Visualization of UV-maps for Polygonal Meshes as a 3D Morphing

Candidato: Giorgio Sepe, Relatore: Marco Tarini

9 October 2024

Summary

Context. In Computed Graphics, 3D uv-mapped meshes are the most common format to describe detailed 3D objects, and are ubiquitously used in contexts such as video-games, virtual reality, scientific visualization, etc. They consist of a polygonized approximation of the surface enriched with a *uv*-map, which describes a mapping of the surface into a planar domain. The construction of a good *uv*-map for a given mesh is an important and difficult task in the creation of these datasets, performed either by trained digital artists or by automatic Geometry Processing algorithms.

Objectives. The objective of this thesis is the design and development of a tool aimed at providing an intuitive data visualization of this mapping, given one instance of a uv-mapped mesh (regardless of its origin). This tool consists in automatically creating and displaying (in real time) the mapping in the form of a physically plausible morphing between the original 3D shape and its 2D counterpart. The intended uses range from a visual qualitative assessment of the quality of a given *uv*-map (in addition to the many numerical assessments commonly employed in literature), to illustration purpose, including in didactic contexts.

Methodology The sought morphing is obtained by identifying a plausible path for every elements of the mesh from its original 3D position in object space to the corresponding 2D position in uv space. Preliminarly, the texture plane is strategically positioned and oriented in 3D to optimize the path of every element. The paths are computer in a variety of alternative methods, ranging from simpler linear interpolation to quaternion based quasi-rigid transformation representations and their interpolation. Postprocessing is used to maximize the local rigidity of the intermediate shapes. Additionally the temporization of each element is differentiated to reduce cluttering.

Experiments and Employed Technologies The investigated approaches have been implemented as a easy to use demonstrative Web application that is capable of importing user data in standard format and to display the morphing to reveal the nature of *uv*-map. The geometry processing part is implemented in C++ using the GLM library (for 3D mathematics for graphics) and the Eigen library (for linear algebra). Emscripten (an LLVM/Clang-based compiler) was used to integrate the C++ source code as WebAssembly into the Web application. The front end is implemented directly in JavaScript using the ThreeJs library for rendering and graphics, the web application is complete with a GUI implemented using standard tools like HTML and CSS.

Results and Conclusions Even though it is difficult to quantitatively measure the effectiveness of the proposed solutions, the resulting interactive animations have been preliminary found to be effective in conveying the quality and characteristics of the inspected *uv*-maps.