

1

DIE ERKENNUNG VON KANTEN UND LINIEN
IN DIGITALEN BILDERN¹⁾

von

*Walter Kropatsch und Franz Leberl,
Technische Universität Graz, Österreich*

ZUSAMMENFASSUNG

Rechenverfahren zur Entdeckung von Kanten und Linien in digitalen Bildern ermöglichen grundlegende Operationen mit besonderer Bedeutung in möglichen photogrammetrischen Anwendungen. Sie dienen der Verbesserung (Vorverarbeitung) von Bildern der Photogrammetrie und Fernerkundung und sind wesentliche Elemente der automatischen oder interaktiven Mustererkennung. Algorithmen unterscheiden sich durch die Wahl der Definition einer "Kante" und "Linie". Der Beitrag gibt eine Übersicht verschiedener Rechenmethoden und deren Definition von Kante und Linie.

ABSTRACT

(Edge and Line Detection in Digital Images)

Edge and line detection algorithms permit basic operations on digital images, that have special significance in potential applications to photogrammetry. They may serve to improve (pre-process) photogrammetric and remote sensing imagery, and are a basic element of automatic or interactive pattern recognition. The different algorithms depend upon the chosen definitions of "edge" and "line". The paper presents a survey of various methods and discusses the underlying edge- and line-definitions.

RESUME

(Détection d'arêtes et de lignes dans les images numériques)

Une des opérations fondamentales dans les images numériques est la détection d'arêtes et de lignes. Ils peuvent servir à l'amélioration des images de photogrammétrie et de télédétection et ils forment un élément fondamentale dans la reconnaissance automatique ou interactive des patrons. Ce papier présente différentes méthodes pour la détection d'arêtes et de lignes.

¹⁾ Vorgelegt der Arbeitsgruppe III/2 "Bildverarbeitung",
Internationale Gesellschaft für Photogrammetrie
Symposium der Kommission III, Moskau, 29. Juli-5. August 1978, UdSSR

1. EINLEITUNG

Der Erforschung der Anwendungsmöglichkeiten der automatischen Mustererkennung in der Photogrammetrie und Fernerkundung wird gegenwärtig wachsendes Gewicht beigemessen. Zwar besteht im allgemeinen ein Gegensatz zwischen der assoziativen, komplexen geistigen Arbeitsweise des Photointerpreten und dem logischen, numerischen Vorgehen im automatischen Verfahren, jedoch bietet die automatische Mustererkennung für eine Reihe von Aufgaben in der Photogrammetrie und Fernerkundung Möglichkeiten der verbesserten Lösung. Als erstes, aber gewiß nicht einziges Beispiel sei hier auf die automatische Messung von künstlich markierten Triangulationspunkten und von Reseaukreuzen verwiesen (Kreiling, 1976; Roos, 1974). Der Bogen spannt sich bis zur heute futuristisch anmutenden rechnergestützten Revision von in digitaler Form abgespeicherten Karteninhalten. Wohl wurde der frühe Optimismus enttäuscht, sinnvolle Anwendungen der "Artificial Intelligence" zu finden, da die automatisierte Bildinterpretation in vielen Fällen bis heute kaum verwirklicht werden konnte. Jedoch sind in der Literatur Feststellungen zu finden, wonach für die meisten Probleme der automatischen Mustererkennung Lösungen anzugeben sind und Schwierigkeiten nur in der Geschwindigkeit und Allgemeinheit dieser Lösungen liegen (Alexsander, 1978).

Ein wesentliches Werkzeug der automatisierten Bildanalyse ist das Aufsuchen von Kanten und Linien in digitalen Bildern. Diese Aufgabe stellt sich bei der Unterteilung und Beschreibung von Bildern, bei der Erkennung von Objekten, und kann schließlich zur Vorverarbeitung (Kantenverstärkung) für die nachfolgende Bildinterpretation wie auch für eine erhöhte Effizienz der Stereokorrelation dienen (Nack, 1976; Kaneko, 1976).

Zur Frage der Kanten- und Linienerkennung besteht eine umfangreiche Literatur, welche dieses Arbeitsgebiet als eines der Fundamente der digitalen Mustererkennung erscheinen läßt. Wir stellen uns in der vorliegenden Arbeit die Aufgabe, eine auf der modernen Mathematik aufbauende Übersicht der theoretischen Grundlagen der Kanten- und Linienerkennung zusammenzustellen. Die Behandlung der mathematischen Aspekte der Bildverarbeitung verlangt eine in der Photogrammetrie und Fernerkundung zum Teil kaum geläufige Darstellungsweise. Daher werden im Anhang die verwendeten Symbole beschrieben.

Nach einer einführenden Übersicht des Vorgehens bei der Kanten- und Linienerkennung werden die Begriffe *digitales Bild*, *Nachbarschaft* zwischen Bildpunkten und *Abstand* eingeführt und zur mathematischen Definition von *Kante*, *Linie* und *Bildregion* verwendet. Die Darstellung schließt an Arbeiten von Rosenfeld (1970), Rosenfeld und Davis (1975) an, weicht jedoch insofern davon ab, als wir hier vom Begriff der *Nachbarschaft* ausgehen, den wir mathematisch zu definieren versuchen.

Die Rechenvorschriften zur Entdeckung von Kanten und Linien sind Inhalt des nächsten Abschnittes. Eine Unzahl von Algorithmen wurde bisher für diese Aufgabe vorgeschlagen und mehrfach übersichtsweise dargestellt (vgl. Davis, 1973; Fram und Deutsch, 1975; und andere). Wir versuchen, die Vielzahl der Verfahren durch eine sinnvolle Einteilung in Gruppen zusammenzufassen und die Vor- und Nachteile der Verfahren abzuschätzen. Damit hoffen wir, im Rahmen der Photogrammetrie und Fernerkundung die Bearbeitung von Fragen der Mustererkennung anzuregen und zu ihrer verstärkten mathematischen Behandlung in diesem Anwendungsbereich beizutragen.

2. KANTEN UND LINIEN IN DIGITALEN BILDERN

In *natürlichen* digitalen Luft- oder Satellitenbildern der Erdoberfläche stellt die Verstärkung von Kanten ein wirkungsvolles Mittel der Bildverbesserung zur nachfolgenden Interpretation (oder Mustererkennung) dar. "Kante" wird hier (vorläufig) als Veränderung oder Diskontinuität der Grauwerte im Bild verstanden.

Der Begriff *Kante* besteht nicht nur für den Grauwert, sondern auch für andere Größen wie zum Beispiel für *Textur* oder *Farbe*. Die selben Rechenverfahren sind zur Erkennung von Grauwert, Textur- oder Farbkanten anwendbar, vorausgesetzt, daß den Elementen im digitalen Bild (Pixels) statt Grauwerten Textur- oder Farbwerte zugeordnet werden.

Im Abschnitt 4. werden zwei Klassen von Verfahren der Kanten- und Liniensuche behandelt: parallele und sequentielle. Obwohl sie vielfach als alternative Verfahren dargestellt werden, ergänzen sie einander. *Parallele* Verfahren gehen von einer Kantenverstärkung aus, der eine Zerlegung des Bildes in Äquidensiten folgt. Nach Pratt (1978, S.479) ist die richtige Wahl der Schwellwerte zur Erzeugung eines (binären) Äquidensitenbildes kritisch und ein Kernproblem. Als Ergebnis wird ein Kantenbild erhalten (Bild von Kantenpunkten). Das Aufsuchen von *Linien* geschieht sodann in einem sequentiellen Verfahren durch Verbinden von Kantenpunkten.

Andere sequentielle Verfahren der *Linienerkennung* gehen nicht von einem (binären) Kantenbild aus, sondern von ursprünglichen Bildern oder ihrer Äquidensitendarstellung.

