

CybAssess: A Class for Online Assessment of Users in a Virtual Reality System

Ícaro L. L. Cunha¹, Liliane S. Machado¹ and Ronei M. Moraes²

¹Departament of Informatics and ²Departament of Statistics
Universidade Federal da Paraíba
Cidade Universitária s/n – 58051-900 – João Pessoa/PB – Brazil

ivellius@yahoo.com.br, ronei@de.ufpb.br, liliane@di.ufpb.br

ABSTRACT

Virtual Reality systems (VRS) allow the creation of immersive environments for realistic simulations of real life situations. Though this characteristic is quite important for the conception of medical applications, the addition of assessment tools make these applications more useful and efficient when used for training purposes. An assessment can either be performed on-line or offline depending on how it is attached to the VRS. This work presents the conception and development of an on-line assessment class for training based on VRS. Results of performance and accuracy tests are also presented for a case study.

Keywords

Simulation Assessment, Virtual Reality Systems, On-line Assessment, Medical Training.

1. INTRODUCTION

In medical learning, surgery simulation is an essential tool for acquiring experience and skills. It can improve patient recovering time and reduce morbidity and mortality indexes related to surgeries. Simulations based on virtual reality system (VRS) can offer an immersive environment that can be visually and physically (by means of specific interaction devices) realistic. In order to reach these characteristics, VRS depend on the quality of images in the environment, on the non-existence of delays during simulations and on interaction modes provided by the system [2]. A VRS training application can incorporate many features: threedimensional visualization, use of realistic models, 3D interaction, visual sharing and supervision/assessment of user interaction [1,13]. This work focuses on the assessment feature of such training simulators based on VRS.

Assessing user interaction during a simulation is important to know user's performance during the simulation and report if, where and when he/she did something wrong during the simulation. From the user interaction, the system can obtain interaction data and compare it with pre-acquired data from an expert. Assessment functionalities can either be made on-line or off-line. The difference between them is that an on-line tool continuously receives data (assessment parameters) from the system and the assessment is made in real-time (RT). An off-line assessment tool (even though it is part of the system) does not receive data from the system during the system operation and the

assessment result is not given in RT. An on-line assessment is important because the loss of interaction data is prevented and makes the assessment tool more efficient to improve user's performance [13, 14].

The first methods proposed for user assessment used to record the user performance in videotapes to a post-analysis by experts [3]. After some time, the user received his/her assessment result. This can be a problem mainly because after some hours the user will probably not remember his/her exact actions when he/she was performing the simulation. Later, other methods for off-line [12, 16, 17] and on-line [9, 11, 13, 14, 15, 18] assessment were proposed.

In this work, it is presented the computational development for a flexible on-line assessment class, called CybAssess, which can be used in different kinds of procedure simulations. This class is part of a system called CyberMed [5], dedicated to the development of VR simulators for medicine. The assessment methodology implemented in CybAssess is based on [13].

2. ASSESSMENT IN VRS BASED SIMULATORS

VRS are able to create interactive environments to represent artificial scenarios whose exploration would not be possible in the real world. At the same time, they can provide scenarios for training and education where traditional assessment and educational materials may fail or become obsolete [7]. Using a simulator, a medicine student can practice in a realistic virtual environment to learn the necessary skills to perform a procedure in the real world.

For a training system to become efficient, it must present a good level of realism. This realism depends on some factors such as RT, visual realism and interaction modes. RT is necessary to provide visual responses to user's interactions, for example. In this case, a slow feedback could decrease the level of realism of the system. Visual realism is important because the user must recognize anatomical structures and textures. Finally, the interaction modes are important to provide the sense of touch and allow some degree of freedom in user actions.

A very useful feature in a training application is the assessment of the user. In Pedagogical terms, if the user knows his/her

performance, then he/she can improve his/her learning. Assessment methods can be classified as off-line or on-line. In the first one of these methods the information about user's interaction during training can be recorded and post-analyzed. Some approaches just compare the final result with the expected one or are videotape records post-analyzed by an expert [3]. Recently, some models for off-line assessment of training have been proposed using Discrete Hidden Markov Models [16] or Continuous Hidden Markov Models (CHMM) [17] to modeling forces and torque during a simulated training in a porcine model. Using an optoelectronic motion analysis and video records, McBeth et al. [12] acquired and compared postural and movement data of experts and residents in different contexts by the use of statistical distributions.

