Bildverarbeitung
Thema 2: Filterung

D. Schlesinger – TUD/INF/KI/IS

Allgemeines

Klassische Anwendung: Entrauschung

(Fast) jeder Filter basiert auf einem Modell (Annahme): Signal + Rauschen

Pipeline: Modell → Aufgabe → Lösung → Algorithmus (Programm)

- Mittelwertfilter: Modell → ... → Lösung
- Medianfilter: Aufgabe → Lösung
- Vergleich: Mitterwertfilter vs. Medianfilter
- Morphologische Filter

- Lineare Filterung: Algorithmen

Mittelwertfilter

Bezeichnungen:
$D \subset \mathbb{Z}^2$ – Definitionsbereich (Gitter), r – Pixel $r = (i, j)$, $r \in D$
Bild ist eine Funktion $x : D \to C$ (Farbraum), x_r – Farbe des Pixels r

Sei y das „Ideale“ Signal (nicht verrauscht) und x das verrauschte Signal (Beobachtung)
Aufgabe: Man sieht x, man berechne y.
Rauschmodell – Wahrscheinlichkeitsverteilung für Abweichungen (Pixel-unabhängig):

$$p(x|y) \sim \prod_r \exp\left[-\frac{\|x_r - y_r\|^2}{2\sigma^2}\right]$$

Aufgabe nach dem Maximum Likelihood Prinzip: $p(y|x) \to \max_y$
Lösung – trivial: $y = x$ (weil unabhängig)

Eine Annahme über das Signal y ist notwendig!!!

Mittelwertfilter

Annahme: In einer kleinen Umgebung $W(r)$ eines Pixels r ist y (fast) konstant.

$$\text{ML:} \qquad \prod_{r' \in W(r)} \exp\left[-\frac{\|x_{r'} - y_r\|^2}{2\sigma^2}\right] \to \max_{y_r}$$

$$\ln \ \ldots : \qquad F(y_r) = \sum_{r'} \|x_{r'} - y_r\|^2 \to \min_{y_r}$$

$$\text{Ableiten:} \qquad \frac{\partial F}{\partial y_r} = \sum_{r'} (x_{r'} - y_r) = \sum_{r'} x_{r'} - |W| \cdot y_r = 0$$

$$\Rightarrow \qquad y_r = \frac{1}{|W|} \sum_{r'} x_{r'}$$

Median Filter

Andere Zielfunktion (Einfachheit halber $C = \mathbb{R}$ – Grauwertbild):

$$F(y_r) = \sum_{r'} |x_{r'} - y_r| \to \min_{y_r}$$

Problem: nicht differenzierbar. Gute Neuigkeit – konvex.

(Ableitung an der Tafel).

Lösung – Medianfilter.

