



05.07.2015

BILDBASIERTE MODELLIERUNG SS 2012 ÜBUNGSBLATT 9

Abgabe: Präsentation der bearbeiteten Aufgaben in der Übung am 10.07.2015.

Für die Programmieraufgaben kann in Gruppen von max. 3 Leuten zusammengearbeitet werden. Dabei muss aber jeder einzelne in der Lage sein, alle Teile des Programms zu erklären. Die Materialien für die Programmieraufgaben sind jeweils erhältlich unter:

<http://www.cg.cs.tu-bs.de/teaching/lectures/ss12/bbm/>

In dieser Aufgabe soll ein einfacher Shape-from-Shading-Algorithmus implementiert werden. Gegeben sind mehrere Bilder eines diffusen Objekts aus derselben Kameraperspektive, aber unter Beleuchtung aus unterschiedlichen (bekannten) Richtungen. Daraus können die Normalen der Geometrie rekonstruiert werden, die anschließend zu einer Tiefenkarte aufintegriert werden.

9.1 Normalen berechnen (10 Punkte)

Für die Intensität I , die von einer diffusen, aus Richtung \vec{l} beleuchteten Oberfläche mit Normale \vec{n} zurückgeworfen wird, gilt $I \propto \vec{l} \cdot \vec{n}$. Jedes Eingabebild definiert I und \vec{l} für jeden Pixel (\vec{l} ist konstant über alle Pixel).

- Stelle ein lineares Gleichungssystem auf, das obigen Zusammenhang für einen Pixel in allen Bildern beschreibt.
- Löse das Gleichungssystem für jeden Pixel, z.B. mittels `cvSolve`.
- Zeige die resultierenden Normalen als Farbbild an und prüfe sie auf Plausibilität.

9.2 Tiefenkarte berechnen (20 Punkte)

Um aus den Normalen eine konsistente Geometrie zu berechnen, muss eine Tiefenkarte gefunden werden, die genau diese Normalen aufweist. Ausgehend von einer flachen Tiefenkarte wird jeder Pixel an den Punkt verschoben, der die Abweichung der tatsächlichen Normalen an dieser Stelle von der in der vorhergehenden Aufgabe berechneten Normalen minimiert. Dieser Vorgang wird wiederholt, bis Konvergenz eintritt.

- Aus der Normalen lassen sich diskretisierte Ableitungen der Tiefenkarte z berechnen:

$$z(x+1, y) - z(x, y) = \frac{n_x(x, y)}{n_z(x, y)} \quad \text{sowie} \quad z(x, y+1) - z(x, y) = \frac{n_y(x, y)}{n_z(x, y)}$$

Löse diese Gleichungen nach $z(x, y)$, $z(x+1, y)$ bzw. $z(x, y+1)$ auf. Dies liefert Sollwerte, die $z(x, y)$, $z(x+1, y)$ bzw. $z(x, y+1)$ gemäß $\vec{n}(x, y)$ und den benachbarten z -Werten haben müssten.

- Implementiere ein iteratives Lösungsverfahren für die Tiefenkarte, indem du in jedem Schritt jeden Tiefenwert durch den Mittelwert seiner Sollwerte ersetzt.
- Implementiere die Randbedingung, dass die Ränder der Tiefenkarte den Wert null haben sollen, indem du sie während der Iterationen auf diesem Wert festhältst.