Shape Matching by Segmentation Averaging

Hongzhi Wang and John Oliensis

Stevens Institute of Technology
{hwang3,oliensis}@cs.stevens.edu

Abstract. We use segmentations to match images by shape. To address the unreliability of segmentations, we give a closed form approximation to an average over all segmentations. Our technique has many extensions, yielding new algorithms for tracking, object detection, segmentation, and edge-preserving smoothing. For segmentation, instead of a maximum a posteriori approach, we compute the "central" segmentation minimizing the average distance to all segmentations of an image. Our methods for segmentation and object detection perform competitively, and we also show promising results in tracking and edge–preserving smoothing.

1 Introduction

The shape of an object (as conveyed by edge curves) is among its most distinctive features, yet many methods for recognition/detection or tracking neglect it. One reason for this is that shape matchers confront a difficult global/local dilemma: local edges carry too little information for reliable matching, while globally the images have too much variability.

We classify shape matching strategies according to their shape representations. Methods representing shape *locally* [4,5] in terms of edgels confront a combinatorial explosion in the number of potential matches. *Global* methods have trouble extracting reliable global contours, and their matching suffers from occlusions and dropouts; also, global matching is hard because of the huge search space of possible deformations. Some recent shape-matching approaches [18,10] use grouped edge fragments as *intermediate* representations. These have more specificity and fewer potential matches than edgels yet occur often enough to survive occlusion and detection failures. But bottom–up grouping isn't reliable, so applying fragment groups to match general images is hard without top-down learning. As a result, [18,10] limit matching to specific objects or classes; they learn distinctive fragment groups for the given object(s) and match these representations. Other *semi–local* representations, e.g. SIFT [14] and HoG [8,6], achieve robustness to shape variation by weakening the shape descriptions, resorting to histograms instead of representing the exact boundary shapes.

We propose an approach to shape matching based on averaging over segmentations. The method combines the advantages of global and local approaches: it can match globally yet efficiently, and is robust to local variation yet remains sensitive to the detailed boundary shapes. Others [21,2] have used segmentation

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 562–575, 2008.

for recognition, but we differ in using it to represent shape. For general recognition, we can apply our method like SIFT as a (semi)local descriptor. Here, to demonstrate its power to match despite large variability, we apply it *globally*, in experiments on tracking and on localizing instances of an object class.

Our technique for averaging segmentations has implications beyond matching. Using it, we derive a segmentation method which gives competitive results on the Berkeley database. We also apply it for edge–preserving smoothing. Unlike previous ones, our smoothings are sensitive to *global* image structures.

2 Shape Matching: Motivations and Overview

Region based shape matching. To resolve the global/local dilemma, as a first step we avoid the local ambiguities of edge matching by instead matching *regions* [3]. As global features, regions have robustness to local shape distortions and occlusions, but they are difficult to extract reliably and can have complex shapes which are hard to represent or match. Since they have closed boundaries, they don't adapt easily for matching open image curves.

Segmentation matching. We next upgrade region matching to matching *segmentations*. This has several advantages: 1) By matching all regions at once, we gain robustness to the grouping failures for any one region; 2) Since segmentations are computed using global image information, they can localize the true edges more accurately than local edge–detection/grouping methods; 3) Segmentations can reveal global shape structures which are more distinctive than local features. This helps overcome difficulties caused by "hallucinated" boundaries; 4) An oversegmentation includes most of the strong curves and may be exploited for matching open as well as closed curves.

A typical segmentation includes both true and hallucinated boundaries. In matching segmentations, we need a similarity measure that detects the true matching boundaries while ignoring the hallucinations. The measure should be insensitive to small shifts and distortions in the boundaries.

A segmentation similarity measure. To achieve this, we propose a new similarity measure. It relates to the mutual information as adapted to segmentations; the basic concept is the *structure entropy* (SE), i.e., the entropy measuring a segmentation's complexity. A segmentation with many small segments has high structure entropy; large segments give low SE. (Note: we compute the structure entropy for individual segmentations, not for the probability distribution over segmentations.) We define the *structure mutual information* (SMI) between segmentations in terms of the joint segmentation obtained by superimposing the individual ones. When two segmentations have matching boundaries, the joint segmentation has large regions and low SE, so the SMI is high. For non–matching segmentations, the joint segmentation has smaller regions and higher SE, so the SMI is low. Fig.1(a) illustrates the idea. The precise definitions are below.

