

# Correction of Systematic Errors in Automatically Produced Boundaries from Low-Contrast Ventriculograms

Jasjit S. Suri <sup>†</sup>, Robert M. Haralick <sup>†</sup>

<sup>†</sup>Intelligent Systems Laboratory  
Department of Electrical Engineering  
University of Washington, Seattle, WA 98195

Florence H. Sheehan <sup>‡</sup>

<sup>‡</sup>Cardiovascular Research & Training Center  
University of Washington Medical Center  
University of Washington, Seattle, WA 98195

## Abstract

Poor contrast in the apex zone and non-homogeneous mixing of the dye with the blood in the left ventricle causes the left ventricle *pixel-based* classifiers operating on ventriculograms [1] to yield boundaries which are not close to *ground truth* boundaries as delineated by the cardiologist. They have a mean boundary error of about 6.4 mm and an error of about 12.5 mm in the apex zone. These errors have a systematic positional and orientational bias, the boundary being under-estimated in the apex zone.

This paper discusses two calibration methods: *the identical coefficient* and *the independent coefficient* to remove these systematic biases. From these methods, we constitute a combined algorithm which reduces the boundary error compared to either of the calibration methods. The algorithm, in a greedy way, computes *which and how many* vertices of the left ventricle boundary can be taken from the computed boundary of each method to best improve the performance.

The corrected boundaries have a mean error of less than 3.5 mm with a standard deviation of 3.4 mm over the approximately  $6 \times 10^4$  vertices in the data set of 291 studies. Our methodology reduces the mean boundary error by 2.9 millimeters over the boundary produced by the classifier. We also show the calibration algorithm performs better in the apex zone where the dye is unable to reach. For end-diastole, it reduces the error in the apex zone by 8.5 millimeters over the *pixel-based* classifier boundaries.

**Key Words:** Left Ventricle, Low Contrast, Boundaries, Calibration, Motion, Quantitative Analysis, Polyline

## I. INTRODUCTION

In the X-ray ventriculograms of the left ventricle (LV), the boundary at the *apex* moves at a different rate than the *inferior* and *anterior* walls during the heart cycle [2], [3]. Besides this, the gray scale left ventriculogram (LVG) have poor contrast with a high level of noise. This noise is due to the scattering of radiation by tissue volume which is not related to the LV [4], artifacts generated by breathing of the patient during catheterization procedure, and interference of ribs and diaphragms. This makes the boundary estimation process of the LV very difficult. The contrast medium (a Barium compound dye) non-uniformly mixes with the blood and the apex zone of the LV typically does not receive much dye [5]. As a result, the initial bound-

aries produced by a *pixel-based* classifier *fall short* (under estimated) in the apex zone with respect to the ground truth boundaries [1]. The calibration procedure discussed in this paper removes all systematic position, orientation, and shape errors from the initial classifier boundaries.

We employ two different calibration techniques: *the identical coefficient* and *the independent coefficient*. Each method produces estimates for the vertices of the polygon bounding the LV. We then combine these vertex sets to form a final boundary. We select in a greedy way, that fixed subset of estimated vertex positions from each method which when combined together minimizes the resulting error between the final estimated polygon boundary and the ground truth.

Our limited database consists of  $N=291$  patient studies, each having  $F=2$  frames, end-diastole (ED) and end-systole (ES), and having a ground truth polygonal boundary of  $P=100$  vertices, and a 100 vertex raw boundary created from *pixel-based* classifier. To produce estimates of performance based on this database which are not biased high, we use a *cross-validation* methodology. The *partition protocol* uses a database of  $N$  patients studies and partitions into  $K$  subsets each containing  $\frac{N}{K}$  studies. Estimates from each calibration transformation are obtained using  $L$  of the  $K$  subsets. Rotating through all  $L$  choose  $K$  combinations, we measure the accuracy of the results on the remaining  $K-L$  subsets using the *polyline distance metric*. The mean and standard deviation of the resulting set of  $N \times F \times P \times \frac{(K-1)!}{(K-L-1)!L!}$  numbers is then used to estimate the overall performance.

