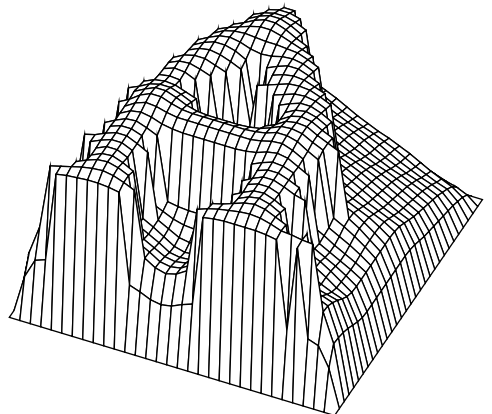
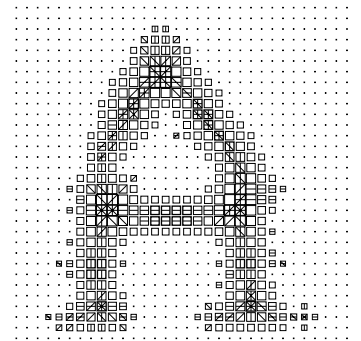
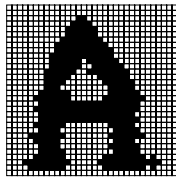


Fuzzy-Skelettierung: Ein Verfahren zur Skelettierung von Bildern

Das Verfahren läßt sich auf Farb-, Grauwert- und Schwarz-Weiß-Bilder anwenden. Zweck des Verfahrens ist die Extraktion derjenigen Bildpunkte (Pixel), die für eine nachfolgende Vektorisierung geeignet sind. Dazu wird jedem Pixel ein Startwert zugeordnet, von dem dann pro Iterationsschritt für jedes Nachbarpixel N iterativ ein bestimmter Wert subtrahiert oder addiert wird. Dies erfolgt unter der Bedingung, daß der Wert $W(N) \neq$ Startwert oder die Farbe unterschiedlich ist. Das Ergebnis kann als dreidimensionale "Gebirgslandschaft" $z = W(\text{Pixel}(x,y))$ betrachtet werden, deren Höhenzüge (Gipfel und Grate) die gesuchten Pixel enthalten. Die Linienbreite ist über die Höhe des Pixels bestimmbar, was die Trennung von Linien verschiedener Breiten mithilfe einer Unterteilung in Höhenintervalle leicht ermöglicht. Die bevorzugte Realisierung des Verfahrens kann aufgrund ihrer Einfachheit auch als IC-Transistor-Schaltung ausgeführt werden, so daß bereits aus der Aufnahme-Einheit (Scanner, Kamera) der Gewinn des Linienskeletts quasi zeitgleich zum Originalbild (Verzögerungszeit $< 1\text{ms}$) möglich ist.

Einsatzgebiete: Zeichnungs-, Text-, Mustererkennung,
maschinelle Augen für Roboter



1 Patentbeschreibung

Die Erfindung ist ein neuartiges Verfahren, durch das diejenigen Pixel (Bildpunkte) eines (üblicherweise digitalen) Farb-, Grauwert- oder Schwarz-Weiß-Bildes herausgefiltert werden, die für ein nachgeschaltetes Verfahren der Mustererkennung von Bedeutung sind. Als wesentliches Einsatzgebiet des Verfahrens ist die Bestimmung der Pixel zu nennen, die auf der Mittellinie einer langgestreckten, unregelmäßig berandeten ("zerklüfteten") Fläche liegen. Solche Flächen entstehen z. B. bei der Aufnahme von Schriftstücken oder technischen Zeichnungen durch einen Scanner oder eine (elektronische) Kamera.

Die Extraktion von Linienmerkmalen geschieht nach aktuellem Stand der Technik gemäß folgender drei Prinzipien:

1. Transformation in den Frequenz-Raum (Fourier-Arithmetik)
2. Geometrische Ermittlung der Mittellinien (z. B. Mittellinienbestimmung nach [ogen])
3. Skelettierung durch Entfernung der Randpunkte, bis die Strukturen max. 1-2 Pixel breit sind (z. B. Verfahren nach [chin])

Die Verfahren nach 1. sind mathematisch, folglich auch technisch aufwendig und dabei im Ergebnis für Nicht-Experten wenig anschaulich. Dies ist dadurch verursacht, daß ein Pixel des Original-Bildes nicht eindeutig (bijektiv) einem Punkt im Frequenz-Raum zugeordnet wird, sondern einer Folge von Bildpunkten Punkte im Frequenz-Raum zugeordnet werden.

Die Verfahren nach 2. werden mit zunehmender Zerklüftung der Ränder, welche von der Qualität der Vorlagen abhängt, zumindest aufwendiger in der Durchführung, meist jedoch auch komplexer oder sogar instabil im Ergebnis.

Die Ergebnisse nach Verfahren 3. sind als Ausgangspunkt für eine nachgeschaltete Vektorisierung am günstigsten, führen aber zu aufwendigen Mustervergleichen und komplettem Informationsverlust über die Linienbreite. Außerdem ist die Rückgewinnung des Originalbildes, das im Ergebnis nicht enthalten ist, ebenfalls aufwendig.

