

Abteilung Stochastik  
- Universität Ulm –

Seminar: Bayessche Ansätze in der Bildanalyse

# Stochastische Bildmodelle und deren Anwendung

Said Bakadir

# Inhalt

- Einleitung
- Beispiele von Bildmodellen
- Simulationen für das Ising-Modell
- Schätzung der Bilder
- Schätzung der Parameter
- Graustufenbilder

# Einleitung

- Wir betrachten Bilder als zufällige Markov Felder.
- Die Äquivalenz von zufälligen Markov Feldern und Gibbs Markov Feldern liefert:

$$\pi(\mathbf{x}) = \frac{1}{Z} \exp\left(-\sum_{c \in N} V_c(\mathbf{x})\right)$$

- Wobei:

$\mathbf{X} = (X_i)_{i \in S}$  ein zufälliges Markov Feld, oder ein Bild

S: Indexmenge (z.B alle Pixel eines Bildes),

L: Teilmenge in  $\mathbb{R}$  (Phasenraum),  $x_i \in L$

N: Nachbarschaftssystem auf S.

C: Element in N (alle Nachbarn einer Stelle).

$V_c$ : Potentialfunktion

$$Z = \sum_{\mathbf{x}} \exp\left(-\sum_{c \in N} V_c(\mathbf{x})\right)$$

# Beispiele

- **Ising-Modell :**

für Bilder mit nur zwei Farben

$$L = \{0, 1\}$$

$$\pi(x) = \frac{1}{Z(\beta)} \exp\left(\beta \sum_{i \sim j} I_{[x_i = x_j]}\right)$$

$\beta$  ein positiver Parameter.

- **Potts -Modell:**

für Bilder mit mehreren Farben

$$L = \{0, 1, \dots, M\}$$

Die A-Priori-Verteilung von  $x$  hat die gleiche Form wie beim Ising-Modell

- **Auto-binomiales Modell**

dient dazu, die Textur zu beschreiben

$$L = \{0, 1, \dots, M\}$$

$$V_C(\mathbf{x}) = \begin{cases} -\ln \binom{M}{x_i} - \alpha_i x_i & \text{falls } C = \{i\} \\ -\beta_{i,j} x_i x_j & \text{falls } C = \{i, j\} \end{cases}$$

die bedingte Verteilung von  $x_i$  ist binomial mit Parametern  $M$  und  $q$ , d.h

$$\pi(x_i | \mathbf{x}_{N_i}) = \binom{M}{x_i} q^{x_i} (1-q)^{M-x_i}$$

wobei:

$$q = \frac{\exp(\alpha_i + \sum_{j \in N_i} \beta_{i,j} x_j)}{1 + \exp(\alpha_i + \sum_{j \in N_i} \beta_{i,j} x_j)}$$

# Verrauschte Bilder

- Im Allgemeinen wird das Bild  $\mathbf{x}$  nicht genau aufgenommen
- Es werden Daten  $\mathbf{y}$  anstatt  $\mathbf{x}$  beobachtet.
- Bayes-Ansatz hilft bei der Konstruktion der A-Posteriori-Verteilung  $\pi(\mathbf{x}|\mathbf{y})$ .
- $\mathbf{y}=\varphi(\mathbf{x})$  , wobei  $\varphi$  eine Funktion ist. Sie liefert Informationen über das Rauschen
- **Annahme:**

$$\pi(\mathbf{y} | \mathbf{x}) = \pi(y_1 | x_1)\pi(y_2 | x_2)\dots\dots\dots\pi(y_n | x_n)$$

wobei  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  und  $\mathbf{y}=(y_1, y_2, \dots, y_n)$

# Beispiele

- Gaußsches additives Rauschen

$$\mathbf{y} = \mathbf{x} + \boldsymbol{\varepsilon}$$

wobei  $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 \mathbf{I})$ ,  $\mathbf{x}$  und  $\boldsymbol{\varepsilon}$  sind unabhängig

$$\pi(y_i | x_i) \propto \exp\left(-\frac{1}{2\sigma^2}(y_i - x_i)^2\right)$$

daraus folgt

$$\pi(\mathbf{y} | \mathbf{x}) \propto \exp\left(-\sum_{i=1}^n \frac{1}{2\sigma^2}(y_i - x_i)^2\right)$$