Es zeigt sich also, daß das Aufsuchen von Linien im digitalen Bild
a) mit den Grau-, Textur- oder Farbwerten des ursprünglichen Bildes oder
b) mit den unter Verwendung von Schwellwerten abgeleiteten Äquidensiten abläuft. Als Vorverarbeitungsschritt zur Äquidensitendarstellung kann eine (b1) Kantenverstärkung oder (b2) Erfassung von Flächen gleicher oder ähnlicher Grauwerte, Textur oder Farbe ("Region Growing") dienen.

Zur Definition von Kanten und Linien ist es notwendig, Bildpunkte zueinander in Beziehung zu setzen. Hierbei sind "Nachbarschaftsbeziehungen" das wesentliche Werkzeug. Daher steht im folgenden Abschnitt eine mathematische Definition der Nachbarschaft zentral. Diese Definition, und die daraus folgende Definition der Kante, Linie usw. beruht auf der geometrischen Beziehung zwischen den Elementen des Bildbereiches (Pixels), nicht aber ihren Grauwerten (oder Textur und Farbe).

3. DEFINITIONEN UND MATHEMATISCHE GRUNDLAGEN

3.1 Definition eines digitalen Bildes

Digitale Bilder werden durch eine Funktion

$$(3.1) \quad g : B \longrightarrow G$$

eines endlichen k -dimensionalen Bildbereiches $B \subset \mathbb{N}^k$ beschrieben (\mathbb{N} seien die natürlichen Zahlen $1, 2, \dots$; k meist 2), welche B in einen ebenfalls endlichen Grauwertbereich $G \subset \mathbb{N}$ abbildet. Die Anzahl der Elemente von G ist wegen der binären Darstellung der Grauwerte im Computer meist eine Potenz von 2, sodaß gilt:

$$(3.2) \quad \text{card } (G) = 2^b \quad (b \dots \text{Bitanzahl per Grauwert})$$

Im Fall eines 2-dimensionalen Bildbereichs gilt:

$$(3.3) \quad B = R \times C \subset \mathbb{N}^2 \quad (R \dots \text{Zeilen; } C \dots \text{Spalten})$$

und die Funktion g kann in Form einer "Grauwertmatrix" dargestellt werden, wie es für Satellitenbilder eines einzelnen Frequenzbereiches üblich ist. Für Mehrfachbilder, zum Beispiel multispektrale Aufnahmen, gilt:

$$B \subset \mathbb{N}^3.$$

3.2 Definition einer "Nachbarschaft"

Auf dem Bildbereich B sei nun eine Nachbarschaft $\Gamma(x)$ eines Punktes x definiert. Daraus wird eine Metrik abgeleitet.

Definition (1):

$\Gamma(x) \subset \mathbb{Z}^k$ heißt Menge der "Nachbarn" von $x \in B$, wenn gilt:

- a) $x \notin \Gamma(x)$
- b) $x \in \Gamma(y) \iff y \in \Gamma(x)$.

\mathbb{Z} ist die Menge der ganzen Zahlen.

Definition (2):

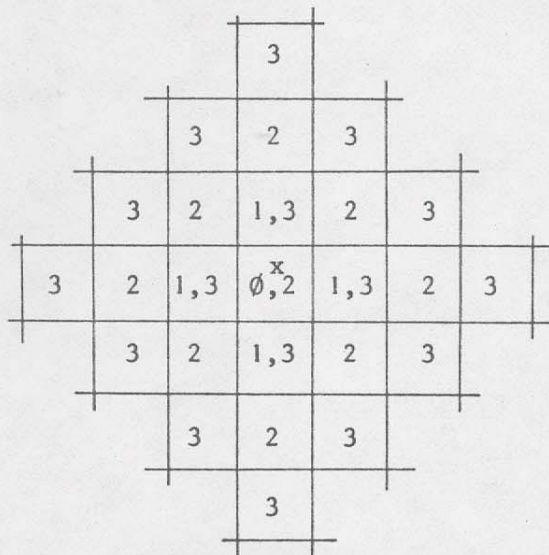
$\{\Gamma^n(x) \subset B \mid n \in \mathbb{N}\}$ sei ein Nachbarschaftssystem um $x \in B$, wenn gilt:

- a) $\bigvee_{x \in B} \Gamma^0(x) := \{x\}$;
- b) $\Gamma^1(x) := \Gamma(x) \cap B$;
- c) $\Gamma^{n+1}(x) := \bigcup_{y \in \Gamma^n(y)} \Gamma(y) \cap B$.

Aus den Definitionen (1) und (2) ersehen wir, daß $\Gamma(x)$ nicht notwendigerweise, wohl aber $\Gamma^1(x)$ eine Teilmenge von B zu sein hat. $\Gamma(x)$ ist nämlich eine Teilmenge der ganzen Zahlen \mathbb{Z}^k , kann also auch negative Elemente haben, während $\Gamma^1(x)$ nur positive Elemente hat. Der Unterschied zwischen $\Gamma^1(x)$ und $\Gamma(x)$ liegt darin, daß, wie schon erwähnt, $\Gamma(x)$ Elemente außerhalb des Bildbereiches B beinhalten kann. Diese könnten im Gegensatz zu B, da $B \subset \mathbb{N}^k$, auch negative Werte haben.

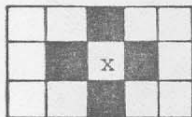
Man beachte weiters, daß die Nachbarschaft n-ter Ordnung $\Gamma^n(x)$ nicht unbedingt jene n-1-ter Ordnung $\Gamma^{n-1}(x)$ enthält!

Wir beschreiben in Tabelle 1 einige Nachbarschaften, wobei elementare Bewegungen von Figuren auf einem Schachbrett als Vorstellungshilfe dienen können. Figur 1 enthält für einen Punkt x eine graphische Darstellung von $\Gamma_4(x)$ -Nachbarschaften erster, zweiter und dritter Ordnung.



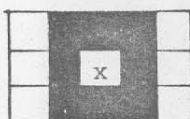
a) Figur 1: Γ_4 -Nachbarschaften 0., 1., 2. und 3. Ordnung eines Punktes $x \in \mathbb{R} \times \mathbb{C}$.

B 1) "4-Nachbar" (Rosenfeld, 1970):



$$\Gamma_4(x) = \{\blacksquare\} \text{ Beachte: } \text{card}(\Gamma_4(x)) = 4$$

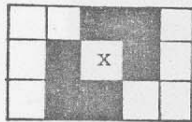
B 2) "8-Nachbar" Königszug (Rosenfeld, 1970):



$$\Gamma_8(x) = \{\blacksquare\} \text{ Beachte: } \text{card}(\Gamma_8(x)) = 8$$

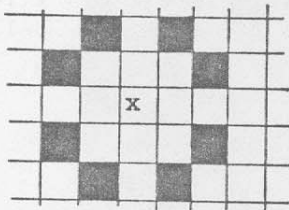
Tabelle 1 (wird fortgesetzt)

B 3) "Schräge Nachbarmengen" könnten bei Bearbeitung geometrischer verzerrter Bilder sinnvoll sein:



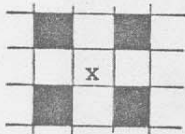
$$\Gamma_v(x) := \{ \blacksquare \}$$

B 4) Als Beispiel einer besonderen Nachbarschaft sei die "Springer"-beziehung definiert:



$$\Gamma_{sp}(x) := \{ \blacksquare \}$$

B 5) Den Zugmöglichkeiten beim "Dame"-Spiel entspricht:



$$\Gamma_D(x) := \{ \blacksquare \}$$

(Fortsetzung von voriger Seite)

Tabelle 1: Fünf Beispiele von Nachbarschaften in digitalen Bildern.

