Segmentation Induced by Scale Invariance

Stella X. Yu
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720-1776, USA

Abstract

Perceptual organization is scale-invariant. In turn, a segmentation that separates features consistently at all scales is the desired one that reveals the underlying structural organization of an image. Addressing cross-scale correspondence with interior pixels, we develop this intuition into a general segmenter that handles texture and illusory contours through edges entirely without any explicit characterization of texture or curvilinearity. Experimental results demonstrate that our method not only performs on par with either texture segmentation or boundary completion methods on their specialized examples, but also works well on a variety of real images.

1. Introduction

The task of image segmentation is to organize pixels into regions of homogeneous features based on measurements taken from intensity values. This problem is easy if adjacent regions have different but uniform intensity (areas B, W in Fig. 1a): compute contrast and declare edges as boundaries. However, this idea of *edges-for-boundaries* runs into trouble in areas T and C in Fig. 1a: T has numerous edges *within* the region, whereas C has no edges *along* the boundary. The goal of this paper is to show that, by formalizing the single notion of scale invariance for segmentation, we can develop a general segmenter that is based entirely on edges yet able to deal with both texture and illusory contours.

Texture and illusory contours, present in the same image as in Fig. 1a, have often been studied in isolation using typical images as in Fig. 1b,c. Clearly, texture segregation requires pooling features over a local window in order to suppress within-region edges [3, 2], while contour completion requires high-precision edge localization in order to complete gaps along the boundaries [12]. These requirements do not function at the same scale. Consequently, boundary completion methods are easily confused by massive texture edges (Fig. 1b:Canny), while texture segmentation methods are oblivious to sharp contrast at boundaries (Fig. 1c:Pb). Furthermore, both methods have trouble producing satisfactory results on what they are designed for: determining the pooling window size for texture (Fig. 1b:Pb) and edge filter size for illusory contours (Fig. 1c:Canny) is crucial to their success, yet estimating the optimal size locally is nontrivial if not impossible [13].

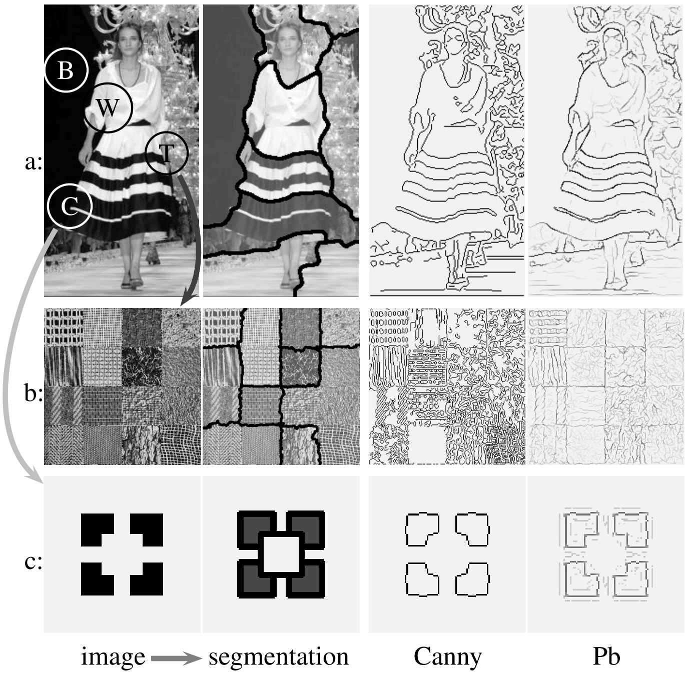


Figure 1. Segmentation should handle both texture and illusory contours. For a: a general image (313 × 166), b: a typical texture image (256 × 256), c: a typical illusory contour image (100 × 100), columns 2-4 show our results in comparison with those from two local boundary detectors: Canny and probabilistic boundary detectors [9]. Default parameters in MATLAB implementations were used. By design, Canny uses hysteresis to complete boundary gaps without considerations for texture, while Pb uses textons to suppress texture edges without considerations for illusory contours. Furthermore, their performance on what they are designed for is critically dependent on local scale estimation, which is nontrivial.

Some universal approaches to segmentation have also been proposed [8, 11]. Their idea for reconciling the dilemma posed by texture and illusory contours is to explicitly model and monitor both of them, determine which phenomenon dominates and then choose the right segmentation tool to apply. For a generative approach such as [11], it remains to be seen whether it is applicable and practical for general images other than those that fit a small number of constituent models. For a discriminative approach such as [8, 4, 9], as can be seen in Fig. 1:Pb, employing more features does not always help: texture is not fully suppressed and boundaries could be hallucinated in simple images.

Therefore, explicitly monitoring texture and contours not only wastes computation, but also introduces cross interference against each other. The reason why a local method which attempts an ever complex operation to suppress texture and promote weak contours can hardly succeed is that both texture and contours are intrinsically multiscale phenomena, and they are best dealt with at a global integration level. This is the alternative explored in this paper.

The key insight is simple: the perceived organization of an image remains largely intact when the image is scaled (Fig. 2). In turn, if a segmentation satisfies this necessary condition of scale invariance, that is, it gets a good score at each scale, then it probably *is* the grouping that we desire.

