekt reflektiert werden auf der Netzhaut des Auges in einem Fluchtpunkt schneiden. In der 3D-Computergrafik dient die Kamera als Auge des Betrachters, in ihr schneiden sich die Strahlen. Ihren Öffnungswinkel kann man frei wählen, was verschiedene Brennweiten simuliert. Zwischen Kamera und Szene befindet sich eine Bildebene, auf die die dreidimensionale Szene projiziert wird, ähnlich einer Leinwand. Der Bereich, der von der Kamera erfasst und gerendert wird, wird zum Betrachter hin von der Near-Plane und nach hinten von der Far-Plane begrenzt. Der erfasste Bereich zwischen Near- und Far-Plane wird als Frustum bezeichnet, er bildet die Form einer Pyramide mit abgeschnittener Spitze. Objekte außerhalb werden nicht berücksichtigt. Das Zuschneiden auf den Bereich des Frustums wird im Englischen als Clipping bezeichnet.

## 0813 Parallelprojektion

Die Parallelprojektion erzeugt ein weniger realistisches Bild. Bei ihr werden Objekte durch parallele Strahlen auf die Bildebene projiziert. Das Frustum hat daher die Form eines Quaders. Objekte erscheinen unabhängig von der Entfernung zum Betrachter gleich groß. Diese Art der Projektion wird vor allem für technische Zeichnungen und Video Games verwendet, dort jedoch oft fälschlicherweise als Isometrie bezeichnet.

## 0814 Projektion

Du siehst dasselbe Bild einmal als Perspektivische Projektion links, und als Parallelprojektion rechts. Rotiere nun die Kamera, um die Auswirkungen der verschiedenen Projektionsarten zu verstehen.

## 9 Rendering

Wird nachgeliefert.

# 10 Shader

## 1001 (A) Einleitung

Shader sind spezielle Programme, welche in Computerspielen, in der Postproduktion von Videoinhalten und bei Computer Generated Imagery, kurz CGI, zum Einsatz kommen. Shader waren ursprünglich, wie der Name schon sagt, für das Schattieren bzw. Erzeugen von verschiedenen Stufen von Licht, Dunkelheit und Farben in der Computergrafik zuständig. Beispiele hierfür sind Flat-, Gouraud - und Phong-Shading, welche fest im Grafikchip verbaut sind. Heutzutage können Shader frei programmiert werden und übernehmen auch Aufgaben, die nichts mit dem ursprünglichen Schattieren zu tun haben, wie z.B. das Erzeugen neuer Geometrien.

Die Verarbeitung von Shadern findet in den sogenannten Shadereinheiten statt. Diese befinden sich in Grafikchips von Grafikkarten, welche ein wichtiger Bestandteil in Computern, Spielekonsolen, Smartphones und anderen vergleichbaren Geräten sind. In Grafikkarten gibt es eine sogenannte Grafikpipeline, welche die Reihenfolge der auszuführenden Shader festlegt. Zur Nutzung von Shadern ist noch eine programmierbare Schnittstelle nötig, wie z.B. DirectX oder OpenGL.

## 1002 (A) Flat-Shading

Flat-Shading, manchmal auch Constant-Shading genannt, ist ein sehr einfaches Schattierungsverfahren, da pro Polygon nur eine Farbe möglich ist. Wenn man z.B. ein Dreieck als Polygon nimmt, wird der Farbwert unter anderem aus dem Flächen-Normalenvektor, der Flächenfarbe und der Lichtintensität berechnet. Danach werden alle Pixel des Polygons auf diese Farbe gesetzt. Als Ergebnis erhält man dann, besonders bei gekrümmten Oberflächen, eine Facettenartige Darstellung. Das ist auch der größte Nachteil des Flat-Shading, weil es dadurch zum sogenannten Mach-Band-Effekt kommt, wodurch die entstandenen Kanten besonders stark vom menschlichen Auge wahrgenommen werden.

Um die facettenartige Darstellung zu vermindern, muss die Anzahl der Polygone erhöht werden, wodurch der Rechenaufwand aber steigt. Aufgrund der genannten

Nachteile kommt das Flat-Shading meistens bei Objekten mit ebenen Flächen wie z.B. Quader, Würfel oder Pyramiden zum Einsatz.

