

Segmentation and Tracking in Echocardiographic Sequences: Active Contours Guided by Optical Flow Estimates

Ivana Mikić,* Slawomir Krucinski, and James D. Thomas, *Associate Member, IEEE*

Abstract—This paper presents a method for segmentation and tracking of cardiac structures in ultrasound image sequences. The developed algorithm is based on the active contour framework. This approach requires initial placement of the contour close to the desired position in the image, usually an object outline. Best contour shape and position are then calculated, assuming that at this configuration a global energy function, associated with a contour, attains its minimum. Active contours can be used for tracking by selecting a solution from a previous frame as an initial position in a present frame. Such an approach, however, fails for large displacements of the object of interest. This paper presents a technique that incorporates the information on pixel velocities (optical flow) into the estimate of initial contour to enable tracking of fast-moving objects. The algorithm was tested on several ultrasound image sequences, each covering one complete cardiac cycle. The contour successfully tracked boundaries of mitral valve leaflets, aortic root and endocardial borders of the left ventricle. The algorithm-generated outlines were compared against manual tracings by expert physicians. The automated method resulted in contours that were within the boundaries of intraobserver variability.

Index Terms—Active contours, boundary detection, optical flow, snakes, ultrasound.

I. INTRODUCTION

THE algorithm presented in this paper was originally developed for segmentation and tracking of mitral valve leaflets in ultrasound image sequences [1], but was later successfully applied to segmentation of other structures such as the endocardial surface of the left ventricle and the aortic root. Ultrasound image segmentation facilitates both clinical diagnoses of cardiac diseases and research on heart mechanics and function. Computerized studies, such as finite element analysis of cardiac structures, benefit from this dynamic segmentation since results of computer simulations can be verified *in vivo*. This, in turn, could lead to better understanding of cardiac valve mechanics and the mechanisms of valve failure, the

Manuscript received December 16, 1996; revised March 5, 1998. This work was supported by National Aeronautics and Space Administration, Houston, TX, under Grant NCC9-60. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was N. Ayache. Asterisk indicates corresponding author.

*I. Mikić is with the Visual Computing Laboratory, Department of Electrical Engineering, University of California, San Diego, CA 92122 USA (e-mail: ivana@vision.ucsd.edu).

S. Krucinski is with the Department of Biomedical Engineering, Cleveland Clinic Foundation, Cleveland, OH 44195 USA.

J. D. Thomas is with the Department of Cardiology, Cleveland Clinic Foundation, Cleveland, OH 44195 USA.

Publisher Item Identifier S 0278-0062(98)04883-6.

dynamics of ventricular ejection, and the influence of heart disease on cardiac output.

Among medical imaging techniques, ultrasound is particularly attractive because of its good temporal resolution, noninvasiveness and relatively low cost. In clinical practice, analysis of ultrasound images still relies on manual outlines produced by expert physicians. The process of manual outlining is slow and tedious; outlines are subjective and vary between different observers. Effective processing of the information contained in ultrasound images calls for computer-assisted segmentation techniques.

The developed algorithm is based on an active contour framework. Active contours, or snakes, represent a special case of multidimensional deformable models, an actively researched model-based approach to medical image analysis [2]. Section II of this paper gives an overview of some of the existing snake models and techniques for object tracking in image sequences. The proposed algorithm is described and formulated in Section III. The results are presented in Section IV and the conclusion follows in Section V. The algorithm for estimating optical flow is described in the Appendix.

II. BACKGROUND

Active contour models were first proposed by Kass *et al.* [3]. This semiautomated image segmentation technique takes a rough user input on the location of an object and refines the object outline based on the computer-generated edge map. The active contour (snake) \mathbf{v} is defined as a continuous deformable curve

$$\mathbf{v}(s) = \langle x(s), y(s) \rangle \quad (1)$$

where s denotes the normalized contour length. An energy that is a function of the contour's shape and position is associated with the snake. The optimal contour configuration is defined as one having the minimum energy. In general, the energy function consists of the internal and the external energy

$$E_{\text{snake}} = \int_0^1 \left(E_{\text{int}}(\mathbf{v}(s)) + E_{\text{img}}(\mathbf{v}(s)) \right) ds. \quad (2)$$

E_{int} represents the internal energy that makes the contour resistant to bending and stretching, and E_{img} is the energy associated with the image that forces the contour to lock onto the edges. Since the position and shape of the snake are modified by the energy minimization process, the lowest

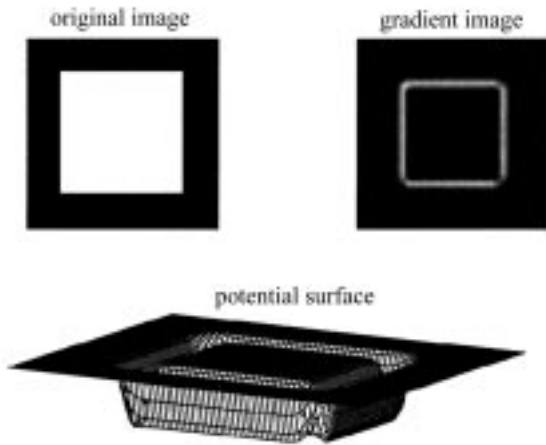


Fig. 1. Example illustrating the concept of a potential surface. The height of the potential surface is equal to the negative of the value of the image gradient at the corresponding location.

values of E_{img} should be at locations where the image gradient is large, i.e., edge locations. Image energy is, therefore, often defined as

$$E_{\text{img}} = -\|\nabla I\|^2 \quad (3)$$

where I is the image intensity. The continuity and smoothness of the model help overcome the problems of gaps and noise in the object boundaries. Various approaches have been proposed for finding the optimal contour. Among them are finite-difference methods [3], [4], [11], finite-element methods [5], [6], [11], dynamic programming [7], [8], and simulated annealing [9], [10].

Leymarie and Levine's model [11] offers an intuitively appealing interpretation of snakes. The inverted image gradient can be visualized as a surface over the image plane. This so-called potential surface has valleys at locations that correspond to the edges in the original image (Fig. 1). If the contour has a mass, it will fall down the slope of the potential surface to the bottom of the valley through the action of gravitational forces. The position of the contour at the bottom of the valley corresponds to locking onto the desired edge in the original image. During the contour fall, the potential energy converts to kinetic energy. The kinetic energy is dissipated by damping, which causes the contour to settle.

Lobregt and Viergever [12] proposed a discrete contour model. In their formulation, the contour is defined as a set of vertices connected by line segments. The length of the segments defines the resolution of the model. Internal forces are based on the position of the vertex and its neighbors, constraining the local curvature. The deformation process is carried out in a number of discrete steps in which acceleration, velocity, and position are updated for each of the vertices. This formulation of the active contour is easier to manipulate since all the equations are discrete.

Objects of complex shapes or unknown topologies are best handled by topology-independent algorithms [13]–[15]. However, *a priori* knowledge about the shape of the object of interest can be incorporated into the deformable model to improve its performance. The use of deformable shape

templates is one approach [16]–[18]. Shape models based on superquadrics can also be used [19], [20]. In the algorithm to be presented, the *a priori* knowledge on the object shape is introduced through the use of forces that try to preserve certain shape features.

The deformable contour model relies on initial placement of the contour in the vicinity of the edge of interest, since the snake settles in local energy minima. If placed too far away, the contour can be attracted by some other edges or just be out of reach of any. Active contours can be used for tracking objects in image sequences by using the optimal contour from the previous frame as the initial contour in the next frame. Such an approach obviously works only for small frame-to-frame displacements of the objects of interest.

