Multi-Segmentation and Annotation of 3D Surface Meshes

Marco Attene, Silvia Biasotti, Michela Mortara, Giuseppe Patané, Francesco Robbiano, Riccardo Albertoni, Chiara Catalano, Simone Marini, Michela Spagnuolo and Bianca Falcidieno

Rapporto Tecnico N. 6/2006
Genova, Agosto 2006.

Abstract

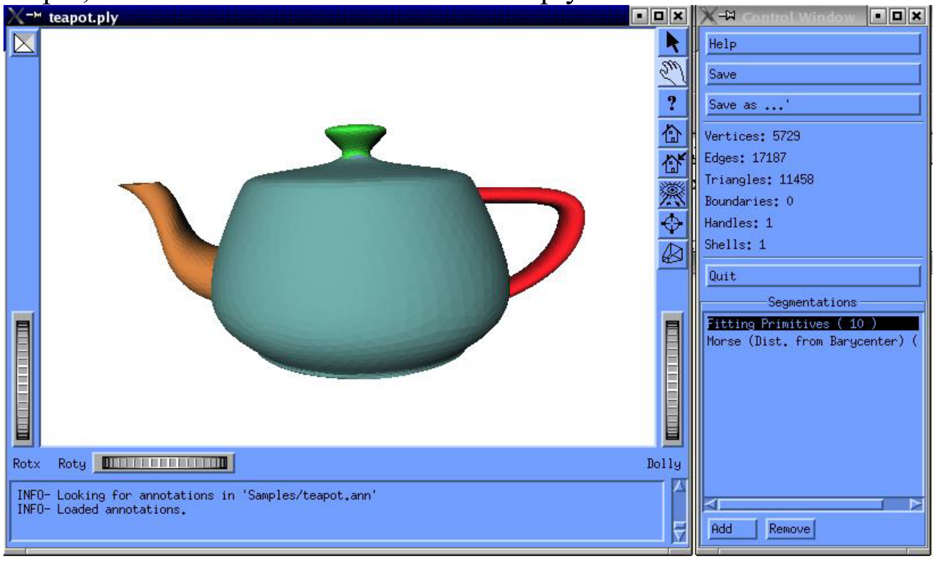
The ShapeAnnotator is a graphical tool designed to assist an expert user in the task of annotating a surface mesh with concepts belonging to a domain of expertise.

The user is first required to identify meaningful surface features; Each such feature can then be "labeled" with a concept, and the surface mesh, along with the features identified and their labels, can be saved.

Feature identification

Here, a surface feature is defined to be a connected subset of the mesh faces. For the purpose of identifying surface features, the tool provides a set of mesh *segmentation* algorithms. Each such algorithm produces a partition of the mesh faces into *surface regions*. Each region may be picked by the user through simple mouse clicks, and added to the set of interesting *features* to be annotated.

An apposite window shows the surface mesh colored according to the currently selected features which, in this case, may overlap and do not necessarily cover the entire surface (i.e., differently from a single segmentation, the whole set of selected features do not necessarily form a partition of the mesh faces). Within this window, each feature can be modified through morphological operators; these include the growth, the shrinkage, the removal or the creation of a segment, the union of two segments into a single one, and so on. All such operations can be performed on the mesh interactively through mouse clicks, and the ShapeAnnotator takes care of avoiding undesirable configurations such as, for example, the creation of disconnected or empty features.



```
Vertices: 5729
Edges: 17187
Triangles: 11458
Boundaries: 0
Handles: 1
Shells: 1
```

```
Fitting Primitives ( 10 )
Morse (Dist. from Barycenter) (
```

```
INFO- Looking for annotations in 'Samples/teapot.ann'
INFO- Loaded annotations.
```

Figure 1: A segmention of the teapot model.

In order to provide enough flexibility, the ShapeAnnotator has a plugin-based architecture. Specifically, segmentation algorithms are not hard-coded within its executable; instead, they are implemented separately, compiled as plugins and loaded at startup by the main executable. In this way, if the domain of knowledge requires a particular feature to be identified, a proper segmentation method can be implemented and used through the ShapeAnnotator.

Domain of knowledge

During the actual annotation step, the user picks the desired concepts from a given domain of knowledge which, in this version, is formalized as an ontology represented through an xml-rdf file.

The ShapeAnnotator parser converts the rdf file into an internal set of triplets of the type (subject, predicate, object). Concepts representing predicates (i.e, relations or attributes) cannot be used to tag

surface features, while they are useful to drive the user during the navigation of the ontology. The rdf file must have a reference to the concept "Class"; such a concept will be used as the starting point to browse the ontology from scratch.
From now on, we say that two concepts are neighbors of each other if there is a *triplet* for which the two concepts are *subject* and *object* respectively.

