Image Segmentation

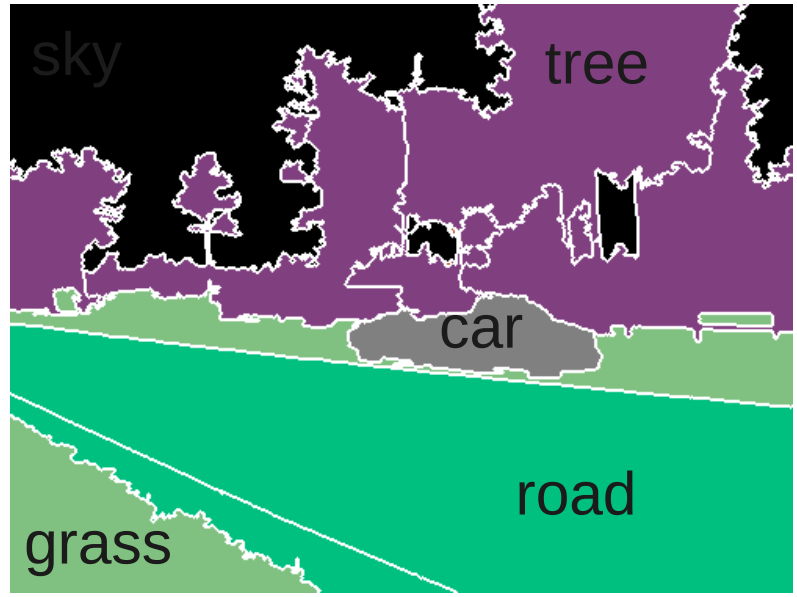
P J Narayanan

CS5765. Computer Vision. Spring 2013

CVIT, IIIT, Hyderabad

What is Segmentation?

- Assigning a category label to each region/pixel



- Is appearance sufficient to make these decisions?
- Is the answer independent of the questioner? Can a pixel be assigned a unique label?
- If you can recognize better, we can segment better. If we can segment better, we can recognize better!

Why is it hard?

- Great variety in appearance, size, shape, etc
- No notion of the inherent **semantics**



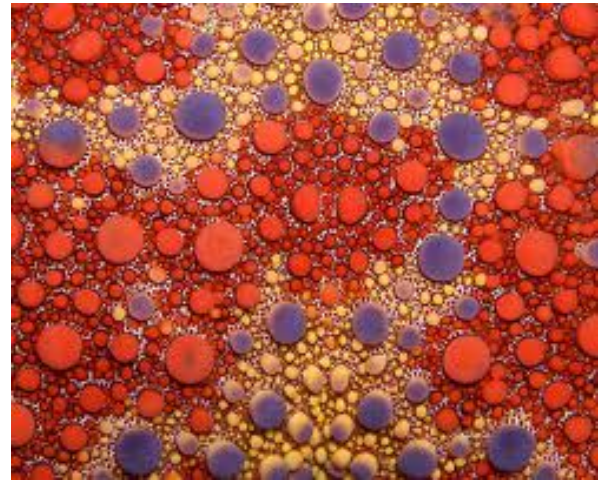
- Classigal, Impressionist, Cubist, ...

Why is it hard?

- Great variety in appearance, size, shape, etc
- No notion of the inherent **semantics**



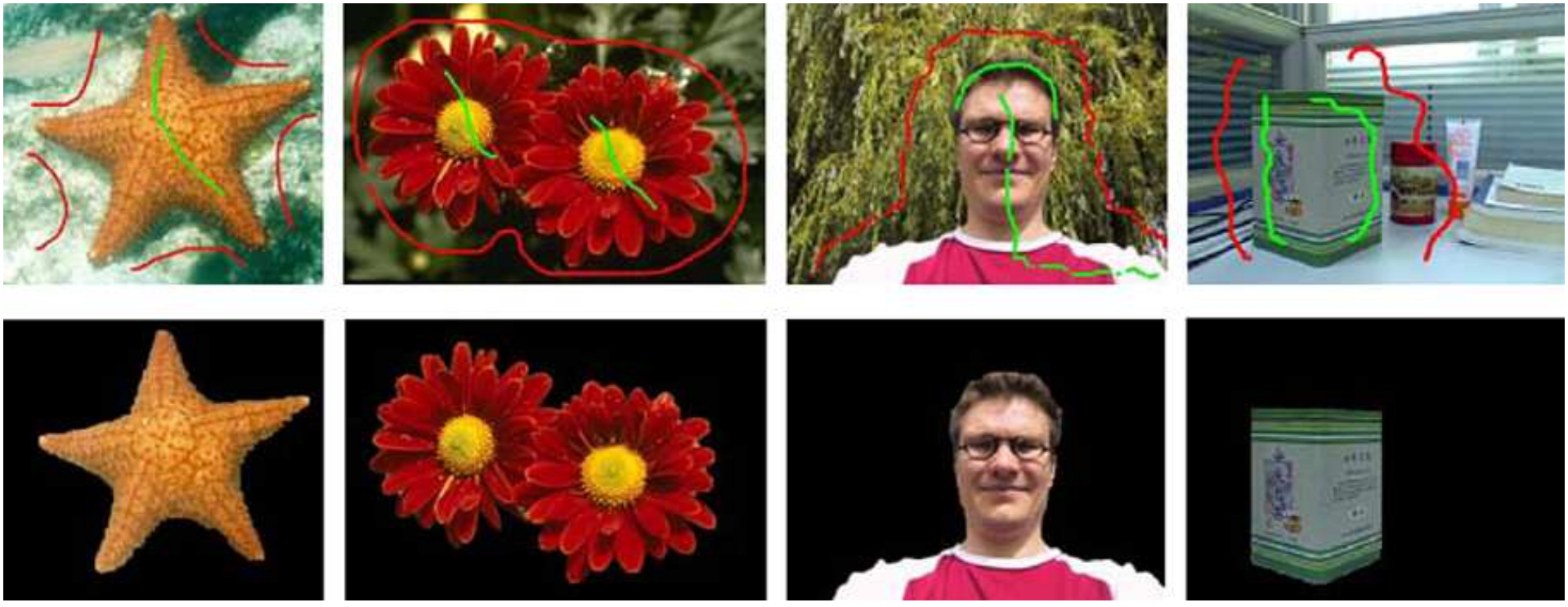




Traditional Methods

- **Thresholding:** Use one or more suitable thresholds to divide the image into “regions”. The histogram can help in finding thresholds
- **Region growing:** Start with a seed pixel/region, grow it in all directions by including pixels that meet the similarity criterion
  - Equivalent to finding the connected component in a graph where two neighboring pixels are connected if “similar”
- **Clustering:** Form vectors of **rgb** values and pixel coordinates **ij** at each pixel. Cluster these vectors into a number of natural units. Trades of RGB similarity with pixel proximity

