ENHANCING LEFT-VENTRICULAR SHORT-AXIS ECHOCARDIOGRAPHIC CINELOOPS WITH A PULSE COUPLED NEURAL NETWORK

James Wolfer* James Robergé† Jeffrey Soble‡

ABSTRACT

Observing the motion of endocardial borders in echocardiographic cineloops is an important component for both the qualitative and quantitative analysis of regional left ventricular function. Consequently, using the computer to enhance endocardial border pixels is an important first step both for annotating and eventual computer analysis of echocardiograms. Previous work has demonstrated the viability of pulse coupled neural networks for enhancing echocardiograms acquired using an acoustical contrast agent. In this study we investigate using pulse coupled neural networks as a visualization tool, highlighting, frame-by-frame, the endocardial border in left-ventricular short-axis cineloops.

Index Terms

Computer Vision, Medicine, Health, Echocardiology, Neural Networks, AI

Introduction

The detection of potential border pixels is an important step in many computer vision applications, including the analysis of echocardiographic images. Similarly, the assessment of many diseases of the heart center around the morphology of the left ventricle over time[1]. Identifying the myocardial-blood pool border, or endocardial border, is one important step toward assessing the contractile pattern of the myocardium. Unfortunately, echocardiogram images often present challenges to computer vision systems due to low contrast and noise.

Many approaches to isolating the endocardial border have been described. Typically they include low level image processing, noise reduction, and edge detection. Examples of this processing include the popular Soble and Canny edge detectors.

Post-processing is also used to identify edges and regions. Post-processing often incorporates information to form models such as active shapes and contours, and mathematical morphology[2].

*Indiana University South Bend, South Bend, IN, USA, email: jwolfer@iusb.edu

†Cyberpulse L. L. C., Highland Park, IL, USA

‡Rush Medical Center, Chicago, IL USA, Jeffrey_S_Soble@rsh.net

Pulsed Coupled Neural Networks

Recently there has been enhanced interest in biologically inspired models of computation. These range from genetic algorithms to models of the immune system. Of these models, artificial neural networks have enjoyed the most exposure in applications. Examples in echocardiology include LV shape recovery[3] and LV contour detection[4]. These neural networks are engineering metaphors inspired by contemporary models of neural interactions, and are not models of any specific biological process. For example, the popular multi-layered perceptron encodes information based upon the average firing rate of these abstract neurons, ignoring the temporal relationships between the individual neurons.

In contrast, the Pulse Coupled Neural Network (PCNN) attempts to model neuron interactions in time. Based upon the work of Eckhorn[5, 6] modeling interactions in the visual cortex of the cat, and more recently, primates, the PCNN forms a high order network which pulses through time forming a succession of binary pulses. When the input is an image, this results in a series of binary images. Attractive aspects of the PCNN for echocardiology include relative immunity to translation, scale, and rotation in the image[7]. Imaging applications of pulse-coupled neural networks include contrast echocardiology[8], image shadow removal[9], and digital mammogram segmentation[10].

Figure 1 shows a schematic diagram of a single PCNN neuron. It is divided into three primary functions: feeding, linking, and pulse generation. This PCNN neuron is modeled by the following equations[7, 11]:

$$F_{ij}(t) = e^{-\alpha_F \delta t} F_{ij}(t-1) + S_{ij} + V_F \sum_{kl} W_{ijkl} Y_{kl}(t-1)$$

$$L_{ij}(t) = e^{-\alpha_L \delta t} L_{ij}(t-1) + V_L \sum_{kl} M_{ijkl} Y_{kl}(t-1)$$

$$U_{ij}(t) = F_{ij}(t)(1 + \beta L_{ij}(t))$$

$$Y_{ij}(t) = \begin{cases} 1 & \text{if} \quad U_{ij}(t) > \Theta_{ij}(t) \\ 0 & \text{Otherwise} \end{cases}$$

$$\Theta_{ij}(t) = e^{-\alpha_\Theta \Delta t} \Theta_{ij}(t-1) + V_\Theta Y_{ij}(t)$$

