Evaluation System Based on EFuNN for On-Line Training Evaluation in Virtual Reality

Ronei Marcos de Moraes[1] and Liliane dos Santos Machado[2]

[1] UFPB - Federal University of Paraíba - Department of Statistics,
Cidade Universitária s/n, 58051-900, João Pessoa, PB, Brazil
ronei@de.ufpb.br

[2] UFPB - Federal University of Paraíba - Department of Computer Sciences,
Cidade Universitária s/n, 58051-900, João Pessoa, PB, Brazil
liliane@di.ufpb.br

Abstract. In this work is proposed a new approach based on Evolving Fuzzy Neural Networks (EFuNNs) to on-line evaluation of training in virtual reality worlds. EFuNNs are dynamic connectionist feed forward networks with five layers of neurons and they are adaptive rule-based systems. Results of the technique application are provided and compared with another evaluation system based on a backpropagation trained multilayer perceptron neural network.

1 Introduction

Nowadays, with recent technological advances, several kinds of training are made in virtual reality (VR) environments. For example, military combat strategies, surgery and other critical works that involve human risks. So, very realistic VR systems have been developed with training objectives to immerge the user into a virtual world where real situations can be simulated. Simulators based on VR for training need high-end computers to provide realistic haptics, stereoscopic visualization of 3D models and textures [1]. However, it is very important to know the quality of the training and what is the trainees performance. So important as that is the existence of an on-line evaluation coupled to the system, so the trainee can evaluate himself and improve his learning. On-line evaluators must have low complexity to do not compromise simulations performance, but they must have high accuracy to do not compromise evaluation [10]. Because VR worlds are approaches of real worlds, exact measures correspondence between both worlds are not possible in VR simulators. In some applications, data collected from user's interaction cannot be adequated to classical statistical distributions [12].

Some simulators for training already have a method of evaluation. However they just compare the final result with the expected one or are videotape records post-analyzed by an expert [1]. The first models for off-line or on-line evaluation of training were proposed in the year 2000 [4,14]. Since that, statistical models as Hidden Markov Models [11,14], Fuzzy Hidden Markov Models [10], statistical distributions [9] and Fuzzy Gaussian Mixture Models [12] were proposed for training evaluation.

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 778–785, 2005.

The paper by Rosen *et al.* [13] was proposed for off-line evaluation of training applied to laparoscopic procedures performed in guinea pigs. Using an optoelectronic motion analysis and video records, McBeth *et al.* [9] acquired and compared postural and movement data from experts and residents in different contexts by the use of statistical distributions. Other papers [4,10,11] were proposed for on-line evaluation training in VR simulators. Recently, Machado and Moraes proposed the use of Neural Networks to perform that evaluation [6] to solve the problem of data not adequated to classical statistical distributions.

In 2001 Kasabov [3] proposed a new class of Fuzzy Neural Networks named Evolving Fuzzy Neural Networks (EFuNNs). EFuNNs are structures that evolve according determined principles. EFuNNs have low complexity and high acuracy and, as mentioned before, these features are important to an evaluator from training using VR. In this paper we propose an evaluation system based on EFuNNs for VR simulators and tested it using a bone marrow harvest simulator [5]. Results of the new evaluator are provided and compared with an evaluation system based on a multilayer perceptron (MLP) neural network.

2 VR Simulators and On-Line Evaluation

VR refers to real-time systems modeled by computer graphics that allow user interaction and movements with three or more degrees of freedom [1]. More than a technology, VR became a new science that joins several fields as computer sciences, engineering and cognition. VR worlds are 3D environments created by computer graphics techniques where one or more users are immersed totally or partially to interact with virtual elements. The quality of the user experience in this virtual world is given by the graphics resolution and by the use of special devices for interaction. Basically, the devices stimulate human senses as the vision, the audition and the touch (haptic devices) [1]. There are many purposes for VR systems, but a very important one is the simulation of procedures for training. In medicine, VR based training provides significant benefits over other methods, mainly in critical procedures where a mistake can result in physical or emotional impact on human beings [1].

