

Bayessche Texturanalyse

Seminar: Bayessche Ansätze in der Bildanalyse

Patrick Eisele

Fakultät für Mathematik und Wirtschaftswissenschaften
Universität Ulm

12.Juni 2006

Inhaltsverzeichnis

- 1 Einleitung
- 2 Textur-Modellierung
 - Grundlagen
 - ψ -Modell
- 3 Textur-Segmentierung
 - Grundlagen
 - Bayessche Textur-Segmentierung
- 4 Textur-Klassifizierung
 - Grundlagen
 - Kontextuelle Bayessche Textur-Klassifizierung
 - Nicht-kontextuelle Bayessche Textur-Klassifizierung
- 5 Anhang
 - Literaturangaben

Textur

- Der Begriff Textur wird in vielen verschiedenen wissenschaftlichen Disziplinen verwendet
- Es besteht keine einheitliche Definition des Begriffes Textur
- Verschiedene Definitionen je nach Anwendungsgebiet und gewähltem Ansatz
- Gängige Definitionen im Bereich der Bildanalyse:

Definition Textur

Definition Textur

Textur ist ein Attribut, das die räumliche Anordnung der Grauwerte in einer Region darstellt.

Definition Textur

Definition Textur

Textur ist eine visuelle Charakteristik eines Bildsegmentes, die das Segment einer bestimmten Klasse zuordnet. Diese Klassen sind oft mit physischen Interpretationen verbunden (z.B. Holz, Gras, etc.).

Texturanalyse

- Ziel der Texturanalyse ist es, mit Hilfe geeigneter Parameter Texturen zu **beschreiben und erzeugen**, Texturen zu **unterscheiden** und sie zu **klassifizieren**
- Texturanalyse besteht aus den Disziplinen :
Textur-Modellierung, Textur-Segmentierung,
Textur-Klassifizierung

Aufgabenbereiche

- **Textur-Modellierung**
Dienen zur stochastischen Beschreibung und Erzeugung von Texturen
- **Textur-Segmentierung**
Bildsegmente mit unterschiedlicher texturaler Ausprägung werden voneinander abgegrenzt
- **Textur-Klassifizierung**
Dient zur Erkennung von bestimmten Texturen und deren Einordnung in vordefinierte Klassen

Textur-Modellierung: Grundlagen

- Texturmodelle dienen dazu, Texturen zu beschreiben und zu erzeugen
- Im stochastischen Zusammenhang ist ein Texturmodell eine parametrische Familie von Zufallsfelder, die von einer gewissen Anzahl an Hyperparametern abhängt
- Eine spezifische Textur in einer Familie ist charakterisiert durch eine Anzahl an speziellen Hyperparametern
- Wahl eines Texturmodelles zur Beschreibung/ Erzeugung von Texturen hat zwei Aspekte :
 1. Angemessene Modellklasse zu wählen
 2. Passende Hyperparameter zu bestimmen

Überblick

- Ziel: Texturen stochastisch mit Hilfe eines Gibbs-Zufallsfeldes zu beschreiben
- Ein Zufallsfeld Y ist ein Gibbs-Feld, falls dessen Verteilung folgende Darstellung hat:

$$\Pi(y) = \frac{1}{Z} \exp\{-K(y)\} \quad (1)$$

- \Rightarrow Finde geeignete Energiefunktion $K(y)$, um Texturen zu beschreiben
- Im weiteren Verlauf soll eine Modellklasse näher beschrieben werden : Das Ψ -Modell