3.4 Ableitung einer Metrik

Um einen Abstand zwischen zwei Elementen von B einzuführen ist es notwendig, daß alle Punktpaare $(x,y) \in B^2$ miteinander durch die Nachbarschaftsbeziehung verbunden sind. Bestehen Punktpaare, die in dieser Beziehung niemals einander Nachbarn beliebiger Ordnung werden können, so ist ein Abstand nicht definiert. Daher der

Satz (1):

Gibt es für jeden Punkt $x \in B$ ein Nachbarschaftssystem

$$\{ \Gamma^i(x) \} \text{ und gilt außerdem}$$

$$(3.4) \quad \forall_{x \in B} \quad \exists_{n \in \mathbb{N}_0} \quad B = \bigcup_{i=0}^n \Gamma^i(x) \quad (\mathbb{N}_0 \dots \text{natürliche Zahlen und Null})$$

dann wird durch

$$(3.5) \quad d(x,y) := \min \left\{ k \in \mathbb{N}_0 \mid x \in \Gamma^k(y) \right\}$$

ein Abstand auf B definiert.

¹⁾
Beweis:

$$(*) \quad d(x,x) = \emptyset \quad \text{da} \quad \Gamma^{\emptyset}(x) = \{x\}.$$

$$(*) \quad d(x,y) = \emptyset \implies \min \{k \in \mathbb{N}_0 \mid x \in \Gamma^k(y)\} = \emptyset \implies x \in \Gamma^{\emptyset}(y) \implies x=y$$

$$(*) \quad d(x,y) \geq \emptyset \quad \text{trivial}$$

$$(*) \quad d(x,y) = d(y,x) \quad \text{ist zu zeigen:}$$

Hierzu beweisen wir erst, daß es ein $x \in \Gamma^k(y)$ geben muß und auch gilt:

$$y \in \Gamma^k(x).$$

Wegen der Definition von $d(x,y)$ in (3.5) gibt es ein minimales $k=d(x,y)$, sodaß:

$$x \in \Gamma^k(y).$$

Die Existenz eines solchen $k \in \mathbb{N}$ ist wegen der Voraussetzung (3.4), also wegen $y \in B$ und

$$B = \bigcup_{i=\emptyset}^n \Gamma^i(y),$$

gesichert und außerdem gilt $k \leq n$.

Auf Grund der Definition 2c) von Γ^k folgt folgender Zusammenhang:

$$x \in \Gamma^k(y) \implies \exists z_1, z_2, \dots, z_{k-1} \in B \text{ mit } \begin{cases} x \in \Gamma(z_{k-1}) \\ z_{i+1} \in \Gamma(z_i) \text{ für } i=1, \dots, k-2 \\ z_1 \in \Gamma(y) \end{cases}$$

Wegen Definition 1b) gilt aber auch umgekehrt:

$$\exists z_1, \dots, z_{k-1} \in B \text{ mit } \begin{cases} z_{k-1} \in \Gamma(x), \\ z_i \in \Gamma(z_{i+1}) \text{ für } i = 1, \dots, k-2 \\ y \in \Gamma(z_1). \end{cases}$$

Dies bedeutet, daß

$$y \in \Gamma^k(x)$$

und

$$d(y,x) \leq d(x,y) = k.$$

¹⁾ Die 5 Schritte des Beweises werden mit einem (*) gekennzeichnet.

Durch formales Vertauschen von x und y erhält man

$$d(x,y) \leq d(y,x)$$

und als Ergebnis

$$d(x,y) = d(y,x)$$

was zu beweisen war.

(*) $d(x,z) \stackrel{!!}{\leq} d(x,y) + d(y,z)$ ist zu zeigen:

Wie zuvor erhält man

$$d(x,y) = k \implies x \in \Gamma^k(y),$$

$$d(y,z) = m \implies y \in \Gamma^m(z),$$

$$x \in \Gamma^k(y) \implies \exists r_1, r_2, \dots, r_{k-1} \in B \text{ mit } \begin{cases} x \in \Gamma(r_{k-1}) \\ r_{i+1} \in \Gamma(r_i) \text{ für } i=1, \dots, k-2, \\ r_1 \in \Gamma(y) \end{cases}$$

$$y \in \Gamma^m(z) \implies \exists s_1, s_2, \dots, s_{m-1} \in B \text{ mit } \begin{cases} y \in \Gamma(s_{m-1}) \\ s_{i+1} \in \Gamma(s_i) \text{ für } i=1, \dots, m-2, \\ s_1 \in \Gamma(z) \end{cases}$$

Setzt man nun

$$t_i := \begin{cases} s_i & \text{für } i = 1, 2, \dots, m-1 \\ y & \text{für } i = m \\ r_{i-m} & \text{für } i = m+1, \dots, m+k-1, \end{cases}$$

so hat man eine Folge $t_1, t_2, \dots, t_{m+k-1}, t_i \in B$, für die gilt:

$$\begin{cases} x \in \Gamma(t_{m+k-1}), \\ t_{i+1} \in \Gamma(t_i) & \text{für } i = 1, 2, \dots, m+k-2, \\ t_1 \in \Gamma(z). \end{cases}$$

Das bedeutet: $x \in \Gamma^{k+m}(z)$ und

$$d(x,y) \leq k+m = d(x,y) + d(y,z),$$

was zu beweisen war.

3.5 Anwendung der Metrik auf Beispiele (B1) bis (B5)

Mit obiger Metrik ergeben sich für Beispiele B1 und B2 der Tabelle 1 aus der Anschauung (Schachbrett, Figuren mit Zugmöglichkeit wie in den Beispielen) folgende Abstandsfunktionen für $B \subset \mathbb{N}^k$ ($x = (x_1, x_2, \dots, x_k)$, $y = (y_1, y_2, \dots, y_k)$) :

$$d_4(x, y) = \sum_{i=1}^k |x_i - y_i|$$

$$d_8(x, y) = \max_{i=1, \dots, k} (|x_i - y_i|)$$

Für Beispiele B3 und B4 existiert die Abstandsfunktion, ist aber wesentlich komplizierter. Für Beispiel B5 ist Bedingung 3.4 nicht erfüllt. Bekanntlich zieht man beim Dame-Spiel entweder auf den schwarzen oder auf den weißen Feldern des Schachbretts. Somit ist die Gesamtheit B der Felder niemals durch die Vereinigung von n Nachbarschaften $\Gamma_D^n(x)$ zu erhalten.

3.6 Definition von Zusammenhangsbeziehungen

Definition (3): Eine Folge $(p_i)_{i=1, \dots, n}$ von Bildpunkten $p_i \in B$ heißt Weg von p_1 nach p_n , wenn

$$\forall 1 \leq k < n \quad p_{k+1} \in \Gamma^1(p_k) \cup \{p_k\}.$$

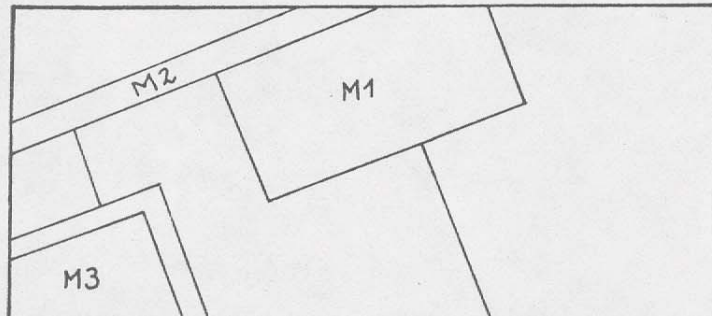
Definition (4): Eine Menge $M \subset B$ heißt: *zusammenhängend*, wenn

$$\forall (x, y) \in M \quad \exists \text{ Weg}(p_i) \text{ von } x \text{ nach } y \quad \forall_i p_i \in M$$

Definition (5): Eine Menge $M \subset B$ heißt *einfach zusammenhängend*, wenn sie

- zusammenhängend ist und auch
- $\bar{M} = B - M$ zusammenhängend ist, wobei in \bar{M} eine andere Nachbarschaftsbeziehung als in M gelten kann (vergleiche dazu Rosenfeld (1970)).

