

Ausarbeitung  
zum Thema

# **Fréchet-Distanzen zum Messen der Ähnlichkeit von Polygonen**

im Rahmen des  
Informatik-Seminars  
WS 2011/12

erstellt von  
**Tobias Möllmann**

# Inhaltsverzeichnis

1 Einleitung.....	3
2 Definitionen.....	4
3 Die Hausdorff-Distanzfunktion.....	5
4 Die Fréchet-Distanzfunktion.....	6
5 Berechnung der Fréchet-Distanz.....	7
5.1 Aufbau des Algorithmus.....	7
5.2 Berechnung des Entscheidungsproblems.....	7
5.3 Berechnung der Fréchet-Distanz.....	10
6 Fazit.....	12
Literaturverzeichnis.....	13

## 1 Einleitung

In vielen Anwendungen spielt die Erkennung von Objekten eine entscheidende Rolle. Einige Beispiele dafür sind in der Erkennung von handgeschriebenen Ziffern, beim Morphing oder der Messung der Ähnlichkeit von Proteinstrukturen zu finden. Ein entscheidender Ansatz bei der Erkennung von Objekten ist dabei der Abgleich von Objektformen.

Eine Form ist dabei ein geometrisches Modell, welches aus Punkten, Kurven oder Flächen besteht. Generell werden Kanten detektiert, um die Formen der in einem Bild vorkommenden Objekte zu ermitteln.

Das Shape Matching beschäftigt sich mit der Frage, wie die Ähnlichkeit von Formen bestimmt werden kann. Ein Ansatz dabei ist ein Distanzmaß, welches die Unterschiede der Formen ausdrückt. Im folgenden Kapitel wird daher definiert, was ein Distanzmaß ist und welche wesentlichen Konzepte es im Zusammenhang damit gibt. Zwei solche Distanzmaße sind die Hausdorff-Distanz sowie die Fréchet-Distanz, die in den Kapiteln drei und vier vorgestellt werden. Es wird darauf eingegangen, in welchen Fällen die Nutzung der Hausdorff-Distanz gegenüber der Fréchet-Distanz eher problematisch sein kann. Im fünften Kapitel wird ein im Jahr 1995 von H. Alt und M. Godau erstmalig veröffentlichter Algorithmus vorgestellt, der als erster seiner Art die Fréchet-Distanz in polynomieller Zeit berechnet. Im abschließenden Kapitel folgt ein Fazit zu diesem Thema.

## 2 Definitionen

Distanzfunktionen vergleichen jeweils die Feature-Werte zweier Objekte und drücken das Ergebnis durch einen reellen Wert aus:

$$d : O \times O \rightarrow \mathbb{R}$$

$O$ : Menge von Objekten

Eine Distanzfunktion  $d$  heißt Metrik auf  $O$ , wenn die Funktion folgende Eigenschaften aufweist:

- Positivität:  $d(x, y) \geq 0$
- Selbstidentität:  $d(x, y) = 0 \Leftrightarrow x = y$   
Eine Distanz von 0 wird als maximale Ähnlichkeit aufgefasst.
- Symmetrie:  $d(x, y) = d(y, x)$
- Dreiecksungleichung:  $d(x, y) \leq d(x, z) + d(z, y)$

Es wird von einem metrischen Raum  $(O, d)$  gesprochen.

Hier zwei einfache Beispiele für Distanzfunktionen mit Metrikeigenschaften:

- Eine Distanzfunktion auf der Menge der reellen Zahlen  $\mathbb{R}$  ist der absolute Betrag der Differenz zweier Zahlen:

$$d_{abs} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_0^+ \text{ mit } d_{abs}(x, y) \rightarrow |x - y|$$

- Die euklidische Distanzfunktion ist auf Tupel der Menge  $\mathbb{R}^n$  gegeben durch:

$$d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_0^+ \text{ mit } d(x, y) \rightarrow \sqrt{\sum_{i=1}^n (x[i] - y[i])^2}$$

Die im nachfolgenden behandelte Haudorff-Distanzfunktion sowie die Fréchet-Distanzfunktion besitzen ebenfalls diese Metrikeigenschaften.

Ein weiteres Konzept im Zusammenhang der Berechnung der Ähnlichkeit von Objekten ist die Invarianz. Durch sie wird ausgedrückt, welche Merkmale zur Berechnung des Ähnlichkeitsmaßes nicht herangezogen werden sollen. Ein Ähnlichkeitsmaß ist invariant gegenüber einer Gruppe  $G$ , wenn für alle Transformationen  $g$  aus  $G$  gilt:  $d(g(x), y) = d(x, g(y)) = d(x, y)$

Beispiele für mögliche invariante Operationen sind:

- Translation
- Skalierung
- Rotation

Der in dieser Arbeit vorgestellte Algorithmus zur Berechnung der Fréchet-Distanz besitzt keine dieser Invarianzen. Allerdings wird darauf eingegangen, wie sich die Berücksichtigung einiger Invarianzen im Laufzeitverhalten des Algorithmus auswirken.