A multiscale approach [11] can be used to overcome this problem to some degree. The image forces are computed at several different scales. At the coarsest scale, they attract contours from the largest distances. Once the contour settles, new image forces are computed at the next finer scale and the position of the contour is adjusted. This refinement is repeated until the finest scale is reached. The final contour from the previous frame is still used as the initial contour in the next frame, but now the larger displacements can be handled by the model. However, there is a limitation on the range of scales that can be used and it depends on the size of the object of interest. Therefore, even though the range of displacements that can be accounted for is increased, for problems such as tracking of cardiac valve leaflets, that is still not sufficient.

Geiger *et al.* [8] proposed a method based on dynamic programming for detecting the optimal contour position. In their approach, the tracking is performed by subsampling the contour from the previous frame at high curvature points and using these points as initial ones in the next frame. Selected points define the search window in which the optimal contour is detected through the dynamic programming approach. Although, this algorithm successfully handles deformation, it works on the assumption that the object does not move (translate) much.

A motion model can be designed to describe the dynamic behavior of the snake. It is often used as a system model for the Kalman filter tracker [21], [22]. Motion models can be quite sophisticated and their parameters can be learned using a training sequence [23]. These methods show good results, but, due to the highly irregular and rapid motion of the mitral valve, we believe that a motion model that covers a complete cardiac cycle is not suitable for tracking the mitral valve.

Another tracking algorithm that also uses a Kalman filter to track the object and estimate true motion has been proposed in [24]. The observations for this Kalman filter are optical flow estimates obtained by application of the block-wise Horn and Schunk algorithm. With such observations, the tracker should be able to more successfully handle sudden movements of the object of interest. However, the use of optical flow as the only input to the algorithm means that the current position and shape of the object are estimated by integrating object motion. This results in accumulation of errors in the optical flow estimates over time. This model does not provide a way of incorporating the current image information into the current estimate of object shape and position.

In the algorithm to be presented, the tracking problem is solved by using optical flow estimates as initial conditions when integrating the equation of motion. In other words, optical flow is used to push the contour toward the new position of the object. In the simpler algorithm proposed by Ayache *et al.* [25], an energy measure that tends to preserve the matching of high curvature points and to enforce a smooth field of displacements vectors between contours in two consecutive frames is minimized (such an approach also requires previous edge detection). In the method we propose, instead of matching pieces of contours around high curvature points, through the use of optical flow, the best matches for image regions around each vertex of the contour are found in the next frame. The benefits of the latter approach are notable in the high deformation frames (our algorithm required user intervention in only one frame in three sequences showing the mitral valve, while the algorithm presented in [25] required manual intervention in four frames in one image sequence).

III. METHOD

The formulation presented here is similar to Lymarie and Levine's model [11]. The contour is defined as a set of M vertices, snaxels, connected by line segments. The position of the snake is described by an $M \times 2$ matrix \mathbf{X} , where the i th row contains coordinates of the i th snaxel

$$\mathbf{X} = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ \dots & \dots \\ x_{M-1} & y_{M-1} \end{bmatrix}. \quad (4)$$

The contour motion is a result of interplay between its mechanical properties (defined as its inertia, internal dissipation, and elastic stiffness) and its potential energy (which is inversely proportional to the image gradient). The equation of motion for this model is

$$\mathbf{M}\ddot{\mathbf{X}}(t) + \mathbf{C}\dot{\mathbf{X}}(t) + \mathbf{K}\mathbf{X}(t) = \mathbf{F}(t) \quad (5)$$

where \mathbf{F} is the matrix of x and y components of image forces at each vertex, $\mathbf{M} = \mu \cdot \mathbf{I}_M$ where μ is the mass assigned to each vertex, $\mathbf{C} = \gamma \cdot \mathbf{I}_M$ where γ is the constant damping density and \mathbf{K} is an $M \times M$ stiffness matrix. Dots above the \mathbf{X} represent the derivatives in time.

Two initial conditions are needed for solving the second-order differential equation. If the initial position $\mathbf{X}(0)$ and velocity $\dot{\mathbf{X}}(0)$ are known, (5) has a unique solution. This suggests the way to incorporate the optical flow information into the active contour model. For the first frame of the image sequence, $t = 0$, a user defines the initial contour $\mathbf{X}(0)$, and the initial velocity vector, $\dot{\mathbf{X}}(0)$, is set to zero. Once the contour settles in the first frame, the subsequent initializations are done automatically. The initial position is given by the final position from the preceding frame, and the initial velocities of vertices are given by the optical flow estimates.

Optical flow estimates give the average velocity of the object between two frames, not the initial velocity. They can, however, be used as initial velocities for two reasons. First, we are using a small value for the damping factor, and second, since the final contour position is adjusted through the action of image forces, we only need rough estimates of initial velocity.

To estimate optical flow, we used a multiscale version of Singh's algorithm [26], which is capable of detecting large displacements. The details of that method are given in the Appendix.

A. Integration of the Equation of Motion

1) *Single-Step Time Marching Scheme:* The equation of the dynamic equilibrium of the contour (5) can be integrated using a single-step time-marching scheme (SSTMS) [27]. In the SSTMS, the time t is divided into several intervals, or time steps Δt_i . In each time step, the position and the velocity of the object are updated. Calculations performed for each time interval are

$$\begin{aligned} \text{i)} \quad & \tilde{\mathbf{X}}_{n+1} = \mathbf{X}_n + \dot{\mathbf{X}}_n \Delta t \theta_1 \\ & \dot{\tilde{\mathbf{X}}}_{n+1} = \dot{\mathbf{X}}_n \\ \text{ii)} \quad & \alpha_n = \left(\mathbf{M} + \Delta t \theta_1 \mathbf{C} + \frac{\Delta t^2}{2} \theta_2 \mathbf{K} \right)^{-1} \\ & \quad \times (\mathbf{F} - \mathbf{C}\dot{\tilde{\mathbf{X}}}_{n+1} - \mathbf{K}\tilde{\mathbf{X}}_{n+1}) \\ \text{iii)} \quad & \mathbf{X}_{n+1} = \mathbf{X}_n + \Delta t \dot{\mathbf{X}}_n + \alpha_n \frac{\Delta t^2}{2} \\ & \dot{\mathbf{X}}_{n+1} = \dot{\mathbf{X}}_n + \alpha_n \Delta t \end{aligned} \quad (6)$$

where n is the time step number, $\mathbf{X}_0 = \mathbf{X}(0)$ and $\dot{\mathbf{X}}_0 = \dot{\mathbf{X}}(0)$, and θ_1 and θ_2 are constants. The unconditional stability of the integration is guaranteed [27] if $\theta_1 \geq 0.5$ and $\theta_2 \geq \theta_1$. Steps i)–iii) are repeated for each of the time intervals. The problem is nonlinear if the stiffness matrix \mathbf{K} is not constant and depends on the position of the contour. In such a case, the equation of motion takes a more general form

$$\mathbf{M}\ddot{\mathbf{X}} + \mathbf{C}\dot{\mathbf{X}} + \mathbf{P}(\mathbf{X}) = \mathbf{F}. \quad (7)$$

In this case, the problem is linearized inside a time interval

$$\mathbf{P}(\mathbf{X}) = \mathbf{P}(\mathbf{X}_n) + \bar{\mathbf{K}}\Delta\mathbf{X} \quad (8)$$

where $\bar{\mathbf{K}}$ is the value of \mathbf{K} at some average value of \mathbf{X} inside the current interval. (7) has to be solved iteratively using steps ii) and iii) of the procedure given in (6). It will be shown that the stiffness matrix in our model depends on the position of the contour. However, the internal forces can be decomposed as $\mathbf{P}(\mathbf{X}) = \mathbf{K}(\mathbf{X})\mathbf{X}$, with stiffness matrix \mathbf{K} recalculated at the beginning of each iteration.

