A New Genetic Approach for the Partitioning Problem in Distributed Virtual Environment Systems [⋆]

Pedro Morillo[1], Pedro López[2], Juan Manuel Orduña[2] and Marcos Fernández[1]

[1] Instituto de Robótica. Universidad de Valencia. SPAIN
[2] Departamento de Informática. Universidad de Valencia. SPAIN
Pedro.Morillo@uv.es, Juan.Orduna@uv.es

Abstract. The *Partitioning problem* is a key issue in the design of Distributed Virtual Environment (DVE) systems based on a server-network architecture. This problem consist of efficiently assigning the clients of the simulation (avatars) to the system servers. Despite the existing literature proposes different evolutive approaches for solving this NP-hard problem, an approach based on genetic algorithms is considered as the current best partitioning mechanism.
In this paper, we analyze the impact of the low diversity of the initial population in this algorithm, and we propose a new mechanism for generating initial populations of higher quality. We also propose a new set of crossover methods oriented to problem specifications. Both improvements define a new genetic algorithm that provides better solutions than any other existing approach, in terms of both quality function and execution time.

1 Introduction

Distributed Virtual Environment (DVE) systems have experienced a spectacular growth last years. These systems allow multiple users, working on different computers that are interconnected through different networks (and even through Internet) to interact in a shared virtual world. This is achieved by rendering images of the environment as a user located at that point in the virtual environment would perceived them. Each user is represented in the shared virtual environment by an entity called *avatar*, whose state is controlled by the user input. Since DVE systems support visual interactions between multiple avatars, every change in each avatar must be propagated to some avatars in the shared virtual environment. DVE systems are currently used in many different applications [20], such as collaborative design [18], civil and military distributed training [12], e-learning [17] or multi-player games [1, 7].

One of the key issues in the design of a scalable DVE system is the *partitioning problem*. It consists of efficiently assigning the workload (avatars) among different servers in the system [8]. The partitioning problem determines the overall performance of the DVE systems, since it has an effect not only on the workload that each server in the system supports, but also on the inter-server communications (and therefore on the network traffic). Despite the partitioning problem in DVE systems has been usually

[⋆] Supported by the Spanish MCYT under Grants DPI-2002-04438-C02-02 and TIC-2003-08154-C06-04 addressed with ad-hoc procedures [20, 9], recent works propose partitioning schemes following evolutive approaches [14–16]. One of these approaches, based on a genetic algorithm, offers good performances in terms of execution time and low values of quality function.

In this paper, we propose a set of improvements for this genetic approach in order to develop more scalable and cost-effective DVE systems. These improvements consists of maximizing the structural diversity of initial population and also of incorporating a new crossover mechanism. In this mechanism, each chromosome of the current population randomly chooses a crossover operation from a small set of oriented crossovers. Performance evaluation results show that, due to its ability of avoiding premature convergence of solutions, the proposed method can provide better solutions while requiring shorter execution time than other methods proposed in the literature.

The rest of the paper is organized as follows: Section 2 describes the partitioning problem and the proposed techniques for solving it. Section 3 shows the implementation of the proposed genetic approach for solving the partitioning problem. Next, Section 4 presents the performance evaluation of the proposed search method. Finally, Section 5 presents some concluding remarks and future work to be done.

2 The Partitioning Problem in DVE systems

Architectures based on networked servers are becoming a de-facto standard for DVE systems [20, 9, 15]. In these architectures, the control of the simulation relies on several interconnected servers. Multi-platform client computers are connected to one of these servers. When a client modifies an avatar, it also sends an updating message to its server, that in turn must propagate this message to other servers and clients. Servers must render different 3D models, perform positional updates of avatars and transfer control information among different clients. Thus, each new avatar represents an increasing in both the computational requirements of the application and also in the amount of network traffic. When the number of connected clients increases, the number of updating messages must be limited in order to avoid a message outburst. In this sense, concepts like areas of influence (AOI) [20] or locales [2] have been proposed for limiting the number of neighboring avatars that a given avatar must communicate with. All these concepts define a neighborhood area for avatars, in such a way that a given avatar must notify his movements (by sending an updating message) only to those avatars located in that neighborhood. These avatars are denoted as neighbor avatars.

