

SUCHBASIERTE VERFAHREN ZUR AUTOMATISCHEN RASTER-VEKTOR* UND VEKTOR-RASTER TRANSFORMATION DER GRAUWERTBILDER

Yonglong Xu, MSc. Institut für Photogrammetrie und
Ingenieurvermessungen, Universität Hannover, BRD, Co.III

Zusammenfassung

Basierend auf den heuristischen Suchtechniken in der künstlichen Intelligenz werden in dieser Arbeit Verfahren zur schnellen automatischen Raster-Vektor- und Vektor-Raster-Transformation digitaler Grauwertbilder mit 100% Genauigkeit vorgeschlagen. Sie können zur Hybridmanipulation, Kontur-Verfolgung und Erzeugung von Binärbildern aus symbolischen Bildern angewendet werden. Die Leistungsfähigkeit und Korrektheit der Transformationen wurden mit Hilfe von 6 unterschiedlich aufgenommenen Grauwertbildern und 2 klassifizierten symbolischen Bildern prüft bzw. belegt.

Search based Methods for Automatic Raster-Vector and Vector-Raster Transformation of Grey Images

Based on the heuristic search techniques in the Artificial Intelligence, Methods for fast automatic raster-vector and vector-raster transformation of digital multigrey images with 100% correctness are presented in this paper. They can be used for hybrid-manipulation, contour following and creating of binary images from symbol images. The efficiency and correctness of the methods are proved by the transformations of 6 differently recorded grey images and 2 symbol images.

KEY WORDS

Raster- und Vektordaten, heuristische Suche, Grauwertbilder, depth-first Search, Backtracking.

1. EINLEITUNG

Seit über 10 Jahren beschäftigt man sich schon im Gebiet der Raster-Vektor-Transformation /Illert/. Da einerseits zur geometrischen Darstellung räumlicher Objekte in GIS meist Vektordaten bevorzugt verwendet werden /Lichtner, 1987,1989/, und andererseits die automatisch gescannten Karten in Rasterform liegen, ist ein Verfahren zur automatischen Raster-Vektor-Konvertierung erforderlich. Ein erwähnenswertes Programm dafür ist RAVEL /Lichtner,1987/. Dabei wird ein abgetastetes Grauwertbild zuerst binärisiert dann vektorisiert. Als Ergebnis bieten sich 2 Möglichkeiten an: Kantenpolygone und Skelette, die jeweils aus Knoten und Linien bestehen /Illert, Klauer/.

Heutzutage gewinnt in GIS bzw. Bildverarbeitung die Hybridmanipulation der Raster- und Vektordaten zunehmend an Bedeutung/Konecny/. Wobei werden kartographische Karten in Vektorform und Bilddaten in Rasterform getrennt gespeichert /Mayer/ und Hybridmanipulation betrifft häufig Visualisierung. Da es schwierig ist, ohne Fehler die beiden Daten gegenseitig zu konvertieren, bleibt die simultane Verarbeitung beider Daten in einem Algorithmus ein offenes

Forschungsthema. Derartige Verarbeitung wird ebenfalls in derzeitiger Bildanalyse stark gewünscht. Dies setzt vor allem eine schnelle, vollautomatische und völlig korrekte Raster-Vektor Umwandlung bzw. Zurückwandlung der Grauwertbilder voraus.

Mit dieser neuen Fragestellung haben wir Verfahren entwickelt, die die wünschenswerten schnellen und automatischen Raster-Vektor und Vektor-Raster Transformationen von Grauwertbildern ermöglichen. In der vorliegenden Arbeit werden erstens drei Sorten von verwendeten Objekten definiert. Anschließend werden die auf die heuristischen Suchstrategien /Bauer, Fu/ in der künstlichen Intelligenz aufgebauten Vorgehensweisen vorgestellt. Zum Schluß werden die Leistungsfähigkeit und Korrektheit durch Beispielstransformationen mit den auf Workstation SUN 4-SPARC in C-Sprache realisierten Programmen verdeutlicht. Dabei wurden acht digitale Bilder eingesetzt: zwei Amateuraufnahmen, eine Luftaufnahme, ein geocodiertes Seasat SAR-Bild, ein Thematic-Mapper TM1-Bild, ein Multispektralscanner-Bild in Blau-Kanal und zwei symbolische Bilder nämlich klassifizierte Thematic-Mapper- sowie SPOT-Daten. Jedes Bild wurde zuerst mit dem Vektorisierungsprogramm in

