Computers & Graphics 33 (2009) 262–269

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Consistent segmentation of 3D models

Aleksey Golovinskiy *, Thomas Funkhouser

Department of Computer Science, Princeton University, 35 Olden Street, Princeton NJ08540, USA

ARTICLE INFO

Article history:
Received 3 December 2008
Received in revised form
27 February 2009
Accepted 3 March 2009

Keywords:
Mesh segmentation
Mesh analysis

ABSTRACT

This paper proposes a method to segment a set of models consistently. The method simultaneously segments models and creates correspondences between segments. First, a graph is constructed whose nodes represent the faces of every mesh, and whose edges connect adjacent faces within a mesh and corresponding faces in different meshes. Second, a consistent segmentation is created by clustering this graph, allowing for outlier segments that are not present in every mesh. The method is demonstrated for several classes of objects and used for two applications: symmetric segmentation and segmentation transfer.

1. Introduction

The goal of our paper is to develop a method that can produce a consistent segmentation of a set of meshes (Fig. 1b). Such a segmentation is useful for a number of applications. Parts can be labeled and put into a knowledge ontology [24], they can be interchanged as part of a modeling tool [15], and they can be put into a searchable database [27]. In addition, we can expand lexical databases such as Wordnet [6] from having part-of-relationships ("a seat is a part of a chair") to having possibly probabilistic spatial relationships ("some chairs have armrests, which are oblong, in front of backs, and above seats"). All these applications have at their heart the problem of this paper: consistently decomposing a set of 3D models into parts.

Many methods have been proposed to segment an individual mesh into parts. While these methods have produced segmentations of increasing quality, consistently segmenting a set of meshes remains challenging. Some mesh segmentation methods use heuristics designed to remain consistent between meshes (such as the shape diameter function [29]). However, segmenting meshes individually ignores important cues available from processing a whole class of objects simultaneously. While several methods [27,15] have been proposed to segment multiple objects, compared to our technique they have shortcomings, such as assuming that each mesh can be segmented individually, or that each mesh has the same number of segments.

Our approach extends the idea of single-mesh segmentations to a segmentation of multiple meshes: we simultaneously segment models and create correspondences between segments. Specifically, we first build a graph whose nodes represent faces of all the models in the set, and whose edges represent links between adjacent faces within a mesh, and between corresponding faces of different meshes. We then cluster the graph, creating a segmentation in which adjacent faces of the same model and corresponding faces between different models are encouraged to belong to the same segment.

Our approach has several advantages. First, segmenting a set of objects in a class simultaneously can produce not only more consistent results across the class, but also better individual segmentations than segmenting each object separately (as demonstrated in Fig. 1). This is because a set of objects helps to identify salient segments that are shared across the set, and because those models that have more obvious segmentation cues help to segment more difficult models. Second, modeling tools are often used to create an object via a hierarchy that is then saved in VRML and other formats, and this hierarchy can be used as "prior" segmentation to give cues to the desired segments. Our approach combines these prior segmentations with other traditional segmentation cues, such as connected mesh components, concavities, and short boundaries between components. Third, by posing the problem as that of clustering a graph representing all the meshes, our method allows for outlier segments in the resulting segmentation, such as detection of armrests only on those chairs that have them. Finally, our method handles models with disconnected components, which is not the case for many other segmentation algorithms.

We demonstrate the effectiveness of our method on several object classes. We then suggest two new applications of our method: (1) creation of a symmetry-respecting segmentation of a single model and (2) transfer of segmentations between a set of models of the same class.

2. Previous work

We review previous work in several related categories: segmentation of individual meshes, segmentation of sets of objects, consistent parameterization, and semantic labeling.

* Corresponding author.
E-mail address: agolovin@cs.princeton.edu (A. Golovinskiy).

