7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition, Bonn, Germany, August 24-27, 2009

Boundaries as Contours of Optimal Appearance and Area of Support

Christina Pavlopoulou Stella X. Yu

Computer Science Department
Boston College, Chestnut Hill, MA 02467
{pavlo,syu}@cs.bc.edu

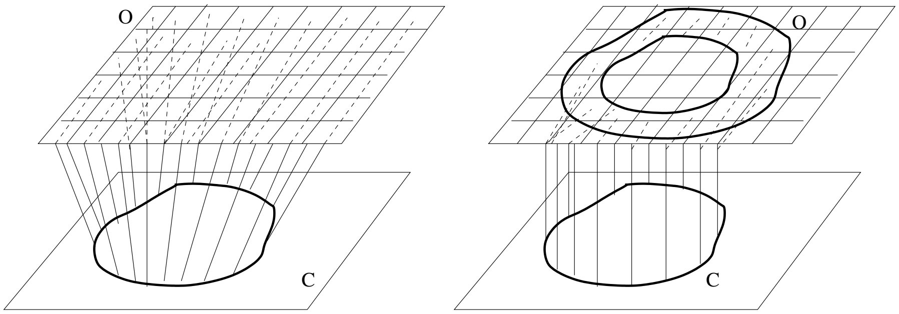
Abstract. Bayesian boundary models often assume that the evidence for each contour is derived from the entire image. Consequently, the normalization term in the Bayes rule is the same for every contour and becomes irrelevant when seeking the optimal. However, in practice these models only use the vicinity of a contour, making the normalization term contour-specific. We propose a formulation that acknowledges the normalization term and includes it in the optimization. We show that it can be interpreted as a confidence measure promoting contours which are far better than other nearby candidate contours. We validate our approach in an interactive boundary delineation setting and demonstrate that complex boundaries can be extracted with significantly smaller amount of user input than when traditional Bayesian models are employed.

1 Introduction

The Bayesian formulation for finding boundaries in the image seeks contour C with the maximal posterior probability $P(C|O)$ given observation O:

$$P(C|O) = \frac{P(O|C)\ P(C)}{P(O)} \tag{1}$$

Our work concerns the normalization term $P(O)$ in the formulation (Fig 1).



a: traditional dependencies **b:** proposed dependencies

Fig. 1. The observation O for a contour C depends on the entire image in traditional Bayesian models (**a**), and only on the vicinity of C in our model (**b**).

Previous work has assumed that O always consists of features extracted from the entire image and it is the same for every C (Fig. 1a). Hence the normalization term $P(O)$ becomes irrelevant and can be ignored during the optimization of $P(C|O)$. However, this assumption does not typically hold in implementations: each contour is often characterized with locally extracted features (Fig. 1b). In other words, O and thus $P(O)$ are in fact different for different contour C's, and $P(O)$ cannot be omitted from the optimization; it affects the maximum of $P(C|O)$ and the optimal contour C.

Our formulation adopts the same criterion in Eqn. 1, but acknowledges the normalization term $P(O)$, so that boundaries are optimal contours in terms of both appearance and area of support. An optimal contour not only best explains the image evidence $P(O|C)$ (e.g. delineating the intensity discontinuities in the image), and have the desired properties $P(C)$ (e.g. smooth), but they are also the best candidates in their vicinity with respect to their local evidence $P(O)$ (e.g. weak but distinctive boundaries).

The normalization term not only helps promote distinctively better contours, but also addresses the length bias problem which plagues energy models for boundaries: short contours automatically have a lower energy than longer ones.

The bias problem has been long recognized and tackled in a number of ways. One approach is to provide a good initialization of boundaries [1, 2], or guide the user during the delineation process [3–5]. A significant amount of human intervention is often required in these cases. Other approaches have incorporated heuristics in the optimization method [6–8]. These methods essentially extract the boundary in a piecewise fashion and it is unclear whether the collection of these boundary segments is optimal. Additional image features [9–11] and stronger contour priors [12–14] have also been explored. Such methods impose additional constraints but do not fundamentally tackle the bias problem. The most direct attempt to solving the bias problem has been to normalize the contour goodness score by the length of the contour [15, 16]. These approaches however are applicable to closed contours and do not admit user interaction.

