MULTIPLE SNAKES: A GUIDING SCHEME FOR OBJECT SEGMENTATION

Thitiwan Srinark and Chandra Kambhamettu

Video/Image Modeling and Synthesis (VIMS) Lab.
Department of Computer and Information Sciences
University of Delaware, Newark, DE 19716
srinark/chandra@cis.udel.edu
http://www.cis.udel.edu/˜vims

ABSTRACT

A framework for object segmentation using deformable contours is presented. This framework aims to extract multiple objects whose shapes are similar, but whose image qualities are different. Objects with good image qualities are called "strong objects," and objects with bad image qualities are called "weak objects." Our framework is based on the deformable contours method where a new kind of energy is introduced to handle sharing of properties between objects, and also to allow contours of strong objects to guide the deformation of contours of weak objects. Weighting parameters are automatically generated and used as control parameters in the contour guiding scheme. Dynamic programming is used for implementation of the multiple snake optimization framework. Comprehensive testing has been performed on natural images and synthetic images and the results are encouraging.

1. INTRODUCTION

We present a general scheme to perform segmentation of multiple objects under the condition that the shape of objects in the image are similar to each other, but these objects have different image qualities. This paper is an extension of the work in [1]. In [1], the authors handle only symmetric objects, and require the usage of a geometric model in segmentation. In real world, objects are not symmetric, they have arbitrary orientation with respect to each other, and cannot be represented by geometric models. In this work, we focus on the study of segmenting similar objects whose shapes can be arbitrary, and whose orientations can be different. Our method is based on snakes or active contours [2].

There has been significant area of research on the active contours. Cohen *et. al* introduced balloon forces on active contour models [3, 4]. Akgul *et. al* implemented tracking of tongue surface from ultrasound image sequences under additional spatial and temporal constraints [5]. Abe *et. al* [6] proposed use of multiple active contours to extract an object region. Chalana *et. al* used the multiple active contour model for cardiac boundary detection on echocardiographic sequences [7]. It may be noted that none of the previous methods can be applied to extract objects with bad image qualities or missing shape information.

In this work, conceptually every object has its own contour or snake that estimates boundary of the object by using its gradient information. As different objects have dissimilar image qualities, the objects with good image qualities are called "strong objects," and the objects with bad image qualities are called "weak objects." Objects are said to have good image qualities when they can be easily distinguishable from the background. In other words, object features such as edges are not disrupted by illumination, other objects, or noise. In contrast, objects with bad image qualities have unclear, noisy and/or discontinuous edge boundaries. Boundaries of strong objects are easy to segment, while boundaries of weak objects are difficult to detect. Thus in this framework, we use the contours of strong objects as templates or prototypes for weak objects' contour extraction whereby the contours of weak objects are guided by. Our framework is different from deformable template such that it does not require a prior shape model of objects from the user. The shape template is extracted from the image.

This paper is organized as follows. In Section 2, we explain the multiple snake energy formulation. In Section 3, we present the multiple snake optimization framework. In Section 4, we demonstrate experiments and segmentation results. Finally, in Section 5, we present our conclusions.

2. MULTIPLE SNAKE ENERGY FORMULATION

The snake energy is reformulated to serve this scheme by proposing a new kind of energy. This energy measures similarity between object contours and templates. It is called "group energy," and we classify the internal and external energies as "individual energies," because they are derived

from features of each individual object. Additionally in some applications there may be some restrictions or preferences with regards to the spatial relation between the individual objects (such as no-overlap, etc.). Thus, the third energy, constraint energy, is defined to include these restrictions or preferences.

In our framework, we have an assumption that estimated centroids of objects are pre-defined, so that our active contours expand outward, toward the objects' salient features. Suppose there are M objects in the image, and each object has N contour points. Let (x_{kc}, y_{kc}) be the estimated center point of the object k, where $1 \leq k \leq M$. Let (x_{ki}, y_{ki}) be a contour point of the object k, where $1 \leq i \leq N$.

