

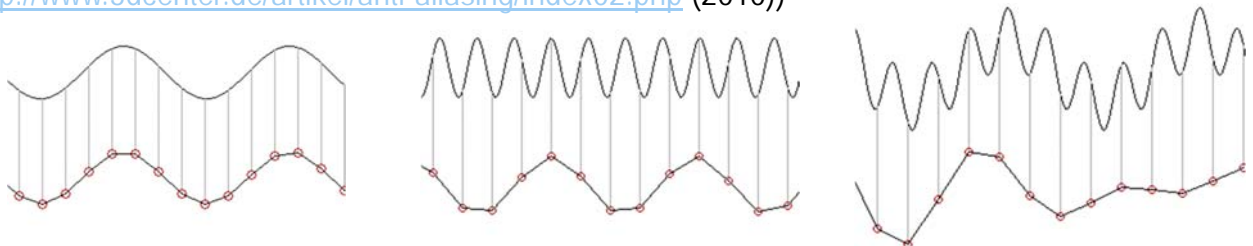
2. Operationen auf dem Bilddraster - 2.3) Antialiasing

- ▶ **Aliasing** entsteht u.a. *bei der Kameraaufnahme* von diskreten, digitalen Bildern, wenn das Bild in Bezug auf das Abtastgitter zu hohe Frequenzen enthält (Verletzung des Abtasttheorems).

- Die höchste zulässige Frequenz wird als Nyquistfrequenz bezeichnet. Sie ist gleich der halben Abtastfrequenz.

$$k_{x,Nyqu.} = \frac{1}{2 \cdot \Delta x}, \quad \Delta x = \text{Pixelabstand}$$

- Aliasing im Zusammenhang mit der Kameraaufnahme ist BV-typisch → Vorlesung BV&ME
- ▶ Zum besseren Verständnis des Grundproblems aber drei kleine Bildchen aus dem Internet (<http://www.3dcenter.de/artikel/anti-aliasing/index02.php> (2010))



- Der in der Mitte dargestellte Aliasing-Artefakt macht deutlich, dass Frequenzen oberhalb der halben Sampling-Frequenz *in den zulässigen Frequenzbereich zurückgespiegelt* werden. Aliasing entsteht also durch Unterabtastung. Nachträgliches Filtern kann das Aliasing nicht mehr beseitigen. (Bild rechts: *selbst zulässige Anteile in Gemischen* - hier Mix der beiden linken Frequenzen - *lassen sich nicht reproduzieren*)

2.3) Antialiasing (CG)

- ▶ Aliasing kann aber auch bei der *Ausgabe von Graphiken oder Bildinhalten auf Ausgabegeräten* entstehen, denn dieser Prozess ist in der Regel mit einer Rasterung (rasterisation) verbunden.
- *Dies trifft massiv die Computergraphik und steht deshalb hier im Mittelpunkt!*

- Aliasing entsteht in der Computergraphik insbesondere dann, wenn die

Graphiken in viel höherer Auflösung

vorliegen *als das Raster des Ausgabegerätes* (z.B. 3D-Spiele, feinste Texturen)

- Da sich Aliasing-Artefakte nicht nachträglich beseitigen lassen, kann Anti-Aliasing nicht erst dann ansetzen, wenn das Bild bereits fertig gerendert ist.

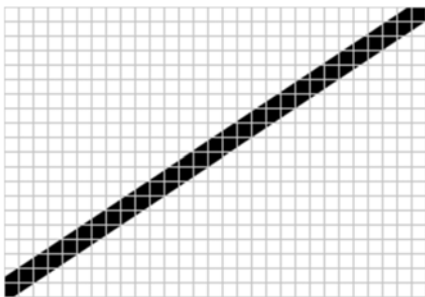
► Verfahrensübersicht

- Supersampling-Verfahren (Supersampling und Downsampling)
- Supersampling mit rotated grids
- Umsetzung / Realisierungsvarianten → Superbackbuffer (Supersampling) und Multisample-Buffer (Multisampling)
- Multisampling zur reinen Kantenglättung
- Quincunx / Überfilterung
- Kombinationen und pseudozufälliges Supersampling

2.3) Antialiasing (CG)

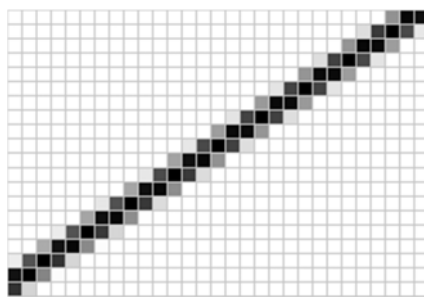
► Supersampling-Verfahren (Supersampling und Downsampling) - Problemdiskussion

