University of California
Lawrence Livermore National Laboratory

ISCR

Visus Package for Yorick

Daniel Wilhelm, Purdue University

Steven Langer, AX Division
Lawrence Livermore National Laboratory

Parallel I/O, efficient level-of-detail queries, and support for restarts with an arbitrary number of processors are provided as a Visus plug-in for Yorick. The plug-in allows near-interactive visualization of results from pf3d, a laser-plasma interaction code that runs on large supercomputers.

Overview

Visus is a distributed framework enabling real-time visualization by LLNL scientists Valerio Pascucci, Giorgio Scorzelli, and Daniel Laney.

Visus stores visualization data in space-filling curve order (Figure 1) on disk, placing neighboring zones nearby. In typical scan line storage, zones above or below another are far apart on disk, rendering disks' caching abilities ineffective.

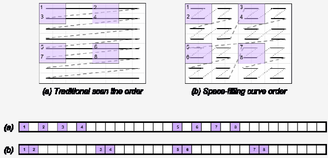

1
2
3
4
5
6
7
8
(a) Traditional scan line order
(b) Space-filling curve order
(a)
(b)

FIGURE 1. Space-filling curves yield serial data order, optimal for parallel file systems.

Yorick, an interpreted scripting language for scientific computing, was developed at LLNL by Dave Munro. In version 1.6, plug-ins are supported, allowing libraries such as MPI to be wrapped and included in scripts without recompilation of Yorick.

Objectives

The first objective of the present project is to provide a set of Visus wrappers as a Yorick plug-in, encapsulating the functionality of the Visus C code within Yorick.

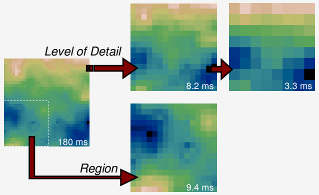
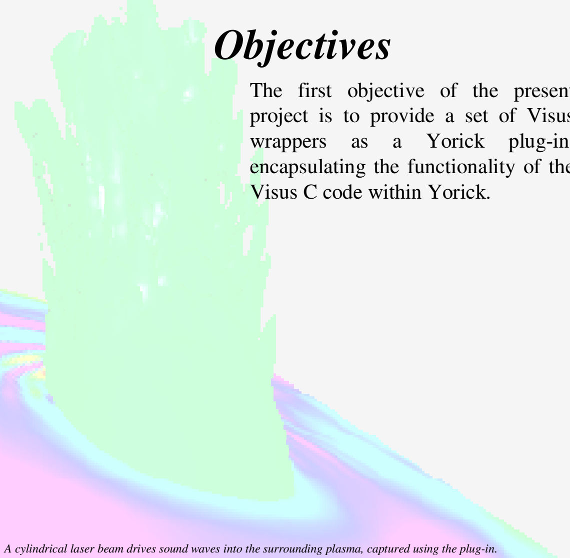


FIGURE 2. The plug-in can retrieve regions and lower detail slices at interactive rates.

The second objective is to enhance low-level functionality of Visus through high-level routines, making it easy for users to employ Visus in current simulations (Figure 2).

The third objective is to apply the plug-in to a real-world application such as pf3d, a laser-plasma interaction code.

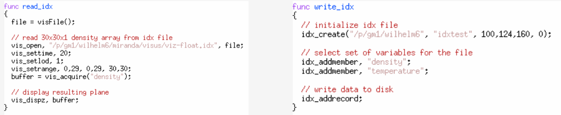


A cylindrical laser beam drives sound waves into the surrounding plasma, captured using the plug-in.

Application — pf3d

An excellent candidate for this plug-in is pf3d, a laser-plasma interaction code. Not only does pf3d already run under the control of Yorick, but it also uses extremely large grids (up to 12.7 billion zones).

The plug-in provides pf3d users with high-level functions to write Visus dumps (Figure 3), and it supports post-processing on single processors and multiple processors (using Yorick's MPI package, MPY).



```
func read_idx
{
  file = visFile();

  // read 30x30x1 density array from idx file
  vis_open, "/p/gm1/wilhelm6/miranda/visus/viz-float.idx", file;
  vis_settime, 20;
  vis_setlod, 1;
  vis_setrange, 0,29, 0,29, 30,30;
  buffer = vis_acquire("density");

  // display resulting plane
  vis_dispz, buffer;
}
```

```
func write_idx
{
  // initialize idx file
  idx_create("/p/gm1/wilhelm6", "idxtest", 100,124,160, 0);

  // select set of variables for the file
  idx_addmember, "density";
  idx_addmember, "temperature";

  // write data to disk
  idx_addrecord;
}
```

FIGURE 3. Sample code using the plug-in, illustrating reading and writing IDX files from Yorick. (Visus writes files in the IDX format.)

Results

The Visus package was successfully tested to read and write IDX files, and data exploration at low detail levels is near-interactive (Figure 4). As expected, the reading data throughput rate is roughly constant with buffer size (Figure 5; the anomalous right-most datapoint is a temporary Visus bug).

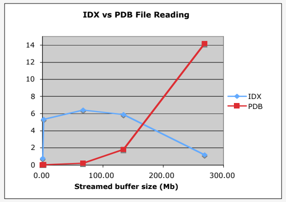


FIGURE 5. At low levels of detail, IDX files provide higher transfer rates than PDBs.

High-level routines support the easy display of retrieved data, including routines which produce slicing-plane animations.

Iso-surfaces and other visualization techniques are also supported, and a package to convert pf3d's existing PDB files to IDX format is planned for future work.

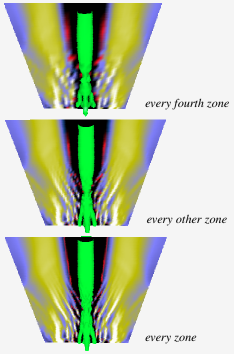
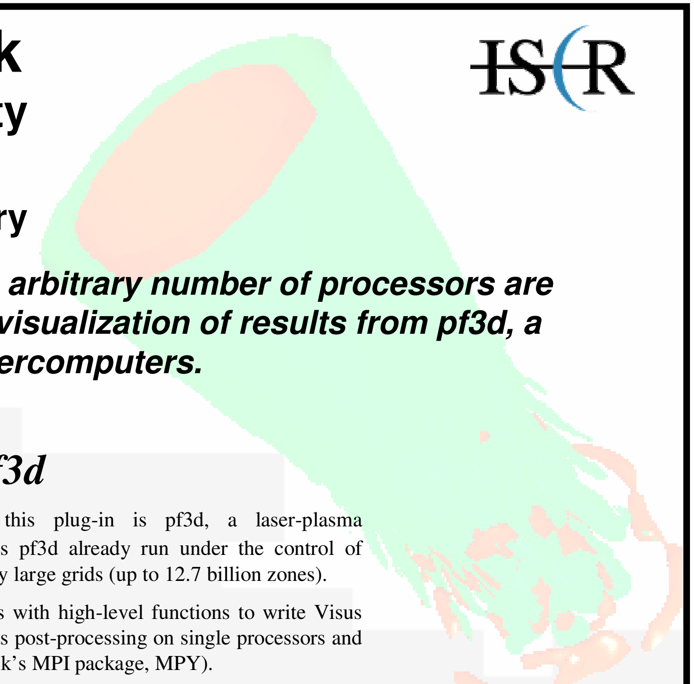


FIGURE 4. The user can view key features of the dataset quickly at a low level of detail.



In the upper-right background, two iso-surfaces from a high-intensity laser beam are captured using the plug-in.

This work was performed under the auspices of the U.S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

UCRL-POST-214354