2.1. Individual Energy

The individual energy consists of two energies, internal and external. In practice, the discretized version of the formulation is used because it is more flexible and computationally less expensive [5]. The individual energy of the object k is defined as

$$E_{individual}(k) = E_{ext}(k) + E_{int}(k).$$

2.1.1. Internal energy

The internal energy is to preserve smoothness and continuity. Let v_i represent the contour point (x_i, y_i). The internal energy at the contour point v_i is defined as

$$S(i) = \alpha_i |v_i - v_{i-1}|^2 + \beta_i |v_{i-1} - 2v_i + v_{i+1}|^2,$$

where v_{i-1}, v_i and v_{i+1} are consecutive contour points. The parameters α_i and β_i are weighting parameters of the snake of the object k. The internal energy of the object k is defined as

$$E_{int}(k) = \sum_{i=1}^{N} S(i).$$

2.1.2. External energy

The external energy is represented by gradient of the image. The gradient at any position (x, y) is

$$G(x,y) = -|\nabla I(x,y)|^2 = -\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2\right),$$

where $I(x, y)$ is the image intensity at the location (x, y). The gradient energy of an object snake is calculated at every contour point, thus the total gradient energy, or external energy of the object k is

$$E_{ext}(k) = \sum_{i=1}^{N} G(x_{ki}, y_{ki}).$$

2.2. Group Energy

The group energy measures the variance of properties between objects in the image. In our framework, the group energy deals with shared properties between objects, such as shape. Thus, we define the group energy based on shape similarity between objects. It is a metric that measures how much contours of objects are different from each other. We use this metric to decide how weak contours should be updated in order to converge to its correct shape.

From the active contour model which is a set of ordered points, we like to compare how contours are different. Therefore, we use the internal energy information. Since the internal energy describes smoothness and continuity which are shape properties of objects, we apply them to formulate the group energy.

Suppose the internal energy at a contour point i of the template contour t is $S_t(i)$, and the internal energy at a contour point j of an object k is $S_k(j)$. The group energy between the template and the object k is

$$E_{group}(k) = \sum_{i=1}^{N} |S_t(i) - S_k(j)|,$$

where every contour point i from the template t has its corresponding contour point j from the object k.

The problem is how to match corresponding contour points between objects when their orientations are different. We want to approximate how much a list of contour points on these closed contours have been circular shifted compared to the other list. We call the amount of shifting, "shifting factor." The shifting factor is calculated from the internal energy of every contour point of both the template and object k. Suppose we consider two contours, a template contour t and an object contour k. Let the lists of contour points of t and k be $[v_{t1}, v_{t2}, ..., v_{tN}]$ and $[v_{k1}, v_{k2}, ..., v_{kN}]$, respectively. Let lists of corresponding internal energies of t and k be

$$[S_t(1), S_t(2), ..., S_t(N)],$$
$$[S_k(1), S_k(2), ..., S_k(N)].$$

We consider each list of internal energies as a one dimensional discrete curve $S(x)$, where S is the internal energy and $x = 1, 2, ..., N$. We want to find how much a curve is circular shifted in order to get the best match. The best match here is to have the highest similarity, or the lowest difference cost. The cost is defined as

$$K = \sum_{i=1}^{N} |S_t(i) - S_k(i + \varsigma_k)|^2,$$

where ς_k is the shifting factor, and $1 \leq \varsigma_k \leq N$. Consequently, we find ς_k that minimizes the cost K. However, due to the image characteristics of weak contours, they can

misrepresent object shape. Thus, only partial contour information can be used. Therefore, we need to analyze the weak contours before matching so that we segment a weak contour into small segments based on the current external and internal energies of contour points. There are two types of segments, strong and weak.

