Reduction of Computations in Physics-Based Animation Using Level of Detail

Roman Berka

Department of Computer Science and Engineering, Faculty of El. Engineering,
Czech Technical University in Prague
Karlovo nám. 13, 121 35 Praha 2
Czech Republic
e-mail: berka@sgi.felk.cvut.cz

Abstract

A new approach based on the reduction of computations during animation of physics-based models is presented in this paper. The first part describes an approach called here *DLOD - Dynamic Level Of Detail.* Problems related to the *DLOD* are discussed in the second part and the last part deals with an experimental object oriented library using a *DLOD* mechanism.

keywords: physics-based animation, level of detail

1 Introduction

Computations of physics-based models are very time-consuming, which is a significant problem (*e.g.,* in *Virtual Reality* applications [10]), where the real time animations of natural objects are required. Since a reduction of handled parameters decreases the control quality, it is necessary to find a compromise how to reduce computation cost without the loss of nature of motion.

A large set of geometrical and physical computations is to be performed during the animation of physics-based models [5, 6, 7, 8, 9]. Some of these computations are less significant because resulting motions do not appear in an end-effect *(e.g.,* swinging hands and legs of human walk in a large distance from an observer).

One approach to the reduction of redundant computations is the use of *LOD* (*Level Of Detail* [4]) which is a method based on the reduction of computations at a level of geometry dependent on the distance between the view-point and a visualized scene. The significant problem of *LOD* is to achieve continual transitions between single levels of detail when the distance changes. One possibility is to create one instance of the model for each level but it needs high requirements for the memory size. The next approach uses algorithms based on the methods, where dynamic reductions representing the model are performed on one data-set only.

LOD is usually applied in *VR (Virtual Reality)* applications, where a large set of geometrical data is computed or transmitted. Such an application of *VR is VRML* (*Virtual Reality Modeling Language* [1, 2]).

LOD is a mechanism working on the geometry of the model. The mechanism presented in this paper focuses on the application of *LOD* to the motion of animated objects. A similar idea is mentioned in [3] as *Level Of Behavior (LOB)* which is analogous to *Level Of Detail. LOB* uses flagging certain neural pathways which are shut down, when the *Level Of Stimulus (LOS) is low.* This approach differs in the criteria determining whether the computation of the motion will be omitted or not.

2 The Background

The idea presented here is based on the physiological features of the human eye which registers some kinds of motion in different ways according to the distance (moving object-observer), the speed of the moving object and quantity of the motions. With respect to the similarity to *LOD,* this methodology is here referred to as *DLOD (Dynamic Level Of Detail).*

Since *DLOD* is connected with a few problems described below, this approach is not possible to generalize to all kinds of motion. That is why three classes of motions, which *DLOD* is available for, are demarcated:

The *first class* includes the cases, where the distance is so high that an observer cannot register the motions with a small range. Fig. 1 shows an oscillating board with a given frequency *f* and amplitude *A* (the range of motion):

$$A = 2l \sin\frac{\delta}{2}, \qquad (1)$$

where l is an approximation of the length of the oscillating part of the board and δ is an angle between two extreme positions of the end of the board. The conditions for omitting calculations can be given as

$$d > d_t \qquad (2)$$

$$d_t » A \qquad (3)$$

where d_t is a threshold value of distance d between the board and an observer.

The *second class* represents cases, where the motions are so fast that the moving objects are registered by the human eye as a blurred, transparent solid. Let Θ be a status variable describing the current position of a moving object *(e.g.,* angle between the oscillating part of the board from Fig. 1 and its equilibrium position). Then the rate of position changes is denoted as $\dot{\Theta}$ which is the time derivation of Θ. If condition $\dot{\Theta} > \omega_t$, where ω_t is the threshold value, is true then the whole space or its part deliminated the by moving object is substituted by a transparent solid and the computations of the motion are omitted. In this case the computation complexity can be reduced by such a substitution, and a more realistic motion can -be achieved.

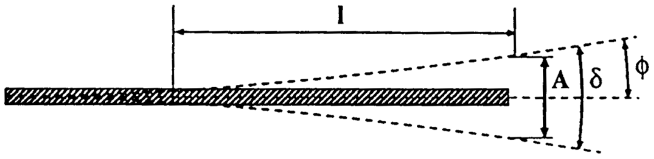

Figure 1: Simple example of an oscillating object

The *third class* includes cases, where a greater number of models is to be controlled. Since the human eye does not perceive every single detail of a large set of motions *(e.g.,* a walking crowd), these motions can be simplified. The simplification can be done by grouping the moving objects and applying a control pattern only once to the whole group. If conditions (2, 3) are applied the threshold value d_t can be decreased. The computation complexity can be rapidly reduced in cases belonging to this class.

