Announcements

- Readings for today:
  - *Markov Random Field Modeling in Computer Vision.* Li. First two chapters on reserve.
  - ``Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images,'' Geman and Geman. On reserve.
  - ``Fast Approximate Energy Minimization via Graph Cuts'', by Boykov, Veksler, and Zabih.

Markov Random Fields

- Markov chains have 1D structure
  - At every time, there is one state.
  - This enabled use of dynamic programming.
- Markov Random Fields break this 1D structure.
  - Field of sites, each of which has a label, simultaneously.
  - Label at one site dependent on others, no 1D structure to dependencies.
  - This means no optimal, efficient algorithms.

Definitions

- S indexes a discrete set of sites.
  - S = {1, …, m}
  - S = {(i,j) | $1 <= i, j <= n$} for nxn grid.
- L_d = discrete set of labels, eg. {1, … M}.
  - Labels could be continuous, but we skip that.
- A *labeling* assigns a label to every site,

$f = \{f_1, \ldots f_m\}$. f_i is the label of site *i.*

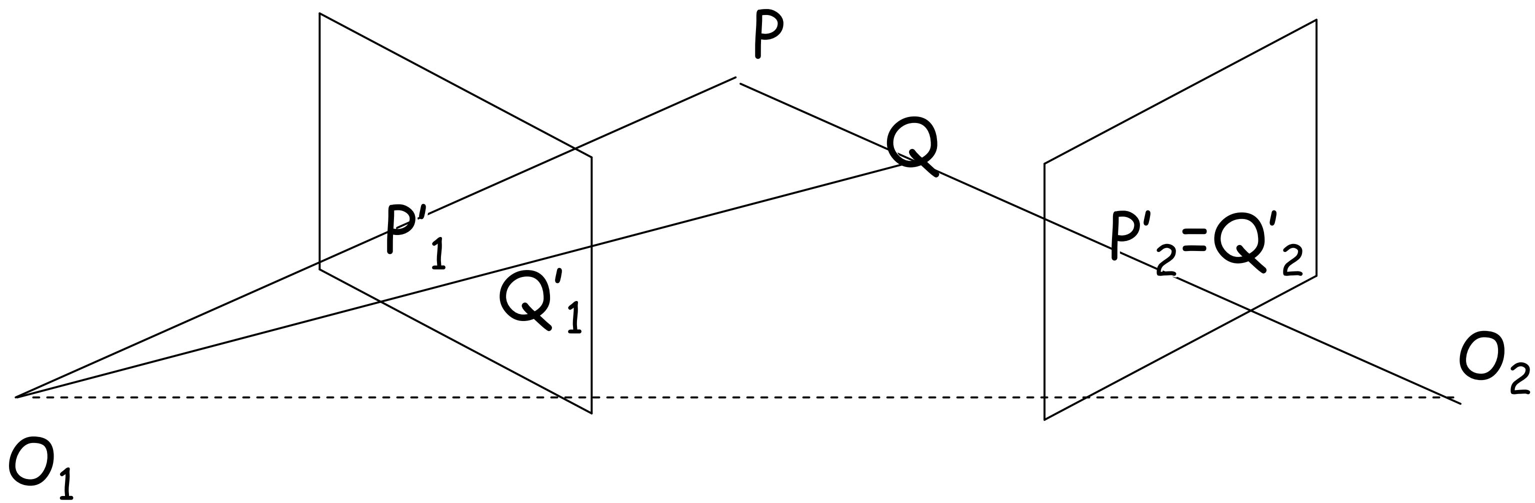
Neighborhoods

- Neighborhood specifies dependencies.
  - $N = \{N_i \mid \text{for all } i \text{ in } S\}$
  - N_i is neighborhood of *i*. *j* in N_i means *i* and *j* are neighbors.
    - A site is not its own neighbor.
    - Neighborhood is symmetric.
- Neighborhood -> conditional indep.
  - *F* is an MRF on *S* w.r.t. *N* iff:
    - $P(f) > 0$
    - $P(f_i \mid f_{S-\{i\}}) = P(f_i \mid f_{N_i})$

Example: Image Segmentation

- Each segment has a constant property corrupted by i.i.d. noise
- Every pixel is a site.
- Label is intensity, uncorrupted by noise.
- Label depends on *observation;* pixel corrupted by noise.
- Also depends on other labels.
  - If you see an image with one pixel missing, you can guess value of missing pixel pretty well.