Figur 2 erläutert diese Begriffe.



Figur 2: $M_1, M_3 \dots$ einfach zusammenhängend;
Mit Γ_4 oder Γ_8 ist M_2 nicht einfach zusammenhängend.

3.7 Definition einiger Begriffe der digitalen Bildverarbeitung

Die Definitionen (3), (4) und (5) werden für die Definition der grundlegenden Begriffe Kante, Linie und Gebiet in einem Digitalbild benötigt:

Definition (6):

- a) Eine *Kante* e in einem digitalen Bild ist ein Paar $(i, o) \in B^2$ von Nachbarbildpunkten, wenn gilt:

$$i \in \Gamma(o).$$

- b) Unter dem *Gewicht* oder *Stärke* einer Kante versteht man eine Funktion

$$w : B^2 \longrightarrow \mathbb{R} \quad (\mathbb{R} = \text{Zahlenkörper der reellen Zahlen})$$

- c) Die *Richtung* einer Kante wird ebenfalls durch eine Funktion

$$\varrho : B^2 \longrightarrow \mathbb{R}$$

ausgedrückt.

Somit definiert *jedes benachbarte* Punktepaar eine Kante. Erst ihr *Gewicht*, welches mittels der Grauwerte (Textur oder Farbe) des Punktepaares bestimmt wird, verleiht dieser Kante ihre physikalische Bedeutung!

Definition (7):

- a) Eine *Linie* L ist eine endliche Folge von Kanten $(e_i, i = 1, \dots, n)$, wenn je zwei aufeinanderfolgende Kanten e_i und e_{i+1} in einer Nachbarschaftsbeziehung \mathcal{L} bezüglich B^2 stehen:

$$e_{i+1} \in \mathcal{L}(e_i) \quad (\mathcal{L} \text{ Nachbarmenge bezüglich Punkten aus } B^2, \text{ vgl. Definition (1)}).$$

- b) Sei $(e_i)_{i=1, \dots, n}$ eine Linie L ;
 L heißt geschlossen, wenn $e_1 \in \mathcal{A}(e_n)$. Jede nicht geschlossene Linie wird als offen bezeichnet.
- c) Die *Länge* l einer Linie sei definiert als die Anzahl der Kanten, die sie erzeugen.

$$l(L) = n, \text{ wenn } L = (e_i)_{i=1, \dots, n}$$

Definition (8): Eine *Region* (Fläche, Gebiet) in einem digitalen Bild ist jede einfach zusammenhängende Teilmenge $R \subset B$.

Definition (9): Als *Randlinie* L einer Region R bezeichnet man eine Linie

$$L = (e_i)_{i=1, \dots, n} \text{ von Kanten } e_i = (p_i, q_i) \in B^2 \text{ wenn gilt:}$$

$$\forall e_i = (p_i, q_i) \in L \quad p_i \in R \ \& \ q_i \notin R$$

Definition (10): Eine Folge $(b_i)_{i=1, \dots, n}$ von Bildpunkten $b_i \in B$ heißt *Rand* der Region R genau dann, wenn für alle Kanten $e_i = (p_i, q_i)$ der geschlossenen Randlinie $L = (e_i)_{i=1, \dots, m}$ von R gilt:

$$\exists_{1 \leq j \leq n} p_i = b_j \ \& \ j \leq i$$

3.8 Bemerkungen zu den Definitionen:

Die Begriffe *Kante*, *Linie* und *Rand* sind in der Literatur nicht einheitlich definiert. Daher wurden hier Definitionen formuliert, welche einander nicht überschneiden, aber auch nicht im Gegensatz zu jenen in der Literatur bekannten Definitionen stehen. Ein wesentlicher Unterschied zu bisherigen Definitionen besteht im Versuch, von einer mathematischen Formulierung der Nachbarschaftsbeziehungen auszugehen.

Zur Illustration für bisher gebräuchliche Definitionen, zum Beispiel der "Kante", seien im folgenden zwei Beispiele gebracht.

Beispiel (1) ist der sehr wichtigen Arbeit von Davis (1973) entnommen, wonach: "Eine Kante" die Grenze zwischen zwei Regionen, R_1, R_2 von verschiedenen konstanten Grauwertbereichen ist." Diese der physikalischen Realität entnommene Kantendefinition kann man auf digitale Bilder in der zuvor benutzten Schreibweise folgend übersetzen:

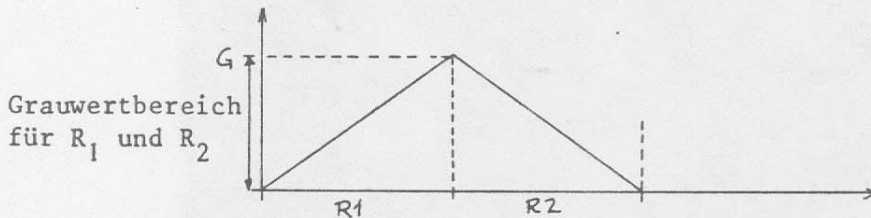
B sei der Bildbereich, R_1 und R_2 zwei Regionen (Def.(8)) mit $R_1 \cap R_2 = \emptyset$ in B, die eine gemeinsame Randlinie L (Def. (9)) haben:

$$(p, q) \in L \iff p \in R_1 \ \& \ q \in R_2 .$$

Gilt außerdem:

$$(3.6) \quad \exists a < b < c \in G \left(\bigvee_{x \in R_1} \bigvee_{x_2 \in R_2} a \leq g(x_1) \leq b \leq g(x_2) \leq c \right) \vee \left(\bigvee_{x \in R_1} \bigvee_{x_2 \in R_2} a \leq g(x_2) \leq b \leq g(x_1) \leq c \right)$$

dann bezeichnet Davis (1973) alle $(p, q) \in L$ als "Kanten". Probleme bei der Erkennung ergeben sich mit dieser Definition, wenn die Trennung der Grauwertbereiche der Regionen R_1 und R_2 mit Formel (3.6) nicht möglich ist. Als Beispiel sei die von Davis (1973) erwähnte "Dachkante" angeführt, deren Grauwertbereiche identisch sind.



Figur 3: Dachkante

Beispiel (2) ist einem Algorithmus von Andrus et al (1974) entnommen und zeigt die Verwendbarkeit unseres Konzepts für multispektrale Bilder. Hier ist $B \in \mathbb{N}^3$, wobei die 3. Komponente der Nummer des Spektralbandes entspricht. Basierend auf der Nachbarschaft

$$\Gamma(x, y, k) := \left\{ (u, v, k) \mid \sum_{i=1}^w |u-x| + |v-y| = 1 \right\}$$

(vgl. Γ_4) wird die Gewichtsfunktion w einer Kante folgend berechnet (n sei die Anzahl der Bänder):

$$w((x_1, y_1, k_1), (x_2, y_2, k_2)) := n^{-1} \sqrt{\sum_{i=1}^n (g(x_2, y_2, i) - g(x_1, y_1, i))^2}$$

(Beachte: w ist unabhängig von den Kanälen k_1 und k_2 der Bildpunkte!)

Auch die Richtungen der Kanten $((x_1, y_1, k_1), (x_2, y_2, k_2))$, horizontal ($x_1 = x_2$) und vertikal ($y_1 = y_2$), haben eine entscheidende Bedeutung für den Algorithmus von Andrus et al (1974).

