

An Adaptive-Focus Deformable Model Using Statistical and Geometric Information

Dinggang Shen and Christos Davatzikos

Abstract—An active contour (snake) model is presented, with emphasis on medical imaging applications. There are three main novelties in the proposed model. First, an attribute vector is used to characterize the geometric structure around each point of the snake model; the deformable model then deforms in a way that seeks regions with similar attribute vectors. This is in contrast to most deformable models, which deform to nearby edges without considering geometric structure, and it was motivated by the need to establish point-correspondences that have anatomical meaning. Second, an adaptive-focus statistical model has been suggested which allows the deformation of the active contour in each stage to be influenced primarily by the most reliable matches. Third, a deformation mechanism that is robust to local minima is proposed by evaluating the snake energy function on segments of the snake at a time, instead of individual points. Various experimental results show the effectiveness of the proposed model.

Index Terms—Active contour, snake, statistical shape models, adaptive focus deformable model.

1 INTRODUCTION

DEFORMABLE models [1] have been used extensively in image analysis, especially in medical or biological imaging applications [2], [3]. In cases where strong a priori knowledge about the object being analyzed is available, it can be embedded into the formulation of the snake model [4]. For applications in which a training set is available, statistical models can be applied [5]. Cootes et al. [6] have developed a technique for building compact models of the shape and appearance of variable structures in 2D images, based on the statistics of labeled images containing examples of the objects. A hierarchical statistical modeling framework is also developed for representation, segmentation, and tracking of 2D deformable structures in image sequences [7]. Flexible Fourier contour and surface models [8], performing statistics on Fourier parameters [9], were applied into the segmentation of 2D and 3D objects from MRI volume data. Besides, researchers [10] also considered incorporating a type of smoothness constraint into the covariance matrix where neighboring points are correlated. An extensive review of deformable models can be found in [2].

In this paper, we investigate a hierarchical method using a series of affine transformations to deform the initial model onto the desired object. The main characteristics of our model are the following:

1. A model contour is first constructed from a training set. Attached to each point of the model are two kinds of information. First, an *attribute vector* is used, which reflects the shape characteristics of the model around each point, from a local and finer scale to a more global and coarse scale. The attribute vectors are essential in our formulation since they distinguish different parts of a boundary according to their shape properties. Second, our model uses *statistical information* about the expected shape

• The authors are with the Department of Radiology, JHOC 3230, School of Medicine, Johns Hopkins University, 601 N. Caroline St., Baltimore, MD 21287. E-mail: dgshen@cbmv.jhu.edu, hristos@rad.jhu.edu.

Manuscript received 29 June 1999; revised 27 Dec. 1999; accepted 15 May 2000.

Recommended for acceptance by Y.-F. Wang.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110145.

variation, which is expressed in a way that allows the model to emphasize particular aspects of the shape to be reconstructed. This formulation overcomes the limitation of previous statistical shape models [6] in which larger features of a shape dominate over relatively smaller, yet important features merely because their large size influences the measures of shape variability. The adaptive-focus model also allows for a hierarchical formulation in which the focus of the deformable model can be steered each time toward matches of relatively higher confidence.

2. The degree of similarity between a particular snake configuration and the model is not merely Euclidean distance in the 2D or 3D space, but rather the distance between their corresponding attribute vectors.
3. The snake deformation is achieved by employing a hierarchy of transformations of a varying degree of smoothness. Importantly, our model doesn't deform snake points individually, but it deforms segments of the snake at a time. It is shown that this procedure helps the snake avoid local minima.

Our approach is particularly suited to biomedical applications for three reasons. First, the snake does not deform freely to nearby edges. Instead, it looks for points in an edge map that have similar geometric structure. This implies that the points in the model deform toward anatomically corresponding points in a biomedical image. Second, statistical knowledge about the anatomy can be used to restrict the snake deformation to the space of anatomically meaningful shapes. Finally, our model is robust to missing or erroneous data, which is often the case in histological sections.

2 GEOMETRIC DESCRIPTION OF A SHAPE

One of the elements of our deformable model is an attribute vector that is attached to each point along the contour. The attribute vector holds parameters that describe the shape characteristics of the model contour around each point. Throughout this paper, we have used a number of affine-invariants to form the attribute vector; each attribute is the area of a triangle shown in Fig. 1. Areas of larger triangles represent more global properties of the contour. It is not hard to see that the attribute vector corresponding to V_i is different from attribute vectors of other points. We use this fact in the deformation process to preserve the configuration of the snake. For the detailed derivation of the affine-invariant attribute, please refer to [11].

Let's define a contour C by an ordered set of points, $\{V_i = (x_i, y_i) \mid i = 1, 2, \dots, N\}$. For the i th point V_i , its corresponding attribute vector is defined as the areas of R triangles, $F_i = [f_{i,1} \ f_{i,2} \ \dots \ f_{i,R}]^T$. Here, $f_{i,vs}$ is the area of a triangle $V_{[i-vs]} V_i V_{[i+vs]}$ and $[i+vs] = [i+vs+N]\%N$. Notice that, if vs is close to 1, then $f_{i,vs}$ reflects the local shape information. As vs increases, $f_{i,vs}$ gradually captures more global shape information. Therefore, the attribute vector F_i integrates different levels of shape information around the i th point. It can be made exactly affine-invariant by the normalization

$$\hat{F}_i = \frac{F_i}{\sum_{i=1}^N \sum_{vs=1}^R |f_{i,vs}|},$$

where $\hat{F}_i = [\hat{f}_{i,1} \ \hat{f}_{i,2} \ \dots \ \hat{f}_{i,R}]^T$. Thus, the shape information of the curve C can be described by a set of affine-invariant attribute vectors, $\{\hat{F}_i, i = 1, 2, \dots, N\}$.