Example: Stereo

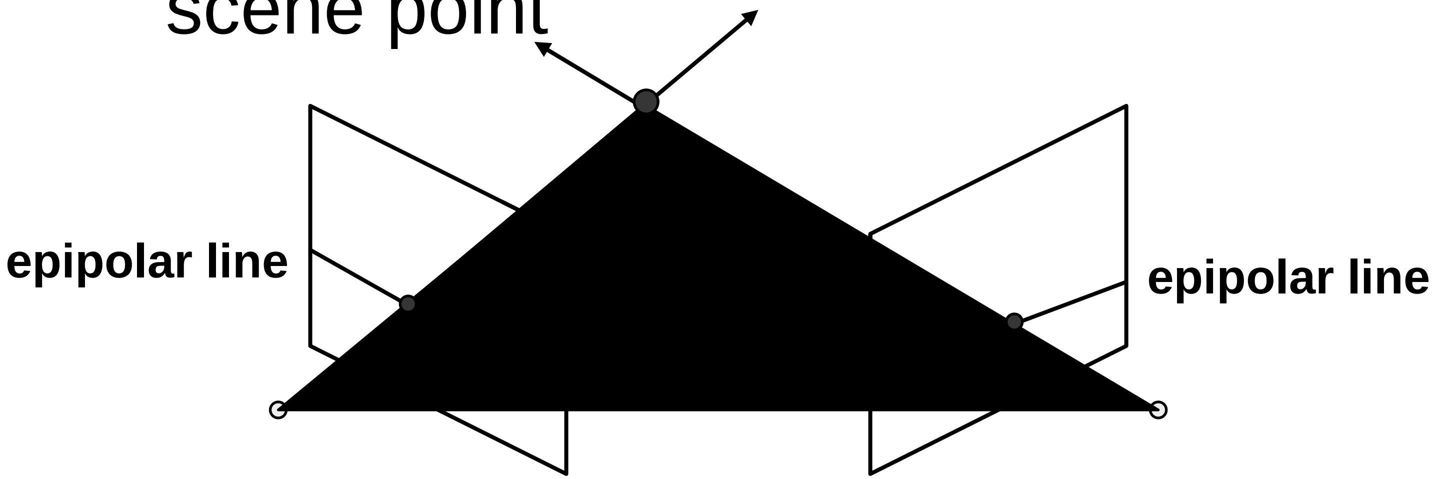


Depth can be recovered with two images and triangulation.

(Camps)

Stereo correspondence

- Determine Pixel Correspondence
  - Pairs of points that correspond to same scene point



