CSci 4968 and 6270 — Computational Vision
Lectures 20-22 — Energy Minimization, Segmentation, Stereo and Graph Cuts

Charles Stewart
Department of Computer Science
Rensselaer Polytechnic Institute
8-15 November 2009

Overview

- Shift our focus to computing properties at each pixel of an image. We will view this property as a "label".
- Examples of primary interest:
  - In foreground-background segmentation, we have two labels — you guessed it — "object" (usually represented by a 1) and "background" (usually represented by a 0).
  - In stereo matching, we represent the "disparity" between images — the amount of shift along the epipolar lines — by a discrete set of values. These values become the labels.
- As background, we will start by looking at contour-based segmentation methods.
- Then, we formulate the segmentation energy we will minimize with graph cuts algorithms.
- Mapping to a graph and the graph cut solution.
- More advanced work in graph-cuts based segmentation.
- Application to stereo matching.
- **Project suggestions:**
  - Investigate more advanced techniques and properties of graph-cuts algorithms.
  - Explore the use of graph-cuts segmentation for a wide variety of applications.

Background Reading

- Background on segmentation and active contours may be found in Chapter 5 of Szeliski.
- The material on using graph cuts for segmentation is largely drawn from the paper, Boykov and Funka-Lea, Graph Cuts and Efficient N-D Image Segmentation, *International Journal of Computer Vision* 70:2, 109-131, 2006. If you are interested in a final project on segmentation and graph-cuts, you should start here. Sections 2 and 3 are the most helpful.
- The material on stereo is drawn from the paper, Boykov, Veksler and Zabih, Fast Approximation Energy Minimization via Graph Cuts, *IEEE Transactions on Pattern Analayis and Machine Intelligence* 23:11, 1222-1239, 2001. This paper is a bit more technical than the Boykov and Funka-Lea article.

Snakes, Active Contours and Level Sets

- Here is a quick overview of contour-based techniques for segmentation. Most work related to this topic was completed in the period 1987-2001.
- Goal is to find a curve in an image, perhaps requiring it to be closed. This curve should follow some boundary properties of the image.
- Curve is a vector-valued parametric function $\mathbf{C}(s)$, giving the image coordinates of the curve points for each value of s.
- Formulate in terms of energy:

  $$E(\mathbf{C}(s)) = \int_0^1 E_I(\mathbf{C}(s))ds + \int_0^1 E_S(\mathbf{C}(s))ds \tag{1}$$

  where

  - E_I is the "image" or "data" energy, based on the contents of the images
  - E_S is the "internal" or "smoothness" energy of the curve itself.
- A simple example the image energy is

  $$E_I(\mathbf{C}(s)) = -|\nabla I(\mathbf{C}(s))| \tag{2}$$

  which means the contour prefers area of high gradient magnitude.
- A simple example of the smoothness energy is

  $$E_S(\mathbf{C}(s)) = \lambda_1 \|\mathbf{C}'(s)\|^2 + \lambda_2 \|\mathbf{C}''(s)\|^2 \tag{3}$$

  These tend to shorten and straighten the curve.
- Original solutions to these were based on iterative finite-difference and finite-element methods, essentially solving differential equations on the image domain.
- Other solution methods:
  - Form contour in terms of B-splines and solve for a set of parameters
  - Fix two ends of the contour and solve using dynamic programming methods
- Ensuring a consistent topology of the contour is a difficult problem.
- An alternative approach is to embed the contour in what's known as a *level-set*, which is a function defined across the entire image domain. The contour is the "zero level set" of this function.
- The easiest way to think about this is as a signed distance function, where
  - Each pixel location records the minimum distance from the pixel to any point on the contour.

  - Points inside the region enclosed by the contour have negative distance and points outside the region have positive distance.
  - Zeros of the distance function may be found to subpixel accuracy in a number of ways.

- A pde formulation is designed to allow the level set function (the distance function) to evolve, using similar terms to the active contour methods above.

- Topological problems are gone.

- Solutions to these problems are primarily taken from the domain of differential equations and numerical methods.

- In all level-set and active contour methods, initialization is very important!

Overview of Graph-Cuts for Segmentation

- Focus primarily on the region instead of the boundary: label which pixels correspond to the object and which correspond to the background. The boundary, if desired, is found by looking for neighboring pixels having different labels.

