



Nicolas Rougier

Researcher in Computational Neurosciences
INRIA / Institute for Neurodegenerative Diseases

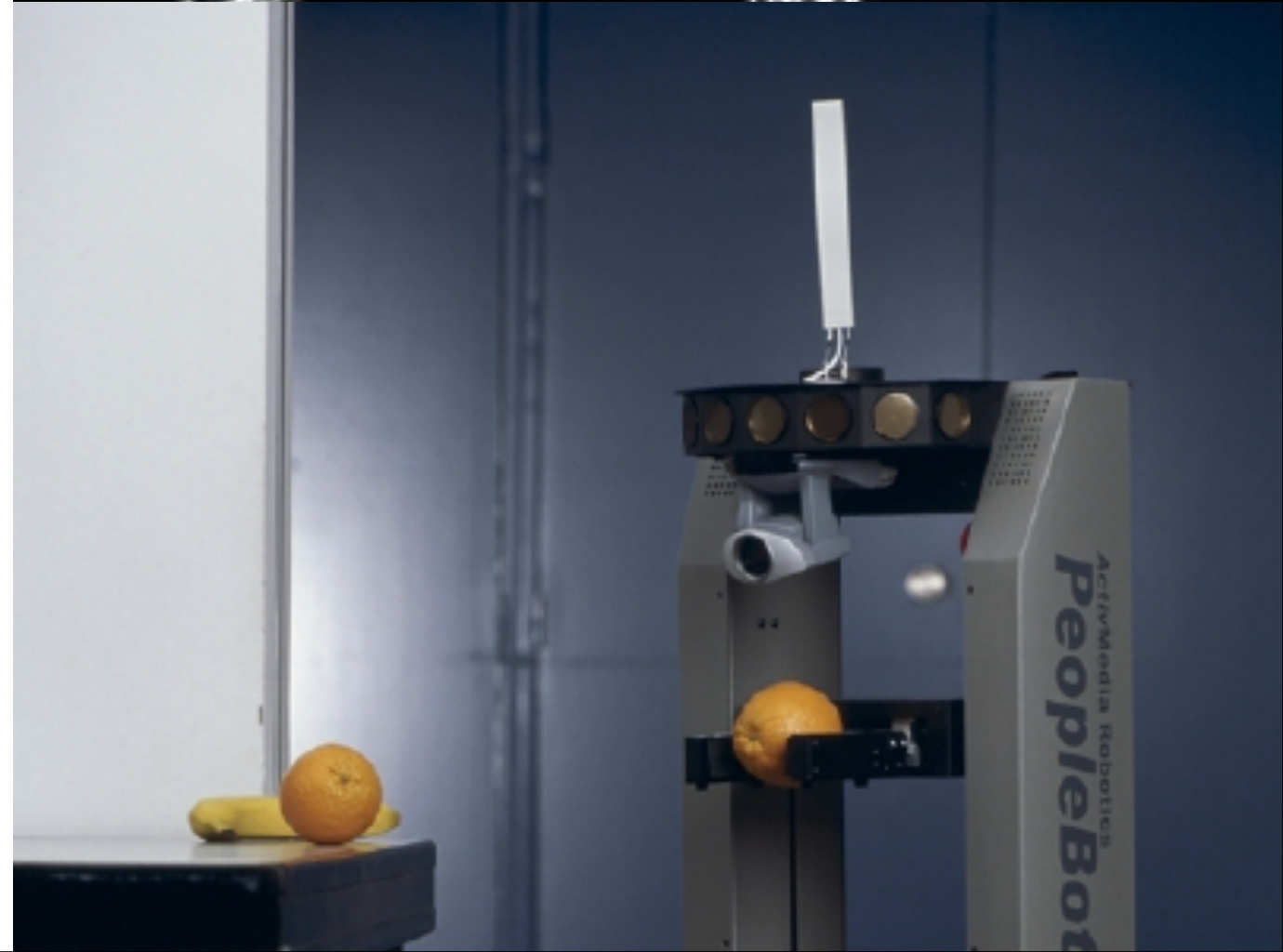
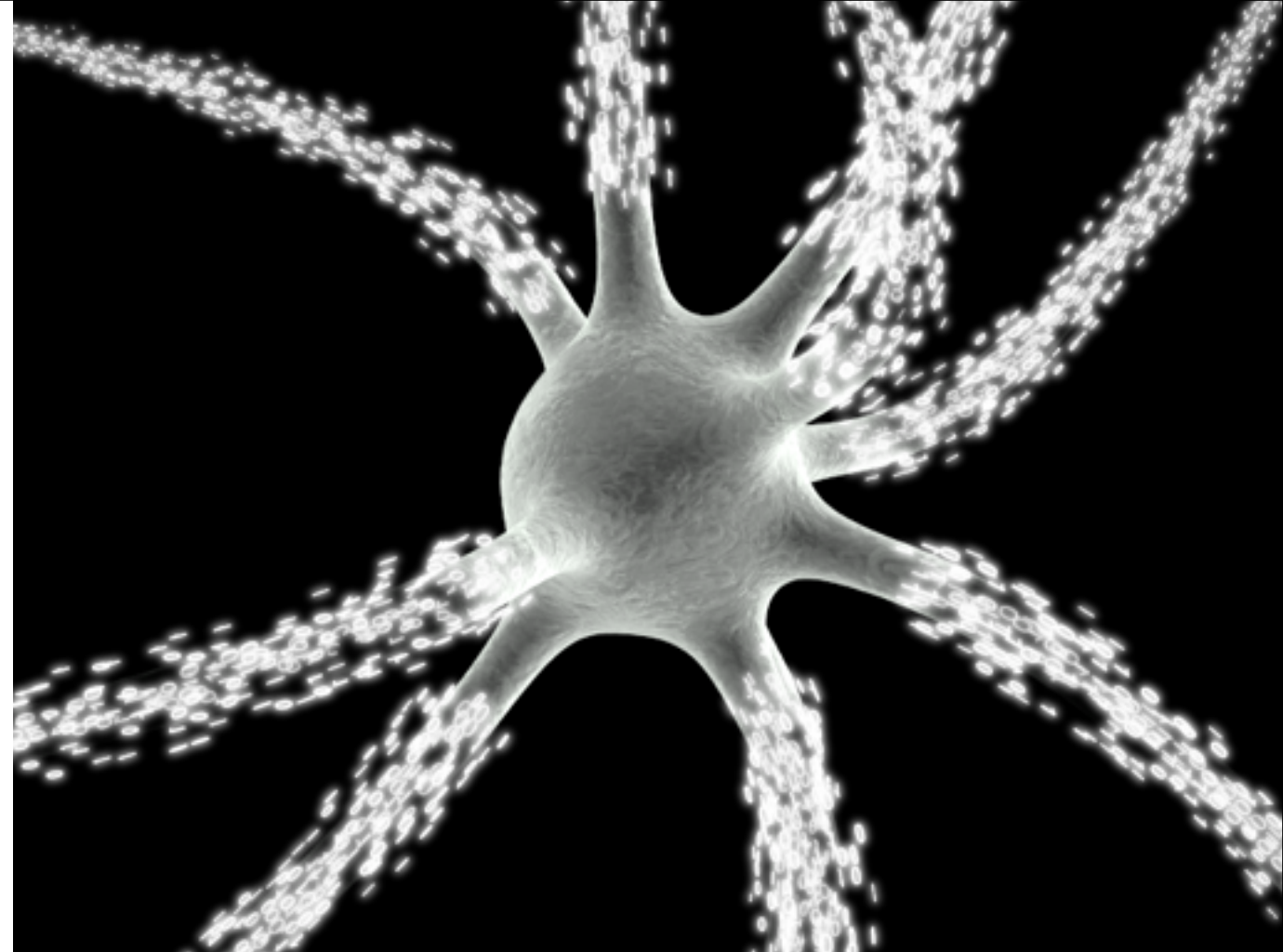
Biographical Sketch