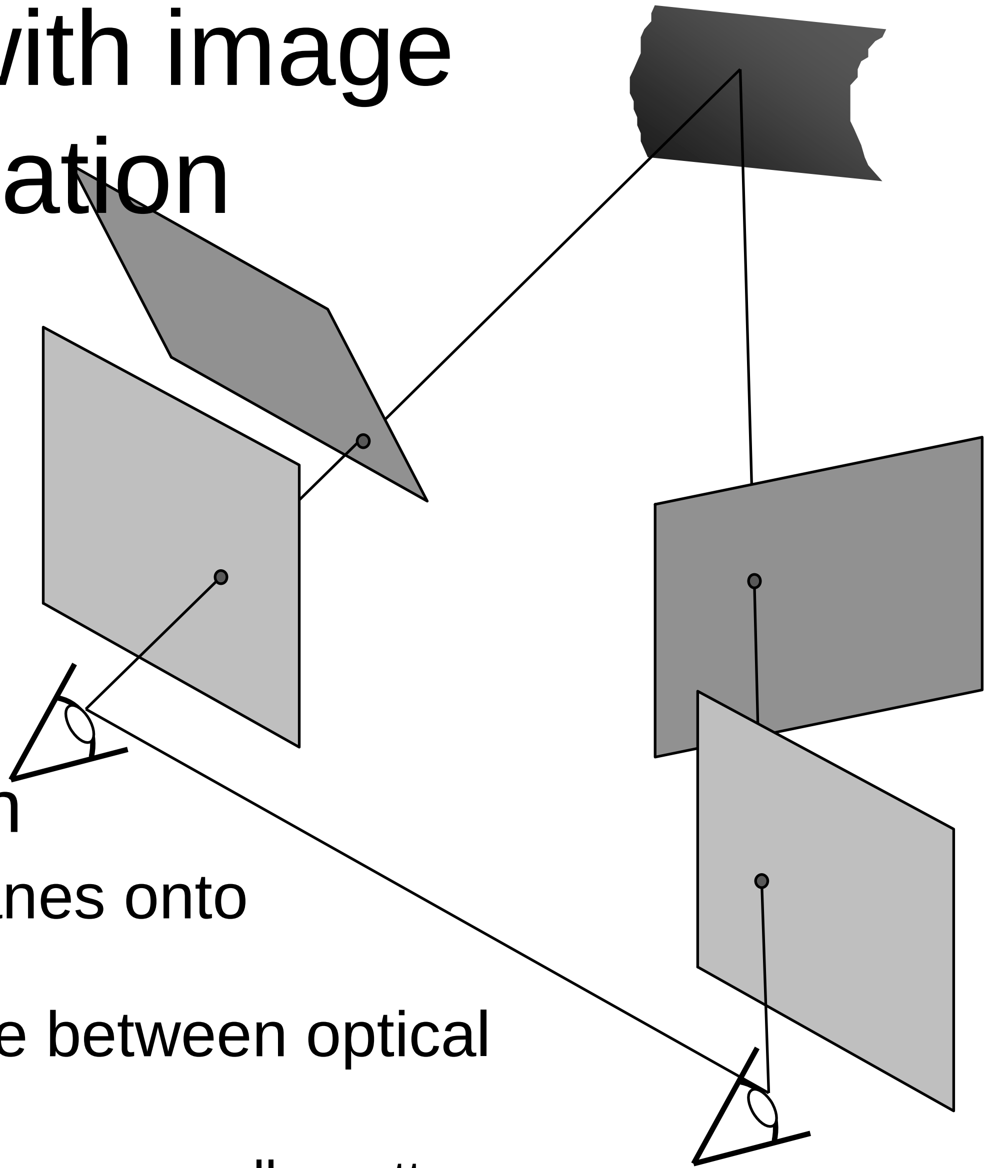
- Epipolar Constraint
  - Reduces correspondence problem to 1D search along *conjugate epipolar lines*

(Seitz)

Simplest Case

- Image planes of cameras are parallel.
- Focal points are at same height.
- Focal lengths same.
- Then, epipolar lines are horizontal scan lines.