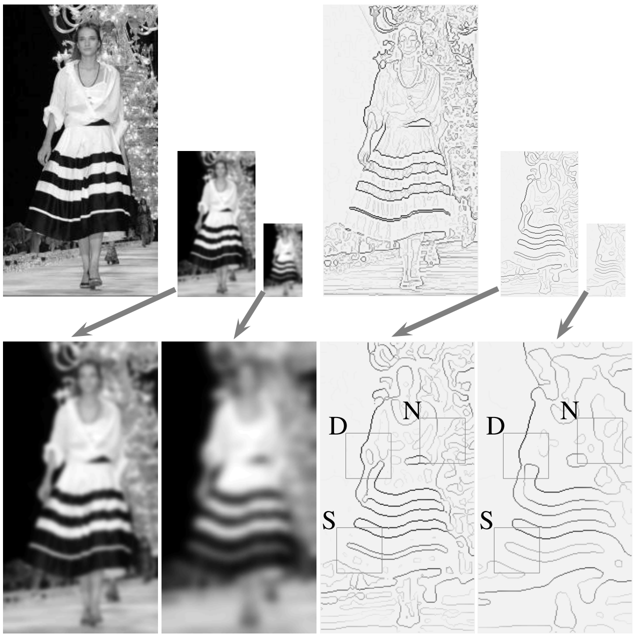


Figure 2. Scale invariance of segmentation. Row 1: Perceived grouping of an image remains stable despite that pixels and edges change their identities across scales. Row 2: A key issue in scoring a segmentation across scales is correspondence, since as scale increases, pixels become fuzzier, and edges could shift locations(S), disappear(D) or newly emerge(E).

Scale invariance of grouping is often illustrated by the Gestalt law of proximity, where equal scaling of horizontal and vertical separations of a dot array is shown to have little effects on the perceived grouping [7]. However, such invariance to global scaling is trivial: the elements to be grouped, *the dots*, remain in one-to-one correspondence over scales, and the feature used in grouping, *the distance*, remains consistent and indicative of the same grouping over scales.

What is truly remarkable about the scale invariance of perceptual organization is that the perceived grouping of an image remains stable despite the fact that pixels at different scales designate different grouping elements and the features that we are able to extract also change characteristics across scales (Fig. 2). A coarse-scale pixel denotes a local assembly of fine-scale pixels. However, it does not represent them equally well. Furthermore, edges extracted at a coarse scale could shift locations from their fine-scale counterparts, disappear or emerge simply as new features that do not have a counterpart at fine scales.

Hence, to evaluate a grouping at multiple scales, pixel correspondence must be established. Observe that both aforementioned deformation problems are most severe for border pixels, while interior pixels often sustain their correspondence over scales. Our idea is thus to find interior pixels at each scale and use them to relate to the next scale. They often appear as medial axes [5] at various scales.

Roughly speaking, as scale increases, texture evens out, while boundary gaps become smaller and thus easier to complete. A further insight is illustrated in Fig. 3. The homogeneity of local statistics has always been the property used to tackle texture segmentation [3, 2], whereas contour continuity has always been the property used to tackle

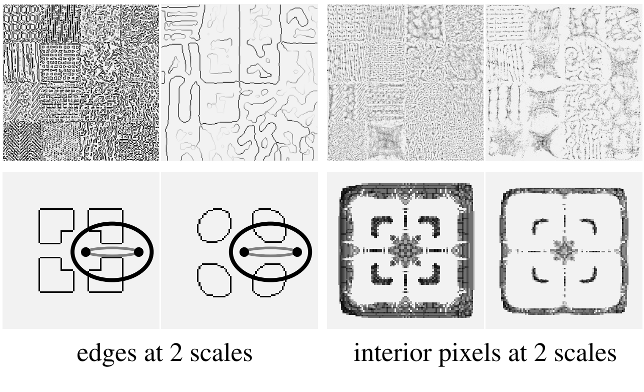


Figure 3. Why a multiscale framework can deal with both texture (Fig. 1b) and illusory contours (Fig. 1c). Row 1: Texture boundaries are revealed in multiscale edges which consistently separate interior pixels at their corresponding scales. Row 2: Boundary gaps are completed for conforming to grouping cues obtained with larger apertures. Lack of edges in a small surround (thin outlines) tends to group the two marked pixels, but configuration of edges in a larger surround (thick outlines) tends to segregate them.

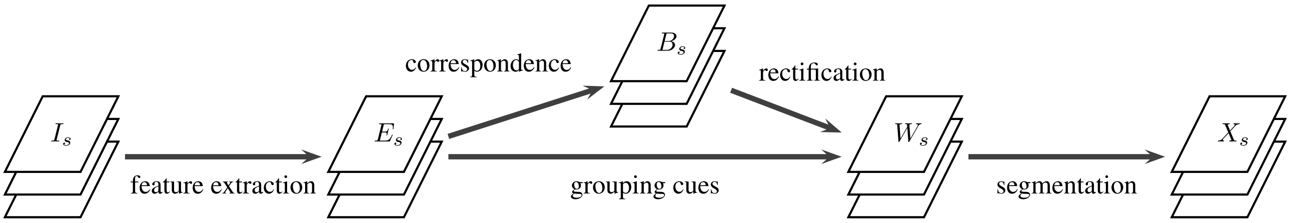


Figure 4. Algorithm overview. I_s: Multiscale versions of the original image are computed. E_s: Edges are extracted at each scale. B_s: Interior pixels are subsequently computed. W_s: Grouping cues that are originally defined on coarse-scale pixels from coarse-scale edges are rectified for original pixels. X_s: Optimizing the total goodness of grouping based on W_s leads to a hierarchy of coarse-to-fine segmentations. These are the segmentations that respect scale invariance, with texture and illusory contours naturally taken care of.

boundary completion [10, 12], yet both of them result naturally when an image is examined across scales.

