



Mitteilung 01 | 19 Oktober 2009 mit Ergänzungen

Allgemeines zum Kurs

Zum Inhalt

Die Grafische Datenverarbeitung hat es mit Bildern zu tun. Das kann sich jeder denken. Mit Bildern, die mit Programmen berechnet und auf Bildschirmen gezeigt, aber auch anders sichtbar gemacht werden. Diese Bilder werden berechnet & gezeigt, nicht fotografiert & gezeigt, oder gezeichnet & gezeigt.

Die Bilder können still stehen oder bewegt sein. Statisch oder dynamisch. Man mag auch interaktiv in sie eingreifen. Der Unterschied zwischen Computergrafik und Bildbearbeitung (oder Bildverarbeitung) wird immer geringer. (Frage: Was ist darüber hinaus *Computer Vision*, was mag *Computational Geometry* sein? Daraus machen wir eine erste Übungsaufgabe!)

Ist es nicht merkwürdig oder bemerkenswert: der Computer, die Maschine also, mit der wir es ständig zu tun haben, die uns ein klein wenig ans Herz gewachsen sein mag (ist es nicht so?) – der Computer heißt doch wohl so, weil er zum Rechnen da ist, zum Berechnen. Oder nicht?

Aber in der Grafischen Datenverarbeitung geht es um Bilder, also eher ums Zeichnen und Malen, sollte man denken. Will ich auch denken. Also haben wir ein Problem, zum Glück: in der Grafischen Datenverarbeitung, in der Computergrafik, geht es um einen schönen Widerspruch. Um die Aufgabe nämlich, zu rechnen statt zu zeichnen, das Rechnen aber als ein Zeichnen erscheinen zu lassen. Oder anders: die Maschine zum Zeichnen nehmen, die nicht zeichnen kann.

So ist es mit allem in der Informatik. Man macht da etwas, etwas sehr Mathematisches, das aber andererseits längst die Mathematik hinter sich gelassen hat (in welchem Sinne?), meint dabei jedoch etwas Anderes als unmittelbar das, was man da macht. (Ist das eine dunkle Formulierung?) Die Informatik, so scheint es, ist so etwas wie das große, das systematische und prinzipielle *Als-ob*. Darüber sollten wir mal nachdenken. Jede für sich, alle zusammen. Beim Mensagang, in der Straßenbahn oder in einer Vorlesung über Netzwerke oder Compiler. Wer weit ausgreifen will, mag gelegentlich in die Bibliothek gehen (was sich lohnen mag). Und nach Hans Vaihinger, *Die Philosophie des Als-Ob* suchen. Ein ziemlich altes Buch. Was kostet es im Internet?

Die Erzeugung von Bildern mit und auf dem Computer wirft zwei große Themen auf: die Modellierung und die Darstellung. Die Modellierung bezieht sich auf die *geometrischen* Eigenschaften der Gegenstände, die im Bild erscheinen sollen, also auf das „Was“. Was ist es, das im Bild erscheint?

Die Darstellung hingegen bezieht sich auf die *grafischen* Eigenschaften der Gegenstände, die im Bild erscheinen, also auf das „Wie“. Wie wird das gezeigt, das im Bild erscheint? Die Geometrie hat es mit dem unsichtbaren, die Grafik mit dem sichtbaren Aspekt der Bilder mit und auf dem Computer zu tun.

Bedenkt mal, wie es zum Bild gekommen sein mag, bevor es einen Computer gab! Und einen Computer gibt es noch gar nicht so lange. Und Bilder auf oder aus dem Computer gibt es noch weniger lange. (Beides: schon wieder Aufgaben!)

In den Grundlagen der grafischen Datenverarbeitung – mit denen wir es in diesem Kurs zu tun haben – geht es um eine erste theoretische und praktische Begegnung mit den algorithmisch beschriebenen und dargestellten Bildern. Es geht um eine Einführung in das begeisternde Gebiet der Bilder auf und mit dem Computer. Unsere Themen werden, grob gesprochen, sein: mathematische Grundlagen, geometrische Modellierung, grafische Darstellung, Rasterbilder, OpenGL. Ich werde mich in gewisser Weise auf zwei Bücher stützen (aber auch auf andere):

Francis S. Hill, Jr. & Stephen M. Kelley, *Computer Graphics Using OpenGL*. Upper Saddle River, NJ.: Pearson Prentice Hall 2007

Michael Bender & Mafred Brill, *Computergrafik. Ein anwendungsorientiertes Lehrbuch*. München: Carl Hanser 2003

Ich habe das erste Buch schon mehrmals empfohlen. Ohne allzu große Überzeugung. Auf sehr große Gegenliebe war es auch nicht gestoßen bei den Studierenden, eher auf kluge Kritik. Wenn ich es dennoch wieder empfehle, so heißt das Folgendes. Als amerikanisches Buch ist es jedenfalls relativ einfach zu lesen. Solange Ihr ein Lehrbuch studiert, ist es in Grenzen egal, was und wie es das behandelt, wozu es geschrieben ist. Das genannte Buch stützt sich auf OpenGL, was ein Vorteil ist. Es gibt aber viele andere (eine Literaturliste kommt in Kürze). Das aber solltet Ihr tun: regelmäßig ein Lehrbuch zur Hand zu nehmen. Klingt arg altmodisch, oder? Wie seht Ihr das?