In on-line methods, the analysis is performed during the user training and the assessment result is provided at the end of the simulation. Machado et al. [11] proposed the use of a fuzzy rule-based system to on-line assessment of training in virtual worlds. Machado and Moraes proposed also the use of Maximum Likelihood method [13]. They proposed an assessment method with two-stage using Gaussian Mixture Models and Relaxation Labeling [15] to provide an on-line assessment for simulators or training systems based on virtual reality. Besides, it was proposed an assessment method for training performed in different stages, using a set of different fuzzy rule-based system [9]. Zhang et al. [18] used a set of descriptive statistics to show deviations of perfect cut in laparoscopic surgery. The training session could be recorded for posterior analysis. Thus, this method could be also used for off-line assessment.

For users it is more convenient to know his/her performance immediately after the end of training. However, there are two essentials pre-requirements for these methods: the assessment method must be fast so that it does not compromise the simulation performance and it must be accurate so it does not compromise the assessment itself. In the next section, it is presented an assessment methodology which obeys these pre-requisites.

3. MAXIMUM LIKELIHOOD METHOD

This section presents the method for training assessment, based on Maximum Likelihood. Maximum likelihood decision rule is the most common statistical technique used in data classification. The Maximum Likelihood Method (MLM) is a supervised method, i.e. it needs previous information to perform an assessment. That previous information is used to construct a statistical knowledge about the problem and it is stored in parameters inside a statistical model. As in data classification, n classes must be predefined and it is necessary to collect information about each one of them to construct its particular knowledge. This process is called calibration of model.

Formally, let $C = (C_1, C_2, \dots, C_n)$ be a set of classes of performance, where n is the number of classes of performance. To determine the most probable class of performance for a vector of training data x , by conditional probabilities [8]:

$$P(C_i|X), \text{ where } i = 1, 2, \dots, n \quad (1)$$

The probability done by (1) gives the likelihood that for a data vector X , the correct class is w_i . The classification rule is performed according to:

$$X \in C_i \text{ if } P(C_i|X) > P(C_j|X) \text{ for all } i \neq j. \quad (2)$$

However, all the probabilities done by (1) are unknown. So, if we have sufficient information available for each class of performance, we can estimate that probabilities, denoted by $P(X|C_i)$. Using the Bayes Theorem:

$$P(C_i|X) = [P(X|C_i) P(C_i)] / P(X), \quad (3)$$

where
$$P(X) = \sum_{i=1}^n P(X|C_i) P(C_i).$$

As $P(X)$ is the same for all classes C_i , then it is not relevant for data classification. In Bayesian theory, $P(C_i)$ is called a priori probability for C_i and $P(C_i|X)$ is a posteriori probability for C_i where X is known. Then, the classification rule done by (2) is modified:

$$X \in C_i \text{ if } P(C_i|X) P(C_i) > P(C_j|X) P(C_j) \text{ for all } i \neq j. \quad (4)$$

The equation (4) is known as Maximum Likelihood decision rule or Maximum Likelihood classification. However, it can be convenient to use [8]:

$$\begin{aligned} g(X) &= \ln [P(X|C_i) P(C_i)] \\ &= \ln [P(X|C_i)] + \ln [P(C_i)] \end{aligned} \quad (5)$$

where $g(X)$ is known as log-likelihood function and it is known as discriminant function.

It is important to note that if statistical distribution of training data can assume multivariate Gaussian distribution, the use of (5) has interesting computational properties [6]. If training data cannot assume that distribution, the equation (5) can provides a significant reduction of computational cost of implementation.

The equation (5) can be used to modify the formulation done by (4):

$$X \in C_i \text{ if } g_i(X) > g_j(X) \text{ for all } i \neq j. \quad (6)$$

It is important to note that if statistical distribution of training data can assume multivariate Gaussian distribution, the use of (5) has interesting computational properties [8]. If training data cannot assume that distribution, the equation (5) can provide a significant reduction of computational cost of implementation.

