MRF based Image Segmentation

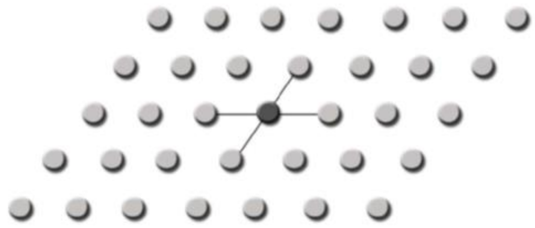
Prof. Szirányi, Tamás
Prof. Kató, Zoltán
Prof. Benedek, Csaba

Markov Random Fields in Image Segmentation

- Segmentation as pixel labeling
- Probabilistic approach
  - Segmentation as MAP estimation
  - Markov Random Field (MRF)
  - Gibbs distribution & Energy function
- Classical energy minimization
  - Simulated Annealing
  - Markov Chain Monte Carlo (MCMC) sampling
- Example MRF model & Demo
- Parameter estimation (EM)

Markov Random Fields in Image Segmentation
main principle

- Mapping the image to a graph
  - nodes are assigned to the different pixels, and the edges connect pixels which are in interaction
- Segmentation as pixel labeling:
  - each pixel gets a class-label from a task-dependent label set Λ



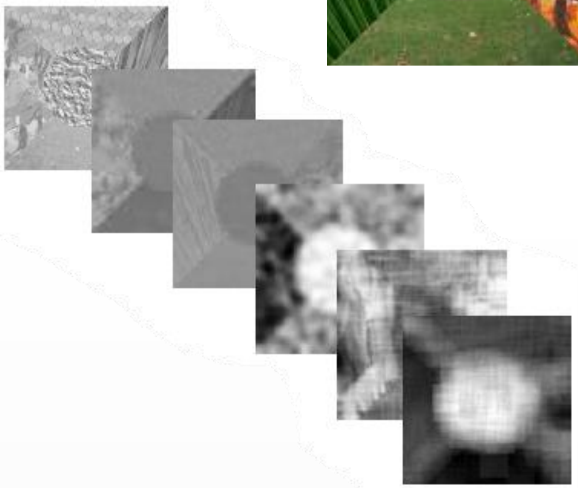
- Inverse problem formulation:
  - Instead of finding a direct algorithm to find the optimal labeling, we construct a (pseudo-) probability function which assigns a likelihood value to each possible global segmentation, then an optimization process attempts to find the labeling with the highest confidence
- What does the probability function depend on?
  - local feature vectors at each pixel (color, texture etc)
    - classes in Λ are as stochastic processes, described by different feature distributions
  - label consistency (soft) constraints between neighboring pixels
    - e.g. for preferring smooth segmentation map we penalize if two neighboring nodes have different labels

Segmentation as a Pixel Labelling Task

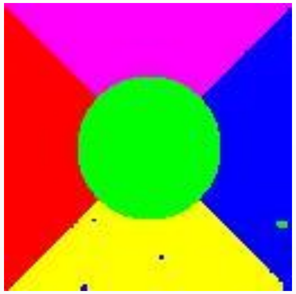
- Extract features from the input image
  - Each pixel s in the image has a feature vector $\bar{f}_s$
  - For the whole image, we have:

    $$f = \{\bar{f}_s : s \in S\}$$
- Define the set of labels Λ
  - Each pixel s is assigned a label $\omega_s \in \Lambda$
  - For the whole image, we have:

    $$\omega = \{\omega_s : s \in S\}$$
  - Ω: set of all possible ω labelings (i.e. $\omega \in \Omega$)
- For an $N \times M$ image, there are $|\Omega| = |\Lambda|^{NM}$ possible global labelings.
  - **Which one is the right segmentation?**



f



ω

Source: Zoltan Kato, http://www.inf.u-szeged.hu/~kato/

Probabilistic Approach, MAP

- Define a ***probability measure*** on the set of all possible labelings and select the most likely one.
- $P(\omega|f)$ measures the probability of a labelling, given the observed feature f
- Our goal is to find an optimal labeling $\widehat{\omega}$ which ***maximizes*** $P(\omega|f)$
- This is called the ***Maximum a Posteriori*** **(MAP)** estimate:

$$\widehat{\omega} = \underset{\omega \in \Omega}{\operatorname{argmax}} P(\omega|f)$$

Bayesian Framework

- By Bayes Theorem, we have

likelihood

prior

$$P(\omega|f) = \frac{P(f|\omega)P(\omega)}{P(f)} \propto \underbrace{P(f|\omega)}_{\text{likelihood}}\underbrace{P(\omega)}_{\text{prior}}$$

- $P(f)$ is constant
  - it does not depend on the actual labeling!
- We need to define $P(f|\omega)$ and $P(\omega)$ in our model

We will use **Markov Random Fields**

Why MRF Modelization?

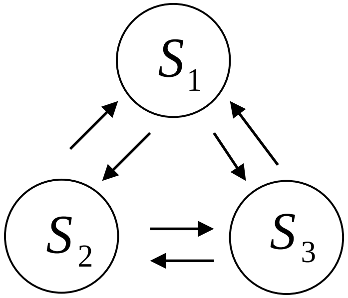
- In real images, regions are often **homogenous**; neighboring pixels usually have similar properties (intensity, color, texture, …) $\longrightarrow$ prior neighborhood constraints vs. noisy pixel level desciptors
- **Markov Random Field (MRF)** is a probabilistic model which captures such contextual constraints
  - Well studied, strong theoretical background
  - Allows Monte-Carlo Markov Chain (MCMC) sampling of the (hidden) underlying structure $\longrightarrow$ **Simulated Annealing**
  - Fast and exact solution for certain type of models $\longrightarrow$ **Graph cut** [Kolmogorov]

What is MRF?

- To give a formal definition for Markov Random Fields, we need some basic building blocks
  - Observation Field and (hidden) Labeling Field
  - Pixels and their Neighbors
  - Cliques and Clique Potentials
  - Energy function
  - Gibbs Distribution

Markov Chains vs Markov Random Fields

- Recap: Discrete Markov Chains: discrete time, discrete state stochastic processes
  - Given: set of possible states $S_1, S_2, \ldots S_N$
  - q_t: state at time t, $(t = 1, \ldots T)$
  - Observed state sequence: $q_1, q_2, \ldots q_T$
  - Markov property:

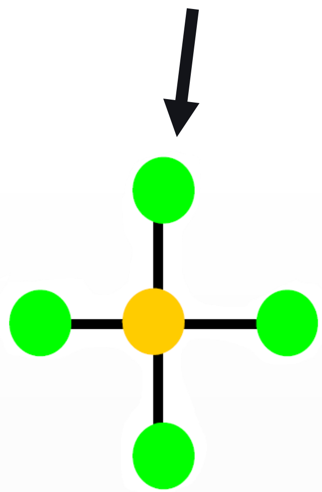


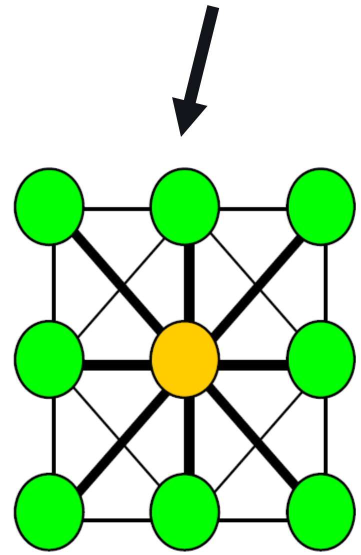
$$P\left(q_t = S_j \,|\, q_{t-1} = S_i\right) = P\left(q_t = S_j \,|\, q_{t-1} = S_i, q_{t-2} = S_k, \ldots q_1 = S_l\right)$$

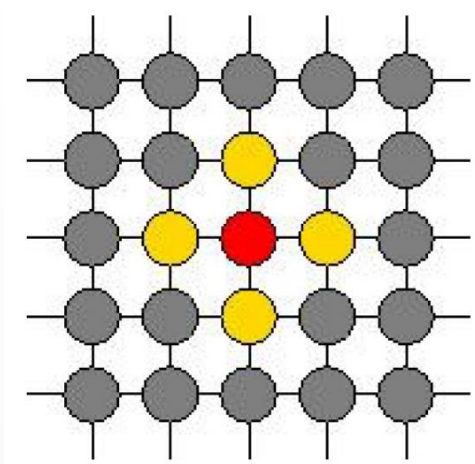
  - Conditional probability of the current state only depends on the previous state (i.e. only neighboring states interact – in time)
- Markov Random Fields: instead of temporal neighboring states, we consider the spatially neighboring pixels
  - Pixel labels are not independent, however, direct dependence is only considered between the spatial neighbors

