DISSERTATION

Snakes für Aufgaben der digitalen Photogrammetrie und Topographie

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften

unter der Leitung von

O.Univ.Prof. Dr. Karl Kraus

Institut für Photogrammetrie und Fernerkundung
der Technischen Universität Wien (E122)

eingereicht an der Technischen Universität Wien,
Fakultät für Technische Naturwissenschaften und Informatik

von

DI. Martin Kerschner
Matrikelnummer 8825632
Haberlgasse 22/14
A-1160 Wien

Wien, am 5.6.03

Martin Kerschner

Diese Arbeit wurde vom Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) unter der Projektnummer P13725-MAT unterstützt.

Danksagung:

Ich möchte mich bei Prof. Karl Kraus bedanken für die Betreuung dieser Dissertation, sein stetes Bemühen um den finanziellen Rückhalt während der Forschungsarbeiten und insbesondere für die Unterstützung zur Fertigstellung nach meinem Berufswechsel.

Ebenso möchte ich Prof. Horst Bischof für die rasche Begutachtung sowie für seine fachlichen Rückmeldungen danken.

Ich möchte meinen Eltern danken, dass sie mir das Studium ermöglicht haben und auch die Erstellung der Dissertation unterstützt haben.

Meinen ehemaligen Kollegen am I.P.F. sowie meinen neuen Kollegen gilt ebenso Dank für die Unterstützung.

Kurzfassung

Snakes gelten in der *Computer Vision* als ein allgemein bekanntes Verfahren zum automatisierten (halbautomatischen) Extrahieren von Kanten und Linien in digitalen Bildern. Ausgehend von einer groben Näherung der Form der gesuchten Kurve und ihrer Lage im Bild verbessert die *Snake* ihre Form und Lage durch Optimierung eines komplexen Energiefunktionais. Dabei soll sie die gesuchte Kurve detailgetreu nachbilden. Die Stärken von *Snakes* liegen in ihrer Robustheit gegenüber Rauschen im Bild und Lücken in der abgebildeten Kurve. Zahlreiche kritische Konstellationen wurden allerdings publiziert, die ihr Konvergenzverhalten und ihre Robustheit beeinträchtigen. Es soll beurteilt werden inwieweit sich *Snakes* auch angesichts ihrer Schwächen für Aufgaben in der digitalen Photogrammetrie und Topographie eignen.

Schwächen der Methode waren für viele dem ursprünglichen Aufsatz folgende Publikationen Anlass zur Verbesserung der Methode. Ein Schwerpunkt dieser Arbeit ist eine umfassende Zusammenstellung und Beurteilung erweiterter und verbesserter Ansätze. Weiters werden einige im Rahmen der Dissertation erarbeitete Bausteine zu einem für die jeweilige Anwendung optimalen Ansatz vorgestellt.

Die Probleme der *Snakes* können auf drei verschiedene Arten gelöst werden:

- Wahl des Optimierungsalgorithmus: Fünf verschiedene Optimierungsmethoden werden präsentiert. Neben der ursprünglichen Lösung über Variationsrechnung werden die Lösungen mittels dynamischer Programmierung, mittels kleinster-Quadrate-Ausgleichung, mittels eines *Level-Set*-Ansatzes sowie mittels *Simulated Annealing* untersucht.
- Anpassung der Energiefunktion: Die Energiefunktion besteht aus mehreren Termen für unterschiedliche Zwecke. Die vielfältigen publizierten Formulierungen der Energieterme werden analysiert. Insbesondere die internen Energieterme, die für eine glatte Form der *Snake* verantwortlich sind, werden kritisch betrachtet. Neue Formulierungen werden vorgeschlagen, die das Schrumpfen der Kurve unterbinden sollen.
- Einsatz einer speziellen Anwendungsstrategie: Hierarchische Strategien, eine Strategie vom Groben ins Feine oder die Optimierung zwischen zwei Punkten wurden in der Literatur bereits vorgeschlagen. Im Rahmen dieser Arbeit wurden andere Strategien entwickelt: die Unterteilung in Segmente sowie die Verwendung zweier *Snakes (Twin Snakes).*

Die Arbeit schließt mit einigen möglichen Anwendungen von *Snakes* im Bereich der digitalen Photogrammetrie und der Topographie. Neben der Kanten- und Linienextraktion im dreidimensionalen Raum wird der Einsatz von *Snakes* für die Schnittliniensuche bei der Orthophotomosaikierung sowie für die Detektion von Kanten in Geländemodellen vorgeschlagen.

Abstract

Snakes are well known in computer vision as an automated (semiautomatic) tool for extracting edges and lines in digital images. Starting from a rough approximation of the desired curve regarding both its shape and its location in the image, the snake improves its state by optimizing a complex energy functional. The aim is to reconstruct the desired curve in detail. The benefits of snakes lie in their robustness with respect to noise in the image and gaps in the image curve. However, numerous critical constellations were published, which impair their convergence behavior and their robustness. Having their weaknesses in mind we will evaluate whether snakes are suitable for applications in digital photogrammetry and topography.

After the first publication of snakes many papers followed presenting improvements in order to overcome some weaknesses. As an emphasis of this thesis, a comprehensive comparison and evaluation of extended and improved snake models is given. Further, some components developed in the course of the thesis are presented which make the snake's energy function optimal for a specific application.

The problems of snakes can be solved by three different approaches:

- Selecting an optimization algorithm: Five different optimization methods are presented. Beside the original solution using variational calculus, the solutions by means of dynamic programming, by least squares adjustment, by using a level set method as well as by using a simulated annealing method are considered.
- Adapting the energy function: The energy function consists of several terms for different purposes. Various published formulations of energy terms are analyzed. In particular, internal energy terms being responsible for a smooth shape of the snake are examined critically. Alternative formulations are suggested which prevent the snake from shrinking.
- Employing a special application strategy: Hierarchical strategies, strategies from coarse to fine, or optimizing the path between two points were already suggested in literature. In this thesis, further strategies are developed: partitioning the snake into segments as well as using two coupled snakes (twin snakes).

The thesis closes with some possible applications of snakes in the context of digital photogrammetry and topography. Beside edge and line extraction in the three-dimensional space, we use snakes for detecting the optimal seam line in orthophoto mosaics as well as for extracting fold lines in terrain models.

Inhalt

1 Einleitung

1.1 Motivation

Digitale Photos repräsentieren Abbilder der realen Welt. Die Information über die abgebildeten Objekte ist als einfache zweidimensionale Verteilung von Helligkeitswerten oder Farbwerten verspeichert. Mit der Interpretation digitaler Bilder und der Ableitung strukturierter Information aus den Bildern beschäftigt sich die Forschungsdisziplin *computer vision* (Maschinelles Sehen).

Die Photogrammetrie dient der Zusammenführung der aus Photographien abgeleiteten Daten zur Rekonstruktion dreidimensionaler Objekte. Dabei macht sie sich Methoden der *computer vision* zunutze, um zu strukturierten Informationen aus digitalen Photos zu kommen. In der modernen Photogrammetrie spielen linienhafte Informationen eine bedeutende Rolle. Die Ableitung des Verlaufs von Kurven in Bildern ist dazu die Voraussetzung. Diese Daten werden bisher oft nur manuell erfasst. Automatisierte Methoden bieten sich zur Unterstützung der Erfassung an.

In der *computer vision* gilt die Methode der *snakes (active contours)* als ein allgemein bekanntes Verfahren zum automatisierten (halbautomatischen) Extrahieren von Kanten und Linien in digitalen Bildern. Dabei werden mehrere verschiedenartige Kriterien, die für die gesuchten Kanten bzw. Linien typisch sind, in einer Energiefunktion kombiniert. Mittels Energieoptimierung verformen und verschieben sich die Kurven bis die gesuchte Kante/Linie im Bild gefunden ist. *Snakes* sind ein verhältnismäßig robustes Extraktionswerkzeug für ungleichmäßig stark ausgeprägte Kanten, auch in verrauschten Bildern.

Die Anwendungsmöglichkeiten für *snakes* sind breit gestreut. Sie reichen von

- Kanten- oder Linienextraktionsaufgaben (z.B. Kass et al, 1987)
- über Extraktion von Regionen (z.B. Ronfard, 1994; Zhu, Yuille, 1996; Ivins, Porill, 1998 oder Torre, Radeva, 2000)
- über Zuordnungsaufgaben
  - zweier Bilder zueinander *(feature based matching)*
  - oder eines Modells zu einem Bild (Cootes et al., 1995)
- über Aufgaben der Verfolgung von Objektkonturen
  - in Zeitreihen für Bewegungsverfolgung *(motion tracking)* und Bewegungsanalyse (z.B. Delagnes et al., 1995 oder Hoch, Litwinowicz, 1996)
  - oder in *slices* (Schichten) von volumetrischen Rastermodellen (z.B. Cohen, Cohen, 1993)
- bis zu Verdrängungsaufgaben bei der kartografischen Generalisierung (Meier, 2000).

In medizinischen Anwendungen werden *snakes* bereits praktisch eingesetzt (McInerney, Terzopoulos, 1996), etwa bei der Größenbestimmung der Köpfe von Föten in Ultraschallbildern (Zimmer, Akselrod, 2000), oder bei der Rekonstruktion von Blutgefäßen in IVUS (*intravascularultrasound)* Ultraschallaufnahmen (Kovalski et al., 2000) oder für die Detektion von Herzkammerdefekten in Echokardiogrammen (Lassige et al., 2000).

Auch für photogrammetrische Anwendungen wurde für manche Aufgaben der Einsatz von *snakes* vorgeschlagen:

- Die ersten Anwendungen für Merkmalsextraktion in der Photogrammetrie mittels *snakes* zeigte Eberhard Gülch bereits wenige Jahre nach der ersten Veröffentlichung der Methode auf (Gülch, 1990).
- Gülch (1995) verwendete *snakes* auch für die Detektion von Signalen für die automatisierte absolute Orientierung in der Aerotriangulation.
- Die Extraktion von 3D-Kurven durch die simultane Optimierung mehrerer 2D-*snakes* in einem Bildverband wird von Trinder, Li (1996) bzw. Grün, Li (1997) präsentiert.
- In photogrammetrischen Anwendungen sind *snakes* unter anderem für die Extraktion der folgenden Merkmale in Luftbildern geeignet:
  - Straßen (z.B. Grün, Li, 1997, Mayer et al., 1998)
  - Felder (z.B. Torre, Radeva, 2000)
  - Gebäudeumrisse (Martine, 2001)

Die Vielzahl an Veröffentlichungen, die Verbesserungen der Methode versprechen, zeigt, dass immer noch Probleme bestehen, die vielleicht den großen Durchbruch in der Praxis gehemmt haben. Es gibt kein allgemeines *snake*-Modell, das für alle Arten von Bildern, für alle Typen von Kurven, für beliebige Maßstäbe, etc. optimal geeignet ist. Im Gegenteil, jede spezielle Anwendung von *snakes* erfordert die Anpassung der Methode um bestmögliche Erfolge zu erzielen. In vielen Veröffentlichungen zum Thema *snakes* werden daher spezielle Erweiterungen und Anpassungen für bestimmte Anwendungen vorgeschlagen.

In dieser Arbeit sollen mögliche Anwendungsfelder in der digitalen Photogrammetrie und Topographie untersucht werden. Wenngleich Versuche der Anwendung in diesen Bereichen bereits publiziert wurden, steht der große Einsatz von *snakes* in kommerziellen photogrammetrischen Systemen noch aus. Die Gründe dafür sollen in dieser Arbeit aufgezeigt werden.

Ausgehend von einer umfassenden Analyse des *snake*-Modells und publizierter Verbesserungen wird in dieser Arbeit beurteilt, für welche Aufgaben sich der praktische Einsatz lohnt und wie sich das *snake*-Modell für die Anwendung anpassen lässt. Dazu werden Vorschläge zur besseren mathematischen Formulierung der Energieterme gemacht, die das Schrumpfen von *snakes* verhindern. Ein zentraler Bestandteil der Arbeit ist die Vorstellung von Strategien, nach denen *snakes* eingesetzt werden können, um die verbleibenden Probleme zu lösen.

1.2 Aufbau der Arbeit

Die Arbeit beginnt mit einem Theorieteil, in dem in drei Kapiteln die Methode der *snakes* präsentiert wird. Zuerst wird in Kapitel 2 der generelle *snake*-Ansatz ausgehend von Kass et al. (1987) beschrieben. Die zu Grunde liegende Energiefunktion und die drei verwendeten Typen von Energietermen werden vorgestellt. Am Ende dieser Einführung in die Theorie werden bereits Stärken und Schwächen erwähnt.

Kapitel 3 widmet sich der verschiedenen Energieoptimierungsalgorithmen. Manche dieser Algorithmen sind auf bestimmte Klassen von Kurven beschränkt (z.B. geschlossene Polygone). Derartige Einschränkungen werden in den entsprechenden Abschnitten erwähnt. Dieses Kapitel schließt mit einer Gegenüberstellung der Stärken und Schwächen der verschiedenen Optimierungsmethoden.

In Kapitel 4 werden die einzelnen Energieterme im Detail analysiert. Diverse alternative Formulierungen der Energieterme werden präsentiert. Es wird ein neuer Vorschlag zur Formulierung der internen Energie gemacht, der das Schrumpfen der *snake* verhindert. Zusätzlich wird ein im Rahmen der Dissertation entwickelter externer Energieterm präsentiert, der es erlaubt, dass zwei gekoppelte *snakes (twin snakes)* eine parallele Lage und Form annehmen.

Der zweite Teil (Kapitel 5) widmet sich der Strategien, nach denen sich *snakes* effizient einsetzen lassen. Automatisierte Methoden erreichen selten eine zufrieden stellende Kombination aus Automatisierungsgrad einerseits und Vollständigkeit, Zuverlässigkeit und Genauigkeit der Ergebnisse andererseits. Mit den präsentierten Strategien soll versucht werden, die Anzahl an richtig extrahierten Elementen zu erhöhen, auf der anderen Seite aber auch eine verlässliche Diagnose des Scheiterns zu finden.

Es wird die Frage erörtert, inwieweit die *snake* autonom erkennen kann, ob und in welchen Bereichen sie erfolgreich war. Dazu wird eine Strategie präsentiert, wie eine *snake* (für einzelne Segmente) lokale Minima verlassen kann um das globale Minimum zu finden. Weiters wird eine Strategie für Doppellinien *(twin snakes)* präsentiert. Sie eignen sich für Aufgaben in der Photogrammetrie, wo vielfach parallele Kanten zu extrahieren sind.

Der dritte Teil (Kapitel 6) präsentiert Anwendunaspotenziale von *snakes* in der digitalen Photogrammetrie und Topographie:

- die Erweiterung des Modells auf die Extraktion von Kanten und Linien in einem Verband von Photos mit bekannten Orientierungen
- die Anpassung der Methode für die Suche von Schnittlinien in der Orthophotomosaikierung
- die Anpassung der Methode für die Extraktion von Geländekanten in dichten topographischen Punktdaten

Die Arbeit schließt mit einer Zusammenfassung.

1.3 Automation in der digitalen Photogrammetrie

Ein Schwerpunkt der Forschung in der Photogrammetrie galt in den letzten Jahren der Automation verschiedenster Auswerteverfahren. Die rasante Steigerung der Leistung moderner Computer (sowohl in Rechengeschwindigkeit als auch bei der Datenspeicherung) in erschwinglichem Preisrahmen begünstigte die Entwicklung der digitalen Photogrammetrie. In ähnlicher Weise entwickelte sich die geometrische Auflösung von Satellitenbildern. Neue hoch auflösende Sensoren mit Bodenauflösungen im Bereich eines Meters und sogar darunter versprechen weitere Anwendungsfelder für automatische Verfahren zur Extraktion von Information und lassen damit die Wissenschaften Fernerkundung und digitale Photogrammetrie enger zusammenwachsen.

Viele Schritte in der photogrammetrischen Arbeitskette werden bereits mehr oder weniger automatisiert durchgeführt. Einen aktuellen Überblick des Stands der Automatisierung photogrammetrischer Arbeitsschritte bietet Rottensteiner (2001, Seiten 4-6).

Der Automatisierungsgrad einzelner Prozesse kann sehr unterschiedlich sein. Vielfach müssen geeignete Startwerte vorgegeben werden. Oft kann/muss während des Prozesses eingegriffen werden. Manche Entscheidungen werden an den Operateur delegiert. Schließlich muss meist das Ergebnis kontrolliert und vielfach auch korrigiert werden.

Photogrammetrische Auswertesysteme können bezüglich der Automation der Arbeitsschritte und der eingesetzten Prozesse in folgende Gruppen eingeteilt werden (Gülch, 2000):

1. Interaktive Systeme: Die Aktionen werden rein manuell durchgeführt.
2. Halbautomatische (semiautomatische) Systeme: Das System läuft unter der Kontrolle eines Operateurs. Dabei sollen mühsame standardisierbare Aktionen automatisiert werden, während komplexe Entscheidungen dem Operateur überlassen werden.
3. Automatisierte Systeme: Das System arbeitet ohne Interaktion des Operateurs. Startwerte müssen eventuell vorgegeben werden. Die Ergebnisse müssen vom Operateur auf Richtigkeit und Vollständigkeit geprüft werden.
4. Autonome Systeme: Das System arbeitet ohne jegliche Interaktion des Operateurs und kontrolliert in einem Eigendiagnoseschritt selbstständig die Ergebnisse.

Bei der Konzeption eines automatisierten Verfahrens ist es oft vorteilhaft die Zielsetzung bezüglich Automatisierungsgrad vorweg abzuklären. Verfahren, die vollautomatisch konzipiert sind, bieten dem Operateur meist wenige Möglichkeiten in den Prozess einzugreifen. Vielfach bleibt es dem Anwender überlassen, lediglich das Ergebnis des automatischen Prozesses zu kontrollieren und gegebenenfalls manuell zu korrigieren oder zu vervollständigen.

Das Ergebnis eines Merkmalsextraktionsprozesses muss bezüglich verschiedener Qualitätskriterien geprüft werden (McKeown et al., 2000):

- Vollständigkeit: Nicht alle der gewünschten Merkmale werden gefunden. Der Prozess begeht Fehler erster Art: Hypothesen werden fälschlicherweise verworfen *(false negative).*
- Richtigkeit: Manche der detektierten Ergebnisse sind keine gewünschten Merkmale. Der Prozess begeht damit Fehler zweiter Art: Falsche Hypothesen werden akzeptiert *(false positive).*
- Genauigkeit: Die geometrische Genauigkeit der richtig extrahierten Ergebnisse *(true positive)* muss ebenso kontrolliert werden.

Die Tendenz in der Forschung, die Produktivität in der Photogrammetrie durch Automatisierung zu steigern, möchte ich an dieser Stelle auch aus sozialpolitischer Sicht hinterfragen. Das Ziel der Forschung sollte nicht darin bestehen, den photogrammetrischen Auswerter überflüssig zu machen. Vielmehr sollte er in seiner Arbeit unterstützt werden, indem er von lästigen, mühsamen und leicht automatisierbaren Tätigkeiten befreit wird. Auch werden die Anforderungen an Operateure, die automatische Systeme bedienen, nicht sinken, was von vielen Forschern versprochen wird. Weiterhin werden auch gut ausgebildete Fachleute notwendig sein um die Ergebnisse automatischer Prozesse beurteilen zu können, solange diese nicht autonom und zuverlässig die Eigendiagnose durchführen können.

Es zeigt sich in der Praxis, dass für Aufgaben, in denen 100% Richtigkeit und Vollständigkeit gefordert sind, vollautomatische Prozesse kaum zu einer Steigerung der Produktivität führen. Selbst wenn nur 5% der Ergebnisse falsch sind und nur 5% der gesuchten Elemente nicht gefunden wurden, müssen diese Fehler anschließend in mühsamer Detailarbeit gesucht werden, damit sie korrigiert werden können. Das Kontrollieren der Ergebnisse kann leicht ebenso viel Zeit erfordern wie die gesamte manuelle Erstellung. Es ist auch leicht zu verstehen, dass die Motivation von Mitarbeitern für diese Aufgaben gering ist, was wiederum zu einer höheren Fehlerquote führt.

Ein vollautomatischer Prozess wird nur dann Akzeptanz in der Praxis finden, wenn er nach obiger Definition autonom arbeitet, d.h. wenn er auch selbstständig und zuverlässig seine Ergebnisse überprüft. In diesem Fall kann der Prozess den Operateur zur Kontrolle an die Stellen führen, wo er unsichere Ergebnisse extrahiert hat oder wo er Ergebnisse mit zu geringer Sicherheit verwerfen will.

Diese Gedanken spielen eine wesentliche Rolle, wenn im zweiten Teil Anwendungsstrategien und im dritten Teil potenzielle Anwendungsfelder von *snakes* in der Photogrammetrie und Topographie erörtert werden.

Die Methode der *snakes,* auf die diese Arbeit aufbaut, ist für halbautomatische Systeme konzipiert. Der Operateur gibt grobe Startwerte vor und steuert die automatische Feinpositionierung der *snake* (vgl. Kapitel 2). Für manche Aufgaben kann die Methode auch in automatisierten oder gar in autonomen Systemen eingesetzt werden (z.B. Abschnitt 6.2).

1.4 Definition von Fachausdrücken

Im letzten Abschnitt der Einleitung sollen einige fundamentale Fachausdrücke eingeführt werden, die im Folgenden verwendet werden. Auch wenn die Bedeutung mancher dieser Ausdrücke klar scheint, soll die Bedeutung im Kontext dieser Arbeit definiert werden um Missverständnisse zu vermeiden.

Viele Fachausdrücke sind dem Englischen entnommen und wurden oft bewusst nicht übersetzt. Sie werden in dieser Arbeit *kursiv* geschrieben, wobei auch die Groß-/ Kleinschreibung dem englischen Original entspricht.

1.4.1 Kante, Linie, Kurve, Polygon, Polygonzug, Region, etc.

Im Fachbereich der digitalen Bildverarbeitung und Mustererkennung spielen Bildkanten eine wichtige Rolle. Kanten *(edges)* sind definiert als Folge von Bildpunkten, an denen Unstetigkeitsstellen in den Grauwerten (oder in den Farbwerten) auftreten. Das Modell, das für Bildkanten verwendet wird, wird auch step edge genannt (Abb. 1-1 (a)). Bildkanten sind klar zu unterscheiden von Geländekanten, die Diskontinuitäten in der ersten Ableitung (der Neigung) darstellen. Mit Linien *(lines)* sind in digitalen Bildern schmale Bänder gemeint (wenige Pixel breit) mit mehr oder weniger homogenen Grau-/Farbwerten innerhalb des Bands und begrenzt von zwei mehr oder weniger parallelen Kanten (Abb. 1-1 (b)). Unter Extraktion linienhafter Elemente soll im Folgenden sowohl Kantenextraktion als auch Linienextraktion verstanden werden.

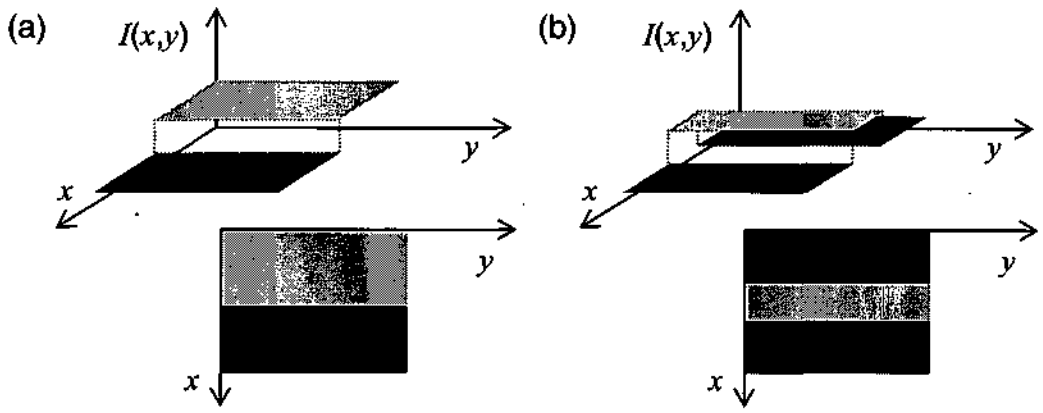


Abb. 1-1: Modell einer (a) Bildkante und einer (b) Linie im Bild mit *x* und *y* den Koordinatenrichtungen der Bildzeilen und -spalten und *I(x,y)* der Bildfunktion (z.B. der Grauwert eines Pixels).

Leider werden die Bezeichnungen Kante *(edge)* und Linie *(line)* in der Literatur nicht konsequent verwendet. In dieser Arbeit sollen die Bezeichnungen durchwegs wie oben definiert verstanden werden.

Ein beliebig geformter, längerer Pfad durch das Bild soll als Kurve bezeichnet werden. Kurven können durch *splines* angenähert werden oder in grober Näherung durch einen Polygonzug.

Kurven können geschlossen sein. Sie umschließen oft Regionen *(regions)* ähnlicher Charakteristik (z.B. mit homogenem Grauwert oder homogener Farbe). In diesem Fall wird die Begrenzung der Region oft *contour* genannt. Daraus leitet sich die Bezeichnung *active contours* ab, die in der Literatur als Synonym für *snakes* verwendetet wird. Geschlossene Kurven werden durch Polygone approximiert.

1.4.2 Zielkurve

Unter Zielkurve soll in dieser Arbeit die im Bild zu extrahierende Kante bzw. Linie verstanden werden.

1.4.3 *Low Level, High Level, Top-Down, Bottom-Up, Model-Driven, Data-Driven*

Automatische Prozesse werden häufig abhängig von den verwendeten Daten, Modellen und Zusatzinformation in Kategorien *(levels)* eingeteilt. Ein *low-level* Prozess setzt auf Daten auf und versucht, bestimmte Merkmale in den Daten zu finden. Ein *bottom-up* Algorithmus versucht, die Ergebnisse eines *low-level* Prozesses mittels eines *high-level* Prozesses zu komplexeren Modellen zu verbinden. *Bottom-up* Algorithmen sind gleichzusetzen mit *data-driven* Algorithmen.

Umgekehrt setzen *top-down* Algorithmen die Kenntnis eines Modells voraus und versuchen, dieses Modell in den Daten zu verifizieren bzw. das Modell in die Daten einzupassen. Daher werden diese Ansätze auch *model-driven* genannt.

2 Theorie der *Snakes*

Zunächst soll ein Überblick gegeben werden, wo sich der *snake*-Ansatz im Vergleich zu anderen Verfahren der Extraktion linienhafter Elemente in digitalen Bildern einreiht. Das Schema in Abb. 2-1 zeigt den Versuch einer Unterteilung:

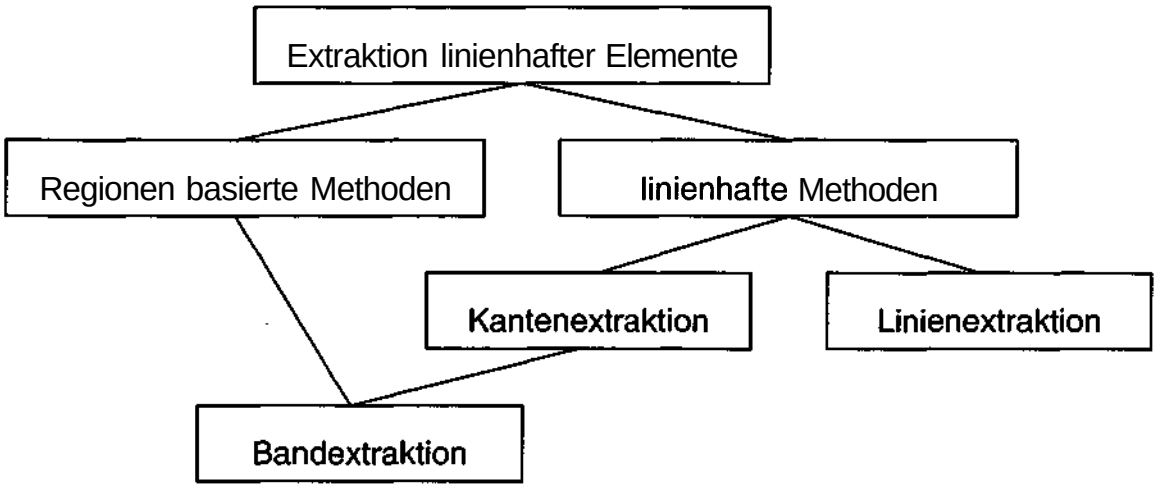


Abb. 2-1: Unterteilung von Verfahren zur Extraktion linienhafter Elemente

Ansätze zur Extraktion linienhafter Elemente können nach ihrer Strategie in Regionen basierte Methoden und linienhafte Methoden eingeteilt werden. Regionen basierte Methoden versuchen aufgrund von Ähnlichkeitsmaßen, benachbarte Pixel zu Regionen zusammenzufassen. Die Grenzen zwischen den Regionen fallen dabei indirekt in Form von geschlossenen Kurven an. Regionen basierte Ansätze sind im Allgemeinen dann zu bevorzugen, wenn die Ähnlichkeit innerhalb der Region groß ist, die Grenze zu benachbarten Regionen allerdings nicht stark ausgeprägt ist.

Entgegengesetzt dazu gehen Methoden der anderen Gruppe direkt auf Regionsgrenzen oder andere linienhafte Elemente in digitalen Bildern los und versuchen diese zu finden. *Snakes* gehören ganz klar in diese Gruppe der linienhaften Methoden. Abhängig vom verwendeten Modell können linienhafte Ansätze in Kantenextraktion oder Linienextraktion unterschieden werden. Der Unterschied zwischen Kanten und Linien wurde in Abschnitt 1.4.1 definiert. *Snakes* können gleichermaßen zur Kanten- wie auch zur Linienextraktion eingesetzt werden.

Eine eigene Gruppe ist die Bandextraktion (die Extraktion mehrerer Pixel breiter Linien), bei der die beiden Begrenzungskanten getrennt extrahiert werden können. Meist werden bei diesen Ansätzen ein linienhaftes Verfahren zur Extraktion der beiden Kanten mit einem Regionen basierten Verfahren für den Bereich zwischen den Begrenzungskanten kombiniert.

2.1 Einführung in *Snakes*

Kass et al. (1987) veröffentlichten ein Werkzeug zum automatisationsunterstützten (halbautomatischen) Messen von Kanten oder Linien in digitalen Bildern. Der Operateur hat dabei die Aufgabe, die grobe Position und den groben Verlauf der Kurve in einem digitalen Bild mittels weniger Punkte vorzugeben. Durch diese Punkte wird eine Kurve (ein Polygonzug mit dichter Punktfolge) gelegt. Dieser Kurve wird eine Energiefunktion zugeordnet, mit deren Hilfe die Qualität der aktuellen Lage und Form bestimmt wird. Wird der automatische, Optimierungsprozess der Energie gestartet, bewegt sich die Kurve im Bild und verändert ihre Form solange bis sie detailgetreu über der Zielkurve zu liegen kommt (siehe Abb. 2-2). Kass et al. gaben diesem Modell den Namen *active contour model* (Kurven, die sich aktiv verformen und verschieben) oder *snake.*

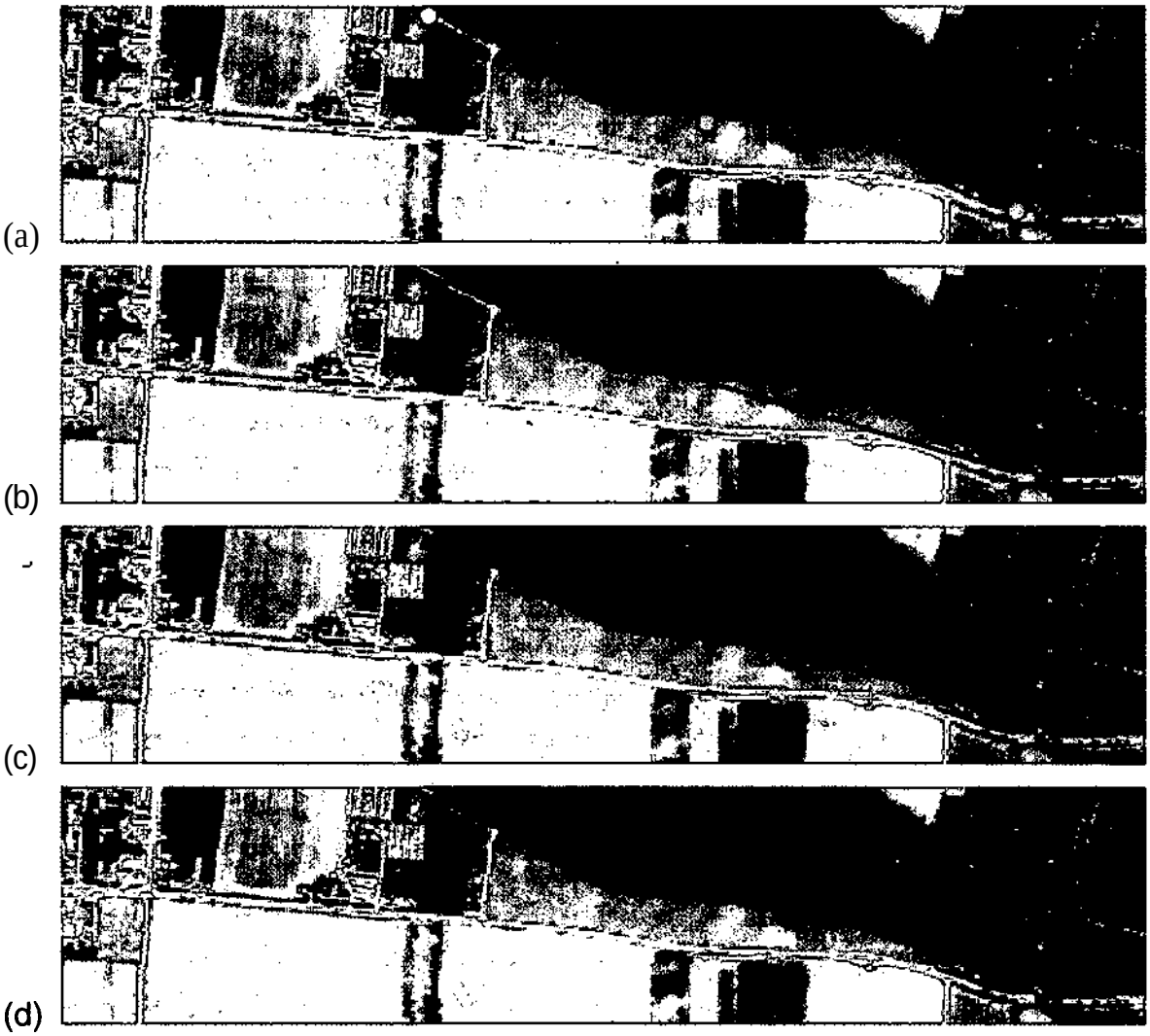


Abb. 2-2: Vorgang einer Extraktion mittels *snakes:* (a) Der Operateur digitalisiert wenige Punkte näherungsweise entlang des Waldrands. (b) Die *snake* wird zwischen diesen Punkten als Polygonzug initialisiert. (c) Etliche Iterationen der Energieminimierung später haben manche Knoten bereits das Optimum erreicht. (d) Schließlich hat die ganze *snake* ein Energieminimum erreicht.

Snakes können als univariate, ebene Spezialform von *deformable models* (McInerney und Terzopoulos, 1996; Terzopoulos und Fleischer, 1988) gesehen werden. In ihrer allgemeinen Definition handelt es sich bei *deformable models* um Objekte (Kurven oder Oberflächen), die in ihren mathematischen Grundlagen Einflüsse aus Geometrie, Physik und Approximationstheorie vereinen. Die Geometrie liefert Modelle zur Beschreibung der Form des Objekts. Die Physik beschreibt Kräfte und Zwänge, unter deren Einfluss sich die Objekte verschieben und verformen. Die Approximationstheorie bietet Verfahren an, diese Objektmodelle in gemessene Daten einzupassen. Der Einsatz von *deformable models* zählt zu so genannten *top-down (model-driven)* Methoden.

Als eine Spezialform von *deformable models* zeichnen sich *snakes* durch ihre Robustheit gegenüber Rauschen im Bild und Lücken in der Bildkante/-linie aus. Andere Namen für ähnliche oder erweiterte Ansätze sind *deformable contours* oder *balloons.*

Der folgende Abschnitt definiert die zu Grunde liegende Energiefunktion als das Schlüsselelement von *snakes.* Die Energieterme werden anhand der ursprünglichen Formulierung von Kass et al. präsentiert. Fallweise wird auch die Abgrenzung zu ähnlichen Methoden angegeben, auf die in dieser Arbeit nicht eingegangen wird. Abschnitt 2.3 fasst die Stärken der *snakes* zusammen, spricht aber auch manche ihrer Probleme an.

2.2 Definition von *Snakes*

Snakes wurden erstmals von Michael Kass, Andrew Witkin und Demetri Terzopoulos bereits 1987 präsentiert. Üblicherweise wird ein zweiter Aufsatz derselben Autoren referenziert, der ein Jahr später im International Journal of Computer Vision (Kass et al., 1988) erschienen ist. Im Folgenden wird diese Definition von *snakes* wiedergegeben.

Eine *snake* ist eine univariate, ebene Kurve im 2D, die innerhalb eines digitalen Bildes definiert ist. Die Kurve sei bezüglich *s* parameterisiert (z.B. die Bogenlänge der Kurve oder die Sehnenlänge eines approximierenden Polygonzugs):

$$\mathbf{v}(s) = (x(s), y(s))^{\mathrm{T}} \tag{2-1}$$

Der Parameterbereich sei begrenzt auf $s \in [0,1]$. Eine *snake* ist innerhalb der Ausdehnungen des ihr zugeordneten Bildes definiert, d.h. die beiden Koordinaten *x* und *y* der Knoten v sind begrenzt durch die Ränder des Bildes. Das Ziel der *snake* ist, eine bestimmte Form und Position zu finden, die das Energiefunktional E^*_{snake} minimiert, welches durch Integration über die gesamte Kurve ermittelt wird[1]:

[1] Das hochgestellte * in E^*_{snake} bezeichnet die über die gesamte Kurve integrierte Energie, während E_{snake} den Anteil eines Knotens an der Gesamtenergie angibt.

$$E^*_{snake} = \int_0^1 E_{snake}(\mathbf{v}(s))\, ds \qquad (2\text{-}2)$$

Die Gesamtenergie einer *snake* E^*_{snake} beschreibt die Qualität ihres momentanen Status. Der Status einer *snake* sei definiert durch ihre Form und ihre Lage im Bild.

Im Allgemeinen wird die Energiefunktion derart formuliert, dass sie bei optimalem Status kleine Werte liefert. Das heißt, wenn sich die *snake* dem zu extrahierenden Merkmal im Bild annähert und gleichzeitig eine glatte Form beibehält, sinkt die Energie der *snake.* Dafür sind verschiedene Terme in der Energiefunktion verantwortlich. Drei Gruppen von Energietermen werden üblicherweise verwendet: interne Terme, Bildterme (photometrische Terme) und externe Terme.[2]

$$E_{snake} = E_{int} + E_{pho} + E_{ext} \qquad (2\text{-}3)$$

2.2.1 Interne Terme

Interne Terme beschreiben die ideale Form der *snake.* Im ursprünglichen Konzept wird die ideale Form als glatt bezeichnet ohne starke Krümmungen oder gar Knicken. Diese Verwendung des Begriffes „glatt" ist ähnlich zur Definition glatter Kurven in der Mathematik (differenzierbare Kurven).

Üblicherweise wird eine Kombination verschiedener interner Terme verwendet. In Anlehnung an *splines* setzen Kass et al. Terme der ersten und zweiten Ableitung der Kurve an. Terme erster Ordnung korrespondieren zur Änderung der Bogenlänge, sollen das Dehnen der Kurve hemmen (Elastizitätskräfte) und dem Modell das Verhalten eines Gummibandes geben. Terme zweiter Ordnung korrespondieren zur Krümmung und sollen dem Modell das Verhalten eines dünnen Stabs aufzwingen:

$$E_{int} = \tfrac{1}{2}\left(\alpha(s)\left|\mathbf{v}_s(s)\right|^2 + \beta(s)\left|\mathbf{v}_{ss}(s)\right|^2\right) \qquad (2\text{-}4)$$

Mit $\alpha(s)$ und $\beta(s)$ als positive Gewichtungsfaktoren soll der relative Einfluss der beiden Komponenten gesteuert werden. Diese Gewichte können entlang der Kurve variieren. Falls beispielsweise der Einfluss des Terms zweiter Ordnung verringert wird (oder $\beta(s_K)$ gar zu Null gesetzt wird), kann die *snake* am Knoten mit Parameterwert s_K eine starke Krümmung haben (oder sogar einen Knick). In der Praxis werden die Gewichte $\alpha(s)$ und $\beta(s)$ meist konstant für die ganze Kurve gesetzt.

Andere Formulierungen der internen Energie werden in so genannten *shape-based* oder *model-based* (Form oder Modell basierten) Methoden verwendet (z.B. Lai, Chin,

---

[2] In dieser Arbeit werden konsequent die Bezeichnungen „interne Terme", „photometrische Terme" und „externe Terme" verwendet mit den im Text definierten Bedeutungen. Falls das digitale Bild nicht photographischen Ursprungs ist, werden photometrische Terme auch „Bildterme" genannt. In der Literatur werden diese Bezeichnungen leider nicht einheitlich verwendet. Verwirrenderweise werden manchmal die photometrischen Terme als externe Terme bezeichnet, obwohl im Konzept von Kass et al. externe Terme eine andere Bedeutung haben.

1995 oder Radeva et al., 1995), bei denen die Form der Zielkurve exakt (oder genähert) vorgegeben ist. In diesem Fall beschreiben die internen Energieterme Unterschiede zwischen der aktuellen und der gewünschten Form und versuchen auf diese Weise, die *snake* in die a priori gegebene Form zu bringen.

Zusätzliche interne Energieterme werden im Modell der *ribbon snakes* eingeführt um eine starke Variation der Breite des Bandes zu unterdrücken (vgl. Abschnitt 5.5.1).

2.2.2 Photometrische Terme

Photometrische Terme (Bildterme) formulieren die ideale Position der *snake* im Bild. Sie sind dafür verantwortlich, die *snake* zur Zielkurve zu ziehen. Die photometrische Energie kann als skalare potenzielle Energie an allen Pixeln des Bildes gesehen werden. Sie hängt nur von der Bildfunktion *I(x,y)* ab und kann daher für das ganze Bild im Voraus berechnet werden. Das so entstandene Bild mit den photometrischen Energiewerten an allen Pixeln wird manchmal als „Energiebild" bezeichnet.

Die photometrische Energie kann für verschiedene Aufgaben auf unterschiedliche Weise von der Bildfunktion abgeleitet werden. Im ursprünglichen Konzept werden zwei Möglichkeiten angegeben. Für Kanten schlagen Kass et al. das negative Quadrat der Magnitude des Gradienten vor (γ ist ein positiver Gewichtsfaktor):

$$E_{pho}^{edge} = -\gamma(s)\left|\nabla I(x,y)\right|^2 \tag{2-5}$$

Diese Formel liefert niedrige Energiewerte an Stellen wo der Gradient der Bildfunktion hoch ist, insbesondere an Bildkanten. Zur Extraktion von dunklen und hellen Linien empfehlen Kass et al. die Bildfunktion direkt zu verwenden:

$$\begin{aligned} E_{pho}^{dark\ line} &= \gamma(s)I(x,y) \\ E_{pho}^{bright\ line} &= -\gamma(s)I(x,y) \end{aligned} \tag{2-6}$$

Die zu Grunde liegende Bildfunktion $I(x,y)$ ist bei Graubildern der Grauwert selbst. Bei Farbbildern kann der Intensitätswert der Farbe verwendet werden. Ein komplexerer Energieterm für Farbbilder wird in Ngoi, Jia (1996) vorgestellt.

