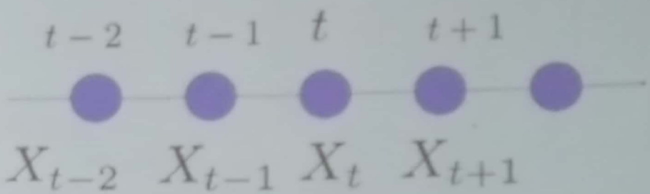
Markov Chain



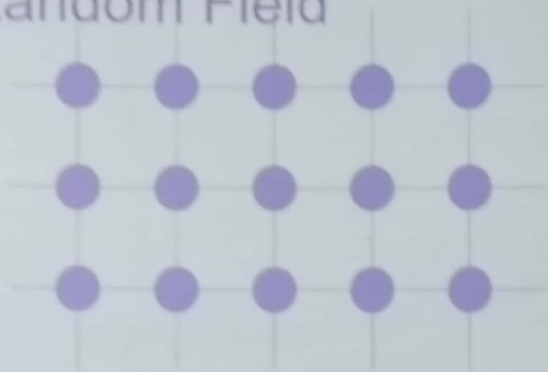
$$\Pr(X_{t+1} = y | X_t = x, X_{t-1}, X_{t-2}, \dots) = \Pr(X_{t+1} = y | X_t = x)$$
$$= K(x, y)$$

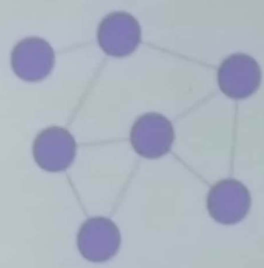
$$P(x_0, x_1, \dots, x_t) = P(x_0)P(x_1|x_0)\dots P(x_t|x_{t-1})$$

Pr(future|present, past) = Pr(future|present)
future $\perp$ past | present

Markov property: conditional independence (limited dependence)
Makes modeling and learning possible

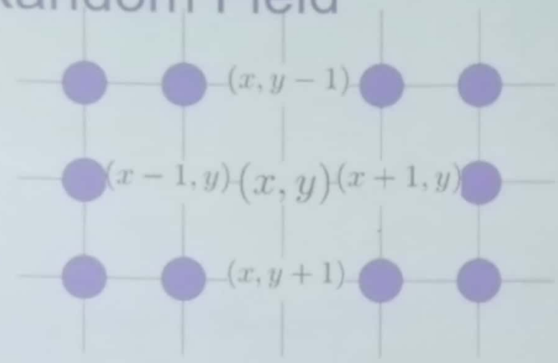
Markov Random Field





Can be generalized to any **undirected** graphs (nodes, edges)
Neighborhood system: each node is connected to its neighbors
neighbors are reciprocal
Markov property: each node only depends on its neighbors

Markov Random Field



$\mathbf{I}(x, y)$

$$P(\mathbf{I}_{x,y} | \mathbf{I}_{-(x,y)}) = P(\mathbf{I}_{x,y} | \mathbf{I}_{\mathcal{N}(x,y)})$$

What is $P(\mathbf{I})$? $\quad \mathbf{I} = (\mathbf{I}_{x,y}, \forall (x,y))$

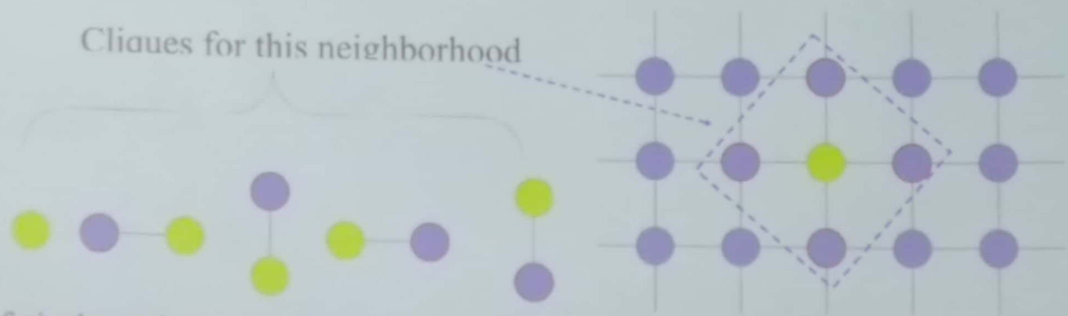
Markov Random Field

Hammersley-Clifford Theorem

$$P(\mathbf{I}) = \frac{1}{Z} \exp\{\sum_{x,y} [a(\mathbf{I}_{x,y}) + b(\mathbf{I}_{x,y}, \mathbf{I}_{x+1,y}) + c(\mathbf{I}_{x,y}, \mathbf{I}_{x,y+1})]\}$$