- bei der Ausgabe von Polygonen auf Rastergeräten kann nur begrenzter Aufwand betrieben werden:



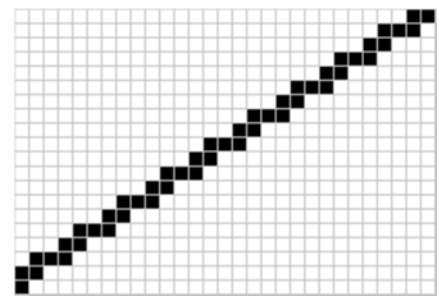
Ideales Objekt

kann so nicht auf Rastergraphik
nicht ausgegeben werden



Ideales Antialiasing

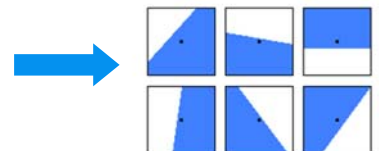
Grauwerte der Randpixel
gemäß Flächenbedeckungs-
grad (Aufwand ist zu hoch)



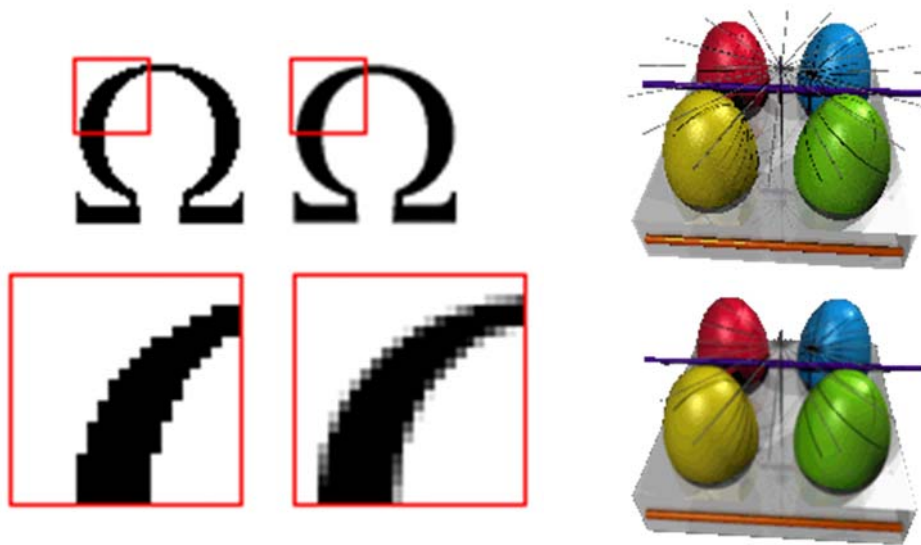
Objektdarstellung ohne Antialiasing

gesetzt wird, wenn Vordergrund
> 50%

Für die Darstellung oben rechts ohne Antialiasing erfolgt
ein *einfacher Test über die Pixelkoordinate*.
Die Qualität derartiger Darstellungen kann nicht
zufrieden stellen!



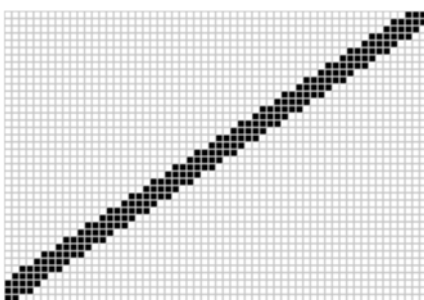
- *Aliasing / Antialiasing von Objektberandungen und Linien*



2.3) Antialiasing (CG)

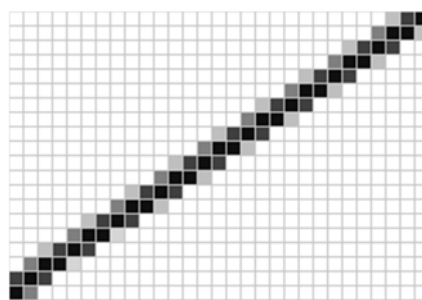
2.3.a) Supersampling

- Statt der zu aufwändigen Flächenberechnung der Pixel (oben diskutiertes ideales Antialiasing) wird die Graphik zunächst in viel **höherer Auflösung gerendert (Supersampling)**



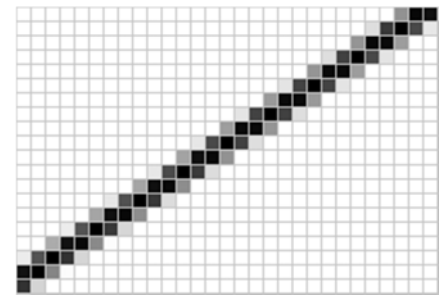
Supersampling

2x2 Subpixel für ein Outputpixel, bei einer Displayauflösung von 1920x1080 → 3840x2160 Pixel.



