Creating a toolkit for volume rendering with a variety of visual styles

Paul Bourke
WASP
University of Western Australia

Motivation

- Visualisation by peers: literal, interactive, stereoscopic.
- Publications: limited colour, perhaps greyscale, single still image.
- Posters: high quality.
- Public spaces: audience doesn't necessarily have the visual language, needs to be engaging.
- Artisic expressions: less need to “honour” the data, evocative representations.
- Raw data export: for incorporation into other environments, pass on to animators.

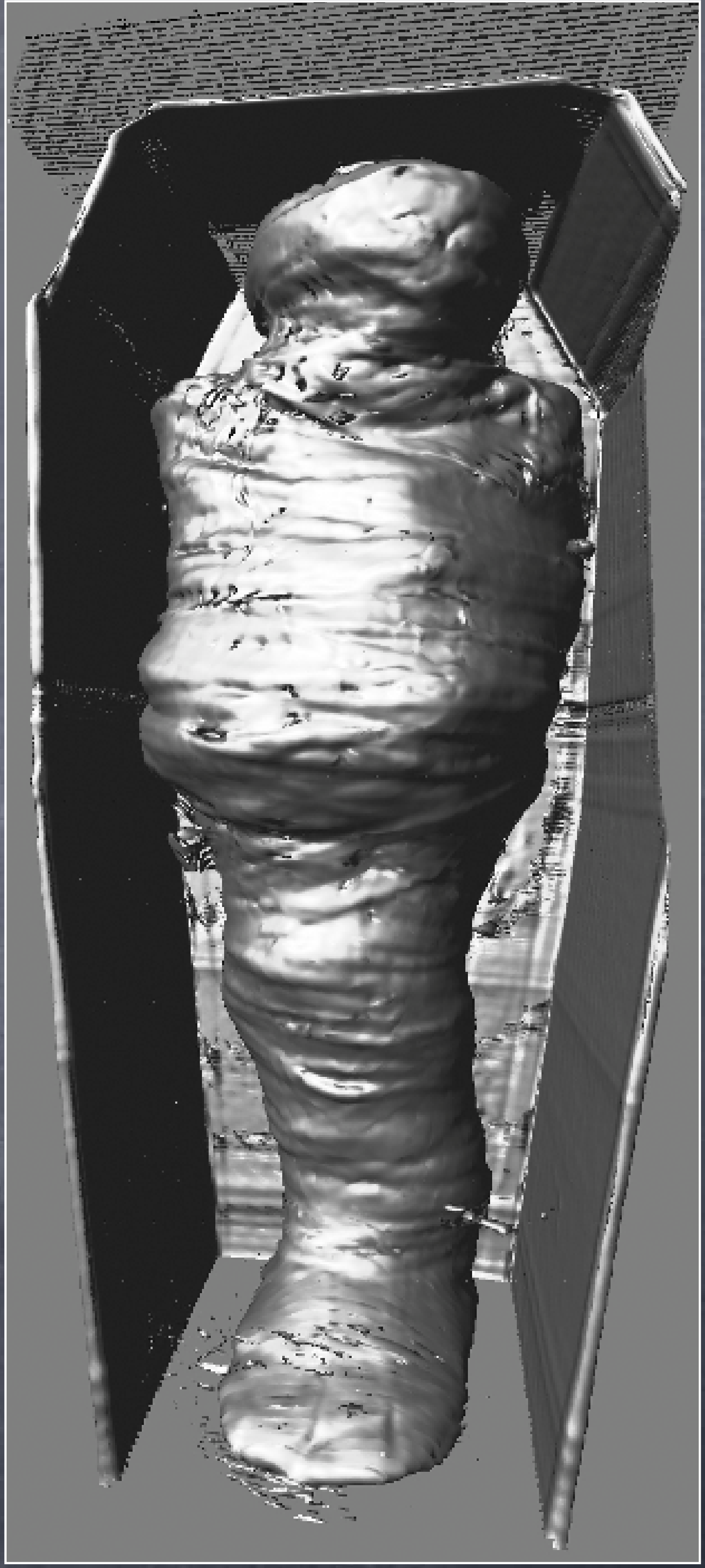
Requirements

- Create a collection of consistent tools that operate on the same dataset to create a wide range of rendering styles. Remove the need for multiple data representations, different conventions, etc.
- Available as source code ... needs to be integrated with other software by the author.
- Allows perspective projection rather than just orthographic ... required for stereoscopic projection.
- Relatively self contained. Many of the traditional solutions (VolPack, VolVis, VTK, MPIRE, SPIRE, VOLUME, ...) carry a lot of extra baggage some of which is getting hard to install and for which cross platform implementation can be problematic.