Background Subtraction



- Goal: Extract the “main” or **foreground** object
- Also, called **foreground-background separation**

Background Subtraction

- Surveillance camera, wants to find when something is happening
- Fixed background, subtract it pixel-by-pixel from current frame!
- Build a model for the background colour of each pixel. When the present value differs from it, it is a foreground pixel!
- What kind of a model?
  - Constant color?
    Slow changes (i.e., average over past k frames)?
  - A Gaussian with a mean and variance?
    Adjust parameters continuously?
  - A model for each pixel? For the whole frame?

Gaussian Mixture Model

- Background and foreground are not of uniform colour or a single Gaussian in colour space
- Model each as a mixture of a few Gaussians, with weights w_i for each

$$p_m(\mathbf{x}) = \sum_{i=1}^{n_m} w_{im} G(\mathbf{x}|\mu_{im}, \sigma_{im}), \quad m = \text{fg, bg}$$

- A Gaussian each for foreground and background
- Given a pixel colour x, evaluate $p(\mathbf{x})$ for foreground and background. Choose the model with greater probability
- The models can adapt slowly by introducing new Gaussians in place of ones that are “obsolete”. Expectation maximization framework to modify weights and Gaussian parameters

MAP, MRF, Energy Minimization, and Graph Cuts for Image Segmentation

Markov Random Field

- Several computer vision problems like image segmentation, stereo matching, etc., are *labelling* problems
- These can be formulated as **maximum a posteriori** estimation in a Bayesian framework
- The formulation is equivalent to minimizing an energy defined over a Markov Random Field
- MRF has stations defined for each pixel and connectivity to other pixels
- Each pixel has a label value which can’t be observed and other properties which can be observed
- Markov property: The value at one pixel is determined by a (small and finite) neighbourhood only.

Posterior Probability and Energy

- Given an observation $\mathbf{z}$ and an event x, Bayes theorem relates posterior and prior probabilities:

$$p(x|\mathbf{z}) = p(\mathbf{z}|x)\ p(x)\ /\ \ (\Sigma\ p(z|x))$$

- Given observations $\mathbf{z}_i$ at each pixel, the best label x_i has the highest $p(x_i|\mathbf{z}_i)$, which corresponds to highest $p(\mathbf{z}_i|x_i)\ p(x_i)$.
- Taking logarithm and changing sign, assign a label

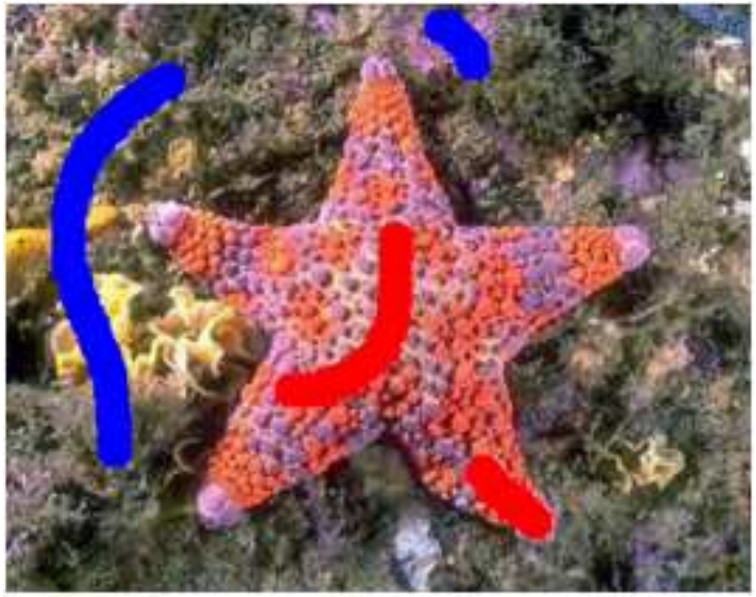
$$x^* = \arg\min_x \ \ -\log p(\mathbf{z}|x) - \log p(x)$$

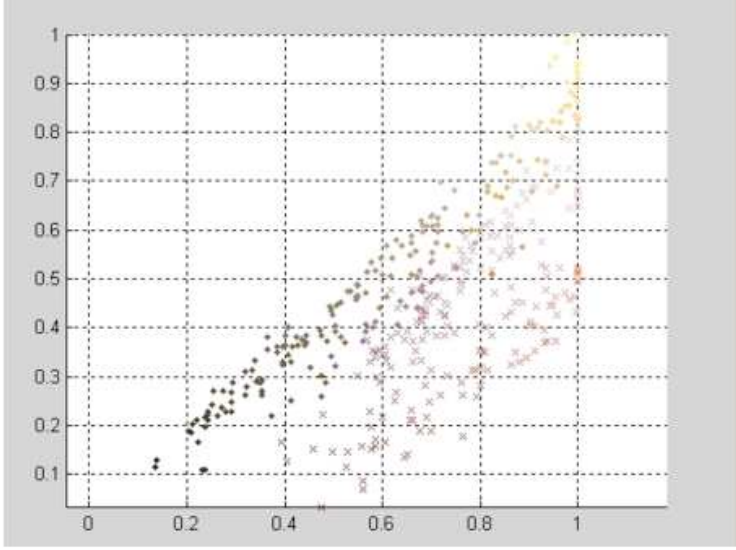
- MRF probabilities can be modelled as Gibbs function or $p(x) = e^{-E(x)}$, where $E(x_i) = \sum_{j \in \mathcal{N}} V(x_i, x_j)$, where $\mathcal{N}$ is a neighbourhood of i.