$Z = \sum_{\mathbf{I}} \exp\{\dots\}$ normalizing constant, partition function

$a(), b(), c()$ potential functions of cliques



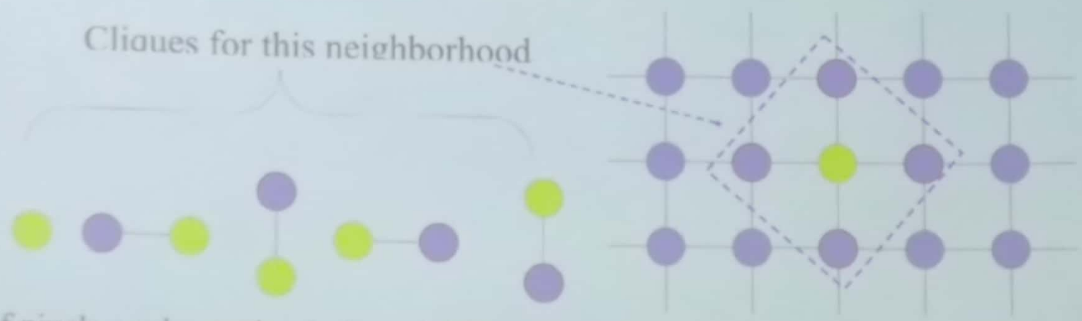
a clique: a set of pixels, each member is the neighbor of any other member

Markov Random Field

Hammersley-Clifford Theorem

$$P(\mathbf{I}) = \frac{1}{Z}\exp\{\sum_{x,y}[a(\mathbf{I}_{x,y}) + b(\mathbf{I}_{x,y}, \mathbf{I}_{x+1,y}) + c(\mathbf{I}_{x,y}, \mathbf{I}_{x,y+1})]\}$$

$$P(\mathbf{I}) = \frac{1}{Z}\exp\{\sum_{\text{cliques}} V(\mathbf{I}_{\text{clique}})\}$$ **Gibbs distribution**



set of pixels, each member is the neighbor of any other member

From Slides by S. Seitz - Universi

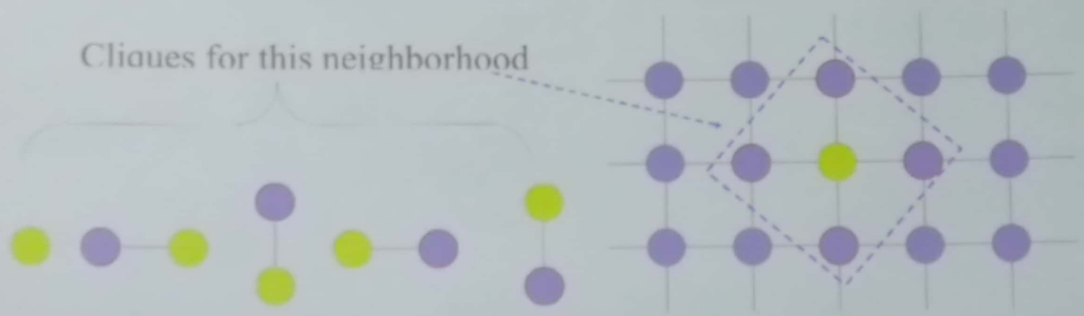
Markov Random Field

Ising Model

$$P(\mathbf{I}) = \frac{1}{Z}\exp\{\sum[\alpha\mathbf{I}_{x,y} + \beta\mathbf{I}_{x,y}\mathbf{I}_{x+1,y} + \beta\mathbf{I}_{x,y}\mathbf{I}_{x,y+1}]\}$$

$$P(\mathbf{I}) = \frac{1}{Z}\exp\{\sum_{s}\alpha\mathbf{I}_s + \sum_{s\sim t}\beta\mathbf{I}_s\mathbf{I}_t\}$$

$$\mathbf{I}_{x,y} \in \{-1,+1\} \qquad s = (x,y)$$



a clique: a set of pixels, each member is the neighbor of any other member

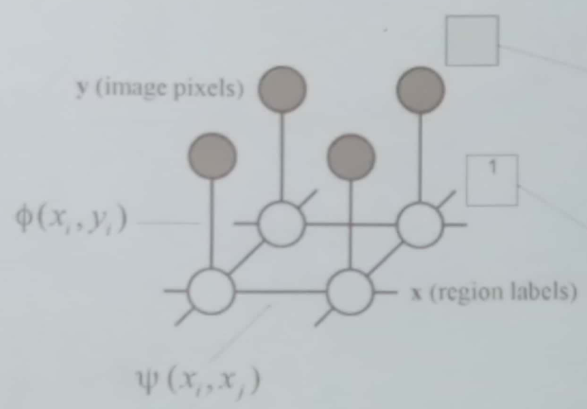
From Slides by S. Seitz - University of Washington

