Consolidation de nuages de points

Stefka GUEORGUIEVA
stefka.gueorguieva@labri.fr

Plan

Surfaces à base de points

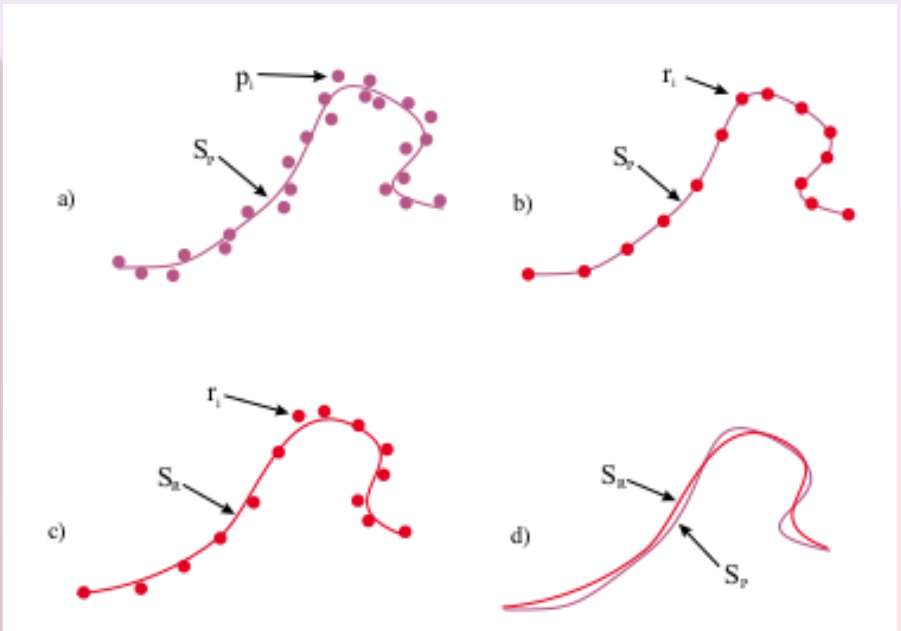
M.Alexa et al.

"Computing and rendering point set surfaces", IEEE TVCG 9(1) Jan 2003,fig 1

Représentation à base de points

Soit $P = p_i$,$0 \leq i \leq n$, un ensemble de points et S_P, une surface *Moving Least Square* sur P

Idée : Remplacer P par $R = r_i$,$0 \leq i \leq m$, $m < n$, et construire S_R, une surface *MLS* sur R, pour approximer S_P



Surfaces à base de points

M.Alexa et al.

"Computing and rendering point set surfaces", IEEE TVCG 9(1) Jan 2003, fig 2

Surfaces à base de points

M.Alexa et al., 2003

La consolidation d'un nuage de points regroupe des traitements géométriques pour lisser les points sur une surface *MLS*

- Filtrage et nettoyage
- Recallage
- Rééchantillonnage
- Classification et segmentation
- Reconstruction 3D

Nuages de points acquis

Photogrammétrie

Reconstruction à partir de plusieurs prises de vues photographiques, Photographie 3D, SIGGRAPH'2000, figure de Kutulakos et al.,"A theory of shape by space carving", 2000.