- Problem is discrete / combinatorial: consider only image pixels and consider two possible labels per pixel.

- Exact solution is computed using max-flow / min-cut algorithms.

- Our discussion:
  - Energy formulation
  - Map to s/t graph, and solve using min-cut
  - Show that the solution is correct.
  - Impost hard constraints
  - Additional techniques and applications

  After these, we will examine stereo matching and the use of graph-cuts.

Energy Formulation for Segmentation

- Think of the image as a set, $\mathcal{P}$, of pixel locations, p.

- A segmentation is an assignment of labels to each pixel p. In other words:

$$a(p) = \begin{cases} 1 & p \text{ is an object pixel} \\ 0 & p \text{ is a background pixel} \end{cases} \tag{4}$$

- We then form a vector $\mathbf{a}$ of these labelings, with one entry in the vector per pixel.

- Similar to active contours, we can write the energy of the labeling:

$$E(\mathbf{a}) = \lambda \sum_{p \in \mathcal{P}} E_R(p) + \sum_{(p,q) \in \mathcal{N}} E_B(p,q). \tag{5}$$

  Here we have introduced a pixel neighborhood set $\mathcal{N}$.

  - At this point, the energy formulation is close to what's known at a Markov Random Field, with E_R and E_B being logarithms of probability terms.
  - Also, in the syntax of E_R and E_B there is an implicit (unstated) dependence on $\mathbf{a}$.

- To formulate $E_R(p)$ we need a statistical model of the object appearance and of the background appearance. We write these as probability functions of the intensities:

  - For the object it will be $Pr(I(p) \mid a(p) = 1)$.
  - For the background it will be $Pr(I(p) \mid a(p) = 0)$.

- These probability models can be learned, which is often the case in medical image segmentation, or they can be gathered from simple manual selection of example object and background regions.

- For $E_B(p,q)$, we need to write it in such a way that the energy is 0 when p and q are from the same region (both object or both background). When p and q are not from the same region, the penalty is highest when the pixel values are closest to each other. This will encourage object/background breaks at locations where the intensities differ.

- Boykov and Funka-Lea suggest

$$E_B(p,q) = \delta_{a(p),a(q)} B(p,q) \tag{6}$$

  where

$$B(p,q) = \exp\left(\frac{-(I(p) - I(q))^2}{2\sigma^2} \right) \frac{1}{\text{dist}(p,q)} \tag{7}$$

  and

$$\delta_{x,y} = \begin{cases} 0 & x = y \\ 1 & x \neq y \end{cases} \tag{8}$$

- We will discuss reasons for this in class; other forms are possible.

Mapping Into A Graph

- We want to map this into s/t (source/terminal) graph for solution by a min-cut / max-flow algorithm.

- Form a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

- Vertices: form one vertex for each pixel; add two special vertices, S and T:

  - S is the "source" and corresponds to the object

  - T is the "terminal" and corresponds to the background
  - Our cut must separate S and T — placing them into two separate connected components.

Therefore,

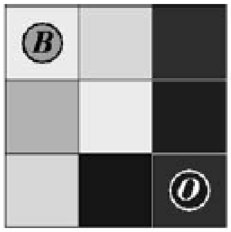
$$\mathcal{V} = \mathcal{P} \bigcup \{S, T\}. \tag{9}$$

- Edges in this undirected graph:
  - For each $p \in \mathcal{P}$, connect p to S and connect p to T. These edges are called *t-links*.
  - Each neighbor relation $\{p, q\}$ forms an edge. These edges are called *n-links*.

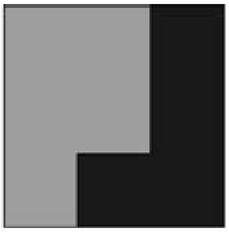
Therefore,

$$\mathcal{E} = \mathcal{N} \bigcup_{p \in \mathcal{P}} \{\{p, S\}, \{p, T\}\} \tag{10}$$

- An example is shown in the following figure from the Boykov and Funka-Lea paper:

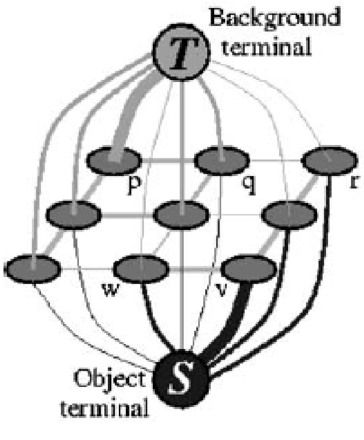


(a) Image with seeds.