- 2013- Full-time researcher
Mnemosyne project, INRIA, Institute for Neurodegenerative Diseases, Bordeaux
- 2011 Habilitation à Diriger les Recherches
- 2002-2012 Full-time researcher
Cortex project, INRIA, Nancy, France
- 2000-2002 Associate Researcher
University of Colorado, Boulder, USA
- 1997-2000 Ph.D, Computer Science
Université Henri Poincaré, Nancy, France

Research

My research activities attempt

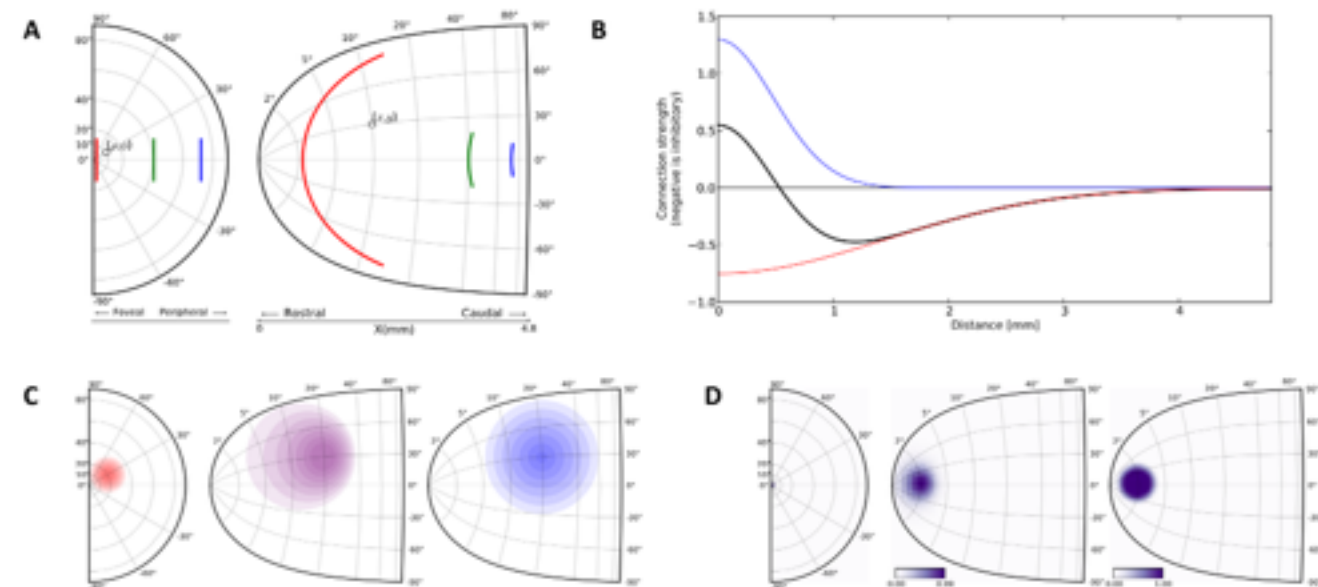
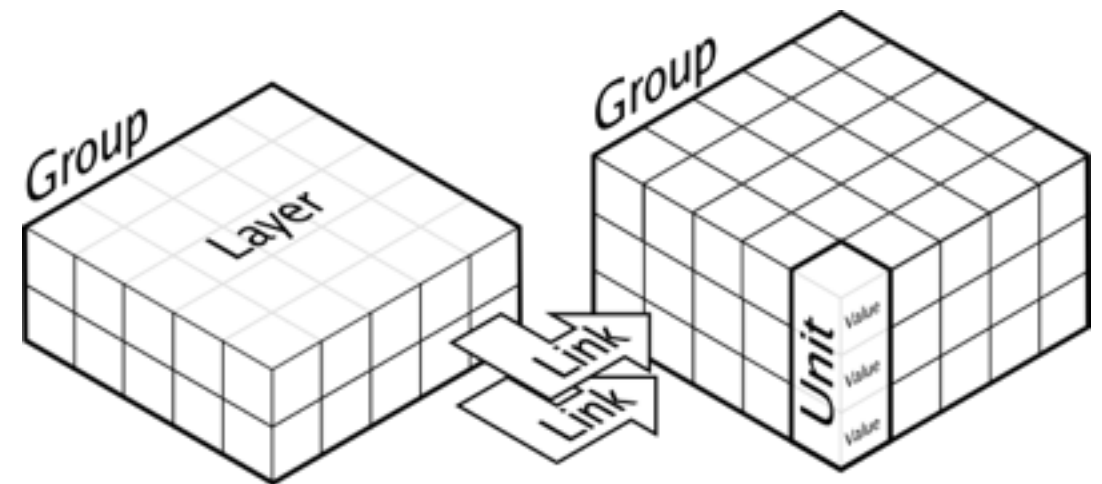
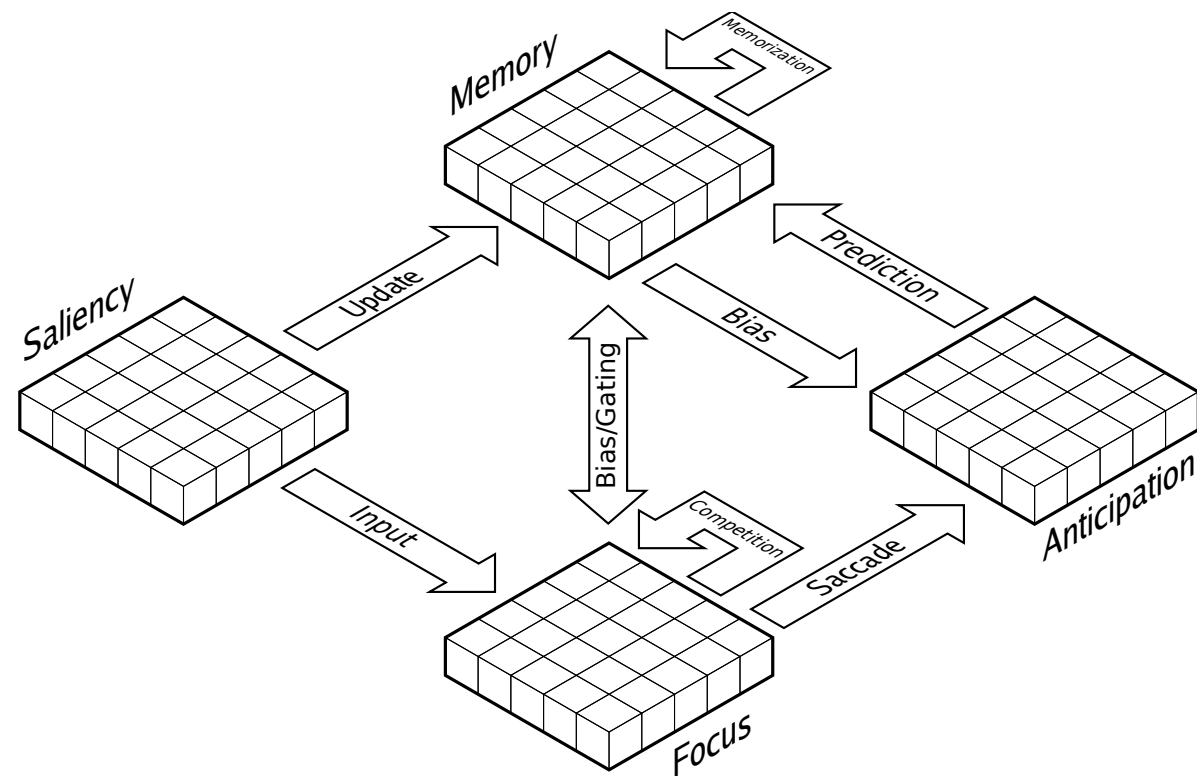
- to understand higher brain functions
- to develop computational models of brain structures
- to emulate behaviour using robotic bodies



