Segmentation Using Multiscale Cues

Stella X. Yu
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720-1776, USA
stellayu@cs.berkeley.edu

Abstract

Edges at multiple scales provide complementary grouping cues for image segmentation. These cues are reliable within different ranges. The larger the scale of an edge, the longer range the grouping cues it designates, and the greater impact it has on the final segmentation. A good segmentation respects grouping cues at each scale.

These intuitions are formulated in a graph-theoretic framework, where multiscale edges define pairwise pixel affinity at multiple grids, each captured in one graph. A novel criterion called average cuts of normalized affinity is proposed to evaluate a simultaneous segmentation through all these graphs. Its near-global optima can be solved efficiently.

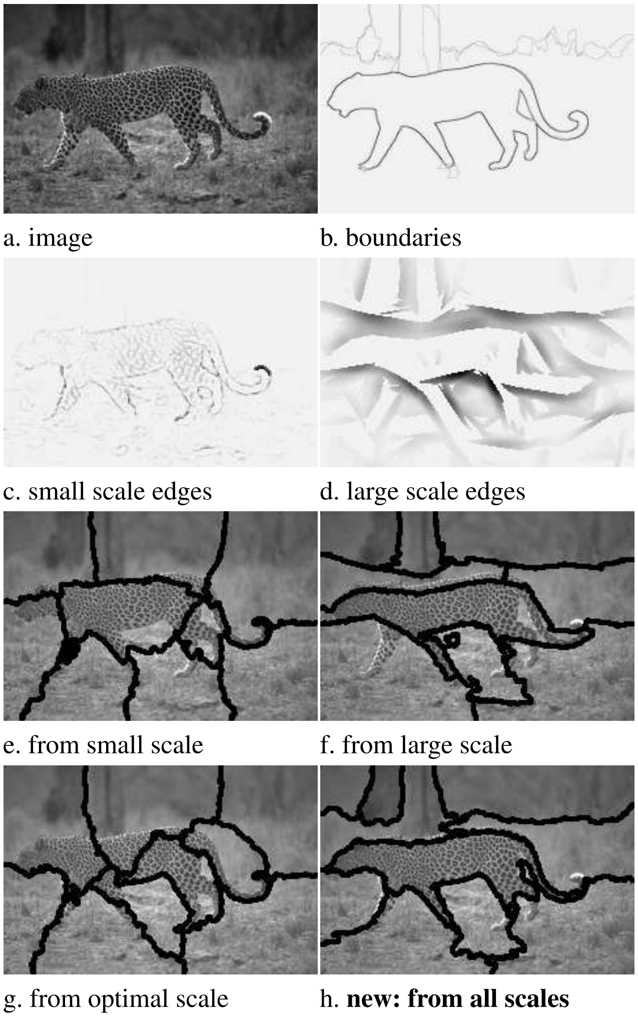
With a sparse yet complete characterization of pairwise pixel affinity, this graph-cuts approach leads to a hierarchy of coarse to fine segmentations that naturally take care of textured regions and weak contours.

1. Introduction

Segmentation is straightforward when edges in the image correspond to real boundaries in the scene. Unfortunately, such an ideal one-to-one correspondence rarely exists in real images(Fig. 1): edges in a textured region do not indicate boundaries, while boundaries may exist at weak contours or between textures that do not give rise to edges.

Inferring boundaries based on low-level edge features is hard; any method could suffer from these massive *false positives* and *false negatives*. However, such a dichotomy is meaningful only when edges are extracted at one scale. When multiple scales are used, there are multiple responses at each pixel location, and they may not agree with each other (Fig. 1c,d).

I'd like to show in this paper that, in fact, the multiscale nature of low-level features holds the promise of solving the two difficulties that segmentation faces. The key ideas are:


a. image b. boundaries

c. small scale edges d. large scale edges

e. from small scale f. from large scale

g. from optimal scale h. **new: from all scales**

Figure 1: Segmentation as resolving ambiguity in correspondence from edges to boundaries. a: image and b: its boundaries marked by human subjects [7]. c,d: edges at two scales. e,f,g: segmentations by normalized cuts [12] based on edges shown in e, f and optimal edges across all scales. h: new result from this work, which utilizes edges at all scales. It groups both big torso and thin tail without losing precision in boundary localization.

1. Both boundaries and edges should not be treated as a binary and single scale phenomenon.
2. There is complementary and useful segmentation information from edges extracted at different scales. Such information is most naturally cast as grouping cues among their pixel supports.
3. If we take reliable grouping cues from each scale and feed all of them to a sensibly designed pixel grouping engine, we can make a global segmentation decision that treat both texture and weak contour problems.

These ideas are in contrast with most current segmentation approaches. Instead of keeping edges at all scales till segmentation, previous works strived to eliminate the scale ambiguity *before* segmentation and use only *one* hopefully better edge map during the segmentation.

For example, [3, 5] tried to determine the optimal scale of each edge by using a winner-take-all mechanism on edge strengths. Such a local decision is often premature, resulting in a segmentation that has both shortcomings: as blind to large regions as that from small scales, and as poor in boundary localization as that from large scale edges (Fig. 1e,f,g). On the contrary, if we use edges at all scales, we can have the best of both worlds (Fig. 1h).