3 HIERARCHICAL SNAKE DEFORMATION

In this section, a hierarchical snake deformation mechanism is proposed. In this mechanism, the segments of the model seek

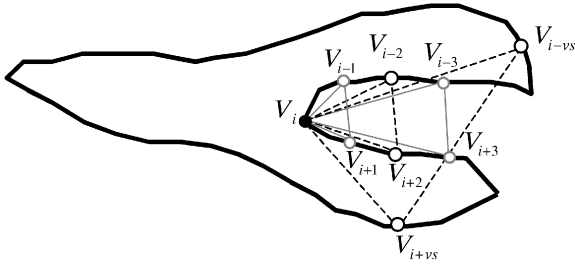


Fig. 1. Schematic representation of the concept of the "attribute vector" on the i th point. The area of a triangle V_{i-1}, V_i, V_{i+1} is used as the v th element of the i th attribute vector F_i . Here, $1 \leq v \leq R$.

image boundaries with the similar shape structure and are not simply influenced by nearby edges. This is achieved by an energy term, which measures the degree of similarity between a particular snake configuration and the model, as reflected by the Euclidean distance between their corresponding attribute vectors. This mechanism is described in Section 3.1. In Section 3.2, a hierarchy of global and local affine transformations is designed as a deformation strategy for the snake. This deformation is very robust to local minima since it deforms the segments of the snake, not individual points. Finally, in order to capture the finest details of the boundary of interest, a technique based on a local-curve fitting is presented in Section 3.3. In the following, the definition of our snake energy is given.

Let's define a snake C_{snake} in a way analogous to the definition of the curve in Section 2, as an ordered set of snaxels, $\{V_i = (x_i, y_i) \mid i = 1, 2, \dots, N\}$. The i th snake segment is defined as the set of $(2R + 1)$ points, $\{V_{[i+vs]} \mid -R \leq vs \leq R\}$ around the i th snaxel V_i (See Fig. 2a, for example). As customary, the total snake energy is defined as the weighted summation of several energy terms:

$$E_{snake} = \sum_{i=1}^N \omega_i E_i = \sum_{i=1}^N \omega_i (E_i^{model} + E_i^{data}), \quad (1)$$

where ω_i is a weighting parameter for the i th snaxel. E_i is composed of two terms: E_i^{model} and E_i^{data} . The term E_i^{model} defines the degree of similarity between the snake and the model. The term

E_i^{data} defines the external energy, aiming at deforming the snake towards a boundary in the image.

3.1 Snake Energy Definition

We now define the model energy term, E_i^{model} , of (1) as the difference between the snake's attribute vector and the model's attribute vector:

$$E_i^{model} = \sum_{vs=1}^R \delta_{vs} \left(\hat{f}_{i,vs}^{Snk} - \hat{f}_{i,vs}^{Mdl} \right)^2, \quad (2)$$

where $\hat{f}_{i,vs}^{Snk}$ and $\hat{f}_{i,vs}^{Mdl}$ are the normalized attribute elements (areas of triangles), respectively, for the snake and the model. The model can either be a representative shape or the result of an averaging procedure. The parameter $\delta_{[vs]}$ is the degree of importance of the v th attribute element $\hat{f}_{i,vs}$ (or the point $V_{[i+vs]}$) in the segment under consideration. R is the number of geometric attributes and the length of snake segment.

The data energy term is usually designed for moving the snake toward a boundary of interest in the image. Accordingly, we require that, in the position of each snaxel V_i , the magnitude of the image gradient is high and the direction of image gradient is similar to the normal vector of the snake. Since we suggest deforming the whole snake segment around each snaxel V_i at a time, the data energy term E_i^{data} for the i th snake segment is obtained by summing individual terms along the i th snake segment:

$$E_i^{data} = \sum_{vs=-R}^R \delta_{[vs]} \left(1 - \left| \nabla I(V_{[i+vs]}) \right| \cdot \left| \vec{h}(V_{[i+vs]}) \cdot \vec{n}(V_{[i+vs]}) \right| \right), \quad (3)$$

where $|\nabla I(V_i)|$, valued between 0 and 1, is the normalized magnitude of the gradient on the snaxel V_i ; $\vec{h}(V_i)$ is the direction of the gradient; $\vec{n}(V_i)$ is the normal vector of the snake in a given snaxel V_i , directed toward the snake interior.

3.2 Snake Deformation Mechanism

We now describe a greedy deformation algorithm [12] that finds a solution that minimizes the energy of the snake segment. In this procedure, a snake segment of $(2R + 1)$ points is deformed by an affine transformation at each deformation stage. The reason for transforming snake segments by affine transformations is that the value of the model energy term E_i^{model} remains unchanged under