Let $\overline{E}_{int}$ and $\overline{E}_{ext}$ be the mean internal energy and mean external energy of every contour point in the weak contour k, respectively, or

$$\overline{E}_{int} = \omega_{int} \sum_{i=1}^{N} S_k(i), \overline{E}_{ext} = \omega_{ext} \sum_{i=1}^{N} G(x_{ki}, y_{ki}),$$

where ω_{int} and ω_{ext} are weighting parameters. A contour point is "strong" if its internal energy is less than $\overline{E}_{int}$, and its external energy is greater than $\overline{E}_{ext}$. A contour point is "weak" if its internal energy is larger than $\overline{E}_{int}$, and its external energy is less than $\overline{E}_{ext}$. We consider the longest consecutive strong contour points to be the "strong segment," and these contour points are set to strong-type. Other contour points not in this strong segment are considered to be in the "weak segment," and they are set to weak-type. The list of internal energies of the weak contour k is modified such that internal energy of every contour point with weak-type is set to a value close to zero. Then we match this modified weak internal energy list with the template internal energy list to find the shifting factor.

Moreover, to make the segment matching process more robust to noise, we apply the scale space technique. Curves of the internal energy are evolved at different scales. A curve is considered as a parametric vector equation $\vec{\tau} = S(u)$, where u is an arbitrary parameter. The curve is smoothened by convolving each coordinate of the curve with a Gaussian function. In continuous form, a new smooth curve is

$$S'(u, \sigma) = S(u) * g(u, \sigma),$$

where $*$ denotes convolution, and $g(u, \sigma)$ is the Gaussian function with a width parameter σ. The width parameter is assigned differently to define different scales of curve evolution. The cost function is redefined as

$$K(\sigma) = \sum_{i=1}^{N} |S'_t(i, \sigma) - S'_k(i + \varsigma_k, \sigma)|^2,$$

$$K' = \frac{1}{W} \sum_{\sigma \in \Sigma} \sigma K(\sigma),$$

where $K(\sigma)$ is the cost of difference at scale σ, K' is the total cost of difference calculated from the weighted sum of K, Σ is the set of scales, and $W = \sum_{\sigma \in \Sigma} \sigma$.

In computing the shifting factors, it can be formulated into the correlation problem by using the 1D-FFT (One-Dimensional Fast Fourier Transform) [8] under a condition that the number of contour points must be the power of 2. At each scale σ, the correlation between $S'_t(i, \sigma)$ and $S'_k(i, \sigma)$ is

$$\mathbf{Corr}(S'_t(\sigma), S'_k(\sigma))_i \overset{\mathsf{DFT}}{\Longleftrightarrow} \mathbf{S'_t}(\sigma)\mathbf{S'_k}(\sigma)^*,$$

where $\mathbf{S'_t}$ and $\mathbf{S'_k}$ are the discrete Fourier transforms of S'_t and S'_k, respectively, and the asterisk denotes complex conjugation. Let $\varsigma_{k\sigma}$ be the shifting factor which gives the maximum correlation of two internal energy functions at scaling of σ, or $\varsigma_{k\sigma} = \mathrm{argmax}_i \, \mathbf{Corr}(S'_t(\sigma), S'_k(\sigma))_i$. Therefore, the shifting factor is then computed as

$$\varsigma_k = \frac{1}{W} \sum_{\sigma \in \Sigma} \sigma \varsigma_{k\sigma},$$

where W is defined as before.

From the segment analysis, the weighting parameters for the group energy of strong-typed contour points are set to relatively low values, while the weighting parameters of weak-typed contour points are relatively high since weak-typed contour points are needed to be guided by the template for their deformation. The weighting parameter for the group energy at a contour point is thus defined as follows:

$$\rho_i = (D_{max} - D_i)/(D_{max} - D_{min}),$$

$$D_i = |S_t(i) - S_k(i + \varsigma_k)|^2,$$

where ρ_i is the weighting parameter for the group energy at v_{ki}, D_i is the difference between the internal energy of the template and weak contour at v_{ki}, and D_{max} and D_{min} are maximum and minimum differences of every D_i, respectively. Finally, the group energy of an object k is

$$E_{group}(k) = \sum_{i=1}^{N} \rho_i |S_t(i) - S_k(i + \varsigma_k)|.$$

3. MULTIPLE SNAKE OPTIMIZATION FORMULATION

As mentioned before, we apply active contours to represent object boundaries, and use a contour as a template to guide deformation of the other contours during the optimization process. This template contour is created from a strong object in the image. The strong object is automatically selected using a heuristic function. The function is computed from the intensity and gradient of pixels inside the contour of each object. This contour is simply computed by minimizing the individual snake energy. Let σ_i and σ_g be the standard deviation of intensity and gradient of pixels inside the object k contour. The heuristic function of the object k is thus defined as

$$h(k) = \alpha_h \sigma_i + \beta_h \sigma_g,$$

where α_h and β_h are weighting parameters. We select the object which gives the smallest heuristic value to be the strong object.