Downsampling

auf die Displayauflösung durch bilineare Interpolation (Mittelung der vier zu einem Makropixel gehörenden Subpixel).



Ideales Antialiasing

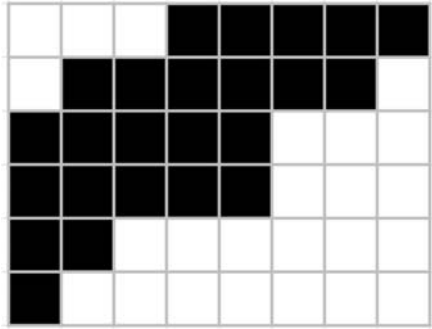
Grauwerte der Randpixel gemäß Flächenbedeckungsgrad (Aufwand ist zu hoch)

- Das Antialiasing im Displayraster ergibt sich schließlich durch Downsampling mittels bilinearer Interpolation.
(neben dem 4x-Supersampling (2·2) sind auch 2x-Supersampling (1·2) und 2.25x-Varianten (1.5·1.5) üblich, die beiden letztere z.B. zur Durchsatzsteigerung)

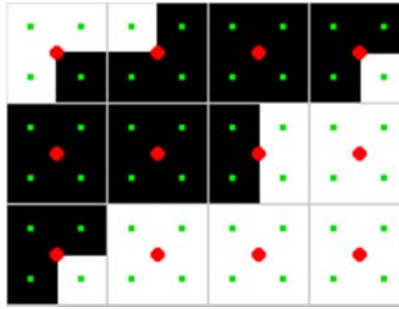
2.3) Antialiasing (CG) - Supersampling

- Zur Erläuterung des Downsamplings:

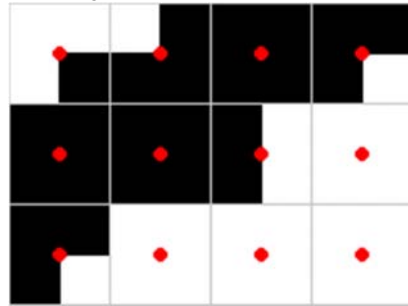
Objekt im Supersampling-Raster



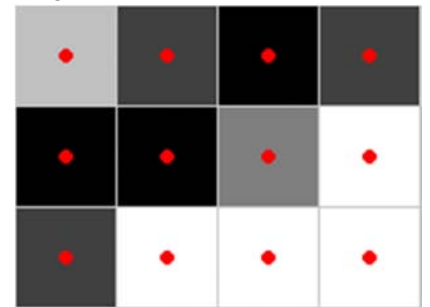
Pixel- und Subpixelpositionen
(hier ordered grid)



Display-Pixelpositionen

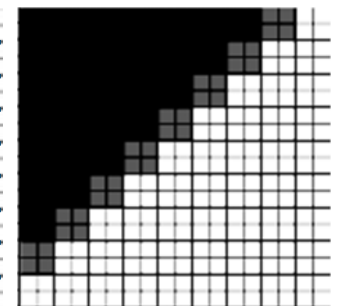
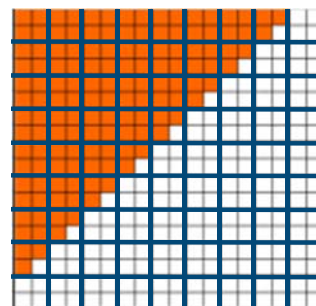
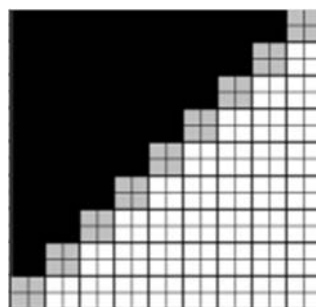
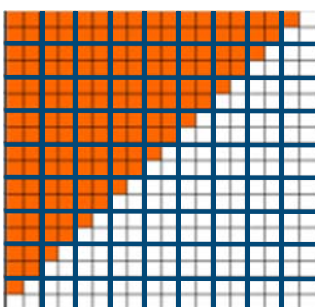