4. CYBASSESS

The CybAssess is a class dedicated to provide user assessment in the CyberMed system. The CyberMed is a VRS based on free software which offers a set of classes for developing simulation applications for medical teaching and training. Its main functionalities are: three-dimensional visualization, use of realistic models, haptics support, spatial interaction, visual sharing, and supervision/assessment of user interaction [5]. An important feature of the system is that it is implemented based on object oriented programming paradigm and each one of its functionalities were created as one or more independent classes. This is important to make each class flexible. The flexibility means that different kinds of applications can be created just setting: kinds of parameters to be assessed, types of interaction devices, types of

collision detection algorithms, types of visualization (monoscopic, spectral multiplexing, polarization multiplexing and temporal multiplexing), among other options. All these functionalities can be used together or independently since the CyberMed guarantees total synchronization of the routines.

Due to the fact that the assessment should be performed on-line, the integration of the CybAssess to the CyberMed had to guarantee direct access to the application assessment parameters. Figure 1 presents an example of a direct integration of the CybAssess to the CyberMed. The Interator is responsible for obtaining values about the position and interaction variables collected from the interaction device in the simulation environment. This data is collected either from a Haptic Device as from a Mouse Interaction. The Environment Parameters utility is responsible to provide other data that might be useful for the assessment, as geometric transformation parameters or ambient light intensity, for examples.

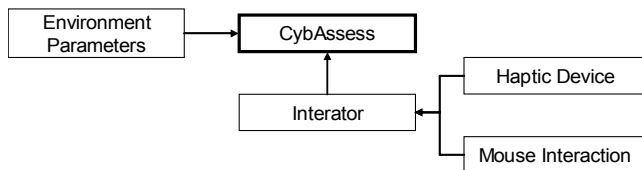


Figure 1. Example of the integration of an assessment tool to VRS utilities.

4.1 Implementation

The CybAssess class was implemented using the programming language C++ (so did the CyberMed system [5]). Figure 2 shows its composition. In fact, the CybAssess hides the implementation of several classes used to support assessment methods and their functions. In a depth view, new methods can be added to the CybAssess by its inheritance and the implementation of the functions needed to a specific method. This implementation will not only allow the integration of an assessment method into the CybAssess, but also to the CyberMed and all its functionalities. To include a new assessment method, a programmer can inherit these features and implement them as he/she wants.

An abstract class CybAssess was developed to store the assessment parameters and other characteristics required by the type of assessment performed in the application. This class supports all other classes by capturing and making available the user interaction data.

A class CybAssessment inherits the characteristics of the abstract class CybAssess and implements them to integrate the method chosen to the CyberMed system. Basically, the CybAssessment provides a façade between the CybAssess and the complete system, at the same time it implements the features necessary to integrate a method to the system. Figure 2 shows how two assessment operations classes can be integrated to the CybAssess.

The CybMLM was designed to be integrated to the CybAssess and implements the MLM for user assessment during training. The CybMLM has two main routines: a calibration stage and an assessment stage.

The CybAssess can use coordinates, geometrical parameters and other information from the calibration stage to obtain values and other statistical characteristics of the classes. With this data (called assessment parameters), the user’s assessment for a simulation can be executed.

4.1.1 Calibration and Assessment Stages

The CybMLM class is composed by two main routines: a calibration routine and an assessment routine. For these routines the class stores two lists: the class of performance list and the interaction data list for storing the necessary data. The class of performance list stores information about the levels of possible performances obtained during the calibration stage of the assess. Examples of data stored in this list are: an nxn covariance matrix (where n is the class dimension); the determinant and inverse of this matrix; the mean vector (of size n) of the data of the class; and the class id. Some data, as the determinant and inverse of matrixes, are stored in a file. It prevents the class from executing calculations during simulation. This feature provides an important economy of processing time. The interaction data list stores the data which will be used either by the calibration stage (for obtaining the class data given by an expert) or by the assessment stage (in this case, obtained from user interaction that will have his performance evaluated). Among the data stored in this list there is a data vector of size nxm (where the size of m depends on the amount of interaction and the rate of storage of the data).

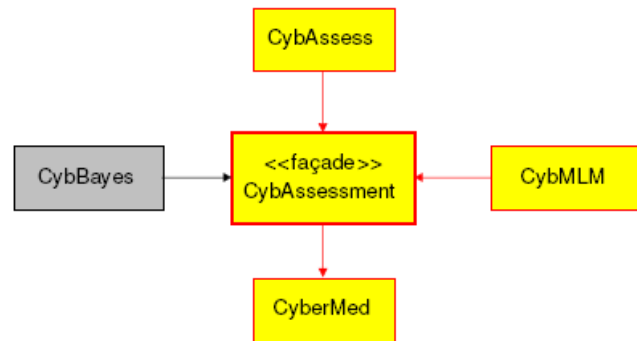


Figure 2. Composition of the CybAssess class set.