Another line of efforts to improve the final edge map is to fix contours of low contrast. For that, Canny edge detector [1] used hysteresis, [2] and most boundary completion approaches [9, 13] employed curvilinearity. However, the approaches developed to tackle weak contour problems often do not work on texture. Their basic assumption is that smooth contours are likely to be boundaries even when they are weak, which is not valid for textures made of strong and smooth contours. The implication here is that the smoothness of boundaries is not necessarily the cause, but rather a consequence of the distinction between regions that define them. Furthermore, in practice it is difficult to reliably determine local smoothness between contour segments. Pursuing the derived properties in 1D while abandoning the underlying region properties in 2D often makes the original problem unnecessarily hard.

To handle both texture and weak contours, [4] explicitly modeled texture with *textons*, and later [6] learned a classifier to estimate the probability of a pixel location being boundary based on local image evidence. All these measures try to settle onto one edge map before segmentation. Although they do not preclude multiscales, I'd like to show that multiscale edges often suffice as front end computation.

Shown in Fig. 2, the new method here has three steps. Edges are first extracted at multiple scales. Every edge map then leads to a set of pixel grouping cues defined over multiple ranges, each captured in an affinity graph in a graph-theoretic framework. Finally, a simultaneous segmentation through all these graphs is sought to optimize the *average cuts of normalized affinity* at each graph. This procedure yields coarse to fine segmentations where large scale properties are captured first and small details are revealed later.

I will detail the three steps of the method in the next section, followed by experimental results and discussions which demonstrate that such a region segmentation scheme based entirely on multiscale edges can deal with texture and weak contours to a large extent.

2. Method Details

Region segmentation is not to group pixels just based on their feature values. It is also about the spatial arrangements of these feature values. Such a spatial context is naturally described in a relational graph [12, 14]. In this framework, every pixel becomes a node, and the likelihood of two nodes belonging together is captured in a weight attached to the edges linking pixel nodes. Image segmentation then becomes a weighted graph partitioning problem.

There are three steps involved in such an approach.

1. *What features to extract from the image?* I will show that edges at multiple scales contain complementary information and all rather than a subset should be used.

2. *What grouping cues to derive from these features?* I will show that edges should be used to derive the likeness between their pixel supports and a multigrid neighbourhood structure can effectively select reliable grouping cues.

3. *What criterion to use to integrate these grouping cues?* I will propose the *average cuts of normalized affinity* as a new criterion for grouping and show that it not only has desired duality property and elegant numerical solution, but also simplifies cue interaction and integration.

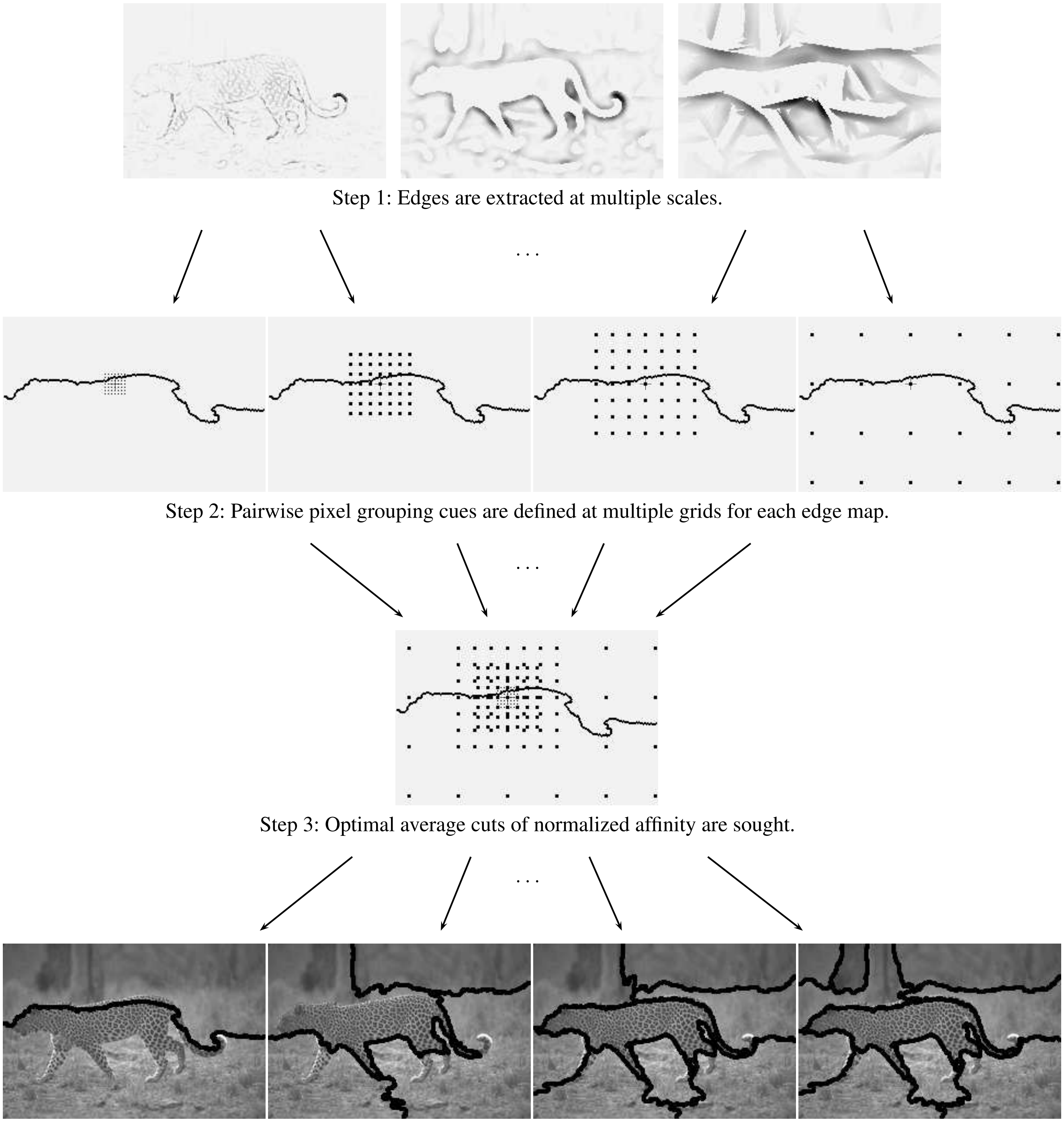
2.1. Image Features: Multiple Edges Maps