Depending on their origin and destination avatars, messages in a DVE system can be intra-server or inter-server messages. Figure 1 shows an example of a DVE system consisting of several servers interconnected through a network. This figure also shows an example of both intra-server and inter-server avatar updating messages.In this figure, avatars are uniformly distributed and they are represented as dots. Each server manages a given number of clients (avatars) and decides which avatars are the destinations for the messages received from other avatars. This figure also shows an example of both intra-server and inter-server avatar updating messages. Inter-server messages are those messages whose origin and destination avatars are assigned to different servers. Otherwise, the message is an intra-server message. In order to design scalable DVE systems, the number of inter-server messages must be minimized. Effectively, when clients send intra-server messages they only concern a single server. Therefore, they are minimizing the computing, storage and communication requirements for maintaining a consistent view of the virtual world to all avatars in a DVE system.

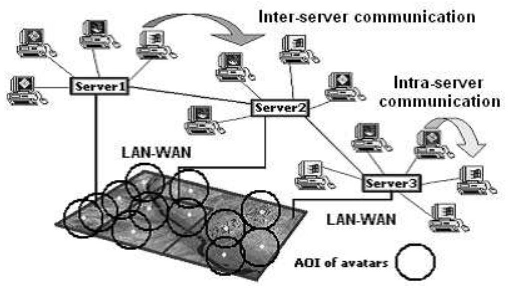

Fig. 1. Example of Multi-server DVE system

Lui and Chan propose in [8] propose a quality function, denoted as C_p, for evaluating each assignment of clients to servers. This quality function takes into account two parameters. One of them consists of the computing workload generated by clients in the DVE system, denoted as C_p^W. In order to minimize this parameter, the computing workload should be proportionally distributed among all the servers in the DVE system, according to the computing resources of each server. The other parameter of the quality function consists of the overall inter-server communication cost for sending the messages generated by all avatars, denoted as C_p^L. In order to minimize this parameter, avatars sharing the same AOI should be assigned to the same server. Quality function C_p is defined as

$$C_p = W_1\, C_p^W + W_2\, C_p^L \tag{1}$$

where W_1 and W_2 are two coefficients that weight the relative importance of the computational and communication workload, respectively ($W_1 + W_2 = 1$). These coefficients should be tuned according to the specific features of each DVE system. Using this quality function (and assuming $W_1 = W_2 = 0.5$) Lui and Chan propose an ad-hoc approach, called LOT, that re-assigns clients to servers [9]. The partitioning algorithm should be periodically executed for adapting the partition to the current state of the DVE system as it evolves (avatars can join or leave the DVE system at any time, and they can also move everywhere within the simulated virtual world). Lui and Chan also have proposed a testing platform for the performance evaluation of DVE systems, as well as a parallelization of the partitioning algorithm [9].

Since the partitioning problem in DVE systems can be considered as a a combinatorial optimization problem, different solutions based on metaheuristic techniques has been proposed [13–16]. Despite [13] describes a constructive strategy based on GRASP, the best partitioning results has been obtaining using evolutive techniques [15]. Among these evolutive techniques a genetic methods offer especially excellent results in terms of execution time and quality of the provided solutions.

Although the genetic approach proposed in [15] performs reasonably well, it is based on the generation of an initial population obtained by deriving a unique solution provided by a k-means clustering algorithm [16]. Despite this feature offers an improved initial population of feasible solutions it, does not focus on maximizing the structural diversity of chromosomes. As described in [11] and [4], this low level of structural diversity can lead the algorithm to reach a local minimum or even a poorer approximation to this value. Additionally, the crossover mechanism used by this algorithm is based on an auto-fertilization technique, where chromosomes are derived following a single-point crossover [5]. This crossover mechanism is excessively generalist, and it is possible to offer new crossover strategies more oriented to problem specifications.

3 A New Genetic Technique for Solving the Partitioning Problem

In order to improve the performance of our current approach for solving the partitioning problem in DVE systems, we propose a new version of the genetic algorithm. This new version focuses on maximizing the structural diversity of initial population and improving the crossover operator. The new algorithm replaces the current generation of the initial population with a new method based on random projections. Additionally, the crossover is performed by randomly choosing a mechanism from a list composed of five different crossover operators. All these five operators are very oriented to the specifications of partitioning problem in DVE systems.

3.1 Generation of a Heuristic Initial Population

Most of metaheuristic are based on the fast generation of an initial population of elements [15, 11]. This initial population usually represents a set of poor solutions to the problem, and it is evolved through a crossover operator until a stopping criterion (for example, a given number of iterations) is reached.