2) *Equilibrium Criterion:* The iteration process is terminated when the contour attains the equilibrium at each time step. In the proposed model, the ratio of contour deformation energy in the current and the first iteration was chosen as an equilibrium measure ε

$$\varepsilon = \frac{\sum_{i=0}^{M-1} d\mathbf{F}_i d\mathbf{X}_i}{\sum_{i=0}^{M-1} (d\mathbf{F}_i d\mathbf{X}_i)_0} \quad (9)$$

where $d\mathbf{F} = \mathbf{M}\alpha + \mathbf{C}\dot{\mathbf{X}} + \mathbf{K}\mathbf{X} - \mathbf{F}$ is the force residue and $d\mathbf{X}$ is the position increment. If ε is less than 5%, it is assumed that the contour has reached the equilibrium.

3) *Calculation of the Contour Position:* The time interval between two successive frames is assumed to be one. This time interval is then divided into N_t intervals for which SSTMS integration is executed. To avoid attraction of the contour

by noise and undesired objects that can be in its way while moving toward the new position of the object of interest, we chose a nonuniform time division. The first interval Δt_0 is set at 0.9 and the remaining 0.1 is divided into $N_t - 1$ equal intervals Δt_n . During Δt_0 , image forces are set to zero to prevent attraction of the contour by undesired valleys of the potential surface. In other words, we are making the potential surface flat for the first 90% of the time. During this time, elastic forces act on the contour, filtering inaccuracies in velocity estimates. During the remaining 10% of the time, image forces are included, making the contour settle on the edges of the object of interest.

The contour is resampled after each iteration to compensate for possible clustering of the snaxels in the local energy minima along the valley of the potential surface. After resampling, distances between snaxels are equal. Since both positions and velocities of points from one iteration are used in the following one, velocities of new snaxels have to be recalculated as well. The position of a new snaxel is a linear combination of positions of two old snaxels and, hence, its velocity is also a linear combination of velocities of the same two snaxels.

B. Contour Stiffness and Internal Forces

In the case of the discrete contour, internal forces depend on positions of vertices. Internal forces can usually be decomposed into the product of the stiffness matrix and the position matrix $\mathbf{F}_{\text{int}} = -\mathbf{K}(\mathbf{X})\mathbf{X}$. Where such decomposition is not possible (as in the model of Lobregt and Viergever [12]), the equation of motion takes on the form of (7).

Depending on the shape of the object of interest, different internal forces can be designed. The role of such forces is to enforce on the contour a shape feature that is characteristic of the object. For example, in the case of heart valve leaflets, assigning the contour elastic properties that try to preserve length will result in failure of the active contour to detect sudden length changes. However, the leaflet shape feature that seems to be preserved throughout the image sequence is leaflet thickness. This fact suggests that in the case of valve leaflets, additional forces that act to preserve contour thickness should be added.

To define such forces, for each point i on the contour, the distance d_i from the matching point on the opposite side is calculated (Fig. 2). Also, the distance d_i^0 calculated at the optimal position in the first frame is stored and considered to be optimal. The internal force that acts on the vertex i is then defined as a force proportional to the relative change in length d_0

$$\begin{aligned} \mathbf{f}_{\text{int}}^i &= k \frac{|d_i| - |d_i^0|}{|d_i^0|} \frac{d_i}{|d_i|} \mathbf{d}_i = k \left(\frac{1}{d_i^0} - \frac{1}{d_i} \right) \mathbf{d}_i \\ &= k \left(\frac{1}{d_i^0} - \frac{1}{d_i} \right) (\mathbf{X}^j - \mathbf{X}^i) \\ &= k \left(\frac{1}{d_i^0} - \frac{1}{d_i} \right) \langle x_j - x_i, y_j - y_i \rangle \end{aligned} \quad (10)$$

where k is the constant elastic coefficient. The form of (10) indicates that in the i th row of the stiffness matrix, all elements

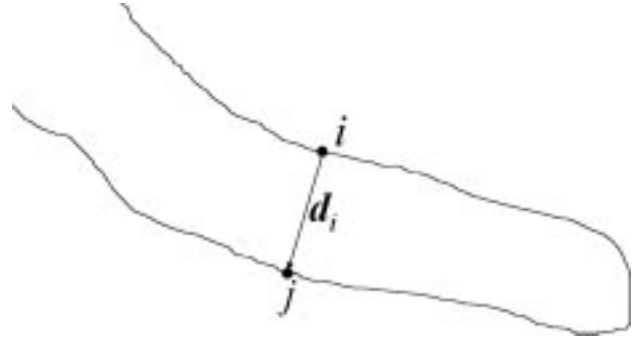


Fig. 2. Internal force acts to preserve the thickness of the leaflet. The force at vertex i will be in the direction of \mathbf{d}_i and its magnitude (positive or negative) will depend on the degree of change in the length of d_i .

are zero, except \mathbf{K}_{ii} and \mathbf{K}_{ij} . If we define

$$k_i = k \left(\frac{1}{d_i^0} - \frac{1}{d_i} \right), \quad \text{then } \mathbf{K}_{ii} = k_i \quad \text{and} \quad \mathbf{K}_{ij} = -k_i.$$

Introduction of such a force prevents the opposite sides of the contour from collapsing. After Δt_0 , the contour is close to the leaflet, but has not reached it completely. At this time, image forces are introduced, and very often both sides of the contour can be attracted by the same side of the leaflet, i.e., be on the slopes of the same valley of the potential surface. Once opposite sides of the contour begin getting closer, the elastic force pushes them apart and brings one side of the contour under the influence of the opposite edge of the leaflet.

For the left ventricle and aortic root image sequences used to test our algorithm, the elastic coefficient k was set to zero. The velocity estimates seemed to be quite accurate and the problem of snaxel clustering was solved by contour resampling in each iteration. The example of tracking the leaflet motion shows that the developed approach is quite flexible. *A priori* knowledge about the shape characteristics of the object of interest can be incorporated into the definition of internal forces, and the rest of the algorithm remains unchanged.