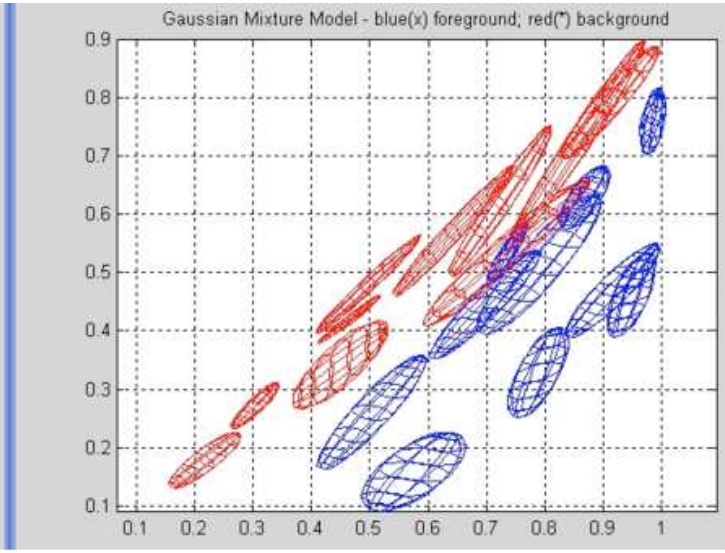
Energy Minimization

- Solution: $\arg\min_x E = E_d(\mathbf{z}, x) + E_p(x)$
- $E_d(x, y)$ is the **data term** of energy which measures how well the measurement fits the model for event x. Example: Given the distribution of colours of mangoes, what is the probability of the given colour?
- $E_p(x)$ is the **prior term** of energy which gives the chance of finding model x there. In an MRF, this depends on the labels in a small neighbourhood only
- E_p is the *clique* potential as it depends on cliques of different sizes in the neighbourhood
- Clique of order 2 is most common, resulting in binary potentials. This is the penalty of differing neighbours

Example: Input, Output, GMM



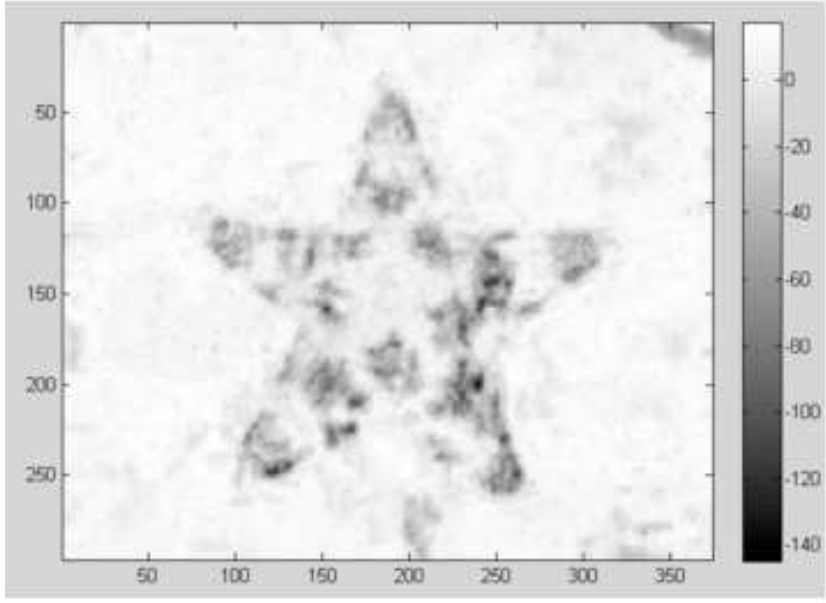




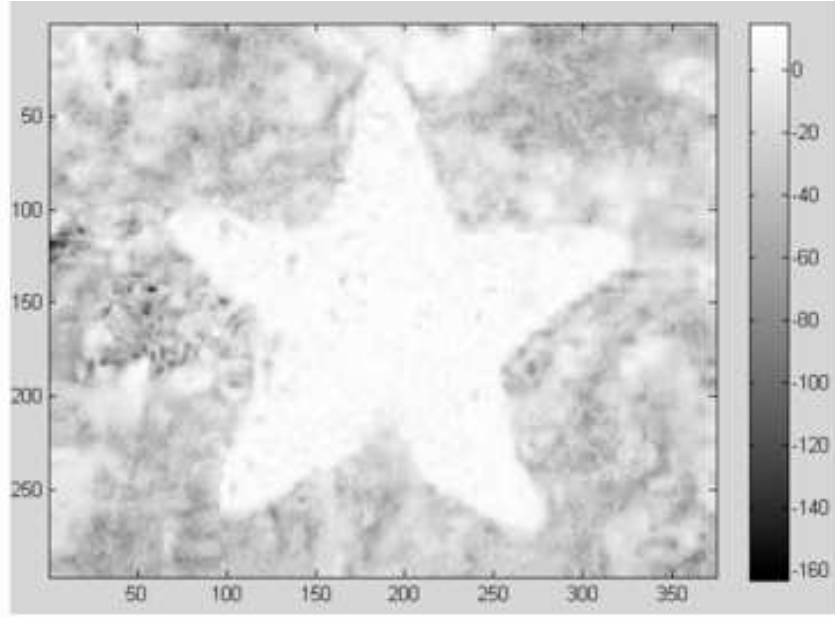
Background & Foreground

Likelihoods $\log p(\mathbf{z}|x)$ for:

$x = 0$ (background)



$x = 1$ (foreground)



Are likelihoods sufficient to assign correct label?

