Annals of University of Craiova, Math. Comp. Sci. Ser.
Volume 34, 2007, Pages 115–123
ISSN: 1223-6934

A Mobile Agents approach for 3D elastic medium modeling and visualization

Claudiu Popirlan and Mihai Dupac

Abstract. The importance of 3D modeling in virtual reality and computational vision was a challenging subject and a long motivation for researchers. In this paper the modeling and visualization of a 3D elastic model using a mobile agents technology is presented. The 3D elastic model is represented with particles connected with elastic springs. The mobile agents are used to extract the forces information (internal and external forces) and change the elastic properties of the model, for a better representation and visualization. The expression of the spring elastic constants are presented and continuously updated based on the forces evaluation. A sailing ship modeled using the 3D elastic medium is simulated and visualized.

2000 Mathematics Subject Classification. Primary 68T99; Secondary 65D17.
Key words and phrases. mobile agents, dynamic modeling, visualization, object oriented implementation.

1. Introduction

Modeling and visualization techniques are indispensable tools in order to interpret the data and to have a better understanding of system dynamics. An elastic model simulation requires the update of the model at each time step according to some deformation law and can be considered like a new object representation at each time step. Mobile agents are a novel way for the modeling of a 3D elastic medium in distributed software systems. Traditional systems are implemented using stationary programs that pass data back and forth across a network. Mobile agents, by contrast, are programs that themselves move from node to node: the computation moves, not just the attendant data.

Using Mobile Agent Technology, Xinfeng Yang *et al.* [2] studied the problem of web-based face recognition, involving not only face detection but also modeling facial feature medium and graph matching schemes. James Nolan it et al. [3] introduces a scalable, flexible agent-based architecture for collaborative image processing. The architecture supports image processing agents, combined knowledge encoding and Agent Communication Language. A mobile agents approach for knowledge representation was developed in [4] using the Recursive Modeling Method, for expected utility calculation.

A successful modeling of deformable surfaces based on a mass-spring approach has been introduced in [13]. Using techniques from the mechanical engineering field [6] such as energy minimization and finite element method Terzopoulos *et al.* [7, 8] studied the modeling and simulations aspects of deformable surfaces. A dynamic modeling based on a model with particles have been studied by Breen *et al.* [9] and Eberhardt *et al.* [10]. Energy-based models have been studied by Carignan *et al.* [11] and Baraff and Witkin [12].

Received: 11 December 2007.

A spring-particles model for melting solids with various forces applied to their neighbors was developed in [14]. A model using springs was developed in [15], and a mass spring system for the study of the elastic body deformation has been studied in [16].

In this paper a mobile agents approach for the modeling and visualization of a 3D elastic model is proposed. The system has been designed and developed under a mobile agents technology to set up the 3D model properties and parameters, resulting the 3D model for display. The 3D model can be viewed now as a network of connected particles (grid). The mobile agents used in the application are defined by the next important properties.

Agents encapsulate a thread of execution along with a bundle of code and data. Each agent runs independently of all others, is self-contained from a programmatic perspective, and preserves all of its state when it moves from one particle to another. This is "strong mobility" as defined in ([1]).

Any agent can move easily across the grid. The underlying structure provides a language-level primitive that an agent can call to move itself to a neighboring particle.

An agent is able to identify and use information about specific resources (forces) to any particle on which it finds itself. In the presented simulation, the nodes where the particles are located are differentiated only by their neighbors (the agents use each node information to update the spring constants).

Agents are small in size, designed to be as minimal as possible.

Any agent cooperate with other agents in order to perform dynamic tasks. The agents read from and write to the associated information on each node, using this facility to coordinate with other agents and to leave information behind for subsequent visitors.

For the study of the 3D elastic model, in response to an exerted dynamical force the 3D elastic model is represented with particles connected with elastic springs. The external/internal forces (gravitational force, forces exerted between the particles, external forces such as wing forces), are evaluated on each particle. The mobile agents extract the forces information and change and update the elastic properties of the model defined using the springs elastic properties. The expression of the spring elastic constants, the position and the velocity of each particle are continuously updated based on the forces evaluation and visualization.