If the initial population has been correctly defined, then the metaheuristic algorithm easily obtains a good approximation to the global optimum. Moreover, as it is described in [11], if the initial population is not randomly selected then the algorithm should maintain a certain level of structural diversity among all the chromosomes, in order to properly represent the whole set of feasible solutions and to avoid the premature convergence of the search.

Taking into account these considerations, we propose a new mechanism, called *Projections algorithm (PA)*, in order to generate the new population of initial solutions. This fast algorithm provides a set of independent and well-diversified initial solutions to the problem.

PA consists of a given number of iterations n_c that defines the number of chromosomes in the population (population size). Each one of the n_c chromosomes represents a complete partitioning solution to the problem where all the N avatars $(A_0, .., A_{N-1})$ in the DVE are assigned to the M servers $(S_0, .., S_{M-1})$ in the system. An iteration consist of four steps (for the sake of clearness, we will consider in the following description that avatars move across a 2-D virtual world. The extrapolation to 3-D worlds is reasonably trivial), illustrated in Figure 2:

First, each avatar in the system is assigned to server S_0 (Fig.2a). Next, a random value θ between 0 and $\pi/2$ is generated. Since all avatars are located on a Cartesian plane, PA draws a straight line which passes through the zero coordinates (0,0) with a slope of θ radians (fig.2b). This line and its perpendicular define a new coordinate axis that is rotated θ radians with respect to the original position. Using a simple affine transformation (fig.2c), the old coordinates (X_i, Y_i) of each avatar can now be expressed with respect to the rotated axis by the new coordinates (X_i', Y_i'). At this point, PA generates two different *binary search trees* with X_i' and Y_i' search keys of each avatar. Once both trees are created, the N/M different avatars in both trees with the highest and the lowest keys are put into four different sets [19]. In order to assign a set of avatars to a server, the third step of PA evaluates separately the different sets of avatars using the C_p function. Since both sets have the same cardinality N/M, PA algorithm only computes the C_p^L term. The term C_p^W is not computed, since it evaluates the standard deviation of the assigned avatars with respect to the the perfect balancing. The last step is to select the set with the lowest C_p^L value, and all avatars contained in this set are assigned to server S_1 (fig.2d). At this point, $N(M-1)/M$ avatars are still assigned to server S_0, and N/M avatars are assigned to server S_1. Next iteration allows server S_0 to lose another N/M avatars, which are assigned to S_2. PA algorithm finishes when the last group of avatars (with a size less or equal to N/M avatars) is assigned to server S_{M-1}. At this point, a number of avatars very close to N/M are assigned to each server server in the DVE system.

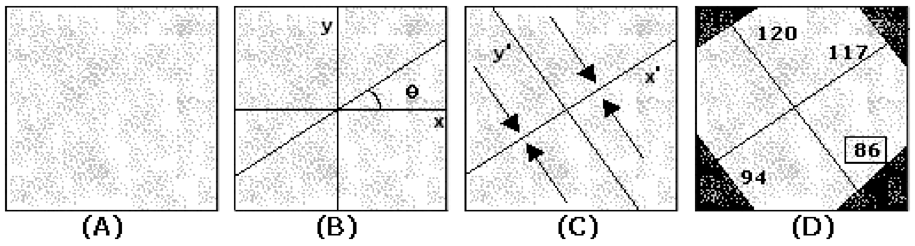

Fig. 2. Generation of the initial population based on *Projection Algorithm*

It is important to mention that the use of 4 binary search trees allows to improve diversity when selecting the sets of avatars to be assigned to the target server for each iteration. The random generation of rotation angles guarantees inherently independent solutions in the generation of individuals of the initial population. Moreover, this population allows the genetic approach to evolve solutions with good values of quality function C_p. These good C_p values are achieved by balancing the number of avatars among the servers and assigning to the same server the avatars that are closely located in the virtual scene.

3.2 Providing Randomness to Chromosome Crossover

In order to evolve chromosomes, the first approach for solving the partitioning problem in DVE systems, based on a genetic algorithm uses an auto-fertilization technique [16]. In this technique, each chromosome generates a new chromosome following a single-point crossover with a probability equal to one [5]. When the crossover operator is applied to all individuals of the population (the child population has been completely created) a elitist selection guarantees the survival of the best individuals. This crossover operator is excessively generalist and has been used often for solving different combinatorial problems.