(Makro)Pixelpositionen und zugeordnete Grauwerte



2.3) Antialiasing (CG) - Supersampling

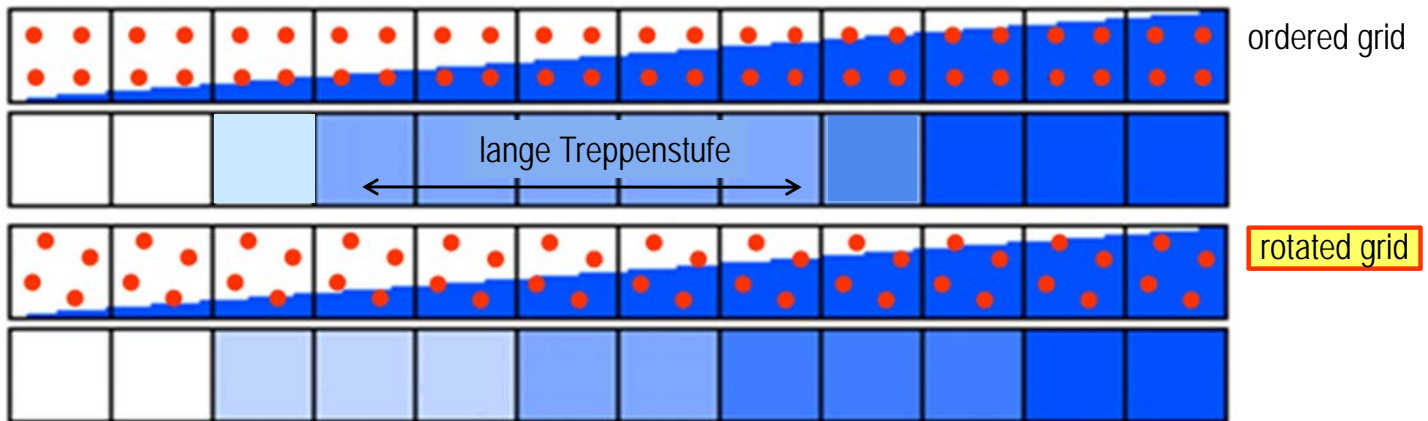
- Wie wirksam ist das Antialiasing bei 4x-Supersampling ?**
 - Ohne Anti-Aliasing kommt pro Pixel genau eine Sampleposition zum Zuge. Das Pixel wird demnach gefärbt oder nicht gefärbt: Das sind zwei mögliche Stufen.
 - Bei vier Subpixeln können minimal 0 und maximal 4 Subpixel im (Makro)Pixel gesetzt sein, d.h. es sind Intensitäten von 0%, 25%, 50%, 75% oder 100% möglich (5 Abstufungen)!
 - Der Zusammenhang ist einfach: Es gibt immer eine Abstufung mehr als Subpixel pro Pixel.
 - Beim idealen Antialiasing entsprechend Flächenbedeckungsgrad gibt es „beliebig“ viele Abstufungen (aus Aufwandsgründen praktisch nicht relevant).
- Gibt es eine Formabhängigkeit ?**
 - Bei 45° gibt es z.B. nur eine Zwischenstufe, und zwar je nach Phasenlage mit 25% oder 75%.
→ Kante „pumpt“ bei Objektbewegung



2.3 Antialiasing (CG) - Supersampling

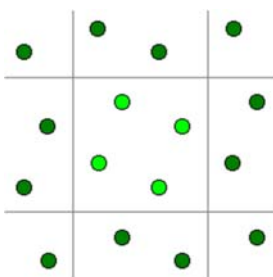
► Weiter zur Formabhängigkeit → Supersampling mit **rotated grids**:

- Kleine Winkel zu den Achsen führen zu langen „Stufen“ der digitalen Polygonkante, die gegenüber 45° eine feinere Grauwertabstufung zur Approximation des Pixel-Überdeckungsgrades zulassen.
- Noch bessere Verhältnisse der Grauabstufung ergeben sich für die flachen Winkel, wenn statt des „ordered grid“ ein „**rotated grid**“ für das Supersampling verwendet wird.
- **Rotated grids** sind dafür bei anderen Winkeln etwas schlechter als das ordered grid. Dies wird aber kaum wahrgenommen, da dort die Treppen der digitalen Geraden kürzer sind.



