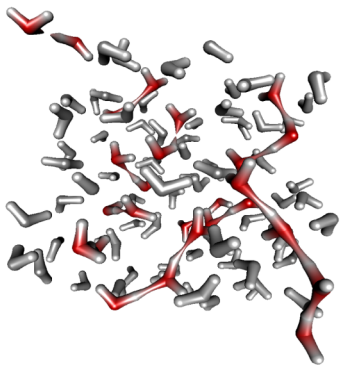
A Rendering Method for Small Molecules up to Macromolcular Systems:

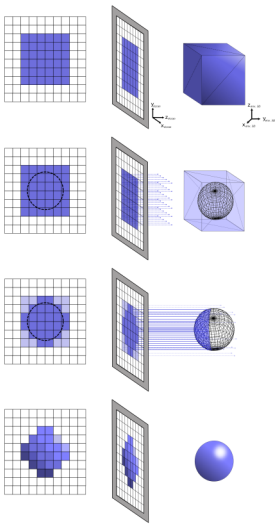
HyperBalls Accelerated by Graphcs Processors

M. Chavent[1], A. Vanel[2], A. Tek[3], B. Levy[4], B. Raffin[2], M. Baaden[31]

[1]CEA, DAM,DIF, 91297 Arpajon, France.
[2]ID-IMAG, CNRS/INPG/INRIA/UJF, Grenoble, France.
[3]Equipe ALICE, Nancy Université, LORIA, 54506 Vandoeuvre-les-Nancy Cedex, France.
[4]Institut de Biologie Physico-Chimique, Laboratoire de Biochimie Théorique, CNRS UPR 9080, F-75005 Paris, France.

H-dond network between water molecules





Ray-Casting Principle

GPU Ray-Casting Principle

Let us consider the example of rendering a sphere (see Figure on the left). In this approach, it is first necessary to represent a simple triangulated envelope such as a cube using OpenGL. This envelope defines a set of pixels on the screen. For each pixel on the screen, we cast a ray using the pixel as origin. Within the envelope, we define a surface equation (sphere) and calculate the potential intersections between the rays and this surface. If the ray intersects the sphere, the original pixel is kept, otherwise the pixel is discarded. Then, using the equation of the sphere, it is possible to define the surface lighting. The interest of this method is that, except for the initial creation of the cubes, the entire process is calculated on the GPU. There are several advantages:

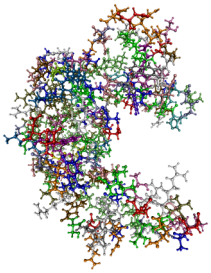
1- Creation of a smooth surface with few triangles.
2- Pixel-accurate surface for all levels of zoom.
3- Accelerated calculations using the GPU capabilities.

This principle has already been used to represent molecular surfaces [1]. Here we extend it in order to represent different types of molecular metaphors.

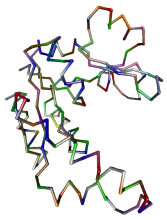
Molecular Representations

Classical representations such as *Ball & Stick*, *Licorice* or *Van der Waals* can be implemented with this method. We have designed a new representation that we named ***HyperBalls***. With this depiction, cylinders are replaced by hyperboloids that can smoothly connect the atom spheres. Interestingly, it is possible to adapt the hyperboloids in order to depict dynamic bond evolution (see water network example at the top) which is not possible using cylinders. Furthermore, this representation can be adapted to represent coarse grained or spring network models (see picture on left). Another application of this visualization is to represent ion coordination. Here, we depict links between the ion and other molecules.

As illustrated by various examples on this poster, it is possible to color molecular metaphors by atom type, chain name, residue name, residue type (not shown) or other user defined colors.

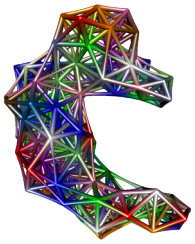


All atoms Ball & Stick



Trace

Coarse Grained Model



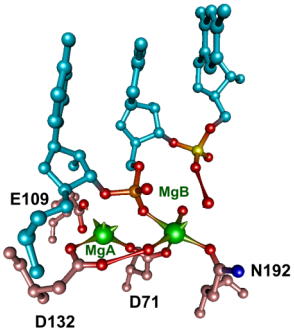
Spring Network Model

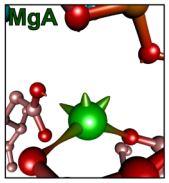
Macromolecular Systems Visualization

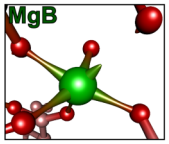
As explained above, this method is most efficient to interactively display a large number of atoms, achieving a high image quality because it requires less triangles to represent atoms or bonds. Thus, if we compare our performance results -measured in frames per second - with existing software, we manage to display the same molecules 5 to 45 times faster using our approach (compared to VMD or PyMol). Such a gain in performance can allow the user to interactively explore huge molecular structures. Furthermore, the performances increase using new grahic cards generation with exactly the same source code (see Diagrams bellow).

To conclude, we present an efficient method to interactively visualize a broad range of molecules, from small atomic structures to huge macromolecular assemblies with a high-quality of rendering. We hope to be able to soon release a freely available molecular viewer based on this method. Please check http://www.baaden.ibpc.fr/ projects/fvnano regularly. For the moment, demos or videos are available on demand.

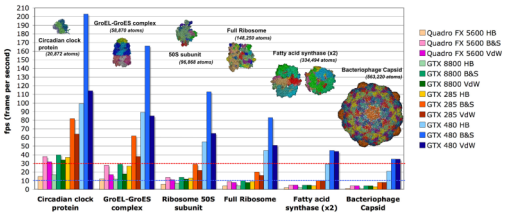
Mg ion coordination visualization

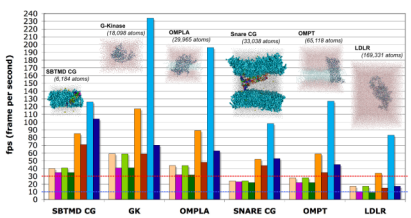




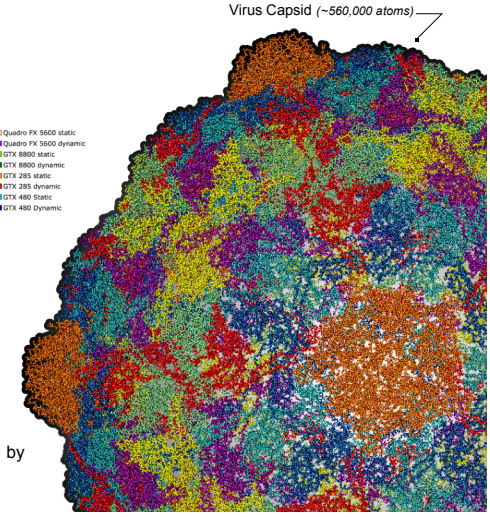


GPU Ray-Casting method performances *(using different Nvidia graphic cards on static and dynamic datasets)*





Virus Capsid *(~560,000 atoms)*



References:

1. Chavent, M.; Levy, B.; Maigret, B. J Mol Graph Model 2008, 27(2), 209-216.

Acknowledgments:

We would like to thank J.P. Nominé (CEA / Bruyère le Chatel) as well as M. Gueroult (DSIMB). This work was supported by an ANR funding (ANR-07-CIS7-003-01).