Some VR simulators for training have a method of evaluation. However they just compare the final result with the expected one or they are videotape records post-analyzed by an expert [1]. It can be not enough to provide a good evaluation. Basically because there are medical procedures where the only sense used is the touch, as in internal exams and minimally invasive surgeries, and the intervention tool trajectory and applied forces inside the body should be known to evaluate the training. In addition, in the second case the student can have forgetten about some details of his training when the evaluation arrives. In these cases, an on-line evaluation system coupled to the VR simulator could supervise the user movements during the internal manipulation of the virtual body and provide the evaluation results to the trainee immediately at the end of the simulation [4].

The VR simulator and the on-line evaluator are independent systems, however they act simultaneously. So, user movements, applied forces, angles, position, torque and other input data can be collected from devices during the simulation to feed the evaluation system [4,13]. Depending on the application, all those variables or some of them will be monitored according to their relevance to the training. It is important to remember that virtual reality based simulators are real time systems. So, the evaluation method requires special attention to do not compromise the simulator performance.

3 Evolving Fuzzy Neural Networks (EFuNNs)

As mentioned before, Evolving Fuzzy Neural Networks (EFuNNs) are structures that evolve according ECOS principles [3]: quick learning, open structure for new features and new knowledge, representing space and time and analyse itself of errors. The EFuNN is a connectionist feed forward network with five layers of neurons, but nodes and connections are created or connected when data examples are presented[3]. The input layer represents input variable of the network as crisp value x. The second layer represents fuzzy quantization of inputs variables. Here, each neuron implements a fuzzy set and its membership function as triangular membership, gaussian membership or other. The third layer contains rule nodes (r_j) which evolve through training. Each one is defined by two connections vectors: $W_1(r_j)$ from fuzzy input layer to rule nodes and $W_2(r_j)$ from rule nodes to fuzzy output layer. These nodes are created during network learning and they represent prototypes of data mapping from fuzzy input to fuzzy output space. In this layer we can use a linear activation function or a Gaussian function. The fourth layer represents fuzzy quantization of the output variables from a function of inputs and from an activation function. The last layer use an activation function to calculate defuzzified values for output variables y.

In the third layer, each $W_1(r_j)$ represents the coordinates of the center of a hypersphere in the fuzzy input space and each $W_2(r_j)$ represents the coordinates of the center of a hypersphere in the fuzzy output space. The radius of the hypersphere of a rule node r_j is defined as $R_j = 1 - S_j$, where S_j is the sensitive threshold parameter for activation of r_j from a new example (x, y). The pair of fuzzy data (x_f, y_f) will be allocated to r_j if x_f is into the r_j input hypersphere and if y_f is into the r_j output hypersphere. For this, two conditions must be satisfied:

a) The local normalized fuzzy distance between x_f and $W_1(r_j)$ must be smaller than R_j. The local normalized fuzzy distance between these two fuzzy membership vectors is done by:

$$D(x_f, W_1(r_j)) = \|x_f - W_1(r_j)\| \,/\, \|x_f + W_1(r_j)\| \tag{1}$$

where $\|a - b\|$ and $\|a + b\|$ are the sum of all the absolute values of a vector that is obtained after vector subtraction $a - b$ or summation $a + b$ respectively.

b) The normalized output error $Err = \|y - y'\| \,/N_{out}$ must be smaller than an error threshold E, where y is as defined before, y' is produced by EFuNN output, N_{out} is the number of outpus and E is the error tolerance of the system for fuzzy output.

If the condictions (a) or (b) are not satisfied, it can be created a new rule node. The weights of rule r_j are updated according to an interactive process:

$$\begin{aligned} W_1(r_j^{(t+1)}) &= W_1(r_j^{(t)}) + l_{j,1}\left(W_1(r_j^{(t)}) - x_f\right) \\ W_2(r_j^{(t+1)}) &= W_2(r_j^{(t)}) + l_{j,2}(A_2 - y_f)A_1(r_j^{(t)}) \end{aligned} \tag{2}$$

where $l_{j,1}$ is the learning rate for the first layer and $l_{j,2}$ is the learning rate for the second layer. In general, it can be assumed they have the same value done by: $l_j = 1/N_{ex}(r_j)$, where $N_{ex}(r_j)$ is the number of examples associated with rule node r_j.

$$A_1(r_j^{(t)}) = f_1(D(W_1(r_j^{(t)}), x_f) \tag{3}$$

is the activation function of the rule $r_j^{(t)}$ and

$$A_2 = f_2(W_2A_1) \tag{4}$$

is the activation of the fuzzy output neurons, when x is presented. For the functions f_1 and f_2 can be used a simple linear function.