Nachbarschaftsbeziehungen

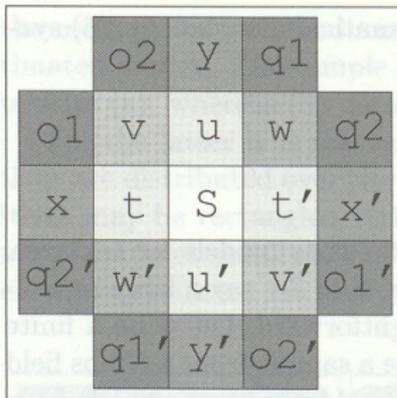
- Sei $S \subset \mathbb{Z}^2$ ein endliches Gitter
- Es gelte **Translations-Invarianz** (Jede Nachbarschaft $\partial\{s\}$ von $s \in \mathbb{Z}^2$ ist eine Verschiebung $s + \partial\{0\}$ einer Nachbarschaft $\partial\{0\}$ von 0):

$$\Pi(Y_s = y_s \mid Y_{\partial(s)} = y_{\partial(s)}) = \Pi(Y_t = (\phi_{t-s}(y))_t \mid Y_{\partial(t)} = (\phi_{t-s}(y))_{\partial(t)}) \quad (2)$$

wobei $(\phi_u(y))_t = y_{t-u}$ die Verschiebung darstellt

- Seien C_i Paarcliquen, die $0 \in \mathbb{Z}^2$ enthalten $i = 1, \dots, d$
- C_i definiert Nachbarschaftsbeziehung $t \overset{i}{\sim} s$ vom Typ i
- $\{s, t\}$ ist Translation von $C_i \Rightarrow s, t$ sind i -Nachbarn

Beispiel



- i -Nachbarschaften von $s \in S$, $i = 1, \dots, 20$
- $\{t, s\}, \{s, t'\}, \{u, s\}, \{s, u'\}$
 i -Nachbarschaften 1.Ordnung
- $\{v, s\}, \{s, v'\}, \{z, s\}, \{s, z'\}$
 i -Nachbarschaften 2.Ordnung
- $\{y, s\}, \{s, y'\}, \{x, s\}, \{s, x'\}$
 i -Nachbarschaften 3.Ordnung
- Rest sind i -Nachbarschaften 4.Ordnung

ψ -Modell

- Energiefunktion des ψ -Modells

$$K(y) = \sum_{i=1}^d \vartheta_i \sum_{t \sim_s^i} \psi(y_s - y_t) \quad (3)$$

- $\psi(\omega)$ eine Abweichungsfunktion, die symmetrisch um 0 und steigend in $|\omega|$ ist
- $\vartheta_i \in \mathbb{R}$ Gewichtungen der i -Nachbarschaften
- je nach Wahl der Funktion ψ und der Gewichtungen ϑ_i der Nachbarschaften erhält man unterschiedliche Texturen

Eigenschaften

- flache Tassenfunktion \Rightarrow Texturen mit fließenden Übergängen
- steile Parabel \Rightarrow Texturen mit klaren Übergängen
- positive $\vartheta_i \Rightarrow$ Gleiche Intensitäten der Nachbarn bevorzugt
- negative $\vartheta_i \Rightarrow$ Unterschiedliche Intensitäten bevorzugt
- kleine Werte $|\vartheta_i| \Rightarrow$ schwaches „Coupling“
- große Werte $|\vartheta_i| \Rightarrow$ starkes „Coupling“

Spezielle ψ -Modelle

- L^2 -Modell:

$$\psi(u) = u^2 \quad (4)$$

- L^1 -Modell:

$$\psi(u) = |u| \quad (5)$$

- φ -Modell:

$$\psi(u) = \min\{\lambda^2 u^2, \alpha\} \quad (6)$$

- Potts Modell:

$$\psi(u) = \begin{cases} 0 & \text{falls } u = 0 \\ 1 & \text{falls } u \neq 0 \end{cases} \quad (7)$$