With refinement, the *edges-for-boundaries* doctrine is correct after all: boundaries manifest themselves in multiscale edges that separate interior pixels consistently, a characteristic that texture edges lack while illusory contours possess as well as ordinary boundaries do.

Illustrated schematically in Fig. 4, our approach formulates scale invariance as an optimization of one type of grouping criterion, based on one type of grouping cue, from one type of feature, i.e., edges. Fig. 1 shows our segmentation results on typical texture and illusory contour images.

The rest of the paper details how we extract features, find the correspondence, derive grouping cues in multiple apertures, rectify them for original pixels, and finally integrate them to discover segmentations that are consistently good over scales. Our results on a variety of images demonstrate that a universal segmenter based on entirely edges is possible without explicitly modelling texture or curvilinearity.

2. Model

A straightforward implementation of scale-invariant segmentation would be: given an image, generate its scaled versions; score a segmentation on every scaled image; search over all segmentations and pick the one that gives the best total score. Four issues are involved.

1. *What features to extract from the image?* We will use one type of feature, that is, edges.

2. *What grouping cues to derive from these features?* We will use one type of grouping cue, that is, pixel affinity from elongated intervening contours.

3. *How to relate segmentations across scales?* We will derive data-driven correspondence between pixels across scales, through which goodness of grouping at coarse scales can be projected back to the original image and be summed.

4. *How to find the optimum of the total goodness efficiently?* We will formulate the segmentation problem in graph theory as a simultaneous cut through a stack of graphs, each capturing the goodness of grouping at a scale. Near-global optima can then be computed efficiently.

2.1. Feature: Multiscale Edges

A coarse-scale image is often obtained by Gaussian smoothing followed by subsampling. Subsampling has two advantages: reduce redundancy and allow the same feature extractor at each scale. However, sufficient smoothing is required in order not to introduce spurious features.

Proposition 1. *To avoid aliasing while subsampling a discrete image with period T following Gaussian smoothing, the Gaussian must have a standard deviation $\sigma \geq T$.*

Proof. According to the Sampling Theorem, an image can be represented by its samples of period T if it is band-limited to the Nyquist frequency $\frac{\pi}{T}$. Since Gaussian(σ) in space domain corresponds to Gaussian($\frac{1}{\sigma}$) in frequency domain, assuming that Gaussian(σ) is practically zero outside the $\pi\sigma$ range, any image convolved with Gaussian(σ) is band-limited to $\frac{\pi}{\sigma}$. Therefore, $\frac{\pi}{\sigma} \leq \frac{\pi}{T}$, or $\sigma \geq T$. □

Therefore, to reduce an image by half, $\sigma = 2$ should be used. The often used filter $[1, 4, 6, 4, 1]$ in constructing a Gaussian pyramid [1] approximates Gaussian of $\sigma = 1.1$ and can cause aliasing. Aliasing in subband images is not an issue for signal coding, but should be avoided when edges are further extracted to be correlates of region boundaries.

Shown in Fig. 5, we first build a Gaussian pyramid with $\sigma = 2$ and subsampling period of 2. An edge pyramid is obtained by applying the same set of quadrature filters F_o

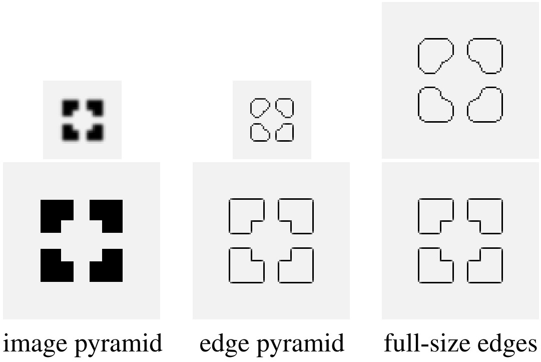


Figure 5. Extract full-size multiscale edges E_s through pyramids.

and F_e [8] to each level of the pyramid. The edge strength is measured by quadrature energy E:

$$E = (I * F_o)^2 + (I * F_e)^2. \quad (1)$$

Upsampling the edge images brings them back to full size. We denote each by E_s, where s is the scale.

2.2. Cue: Elongated Intervening Contours

Intervening contours (IC) between two pixels, p and q, refer to edges that intersect the line connecting them (Fig. 6). The stronger the IC, the less likely that the two pixels are in the same region [8]. We define pixel affinity:

$$A_{IC}(p,q) = \exp\left(-\frac{\max_{t\in \text{line}(p,q)} E(t)}{\sigma_{IC}}\right). \quad (2)$$

This definition effectively encodes pixel intensity similarity (through E), convexity (through line(p,q)) and boundary closure (through region segmentation on pixels).

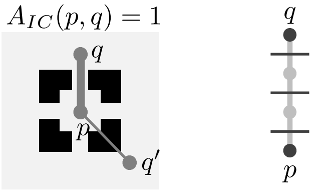


Figure 6. Pixel affinity A_{IC} only looks at local edges cutting the line connecting two pixels. It is susceptible to missing edges.

Just as image features should be extracted at multiple scales, grouping cues from features should also be examined at multiple scales. Illustrated in Fig. 3, the key to illusory contour completion is to capture pixel grouping relationships within multiple apertures. We rationalize the intuition with elongated intervening contours (Fig. 7), log-polar neighbourhood (Fig. 8), and pullback affinity (Fig. 9).