2.2.3 Externe Terme

Während interne Terme und photometrische Terme essenzielle Grundbausteine der Energiefunktion von *snakes* sind, müssen externe Terme nicht unbedingt vorhanden sein. Sie erlauben es globale Kräfte und Zwänge auf *snakes* auszuüben um deren Entwicklung zu steuern. Beispielsweise kann der Operateur in einem interaktiven System die Entwicklung der *snake* beobachten und im Notfall Punkte festlegen, von denen die *snake* angezogen werden soll, bzw. andere, von denen die *snake* ferngehalten werden soll. Diese Punkte wurden von Kass et al. als *springs* bzw. *volcanoes* bezeichnet.

2.2.4 Sonstige Terme

Neben den ursprünglichen drei Gruppen an Energietermen soll noch kurz eine weitere Gruppe angesprochen werden, die in (2-3) nicht angeführt ist. Für die Verfolgung von Objektkonturen in einer Serie von Bildern werden zweckmäßigerweise zusätzliche Geschwindigkeitsterme verwendet, die dem Modell das Verhalten einer Kaiman-Filterung geben (Peterfreund, 1999; Terzopoulos, Szeliski, 1992). Dafür wird oft die Bezeichnung Kaiman *snakes* verwendet.

2.3 Stärken und Schwächen von *Snakes*

Stärken:

- In ihrer ursprünglichen Konzeption als semi-automatisches Messwerkzeug erlauben *snakes* ausgehend von einer Näherung (grobe Form und Lage der Kurve mittels weniger Punkte) linienhafte Elemente detailliert zu extrahieren.
- In einem automationsgestützten Auswerteprozess kann der Operateur die Entwicklung der *snake* beeinflussen, indem er auf einfache Weise externe Kräfte einführt.
- Ebenso kann er durch Anpassen der Gewichte der einzelnen Energieterme die Entwicklung der *snake* beeinflussen.
- Dank der internen Energieterme können *snakes* Lücken in den zu extrahierenden Kurven überbrücken. Dies unterscheidet sie von anderen Merkmalsextraktionsverfahren (z.B. Canny, 1986).
- Durch die Kombination verschiedenartiger Energieterme erlangen *snakes* eine Robustheit gegenüber Rauschen und vereinzelten Störungen im Bild.
- Je nach Anwendung können zusätzliche Energieterme verwendet werden. Die meisten Algorithmen zur Energieoptimierung schränken die Möglichkeiten für die mathematische Formulierung der Energie kaum ein. Daher können für spezifische Anwendungen angepasste Energieterme eingesetzt werden.

Den Vorteilen stehen aber doch einige Nachteile bezüglich der Konvergenz gegenüber, die je nach Einsatzzweck der *snakes* gravierend sein können.

- Das Energiefunktional einer *snake* ist hochgradig nicht-konvex. Das bedeutet, dass neben dem globalen Energieminimum viele lokale Minima existieren, von denen die *snake* gefangen werden kann. Um dies zu vermeiden muss der Ausgangsstatus der *snake* sehr nahe an der Zielkurve liegen (z.B. Klemencic et al., 1995; Lai, Chin, 1995; Xu, Prince, 1998 oder Trinder et al., 2000).
- Der Einfluss der photometrischen Energie sinkt rasch, je weiter die *snake* von der Zielkurve entfernt ist. Daraus ergibt sich ebenfalls die Forderung, dass der Ausgangsstatus der *snake* sehr nahe an der Zielkurve liegen muss um Konvergenz zu gewährleisten (z.B. Cohen, Cohen, 1993).

- Die interne Energie verhindert, dass *snakes* Formen mit starken Krümmungen annehmen. *Snakes* sind daher nicht geeignet um schlauch- oder röhrenförmige Strukturen zu detektieren (Mclnerney und Terzopoulos, 1996).

- Die interne Energie verhindert ebenso, dass starke Einbuchtungen richtig extrahiert werden, insbesondere wenn photometrische Kräfte nicht stark genug sind (Mclnerney, Terzopoulos, 1996; Xu, Prince, 1998).

- Selbst bei starken photometrischen Kräften kann es passieren, dass die *snake* an Stellen starker Krümmung das globale Minimum überspringt (Cohen, Cohen, 1993; Xu et al., 1994).

- Bei nebeneinander liegenden Kanten können *snakes* nicht alle Kanten getrennt extrahieren. Vielfach kann nur die am stärksten ausgeprägte Kante gefunden werden. Im schlechtesten Fall springt die extrahierte Kurve zwischen den Kanten hin und her (Kerschner, 1998; Trinder et al., 2000).

- *Snakes* neigen zum Schrumpfen, da das globale Minimum für Kurven, die ausschließlich internen Energiekräften wie in (2-4) unterliegen, dann erreicht ist, wenn die Kurve zu einem Punkt degeneriert ist. Die interne Energie ist für diesen Fall gleich Null (z.B. Amini et al, 1988; Cohen und Cohen, 1993 oder Hoch und Litwinowicz, 1996).

- Der herkömmliche *snake*-Ansatz bietet keine Sicherheit, dass die *snake* im Laufe ihrer Entwicklung topologische Eigenschaften ihrer Ausgangslage beibehält, insbesondere Kurveneigenschaften des Jordan-Kurventheorems. Danach ist eine Kurve nur dann die Begrenzung einer einzelnen Region, wenn sie geschlossen ist, sich nicht selbst schneidet und sich auch keine Segmente überdecken (Grzeszczuk, Levin, 1997). Während geschlossene *snakes* während ihrer Entwicklung geschlossen bleiben, können die beiden anderen Eigenschaften nicht gewährleistet werden.

- Die Wahl der Gewichte für verschiedene Energieterme bereitet Schwierigkeiten, da die Energiewerte unterschiedliche Wertebereiche haben und sich diese Werte bei Annäherung an das Minimum unterschiedlich schnell verringern (Li, 1997).

Die Schwächen waren meist Anlass zur Verbesserung der Methode. Die folgenden Kapitel widmen sich publizierten Ansätzen, stellen aber auch im Rahmen der Dissertation entwickelte Anpassungen vor. Es gibt mehrere Ansatzpunkte, an denen Verbesserungen durchgeführt werden können:

- Manche Probleme kann man durch eine andere Methode der Energieoptimierung in den Griff bekommen. Diese Methoden werden im folgenden Kapitel 3 beschrieben.

- Andere Probleme können gelöst werden, indem Energieterme ausgetauscht werden oder zusätzliche Energieterme in die Energiefunktion aufgenommen werden. Damit befasst sich Kapitel 4.

- Es bleiben dann noch Probleme, die erst durch eine bestimmte Strategie der Anwendung von *snakes* vermieden oder umgangen werden können. Derartige Strategien werden in Kapitel 5 vorgestellt.

Viele der Verbesserungen bringen andere Nachteile mit sich oder sind nur für eine eingeschränkte Menge an Kurven einsetzbar. Beispielsweise wird oft davon ausgegangen, dass die *snake* eine geschlossene Kurve ist. Gerade für Anwendungen im Bereich der digitalen Photogrammetrie oder der Topographie muss das *snake*-Modell für Kurven mit zwei freien Enden gültig sein.

3 Energieminimierung

Wie wir im vergangenen Abschnitt gesehen haben, ist die gesamte Energie von *snakes* ein komplexes Funktional, das analytisch nicht gelöst werden kann:

$$E^*_{snake} = \int_0^1 \tfrac{1}{2}\left(\alpha(s)|\mathbf{v}_s(s)|^2 + \beta(s)|\mathbf{v}_{ss}(s)|^2\right) + E_{pho}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s))\, ds \qquad (3\text{-D}$$

Wir stehen vor dem Optimierungsproblem eine Kurve mit minimaler Gesamtenergie zu finden. Optimierung bedeutet, die beste Lösung in einer Vielzahl von möglichen Alternativen zu finden. Optimierungsaufgaben findet man in verschiedenen Anwendungen der Wissenschaft, Industrie, Wirtschaft und anderen Bereichen. Neben der *exhaustive search* Methode (erschöpfendes Durchsuchen aller Varianten) wurden viele effizientere Methoden entwickelt (Optimization Online, 2003).

Das konkrete Optimierungsproblem stellt auf Grund des nicht-konvexen Energiefunktionals ein so genanntes *ill-posed* Problem dar. Daraus folgt, dass es neben einem globalen Minimum auch mehrere lokale Minima gibt.

Obwohl man das Optimierungsproblem intuitiv als ein statisches betrachten würde, wird die Aufgabe generell in ein dynamisches Problem übergeführt. Ausgehend von einer Schätzung des Status (Form und Position) verkleinert die *snake* schrittweise ihre Energie durch Veränderung des Status. Durch diesen iterativen Prozess ergibt sich eine fortlaufende Veränderung der Form und Lage im Bild. Die Kurve zeigt Bewegungen und Deformationen, die an Schlangen erinnern und die dem Modell den Namen gaben.

Da Optimierungsaufgaben in vielen (nicht nur technischen) Anwendungen zu lösen sind, gibt es entsprechend viele verschiedene Optimierungsmethoden. Eine Vielzahl dieser numerischen Lösungsansätze wurde auch für das Optimierungsproblem der *snakes* verwendet. Das ursprüngliche Konzept von Kass et al. (1988) wendet Variationsrechnung zur Optimierung an. Dieses Konzept wird im folgenden Abschnitt erklärt. In den Abschnitten 3.2 bis 3.5 werden alternative Lösungen vorgestellt: die Lösung mittels dynamischer Programmierung, ein Ansatz über kleinste-Quadrate-Ausgleichung, die so genannte *level-set*-Methode sowie die Lösung über *simulated annealing.* Für manche dieser alternativen Methoden muss das Optimierungsproblem etwas umformuliert werden. Das Kapitel schließt mit einem Vergleich dieser Optimierungsmethoden nach verschiedenen Kriterien.

3.1 Lösung über Variationsrechnung

Das Problem der Energieminimierung, wie in (3-1) definiert, ist ein Optimierungsproblem der Art: Bestimme eine Kurve $\mathbf{v}(s)$, sodass:

$$\int_a^b F(s, \mathrm{v}, \mathbf{v}_s, \mathbf{v}_{ss})ds \rightarrow \min \tag{3-2}$$

Es wurde bewiesen, dass die Kurve, die diese Optimierungsaufgabe löst, die folgende Euler-Lagrange'sche Differenzialgleichung erfüllen muss (Mathworld, 2003):

$$F_{\mathrm{v}} - \frac{\partial}{\partial s} F_{\mathrm{v}_s} + \frac{\partial^2}{\partial s^2} F_{\mathrm{v}_{ss}} = 0 \tag{3-3}$$

Setzt man die Gewichte der internen Energie in (3-1) konstant, also $\alpha(s) = \alpha$ und $\beta(s) = \beta$, so ergibt sich die Euler Gleichung zu:

$$\frac{\partial(E_{pho} + E_{ext})}{\partial \mathbf{v}} - \alpha\, \mathbf{v}_{ss} + \beta\, \mathbf{v}_{ssss} = 0 \tag{3-4}$$

Eine numerische Lösung dieser Differenzialgleichung kann nach Diskretisierung gefunden werden. Dazu wird die kontinuierliche Kurve $\mathrm{v}(s)$ durch eine Serie von Knoten (Polygonzug) ersetzt:

$$\mathbf{v}_i = (x_i, y_i)^T \tag{3-5}$$

Die Serie der diskreten Knoten wird optimiert. Um das Ergebnis der Optimierung wieder als kontinuierliche Kurve zu erhalten, kann beispielsweise ein *spline* in die verbesserten Positionen der Knoten interpoliert werden (vgl. Kraus, 2000, Kapitel H.2).

Die Euler Gleichung (3-4) kann nun für alle Knoten angeschrieben werden, indem die partiellen Ableitungen durch finite Differenzen ersetzt werden. Im diskretisierten Gleichungssystem werden wieder individuelle Gewichte α_i und β verwendet:

$$\begin{aligned} &\alpha_{i-1}(\mathbf{v}_i - \mathbf{v}_{i-1}) - \alpha_i(\mathbf{v}_{i+1} - \mathbf{v}_i) + \\ &\beta_{i-1}(\mathbf{v}_{i-2} - 2\mathbf{v}_{i-1} + \mathbf{v}_i) - 2\beta_i(\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}) + \beta_{i+1}(\mathbf{v}_i - 2\mathbf{v}_{i+1} + \mathbf{v}_{i+2}) + f_{\mathrm{v}}(\mathbf{v}_i) = 0 \end{aligned} \tag{3-6}$$

Dabei werden in $f(\mathbf{v}_i)$ alle Energieterme zusammengefasst, die nur von der Position eines einzelnen Knotens abhängig sind, also $f(\mathbf{v}_i) = E_{pho}(\mathbf{v}_i) + E_{ext}(\mathbf{v}_i)$.

Für die ersten und letzten beiden Knoten eines offenen Polygonzugs können manche finite Differenzen nicht berechnet werden. Diese werden zu Null gesetzt. Ist das Polygon geschlossen, wird die Serie der Knoten zur Berechnung der Differenzen periodisch fortgesetzt.

Das Gleichungssystem von Euler-Gleichungen vom Typ (3-6) für alle Knoten kann in Matrizenform geschrieben werden:

$$\mathrm{A}\mathrm{v} + \mathbf{f}_{\mathrm{v}} = 0 \tag{3-7}$$

Die Matrix A ist eine symmetrische, pentadiagonale Bandmatrix (lediglich bei geschlossenen Kurven treten in der rechten oberen und der linken unteren Ecke je drei Werte ungleich Null auf). Das Gleichungssystem kann iterativ gelöst werden, indem

man den Vektor der Knoten v als veränderlich mit der Zeit *t* ansetzt, also *v(t).* Die Änderung der Knoten mit der Zeit, also die partielle Ableitung von v nach *t,* multipliziert mit einer negativen Schrittweite γ, wird gleichgesetzt mit der linken Seite von (3-7):

$$-\gamma\frac{\partial \mathbf{v}}{\partial t} = \mathbf{A}\mathbf{v} + \mathbf{f}_{\mathbf{v}} \tag{3-8}$$

Wenn sich die Lösung \(t) stabilisiert und das Gleichgewicht erreicht ist (d.h. die Knoten ändern sich nicht mehr), verschwindet die Ableitung des Knotenvektors nach der Zeit. Es ergibt sich damit eine Lösung von Gleichung (3-7). Dieser Ansatz, die Ableitung verschwinden zu lassen um ein lokales Minimum zu finden, entspricht einem *gradient descent* Algorithmus.

Um iterativ zu einer Lösung zu kommen muss der Optimierungsprozess auch in der Zeit **diskretisiert** werden, also:

$$\frac{\partial \mathbf{v}}{\partial t} \approx (\mathbf{v}_t - \mathbf{v}_{t-1}) \tag{3-9}$$

Damit ergibt sich:

$$-\gamma(\mathbf{v}_t - \mathbf{v}_{t-1}) = \mathbf{A}\mathbf{v}_t + \mathbf{f}_{\mathbf{v}}(\mathbf{v}_t) \tag{3-10}$$

Die Unbekannte v zum Zeitpunkt *t* kommt in dieser Gleichung an drei Stellen vor. Man spricht von einer vollständig impliziten Gleichung. Sie ist wegen der komplizierten Form von $\mathbf{f}_{\mathbf{v}}$ schwierig zu lösen. Um schnellere Konvergenz zu erreichen wird vereinfachend angenommen, dass sich die Ableitungen der photometrischen und externen **Terme** innerhalb eines Zeitschritts nicht ändern, d.h. $\mathbf{f}_{\mathbf{v}}(\mathbf{v}_t) \approx \mathbf{f}_{\mathbf{v}}(\mathbf{v}_{t-1})$. Damit ergibt sich:

$$-\gamma(\mathbf{v}_t - \mathbf{v}_{t-1}) = \mathbf{A}\mathbf{v}_t + \mathbf{f}_{\mathbf{v}}(\mathbf{v}_{t-1}) \tag{3-11}$$

Es handelt sich nun um eine halbimplizite Gleichung (implizit bezüglich der Matrix A, aber explizit bezüglich $\mathbf{f}_{\mathbf{v}}$). Löst man diese Gleichung nach $\mathbf{v}_t$ auf, erhält man schließlich folgende Gleichung für die iterative **Lösung:**[3]

$$\mathbf{v}_t = (\mathbf{A} + \gamma\mathbf{I})^{-1}(\gamma\mathbf{v}_{t-1} - \mathbf{f}_{\mathbf{v}}(\mathbf{v}_{t-1})) \tag{3-12}$$

Dabei ist I die Einheitsmatrix. Der Status der *snake* zum Zeitpunkt *t* lässt sich somit aus dem Status $\mathbf{v}_{t-1}$ ableiten. Der Anfangsstatus vo muss vorgegeben werden. Zur Inversion der Matrix können auf Grund ihrer bandförmigen Struktur effiziente Algorithmen genutzt werden.

---

[3] In Kass et **al. (1987)** fehlt in Formel (19) bzw. (20) das γ **vor** $\mathbf{x}_{t-1}$ bzw. y,.,. Dies ist in Kass et **al.** (1988) korrigiert.

Die Lösung mittels Variationsrechnung hat Stärken und Schwächen verglichen mit anderen Optimierungsmethoden:

- Ein Vorteil liegt in der Effizienz bezüglich Rechenzeit. Die Matrizeninversion muss im Allgemeinen nur einmal durchgeführt werden. Daher ist nur die erste Iteration zeitaufwändig, nachfolgende Iterationen benötigen kaum Rechenzeit. Weitere Inversionen sind nur dann nötig, wenn die Gewichtsfaktoren geändert werden.
- Der größte Nachteil liegt darin, dass vierte Ableitungen der Kurve in die Euler-Gleichungen eingehen. Auf Grund der Diskretisierung der Kurve werden die Ableitungen durch finite Differenzen abgeschätzt. Je höher der Grad der Ableitung, desto stärker steigt der Fehler in der Abschätzung (vgl. Anhang A). Um halbwegs zuverlässige Abschätzungen zu erhalten, muss das Diskretisierungsintervall klein sein, was zu vielen Knoten und daher großen Gleichungssystemen führt.
- Neben der Diskretisierung im Raum wurde auch eine Diskretisierung in der Zeit eingeführt. Dies führt zu Sprüngen der *snake* von einem zum nächsten Status. Gerade dort, wo die Änderung der Energie hoch ist, machen *snakes* auch große Schritte. Nach Cohen, Cohen (1993) birgt dies die Gefahr, dass die *snake* das Minimum überspringt und nicht mehr zu ihm zurückkehrt. Dies kann nur durch ein kleines Zeitintervall verhindert werden, welches über 7 gesteuert werden kann. Kleine Schritte erhöhen natürlich wiederum die Anzahl der Iterationen.

Cohen, Cohen (1993) schlagen vor, (3-11) auch bezüglich der Matrix A explizit zu machen, d.h. $\mathbf{A}\mathbf{v}_t$ durch $\mathbf{A}\mathbf{v}_{t-1}$ zu approximieren. Durch diese Näherung wird der Algorithmus noch schneller, denn man erspart sich jegliche Matrizeninversion. Es ergibt sich die explizite Lösung:[4]

$$\mathbf{v}_t = -\frac{1}{\gamma}\left((\mathbf{A}+\gamma\mathbf{I})\mathbf{v}_{t-1} + \mathbf{f}_v(\mathbf{v}_{t-1})\right) \tag{3-13}$$

Eine alternative Lösungsmethode über Variationsrechnung präsentieren ebenfalls Cohen, Cohen (1993). Sie verwenden eine *finite elements* Methode statt finiter Differenzen.

3.2 Lösung über dynamische Programmierung

Die Lösung des Energieoptimierungsproblems mittels dynamischer Programmierung wurde von Amini et al. (1988) eingeführt. Dynamische Programmierung kann zur Lösung verschiedenster Optimierungsaufgaben verwendet werden. Dabei wird eine Strategie der sequentiellen Entscheidungsfindung angewendet.

---

[4] In Cohen, Cohen (1993) wird $\tau = 1 / \gamma$ sowie $F = -\mathbf{f}_v$ verwendet.

Ein Optimierungsproblem kann nur dann mit dynamischer Programmierung gelöst werden, wenn es eine endliche Menge an möglichen Lösungen gibt. Daher muss auch hier das im kontinuierlichen Raum definierte Problem (3-1) durch eine Formulierung im diskreten Raum ersetzt werden. Dazu wird die kontinuierliche Kurve $\mathbf{v}(s)$ wieder durch eine Serie von Knoten $\mathbf{v}_i$ approximiert. Auch der zweidimensionale Raum, in dem sich der einzelne Knoten bewegen darf, muss diskretisiert werden. Das Intervall wird dabei zweckmäßigerweise der Pixelgröße des zugehörigen Energiebildes gleichgesetzt. Das Integral zur Berechnung der Energie in (3-1) wird nun ersetzt durch die Summe der Energiewerte an allen Knoten:

$$E^*_{snake} = \sum_{i=0}^{n-1} \tfrac{1}{2}\left(\alpha_i \left|\mathbf{v}_i - \mathbf{v}_{i-1}\right|^2 + \beta_i \left|\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}\right|^2\right) + E_{pho}(\mathbf{v}_i) + E_{ext}(\mathbf{v}_i) \qquad (3\text{-}14)$$

Wieder werden finite Differenzen verwendet um Ableitungen zu schätzen. Im Vergleich zur Lösung über Variationsrechnung werden hier jedoch nur Differenzen der ersten und zweiten Ordnung benötigt. Das Diskretisierungsintervall der Kurve (der Abstand der Knoten voneinander) beträgt im Allgemeinen wenige Pixel.

Dynamische Programmierung lässt sich dann einsetzen, wenn nicht alle Variablen eines komplexen Optimierungsproblems simultan voneinander abhängen. Die Idee dahinter ist:

- Wenn man nicht weiß, wie man ein komplexes Problem optimiert, kann man aber vielleicht unabhängige Teilprobleme lösen.
- Durch die Kombination der Lösungen aller Teilprobleme kommt man auf die Lösung des Gesamtproblems.
- Wenn man nicht weiß, welche der Teilprobleme gelöst werden müssen um die Lösung des Gesamtproblems zu finden, löst man alle, die möglicherweise Teil der Gesamtlösung sind.
- Die Effizienz der Lösung über dynamische Programmierung hängt davon ab, ob einzelne Teillösungen von vornherein ausgeschlossen werden können.

Daher müssen wir zunächst das Gesamtproblem in unabhängige Teilprobleme zerlegen. Dies ist in unserem Fall einfach. Die Optimierung jedes einzelnen Knotens stellt ein Teilproblem dar. Der Energieanteil eines Knotens $\mathbf{v}_i$ zur Gesamtenergie der *snake* lautet:

$$E_{snake} = \tfrac{1}{2}\left(\alpha_i \left|\mathbf{v}_i - \mathbf{v}_{i-1}\right|^2 + \beta_i \left|\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}\right|^2\right) + E_{pho}(\mathbf{v}_i) + E_{ext}(\mathbf{v}_i) \qquad (3\text{-}15)$$

Man erkennt, dass das Teilproblem der Optimierung eines Knotens nur von der Position dieses Knotens und den Positionen seines Vorgängers und Nachfolgers abhängt. Dadurch sind die Voraussetzungen zur effizienten Lösung über dynamische Programmierung gegeben.

Im speziellen Fall wird ein so genannter *time-delayed multistage decision process* (zeitverzögerter Multistufen-Entscheidungsprozess) angewandt:

- *time-delayed:* Die Entscheidung, welche Teillösung benötigt wird, wird auf später verschoben.
- *multistage:* Jeder Knoten muss optimiert werden und stellt eine eigene Stufe dar, die zu lösen ist.

In jeder Iteration hat jeder einzelne Knoten einen beschränkten Bereich im diskreten zweidimensionalen Raum, in dem er sich bewegen darf. Dieser diskrete 2D-Raum entspricht dem digitalen Bild mit seinem Pixelraster. Im Beispiel von Abb. 3-1 darf jeder Knoten entweder an seinem Platz (auf seinem Pixel) bleiben oder in eine von acht vordefinierten Richtungen wandern. Daher gibt es neun mögliche neue Positionen dieses Knotens.

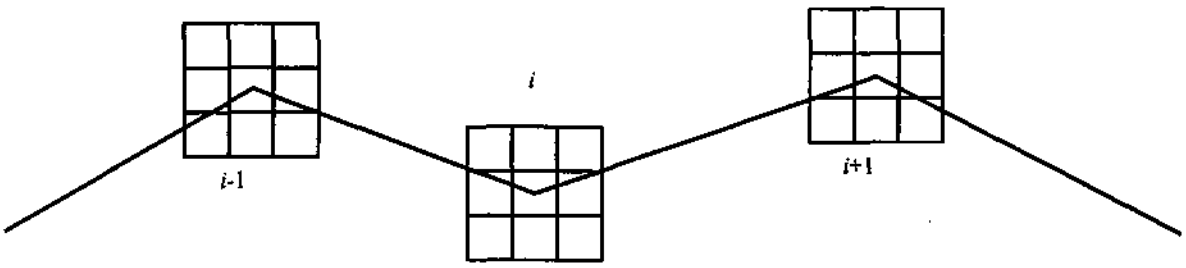


Abb. 3-1: Jeder Knoten der *snake* darf auf seiner Position bleiben oder eine neue Position in seiner Nachbarschaft einnehmen (in diesem Beispiel eine 8-er Nachbarschaft).

Der eigentliche (hinausgezögerte) Entscheidungsprozess ist im speziellen Fall ein so genannter *backward decision process.* Zuerst löst man in einem *forward decision process* viele Teilprobleme.

<u>*Forward decision process:*</u>

Für alle Knoten $\mathbf{v}_i$ stellt sich folgende Aufgabe:

> Angenommen, man weiß, wo die beiden Knoten $\mathbf{v}_i$ und $\mathbf{v}_{i+1}$ liegen, finde für den Knoten $\mathbf{v}_{i-1}$ jene Position, wo der Energieanteil des Knoten $\mathbf{v}_i$ nach (3-15) minimal ist.

Nun weiß man aber noch nicht, wo nach der Optimierung die Knoten $\mathbf{v}_i$ und $\mathbf{v}_{i+1}$ zu liegen kommen werden. Daher muss man für alle potenziellen Positionen dieser beiden Knoten das Teilproblem lösen. Diese Teillösungen werden in Matrizen gespeichert, damit sie später bei der Kombination zur Gesamtlösung verwendet werden können.

Die Struktur dieser Matrizen soll anhand eines Beispiels erklärt werden. Dazu werden zunächst die erlaubten Bewegungen eines Knotens nummeriert: ausgehend von der linken oberen Rasterzelle bei Null beginnend wird zuerst nach rechts und danach nach unten weitergezählt. Das Teilproblem soll noch einmal formuliert werden:

Gegeben sei der Knoten $\mathbf{v}_i$ an der Position k und der Knoten $\mathbf{v}_{i+1}$ an der Position /. Finde das Minimum des Energienanteils $E_i^{min}(k,l)$ für alle möglichen Positionen j des Knotens $\mathbf{v}_{i-1}$.

Im fiktiven Beispiel von Abb. 3-2 sind für einen Abschnitt des Polygons (durchgezogene Linie) die möglichen neuen Positionen der drei Knoten dargestellt. Das erste Teilproblem lautet: k und / seien beide 0 (punktierte Linie); variiere den Knoten $\mathbf{v}_{i-1}$ und finde das Minimum. Der minimale Energieanteil von $E_i^{min}(0,0)$ beträgt beispielsweise 1.23 und wird z.B. dann erreicht, wenn sich der Knoten M auf Position $j_i^{min}(0,0) = 7$ befindet (dieses j gilt für den Knoten i, denn j bezeichnet die Position des jeweiligen Vorgängerknotens). Zur Ermittlung dieser Position müssen neun Varianten geprüft werden.

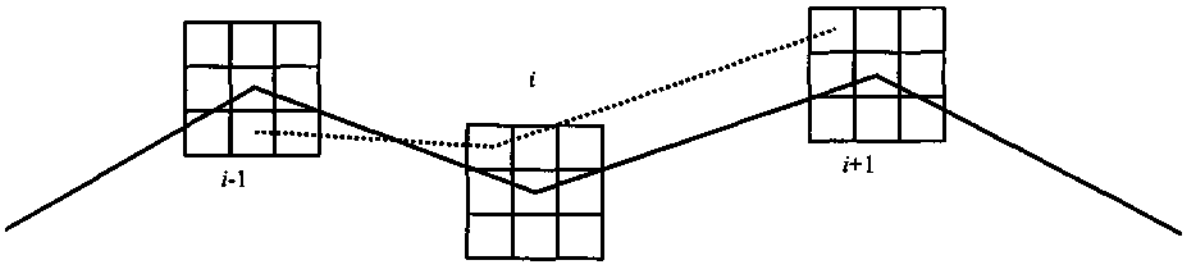


Abb. 3-2: Die punktierte Linie zeigt ein fiktives Beispiel zweier Polygonsegmente, die den Energieanteil des Knotens $\mathbf{v}_i$ für k - 0 und / = 0 minimieren. Der Knoten $\mathbf{v}_{i-1}$ wandert dazu auf die Position mit dem Index 7. Der zugehörige Energieanteil betrage 1.23.

Die Lösungen der Teilprobleme werden in der Positionsmatrix $\mathbf{P}(i,k,l)$ gespeichert. In der Matrix $\mathbf{E}(i,k,l)$ sind die akkumulierten Energieanteile gespeichert. In unserem Beispiel wird $\mathbf{P}(i,0,0)$ auf 7 gesetzt, $\mathbf{E}(i,0,0)$ erhält den Energieanteil des Teilproblems plus die akkumulierten Energieanteile vom Beginn der *snake* bis zu Knoten $\mathbf{v}_{i-1}$: 1.23 + $\mathbf{E}(i-1,7,0)$ (sprich: die akkumulierten Energieanteile bis zum Knoten $\mathbf{v}_{i-1}$, wenn er auf Position 7 liegt und sein Nachfolger auf Position 0).

Die Matrizen sind demnach folgendermaßen zu lesen: Falls der Knoten $\mathbf{v}_i$ auf Position k liegt und der Knoten $\mathbf{v}_{i+1}$ auf Position /, dann weiß man, dass der Knoten $\mathbf{v}_{i-1}$ auf Position $\mathbf{P}(i,k,l)$ liegen muss, damit die Teil-*snake* bis Knoten i die minimale Energie mit dem Wert $\mathbf{E}(i,k,l)$ hat.

Für jeden Knoten wird dieses Teilproblem für alle potenziellen Positionen k des Knotens $\mathbf{v}_i$ und für alle potenziellen Positionen / des Knotens $\mathbf{v}_{i+1}$ wiederholt. Das sind 9^2 = 81 Teilprobleme, in denen je 9 Varianten geprüft werden müssen. Es ergeben sich 9^3 = 729 Kombinationen von drei aufeinander folgenden Knoten, die geprüft werden, von denen 81 verspeichert werden und in die nächste Stufe *(stage)* beim nächsten Knoten weiterkommen.

Die Lösung der 729 Teilprobleme wird für alle Knoten wiederholt, bis man beim vorletzten Knoten angelangt ist (d.h. für einen Polygonzug mit n Knoten: für i = 0 bis einschließlich n-3). Schließlich wird in der letzten Stufe der vorletzte Knoten $\mathbf{v}_{n-2}$ und der letzte Knoten $\mathbf{v}_{n-1}$ variiert.

Backward decision process:

Nun kann die minimale Energie auf einfache Weise als Minimum in der Matrix $E(i,k,l)$ für $i = n-2$, $k \in [0..9]$ und $l \in [0..9]$ gefunden werden. Da alle Energiewerte akkumuliert wurden, ist dieser Wert bereits die Gesamtenergie der *snake.*

Mit den hier ermittelten optimalen Positionen k und l des vorletzten und letzten Knotens kann in der Matrix $\mathbf{P}(i,k,l)$ die optimale Position j des Knoten M ermittelt werden. Setzt man anschließend $i := i-1$, $l := k$, $k := J$, findet man die optimale Position des Knotens davor. Setzt man dieses Verfahren (das so genannte *back-tracing* der P-Matrix) weiter fort, erhält man die optimale Lage und Form der gesamten *snake.*

Damit wurde der Status mit minimaler Energie aus einer endlichen Menge an möglichen Formen und Positionen ausgewählt. Diese Menge ist begrenzt durch die Freiheitsgrade jedes einzelnen Knotens. Andere Implementierungen könnten jeden Knoten in einer 24-er Nachbarschaft statt der 8-er Nachbarschaft wandern lassen oder jeden Knoten nur entlang des Normalvektors auf die Kurve im betreffenden Knoten etc.

Die Lösung muss iterativ erfolgen, da das globale Minimum meist noch nicht in der Nachbarschaft der momentanen *snake* enthalten ist. Erst wenn sich die *snake* nach etlichen Iterationen dem Minimum angenähert hat, kann dieses auch wirklich erreicht werden. In jeder Iteration wird der beschriebene zweiteilige Vorgang wiederholt. Der iterative Optimierungsprozess wird beendet, wenn sich der Status der *snake* nicht mehr ändert.

Auch der Lösungsansatz über dynamische Programmierung hat Stärken und Schwächen:

- Es wird eine effiziente Methode zum Auswählen des besten *snake*-Status aus einer endlichen Menge an möglichen Lösungen programmiert. Das Optimierungsproblem der Komplexität $O(m^n)$ (mit m der Anzahl an möglichen neuen Positionen eines Knotens und n der Anzahl der Knoten) wird zerlegt in n Teilprobleme der Komplexität $O(m^3)$. In Summe beträgt die Komplexität demnach $O(nm^3)$.
- Es wird vielfach behauptet, die Methode liefere das globale Optimum. Diese Behauptung muss eingeschränkt werden. Die Methode garantiert, in jeder Iteration das globale Minimum unter den untersuchten Lösungen zu finden. Allerdings wird die Menge der untersuchten Lösungen durch die beschränkte Bewegungsfreiheit jedes Knotens stark eingeengt. Das globale Minimum ist erst bei der letzten Iteration Teil der Menge der möglichen Lösungen. Da die *snake* in jeder Iteration von einem lokalen Minimum gefangen werden kann, von dem sie sich nicht mehr weg bewegen kann, kann so wie bei den anderen Optimierungsmethoden nicht garantiert werden das globale Minimum zu finden.
- Eine Stärke dieser Methode liegt darin, dass zusätzlich zu den *soft constraints* (alle Kriterien, die in der Energiefunktion formuliert sind und die gemeinsam

optimiert werden) auch so genannte *hard constraints* leicht berücksichtigt werden können. Damit sind Bedingungen gemeint, die nicht verletzt werden dürfen. So kann beispielsweise eine minimale Distanz zwischen zwei aufeinander folgenden Knoten eingeführt werden, damit Knoten nicht zusammenfallen (vgl. Abschnitt 4.1.4). Ein anderer Ansatz zwingt Knoten, die Kantenpixel einer Kantenextraktion (z.B. eine Extraktion mittels Canny, 1986) erreicht haben, nur noch auf solchen Kantenpixeln zu bleiben. Man kann sich weitere Beispiele nützlicher *hard constraints* überlegen.

Williams und Shah (1992) vereinfachen die Methode der dynamischen Programmierung. Sie reduzieren die Komplexität des Optimierungsproblems von $O(nm^3)$ auf $O(nm)$ pro Iteration. Ihr Algorithmus garantiert allerdings nicht mehr, das jeweilige Minimum in jeder Iteration zu finden.

Park und Keller (2001) reduzieren die Bewegungsfreiheit der *snake*-Knoten auf jene Positionen, die vorweg als Pixel auf einer Wasserscheide klassifiziert worden sind. Damit erreichen sie eine wesentliche Beschleunigung des Optimierungsprozesses. Die Wasserscheiden werden in der Potenzialfunktion der photometrischen Energie gesucht. Die Pixel der Zielkurve sind im Allgemeinen als Wasserscheidenpixel klassifiziert. Daneben zählen noch viele weitere Pixel zu den Wasserscheiden. Dennoch bringt dieser Ansatz eine Beschleunigung. Allerdings vermindert er die Robustheit gegenüber Lücken in der Zielkurve.

3.3 Lösung über Ausgleichungsrechnung

Haihong Li passte in seiner Dissertation (1997) das Konzept der *B-snakes* (Menet et al., 1990) für die Lösung über Kleinste-Quadrate-Ausgleichung (*least squares)* an. Dabei wird die interne Energie über *B-splines* gesteuert. Photometrische Energie wird über *least squares template matching* eines einfachen Kantenmusters integriert. Weiters werden zusätzliche Beobachtungen in das Ausgleichungsmodell aufgenommen, analog zu den externen Energietermen der *snake.*

Abb. 3-3: *B-spline*-Kurve mit Kontrollpolygon

Die Koordinaten $(x(s),y(s))^T$ einer *B-spline*-Kurve werden über die Knoten $(X_i,Y_i)^T$ des Kontrollpolygons definiert (vgl. Abb. 3-3):

$$\begin{pmatrix} x(s) \\ y(s) \end{pmatrix} = \sum_{i=1}^{n} N_i^m(s) \begin{pmatrix} X_i \\ Y_i \end{pmatrix} \qquad (3\text{-}16)$$

Die Basisfunktionen N_i sind für *B-splines* vom Grad *m* vordefiniert und bekannt. Eine *B-spline*-Kurve ist daher eindeutig durch das Kontrollpolygon und durch den Grad *m* definiert. Die Unbekannten im mathematischen Ausgleichungsmodell sind die Knoten $(X_i,Y_i)^T$ des Kontrollpolygons.

Die photometrische Energie wird entlang der *spline*-Kurve $\backslash(s) = (x(s),y(s))^T$ gemessen. Die Beobachtungsgleichung für die Zuordnung der diskreten zweidimensionalen Funktion für das photometrische Muster $PM(x,y)$ (das *template)* und der Bildfunktion *I(x,y)* kann formuliert werden:

$$PM(x,y) - e(x,y) = T(I(x,y)) = G(x,y) \tag{3-17}$$

Dabei bezeichnet *T* eine abstrakte Transformation von den Grauwerten des Bildes zu den Werten des photometrischen Musters *(template).* Als *template* wird im Allgemeinen ein Rechteck von 5x5 bis 25x25 Pixeln Ausdehnung verwendet (z.B. Abb. 3-4). Die Beobachtungsgleichung (3-17) besagt, dass das transformierte Bild und das *template* ident sind. *G(x,y)* ist die Beobachtung des *template*, *e(x,y)* ist der (wahre) Fehler in der Gleichung.

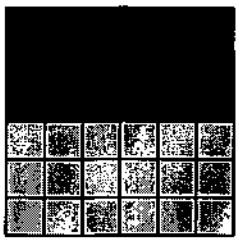


Abb. 3-4: Beispiel eines *template* einer Kante für *least squares template matching.*

Die Methode der Ausgleichung nach kleinsten Quadraten benötigt lineare Beobachtungsgleichungen. Da Gleichung (3-17) diese Bedingung nicht erfüllt, wird sie durch eine lineare Gleichung angenähert, indem die Taylor-Entwicklung nach dem ersten Glied abgebrochen wird. Dazu sind Näherungswerte $(x^0,y^0)^T$ notwendig:

$$-e(x,y) = G_x(x^0,y^0)\Delta x + G_y(x^0,y^0)\Delta y + \left(G(x^0,y^0) - PM(x,y)\right) \tag{3-18}$$

Bildet man die ersten Differenziale des *spline* aus (3-16)

$$\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix} = \sum_{i=1}^{n} N_i^m(s)\begin{pmatrix}\Delta X_i\\ \Delta Y_i\end{pmatrix} \tag{3-19}$$

und ersetzt man damit die Koordinatenzuschläge Ax und Δy in (3-18), ergeben sich die Beobachtungsgleichungen für die unbekannten Knoten des Kontrollpolygons des *B-spline*:

$$-e(x,y) = G_x\sum_{i=1}^{n} N_i^m(s)\Delta X_i + G_y\sum_{i=1}^{n} N_i^m(s)\Delta Y_i + \left(G(x^0,y^0) - PM(x,y)\right) \tag{3-20}$$

Diese Gruppe der Beobachtungsgleichungen, die interne und photometrische Energieanteile formulieren, kann in kompakter Form in Matrizenschreibweise dargestellt werden. Wir ersetzen dabei die negativen (wahren) Fehler *e* durch die in der Ausgleichung in ihrer Quadratsumme minimierten Verbesserungen v.

$$\mathrm{v} = \mathrm{G}_x\mathbf{N}\Delta\mathbf{X} + \mathbf{G}_y\mathbf{N}\Delta\mathbf{Y} - \mathbf{l} \tag{3-21}$$

Durch Zusammenfassen von Matrizen

$$\mathbf{A} = [\mathbf{G}_x\mathbf{N}, \mathbf{G}_y\mathbf{N}]$$
$$\mathbf{X} = [\Delta\mathbf{X}, \Delta\mathbf{Y}]^T \tag{3-22}$$

erhält man die gewohnte Form einer Beobachtungsgleichung für die Ausgleichungsrechnung:

$$\mathrm{v} = \mathbf{AX} - \mathbf{l} \tag{3-23}$$

Mit P als Gewichtsmatrix lautet die Lösung einer Kleinste-Quadrate-Ausgleichung $\mathbf{v}^T\mathbf{Pv} \rightarrow$ min bekanntlich:

$$\mathrm{X} = (\mathbf{A}^T\mathbf{PA})^{-1}(\mathbf{A}^T\mathbf{Pl}) \tag{3-24}$$

Die so genannten *LSB-snakes* (*least squares B-spline snakes)* benötigen weitere Regularisierungsgleichungen um Konvergenz zu gewährleisten (Grün, Li, 1997). Diese zusätzlichen Beobachtungen können als externe Energieanteile interpretiert werden:

- Die Knoten der Ausgangskurve werden als „Passpunkte" eingeführt. Die Verbesserungen zu diesen Beobachtungen werden v_c genannt.
- Die ersten und zweiten Ableitungen der *spline*-Kurve an diesen Punkten werden als weitere Passelemente eingeführt mit den Verbesserungen v_s bzw. v_{ss}.

Das gesamte Optimierungsproblem lautet:

$$\mathbf{v}_{LSB}^T\mathbf{P}_{LSB}\mathbf{v}_{LSB} = \mathbf{v}^T\mathbf{Pv} + \mathbf{v}_c^T\mathbf{P}_c\mathbf{v}_c + \mathbf{v}_s^T\mathbf{P}_s\mathbf{v}_s + \mathbf{v}_{ss}^T\mathbf{P}_{ss}\mathbf{v}_{ss} \rightarrow \mathbf{min} \tag{3-25}$$

Auch für die Optimierung mittels *least-squares*-Ausgleichung sollen Stärken und Schwächen angegeben werden:

- Das mathematische Modell erlaubt die Abschätzung der Genauigkeit des Ergebnisses. Dies kann als entscheidender Vorteil gegenüber anderen Methoden angesehen werden, da dadurch eine Selbstdiagnose möglich wird. Wie in Abschnitt 1.3 ausgeführt, ist die Fähigkeit zur Selbstdiagnose eine Voraussetzung um eine Methode in einem autonomen System einsetzen zu können.
- Ebenso erlaubt das Modell die Homogenisierung von Beobachtungsgleichungen, wodurch die Vergabe von verschiedenen Gewichten für unterschiedliche Beobachtungen leichter verständlich wird. Darauf komme ich in Abschnitt 4.4 zurück.
- Als Hauptnachteil soll angeführt werden, dass die Anforderung an die Ausgangslage der *snake* wesentlich höher sind, da die Lösung des Optimierungsproblems bis zur letzten Iteration von der Ausgangslage abhängig ist. Während bei anderen Optimierungsmethoden die *snake* ihre Ausgangslage nach mehreren Iterationen oft vollständig verlässt und immer nur der aktuelle Status der *snake* für die jeweilige Iteration verwendet wird, gehen bei *LSB-snakes* die Knoten der Ausgangslage sowie die ersten und zweiten Ableitungen des ursprünglichen *spline* als Passelemente in alle Iterationen der Aus-

gleichung ein. Um die Abhängigkeit etwas zu verringern, müsste man nach jeder Iteration die Beobachtungen der zusätzlichen Passelemente (Koordinaten, Tangente und Krümmung der Kurve in den Passpunkten) durch die aus der Ausgleichung entstandenen Werte ersetzen.

