3D Mesh Exploration for Smart Visual Interfaces

Georg Fuchs, Mathias Holst, René Rosenbaum, and Heidrun Schumann

Institute for Computer Science, University of Rostock

Abstract. Today's fast growth of both the number and complexity of digital 3D models, used for a variety of purposes such as system simulation, product presentations or technician training, results in a number of research challenges. Amongst others, the efficient presentation of, and interaction with, such complex models is essential. It therefore has become more and more important to provide the user with a smart visual interface that presents all the information required in the context of the task at hand in a comprehensive way. In this paper, we present a two-stage concept for the task-oriented exploration of 3D polygonal meshes. An authoring tool uses a combination of automatic mesh segmentation and manual enrichment of segments with semantic information that allow the association with specified exploration goals. This information is then used at runtime to adapt the model's presentation to the task at hand. The exploration of the enriched model can further be supported by interactive tools. The use of a 3D lens is discussed as an example.

1 Motivation

Today, more and more processes related to product development, production and servicing are implemented almost exclusively by means of the CAx software family. This results in a fast growth of both the number, as well as the complexity and fidelity, of digital 3D models that are used for a variety of purposes, such as system simulation, product presentations, or technician training.

This trend results in a number of research challenges [1, 2]. Besides questions of how to create and (re)use models or how to preserve semantics across the whole modeling process, the efficient presentation of, and interaction with, such complex models is essential. In other words, it has become more and more important to provide the user with a *Smart Visual Interface* that presents all the information required in the context of the task at hand in a comprehensive way. This includes support of *Mesh Exploration* through such interfaces. That is, to support the user in the process of understanding or verifying the structure and functionality of an object (modeled as a polygonal mesh) within the context of the task at hand. Enriching a model with additional information helps to communicate complex spatial relationships and to identify individual object components, for example, parts of machinery or organs in medical data. Especially, such information can be used to adapt the presentation according to a user-specified exploration goal [3], e.g. by accentuating important components and hiding unimportant ones. Offering interactive exploration further allows to discard the need for predefined animations.

Our definition of a smart visual interface is therefore based on two main aspects: (i) a description of the task that is performed, the *task model*, and (ii) the use of this model for *adaptive representation* of the 3D model for the task at hand.

2 Background & General Approach

The background for our investigations is a mobile maintenance support scenario in the scope of an industry research project [4]. Tasks associated with repair and maintenance work typically follow laid down procedures, which lend themselves to explicit notation in the form of a task model. Generally, a task model forms a hierarchy of compound tasks with subtasks. Subtasks have conditional and/or temporal relationships between them, representing the step-by-step nature of individual working steps. Thus, we proposed a concept of using task models in visual interfaces, as the core of an 'e-Manual' application that presents the user with documentation required for the completion of her current maintenance task.

[4] mainly considered the presentation of visual content that is most common to technical manuals: two-dimensional illustrations and schematics, especially digitized raster images from printed documentation. The challenge with this type of content is the absence of any structure or semantic information about the depicted object: the image is but a 'pixel soup'. In order to be able to highlight task-specific aspects of such an illustration, appropriate Regions of Interest (RoI) have to be defined and attributed with meta data beforehand. RoIs are then assigned a Degree of Interest (DoI) with respect to a given task. These information are finally used to adapt the presentation accordingly. This can for example be the use of Focus & Context distortion, or the accentuation of important regions through hue and saturation adjustments. Fig. 1 gives an illustrative example for this kind of task-based adaptation.

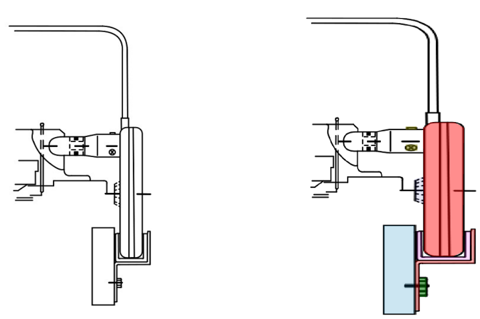

Fig. 1. Example for the task-based adaptation of a 2D raster image (left). Important image regions have been enlarged and accentuated by color adjustment (right).

