Remote scientific visualization of progressive 3D meshes with X3D

Adrien Maglo
Céline Hudelot
Ecole Centrale Paris, MAS

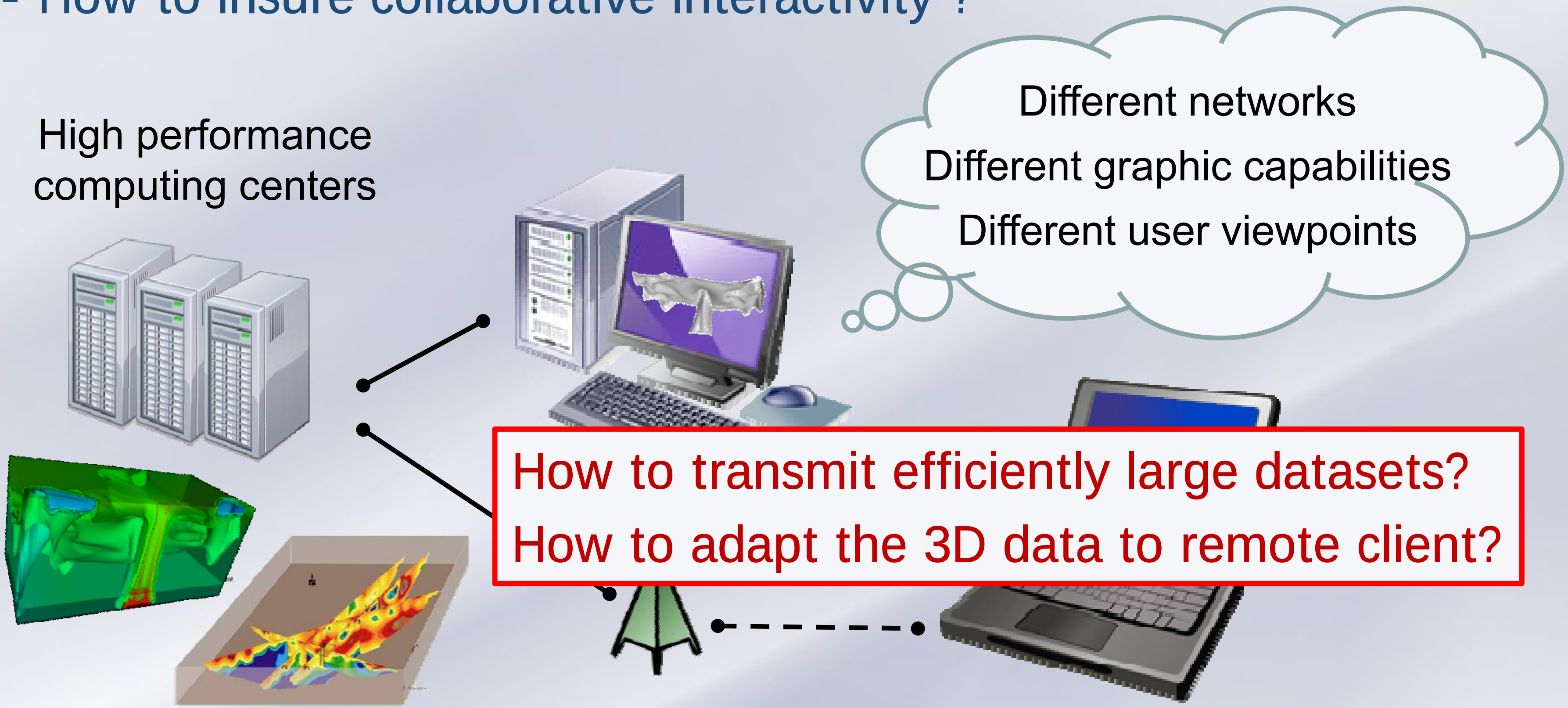
Guillaume Lavoué
Ho Lee
Florent Dupont
Université de Lyon, CNRS, LIRIS