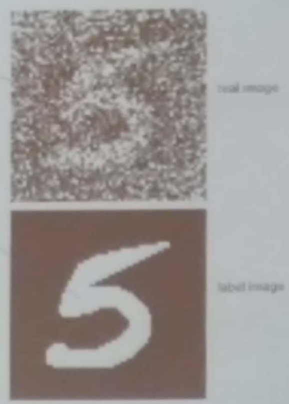
Markov Random Field

Modeling image pixel labels as MRF (Ising)

$$p(x, y) = p(x)p(y|x)$$
$$p(x|y) = p(x, y)/p(y)$$

Bayesian posterior

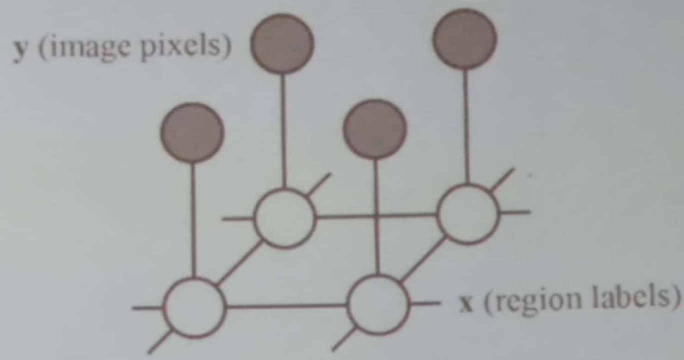




rkov Random Field

$$(\mathbf{x}^*, \theta^*) = \arg\max_{(\mathbf{x},\theta)} P(\mathbf{x}, \theta \mid \mathbf{y})$$

region labels — model param. — image pixels



$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{(i,j)} \psi \underbrace{(x_i, x_j)} \prod_i \phi (x_i, y_i)$$

label

image

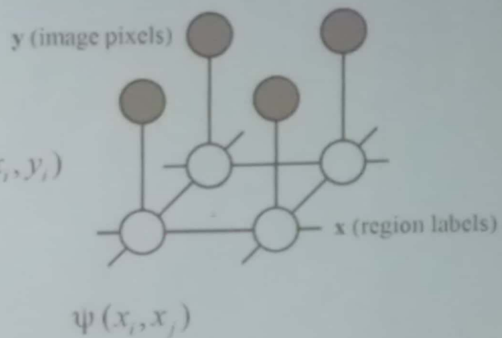
label-label compatibility function enforcing ***smoothness constraint***

neighboring label nodes

image-label compatibility Function enforcing ***data constraint***

local Observations

Markov Random Field



$$\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x} \mid \mathbf{y})$$

$$= \arg\max_{\mathbf{x}} P(\mathbf{x}, \mathbf{y}) \quad P(\mathbf{x} \mid \mathbf{y}) = P(\mathbf{x}, \mathbf{y}) / P(\mathbf{y}) = \frac{1}{Z_1} P(\mathbf{x}, \mathbf{y})$$

$$= \arg\max_{\mathbf{x}} \prod_i \phi(x_i, y_i) \prod_{(i,j)} \psi(x_i, x_j) \quad P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z_2} \prod_i \phi(x_i, y_i) \prod_{(i,j)} \psi(x_i, x_j)$$

$$\phi(x_i, y_i) = G(y_i; \mu_{x_i}, \sigma^2_{x_i})$$

$$\psi(x_i, x_j) = \exp(\delta(x_i - x_j) / \sigma^2)$$

$$\theta = [\mu_{x_i}, \sigma^2_{x_i}, \sigma^2]$$

Slides by R. Huang – Rutgers University

MAP Estimation of MRF Configuration

Observed data

$$\hat{f} = \arg\max_{f} \Pr(f \mid O)$$

Bayes rule

$$\hat{f} = \arg\max_{f} \boxed{\Pr(O \mid f)} \cdot \boxed{\Pr(f)}$$

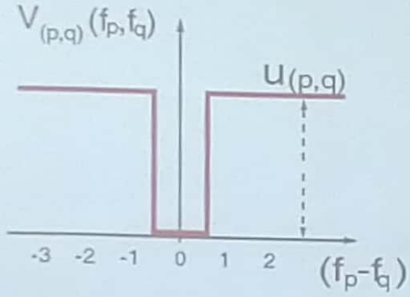
Likelihood function (*sensor noise*)