This similarity measure achieves the desired aims. For example, shifting one segmentation a small amount relative to another creates new small regions in

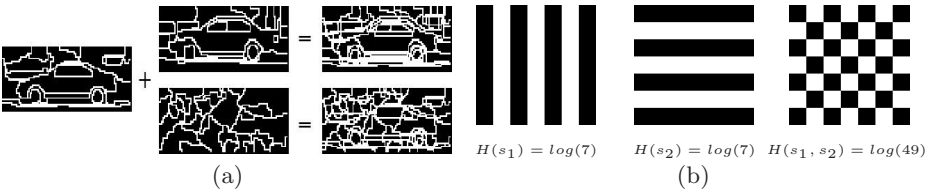


Fig. 1. (a). Segmentation based shape matching. Left 2 columns: segmentations to be matched. Right column: the joint segmentations. Two similar overlapping shapes (top) gives larger regions than overlapping dissimilar shapes (bottom); (b) 2 segmentations and their joint segmentation (right). The segmentations have VoI log(49) and MI 0.

the joint segmentation, but large segments continue to have large overlaps, so the joint SE remains low and the SMI remains high.

However, the approach still depends on the quality of the precomputed segmentations. In many cases, an oversegmentation includes enough of the true boundaries for good results, but it will also include fake boundaries, weakening shape accuracy. Our solution to this problem is our main theoretical contribution. Instead of using actual segmentations, we compute their average similarity, averaging over all possible segmentations for both images weighted by their probabilities. Indistinct boundaries also contribute to the average, so we get good matching even for unstructured images.

Segmentation averaging. On its face, averaging over all segmentations seems impossible. It is clear that we cannot do it exactly. We present an approximation which gives the average similarity between segmentations in closed form.

The main ideas in our approximation are as follows. First, we make a standard approximation to the segmentation probability distribution, representing it in terms of the local affinities between pixels. For example, one can consider normalized cuts (NCuts or NC) [23] as computing the maximum a posteriori (MAP) segmentation for a probability distribution defined by the affinities.

Our key realization is that we can compute the averaged structure entropy (and similarly the averaged SMI) separately in terms of each pixel's contribution. We show that a pixel's contribution to the SE is the geometric mean size of the segment containing it. The geometric mean is also hard to compute exactly, but we exploit a standard technique to approximate it in terms of the arithmetic mean. This approximation is a mild one: roughly, it originates from a Taylor expansion of the geometric mean around the arithmetic mean, and we show both theoretically and experimentally that the higher order terms in this expansion can be neglected. The following sections present the details of our approach.

3 Segmentation Similarity and Averaging

A segmentation similarity measure. We start by defining the structure entropy (SE). Given a segmentation s, we define a r.v. x which we consider as

ranging uniformly over all pixels. More precisely, we define the states of x by the segment labels in s, so the probability p_i for the i_{th} segment is the probability that a pixel lies in that segment, i.e., it is the area ratio of the segment to the whole image. For n segments, the structure entropy for s is

$$H(s) \equiv H(x) = -\sum_{i=1}^{n} p_i \log(p_i). \tag{1}$$

For two segmentations s, s' and corresponding r.v. x, y, the joint structure entropy $H(s, s') \equiv H(x, y)$ is the structure entropy of the joint segmentation. See Fig. 1(b). Let z be the r.v. for the joint segmentation. The possible states of z are the label pairs $(l_s, l_{s'})$ where, for each pixel, l_s and $l_{s'}$ give the containing segment from s and s' respectively. A pixel with labels l_s and $l_{s'}$ lies in the intersection of the corresponding segments.

Having defined the structure entropy, we define the mutual information (MI) of two segmentations in the standard way:

$$H(x; y) = H(x) + H(y) - H(x, y). \tag{2}$$

We call this the *structure mutual information* (SMI). We also define the *variation of information* (VoI) [17] as $\mathrm{V}(x, y) \equiv H(x, y) - H(x; y)$ or, equivalently,

$$\mathrm{V}(x, y) \equiv 2H(x, y) - H(x) - H(y). \tag{3}$$

[17] showed recently that the VoI gives a distance metric for clusterings. In our context, it gives a distance metric on segmentations[1].

3.1 The Averaged Structure Entropy

As described in Section 2, we match images by computing the average similarity of their segmentations. To do this, we need the average of the structure entropy over all possible segmentations. This section describes our main theoretical contribution: an approximation to this average.

Let F be the image. For a given segmentation s, the m_{th} pixel contributes

$$H^{(m)}(s) = -A_F^{-1} \log(A_F^{-1} A(s^{(m)})) \sim \log(A(s^{(m)})) \tag{4}$$

to the SE, where $s^{(m)}$ is the segment containing pixel m, $A(\cdot)$ gives its area, and $A_F \equiv A(F)$ is the total area of F. The SE is the sum of $H^{(m)}(s)$ over all pixels.