Christophe Mouton
EDF R&D

Context

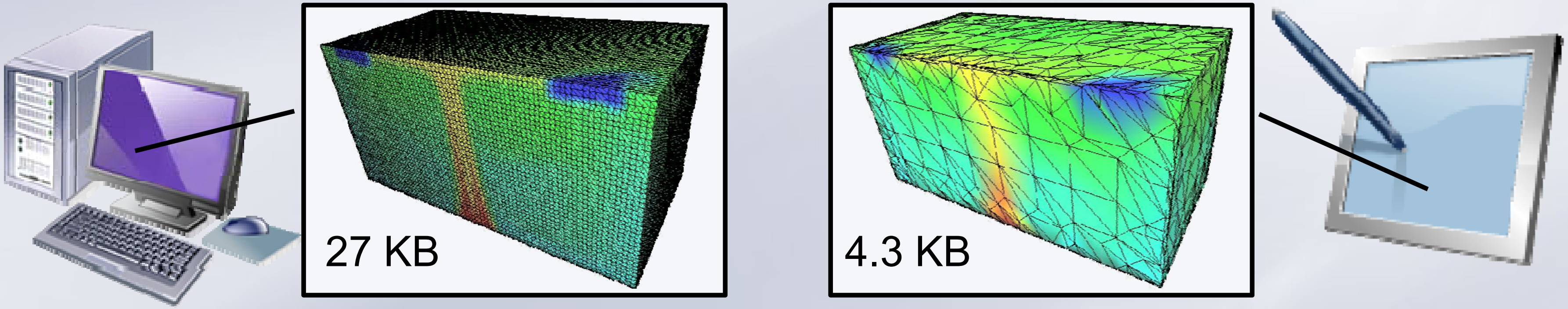
Remote visualization of large 3D scientific datasets

- Scientific simulation lead to gigantic 3D data (1 billion elements)
- Data produced in HPC center but analysed by remote users
- How to insure collaborative interactivity ?



Our proposal: A 3D streaming framework

The idea is to adapt the Level Of Detail according to the user, the bandwidth, the device capabilities



High performance PC
High bandwidth network

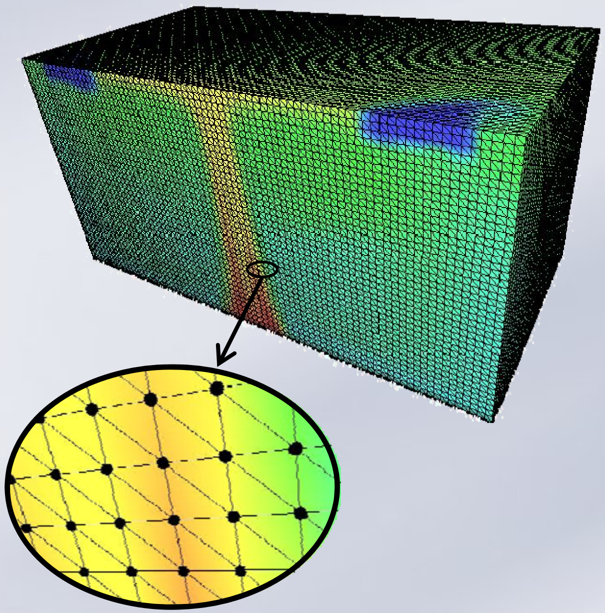
Mobile device
Wireless network

Our framework:

- A progressive mesh compression method for 3D mesh with colors
- Adaptation mechanisms dealing with different constraints (network bandwidth, display capability, etc.)
- A solution for the integration into X3D

The 3D data to visualize

Usually 3D surfacic mesh with color



Geometry	**Connectivity**	**Color**
V1 = (x_1, y_1, z_1)	**F1 = (1, 3, 4)**	**C1 = (r_1, g_1, b_1)**
V2 = (x_2, y_2, z_2)	**F2 = (1, 2, 4)**	**C2 = (r_2, g_2, b_2)**
V3 = (x_3, y_3, z_3)	**F3 = (2, 3, 4)**	**C3 = (r_3, g_3, b_3)**
V4 = (x_4, y_4, z_4)	**F4 = (1, 2, 3)**	**C4 = (r_4, g_4, b_4)**
....	**....**	**....**

File size:

X3D (*<IndexedFaceSet>*) = 953 KB

ZIP = 219 KB

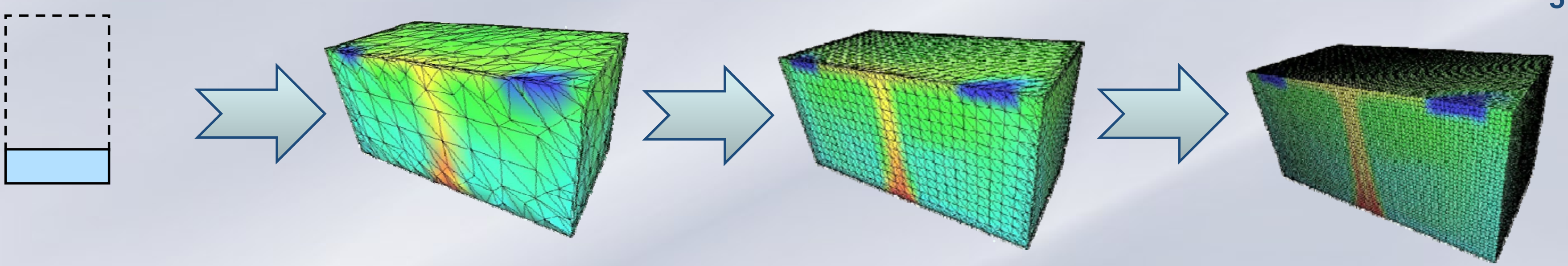
Using mesh compression [Lee et al., 2010] = 27KB