Wie die beiden Beispiele (1) und (2) zeigen, kann das hier entwickelte Konzept auf eine Vielzahl von verschiedensten Algorithmen der Kanten- und Liniensuche in digitalen Bildern angewendet werden.

4. KLASSIFIKATION VON ALGORITHMEN, DIE ZUR ENTDECKUNG VON KANTEN UND LINIEN FÜHREN

Ein Algorithmus beschreibt im allgemeinen einen Prozeß, etwa in der Form einer Programmiersprache. Unter einem Prozeß soll jede Verkettung von Operationen verstanden werden, deren zeitliche Ausführung durch einen Operator oder Automaten gesteuert wird.

Algorithmen zur Kanten- und Linienentdeckung sind oft komplex und setzen sich meist aus einer Folge von Teilalgorithmen zusammen, die im folgenden klassifiziert werden. Tabelle 2 zeigt ein Unterteilungsschema solcher in sich einheitlicher Teile eines Kanten- bzw. Liniensuchalgorithmus, wobei Einteilungen von Davis (1973), Pratt (1978) und Vanderbrug (1973) verarbeitet werden.

4.1 Parallele Algorithmen

"Parallel" wird eine Folge von Operationen O_1, O_2, \dots, O_n dann genannt, wenn sich das Ergebnis bei gleicher Eingabe nicht ändert, wenn man die Reihenfolge der Operationen O_1, O_2, \dots, O_n verändert.

Diese Operationen beziehen sich im Fall der Kanten- und Liniensuche auf Bildpunkte $b \in B$ und eine Umgebung von b k -ter Ordnung $U^k(b)$:

$$U^k(b) := \{ x \in B \mid x \in \Gamma^i(b), \quad i \leq k \}$$

und werden daher auch als "lokal" bezeichnet. Als Ergebnis wird ein neuer Wert $g_1(b)$ des Bildpunktes b berechnet.

$$g_1(b) := O_b(g(U^k(b))),$$

wobei die zu einem Bildpunkt b (oder Umgebung $U(b)$) gehörenden Grauwerte nach Gl. (3.1) mit $g(b)$ (oder $g(U(b))$) bezeichnet werden.

Der parallele Algorithmus besteht darin, für jeden Bildpunkt b diese Operation O_b auszuführen. Da diese Operationen voneinander unabhängig sind, können sie (theoretisch) alle gleichzeitig (=parallel) ausgeführt werden. Das Ergebnis wird üblicherweise wieder in Form einer Grauwertmatrix gespeichert.

Folgende Vorteile können genannt werden:

Können mehrere Operationen auf einem Computer echt parallel ausgeführt werden (z.B. wenn ein STARAN-Computer o.ä. zur Verfügung steht), so sind diese Algorithmen wegen ihrer Einfachheit sehr wirksam. Das Ergebnis ist eine Grauwertmatrix, welche leicht kontrolliert werden kann, da die Datenstruktur jene des Eingabebildes ist und mit denselben Routinen ausgedruckt werden kann.

Als Nachteil muß gelten:

Bei sequentieller Implementation sind parallele Algorithmen den echt sequentiellen Algorithmen unterlegen. Eine sequentielle Implementation wäre das zeilen- und spaltenweise Abarbeiten der Grauwertmatrizen.

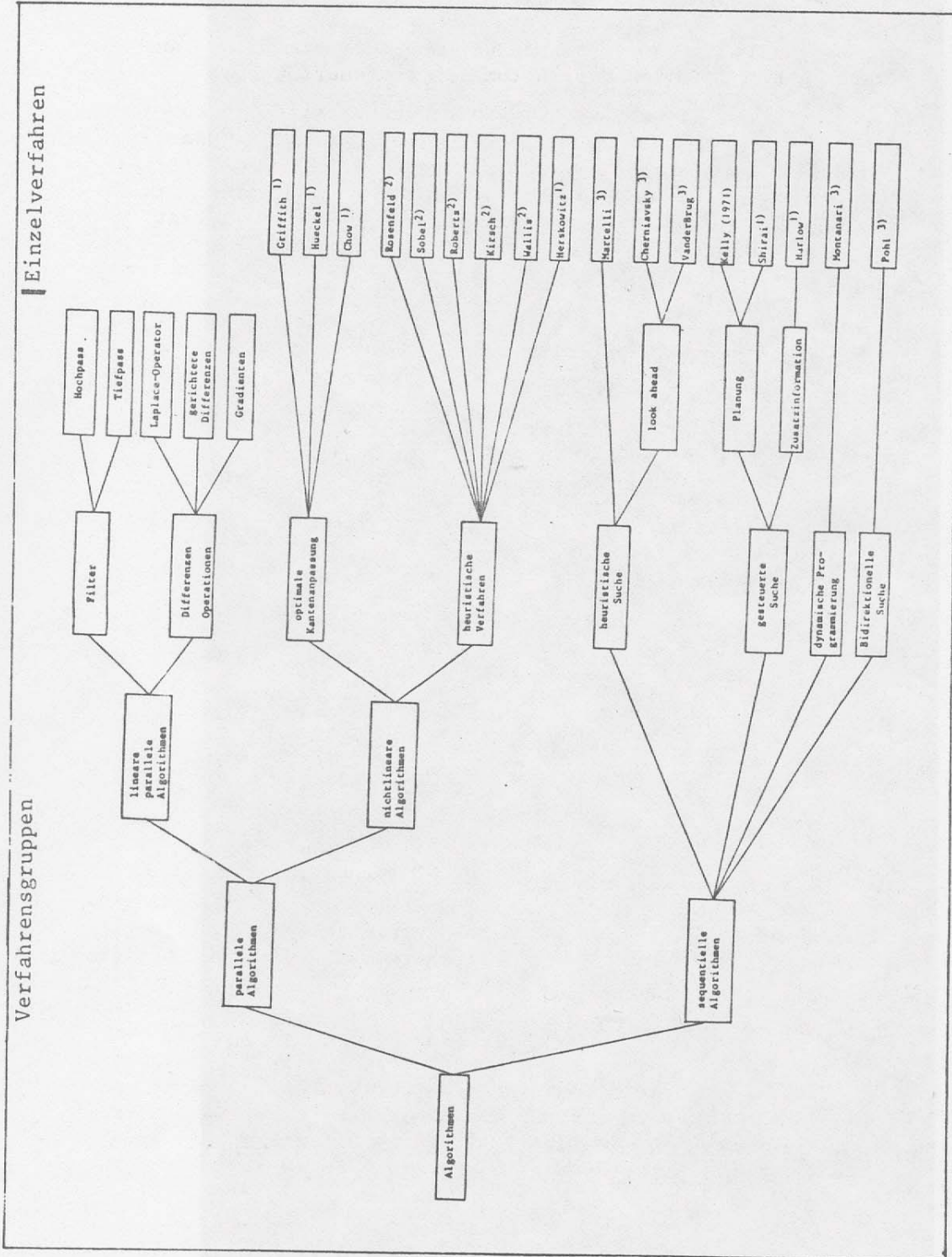


Tabelle 2: Kanten- und Liniensuchverfahren. ¹⁾ aus : Davis (1973);
²⁾ aus: Pratt (1978); ³⁾ aus: Vanderbrug (1973).

Parallele Algorithmen werden sehr oft in der Vorverarbeitungsphase zur Bildverbesserung und Kantenverstärkung verwendet. Daran wird dann ein sequentieller Algorithmus zur Liniensuche angeschlossen. Dieser wird in den wenigsten Fällen alle Bildpunkte durchsuchen müssen, während die parallele Vorverarbeitung prinzipiell alle Bildpunkte transformiert, also zum Teil Transformationen ausführt, welche für das Endergebnis nicht notwendig wären. In solchen Fällen wäre es angebracht, die Operatoren der parallelen Algorithmen in den sequentiellen Algorithmus einzubauen, sodaß nur die Bildpunkte transformiert werden, die vom sequentiellen Algorithmus verwendet werden.

