

# ROBUST LEFT VENTRICLE SEGMENTATION FROM ULTRASOUND DATA USING DEEP NEURAL NETWORKS AND EFFICIENT SEARCH METHODS

Gustavo Carneiro\*, Jacinto Nascimento\*

Instituto de Sistemas e Robótica  
Instituto Superior Técnico  
Av. Rovisco Pais, 1049-001 Lisbon, Portugal

António Freitas, Ph.D.

Hospital Fernando Fonseca  
Cardiology Department  
Amadora, Portugal

## ABSTRACT

The automatic segmentation of the left ventricle of the heart in ultrasound images has been a core research topic in medical image analysis. Most of the solutions are based on low-level segmentation methods, which uses a prior model of the appearance of the left ventricle, but imaging conditions violating the assumptions present in the prior can damage their performance. Recently, pattern recognition methods have become more robust to imaging conditions by automatically building an appearance model from training images, but they present a few challenges, such as: the need of a large set of training images, robustness to imaging conditions not present in the training data, and complex search process. In this paper we handle the second problem using the recently proposed deep neural network and the third problem with efficient searching algorithms. Quantitative comparisons show that the accuracy of our approach is higher than state-of-the-art methods. The results also show that efficient search strategies reduce ten times the run-time complexity.

**Index Terms**— Segmentation of the left ventricle of the heart, deep neural networks, optimization algorithms

## 1. INTRODUCTION

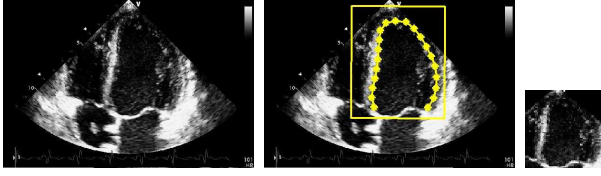
The delineation of the left ventricle (LV) of the heart in ultrasound data is an important tool to produce a quantitative assessment of the health of the heart. The automation of the LV delineation (*i.e.*, segmentation) is desirable in a clinical setting due to the following reasons: 1) it can increase patient throughput; and 2) it can reduce inter-user variation in the LV delineation procedure. However, automatic LV segmentation systems have to handle several problems present in ultrasound imaging, such as: low signal-to-noise ratio, edge dropout and shadows. The solutions proposed so far can be categorized into two classes: 1) low-level methods that use prior models of the LV appearance, and 2) pattern recognition methods based on appearance models automatically built from manually annotated LV images.

Low-level methods [1, 2, 3] consist of segmentation algorithms that use a prior model of the LV based on the assumptions that the myocardium displays brighter, and the blood pool in the LV displays darker than other structures in the image. The main problem with this approach is that the violation of these assumptions may lead to incorrect segmenta-

tions. Intensity independent features to get around this prior model have also been proposed [4, 5, 6], but the such types of model are unlikely to cover all possible imaging conditions of the LV. Pattern recognition methods involve the use of a database of annotated LV images (*i.e.*, a training set) to automatically build a model of the LV appearance [7, 8]. Even though this approach currently holds the most competitive results [9], it still faces a few challenges, such as: the need of a large training set, robustness to imaging conditions unseen in the training set, and the run-time complexity of the search process. Lately, there has been a significant effort to reduce the search complexity. For instance, the marginal space learning (MSL) [10], which partitions the search space into subspaces of increasing complexity, achieves a significant complexity reduction, but the search methods proposed by our paper are orthogonal to it, meaning that our search methods can be easily integrated into MSL. Another contribution [11] was a pattern recognition approach that, given any position in the search space, the method outputs a gradient vector that optimizes the LV segmentation function. This approach is likely to work as long as the searching region is sufficiently close to a local optimum of the objective function. In addition, the training procedure is likely to need a larger training set due to the much higher number of parameters to be learned in the gradient vector.