Wegen des zuletzt genannten Nachteils lässt sich diese Methode für jene Anwendungen nicht einsetzen, bei denen die Ausgangslage zwar nahe der Zielkurve liegt, aber nicht exakt auf ihr (auch nicht in wenigen Punkten). Zu solchen Anwendungen zählt etwa das Verfolgen einer Kurve in einer Serie von Bildern. Für den Einsatz in einem semi-automatischen System, in dem der Operateur die Ausgangslage vorgibt, fällt der Nachteil kaum ins Gewicht, denn der Operateur kann leicht die Ausgangslage in wenigen, exakten Punkten vorgeben.

3.4 Lösung über eine *Level-Set*-Methode

Das bisher beschriebene *snake*-Modell wird manchmal als parametrisches Modell bezeichnet, da in der ursprünglichen Definition die Kurve über den Parameter s parameterisiert ist (siehe Gleichung (2-1)). Im Gegensatz dazu gibt es *geometric deformable models,* die auf der Theorie der *curve evolution* und der *level set* Methode aufbauen (Osher, Sethian, 1988). Dabei werden geometrische Maße verwendet, die unabhängig von der Parameterisierung der Kurve sind. *Geometrie active contours* stellen heute eine bedeutende Alternative zum herkömmlichen *snake*-Ansatz dar. Diese Gruppe von *snakes* wurde ebenfalls in der Literatur intensiv untersucht. In dieser Arbeit können lediglich wenige grundlegende Aspekte dieses Ansatzes beschrieben werden.

Ausgangspunkt dieses Modells ist eine *propagating front* mit krümmungsabhängiger Geschwindigkeit. Die Front ist mit der Flammenfront bei einem Flächenbrand vergleichbar, welche dem Huygens'schen Prinzip folgt: Wenn ein Partikel einmal verbrannt ist, kann es nicht wieder entfacht werden.

Die Entwicklung der *active contour* wird durch eine so genannte Geschwindigkeitsfunktion *(velocity* oder speed *function)* gesteuert. Die Kurve (Front) ist eine geschlossene, sich nicht überschneidende Kurve und bewegt sich dabei entlang ihrer Normalvektoren. Die Geschwindigkeit kann konstant sein, wird aber meist in Abhängigkeit der Krümmung der Kurve gesetzt.

Als zentrale Idee der *level-set*-Methode werden die Kurven *(active contours) zur* Menge der Höhenlinien *(level set)* einer Oberfläche in einer bestimmten Höhe in Beziehung gesetzt. Daraus erkennt man sofort die Einschränkung der Methode auf geschlossene Kurven. Im Speziellen werden die Höhenlinien in Höhe Null betrachtet.

Mathematisch formuliert: Es sei eine Funktion $\psi(\mathbf{x})$ gegeben, aus der sich eine Menge von Kurven mit $\psi = 0$ ergibt:

$$\gamma(t) = \{\mathbf{x} \mid \psi(\mathbf{x}, t) = 0\} \tag{3-26}$$

Die Funktion ψ ist demnach eine implizite Repräsentation der Kurven $\gamma(t)$. Die Darstellung der Kurven ist parameterfrei bzw. intrinsisch. Daher wird wie erwähnt das klassische *snake*-Modell in diesem Zusammenhang auch als parametrisches Modell bezeichnet im Gegensatz zum hier behandelten geometrischen *snake*-Modell.

Die Entwicklung der Kurven $\gamma(t)$ über die Zeit t wird durch das Verschieben der Funktion ψ entlang der vertikalen Achse erreicht. Als Beispiel soll ein Kreis betrachtet werden, dessen Durchmesser mit der Zeit wachsen soll. Die zugehörige Funktion ψ könnte ein Kegel sein, wie in Abb. 3-5 gezeigt.

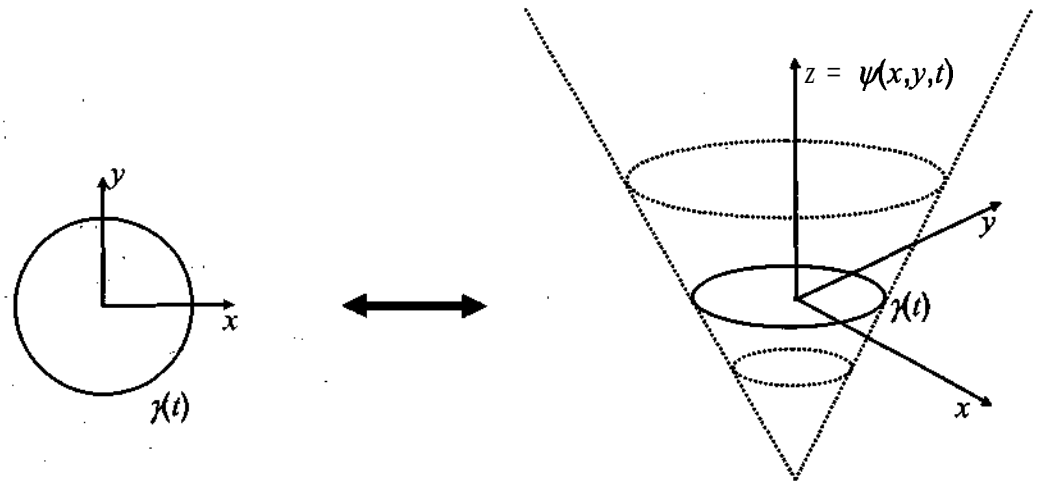


Abb. 3-5: *Level-set*-Formulierung eines expandierenden Kreises. Die Kurve $\gamma(t)$ entspricht der Höhenlinie in Höhe Null der Funktion $\psi(x,y,t)$. Die Entwicklung der Kurve $\gamma(t)$ ergibt sich aus der Bewegung der Funktion ψ entlang der z-Achse mit der Zeit t.

Als Bedingung für die Formulierung der Funktion ψ gilt, dass zum Startzeitpunkt $t = 0$ die Kurve $\gamma(t)$ mit der Ausgangslage der *active* contour übereinstimmen muss:

$$\gamma(t=0) = \{\mathbf{x} \mid \psi(\mathbf{x}, t=0) = 0\} \tag{3-27}$$

Durch die geeignete Formulierung der Funktion ψ kann eine krümmungsabhängige Fortschreitungsgeschwindigkeit der Kurve erreicht werden. Das heißt, dass stärker gekrümmte Teile der Kurve langsamer expandieren als schwächer gekrümmte, sodass für $t \to \infty$ eine gleichmäßig (minimal) gekrümmte Kurve (d.h. ein Kreis) angestrebt wird.

Die Bestimmung der Kurve $\gamma(t)$ geht von der Tatsache aus, dass für die Kurve die Funktion ψ zu jedem Zeitpunkt gleich Null ist, d.h.

$$\psi(\mathbf{x}(t), t) = 0 \tag{3-28}$$

und

$$\frac{\partial}{\partial t}\psi(\mathbf{x}(t), t) = \psi_t + \sum_{i=1}^{N} \psi_{x_i} x_{i_t} = 0 \tag{3-29}$$

mit x_i die i-te Komponente von x und $N = 2$ für bivariate Kurven. Setzt man:

$$\sum_{i=1}^{N} \psi_{x_i} x_{i_t} = \begin{pmatrix} \psi_{x_1} \\ \psi_{x_2} \\ \vdots \\ \psi_{x_N} \end{pmatrix} \begin{pmatrix} x_{1_t} \\ x_{2_t} \\ \\ x_{N_t} \end{pmatrix} = F(\mathbf{x}(t))|\nabla \psi| \tag{3-30}$$

erhält man:

$$\psi_t + F|\nabla \psi| = 0 \tag{3-31}$$

mit gegebenem $\psi(\mathbf{x}, t = 0)$. Es handelt sich um eine Gleichung vom Typ Hamilton-Jacobi, da sich für bestimmte Formen der Geschwindigkeitsfunktion F die standardmäßige Hamilton-Jacobi Gleichung ergibt. Die Lösung dieser Gleichung findet sich in Osher, Sethian (1988).

Malladi et al. (1995) verwenden nun die Methode der *front propagation zur* Extraktion linienhafter Elemente in Bildern. Während im ursprünglichen *front propagation* Modell die Fortschreitungsgeschwindigkeit der Kurve ausschließlich als krümmungsabhängiger Term formuliert wurde, muss nun ein zusätzlicher Term in die Geschwindigkeitsfunktion aufgenommen werden, der die Geschwindigkeit in der Nähe von markanten Bildkanten (oder Linien) reduziert. Dadurch wird die Entwicklung der Kurve gestoppt, sobald die Kurve die zu extrahierende Bildkante erreicht hat. Weiters wird die generelle Fortschreitungsrichtung (Expansion oder Kontraktion) als Geschwindigkeitsterm modelliert.

Man erkennt die Analogie zwischen einerseits der Geschwindigkeitsfunktion beim *front propagation* Modell mit einem krümmungsabhängigen Term und einem zusätzlichen Term, der vom Bildinhalt abhängt, und andererseits der Energiefunktion mit internen Termen und photometrischen Termen beim klassischen *snake*-Ansatz. Tatsächlich haben Caselles et al. (1997) die mathematische Überführung des *front propagation* Ansatzes in den Energie minimierenden Ansatz herleiten können: Ein spezieller Fall der Energie basierten *snake,* nämlich wenn die interne Energie keinen Term zweiter Ordnung besitzt (d.h. $\beta = 0$ in Formel 2-4), ist äquivalent zum Problem der *geodesic active contours,* bei denen eine geodätische Linie in einem Riemann Raum, der aus dem Bild abgeleitet ist, gesucht wird.

Der große Vorteil der geometrischen *snakes* liegt in der Möglichkeit, dass sich die Topologie der Kurve während ihrer Entwicklung ändern kann. Damit ist gemeint, dass sich eine Kurve in mehrere Kurven teilen kann oder dass sich umgekehrt mehrere Kurven vereinen können. Eine Anwendung, für die dies ein entscheidender Vorteil sein kann, ist die Verfolgung einer Objektkontur in aufeinander folgenden Schichtbildern eines volumetrischen Voxelmodells. Dreidimensionale Objekte mit konkaven Objektteilen (Einbuchtungen) bilden sich in manchen Schichtbildern als eine einzige Region ab, in anderen Bildern können sie aber auch auf mehrere Regionen abgebildet sein. Ein anderes Beispiel sind Objekte mit Löchern. Will man die Begrenzung eines derartigen Objekts bestimmen, müssen ebenfalls mehrere Polygone als eine gemeinsame Kontur behandelt werden.

Die Topologieänderung der Kurve bzw. der Kurvenschar $\gamma(t)$ soll anhand des Beispieles in Abb. 3-6 veranschaulicht werden.

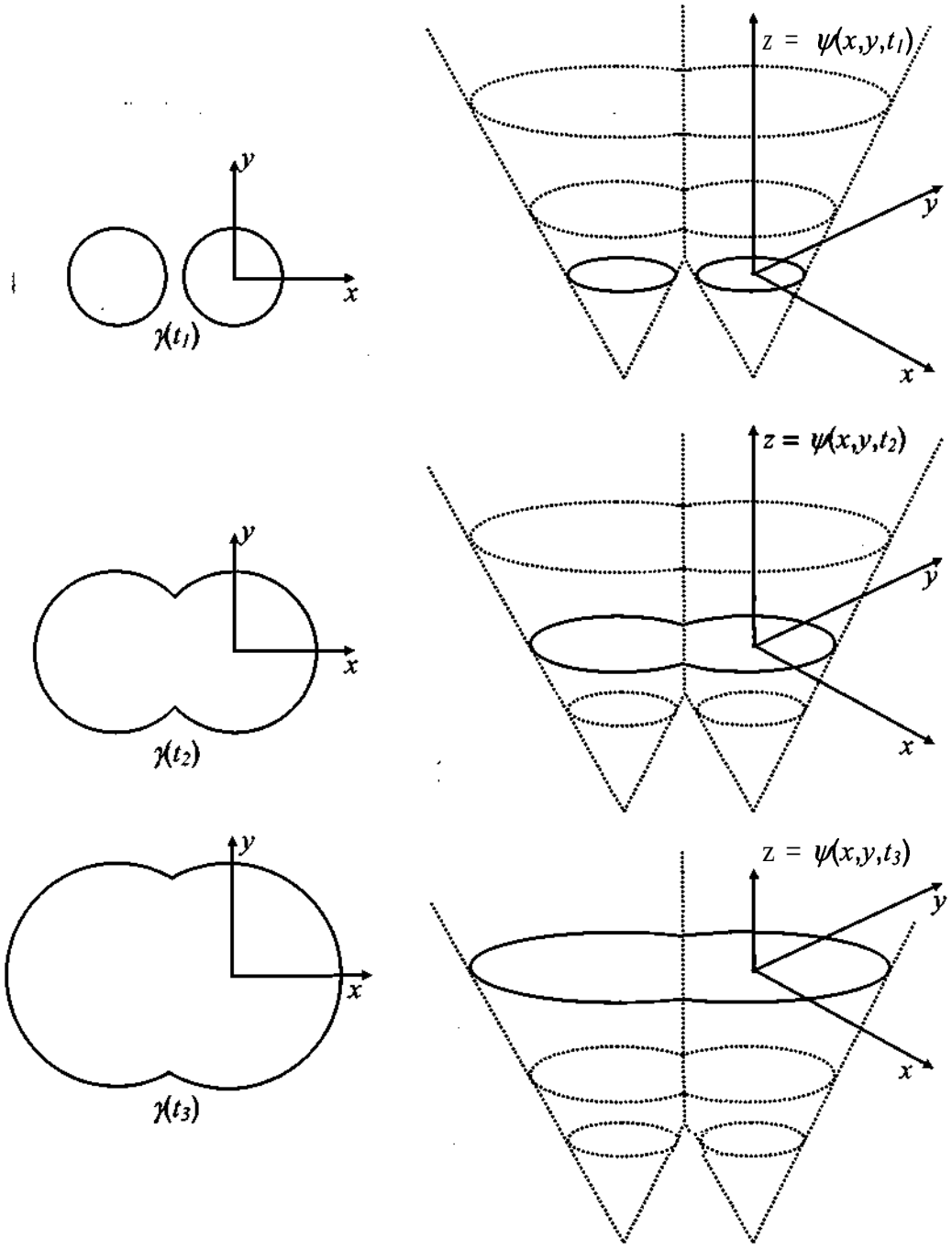


Abb. 3-6: Beispiel einer Kurvenschar *(level set)* zum Zeitpunkt t_1, die zu den Zeitpunkten t_2 und t_3 zu einer einzigen Kurve verschmolzen ist. Derartige Topologieänderungen der *snakes* können mit der *level-set*-Optimierungsmethode realisiert werden.

Abschließend sollen auch für den Lösungsansatz über *level set curve evolution* Vor- und Nachteile zusammengefasst werden.

- Als größter Vorteil geometrischer *snake*-Modelle im Vergleich zu parametrischen Modellen gilt ihre Fähigkeit, sich während der Entwicklung topologisch zu ändern. Damit ist gemeint, dass sich die Kurve in mehrere Teilkurven aufteilen oder sich wieder vereinen kann. Diese Eigenschaft kann von besonderem Interesse sein, wenn mehrere getrennte Objekte in einem Bild gesucht werden ohne im Voraus ihre Anzahl zu kennen.
- Für starke Einbuchtungen ist dieser Ansatz besser geeignet als andere *snake*-Modelle. Wird die Krümmung der Kurve an stark gekrümmten Abschnitten der Kurve sehr hoch, sinkt zwar die Expansionsgeschwindigkeit, doch nach einer größeren Anzahl an Iterationen werden auch diese Abschnitte exakt rekonstruiert.
- Das Ergebnis ist relativ unabhängig von der Ausgangslage der *snake.* Der Grund dafür ist derselbe wie beim Ballonmodell (vgl. Abschnitt 4.1.2): Eine Expansionskraft bewirkt, dass sich die *snake* immer weiter ausbreitet, bis sie an markanten Bilddetails hängen bleibt.

Nachteile:

- Diese Methode lässt sich nur auf geschlossene Kurven anwenden.
- Die Fähigkeit dieses Modells, stärker gekrümmte Abschnitte detailliert extrahieren zu können, wird durch eine Abschwächung der Robustheit gegenüber Lücken in den Zielkurven erkauft. Wenn die *geodesic snake* sogar auch schlauchförmige Einbuchtungen rekonstruieren kann, wird sie auf der anderen Seite durch eine Lücke hindurch immer weiter expandieren.
- In den ersten Modellen geometrischer *snakes* stoppt die *snake* an Bildkanten nicht vollständig, sondern ihre Geschwindigkeit wird nur reduziert. Dadurch konnte es passieren, dass Teile der *snake* die Zielkurve überlaufen: Solange nicht die gesamte Kurve die Zielkurve erreicht hat, kann die Kurvenentwicklung noch nicht abgebrochen werden. Daher überschreitet der Teil der *snake,* der die Zielkurve bereits früh erreicht hat, in weiteren Iterationen langsam die Zielkurve, bis erst die anderen Teile der *snake* die Zielkurve erreicht haben. Schließlich kann es passieren, dass sich die Geschwindigkeit für diesen Teil sogar wieder erhöht und dieser sich von der Zielkurve wieder entfernt.

  Um dies zu vermeiden, setzen verbesserte Modelle die Geschwindigkeit in der Nähe der Zielkurve auf Null (Caselles et al., 1997).
- Diese Methode ist anfälliger auf lokale Minima als andere Methoden. Da an markanten Bilddetails die Geschwindigkeit auf Null gesetzt wird, kann die *snake* ein lokales Minimum nicht überschreiten, auch wenn nur ein Teil der *snake* gefangen ist.

3.5 Lösung über *Simulated Annealing*

Simulated Annealing bedeutet übersetzt „simuliertes Kristallisieren". Die Methode versucht, den physikalischen Prozess der Umwandlung vom flüssigen zum festen Aggregatzustand während des Abkühlens in einem Algorithmus nachzubilden. Ein

derartiger Algorithmus kann auch für die Optimierungsaufgabe der *snakes* verwendet werden (Trinder et al., 2000).

Ein *simulated annealing* Algorithmus generiert Serien von Zuständen nach folgendem Prinzip:

> Ausgehend von einem Status C_i mit der Energie E_i wird der Zustand leicht verändert (Perturbation), woraus sich der Status C_j (mit der Energie E_j) ergibt. Falls dadurch die Energie verringert wurde, wird der neue Status übernommen. Andernfalls wird der neue Zustand mit einer Wahrscheinlichkeit von
>
> $$p = \exp\left(-\frac{E_j - E_i}{kT}\right) \tag{3-32}$$
>
> akzeptiert, wobei T die Temperatur und k die Boltzmann-Konstante bedeutet.

Es muss ein „Generator" definiert werden, der einen neuen Status auf zufällige Weise erzeugen kann. Im Beispiel der *snakes* werden Perturbationen durch das Verschieben von Knoten erreicht. Aus Effizienzgründen wird im Allgemeinen der Suchbereich eingeschränkt, d.h. der generierte Status muss innerhalb bestimmter Grenzen liegen.

Der Generator braucht pro Iteration nur einen einzigen neuen Status zu erzeugen. Falls die Energie des neuen Status größer als die des alten Status ist, muss eine Zufallszahl zwischen Null und Eins bestimmt werden, die entscheidet, ob der neue Status übernommen wird. Eine solche Iteration braucht sehr wenig Zeit.

Kritisch ist die Wahl der Anzahl der Iterationen in jeder Temperaturstufe. Wird der Abkühlungsprozess langsam genug ausgeführt, erreicht das Kristall bei jeder Temperatur ein thermisches Gleichgewicht. Es ist jedoch nicht einfach zu erkennen, wann dieses thermische Gleichgewicht erreicht ist.

Falls sich der Ausgangsstatus der *snake* nicht in der Nähe der Zielkurve befindet, empfiehlt es sich vor dem Abkühlungsprozess die Temperatur zu erhöhen, da es dadurch wahrscheinlicher wird, dass die *snake* über lokale Minima hinwegkommt und zum globalen Minimum findet (Grzeszczuk, Levin, 1997).

Die Stärken und Schwächen sollen auch für diese Methode zusammengefasst werden:

- Die entscheidende Stärke dieser Optimierungsmethode liegt darin, dass Zustandsänderungen, die die Energie der *snake* erhöhen, nicht von vornherein verworfen werden, sondern nur mit geringerer Wahrscheinlichkeit versehen werden. Dadurch kann die *snake* ein lokales Minimum wieder verlassen und findet oft das globale Minimum unabhängig von ihrer Ausgangslage.
- Die Methode erlaubt beliebige Formulierungen der Energiefunktion.

- Eine einzelne Iteration ist zwar schnell, doch werden sehr viele Iterationen in jeder Temperaturstufe benötigt. Die Schwierigkeit liegt darin, zu erkennen, wann die Temperatur gesenkt werden darf. Die Wahl dieser Abkühlungsgeschwindigkeit entscheidet darüber, ob die *snake* von einem lokalen Minimum gefangen wird. Wenn sie schlecht gewählt wurde, kann es durchaus passieren, dass die *snake* ein lokales Minimum detektiert, das von der Ausgangslage sogar weiter entfernt ist, als das globale Minimum.
- Trinder et al. (2000) merken an, dass Probleme dann entstehen, wenn mehrere zu extrahierende Elemente nahe aneinander liegen, da in diesem Fall nicht mehr gewährleistet werden kann, dass die extrahierte *snake* nicht zwischen den Elementen springt.

3.6 Diskussion der verschiedenen Optimierungsmethoden

Trotz der Vielzahl an Veröffentlichungen zum Thema *snakes* bzw. *active contours* und der Vielzahl an alternativen Optimierungsmethoden wurde nie ein umfassender Vergleich der Verfahren präsentiert. Jede einzelne Verbesserung verspricht verbesserte Eigenschaften der *snake* bezüglich eines bestimmten Kriteriums. Meist werden Verbesserungen mit Einschränkungen in einer anderen Eigenschaft erkauft. Oft werden solche Verschlechterungen nicht einmal erwähnt, da sie in der speziellen Anwendung, für die die jeweilige Methode entwickelt wurde, von untergeordneter Bedeutung waren. Beispielsweise setzt die *level-set*-Methode geschlossene Kurven voraus, was für manche Anwendung allerdings eine zu große Einschränkung darstellt.

Eine umfassende vergleichende Untersuchung der Stärken und Schwächen der einzelnen Optimierungsmethoden wäre von großer Bedeutung, will man für eine spezielle Anwendung eine Methode auswählen. Da die Programmierung der unterschiedlichen Algorithmen den Rahmen dieser Arbeit sprengen würde, kann eine sehr fundierte Untersuchung auch hier nicht gegeben werden.

Nichtsdestotrotz soll an dieser Stelle der Versuch gemacht werden, Vor- und Nachteile der einzelnen Methoden gegenüber zu stellen (siehe Tabelle 3-1). Dabei sollen auch Einschränkungen einzelner Ansätze erwähnt werden. Die Bewertung wurde dabei vorwiegend aus Aussagen in den verschiedensten publizierten Artikeln getroffen. Selten wurden die Schwächen einer bestimmten Methode auch in dem Aufsatz erwähnt, in dem die Methode vorgestellt wurde. Meist zeigen erst andere Autoren die Nachteile auf. Ich möchte an dieser Stelle aber auch manche Bewertungen abgeben, die ich aus erwähnten Gründen nicht belegen kann. Beispielsweise der Vergleich der Rechenzeiten konnte lediglich nur grob geschätzt werden.

Ein Kriterium eines Vergleichs ist die Rechenzeit. Diese spielt insofern eine Rolle als das Konzept der *snakes* im Allgemeinen in einem semi-automatischen System Anwendung findet. Daher beobachtet ein Operateur den Vorgang. Die Optimierung der *snake* muss in jedem Fall schneller sein als das detaillierte manuelle Digitalisie-

ren der Zielkurve. Die Rechenzeit ist in Kombination mit der Anzahl der Iterationen zu sehen.

	Variations-rechnung	Dynamische Programm.	*LSB-snakes*	*Level-set snakes*	*Simulated Annealing*
Rechenzeit pro Iteration[5]	1. Iteration mittel, alle weiteren Iterationen gering	je nach Bewegungsfreiheit der Knoten mittel bis groß	groß	kann ich nicht beurteilen (eher klein)	gering (zufällige Auswahl + Energiebe-rechnung)
Anzahl an Iterationen	klein	je nach Bewegungsfreiheit der Knoten mittel bis klein	eher mittel	kann ich nicht beurteilen (eher klein)	sehr groß
Energieterme	Voraussetzung differenzierbar	keine Einschränkungen	Voraussetzung differenzierbar	Formulierung vorgegeben	keine Einschränkungen
hard constraints[6]	nicht möglich	möglich	möglich[7]	nicht möglich	möglich
für jeden Knoten individuelle Gewichte	möglich, erhöht allerdings die Rechenzeit deutlich	möglich	möglich	nicht möglich	möglich
lokales oder globales Minimum	wird von lokalen Minima gefangen	findet garantiert globales Minimum (falls es im Suchbereich ist)	wird von lokalen Minima gefangen	wird von lokalen Minima gefangen	findet globales Minimum (falls es im Suchbereich ist und langsam genug abgekühlt wird)
Abhängigkeit von der Lage der Startkurve	Startkurve darf nicht sehr weit entfernt sein	Startkurve darf nicht sehr weit entfernt sein	die Knoten der Startkurve müssen sehr nahe der Zielkurve liegen[8]	relativ unabhängig	relativ unabhängig (der Suchbereich muss groß genug gewählt werden)
offene / geschlossene Kurven	offen oder geschlossen	offen oder geschlossen	offen oder geschlossen	geschlossen	offen oder geschlossen
Änderung der Topologie möglich	nein	nein	nein	ja	nein

Tabelle 3-1: Zusammenfassung der Stärken und Schwächen verschiedener Optimierungsmethoden.

---

[5] Der Vergleich der Rechenzeiten ist lediglich eine grobe Schätzung.

[6] *Hard constraints* sind Bedingungen, die unbedingt erfüllt sein müssen im Gegensatz zu anderen Kriterien *(soft constraints)* in der Energiefunktion, die als ein Teil der Gesamtenergie optimiert werden.

[7] Mittels zusätzlicher Beobachtungsgleichungen.

[8] Die Koordinaten werden bis zur letzten Iteration als Beobachtungen der Ausgleichung verwendet.

Am schnellsten arbeitet die Lösung über Variationsrechnung. Nur in der ersten Iteration muss eine aufwändige Matrizeninversion durchgeführt werden. Bei der Lösung über dynamische Programmierung sollte die Bewegungsfreiheit der Knoten klein gehalten werden. Dann hält sich auch hier die Rechenzeit in Grenzen.

Ein Kriterium, das für Anwendungen der Praxis eine Rolle spielt, sind die Voraussetzungen, die an erlaubte Energieterme gestellt werden. In dieser Hinsicht sind die Lösungen über dynamische Programmierung und *simulated annealing* am wenigsten restriktiv. Beide erlauben zudem den Einsatz von *hard constraints.*

Die Frage, ob für die einzelnen Knoten individuelle Gewichte gesetzt werden können, mag ebenfalls für manche praktische Anwendungen von Bedeutung sein. Zwar ist es schwierig, für einen speziellen Knoten beispielsweise starke Krümmung zuzulassen, da es schwer vorhersagbar ist, welcher Knoten an der stark gekrümmten Stelle der Zielkurve zu liegen kommt, doch will der Operateur eventuell für einen Abschnitt der *snake* stärkere Krümmungen erlauben oder die photometrische Energie stärker gewichten etc.. Bei den meisten Lösungsmethoden ist dies möglich. Bei der Variationslösung muss allerdings die Matrizeninversion wiederholt werden.

Lokale Minima sind generell für alle Lösungsmethoden ein Problem. Wie erwähnt, wird für die Lösung über dynamische Programmierung behauptet, sie finde garantiert das globale Minimum. Allerdings muss man einschränken, dass sie innerhalb einer Iteration nur einen sehr beschränkten Suchradius hat. Wenn in diesem Suchbereich nur ein lokales Minimum und nicht das globale Minimum enthalten sind, wird die *snake* niemals das globale Minimum finden. Ähnliches gilt für die Lösung über *simulated annealing.*

Die Abhängigkeit von einem Ausgangsstatus der *snake,* der die Zielkurve bereits gut approximiert, kann über das Wechseln der Optimierungsmethode nicht verringert werden. Lediglich *level-set-snakes* beinhalten ein Ballon-Modell (vgl. Abschnitt 4.1.2) und stellen daher nur geringe Anforderungen an den Ausgangsstatus. Die größte Abhängigkeit vom Ausgangsstatus haben wie erwähnt *LSB-snakes.*

Die Einschränkung auf geschlossene Kurven haben nur *level-set-snakes.* Dafür können solche *snakes* sich teilen, sich vereinen oder Löcher modellieren.

Man erkennt, dass nur ein Teil der am Ende des Abschnittes 2.3 zusammengefassten Probleme durch die Wahl einer geeigneten Optimierungsmethode gelöst oder umgangen werden können. Es bleiben weitere Probleme bestehen (z.B. das Schrumpfen, etc.), deren Ursache in der Formulierung der Energiefunktiön gesucht werden muss. Dies folgt im nächsten Kapitel. Probleme, die auch durch eine adaptierte Energiefunktion nicht behoben werden können, kann man eventuell im Kapitel 5 durch die Wahl einer geeigneten Anwendungsstrategie in den Griff bekommen.

4 Energieterme

In diesem Kapitel sollen Terme, die in der Energiefunktion von *snakes* verwendet werden, im Detail untersucht werden. Viele der in Abschnitt 2.3 angesprochenen Schwierigkeiten lassen sich auf Unzulänglichkeiten in der mathematischen Formulierung der Energieterme zurückführen. Für manche Terme wurden in der Literatur inzwischen bessere Formulierungen vorgeschlagen.

Für jede Gruppe von Energietermen (interne, photometrische und externe) sollen in den folgenden Abschnitten die ursprünglich vorgeschlagenen Formulierungen ebenso wie manche verbesserte Varianten geprüft werden, ob sie den gewünschten Einfluss auf die *snake* besitzen und ob sie Nebenwirkungen haben. Teilweise werden eigene Vorschläge zu verbesserten Formulierungen gemacht.

Jeder Energieterm hat einen bestimmten numerischen Bereich an Energiewerten, die der Term annehmen kann. Um den relativen Einfluss der verschiedenen Terme steuern zu können wäre es wünschenswert, wenn der Wertebereich aller Energieterme gleich wäre. Dies ist auf Grund der vollkommen unterschiedlichen Herleitung der Energiewerte nicht einfach zu erreichen. Dieses Thema wird im letzten Abschnitt dieses Kapitels behandelt.

4.1 Interne Energieterme

Interne Energieterme steuern die Form der *snake.* Sie sollen dann niedrige Werte annehmen, wenn die *snake* eine glatte Form ohne starke Krümmungen oder gar Knicke aufweist.

Wie wir erkennen werden, sind sie aber auch für das Schrumpfen der *snake* verantwortlich. Die ideale Form der *snake* ist nämlich dann erreicht, wenn sie zu einem Punkt degeneriert ist. In diesem Abschnitt soll eine neue Formulierung der internen Energie gefunden werden, sodass die *snake* ihre Länge beibehält.

4.1.1 Das ursprüngliche Konzept

Kass et al. (1987) leiten die Terme von jenen Kräften ab, die auch von *splines* eingesetzt werden um die Glattheit zu erreichen. *Splines* sind zusammengesetzte Polynome vom Grad *m,* die eine kontinuierliche Kurve in eine Serie von *N* Knoten interpolieren (Kraus, 2000, Kapitel H.2).

$$\mathbf{v}(s) = \sum_{n=1}^{N-1} \mathbf{p}_n^{[m]}(s)\chi(s) \tag{4-1}$$

mit s als Kurvenparameter, einer Funktion $\chi(s)$:

$$\chi(s) = \begin{cases} 1 & s \in [s_n, s_{n+1}] \\ 0 & \text{sonst} \end{cases} \tag{4-2}$$

und den Polynomen vom Grad *m:*

$$\mathbf{p}_n^{[m]}(s) = \sum_{i=0}^{m} \mathbf{a}_{i,n} (s - s_n)^i \tag{4-3}$$

Häufig werden Polynome dritten Grades verwendet (kubische *splines),* für die C^2-Stetigkeit verlangt wird (so genannte natürliche *splines),* d.h. sowohl die Tangenten als auch die Krümmungen zweier aufeinander folgender Polynome sollen in den Knoten übereinstimmen. Eine andere Methode zur Herleitung eines natürlichen *splines* geht über die Minimierung eines Funktionais (Pfeifer, 2002, Abschnitt 3.4):

$$E_{TP} = \int (\mathbf{v}_{ss}(s))^2 \, ds \tag{4-4}$$

Dieses Funktional wird *thin plate energy* genannt, da es dem bivariaten *spline* das Verhalten einer dünnen Platte gibt bzw. dem univariaten spline das Verhalten eines dünnen Stabs. Daher ist auch der Name Biegeenergie gebräuchlich. Ein anderes Funktional, das vielfach verwendet wird, gibt dem *spline* das Verhalten einer Membran (bzw. dem univariaten *spline* das Verhalten eines Gummibandes). Dieses Funktional wird als *membrane energy* bezeichnet:

$$E_M = \int (\mathbf{v}_s(s))^2 \, ds \tag{4-5}$$

Wie bereits in Abschnitt 2.2.1 (Formel (2-4)) erwähnt, wurde die interne Energie der *snakes* ursprünglich als Kombination von *membrane energy* und *thin plate energy* formuliert:

$$E_{int}(s) = \tfrac{1}{2}(\alpha(s)E_{cont} + \beta(s)E_{curv}) = \tfrac{1}{2}\left(\alpha(s)|\mathbf{v}_s(s)|^2 + \beta(s)|\mathbf{v}_{ss}(s)|^2\right) \tag{4-6}$$

Im *snake*-Modell werden diese Energieterme üblicherweise als *continuity* Term E_{cont} (mit einem Gewicht $\alpha(s)$) und als *curvature* Term E_{curv} (mit einem Gewicht $\beta(s)$) bezeichnet. Der *continuity* Term soll das Dehnen der Kurve hemmen, der *curvature* Term soll starke Windungen der Kurve unterbinden.

In der diskreten Approximation wird die Kurve durch einen Polygonzug approximiert, d.h. durch eine Serie von Polygonpunkten $\mathbf{v}_i$. Die beiden Energieterme werden durch finite Differenzen approximiert:

$$\begin{aligned} E_{com} &\approx |\mathbf{v}_i - \mathbf{v}_{i-1}|^2 \\ E_{curv} &\approx |\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}|^2 \end{aligned} \tag{4-7}$$

Diese Näherungen setzen voraus, dass die Kurve im Einheitsabstand diskretisiert wurde und dass der Parameter *s* die Bogenlänge ist (Williams und Shah, 1992). Diese Voraussetzungen sind im Allgemeinen allerdings nicht erfüllt.

Eines der Hauptprobleme von *snakes* liegt in ihrer inhärenten Tendenz zu schrumpfen, wie verschiedene Autoren feststellen (z.B. Amini et al, 1988, Cohen und Cohen, 1993 oder Hoch und Litwinowicz, 1996). Die Gründe dafür sind in der Formulierung der internen Energie zu suchen. Wenn nicht der Einfluss anderer Energieterme überwiegt, wandern die Knoten von Iteration zu Iteration näher zusammen bis die *snake* nach etlichen Iterationen deutlich an Länge verloren hat (siehe Abb. 4-1).

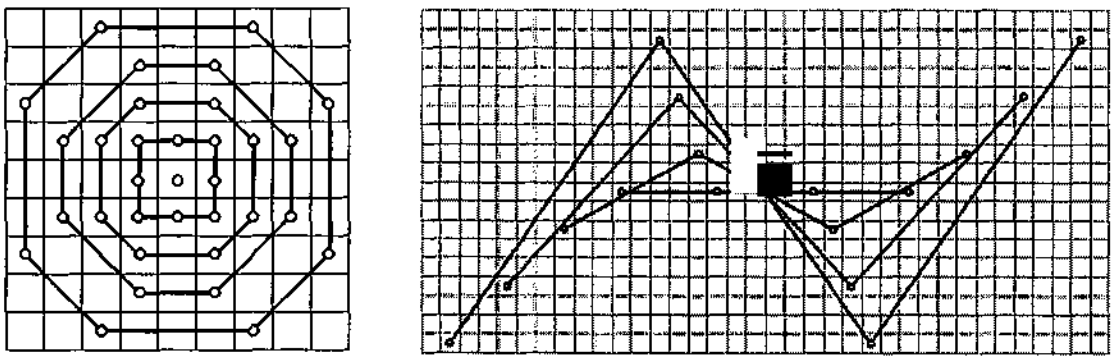


Abb. 4-1: Beispiele einer geschlossenen und einer offenen *snake,* auf die ausschließlich die beiden internen Energieterme von (4-7) gleichgewichtig wirken. Das Achteck auf der linken Seite zieht sich von Iteration zu Iteration mehr und mehr zusammen, bis der minimale Energiezustand erreicht ist, wo alle Punkte ident sind. Auch die offene Zickzacklinie verliert deutlich an Länge (jede dritte Iteration ist dargestellt), bis die optimale Form - eine Gerade - erreicht ist. In der Fortsetzung würde der *continuity* Term bewirken, dass sich die Gerade weiter **zusammenzieht,** bis die vier Knoten am Ende in einem Punkt zusammenfallen.

In der Literatur wurden verschiedene Möglichkeiten präsentiert dieses Problem in den Griff zu bekommen.

4.1.2 Das Ballonmodell

Die Lösung über einen zusätzlichen Energieterm, der eine *inflation force* modelliert (Cohen, Cohen, 1993), wird vielfach als Lösung des Problems angesehen. Dieses Modell kann allerdings nur auf geschlossene Kurven angewendet werden. Das Modell lässt sich anschaulich erklären, indem man sich einen Luftballon vorstellt, in den Luft eingeblasen wird. Ausgehend von einer geschlossenen Kurve, die gänzlich innerhalb der Zielkurve liegt, dehnt sich das Polygon mehr und mehr aus, bis es an der Bildkante/Linie hängen bleibt. Um eine genaue Kantenrepräsentation zu erhalten, wird die letzte Iteration ohne *inflation force* gerechnet.

Das *balloon*-**Modell** löst auch das Initialisierungsproblem. Nun ist es nicht mehr notwendig, dass die Ausgangslage der *snake* nahe an der Zielkurve liegen muss. Es muss lediglich gewährleistet sein, dass der *balloon* gänzlich innerhalb der Zielkurve liegt. Analog zum expandierenden *balloon* mit Inflationskräften in seiner Energiefunktion können auch Deflationskräfte angesetzt werden. In diesem Fall muss die Aus-

gangslage gänzlich außerhalb der Zielkurve liegen und der kontrahierende *balloon* nähert sich von außen an die Zielkurve an.

Inflationskräfte werden auch im Modell der *level set snakes* verwendet (Malladi et al., 1995; vgl. Abschnitt 3.4).

4.1.3 Fixieren der Endpunkte

Hoch und Litwinowicz (1996) präsentieren eine Lösung um für gestreckte offene Kurven das Schrumpfen zu unterbinden. Sie fixieren die Endpunkte der *snake* und lassen sie nicht am Optimierungsprozess teilhaben. Dieser Ansatz stellt natürlich hohe Anforderungen an die Ausgangslage der *snake.*

Besser wäre die Einschränkung, dass sich die Endpunkte der *snake* nur entlang der Normalen auf die Kurve bewegen dürfen.

4.1.4 Einführen von *Hard Constraints*

Für Amini et al. (1988) war das Schrumpfen einer der Ansatzpunkte zur Entwicklung der alternativen Lösung des Optimierungsproblems über dynamische Programmierung (siehe Abschnitt 3.2). Dabei können sie so genannte *hard constraints* einführen (Bedingungen, die nicht Teil der Energiefunktion sind sondern unbedingt erfüllt sein müssen). Auf diese Weise können sie einen Mindestabstand von aufeinander folgenden Knoten beibehalten und so das Schrumpfen unterdrücken.

4.1.5 Alternativer *Continuity* Term

Will man derartige Kunstgriffe über *inflation force,* fixierte Endpunkte oder *hard constraints* vermeiden, sollte man versuchen, durch Umformulieren der Energieterme das Problem an der Wurzel zu beseitigen.

Analysiert man den Einfluss der beiden internen Energieterme, erkennt man rasch, dass der *continuity* Term das Schrumpfen direkt fördert. In (4-7) ist offensichtlich, dass der Beitrag des *continuity* Terms zweier aufeinander folgender Knoten M und *i,* kleiner wird, wenn die beiden Knoten näher zueinander rücken. Betrachtet man die aufsummierte gesamte Energie der *snake,* wird die Tendenz aller Knoten näher zusammenzurücken klar. Es stellt sich die Frage, inwieweit ein Term dieser Art notwendig ist, um eine glatte Kurve zu bewirken. Ursprünglich wurde dieser Term in Anlehnung an *splines* formuliert.

Williams und Shah (1992) stellten bereits fest, dass dieser Term das Schrumpfen fördert. Ebenso ist er dafür verantwortlich, dass sich Knoten in jenen Teilen der *snake* bündeln, wo andere Energieterme klein sind, z.B. wo die Kante im Bild stark ausgeprägt ist.

Sie formulieren einen alternativen Energieterm, mit dem versucht wird, die Knoten immer gleichabständig zu halten. Der *continuity* Term von Williams und Shah E_{cont}^{WS}

beschreibt die Abweichung des Abstands zweier Knoten vom mittleren Abstand zweier Knoten:

$$E_{cont}^{WS} = \bar{d} - |\mathbf{v}_i - \mathbf{v}_{i-1}| \tag{4-8}$$

Dazu muss vor jeder Iteration der mittlere Knotenabstand $\bar{d}$ bestimmt werden.

Ein Nachteil eines *continuity* Terms (4-8) soll anhand eines Beispiels demonstriert werden. Im Vergleich zu anderen Kantenextraktionsverfahren gilt als eine der Stärken von *snakes* die Fähigkeit Lücken zu überbrücken. In Gebieten, in denen die Kante einen schwachen Kontrast hat, scheitert die *low-level* Kantenextraktion. Zwar haben dort auch die photometrischen Energieterme der *snake* hohe Werte, doch die interne Energie sollte die *snake* befähigen solche Gebiete in glatter Weise zu überbrücken.

Abb. 4-2 zeigt ein synthetisches Beispiel. Das Bild repräsentiert die photometrische Energie. Hohe Energiewerte sind in Schwarz dargestellt, niedrige Energiewerte gegen Weiß. Eine horizontale gerade Linie niedriger Energie verläuft durch das Bild von links nach rechts mit einer Lücke dazwischen. Diese Linie hat einen breiten „Anziehungsbereich" mit kontinuierlich abfallender Energie, der - wie wir später sehen werden (Abschnitt 4.2) - essenziell für eine gute Konvergenz der *snake* ist. Dasselbe gilt für die Lücke in der Linie: die Energiewerte sinken kontinuierlich.

