Final Project Proposal - INF01009
An Extension of the Kelvinlet Deformation Inversion to Volumes with Distributed CUDA Processing Visualization

Guilherme G. Haetinger - 00274702

September 1, 2021

Previous Work

My previous work on the subject entailed applying the Pixar elaborated brush on images and videos, warping them in the *XY-axis* and the *XYaxis + Time-axis*, respectively. The main goal of the paper was to implement an inversion of the non-linear *Kelvinlet* equation with the *Gauss Newton Inversion* Method. This was needed when deforming videos, which are hard to be reconstructed by mesh triangulation and very inefficient to interpolate (more details in the paper's *Naive method* section). We obtained really good results at really good efficiency, being able to deform a FullHD video at **300FPS** [2].

Project Description

Volume visualization is a technique that is broadly used. It has applications in medical diagnosis and simulation, as well as geological exploration. Applying local deformations to volumes in these sorts of applications can be of use when simulating possible impacts, distortions and variations of objects. *E.g.* Kids can be born with an assymetrical skull and require surgery to put a side back to place. This could be simulated by Volume deformation, which would not only fix the position of the skull's surface in order to show the post-operatorium to the parents, but would also give an idea to how the organs might rearrange during the procedure.

Visualization techniques used nowadays can make it hard to have satisfactory results with fast deformation and interaction. Some work [1] that has been done in this area "resolves" this problem by implementing tetrahedral interpolation, *i.e.* they seem to solve it by generating a new volume and interpolating the voxels by location to retrieve a renderizeable 3D mesh. This, however, can be considered somewhat slow, performing at **60FPS** on a relatively small volume ($256 \times 160 \times 122$).

As a small comparison, the first application of our inversion technique on a slightly larger volume ($512 \times 512 \times 100$) was able to achieve $\approx$ **40FPS**, but with a more complex deformation process and worse hardware. We think that, with the following proposal, we might be able to surpass these numbers and/or be able to present a much more useable *Real-Time* application.

Implementation Details

Videos are nothing but volumes with very well defined steps in the *Z-axis* (frames). Our implementation of the video rendering process was based on a three-dimensional texture. Thus, so is our volume rendering. This approach makes it easy to read, interpolate and interact with the values.

As we moved on to volume rendering, the efficiency of our deformation worsened as we had to implement a ray marcher that, for each step, calculating a *Gauss-Newton Inversion* of a point in the 3D space. Therefore, we propose to implement this inversion in **CUDA**, which would instead of calculate the respective deformed points on each rendered step, would rewrite the texture! We can have a number of *Bells & Whistles* on top of this, making the deformation even more interactive. We intend to use the *CUDA-OpenGL Interoperability* library to have **CUDA** read and write from texture buffers.

References

[1] J. Gascon, J. M. Espadero, A. G. Perez, R. Torres, and M. A. Otaduy. Fast deformation of volume data using tetrahedral mesh rasterization. In *SIGGRAPH/EG Symp. on Comput. Animation*, pages 181–185, 2013.

[2] G. G. Haetinger and E. S. Gastal. Regularized kelvinlet inversion for real-time image deformation and video time warping. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 124–131. IEEE, 2020.