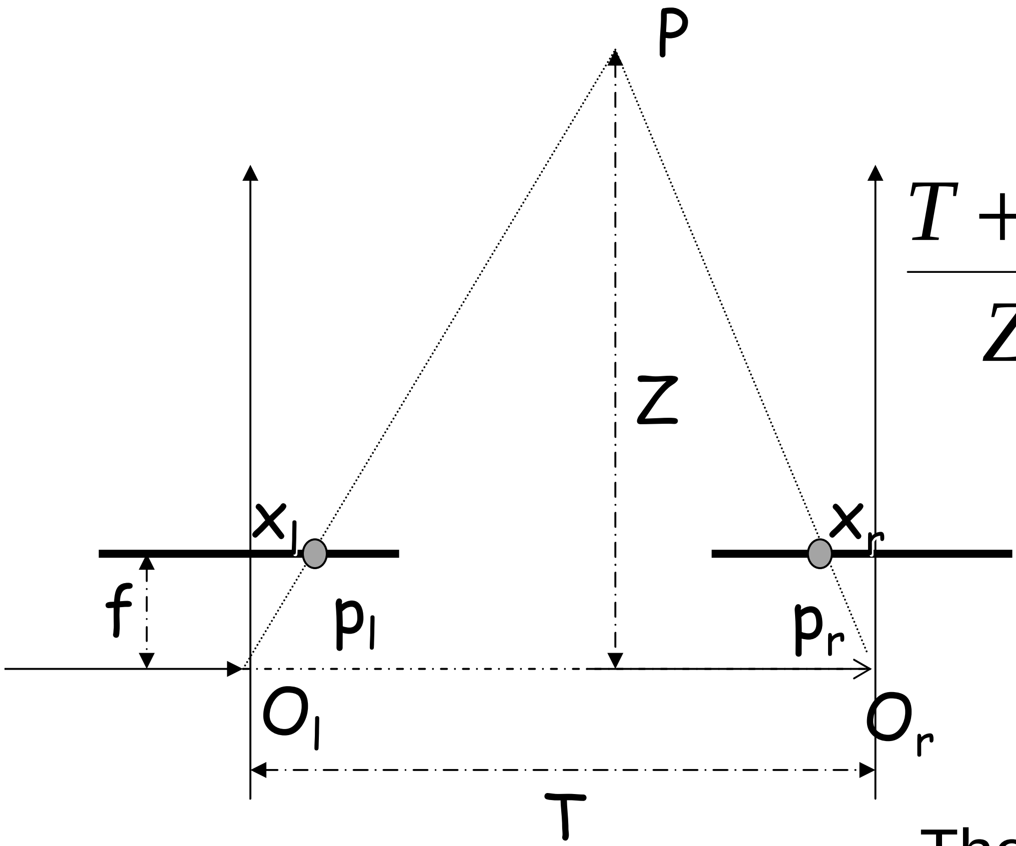
We can always achieve this geometry with image rectification



- Image Reprojection
  - reproject image planes onto common
    plane parallel to line between optical centers
- Notice, only focal point of camera really matters

(Seitz)

Disparity defines correspondences



$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z} \quad Z = f \frac{T}{x_r - x_l}$$

Disparity: $d = x_r - x_l$

$$Z = f\frac{T}{d}$$

Then given Z, we can compute X and Y.

T is the stereo baseline

d measures the difference in retinal position between corresponding points

(Camps)

Correspondence with MRF

- Every pixel is a site.
- Label of a pixel is its disparity.
- Disparity implies two pixels match. Prob. depends on similarity of pixels.
- Disparity at one pixel related to others since nearby pixels have similar disparities.

Neighborhoods are important in stereo

First image Second image

Propagate information in constant regions

Avoid inconsistent, streaky solutions

Using MRFs

- We need to define sites and labels.
- Define neighborhood structure capturing conditional probability structure.
- Assign probabilities that capture problem.
- Find most probable labeling.
- Gibbs Distribution useful conceptualization.

Gibbs Distribution

- Cliques capture dependencies of neighborhoods.
  - {i} is a clique for all i.
  - $\{i_1, i_2, \ldots i_n\}$ is a clique if i_k in N_j for all $1 <= i,j <= n$.

Gibbs Distribution (2)

$$P(f) = \frac{1}{Z} e^{-U(f)/T}$$

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

$$Z = \sum_{f \in F} e^{-U(f)/T}$$

- $U(f)$ is energy function.
- $V_c(f)$ is clique potential
- Z is normalizing value.
  - Sum over all labelings.
- T is *temperature.*

MRF=GRF

- Given any MRF, we can define an equivalent GRF.
  - That means, find an appropriate energy *U(f)*
- To find *f* that maximizes P(*f*) it suffices to minimize:

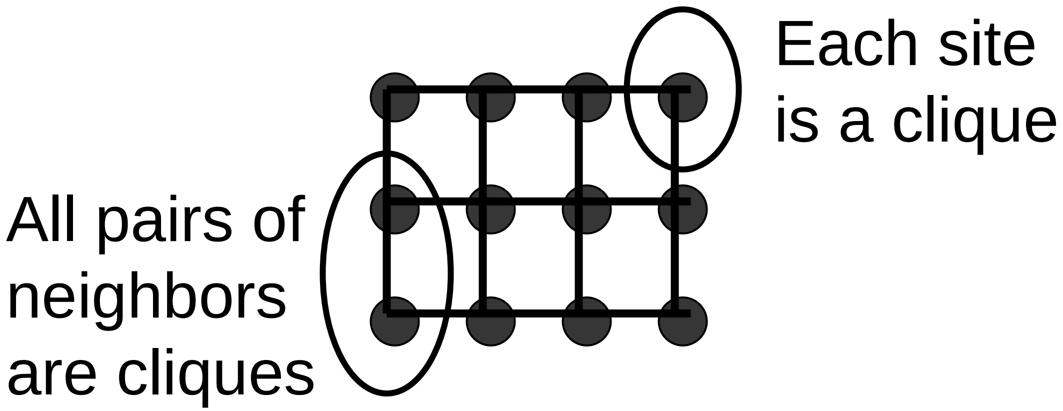
$$U(f) = \sum_{c \in C} V_c(f)$$

Significance

- Not so easy to determine absolute probability of labeling
  - We must sum over all configurations.
  - Exponential
- But we can determine relative probability of labeling efficiently.
  - This was trivial for Markov chains, but not for MRFs.
  - This is all we need to find most probable labeling.

Example: Piecewise Constant Image Restoration

- Every pixel is a site.
- Four connected neighborhoods



- Observation, d_i of intensity at site *i.*

Example, cont'd

$$P(f \mid d) = P(d \mid f)P(f)/P(d)$$

$$\text{Suppose}: d_i = f_i + e_i$$

$$e_i \text{ i.i.d. Gaussian } N(0, \sigma^2)$$

$$P(d \mid f) = \prod P(d_i \mid f_i)$$

$$P(d_i \mid f_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(f_i - d_i)^2 / 2\sigma^2}$$

$$U(d_i \mid f_i) \equiv (f_i - d_i)^2 / 2\sigma^2$$

Prior on labels

$$c = \{i\}, \quad V_c = \alpha_l \text{ for } f_i = l$$

$$c = \{i, j\} \quad V_c = \begin{cases} 0 & f_i = f_j \\ k & \text{else} \end{cases}$$

Prior on discontinuities

Minimize Energy: $\sum U(d_i \mid f_i) + \sum V_c$

Optimization

- Our problem is going to be to choose *f* to minimize this energy.
- Usually this is NP-hard: heuristics or exponential algorithms.
  - Greedy:
    - loop through sites, changing labeling to reduce energy.
    - Constant time to make this decision.

Optimization (2)

– Simulated Annealing (MCMC).

- Pick site, *i*, at random. Let *f'* be old labels, *f* be *f'* with f_i randomly changed.
- *p* = min(1, P(*f/f'*)).
- Replace *f'* with *f* with probability *p.*
- As *T* -> 0 method becomes deterministic. By slowly lowering *T* states of *f* become a Markov chain guaranteed to converge to global optimum.

$$P(f) = \frac{1}{Z} e^{-U(f)/T}$$

- This takes exponential time.

Optimization (3)