Because of the small number of patient studies,  $N=291$  and large number of parameters (about 200 times  $N$ ) in the transformation, there is a danger of *memorization* rather than *generalization* in the estimation of the transformation parameters. As  $P$  decreases, the generalization will be better but the representation of the true LV shape will get worse causing higher error. As  $P$  increases, generalization will be lost but the representation of the true LV shape will get better. With the other parameters  $K, L$  and  $N$  fixed, there will be an *optimal* number of boundary vertices balancing the *representation error* with the *memorization error*. Our protocol finds this *optimal* number.

## II. TWO BOUNDARY CALIBRATORS

Ground truth boundaries refer to the hand delineated boundaries traced by the cardiologist. Raw or initial or

classifier boundaries are the the boundaries produced by the *pixel-based* classification algorithm [1].

In the *identical coefficient* method, the estimated  $x$  and estimated  $y$  coordinates are computed using the *same* linear combination of raw  $x$  and raw  $y$  coordinates associated with that vertex of the LV boundary. In the *independent coefficient* method, the estimated  $x$  and estimated  $y$  coordinates are computed as a *different* linear combination of the raw  $x$  and raw  $y$  coordinates associated with that vertex of the LV boundary. The problem of boundary estimation then reduces to a problem of determining the coefficients of the linear combination. This can be accomplished by solving a regression problem as discussed below.

#### A. Identical Coefficient Method (IdCM) for any frame

Let  $g'_n$  and  $h'_n$  be the row vectors of  $x$ -coordinates and  $y$ -coordinates respectively for the ground truth LV boundaries for patient  $n$ , where  $n = 1, \dots, N$ . Let  $r'_n$  and  $s'_n$  be the row vectors of  $x$ -coordinates and  $y$ -coordinates respectively for the classifier boundary for any patient  $n$ , where  $n = 1, \dots, N$ . For any frame of the heart cycle  $t$ , the calibrated boundary of the LV in LVG using the IdCM, we are:

- Given: Corresponding pairs of ground truth boundaries  $\mathbf{R}$  [ $2N \times P$ ], and the classifier boundaries  $\mathbf{Q}$  [ $2N \times (P + 3)$ ], respectively:

$$\mathbf{R} = \begin{pmatrix} g'_1 \\ h'_1 \\ \dots \\ g'_N \\ h'_N \end{pmatrix} \quad \mathbf{Q} = \begin{pmatrix} r'_1 & \underbrace{1 \ u_{11} \ u_{21}} \\ s'_1 & \underbrace{1 \ v_{11} \ v_{21}} \\ \dots & \dots \\ r'_N & \underbrace{1 \ u_{1N} \ u_{2N}} \\ s'_N & \underbrace{1 \ v_{1N} \ v_{2N}} \end{pmatrix}$$

where,  $(u_{1n}, v_{1n})$ ,  $(u_{1n}, v_{1n})$  and  $(u_{2n}, v_{2n})$ ,  $(u_{2n}, v_{2n})$  are the coordinates for the anterior aspect (AAV), (first vertex of LVC) and inferior aspect (IAV), (last vertex of LVC) of the AoV plane of the LV from ground truth boundary for patient  $n$ . Column with unity is introduced due to the translation effect.

- Let  $\mathbf{A}$  [ $(P + 3) \times P$ ] be the unknown regression coefficient matrix that minimizes  $\|\mathbf{R} - \mathbf{Q}\mathbf{A}\|^2$ .

#### B. Independent Coefficient Method (InCM) for any frame

Using the same notation:  $g'_n$ ,  $h'_n$ ,  $r'_n$  and  $s'_n$ , the calibrated boundary of the LV in LVG using the InCM, we are:

- Given: Corresponding ground truth boundaries  $\mathbf{R}$  [ $N \times 2P$ ], and the classifier boundaries  $\mathbf{Q}$  [ $N \times (2P + 5)$ ] respectively:

$$\mathbf{R} = \begin{pmatrix} g'_1 \ h'_1 \\ \dots \\ g'_N \ h'_N \end{pmatrix} \quad \mathbf{Q} = \begin{pmatrix} r'_1 \ s'_1 \ \underbrace{1 \ u_{11} \ v_{11} \ u_{21} \ v_{21}} \\ \dots \\ r'_N \ s'_N \ \underbrace{1 \ u_{1N} \ v_{1N} \ u_{2N} \ v_{2N}} \end{pmatrix}$$

where, symbols have same meaning.