Let S denote the set of all possible segmentations of F. We define the *averaged structure entropy* ASE by

$$H_\omega(F) = \sum_{s' \in S} p(s'|F) H(s'), \tag{5}$$

[1] We derived our measures independently but later than Meila.

where $p(s'|F)$ is the conditional probability of the segmentation s' given F and $H(s')$ is the SE for s'. The contribution of the m_{th} pixel to the ASE $H_\omega(F)$ is

$$H_\omega^{(m)} = \sum_{s\in S} p(s|F)H^{(m)}(s) \sim \log[\prod_{s\in S} A(s^{(m)})^{p(s|F)}]. \tag{6}$$

Our key insight is: we can evaluate the ASE *without enumerating all segmentations* if we can compute the geometric mean segment size $G \equiv \prod_{s\in S} A(s^{(m)})^{p(s|F)}$.

The geometric mean is hard to compute. A common approximation [27] is $G \approx \mu - \sigma^2/2\mu$, where μ is the arithmetic mean and σ^2 is the variance. Then

$$G \approx E\{A(s^{(m)})\} - \mathrm{Var}\{A(s^{(m)})\}/(2E\{A(s^{(m)})\}). \tag{7}$$

(E is the expectation.) One can conveniently express the arithmetic mean and variance in terms of the *affinity matrix* M_F, defined by

$$M_F(m,n) = \sum_{s\in S} p(s|F)M_s(m,n), \tag{8}$$

where M_s is the *segmentation affinity matrix* for s with entries 1 (the pixels belong to the same segment) or 0. For an image with N pixels, $M_F \in \Re^{N\times N}$. Each entry measures the probability that the two pixels lie in the same segment.

For any pixels m, n, let $\chi_{(m,n)}$ be the indicator variable representing the event that the given pixels belong to the same segment. Then $E\{\chi_{(m,n)}\} = M_F(m,n)$. The m_{th} pixel's segment–size mean and variance are

$$\mathrm{E}\{A(s^{(m)})\} = E[\sum_n \chi_{(m,n)}] = \sum_n M_F(m,n) \tag{9}$$

$$\mathrm{Var}\{A(s^{(m)})\} = \sum_{k,l} cov(\chi_{(m,k)}, \chi_{(m,l)}) = \sum_{k,l} E[\chi_{(m,k)}, \chi_{(m,l)}] - E[\chi_{(m,k)}]E[\chi_{(m,l)}] \tag{10}$$

$$\leq \sum_{k,l} \min(M_F(m,k), M_F(m,l))(1 - \max(M_F(m,k), M_F(m,l))). \tag{11}$$

To apply in practice, we estimate M_F from local image properties, e.g.

$$M_F(m,n) \approx \begin{cases} exp(\frac{-(F_m - F_n)^2}{2\sigma^2}) & \text{if } d(m,n) \leq D; \\ 0 & \text{if } d(m,n) > D, \end{cases} \tag{12}$$

where F_m is the intensity, $d(m,n)$ the distance between pixels m, n, and σ the standard deviation of intensity within a segment. (12) embodies the principle that nearby pixels with similar intensity are more likely to group than distant pixels with different intensities. Other cues, e.g., texture or the presence of an intervening edge [15], could be used as well.

We simplify further by treating the pairwise probabilities (that two pixels belong to the same segment) as independent. Then we have:

$$p\left(s|F\right) \approx \frac{1}{Z} \prod_{m\geq n} M_F\left(m,n\right)^{M_s(m,n)} \left(1 - M_F\left(m,n\right)\right)^{1-M_s(m,n)},$$

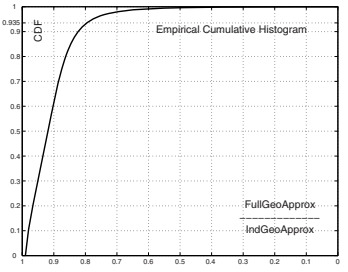


Fig. 2. Empirical distribution shows small impact of the independence assumption

where Z is a normalization constant. As discussed in Section 2, this assumption is a common one and underlies segmentation algorithms such as NC. Using it,

$$\mathrm{Var}\{A(s^{(m)})\} \approx \sum_n M_F(m,n)(1 - M_F(m,n)). \tag{13}$$

(9) and (13) imply $\frac{Var\{A(s^{(m)})\}}{2E\{A(s^{(m)})\}} < 0.5$. Since meaningful segments are sizable, we expect their mean segment size $E\{A(s^{(m)})\} = \mu \gg 0.5$, implying that the arithmetic mean approximates the geometric mean well. We use this approximation in all our experiments, taking $H_\omega(F) \approx \sum_m \log\left(\sum_n M_F(m,n)\right)$.