The file which stores the data of each class of performance is divided in fields of the following format:

```

#ClassID ...
#MeanVector {
...
}
#CovarianceMatrix {
...
}
#CovarianceMatrixDeterminant {
...
}
#CovarianceMatrixInverse {
...
}

```

Each field in the file belongs to one class, composed by subfields that store data obtained during the calibration stage. An expert executes the calibration stage of the assessment class and indicates the information about the performance classes. This calibration is necessary to provide the data used to compose discriminate functions, according to (5), that will classify it during the simulation. During this stage, each class will indicate a type of user performance during the simulation. At least three calibration executions for each class are necessary in the calibration stage. During each calibration execution, the assessment class will create the list of data obtained from expert's interaction. From this list, the performance class list is created and stored on a file used during the assessment stage.

The assessment is executed during the application execution. In this case, the system reads and stores the information on the performance class file and obtains the user's interaction data during the simulation. When the simulation is finished, the assessment class runs to provide the result about user's performance.

5. PERFORMANCE TESTS

In order to test the assessment class performance, a series of simulations (with performance classes of different dimensions) were made. These simulations were based on the scenario of a bone marrow harvest procedure simulation.

The main goal of these simulation tests was to know the accuracy and the assessment capacity of the CybMLM class to perform the assessment stage in real-time. Two sets of data samples were created for each test: a set for a calibration stage and a set for an assessment stage. For each set, 5 classes were defined with their respective variables. The classes represent each category of users' performance and the variables, in this case, represent the interaction data to be analyzed. For each variable, a vector of size of 30000 samples was randomly generated from a Gaussian probability distribution. For these randomly generated variables, the order of intersection between the probabilities distributions for each variable in each dimension is around 30%. This amount of intersection can be considered the worst case, in which the smallest efficiency of the assessment system is waited. From these vectors a total of 10000 were used on the calibration stage, and the other 20000 samples were used during the assessment stage.

5.1 Case Study: Bone Marrow Harvest Procedure

The case study proposed is used to show an example of the characteristics of the CybAssess in the assessment of a bone marrow harvest procedure simulation. A bone marrow harvest simulator was developed to train new doctors for the harvest stage of the bone marrow transplant procedure [10]. The simulation provides an execution environment without visual feedback, except for the external view of the donor's body. In that system, the physician needs to feel the skin and bone layers trespassed by a needle to find the bone marrow and then start the material aspiration (Figure 3) [10]. The simulator uses a haptic device, that operates movements with six degrees of freedom and provides force feedback, to give to the user the tactile sensations felt during the penetration of the patient's body. The device simulates the needle used in the real procedure and the virtual body has tactile

properties similar to the real tissues. The CybAssess with the CybMLM should supervise the user movements during all the simulation (approximately 20 seconds).

To analyze the user's performance, a series of parameters are necessary:

- Interaction object position (3D coordinates), obtained from the haptic device (refers to the tip of the needle);
- Angle of interaction relative to the object surface; this is important to analyze a mistake (shown in Figure 4) where the user could scrape the periosteum surface while trying to make the puncture movement;
- Torque applied when user try to penetrate the periosteum;
- Total time of the simulation;

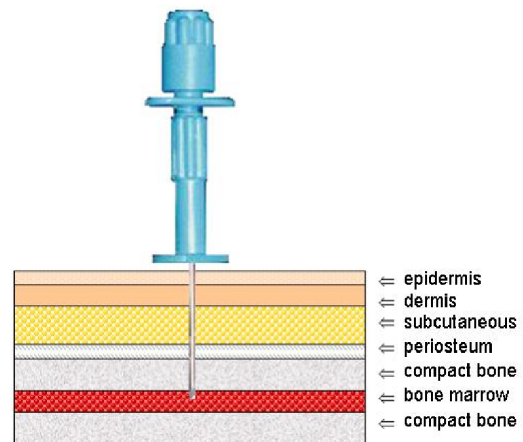


Figure 3. The Tissue Layers Trespassed by Needle in a Bone Marrow Harvest

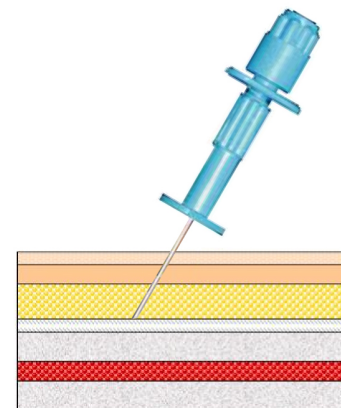


Figure 4. Wrong Angle of Needle Penetration.