In diesem Bild ist ein Beispiel einer *snake* dargestellt. Der mittlere Abstand zwischen den Knoten ist ein wenig kleiner als sechs Pixel. Nehmen wir an, die *snake* habe zwei Energieterme: einen *continuity* Term wie in (4-8) definiert und einen photometrischen Term (Abschnitt 4.2). Der *continuity* Term versucht, alle Knoten gleichabständig zu halten. Die photometrische Energiekomponente lässt folgende Kräfte auf die Knoten wirken: die Knoten 5 bis 7 werden nach links gezogen, 9 bis 11 werden nach rechts gezögen und der Knoten 8 kann entweder nach links oder nach rechts wegwandern um seine photometrische Energie zu verringern.

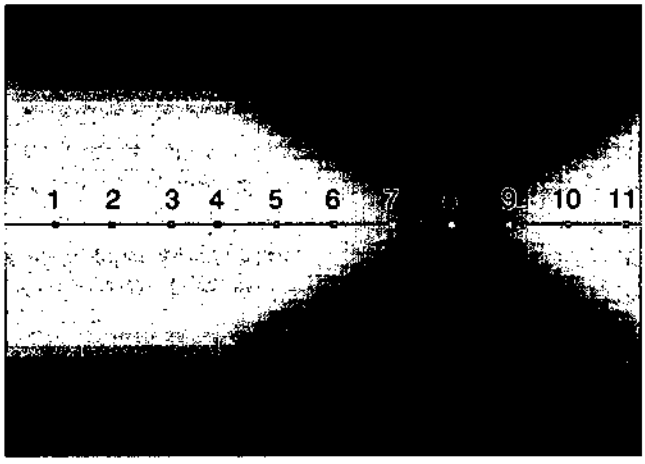


Abb. 4-2: Ein gerades *snake*-Segment überbrückt eine Lücke im Bild der photometrischen Energie (weiß entspricht niedriger Energie)

Das Verhalten der *snake* in der Phase der Energieminimierung hängt stark von der relativen Gewichtung der beiden Energieterme zueinander ab. Solange in diesem Beispiel das Gewicht des *continuity* Terms hoch genug ist, kann die Gesamtenergie der *snake* nicht verringert werden. Würde man nämlich einen Knoten verschieben,

würde der Gewinn durch einen niedrigeren photometrischen Energieanteil kleiner sein, als der Verlust durch einen höheren *continuity* Anteil.

Nehmen wir nun an, die photometrische Energie sei hoch genug gewichtet, dass die Verringerung der photometrischen Energie die Steigerung der *continuity* Energie überwiegen würde. Abb. 4-3 zeigt die Entwicklung der *snake* für diesen Fall. Der Status nach jeder Iteration ist dargestellt. Wie erwartet verschieben sich die Knoten 5 bis 7, aber auch der Knoten 8 nach links. Später wandern dann die Knoten 9 bis 11 nach rechts.

Bereits nach der ersten Iteration wird das Problem mit der Formulierung der *continuity* Energie nach Williams und Shah (4-8) klar. Die Knoten 5 bis 8 bewegen sich nach links auf Grund der starken Reduktion photometrischer Energie. Um die niedrige *continuity* Energie beizubehalten, wandern auch die Knoten 4 und 3 nach links. Dabei machen sie zusätzlich einen Schritt „zur Seite". Dieses Verhalten kann auch physikalisch interpretiert werden: Wenn das gerade *snake* Segment zusammengepresst wird und dabei aber seine Länge behalten soll, wird es anfangen abzuknicken, wie man aus dem Zahlenbeispiel in Abb. 4-4 erkennt.

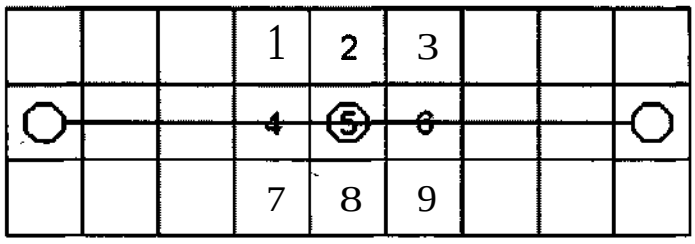


Abb. 4-4: Zahlenbeispiel für Energiewerte nach Williams und Shah: Drei Knoten einer *snake* mit Abstand 4 Pixel, wobei der Sollabstand zweier Knoten 5 Pixel betrage. Anfangs- und Endpunkt der *snake* seien fixiert. Falls der mittlere Knoten auf die jeweilige Position wandert, beträgt die Summe der beiden Energiewerte nach (4-8):

1, 3, 7, 9: (5 - 3.162) + (5 - 5.099) = 1.739
2, 8: (5 – 4.123) + (5 – 4.123) = 1.754
4, 6: (5 - 3.000) + (5 - 5.000) = 2.000
5: (5 - 4.000) + (5 - 4.000) = 2.000

Der minimale Energieanteil wird folglich durch einen diagonalen Schritt und durch ein Knicken der Gerade erreicht.

Die *continuity* Energie nach Williams und Shah bewirkt dadurch sogar ein Verhalten, das dem ursprünglichen Ziel (für einen glatten Kurvenverlauf zu sorgen) entgegengesetzt ist! Ausgerechnet an jenen Stellen, wo die *snake* durch geringe photometrische Energie gut definiert sein sollte, wird sie zackig. Von der Verwendung von Energietermen in der Art von (4-8) kann ich daher nur abraten.

Selbstverständlich kann dieses Verhalten unterdrückt werden, wenn man einen *curvature* Term in die Energiefunktion aufnimmt, wie von Williams und Shah auch vorgesehen ist. Jedoch ist bereits das Bestimmen passender Gewichte eine heikle Aufgabe. Kleine Änderungen der relativen Gewichtung der Energieterme kann die Entwicklung der *snake* stark beeinflussen.

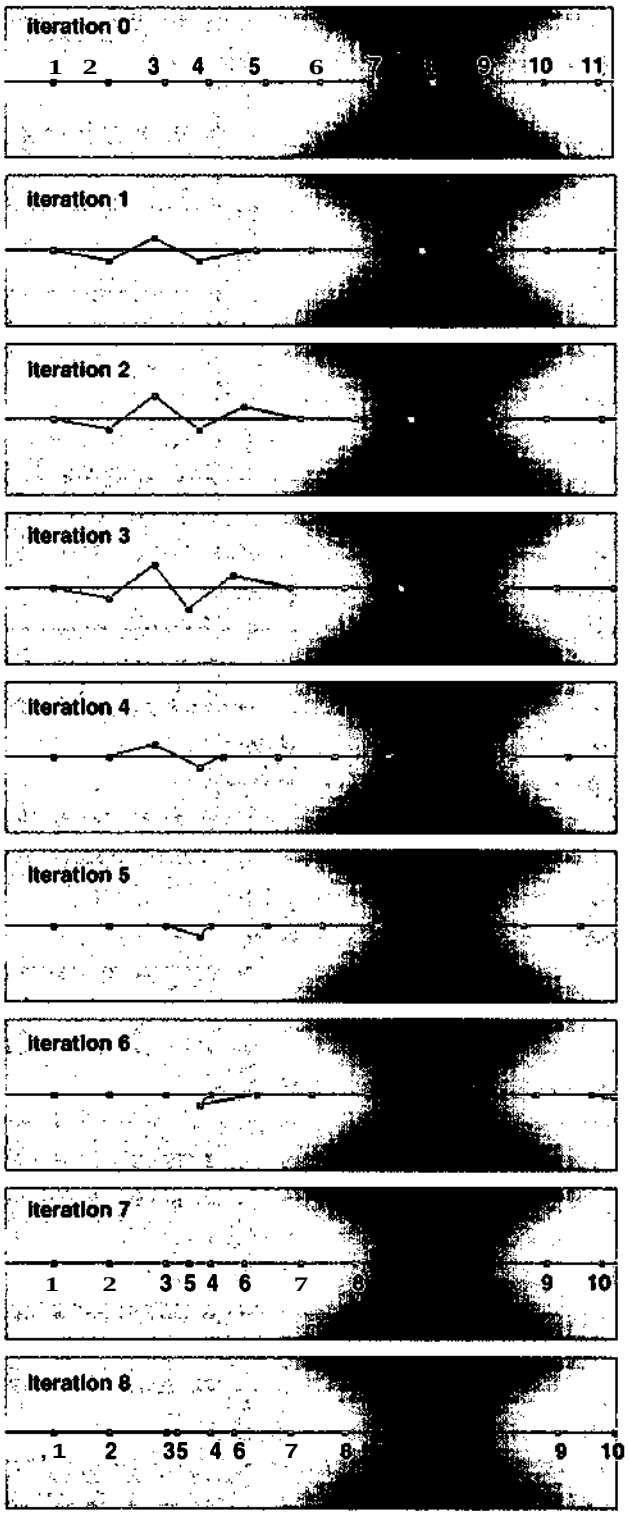


Abb. 4-3: Entwicklung der *snake* mit hoch gewichteten **photometrischen** Termen und niedrig gewichteten *continuity* Termen E^{WS}_{cont} (und keinen *curvature* Termen).

Wenn wir wieder Abb. 4-3 betrachten, erkennen wir weitere Schwierigkeiten als Folge des oben genannten Problems. Nach der sechsten Iteration wird der Knickwinkel des Polygonzugs an Knoten 5 zu einem spitzen Winkel. Eine Iteration später ist Knoten 5 zwar auf die gerade Linie minimaler photometrischer Energie zurückgekehrt, liegt aber zwischen den Knoten 3 und 4. Die *snake* sieht zwar glatt aus (in unserem Beispiel scheint sie sogar eine Gerade zu sein), allerdings ist die Reihenfolge der Knoten vertauscht.

Empirische Studien haben ergeben, dass das letztere Problem bei jenen *snakes* öfter auftritt, deren Knoten knapp beieinander liegen. Verwendet man den *continuity* Term von Williams und Shah, sollten die Knoten meinen Studien zufolge mindestens fünf Pixel voneinander entfernt initialisiert werden.

Zusammenfassend: Der *continuity* Term wie er von Williams und Shah vorgeschlagen wird um die Länge der *snake* von einer Iteration zur nächsten beizubehalten, kann gekrümmte *snakes* verursachen, falls

- andere Energieterme (z.B. die photometrische Energie) mit hohen Gewichten Knoten nahe zusammenrücken lassen und
- die *curvature* Energie ein zu geringes Gewicht hat um das Abknicken zu verhindern.

Übrigens, falls Knoten durch den Einfluss anderer Energieterme voneinander weg bewegt werden, steigt zwar der Energiewert des *continuity* Term von Williams und Shah, doch dies hat keine unbeabsichtigten Seiteneffekte auf die Form der *snake.*

4.1.6 Weglassen des *Continuity* Terms

Tests haben ergeben, dass der *continuity* Term kaum dazu beiträgt, das Ziel einer glatten *snake* zu erreichen. Da er hauptsächlich für das Schrumpfen der *snake* verantwortlich ist, empfehle ich, keinen *continuity* Term in der Energiefunktion zu verwenden.

4.1.7 Alternativer *Curvature* Term

Auch wenn kein *continuity* Term angesetzt wird und ausschließlich ein *curvature* Term laut (4-7) verwendet wird, gibt es Situationen, in denen die *snake* dennoch zum Schrumpfen neigt. Dieser Umstand wurde in veröffentlichten Aufsätzen noch nicht erkannt und soll im Folgenden begründet werden. Die Ursache liegt in der unzureichenden Approximation der Krümmung durch finite Differenzen zweiter Ordnung.

Zur Herleitung des Beweises gehen wir von der in Abb. 4-5 dargestellten Situation aus. Der *curvature* Term soll für den mittleren Knoten $\mathbf{v}_i$ bestimmt werden. Die Koordinaten der drei Knoten können wie folgt angesetzt werden:

$$\mathbf{v}_{i-1} = \begin{pmatrix} x_i - s\cos\nu \\ y_i - s\sin\nu \end{pmatrix}$$

$$\mathbf{v}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \qquad (4\text{-}9)$$

$$\mathbf{v}_{i+1} = \begin{pmatrix} x_i + ks\cos(\nu+\delta\nu) \\ y_i + ks\sin(\nu+\delta\nu) \end{pmatrix}$$

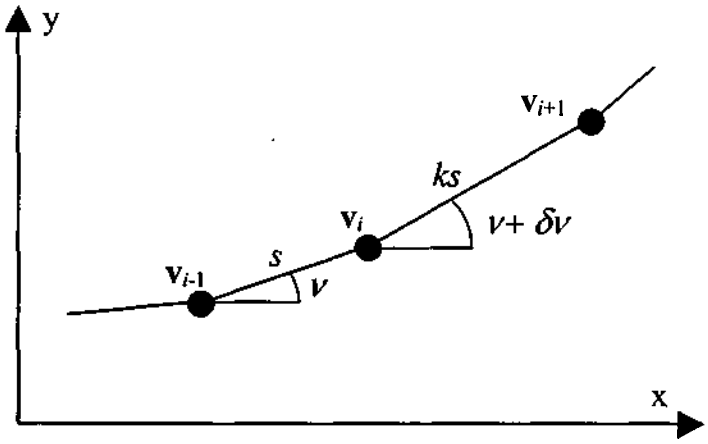


Abb 4-5: Drei aufeinanderfolgende Knoten einer *snake*.

Bildet man die finite Differenz zweiter Ordnung anhand von (4-7), fallen die Koordinaten des Knotens $(x_i, y_i)^T$ heraus und es ergibt sich:

$$|\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}| = \left|\begin{pmatrix} x_i - s\cos\nu - 2x_i + x_i + ks\cos(\nu+\delta\nu) \\ y_i - s\sin\nu - 2y_i + y_i + ks\sin(\nu+\delta\nu) \end{pmatrix}\right| = \left|\begin{pmatrix} -s\cos\nu + ks\cos(\nu+\delta\nu) \\ -s\sin\nu + ks\sin(\nu+\delta\nu) \end{pmatrix}\right|$$

Bildet man die quadrierte Norm des Vektors, erhält man den *curvature* Term wie in (4-7) definiert. Wendet man Summensätze der Winkelfunktionen an, vereinfacht sich der Ausdruck:

$$\begin{aligned}
&E_{curv} = |\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}|^2 = \\
&s^2(\cos^2\nu + \sin^2\nu) - \\
&2ks^2(\cos\nu\cos(\nu+\delta\nu) + \sin\nu\sin(\nu+\delta\nu)) + \\
&k^2s^2(\cos^2(\nu+\delta\nu) + \sin^2(\nu+\delta\nu)) = \\
&s^2 - 2ks^2\cos(\nu + \delta\nu - \nu) + k^2s^2
\end{aligned}$$

Zieht man s^2 heraus, erhält man schließlich eine einfache Darstellung der *curvature* Energie in Abhängigkeit von anschaulichen Größen *s*, $\delta\nu$ und *k*.

$$E_{curv} = s^2(1 - 2k\cos\delta\nu + k^2) \qquad (4\text{-}10)$$

Unter der Voraussetzung $s > 0$ verschwindet demnach der *curvature* Term genau dann, wenn $\delta\nu = 0$ und $k = 1$, d.h. wenn die drei Knoten eine Gerade bilden und der Abstand zwischen den Knoten gleich ist.

Ist diese Bedingung nicht erfüllt (d.h. *8v* ≠ 0 oder $k \neq 1$) und behält man die geometrische Konstellation der drei Knoten bei (d.h. *öv* und *k* werden nicht variiert), sinkt der Energiewert mit dem Quadrat der Entfernung der Knoten! In anderen Worten: Ist der *curvature* Term größer als Null, besteht eine starke Tendenz zum Schrumpfen der *snake,* q. e. d..

Dies lässt sich an dem Achteck (Beispiel aus Abb. 4-1 links) zeigen. Das regelmäßige Achteck hat von allen geschlossenen Kurven mit acht Knoten die niedrigste *curvature* Energie. In jedem der Knoten muss ein Knick von *öv* = *K l* 4 realisiert werden, d.h. die *curvature* Energie kann nicht verschwinden. Wie allerdings eben

hergeleitet, kann die *snake* ihre Energie dadurch verringern, dass sie die Polygonseiten zwischen den Knoten verkürzt. Dies führt zu einem fortschreitenden Schrumpfen des Achtecks, bis es schließlich in einem einzigen Punkt degeneriert und die *curvature* Energie schließlich doch verschwindet. Die Entwicklung der *snake* ist dieselbe wie in Abb. 4-1 dargestellt auch wenn nur der *curvature* Term nach (4-7) verwendet wird.

Alternativer Vorschlag für die *curvature* Energie:

Im Folgenden wird eine neue Formulierung der *curvature* Energie präsentiert, die das Schrumpfen der *snake* verhindern soll:

$$E_{curv} = \left| \frac{\mathbf{v}_{i-1} - \mathbf{v}_i}{|\mathbf{v}_{i-1} - \mathbf{v}_i|} - \frac{\mathbf{v}_i - \mathbf{v}_{i+1}}{|\mathbf{v}_i - \mathbf{v}_{i+1}|} \right|^2 \tag{4-11}$$

Diese Formulierung verwendet statt des Quadrats der finiten Differenz zweiter Ordnung (vgl. (4-7)) das Quadrat der finiten Differenz zweier normierter finiter Differenzen erster Ordnung. Analog zum oben geführten Beweis soll gezeigt werden, dass die neu formulierte *curvature* Energie unabhängig von der Seitenlänge des Polygons ist. Schreibt man (4-11) mit den in (4-9) bzw. Abb. 4-5 definierten Variablen an, erhält man:

$$E_{curv} = \left| \frac{\mathbf{v}_{i-1} - \mathbf{v}_i}{|\mathbf{v}_{i-1} - \mathbf{v}_i|} - \frac{\mathbf{v}_i - \mathbf{v}_{i+1}}{|\mathbf{v}_i - \mathbf{v}_{i+1}|} \right|^2 = \left| \frac{\mathbf{v}_{i-1} - \mathbf{v}_i}{s} - \frac{\mathbf{v}_i - \mathbf{v}_{i+1}}{ks} \right|^2 =$$

$$\left| \begin{pmatrix} \frac{x_i - s\cos\nu - x_i}{s} - \frac{x_i - x_i - k\cos(\nu + \delta\nu)}{ks} \\ \frac{y_i - s\sin\nu - y_i}{s} - \frac{y_i - y_i - ks\sin(\nu + \delta\nu)}{ks} \end{pmatrix} \right|^2 = \left| \begin{pmatrix} -\cos\nu + \cos(\nu + \delta\nu) \\ -\sin\nu + \sin(\nu + \delta\nu) \end{pmatrix} \right|^2$$

Nachdem *s* und *k* durch Kürzen weggefallen ist, ergibt sich:

$$\begin{aligned} E_{curv} &= (\cos^2\nu + \sin^2\nu) - \\ &2(\cos\nu\cos(\nu + \delta\nu) + \sin\nu\sin(\nu + \delta\nu)) + \\ &(\cos^2(\nu + \delta\nu) + \sin^2(\nu + \delta\nu)) = \\ &1 - 2\cos(\nu + \delta\nu - \nu) + 1 \end{aligned}$$

Durch Zusammenfassen ergibt sich der Wert der *curvature* Energie:

$$E_{curv} = 2(1 - \cos\delta\nu) \tag{4-12}$$

Der Energiewert ist demnach nur noch vom Winkel *öv* abhängig. Vergleicht man (4-12) mit (4-10), erkennt man weiters, dass auch die Forderung nach gleichabständigen Knoten zum Erreichen niedriger Energiewerte wegfällt. Es muss allerdings Vorsorge getroffen werden, dass der Abstand zweier aufeinander folgender Knoten größer als Null ist. Falls Knoten zusammenfallen, ergibt sich in (4-11) eine Division durch Null. Daher muss der überflüssige Knoten aus der *snake* entfernt werden.

Am Ende dieses Abschnittes soll zusammengefasst werden, wie bei den in dieser Dissertation entwickelten *snakes* das Schrumpfen unterbunden wird:

- Der *continuity* Term der ursprünglichen Formulierung der Energie von *snakes* (4-7) wurde als hauptverantwortlich für das Schrumpfen erkannt und wird aus der Energiefunktion gestrichen.
- Auch die alternative Formulierung eines *continuity* Terms nach (4-8) hat gravierende Nachteile und wird deshalb nicht verwendet.
- Der *curvature* Term in der ursprünglichen Formulierung von (4-7) kann unter bestimmten Umständen ebenfalls zum Schrumpfen der *snake* führen. Als Alternative wird als einziger interner Energieterm ein *curvature* Term in der Form von (4-11) verwendet.

Führt man dies für das Beispiel aus Abb. 4-1 durch, ergibt sich für das Achteck, dass die Ausgangslage nicht mehr verbesserbar ist (was für ein regelmäßiges Achteck zu erwarten ist). Die Zickzacklinie des rechten Beispiels wird in acht Iterationen (mit Bewegungsfreiheit der Knoten in einer 8-er Nachbarschaft) zu einer Geraden optimiert (vgl. Abb. 4-6).

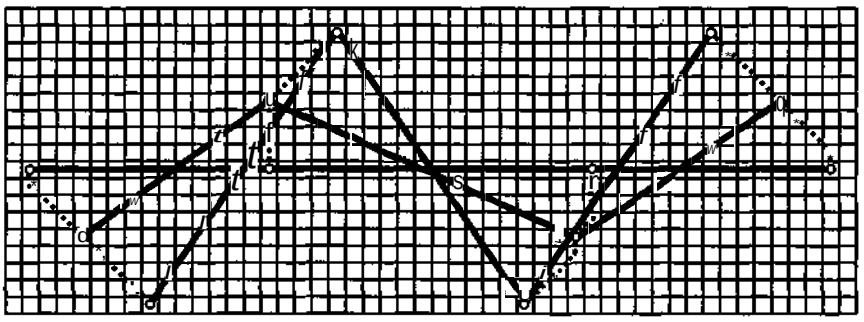


Abb. 4-6: Das rechte Beispiel aus Abb. 4-1 wurde neu optimiert mit einem *curvature* Term nach Gleichung (4-11). Man erkennt deutlich, dass der Term eine Streckung der Zickzacklinie zu einer Geraden bewirkt ohne sie zu verkürzen.

Diese Formulierung der internen Energie soll noch nach unterschiedlichen Gesichtspunkten bewertet werden:

- Rechenzeit: Die Berechnung der *curvature* Energie nach (4-11) ist ohne Zweifel aufwändiger und beeinträchtigt daher die Laufzeit einer *snake*-Optimierung. Dies wird zum Teil dadurch kompensiert, dass kein *continuity* Term berechnet werden muss. Durch die ständige Steigerung der Leistung moderner Rechner verliert dieser Nachteil allerdings mehr und mehr an Bedeutung.
- Gleichabständigkeit der Knoten: Die Knoten der *snake* brauchen nicht gleichabständig entlang der Kurve verteilt zu sein. Dies hat den Vorteil, dass eine *snake* Lücken in Bildkanten überbrücken kann ohne in den Bereichen hoher photometrischer Energie Knoten setzen zu müssen. Obwohl andere Autoren (Amini et al., 1988 oder Williams und Shah, 1992) es als Nachteil bezeichnen, dass sich Knoten in Bereichen niedriger photometrischer Energie bündeln,

möchte ich dies durchaus als Stärke von *snakes* bezeichnen. Es muss lediglich gewährleistet sein, dass die Energieberechnung auch für ungleich verteilte Knoten gültige Werte liefert.

4.2 Photometrische Energieterme

Photometrische **Terme** haben die Aufgabe, die *snake* zu den gesuchten **linienhaften** Bildobjekten (Kanten, Linien, etc.) zu ziehen. Dafür ist eine **Skalarfunktion** (Potenzial) zuständig, die aus dem Bild abgeleitet wird. Sie liefert für jedes Pixel des Bildes einen Energiewert.

Im ursprünglichen *snake*-**Ansatz** (Kass et **al., 1987)** sind wahlweise zwei verschiedene photometrische Terme vorgesehen. Die einfachste Bildenergie ist die Bildfunktion selbst:

$$E_{pho}^{line} = I(x,y) \tag{4-13}$$

Durch diesen simplen Energieterm erreicht die *snake* an dünnen, dunklen Linien auf hellem Hintergrund minimale Energie. Durch ein negatives Vorzeichen in (4-13) kann die *snake* zur Extraktion heller Linien verwendet werden.

Auch Kanten können mit einem einfachen Energieterm gefunden werden:

$$E_{pho}^{edge} = -\left|\nabla I(x,y)\right|^2 \tag{4-14}$$

Durch diesen Term wird die photometrische Energie an jenen Stellen im Bild niedrig, an denen der Gradient der Bildfunktion stark ist. Dies ist typischerweise an Bildkanten der Fall.

Das größte Problem in der Formulierung der photometrischen Energie besteht darin, dass die *snake* von der Zielkurve oft nicht angezogen wird, wenn sie zu weit entfernt ist. Zwar ist die photometrische Energie bei Erreichen der Zielkurve minimal, doch ist sie in homogenen Bildbereichen nahezu konstant. Damit sinkt sie nicht ausreichend bei Annäherung von der Ferne. Stellt man sich das Energiepotenzial als Gebirge vor, bilden die zu extrahierenden Kurven enge Schluchten niedriger Energie innerhalb von Plateaus hoher Energie, begrenzt von steilen Wänden. Wünschenswert wären breite, von flachen Flanken begrenzte Täler niedriger Energie, damit sich die *snake* durch kontinuierliches „Absteigen" annähern kann.

Damit die *snake* eine Kurve auch dann rekonstruieren kann, wenn kein breiter Anziehungsbereich vorhanden ist, muss sich zumindest ein Teil der *snake* mit der Zielkurve decken. Dann sorgen im Idealfall die internen Energieterme dafür, auch die benachbarten Teile der *snake zur* möglichen Fortsetzung der Zielkurve zu ziehen.

Die Konvergenz des Verfahrens hängt demnach stark von günstigen Startwerten ab. Die Forderung, dass sich zumindest ein Teil der *snake* mit der Zielkurve decken muss, damit das Verfahren konvergiert, stellt eine starke Einschränkung für die

Ausgangslage der *snake* dar. Folglich führen verschiedene Startpositionen zu unterschiedlichen Ergebnissen.

Die Kantenextraktion mittels *snakes* war als robustes Verfahren konzipiert, das relativ unabhängig vom Rauschen oder anderen Artefakten im Bild arbeiten sollte. Eine Forderung für robuste Verfahren lautet allerdings auch, dass geringe Änderungen in den Parametern und Startwerten keine Auswirkung auf das Ergebnis haben dürfen. Diese Forderung ist nicht erfüllt, wenn eine geringe Änderung der Ausgangslage über Erfolg oder Misserfolg der Methode entscheidet.

Mehrere Autoren versuchten daher, das Problem zu lösen. Während bei vielen Verbesserungsvorschlägen die photometrische Energie ein unveränderliches Energiepotenzial bleibt, wird vielfach auch versucht, die photometrische Energie im Laufe der Annäherung der *snake* an die Zielkurve anzupassen oder auszuwechseln.

Im Folgenden werden zwei Möglichkeiten vorgestellt, wie ein Konvergenzbereich rund um die Kurve geschaffen werden kann (Glättung oder Distanztransformation). Anschließend werden noch andere Vorschläge präsentiert, wie die photometrische Energie abgewandelt werden kann um das *snake*-Modell robuster zu machen.

4.2.1 Glättung

Um den Anziehungsbereich zu vergrößern, wird vielfach das Gradientenbild mit einem Gauß-Filter geglättet:

$$E_{pho}^{edge} = -\left(G_\sigma(x,y) * |\nabla I(x,y)|\right)^2 \tag{4-15}$$

Der Gauß-Filter ist ein Glättungsfilter und besitzt die Wirkung eines Tiefpassfilters, d.h. die hohen Bildfrequenzen werden geschwächt. Zusätzlich zur Abschwächung der Amplitude des Signals führt der Gauß-Filter aber auch zur Verringerung der Frequenz eines hochfrequenten Signals, was zur erwünschten Verbreiterung des Anziehungsbereiches führt. Die Breite dieses Bereichs kann über das *a* der Gauß-Funktion gesteuert werden (vgl. Abb. 4-7):

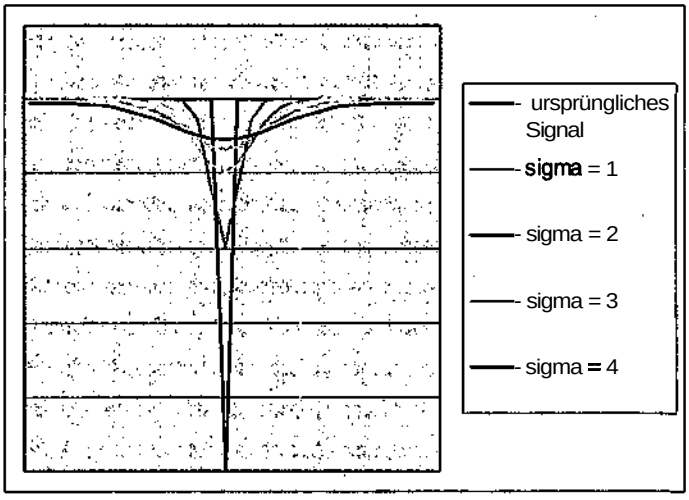


Abb. 4-7: Verbreiterung des Anziehungsbereichs eines Signals bei gleichzeitiger Abnahme seiner Stärke durch Gauß'sche Filterung mit unterschiedlichen Werten für *a*.

$$G_\sigma(r) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{r^2}{2\sigma^2}}$$

Darin ist *r* der Abstand der Elemente der Faltungsmatrix vom zentralen Pixel:

$$r = \sqrt{\delta x^2 + \delta y^2}$$

In Abb. 4-7 erkennt man aber deutlich, dass bei Verbreiterung des Anziehungsbereichs mittels Gauß'scher Glättung die Stärke des Signals drastisch abnimmt. Es können daher nur mehr überaus markante Bildkanten von *snakes* gefunden werden. Solche stark ausgeprägte Kanten würden aber auch problemlos von einem anderen Kantenextraktionsalgorithmus (z.B. Canny, 1986) detektiert werden. Daher bringt der Ansatz über starke Gauß'sche Glättung nicht die gewünschte Überlegenheit von *snakes* gegenüber herkömmlicher Kantenextraktion.

Kass et al. (1987) experimentierten auch mit einer Glättung des Energiebildes unter Verwendung der Kantenextraktionstheorie von Marr und Hildreth (1980), bei der Nulldurchgänge *(zero crossings)* in der zweiten Ableitung gesucht werden. Dazu wird der folgende Energieterm angesetzt[9]:

$$E_{pho}^{LoG} = \left(G_\sigma * \nabla^2 I\right)^2 \tag{4-16}$$

Das Symbol ∇^2 wird meist als Laplace-Operator A bezeichnet. Die Berechnung des Energiebilds nach (4-16) kann effizienter erfolgen: Anstatt zuerst den Laplace-Operator auf das Bild anzuwenden und das Ergebnis mit einem Gauß'schen Filter zu falten, kann der Operator zuerst auf den Gauß'schen Filter angewendet werden und anschließend das Bild mit dem kombinierten Filter, dem so genannten *LoG*-Operator (*Laplacian of Gaussian),* gefaltet werden.

Der Energieterm von (4-16) nimmt allerdings nicht nur bei Nulldurchgängen minimale Werte an, sondern auch für homogene Regionen mit kaum variierenden Grau-/Farbwerten, weil dort kaum eine Krümmung in der Bildfunktion auftritt. Außerdem hat die Bestimmung von Kanten (Extrema der ersten Ableitung) durch die Detektion von Nulldurchgängen der zweiten Ableitung den Nachteil, dass die aus diskreten Ableitungen geschätzten höheren Ableitungen unzuverlässiger werden (vgl. Anhang A).

Kass et al. wendeten den *LoG* in einem *scale-space* Verfahren an. Zuerst wird die *snake* in einem unscharfen, mit großem σ geglätteten Energiebild detektiert. Zur nachfolgenden genauen Extraktion wird die Schärfe durch Verkleinern des *a* kontinuierlich erhöht.

Ein dem *scale-space* Verfahren ähnlicher, hierarchischer Ansatz mit diskreten „*scale*"-Stufen wird in Abschnitt 5.1 vorgestellt.

4.2.2 Distanztransformation

Breitere Anziehungsbereiche können erzielt werden, wenn als photometrische Energie in jedem Pixel die Distanz zum nächsten Kanten- oder Linienpixel verwendet wird (Cohen, Cohen, 1993 oder Trinder, Li, 1996). Dazu wird das Ergebnis einer vorher-

---

[9] In Kass et al. (1987 und 1988) ist dieser Term irrtümlich mit einem negativen Vorzeichen versehen.

gehenden *low-level* Kantenextraktion verwendet (z.B. Canny, 1986). Gülch (1996) verwendet das Ergebnis des Förstner-Operators. Für die Extraktion heller dünner Linien lässt sich auch ein auf Straßenextraktion spezialisierter Operator verwenden (z.B. Fischler et al., 1981).

Unter Verwendung der Klassifizierung in Kanten- und Nicht-Kantenpixel wird die photometrische Energie definiert:

$$E_{pho} = d(x, y) \tag{4-17}$$

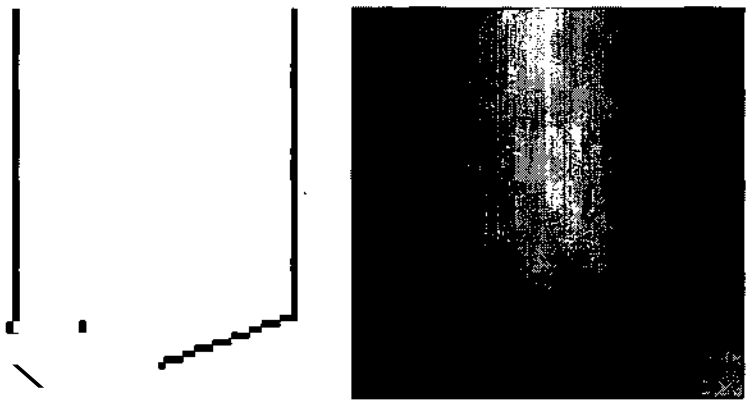


Abb. 4-8: Links: Binärbild mit Kantenpixeln. Rechts: Zugehöriges Distanzbild (je größer die Distanz zu einem Kantenpixel, desto heller ist das Pixel).

Dabei ist $d(x,y)$ die euklidische Distanz des Pixels an der Stelle *(x,y)* zum nächstgelegenen Kanten- bzw. Linienpixel. Das Energiebild entspricht einem Distanzbild *(distance map).* Meist werden Approximationen der euklidischen Distanz verwendet, für deren Ableitung effiziente Algorithmen *(distance transformation, chamfering)* bestehen (Borgefors, 1986). In Abb. 4-8 sind ein Binärbild mit Kantenpixeln und das zugehörige Distanzbild dargestellt. Abb. 4-9 zeigt den Vergleich der Gradienten der photometrischen Energie vor und nach einer Distanztransformation.

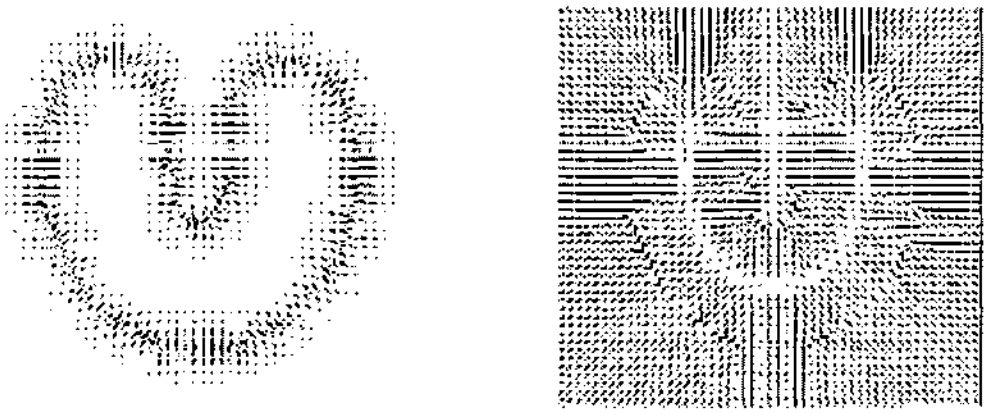


Abb. 4-9: Verbreiterung des Anziehungsbereichs einer *snake* durch Verwendung einer Distanztransformation. Dargestellt ist in beiden Fällen der Gradient der photometrischen Energie. Die Länge der Vektoren entspricht dem Betrag der Gradienten (aus Xu, Prince, 1998).

Als Nachteil dieser Vorgangsweise soll angemerkt werden, dass zur Herleitung der photometrischen Energie bereits Entscheidungen getroffen werden müssen. Bei jedem Kantenextraktionsalgorithmus werden Pixel über einen (oder mehrere) Schwellwert(e) in Kantenpixel oder Nicht-Kantenpixel klassifiziert. Erst nach dieser Klassifikation wird die Robustheit der *snakes* ausgenutzt um vereinzelte Fehlklassifikationen zu ignorieren und Lücken in der Kante zu überbrücken. Allerdings können an dieser Stelle fälschlicherweise als Nicht-Kantenpixel klassifizierte Pixel nicht mehr berücksichtigt werden. Falls der (die) Schwellwert(e) schlecht gewählt wurde(n), kann es passieren, dass schwach ausgeprägte Kanten gänzlich ausgelöscht werden. Daher schlagen Cohen, Cohen (1993) vor, die Energiewerte nach (4-17) redundant in Kombination mit dem Gradienten der Bildfunktion einzusetzen.

4.2.3 Einbeziehung der Orientierung des Gradienten:

Bei Verwendung von photometrischen Energietermen wie (4-14) oder (4-15) wird als Kriterium für Kanten der Gradient der Bildfunktion verwendet. Der Gradient der Bildfunktion hat folgende Eigenschaften:

- Der Absolutbetrag des Gradienten ist in der Nähe einer Bildkante groß.
- Die Richtung des Gradienten steht normal auf die Richtung der Bildkante.
- Der Betrag des Gradienten sinkt schnell entlang seiner Richtung (normal zur Bildkante).
- Der Gradient ist nahe Null in homogenen Bildbereichen.

Während die erste dieser Eigenschaften für die Konvergenz von *snakes* notwendig ist, verursachen die dritte und vierte Eigenschaft wie erwähnt einen schmalen Konvergenzbereich. Die zweite Eigenschaft wurde bisher allerdings überhaupt noch nicht berücksichtigt. Die Richtung des Gradienten kann durchaus zur Steigerung der Robustheit von *snakes* verwendet werden.

Während der Energieminimierung wird im Allgemeinen der Zusammenhang der photometrischen Energie benachbarter Knoten der *snake* nicht berücksichtigt. Es wird nicht überprüft, ob benachbarte Knoten zur selben Zielkurve gezogen werden oder ob verschiedene Teile der *snake* unterschiedliche Kurven detektieren (vgl. Abb. 4-10).

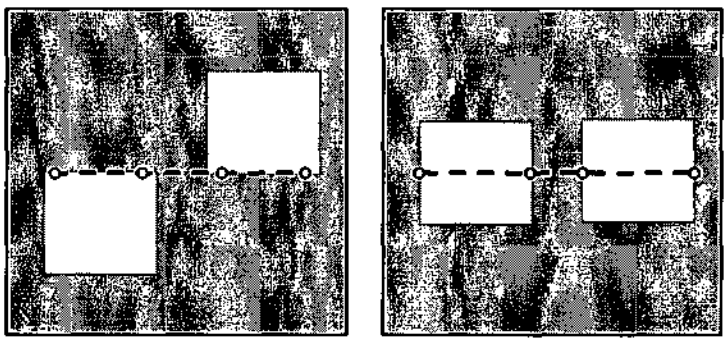


Abb. 4-10: Zwei scheinbar optimale *snakes:* minimale interne und photometrische Energie (jeweils eine Gerade mit Knoten, die alle an Stellen mit hohem Gradient der Grauwertfunktion liegen)

Bei einer schlechten Ausgangslage kann die *snake* zu einer sich teilweise überdeckenden Kurve degenerieren (vgl. Abb. 4-11). Damit erfüllt die Kurve das Jordan - Kurventheorem nicht mehr (Grzeszczuk und Levin, 1997): eine Kurve kann nur dann die Begrenzung einer zusammenhängenden Region sein, wenn sie keine Selbstüberschneidungen und auch keine deckungsgleichen Abschnitte aufweist.

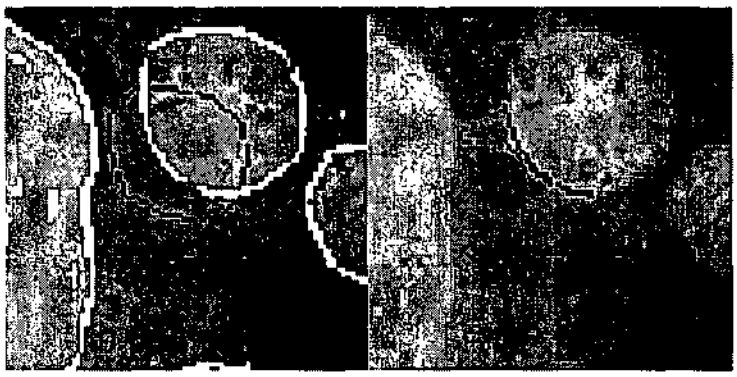


Abb. 4-11: Links: Ausgangslage der *snake* (schwarz) und Kantenkandidatenpixel (weiß). Rechts: Degenerierte *snake* nach etlichen Iterationen ohne Berücksichtigung der Orientierung des Gradienten (aus Radeva et al., 1995).

Daher sollte auch die Orientierung des Gradienten im photometrischen Energieterm berücksichtigt werden. Ein Ansatz wäre, die photometrische Energie nur dann klein zu setzen wenn die Orientierung des Gradienten ähnlich zur Orientierung der Normalen auf die *snake* verläuft (Radeva et al., 1995).

Einfacher lässt sich dies erreichen, wenn als photometrischer Term anstatt des Betrags des Gradienten die Projektion des Gradienten auf den normierten Normalvektor n der *snake* an der Stelle *s* verwendet wird (Mayer et al., 1998):

$$E_{pho}(\mathbf{v}(s)) = -(\nabla I(\mathbf{v}(s)) \cdot \mathbf{n}(s))^2 \tag{4-18}$$

Im Fall, dass der Gradient des Energiepotenzials und die Normale auf die *snake* an der Stelle *s* in dieselbe Richtung weisen, ist (4-18) äquivalent zu (4-14). Im Fall, dass die beiden normal aufeinander stehen, steigt der Energiewert auf Null.

Dieser Ansatz alleine löst allerdings nicht das linke Beispiel in Abb. 4-10. Um zu erreichen, dass die Orientierung des Gradienten entlang der *snake* nicht umspringt, muss das Vorzeichen des inneren Produkts in (4-18) berücksichtigt werden und die Orientierung des Normalvektors auf die *snake* immer in dieselbe Richtung weisen wie die Gradientenrichtung der Bildfunktion. Die photometrische Energie ergibt sich dann zu:

$$\begin{aligned} e &= \nabla I(\mathbf{v}(s)) \cdot \mathbf{n}(s) \\ E_{ph}o &= -\operatorname{sign}(e) e^2 \end{aligned} \tag{4-19}$$

Damit die Orientierungen der Gradienten und der Normalvektoren auf die *snake* übereinstimmen, muss der Operateur beim Festlegen der Ausgangslage der *snake* darauf achten, die Punkte in einer solchen Reihenfolge zu digitalisieren, dass die helle Seite der Bildkante links der Kurve und die dunkle Seite rechts der Kurve liegt (unter der Voraussetzung, dass der Normalvektor an die *snake* nach rechts weist und die Bildfunktion für helle Bereiche hohe Werte liefert).

