CCIT Report 412 January 2003

Hierarchical Mesh Decomposition

Sagi Katz and Ayellet Tal
Department of Electrical Engineering
Technion

Abstract

Cutting up a complex object into simpler sub-objects is a fundamental problem in various disciplines. In image processing, images are segmented while in computational geometry, solid polyhedra are decomposed. In recent years, in computer graphics, polygonal meshes are decomposed into sub-meshes. In this paper we propose a novel hierarchical mesh decomposition algorithm. Our algorithm not only computes the meaningful components but also avoids over-segmentation and jaggy boundaries between components. We also demonstrate the utility of the algorithm in two applications: control-skeleton extraction and metamorphosis.

Keywords: Mesh decomposition, mesh segmentation, deformation, metamorphosis

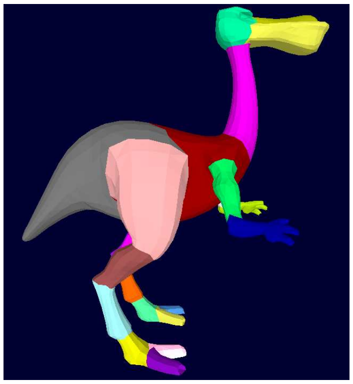

Figure 1: Decomposition of a dino-pet

1 Introduction

A hard problem might become easier if only the objects at hand could be cut up into smaller and easier to handle sub-objects. In computational geometry, solid convex decomposition, and in particular tetrahelization, has been exhaustively investigated, e.g., [Chazelle 1984; Bajaj and Dey 1992; Ruppert and Seidel 1992; Aronov and Sharir 1994; Chazelle and Palios 1994]. Similarly, in image processing, image segmentation has been considered a fundamental problem, which is a necessary pre-processing step for many higher-level computer vision algorithms [Wu and Leahy 1993; Sharon et al. 2000; Shi and Malik 2000; Gdalyahu et al. 2001]. The last few years have witnessed a growing interest in mesh decomposition for computer graphics applications [Chazelle et al. 1997; Gregory et al. 1999; Mangan and Whitaker 1999; Li et al. 2001; Shlafman et al. 2002].

In metamorphosis [Gregory et al. 1999; Zockler et al. 2000; Shlafman et al. 2002], mesh decomposition is used for establishing a correspondence. Compression [Karni and Gotsman 2000] and simplification [Zuckerberger et al. 2002] use decomposition for improving their compression rate. In 3D shape retrieval, a decomposition graph serves as a non-rigid invariant signature and decomposition must be applied automatically to large databases [Zuckerberger et al. 2002]. In collision detection, decomposition facilitates the computation of bounding-volume hierarchies [Li et al. 2001]. We believe that the spectrum of applications which will benefit from mesh decomposition will grow even more in the future. Other potential applications include modification of objects, modeling by parts and texture mapping by parts.

Several approaches have been discussed in the past for decomposing meshes. In [Chazelle and Palios 1992; Chazelle et al. 1997] a convex decomposition scheme is proposed, where a patch is called convex if it lies entirely on the boundary of its convex hull. Convex decompositions are important for applications such as collision detection. However, small concavities in the objects result with over-segmentation, which might pose a problem for other applications (i.e., metamorphosis). In [Mangan and Whitaker 1999] a watershed decomposition is described. In this case, a post-processing step resolves over-segmentation. One problem with the algorithm is the dependency on the exact triangulation of the model. Furthermore, the meaningful components, even planar ones, might get undesirably partitioned. In [Li et al. 2001], skeletonization and space sweep are used. Nice-looking results are achieved with this algorithm. However, smoothing effects might cause the disappearance of features for which it is impossible to get a decomposition. Moreover, the skeleton must be a tree, and thus loops and open meshes might pose a problem. In [Shlafman et al. 2002] a K-means based clustering algorithm is proposed. The meaningful components of the objects are found. However, the boundaries between the patches are often jagged and not always correct.

In this paper we propose a new algorithm for decomposing meshes. Our work is related to that of [Shlafman et al. 2002], but it improves upon it in several aspects: our algorithm is hierarchical, handles arbitrary meshes (regardless of their connectivity), and avoids over-segmentation and jaggy boundaries. We elaborate below.

Previous algorithms produce "flat" decompositions. As a consequence, should the number of components be refined, the whole decomposition has to be calculated from scratch. Moreover, components which belong to a refined decomposition need not necessarily be contained in components of a coarser decomposition. A main deviation of our algorithm from previous ones is being *hierarchical*.

Another deviation of the current algorithm is the way boundaries between components are handled. Previously, the focus has been on generating either meaningful components or components which comply with certain geometric properties. The boundaries between the components, however, were a by-product of the process. As a result, the boundaries were often too jagged [Chazelle et al. 1997; Mangan and Whitaker 1999; Shlafman et al. 2002] or too straight [Li et al. 2001] in a way that did not always fit the model. The current algorithm aims at avoiding jagginess, by specifically handling the boundaries.

Finally, the algorithm avoids over-segmentation and decomposes the objects into meaningful components, as illustrated in Figure 1