Where F is the feeding component, L the linking component, U the neuron internal activity, Y the neuron output, and Θ the dynamic threshold. M and W represent encode weights from the individual inputs in the receptive field for the feeding and linking functions respectively, and

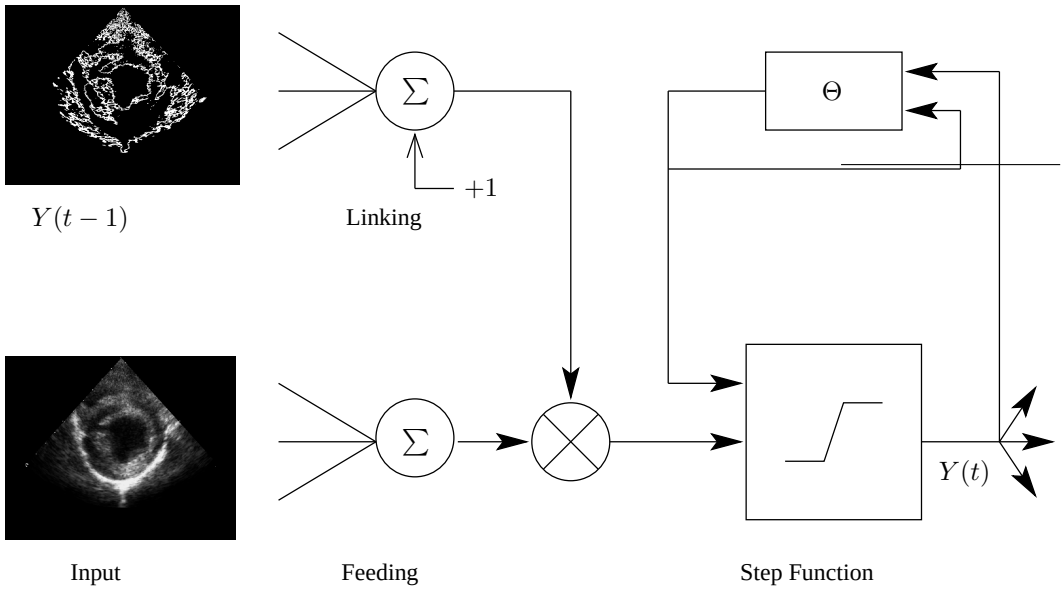


Figure 1. PCNN Schematic

β is the linking strength. These equations are applied in sequence at each iteration of the simulation.

In image processing, an individual neuron receives input to its feeding function from a single, scaled, gray level pixel in the original image along with a receptive field consisting of a weighted neighborhood. This results in one artificial neuron being directly stimulated by a corresponding pixel and its neighbors from the input image, preserving the geometric structure of the image.

Assuming that the threshold is initially set to zero, any activity at the input will cause a corresponding output from the pulse generation. This, in turn, raises the threshold suppressing subsequent output. As the threshold decays neurons with activity exceeding the threshold pulse, reestablishing a high threshold for them, but also raising the probability that adjacent neurons will be fire at the next iteration as a result of the linking feedback to the receptive field. In this sense each artificial neuron can be seen as initiating an autowave of activity which propagates until colliding with another wavefront.

Figure 2 illustrates the action of the PCNN when stimulated by the first frame in the short-axis cineloop shown in Figure 3. For this, and all subsequent images, alpha was empirically fixed at 10.0, 1.0, and 15.0 for the feeding, linking, and threshold computations respectively, and beta was 0.1.

Initially, at iteration one, every non-zero pixel causes the PCNN to fire driving the threshold high as indicated by the entirely white image. Over time the threshold decays until neurons connected to those pixels providing the highest stimulation fire, in this case at iteration seventeen. This, in turn, stimulates the surrounding pixels causing them to fire if they are close to their respective thresholds.

Application

Software to simulate the PCNN was written in Python, and then applied to the left-ventricular short-axis cineloop for a healthy canine. For each frame processed, the program reads the frame and represents the pixels as floating point values between zero and one. The resulting image forms the input to the feeding function of the PCNN.