Die praktische Seite des Kurses wird sich in Aufgaben zeigen, von denen ein Teil zu programmieren sein wird. Die Programmierung geschieht in C++ (oder C oder Java oder Processing) und stützt sich in den eigentlichen grafischen Anteilen auf den Standard OpenGL. Kenntnisse und Erfahrungen in C++ (oder ähnlich) sind Voraussetzung oder müssen selbst nachgeholt werden. (Ihr habt zwei Semester studiert oder mehr, könnt also erstens programmieren, zweitens in einer dieser Sprachen. Und sollte das nicht der Fall sein, so empfehle ich dringend, dieses Semester dazu zu nutzen, dass Ihr Euch in einer kleinen Gruppe organisiert, die selbständig das Programmieren in C++ an Hand von Grafik, also gleichzeitig mit OpenGL, lernt. Macht das, bitte, und lasst eher einen anderen Kurs dafür sausen!)

Zu OpenGL wird es im Rahmen unserer Veranstaltung eine allgemeine Anleitung geben, jedoch keine intensive angeleitete Schulung. (Ihr studiert Informatik oder Digitale Medien, nicht Programmieren.) Wir werden gemeinsam Schritte organisieren, die die notwendigen Fertigkeiten unter den Teilnehmenden verbreiten und verbessern sollen. Das erfordert eine gewisses Maß an Zusammenarbeit. – Hier ist eine weitere Empfehlung (auch im Netz zu finden!):

Dave Shreiner, Mason Woo, Jackie Neider & Tom Davis, *OpenGL Programming Guide*. Upper Saddle River, NJ.: Addison-Wesley 2006

Der vierstündige Kurs findet wöchentlich am **Montag** in zwei Teilen statt. Von **8:30 Uhr** pünktlich bis **10:00** in **GW2, B 2900**. Und nachmittags von **16 bis 18** Uhr in Raum **MZH 7250**.

Grob sind die Stunden in Erzählung und Übung aufgeteilt, jedoch werde ich diese Trennung nicht allzu streng nehmen. Es ist eine inhaltliche, keine zeitliche Trennung. Dadurch möchte ich größere Flexibilität erreichen. Einem Vortrag folgen, an einer Diskussion teilnehmen, eine Aufgabe allein bearbeiten, einen Text allein studieren, das alles aber auch gemeinsam mit einer Freundin, sich mit der Lösung einer Aufgabe zu befassen: das sind in etwa die Arten von Tätigkeit, die Ihr hier üben könnt und sollt. In irgendeiner Mischung, die ich nicht strikt trennen werde, weil ich es nicht will.

Die Veranstaltungszeit erstreckt sich über 14 Wochen. Wir werden uns zwölfmal gemeinsam treffen, denn zweimal bin ich zu Konferenzen weg. Diese beiden Male organisieren wir so, dass Ihr selbständig arbeitet. Ich suche zwei bis vier Freiwillige, die dazu besondere Aufgaben übernehmen. Wir haben gemeinsam entschieden, statt einer Verlängerung der Montagstreffen (bis 19 Uhr) einen ganzen Studientag als Nachholtermin zu organisieren: Samstag, 23. Januar 2010, von 10 bis 18 Uhr.

Die 14 Treffen machen in der Summe 56 Stunden aus, sagen wir also 60 Std. Bleiben 120 Std. Arbeit, damit Ihr die 6 Punkte ECTS mit gutem Gewissen einsammeln könnt, um die es hier als europäisches Maß geht (1 Punkt = 30 Std.). Unterstellen wir pro Woche etwa 6 Stunden Arbeit für die Übungsaufgaben und das Nacharbeiten des Stoffes, so kommen weitere 84 Std. zusammen, sagen wir 80 Std. Bleiben noch mal 40 Std. offen. Die nutzt Ihr zur Vorbereitung auf das Fachgespräch (dessen Termin Ihr Euch schon jetzt rot im Kalender anstreichen müsst: **16. Februar** oder **29. März!**).

Ihr dürft nicht denken, Grafische Datenverarbeitung ginge ohne Mathematik. Wir werden da zwar nichts Schreckliches zu tun kriegen, aber ich empfehle sehr, die Chance wahrzunehmen, vieles von dem aufzufrischen, was Euch mathematisch ein wenig aus dem Blick geraten sein mag.

