

Graphik: Agenda

- Einführung
- Monitor/Display-Technik
- Anforderungen für 2D-Graphik

- 3D-Graphik
- Polygonbasierte Modellierung
- Texturen
- aktuelle Trends

- Voxel-basierte Graphik

Graphik: Literatur

www.nvidia.com , www.ati.com , www.3dfx.com	(GeForce / Radeon / Voodoo Graphikchips)
www.videologic.com	(PowerVR)
www.madonion.com	(3DMark Benchmark)
www.microsoft.com/directx	(Microsoft DirectX homepage und download)
www.sharkyextreme.com , www.3dconcept.ch	(aktuelle Infos und Tests)
www.tomshardware.com	(diverse Tests)
www.vesa.org	(Video Electronics Standards Association)

diverse c't Artikel, u.a. 19/99-248 (Direct3D/OpenGL), zuletzt 08/2000 (GeForce Test)

c't Artikelserie "3D a la carte", 4/89 bis 9/89 (volle Renderpipeline mit Pascal-Code)

www.flipcode.com/voxnut (Tutorial Voxel-Graphik)

Foley, van Dam, Fundamentals of Interactive Computer Graphics

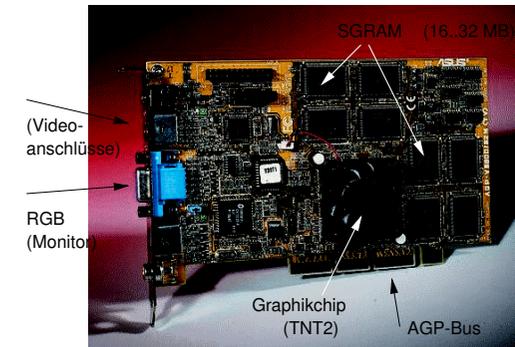
Bargen, Donnelly: Inside DirectX, Microsoft Press, 1998

Andre LaMothe: Tricks of the Windows Game Programming Gurus, Sams 1999

Alan Watt, 3D Computer Graphics, Addison-Wesley 1993

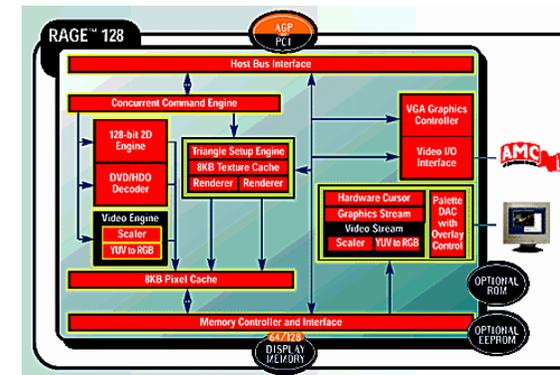


Graphik: typische Graphikkarte



- ASUS V3880 mit TNT2-Chip, 32 MByte SGRAM

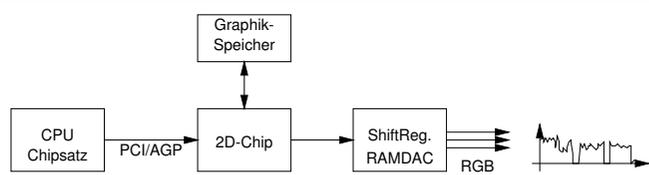
3D: Architektur ATI Rage 128



- separate Einheiten für 2D (VGA), 3D, Video
- diverse On-Chip Caches
- RGB-Ausgang zum Monitor, Video Ein/Ausgänge

[www.ati.com]

Graphik: Grundfunktionen



Bilddarstellung

- Auslesen des Bildspeichers (gewählte Auflösung / Wiederholrate)
- Color-Lookup-Table ("RAMDAC")
- serielles RGB/Videosignal mit Sync-Impulsen

2D-Funktionen

- Linien (Bresenham-Algorithmus)
- Rechtecke füllen / kopieren (BitBlit)
- Schrift