Prior (*MRF model*)

$$\hat{f} = \arg\max_{f} \exp\left\{ \sum_{p} \ln \boxed{g_p(O \mid f_p)} - \sum_{(p,q)} \boxed{V_{(p,q)}(f_p, f_q)} \right\}$$

$$\prod_{i} \phi(x_i, y_i) \qquad \prod_{(i,j)} \psi(x_i, x_j)$$

Generalized Potts Model



Clique potential

$$V_{(p,q)}(f_p, f_q) = \boxed{u_{\{p,q\}}} \cdot \delta(f_p \neq f_q)$$

Penalty for discontinuity at (p,q)

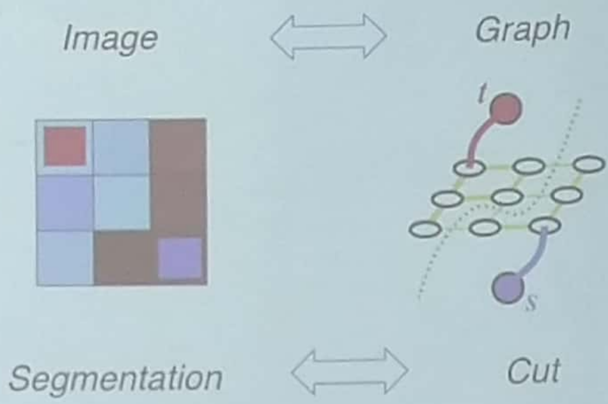
Energy function

$$E(f) = -\sum_{p} \ln g_p(O \mid f_p) + 2 \sum_{\{p,q\}} \boxed{u_{\{p,q\}}} \cdot \delta(f_p \neq f_q)$$

Generalized Potts Model

The main idea of graph-cut based MRF optimization is to construct a graph such that

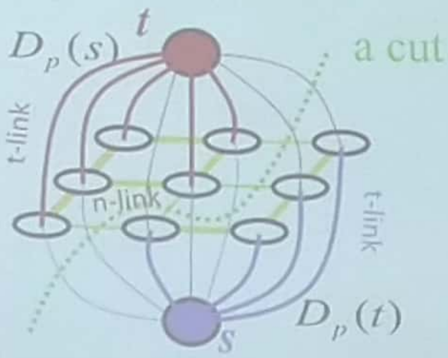
- there is a one-to-one correspondence between cuts of graph and configurations of the MRF
- the total cost of the cut is exactly the same as the total energy of the configuration.



Generalized Potts Model

$$E(f) = \sum_{p} D_p(f_p) + \sum_{\{p,q\}} u_{\{p,q\}} \cdot \delta(f_p \neq f_q)$$

t-links n-links $f_p \in \{s,t\}$



A s-t cut corresponds to one configuration of binary labeling

$$C(S,T) = E(f)$$

What about other energy functions

Let E be a function of n binary variables with the following form:

$$E(L) = \sum_{p} D_p(f_p) + \sum_{p,q \in N} V(f_p, f_q)$$

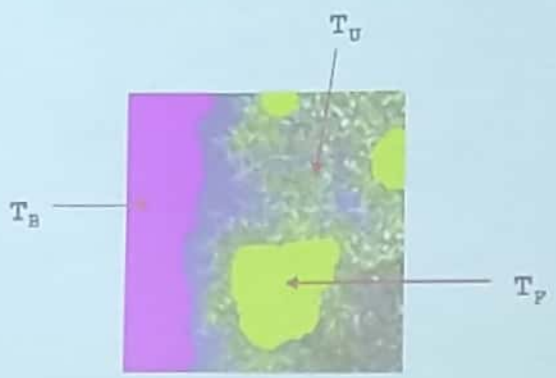
Then, E is graph-representable if and only if [3]:

$$V(0,0) + V(1,1) \leq V(0,1) + V(1,0)$$

Functions satisfying above condition are called regular

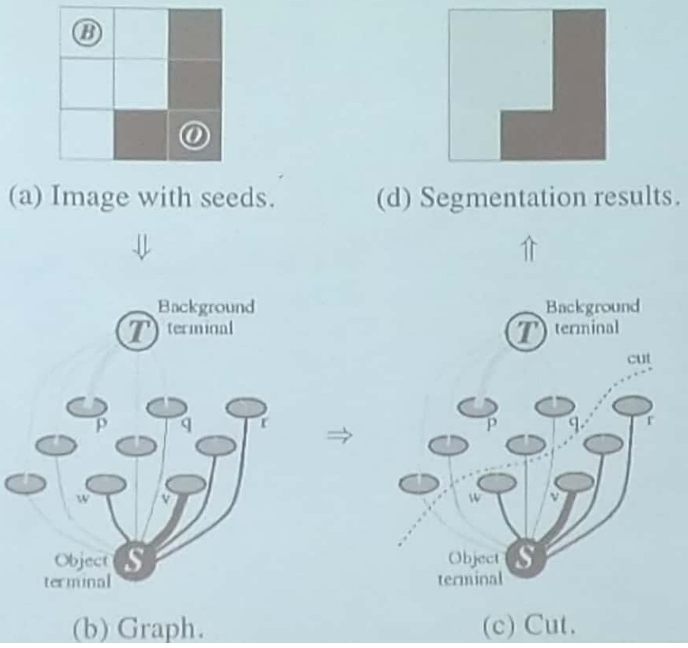
Graph Cut for Image Segmentation - ICCV 2001