Definition – Neighbors

- For each pixel, we can define some surrounding pixels as its neighbors.
- Example: 1st order neighbors and 2nd order neighbors





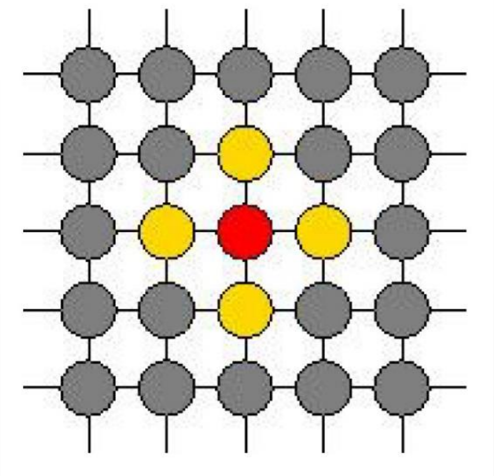


Definition – MRF

- The labeling field X can be modeled as a Markov Random Field (MRF) if
  1. For all $\omega \in \Omega$: $\mathrm{P}(X = \omega) > 0$
  2. For every $s \in S$ and $\omega \in \Omega$:

$$\mathrm{P}(\omega_s | \omega_r, r \neq s) = \mathrm{P}(\omega_s | \omega_r, r \in N_s)$$

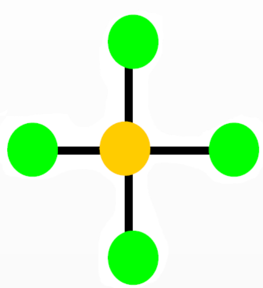
  - N_s denotes the neighbors of pixel s



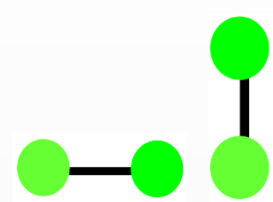
Definition – Clique

- The H-C theorem provides us an easy way of defining MRF models via ***clique potentials***.
- A subset $\mathrm{C} \subseteq S$ is called a ***clique*** if every pair of pixels in this subset are neighbors.
- A clique containing *n* pixels is called ***n***

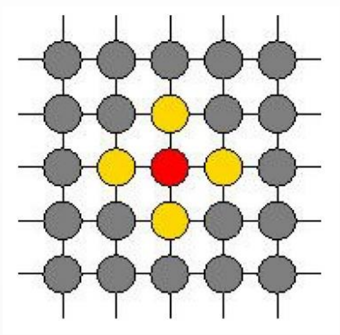
$$\mathrm{C} = C_1 \cup C_2 \cup \ldots \cup C_K$$



singleton



doubleton

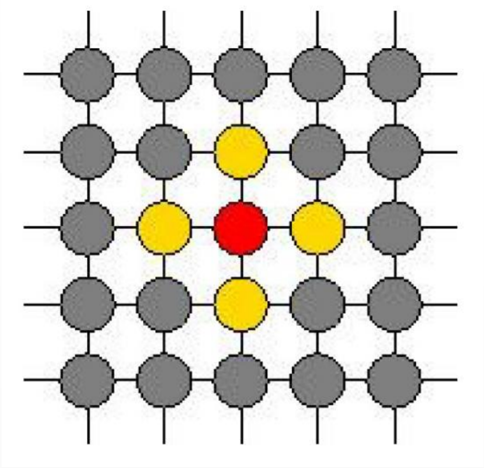


Definition – Clique Potential

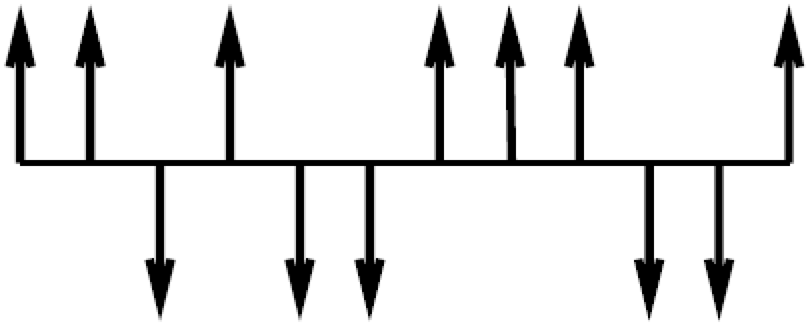
- For each clique c in the image, we can assign a value $V_c(\omega)$ which is called ***clique potential*** of c, where ω is the configuration of the labeling field
- The sum of potentials of all cliques gives us the energy $U(\omega)$ of the configuration ω.

$$U(\omega) = \sum_{c \in C} V_c(\omega) =$$

$$= \sum_{i \in C_1} V_{C_1}(\omega_i) + \sum_{(i,j) \in C_2} V_{C_2}(\omega_i, \omega_j) + \cdots$$



Ising model: up/down energies, Gibbs distribution



$$\delta_i(\omega) = \begin{cases} 1 & \text{if } \omega_i \text{ is } up \\ -1 & \text{if } \omega_i \text{ is } down \end{cases}$$

$$U(\omega) = -J \sum_{i,j} \delta_i(\omega)\delta_j(\omega) - mH \sum_i \delta_i(\omega)$$

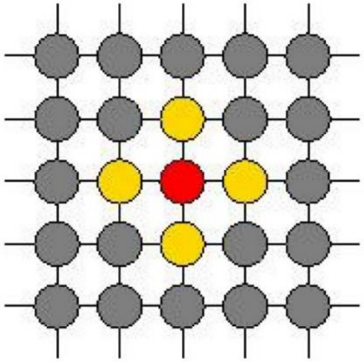
$$P(\omega) = \frac{\exp\left(-\frac{1}{kT}U(\omega)\right)}{Z}$$

$$Z = \sum_{\omega \in \Omega} \exp\left(-\frac{1}{kT}U(\omega)\right)$$

MRF segmentation model

- Pixel labels (or classes) are represented by Gaussian distributions

$$P(f_s|\omega_s) = \frac{1}{\sqrt{2\pi}\sigma_{\omega_s}} \exp\left(-\frac{\left(f_s - \mu_{\omega_s}\right)^2}{2\sigma_{\omega_s}^2}\right)$$

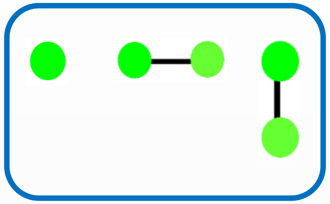


- Clique potentials
  - ***Singleton***: proportional to the likelihood of features given ω : $\log P(f|\omega)$
  - ***Doubleton***: favors similar labels at neighboring pixels – ***smoothness prior***

$$V_{C_2}(i,j) = \beta\delta\left(\omega_i, \omega_j\right) = \begin{cases} -\beta & \text{if } \omega_i = \omega_j \\ +\beta & \text{if } \omega_i \neq \omega_j \end{cases}$$

  - as β increases, regions become more homogenous

Cliques



$$\hat{\omega} = \min_{\omega \in \Omega} \sum_{s \in \mathcal{S}} V_1(\omega_s, f_s) + \sum_{C \in \mathcal{C}} V_2(\omega_C)$$

with $V_2(\omega_C)$ = second order clique-potentials, which favour **similar classes for neighboring pixels**:

$$V_2(\omega_C) = V_{\{s,r\}}(\omega_s, \omega_r) = \begin{cases} -\beta & \text{if } \omega_s = \omega_r \\ +\beta & \text{if } \omega_s \neq \omega_r \end{cases}$$

and

$$V_1(\omega_s, f_s) = \log(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2}$$

Hypothesis:

- $P(f|\omega)$ ***is Gaussian***
- $P\ (\omega)\ is\ Markovian$

Bayesian Probability

$$P(\omega_s) = \exp\left(-\sum_{(s,r)\in C_2} \beta\delta(\omega_s, \omega_r)\right)$$

$$V_{C_2}(i, j) = \beta\delta(\omega_s, \omega_r) = \begin{cases} -\beta & \text{if } \omega_r = \omega_s \\ +\beta & \text{if } \omega_r \neq \omega_s \end{cases}$$

$$P(f_s|\omega_s) = \frac{1}{\sqrt{2\pi}\sigma_{\omega_s}} \exp\left(-\frac{\left(f_s - \mu_{\omega_s}\right)^2}{2\sigma_{\omega_s}^2}\right)$$

$$P(\omega|f) = \frac{P(f|\omega)P(\omega)}{P(f)} \propto P(f|\omega)P(\omega)$$

Bayesian Probability max vs. Energy min

$$P(\omega|f) = \frac{P(f|\omega)P(\omega)}{P(f)} \propto$$

$$\propto \frac{1}{\sqrt{2\pi}\sigma_{\omega_s}} \exp\left(-\frac{\left(f_s - \mu_{\omega_s}\right)^2}{2\sigma_{\omega_s}^2}\right) * \exp\left(-\sum_{(s,r)\in C_2} \beta\delta(\omega_s, \omega_r)\right)$$

$$P(\omega|f) \propto \exp\left(\left(-\frac{\left(f_s-\mu_{\omega_s}\right)^2}{2\sigma_{\omega_s}^2}\right) + \left(-\sum\nolimits_{(s,r)\in C_2} \beta\delta(\omega_s, \omega_r)\right)\right) = \exp\left(-U(\omega)\right)$$

$$U(\omega) = -ln\left(P(\omega|f)\right) = -ln\left(\exp\ -\left(\sum_{(s,r)\in C_2} \beta\delta(\omega_s, \omega_r) + \left(\frac{\left(f_s - \mu_{\omega_s}\right)^2}{2\sigma_{\omega_s}^2}\right)\right)\right)$$

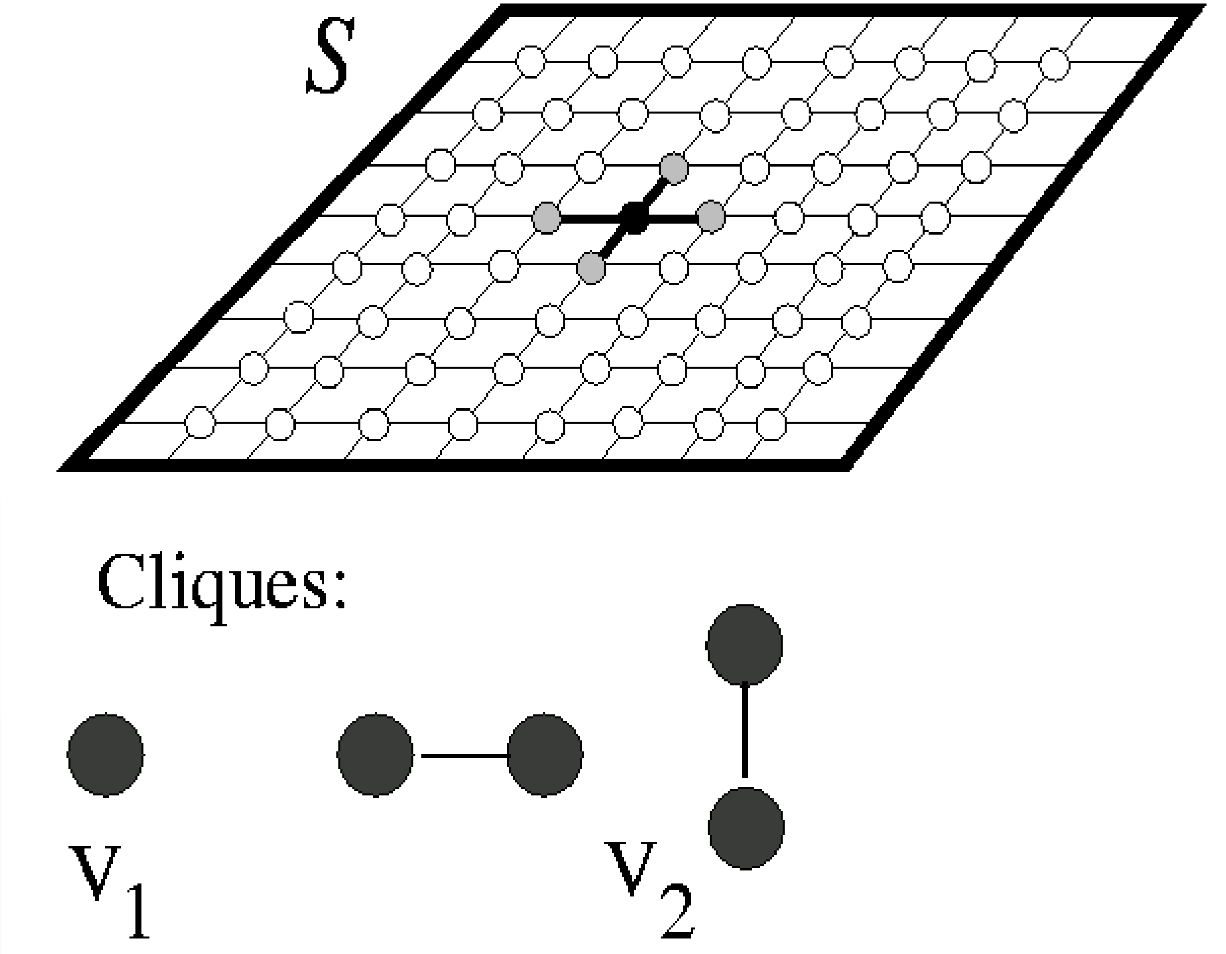
Hammersley-Clifford Theorem

- The **Hammersley-Clifford Theorem** states that a random field is a MRF if and only if $P(\omega)$ follows a **Gibbs distribution**.

$$P(\omega) = \frac{1}{Z}\exp\big(-U(\omega)\big) = \frac{1}{Z}\exp\left(-\sum_{c\in C} V_c\,(\omega)\right)$$

  - where $\mathrm{Z} = \sum_{\omega\in\Omega}\exp\big(-U(\omega)\big)$ is a normalization constant
- *Practical consequence:*
  - probability functions of MRFs have a special form: they can be factorized into small terms called ***clique potentials***, which can be locally calculated on the graph
  - this property makes possible to design the ***probability function*** in a modular way, and enables using efficient iterative optimization techniques
  - Technical note: instead of maximizing this probability function we usually minimize the minus logarithm of it which is called ***energy function***

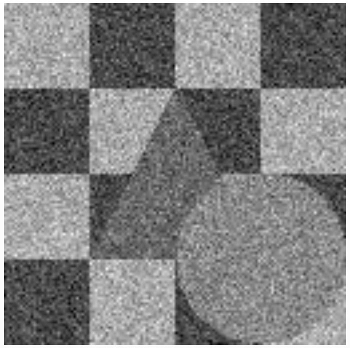
Local steps – global optimum



Segmentation of grayscale images: A simple MRF model

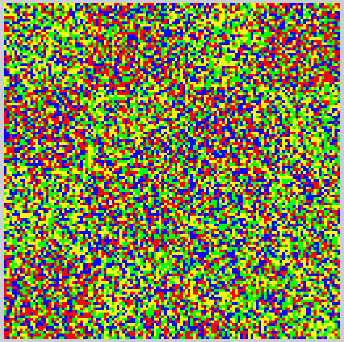
- Construct a segmentation model where regions are formed by spatial clusters of pixels with similar intensity:

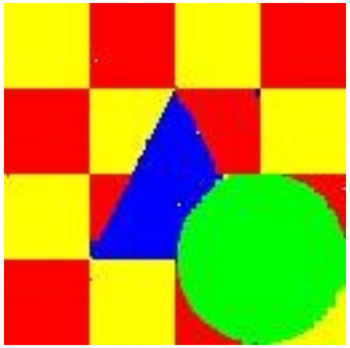
Model parameters



Input image

MRF segmentation model
+
find MAP estimate $\hat{\omega}$

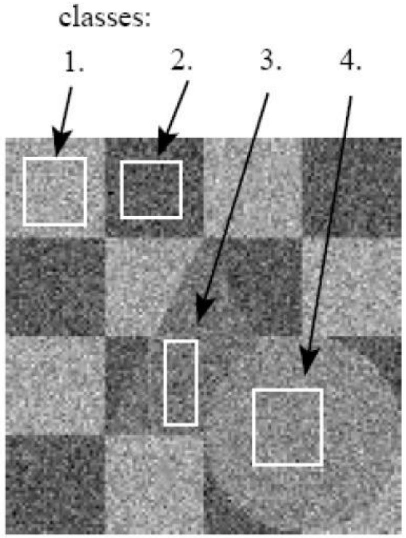




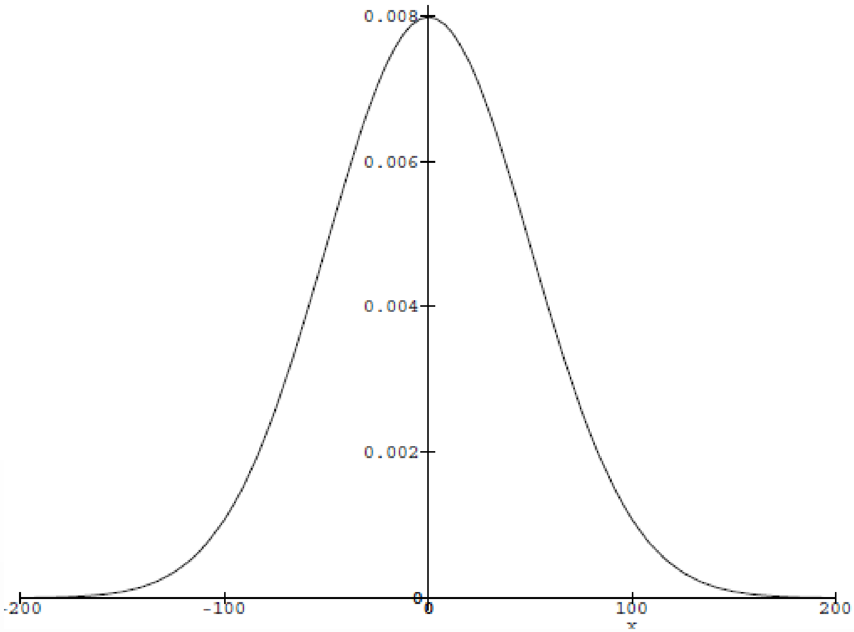
Segmentation $\hat{\omega}$

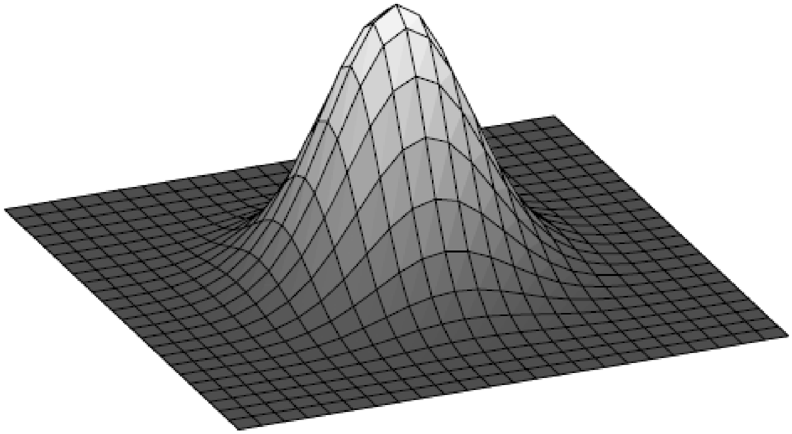
Model parameters

- Doubleton potential β
  - less dependent on the input →
    - can be fixed a priori
- Number of labels $|\Lambda|$
  - Problem dependent →
    - usually given by the user or
    - inferred from some higher level knowledge
- Each label $\lambda \in \Lambda$ is represented by a Gaussian distribution $\mathrm{N}(\mu_\lambda, \sigma_\lambda)$:
  - estimated from the input image



Gaussian/Normal distribution





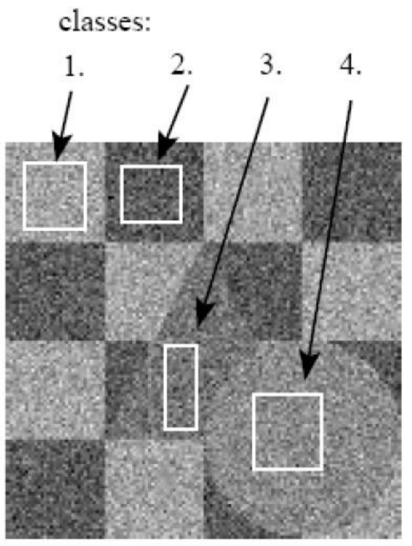
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Model parameters

- The class statistics (mean and variance) can be estimated via the ***empirical mean and variance***:

$$\forall \lambda \in \Lambda: \qquad \mu_\lambda = \frac{1}{|S_\lambda|} \sum_{s \in S_\lambda} f_s$$

$$\sigma_\lambda^2 = \frac{1}{|S_\lambda|} \sum_{s \in S_\lambda} (f_s - \mu_\lambda)^2$$



  - where S_λ denotes the set of pixels in the training set of class λ
  - a training set consists in a representative region selected by the user

Energy calculus for the optimization of MRF

$$\mathcal{E}_s(\omega) = \frac{\left(\mu_{\omega_s} - \mu_s\right)^2}{2\sigma_s^2} + \sum_{\{s,r\}\in\mathcal{C}_2} V(\omega_s, \omega_r)$$

$$V(\omega_s, \omega_r) = \begin{cases} -\beta, & \text{if } \omega_s = \omega_r \\ +\beta, & \text{if } \omega_s \neq \omega_r \end{cases}$$

Deviation from the measured

Deviation from the neighbors

Energy function

- Now we can define the energy function of our MRF model:

$$U(\omega) = \sum_{s} \left(\log\left(\sqrt{2\pi}\sigma_{\omega_s}\right) + \frac{\left(f_s - \mu_{\omega_s}\right)^2}{2\sigma_{\omega_s}^2} \right) + \sum_{s,r} \beta\delta(\omega_s, \omega_r)$$

- Recall:

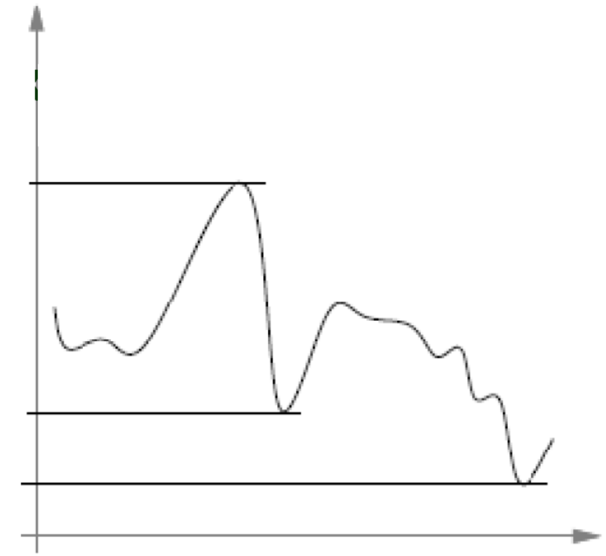
$$P(\omega) = \frac{1}{Z}\exp\left(-U(\omega)\right) = \frac{1}{Z}\exp\left(-\sum_{c \in C} V_c(\omega)\right)$$

- Hence:

$$\hat{\omega}^{MAP} = \underset{\omega \in \Omega}{\operatorname{argmax}} P(\omega | f) = \underset{\omega \in \Omega}{\operatorname{argmin}} U(\omega)$$

Optimization

- Problem reduced to the minimization of a **non-convex** energy function
  - Many local minima
- Gradient descent?
  - Works only if we have a *good* initial segmentation
- Simulated Annealing
  - Always works (at least in theory)

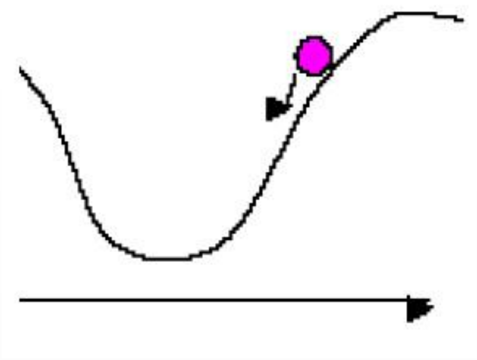


ICM (Iterated Conditional Mode)
~Gradient descent approach [Besag86]

1. Start at a „good” initial configuration ω^0 and set $k = 0$.
2. For each configuration which differs *at most* in one element from the current configuration ω^k (they are denoted by $\mathcal{N}_{\omega^k}$), compute the energy $U(\eta)$ ($\eta \in \mathcal{N}_{\omega^k}$).
3. From the configurations $\mathcal{N}_{\omega^k}$, select the one which has the minimal energy:

$$\omega^{k+1} = \underset{\eta \in \mathcal{N}_{\omega^k}}{\operatorname{argmin}} U(\eta)$$

4. Goto Step 2, with $\mathrm{k} = \mathrm{k} + 1$until convergence obtained (for example the energy change is less than a certain threshold).