### 3 Die Hausdorff-Distanzfunktion

Die Hausdorff-Distanz  $d_H$  berechnet die Distanz zwischen zwei endlichen Mengen  $A$  und  $B$ , die Teilmengen einer Objektmenge  $O$  sind. Zwischen zwei Elementen sei die Grunddistanz  $d$  definiert. Die Distanzfunktion ist folgendermaßen definiert:

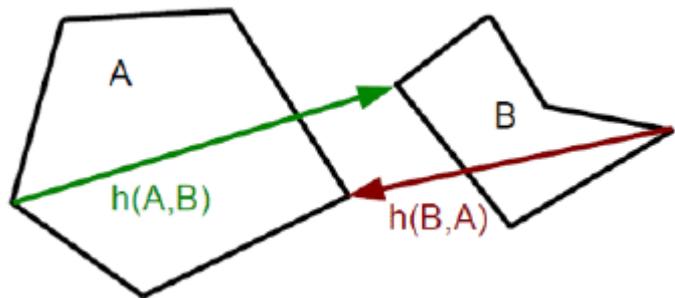
$$d_H(A, B) = \max(h(A, B), h(B, A))$$

$$\text{mit } h(X, Y) = \sup_{x \in X} \inf_{y \in Y} d(x, y)$$

$h(X, Y)$  wird als gerichtete Hausdorff-Distanz bezeichnet. Sie berechnet die maximale Distanz eines Elements der Menge  $X$  zum jeweils nächsten Element der Menge  $Y$ .

**Beispiel:**

Die nebenstehende Abbildung zeigt zwei durch Polygone beschriebene Mengen in  $\mathbb{R}^2$ . Die euklidische Distanz sei als Distanzmaß gewählt. Es sind die jeweiligen gerichteten Hausdorff-Distanzen eingezeichnet. Die gesamte Hausdorff-Distanz entspricht dem Maximum dieser Werte.

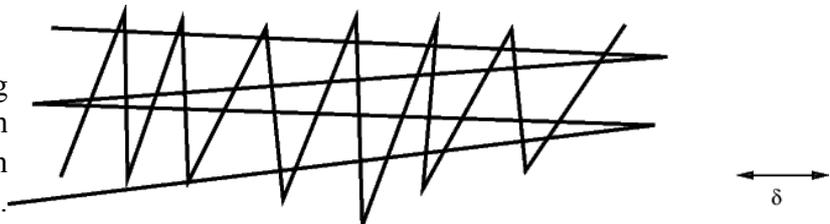


Quelle: [http://www.wolfgang-bueschel.de/wp-content/plugins/downloads-manager/upload/distance\\_measures.pdf](http://www.wolfgang-bueschel.de/wp-content/plugins/downloads-manager/upload/distance_measures.pdf)

Der Vorteil dieser Distanzfunktion liegt in der breiten Anwendbarkeit, da außer der Einschränkung, dass beide Inputmengen endlich sein müssen, keine weitere Restriktion existiert. Allerdings ist es in einigen Anwendungsfällen problematisch, dass die Wertepaare komplett unabhängig betrachtet werden. Eine Ordnung der Elemente wird nicht berücksichtigt.

**Beispiel:**

Nebenstehende Abbildung zeigt zwei Polygonzüge im  $\mathbb{R}^2$  mit euklidischem Abstand als Distanzmaß.



Die Formen sind sehr unterschiedlich. Dennoch ist die Hausdorff-Distanz wegen des Übereinanderliegens der Polygonzüge relativ gering. Dies liegt daran, dass die Polygonkurven als Mengen ohne Ordnung der Elemente betrachtet werden.

## 4 Die Fréchet-Distanzfunktion

Eine Distanz-Funktion, die dieses Problem besser löst ist die Fréchet-Distanz.

In der Literatur wird die Fréchet-Distanz durch folgendes anschauliches Beispiel erklärt:

Ein Mann und ein Hund bewegen sich auf eigenen nicht geschlossenen Kurven vorwärts. Sie können unterschiedliche Geschwindigkeiten annehmen, dürfen sich aber nicht rückwärts bewegen. Die Fréchet-Distanz entspricht der minimalen Länge einer Hundeleine, die notwendig ist, damit beide ihren gesamten Weg ablaufen können.

Die Fréchet-Distanz ist allgemein auf parametrisierbaren stetigen Kurven definiert, damit ist das Anwendungsgebiet gleichzeitig sehr eingeschränkt. Hier werden nur nicht geschlossene Kurven wie in dem Beispiel behandelt, die als Funktionen über einen Parameter definiert sind. Durch diese Parametrisierung sind die Punkte der Kurve implizit geordnet.

