



Aufgabenblatt 7: Hierarchische Datenstrukturen – Binary Space Partition

In diesem Aufgabenblatt Generieren Sie für eine Menge von Objekten (hier einfach Punkte im 2D, eingebettet in den 3D durch $z=0$) einen BSP-Baum und generieren eine Back2Front-Sortierung der Objekte für eine gegebene Beobachter-Position. Als Vorgabe finden Sie auf der EMIL-Seite eine Klasse, die die Datenstruktur implementiert (BSPTreeNode) und einen neuen Szenengraphknoten, der einen BSP-Baum darstellen kann (BSPNode). Im Framework finden Sie außerdem eine Klasse PrincipalComponentAnalysis zum Durchführen der Hauptkomponentenanalyse für eine Menge von 3D-Punkten. Machen Sie sich zunächst mit den Vorgaben vertraut.

Als Vorgabe finden Sie auch eine Klasse BSPTreeTools mit Methodenrumpfen für die Aufgaben. Diese können Sie als Hilfestellung verwenden, müssen aber nicht.

Verwenden Sie als Datenpunkte Zufallspunkte aus dem Bereich $d = ([-1,1], [-1,1], 0)$

Aufgabe 5.1: BSP-Baum aufbauen

Schwerpunkte: Aufbau eines BSP-Baumes, Szenen-Unterteilung durch Hauptkomponentenanalyse (eng. Principal Component Analysis, PCA)

Idee: Beim Aufbau eines BSP-Baumes muss in jedem Schritt für eine Menge von Punkten eine Teilungsebene gefunden werden. Diese Ebene sollte die Punktmenge möglichst halbieren. Um eine solche Teilungsebene für eine Menge von finden, verwenden wir Hauptkomponentenanalyse (PCA). Eine Einführung in das Thema finden Sie unter anderem hier: [1].

Zusammenfassung des Konzeptes: Für die Punkt wird eine Kovarianzmatrix gebildet (im 3D eine 3x3-Matrix). Eine Singulärwertzerlegung bestimmt die Eigenwerte und Eigenvektoren der Matrix (3 Eigenwerte und 3 zugehörige 3D-Eigenvektoren). Der Eigenvektor zum größten Eigenwert liegt in der Richtung, in der die Datenpunkte die stärkste Ausdehnung aufweisen. Diese Richtung verwenden wir als Normalenrichtung der Trennungsebene. Als Punkt der Trennungsebene verwenden wir den Schwerpunkt der Datenpunkte.

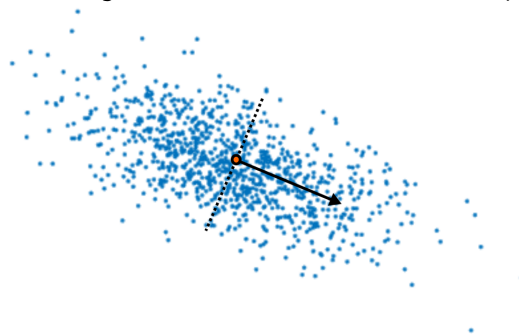


Abbildung 1: Beispiel einer Hauptkomponentenanalyse zur Unterteilung einer Punktemenge mittels einer Trennungsebene: Die PCA für die blauen Datenpunkte liefert Eigenvektoren und zugehörige Eigenwerte. Der Eigenvektor zum größten Eigenwert ist als Pfeil dargestellt. Diesen verwendet man als Normale der Trennungsebene. Die Ebene läuft außerdem durch den Schwerpunkt der Datenpunkte.

Aufgabe:

- Schreiben Sie einen rekursiven Algorithmus, der für eine Menge von Punkten mittels PCA eine Trennungsebene bestimmt. Basierend auf der Trennungsebene wird die Punktmenge in zwei Teile geteilt. Für jeden der Teile wird der Algorithmus rekursiv aufgerufen. Damit ergibt sich die BSP Baumstruktur. Die Rekursion endet, wenn nur noch ein Punkt auf einer Seite einer Teilungsebene übrig bleibt. Dieser Punkt wird als Element im entsprechenden Knoten abgelegt.
- Ein Sonderfall ergibt sich, wenn in der Punktmenge genau zwei Punkt übrig sind. Dann schlägt die PCA fehl. Um stattdessen die Trennungsebene zu bestimmen geht man folgendermaßen vor:
 - Punkt der Ebene: Schwerpunkt der beiden Datenpunkte
 - Normale der Ebene: Differenzvektor der beiden Datenpunkte (normiert)

Aufgabe 5.2: Back2Front-Sortierung

Schwerpunkte: Back2Front Sortierung

Aufgabe:

- Implementieren Sie den Algorithmus zur Finden der Back2Front-Sortierung aus der Vorlesung.

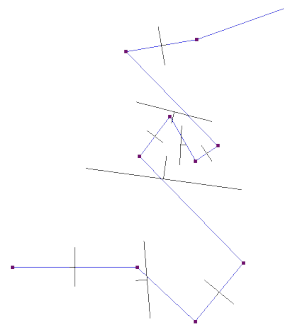


Abbildung 2: BSP-Baum für 10 Datenpunkte (rot). Die Trennungsebenen sind in schwarz zusammen mit ihren Normalen dargestellt. Je weiter unten im Baum eine Trennungsebene ist, desto kürzer ist sie gezeichnet. Der blaue Linienzug stellt die Back2Front-Sortierung zum Augpunkt (blau) dar.

Quellen

[1] Wikipedia – Hauptkomponentenanalyse: <https://de.wikipedia.org/wiki/Hauptkomponentenanalyse>, abgerufen am 18.07.2016