Extending the maintenance e-Manual to also include 3D representations would increase the learning efficiency. 3D models generally support the interactive exploration of complex spatial configurations much better than 2D illustrations. For example, the 3D representation of a piece of machinery can help to identify and locate an attachment part. 3D modeling software usually employ hierarchical structures (scene graphs) that represent a segmentation into independent components. However, such model hierarchies and/or its associated meta data may not be available due to limitations of the data format, intellectual property protection considerations, or simply because no meta data was generated for the model components in the first place. Analogous to raster graphics, such 3D models mostly are but a set of triangles without semantics, a 'triangle soup'. Therefore in the 3D case, defining RoIs becomes a task of first segmenting the triangle mesh into meaningful components. Task-based adaptation of the representation could then include the automatic choice of a suitable initial viewpoint, the component-wise application of Level-of-Detail (LoD) techniques, highlighting important parts, and the omission or transparent representation of occluding geometry.

In this paper, we therefore concentrate on integrating task models with 3D representations to obtain a task-oriented visual output of complex triangle meshes. This imposes two additional requirements to the modeling process: (i) enriching the 3D model with semantics, and (ii) associating aspects of the model with the task at hand. These crucial stages of the process can not be performed solely by automatic means. While segmentation algorithms form the basis for a semantic decomposition of the triangle mesh, an interactive inspection step is needed nonetheless to ensure that components of the model can be appropriately associated with the (sub-)task at hand. We will show how this inspection can be supported by an appropriate exploration viewer and interactive 3D lenses.

The following Sect. 3 gives an overview of related work as well as references to our previous work on task-oriented 2D representations. The general approach is discussed in Sect. 4. Section 5 gives a summary and some concluding remarks.

3 Related work

The need to enhance the raw geometry with additional high-level shape information is an important topic in both industry and academia. Thus results the requirement to decompose a model into meaningful components. To this end, a number of different segmentation methods has been developed [2, 5, 6]. For a good recent overview, see [7].

The segmentation of a surface mesh can be carried out either according to purely geometric aspects [8, 9] or semantic-oriented [10–16]. In the former case the mesh is decomposed into patches that are equal with respect to a certain property (curvature, distance to reference plane) whereas latter methods try to identify object parts that correspond to relevant features of the object's shape. Geometry-based approaches can also be used as a pre-process to the detection of relevant features. Semantic-oriented approaches have enjoyed increasing attention in recent research because they support morphing, 3D surface reconstruction, skeleton extraction well as the modeling by composition paradigm which is based on the natural decomposition of surfaces. Yet the majority of approaches in computer graphics were not designed to detect features in a specific context such as product modeling in the industrial production process [17]. Generally, different algorithm perform better for certain types of model features. For this reason, Attene et al. [18] propose what they call *multi-segmentation*, i.e. to use several segmentation algorithms in parallel and to mix-and-match their individual segmentation results.

Our approach is based on the *feature point & core extraction* approach presented in [12] that aims at creating pose- and proportion-invariant segmentations. It works in three steps: transformation of the mesh vertices into a pose-invariant MDS (multi-dimensional scaling) representation, robust extraction of prominent feature points from the MDS domain, and finally the extraction of the object's core component. The latter is done by mirroring to the transformed mesh vertices on the object's bounding sphere and subsequent vertex classification. The algorithm achieves consistent segmentations of similar models that differ only in pose or component proportions. It also adheres to the minima rule [19] by detecting limbs and protrusions that can be visually separated along local concavities from the object's core. This makes the method suitable for a wide range of both natural (bodies with attached limbs) and technical (machinery with attachment parts) models.

Besides segmentation methods for the semantic decomposition of triangle meshes, the efficient presentation of these complex models for various application purposes is another important research topic. Interactive tutoring systems need to communicate specific aspects of a complex model (e.g. [3]). A primary question therefore is how to accentuate relevant features while minimizing the visual impact of irrelevant or occluding objects. Non-photorealistic rendering styles [20] are a good way to highlight important shape features. Transparency [21, 22] and cutaway views [22, 23] are a very efficient static method to remove or subdue irrelevant occluding geometry. Some interactive tutoring systems automatically determine optimal views which minimize occlusions [24] or plan camera movements in order to guide the interactive exploration [25]. Dynamic labeling is a method to establish co-referential relations between textual and visual elements [26].