DLOD is connected with the sort of problems which are specified in the following three questions:

1. How to eliminate *problem of background motions?*

2. How to determine motions which are to be omitted?

3. How to achieve continuous transitions of the level of detail changing the distance between an observer and a moving object?

The first problem appears when the conditions (2, 3) are true for an moving object which is attached to any other one, for which the conditions are false. Fig. 2 shows the situation when an observer watches a swinging pendulum in distance d_1 which is attached to a second pendulum by a string in distance d_2. If the second pendulum meets conditions (2, 3), computations can not be omitted because the conditions aren't true for the pendulum closer to the observer's eye.

In these cases the reduction can be applied only on periodic motions, where the behavior pattern produced by the motion generating object can be predicted. Living creatures produce a number of periodic motions, when the most typical ones are walking, running, or cycling. This could also be the answer to the second question. Since most of the locomotion patterns produced by living creatures are applied periodically, the restriction to only several types of motions is not so significant.

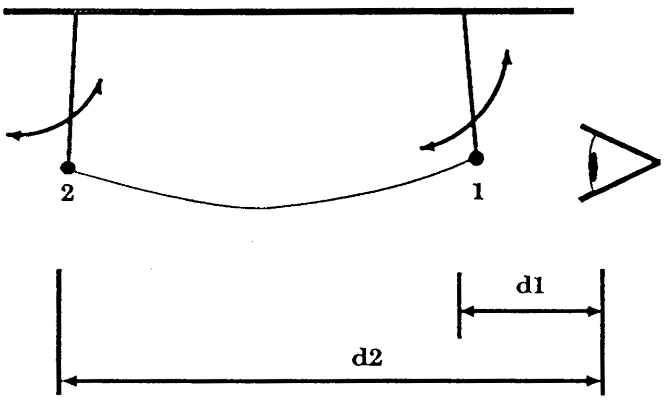

Figure 2: Locality problem

While processing a locomotion pattern simplified by *DLOD,* the changes of the distance must be detected and detailed computations are switched on. Then the process continues by evaluation of all motions. Such a method is also used when the type of the pattern changes.

The third question has no meaning in the first and second classes because for each motion there are two cases - it can be either omitted or computed (in the second case the continuous changes are given by viewing transformations). In the second class more threshold values ω_t can be used, and each of them is then assigned by the attributes of the supplying solid (e.g., the degree of transparency). As the attribute, such as a transparency, of the supplying solid changes according to the dynamic level of detail, the effect of continual 'Change of frequency is then produced.

3 Experimental Library

The approach presented in the previous sections has been implemented as a library of generic objects using C++ and Open GL. The library structure is shown in Fig. 3. All objects generated by the library can be classified into several types:

Scene Administrator is created only once in the scene. It controls all other objects via messages, and administrates graphical environment.

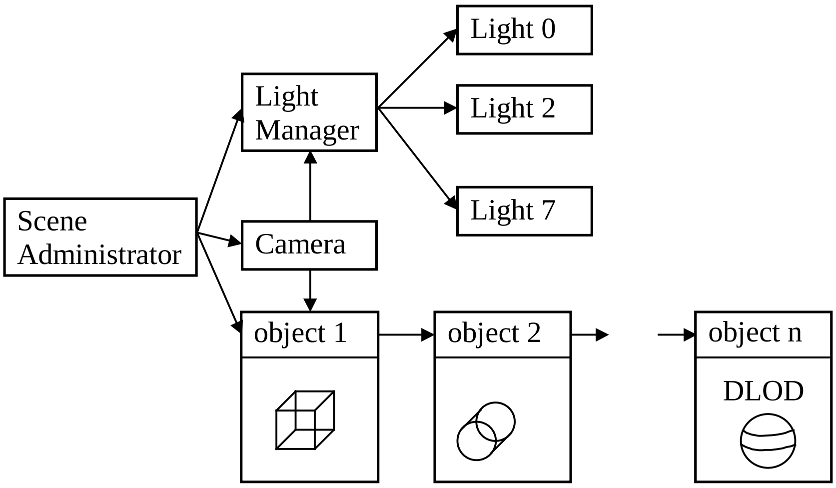

Figure 3: Library structure

Light represents one of possible types of light source (ambient, spot or directional). It can have the same attributes as the other general objects (position, color, or orientation).

Light Manager controls all light sources in the scene. The Light Manager is directly controlled by the Scene Administrator.

Camera represents a point of view (position and orientation) of an observer. This object has the same attributes as the general object.

General object has capabilities needed in animation. It contains information about position, orientation and definition of transformations. All other objects which can contain other definitions *(e.g.,* geometry) are derived from this one.

The last mentioned object is most important, since it contains the core of the *DLOD* mechanism. The computations are based on the determination of two following parameters:

1. *camera_dist* distance between the camera and a given object,
2. *motion_range* size of space demarcated by the periodical motion of a given object.