* Die Arbeit unter der Betreuung von Prof. Dr. mult. G. Konecny mit der Unterstützungen der Zentralen Kommission für die Förderung des wissenschaftlichen und künstlerischen Nachwuchses an der Universität Hannover und der Gottlieb Daimler- und Karl Benz-Stiftung durchgeführt.

Vektordaten umgewandelt, die dann mit dem Rasterisierungsprogramm in Rasterdaten zurückgewandelt wurden. Die jeweiligen Bilder und ihre Zurückgewandelte erwiesen sich als 100% identisch. Die durchschnittlich verbrauchte CPU-Zeit betrug lediglich 1.61 Millisekunden (ms) pro Pixel bei der Vektorisierung und 0.07 ms pro Pixel bei der Rasterisierung.

Bezüglich Anwendungen ist auf der Basis der Verfahren ein Programm zur schnellen Bestimmung von Regionsecken durch Hybridmanipulation entwickelt. Auch weitere Anwendungen wie Binärbilderstellung aus symbolischen Bildern und Konturverfolgung sind mit Hilfe der Verfahren ohne weiteres verwirklicht.

2. OBJEKTDEFINITIONEN

Hierbei werden zur Darstellung der Vektordaten 3 Sorten von Objekten nämlich Punkte, Linien und Polygone unterschieden.

- Ein Punkt ist ein Mittepixel in einem 3x3 Fenster aus einem Grauwertbild, dessen Grauwert unterschiedlich von den Grauwerten seiner 8 Nachbarspixeln ist, z.B. Mittepixel in Abb. 2.1(a) ist ein Punkt.
- Eine Linie ist eine Menge von nebeneinander stehenden Pixeln mit demselben Grauwert in einem Grauwertbild, die nicht geschlossen ist. Abb.2.1(b) zeigt ein Beispiel auf.
- Als ein Polygon ist eine geschlossene Linie definiert. Ein Polygon ist also praktisch die Kante (auch als Kontur oder Rand bezeichnet) einer homogenen Fläche mit demselben Grauwert in einem Grauwertbild. Ein entsprechendes Beispiel zeigt Abb. 2.1(c) an.

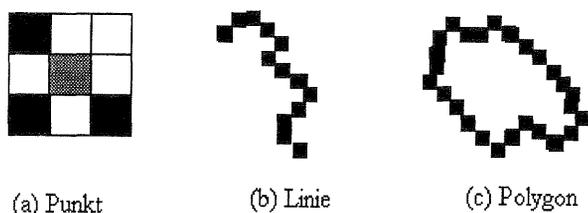


Abb. 2.1 Definitionen der Vektorobjekte

3. VEKTORISIERUNG

Die Umwandlung der Rasterdaten in die Vektordaten wird durch eine heuristische Zustandsraumsuchstrategie verwirklicht, die sich durch Abb. 3.1 veranschaulichen lässt. Zur Darstellung des Zustandes wird jeweils ein 3x3 Fenster ausgewählt. In Abb. 3.1(a) bildet ein 3x3