In addition to the above static accentuation/de-accentuation methods, so-called *lens techniques* can be used to dynamically adapt the presentation in a user-defined Region of Interest. Magic Lenses were first introduced by Bier et al. [27, 28] as a focus & context technique for 2D interfaces. A Magic Lens is a movable, semi-transparent user interface element that changes the representation of data viewed through it. Generally, lenses can be used not only for magnification but also numerous other effects, such as filtering (remove objects under lens to reduce clutter, e.g. [29]), adaptation of the presentation (e.g. a change of rendering style) and Level-of-Detail (data through the lens is rendered at a higher resolution, e.g. [30]). Several lenses can be combined to produce composite effects where they intersect.

The Magic Lens metaphor was extended to three dimensions by Viega et al. [31]. They implemented two types of 3D lenses: a 'flat' lens that projected a volume of influence into the scene, and a volumetric lens that affected content falling within the space of a cube. Both approaches exploited hardware support for clipping planes which made it possible to divide the scene into 'lensed' and 'un-lensed' spaces in real-time. The use of 3D lenses as interactive tools for exploration and interaction with virtual 3D scenes also attracted more recent research in the domain of augmented reality applications [32].

To summarize, there exist a number of segmentation algorithms, semantic-oriented approaches in particular, that can form the basis for the annotation of a 3D model, as well as rendering methods for their efficient presentation in the context of the task at hand. There are approaches [18, 33] similar to ours. These however concentrate on ontology browsing for shape annotation and retrieval, especially [18] having a strong emphasis on the exploitation of multi-segmentation. For our purposes, the choice of a semantic-oriented algorithm with good overall applicability for decomposing arbitrary models into meaningful components, which can be associated with (sub-)tasks from the task model, is sufficient as a starting point. We therefore chose the feature point & core extraction algorithm [12] to illustrate our approach.

4 Concept

The task-oriented approach can be broken down into two stages: the model authoring phase (including inspection), and rendering the model in the *Exploration Viewer* as part of the smart visual interface, using the task model for adaptation of the presentation (Fig. 2).

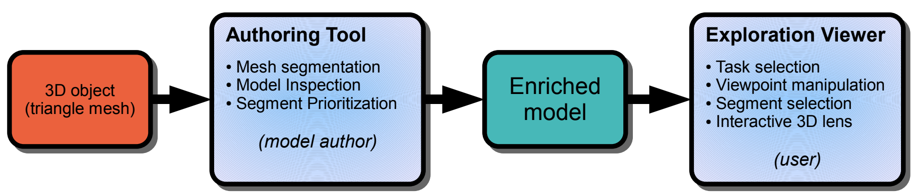

Fig. 2. Overview of the task-based model exploration process.

The following Sect. 4.1 discusses the individual steps of the authoring process. The adaptation of the enriched model representation for a given task and the use of interactive 3D lenses is described in Sect. 4.2.

4.1 Model Authoring

The authoring phase can be broken down into three subsequent operations that are performed on the model to obtain a suitable model enrichment:

1. Mesh Segmentation
2. Model Inspection
3. Segment Prioritization

Mesh Segmentation is the first step in the authoring process. As a preprocessing step to the actual segmentation, a mesh simplification is performed to build a hierarchy with decreasing LoD. This not only improves performance of the segmentation algorithm, it also reduces the susceptibility of automatic algorithms for small perturbances of the surface, such as scanning artifacts. Moreover, it later allows to select different LoDs per segment to adapt the visual representation (cf. Sect. 4.2).

The model author starts by selecting the LoD in the hierarchy at which the segmentation should be performed using a slightly modified feature point & core extraction algorithm. In the original approach [12], the object core was extracted by mirroring the mesh vertices on the bounding sphere. Any vertex that maps to the convex hull of the mirrored vertex set is classified as belonging to the core. In order to improve the segmentation results for arbitrary objects that may exhibit bumpy surfaces or scanning artifacts, we introduced an additional parameter to classify vertices with a maximal hull distance $d \geq 0$ as core vertices as well. Also, unlike proposed in [12], our prototype does not yet apply a MDS transformation of the mesh vertices before feature point detection. However, for rigid technical models that are not posed the results are quite acceptable.