5.2 Results

A total of 40 simulations tests were made in a commodity PC AMD 64 3200 with 1.0 GB of RAM running Fedora Core Linux 4, for 32 bits. The amount of data used for the assessment stage is

equivalent to a 20 seconds simulation with a haptics device frequency set at its maximum (at 1000 data/second). This time of simulation is equivalent to the average amount taken in marrow harvest procedure simulation [10]. In this case, the simulation was made using the maximum frequency allowed by haptic device.

The Table I presents a comparison of response time during the assessment stage for five classes of performance, where in each case presents a different dimension of variable set (2, 3, 4 and 5). For an on-line training assessment the user must receive the assessment report at most 1.5 seconds after the end of training.

Using the computer mentioned above, it can be measured simultaneously, a set of dimension five or five variables with dimension equal to one.

Table I. Comparison of average time of response for the performance tests made on the assessment class with 5 classes of performance and variable dimension.

Dimension (variable set)	Average Time	Deviation (number of simulations = 40)
2	0.015 seconds	0.0055
3	0.021 seconds	0.0037
4	0.030 seconds	0.0052
5	0.035 seconds	0.0059

Figures 5 to 9 presents comparison charts of the assessment precision. In each case the number of variable sets in the classes (performance class dimension) is different. Each chart shows the variation of the accuracy (calculated using the Kappa coefficient for comparisons [4]) when the size of variable set also varies. By the comparison of the 5 charts, a growth in the accuracy of the assessment class can be noticed when the performance class dimension becomes higher.

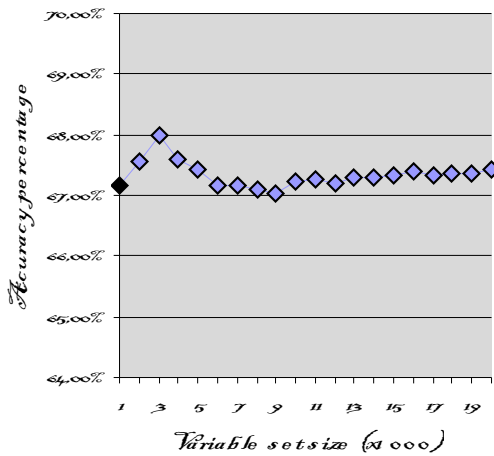


FIGURE 5. COMPARISON CHART OF ASSESSMENT ACCURACY WHEN VARIABLE SET SIZE VARIES FOR

CLASSES WITH VARIABLE SET DIMENSION EQUAL TO ONE.

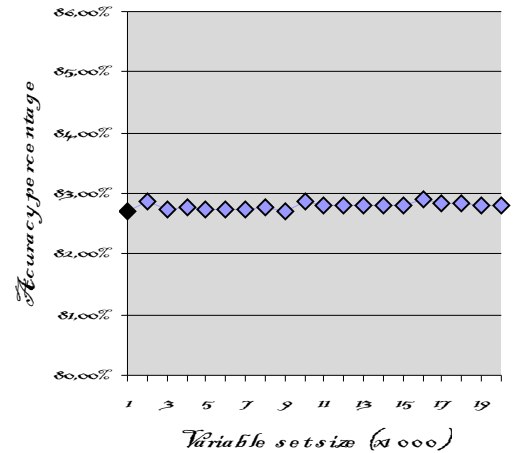


FIGURE 6. COMPARISON CHART OF ASSESSMENT ACCURACY WHEN VARIABLE SET SIZE VARIES FOR CLASSES WITH VARIABLE SET DIMENSION EQUAL TO TWO.