After a strong object is selected, we set up the template from the strong object's contour, and also contours of weak objects by minimizing the individual energy. We then calculate shifting factors and weighting parameters from contours of weak objects and the template.

Next we initialize weak contours from the template contour as follows. First we calculate the translation vector, T_k, from the center of the template object to the center of each weak object. Then we translate the template contour to each weak object by T_k. In case that weak objects are not oriented along the template object, the contour template must be first rotated according to the shifting factor before being translated to weak objects.

The last step is to minimize the snake energy consisting of both the individual and group energies of every weak object, and also the constraint energy in case it is included. The optimization technique is based on dynamic programming [9, 5]. This optimization continues until there is no significant change in the snake energy, and/or the constraint energy is held. Fig. 1 explains the algorithm for our multiple snake optimization framework.

4. EXPERIMENTS AND RESULTS

Our framework is tested with both natural images and synthetic images. In case of natural images, results from applying the multiple snake framework are compared with results from applying the simple snake, which includes only the individual energy. In the case of synthetic images, we evaluated the framework's performance by measuring accuracy of result contours when compared to the ground truth.

4.1. Natural images

We tested the framework with two natural images, consisting of leaves and cherries. The image of "leaves" is shown in Fig. 2 (a) and the image of "cherries" is shown in Fig. 2 (b). Objects that we want to segment have been alphabetically labeled. In both images, an approximate center of each object that we want to segment is defined. For strong object selection, the top-left leaf and the top-left cherry are chosen by the heuristic function to be the strong objects in the leaf image and cherry image, respectively. The comparison results are illustrated in Fig. 3, and Fig. 4.

From the experiments, in the leaf image, overall leaves look similar, but actually each leaf has its own unique shape. This makes segmentation more difficult. However, our framework still can approximate each leaf shape and gives better results than results from applying the simple snake. In the cherry image, the cherry shape is less complex than the leaf

STEP 1 Compute the contour of each object by minimizing the individual energy,

$$E_{individual}(t) = E_{ext}(t) + E_{int}(t)$$

STEP 2 Apply the heuristic function to select the strong object

$$h(k) = \alpha_h \sigma_i + \beta_h \sigma_g$$

STEP 3 Set up the template contour t of from the contour of the strong object

STEP 4 For each weak object k, which has N contour points,

- **(a)** Set up the contour of the weak object from the contour gotten in **STEP 1**
- **(b)** Calculate the shifting factor ς_k for each weak object by minimizing the cost function K', which is the difference between the internal energies of the template and the weak object k at different scales of contour evolution
- **(c)** Calculate the weighting parameter, ρ_i, of the group energy for each contour point
- **(d)** Initialize the weak contour from the template contour
- **(e)** Apply dynamic programming to minimize the total snake energy and the constraint energy (if it is defined) to generate the result contour for the weak object k

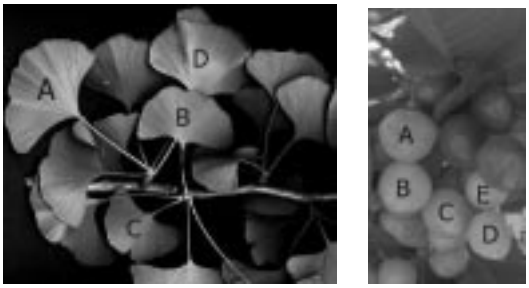
$$E_{total}(k) = E_{individual}(k) + E_{group}(k)$$

$$E_{group}(k) = \sum_{i=1}^{N} \rho_i |S_t(i) - S_k(i + \varsigma_k)|$$

The result contour is returned when the change of the snake energy or the constraint energy are less than some pre-defined thresholds.

