

Praktikums Protokoll

Teilnehmer
Sebastian Stumpf
Felix Schramm

Fach	Computational Geometry
Abgabe	Praktikumsabgabe 4 – Konvexe Hüllen
Datum	04.07.15

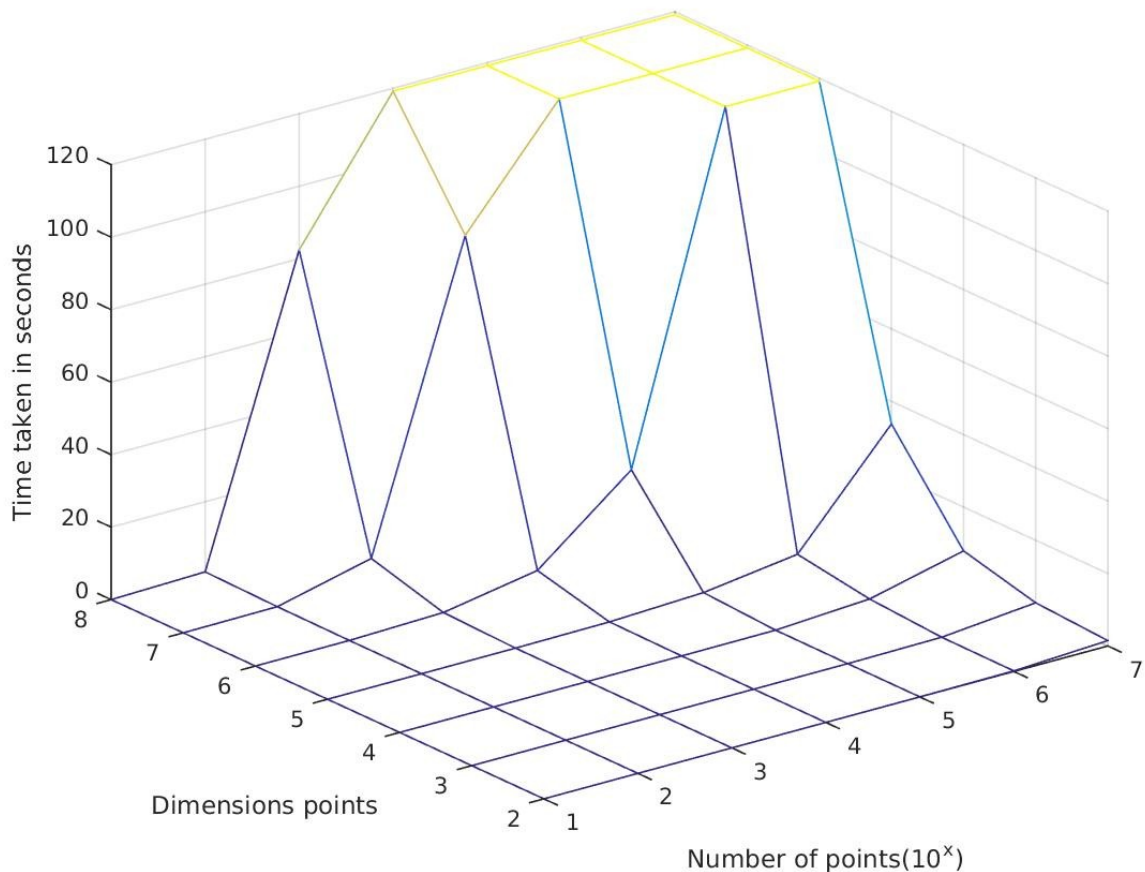
Aufgabenbeschreibung

Installieren Sie das Programm qhull, erzeugen Sie zufällige Punktemengen und berechnen Sie mit qhull konvexe Hüllen, auch in höheren Dimensionen (qhull bringt ein Werkzeug zur Erzeugung von Punktmengen mit). Plotten Sie die Zeiten für zunehmende Punktzahlen bei unterschiedlichen Dimensionen (2-8).

Lösung

- Installation via apt-get install linux ftw
- Zur Messung wurden Bash-Skripte erzeugt, die eine Reihe von Testwerten an rbox und qhull übergeben und die gemessene Zeit von qhull in eine Datei schreiben. Es kann ein Time-out gesetzt werden, nachdem die Messung abgebrochen wird und für die jeweilige Kombination aus Dimension und Exponent über Zehn ein Überschreiten der Zeit in der Datei kenntlich gemacht wird.
- Ein weiteres Skript erzeugt aus dieser Datei ein Matlab-Skript, dass die Messergebnisse plottet.

Ergebnisse



Erkenntnisse

Der Algorithmus ist für zwei Dimensionen sehr schnell und der Arbeitsspeicher zum Speichern der Punkte geht aus, bevor die Berechnung der konvexen Hülle länger als 10 Sekunden dauert.

Die Dauer der Berechnung scheint auch stark von der konkreten Verteilung der Punkte abzuhängen.

Sieht man sich die benötigten Zeiten für zwei Dimensionen an, fällt auf dass trotz einer jeweiligen Verzehnfachung der Punktemenge die nicht im selben Maße ansteigt:

10 -> 0.000104s

100 -> 0.000293s (2,8 * T(10))

1000 -> 0.002634s (9,0 * T(100))

10000 -> 0.01555s (5,9 * T(1000))

Bei der Betrachtung des Graphen scheinen Funktionswerte mit gleicher Zeit im 45° Winkel zu den Achsen der Dimension und der Punktemenge zu verlaufen. Demnach wirkt sich die Erhöhung der Dimension um eins genauso stark auf die benötigte Zeit aus, wie eine Verzehnfachung der verwendeten Punkte.

Anmerkung zur Verwendung der Skripte

QHuller misst die Zeit von qhull für verschiedene Dimensionen und Punktemengen und schreibt die Ergebnisse in times.log. Die Messungen werden nach Ablauf des Time-outs t abgebrochen.

z.B.: QHuller -d 8 -e 7 -t 600

mkmatlab generiert aus der times.log ein Matlabskript, das die gemessenen Werte plottet.

z.B.: mkmatlab -i times.log -o plottimes

(Liest aus times.log, erzeugt plottimes.m und plottimes.jpg)