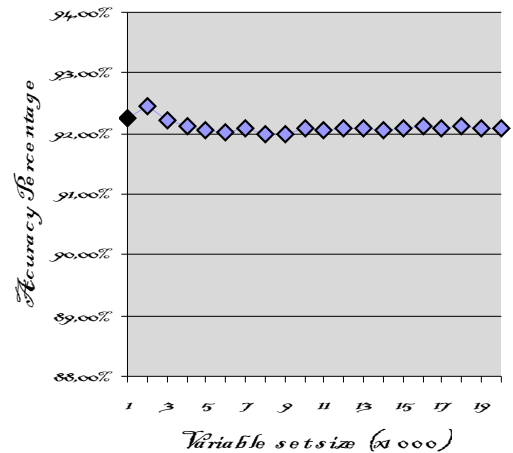


FIGURE 7. COMPARISON CHART OF ASSESSMENT ACCURACY WHEN VARIABLE SET SIZE VARIES FOR CLASSES WITH VARIABLE SET DIMENSION EQUAL TO THREE.

On the Figure 5 the accuracy is quite low compared to the others, this is expected due to the nature of the assessment method being used. Since one dimensional variable set (in this case) has only one parameter being analyzed this makes it less accurate. The more the number of parameters is being analyzed the better the accuracy, Figures 7, 8 and 9 show accuracy above 90%. The simulations shown in them are the ones that are more expected to be done for the test case, a variable set size equal to 3 can

represent the x, y, z coordinate position of the interaction object as the assessment parameters. The other two parameters (one of them incremented in Figure 8 and the other in Figure 9) can both represent penetration torque and angle of interaction relative to the object surface.

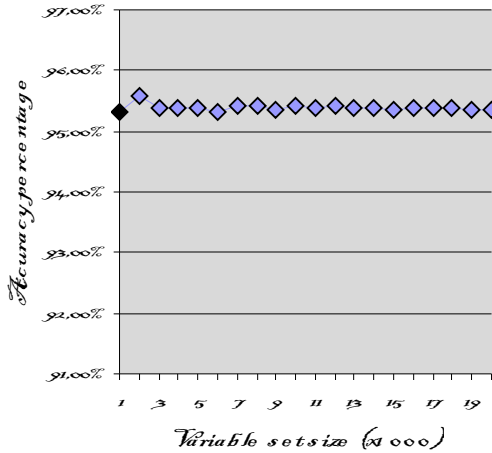


FIGURE 8. COMPARISON CHART OF ASSESSMENT ACCURACY WHEN VARIABLE SET SIZE VARIES FOR CLASSES WITH VARIABLE SET DIMENSION EQUAL TO FOUR.

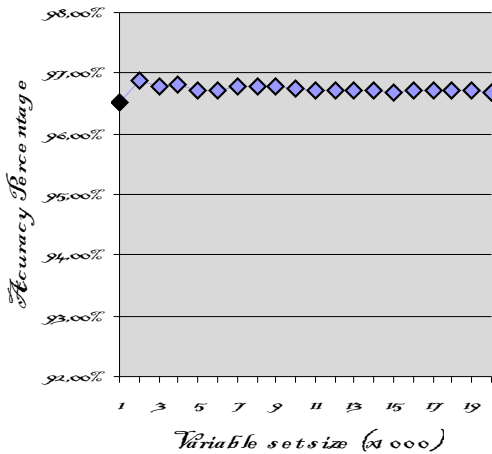


FIGURE 9. COMPARISON CHART OF ASSESSMENT ACCURACY WHEN VARIABLE SET SIZE VARIES FOR CLASSES WITH VARIABLE SET DIMENSION EQUAL TO FIVE.

The variable set size shown here represents the amount of data of each assessment parameter that are being analyzed. In each chart the size in the simulations goes from 1000 to 20000, being all of these simulations done at the 20 second period of time (rates of 50

to 1000 data per second). In this particular case of the assessment class (being it done as an on-line assessment) high rates are more acceptable for preventing the loss of user interaction data.

6. CONCLUSION

This paper presented the CybAssess, a set of classes developed for the CyberMed system. The CybAssess provides interfaces to support assessment methods and is completely synchronized with other tasks in a VRS developed with the CyberMed.

The design of the classes allows a high-level use of assessment methods for programmers and leaves internal to the system the details of the methods implementations. However, low-level access to assessment functions are also available and can be used by expert programmers. A class called CybMLM was designed and implemented to provide support to the MLM assessment method.

The assessment system implemented has growing accuracy, in direct function of the dimension of the variables set used, which was demonstrated by the case study. Besides, the accuracy is satisfactory, even using a set of variables with low dimensions, in this case dimension equal to three).