In this paper, we address two of the problems present in pattern recognition methods, namely: 1) robustness to imaging conditions unseen in training data, and 2) run-time complexity of the search process. In order to handle the robustness to imaging conditions, we move away from the use of boosting classifiers [8], and rely on the use of deep neural network classifiers [12] along with robust decision processes (instead of maximum a posteriori [8]). The main advantage of deep neural networks is its ability to produce more abstract feature spaces for classification and to automatically generate optimum feature spaces directly from image data. In order to tackle the complexity issue, we study the use of optimization algorithms of first and second orders [13]. The main difference compared to the work by Zhou and Comaniciu [11] is that we compute the gradient vector and Hessian matrix directly from the output of the classifiers, imposing no additional requirements for the training set. We show quantitative comparisons between our method and state-of-the-art approaches [7, 8, 9], and the results not only show a superior performance of our approach, but they also display that efficient search methods maintain the original accuracy of the method while reducing ten times the run-time complexity.

\*This work was supported by project the FCT (ISR/IST plurianual funding) through the PIDDAC Program funds.



**Fig. 1.** Original training image (left) and the manual LV delineation (center) with the rectangular patch representing the canonical coordinate system for the delineation points (markers). The right image shows the patch (extracted from the canonical coordinate system) used to train the rigid classifier.

## 2. SEGMENTATION OF THE LEFT VENTRICLE

The problem we wish to solve is to automatically produce the LV segmentation, represented by a set of points  $\mathcal{S} = \{\mathbf{s}_i\}_{i=1..N}$ , with  $\mathbf{s}_i \in \mathbb{R}^2$ , given an ultrasound image  $I$ . We assume the existence of a training set  $\mathcal{D} = \{(I, \theta, \mathcal{S})_i\}_{i=1..M}$ , with LV images  $I_i$ , the respective manual annotation  $\mathcal{S}_i$  and the parameters of a rigid transformation  $\theta_i \in \mathbb{R}^5$  (position  $\mathbf{x} \in \mathbb{R}^2$ , orientation  $\gamma \in [-\pi, \pi]$ , and scale  $\sigma \in \mathbb{R}^2$ ) that aligns rigidly the annotation points to a canonical coordinate system (see Fig.1). Our objective is to find the LV contour with the following decision function:

$$\mathcal{S}^* = \mathbb{E}[\mathcal{S}|I, y = 1, \mathcal{D}] = \int_{\mathcal{S}} \mathcal{S} p(\mathcal{S}|I, y = 1, \mathcal{D}) d\mathcal{S}, \quad (1)$$

where  $y = 1$  is a random variable indicating the presence of LV in image  $I$ . Notice that the common goal in pattern recognition methods is to find the parameter  $\mathcal{S}$  maximizing the probability function  $p(\mathcal{S}|I, y = 1, \mathcal{D})$ , but the use of expectation  $\mathbb{E}[\cdot]$  in (1) provided a more robust decision process. Eq. 1 can be expanded using

$$p(\mathcal{S}|I, y = 1, \mathcal{D}) = \int_{\theta} p(\mathcal{S}|\theta, I, y = 1, \mathcal{D}) p(\theta|I, y = 1, \mathcal{D}) d\theta. \quad (2)$$

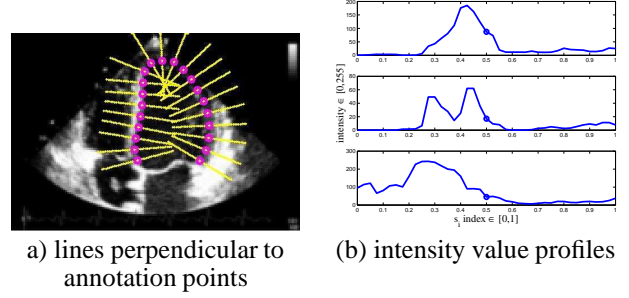
The first right-hand side (RHS) term in (2), representing the non-rigid part of the detection, is defined as follows:

$$p(\mathcal{S}|\theta, I, y = 1, \mathcal{D}) = \prod_i p(\mathbf{s}_i|\theta, I, y = 1, \mathcal{D}), \quad (3)$$

where  $p(\mathbf{s}_i|\theta, I, y = 1, \mathcal{D})$  represents the probability that the point  $\mathbf{s}_i$  is located at the LV contour. Assuming that  $\psi$  denotes the parameter vector of the classifier for the non-rigid contour, we compute

$$p(\mathbf{s}_i|\theta, I, y = 1, \mathcal{D}) = \int_{\psi} p(\mathbf{s}_i|\theta, I, y = 1, \mathcal{D}, \psi) p(\psi|\mathcal{D}) d\psi. \quad (4)$$