Consider filters shown in Fig. 3. They are tuned to detect edges of different shapes, parameterized by $\rho = [\rho_s, \rho_e, \rho_o]$, where ρ_s, ρ_e and ρ_o refer to scale, elongation, and orientation respectively. These quadrature pairs of filters, denoted as $F_o(\rho)$ and $F_e(\rho)$, differ in their spatial phases. The odd-phase filters are essentially the first-order derivatives, whereas the even-phase filters are the second-order derivatives, both smoothed with Gaussians specified by ρ. Given image I, the quadrature energy $E(\rho)$ is defined as [4]:

$$E(\rho) = (I * F_o(\rho))^2 + (I * F_e(\rho))^2, \tag{1}$$

where $*$ is the convolution operator. $E(\rho)$ has a maximum response for contours of shape ρ, whereas the zero-crossings of filter $F_e(\rho)$ locate the positions of the edges.

Traditionally, edges are considered a binary phenomenon and multiple filter responses at a single pixel only serve as a means to derive a better estimate of the edge. This is often done by a competition of the edge energies, which are



Step 1: Edges are extracted at multiple scales.

Step 2: Pairwise pixel grouping cues are defined at multiple grids for each edge map.

Step 3: Optimal average cuts of normalized affinity are sought.

Results: coarse to fine segmentations

Figure 2: Method overview. 1)Edges are first extracted at multiple scales. 2)Edges at each scale define pixel grouping cues at multiple ranges. Each range gives rise to one affinity graph. All pixels have the same number of neighbours in every graph. The difference is that the neighbours are placed on a regular grid with a spacing increasing with the scale of the edges. Shown here is the neighbourhood structure of a marked pixel. The boundaries of a binary segmentation are overlaid on every graph. As the neighbourhood gets larger, more pixels are involved in evaluating segmentation boundaries. 3)A simultaneous segmentation through all these graphs is sought so that the grouping cues at each scale are respected. Computationally it is reduced to partitioning on one equivalent affinity graph and it can be computed efficiently. 4)Final segmentations exhibit a coarse to fine organization, where larger scale properties are foremost respected.

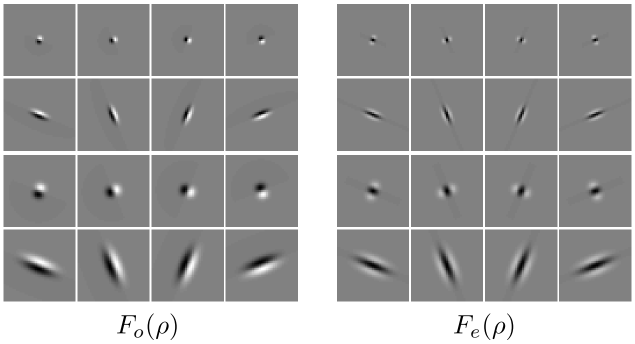


Figure 3: Linear filters of 4 orientations, 2 elongations, 2 scales, in both odd and even phases that form quadrature pairs.

made comparable by normalizing the L_1 norm of each filter [5, 3]:

$$E^* = \max_{\rho} E(\rho), \tag{2}$$

where E^* is the final edge energy from the winning filter.

Although the idea is straightforward, one edge map has trouble recognizing contours at different scales. Already evident in Fig. 1c and d, while small scale edges follow details and curves, large scale edges acknowledge weak contours and ignore texture. The information from small and large filters is complementary (See Table 1), yet when they compete in terms of edge energy, the final "optimal" edge map lose good cues at both scales (Fig. 4).

	small filter response	large filter response
pros	precise localization	insensitive to texture
	closely follow curves	detect weak contours
cons	sensitive to noise	poor localization
	miss weak contours	extend straight lines

Table 1: Edges at different scales encode complementary information and reflect the trade-off between the robustness to detect contrast and the flexibility to follow curves.

