3D Diff: An Interactive Approach to Mesh Differencing and Conflict Resolution

Jozef Doboš,* Anthony Steed
University College London

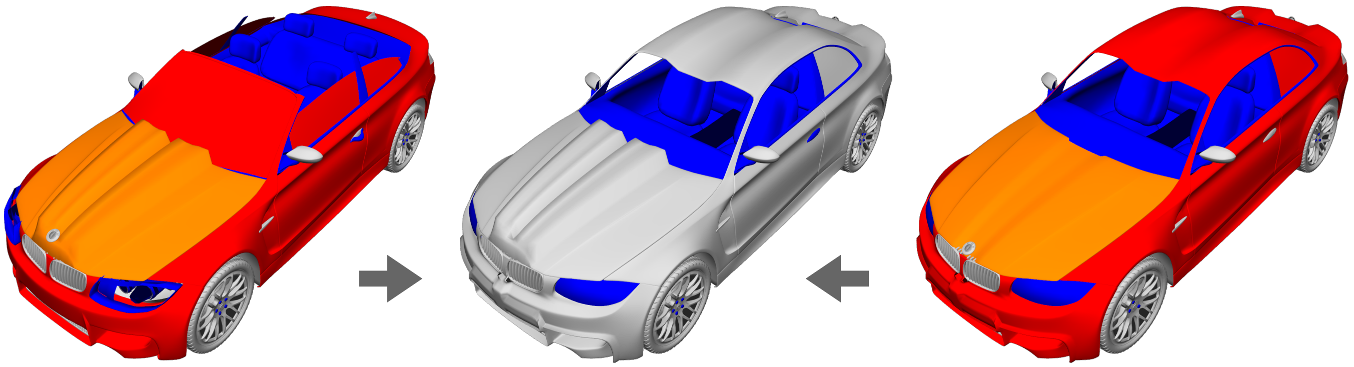

Figure 1: *3-way 3D diff: Two revisions (left and right) of the same model are compared against their common ancestor and each other to suggest automatic conflict resolution (in blue). Conflicts are in red, non-conflicting modifications in blue, user selection in orange and manually merged results in gray. Our user interface facilitates the 2- and 3-way 3D diff. Model attribution: Blender Foundation.*

1 Introduction

We introduce *3D diff*, a novel approach to interactively resolve conflicting mesh edits. Our method performs conflict detection and visualization and allows the user to quickly select one of the two revisions for each conflicted scene graph (SG) node. By further integrating the knowledge about a common ancestor of the two models, a.k.a. 3-way diff [Mens 2002], we are able to automatically resolve more conflicts than in a standard 2-way comparison. Our method is particularly useful for revision management of 3D models.

In general, comparing and merging two models is a complex and time consuming task especially when concurrent edits by multiple people are to be integrated. When combining modified versions of the same 3D scene, popular modelling packages such as Autodesk 3ds Max, Maya or Blendersuperimpose the meshes but do not aid the merging process any further. Despite the advantages of vertex-level editing, most of the time it would be satisfactory to simply preserve whole sections from the individual model versions to form a combined result.

In their recent revision control system for images, Chen et al. [2011] implemented a visual diff as a playback of recorded image edits. State of the art side-by-side 3D model comparison is offered by Provenance Explorer for Maya[1] which, similarly, relies on stored edit sequences. In contrast, our prototype works independently from editing software like many line-based differencing tools for source code management [Mens 2002]. Closest to our approach is the abandoned Art Diff for Subversion project[2] that loads 3D files for a basic visual comparison but does not detect conflicts nor does it support merging.

2 Our Approach

During the modelling stage, we assign each SG node a universally unique identifier (UUID) so that the same parts of a model can easily be matched when comparing models' different revisions. To ensure that our solution is independent from the modelling software, we decompose most common 3D file formats into their SG components via the Open Asset Import Library (Assimp[3]).

Our stand-alone model viewer performs an early reject byte-by-byte comparison of SG nodes from various revisions that share the same UUID. When there are discrepancies in the two models, they are treated as conflicting edits (red in Fig. 1). However, in a 3-way comparison, we add extra information about the common ancestor of the differenced models. If one of the versions is the same as the original, i.e. no changes have been made, the other must be the intended *modification* to be automatically preserved during the merge process (blue). Nevertheless, the user has to confirm all changes including modifications as these could visually interfere with other parts of the model. To perform a fast merge, the user can select a version for each conflicted SG node in a color-coded conflicts list and inspect the proposed combined result visually.

3 Evaluation and Conclusions

Based on the initial evaluation of our prototype, we believe that such approach to mesh differencing and merging can significantly speed up the revision control management of 3D models. What is more, this technique allows for comparison of otherwise incompatible 3D file formats. Integration into modelling packages via their plug-in frameworks is also possible. In the near future, we plan to make this work open source and will investigate automated camera navigation for better context understanding, bounding box intersection detection and vertex-level merging.

References

Chen, H.-T., Wei, L.-Y., and Chang, C.-F. 2011. Nonlinear revision control for images. *ACM Trans. Graph. 30*, 4 (Aug.), 105:1–105:10.

Mens, T. 2002. A state-of-the-art survey on software merging. *IEEE Transactions on Software Engineering 28*, 5, 449–462.

*e-mail: j.dobos@cs.ucl.ac.uk

[1] http://vistrails.com/maya.html

[2] http://code.google.com/p/artdiff/

[3] http://assimp.sourceforge.net

SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008