Visual Grouping:
Contours and Regions in Natural Images

Jitendra Malik
University of California at Berkeley
From Pixels to Perception

- Water
- Grass
- Sand
- Tiger
- Eye
- Head
- Legs
- Tail
- Shadow
- Mouth
Recognition, Reconstruction & Reorganization
Each of the 6 directed arcs in this diagram is a useful direction of information flow.
Berkeley Segmentation DataSet [BSDS]

Consistency

- \( A, C \) are refinements of \( B \)
- \( A, C \) are mutual refinements
- \( A, B, C \) represent the same percept
  - **Attention** accounts for differences

Two segmentations are consistent when they can be explained by the same segmentation tree (i.e., they could be derived from a single perceptual organization).
Contours can be defined by any of a number of cues (P. Cavanagh)
Boundaries of image regions defined by a number of cues

- Brightness
- Color
- Texture
- Motion (in video)
- Binocular Diparity (if available)
- Familiar objects
Cue-Invariant Representations

Gray level photographs
Objects from motion
Objects from luminance
Objects from texture
Objects from disparity
Line drawings

Grill-Spector et al., Neuron 1998
Challenges: texture cue, cue combination

Goal: learn the posterior probability of a boundary $P_b(x,y,\theta)$ from local information only
Oriented Feature Gradient
Visual Texture

Jitendra Malik
University of California, Berkeley
Example Natural Materials

- Terrycloth
- Rough Plastic
- Plaster-b
- Sponge
- Rug-a
- Painted Spheres

Columbia-Utrecht Database (http://www.cs.columbia.edu/CAVE)
Example Natural Materials

- Terrycloth
- Rough Plastic
- Plaster-b
- Sponge
- Rug-a
- Painted Spheres

Columbia-Utrecht Database (http://www.cs.columbia.edu/CAVE)
Materials under different illumination and viewing directions

Different illumination and viewing directions

Plaster-a
Crumpled Paper
Concrete
Plaster-b (zoomed)
Julesz’s texton theory

1. Human Vision operates in two distinct modes
   1. Pre-attentive vision - parallel, instantaneous, without scrutiny, independent of the number of patterns
   2. Attentive vision - serial search by focal attention in 50 ms steps limited to a small aperture as in form recognition

2. Textons are
   1. Elongated blobs - e.g. rectangles, ellipses, line segments with specific orientations, widths and lengths
   2. Terminators - ends of line segments
   3. Crossings of line segments
Orientation is a texton
Terminators are textons
Texture gradient: \( \max_i (|\nabla \text{PIR}_i \ast G_\sigma|) \)
Six sample textures

(L +) [232]

(+ ○) [201]

(Δκ) [154]

(+ T) [136]

(T L) [97]

(R-R) [50*]
Table 3. Comparison of Predictions from Texture Segmentation Algorithm with Two Sets of Psychophysical Data\(^a\)

<table>
<thead>
<tr>
<th>Texture Pair</th>
<th>Discriminability</th>
<th>Predicted Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ o</td>
<td>100 (saturated)</td>
<td>n.a.</td>
</tr>
<tr>
<td>+ □</td>
<td>88.1</td>
<td>n.a.</td>
</tr>
<tr>
<td>L +</td>
<td>68.6</td>
<td>0.736</td>
</tr>
<tr>
<td>L M</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>Δ'</td>
<td>52.3</td>
<td>0.4–0.55</td>
</tr>
<tr>
<td>+ T</td>
<td>37.6</td>
<td>0.496</td>
</tr>
<tr>
<td>+ X</td>
<td>30.3</td>
<td>n.a.</td>
</tr>
<tr>
<td>T L</td>
<td>30.6</td>
<td>0.421</td>
</tr>
<tr>
<td>L₂ M₂</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>R-mirror-R</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
</tbody>
</table>
Texture Recognition

Felt?
Polyester?
Terrycloth?
Rough Plaster?
Leather?
Plaster?
Concrete?
Crumpled Paper?
Sponge?
Limestone?
Brick?
2D Textons

• Goal: find canonical local features in a texture;

1) Filter image with linear filters:

2) Vector quantization (k-means) on filter outputs;

3) Quantization centers are the textons.

