

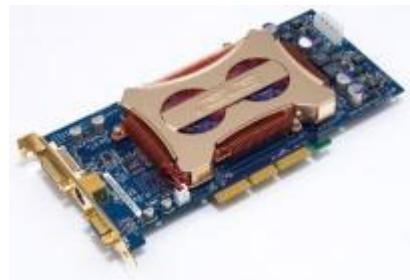
Teil V Grafikkarten

1.	Allgemeines	2
2.	Historie	2
3.	Aufbau und Funktionsweise	4
3.1.	Hardwareschnittstellen zum System (PC).....	4
3.2.	Grafikspeicher	5
3.3.	Grafikprozessor	7
3.3.1.	DirektX und OpenGL	8
3.3.2.	Bildsynthese.....	9
3.3.3.	Texture Mapping.....	10
3.3.4.	Antialiasing.....	13
3.3.5.	Anisotropes Filtern	15
3.3.6.	Multi-GPU-Techniken	16
3.4.	Kühlösungen.....	17
3.5.	Ausgänge	17

1. Allgemeines

Eine **Grafikkarte** steuert in einem Computer die Bildschirmanzeige.

Bei Ausführung eines Programmes berechnet der Prozessor die Ausgabedaten, leitet diese an die Grafikkarte weiter und diese wandelt die Daten so **in digitale oder analoge Signale** um, dass der Monitor alles als Bild wiedergeben kann.



Grafikkarten werden entweder

- als **PC-Erweiterungskarten** (über die Bussysteme ISA, VLB, PCI, AGP oder über **PCI-Express**) mit der Hauptplatine verbunden, oder
- sind **im Chipsatz** auf der Hauptplatine enthalten (**Onboard-Variante**), oder
- sind direkt **im Prozessor** verbaut (**Accelerated Processing Unit – APU**).

2. Historie

Das Grafikkarten-Prinzip wurde in Serienprodukten erstmals beim Mikrocomputer **Apple II** verwendet, dessen auf der Hauptplatine integrierte Grafikfähigkeiten durch zusätzlich zu erwerbende Steckkarten verbessert werden konnten.

Der erste IBM-PC kam 1981 mit einer Karte auf den Markt, die lediglich die einfarbige Darstellung von Text ermöglichte (**MDA** = Monochrome Display Adapter). Die Firma **Hercules** bot 1982 eine sehr viel bessere Karte an, die Hercules Graphics Card.

Bis 1989 setzten sich die **Farb-Grafikkartentypen** als Standard durch:

- 1981 die **CGA-Karte** (4 Farben bei 320×200 Punkten; 2 weitere Farbmodi; liefert digitale Signale und braucht spezielle CGA-Monitore)
- 1984 die **EGA-Karte** (16 Farben aus einer Palette von 64 Farben und vier Bit Farbtiefe bei einer Bildauflösung von 640×350 Pixeln; liefert digitale Signale und braucht spezielle EGA-Monitore)
- 1989 die **VGA-Karte** (64 kB Speicher; 256 Farben bei 320×200 Pixeln bzw. 640×480 Punkten in 16 Farben; liefert analoge Signale für analoge Monitore → Farbenvielfalt)

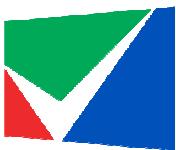


VGA und alle Nachfolger sind EGA und CGA kompatibel.

Auch heute noch ist der VGA-Modus (640×480 Punkte in 16 Farben) der „**Notfall-Modus**“ bei PCs; nur bis zu diesem Modus kann die Hardware aller heutigen PC-Grafikkarten von der Software auf einheitliche Weise angesprochen werden.

Die weiteren Bezeichnungen **SVGA**, **XGA usw.** sind keine neuen Grafikkartenstandards, und stehen heute als Kurzbezeichnungen für Bildschirmauflösungen, zB XGA: 1024×768 Punkte.

- **SVGA:** Super-VGA Karten sind erweiterte VGA-Karten mit mehr Speicher (256 kB). Sie konnten Auflösungen bis 800×600 darstellen, die Möglichkeit 256 Farben anzugeben, stieg auf 640×480 .
- **XGA:** E(x)tended-VGA Karten hatten 512 kB Speicher oder mehr. Die XGA-Auflösung schlechthin ist 1024×768 . Neben dem 256 Farben Modus wurde auch der **High Color Modus** mit 16 Bit (65.536 Farben) und der **True Color Modus** mit 24 Bit und somit 16,7 Mio. Farben eingeführt.



VE_SA

Bei den hochauflösenden Modi gab es den reinsten Wildwuchs in Sachen Standards. Das Grafikkarten-Standardisierungs-Komitee **VESA (Video Electronics Standards Association)** setzte daher die **VBE**, die **VESA BIOS Extensions** fest, mit denen man standardisiert auf diese Modi zugreifen kann.

Die aktuelle Version des Standards ist **3.0**. Er wird von den meisten modernen Grafikkarten unterstützt, allerdings heute kaum noch von der Anwendungssoftware verwendet, da heutzutage die Hardware von der Anwendung nicht mehr direkt angesprochen wird (s.u. **DirectX** und **OpenGL**). VESA wirkt heute noch bei verschiedenen **Standardisierungen**, wie zB bei den **DVI-D-** und **DVI-I**-Verbindungstechniken für digitale Videodaten, mit.

Bis etwa 1990 beschränkten sich die Grafikkarten darauf, den Inhalt des Video-RAM über einen sogenannten **RAMDAC-Baustein** in Ausgangssignale für den Monitor umzuwandeln. Der Programmierer konnte im Wesentlichen nur den **Textmodus** nutzen sowie im **Grafikmodus** einzelne Pixel auf eine bestimmte Farbe setzen.

Auf die erste Generation der Grafikkarten folgten zwei weitere:

- die sogenannten „**Windows-Beschleuniger**“ und
- die **3D-Beschleuniger**.

Ab 1990 entwickelten sich die Grafikkarten zu eigenständigen kleinen Computern mit eigener **GPU (Graphics Processing Unit)**, einer sog. *Graphics*- oder *Pixel Engine* oder dt. Grafikprozessor, bei dem man nicht nur einzelne Pixel setzen konnte, sondern dem man Befehle zum Zeichnen von Linien und Füllen von Flächen schicken konnte (Windows-Beschleuniger). Diese Funktionen beschleunigten vor allem das Verschieben der Fenster (Windows) der grafischen Benutzeroberfläche, daher der Name. Das **Konzept der Zusatzfunktionalität** wurde mit der Zeit immer weiter geführt, so wurden zB seit 1995 auch Funktionen zur Beschleunigung der Videowiedergabe (zB im AVI-Format) und Dekodierung von komprimierten Videodaten (zB MPEG) eingeführt (*Videobeschleunigung*). Diese Funktionen wurden vorher auf separaten Steckkarten angeboten.

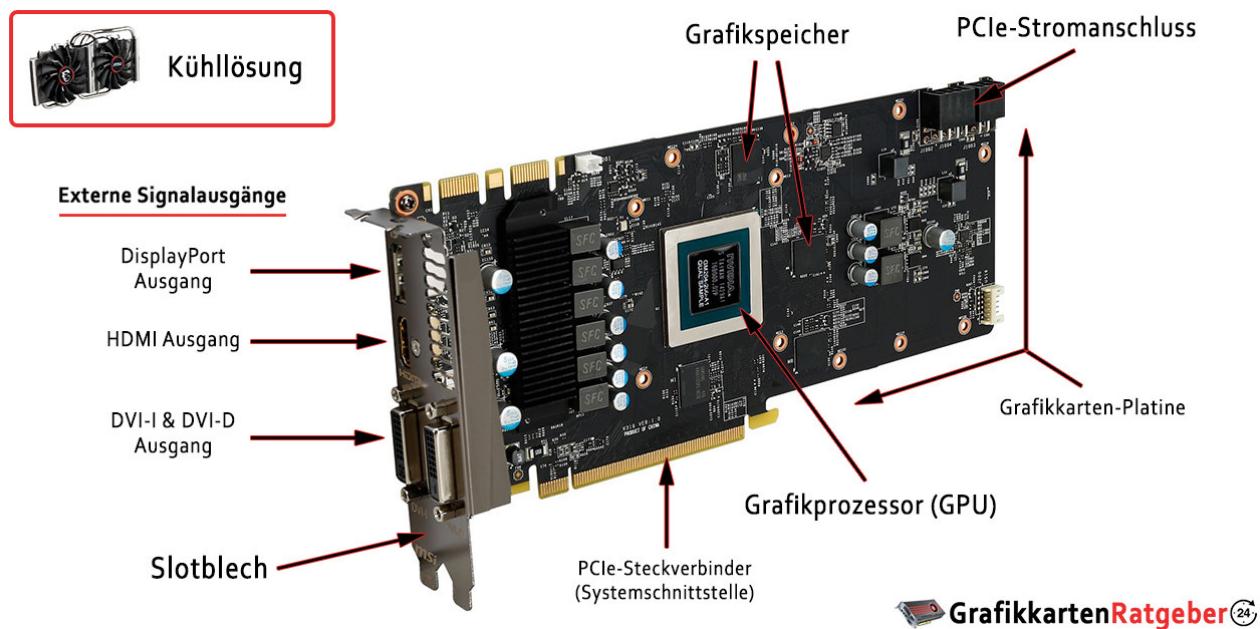
Nachdem mit Doom der große Boom der 3D-Spiele Mitte der 90er-Jahre begonnen hatte, kam bald von **3dfx** der erste brauchbare **3D-Beschleuniger**, der Voodoo Graphics-Chipsatz. Einem 3D-Beschleuniger kann ein Programm in einem dreidimensionalen Raum die **geometrischen Figuren in Form von Polygonen und die Texturen** angeben, mit denen die Flächen der Polygone gefüllt werden sollen (rendern). Diese recht simple, aber rechenintensive Aufgabe hatte in den frühen 3D-Spielen noch die CPU übernehmen müssen; nun konnte sie an die Grafikkarte delegiert werden, was zu einer massiven Leistungssteigerung von 3D-Spielen führte (bessere Bildauflösung, wesentlich realistischere Bilder).



