# Ventriculogram segmentation using boosted decision trees.

John Alan McDonald and Florence H. Sheehan
School of Medicine, University of Washington

## ABSTRACT

Left ventricular status, reflected in ejection fraction or end systolic volume, is a powerful prognostic indicator in heart disease. Quantitative analysis of these and other parameters from ventriculograms is infrequently performed due to the expense of manual segmentation. We present a method for semi-automatic segmentation of ventriculograms based on a two-stage boosted decision-tree pixel classifier. The classifier determines which pixels are inside the ventricle at key end-diastole and end-systole frames. The classifier is *semi*-automatic, requiring a user to select 3 points in each frame: the endpoints of the aortic valve and the apex. The classifier uses about 90 feature images, computed from the raw ventriculogram image frames, including simple per pixel gray-level statistics (e.g. median brightness) and image geometry (e.g. coordinates relative to user supplied 3 points). Border pixels are determined from segmented images using dilation and erosion. A curve is then fit to the border pixels, minimizing a penalty function that trades off fidelity to the border pixels with smoothness. Volumes and ejection fraction are estimated from border curves using standard area-length formulas. On independent test data, the differences between automatic and manual volumes (and ejection fractions) are similar in size to the differences between two human observers.

**Keywords:** Image segmentation, pixel classification, computer vision, CART, Adaboost.M1, machine learning, data mining, angiography, heart disease.

## 1. INTRODUCTION

Ventriculograms are cine X-ray images, in which an X-ray opaque contrast fluid is injected into the left ventricle (LV) of a patient's heart. These images are typically used to determine tracings of the endocardial boundary at end diastole (ED), when the heart is filled with blood, and at end systole (ES), when the heart is at the end of a contraction during the cardiac cycle. By manually tracing the contour or boundary of the endocardial surface of the heart at these two extremes in the cardiac cycle, a physician can determine the size and function of the left ventricle and can diagnose certain abnormalities or defects in the heart. However, such quantitative analysis is infrequently performed due to the labor required for manual segmentation. None of the many methods developed for automated segmentation has achieved clinical acceptance. We present a new method for semi-automatic segmentation of ventriculograms based on a very accurate two-stage boosted decision-tree pixel classifier. On independent test data, the classifier error rate is about 1%. The differences between semi-automatic and manual segmentation are the same magnitude as differences between human observers.

The key innovation is the use of boosted decision trees for pixel classification. Boosted decision trees have 2 principal advantages: (1) they produce excellent classifiers ("the best off-the-shelf method for classification"[1,2]) and (2) they provide standard measures of feature importance (see, for example, Hastie, Tibshirani, and Friedman[2], sec 10.13.1), which can be used to develop optimal feature sets.

Decision trees have been frequently used for problems like medical diagnosis, spam detection, etc., but have only been rarely used for pixel classification, perhaps because of misconceptions about the computational expense (see for example discussions in Sui[3] and Song[4]). In fact, decision trees allow particularly efficient implementations when the features take on values in small ranges of integers, like the [0-255] range of a pixel in an 8-bit gray scale image.

## 2. SEGMENTATION

Our method has 3 main parts: feature image calculation, pixel classification, and curve fitting.

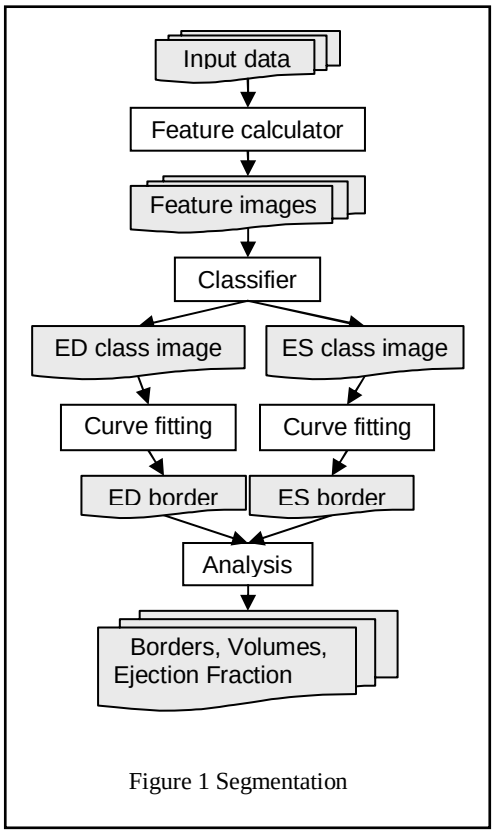


Figure 1 Segmentation

### 2.1 Screening

Our segmentation method is an example of supervised machine learning, which means it is trained from manually segmented examples. Although it could in principle be applied to any ventriculogram, it is unlikely to perform well on ventriculograms that differ in significant ways from the data used in training. In particular, this means that the automatic method should only be used on cases that pass a screening protocol used to select cases suitable for manual segmentation. We have added a few more criteria to this screening protocol to warn if the automatic method is used with cases very different from the training data.

Screening rules include: normal heart beat during the chosen ED an ES frames, proper RAO viewing angles, a minimum frame rate, no significant panning during the relevant frames, and adequate contrast.

### 2.2 Feature image calculation

The feature calculation step takes raw ventriculogram image frames and a small amount of user input, and computes a set of (currently about 90) 8 bit gray-level features images for input to the classifier. The specific feature images used was determined through a series of cross-validation experiments, discussed in section 3.

The entire feature image calculation process, described below, takes about 5-10 seconds on a 1 GHZ Pentium III.