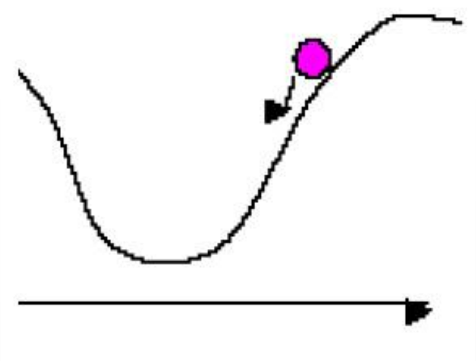
ICM (Iterated Conditional Mode)
ICM for mage segmentation models

1. Start at a „good" initial segmentation ω^0 and set $k = 0$.
2. For each segmentation which differs *at most* in one pixel's label (pixel *s*) from the current segmentation ω^k (they are denoted by $\mathcal{N}_{\omega^k}$), compute the energy $\Delta U(\eta) = U(\eta) - U(\omega^k)$ $(\eta \in \mathcal{N}_{\omega^k})$.
3. From the configurations $\mathcal{N}_{\omega^k}$, select the one which has the minimal energy:

$$\omega^{k+1} = \underset{\eta \in \mathcal{N}_{\omega^k}}{\operatorname{argmin}} \Delta U(\eta)$$

4. Goto Step 2, with $\mathrm{k} = \mathrm{k} + 1$until convergence obtained (for example the energy change is less than a certain threshold).

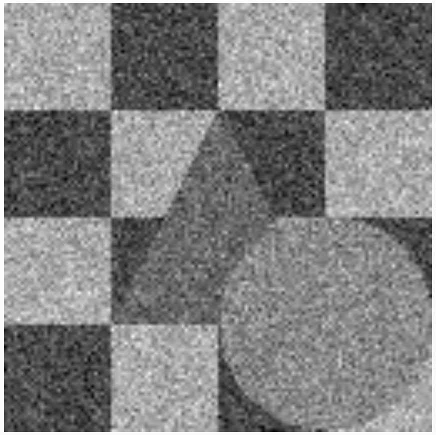
Only depens on pixel s and its four neighbors



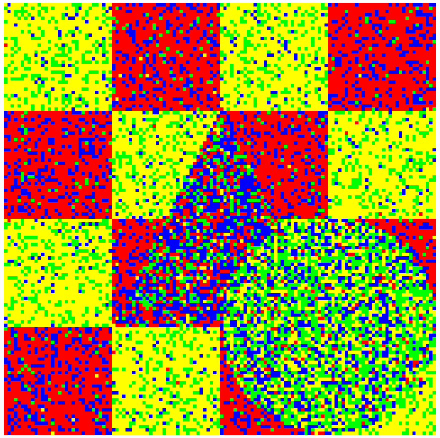
ICM initialization

- Per-pixel Maximum a Posteriori (MAP) estimate:

$$\omega_s^0 = \underset{\lambda \in \Lambda}{\operatorname{argmin}} \left(\log(\sqrt{2\pi}\sigma_\lambda) + \frac{(f_s - \mu_\lambda)^2}{2\sigma_\lambda^2} \right)$$

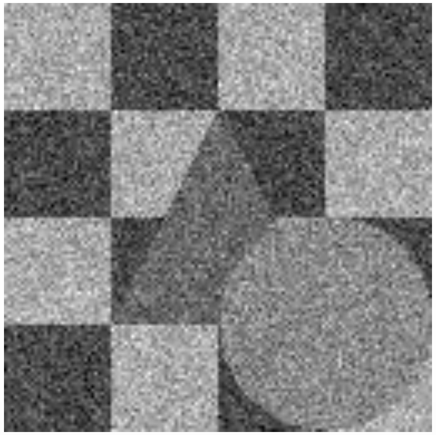


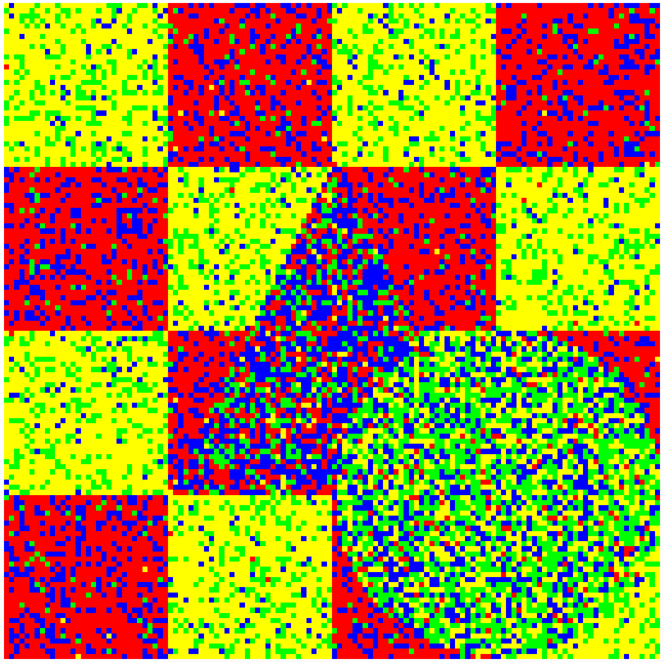
Input image



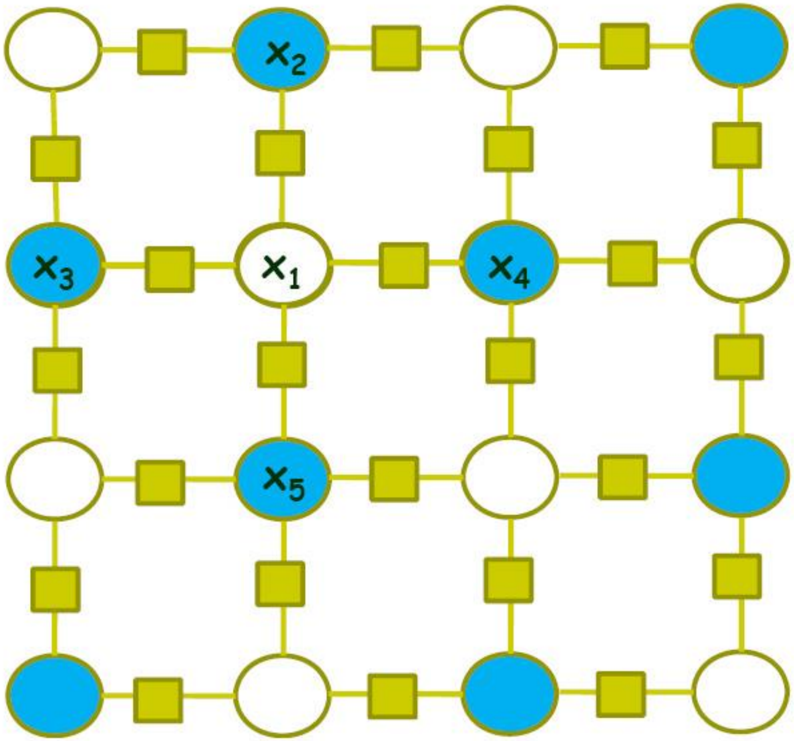
Initial label map

ICM optimization steps

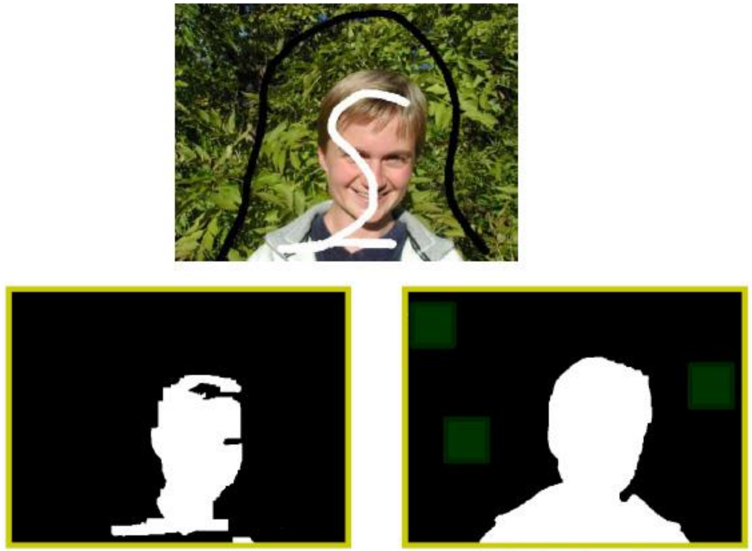




ICM vs. Simulated Annealing




means observed



ICM   Global min

Can get stuck in local minima!