Validation of pairwise independence. For us the pairwise–independent approximation is especially appropriate, since we only keep the affinities of nearby pixels (only nearby affinities can be reliably estimated) and these are mostly near 1 with small covariances, see (11). As a check, Fig. 2 shows the empirical distribution of $\frac{\bar{G}_{Full}}{G_{Ind}}$ over real images, where $\bar{G}_{Full}$ is a lower bound on the geometric mean computed from (11) without the independence assumption, and G_{Ind} is the mean computed assuming independence.

The images are the Berkeley segmentation training data [16] (200 images here resized to 80×120). For any image, each pixel gives a sample of the lower bound. We computed the affinity matrices using the Gaussian (12) with deviation $\sigma = 10$ (intensity range [0,255]) and $D = 5$. For over 93.5% of the pixels, the ratio lower bound is above 0.8, implying that the independence assumption has little effect in practice. Note that Fig. 2 plots a lower bound on the ratio, so the ratio itself will have an even better empirical distribution.

A modification. A potential issue is that estimates of the pairwise probabilities in M_F are reliable only over a small region (D in (12) should be small). Thus, we average the structure entropy with respect to the neighborhood size instead of the full image size, redefining the ASE as:

$$H_\omega = -\frac{1}{A_F} \sum_m \log\left(\frac{E\{A(s^{(m)})\}}{A_m}\right) \tag{14}$$

where A_m is the neighborhood size for the m_{th} pixel.

3.2 Shape Similarity from Averaging Segmentations

We compare the shapes in two images by computing the average distance between the image segmentations. Recall the definition (3) of VoI, which gives a metric on segmentations. The average value of this metric, averaged over all possible segmentations for two images F_1 and F_2, is $V_\omega(F_1, F_2) \equiv 2H_\omega(F_1, F_2) - H_\omega(F_1) - H_\omega(F_2)$, where the $H_\omega(F_a)$ are the ASE for the two images, and

$$H_\omega(F_1, F_2) = \sum_{s_1} \sum_{s_2} p(s_1|F_1)p(s_2|F_2)H(s_1, s_2) \tag{15}$$

is the ASE of the joint segmentations for the two images. As before, we approximate $H_\omega(F_1, F_2) \approx \sum_m \log\left(\sum_n M_{F_1F_2}(m, n)\right)$; the joint affinity matrix is:

$$M_{F_1F_2}(m, n) = M_{F_1}(m, n)M_{F_2}(m, n). \tag{16}$$

The VoI $\mathrm{V}(s, s')$ has the joint structure entropy $H(s, s')$ as an upper bound. When one searches for the most similar segmentation to a given segmentation or image, this biases the result toward segmentations of low complexity. To compensate for the bias, we normalize our averaged distance, using $\Delta(F_1, F_2) \equiv \frac{V_\omega(F_1,F_2)}{H_\omega(F_1,F_2)}$ as our measure for image comparisons. Note that $0 \leq \Delta \leq 1$, where $\Delta = 1$ implies that the images are very dissimilar. Our approximations give

$$\Delta(F_1, F_2) \approx 2 - \frac{\sum_m \log\left(\sum_{nn'} M_{F_1}(m, n)M_{F_1}(m, n')\right)}{\sum_m \log\left(\sum_n M_{F_1}(m, n)M_{F_2}(m, n)\right)} = 2 - \frac{\sum_m \log\left(\bar{A}_{1m}\bar{A}_{2m}\right)}{\sum_m \log\left(\bar{A}_{\mathrm{J}m}\right)}, \tag{17}$$

where $\bar{A}_{am}$ is the mean size of the containing segment for the individual or joint segmentation. Roughly, Δ measures the statistical independence of the segment sizes in the two images. Sec. 5 applies Δ in tracking and detection experiments.

Comparison to other measures. The Probability Rand (PR) index [26] can also be considered a similarity metric based on affinity matrices:

$$PR(F_1, F_2) \propto \sum_m [2\sum_n M_{F_1F_2}(m, n) - \sum_n M_{F_1}(m, n) - \sum_n M_{F_2}(m, n)] \tag{18}$$

Note that PR sums the affinities separately, ignoring all spatial interactions between nearby pixels. Another affinity-based measure, the similarity template [25] (ST), includes the spatial interactions for each image separately but not for the joint affinity. Our metric does include spatial interactions. Sec. 5 shows experimental comparisons of our approach with PR and ST.