A typical ventriculogram includes 300-400 raw gray-level image frames. The user input consists of the key ED and ES frames to segment, and locations for a small number (2-3) of anatomic landmarks in the chosen ED and ES frames. Figure 8 shows an example of the user-chosen raw ED and ES frames with possible locations for the 3 anatomic landmarks.

There are 2 main types of feature images: (1) geometry features, which encode either absolute pixel location, or location relative to the anatomic landmarks, and (2) gray-level statistics, for example, the maximum frame-to-frame change in gray level of each pixel.

Figure 9 shows 2 relative geometry feature images, in which the gray-level encodes the position of each pixel projected on an apex-base axis determined from the 3 input points.

The process for computing gray-level feature images is outlined in Figure 2.

We first select 2 subsets of the raw frames: an interval covering approximate 2 heart beats centered on the ED frame, and a similar, shorter interval centered on the ES frame.

We then calculate the feature images through: (1) masking, (2) flicker removal, and (3) gray-level statistics calculation. The first 2 parts are essentially preprocessing, aimed at making the raw image gray level statistics more comparable from case to case. The last creates the actual feature images that are used by

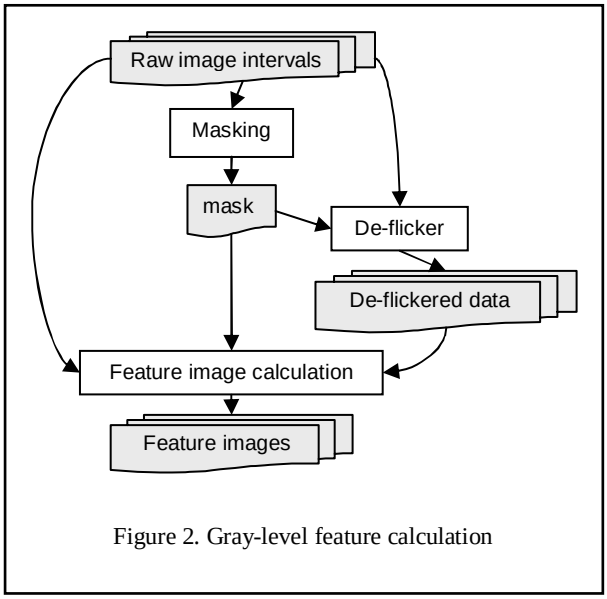


Figure 2. Gray-level feature calculation

the classifier.

### 2.2.1 Masking

Ventriculograms typically have a fairly large outer region of non-informative black or dark gray pixels, as can be seen in Figure 8. The size of these outer dark regions varies from case to case.

Some of the pixels are due to the x-ray imaging device/software, which creates a square 512x512 image even when the actual x-ray data covers a smaller, often non-rectangular, region. In order to minimize the x-ray dose to the patient, shutters may be placed in the image frame, which further obscure the outer parts of the image.

Some of the classifier features rely on gray level statistics being similar from case to case. To make these statistics as comparable as possible, I construct mask images for each case. Only pixels within the mask are used in the subsequent steps: flicker removal, feature extraction, classification, curve fitting, and analysis.

The current method uses a simple fixed mask for all cases, which has so far proved adequate. This mask is visible in Figure 10. A better alternative for the future would be to train a separate classifier to segment background foreground on a case-by-case basis.

### 2.2.2 Flicker removal

Ventriculogram image sequences often have significant flicker --- instantaneous jumps in overall brightness due to instability in the imaging device, unrelated to useful gray level variation. The jump may be complete between 2 frames, but it is often the case that there is a frame or 2 in which the upper quarter or so of the image is overall brighter or darker than the rest.

Many of our gray-level feature images are based on estimates of rates of gray level change over time, and are seriously disturbed by flicker.

The de-flickering process is outlined in Figure 3. I use repeated median regression to adjust each image frame to be more similar to a median image frame[5]. Repeated median regression is highly resistant, and will ignore up to 1/2 the data is determining its fit. This allows it to fit to the constant part of the image, ignoring the pixels that change due to ventricle motion, and also to do something reasonable with the partial jump frames.

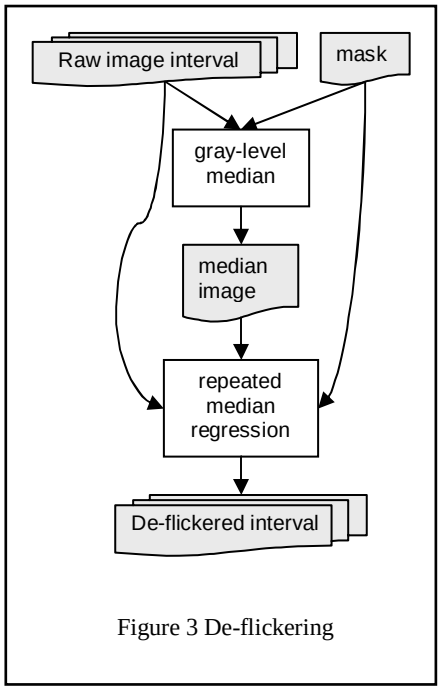


Figure 3 De-flickering

### 2.2.3 Gray-level statistics.

The gray-level feature images are computed from 4 masked image sequences: the raw and de-flickered ED and ES ventriculogram frame intervals. A set of gray-level statistics images is computed for each of these 4 sequences, with different statistics used for each of the sequences.

For example, Figure 10 shows 2 actual feature images: the result of the following calculation applied to both the de-flickered ED sequence and to the raw ES sequence:

First difference: $b[x,y,t] = a[x,y,t+1] - a[x,y,t]$.