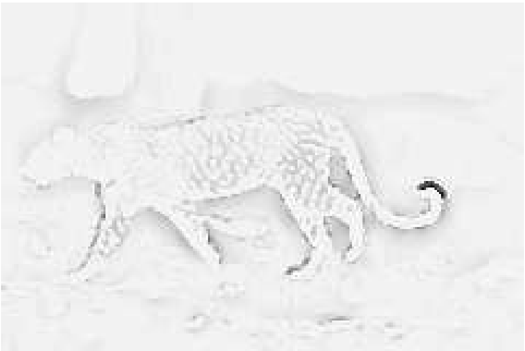


Figure 4: The optimal edge map E^* falls short of representing boundaries at multiple scales. For example, long weak contours detected by large scale filters on the back of the cheetah are gone in the final edge map.

It is not surprising that we need multiple edge maps. Edge detectors are designed for isolated edges [1]. Only in isolation is it meaningful to compute the optimal scale, elongation and orientation of an edge by comparing the response strengths [3, 5]. In real images, however, corners and junctions are ubiquitous, object parts of various scales and shapes co-exist. Such a one-edge-map approach is bound to fall short of representing all of them.

2.2. Grouping Cues: Pixel Affinity at Multigrids

To turn edges into grouping cues, I'd like to generalize the intervening contour idea [4] to multiple edge maps. Two pixels do not like each other if there is an edge encountered on the straight line connecting them. The stronger the edge energy, the less affinity between them.

Formally, for pixels i and j, let $A_{IC}(i, j; \rho)$ be their affinity based on edges at scale ρ_s and elongation ρ_e:

$$A_{IC}(i, j; \rho) = \exp\left(-\frac{\max_{t \in \vec{ij}, \rho_o} E(t; \rho)}{\sigma \cdot \max_{\rho_o} E(\rho)}\right), \tag{3}$$

where $\vec{ij}$ denotes the set of pixels on the line between pixel i and j, and σ is a parameter controlling the sensitivity of affinity to edge energy. Notice that orientation is no longer a parameter of A. In effect, we are dealing with a total of #scales × #elongations edge maps.

Although the above affinity is defined for every pair of pixels in the image, it is not equally reliable everywhere. The range of applicability is lower bounded by the edge localization certainty at that scale, and upper bounded by the grouping principle of proximity.

First, the larger the size of the filter, the less certainty in the localization of an edge. Since contrast information is pooled over a large area, when an edge fires between two closeby pixels i and j, the real cause could lie anywhere within the receptive field of the filter. Therefore, $A_{IC}(i, j; \rho)$ only becomes reliable when i and j are some distance apart.

This certainty distance can be characterized by some dimension of the filter. Let d, which I refer to as the *characteristic distance* of the filter ρ, be defined as:

$$d(\rho) = \rho_s * \rho_e. \tag{4}$$

It will become the unit of distance at affinity graphs derived from the edge map parameterized by ρ.

Secondly, intervening contour, like any local grouping cue, is subject to proximity. Otherwise, the noise in image features gets blown up in the affinity. For instance, two pixels separated by a long weak contour could be mistaken as friends due to the absence of edges at a small scale. This is commonly known as the "leakage problem", the type of problem which a local segmenter is especially sensitive to.

Although a global segmenter is more robust to such noise, we can reduce the noise by only admitting affinity between pixels within a certain neighbourhood.

As the scale of edges gets larger, the number of neighbours increases tremendously. A multigrid neighbourhood is adopted to keep complexity under control. Every affinity graph has N pixel nodes, where N is the total number of pixels in the image I. Every pixel node has $(2r+1)\times(2r+1)$ neighbours, where r is the neighbourhood radius. Edges at each scale give rise to affinity graphs at multiple grids. The grid distance increases with the scale of the edges.

Formally, let W be the weight matrix for the affinity graph parameterized by grid distance g and filter ρ:

$$W(i,j;\rho,g) = A_{IC}(i,j;\rho)\delta(\text{dist}(i,j) = k\cdot g\cdot d(\rho), k \le r), \tag{5}$$

where δ is a binary indicator function (1 if the argument is true and 0 otherwise) and $\text{dist}(i,j)$ denotes the block distance between pixels i and j. As a concrete example, consider neighbourhood radius $r=1$, filter $d(\rho)=2$ and grid $g=1,2$. That is, two affinity graphs are derived from this edge map. In the first graph, each pixel has 9 neighbours spaced at the pixel grid of $[-2,0,2]\times[-2,0,2]$, while in the second graph, its 9 neighbours are spaced at $[-4,0,4]\times[-4,0,4]$, where $(0,0)$ is the pixel itself.

Such a multigrid structure results in a total of:

$$M = \#\text{scales}\times\#\text{elongations}\times\#\text{grids} \tag{6}$$

sparsely connected affinity graphs. Let their weight matrices be denoted by $W_1, W_2, \ldots, W_M$. The affinity a pixel has with its $M\cdot(2r+1)^2$ neighbours gives a complete characterization of its grouping preference.