Max Likelihood Decision

Use maximum of likelihoods to assign the label:

Decision is not satisfactory as no neighbourhood information is used.

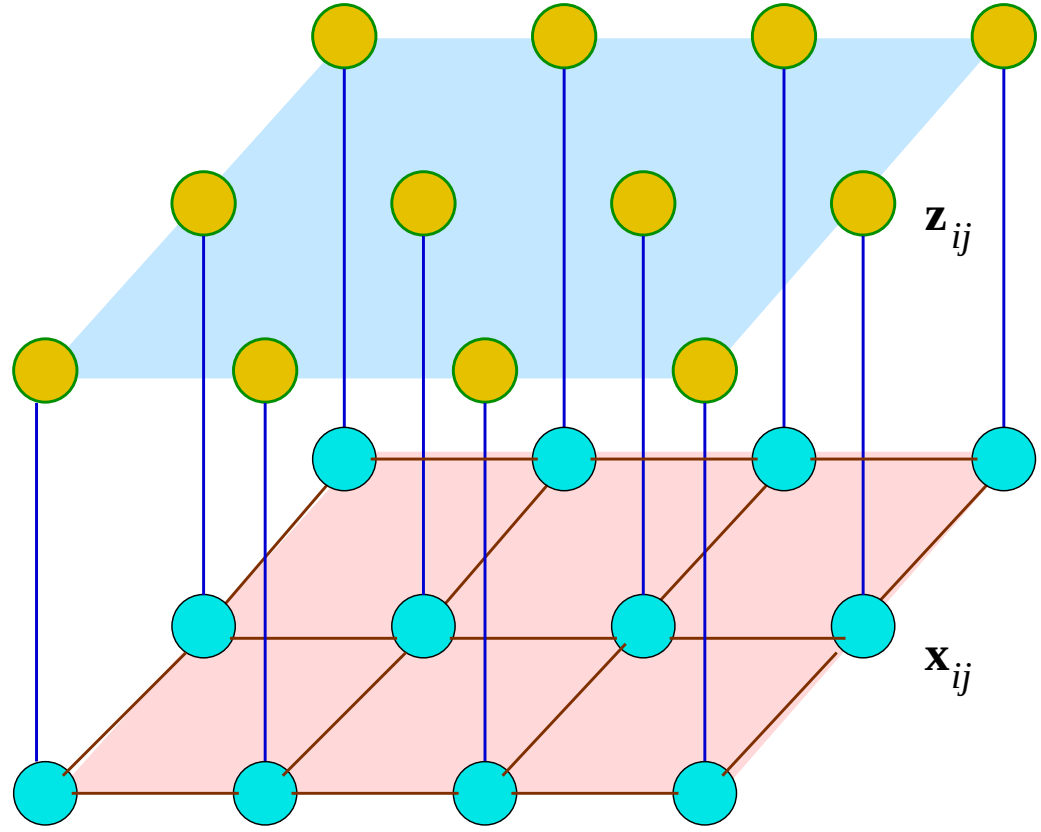
Maximum Posterior Decision

Using maximum a posteriori (MAP) probabiliies:

Can change the weights of likelihood (E_d) and prior (E_p) terms to get different results.

MRF Structure

- Observed values $\mathbf{z}_{ij}$ and hidden labels x_{ij} at each pixel
- Labels depend on observations and labels in a small neighbourhood
- Encodes the relationship:



$E(x, z) = E_d(x, z) + E_p(x)$ where: E_d is the data term of the energy and E_p is the prior or smoothness term

- We seek the solution $x^* = \underset{x}{\operatorname{argmin}}\, E(x)$

MRF Energy

- We seek maximum of $p(x|\mathbf{z})$ over the whole image, which is given by the terms for each pixel due to their independence

$$p(x|\mathbf{z}) = p(\mathbf{z}|x)\ p(x) = \left[\prod_{i,j} p(\mathbf{z}_{ij}|x_{ij})\right] \left[\prod_{i,j} p(x_{ij})\right]$$

- Translates to minimizing energy with a weight factor w:

$$E(x, \mathbf{z}, w) = \sum_{i,j} \theta_{ij}(x_{ij}, \mathbf{z}_{ij}) + w \sum_{i,j,k,l} \theta_{ijkl}(x_{ij}, x_{kl}, \mathbf{z}_{ij}, \mathbf{z}_{kl})$$

- Energy is the sum of data terms and smoothness terms from each pixel. Prior can be weighted differently
- How do we minimize the energy? Graph Cuts!

Graphs and Cuts

- A graph $G = (V, E)$ consists of vertices/nodes V and edges in E connecting them. An edge e_{ij} has a weight w_{ij}. Edges are directed, in general
- A **cut** $C(s, t)$ for $s, t \in V$ is a set $E' \subset E$ of edges which separate (or disconnect) nodes s and t when removed
- Cut value: sum of weights removed $= \sum_{e_{ij} \in E'} w_{ij}$
- A **min cut** $C^*(s, t)$ is a cut with minimum value (the least sum of weights of edges in E', among all possible cuts)

Goal: **Create a graph such that value of the min cut corresponds to the minimum energy of a given MRF!!**

Graph Representable Energy

- Total energy: $E(f) = \sum_p D_p(f_p) + \sum_p \sum_{q \in \mathcal{N}_p} V_{pq}(f_p, f_q)$

  At a pixel p: f is the label, D the dataterm, V the smoothness term and $\mathcal{N}$ the neighbourhood.