Graphik: Bresenham-Algorithmus

```

{ line slope 0 < slope < 1 }
procedure BRESENHAM( x1, y1, x2, y2, color: integer );
var dx, dy, incr1, incr2, d, x, y, xend: integer;
begin
  dx := ABS( x2-x1 );
  dy := ABS( y2-y1 );
  d := 2 * dy - dx;
  incr1 := 2 * dy;           { used for increment if d < 0 }
  incr2 := 2 * (dy-dx);     { used for increment if d >= 0 }

  if (x1 > x2) then begin
    x := x2;
    y := y2;
    xend := x1;
  else begin
    x := x1;
    y := y1;
    xend := x2;
  end
  WRITE_PIXEL( x, y, color );      { first point on line }

  while (x < xend) do begin
    x := x + 1;
    if (d < 0) then
      d := d + incr1;
    else begin
      y := y + 1;
      d := d + incr2;
    end
    WRITE_PIXEL( x, y, color ); { selected point near line }
  end
end {BRESENHAM}

```

Graphik: Speicherbandbreite

Bandbreitenbedarf für 2D-Graphik:

	Auflösung	Farbtiefe	f/Hz	MB / s
CGA	24x80 (x8)	4	75	0.576
ATARI ST	640x400	1	72	2.3
EGA	480x350	4	60	5
VGA	640x480	8	75	23
XGA	1024x768	16	75	118
UGA	1600x1200	32	100	768
BX-SDRAM-100				250
RAMBUS PC800				< 1600

- ohne Koordinatentransformation
- ohne Z-Buffer, Double-Buffering, Texturen

Graphik: Speicherbandbreite 3D

Speicherbedarf bei: 1024x768 Pixel, 16-bit Farben, 30 fps

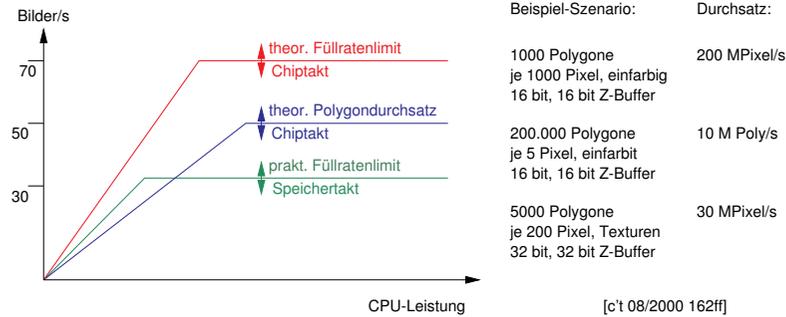
- Framebuffer: 1024x768x2 Byte 1.5 MB
- double buffering: 1.5 MB
- Z-Buffer 1.5 MB
- Rest frei für Texturen

Bandbreite, Annahme: im Mittel 3 überlappende Polygone / Pixel

- Bildwiedergabe (75 Hz): 118 MB/s
- double-buffer zeichnen (30fps) 47 .. 141 MB/s
- Z-Buffer lesen und schreiben (30fps *(3 + 1)) 188 MB/s
- Texturen (4 Zugriffe a 16 bit pro Pixel, 30 fps * 3) 1132 MB/s

=> Texturmapping ist der Engpaß ("Füllrate")

Graphik: Füllrate / Polygondurchsatz



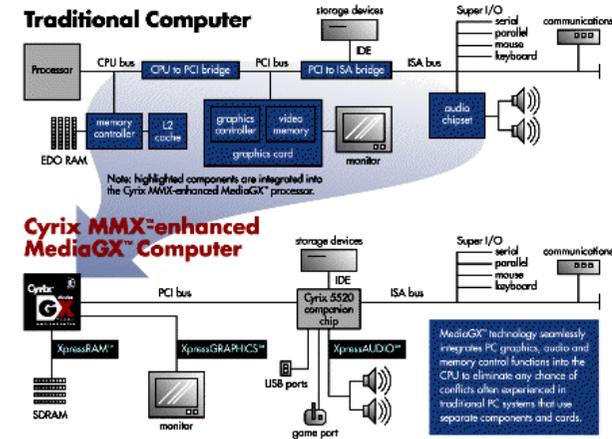
- drei zentrale Kenngrößen für Graphikchip-Bewertung
- extremer Einfluß der Textur-Algorithmen
- derzeit meistens durch Füllrate limitiert