Kann diese Konvention vom Operateur nicht gewährleistet werden, kann versucht werden den Umlaufsinn der *snake* automatisch zu ermitteln, damit die Bedingung erfüllt ist. Dazu könnte die *snake* die photometrische Energie entlang der gesamten Ausgangslage nach beiden Varianten berechnen und den Umlaufsinn so festlegen, dass sich die niedrigere Energie ergibt. Dies wird dann zum Erfolg führen, wenn die Ausgangslage die Zielkurve bereits gut annähert.

4.2.4 Interpretation des photometrischen Energiepotenzials als Vektorfeld *(Gradient Vector Flow Snakes)*

Die photometrischen Energiewerte für jedes Pixel stellen wie erwähnt eine Potenzialfunktion dar, also ein Skalarfeld. Aus einem Potenzial lässt sich ein Vektorfeld, das Gradientenfeld ableiten.

$$\mathbf{g}(x,y) = \nabla E_{pho}(x,y) = \begin{pmatrix} \dfrac{\partial E_{pho}(x,y)}{\partial x} \\ \dfrac{\partial E_{pho}(x,y)}{\partial y} \end{pmatrix} \tag{4-20}$$

Verwendet man zur Energieoptimierung anstatt des Gradientenfeldes ein modifiziertes Vektorfeld, lassen sich manche Probleme der *snakes* lösen (Xu, Prince, 1998). *Snakes,* die ein modifiziertes Vektorfeld als photometrische Kräfte verwenden, werden *GVF snakes (gradient vector flow snakes)* genannt.

Um die Optimierung mittels Variationsrechnung über ein Vektorfeld durchführen zu können, müssen wir das Energieoptimierungsproblem neu interpretieren. Wir gehen von Gleichung (3-4), der Euler-Lagrange'schen Differenzialgleichung des Abschnitts 3.1, aus. Ignoriert man externe Energieterme und setzt man:

$$\begin{aligned} \mathbf{F}_{int} &= \alpha\, \mathbf{v}_{ss} - \beta\, \mathbf{v}_{ssss} \\ \mathbf{F}_{pho} &= \frac{\partial E_{pho}}{\partial \mathbf{v}} \end{aligned} \tag{4-21}$$

kann die Gleichung (3-4) als Kräftegleichgewicht interpretiert werden:

$$\mathbf{F}_{int} = \mathbf{F}_{pho} \tag{4-22}$$

wobei die internen Kräfte gegenüber den photometrischen Kräften im Gleichgewicht stehen.

Die photometrischen Kräfte bilden ein dichtes Vektorfeld, das sich aus den Gradienten des Energiepotenzials ableiten lässt. Ein Gradientenfeld wird oft auch als konservatives Vektorfeld bezeichnet. Ein derartiges Feld ist wirbelfrei, d.h. $\mathrm{rot}\,\mathbf{g} = 0$, und quellfrei, d.h. $\mathrm{div}\,\mathrm{g} = 0$, mit:

$$\text{rot}\,\mathbf{g}(x,y) = \nabla \times \mathbf{g} = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \times \begin{pmatrix} g_1(x,y) \\ g_2(x,y) \end{pmatrix} = \frac{\partial g_2}{\partial x} - \frac{\partial g_1}{\partial y}$$

$$\text{div}\,\mathbf{g}(x,y) = \nabla \cdot \mathbf{g} = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \cdot \begin{pmatrix} g_1(x,y) \\ g_2(x,y) \end{pmatrix} = \frac{\partial g_1}{\partial x} + \frac{\partial g_2}{\partial y} \tag{4-23}$$

Darin sind g_1 und g_2 die Komponenten des Gradientenvektors.

Snakes sind bekanntlich ungeeignet konkave Einbuchtungen zu extrahieren (vgl. Abb. 4-12).

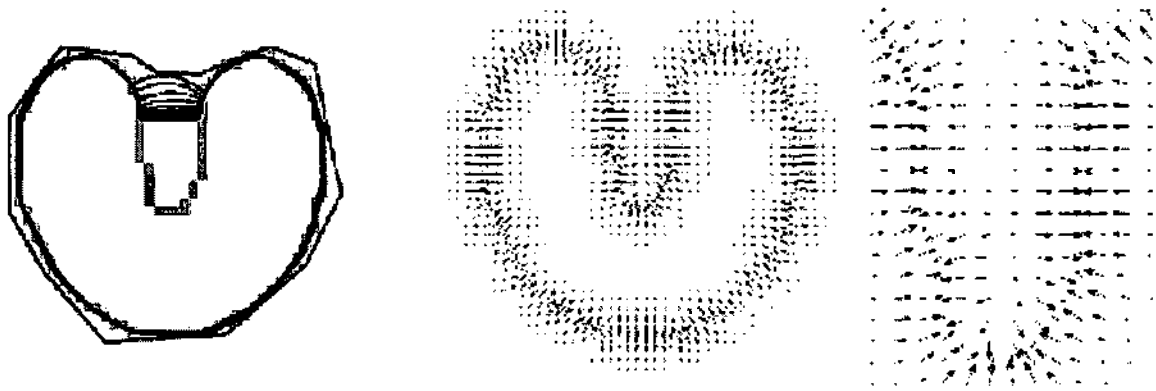


Abb. 4-12: Links: Die *snake* (schwarz) passt sich im Zuge der Energieminimierung der Zielkurve (grau) an. Lediglich die Einbuchtung wird durch eine gerade Linie überbrückt. Mitte: Das Gradientenfeld der photometrischen Energie lässt die Linie minimaler photometrischer Energie erkennen. Photometrische Energieterme versuchen im Zuge der Energieminimierung Knoten der *snake* entlang der dargestellten Vektoren in ein Minimum zu bewegen. Rechts: Im Detail erkennt man allerdings, warum die *snake* nicht weiter in die Einbuchtung gezogen wird. Die Richtungen der Gradienten sind in der Mitte der Einbuchtung horizontal. Daher gibt es keine Kräfte, die die *snake* nach unten ziehen würden (aus Xu, Prince, 1998).

Um das Verhalten der *snakes* bei der Extraktion von Einbuchtungen zu verbessern, schlagen Xu und Prince (1998) ein spezielles Vektorfeld $\mathbf{w}$ vor. Das Vektorfeld ist in Abb. 4-13 zu sehen. Dieses Vektorfeld ist nicht mehr wirbelfrei, allerdings noch quellfrei, d.h. $\text{rot}\,\mathrm{w} \neq 0$ aber $\text{div}\,\mathrm{w} = 0$.

Diese Definition eines Vektorfeldes verbreitert zudem noch den Anziehungsbereich der *snakes.* Damit hängen *GVF snakes* weniger von einer guten Ausgangslage ab als herkömmliche *snakes.*

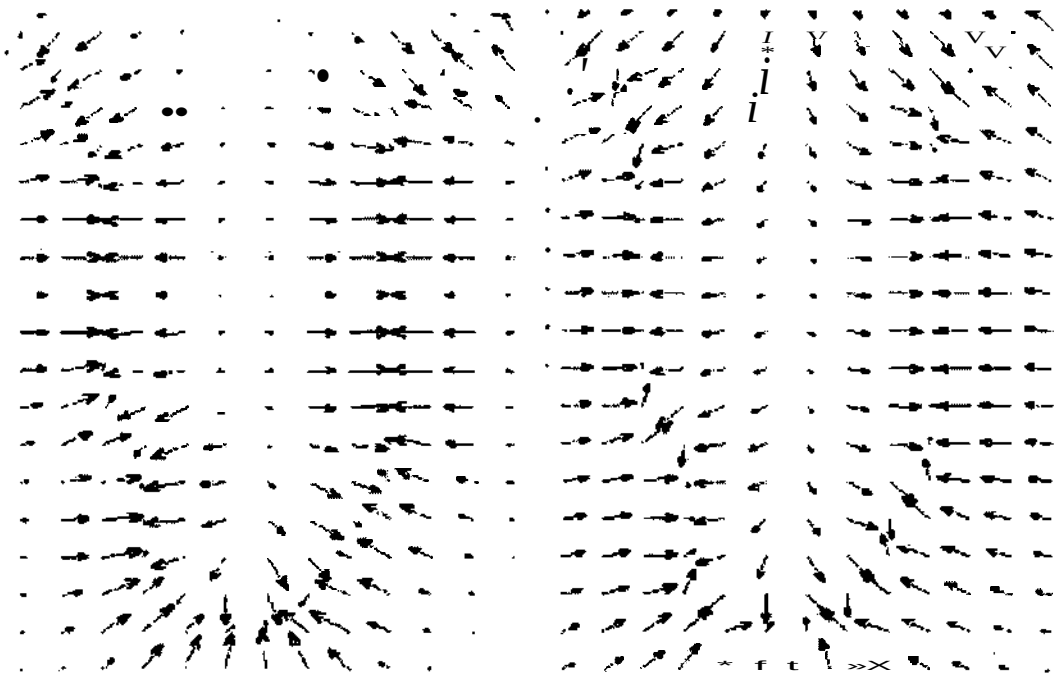


Abb. 4-13: Links: Dasselbe Detail des Vektorfelds wie in Abb. 4-12. Rechts: Das Vektorfeld der *GVF snakes.* Man erkennt, dass die Vektoren innerhalb der konkaven Einbuchtung leicht nach unten zeigen und es damit der *snake* ermöglichen, die Einbuchtung zu extrahieren (aus Xu, Prince, 1998).

Als Nachteil dieses Ansatzes soll die enorme Rechenzeit erwähnt werden (bei optimierter Implementierung etwa um den Faktor 7 langsamer, vgl. Xu, Prince, 1998). Weiter soll erwähnt werden, dass der Ansatz auf die Lösung mittels Variationsrechnung beschränkt ist (wobei die Integration in andere Optimierungsmethoden theoretisch möglich ist).

4.2.5 Einbeziehung der Farbinformation

Trotz der Vielfalt an Veröffentlichungen im Bereich der *snakes* wurde kaum eine Formulierung der photometrischen Energie publiziert, die mehr als nur einen Kanal der Farbbilder verwendet. Nahezu alle Ansätze leiten die photometrische Energie aus einem Kanal ab. Dabei wird meist der Intensitätskanal nach einer Transformation des Farbbilds vom RGB-Raum *(red I green I blue)* in den IHS-Raum (*intensityl hue I Saturation)* verwendet. Der Grund liegt vermutlich darin, dass der Informationsgehalt in Farbbildern nur unwesentlich höher ist als in Graubildern, der Speicherbedarf sich allerdings aufgrund der drei Farbdimensionen verdreifacht und auch die Rechenzeit sich dementsprechend erhöht.

Diese Redundanz der Information in Farbbildern stellen auch Ngoi und Jia (1996) fest. Sie sehen die Farbinformation natürlicher Szenen immerhin als zweidimensional an. Sie verwenden als photometrische Energie eine Kombination aus Farbkontrast und Farbintensität.

4.2.6 Regionen basierte Formulierung photometrischer Energie

Werden *snakes* für die Segmentierung von Regionen verwendet, sollte die Regionen basierte Information in die Formulierung der photometrischen Energie integriert werden. Da in dieser Arbeit der Schwerpunkt auf Kantenextraktion liegt, wird nur auf die entsprechende Literatur verwiesen: z.B. Ronfard, 1994; Zhu, Yuille, 1996; Ivins, Porill, 1998 oder Torre, Radeva, 2000.

4.3 Externe Energieterme

Externe Energieterme erlauben es, globale Kräfte und Zwänge auf *snakes* auszuüben um deren Entwicklung zu steuern. Im ursprünglichen Konzept der *snakes* waren zwei Typen externer Kräfte vorgesehen: Anziehungskräfte zu bestimmten Punkten - zu so genannten *springs* - und Abstoßungskräfte von anderen Punkten - von so genannten *volcanoes.* Über das Hinzufügen solcher Punkte sollte der Operateur während der iterativen Energieminimierung eingreifen können, um Teile der *snake* in eine bestimmte Richtung zu ziehen bzw. von bestimmten Regionen fernzuhalten.

Als Erweiterung dieses Konzepts sollen die Anziehungskräfte nicht nur zu einzelnen Punkten, sondern zu einer ganzen Kurve (Referenzkurve) vorgestellt werden. Dazu setze ich als Energieterm das Quadrat der Entfernung *d* zur Referenzkurve.

$$E_{ext} = d^2 \tag{4-24}$$

Unter der Annahme, dass sowohl die *snake* als auch die Referenzkurve als Polygonzug gegeben sind, soll der Abstand d_i des Knotens $\mathbf{v}_i$ der *snake* von der Referenzkurve folgendermaßen definiert sein:

$$d_i = \begin{cases} \min\limits_j \left(|s_{i,j}|\right) & 0 < r_{i,j} < l_j \\ \min\limits_j \left(\sqrt{r_{i,j}^2 + s_{i,j}^2}\right) & \text{sonst} \end{cases} \tag{4-25}$$

Darin ist j das j-te Polygonsegment der Referenzkurve. Dieses Polygonsegment habe die Länge l_j. Mit s und r sind Koordinaten in einem lokalen Polygonsegment-Koordinatensystem gemeint, wie in Abb. 4-14 gezeigt.

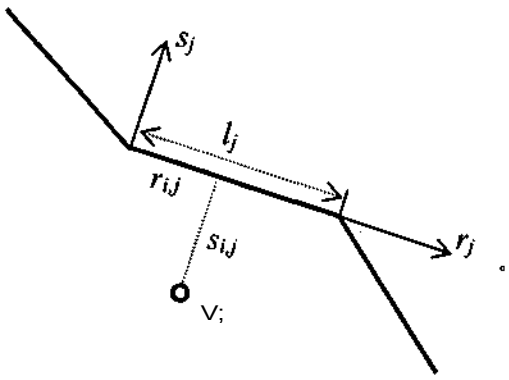


Abb. 4-14: Definition eines lokalen Koordinatensystems für das j-te Segment des Referenzpolygons. Der Knoten $\mathbf{v}_i$ hat die Koordinaten $(r_{i,j}, s_{i,j})$.

Formel (4-25) lässt sich geometrisch deuten: Der Abstand des Knotens $\mathbf{v}_i$ von der j-ten Polygonseite der Referenzkurve ist gleich dem Normalabstand dieses Knotens von der Seite falls die Projektion dieses Knotens auf die Polygonseite innerhalb des Poly-

gonsegments liegt oder andernfalls gleich der euklidischen Distanz des Knotens vom Knoten des Referenzpolygons. Diejenige Polygonseite, für die dieser Abstand minimal ist, bestimmt d_i.

Der Abstand der *snake* von der als Polygonzug gegebenen Referenzkurve wird demnach entweder als Abstand von Punkt zu Polygonseite oder als Abstand von Punkt zu Punkt berechnet. Ein Beispiel, in dem Knoten beider Art vorkommen, ist in Abb. 4-15 gegeben.

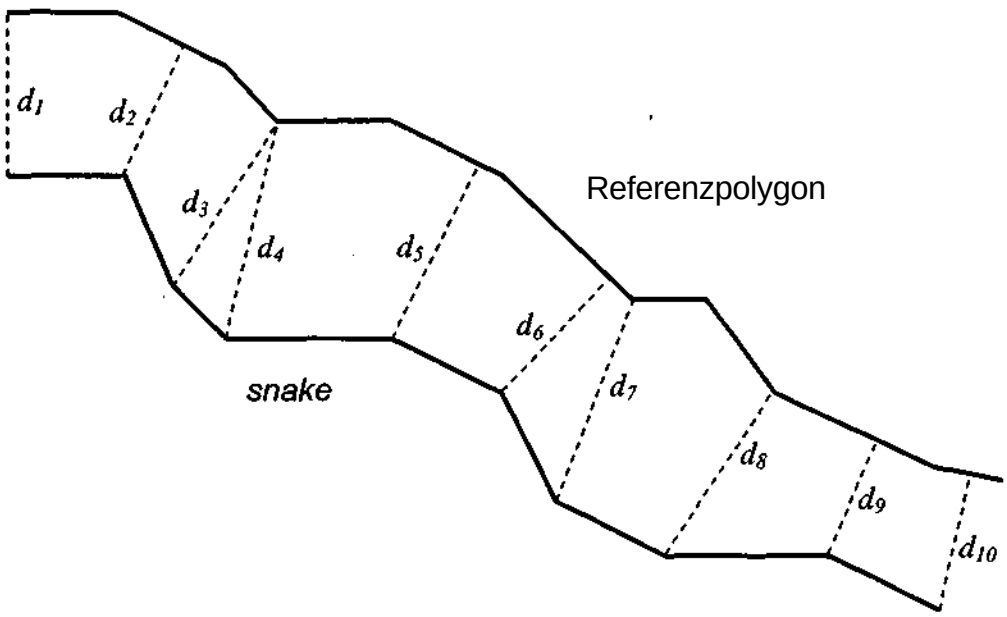


Abb. 4-15: Abstände der Knoten der *snake* vom Referenzpolygon. Für die Knoten 1, 2, 5, 6, 9 und 10 wird der Abstand als Normalabstand von der nächstgelegenen Polygonseite der Referenzkurve bestimmt. Für die Knoten 3, 4, 7 und 8 ergibt sich der Abstand als euklidische Distanz zum nächstgelegenen Knoten des Referenzpolygons.

Der externe Energieterm aus (4-24) kann in leicht modifizierter Form verwendet werden, um die *snake* zu einer gedachten Parallelkurve der Referenzkurve zu ziehen. Mit d_0 als Sollabstand der *snake zur* Referenzkurve ergibt sich als Energieterm:

$$E_{ext} = (d - d_0)^2 \tag{4-26}$$

Dieser Energieterm erreicht auf beiden Seiten der Referenzkurve ein Minimum. Eine der beiden Minima kann jedoch meist ausgeschlossen werden.

Ein Energieterm dieser Art wird bei *twin snakes* verwendet, die in Abschnitt 5.5.2 beschrieben werden (Kerschner, 1998).

4.4 Homogenisierung unterschiedlicher Energieterme

In den letzten Abschnitten wurden Energieterme der unterschiedlichsten Art und Herkunft präsentiert. Jeder dieser Energieterme hat seinen eigenen bestimmten Bereich an Energiewerten, die der Term annehmen kann. Ebenso hat jeder Term seine eigene Charakteristik, wie stark und wie schnell er sich bei Annäherung der *snake* an eine Bildkante verkleinert.

Diese individuellen Charakteristika verschiedener Energieterme machen es dem Operateur nahezu unmöglich, den relativen Einfluss der verschiedenen Terme intuitiv durch die Wahl geeigneter Gewichte steuern zu können. Die Wahl der Gewichte ist ein empirischer Prozess, der meist leider für jede Anwendung erneut durchgeführt werden muss. Die Auswirkung der Variation von Kontrast, Grauwertumfang, Auflösung und anderer Charakteristika der Bilder auf das Verhalten von *snakes* ist kaum vorhersagbar.

Wünschenswert wäre eine automatische Homogenisierung des Wertebereichs der Energieterme, sodass der Operateur den relativen Einfluss von Energietermen leicht durch die Vergabe von Prozentwerten angeben könnte. Dieses Thema wurde trotz der Unmenge an Veröffentlichungen zum Thema *snakes* kaum in der Literatur angesprochen. In Ermangelung ausreichenden Verständnisses der einzelnen Energieterme werden in den Publikationen vielfach Einheitsgewichte verwendet.

Die Lösung über Ausgleichungsrechnung (*LSB-snakes*, Abschnitt 3.3) hat den Vorteil, dass alle Energieterme in Form von Beobachtungen in das Ausgleichungssystem eingehen. Die Ausgleichungsrechnung erlaubt das Homogenisieren der Verbesserungsgleichungen der Beobachtungen (Kraus, 1996, Abschnitt B 3.5.10).

Chen und Sun (2000) regen an, den Prozess der Merkmalsextraktion in zwei Stufen zu unterteilen. In einer ersten Stufe können optimale Gewichte aus einem Trainingsdatensatz ermittelt werden, die anschließend in einem zweiten Schritt für die Detektion weiterer Elemente verwendet werden können.

In dieser Arbeit wird eine einfache Homogenisierung verwendet, wobei zunächst vor einer Iteration der Energieminimierung für den aktuellen Status und alle möglichen neuen Zustände der *snake* die minimalen und maximalen Anteile der Knoten an der Gesamtenergie ermittelt werden. Diese kleinsten und größten Energiewerte werden für jeden Energieterm bestimmt. Die Energieterme werden mit einem einfachen linearen Modell verschoben und skaliert, sodass jeder Energieterm für den aktuellen Status der *snake* Werte zwischen 0 und 1 liefert. Dieser Ansatz ist ähnlich zu dem von Williams und Shah (1992), der jedoch die Minima und Maxima für jeden Knoten gesondert bestimmt.

Am Ende dieses Kapitels möchte ich zusammenfassen, dass in der umfassenden Literatur zum Thema *snakes* viele verschiedene Formulierungen der Energieterme publiziert wurden. Das *snake*-Modell kann durch die geeignete Anpassung der Energiefunktion für verschiedene Aufgaben adaptiert werden. Man muss jedoch genau prüfen, welche Nebenwirkungen einzelne Energieterme haben können.

5 Anwendungsstrategien

Als am Ende des Abschnitts 2.3 Schwächen der *snakes* aufgelistet wurden, wurden drei Blöcke an Maßnahmen präsentiert, wie man die Probleme in den Griff bekommen kann. Die Wahl des am besten geeigneten numerischen Verfahrens zur Lösung des Optimierungsproblems wurde in Kapitel 3 behandelt. Der Anpassung der Energiefunktion war Kapitel 4 gewidmet. Bleibt als dritte Maßnahme die Anpassung der Strategie, nach der *snakes zur* Kurvenextraktion eingesetzt werden. Die Strategie kann zwar die dem *snake*-Ansatz inhärenten Probleme nicht beseitigen, bietet aber Verfahrensweisen, wie fehlgeleitete *snakes* (oder Abschnitte fehlgeleiteter *snakes)* aus lokalen Minima gezogen werden können.

In diesem Kapitel sollen automatisierte Strategien zur erfolgreichen Anwendung von *snakes* vorgestellt werden. Diese Strategien zielen auf die Behebung folgender Probleme von *snakes* ab:

- Abhängigkeit von einer guten Ausgangslage: Für das ursprüngliche *snake*-Modell muss die Ausgangslage die Zielkurve bereits sehr gut annähern, damit das Verfahren konvergiert.
- Lokale Minima: Teile der *snake* werden in lokale Minima gezogen und können diese nicht mehr verlassen.
- Nebeneinander liegende Kanten: Extraktion schlauchförmiger bzw. bandförmiger Bildelemente (breite Linien) mittels zweier gekoppelter *snakes.*

Die ersten drei Abschnitte präsentieren publizierte Vorschläge von Strategien. Anschließend werden im Rahmen der Dissertation erarbeitete Strategien vorgestellt.

5.1 Hierarchische Annäherung

Bereits im ursprünglichen Ansatz von Kass et al. (1987) war die Lösung im *scale-space* vorgesehen, um den Anziehungsbereich einer Kante und damit den Konvergenzbereich der *snake* zu vergrößern (vgl. Abschnitt 4.2.1). In Analogie zum kontinuierlich definierten *scale-space* Ansatz kann man einen diskreten, hierarchischen Ansatz betrachten, bei dem zusätzlich zur Glättung der Bilder die Auflösung der Bilder reduziert wird. Dabei wird eine Bildpyramide in mehreren Niveaus erzeugt. Der Reduktionsfaktor zwischen den Niveaus kann frei gewählt werden. Meist wird ein Faktor von zwei in beiden Koordinatenrichtungen des Bildes verwendet, sodass die Bildgröße auf ein Viertel sinkt.

Ein Ansatz zur hierarchischen Lösung von *snakes* wird in Leroy et al. (1996) präsentiert. Dabei wenden sie das Ballon-Modell (vgl. Abschnitt 4.1.2) in einer Bildpyramide an. Ihr Ziel ist lediglich die Beschleunigung des Optimierungsprozesses und nicht die Verbreiterung des Konvergenzbereichs, denn das Ballon-Modell benötigt keinen breiten Anziehungsbereich. Die Fortschreitungsrichtung wird von den Inflations- bzw. Deflationskräften bestimmt.

Die Pyramide hat in ihrem Fall Reduktionsfaktor 2. Analog zur Reduktion der Anzahl an Pixeln wird in jedem Pyramidenniveau auch die Anzahl der Knoten der *snake* angepasst. Der Optimierungsprozess löst die Optimierungsaufgabe zunächst für ein kleines Bild am obersten Niveau der Pyramide (Niveau mit der geringsten Auflösung). Anschließend wird das Ergebnis auf das nächst tiefere Niveau projiziert, wobei die Anzahl der Knoten verdoppelt wird. Dies gibt bereits eine sehr gute Ausgangslage der *snake* auf diesem Niveau. Nach der Optimierung auf diesem Niveau wird der Vorgang wiederholt, bis die *snake* auf dem untersten Niveau das Energieminimum erreicht hat.

Die Rechenzeit reduziert sich aufgrund zweier Tatsachen. Auf der einen Seite macht die *snake* auf höheren Pyramidenniveaus im Verhältnis größere Schritte. Auf der anderen Seite hat die *snake* auf höheren Niveaus weniger Knoten zu optimieren. Die Einsparung beträgt laut Leroy et al. 55%.

Ich möchte allerdings an dieser Stelle darauf hinweisen, dass bei Verwendung eines Glättungsfilters zur Pyramidenerzeugung die Schärfe einer Kante deutlich abnimmt je höher das gewählte Pyramidenniveau ist (vgl. Abb. 4-7 in Abschnitt 4.2.1). Unter Umständen findet die *snake* auf dem Startniveau keine brauchbare Lösung, da die gesuchte Kante zu wenig stark ausgeprägt ist. Für diesen Zweck müsste die Pyramide durch einfaches Wegstreichen von Zeilen und Spalten erzeugt werden. Trotzdem führt der hierarchische Ansatz zur Verbreiterung des Konvergenzbereichs selten zum erhofften Erfolg. Für Linienextraktion kann ein hierarchischer Ansatz überhaupt nicht eingesetzt werden, da eine schmale Linie in höheren Pyramidenniveaus sehr schnell verschwindet.

Eine Anwendung, bei der ein hierarchischer Ansatz gerechtfertigt ist, wird in Abschnitt 6.2 (Schnittliniensuche für die Orthophotomosaikierung) präsentiert.

5.2 Vom Groben ins Feine

Diese Strategie zielt auf die Überwindung lokaler Minima ab. Gehen wir davon aus, dass die *snake* nach dem iterativen Prozess der Energieminimierung einen Status erreicht hat, in dem zwar der überwiegende Teil der *snake* die Zielkurve gefunden hat, in dem allerdings einzelne Segmente der *snake* von lokalen Minima gefangen sind. Eine solche Situation tritt häufig dann auf, wenn die *snake* aufgrund anderer Maßnahmen (beispielsweise eine *snake* vom Ballon-Typ oder eine *snake* mit externen Energietermen) fähig ist, sich aus größerer Entfernung an die Zielkurve anzunähern. Falls auf dem Weg zum globalen Minimum eine geringer ausgeprägte Kante liegt, detektiert ein Teil der *snake* dieses lokale Minimum und bleibt von diesem gefangen. Ein derartiges Beispiel ist in Abb. 5-1 dargestellt.

Radeva et al. (1995) setzen zur Lösung dieses Problems eine Strategie „vom Groben ins Feine" (Multi-Schwellwert Schema) ein. Zuerst wird die *snake* ausschließlich von markanten Bildmerkmalen verformt und verschoben. Damit soll die *snake* deutlich ausgeprägte Kanten/Linien finden und gegebenenfalls auch über lokale Minima

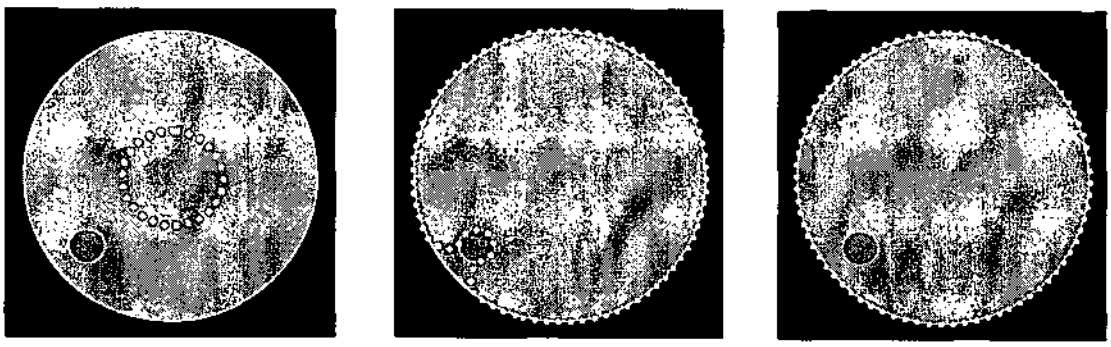


Abb. 5-1: Symbolische Skizze einer *snake* (Ballon-Modell). Links: Die Zielkurve sei die Umfahrung des großen runden Objekts, das sich kontrastreich vom dunklen Hintergrund abhebt. Das kleine runde Objekt sei eine Bildstörung. Die punktierte weiße Linie ist die Ausgangslage der *snake.* Mitte: Das Ergebnis, das eine *snake* ohne spezielle Anwendungsstrategie liefern würde. Rechts: Das gesuchte Ergebnis ergibt sich beispielsweise durch die Lösung des Optimierungsproblems mit ausschließlich sehr kontrastreichen Kanten. Anschließend wird die Optimierung mit allen Kanten nachgeschaltet, wobei sich in diesem Beispiel das Ergebnis nicht mehr ändern wird.

hinwegkommen. Erst anschließend wird der Schwellwert gesenkt und dadurch auch weniger stark ausgeprägte Bildmerkmale zum feinen Einpassen verwendet.

Als Nachteil dieser Methode soll angemerkt werden, dass als Parameter ein Schwellwert vorgegeben werden muss, der sehr großen Einfluss auf das Ergebnis hat. Robuste Verfahren zeichnen sich dadurch aus, dass eine geringe Änderung der Parameter nur einen geringen Einfluss auf das Ergebnis hat. In diesem Fall kann allerdings eine geringe Änderung des Schwellwerts darüber entscheiden, ob die *snake* weniger stark ausgeprägte Kanten ignoriert oder detektiert. Diese Strategie kann daher nicht als robust bezeichnet werden.

5.3 Zwei Punkte als Startwerte

Bei der klassischen Strategie zum Einsatz von *snakes* gibt der Operateur die Kurve mit einigen Punkten mehr oder weniger detailliert als Startwert vor. Die Abhängigkeit des Verfahrens von der Güte dieser Ausgangslage stellte sich als eine der wesentlichsten Schwächen von *snakes* heraus.

Eine alternative Strategie verlangt als Startwerte lediglich zwei Punkte. Diese zwei Punkte müssen allerdings exakt auf der zu extrahierenden Linie/Kante liegen. Zwischen den beiden Punkten wird der Pfad der minimalen Energie gesucht. Die Anforderungen an den Operateur für die Initialisierung sinken dadurch. Diese Methoden versprechen, weniger anfällig auf lokale Minima zu sein, die durch vereinzelte Kanten oder Rauschen entstehen.

Auf der anderen Seite muss betont werden, dass mit dieser Strategie die Robustheit gegenüber Lücken in der Zielkurve abnimmt. Gerade die Fähigkeit Lücken zu überbrücken wird vielfach als eine der großen Stärken von *snakes* angesehen.

Neuenschwander et al. (1997) präsentierten einen derartigen Ansatz unter dem Namen *ziplock snakes* (Reißverschluss-*snakes*). Die als Startwerte gegebenen zwei Punkte werden zunächst durch eine Gerade verbunden und Knoten werden in regelmäßigen Abständen auf dieser Gerade eingeführt. In der Optimierungsprozedur sind zunächst alle Knoten der Gerade passiv. In der Betrachtungsweise als *snake* heißt das, die Knoten unterliegen ausschließlich internen Zwängen. Beginnend an beiden Enden werden die Knoten nach und nach aktiv, d.h. die photometrischen Energieterme werden sukzessive aktiviert. Während der Optimierung besteht die *ziplock snake* aus drei Segmenten: je ein Segment von den beiden Endpunkten bis zum jeweils letzten aktiven Knoten sowie ein gerades Segment inaktiver Knoten in der Mitte (siehe Abb. 5-2).

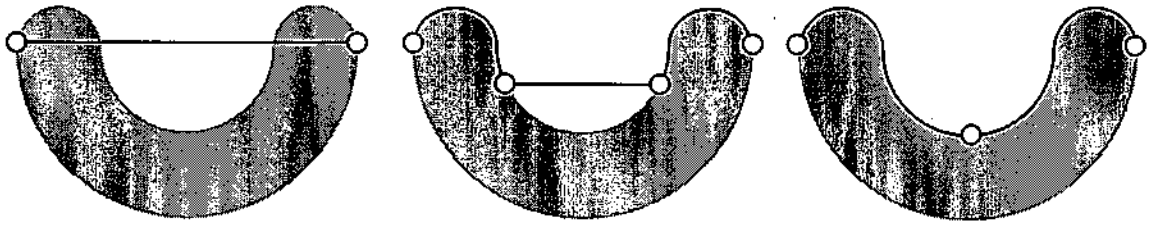


Abb. 5-2: Entwicklung von *ziplock snakes* an einem synthetischen Beispiel. Die Kreise markieren Anfangs- und Endpunkt sowie die von diesen Punkten jeweils am weitesten entfernten aktiven Knoten.

Die Optimierung der *ziplock snakes* erfolgt über Variationsrechnung (vgl. Abschnitt 3.1). Abb. 5-3 zeigt Beispiele eines Luftbildes, bei denen zusätzlich zu den Startpunkten auch die Tangentenrichtung der *snake* in diesen Punkten festgesetzt wurden (normal auf die Richtung des Gradienten der Bildfunktion).

Einen ähnlichen Ansatz, aber eine unterschiedliche Lösungsmethode präsentieren Cohen und Kimmel (1997). Sie setzen zur Lösung die Theorie der *curve evolution* ein (vgl. Abschnitt 3-4). Sie benötigen einen Startpunkt, einen Endpunkt sowie einen rechteckigen Bereich innerhalb des Bildes, in dem die Lösung liegen muss. In diesem Rechteck wird zwischen den beiden Endpunkten der *minimal path* gesucht. Dazu wird jedes Pixel innerhalb des angegebenen Bildausschnitts und jeder mögliche Verbindungspfad zum Startpunkt untersucht, wobei jeweils die über den Pfad integrierte Energie ermittelt wird. Das Minimum dieser integrierten Energien gibt den optimalen Pfad für dieses Pixel. Zusätzlich wird eine Obergrenze für die Krümmung entlang des minimalen Pfades eingeführt. Ist man am Endpunkt angelangt, erhält man garantiert das globale Minimum aller Pfade innerhalb des angegebenen Rechtecks.

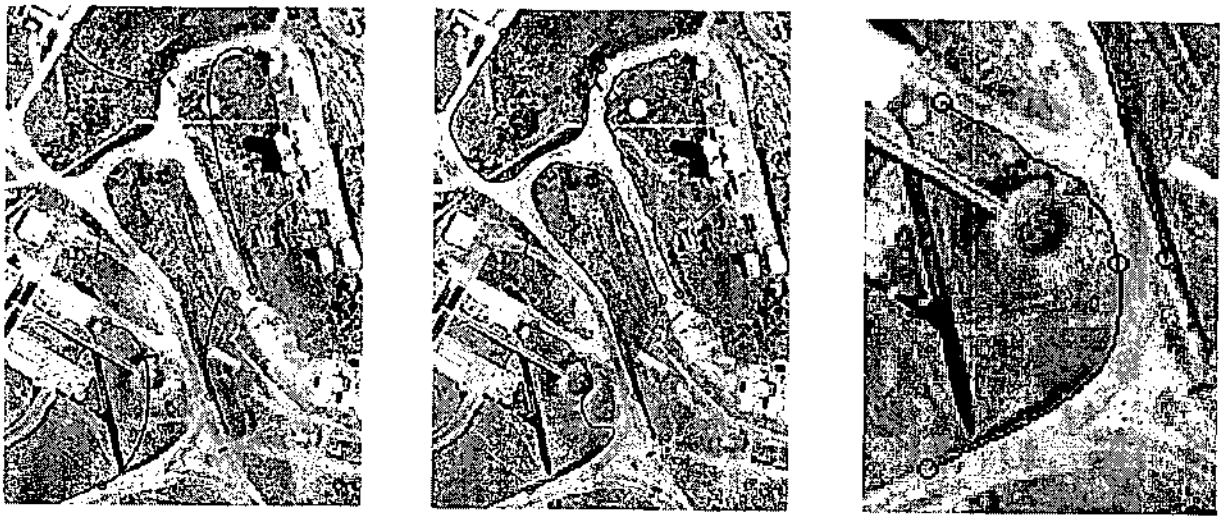


Abb. 5-3: Beispiele von *ziplock snakes* an einem Luftbild (aus Neuenschwander et al., 1997). Links: die Startpunkte von vier *snakes.* Die *snakes* wurden unter Berücksichtigung der Tangentenrichtung in den Startpunkten initialisiert. Mitte: Drei der *snakes* haben nach dem Optimierungsprozess den gewünschten Straßenrand detektiert. Rechts: Für die linke untere *snake* wird ein weiterer Knoten interaktiv vorgegeben, um die *snake* aus einem lokalen Minimum zu befreien.

Andere Methoden verwenden Algorithmen zur Suche des *minimal path* in einem Graphen, der aus den Pixeln des Energiebildes gebildet wird (z.B. Han et al., 2001).

5.4 Unterteilung in Segmente

Diese Strategie zielt ebenfalls auf die Überwindung lokaler Minima ab. Ich gehe wieder davon aus, dass ein Abschnitt der *snake* in einem lokalen Minimum gefangen ist. Ein derartiges Beispiel wurde in Abb. 5-1 dargestellt.

Durch die Analyse der Energiewerte entlang der *snake* sollen nun Abschnitte festgelegt werden, in denen die *snake* ausreichend niedrige Energiewerte erreicht hat, und andere, in denen die Energiewerte auffallend groß sind. Im Sinne der Definition von Abschnitt 1.3 ist dieser Selbstdiagnoseschritt eine Grundvoraussetzung zum Einsatz von *snakes* in autonomen Systemen.

Die Ergebnisse dieser Selbstdiagnose können dem Benutzer als „Ampelergebnis" präsentiert werden *(traffic light system,* Förstner, 1995). Dabei sollen vertrauenswürdige Ergebnisse in Grün dargestellt werden, unsichere in Gelb und verworfene Ergebnisse in Rot. Umgelegt auf eine *snake* nach der Optimierung gilt: Jene Segmente der *snake,* deren mittlerer Energieanteil pro Knoten unter einem vom Operateur bestimmten Schwellwert liegt, können im Allgemeinen in Grün ausgewiesen werden. Liegt der mittlere Energieanteil allerdings signifikant höher als in benachbarten Segmenten, soll dieses Segment in Gelb gekennzeichnet werden.

Segmente, deren mittlerer Energieanteil pro Knoten einen zweiten Schwellwert übersteigt, sollen in Rot gezeichnet werden.

Das Auftrennen der Kurve in Segmente wird im folgenden Abschnitt erklärt. Die Wahl der Schwellwerte ist eine schwierige Aufgabe und wird in Abschnitt 5.4.2 behandelt. Es wäre wünschenswert, die Schwellwerte automatisch aus den Energiewerten der *snake* abzuleiten.

Nachdem die *snake* in grüne, gelbe und rote Segmente unterteilt worden ist, können die Segmente in einer semi-automatischen Anwendung dem Operateur präsentiert werden. Dieser muss dann eine Eingriffsmöglichkeit haben. Er muss das Segment der *snake* editieren können. Nach dem interaktiven Verschieben einzelner Knoten kann das Teilsegment der *snake* neu optimiert werden, sodass nun hoffentlich das „richtige" Minimum detektiert wird. Andernfalls kann durch Hinzufügen von *springs* und/oder *volcanoes* das Segment zu bestimmten Punkten hingezogen werden bzw. von anderen Punkten weggedrängt werden (vgl. Abschnitt 4.3).

Auch in diesem Zusammenhang kann man eine Automatisierung anstreben um das unsichere oder fehlerhafte Segment aus dem lokalen Minimum zu ziehen. Dieses Ziel wird im Abschnitt 5.4.3 verfolgt.

5.4.1 Automatische Unterteilung der *snake* in Segmente

Zur Unterteilung der *snake* in einzelne Segmente werden die Energiewerte der einzelnen Knoten entlang der gesamten *snake* analysiert. Der Anteil eines Knotens an der Gesamtenergie der *snake* wird im Folgenden als Energieanteil bezeichnet. Jeder Energieanteil setzt sich wiederum aus internen, photometrischen und ev. externen Anteilen zusammen.

Das Segmentierungsverfahren geht von folgenden Annahmen aus:

- Die *snake* hat entlang ihrer gesamten Länge ein Minimum erreicht, wobei allerdings einzelne Segmente nicht die Zielkurve sondern eine andere Bildkante/Linie detektiert haben.
- Der überwiegende Teil der *snake*-Knoten hat die Zielkurve erreicht.
- Das Verbindungsstück zwischen jenen Segmenten, die unterschiedliche Kurven detektiert haben, enthält zumindest einen Knoten. Für diesen Knoten ist der Energieanteil signifikant höher als für die anderen Knoten.

Der dritte Punkt ist notwendig, damit das Verfahren die Grenze der Segmente festlegen kann. Es muss ein Knoten in der Zone hoher photometrischer Energie liegen. Allerdings ist diese Forderung bei Formulierungen der internen Energie, bei denen die Knoten nicht gleichabständig gehalten werden, im Allgemeinen nicht erfüllt. Die Knoten gleiten dann entlang der *snake* in Bereiche niedriger photometrischer Energie. In diesen Zonen bündeln sich die Knoten, während das Verbindungsstück hoher photometrischer Energie keinen Knoten enthält (vgl. Abschnitt 4.1.7). Die Unterteilung in Segmente beginnt daher mit folgendem Schritt:

Schritt 1: Einfügen zusätzlicher Knoten

Falls der Abstand zweier aufeinander folgender Knoten der *snake* einen Schwellwert übersteigt (z.B. der doppelte Wert des Knotenabstands des Ausgangsstatus der *snake),* werden so viele zusätzliche Knoten auf der Geraden zwischen den betroffenen Knoten interpoliert und in die *snake* eingefügt, wie notwendig sind, dass der Abstand der Knoten unter den Schwellwert sinkt.

Anschließend wird eine einzige Iteration der Energieminimierung durchgeführt, damit jeder neu eingefügte Knoten die Chance erhält, die bestmögliche Position innerhalb eines begrenzten Bewegungsfreiraums einzunehmen.

