

# Uebung 4

Jan Fässler

24. November 2013

## 1 Einleitung

Dieses Dokument erklärt meinen Lösungsansatz der Uebung 4 im Modul Effiziente Algorithmen. Meine Lösung behandelt die **Variante A** der Übung.

## 2 Aufgabestellung

Entwickeln und implementieren Sie einen Algorithmus, welcher eine Anzahl Punkte in der xy-Ebene (durch Mouse-Klicks in einem einfach GUI definiert) mit einem möglichst kleinen Rechteck umschliesst.

## 3 Algorithmus

Der Algorithmus funktioniert so, dass er die konvexe Hülle der Punkte betrachtet und diese mit Linien verbindet. Für diese Linien wird der Winkel berechnet, in dem sie zur X-Achse stehen. Die Punkte der konvexen Hülle werden jeweils um den negativen Wert der Winkel gedreht. Dadurch kann das Rechteck einfach durch das Herausfinden der min/max Werte der X- und Y-Achse bestimmen und dessen Fläche berechnet werden.

Von diesen Rechtecken wird dann dasjenige genommen, welches die kleinste Fläche hat. Durch Rotieren der Eckpunkte um den Winkel der Linie aus der konvexen Hülle kann das Rechteck einfach gefunden werden.

## 4 Implementierung

Die Implementierung des Algorithmus ist in der Rectangle Klasse gemacht. Diese Klasse kann entweder mit 4 Double Werte (min/max für X- und Y-Achse) oder mit einer Liste von Punkten erzeugt werden. Wird der Konstruktor mit einer Liste von Punkten gefüttert und er definiert das Rechteck mit der Hilfe des Algorithmus.

Aus Gründen der Effizienz, darf die Liste nur die Punkte auf der konvexen Hülle beinhalten. In der Application Klasse wird beim Einfügen eines neuen Punktes nur der neue Punkt und die Punkte der alten konvexen Hülle genommen für die Berechnung der neuen Hülle.

Die konvexe Hülle wird mit der JarvisMarch Implementation von Manfred Vogel berechnet.

## 5 Fazit

Der Algorithmus läuft mit einer Laufzeit von  $O(n^2)$ . Wobei n die Anzahl der Punkte auf der konvexen Hülle repräsentiert.