Euklidische Abstandstransformation mit Anti-Aliasing

Georg Jahn
mail@gjahn.com

nach einer Veröffentlichung von
Stefan Gustavson und Robin Strand (2010)

24. Januar 2012
Seminar Computergrafik

Euklidische Abstandstransformation mit Anti-Aliasing
Übersicht

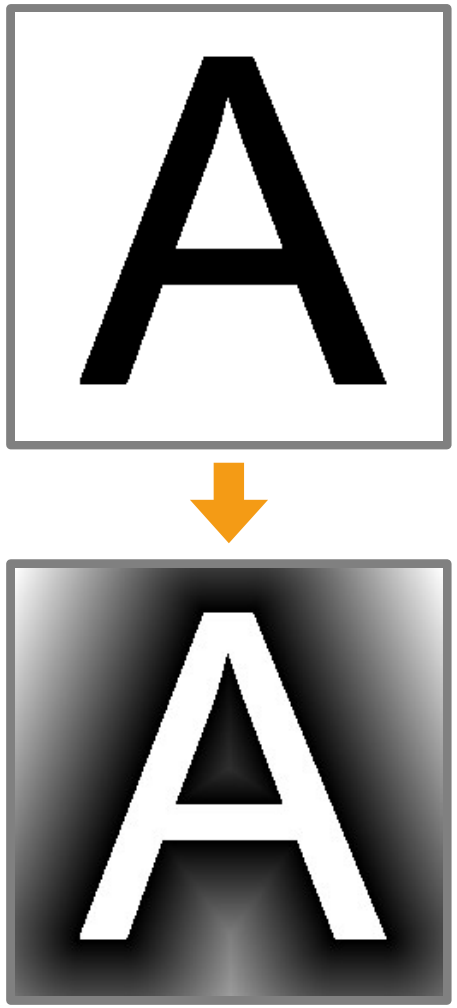
- Abstandstransformationen
  - Motivation
  - Metriken
  - Algorithmen
- Anti-Aliasing in der Abstandstransformation
  - Motivation
  - Algorithmus nach Gustavson und Strand
  - Ergebnisse

Teil 1

Pfadgenerierte und euklidische **Abstandstranformationen**

Abstandstransformationen
Definition

- Eingabe: Hintergrundmenge
  Teilmenge $A \subset \mathbb{R}^2$
- Ausgabe: Abstandsfunktion zur Menge
  $f: \mathbb{R}^2 \setminus A \rightarrow \mathbb{R}$
- Die Abstandstransformation ergibt für jeden Punkt der Vordergrundmenge den Abstand zum nächsten Punkt der Hintergrundmenge.



Abstandstransformationen
Diskrete Definition

- Eingabe: Binäres Pixelbild

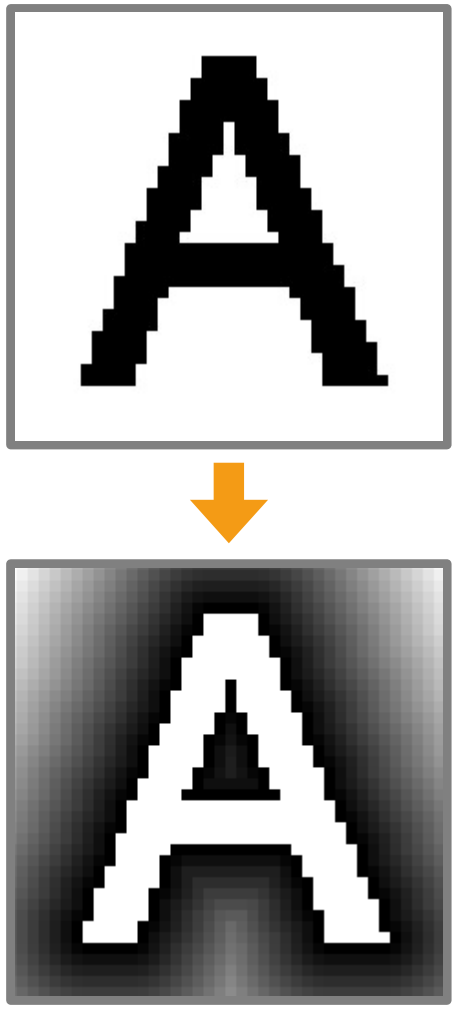
$$A \in \{0,1\}^{n \times m}$$

- Ausgabe: Abstandsfunktion zum nächsten gesetzten/ungesetzten Pixel

$$F \in \mathbb{R}^{n \times m}$$

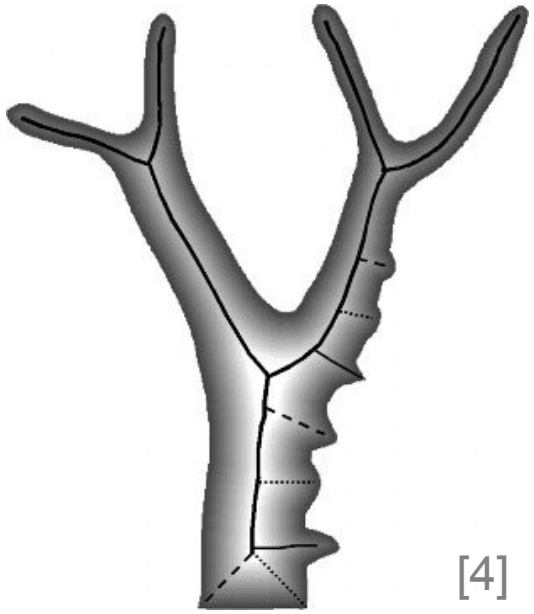
- Dabei ergibt sich der Abstand aus dem vertikalen und horizontalen Abstand entsprechend einer gewählten Metrik:

$$d[x, y]: \mathbb{Z}^2 \rightarrow \mathbb{R}$$

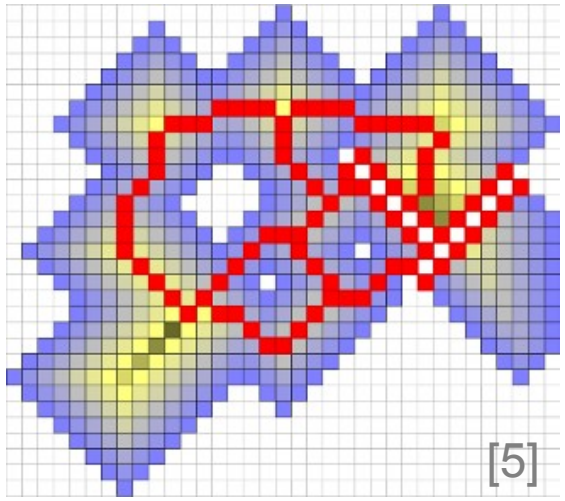


Abstandstransformationen
Motivation/Anwendungen

- Bildbearbeitung
  - Ausdünnung/Erweiterung von Formen
  - Schatten, Weichzeichnen
- Algorithmik
  - Skelettierung
  - Objekterkennung und -analyse
  - Wegfindung



[4]

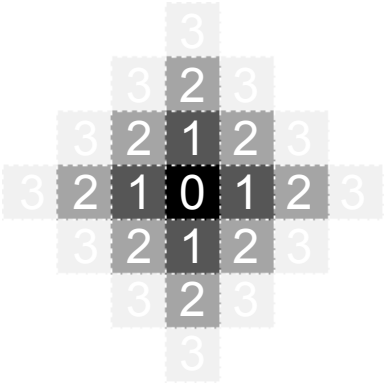


[5]

Abstandstransformationen
Metriken

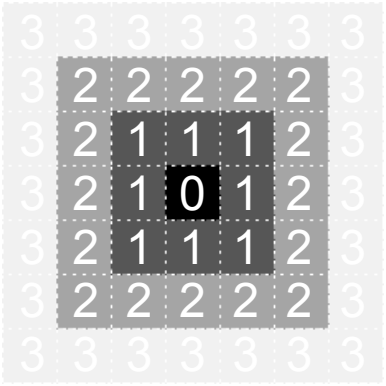
- Manhattan-Metrik

$$d(x, y) = |x| + |y|$$



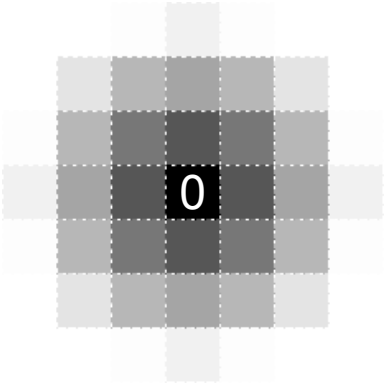
- Chessboard-Metrik

$$d(x, y) = \max(|x|, |y|)$$



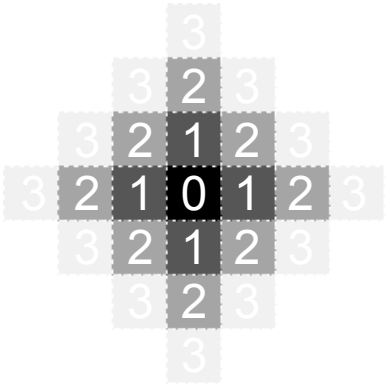
- Euklidische Metrik

$$d(x, y) = \sqrt{x^2 + y^2}$$



Abstandstransformationen
Algorithmen

- Gesucht: „Schnelle“ Algorithmen
  mit Laufzeit $O(\mathrm{n})$, n bezeichnet die Pixelzahl.
- Bei „pfadgenerierten“ Metriken, z.B. Manhattan:
  - Wert kann von umliegenden Werten her propagiert werden.
  - Propagation in maximal zwei verschiedene Richtungen.
  - Reihenfolge ist dabei egal.