Samples

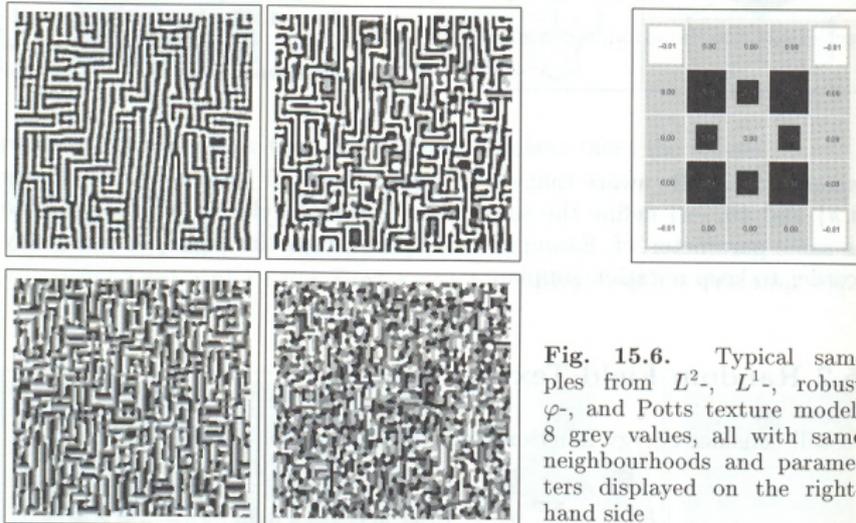


Fig. 15.6. Typical samples from L^2 -, L^1 -, robust φ -, and Potts texture model, 8 grey values, all with same neighbourhoods and parameters displayed on the right-hand side

Abbildung: Samples des L^2 -, L^1 -, φ -, Potts-Modells (8 Grauwerte, gleiche Nachbarschaften und Parameter)

φ -Modell-Samples

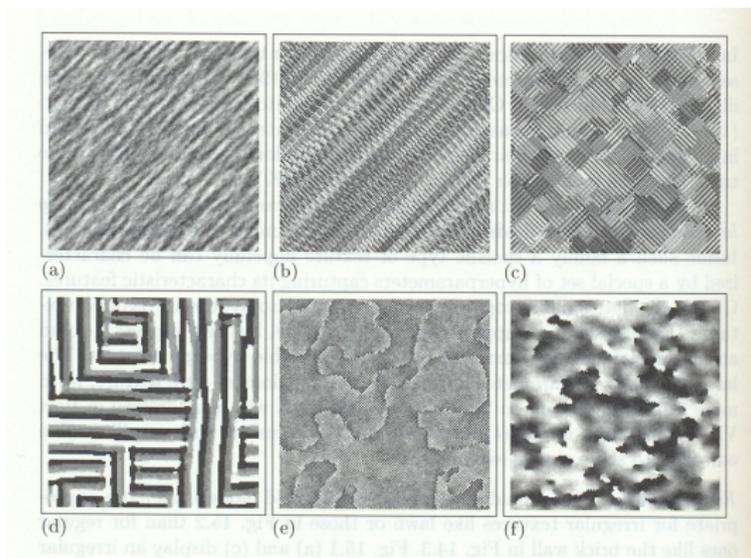


Abbildung: Samples des φ -Modells: (a-d) mit Gibbs-Sampler und 16, 8, 8, 8 Grauwerten; (e-f) mit Exchange-Algorithmus und 10 Grauwerten