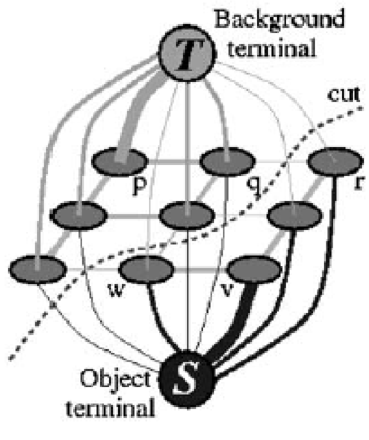
(d) Segmentation results.

⇓ ⇑



(b) Graph.

⇒



(c) Cut.

- Now, the all-important edge weights:

edge	**weight (cost)**	**for**
$\{p, q\}$	$B(p, q)$	$\{p, q\} \in \mathcal{N}$
$\{p, S\}$	$\lambda Pr(I(p) \mid a(p) = 0)$	$p \in \mathcal{P}$
$\{p, T\}$	$\lambda Pr(I(p) \mid a(p) = 1)$	$p \in \mathcal{P}$

Note the reversal here: with the cost of the connection between p and the background terminal being the cost, in the energy function, of labeling p as a foreground node.

Max-Flow / Min-Cut to Solve the Segmentation Problem

- A cut $\mathcal{C} \subset \mathcal{E}$ of an s/t graph is a subset of the edges such that S and T are in different connected components.
- The weight of a cut is
$$|C| = \sum_{e \in \mathcal{C}} w(e) \tag{11}$$
- A cut $\mathcal{C}$ is said to be *feasible* if
  - for each $p \in \mathcal{P}$ exactly one t-link is in $\mathcal{C}$;
  - for each $\{p, q\} \in \mathcal{N}$, $\{p, q\} \in \mathcal{C}$ if and only if p and q are t-linked to different terminals.
- Note how a feasible cut $\mathcal{C}$ defines our segmentation quite simply:
  - if p is connected to T following the cut, p is a background pixel;
  - if p is connected to S following the cut, p is an object pixel.
- $\hat{\mathcal{C}}$ is a *min-cut* if no other cut has lower weight.
- The Ford-Fulkerson algorithm can be used to solve for the min-cut in polynomial time.
- We then need to prove:
  - $\hat{\mathcal{C}}$ is feasible, which means it determines our segmentation.
  - $|\hat{\mathcal{C}}|$ yields the minimum energy segmentation.

Adding Hard Constraints

- In an interactive scenario, we can identify a set of pixels $\mathcal{O}$ that are definitely inside the segmented object and a set of pixels $\mathcal{B}$ that are definitely outside the segmented object.
- These can be used to establish object and background intensity statistics.
- These pixels are easily added to the s/t graph:
  - For $p \in \mathcal{B}$, $w(\{p, T\}) = \infty$ and $w(\{p, S\}) = 0$.
  - For $p \in \mathcal{O}$, $w(\{p, T\}) = 0$ and $w(\{p, S\}) = \infty$.

  Note again that the weight of the edge to the terminal that p must remain connected to gets infinite weight.
- It is easily shown that there must be a closed "contour" of n-links cut that separate the vertices in $\mathcal{B}$ from the vertices in $\mathcal{O}$. This is similar to the zero level set of level set methods. This connection has been explored extensively in recent papers.

Applications, Additional Techniques and Limitations

- Interactive segmentation for photo editing and medical image segmentation.
- Add directed edges. These can be used to enforce different types of weights based on intensity properties.
- Fast editing, where the entire max-flow / min-cut need not be re-estimated.
- Multi-way segmentation. This uses techniques originally developed for the stereo matching problem, which we will discuss next...