Das Ergebnis, welches die Anwendung eines erfindungsgemäßen Verfahrens liefert, weist folgende Eigenschaften auf:

1. Das Originalbild ist enthalten.
2. Der für jedes Pixel bestimmte Höhenwert ist ein direktes Maß für den Abstand des Pixels zum Rand der Fläche.
3. Die Breite der Fläche in den Punkten auf den Mittellinien ist ebenso wie die Lage dieser Linien selbst wegen 2. ebenfalls durch den Höhenwert direkt bestimmbar.

Ferner gilt für das Verfahren, daß es

4. stabil gegenüber zerklüfteten Rändern bzgl. Aufwand und Ergebnis ist,
5. algorithmisch so einfach ist, daß es durch einfache IC-Transistor-Schaltungen direkt in der Aufnahme-Einheit realisiert werden kann, wodurch eine
 5. a) parallele Verarbeitung aller Pixel möglich ist, was zu
 5. b) Rechenzeiten von $< 1\text{ms}$ für das gesamte Bild führt und

6. seinen Einsatz innerhalb eines maschinellen Auges (z. B. für Roboter) aufgrund des minimalen Platzbedarfs erlaubt

Das Verfahren wird in zwei Bearbeitungsstufen durchgeführt:

1. Bestimmung von Höhenwerten durch Iteration
2. Bestimmung der Mittellinienpixel als lokale Maximalwerte der Flächen oder von bestimmten Schnittmengen der Pixelumgebung

In der ersten Stufe wird jedem Pixel zunächst ein Höhenwert als Startwert zugeordnet, der dann abhängig von Farbe und Höhe der Nachbarpixel in jedem der nachfolgenden Iterationsschritte reduziert wird. In S/W-Bildern kann man auch Vordergrund und Hintergrund trennen und in der Höhe aufeinander "zuwachsen" lassen. Die Iteration kann entweder auf eine bestimmte Anzahl von Schritten fixiert oder solange durchgeführt werden, bis eine Abbruchbedingung eintritt. Als Abbruchbedingung kann z. B. das Erreichen oder Unterschreiten/überschreiten eines minimalen/maximalen Höhenwertes verwendet werden.

Ist die Iteration abgeschlossen, lassen sich die Mittellinienpixel in der zweiten Stufe als Pixel mit maximalem Wert bezüglich der Höhe auf einer vollständigen oder durch einen Schnitt reduzierten Umgebung des Pixels bestimmen.

Im folgenden soll die Erfindung zunächst anhand des bevorzugten Ausführungsbeispiels erläutert werden, bevor die allgemeingültige mathematische Beschreibung für die Patentansprüche aufgestellt wird.

1.1 Bevorzugte Ausführung

Die bevorzugte Ausführung arbeitet auf S/W-Bildern, die aus der digitalen Erfassung von Texten und technischen Zeichnungen erzeugt werden und sich aus gleich großen, quadratischen Pixeln zusammensetzen. Als Nachbarpunkte werden nur die acht unmittelbar angrenzenden Pixel (horizontal, vertikal und diagonal) berücksichtigt (siehe **Figur 1**). Schwarze Pixel bilden den Vordergrund, weiße den Hintergrund.

Jedes schwarze Pixel erhält als Startwert 255, jedes weiße 0. Der Wert eines schwarzen Pixels wird in jedem Iterationsschritt pro Nachbarpixel, dessen Höhe $\neq 255$ ist, um 1 reduziert. War der Wert des Pixels zu Beginn dieses Iterationsschrittes bereits $\neq 255$, wird nochmals 1 abgezogen. Maximal wird der Wert dadurch um 9 Einheiten pro Iterationsschritt reduziert.

Für Werte $\neq 0$ wird der Wert weißer Pixel entsprechend um jeweils 1 erhöht. Der Wert weißer Pixel wächst demzufolge um maximal 9 Einheiten pro Iterationsschritt.

Damit Vordergrund und Hintergrund eindeutig trennbar sind, werden für schwarze Pixel 128 als minimaler und für weiße Pixel 127 als maximaler Wert festgelegt.

Die reduzierten Umgebungen werden wie in **Figur 2** gewählt, d. h. aus den Umgebungen werden nur die Nachbarn einer Richtung (horizontal, vertikal, diagonal) berücksichtigt.

Alle Punkte, die den maximalen (Hintergrund: minimalen) Wert bezüglich mindestens einer der neun Umgebungen \mathcal{U} , $\mathcal{R}_1 \dots \mathcal{R}_8$ besitzen, bilden das sogenannte "Fuzzy-Skelett", welches die Mittellinienpunkte enthält.

Anschaulich bilden die lokalen Maximalwerte die Gipfel und die Maxima auf den reduzierten Umgebungen die Grate des Höhengebirges.

(Siehe Anwendungsbeispiele: **Figuren 3 – 6**)

1.2 Mathematische Beschreibung

Da Grauwert- und Farb-Bilder gleichartig verarbeitet werden, werden ab hier beide Bild-Typen nicht mehr unterschieden und mit "Farb-Bild" bezeichnet.