Curvature shading

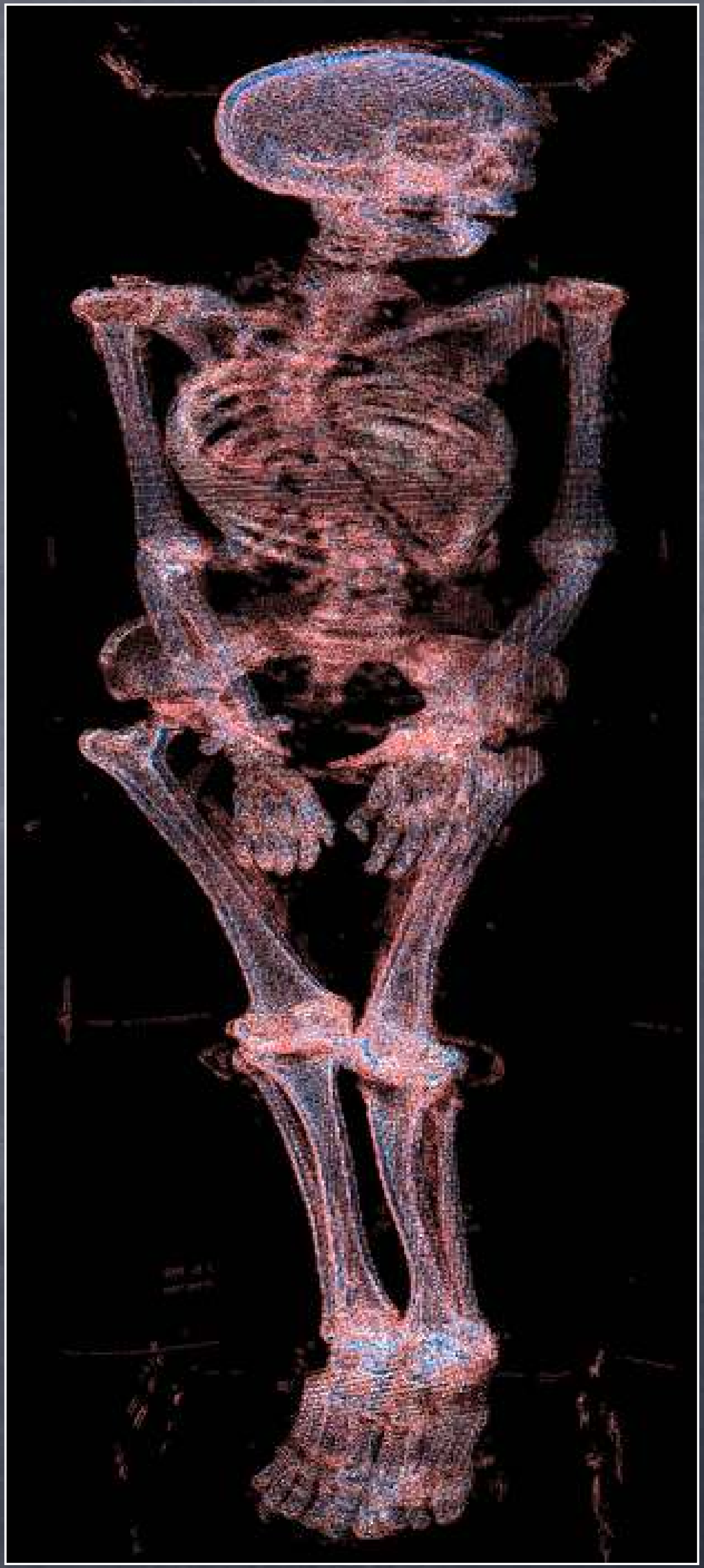
Exploratory dataset

- Mummy, unopened but with some drill holes. (Courtesy MONA)
- CAT scanned at 512x512x512 (Hobart Hospital)
- 16 bit voxels => 266MB.
- DICOM files, header plus one file per 512 slices.
- Raw CAT scan data, no data smoothing or filtering applied.
- Note: incorrect slice separation used in renders shown here, slice thickness was not available at the time.