C. Image Forces

To calculate image forces, we first need to define the potential surface. It is computed as

$$\mathbf{H}(i, j) = -(\mathbf{G}_\sigma * \|\nabla \mathbf{I}\|^2)(i, j) \quad (11)$$

where \mathbf{I} is the image intensity, \mathbf{G}_σ is a two-dimensional (2-D) Gaussian mask with standard deviation σ and $*$ is a 2-D convolution. The standard deviation of the Gaussian mask has to be chosen carefully. When the object of interest is very small in at least one dimension (such as in heart valve leaflets, which are very thin), too large value for σ may completely blur the edges of the object. We used the value 1.2 for the mitral valve, and 1.8–1.9 for other image sequences of both a healthy and diseased heart. These numbers were chosen experimentally. Components of the image force at each vertex are calculated as the negative of the partial derivatives of \mathbf{H} in corresponding directions

$$f_x^i = g \left[-\frac{\partial \mathbf{H}[x_i, y_i]}{\partial x} \right]_{\text{norm}}, \quad f_y^i = g \left[-\frac{\partial \mathbf{H}[x_i, y_i]}{\partial y} \right]_{\text{norm}} \quad (12)$$

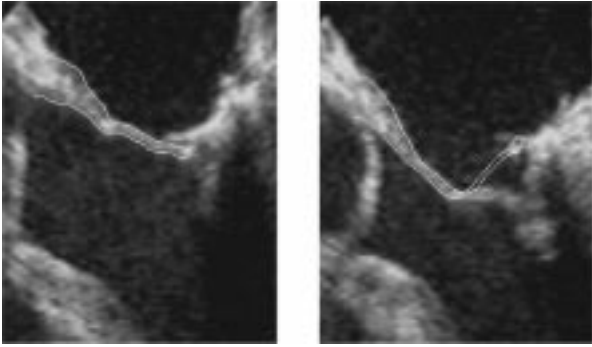


Fig. 3. This example illustrates the problem with optical flow estimates for the leaflet tip in the opening phase.

where index *norm* denotes that the variable is normalized to values from zero to one. The Sobel operator [28] was used to calculate both \mathbf{H} and \mathbf{F} . Parameter g is used to control the amplitude of image forces.

D. Velocity Preprocessing for Heart Valve Leaflets

Ambiguity that is sometimes present in ultrasound images can lead to inaccurate velocity estimates. Introduction of *a priori* knowledge on the structure's motion can help to resolve this ambiguity. Such processing of optical flow estimates was necessary for mitral valve leaflets. It should be noted that this step was not needed for image sequences of the left ventricle and the aortic root.

The problem with tracking the motion of the mitral valve leaflets occurs in the opening phase. In the closed position, the tips of the valve's two leaflets overlap in the image. Once the valve opens, the tips move to opposite sides. However, both these regions—now moving in different directions—belonged to the same region in the previous frame. The optical flow algorithm fails to resolve this ambiguity, and the velocity of the tip of the anterior leaflet is thereby calculated incorrectly (Fig. 3).

Since this overlap happens only at the tip, it is possible to correct the velocities of points at the tip using known values at other points. The motion of the leaflet can be roughly approximated as a rotation around the axis placed at the valve annulus. In this case, velocities increase approximately linearly as one moves from the annulus toward the tip. The preprocessing is performed only in frames in which velocity estimates are very large (i.e., displacements of more than three times the leaflet thickness), since the problem occurs only in these frames. Velocities are processed by calculating the current and average velocity increment as one moves down the leaflet. If a drastic change in the current increment is found, that velocity value is discarded and a new one is calculated from the previous one, increased by the average increment. The contour is resampled at this time, so the distance between neighboring points is constant. In Fig. 4(b) and (c), y components of optical flow velocities for one characteristic frame are shown for contour points before and after this preprocessing. Points are numbered from 0 to 74, as in Fig. 4(a).

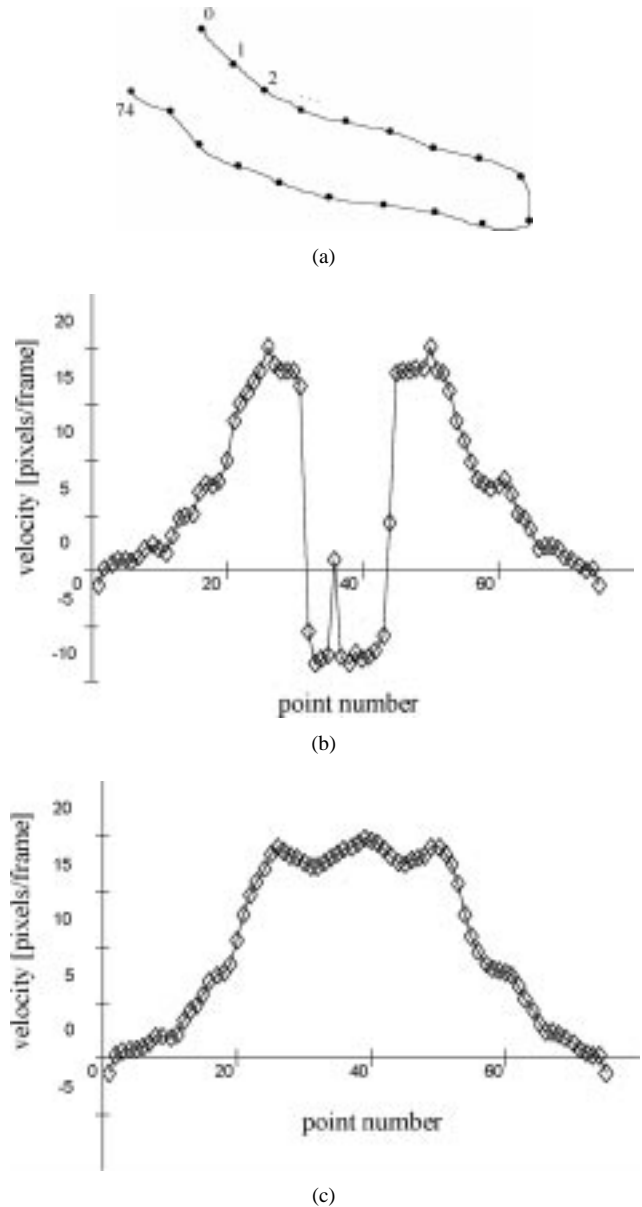


Fig. 4. Velocity preprocessing results: (a) Numbering of points on the contour; valve annulus is at points 0 and 74 and points around the middle belong to the tip, (b) y component of optical flow velocity estimates at each vertex, and (c) results after preprocessing.

To compensate for the possible errors in post processed velocity and to allow for proper modification of the position of the outline formed by the snake, in frames in which the velocity correction has been performed, the value of the first step in the SSTMS Δt_0 was reduced to 0.5 and the total number of steps was increased. This fine-tuning allows image forces to act sooner and possibly hold back parts of the contour whose velocities can become too large.

IV. RESULTS

Performance of the algorithm was evaluated on eight echocardiographic sequences, each covering one complete cardiac cycle. In three sequences the algorithm tracked the endocardial borders of the left ventricle; the first two sequences were recorded using the transthoracic 2.5-MHz transducer

TABLE I
VALUES OF PARAMETERS USED IN THE ALGORITHM

mass (μ)	damping (γ)	elastic coefficient (k)	amplitude of potential surface (g)	θ_1	θ_2	number of time steps (N_t)
1.0	0.5	1.0 (leaflet) 0.0 (other)	700.0	1.2	1.5	21

on the Power Vision SSA-384 ultrasound system (Toshiba, Japan). The pixel size in these images was $0.286 \text{ mm} \times 0.291 \text{ mm}$. The third sequence was recorded on the HP Sonos1500 ultrasound system (Hewlett-Packard, Andover, Massachusetts) with pixel size of $0.76 \text{ mm} \times 0.47 \text{ mm}$. Mitral valve borders were tracked in three sequences and the aortic root in two. These five sequences were recorded using the transesophageal 5-MHz transducer on the HP Sonos 1500; pixel size for these images was $0.76 \text{ mm} \times 0.47 \text{ mm}$. At least three sets of expert physician manual outlines were obtained for each of the sequences. These manual tracings were compared with the boundary outlines generated by the active contour algorithm.