- Let  $\mathbf{A}$  [ $(2P + 5) \times 2P$ ] be unknown regression coefficient matrix that minimizes  $\|\mathbf{R} - \mathbf{Q}\mathbf{A}\|^2$ .

Then for any classifier boundary matrix  $\mathbf{Q}$  produced by the image processing algorithm, the calibrated coordinates of the boundary are given by  $\mathbf{Q}\hat{\mathbf{A}}$ , where  $\hat{\mathbf{A}}$  is the estimated coefficients. The above two methods are different in

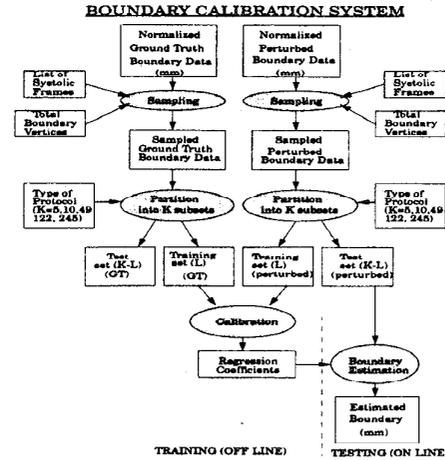


Fig. 1. Boundary Calibration System for running IdCM and InCM algorithms for any frame  $t$  of the heart cycle. We sample and interpolate the 100 vertex polygonal boundary to  $P_3$  vertices, then partition the database into  $K$  subsets,  $L$  used for training and  $K - L$  used for testing. Magnification conversion factors are used to convert pixels to mm, where 1 pixel = 0.39 mm.

the way the calibration model is set up. In IdCM formulation, the coefficients that multiply  $g'_n$  also multiply  $h'_n$ , hence the name *identical coefficient* method. In InCM, the new  $(x, y)$ -coordinates of the vertices of each boundary is a *different* linear combination of the old  $(x, y)$ -coordinates for the polygon, hence the name *independent coefficient* method. For IdCM, the number of coefficients estimated in the  $\hat{\mathbf{A}}$  matrix is  $(P + 3) \times P$ . For InCM, the number of coefficients estimated is  $(2P + 5) \times 2P$ . Thus the InCM requires around 4 times the number of coefficients of IdCM, and this difference represents a significant factor in the ability of the technique to *generalize* rather than *memorize* for our data size ( $N$ ).

### III. GREEDY ALGORITHM: CALIBRATION BY VERTEX

Fig. (1) shows the overall boundary calibration system, where the heart of the system is the IdCM or InCM *calibrator*. In either case of the calibration model, we find the off-line coefficient matrix  $\mathbf{A}(t)$  to minimize:

$$\epsilon_{ifr}^2 = \|\mathbf{R}(t) - \mathbf{Q}(t)\hat{\mathbf{A}}(t)\|^2. \quad (1)$$

Using the *partition protocol*, and generalizing for any frame  $t$ , the minimizing  $\hat{\mathbf{A}}_{tr}$  and estimated boundaries  $\hat{\mathbf{R}}_{te}$  on the test set  $\mathbf{Q}_{te}$ , are:

$$\hat{\mathbf{A}}_{tr} = (\mathbf{Q}_{tr}^T \mathbf{Q}_{tr})^{-1} \mathbf{Q}_{tr}^T \mathbf{R}_{tr}, \quad \hat{\mathbf{R}}_{te} = \mathbf{Q}_{te} \hat{\mathbf{A}}_{tr} \quad (2)$$