• Spatial distribution of textons defines the texture;
2D Textons (cont’d)
Texton Labeling

• Each pixel labeled to texton $i$ (1 to K) which is *most similar in appearance*;

• Similarity measured by the Euclidean distance between the filter responses;
Material Representation

• Each material is now represented as a spatial arrangement of symbols from the texton vocabulary;

• Recognition --- ignore spatial arrangement, use histogram (K=100);
Histogram Models for Recognition (Leung & Malik, 1999)
Similarity of materials

• Similarity between histograms measured using chi-square difference:
### Similarity Matrix

<table>
<thead>
<tr>
<th>Material</th>
<th>Felt</th>
<th>Terrycloth</th>
<th>Rough Plastic</th>
<th>Leather</th>
<th>Sandpaper</th>
<th>Pebbles</th>
<th>Plastera</th>
<th>Plasterb</th>
<th>Rough Paper</th>
<th>Artificial Grass</th>
<th>Roof Shingle</th>
<th>Aluminum Foil</th>
<th>Cork</th>
<th>Rough Tile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Felt</td>
<td>0.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Terrycloth</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Rough Plastic</td>
<td>0.0</td>
<td>0.0</td>
<td>0.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Leather</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Sandpaper</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Pebbles</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Plastera</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Plasterb</td>
<td>0.0</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.8</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Rough Paper</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Artificial Grass</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Roof Shingle</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>1.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Aluminum Foil</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Cork</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td>Rough Tile</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Oriented Feature Gradient
Individual Features

- 1976 CIE L*a*b* colorspace
- Brightness Gradient $BG(x,y,r,\theta)$
  - Difference of L* distributions
- Color Gradient $CG(x,y,r,\theta)$
  - Difference of $a*b*$ distributions
- Texture Gradient $TG(x,y,r,\theta)$
  - Difference of distributions of V1-like filter responses

These are combined using logistic regression
Filter Outputs
Texture gradient = Chi square distance between texton histograms in half disks across edge
Various Cue Combinations (Martin, Fowlkes, Malik, 2004)
Exploiting global constraints: Image Segmentation as Graph Partitioning

Build a weighted graph $G = (V, E)$ from image pixels $V$ and connections between pairs of nearby pixels $E$.

Partition graph so that similarity within group is large and similarity between groups is small — *Normalized Cuts* [Shi & Malik 97].
$W_{ij}$ small when intervening contour strong, small when weak.

$C_{ij} = \max P_b(x,y)$ for $(x,y)$ on line segment $ij$; \quad $W_{ij} = \exp \left(-\frac{C_{ij}}{\sigma}\right)$
How to partition a graph

- We can find the minimum cut efficiently, but this tends to break the graph into isolated little pieces
Normalized Cut is a better measure..

We normalize by the total volume of connections

\[ N \text{cut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \]

where \( \text{assoc}(A, V) = \sum_{u \in A, t \in V} w(u, t) \)
Solving the Normalized Cut problem

- Exact discrete solution to Ncut is NP-hard even on regular grid [Papadimitriou’ 97]
- We first transform to

\[
\min_x N\text{cut}(x) = \min_y \frac{y^T(D - W)y}{y^TDy}
\]

with the condition \( y(i) \in \{1, -1\} \) and \( y^TD1 = 0 \)

- Drawing on spectral graph theory, good approximation can be obtained by solving a generalized eigenvalue problem.

\[(D - W)y = \lambda Dy.\]
Normalized Cuts as a Spring-Mass system

• Each pixel is a point mass; each connection is a spring:

\[ (D - W) y = \lambda Dy \]

• Fundamental modes are generalized eigenvectors of

\[ (D - W) y = \lambda Dy \]
Eigenvectors carry contour information
Comparison to other approaches (2009)
Fast Edge Detection Using Structured Forests