Nach dem Satz von Bayes folgt,

$$\pi(\mathbf{x} | \mathbf{y}) = \frac{\pi(\mathbf{y}, \mathbf{x})}{\pi(\mathbf{y})} = \frac{\pi(\mathbf{y} | \mathbf{x}) \cdot \pi(\mathbf{x})}{\pi(\mathbf{y})}$$

wir betrachten binäre Bilder. Die A-Posteriori-Verteilung ist

$$\pi(\mathbf{x} | \mathbf{y}) \propto \exp\left(\beta \sum_{i=j} I_{[x_i=x_j]} + \sum_{i=1}^n -\frac{1}{2\sigma^2}(y_i - x_i)^2\right)$$

- **Binäres Rauschen (Flip noise)**

Tritt ein, wenn die falsche Farbe an der Stelle  $i$  mit einer Bernoulli Wahrscheinlichkeit auftritt.

$$\pi(y_i | x_i) = \begin{cases} 1-p & \text{falls } y_i = x_i \\ p & \text{falls } y_i \neq x_i \end{cases}$$

$$\text{sei } k := \sum_{i=1}^n I_{[x_i=y_i]}$$

$$\begin{aligned} \pi(y | x) &= (1-p)^k p^{n-k} \\ &= \exp(k \ln(\frac{1-p}{p}) + n \ln(p)) \end{aligned}$$

$$\text{d.h. } \pi(y | x) \propto \exp\left(\sum_{i=1}^n I_{[x_i=y_i]} \ln\left(\frac{1-p}{p}\right)\right)$$

$$\text{wir haben } \pi(x) \propto \exp\left(\beta \sum_{i \sim j} I_{[x_i=x_j]}\right)$$

Nach dem Satz von Bayes folgt

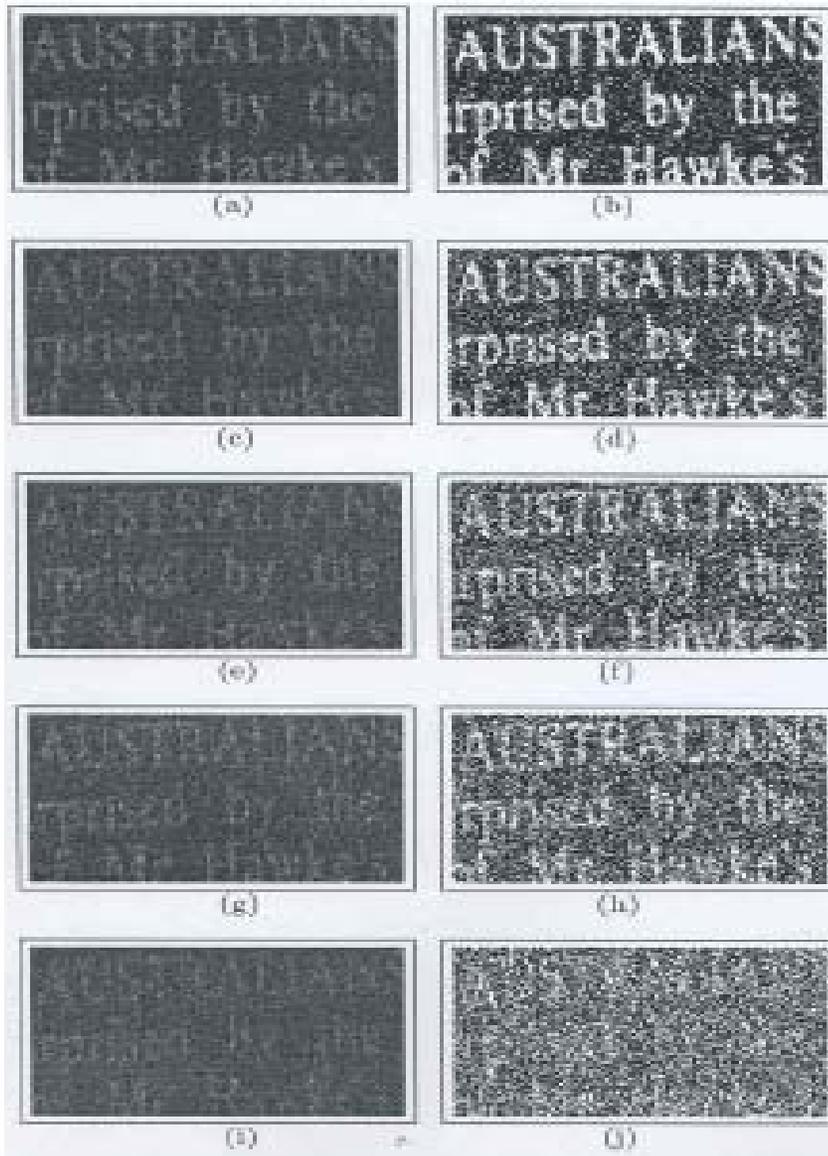
$$\pi(\mathbf{x} | \mathbf{y}) = \frac{\pi(\mathbf{y} | \mathbf{x})\pi(\mathbf{x})}{\pi(\mathbf{y})}$$