Graphik: Unified memory architecture

UMA := gemeinsamer Haupt- und Graphikspeicher

- uraltes Prinzip (8-bit μ Ps, Atari ST, ...)
- doppelte Motivation:
 - minimale Kosten, da kein separater Graphikspeicher
 - CPU kann direkt die Graphikdaten manipulieren
 - aber Speicherbandbreite stark reduziert
 - Speicherkapazität genauso wie bei separatem Graphikspeicher
- interessant vor allem für low-cost Rechner
- Beispiele: Intel i810 Chipsatz, Cyrix MediaGX
- für aktuelle 3D-Anwendungen ungeeignet

Graphik: Cyrix MediaGX



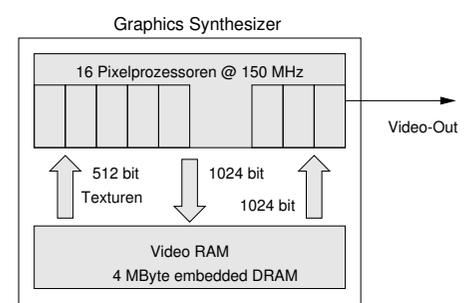
- PC mit nur drei Chips (CPU, Companion, Super IO) + DRAM

Graphik: Playstation2 Emotion Engine

UMA := gemeinsamer Haupt- und Graphikspeicher

- uraltes Prinzip (PC/AT CGA, Atari ST, ...)
- doppelte Motivation:
 - minimale Kosten, da kein separater Graphikspeicher
 - CPU kann direkt auf Graphikdaten zugreifen
 - aber Speicherbandbreite stark reduziert
 - Speicherkapazität genauso wie bei separatem Graphikspeicher
- interessant vor allem für low-cost Rechner
- Beispiele: Intel i810 Chipsatz, Cyrix MediaGX
- für aktuelle 3D-Anwendungen ungeeignet