2. Mobile Agent Computing Technology

Comparing with earlier paradigm such as process migration or remote evaluation in distributed computing, mobile agent model is becoming popular for network programming. Traditional client/server paradigm relies on handshake mechanism to communicate over a network. The client requests information, while the server responds. Each request/response has to be a complete round trip on the network. The emerging mobile-agent paradigm has redefined the way Internet-based applications work. As an autonomy software entity with pre-defined functionality and certain intelligence, mobile agent is capable of migrating autonomously from one host machine to another, making its request to the server directly and performing tasks on behalf of its master. Furthermore, following certain working pattern, multi-agents can friendly cooperate to accomplish a more complicated task, providing a dynamic and flexible platform for a wide range of software applications. During the past few years, more

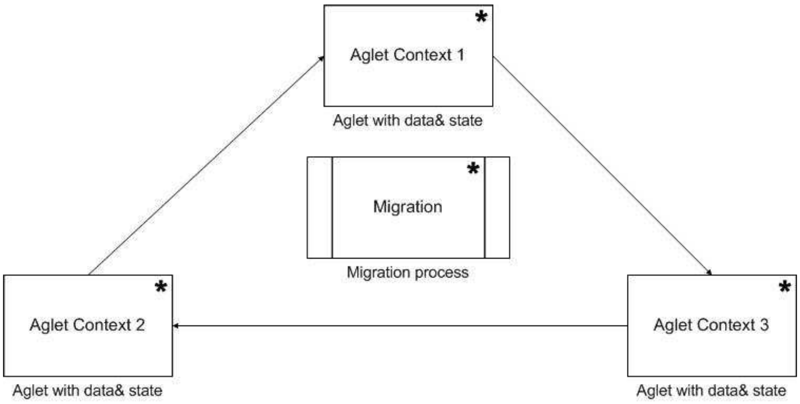

Figure 1. Basic migration paradigm of an aglet

than dozen Java-based agent systems have been developed. The Java Virtual Machine, standard security manager and two other functional facilities, namely object serialisation and remote method invocation(RMI) have made it simple to build mobile agent workbench. Of all these available Java-based systems, General Magics Odyssey ObjectSpaces Voyager and IBMs Aglet are three leading commercial ones. A detailed discussion on current commercial and research-based agent systems can be found in [5].

The Aglet system developed by IBM is chosen as the implementation example in our proposed system. Although it is not a full-fledged platform until now, it shows promises as a functional technology that fits very well into the Java world. An aglet is characterized as: lightweight objects migration, built with persistent support, event driven and so on.

Central to the aglet architecture is the context, which is the server environment for aglet execution. When an aglet has finished its work in a context, its state and data will be serialised to a stream of bytes and exported to the new context through Agent Transfer Protocol. The basic migration paradigm of aglet can be illustrated as Fig. 1.

3. 3D Elastic Model Description

For the study of the 3D elastic model, a 3D mechanical system [16] with particles in contact is proposed. The contact between particles is modeled with springs, resulting a lumped mass lumped spring model as shown in Fig. 2.

The 3D mechanical system is described as n interconnected particles $P_1, ..., P_n$, $P_i \neq P_j$, $\forall i \neq j$, $i, j \in 1, 2, ..., n$ included on the system. A moving reference frame, to describe the relative position of each particle is considered. For the moving reference frame $O'x'y'z'$ the relative position of a particle P_l with respect to a particle P_k is described in a spherical coordinates system by $(d_{P_kP_l}, \phi_{P_kP_l}, \theta_{P_kP_l})$, where $d_{P_kP_l}$ represents the distance between the particles P_k and P_l, $\theta_{P_kP_l}$ the azimuth angle and $\phi_{P_kP_l}$ is the horizontal azimuth angle.