Our formulation by design does not favor degenerate solutions such as short contours. The normalization term serves as a confidence measure and only favors contours which are significantly better than other candidates.

Including the normalization term results in a more complex criterion. However, the criterion can be globally optimized using dynamic programming in polynomial time, with little loss in computational efficiency.

We develop and analyze our Bayesian formulation in Section 2, address computational issues related to the global optimization in Section 3, evaluate our method in Section 4, and conclude in Section 5.

2 Bayesian Formulation

We first develop our formulation and explain why the contour-specific normalization term favors contours distinctive in their vicinity. In this sense, our method is similar to non-maximum suppression used in edge detection [17]. While both methods enhance boundaries, non-maximum suppression does so in a heuristic and local fashion, whereas our method is principled and operates globally.

We then relate our criterion to entropy and show that the normalization term can be seen as a confidence measure of the quality of a boundary.

2.1 Criterion: Contour-Dependent Observations

Let O_C denote the observations associated with candidate contour C (Fig. 1b). We assume that contour points C_i are conditionally independent and that the observations are conditionally independent given the contour points. We have:

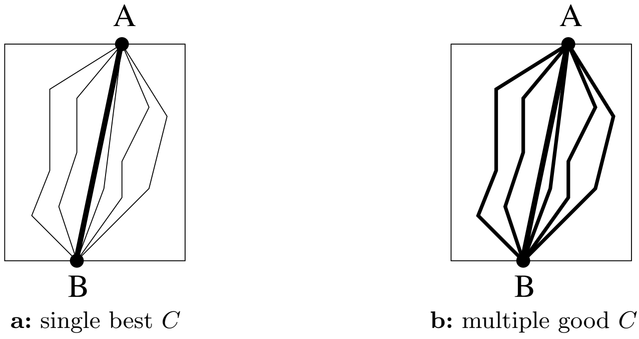
$$\log P(C|O_C) = \sum_i \{\log P(O_{C_i}|C_i) + \log P(C_i|C_{i-1})\} - \log P(O_C) \tag{2}$$

$$P(O_C) = \sum_C P(O_C|C)\ P(C) \tag{3}$$

The difference of Eqn. 2 with all previous Bayesian formulations is the term $P(O_C)$. In traditional formulations, it is the same for all contours and does not play any role in the optimization. When this term is not present, the probability of a contour decreases monotonically with its length; short contours are significantly more likely than long ones.

In contrast, our criterion contains two competing terms: While the term $\log P(O_C|C) + \log P(C)$ expresses the quality of a boundary in terms of features and smoothness, the term $P(O_C)$ sums the probabilities of all the possible contours in the vicinity of the contour C. The favored contours are not just the ones with high $\log P(O_C|C) + \log P(C)$, but also the ones with low $P(O_C)$. The latter occurs when all the contours in the vicinity of C have very low probabilities, or in other words, when C is the best contour in a given image area.

We show how the normalization term promotes certain contours in the simple case where all the candidate contours except the optimal one have the same cost (Fig. 2). Assume the desired contour connecting points A and B is the straight



a: single best C **b:** multiple good C's

Fig. 2. The thickness of the line indicates the quality of the boundary. **a)** There is a single best candidate connecting points A and B (straight line). **b)** There exist several good candidate contours connecting points A and B.

line C^*. In Fig. 2a, C^* is a much better candidate than other possible contours connecting A and B, unlike Fig. 2b where other good candidates exist as well. In both cases the quantity $P(O_C|C^*)\ P(C^*)$ is α. The corresponding quantities $P(O_C|C)P(C)$ for all other candidate contours C are β and γ for Fig. 2 a and b respectively, with $\gamma > \beta$. If there are $n+1$ possible ways of connecting points A and B, then:

$$\log(n\,\beta + \alpha) < \log(n\,\gamma + \alpha)$$
$$\log \alpha - \log(n\,\beta + \alpha) > \log \alpha - \log(n\,\gamma + \alpha)$$

Our criterion favors contours for which there are no other competitors in the same neighborhood. Degenerate solutions of very short contours in image areas with no characteristic features will receive very low probability.