Sei \mathcal{P} die Menge aller Pixel eines Bildes B . Man definiert:

$$\begin{aligned}
 &\text{für s/w-Bilder} \quad \mathcal{V} \text{ (Vordergrund)} : \quad \mathcal{V} \subset \mathcal{P} \\
 &\quad \quad \quad \mathcal{H} \text{ (Hintergrund)} : \quad \mathcal{H} := \mathcal{P} \setminus \mathcal{V} \\
 \\
 &\text{für Farb-Bilder} \quad P \in \mathcal{V}(P) : \quad \mathcal{V}(P) \subset \mathcal{P} \\
 &\quad \quad \quad P \notin \mathcal{H}(P) : \quad \mathcal{H}(P) := \mathcal{P} \setminus \mathcal{V}(P)
 \end{aligned} \tag{A}$$

Zur Vereinfachung der Gleichungen wird im folgenden nicht mehr zwischen $\mathcal{V}(P)$ und \mathcal{V} bzw. $\mathcal{H}(P)$ und \mathcal{H} unterschieden, da aus dem Kontext (farbig oder graugestuft bzw. s/w) ersichtlich ist, ob Vorder- und Hintergrund von P abhängen oder nicht.

Für $P^0 \in \mathcal{P}^0 := \mathcal{P}$ wird festgelegt (Iterationsstart):

$$\begin{aligned}
 |P^0| &:= \max && \text{für } P \in \mathcal{V} \\
 |P^0| &:= \min && \text{für } P \in \mathcal{H} \\
 &\text{mit } \max > \min \in \mathbf{R}
 \end{aligned} \tag{B}$$

(Der Fall $P \in \mathcal{H}$ kann nur in s/w-Bildern auftreten, da für farbige Bilder gemäß Definition gilt: $P \in \mathcal{V}(P) \forall P$)

$P^{m-1} \rightarrow P^m$ (Iterationsschritt):

$$\mathcal{P}^m := \{P^m := f^{m-1}(P^{m-1}) \mid P^{m-1} \in \mathcal{P}^{m-1}\}$$

Dabei ist f^m definiert durch:

$$f^m(P^m) := \begin{cases} \max(s + \epsilon, |P^m| + \text{lokAdd}(P^m)) & \text{falls } |P^m| > s \\ \min(s - \epsilon, |P^m| + \text{lokAdd}(P^m)) & \text{falls } |P^m| < s \end{cases} \tag{C}$$

mit $\min < s < \max$, $u \in \mathcal{U}_m(P^m)$, $s \in \mathbf{R}$, $\epsilon \in \mathbf{R}^{>0}$. $\mathcal{U}_m(P^m)$ ist eine beliebige, aber für jedes P^m eindeutig bestimmte Umgebung von P^m .

Die lokale Additionsfunktion ist definiert durch:

$$\text{lokAdd}(P^m) := \sum_{u \in \mathcal{U}_m(P^m)} \omega_m(u, P^m) \delta(u, P^m) \tag{D}$$

Die Gewichtsfunktion ω_m und die Bewertungsfunktion δ operieren auf folgenden Mengen:

$$\begin{aligned}\omega_m &: \mathcal{P} \times \mathcal{P} \rightarrow \mathbf{R}^{>0} \\ \delta &: \mathcal{P} \times \mathcal{P} \rightarrow \{-1, 0, 1\}\end{aligned}$$

Für s/w und farbige Bilder muß die Bewertungsfunktion δ jeweils unterschiedlich definiert werden, und zwar

für Farbe

$$\delta(u, P^m) := \begin{cases} -1, & \text{falls } (|u| < \max \vee u \in \mathcal{H}(P^m)) \\ & \wedge s + \epsilon < |P^m| \\ 0, & \text{sonst} \end{cases} \quad (\text{E1})$$

für s/w

$$\delta(u, P^m) := \begin{cases} -1, & \text{falls } |u| < \max \wedge s + \epsilon < |P^m| \\ 1, & \text{falls } |u| > \min \wedge |P^m| < s - \epsilon \\ 0, & \text{sonst} \end{cases} \quad (\text{E2})$$

Die untere Schranke für Vordergrundpixel - s - dient der sicheren Unterscheidung von Vorder- und Hintergrund in s/w-Bildern. Es gilt nämlich $\forall m, P^m \in \mathcal{P}^m$:

$$\text{entweder } |P^m| \geq s + \epsilon \text{ für } P \in \mathcal{V} \quad \text{oder} \quad |P^m| \leq s - \epsilon \text{ für } P \in \mathcal{H}$$

Die Fuzzy-Funktion *lokMax* liefert nun das lokale Maximum durch:

$$\text{lokMax}(P^m) := \max \left\{ |p^m| \mid p^m \in \mathcal{U}(P^m) \right\} \quad (\text{F})$$

Die Gipfel erhält man durch die Fuzzy-Menge B ("Berg"):

$$B(P^m) := \begin{cases} |P^m|, & \text{falls } |P^m| = \text{lokMax}(P^m) \\ s + \epsilon, & \text{sonst} \end{cases} \quad (\text{G})$$

Alle Pixel P^m mit $|B(P^m)| \geq s + \epsilon$ liegen entweder auf einem Gipfel ("echtes lokales Maximum") oder auf einem Grat.

Da die *lokMax*-Funktion das Maximum über alle Pixel einer Umgebung von P bildet, ist das notwendige Kriterium für lokale Maxima erfüllt, wenn $|B(P^m)| > s + \epsilon$ gilt. Deshalb sind alle lokalen Maxima in der Menge

$$\left\{ P^m \in \mathcal{P}^m \mid |B(P^m)| > s + \epsilon \right\}$$

enthalten.