User provides a trimap $T = \{ T_F, T_B, T_U \}$ which partitions the image into 3 regions: foreground, background, unknown.



Scanned by CamScanner

Graph Cut for Image Segmentation - ICCV 2001



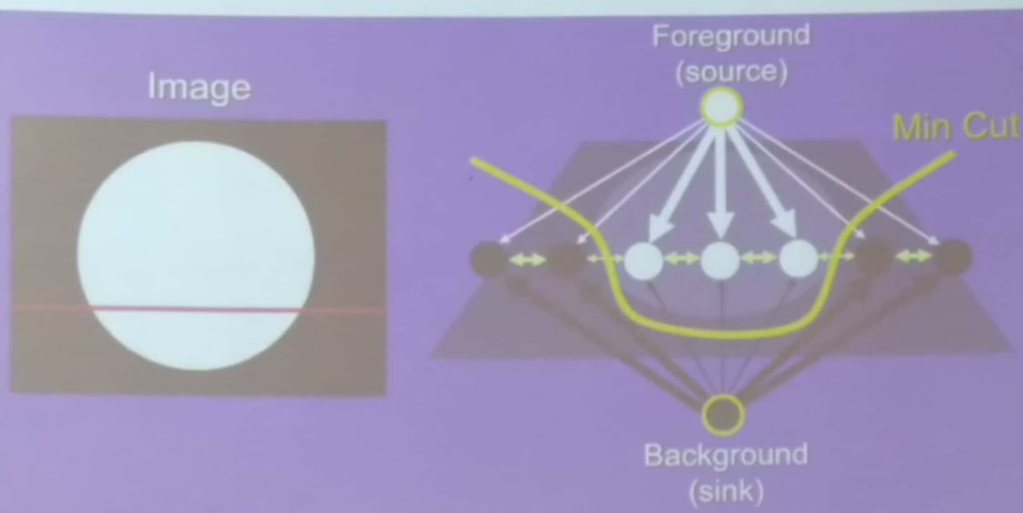
(a) Image with seeds.

(b) Graph.

(c) Cut.

(d) Segmentation results.

Graph Cut for Image Segmentation - ICCV 2001



Cut: separating source and sink; Energy: collection of edges

Min Cut: Global minimal enegry in polynomial time

Graph Cut for Image Segmentation - ICCV 2001

$$E(A) = \lambda \cdot R(A) + B(A)$$

where

$$R(A) = \sum_{p \in P} R_p(A_p)$$

$$B(A) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(A_p, A_q)$$

and

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{otherwise.} \end{cases}$$

Goal: Find Segmentation, A, which minimizes E(A)

- A – Proposed Segmentation
- E(A) – Overall Energy
- R(A) – Degree to which pixels fits model
- B(A) – Degree to which the cuts breaks up similar pixels

aph Cut for Image Segmentation - ICCV 2001

edge	weight (cost)	for
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in \mathcal{N}$
$\{p,S\}$	$\lambda \cdot R_p(\text{"bkg"})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \in \mathcal{O}$
	0	$p \in \mathcal{B}$
$\{p,T\}$	$\lambda \cdot R_p(\text{"obj"})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	0	$p \in \mathcal{O}$
	K	$p \in \mathcal{B}$

- Pixel links based on color/intensity similarities
- Source/Target links based on histogram models of fore/background

$$K = 1 + \max_{p \in \mathcal{P}} \sum_{q:\ \{p,q\} \in \mathcal{N}} B_{\{p,q\}}$$

Graph Cut for Image Segmentation - ICCV 2001

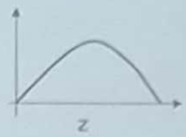
$$
\begin{aligned}
R_p(\text{“obj”}) &= -\ln \Pr(I_p|\mathcal{O}) \\
R_p(\text{“bkg”}) &= -\ln \Pr(I_p|\mathcal{B}).
\end{aligned}
$$

$$B_{\{p,q\}} \propto exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}.$$