Vergleich: Gaussches Rauschen

(a) Original

(b) Verrauscht

(c) Gaussche Glättung

(d) Medianfilter

Vergleich: „Salz und Pfeffer“ Rauschen

(e) Original

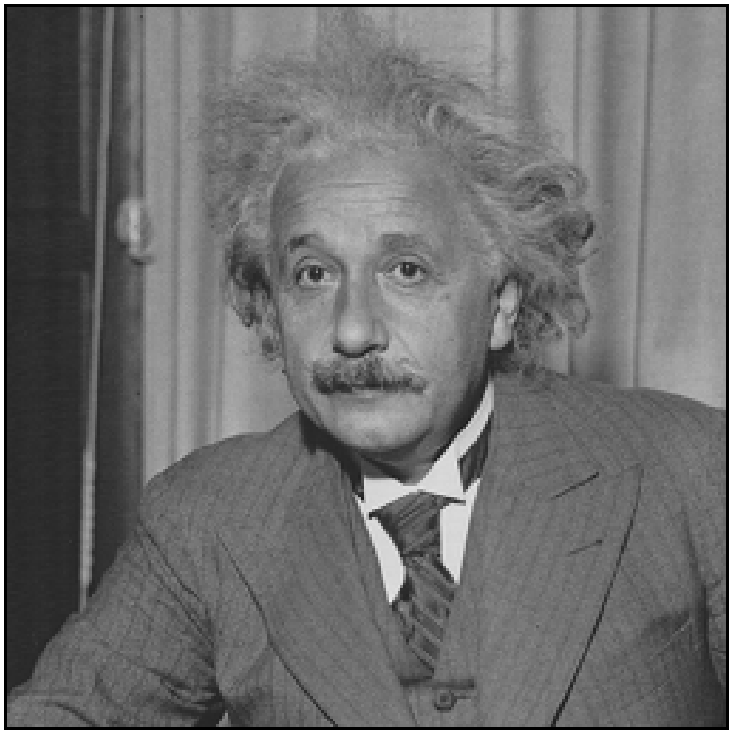
(f) Verrauscht

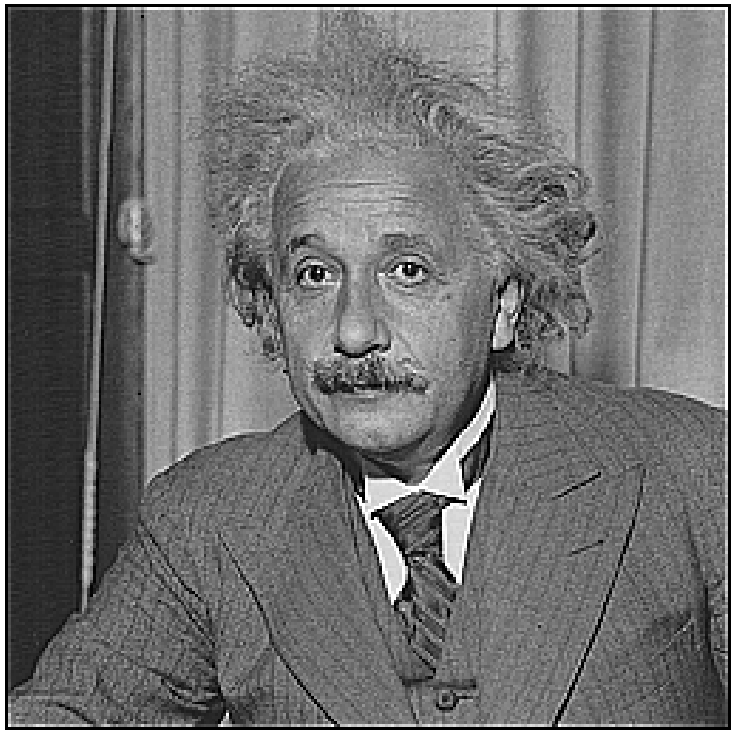
(g) Gaussche Glättung

(h) Medianfilter

Andere Vorgehensweisen

Beispiel – Sharpening:
Unterstütze Signalanteile bestimmter Frequenzen (z.B. Kanten – hohe Frequenz)





Andere: Wiener Filter, Gaussches Filter, Blur (Binominal Filter) etc.

Mehr allgemein: Fourier Analyse

Morphologische Filter

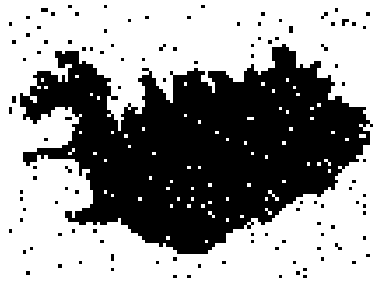
Ursprünglich – für binäre Bilder $x : D \to \{0, 1\}$
Morphologische Operation: y_r ist eine Boolsche Funktion von $x_{W(r)}$

Beispiele:

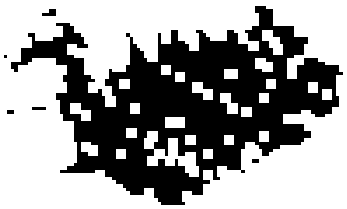
UND (Deliation) – $y_r = \bigwedge_{r' \in W(r)} x_{r'}$

ODER (Erosion) – $y_r = \bigvee_{r' \in W(r)} x_{r'}$

(i) Original



(j) Verrauscht



(k) Deliation

(l) Erosion

(m) Erosion

(n) Deliation

Morphologische Filter

Erweiterung auf Grauwertbilder $x : D \to \mathbb{R}$

1) Erzeuge Binärbilder durch Schwellwertoperator

$$x_r^\Theta = \mathbb{1}(x_r > \Theta)$$

   für alle Θ

2) Wende die gewählte morphologische Operation auf alle Binärbilder x^Θ an:

$$y_r^\Theta = f(x_W^\Theta)$$

3) Wähle y_r als kleinste Θ mit $y_r^\Theta = 1$

Beispiele:
f ist UND – $y_r = \min\limits_{r' \in W(r)} x_{r'}$
f ist ODER – $y_r = \max\limits_{r' \in W(r)} x_{r'}$

RGB – Getrennt für R, G und B

(o) Original

(p) Verrauscht

(q) min

(r) max

(s) max

(t) min

Fazit

Explizite Annahmen über das Signal (lokale Eigenschaften) – lokale Filterung

Signal wird durch seine Autokorrelationen beschrieben – Wiener Filter

Signal ist die Überlagerung der Signale bestimmter Frequenzen – Fourier Analyse

Signal wird durch eine Differentialgleichung beschrieben – Variationelle Ansätze

Explizite Modellierung statistischer Abhängigkeiten zwischen den Pixelwerten – MRF

u.s.w.