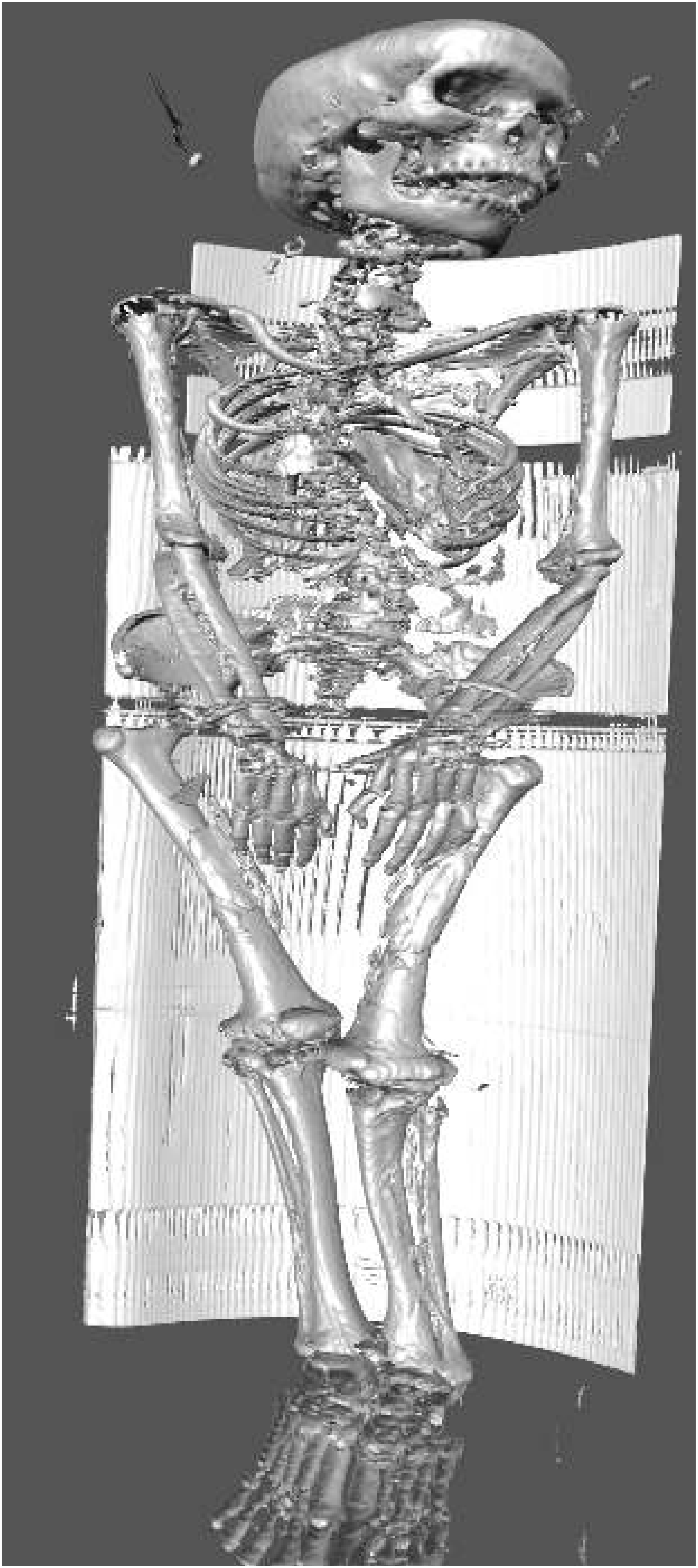
Point clouds

- Points drawn between voxels that span an isolevel.
- Lame?
- Excellent for initial data exploration.
- Multiple isolevels at once, quickly and interactively scan the volumetric space.
- Working on point cloud occlusion.