Graph Cut for Image Segmentation - ICCV 2001

- The image is an array $z = (z_1, \ldots z_N)$ of grey values indexed by the single index n.
- The segmentation of the image is an alpha-channel, or, a series of opacity values $\alpha=(\alpha_1, \ldots, \alpha_N)$ at each pixel with $0 \leq \alpha_n \leq 1$.
- The parameter θ describes the foreground/background grey-level distributions. i.e. a pair of histogram of gray values:

$$\theta = \{h(z;\alpha), \alpha = 0,1\} \quad \Longleftrightarrow$$



Graph Cut for Image Segmentation - ICCV 2001

$$\mathbf{E}(\alpha,\theta,\mathbf{z}) = \mathbf{U}(\alpha,\theta,\mathbf{z}) + \mathbf{V}(\alpha,\mathbf{z})$$

- U evaluates the fit of the opacity α to the data $\mathbf{z}$
  - i.e. it gives a good score (low score) if α looks like it's consistent with the histogram.

$$U(\alpha,\theta,z) = \sum_n -\log h(z_n;\alpha_n)$$

- V is a smoothness term which penalizes if there is too much disparity between neighboring pixel values.

$$V(\underline{\alpha},\mathbf{z}) = \gamma \sum_{(m,n)\in\mathbf{C}} dis(m,n)^{-1} [\alpha_n \neq \alpha_m] \exp -\beta(z_m - z_n)^2.$$

$$\beta = \left(2\left\langle (z_m - z_n)^2 \right\rangle\right)^{-1}$$

Graph Cut for Image Segmentation - ICCV 2001

$$\mathbf{E}(\boldsymbol{\alpha},\boldsymbol{\theta},\mathbf{z}) = \mathbf{U}(\boldsymbol{\alpha},\boldsymbol{\theta},\mathbf{z}) + \mathbf{V}(\boldsymbol{\alpha},\mathbf{z})$$

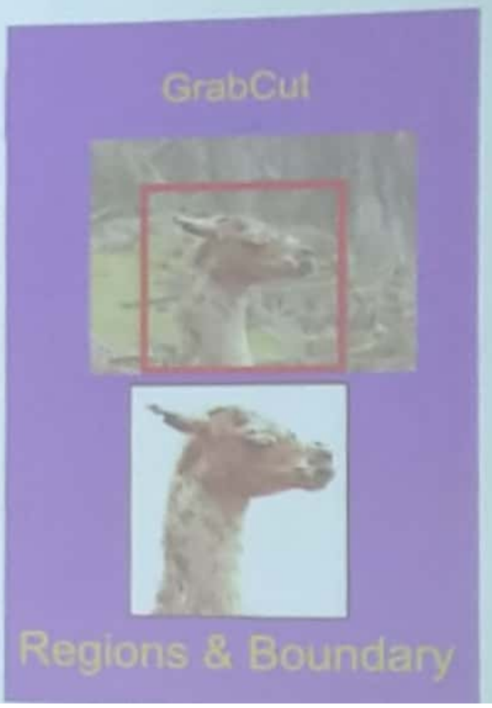
- Given the energy model we can obtain a segmentation by finding

$$\hat{\alpha} = \arg\min_{\alpha} E(\alpha,\theta)$$

- Which can be solved using a minimum cut algorithm which gives you a hard segmentation, $\boldsymbol{\alpha} = \{0,1\}$, of the object.

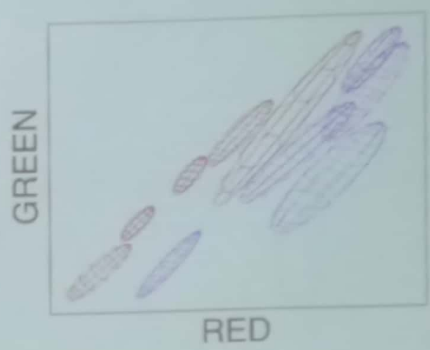
GrabCut - CVPR 2004

- The monochrome image model is replaced for color by a Gaussian Mixture Model (GMM) in place of histograms.
- One shot min-cut solution is replaced by an iterative procedure that alternates between estimation and parameter learning
- Allow for incomplete labeling, i.e. the user need only specify the background trimap T_B (and implicitly the unknown map T_U)