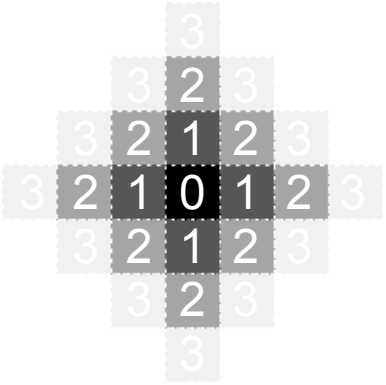


Abstandstransformationen
Metriken

Metrik ist „pfadgeneriert“

- Manhattan-Metrik

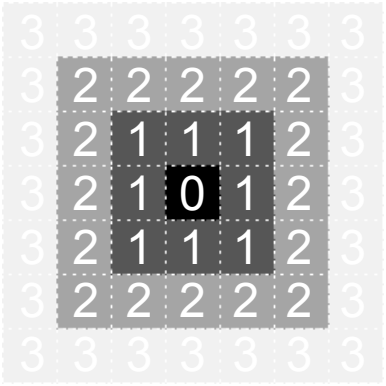
$$d(x, y) = |x| + |y|$$



✓

- Chessboard-Metrik

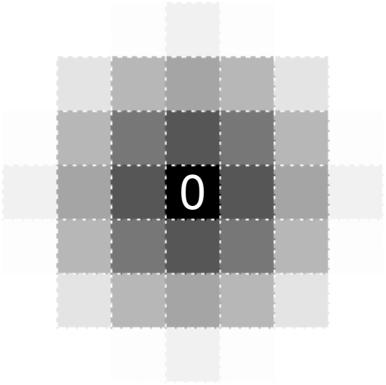
$$d(x, y) = \max(|x|, |y|)$$



✓

- Euklidische Metrik

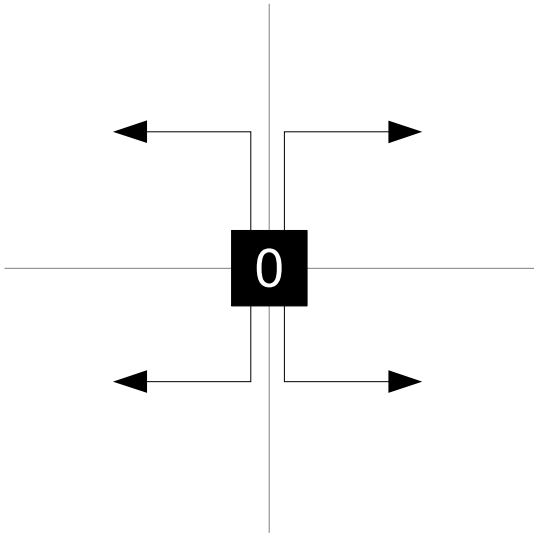
$$d(x, y) = \sqrt{x^2 + y^2}$$



✗

Algorithmen zur Abstandstransformation
Rosenfeld und Pfaltz (1968)

- Eigenschaften der Pfadgeneration erlauben einfachen sequentiellen Algorithmus:
  - Initialisierung mit Abstand 0 bzw. ∞
  - Propagation der Werte in die je zwei Richtungen der vier Quadranten einzeln.
  - Bestimme Minimum der vier Suchläufe.



- Art der Suchläufe kann noch leicht optimiert werden, wichtig jedoch: $O(\mathrm{n})$
- Jedoch nicht für nicht-pfadgenerierte Metriken!

Algorithmen zur Abstandstransformation
Fast-Marching

- Idee: Die Eikonalgleichung beschreibt die Ankunftszeit eines Lichtstrahls, der sich in alle Richtungen ausbreitet:

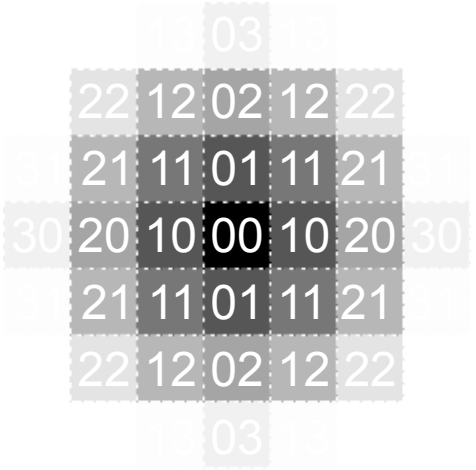
$$(\nabla \phi)^2 = \frac{1}{c^2}$$

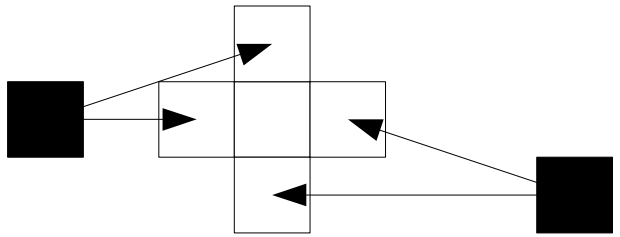
- Numerische Lösung mit Laufzeit $O(\mathrm{n})$ von Sethian (1999)
  - Um das Bild nicht n-mal abscannen zu müssen: Einteilung in Punkttypen „tot“, „Grenze“, „weit weg“

→ Gute Ergebnisse zur euklidischen Abstandstransformation (EAT).

Algorithmen zur Abstandstransformation
SED4

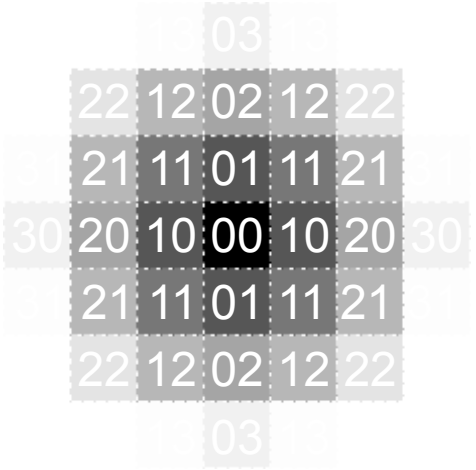
- Danielsson schlägt 1980 die Vektor-Propagation anstatt der Wert-Propagation vor.
- Der sequentielle Algorithmus lässt sich damit auf die euklidische Metrik übertragen.
- Annahme dabei: Vektor eines Pixels hängt nur von seinen vier nächsten Nachbarn ab.