Waren die 3D-Beschleuniger der ersten Generation noch auf eigenen Steckkarten verbaut, durch die das Grafiksignal der im System verbauten 2D-Grafikkarte durchgeschleift wurde, setzten sich bald Lösungen durch, die 2D- und 3D-Funktionalität auf einer Karte vereinten.

Heute werden mit der **Multi-GPU-Technik (SLI und Crossfire)** zwei oder mehr 3D-Grafikkarten bzw. -prozessoren parallel geschaltet, um noch mehr Grafikelemente je Zeiteinheit berechnen zu können.

3. Aufbau und Funktionsweise

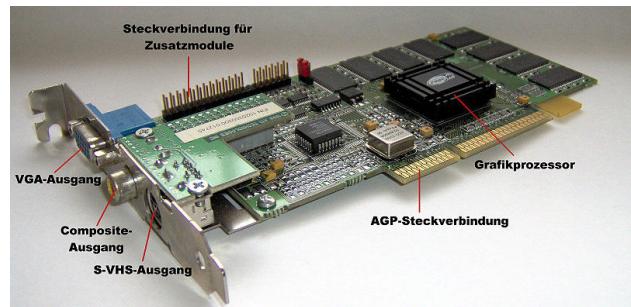


3.1. Hardwareschnittstellen zum System (PC)

Die bekanntesten PC-Hardwareschnittstellen für Grafikkarten sind **AGP** und **PCI-Express**, (früher auch PCI, ISA oder VESA Local Bus). Diese Schnittstellen sind entweder Bussysteme oder Direktverbindungen (AGP, PCI-Express), die den **Buscontroller mit der Grafikkarte verbinden**. Im Idealfall funktionieren alle konformen Grafikkarten mit allen konformen Controllern.

In den Anfangszeiten wurden Grafikkarten über den sogenannten „**internen Systembus**“ (zB ISA oder VESA Local Bus) mit Strom versorgt und betrieben. Da die Anforderungen in den letzten Jahren in Form von höheren Auflösungen und aufwendigeren Multimedia- und 3D-Darstellungen enorm gestiegen sind, wurden auch die Datenmengen immer größer, die zur Grafikkarte transportiert und dort verarbeitet werden mussten.

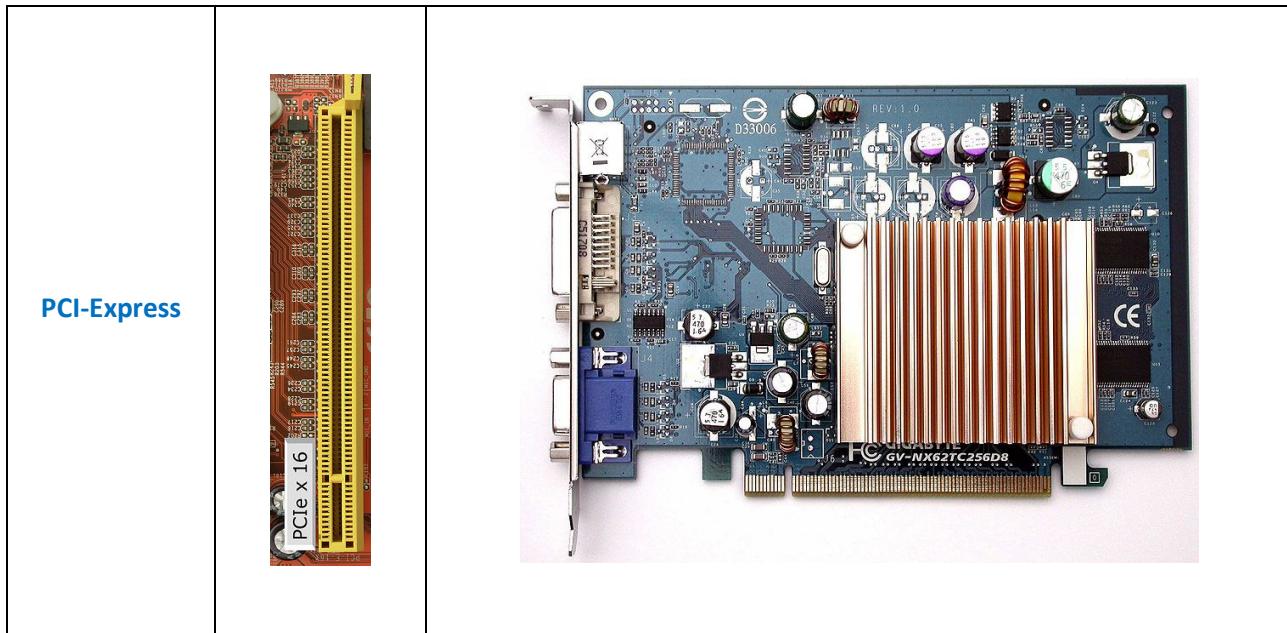
Zuerst gab es den standardmäßigen **PCI-Anschluss**. Dann folgte der modernere **Accelerated Graphics Port (AGP)-Anschluss**. Dieser Steckplatz auf dem Mainboard wurde ausschließlich für Grafikkarten entwickelt. Der AGP-Anschluss (**AGP 8x**) hat nach einigen Jahren nicht mehr ausgereicht und wurde deshalb durch den heutigen **PCI-Express-Anschluss** ersetzt, welcher viel schneller ist.



PCI-Express

PCI-Express („Peripheral Component Interconnect Express“, abgekürzt: **PCIe** oder **PCI-E**) ist ein Standard zur Punkt-zu-Punkt-Verbindung von Peripheriegeräten mit dem Chipsatz. PCIe ist der (serielle) Nachfolger von PCI, PCI-X und AGP und bietet im Vergleich zu seinen (parallel) Vorgängern eine höhere Datenübertragungsrate.

Im PC-Bereich wird meist **PCIe-x16** zur Anbindung einer Grafikkarte verwendet (PCI-Express for Graphics, PEG).



Ein PCI-Express-Steckplatz kann das daran angeschlossene Gerät mit **Strom** versorgen – bei einem PEG-(PCIe-x16)-Slot maximal **75 Watt** (AGP maximal 25 Watt). Für Hochleistungsgrafikkarten kann über Zusatzstecker zur Stromversorgung die maximale Aufnahmefähigkeit auf bis zu 300 Watt gesteigert werden.

3.2. Grafikspeicher

Der Grafikspeicher dient zur **Ablage der im Grafikprozessor (GPU) verarbeiteten Daten** sowie als Bildspeicher („**Framebuffer**“): Das sind digitale Bilder, die später auf dem Computer-Bildschirm ausgegeben werden.

Die **Größe** des Grafikspeichers wurde **früher** durch die maximale Farbtiefe und Bildauflösung bestimmt.

