Abstract

Mesh quality is one of the key factors in the success and accuracy of computational fluid dynamics simulations. While current simulations are paving the way towards extreme-scale problems, mesh quality exploration becomes challenging. Due to the fact that such simulations are time consuming, an immediate on-the-fly mesh analysis tool is strongly desired in order to adjust running simulations and prevent failure.

We present an interactive mesh exploration tool in a virtual environment which employs volume rendering techniques with user defined regions of interest as well as error thresholding functionality. Our approach provides an in-situ visualization of the mesh quality for an ongoing simulation process, allowing domain experts to locate and modify ill-defined mesh regions at the current simulation time step. Furthermore, virtual environments enable detailed inspection and exploration of large meshes, reducing visual clutter and allowing for rapid and interactive identification of mesh errors.


1 INTRODUCTION

Current Computational Fluid Dynamics (CFD) simulations such as those used in airplane design are dealing with complex scenes that consist of large-scale meshes. With the increasing computational power provided by new hardware, these simulations are also growing towards extreme and exa-scales. Due to their complexity and overwhelming sizes, convergence can take multiple weeks even on large-scale cluster systems. Moreover, simulation processes often crash or fail to converge due to poorly defined meshes, necessitating a restart of the whole simulation with modified parameters. To speed up the simulation process, an on-the-fly mesh exploration tool is desired. It should allow the user to explore the mesh quality at the current simulation time step in an interactive manner, and it should provide the simulation experts with the possibility to monitor and modify the meshes where necessary.

Mesh quality visualization in a virtual environment enables a detailed and immersive exploration of the mesh. The free view point and immersive display system enhances the sense of orientation and perception of depth, thus allowing for a rapid location of critical mesh regions.

To enable the usage of virtual environments, it is important to maintain interactive frame rates. This requirement is already a tremendous challenge when large-scale simulations with huge amount of data should be visualized. Another important fact while visualizing extreme scale data sets is that the size of a single mesh cell might be even smaller than a screen pixel. Therefore, resampling of the simulation grid needs to be done without losing important information.

To address the above issues, we present an approach for interactive mesh exploration in a virtual environment. The user interactively defines a region of interest. Then inside this region, cell-based mesh errors are re-sampled to a volume containing the maximum error of all intersecting cells. After transferring this volume back to the virtual environment, the mesh error distribution can be explored by interactively changing the used transfer function for volume rendering.

2 FRAMEWORK AND ARCHITECTURE

Figure 1 shows the architecture of our system. A large-scale CFD simulation is running in parallel on a cluster of simulation nodes. To study the mesh quality, a VR-visualization front-end is connected to this running simulation. The VR front-end provides the user with a possibility to interactively define a region of interest to constrain analysis to relevant areas. This region of the mesh is then extracted and re-sampled in parallel on the involved simulation nodes. Finally, the re-sampled regions are collected by a designated master node and transferred back to the visualization front-end, where mesh errors are visualized using volume rendering techniques. If the front-end runs in a tiled display mode, the extracted volume is further distributed by the front-end’s master node to clients.

The visualization enables the user to identify problematic mesh regions using bounding boxes. The extents of these boxes are then transmitted to the simulation nodes via the master node. Each node independently determines whether its mesh region intersects the bounding box. A mesh refinement algorithm is then applied to the intersecting cells to improve mesh quality. The simulation then continues with the adapted mesh.

Figure 1: System architecture

The visualization enables the user to identify problematic mesh regions using bounding boxes. The extents of these boxes are then transmitted to the simulation nodes via the master node. Each node independently determines whether its mesh region intersects the bounding box. A mesh refinement algorithm is then applied to the intersecting cells to improve mesh quality. The simulation then continues with the adapted mesh.
edge measurement of mesh quality will not be visible any more. In our approach, we use a simple per-poses results in very occluded visualizations and a single mesh cell will not be visible any more. In our approach, we use a simple per-edge measurement of mesh quality:

$$\text{Error}_\text{edge} = (V(P_1) - V(P_0)) ||P_1 - P_0||^2$$

with $V(P)$ the pressure coefficient of the field at point $P$. For every cell, the error is taken as the maximum of the edge errors, which is $\text{Error}_\text{cell} = \max \{\text{Error}_\text{edge}\}$.

Visualizations of large meshes are often slow to compute and it is hard to extract useful information out of the produced image due to occlusion. To avoid loading the complete mesh data and to minimize the amount of time required for visualization, we employ the following visualization paradigms in order to provide a meaningful and intuitive representation of the mesh quality.

To limit the amount of mesh data to be visualized, we incorporate a region of interest method [3] into our approach. This allows the user to interactively navigate smoothly through the data set. The bounding box of the user defined region, which is adjustable in size and position, is then sent to the simulation back-end. We extract the mesh cells within this region in a distributed manner on the simulation back-end, using the same partitioning as the simulation. To speed up the cell search, which can be very time consuming, a cell-tree approach is used [2]. Cells error values are then re-sampled to a regular volume grid. Finally, the down-sampled volume data is collected and sent back to the visualization front-end for rendering.

We utilize a volume rendering technique based on texture mapping for visualizing the extracted mesh error. Different values of errors are mapped to different colors in the volume. With this technique, users can identify regions containing critical cells in an efficient way. Initially, a full range of error values are mapped to the volume. However, the user is typically only interested in a certain range of error values, normally above a certain threshold. Brushing, as a common information visualization technique [4], allows the user to interactively select only a subset of the complete data. In this case, we implemented brushing functionalities allowing the user to define and limit the error range. As a result, only a minimal amount of volume is mapped to a visible color. Irrelevant information is filtered out and regions of poorly defined mesh can be identified fast and easily.

4 Preliminary results

We illustrate our approach with a test data set which simulates the airflow around an airplane wing. The simulation mesh is partitioned into four simulation blocks. One region of interest consist of ten percent of the original mesh. This region is then mapped into a 256x256x256 voxel volume. Even in this small test case, the diameter of mesh elements is smaller than the used voxel size. Therefore, each voxel contains the maximum error of all included cells. The volume rendering on the front-end is performed at interactive frame rates.

Figure 2(c) shows a user exploring the mesh quality of a airplane data set in front of a powerwall. With the implemented tool, the user is able to smoothly navigate through the whole mesh and quickly identify and locate the red region of cells which need to be modified. The usage of error brushing is demonstrated in Figure 2(a) and (b). In the former image, mesh errors with a range of [0.05, 0.32] are mapped to different colors. In fact, only the regions colored in red are of user’s interest and have a high value in error function. By limiting the error value to a smaller range (Figure 2(b)), insignificant information is removed, leaving only the mesh region with bad quality unveiled.

5 Future work

Our future work includes extending our approach to larger-scale data. Interaction with extreme scale data can be improved in order to navigate more efficiently through huge simulation scenarios. We also plan to investigate further visualization techniques for mesh quality visualization and to search for better metaphors for the selection of critical regions.

References