Zum Leistungsnachweis in Grafische Datenverarbeitung

Der Kurs bringt 6 Punkte nach ECTS. Diese Zahl scheint nach der neuen Studienorganisation das Interessanteste geworden zu sein, nach dem Studierende ihr Studium ausrichten. Ihr wisst, dass 6 Punkte zu 180 Stunden Arbeit äquivalent sind. Das ist die Währung, die hier gilt. Dadurch, dass Ihr diese Menge Zeit für diesen Kurs aufbringt, bekommt Ihr die Punkte, egal, ob Ihr eine gute oder schlechte Leistung zeigt. Die Punkte sind das quantitative Maß, nach dem Eure Leistung gemessen wird. (Nebenbei: verfolgt die Kritik an der *Sauce Bolognese*, genannt Bachelor.)

Die Note, die es auf dem Leistungsnachweis zusätzlich geben wird, drückt hingegen die Qualität Eurer Leistung aus. Ohne die 180 Stunden bekommt Ihr gar nichts. Mit den 180 Stunden schafft Ihr also die notwendige Voraussetzung dafür, dass Ihr auch eine Note bekommen könnt.

Die Bedingung für den Erwerb des Leistungsnachweises in diesem Kurs lautet recht einfach und einsichtig so:

Einen Leistungsnachweis kann erlangen, wer ständig aktiv am Kurs teilnimmt und mitarbeitet, dabei die regelmäßig gestellten Übungsaufgaben bearbeitet und das Fachgespräch nach Ende der vorgesehene Treffen besteht. Alternativ kann man eine halbstündige mündliche Prüfung ablegen.

Übungsaufgaben wird es alle zwei Wochen geben. Sie sind *alle* zu bearbeiten. Die Bearbeitungen müssen bis zum jeweils angegebenen Zeitpunkt eingereicht werden. Die Abgabetermine dürfen *nicht* überschritten werden. Übungsaufgaben können allein oder in Zweiergruppen bearbeitet werden. Wer unbedingt zu dritt arbeiten will, bekommt zusätzliche Aufgaben, die zu bearbeiten sind.

Eure Abgabe muss minimale formale Vorschriften einhalten (s. Aufgabenblatt 1). Ihr gebt Text, Abbildungen, Programmtext (selektiv) ab. Alles muss in einem Dokument zusammengefasst werden, das Ihr mir im Format *pdf* als Anhang zu einer *Email*-Nachricht schickt. Alternativ: auf *Papier* ausgedruckt oder handschriftlich.

Die Übungsaufgaben erhaltet Ihr mit Kommentaren zurück. Sie werden nicht benotet, bekommen auch keine Punkte. Ihr übt. Meine Kommentare, auch wenn sie sparsam sind, sagen Euch einiges. Wenn Ihr teilnehmt (was Ihr müsst), erhaltet Ihr stets Gelegenheit, durch aktives Mitmachen ein Gefühl dafür zu entwickeln, wo Ihr steht, was geht und was nicht.

Das Fachgespräch dauert pro Person 15 Minuten. Die mündliche Prüfung dauert 30 Minuten. Über Einzelheiten verständigen wir uns zum gegebenen Zeitpunkt rechtzeitig vor Ende der Veranstaltung. Die Übungsaufgaben werden nicht benotet, da sie zur Übung anregen sollen. Ihr müsst Euch intensiv mit ihnen befassen und sie schriftlich, bzw. durch Programmierung, bearbeiten.

Rahmen-Zeitplan

19 Oktober	Einführung, Überblick. Rendering Equation. Verständigung & Absprachen zur Organisation. Bestimmung einer Tutoren-Gruppe für OpenGL – <i>Ausgabe Blatt 1</i>
26 Oktober	Erster Übungstermin zu OpenGL (Tutoren-Gruppe!) [FN in Berkeley]
2 November	Rastergrafik – <i>Ausgabe Blatt 2, Einreichung Blatt 1</i>
9 November	Zweiter Übungstermin zu OpenGL (Tutoren-Gruppe!) [FN in Krems]
16 November	Geometrie – <i>Ausgabe Blatt 3, Einreichung Blatt 2</i>
23 November	Geometrie
30 November	Geometrie – <i>Ausgabe Blatt 4, Einreichung Blatt 3</i>
7 Dezember	Die Abbildungssituation. Kamera
14 Dezember	Beleuchtungsmodellierung. Shading – <i>Ausgabe Blatt 5, Einreichung Blatt 4</i>
4 Januar	Strahlverfolgung
11 Januar	Strahlverfolgung – <i>Ausgabe Blatt 6, Einreichung Blatt 5</i>
16. Januar (Samstag)	11 bis 19 Uhr (Nachholtermin für zwei ausgefallene Treffen) Strahlverfolgung. Radiosity. Gemischtes zu Übungsaufgaben
18 Januar	z-Puffer Algorithmus
25 Januar	Splines – <i>Einreichung Blatt 6</i>
1 Februar	Abschließendes, Zusammenfassung, Rückblick, Kritik, Vorblick auf Fachgespräche
16 Februar	Fachgespräche und mündliche Prüfungen! Einen der Termine oder beide unbedingt
29 März	notieren. Festlegungen auf einen der beiden Termine erfolgen vor Weihnachten