

Assignment (20%)

ICP3029 / ICP4127

Deadline: 11th April 2014

The assignment is due at midnight on Friday 11th April. Submissions must be made via Blackboard. Please take into account upload times and internet connections when considering how much time you have remaining. Late submissions will be dealt with in accordance with University policy.

The assignment is based on a set of functions to build 3D implicit surfaces. These represent contours (or isosurfaces) through a scalar field in 3D. It is a relatively common methods for creating smooth 3D surfaces. Figure 1 shows slices of such a scalar field. It is built using 3 control points. Figure 2 isosurfaces of the same scalar field. They are extracted using the Marching-cubes algorithm with different threshold values.

It is not required to know what implicit surfaces are to complete this assignment. To know more about implicit surfaces, a good tutorial is available at: <http://paulbourke.net/geometry/impliciturf/>.

To extract and build the code that is provided – after having set the development environment on HPC Wales – proceed as follows:

```
$ tar zxvfp MPI_assignment.tar.gz # Decompress the archive
$ cd MPI_assignment               # Go to the assignment directory
$ ls                             # See the content of the directory
$ cd build                       # Go to the build directory
$ make                           # Build the project
$ ./test                         # Run the test program
```

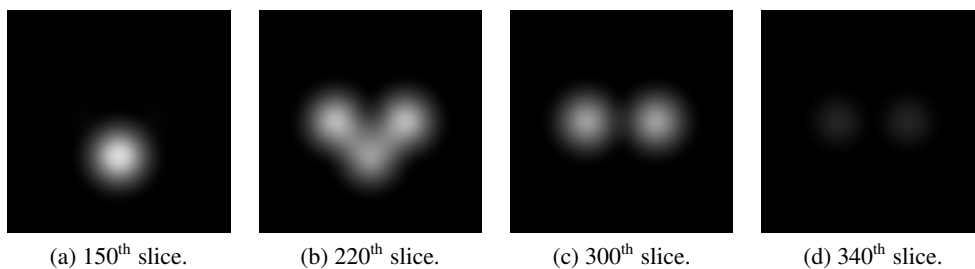


Figure 1: Slices of a scalar field build using 3 control points with the *Meta Balls* density function.

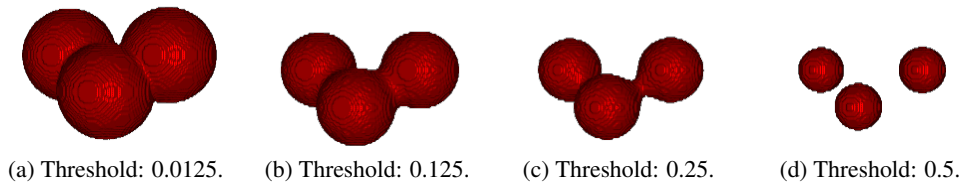


Figure 2: Isosurfaces extracted from the volume presented in Figure 1 using marching cubes with different threshold values.

When you run the test program, a volume data set will be created. It contains $512 \times 512 \times 512$ voxels stored sequentially using floating-point numbers with single precision (i.e. 32 bits). To view the content of the file, you may use ImageJ on your standard PC/laptop at home or in the lab.

We provide the initial working – but inefficient – C++ program (it is not parallelised). The assignment is to make this more efficient (as described in class) using parallel processing with MPI, and provide evidence of such by timing both the original and improved version. Experiments should include:

- Using various numbers of control points,
- Using various numbers of processors, and
- Comparing the performance using the GNU (g++) and Intel (icc) compilers.

Your final submissions should include the code and a report that discusses the changes you made to the program and the experiments showing performance. You should discuss for what problems sizes parallelisation is effective. You should submit results showing speed-up curves. The initial inefficient code is in the file `src/ImplicitSurface.cxx` in the function `void voxelise(...)`.