Samples

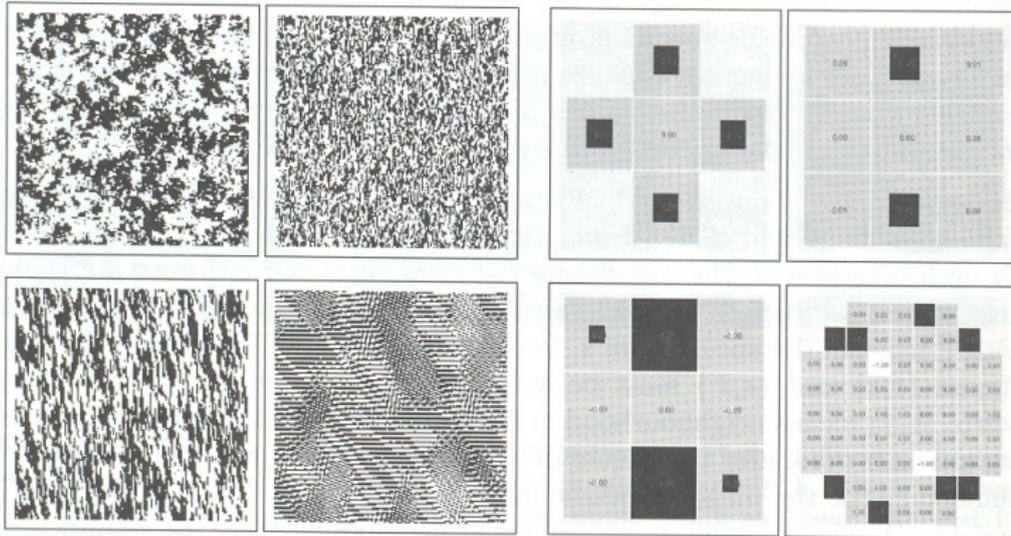


Abbildung: Samples eines binären Auto-Modells mit verschiedenen Nachbarschaften und Parametern

Anmerkungen

- Spezielle Textur ist durch Wahl der Nachbarschaftsbeziehung und deren Gewichtungen ϑ_i vollständig charakterisiert
- „*supervised learning*“ um angemessene Hyperparameter ϑ_i (Die Parameter werden mit Hilfe von Ausschnitten, die reine Textur enthalten geschätzt)
- Zahlreiche Textur-Modelle, die jeweils spezielle Eigenschaften von Texturen widerspiegeln
- All diese Modelle hängen von einer gewissen Anzahl von Hyperparametern ab

Textur-Segmentierung: Idee

- Texturen sind charakterisiert durch eine gewisse Anzahl an Eigenschaften
- \Rightarrow Texturen mit Hilfe einer minimalen Anzahl an charakteristischen Eigenschaften unterscheiden
- meist sind das einfache Eigenschaften der Intensitätsstruktur
- Texturen sind unterschiedlich, falls sie in mind. einer Eigenschaft merklich abweichen

Formale Idee

- Sei $(y_s)_{s \in S}$ Intensitätskonfiguration auf endlichem Gitter S
- Seien B und D Regionen von Pixeln, L Menge von Labels
- Regionen enthalten gleichartige (bzw. unterschiedliche) Textur
⇒ Regionen bekommen gleiches (bzw. unterschiedliches) Labels

Formale Idee

- Wähle geeignete Eigenschaften $\phi^{(i)}$, d.h. Abbildungen

$$y_D \mapsto \phi^{(i)}(y_D) \in \Phi^{(i)} (= \mathbb{R}^d) \quad (8)$$

- starre Bedingung für Gleichheit wäre:

$$d^{(i)}(\phi^{(i)}(y_B), \phi^{(i)}(y_D)) \leq c^{(i)} \quad \forall i \quad (9)$$

- wobei $c^{(i)}$ Schranken und $d^{(i)}$ ein Abstand auf $\Phi^{(i)}$
- falls diese Bedingung erfüllt ist, bekommen B und D gleiche Labels $I_B = I_D$

Bayes-Ansatz

- Beschreibung eines Bayesschen Ansatzes zur Textursegmentation nach D. Geman
- Ziel ist es eine Energiefunktion in folgender bekannter Form aufzustellen:

$$H(y, l) = K_{PL}(y, l) + V_L(l) \quad (10)$$

- wobei $K_{PL}(y, l)$ die Energiefunktion der Pixel-Label-Interaktion
- und $V_L(l)$ die a priori-Energie der Label-Label-Interaktion
- Minimieren dieser Energiefunktion \Rightarrow Optimales Labelling/Segmentierung

Voraussetzungen

- Quadratisches Gitter $S = \{(j, k) : 1 \leq j, k \leq n\} \subset \mathbb{Z}^2$
 mit einer endlichen Menge an Intensitäten G
- Teil-Gitter U von S :

$$U = U_\varrho = \{(j\varrho + 1, k\varrho + 1) : 0 \leq j, k \leq (n - 2)/\varrho\}, \quad \varrho \in \mathbb{Z} \quad (11)$$

- Pixel-Blöcke $D_u \subset S$ mit Zenrum $u \in U$, die S gleichmäßig aufteilen
- $(y_s)_{s \in S}$ Pixelprozess; $(l_u)_{u \in U}$ Labelling