STEP 5 Test constraints across objects (in case they are defined); if these constraints hold, return the contour results, otherwise, repeat **STEP 4** by skipping **(a)**, **(b)**, **(c)** and **(d)**

Fig. 1. Algorithm for the Multiple Snake Optimization.



(a) (b)

Fig. 2. (a). Leaf image (from Department of Botany, UW-Madison: http://botit.botany.wise.edu); (b). Cherry image (from the Hiddenvilla organization: www.hiddenvialla.org).

shape. Our framework performs excellently and gives better results than results from applying the simple snake.

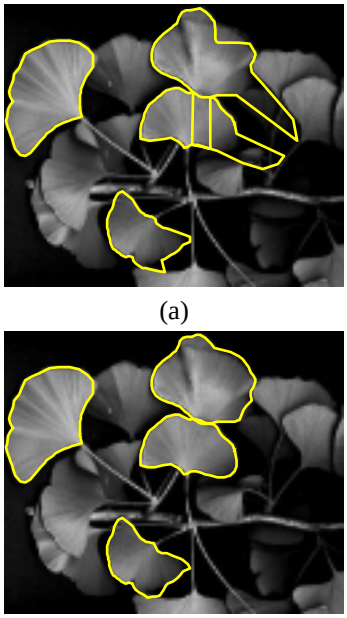
4.2. Synthetic images

We tested the framework with two synthetic images. The first one is a noise-free image with good image qualities. We apply the simple snake optimization to segment objects in this image. The result contours are used as our ground truth. The second image is from modification of the first image. Objects in this image are distorted with noise. Fig. 5 (a) and Fig. 6 (a) illustrate the first and second images, respectively. Fig. 5 (b) shows the ground truth contours.

We applied both the multiple snake and the simple snake optimizations to segment objects in the noisy image. To apply the multiple snake framework to the noisy image, the template object is assigned to the light bulb which is on the top-left of the image, and the other objects are considered as weak objects. The resulting contours of both experiments are then validated by comparing with the ground truth. For results of each optimization, we measure a mean sum of squared distances (MSSD) between result contours and the ground truth contours. Formally, given two deformable contours $U = [u_1, u_2, ..., u_n]$ and $V = [v_1, v_2, ..., v_n]$, the difference between two contours is calculated as [10],

$$\text{MSSD}(U,V) = (\sum_{i=1}^{n} \min_j (v_i - u_j)^2 + \sum_{i=1}^{n} \min_j (u_i - v_j)^2)/2n.$$

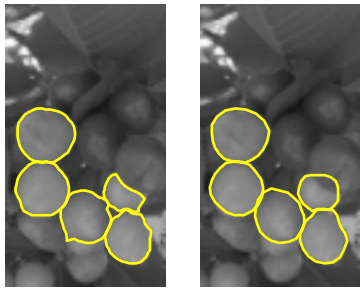
Fig. 6 (b) demonstrates contour results from applying the simple snake optimization, and Fig. 6 (c) shows contour results from applying our multiple snake framework. Table 1 shows MSSD values of the contour results of light bulbs. From the MSSD table, all of the MSSD values of the multiple snake are less than those of the simple snake. From the segmentation results in Fig. 6, one can see that the contour results of the multiple snake visually show better approximation of object boundaries than the contour results of the



(a)

(b)

Fig. 3. Leaf images: results of segmenting four leaves (A, B, C, and D): (a) shows results from applying the simple snake; (b) shows results from applying the multiple snake

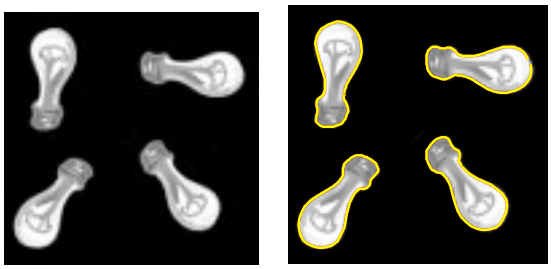


(a) (b)

Fig. 4. Cherry images: results of segmenting five cherries (A, B, C, D, and E): (a) shows results from applying the simple snake; (b) shows results from applying the multiple snake.