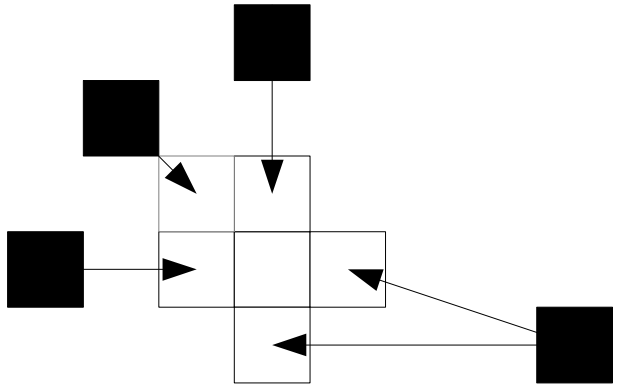




Algorithmen zur Abstandstransformation
SED4

- Danielsson schlägt 1980 die Vektor-Propagation anstatt der Wert-Propagation vor.
- Der sequentielle Algorithmus lässt sich damit auf die euklidische Metrik übertragen.
- Annahme dabei: Vektor eines Pixels hängt nur von seinen vier nächsten Nachbarn ab.
  - Falsch! Gegenbeispiel:
    Maximaler Fehler ist ca. 0,29 Pixel.





Algorithmen zur Abstandstransformation
SED8 & mehr

- SED8 betrachtet sogar die 8 umliegenden Pixel.
- Es gibt noch immer fehlerhafte Situationen.
  - Maximaler Fehler beträgt etwa 0,027 Pixel.
- Kann theoretisch zur Fehlerbeseitigung stets erweitert werden, z. B. SED16.

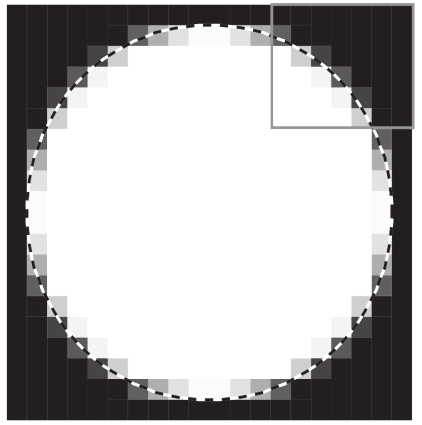
- Vorteile/Nachteile bei der Vektor-Propagation:
  - ✔ Algorithmen sind einfach.
  - ✔ Nicht nur der Abstand, sondern auch Vektor zum nächsten Pixel ist verfügbar.
  - ✗ Algorithmen sind speicherintensiver.

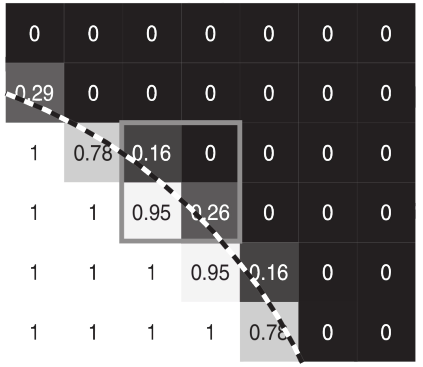
Teil 2

Euklidische Abstandstransformationen **mit Anti-Aliasing**

Anti-Aliasing in der Abstandstransformation
Motivation

- Daten aus der realen Welt sind nicht binär.
- Formen aus der realen Welt lassen sich nur schlecht auf einem Gitter approximieren.





Anti-Aliasing in der Abstandstransformation
Motivation

- Daten aus der realen Welt sind nicht binär.
- Formen aus der realen Welt lassen sich nur schlecht auf einem Gitter approximieren.
- Die resultierenden Ungenauigkeiten zeigen sich vor allem bei der weiteren Verwendung:

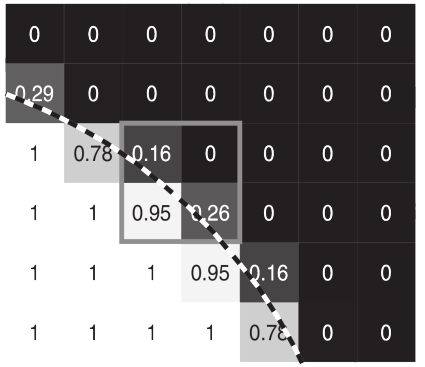
Original | Fehler bei der EAT | Differential der EAT

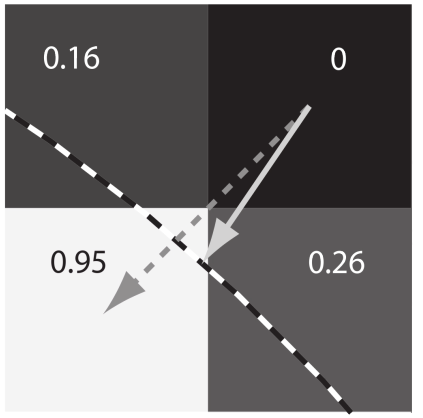
Anti-Aliasing in der Abstandstransformation
Idee

- Vorgeschlagene Abstandstransformation:

  Operiert auf nicht-binären Pixeln, d. h. im Intervall [0, 1] (abgetastete Form).

