3.3 Segmentierung

In diesem Abschnitt werden wir uns mit dem Problem der *Segmentierung* von Bildern beschäftigen, d.h. der Aufgabe des automatischen Erkennens von Teilen des Bildes mit bestimmter Struktur. Man unterscheidet meist in

- *Kantenbasierte Segmentierung*, bei der man Kanten, also Kurven / Oberflächen entlang derer das Bild unstetig ist, zu erkennen versucht, unabhängig davon welche Bereiche sie abgrenzen.
- *Objektbasierte Segmentierung*, bei der man verschiedene Objekte (also unterschiedliche Teilgebiete) zu erkennen versucht.

Die Grenzen zwischen den beiden Techniken sind oft sehr fliessend, da ja Kanten meist mit Objekträndern übereinstimmen. Interessanter ist die Unterscheidung eher für die Darstellung der Unbekannten. In diesem Fall wollen wir ja keine Bilder (Funktionen) mehr berechnen, sondern aus den Bildern geometrische Objekte, entweder Teilmannigfaltigkeiten Γ mit Kodimension eins (Kanten) oder Teilgebiete D(Objekte). Da in der (numerischen) Praxis die Berechnung von Mengen schwierig ist (insbesondere wegen fehlender Vektorraumstruktur) greift man gerne auf verschiedene Darstellungen über Funktionen zurück. Die wichtigsten Beispiele sind:

- *Parametrisierung* von Mannigfaltigkeiten durch Abbildungen $\gamma : \mathbb{R}^{d-1} \to \Gamma \subset \mathbb{R}^d$.
- *Indikatorfunktion* χ eines Gebiets D, mit $\chi(x) = 1$ für $x \in D$ und $\chi = 0$ für $x \notin D$.

Wir beginnen mit einfachen Filtern und Thresholding-Algorithmen, und werden dann einige Ansätze basierend auf Variationsmethoden diskutieren.

3.3.1 Thresholding

Das einfachste und vielleicht intuitivste Verfahren zur objektbasierten Segmentierung ist das Thresholding. Dabei legt man einen Intensitätswert t fest und klassifiziert das Bild einfach in zwei Teilbereiche

$$\Omega_1 = \{x \mid f(x) < t\}, \qquad \Omega_2 = \{x \mid f(x) \geq t\}. \tag{3.24}$$

Dieses Verfahren funktioniert sehr gut wenn man etwa ein dunkles Objekt auf hellem Hintergrund sucht. Bei komplexeren Bildern ist das Verfahren sehr limitiert, kann aber zumindest oft gute Startwerte für andere Segmentierungstechniken liefern oder

wird oft als Teilschritt in andere Algorithmen eingebaut. Um einen passenden Wert von t zu finden, kann man wieder das Histogramm betrachten, idealerweise gibt es klare lokale Minima oder sogar Bereiche mit Löchern im Histogramm, die sich dann für die Wahl von t anbieten.

Man sieht leicht, dass man Thresholding (mit $D = \Omega_1$) als Variationsproblem für die Indikatorfunktion schreiben kann als

$$\int_\Omega \chi(x)(f(x) - t)\ dx \to \min_\chi . \tag{3.25}$$

Dieses Problem kann man sogar als lineare Optimierung relaxieren, lässt man v mit $0 \leq v(x) \leq 1$ zu, dann ist auch das Minimum von

$$\int_\Omega v(x)(f(x) - t)\ dx \to \min_v \tag{3.26}$$

dasselbe. Ein Problem beim einfachen Thresholding ist dass auch Pixel, die nur durch Rauschen eine andere Farbe erhalten, segmentiert werden. Dies kann man durch Regularisierung vermeiden. Ein einfaches und häufig verwendetes Zielfunktional ist die Länge / Oberfläche der Kanten. Wie wir später noch aus der Co-Area Formel sehen werden, ist diese gleich der totalen Variation der Indikatorfunktion und man erhält das Optimierungsproblem

$$\int_\Omega \chi(x)(f(x) - t)\ dx + \alpha \int_\Omega |\nabla \chi|\ dx \to \min_\chi . \tag{3.27}$$

Ändert man hier das Gebiet D um kleine isolierte Bereiche, z.B. Würfel mit Kantenlänge R, dann wird das erste Funktional (ein Volumsintegral) höchstens in der Ordnung R^d grösser, während das Regularisierungsfunktional um einen Term der Ordnung αR^{d-1} kleiner wird. Für $R < \mathcal{O}(\alpha)$ wird dadurch das Gesamtfunktional kleiner, das Minimum wird also nicht diese Rauschbereiche segmentieren. Man sieht hier auch gleich, dass der Regularisierungsparameter α hier die minimale Grösse der zu segmentierenden Objekte festlegt.