2.3. Grouping Criterion: Average Cuts of Normalized Affinity

Let $\mathbb{V}$ denote the set of all N pixel nodes. Segmenting these pixels into K groups is to decompose $\mathbb{V}$ into K disjoint sets, i.e., $\mathbb{V} = \cup_{l=1}^{K}\mathbb{V}_l$ and $\mathbb{V}_k\cap\mathbb{V}_l = \varnothing, \forall k\ne l$. Let the K-way partitioning be denoted by $\Gamma_{\mathbb{V}}^{K}$.

From the point of view of each individual node, it agrees with a global segmentation if its grouping preference is mostly satisfied within its own group. The grouping preference of node j toward node i can be measured by the normalized affinity $\text{naff}(i,j;W)$:

$$\text{naff}(i,j;W) = \frac{W(i,j)}{\sum_{i\in\mathbb{V}} W(i,j)}. \tag{7}$$

$\text{naff}(i,j;W)$ is an ego-centric measure and it is not symmetric between i and j. How much j likes(dislikes) group Q can then be measured by the total normalized affinity between j and all the nodes in(outside) Q:

$$\text{like}(Q,j;W) = \sum_{i\in Q}\text{naff}(i,j;W), \tag{8}$$

$$\text{dislike}(Q,j;W) = \sum_{i\in\mathbb{V}\setminus Q}\text{naff}(i,j;W). \tag{9}$$

From the point of view of each group, it agrees with the global segmentation if the average likeness toward the group is high, or the average dislikeness is low. That is:

$$\max\varepsilon(\Gamma_{\mathbb{V}}^{K};W) = \frac{1}{K}\sum_{l=1}^{K}\frac{\sum_{j\in\mathbb{V}_l}\text{like}(\mathbb{V}_l,j;W)}{|\mathbb{V}_l|}, \tag{10}$$

$$\min\bar{\varepsilon}(\Gamma_{\mathbb{V}}^{K};W) = \frac{1}{K}\sum_{l=1}^{K}\frac{\sum_{j\in\mathbb{V}_l}\text{dislike}(\mathbb{V}_l,j;W)}{|\mathbb{V}_l|}, \tag{11}$$

where $|\cdot|$ denotes the cardinality of a set. Since $\text{like}(Q,j)+\text{dislike}(Q,j)=1$, these two criteria are equivalent. I will refer to either of them as the average cuts of normalized affinity criterion. Intuitively, if friends all stay together so that everyone likes his group, then the community would favor the global segmentation.

The above criterion is naturally extended to M weighted graphs defined independently on $\mathbb{V}$:

$$\varepsilon(\Gamma_{\mathbb{V}}^{K};W_1,\ldots,W_M) = \sum_{s=1}^{M}\varepsilon(\Gamma_{\mathbb{V}}^{K};W_s). \tag{12}$$

Next I am going to show that near-global optima of this criterion can be computed efficiently. Following the notation in [15], the average cuts criterion can be written as an optimization program in X:

$$\text{maximize}\quad \varepsilon(X) = \frac{1}{K}\sum_{l=1}^{K}\frac{X_l^TAX_l}{X_l^TX_l} \tag{13}$$

$$\text{subject to}\quad X\in\{0,1\}^{N\times K},\qquad X\,1_K = 1_N, \tag{14}$$

where X is an $N\times K$ binary matrix. $X(i,l)=\delta(i\in\mathbb{V}_l)$ indicates whether pixel i belongs to group l. X^T denotes the transpose of X. 1_n denotes the $n\times 1$ vector of all 1's. A is the total normalized affinity matrix defined as:

$$A = \sum_{s=1}^{M}W_sD_s^{-1}, \tag{15}$$

$$D_s = \text{Diag}(1_N^TW_s), \tag{16}$$

where D_s is the *degree matrix* of W_s, and $\text{Diag}(\cdot)$ denotes a diagonal matrix formed from its vector argument. Note that A is asymmetric.

The above optimization problem can be rewritten as:

$$\text{maximize}\quad \varepsilon(Z) = \frac{1}{K}\,\text{tr}(Z^TAZ), \tag{17}$$

$$\text{subject to}\quad Z^TZ = I_K, \tag{18}$$

where $Z = X(X^T X)^{-\frac{1}{2}}$ is called *scaled partition matrix* [15], tr denotes the trace of a matrix, and I_K denotes the $K \times K$ identity matrix.

Relaxing Z into the continuous domain turns the discrete problem into a tractable continuous optimization problem. The following proposition can be proven using $\mathrm{tr}(A) = \mathrm{tr}(A^T)$ and Lagrange multipliers.

Proposition 1 (Optimal Eigensolution). *The global optimum of $\varepsilon(Z) = \frac{1}{K}\,\mathrm{tr}(Z^T A Z)$ subject to $Z^T Z = I_K$ is achieved by the first K leading eigenvectors of $\bar{A} = A + A^T$, with the optimal objective value as the half of the average of the first K leading eigenvalues:*

$$\varepsilon([V_1, \ldots, V_K]) = \frac{1}{2K} \sum_{l=1}^{K} s_l = \max_{Z} \varepsilon(Z) \tag{19}$$

$$\bar{A} V_l = s_l V_l, \quad s_1 \geq s_2 \geq \ldots, \quad l = 1, \ldots, K. \tag{20}$$

Near-global discrete optima can be computed subsequently using the method described in [15].