In practice, we run a maximum a posteriori learning procedure of the classifier parameters, which produces  $\psi_{\text{MAP}}$ , meaning that in the integral (4) we have  $p(\psi|\mathcal{D}) = \delta(\psi - \psi_{\text{MAP}})$ , where  $\delta(\cdot)$  denotes the Dirac delta function. Also, instead of computing the probability  $p(\mathbf{s}_i|\theta, I, y = 1, \mathcal{D})$ , we train a regressor that indicates the most likely edge location (see Fig.2); this roughly means that  $p(\mathbf{s}_i|\theta, I, y =$



**Fig. 2.** Intensity value profiles (from inside to outside the LV) of the lines drawn perpendicularly to annotation points.

$1, \mathcal{D}) = \delta(\mathbf{s}_i - \mathbf{s}_i^r(\theta, I, y = 1, \mathcal{D}))$ , with  $\mathbf{s}_i^r(\cdot)$  being the regressor result for the  $i^{\text{th}}$  contour point, so Eq. 2 is effectively  $\int_{\theta} \mathcal{S}^r(\theta, I, y = 1, \mathcal{D}) p(\theta|I, y = 1, \mathcal{D}) d\theta$ .

The second RHS term in (2) represents the rigid detection, which is denoted as

$$p(\theta|I, y = 1, \mathcal{D}) = \mathcal{Z} p(y = 1|\theta, I, \mathcal{D}) p(\theta|I, \mathcal{D}) \quad (5)$$

where  $\mathcal{Z}$  is a normalization constant,  $p(\theta|I, \mathcal{D})$  is a prior on the parameter space, and

$$p(y = 1|\theta, I, \mathcal{D}) = \int_{\gamma} p(y = 1|\theta, I, \mathcal{D}, \gamma) p(\gamma|\mathcal{D}) d\gamma, \quad (6)$$

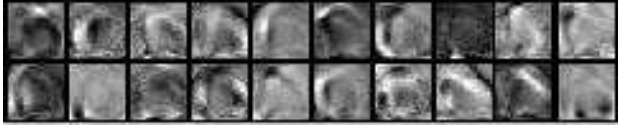
with  $\gamma$  being the vector of classifier parameters, which are estimated through a maximum a posteriori learning procedure, producing  $\gamma_{\text{MAP}}$ . This means that in (6)  $p(\gamma|\mathcal{D}) = \delta(\gamma - \gamma_{\text{MAP}})$ .

### 2.1. Deep Neural Network

The effective use of large-scale conventional neural network classifiers (with several hidden layers and thousands of nodes) is limited because backpropagation [14] (algorithm to estimate the classifier parameters) converges only when the initial guess for the parameter values are close to a local optimum of the optimization function. Hinton et al. [12] found a way to provide such initial guesses through unsupervised training of multiple layers of restricted Boltzmann machines (RBM), which are represented by a hidden and a visible layer of stochastic binary units with connections only between layers (*i.e.*, no connections within layers). After the parameters of several layers of RBMs were learned, the whole network is trained using backpropagation to adjust the weights to a local maximum for the regressor and classifier functions. For the regressor in (4), we find the solution for the maximization function  $\psi_{\text{MAP}} = \arg \max_{\psi} p(\{\mathcal{S}_i\}_{i=1..N} | \{(I, \theta)_i\}_{i=1..N}, \psi)$ , where  $(I, \theta, \mathcal{S})_i \in \mathcal{D}$ . For the classifier (6), we find the solution for  $\gamma_{\text{MAP}} = \arg \max_{\gamma} p(y = 1 | \{(I, \theta)_i\}_{i=1..N}, \gamma)$ .

### 2.2. Efficient Search Methods

For the of detection of the LV in (1), there is a five dimensional space for the rigid detection and  $N$  dimensions for the non-rigid search space, resulting in a search space of  $K^{5+N}$  samples, which is too high for most of practical values of



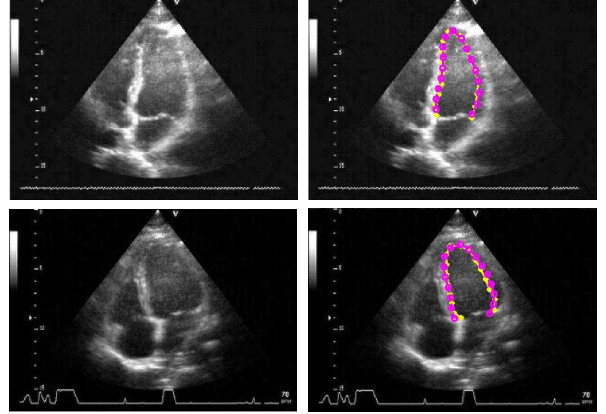
**Fig. 3.** Subset of learned features for classifier at  $\sigma = 4$ .