$$\pi(\mathbf{x} | \mathbf{y}) \propto \exp\left(\beta \sum_{i \sim j} I_{[x_i = x_j]} + \sum_{i=1}^n h_i(x_i, y_i)\right) \quad (*)$$

$$\text{wobei : } h_i(x_i, y_i) = I_{[x_i = y_i]} \ln\left(\frac{1-p}{p}\right)$$

- Es ist zu beachten, dass die A-Posteriori-Verteilung in dem Gaußschen Rauschen-Modell die Form (\*) hat, wobei

$$h_i(x_i, y_i) = -\frac{1}{2\sigma^2} (y_i - x_i)^2$$



das Ausgangsbild x

Die beobachtete Bilder y

**Gaußsches Rauschen**

- (a)  $\sigma^2 = 0.1$
- (c)  $\sigma^2 = 0.4$
- (e)  $\sigma^2 = 0.5$
- (g)  $\sigma^2 = 0.6$
- (i)  $\sigma^2 = 0.8$

**binäres Rauschen**

- (b)  $p = 0.1$
- (d)  $p = 0.2$
- (f)  $p = 0.25$
- (h)  $p = 0.3$
- (j)  $p = 0.4$

# Simulationen für das Ising-Modell

$$\pi(x_i | x_{-i}) \propto \exp(\beta \sum_{i \sim j} I_{[x_i = x_j]})$$

sei  $n_i^b = \sum_{i \sim j} I_{[x_j = 1]}$  die Anzahl der schwarzen Nachbarn von  $i$ .

$n_i^w = \sum_{i \sim j} I_{[x_j = 0]}$  die Anzahl der weißen Nachbarn von  $i$ .

$$\pi(x_i = 1 | x_{-i}) = \frac{\exp(\beta n_i^b)}{\exp(\beta n_i^b) + \exp(\beta n_i^w)}$$