Simulated Annealing: accept a move even if energy increases (with certain probability)

Slide adopted from C. Rother ICCV'09 tutorial: `http://research.microsoft.com/`

Simulated Annealing Modified Metropolis Dynamics (MMD)

1. Set $k = 0$ and initialize ω randomly. Choose a sufficiently high initial temperature $T = T_0$.
2. Construct a trial perturbation η from the current configuration ω such that η differs only in one element from ω.
3. **(Metropolis criteria)** Compute $\Delta U = U(\eta) - U(\omega)$ and accept η if $\Delta U < 0$ else accept with probability $\exp(-\Delta U/T)$ (analogy with thermodynamics):

$$\omega = \begin{cases} \eta & \text{if } \Delta U \leq 0 \\ \eta & \text{if } \Delta U > 0 \text{ and } \xi < \exp(-\Delta U/T) \\ \omega & \text{otherwise} \end{cases}$$

where ξ is a uniform random number in $[0,1[$.

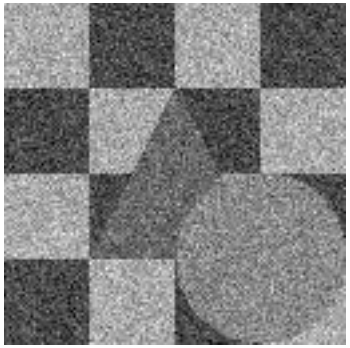
4. Decrease the temperature $T = T_{k+1}$ and goto step 2 with $\mathrm{k} = \mathrm{k} + 1$ until the system is frozen.

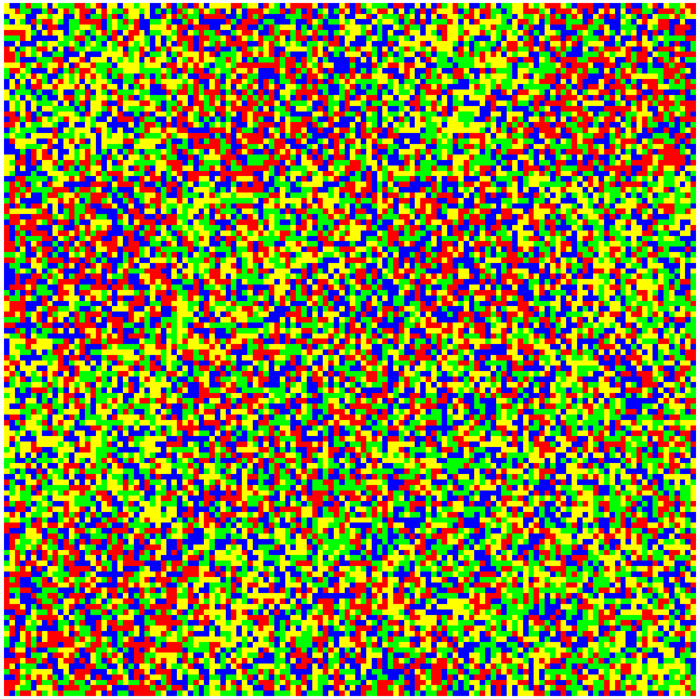
Temperature Schedule

- **In theory:** should be logarithmic – **in practice:** exponential schedule is reasonable
- **Initial temperature:** set it to a relatively low value (~4) → faster execution
  - must be high enough to allow random jumps at the beginning!
- **Schedule:** $T_{k+1} = c \cdot T_k, \quad k = 0,1,2, \dots$ (e.g. $c = 0.95$).
- **Stopping criteria:**
  - Fixed number of iterations
  - Energy change is less than a threshold

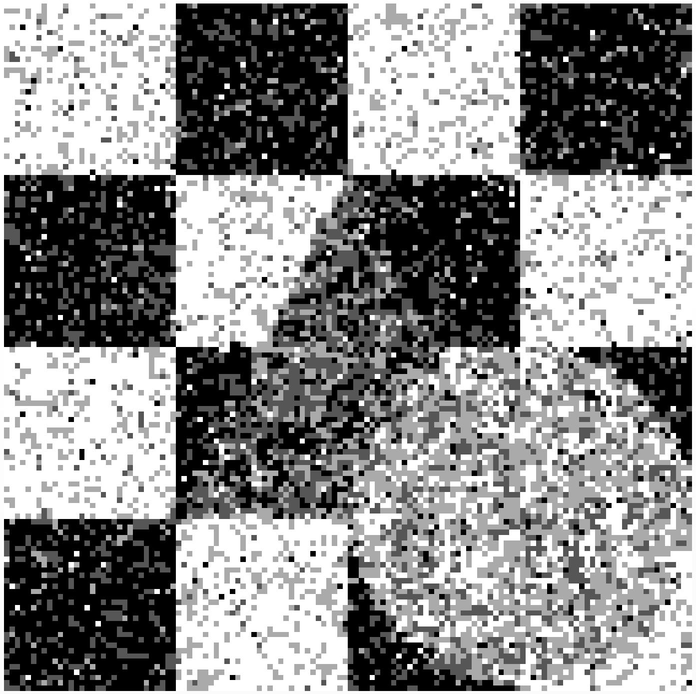
MMD segmentation

- Starting MMD: random label map!