The mean absolute distance (MAD) [29] was the measure used to compare two contours. The MAD between two contours A and B is defined as

$$e(A, B) = \frac{1}{2} \left\{ \frac{1}{n} \sum_{i=1}^n d(\mathbf{a}_i, B) + \frac{1}{m} \sum_{i=1}^m d(\mathbf{b}_i, A) \right\} \quad (13)$$

where $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$, $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m\}$ represent coordinates of points which define the respective contours and $d(\mathbf{a}_i, B)$ is the distance from point to the closest point on the contour B

$$d(\mathbf{a}_i, B) = \min_j \|b_j - \mathbf{a}_i\| \quad (14)$$

For closed contours (endocardial borders of the left ventricle), the correlation between the area enclosed by the snake and the average area enclosed by manual outlines was also calculated. In the sections that follow, the results of the experiments are presented. Table I shows the values of snake parameters used in these.

The values of parameters in Table I were determined experimentally. The final result does not change notably with changes in θ_1 , θ_2 (as long as conditions for the stability of the integrating procedure are satisfied). Increasing N_t slows down the algorithm, without visible differences in the results. Decreasing N_t can result in “missing” fine detail of the object outline. The relationship between four other parameters determines the behavior of the contour. Changing their values while maintaining the ratios is equivalent to multiplying (5) with a constant. That is why we chose mass to be of unit value. Increasing damping γ can result in the contour not reaching far enough in the first 90% of the integration, while decreasing it leads to instability. Too large stiffness k also leads to instability, while low k can result in collapsing of the opposite sides of the contour (mitral valve tracking). The amplitude of the potential surface g needs to be increased if the contour does not closely follow fine detail in the object outline. However, too large g can also lead to instability. Changing

TABLE II
RESULTS OF THE COMPARISON BETWEEN THE ALGORITHM GENERATED CONTOURS AND THE MANUAL OUTLINES FOR SEQUENCES CONTAINING SHORT AXIS VIEW IMAGES OF THE LEFT VENTRICLE

	Sequence 1 (three sets of manual outlines)	Sequence 2 (four sets of manual outlines)	Sequence 3 (five sets of manual outlines)
Distance between snake and outlines (mean/standard deviation) [mm]	1.148/0.199	1.286/0.348	2.071/0.409
Distance between different manual outlines (mean/standard deviation) [mm]	1.185/0.275	1.172/0.33	1.708/0.359
Correlation coefficient between the area enclosed by the snake and the average area enclosed by manual outlines	0.9907	0.9564	0.9928

any of these parameters by 10% or less does not produce any visible changes in the contour behavior.

A. Tracking the Endocardial Borders

The algorithm was tested on three ultrasound sequences containing short axis view images of the left ventricle. Some frames from the first sequence are shown in Fig. 5 with the algorithm-generated contour.

Three sets of the manual outlines were obtained for the first sequence, four for the second and five for the third. Table II shows the standard deviation and the mean distance (average MAD for the whole sequence) between the algorithm generated contours and the manual outlines, the standard deviation and the mean distance between different manual outlines and the correlation coefficient between the average areas outlined by the observers in each frame and the areas enclosed by the algorithm-generated contour. These experiments show that the algorithm tracings compare well to manual outlines for the short axis view images of the left ventricle.

B. Tracking the Mitral Valve Leaflet Boundary

The algorithm performance was evaluated on three sequences of long axis view images of the mitral valve. Four sets of manual outlines were obtained for the first sequence, three sets for the second and five sets for the third. Characteristic frames from the first sequence, with the algorithm generated contours, are shown in Fig. 6 and from the second sequence in Fig. 7. Tracking shown in Fig. 6 required manual intervention in the frame shown in the upper right corner. In that frame, the optical flow estimation algorithm did not recognize the very irregular motion of the tip of the leaflet. The method for velocity correction did not produce satisfying results since the assumption of near rotational motion was violated. In that frame, the tip of the leaflet was manually moved close to the correct position. The algorithm proceeded automatically in the following frames.

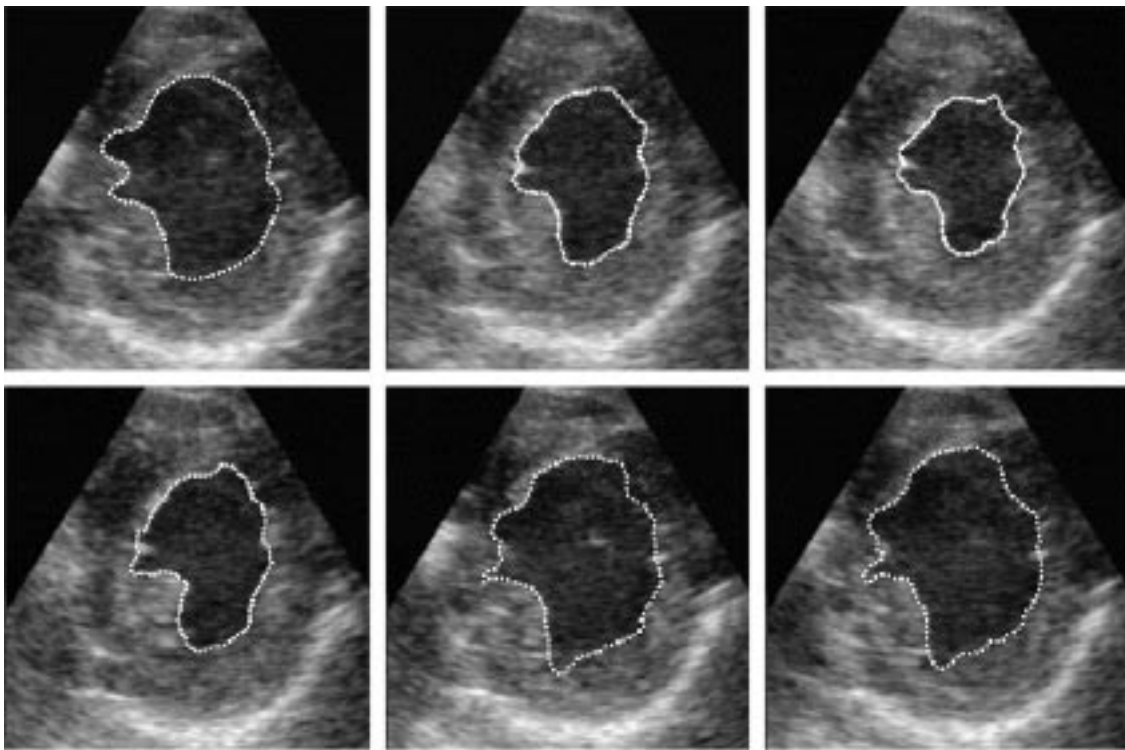


Fig. 5. Characteristic frames showing the tracking of the endocardial border of the left ventricle.