The first parameter is derived from the object using a position information from messages periodically sent by a *camera.* The *motion_range* parameter depends on a geometrical structure of the object and its computation is defined at the level of a class which is derived from a *general object.* Both parameters *camera_dist* and *motion_range* are then used for the decision, where the motion computations will be skipped or not. This choice is executed in defined time-slices using a criterion which compares the ratio of the *camera_dist* and the *motion_range* with the chosen threshold value d_t. This solution makes every object possible to use its own criteria.

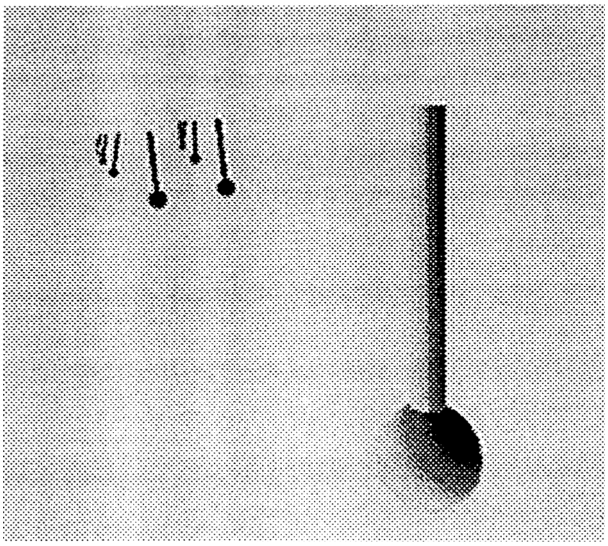

Figure 4: Application of DLOD

The implementation of the *DL OD* mechanism is now very simple, and that is why computations connected with *DLOD* represent a small overhead. A speedup strongly depends on the number of moving objects on the scene and their location. The Fig. 4 shows the scene with a set of swinging pendulae in different distances from the camera. Using the *DLOD* mechanism, 35% of motion computations were omitted. This is very significant reduction of the computations which represent very high overhead.

4 Future Work

The library presented in the previous section is the experiment which does not solve all problems mentioned in section 2. One of the problems is connected with situations where high number of motions is computed. These motions are assigned in the third class in section 2.

The reduction of computations in the third class of motions can not be solved locally at the level of objects. This is the problem of the whole structure (or its subset) and the solution requires more sophisticated mechanism of communication among all participating objects.

5 Conclusion

The approach presented in this paper is the subject of continuing research (another motivating element is the knowledge that the human eye registers motions inside the visual angle and motions outside this angle registered by peripheral vision in different ways), but it is evident that methods using *DLOD* can bring significant results in the reduction of computations. For example, it can be applied in some types of secondary motions (*e.g.,* weaving dresses [9]), whose computations are very expensive. Such a mechanism of the dynamic reductions can be used, above all, in large scenes with a high number of moving objects and in *Virtual Reality* applications. *DLOD* would be the next step to real time computer animation of physics-based models.

References

[1] G. Bell, A. Parisi, and M. Pesce. *The Virtual Reality Modeling Language, Version 1.0, Specification, 1995.*

[2] G. Bell, C. Rikk, and M. Chris. *The Virtual Reality Modeling Language, Version 2.0, Specification, 1996.*

[3] Digital Space Corporation. The Need for Nerves. http://www.digitalspace.com/nerves/backgrnd/need.html.

[4] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolutional. Analysis of Arbitrary Meshes. In *Computer Graphics Proceedings,* Los Angeles, August 1995.

[5] Radek Grzeszcuk and Demetri Terzopoulos. Automated Learning of Muscle-Actuated Locomotion Through Control Abstraction. In Com*puter Graphics Proceedings,* Los Angeles, August 1995.

[6] Hyeongseok Ko and Norman I. Badler. Animating Human Locomotion with Inverse Dynamics. *IEEE Computer Graphics and Applications,* March 1985.

[7] E. Kokkevis, D. Metaxas, and Norman I. Badler. User-Controlled Physics-Based Animation for Articulated Figures. In *Proceedings of Computer Animation '96,* Geneva, June 1996.

[8] Demetri Terzopoulos, Xiaoyuan Tu, and Radek Grzeszcuk. Artificial Fishes: Autonomous Locomotion, Perception, Behavior, and Learning in Simulated Physical World. In *Siggraph Tutorials, Los* Angeles, August 1995.

[9] Nadia M. Thalmann, Pascal Volino, Ying Yang, Shen Jianhua, and Daniel Thalmann. 3D Clothes and Fashion Show. In *Winter School of Computer Graphics and Visualization* '96, volume 1, Plzeft, February 1996.

[10] Michael J. Zyda. Networking large-scale virtual environments. In Pro*ceedings of Computer Animation* '96, Geneva, June 1996. Swiss National Institute of Technology, University of Geneva.