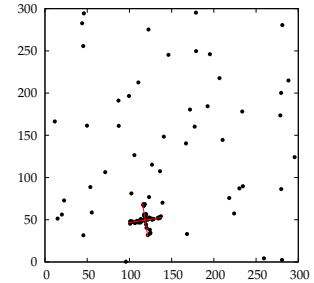


## Übungsblatt 1

Abgabe: 25.03.2013

## 1. Avogadro-Projekt

Sie arbeiten im Rahmen des Avogadro-Projektes an der genauen Positionierung der Silizium-Kugel. Dafür befindet sich eine Gravur in Form eines Pluszeichens auf der Kugel. Eine Kamera nimmt ein  $300 \times 300$  Bild davon auf. Ihr Kollege hat bereits ein Programm geschrieben, das die Bilder auswertet und aus den Reflexionen mögliche Punkte auf dem Zeichen findet. Der Algorithmus arbeitet recht zuverlässig, markiert aber auch einige zufällige Punkte im ganzen Bild. Ein Beispieldatensatz ist in der nebenstehenden Abbildung zu sehen. Ihre Aufgabe besteht darin, den Mittelpunkt des Kreuzes aus diesen Daten zu finden (rotes Kreuz=Fit).



- Begründen Sie, warum es nicht einfach möglich ist, die Aufgabe durch Fitten (least squares) einer Funktion an die Datenpunkte zu erledigen. **(2 P)**
- Die Parametrisierung des Zeichens soll über die Position  $(x_0, y_0)$  des Mittelpunktes und den Winkel  $\varphi$  zwischen  $x$ -Achse und einem Arm des Zeichens erfolgen. Welche Randbedingungen (box constraints) sind für diese Parameter sinnvoll? **(2 P)**

## 2. Programmieraufgabe

Ein Kollege, der vor Ihnen an der Aufgabe gearbeitet hat, hat bereits ein Programm geschrieben, das versucht, die Aufgabe mit Hilfe von particle swarm optimisation (PSO) zu lösen. Es funktioniert aber noch nicht, weil die Zielfunktion (objective function) bisher nur den Abstand zum ersten Punkt minimiert. Sie können das Programm mit Makefile und Beispieldateien auf <http://www.auriocus.de/PSO> herunterladen. Nachdem Entpacken übersetzt der Aufruf „make“ das Programm, testet es und stellt das Ergebnis graphisch dar.

- Ändern Sie die Zielfunktion so, dass die Anzahl der gültigen Punkte maximiert wird. Sie wissen, dass die Arme des Kreuzes  $armlength = 20$  Pixel lang sind, und dass der Algorithmus des Kollegen höchstens um  $maxdist = 2$  Punkte daneben liegt. Ein Punkt ist also gültig, wenn er maximal  $armlength$  Pixel vom Zentrum  $(x_0, y_0)$  entfernt ist und höchstens  $maxdist$  Pixel vom nächstgelegenen Arm entfernt ist. *Hinweis:* Im Quellcode befindet sich bereits eine Funktion, die den Abstand eines Punktes von einer Geraden berechnet. **(4 P)**
- Sie möchten den Mittelpunkt noch genauer bestimmen und beschließen daher, die Zielfunktion so zu ändern, dass die Summe der Abstände der gültigen Punkte vom nächstgelegenen Arm minimiert wird. (Minimieren kann erreicht werden, indem man  $-f(x)$  maximiert.) Nun funktioniert das Programm aber plötzlich nicht mehr. Welcher Wert der Zielfunktion wird erreicht, und warum? **(4 P)**
- Die Vorteile beider Methoden aus 2a und 2b erhält man, wenn man zu der Summe aus Aufgabe 2b für jeden ungültigen Punkt  $maxdist$  addiert (penalizing). Ändern Sie das Programm so, dass es in diesem Sinne funktioniert. Warum wird dadurch das Problem aus 2b vermieden? **(4 P)**
- Im Datenverzeichnis sind 20 Dateien abgelegt, die aus einer Messreihe kommen. Dabei wurde die Kugel gleichmäßig entlang einer Geraden verschoben. Wie könnte man mithilfe dieser Daten die Auflösung (precision) des Programms bestimmen? *Hinweis:* Falls Sie 2c nicht gemacht haben, können Sie die Musterlösung benutzen. Diese befindet sich in kompilierter Form für Linux als `solution_kreuzfit` mit im ZIP-File. **(2 P)**