First, to derive desired pixel grouping cues in the presence of illusory contours, edges should be lengthened as distance(p,q) increases (Fig. 7). We formalize the pixel affinity from elongated intervening contours (EIC) using a recursive definition on a *kernel*. The kernel $K(p,d)$ denotes the affinity between p and its *immediate* neighbour $p+d$ along direction d. For example, as in Eqn. (2), edges at scale s define the following kernel $K_E(s)$:

$$K_E(p,d;s) = \exp\left(-\frac{\max(E_s(p), E_s(p+d))}{\sigma_{IC}}\right). \quad (3)$$

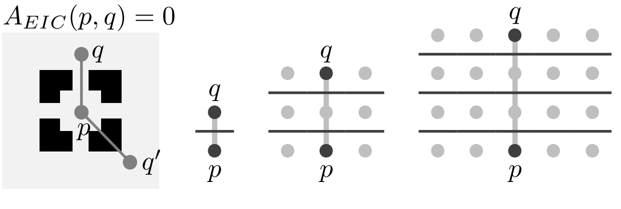


Figure 7. Pixel affinity A_{EIC} looks at increasingly lengthened edges that cut through line(p,q) as distance(p,q) increases. It is resistant to missing edges along illusory contours.

The EIC affinity from kernel K is defined by:

$$K^1(p,d) = K(p,d), \quad (4)$$

$$K^n(p,d) = \min_{q\in\{p-d^\perp, p, p+d^\perp\}} K^{n-1}(q,d), \quad (5)$$

$$A_{EIC}(p, p+n\cdot d; K) = \min_{t=0:n-1} K^n(p + t\cdot d, d), \quad (6)$$

where $d^\perp$ is the direction perpendicular to d. $K^n(p,d)$ is the affinity between p and $p+d$ based on EIC of length $2n-1$, whereas A_{IC} assumes $K^n = K$. Note that the recursion is performed independently along each direction.

Second, to take advantage of the redundancy introduced by the edge lengthening, we only need pixel affinity for log-polar neighbours. This efficiently captures a pixel's grouping preference in a large neighbourhood (Fig. 8).

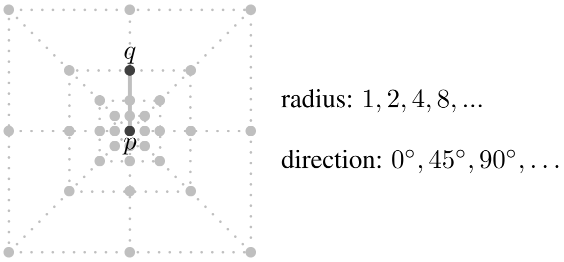


Figure 8. Log-polar neighbourhood (of the center pixel p).

The EIC affinity encodes curvilinearity by carving the grouping relationships among the pixels that give rise to boundaries. In Fig. 9a, a pixel has large affinity with the adjacent pixel across an illusory contour, but little affinity with far neighbours due to blockage from nearby edges.

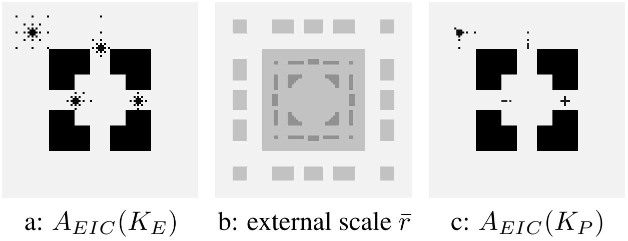


Figure 9. Pullback affinity. a, c show the affinity fields at the same four pixels overlaid on the image. b: external scale at each pixel, lighter gray for larger values. K_P is derived by pulling $A_{EIC}(K_E)$ values at radius $\bar{r}$ back to radius 1. Pullback affinity $A_{EIC}(K_P)$ propagates the kernel K_P to the full neighbourhood.

Third, we can thus further improve the accuracy of pixel grouping cues by allowing each pixel to pull its affinity with far neighbours back to its immediate neighbours. The distance at which we pull back the affinity depends on some measure of the external scale [6] of the pixel. It is defined as the smallest radius beyond which $A_{EIC}(K)$ is all trivial:

$$\bar{r}(p) = \min\{r : A_{EIC}(p, p+(r+1)\cdot d; K) < \theta, \forall d\}, \quad (7)$$

where θ is a threshold that indicates an affinity value trivial. $K = K_E$ if $s = 1$; $K = K_B$ otherwise (defined in the next section). Note that A_{EIC} is a non-increasing function of r for any direction d. The *pullback kernel* K_P is defined by the affinity with far neighbours at external scales:

$$K_P(p, d) = A_{EIC}(p, p + \bar{r}(p) \cdot d; K). \quad (8)$$

Shown in Fig. 9c, K_P not only assimilates the external scale of individual pixels, but also robustly captures their non-isotropic preferences. Since the pullback kernel reflects pixel grouping cues seen from a larger aperture, the subsequently propagated affinity $A_{EIC}(K_P)$ is more effective at fulfilling contour completion requirements.

Our pullback affinity can be viewed as deblurring illusory contours. The external scale $\bar{r}$ adaptively determines the extent of blurring and the pullback affinity $A_{EIC}(K_P)$ reflects the sharpened boundaries after deblurring.

2.3. Pixel Correspondence across Scales

When an image is smoothed, pixels become fuzzier, corners rounded and edges shifted. A pixel originally inside a corner could become outside a corner (Fig. 5). Clearly, grouping cues for fuzzy pixels cannot be directly transferred to original pixels, except for those inside regions. We thus use interior pixels as correspondence hubs, i.e., for an interior pixel, we use the coarse-scale pixel at the same location to represent it at that scale; for a border pixel, we use the coarse-scale representative of its best interior friend.