Progressive mesh compression

Encoding

Size reduction

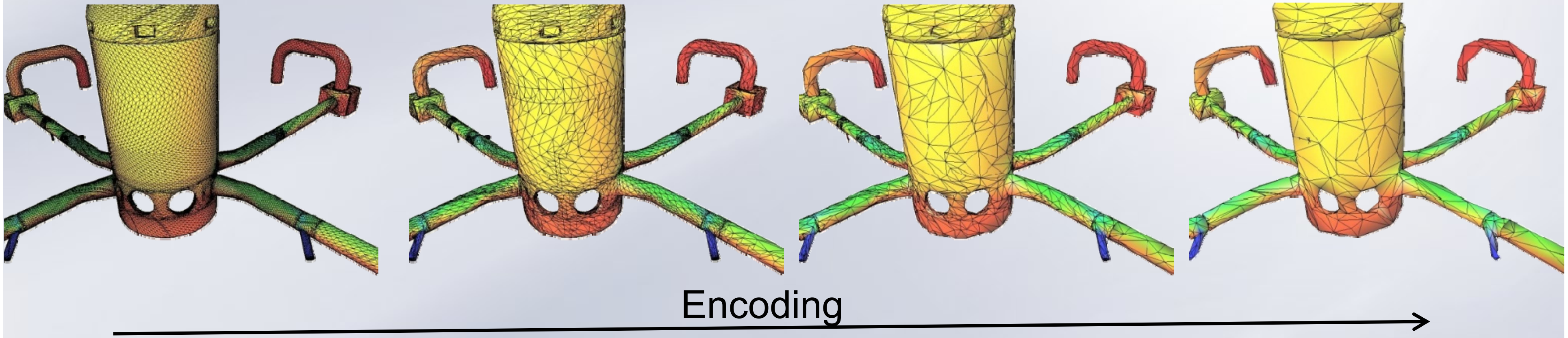
Decoding



Level of Details management

Progressive mesh compression

The mesh is iteratively simplified....



...and the information necessary to reverse the process are encoded

Original mesh = Base mesh + sequence of refinement

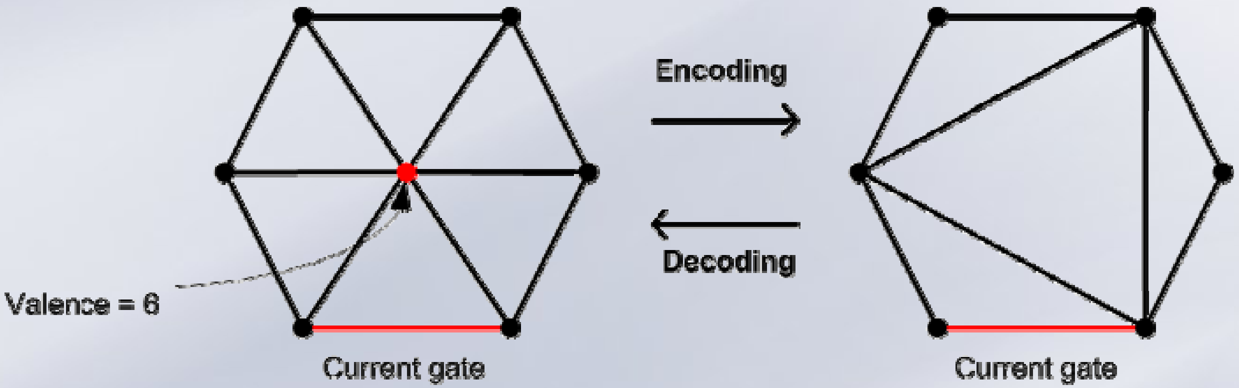
Progressive Meshes – Hoppe 1996
Progressive Forest Split – Taubin et al. 1998
Valence-driven Approach – Alliez and Desbrun 2001
Kd-tree Decomposition – Gandoin and Devillers 2002
Octree Decomposition – Peng and Kuo 2005
Incremental Parametric Refinement – Valette et al. 2009

