Chapter 1.9
Adaptive Algorithms for Intelligent Geometric Computing

M. L. Gavrilova
University of Calgary, Canada

INTRODUCTION

This chapter spans topics from such important areas as Artificial Intelligence, Computational Geometry and Biometric Technologies. The primary focus is on the proposed *Adaptive Computation Paradigm* and its applications to surface modeling and biometric processing.

Availability of much more affordable storage and high resolution image capturing devices have contributed significantly over the past few years to accumulating very large datasets of collected data (such as GIS maps, biometric samples, videos etc.). On the other hand, it also created significant challenges driven by the higher than ever volumes and the complexity of the data, that can no longer be resolved through acquisition of more memory, faster processors or optimization of existing algorithms. These developments justified the need for radically new concepts for massive data storage, processing and visualization. To address this need, the current chapter presents the original methodology based on the paradigm of the *Adaptive Geometric Computing*. The methodology enables storing complex data in a compact form, providing efficient access to it, preserving high level of details and visualizing dynamic changes in a smooth and continuous manner.

DOI: 10.4018/978-1-60960-818-7.ch1.9

The first part of the chapter discusses adaptive algorithms in real-time visualization, specifically in GIS (Geographic Information Systems) applications. Data structures such as Real-time Optimally Adaptive Mesh (ROAM) and Progressive Mesh (PM) are briefly surveyed. The adaptive method *Adaptive Spatial Memory (ASM)*, developed by R. Apu and M. Gavrilova, is then introduced. This method allows fast and efficient visualization of complex data sets representing terrains, landscapes and Digital Elevation Models (DEM). Its advantages are briefly discussed. (Figure 1)

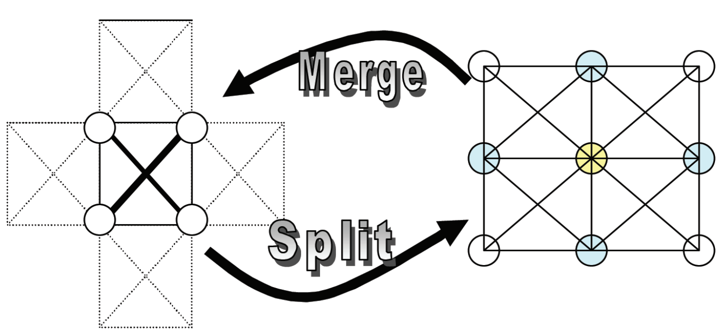
The second part of the chapter presents application of adaptive computation paradigm and evolutionary computing to missile simulation. As a result, patterns of complex behavior can be developed and analyzed. The final part of the chapter marries a concept of *adaptive computation* and *topology-based techniques* and discusses their application to challenging area of *biometric computing*.

BACKGROUND

For a long time, researchers were pressed with questions on how to model real-world objects (such as terrain, facial structure or particle system) realistically, while at the same time preserving rendering efficiency and space. As a solution, grid, mesh, TIN, Delaunay triangulation-based and other methods for model representation were developed over the last two decades. Most of these are static methods, not suitable for rendering dynamic scenes or preserving higher level of details.

In 1997, first methods for dynamic model representation: Real-time Optimally Adapting Mesh *(ROAM)* (Duchaineauy et. al., 1997, Lindstrom and Koller, 1996) and Progressive Mesh (PM) (Hoppe, 1997) were developed. Various methods have been proposed to reduce a fine mesh into an optimized representation so that the optimized mesh contains less primitives and yields maximum detail. However, this approach had two major limitations. Firstly, the cost of optimization is very expensive (several minutes to optimize one medium sized mesh). Secondly, the generated non-uniform mesh is still static. As a result, it yields poor quality when only a small part of the mesh is being observed. Thus, even with the further improvements, these methods were not capable of dealing with large amount of complex data or significantly varied level of details. They have soon were replaced by a different computational model for rendering geometric meshes (Li Sheng et. al. 2003, Shafae and Pajarola, 2003). The model employs a continuous refinement criteria based on an error metric to optimally adapt to a more accurate representation. Therefore, given a mesh representation and a small change in the viewpoint, the optimized mesh for the next viewpoint can be computed by refining the existing mesh.

Figure 1. Split and merge operations in ASM model



Related Content