Fenster mit einem Eingangspixel (Pixel 1) den Anfangszustand, und in Abb. 3.1(b) beschreibt ein 3x3 Fenster mit einem Ausgangspixel (Pixel 5) einen Zielzustand. Pixel 9 ist ein Mittepixel, der wie der Eingangspixel und Ausgangspixel denselben Grauwert des zu vektorisierenden Objektes besitzt. Von dem Anfangszustand aus wird ein entsprechender Zielzustand gesucht. Da in Abb. 3.1 Pixel 4 oder Pixel 6 keinen Grauwert des zu vektorisierenden Objektes hat, ist Pixel 5 als ein Ausgangspixel festgestellt. Danach wird ein neues 3x3 Fenster mit dem Mittepixel als der neue Eingangspixel und dem Ausgangspixel als der neue Mittepixel eingerichtet, und ein neuer Ausgangspixel wird mit derselben Methode herausgefunden. Diese Vorgehensweise wird so lange wiederholt, bis entweder kein Ausgangspixel gefunden werden kann oder der letztlich festgestellte Ausgangspixel mit dem ersten Eingangspixel im ersten Fenster übereinstimmt. Damit ist die Suche nach einem Vektorobjekt beendet.

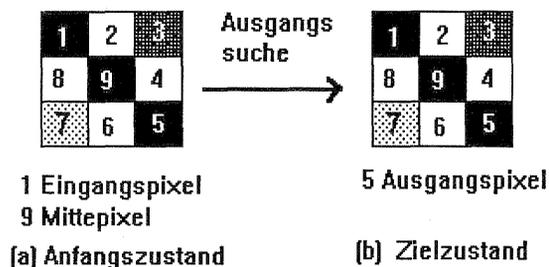


Abb. 3.1 Zustandsraumsuche

Um Fehler und blinde Suchvorgänge zu vermeiden und um die Leistung zu erhöhen, werden bei der ZielzustandSuche 3 heuristische Informationen benutzt :

- Suche des Ausgangspixels im Uhrzeigersinn nämlich in der Reihenfolge von Pixel 2 bis Pixel 8 in Abb. 3.1 zur Erhaltung der Konsistenz.
- Beschränkung der 4 Eckpixel in einem Fenster als Ausgangspixel unter den bestimmten Bedingungen, um Kreuzung bzw. Überlagerung zwischen Objekten zu vermeiden.
- Anzahl des Pixels mit dem Grauwert des Mittepixels innerhalb des Fensters zur Anzeige der möglichen Ausgangszahl. Gleich die Anzahl gerade 3, so existiert ein eindeutiger Ausgangspixel, z.B. Pixel 5 in Abb. 3.1. Ist die Anzahl größer als 3, so bestehen mehr als ein Kandidat des Ausgangspixels. Welcher davon ein plausibeler Ausgangspixel ist, kann schlecht mit

Hilfe der vorhandenen Information innerhalb des Fensters abgeschätzt werden. Dieses kritische Problem mit einer gewissen Unsicherheit lässt sich durch eine Strategie Depth-First-Search mit Backtracking automatisch lösen: dabei wird ein beliebiger Kandidat einfach als der Ausgangspixel des Fensters angenommen und die anderen Kandidaten werden für das eventuelle Backtracking aufbewahrt, damit die Suchvorgänge weiter durchführen können. Gleich die Anzahl 2, die kein Ausgangspixel mehr impliziert, so wird entweder ein automatisches Backtracking vorgenommen, falls entsprechende Pixeln für das Backtracking noch vorhanden sind, oder die Suchvorgänge werden gestoppt: ein Vektorobjekt ist erstellt. Abb. 3.2 zeigt diese Strategie, wobei jede Kante für ein Ausgang steht und jeder Knoten für ein Fenster steht. Die Ziffern zeigen die Bearbeitungsreihenfolge.

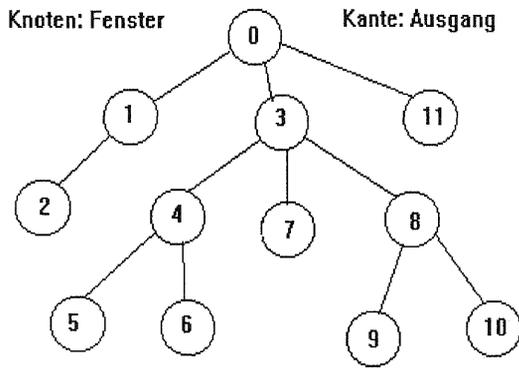


Abb.3.2 Depth-First-Search mit Backtracking

Wenn ein Vektorobjekt bereits erzeugt ist, wird es überprüft, ob dieses Objekt weiter unterteilt werden kann. Beispielsweise wird das Objekt wie in Abb. 3.3 durch eine Binärbaum-Strategie mit Depth-First-Search in 4 Objekte weiter unterteilt.