Bsp.: Gewünschte Auflösung: 1600×1200 mit einer Farbtiefe von 24 Bit (*True Color*)

$$\begin{aligned} \text{Benötigter Grafikspeicher: } & 1600 * 1200 = 1.920.000 \text{ Pixel insgesamt} \\ & 1.920.000 * 24 \text{ Bit} = 46.080.000 \text{ Bit} \\ & 46.080.000 \text{ Bit} / 8 = 5.760.000 \text{ Byte} = \text{ca. 5,76 Megabyte} \end{aligned}$$

Da früher Grafikkarten in der Regel mit 4 oder 8 Megabyte Grafikspeicher ausgeliefert wurden, hätte man für die gewünschte Einstellung eine Grafikkarte mit mindestens 8 Megabyte Grafikspeicher verwendet.

Heute werden ausschließlich Grafikkarten mit **sehr viel mehr Speicher** gebaut, als zur reinen Bildspeicherung notwendig wäre. Beim **Rendern dreidimensionaler Grafiken** werden hier zusätzlich zum Framebuffer die Daten der Objekte, beispielsweise Größe, Form und Position, sowie die Texturen, die auf die Oberfläche der Objekte gelegt werden, gespeichert. Besonders die immer **höher auflösenden Texturen** haben für einen starken Anstieg der Speichergröße bei aktuellen Grafikkarten gesorgt. So liegt die Speichergröße aktueller Grafikkarten bereits im Gigabytebereich (2 - 8 GB). Bei Spielegrafikkarten ist die Obergrenze Ende 2017 bei **24 GB GDDR5-Speicher**.



Welche Informationen werden im Grafikspeicher abgelegt?

- **Framebuffer**

In diesem regelmäßig durch den RAMDAC ausgelesenen Speicherbereich finden kontinuierlich die Berechnungen zur Bildsynthese statt. Die Größe des Framebuffers ist abhängig von der verwendeten Auflösung (zum Beispiel 1024×768), der benutzten Farbtiefe (zum Beispiel 16 Bit pro Pixel), dem Antialiasing-Modus und dem verwendeten Framebufferkonzept (zum Beispiel Doppelpufferung, Dreifachpufferung).

- **z-Buffer**

Hier wird für jedes Bildschirmpixel ein Wert zur Tiefeninformation gespeichert. Die Größe ist wiederum abhängig von der gewählten Genauigkeit der Werte; üblich sind 24 und 32 Bit.

- **Vertex-Shader- und Pixel-Shader Programme**

Seit DirectX 8.0 werden Spieleentwicklern maschinencodeähnliche Operationen zur Verformung von Objekten oder zur Umsetzung bestimmter grafischer Effekte (zum Beispiel Schattenwurf, Spiegelung) angeboten. Diese meist sehr kleinen Programme werden direkt im Grafikspeicher abgelegt.

- **Geometriedaten**

Mit der Einführung von DirectX wurde das Dreieck als Standardprimitiv zur 3D Darstellung festgelegt. Somit besteht nahezu jede im Grafikmodus darstellbare Szene aus mit Dreiecken zusammengesetzten Objekten (Polygone). Die Geometriedaten ordnen ua. jedem Dreieck die Eckpunkte zu. Die Größe dieses Speicherbereichs ist abhängig von der Komplexität der berechneten Szene (aktuell bis zu 500.000 Dreiecke), das heißt, je mehr Dreiecke zu verarbeiten sind, desto größer sind die Geometriedaten.

- **Texturdaten**

Alle verwendeten Texturen einer Szene werden meist aus Platzgründen komprimiert im Grafikspeicher abgelegt. Dieser Bereich nimmt den größten Anteil am Grafikspeicher ein und ist von sehr vielen Faktoren abhängig, zum Beispiel Anzahl, Größe (bis zu 8192×8192 Pixel) und Farbtiefe der verwendeten Texturen.

Bei vielen **Onboard-Lösungen** wird der Hauptspeicher des Systems als Grafikspeicher genutzt, dies wird als **Shared Memory** bezeichnet. Der Zugriff erfolgt über das jeweilige Bussystem und ist deshalb langsamer als direkt angebundener Speicher.

3.3. Grafikprozessor

Der Grafikprozessor dient zur **Berechnung der Bildschirmausgabe**, er übernimmt rechenintensive Aufgaben der 2D- und 3D-Computergrafik und entlastet dadurch den Hauptprozessor (CPU). Die Funktionen werden über **Software-Bibliotheken** wie DirectX oder OpenGL angesteuert. Die freigewordene CPU-Zeit kann somit für andere Aufgaben verwendet werden.



Mitte der 1990er Jahre kamen die ersten **3D-Beschleuniger** auf den Markt. Diese Grafikprozessoren waren in der Lage, einige Effekte und dreiecksbasierte Algorithmen (wie ua. z-Buffern, Texture Mapping) und Antialiasing selbstständig durchzuführen. Besonders dem Bereich Computerspiele verhalfen solche, zusätzlich zu installierenden Steckkarten (zB 3dfx Voodoo Graphics), zu einem Entwicklungsschub.

Heutzutage sind GPUs wegen ihrer Spezialisierung auf Grafikberechnungen **den CPUs in ihrer Rechenleistung überlegen**. Als Hinweis diene die Transistoranzahl des aktuellen Grafikprozessors von NVIDIA (**GK110 Kepler**; über 7 Milliarden Transistoren). Die Entwicklung der Integrationsdichte der Grafikprozessoren hat mit einem jährlichen Faktor von 2,4 sogar das Mooresche Gesetz übertrffen.

Eigenschaften

- Unterstützung der Grafikschnittstellen **DirectX** und **OpenGL**
- **Bildsynthese**, mit Hilfe von
 - mindestens einem *Raster Operation Processor* (ROP), auch bekannt als *Render Output Unit* oder *Raster Operations Pipeline* und
 - möglichst vielen *Stream-Prozessoren* für eine flüssige Darstellung.
- **Texture Mapping** – Musterabbildung, ua. mit Hilfe mindestens einer *Texture Mapping Unit* (TMU)
→ = Texturing-Einheit
- **Antialiasing** – zum Teil winkelunabhängige Kantenglättung
- **Anisotropes Filtern** – Schärfentiefe von Texturen
- freie Programmierbarkeit nahezu jeder GPU-Komponente (ua. Shader, beinhaltet T&L ... Transform and Lightning)
- **Multi-GPU-Techniken** – Zusammenarbeit mehrerer Grafikprozessoren

3.3.1. DirectX und OpenGL

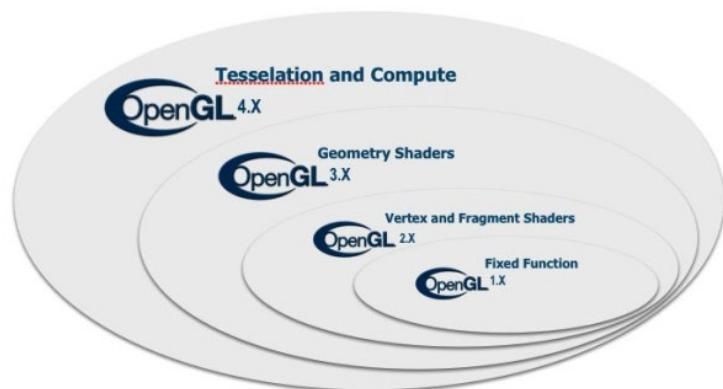
DirectX (aktuelle Version 12) ist eine Sammlung von Programmierschnittstellen (englisch: *Application Programming Interface*, kurz **API**) für multimedial intensive Anwendungen (besonders Spiele) auf der Windows-Plattform.

DirectX 12

Die DirectX-Sammlung von Software-Komponenten deckt nahezu den gesamten Multimediacbereich ab. Vorrangig kommt es zum Einsatz bei der Darstellung komplexer 2D- und 3D-Grafik, bietet aber auch Unterstützung für Audio, diverse Eingabegeräte (zum Beispiel Maus, Joystick) und Netzwerkkommunikation.

DirectX ermöglicht direkte Zugriffe auf die Hardware des Systems, ohne die Programme von der Hardware abhängig zu machen. So wird Spieleentwicklern eine **Hardware-Abstraktions-Schicht für die Spieleprogrammierung** zur Verfügung gestellt, mit der langsame Schnittstellen umgangen werden. Funktionen, die von der Hardware und damit nicht von der HAL bereitgestellt werden können, werden in der Hardware-Emulations-Schicht emuliert.

OpenGL (**Open Graphics Library**; aktuelle Version 4.6 / Juli 2017) ist eine Spezifikation für eine plattform- und programmiersprachenunabhängige Programmierschnittstelle zur Entwicklung von 2D- und 3D-Computergrafik. Der OpenGL-Standard beschreibt etwa 250 Befehle, die die Darstellung komplexer 3D-Szenen in Echtzeit erlauben. Zudem können andere Organisationen (zumeist Hersteller von Grafikkarten) proprietäre Erweiterungen definieren.



