

Praktikum 3

Nachbarschaften und Pfade

M. Thaler, 8/2014, ZHAW

1 Einführung

In der Bildverarbeitung spielen bei vielen Anwendungen Nachbarschaftsbeziehungen zwischen Pixeln eine wichtige Rolle. In diesem Praktikum lernen Sie am Beispiel eines Pfades, wie Nachbarschaftsbeziehungen effizient bestimmt und angepasst resp. konvertiert werden können. Die Verfahren zur Bestimmung für Nachbarschaftsbeziehungen lassen sich auch auf andere Algorithmen der Bildverarbeitung anwenden.

2 Lernziele

- Sie können 8-Pfade in *eindeutige* Pfade konvertieren (eindeutig: m-Nachbarschaften werden dabei in d-Nachbarschaften konvertiert).
- Sie wissen, wie solche Konvertierungsalgorithmen effizient implementiert werden können.
- Sie können die Länge von Pfaden bestimmen.

3 Aufgaben

Die Files zu diesem Praktikum finden Sie Praktikumsverzeichnis auf dem WEB.

3.1 Pfadkonvertierung

Das File `path.tif` enthält ein Binärbild mit einem 8-connected Pfad, der überall 1 Pixel breit ist, das File `test.tif` enthält ein einfaches Muster zum Testen ihrer Algorithmen. Dieser Pfad soll in einen Pfad konvertiert werden, der keine m-Nachbarschaften mehr enthält, d.h. alle m-Nachbarschaften sollen in d-Nachbarschaften konvertiert werden. Eine m-Nachbarschaft wird durch folgende Pixelanordnung beschrieben:

$$q_1 \in N_4(p), q_2 \in N_4(p) \text{ und } q_1 \in N_D(q_2)$$

- Überlegen Sie sich in einem ersten Schritt, welche Pixelkonfigurationen m-benachbart sind und wie diese Pixelkonfiguration verändert werden müssen, um sie d-benachbart zu machen. Hinweis: skizzieren Sie entsprechende Pixelanordnungen.
- Wie lassen sich Ihre Überlegungen von oben effizient implementieren? **Hinweis:** Gehen Sie vorerst von einer 3×3 -Umgebung aus, d.h. das aktuelle Pfad-Pixel p_p wird als Zentrumpixel angenommen und die Umgebung enthält alle Pixel in $N_8(p_p)$. Werden zudem alle Pixel in $N_8(p_p)$ mit

1	2	4
128		8
64	32	16

2er-Potenzen gewichtet, können alle möglichen Pixel-Muster durch eine eindeutige Zahl beschrieben werden (im wesentlichen ein Bit pro Umgebungspixel).

Überlegen Sie sich nun, welche Pixel-Muster in dieser Anwendung überhaupt vorkommen können und identifizieren Sie die *eindeutige Zahl* dieses Musters. Wenn nicht alle Muster aus der N_8 Umgebung möglich sind, müssen nur diejenigen Pixel in $N_8(p_p)$ mit einer 2er-Potenz versehen werden, die auch wirklich auftreten können. Die *Muster-Zahl* kann nun als Adresse für eine Lookup-Tabelle verwendet werden, in der abgelegt ist, ob das Zentrums-Pixel p_p gelöscht werden muss oder nicht.

Hinweis: Die Pfadanpassung kann auch mit einer 2×2 Umgebung realisiert werden.

- Besprechen Sie Ihre Lösung mit dem Dozenten und implementieren Sie anschliessend Ihren Algorithmus mit MATLAB. Überprüfen Sie visuell, ob Ihr Algorithmus richtig arbeitet.
- Der hier implementierte Algorithmus arbeitet korrekt für Pfade ohne Abzweigungen. Überlegen Sie sich, ob der Algorithmus auch noch korrekt für Pfade mit Abzweigungen arbeitet? Was könnte in diesem Fall das Problem sein?

3.2 Pfadlänge

Es ist oft notwendig die Länge eines Pfades, z.B. den Umfang eines Objektes oder die Länge einer Strecke zu bestimmen.

- Bestimmen Sie die Pfadlänge des Pfades ohne m-Nachbarschaften, den Sie aus dem Bild im File `path.tif` bestimmt haben. Erweitern Sie dazu Ihr MATLAB-Programm.
- Bestimmen Sie die Pfadlänge direkt aus dem Originalbild, indem Sie die Regeln für m-Nachbarschaften beachten. Implementieren Sie den Algorithmus wiederum mit MATLAB und vergleichen Sie das Resultat mit dem Resultat des Pfades ohne m-Nachbarschaften.

3.3 Pfadkonvertierung $d \rightarrow m$

Der Pfad in `path.tif` ist an einigen Stellen nur d-benachbart. Implementieren Sie einen Algorithmus, der an diesen Orten ein zusätzliches Pixel so einfügt, sodass diese Stellen m-benachbart werden. Überprüfen Sie wiederum visuell, ob Ihr Algorithmus richtig arbeitet.