The proposed technique is based on the random selection of crossover operators from a list. This list, which is accessed for each derivation, consists of five operators very oriented to problem specifications. The operator list consists of the following elements:

Operator 1 Random exchange of the current assignment for two *border* avatars. A given avatar A_i is a border avatar if it is assigned to a certain server S_r in the initial partition and any of the avatars in its AOI is assigned to a server different from S_r [14].

Operator 2 Once a border avatar A_i has been randomly selected, it is randomly assigned to one of the servers S_f hosting the border avatars of A_i.

Operator 3 Besides the step described in the previous operator, if it exists an avatar A_j such that A_j is assigned to S_f and it is a neighbor avatar of A_i, then A_j is assigned to S_r.

Operator 4 Since each avatar generates a certain level of workload in the server where it is assigned to [8], then it is possible to sort the servers of a DVE system according to the level of workload they support. If S_m and S_n are the servers with the highest and the lowest level of workload in the system, respectively, then a random avatar A_k assigned to S_m is assigned to S_n.

Operator 5 Besides the step described in the previous operator, a random avatar A_l, initially assigned to S_n, is now assigned to S_m.

4 Performance Evaluation

This section presents the performance evaluation results obtained with the proposed new genetic algorithm described in the previous section when it is used for solving the partitioning problem in DVE systems. Following the standard evaluation methodology described in [8] and used in [9, 13–16], we have empirically tested the new approach in two examples of a DVE system: a SMALL world, composed by 13 avatars and 3 servers, and a LARGE world, composed by 2500 avatars and 8 servers. We have considered two parameters: the value of the quality function C_p for the partition provided by the proposed search method and also the computational cost, in terms of execution time, required by the search method in order to provide that partition.

Our evaluation tool models the behavior of a generic DVE system with a server-network architecture on a real network of heterogeneous computers. Each server is implemented in a single PC, while up to 50 clients is allocated in the same PC. Following this configuration, a battery of DVE systems was tested. This battery was composed by 400 SMALL worlds and 300 LARGE worlds. We have used a 10 Mbps Ethernet as the interconnection network. The hardware platform used for the evaluation has been a Pentium IV at 1.7 GHz 256 Mbytes of RAM. The operating system was Windows 2000 Professional operating system.

4.1 Tuning of Genetic Algorithm

As described in [14, 16], the parameters of the genetic algorithm that should be tuned in order to achieve optimal performance are the population size, the number of generations, and the mutation rate.

Figure 3 shows the convergence of the proposed approach as the number of generations and mutation rate vary in a LARGE DVE configuration. This convergence is expressed in terms of fitness function C_p. Due to space limitations, the variation of the population size is not shown in this figure. The incidence of this parameter in the behavior of the algorithm is very similar to the incidence of the number of generations in the same DVE system.

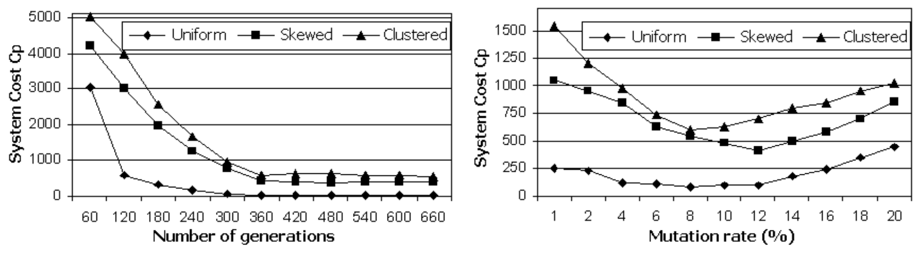

Fig. 3. C_p values obtained for different amounts of generations and mutation rates in a LARGE DVE system

Figure 3 shows (the graphic on the left) that for all the considered distributions, the C_p value provided by the proposed algorithm decreases as the number of iterations increases, until a value of 400 iterations is reached. From that point, quality function C_p slightly decreases or remain constant, depending on the considered distribution of avatars. The same behavior is shown for the population size when it reaches values close to 20 chromosomes. Although it is not shown here, values bigger than this number of iterations require too much execution time and do not reach significantly better solutions in terms of C_p. On other hand, a different behavior is observed when the mutation rate is varied (graphic on de right in figure 3). The values of quality function C_p provided by the proposed algorithm decrease as this parameter grows, until the system explore values close to 8-9%. From these points, genetic algorithm starts to obtain worse solutions in term of C_p. The reason for this behavior is that if an excessive number of mutations is performed for a given generation of individuals, then the evolving process of maintaining a set of high quality solutions is excessively degraded.