Entsprechend können die Grate G durch Kombination der folgenden Funktionen G_r und G'_r bestimmt werden:

$$G_r(\mathcal{R}_i(P^m), P^m) := \begin{cases} |P^m|, & \text{falls } |P^m| > \max_{r \in \mathcal{R}_i(P^m)} \\ s + \epsilon, & \text{sonst} \end{cases} \quad (\text{H1})$$

$$G'_r(\mathcal{R}_i(P^m), P^m) := \begin{cases} |P^m|, & \text{falls } |P^m| \geq \max_{r \in \mathcal{R}_i(P^m)} \\ s + \epsilon, & \text{sonst} \end{cases} \quad (\text{H2})$$

wobei $\mathcal{R}_i(P) \subset \mathcal{U}(P) \setminus g_i(P)$ als Teil-Menge aller Umgebungspunkte $\mathcal{U}(P)$ von P definiert wird, die nicht auf der zu testenden Grat-Menge $g_i(P) \subset \mathcal{U}(P)$ liegen.

Die Funktionen ω_m , die Umgebungsmengen $\mathcal{U}_m(P^m)$ und $\mathcal{U}(P)$, die Grat-Menge $g_i(P)$ sowie $\mathcal{R}_i(P)$ und – für Farb-Bilder – die Vordergrund-Mengen $\mathcal{V}(P)$ sind unter Einhaltung der oben getroffenen Bedingungen frei wählbar, sie bestimmen Form und Verlauf des Höhengebirges.

Das Fuzzy-Skelett $F := G \cup B$ enthält alle Mittellinien-Punkte und $|P|$ ist ein direktes Maß für den Abstand des Punktes P vom nächstgelegenen Rand der Fläche, in der er liegt. Dadurch eignet sich das Fuzzy-Skelett als Ausgangsbasis für eine nachgeschaltete Vektorisierung.

1.3 Parameter-Wahl für die bevorzugte Ausführung

Für die bevorzugte Ausführung werden $\omega_m, \mathcal{U}_m, \mathcal{U}, \mathcal{R}_i$ und die Pixelmenge \mathcal{P} (für alle m identisch) wie folgt gewählt:

$$\begin{aligned} \mathcal{P} &:= \{P_{i,j} \mid \min_x \leq i \leq \max_x \wedge \min_y \leq j \leq \max_y\} & (1) \\ \mathcal{U}_m(P_{x,y}^m) &:= \{P_{i,j} \in \mathcal{P} \mid x-1 \leq i \leq x+1 \wedge y-1 \leq j \leq y+1\} & (2) \\ \mathcal{U} &:= \mathcal{U}_m(P_{x,y}^m) \cup \{P_{i,j} \mid (i,j) \in \{(x-2,y), (x+2,y), (x,y-2), (x,y+2)\}\} & (3) \\ \omega_m(P_{x,y}^m) &\equiv 1 \quad \forall u \in \mathcal{U}_m(P_{x,y}^m) & (4) \\ \mathcal{R}_1(P_{x,y}^m) &:= \{P_{i,j} \mid i \in \{x-1, x+1\}, y = j\} & (5) \\ \mathcal{R}_2(P_{x,y}^m) &:= \{P_{i,j} \mid i \in \{x-2, x+1\}, y = j\} & (6) \\ \mathcal{R}_3(P_{x,y}^m) &:= \{P_{i,j} \mid i \in \{x-1, x+2\}, y = j\} & (7) \\ \mathcal{R}_4(P_{x,y}^m) &:= \{P_{i,j} \mid x = i, j \in \{y-1, y+1\}\} & (8) \\ \mathcal{R}_5(P_{x,y}^m) &:= \{P_{i,j} \mid x = i, j \in \{y-2, y+1\}\} & (9) \\ \mathcal{R}_6(P_{x,y}^m) &:= \{P_{i,j} \mid x = i, j \in \{y-1, y+2\}\} & (10) \\ \mathcal{R}_7(P_{x,y}^m) &:= \{P_{i,j} \mid (i,j) \in \{(x-1, y-1), (x+1, y+1)\}\} & (11) \\ \mathcal{R}_8(P_{x,y}^m) &:= \{P_{i,j} \mid (i,j) \in \{(x-1, y+1), (x+1, y-1)\}\} & (12) \end{aligned}$$

D. h., die Pixelmenge \mathcal{P} besteht aus quadratischen Pixeln und die Umgebungen sind 3 x 3 Pixel große Rechtecke mit dem Bezugspixel im Mittelpunkt plus jeweils das Pixel an der Seitenmitte.

Ein Punkt P liegt in der Gratmenge G , falls gilt:

$$\begin{aligned} G_r(\mathcal{R}_1) \vee G_r(\mathcal{R}_4) \vee G_r(\mathcal{R}_7) \vee G_r(\mathcal{R}_8) \\ \vee (G'_r(\mathcal{R}_1) \wedge (G_r(\mathcal{R}_2) \vee G_r(\mathcal{R}_3))) \\ \vee (G'_r(\mathcal{R}_4) \wedge (G_r(\mathcal{R}_5) \vee G_r(\mathcal{R}_6))) \end{aligned}$$

Eine graphische Darstellung liefern die **Figuren 1** und **2**.