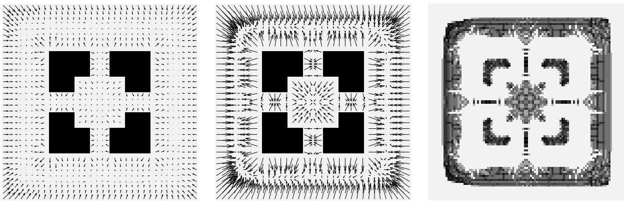
We rationalize the *best interior friend* as the center of the mass of a pixel's affinity field. It stays at the same location for an interior pixel, while shifting towards the region interior for a border pixel. Formally, for pixel p at scale s, let $B_{s,t}(p)$ be its corresponding best interior friend at a coarser scale t. We iterate the center of mass computation n times ($n = 5$ is used) to obtain one-step forward mapping $B_{s,s+1}$, and concatenation gives the mapping from scale 1 to s:

$$B^0_{s,s+1}(p) = p, \quad (9)$$

$$B^n_{s,s+1}(p) = \frac{\sum_q A_{EIC}(p, q; K_E(s)) B^{n-1}_{s,s+1}(q)}{\sum_q A_{EIC}(p, q; K_E(s))}, \quad (10)$$

$$B_{1,s+1} = B_{s,s+1} B_{1,s}, \qquad B_{s,s+1} = B^5_{s,s+1}. \quad (11)$$

Fig. 10 illustrates the evolution of the forward mapping and the emergence of medial axes as best interior friends.



a: $B^0_{1,2}$ b: $B^5_{1,2}$ c: interior pixels

Figure 10. Pixel correspondence is established by mapping pixels to their best interior friends. a,b: Forward mapping at the first and the final iteration. c: Number of votes each pixel gets as the best interior friend. The darker the more votes (more medial-axis like).

With $B_{1,s}$, we define *backprojected* kernel K_B for the affinity between original pixels based on scale-s edges:

$$K_B(p, d; s) = A_{EIC}(B_{1,s}(p), B_{1,s}(p+d); K_E(s)). \quad (12)$$

Note that $K_B(p, d; s)$ is defined for pixels at a distance of 2^{s-1} pixels apart (Fig. 11), since coarse-scale edges are only meaningful at that resolution [13]. The convexity implied by EIC is preserved in $A_{EIC}(K_B)$, since it is propagated from K_B to the full neighbourhood.

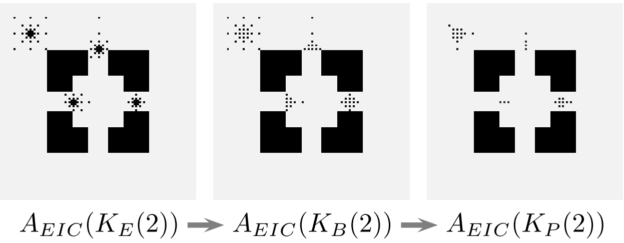

$A_{EIC}(K_E(2)) \rightarrow A_{EIC}(K_B(2)) \rightarrow A_{EIC}(K_P(2))$

Figure 11. Pixel affinity from coarse-scale edges. For the same four pixels in Fig. 9, here are their affinity fields for scale-2 pixels, their projection to scale-1 pixels, and then the pullback version at scale 1. Note that neighbour distance increases to 2 at scale 1.

The backprojected affinity $A_{EIC}(K_B)$ is more accurate for both texture and weak contours. The reason for the former is that pixels originally separated by texture edges tend to have large affinity between their coarse-scale representatives, while pixels across illusory contours are mapped farther away and thus more likely to be segregated by EIC. When examined across scales, pixel grouping cues based entirely on edges are capable of both suppressing texture and promoting illusory contours.

Our rectification from $A_{EIC}(K_E)$ to $A_{EIC}(K_B)$ is essentially encoding coarse-scale dominant structures without losing original pixel resolution via a denoising-deblurring procedure: $A_{EIC}(K_E)$ accentuates coarse-scale feature details as in denoising, whereas the backprojected version $A_{EIC}(K_B)$ sharpens the boundaries as in deblurring.

To recapitulate, edges at scale s lead to EIC affinity $A_{EIC}(K_E)$ between pixels at scale s. It is first rectified to $A_{EIC}(K_B)$ for pixels at scale 1, and then pulled back to generate $A_{EIC}(K_P)$ (Fig. 11). If we have a total of n_s scales, we end up with a stack of $2n_s$ pixel affinity measurements all defined on the original pixels. We denote them by matrices $W_t, t = 1 : M, M = 2n_s$.

2.4 Grouping Criterion and Its Solution

To score a grouping with pairwise pixel affinity, we adopt the criterion of average satisfaction that k groups assume:

$$\text{goodness of grouping} = \frac{1}{k}\sum_{l=1}^{k} \frac{\sum_{p \in \text{group } l} \text{satisfaction}(p)}{\text{size of group } l} \tag{13}$$

$$\text{satisfaction}(p) = \frac{\sum_q \text{affinity}(p, q) : p, q \text{ in one group}}{\text{total affinity } p \text{ has}}. \tag{14}$$

The normalization by the size of each group is essential for promoting size balance. Otherwise, a trivial grouping that isolates a lone pixel from the rest would be highly desirable.

