International Journal of Image and Graphics, Vol. 1, No. 4 (2001) 647–658

ACTIVE FOURIER CONTOUR APPLIED TO REAL TIME 3D ULTRASOUND OF THE HEART

GEORGE D. STETTEN*

Department of Bioengineering, University of Pittsburgh,
3700 O'Hara Street Pittsburgh, PA 15261, USA
george@stetten.com

REBEKAH DREZEK

Department of Electrical and Computer Engineering,
University of Texas at Austin, USA

We describe an active contour based on the elliptical Fourier series, and its application to matrix-array ultrasound. Matrix-array, or Real Time 3D (RT3D), ultrasound is a relatively new medical imaging modality that scans a 3D-volume electronically without physically moving the transducer, allowing for real-time continuous 3D imaging of the heart. With the goal of automatically tracking the heart wall, an active contour has been developed using the elliptical Fourier series to find perpendicular lines intersecting an initial contour. The neighborhood defined by these perpendiculars is mapped into a rectangular space, called a *swath*, whose vertical axis represents the inside-vs-outside dimension of the contour (perpendicular to the contour), and whose horizontal axis represents parametric distance along the contour (tangent to the contour). A dynamic programming technique is then used to find the optimum error function traversing the rectangle horizontally, and this error function is mapped back into image space to yield a new contour. The method does not iterate, but rather simultaneously searches for *the* optimum contour within a limited domain. Results are presented applying the technique to RT3D ultrasound images of *in vivo* hearts.

Keywords: Adaptive Contour; Snake; Elliptical Fourier; Flow Integration Transform; RT3D Ultrasound.

1. Introduction

Real-time 3D ultrasound based on a matrix-array transducer is a relatively new imaging modality capable of capturing 22 volumetric images of the heart per second.[1–3] This permits a potentially useful continuous measure of ventricular volume, the so-called *volumetricardiogram* (VCG) assuming a method can be developed to automatically locate and delineate the endocardial boundary of the cardiac ventricle. We have previously developed the Flow Integration Transform (FIT)[4,5]

[*] 749 Benedum Hall, Department of Bioengineering, University of Pittsburgh, 3700 O'Hara Street, Pittsburgh, PA 15261, USA.

and successfully performed fully automated volume determination on balloons, assuming them to be a stack of circular cross-sections.[6] To make the FIT adaptive to more complicated shapes, the approach described in this paper has been explored with encouraging results. It assumes that an initial contour has been established in the form of a Freeman chain code,[7] by means of the FIT or other shape detection algorithm, or (as in the demonstration cases presented here) by manually placing an initial circular contour. The algorithm permits the contour to adapt to locate gradients in the image, causing appropriate changes in length, shape, and location of the contour to match the target. We make a distinction here between what usually are called snakes, deformable contours, active contours, or active shapes,[8–12] and what has been referred to as a *live wire*.[13,14] Whereas the former usually employ an iterative search to maximize (or minimize) some merit (or energy) function, the latter employ dynamic programming to simultaneously search through a domain for *the* optimal solution in a single step. In the interest of speed, our method works in the latter fashion, although, in its method of initialization, our method resembles more the active contour approach, permitting greater incorporation of prior information. We thus combine beneficial aspects of both approaches.

2. Methods

2.1. *Elliptical Fourier series*

The search for an optimal contour may proceed by searching in the region of the image adjoining an initial contour. The region may be mapped by establishing local perpendiculars to the initial contour. Finding perpendiculars to a chain code requires some method capable of producing finer gradation in angle than is evident from the individual steps in a chain code. Such a method is the elliptical Fourier series. The application of the Fourier transform to image data is a classical approach in image processing, but has also been applied to describing parametric shapes, such as contours in 2D,[15–17] or surfaces in 3D.[18,19] We constrain ourselves here to their use in describing contours in 2D.

Any closed path can be separated into periodic functions in the x and y dimensions, as shown in Fig. 1.