Color
Lee et al., 2010
Cirio et al. 2010
Cai et al, 2007

The compression algorithm

Based on iterative simplification [Alliez and Desbrun, 2001]

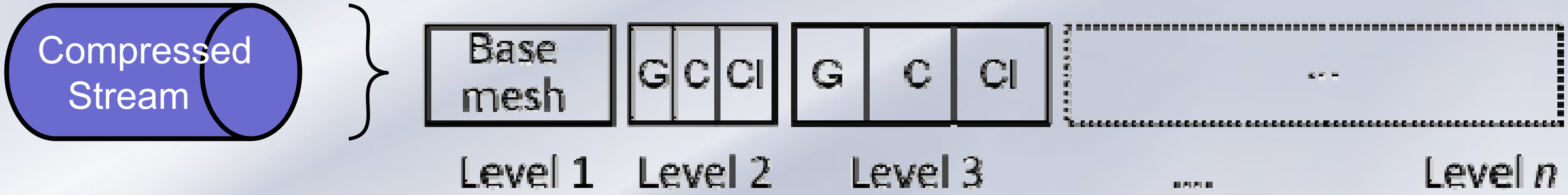
Adaptive quantization + color prediction scheme [Lee et al. 2010]



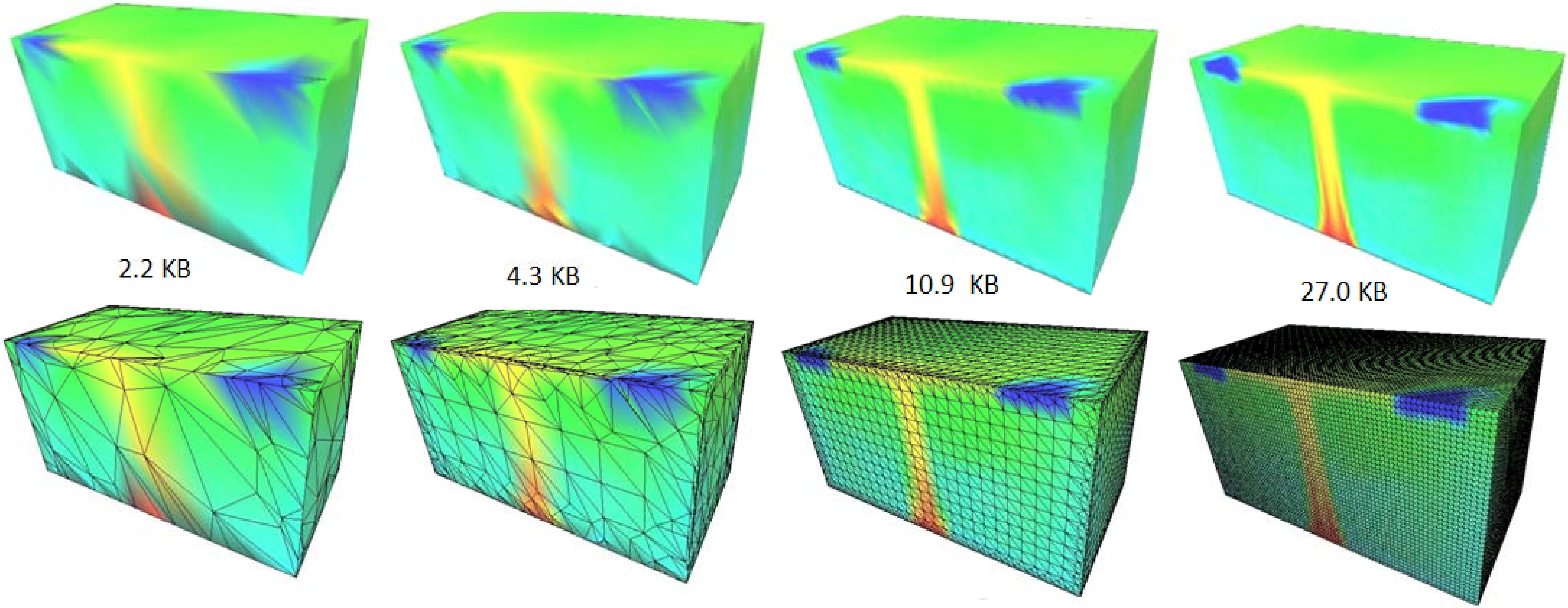
Iterative decimation of a set of independent vertices