2.2 Analysis: Entropy Interpretation

We show that our criterion (Eqn. 2) can be understood from an entropy point of view. Let $P(O_C|C)\ P(C) = \beta_j$, where each β_j corresponds to a different candidate contour C. β_j is the probabilistic cost of a contour according to Eqn. 1 without the normalization term and we will refer to it as "cost". We also have $P(O_C) = \sum_j \beta_j$. The log probability of Eqn. 2 is a function of β_j:

$$E(\beta_j) = \log \beta_j - \log \sum_i \beta_i \tag{4}$$

The maximum of $E(\beta_j)$ is obtained at 0, since $E(\cdot)$ is the log of a probability distribution:

$$E(\beta_j) = 0 \Rightarrow \beta_j = \beta_j + \sum_{i \neq j} \beta_i \Rightarrow \sum_{i \neq j} \beta_i \simeq 0 \tag{5}$$

The last condition holds when all the contours, except the j-th one, have costs close to 0. The minimum of $E(\beta_j)$ is achieved at $-\infty$ and this occurs when

$$\sum_i \beta_i \simeq \infty \tag{6}$$

i.e., when there are many strong candidate paths in the given image area.

The behavior $E(\beta_j)$ is reminiscent of the inverse behavior of the entropy of a distribution. Most informative or high-entropy distributions are the ones who do not favor any particular data points. For example, the most informative one-dimensional distribution is the uniform distribution. On the other hand, least informative distributions are the ones favoring a single value. $E(\beta_j)$ is maximized when a single candidate contour is assigned high cost and is minimized when all the candidate contours have very high costs.

In fact, we can find an entropy lower bound for $E(\beta_j)$, which offers an interesting interpretation of the normalization term $P(O_C)$. We have:

$$E(\beta_j) \geq \log \beta_j - \sum_i \log \beta_i \geq \log \beta_j - \sum_i \beta_i \log \beta_i \tag{7}$$

where the above holds for $\beta_i < 1$, for all i. The term $-\sum_i \beta_i \log \beta_i$ is a pseudo-entropy term since the costs β_i do not sum up to 1. The entropy of a distribution can be seen as a measure of the uncertainty of the distribution. Thus, the normalization factor $P(O_C)$ can be seen as a confidence measure regarding the image location a contour belongs to. Contours that belong to low-uncertainty image regions, that is, they are the sole candidates, are assigned high cost. On the other hand, contours from high-uncertainty image regions are assigned low costs.

3 Optimization Using Dynamic Programming

Our criterion can be globally optimized with dynamic programming. The algorithm proposed merges the optimization scheme employed by the computer vision community for $P(O_C|C)\ P(C)$ ([18, 2, 19, 3–5]) with the algorithm used to calculate $P(O_C)$ [20]. We first show how to calculate $P(O_C)$ using scaling according to [20]. We then describe some additional approximations needed. Finally, we show how to optimize our criterion in a graph-based framework using Dijkstra's algorithm in low-order polynomial time. Dijkstra's algorithm can globally optimize our criterion when points on the boundary to be extracted are known (either automatically or via user input).

3.1 Calculation of $P(O_C)$

We calculate $P(O_C)$ using the well-known forward-backward algorithm employed in HMM inference problems [20]. Let $C = (c_1, \ldots c_n)$ be the hidden random variables corresponding to a contour of n points. The set of the values each of these random variables can take is $D = \{0, \ldots, 7\}$ corresponding to the possible directions employed by the chain code curve representation. Let also $O_C = \{O_{c_1}, \ldots, O_{c_n}\}$ be the observations associated with the individual contour points.

For the forward-backward algorithm, we define

$$\alpha_i(d_k) = P(O_{c_1}, \ldots, O_{c_i}, q_i = d_k)$$

where q_i is the label assigned to the i-th contour point c_i and d_j takes values from $D = \{0, \ldots, 7\}$. Recursively we can compute:

$$\alpha_1(d_k) = P(O_{c_1}|c_1 = d_k) \tag{8}$$

$$\alpha_{i+1}(d_k) = \left(\sum_{d_j} \alpha_i(d_j) P(c_{i+1} = d_j | c_i = d_k) \right) P(O_{c_{i+1}} | c_{i+1} = d_k) \tag{9}$$