Nuages de points acquis

Scanner laser 3D

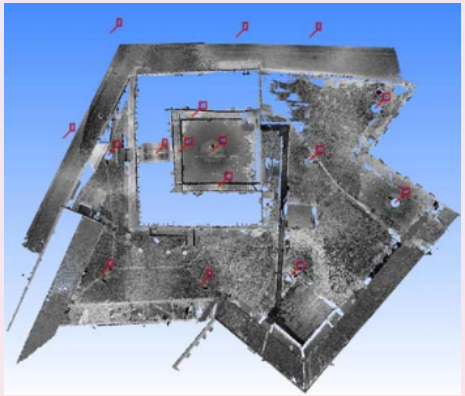
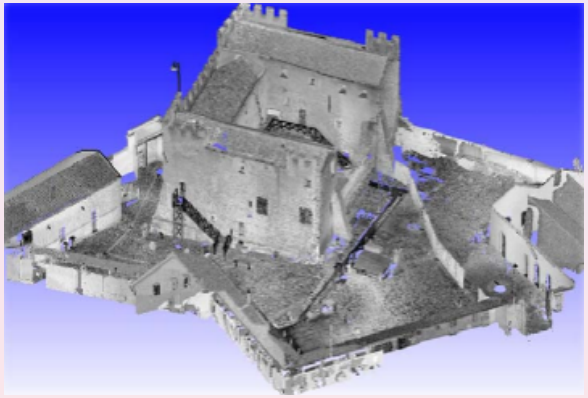
- Traitements : acquisition, recalage, intégration, décimation, reconstruction, remaillage (simplification, subdivision, lissage, LOD, ...)
- Bernardini et al.,"The 3D model acquisition pipeline", Comput. Gr. Forum, 2002

Fig. 1. (left) A photograph of Michelangelo's Florentine Pietà. (right) A synthetic picture from our three-dimensional computer model.

Nuages de points acquis

Scanner laser 3D

"Laser Scanner Survey to cultural heritage conservation and restoration",G.Vacca et al, XXII ISPRS Congress, 2012

Classification

M.Pauly et al.

"Efficient simplification of point-sampled surfaces",
IEEE Visualisation. 2002, 163-170

Regroupement par croissance de régions

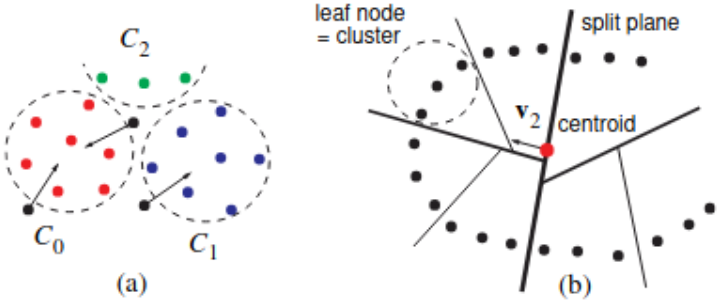


Figure 4: *(a) Clustering by incremental region growing, where "stray samples" (black dots) are attached to the cluster with closest centroid. (b) Hierarchical clustering, where the thickness of the lines indicates the level of the BSP tree (2D for illustration).*

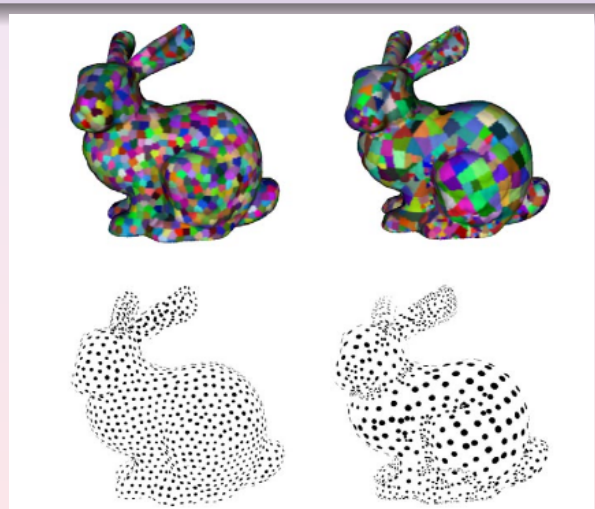


Figure 5: *Clustering: Uniform incremental clustering is shown in the left column, adaptive hierarchical clustering in the right column. The top row illustrates the clusters on the original point set, the bottom row the resulting simplified point set (1000 points), where the size of the splats corresponds to cluster size.*

Classification

M.Pauly et al., 2002

Simplification itérative :
suppression de points,
contraction de paires de points,
k-plus proches voisins, ...