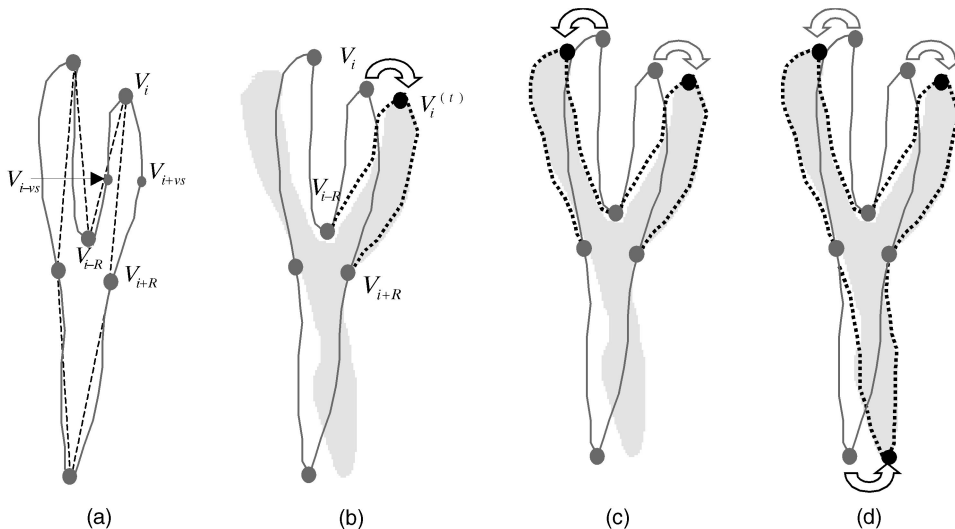


Fig. 2. The definition of the i th snake segment and its affine transformation. The black arrows in (b), (c), (d) indicate the affine transformations of the current snake segments, while the gray arrows represent the affine transformations already completed. (a) the i th snake segment, with $(2R + 1)$ snaxels; (b), (c), (d) the affine transformations of different snake segments to the gray object.

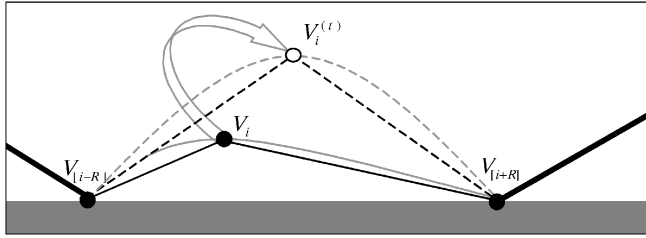


Fig. 3. Constraints imposed on the deformation of a snake segment. Gray smooth curves (dotted or not) represent the segment under transformation, from V_i to $V_i^{(t)}$. See text for detailed meanings.

affine transformation of the i th snake segment. Accordingly, the new configuration of a particular snake segment can be determined directly by minimizing an energy term E_i^{data} that is found by integrating the individual energy terms along the snake segment.

The affine-transformation of a snake segment can be completed by the affine transformation of the triangle $V_{[i-R]}V_iV_{[i+R]}$. Let a tentative position of V_i in the greedy algorithm be $V_i^{(t)}$. Then, this tentative selection of V_i will lead to the shape change of the related triangle from the original triangle $V_{[i-R]}V_iV_{[i+R]}$ to the new triangle $V_{[i-R]}V_i^{(t)}V_{[i+R]}$ (Fig. 2b). The related deformation procedure can be viewed as an affine transformation A . Suppose that all the snaxels ($V_{[i-R]} \sim V_{[i+R]}$) on the i th snake segment move with the three vertices ($V_{[i-R]}$, V_i , and $V_{[i+R]}$) according to the same affine transformation A . Then, the value of the energy term E_i^{model} will remain unchanged. In the greedy algorithm, we only need to examine the match between the currently affine-deformed snaxels and the object boundaries by minimizing E_i^{Data} . Fig. 2b shows the deformation of the i th snake segment, while deformations for the upper-left and the lower snake segments can be found, respectively, in Fig. 2c and Fig. 2d.

In practice, we have to confine the motion domain of the i th snaxel. For example, in Fig. 3, if the i th snaxel V_i moves from the current position to any position in the gray area, this motion will reverse the corresponding segment. Since this transformation, which is determined from three points is applied to the whole snake segment, reversal like the one described above sometimes causes extreme and unrealistic shape deformations. Moreover, if V_i moves into the domain below the thick black lines and above the

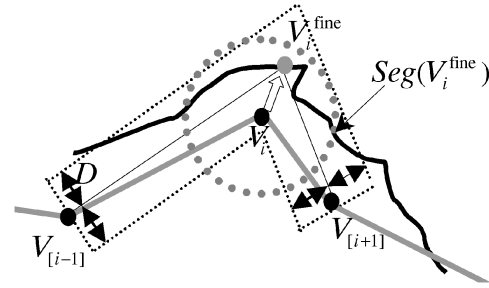


Fig. 4. Local-curve fitting. See the text for the meaning of these points and curves.

gray area, the i th segment will intersect other segments. To avoid such cases of the curve being reversed, the sign of the determinant of A should be positive. Also, to avoid the cases of the curve being self-intersected, the new position of the i th snaxel $V_i^{(t)}$ should be far from the thick black lines.

Our deformation mechanism is implemented hierarchically and it can be summarized next:

1. Use a large value of R to determine the best affine-transformed configuration of the snake segment around V_i . At this point, the number of snake segments that are considered is small and the search area is large.
2. Reduce the value of R to update the affine-transformed configuration of the snake segment around V_i . The number of snake segments that are considered becomes larger, while the search area becomes smaller.
3. Finally, set R equal to 1. Update the affine-transformed snake segment around V_i . At this level, all snaxels are considered.

3.3 Fine Deformation by Local-Curve Fitting

The deformation mechanism in Section 3.2 is very robust, but often at the expense of smoothing out the very fine details of the boundary. Actually, various local features such as curvature can be used together for guiding the snake to exactly localize the desired object boundary [13]. In order to achieve better conformity to the shape of an object, we employ a curve fitting procedure that is described next. This procedure constitutes the final fine-tuning step of our algorithm.