The probability of the observations is now given as:

$$P(O_C) = \sum_{d_j \in D} \alpha_n(d_j) \tag{10}$$

3.2 Scaling

The calculation of $P(O_C)$ involves multiplications of very small quantities and very quickly the results are outside the range of machine precision. To this need we need to apply the scaling procedure in [20], so that each time an $\alpha_i(d_j)$ value is computed, it is scaled by s_i:

$$s_i = \frac{1}{\sum_{d_j} \alpha_i(d_j)} \tag{11}$$

$$\hat{\alpha}_i(d_j) = \frac{\alpha_i(d_j)}{\sum_{d_j} \alpha_i(d_j)} \tag{12}$$

With this scaling method, $\log P(O_C)$ is given by::

$$\log P(O_C) = \sum_{i=1}^{n} \log \frac{1}{s_i} \tag{13}$$

3.3 Approximations

$\log P(O_C)$ is usually computed as $\log \sum_{i=1}^{n} e^{-x_i}$. The summation of exponentials often approaches 0 very fast. To calculate it reliably, we have:

$$\sum_{i=1}^{n} e^{-x_i} = e^{-x_m} \left(1 + \sum_{x_i \neq x_m} e^{-(x_i - x_m)}\right) = e^{-x_m}\ (1+S) \tag{14}$$

$$\text{where} \quad x_m = \min_i x_i \tag{15}$$

$$\text{Therefore,} \quad \log \sum_{i=1}^{n} e^{-x_i} = -x_m + \log(1+S) \tag{16}$$

$$\text{where} \quad S = \sum_{x_i \neq x_m} e^{-(x_i - x_m)} \tag{17}$$

When $|S| < 0.1$, we can use the approximation $\log(1+S) \simeq S$.

3.4 Graph-based Optimization

We use Dijkstra's algorithm [21] to simultaneously compute $P(O_C)$ and find the optimal boundary. We assume that points $\{A_1, A_2, \cdots, A_p\}$ on the desired boundary are given. The optimal boundary passing from $\{A_1, A_2, \cdots, A_p\}$ is the concatenation of the optimal contours connecting A_1 to A_2, A_2 to A_3 and so on.

To find the optimal contour connecting two given points, we represent the image with a graph, where each pixel corresponds to a graph node and each node is connected with its 8 neighbors. The weight between adjacent nodes u, v consists of two terms. The first term is a constituent of $P(O_C|C)\ P(C)$ and the second of $P(O_C)$. Assuming u, v are points on the desired boundary, we have:

$$w(u,v) = \log P(O_u, O_v|u,v) + \log P(v|u)) - \log \sum_{d_j} \hat{\alpha}_v(d_j) \tag{18}$$

where the summation takes place over all possible directions d_j. The calculation of $\hat{\alpha}_v(d_j)$ depends on values calculated at neighboring nodes. Thus, the complexity of Dijkstra's algorithm increases by a small multiplicative factor, equal to the number of different directions d_j (in our case 8).

4 Experimental Validation

We first explain our choices regarding the calculations of $P(O_{c_i}|c_i)$ and $P(c_{i-1}|c_i)$, and then present boundary delineation results on a variety of images.

4.1 Feature Calculations

For reliable computations, we will assume second-order dependencies among the contour points and we will compute $P(O_{\{c_i\}}|c_{i+1}, c_i, c_{i-1})$ and $P(c_{i+1}|c_i, c_{i-1})$, where $O_{\{c_i\}}$ denotes the observations associated with contour points c_{i+1}, c_i, c_{i-1}.

The term $P(O_{\{c_i\}}|c_{i+1}, c_i, c_{i-1})$ is computed by estimating how well the pixels in the vicinity of c_{i+1}, c_i, c_{i-1} belong to the two sides of the boundary. The statistical model required for this task is computed based on pixel labeling provided by the user in small image areas. Contour points c_{i+1}, c_i, c_{i-1} divide the pixels in their vicinity into two regions R_I and R_{II}, as shown in Fig. 3. If $M_I(p)$ and $M_{II}(p)$ are two functions estimating how well a pixel p is classified as belonging to side I or side II of the desired boundary, then

$$P(O_{\{c_i\}}|c_{i+1}, c_i, c_{i-1}) = \sum_{p \in R_I} M_I(p) + \sum_{p \in R_{II}} M_{II}(p) \tag{19}$$

The prior $P(c_{i+1}|c_i, c_{i-1})$ is defined so that it takes higher values for contour points forming a straight line than for contour points that form an angle.