Eine Kurve ist dabei ein kontinuierliches Mapping:

$$f: [a, b] \rightarrow V$$

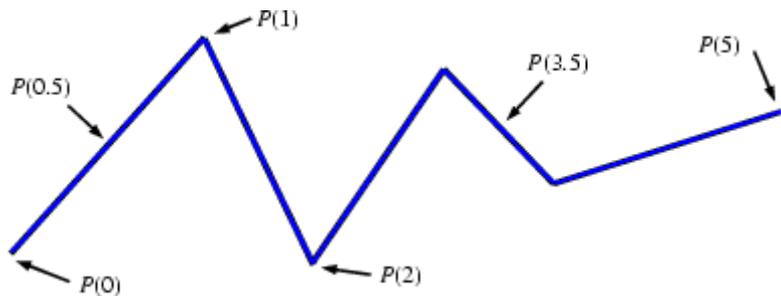
mit  $a, b \in \mathbb{R}$ ,  $a < b$  und  $V$  als unabhängigen euklidischen Vektorraum.

Ein Polygonzug/eine Polygonkurve ist eine Kurve  $P$ :

$$P: [0, n] \rightarrow V$$

mit  $n \in \mathbb{N}$ ,  $n$  wird als Länge des Polygonzugs bezeichnet.

Ein Polygonzug  $P$  kann mit einem Parameter  $a \in \mathbb{R}$  parametrisiert werden, mit  $P(0)$  als ersten und  $P(n)$  als letzten Vertex. Rechts ist ein Beispiel für einen solchen Polygonzug dargestellt.



Quelle: <http://www.cim.mcgill.ca/~stephane/cs507/param.gif>

Seien folgende Kurven gegeben:

$$f: [a, a'] \rightarrow V$$

$$g: [b, b'] \rightarrow V$$

So ist die Fréchet-Distanz folgendermaßen definiert:

$$d_F := \min_{\substack{\alpha: [0,1] \rightarrow [a,a'] \\ \beta: [0,1] \rightarrow [b,b']}} \max_{t \in [0,1]} \|f(\alpha(t)) - g(\beta(t))\|$$

mit  $\alpha, \beta$  als kontinuierliche und steigende Funktionen mit  $\alpha(0) = a$ ,  $\alpha(1) = a'$ ,  $\beta(0) = b$  und  $\beta(1) = b'$

## 5 Berechnung der Fréchet-Distanz

### 5.1 Aufbau des Algorithmus

Im folgenden soll ein Algorithmus zur Berechnung der Fréchet-Distanz vorgestellt werden. Dieser Algorithmus wurde erstmalig im Jahr 1995 von H. Alt und M. Godau veröffentlicht. Es ist der erste Algorithmus, der das Problem der Berechnung der Fréchet-Distanz mit Polynomialzeit-Verhalten lösen konnte. Die Laufzeit des Algorithmus zur Berechnung der Distanz zweier Polygonzüge  $P$  und  $Q$  mit  $p$  bzw.  $q$  Segmenten beläuft sich auf  $O(pq \cdot \log(pq))$ . Für zusätzliche Berücksichtigung der Translationsinvarianz existiert ein Algorithmus mit der Laufzeit  $O((pq)^3(p+q) \cdot \log(p+q))$ . Dieser soll hier jedoch nicht betrachtet werden, sondern nur der Basialgorithmus.

In diesem Algorithmus wird die Fréchet-Distanz ausgehend von dem Entscheidungsproblems berechnet. Die Laufzeit zur Lösung des Entscheidungsproblems ist  $O(pq)$ .

### 5.2 Berechnung des Entscheidungsproblems

Das Problem der Fréchet-Distanz als Entscheidungsproblem kann wie folgt aufgefasst werden:

Gegeben: Polygonzüge  $P$  und  $Q$  und ein  $\varepsilon \geq 0$

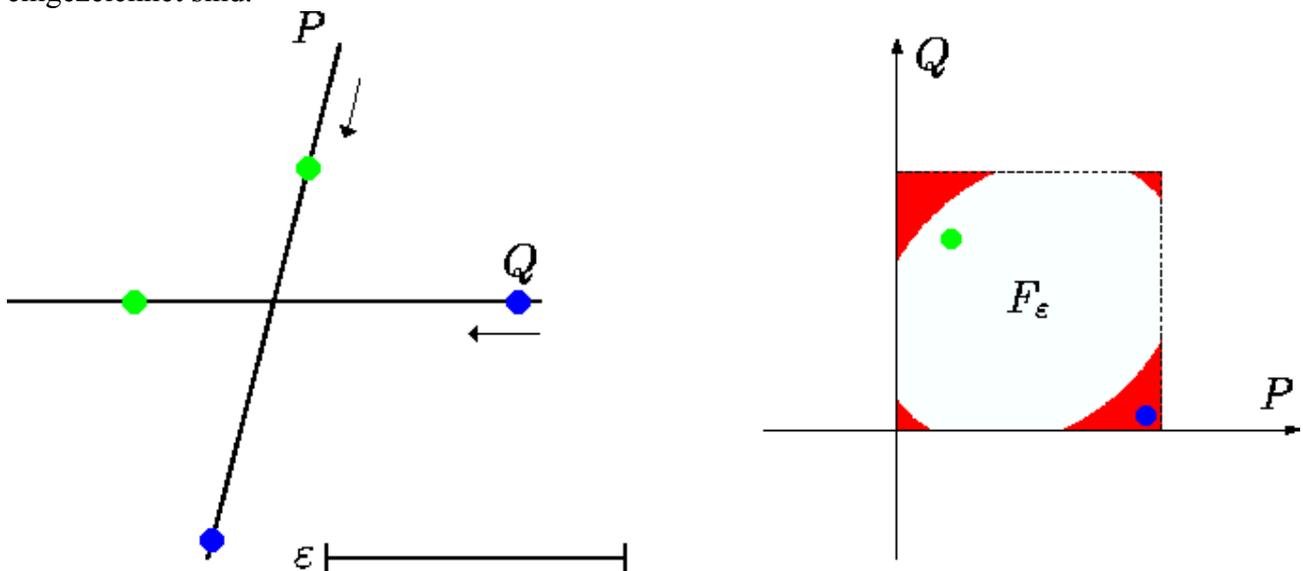
mit  $P:[0,p] \rightarrow V$  und  $Q:[0,q] \rightarrow V$

Frage: Gilt  $d_F(P,Q) \leq \varepsilon$ ?

Zunächst soll angenommen werden, dass beide Polygonzüge aus je einer Kante bestehen, also  $p = q = 1$ . Dann definieren wir:

$$F_\varepsilon := \{(s,t) \in [0,1]^2 \mid d(P(s),Q(t)) \leq \varepsilon\}$$

$F_\varepsilon$  ist die Menge, die alle Punktepaare beinhaltet, einer auf  $P$ , einer auf  $Q$ , deren Distanz höchstens  $\varepsilon$  beträgt.  $F_\varepsilon$  wird als Freiraum bezeichnet. Nachfolgend ist eine Abbildung aufgeführt, in der beide Polygonzüge  $P$  und  $Q$ , ein  $\varepsilon$  und  $F_\varepsilon$  als weiße Fläche im Einheitsquadrat eingezeichnet sind.



Quelle: [http://www.cim.mcgill.ca/~stephane/cs507/free\\_cell\\_fig1.gif](http://www.cim.mcgill.ca/~stephane/cs507/free_cell_fig1.gif)

Die beiden grünen Punkte auf den Polygonen  $P$  und  $Q$  haben einen Abstand, der kleiner als  $\varepsilon$  ist, wie dem Freiraum-Diagramm entnommen werden kann. Die beiden blauen Punkte haben jedoch einen größeren Abstand, daher befindet sich das blaue Punktepaar nicht im Freiraum.

Es ist festzustellen, dass  $F_\varepsilon$  die Schnittmenge des Einheitsquadrates mit einer Ellipse ist. Somit ist der Freiraum konvex.

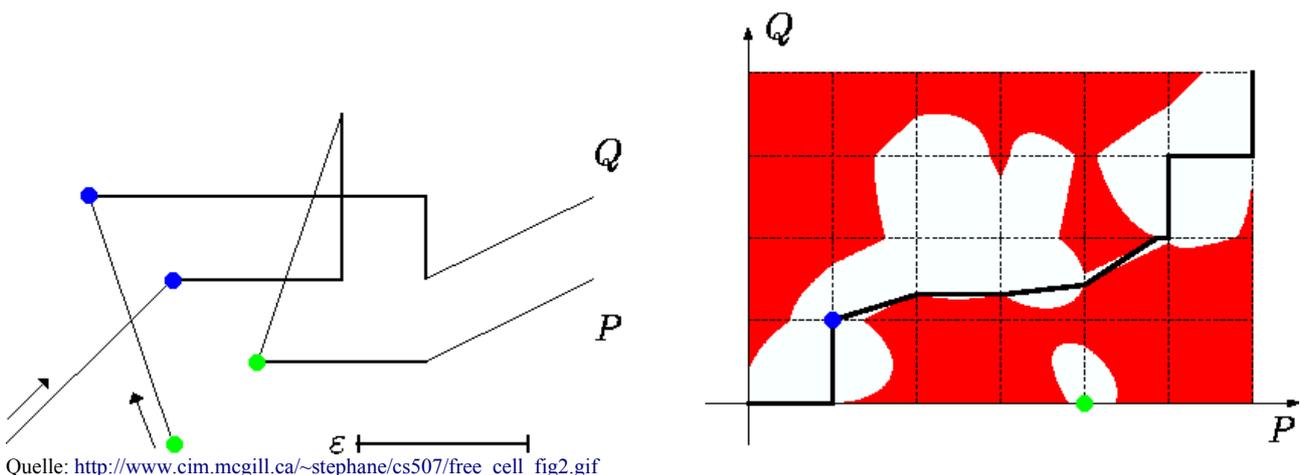
Beweis für  $\mathbb{R}^2$ :

