Prof. G. Zachmann
P. Meißenhelter
R. Fischer
A. Mühlenbrock

Universität Bremen
Fachbereich 3
Arbeitsgruppe CGVR
19. November 2021

Wintersemester 2021/22

Übungen zu Computergraphik - Blatt 5

Abgabe am 29.11.2021, 23:59 Uhr

Aufgabe 1 (Rasterisierung einer Linie, *2,5 Punkte*)

In der Vorlesung wurde ein einfacher, inkrementeller Algorithmus zur Rasterisierung von Linien vorgestellt (Folien 7-9). Zeigen Sie anhand eines Rechenbeispiels, wie die Linienparameter $tnext_x$ und $tnext_y$ sowie die Deltas dtx und dty berechnet werden können. Gegeben sei dazu eine Linie von $P_0 = (1, 0)$ nach $P_1 = (5, 3)$.

Aufgabe 2 (Interpolation, *2,5+1,5 Punkte*)

a) Implementieren Sie das in der Vorlesung gezeigte Verfahren zur Interpolation von Attributen auf einer Linie. Nutzen Sie dazu das Framework „Midpoint“ auf der CGVR Website. In der Datei `glwidget.cpp` befindet sich die Funktion `rasterLine()`, in der bereits die gerasterten Koordinaten einer Linien vorgegeben sind. Ihre Aufgabe ist es, die Farbe inkrementell zu berechnen (d.h. nicht in jedem Schritt komplett neu!). Das Ergebnis sollte dem in Abbildung 1 entsprechen.

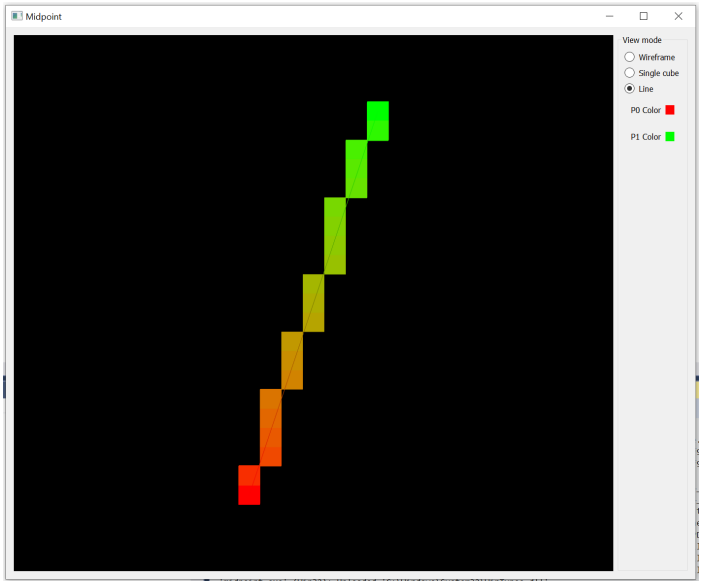

Abbildung 1: Farbinterpolation

b) Wir entwickeln im Folgenden einen einfachen Algorithmus zur Vergrößerung von Bildern. Wir nehmen an, dass das Eingabebild ein Grauwert-Bild ist, gegeben als $I(x, y)$. Nehmen Sie an, dass die Pixelwerte von I auf einem Gitter gegeben sind, mit $x = 0, ..., w$, und $y = 0, ..., h$ (siehe Abb. 2). Das Bild soll auf ein Bild $\hat{I}$ der Größe $[0, W] \times [0, H]$ vergrößert werden (s. Abb.). Der Grauwert

von $\hat{I}(x, y)$ müsste also an der Stelle $I(\frac{w}{W}x, \frac{h}{H}y)$ ausgelesen werden. An dieser Stelle befindet sich aber i.A. kein Pixel, sondern an den vier Gitterpositionen darum herum. D.h. in dieser Aufgabe sind die Pixelwerte an den Gitterknoten gegeben, nicht den Mittelpunkten der Zellen! Geben Sie an, wie man mittels Interpolation einen Grauwert berechnen kann.

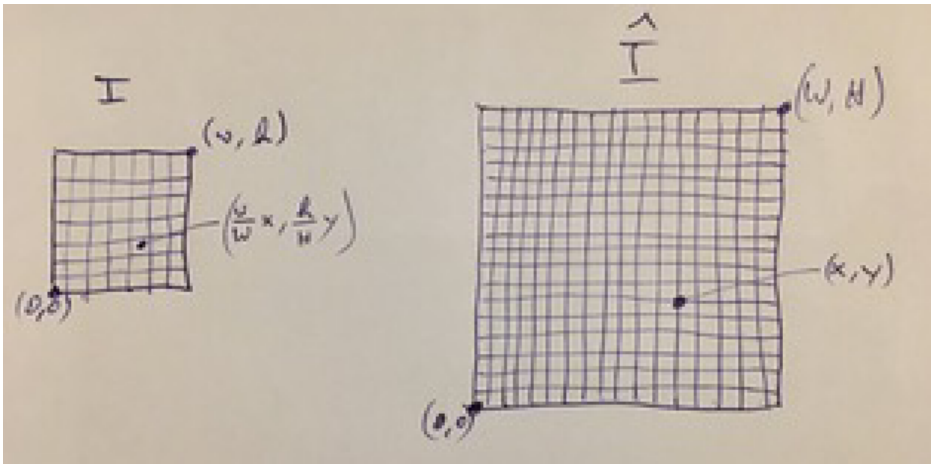

Abbildung 2: Bildvergrößerung

Aufgabe 3 (Continuous Collision Detection, *1+2,5 Punkte*)

a) Skizzieren sie mindestens zwei unterschiedliche Fälle, in denen das in der Vorlesung genannte Ausschlusskriterium 2 zum Test auf Nulldurchgang in der Continuous Collision Detection fälschlicherweise ein negatives Ergebnis liefert (Folie 34).

b) In der Vorlesung haben wir das Voxmap-Pointshell Verfahren zur Continuous Collision Detection kennen gelernt. Der 1. Schritt besteht hierbei daraus, alle Voxel zu finden die vom Liniensegment getroffen werden, siehe Abb. 3. Dies ist in dem Framework „Midpoint“ teilimplementiert und soll vervollständigt werden. Dazu muss, wie zuvor, die Funktion `rasterLine()` in der Datei `glwidget.cpp` bearbeitet werden. Die Farbe spielt hierbei keine Rolle.

Tipp: Für einen einfachen Schnitttest zwischen Liniensegment und Ebene bietet es sich an, von der Linie in Parameterform und der Ebene in Normalenform auszugehen.

Hinweis: Mittels der Variable `interpEx` kann vor dem Kompilieren zwischen der Anzeige der Lösung von Aufgabe 2a (`interpEx=true`) und 3a (`interpEx=false`) gewechselt werden.

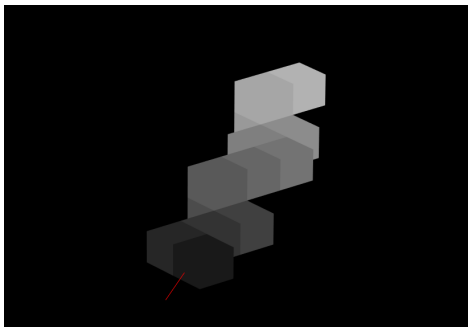

Abbildung 3: Voxel die von Strahl getroffen werden.