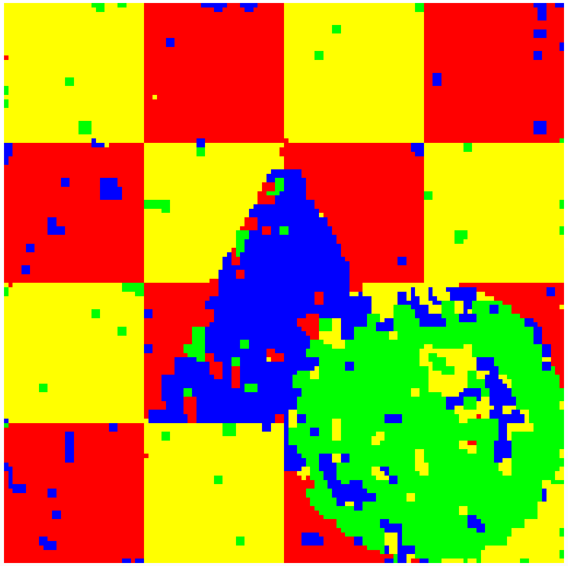




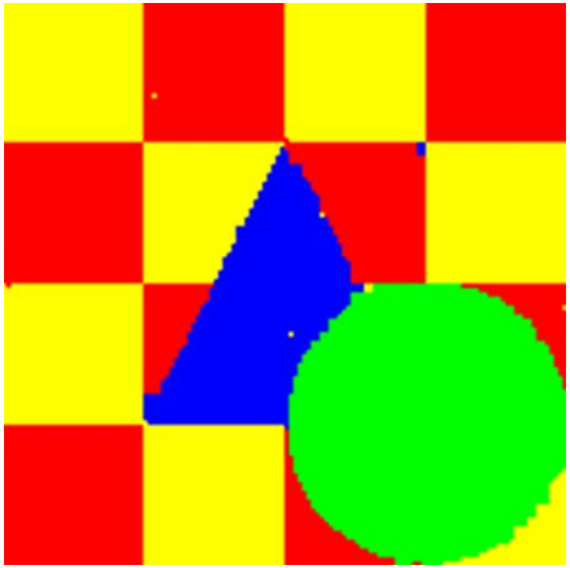
4 color MRF optimization



ICM vs MMD



ICM result

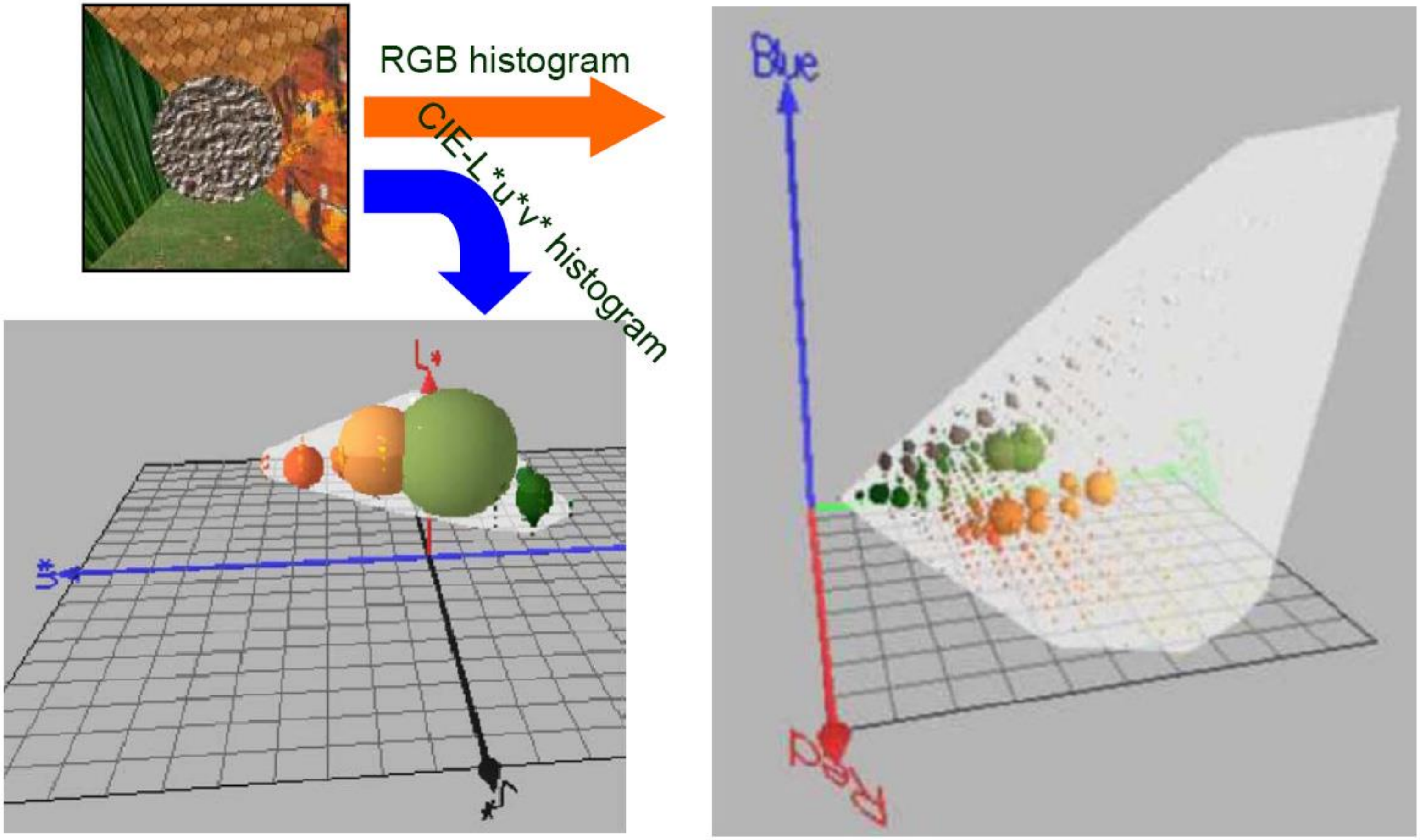


MMD result

MRF Summary

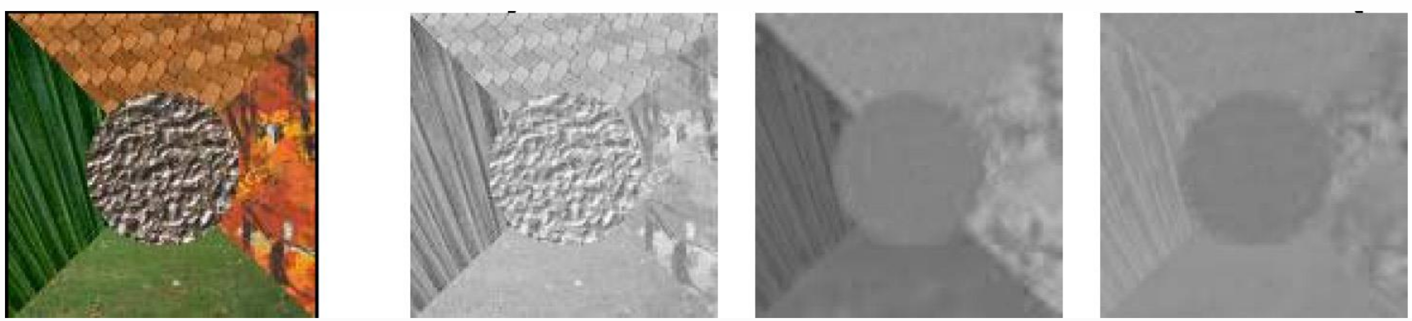
- Design your model carefully
  - Optimization is just a tool, do not expect a good segmentation from a wrong model
- What about other than graylevel features?
  - Extension to color is relatively straightforward

What color features?



Extract Color Feature

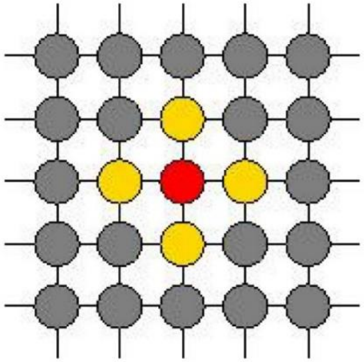
- We adopt the CIE-L*u*v* color space because it is ***perceptually uniform***.
  - Recap from earlier slides: similarly to CIE-L*a*b*, color difference can be measured here by Euclidean distance of two color vectors.
- We convert each pixel from RGB space to CIEL*u*v* space
  - We have 3 color feature images



L*   u*   v*

Color MRF segmentation model

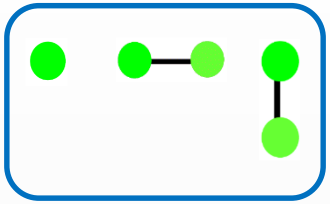
- Pixel labels (or classes) are represented by three-variate Gaussian distributions



$$P(f_s|\omega_s) = \frac{1}{\sqrt{2\pi}|\Sigma_{\omega_s}|}\exp\left(-\frac{1}{2}(\bar{f_s} - \bar{\mu}_{\omega_s})\Sigma_{\omega_s}^{-1}(\bar{f_s} - \bar{\mu}_{\omega_s})^T\right)$$

- Clique potentials
  - ***Singleton***: proportional to the likelihood of features given ω : $\log P(f|\omega)$
  - ***Doubleton***: favors similar labels at neighboring pixels – ***smoothness prior***

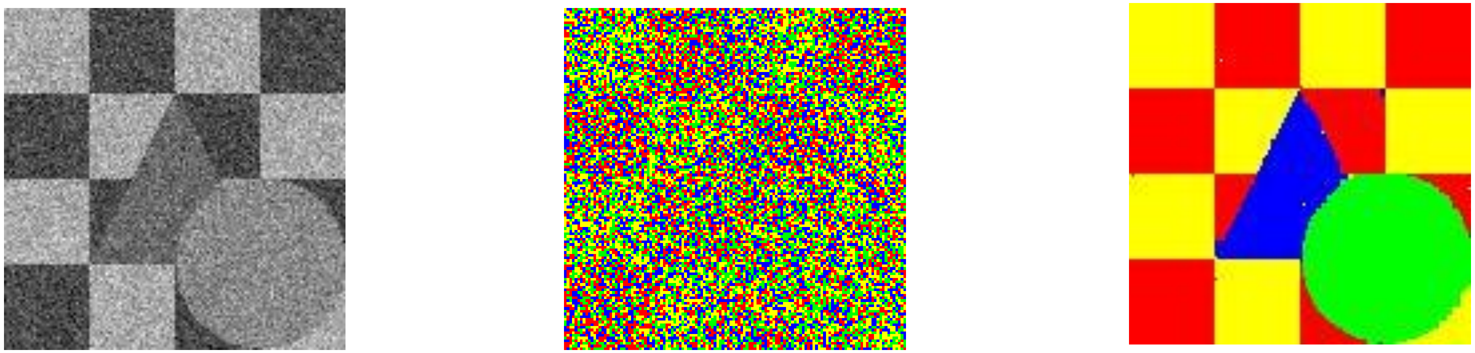
Cliques



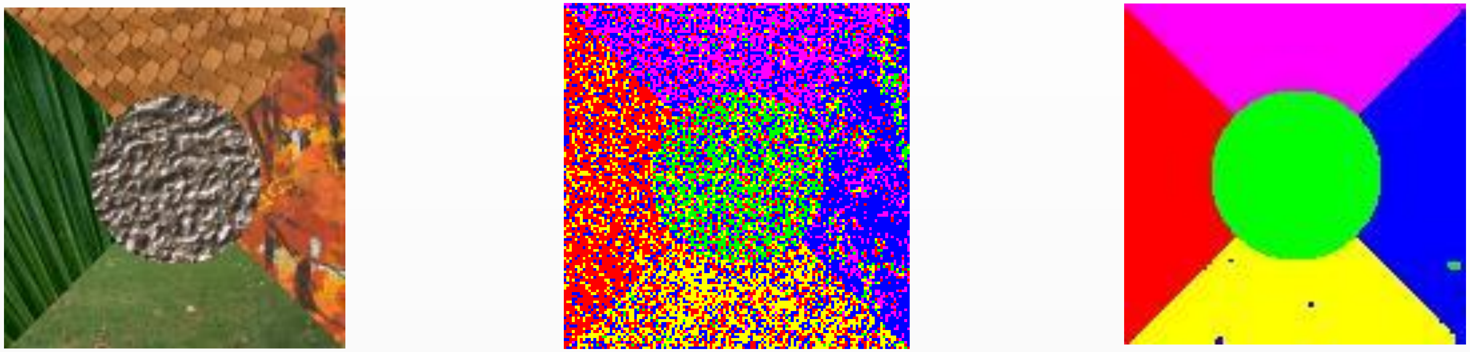
$$V_{C_2}(i,j) = \beta\delta(\omega_i, \omega_j) = \begin{cases} -\beta & \text{if } \omega_i = \omega_j \\ +\beta & \text{if } \omega_i \neq \omega_j \end{cases}$$