$K \in [10^2, 10^3]$  and  $N \in \{10, \dots, 25\}$ . Running the search procedure on the image pyramid, with one classifier per image scale, reduces the search space significantly. The advantage here is to reduce the number of samples in the coarsest scale to  $K_{\text{coarse}}$ , and move to finer scales only the best  $K_{\text{fine}} \in [10, 30]$  candidates. Note that the search procedure in fine scales needs to happen only around the current search point, meaning  $3^5$  (3 points in 5 dimensions) samples for each of the  $K_{\text{fine}}$  positions. Moreover, performing the non-rigid search only after the rigid search is done means a total search space of  $K_{\text{coarse}}^5 + (\#\text{scales} - 1) \times K_{\text{fine}} \times 3^5 + N \times K_{\text{fine}}$ .

Our first contribution to reduce the search space is to assume a prior distribution on the coarse search space, and sample  $K_{\text{coarse}}$  times from this distribution (Monte-Carlo sampling), which means a search space of  $K_{\text{coarse}} + (\#\text{scales} - 1) \times K_{\text{fine}} \times 3^5 + N \times K_{\text{fine}}$ . Our second contribution is the implementation of efficient search procedures in order to reduce the exhaustive search of  $3^5$  points around the hypotheses. We propose two methods that are widely used in optimization algorithms, which are: gradient descent and Newton step [13]. These methods work for convex functions, and their use in non-convex functions, such as the ones produced by the deep neural net classifiers, only works with a sufficiently large number of  $K_{\text{coarse}}$ . In gradient descent,  $\nabla p(y = 1|\theta, I, \mathcal{D}, \gamma_{\text{MAP}})$  is computed numerically using central difference, representing a computation of the classifier in 10 points of the search space (five parameters times two points) plus the line search in 10 points. By limiting the number of iterations between one and five for each hypothesis, the search space is then reduced to 20 to 100 points, which is smaller than  $3^5 = 243$ . In theory, a faster convergence can be achieved with the Newton step, but the computation of the Hessian matrix, gradient and line search involves 25+10 search space points. Limiting the number of iterations between one and five means that the complexity of this step for one hypothesis is between 35 to 175, which is also smaller than  $3^5 = 243$ .

### 2.3. Training and Detection Procedures

For the training procedure, we use a set of 400 ultrasound images (from 12 sequences) of left ventricles annotated by experts. For the rigid classifier, we build an image scale space  $L(\mathbf{x}, \sigma) = G(\mathbf{x}, \sigma) * I(\mathbf{x})$ , where  $G(\mathbf{x}, \sigma)$  is the Gaussian kernel,  $*$  is the convolution operator,  $I(\mathbf{x})$  is the input image,  $\sigma$  is the image scale parameter, and  $\mathbf{x}$  is the image coordinate. We train three separate classifiers (6); one for each scale  $\sigma = \{4, 8, 16\}$ . The positive and negative training sets are defined based on a scale-dependent margin  $m_\sigma$  that increases by a factor of two after each octave. Positives for  $L(\mathbf{x}, \sigma)$  are randomly generated *inside* the range  $[\theta - m_\sigma/2, \theta + m_\sigma/2]$ , and negatives are randomly generated *outside* the range  $[\theta - m_\sigma, \theta + m_\sigma]$ , where  $\theta$  is the parameter vector representing the rigid transformation of the LV annotation. Notice in Fig. 3



**Fig. 4.** Example of the first (top row) and second (bottom row) test sequences. The yellow, solid line displays the manual annotation, while the magenta dashed line shows the results from our system.