The functions $x(t)$ and $y(t)$ are periodic and can each be represented by its Fourier series.

$$\begin{aligned} x(t) &= \frac{a_0}{2} + \sum_{n=1}^{N} (a_n \cos(nt) + b_n \sin(nt)) \,, \\ y(t) &= \frac{c_0}{2} + \sum_{n=1}^{N} (c_n \cos(nt) + d_n \sin(nt)) \,. \end{aligned} \tag{1}$$

The accuracy of the approximation depends on N and the number of samples in the actual discrete path. Note that at each harmonic there are four parameters

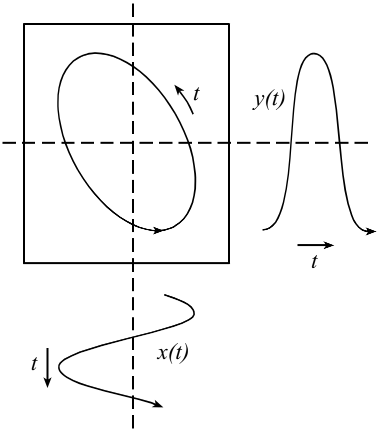


Fig. 1. The elliptical Fourier series consists of a separate Fourier series for each of the x and y components of the parametrically described periodic function of the closed contour.

instead of the usual two:

$$
\begin{aligned}
a_n &= \int_0^{2\pi} x(t)\cos(nt)\,dt\,, \quad & b_n &= \int_0^{2\pi} x(t)\sin(nt)\,dt\,,\\
c_n &= \int_0^{2\pi} y(t)\cos(nt)\,dt\,, \quad & d_n &= \int_0^{2\pi} y(t)\sin(nt)\,dt\,.
\end{aligned}
\tag{2}
$$

By performing these discrete Fourier transforms on the x and y components of the chain code, we can produce a spectrum with as many harmonics as desired, limited by the Nyquist criterion to half the number of steps in the chain code. We can now use the Fourier description of the contour to find perpendiculars to the contour at any point. Let $\dot{x}(t)$ and $\dot{y}(t)$ be the x and y components of the derivative of the contour with respect to t. Taking the derivative of the Fourier series we get:

$$
\begin{aligned}
\dot{x}(t) &= \sum_{n=1}^{N} n(-a_n\sin(nt) + b_n\cos(nt))\,,\\
\dot{y}(t) &= \sum_{n=1}^{N} n(-c_n\sin(nt) + d_n\cos(nt))\,.
\end{aligned}
\tag{3}
$$

Let us imagine that somewhere along a given perpendicular there is a better boundary, one to which we would like our original contour to adapt. Let us further define the error e to be the distance from the original contour to that boundary along the perpendicular. Using the standard slope and intercept formulation, we can now solve for the location of the boundary point at $(x(e,t), y(e,t))$ as

$$
\begin{bmatrix} x(e,t) \\ y(e,t) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \frac{e}{\sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix}. \tag{4}
$$

We have rotated and normalized the vector $(\dot{x}(t), \dot{y}(t))$ to get the slope of the perpendicular to the contour. Note that e is signed, with $(e > 0)$ inside and $(e < 0)$ outside the contour. Our goal is to find an error function $E(t)$ that links as many of the best boundary points together as possible, within some neighborhood of the initial contour which we call the *swath.*

2.2. *Error function in the swath*

Using the perpendiculars, we can map the region surrounding the contour, or *swath*, onto a rectangular space as shown in Fig. 2. The original contour is mapped onto the horizontal line at $e = 0$, which we may write as the error function $e(t) = 0$. One can imagine strolling along this straightened version of the original contour from 0 to 2π, with the inside of the image on the right and the outside on the left. Although originating from a 2D contour, a dimension of the contour's shape has been suppressed by straightening it into the horizontal dimension of a rectangular space.

After mapping the swath into this rectangular space, we then look for boundaries running along the swath on either side of the original contour. Our goal is to connect the optimal continuous set of such boundaries. At each location within the swath, we find the boundary component parallel to the contour by computing the dot product of the image gradient and the perpendicular, to come up with a *boundariness* measure, B. For each location within the swath

$$B(e, t) = \left| \frac{\partial I(e, t)}{\partial e} \right| . \tag{5}$$

As is commonly done in image processing, the derivative operator is scaled to prevent high-frequency noise from dominating the desired boundaries.