Annotation
Once an interesting feature is identified (thus, after running at least one segmentation algorithm and, possibly, some morphological operations), the user may want to annotate it. To do so, it is necessary to click on the feature and to select "Add annotation" from the popup menu displayed. A new window is opened in which the concept "Class" is shown in the center and all its neighbors are shown around it (see Figure 2). If the user clicks on a concept, this becomes the new center, and its direct neighbors are shown around it. When (and if) the user finds a concept that properly describes

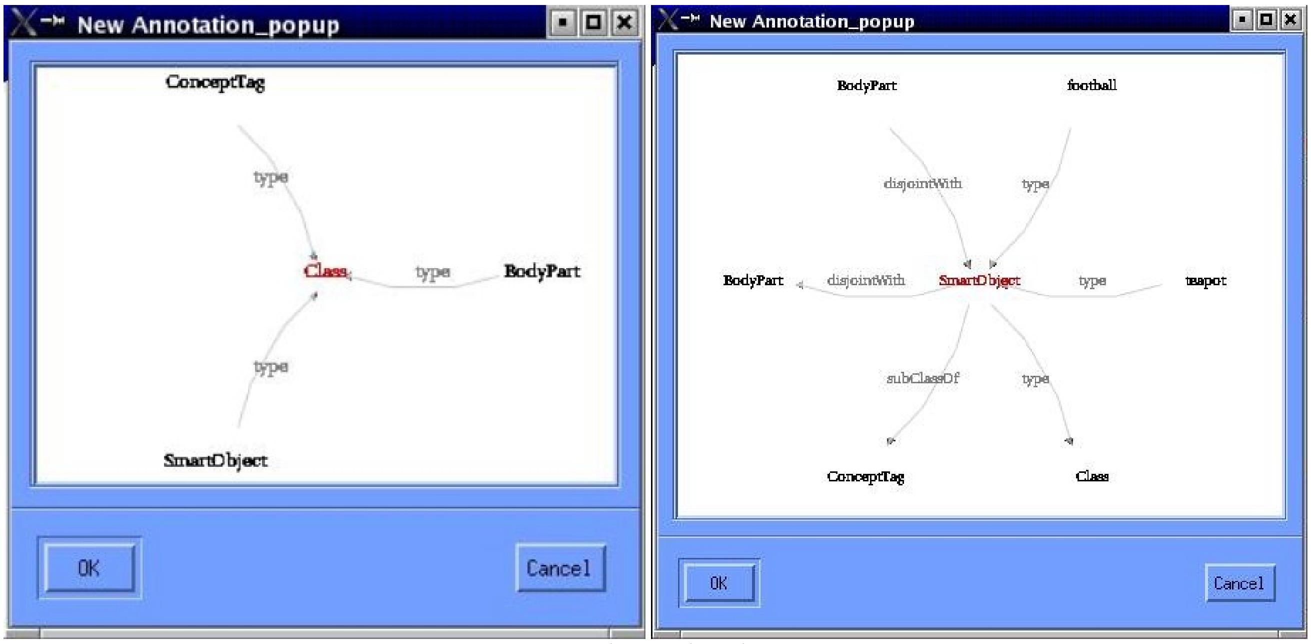


Figure 2: The ontology browser.

the feature, this can become its annotation by simply clicking on the OK button.

If a feature is already annotated, it is possible to remove the existing annotation or to modify it. In the latter case, the ontology browser starts from the existing concept, and not from the more generic "Class".

Startup
When launched, the ShapeAnnotator executable scans the directory *Plugins* and looks for files with extension ".rpd". Each such file is expected to be a text file describing a *mesh segmentation* algorithm.
Also, the executable looks for a file called "default_ontology.rdf" in the directory "Ontology". This file is parsed to create the internal representation of the ontology graph used to annotate the mesh.
Finally, the desired mesh is loaded (its filename is specified as a command-line argument) and shown by the tool.

Annotation process
On the left hand side of the GUI the user finds an area containing the set of segmentations currently "attached" to the mesh, each along with the parameters used by the algorithm to generate the

segmentation. Clearly, at the beginning no segmentation is shown. The button “Add” below the segmentation list can be used to add a new segmentation. When clicked, a popup menu is displayed in which all the segmentation algorithms loaded as plugins are shown. After picking one of them, the user sets its parameters and runs the algorithm. When done, a preview of the segmentation is displayed; if the user finds it useful (i.e., it contains some interesting features to annotate) he/she may add it to the list of segmentations. Each segmentation can be displayed by clicking on its name in the list. Once displayed, the user can click on the regions representing interesting features to add them to the set of features to annotate shown in the apposite window. Each selected feature can be both edited through the morphological operators (popup menu displayed by left-clicking on the feature itself), and annotated (popup menu by right-clicking).

File Formats

Accepted file formats for the input mesh are (subsets of): VRML (1.0 and 2.0), OpenInventor IV 2.1, OFF, PLY, OBJ and IMATI's Ver-Tri.

The ontology representing the domain knowledge must be a single well-formed xml-rdf file.

The segmented mesh is saved in ascii PLY format, while possible annotations are saved in an additional xml file with the same name as the mesh file but extension *.ann.

An annotated mesh saved as above can be loaded along with its segmentations and annotations for further processing.