Light Bulbs	Simple Snake	Multiple Snake
Top-right	6.6095	3.1939
Bottom-right	2.3597	1.2947
Bottom-Left	3.3411	1.9876

Table 1. The MSSD values (unit in pixels) comparing the contour results of light bulbs with the ground truth using the simple snake and multiple snake techniques.



(a) (b)

Fig. 5. Light bulb images: (a). The original noise-free image; (b). Ground truth contours from applying the simple snake to the noise-free light bulb image.

simple snake, and also that the multiple snake is more robust to noise than the simple snake.

5. CONCLUSION

We present a novel framework for the multiple snake optimization. The framework includes basic snake energies as the individual energy, and introduces a new kind of energy called group energy. The group energy incorporates sharing of information between objects and supports the snake guiding strategy. This strategy allows weak snakes to be guided by strong snakes to deform themselves to proper shape, thus fitting to the boundaries of objects. Our framework can also handle arbitrary shape objects which are arbitrary oriented with respect to each other.

The framework has been applied to segment objects in natural images such as images of leaves and cherries. We quantify the framework performance by testing it with synthetic images. From the experiments, it is seen that the results of using both individual and group energies in the optimization are better and more robust to noise than using only the individual energy,

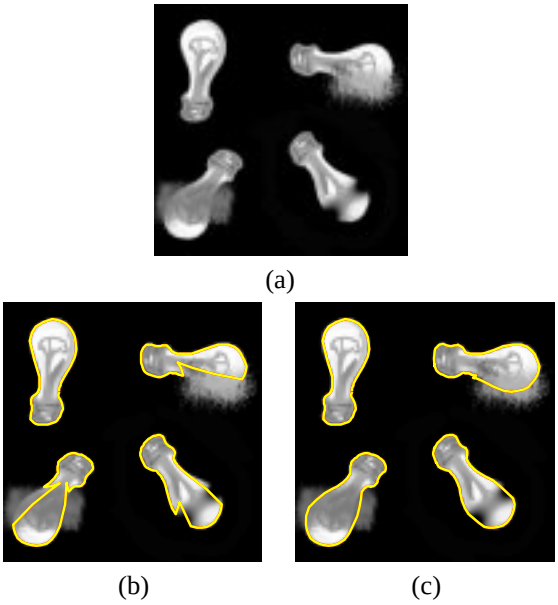
6. ACKNOWLEDGMENTS

The authors would like to acknowledge funding from NSF Grant CISE CDA-9703088. We would like to thank the Department of Botany, University of Wisconsin-Madison, and the Hidden Villa Organization for providing us the leaf image and the cherry image, respectively.

7. REFERENCES

[1] T. Srinark and C. Kambhamettu. A framework for multiple snake. In *Proceedings of Computer Vision and Pattern Recognition*, volume 2, 2001.

[2] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Computer Vision*, 1987.



(a)

(b) (c)

Fig. 6. Light bulb images: (a). The modified version of the original light bulb image by adding noise; (b). Results contours by the simple snake; (c). Results contours by the multiple snake

[3] L. D. Cohen. Note on active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, 1991.

[4] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.

[5] Y. S. Akgul and C. Kambhamettu. A scale-space based approach for deformable contour optimization. In *Proceedings of 2nd Int. Conf. on Scale-Space Theories in Computer Vision*, 1999.

[6] T. Abe and Y. Matsuzawa. A region extraction method using multiple active contour model. In *Proceedings of Computer Vision and Pattern Recognition*, 2000.

[7] V. Chalana, D. T. Linker, D. R. Haynor, and Y. Kim. A multiple active contour model for cardiac boundary detection on echocardiographic sequences. *IEEE Trans. on Medical Imaging*, 15:290–298, 1996.

[8] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipe in C : The Art of Scientific Computing*. Cambridge University Press, 1993.

[9] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using Dynamic Programming for Solving Variational Problems in Vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.

[10] V. Chalana and Y. Kim. A methodology for evaluation of boundary detection algorithms on medical images. *IEEE Trans. on Medical Imaging*, 16:642–652, 1997.