$\hat{\mathbf{A}}_{tr}$  is calculated using a singular value decomposition. Given the test set ( $\mathbf{Q}_{te}$ ) or training set ( $\mathbf{Q}_{tr}$ ), we can estimate the calibrated boundary for IdCM and InCM as:

$$\hat{\mathbf{R}}_{id} = \mathbf{Q}_{te} \hat{\mathbf{A}}_{id}, \quad \& \quad \hat{\mathbf{R}}_{in} = \mathbf{Q}_{te} \hat{\mathbf{A}}_{in} \quad (3)$$

### A. Best InCM index selection: Greedy Algorithm

Fig. (2) shows the object process diagram for combining the IdCM and InCM LV boundaries using a Greedy algorithm. We select a fixed subset of estimated vertex positions from these technique which when combined together, minimizes the resulting error between the final estimated polygon boundary and the physician traced LV boundary. Let  $S$ ,  $S_{id}$  and  $S_{in}$  be three sets consisting of *all* vertices, IdCM pool vertices and InCM pool vertices respectively. Let  $\hat{R}_{id}$  and  $\hat{R}_{in}$  be the estimated boundary matrices from IdCM and InCM techniques. Let  $R_{gt}$  and  $R$  be the matrices consisting of  $(x, y)$  coordinates from original ground truth with  $P_1=100$  and sampled  $P_s$  vertices respectively. Initially all the vertices are considered in the IdCM pool and the error is computed. Denote its error by  $\epsilon_{id}$ . Now we select that vertex from IdCM pool which when combined with InCM pool vertices yields an estimated boundary error lower than  $\epsilon_{id}$ . This procedure is repeated until there is no further improvement. If  $\epsilon$  be the error at any time in the Greedy *do-while* loop, then, the Greedy boundary calibration algorithm consists of the following steps.

#### Greedy Boundary Calibration Algorithm

```

 $S_{id}=S; S_{in}=\phi, \epsilon=0, gc=0$  /* greedy counter */
While ( $\epsilon \leq \epsilon_{id}$ ) do
  gc++
  For each  $i \in S_{id}$ , /* total vertices are  $P_{id}$  */
     $S_{id} = S_{id} - \{i\}$  ;  $S_{in} = S_{in} \cup \{i\}$ 
    Combine IdCM ( $\hat{R}_{id}$ ) and InCM ( $\hat{R}_{in}$ ):
     $\hat{R}_{com} = \text{Combine}(R_{id}, R_{in}, N, P_2, S_{id}, S_{in}, gc)$ 
    Performance Evaluation:
     $\epsilon_i = \text{PolyPerformance}(\hat{R}_{com}, R_{gt}, N, P_1, P_2)$ 
  end /* end of for loop */
  ArgMin Comp.: Min. error & best vertex  $j$  selection:
  ( $\epsilon_{min}, j$ ) = ArgMin( $\epsilon[i], P_{id} - gc$ )
  if ( $\epsilon_{min} < \epsilon$ ) then  $S_{id}=S_{id}-\{j\}$  ;  $S_{in}=S_{in} \cup \{j\}$ 
  else break; endif
end /* end of while loop */

```

### IV. POLYLINE BASED PERFORMANCE

The polyline distance  $D_s(B_1 : B_2)$  between two polygons representing boundary  $B_1$  and  $B_2$  is symmetrically defined as the average distance between a vertex of one polygon to the boundary of the other polygon. To define this measure precisely, first requires having defined a distance  $d(v, s)$  between a point  $v$  and a line segment  $s$ . The distance  $d(v, s)$  between a point  $v$  having coordinates  $(x_0, y_0)$ , and a line segment having end points  $(x_1, y_1)$  and  $(x_2, y_2)$  is:

$$d(v, s) = \begin{cases} \min\{d_1, d_2\}; & \text{if } \lambda < 0, \lambda > 1 \\ |d^\perp|; & \text{if } 0 \leq \lambda \leq 1, \end{cases} \quad (4)$$

where

$$\begin{aligned} d_1 &= \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \\ d_2 &= \sqrt{(x_0 - x_2)^2 + (y_0 - y_2)^2} \\ \lambda &= \frac{(y_2 - y_1)(y_0 - y_1) + (x_2 - x_1)(x_0 - x_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\ d^\perp &= \frac{(y_2 - y_1)(x_1 - x_0) + (x_2 - x_1)(y_0 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \end{aligned} \quad (5)$$