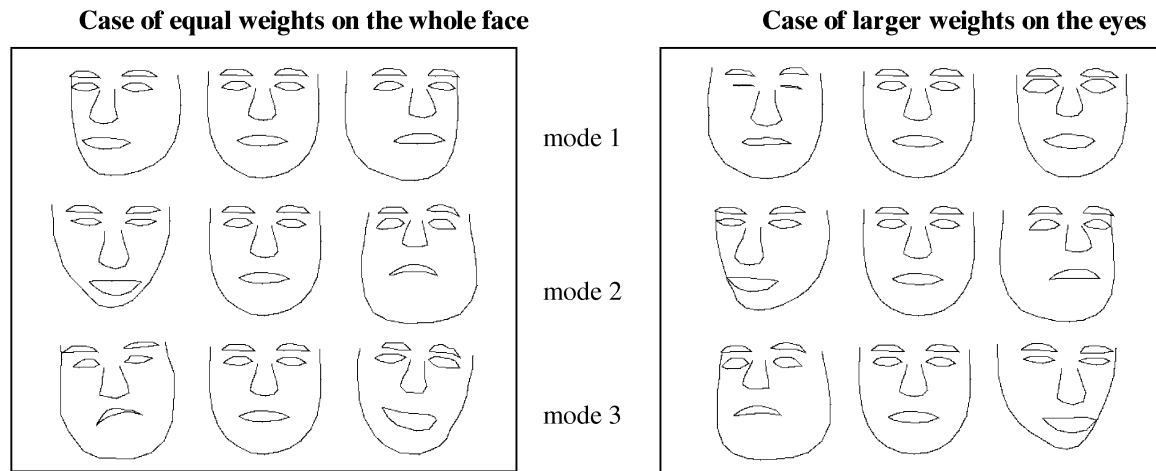


Fig. 5. An illustration of the differential weighting strategy in our adaptive-focus statistical model. The left side of the figure shows the case of equal weights on the whole face, while the right side of the figure shows the case of larger weights assigned to the eyes. For both cases, the face mode i corresponds to the eigenvector with the i th largest eigenvalue and the face shapes in the i th row show the effect of varying parameter of the face mode i in turn between ± 3 s.d.. It can be observed from the left side of the figure that the variation of eyes is not represented by the first three eigenvectors. However, on the right side of the figure, the first face mode clearly incorporates variation of the eyes after emphasizing the landmarks of the eyes.

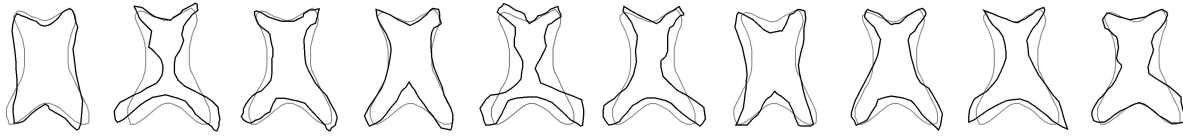


Fig. 6. Some aligned samples that are used in our model. The gray contour denotes the *standard* shape, while the black one denotes the aligned shape. There are in total 118 samples in the training set of ventricles.

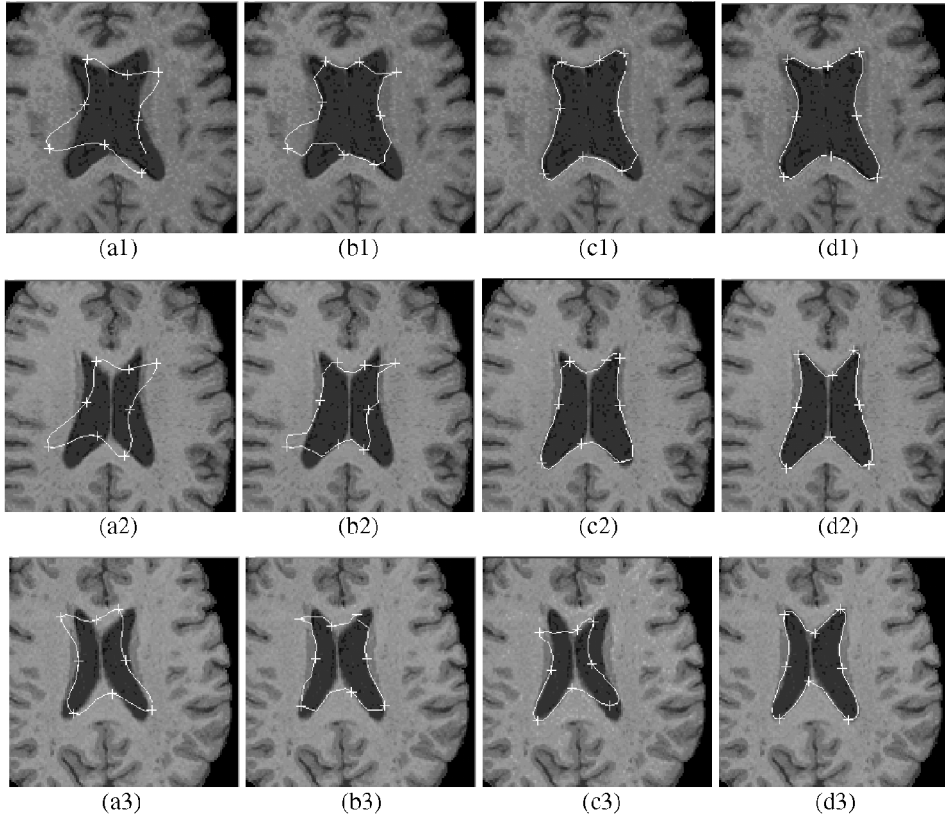


Fig. 7. Qualitative comparisons of our algorithm with the standard snake and ASM. Contours in (a1, a2, a3) are the manual initializations, where crosses “+” are labels used to track individual points; (b1, b2, b3) the result of the standard snake; (c1, c2, c3) the results of ASM; (d1, d2, d3) the results of our method.

