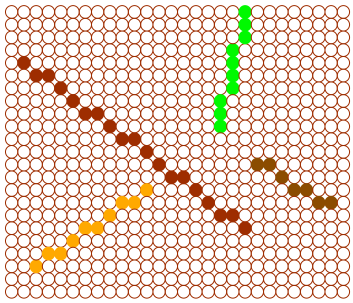
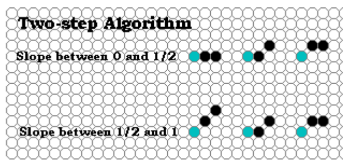
Geschwindigkeitssteigerung

- Sind rasterisierte Linien symmetrisch?
- Abhängig von der Länge:
  - Gerade # an Pixel → ja
  - Ungerade # an Pixel → ja, bis auf 1 Pixel
- Idee: zeichne von beiden Seiten [Rokne et al., 1990]
- Man kann 2 Pixel zeichnen mit:
  - 1 Vergleich
  - 1 Update der Entscheidungsvariable d
- Weiterhin: mit 1 Test kann man die nächsten 2 Pixel entscheiden [Wyvill et al., 1990]

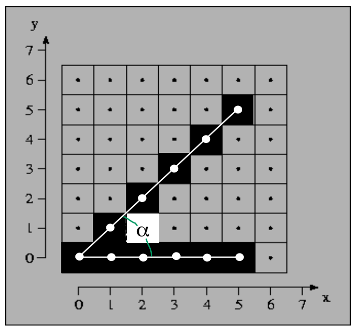




Weitere Überlegungen

- Einheitliche Stärke und Helligkeit
  - Bei gleicher Pixelzahl sind schräge Linien länger als horizontale
  - Ändere Intensität der Linie gemäß der Steigung
  - Skaliere den Grauwert um den Faktor

$$\cos(45^\circ - \alpha)\ , \quad \alpha = 0^\circ \ldots 45^\circ$$



- Was ist bei gemusterten Linien? (gestrichelte Linie, etc.)

Historische Randnotiz

- Bekannt als DDA (digital differential analyzer)

MADDIDA (Magnetic Drum Digital Differential Analyzer, Northrop Aircraft) 1952

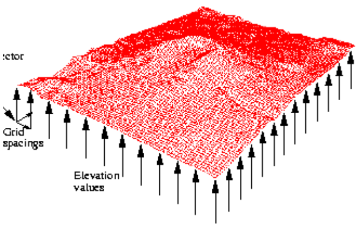
Beispiel für eine Anwendung der DDA-Technik: Ray-Tracing von Height Fields [Henning & Stephenson, 2004]

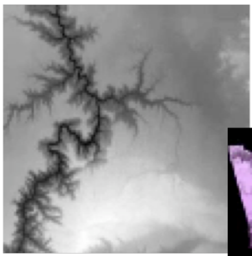
- Height Field = Alle Arten von Flächen, die sich als Funktion

$$z = f(x, y)$$

schreiben lassen

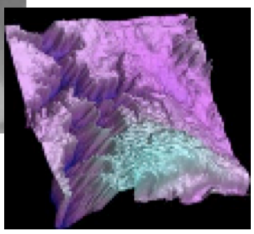
  - Z.B.: Terrain, Meßwerte über einer Ebene, 2D-Skalarfeld, …





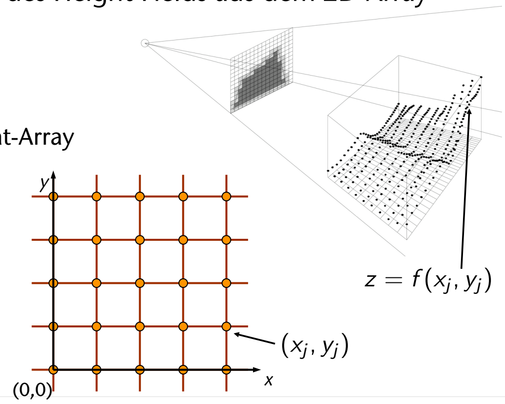
Height field (= Bitmap)

Rendered



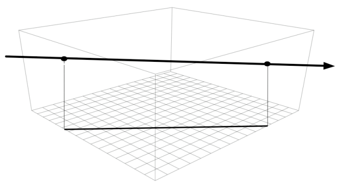
Situation

- Die naive Methode, ein Height-Field zu raytracen:
  - Konvertiere das n×n Feld in $2n^2$ Dreiecke, teste Strahl gegen jedes
  - Probleme: langsam, benötigt viel Speicher
- Ziel: direktes Ray-Tracing des Height-Fields aus dem 2D-Array
- Gegeben:
  - Strahl
  - Feld [0...n]×[0...n] als Float-Array
  - Höhenwerte liegen auf den Gitterknoten vor