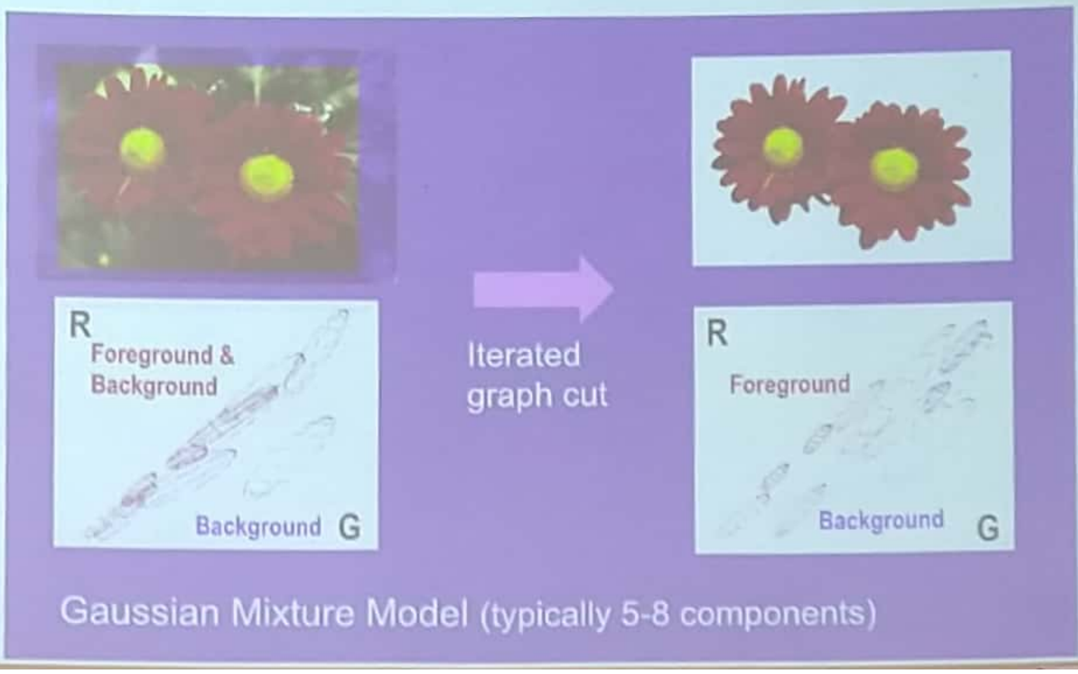
GrabCut - CVPR 2004

- Each pixel z_n is now in RGB color space
- Color space histograms are impractical so we use a Gaussian Mixture Model (GMM)
  - 2 Full-covariance Gaussian mixtures with K components (K ~ 5).
  - One for foreground, one for background.
- Add to our model a vector $\boldsymbol{k} = \{k_1 \ldots k_N\}$, with k_i in $\{1 \ldots K\}$
- k_i assigns the pixel z_i to a unique GMM component (Either from F.G. or B.G. as α dictates)



Scanned by CamScanner

GrabCut - CVPR 2004



GrabCut - CVPR 2004

- Must incorporate k into our model:

$$E(\alpha,k,\theta,z) = U(\alpha,k,\theta,z) + V(\alpha,z)$$

where

$$U(\alpha,k,\theta,z) = \sum_n D(\alpha_n,k_n,\theta,z_n)$$

- $D(\alpha_n,k_n,\theta_n,z_n) = -\log p(z_n \mid \alpha_n,k_n,\theta) - \log \pi(\alpha_n,k_n)$
- Where $\pi(\cdot)$ is a set of mixture weights which satisfy the constraint

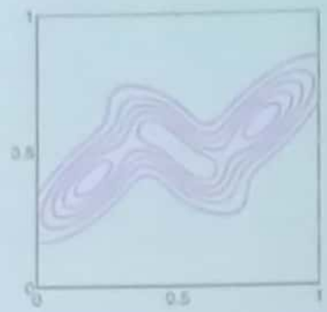
$$D(\alpha_n,k_n,\underline{\theta},z_n) = -\log \pi(\alpha_n,k_n) + \frac{1}{2}\log\det\Sigma(\alpha_n,k_n) + \frac{1}{2}[z_n - \mu(\alpha_n,k_n)]^\top \Sigma(\alpha_n,k_n)^{-1}[z_n - \mu(\alpha_n,k_n)].$$

GMM

$$\mathcal{N}(x \mid \underbrace{\mu}_{\text{mean}},\underbrace{\Sigma}_{\text{covariance}}) = \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right\}$$

$$\ln p(x \mid \mu,\Sigma) = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)$$

$$\ln p(X \mid \mu,\Sigma,\pi) = \sum_{n=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k N(x_n \mid \mu_k,\Sigma_k)\right\}$$