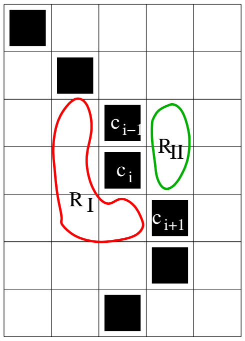


Fig. 3. The data term is calculated based on how likely the pixels in the vicinity of the contour points belong to the two sides of the desired boundary.

4.2 Boundary Delineation Results

We evaluate our formulation in an interactive boundary finding application where the user places seed points sequentially in a manner similar to [3]. Figures 4, 5, and 6 show in three columns the part of the image used to statistically characterize the interior of the object and the background, the delineation results obtained using Eqn. 1 without and with the normalization term. All the results were obtained using the same parameters $\lambda_c = 0.2$, $\lambda_s = 0.1$, and the same training data acquired at the beginning of the delineation process.

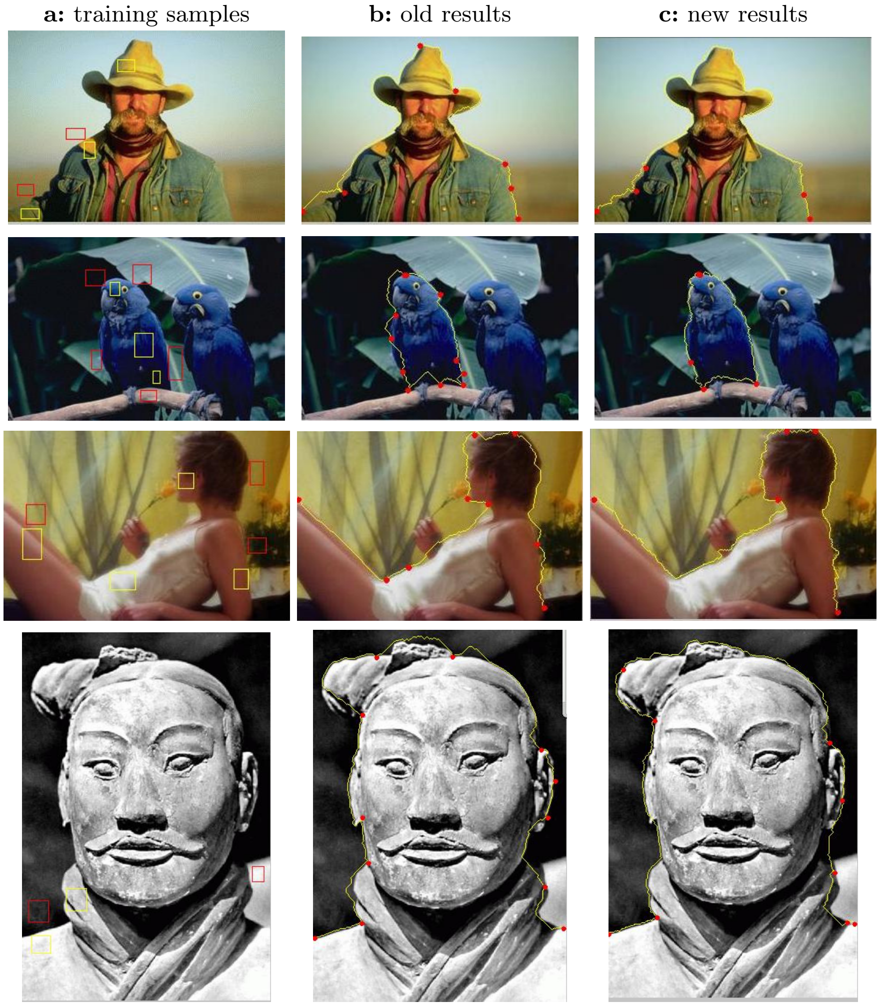


Fig. 4. Segmentation given user clicked boundary points (red dots) on images with complex boundaries. **a)** Windows mark training samples for foreground (yellow) and background (red). **b)** Results from traditional models. **c)** Our results.