Maximum: $c[x,y] = \max(b[x,y,*])$

Histogram equalization: $d[x,y] = \text{rank}(c[x,y],c)$

Blurring: $e = \text{smooth}(d)$

### 2.3 Pixel classification

The pixel classification step determines which pixels are inside the ventricle at the key ED (end-diastole) and ES (end-systole) frames. Our classifier is based on boosted decision trees, essentially as described by Friedman et al[2,6].

A decision tree classifier recursively partitions feature space by splitting on the value of one feature at a time. The leaves of the tree correspond to rectangular regions in feature space, each of which is assigned to a particular class. There are a variety of methods for building decision trees from training data; one of the most successful is CART[2,7].

Decision trees are commonly used for problems like medical diagnosis, but have only been rarely used for pixel classification, perhaps because of misconceptions about the computational expense (see for example discussions in Sui[3] and Song[4]). In fact, decision trees allow particularly efficient implementations when the features take on values in small ranges of integers, like the [0-255] range of a pixel in an 8-bit gray scale image.

A valid criticism of decision trees is that the partition of feature space into rectangular regions results in class boundaries (in n-dimensional feature space) that are excessively jagged. A general approach to improving this is to use decision forests, where the classification is determined by averaging a number of decision trees. The most successful decision forest methods are the variations on "boosting"[2, 5, 8, 9].

Decision forests typically limit the size of the individual trees to a relatively small number of nodes. We use 256 tree forests of 8 leaf trees. The specific numbers were chosen to balance accuracy with classification and training time.

The classifiers were trained using AdaBoost.M1[8,9], essentially as described in Hastie, Tibshirani, and Friedman[2], chapter 10. The individual trees were grown in a greedy fashion using the Gini index (see Hastie, Tibshirani, and Friedman[2], chapter 9), terminating when the fixed number of leaves was reached.

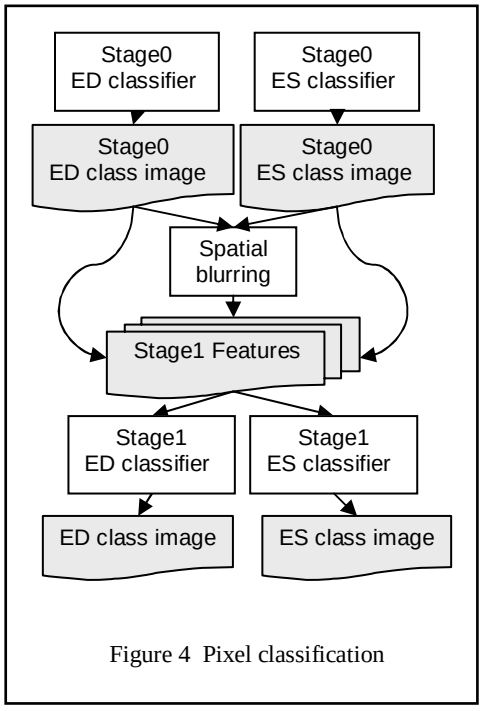

Figure 4 Pixel classification

We use a two-stage strategy, similar to that of Kamath et al.[10]. The first stage consists of 2 boosted decision-tree classifiers, one for the ED frame and one for the ES frame. The class images produced by the first stage for both ED and ES are blurred and added to the feature image set for the second stage, which also consists of 2 boosted decision trees. The two-stage strategy allows a first estimate of whether a pixel and its neighbors are inside at ED/ES to be used to refine the final estimate.

Training the two-stage classifier (including 4 boosted decision trees) takes about 6-24 hours on a 1 GHZ PIII, for training sets of about 300 cases and 80-300 feature images.

Classifying a new ventriculogram takes about 5-10 seconds, also on a 1 GHZ Pentium III.

### 2.4 Curve fitting

On independent test data, the classifier is error rate is about 1-2%. This is accurate enough that any reasonable approach for extracting a boundary from a binary class image will work.

We first use dilation and erosion[11] to determine a set of boundary pixels, shown in Figure 8.

We then fit a piecewise smooth subdivision curve to the boundary pixels using a restricted version of the surface fitting method described by Hoppe et al.[12] The fitting is done by minimizing a penalty function that combines the sum of squared distances from the boundary pixels to the curve and standard measures of curve smoothness. The minimization is done using non-linear conjugate gradients. The main reason for using subdivision curves, rather than some more familiar standard spline curve representation, was ease of implementation, which merely required simplifying existing code for fitting subdivision surfaces. However, subdivision curves also have the advantage that it is relatively simple to write down closed-form expressions for derivatives of the penalty function with respect to the positions of the control points.

Boundary pixel extraction and curve fitting takes 2-3 seconds on a 1 GHZ PIII.

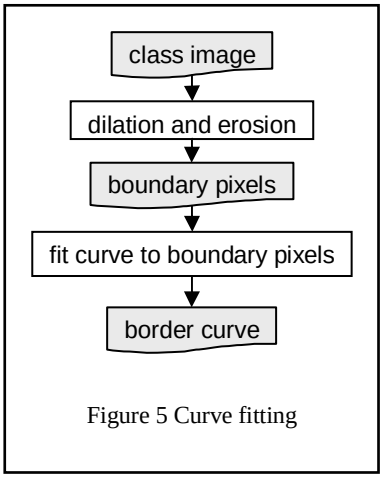

Figure 5 Curve fitting

## 3. FEATURE SELECTION THROUGH CROSS-VALIDATION

An important advantage of using a classifier based on decision trees is that it makes it feasible to screen large numbers of possible feature images.