As discussed in Sect. 3, the performance of automatic algorithms generally depends strongly on the model geometry and its prevalent features. Using multi-segmentation [18] is one way to ensure meaningful results. However, our approach is to iteratively refine the initial results with the help of interactive inspection of the partially segmented model instead. This ensures that besides geometric features, also application-specific features can be taken into account. An technical domain example are weld seams in the concavities between model segments that would hardly be detected as independent segments by automatic methods.

Model Inspection During model inspection, the author can verify whether the detected segments match the desired result. To this end, the model is rendered with the segment patches assigned random colors. The author can manipulate the 3D view (zoom, rotate) to inspect each segment patch. Individual segments can be selected, either by picking them in the 3D view or from a list control (cf. Fig. 4), and sub-segmented further by iteratively applying the segmentation algorithm to the associated triangle patch.

Another option during the inspection phase is to manually segment parts of the model. Manual segmentation offers an alternative way to overcome algorithm limitations with great flexibility as it does not rely on its suitability for a given geometry. By interactively placing simple shapes, such as spheres or cubes, in the 3D scene the author can quickly define Regions of Interest (Fig. 3a). Parts of the mesh intersecting these volumes can be assigned to a new segment, or added to an existing one.

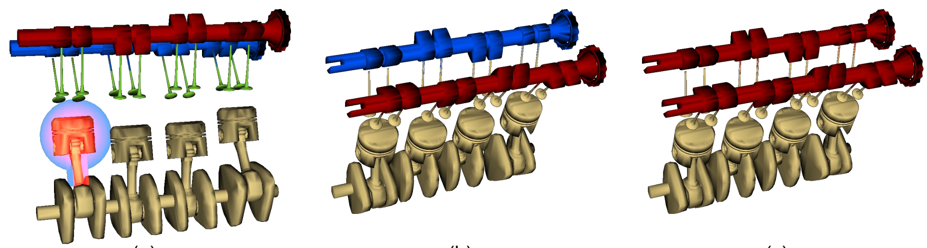
(a) (b) (c)

Fig. 3. Definition of 3D RoIs of an object to obtain a first coarse segmentation (a). Refinement is possible by selecting and further subdividing segments, or by merging multiple segments (b, c).

The resulting segmentation of the current iteration can be refined even further by three additional operations:

- expanding a segment by adding a triangle strip at its boundary,
- shrinking a segment by removing a triangle strip from its boundary, and
- merging two segments into one (Fig. 3b-c).

Expanding and shrinking segments can also be applied to obtain a disjunctive, exhaustive segmentation of the entire model, if so desired. This is done by expanding patches into mesh areas yet unassigned to any segment, and shrinking overlapping segments in the respective areas. Note that the final segmentation does neither need to be disjunctive – triangles may be associated to more than one segment with fuzzy borders – nor exhaustive if some regions of the surface do not belong to any *meaningful model component*, i.e. relevant to at least one task.

Finally, after the segmentation for the selected LoD has been completed, the result is mapped onto all other levels of the model hierarchy. The surface mesh can therefore be presented with variable LoD in the exploration viewer.

Segment Prioritization After the desired segmentation has been achieved that reflects the meaningful model components, the author can annotate them with semantic information. Basis of the annotation process is the task model for a given maintenance task (cf. [4]).

Therefore, our authoring tool allows to associate, for every task, each segment a priority or *importance* normalized to the [0..1] value range. Therefore, importance values express the Degree of Interest with respect to the task at hand and are therefore stored in the nodes of the task model. The default importance value is 0, i.e. the component is irrelevant, while segments with assigned values > 0 constitute meaningful model components in the context of the task at hand.

Optionally, segments can also be associated with basic rendering attributes (color, material, transparency) and style (e.g. contour sketches, wire frame, translucent, solid). This can be used to enforce visual grouping of related components, for example by color-coding functional groups of components. Also, the author can define a good initial viewpoint on the model for each task.