In particular, we select the position V_i^{fine} of the snaxel V_i from the nonzero Canny edge points in the neighborhood of the snaxel V_i . This position V_i^{fine} is the one for which the snake segment $\overrightarrow{V_{[i-1]}V_i^{\text{fine}}V_{[i+1]}}$ and the locally-connected image edge segment are in best agreement. The maximum-compression process employed by the Canny edge detector results in edge maps with relatively few nonzero points. Therefore, the search for edges in the neighborhood of a snaxel can be performed fast. In Fig. 4, the neighborhood of the snaxel V_i is drawn as a gray dotted circle. The snake segment $\overrightarrow{V_{[i-1]}V_iV_{[i+1]}}$ is shown as a thick gray curve, connecting three consecutive snaxels $V_{[i-1]}$, V_i and $V_{[i+1]}$. The snake segment $\overrightarrow{V_{[i-1]}V_i^{\text{fine}}V_{[i+1]}}$ resulting from a tentative placement of the i th snaxel at V_i^{fine} is shown as a thin black curve. The locally-connected image edge segment, with the fine point V_i^{fine} on it, is shown as a thick black curve. Notice here, for different selected position of V_i^{fine} , the locally connected image edge segments might be different. In Fig. 4, the value D is used to represent the size of the neighborhood of the local snake curve $\overrightarrow{V_{[i-1]}V_i^{\text{fine}}V_{[i+1]}}$, which is enclosed by the dotted lines. The degree of similarity between the extracted

edge segment $\text{Seg}(V_i^{\text{fine}})$ and the local snake $\overrightarrow{V_{[i-1]}V_i^{\text{fine}}V_{[i+1]}}$ is defined as the total length of the extracted edge segment contained in the neighborhood of $\overrightarrow{V_{[i-1]}V_i^{\text{fine}}V_{[i+1]}}$:

$$\text{Length}(\text{Seg}(V_i^{\text{fine}}), \overrightarrow{V_{[i-1]}V_i^{\text{fine}}V_{[i+1]}}).$$

The procedure of curve fitting, determining the final position of the snaxel V_i , is summarized next:

1. Suppose V_i^{fine} is the position of a selected nonzero Canny edge point in the neighborhood of the snaxel V_i .
2. Regard this Canny edge point as the seed point and then track the connected edge segment, $\text{Seg}(V_i^{\text{fine}})$, of this seed point from Canny edge map. Different V_i^{fine} probably extracts different image edge segment.
3. Calculate the degree of similarity,

$$\text{Length}(\text{Seg}(V_i^{\text{fine}}), \overrightarrow{V_{[i-1]}V_i^{\text{fine}}V_{[i+1]}}),$$

between the extracted edge segment $\text{Seg}(V_i^{\text{fine}})$ and the snake curve $\overrightarrow{V_{[i-1]}V_i^{\text{fine}}V_{[i+1]}}$.

4. The final fine position of V_i is determined by the best position V_i^{fine} , which maximize the fitting degree

TABLE 1
Quantitative Comparisons of Our Algorithm with the Standard Snake and ASM, Using the Results Given in Fig. 7

	Fig.7(a1)		Fig.7(a2)		Fig.7(a3)	
	average dist.	max dist.	average dist.	max dist.	average dist.	max dist.
Standard snake	6.9	28.3	6.8	29.5	3.3	18.4
ASM	2.5	18.0	2.8	11.7	5.9	22.8
Our algorithm	1.3	4.1	1.2	4.0	1.7	4.6

The unit of the value given in the table is at pixel size.

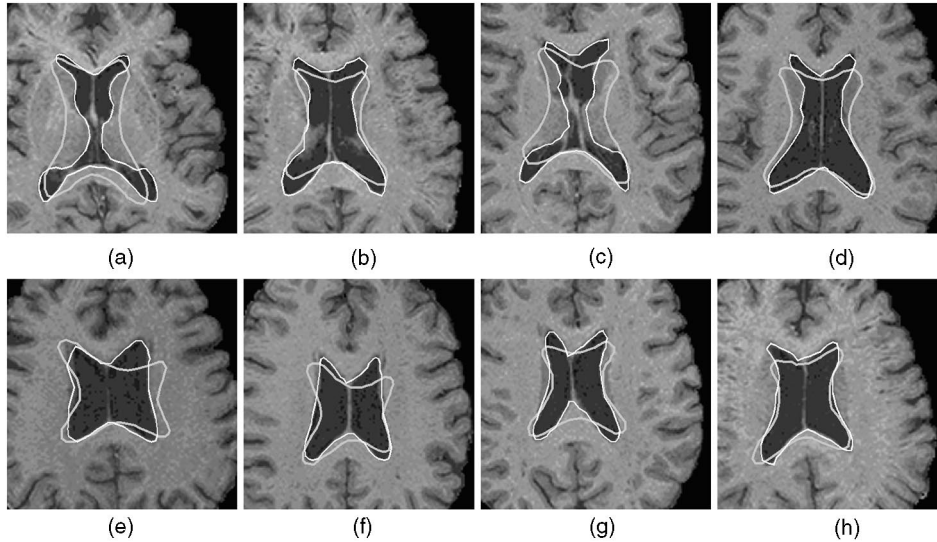


Fig. 8. (a)-(h) More segmentation results of MRI brain images by our algorithm.

$$Length(Seg(V_i^{fine}), \overrightarrow{V_{[i-1]}^{fine} V_{[i+1]}^{fine}})$$

in the whole neighborhood of the snaxel V_i .