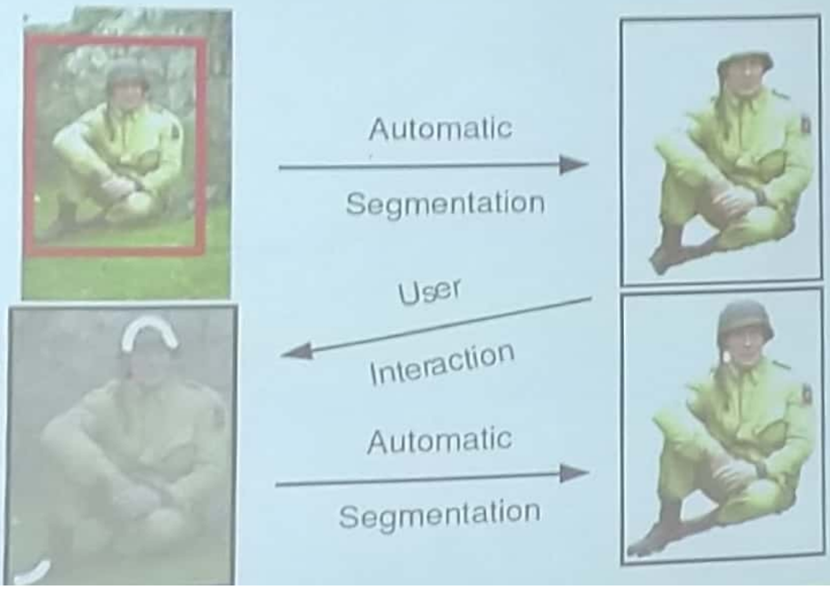
GrabCut - CVPR 2004

- Our θ becomes

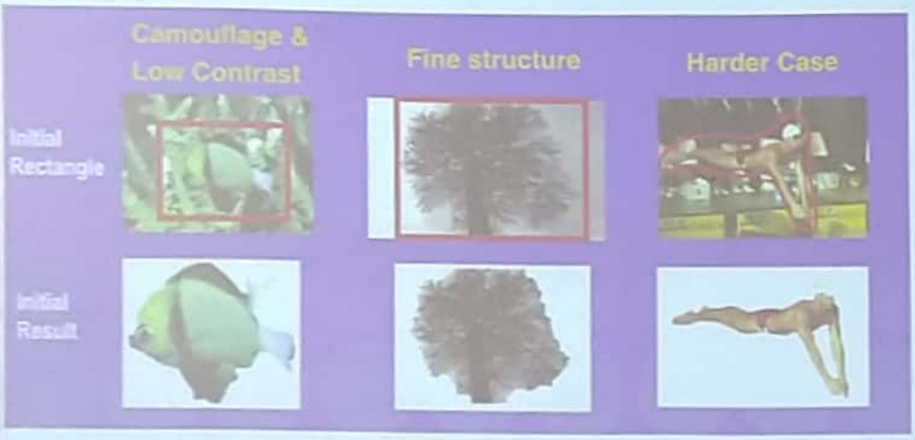
$$\theta=\{\underbrace{\pi(\alpha,k)}_{\text{weights}},\ \underbrace{\mu(\alpha,k)}_{\text{means}},\ \underbrace{\Sigma(\alpha,k)}_{\text{cov.}},\ \underbrace{\alpha=0,1}_{\text{fg/bg.}},\ \underbrace{k=1\ldots K}_{\text{mixture component}}\}$$

- Total of 2K Gaussian components

GrabCut - CVPR 2004



GrabCut - CVPR 2004



Initialisation

- User initialises trimap T by supplying only T_B. The foreground is set to $T_F = \emptyset$; $T_U = \overline{T}_B$, complement of the background.
- Initialise $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$.
- Background and foreground GMMs initialised from sets $\alpha_n = 0$ and $\alpha_n = 1$ respectively.

Iterative minimisation

1. *Assign GMM components to pixels:* for each n in T_U,
$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n).$$
2. *Learn GMM parameters* from data $\mathbf{z}$:
$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$
3. *Estimate segmentation:* use min cut to solve:
$$\min_{\{\alpha_n:\, n \in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$
4. Repeat from step 1, until convergence.
5. Apply border matting (section 4).

User editing

- *Edit:* fix some pixels either to $\alpha_n = 0$ (background brush) or $\alpha_n = 1$ (foreground brush); update trimap T accordingly. Perform step 3 above, just once.
- *Refine operation:* [optional] perform entire iterative minimisation algorithm.

References

[1] Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, pp. 1222-1239, 2001.

[2] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *Proceedings of the Ninth IEEE International Conference on Computer Vision-Volume 2*, 2003, pp. 26.

[3] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 147, 2004.

[4] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *International Journal of Computer Vision,* vol. 70, pp. 109-131, 2006.

[5] C. Rother, V. Kolmogorov and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH 2004 Papers,* 2004, pp. 314.

[6] V. Kolmogorov and C. Rother, "Minimizing nonsubmodular functions with graph cuts-a review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1274-1279, 2007.

[7] Hiroshi Ishikawa, "Transformation of General Binary MRF Minimization to the First Order Case," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 Mar. 2010.