Decision trees (and forests) are largely unaffected by the addition of uninformative feature images. The classification accuracy and time to classify are essentially unchanged. The training time increases, in the worst case, linearly in the number of features. Adding informative features, even redundant informative features, can in fact reduce the training and classification time while improving the classification accuracy.

There are standard measures for ranking feature importance in decision trees and forests (see for example, Hastie, Tibshirani, and Friedman[2], sec 10.13.1), which make it easy to identify and discard uninformative features, and also possible to reduce redundant informative features to a good minimal set.

We screen features using cross-validation experiments. Cross-validation means breaking the training data cases into *n* approximately equally sized subsets. For each of the *n* subsets, a classifier is trained using the cases in the other *n-1* subsets, comprising *(n-1)/n* of the data. Then that classifier is applied to the reserved *1/n* of the cases to classify each pixel as one of target pixel classes. Because none of the reserved cases were used to train the classifier, the result is a fair estimate of the performance of the candidate segmentation method on future data. After repeating *n* times, we have fair classifications of all the training data.

## 4. COMPARISON WITH INTEROBSERVER DIFFERENCES

We developed our segmentation method using data inherited from a previous project[3]. The data consists of 312 RAO ventriculograms, from 4 Japanese hospitals, mostly produced in 1997-8. Each case was segmented by a human observer; with different observers were assigned to different cases in an arbitrary way. 294 cases were used for training, and 18 were reserved as independent test data. Unfortunately, it is not feasible at present to collect additional, comparable independent test data to increase the test set to a more reasonable number (eg. 60+ cases) or to better balance the assignment of observers to cases.

The set of features used by the pixel classifiers was developed in a series of cross-validation experiments over a period of about 1 year. Although a single cross-validation experiment provides a fair estimate of the accuracy of the method, the sequence of experiments gradually introduces an optimistic bias. To assess the accumulated bias, we the 18 independent test cases, which did not influence the training in any way.

There is no gold standard for ventriculogram segmentation. In principle, one could attempt train an automatic method to exactly reproduce the results of a specific human 'expert', but this would be both unrealistic and inappropriate. No two human observers would segment a given image in exactly the same way. Choosing one observer to use for training an automatic method would be equivalent to nominating that observer as *the* expert, which would be difficult to justify. In any case, no single expert could be considered an absolute gold standard, because no such expert would segment a given image in exactly the same way twice.

The best result we can hope for is that our automatic method produces results that differ from manual segmentation by about as much as 2 human observers differ from each other. To assess inter-observer differences, we had a second observer re-segment 20 cases. (Because the second observer was the original observer on some of the test cases, and no other qualified observer is currently available, it was not feasible to use the 18 test cases to measure inter-observer differences.)

Figure 6 and Figure 7 show comparisons of inter-observer differences with the differences between automatic and manual segmentation (labeled "PICL") for 3 key statistics used in evaluating left ventricular status: ED and ES volumes and ejection fractions. Volumes are estimated from the boundary curves using the area-length method[13]. Ejection fraction is (ED_Volume – ES_Volume)/ED_Volume.

Standard summary statistics for this purpose include the RMS (root mean square), mean absolute, and median absolute difference between the result of the automatic segmentation method and the result of manual tracing. In clinical application, we expect the output of any such automatic method to be reviewed by a human expert, who will edit or reject large errors. The true cost in a problem like this should be roughly proportional to the number of large errors, and

relatively independent of the size of the large errors. This suggests that more 'robust' summary statistics, like median and mean absolute difference, are better measures of success than traditional least squares statistics like the RMS difference.

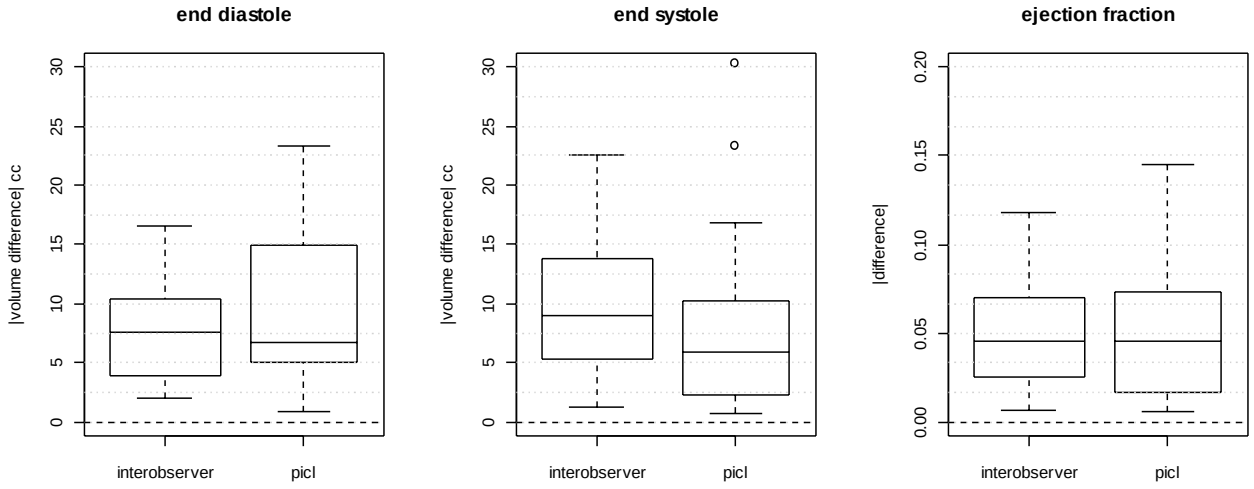


Figure 6. Boxplots comparing interobserver differences with automatic-manual differences ("picl") for ED and ES volumes and ejection fractions.

| Mean absolute differences | Interobserver | PICL |
|---|---|---|
| End Diastolic Volume, ml | 6.38 | 8.43 |
| End Systolic Volume, ml | 9.19 | 8.13 |
| Ejection Fraction | 0.050 | 0.051 |

Figure 7 Mean absolute interobserver and automatic-manual ("picl") differences.