The Stereo Matching Problem

- Given: two images taken (nearly) simultaneously of a static scene by either two different cameras or one camera with a small translational motion between the two viewpoints. This motion is usually perpendicular to the line of sight of the camera.
- Problem: compute the depth of each pixel in each image
- Our discussion:
  - Reconsider camera geometry, returning to the simplified model of perspective projection that we started with.
  - Depth estimation as a matching problem — the "correspondence problem"
  - An energy formulation for the correspondence problem
  - Multi-way labeling
  - The α-expansion algorithm
  - Finding each α-expansion using a graph-cut formulation.

Stereo Camera Geometry and Image Disparity

- Our model for the problem is two cameras with a horizontal translation, perpendicular to the line of sight of the camera.
- As we will show in class, this implies that if (u, v) is the image coordinate of a point in the left image, then its matching point will be along the same row in the right image.
- The coordinates of this right image point are $(u - d, v)$ where $d > 0$ is what's know as the *disparity*.
- It is easy to show that disparity and depth (z) are inversely proportional to each other.
- In practice this simple camera geometry is difficult to achieve solely through construction of a stereo camera set-up. Intead we need either calibration or computation of the fundamental matrix, followed by a (hopefully slight) re-mapping of the two images, a process known as *rectification*.

The Correspondence Problem: Dense Matching and Constraints

- Our goal is to compute disparity at each pixel in an image:
    - Contrast this with the output of structure-from-motion algorithms where 3d point locations are only known for keypoints.
- The estimation of disparity is based on two fundamental assumptions:
    - The intensities of corresponding pixel locations are the same
    - Disparity (inverse depth) varies slowly throughout the images except at discontinuities.
- When does this fail:
    - Cameras with different response functions and gain controls
    - Specular (actually, non-Lambertian) surfaces
    - At occlusion boundaries there are pixels that are seen in only one of the two images.
    - Scenes with large numbers of discontinuities (branches of trees)
- For implementing and testing stereo algorithms — as opposed to building complete robotic systems — we will also assume that the range of depths, and therefore of image disparities, is can be established manually as a parameter.
    - The disparity range will be denoted d_{min} to d_{max}.
- In short the function we want to compute is $d(u, v)$ for each pixel (u, v) in the (say) left image.

A Stereo Matching Energy Equation

- Once again we will use $\mathcal{P}$ as a set of pixel locations and refer to a particular pixel as $p \in \mathcal{P}$. In terms of our immediately-preceding discussion of stereo, we have $p = (u, v)$.
- We will also have a set of neighborhood relations $\mathcal{N}$ and write $\{p, q\} \in \mathcal{N}$ for pixel locations that are neighbors.
- Instead of using d for the disparities we want to compute, we will use f and write f_p for the disparity at pixel p.
    - This is consistent with the notation in the Boykov, Veksler and Zabih paper and makes it clear that the α-expansion algorithm we are going to discuss is more general than the stereo disparity problem.
- The energy equation is

$$E(f) = \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q) + \sum_{p \in \mathcal{P}} D_p(f_p) \tag{12}$$

  where D_p is a data term and $V_{p,q}$ is a smoothness term.

- We will consider each of these terms in succession.
- We will assume that f_p, for each $p \in \mathcal{P}$, is taken from a discrete set of possible disparities values, e.g. $f_p \in \mathcal{L}$, where $\mathcal{L} = \{d_{\text{min}}, \ldots, d_{\text{max}}\}$.

The Data Term

- We will write the images as I_L and I_R.
- The data term could be as simple as the difference in intensities, so that

$$D_p(f_p) = \min((I_L(p) - I_R(p - f_p))^2, \text{const}) \tag{13}$$

  where const is an unspecified constant (for our discussion, at least) that caps the error penalty for differences in intensity.

- A more sophisticated error penalty term, proposed by Birchfield and Tomasi in a 1998 paper in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, does a better job accounting for the discrete nature of the image pixel locations:

$$\begin{aligned} C_f(p, f_p) &= \min_{f_p - 1/2 \le x \le f_p + 1/2} |I_L(p) - I_R(p - x)| && (14) \\ C_b(p, f_p) &= \min_{p - 1/2 \le x \le p + 1/2} |I_L(x) - I_R(p - f_p)| && (15) \\ D_p(f_p) &= \left[\min\{C_f(p, f_p), C_b(p, f_p), \text{const}\}\right]^2. && (16) \end{aligned}$$

  We can think of the sub-pixel values of I_L and I_R being computed through linear interpolation. The actual computation of the minimum can be achieved using a constant time algorithm.