2.4. Algorithm

Given an image I, a set of filter parameters ρ's, a set of grid spacing parameters g's, affinity parameter σ, affinity neighbourhood radius r, number of segments K, image segmentation is performed by:

Step 1: Compute edge energy at multiple scales:

$$E(\rho) = (I * F_o(\rho))^2 + (I * F_e(\rho))^2$$

Step 2: Compute pixel affinity at multiple grids:

```
$t = 0$
For $s = 1$ to #scales,
    For $e = 1$ to #elongations,
        $d = \rho_s * \rho_e$
        For $g = 1$ to #grids,
            $t = t + 1$
            For $j = 1$ to $N$,
                For $i$ such that dist$(i, j) = k \cdot g \cdot d$, $k \leq r$,
                    $W_t(i, j) = A_{IC}(i, j; \rho)$.
```

Step 3: Compute average cuts of normalized affinity:

```
$A = W_1 D_1^{-1} + \ldots + W_M D_M^{-1}$
$\bar{A} = A + A^T$
Solve for the first $K$ eigenvectors $V$ of $\bar{A}$
Obtain a discrete segmentation from $V$.
```

3. Experiments and Discussions

The same set of parameters are used for all the 400 images tested: $\rho_s = 1, 3, 5$, $\rho_e = 1, 2, 3$, $\rho = 8$ orientations, $g = 1, 2, 3$, $\sigma = 0.02$, $r = 3$, $K = 10$. Fig. 5 and Fig. 6 are sample results from the Berkeley Segmentation Dataset [7] and Berkeley baseball player dataset [8].

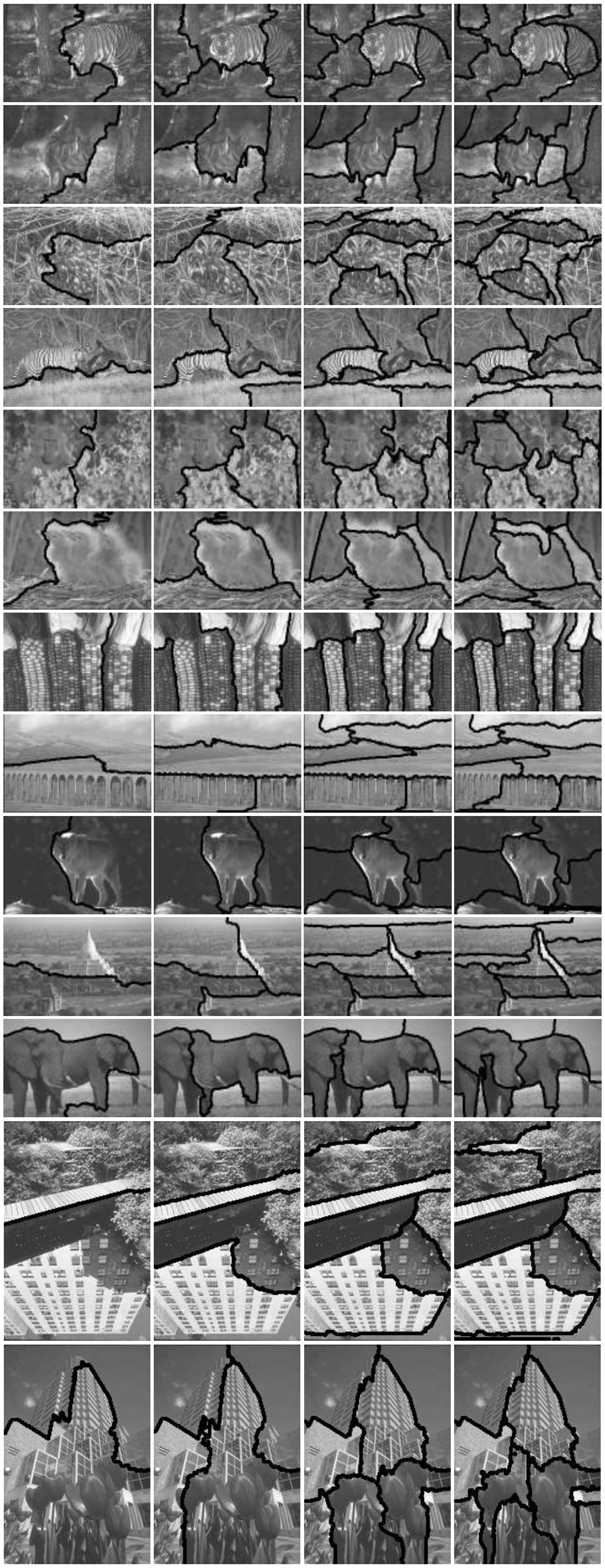


Figure 5: Segmentations with $K = 2, 4, 8, 10$.