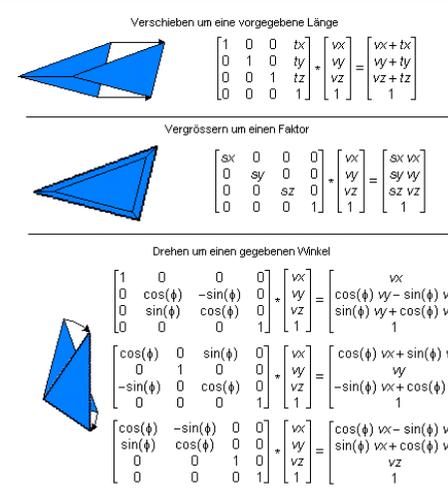
Graphik: Playstation2-Architektur



Graphikspeicher als embedded-DRAM

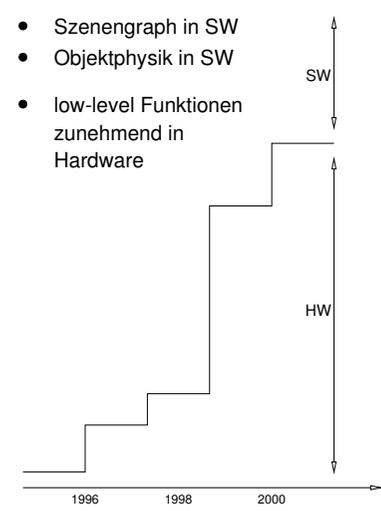
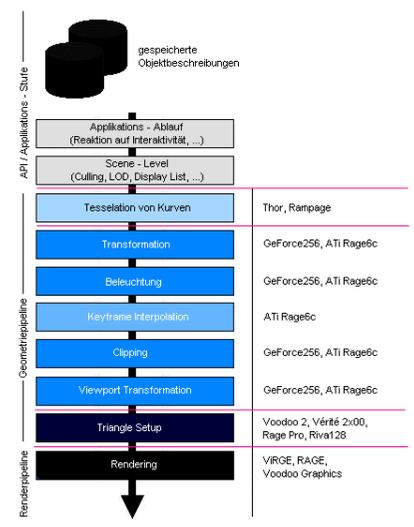
- 2560 bit Speicherinterface (on chip)
- extrem hohe Bandbreite (vermutlich TB/s), Füllrate 48 GB/s
- vergleiche IRAM-Konzept
- technologiebedingt derzeit nur 4 MB Kapazität
- maximale Auflösung nur 640x480

Graphik: 3D-Transformationen



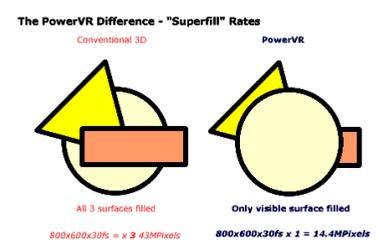
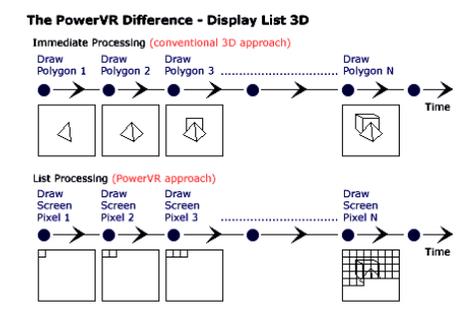
- homogene Koordinaten
- Translation
- Skalierung
- Rotation
- perspektivisch korrekt

Graphik: 3D-Pipeline



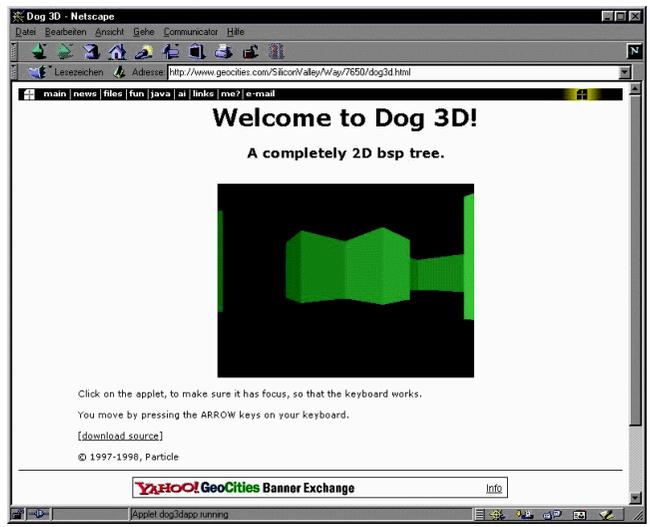
- Szenengraph in SW
- Objektphysik in SW
- low-level Funktionen zunehmend in Hardware

Graphik: PowerVR Konzept



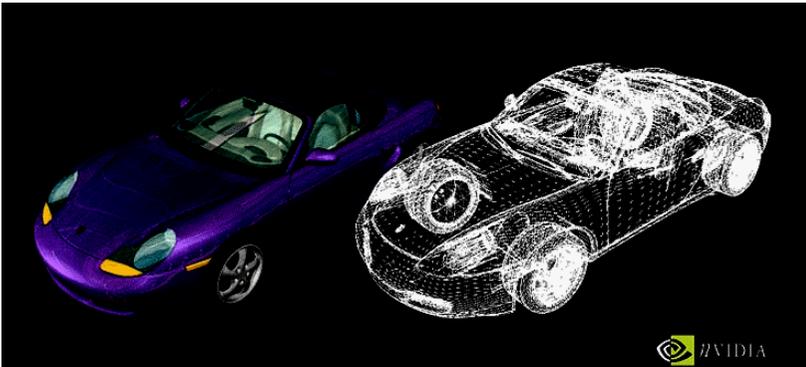
- interessantes Konzept
- erfordert angepasste Applikation
- oder Umsortieren im Treiber
- Dreamcast / SEGA "Naomi"