  Vektoren zeigen nicht mehr zum Mittelpunkt der Pixel, sondern zum geschätzten Rand.





- Die Krümmung der Form wird als klein gegenüber der Pixelgröße angenommen.
- So soll eine Genauigkeit weit unter der Pixelgröße erreicht werden.

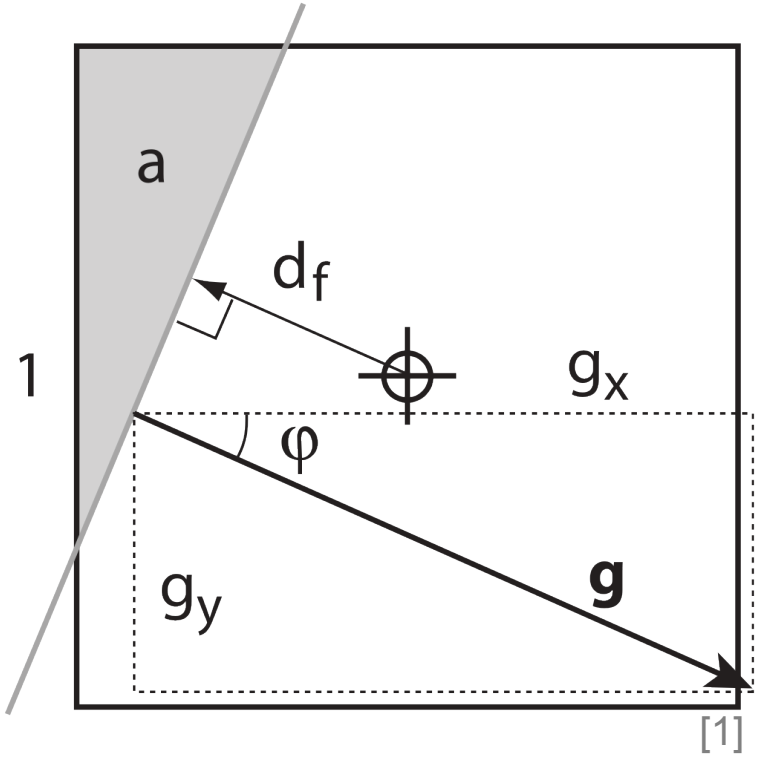
Anti-Aliasing in der Abstandstransformation
Konstruktionen

- Ziel:

  Bei gegebener Pixelfüllung und gegebener Kantenrichtung soll die Position der Kante berechnet werden.

- In der Skizze:

  Ein Zusammenhang zwischen d_f und a (in Abhängigkeit von φ)



[1]

Anti-Aliasing in der Abstandstransformation
Konstruktionen

- Zwischenergebnisse:

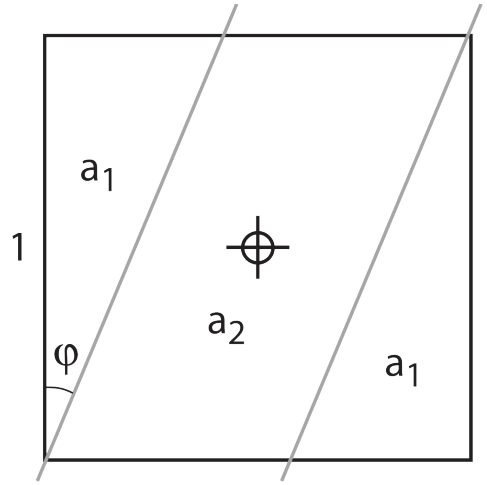
$$a_1 = \tan\phi$$

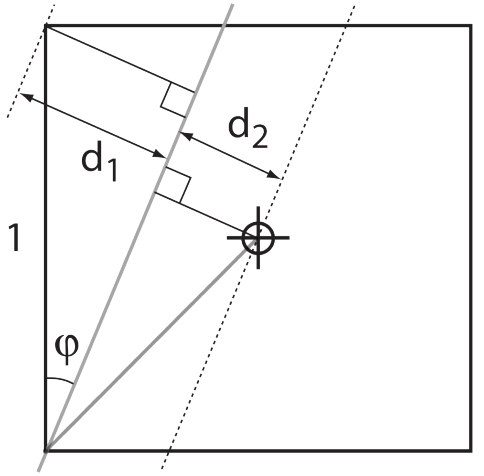
$$a_2 = 1 - 2a_1$$

$$d_1 = \sin\phi = g_y$$

$$d_2 = (1/\sqrt{2})\sin(\pi/4 - \phi)$$

$$a = \begin{cases} 0, & d_f \le -d_1 - d_2 \\ (d_1 + d_2 + d_f)^2 (a_1/d_1^2), & -d_1 - d_2 \le d_f \le -d_2 \\ a_1 + (a_2/2)(1 + d_f/d_2), & -d_2 \le d_f \le d_2 \\ 1 - (d_1 + d_2 - d_f)^2 (a_1/d_1^2), & d_2 \le d_f \le d_1 + d_2 \\ 1, & d_f \ge d_1 + d_2 \end{cases}$$





Anti-Aliasing in der Abstandstransformation
Konstruktionen

- Zwischenergebnisse:

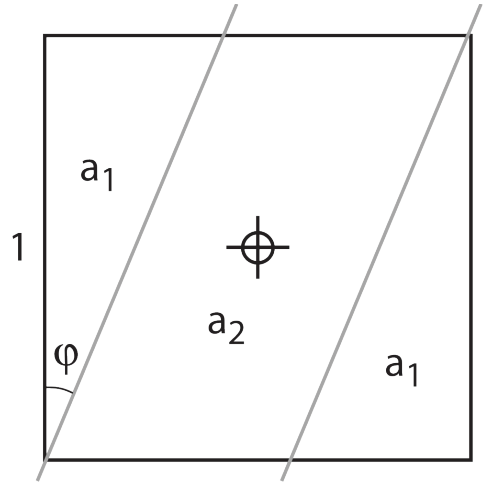
$$a_1 = \tan\phi$$

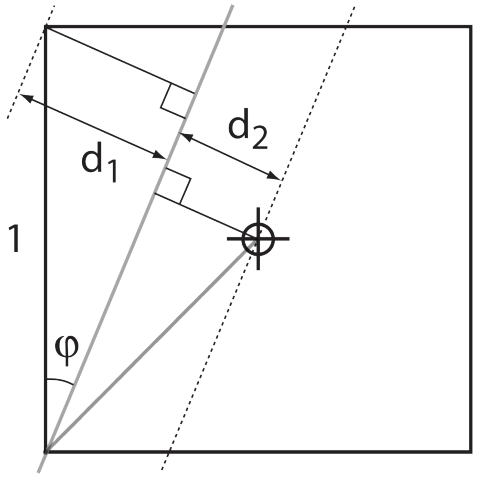
$$a_2 = 1 - 2a_1$$

$$d_1 = \sin\phi = g_y$$

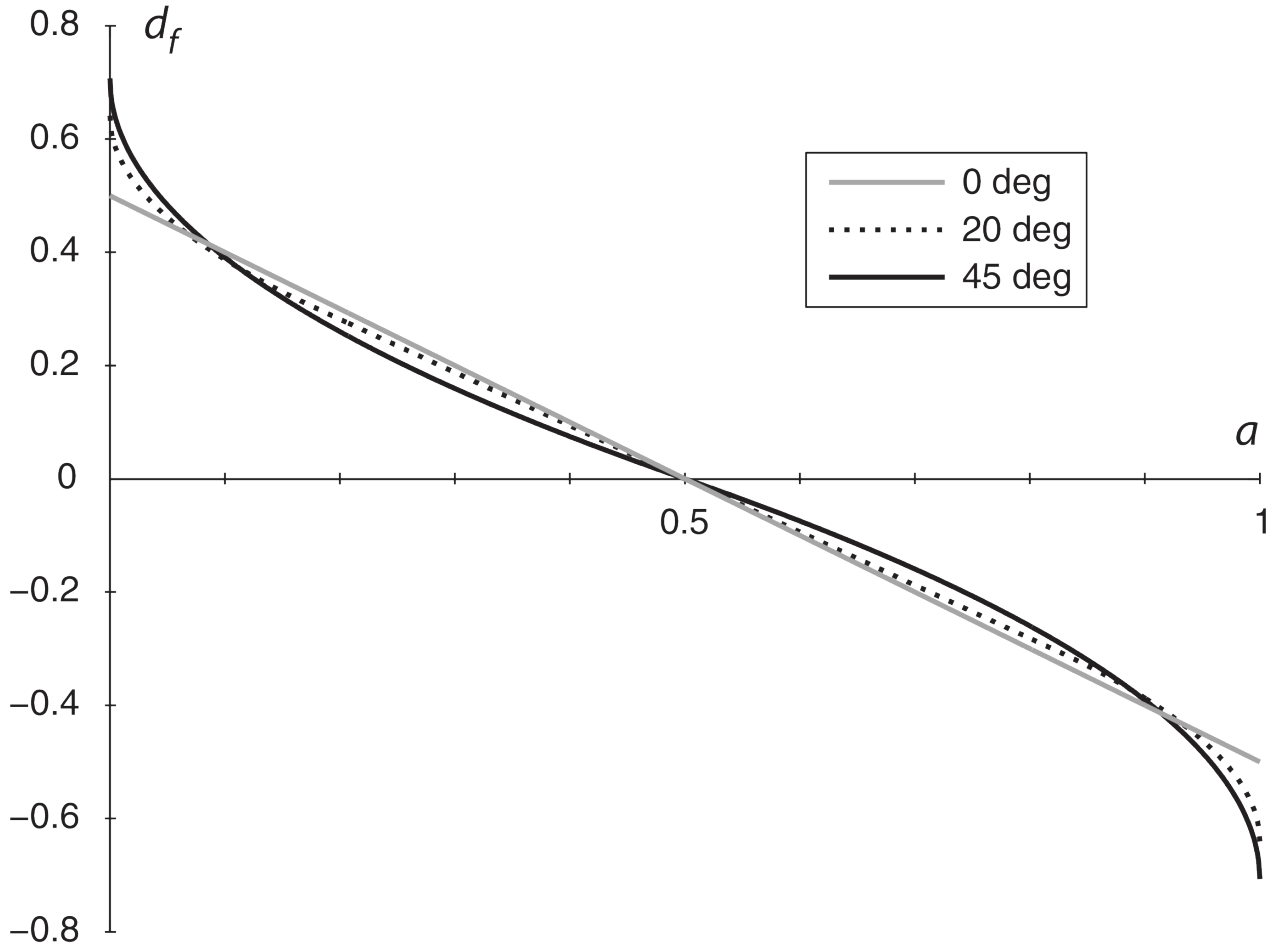
$$d_2 = (1/\sqrt{2})\sin(\pi/4 - \phi)$$