2.3 Antialiasing (CG) - Supersampling

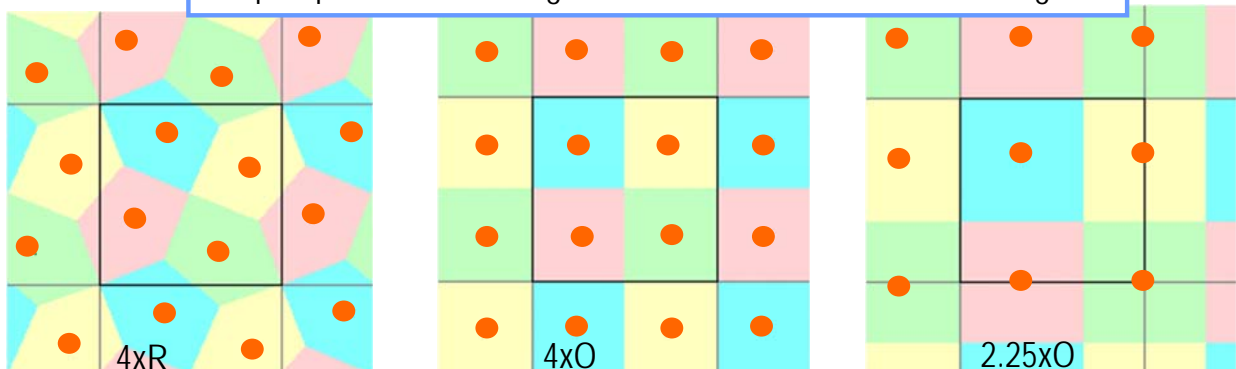
■ rotated grids:



✓ Vorteile:

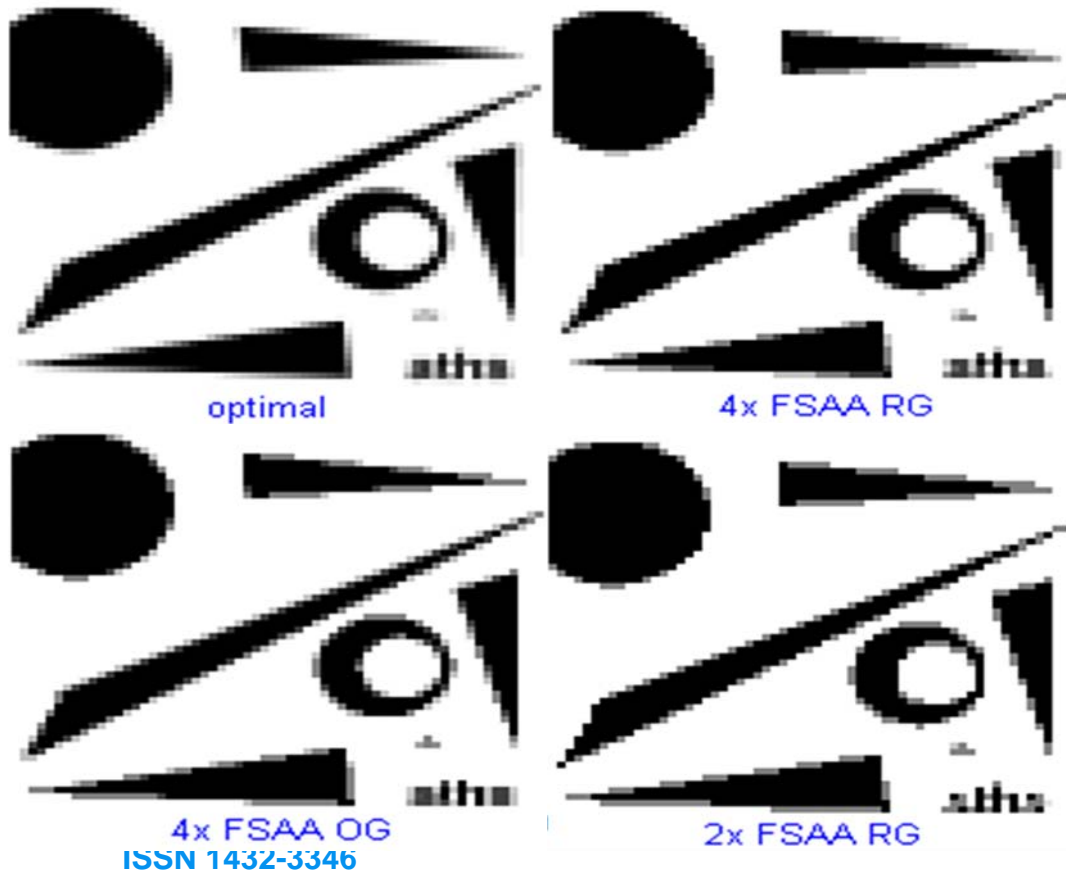
- Gute Grauabstufung bei sehr flachen Kanten zur Zeilen- oder Spaltenrichtung. *
- Optimaler Winkel liegt bei ca. 20°-30°.
- Rotated grid ist oft flexibel einstellbar ** (z.B. VSA-100 Chip).
- Sehr dünne Linien bleiben auch bei Bewegung zusammenhängend *** (Vermeidung von „Line Popping“)