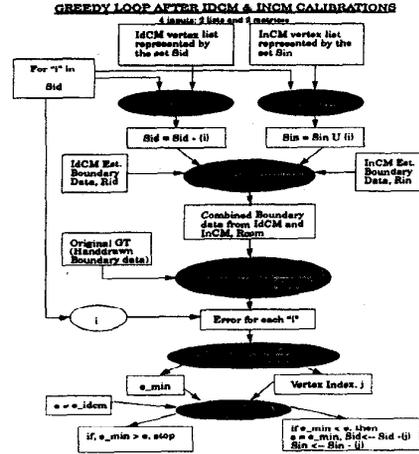


Fig. 2. For loop for combining the IdCM and InCM calibrations used in the greedy do-while loop. The data structure of the do-while loop is implemented in the form of lists where the best vertex is computed and emptied from the IdCM pool and unioned in the InCM pool. The do-while loop is implemented independently for ED and ES frames.

The distance  $d_b(v, B_2)$  measuring the polyline distance from vertex  $v$  to the boundary  $B_2$  is defined by:

$$d_b(v, B_2) = \min_{s \in \text{sides } B_2} d(v, s) \quad (6)$$

The distance  $d_{vb}(B_1, B_2)$  between the vertices of polygon  $B_1$  and the sides of polygon  $B_2$  is defined as the sum of the distances from the vertices of the polygon  $B_1$  to the closest side of  $B_2$ .

$$d_{vb}(B_1, B_2) = \sum_{v \in \text{vertices } B_1} d(v, B_2)$$

On reversing the computation from  $B_2$  to  $B_1$ , we can similarly compute  $d_{vb}(B_2, B_1)$ . Using Eq. 6, the polyline distance between polygons,  $D_s(B_1 : B_2)$  is defined by:

$$D_s(B_1 : B_2) = \frac{d_{vb}(B_1, B_2) + d_{vb}(B_2, B_1)}{(\#\text{vertices} \in B_1 + \#\text{vertices} \in B_2)} \quad (7)$$

Using the definition of the polyline distance between 2 polygons, we can now compute the overall mean error of the system, denoted by  $e_{NFP}^{poly}$  and defined by:

$$e_{NFP}^{poly} = \frac{\sum_{i=1}^F \sum_{n=1}^N D_s(G_{nt}, C_{nt})}{F \times N} \quad (8)$$

where,  $D_s(G_{nt}, C_{nt})$  is the polyline distance between the ground truth  $G_{nt}$  and calibrated polygons  $C_{nt}$  for patient study  $n$  and frame number  $t$ . Using the definition of the polyline distance between 2 polygons, the standard deviation can be computed as:

$$\sigma_{NFP}^{poly} = \left\{ \frac{\sum_{i=1}^F \sum_{n=1}^N \sum_{v \in \text{ver. } G_{nt}} (d_b(v, C_{nt}) - e_{NFP}^{poly})^2}{N \times F \times (\#\text{ver.} \in B_1 + \#\text{ver.} \in B_2)} + \frac{\sum_{i=1}^F \sum_{n=1}^N \sum_{v \in \text{ver. } C_{nt}} (d_b(v, G_{nt}) - e_{NFP}^{poly})^2}{N \times F \times (\#\text{ver.} \in B_1 + \#\text{ver.} \in B_2)} \right\}^{\frac{1}{2}} \quad (9)$$

### A. Error per vertex & Error per Arc Length

Using the polyline distance formulae, we can compute the *error per vertex* from one polygon (ground truth) to another polygon (calibrated). This is defined as the mean error for a vertex  $v$  over all the patients and all the frames. The *error per vertex* for a fixed vertex  $v$  when computed from ground truth to calibrated boundary, defined by:

$$e_v^{GC} = \frac{\sum_{t=1}^F \sum_{n=1}^N d_b(v, G_{nt})}{F \times N} \quad (10)$$