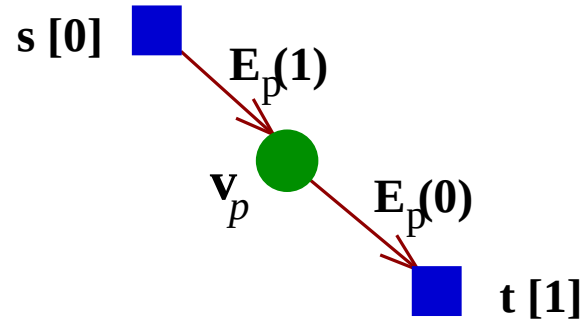
- Binary labelling: assign a label $f_p \in \{0, 1\}$ to pixel p

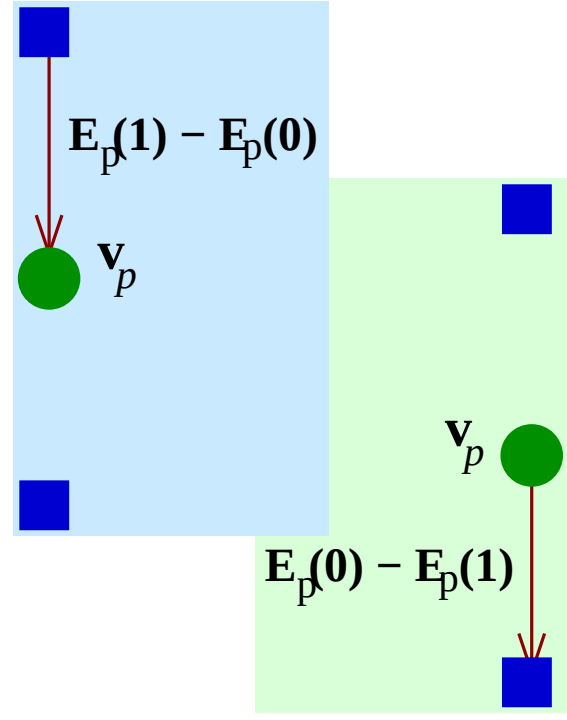
- A function E of n binary variables is *graph representable* if a graph $G = (V, E)$ with teminal nodes s, t can be found such that for any configuration of $x_1, \cdots, x_n$, the value of E equals a constant plus the value of min cut $C(s, t)$

- Proof by construction. We will construct a graph such that its min *s*-*t* cut value is same as the energ $E(f)$. We restrict to 2-cliques or pair-wise potentials in V

Graph Construction: Unary Term

- The graph has n nodes for each x_i and two terminal nodes s and t.
- The unary term is a function of 1 variable.
- For unary term $D_p(f_p)$, if $E_p(0) > E_p(1)$, add and edge from node v_p to t with weight $E_p(0) - E_p(1)$. Otherwise add and edge from node s to v_p with weight $E_p(1) - E_p(0)$
- Mincut value equals the energy (plus a constant)





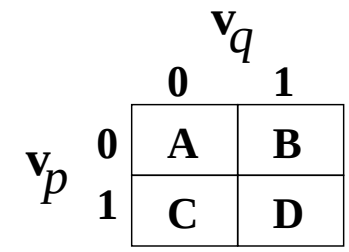
Graph Construction

- If $E_p(0) < E_p(1)$, v_p should get label 0, or the edge to t should be in the cut. Else v_p should get label 1, or the edge from s should be in the cut
- Since the graph is a simple one, the cut will be the edge with lower weight. Thus, connect v_p from s with weight $E_p(1)$ and to t with weight $E_p(0)$!
- A simpler graph will have a single edge of weight equal to the positive difference, from s or to t

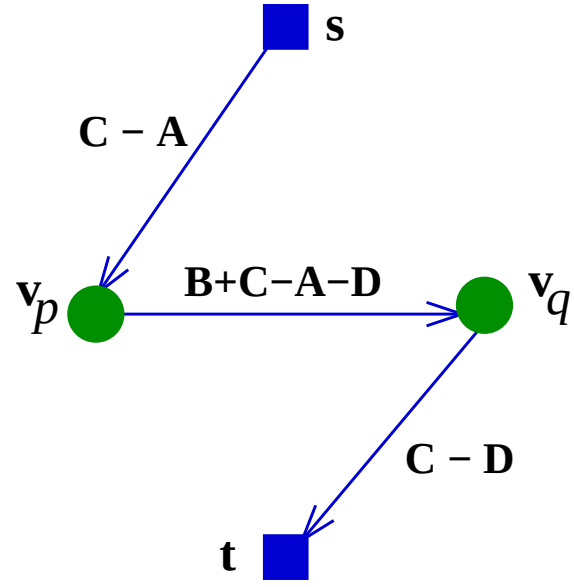
- First term has wt 0 to t and $C - A$ from s for v_p. Second has 0 to t and $D - C$ from s for v_q
- Third term gives the energy when v_p has label 0 and v_q label 1. Connect it as an edge from v_p to v_q.

Graph Construction: Binary Term

- 4 combinations for 2 nodes.
- Energies are A, B, C, D. Split into 4 combinations. One is a constant and won't affect
- Second is a unary function on v_p. Add edges with weight C from s and weight A to t
- Third is also a unary function on v_q. Add edges with weight C to t and D from s
- Fourth is an n-edge with an edge from v_p to v_q.
- Mincut value equals the energy (plus const)



0	0	+	0	D−C	+	0	B+C−A−D
C−A	C−A		0	D−C		0	0



The 4 Cases

- A is minimum: Mincut is edge to t as $C - A > C - D$ and $(B - A) + (C - D) > C - D$. Labels $\mathbf{0}, \mathbf{0}$
- D is minimum: Mincut is edge from s as $C - D > C - A$ and $(B - D) + (C - A) > (C - A)$. Labels $\mathbf{1}, \mathbf{1}$
- B is minimum: Mincut is the n-edge as $C - D > (B - A) + (C - D)$ and $C - A > (B - D) + (C - A)$. Labels $\mathbf{0}, \mathbf{1}$
- C is minimum: $A - C$ edge from v_p to t and $D - C$ edge from s to v_q. No path from s to t. Labels $\mathbf{1}, \mathbf{0}$
- This is true only if $(B + C - A - D) > 0$, a condition knowns as **regularity** or **submodularity**
- Sounds reasonable as cost of different labels is expected to be greater than cost of same label

Additivity Property

- If two functions f_1 and f_2 are graph representable, the function $f = f_1 + f_2$ is also graph representable and the mincut property holds
- In our case, the nodes are the same. Edges are the union of the two graphs. Weights add up on edges between common nodes
- Thus, all submodular energy functions can be represented using a graph, whose mincut corresponds to the minimum energy configuration!!