When a vertex is removed the patch is retriangulated

Connectivity (C), geometry (G) and color (Cl) are encoded



Decoding example

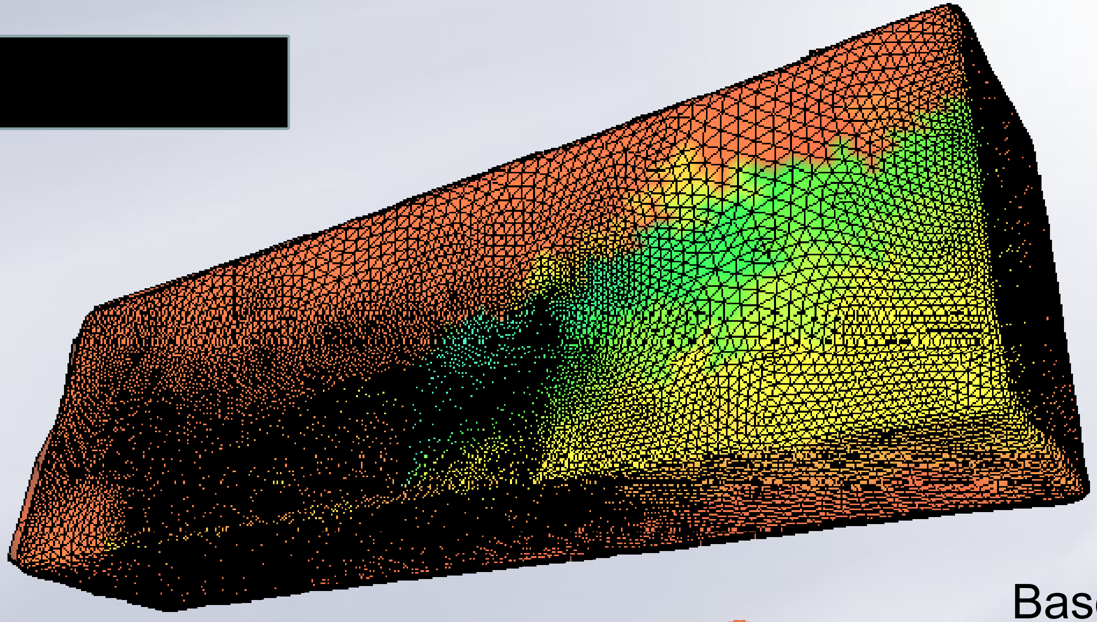


Original X3D size = 953 KB

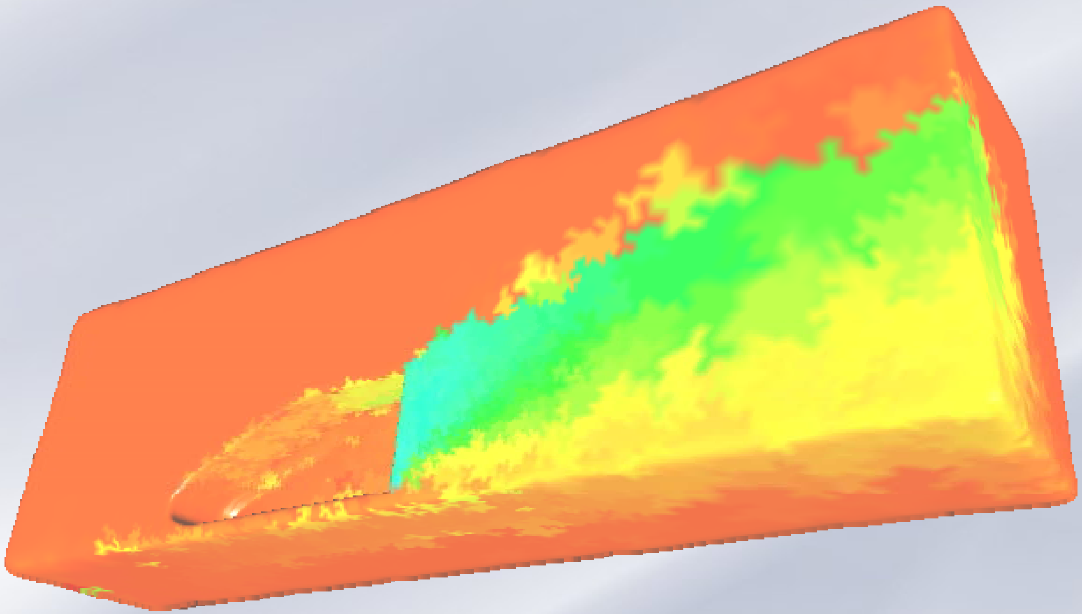
Decoding example

[Lee et al., 2010]