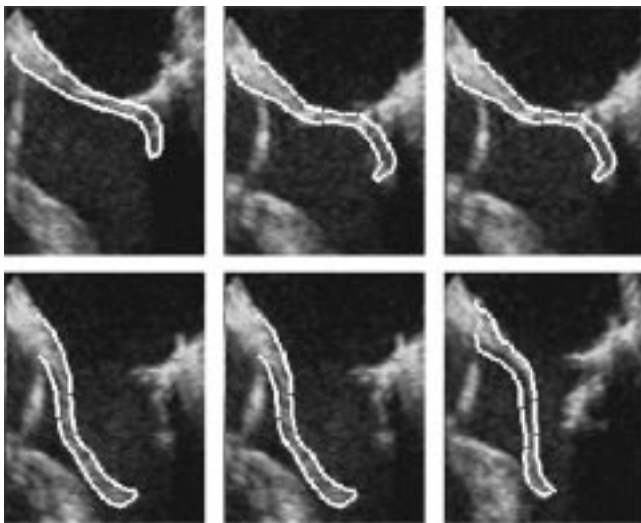


Fig. 6. Characteristic frames showing the tracking of the mitral valve leaflet boundary (sequence 1). Manual correction was necessary in the frame shown in the upper right corner.

Table III shows the results of the experiments with sequences containing images of the mitral valve. As one could expect, the differences between contours are larger than for the images of the left ventricle, but the algorithm performance is still comparable to the results of manual tracings.

C. Tracking of the Aortic Root

Finally, the algorithm was tested on two sequences of images of the aortic root. The motion of the aorta is close to translation and the algorithm performed very well. However, the structures in the image were moving to the left throughout

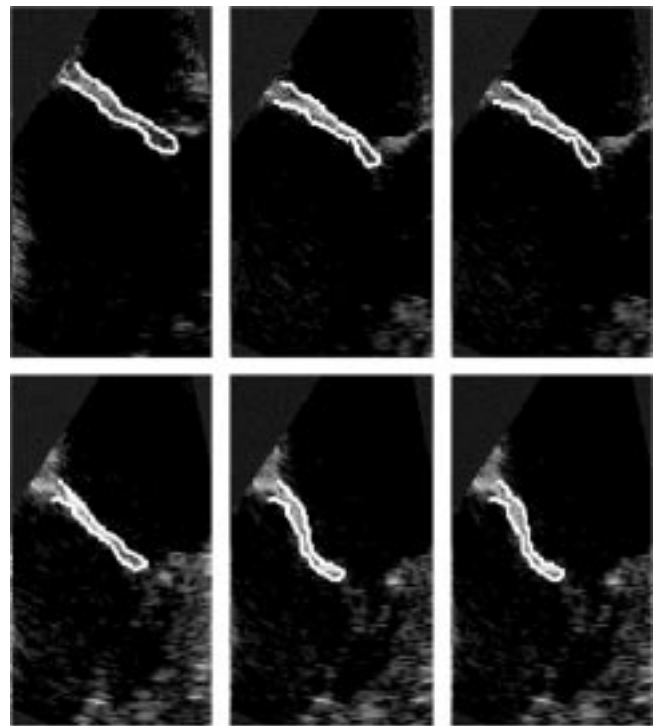


Fig. 7. Characteristic frames showing the tracking of the mitral valve leaflet boundary (sequence 2).

the sequence. While the algorithm generated contour tended to move with the structures that were outlined in the first frame and maintain its original length, the manual outlines were getting longer, incorporating the wall of the aorta that was appearing in the right part of the image. That is the reason the measures of algorithm performance were lower

TABLE III
RESULTS OF THE COMPARISON BETWEEN THE ALGORITHM
GENERATED CONTOURS AND THE MANUAL OUTLINES FOR
SEQUENCES CONTAINING IMAGES OF THE MITRAL VALVE

	Sequence 1 (four sets of manual outlines)	Sequence 2 (three sets of manual outlines)	Sequence 3 (five sets of manual outlines)
Distance between snake and outlines (mean/ standard deviation) [mm]	2.14/0.759	1.774/0.977	1.181/0.15
Distance between different manual outlines (mean/standard deviation) [mm]	2.27/1.19	1.256/0.938	0.885/0.141

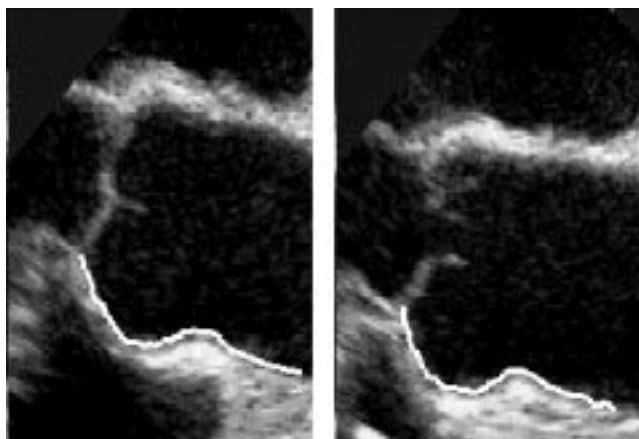


Fig. 8. Two frames illustrating the tracking of the aorta.

TABLE IV
RESULTS OF THE COMPARISON BETWEEN THE ALGORITHM-GENERATED
CONTOURS AND THE MANUAL OUTLINES FOR
SEQUENCES CONTAINING IMAGES OF THE AORTIC ROOT

	Sequence 1 (three sets of manual outlines)	Sequence 2 (five sets of manual outlines)
Distance between snake and outlines (mean/standard deviation) [mm]	1.528/0.443	2.087/0.242
Distance between different manual outlines (mean/ standard deviation) [mm]	1.268/0.401	1.65/0.357

than one would expect. Fig. 8 shows two frames from the first sequence with the algorithm generated contour. Table IV shows the comparisons between manual outlines and the algorithm generated contours for both sequences.

V. DISCUSSION

The analysis described and results presented show that the algorithm performs well on three different types of images. What is important to note is the very different nature of motion present in the three types of image sequences tested. Motion of the aorta is very close to translation and displacements are not very large. The motion of the mitral valve is very irregular. Frame to frame displacements in the opening and closing

phases are several times larger than the leaflet thickness. At those phases, the leaflet rotates, translates and deforms at the same time. Displacements of the left ventricle are not as large as those seen with mitral valve leaflets, but they can be significant. Contraction and relaxation of the ventricle cause large deformations of the endocardial border. In all eight sequences, algorithm generated contours compared well with manual outlines.

An important property of the active contour algorithm that is hard to quantify, but very easy to perceive, is the steadiness of the generated outlines. When playing the sequence, manual outlines appear very jittery which makes quantification of regional wall motion, for example, very difficult. Algorithm generated contours, in contrast, appear very stable and consistent from frame to frame.¹

The user can intervene if the final object outline is not satisfying. In our experiments, only in one frame of one sequence was this intervention performed. The extent of user involvement is illustrated by the fact that in eight ultrasound sequences containing more than two hundred image frames, the user's interaction was needed in just nine frames: each of the eight sets required an initial contour and one set required a correction in one frame. Furthermore, the user did not need to draw a precise outline of the object, but rather to specify a relatively small set of vertices of a polygonal line that was used as a rough initial contour.

The optical flow is calculated using a multiresolution approach. It is natural to question the advantage of using multiresolution optical flow instead of a multiresolution snake. The problem for the multiresolution snake is that the contour has to settle at the coarsest resolution before it's propagated to the next. When the displacement of the structure is larger than the object size, which does happen in several frames in sequences showing heart valves, the contour cannot be attracted to the new position. If the potential surface is blurred enough to attract the snake from such a distance, the object boundaries can be completely eliminated and also a different, undesired strong edge in the image could attract the contour. This problem does not exist if we use multiresolution optical flow. At lower resolutions, the object of interest can be significantly blurred. However, the object is blurred to the same degree in all frames, so matches for pixel neighborhoods can still be found successfully.