Schritt 2: Bestimmung des Segments mit geringster Energie

Zunächst wird eine minimale Segmentlänge n_{min} vorgegeben, die es später erlaubt, ein Segment als eigenständige *snake* zu behandeln (z.B. 5 Knoten). Anschließend wird mit einem gleitenden Mittelwertfilter das Segment der Länge n_{min} bestimmt, für das der mittlere Energieanteil der Knoten minimal ist:

$$\overline{E}_{min} = \frac{\min\limits_{j \in [1, N-n_m} \left(\sum_{i=j}^{j+n_{min}} E(\mathbf{v}_i) \right)}{n_{min}} \tag{5-1}$$

mit *N* als Anzahl der Knoten der *snake.*

Schritt 3: Verbreiterung des Segments geringster Energie

Ausgehend von diesem Segment sollen in beide Richtungen Knoten hinzugefügt werden solange sich deren Energieanteile nur geringfügig vom mittleren Energieanteil $\overline{E}$ des Segments unterscheiden. Dazu wird ein statistischer Ausreißertest nach Grubbs (NIST/SEMATECH, 2003) durchgeführt. Unter der Annahme, dass die Energieanteile der Knoten normalverteilt sind, wird die Hypothese geprüft, ob der Energieanteil des nächsten Knotens zur selben Grundgesamtheit wie die Energieanteile des momentanen Segments gehört. Die Testgröße *A* berechnet sich zu:

$$\Delta = \frac{\left| E - \overline{E} \right|}{s} \tag{5-2}$$

Dabei ist $\overline{E}$ der Mittelwert der Stichprobe, *E* der Energieanteil des neuen getesteten Knotens und *s* die aus der Stichprobe vom Umfang n geschätzte Standardabweichung:

$$s = \sqrt{\frac{\sum_{i=1}^{n} \left(E_i - \overline{E} \right)^2}{n-1}} \tag{5-3}$$

Dabei wird der getestete Wert bei der Berechnung von Mittelwert und Standardabweichung mit verwendet.

Die statistische Sicherheit *S*, mit der diese Entscheidung getroffen wird, wird für unseren Test verhältnismäßig niedrig angesetzt. Im Zweifel soll nämlich die Hypothese eher abgelehnt werden als angenommen. Damit ergeben sich im Zweifel mehrere und kürzere Segmente. Die Sicherheitsgrenzen des Grubbs-Tests können statistischen Büchern entnommen werden (vgl. Abb. 5-4).

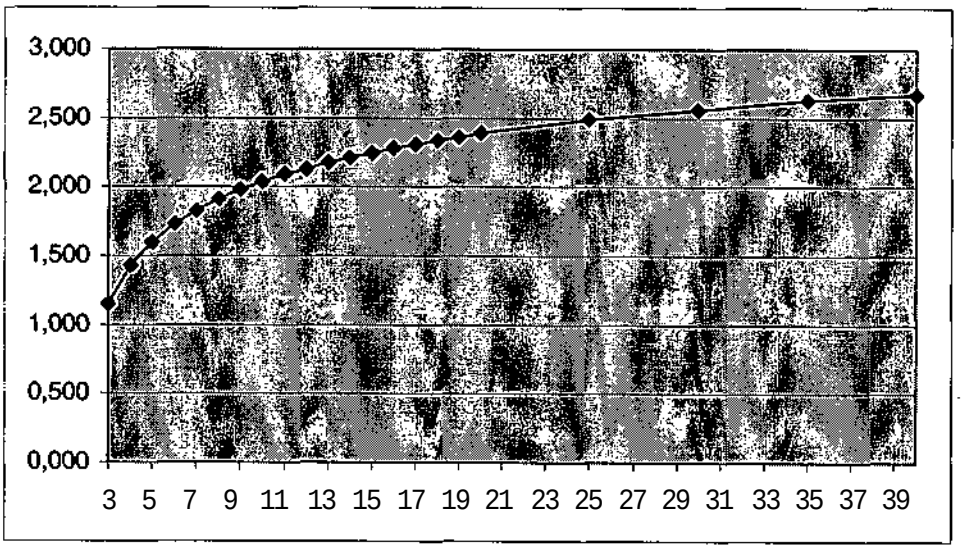


Abb. 5-4: Sicherheitsgrenzen $\Delta_{S=90}$ des Ausreißertests nach Grubbs für unterschiedlich große Stichproben der Mächtigkeit *n* bei einer statistischen Sicherheit von *S* = 90%.

Die Hypothese, dass die Abweichung des Energieanteils des nächsten Knotens von der mittleren Energie des momentanen Segments nur zufällig ist, ist dann zu akzeptieren, wenn:

$$\Delta \leq \Delta_S \tag{5-4}$$

In diesem Fall wird der untersuchte Knoten zum aktuellen Segment hinzugefügt. Die Verbreiterung des Segments in einer Richtung wird abgebrochen, wenn der Knoten die Ungleichung (5-4) nicht mehr erfüllt und als Ausreißer eingestuft wurde oder das Ende der *snake* erreicht wurde. Das erste Segment ist abgeschlossen, wenn die Verbreiterung in beide Richtungen beendet ist.

Schritt 4: Bestimmung der Segmente mit der nächst-niedrigeren Energie

Analog zu Schritt 2 wird mit all jenen Knoten, die nicht im Segment der niedrigsten Energie enthalten sind, das Segment mit der zweit-niedrigsten Energie gesucht. Anschließend wird dieses Segment wie in Schritt 3 beschrieben verbreitert. Dieser Vorgang wird so lange wiederholt bis entweder alle Knoten einem Segment zugeordnet sind oder nur noch Serien von Knoten übrig sind, die kürzer als n_{min} sind (die minimale Länge eines erlaubten Segments).

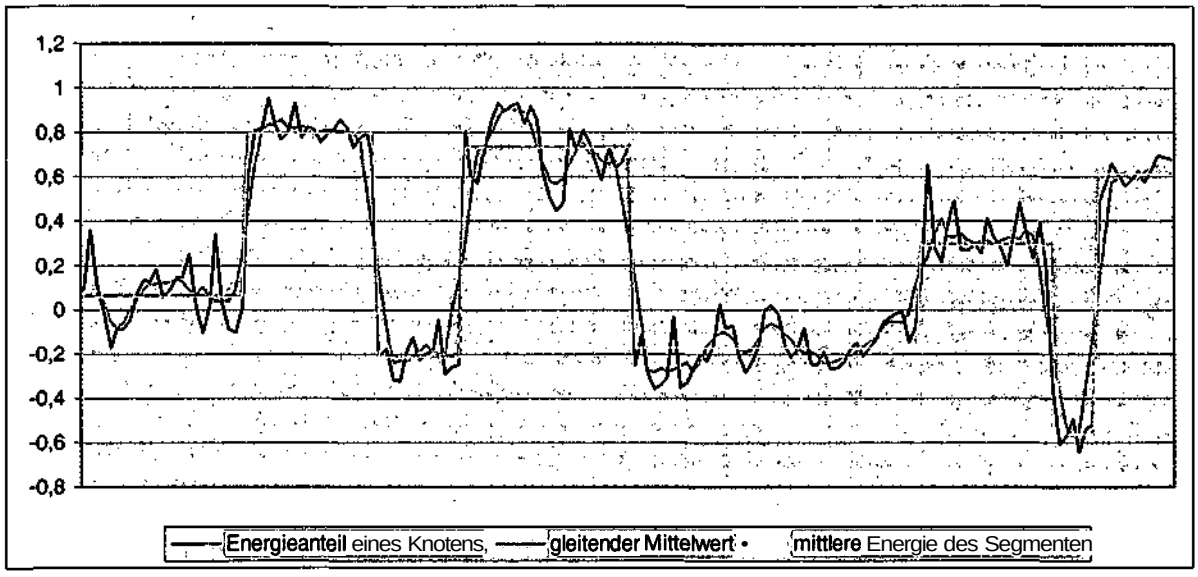


Abb. 5-5: Oben: Unterteilung der *snake* in Segmente signifikant unterschiedlicher Energie und Klassifizierung als „Ampelergebnis". Die Ausgangslage der *snake*

wurde als Polygonzug mit fünf Knoten digitalisiert (rot). Die *snake* hat großteils die Randsteinkante detektiert. Teile der *snake* (im Vordergrund) wurden allerdings zur inneren Begrenzung des Randsteines gezogen. Diese Segmente wurden automatisch erkannt und in Rot ausgewiesen (Schwellwert 0.4). Der stark gekrümmte Abschnitt hat ebenfalls höhere Energiewerte und wurde deshalb in Gelb gezeichnet (Schwellwert 0.2). Am Übergang zum Schatten steigen die Energiewerte sprunghaft an. Dieses Segment fällt knapp in den roten Bereich (mittlere Energie 0.6). Unten: Die Energieanteile der Knoten der *snake* von rechts unten nach links oben (blau). In Magenta ist der gleitende Mittelwert der kürzest möglichen Segmente (5 Knoten) dargestellt. In Gelb sieht man die mittleren Energiewerte der Segmente.

Schritt 5: Zuordnung der verbleibenden Knoten zu bestehenden Segmenten

Die Knoten, die noch in keinem Segment enthalten sind, zeichnen sich dadurch aus, dass sie eher hohe Energieanteile besitzen. Da diese Serien von Knoten zu kurz sind um ein eigenes Segment zu bilden, werden sie zu demjenigen Nachbarsegment hinzugefügt, das die höhere Energie besitzt.

Damit sind schlussendlich alle Knoten genau einem Segment zugeordnet. Nun können die Segmente anhand ihrer mittleren Energieanteile pro Knoten klassifiziert werden. Ein Beispiel der Unterteilung in Segmente ist in Abb. 5-5 zu sehen.

5.4.2 Klassifizierung der Segmente als Ampelergebnisse

Zur Klassifizierung der Segmente müssen zwei Schwellwerte $E\backslash$ und E_2 vorgegeben werden, anhand derer die Zugehörigkeit des Segments zu einer Klasse K entschieden werden kann. Die Klassen geben an, ob ein Segment als sicheres, unsicheres oder unbrauchbares Ergebnis zu klassifizieren ist:

$$K = \begin{cases} grün: & \overline{E} < E_1 \\ gelb: & E_1 < \overline{E} \leq E_2 \\ rot: & \overline{E} > E_2 \end{cases} \tag{5-5}$$

Die Festlegung der Schwellwerte ist der heikelste Punkt und die Schwachstelle in dieser Methode. Da die Verhältnisse von einer Extraktionsaufgabe zur anderen stark variieren (vor allem der Bildkontrast, aber auch die geometrische Auflösung etc.), sind die Schwellwerte kaum von einem Projekt auf das andere übertragbar. Die Homogenisierung der Energieterme (vgl. Abschnitt 4.4) bietet die Chance, allgemeingültige Schwellwerte festzulegen. So wurde das Beispiel in Abb. 5-5 mit vordefinierten Werten $E\backslash = 0.2$ und $E_2 = 0.4$ klassifiziert.

Ein praktikabler Ansatz zur interaktiven Festlegung der Schwellwerte weist dem Operateur bei der ersten extrahierten Kurve alle Segmente in Grau aus. Der Operateur identifiziert eines der Segmente und ordnet ihm eine Klasse K_0 e $\{grün, gelb, rot\}$ zu. Der Algorithmus wählt die Schwellwerte zunächst nach folgendem Schema:

$$\begin{cases} E_1 = 2\overline{E} & E_2 = 4\overline{E} & \text{für } K_0 = \text{grün} \\ E_1 = \frac{2}{3}\overline{E} & E_2 = \frac{4}{3}\overline{E} & \text{für } K_0 = \text{gelb} \\ E_1 = \frac{2}{5}\overline{E} & E_2 = \frac{4}{5}\overline{E} & \text{für } K_0 = \text{rot} \end{cases} \tag{5-6}$$

Nachdem die Schwellwerte anhand des ersten Segments bestimmt worden sind, werden alle Segmente in der entsprechenden Farbe gezeichnet. Der Operateur kontrolliert die Klassifizierung und ändert gegebenenfalls die Farbe einzelner Segmente. Der Algorithmus versucht, die Schwellwerte jeweils so anzupassen, dass alle vom Operateur vorgegebenen Klassen erreicht werden. Dazu wird er bald das Verhältnis 1:2 zwischen E_1 und E_2 nicht aufrechterhalten können. Falls der Operateur eine Klassifizierung vorgeben will, die sich nicht mehr in den Schwellwerten abbilden lässt, muss eine Fehlermeldung ausgegeben werden.

Bei der nächsten extrahierten Kurve können die Schwellwerte wieder benutzt werden. Nichtsdestotrotz muss der Operateur wieder die Möglichkeit bekommen, die Klassifizierung anzupassen.

5.4.3 Automatisiertes Verbessern von Segmenten hoher Energie

Nach dem Klassifizierungsschritt erhält man als Ergebnis, welche Segmente der *snake* ausreichend niedrige Energie erreicht haben, und welche Segmente hohe Energie aufweisen. Jedenfalls haben alle Segmente ein Minimum erreicht (ein lokales oder das globale). Weitere Iterationen mit derselben Energiefunktion bringen keine Änderung des Status. Damit eine Verbesserung des Status möglich wird, müssen die schlechten Segmente der *snake* zuerst aus dem lokalen Minimum befreit werden. Soll dies automatisiert erfolgen, ist dazu eine alternative Energiefunktion notwendig, welche sich durch Verändern der Gewichte der Energieterme oder durch Hinzufügen weiterer Energieterme ergibt. Als zusätzliche Energieterme kommen vor allem externe Terme in Frage.

Wie erwähnt, muss das Segment aus einem Minimum gezogen werden. Das Minimum wurde meist dadurch erreicht, dass sich die *snake* in ein Tal der photometrischen Energie bewegt hat. Um dieses Tal wieder verlassen zu können, muss der Einfluss der photometrischen Energie reduziert werden, indem ihr Gewicht herabgesetzt wird.

Im Falle, dass die Energiefunktion keine externen Terme enthält, kann aber nicht gewährleistet werden, dass sich die *snake* nach dem Abgewichten der photometrischen Energie in die richtige Richtung weiterbewegt. In Abb. 5-6 sind zwei schematische Beispiele gezeigt. Im linken Beispiel wird die *snake* durch Deaktivieren der photometrischen Energieterme das lokale Minimum verlassen können. Im rechten Beispiel wird das Abgewichten der photometrischen Energie allerdings nicht dazu führen, dass sich die *snake* in Richtung globales Minimum bewegt.

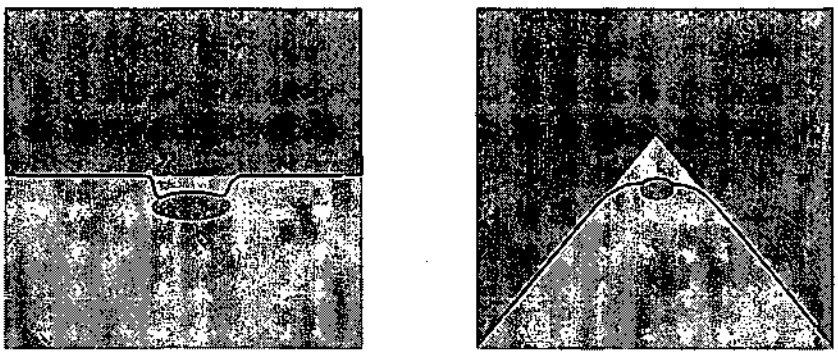


Abb. 5-6: Zwei Beispiele mit einem Segment in der Mitte, das von einer falschen Bildkante gefangen ist. Im linken Beispiel hat dadurch auch die interne Energie einen höheren Wert. Im rechten Beispiel führt das fehlerhafte Segment nicht zu einer Steigerung der internen Energie.

Abb. 5-6 zeigt, dass die automatisierte Strategie zum Verlassen lokaler Minima nur dann sicher zum Erfolg führt, wenn die *snake* ohne Einfluss photometrischer Energie eine definierte **Fortschreitungsrichtung** hat. Falls die interne Energie dies nicht garantiert, kann dies mittels des Ballon-Modells oder durch Einbeziehung externer **Terme** erreicht werden.

Die detaillierte Strategie zur Überwindung lokaler Minima besteht wieder aus mehreren Schritten:

Schritt 1: Vorbereitung eines *snake*-Segments

Alle aufeinander folgenden Segmente der *snake,* für die diese Strategie angewendet werden soll (im Allgemeinen alle aufeinander folgenden gelben und roten Segmente), werden zu einem Segment zusammengefasst und es wird eine eigene *snake* daraus gebildet. Folgt dieses Segment auf ein grünes Segment, werden die beiden letzten Knoten des grünen Segments als unveränderlicher Anfangspunkt und feste Tangente übernommen. Dasselbe gilt, falls ein grünes Segment auf das Ende der neuen *snake* folgt. Damit ist gewährleistet, dass bei Erfolg der Strategie die neue *snake* den entsprechenden Teil der alten *snake* ersetzen kann ohne dass Knicke entstehen können.

Beispiel: Angenommen, nach der Unterteilung entstanden drei Segmente: ein grünes Segment mit den Knoten 1 bis **10,** ein gelbes Segment mit den Knoten **11** bis 20 und ein rotes Segment mit den Knoten 21 bis 30. Es wird eine neue *snake* gebildet mit den Knoten 9 bis 30, wobei die beiden Knoten 9 und 10 während der Energieoptimierung nicht verschoben werden dürfen. Falls die Strategie zum Erfolg (niedrigere Energie für die neue **Teil-*snake***) führt, werden die Knoten **11** bis 30 durch die Knoten 3 bis 22 der neuen *snake* ersetzt.

Schritt 2: Speichern des Status des *snake*-Segments

Die Energie der in Schritt 1 gebildeten Teil-*snake* wird gespeichert um am Ende beurteilen zu können, ob ein verbesserter Status erreicht wurde.

Schritt 3: Festlegen einer alternativen Energiefunktion

Die alternative Energiefunktion entsteht im Allgemeinen aus der primären Energiefunktion durch Abwerten der photometrischen Terme (z.B. Gewicht Null). Falls die Energiefunktion keine externen Energieterme enthält, die sicherstellen, dass sich die Teil-*snake* ohne Einfluss photometrischer Terme von der aktuellen Lage wegbewegt, muss der Operateur dazu aufgefordert werden, durch Hinzufügen von *springs* oder *volcanoes* (vgl. Abschnitt 4.3) externe Terme für die alternative Energiefunktion bereitzustellen.

Schritt 4: Befreiung aus dem Minimum

Dazu werden einige wenige Iterationen der Energieoptimierung mit der alternativen Energiefunktion durchgeführt. Die Anzahl an notwendigen Operationen hängt von der Breite des Anziehungsbereichs von Tälern photometrischer Energie ab. Es muss gewährleistet sein, dass die Teil-*snake* im nächsten Schritt nicht wieder in das lokale Minimum zurückkehrt.

Schritt 5: Optimierung mit der primären Energiefunktion

Nun wird wieder unter Verwendung der ursprünglichen Energiefunktion ein Minimum der Energie gesucht. Als Ergebnis sollte ein neuer Status der Teil-*snake* gefunden werden.

Schritt 6: Bewertung des neuen Status

Falls die Energie des neuen Status niedriger ist, als die im Schritt 2 gespeicherte Energie des alten Status, wird für die ursprüngliche *snake* das betroffene Segment durch die Teil-*snake* ausgetauscht.

Falls sich die *snake* nach dem Anwenden der sechs Schritte auf alle Segmente verändert hat, beginnt das Verfahren erneut mit dem Unterteilen in Segmente (Abschnitt 5.4.1). Die roten und gelben Segmente sollten nach und nach kürzer werden. Im Idealfall bleibt schließlich nur ein grünes Segment übrig.

5.5 Kopplung zweier *Snakes*

In diesem Abschnitt soll eine Methode der Koppelung zweier *snakes* vorgestellt werden, die es erlaubt parallele Kurven zu finden. Anwendung finden diese Modelle beispielsweise bei der Straßenextraktion. Zuerst wird der Ansatz der *ribbon snakes* vorgestellt. Anschließend werden die im Rahmen der Dissertation entwickelten *twin snakes* präsentiert. Schließlich werden die gekoppelten *snakes zur* robusteren Extraktion einer einzelnen Kurve spezialisiert.

5.5.1 Linienextraktion mit *Ribbon Snakes*

Beim Modell der *ribbon snake* (Fua, 1996; Mayer et al., 1998) werden die Elemente der Knoten $\mathbf{v}(s)$ der *snake* um die Breite $w(s)$ erweitert (vgl. Abb. 5-7):

$$\mathbf{v}(s) = (x(s), y(s), w(s))^{\mathrm{T}} \qquad (5\text{-}7)$$

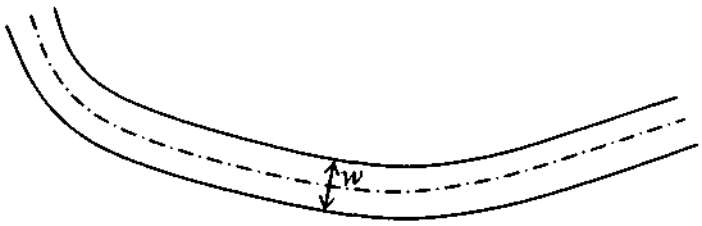


Abb. 5-7: Bei der *ribbon snake* wird die Mittellinie und die Breite der Linie modelliert.

Die photometrische Energie wird auf der Normalen an die Kurve in $(x(s),y(s))^{\mathrm{T}}$ im Abstand $w/2$ rechts und links der Kurve bestimmt. Für die Breite der Linie werden dieselben internen Energieterme angesetzt wie für die Koordinaten der Mittellinie.

Abb. 5-8: Änderungen der Breite der Linie (z.B. eine Pannenbucht an der Straße) führen zu Verschwenkungen der Achse der *ribbon snake.*

Das Modell der *ribbon snakes* ist gut geeignet für Linien, deren Breite sich nicht ändert. Das Modell ist allerdings schlecht für ungleichmäßig breite Linien einsetzbar, da Änderungen in der Breite ungewünschte Verformungen der Mittellinie hervorrufen (vgl. Abb. 5-8).

5.5.2 Das Modell der *Twin Snakes*

Der im Rahmen dieser Dissertation entwickelte Ansatz betrachtet die beiden Begrenzungslinien als getrennte *snakes.* Da die beiden Kurven eine ähnliche Form haben, werden sie als Zwillinge bezeichnet *(twin snakes;* Kerschner, 1998).

Die Energiefunktion der beiden *snakes* wird um Energieterme erweitert, die genau dann minimal werden, wenn die eine *snake* einen konstanten Abstand von der anderen Kurve erreicht hat. Dies wird mit einem externen Energieterm erreicht, wie er in Abschnitt 4.3 präsentiert wurde, wobei die dort angegebene Referenzkurve der jeweilige Zwillingspartner ist (Gleichung 4-26). Der Abstand, den die beiden *snakes* voneinander haben sollen, wird vom Operateur vorgegeben. Die externen Energieterme in der Energiefunktion bewirken somit, dass jede *snake* zu einer Parallelkurve der anderen angezogen wird.

Die Vorgabe des Sollabstands ist in einem halbautomatischen System, in dem der Benutzer den Prozess steuert, durch Digitalisieren zweier Punkte leicht möglich. Für automatische Anwendungen muss der Abstand von vornherein vorgegeben sein. Je besser der Sollabstand abgeschätzt wird, desto größer ist die Chance, dass der Prozess die Zielkurven detektiert.

Wie in Abschnitt 2.3 festgestellt wurde, besteht eines der Hauptprobleme von klassischen *snakes* in der Notwendigkeit eines sehr guten Ausgangsstatus (Lage und Position), damit das Verfahren zur gewünschten Kurve konvergiert. Liegt die Ausgangslage der *snake* zu weit von der Kante entfernt, liegt sie bald außerhalb des Anziehungsbereichs der Kante.

Bei *twin snakes* ist dennoch eine Anziehungskraft gegeben, nämlich die Anziehung zu ihrem Zwillingspartner bzw. einer Parallelkurve des Zwillingspartners. Daher vermindert dieser Ansatz die Abhängigkeit der *snake* von einer guten Ausgangslage.

Das Problem der Abhängigkeit vom Ausgangsstatus wurde bereits mehrfach in dieser Arbeit angesprochen. Die Methode der *twin snakes* hat einen Grundgedanken mit dem Ballon-Modell gemein (vgl. Abschnitt 4.1.2): Bevor die Reduktion der photometrischen Energie wirksam wird, muss auf eine andere Weise eine Fortschreitungsrichtung vorgegeben werden. Beim Ballon-Modell geschieht dies durch Inflationskräfte, die den Ballon aufblasen. Voraussetzung an die Ausgangslage des Ballons ist lediglich, dass er vollständig innerhalb der Zielkurve liegt. Eine analoge Voraussetzung gilt für die Ausgangslage der beiden *twin snakes:* Die gesuchten parallelen Kurven müssen zwischen den Kurven des Ausgangsstatus liegen.

In Abb. 5-9 sind die Stadien der Optimierung anhand eines Beispiels zu sehen. Dargestellt ist die hintere Tür eines Autos. Das Photo wurde unter natürlichen Beleuchtungsverhältnissen aufgenommen und anschließend gescannt. Die Zielkurve sei der ca. 10 dunkle Pixel breite Spalt zwischen Karosserie und Tür. Es handelt sich daher um eine Linienextraktionsaufgabe. Man erkennt, dass die Zielkurve wie gefordert zwischen den Ausgangslagen der beiden *snakes* liegt, dass aber auch die annähernd parallel verlaufende Karosseriebegrenzung dazwischen liegt (Abb. 5-9 (a)).

Abb. 5-9 (b) zeigt die *snake* in Weiß und den Ausgangsstatus in Grau nach etwa 15 Iterationen. Die Verringerung der externen Energie (aufgrund der Anziehungskräfte der beiden Zwillingspartner zueinander) überwiegt die Verringerung aller anderen Energieterme und lässt die beiden *snakes* näher zusammenrücken. Nach etlichen weiteren Iterationen haben Teile der *snakes* ein Minimum der photometrischen Energie erreicht (Abb. 5-9 (c)). Bald darauf erreichen beide *snakes* ein Minimum entlang ihrer gesamten Länge (Abb. 5-9 (d)). Weitere Iterationen können die Energie nicht mehr verringern, da die Verringerung der externen Energie nicht ausreicht um die Steigerung der photometrischen Energie zu überwiegen. Die obere *snake* hat die Zielkurve detektiert. Die untere *snake* wurde allerdings wie befürchtet von der Karosseriebegrenzung gefangen.

Nun folgt ein Selbstdiagnoseschritt anhand der Segmentierungsstrategie von Abschnitt 5.4 in leicht modifizierter Art: Anstatt die Anteile der Knoten an der Energie auf Ausreißer zu testen, wird direkt der Abstand der beiden Kurven voneinander untersucht. Die *twin snakes* überprüfen, ob sie einen Abstand voneinander erreicht haben, der etwa dem vom Operateur vorgegebenen Wert entspricht. In unserem Beispiel ist der Abstand durchwegs 50 Pixel oder mehr. Da der Operateur etwa 10 Pixel Abstand vorgegeben hat, wird das Ergebnis nicht akzeptiert. Offensichtlich hat eine der

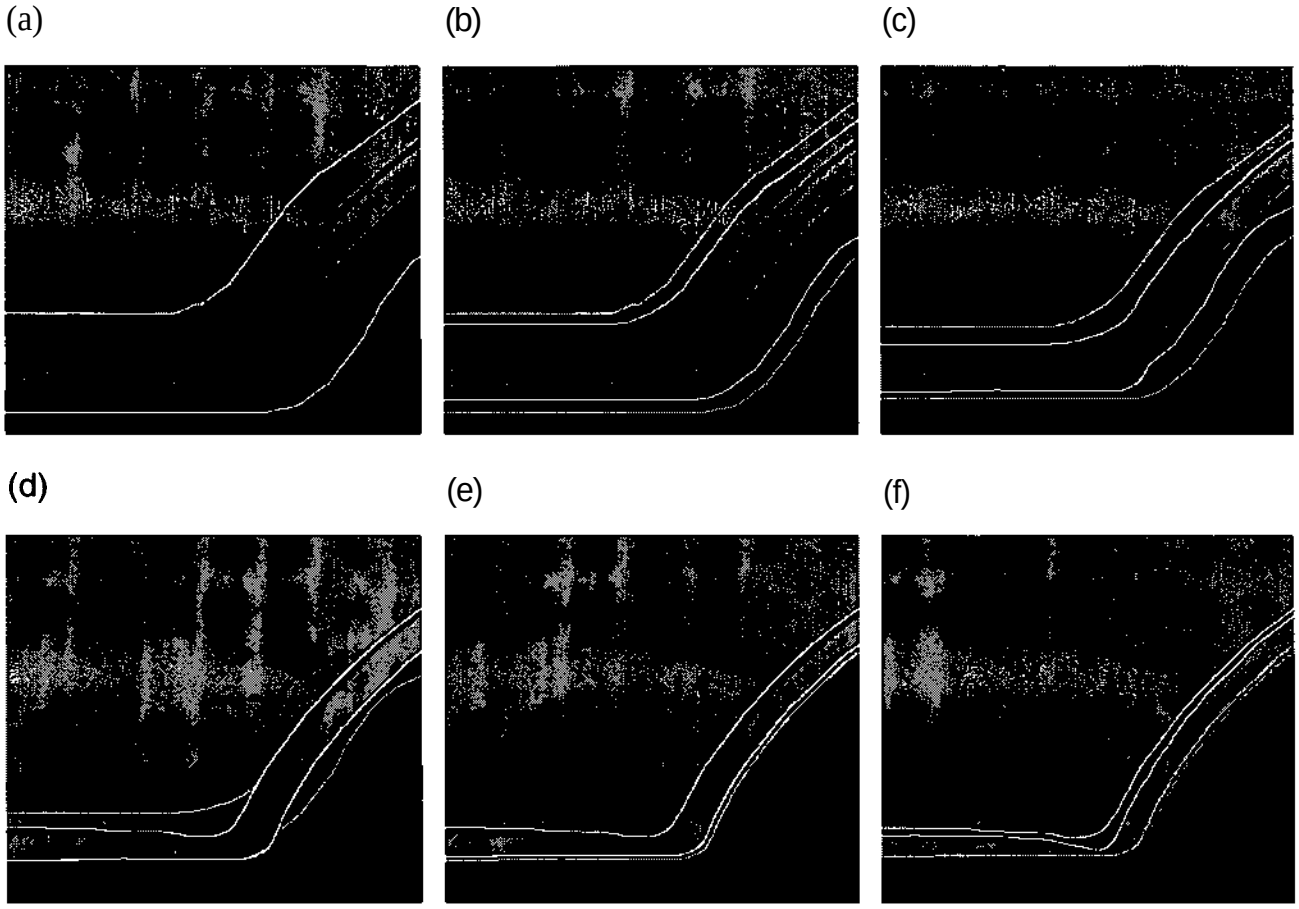


Abb. 5-9: Entwicklung von *twin snakes* (der aktuelle Status in Weiß, der Status des jeweiligen vorigen Bildes in Grau): (a) Ausgangstatus, (b) nach 15 Iterationen der Energieminimierung, (c) Teile der *snakes* haben eine Bildkante erreicht, (d) vorläufig zu Ende iterierter Status, (e) nach 5 Iterationen ohne Berücksichtigung photometrischer Energie, (f) der nächste zu Ende iterierte Status (Energieminimierung mittels dynamischer Programmierung; aus Kerschner, 1998).

beiden *snakes* (oder beide) eine falsche Bildkante detektiert. Beide Zwillingspartner werden in ihrer gesamten Länge in Rot ausgewiesen.

Um mit der Optimierung fortsetzen zu können muss diejenige *snake,* die auf ihrem Weg zur Zielkurve von dem Energieminimum einer anderen Kante gefangen wurde, aus diesem Minimum gezogen werden. Die Entscheidung, welche der beiden Kanten im falschen Minimum liegt, soll automatisch getroffen werden. Dafür werden die mittleren Energieanteile pro *snake*-Knoten verglichen. In unserem Beispiel hat tatsächlich die untere *snake* höhere mittlere Energiewerte als die obere.

Würde der Prozess an dieser Stelle die falsche *snake* auswählen um sie aus dem Minimum zu ziehen und weiter wandern zu lassen, würde der Prozess diese Fehlentscheidung am Ende daran erkennen, dass die beiden *snakes* zusammenfallen. Der Prozess hätte dann eine einzige Kante gefunden, aber nicht wie gefordert eine Linie

der gewünschten Breite. Daher muss sich der Prozess den Status der *twin snakes* zum Entscheidungszeitpunkt merken, damit er im Falle des Scheiterns zu diesem Status zurückkehren und den anderen Kandidaten auswählen kann.

Nachdem also bestimmt wurde, welche der beiden *snakes* aus dem Minimum befreit werden soll, muss die alternative Energiefunktion für den Schritt 3 aus Abschnitt 5.4.3 bestimmt werden. Für *twin snakes* ist dies einfach, da die Fortschreitungsrichtung durch die Anziehungskräfte zum Zwillingspartner vorgegeben ist. Es wird einfach das Gewicht der photometrischen Terme auf Null gesetzt. Nach wenigen Iterationen (Abb. 5-9 (e)) wird die photometrische Energie wieder aktiviert und die Optimierung für diese *snake* fortgesetzt bis sie das nächste Minimum erreicht hat (Abb. 5-9 (f)).

Nun haben die beiden *snakes* entlang ihrer gesamten Länge einen Abstand von maximal 20 Pixeln. Dies ist innerhalb der Toleranz. Betrachtet man das Ergebnis, erkennt man allerdings, dass die untere *snake* in einem kurzen Stück nicht entlang des Türspalts läuft, sondern auf einem schwarzen Fleck hängen geblieben ist (ein aufgeklebtes Signal für die photogrammetrische Bestimmung der Orientierung des Photos).

Führt man wieder die Segmentierungsstrategie durch, wird genau dieses Segment als unsicher ausgewiesen. Es wird automatisch versucht dieses Segment zu verbessern, was in diesem Beispiel zum Erfolg führt (vgl. Abb. 5-10).

(g) (h)

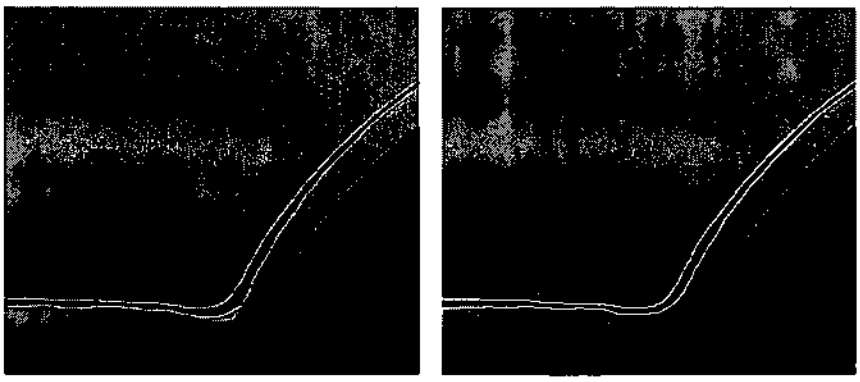


Abb. 5-10: Fortsetzung von Abb. 5-9: (g) Ein einzelnes Segment der *snake* wurde als unsicher ausgewiesen und wird optimiert. (h) Das endgültige Ergebnis hat die Ziellinie richtig extrahiert.

Der Ansatz der *twin snakes* eignet sich demnach besonders, um in Kombination mit der Segmentierungsstrategie (vgl. Abschnitt 5.4) eingesetzt zu werden. Die gemeinsame Anwendung der beiden Strategien erlaubt, es alle Schritte automatisch durchzuführen.

5.5.3 Spezialfall: *Twin Snakes* für die Extraktion einer einzelnen Kurve

Setzt man den gewünschten Abstand der beiden *snakes* auf Null, können im Spezialfall *twin snakes* auch dazu verwendet werden, dass sich zwei *snakes* einer einzigen Kurve von entgegen gesetzten Seiten annähern. In diesem Fall kann der Ansatz der *twin snakes* auch als Verallgemeinerung der dualen *snakes* angesehen werden (Gunn, Nixon, 1997). Darin werden zwei Ballone angesetzt, wobei einer mit Inflationskräften („Luft wird eingeblasen") modelliert wird, der andere mit Deflationskräften („Luft wird ausgelassen"). Während dieses Modell nur für geschlossene Kurven verwendet werden kann, können *twin snakes* auch auf offene Kurven angewendet werden.

Dieses Modell fügt dem einfachen *snake*-Modell die Vorteile der *twin snakes* hinzu. Die großen Stärken dieser Kombination sollen noch einmal zusammengefasst werden:

- Neben der Verringerung der Abhängigkeit von der Ausgangslage schafft dieses Modell die Möglichkeit zu einer verlässlichen Eigendiagnose, wie sie für den Einsatz von *snakes* in einem autonomen System Voraussetzung ist (vgl. Abschnitt 1.3).
- Die Selbstdiagnose arbeitet sehr zuverlässig, da überprüft wird, ob die beiden Zwillingspartner bei Annäherung von entgegengesetzten Seiten die selbe Kurve detektiert hat.
- Die Strategie findet garantiert das globale Minimum zwischen den Ausgangslagen der beiden *snakes.* Dadurch wird die Strategie unempfindlich auf lokale Minima.
- Der Prozess arbeitet automatisch. Alle notwendigen Entscheidungen werden selbstständig getroffen und Fehlentscheidungen im Nachhinein korrigiert.

6 Anwendungen in Photogrammetrie und Topographie

In diesem Kapitel sollen potenzielle Einsatzmöglichkeiten von *snakes* in der Praxis diskutiert werden.

Verschiedene Kriterien spielen eine Rolle, ob ein System in der Praxis akzeptiert wird. Dazu zählen nicht nur die Genauigkeit der Ergebnisse, sondern viel mehr die Zuverlässigkeit und die Vollständigkeit der gelieferten Ergebnisse (vgl. Abschnitt 1.3).

Ein weiteres Kriterium ist die Robustheit des Systems. Ist die Methode fähig mit Bildern verschiedener Belichtung und unterschiedlichen Kontrasts umzugehen? Ist die Methode unsensibel auf Rauschen? Konvergiert die Methode von unterschiedlichen Ausgangsdaten startend zur selben Lösung? Wie stark ändert sich das Ergebnis bei einer geringfügigen Variation der Parameter?

Einige Antworten auf diese Fragen wurden bereits im theoretischen Teil dieser Arbeit gegeben. Ausreichende Robustheit ist meist erst nach Anpassen des *snake*-Modells (der Optimierungsmethode und der Energiefunktion) an die spezifischen Eigenheiten einer Anwendung gegeben. Zudem ist im Allgemeinen der Einsatz einer speziellen Strategie sinnvoll. Parameter der Prozesse sollen so wenig wie möglich nötig sein. Die Parameter, die notwendig sind um den Prozess zu steuern, sollen anschaulich und interpretierbar sein.

In diesem Kapitel werden drei Anwendungsgebiete für *snakes* in der digitalen Photogrammetrie und Topographie vorgestellt. Zunächst wird untersucht, inwieweit sich das *snake*-Modell auf den dreidimensionalen Raum erweitern lässt. Dabei interessiert uns vor allem, wie eine im 3D modellierte *snake* in einem Verband mehrerer Photos mit bekannten Orientierungsparametern zueinander simultan die photometrische Energie in allen Bildern optimieren kann.

Die anderen beiden Abschnitte behandeln zwei spezielle Anwendungen, die im Rahmen der Dissertation bearbeitet wurden: die Schnittliniensuche bei der Orthophotomosaikierung und die Detektion von Kanten in topographischen Geländemodellen.

6.1 Dreidimensionale *Snakes*

Das klassische *snake*-Modell beschreibt eine univariate, ebene Kurve in einem zweidimensionalen Bild, wobei aber auch einige Verallgemeinerungen auf ein dreidimensionales Modell veröffentlicht wurden. Dabei müssen zwei Gruppen von 3D-Erweiterungen unterschieden werden.

Die erste Gruppe betrachtet die Oberfläche eines dreidimensionalen Körpers. Die Schnitte der Oberfläche mit Schichten eines Voxelmodells bilden sich als univariate, ebene Begrenzungskurven einer Region ab (z.B. Cohen, Cohen, 1993 oder Pardo, Radeva, 2000).

Im einfachen Ansatz kann das Modell der *snake* als univariates Modell bestehen bleiben. Dabei wird die Fähigkeit von *snakes* ausgenutzt, Kurven in einer Serie von Bildern zu verfolgen. Die *snake* verfolgt die Begrenzung der Region von einem Schichtbild zum nächsten. Die Ergebnisse der *snake* in jedem Schichtbild werden im Nachhinein zu einer Oberfläche kombiniert.

Wesentlich komplexer und rechenaufwendiger wird das Modell, wenn direkt die bivariate Oberfläche des Körpers modelliert wird. Dazu werden interne Energieterme für die Oberfläche im 3D-Raum angesetzt. Sie sorgen für eine glatte Form der Oberfläche. Die photometrische Energie wird für jeden Oberflächenpunkt aus dem entsprechenden Schichtbild abgeleitet.

Die zweite Gruppe der Erweiterungen des *snake*-Modells in den 3D-Raum ist die für Anwendungen der Photogrammetrie wesentlich interessantere. Dabei wird die *snake* als univariate, dreidimensionale Kurve modelliert, die in mehreren Bildern abgebildet ist. Während der Spezialfall der Abbildung in Stereobildpaaren von Forschern der *computer vision* und *pattern recognition* Gruppen mehrfach untersucht worden ist (Cham, Cipolla, 1997; Canero et al., 2000; Sbert, Sole, 2000), finden sich für den allgemeinen Fall beliebig vieler Bilder kaum Veröffentlichungen, obwohl dies das Spezialgebiet von Photogrammetern wäre.

Die Photogrammetrie beschäftigt sich mit der Rekonstruktion von Flächen und Kurven dreidimensionaler Objekte aus den Abbildungen in mehreren Photos. Nach der Ermittlung der Orientierungen der Photos (Lage und Stellung der Photos im Raum) sind die Transformationsgleichungen zwischen dem dreidimensionalen Koordinatensystem (Referenzsystem) und den lokalen Bildkoordinatensystemen bekannt (Kraus, 1993). Zur Bestimmung der Bildkoordinaten der Kurven bieten sich *snakes* als robuste Erfassungsmethode an.

Man kann sich wieder einen einfachen Ansatz zur Verwendung von *snakes* für die photogrammetrische Kurvenerfassung überlegen. Für die gesuchte dreidimensionale Kurve könnte man in jedem Photo eine eigene *snake* ansetzen. Jede *snake* könnte unabhängig von den anderen ihren Status optimieren. Schließlich könnte die 3D-Kurve aus den 2D-snakes rekonstruiert werden. Dieser Ansatz ignoriert allerdings die Abhängigkeit der 2D-Kurven voneinander.

Daher wird besser ein anderer Ansatz verfolgt. Die Kurve wird im 3D-Raum modelliert. Interne Energieterme beschreiben die ideale dreidimensionale Form der Kurve. Die photometrische Energie wird über alle Photos gemittelt, in denen ein Knoten der Kurve abgebildet ist. Die Energiefunktion enthält daher für all diese Photos einen photometrischen Energieterm. Während der Energieminimierung wird die Kurve im 3D-Raum so lange verformt und verschoben bis sich die Projektionen der Kurve in den Photos an die 2D-Zielkurven angenähert haben. Der beschriebene Ansatz wurde mittels Ausgleichungsrechnung in Li (1997) realisiert.

Damit eröffnet sich als Anwendungsmöglichkeit die gesamte photogrammetrische Linienvermessung. Während für Vegetationsgrenzen einzelne *snakes* modelliert

werden, kann für Straßen oder Flüsse die *twin snakes* Strategie vorteilhaft eingesetzt werden.

Der Arbeitsablauf besteht aus folgenden Schritten:

- Auswahl eines *snake*-Modells (Einzel-*snake* oder *twin snakes)*
- grobes Digitalisieren der Kurve in zumindest zwei Bildern. Mit diesen vereinfachten 2D-Kurven wird die Näherung der 3D-Freiformkurve durch Vorwärtsschnitt gebildet (Forkert et al., 1995)
- Energieoptimierung der *snake*
- Kontrollieren des Ergebnisses und gegebenenfalls Editieren einzelner Segmente in den Photos mit erneutem Vorwärtsschnitt

6.2 Schnittliniensuche für die Orthophotomosaikierung

Diese im Rahmen der Dissertation entwickelte Anwendung wurde im „ISPRS Journal of Photogrammetry and Remote Sensing" detailliert beschrieben (Kerschner, 2001). Ich möchte daher an dieser Stelle nur eine Zusammenfassung bringen und die Methode zu den in dieser Dissertation ausführlich behandelten theoretischen Hintergründen in Beziehung setzen.