4 ADAPTIVE-FOCUS STATISTICAL MODEL

In this section, an adaptive-focus statistical model is suggested. This formulation overcomes the limitation of other statistical shape models. In the following, we give an overview of our formulation by indicating the limitations of previous methods. The brief description of our formulation is given in Sections 4.1 and 4.2.

In ASM [6], all landmarks are treated equally. Consequently, components of the shape that are either more variable or are larger and therefore are comprised of a larger number of landmarks will dominate over either smaller or less variable components. Such components, however, can still be very important and, therefore, should be captured accurately. Moreover, our confidence of a certain component might be relatively higher than the rest of the shape. Therefore, we might want to focus, at first, on those reliable parts of the shape and, subsequently, to shift the focus to other parts as they become closer to their respective targets and therefore more reliable.

As an example, we use face reconstruction. In ASM, the top three eigenvectors do not represent the variations of the eyes, as can be observed on the left side of Fig. 5. The reason is that the size of the eyes is smaller than the sizes of other components of the face, which makes the variation of eyes lower than other components. Moreover, the variation of the eyes is almost independent of the variations of other components. Accordingly, the eigenvectors corresponding to the largest eigenvalues, which are typically used to represent shape variations, do not reflect variation of the eyes. However, for some applications [14], it is very useful to exactly reconstruct the motion of the eyes. Therefore, the statistical model

should be able to capture the variation of all the important components of a shape and, possibly, to adaptively emphasize particular aspects of the shape to be reconstructed. One way to achieve this is via assigning different weights to different components. In particular, smaller and important features can be weighted relatively more. The right side of Fig. 5 shows the effects of assigning large weights (five times larger than others) to the landmarks of the eyes. In the first row of the right side of Fig. 5, the variation of the eyes is exactly represented by the top eigenvector that corresponds to the largest eigenvalue. By varying this face shape parameter between ± 3 s.d., we can capture the opening and closing of the eyes. The face database used in this example is from the Bern face database at <ftp://iamftp.unibe.ch/pub/Images/FaceImages/>.

Besides accounting for differences in the size of a component, differential weighting also provides a means for focusing attention to individual components. For example, in segmenting basal ganglia and ventricular boundaries from brain images (Fig. 10), a priori knowledge is available that reflects our level of confidence in certain parts of the shape under consideration. For example, the ventricle usually has a stronger boundary in the brain image and, thus, is easier to find. Therefore, we can initially focus on the ventricle by assigning a higher weight to it. As the ventricular boundaries are detected, our confidence in the neighboring basal ganglia becomes much higher because of their relatively fixed location relative to the ventricles.

It should be indicated here that a different weighting matrix employed usually leads to a different subspace of training sampling. The reason is that a different weighting matrix makes a different set of complete eigenvectors and, thus, a different set of truncated eigenvectors used to represent the subspace of training samples.

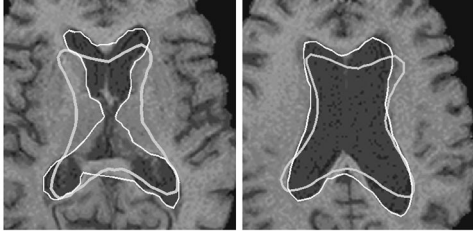


Fig. 9. Examples of the poor performance of our algorithm, due to local minima.

4.1 Shape Alignment and Weighting Strategy

We need to align samples into a common coordinate frame before performing the weighting strategy and the statistics on them. One training sample will be selected as a *standard* shape and others are transformed to best fit this *standard* shape. The alignment technique is obtained from paper [11]. Let's define $\{(x_i^{align}, y_i^{align}) \mid i = 1, 2, \dots, N\}$ as the aligned shape of the training shape $\{V_i \mid i = 1, 2, \dots, N\}$. Fig. 6 shows some examples of shape alignment.

For simplicity, in the notation, the aligned shape is stacked and represented by a $2N$ -element column vector

$$P = [x_1^{align}, y_1^{align}, \dots, x_N^{align}, y_N^{align}]^T.$$

Then, we can weight different points of the snake differently and obtain a weighted vector $S = W \cdot P$, by producing a diagonal weighting matrix W before the column vector P . Here, W is a $2N \times 2N$ matrix, and S is a $2N$ -column vector.

4.2 Mapping the Snake to the Space Derived from the Training Set

Now, we describe how we use the statistical shape information to constrain the deformable model in the space of allowable (or likely) configuration. We first calculate both the *average* vector S_{mean} and the covariance matrix ($2N \times 2N$) from a set of the training vectors, $\{S\}$. $2N$ eigenvectors of the covariance matrix can be calculated and ranked by the size of their corresponding eigenvalues. From the statistical theory, M eigenvectors corresponding to M highest eigenvalues can be selected as the basis of the shape subspace of the training samples. For simplicity, in the notation, we stack these M eigenvectors as a $2N \times M$ -sized matrix H , where each column is one of eigenvectors.

After some algebraic manipulations, we arrive at the following formula for fitting the model instance P' to the point vector P of the aligned snake shape:

$$P' = T_1 \cdot T_2 \cdot P + T_3, \quad (4)$$

where $T_1 = W^{-1} \cdot H$, $T_2 = H^T \cdot W$, and

$$T_3 = (W^{-1} - W^{-1} \cdot H \cdot H^T) S_{mean}.$$

The sizes of the matrices T_1 , T_2 , and T_3 are, respectively, $2N \times M$, $M \times 2N$, and $2N \times 1$. Once obtaining the best model instance P' , we can transform P' back to update the positions of snaxels in the current snake.