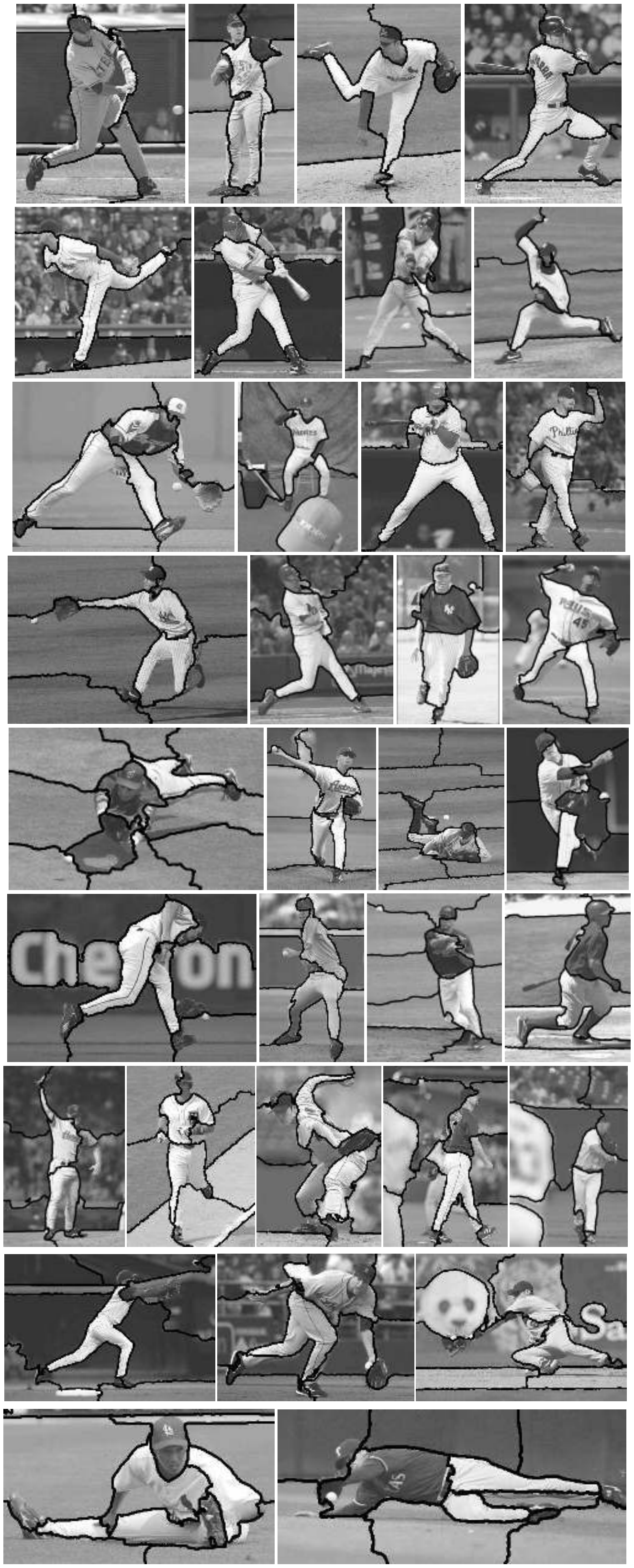


Figure 6: $K = 7$ for rows 1-4 and $K = 10$ for rows 5-9.

Hierarchy of coarse to fine segmentations. As already seen in Fig. 2, these results exhibit coarse to fine segmentations. There is never a case that a binary segmentation picks up a very small region, although the boundaries may not be precise. For example, Row 4 in Fig. 5, the segmentation boundaries do not snap onto the back of the tiger till $K = 8$. Rows 9-10 in Fig. 5 also show that boundaries of weak contrast are automatically completed at a fine level segmentation. The reasons are: 1) affinity at larger scales only cares about grouping relationships at a larger distance; 2) cuts on these graphs involve the grouping preference on more nodes as they are connected over a larger neighbourhood. In other words, priority is given to the goodness of grouping at coarser scales. Thus fine details, captured in small scale affinity graphs, are honored later.

Handle texture and weak contours. Such a coarse to fine segmentation mostly results from the multiscale affinity at long ranges. [2, 4, 6] have also studied the problem of obtaining better affinity measures. They focused on richer image features. The multiscale nature of grouping cues has not been explored. The results here show that in the multiscale integration framework, a segmenter using entirely edges can handle most texture and weak contours in real images. Explicit extraction of textons may not be needed. Fixing edge maps for weak contours may not be needed.

The average cuts of normalized affinity criterion allows cue integration and offers numerical advantage over the normalized cuts criterion. Although both criteria share duality and normalization of the affinity, the normalized cuts criterion [12, 14] does not handle simultaneous cuts through multiple graphs. It cannot be reduced to normalized cuts on a single weight matrix. Therefore, even if we could extend its definition to multiple graphs as in Eqn. (12), there is no simple numerical solution.

The difference between the two criteria is illustrated in Fig. 7. At the same level of segmentation, average cuts have more precise boundaries and are better at grouping regions of multiple scales. Such differences are typical and can be understood from their definitions. Average cuts of the normalized affinity have a clear interpretation of optimizing the goodness of grouping at each scale. Fine details which do not conflict with coarser scale grouping cues are encouraged to come out at a coarse level segmentation. On the contrary, normalized cuts of the normalized affinity treat neighbours at all scales altogether, thus small details that only influence the grouping of a few nodes are too insignificant to be acknowledged at a coarse level segmentation.

