Computer Vision: Pictorial Structures

André Steinborn - TUD/INF/KI/IS/AS

17. November 2011

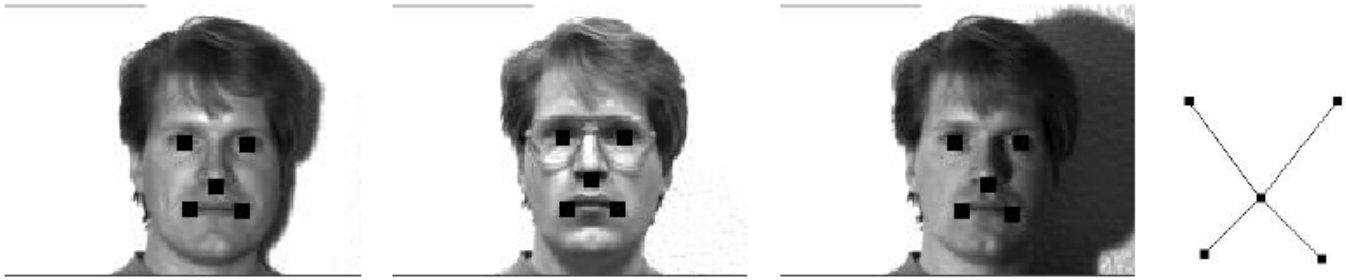
- Ziel: Lokalisierung von Objekten in Bildern.
- Objektmodell: Menge von Teilen, wobei einige Teile miteinander verbunden sind.

(Felzenszwalb, Huttenlocher 2005)

Literatur: Petro F. Felzenszwalb, Daniel P. Huttenlocher 2005: Pictorial Structures for Object Recognition.

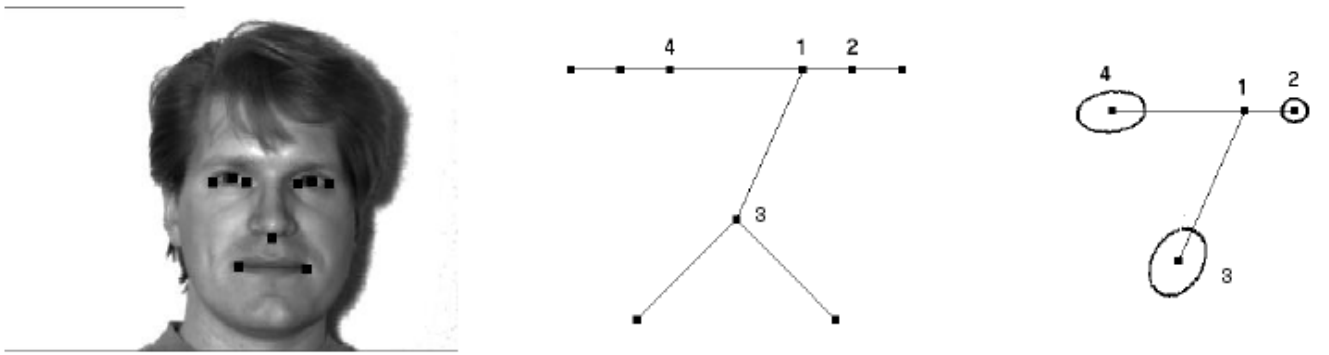
- Für jeden Anwendungsfall muss definiert werden:
  - Raum der Lokalisationen.
  - Erscheinungsmodell für jedes Teil.
  - Art der Verbindungen zwischen den Teilen.
- Drei Beispiele:
  - Iconic Model.
  - Articulated Models.
  - Boundary Models.

Modell für Gesichtserkennung mit 5 Teilen:

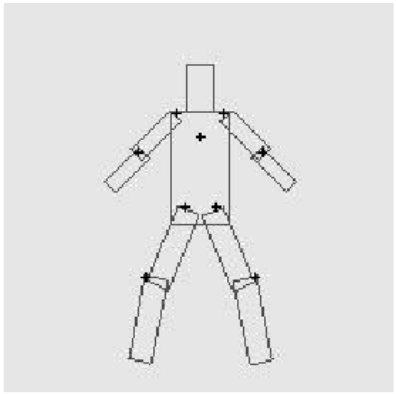
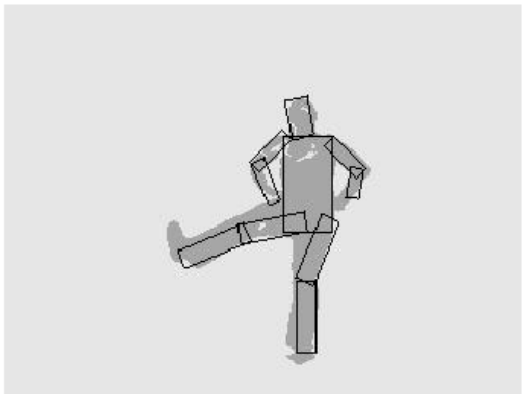