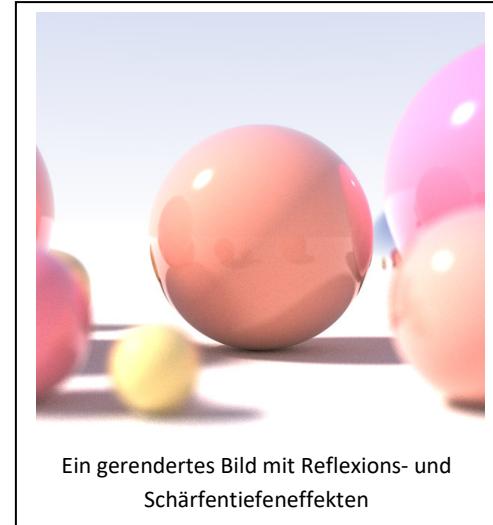
Die Implementierung des OpenGL-API erfolgt in der Regel durch Systembibliotheken, auf einigen Betriebssystemen auch als Teil der Grafikkarten-Treiber. Diese führen entsprechende Befehle der Grafikkarte aus, insbesondere müssen auf der Grafikkarte nicht vorhandene Funktionen durch die CPU emuliert werden.

3.3.2. Bildsynthese

Rendern oder **Bildsynthese** bezeichnet in der 3D-Computergrafik die Erzeugung eines Bildes aus einer Szene. Eine **Szene** ist ein virtuelles räumliches Modell, das Objekte und deren Materialeigenschaften, Lichtquellen, sowie die Position und Blickrichtung eines Betrachters definiert. Computerprogramme zum Rendern von Bildern werden *Renderer* genannt.

Beim Rendern müssen üblicherweise folgende Aufgaben gelöst werden:

- **Verdeckungsberechnung**
→ die Ermittlung der vom virtuellen Betrachter aus sichtbaren Objekte
- **Shading**
→ die Simulation des Aussehens von Oberflächen, beeinflusst durch deren Materialeigenschaften
- **Beleuchtung**
die Berechnung der Lichtverteilung innerhalb der Szene, die sich unter anderem durch die indirekte Beleuchtung zwischen Körpern äußert.

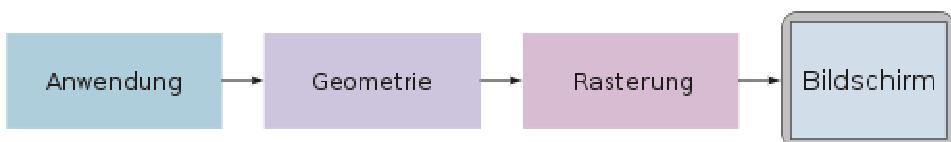


Echtzeitrendern

Beim Echtzeitrendern wird eine Reihe von Bildern schnell berechnet und die zugrundeliegende Szene vom Anwender interaktiv verändert. Die Berechnung erfolgt ausreichend schnell, dass die **Bildfolge als dynamischer Prozess** empfunden wird. Ab einer Bildfrequenz von etwa 6 fps ist eine interaktive Benutzung möglich, bei 15 fps lässt sich mit Sicherheit von Echtzeit sprechen. Auf modernen Computern wird das Echtzeitrendern durch Hardwarebeschleunigung mittels Grafikkarten unterstützt. Grafikhardware unterstützt in der Regel nur **Punkte, Linien und Dreiecke als grafische Grundobjekte**.

Bilderzeugungspipeline

Beim Echtzeitrendern beschreibt die **Grafikpipeline** den Weg von der Szene bis zum fertigen Bild. Die Grafikpipeline führt Berechnungen parallel aus und kann in drei große Schritte aufgeteilt werden:
Anwendung, Geometrie und Rasterung.



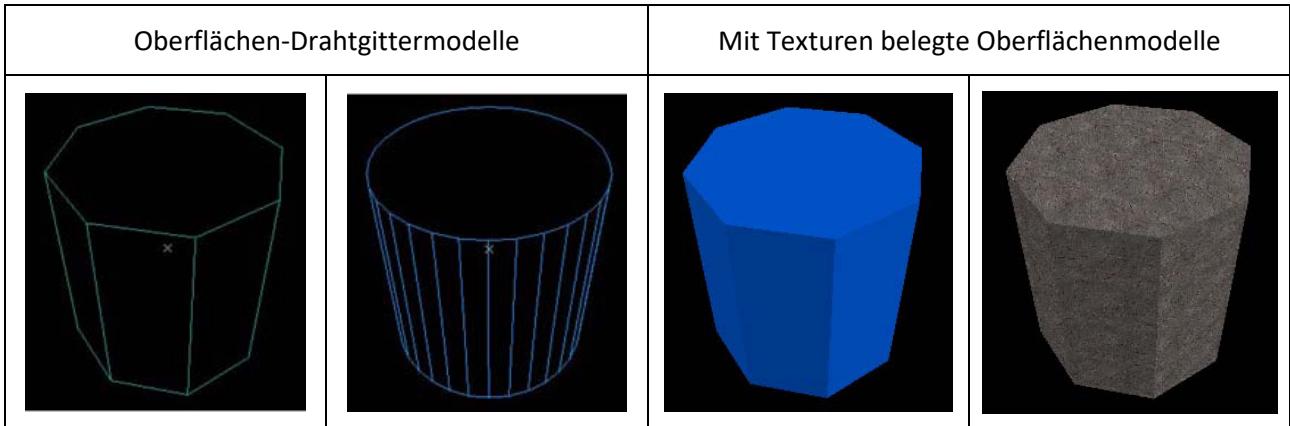
Ablauf:

1. **Anwendungsprogramm** (erzeugen der Szene, zerlegen in Primitive, Szenenbeschreibung)
2. **Geometrieprozessor** (geometrische Transformationen, Clippen an Bildgrenzen, perspektivische Transformationen, Beleuchtungssetup, evtl. Verdeckungsrechnung)
3. **Displayprozessor** (Scan-Konvertierung: Zerlegung in Rasterpunkte, Anti-Aliasing, Verdeckungsrechnung, Beleuchtungsrechnung und Shading, Texturierung)
4. **Bild-Speicher** (gerasterte Bilddaten)
5. **Videocontroller** (Filterung, Gammakorrektur, Bildsignalerzeugung)
6. Monitor

1. wird von CPU geleistet, 2.-5. von Grafikkarte

3.3.3. Texture Mapping

Texture Mapping (deutsch etwa **Musterabbildung**) dient dazu, die Flächen dreidimensionaler Oberflächenmodelle mit zweidimensionalen Bildern – sogenannten „**Texturen**“¹ – und Oberflächeneigenschaften auszustatten (= „Überzug“ für 3D-Modelle). Texturen lassen computergenerierte Bilder detailreicher und realistischer erscheinen, ohne dass das zugrundeliegende Modell selbst verfeinert werden muss.



Durch **Tessellation** wird ein bereits vorhandenes Dreiecksgitter, aus dem zB. fast alle 3D-Modelle in Spielen aufgebaut sind, feiner unterteilt. Voraussetzung ist eine entsprechende DirectX 11 oder OpenGL 4-GK!



Video: <http://www.hardwareclips.com/video/1160/nVidia-Alien-vs-Triangles-Tessellation-VideoDemo>

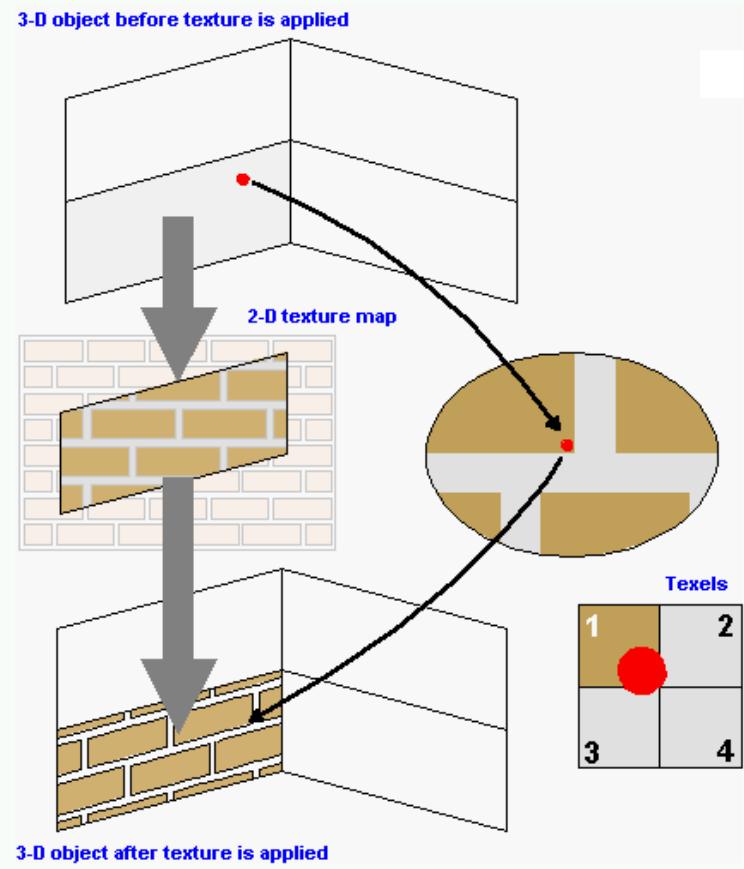
¹ Durch Verwendung einer **Bitmap** realistisch wirkende Oberfläche eines 3D Objektes, zB Ziegelsteine, Holz oder Stahl.