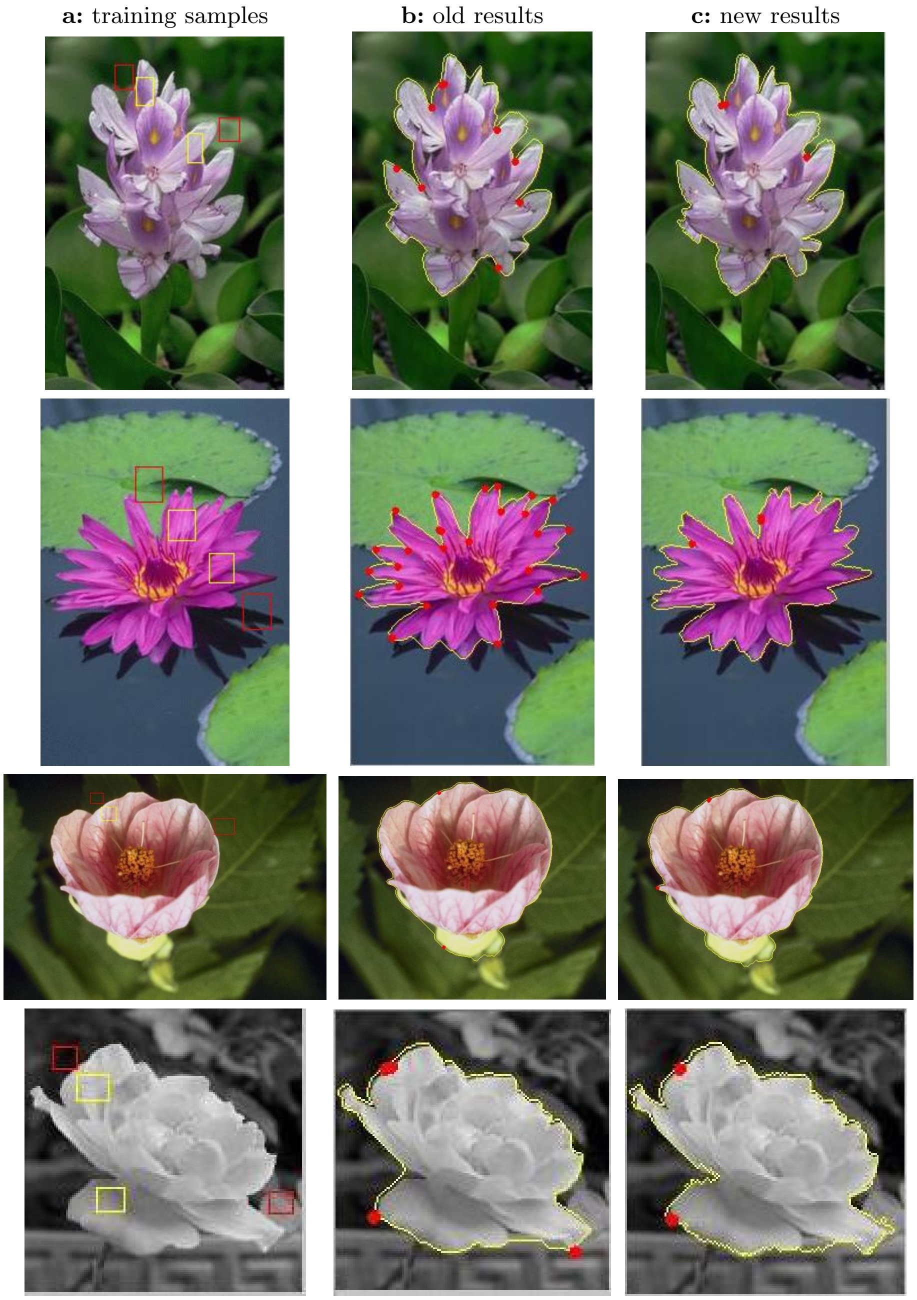
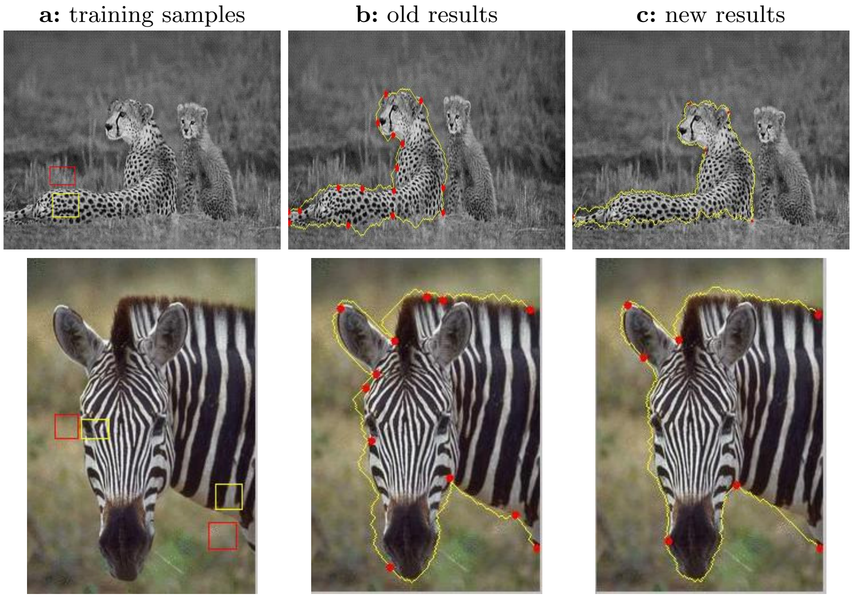