We calculate this boundariness everywhere within the swath rectangle, producing a 2D image as shown in Figs. 4(b) and (e). High intensity areas in the swath rectangle represent boundaries in the neighborhood of, and parallel to, the original

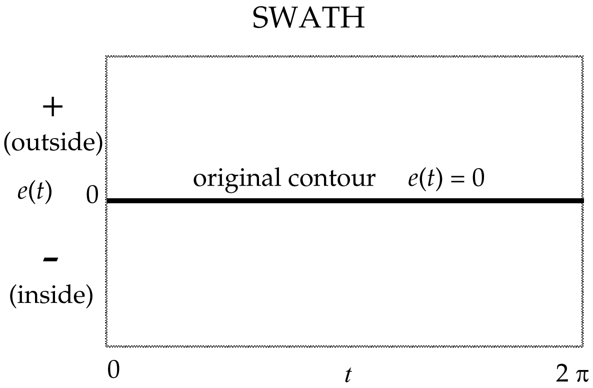


Fig. 2. The *swath* around the initial contour is mapped into a rectangular space defined by the error function $e(t)$ along local perpendiculars to the initial contour.

contour, and it is *these* boundaries, after all, to which we hope the new improved boundary will adapt. The optimum adaptation for the contour may be found by maximizing the *merit* M of the error function, by choosing the error function which passes through as many strong parallel boundaries as possible. (This optimum error function is, in fact, shown superimposed on the swath in Figs. 4(b) and (e)). The merit of a particular error function may be expressed in the continuous domain as

$$M_E = \int_0^{2\pi} B(E(t), t)\, dt\,, \tag{6}$$

where M_E is the accumulated merit along the particular error function $E(t)$. Having found the best error function

$$E(t) = \arg\max M_E\,, \quad \forall\, E \tag{7}$$

a new contour can be formed by re-mapping the original contour plus the error function from the rectangular swath back into image space. We ensure the smoothness of this new contour by constraining the maximum slew rate,

$$\left| \frac{\partial E(t)}{\partial t} \right| \leq 1\,, \tag{8}$$

and we ensure the contour's periodicity by requiring that the error function itself be periodic,

$$E(2\pi) = E(0)\,. \tag{9}$$

2.3. *Finding the optimum error function*

To accomplish this in the discrete domain, let us store the values for $B(e, t)$ boundariness in an $m \times n$ rectangular *swath matrix* [again, see Figs. 4(b) and (e)], where m is the number of samples along the perpendicular, and n is the number samples for the parameter t over the range 0 to 2π. To find the best error function we might compare the merit of every possible horizontal path across the swath matrix. However, we will accept the constraint that the error function must cross each column in the matrix once and only once, i.e. that $E(t)$ is a proper function. Furthermore, $E(t)$ cannot move up or down more than one step for every horizontal step along the path, constraining its smoothness. Still, an enormous number of total possible paths exist, $O(m3^n)$, with the number of periodic paths being only somewhat fewer, $O(3^n)$. Luckily, this number may be greatly reduced by using the following dynamic programming technique.

Consider the merit function for paths that cross a *portion* of the swath matrix, from a *first* column at $t = 0$ to some arbitrary *last* column in the middle somewhere. Let us construct the set of paths that contains every optimal path from each starting location to each ending location, between $e = f$ in the first column to $e = l$ in the last column of that portion of the swath matrix. Since f and l each have m possible values, there are m^2 paths in the set, each one being the optimal path from a particular starting and ending location. Our merit function $M_t(f, l)$ for a particular path takes two arguments, f and l, and has a single subscript denoting the t, which is currently considered the last column. Starting with $t = 0$, we initialize the accumulated merit for each path as

$$M_0(f, l) = \left| \begin{array}{ll} B(f, 0)\,, & f = l\,, \\ -\infty\,, & f \neq l\,, \end{array} \right. \tag{10}$$

using a very large negative value ($-\infty$) to invalidate paths whose beginnings and endings are not identical. We then proceed to the right, at each step comparing the merits of paths ending in the previous column in rows l, $l-1$, and $l+1$, choosing to link to the one with the highest merit. We thus exclude inferior paths along the way, and need store only m^2 paths and their associated merits. The iteration proceeds as follows:

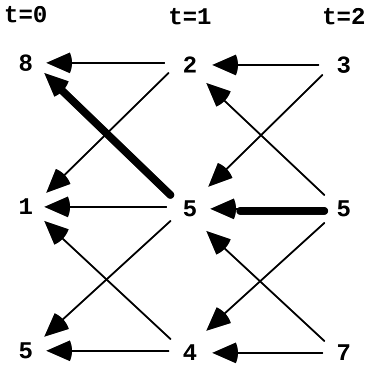
$$\text{for } t = 0 \quad \text{to} \quad t = m - 1$$

$$M_{t+1}(f, l) = B(l, t+1) + \max \left| \begin{array}{l} M_t(f, l-1) \\ M_t(f, l) \\ M_t(f, l+1) \end{array} \right. . \tag{11}$$

Care must be taken with $l+1$ and $l-1$ not to exceed the limits of the matrix.

An example of the process is shown in Fig. 3, using a 3 × 3 matrix of boundariness values. The algorithm proceeds column by column, from $t = 0$ to $t = 2$. At each step, nine possible paths (including all permutations of values for f and l) are considered. The accumulated merit of each path at each step is shown in the table. At each step, paths link to the best of the adjacent possible previous paths, choosing between the row above, the row below, or the same row. Only two possible prior paths are available in the top or bottom row, to stay inside the matrix. The boundariness of each step is then added to the accumulated merit. Choosing only adjacent rows limits the slew rate to $|\partial E(t)/\partial t| \leq 1$, as noted above, and is essential to the efficiency of the algorithm. The slew rate can, in effect, be adjusted by varying n, the number of perpendiculars (columns in the swath matrix). Without a sufficiently high n, the error function will not be able to effectively link boundaries along the swath with widely different values of e.

At the last step, a maximum value is chosen from the subset of all periodic paths (marked with "P" in Fig. 3), i.e. those paths whose final l is within one row of its f. In this example, the maximum periodic path ($f = 1, l = 2$) has a value of 18. The path ($f = 1, l = 3$) has a greater value, 20, but is excluded because it is not periodic. This constraint guarantees periodicity, and is the discrete equivalent of Eq. (9), above. Since all permutations of values for f and l, both periodic and nonperiodic, are considered at each step along the way, a guaranteed maximum periodic path can be chosen from the subset of paths that are periodic at the final step.