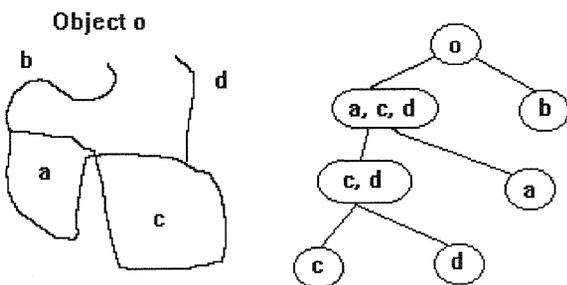


Abb. 3.3 ein Binärbaum zur Unterteilung des Objektes in 4 Objekte

Schließlich wird jedes Objekt überprüft, ob es andere Objekte enthält. Falls ja, werden die

enthaltenen Rasterobjekte nacheinander in die entsprechenden Vektorobjekte umgewandelt. Dies geschieht wiederum durch eine Strategie Depth-First-Search mit Backtracking, die in Abb. 3.4 illustriert wird. Ebenfalls stehen dabei die Ziffern für die Bearbeitungsreihenfolge.

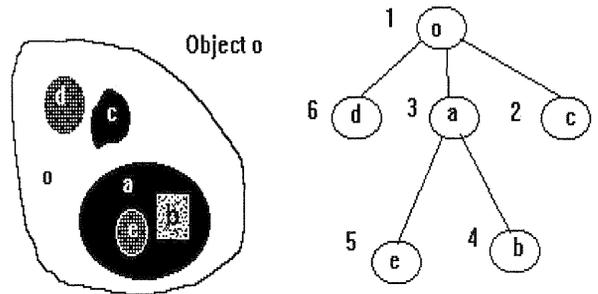


Abb. 3.4 Depth-First-Search mit Backtracking zur Vektorisierung der enthaltenen Objekte

4. RASTERISIERUNG

Bei dieser Transformation wird jeweils ein Vektorobjekt in Rasterdaten umgewandelt. Das Prinzip einer solchen Transformation lässt sich durch Abb. 4.1 erläutern. In Abb.4.1 sei P ein Pixel, V sei ein zu rasterisierendes Vektorobjekt, und S sei die Anzahl der Schnittpunkte auf V, deren X-Koordinaten kleiner als X-Koordinate des Pixels P. Ist S gerade, so liegt P außerhalb V. Ist S ungerade, so fällt P innerhalb V und Pixel P wird der Grauwert von V zugeordnet.

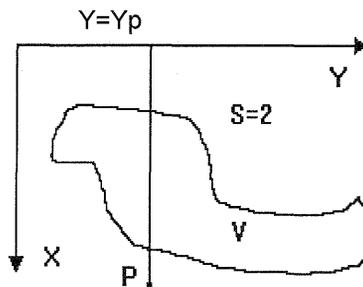


Abb. 4.1 Das Prinzip der Rasterisierung

Um die Transformation zu beschleunigen, wurden bei der Programmierung einige effektive Maßnahmen ergriffen, die hier nicht mehr näher besprochen werden. Die Wirkung zeigt sich durch die Transformationen der Grauwertbilder in dem nächsten Abschnitt.

Falls ein Objekt andere Objekte enthält, werden die enthaltenen Objekte zuerst rasterisiert. Ebenfalls erfolgt dies durch eine Strategie Depth-First-Search

mit Backtracking aber in einer gerade umgekehrten Reihenfolge gegenüber der Vektorisierung, die sich durch Abb. 4.2 anschaulich präsentieren läßt. Wie in Abb. 3.4 stehen hier die Ziffern auch für die Bearbeitungsreihenfolge.