Subpixelpositionen und zugeordnete Flächen für verschiedene „grids“



2.3) Antialiasing (CG) - Supersampling

- ordered und rotated grids (Vergleich) *:



4x FSAA RG: vierfach
Fix Sampling Anti-Aliasing
Rotated Grid
(die anderen Bezeich-
nungen entsprechend)

CG_KHF_V5 - 11

2.3) Antialiasing (CG)

2.3.b) Supersampling und Multisampling

- Realisierung von rotated Grids durch Superbuffer (Supersampling) und Multisample-Buffer (Multisampling):

✓ Variante 1: Realisierung des rotated grids durch Superbuffer



Nachteil:

Anpassung der Rasterkonvertierung an die verschobenen Positionen erforderlich.

Vorteil:

Textur wird subpixelgerecht eingetragen

✓ Variante 2: Realisierung des rotated grids durch Multisample-Buffer
Mehrfachrendering in normaler Größe mit leicht versetzter Geometrie (Offsetraster, Vertex-Jittering, parallel *).



Vorteil:

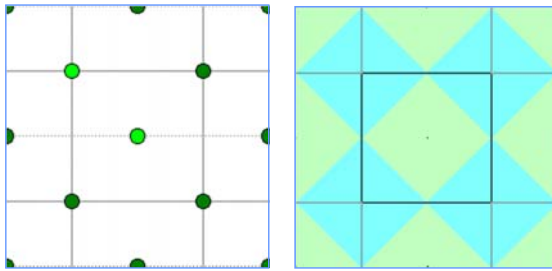
- Keine Veränderung im Redering
- Downsampling ist leicht in den RAMDAC auslagerbar

Nachteil: nur ein Texturwert (der des Makropixels) für die Subpixel **.

2.3) Antialiasing (CG) - Multisampling

► reine Kantenglättung

- Das extrem störende Aliasing bei Texturen wird in vielen Systemen bereits beim Auslesen der Werte für die Pixeltextur unterdrückt (Texturfilterung, bei modernen Systemen anisotrop, „Footprints“).
- Durchsatzserhöhung:**
 - Beschränkung auf reine Subpixel-Kantenfilterung.** Allen Subpixeln des Pixels werden (*parallel, d.h. gleichzeitig*) die gleichen Texturwerte zugeordnet (**Multisampling**)
 - Reduktion auf 2 Subpixel im rotierten Raster** (45°-Raster ist dann „optimal“, <http://www.3dcenter.org/artikel/anti-aliasing/index09.php>)



Beispiel: nVidia (GeForce-Serie) *

Multisampling:

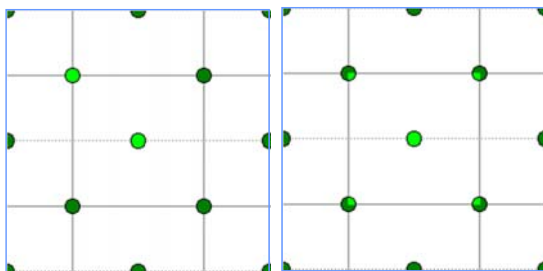
Jedes Hardware-Anti-Aliasing-Verfahren, welches nur Kantenglättung liefert, das also *für alle Subpixelpositionen gleiche (gefilterte) Texturwerte* einträgt, nennt man MULTISAMPLING

2.3) Antialiasing (CG)

2.3.c) Quincunx

► Quincunx → Überfilterung

- Ausgangspunkt ist das 2x Multisampling** mit rotated grid!
- Entsprechend dem **Grundprinzip** des Anti-Aliasing sollen **Aliasingartefakte durch Erzeugung höher aufgelöster Information reduziert** werden. Bei dem vorausgesetzten Muster ist der Informationszuwachs durch die **doppelte Anzahl von Samples gekennzeichnet, egal wie ausgewertet wird!!**



links: die zwei Subpixel bei 2x AA RG

rechts: Quincunx → Auswertung des zentralen Subpixels und seiner vier nächsten Nachbarn (Gewichte 1, 4 mal ¼)

■ Quincunx-Verfahren:

- Information für die Kantenglättung beruht nach wie vor auf 2 Subpixeln
- Entspricht einer zusätzlichen Tiefpass-Überfilterung. Durch die Unschärfe sehen Polygonkanten glatter aus.
- Harte Kanten sind gar nicht mehr möglich, dadurch wird auch „Zappeln“ an Polygonrändern reduziert.
- Aber Nachteil !!! Texturinformation, die nur zu 2 Subpixeln gehört, wird verschmiert ! ***