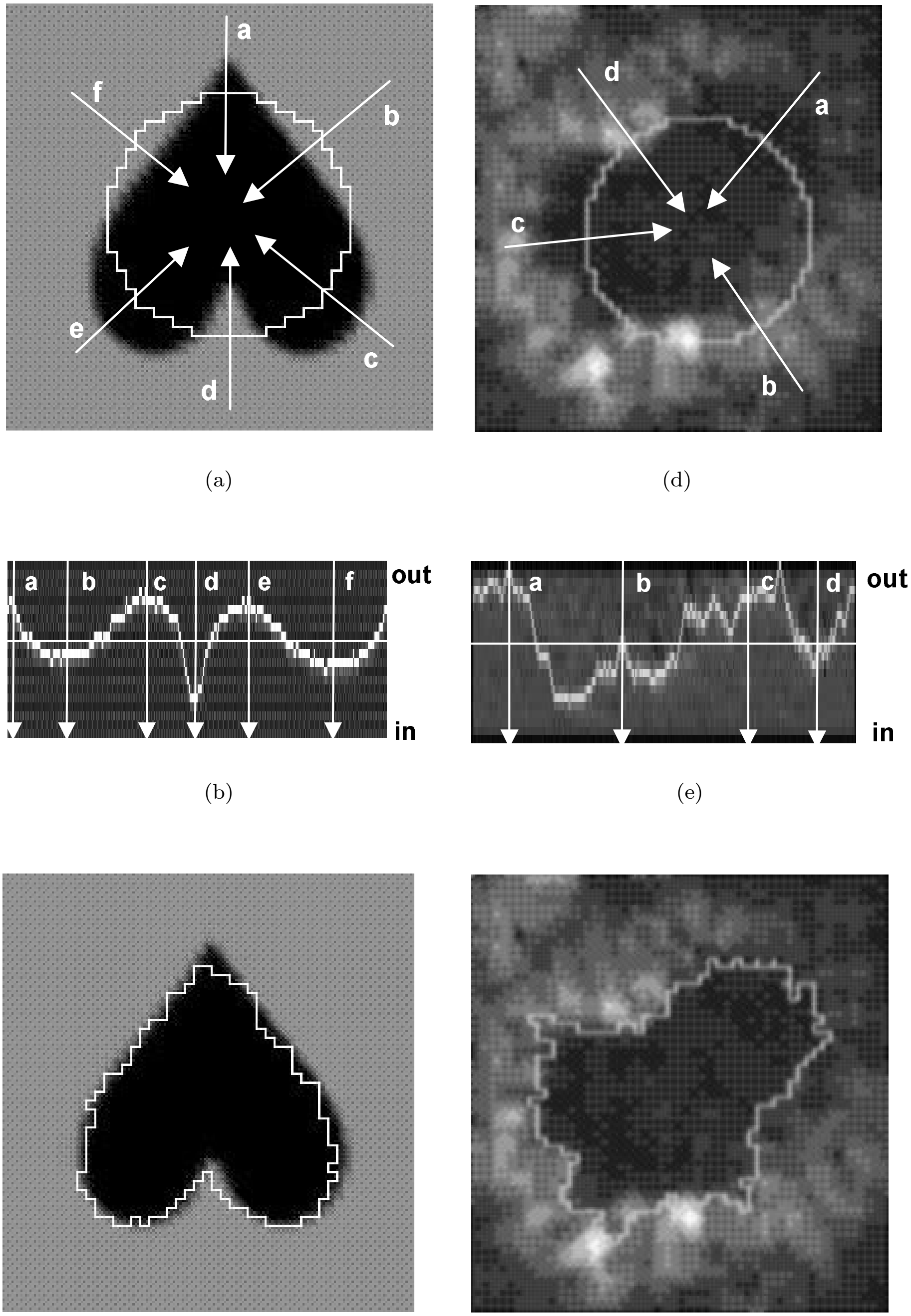
f	l	t=0	t=1	t=2	
1	1	8	8+2=10	13+3=16	P
1	2	-inf	8+5=13	13+5=18	P
1	3	-inf	-inf+4=-inf	13+7=20	
2	1	-inf	1+2=3	6+3=9	P
2	2	1	1+5=6	6+5=11	P
2	3	-inf	1+4=5	6+7=13	P
3	1	-inf	-inf+2=-inf	10+3=13	
3	2	-inf	5+5=10	10+5=15	P
3	3	5	5+4=9	10+7=17	P

Fig. 3. Dynamic programming technique for finding the optimum path (bold on left, gray on right) through the first three columns of a swath. For $t = 0$, all paths for which $f \neq l$ are given a value of $-\infty$ ("$-$ inf").

To reconstruct the final adapted contour, the corresponding points along the perpendiculars are mapped from the rectangular swath back into image space. This series of points in image space are connected by line segments, and converted to an 8-neighbor chain code using Bresenham's method.[20]

3. Results

Figure 4 shows two examples of the 1D Swath. Figure 4(a) shows a graphical heart and a manually placed initial circular contour. The resulting rectangular swath matrix is shown in Fig. 4(b), with the optimal error path shown in white. Representative perpendiculars are shown in Figs. 4(a) and (b), identifying the target boundary as it passes between inside and outside the initial contour. Figure 4(c) shows the final adapted contour after re-mapping the error function back into image space. A similar sequence is shown in Figs. 4(d), (e) and (f) using RT3D ultrasound to produce a constant range slice through an *in vivo* canine heart with contrast in the myocardium. The initial manually placed circular contour produces a swath whose error function is clearly detected, yielding an adapted contour that matches the ventricle.



(a)

(d)

(b)

(e)

(c)

(f)

Fig. 4. Two examples of a single iteration of the swath on a graphical test object (a)–(c) and an ultrasound image of the cardiac left ventricle (d)–(f). See text for details.