- **Algorithmus 1: Gibbs-sampler**

Wähle  $x = 0$  oder  $1$ ,

for  $t=1$  to  $N$  do

  for  $j=1$  to  $n$  do

$i = \text{Next}(j)$

$U \sim \text{Uniform}(0,1)$

$$p = \frac{\exp(\beta n_i^b)}{\exp(\beta n_i^b) + \exp(\beta n_i^w)}$$

    if  $U < p$  then  $x_i = 1$

      else  $x_i = 0$

    end if

  end for

End for

$\text{Next}(j)$  gibt an, welches Pixel aktualisiert wird, z.B

- $\text{Next}(j) = j$
- $\text{Next}(j)$  wird zufällig gewählt

- Algorithmus 2: Metropolis-sampler

Wähle  $x=0$  oder  $1$ ,

for  $t=1$  to  $N$  do

  for  $j=1$  to  $n$  do

$i=\text{Next}(j)$

$x'_i = 1 - x_i$

$d = \exp(\beta \sum_{i \sim j} I_{[x_i=x_j]})$

$d' = \exp(\beta \sum_{i \sim j} I_{[x'_i=x_j]})$

$p = \min(1, \frac{d'}{d})$

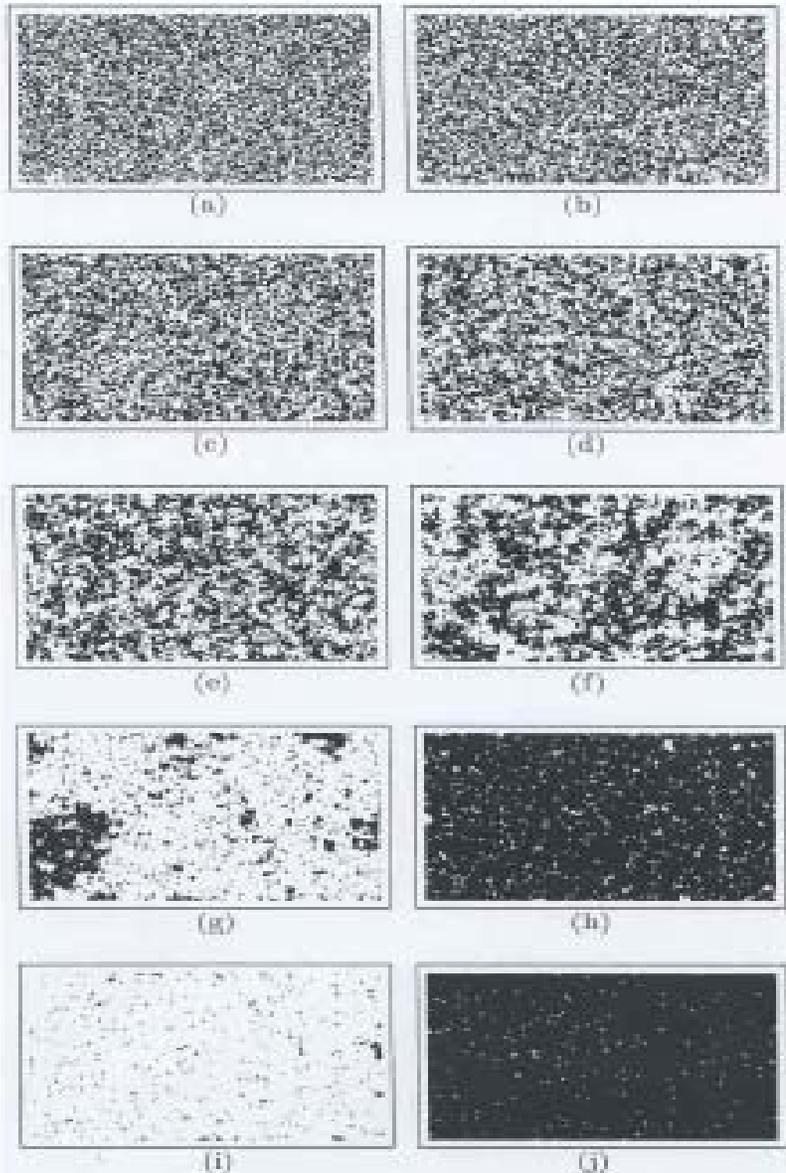
$U \sim \text{Uniform}(0,1)$

    if  $U < p$  then  $x_i = x'_i$

    end if

  end for

end for



### Simulation für das Ising-Modell mit verschiedenen Betas

(a)  $\beta= 0.3$  (b)  $\beta=0.4$  (c)  $\beta= 0.5$  (d)  $\beta=0.6$  (e)  $\beta= 0.7$   
 (f)  $\beta=0.8$  (g)  $\beta= 0.9$  (h)  $\beta=1.0$  (i)  $\beta= 1.1$  (j)  $\beta=1.3$

- Algorithmus 3: Metropolis sampler für das verrauschte Ising-Modell

Initialisiere  $x$ ,

Gebe  $y$  und die Rauschparameter ein

for  $t=1$  to  $N$  do

  for  $j=1$  to  $n$  do

$i = \text{Next}(j)$

$x'_i = 1 - x_i$

$$d = \beta \sum_{i \sim j} I_{[x_i = x_j]} + h_i(x_i, y_i)$$

$$d' = \beta \sum_{i \sim j} I_{[x'_i = x_j]} + h_i(x'_i, y_i)$$

$$p = \exp(\min(0, d' - d))$$

$U \sim \text{Uniform}(0,1)$

    if  $U < p$  then  $x_i = x'_i$

    end if

  end for

end for

- Algorithmus 3 ist numerisch stabiler, da er log-Dichten benutzt.

# Schätzung der Bilder

- $\mathbf{x}$  das zugrundeliegende Bild
- $\mathbf{y}$  die beobachtete Daten
- $\mathbf{z}$  ein Schätzer von  $\mathbf{x}$
- Um den optimalen Schätzer  $\mathbf{z}$  von  $\mathbf{x}$  zu finden, konstruieren wir eine so genannte **Verlustfunktion**, die die Differenz von  $\mathbf{z}$  zu  $\mathbf{x}$  angibt.