## 5. AN EXAMPLE

Figure 8 --- Figure 15 show the stages from processing for one of the 18 test cases. This case was chosen to have roughly median accuracy (among the test cases) in ED and ES volumes and ejection fraction.

One thing to note is that, in this particular case, the Stage 0 classifier output may have produced a more accurate curve than the Stage 1, particularly for ES. In general, however, the Stage 1 results are better than Stage 0.

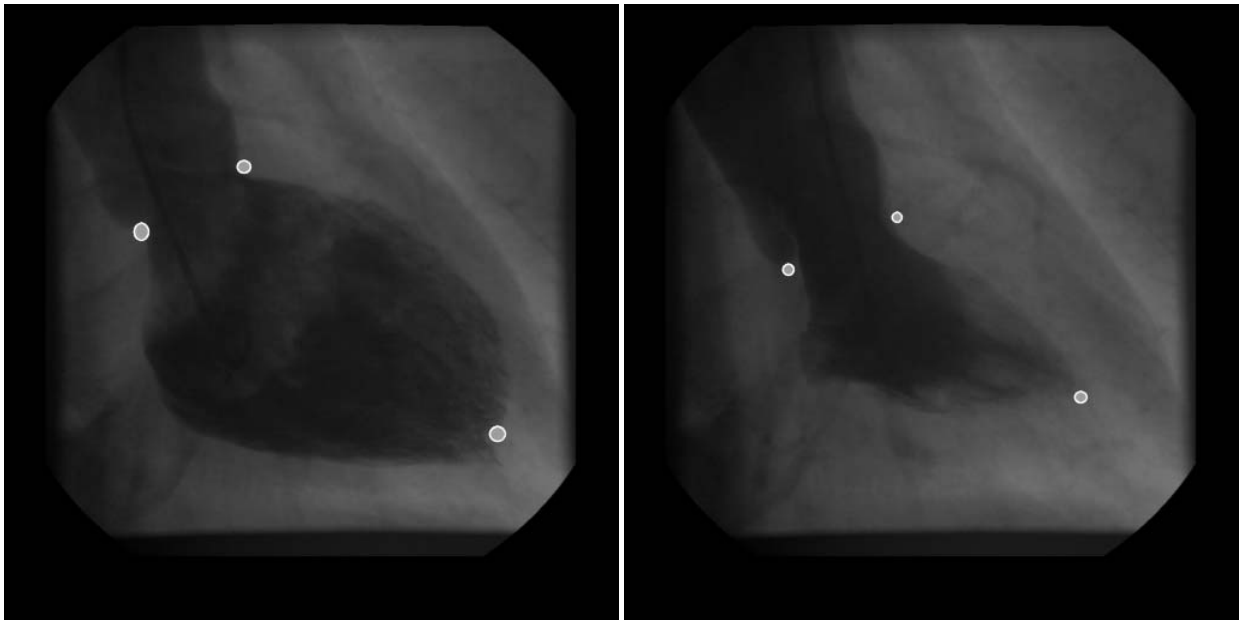

Figure 8 Raw ED and ES frames with possible user input points

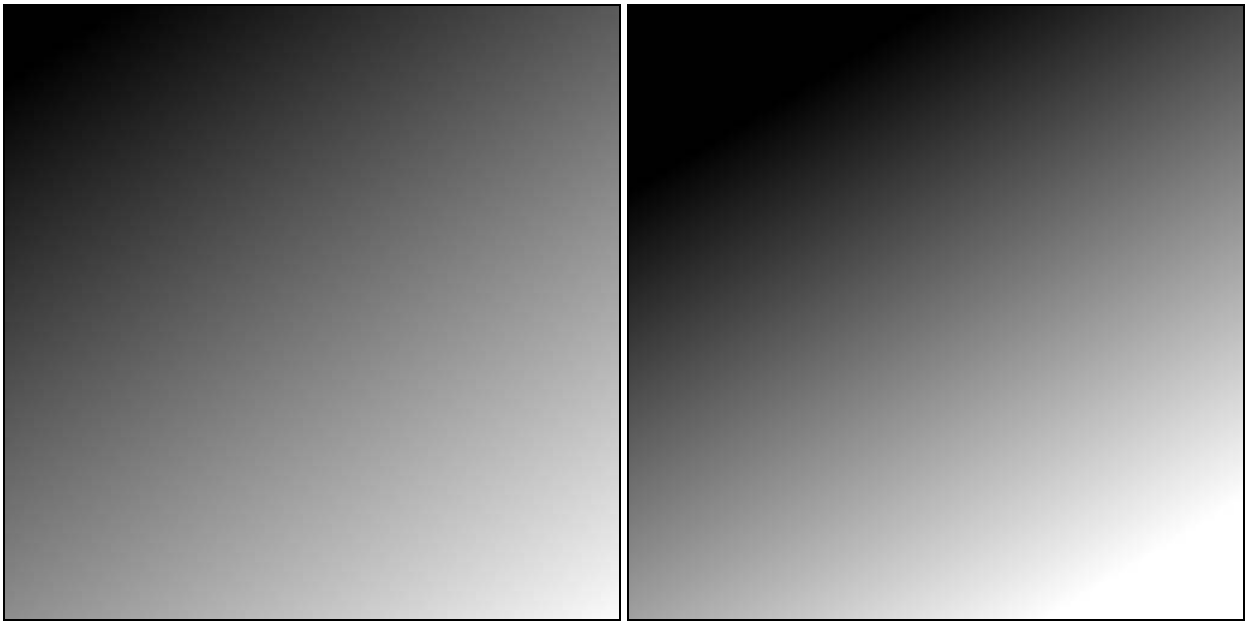

Figure 9 Two relative geometry feature images: apex-base axial coordinate for ED and ES.

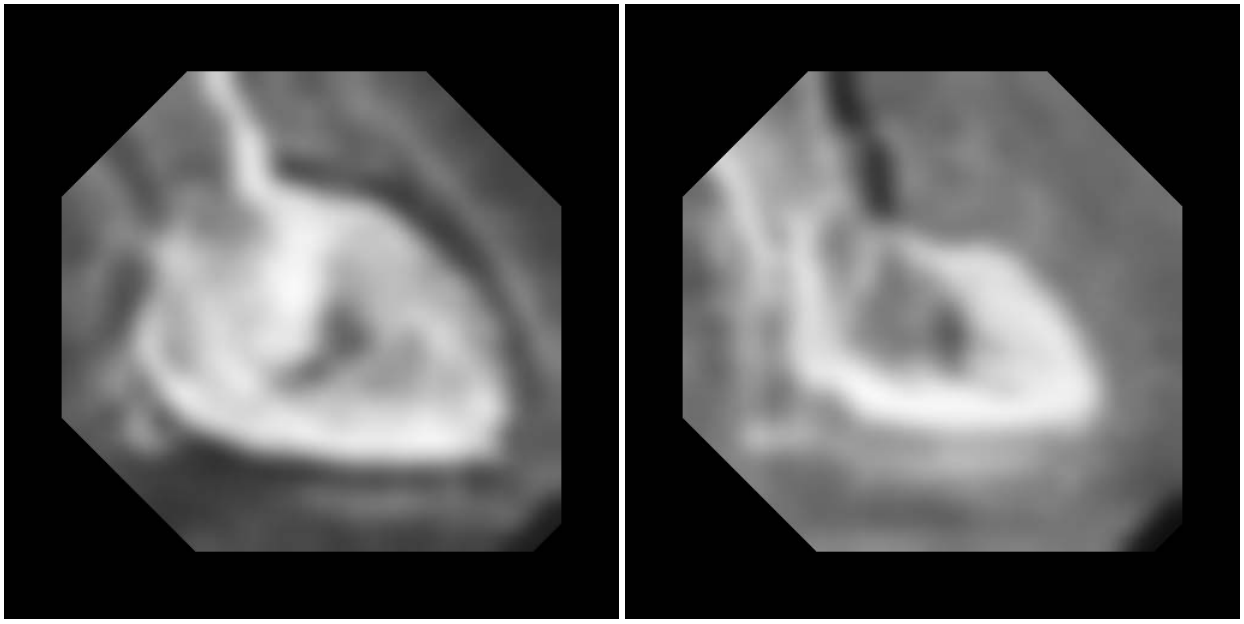

Figure 10 Two gray level feature images: smoothed, histogram-equalized, maximum frame-to-frame gray level change for intervals around the ED and ES frames. ED is de-flickered, ES uses raw image frames.