Texturinterpolation

Ein Pixel einer Textur wird als „**Texel**“ bezeichnet. **In der Regel ist es nicht möglich einen Bildschirm-Pixel exakt einem Texel zuzuordnen.** Vielmehr liegt ein Pixel meist zwischen mehreren Texeln. Es gilt also, zu entscheiden, wie aus den Farbwerten der umliegenden Texel der Farbwert für den Pixel gewonnen wird: Man benötigt ein geeignetes **Interpolationsverfahren**.

Das einfachste und schnellste solche Verfahren besteht darin, einfach den nächstliegenden Texel auszuwählen; man nennt dies **nearest neighbor** („nächster Nachbar“) oder auch **point sampling** („punktweise Stichprobe“).

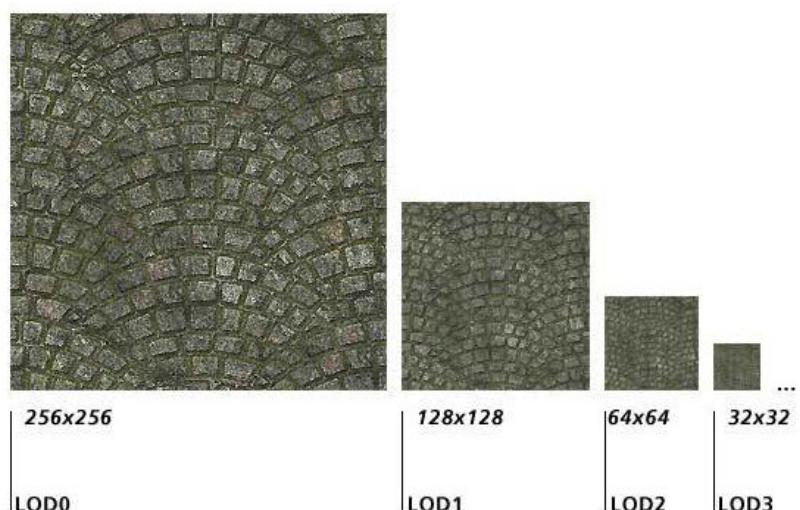
Beim aufwändigeren **bilinearen Filtern** wird der gesuchte Farbwert aus den vier umliegenden Texeln in Abhängigkeit ihrer Entfernung interpoliert (= „Box-Filter“).



- **MIP-Mapping**

Die obigen Techniken werden angewandt, solange der Rasterabstand der Pixel kleiner als jener der Texel ist, einem beliebigen Pixel also höchstens ein Texel zugeordnet wird. Ist der **Rasterabstand der Pixel jedoch größer als jener der Texel**, so entspricht einem Pixel gleich ein ganzer Bereich einer Textur. Hier bedient man sich sog. MIP-Maps. Diese enthalten neben der Originaltextur Kopien der **Textur mit abnehmender Größe**, sogenannte „Detailstufen“ (= *level of detail*, LOD).

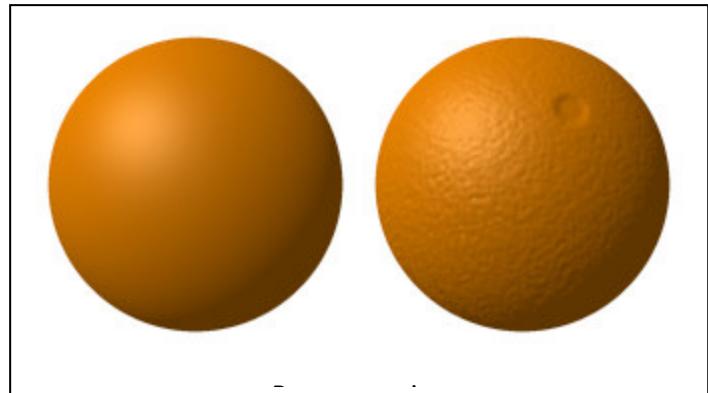
Man wählt daraus die größte Detailstufe aus, die den gewöhnlichen Zustand „Pixel kleiner als Texel“ wieder herstellt, und arbeitet darauf wie auf der Originaltextur. Zusätzlich zu den bisherigen Interpolationsverfahren bietet sich hier die Möglichkeit, zwischen zwei aufeinanderfolgenden Detailstufen eine weitere lineare Interpolation durchzuführen; in



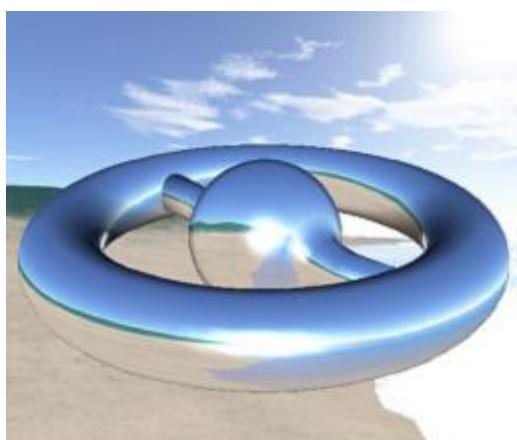
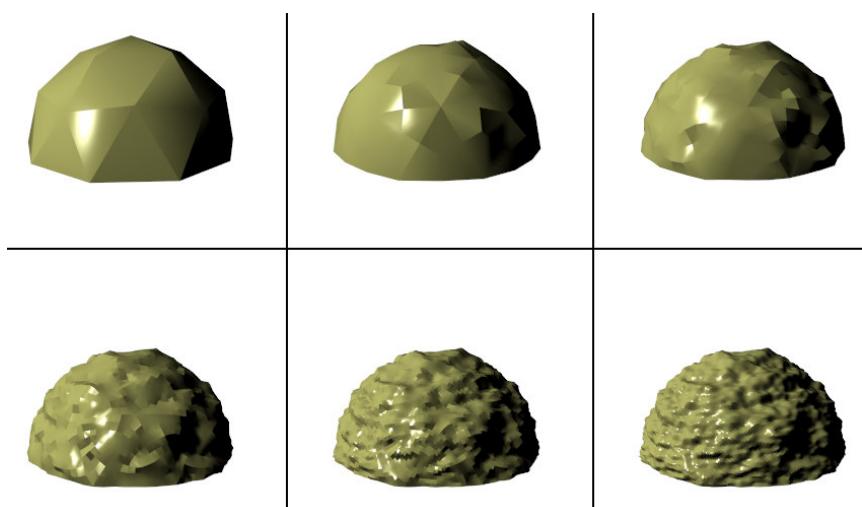
Kombination mit der bilinearen Filterung erhält man so eine **trilineare Filterung**.

Es gibt verschiedene Verfahren, um eine von einer Textur bedeckte **Oberfläche dreidimensional** erscheinen zu lassen:

- Beim **Bumpmapping** wird die Geometriekomplexität des Objekts nicht erhöht. Die benötigten Detail-Informationen werden in der Textur gelagert, mit deren Hilfe Schattierungen auf eine Oberfläche gezeichnet werden. Bumpmapping ist somit nur eine (wenngleich wirksame) **Illusion**, welche Oberflächenunebenheiten simuliert, die in der Geometrie des Modells gar nicht vorhanden sind. Sobald man ein Objekt aber „seitlich“ betrachtet gehen die Effekte, die durch bloße Farbänderungen eine Struktur simulieren, verloren.



- Beim **Displacement Mapping** werden mit der Information aus der Textur zusätzliche Polygone erstellt. Das heißt, betrachtet man eine Oberfläche aus der Nähe in einem flachen Winkel, so behält sie trotzdem ihre Struktur – da die Geometrie verändert wurde. Auch im Zusammenhang mit Lichtquellen und Schatten bringt Displacement Mapping einen größeren Realismus.