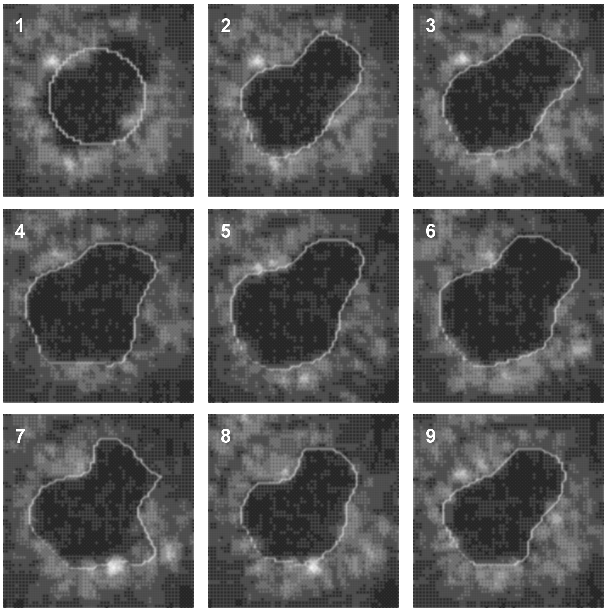


Fig. 5. Contour initially placed on a constant range slice through the RT3D data (1) is seen to adapt to that frame (2) and then to follow the shape of the ventricle through sequential time frames (2)–(9).

Figure 5 shows a sequence of constant range slices through an *in vivo* canine heart with contrast in the myocardium. Frame 1 shows the location of an initial circular contour placed manually. Frame 2 shows the result of a single iteration of the swath algorithm on the initial time frame. Frames 2–9 show sequential time frames with a single iteration of the algorithm in each frame resulting in a fairly accurate tracking of the endocardial border. Notice that the single significant case of straying in frame 7 corrects itself in frame 8.

Figure 6 shows adaptation of a contour originally placed on a single slice through a 3D scan of the cardiac left ventricle using RT3D ultrasound. Whereas in Fig. 5 the contour adapted to a temporal sequence, in Fig. 6 it adapts to a *spatial* sequence across the third dimension. The adapted contour in one slice is used as the starting contour in the adjoining slice. The result is a stack of contours depicting the entire 3D shape of the left ventricle. This stack of paths was then used as the starting points for the 3D data from the next time frame, similar to the process shown in Fig. 5, but with all slices progressing simultaneously (not shown).

4. Conclusions

We have demonstrated an active contour based on an elliptical Fourier representation, which uses an efficient dynamic programming scheme to find the optimal adaptation to boundaries in the image within a swath surrounding the initial contour.

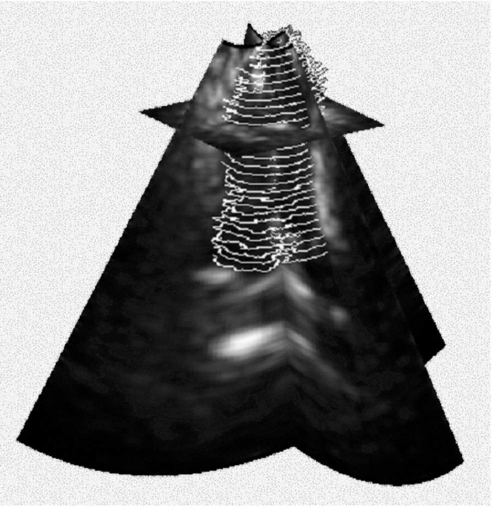


Fig. 6. Contour initially placed on a single slice through an RT3D ultrasound scan of the heart adapts through neighboring slices to match the 3D shape of the left ventricle.

It has successfully tracked the anatomical boundary of the endocardium in noisy RT3D ultrasound data, a task with considerable clinical significance. We consider these results very promising, and are actively exploring methods for automatically placing the initial contour so that the entire procedure could be automatic.

Acknowledgments

Supported through a Whitaker Foundation Biomedical Engineering grant, NSF grant CDR8622201 NIH grants 1K08HL03220 and HL46242, and equipment support from Silicon Graphics, Inc. Assistance in Fig. 6 from Chikai Ohazama. RT3D ultrasound data generously provided by Takahira Ota, Craig Fleishman, Eric Lily, Cleveland Lewis, Thomas Ryan, Donald Glower, Joseph Kisslo and Olaf von Ramm.