4 Application to Segmentation and Smoothing

Segmentation algorithms such as NC can be considered as computing the MAP segmentation. The wide divergence in segmentations found by different methods, and the imbalance between small/large segments, suggest that the probability of segmentations has a broad asymmetric peak. For any such r.v., the mean is the estimator with least variance and usually superior to the MAP. Can we use our averaging technique to approximate the mean segmentation?

The central segmentation. Since the mean has least variance, we can compute it for a r.v. x as $\bar{x} = \text{argmin}_x \sum_y p(y)|y-x|^2$. Recall that $V(s,s')$ gives a metric for segmentations. Given an image F, we define its *central segmentation*

$$\hat{s} \equiv \text{argmin}_s \sum_{s' \in S} p(s'|F)\text{V}(s',s) \equiv \text{argmin}_s V_\omega(s,F). \tag{19}$$

$\hat{s}$ is "central" in that it minimizes the average distance V to all segmentations of F. It is the mean segmentation with respect to the distance metric $\sqrt{V}$. We choose $\hat{s}$ partly for convenience, since we already approximated V_ω; also, we expect *better* segmentations using $\sqrt{V}$ and $\hat{s}$ than for the metric V.

Our reasons for this expectation are as follows. A typical image has many qualitatively different yet plausible segmentations, implying $p(s|F)$ has many large peaks. Averaging over all s combines qualitatively different segmentations, whereas we want the center of the dominant peak. Using the metric $\sqrt{V}$ for the average gives a robust estimate with more resistance to outlier segmentations.

Note that [11] also segments by averaging over segmentations. Differences include: [11] computes the mean affinity matrix, not the segmentation directly, and averages by *sampling*. Our result is closed form and applies more generally, e.g., to matching; [11] focuses just on segmentation.

We again normalize the averaged distance, redefining $\hat{s} \equiv \text{argmin}_s \frac{V_\omega(s,F)}{H_\omega(s,F)} \equiv \text{argmin}_s \Delta(s,F)$, where $H_\omega(s,F) \equiv \sum_{s' \in S} p(s'|F)H(s,s')$.

Fig. 3. Segmentation results for naive greedy merging

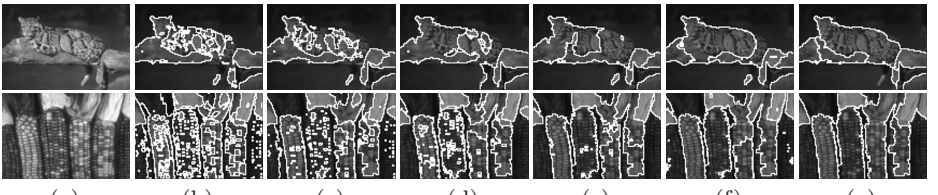
Segmentation algorithms. We use the simple affinity matrix M_F of (12) to compute $V_\omega(s,F)$, $H_\omega(s,F)$ in all our experiments. We compute $\hat{s}$ by iteratively minimizing $\Delta(s,F)$. We used two iterations; the first is fast greedy merging (GM). We start with each pixel as a segment, then merge neighbor segments if this decreases $\Delta(s,F)$. Because M_F is nonzero just over small neighborhoods, we can compute the merged segmentation with linear cost $O(NW)$, where W is the neighborhood size determined by D in (12). We used $D = 5$ for GM, giving $W = 121$. To avoid local minima, one can repeat the merge several times, starting with a small σ in M_F (e.g., 1) and then gradually increasing it to a specified value. Fig.3 shows this naive method can give excellent segmentations, indicating the robustness of our segmentation criterion $\Delta(s,F)$.

Our second method uses gradient descent. We represent a segmentation by *real-valued* labels s_m at each pixel m, with neighboring pixels in the same segment *iff* their labels differ by < 1. Letting I denote the indicator function,

$$H(s) = -\sum_m \log \frac{\sum_n M_s(m,n)}{A_m} = -\sum_m \log \frac{\sum_n I(|s_n - s_m| < 1)}{A_m} \tag{20}$$

$$H_\omega(s,F) = -\sum_m \log \frac{\sum_n I(|s_n - s_m| < 1) M_F(m,n)}{A_m}. \tag{21}$$

We initialize s to the original intensity image. Since we require smooth gradients, we approximate the derivative of $I(|s_m - s_n| < 1)$ by a smooth function.[2] Using a smoother approximation speeds up convergence and extends the search for $\hat{s}$ over a larger range. Note that we do *not* change our criterion $\Delta(s,F)$ for a good segmentation. Though our approximations may cause $\Delta(s,F)$ to increase after some iterations, at the end the algorithm outputs the s giving the least $\Delta(s,F)$. For even faster convergence, we add a "force" term $\gamma \frac{\partial H_\omega(s,F)}{\partial s_m}$ to the gradient, where we set $\gamma = 0.5$. Again, adding this term widens the search but does not affect our criterion $\Delta(s,F)$. The force acts as a regularization that focuses the search on simpler segmentations, helping overcome local minimum. With these changes, we usually get convergence in a few hundred iterations. We always ran for 500 iterations in our experiments, which takes a few seconds on a 3.2G Hz AMD for images with 10000 pixels (code available on our web page).