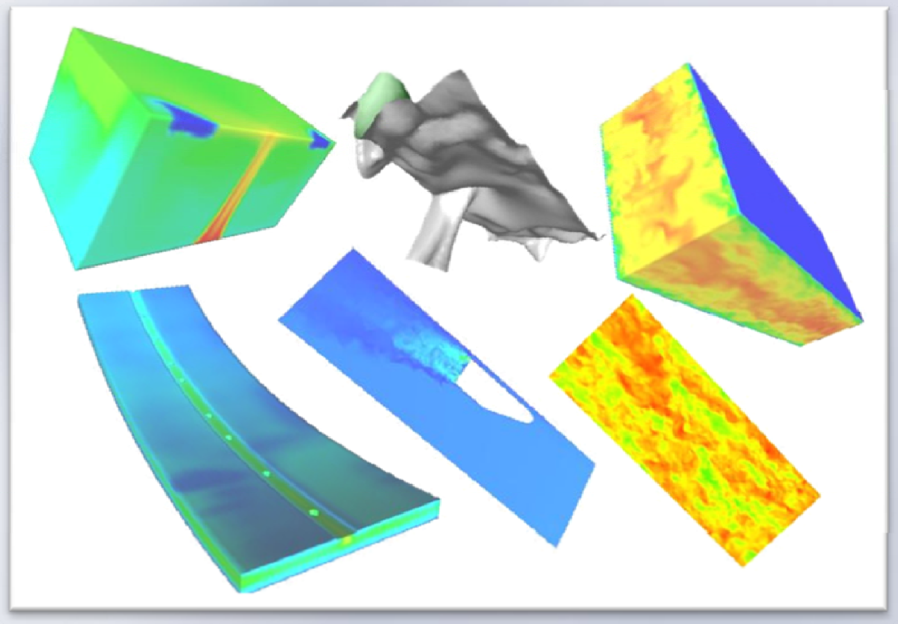
Original = 2743KB



Base mesh



Compression results



3D models coming from Fluid dynamics and geophysics

Compression ratio around 3-4%
Gzip gives around 15-20%

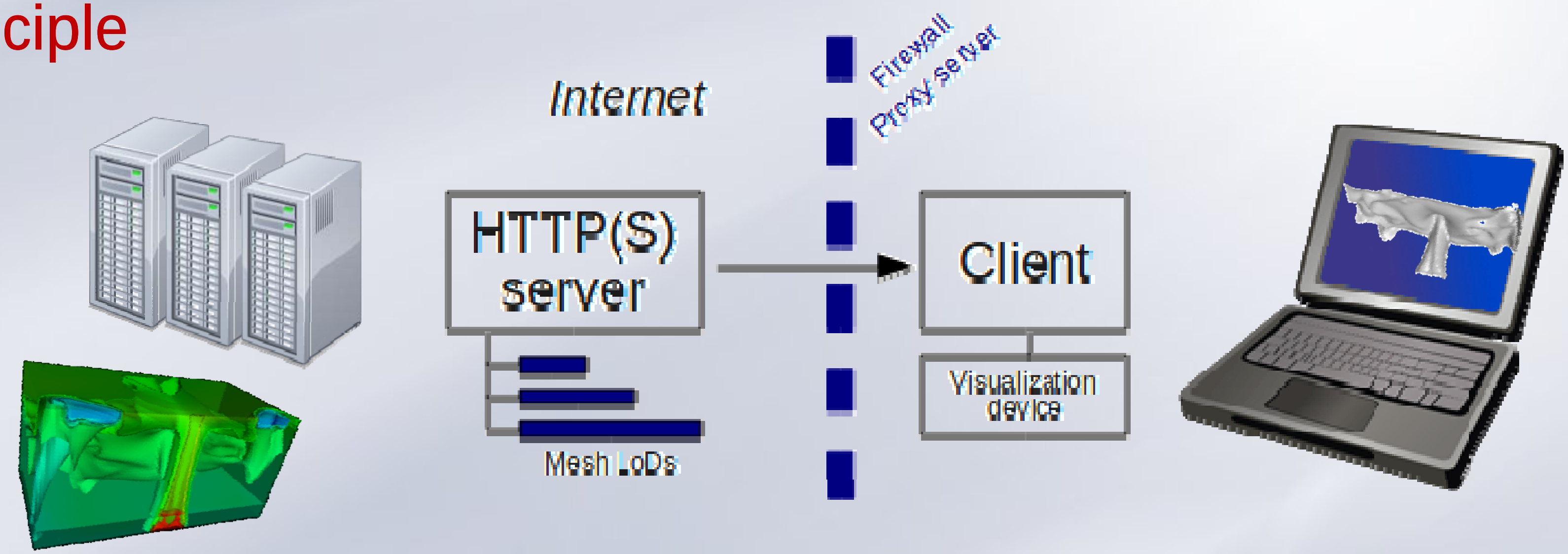
Whole decompression time between 0.15 seconds (4.3 Kvertices) and 6 seconds *(123 Kvertices)*