Pixel-Label-Interaktion

- Pixel-Label-Interaktion :

$$K_{PL}(y, l) = \sum_{u \stackrel{\rho}{\sim} v} \Psi_{uv}(y) \Phi_{uv}(l) \quad (12)$$

- $u \stackrel{\rho}{\sim} v$ bedeutet, u und v sind Nachbarn in U_ρ
- $\Phi_{uv}(l) = \delta_{l_u, l_v}$
- Ψ_{uv} Straffunktion für Verschiedenheit in charakt. Eigenschaften der Texturen um u und v

Charakt. Eigenschaften

- Geman benutzt als charakt. Eigenschaften von Texturen folgende transformierte Daten:
 - $\phi^{(1)}(y_s) = y_s$ (Identität)
 - $\phi^{(2)}(y_s) = \max\{y_t : t \in D\} - \min\{y_t : t \in D\}$ (Intens.-Spanne)
 - $\phi^{(3)}(y_s) = |y_s - \frac{1}{|\partial D|} \sum_{t \in \partial D} y_t|$ (Residuum)
 - $\phi^{(4)}(y_s) = |y_s - (y_{s+(1,0)} - y_{s-(1,0)})/2|$ (Richtungsresiduum)
 - $\phi^{(5)}(y_s) = |y_s - (y_{s+(0,1)} - y_{s-(0,1)})/2|$ (Richtungsresiduum)

Kolmogorov-Smirnov-Abstand

- Zur Messung der Verschiedenheit in charakt. Eigenschaften wird der Kolmogorov-Smirnov-Abstand benutzt
- Betrachte die empirische Verteilungsfunktion:

$$F_{\phi^{(i)}(y_D)} : \mathbb{R} \mapsto [0, 1], \quad F_{\phi^{(i)}(y_D)}(\tau) = \frac{|\{r \in D : \phi^{(i)}(y_r) \leq \tau\}|}{|D|} \quad (13)$$

- Für den Kolmogorov-Smirnov-Abstand gilt:

$$d(\phi^{(i)}(y_B), \phi^{(i)}(y_D)) = \max\{|F_{\phi^{(i)}(y_D)}(\tau) - F_{\phi^{(i)}(y_B)}(\tau)| : \tau \in \mathbb{R}\} \quad (14)$$

Strafffunktion Ψ

- Kolmogorov-Smirnov-Abstand der transformierten Daten $\phi^{(i)}(y_{D_u})$ und $\phi^{(i)}(y_{D_v})$ überschreitet einen Grenzwert $c^{(i)}$
 \Rightarrow Texturen als unterschiedlich bewerten
- Mögliche Wahl der Strafffunktion Ψ :

$$\Psi_{uv}(y) = \begin{cases} 1 & \text{falls } d(\phi^{(i)}(y_{D_u}), \phi^{(i)}(y_{D_v})) > c^{(i)} \text{ für ein } i \\ -1 & \text{falls } d(\phi^{(i)}(y_{D_u}), \phi^{(i)}(y_{D_v})) \leq c^{(i)} \text{ für alle } i \end{cases} \quad (15)$$

Label-Label-Interaktion

- Um das Modell zu komplettieren, werden noch Straffunktionen für unerwünschte Labelkonfigurationen definiert
- Unerwünschte Labelkonfigurationen können sein:
- Eine Region ist „schwach in u “
- Eine Region ist „dünn in u “

„schwach in u “

- „schwach in u “ bedeutet, dass das Label l_u in einer Region um u schwach vertreten ist
- Eine mögliche Straffunktion könnte sein:

$$\delta_u^s(l) = \begin{cases} 1 & \text{falls } |\{v \in E_u : l_u = l_v\}| < 9 \\ 0 & \text{sonst} \end{cases} \quad (16)$$