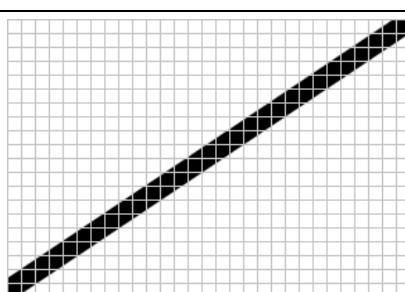
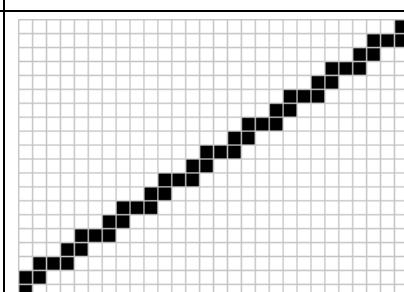
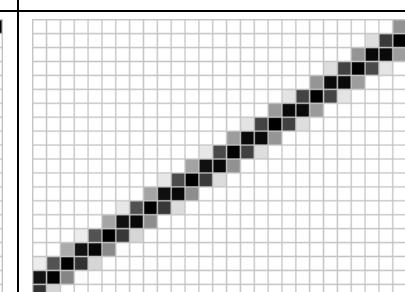


- Beim **Environment Mapping** wird mit Hilfe der Textur eine Spiegelung vorgetäuscht. Dabei wird die Umgebung des spiegelnden Objekts als Textur gespeichert und auf die Oberfläche des Objekts abgebildet

3.3.4. Antialiasing

Antialiasing bezeichnet Techniken zur Verminderung von unerwünschten, durch das begrenzt aufgelöste Pixelraster verursachten Effekten wie dem Alias-Effekt (= zu geringe Abtastung bei der Digitalisierung von Bildinformationen) oder dem Treppeneffekt, die bei der Erzeugung einer Computergrafik entstehen können. Wird Antialiasing vorrangig dazu eingesetzt, um das durch den Treppeneffekt verursachte kantige Erscheinungsbild abzumildern, so spricht man auch von Kantenglättung.

Beispiel: Ein mathematisch unendlich genau genaues Gebilde soll auf ein endlich genaues Raster abgebildet werden.

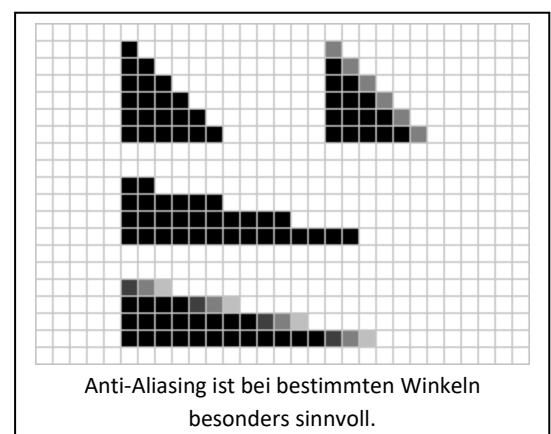
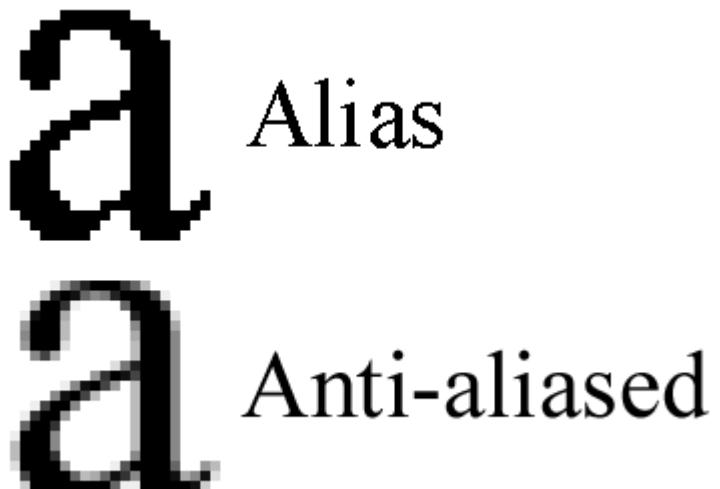
Original: unendlich genau	Eine Möglichkeit der Darstellung auf dem Bildschirm.	Darstellung mit „Antialiasing“ (Kantenglättung)
		

Verfahren

Ohne Anwendung von Antialiasing wird für jedes Pixel nur einmalig an einer relativ zum Pixel gleichbleibenden Position abgetastet. Beim Antialiasing wird die **Bildbeschreibung an mehreren und/oder relativ zum Pixel unterschiedlichen Positionen ausgewertet**.

Aus den so ermittelten Werten wird die **Farbe des Pixels gemäß einem Rekonstruktionsfilter berechnet**.

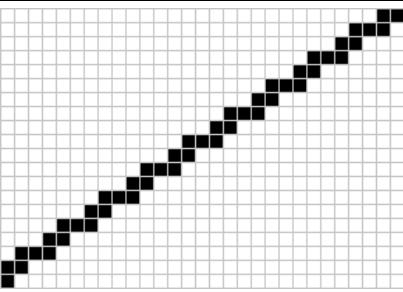
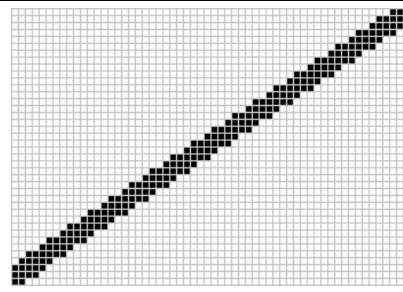
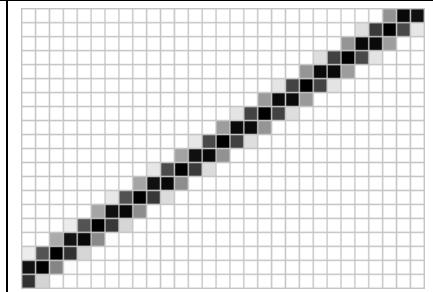
Durch die Wahl eines geeigneten Abtastverfahrens lässt sich **Prealiasing**, durch die Wahl geeigneter Rekonstruktionsfilter **Postaliasing** vermindern oder vermeiden. Die Bezeichnung „Antialiasing“ ist insofern irreführend, als Antialiasing nicht nur gegen Alias-Effekte, sondern auch gegen den Treppeneffekt und andere unerwünschte Effekte, etwa durch das Pixelraster fallende kleine Figuren, angewandt wird.



Hardware-Implementierung

Zum Echtzeitrendern mittels Grafikkarten ist Antialiasing meist **direkt in der Hardware implementiert**.

Die konzeptuell einfachste Methode ist **Full-Scene Antialiasing (FSAA)**. Dabei werden die Bilder mit höherer Auflösung (zB 2-fach, 4-fach, 8-fach) gerendert und anschließend heruntergerechnet (Variante des sog. *Supersamplings*).

„Original“	Auflösung und damit Samplerate erhöhen → SS	Ergebnis nach dem herunterrechnen
		

Eine andere Methode des Supersamplings verwendet den **Akkumulationspuffer**. Der Akkumulationspuffer ist ein Speicherbereich für Bilder, die dort nacheinander aufaddiert werden und von denen anschließend der Durchschnitt gebildet wird. Er steht auf einer Grafikkarte in der Regel zusätzlich zum Framepuffer bereit und wird auch für Tiefunschärfe- und Bewegungsunschärfeberechnungen eingesetzt.

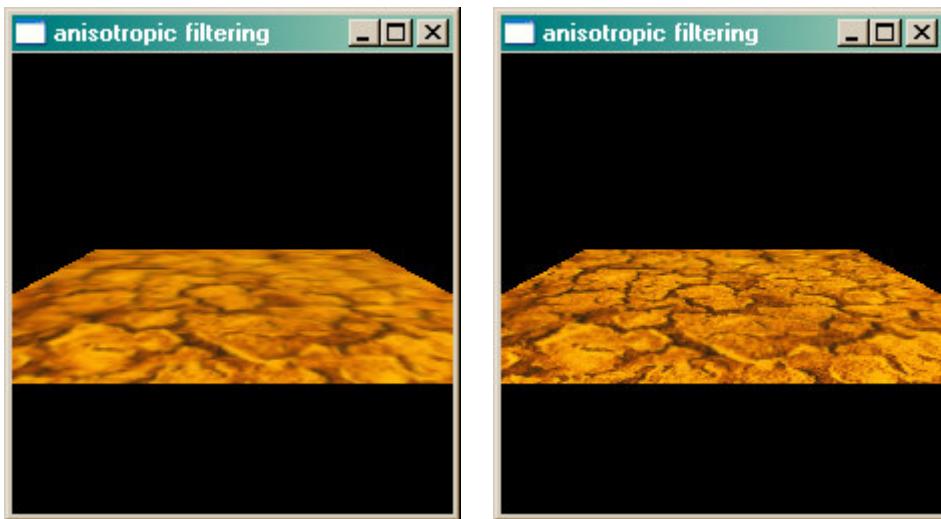
Um ein Bild mit vier Abtastpunkten pro Pixel zu rendern, werden nacheinander vier Bilder in den Puffer geschrieben, die jeweils um den Bruchteil eines Pixelabstands in alle Richtungen versetzt sind und ein Durchschnitt der Farbwerte berechnet.