Wenn nämlich die Polygone  $P$  und  $Q$  von affinen Mappings über  $[0,1]$  auf affine Mappings  $P'$  und  $Q'$  über  $\mathbb{R}$  erweitert werden, so ist das Mapping  $f(s,t):\mathbb{R}^2 \rightarrow \mathbb{R}^2$  mit  $f(s,t) = P'(s) - Q'(t)$  ebenfalls affin und es gilt  $F_\varepsilon = f^{-1}(D_\varepsilon) \cap [0,1]^2$ , wobei  $D_\varepsilon$  die Menge aller Punkte ist, deren Abstand nach gegebenem Distanzmaß zum Ursprung höchstens  $\varepsilon$  ist.

Wir erweitern nun  $F_\varepsilon$  auf Polygonzüge  $P$  und  $Q$  der Längen  $p, q$ :

$$F_\varepsilon := \{(s,t) \in [0,p] \times [0,q] \mid d(P(s), Q(t)) \leq \varepsilon\}$$

Der zu den Polygonzügen gehörende Freiraum setzt sich aus allen Freiräumen von allen Paaren, die ein Segment aus  $P$  und eines aus  $Q$  enthalten, zusammen. Nachfolgende Abbildung zeigt zwei Polygonzüge  $P$  und  $Q$ , eine Distanz  $\varepsilon$  und das zugehörige Freiraum-Diagramm.



Quelle: [http://www.cim.mcgill.ca/~stephane/cs507/free\\_cell\\_fig2.gif](http://www.cim.mcgill.ca/~stephane/cs507/free_cell_fig2.gif)

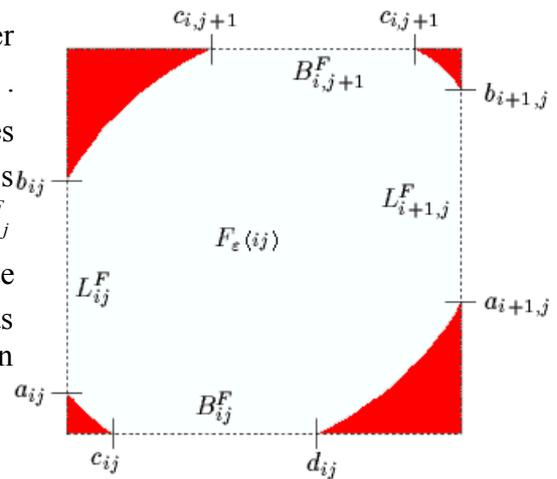
Es sind wieder zwei Punktepaare farblich im Diagramm dargestellt, um die Erzeugung zu verdeutlichen. Das blau markierte Punktepaar befindet sich am Ende der ersten Segmente von  $P$  und  $Q$  und ist daher im Diagramm mit  $s = t = 1$  eingezeichnet. Das grün markierte Punktepaar befindet sich am Ende des vierten Segments von  $P$  und am Anfang des ersten Segments von  $Q$ . Daher ist dieses Paar bei  $s = 4, t = 0$  im Diagramm eingezeichnet. Bei beiden Punktepaaren ist die euklidische Distanz der Punkte kleiner gleich  $\varepsilon$ , daher befinden sie sich im weiß markierten Freiraum.

Der hier vorgestellte Algorithmus basiert auf folgender Erkennung:

Für Polygonzüge  $P$  und  $Q$  gilt  $d_F(P,Q) \leq \varepsilon$  genau dann, wenn im zugehörigen Freiraum  $F_\varepsilon$  von  $(0,0)$  nach  $(p,q)$  eine in beiden Koordinaten monotone Kurve existiert.

Die obigen Abbildung zeigt eine solche Kurve in dem Diagramm, sodass in dem Fall  $d_F(P,Q) \leq \varepsilon$  gilt.

Sei  $F_\varepsilon(i, j)$  mit  $1 \leq i \leq p$  und  $1 \leq j \leq q$  der Freiraum der entsprechenden Segmente  $P(i-1)P(i)$  und  $Q(j-1)Q(j)$ . So sei  $L_{i,j}^F$  das linke Liniensegment von  $F_\varepsilon(i, j)$ , welches durch die Schnittmenge der linken Kante des Einheitsquadrates und dem Freiraum entsteht, und  $B_{i,j}^F$  das untere.  $L_{i+1,j}^F$  und  $B_{i,j+1}^F$  sind entsprechend das rechte und das obere Liniensegment, bzw. das linke und das untere Liniensegment der beiden rechts und oben angrenzenden Zellen.