Therefore, tuning of the selected parameters for the proposed genetic method and for the LARGE DVE systems has been 400 iterations, 20 chromosomes and a mutation rate of 9%.

4.2 Evaluation Results

For comparison purposes, we have evaluated the performance of the proposed approach, as well as the *linear optimization technique* (LOT) described in [9] and also the basic genetic approach (BGA) presented in [16]. The latter method currently provides the best results for the partitioning problem in DVE systems. In the case of SMALL worlds we have also performed an exhaustive search through the solution space, obtaining the best partition as possible. Since this exhaustive method requires the exploration of a domain composed by 3^{13} (1.594.323) different solutions in SMALL worlds, this problem becomes unaffordable when LARGE worlds are considered (8^{2500} different solutions). On other hand, since the performance of the heuristic search methods may heavily depend on the location of the avatars, we have considered three different distributions of avatars: uniform, skewed, and clustered distribution.

Table 1 and table 2 show the evaluation results for a SMALL and LARGE virtual worlds, respectively. These tables show the C_p values corresponding to the final partitions provided by two versions of proposed genetic approach for a SMALL and LARGE virtual worlds, and also the execution times required in order to obtain these final partitions. Additionally, they also show the same results obtained for BGA and LOT methods.

	Uniform distribution		**Skewed distribution**		**Clustered distribution**	
	T_{exe}(sec.)	C_p	T_{exe}(sec.)	C_p	T_{exe}(sec.)	C_p
Exhaustive	3.411	6.54	3.843	7.04	4.783	7.91
LOT	0.0009	6.56	0.001	8.41	0.0011	8.89
BGA	0.002	6.54	0.003	7.04	0.005	7.91
V1	0.002	6.54	0.003	7.041	0.006	7.91
V2	0.003	6.54	0.003	7.04	0.006	7.91

Table 1. Results for a battery of SMALL DVE systems

	Uniform distribution		**Skewed distribution**		**Clustered distribution**	
	T_{exe}(sec.)	C_p	T_{exe}(sec.)	C_p	T_{exe}(sec.)	C_p
LOT	30.94	1637.04	32.18	3460.52	43.31	5903.80
BGA	6.65	1832.2	13.79	2825.6	29.22	4905.93
V1	6.24	547.2	14.05	612.9	28,74	1002.51
V2	6.41	321.3	14.59	450.8	28.65	791.94

Table 2. Results for a battery of LARGE DVE systems

In order to evaluate the improvements introduced by the methods proposed in this paper, *V1* version implements the BGA approach where the initial population has been created following the projection algorithm presented in section 3.2. In addition to this improvement, V2 version not only starts with this new initial population but also it implements the crossover mechanism described in section 3.1.

These tables show that both V1 and V2 approaches obtain similar results than LOT and BGA methods for all considered distribution of avatars in SMALL worlds. However, as it is described in [9] and [15], the main purpose of a partitioning method is to improve the scalability of DVE systems. Therefore, it must provide a significant performance improvement when it is used in LARGE DVE systems. The results obtained for the LARGE world show that the quality of the provided partitions are increased in terms of C_p values. Both V1 and V2 approaches require similar execution times than BGA method. However, V2 method is able to decrease C_p function from 1832.2s. to 321.3s when uniform distributions of avatars are considered. In the case of skewed and clustered distributions, C_p values are decreased in a similar proportion by both methods V1 and V2. These results show that by simply improving the structural diversity of the initial population (V1 version) it is possible to evolve the population of chromosomes until more efficient solutions.

5 Conclusions and Future Work

Current Distributed Virtual Environments (DVE) are usually designed following server-network architectures. In these architectures, a NP-hard problem called the partitioning problem has become a critical issue in order to design efficient and scalable DVE systems.