4.1.1 Lineare parallele Algorithmen

Parallele Algorithmen, die aus einer Folge linearer Operatoren bestehen, werden als linear bezeichnet. Der Operator $O: g(B) \rightarrow g_1(B)$ heißt linear, wenn

$$\forall x, y \in g(B) \quad \forall a, b \in \mathbb{R} \quad O(ax+by) = aO(x) + bO(y)$$

Unter den linearen, parallelen Kantensuchalgorithmen findet man klassische Verfahren, welche in Handbüchern der Bildverarbeitung beschrieben sind (Pratt, 1978): Hochpass-Filterung, gerichtete numerische Differentiation, numerische Gradienten, Laplace-Operator. Diese Verfahren lassen sich durch eine einfache Operation g_m mit Masken m darstellen.

Unter einer Maske versteht man eine (lokale) Funktion $m_x: U(x) \rightarrow \mathbb{R}$ auf einer Umgebung $U(x)$ eines Bildpunktes $x \in B$. Für den Wert $m_x(b)$ von $b \in U(x)$ ist nur die relative Lage von b bezüglich x ausschlaggebend.

Die Operation g_m berechnet sich dann wie folgt:

$$g_m(b) = \sum_{x \in U(b)} g(x) \cdot m_b(x)$$

Im folgenden seien einige Beispiele für Masken angegeben, welche sich auf den 2-dimensionalen Bildbereich

$B = R \times C$ und die Nachbarschaft Γ_g beziehen:

(a) $\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$ horizontale Vorwärtsdifferenz auf $U^1(x)$ (siehe z.B. Eberlein (1976) und O'Gorman & Clowes (1976));

(b) $\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$ Laplace-Operatormaske auf $U^1(x)$ (siehe z.B. Carton, 1973);

(c) $\begin{matrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{matrix}$ doppelte horizontale Vorwärtsdifferenz auf $U^1(x)$ (vgl. Pratt, 1978)

(d) $\begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{matrix}$ entspricht einer Tiefpaßfilterung

(e) Montoto (1977) verwendet iterativ eine Art Laplace-Operator:

$$m_b(b) = (\text{card}(U(b)) - 1) / \text{card}(U(b)) \quad \text{und}$$

$$m_b(x) = -1 / \text{card}(U(b)) \quad \text{für } x \neq b$$

Der Unterschied zum Laplace-Operator (b) besteht darin, daß für Umgebungen $U^i(x)$, $i > 1$ nur der Mittelpunkt b ausschlaggebend ist und alle anderen Punkte der Umgebung gemittelt werden, während beim Laplace-Operator auch die Umgebungspunkte unterschiedlich gewichtet werden. Für eine 9×9 Umgebung U^4 verwendet Montoto (1977) folgende Maske:

$$\begin{array}{cccccccc} -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} \\ -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} \\ -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} \\ -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} \\ -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} \\ -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} \\ -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} \\ -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} \\ -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} \end{array}$$

→ Laplace-Operator
für Γ_8

4.1.2 Nicht lineare parallele Algorithmen

Hier findet man auf nicht-linearen Funktionen basierende Verfahren. Bekannte Operatoren wurden von Roberts, Sobel, Kirsch, Wallis und Rosenfeld angegeben. Eine detaillierte Behandlung bringt Pratt (1978). Nicht-lineare Operatoren sind rechenintensiver als lineare, daher beschränken sich die Methoden meist auf Umgebungen $U^1(x)$ unter Verwendung der 8-Nachbarschaft. Einzig Rosenfeld verarbeitet größere Umgebungen:

$$U^m(x,y) = \left\{ (u,w) \mid (u=x \ \& \ -m \leq w-y < m) \vee (w=y \ \& \ m \leq u-y < m) \right\}$$

wobei $m = 2^k$, $k \in \mathbb{N}$. Darauf wendet er einen gemittelten Differenzenoperator an und bildet dann das Produkt über diese Differenzen für Werte von $m \leq M$.

Eine andere Gruppe von Verfahren geht von einem Modell einer Kante aus, das mit den Mitteln der Statistik und der Optimierungstheorie angenähert wird. Hiezu zählen Methoden von Griffith, Hueckel und Chow, welche von Davis(1973) behandelt werden.

4.2 Sequentielle Algorithmen

In sequentiellen Algorithmen ist die Reihenfolge der Operationen O_1, \dots, O_n von entscheidender Bedeutung für das Ergebnis. Die Informationen, die bis zur Operation O_i , $i < n$ errechnet werden, werden für die Ausführung der Operation O_{i+1} verwendet.

Bei den sequentiellen Algorithmen kann die Bedeutung der Nachbarschaftsfunktion $\Gamma(x)$ ermessen werden, wie der folgende allgemeine Liniensuchalgorithmus zeigt.

4.2.1 Ein genereller sequentieller Liniensuchalgorithmus

Um das Konzept einer sequentiellen Liniensuche zu beschreiben, wird in Tabelle 3 ein allgemeiner Algorithmus angegeben. Es wird von einer Menge X ausgegangen, in der die Lösung gesucht wird. Es treten die folgenden Bezeichnungen und Größen auf:

$$x_0, x_n \in X$$

last(x) eine zu einem Punkt $x \in X$ gehörende Größe, die auf den Vorgänger von x zeigt ("Zeiger");

nil leerer Zeiger (für das Startelement x_0);

$\psi(x) \in \mathbb{R}$ minimaler Abstand des Elements x von x_0 in Kostenschritten ;

$S \subset X$ Menge der Alternativelemente, die zur Auswahl stehen,

- $\Gamma^1(x) \subset X$ Nachbarmenge auf X ;
 end(x) logische Funktion mit den Werten .true. für Ziel-
 punkte und .false. sonst;
 $c(x,y) \in \mathbb{R}$ Kostenfunktion (Abstand) zwischen den zwei benach-
 barten Elementen x und y ;
 $h^1(x) \in \mathbb{R}$ reellwertige Funktion, als "Kostenschätzung" oder
 "Heuristik" bezeichnet, für den minimalen Weg
 von x zum Ziel;
 $\Gamma(\text{nil}) = \emptyset$ die leere Menge (per Definition).

```

begin
 $x_n := x_0$ ; last( $x_0$ ): = nil;
for all  $x \in X$  do  $\varphi(x) := \infty$  ;  $\varphi(x_0) := \phi$ ;
S: = {  $x_0$  } ;
while  $\neg$  end( $x_n$ ) &  $S \neq \emptyset$  do
    begin
        > S: = S - {  $x_n$  }; EXCLUDE( $x_n$ )
        for all  $x \in (\Gamma^1(x_n) - \Gamma^1(\text{last}(x_n)))$  do
            begin
                 $\varphi' := \varphi(x_n) + c(x_n, x)$  ;
                if  $\varphi' < \varphi(x)$  then
                    begin
                         $\varphi(x) := \varphi'$  ;
                        S: = S  $\cup$  {  $x$  } ; last(x): =  $x_n$ 
                    end
                end;
            m: =  $\infty$  ;
            for all  $x \in S$  do
                if  $\varphi(x) + h^1(x) < m$  then
                    begin
                        m: =  $\varphi(x) + h^1(x)$ ;  $x_n := x$ 
                    end
                end
            end
        end
    end
end

```

Tabelle 3: Allgemeine Formulierung der sequentiellen Liniensuche

Ergebnis:

1) Wenn es einen Weg von x_0 zu einem Zielpunkt gibt:

(*) $\text{end}(x_n)$

(*) $\exists x_1, x_2, \dots, x_{n-1} : \forall_{1 \leq i \leq n} \text{last}(x_{i+1}) = x_i \text{ \& } (x_0, x_1, \dots, x_n) \text{ ist ein Lösungsweg;}$

(*) Der Lösungsweg (x_0, x_1, \dots, x_n) ist optimal bezüglich der Kostenfunktion c und $\psi(x_n)$ sind die Kosten für den Lösungsweg.