Fig. 4 shows a screenshot of the authoring tool with the inspection 3D view and the task pane for importance value adjustment.

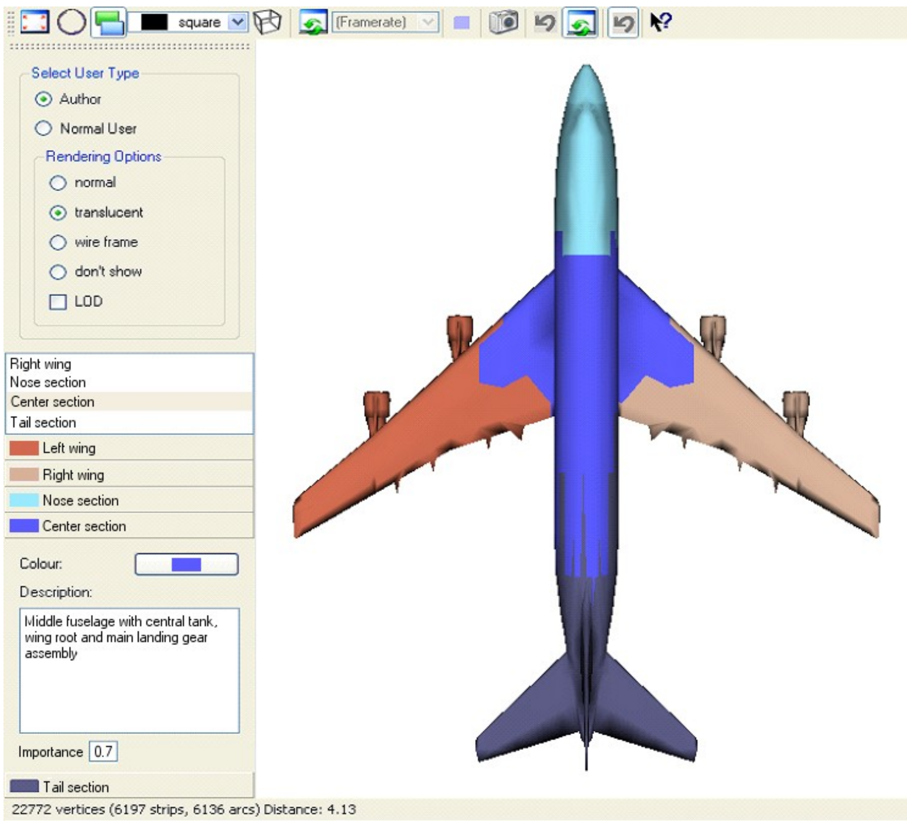

Fig. 4. Screenshot of the authoring tool prototype. The 3D inspection view shows the current segmentation of the model. The control panel on the left is used to set the segment importance value and other properties for the currently selected task.

4.2 Exploration Viewer

The exploration viewer as part of the smart visual interface renders the model and allows to interactively explore it. The viewer automatically creates an *initial view* of the 3D model according to the information from the task model. The importance value thereby is the primary mechanism to adapt the presentation of the model to the task at hand. It is used to control the following aspects:

Initial viewpoint selection: For an optimal view on the 3D representation for a given task, the virtual camera should be positioned so that at least the most important model components are initially visible. This viewpoint is either stored in the associated task model node, or the importance values from the task model can be used to control appropriate viewfinding algorithms like [34].

Level of Detail selection: Components with the maximum importance use the highest-resolution base mesh, whereas less relevant components are rendered at correspondingly lower LoD. Irrelevant segments (zero importance) are rendered at the lowest LoD, or are even culled from the scene, i.e. not rendered at all.

Accentuation: To efficiently convey the important aspects of the task at hand, the attention of the user should be guided to the relevant components. This is achieved by accentuating important model components and visually subduing (de-accentuating) unimportant ones according to the values from the task model. There is a vast number of methods for accentuation (cf. Sect. 3). Examples include the simple adjustment of material properties such as hue, saturation or the alpha value (Fig. 5a), the selection of rendering styles (Fig. 5b) including NPR techniques [20], and the creation of cutaway views [23].

