

# Übungen

## Geometrische Szenenrekonstruktion

Sommersemester 2006

Prof. Dr. Stefan Posch



Institut für Informatik

Universität Halle

## Blatt 4

### Aufgabe 4.1

Gegeben die Matrix  $P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ .  $P_1$  bildet Punkte  $\vec{X} \in P^3$  nach  $P^2$  ab.

Welcher Form von Projektion entspricht das? (Welchen Effekt hat  $P_1$  auf Punkte etc. und wo liegt das Kamerazentrum?)

Angenommen ein Punkt  $\vec{X}$  ist nicht in Kamerakoordinaten gegeben, sondern in Weltkoordinaten. Wie kann man das Kameramodell  $P_1$  erweitern zu  $P_2$ , um dem Rechnung zu tragen (d.h. direkt von Weltkoordinaten zu Bildkoordinaten abzubilden)?

Geht man zusätzlich davon aus, daß die Skalierungsfaktoren in die beiden Achsenrichtungen nicht gleich sind (vgl. endliche CCD Kamera), wie muss man dann  $P_2$  verändern?

Letztlich kann es vorkommen, daß die Achsen der Kamera nicht mehr parallel zueinander stehen (skew), wie erhält man dann  $P_4$ .

### Aufgabe 4.2

Schreiben sie ein Programm, daß Übung 3.1 per SVD löst. Gehen sie davon aus, daß die Koordinaten der drei Punkte in einem File als float Werte je Komponente in einer Zeile stehen. Geben sie die Ebene durch diese Punkte als homogenen Vektor aus, wobei die letzte Komponente 1 (oder 0 für ideale Punkte) sein soll.

Sie können zur Berechnung der SVD z.B. die gnu scientific library (gsl) benutzen. Das ist eine C-Library, über die Sie unter <http://sources.redhat.com/gsl/> mehr Informationen finden. (Man muss

```
#include <gsl/gsl_matrix.h>
```

```
#include <gsl/gsl_vector.h>
```

```
#include <gsl/gsl_linalg.h>
```

in das Programm einbinden und beim Linken die Optionen **-lgsl -lgsliblas -lm** benutzen. )

### Aufgabe 4.3

Schreiben sie ein Programm, daß aus sechs ( $5\frac{1}{2}$ ) Punktkorrespondenzen (jeweils komponentenweise in einer eigenen Zeile aus einem file einzulesen) als Eingabe nimmt und die zugehörige Kameramatrix  $P$  als Ausgabe erzeugt (d.h. sie sollen die DLT imple-

mentieren).

#### **Aufgabe 4.4**

Optional: Wenn sie Lust haben, versuchen sie herauszufinden, wie man die Kameramatrix  $P$  dekomponieren kann, so daß die intrinsischen und extrinsischen Kameraparameter (z.B. Kamarazentrum und Brennweiten) ablesbar sind und lassen sie ihr Programm diese auch ausgeben.

(Die Zerlegung kann man mit Hilfe der RQ-Zerlegung erreichen (RQ nicht QR)).

Unter *calib.pgm* liegt ein Bild eines Kalibrieremusters. Die Kreismittelpunkte liegen genau 4cm auseinander (in Weltkoordinaten :). Messen Sie die geeignete Bildkoordinaten von einer ausreichenden Anzahl Punkten im Bild nach (unter Unix zeigt z.B. der xv Koordinaten an, wenn man die Mittlere Taste drückt) und benutzen Sie die so gefundenen Punktkorrespondenzen, um mit Ihrem Programm die Kameramatrix zu berechnen und zerlegen Sie sie danach.

Folgende Kamera-Daten wurden beim Entstehen der Aufnahme (ungefähr) gemessen:

Brennweite ca. 20mm

Kamera ca. 45 Grad nach unten geneigt

Mittlerer Abstand des Testmusters von der Kamera ca. 180cm (entlang des Boden gemessen)

Höhe der Kamera zum Boden ca. 170cm.