- wobei E_u z.B. ein 5x5 Block in U_ρ ist mit Zentrum u

„dünn in u “

- „dünn in u “ bedeutet, dass das Label l_u in vertikaler und/oder horizontaler Richtung isoliert ist
- Eine mögliche Straffunktion könnte sein:

$$\delta_u^{th}(l) = \begin{cases} 2 & \text{falls } l_{u-(\varrho,0)} \neq l_u, l_{u+(\varrho,0)} \neq l_u, l_{u-(0,\varrho)} \neq l_u, l_{u+(0,\varrho)} \neq l_u \\ 1 & \text{falls } l_{u-(\varrho,0)} \neq l_u, l_{u+(\varrho,0)} \neq l_u \text{ oder } l_{u-(0,\varrho)} \neq l_u, l_{u+(0,\varrho)} \neq l_u \\ 0 & \text{sonst} \end{cases} \quad (17)$$

Gesamt-Energie

- Als Energie für die Label-Label-Interaktion ergibt sich:

$$V_L(l) = \sum_{u \in U_\varrho} (\delta_u^s(l) + \delta_u^{th}(l)) \quad (18)$$

- Die komplette Energiefunktion ergibt sich daraus zu:

$$H(y, l) = K_{PL}(y, l) + V_L(l) \quad (19)$$

$$H(y, l) = \sum_{u \sim v} \Psi_{uv}(y) \Phi_{uv}(l) + \sum_{u \in U_\varrho} (\delta_u^s(l) + \delta_u^{th}(l)) \quad (20)$$

Textur-Klassifizierung: Allgemeines

- Bilder bestehen meist aus einer gewissen Anzahl an Texturen
- Textur-Klassifizierung dient dazu, verschiedene vordefinierte Texturen zu erkennen und mit entsprechenden Labels auszuzeichnen
- Im Gegensatz zur Segmentierung werden also verschiedene Texturen nicht nur getrennt, sondern auch identifiziert

Voraussetzungen

- Sei $(y_s)_{s \in S}$ Intensitätskonfiguration auf endlichem Gitter S
- Sei U ein Teil-Gitter von S und $(D_u)_{u \in U}$ Pixelblöcke die S bedecken
- Sei eine Liste von speziellen Referenz-Texturen gegeben und jede dieser Texturen werde durch ein Label $l \in L$ repräsentiert
- Finde passende Labels l_{D_u} zu jedem Block D_u

Minimaler-Abstand-Klassifizierer

- Für jede Referenz-Textur j mit $l_j \in L$ repräsentiert der Eigenschaften-Vektor einen Punkt $P_j \in \mathbb{R}^d$
- \mathbb{R}^d wird folgendermaßen in Regionen R_j aufgeteilt:

$$R_j = \{v \in \mathbb{R}^d \mid d(v, P_j) \leq d(v, P_k), \forall k \neq j\} \quad (21)$$

d sei euklidischer Abstand

- Die Eigenschaften eines Pixelblocks definieren einen Punkt $P_D \in \mathbb{R}^d$
- $P_D \in R_j \Rightarrow D$ hat die Textur j

Bayesscher Ansatz

- Das Labelling ist selbst ein Muster, daher möglicherweise Anforderungen bezüglich dessen Struktur in das Modell integrieren
- \Rightarrow Bayessche Ansatz zur Textur-Klassifikation unterteilt in **kontextuellen** und **nicht-kontextuellen Ansatz**
- **Kontextuell:** Erwartungen an das Labelmuster werden durch angemessene apriori-Verteilung mit in das Modell eingebracht
- **Nicht-kontextuell:** Nur Intensitätskonfiguration der Pixel und apriori-Verteilung bzgl. der relativen Häufigkeit der Labels wird verwendet

Kontextuelle Klassifizierung

- Ziel ist es eine Energiefunktion in folgender bekannter Form aufzustellen:

$$H(y, l) = K_{PL}(y, l) + K_L(l) \quad (22)$$

- wobei $K_{PL}(y, l)$ die Energiefunktion der Pixel-Label-Interaktion
- und $K_L(l)$ die a priori -Energie der Label-Label-Interaktion

Voraussetzungen

- Um die Energiefunktion der Pixel-Label-Interaktion zu bestimmen, verwendet man z.B. das ψ -Modell:

$$K_{PL}(y, j) = \sum_{i=1}^d \vartheta_i^{(j)} \sum_{t \sim_s^i} \psi(y_s - y_t), \quad j \in L \quad (23)$$