Beim **Antialiasing von Texturen** werden einige besondere Techniken wie Mip Mapping (siehe dazu oben: Texture Mapping) und anisotropes Filtern (siehe unten) verwendet.

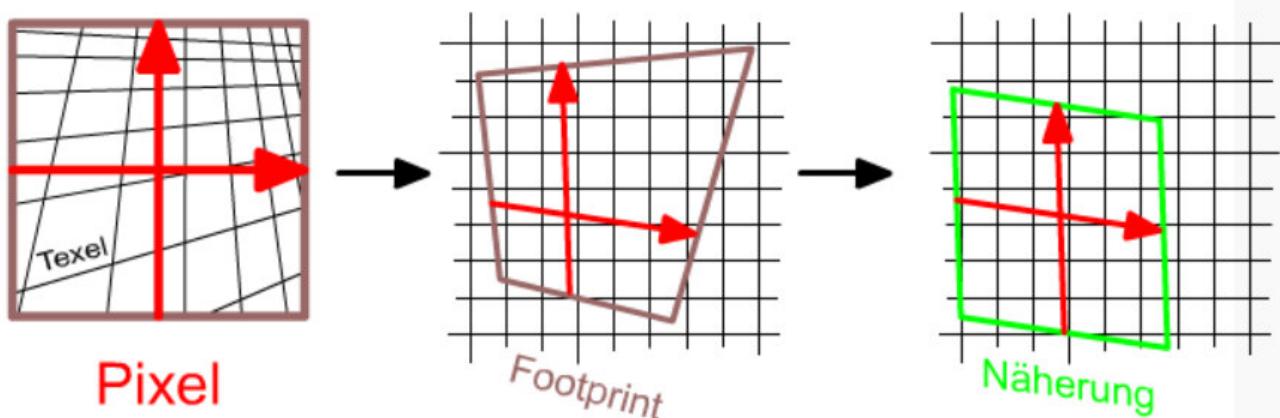
3.3.5. Anisotropes Filtern

Anisotropische Filterung ist eine Technik, die häufig eingesetzt wird, um unscharfe verschwommene Texturen unter einem flachen Betrachtungswinkel zu vermeiden.

Meistens werden Mipmaps und lineare Filter benutzt, um die Qualität von Texturen zu erhöhen. Unter einem flachen Betrachtungswinkel erscheinen diese jedoch „verwaschen“ (siehe Bild unten links). Bei anisotroper Filterung wird der Betrachtungswinkel berücksichtigt und es entstehen immer geschärfte Texturen.



Footprint Assembly ist ein Algorithmus zum anisotropen Filtern von Texturen. Die **Texturverzerrung** kann dabei in jeder beliebigen Richtung **berechnet** werden. Außerdem lässt sich der Algorithmus gut mit Techniken wie MIP-Mapping, bilinearem oder trilinearem Filtern kombinieren.



Beim Footprint Assembly wird der Footprint¹ vereinfachend als Parallelogramm angenommen. Hierzu wird der **Pixelmittelpunkt** in die **Texturkoordinaten** projiziert und gibt den Mittelpunkt des Parallelogramms an. Die Vektoren, die das Pixel entlang der beiden Hauptachsen aufspannen, werden ebenfalls in das Texturkoordinatensystem projiziert. Sie spannen nun das Parallelogramm auf.

Vielfach wird auch das Parallelogramm als Footprint bezeichnet.

¹ Die **Projektion eines quadratischen Bildschirmpixels** (Kantenlänge 1) **auf die Texturebene**, wird Footprint genannt.

3.3.6. Multi-GPU-Techniken

In der 3D-Computergrafik werden verschiedene Verfahren zur **Lastverteilung der Rechenarbeit auf mehrere Grafikchips** oder Grafikkarten eingesetzt. Der Sammelbegriff für diese unterschiedlichen Techniken lautet **Multi-GPU**. Obwohl jeder Hersteller ähnliche Betriebsmodi anbietet, ist der Name der jeweiligen Technik eine spezielle Marketing-Wortschöpfung

Scalable Link Interface – SLI

ist eine Multi-GPU-Technik von **Nvidia**, die die Zusammenschaltung von zwei oder mehr Grafikchips zur Leistungssteigerung beim Rendern (SLI Frame Rendering) oder den Einsatz von bis zu vier Bildschirmen (SLI Multi View) ermöglicht.

Theoretisch entspricht die **Renderleistung** im Mittel der Rechenleistung des langsamsten Grafikprozessors mal die Anzahl der verwendeten Grafikprozessoren. Bei zwei baugleichen Grafikkarten wird somit eine theoretische Verdoppelung der Rechenleistung erreicht, in der Praxis sind allerdings Leistungssteigerungen von etwa 30 bis 90 % möglich

Voraussetzungen:

- SLI-fähigen Chipsatz
- mindestens zwei PCI-E-Slots
- SLI-Brücke ... heute nicht mehr unbedingt notwendig, aber bei schnellen GK sinnvoll
- identische Grafikchips (nicht unbedingt baugleiche GK)



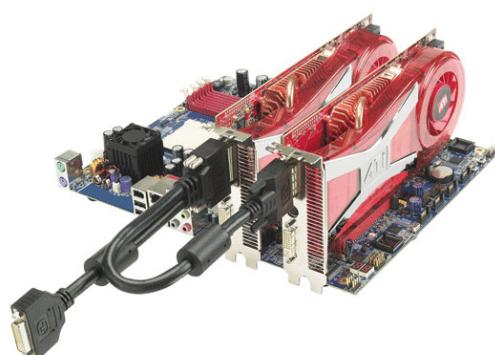
SLI-Verbund von zwei Grafikkarten (unten), Steckbrücke (rechts) und Steckverbinder (links) zur Verbindung

ATI Crossfire

Als **ATI Crossfire** bezeichnet das Unternehmen AMD (früher ATI Technologies) eine Multi-GPU-Technik, um zwei Grafikkarten in einem PCI-Express-System gleichzeitig zu betreiben.

Es gibt drei verschiedene Möglichkeiten, einen Crossfire-Verbund zu erstellen:

- Ein **spezielles Monitorkabel**, über welches beide Grafikkarten extern verbunden werden. Hier wird eine so genannte Master-Karte (am Namen „Crossfire Edition“ erkennbar) benötigt, die mit einer Slave-Karte (übliche Version) kombiniert wird. Ein spezieller Chip auf dieser Master-Karte mischt das Ausgabebild der zwei Grafikkarten zusammen.
- **Internes Crossfire**: Über zwei interne Brücken werden zwei baugleiche Grafikkarten verbunden.
- Kommunikation zwischen den Karten wird **über den PCI-Express-Bus**



Der Unterschied zum SLI von NVIDIA ist, dass bei Crossfire Daten generell **bidirektional** (Datenübertragung in beide Richtungen Punkt zu Punkt) ausgetauscht werden können.

