

Computational Geometry - Abgabe 4

1st Bartolovic Eduard
Hochschule München
München, Deutschland
eduard.bartolovic0@hm.edu

I. KONVEXE HÜLLE

Für die Berechnung der konvexen Hülle wurde das Programm *qhull* [1] verwendet. *qhull* verwendet den Quickhull Algorithmus um die konvexe Hülle zu berechnen. *qhull* lässt sich per Kommandozeile ansprechen. Ein Beispiel für einen Aufruf:

```
rbox 3 D2 | qconvex s o TO result
```

Der Befehl *rbox* generiert eine gewisse Anzahl an Punkten in einer definierten Dimension. Der Befehl *qconvex* berechnet die konvexe Hülle. Mit einer Pipe können die Punkte direkt *qconvex* übergeben werden.

II. LAUFZEIT VON *qhull*

Es wurde die Laufzeit von *qhull* untersucht. Hierbei wurden verschiedene Konfiguration bezüglich Anzahl und Dimension getestet. Um den Einfluss von Ausreißern zu reduzieren, wurde das mittel aus fünf Durchläufen gebildet. Um das Problem etwas zu automatisieren, entwickelte ich ein Batchfile welches *qhull* mit definierten Konfigurationen automatisch fünf mal gestartet hat. Die Anzahl der Punkte, die getestet worden sind:

- 1000
- 10000
- 100000
- 500000
- 1000000
- 2500000
- 5000000
- 7500000
- 10000000
- 20000000

Jede Punkteanzahl wurde mit den Dimensionen 2 bis 8 getestet.

Qhull ist auf Windows nur für 32 Bit kompiliert. Dies resultiert darin, dass *Qhull* nur 4 GB Arbeitsspeicher allokalieren kann. Deshalb schlägt ab der Dimension 5 bei einer höheren Punktezahl das Programm fehl. Da aber die Laufzeiten extrem ansteigen, ist dies nicht weiter tragisch.

Die Ergebnisse der Messungen sind in den beiden Abbildungen 1 und 2 zu sehen.

Wie gut in den Messungen zu sehen ist steigt der Aufwand mit Anzahl der Punkte und Dimensionen. Die Laufzeit steigt deutlicher mit der Dimension als mit den Punkten.

Die gemessenen Laufzeitsteigerungen der Punkte entsprechen

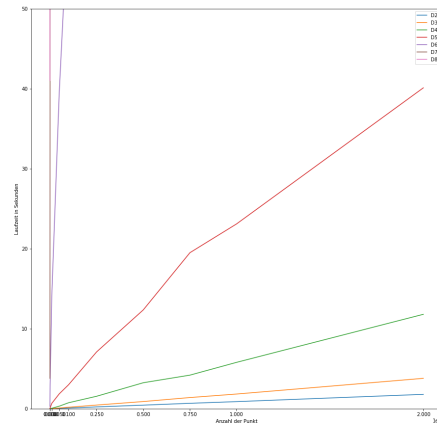


Abbildung 1. Messung der Laufzeit abhängig zur Anzahl der Punkte und Dimensionen

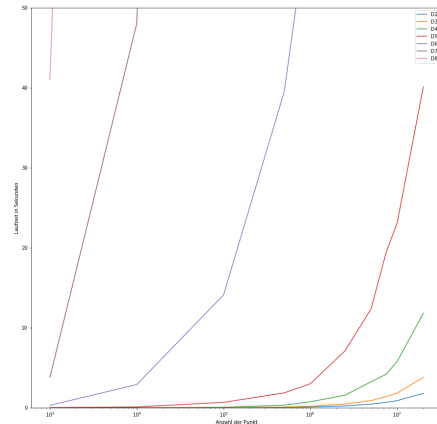


Abbildung 2. Messung der Laufzeit abhängig zur Anzahl der Punkte und Dimensionen (X-Achse ist Log Skaliert)

grob der durchschnittlichen Komplexität von Quickhull mit $\mathcal{O}(n * \log(v))$ bei Dimensionen von 2 und 3. n entspricht der Anzahl der Input Punkte. v ist die Anzahl der Punkte der Konvexen Hülle.

Ab einer Dimension von größer 3 steigt der Aufwand auf $\mathcal{O}(\frac{n \cdot f_v}{v})$. f_v sei die maximale Anzahl von Facetten für eine konvexe Hülle aus den Eckpunkten.

In Abhängigkeit der Dimension wächst f_v sehr schnell mit:

$$\mathcal{O}\left(\frac{v^{\text{floor}(d/2)}}{\text{floor}(d/2)!}\right)$$

[2]

LITERATUR

[1] <http://www.qhull.org/html/qconvex.htm>

[2] <http://www.geom.uiuc.edu/~bradb/qhull3.1/html/qh-in.htm>

III. ANHANG

Batchfile zum Testen der Laufzeit von Q-Hull:

```
rbox 5000000 D2 | qconvex s o TO result1
rbox 5000000 D2 | qconvex s o TO result2
rbox 5000000 D2 | qconvex s o TO result3
rbox 5000000 D2 | qconvex s o TO result4
rbox 5000000 D2 | qconvex s o TO result5
```