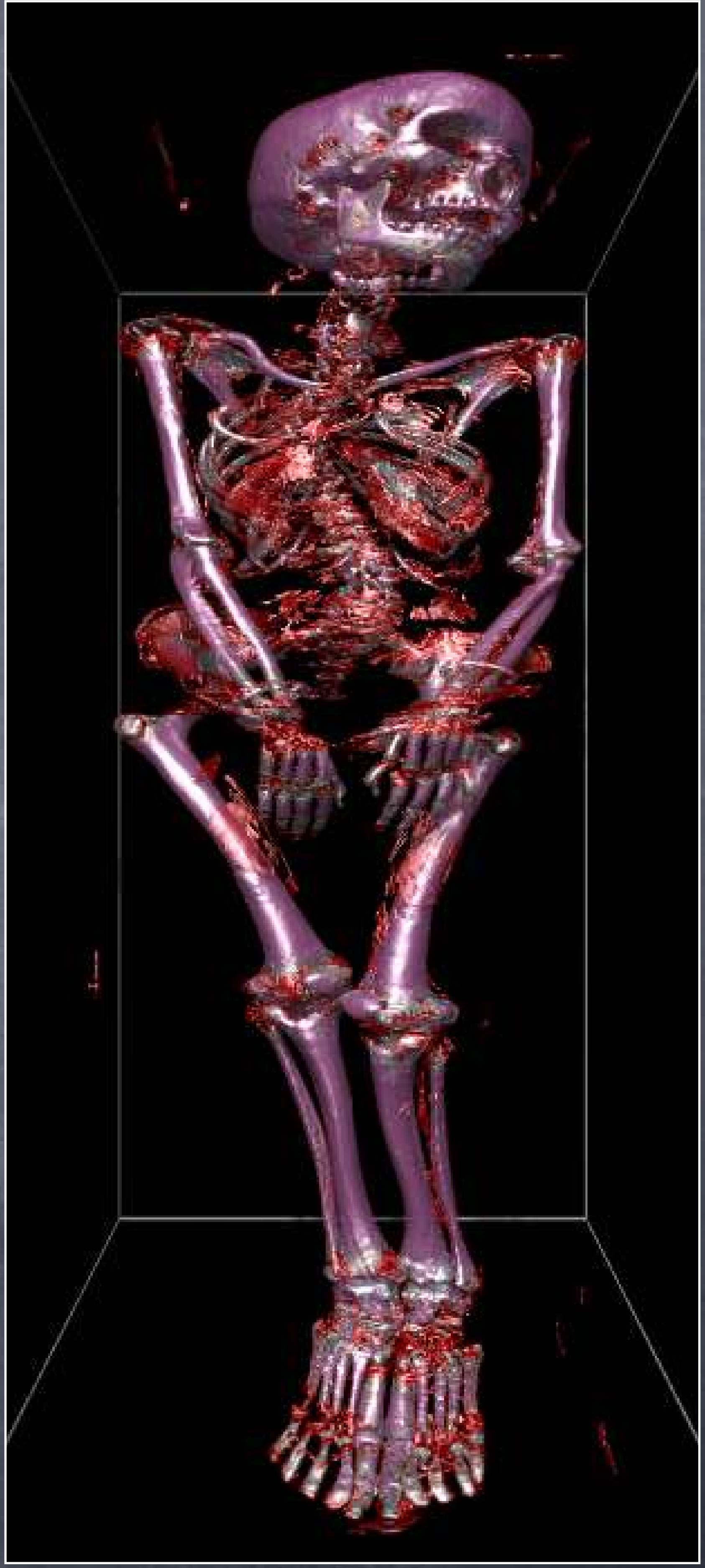
Isosurfaces: Standard fare

- Marching cubes, tetrahedrons and similar.
- Not a particularly efficient mesh representation. Various mesh simplification algorithms exist, surface relaxation, etc. Using my own algorithms and ideas implemented in polyr (Arup Nielsen)
- Choose vertex normal from volume gradient or mesh connectivity, a variety of algorithms available.
- Usually easy to export to commercial animation packages, 3DStudioMax, Lightwave, Maya, etc.
- Problem: choosing a single isolevel.
- Throwing so much information away!