3D: Binary Space Partitioning



PC-Technologie | SS 2001 | 18.214

3D: Wireframe / Phong-Shading



- runde Formen erfordern sehr feines Netz
- oder bessere Modellierung (NURBS etc.)

PC-Technologie | SS 2001 | 18.214

3D: Backface-Culling



Backface-Culling: ("Rückseiten-Aussortierung")

- Flächen werden nur dargestellt, wenn Normale "nach vorne" zeigt
- Objektvorderfläche an Sichtpyramide abgeschnitten

PC-Technologie | SS 2001 | 18.214

3D: Polygonanzahl und Bildqualität



- Optimierung der Polygonanzahl ist interessantes Problem !

PC-Technologie | SS 2001 | 18.214

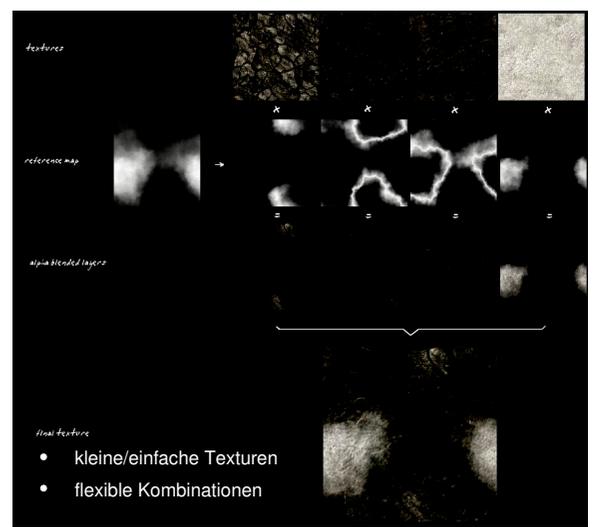
3D: Shading vs. Texturen



[c't 10/97 150]

- MotoRacer mit und ohne Texturen (wegen Fehler der Graphikkarte)
- einfaches Shading nur für Plastik / Metall-Oberflächen geeignet

3D: überlagerte Texturen



- kleine/einfache Texturen
- flexible Kombinationen

[www.flipcode.com/voxtut]

3D: Clipping-Fehler



[Tomb Raider 2, 640x480]

- zu geringe Genauigkeit der Kollisionserkennung / des Z-Buffers
- bekanntes Problem in TR2
- allgemein: "binary space partitioning" vs. transparente Objekte

Texturen: bilinear vs. ungefiltert



bilinear gefiltert | ohne Filter:

- wesentlich verringerte Pixelstrukturen
- Transparenz über Alpha-Blending:
- keine Glättung der Objektkanten:

vierfacher Rechenaufwand
 Fenster auch geglättet
 => Antialiasing notwendig

Graphik: Karten-Vergleich

Grafikkarten mit anderen Chips unter Direct3D 6.1 und OpenGL	3D Mark 99 MAX, Racing			3D Mark 99 MAX, 1st Person			Unreal 2.5 ¹			Expendable, 1024/16			Expendable, 1280/16			Expendable, 1024/32			Quake II, Crosses		
	besten #			besten #			besten #			besten #			besten #			besten #			besten #		
Siemens III S540	33	34	31	31	31	20	20	24	16	24	16	20	20	20	30	30	30	30	30	30	30
Winner II	33	34	31	20	21	19	13	13	17	17	17	17	17	17	26	26	26	27	27	27	27
Beast 99	25	25	32	18	18	21	14	14	17	17	17	17	17	17	26	26	26	27	27	27	27
S89	34	35	33	21	22	26	16	16	20	20	20	31	31	31	31	31	31	31	31	31	31
Rage 128	32	34	33	25	25	26	18	18	24	24	24	31	31	31	33	33	33	33	33	33	33
Millennium G400	34	34	36	27	27	44	37	37	41	41	41	39	39	39	37	37	37	37	37	37	37
Voodoo 3 3000	34	47	36	44	49 ²	44	41	41	43	43	43	47	47	47	47	47	47	47	47	47	47