Bei der Erstellung von Orthophotos für einen größeren Bereich müssen im letzten Schritt meist mehrere benachbarte und teilweise überlappende Orthophotos zu einem Mosaik vereinigt werden. Die Schnittlinie soll dabei so festgelegt werden, dass im Mosaik der Übergang von einem auf das nächste Orthophoto nicht erkennbar ist. Benachbarte Orthophotos sehen im Überlappungsbereich allerdings mehr oder weniger unterschiedlich aus. Dafür gibt es verschiedene Ursachen zumal die Herstellung eines Orthophotos aus vielen Schritten besteht, von denen jeder das Aussehen des Orthophotos bezüglich Farb-/Grauwerte und Geometrie (z.B. umgeklappte Objekte) beeinflusst.

Will man eine Schnittlinie durch den Überlappungsbereich ziehen, könnte man ausgeprägte Bildkanten verwenden, da an diesen ohnehin Unstetigkeiten in den Farb-/Grauwerten auftreten und daher der Übergang von einem auf ein anderes Bild nicht auffällt. Da aber nicht gewährleistet ist, durch den gesamten Überlappungsbereich einen Pfad entlang von Bildkanten zu finden, wollen wir hier ein anderes Kriterium für die optimale Schnittlinie einsetzen: die beiden Bilder sollen sich entlang der Schnittlinie möglichst wenig voneinander unterscheiden. Bringt man diese Forderung in eine mathematische Formulierung, kann sie in der Energiefunktion einer *snake* umgesetzt werden. Auf diese Weise kann eine *snake* dazu eingesetzt werden, den Pfad der kleinsten Differenz bezüglich einer Kombination verschiedenster Kriterien zu detektieren.

Die aus dem Vergleich der beiden Orthophotos abgeleiteten Forderungen an die Schnittlinie beinhalten:

- eine möglichst hohe Ähnlichkeit der Farbwerte der beteiligten Orthophotos (vor allem im Farbton und der Farbintensität) und
- eine möglichst hohe Ähnlichkeit der Textur: Objekte auf der Geländeoberfläche, die in dem Modell, das zur Orthophotoumbildung verwendet worden ist, nicht enthalten sind (z.B. Bäume), bleiben im Orthophoto geometrisch verzerrt. Sie bewirken eine stark ausgeprägte Textur in radialer Richtung. Diese radiale Richtung hat in benachbarten Photos immer unterschiedliche Orientierung. Daher soll die Schnittlinien-*snake* solche Gebiete starker radialer Textur meiden.

Da diese beiden Kriterien aus den Photos abgeleitet werden, werden diese Forderungen in den photometrischen Termen formuliert. Bevor die Energieoptimierung startet, werden die Energiewerte für jedes Pixel des Überlappungsbereichs der beiden Orthophotos vorausberechnet in einem eigenen Energiebild gespeichert.

Zur Suche der optimalen Schnittlinien werden drei der in Kapitel 5 beschriebenen Anwendungsstrategien eingesetzt: *twin snakes* mit Sollabstand Null (Abschnitt 5.5.3) in einer hierarchischen Annäherung (Abschnitt 5.1) und mit automatischer Optimierung einzelner Segmente (Abschnitt 5.4).

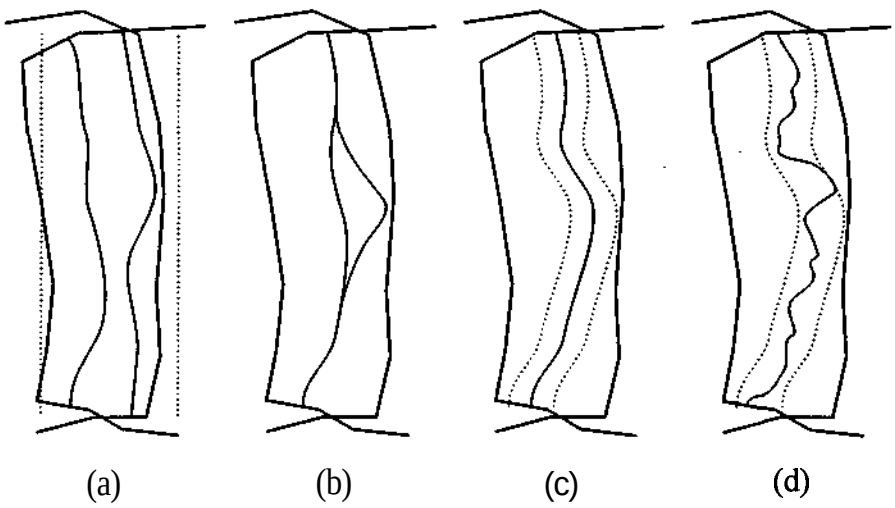


Abb. 6-1: Hierarchische Detektion der optimalen Schnittlinie: (a) punktierte Geraden: die Ausgangslage der *twin snakes* an den Rändern des Überlappungsbereichs der Orthophotos; durchgezogene Kurven: Status der *twin snakes* während der Energieoptimierung; (b) die beiden *snakes* haben abgesehen von einem Segment in der Mitte denselben Status erreicht; (c) nach der Optimierung einzelner Segmente wurde die optimale Kurve auf dem obersten Pyramidenniveau erreicht; punktiert: die Ausganglage der *twin snakes* für das nächst niedrigere Pyramidenniveau; (d) nach dem Wechsel auf das niedrigere Pyramidenniveau wurde die optimale Schnittlinie bestimmt (aus Kerschner, 2001).

Die Ausgangspositionen der *twin snakes* liegen an den beiden entgegengesetzten Rändern des Überlappungsbereichs (vgl. Abb. 6-1). Durch die gegenseitige Anziehung bewegen sie sich im Zuge der Energieminimierung aufeinander zu. Wenn die

beiden Zwillingspartner in allen Segmenten übereinstimmen, ist die optimale Schnittlinie gefunden.

Dieser Prozess wird zunächst auf dem obersten Niveau einer Bildpyramide durchgeführt, die aus dem Energiebild abgeleitet wurde. Nachdem die Energieoptimierung auf diesem Niveau abgeschlossen ist, wird auf das nächst niedrigere Niveau der Energiebild-Pyramide gewechselt. Dabei wird aus der Kurve des höheren Niveaus ein Band gebildet, dessen Begrenzungen auf das nächst niedrigere Niveau projiziert werden und als neue Ausgangspositionen verwendet werden.

Die hierarchische Annäherung ist in diesem Zusammenhang nicht nur praktisch notwendig (die beiden *snakes* würden bei voller Auflösung viel zu lange brauchen und zu viele lokale Minima zu überwinden haben, bis sie zusammengefunden haben), sondern auch theoretisch gerechtfertigt: Die höheren Pyramidenniveaus enthalten gemittelte Energiewerte aus den tieferen Niveaus. Der grobe Verlauf der Schnittlinie wird auf einem sehr grob aufgelösten Niveau mit gemittelten Energiewerten bereits festgelegt. Dabei werden großräumig Bereiche gesucht, in denen die Energie am niedrigsten ist. Der feine Verlauf innerhalb dieser groben Rasterzellen ergibt sich dann sukzessive durch Hinabsteigen in der Bildpyramide.

Die präsentierte Methode arbeitet automatisch und ohne jegliche manuelle Interaktion. Da in das Modell der *twin snakes* mit Sollabstand Null auch eine Selbstdiagnose integriert ist (vgl. Abschnitt 5.5.3), kann die Methode auch in einem autonomen System (nach der Definition von Abschnitt 1.3) eingesetzt werden. Voraussetzung dafür ist lediglich, dass die Kriterien, nach denen ein Operateur das Ergebnis als optimal einstufen würde, vollständig in der Energiefunktion berücksichtigt werden. Ein Ergebnis ist in Abb. 6.2 dargestellt.

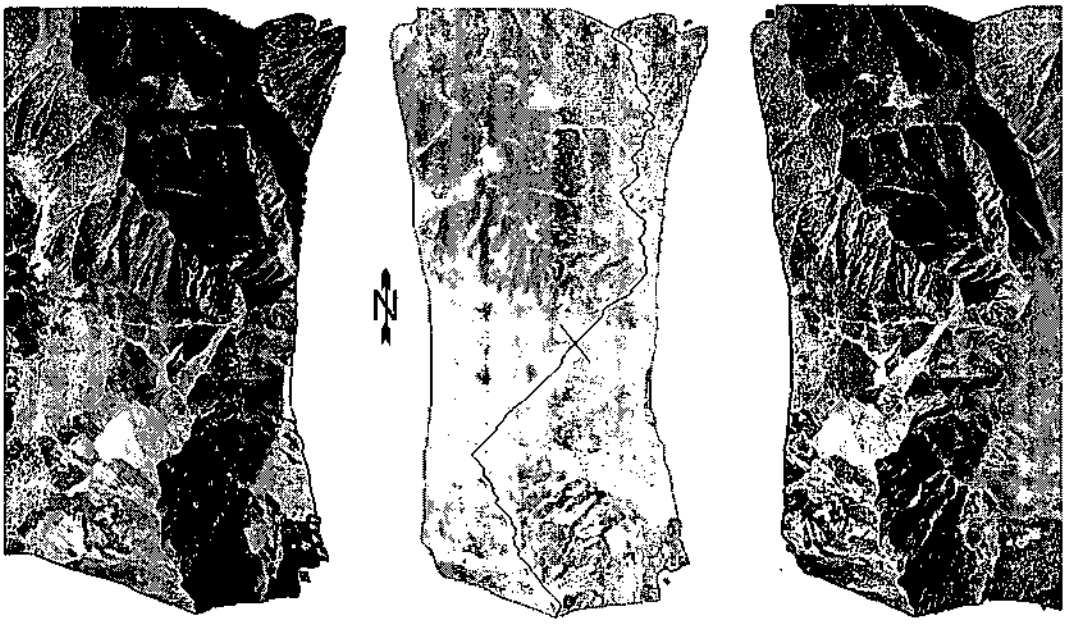


Abb. 6.2: Schnittliniensuche im Überlappungsbereich zweier benachbarter Orthophotos (links und rechts). Mitte: Das für die *snake*-Optimierung verwendete Energiebild (je heller, desto geringer ist die Energie). In diesem Energiebild ist die von den beiden *snakes* gefundene optimale Schnittlinie eingezeichnet.

Farbabbildungen zu diesem Beispiel können in hoher Auflösung auf der Internet-Seite http://www.ipf.tuwien.ac.at/mk/mosaic/mosaicking.htm betrachtet werden. Die Rechenzeit, um in diesem Überlappungsbereich von 2100 x 3800 Pixeln die Schnittlinie zu finden, betrug etwa drei Minuten mit einer Prototypimplementierung (Pentium II, 266 MHz, 128 Mb RAM). Die meiste Zeit benötigte die Bildverarbeitung, d.h. die Ableitung des Energiebildes. Aufgrund der hierarchischen Energieoptimierungsstrategie war weniger als eine Minute notwendig um die endgültige Lage der Schnittlinie festzulegen.

Verwendet man nach der beschriebenen Methode *snakes* zur Detektion von Schnittlinien in Mosaiken von Orthophotos, nützt man die Fähigkeit von *snakes,* die global optimale Kurve zu finden. Diese Anwendung zeigt,

- dass sich eine durch unterschiedliche Kriterien definierte „Optimalität" in eine geeignete Formulierung der Energieterme bringen lässt und
- dass *snakes* unter Einsatz spezieller Strategien in manchen Anwendungen autonom arbeiten können.

6.3 Ableitung von geomorphologischen Strukturlinien

Die Qualität eines auf einem Raster basierenden DTMs *(digital terrain model,* digitales Geländemodell) hängt entscheidend von der Integration geomorphologischer Strukturinformation ab. Man spricht bei solchen Modellen von hybriden DTMs (Kraus, 2000, Abschnitt H 3.4). Zusätzlich zu den in einem regelmäßigen Raster vorliegenden Höhendaten werden weitere punktbezogene oder linienbezogene Informationen verspeichert. Als punktbezogene Strukturinformation gelten Kuppen- oder Muldenpunkte. Linienhafte Strukturinformation sind etwa Formlinien *(form lines)* oder Kanten *(fold lines).* Erstere modellieren eine starke Geländekrümmung quer zur Formlinie, zweitere erlauben Knicke in der modellierten Geländeoberfläche.

Im Folgenden wird untersucht, inwieweit sich *snakes* dazu einsetzen lassen, automatisiert linienhafte Strukturinformation aus Einzelpunkt-Massendaten abzuleiten. Dies wird umso eher möglich sein, je dichter die topographischen Daten vorhanden sind. Die Aufgabe ist insofern ein aktuelles Forschungsthema als durch moderne automatische Erfassungsmethoden (LIDAR, Bildkorrelation: Kraus, 2000, Abschnitte I 2.2 und I 2.1.3.2 c) Daten in sehr hoher Dichte, allerdings ohne Strukturinformation anfallen. Die nachträgliche Ableitung linienhafter Strukturelemente bringt zwei entscheidende Vorteile:

- Qualitätssteigerung des DTMs: Knicke im Gelände lassen sich nur durch Einbeziehung von Kanten in das DTM exakt modellieren.
- Verringerung der Datenmenge: Die topographische Information ist für wenig bewegtes Gelände in den Punktdaten redundant verspeichert. Zwar ist die hohe Punktdichte erforderlich um Geländeknicke wenigstens näherungsweise repräsentieren zu können, doch würde für den Großteil des Geländes auch eine geringere Punktdichte ausreichen.

Das Datenvolumen des DTMs kann entscheidend verkleinert werden, wenn man ein hybrides Geländemodellierungsverfahren einsetzt. Wegen der zusätzlich verspeicherten qualitativen Strukturinformation muss die Dichte der regelmäßigen Rasterpunkte nun nicht mehr der Dichte der Ausgangsdaten entsprechen ohne dass man Einbußen des Detaillierungsgrads in Kauf nehmen muss.

Geomorphologische Strukturlinien werden bis dato meist manuell, photogrammetrisch identifiziert und ausgewertet (Kraus, 2000, Abschnitt I 2.1.3.1). Erste Versuche zur Automatisierung dieses zeitaufwändigen Arbeitsschritts werden im folgenden Abschnitt präsentiert. Anschließend wird untersucht, inwieweit sich *snakes* für eine halbautomatische Messung von Geländekanten eignen.

6.3.1 Ansätze zur automatisierten Kantenextraktion

Je nach Erfassungsmethode fallen die Daten in einer mehr oder weniger regelmäßigen Anordnung an. Während Daten aus Bildkorrelation meist im quadratischen Raster abgeleitet werden, trifft dies für **LIDAR-Daten** nicht zu. Die Ansätze zur automatisierten **Detektion** von geomorphologischen Kanten in topographischen Punktdaten lassen sich demnach in zwei Gruppen unterteilen:

1. Detektion in unregelmäßigen Punktanordnungen
2. Detektion in einem regelmäßigen Raster

Als Vertreter der ersten Gruppe soll eine **Flächenmodellierungsmethode** erwähnt werden (Briese et **al.**, 2002). Dabei werden Geländekanten *(fold lines)* lokal durch den Schnitt eines gleitenden Ebenenpaares modelliert. Die Ebenen werden aus den topographischen Daten rechts und links der Kante geschätzt. Die Methode kann Kanten mit großer Genauigkeit extrahieren, benötigt allerdings gute Näherungswerte.

Gemäß der Unterteilung am Beginn des Kapitels 2 zählt diese Methode zu den Regionen basierten Ansätzen. Dabei wird versucht, Pixel mit gleicher Neigung zu Regionen zusammenzufassen, deren Grenzen Kanten sein können (aber nicht müssen, wenn der Knick nicht ausgeprägt genug ist). Umgekehrt geht man bei Kanten basierten Methoden von vornherein nur auf stark ausgeprägte Kanten los.

Methoden der zweiten Gruppe setzen ein regelmäßiges Raster von Punkten voraus. In der Bildverarbeitung spricht man dann meist von Distanzbildern *(range images),* die in jedem Pixel die Distanz zu einem Objekt gespeichert haben. Falls diese Bilder in polarer Geometrie vorliegen (z.B. aus Laserscanner-Daten abgeleitete Bilder), sollten sie in ein **kartesisches** Koordinatensystem transformiert werden, damit anschließende Neigungs- oder Krümmungsanalysen plausibel sind.

In Distanzbildern sind Kanten zu extrahieren, wobei es sich bei den hier gesuchten Kanten nicht um step *edges* gemäß der Definition in 1.4.1 handelt, sondern um Begrenzungslinien von Flächen gleicher Neigung. Kanten treten nicht nur zwischen Flächen unterschiedlicher Steilheit auf, sondern auch zwischen Flächen gleicher Steilheit aber unterschiedlicher Exposition (siehe Abb.6-3).

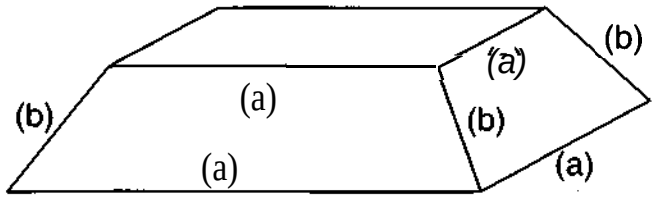


Abb. 6-3: Plateau mit zwei Typen von Geländekanten: Kanten vom Typ (a) trennen Flächen unterschiedlicher Steilheit. Kanten vom Typ (b) trennen Flächen gleicher Steilheit, aber unterschiedlicher Orientierung der Flächennormalen.

Die in den letzten Jahren publizierten Raster basierten Kantenextraktionsverfahren (Gomes Pereira, Wicherson, 1999; Gomes Pereira, Janssen, 1999) wenden einfache Kantenextraktionsalgorithmen auf das Distanzbild an und beschränken sich auf Kantenmodelle einer abrupten Änderung der Steilheit des Geländes ohne die Orientierung der Neigung zu berücksichtigen. Brügelmann (2000) verwendet die Quadratsumme der Hauptkrümmungen der Oberfläche als Maß für die Homogenität einer Region. Beyer (2003) schätzt die Krümmung des Geländes aus *wavelet*-Koeffizienten ab, welche ebenfalls zur Kantendetektion verwendet werden könnten.

6.3.2 Schwierigkeiten der automatisierten Kantenextraktion

Die Punktwolke, in der geomorphologische Strukturen zu suchen sind, kann als diskretes Modell der topographischen Oberfläche gesehen werden. In diesem werden im Allgemeinen Extrema der Krümmung gesucht. Zur Bestimmung der Krümmung muss die zweite Ableitung der Oberfläche gebildet werden. Die Probleme dabei sollen nicht unerwähnt bleiben:

- Bezüglich der Genauigkeit von zweiten Ableitungen, die aus diskreten Daten (in unserem Fall einer diskreten Repräsentation der Geländeoberfläche) abgeleitet werden, will ich auf Anhang A verweisen. Zufällige Fehler in den Daten werden demgemäß schon in der ersten Ableitung verstärkt. Erst recht in der zweiten Ableitung treten sie in vielfacher Stärke auf.
- Aus einem diskretisierten Modell können Kanten nur dann getrennt voneinander abgeleitet werden, wenn sie dem Abtasttheorem genügen (Kraus, 2000). Zusätzlich muss berücksichtigt werden, dass die Stärke der aus diskreten Daten geschätzten zweiten Ableitung mit steigender Diskretisierungsweite stark abnimmt (vgl. Anhang A). Dies wird auch anhand der Abbildung 6-4 klar.

Daraus folgt, dass Kanten nur in sehr dichten Daten gefunden werden können. Je größer der Punktabstand, desto niedriger müssen beispielsweise Schwellwerte angesetzt werden und desto größer ist die Gefahr von Fehlklassifikationen.

Dicht aufgenommene Daten des Geländes stammen meist aus einer automatischen Erfassungsmethode. Sie enthalten im Allgemeinen viele Punkte, die nicht Teil der Geländeoberfläche sind (Vegetation, künstliche Objekte auf der Oberfläche etc.).

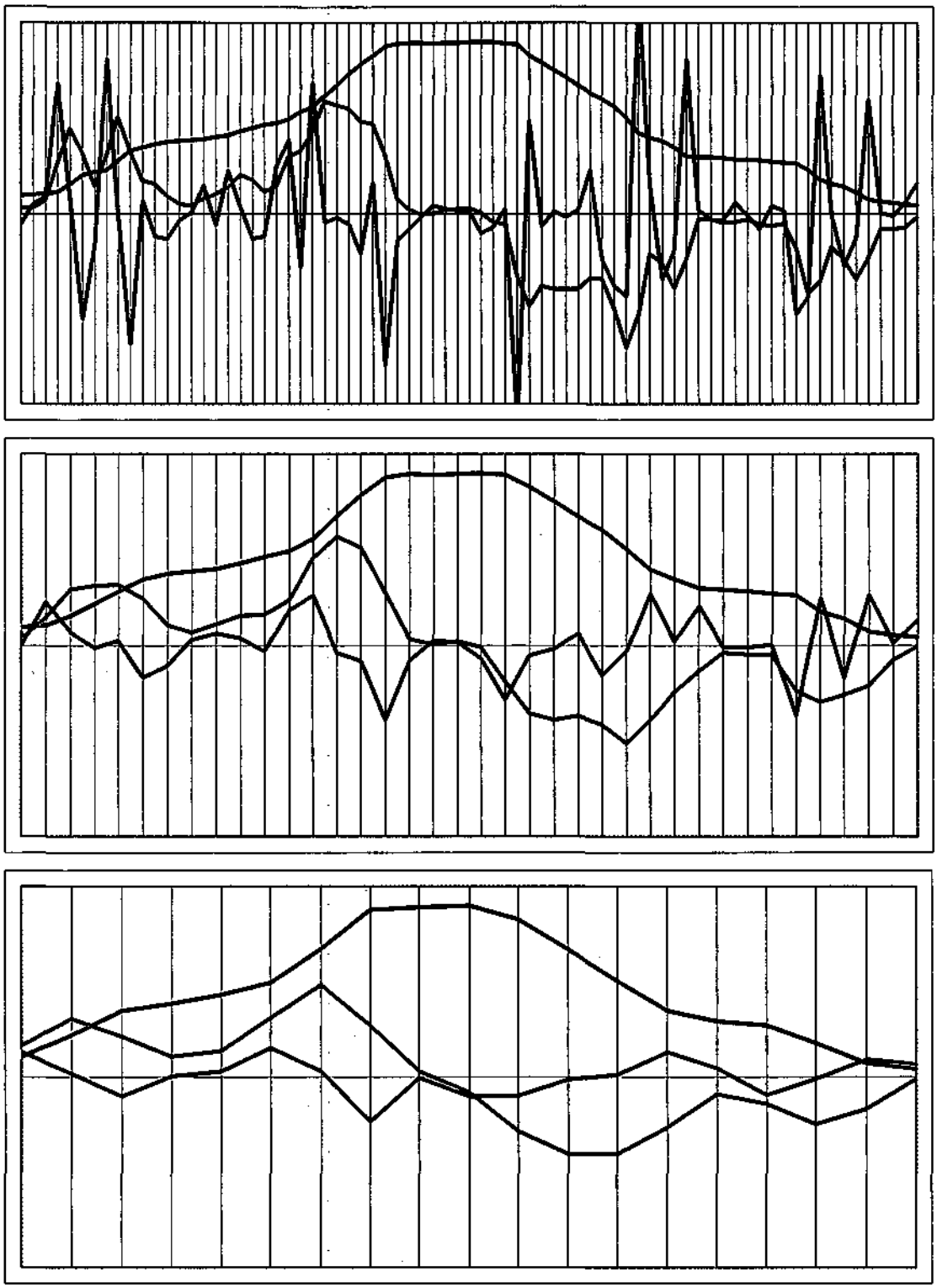


Abb. 6-4: Ein Querprofil zu einem Damm (Beispiel aus dem Nationalpark March-Donau-Auen). Das Höhenprofil ist als schwarze Linie dargestellt. Die Länge des Profils beträgt etwa 35 m, der Höhenunterschied etwa 4.30 m. Das **Diskretisierungsintervall** beträgt 0.5 m (oben), 1 m (Mitte) bzw. 2 m (unten). Der rote Polygonzug zeigt die aus den diskreten Daten geschätzten ersten Ableitungen, der blaue Polygonzug die zweiten Ableitungen. Es ist deutlich zu erkennen, dass die Stärke der geschätzten zweiten Ableitung mit größerer Diskretisierungsweite deutlich abnimmt.

Eine Kantenextraktion in diesen Originaldaten würde viele unerwünschte Kanten liefern und auch die gewünschten Kanten verfälschen. Daher muss ein Extraktionsalgorithmus in einem gefilterten Geländemodell arbeiten, in dem Häuser, Bäume, Büsche und andere Objekte entfernt worden sind (zur Erzeugung gefilterter Modelle siehe Kraus, 2000, Kapitel H 3).

Eine Filterung ist auch aufgrund des ersten Punktes (siehe oben) notwendig. Klarerweise wird durch eine starke Filterung das Signal (die Kante) ebenfalls geglättet. Kanten können daher nicht mehr detektiert werden sondern lediglich Bereiche mit auffallend hoher Krümmung.

6.3.3 Geländekanten mit *Snakes*

Wie in Abschnitt 1.3 festgestellt, führen automatische Extraktionsverfahren im Allgemeinen zu Fehlern, die folgenden beiden Typen zugerechnet werden können: mangelhafte Extraktion oder falsche Extraktion. Auch bei der Extraktion von Geländekanten müssen Ergebnisse von automatischen Prozessen überprüft und nachbearbeitet werden, um Lücken zu schließen und überflüssige Kanten zu entfernen. Folglich des in Abschnitt 1.3 aufgestellten Paradigmas, nur dann vollautomatische Prozesse einzusetzen, wenn sie fast ausschließlich richtige und vollständige Ergebnisse liefern, und als Konsequenz der im letzten Abschnitt angesprochenen Probleme soll auch für die Geländekantensuche ein halbautomatischer Ansatz entwickelt werden.

Die Methode wird anhand von Daten eines Laser-Scanner-Flugs der Firma TopoSys im Nationalpark March-Donau-Auen (Niederösterreich) getestet. Die Daten wurden mit einer unregelmäßigen Dichte mit 15 cm Punktabstand in einer Richtung und 1.5 m in der anderen Richtung erfasst (Kölbl et al., 2002). Aus diesen Daten wurde ein regelmäßiges Raster mit Gitterweite 1 m abgeleitet.

Die *snake*-Methode bietet sich zur Erfassung von geomorphologischen Kanten an (Rieger et al., 1999). Gründe dafür sind:

- Robustheit: Wegen der erwähnten Verstärkung der Datenungenauigkeit in den geschätzten Ableitungen sind robuste Methoden zur Extraktion von Kanten gefragt. In Abschnitt 2.3 wurde erläutert, dass *snakes* besser geeignet sind in verrauschten Daten Linien zu finden, als klassische Kantenverfolgungsalgorithmen.
- Möglichkeit der Integration verschiedener an die Aufgabe angepasster Energieterme. Geeignete Energieterme sollen nun in diesem Abschnitt hergeleitet werden.

Der *snake*-Ansatz reiht sich bei den Kanten basierten Methoden ein. Im Gegensatz zu Regionen basierten Ansätzen werden nur lokale Charakteristika von Geländekanten verwendet. Diese können als bildbezogene Energieterme formuliert werden. Zunächst wollen wir die Charakteristika von Geländekanten analysieren um anschließend die passenden Energieterme zu formulieren.

Kanten im Gelände können zwar natürlicher Herkunft sein, doch treten sie in weitaus größerem Maß in Bereichen des vom Menschen künstlich angepassten Geländes

auf. In Abb. 6-5 ist ein künstlich angelegter Damm in den March-Donau-Auen skizziert. Betrachtet man die Vielzahl von Kanten, kann man folgende Eigenschaften festhalten:

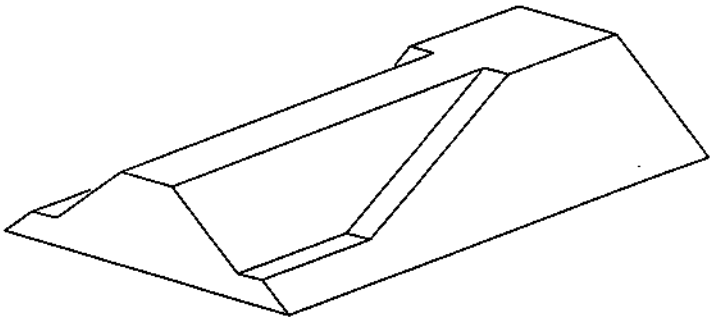


Abb. 6-5: Schematische Darstellung der Kanten eines künstlichen Damms im Nationalpark March-Donau-Auen. Das Querprofil im Vordergrund entspricht dem Profil von Abb. 6-4.

- Kanten bewirken einen Knick in Profilen quer zur Kante bzw. eine Unstetigkeitsstelle der ersten Ableitung in diesen Profilen.
- Die Geländeoberfläche hat meist geringe Krümmung in Richtung der Kante.
- Viele, speziell künstliche Geländekanten haben entlang der gesamten Kurve einen nur gering gekrümmten Verlauf.

Die ersten beiden Kriterien betreffen lokale Eigenschaften der Oberfläche. Diese Eigenschaften werden als Bildterme in die Energiefunktion aufgenommen. Die dritte Eigenschaft betrifft den Verlauf der Kurve, für den die passenden Energieterme bereits im klassischen *snake*-Ansatz enthalten sind.

Interne Energieterme:

Das dritte Kriterium entspricht der Denkweise der internen Energieterme. In den internen Termen des ursprünglichen *snake*-Konzeptes (4-7) ist bereits die Forderung einer geringen Krümmung entlang der Kurve formuliert, allerdings nur für zweidimensionale *snakes.* In Analogie zum Krümmungsterm der Grundrisskurve der Kanten-*snake* (4-11) kann ein zusätzlicher Krümmungsterm in der Höhe angesetzt werden:

$$E^{z}_{curv} = \left| \frac{z_{i-1} - z_i}{|\mathbf{v}_{i-1} - \mathbf{v}_i|} - \frac{z_i - z_{i+1}}{|\mathbf{v}_i - \mathbf{v}_{i+1}|} \right|^2 \tag{6-1}$$

Dabei werden die z-Werte der Knoten nicht in der Kurve modelliert sondern aus dem DTM durch Interpolation bestimmt.

Es wäre auch denkbar anstelle eines x/y-Krümmungsterms und eines separaten z-Krümmungsterms nur eine einzige räumliche Krümmung in der Energiefunktion zu modellieren. Ich bevorzuge die getrennten Terme, da dann auch unterschiedliche Gewichte für die verschiedenen Krümmungsterme vergeben werden können.

Der kontinuierliche Verlauf der *snake* in der Höhe hilft, das Springen der *snake* zwischen zwei benachbarten Kanten zu unterbinden. Dies ist wichtig, da Gelände-

kanten oft in doppelter, paralleler Anordnung auftreten (Böschungsoberkante und -unterkante) oder überhaupt in vielfacher paralleler Anordnung (vgl. Abb. 6-6).

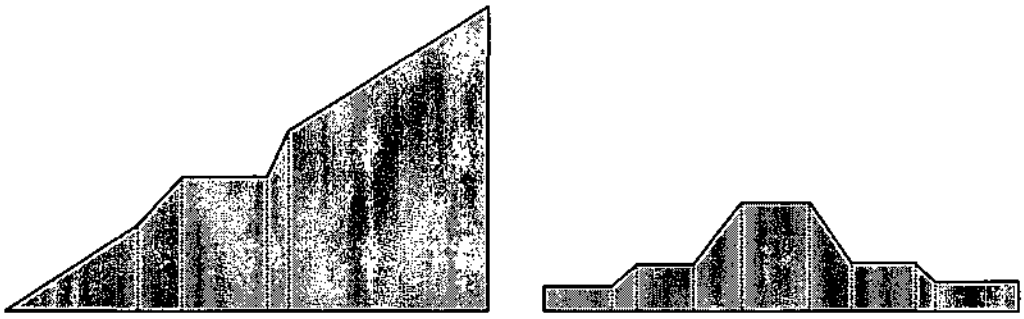


Abb. 6-6: Vier Geländekanten in einem (überhöhten) Querprofil einer in den Hang gebauten Straße und acht Kanten in einem Querprofil eines Damms (vgl. Abb. 6-5).

Bei dieser speziellen Anordnung von Bruchkanten empfiehlt sich die mehrfache Anwendung der Strategie der *twin snakes* (vgl. Abschnitt 5.5.2).

Bildterme:

Die ersten beiden Kriterien sollen als Bildterme in die Energiefunktion aufgenommen werden. Dabei entspricht dem Bild das diskrete Höhenmodell, also ein Distanzbild *(range image).*

Im diskreten Höhenmodell sind Unstetigkeiten der ersten Ableitung, wie sie als primäre Eigenschaft von Kanten erkannt wurden, allerdings nicht enthalten. Es kann lediglich ein lokales Extremum in der geschätzten zweiten Ableitung festgestellt werden. Die ersten beiden Kriterien sollen daher für diskrete Oberflächenmodelle umformuliert werden:

1. Der Absolutbetrag der Krümmung quer zur Kante ist hoch.
2. Die Krümmung entlang der Kante ist nahe Null.

Die Krümmung der geschätzten Oberfläche spielt demnach eine zentrale Rolle. Auch die in Abschnitt 6.3.1 erwähnten Verfahren basieren auf der Krümmung. In der Differentialgeometrie kennt man verschiedene Arten von Krümmung. In unserem Fall bieten sich die beiden Hauptkrümmungen (die minimale und maximale Krümmung aller Normalschnittkurven) als Schätzungen der Krümmung entlang und quer zur Kante an.

Wir gehen von der folgenden Formulierung einer Oberfläche aus:

$$\mathbf{x}(u,v) = \begin{pmatrix} u \\ v \\ z(u,v) \end{pmatrix} \tag{6-2}$$

Es handelt sich dabei um eine Oberfläche, bei der für jedes Paar der Parameter *u* und v genau ein z-Wert gegeben ist. Setzt man für *u* = *x* und v = *y,* erhält man eine gebräuchliche Oberflächenmodellierung des Geländes, die allerdings keine Überhänge modellieren kann. Man spricht in diesem Fall vielfach von einer 2.5-D Repräsentation des Geländes. Für echte dreidimensionale Oberflächen müssen aufwendigere Modelle angesetzt werden (Pfeifer, 2002). Für unsere Zwecke reicht diese Form von Oberfläche.

Um die in der Differentialgeometrie definierten Hauptkrümmungen ableiten zu können, bestimmen wir zunächst die ersten und zweiten Ableitungen der Oberfläche:

$$\mathbf{x}_u(u,v)=\begin{pmatrix}1\\0\\z_u(u,v)\end{pmatrix} \qquad \mathbf{x}_v(u,v)=\begin{pmatrix}0\\1\\z_v(u,v)\end{pmatrix}$$
$$\mathbf{x}_{uu}(u,v)=\begin{pmatrix}0\\0\\z_{uu}(u,v)\end{pmatrix} \quad \mathbf{x}_{uv}(u,v)=\begin{pmatrix}0\\0\\z_{uv}(u,v)\end{pmatrix} \quad \mathbf{x}_{vv}(u,v)=\begin{pmatrix}0\\0\\z_{vv}(u,v)\end{pmatrix} \tag{6-3}$$

Aus den ersten Ableitungen lassen sich die Koeffizienten der ersten Fundamentalform ableiten:

$$\begin{aligned} E &= \mathbf{x}_u\mathbf{x}_u = 1+z_u^2 \\ F &= \mathbf{x}_u\mathbf{x}_v = z_u z_v \\ G &= \mathbf{x}_v\mathbf{x}_v = 1+z_v^2 \end{aligned} \tag{6-4}$$

Der Normalvektor auf die Oberfläche lautet:

$$\mathbf{n}=\frac{\mathbf{x}_u\times\mathbf{x}_v}{\|\mathbf{x}_u\times\mathbf{x}_v\|}=\frac{1}{\sqrt{1+z_u^2+z_v^2}}\begin{pmatrix}-z_u\\-z_v\\1\end{pmatrix} \tag{6-5}$$

Bezeichnet man den Normierungsbruch mit *k,* ergeben sich die Koeffizienten der zweiten Fundamentalform zu:

$$\begin{aligned} L &= \mathbf{x}_{uu}\mathbf{n} = kz_{uu} \\ M &= \mathbf{x}_{uv}\mathbf{n} = kz_{uv} \\ N &= \mathbf{x}_{vv}\mathbf{n} = kz_{vv} \end{aligned} \tag{6-6}$$

Nun sind alle Voraussetzungen gegeben um die Hauptkrümmungen κ_1 und κ_2 der Oberfläche aus der folgenden quadratischen Gleichung zu berechnen:

$$\kappa^2-\frac{EN-2FM+GL}{EG-F^2}\kappa+\frac{LN-M^2}{EG-F^2}=0 \tag{6-7}$$

Die Lösung dieser Gleichung ist trivial. Die Krümmungen können sowohl positive wie auch negative Werte annehmen. Für unsere Anwendung sind die Absolutbeträge der

Krümmungen relevant. Die betragsmäßig größere der beiden Hauptkrümmungen soll mit κ_{max}, die kleinere mit κ_{min} bezeichnet werden:

$$\begin{aligned} \kappa_{max} &= \max(|\kappa_1|,|\kappa_2|) \\ \kappa_{min} &= \min(|\kappa_1|,|\kappa_2|) \end{aligned} \tag{6-8}$$

Diese beiden Hauptkrümmungen sollen nun in den Energietermen verwendet werden. Die größere der beiden Hauptkrümmungen (κ_{max}) soll **maximiert** werden, während die kleinere (κ_{min}) klein bleiben soll. Es bietet sich folgender Energieterm an:

$$E_\kappa = \kappa_{min} - \kappa_{max} \tag{6-9}$$

Je größer κ_{max} im Vergleich zu κ_{min}, desto geringer wird der Energiewert. Die Energiewerte können für jeden Rasterpunkt des Geländes vorausberechnet werden: Aus einem Geländemodell mit der Parameterisierung der Oberfläche $u = x$ und $v = y$ werden die Hauptkrümmungen abgeleitet. Dazu werden die Ableitungen der Oberfläche $z(x,y)$ aus den Differenzenquotienten geschätzt. Die geschätzten Hauptkrümmungen können geeignet skaliert und in 256 Stufen **diskretisiert** werden um sie in einem digitalen Bild (Energiebild) speichern zu können. In Abb. 6-7 sind Bilder der beiden Hauptkrümmungen für ein Beispiel in den March-Donau-Auen dargestellt. Die ersten Ableitungen wurden aus einem gefilterten Geländemodell mit dem Softwaresystem SCOP (Modul *slope;* SCOP, 2003) erzeugt. Die zweiten Ableitungen wurden ausgehend von diesen Werten berechnet. Man erkennt deutlich die vielen parallelen Kanten des Dammes sowie die Auffahrten.

Zu jeder dieser Hauptkrümmungen lässt sich auch die Hauptkrümmungsrichtung bestimmen. Sie gibt die Orientierung desjenigen Normalschnitts **an**, für den die Krümmung das Extremum annimmt. Die Hauptkrümmungsrichtungen sind die Lösungen des linearen Gleichungssystems:

$$\left(\begin{pmatrix} L & M \\ M & N \end{pmatrix} - \kappa \begin{pmatrix} E & F \\ F & G \end{pmatrix}\right)\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{6-10}$$

Analog zur Überlegung in Abschnitt 4.2.3 (Einbeziehung der Richtung des Gradienten in die Formulierung der photometrischen Energie) kann man fordern, dass die *snake* normal auf die Hauptkrümmungsrichtung $\mathbf{k}_{max}$ der maximalen Hauptkrümmung κ_{max} verlaufen soll. Ähnlich zu Gleichung (4-18) schlage ich vor, das mit der Länge aus (6-9) multiplizierte innere Produkt zweier Vektoren zu verwenden: einerseits des Vektors in Hauptkrümmungsrichtung $\mathbf{k}_{max}$ für die Position $\mathbf{v}(s)$ und andererseits des Normalvektors n auf die *snake* an der Stelle *s:*

$$E_\kappa(\mathbf{v}) = (\kappa_{min}(\mathbf{v}) - \kappa_{max}(\mathbf{v}))|\mathbf{k}_{max}(\mathbf{v}) \cdot \mathbf{n}(s)| \tag{6-11}$$

In Abb. 6.8 sieht man Beispiele, wie *snakes* die Böschungskanten des Damms aus Abb. 6.7 suchen.

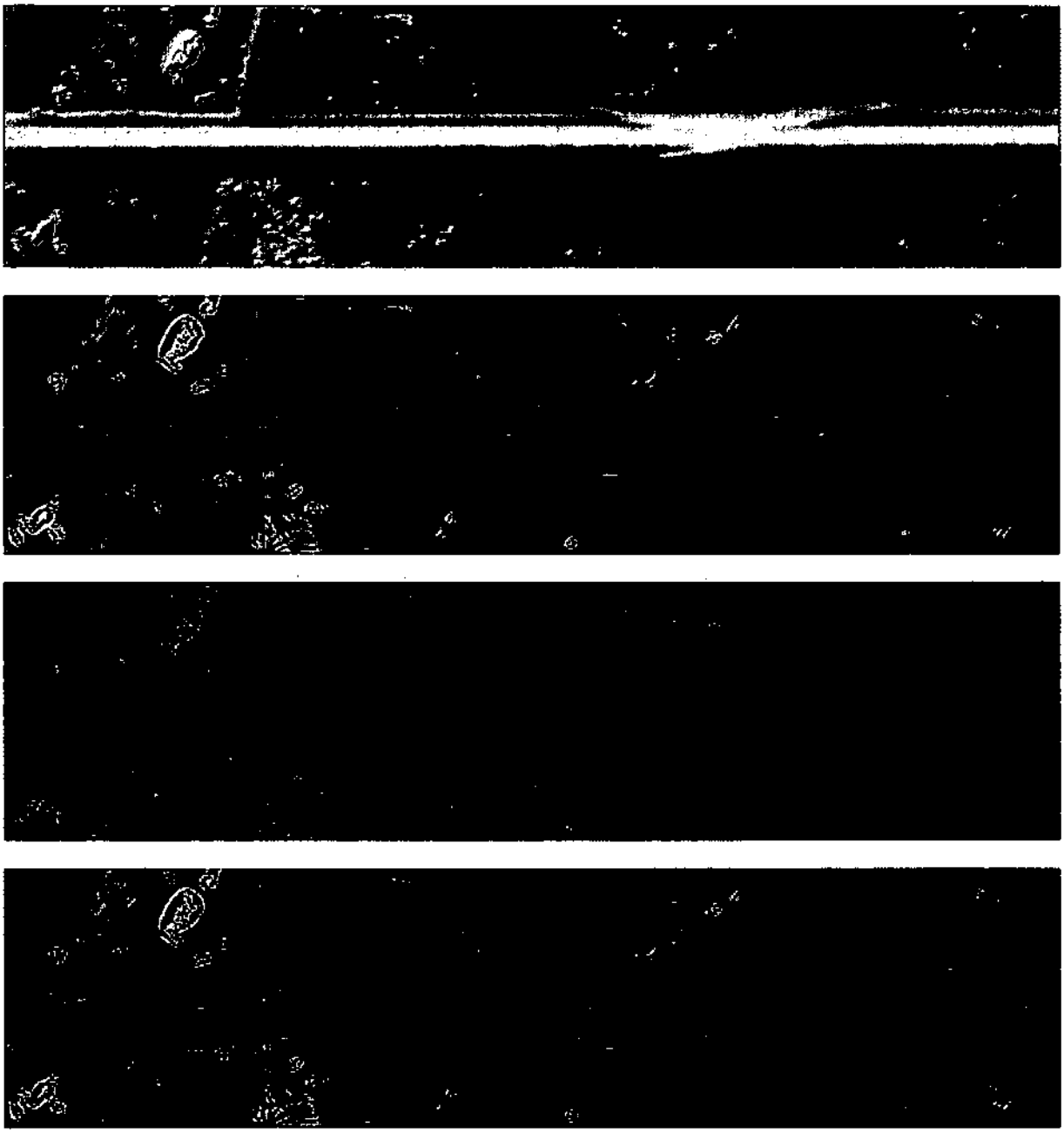


Abb. 6-7: Von oben nach unten: (1) Geländemodell aus dichten LIDAR Daten im Nationalpark March-Donau-Auen, (2) maximale Hauptkrümmung, (3) minimale Hauptkrümmung, (4) Differenz der Hauptkrümmungen.

