

# Einführung in die Computergeometrie, SS 2013

## Geometrische Algorithmen in der Ebene

### Abgabe bis zum 25.06.2013



#### Aufgabe 1 Schnitt von Gerade und Kreis

Aus einer Textdatei werden die Daten für eine Gerade  $g$  (zwei Punkte  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$ ) und für einen Kreis  $k$  (Mittelpunkt  $\mathbf{m} \in \mathbb{R}^2$  und Radius  $r$ ) eingelesen. Diese Datei kann etwa so aussehen:

```
70. 20.  
0. 80.  
20. 20. 40.
```

Zu berechnen sind die Schnittpunkte von  $g$  und  $k$ , wenn solche existieren.

Das Resultat soll in einer EPS-Datei visualisiert werden (Gerade  $g$ , Kreis  $k$ , Schnittpunkte).

#### Aufgabe 2 Unterteilungsalgorithmus von Chaikin

Das Programm soll eine Anzahl  $n$  und eine weitere Anzahl  $k$  aus einer Textdatei einlesen. Anschließend sollen  $n$  Punkte  $\mathbf{p}_0, \dots, \mathbf{p}_{n-1}$  des  $\mathbb{R}^2$  eingelesen werden. Diese Datei kann zum Beispiel so aussehen:

```
4 3  
0. 0.  
10. 60.  
50. 40.  
70. 0.
```

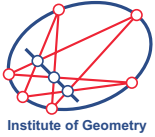
Mit Hilfe der Rekursionsformel

$$\mathbf{q}_{2i} = \frac{3}{4}\mathbf{p}_i + \frac{1}{4}\mathbf{p}_{i+1},$$

$$\mathbf{q}_{2i+1} = \frac{1}{4}\mathbf{p}_i + \frac{3}{4}\mathbf{p}_{i+1} \text{ mit } i = 0, \dots, n-2$$

(Unterteilungsalgorithmus von Chaikin) sollen aus den gegebenen  $n$  Punkten  $2(n-2)$  neue Punkte  $\mathbf{q}_0, \dots, \mathbf{q}_{2n-3}$  gewonnen werden. Anschließend wendet man dieselbe Regel auf die Punkte  $\mathbf{q}_i$  an. Insgesamt wird die Regel  $k$ -mal angewendet. Das entstehende Polygon heißt dann  $k$ -tes Chaikin-Polygon.

Das Resultat soll in einer EPS-Datei visualisiert werden (Eingabepolygon  $\mathbf{p}_i$  und  $k$ -tes Chaikin-Polygon).



# Einführung in die Computergeometrie, SS 2013

## Geometrische Algorithmen in der Ebene

### Abgabe bis zum 25.06.2013



#### Aufgabe 3 Schnitt von zwei achsenparallelen Rechtecken

Aus einer Textdatei sollen vier Punkte **a, b, c, d** des  $\mathbb{R}^2$  eingelesen werden. Diese Textdatei kann wie folgt aussehen:

```
0.    0.  
50.   70.  
-20. -30.  
40.   50.
```

Es soll das Schnittrechteck der beiden achsenparallelen Rechtecke mit den Diagonalen **ab** bzw. **cd** bestimmt werden, sofern es existiert.

**Tipp:** Projizieren Sie die beiden gegebenen Rechtecke normal auf die  $x$ -Achse bzw.  $y$ -Achse und ermitteln Sie die Schnitte der entstehenden Intervalle.

Das Resultat soll in einer EPS-Datei visualisiert werden (Eingaberechtecke und Schnittrechteck).

#### Aufgabe 4 Kreisbogen durch drei Punkte

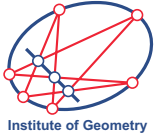
Aus einer Textdatei sollen drei nichtkollineare Punkte  $A, B, C$  der Ebene eingelesen werden. Diese Textdatei kann wie folgt aussehen:

```
70.   5.  
10.  70.  
-40. 20.
```

Es soll der Kreisbogen  $k$  bestimmt werden, der von  $A$  über  $B$  nach  $C$  läuft.

**Tipps:** Bestimmen Sie zuerst den Mittelpunkt  $M$  von  $k$  durch den Schnitt der beiden Streckensymmetralen von  $A, B$  und  $B, C$ , dann den Radius  $r$  von  $k$ . Um den Zentriwinkel  $\alpha = \angle AMC$  zu bestimmen, verwenden Sie am besten die Determinante  $\det(\overrightarrow{MA}, \overrightarrow{MC})$  und die C-Funktion 'atan2'.

Das Resultat soll in einer EPS-Datei visualisiert werden (Punkte  $A, B, C$ , Kreisbogen  $k$ , Mittelpunkt  $M$ ).