Distributed Asynchronous Numerical Adaptive

dana.loria.fr

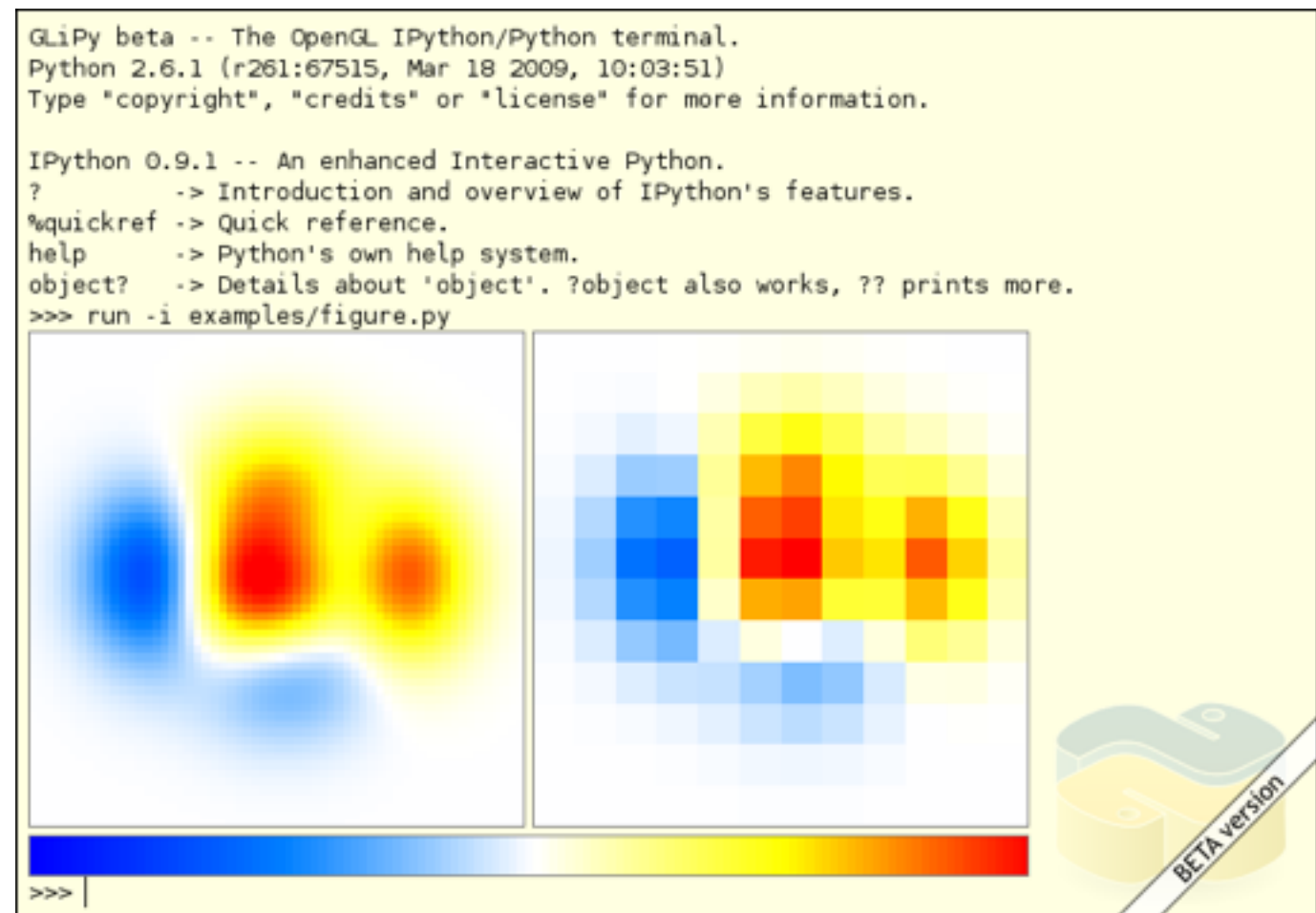
- Intensive computation
- No analytical model
- No time to lose in visualisation
- Visualisation needs to be **fast**