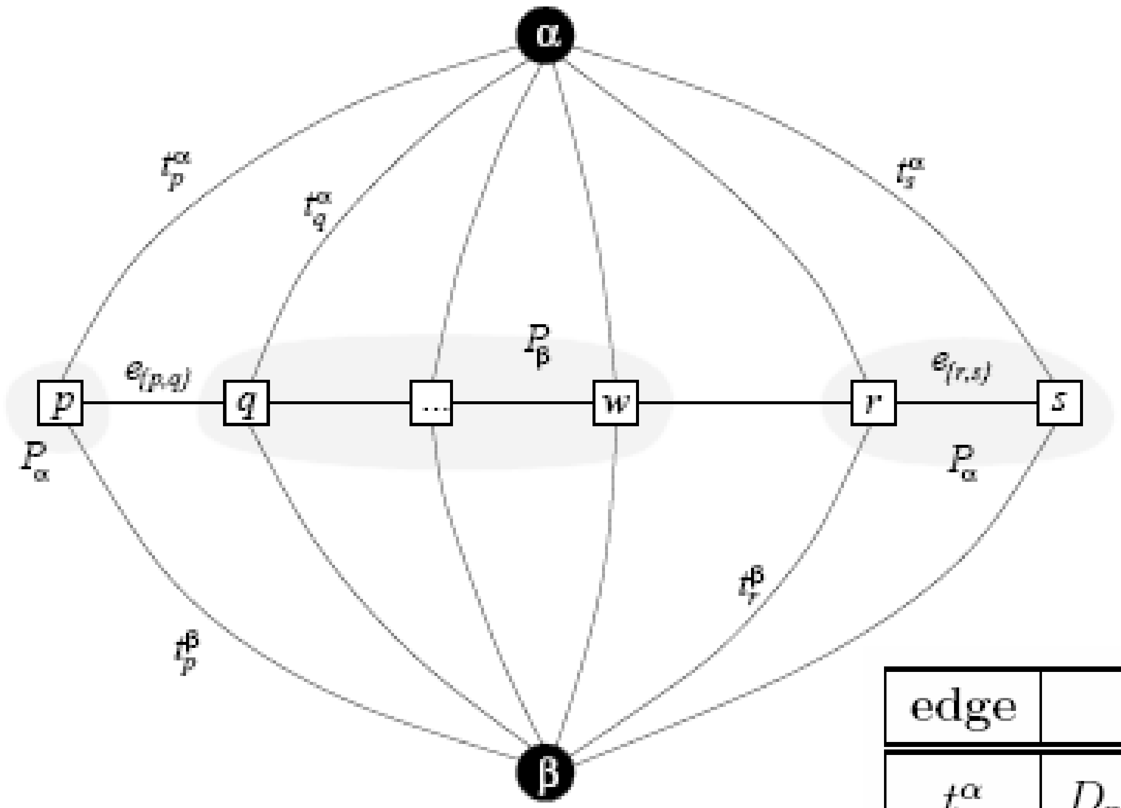
- Belief Propagation.
  - At each step, a site collects information from neighbors on their probable labeling. Passes info to each neighbor based on info from other neighbors (avoids repeating to neighbor what that neighbor has told.
  - In graph with no loops, like dynamic programming, forward-backward method.
  - In general MRF, heuristic (that has been analyzed). (eg., Yedidia, Freeman and Weiss).

Optimization (4)

- Graph cuts. (eg., Boykov, Veksler, and Zabih).
  - Instead of changing one label at a time, change many.
    - This allows alg. to escape many local mins.
  - Swap moves
    - For a pair of labels, α and β, find best relabeling of vertices with those two labels, using those two labels.

    α-expansion. Find best relabeling of all vertices so that they now are labeled α.
  - Both relabels can be posed as a graph cut problem, solved optimally in polynomial time.

α–β swap



edge	weight	for
t_p^α	$D_p(\alpha) + \sum_{q \in \mathcal{N}_p, q \notin \mathcal{P}_{\alpha\beta}} V(\alpha, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
t_p^β	$D_p(\beta) + \sum_{q \in \mathcal{N}_p, q \notin \mathcal{P}_{\alpha\beta}} V(\beta, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
$e_{\{p,q\}}$	$V(\alpha, \beta)$	$\{p,q\} \in \mathcal{N}$, $p,q \in \mathcal{P}_{\alpha\beta}$

Min-Cut gives best swap

- Min-cut
  - requires edge to one label be cut.
  - Cut between neighbors w/ diff. labels.
- Link to each label is cost of applying that label; cut means label is applied.
- Link between pixels = neighborhood cost (0 when same label).