- Importantly, since we have a discrete set of labels, we will assume that $D_p(f_p)$ can be precomputed and stored for all $p \in \mathcal{P}$ and all $f \in \mathcal{L}$.

The Smoothness Term

- Many forms for $V_{p,q}(f_p, f_q)$ are possible.
- If the labels are linearly ordered, we can have

$$V_{p,q}(f_p, f_q) = \min((f_p - f_q)^2, \text{const}) \tag{17}$$

- In stereo, the constant is usually pretty small because after a even a relatively small change in disparity, we know the points must be from different surfaces and at that point we don't care how big the change in disparity is.
- This leads to the idea of using the Potts model:

$$u(f_p, f_q) = \begin{cases} 0 & f_p = f_q \\ 1 & f_p \neq f_q \end{cases} \tag{18}$$

  and a resulting

$$V_{p,q}(f_p, f_q) = K \cdot u(f_p, f_q) \tag{19}$$

- We can augment this with a term that encourages breaks in disparity to occur near intensity discontinuities. This could be as simple as

  $$V_{p,q}(f_p, f_q) = u(f_p, f_q) \cdot v(I_p, I_q) \tag{20}$$

  where

  $$v(I_p, I_q) = \begin{cases} 2K & |I_p - I_q| \leq 2\sigma \\ K & |I_p - I_q| > 2\sigma \end{cases} \tag{21}$$

  where σ is an estimate of the noise in the image intensities.

- Finally, for what follows, it will be important that $V_{p,q}(f_p, f_q)$ is a *metric*, which means it must satisfy the following properties:

  1. $V_{p,q}(f_p, f_q) \geq 0$, with equality if and only if $f_p = f_q$.
  2. $V_{p,q}(f_p, f_q) = V_{p,q}(f_q, f_p)$
  3. $V_{p,q}(f_p, f_q) < V_{p,r}(f_p, f_r) + V_{r,q}(f_r, f_q)$

  The last of these is the *triangle inequality*.

The Expansion-Move Algorithm

- The segmentation problem we solved with graph-cuts only had two labels, object and background. The labeling problem for stereo has many more possible labels.

- Therefore, the graph-cuts algorithm can not be used immediately.

- Instead, we will include the graph-cuts solution as a step in the many-labels labeling algorithm.

- Boykov, Veksler and Zabih present two techniques, the $\alpha - \beta$ swap and the α expansion (also called the expansion move). We will only consider the α expansion.

- Given a labeling, f, an "expansion move" for a given label is a new labeling f', such that

  - All pixels currently labeled $f_p = \alpha$ remain unchanged.
  - 0 or more pixels labeled $f_p \neq \alpha$, become assigned $f'_p = \alpha$.

  This is called an α-expansion

- Here is the algorithm:

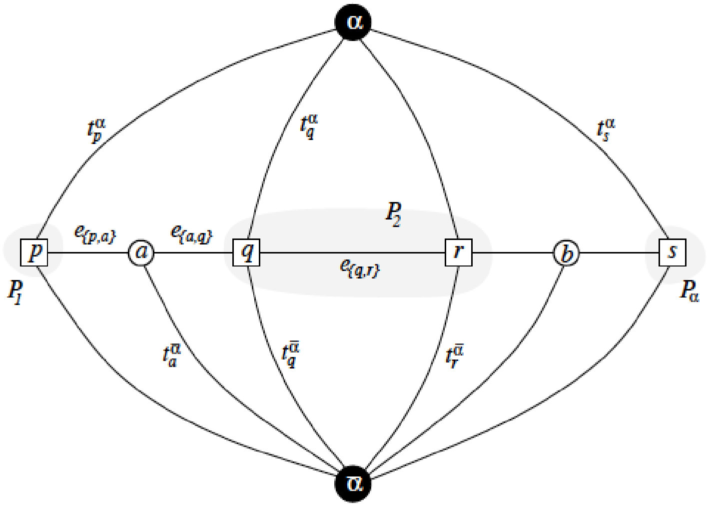
  1. Assign an initial labeling (initial set of disparities) to the image, perhaps just randomly;
  2. `expanded = false`;
  3. for each label $\alpha \in \mathcal{L}$
     (a) Find the α-expansion f' producing the greatest reduction in energy.
     (b) If $E(f') < E(f)$ then set $f = f'$ and set `expanded = true`
  4. If `expanded`, go to step 2.

