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

Universität Bremen
Fachbereich 3
Arbeitsgruppe CGVR
3. Dezember 2021

Wintersemester 2021/22

Übungen zu Computergraphik - Blatt 7

Abgabe am 13.12.2021, 23:59 Uhr

Aufgabe 1 (Shader - Depth-of-Field in Echtzeit, *10 Punkte*)

Auf der Vorlesungshomepage finden Sie das Framework `DoF_Framework`. Ziel dieser Aufgabe ist es, die durch OpenGL dargestellte Szene in Echtzeit zu filtern. Da man für das Filtering zu jedem Fragment die benachbarten Fragmente einlesen muss, ist ein 2-Pass Verfahren nötig.

Im ersten Renderpass wird die Szene in ein sogenanntes Framebuffer-Objekt (FBO). gerendert. Ein FBO kann man sich vereinfacht als einen Bereich im Textur/Bild-speicher der Graphikkarte vorstellen. Im zweiten Renderpass wird das FBO als Textur/Bild eingebunden. Im Fragmentshader haben Sie nun über die Textur Zugriff auf die gerenderte Szene. Um zu gewährleisten, dass die komplette Textur verarbeitet wird, wird ein bildschirmfüllendes Rechteck gezeichnet, welches die Fragmentshaderaufrufe "anwirft" (ein Shaderaufruf pro Texel). Die Textur-Auflösung entspricht in unserem Framework genau der Auflösung des mit `glViewport()` festgelegten OpenGL-Zeichenfensters (1024x786 px).

Es empfiehlt sich die folgenden Unteraufgaben in separaten Unterfunktionen zu lösen, um die verschiedenen Implementierungen/Verbesserungen im Nachhinein zu vergleichen.

a) Schreiben Sie einen Fragmentshader, welcher einen Filterkernel K der Grösse 3×3 auf die Textur T anwendet und die Fragmentfarbe entsprechend setzt, d.h.

$$\text{FragColor}(x,y) = \sum_{i=-1}^{1}\sum_{j=-1}^{1} K(i,j)T(x-i,y-j) \tag{1}$$

Nehmen Sie `DepthOfField.fragmentshader` als Gerüst für Ihren Shader. Der Kernel (Box-blur) $3\times3-$ Matrix wird als Uniform-Variable an den Shader übergeben. Hinweis: Mit wiederholter Anwendung des Box-blur (Mittelwert-Filter) nähert man sich dem Gaussian-blur (Gauß-Filter) an.

b) Neben dem Farbbild kann im FBO auch der Z-Buffer gespeichert werden und im zweiten Pass als eigene Textur ausgelesen und benutzt werden. Damit ist es möglich, Effekte wie Depth-of-Field zu realisieren. Erweitern Sie den Shader so, dass der Kernel nur auf Texel, die innerhalb des "Circle of Confusion" von Nachbarpixeln liegen, angewendet wird. Die Focal Depth wird in der vordefinierten Uniform-Variable `focalDepth` übergeben. Versuchen sie die Kernel bzw. Filtergröße variable zu halten, so dass Sie einfach im Code zwischen der Filtergröße (3x3, 5x5,...) wechseln können. Also beim Mittelwert-Filter nicht mehr auf die übergebene $3\times3-$Matrix angewiesen sind. Schauen Sie sich die Folie 22 an.

c) Erweitern Sie den Shader nun so, dass die Größe bzw. der Wirkungsbereich des Kernels mit der Entfernung des Texels von der Focal Depth skaliert. Überlegen Sie sich eine Methode, Ihren Kernel und seinen Wirkungsbereich sinnvoll und kontinuierlich zu vergrößern. Sie brauchen hier nicht mehr den übergebenen Kernel benutzen, sondern können ihn komplett im Shader generieren. Schauen Sie sich die Folien 24-25 an.