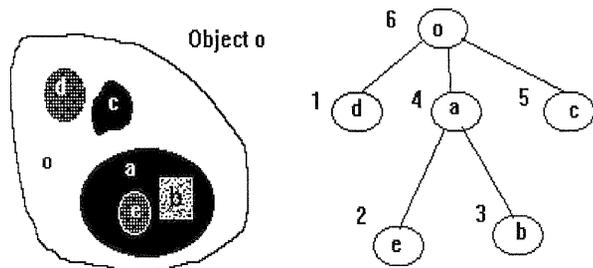


Abb. 4.2 Depth-First-Search mit Backtracking zur Rasterisierung der enthaltenen Objekte

5. PRÜFUNG DER LEISTUNGSFÄHIGKEIT

Aufgrund der oben präsentierten Verfahren wurden zwei entsprechende Programme zur Vektorisierung bzw. Rasterisierung auf Workstation SUN 4-SPARC in C-Sprache geschrieben. Zum Test der Leistungsfähigkeit und Korrektheit der Verfahren wurden zwei Amateuraufnahmen, eine Luftaufnahme, ein Multispektralscanner-Bild in Blau-Kanal, ein geocodiertes Seasat SAR-Bild, ein Thematic-Mapper TM1-Bild sowie zwei symbolische Bilder nämlich klassifizierte Thematic-Mapper- und SPOT-Daten ausgewählt. Tabelle 5.1 gibt die notwendigen Informationen über diese Bilder an. Das SAR-Bild und TM1-Bild wurden vor der Transformation jeweils durch Histogrammlinearisierung verstärkt. Das Fassade-Bild hatte 464x682 Pixeln, während andere jeweils 512x512 Pixeln hatten. In Abb. 5.1 werden die acht Bilder verkleinert aufgeführt.

Jedes Bild wurde zuerst mit dem Raster-Vektor Transformationsprogramm in Vektor-Daten umgewandelt, die dann mit dem Vektor-Raster Transformationsprogramm in Raster-Daten zurückgewandelt wurden. Es hatte sich herausgestellt, daß die jeweilige Bilder mit ihren zurückgewandelten Raster-Daten 100% identisch waren.

Der CPU-Zeitaufwand der Transformationen befindet sich in Tabelle 5.2. Je nach den Objektgrößen, Objekt- und Grauwertzahlen änderte sich die verbrauchte CPU-Zeit zwischen 0.41 - 2.29 ms pro Pixel bei der Vektorisierung, und zwischen 0.05 - 0.10 ms pro Pixel bei der Rasterisierung. Die durchschnittliche CPU-Zeit lag bei der Raster-

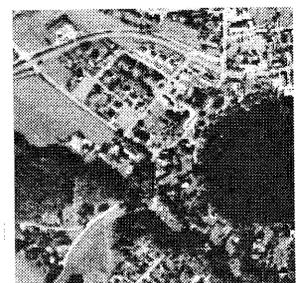
Vektor-Transformation an 1.61 ms pro Pixel, und bei der Vektor-Raster-Transformation an 0.07 ms pro Pixel. Also lief Rasterisierung wesentlich schneller als Vektorisierung.

Tabelle 5.1 Die eingesetzten Bilddaten

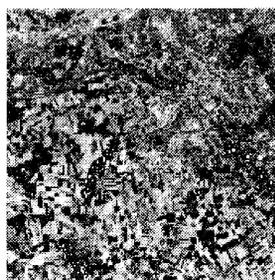
Bild Abb.5.1	Gebiet	Größe (Pixel)	Zahl der Grauwert
a		512x512	197
b	Oberpfaffenhofen	512x512	91
c	Hannover	512x512	46
d	Bonn	512x512	79
e		464x682	231
f	Hannover	512x512	168
g	Frankfurt	512x512	12
h	Hannover	512x512	9