The relations between the Spherical coordinates and the Cartesian coordinates for the mass center of a particle P_l with respect to the mass center of a neighbor particle

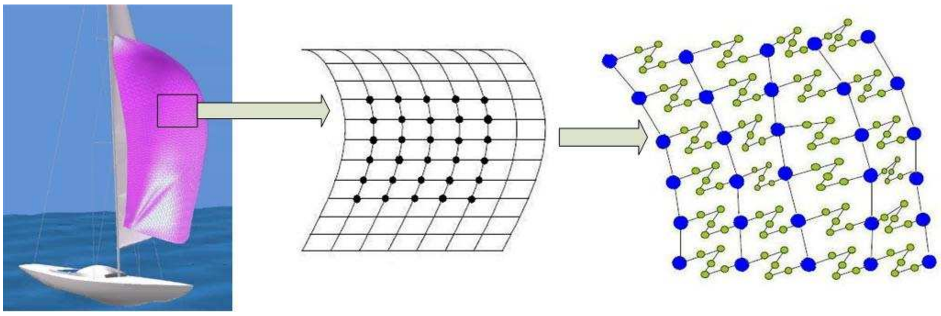

FIGURE 2. Contact between particles modeled using springs

P_k can be written as

$$\begin{aligned} x'_{P_kP_l} &= d_{P_kP_l}\sin\theta_{P_kP_l}\cos\phi_{P_kP_l}, \\ y'_{P_kP_l} &= d_{P_kP_l}\sin\theta_{P_kP_l}\sin\phi_{P_kP_l}, \\ z'_{P_kP_l} &= d_{P_kP_l}\cos\theta_{P_kP_l}. \end{aligned}$$

The position vector of a particle P_l with respect to the mass center of a neighbor particle P_k can be expressed in the $O'x'y'z'$ Cartesian reference frame as

$$\mathbf{r'}_{P_lP_k} = x'_{P_kP_l}\mathbf{\imath} + y'_{P_kP_l}\mathbf{j} + z'_{P_kP_l}\mathbf{k},$$

where $\mathbf{\imath}$, $\mathbf{j}$, $\mathbf{k}$ represents the units vector of the attached Cartesian reference frame.

The position of each particle P_i, $i = 2, ..., n$ can be defined in a fixed Cartesian reference frame $Oxyz$, using the position vectors $\mathbf{r}_{P_1} = \lambda_x\mathbf{\imath} + \lambda_y\mathbf{j} + \lambda_z\mathbf{k}$ of the particle P_1, and $\mathbf{r'}_{P_lP_{l+1}}$, $l = 1, 2, ..., i-1$. One can write

$$\mathbf{r}_{P_i} = \mathbf{r}_{P_1} + \sum_{l=1}^{i-1}\mathbf{r'}_{P_lP_{l+1}}.$$

The fixed reference frame $Oxyz$ is considered to be parallel with the moving reference frame $O'x'y'z'$, i.e. $Ox\|O'x'$, $Oy\|O'y'$ and $Oz\|O'z'$. Moreover, the fixed reference frame have the same orientation with the moving reference frame $O'x'y'z'$ attached to each particle. The velocity vector $\mathbf{v}_{P_i}$ of a particle P_i, $i = 2, ..., n$ is the derivative with respect to time of the position vector of $\mathbf{r}_{P_i}$, $\mathbf{v}_{P_i} = \dot{\mathbf{r}}_{P_i}$. The acceleration vector $\mathbf{a}_{P_i}$ of the particle P_i, $i = 2, ..., n$ is the derivative with respect to time of the position vector of $\mathbf{r}_{P_i}$. The fundamental law of dynamics $\mathbf{F}_{P_i} = m_{P_i}\mathbf{a}_{P_i}$, described the evolution of the elastic system. In the previous law m_{P_i} represents the mass of each particle P_i and $\mathbf{a}_{P_i}$ represents the acceleration related to the force $\mathbf{F}_{P_i}$, force regarded as the combination of external and internal forces acting on the particle P_i. The external forces are contact forces, gravitational force, and damping. The damping is an important factor that can be used in computation of the mechanical energy dissipation for the elastic model. The gravitational force acting on a particle P_i is $\mathbf{G}_{P_i} = m_{P_i}\mathbf{g}$, where $\mathbf{g}$ is the gravitational acceleration. The internal force $\mathbf{F}^{int}_{P_i}$ expressed as