3.4. Kühllösungen

Aufgrund der hohen thermischen Verlustleistung durch die zunehmende Komplexität von Grafikprozessoren bzw. teilweise auch des Grafikspeichers sind ähnlich **aufwendige Kühllösungen** wie bei Prozessorkühlern notwendig. Grafikkarten verbrauchen mit einem Grafikprozessor bis zu **300 W Leistung** und mehr, die vollständig als Wärmeenergie abgeführt werden muss. Dazu existieren mehrere Ansätze:

- **passive Luftkühlung** – durch einen Kühlkörper wird die thermische Energie durch Konvektion an die Umgebungsluft abgegeben. Das ist nur bei geringen Leistungen oder mit sehr großen Kühlköpfen möglich. Oft werden auf beiden Seiten der Grafikkarte großflächige Kühlköpfe angebracht, welche mit einer Heatpipe verbunden sind. Zum Problem kann dabei führen, dass die Kühlköpfe ein hohes Gewicht erreichen, was zu einer hohen mechanischen Belastung des Steckplatzes führt.
- **aktive Luftkühlung** – die thermische Energie wird über einen Kühlkörper an die Umgebungsluft abgegeben, welche durch Lüfter umgewälzt wird. Das ist die einfachste und preiswerteste Variante, große Wärmemengen abzuführen, verursacht allerdings auch Störgeräusche.
- **Wasserkühlung** – wenn für die CPU eine Wasserkühlung eingesetzt wird, kann auch die Grafikkarte in diesen Kreislauf eingebunden werden. Die thermische Energie wird dann an das Wasser im Kreislauf und von dort über einen Radiator an die Umgebungsluft abgegeben. Das ermöglicht einen Transport von großen Wärmemengen, ist aber auch die aufwendigste und teuerste Kühllösung. Es existieren auch einige Karten mit vorinstallierter Wasserkühlung mit dem Vorteil, dass die Garantie erhalten bleibt.



passiver beidseitiger Kühlkörper
(Radeon 9600 XT)



aktiver Luftkühler (PNY Geforce 6600 GT)

3.5. Ausgänge

Der RAMDAC (Random Access Memory Digital/Analog Converter) ist ein Chip, der für die **Umwandlung von digitalen (Videospeicher) in analoge Bildsignale (Monitor)** verantwortlich ist. Von ihm werden die Signalausgänge angesteuert. Er kann auch im Grafikprozessor integriert sein.

Externe Signalausgänge

VGA-Out  / Mini-VGA 

An einer 15-poligen D-Sub-Buchse wird ein analoges RGB-Signal bereitgestellt. Unter beengten Platzverhältnissen ist der Ausgang auch als Mini-VGA ausgeführt (zB beim Apple iBook). Über ein VGA-Kabel mit entsprechendem Stecker werden CRT-Monitor (Röhrenmonitor), Projektor oder Flachbildschirm (TFT) angeschlossen.

DVI-Out / Mini-DVI

Der DVI-Ausgang liefert ein **digitales Signal** und damit die beste erreichbare Bildqualität an Bildschirmen mit DVI-Eingang. Die meisten heutigen Grafikkarten sind mit einem DVI-I-Anschluss (*i* für *integrated*) ausgestattet und liefern damit zusätzlich ein analoges RGB-Bildsignal. Somit können mit einem (meist beiliegenden) passiven Adapter auch Bildschirme mit analogem D-Sub-Eingang angeschlossen werden, die Bildqualität entspricht dann jedoch weitestgehend der des D-Sub-Ausgangs. Es existieren weiterhin die Varianten DVI-D mit ausschließlich digitalen Signalleitungen und DVI-A mit ausschließlich analogen Signalleitungen. Es existiert, wie bei VGA, auch eine Mini-DVI-Variante für Notebooks ohne Platz für eine vollwertige Buchse.

HDMI-Out

Seit 2007 werden auch Grafikkarten mit HDMI (High Definition Multimedia Interface)-Ausgang angeboten. Hier wird das Videosignal ebenfalls **digital** und gegebenenfalls mit **HDCP verschlüsselt** ausgegeben. Über HDMI können auch DVI-D-Signale übertragen werden, womit DVI-Geräte kompatibel zu HDMI sind. Die Unterstützung von HDCP ist bei DVI jedoch optional, so dass nicht alle Geräte derartig geschützte Signale wiedergeben können. Die Übertragung von **Tonsignalen** ist jedoch nur über HDMI-Verbindungen möglich. Aktuelle Version **HDMI 2.0a**.



DisplayPort / Thunderbolt

DisplayPort ist ein Verbindungsstandard für Bild- und Tonsignale. Er ist **kompatibel zu VGA, DVI und HDMI** und unterstützt die Kopierschutzverfahren **HDCP** und DPCP (DisplayPort Content Protection). Die Verbreitung beschränkt sich auf wenige Grafikkarten und eine proprietäre Version am MacBook (**Mini DisplayPort**). **Thunderbolt** ist eine mechanisch und elektrisch mit Mini DisplayPort abwärtskompatible Anschlussbauform. Dies ist kein reiner Monitoranschluss mehr, sondern eine universelle Datenschnittstelle in der Art von USB.

Zusätzliche Signalausgänge und auch -eingänge sind je nach Karte unterschiedlich realisiert.

TV-Out (auch **Video-Out**)

Der als Cinch- oder S-Video-Buchse ausgeführte **TV-Ausgang** (engl. TV-Out) kann mit einem Fernseher oder Projektor verbunden werden. Man kann so mit mehreren Bildschirmen (PC-Bildschirm + Fernseher) arbeiten. Allerdings ist die Signalqualität des Anschlusses meist nicht sehr hoch.

Component-Out

Über den Komponenten-Ausgang (drei Cinch-Buchsen) werden **HDTV-Videodaten analog YPbPr-farbkodiert** ausgegeben. Der Ausgang ist wie meist nicht direkt auf der Grafikkarte ausgeführt.

TV-In (auch Video-In)

Neben einem Ausgang verfügen manche Karten auch über einen **TV-Eingang** zum Digitalisieren von externen analogen Videoquellen. Da es kaum Situationen gibt, bei denen TV-Out und TV-In gleichzeitig gebraucht werden, sowie aus Platz- und Kostengründen, sind TV-In und TV-Out oft in einer Buchse realisiert (Video-In-Video-Out). In diesem Fall ist eine gleichzeitige Nutzung als TV-In und TV-Out nicht möglich.

Ausblick

Der Weg ist vorgezeichnet hin zu einer **allgemeinen Schnittstelle**, die sowohl die Stromversorgung als auch die Datenübertragung (zB. Ausgabe von Audio- und Videodaten) erlaubt. Beispiele:

USB 3.1 mit Stecker Typ C,



Mobile High-Definition Link (MHL) und

Thunderbolt.



Übung

Du hast von deinen Verwandten zu Ostern insgesamt unglaubliche 500,- € geschenkt bekommen und möchtest dir nun eine neue Spielegrafikkarte für deinen PC kaufen.

Vergleiche dazu zwei beliebige Grafikkarten, die um dieses Geld erhältlich sind anhand der technischen Daten und von Performancetests.

Notiere auch Händler (inkl. Link) und Kaufpreis und argumentiere deine Kaufentscheidung.

Wiederholungsfragen zum Kapitel 5 Grafikkarten

1. Erkläre den grundlegenden Zweck einer Grafikkarte in einem PC-System!
2. Skizziere die historische Entwicklung der Grafikkarten!
3. Was sind die VESA BIOS Extensions?
4. Was sind „Windows-Beschleuniger“?
5. Wie heißt die gegenwärtige GK-Generation und welche Eigenschaften weist sie auf?
6. Nenne die Hauptkomponenten einer Grafikkarte!
7. Welche Schnittstellen für GK kennst du?
8. Wodurch ist eine PCIe-Schnittstelle gekennzeichnet?
9. Welche Aufgaben übernimmt der Grafikspeicher?
10. Warum benötigen Spiele-GK Grafikspeicher im GB-Bereich?
11. Was versteht man unter „Shared Memory“?
12. Welche Aufgaben übernimmt der Grafikprozessor?
13. Welche Software-Bibliotheken kennst du?
14. Wozu dienen DirectX und OpenGL?
15. Was versteht man unter Bildsynthese?
16. Was ist eine Szene?
17. Welche Aufgaben müssen beim Rendern gelöst werden?
18. Was versteht man unter Echtzeitrendern?
19. Erkläre die Bilderzeugungspipeline!
20. Was sind Texturen?
21. Erkläre „Texture Mapping“!
22. Was versteht man unter „Tessellation“?
23. Erkläre die Texturinterpolation!
24. Was ist ein Texel?
25. Erkläre MIP-Mapping!
26. Welche Verfahren sind geeignet Oberflächen dreidimensional erscheinen zu lassen?
27. Wodurch unterscheiden sich „Bump Mapping“, „Displacement Mapping“ und „Environment Mapping“?
28. Was versteht man unter „Antialiasing“?
29. Welche Hardware-Implementierungen für AA können unterschieden werden?
30. Erkläre „Anisotropes Filtern“!
31. Was ist ein „Footprint“?
32. Warum werden Multi-GPU-Techniken eingesetzt?
33. Welche Multi-GPU-Techniken kennst du?
34. Grenze SLI von Crossfire ab!
35. Welche Möglichkeiten gibt es die GPU zu kühlen?
36. Beschreibe die Signalausgänge einer Grafikkarte!
37. Welche GK-Ausgänge liefern analoge, welche digitale Signale?
38. Was ist HDCP?
39. Wohin geht der Videoschnittstellentrend?