Given previously obtained affinity W_t, $t = 1 : M$, we evaluate a segmentation using:

$$\text{total score} = \sum_{t=1}^{M} \text{goodness of grouping from affinity } W_t. \tag{15}$$

Such a criterion has been considered in [13]. It can be described in a graph-theoretic framework as a simultaneous cut across a stack of graphs with weight matrices $\{W_t\}$. The optimal solution in the relaxed continuous domain is shown to be the eigenvectors corresponding to the k largest eigenvalues of W, where

$$W = \sum_{t=1}^{M} W_t D_t^{-1} + D_t^{-1} W_t, \tag{16}$$

and D_t is a diagonal matrix with $D_t(p, p) = \sum_q W_t(p, q)$. Ideally, pixels in the same group are mapped to similar values in the continuous eigensolutions (Fig. 12).

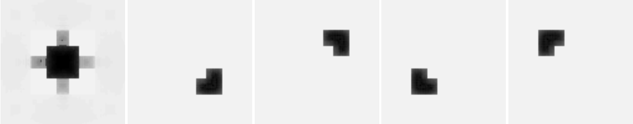


Figure 12. The first 5 eigenvectors. Thresholding these images with mean values gives the center illusory square and 4 L-shapes.

In addition to encoding the right cues for texture segmentation and contour completion in multiscale affinity, the integration criterion which encapsulates scale invariance through Eqn. (13) to (15) helps to bring out the desired segmentation. Here, the satisfaction of individual pixels is defined by the proportion of affinity contained within its group, so that the total affinity a pixel has (*rich or poor*) does not matter. This boosts in the total score the significance of pullback affinity and backprojected coarse-scale affinity, which often have fewer but more accurate connections. For example, consider a pixel near an illusory contour in Fig. 9. $A_{EIC}(K_E)$ dominates $A_{EIC}(K_P)$ in terms of total affinity. However, as the total affinity is discounted, separation from its neighbours across the illusory contour induces a $\%100$ satisfaction for $A_{EIC}(K_P)$, driving the optimal segmentation to occur along illusory contours.

2.5 Algorithm

Given affinity parameters σ_{IC}, neighbourhood radius r, significance threshold θ, number of scales n_s, number of segments k, segmentation of image I is done in 3 steps.

```
Step 1: Compute full-size edges at multiple scales:
  $I_s = \text{subsample}(I_{s-1} * \text{Gaussian}(2), 2), I_1 = I, s = 2 : n_s$
  $E_s = \text{upsample}((I_s * F_o)^2 + (I_s * F_e)^2, 2^{s-1}), s = 1 : n_s$

Step 2: Compute pixel grouping cues at multiple scales:
  $t = 0; B_{1,1}(p) = p, \forall p$
  For $s = 1 : n_s$,
    Compute affinity kernel from edges at scale $s$
      $K_E(p, d; s) = \exp\left(-\frac{\max(E_s(p), E_s(p+d))}{\sigma_{IC}}\right), \forall p, d.$

    Find correspondence by one-step forward mapping
      $B^0_{s,s+1}(p) = p, \forall p$
      $B^n_{s,s+1}(p) = \frac{\sum_q A_{EIC}(p,q;K_E(s)) B^{n-1}_{s,s+1}(q)}{\sum_q A_{EIC}(p,q;K_E(s))}, n = 1 : 5$
      $B_{1,s+1} = B_{s,s+1} B_{1,s}, \qquad B_{s,s+1} = B^5_{s,s+1}.$

    Rectify affinity by backprojection, with $d$ scaled by $2^{s-1}$
      $K_B(p, d; s) = A_{EIC}(B_{1,s}(p), B_{1,s}(p + d); K_E(s)), \forall p, d$
      $W_t = A_{EIC}(K_B(s)), t = t + 1$

    Derive pullback affinity at scale 1:
      $\bar{r}(p) = \min\{r : A_{EIC}(p, p + (r + 1)d; K_B(s)) < \theta, \forall d\}, \forall p$
      $K_P(p, d; s) = A_{EIC}(p, p + \bar{r}(p) \cdot d; K_B(s)), \forall p, d$
      $W_t = A_{EIC}(K_P(s)), t = t + 1.$

Step 3: Compute graph cuts:
  $W = \sum_{t=1}^{M} W_t D_t^{-1} + D_t^{-1} W_t$
  Solve for the first $k$ eigenvectors $V$ of $W$
  Obtain a discrete segmentation from $V$ (e.g. [14]).
```

3. Results and Discussions

We implement the algorithm in MATLAB. $\sigma_{IC} = 0.02$, $r = 2^5$, $\theta = 0.3$, $k = 10$, $n_s = 4$, and quadrature filters of 8 orientations are used for all our images.

In Fig. 13 and Fig. 14, we compare results from two advanced edge detectors: Canny and Pb as in Fig. 1, and two