As described, the PCNN produces a sequence of binary images representing the neuron firing pattern at each iteration. Consistent with other work[8], we include user interaction to identify the iteration at which the best fit to the data occurs. Specifically:

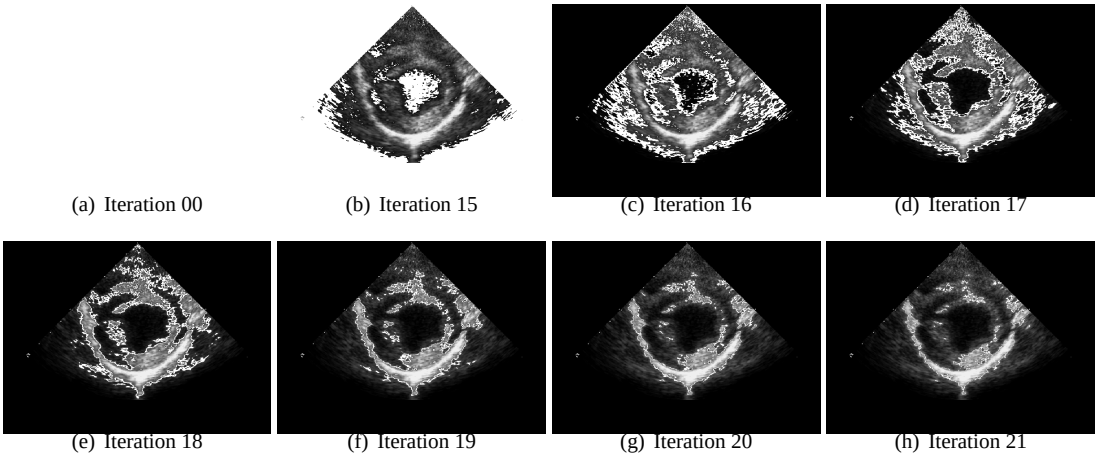
1. The user selects a prototype frame from the original cineloop.

2. The PCNN is applied to the selected frame producing a time-series of binary images representing the activity at each iteration.

3. The user selects the iteration at which the PCNN produces the best visual fit to the desired image features.

4. The PCNN is applied to all frames in the original cineloop, with activity at the selected iteration defining the image enhancement.

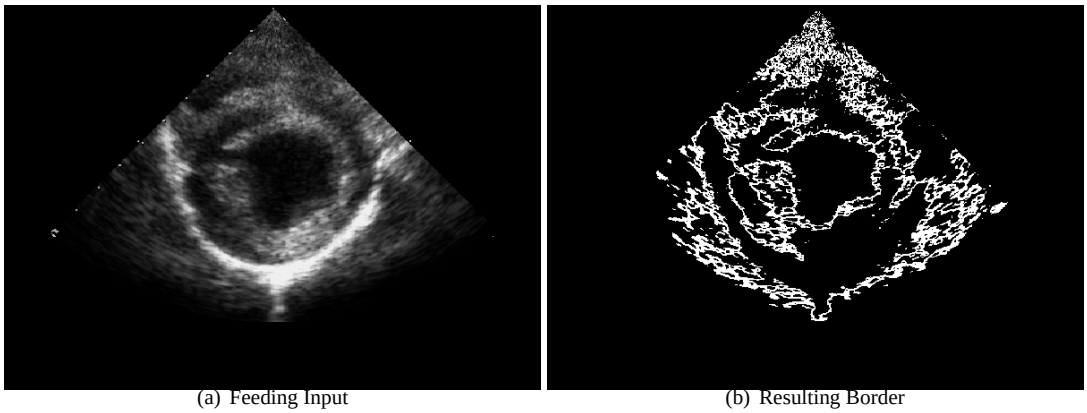
Again referring to Figure 2, we note that iteration eighteen forms the best visual fit. The PCNN is then applied to each frame in the cineloop, with its output at the selected iteration overlayed on the source frame, forming a new image sequence. Figure 4 shows a sequence of original cineloop frames and 5 shows the annotated frames. Note that the corresponding highlights, or annotations, track the