Glipy (python, 2007)

www.loria.fr/~rougier/coding/glipy/index.html

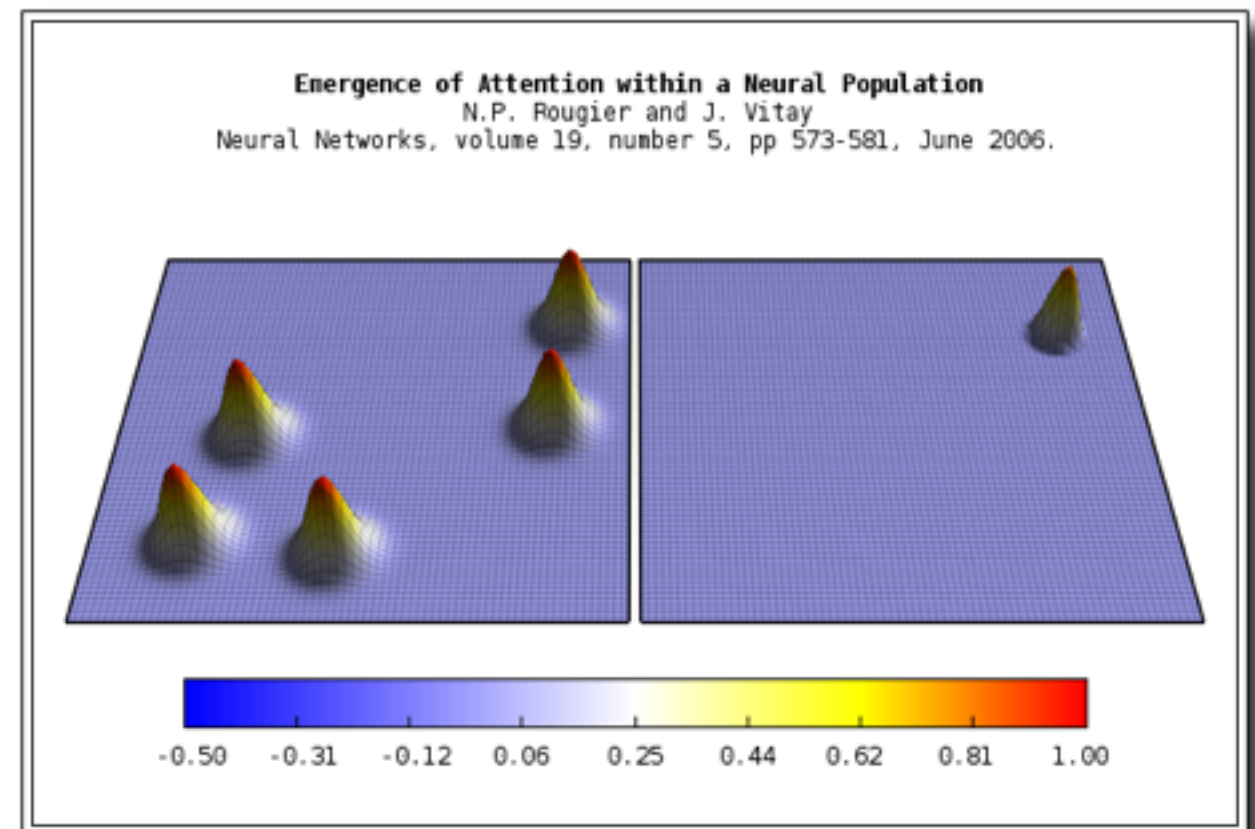
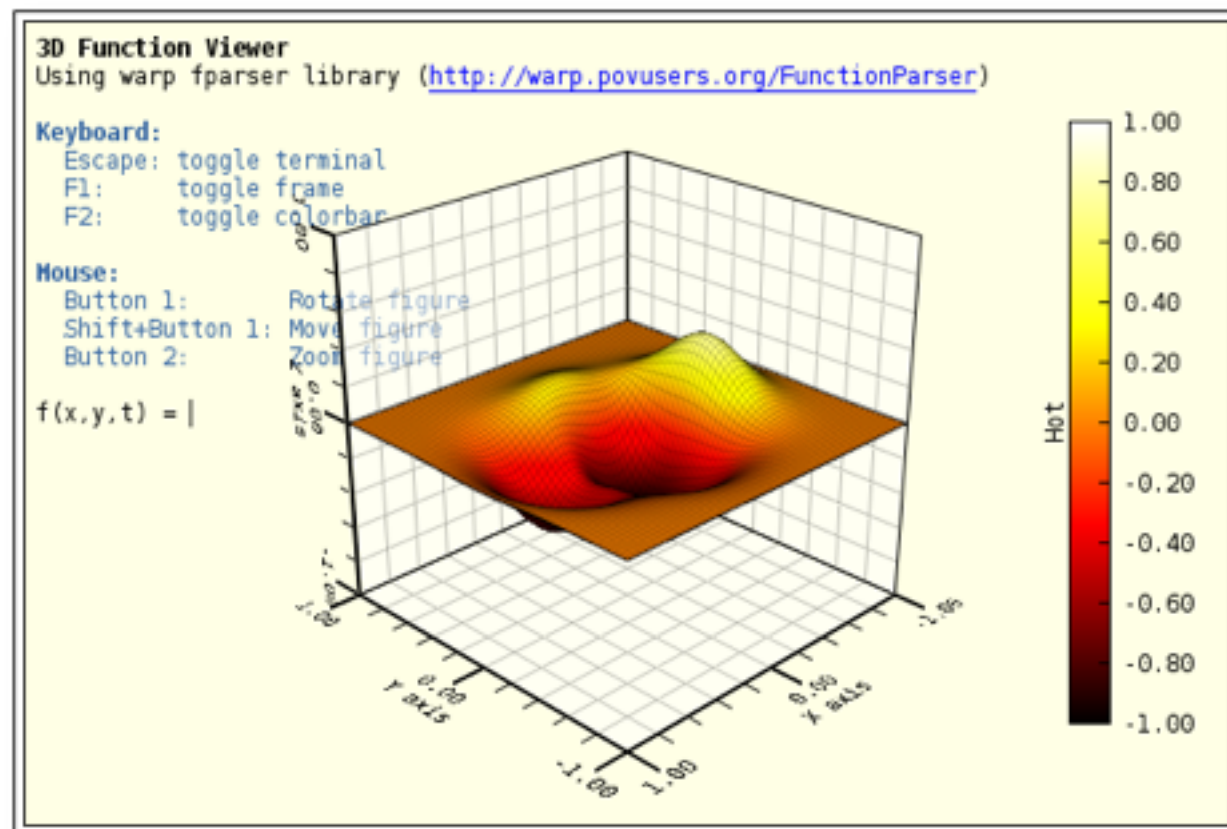
The goal of glipy is to create a comprehensive environment for interactive and exploratory computing using an interactive Python terminal and an architecture for embedding various graphical elements directly within the terminal.