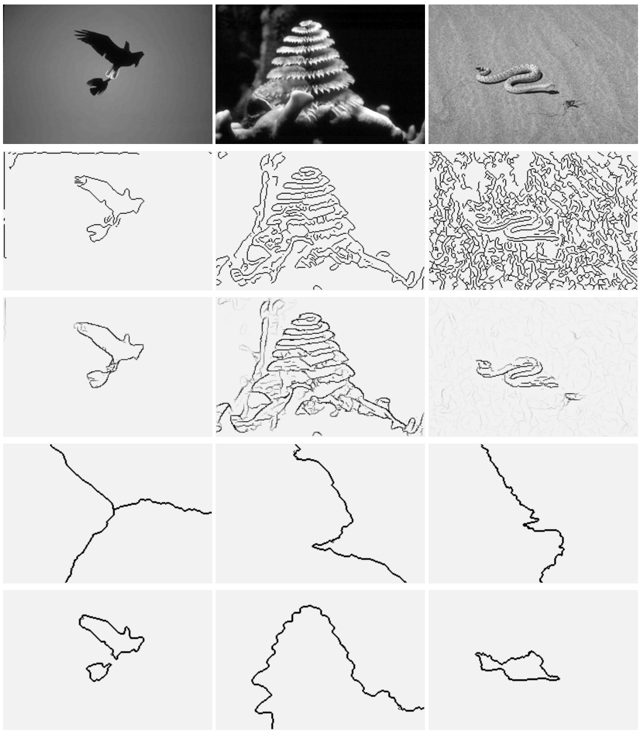


Figure 13. Comparison of results on BSDS images [9]: Canny, Pb, previous work [13] and this work. For the latter two, the number of segments is manually chosen in order to outline objects of interest with the fewest segments. Canny and Pb can only tell pixels on/off a boundary. There is no sense of pixel grouping or differentiation of boundary scales. Texture is not fully suppressed. These images have *gaps* and *elongated shapes* that cause conflicting cues at different scales. Without cross-scale consistency, previous work easily gets confused; with cross-scale consistency, this work is able to discover the major structures characterized by scale invariance.

region segmenters: previous work, which uses multiscale edges but without addressing gap completion or cross-scale consistency [13], and this work. These results demonstrate that: 1) at the feature level, texture edges cannot be fully suppressed, nor can contour gaps be correctly completed, despite the best efforts to estimate optimal local scales; 2) employing implicit gap completion and enforcing cross-scale consistency, our edge-based approach can segment out large scale-invariant regions without losing fine details of their boundaries.

We can collapse our coarse-to-fine boundary maps into a probabilistic boundaryness measure (Fig. 15) and benchmark our segmenter as in [9, 4]. Higher precision but lower recall is expected over a local detector (e.g. Canny or Pb in Fig. 1), since our global region segmenter finds big pieces first (Fig. 15, Fig. 16).

These are low-level image segmentations based entirely on intervening contours, which encode Gestalt laws of in-

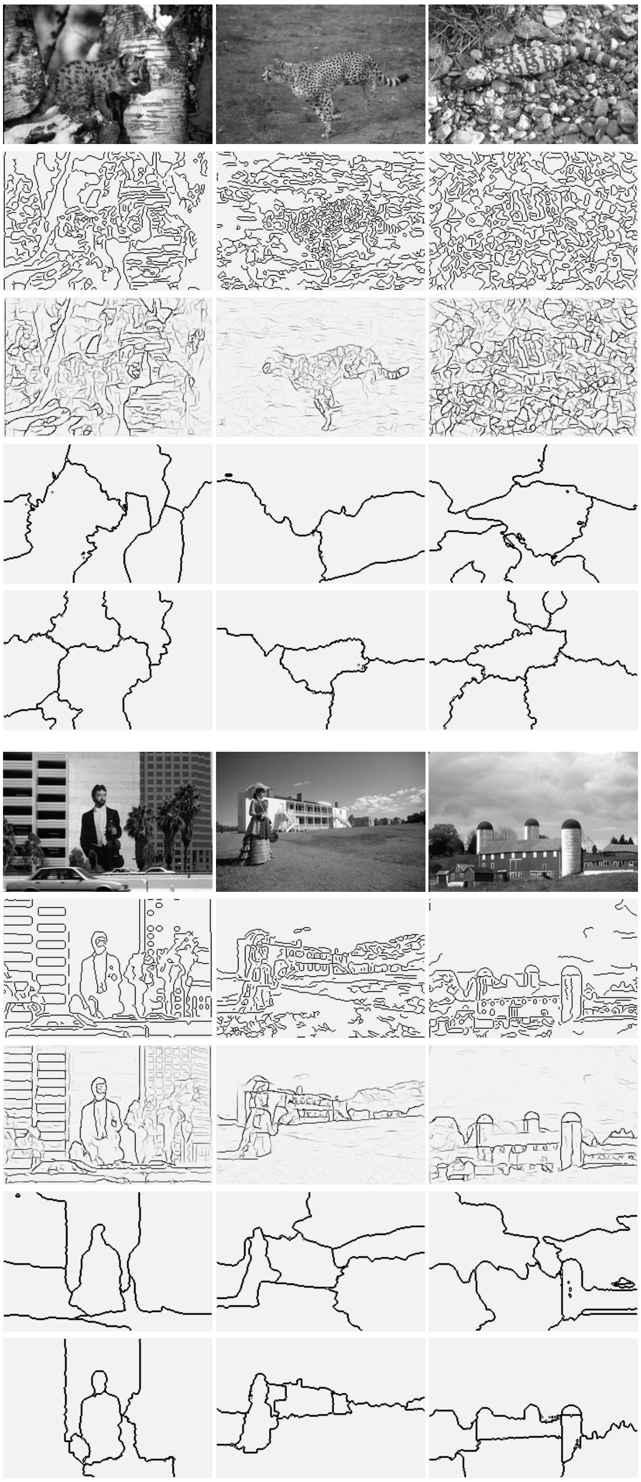


Figure 14. Comparison of Canny, Pb, previous work and this work on more BSDS images. Note that our segmentations have coarse-scale regions, yet they retain fine boundary details. It takes about 2 minutes to segment these 160 × 240 images on a linux machine with 2GHz CPU and 2GB memory. There is no need to learn any models or parameters from some training images about either texture or contour completion.