2.3) Antialiasing (CG)

2.3.d) Pseudozufälliges Supersampling

► Kombinationen und Pseudozufälliges Supersampling

- Grundsätzlich ist die Kombination von Supersampling, Multisampling und Quincunx möglich und einige Graphikkarten unterstützen solche Überlegungen in jeweils festgelegten Grenzen (ordered oder rotated grid, gemischtes Sampling, Downsampling-Ansatz). Der Gewinn hält sich bisher in Grenzen, bei unqualifiziertem Mix ergeben sich eher Nachteile.
- Bei Überwindung der für Füllrate und Bandbreite gegebenen Grenzen **überwiegen die Vorteile des Supersamplings**.
- Ordered grid und rotated grid weisen bei bestimmten Strukturklassen Vor- und Nachteile auf. Die **verbleibenden Artefakte** **wiederholen sich bei großen Flächen**, so dass derartige **Muster** vom Menschen oft **als störend empfunden** werden. → aus diesen und ähnlichen Überlegungen → **Ansätze für die Weiterentwicklung**:
 - Pseudozufällige Auswahl von Abtastmustern für das Supersampling
 - Nachträgliche Abminderung regelmäßiger Strukturen durch **vorsichtiges Verrauschen** (Rauschfilter)
 - Entfernungsabhängiges Antialiasing *

2.3) Antialiasing (CG) – pseudozufälliges Supersampling

■ Beispiel Radeon 8500

- Smoothvision baut auf **Subpixelblöcken mit bis zu 16 Positionen auf**.

| AA - Methode (Qualität) | Blockgröße (in Pixeln) |
|-------------------------|------------------------|
| 2x | 8 |
| 3x | 5 |
| 4x | 4 |
| 5x | 3 |
| 6x | 2 |

Beispiel für 4x AA PRG

4 Pixel mit jeweils 4
Subpixelpositionen →
16 Positionen
(Bild rechts)

- 2x AA → Subpixelblock bedient 8 Pixel (4x AA 4 Pixel usw.)
- Pro Subpixelblock wird ein Sample-Positionsmuster, bestehend aus 8 Koordinaten, zufällig gewählt, das Basis für alle Pixel im Block ist.
- Beim 2x AA werden davon 6 Koordinaten weggelassen, für jedes Pixel anders und zufällig. Für 4x entfallen 4 usw.

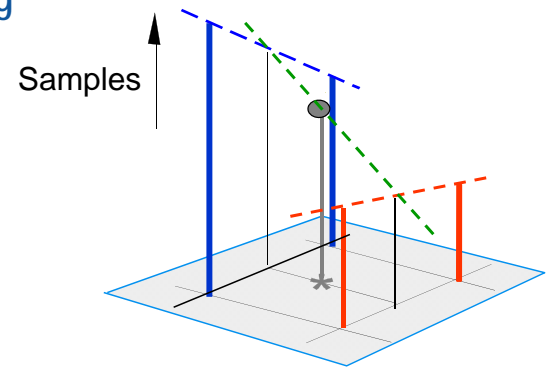


2.3) Antialiasing (CG)

2.3.e) Filterung zum Downsampling und Texturfilterung

■ abschließende Betrachtung zum Downsampling

- Beim Anti-Aliasing zur Glättung von Polygonkanten kommt für das Downsampling die Mittelwertbildung in Frage (lineare Filterung (2x – AA), bilineare Filterung (4x – AA)), gleichgültig ob ordered oder rotated grid.
- Beim pseudozufälligen Supersampling ist entsprechend der „frei gewählten“ Positionen der „Subpixel“ zu modifizieren (z.B. Gewichte nach Abstand der Abfragepositionen zur Makropixelposition). Evtl. wird vereinfachend aber auch nur der Abfrageeffekt genutzt.
- Beim Anti-Aliasing für Texturen sind, je nach Blick auf die Textur und je nach Ort viel mehr Abtastwerte ins Downsampling einzubeziehen (ortsvariante, anisotrope Filterung, siehe Abb. und Kapitel Texturen)



Beispiel: 4x AA ordered grid

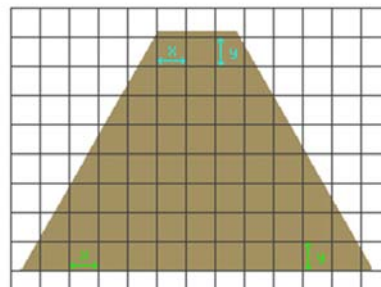
rot und blau: Subpixelwerte aus dem Supersampling oder dem Multisampling