## 1. Eine Metrik $L(\mathbf{x}, \mathbf{z})$

ein Schätzer  $\mathbf{z}''$  ist besser als  $\mathbf{z}'$  falls  $L(\mathbf{x}, \mathbf{z}'') < L(\mathbf{x}, \mathbf{z}')$

falls  $\mathbf{x}$  bekannt ist, wird nach der Minimalstelle von  $L$  gesucht.

falls  $\mathbf{x}$  unbekannt ist, wird der A-Posteriori erwartete Abstand zwischen dem Schätzer  $\mathbf{z}$  und  $\mathbf{x}$  betrachtet, d.h:

$$E_{\mathbf{x}|\mathbf{y}} L(\mathbf{x}, \mathbf{z}) = \sum_{\mathbf{x}} \pi(\mathbf{x} | \mathbf{y}) L(\mathbf{x}, \mathbf{z}) \quad (*)$$

der optimale Bayes Schätzer ist die Minimalstelle von (\*)

- Wir betrachten den Vektor  $\mathbf{e}$ , wobei

$$e_i = \mathbb{I}_{[x_i \neq z_i]}$$

## 2. MPM- Schätzer (marginal posterior mode)

- Die entsprechende Verlustfunktion ist

$$L_{\text{MPM}}(\mathbf{e}) = \sum_{i=1}^n e_i$$

- Sie zählt die Pixel der falschen Farbe
- Der MPM Schätzer entspricht dem Bild mit minimaler Anzahl an Pixel falscher Farbe.
- Zur Berechnung der MPM Schätzer berechnen wir zuerst den Erwartungswert des Verlusts.

$$\begin{aligned}
E_{x|y}(L_{\text{MPM}}(\mathbf{e})) &= \sum_{i=1}^n E_{x_i|y} e_i = \sum_{i=1}^n P(x_i \neq z_i | y) \\
&= \text{const} - \sum_{i=1}^n P(x_i = z_i | y)
\end{aligned}$$

der MPM-Schätzer

$$\mathbf{x}_{\text{MPM}} = \arg \min_{\mathbf{z}} \left\{ - \sum_{i=1}^n P(x_i = z_i | y) \right\}$$

$$\mathbf{x}_{\text{MPM}} = \sum_{i=1}^n \arg \max_{z_i} (P(x_i = z_i | y))$$

Zur Berechnung von  $\mathbf{x}_{\text{MPM}}$  ziehen wir N Stichproben aus der A-Posteriori-Verteilung

$$\mathbf{x}_{\text{MPM},i} = \begin{cases} 1 & \text{falls } \#\{x_i=1\} \geq N/2 \\ 0 & \text{falls } \#\{x_i=1\} \leq N/2 \end{cases}$$

### 3. MAP-Schätzer (Maximum A Posteriori)

- Die Verlustfunktion

$$L_{\text{MAP}}(\mathbf{e}) = 1 - \prod_{i=1}^n (1 - e_i)$$

- Unabhängig von der Anzahl der falschen Klassifikationen nimmt diese Funktion den Wert 1 an, wenn mindestens eine falsche Farbe notiert wird, ansonsten den Wert 0.
- Der entsprechende MAP-Schätzer:

$$\mathbf{x}_{\text{MAP}} = \arg \max_z (\pi(\mathbf{z} | \mathbf{y}))$$

- Der MAP-Schätzer wird berechnet durch  
das Ziehen von Stichproben aus der A-Posteriori-Verteilung, oder  
es wird der „**simulated annealing**“ Algorithmus verwendet

# Simulated annealing

- Stochastischer Algorithmus für kombinatorische Optimierung.
- Wird in der Physik verwendet. Durch langsames Abkühlen einer Substanz wird nach der niedrigsten Energiekonfiguration gesucht.
- Es basiert auf den stochastischen Methoden, z.B Metropolis Algorithmus
- Diese Methoden erzeugen jedes Mal eine Konfiguration des MRF.
- Es wird eine Funktion (Temperaturfunktion) eingeführt, die diese Erzeugungen kontrolliert.

Ziel:  $x^* := \arg \max_x \pi(x | y)$

- wir betrachten  $\pi_T(x) \propto (\pi(x | y))^{1/T}$

Also  $x^* = \arg \max_x \pi_T(x)$

- wenn  $T$  gegen  $0$  konvergiert, tritt  $x^*$  mit der größten Wahrscheinlichkeit ein.
- für  $T=0^+$  wird durch einen MCMC-Algorithmus eine lokale Lösung gefunden.
- Um die globale Lösung zu finden, wird  $T=T(t)$  betrachtet ( $t>0$ ),  
und man lässt  $t$  gegen  $\infty$  konvergieren.