Beobachtungsmodell als Vektor der Filterantworten der Filter:

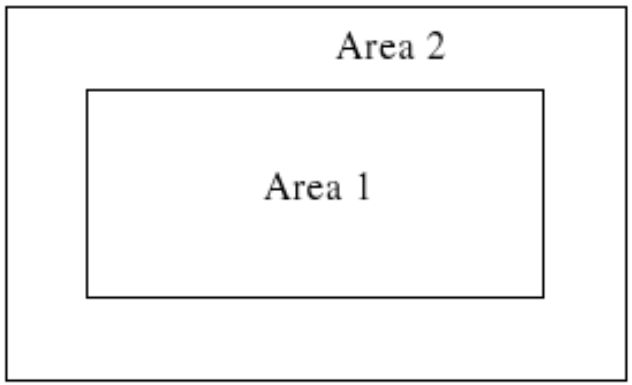
Modell für Gesichtserkennung mit 9 Teilen:

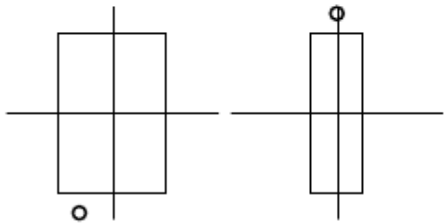
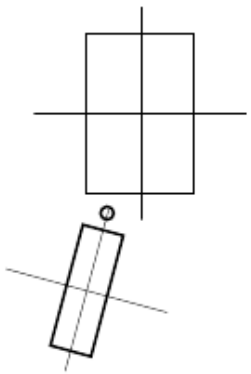


Modell für verschiedene Posen von Menschen:

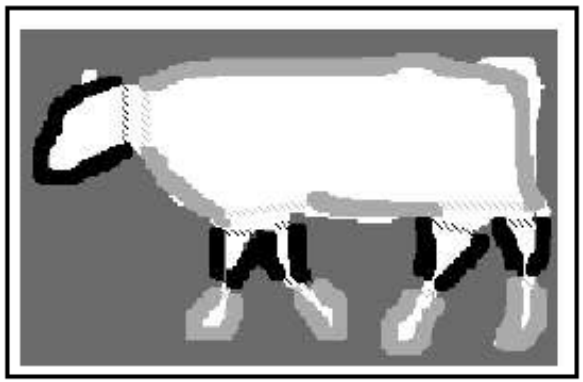



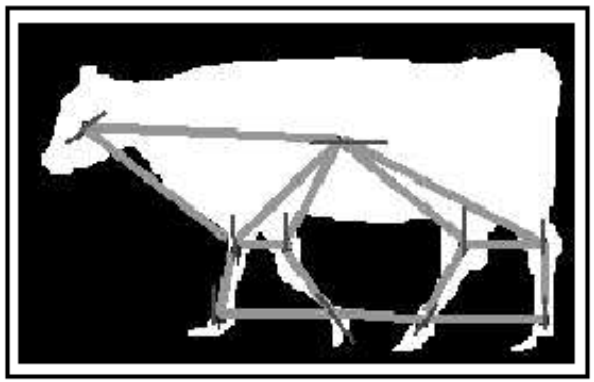

Beobachtungsmodell und Strukturmodell:



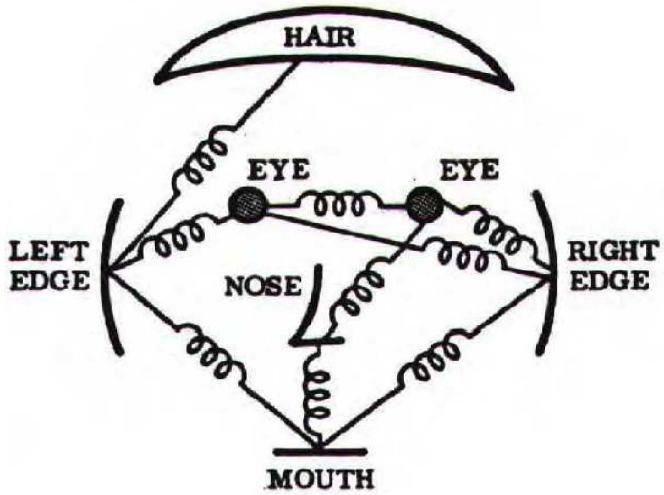



- Beobachtungsmodell:
  - Form der Umrandung.
  - Textur.
- Strukturmodell:
  - Vollverbundener Graph.
  - Potts-Modell.





Literatur: Kumar, Torr, Zisserman 2004: Extending Pictorial Structures for Object Recognition.



(Fischler, Elschlager 1973)

- Konfiguration $L = (l_1, \ldots, l_n)$ ist Positionslabeling, wobei l_i die Position des Teils v_i beschreibt.
- Die Bildbewertungsfunktion $m_i\,(l_i)$ bewertet die Position l_i des Teils v_i als Grad der Abweichung von einem Idealwert unter Berücksichtigung der üblichen Streuung.
- Die Paarbewertungsfunktion $d_{ij}\,(l_i, l_j)$ bewertet die relative Lage der Teile v_1 und v_2, jeweils an den Positionen l_1 und l_2.

- Sei I ein Bild, L eine Konfiguration der Teile und θ ein Satz von Modellparametern.
- Zu betrachtende Verteilungen:
  - Likelihood: $p\left(I|L;\theta\right)$.
  - A-priori Verteilung: $p\left(L;\theta\right)$.
  - A-posteriori-Verteilung:

$$p\left(L|I;\theta\right) \propto p\left(I|L;\theta\right) \cdot p\left(L;\theta\right)$$

- Zu betrachtende Probleme:
  - Maximum-A-Posteriori (MAP) -Schätzung (Lokalisierung).
  - Lernen der Modellparameter.

- Modellparameter: $\theta = (u, E, c)$, wobei
  - $u = \{u_1, \ldots, u_n\}$ sind Erscheinungsparameter für jedes der n Teile.
  - E ist die Menge der Kanten und beschreibt, welche Teile miteinander verbunden sind.
  - $c = \{c_{ij} | (v_i, v_j) \in E\}$ sind die Parameter für die Verbindungen zwischen den Teilen.

Die statistische Modellierung des Bildentstehungsprozesses erfolgt durch die Likelihoodfunktion:

$$p\left(I|L,\theta\right) = p\left(I|L;u\right)$$
$$\propto \prod_{i=1}^{n} p\left(I|l_i;u_i\right)$$

wobei die Zahlen $p\left(I|l_i;u_i\right)$ die Positionen der Teile auf der Grundlage des Bildes bewerten.
Solange sich die Teile nicht überlappen ist das Produkt über diesen Zahlen eine gute Approximation der gewünschten Wahrscheinlichkeitsverteilung.

Modellierung des Vorwissens durch die A-priori-Verteilung der Konfigurationen. Für azyklische Graphen $G = (V, E)$ ergibt sich:

$$\begin{aligned} p(L;\theta) &= p(L;E,c) \\ &= \frac{\prod_{(v_i,v_j)\in E} p(l_i,l_j;c_{ij})}{\prod_{v_i\in V} p(l_i;c)^{deg(v_i)-1}} \\ &\propto \prod_{(v_i,v_j)\in E} p(l_i,l_j;c_{ij}) \end{aligned}$$

Die Proportionalitätsrelation in der letzten Zeile gilt, weil kein Vorwissen über die absoluten Positionen l_i modelliert wird und somit $p(l_i;c)$ gleichverteilt sein muss, weswegen der Nenner in der mittleren Zeile konstant ist.

Beachte: Sowohl $p(l_i,l_j;c_{ij})$ als auch $p(L;E,c)$ sind uneigentliche A-priori-Verteilungen, ergeben aber in Verbindung mit der Likelihoodfunktion eine eigentliche A-posteriori-Verteilung.