(a) Iteration 00 (b) Iteration 15 (c) Iteration 16 (d) Iteration 17

(e) Iteration 18 (f) Iteration 19 (g) Iteration 20 (h) Iteration 21

Figure 2. PCNN Iterations



(a) Feeding Input (b) Resulting Border

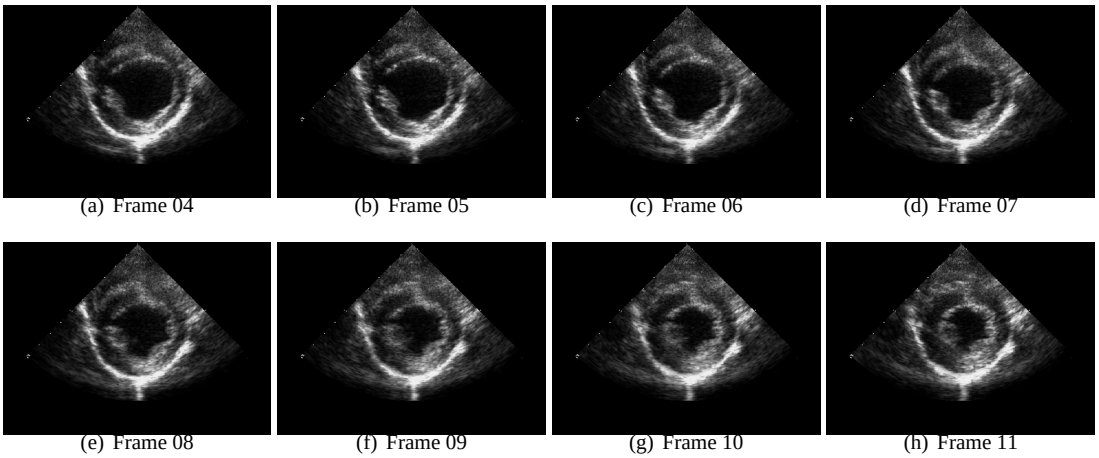
Figure 3. Sample Cineloop Frame and Corresponding PCNN Enhancement

tissue borders as the ventricular morphology changes with contraction.

To further illustrate the applicability of this approach, we demonstrate it using an ischemic cineloop sequence. Figure 6 shows a series of ischemic cineloop frames. Again the user selects the best-fit iteration for the PCNN, which is then applied to each frame of the loop. The result is illustrated in Figure 7. While not apparent in the printed images, this sequence contains many more noise artifacts, making the border-tracking more difficult. In this case additional image pre-processing, such as low-pass filtering, could be useful but we elected to work with the unprocessed cineloop to demonstrate the PCNN response in a relatively harsh environment. Note that, while there are more artifacts, the PCNN still manages to track the endocardial border of the ischemic myocardium through the contraction cycle.

Summary

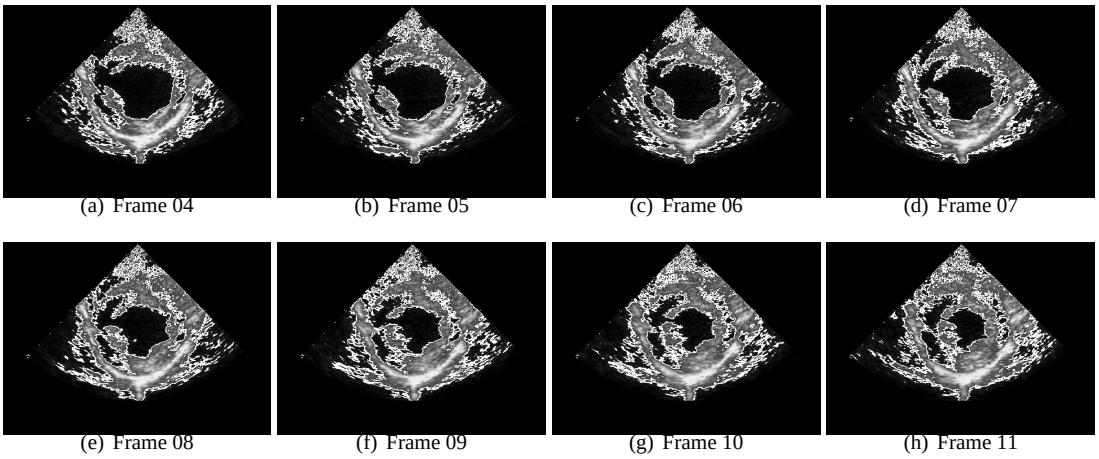
There are, in addition to its direct application, other potential applications for this cineloop enhancement. Derived images, such as PCNN border-enhanced Synthetic M-mode images[12, 13] could augment regional wall motion studies, for example. We would also like investigate its potential in a hybrid solution to border enhancement with active contours. It should be noted that the images presented here represent scan-converted gray-scale data. We would like to extend our investigation to consider the raw radio frequency data from the machine. Finally, our initial studies lead us to believe that pulse coupled neural networks hold promise for the automated enhancement of left ventricular endocardial borders. Ultimately, this could lead to computer-assisted, quantitative assessment left ventricular function in clinical practice.