that the type of features automatically learned from this training process resembles wavelets. The non-rigid regressor is trained at  $\sigma = 4$ , where each training sample is a line of 41 pixels of length extracted perpendicularly from the LV contour points (see Fig. 2) and the label to learn is the pixel index in  $\{1, \dots, 41\}$  that is closest to the LV contour. Running a cross-validation procedure with 200 images for training and 200 images for validation, the following parameters were estimated: 1) number of nodes per layer of regressor network: 41 (visible), 50 (hidden 1), 50 (hidden 2), 250 (hidden 3), 1 (output); 2) number of nodes per layer of the classifier networks: 16, 49, 196 (visible layers at  $\sigma = \{16, 8, 4\}$ , respectively), 50 (hidden 1), 50 (hidden 2), 100 (hidden 3), 2 (output); 3) the prior distribution  $p(\theta|I, \mathcal{D})$  used in (5): uniform; 4)  $K_{\text{coarse}} = 10^3$  and  $K_{\text{fine}} = 10$ .

The detection procedure consists of running the rigid classifier at scale  $\sigma = 16$  on the  $K_{\text{coarse}}$  initial hypotheses. From this detection, cluster the hypotheses (using k-means algorithm) and select the top  $K_{\text{fine}}$  clusters in terms of the best hypothesis within each cluster. Then run the rigid classifier at scale  $\sigma = 8$  on these hypotheses and repeat the procedure for scale  $\sigma = 4$ . Finally, run the model represented by (2) over the final top  $K_{\text{fine}}$  hypotheses. Note that we substitute the integral in (1) for an average over the top hypotheses.

## 3. EXPERIMENTS

We use the following three metrics to compare the output of the detector with the reference contours, namely [9]: the Hausdorff distance, the average distance, and the Hamme distance [15]. Assuming that  $\mathcal{X} = \{\mathbf{x}_i\}_{i=1..N}$  is the automatically estimated contour from a system and  $\mathcal{S} = \{\mathbf{s}_i\}_{i=1..N}$  is the manual delineation, we first define the smallest point to curve distance as  $d(\mathbf{x}_i, \mathcal{S}) = \min_j \|\mathbf{s}_j - \mathbf{x}_i\|_2$ . The average distance between two curves is defined by:

$$d_{\text{avg}}(\mathcal{X}, \mathcal{S}) = \frac{1}{N} \sum_i d(\mathbf{x}_i, \mathcal{S}), \quad (7)$$

**Table 1.** Comparisons in the sequences (Fig. 4).

Sequence One			
Approach	Hamm. (9)	Aver. (7)	Hausd. (8)
Full	<b>0.1847</b>	<b>3.2891</b>	20.0894
GradDes	0.2060	3.6472	19.2007
Newton	0.1991	3.5580	<b>18.6611</b>
MMDA[9]	0.2472	4.8457	22.4766
COM[7, 8]	0.2083	3.8947	20.4781
Sequence Two			
Approach	Hamm. (9)	Aver. (7)	Hausd. (8)
Full	0.1777	3.0829	19.8815
GradDes	<b>0.1661</b>	<b>2.9936</b>	19.5589
Newton	0.2158	3.6345	21.1893
MMDA[9]	0.2431	4.8748	20.2606
COM[7, 8]	0.1865	3.3719	<b>17.2148</b>

and the Hausdorff distance is defined as follows [16]:

$$d_{max}(\mathcal{X}, \mathcal{S}) = \max \left( \max_i \{d(\mathbf{x}_i, \mathcal{S})\}, \max_j \{d(\mathbf{s}_j, \mathcal{X})\} \right). \quad (8)$$

Finally, the Hammoude distance [15] is defined by:

$$d_H(\mathcal{X}, \mathcal{S}) = \frac{\#((R_{\mathcal{X}} \cup R_{\mathcal{Y}}) - (R_{\mathcal{X}} \cap R_{\mathcal{Y}}))}{\#(R_{\mathcal{X}} \cup R_{\mathcal{Y}})}, \quad (9)$$

where  $R_{\mathcal{X}}$  represents the image region delimited by the contour  $\mathcal{X}$ , and similarly for  $R_{\mathcal{S}}$ .

The performance of the tracker was measured by comparing the contour estimates with reference contours provided by a cardiologist of Fernando Fonseca Hospital (Amadora, Portugal). Note that these images were not included in the 400 images of the training set. The cardiologist segmented 80 images: 40 images from two sequences (see Fig. 4). For the comparison, we present the results obtained with state-of-the-art trackers for the left ventricle recently proposed by Comaniciu et al. [7, 8] and by Nascimento [9], applied on the same data. Table 3 shows the comparisons for the two sequences with the results of our approach in rows “Full” (original search), “GradDes” (gradient descent), and “Newton” (Newton step). The rows “MMDA” and “COM” show the respective results by Nascimento [9] and Comaniciu [7, 8]. In this table the best value for each measure and sequence is highlighted.