2) Wenn es keinen Weg von x_0 zu einem Zielpunkt gibt:

(*) $S = \emptyset \text{ \& } \neg \text{end}(x_n).$

4.2.2 Wahl der Parameter für spezielle Suchverfahren

Bei der *heuristischen Suche* einer Linie werden die Funktionen $c(x,y)$ und $h'(x)$ auf Grund der Grauwerte $g(x)$ und $g(y)$ bestimmt. Martelli (siehe VanderBrug, 1973; Tabelle 2) verwendet als Grundmenge \mathcal{X} die Menge aller auf $B=R \times C$ möglichen Kanten. Die Nachbarmenge Γ entspricht den \mathcal{L} von Definition (7) mit folgenden Spezifikationen:

$$\forall \text{ Kanten } (x,y) : = \left\{ (x,a), (a,b), (b,y) \mid a \in \Gamma_4^1(x) \cap \Gamma_8(y), b \in \Gamma_4^1(y) \cap \Gamma_8(x) \right\}$$

Die Kostenfunktion berechnet sich aus

$$c((P_1, Q_1), (P_2, Q_2)) := \max(\emptyset, 2.M - (g(P_1) + g(P_2)) + (g(Q_1) + g(Q_2)))$$

wobei M als Kontrast $g(P) - g(Q)$ einer optimalen Kante (P, Q) gesetzt wird. Als Startwert wird eine Kante $x_0 = ((1, s_0), (1, s_0 + 1))$ in der 1. Zeile des Bildes gewählt und die Abbruchfunktion lautet

$$\text{end}((P_1, Q_1), (P_2, Q_2)) = (P_1 = \text{card}(R)) \text{ \& } (P_2 = \text{card}(R))$$

Folgende Heuristik h' wird verwendet:

$$h'((P_1, Q_1), (P_2, Q_2)) = \begin{cases} \emptyset & P_1 \geq \max \left\{ P \mid (P, Q) \in S \right\} - 3 \\ \infty & \text{sonst} \end{cases}$$

Ein anschauliches Beispiel für den Algorithmus findet sich im Anhang B.

Für die von VanderBrug (1973) beschriebenen Look-Ahead- und Cherniavsky-Algorithmen wird $h'(x)$ durch Schätzung der Kosten innerhalb der Umgebung $U(x)$ berechnet.

Eine weitere effiziente Gruppe von Suchverfahren wird unter dem Begriff *gesteuerte Suche* zusammengefaßt. Hier wird die Suche durch (für den Algorithmus) äußere Informationen gesteuert, die den Wert von $h'(x)$ bestimmen und so den Ablauf der Suche beeinflussen. Eine spezielle Form der gesteuerten Suche ist die *Planung* (Kelly, Shirai, siehe Tabelle 2). Dabei sind für den Algorithmus aus einem vorherigen Arbeitsgang Zwischenziele bekannt, über die die Lösung laufen soll. Es muß also h' so gewählt werden, daß es womöglich immer den Abstand bis zum nächsten Zwischenziel angibt.

Harlow (Tabelle 2) gab einen Algorithmus an, bei dem die Information für die gesteuerte Suche aus einer zuvor bekannten Beschreibung der Bildszene stammt.

Das Verfahren der *dynamischen Programmierung* nach einer Methode von Montanari (Tabelle 2) ist im Grunde eine Vereinfachung des generellen Algorithmus durch Setzen von

$$\forall_{x \in X} h'(x) = \emptyset$$

Weiters wird die Nachbarmenge $\Gamma(x)$ noch durch Bedingungen für die Grauwerte $g(y)$ für $y \in \Gamma(x)$ eingeschränkt. Außerdem liegt hier nicht die Menge der Kanten (siehe Definition (6a)) zugrunde, sondern $X = B$.

Ein Verfahren, das nicht ganz in das Schema der generellen sequentiellen Liniensuchalgorithmen paßt, ist die *bidirektionelle Suche*. Es wird dabei von einem Startpunkt und einem Zielpunkt gleichzeitig ausgegangen und je eine Menge S für den Startpunkt x_0 (wie im Algorithmus 4.2.1) und eine Menge T für den Zielpunkt berechnet. Am Prinzip des Suchalgorithmus ändert sich dadurch nur die Funktion $\text{end}(x)$, die bei der bidirektionellen Suche meist sehr komplex wird. Es kann nämlich $\text{end}(x)$ nur wahr sein für $x \in T \cap S$! Pohl (vgl. Tabelle 2) hat diesen Typ von Suchalgorithmus untersucht.

4.2.3 Vor- und Nachteile sequentieller Algorithmen

Sequentielle Algorithmen sind dafür konzipiert, aus einer Menge von möglichen Lösungen die optimale zu finden. Wie rasch das Ergebnis gefunden wird, hängt zum Großteil von der Wahl der "Kostenschätzung" h' ab. Wäre die perfekte Heuristik h bekannt, so benötigte der Algorithmus mit $h' = h$ für das Finden der Lösung (x_0, x_1, \dots, x_n) genau n Schritte. Ein Nachteil dieser Algorithmen liegt darin, daß bei Fehlen einer Lösung alle Möglichkeiten untersucht werden und somit ein sehr großer Rechenaufwand notwendig wird.

5. SCHLUSS

Wir haben hier einige Grundsätze und Verfahren der Kanten- und Liniensuche zusammenfassend dargestellt, wobei wir von mathematischen Definitionen des *Bildes*, der *Nachbarschaft* usw. ausgegangen sind. Es wird eine Unterscheidung in parallele und sequentielle Algorithmen vorgenommen, wobei parallele Verfahren Kanten und sequentielle Methoden Linien suchen.

Zum Problem der Kanten- und Liniensuche bestehen sehr viele verschiedene Lösungen, zu denen hier eine geschlossene Darstellung vorgestellt wird. Wir behandeln insbesondere die linearen parallelen Verfahren der Kantenverstärkung, welche durch Einführung einer Grauwertschwelle (Äquidensitendarstellung) zur Kantensuche ergänzt wird. Es wird überdies ein allgemeiner sequentieller Liniensuchalgorithmus formuliert, aus dem mehrere besondere Methoden ableitbar sind.

Mit der vorliegenden Arbeit wird ein Beitrag zur mathematischen Behandlung einer grundlegenden Aufgabe der digitalen Mustererkennung und Bildverarbeitung geliefert. Diese Arbeitsgebiete gewinnen derzeit im Rahmen der Photogrammetrie und Fernerkennung wesentliches Gewicht.

DANK

Diese Arbeit wurde unter anderem aus Grant DAERO-78-G-044 des U.S. European Research Office, London, gefördert.