  - as β increases, regions become more homogenous

Segmentation examples



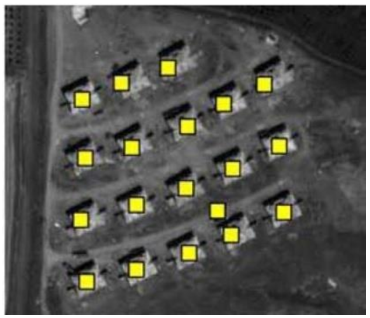
gray level based segmentation



color image segmentation

Markovian/Marked Point Process (MPP)

Robust shape searching on probabilistic description

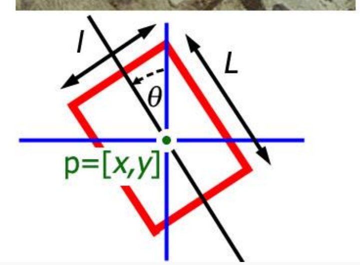


Point Process: any number of objects with featuring points ina 2-D (or 3-D) space – centrum of buildings, etc.

$$\overline{p} = \{p_1, p_2, \ldots, p_{n(\overline{p})}\}$$

- Markers: description of shape geometry (e.g. rectangle length and width, orientation)

$$M = \left]-\frac{\pi}{2}, \frac{\pi}{2}\right] \times [L_{min}, L_{max}] \times [l_{min}, l_{max}]$$



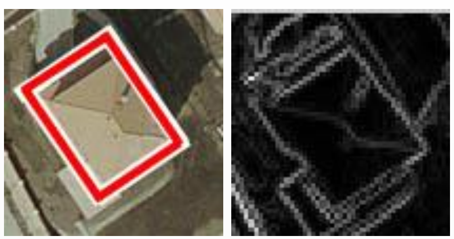
- Marked object = point+markers

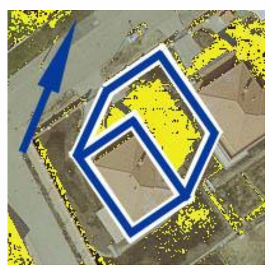
$$u_i \in K \times M$$

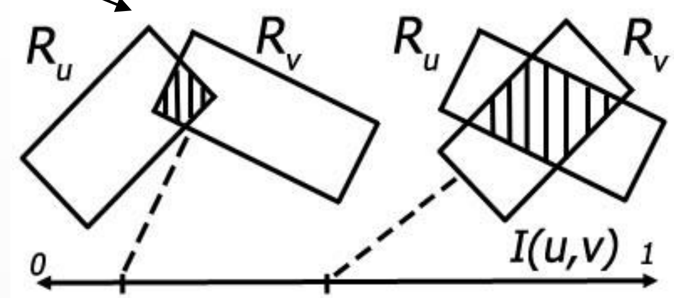
MPP energy function

- Any configuration $$\omega = \{u_1, u_1, \ldots, u_{n(\omega)}\}$$
- Space of config: $$\Omega = \bigcup_{n=0}^{\infty} \Omega_n \qquad \Omega_n = \{\{u_1, u_2, \ldots, u_n\} \mid u_i \in K \times M\}$$
- Energy function: data(d) and prior part(p)

$$\Phi(\omega) = \Phi_d(\omega) + \Phi_p(\omega)$$

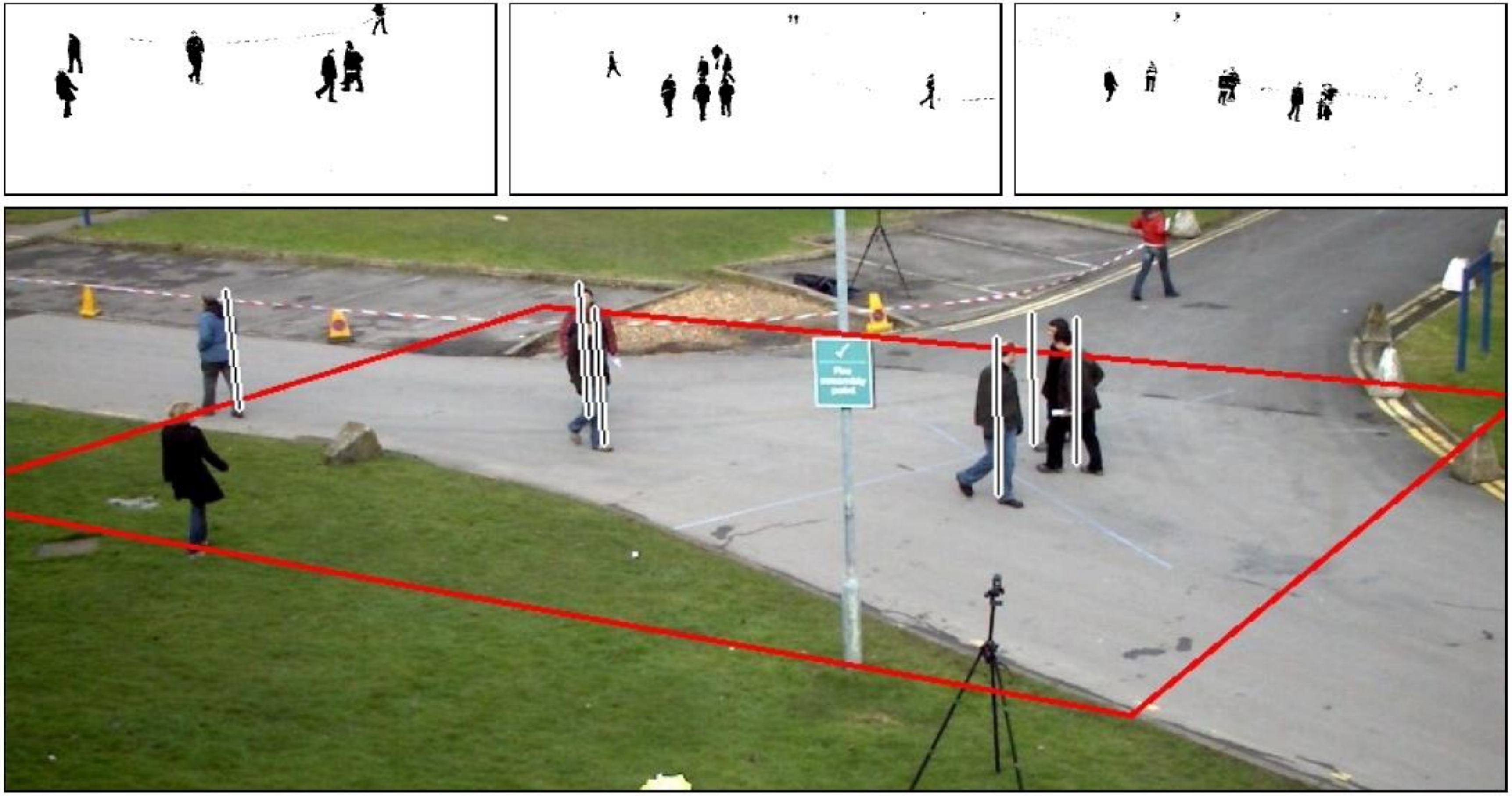






- Optimal configuration: $$\omega_{opt} = \arg\min_{\omega \in \Omega} \Phi(\omega)$$
- Efficient optimization: RJMCMC, MBD

Detection of pedestrians and their height from multiview data – fusion in 3D



Ákos Utasi, Csaba Benedek: Multi-Camera People Localization and Height Estimation using Multiple Birth-and-Death Dynamics. Workshop on Visual Surveillance, 2010