Multiple isosurfaces

- Partially solves the choice of isosurface problem.
- Each isosurface can be semi-transparent and different colours giving many more colour combinations than the number of isosurfaces.
- Potentially large models but mesh reduction and relaxation improves that. Generally interactive, mesh representation has shared vertices and normals.
- Can also be rendered for shadows and other effects, as well as exported to traditional modelling/animation/rendering packages.
- Still investigating derivation of triangle stripes for further OpenGl drawing efficiency, doesn't seem trivial.



Ray Casting

- Simplistic algorithm:

```
calculate entry point of camera ray with volume
while ray is in volume
    estimate value at current ray position
    calculate/accumulate colour and opacity
    advance ray
```

- Diffuse/specular surface shading and shadows possible.

- Rich set of transfer functions. Including 2D functions of voxel value and gradient.

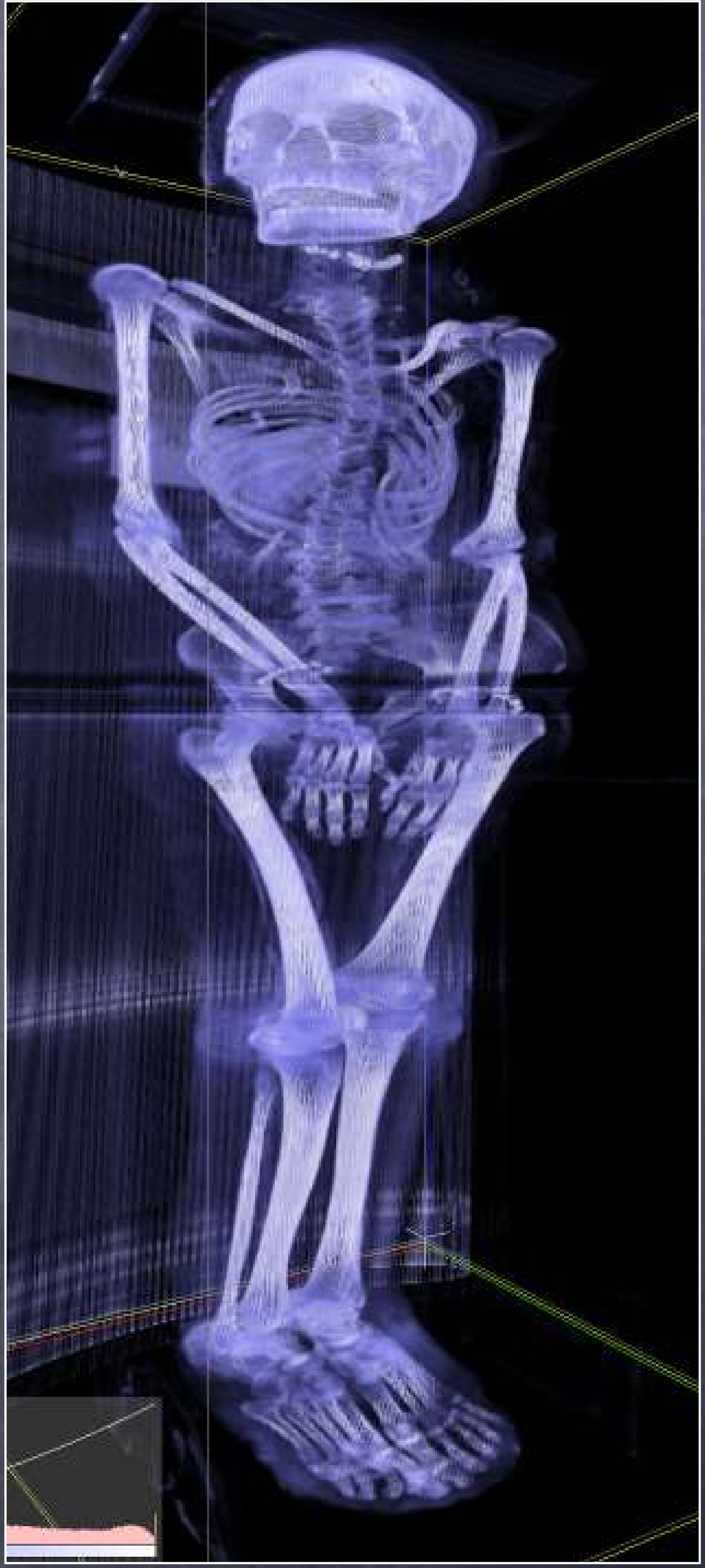
- Lots of optimisation potential
  - early ray termination
  - empty space skipping
  - hierarchical octree volumes
  - . . . .