In this paper, we have analyzed and improved a recent method based on a genetic algorithm designed for solving this problem. This method currently provides the best results for the partitioning problem in DVE systems. One of the proposed improvements for this genetic method consists of using a new stochastic algorithm for the generation of the initial population that maximizes the structural diversity of chromosomes. On other hand, we have proposed the replacement of the traditional single-point crossover operator by a pool of five different crossover mechanisms oriented to problem specification.

Performance evaluation results show that the proposed implementation of the genetic method provides better solutions to the partitioning problem than the current approaches to the problem. Therefore, the proposed approach can improve the efficiency and scalability of DVE systems.

As future work to be done, we plan to design a parallel implementation of GRASP approach. This new design will be based on a master-slave configuration and will be implemented in conjunction with a post-optimization procedure.

References

1. M. Abrash, “Quake’s game engine: The big picture”, *Dr. Dobb’s. Journal*.Spring, 1997.

2. D.B.Anderson, J.W.Barrus, J.H.Howard, Building multi-user interactive multimedia environments at MERL, in *IEEE Multimedia*, 2(4), pp.77-82, Winter 1995.
3. T.A. Funkhouser, “Network Topologies for Scalable Multi-User Virtual Environments”, in *Proceedings IEEE VRAIS ‘96*, San Jose, CA. April, 1996.
4. J.H. Holland and D.E. Goldberg, “Genetic algorithms and machine learning: Introduction to the special issue on genetic algorithms”, *Machine Learning*, 3, 1998.
5. K.E. Kinnear, “Alternatives to Automatically Function Definition”, in Advances in Genetic Programming 1994, *MIT Press*, pp. 119-141.
6. E. Kirshenbaum, “Genetic Programming With Statically Scoped Local Variables”, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2000*, July, 2000, pp. 459-468
7. M. Lewis and J. Jacboson, “Game Engines in Scientific Research”, in *Communications of the ACM*, Vol 45. No.1, January 2002.
8. J.C.S. Lui, M.F. Chan and K. Oldfield, “Dynamic Partitioning for a Distributed Virtual Environment”, *Dpt. of Computer Science*, The Chinese University of Hong Kong, 1998.
9. J.C.S. Lui and M.F. Chan, “An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems”, *IEEE Trans. Parallel and Distributed Systems*, Vol. 13, No.3, pp. 193-211. March 2002.
10. M.R. Macedonia, “A Taxonomy for Networked Virtual Environments”, *IEEE Multimedia*, 4(1) 48-56, January-March 1997.
11. Z. Michalewicz, “Genetic Algorithms + Data Structures = Evolution Programs”, *Springer-Verlag*, Second Edition, 1992.
12. D.C.Miller and J.A. Thorpe, “SIMNET: The advent of simulator networking”, in *Proceedings of the IEEE*, Vol. 83, No.8, pp. 1114-1123. August, 1995.
13. P. Morillo and M. Fernández, “A GRASP-based algorithm for solving DVE partitioning problem”, *in Proceedings of 2003 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPS-03*), Nice, France, April, 2003.
14. P. Morillo, M.Fernández and J.M.Ordua, “An ACS-Based Partitioning Method for Distributed Virtual Environment Systems”, *in Proceedings of 2003 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPS-03)*, Nice, France, April, 2003.
15. P. Morillo, M.Fernández and J. M. Orduña, “A Comparison Study of Modern Heuristics for Solving the Partitioning Problem in Distributed Virtual Environment Systems”, in *International Conference in Computational Science and Its Applications (ICCSA’ 2003)*, volume 2669 of Springer LNCS, pp. 458-467, Montreal, Canada. May 2003.
16. P. Morillo, M.Fernández and N.Pelechano, “A grid representation for Distributed Virtual Environments”, *in Proceedings of 2003 1st European Across Grids Conference*, Santiago de Compostela, Spain, February, 2003.
17. T. Nitta, K. Fujita and S. Cono, “An Application Of Distributed Virtual Environment To Foreign Language”, in *Proceedings of FIE’2000. IEEE Education Society.* Kansas City, Missouri, October 2000.
18. J.M. Salles, Ricardo Galli, A. C. Almeida et al, “mWorld: A Multiuser 3D Virtual Environment”, in *IEEE Computer Graphics*, Vol. 17, No. 2. March-April 1997.
19. R. Sedgewick, “Algorithms in C”, 3rd ed., Addison-Wesley, 1998.
20. S.Singhal and M.Zyda, “Networked Virtual Environments”, *ACM Press*, New York, 1999.