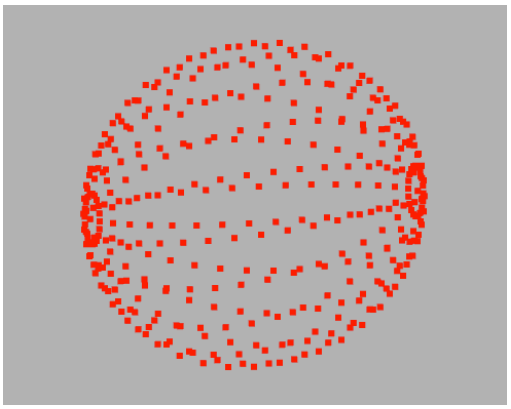
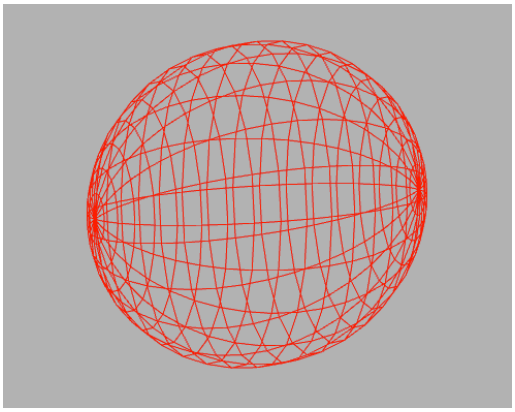
Prof. Dr.-Ing. Hartmut Schirmacher
Fachbereich Informatik und Medien

Computergrafik 2: Aufgabe 2.1

Parametrische Fläche abtasten



In dieser Aufgabe: 3D-Punkte



Später: Wireframe

Lernziele / Motivation

In dieser Übung erzeugen Sie als Vorstufe einer 3D-Oberfläche eine Punktdarstellung einer parametrischen Fläche und stellen Sie mit Hilfe des WebGL-Übungsframeworks dar.

Übungsframework

Für diese Übung laden Sie sich bitte das Mini-Framework zur Aufgabe 2 aus Moodle herunter und entpacken Sie es in einem entsprechenden Verzeichnis neben dem zur Aufgabe 1. Bitte beachten Sie, dass das Verzeichnis `lib/` aus Aufgabe 1 neben dem Verzeichnis `cg2-a02/` liegt!

Mit dem Framework beschäftigen wir uns noch ausführlicher in der nächsten SU und Teilaufgabe; hier konzentrieren wir uns zunächst nur auf das Erzeugen einer parametrischen Geometrie. Bitte beachten Sie, dass das Übungsframework dynamisch Shader-Dateien nachlädt; dafür ist es je nach verwendetem Browser erforderlich, die Dateien über einen lokalen Webserver abzurufen und nicht direkt aus dem Filesystem (siehe Warmup-Aufgabe).

Aufgabe 2.1: `Ellipsoid` als 3D-Punkte darstellen

Stellen Sie `index.html` im Browser dar und drücken Sie die Tasten 'X' oder 'Y' (mit und ohne Shift) und schalten sie die Animation ein. Sie sehen ein 3D-"Band", dargestellt als Punktwolke.

Studieren Sie das Modul `models/band.js`. Es besteht aus

- einem Konstruktor, der ein Array mit den Koordinaten von 3D-Punkten füllt und aus diesem Array dann ein Vertex Buffer Object (VBO) erzeugt;
- einer `draw()`-Methode, die dieses VBO mittels `drawArrays()` als einzelne Punkte darstellt.

Studieren Sie das Modul `scene.js`. Auch die Szene besteht im wesentlichen aus einem Konstruktor und einer `draw()`-Funktion (sowie einer `rotate()`-Funktion, dazu später mehr). Im Konstruktor werden WebGL-Programme und Szenenobjekte angelegt, und in der `draw()`-Methode werden diese dann immer wieder gezeichnet, jedoch mit durch die Animation veränderter Transformation.

Im Konstruktor der Szene wird das zu zeichnende Band-Objekt angelegt, sowie ein `ParametricSurface`-Objekt. Der Konstruktor des `ParametricSurface`-Objekts erhält dabei eine Funktion, welche zu einem gegebenen Paar (u,v) jeweils ein Array mit drei Koordinaten [x,y,z] zurückliefert. Die übergeben Funktion definiert ein Ellipsoid, siehe z.B. http://en.wikipedia.org/wiki/Ellipsoid . Als weiteres Argument übergibt die Szene dem `ParametricSurface`-Konstruktor ein `config`-Objekt mit den Wertebereichen der Parameter u und v (u_{min}, u_{max}, v_{min}, v_{max}) sowie der gewünschten Anzahl von Segmenten in u- und v-Richtung.

Implementieren Sie nun das Modul `models/parametric.js`, welches Sie bereits als Skelett vorfinden. Orientieren Sie sich dabei an `band.js`. Werten Sie auf dem angegebenen u-v-Gitter die Fläche durch Aufruf der angegebenen Funktion aus, füllen Sie ein VBO mit diesen Punkten, und stellen Sie die Punkte in der `draw()`-Funktion mittels `drawArrays()` dar.

Zusatzaufgabe (nur für das Erreichen einer 1.0):

Verwenden Sie Ihre ParametricSurface, um noch zwei weitere parametrische Flächen zur Szene hinzuzufügen, z.B. einen Torus und eine beliebige weitere Fläche (siehe z.B. , http://www.3d-meier.de/tut3/Seite0.html)

Abgabe

Diese Aufgaben sind der erste von mehreren Teilen der Aufgabe 2. Die Abgabe der gesamten Aufgabe 2 soll via Moodle bis zu dem dort angegebenen Termin erfolgen. Verspätete Abgaben werden wie in den Handouts beschrieben mit einem Abschlag von 2/3-Note je angefangener Woche Verspätung belegt. Geben Sie bitte pro Gruppe jeweils nur eine einzige `.zip`-Datei mit den Quellen Ihrer Lösung sowie mit den ggf. geforderten Screenshots ab.

Demonstrieren und erläutern Sie dem Übungsleiter Ihre Lösung *in der nächsten Übung nach dem Abgabetag*. Die Qualität Ihrer Demonstration ist, neben dem abgegebenen Code, ausschlaggebend für die Bewertung! Es wird erwartet, dass alle Mitglieder einer Gruppe anwesend sind und Fragen beantworten können.