13. Graph Cut Optimization
Stitching a wide-angle view
Texture synthesis with graph cuts

Synthesized Texture (Initialization)

Seam Boundaries

Sample Texture

Synthesized Texture (After 5 steps of Refinement)

Step 2

Step 3

Step 4

Step 5

Seam Costs

[Kwatra et al. 2005]
Accounting for existing seams

Existing Pixels A

New Patch B

Figure 3: (Left) Finding the best new cut (red) with an old seam between pixels 1 and 8. (Right) An example of a graph for texture synthesis, where nodes 1, 2, 4, 5, 7, and 8 correspond to our pixels of interest. The figure illustrates how to incorporate existing seams into the graph cut problem to improve texture synthesis.

Existing Pixels A

New Patch B

Figure 3: (Left) Finding the best new cut (red) with an old seam between pixels 1 and 8. (Right) An example of a graph for texture synthesis, where nodes 1, 2, 4, 5, 7, and 8 correspond to our pixels of interest. The figure illustrates how to incorporate existing seams into the graph cut problem to improve texture synthesis.

[Boykov et al. 1999]
Graph cut texture results

Figure 8: 2D texture synthesis results. We show results for textured and natural images. The smaller images are the example images used for synthesis. Shown are CHICK PEAS, TEXT, NUTS, ESCHER, MACHU PICCHU, CROWDS and SHEEP from left to right and top to bottom.

Figure 9: Comparison of our graph cut algorithm with Image Quilting [Efros and Freeman 2001]. Shown are KEYBOARD and OLIVES. For OLIVES, an additional result is shown that uses rotation and mirroring of patches to increase variety. The quilting result for KEYBOARD was generated using our implementation of Image Quilting; the result for OLIVES is courtesy of Efros and Freeman.

[Kwatra et al. 2005]
Segmentation with graph cuts

Boykov and Funka-Lea

Figure 3. A simple 2D segmentation example for a 3×3 image. The seeds are $O = \{v\}$ and $B = \{p\}$. The cost of each edge is reflected by the edge's thickness. The boundary term (4) defines the costs of n-links while the regional term (3) defines the costs of t-links. Inexpensive edges are attractive choices for the minimum cost cut. Hard constraints (seeds) (8,9) are implemented via infinity cost t-links. A globally optimal segmentation satisfying hard constraints can be computed efficiently in low-order polynomial time using max-flow/min-cut algorithms on graphs (Ford and Fulkerson, 1962; Goldberg and Tarjan, 1988; Cook et al., 1998).

2.1. Basic Ideas and Background Information

First, we will introduce some terminology. A graph $G = \langle V, E \rangle$ is defined as a set of nodes or vertices $V$ and a set of edges $E$ connecting "neighboring" nodes. For simplicity, we mainly concentrate on undirected graphs where each pair of connected nodes is described by a single edge $e = \{p, q\} \in E$.

A simple 2D example of an undirected graph that can be used for image segmentation is shown in Fig. 3(b). The nodes of our graphs represent image pixels or voxels. There are also two specially designated terminal nodes $S$ (source) and $T$ (sink) that represent "object" and "background" labels. Typically, neighboring pixels are interconnected by edges in a regular grid-like fashion. Edges between pixels are called n-links where n stands for "neighbor". Note that a neighborhood system can be arbitrary and may include diagonal or any other kind of n-links. Another type of edges, called t-links, are used to connect pixels to terminals. All graph edges $e \in E$ including n-links and t-links are assigned some nonnegative weight (cost) $w_e$. In Fig. 3(b) edge costs are shown by the thickness of edges.

An s-t cut is a subset of edges $C \subset E$ such that the terminals $S$ and $T$ become completely separated on the induced graph $G(C) = \langle V, E \setminus C \rangle$. Note that a cut [Boykov & Funka-Lea 2006]
$\alpha$-Expansion

[Boykov et al. 2001]
\(\alpha\)-Expansion algorithm

1. Start with an arbitrary labeling \(f\)
2. Set success := 0
3. For each label \(\alpha \in \mathcal{L}\)
   3.1. Find \(\hat{f} = \arg\min E(f')\) among \(f'\) within one \(\alpha\)-expansion of \(f\)
   3.2. If \(E(\hat{f}) < E(f)\), set \(f := \hat{f}\) and success := 1
4. If success = 1 goto 2
5. Return \(f\)
Multi-way cuts: Photomontage

Figure 1 From a set of five source images (of which four are shown on the left), we quickly create a composite family portrait in which everyone is smiling and looking at the camera (right). We simply flip through the stack and coarsely draw strokes using the designated source image objective over the people we wish to add to the composite. The user-applied strokes and computed regions are color-coded by the borders of the source images on the left (middle).

[Agarwala et al. 2004]
Multi-way cuts: Photomontage

The digital photomontage process begins with a set of source images. For best results, the source images should generally be related in some way. For instance, they might all be of a changing scene, but with the camera locked down to the same scene but with different lighting or camera positions. Or they might all be of a specific specimen, such as a fly's head with the focus plane at a different depth for each image. We denote this set the source image stack.

When a composite is created, each pixel specifies a property that the user would like to see at that location. This property is specified by the user through an image objective to compute the graph-cut composite automatically (top left, with an inset to show detail, and the labeling shown directly below). A summary of the general image objectives that may be applied is shown in Table 2.1. These objectives are applied to a single pixel (subject to a particular histogram quantization metric), to only a few pixels through a “painting”-style interface, globally to the entire image or to only a few pixels through a “painting”-style interface, globally to only a few pixels through a “painting”-style interface.

Table 2.1: Image Objectives

<table>
<thead>
<tr>
<th>Objective</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designated color</td>
<td>A specific desired color to either match or avoid</td>
</tr>
<tr>
<td>Minimum color</td>
<td>The least or most similar color</td>
</tr>
<tr>
<td>Maximum color</td>
<td>The color least or most similar to the current composite color</td>
</tr>
<tr>
<td>Maximum luminance</td>
<td>The darkest or lightest pixel in the span</td>
</tr>
<tr>
<td>Maximum contrast</td>
<td>The pixel from the span with the maximum contrast</td>
</tr>
<tr>
<td>Maximum difference</td>
<td>The color most different from that of the current composite color</td>
</tr>
<tr>
<td>Maximum likelihood</td>
<td>The least or most common pixel in the span</td>
</tr>
</tbody>
</table>

Figure 2: Photomontage process.