Number of levels between 10 and 26

Name	Original (KB)	Compressed (KB)	Ratio
velocity_CP	7739	227.9	2.94%
velocity	2979	83.4	2.80%
radiator	953	27.0	2.83%
radiator_Iso	713	19.6	2.75%
vistcurb_CP	354	16.0	4.51%
tank	10065	266.0	2.64%

Our adaptation/streaming framework

Principle



Download (using HTTP protocol) and render at each instant the most relevant level of details of each 3D object

How to chose the best Level Of details? It depends on:

- The network bandwidth
- The terminal graphic capabilities
- The user viewpoint.

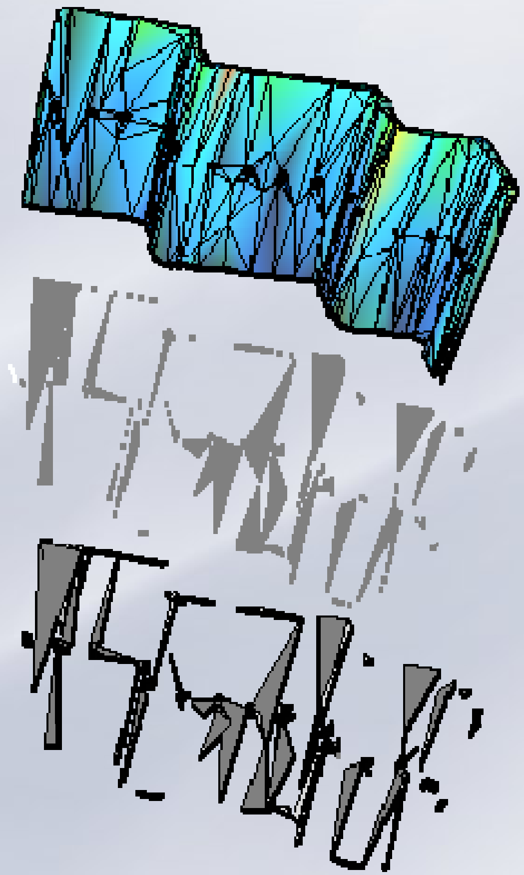
Our adaptation parameters

- ❑ Total downloading time,
- ❑ Total quantity of memory taken by all LOD data,
- ❑ 3D scene rendering time,
- ❑ Element distinction metric

Based on off screen rendering

1. Select random triangles
2. Filled mode rendering:
   – Pixel counting → n_2
3. Line mode rendering:
   – Pixel counting → n_1
4. Metric computation :

$$f_{ED} = \frac{n_1}{n_2}$$



The adaptation algorithm

Triggered by events (view point change, object visibility modifications, object mesh refinement or coarsening ...)

At each event, 3 choices (driven by the rendering time):

- Do nothing.
- Coarsen the mesh with the highest element distinction metric (too much details regarding the resolution)
- Refine the mesh with the minimal element distinction metric with the constraints to not exceed:
  - The maximum total memory used,
  - The maximum total downloading time,
  - And the maximum element distinction metric value.

Integration to X3D

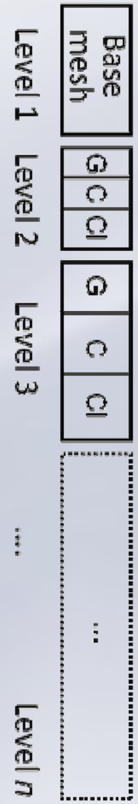
A node <ProgrTriSet> defines a progressive 3D model

Nodes <ProgrLOD> provide the URLs of every LOD

The compression has been already done offline

The client receive the X3D file then he choses the level to download according to our adaptation algorithm

```
<ProgrTriSet>
        <ProgrLOD level=”1” url=”http://collaviz.org/lod1.ps” />
        <ProgrLOD level=”2” url=”http://collaviz.org/lod2.ps” />
        <ProgrLOD level=”3” url=”http://collaviz.org/lod3.ps” />
        <ProgrLOD level=”4” url=”http://collaviz.org/lod4.ps” />
        ...
</ ProgrTriSet>
```