Piotr Dollár and C. Lawrence Zitnick
Microsoft Research
{pdollar,larryz}@microsoft.com

Abstract—Edge detection is a critical component of many vision systems, including object detectors and image segmentation algorithms. Patches of edges exhibit well-known forms of local structure, such as straight lines or T-junctions. In this paper we take advantage of the structure present in local image patches to learn both an accurate and computationally efficient edge detector. We formulate the problem of predicting local edge masks in a structured learning framework applied to random decision forests. Our novel approach to learning decision trees robustly maps the structured labels to a discrete space on which standard information gain measures may be evaluated. The result is an approach that obtains realtime performance that is orders of magnitude faster than many competing state-of-the-art approaches, while also achieving state-of-the-art edge detection results on the BSDS500 Segmentation dataset and NYU Depth dataset. Finally, we show the potential of our approach as a general purpose edge detector by showing our learned edge models generalize well across datasets.
Fig. 1. Edge detection results using three versions of our Structured Edge (SE) detector demonstrating tradeoffs in accuracy vs. runtime. We obtain realtime performance while simultaneously achieving state-of-the-art results. ODS numbers were computed on BSDS [1] on which the popular gPb detector [1] achieves a score of .73. The variants shown include SE, SE+SH, and SE+MS+SH, see §4 for details.
Fig. 2. Illustration of the decision tree node splits: (a) Given a set of structured labels such as segments, a splitting function must be determined. Intuitively a good split (b) groups similar segments, whereas a bad split (c) does not. In practice we cluster the structured labels into two classes (d). Given the class labels, a standard splitting criterion, such as Gini impurity, may be used (e).
\( x \in \mathbb{R}^{32 \times 32 \times K} \) where \( K \) is the number of channels. We use features of two types: pixel lookups \( x(i, j, k) \) and pairwise differences \( x(i_1, j_1, k) - x(i_2, j_2, k) \), see §2.

Inspired by Lim et al. [31], we use a similar set of color and gradient channels (originally developed for fast pedestrian detection [12]). We compute 3 color channels in CIE-LUV color space along with normalized gradient magnitude at 2 scales (original and half resolution). Additionally, we split each gradient magnitude channel into 4 channels based on orientation. The result is 3 color, 2 magnitude and 8 orientation channels, for a total of 13 channels.

We blur the channels with a radius 2 triangle filter and downsample by a factor of 2, resulting in \( 32 \cdot 32 \cdot 13/4 = 3328 \) candidate features \( x \). Motivated by [31], we also compute pairwise difference features. We apply a large triangle blur to each channel (8 pixel radius), and downsample to a resolution of \( 5 \times 5 \). Sampling all candidate pairs and computing their differences yields an additional \( \binom{5 \cdot 5}{2} = 300 \) candidate features per channel, resulting in 7228 total candidate features.
Fig. 9. Results on BSDS500. Structured edges (SE) and SE coupled with hierarchical multiscale segmentation (SE+multi-ucm) [2] achieve top results. For the SE result we report the SE+MS+SH variant. See Table 1 for additional details including method citations and runtimes. SE is orders of magnitude faster than nearly all edge detectors with comparable accuracy.
We trained a detector that combine multiple cues to find the posterior probability of a boundary/edge $P_b(x,y,\theta)$. After that we used Normalized Cuts (Shi & Malik) to find regions.
Multiscale Combinatorial Regions
Arbelaez, Pont-Tuset, Barron, Marques & Malik, CVPR 2014
Fig. 5. **Object segmentation as combinatorial optimization**: Examples of objects (b), (c), formed by selecting regions from a hierarchy (a).
Examples
Fig. 8. **Object Proposals: Jaccard index at instance level.** Results on SegVOC12, SBD, and COCO.
Fig. 9. **Object Proposals: Jaccard index at class level.** Results on SegVOC12, SBD, and COCO.