



## Bildgenerierung

Wintersemester 2021 / 2022

### Übungsblatt 3

#### Aufgabe 7 (*Gefüllte Polygone*)

Schreiben Sie eine Funktion

```
void drawPatternPolygon( Drawing& pic, const vector<IPoint2D>& ecken,  
                        const QImage& muster )
```

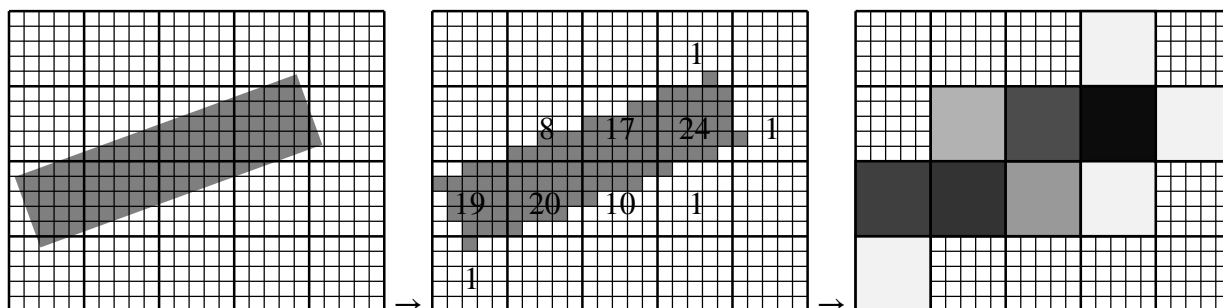
die mittels Algorithmus 3.22 der Vorlesung ein mit einem Muster gefülltes Polygon zeichnet.

#### Hinweise:

- im Verzeichnis `/home/bildgen/Aufgaben/polygone-3` auf dem CIP-Cluster finden Sie ein Rahmenprogramm und Beispiel-Eingabedateien.
- Schreiben Sie zunächst eine Funktion, welche das Polygon mit einer beliebigen Farbe füllt und modifizieren Sie diese anschließend, so dass mit einem Muster gefüllt wird.
- Durchlaufen Sie die Zeilen  $y_{min}$  bis  $y_{max}$ , die das Polygon enthalten, und datieren Sie in jedem Schritt eine *Tabelle der aktiven Kanten* auf. Es ist hilfreich, dies als eine Liste zu implementieren, die zu jeder Kante, welche die aktuelle y-Scan-Line schneidet, deren oberen Endpunkt, den aktuellen x-Wert sowie die Steigung enthält. Die Kanten werden nach x und ggf. nach Steigung sortiert gespeichert.
- Sorgen Sie dafür, dass die Füllung translationsinvariant, also unabhängig davon ist, an welcher Stelle des Bildes sich das Polygon befindet.
- Der Befehl `pic.drawPoint(x, y, QColor(muster.pixel(xmuster,ymuster)))` malt im Bild `pic` an die Stelle `(x, y)` einen Punkt in der Farbe, die das Muster `muster` an der Stelle `(xmuster, ymuster)` enthält. Breite und Höhe des Musters können Sie mit `muster.width()` und `muster.height()` ermitteln.

### Aufgabe 8 (Ein einfaches Antialiasing-Verfahren für Linien)

Ergänzen Sie in der Datei `antial.cc` unter `/home/bildgen/Aufgaben/anti-aliasing` die Funktion `drawAntialiasedWideLine`, die geglättete Linien beliebiger Breite zeichnet.



Verwenden Sie hierzu ein in beide Richtungen um einen Faktor  $f$  verfeinertes Raster (in der Abbildung ist also  $f = 5$ ). Fassen Sie die Linie in diesem feineren Raster als Rechteck auf und bestimmen Sie, welche (Sub-)Pixel eingefärbt werden müssen. Die Intensität der Pixel im Originalraster wird dann proportional zur Anzahl der eingefärbten Subpixel gewählt.

Verwenden Sie bei der Programmierung *kein* zusätzliches zweidimensionales Feld; merken Sie sich statt dessen für jede Subpixelzeile Anfangs- und Endpunkt des Rechtecks.

Testen Sie auch mit Linienbreiten  $< 1$ .

### Aufgabe 9 (Punktspiegelung im Raum)

Gegeben sei ein Punkt  $S = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \in \mathbb{R}^3$ .

Bestimmen Sie die Transformationsmatrix  $T \in \mathbb{R}^{4 \times 4}$ , die die Punktspiegelung an  $S$  in homogenen Koordinaten beschreibt.

**Abgabe:** Do., 11.11.2021, 16:15 Uhr

Senden Sie Ihre Lösungen der Theorie-Aufgaben und Ihre Programme per E-Mail an [bildgen@studs.math.uni-wuppertal.de](mailto:bildgen@studs.math.uni-wuppertal.de).