## 1003 (A) Gouraud-Shading

Das Gouraud-Shading wurde nach seinem Entwickler Henri Gouraud benannt, der es erstmals 1971 vorstellte.

Das Besondere am Gouraud-Shading ist, dass im Gegensatz zum Flat-Shading, Farbverläufe dargestellt werden können. Dafür werden die Normalenvektoren an den Vertices berechnet. Diese erhält man durch den Mittelwert der Normalen aller angrenzenden Polygone. Danach werden durch Interpolation die Farbwerte an den Vertices berechnet.

Durch dieses Verfahren erscheinen die Kanten der Polygone weniger hart, wodurch Objekte besser rund oder gekrümmt dargestellt werden können. Ein Nachteil ist die bei manchen Objekten fehlerhafte Darstellung von Glanzlichtern und das Vorkommen von Sprüngen im Farbverlauf.

## 1004 (A) Phong-Shading

Das Phong-Shading, benannt nach seinem Entwickler Bui Tuong Phong, wurde erstmals 1975 vorgestellt.

Bei diesem Verfahren werden zu Beginn, wie beim Gouraud-Shading auch, die Normalen an den Vertices eines Polygons berechnet. Daraufhin wird beim Einfärben der Pixel eine Normale zwischen den Eckpunktnormalen interpoliert, mit der die Farbe entsprechend der Beleuchtung ausgewertet wird. Dadurch erhält jedes Pixel die entsprechende Beleuchtung und somit auch die korrekte Farbe.

Das Phong Shading liefert aufgrund von glänzenden Oberflächen und Specular highlights, welche hier für den hellen Fleck sorgen, ein realistischeres Ergebnis als das Gouraud Shading, ist aber auch deutlich rechenintensiver.



## 1005 (I) Vergleich zwischen Flat-, Gouraud - und Phong-Shading

Hier kannst du die verschiedenen Shader anhand eines Objekts vergleichen, indem du unten einen Shader auswählst.

## 1006 (A) Vertex Shader

Vertex-Shader sind Programme, welche im Verlauf der Grafikpipeline in den Shadereinheiten einer Grafikkarte ausgeführt werden. Diese verarbeiten die sogenannten Vertices, bei denen es sich um Eckpunkte eines 3D-Modells handelt.

Beim Vertex-Shader werden die Koordinaten der einzelnen Vertices im dreidimensionalen Raum für die zweidimensionale Darstellung transformiert. Dadurch lässt sich die Geometrie und somit die Form von Objekten beeinflussen, was sich wiederum auch auf die Beleuchtung auswirken kann. Da der Shader aber pro Vertex aufgerufen wird, kann er keine neuen Punkte zum 3D-Modell hinzufügen.

## 1007 (I) Vertex Shader

Hier ist ein Vertex Shader aktiv, welcher durch verschieben des Reglers Einfluss auf die Geometrie der Wellen nimmt.

## 1008 (A) Geometry-Shader

Der Geometry-Shader wird in der Grafikpipeline nach dem Vertex Shader aufgerufen, um neue primitive Geometrien aus bereits vorhandenen Punkten, Linien und Dreiecken zu erzeugen und diese erneut in die Grafikpipeline einzufügen.

Beispiele für die Anwendung des Geometry-Shader sind die Erzeugung von Schattenvolumen oder die Erzeugung von Fell- oder Haargeometrie.

Um Haare zu erzeugen, erhält der Geometry-Shader die benötigten Vertices vom Vertex-Shader als Input. Diese Vertices werden dann durch mehrere Kopien ersetzt, wodurch eine Haar-Struktur entsteht. Nach der Verarbeitung werden die fertigen Fragmente an den Pixel-Shader weitergegeben.

## 1009 (A) Pixel- / Fragment-Shader

Pixel-Shader, auch Fragment-Shader genannt, werden ebenfalls in den Shadereinheiten einer Grafikkarte ausgeführt. Der Shader ist für die Farbberchnung der einzelnen Pixel, auch Fragmente genannt, zuständig und folgt in der Grafikpipeline auf den Geometry-Shader.