Die Wahrscheinlichkeiten der Konfigurationen L nachdem ein Bild I beobachtet wurde wird durch die A-posteriori-Verteilung beschrieben:

$$\begin{aligned} p\left(L|I;\theta\right) &= \frac{p\left(I|L;\theta\right)\cdot p\left(L;\theta\right)}{\sum_{L'} p\left(I|L';\theta\right)\cdot p\left(L';\theta\right)} \\ &= \frac{p\left(I|L;\theta\right)\cdot p\left(L;\theta\right)}{p\left(I;\theta\right)} \\ &\propto p\left(I|L;\theta\right)\cdot p\left(L;\theta\right) \\ &= \prod_{i=1}^{n} p\left(I|l_i;u_i\right)\cdot \prod_{(v_i,v_j)\in E} p\left(l_i,l_j;c_{ij}\right) \end{aligned}$$

- Sei I ein Bild, L eine Konfiguration der Teile und θ ein Satz von Modellparametern.
- Zu betrachtende Verteilungen:
  - Likelihood: $p\left(I|L;\theta\right)$.
  - A-priori Verteilung: $p\left(L;\theta\right)$.
  - A-posteriori-Verteilung:

$$p\left(L|I;\theta\right) \propto p\left(I|L;\theta\right) \cdot p\left(L;\theta\right)$$

- Zu betrachtende Probleme:
  - Maximum-A-Posteriori (MAP) -Schätzung (Lokalisierung).
  - Lernen der Modellparameter.

Maximum-a-posteriori (MAP) Schätzung:

$$
\begin{aligned}
L^* &= \arg\max_L \left[\prod_{i=1}^{n} p\left(I|l_i; u_i\right) \cdot \prod_{\left(v_i, v_j\right) \in E} p\left(l_i, l_j; c_{ij}\right) \right] \\
&= \arg\min_L - \ln \left[\prod_{i=1}^{n} p\left(I|l_i; u_i\right) \cdot \prod_{\left(v_i, v_j\right) \in E} p\left(l_i, l_j; c_{ij}\right) \right] \\
&= \arg\min_L \left[\sum_{i=1}^{n} \underbrace{- \ln p\left(I|l_i; u_i\right)}_{m_i(l_i)} + \sum_{\left(v_i, v_j\right) \in E} \underbrace{- \ln p\left(l_i, l_j; c_{ij}\right)}_{d_{ij}\left(l_i, l_j\right) - \eta_{ij}} \right] \\
&= \arg\min_L \left[\sum_{i=1}^{n} m_i\left(l_i\right) + \sum_{\left(v_i, v_j\right) \in E} d_{ij}\left(l_i, l_j\right) \right]
\end{aligned}
$$

Die resultierende Aufgabe nennt man $(\min, \sum)$-Aufgabe oder Energieminimierungsaufgabe.

Unter Ausnutzung der Baumstruktur kann die Dynamische Programmierung angewandt werden:

$$l_j^* = \arg\min_{l_j} \begin{cases} m_j\left(l_j\right) + \sum_{v_c \in C_j} B_c\left(l_j\right) & Wurzel \\ m_j\left(l_j\right) + d_{ij}\left(l_i, l_j\right) + \sum_{v_c \in C_j} B_c\left(l_j\right) & Knoten \\ m_j\left(l_j\right) + d_{ij}\left(l_i, l_j\right) & Blatt \end{cases}$$

wobei

$$B_j\left(l_i\right) = \begin{cases} \min_{l_j}\left(m_j\left(l_j\right) + d_{ij}\left(l_i, l_j\right)\right) & Blatt \\ \min_{l_j}\left(m_j\left(l_j\right) + d_{ij}\left(l_i, l_j\right) + \sum_{v_c \in C_j} B_c\left(l_j\right)\right) & Knoten \end{cases}$$