Mögliche Anwendungsbereiche von Thresholding-Verfahren sind helle Bereiche in der Emissionstomographie und auch die Fluoreszenzmikroskopie, in der die interessanten Objekten in rot oder grün gefärbt werden, und damit durch Segmentierung des entsprechenden Farbkanals gefunden werden können.

3.3.2 Kantendetektoren

Kantendetektoren sind meist einfache Differentialoperatoren angewandt auf das Bild. Die Idee dabei ist, dass Ableitungen an Kanten besonders gross sind und damit

die Kanten in den entstehenden Funktionen besonders hohe Werte annehmen. Ein einfacher und doch oft verwendeter Filter ist der Sobel-Filter

$$S(f)(x) = \|\nabla f(x)\| \tag{3.28}$$

Eine andere Möglichkeit, basierend auf zweiten Ableitungen, ist der Marr-Hildreth Filter

$$M(f)(x) = \Delta f(x) = \sum_i \partial_{x_i} f(x). \tag{3.29}$$

Da bei verrauschten Bildern die grössten Ableitungen durch das Rauschen auftreten, kombiniert man solche Filter normalerweise mit einem Preprocessing-Step in dem man das Bild glättet, typischerweise durch Faltung mit einem Gauss-Kern G, man wendet also den Filter auf $G*f$ an. Bei der Faltung werden Kanten zwar verschmiert, die Ableitung sollte aber dennoch ihre grössten Werte noch an den ursprünglichen Kantenorten haben. Das Rauschen wird hingegen unterdrückt.

3.3.3 Clusteralgorithmen

Bei Clusteralgorithmen sucht man Objekte, die möglichst ähnlich sind, oft in dem die Intensität nahe an einem gemeinsamen Mittelwert ist. Üblicherweise werden Clusteralgorithmen, die eigentlich aus der Datenanalyse stammen, diskret formuliert. Wegen der Einheitlichkeit werden wir sie aber auch kontinuierlich formulieren. Der bekannteste Cluster-Algorithmus ist der K-Means Algorithmus, bei dem man versucht das Bild in K Klassen einzuteilen, bei denen die Intensität nahe einem Mittelwert μ_k ist. Wir brauchen dafür K Indikatorfunktionen χ_k und minimieren

$$J(\chi_1, \ldots, \chi_K, \mu_1, \ldots, \mu_k) = \sum_{k=1}^{K} \int_\Omega \chi_k(x)(f(x) - \mu_k)^2 \, dx \tag{3.30}$$

unter der Nebenbedingung

$$\sum_{k=1}^{K} \chi_k(x) = 1. \tag{3.31}$$

Man sieht aus der Optimalität sofort, dass μ_k der Mittelwert über die k-te Klasse ist, d.h.

$$\mu_k = \frac{\int_\Omega \chi_k(x) f(x) \, dx}{\int_\Omega \chi_k(x) \, dx}. \tag{3.32}$$

Eine Weiterentwicklung von K-Means ist der Fuzzy C-Means Algorithmus, bei dem man die exakte Zugehörigkeit zu einer Klasse (die Indikatorfunktion) durch eine

Wahrscheinlichkeit $v_k(x)$der Zugehörigkeit

$$J(v_1, \ldots, v_K, \mu_1, \ldots, \mu_K) = \sum_{k=1}^{K} \int_\Omega v_k(x)^q (f(x) - \mu_k)^2 \, dx \tag{3.33}$$

wieder mit der Nebenbedingung

$$\sum_{k=1}^{K} v_k(x) = 1. \tag{3.34}$$

Cluster-Algorithmen zur Segmentierung suchen nur nach Regionen ähnlicher Intensitätswerte, haben aber keine Information über den örtlichen Zusammenhang, auch hier können einzelne verrauschte Pixel entfernten Regionen zugeordnet werden. Um sinnvolle Segmentierungsergebnisse zu erhalten, kann man wieder eine Regularisierung mit der Länge / Oberfläche der Kantenmenge, d.h. des Rands des segmentierten Clusters hinzufügen. Im Fall $K = 2$ ergibt dies den Chan.Vese Algorithmus, die Minimierung von

$$J(v, \mu_1, \mu_2) = \int_\Omega v(x)(f(x)-\mu_1)^2 \, dx + \int_\Omega (1-v(x))(f(x)-\mu_2)^2 \, dx + \alpha \int_\Omega |\nabla v(x)| \, dx. \tag{3.35}$$