Moreover, the task model can provide additional information that may be used in conjunction to the enriched model description. Two examples are

- to provide names or short designations that can be used to dynamically label segments (our labeling approach is described in [35]), and
- to provide a textual description of work activities on, or additional information about the component(s). These can be displayed as text juxtaposed to the 3D view, or communicated via Text-to-Speech output in multi-modal interfaces, as discussed in [4].

As these information reside in the task model they can be collected and maintained independently from the model authoring process.

However, while the above measures generate appropriate initial views, a single illustration often does not suffice to depict all salient visual objects due to (partial) occlusions. Therefore, the ability to interactively *explore* the model is still important.

The exploration viewer supports this in a number of ways. Besides the obvious view manipulation, it is always possible to select and thus focus on an arbitrary model component. This overrides the importance value for the associated segment from the task model and the visual representation is updated accordingly (Fig. 5c). Like in the authoring tool, component selection is realized either through picking in the 3D view or through selecting the associated list entry (cf. Fig. 4).

Another tool for exploring the model in a flexible way is the use of interactive 3D lenses that can be positioned freely in 3D space (Fig. 6). In doing so, the user defines a dynamic RoI in which rendering parameters are changed independently

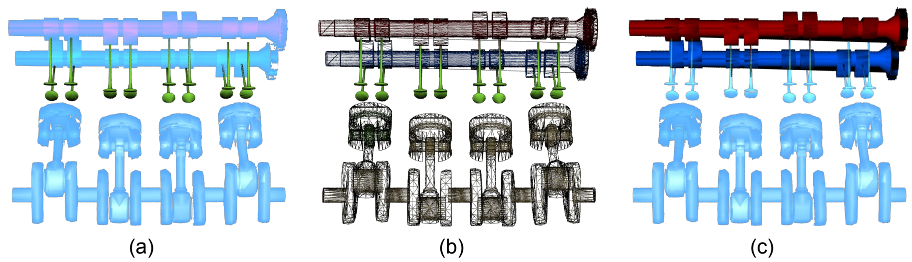

Fig. 5. Example for the application of different rendering styles for segment accentuation (a, b). The valves are defined as the most important component for the current task, other components are visually subdued. (c) shows the result of manually selecting two other segments (both camshafts) as the focus, overriding the importance values from the task model.

of segment boundaries. In our prototype, we so far employ only a single lens that changes the LoD of the mesh. This allows a user to view those parts of the model in full fidelity even if their predefined relevance for the task at hand is minimal. However, other lens functions such as switching the rendering style or an 'X-ray' effect to interactively reduce occlusion could be realized as well.

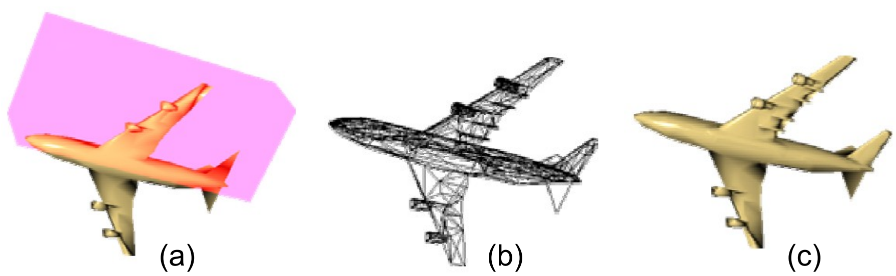

Fig. 6. Applying a 3D lens to manipulate the level of detail in selected object regions. (a) Lens volume, (b) resulting hybrid LoD mesh, (c) visual result.

5 Summary

In this paper we presented a concept for the task-oriented exploration of 3D polygonal meshes. It uses a task model as well as a combination of automatic mesh segmentation and manual enrichment of segments. The association of model components with importance values in the task model is a versatile means for the adaptation of the presentation to the task at hand. The exploration of the enriched model can further be supported by interactive tools such as 3D lenses.