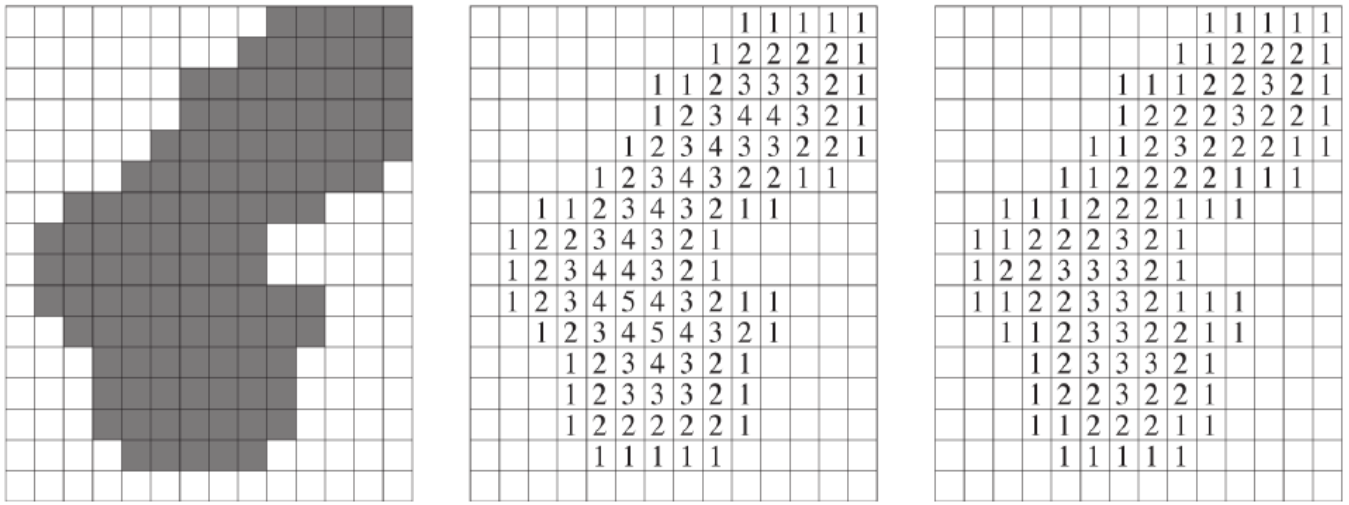
Der Viterbi-Algorithmus läuft in $O\left(h^2 \cdot n\right)$, wobei h die Anzahl der möglichen Positionen der Teile ist.

Distance Transformation I

Klassische Distance Tranformation für binäre Bilder $f : \mathcal{G} \rightarrow \{0, \infty\}$ über einem Gitter $\mathcal{G}$ für eine Punktmenge $\{p|f\,(p) = 0, p \in \mathcal{G}\} = P \subseteq \mathcal{G}$ (weiße Pixel):

$$\begin{aligned}\mathcal{D}_P\,(p) &= \min_{q \in P} d\,(p, q) \\ &= \min_{q \in \mathcal{G}} \left(d\,(p, q) + \mathbf{1}\,(q)\right)\end{aligned}$$

wobei $\mathbf{1}\,(q) = q \in P ? 0 : \infty$.



(Klette, Rosenfeld 2004)

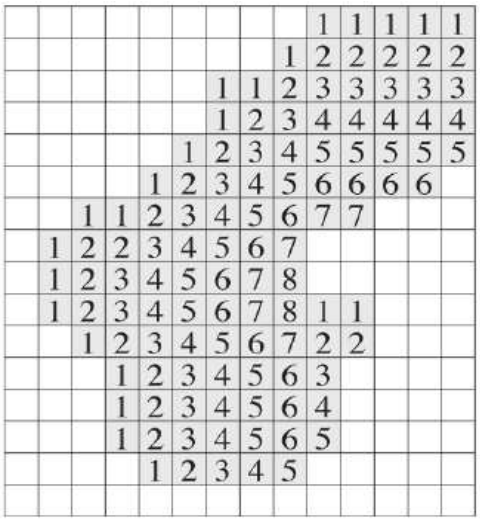
Literatur: *Reinhard Klette, Azriel Rosenfeld 2004: Digital Geometry - Geometric Methods for Digital Picture Analysis.*

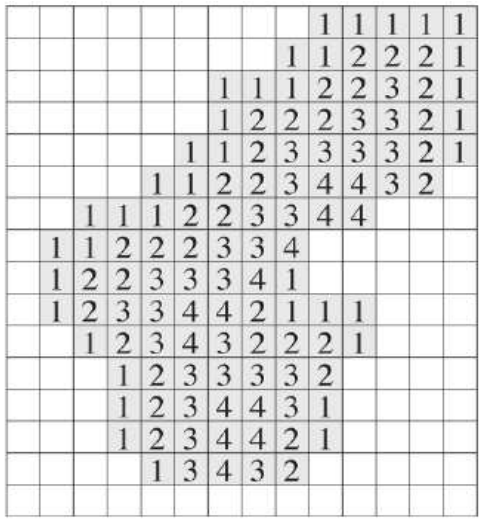
Distance Transformation II

Two-Pass-Algorithmus:

- Forward Scan:

$$f_1(p) = \begin{cases} 0 & if\ p \in P \\ \min\{f_1(q) + 1 | q \in B(p)\} & if\ p \notin P \end{cases}$$





(Klette, Rosenfeld 2004)

- Backward Scan:

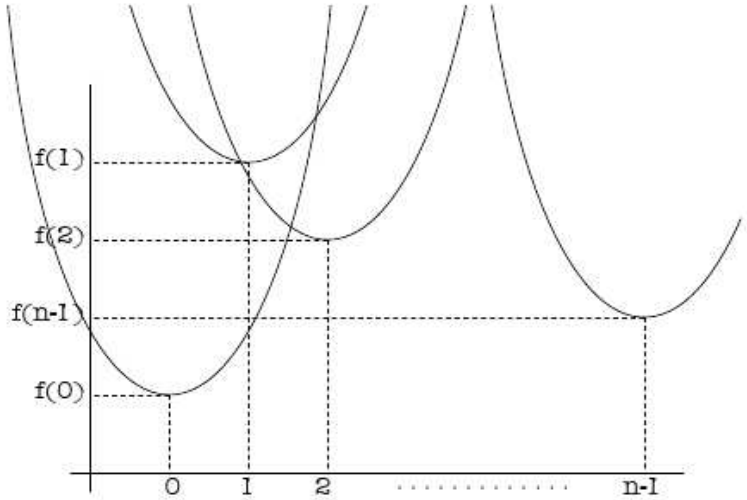
$$f_2(p) = \min\{f_1(p), f_2(q) + 1 | q \in A(p)\}$$

Verallgemeinerte Distance Transformation

Verallgemeinerung der Distance Transformation für beliebige Funktionen $f : \mathcal{G} \rightarrow \mathbb{R}$ über einem Gitter $\mathcal{G}$:

$$\mathcal{D}_f(p) = \min_{q \in \mathcal{G}} \left(d(p, q) + f(q) \right)$$

wobei man für binäre Funktionen $f : \mathcal{G} \rightarrow \{0, \infty\}$ und $f(q) = \mathbf{1}(q)$ die klassische Distance Transformation erhält.



(Felzenszwalb, Huttenlocher 2004)

Die Zeitkomplexität ist $O(h)$, wenn h die Anzahl der Gitterpunkte in $\mathcal{G}$ ist.

Literatur: Petro F. Felzenszwalb, Daniel P. Huttenlocher 2004: Distance Transforms of Sampled Functions.

Verallgemeinerte Distance Transformation:

$$\mathcal{D}_f(p) = \min_{q \in \mathcal{G}} (d(p,q) + f(q))$$

Für die MAP-Schätzung braucht man:

$$B_j(l_i) = \min_{l_j} \Big(\underbrace{d_{ij}(l_i, l_j)}_{d(l_i,l_j)} + \underbrace{m_j(l_j) + \sum_{v_c \in C_j} B_c(l_j)}_{f(l_j)} \Big)$$

$$= \min_{l_j} (d(l_i, l_j) + f(l_j))$$

Der Algorithmus läuft in $O(h' \cdot n)$, wobei h' die Anzahl der möglichen Positionen im transformierten Raum ist.

- Wegen Baumstruktur:
  - MAP-Schätzung mit dynamischer Programmierung möglich.
  - Zeitkomplexität: $O\left(h^2 \cdot n\right)$.
- Wegen Mahalanobisabstand:
  - Berechnen der Zahlen $B_j\left(l_i\right)$ als Distance Transformation (Zeitkomplexität: $O\left(h'\right)$) möglich.
  - Zeitkomplexität für MAP-Schätzung: $O\left(h' \cdot n\right)$.