Optical flow estimation algorithms are known to be computationally expensive (around 20 min for an average sequence covering one complete cardiac cycle on an SGI Indy). However, these algorithms do not need user's supervision. That is the reason we perform optical flow calculations separately, and store the calculated values into a file. The user only supervises the tracking algorithm that reads prepared optical flow estimates as needed.

VI. CONCLUSION

We have presented a new active contour model for tracking objects in image sequences. The model successfully handles

¹Quicktime movies showing types of sequences presented in this paper with algorithm generated contours and manual outlines can be viewed at: <http://vision.ucsd.edu/~ivana/snakes.htm>.

large frame-to-frame displacements of the object of interest. The presented algorithm requires the user's interaction only in the first frame of the sequence. Optical flow information is used to estimate the initial position of the contour in subsequent frames. Occasionally, for very irregular and large displacements, the optical flow algorithm may fail to compute the displacements with sufficient accuracy. *A priori* knowledge of the type of motion can in that case improve the performance of the algorithm.

Accurate tracking of cardiac boundaries would bring a number of benefits to echocardiography. For example, the contraction and relaxation sequence of the left ventricle can provide important clues to disease states. Patients with coronary artery disease typically display delayed relaxation in the ventricular wall supplied by the stenosed coronary artery, even if the overall wall motion is normal [30]. Manual analysis of the timing of contraction on a frame-by-frame basis is very tedious; the automated contour tracking described here could make such a synchronicity analysis available for routine clinical use. Also, the three-dimensional (3-D) reconstruction of cardiac structures requires accurate segmentation which is impractical by manual means. The ability to extract moving valvular structures from these images is particularly promising for 3-D analysis of valvular mechanics [31].

A limitation of all echocardiographic image processing is the inherently poor image quality of the ultrasound image. Fortunately, a number of recent advances have significantly improved signal-to-noise ratio in echocardiography. For example, the improved dynamic range in contemporary commercial instruments (up to 150 dB) has allowed the display of weak harmonic echoes resulting from nonlinear acoustic propagation; these have considerably fewer near-field and side-lobe artifacts than images resulting from the fundamental frequency. Also, display of the Doppler power within the myocardium considerably improves the delineation of endocardial borders. Advances such as these improve the chances that the algorithm described here will be widely applicable to echo images.

Our algorithm was designed for and tested on ultrasound images. However, there is nothing in this method that is specific to this imaging modality. Considering the relatively poor quality of ultrasound images and the very irregular nature of the motion of structures that this algorithm successfully tracks, it is reasonable to expect even better performance for other types of images.

APPENDIX

CALCULATION OF OPTICAL FLOW ESTIMATES

The optical flow field is the distribution of velocities of pixels in an image sequence. Methods for optical flow estima-

tion often try to find the best match between small image regions in two consecutive frames. It is assumed that the point has moved to the location of the best match and the corresponding displacement is taken as the optical flow vector. Due to the well-known aperture problem, such approaches can recover only the component of flow normal to the intensity edges. The full optical flow estimate can be calculated by propagating information from the regions that do not suffer from the aperture problem, such as corners.

Singh's optical flow estimation algorithm [26] contains two steps. In the first step, a small window W_p of size $(2n+1)(2n+1)$ is formed around a pixel (x, y) in the first frame (we use $n=1$). Also a search area W_s of size $(2N+1)(2N+1)$ is established around location (x, y) in the second frame. The size of W_s depends on the assumed maximal displacement of pixels between two successive frames. We use $N=2$. The correlation between values of pixels in W_p and such window around each pixel in W_s provides a similarity measure. This similarity measure can be used as a probability mass function with its value at each position in W_s representing the likelihood of the corresponding displacement.

To compare two pixel neighborhoods, Singh uses the sum of square differences (SSD)

$$E_c(u, v) = \sum_{i=-n}^n \sum_{j=-n}^n [I(x+i, y+j, t) - I(x+u+i, y+v+j, t)]^2, \\ -N \leq u, \quad v \leq N$$

and for the probability mass function

$$R_c(u, v) = e^{-kE_c(u, v)}$$

where we chose k such that the minimum response is close to zero (0.001). Having defined the probability mass function, one can compute the expected value of the velocity $U_{cc} = (u_{cc}, v_{cc})$ as

$$u_{cc} = \frac{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)u}{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)} \\ v_{cc} = \frac{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)v}{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)}$$

the corresponding covariance matrix, as shown at the bottom of the page, and the associated error (called conservation error) can be expressed as

$$(U - U_{cc})^T S_{cc}^{-1} (U - U_{cc}).$$

In the second step of the algorithm, the velocity estimates are refined by the use of neighborhood information. The velocity estimates for the neighboring pixels can be viewed

$$S_{cc} = \begin{bmatrix} \frac{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)(u-u_{cc})^2}{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)} & \frac{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)(u-u_{cc})(v-v_{cc})}{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)} \\ \frac{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)(u-u_{cc})(v-v_{cc})}{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)} & \frac{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)(v-v_{cc})^2}{\sum_{u=-N}^N \sum_{v=-N}^N R_c(u, v)} \end{bmatrix}$$

$$S_n = \begin{bmatrix} \frac{\sum_{x_i \in W_p} R_n(u_i, v_i)(u_i - \bar{u})^2}{\sum_{x_i \in W_p} R_n(u_i, v_i)} & \frac{\sum_{x_i \in W_p} R_n(u_i, v_i)(u_i - \bar{u})(v_i - \bar{v})}{\sum_{x_i \in W_p} R_n(u_i, v_i)} \\ \frac{\sum_{x_i \in W_p} R_n(u_i, v_i)(u_i - \bar{u})(v_i - \bar{v})}{\sum_{x_i \in W_p} R_n(u_i, v_i)} & \frac{\sum_{x_i \in W_p} R_n(u_i, v_i)(v_i - \bar{v})^2}{\sum_{x_i \in W_p} R_n(u_i, v_i)} \end{bmatrix}$$

as measurements of the true velocity of the central pixel. All those measurements are not equally reliable. It is more likely that velocities of pixels closer to the central pixel are equal to its true velocity than it is for the pixels that are farther away. Bearing that in mind, weights are assigned to velocities calculated for neighboring pixels. Those weights drop with the increasing distance from the central pixel (a 2-D Gaussian mask in a window W_p of size $(2w + 1) \times (2w + 1)$ is used, where $w = 1$). These weights can be interpreted as a probability mass function $R_n(u_i, v_i)$ (i is an index for pixels in W_p) in a uv space. Now, velocity estimate $\bar{U} = (\bar{u}, \bar{v})$ for the central pixel can be calculated as

$$\bar{u} = \frac{\sum_{x_i \in W_p} R_n(u_i, v_i)u_i}{\sum_{x_i \in W_p} R_n(u_i, v_i)} \quad \bar{v} = \frac{\sum_{x_i \in W_p} R_n(u_i, v_i)v_i}{\sum_{x_i \in W_p} R_n(u_i, v_i)}$$

the corresponding covariance matrix, as shown at the top of the page, and the associated error (called neighborhood error) as

$$(U - \bar{U})^T S_n^{-1} (U - \bar{U})$$

The sum of neighborhood and conservation errors represents the squared error of the velocity estimate U

$$\varepsilon^2 = (U - \bar{U})^T S_n^{-1} (U - \bar{U}) + (U - U_{cc})^T S_{cc}^{-1} (U - U_{cc}).$$

The estimate that minimizes this error is found by setting the gradient of the error with respect to U to zero. That gives

$$U = [S_{cc}^{-1} + S_n^{-1}] [S_{cc}^{-1} U_{cc} + S_n^{-1} \bar{U}].$$

Gauss–Seidel relaxation [32] gives an iterative solution for

$$U^{n+1} = [S_{cc}^{-1} + S_n^{-1}] [S_{cc}^{-1} U_{cc} + S_n^{-1} \bar{U}^n]$$

where $U^0 = U_{cc}$ and n is the iteration number. Iterations are stopped when the difference between two successive solutions is smaller than some specified value (we used 0.05).