The concept has been prototypically implemented using C++, OpenGL and Qt. The authoring tool supports hierarchical mesh simplification, automatic and manual segmentation and segmentation prioritization. Automatic segmentation algorithms are integrated as plug-ins. Currently, only a slightly modified feature point & core extraction algorithm [12] is available. The exploration viewer so far supports the importance-driven selection of the segment's LoD and its rendering style (culled, wire frame, translucent, darkened solid, solid).

Future work will go along several lines. The automatic segmentation capabilities could be improved by implementing additional algorithm plug-ins. Good candidates for technical models are Plumber [15] and Hierarchical fitting primitives [10]. Further improvements to the exploration viewer could include support for importance-driven NPR (e.g. from line sketches to stippling to fully shaded), improved viewpoint selection, and additional lens effects for interactive exploration. We further strive to evaluate the proposed concept and planned extensions with real-world data in cooperation with our industry partner within the research project [4]. The ECS International Ltd. is a supplier of software solutions for Product Data Management (PDM) in engineering, where interaction with and inspection of complex technical 3D models features prominently.

References

1. Mayer, J.: The New Main Research Areas of Fraunhofer IGD. Computer Graphics topics **19**(2) (2007) 18–19
2. Attene, M., Biasotti, S., Falcidieno, B., Mortara, M., Patanè, G., Spagnuolo, M.: Topological, geometric and structural approaches to enhance shape information. In: Proceedings of Eurographics Italian Chapter, Catania (I) (22-24 Feb 2006) pp. 7–13
3. Preim, B., Ritter, F.: Techniken zur Hervorhebung von Objekten in medizinischen 3d-Visualisierungen. In: Simulation und Visualisierung. (2002) 187–200
4. Fuchs, G., Reichart, D., Schumann, H., Forbrig, P.: Maintenance Support – Case Study for a Multimodal Mobile User Interface. In: IS&T/SPIE's 16th Ann. Symp. Electronic Imaging: Multimedia on Mobile Devices II. (2006)
5. Chazelle, B., Dobkin, D.P., Shouraboura, N., Tal, A.: Strategies for polyhedral surface decomposition: an experimental study. In: SCG '95: Proceedings of the eleventh annual symposium on Computational geometry, New York, NY, USA, ACM (1995) 297–305
6. Li, X., Toon, T.W., Huang, Z.: Decomposing polygon meshes for interactive applications. In: I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics, New York, NY, USA, ACM (2001) 35–42
7. Attene, M., Katz, S., Mortara, M., Patanè, G., Spagnuolo, M., Tal, A.: Mesh segmentation - a comparative study. IEEE International Conference on Shape Modeling and Applications (SMI'06) **0** (2006) 7
8. Garland, M., Willmott, A., Heckbert, P.S.: Hierarchical face clustering on polygonal surfaces. In: I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics, New York, NY, USA, ACM (2001) 49–58
9. Mangan, A.P., Whitaker, R.T.: Partitioning 3d surface meshes using watershed segmentation. IEEE Transactions on Visualization and Computer Graphics **5**(4) (1999) 308–321