2 Patentansprüche

1. Verfahren zur Bestimmung eines Wertes $|P|$, der äquivalent zum Abstand des Pixels (Bildpunktes) P zum nächstgelegenen Flächenrand ist, für alle Pixel eines zweidimensionalen Bildes durch ein- oder mehrmalige Anwendung der folgenden Schritte:

- (a) **Startschritt:** Jedem Pixel P aus der Menge aller Pixel \mathcal{P} eines Bildes B wird ein Wert $|P^0|$ wie folgt zugeordnet:

$$\begin{aligned} \text{für s/w-Bilder} \quad \mathcal{V} \text{ (Vordergrund)} : \quad & \mathcal{V} \subset \mathcal{P} \\ & \mathcal{H} \text{ (Hintergrund)} : \quad \mathcal{H} := \mathcal{P} \setminus \mathcal{V} \\ \\ \text{für Farb-Bilder} \quad P \in \mathcal{V}(P) : \quad & \mathcal{V}(P) \subset \mathcal{P} \\ P \notin \mathcal{H}(P) : \quad & \mathcal{H}(P) := \mathcal{P} \setminus \mathcal{V}(P) \end{aligned} \tag{A}$$

(Im folgenden wird zur Vereinfachung der Gleichungen nicht mehr zwischen $\mathcal{V}(P)$ und \mathcal{V} bzw. $\mathcal{H}(P)$ und \mathcal{H} unterschieden.)

$$\begin{aligned} |P^0| &:= \max && \text{für } P \in \mathcal{V} \\ |P^0| &:= \min && \text{für } P \in \mathcal{H} \\ &\text{mit } \max > \min \in \mathbf{R} \end{aligned} \tag{B}$$

- (b) **Iterationsschritt:** Der Wert $|P^m|$ wird durch $|P^m| := f^{m-1}(P^{m-1})$ bestimmt, wobei f als die gewichtete Summe über alle Nachbarpunkte den Wert jeweils erhöht oder reduziert:

$$f^m(P^m) := \begin{cases} \max(s + \epsilon, |P^m| + \text{lokAdd}(P^m)) & \text{falls } |P^m| > s \\ \min(s - \epsilon, |P^m| + \text{lokAdd}(P^m)) & \text{falls } |P^m| < s \end{cases} \tag{C}$$

mit $\min < s < \max$, $u \in \mathcal{U}_m(P^m)$, $s \in \mathbf{R}$, $\epsilon \in \mathbf{R}^{>0}$. $\mathcal{U}_m(P^m)$ ist eine beliebige, aber für jedes P^m eindeutig bestimmte Umgebung von P^m und die lokale Additionsfunktion ist definiert durch:

$$\text{lokAdd}(P^m) := \sum_{u \in \mathcal{U}_m(P^m)} \omega_m(u, P^m) \delta(u, P^m) \tag{D}$$

Die Gewichtsfunktion $\omega_m : \mathcal{P} \times \mathcal{P} \rightarrow \mathbf{R}^{>0}$ ist unter der Einschränkung $\omega_m(x, y) > 0 \quad \forall x, y$ frei wählbar. Die Bewertungsfunktion δ muß für s/w und farbige Bilder jeweils unterschiedlich definiert werden, und zwar

für Farbe

$$\delta(u, P^m) := \begin{cases} -1, & \text{falls } (|u| < \max \vee u \in \mathcal{H}(P^m)) \\ & \wedge s + \epsilon < |P^m| \\ 0, & \text{sonst} \end{cases} \tag{E1}$$

für s/w

$$\delta(u, P^m) := \begin{cases} -1, & \text{falls } |u| < \max \wedge s + \epsilon < |P^m| \\ 1, & \text{falls } |u| > \min \wedge |P^m| < s - \epsilon \\ 0, & \text{sonst} \end{cases} \tag{E2}$$

2. Verfahren nach 1. mit den folgenden Parametern (siehe auch Kap. 1.3 für Binärbilder):

$$\mathcal{P} := \{P_{i,j} \mid \min_x \leq i \leq \max_x \wedge \min_y \leq j \leq \max_y\} \quad (1)$$

$$\mathcal{U}_m(P_{x,y}^m) := \{P_{i,j} \in \mathcal{P} \mid x-1 \leq i \leq x+1 \wedge y-1 \leq j \leq y+1\} \quad (2)$$

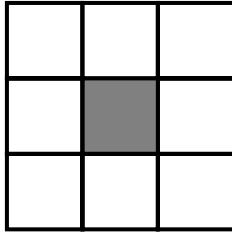
$$\omega_m(P_{x,y}^m) \equiv 1 \quad \forall u \in \mathcal{U}_m(P_{x,y}^m) \quad (4)$$