Similar to the VRML extension from Fogel et al., WEB3D 2001

Demonstration using a Real case

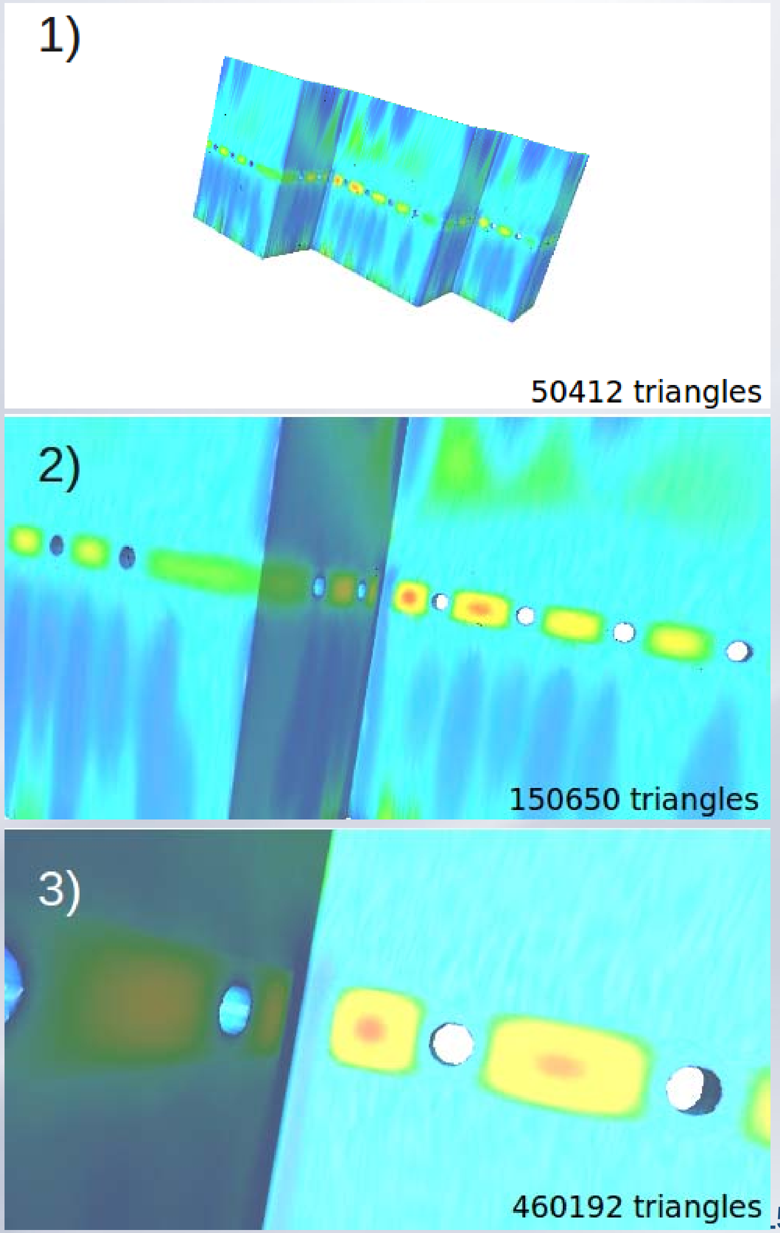
Thermal analysis of bolts of a french pwr core internal baffle structure
1.1 billions elements, 40GB of 3D data
Need a 96GB RAM workstation

Client on a station with 2,8 Ghz proc. and a NVIDIA Quadro FX 580.

Visualization of a 460Kfacets data.

The client starts with a global view point and then zooms toward the model

1) Level of detail 29/34 (27% of the data) rendering time: 45ms
2) Level of detail 32/34 (52% of the data) Rendering time: 230ms
3) Level of detail 34/34 (100% of the data) Rendering time: 266ms



Conclusion

Adaptive and progressive streaming framework dedicated to remote scientific visualization

- ❑ Progressive compression for 3D meshes with color
- ❑ Adaptation mechanisms
- ❑ X3D integration

What next?

- ❑ Random access decompression
- ❑ Other associated data

This work has been founded by French National Research Agency (ANR) through COSINUS program (project COLLAVIZ)

The 3D models presented in these slides are courtesy of EDF

THANK YOU!