$$\mathbf{F}^{int}_{P_i} = \sum_{j\neq i} k^c_{ij}\left[\frac{\|\mathbf{r}_{P_iP_j}\| - l_{P_iP_j}}{\|\mathbf{r}_{P_iP_j}\|}\right]\mathbf{r}_{P_iP_j}, \tag{1}$$

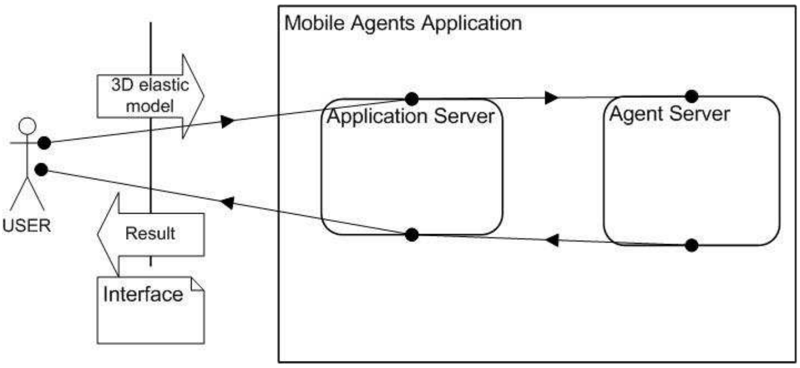

Figure 3. General structure of the application

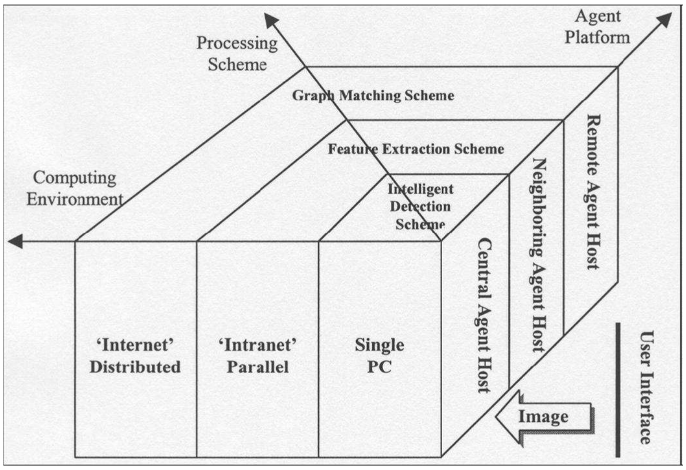

Figure 4. Mobile Agents System implementation

is the resultant of the springs tension linking the particle P_i to the neighbor particles, k_{ij}^c is the spring stiffness and $l_{P_iP_j}$ is the length of the spring linking the particle P_i with the particle P_j.

For the derivation of the spring constants a procedure developed in Dupac *et al.* [15] is used. For the mass spring model shown in Fig. 2, the particles are considered to be spheres having the same radius r_0. Each particle is connected only with the neighbor particles through linear springs. The springs stiffness are calculated as

$$k_n = \frac{3}{2}\frac{EV}{Mr_0^2(1-2\nu)}, \quad k_s = \frac{3}{2}\frac{EV(4\nu-1)}{Mr_0^2(2\nu^2+\nu-1)}, \quad k_t = \frac{3}{2}\frac{EV(4\nu-1)}{Mr_0^2(2\nu^2+\nu-1)},$$

where E is the Young's modulus and ν is the Poisson's ratio of the material (for more details about the stiffness computation see Dupac *et al.* [15]).