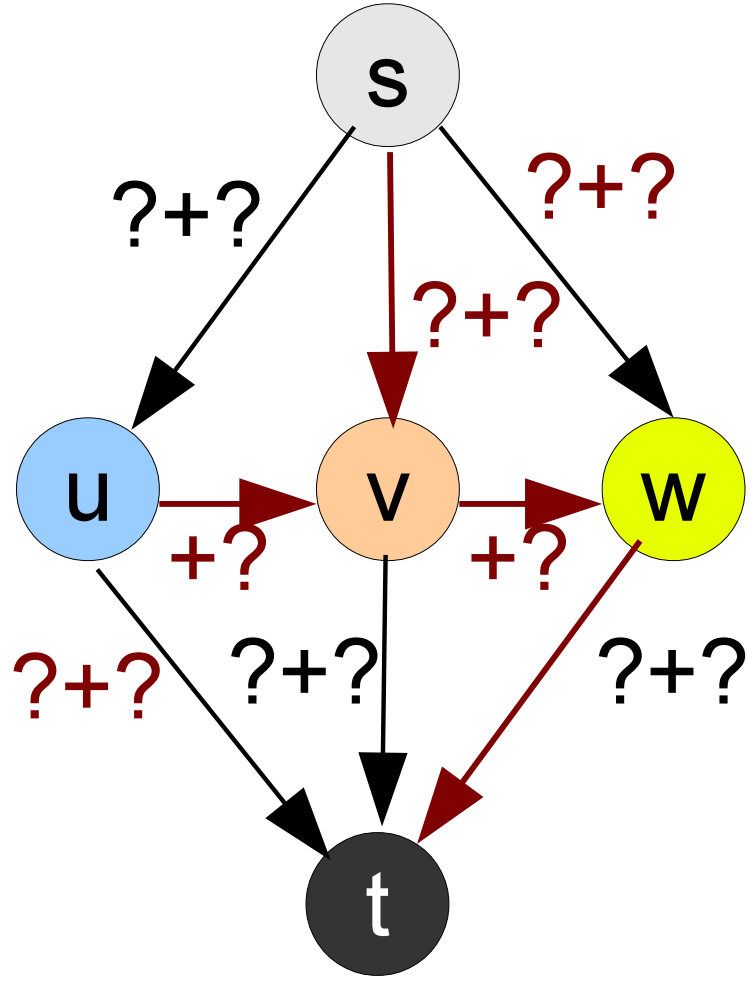
An Example

An MRF with 3 nodes u, v, w have unary energies $u_0 = 2, u_1 = 5$, $v_0 = 6, v_1 = 0$, $w_0 = 4, w_1 = 2$

u is a neighbor of v and v is a neighbour of w

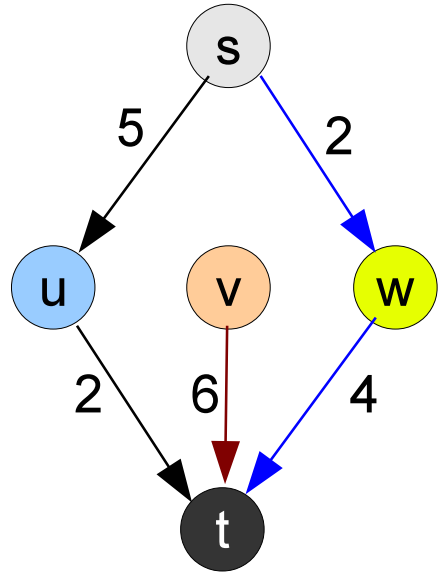
Binary energy between a pair is 0 if they have the same label and 2 if they have different labels

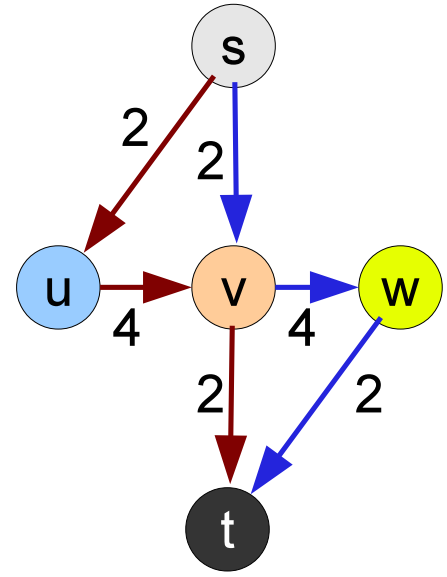
Graph construction? Minimum cut? Labels for each node?



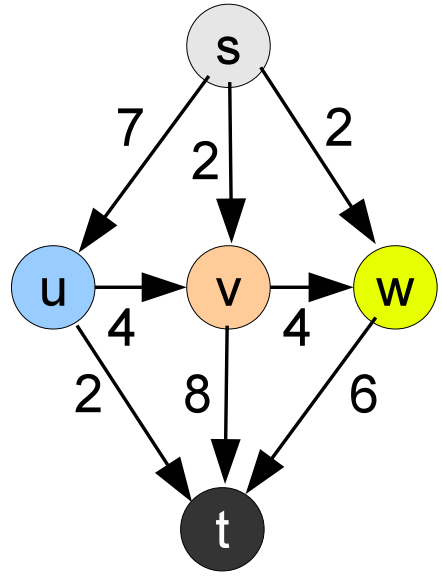
Graph Construction 1

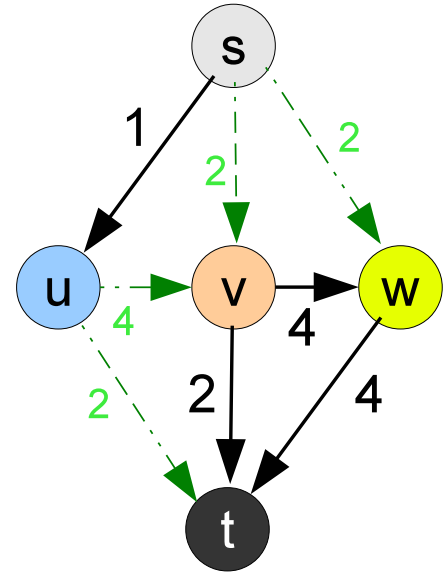
Unary and Binary Graph components:





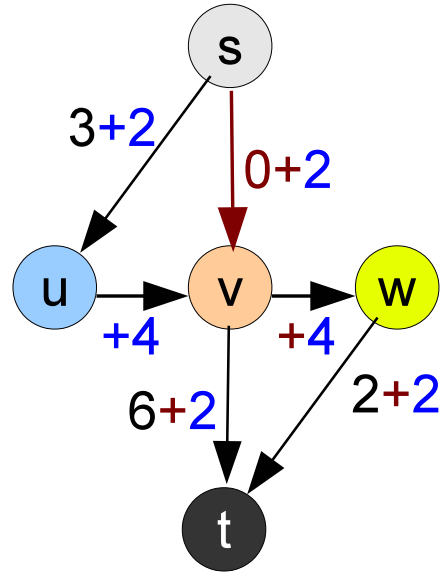
Final Graph and its Mincut:

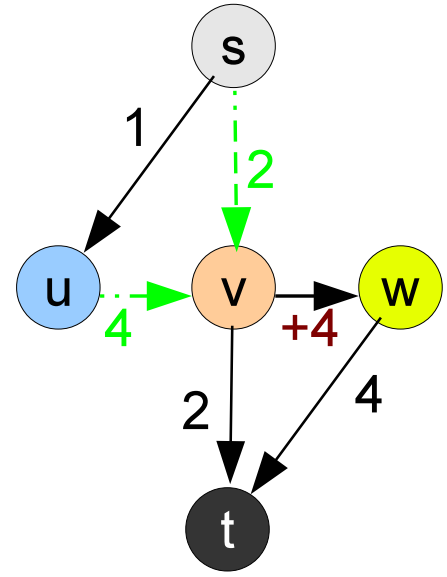




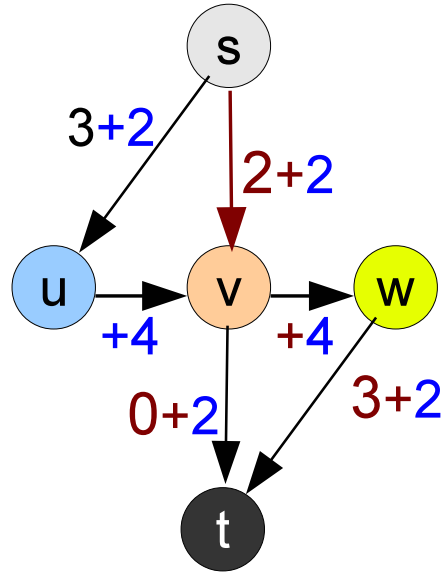
Examples ... contd

Alternate construction and its Mincut:





A different graph and its Mincut:



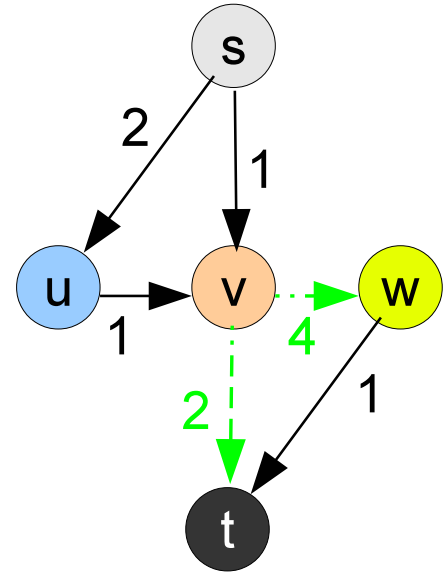


Image Segmentation

- Energy: $E(f) = \sum_p D_p(f_p) + \sum_p \sum_{q \in \mathcal{N}_p} V_{pq}(f_p, f_q)$
- Data term D comes from the GMMs for foreground and background. Find $p(z_p|f_p)$ where z_p is the colour and f_p the current label. Take its logarithm and change sign.

  Smoothness term, Part 1: $V(f_p, f_q) = 0$ if $f_p = f_q$

  Part 2:

  Linear cost: $V(f_p, f_q) = k\,|f_p - f_q|$

  Clamping: $V(f_p, f_q) = k\,\min(m, |f_p - f_q|)$

  Potts model: $V(f_p, f_q) = k$ if $f_p \neq f_q, 0$ otherwise

  Contrast sensitivity: $k = e^{-\beta|z_p - z_q|}$
- Discontinuity preservation: We want to promote continuity but not prohibit discontinuity. Linear cost is NOT discontinuity preserving.

Graph Mincut

- **maxflow mincut theorem**: Mincut of a graph can be computed using the maxflow algorithm, where s is an infinite source and t and infinite sink. Directed edge of weight w can allow w units to flow in its direction. (From 1956)
- Maxflow algorithms: Many.
- **Ford-Fulkerson Augmenting Path** algorithm and **Goldberg-Tarjan Push Relabel** algorithm are popular

Ford-Fulkerson Algorithm

- Find a path from s to t with possible flow. Use BFS for the same
- Possible flow is the minimum residual capacity of all edges in the path
- Push that much flow from s to t. Subtract the flow from the residual flow at each edge involved. At least one edge will saturate
- Repeat until no more path can be found
- The collection of saturated edges is the the cut that separates s from t

A sequential algorithm, but very efficient in practice. Also known as the *augmenting path* algorithm

Goldberg-Tarjan Algorithm

- Nodes can temporarily have excess flow or storage
- Each node has a height value, equal to the distance from the sink
- Each node pushes its excess flow through each outgoing edge to a node that is 1 level lower in height, subject to its residual capacity
- Update excess flow at each node and capacities of each edge
- If no flow is possible, relabel nodes by recalculating heights. Repeat pushing
- When no relabel or push is possible, terminate. Edges that are saturated (i.e., have 0 capacity) form the cut

A parallelizable algorith, also known as the *push relabel* algorithm.