10. Attene, M., Falcidieno, B., Spagnuolo, M.: Hierarchical mesh segmentation based on fitting primitives. Vis. Comput. **22**(3) (2006) 181–193
11. Katz, S., Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Trans. Graph. **22**(3) (2003) 954–961
12. Katz, S., Leifman, G., Tal, A.: Mesh segmentation using feature point and core extraction. The Visual Computer **21**(8–10) (September 2005) 649–658
13. Liu, R., Zhang, H.: Segmentation of 3d meshes through spectral clustering. In: PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference, Washington, DC, USA, IEEE Computer Society (2004) 298–305
14. Mortara, M., Patanè, G., Spagnuolo, M., Falcidieno, B., Rossignac, J.: Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. Algorithmica **38**(1) (2003) 227–248
15. Mortara, M., Patanè, G., Spagnuolo, M., Falcidieno, B., Rossignac, J.: Plumber: a method for a multi-scale decomposition of 3d shapes into tubular primitives and bodies. In: SM '04: Proceedings of the ninth ACM symposium on Solid modeling and applications, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2004) 339–344
16. Zhou, Y., Huang, Z.: Decomposing polygon meshes by means of critical points. In: MMM '04: Proceedings of the 10th International Multimedia Modelling Conference, Washington, DC, USA, IEEE Computer Society (2004) 187
17. Bespalov, D., Shokoufandeh, A., Regli, W.C., Sun, W.: Local feature extraction using scale-space decomposition. In: Computers und Information in Engineering Conference (DETC 2004-57702). ASME Design Engineering Technical Conferences, ASME Press (Sept. 2004)
18. Attene, M., Robbiano, F., Spagnuolo, M., Falcidieno, B.: Semantic annotation of 3d surface meshes based on feature characterization. Lecture Notes in Computer Science **Vol. 4816 (SAMT'07 Procs.)** (2007) pp. 126–139
19. Hoffman, D., Richards, W.: Parts of recognition. In Pinker, S., ed.: Visual Cognition. MIT Press, London (1985) 65Ũ96
20. Strothotte, T., Schlechtweg, S.: Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation. Morgan Kaufman Publisher, Los Altos (2002)
21. Diepstraten, J., Weiskopf, D., Ertl, T.: Transparency in interactive technical illustrations. Computer Graphics Forum **21**(3) (2002) 317–325
22. Bruckner, S., Groller, M.E.: Volumeshop: An interactive system for direct volume illustration. vis **00** (2005) 85
23. Diepstraten, J., Weiskopf, D., Ertl, T.: Interactive cutaway illustrations. Computer Graphics Forum **22**(3) (2003) 523–532
24. Götzelmann, T., Hartmann, K., Strothotte, T.: Contextual grouping of labels. In Schulze, T., Horton, G., Preim, B., Schlechtweg, S., eds.: 17th Conference on Simulation and Visualization, SCS Publishing House (2006) 245–258
25. Beckhaus, S., Ritter, F., Strothotte, T.: Guided exploration with dynamic potential fields: the cubicalpath system. Computer Graphics Forum **Vol. 20**(no. 4) (Dec. 2001) pp. 201–210
26. Ali, K., Hartmann, K., Strothotte, T.: Label Layout for Interactive 3D Illustrations. Journal of WSCG **13**(1–3) (2005) 1–8
27. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., DeRose, T.D.: Toolglass and Magic Lenses: The See-Through Interface. Computer Graphics **27**(Annual Conference Series) (1993) 73–80
28. Bier, E.A., Stone, M.C., Fishkin, K., Buxton, W., Baudel, T.: A taxonomy of see-through tools. In: Proceedings of CHI'94 conference, Addison-Wesley (April 1994) 358–364

29. Ellis, G., Bertini, E., Dix, A.: The sampling lens: Making sense of saturated visualizations. In: Proceedings of CHI'05, Portland, OR, USA, ACM Press (2005) 1351–1354
30. Trapp, M., Glander, T., Buchholz, H., Döllner, J.: 3d generalization lenses for interactive focus + context visualization of virtual city models. In: Proceedings 12th International Conference Information Visualisation (IV'08), London, UK, IEEE Computer Society (July 8–11 2008)
31. Viega, J., Conway, M.J., Williams, G., Pausch, R.: 3d magic lenses. In: UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology, New York, NY, USA, ACM (1996) 51–58
32. Looser, J., Billinghurst, M., Cockburn, A.: Through the looking glass: the use of lenses as an interface tool for augmented reality interfaces. In: GRAPHITE '04: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, New York, NY, USA, ACM (2004) 204–211
33. Götzelmann, T., Götze, M., Ali, K. Hartmann, K., Strothotte, T.: Annotating Images through Adaptation: An Integrated Text Authoring and Illustration Framework. Journal of WSCG **15**(1–3) (2007) 115–122
34. Alcocer, P.V.: On the Selection of Good Views and its Application to Computer Graphics. Phd thesis, University of Catalana, Barcelona, Spain (2004)
35. Fuchs, G.A., Luboschik, M., Hartmann, K., Ali, K., Strothotte, T., Schumann, H.: Adaptive labeling for interactive mobile information systems. IV **0** (2006) 453–459