Alle Tests durchgeführt unter Windows 98 und DirectX 6.1, sofern nicht anders angegeben bei 1024x768, 16 Bit, 75 Hz Bildwiederholrate, 'Wait for VSync' off, Audio ein, alle Werte in Frames pro Sekunde [fps].
 1024: 1024 x 768, 1280: 1280 x 1024, /16: HiColor 16 Bit, /32: TrueColor 32 Bit
 Pentium II, 400 MHz Pentium III, 500 MHz ¹ Direct3D, shiny surfaces ein ² via GlideAPI

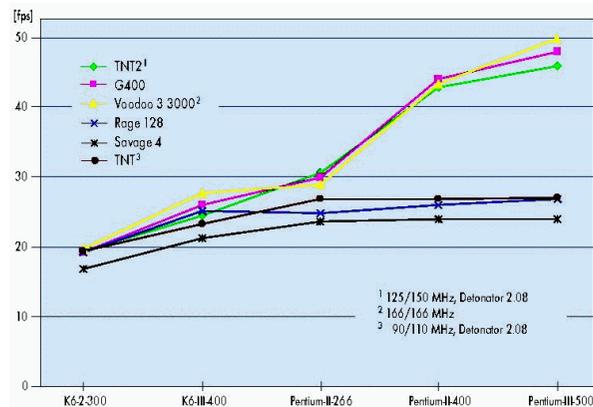
- Benchmark-Daten für Savage / Rage 128 / G400 / Voodoo 3
- Karten mit Riva TNT/TNT2 ähnlich
- alle Spiele in 1024x768x16 spielbar
- Voodoo3 Glide-API performanter als DirectX 6.1

Trends: Fotorealismus



- viele Polygone, gute Texturen, Schatten, . . .

Graphik: Karten-CPU-Vergleich



- Limitierung durch Füllrate / Polygondurchsatz gut sichtbar
- FPU des Pentium-III deutlich besser als K6
- Athlon/Duron etwa gleichwertig zum PIII

Trends: Texturkompression



256x256x16 Pixel = 128 KB

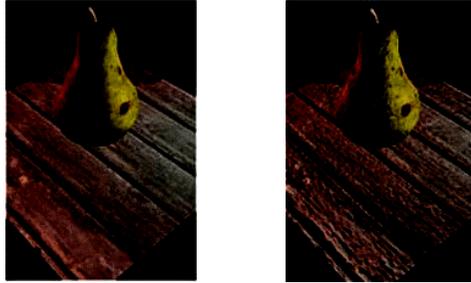


2048x2048x16 Pixel = 8 MB

hochauflösende Texturen wichtig für:

- Hintergrund / Himmel / usw.
- Nahansichten
- Kompression notwendig, um Kapazität/Bandbreite zu sparen
- diverse Verfahren, u.a. DirectX7

Trends: Bump mapping



normal

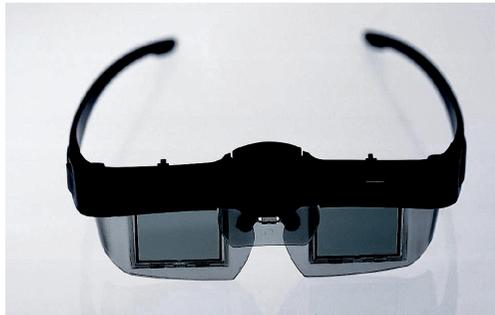
bump mapping

- Überlagerung mehrerer Texturen
- für "rauhe" Oberflächen
- statt aufwendiger Polygonmodellierung