Grau (in der Mitte): Makropixelwert

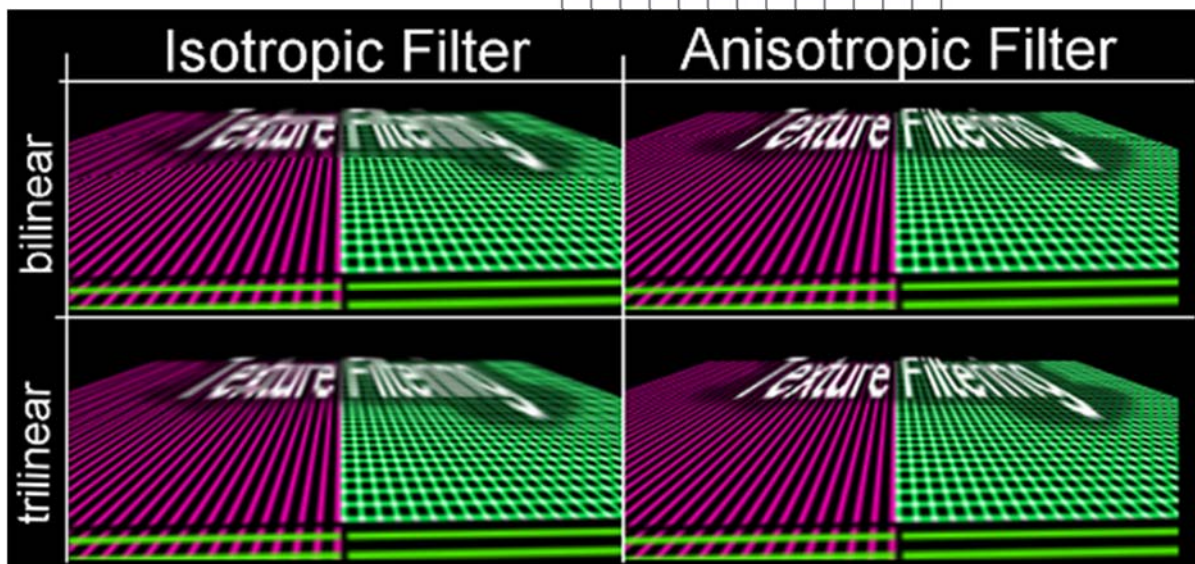


2.3) Antialiasing (CG) – Aliasingaspekt bei Texturen

- Anisotrope Filterung von Texturen (Kapitel Texturen, Prof. Brüderlin)



Bezug zwischen Raster und perspektivisch verzerrter rechteckiger Textur (symbolisch)



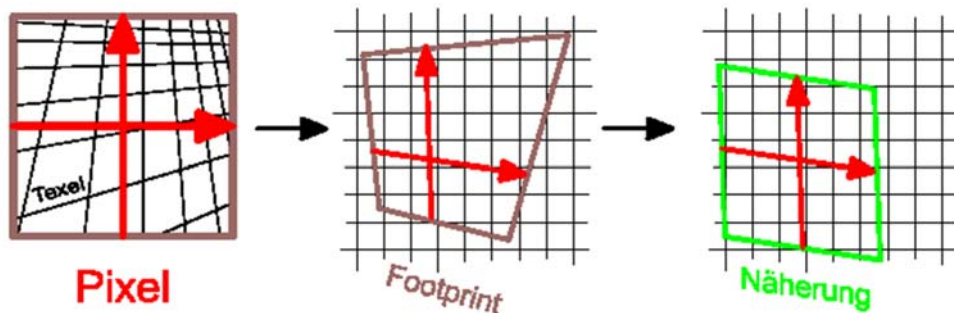
2.3) Antialiasing (CG)

■ Anisotrope Filterung von 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 oder auch Summed Area Tables kombinieren.

Ein **Footprint** ist die Projektion eines Pixels (linkes Bild) in das (ebene) Koordinatensystem der Textur. Es ergibt sich ein konvexes Viereck (Bild in der Mitte). Der integrale Farbwert ergibt sich aus den Farbwerten der Texturpixel.

Um Rechenaufwand zu sparen, wird der Footprint beim 'Footprint Assembly' durch ein Parallelogramm angenähert. Es wird über die Projektion der Pixelmitte platziert und durch die Projektion der Pixelachsen aufgespannt (Bild rechts)



Bildquelle: http://de.wikipedia.org/wiki/Footprint_Assembly