Coarse-to-Fine Strategy: When it is known in advance that velocities are large, one could increase the size of the search window W_s . Doing that would significantly increase the computational load of the algorithm. Instead, Singh suggested the multiresolution approach. The images are decomposed into the Gaussian pyramid [33]. Velocities are first estimated at the low resolution level. Then, at the next finer resolution, these estimates are used as a first guess in the estimation process, and the estimates are refined. This means that instead of searching in the window around (x, y) in the second image, we search around $(x + \tilde{u}, y + \tilde{v})$, where \tilde{u} and \tilde{v} are velocity estimates propagated from the coarser level. Using this approach, with 5×5 search window, velocities of up to 30 pixels per frame can be detected using four resolution levels.

ACKNOWLEDGMENT

The authors would like to thank S. Chandra and R. Shekhar for writing the user interface. They would also like to thank D. Leung, M. Pu, L. Rodriguez, G. Scalia, and M. Stugaart for hand tracing the boundaries of the structures of interest in the test sequences.

REFERENCES

- [1] I. Mikić, S. Krucinski, and J. D. Thomas, "Segmentation and tracking of mitral valve leaflets in echocardiographic sequences: Active contours guided by optical flow estimates," *SPIE Med. Imag.*, vol. 2710, pp. 311–320, 1996.
- [2] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Med. Image Anal.*, vol. 1, no. 2, pp. 91–108, 1996.
- [3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," in *Proc. 1st Int. Conf. Computer Vision*, 1987, pp. 259–268.
- [4] L. D. Cohen, "On active contours models and balloons," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211–218, 1991.
- [5] L. D. Cohen and I. Cohen, "A finite element method applied to new active contour models and 3-D reconstruction from cross sections," in *Proc. 3rd Int. Conf. Computer Vision*, 1990, pp. 587–591.
- [6] L. D. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, Nov. 1993.
- [7] A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 855–867, Sept. 1990.
- [8] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 294–302, Mar. 1995.
- [9] R. P. Grzeszczuk and D. N. Levin, "Brownian strings: Segmenting images with stochastically deformable contours," in *Proc. 3rd Conf. Visualization in Biomedical Computing, SPIE Proc.*, 1994, vol. 2359, pp. 72–89.
- [10] C. S. Poon, M. Braun, R. Fahrig, A. Ginige, and A. Dorell, "Segmentation of medical images using an active contour model incorporating region-based image features," in *Proc. 3rd Conf. Visualization in Biomedical Computing, SPIE Proc.*, 1994, vol. 2359, pp. 90–97.
- [11] F. Leymarie and M. D. Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 617–634, June 1993.
- [12] S. Lobregt and M. A. Viergever, "A discrete active contour model," *IEEE Trans. Med. Imag.*, vol. 14, pp. 12–24, Feb. 1995.
- [13] F. Leitner and P. Cinquin, "From splines and snakes to snake splines," *Geometric Reasoning: From Perception to Action, Lecture Notes in Computer Science*, vol. 708, C. Laugier, Ed. New York: Springer-Verlag, 1990, pp. 477–484.
- [14] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 158–175, Feb. 1995.
- [15] T. McInerney and D. Terzopoulos, "Topologically adaptable snakes," in *Proc. 5th Int. Conf. Computer Vision*, 1995, pp. 840–845.
- [16] P. Lipson, A. Yuille, D. O'Keefe, J. Cavanaugh, J. Taaffe, and D. Rosenthal, "Deformable templates for feature extraction from medical images," in *Proc. 1st Eur. Conf. Computer Vision (ECCV'90)*, 1990, pp. 477–484.
- [17] A. L. Yuille and P. W. Hallinan, "Deformable templates," *Active Vision*, A. Blake and A. Yuille, Eds. Cambridge, MA: MIT Press, 1992, pp. 21–38, 1992.
- [18] P. Clarysse, F. Poupon, B. Barbier, and I. E. Magnin, "3-D boundary extraction of the left ventricle by a deformable model with *a priori* information," in *Proc. IEEE Int. Conf. Image Processing*, 1995, pp. 492–495.

- [19] E. Bardinet, L. D. Cohen, and N. Ayache, "Tracking and motion analysis of the left ventricle with deformable superquadratics," *Med. Image Anal.*, vol. 1, no. 2, pp. 129–149, 1996.
- [20] C. W. Chen, T. S. Huang, and M. Arrott, "Modeling, analysis, and visualization of the left ventricle shape and motion by hierarchical decomposition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 342–256, Apr. 1994.
- [21] D. Terzopoulos and R. Szeliski, "Tracking with Kalman snakes," *Active Vision*, A. Blake and A. Yuille, Eds. Cambridge, MA: MIT Press, 1992, pp. 3–20.
- [22] D. Metaxas and D. Terzopoulos, "Shape and nonrigid motion estimation through physics-based synthesis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 580–591, June 1993.
- [23] A. Blake, M. Isard, and D. Reynard, "Learning to track the visual motion of contours," *Artificial Intell.*, no. 78, pp. 179–212, 1995.
- [24] A. Pentland and B. Horowitz, "Recovery of nonrigid motion and structure," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 730–742, July 1991.
- [25] N. Ayache, I. Cohen, and I. Herlin, "Medical image tracking," *Active Vision*, A. Blake and A. Yuille, Eds. Cambridge, MA: MIT Press, 1992, pp. 285–302.
- [26] A. Singh, *Optic Flow Computation: A Unified Perspective*. Piscataway, NJ: IEEE Comput. Soc. Press, 1992.
- [27] O. C. Zienkiewicz, W. L. Wood, N. W. Hine, and R. L. Taylor, "A unified set of single step algorithms," *Int. J. Numerical Meth. Eng.*, vol. 20, pp. 1529–1552, 1984.
- [28] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989, p. 349.
- [29] 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. Med. Imag.*, vol. 15, pp. 290–297, June 1996.
- [30] H. Kondo, T. Masuyama, K. Ishihara, T. Mano, K. Yamamoto, J. Naito, R. Nagano, S. Kishimoto, J. Tanouchi, M. Hori *et al.*, "Digital subtraction high-frame-rate echocardiography in detecting delayed onset of regional left ventricular relaxation in ischemic heart disease [see comments]," *Circ.*, vol. 91, pp. 304–312, 1995.
- [31] C. S. Breburda, I. Vesely, A. R. Brewer, N. B. Ratliff, D. M. Cosgrove, III, J. D. Thomas, and B. P. Griffin, "Quantification of mitral valve prolapse by three-dimensional echocardiography: Correlation with biomechanical characteristics of excised valve tissue (abstract)," *J. Amer. Coll. Cardiol.*, vol. 29, p. 371A, 1997.
- [32] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*. Cambridge: Cambridge Univ. Press, pp. 60–72, 1988.
- [33] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, Apr. 1983.