Fig. 5. Color image segmentation given user clicked boundary points (red dots). **a)** Windows mark training samples for foreground (yellow) and background (red). **b)** Results from traditional models that ignore the normalization term. **c)** Our results.



a: training samples **b:** old results **c:** new results

Fig. 6. Texture image segmentation results. Same convention as Fig. 5.

Table 1 summarizes the number of mouse clicks required to delineate the object boundaries, as a measure of the method effectiveness. Since the user is always part of the interactive segmetnation system, the correctness of the output of the segmentation algorithm is not to be contested.

image name	# clicks in old method	# clicks in our method
cheetah	16	**5**
zebra	12	**7**
cowboy	7	**5**
parrots	10	**4**
iris	8	**2**
pink flower	2	2
fuchsia flower	24	**2**
white flower	3	**2**
statue	12	**9**
woman	9	**5**
peppers1	2	2
peppers2	7	**4**
cover1	2	2
cover2	7	**3**
liver1	4	4
liver2	**4**	5
liver3	6	**3**

Table 1. Number of mouse clicks required to delineate the various boundaries using the traditional formulation and our probabilistic criterion.

Our criterion significantly outperforms traditional Bayesian formulations. The most drastic difference between the amount of mouse clicks required to delineate a boundary is for the image “cheetah”. The reason for this can be seen in Fig. 7, where the classification of the image pixels is shown. These results were obtained using the classifier trained on the data shown in Fig. 6. The results are very noisy and the original criterion has trouble localizing the object boundary. On the other hand, the observation-dependent criterion is quite effective in eliminating the noise and tracking the desired discontinuities. In general, the probabilistic criterion consistently produces contours that adhere to the object boundary more faithfully, with fewer mouse clicks.

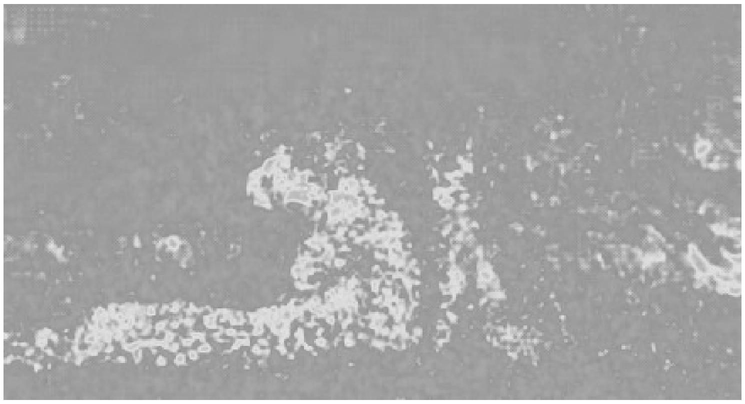


Fig. 7. Classification results using the classifier built from the training data in Fig. 6

.