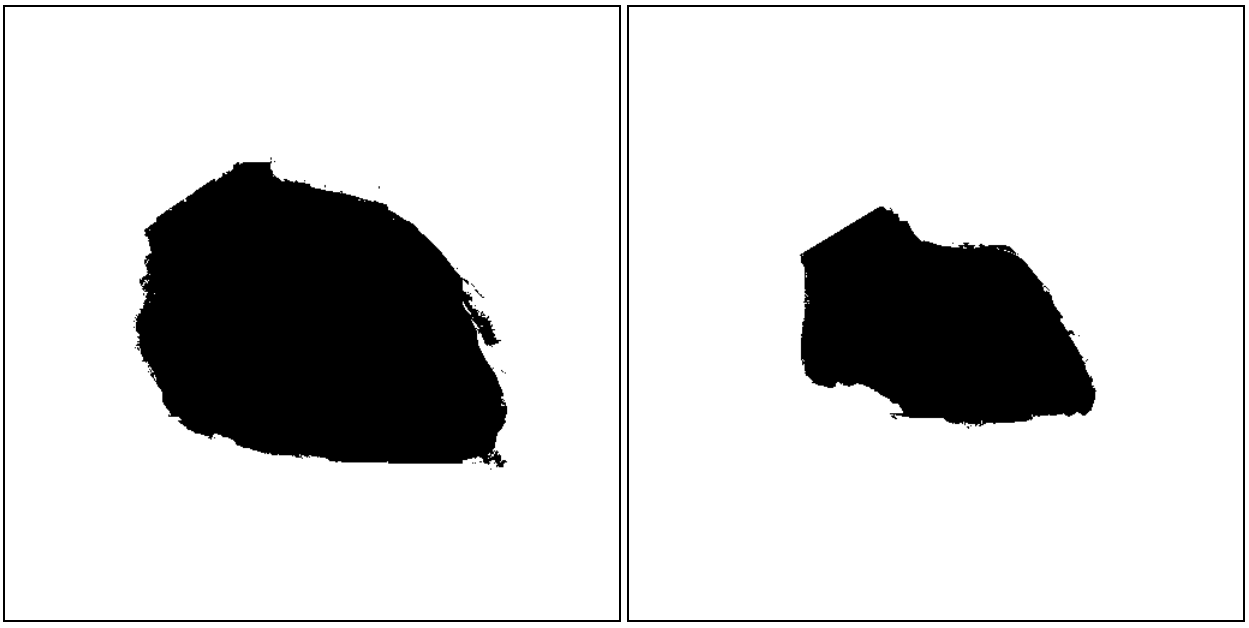

Figure 11 Stage 0 class images

Figure 12 Blurred stage 0 class images

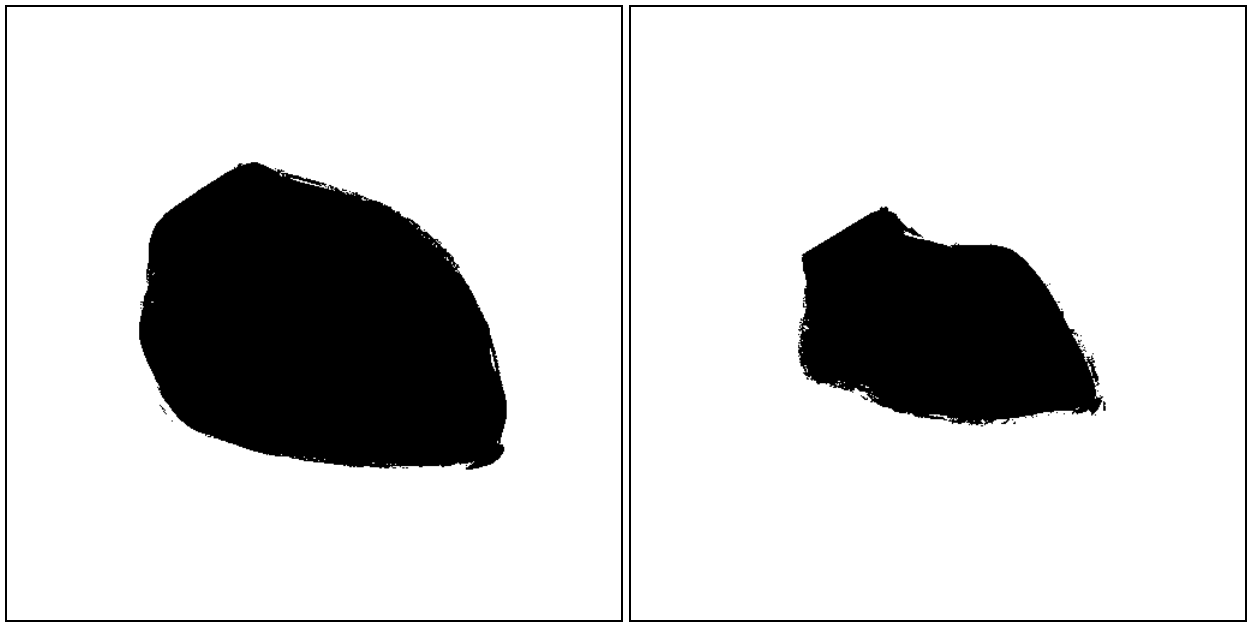

Figure 13 Stage 1 class images

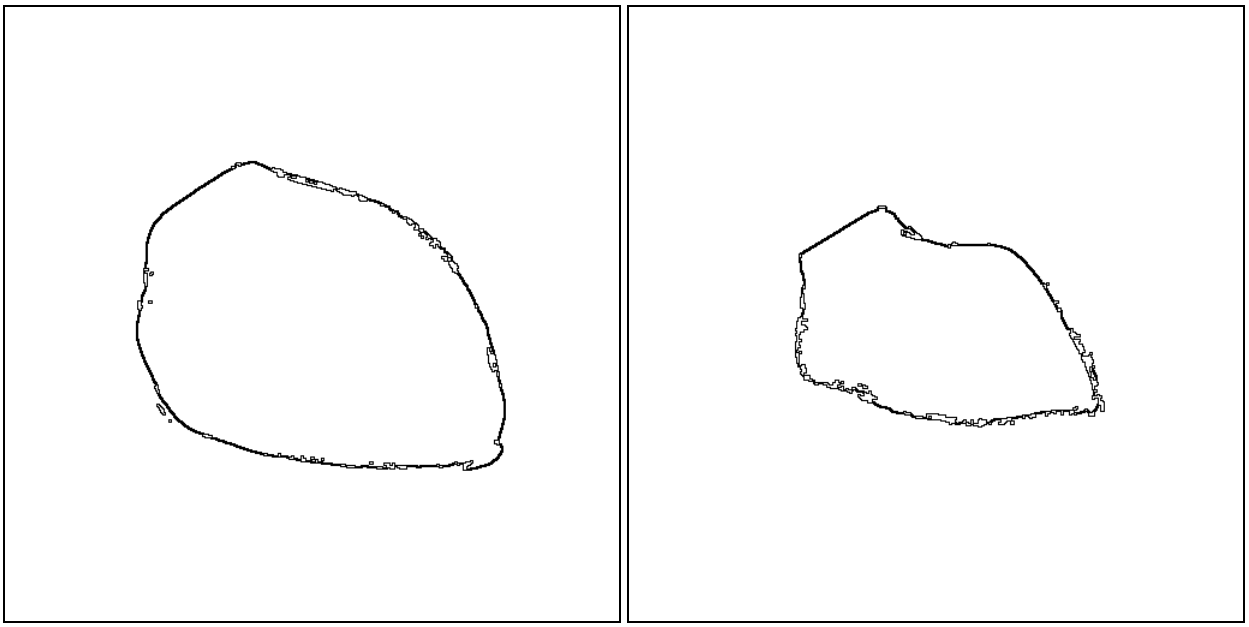

Figure 14 Boundary pixels.