- $I = (I_u)_{u \in U}$ sei das Labelmuster

Pixel-Label-Interaktion

- Unter der Bedingung, dass das gesamte Bild aus reiner Textur j besteht, erhält man folgendes Texturmodell:

$$\Pi(y \mid I_u = j, u \in U) \propto \exp\{-K_{PL}(y, j)\} \quad (24)$$

- A priori-Energie K_{PL} der Pixel-Label-Interaktion ist lokal unabhängig $\Rightarrow K_{PL}$ lässt sich aus lokalen Termen zusammensetzen:

$$J_{PL}(y, j, s) = \sum_{i=1}^d \vartheta_i^{(j)} \sum_{t \sim_s^i} \psi(y_s - y_t), \quad j \in L \quad (25)$$

Pixel-Label-Interaktion

- Für die Blockbezogenen Energien ergibt sich daraus:

$$K_{PL}(y, j, u) = \alpha^{-1} \sum_{s \in D_u} J(y, j, s) \quad (26)$$

- wobei α ein Normierungsfaktor ist, so dass jede Paar-Clique nur einmal auftaucht \Rightarrow Gesamt-Pixel-Label-Energie für ein Bild mit reiner Textur j darstellbar als:

$$K_{PL}(y, j) = \sum_u K_{PL}(y, j, u) \quad (27)$$

- Für die Energie der Pixel-Label-Interaktion ergibt sich somit:

$$K_{PL}(y, l) = \sum_u K_{PL}(y, l_u, u) \quad (28)$$

Label-Label-Interaktion

- Energieterm K_L modelliert Erwartungen bzgl. der Struktur des Labellings
- Wenn man z.B. erwartet das Texturen gebietsweise auftreten kann die Label-Label-Interaktion wie folgt gewählt werden:

$$K_L(l) = -\eta \sum_{u \sim v} \delta(l_u, l_v), \quad \eta > 0 \quad (29)$$

Gesamtenergie

- Für die Gesamtenergie gilt somit:

$$H(y, l) = K_{PL}(y, l) + K_L(l) \quad (30)$$

bzw.:

$$H(y, l) = \sum_u K(y, l_u, u) - \eta \sum_{u \sim v} \delta(l_u, l_v) \quad (31)$$

- Der Bayes-Schätzer ergibt sich daraus als MAP-Schätzer

Nicht-kontextuelle Klassifizierung

- Beim nicht-kontextuellen Ansatz wird keine spezielle Struktur des Labelmusters erwartet
- A priori-Verteilung enthält nur Informationen bzgl. der relativen Häufigkeit der Label
- Es genügt die a posteriori-Randverteilungen

$$l_u \mapsto \Pi_u(l_u | y) \quad (32)$$

zu betrachten, und daraus den Bayes-Schätzer als MPM-Schätzer zu bestimmen

Nicht-kontextuelle Klassifizierung

- Die a posteriori-Randverteilung eines Labels l_u in $u \in U$ hat folgende Form:

$$\Pi_u(l_u | y) = Z(y)^{-1} \pi(l_u) P_u(y | l_u) \quad (33)$$

- $\pi(l_u)$, die apriori-Verteilung der Label
- $P_u(y | l_u) = P(y | \text{Textur } l_u \text{ in } u)$ die bedingte Verteilung von y unter l_u
- durch Maximieren der einzelnen Randverteilungen ergibt sich der MPM-Schätzer

Anmerkung

- Es können noch zusätzlich Labels eingeführt werden:
- **doubt-label**: Für Bereiche, in denen Zweifel bzgl. des Labellings herrschen, d.h. in denen die Wahrscheinlichkeit für die Label unter eine gewisse Grenze fallen
- **out-label**: Da die Anzahl verschiedener Texturen in der Regel sehr groß und ungewiss ist oder man nur spezielle Texturen auszeichnen will, weist man allen anderen Bereichen das out-Label zu

Literaturangaben

- G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Springer, 2nd ed. 2003
- B. Chalmond. *Modelling and Inverse Problems in Image Analysis*. Springer, 2003
- <http://de.wikipedia.org>