3. Verfahren nach 1. oder 2. realisiert durch Soft- oder Hardware oder eine Mischung aus beidem.

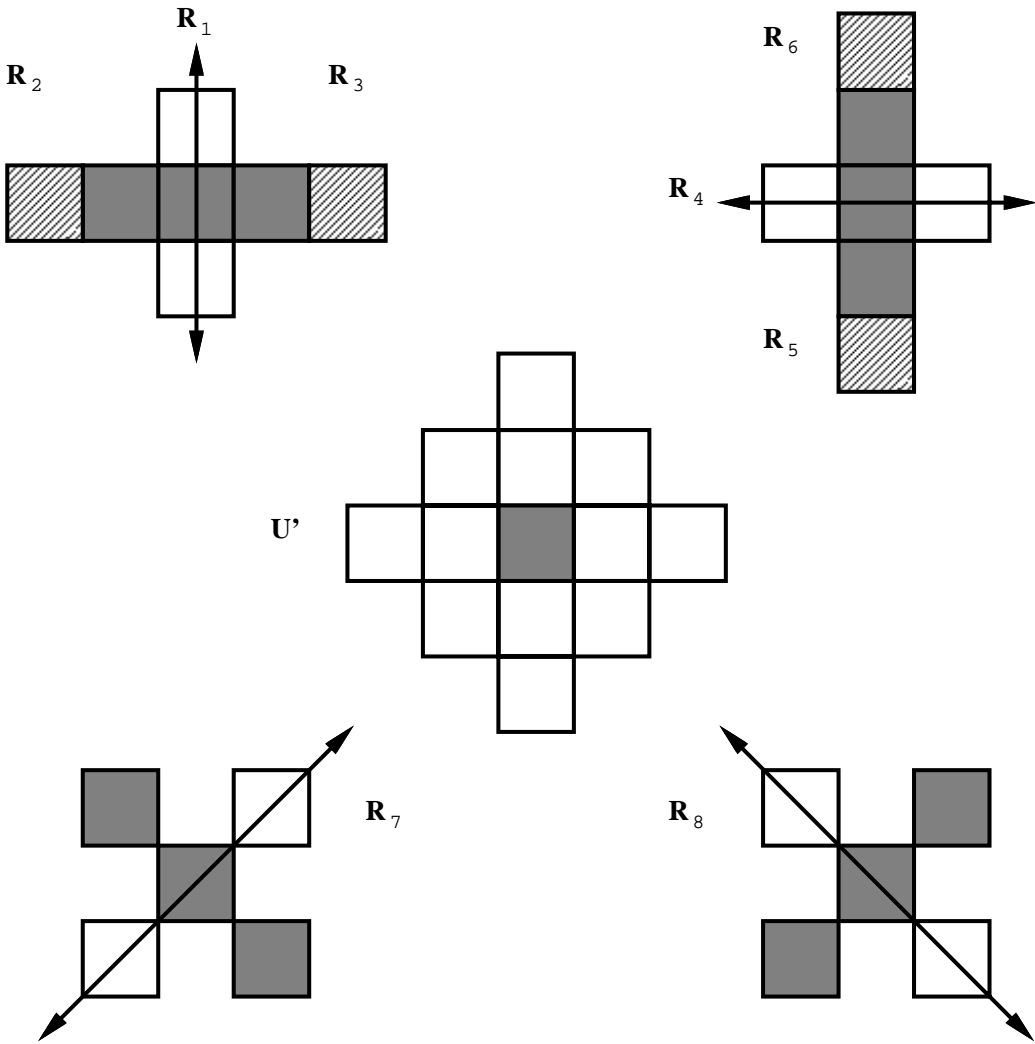
Die abschließende Bestimmung des ‘‘Fuzzy-Skeletts‘‘ F durch Ermittlung der Gipfel B und Grate G gemäß (F), (G), (H1) und (H2) wird nicht patentrechtlich beansprucht, da das Verfahren dem Stand der Technik entspricht.