Die Funktion  $T$  hat folgende Eigenschaften:

$T$  ist monoton fallend

$T \rightarrow 0$ , wenn  $t \rightarrow \infty$ .

hier wählen wir  $T(t)=T_0 \eta^{t-1}$

z.B.  $T_0 = 4$  und  $\eta=0.999$

## Algorithmus 4: „simulated annealing“ für das verrauschte Ising-Modell

Initialisiere  $x$ ,  $T=T_0$ ,

Gebe  $y$  und die Rauschparameter ein

for  $t=1$  to  $N$  do

  for  $j=1$  to  $n$  do

$i=\text{Next}(j)$

$$x'_i = 1 - x_i$$

$$d = \beta \sum_{i \sim j} I_{[x_i=x_j]} + h_i(x_i, y_i)$$

$$d' = \beta \sum_{i \sim j} I_{[x'_i=x_j]} + h_i(x'_i, y_i)$$

$$p = \exp(\min(0, d' - d) / T)$$

$U \sim \text{Uniform}(0,1)$

    if  $U < p$  then  $x_i = x'_i$   
  end if

  end for

$T=T(t)$

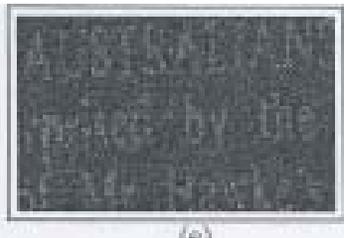
end for

Return  $x$

## Gaußsches Rauschen

MAP

MPM

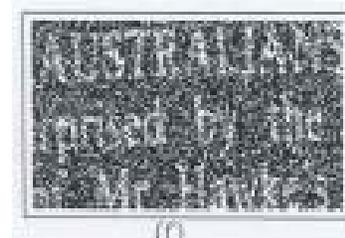


mit Gaußschem Rauschen  
erzeugtes Bild ( $\sigma^2=0.5$ )

## binäres Rauschen

MAP

MPM



mit binären Rauschen  
erzeugtes Bild ( $p=0.25$ )



das zugrundeliegende  
Bild

# Schätzung der Parameter

- Wir betrachten binäre Bilder mit Gaußschen Rauschen.

$$\pi(\mathbf{x} | \mathbf{y}) \propto \frac{(\sigma^2)^{-n/2}}{Z(\beta)} \exp\left(\beta \sum_{i \sim j} I_{[x_i = x_j]} - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu_{x_i})^2\right)$$

- Es gibt vier Parameter zu schätzen:

$\beta$

$\sigma^2$

$\mu_0$  Farbenwert für den Hintergrund

$\mu_1$  Farbenwert für den Vordergrund

- Falls das Bild  $x$  bekannt ist, können wir durch die Maximum Likelihood Methode die Parameter

$\sigma^2, \mu_i, i=0,1$  schätzen.

$$(\hat{\sigma}^2, \hat{\mu}_0, \hat{\mu}_1) = \arg \max(\pi(y | x; \sigma^2, \mu_0, \mu_1))$$

$$\hat{\beta} = \arg \max(\pi(x; \beta))$$

- Nach der Ableitung der Likelihoodfunktion erhalten wir :

$$\hat{\mu}_j = \frac{1}{\#\{i : x_i = j\}} \sum_{i: x_i=j} y_i \quad j=0,1$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{\mu}_{x_i})^2$$

- Da die Normalisierungskonstante  $Z$  von  $\beta$  abhängt, wird der Maximum Pseudo Likelihood für die Schätzung von  $\beta$  verwendet.

$$\pi(\mathbf{x} \mid \beta) = \pi(x_1 \mid x_2, \dots, x_n, \beta) \pi(x_2 \mid x_3, \dots, x_n, \beta) \dots \pi(x_n \mid \beta)$$

$$\pi(x_i \mid \mathbf{x}_{-i}, \beta) = \pi(x_i \mid x_{\partial i}, \beta)$$

$$\text{PSL}(\beta) = \prod_{i=1}^n \pi(x_i \mid x_{\partial i}, \beta)$$

$$\pi(x_i \mid x_{\partial i}, \beta) = \frac{1}{1 + \exp(-\beta \sum_{i \sim j} I_{[x_i = x_j]} - I_{[x_i \neq x_j]})}$$

- Die Maximumstelle von  $\text{PSL}(\beta)$  ist der Schätzer für  $\beta$ .
- Sie wird numerisch berechnet.