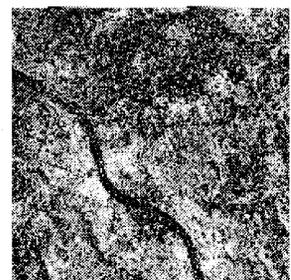
a. Mädchen



b. Luftbild



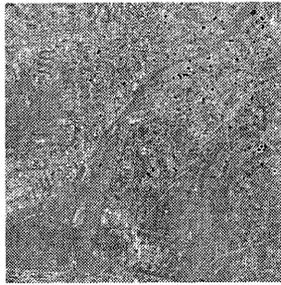
c. TM1-Bild



d. SAR-Bild



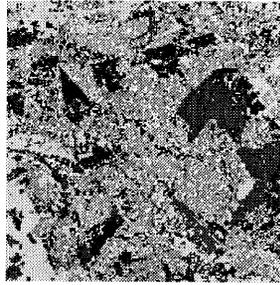
e. Fassade



f. Multispektralscanner-Bild



g. symbolische TM



h. symbolische SPOT

Abb. 5.1 8 verkleinerte Testbilder

Tabelle 5.2 die verbrauchte CPU-Zeit (Sekunden)

Bild Abb.5.1	Zeit bei Vektorisierung	Zeit bei Rasterisierung	Zahl der Objekte
a	496.8	14.9	182,426
b	423.2	17.8	151,661
c	444.4	14.2	170,833
d	581.7	15.9	226,708
e	725.7	28.1	265,087
f	584.3	16.9	219,105
g	129.2	26.7	29,229
h	107.4	22.0	21,911

6. ANMERKUNG

Neue Möglichkeiten werden durch die Einbeziehung der Verfahren zur schnellen Vektorisierung und Rasterisierung in die heutigen Bildanalyse-Systeme erschlossen. Damit ist es möglich, die gleichzeitige Manipulation der Raster- und Vektordaten innerhalb eines Prozesses vorzunehmen. Die vektorisierten Daten können für eine zweckmäßige Bildanalyse eingesetzt werden. Danach können die Daten wieder in Rasterdaten zurückgewandelt und darauf Raster-Operationen angewendet werden, wenn es notwendig ist. Diese Vorgänge sind praktisch stets

wiederholbar. Aufgrund der Verfahren sind z.B. 3 Programme zur Binärbilderzeugung aus symbolischen Bildern, zur Konturverfolgung und zur schnellen Findung von Regionsecken durch Hybridmanipulation geschrieben.

7. LITERATUR

Bauer, k.; I. Büttel; L. Eberhard; M. Hälker; H. Lehner; K. Micholka und R. Paulsburg, 1987 : Expertensystem - Einführung in Technik und Anwendung, Siemens Aktiengesellschaft, Berlin und München.

Fu, Kingsun; Z. Cai and G. Xu, 1987 : Artificial Intelligence and its Application, Beijing, Publishing House in Qinghua University.

Illert, A., 1987: Verfahren zur Raster-Vektor-Transformation, Institut für Kartographie, Universität Hannover, 4. Kontaktstudium.

Klauer, R.H., 1986 : Automatisierte Digitalisierung und Strukturierung von Strichvorlagen, ZfV, 4.

Konecny, G., 1989 : Beilage zum Einführung in das Seminar, Seminar Photogrammetrisch-kartographische Aspekte Geographischer Informationssysteme, Hannover, BRD.

Lichtner, W., 1987: RAVEL - Ein Programm zur Raster-Vektor-Transformation, KN, 2.

Lichtner, W. und A. Illert, 1989 : Entwicklungen zur kartographischen Mustererkennung, Geo-Informationssysteme: Anwendungen - neue Entwicklungen, Karlsruhe: Wichmann.

Mayer, H. und H. Yong, 1990 : Konzept zur hybriden Verarbeitung von Vektor- und Rasterdaten, 10. DGPF-Tagung, Darmstadt, BRD.

Niemann, H. und H. Bunke, 1987 : Künstliche Intelligenz in Bild- und Sprachanalyse, B. G. Teubner Stuttgart.

Sonne, B., 1989 : Hybrid Geo-Information Systems : A New Instrument for Applications Using Remote Sensing Data, Seminar Photogrammetrisch-kartographische Aspekte Geographischer Informationssysteme, Hannover, BRD.