### Aufgabe 5 Lineare Regression

Es sollen eine Anzahl  $n$  und eine Liste von  $n$  Punkten  $(x_i, y_i)$  des  $\mathbb{R}^2$  aus einer Textdatei eingelesen werden ( $i$  läuft von 0 bis  $n - 1$ ). Diese Textdatei kann z.B. wie folgt aussehen:

```
6
0.  0.
20. 22.
40. 37.
60. 64.
80. 75.
100. 105.
```

Das Programm soll die Aufgabe lösen, eine Gerade mit der Gleichung  $kx + d = y$  zu finden, die die gegebenen Punkte bestapproximiert, d.h. die Unbekannten  $k$  und  $d$  sollen so bestimmt werden, dass die Gleichungen  $kx_i + d = y_i$  bestmöglich erfüllt sind. Im allgemeinen können diese  $n$  linearen Gleichungen in 2 Variablen  $k, d$  nur dann exakt erfüllt werden, wenn  $n = 2$  ist. Man schreibt das Gleichungssystem in Matrizenform an:

$$\begin{pmatrix} x_0 & 1 \\ x_1 & 1 \\ \vdots & \vdots \\ x_{n-1} & 1 \end{pmatrix} \cdot \begin{pmatrix} k \\ d \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

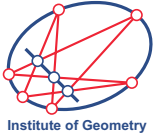
Die  $(n \times 2)$ -Matrix links heiÙe  $A$ , die rechte Spalte heiÙe  $s$ . Die 'bestmögliche Lösung'  $(k, d)$  dieses überbestimmten Gleichungssystems ist diejenige, wo

$\sum (kx_i + d - y_i)^2$  minimal ist. Wie bekannt (bzw. wie man leicht selbst nachrechnet) wird dieses Minimum als Lösung des  $(2 \times 2)$ -Gleichungssystems  $(A^T A) \cdot \begin{pmatrix} k \\ d \end{pmatrix} = A^T s$  gefunden.

Die Aufgabe besteht nun darin, die Matrix  $A^T A$  und den Spaltenvektor  $A^T s$  zu bestimmen und das dazugehörige lineare Gleichungssystem zu lösen.

*Bemerkung:* vgl. die Vorlesungen aus Linearer Algebra und/oder Numerischer Mathematik, zum Thema lineare Regression und der Methode der kleinsten Fehlerquadrate.

Das Resultat soll in einer EPS-Datei visualisiert werden (Eingabepunkte und Regressionsgerade).



Einführung in die Computergeometrie, SS 2013  
Geometrische Algorithmen in der Ebene  
Abgabe bis zum 25.06.2013



**Aufgabe 6 Unterteilungsalgorithmus von Lane und Riesenfeld**

Das Programm soll eine Anzahl  $n$  und eine weitere Anzahl  $k$  aus einer Textdatei einlesen. Anschließend sollen  $n$  Punkten des  $\mathbb{R}^2$  aus dieser Datei eingelesen werden. Eine solche Datei kann zum Beispiel so aussehen:

```
6      3
0.     0.
0.     40.
20.    40.
40.    20.
60.    0.
40.   -20.
```

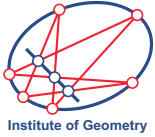
Mit Hilfe der Rekursionsformel

$$\mathbf{q}_{2i} = \frac{1}{2}(\mathbf{p}_i + \mathbf{p}_{i+1}), \quad i = 0, \dots, n-2,$$

$$\mathbf{q}_{2i+1} = \frac{1}{8}(\mathbf{p}_i + \mathbf{p}_{i+2}) + \frac{3}{4}\mathbf{p}_{i+1}, \quad i = 0, \dots, n-3$$

(Unterteilungsalgorithmus von Lane und Riesenfeld) sollen aus den gegebenen  $n$  Punkten  $2n-3$  neue Punkte  $\mathbf{q}_0, \dots, \mathbf{q}_{2n-4}$  gewonnen werden. Anschließend wendet man dieselbe Regel auf die Punkte  $\mathbf{q}_i$  an. Insgesamt wird die Regel  $k$ -mal angewendet. Das entstehende Polygon heißt dann  $k$ -tes Lane-Riesenfeld-Polygon.

Das Resultat soll in einer EPS-Datei visualisiert werden (Eingabepolygon  $\mathbf{p}_i$  und  $k$ -tes Lane-Riesenfeld-Polygon).