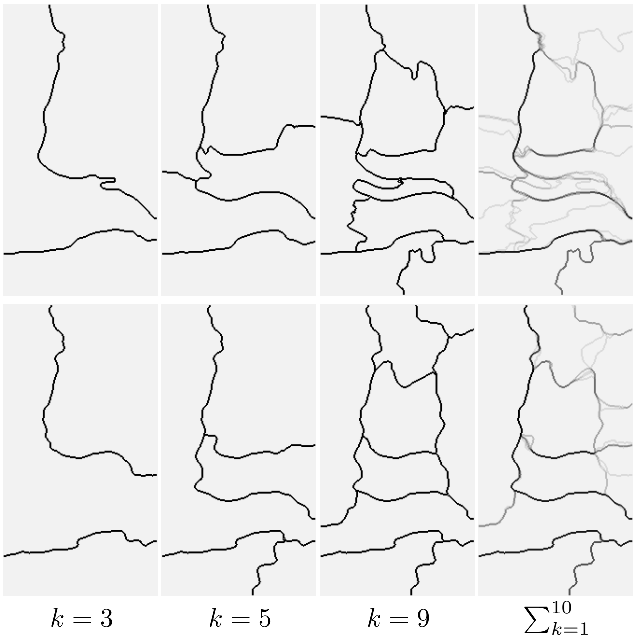


Figure 15. Coarse to fine segmentations for the image in Fig. 1a (Columns 1-3). Column 4: one probabilistic boundary map obtained by summing all the boundaries from the coarse-to-fine segmentations. Row 1: Previous work [13]. Row 2: this work. With gap completion and cross-scale consistency, we have more accurate and consistent boundaries. These are also relatively major boundaries with respect to those in Fig. 1a:Canny or Pb.

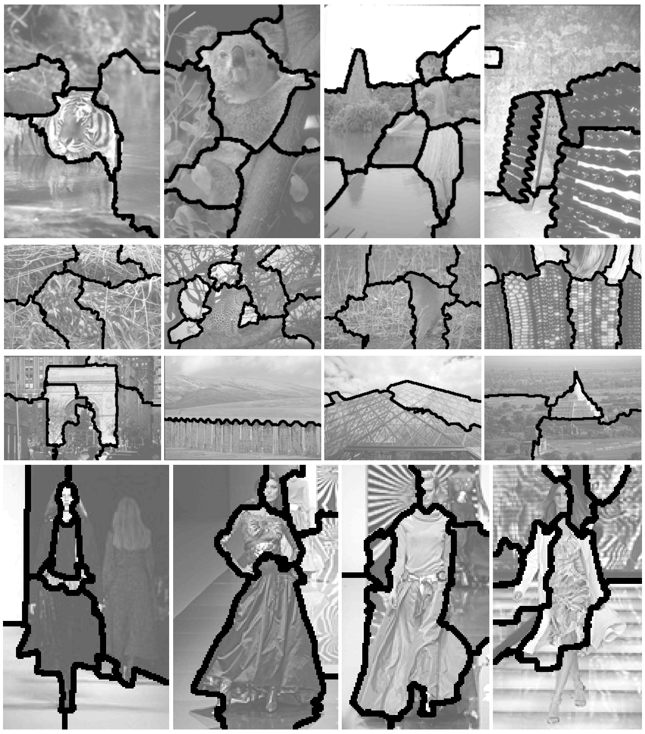


Figure 16. More segmentation results on images that have prominent texture, clutter or illusory contours.

tensity similarity, convexity, closure and curvilinearity. The results thus reflect a tradeoff among these multitudes of grouping factors. To human observers, we are constantly subject to object attention and various forms of perceptual constancy. To our algorithm, there are only 2 essential parameters, σ_{IC} and r, which are applied everywhere in an image. For example, to make sure not to miss weak edges, a very small σ_{IC} is used. This makes even unnoticeable shadow edges significant. Nevertheless, our method is able to rely on the more stable descriptors to handle both texture and illusory contours: that is, grouping consistency across scales for the former and robust spatial configuration over longer ranges for the latter (Fig. 3). A global integration over these local descriptors reveals foremost large low-level organizations that a good segmentation should have.

Acknowledgments

The author thanks anonymous reviewers for helpful comments. This research was funded by DOD/ONR-MURI #FD N00014-01-1-0890 grant.

References

[1] P. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communication*, COM-31:532–40, 1983.

[2] T. Hofmann, J. Puzicha, and J. M. Buhmann. Unsupervised texture segmentation in a deterministic annealing framework. *PAMI*, 20(8):803–18, 1998.

[3] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–7, 1981.

[4] J. Kaufhold and A. Hoogs. Learning to segment images using region-based perceptual features. In *CVPR*, 2004.

[5] B. Kimia, A. R. Tannenbaum, and S. W. Zucker. Shapes, shocks, and deformations, i: the components of shape and the reaction-diffusion space. *IJCV*, pages 189–224, 1995.

[6] J. Koenderink. The structure of images. *Biological Cybernetics*, pages 363–70, 1984.

[7] M. Kubovy and A. O. Holcombe. On the lawfulness of grouping by proximity. *Cognitive Psychology*, 35:71–89, 1998.

[8] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 2001.

[9] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.

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

[11] Z. Tu and S. Zhu. Parsing images into region and curve processes. In *ECCV*, 2002.

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

[13] S. X. Yu. Segmentation using multiscale cues. In *CVPR*, Washington DC, June 27 - July 2 2004.

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