Pixel-Shader werden genutzt, um eine realistische Darstellung von Oberflächen- und Materialeigenschaften zu erreichen oder eine Texturdarstellung zu bewerkstelligen bzw. zu verändern. Ein Pixel kann dabei aus mehreren Fragmenten bestehen, was zum Beispiel bei Transparenz der Fall ist, wenn ein Objekt hinter einer Scheibe steht. Die richtige Farbe wird über eine Beleuchtungsberechnung zugewiesen.

## 1010 (A) Toon-/Cel-Shading

Beim Toon-Shading, auch Cel Shading genannt, handelt es sich um eine Technik zum nicht-fotorealistischen Rendern von 3D-Computergrafiken. Als Ergebnis erhält man bei diesem Verfahren eine Optik, die der von gezeichneten Comics oder Zeichentrickfilmen entspricht.

Um diesen Effekt zu erhalten, wird auf weiche Verläufe verzichtet und es kommen nur drei oder vier Helligkeitsstufen zum Einsatz. In der Regel sind das weiß, hellgrau und dunkelgrau. Außerdem verzichtet man meistens auf eine Textur und verwendet nur einzelne Farbtöne.

Um die schwarzen Linien zu erhalten, welche die innere und äußere Kontur des Objekts darstellen, invertiert man Polygone die aufgrund der Perspektive nicht sichtbar wären. Dafür wird das Backface Culling, welches nicht sichtbare Polygone aufgrund der Performanceverbesserung entfernt, rückgängig gemacht. Dies wird

teilweise mehrmals mit leichten Variationen durchgeführt, um eine bessere Kontur zu erhalten. Die neu hinzugefügten Polygone werden zum Schluss schwarz gefärbt.

## 1011 (A) Programmierung eines Shaders

Der Code eines Shaders erinnert stark an die Programmiersprache C, weshalb es viele Parallelen zu dieser und auch anderen Programmiersprachen gibt. Dieses einfache Beispiel wurde in der OpenGL Shading Language, kurz GLSL, geschrieben.

Shader haben in der Regel folgenden Aufbau. Als erstes wird die Version deklariert, gefolgt von den Input- und Output-Variablen. In der Main-Funktion werden die Inputs verarbeitet und das Ergebnis an die Output-Variablen weitergegeben.

In diesem Beispiel wird ein Vertex Shader verarbeitet. Als Input wird eine Vertex-Position benutzt, welche als Output eine Farbe erhalten soll. In der Main-Funktion wird die Position von Vektor 3 an Vektor 4 übergeben und danach die Farbe zugewiesen.

Der Output vom Vertex Shader wird dann als Input beim Fragment Shader genutzt. Dort wird die Farbe für ein Fragment übernommen.

Als Ergebnis erhält man ein Fragment mit der zuvor einprogrammierten Farbe.

# 11 Modellierung

## 1101 Einführung

Die 3D-Modellierung ist ein Verfahren, bei dem mithilfe von Software dreidimensionale Meshes mathematisch dargestellt werden.

Solche dreidimensionalen Modelle werden in einer Vielzahl von Anwendungsgebieten eingesetzt.

In Film, Games, Produktentwicklung, Wissenschaft und Medizin kommen 3D-Modelle zum Visualisieren, Simulieren und Rendern grafischer Entwürfe weitläufig zum Einsatz.

## 1102 Grundformen der 3D-Modellierung

In der 3D-Computergrafik arbeitet man beim Modellieren eines Objektes oft mit Grundformen.

Dies sind z.B. die Kugel, Kubus und Zylinder.

Durch Hinzufügen, Entfernen oder Verschieben einzelner Vertices, Edges oder Faces können die Körper individuell verändert werden.

## 110201 Grundformen der 3D-Modellierung – Interaktion

Wähle einen der Körper aus und schau dir diesen genauer an.

## 1103 High- und Low-Poly-Modelle

Low- und High-Poly sind Begriffe aus der 3D-Modellierung. Dieser Begriff bezieht sich dabei auf die Anzahl der verwendeten Polygone aus denen sich das Mesh zusammensetzt.

Ein Low-Poly-Modell besteht aus wenigen Polygonen, wogegen ein High-Poly-Modell demnach aus sehr vielen Polygonen besteht. Dies dient dazu, um auch die kleinsten Details eines Objektes detailgetreu darzustellen.

