11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20 - 25, 2014, Barcelona, Spain

Detail-preserving mesh simplification for scientific visualization

Miguel A. Pasenau[1] and Carlos Andújar[2]

[1] Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE)., Gran Capitán s/n, Edificio C1 - Campus Nord UPC, 08034 Barcelona, Spain, miguel@cimne.upc.edu https://web.cimne.upc.edu/users/miguel/

[2] Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, Edificio C1, 08034 Barcelona, Spain, andujar@lsi.upc.edu and http://www.lsi.upc.edu/~virtual/home/index.html

Key Words: *High performance computing, detail-preserving surface mesh simplification, vertex clustering, quadric error metric.*

Dealing with large simulation data is a growing challenge. Current computational capabilities allow engineers to tackle large and complex simulations. The visualization of these large-scale problems is becoming an issue as current solutions rely on powerful hardware resources very close to the data being generated. Scientists located at remote sites need to download the simulation data to its own computer for local visualization, resulting in large download times, or visualize the simulations results remotely, resulting is poor interaction. One way to accelerate the visualization is to provide a simplified view while the user interacts with the model. Given a polygonal mesh, the goal is to generate another mesh which approximates the underlying shape but includes fewer primitives. Early methods focused only on preserving the overall shape of the geometric model, whereas current methods also handle meshes with per-vertex attributes, like colours and texture coordinates, so that both the mesh shape and the mesh appearance are preserved. This work describes several detail-preserving techniques based on vertex clustering and quadric error functions to simplify efficiently large simulation meshes. These techniques have also been incorporated into GiD, a pre- and post-processor application.

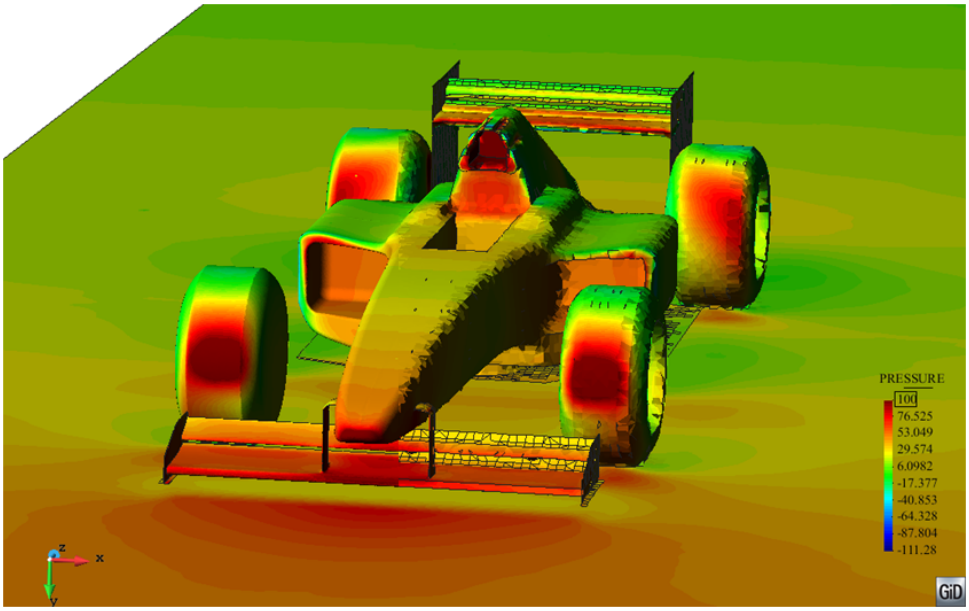

Figure 1: left half image shows the original car surface with 5 million triangles, which achieved a frame rate of 0.3 fps on a Intel 1,3 GHz dual Pentium laptop. The right half image shows the same model simplified to 233 thousand triangles, achieving a frame rate of 7 fps.