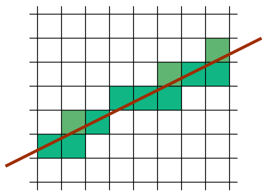


Algorithmus

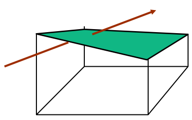
1. Dimensionsreduktion
   - Projiziere Strahl in xy-Ebene



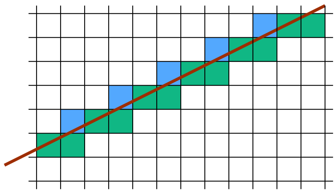
2. Alle Zellen der Reihe nach besuchen, die vom Strahl geschnitten werden (und nur diese)
   - Ähnlich zu Scan-Conversion, aber mit zusätzlichen Zellen



3. Strahl testen gegen das Flächenstück, das von den 4 Höhenwerten an den Ecken aufgespannt wird



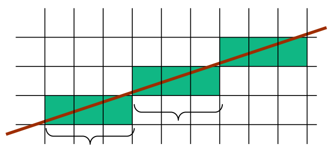
- Vereinfachungen zunächst:
  1. Betrachte (unendliche) Linien mit $y = mx$, $0 \leq m \leq 1$
  2. Betrachte nur die Folge der grünen Zellen = Zellen, die an ihrer linken Kante von der Linie geschnitten werden



- Terminologie:
  - Zelle wird identifiziert durch deren linken unteren Eckpunkt (x_j, y_j)
  - Span := Folge von Zellen mit gleicher y-Koord.
  - Länge des j-ten Spans = r_j

- Beobachtung: die diskrete Linie ist vollständig durch die Folge der Span-Längen definiert, denn

$$(x_{j+1}, y_{j+1}) = (x_j + r_j, y_j + 1)$$



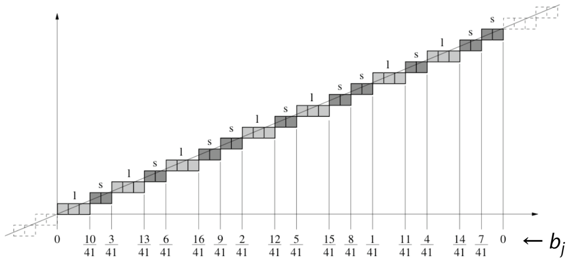
- Satz (o. Bew.):
  Alle Spans der diskretisierten Linie haben nur eine von höchstens zwei verschiedenen Längen, nämlich

$$\forall j : \; r_j = r \; \vee \; r_j = r + 1$$

- Klar ist:

$$\frac{1}{2} \leq m \leq 1 \; \Rightarrow \; r = 1$$

- Beispiel:



- Beobachtung: wenn wir ein seeehr langes Segment der Linie betrachten, dann gilt

$$\frac{\#\ \text{Spans}}{\#\ \text{Zellen}} = \frac{\Delta y}{\Delta x} \approx m$$

- Folge: aus der Steigung kann man die Span-Länge *r* (bzw. *r*+1) berechnen:

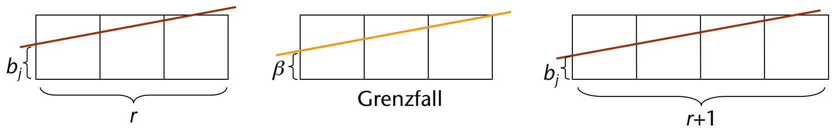
$$\frac{1}{m} = \text{ mittlere Span-Länge}$$

$$= \text{Mittelwert von } r \text{ und } r+1 \ \Rightarrow$$

$$r = \left\lfloor \frac{1}{m} \right\rfloor \ , \ r+1 = \left\lceil \frac{1}{m} \right\rceil$$

- Im Folgenden: Berechnung von r_j , m.a.W., Methode zur Entscheidung, ob man einen "langen Span" oder einen "kurzen Span" hat

- Wovon hängt es ab, ob man einen langen / kurzen Span hat?