4.3 An Algorithm of Statistical Snake Refinement

The algorithm for refining snake by the adaptive-focus statistical model is summarized as follows:

1. Align the snake contour with the selected *standard* shape by using an alignment matrix A^{align} , which is calculated by the alignment method. Afterward, stack the aligned snake as a column point vector

$$P = [x_1^{align}, y_1^{align}, \dots, x_N^{align}, y_N^{align}]^T.$$

2. Use (4) to correct the point vector P into a new vector P' .
3. Update snake contour by transforming P' back to the original coordinate space of the snake contour via the inverse matrix of A^{align} .

5 EXPERIMENTS

The complete algorithm for our method is summarized in Section 5.1. To evaluate our algorithm, two sets of experiments are presented. The first set of experiments (Section 5.2) demonstrates the performance of our whole model in the case of the *identical* weights in the matrix W ; this corresponds to the nonadaptive focus model. The qualitative and quantitative evaluations of our algorithm, compared to the standard snake [15] and ASM, are presented. The second set of experiments (Section 5.3) shows the performances of our adaptive-focus model. In all experiments, the initialization of the snake is provided by the user and the set of training samples is the same. There are a total of 118 samples in the training set of ventricles. See Fig. 6 for some aligned samples.

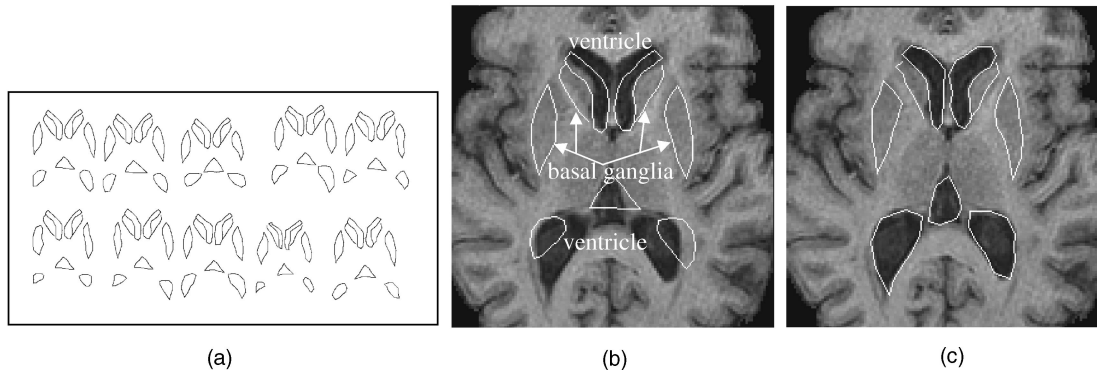


Fig. 10. Example using spatially variable weighting. Larger weights are assigned to ventricles and smaller weights to basal ganglia. (a) Training samples, (b) a 100-point model and a testing image, and (c) final result.

5.1 Complete Algorithm

The multiresolution implementation of our algorithm is reached by subsampling a set of snaxels along the snake contour and deforming their corresponding snake segments. The complete algorithm using both hierarchical affine transforms and adaptive-focus statistics is as follows:

1. Select a small number of snake segments. The length of each snake segment is $(2R + 1)$ snaxels, initially, and therefore the corresponding search area is relatively larger.
2. For every selected snake segment, determine its best affine-transformed configuration by minimizing its external energy term E_i^{Data} . Here, the local affine-transformation technique is employed to locally deform the snake segment that is considered (details in Section 3.2).
3. Align the current snake configuration with the standard model contour by using the affine-transformation matrix A^{align} calculated from the snake to the model. Then, stack the aligned snake as a point vector

$$P = [x_1^{\text{align}}, y_1^{\text{align}}, \dots, x_N^{\text{align}}, y_N^{\text{align}}]^T$$

(Section 4.1).

4. Map the point vector P into the new vector P' using the adaptive-focus statistical model, $P' = T_1 \cdot T_2 \cdot P + T_3$, which is described in Section 4.2. Then, transform P' back into the original coordinate space of the snake via the inverse matrix of A^{align} and update the snake.
5. Increase the number of snake segments to be considered and decrease both the length of snake segment $(2R + 1)$ and the size of search area of every selected snake segment. Notice that decreasing the length of the snake segment is equivalent to decreasing R . If the length of the snake segment is 3, i.e., $R = 1$, then go to Step 6; otherwise, repeat Steps (2-5).
6. Use the local-curve fitting to refine the configuration of the snake (Section 3.3).

5.2 Experimental Results Using Our Whole Model: The Case of Identical Weights

A key property of the proposed method is its tendency to maintain the geometric shape of the snake model during the shape deformation procedure. Fig. 7 qualitatively compares the performances of our method, the standard snake, and ASM in detecting ventricles from the MRI brain images (256×256). Initializations for three different MRI images are provided in Figs. 7a1, 7a2, and 7a3, where crosses “+” are used as labels in order to track the displacements of individual points. Using the standard snake, the final results are shown in Figs. 7b1, 7b2, and 7b3. In addition to being trapped by erroneous edges, the standard snake did not preserve any anatomical homology during the deformation, as reflected by the positions of crosses after the snake deformation. ASM is able to preserve the model shape in the deformation procedure, however, sometimes owing to local minimum problem its final results Figs. 7c1, 7c2, and 7c3 are still unsatisfactory. On the contrary, our method gives good results, as shown in Figs. 7d1, 7d2, and 7d3. Table 1 gives the quantitative comparisons of our algorithm with the standard snake and ASM, based on the results in Fig. 7. The average distance of the final contour from the contour marked by experts is given, at units of pixel. Our algorithm has the least average distances ($1.2 \sim 1.7$ pixels) for all three examples. Also, the maximal distance was calculated. Our algorithm still has the least maximal distances ($4.0 \sim 4.6$ pixels). Obviously, in reality the initialization will not be as bad as the one used in these experiments. We used this initialization to demonstrate the robustness of our algorithm to local minima and therefore to the snake’s initialization, which is known to be very important in most deformable models.