Similarly, we can compute the *error per vertex* from calibrated to ground truth using Eq. (10). *Error per arc length* is computed in the following way: For the values  $e_v^{GC}$  where  $v=1, 2, 3, \dots, P_1$ , we construct a curve  $f^{GC}$  defined on the interval  $[0,1]$  which takes the value  $e_v^{GC}$  at point  $x$  which is the *normalized arc length* to vertex  $v$  and whose inbetween values are defined by linear interpolation. We compute the curve  $f^{CG}$ , from calibrated boundary to ground truth boundary in a similar way. We then add algebraically these two curves to yield the final *error per arc length*, given as:  $f = \frac{f^{GC} + f^{CG}}{2}$ .

### V. DATA ANALYSIS: 3 ALGORITHMS

The performance of the system can be judged by evaluating the error measures on the test data set,  $Q_{te}$ . We here show the performance of IdCM, InCM and the Greedy algorithms. All our performance is with respect to the original ground truth boundaries having  $P_1=100$  vertices. We analyze the data in about 9 different ways [6] but we present here the main three.

#### A. Data Analysis 1: Vertex Optimization & Pooling

We find the mean error ( $e_{NFP}^{poly}$ ) as a function of the number of sampled polygon boundary vertices  $P_2$  on the LVC. The optimization curve and operating point is shown in fig. 3 (left). We use  $N=291$ ,  $K=145$ ,  $L=144$ . We now vary the number of vertices  $P_2$  on the LV polygon varying it from 10 vertices to 90 vertices with 5 vertex increments. If  $L=144$  are the training sets then for each combination there are  $K-L$  test set boundaries on which the error is computed. We choose the number of vertices  $P_2$  to minimize the error on the test set. Since there are  ${}^K C_{L-145}$  trials, each trial has  $(K-L)$  subsets, each subset consists of  $\frac{N}{K}$  patients (in a protocol if  $\frac{N}{K}$  is not a perfect division, then for the last trial in  ${}^K C_L$  combinations, we have  $(K-L+r_p)$  patients as testing set, where  $r_p$  is a remainder number of  $\frac{N}{K}$ ) and each patient consists of  $P_2$  vertices and  $F=2$  frames. We thus get the total number of vertices as:  $F \times {}^K C_L \times (K-L) \times \frac{N}{K} \times P_2$ , resulting in:  $N \times F \times P_2 \times \frac{(K-1)!}{(K-L-1)!L!}$  vertices for each  $(N, K, L, F, P_2)$  tuple. We see from the fig. 3 (left) that the *optimal* number of vertices in InCM are 15, *half* the optimal number of vertices in IdCM which are 30, the reason being that the number of coefficients that have to be estimated in InCM is about *four* times the number of coefficients that have to be estimated in IdCM. Fig. 3 (right) shows the effect of the Greedy calibration algorithm, where,

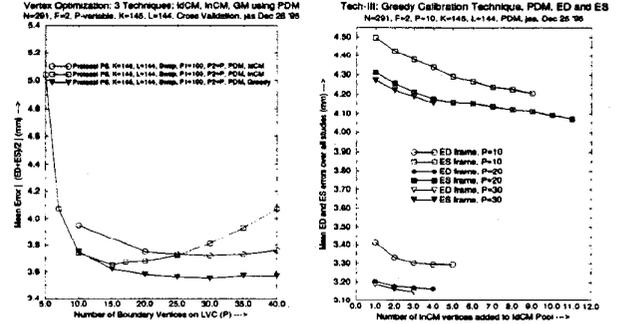


Fig. 3. Left: Vertex Optimization using the polyline distance metric (PDM) for 3 calibration techniques, Idcm, Incm and Greedy. The IdCM operating point is 30 vertices and InCM operating point is 15 vertices. Greedy does the best. Partition Protocol Parameters:  $N=291$ ,  $F=2$ ,  $K=145$ ,  $L=144$ ,  $P_1=100$ ,  $P_2=30$ . Mean error for IdCM=3.8 mm, InCM=3.9 mm and Greedy=3.5 mm. So greedy improve by 0.3 mm over IdCM method. Right: Plot showing the reduction in the error for ED and ES frames when some vertices are calibrated using IdCM and others using InCM. With the increase InCM pool, the error drops.

