Hybrid visualization for maxillofacial surgery planning and simulation

1. Nowadays, computer-assisted surgical simulation is widely used in medicine for prediction and simulation of surgical treatments, and in our case of study, in maxillofacial surgeries. In this field, surgeons demand a way to be able to predict the new facial outlook of their patients. Also, it improves communication among the medical team in the planning stage.

2. Maxillofacial surgery treats diseases, injuries and abnormalities in the head by bone repositioning, restoration and implants. As an illustration, we show some kinds of osteotomies used in surgery. This project is focused in surgeries that involve maxilla and mandible...

3. ...and its main goal is the prediction of the final appearance of the patient. We introduce an new planning stage, allowing its visualization in 3D while using an intuitive 2D interface for osteotomies definition, as well as we describe the visualization techniques used during planning and simulation. For more information about the simulation module, we refer to a previous work.

4. Our tool provides the surgeon with an inspection tool of the data, including the 3D visualization and reconstruction of the model, navigation, and definition of regions of interest for more detailed inspection.

5. In general, there are four main steps in the planning and simulation workflow: read the patient input data (which usually comes from CT scanners), segmentation and 3D visualization, surgery planning, and simulation and results visualization. I'm going to describe the visualizations steps and the planning module.

6. The visualization pipeline starts with the input reading. To overcome memory constrains and reach interactive framerates, we decided to resize the dataset by two. In this way, there is room to precompute information like the gradient. It is important to note that the quality of the final image is still enough for our purposes, although we aim to use the full data when the framerate does not become affected. After data is read, and under user interaction, materials and cutting planes are defined, while a hybrid model is rendered.

7. Volumetric visualization is a common and powerful tool in medical data exploration. There is a huge state of the art about this topic, but we will focus our attention in the full volume rendering. This technique casts light rays through the volume and composites the final image.

Currently, it is very common to implement the ray caster in a fragment shader since each ray corresponds with a pixel of the rendered image.

8. There are several ways to compute the light rays, one useful and general approach is the use of proxy geometries to obtain the entry-exit points of the rays within the volume. The entry and exit points are computed using the per-vertex color of the proxy geometry.

9. Although the shape of the proxy geometry is the bounding box of the volume, we have made an extension, using a vertex shader, so any convex object can be used (if it is concave, more than one entry-exit points pair may be obtained and the renderer becomes more complex, usually requiring several passes to get a single image). In this way, regions of interest (or work volumes) can be defined straightforwardly.

10. This is the general equation used in the vertex shader, where p is the vertex, o the center of the work volume, B its bounding box and D the dimension of the original volume.

11. With the same concept of regions of interest, clipping planes can be applied to the model. The only difference is how the user defines the clipping plane, in this case using a tool to rotate and translate the plane. Using the same approach, immersion in the volume can be performed using proxy geometries, although it is necessary to make some modification, since the proxy geometry is clipped by the near-Z plane (then no entry points are obtained). We solved this issue by computing the intersection of the proxy geometry with the near-Z plane. Since the proxy is already being clipped, only the intersection is needed and, as the original volume of work is convex, the intersection is convex too, so only the vertices have to be computed and then sorted around the camera direction.

12. In order to visualize the distinct tissues, optical properties such as color and opacity are associated to different ranges of density. These materials are mapped from the volumetric data using transfer functions. The main drawback is that they do not provide enough information and produce highly artificial images; to improve them, light computation must be included in the ray caster, reducing considerably the framerate. Transfer styles adds more information preserving the simplicity of basic transfer functions. Transfer styles use sphere maps to create an image-based illumination for each material, using not only the density value of the voxel but also the projected normal on camera space to map the optical properties. Our tool provides the surgeons with several predefined materials, such as classical bone and soft tissue, highly specular bone for better contrast and illustrative bone. We assumed an spherical head (you know, be an spherical cow...), so normals follow, in general terms, an spherical distribution...

13. Based on it, we have also designed a material to produce images with an area of focus, reducing the opacity of the center of the sphere map, so voxels whose normals are oriented to the camera become translucent.

14. There also exist multidimensional transfer functions, that make use of not only the density value of each voxel to differentiate tissues, but also the gradient magnitude. It increases the visual differentiation between materials, but there is a huge complexity in creating the transfer function. Instead, we have used a gradient magnitude opacity modulation to remark density changes...

15. ...it is, boundary between tissues.

16. The opacity function associated with these materials is usually a step function, easy to define and very intuitive, but may produce aliasing artifacts. To reduce them, we have developed an opacity level smoother, by the use of Bézier curves, which is applied at the time of the transfer function definition.

17. The planning stage consists in the definition of different osteotomies specified by the surgeon. Working with 3D objects is still difficult, so we decided to create a 2D planning definition, at the time that the 3D planning is shown in a secondary visualization, making this process very intuitive and usable.

18. To reduce the complexity of this representation, our planning module uses a few planes to define osteotomies...

19. ...and to specify the different pieces of the skull. Although it is a big simplification from the actual osteotomies, most of the detail removed does not have a direct effect on the outlook appearance of the patient, so can be discarded in favor of an easier definition. Along with the planes definition, pieces movements are specified too.

20. OK, volumetric visualization is great for medical visualization, but… it lacks of a way to easily incorporate additional elements such as instruments or, in our case, the planning definition. The main issue lies in the ray-marching algorithm, which casts the whole volume regardless occlusions from other objects. Several solutions have been presented, many of them based on multipass renderers that switch between two casters, one for the volumetric data and one for the polygonal objects. Among them, the presented by Nes makes use of

sorted proxy geometries from the polygonal data, to subdivide the rays. We noted that if no translucent objects are used, this method can be simplified even more to skip the sorting phase. Instead of rendering several times all the scene and peel out each layer, objects and the back faces of the proxy geometry are rendered together using the vertex shader described previously. In this way, the end points of each ray are obtained.

21. Then, the volumetric visualization is performed and the final image is composited with a normal rendering of the polygonal data.

22. Once the surgery has been simulated, many of the initial conditions of our data are lost, mainly its volumetric structure. The simulation module uses a mass-spring model which starts with an uniform grid equivalent to the volumetric data; but this equivalence is lost in many parts of the model after the simulation and the volumetric visualization is no longer possible. We decided to use a mesh representation of the patient skin, obtained in a segmentation process. For performance reasons, the data used for simulation is a reduced set of the original, and it doesn’t have the appropriate resolution for visualization. Instead, we extract two skin models from the data set (one high quality for visualization and one with less density for the simulation). Using the low resolution model from the simulation, we morph the high quality model from its pre-surgery state. To create the mesh, we used the GPU Local Triangulation algorithm, that has shown reconstruction times around the 10 seconds in most of the tests. In general, GLT performs local Delaunay triangulations in a parallel fashion.

23. To avoid reconstruction errors that a Delaunay triangulation may cause on a point set with a grid structure, an small perturbation is initially applied to all the points. When the reconstruction is completed, this perturbation is discarded.

24. We have presented a hybrid visualization pipeline for maxillofacial surgery planning and simulation, combining different volumetric visualization techniques to render the patient data, along with polygonal representations for external objects, specially the surgery planning. We also remark the planning module; based on the fact that many surgery details do not have influence on the external appearance of the patient, the complexity of the planning definition can be reduced. This fact has been remarked by the surgeons who have tested the application, because the time they spend in the planning stage has been reduced.

Thanks