# Graustufenbilder

- Vorher war der Phasenraum eine endliche Menge.
- Jetzt ist der Phasenraum ein Intervall.
- Das Bild ist stückweise glatt.
- Es besteht aus Bereichen getrennt durch Kanten.
- Das Bild ist eine Realisation eines MRF

$$\pi(\mathbf{x}) \propto \exp(-U(\mathbf{x}))$$

$$\text{mit } U(\mathbf{x}) = \sum_{C \in \mathcal{N}} V_C(\mathbf{x}) \quad \text{die Energiefunktion}$$

- Ziel: von den beobachteten Daten  $y$  werden die Kanten bestimmt.
- Wir nehmen an, dass  $y$  durch das Gaußsche Rauschen erzeugt sind.
- Wir betrachten das 4-Nachbarschaftssystem
- Die Potentialfunktion hat die Form

$$V_C(\mathbf{x}) = g(x_i - x_j)$$

wobei  $C = \{i, j\}$  und  $g$  eine Funktion ist, die die Verletzung der Glattheit bestraft

Für stückweise glatte Bilder nimmt man  $g(x_i - x_j) = \min\{(x_i - x_j)^2, \alpha\}$

- Die Kantenbestimmung hängt eng zusammen mit der Wiederherstellung der Bilder.
- Die Kanten sind die Bereiche im Bild, in denen die Glättung verletzt ist.

- Geradenprozess zur Entdeckung der Kanten

- $y = x + \varepsilon$  wobei  $\varepsilon \sim N(0, \sigma^2 I_n)$

- $\pi(y | x) \propto \exp\left(\sum_{i=1}^n \frac{(y_i - x_i)^2}{2\sigma^2}\right)$

$$U(y|x) = \sum_{i=1}^n \frac{(y_i - x_i)^2}{2\sigma^2} \quad \text{die Likelihood Energie}$$

- $U(x) = \sum_{i \sim j} g(x_i - x_j)$

- $U(x | y) = \sum_{i=1}^n (y_i - x_i)^2 + \lambda \sum_{i \sim j} g(x_i - x_j)$  die A-Posteriori-Energie

Es wird die Funktion  $g$  modifiziert:

$$g(x_i, x_j, L_{ij}) = (x_i - x_j)^2 (1 - L_{ij}) + \alpha L_{ij}$$

wobei  $L_{ij} = \begin{cases} 1 & \text{falls es zwischen den Nachbarn } i \text{ und } j \text{ eine Kante gibt} \\ 0 & \text{sonst} \end{cases}$

$$U(\mathbf{x} | \mathbf{y}) = \sum_{i=1}^n (y_i - x_i)^2 + \lambda \sum_{i \sim j} g(x_i, x_j, L_{ij})$$

Blake und Zisserman, 1987, haben gezeigt:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \left\{ \sum_{i=1}^n (y_i - x_i)^2 + \lambda \sum_{i \sim j} g(x_i - x_j) \right\}$$

wobei  $g(x_i - x_j) = \min\{(x_i - x_j)^2, \alpha\}$  ohne Berücksichtigung der  $L_{ij}$

Nachdem  $\mathbf{x}^*$  berechnet wird, bestimmt man die Kanten durch

$$x_{ij}^* = \begin{cases} 1 & \text{falls } (x_i - x_j)^2 > \alpha \\ 0 & \text{sonst} \end{cases}$$

# Zusammenfassung

- Zufällige Markov Felder bieten eine Modellierung der Bilder
- Aus Markov-Gibbs Äquivalenz werden verschiedene Bildmodelle entwickelt
- diese Modelle haben viele Anwendungen in der Wiederherstellung der Bilder, Entdeckung ihrer Komponenten usw.
- Andere Anwendungen sind in den nächsten Vorträge zu erfahren

# Literatur

- . Chalmond *"Modelling and Inverse Problems in Image Analysis"* Springer, 2003.
- . S.Z. Li *"Markov Random Field Modeling in Computer Vision,,* Springer, 1995.
- . W.H.Press, S.A.Teukolsky, W.T.Vetterling, B.P. Flannery  
*„Numerical Recipes in C++: The Art of scientific computing"*
- . J.Möller *"Spatial Statistics and Computational Methods"*