ED and ES frame errors drop when the IdCM pool vertices are transferred to InCM pool. This is implemented using the Greedy *do-while* loop where some columns (or vertices) of IdCM matrix  $\hat{R}_{id}$  are replaced by corresponding columns (or vertices) of InCM matrix,  $\hat{R}_{in}$  yielding  $\hat{R}_{com}$ . Fig. 3 (left) shows that the Greedy algorithm reduces the error by 0.3 mm over IdCM. The best performance over all the three techniques is by the Greedy algorithm with the *optimal* number of vertices being 30. Note also that in Greedy calibration technique, the error does not rise very sharply after 30 vertices but rises gradually by  $(\frac{1}{100})^{th}$  of a mm.

#### B. Data Analysis 2,3: Cumulative distribution of $(\frac{ED+ES}{2})$ errors and Error Per Arc Length along LVC

Fig. 4 shows the cumulative distribution of end frame errors ( $\frac{ED+ES}{2}$ ) from IdCM, InCM and Greedy algorithms. Fig. 5 demonstrates the mean *error per arc length* along the LVC. The abscissa shows the length of the arc starting from AAV. The ordinate shows the error at each vertex in mm. As seen in the plot, the mean *error per vertex* is largest near the middle of the normalized arc length which is close to the apex of the LV. Thus the error is maximum in the apex region. We see that Greedy algorithm does *best* in the apex zone compared to the IdCM and InCM methods. The *error per vertex* in fig. 5 shows that in the ED frame, the apex zone error is reduced by 8.5 mm (from 12.5 mm to about 4 mm), while in ES frame, the apex zone error is reduced by 3 mm (from 9 mm to 6 mm). The corresponding mean error over ED and ES frames of the *pixel-based* boundaries was 6.4 mm which is reduced to 3.8 mm in IdCM and 3.5 mm in Greedy. As per our assumption, the error is least at the end points of the LVC since the AoV plane is known, thus the *error per vertex* curve drops at both ends (fig. 5).

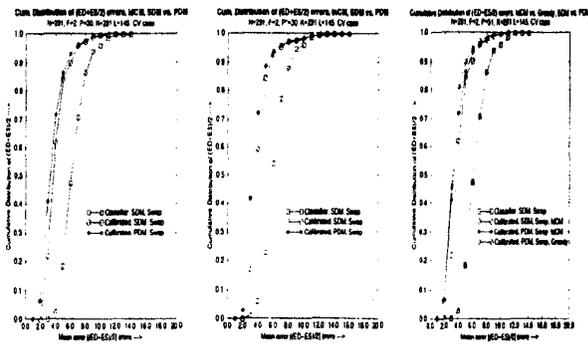


Fig. 4. Cumulative distribution vs. mean error of  $\frac{BD+ES}{2}$  errors. Left: Identical coefficient method. Middle: Independent coefficient method. Right: Greedy vs. IdCM. The curves shows that 80% patient estimated boundaries have error  $\leq 4$  mm in IdCM, while 72% of the patients have error  $\leq 4$  mm in InCM, and 81% of the patients have error  $\leq 4$  mm in Greedy. Calibration Parameters:  $N=291$ ,  $F=2$ ,  $K=145$ ,  $L=144$ ,  $P_1=100$ ,  $P_2=30$ .

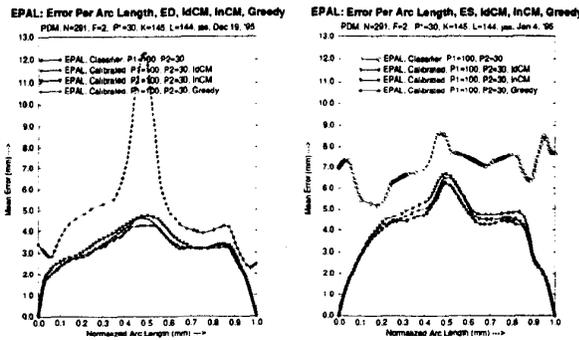


Fig. 5. Mean error per arc length (EPAL) using polyline distance method, superposition of 4 curves: the initial classifier boundary error, IdCM, InCM and Greedy calibration method. Left: ED frame Right: ES frame. Greedy does the best out of all the 3 algorithms presented for both ED and ES frames. Greedy method also does better in the apex zone (0.4-0.5), compared to all other vertices. In ED frame, apex error reduces by 8.5 mm and in ES frame, apex error reduces by 3 mm.