### Aufgabe 7 Schnitt von 2 Strecken

Das Programm soll vom Benutzer die Eingabe von 4 Punkten **a**, **b**, **c**, **d** des  $\mathbb{R}^2$  verlangen und anschließend herausfinden, ob die beiden Strecken mit den Endpunkten **a**, **b** bzw. **c**, **d** gemeinsame Punkte besitzen. Die Schnittmenge der beiden Strecken soll bestimmt werden. Sie kann leer oder ein Punkt oder eine Strecke sein.

Das Programm ist jedenfalls anhand der folgenden Eingaben zu testen:

| <b>a</b> | <b>b</b>   | <b>c</b>    | <b>d</b>        |
|----------|------------|-------------|-----------------|
| (0., 0.) | (60., 60.) | (-30., 50.) | (70., 0.)       |
| (0., 0.) | (40., 40.) | (20., 20.)  | (60., 60.)      |
| (0., 0.) | (20., 20.) | (40., 40.)  | (60., 60.)      |
| (0., 0.) | (40., 40.) | (40., 0.)   | (0., 40.)       |
| (0., 0.) | (40., 40.) | (0., 60.)   | (59.9., 59.999) |

**Hinweis:** Das Hauptproblem bei diesem Beispiel besteht darin, die Gleichungen der Verbindungsgeraden der Punkte **a**, **b** sowie **c**, **d** zu bestimmen und herauszufinden, ob diese Gleichungen dieselbe Gerade beschreiben. Es ist vorteilhaft herauszufinden, ob die beiden Endpunkte einer Strecke auf derselben Seite der Trägergeraden der anderen Strecke liegen, oder auf verschiedenen Seiten.

Das Resultat soll in einer EPS-Datei visualisiert werden (Input-Strecken und Schnittmenge, sofern existent).

### Aufgabe 8 Algorithmus von de Casteljau

Das Programm soll eine Anzahl  $n$  aus einer Textdatei einlesen. Anschließend sollen  $n + 1$  Punkte des  $\mathbb{R}^2$  aus dieser Datei eingelesen werden. Eine solche Datei kann zum Beispiel so aussehen:

```
3
0. 0.
20. 80.
60. 60.
70. 0.
```

Die eingelesenen Punkte seien mit  $\mathbf{p}_{0,0}, \dots, \mathbf{p}_{n,0}$  bezeichnet. Nach Wahl eines Parameters  $t \in [0., 1.]$  seien Punkte  $\mathbf{p}_{i,j}$  mit Hilfe der Rekursionsformel

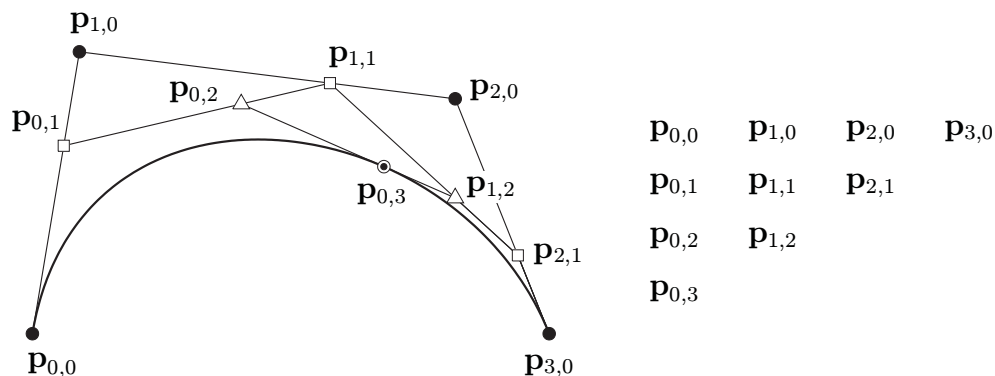
$$\mathbf{p}_{i,j} = (1 - t)\mathbf{p}_{i,j-1} + t\mathbf{p}_{i+1,j-1}, \quad j = 1, \dots, n; \quad i = 0, \dots, n - j;$$

definiert (Rekursion von Paul de Casteljau zum Auswerten der durch die Kontrollpunkte  $\mathbf{p}_{0,0}, \dots, \mathbf{p}_{n,0}$  definierten Bézierkurve beim Parameterwert  $t$ ). Figur 1 zeigt den Algorithmus für  $n = 3$ .

Die Bézierkurve besitzt die Parameterdarstellung

$$\mathbf{x}(t) = \mathbf{p}_{0,n}(t), \quad t \in [0, 1]$$

Das Resultat soll in einer EPS-Datei visualisiert werden (Eingabepolygon  $\mathbf{p}_{i,0}$  samt Ecken und Bézierkurve).



**Figur 1.** Algorithmus von de Casteljau für  $n = 3$ .