SciGL (C++, 2008)

www.loria.fr/~rougier/coding/scigl/index.html

SciGL (Scientific OpenGL Visualization ToolKit) aims at facilitating the development of scientific visualization by providing a set of classes for rapid prototyping of scientific visualization software.



Glumpy (python, 2011)

code.google.com/p/glumpy/

glumpy is a small python library for the rapid vizualization of numpy arrays, (mainly two dimensional) that has been designed with efficiency in mind.

BSD LICENSED

GLUMPY

OPENGL+NUMPY

FAST NUMPY VISUALIZATION USING OPENGL
HTTP://CODE.GOOGLE.COM/P/GLUMPY/
CONTACT: NICOLAS.ROUGIER@INRIA.FR

SOMEWHERE BETWEEN Numpy...
... AND MATPLOTLIB

WOW !!!
IT'S REALLY SUPER FAST !

SPARE INTERPOLATION ON THE GPU:
LINEAR, BILINEAR AND CUBIC

LINEAR, LEVELING,
LIGHTING ARE DONE
USING SHADERS

NUMPY 2D FLOAT ARRAY CAN BE RENDERED AS A
HEIGHTFIELD WITH 150 LINES.

REAL TIME FLUID
SIMULATION ...

... IN PURE PYTHON
(NUMPY+SCIPY+GLUMPY)

RENDER 2D FLOAT ARRAY...