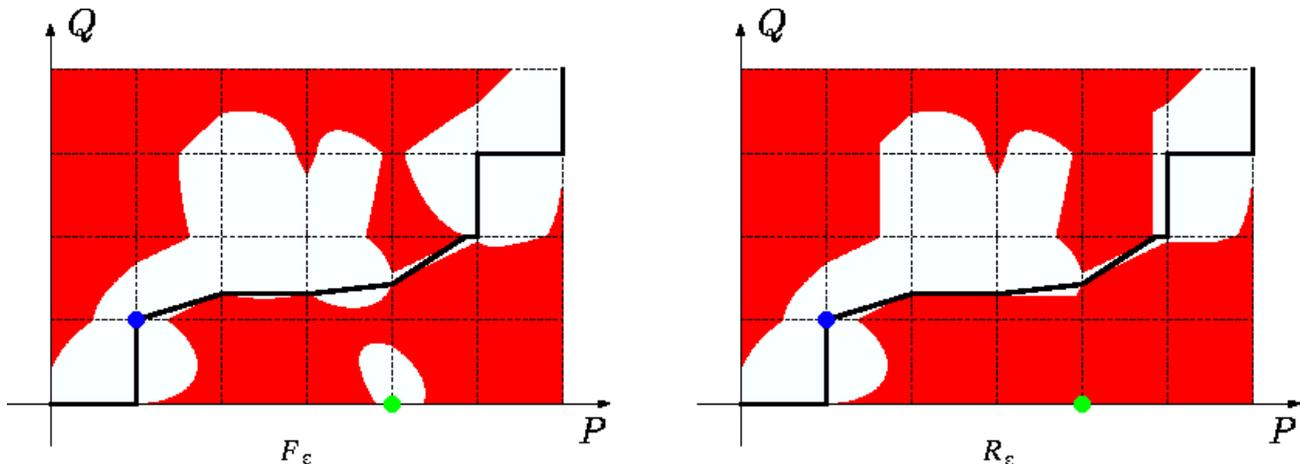
Außerdem definieren wir:

$$R_\varepsilon = \{(s, t) \in F_\varepsilon \mid \text{es existiert eine monotone Kurve von } (0,0) \text{ nach } (s, t)\}$$

Quelle:

[http://www.cim.mcgill.ca/~stephane/cs507/free\\_cell\\_fig3.gif](http://www.cim.mcgill.ca/~stephane/cs507/free_cell_fig3.gif)

$R_\varepsilon$  enthält also alle Punkte, die im Freiraum durch eine solche monotone Kurve erreichbar sind.  $R_\varepsilon$  wird monotoner Freiraum genannt. Nachfolgende Abbildung zeigt den Unterschied zwischen  $R_\varepsilon$  und  $F_\varepsilon$



Quelle: [http://www.cim.mcgill.ca/~stephane/cs507/free\\_cell\\_fig4.gif](http://www.cim.mcgill.ca/~stephane/cs507/free_cell_fig4.gif)

Analog zu den Liniensegmenten in  $F_\varepsilon(i, j)$  definieren wir für  $R_\varepsilon(i, j)$   $L_{i,j}^R$  und  $B_{i,j}^R$  als linkes bzw. unteres Liniensegment im monotonen Freiraum. Es gilt  $d_F(P, Q) \leq \varepsilon$  genau dann, wenn  $(p, q) \in L_{p+1,q}^R$ . Für gegebene  $L_{i,j}^R$ ,  $B_{i,j}^R$ ,  $L_{i+1,j}^F$  und  $B_{i,j+1}^F$  können  $L_{i+1,j}^R$  und  $B_{i,j+1}^R$  in konstanter Zeit berechnet werden. Mit diesen Erkenntnissen kann der folgende Algorithmus erstellt werden (mit Laufzeitangaben der einzelnen Schritte):

```

for each feasible pair (i, j) do compute  $L_{i,j}^F$  and  $B_{i,j}^F$   $O(pq)$ 
for i := 1 to p do determine  $B_{i,1}^R$   $O(p)$ 
for i := 1 to q do determine  $L_{1,j}^R$   $O(q)$ 
for i := 1 to p do  $O(pq)$ 
    for i := 1 to q do
        construct  $L_{i+1,j}^R$  and  $B_{i,j+1}^R$  from  $L_{i,j}^R$ ,  $B_{i,j}^R$ ,  $L_{i+1,j}^F$ ,  $B_{i,j+1}^F$ 
answer "yes" if  $(p, q) \in L_{p+1,q}^R$  "no" otherwise.

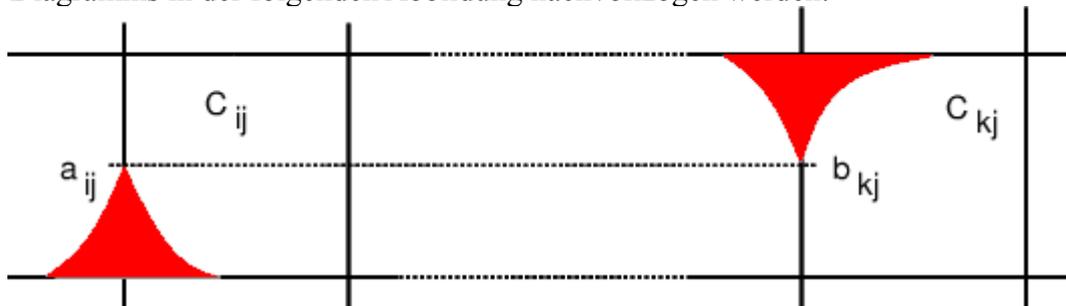
```