Low-Poly findet bei Echtzeitrendering z. B. in der Augmented bzw. Virtual Reality und auch in 3D-Spielen Anwendung. Mittlerweile gehören die Low-Poly Modelle auch zu einer visuellen Stilrichtung.

High-Poly verwendet man dagegen bei fotorealistischen 3D-Renderings oder z. B. bei 3D-Animationen mit einem Zoom-in-Effekt, also einem Detailausschnitt eines Renderings.

## 110301 High- und Low-Poly-Modelle – Interaktion

Stelle über den Schieberegler die Polygonanzahl des 3D-Objekts ein und schau dir an, wie sich das Objekt dabei verändert.

## 1104 Extrudieren

Das Wort Extrudieren kommt aus dem Lateinischen Extrudo und bedeutet hinausstoßen, -treiben, -drängen.

Beim Extrudieren werden zusätzliche Vertices, Edges und Faces aus den Vertices, Edges oder Faces eines bestehenden Objekts gebildet.

Die extrudierten Vertices, Edges oder Faces bleiben dabei weiterhin mit dem Ursprungskörper verbunden.

Beim Extrudieren handelt es sich um eines der wichtigsten Werkzeuge bei der Modellierung. Aus einer Grundform kann auf diese Art und Weise ein komplexer Körper erstellt werden.

## 1105 Lathe/Rotationskörper

Um Körper wie z. B. Vasen, Tassen, Kelche oder Schachfiguren zu modellieren, besteht die Möglichkeit dies mittels spezieller Extrusion durchzuführen.

Dabei wird ein Kurvenzug anhand eines Pfades an der Drehachse des Pfades extrudiert. Das ist zu vergleichen wie bei einer Rotation an einer Drehbank oder auch Lathe genannt.

In der 3D-Modellierung wird hierfür zunächst der Querschnitt, des zu modellierenden Körpers erzeugt.

Dieser wird dann um eine Rotationsachse um 360° gedreht.

Dabei erstellt das Programm automatisch die benötigten Splines. Diese sorgen dafür, dass die Polygone bei gleichbleibender Höhe dupliziert und gedreht werden.

## 1106 Mirror-Modifier/Operation

Bei spiegelsymmetrischen Objekten ist es vorteilhaft, nur eine Seite des 3D-Objektes zu erstellen.

Daraufhin benutzt man den Mirror-Modifier um es auf die andere Seite zu spiegeln.

Der Mirror-Modifier kann grundsätzlich auf alle Achsen, also x-, y- und z-Achse, angewandt werden.

## 110601 Mirror-Modifier/Operation – Interaktion

Wähle eine der Achsen aus und spiegle das Objekt an dieser.

## 1107 Subdivision

Subdivision dient bei der 3D-Modellierung dazu, einen Körper mit wenig Polygonen und harten Kanten, in einen Körper mit vielen Polygonen und weichen Kanten umzuwandeln. Dabei wird auf einem einfachen Körper gearbeitet das nachträglich noch verändert werden kann. Das Problem bei der Computergrafik ist das viele glatte Flächen, mithilfe von Subdivision, an gerundete Flächen angenähert werden müssen.

Dabei erzeugt jeder Rekursionsschritt bei viereckigen Flächen  $4 \text{ hoch } n$  neue Flächen. Bei dreieckigen Flächen erzeugt jeder Rekursionsschritt  $3 \text{ mal } 4 \text{ hoch } (n-1)$  neue Flächen.

Beim ersten Rekursionsschritt erhält ein Kubus auf diese Weise vier neuen Flächen. Mit dem zweiten Rekursionsschritt sind es 16 neue und mit dem dritten 64 neue Flächen.

Beim Aktivieren des Subdivision, werden die geraden Linien des Objektes den gekrümmten angepasst.

In der Computergrafik wird der Catmull Clark sehr häufig eingesetzt und wurde mit einem Oscar ausgezeichnet.

## 110701 Subdivision Surface – Interaktion

Stelle mit dem Schieberegler die Anzahl der Rekursionsschritte ein und beobachte die Auswirkungen auf das Objekt.

## 1108 Sculpting

Beim Sculpting verändert der Anwender interaktiv die Form eines Modells. Dabei arbeiten Sculpting Tools auf verschiedenen Auflösungsstufen.