Literatur

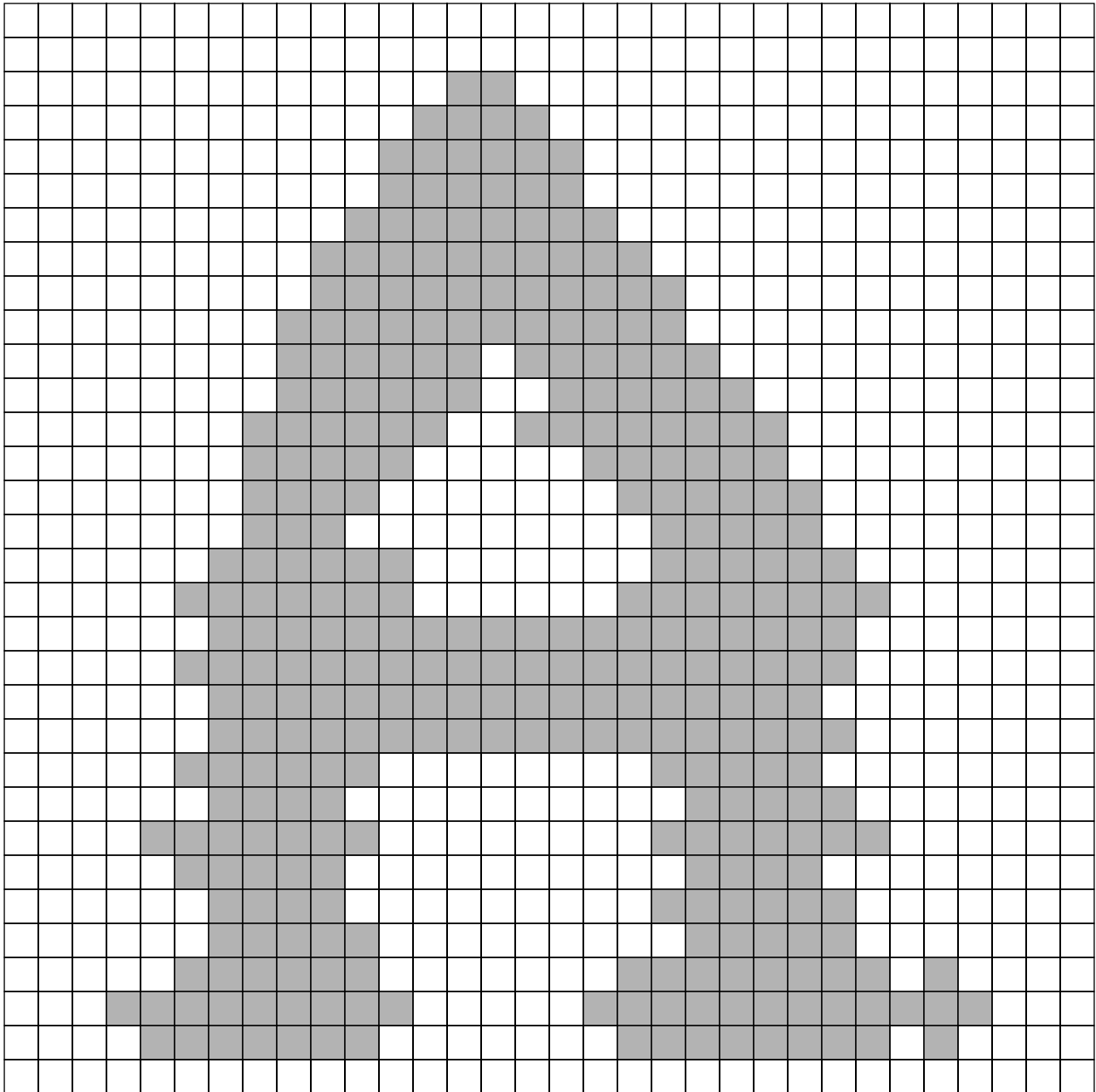
- [chin] Roland T. Chin, Ben K. Jang.
A Reconstructable One-Pass Parallel Thinning
SPIE Vol. 1657 Image Processing Algorithms and Techniques III 1992, S.437 - 448
- [ogen] Robert Ogeniewicz, Olaf Kübler, Ferdinand Klein, Ulrich Kienholz.
Lage- und skalierungsinvariante Skelette zur robusten Beschreibung und Erkennung binärer Formen
Informatik Fachberichte 219, S. 44 - 51
Mustererkennung 1989, 11. DAGM Symposium 1989, Springer Verlag, Hamburg

