

Uebung 5 Dokumentation

Jonas Schwammberger

5. Semester (HS 2013)

1 Algorithmus für die Konvexe Hülle

Für die Berechnung der Konvexen Hülle standen zwei Algorithmen zur Verfügung. Der Algorithmus von Graham und von Jarvis March. Welcher Algorithmus ($O(n \log n)$ versus $O(n^2)$) besser ist, hängt vom Eingabeverhalten des Benutzers ab. Da ich zum durchschnittlichen Verhalten keine Daten habe wurde Graham ausgewählt. Der Algorithmus von Graham wurde aus Interesse von mir selber implementiert und modifiziert.

1.1 Modifikation

Die Modifikation beschränkt sich darauf, dass nach dem Abbrechen der Hauptschleife nochmals eine `isConvex()` Prüfung mit Punkt P_0 , P_N und P_{N-1} . Im Falle das P_N auf der Strecke zwischen P_0 und P_{N-1} ist P_N nicht in der minimalen konvexen Hülle. Dies prüft der Algorithmus von Graham nicht, deshalb wurde diese Modifikation vorgenommen.

2 Algorithmus für die Berechnung des minimalen Rechtecks

Der Algorithmus startet mit dem minimalen achsenparallelen Rechteck der Konvexen Hülle. Nun werden alle Strecken des Rechtecks um den Berührungspunkt gedreht, bis sie sich eine Strecke einen neuen Punkt der Konvexen Hülle berührt. Bei jeder Berührung ist ein Kandidat für das minimale Rechteck. Wenn sich die Strecke mit einem neuen Punkt berührt hat, wird sie nun um den neuen Berührungspunkt weiter gedreht. Alle Strecken werden maximal um 90° gedreht, dann ist die Ausgangslage erreicht und das minimale Rechteck mindestens ein Mal besucht.

Somit hat dieser Algorithmus einen Worst-Case von $O(n)$. Er kann schneller abbrechen falls die Figur kleine Innenwinkel aufweist.

Der Algorithmus braucht mindestens vier Punkte, deshalb muss beim Input von drei Punkten eine Sonderbehandlung durchgeführt werden.

2.1 Sonderbehandlung bei drei Punkten

Bei drei Punkten muss man nur das minimale Rechteck eines Dreiecks finden. Das minimale Rechteck liegt immer mit einer Seite an der längsten Seite des Dreiecks.

2.2 Implementation

2.3 Nachteile

Der grösste Nachteil dieser Implementierung ist, dass für die Drehung Floating Point Datentypen verwendet werden müssen. Bei der Konvertierung von Floating Point zu Integer kommen oft Fehler von ± 1 vor, welches auf der Graphik sofort sichtbar ist.

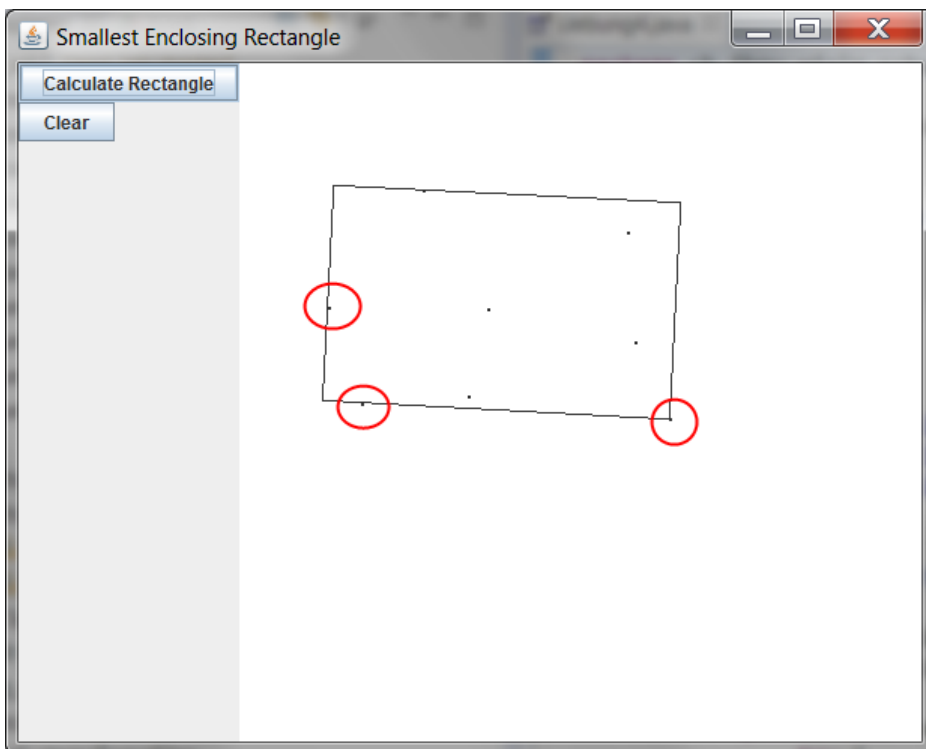


Abbildung 1: Fehler in der Graphik