(a) Frame 04 (b) Frame 05 (c) Frame 06 (d) Frame 07

(e) Frame 08 (f) Frame 09 (g) Frame 10 (h) Frame 11

Figure 4. Cineloop Frames



(a) Frame 04 (b) Frame 05 (c) Frame 06 (d) Frame 07

(e) Frame 08 (f) Frame 09 (g) Frame 10 (h) Frame 11

Figure 5. Processed Cineloop Frames

References

[1] Arthur E. Weyman. *Principles and Practice of Echocardiography*, pages 575–655. Lea and Febiger, 1994.

[2] E.R. Davies. *Machine Vision: Theory, Algorithms, Practicalities, Third Edition*. Morgan Kaufmann, 2005.

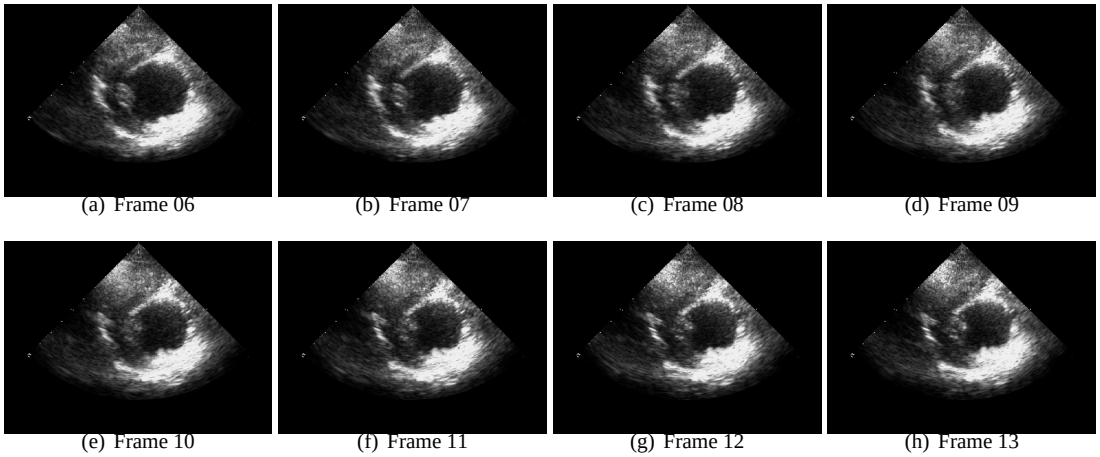
[3] R. Poli, G. Coppini, R. Nobili, and G. Valli. Lv shape recovery from echocardiographic images by means of computer vision techniques and neural networks. In *Computers in Cardiology 1991*, pages 117–120, 1991.

[4] M. Sussner, M. Budil, Th Strohmer, M. Greher, G. Porenta, and Th Binder. Contour detection using artificial neuronal network pre-segmentation. In *Computers in Cardiology 1995*, pages 737–740, 1995.

[5] Reinhard Eckhorn. Neural mechanisms of scene segmentation: Recordings from the visual cortex suggest basic circuits for linking field models. *IEEE Transactions on Neural Networks*, 10(3):464–479, May 1999.