Die Eigenschaften des Signals (das Modell, die Annahmen, die Art der zu lösenden Aufgabe etc.) sind ausschlaggebend für die Wahl des Filters.

Eine „andere“ Aufgabe

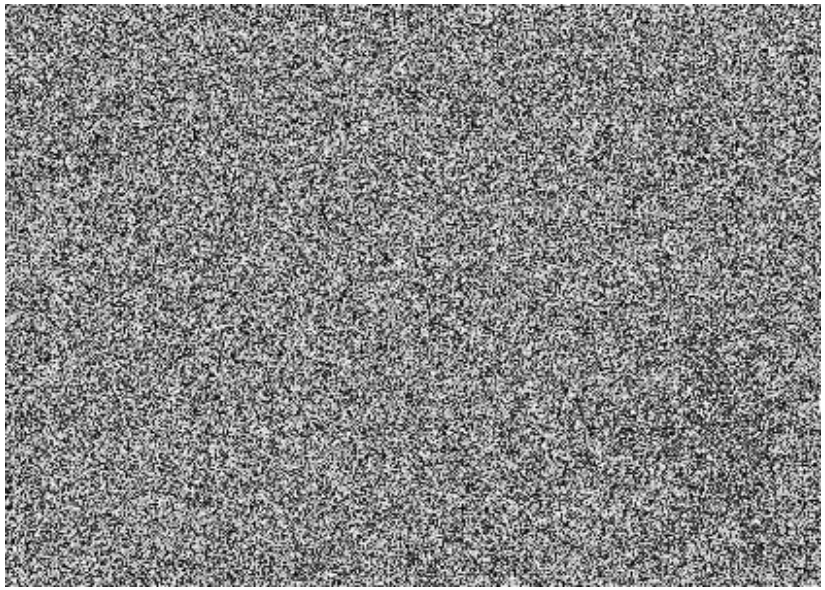
Glättung des Hintergrunds verbessert die räumliche Wahrnehmung der Szene

(Xue Bai, Guillermo Sapiro)

Umgekehrte Aufgabe: Rekonstruiere 3D aus Unschärfe.
Konsequenz: Die Filter liefern keine “Antworten“, sie geben ein lokales Maß für die höheren Stufen der Verarbeitung.

Eine noch andere Aufgabe

„Salz und Pfeffer“ Rauschen mit 90%



Das Modell entspricht dem Bild „exakt“. Dies macht genaue Rekonstruktion möglich.

Lineare Filterung – Algorithmen

Eindimensional: $r \in \mathbb{N}$

Beispiel: Mittelwertfilter $y_r = \frac{1}{|W|} \sum_{r'=r-w}^{r+w} x_{r'} \qquad (|W| = 2w+1)$

Naiver Algorithmus (direkt nach der Formel):

for $r = 0$ `bis` n
  $sum = 0$
  for $r' = r - w$ `bis` $r + w$
    $sum = sum + x_{r'}$
  $y_r = sum/|W|$

Zeitkomplexität: $n \cdot |W|$

Idee:

$$\sum_{r'=r-w}^{r+w} x_{r'} = \sum_{r'=0}^{r+w} x_{r'} - \sum_{x'=0}^{r-w-1} x_{r'} = \tilde{x}_{r+w} - \tilde{x}_{r-w-1}$$

Besserer Algorithmus:

Berechne $\tilde{x}_r$**:**

for $r = 0$ **bis** n

$\tilde{x}_r = \tilde{x}_{r-1} + x_r$

Berechne y_r**:**

for $r = 0$ **bis** n

$y_r = (\tilde{x}_{r+w} - \tilde{x}_{r-w-1})/|W|$

Zeitkomplexität: n

Lineare Filterung – Algorithmen

Verallgemeinerung auf 2D:
Verwendung des „Integralbildes“ $\tilde{x}$

Berechne $\tilde{x}_r$**:**

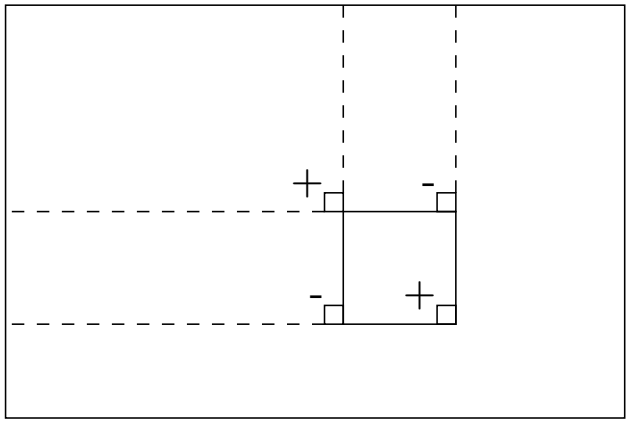
for $(i,j) = (0,0)$ **bis** (m,n)
(zeilenweise)

$$\tilde{x}_r = \tilde{x}_{i,j-1} + \tilde{x}_{i-1,j} - \tilde{x}_{i-1,j-1} + x_r$$

Berechne y_r**:**

for $(i,j) = (0,0)$ **bis** (m,n)

$$y_r = (\tilde{x}_{i+w,j+w} - \tilde{x}_{i-w,j+w} - \tilde{x}_{i+w,j-w} + \tilde{x}_{i-w,j-w})/|W|$$



Ein etwas komplizierteres Beispiel (wieder 1D):

$$y_r = \sum_{r'=r-w}^{r} (w - r + r') \cdot x_{r'}$$

Man überlege, ob sich y_{r+1} aus dem y_r effizient berechnen lässt:

$$\begin{aligned}
y_{r+1} &= \sum_{r'=r+1-w}^{r+1} (w - r - 1 + r') \cdot x_{r'} \\
&= \sum_{r'=r+1-w}^{r+1} (w - r + r') \cdot x_{r'} - \sum_{r'=r+1-w}^{r+1} x_{r'} \\
&= \sum_{r'=r-w}^{r} (w - r + r') \cdot x_{r'} + w \cdot x_{r+1} - \sum_{r'=r-w}^{r} x_{r'} \\
&= y_r + w \cdot x_{r+1} - \bar{x}_r
\end{aligned}$$

$\bar{x}_r$ kann mit linearer Zeitkomplexität berechnet werden (Mittelwertfilter)
$\Rightarrow$ die gesamte Zeitkomplexität ist linear.

Lineare Filterung – Algorithmen

Faltungen allgemein:

$$y = x * g$$

$$y_i = \sum_{j=-\infty}^{\infty} x_{i-j} \cdot g_j$$

Eigenschaften:

- sind kommutativ, d.h. $x * g = g * x$;
- sind assoziativ, d.h. $(x * g^1) * g^2 = x * (g^1 * g^2)$;

Identische Faltung (ändert das Signal nicht): $g^I_j = \mathbb{1}(j = 0)$

Inverse Faltungen: $g * g^{-1} = g^I$
Beispiel ($j = 0$ ist fett gekennzeichnet):

$$\begin{array}{rcll} g^{diff} & = & [\ldots, 0, 0, 0, \mathbf{1}, -1, 0, \ldots] & \text{Differential Operator} \\ g^{int} & = & [\ldots, 0, 0, 0, \mathbf{1}, \quad 1, 1, \ldots] & \text{Integral Operator} \end{array}$$

Lineare Filterung – Algorithmen

Der Trick zur effizienten Berechnung basiert auf der folgenden Umwandlung:

$$x * g = x * g^I * g = x * g^{int} * g^{diff} * g = (x * g^{int}) * (g * g^{diff})$$

oder sogar mehr

$$x * g = (x * g^{int} * \ldots * g^{int}) * (g * g^{diff} * \ldots * g^{diff})$$

Die Faltung $x * g^{int}$ braucht lineare Zeitkomplexität,
mit $\tilde{g} = g * g^{diff} * \ldots * g^{diff}$ (Vorberechnung) wird erreicht, dass $\tilde{g}$ schwach besetzt ist.

Andere Filterungen – Algorithmen

Man hat Median für $x_{W(r)}$, man berechne Median für $x_{W(r+1)}$

Die Idee: die Menge $x_{W(r)}$ wird geordnet gehalten und aktualisiert.
Zeitkomplexität – $n \ln |W|$

Zur selbständigen Überlegung – Faltung mit dem Exponent

$$y_r = \sum_{r'=-\infty}^{r} x_{r'} \cdot \exp[-\tau(r - r')]$$