5 Conclusions

Traditional Bayesian criteria for boundaries assume that the evidence for a candidate contour is derived from the entire image, and the normalization term can thus be omitted during optimization. In practice however, evidence in the vicinity of the contour is employed, and the normalization term is contour-specific and cannot be ignored. Our formulation explicitly acknowledges this term and it extacts the boudaries optimal in both appearance and area of support.

The normalization term helps promote contours that are better than alternatives in their vicinity. Consequently, it alleviates the length bias problem present in traditional Bayesian formulations. Degenerate solutions such as short contours in featureless image areas are no longer favored by design.

Our formulation has the same asymptotic complexity as previous Bayesian formulations, as it can be globally optimized with dynamic programming.

We validate our method with an interactive boundary delineation application, where significantly fewer mouse clicks are needed to extract complex boundaries.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. In: Int'l Conference on Computer Vision, IEEE (1987)
2. Geiger, D., Gupta, A., Costa, L., Vlotzos, J.: Dynamic Programming for Detecting, Tracking, and Matching Deformable Contours. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(3) (1995) 294–302
3. Mortensen, E., Barrett, W.: Intelligent Scissors for Image Composition. In: SIGGRAPH. (1995)
4. Mortensen, E., Barrett, W.: Interactive Segmentation with Intelligent Scissors. Graphical Models and Image Processing **60**(5) (1998)
5. Falcao, A., Udupa, J., Samarasekera, S., Sharma, S.: User-Steered Image Segmentation Paradigms: Live Wire and Live Lane. Graphical Models and Image Processing **60** (1998) 233–260
6. Neuenschwander, W., Fua, P., Iverson, L., Szekely, G., Kubler, O.: Ziplock Snakes. Int'l Journal of Computer Vision **25**(3) (1997) 191–201
7. Mortensen, E., Barrett, W.: A Confidence Measure for Boundary Detection and Object Selection. In: Proc. Computer Vision and Pattern Recognition, IEEE (2001)
8. Mortensen, E., Jia, J.: A Bayesian Network Framework for RealTime Object Selection. In: Proc. Workshop on Perceptual Organization on Computer Vision, IEEE (2004)
9. Cohen, L.: On Active Contour Models and Balloons. Computer Vision Graphics and Image Processing: Image Understanding **52**(2) (1991) 211–218
10. Paragios, N., Deriche, R.: Geodesic Active Contours and Level Set Methods for Supervised Texture Segmentation. International Journal On Computer Vision **46**(3) (2002) 223–247
11. O. Gérard and T. Deschamps and M. Greff and L. D. Cohen: Real-time Interactive Path Extraction with on-the-fly Adaptation of the External Forces. In: European Conference in Computer Vision. (2002)
12. Sullivan, J., Blake, A., Isard, M., MacCormick, J.: Bayesian Object Localisation in Images. Int. J. Comput. Vision **44**(2) (2001) 111–135
13. Allili, M., Ziou, D.: Active contours for video object tracking using region, boundary and shape information. Signal, Image and Video Processing **1**(2) (2007) 101–17
14. Joshi, S.H., Srivastava, A.: Intrinsic bayesian active contours for extraction of object boundaries in images. Int. J. Comput. Vision **81**(3) (2009) 331–355
15. Jermyn, I., Ishikawa, H.: Globally optimal regions and boundaries as minimum ratio weight cycles. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(10) (2001) 1075–1088
16. Schoenemann, T., Cremers, D.: Globally optimal image segmentation with an elastic shape prior. In: IEEE International Conference on Computer Vision (ICCV). (October 2007)
17. Rosenfeld, A., Kak, A.C.: Digital Picture Processing. Academic Press (1982)
18. Amini, A., Weymouth, T., Jain, R.: Using Dynamic Programming for Solving Variational Problems in Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(9) (1990) 855–867
19. Cohen, L., Kimmel, R.: Global Minimum for Active Contour Models: A Minimal Path Approach. International Journal on Computer Vision **24**(1) (1997) 57–78
20. Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE **77**(2) (1989)
21. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. McGraw Hill (1990)