$$d_f = \begin{cases} (g_x + g_y)/2 - \sqrt{2 g_x g_y a}, & 0 \le a \le a_1 \\ (0.5 - a)\, g_x & a_1 \le a \le 1 - a_1 \\ -(g_x + g_y)/2 + \sqrt{2 g_x g_y (1-a)} & 1 - a_1 \le a \le 1 \end{cases}$$





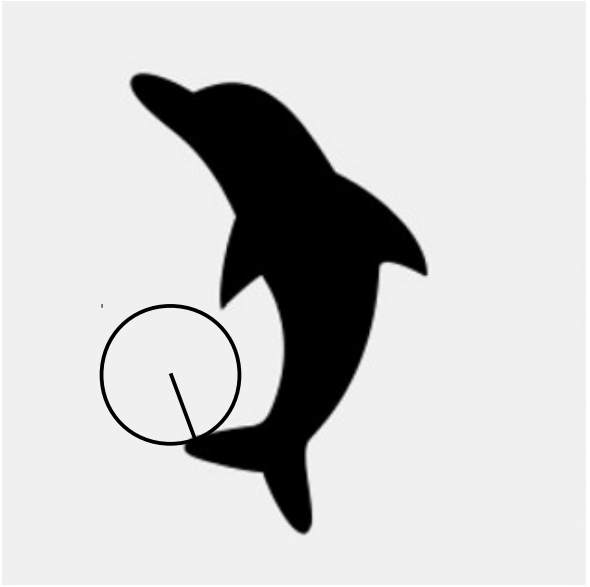
Anti-Aliasing in der Abstandstransformation
Konstruktionen



- Lässt sich gut linear approximieren, wenn φ unbekannt.

Anti-Aliasing in der Abstandstransformation
Der Algorithmus von S. Gustavson und R. Strand

- Benutze neues Abstandsmaß als $d(x, y, \varphi)$ bei einem Vektor-Propagations-Algorithmus (z. B. SED8).
- Wie bestimmt man nun φ?
  - Weit weg von der Kante:
    Kante ist etwa senkrecht zur Verbindungslinie.
    (Dies ist asymptotisch korrekt.)



  - Nah an der Kante:
    g_x und g_y werden mithilfe von zwei 3×3-Gradientenfiltern abgeschätzt.

Anti-Aliasing in der Abstandstransformation
Ergebnisse

- Der Algorithmus erreicht bei einer Reihe von Testbildern eine durchschnittliche Genauigkeit von 0,02 Pixel.
- An harten Ecken Fehler bis zu ca. 0,5 Pixel.
- Auswirkung auf die Geschwindigkeit:

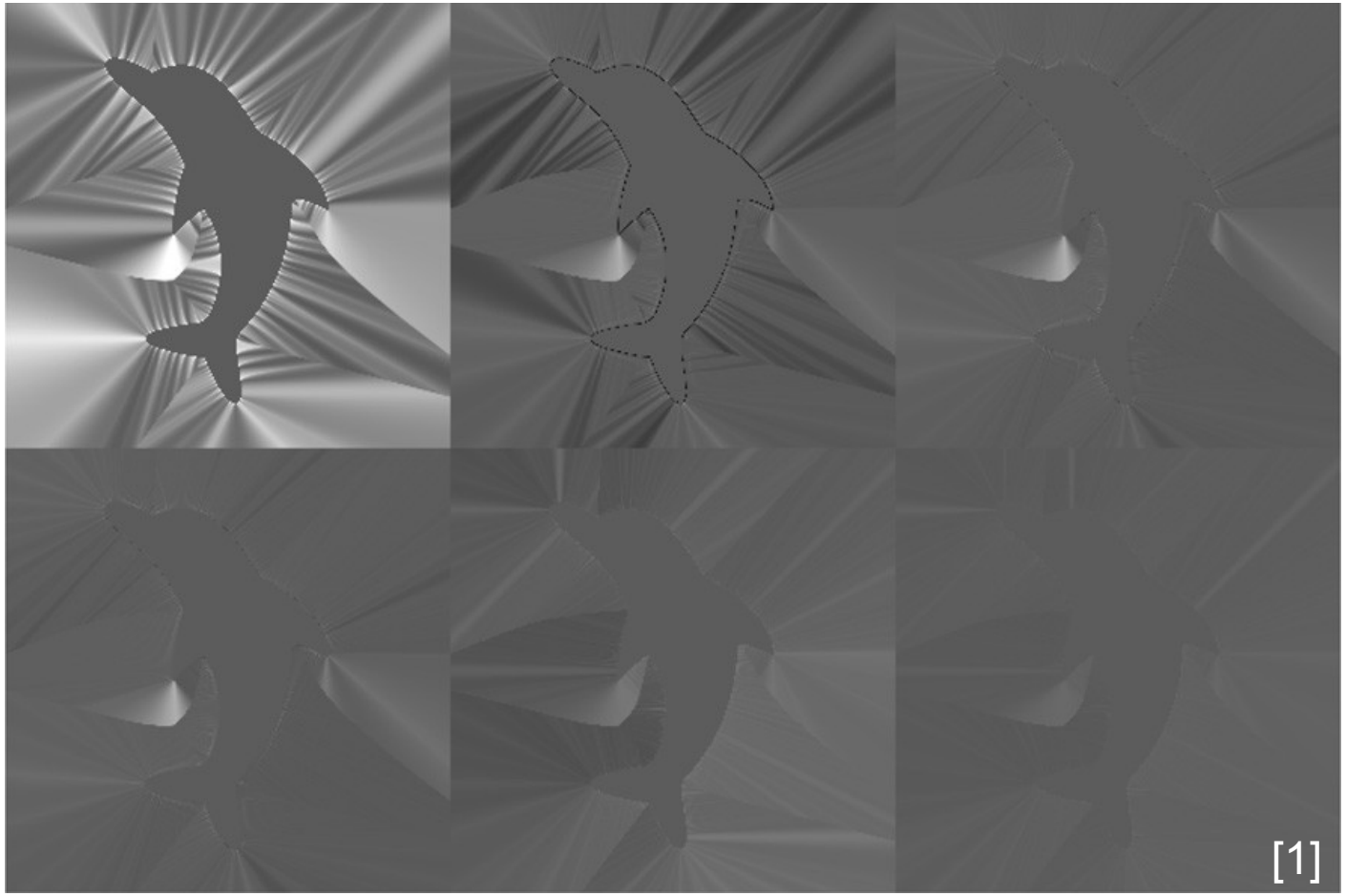
Binäre EAT	AA $\varphi = 0$	AA $\varphi =$ senkr.	AA vollst.
0.33 s	0.83 s	3.2 s	3.9 s
1.0×	2.5×	9.7×	11.8×