Um auf einen Körper Sculpting anwenden zu können muss dieser viele Polygone besitzen, die mithilfe von Subdivision erzeugt wurden.

Beim Sculpting werden nun die Eckpunkte in einem Mesh durch Malen verändert.

Dabei können die Eckpunkte eingedrückt, ausgestülpt, zusammengekniffen, eingekerbt, geglättet oder beschnitten werden.

Diese Methode eignet sich um natürliche Muster wie zum Beispiel Schuppen oder Objekte zu modellieren.

## 1109 Boolean Modifier/Operation

Ein Boolean Modifier hilft dem Anwender ein aus verschiedenen Meshes zusammengesetztes neues Mesh zu erzeugen.

Hierbei gibt es drei Einstellungen für den Modifier: zuerst den Intersect, welcher die Schnittmenge zweier Objekte bildet.

Des Weiteren gibt es die Einstellung Union, welche eine Vereinigung zweier Objekte bildet.

Und als letztes gibt es Difference, welche dazu dient ein Objekt von einem anderen abzuziehen.

### 110901 Boolean Modifier/Operation – Interaktion

Wähle eine der Einstellungen aus und sieh, wie sich dadurch das Endresultat ändert.



## 12 Texturen

### 1201 (A) Einleitung

Eine Textur dient im Bereich der Computergrafik dazu Oberflächen von 3D-Objekten realistischer und detailreicher darzustellen.

Die Textur besteht meistens aus einem zweidimensionalen Bild, welches dem 3D-Objekt eine Struktur verleiht. Wie die Textur auf das jeweilige 3D-Objekt projiziert wird, kann auf verschiedene Arten und Weisen geschehen. Diese Verfahren nennt man Mapping-Verfahren.

Der Begriff Mapping bedeutet Zuordnung oder Abbildung.

### 1202 (I) Einleitung

Wähle den Kubus oder die Kugel und anschließend eine Textur aus der vordefinierten Liste aus und lasse dir den Vergleich von dem 3D-Objekt ohne und mit Textur anzeigen. Betrachte das 3D-Objekt dabei von allen Seiten.

### 1203 (A) Texturkoordinaten

Nun stellt sich die Frage, wie die Textur auf das Mesh projiziert wird. Zuerst wird die zu texturierende Fläche im 3D-Raum definiert. Anschließend ordnet der Benutzer die definierten 3D-Koordinaten mit Hilfe geeigneter Werkzeuge den Vertices des Polygons zu.

Die Textur besteht aus Pixeln, welche auch Texel genannt werden.

Texturkoordinaten werden in einem kartesischen Koordinatensystem dargestellt. Die Achsen werden in der Regel mit u und v beschriftet um sie von x und y unterscheiden zu können.

Der Wertebereich ist dabei jeweils von 0 bis 1.

## 1204 (A) UV-Mapping

Beim UV-Mapping bildet man die Textur mit einer einfachen Abwicklung in der geeigneten Software von Hand. Der Prozess wird auch als Unwrapping bezeichnet.

Bildlich kann man sich die Abwicklung so vorstellen, dass das 3D-Objekt an bestimmten Stellen "aufgeschnitten" und anschließend "abgewickelt" wird, wie in diesem Fall die Weltkugel.

Versuche den Prozess der Abwicklung nachzuvollziehen.

## 1205 (I) UV-Mapping

Wähle den Pinsel und suche dir eine beliebige Farbe aus. Nun kannst du die von Hand abgewickelte Textur des Roboters oder den Roboter direkt bemalen.

## 1206 (A) Mip-Mapping

Das Mip-Mapping ist eine Anti-Aliasing-Methode. In der Computergrafik tritt der Alias-Effekt beim Betrachten von einer Textur mit hoher Auflösung aus weiter Entfernung auf, wodurch Muster und Verzerrungen entstehen, die im Originalbild nicht enthalten sind.

Beim Mip-Mapping werden von einer Textur mehrere vorberechnete skalierte Texturen mit sinkender Auflösung berechnet. Dabei wird in jedem Schritt die Kantenlänge des Originals halbiert.

Ist das texturierte Polygon nahe beim Betrachter oder groß skaliert, kommt eine große

Textur mit hoher Auflösung zum Einsatz. Ist es jedoch weiter entfernt oder klein skaliert, wird eine kleinere Textur mit geringerer Auflösung verwendet.