4. Mobile Agents System implementation

4.1. Mobile Agents for the 3D elastic structure modeling. To make accessing the system as easy as possible, a point-to-point input/output structure is explicitly

Figure 5. Sail of the Sailing Ship Initial Deformation

established to connecting the input/output device and the application server. A user interface is used to accept the input (3D elastic model) and return the updated model after mobile agents processing. The general structure of application is shown in Fig. 3.

To achieve high performance and better robustness, a multi-agent system has been proposed. The system framework can be viewed as an operational model, as shown in Fig. 4:

- A central mobile agent host block implementing an intelligent detection scheme on a single PC.
- A neighboring mobile agent host block implementing a feature extraction scheme in a parallel-computing environment.
- A remote mobile agent host block implementing a remote graph-grid scheme in a distributed computing environment.

Remote graph-grid (shown in Fig. 2) scheme is implemented using the Aglet system. The input is the 3D elastic model, described in previous sections. Aglet system makes more sense if the mobile agent moves to the remote data source for searching and modeling, rather than transferring large volumes of data over the network for processing. In this implementation, we explicitly create a modeling agent, initializing it with various algorithms and dispatching it to the 3D elastic model structure. Upon reaching a new node, the modeling agent interacts with remote agents and communicates with the databases for searching and updating. A modeling scheme will be performed: geometric-based coarse modeling and dynamic-link-architecture based on the 3D elastic medium. After the modeling agent achieved its pre-defined goal, it migrate to the next host until returning home with the results. Some advantages of this scheme can be summarized as reduced design work and better bandwidth usage.

FIGURE 6. Sail of the Sailing Ship Deformation

5. Results

The results of the 3D elastic medium modeling and visualization using Mobile Agents are presented. The Mobile Agents used in the application display the improved 3D elastic model (sail of the sailing ship) on the interface, as shown in Fig. 5.

The important aspects of the Mobile Agents has been considered to be the easy implementation and the update of the model properties for realistic visualization. The 3D elastic model initial shape shown in Fig. 5 have been generated based on the 3D mass-spring approach. The final shape shown in Fig. 6 have been obtained based on the Mobile Agents approach. The graphical interface of the application is used to represent the 3D elastic model (sail of the sailing ship), shown as 3D surfaces.

The simulation of the sail of the sailing ship wind deformation was implemented for an elastic material having the next properties: elastic modulus $E = 5900000$ N/m^2, Poisson ratio $\nu = 0.47$, shear modulus $G = 2450000$ N/m^2 and a mass density of $\rho = 930$ Kg/m^3. The gravitational acceleration is considered to be $g = 9.807$ m/s^2. The sail of the sailing ship shown in Fig. 5 is consider to have 3.0 mm thickness.

Snapshots of the dynamic deformation of the 3D elastic model of sail of the sailing ship are presented in Fig. 5 and Fig. 6.

For the deformation associated with Fig. 5 not all the springs have a reasonable deformation due to the random simulating wind force. Using the Mobile Agents technology the maximum amplitude of the sail of the sailing ship located on the area with maximum wind force was extracted and updated. The final shape obtained by changing the damping coefficients and spring constants when using Mobile Agents is shown in Fig. 6. The results illustrates that improved visualization may be obtained when using Mobile Agents.

6. Conclusions

This paper presents a mobile agent system for 3D elastic medium modeling and visualization. In contrast to current modeling application systems, which suffer from slow performance and platform dependence, a structural and operational mobile agent system is introduced to achieve high performance and better flexibility. The mobile agents are used to extract and update information regarding model properties for a better representation and visualization. This illustrates that highly complicated deformable surfaces can be modeled using mobile agents technology, and improved high-quality visualization can be done for the resulting objects.

References

[1] J. Baumann, *Mobile Agents: Control Algorithms*, Lecture Notes in Computer Science, Springer, 2000.