The average time of response for assessment is satisfactory using set of variables with dimension equal to five, for a commodity computer utilized. With better computers this frontier can be trespassed.

7. ACKNOWLEDGMENTS

This work is partially supported by Brazilian Council for Scientific and Technological Development, CNPq (Process 303444/2006-1 and process CT-INFO-CNPq 506480/2004-6) and Brazilian Research and Projects Financing, FINEP (Grant 01-04-1054-000).

8. REFERENCES

- [1] Basdogan, C.; De, S.; Kim, J.; Muniyandi, M.; Kim, H., et al., Haptic Minimally Invasive Surgical Simulation and Training, *IEEE Computer Graphics and Applications*, Vol. 24, No. 2. March-April 2004.
- [2] Burdea, G. e Coiffet, P. *Virtual Reality Technology*, New York: Wiley, 2nd ed, 2003.
- [3] Burdea, G., Patounakis, G., Popescu, V. and Weiss, R.E. *Virtual Reality Training for the Diagnosis of Prostate Cancer*. *Virtual Reality Annual International Symposium Proceedings*. IEEE, 1998. pp. 190-197.
- [4] Cohen, J., A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, v.20, p.37-46, 1960.
- [5] Cunha, I. L. L.; Machado, L. S.; Moraes, R. M. Unifying Data Structures for Virtual Reality Applications. *Proceedings of World Congress on Computer Science, Engineering and Technology Education (WCCSETE'2006)*. Marçõ, Santos, Brasil, 2006, 404-408.
- [6] Duda, R.; Hart, P.; Stork, D. *Pattern Classification*. *Wiley-Interscience*. 2nd ed., 2000.

- [7] Hartman, N. W., Connolly, P. E., Gilger, J. W., Bertoline, G. R., and Heisler, J. Virtual reality-based spatial skills assessment and its role in computer graphics education. In *Proc. SIGGRAPH*. ACM Press, New York, USA, 2006.
- [8] Johnson, R. A.; Wichern, D. W., *Applied Multivariate Statistical Analysis*. Prentice Hall, 4th ed., 1998.
- [9] Machado, L. S.; Valdek, M. C. O. ; Moraes, R. M.; Assesment of Gynecological Procedures in a Simulator Based on Virtual Reality. Proc. of the 7th International FLINS Conference on Applied Artificial Intelligence (FLINS'2006), Italy, 2006, pp. 799-804.
- [10] Machado, L. S., Mello, A. N., Lopes, R. D., Odone, Filho, 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*, v 81, 2001, 293-297.
- [11] 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.
- [12] McBeth, P. B. et al. Quantitative Methodology of Evaluating Surgeon Performance in Laparoscopic Surgery. *Studies in Health Technology and Informatics*. Vol. 85. 2002, pp. 280-286.
- [13] Moraes, R. M., and Machado, L. S. Maximum Likelihood for On-line Evaluation of Training Based on Virtual Reality. *Proceedings of Global Congress on Engineering and Technology Education (GCETE'2005)*. Santos, Brasil, Março 2005, p.299-302.
- [14] Moraes, R. M.; Machado, L. S. A Fuzzy Bayes Evaluator for On-Line Training Evaluation Based on Virtual Reality. *Proceedings of World Congress on Computer Science, Engineering and Technology Education (WCCSETE'2006)*. Março, Santos, Brasil, 2006, 399-403.
- [15] Moraes, R. M.; Machado, L. S. “Gaussian Mixture Models and Relaxation Labeling for On-line Evaluation of Training in Virtual Reality Simulators”. *Annals of International Conference on Engineering and Computer Education (ICECE'2003)*. March, Santos. 2003. [CDROM].
- [16] Rosen J., Richards, C., Hannaford, B. and Sinanan, M. Hidden Markov Models of Minimally Invasive Surgery. *Studies in Health Technology and Informatics*. Vol. 70. 2000, pp. 279-285.
- [17] 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*. Vol. 81. 2001, pp. 417-423.
- [18] Zhang, H., Payandeh, S., Dill, J., and Lomax, A. J. Acquiring laparoscopic manipulative skills: a virtual tissue dissection training module. *Studies in Health Technology and Informatics – Medicine Meets Virtual Reality*, v 98, 2004, 419-421.