[6] Reinhard Eckhorn, Alexander M. Gail, Andread Bruns, Andreas Gabriel, Basim Al-Shaikhli, and Mirko Saam. Different types of signal coupling in the visual cortex related to neural mechanisms of associative processing and perception. *IEEE Transactions on Neural Networks*, 15(5):1039–1052, September 2004.

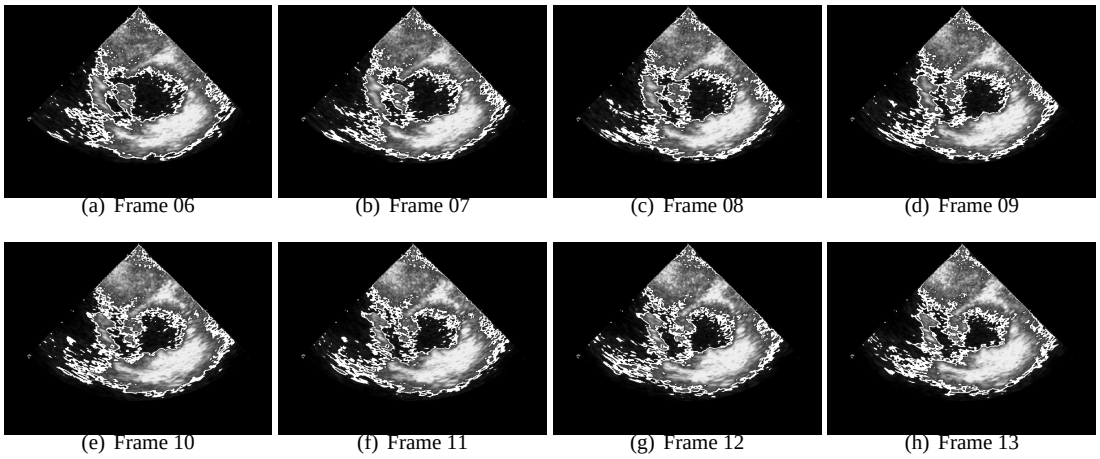
[7] Thomas Lindblad and Jason M. Kinser. *Image Processing using Pulse-Coupled Neural Networks*. Springer Verlag, 1998.



(a) Frame 06 (b) Frame 07 (c) Frame 08 (d) Frame 09

(e) Frame 10 (f) Frame 11 (g) Frame 12 (h) Frame 13

Figure 6. Ischemic Cineloop Frames



(a) Frame 06 (b) Frame 07 (c) Frame 08 (d) Frame 09

(e) Frame 10 (f) Frame 11 (g) Frame 12 (h) Frame 13

Figure 7. Processed Ischemic Frames

[8] J. Wolfer, S.H. Lee, J. Sandelski, R. Summerscales, J.S. Soble, and J. Robergé. Endocardial border detection in contrast enhanced echocardiographic cineloops using a pulse coupled neural network. In *Computers in Cardiology 1999*, 1999.

[9] Xiaodong Gu and Liming Zhang. Image shadow removal using pulse coupled neural network. *IEEE Transactions on Neural Networks*, 16(3):692–698, May 2005.

[10] Aboul Ella Hassanien and Jafar M. Ali. Digital mammogram segmentation algorithm using pulse coupled neural networks. In *Proceedings of the Third International Conference on Image and Graphics*, 2004.

[11] John L. Johnson and Mary Lou Padgett. Pcnn models and applications. *IEEE Transactions on Neural Networks*, 10(3):480–496, May 1999.

[12] J. S. Soble, I. Song, M. Bauer, A. Neumann, D. Kim, J. Robergé, and R. H. Marcus. Computer-generated synthetic m-mode: A new method for measurement of wall thickening as a parameter of regional lv systolic function. *Proceedings of the 68th Annual Scientific Session AHA*, 1995.

[13] M. Bauer, J. S. Soble, I. Song, A. Neumann, P. Liebson, J. Parrillo, J. Robergé, and R. H. Marcus. Accuracy of computer-generated synthetic m-mode for left ventricular wall thickening. *JACC*, 27(7), 1996.