References

1. S. W. Smith, H. G. Henry, and O. T. Ramm, "High-speed ultrasound volumetric imaging system — Part I: Transducer design and beam steering," *IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control* **38**, 100–108 (1991).
2. O. T. Ramm, S. W. Smith, and H. G. Jr. Pavy, "High-speed ultrasound volumetric imaging system — Part II: Parallel processing and image display," *IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control* **38**, 109–115 (1991).
3. G. Stetten, T. Ota, C. Ohazama, C. Fleishman, J. Castelucci, J. Oxaal, T. Ryan, J. Kisslo, and O. T. Ramm, "Real-time 3D ultrasound: A new look at the heart," *J. Cardiovascular Diagnosis and Procedures* **15**, 73–84 (1998).
4. G. Stetten and R. Morris, "Shape detection with the flow integration transform," *Information Sci.* **85**, 203–221 (1995).
5. G. Stetten, "Quadrature shape detection using the flow integration transform," US Patent 5,550,933, August 27, 1996.

6. G. Stetten, M. Caines, C. Ohazama, and O. T. Ramm, "The volumetricardiogram (VCG): Volume determination of cardiac chambers using 3D matrix-array ultrasound," in *Proceedings of the SPIE Symposium on Medical Imaging* **2432**, 185–196 (1995).
7. H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electronic Computers* **EC-10**, 260–268 (1961).
8. V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *IEEE 5th International Conference on Computer Vision*, 694–699 (1995).
9. H. Tek and B. B. Kimia, "Image segmentation by reaction-diffusion bubbles," *IEEE 5th International Conference on Computer Vision* (1995).
10. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vision* **1**, 321–331 (1987).
11. T. McInerney and D. Terzopoulos, "Topologically Adaptable Snakes," *IEEE*, 840–845 (1995).
12. A. Blake and A. Yuille, *Active Vision* (MIT Press, Cambridge, MA, 1992).
13. E. Mortensen, B. Morse, W. Barrett, and J. Udupa, "Adaptive boundary detection using 'Live-Wire'two-dimensional dynamic programming," in *Comput. Cardiology*, IEEE (Computer Society Press, Durham, North Carolina, 1992) 635–638.
14. W. A. Barrett and E. N. Mortensen, "Fast, accurate and reproducible live-wire boundary extraction," in *4th International Conference in Biomedical Computing* (Springer, Hamburg, Germany, 1996) 183–192.
15. G. H. Granlund, "Fourier preprocessing for hand print character recognition," *IEEE Trans. Comput.* **C-22**, 195–201 (1972).
16. F. P. Kuhl and C. R. Giardina, "Elliptic fourier features of a closed contour," *Computer Graphic and Image Processing* **18**, 236–258 (1982).
17. C. T. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. Comput.* **C-21**, 269–281 (1972).
18. A. Matheny D. B. Goldgof, "The use of three- and four-dimensional surface harmonics for rigid and nonrigid shape recovery and supression," *IEEE Trans. Paterrn Analysis and Machine Intelligence* **17**, 967–981 (1995).
19. L. H. Staib and J. S. Duncan, "Model-based deformable surface finding for medical images," *IEEE Trans. Medical Imaging* **15**, 1–12 (1996).
20. J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal* **4**, 25–30 (1965).

George Stetten received his B.S. in Engineering and Applied Physics from Harvard University in 1976. After working in the future MIT Media Lab and at the Woods Hole Oceanographic Institute, he received an M.S. degree in Neurobiology/Computing from New York University in 1986, and an M.D. degree from the SUNY Syracuse in 1991. He earned his doctoral degree in 2000 in Biomedical Engineering at the University of North Carolina, Chapel Hill while maintaining faculty status at Duke University. Dr. Stetten is currently an Assistant Professor in Bioengineering at the University of Pittsburgh and a Research Scientist at the Robotics Institute at Carnegie Mellon University. His research involves a new method of identifying, measuring, and visualizing anatomical structures in 3D image data, based on local properties of shape.

Rebekah Drezek received her B.S.E. degree (1996) in electrical engineering from Duke University, her M.S. degree (1998), and her Ph.D. degree (2001) in electrical engineering from the University of Texas at Austin. She is currently an Odyssey Fellow at M.D. Anderson Cancer Center and will join the faculty at Rice University as an Assistant Professor in the bioengineering department in July 2002. Her current research focuses on development of optical imaging techniques for minimally invasive medical diagnosis.