Simplified multiscale interactions. How to integrate grouping cues at multiple scales has not been well studied. A related work is [11], but it is more concerned with the speed and the granularity of low-level segmentation. A common belief is that cues at coarse scales tend to prime the grouping behaviour at fine scales and a hierarchical prece-

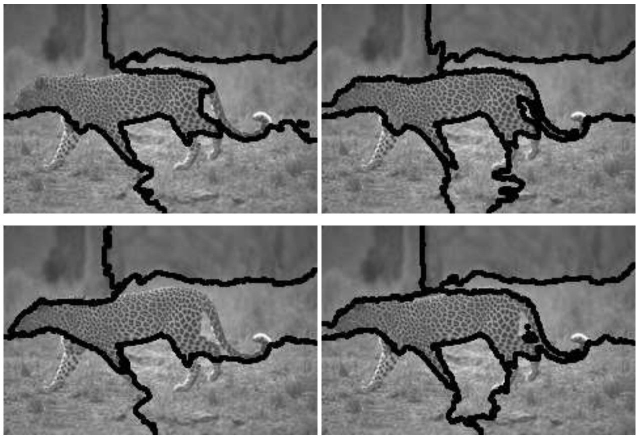


Figure 7: Segmentations from the average cuts (row 1) and the normalized cuts (row 2) of $\bar{A}$ respectively. $K = 5, 7$. Average cuts follow curvy boundaries better and also allow the thin tail to emerge together with the big torso, whereas normalized cuts can only take care of big structures at coarse level segmentation.

dence is often assumed [10]. This work suggests a simple alternative: to make the best use of cues across scales, just let them contribute within their most reliable ranges.

Numerically fast and efficient. For N pixels, M graphs, $B = (2r + 1)^2$ neighbours, $W_1 \ldots, W_M$ altogether have $N \cdot M \cdot B$ nonzero entries. The number of connections is linear with respect to N, yet they contain long range cues as far as e.g. $r \cdot \max g \cdot \max d(\rho) = 3 \cdot 3 \cdot 15 = 135$ pixels away! A further reduction is achieved by choosing the top few best friends. All results here are obtained with $B = 10$, and the total running time in MATLAB on a PC with 2GHz CPU and 2GB memory is about 100s for those in Fig. 5 (size: 160×240) and 600s for Fig. 6 (size: 400×400). Two factors contribute to the speed. One is the multigrid technique that keeps the connections sparse. The other is that affinity at all ranges makes the eigensolution converge faster. In short, sparse yet complete cues make grouping easier.

Finally, I'd like to conclude the paper with results on two extreme images that illustrate the problems of texture and weak contours (Fig. 8), motivating future research on multiscale, texture segmentation and boundary completion.

Acknowledgments

The author thanks Jitendra Malik, Andras Ferencz, Andrea Frome, Erik Learned-Miller and Deva Ramanan for comments and DOD/ONR-MURI #FD N00014-01-1-0890 grant for financial support.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–98, 1986.

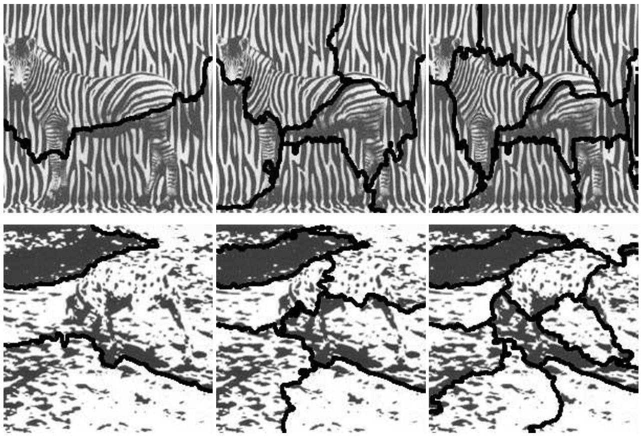


Figure 8: Multiscale, texture and illusory contours.

[2] T. Leung and J. Malik. Contour continuity in region based image segmentation. In *European Conference on Computer Vision*, pages 544–59, 1998.

[3] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–54, 1990.

[4] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 2001.

[5] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of Optical Society of America*, 7:923–32, 1990.

[6] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using brightness and texture. In *Neural Information Processing Systems*, 2002.

[7] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[8] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, 2004.

[9] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Math.*, pages 577–684, 1989.

[10] X. Ren and J. Malik. A probabilistic multi-scale model for contour completion based on image statistics. In *European Conference on Computer Vision*, pages 312–27, 2002.

[11] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *CVPR*, pages 70–7, 2000.

[12] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[13] L. R. Williams and D. W. Jacobs. Local parallel computation of stochastic completion fields. *Neural computation*, 9(4):859–81, 1997.

[14] S. X. Yu. *Computational Models of Perceptual Organization.* PhD thesis, Carnegie Mellon University, May 2003.

[15] S. X. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, Nice, France, 11-17 Oct 2003.