- Optimierungen:
  - Tabellierung der $d_f(a, \varphi)$-Funktion
  - Frühes Aussortieren (early-out) durch $-1 < d_f < 1$

Anti-Aliasing in der Abstandstransformation
Vergleich mit herkömmlicher Methode

Auflösung: 1fach 8fach 16fach

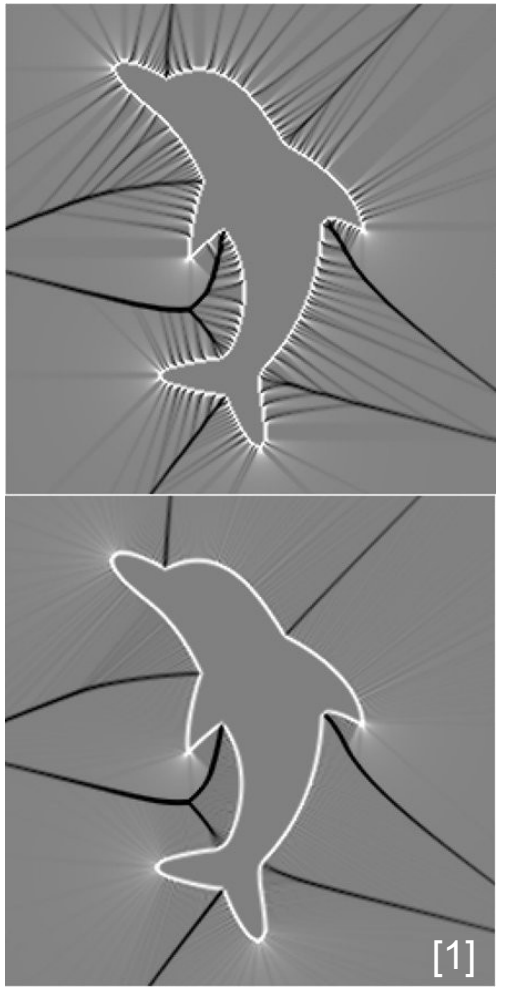
- EAT normal
- EAT mit AA



φ = 0 φ = senkr. φ voll berechnet

Ausblick
Anwendungsmöglichkeiten

- Skelettierung, da das Differential deutlich besser ist (siehe rechts).
- Subpixel-Ausdünnung/-Erweiterung.
- Echtzeit-Rekonstruktion von scharfen Kanten aus Bildern.
- Alles Weitere, bei dem das Differential der EAT erforderlich ist.
- Ersparnis von Rechenzeit und Speicherkapazität.


[1]

Ausblick
Literatur und Quellen

[1] Anti-Aliased Euclidean Distance Tranform
Stefan Gustavson und Robin Strand, Pattern Recognition Letters, 2011

[2] Euclidean Distance Mapping
Per-Erik Danielsson, Computer Graphics and Image Processing, 1980

[3] A fast marching level set method for advancing fronts
James Albert Sethian, Proc. Natl. Acad. Sci. USA, 1996

[4] New Vessel Analysis Tool for Morphometric Quantification
http://radiographics.rsna.org/content/24/1/287/F5.expansion.html

[5] Shape-Thinning
http://liris.cnrs.fr/dgtal/gallery/shape-thinning-8-4/

Die Internetquellen wurden am 22.01.2012 überprüft.

Gibt es noch Fragen?

Georg Jahn
mail@gjahn.com