Fig. 8 shows more results of segmenting ventricles from the MRI brain images by using our method. Gray contours denote the initializations and white contours denote the final results. Despite the challenging initializations, the results were very good.

Since we are using a greedy algorithm, we can’t guarantee that all global minima are found. Fig. 9 gives two examples of failure of our algorithm after convergence to local minima. We attribute the failure to the fact that the model and the objects are very dissimilar. In real medical imaging applications, one doesn’t encounter cases like the one in the left of Fig. 9, since the model typically is much more similar to the shape of interest.

5.3 Experimental Results Using the Adaptive-Focus Deformable Model

The goal in this experiment was to segment the basal ganglia and ventricular boundaries from MR brain images. In this application, the weights of the adaptive-focus statistical model are determined by the degrees of our confidences on the components of the shape under consideration. Larger weights are assigned to the landmarks of the ventricles since the ventricles usually have a stronger boundary that is easier to locate. Smaller weights are assigned to the landmarks of the basal ganglia, which have relatively unreliable and fuzzy boundaries often confounded by adjacent cortical edge. Fig. 10b shows a 100-point model, derived from a set of 10 samples (Fig. 10a), on the original testing image (256×256). The deformation is driven primarily by the ventricular boundaries of the model. As those get close to their final positions, the adjacent basal ganglia boundaries also get close to their final positions and therefore become more reliable features to drive the deformation. By using this strategy, a satisfactory result is obtained and shown in Fig. 10c. Notice that the deformation from the initialization to the final result is large (particularly for the lower part of this object).

6 CONCLUSION AND FUTURE WORK

In this paper, we have proposed an adaptive-focus deformable model for segmenting 2D deformable objects from images. The source codes for our adaptive-focus deformable model are freely available from the site, <http://pandora.cbm.v.jhu.edu/~dgshen/SnakeCode.htm>.

Several extensions of our methodology are possible. First, the boundary points are currently used to represent the object shape and also the snake segment. For speeding up our technique, the snake segment can be expressed as a B-spline. Second, extension of our technique to 3D is also possible by redefining the geometric features. Finally, some automatic techniques should be studied. For example, the weighting matrices should be determined automatically by employing learning method. It is particularly important when applying our deformable model to complex medical images.

REFERENCES

- [1] A.K. Jain, Y. Zhong, and S. Lakshamanan, “Object Matching Using Deformable Templates,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 267–278, Mar. 1996.
- [2] T. McInerney and D. Terzopoulos, “Deformable Models in Medical Image Analysis: A Survey,” *Medical Image Analysis*, vol. 1, no. 2, pp. 91–108, 1996.
- [3] C. Davatzikos, “Spatial Transformation and Registration of Brain Images Using Elastically Deformable Models,” *Computer Vision and Image Understanding*, vol. 66, no. 2, pp. 207–222, May 1997.
- [4] K.F. Lai and R.T. Chin, “Deformable Contour: Modeling and Extraction,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, pp. 1,084–1,090, Nov. 1995.
- [5] M. Miller, A. Banerjee, G. Christensen, S. Joshi, N. Khaneja, U. Grenander, and L. Matejic, “Statistical Methods in Computational Anatomy,” *Statistical Methods in Medical Research*, vol. 6, pp. 267–299, 1997.
- [6] T.F. Cootes, D. Cooper, C.J. Taylor, and J. Graham, “Active Shape Models—Their Training and Application,” *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

- [7] C. Kervrann and F. Heitz, "A Hierarchical Markov Modeling Approach for the Segmentation and Tracking of Deformable Shapes," *Graphical Models and Image Processing*, vol. 60, no. 3, pp. 173–195, May 1998.
- [8] G. Szekely, A. Kelemen, C. Brechbuhler, and G. Gerig, "Segmentation of 2-D and 3-D Objects from MRI Volume Data Using Constrained Elastic Deformations of Flexible Fourier Contour and Surface Models," *Medical Image Analysis*, vol. 1, no. 1, pp. 19–34, 1996.
- [9] L.H. Staib and J.S. Duncan, "Boundary Finding with Parametrically Deformable Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1,061–1,075, Nov. 1992.
- [10] Y. Wang and L.H. Staib, "Boundary Finding with Correspondence Using Statistical Shape Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 338–345, June 1998.
- [11] H.H.S. Ip and Dinggang Shen, "An Affine-Invariant Active Contour Model (AI-Snake) for Model-Based Segmentation," *Image and Vision Computing*, vol. 16, no. 2, pp. 135–146, 1998.
- [12] D.J. Williams and M. Shah, "A Fast Algorithm for Active Contours and Curvature Estimation," *Computer Vision, Graphics, and Image Processing*, vol. 55, pp. 14–26, 1992.
- [13] Y.F. Wang and J.F. Wang, "Surface Reconstruction Using Deformable Models with Interior and Boundary Constraints," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 5, pp. 572–579, May 1992.
- [14] L. Yin and A. Basu, "Integrating Active Face Tracking with Model Based Coding," *Pattern Recognition Letters*, vol. 20, no. 6, pp. 651–657, 1999.
- [15] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision*, vol. 1, pp. 321–331, 1988.