0097-8493/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2009.03.010

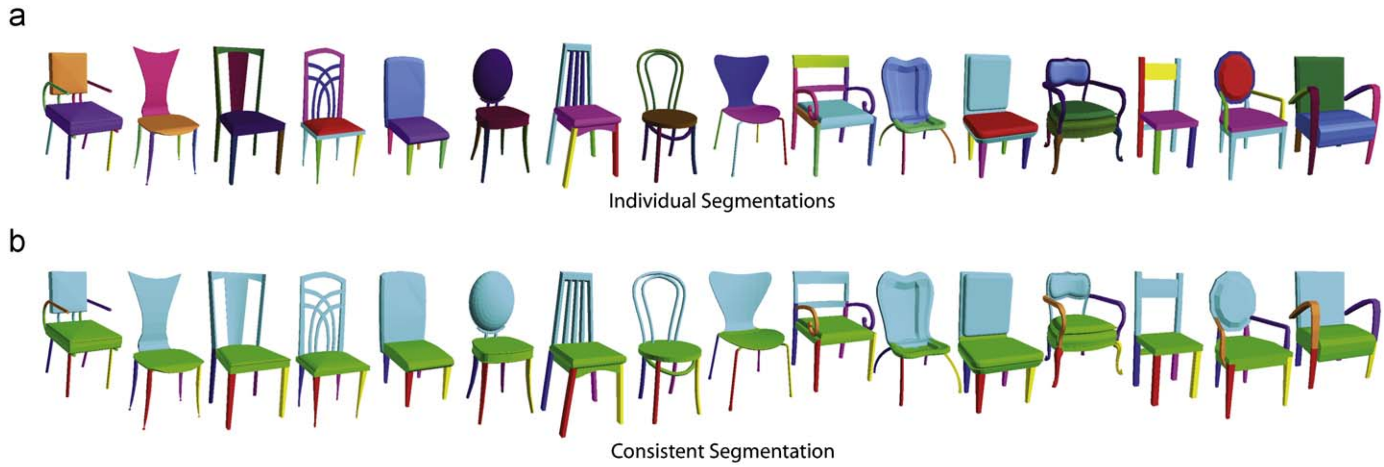

Fig. 1. Individual segmentations of a set of chairs are shown in (a). Instead, our method creates a consistent segmentation of a set of meshes (b), which allows outlier segments, such as armrest segments in those chairs that have armrests. Consistent segmentations not only bring similar object parts into correspondence, but create better part decompositions for each mesh than individual segmentations do. Note that in many individual segmentation, the backs of the chairs are either broken into several components, or aggregated with the rear legs.

Segmentation: Many mesh segmentation methods exist in the graphics literature that aim to decompose a mesh into functional parts; a recent survey can be found in [28,1], and comparisons between several algorithms appear in [3]. These approaches aim to create segments that are well formed according to some predefined low-level criteria: the segments are convex, boundaries lie along concavities, etc. They use techniques such as K-means [30], graph cuts [12], hierarchical clustering [7,8,11], random walks [16], core extraction [13], tubular primitive extraction [21], spectral clustering [19], and critical point analysis [18]. While some research tries to use segmentation criteria that are consistent between meshes in a class (such as shape diameter function [29]) or between articulated versions of a model (such as diffusion distance [5]), it is difficult in general to run segmentation methods independently on a set of meshes and obtain results with corresponding segments (Fig. 1a). We expand such segmentation techniques to simultaneously segment a set of meshes.

Segmentation of sets of objects: Several computer graphics papers have been written that involve segmentation of sets of objects. Part analogies [27], for example, segments each model into parts, and then creates a distance measure between parts that takes into account both local shape signatures and the context of the parts within a hierarchical decomposition. The main output of this is a catalog of parts with inter-part distances, which can then be used to create a consistent segmentation. We differ from this approach in that instead of first creating independent segmentations of the models and then finding correspondences between these segments, we create segments and correspondences between them simultaneously. Segmenting models and finding segment correspondences simultaneously improves segmentation quality because (i) a set of objects helps to identify salient segments that are shared across the set and (ii) those models that have more obvious segmentation cues help to segment more difficult models.

Another related work is Shuffler [15], a modeling tool that allows the user to swap a segment in one mesh for a segment in another. Shuffler creates a mutually consistent segmentation between a pair of meshes, in which the segments are in one-to-one correspondence. We differ from their approach in two ways. First, while Shuffler's method could in principle be extended to segmenting a set of objects, their work demonstrates only pairwise segmentations. Second, we do not assume that all models contain all parts but instead allow outlier segments, making the problem more difficult, but allowing richer decompositions.

Some papers in the computer vision literature aim to obtain a segmentation of the same object or similar objects in different images: [17] created an implicit shape model for segmenting a class of objects from examples, [25] used a generative graphical model to improve segmentation by using two images rather than one, and [31] used spectral clustering to find correspondences and matching segments. The last method constructs a joint-image graph that encodes intra-image similarities as well as inter-image correspondence, which is similar to the graph we construct. Our method extends this idea to the 3D model domain, where the challenges lie not in disconnecting foreground objects from a cluttered background, but in using geometry to create precise segmentations.

Consistent parameterization: Another approach to creating a consistent segmentation may be to create a continuous mapping between the objects, segment one of them, and transfer the segmentation to the others. A large body of work exists that deals with consistently parameterizing a set of meshes. Some papers offer general methods for establishing a mapping between surfaces (e.g., [22,14,26]) while others focus on consistent parameterization for a particular class of objects such as faces [4] and bodies [2]. Once such mapping is created, surface attributes such as texture coordinates or displacement maps can be transferred. Unfortunately, many real-world object classes have a large degree of intra-class variation and inconsistent mesh quality and topology, making it difficult or impossible to find a single consistent parameterization across the entire class, unless the meshes are very similar (such as faces [4] or human bodies [2]). For example, there is no continuous global mapping from a chair with armrests to one without them. In this paper, we propose a method to find consistent segmentations without first finding a continuous parameterization.

Semantic labeling: Several papers address the problem of assigning semantic labels to meshes. ShapeAnnotator [24] describes an effort to put 3D models into a knowledge base, and consistently annotate their parts. They do not provide an automatic algorithm to do so, instead offering a tool that presents the output of several automatic segmentation algorithms such as those described above, and allows the user to select sections from each output to form segments and annotate them. Such efforts would benefit from a tool such as ours that creates a consistent segmentation of meshes automatically.