In terms of run-time complexity, the number of floating point multiplications for the classifier at  $\sigma = 16$  is  $O(8 \times 10^6)$ , at  $\sigma = 8$  is  $O(2.5 \times 10^7)$ , at  $\sigma = 4$  is  $O(9.8 \times 10^7)$ , and the regressor is  $O(2.6 \times 10^7)$ . Given these numbers, the “Full” search average complexity is  $O(3.5 \times 10^{11})$ , while the average complexity for “GradDes” is  $O(2 \times 10^{10})$  and for “Newton” is  $O(3 \times 10^{10})$ .

#### 4. CONCLUSION AND FUTURE WORK

The pattern recognition approach for LV segmentation in ultrasound data presented in this paper shows evidence of robustness to imaging conditions absent in training data. Also,

efficient search approaches reduce the run-time complexity without affecting the accuracy of the method. Quantitative comparisons against state-of-the-art systems show the superiority of our method in publicly available datasets. We are now studying the introduction of multiple models (e.g., diastole and systole) to improve even more the robustness of the approach. Furthermore, we also plan to introduce a dynamic model to speed up the search process and get around low confidence detections.

**Acknowledgements:** The authors would like to thank G. Hinton and R. Salakhutdinov for making the deep neural network code available online.

#### 5. REFERENCES

- [1] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” in *ECCV*, 1998, pp. 484–498.
- [2] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *IJCV*, 4(1), pp. 321–331, 1987.
- [3] N. Paragios, “A level set approach for shape-driven segmentation and tracking of the left ventricle,” *IEEE TMI*, 21(9), pp. 773–776, 2003.
- [4] O. Bernard, B. Touil, A. Gelas, R. Prost, and D. Friboulet, “A RBF-based multiphase level set method for segmentation in echocardiography using the statistics of the radiofrequency signal,” in *ICIP*, 2007, pp. 157–160.
- [5] N. Lin, W. Yu, and J. Duncan, “Combinative multi-scale level set framework for echocardiographic image segmentation,” in *Medical Image Analysis*, 2003, 7(4) pp. 529–537.
- [6] A. Sarti, C. Corsi, E. Mazzini, and C. Lamberti, “Maximum likelihood segmentation of ultrasound images with Rayleigh distribution,” in *IEEE T. on Ult., Fer. and F.C.*, 2005, 52(6) pp. 947–960.
- [7] D. Comaniciu et al., “Robust real-time myocardial border tracking for echocardiography: An information fusion approach,” *IEEE TMI*, 23(7), pp. 849–860, 2004.
- [8] B. Georgescu et al., “Databased-guided segmentation of anatomical structures with complex appearance,” in *CVPR*, 2005.
- [9] J. C. Nascimento and J. S. Marques, “Robust shape tracking with multiple models in ultrasound images,” *IEEE TIP*, 17(3), pp. 392–406, 2008.
- [10] Y. Zheng et al., “Four-chamber heart modeling and automatic segmentation for 3-d cardiac ct volumes using marginal space learning and steerable features,” *IEEE TMI*, 27(11), pp. 1668–1681, 2008.
- [11] S. Zhou and D. Comaniciu, “Shape regression machine,” in *IPMI*, 2007, pp. 13–25.
- [12] R. Salakhutdinov and G. Hinton, “Learning a non-linear embedding by preserving class neighbourhood structure,” in *AI and Statistics*, 2007.
- [13] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, March 2004.
- [14] D. Rumelhart, G. Hinton, and R. Williams, “Learning representations by back-propagating errors,” *Nature*, no. 323, pp. 533–536, 1986.
- [15] A. Hammoude, *Computer-assited Endocardial Border Identification from a Sequence of Two-dimensional Echocardiographic Images*, Ph.D. thesis, University Washington, 1988.
- [16] D. Huttenlocher, G. Klanderman, and W. Rucklidge, “Comparing images using hausdorff distance,” *IEEE TPAMI*, 15(9), pp. 850–863, 1993.