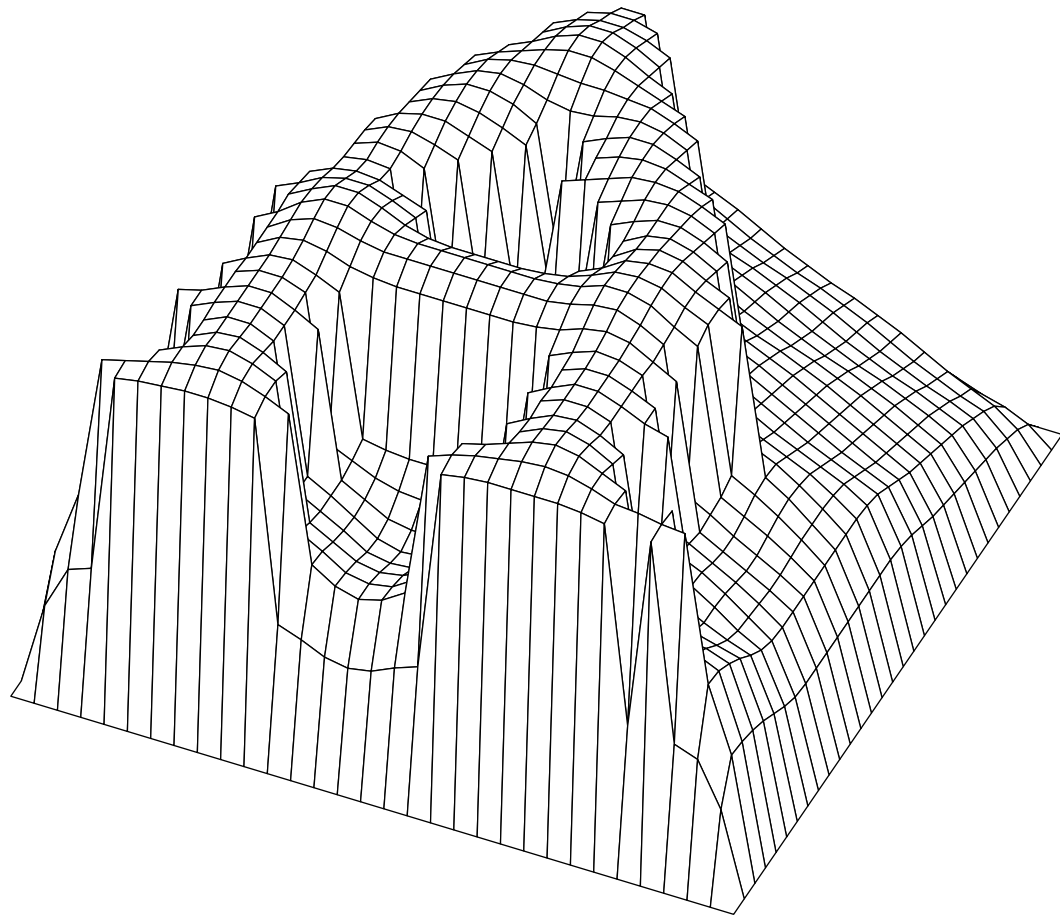


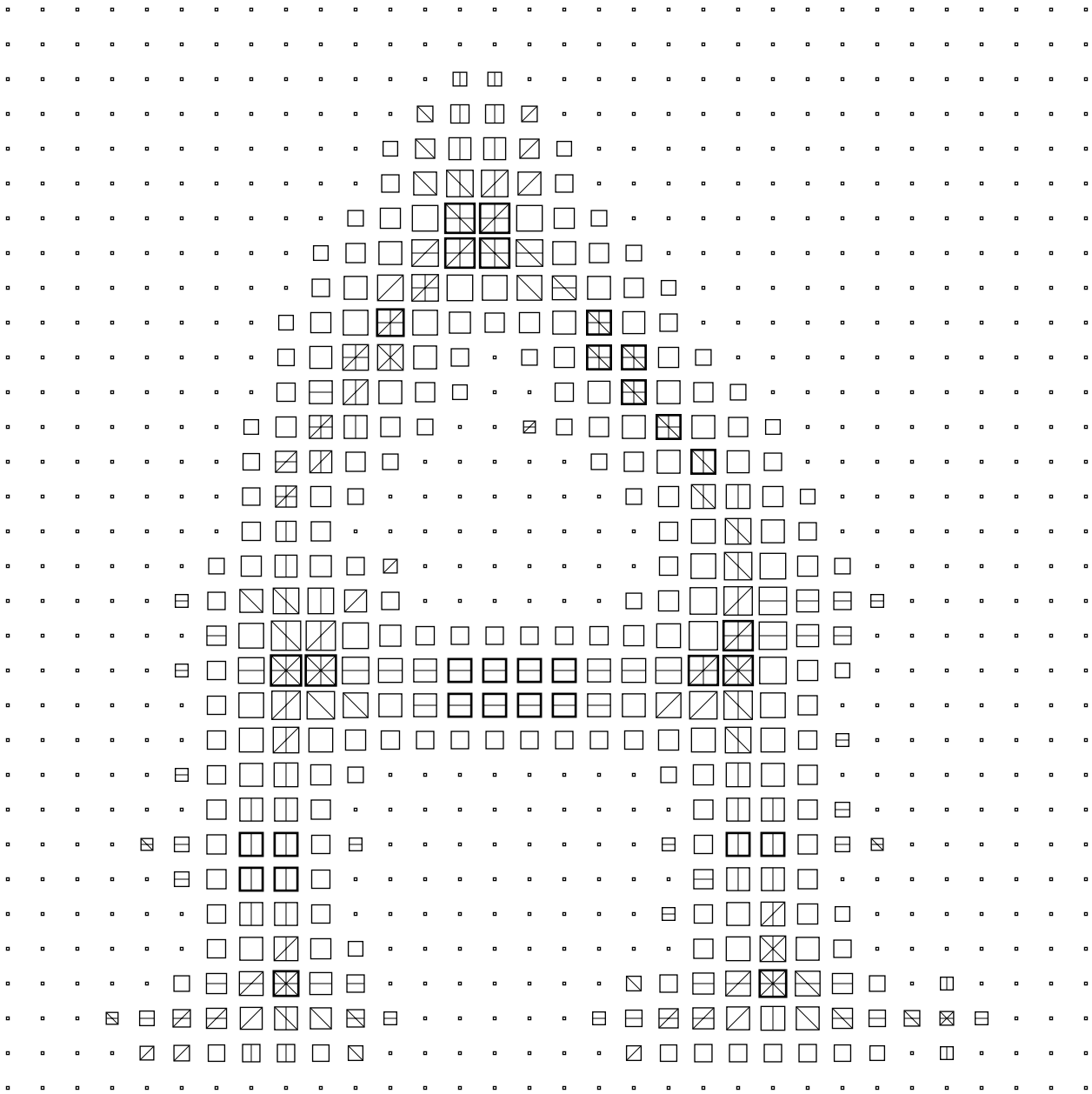
Figur 1: Umgebung U

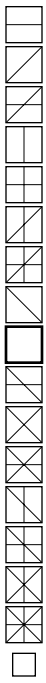


Figur 2: Reduzierte Umgebungen R_i und deren Grundmenge U'









x Grat
xy Grat
x,xy Grat
y Grat
x,y Grat
x,xy Grat
x,xy,y Grat
-xy Grat
lokales Maximum (fett)
x,-xy Grat
xy,-xy Grat
x,xy,-xy Grat
y,-xy Grat
x,y,-xy Grat
x,xy,-xy Grat
x,xy,y,-xy Grat
normales "Ein"-Pixel
normales "Aus"-Pixel