(a) (b) (c) (d) (e) (f) (g)

Fig. 4. a). Input images (80×120); b)-g). Segmentations after 1 - 6 optimization rounds. After each round, we update standard deviations via (22) to encourage merging of small segments. 1_{st} row: $(D,\sigma,\alpha,t) = (2,50,0.25,20)$;$2_{nd}$ row: $(D,\sigma,\alpha,t) = (2,30,0.5,20)$.

Initially, we set σ in M_F to a constant. This doesn't allow for changes across the image [9], as occur especially in textured regions. To deal with such changes, we adapt σ locally based on the current segmentation:

$$\sigma_{(m,n)} \longleftarrow \sigma_{(m,n)}(1 + e^{-\frac{A(s^{(m)})}{t}})(1 + e^{-\frac{A(s^{(n)})}{t}}), \tag{22}$$

where the parameter t acts as a threshold that encourages segments smaller than t pixels to merge into their neighbors, with little effect on large segments. Finally, we repeat the whole round of gradient descent/σ-update. The results stop changing after a few rounds; we ran 6 rounds in all our segmentation experiments. As Fig. 4 shows, updating σ highlights the large salient structures, adding some stability in textured regions. (We don't model texture explicitly as in [22], so we cannot expect competitive performance in texture segmentation.)

[2] $\max(1, |s_m - s_n|)^{-\alpha} \text{sign}(s_m - s_n)$. We used $\alpha = 0.25, 0.5$.

Smoothing. Segmentation relates closely to edge–preserving smoothing; in fact, one can consider it as a piecewise constant smoother. We implement edge–preserving smoothing by finding the "most similar" image to the original image F according to our criterion Δ. The algorithm is steepest descent, similar to the second segmentation procedure (but without the derivative approximation or extra force), except we optimize Δ over images instead of segmentations.

The most similar image F_{sim} does *not* equal the original. Instead, one can show that it has a segmentation probability distribution which clusters around $\hat{s}$, the central segmentation of F. As a result, F_{sim} agrees with the boundaries of F and varies smoothly over its non–boundary regions (to discourage unlikely segmentations). Unlike previous methods based on local computations, our approach smooths according to the image's *global* optimal structures. By averaging over probabilities, it can adjust the smoothing according to boundary strength and smooth *across boundaries*, not just within segments. Fig. 5 shows our algorithm gives appealing smoothings which preserve accurate contours.

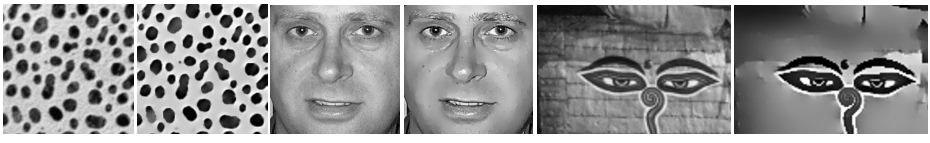


Fig. 5. Left: the original images; Right: results after 30 iterations of smoothing. $D = 2$. The σ used are 60,20,60 respectively.

5 Experiments: Segmentation, Detection, and Tracking

Image segmentation. Fig. 6 shows sample segmentations by our methods compared to the best ones from EDISON mean-shift (MS) [7]. On images with faint structures and large local ambiguity, MS often gives less accurate boundaries. For a quantitative comparison, we use the Berkeley segmentation test set [16] with each gray image resized to 80×120. Since the standard boundary-consistency criterion isn't optimal for evaluating segmentations, we use region consistency as our criterion. We consider multiple human-labeled segmentations as giving a probability distribution for the "ground truth," and evaluate segmentations by their averaged distance from this distribution using $\Delta(s_1, s_2) = V(s_1, s_2)/H(s_1, s_2)$. (Note this is *not* the $\Delta(s, F)$ minimized by our algorithm!) We applied MS using default parameters and intensity cues only. Our method used $D = 2, \alpha = 0.5,\ t = 2$. We also tested NC [23]. For each method, we obtain 3 segmentations per image by: varying the minimum region size as 5, 10, and 50 (MS); choosing $\sigma = 40, 60, 100$ (ours); specifying 60,40,20 segments (NC). The parameters are chosen s.t. the segmentations for different methods contain similar number of segments. Table 1 shows our method does best.