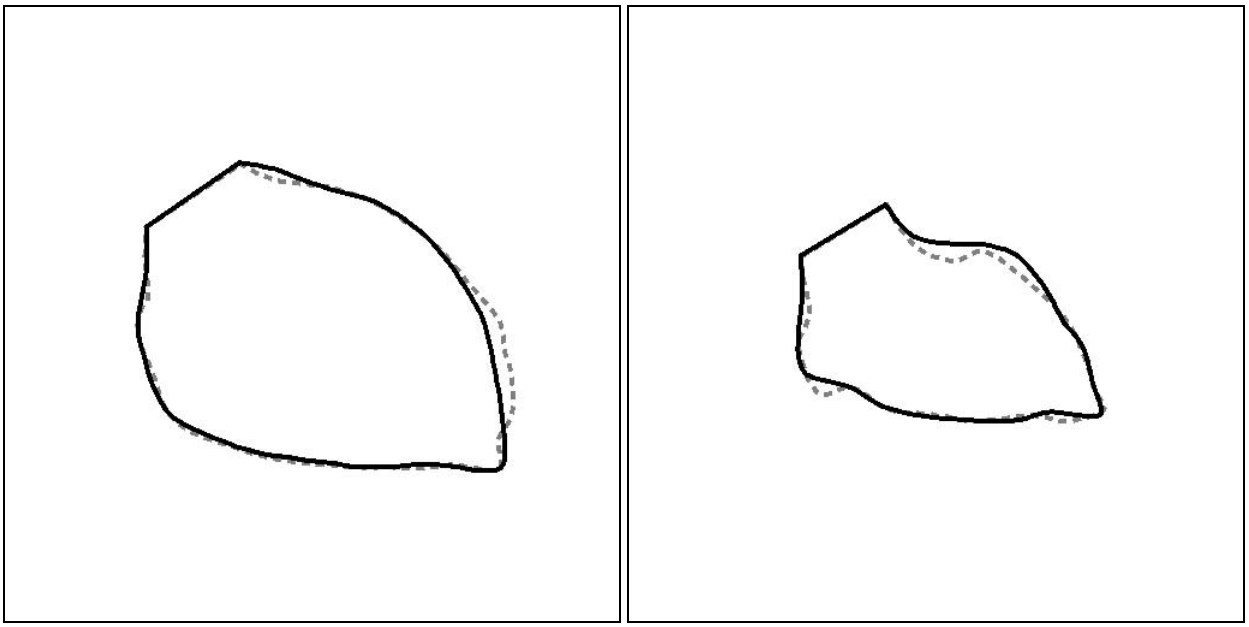

Figure 15  Fitted curve (black) and manual curve (dashed gray).

## ACKNOWLEDGEMENTS

This research was supported by grants from the Goodman Co., Ltd. (Nagoya, Japan).

## REFERENCES

1. Leo Breiman, "Arcing Classifiers (with discussion)," *Annals of Statistics* **26**: p. 801-849, 1998.
2. Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2001.
3. Lei Sui, "Automated left ventriculogram boundary detection". Ph.D. Thesis, Bioengineering, U of Washington, 2000.
4. Mingzhou Song, "Integrated Surface Model Optimization from Images and Prior Shape Knowledge," Ph.D. Thesis, Electrical Engineering, U. of Washington, 2002.
5. Andrew F. Siegel, "Robust regression using repeated medians," *Biometrika* **69**:242--244, 1982.
6. Jerome Friedman, Trevor Hastie, and Robert Tibshirani, "Additive Logistic Regression," *Annals of Statistics* **28**:337-374, 2000.
7. Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone, *Classification and Regression Trees*, Wadsworth, 1984.
8. Yoav Freund and Robert Schapire, "Experiments with a new boosting algorithm," *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, p. 325-332, 1996.
9. Yoav Freund and Robert Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *Journal of Computer and System Sciences* **55**: 119-139, 1997.
10. Chandrika Kamath, Sailes K. Sengupta, Douglas Poland, and John A. H. Futterman, "On the use of machine vision techniques to detect human settlements in satellite images," *Image Processing: Algorithms and System II, SPIE Electronic Imaging*, Santa Clara CA, January 22 2003.
11. Milan Sonka, Vlaclav Hlavac, and Roger Boyle, *Image processing, analysis, and machine vision*, PWS Publishing 1999.
12. Hugues Hoppe, Anthony DeRose, Tom Duchamp, Mark Halstead, Hubert Jin, John Alan McDonald, Jean Schweitzer, and Werner Stuetzle. "Piecewise smooth surface reconstruction," *Computer Graphics, Proceedings, Annual Conference Series, (SIGGRAPH94)* 1994.
13. H.T. Dodge, H. Sandler, D.W. Ballew, and J.D. Lord, Jr., "The use of biplane angiocardiography for the measurement of left ventricular volume in man," *Am Heart J* **60**:762-776, 1960.