SEAMLESS INTERPOLATION
WITH MATPLOTLIB
USING THE AGG BACKEND

ANY VARIABLE
CAN BE EDITED

WHILE
ARRAY IS
RENDERED BY
GLUMPY

FRAME IS
RENDERED BY
MATPLOTLIB

BASIC GUI CONTROLS VIA
THE ANTIWISDOM WINDOW

MINIMAL SCRIPT TO DISPLAY AN ARRAY

```
import numpy, glumpy
window = glumpy.Window(512,512)
t = numpy.random.random((512,512))
c = glumpy.Color(1,1,1)

window.event
def on_draw():
    window.clear()
    t.blit(0,0,window.width,window.height)
window.run_forever()
```

GLUMPY IS A SMALL PYTHON LIBRARY FOR THE RAPID VISUALIZATION OF NUMPY ARRAYS (MAINLY TWO DIMENSIONAL) THAT HAS BEEN DESIGNED WITH EFFICIENCY IN MIND. IF YOU WANT TO DRAW NICE FIGURES FOR INCLUSION IN A SCIENTIFIC ARTICLE, YOU'D BETTER USE MATPLOTLIB. IF YOU WANT TO HAVE A SENSE OF WHAT'S GOING ON IN YOUR SIMULATION WHILE IT IS RUNNING, THEN MAYBE GLUMPY CAN HELP YOU.

GLUMPY USES OPENGL TEXTURES TO REPRESENT ARRAYS SINCE IT IS PROBABLY THE FASTEST METHOD OF VISUALIZATION ON MODERN GRAPHIC HARDWARE. HOWEVER, THE DRAWBACK IS THAT IT IMPLIES SOME RESTRICTION ON THE TYPE AND SHAPE OF ARRAYS THAT CAN BE VISUALIZED USING THIS METHOD. THE DTYPE OF ARRAY MUST BE ONE OF NUMPY.INT8 OR NUMPY.FLOAT32 AND THE SHAPE OF THE ARRAY MUST BE ONE OF M, MxN OR MxNxN. APART FROM PURE RENDERING PERFORMANCES, OPENGL TEXTURES OFFER THE ADVANTAGE OF BEING ABLE TO USE SHADERS THAT CAN ALTER THEIR RENDERING. GLUMPY USES SUCH SHADERS TO IMPLEMENT COLOR LOOKUP TABLES (I.E. COLORMAP), FILTERING (NEAREST / BILINEAR / CUBIC) AND DISPLACEMENTS (HEIGHTMAPS). IN OTHER WORDS, RENDERING IS DONE ENTIRELY ON THE GRAPHIC CARD, SAVING CPU TIME FOR SIMULATION.

TO BE CONTINUED...

FAST & SIMPLE

TURNING PATTERNS USING DUNE (HTTP://DUNE.LORIA.FR)

... DISPLAYED IMMEDIATELY

ANY ARRAY MODIFICATION IS ...

INTERACTIVE SESSION USING THE PYTHON SHELL

Freetype-gl (C, 2012)