5. Return the labeling f.

- Given this outline, obviously, the primary step to consider is to find the optimal α-expansion for any label.
- Typically, the overall algorithm converges in a relatively small number of iterations through the outer loop.

Solving For Each Expansion-Move Optimally Using Graph-Cuts

- We will structure the graph. In forming the vertices and edges it will become clear how the cut will solve for the desired α-expansion.
- For the label $\alpha \in \mathcal{L}$ currently being considered, let ${}_{\alpha}$ be the set of pixels that currently have label α, i.e.
$$\mathcal{P}_\alpha = \{p \mid p \in \mathcal{P} \text{ and } f_p = \alpha\} \tag{22}$$
These pixels will not change labels during this expansion move.
- The terminal T — remember this was the background terminal — is now called $\overline{\alpha}$.
- The terminal S — formerly called the object terminal – is now called α.
- As before, there is one vertex for each $p \in \mathcal{P}$ and these vertices are each connected to α and $\overline{\alpha}$.
- For each pair $\{p, q\} \in \mathcal{N}$ such that $f_p \neq f_q$, we add a special vertex $a_{p,q}$ and connect it via an edge to p, to q and to $\overline{\alpha}$.
- An example from the Boykov, et al paper is shown below



- Before explaining the figure, it is very important to note that the edges in an cut, $\mathcal{C}$, now specify the connections that are actually part of the energy formulation. One implication of this is that if the connection from p to α is cut, then p will have label α in the expansion move.

- Getting a bit more formal, the set of vertices is

$$\mathcal{V}_\alpha = \left\{ \mathcal{P} \bigcup \{\alpha, \overline{\alpha}\} \bigcup \{a_{p,q} \mid \{p,q\} \in \mathcal{N} \text{ and } f_p \neq f_q\} \right\} \tag{23}$$

- Now for the edges and their weights. First, for $\{p \in \mathcal{P}_\alpha\}$ — the pixels currently having the label α:

Name	Edge	Weight
t_p^α	$\{p, \alpha\}$	$D_p(\alpha)$
$t_p^{\overline{\alpha}}$	$\{p, \overline{\alpha}\}$	∞

  Note that this means the connection t_p^α will always be severed for pixels that are currently labeled $f_p = \alpha$.

- Next, for $\{p \in \mathcal{P} - \mathcal{P}_\alpha\}$ — the pixels that currently do not have the label α:

Name	Edge	Weight
t_p^α	$\{p, \alpha\}$	$D_p(\alpha)$
$t_p^{\overline{\alpha}}$	$\{p, \overline{\alpha}\}$	$D_p(f_p)$

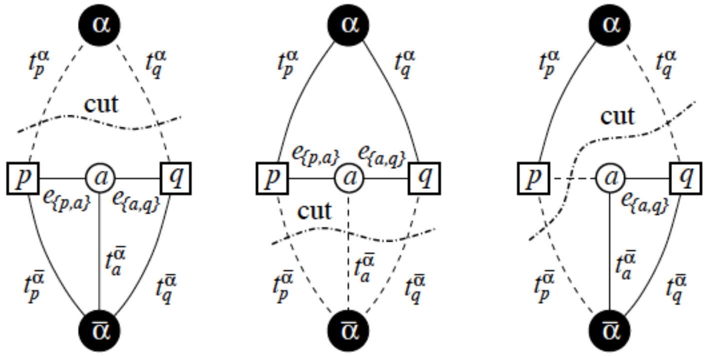
  where f_p is the pixel's current label (which is not α).

- For pixel pairs $\{p, q\} \in \mathcal{N}$, where currently $f_p = f_q$, add an edge $e_{p,q}$ with weight $V_{p,q}(f_p, \alpha)$.
  - Each of these edge links that is severed implies that a cost of $V_{p,q}(f_p, \alpha)$ is added to the cut/energy.
  - Note that this is where property 2 of V is required.