Für Polygonzüge  $P$  und  $Q$  der Länge  $p$  bzw.  $q$  entscheidet dieser Algorithmus in der Zeit  $O(pq)$ , ob  $d_F(P, Q) \leq \varepsilon$ .

### 5.3 Berechnung der Fréchet-Distanz

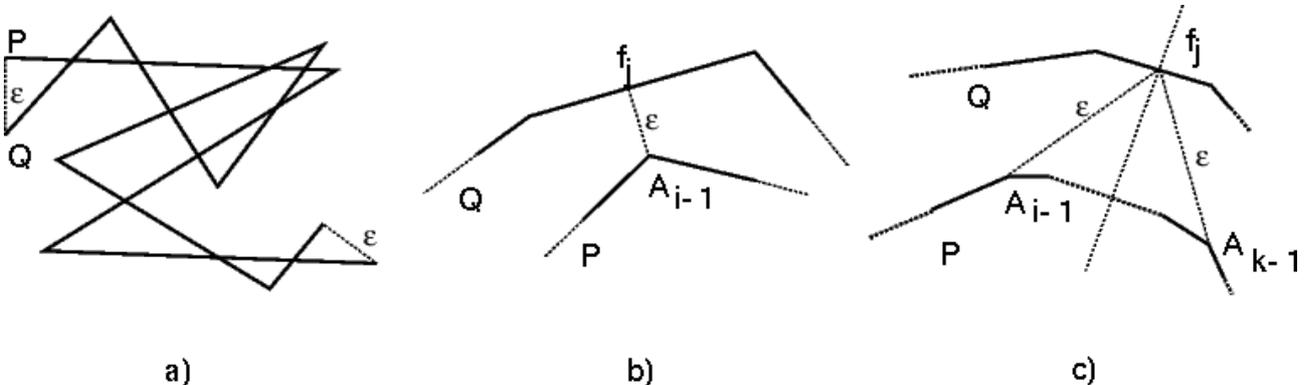
Ausgehend von der Lösung des Entscheidungsproblems soll nun die eigentliche Fréchet-Distanz  $d_F(P, Q)$  berechnet werden. Nehmen wir an, dass wir mit  $\varepsilon = 0$  starten und es kontinuierlich erhöhen. So wird der Freiraum  $F_\varepsilon$  mit wachsendem  $\varepsilon$  ebenfalls immer größer. Wir suchen das kleinste  $\varepsilon$ , sodass eine monotone Kurve im Freiraum-Diagramm von  $(0,0)$  nach  $(p, q)$  entsteht. Dieses kleinste  $\varepsilon$  kann nur in folgenden Fällen entstehen:

- wenn  $(0,0) \in F_\varepsilon$  und  $(p, q) \in F_\varepsilon$ . Dies bedeutet, dass die Distanz der Start- bzw. Endpunkt der Polygonzüge gleich  $\varepsilon$  ist.
- wenn ein Liniensegment  $L_{i,j}^F$  oder  $B_{i,j}^F$  im Freiraum für ein paar  $(i, j)$  nicht leer wird. In diesem Fall ist die Distanz zwischen dem Eckpunkt einer Kurve und der Kante der anderen Kurve gleich  $\varepsilon$ .
- wenn  $a_{ij} = b_{kj}$  oder  $c_{ij} = d_{ik}$  für einige  $i, j, k$ . Dies bedeutet, dass eine mögliche horizontale oder vertikale Passage im Diagramm geöffnet wird und kann im Ausschnitt des Freiraum-Diagramms in der folgenden Abbildung nachvollzogen werden:



Quelle: [http://www.cim.mcgill.ca/~stephane/cs507/critical\\_value\\_c\\_fig.gif](http://www.cim.mcgill.ca/~stephane/cs507/critical_value_c_fig.gif)

Die folgende Abbildung zeigt die geometrische Bedeutung der Fälle a), b), c):



Quelle: [http://www.cim.mcgill.ca/~stephane/cs507/critical\\_values\\_fig.gif](http://www.cim.mcgill.ca/~stephane/cs507/critical_values_fig.gif)

Nun ist zu überlegen, wie viele dieser kritischen Werte für  $\varepsilon$  existieren. Für den Fall a) gibt es  $O(1)$ , für den Fall b)  $O(pq)$  und für den Fall c)  $O(p^2q + pq^2)$  solcher zu berücksichtigenden Werte. Insgesamt sind also  $O(p^2q + pq^2)$  Werte von  $\varepsilon$  kritisch. Jeder dieser Werte kann in der Zeit  $O(1)$  berechnet werden.