Die Beispiele zeigen viel versprechende Erfolge. In Geländemodellen, die durch Filterung erzeugt wurden, können Linien starker Krümmung detektiert werden. Im Gegensatz zur Schnittliniensuche in Orthophotomosaiken ist die Suche von Kandidaten für Kanten als interaktive Anwendung konzipiert. Der Operateur legt mittels weniger Punkte den groben Verlauf einer Kurve fest, wo er eine Geländekante vermutet. Die *snake* bestimmt den detaillierten Verlauf der Kurve. Das Ergebnis der *snake* kann nun als Ausgangslage für eine genaue Bestimmung über gleitende Ebenenpaare (Briese et al., 2002) herangezogen werden. Zur Bestätigung dieser Vorgehensweise und dieses Ansatzes sind noch weitere Tests erforderlich.

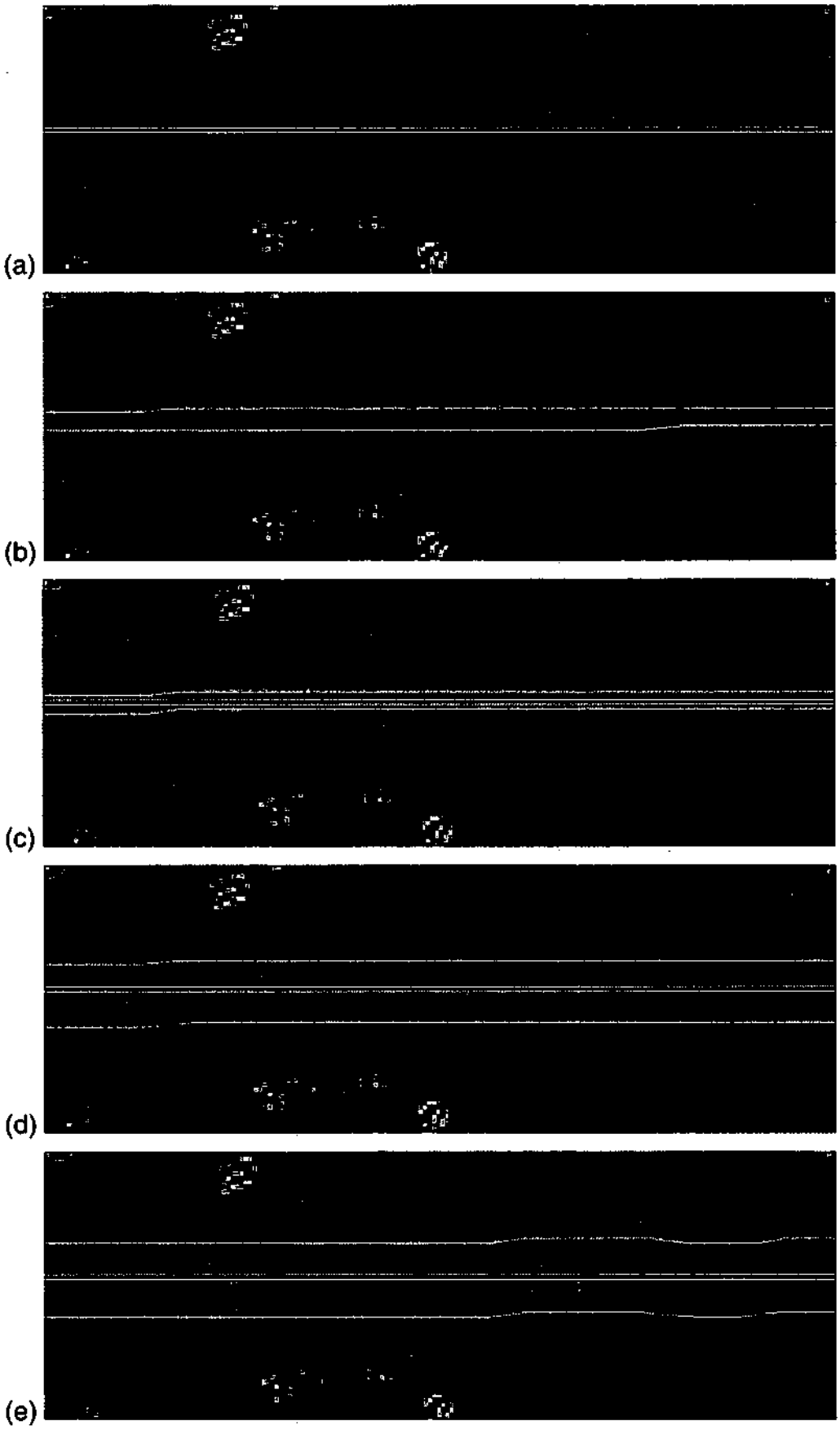


Abb. 6-8: *Twin snakes* zur Detektion von Böschungskanten: (a) Die Ausgangslage der beiden ***snakes***. (b) Zu Ende **iterierter** Zustand für den Sollabstand 4 Pixel. Die Klassifizierung von Segmenten ist bereits erfolgt. Ein Segment

wurde als unsicher (gelb) ausgewiesen, die anderen Segmente klassifizierte der Algorithmus als gut. (c) Das gelbe Segment wurde weiter optimiert und ein grüner Zustand für die Böschungsoberkante wurde entlang der gesamten Länge gefunden. (d) Zur Extraktion der Böschungsunterkante wurde der Sollabstand auf 16 Pixel erhöht. Der Ausgangsstatus für die beiden *snakes* ergibt sich nach etlichen Iterationen ohne Berücksichtigung photometrischer Terme. (e) Nach dem Reaktivieren der photometrischen Terme finden die beiden *snakes* die Böschungsunterkanten, wobei allerdings zwei Segmente als unsicher (gelb) ausgewiesen werden. Diese Segmente können nun nicht mehr verbessert werden.

7 Zusammenfassung

In dieser Arbeit wurden das ursprüngliche Modell von *snakes (active contours)* und eine Vielzahl von Vorschlägen zur Verbesserung, Modifikation und Erweiterung des Modells in Hinblick auf potenzielle Anwendungen in der digitalen Photogrammetrie und Topographie behandelt.

Zu Beginn der Arbeit wurden Stärken und Schwächen von *snakes* zusammengefasst. Anschließend wurden viele publizierte Modifikationen präsentiert. Es wurde eine neue Formulierung der internen Energieterme gefunden, die das Schrumpfen der *snake* unterbindet. Zwei neue Strategien der Anwendung von *snakes* wurden entwickelt: die Unterteilung in Segmente einschließlich des Befreiens einzelner Segmente aus lokalen Minima sowie die Kopplung zweier *snakes (twin snakes).*

Die *twin snakes* Strategie mit Sollabstand Null der beiden *snakes* in Kombination mit der Segmentierungsstrategie erlaubt es, eine einzelne Zielkurve von zwei entgegengesetzten Seiten zu detektieren. Diese Vorgehensweise findet garantiert das globale Minimum und wurde erfolgreich zur Schnittliniensuche in Orthophoto-Mosaiken eingesetzt. Das Potenzial von *snakes* wurde auch an einem Beispiel zu Geländekantendetektion in DTMs gezeigt.

An dieser Stelle sollen noch einmal die Probleme von *snakes* wiederholt werden, die oft den Einsatz in der Praxis behindern. Dazu werden die Nummern jener Abschnitte angegeben, in denen eine alternative Optimierungsmethode, eine angepasste Energiefunktion oder eine spezielle Anwendungsstrategie beschrieben werden, mit denen man das jeweilige Problem lösen oder zumindest in den Griff bekommen kann:

- Konvergenz zu lokalen Minima anstatt bei schlechter Ausgangslage der *snake* das globale Minimum zu finden: Abschnitte 3.5, 5.2, 5.3, 5.4, 5.5.3
- Beschränkte Reichweite der Anziehungskraft der Zielkurve: 3.4, 4.1.2, 4.2.1, 4.2.2, 4.3, 5.1, 5.5.2, 5.5.3
- Einschränkungen bezüglich der Form der Kurven:

  schlauch- oder röhrenförmige Objekte: 3.4, 5.5

  konkave Einbuchtungen, etc.: 3.4, 4.1.2, 4.2.4
- Überlaufen der Zielkurve: 3.2 oder 3.5
- Überspringen auf eine daneben liegende Kurve: 4.2.3, 5.5
- Schrumpfen der *snake:* 4.1.2 bis 4.1.7
- Selbstüberschneidungen: 4.2.3
- Schwierige Wahl der Gewichte für unterschiedliche Energieterme: 3.3, 4.4

Eine wesentliche Entscheidung, die bei der Auswahl des für eine bestimmte Anwendung besten *snake*-Modells getroffen werden muss, ist die Wahl einer Optimie-

rungsmethode. Fünf verschiedene Optimierungsmethoden wurden beschrieben und der Versuch eines Vergleichs wurde präsentiert. Allerdings kann eine fundierte Gegenüberstellung nur dann gegeben werden, wenn alle Methoden an denselben Beispielen mit denselben Startwerten getestet wurden. Ein derart umfassender Vergleich hätte den Rahmen dieser Arbeit gesprengt, da dies nur durch großen Programmieraufwand gemacht werden kann. Ein solcher Test ist dann möglich, wenn die Testaufgabe an die jeweiligen Urheber der Methode gestellt wird. Wenn *snakes* in photogrammetrischen Auswertesystemen Praxisreife erlangen sollen, wäre ein solcher Test eine wichtige Vorstufe. Er könnte von Organisationen wie die EuroSDR (vormals OEEPE; EuroSDR, 2003) koordiniert werden.

Zum Abschluss möchte ich noch einmal betonen, dass das Modell der *snakes* durchaus nicht immer problemlos arbeitet. Die meisten dieser Schwierigkeiten lassen sich allerdings lösen. Jedenfalls erfordert der Einsatz von *snakes* für Aufgaben der digitalen Photogrammetrie und Topographie die Adaption der Methode an die besonderen Gegebenheiten einer Anwendung. Ist in einem System eine vollständige Automatisierung vorgesehen, sind *snakes* im Allgemeinen nicht die beste Wahl. Sie wurden für halbautomatische Anwendungen konzipiert, in denen sie dem Operateur mühsame Detailarbeit abnehmen können.

Anhang A

Genauigkeit von Ableitungen

Viele der in dieser Arbeit vorgestellten Methoden setzen stark auf Ableitungen von Kurven bzw. Flächen auf. Dabei muss berücksichtigt werden, dass diese Kurven bzw. Flächen meist nicht analytisch beschreibbar sind, sondern aus diskreten Daten interpoliert werden. Falls gemessene Daten verwendet werden, muss weiter beachtet werden, dass Messdaten immer mit zufälligen Fehlern behaftet sind.

In diesem Anhang soll anhand von empirisch abgeleiteten Datenreihen die Genauigkeit von Ableitungen veranschaulicht werden, die

- mit unterschiedlicher Diskretisierungsweite bzw.
- aus verrauschten Daten

abgeschätzt werden.

Das Diagramm in Abb. **A-1** zeigt die Referenzkurve mit ihren Ableitungen bis zur vierten Ordnung. Es handelt sich um eine Sinusschwingung, die Werte zwischen 0 und 1 annimmt. Die vierte Ableitung ist wieder ident mit der Kurve selbst.

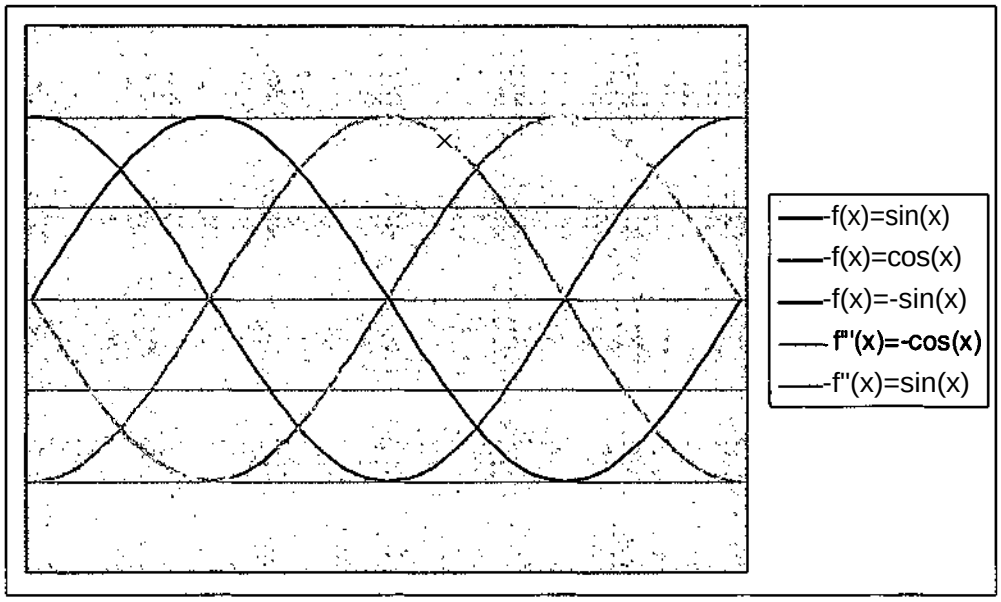


Abb. **A-1:** Sinusschwingung mit Ableitungen.

<u>Schätzungen der Ableitungen:</u>

Die erste Ableitung ist über den **Differentialquotienten** wie folgt definiert:

$$f'(x) = \lim_{dx \to 0} \frac{f(x+dx) - f(x)}{dx} \tag{A-1}$$

Liegt eine Kurve oder Fläche nur in diskreten Daten vor, können Ableitungen nur über Differenzenquotienten geschätzt werden. Ersetzt man das infinitesimal kleine *dx* durch eine Differenz δx, erhält man einen Differenzenquotienten als Abschätzung für die erste Ableitung. Diese bezieht sich allerdings genau genommen auf einen Abszissenwert in der Mitte zwischen *x* und *x*+ *Sx.*

$$f'\left(x + \frac{\delta x}{2}\right) \approx \frac{f(x+\delta x) - f(x)}{Sx} \tag{A-2}$$

Ist man aber an einer Schätzung der ersten Ableitung an der Stelle *x* interessiert und liegen die Funktionswerte nur in diskreten Abständen *Sx* vor, muss die erste Ableitung folgendermaßen geschätzt werden:

$$f'(x) \approx \frac{f(x+\delta x) - f(x-\delta x)}{25x} \tag{A-3}$$

Bei der Abschätzung der zweiten Ableitung kann der Differenzenquotient zwischen den Schätzungen für $f'(x + \delta x/2)$ und $f'(x - \delta x/2)$ verwendet werden:

$$f''(x) \approx \frac{f(x+\delta x) - 2f(x) + f(x-\delta x)}{\delta x^2} \tag{A-4}$$

Die dritte Ableitung wird auf ähnliche Weise geschätzt. Dazu benötigen wir zuerst:

$$f''\left(x + \frac{\delta x}{2}\right) \approx \frac{f'(x+\delta x) - f'(x)}{\delta x} \approx \frac{f(x+2\delta x) - f\left(x+\delta x\right) - f(x) + f(x-\delta x)}{2\delta x^2}$$

Damit ergibt sich die dritte Ableitung:

$$f'''(x) = \frac{f''\left(x + \frac{\delta x}{2}\right) - f''\left(x - \frac{\delta x}{2}\right)}{\delta x} \approx \frac{f(x+2\delta x) - 2f(x+\delta x) + 2f(x-\delta x) - f(x-2\delta x)}{2\delta x^3} \tag{A-5}$$

Zur Schätzung der vierten Ableitung benötigen wir wieder:

$$f'''\left(x + \frac{\delta x}{2}\right) \approx \frac{f''(x+\delta x) - f''(x)}{Sx} \approx \frac{f(x+2\delta x) - 3f(x+\delta x) + 3f(x) - f(x-\delta x)}{\delta x^3}$$

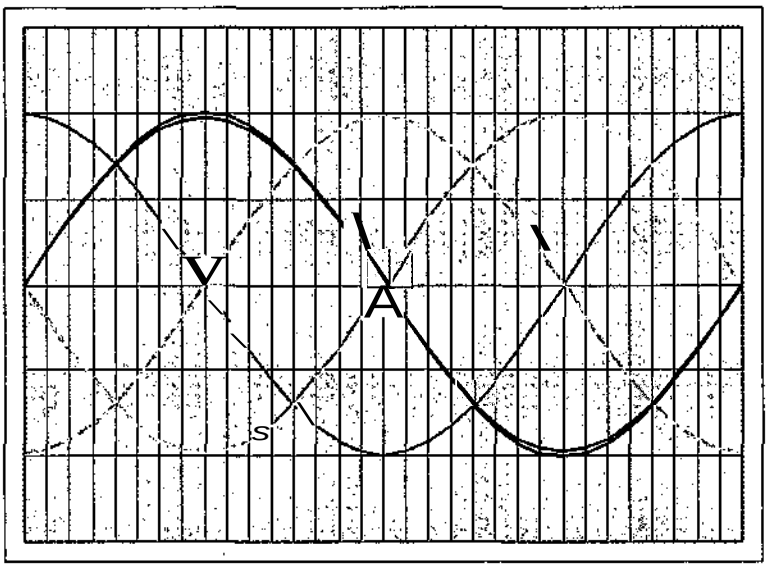
Es ergibt sich:

$$f^{IV}(x) \approx \frac{f'''\left(x + \frac{\delta x}{2}\right) - f'''\left(x - \frac{\delta x}{2}\right)}{Sx} \approx$$

$$\approx \frac{f(x+2\delta x) - 4f(x+\delta x) + 6f(x) - 4f(x-\delta x) + f(x-2\delta x)}{\delta x^4} \tag{A-6}$$

Einfluss der Diskretisierung:

Die Dichte der diskreten Daten ist oft nicht hoch genug, um die Ableitung genau genug schätzen zu können. Je höher die Ableitung, desto größer der Fehler der Abschätzung. Abbildung A-2 zeigt die geschätzten Ableitungen bei einer Diskretisierungsweite von $n / 16 = 0.196$ und darunter die Diskrepanzen zu den exakt berechneten Ableitungen (Soll — Ist).



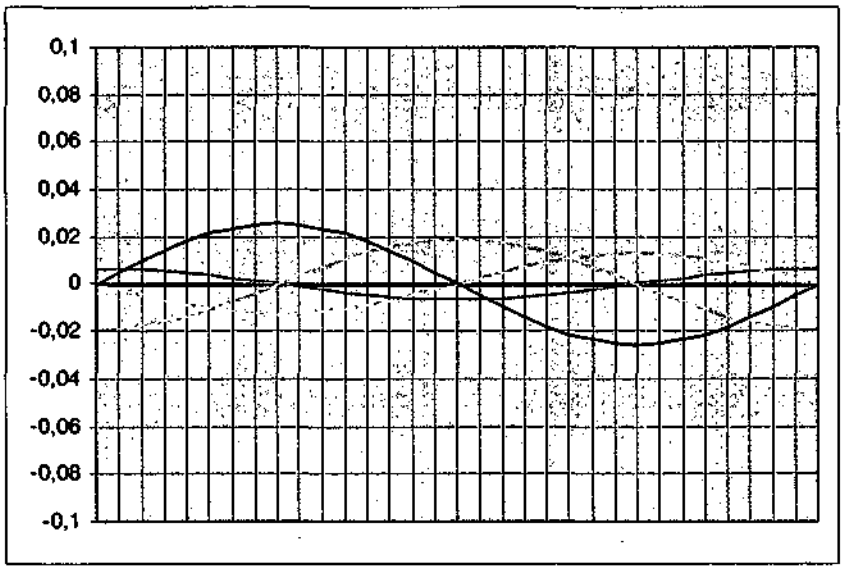


Abb. A-2: Oben: Ableitungen geschätzt aus Differenzenquotienten bei einer Diskretisierungsweite von $n / 16$. Unten: Abweichungen von den Sollwerten.

Man erkennt, dass die Differenzenquotienten die Ableitungen durchwegs zu klein schätzen. Wie erwartet, wächst die Diskrepanz mit steigender Ordnung der Ableitung. In Abbildung A-3 wurde die Diskretisierungsweite verdoppelt:

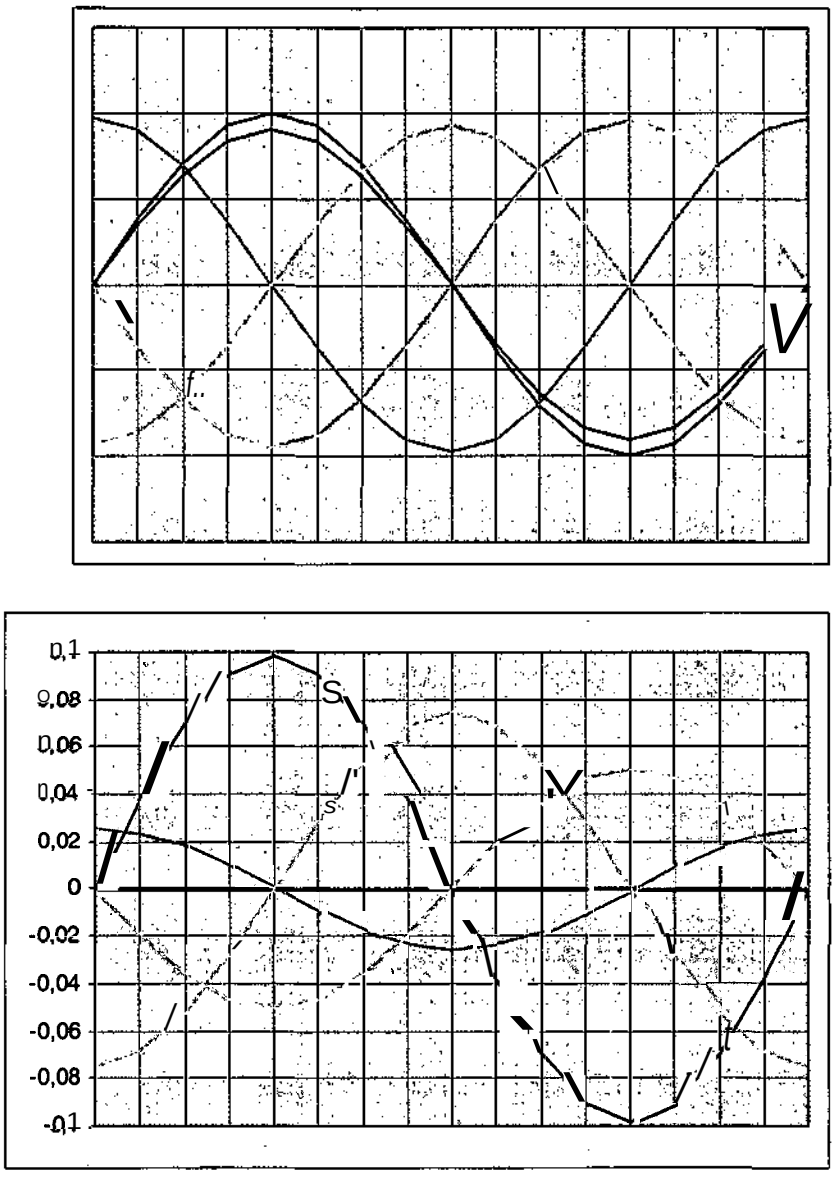


Abb. A-3: Ableitungen geschätzt aus Differenzenquotienten bei einer Diskretisierungsweite von π / 8. Unten: Abweichungen von den Sollwerten.

Die folgende Tabelle zeigt die Zunahme der Diskrepanzen bei Verdoppelung der Diskretisierungsweite. Die relative Zunahme wurde bezogen auf den Funktionswert 1, bei dem der maximale Fehler auftritt.

	$f'(x)$	$f''(x)$	$f'''(x)$	$f^{IV}(x)$
maximaler Fehler bei $Ax = \pi/16$	0.006	0.013	0.019	0.025
maximaler Fehler bei $An = \pi/8$	0.025	0.050	0.075	0.098
Zunahme des maximalen Fehlers	1.9 %	3.8 %	5.5 %	7.3 %

Einfluss des Rauschens:

Als nächstes soll veranschaulicht werden, wie sich zufällige Fehler (verrauschte Daten) auf die Ableitungen auswirken. Die Sinusschwingung wurde zufällig verfälscht, wobei der Absolutbetrag des addierten Rauschens maximal 0.001 betrug, also in Bezug auf den maximalen Funktionswert 1.0 ein Promille. In Abb. A-4 ist auf der linken Seite die verrauschte Ausgangskurve mit ihren Ableitungen bis zur dritten Ordnung dargestellt. Die Ableitungen wurden aus den Differenzenquotienten bei einer Diskretisierungsweite von $Sx = n / 32 = 0.098$ abgeschätzt (Formeln (A-3) bis (A-6)). Würde die Diskretisierung weniger dicht durchgeführt werden (eine höhere Diskretisierungsweite), würden auch die Fehler der Ableitungen sinken. Die rechte Seite zeigt die Differenzen zwischen den Sollwerten und den verrauschten Werten.

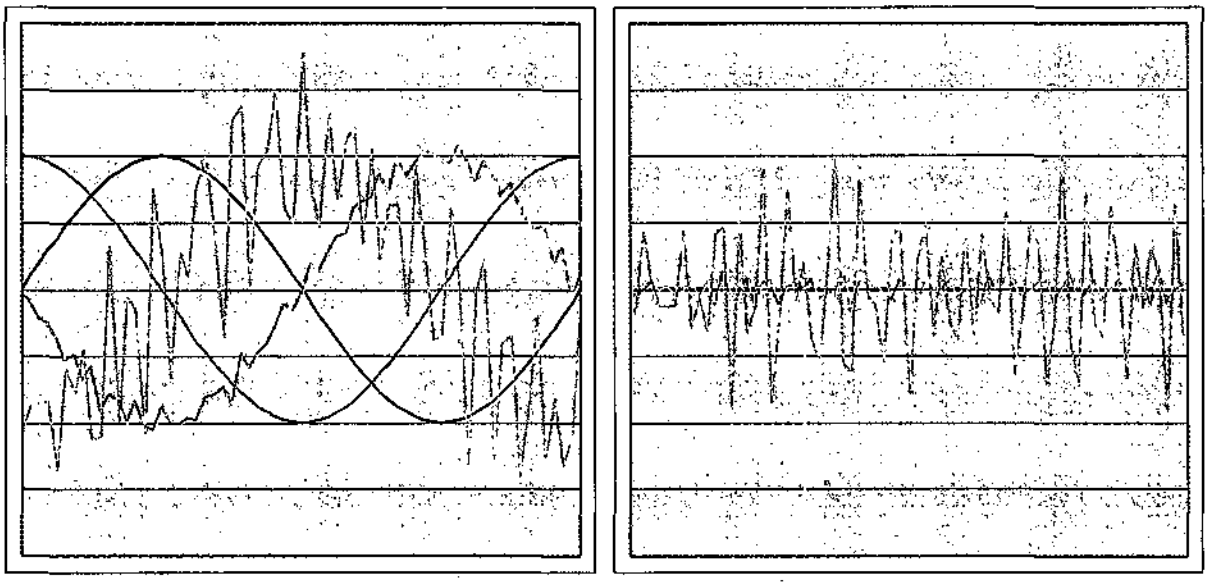


Abb. A-4: Links: aus Differenzenquotienten geschätzte Ableitungen der Ordnung eins bis drei. Rechts: Abweichungen von den Sollwerten.

Man erkennt deutlich, dass der Betrag des Fehlers stark mit der Ordnung der Ableitung steigt. Die geschätzten vierten Ableitungen nehmen sogar Werte zwischen -50 und +50 an und würden nicht mehr auf die Zeichenfläche von Abb. A-4 passen. Die folgende Tabelle zeigt die maximale Abweichung (absolut und relativ bezogen auf einen maximalen Funktions- bzw. Ableitungswert von 1.0):

	$f(x)$	$f'(x)$	$f''(x)$	$f'''(x)$	$f^{IV}(x)$
maximaler absoluter Fehler	0.001	0.006	0.16	1.0	54
maximaler relativer Fehler	0.1 %	0.6 %	16 %	100%	5400 %

References

Amini, A.A., Tehrani, S., Weymouth, T.E., 1988. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In: Proceedings of the Second International Conference on Computer Vision, ICCV 1988, pp. 95-99.

Beyer, G., 2003. Terrain inclination and curvature from wavelet coefficients. Approximation formulae for the relief. Journal of Geodesy (76), pp. 557-568.

Borgefors, G., 1986. Distance Transformations in Digital Images. Computer Vision, Graphics and Image Processing, CVGIP 34(3), pp. 344-371.

Briese, C., Kraus, K., Pfeifer, N., 2002. Modellierung von dreidimensionalen Geländekanten in Laser-Scanner-Daten. In: Festschrift anlässlich des 65. Geburtstags von Herrn Prof. Dr. Siegried Meier, TU Dresden, Inst. f. Planetare Geodäsie, pp. 47-52.

Brügelmann, R., 2000. Automatic breakline detection from airborne laser range data. In: International Archives of Photogrammetry and Remote Sensing, vol. XXXIII, part B3, Amsterdam, pp. 109-116.

Canero, C., Radeva, P., Toledo, R., Villanueva, J.J., Mauri, J., 2000. 3D Curve reconstruction by biplane snakes. In: Proceedings of the International Conference on Pattern Recognition, ICPR 2000, vol. 4, pp. 4563-4566.

Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698.

Caselles, V., Kimmel, R., Sapiro, G., 1997. Geodesic active contours. International Journal of Computer Vision 22(1), pp. 61-79.

Cham, T.J., Cipolla, R., 1997. Stereo coupled active contours. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, CVPR 1997, pp. 1094-1099.

Chen, D.H., Sun, Y.N., 2000. A self learning segmentation framework - the Taguchi approach. Computerized Medical Imaging and graphics, vol. 24, pp. 283-296.

Cohen, L.D., Cohen, I., 1993. Finite element methods for active contour models and balloons for 2-D and 3-D images. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-15, no. 11, pp. 1131-1147.

Cohen, L.D., Kimmel, R., 1997. Global minimum for active contour models: a minimal path approach. International Journal of Computer Vision 24(1), pp. 57-78.

Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J., 1995. Active shape models - their training and application. Computer Vision and Image Understanding 61(1), pp. 38-59.

Delagnes, P., Benois, J., Barba, D., 1995. Active contours approach to object tracking in image sequences with complex background. Pattern Recognition Letters, vol. 16, pp. 171-178.

EuroSDR, 2003. European Spatial Data Research, http://www.eurosdr.org.

Fischler, M.A., Tennenbaum, J.M., Wolf, H.C., 1981. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. Computer Graphics and Image Processing 15, pp. 201-223.

Forkert, G., Kerschner, M., Prinz, R., Rottensteiner, F., 1995. Reconstruction of free-formed spatial curves from digital images. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXX, Part 5W1, Zurich, pp. 163-168.

Förstner, W., 1995. The role of robustness in computer vision. In: Vision Milestones 95, Ed. by Pinz, A,. Burger, W., chapter 6, Österreichische Gesellschaft für Künstliche Intelligenz.

Fua, P., 1996. Model-base optimization: accurate and consistent site modeling. In: International Archives of Photogrammetry and Remote Sensing, vol. XXXI, part B3, Vienna, pp. 222-233.

Gomes Pereira, L.M., Janssen, L.L.F., 1999. Suitability of laser data for DTM generation: a case study in the context of road planning and design. ISRPS Journal of Photogrammetry & Remote Sensing, vol. 54, pp. 244-253.

Gomes Pereira, L.M., Wicherson, R.J., 1999. Suitability of laser data for deriving geographical information: a case study in the context of management of fluvial zones. ISRPS Journal of Photogrammetry & Remote Sensing, vol. 54, pp. 105-114.

Grün, A., Li, H., 1997. Semi-automatic linear feature extraction by dynamic programming and LSB-snakes. Photogrammetric Engineering and Remote Sensing 63(8), pp. 985-995.

Grzeszczuk, R.P., Levin, D.N., 1997. "Brownian strings": segmenting images with stochastically deformable contours. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 19(10), pp. 1100-1114.

Gülch, E., 1990. Extraction of contours in digital images by active contour models. In: International Archives of Photogrammetry and Remote Sensing, vol. XXVIII, part 3/2, Wuhan, China, pp. 211-220.

Gülch, E., 1995. From control points to control structures for absolute orientation and aerial triangulation in digital photogrammetry. Zeitschrift für Photogrammetrie und Fernerkundung ZPF 3/1995, pp. 130-136.

Gülch, E., 1996. Deformable models as a photogrammetric measurement tool - potential and problems. In: International Archives of Photogrammetry and Remote Sensing, vol. XXXI, part B3, Vienna, pp. *279-284.*

Gülch, E., 2000. Digital systems for automated cartographic feature extraction. In: International Archives of Photogrammetry and Remote Sensing, vol. XXXIII, part B2, Amsterdam, pp. 241-256.

Gunn, S.R., Nixon, M.S., 1997. A robust snake implementation; a dual active contour. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 19(1), pp. 63-68.

Han, C., Hatsukami, T.S., Hwang, J.-N., Yuan, C., 2001. A fast minimal path active contour model. IEEE Transactions on Image Processing 10(6), pp. 865-873.

Hoch, M., Litwinowicz, P.C., 1996. A semi-automatic system for edge tracking with snakes. The Visual Computer 12(2), pp. 75-83.

Ivins, J., Porrill, J., 1998. Constrained active region models for fast tracking in color image sequences. Computer Vision and Image Understanding 72(1), pp. 54-71.

Kass, M., Witkin, A., Terzopoulos, D., 1987. Snakes: active contours models. In: Proceedings of the First International Conference on Computer Vision, ICCV 1987, pp. 259-268.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: active contours models. International Journal of Computer Vision 1(4), pp. 321-331.

Kerschner, M., 1998. Homologous twin snakes integrated in a bundle block adjustment. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXII, Part 3/1, Columbus, Ohio, 1998, pp. 244-249.

Kerschner, M., 2001. Seamline detection in colour orthoimage mosaicking by use of twin snakes. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 56/1, pp. 53-64.

Klemencic, A., Kovacic, S., Leonardis, A., 1995. Redundant initialization and selection of active contour models for image segmentation. In Solina, F., Kropatsch, W.G. (eds.): Visual modules. Proceedings of the 19th ÖAGM and 1st SDRV Workshop, Schriftenreihe der OCG, Band 81, R. Oldenburg, Wien-München 1995, pp. 240-248.

Kölbl, Ch., Kraus, K., Oberhofer, A., 2002. Hochgenaues Geländemodell aus Laser-Scanner-Daten: Aufbau und Anwendung. Österreichische Wasser- und Abfallwirtschaft, Nr. 54/1-2, pp. 17-23.

Kovalski, G., Beyar, R., Shofti, R., Azhari, H., 2000. Three-dimensional automatic quantitative analysis of intravascular ultrasound images. Ultrasound in Medicine & Biology, 26(4), pp. 527-537.

Kraus, K, 1993. Photogrammetry, Volume 1, Fundamentals and standard processes. Dümmler, Bonn.

Kraus, K., 1996. Photogrammetrie, Band 2, Verfeinerte Methoden und Anwendungen. Dümmler Verlag, Bonn.

Kraus, K., 2000. Photogrammetrie, Band 3, Topographische Informationssysteme. Dümmler Verlag, Köln.

Lai, K.F., Chin, R.T., 1995. Deformable contours: modeling and extraction. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 17(11), pp. 1084-1090.

Lassige, T.A, Benkeser, P.J., Fyfe, D., Sharma, S., 2000. Comparison of septal defects in 2D and 3D echocardiography using active contour models. Computerized Medical Imaging and Graphics, 24(6), pp. 377-388.

Leroy, B., Herlin, I.L., Cohen, L.D., 1996. Multi-resolution algorithms for active contour models. In: 12th International Conference on Analysis and Optimization of Systems, pp. 58-65.

Li, H., 1997. Semi-automatic road extraction from satellite and aerial images. Dissertation am Institut für Geodäsie und Photogrammetrie der ETH Zürich, Mitteilungen Nr. 61.

Malladi, R., Sethian, J.A., Vemuri, B.C., 1995. Shape modeling with front propagation: a level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 17(2), pp. 158-175.

Marr, D., Hildreth, E., 1980. Theory of edge detection. In: Proceedings of the Royal Society of London, Vol. B 207, pp. 187-217.

Martine, H.M., 2001. Towards automatic modeling of buildings in informal settlements from aerial photographs using deformable active contour models (snakes). Ph.D. Thesis, Department of Geomatics, Faculty of Engineering and the Built Environment, University of Cape Town, South Africa.

Mathworld, 2003. http://mathworld.wolfram.com/.

Mayer, H., Laptev, I., Baumgartner, A., 1998. Multi-scale and snakes for automatic road extraction. In: 5th European Conference on Computer Vision, pp. 720-733.

McInerney, T., Terzopoulos, D., 1996. Deformable models in medical image analysis: a survey. Medical Image Analysis 1(2), pp. 91-108.

McKeown, D.M., Bulwinkle, T., Cochran, S., Harvey, W., McGlone, C., Shufelt, J.A., 2000. Performance evaluation for automatic feature extraction. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, Part B2, Amsterdam, pp. 379-394.

Meier, S., 2000. Die Snakes-Approximation als Hilfsmittel der Geodäten-Verarbeitung. Allgemeine Vermessungsnachrichten AVN 2/2000, pp. 50-57.

Menet, S., Saint-Marc, P., Medioni, G., 1990. B-snakes: implementation and application to stereo. In: Proceedings of the Image Understanding Workshop, DARPA 1990, pp. 720-726.

Neuenschwander, W., Fua, P., Iverson, L, Székely, G., Kubier, O., 1997. Ziplock snakes. International Journal of Computer Vision 25(3), pp. 191-201.

Ngoi, K.P., Jia, J.C., 1996. A new colour image energy for active contours in natural scenes. Pattern Recognition Letters 17, pp. 1271-1277.

NIST/SEMATECH, 2003. e-Handbook of statistical methods, http://www.itl.nist.gov/div898/handbook/eda/section35h.htm.

Optimization Online, 2003. http://www.optimization-online.org.

Osher, S., Sethian, J.A., 1988. Fronts propagation with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics, vol. 79, pp. 12-49.

Pardo, X.M., Radeva, P., 2000. Discriminant snakes for 3D reconstruction in medical images. In: Proceedings of the International Conference on Pattern Recognition, ICPR 2000, Vol. 4, pp. 4336-4339.

Park, J., Keller, J.M., 2001. Snakes on the Watershed. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 23(10), pp. 1201-1205.

Peterfreund, N., 1999. Robust tracking of position and velocity with Kaiman snakes. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 21(6), pp. 564-569.

Pfeifer, N., 2002. 3D Terrain Models on the Basis of a Triangulation. Geowissenschaftliche Mitteilungen (Veröffentlichung des Institutes für Photogrammetrie und Fernerkundung der TU Wien), Heft Nr. 65, 2002.

Radeva, P., Serrat, J., Marti, E., 1995. A snake for model-based segmentation. In: Proceedings of the Fifth International Conference on Computer Vision, ICCV 1995, pp. 816-821.

Rieger, W., Kerschner, M., Reiter, T., Rottensteiner, F., 1999. Roads and buildings from laser scanner data within a forest enterprise. In: International Archives of

Photogrammetry and Remote Sensing, Vol. XXXII, Part 3W14, La Jolla, California, pp. 185-191.

Ronfard, R., 1994. Region-based strategies for active contour models. International Journal of Computer Vision, 13(2), pp. 229-251.

Rottensteiner, F., 2001. Semi-automatic extraction of buildings based on hybrid adjustment using 3D surface models and management of building data in a TIS. Geowissenschaftliche Mitteilungen (Veröffentlichung des Institutes für Photogrammetrie und Fernerkundung der TU Wien), Heft Nr. 56, 2001.

Sbert, C., Sole, A.F., 2000. Stereo reconstruction of 3D curves. In: Proceedings of the International Conference on Pattern Recognition, ICPR 2000, vol. 1, pp. 1912-1915.

SCOP, 2003. http://www.ipf.tuwien.ac.at/products/.

Terzopoulos, D., Fleischer, K., 1988. Deformable models. The Visual Computer, vol. 4(6), pp. 306-331.

Terzopoulos, D., Szeliski, R., 1992. Tracking with Kaiman snakes. In: Blake, A., Yuille, A. (eds), Active Vision, MIT Press, pp. 3-20.

Torre, M., Radeva, P., 2000., Agricultural field extraction from aerial images using a region competition algorithm. In: International Archives of Photogrammetry and Remote Sensing, vol. XXXIII, part B3, Amsterdam, pp. 889-896.

Trinder, J.C., Li, H., 1996. Extraction of man-made features by 3-D active contour models. In: International Archives of Photogrammetry and Remote Sensing, vol. XXXI, part B3, Vienna, pp. 874-879.

Trinder, J.C., Maulik, U., Bandyopadhyay, S., 2000. Semi-automated feature extraction using simulated annealing. In: International Archives of Photogrammetry and Remote Sensing, vol. XXXIII, part B3, Amsterdam, pp. 905-911.

Williams, D.J., Shah, M., 1992. A fast algorithm for active contours and curvature estimation. Computer Vision Graphics and Image Processing CVGIP: Image Understanding 55(1), pp. 14-26.

Xu, C., Prince, J.L., 1998. Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing 7(3), pp. 359-369.

Xu, G., Segawa, E., Tsuji, S., 1994. Robust active contours with insensitive parameters. Pattern Recognition 27(7), pp. 879-884.

Zimmer, Y., Akselrod, S., 2000. Image segmentation in obstetrics and gynecology. Ultrasound in Medicine & Biology, Volume 26, Supplement 1, pp. S39-S40.

Zhu, S., Yuille, A., 1996. Region competition: Unifying snakes, region growing and Bayes/MDL for multiband image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 18(9), pp. 884-900.

Lebenslauf

Martin Kerschner

25. April 1970	geboren in Steyr (Oberösterreich). Eltern: Dkfm. Helmuth und Sigrid Kerschner
1976 – 1980	Volksschule in Waidhofen/Ybbs (Niederösterreich)
1980 – 1988	Bundesrealgymnasium in Waidhofen/Ybbs
1988 – 1995	Studium: Vermessungswesen an der Technischen Universität Wien
	Schwerpunkte: Photogrammetrie, digitale Bildverarbeitung
	Diplomarbeit: Kantenextraktion aus digitalen Bildern und Verfolgung glatter Linien
	am Institut für Photogrammetrie und Fernerkundung (I.P.F.)
1995 – 2002	Forschungsassistent am I.P.F.
	finanziert unter anderem vom FWF (Fonds zur Förderung der Wissenschaftlichen Forschung, Projektnummer P13725-MAT)
Juli 1998	Gewinner des „D.C. Brown Best Poster Award" im Rahmen der D.C. Brown International Summer School in Geomatics für den Beitrag „Twin Snakes for High Resolution Line Extraction" Prof. Rongxing Li
seit 2003	angestellt am VRVis Kompetenzzentrum für Virtual Reality und Visualisierung