When a new example is associated with a rule r_j, the parameters R_j and S_j are changed:

$$\begin{aligned} R_j^{(t+1)} &= R_j^{(t)} + D\left(W_1(r_j^{(t+1)}), W_1(r_j^{(t)})\right) \\ S_j^{(t+1)} &= S_j^{(t)} - D\left(W_1(r_j^{(t+1)}), W_1(r_j^{(t)})\right) \end{aligned} \tag{5}$$

If exists temporal dependencies between consecutive data, the connection weight W_3 can capture that. The connection W_3 works as a *Short-Term Memory* and a feedback connection from rule nodes layer. If the winning rule node at time $(t-1)$ was $r_{max}^{(t-1)}$ and at time (t) was $r_{max}^{(t)}$, then a link between the two nodes is established by:

$$W_3\left(r_{\max}^{(t-1)}, r_{\max}^{(t)}\right) = W_3\left(r_{\max}^{(t-1)}, r_{\max}^{(t)}\right) + l_3A_1\left(r_{\max}^{(t-1)}\right)A_1\left(r_{\max}^{(t)}\right) \tag{6}$$

where $A_1\left(r_{\max}^{(t)}\right)$ denotes the activation of a rule node r at a time (t) and l_3 defines a learning rate. If $l_3 = 0$, no temporal associations are learned in an EFuNN.

The EFuNN learning algorithm starts with initial values for parameters [3]. According to mentioned above, the EFuNN is trained by examples until convergence. When a new data example $d = (x, y)$ is presented, the EFuNN either creates a new rule r_n to memorize the new data (input vector $W_1(r_n) = x$ and output vector $W_2(r_n) = y$) or adjusts the winning rule node r_j [3].

4 The Evaluation System

The new methodology proposed in this work was applied in training evaluation over a bone marrow harvest simulator based on VR [4]. The bone marrow is a tissue found inside the bones and used for transplant. Its harvest is a medical procedure performed without any visual feedback except the external view of the donor body. Basically, the physician needs to feel the skin and tissue layers trespassed by the needle to find the bone marrow and then start the material aspiration. Each layer has specific properties as density and elasticity. The bone marrow harvest simulator uses a robotic arm, that offers six degrees of freedom movements and force feedback in the x, y and z axis, to simulate the needle used in a real procedure [5]. So, the goal of the bone marrow simulator is to train the needle insertion stage. The system presents the pelvic region and the robotic arm. The Figure 1 presents the simulator and the layers trespassed by the needle during the bone marrow harvest.

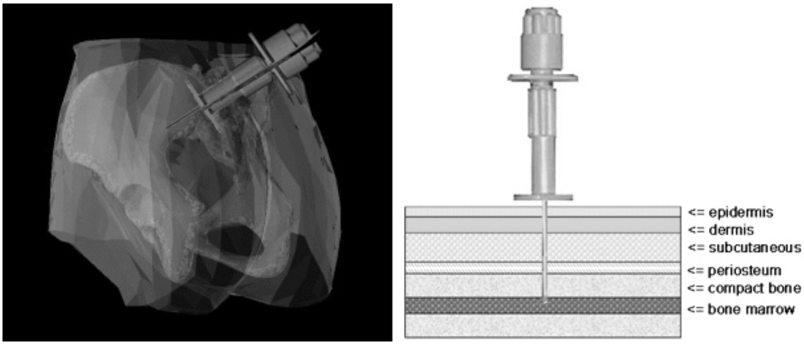


Fig. 1. The Bone Marrow Harvest simulator based on VR and the pelvic tissue layers of human body

An evaluator of performance based on EFuNN and coupled to the bone marrow harvest simulator was implemented. For reasons of general performance of the VR simulator were monitored the following variables: spatial position, velocities, forces and time on each layer. At first moment, the system was calibrated by an expert, according K classes of performance defined by expert. The number of classes of performance was defined as $K = 4$: 1) correct procedures, 2) aceptable procedures, 3) bad executed procedures and 4) very bad executed procedures. So, the classes of performance for a trainee could be: ”you are well qualified”, ”you need some training yet”, ”you need more training” and ”you are a novice”. When a trainee uses the system, his performance is compared with each expert classes of performaces and the EFuNN assigns the better class according the trainee’s performance. At the end of training, the evaluation system reports to trainee his classification.