Table 1. The mean average distance to "ground truth" segmentations

average segment #	Ours	MS	NC
20	0.644	0.653	0.662
40	0.632	0.654	0.685
60	0.629	0.657	0.699

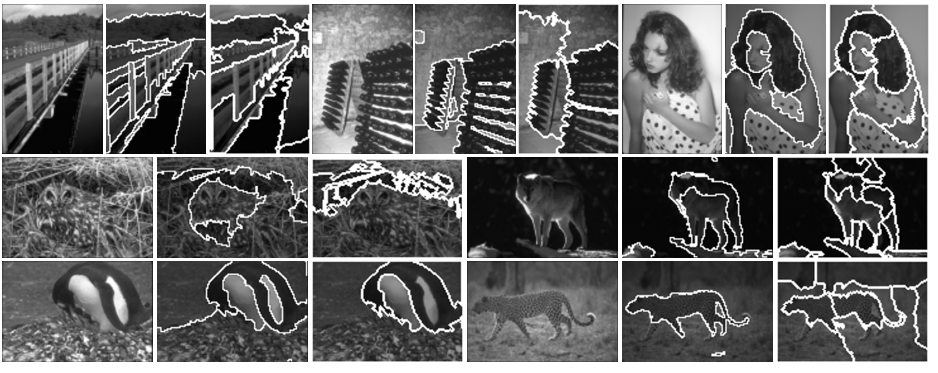


Fig. 6. Sample segmentation results. 1_{st}: Input images, size 120×80 or 80×120; 2_{nd}: segmentations by our approach; 3_{rd}: segmentations by mean shift (EDISON).

Detection. The "bag of features" (BoF) approach to recognition compares images according to the appearance (textures) of small patches. To cope with global variability, it takes a resolutely local approach, neglecting spatial layout almost completely. More recent work uses some layout information [12], or local shape descriptors [18,10,24], or a combination of local appearance and shape [19], and gives improved performance especially for object detection (i.e., localization). As discussed in the introduction, the local–shape approaches rely on supervised learning: to match an image, they must first build a descriptor of its contents from training images.

Our approach can match shapes for general images. It complements the local–appearance methods, and we could combine the approaches by treating ours as another semi-local descriptor (but for shape). This would be the appropriate strategy for recognition of general non–rigid objects.

Here, we concentrate on testing our method, applying it for detection on its own. To demonstrate its robustness against global variability, we apply it as a *global* descriptor: We detect an instance by measuring test images against a template containing the *whole* object plus context. We use minimal learning, detecting objects by thresholding our similarity measure with non-maximum suppression [1]. We ran detection experiments on the UIUC car side-views and the CalTech car rear-view and face data. The UIUC data contains 550 positive exemplars. We used their average affinity matrix as the car-side template, detecting cars in test images by thresholding the normalized distance from the

template. For the face and car rear-view data, we manually cut 10 positive examples from the first 10 test images and detected by thresholding the average distance from these exemplars. Templates were scanned across the test images.

Instead of using (12), we estimated the affinities separately for each image based on its statistics. Given image F, let h_r^F be the empirical histogram of the absolute intensity difference between pixels at a relative distance r. To get the affinities, we normalize so that $h_r^F \in [0,1]$ and average over r:

$$h^F(i) = D^{-1} \sum_{r=1:D} h_r^F(i) \tag{23}$$

The normalization gives higher weight to the closer pixels. The choice for the cut off D should reflect the scale of the object. For the car-side, car-rear, and face data, we used $D = 2, 7, 5$ respectively.

Table 2. Detection performances: equal error rates (percentage)

	Ours	[25]	[26]	[18]	[24]	[19]
Faces	98.1	93	92	96.4	97.2	99
Crear	100	98.2	86	97.7	98.2	100
Cside	90	–	–	85	–	93.8

As expected, our method gives better results than [25,26] (see Table 2). Our results are also better than [18,24], which learn local shape descriptors from training images, assuming, as we do, that the object is delineated by a bounding box. [19] does better on car-side and slightly better on faces, but unlike us uses both shape and appearance. Considering all reported results on these data, our performance on faces and car-rear views is close to the state of the art with many fewer exemplars, with slightly worse results for the car-side data. The latter contains images with significant occlusion, so our global matching strategy cannot compete with approaches based on local features. The best result for the car–side data is 97.5% [13], but this algorithm unlike ours has a verification stage; without this stage, it gives performance similar to ours.

Tracking. We next present a simple application of our match measure to tracking. Many shape–based trackers use global active contours; these require good initializations around the object of interest. Other approaches weaken the shape representation to make it robust to shape distortion, for instance representing the object in terms of histograms over gradients, e.g. [8].