Shader implementation

- Fragment shader samples a 3D texture volume along rays from the camera.
- spvolren: "A Simple and Flexible Volume Rendering Framework for Graphics-Hardware-based Raycasting" by Simon Stegmaier, Magnus Strengert, Thomas Klein.
- Interactive performance, but depends on many factors, eg: ray advance step size.
- Standards and how future proof?
- Current development: adding stereoscopic support and additional shaders to create further rendering styles.

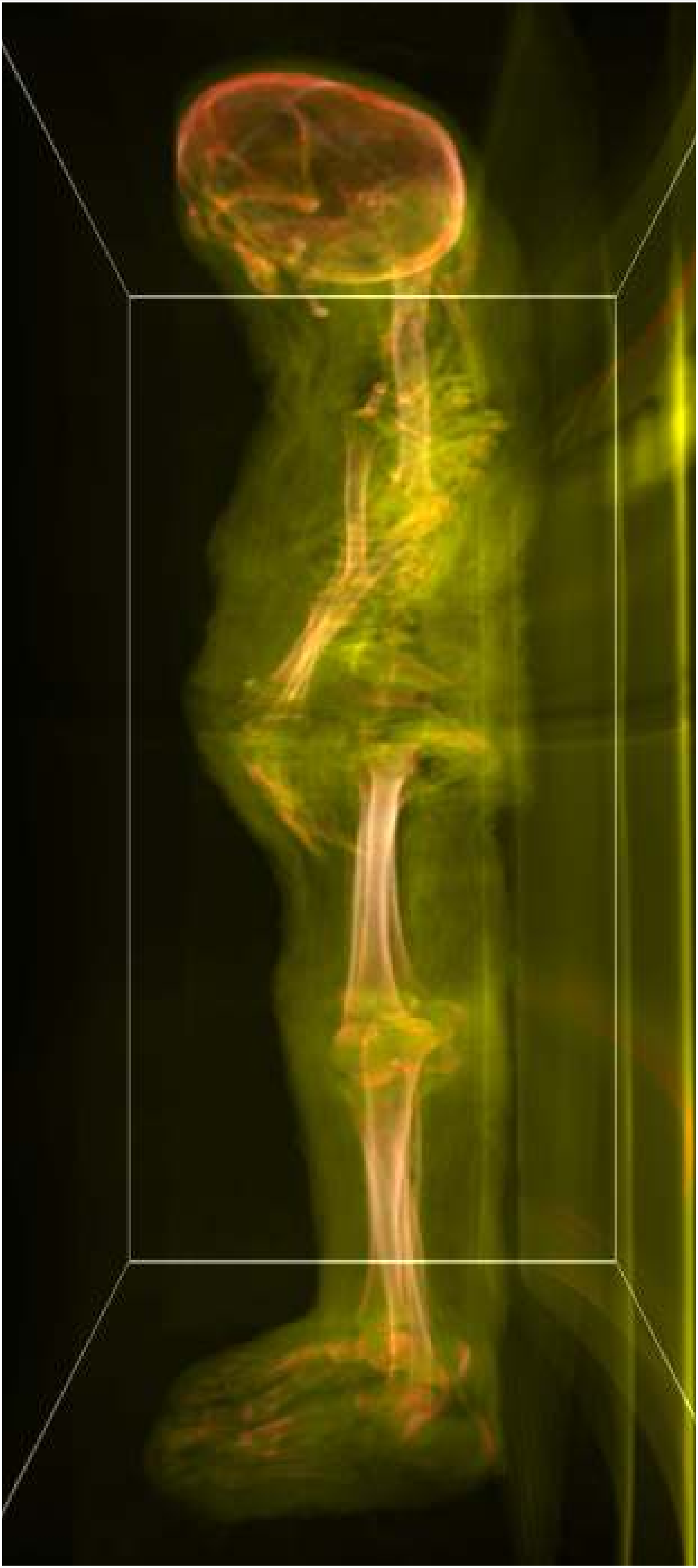
Brute force texture mapping

- Developed by myself in 2000.
- Back to front blending of textured planes.
- Interactive (given enough texture memory).
- Improve memory requirements by not precomputing texture for all three view orientations. Consequence is a delay when switching between view orientations.
- Uses 16 bit opacity and colour mapping functions.
- Various tricks possible to improve performance: sub-sectioning and sub-sampling during navigation.



Emissive media

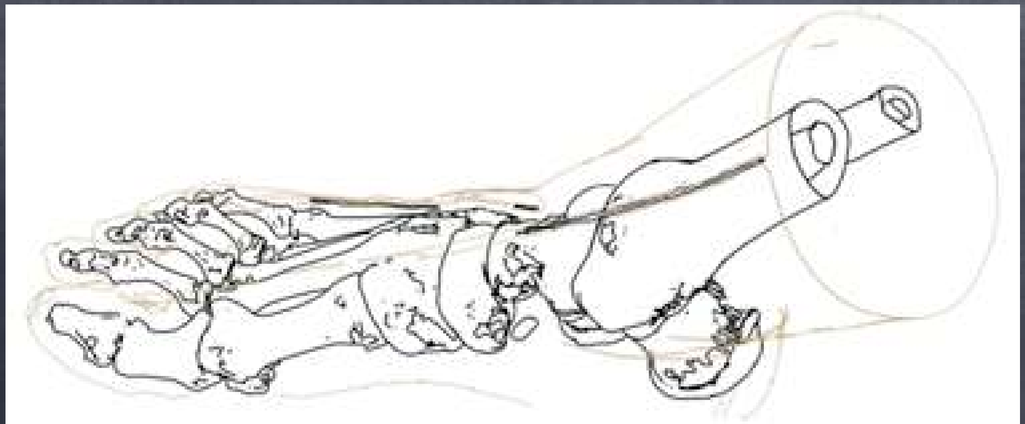
- The density of the voxels determines the brightness and colour of the light emitted.
- Based upon a simple fluorescence model, no shadowing.
- Continuous colour/brightness mappings.
- No occlusion/absorption so greatly improved result when viewed stereoscopically.
- Currently experimenting with full more physically realistic media properties: emissivity, absorption, scattering.



Angry Mummy

Summary

- Somewhat surprised by how few freely available tools are available that are not “old”. Lots of papers but very few groups releasing source code.
- Impressed with the interactive performance possible on high end, but still consumer, graphics cards.
- Happy with the visual variety achieved so far.
- Keen to develop further non-photorealistic and diagrammatic/sketch styles. For example: Michael Burns et al, “Line Drawings from Volume Data”



Questions / comments

- Credit for CAT scan data of mummy: Museum of Old and New Art, Hobart, Tasmania.

- polyr: “Polygon generation program” by Arup Nielsen.

- spvolren: “A Simple and Flexible Volume Rendering Framework for Graphics-Hardware–based Raycasting” by Simon Stegmaier, Magnus Strengert, Thomas Klein.

- Contour-Based Surface Reconstruction using Implicit Curve Fitting, and Distance Field Filtering and Interpolation by Jeffrey Marker et al.

- Marching cubes: author.
- Textured planes: glvol by author.
- Surface relaxation and simplification: author.
- Media rendering: POVRay with media modifications by author.
- Mesh viewing/rendering: GeomView (Geometry Centre), stereo2 by author.