[2] X. Yang , Y. Xing, A. Nguyen, A Web-Based Face Recognition System Using Mobile Agent Technology, *The Eighth Australian World Wide Web Conference (AusWeb02)*, (2002).

[3] J. Nolan, A. Sood, R. Simon, Agent-Based, Collaborative Image Processing In A Distributed Environment, *Proceedings of the 5th International Conference on Autonomous Agents (ACM Agents 2001)*, 228-235, (2001).

[4] Claudiu Popirlan, Cristina Popirlan, Mobile Agents communication for knowledge representation, *11th World Multi-Conference on Systemics, Cybernetics and Informatics, (WMSCI 2007)*, Vol. I, Orlando, Florida, USA, 92-96, (2007).

[5] J. Kiniry, D. Zimmerman, Special Feature: A Hands-on Look at Java Mobile Agents, *IEEE Internet Computing, Vol. 1, No.4*, 21-30, (1997).

[6] D. Metaxas, *Physics-Based Deformable Models*, Kluwer Academic Publishers, 1996.

[7] D. Terzopoulos, J.C. Platt, A.H. Barr, Elastically deformable models. *Computer Graphics*, (Proc. SIGGRAPH), **21**, 205214, (1987).

[8] D. Terzopoulos, K. Fleischer, Deformable models, *Visual Computer*, **4**, 306331, (1988).

[9] D. Breen, D. House, M. Wozny, Predicting the drape of woven cloth using interacting particles, *Computer Graphics Proceedings*, *Annual Conference Series*, **94**, 365372, (1994).

[10] B. Eberhardt, A. Weber, W. Strasser, A fast, flexible, particle-system model for cloth draping, *IEEE Computer Graphics and Applications*, **16**, 5259, (1996).

[11] M. Carignan, Y. Yang, N. M. Thalmann, D. Thalmann, Dressing animated synthetic actors with complex deformable clothes, *Computer Graphics Proceedings*, *Annual Conference Series*, **92**, 99104, (1992).

[12] D. Baraff, A. Witkin, Large steps in cloth simulation, *Computer Graphics Proceedings*, *Annual Conference Series*, **98**, 4354, (1998).

[13] X. Provot, Deformation constraints in a massspring model to describe rigid cloth behaviour, *Proceedings of Graphics Interface* **95**, 141155, (1995).

[14] D. Terzopoulos, J. Platt, K. Fleischer, Heating and melting deformable models (from goop to glop), *Graphics Interface*, 219-226, (1989).

[15] M. Dupac, D. G. Beale, R. A. Overfelt, 3D Lumped Mass/Lumped Springs Modelling and Nonlinear Behavior of a Levitated Droplet, *Nonlinear Dynamics* , **42**(1), 25-42, (2005).

[16] Claudiu Popiraln, M. Dupac, A Web-Visualization Approach for 3D Elastic Medium Modeling and Simulation, *Proceedings of the 19th IASTED International Conference on Modelling and Simulation*, Quebec City, Quebec, Canada, May, 26-28, (2008) (accepted).

[17] B. Eckel, *Thinking in Java*, Prentice Hall (4-th Edition), 2006.

[18] D. Brutzman, The virtual reality modeling language and Java, *Communications of the ACM*, **41**, 57-64, (1998).

[19] A.G. Bruzzone, G. Berrino, Modelling of urban services by VRML & Java, *Proceedings of the 1999 Int. Conf. on Web-Based Modeling & Simulation*, San Francisco, CA, 34-38, (1999).

(Claudiu Popirlan) University of Craiova Faculty of Mathematics and Computer Science, Department of Computer Science 13 Alexandru Ioan Cuza Street, Craiova, 200585, Romania
E-mail address: `popirlan@inf.ucv.ro`

(Mihai Dupac) University of Craiova Faculty of Mathematics and Computer Science, Department of Computer Science 13 Alexandru Ioan Cuza Street, Craiova, 200585, Romania
E-mail address: `mihai.dupac@inf.ucv.ro`