- Finally, consider the three edges from $a_{p,q}$, the auxiliary vertex added between two neighboring pixels/vertices currently having different labels:

Name	Edge	Weight
$t_a^{\overline{\alpha}}$	$\{a, \overline{\alpha}\}$	$V_{p,q}(f_p, f_q)$
$e_{p,a}$	$\{p, a\}$	$V_{p,q}(f_p, \alpha)$
$e_{a,q}$	$\{a, q\}$	$V_{p,q}(\alpha, f_q)$

  Using the following figure and recalling the triangle inequality (property 3 above), we can show that at most one of these edges can be severed by a cut.



In particular, there are four cases:

1. If p and q both have their connections severed from α, meaning that both are labeled $f'_p = f'_q = \alpha$ in the new labeling, then none of the edges connected to $a_{p,q}$ are cut and the neighborhood cost $V_{p,q}(f'_p, f'_q) = V_{p,q}(\alpha, \alpha) = 0$ in the new energy. This is illustrated on the left side of the above figure.
2. If p and q both have their connections to $\overline{\alpha}$ cut then, because of the triangle inequality on V, the t-link $t_a^{\overline{\alpha}}$ must also be cut. The cost of this cut is the $V_{p,q}(f_p, f_q)$ value for the current labeling. This is illustrated in the center of the above figure.
3. If p has its connection to $\overline{\alpha}$ cut and q has its connection to α cut, then f_p remains unchanged in the new labeling (i.e. $f'_p = f_p$) and $f'_q = \alpha$ in the new labeling. This means that the neighbor energy of $\{p, q\}$ must be $V_{p,q}(f_p, \alpha)$, corresponding to cutting $e_{p,a}$. The other two edges, $e_{q,a}$ and $t_a^{\overline{\alpha}}$, can not be cut because of the triangle inequality. This is illustrated on the right side of the above figure.
4. If p has its connection to α cut and q has its connection to $\overline{\alpha}$ cut, then $f'_p = \alpha$ and $f'_q = f_q$ in the new labeling. This means that the neighbor energy of $\{p, q\}$ must be $V_{p,q}(\alpha, f_q)$, corresponding to cutting $e_{a,q}$. The other two edges, $e_{p,a}$ and $t_a^{\overline{\alpha}}$ can not be cut because of the triangle inequality. While not shown in the above figure, this case is the mirror image of the diagram on the right.

- This completes the construction of the graph.
- Intuitively, it is fairly clear from the construction that the minimum cut of this graph gives the optimal α-expansion move. We will not give a formal proof here.
- See paper and the Middlebury website `http:://vision.middlebury.edu/stereo/` for example results.

Properties

- Each α-expansion by itself is optimal
- Overall expansion move converges to a "strong local minimum" of the energy.
- Finding the global minimum is NP-Hard

- In practice, when ground-truth disparities are available, the energy of the labeling actually found is less than the energy of the ground-truth labeling! This means that improvements can be obtained more through improving the energy formulation than improving the search algorithm. We are restricted here, however, by requiring than $V_{p,q}$ be a metric.

Enhancements and Alternatives

- Main alternative approach is "loopy belief propagation" algorithm.
- Many, many variations on the basic energy formulation have been presented. Ideas include:
  - Include explicit labeling of occluded pixels
  - Left/right symmetric matching.
  - More sophisticated, statistically-derived smoothness terms.
  - Layering and initial disparity plane fitting followed by graph-cuts within each plane.
- Many of these, together with experimental results, are gathered at the Middlebury stereo website `http:://vision.middlebury.edu/stereo/`

Applications

- View interpolation and image-based rendering. Here the occcluded pixel problem is quite important. How well do current algorithms do?
- Robot navigation.

Summary of Our Stereo Discussion

- Assumptions about stereo camera geometry lead to a one-dimensional search for the disparity at each pixel.
- Disparity computation problem formulated as an energy minimization problem with a data term and a smoothness term.
- This is treated in turn as a labeling problem with a discrete set of labels to represent the possible disparities.
- The Expansion Move algorithm, or $alpha$-expansion algorithm,
- For each possible label, the optimal expansion move is found in polynomial time using graph-cuts.
- While the overall solution is NP-Hard, the actual solutions found in practice have energy as low or lower than those of ground-truth disparities.
- Graph-based algorithms have seen substantially-increased use in computer vision over the past decade.