Der große Vorteil des Mip-Mappings besteht darin, dass die verschiedenen Maps zum Zeitpunkt des Renderns bereits vorberechnet sind.

## 1207 (A) Bump-Mapping

Eine normale 2D-Textur verleiht einem Objekt keine Oberflächeneigenschaften wie Höhe und Tiefe. Das Bump-Mapping dient dazu, einem 3D-Objekt diese Eigenschaften zu verleihen, ohne dabei die Geometrie des 3D-Objektes zu beeinflussen.

Beim Bump-Mapping wird auf Pixelebene lediglich die Normale der Oberfläche verändert. Es wird mit Hilfe von Schattierung und Reflexion eine Illusion von Tiefe auf dem 3D-Objekt erzeugt. Das Bump-Mapping verwendet in der Regel verschiedene Graustufen und beschreibt damit Höhenunterschiede auf dem Objekt. Somit ist das Bump-Mapping ein reiner Beleuchtungseffekt.

Der Vorteil des Bump-Mapping besteht darin, dass bei diesem Mapping-Verfahren der Speicherplatz und die Rendering-Zeit ziemlich gering bleiben.

## 1208 (A) Normal-Mapping

Das Normal-Mapping ist eine Abwandlung des Bump-Mapping-Verfahrens. Es zielt ebenfalls darauf ab, einen größeren Detailreichtum eines 3D-Objektes in Form von Höhe und Tiefe zu erzeugen.

Beim Normal-Mapping wird die Anzahl der Polygone nicht erhöht und die Oberfläche nur optisch verformt. Alle Informationen über die Ausrichtung der Normalen, die für die Beleuchtung wichtig sind, werden in Form von RGB-Werten von einem hoch aufgelösten auf ein niedrig aufgelöstes Objekt übertragen. Die optischen Details gehen dabei nicht verloren.

Der Nachteil dieses Mapping-Verfahrens ist, dass es stark vom Betrachtungswinkel

abhängig ist.

## 1209 (A) Displacement-Mapping

Das Displacement-Mapping löst dasselbe Problem wie das Bump- und das Normal-Mapping, es verleiht dem 3D-Objekt ebenfalls Vertiefungen und Erhöhungen.

Der Unterschied zwischen den Mapping-Verfahren ist, dass beim Displacement Mapping die Geometrie des 3D-Objektes verändert wird. Das Displacement-Mapping findet also auf polygonaler Ebene statt.

Durch "Verschieben" beziehungsweise "Verdrängen" des Materials werden dem 3D-Objekt die gewünschten Oberflächeneigenschaften verliehen.

## 1210 (I) Vergleich

Hier wird der Unterschied zwischen dem Normal-Mapping und dem Displacement-Mapping dargestellt. Betrachte die texturierten 3D-Objekte von allen Seiten.

## 1211 (A) Environment-Mapping

Mit dem Environment-Mapping werden in der Computergrafik Spiegelungen, durchlässige Objekte und Beleuchtungs-Simulationen dargestellt.

Dabei wird die Umgebung, bestehend aus Objekten und Lichtquellen, in der sich das Objekt befinden soll, als eine Umgebungs-Textur gespeichert und auf das 3D-Objekt projiziert. Wie dieser Vorgang funktioniert erfährst du im nächsten Kapitel. Das 3D-Objekt muss im Verhältnis zu seiner Umgebung eher klein sein, damit die Spiegelung realistisch wirkt.

Beim Environment-Mapping wird in zwei Arten unterschieden: das sphärische und das kubische Environment-Mapping. In Folge wird nur Letzteres beschrieben.

## 1212 (A) Kubisches Environment-Mapping

Bei dem kubischen Environment-Mapping wird die Umgebung auf einem Kubus abgebildet.

Es werden sechs 2D-Umgebungs-Texturen verwendet, die zusammen die Fläche eines Kubus bilden.

Die sechs einzelnen Umgebungs-Texturen entstehen durch das Fotografieren oder Rendern aus Position des Objektmittelpunkts in einem Winkel von 90 Grad. Dabei müssen die sechs Würfelflächen randlos abgedeckt werden.

## 13 Animation