github.com/rougier/freetype-gl

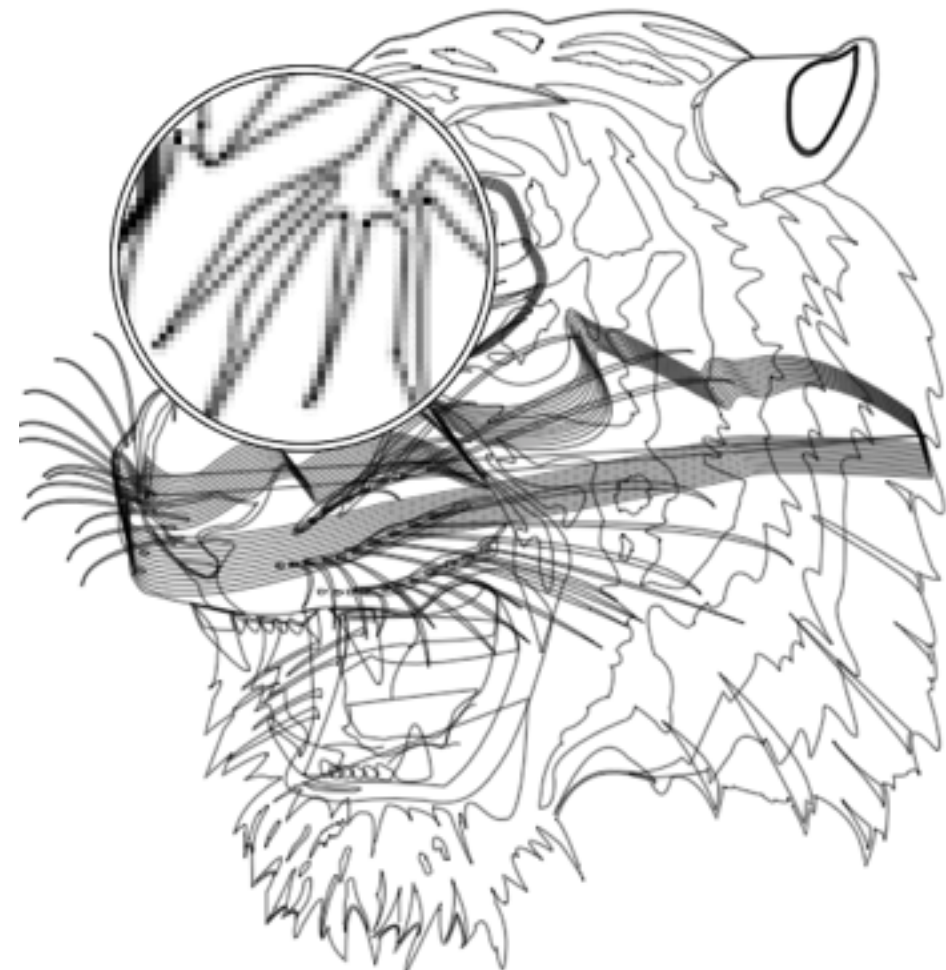
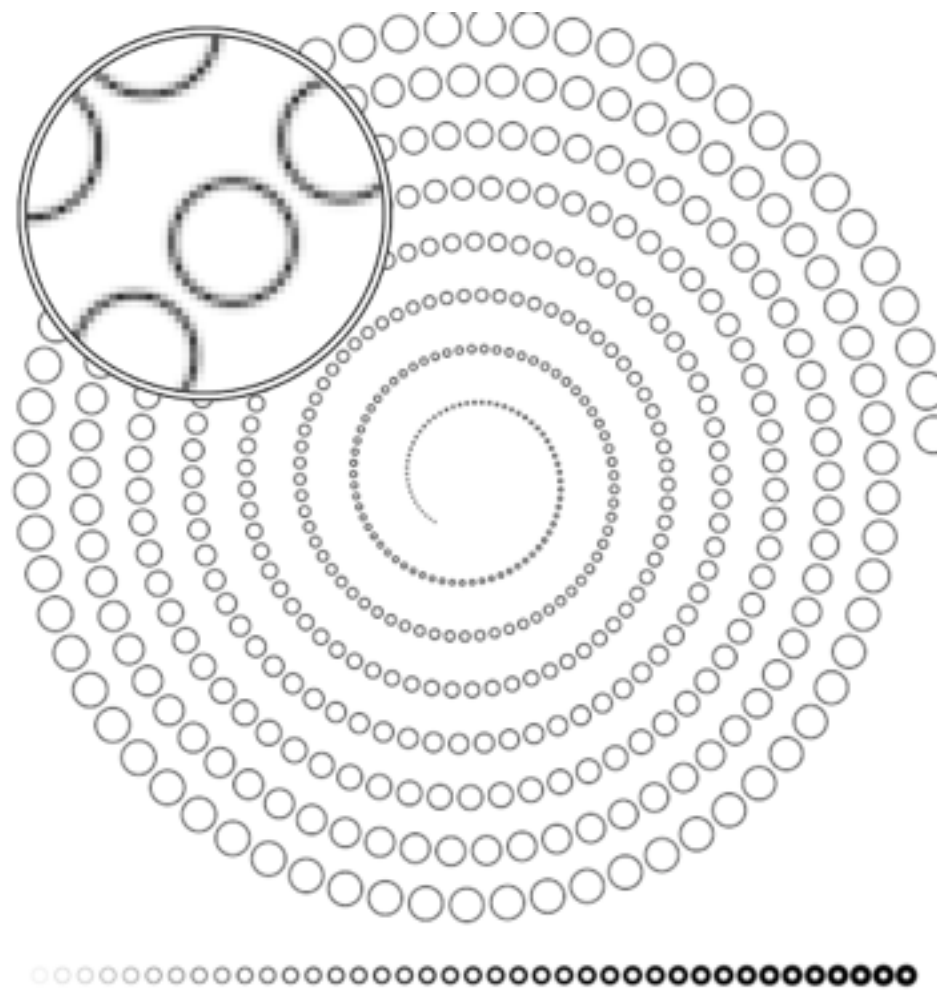
A quick OpenGL/Freetype example for displaying a unicode text using a (single) vertex buffer. The idea is simply to tightly pack every necessary glyphs into a single texture and to generate a single vertex buffer to draw the text.

[illegible]

gl-agg (python, 2013)

github.com/rougier/gl-agg

A quick OpenGL/Freetype example for displaying a unicode text using a (single) vertex buffer. The idea is simply to tightly pack every necessary glyphs into a single texture and to generate a single vertex buffer to draw the text.



Related Bibliography

- *Ten Simple Rules for Better Visuals*
N. P. Rougier & P.E. Bourne, PLOS Computational Biology (submitted)
- *Shader-based Antialiased Dashed Stroked Polylines*
N. P. Rougier. Journal of Computer Graphics Techniques, 2.2 (2013).
- *Higher Quality 2D Text Rendering*
N. P. Rougier. Journal of Computer Graphics Techniques, 2.1 (2013).