- Fazit: falls $b_j \geq \beta$, dann kurzer Span, sonst langer Span
- Bestimmung von β: $b_j = mx_j - y_j$

$$b_{j+1} - b_j = mr_j - 1$$

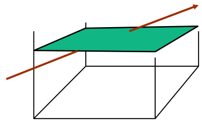
Im Grenzfall ist $b_{j+1} = 0$ und $b_j = \beta$, also

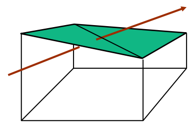
$$\beta = 1 - mr = 1 - m \left\lfloor \frac{1}{m} \right\rfloor$$

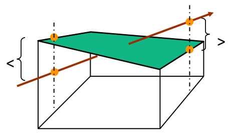
- Das nächste b_{j+1} ist also:
  - falls kurzer Span → $b_{j+1} = b_j - \beta$
  - falls langer Span → $b_{j+1} = b_j + m - \beta$
- Damit hat man einen iterativen, sehr effizienten Algo zur Aufzählung aller Zellen, die von einer Linie getroffen werden.
- Weiteres (lästiges) Detail:
  - Bei einem Strahl ist der erste Span i.A. gekürzt
  - Soll hier nicht weiter vertieft werden

Schnitttest Strahl—Flächenstück in der Zelle

- Naive Methoden:
  - "Nearest-Neighbor":
    - Bestimme mittlere Höhe aus den 4 Höhenwerten an den Ecken
    - Schneide Strahl gegen horizontales Quadrat mit dieser mittleren Höhe
    - Sehr ungenau
  - "2 Dreiecke":
    - Konstruiere 2 Dreiecke aus den 4 Punkten über den Ecken
    - Knick innerhalb der Zelle, Aufteilung in Dreiecke nicht eindeutig
- Besser: "bilineare Interpolation"
  - Betrachte Fläche als parabolisches Hyberboloid
  - Bestimme Höhe am Rand über/unter dem Strahl durch lineare Interpolation
  - Vergleiche Vorzeichen
  - Bestimmt ggf. Schnittpunkt & Normale







Speedup gegenüber einfachem DDA

- O(n) bei DDA (z.B. Midpoint),
  O(n/r) mit der Span-basierten Methode,
  n = Anzahl Zellen auf dem Strahl, r = mittlere Span-Länge
- In Zahlen:
  - Ca. Faktor 2 schneller über alle mögliche Orientierungen des Strahls

Beispiele für Terrain

Turtmann Valley Dataset

- **3 datasets** of **4k x 4k** height-samples each @ 2m planar, 0.25m vertical inter-pixel spacing
- Normal-maps derived from input height-map (**3x4096x4096**), mixed **JPEG** and **S3TC** compression
- Compressed dataset size: **33 MB**
- Flight speed is around 540 km/h ~ **Mach 0,5**

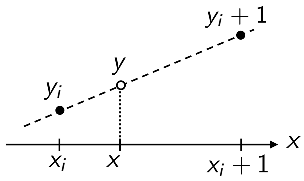
Bonn University

Valles Marineris, Mars - http://mars.jpl.nasa.gov

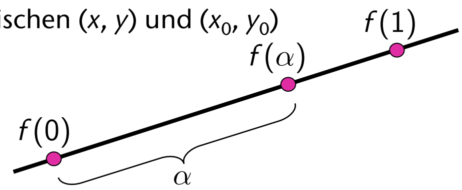
Interpolation von Attributen

- Häufig haben Eckpunkte weitere Attribute (außer der Pos.)
  - Z.B. verschiedene Farben
- Ziel: ein gleichmäßiger Farbverlauf entlang der Linie
- Idee: lineare Interpolation
- Im 1D: $f(x) = (1 - \alpha)y_0 + \alpha y_1$

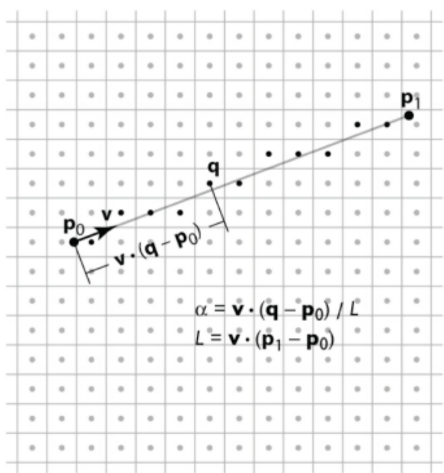
  mit $\alpha = (x - x_0)/(x_1 - x_0)$



- Im 2D ist α gerade die normierte(!) Distanz zwischen (x, y) und (x_0, y_0)



- Pixel liegen oft nicht genau auf der Linie
- Definiere 2D Funktion zur Projektion auf die Linie
  - Ist linear in x und y



- Pixel liegen oft nicht genau auf der Linie
- Definiere 2D Funktion zur Projektion auf die Linie
  - Ist linear in x und y
  - Für die Interpolation kann DDA verwendet werden