- Gegeben:
  - Eine Menge von repräsentativen Beispielbildern: $\{I^1, \ldots, I^m\}$
  - sowie korrespondierende Konfigurationen: $\{L^1, \ldots, L^m\}$.
- Gesucht: Modellparameter $\theta = (u, E, c)$.
- Maximum-Likelihood (ML) - Schätzung:

$$\begin{aligned}
\theta^* &= \arg\max_{\theta} \quad p\left(I^1, \ldots, I^m, L^1, \ldots, L^m; \theta\right) \\
&= \arg\max_{\theta} \prod_{k=1}^{m} p\left(I^k, L^k; \theta\right) \\
&= \arg\max_{\theta} \prod_{k=1}^{m} p\left(I^k | L^k; \theta\right) \cdot \prod_{k=1}^{m} p\left(L^k; \theta\right) \\
&= \arg\max_{\theta} \left[\sum_{k=1}^{m} \ln p\left(I^k | L^k; u\right) + \sum_{k=1}^{m} \ln p\left(L^k; E, c\right)\right]
\end{aligned}$$

Wegen der Unabhängigkeit der beiden Summanden:

$$\begin{aligned}
u^* &= \arg\max_{u} \sum_{k=1}^{m} \ln p\left(I^k | L^k; u\right) \\
(E, c)^* &= \arg\max_{E,c} \sum_{k=1}^{m} \ln p\left(L^k; E, c\right)
\end{aligned}$$

ML-Schätzung für die Erscheinungsparameter:

$$
\begin{aligned}
u^* &= \arg\max_u \sum_{k=1}^{m} \ln p\left(I^k | L^k; u\right) \\
&= \arg\max_u \sum_{k=1}^{m} \ln \prod_{i=1}^{n} p\left(I^k | l_i^k; u_i\right) \\
&= \arg\max_u \sum_{k=1}^{m} \sum_{i=1}^{n} \ln p\left(I^k | l_i^k; u_i\right) \\
&= \arg\max_u \sum_{i=1}^{n} \sum_{k=1}^{m} \ln p\left(I^k | l_i^k; u_i\right)
\end{aligned}
$$

Weswegen die Erscheinungsparameter für jedes Teil unabhängig voneinander gelernt werden können:

$$
u_i^* = \arg\max_{u_i} \sum_{k=1}^{m} \ln p\left(I^k | l_i^k; u_i\right)
$$

ML-Schätzung für die Verbindungsparameter c_{ij}:

$$(E, c)^* = \arg\max_{E,c} \sum_{k=1}^{m} \ln p\left(L^k; E, c\right)$$

Unter Ausnutzung der Baumstruktur erhält man

$$\begin{aligned}(E, c)^* &= \arg\max_{E,c} \sum_{k=1}^{m} \ln \prod_{(v_i, v_j) \in E} p\left(l_i^k, l_j^k; c_{ij}\right) \\ &= \arg\max_{E,c} \sum_{(v_i, v_j) \in E} \sum_{k=1}^{m} \ln p\left(l_i^k, l_j^k; c_{ij}\right)\end{aligned}$$

Somit können die Parameter c_{ij} für jede mögliche Verbindung bereits ohne Kenntnis von E gelernt werden:

$$c_{ij}^* = \arg\max_{c_{ij}} \sum_{k=1}^{m} \ln p\left(l_i^k, l_j^k; c_{ij}\right)$$

ML-Schätzung für die Kantenmenge E:

$$(E, c)^* = \arg\max_{E,c} \sum_{(v_i,v_j)\in E} \sum_{k=1}^{m} \ln p\left(l_i^k, l_j^k; c_{ij}\right)$$

Für bekannte c_{ij}^* erhält man für die Kantenmenge

$$\begin{aligned} E^* &= \arg\max_{E} \sum_{(v_i,v_j)\in E} \sum_{k=1}^{m} \ln p\left(l_i^k, l_j^k; c_{ij}^*\right) \\ &= \arg\max_{E} \sum_{(v_i,v_j)\in E} \underbrace{\ln \prod_{k=1}^{m} p\left(l_i^k, l_j^k; c_{ij}^*\right)}_{-q(v_i,v_j)} \\ &= \arg\min_{E} \sum_{(v_i,v_j)\in E} q\left(v_i, v_j\right) \end{aligned}$$

Man erhält das bekannte Problem des minimal aufspannenden Baumes (MST). Dieses kann mit dem bekannten Kruskal-Algorithmus in $O\left(n^2 \cdot \ln n\right)$ gelöst werden.