[www.ati.com]

PC-Technologie | SS 2001 | 18.214

Trends: Stereo



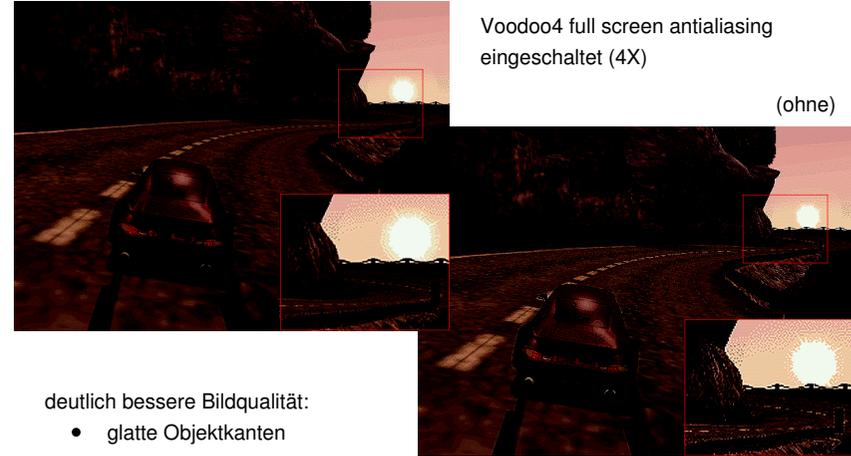
Shutterbrille:

- abwechselndes Abdunkeln der Augen
- Anzeige des linken / rechten Stereobildes
- per Treiber: ohne Modifikation der Applikation
- erfordert doppelte Framerate

[www.elsa.com]

PC-Technologie | SS 2001 | 18.214

Trends: 3dfx FSAA-Demo



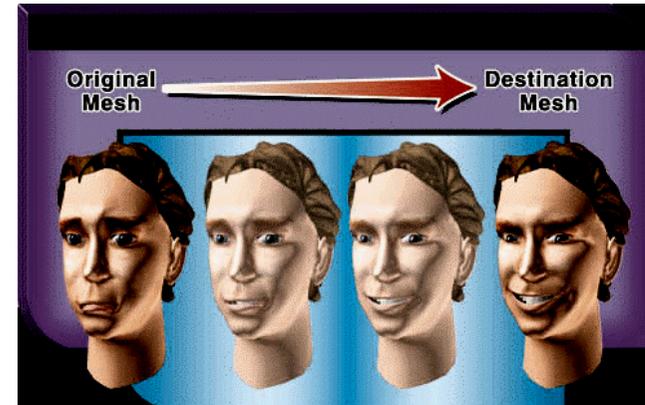
deutlich bessere Bildqualität:

- glatte Objektkanten
- weniger Flimmern
- aber Schärfeverlust

[www.3dfx.com]

PC-Technologie | SS 2001 | 18.214

Trends: "Keyframe-Interpolation"



- Interpolation zwischen Keyframes
- direkt auf der Graphikkarte
- für flüssigere Animationen

[www.ati.com]

PC-Technologie | SS 2001 | 18.214

Trends: Max Payne "bullet cam"



[Max Payne Demo]
[www.pcgames.de]

- Kineffekte in Echtzeit
- Zeitlupe, Replay, ...
- Problematik Gewaltverherrlichung ?!

Voxel: Beispiel DeltaForce2



- Voxel-basierte Modellierung der Umgebung
- ermöglicht kleine Objekte (=> Gras, Rauch)
- Häuser, Darsteller als Polygonmodelle

[www.deltaforce2hq.com]

Voxel: Beispiel DeltaForce2



- weitläufigere Landschaften als mit Polygonmodellierung

[www.deltaforce2hq.com]