Damit erhalten wir folgenden Algorithmus zur Berechnung der Fréchet-Distanz:

1. Determine all critical values of  $\varepsilon$
2. Sort them
3. Do a binary search on the sorted sequence in each search step solving the decision problem, continuing with the half containing smaller critical values if it has a positive answer and with the half containing larger critical values otherwise.

Nun soll die Laufzeit der einzelnen Schritte betrachtet werden:

1. Es gibt  $O(p^2q + pq^2)$  kritische Werte, die in jeweils konstanter Zeit berechnet werden können. Daher benötigt dieser Schritt die Zeit:  $O(p^2q + pq^2)$ .  
Sei  $p \geq q$ , gilt somit:  $O(p^3)$
2. Das Sortieren aller Werte benötigt wegen der Anzahl folgende Laufzeit:  
 $O((p^2q + pq^2) \log(p^2q + pq^2)) = O((p^2q + pq^2) \log(pq))$   
Sei  $p \geq q$ , gilt somit:  $O(p^3 \log(p))$
3. Bei diesem Schritt muss zum einen eine binäre Suche durchgeführt werden, die  $O(\log(p^2q + pq^2)) = O(\log(pq))$  Zeit benötigt, zum anderen muss jeweils das Entscheidungsproblem mit der Zeitkomplexität  $O(pq)$  gelöst werden. Somit benötigt dieser Schritt folgende Laufzeit:  $O(pq \log(pq))$   
Sei  $p \geq q$ , gilt somit:  $O(p^2 \log(p))$

Die Gesamtlaufzeit des Algorithmus wird also durch den zweiten Schritt mit der Laufzeit von  $O((p^2q + pq^2) \log(pq))$ , bzw. bei  $p \geq q$  mit  $O(p^3 \log(p))$  bestimmt.

Die Laufzeit zur Berechnung der Fréchet-Distanz kann asymptotisch noch verbessert werden, indem Meggido's Technik der parametrischen Suche angewandt wird. Damit wird eine Gesamtlaufzeit von  $O(pq \log(pq))$  erreicht. Dieses Verfahren soll jedoch nicht in dieser Arbeit vorgestellt werden, da es trotz des geringen asymptotischen Laufzeitverhaltens wegen extrem hoher Laufzeitkonstanten nicht praxistauglich ist. Weitere Informationen über das Verfahren können der Originalarbeit von H. Alt und M. Godau entnommen werden.

## 6 Fazit

In dieser Arbeit wurde ein Verfahren vorgestellt, welches als erster Algorithmus das Problem der Berechnung der Fréchet-Distanz von nicht geschlossenen Polygonzügen in polynomieller Zeit lösen konnte. Die wesentlichen Erkenntnisse des Algorithmus ist die Erzeugung des Freiraum-Diagramms zu zwei Polygonzügen  $P$  und  $Q$  und einem Wert  $\varepsilon$  sowie der Sachverhalt, dass in diesem Diagramm eine monotone Kurve existieren muss, damit die Fréchet-Distanz zwischen  $P$  und  $Q$  kleiner gleich  $\varepsilon$  ist. Ausgehend von diesen Erkenntnissen ist der eigentliche Algorithmus einfach aufgebaut.

In der Arbeit wurde nicht erklärt, wie die Fréchet-Distanz zwischen zwei geschlossenen Polygonen berechnet wird. Ein Algorithmus dafür baut auf dem hier erläuterten Verfahren auf und hat die Laufzeit  $O(pq \log^2(pq))$  für die Berechnung der Distanz zwischen zwei Polygonen  $P$  und  $Q$  der Längen  $p$ ,  $q$ . Dieser Algorithmus ist in der Originalarbeit von H. Alt und M. Godau näher erläutert.

## Literaturverzeichnis

- H. Alt, M. Godau: Computing the Fréchet distance between two polygonal curves (1995)
- H. Alt, L. J. Guibas: Discrete Geometric Shapes: Matching, Interpolation, and Approximation (1997)
- H. Alt, A. Efrat, G. Rote, C. Wenk: Matching planar maps (2002)
- W. Büschel: Ähnlichkeitsmaße zur Formerkennung (2008)

### Internetquellen:

- Computing the Fréchet distance between two polygonal curves (Stand 2002):  
<http://www.cim.mcgill.ca/~stephane/cs507/Project.html>