Our approach tracks an object by its detailed curve shapes. As a shape tracker, it can localize the object more accurately than histogram–based ("blob") trackers. Since our matching measure has a built-in robustness to shape distortions, we can implement tracking essentially as simple template matching.

The user selects a window around an object in the starting image, and the algorithm tracks by moving the window to the best shape match in each new image (no motion, background information, or learning). Currently, we use brute

Fig. 7. Tracking results. Bottom corner of each frame shows expected segment size for each pixel of current template: dark pixels lie near boundaries. Our brute-force search Matlab routine takes a few sec/frame with a 3200HZ AMD.

force search to find the best location; using iterative gradient ascent would give a faster algorithm. To handle occlusion, we include history into the current window representation, updating its affinity matrix description by $M_F^{(n)} = (19M_F^{(n-1)} + M_{F(n-1)})/20$, where $M_{F(n-1)}$ is the affinity matrix for image $n-1$ alone, and we use $M_F^{(n)}$ for matching. Our results (e.g., Fig. 7) on a PETS 2007 sequence, and on outdoor and indoor sequences from [20], show the method's robustness to occlusion, deformation, camera motion, and changes in illumination, scale and pose. Our tracking is near perfect; for complete results see our web page.

6 Conclusion

We use image segmentations for shape matching. Our approach can match curves without correspondence, over large-scale image regions, and with good robustness to local shape variations and occlusion. It can exploit global shape structures, which are more distinctive than local features. To address the unreliability of image segmentations, we describe a closed form approximation to an average over all segmentations. Our approach has many extensions, yielding algorithms for tracking, segmentation, and edge–preserving smoothing. In addition, we can apply our approach for the objective evaluation of segmentation algorithms, and for comparing computed segmentations to multiple "ground–truth" segmentation produced by humans. Finally, since our approach compares signals based on their internal structure, it can match signals from different modalities, e.g. images from different frequency bands, or visual images matched to sonar data.

References

1. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. In: Tistarelli, M., Bigun, J., Jain, A.K. (eds.) ECCV 2002. LNCS, vol. 2359, pp. 113–127. Springer, Heidelberg (2002)
2. Ahuja, N., Todorovic, S.: Learning the taxonomy and models of categories present in arbitrary images. ICCV (2007)
3. Basri, R., Jacobs, D.: Recognition using region correspondences. In: IJCV (1997)
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. PAMI 24(4), 509–522 (2002)
5. Berg, A., Malik, J.: Geometric blur for template matching. In: CVPR (2001)
6. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIRV (2007)

7. Comaniciu, D., Meer, P.: Mean shift: a robust approach towards feature space analysis. IEEE Trans. PAMI 24(5), 603–619 (2002)
8. Danal, N., Triggs, B.: k Histograms of oriented gradients for human detection. In: CVPR (2005)
9. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV 59(2) (2004)
10. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detection with deformable shape models learnt from images. In: CVPR (2007)
11. Gdalyahu, Y., Weinshall, D., Werman, M.: Self organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database. IEEE Trans. PAMI 23(10), 1053–1074 (2001)
12. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene catergories. In: CVPR (2006)
13. Leibe, B., Leonardis, A., Schiele, B.: Combined object catergorization and segmentation with an implicit shape model. In: ECCV workshop on SLCV (2004)
14. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
15. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. IJCV (2001)
16. Martin, D.R., Fowlkes, C.C., Tal, D., Malik, J.: A database of human segmented natural images and its applications to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
17. Meila, M.: Comparing clusterings by the variation of information. In: COLT (2003)
18. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for object detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 575–588. Springer, Heidelberg (2006)
19. Opelt, A., Pinz, A., Zisserman, A.: Fusing shape and appearance information for object catergory detection. In: BMVC (2006)
20. Ross, D., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. In: IJCV (2007)
21. Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
22. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A.: Hierarchy and adaptivity in segmenting visual scenes. Nature (2006)
23. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. PAMI 22(8), 888–905 (2000), `http://www.cis.upenn.edu/~jshi/software/`
24. Shotton, J., Blake, A., Cipolla, R.: Multi-scale categorical object recognition using contour fragments. IEEE Transactions on PAMI (2008)
25. Stauffer, C., Grimson, E.: Similarity templates for detection and recognition. In: CVPR (2001)
26. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Trans. PAMI 29(6), 929–944 (2007)
27. Young, W.E., Trent, R.H.: Geometric mean approximation of individual security and portfolio performance. J. Finan. Quant. Anal. 4, 179–199 (1969)