Multilabel Optimization

- Can create a graph with k terminal nodes. Minimum of the energy corresponds to the multiway cut of the graph
- Multiway cut is an NP Hard problem with no easy solutions
- Approximate solutions: Using **move making** algorithms
- α-expansion: Some pixels change their label to α. Other pixels retain old labels. Cycle through all labels until convergence
- $\alpha\beta$-swap: Consider if any pixel can change label from α to β or vice versa. Other pixels retain old labels. Cycle through all pairs until convergence

α-Expansion Algorithm

Assign arbitrary labels to pixels

Repeat until convergence // Perform cycles

  For label $\alpha = 1$ to *maxL* // Perform iterations

    Construct a graph with α versus rest

    Perform a 2-level graph cut to label pixels α

    If energy reduces, keep the new configuration

Performs l bilevel optimizations per cycle

$\alpha\beta$-Sweep Algorithm

Assign arbitrary labels to pixels

Repeat until convergence // Perform cycles

  For label pairs α, β // Perform iterations

    Construct a graph with α versus β

    Perform a 2-level graph cut to change or retain label

    If energy reduces, keep the new configuration

α-expansion is preferred in most situation as it works better

Stereo Matching

- Data term is based on the goodness of match for current disparity f_p
- This can be evaluated for the left pixel by shifting it by f_p and evaluating a match using SAD, SSD, NC, etc.
- A symmetric cost can be taken considering match of left with right and right with left
- V is based on smoothness.
- Potts model: $V(f_p, f_q) = k$ if $f_p \neq f_q$
- Static cues: Contrast sensitive value for the constant k. Cost is high to assign different labels to neighbouring pixels of similar colour. Cost is moderate to do so if colours are different!

Example: Tsukuba Image

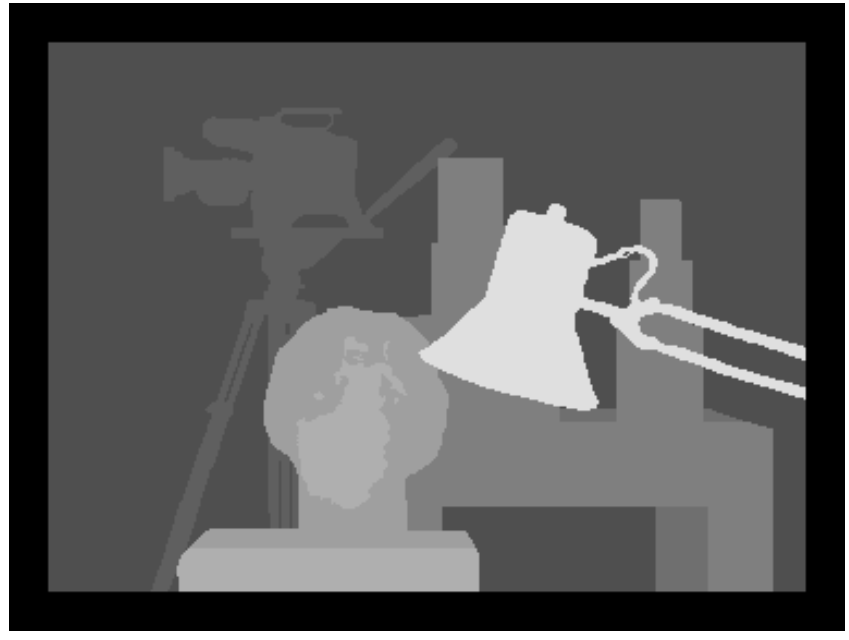


Image Denoising

- Data term: Difference between observation and current label
- $D(f_p) = |f_p - I_p|^2$
- Smoothness term: Clamped model or Potts model
- Linear model results in oversmoothing

Segmentation: Other Methods

- **Mean-Shift Segmentation:** Generalization of GMMs to include non-parametric distributions of appearance. Learn the distribution and move to its nearest maximum
- **Pixel Affinities:** Define *affinity* of a pixel to another one based on a suitable similarity measure. For N pixels, define an $N \times N$ *affinity matrix* that lists all pixel affinities. The eigenvectors of this matrix gives natural grouping or clustering or segmentation of the image. **Normalized Cuts** is a good method in this class.
- **Tracking:** Track or segment an object across time given a video. This is a rich problem with a lot of literature to support it. Important problem in video procesing.

What is Computer Vision?

Computer processing of visual inputs: images and videos.

Making sense out of them. Describing them.

Does computer vision mimic the human vision?
– Certainly in many of its goals
– Why? Human vision is among the best!
– Sophisticated and efficient but not understood well

Do we process visual inputs how humans do? Not necessarily, though we try to draw inspiration from it as often as is convenient!!

Human visual system needn’t limit a computer vision system!

Three "Urges" on Seeing a Picture

1. **To group** proximate and similar parts of the image into meaningful "regions".

   Called **segmentation** in computer vision.

2. **To connect to memory** to recollect previously seen objects.

   Called **recognition** in computer vision.

3. **To measure** quantitative aspects such as number and sizes of objects, distances to/between them, etc.

   Called **reconstruction** in computer vision.

Jitendra Malik, Mysore Park, December 2011

Computer Vision

- An area with some past and a bright future
- Interdisciplinary: Signal Processing, Algorithmics, Machine Learning, Geometry, Perception, etc.
- Several applications have been developed already, but ...
- Sky is really the limit as vision is **very** important to humans

Thank You!

Many figures are from Web resources and from

MRF Optimization Tutorial

by **Blake, Rother, Kohli, and Kumar**

Thanks also due to Computer Vision researchers worldwide for slides, images, and other web resources