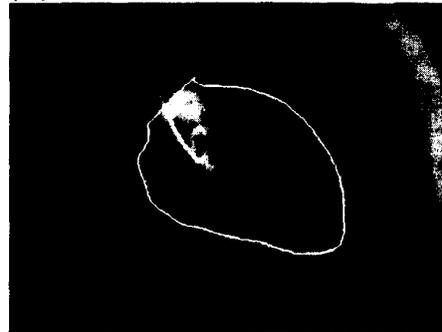
## VI. CONCLUSIONS

We presented a *Greedy calibration algorithm* for calibrating the initial *pixel-based* classifier boundaries which takes best of the two calibration methods: the *identical coefficient* and the *independent coefficient*. The mean error over ED and ES frames using a *cross-validation* protocol and *polyline distance metric* is 3.5 millimeters over the data base of 291 patient studies. The Greedy algorithm is an improvement over the *identical coefficient* method by 0.3 mm which is significant for the accuracy of the overall calibration system. The Greedy algorithm also shows the best performance in the apex zone of the LV where the dye was unable to reach, reducing the error by about 8.5 mm. We thus see that the calibration step constitutes a very signif-

icant last step for final boundary estimation.



(a1) ED Frame: GT and Classifier or Initial



(a2) ED Frame: GT and Calibrated (Estimated)

Fig. 6. Results of Greedy algorithm. Upper: (a1) Classifier (thin) vs. ground truth (GT), (thick). Bottom: (a2) Calibrated (thin) vs. ground truth (GT) (thick). Background: Left Ventriculogram (LVG), Calibration Parameters:  $N=291$ ,  $K=145$ ,  $L=144$ ,  $F=2$ ,  $P_1=100$ ,  $P_2=30$ , Mean end frame error  $\left(\frac{BD+ES}{2}\right)=1.16$  mm, Mean error  $(e_{NPF}^{poly})=3.50$ , Std. Deviation  $(\sigma_{NPF}^{poly})=3.40$  mm.

## VII. ACKNOWLEDGEMENTS

First author would like to thank Drs. Poggio (MIT), Shapiro, Zick, Lytle, Meldrum and Stuetzle for their motivation and discussions. We also thank Ms. Nan from CVRTC for tracing the LV borders.

## REFERENCES

- [1] C. K. Lee, *Automated Boundary Tracing Using Temporal Information*, PhD Thesis, Dept. of Electrical Engineering, University of Washington, Seattle, 1994.
- [2] Silvio E. Papapietro and L. Richard Smith and William P. Hood and Richard O. Russell and Charles E. Rackley and William J. Rogers, *An Optimal method for angiographic definition and quantification of regional left ventricular contraction*, Computers in Cardiology, p294-p295, September, 1978.
- [3] Florence H. Sheehan et al., *Variability in the measurement of regional left ventricular wall motion from contrast angiograms*, CIRCULATION, Ventriculography, vol. 68, no. 3, p550-p559, 1983.
- [4] U.S. Patent's: 4,952,805, 5,164,993, 5,268,967, 4,731,863, 5,028,782, 4,028,079, 4,962,539.
- [5] Jasjit S. Suri and R. M. Haralick, *LV Apex Modeling*, Tech. report #: ISL-TR-95-04, Dept. of EE, ISL, Univ. of Washington, Seattle, 1995.
- [6] Jasjit S. Suri, Robert M. Haralick and Florence H. Sheehan, *Greedy Algorithm for Error Correction in Automatically Produced Boundaries in Low Contrast Ventriculograms*, Submitted to IEEE Transactions in Medical Imaging.