To the calibration of EfuNN based training evaluator, an expert executed the procedure approximately a hundred times. The information about performance was acquired using an Evolving Fuzzy Neural Networks and using activation functions done by (3) and (4) for each class. For a controlled and impartial analysis, the procedure was executed hundreds of times by several users. After that, the data collected from these trainings were manually rotuled according to the expert specifications. For each class of performance were selected two hundred cases. These cases were used to validate the evaluation system based on EFuNN. The percentual of correct classification obtained was 98.625% and the Mean Square Error was 0.017956, with 11 misclassifications.

From these data, it was generate the classification matrix showed in the Figure 2. The diagonal of that matrix shows the correct classification. In the other cells, we can observe the mistakes of classifications.

200	0	0	0
0	200	0	0
1	1	189	9
0	0	0	200

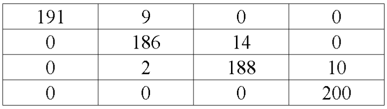
Fig. 2. Classification matrix performed by EFuNN based evaluator

It was used the Kappa Coefficient [2] to perform the comparison of the classification agreement. From the classification matrix obtained, the Kappa coefficient for all samples was $K = 98.1667\%$ with variance $\sigma_K^2 = 3.012 \times 10^{-5}$. The Kappa coefficients for each class of performance were: for class 1, $K_1 = 100.000\%$; for class 2, $K_2 = 100.000\%$; for class 3, $K_3 = 92.799\%$ and for class 4, $K_4 = 100.000\%$. That performance is very acceptable and shows the good adaptation of EFuNN in the solution of evaluation problem.

Another important result is the computational performance of the evaluator system: as EFuNN has low computational complexity, other variables could be monitored without degradation of the performance to the virtual reality simulation.

5 Comparison with a MLP Neural Network

A comparison was performed with a Backpropagation trained MLP Neural Network as those proposed by Moraes and Machado [6]. The MLP Neural Network was configured and calibrated by the expert for the same four classes used before. The same eight hundred samples of training (two hundred of each class of performance) were used for a controled and impartial comparison between the two evaluation systems. In this case, after several tests, the better choice was a MLP with four layers with 9, 7, 4, 1 neurons respectively. Nine neurons in the input layer, seven and four in the hidden layers and one in the output layer. The percentual of correct classification obtained was 95.625% and the Mean Square Error was 0.042656, with 35 misclassifications.



191	9	0	0
0	186	14	0
0	2	188	10
0	0	0	200

Fig. 3. Classification matrix performed by MLP Neural Network based evaluator

From the classification matrix obtained (presented in the Figure 3), the Kappa coefficient for all samples was $K = 94.1667\%$ with variance $\sigma_K^2 = 9.293 \times 10^{-5}$. The Kappa coefficients for each class of performance were: for class 1, $K_1 = 94.089\%$; for class 2, $K_2 = 90.713\%$; for class 3, $K_3 = 91.973\%$ and for class 4, $K_4 = 100.000\%$. That performance is good and shows that MLP Neural Network is a competitive approach in the solution of evaluation problem.

We could observe few mistakes in classification performed by MLP Neural Network based evaluator. However, it can see by Figures 2 and 3 and by other information (Kappa coefficients and Mean Square Errors) that the performance of evaluator based on MLP Neural Network is lower than the one of the evaluator based on EFuNN.

About computational performance of evaluator system, some MLP Neural Network tested with 5 or more layers caused performance problems to the VR simulation. However, those MLP neural nets were not the nets with better performance for this task.