REFERENZEN:

- Aleksander I. (1978): Improving Patterns of Recognition, Computing Europe, 6 Febr. 1978, p.18.
- Andrus Jan F. et al (1974): Digital Image Registration Methods Using Boundary Maps, IEEE Transactions on Computer, September.
- Carton E.J. et al (1974): Some Basic Edge Detection Techniques, Comp. Science Center, TR-277, Univ. of Maryland, USA
- Davis L.S. (1973) : A Survey of Edge Detection Techniques, Comp. Science Center, TR-273, Univ. of Maryland, USA
- Eberlein R.B. (1976): An Iterative Gradient Edge Detection Algorithm, Comp. Science Center, MD 20742, Univ. of Maryland, USA

- Fram J., E.Deutsch (1975): On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance, IEEE Trans. on Computers, Vol. C-24, No.6, pp. 616-628
- Kaneko T. (1976): Evaluation of LANDSAT, Image Registration Accuracy, Photogramm. Eng. and Remote Sensing, Vol. 42, pp. 1285-1209
- Kelly M.D. (1971): Edge Detection in Pictures Using Planning, Machine Intell. 6, Edinburgh Univ. Press, Scotland
- Kreiling W. (1976): Automatische Auswertung in Stereobildern durch digitale Korrelation, Pres.Paper, Comm.III, 13th Congress of ISP, Helsinki, Finland
- Martelli (1972): Edge Detection Using Heuristic Search Methods, Computer Graphics and Image Processing, Vol. 1, No.2, pp.169-182
- Montoto (1977): Digital Detection of Linear Features in Satellite Imagery Proceedings, Intl.-Symp., Image Proc., Graz,Austria
- Nack, M.L. (1975): Temporal Registration of Multi-Spectral Digital Images Using Their Edge Images, Goddard Space Flight Center, Contract NAS 5-1199, NASA, USA
- O'Gorman F., M.Clowes (1976): Finding Picture Edges Through Collinearity of Feature Points, IEEE Transactions on Computers, Vol. C-25, No.4
- Pratt (1978): Digital Image Processing, Image Enhancement, Publ. by J.Wiley & Son, 1978
- Roos M. (1975): Automatic Reseau Measuring Equipment (ARME), Photogr. Eng. & Remote Sensing, Vol. 41, pp. 1109-1115
- Rosenfeld A.(1970): Connectivity in Digital Pictures, Journal of the Assoc. for Comp. Mach, Vol.17, No.1, pp.146-160
- Rosenfeld A., L.Davis (1975):A Note on thinning, TR-381, Comp.Science Center, Univ.of Maryland, USA
- VanderBrug G.J. (1973):Edge Following in Noisy Pictures, Comp.Science Center, TR-273, Univ. of Maryland, USA

ANHANG A

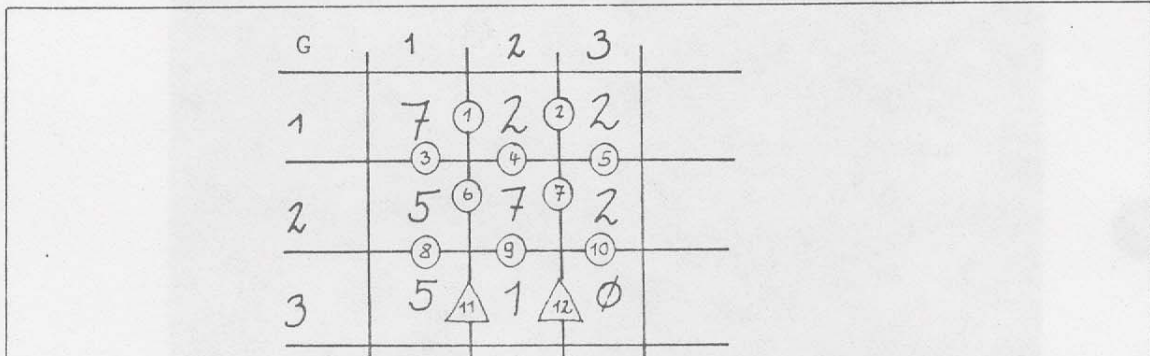
Liste der verwendeten Symbole

C	Teilmenge von
N	natürl. Zahlen
$:=$	Definition
$f:A \rightarrow B$	Abbildung einer Menge A in die Menge B
card	Kardinalzahl
\in	aus, Element von
\notin	nicht aus, nicht enthalten in
\iff	Äquivalent
Z	Menge der ganzen Zahlen
$\forall_{x \in B}$	Für alle x aus B
$\{ \dots \}$	Mengenbezeichnung
\emptyset	leere Menge
\cap	Durchschnitt von 2 Mengen (Elemente, die beiden angehören)
\cup	Vereinigung zweier Mengen
\exists	Es gibt
$\exists!$	Es gibt genau ein Element x
$ $	für welches
\implies	daraus folgt
$!!$ $a < b$	Es ist zu zeigen, daß a kleiner als b ist.
\vee	logisches "oder"
$\&$	logisches "und"

ANHANG B

Ein Beispiel für die Ausführung des sequentiellen Liniensuchalgorithmus von Martelli (1972)

Es wird eine Linie von oben nach unten auf einem Bildausschnitt (Figur B.1) gesucht:



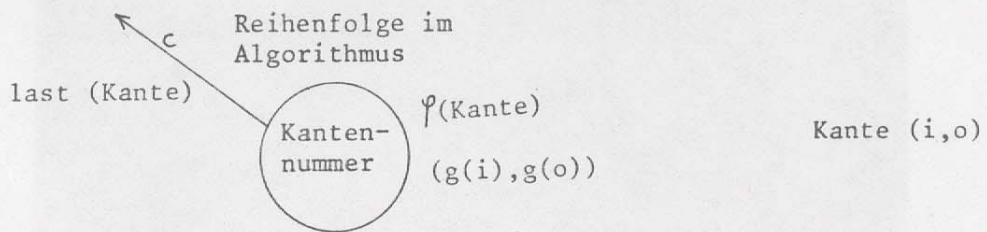
Figur B.1

In den Kästchen die Grauwerte (z.B.: $g(1,1) = 7$) und auf den Linien in Kreisen ○ und Dreiecken △ die Nummern der Kanten. △11 und △12 bezeichnen die Zielkanten.

Ausgegangen wird von Kante ①. Als Kostenfunktion wird zunächst

$$(B.1) \quad c((P_1, Q_1), (P_2, Q_2)) = 14 - (g(P_1) + g(P_2)) + (g(Q_1) + g(Q_2))$$

gewählt. h' wirkt sich in diesem Beispiel nicht aus, da nur 3 Zeilen betrachtet werden. Im folgenden Schema wird der Ablauf des Algorithmus durch folgende Bezeichnungen beschrieben.



Die Lösung ist in Figur B.2 dargestellt. Sie ist durch Doppelpfeile gekennzeichnet. Zum Vergleich verschiedener Varianten dieser Algorithmen können folgende Werte herangezogen werden:

- Anzahl Schritte: 9
- Anzahl berechneter Alternativen: 14
- gefundene Lösung: ①, ④, ⑦, ⑨, △11
- Kosten für den Lösungsweg: $\psi(\triangle 11) = 15$
- Alternativen S nach dem Finden der Lösung:

$$S = \{ \textcircled{8}^{19}, \textcircled{12}^{16}, \textcircled{10}^{19} \}$$

Figur B.3 enthält zum Vergleich dasselbe Beispiel mit der Kostenfunktion:

$$(B.2) \quad c((P_1, Q_1), (P_2, Q_2)) = 1\emptyset - (g(P_1) + g(P_2)) + g(Q_1) + g(Q_2)$$

Wir finden hier:

Anzahl Schritte: 5

Anzahl berechneter Alternativen: 13

gefundene Lösung: ①, ④, ⑦, ⑨, $\triangle 11$

Kosten für den Lösungsweg $\psi(\triangle 11) = \emptyset$ (!)

Alternativen S nach dem Finden der Lösung:

$$S = \{ \textcircled{3}^3, \textcircled{6}^2, \textcircled{8}^4, \textcircled{12}^4, \textcircled{10}^7, \textcircled{5}^5, \textcircled{2}^5 \}.$$

In diesem speziellen Beispiel ist also die Wahl der Kostenfunktion (B.2) vorzuziehen; sie liefert die Lösung mit der geringsten Anzahl Schritte.