6 Conclusions and Further Works

In this paper we presented a new approach to on-line training evaluation in virtual reality simulators. This approach uses an evaluator based on EFuNN and solves the main problems in evaluation procedures. Systems based on this approach can be applied in virtual reality simulators for several areas and can be used to classify the trainee into classes of learning giving him a real position about his performance.

A bone marrow harvest simulator based on virtual reality was implemented to serve as base of the performance tests. The performance obtained by evaluation system based on EFuNN was compared with a backpropagation trained multi-layer perceptron neural network. Based on the obtained data, it is possible to conclude that the evaluation system based on EFuNN presented a superior performance when compared with an evaluation system based on MLP Neural Network for the same case.

By their qualities, this approach could be used for Web-based simulation evaluation also, using plug-ins or agents to collect information about the different variables of user's simulations. In the future, evaluation systems like this can help training in telemedicine.

References

1. Burdea, G. and Coiffet, P., Virtual Reality Technoloy. 2nd ed. New Jersey, Addison-Wesley, 2003.
2. Cohen, J., A coefficient of agreement for nominal scales. Educational and Psychological Measurement, v.20, p.37-46, 1960.
3. Kasabov, N., Evolving Fuzzy Neural Network for Supervised/Unsupervised On-line, Knowledge-based Learning, IEEE Trans. on Man, Machine and Cybernetics, v.31, n.6, 2001.
4. Machado, L. S., Moraes, R. M. and Zuffo, M. K., Fuzzy Rule-Based Evaluation for a Haptic and Stereo Simulator for Bone Marrow Harvest for Transplant. 5th Phantom Users Group Workshop Proceedings, 2000.
5. Machado, L. S., Mello, A. N., Lopes, R. D., Odone Fillho, V. and Zuffo, M. K., A Virtual Reality Simulator for Bone Marrow Harvest for Pediatric Transplant. Studies in Health Technology and Informatics - Medicine Meets Virtual Reality, vol. 81, p.293-297, January, 2001.
6. Machado, L. S.; Moraes, R. M., Neural Networks for on-line Training Evaluation in Virtual Reality Simulators. Proc. of World Congress on Engineering and Technology Education, Brazil, p. 157-160, 2004.
7. Mahoney, D.P. The Power of Touch. Computer Graphics World, v.20, n. 8, p. 41-48, August, 1997.
8. Massie, T., Salisbury, K., The PHANToM Haptic interface: A device for probing virtual objects. ASME Winter Annual Meeting , DSC. v. 55-1. p. 295-300. 1994
9. McBeth, P. B. et al., Quantitative Methodology of Evaluating Surgeon Performance in Laparoscopic Surgery. Studies in Health Technology and Informatics: Medicine Meets Virt. Real., v 85, p. 280-286, Jan., 2002.
10. Moraes, R. M., Machado, L. S., Fuzzy Hidden Markov Models for on-line Training Evaluation in Virtual Reality Simulators. In Computational Intelligent Systems for Applied Research. World Scientific, Singapore, p. 296-303, 2002.
11. Moraes, R. M.; Machado, L. S., Hidden Markov Models for Learning Evaluation in Virtual Reality Simulators. International Journal of Computers & Applications, v.25, n.3, p. 212-215, 2003.
12. Moraes, R. M.; Machado, L. S., Fuzzy Gaussian Mixture Models for on-line Training Evaluation in Virtual Reality Simulators. Anals of the International Conference on Fuzzy Information Processing (FIP'2003). March, Beijing. v. 2, p. 733-740, 2003
13. Rosen J., Richards, C., Hannaford, B. and Sinanan, M., Hidden Markov Models of Minimally Invasive Surgery, Studies in Health Technology and Informatics - Medicine Meets Virtual Reality, vol. 70, p. 279-285, January, 2000.
14. Rosen, J., Solazzo, M., Hannaford, B. and Sinanan, M., Objective Laparoscopic Skills Assessments of Surgical Residents Using Hidden Markov Models Based on Haptic Information and Tool/Tissue Interactions. Studies in Health Technology and Informatics - Medicine Meets Virtual Reality, vol. 81, p. 417-423, January, 2001.