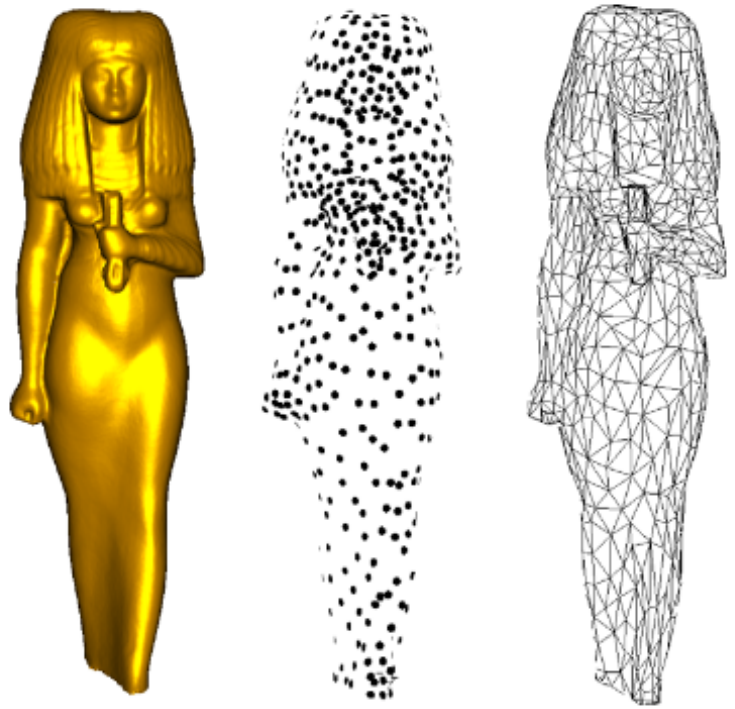


Figure 6: *Iterative simplification of the Isis model from 187,664 (left) to 1,000 sample points (middle). The right image shows all remaining potential point-pair contractions indicated as an edge between two points. Note that these edges do not necessarily form a consistent triangulation of the surface.*

Classification

M.Pauly et al., 2002

Simulation de particules

Figure 7: *Simplification by adaptive particle simulation. The left image shows the original model consisting of 75,781 sample points. On the right, a simplified version consisting of 6,000 points is shown, where the size of the splats is proportional to the repelling force of the corresponding particle.*

Simplification basée à la préservation des caractéristiques de forme

K.Zhang et al.

"Feature-preserved point cloud simplification based on Natural Quadric Shape Models", Appli.Sci. 2019.9,2130, fig.12

(a) Uniforme
(b) Adaptative
(c) Caractéristique de forme : la courbure
(d) FPPS, caractéristiques de forme (plan, sphère, ellipsoid, cylindre et cone)

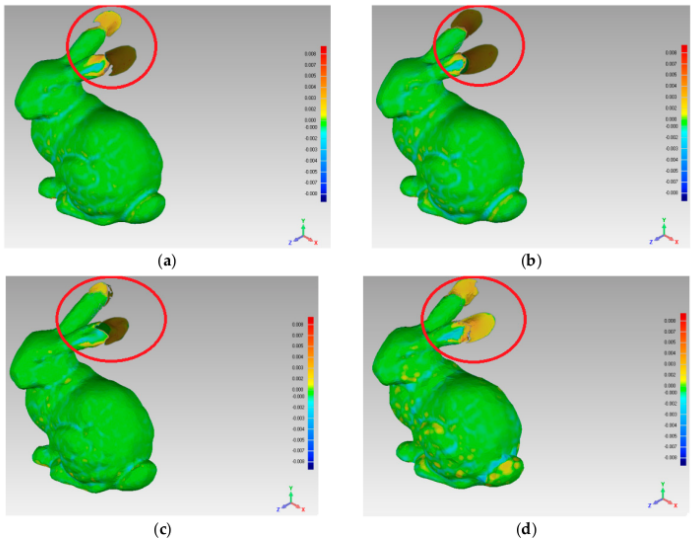
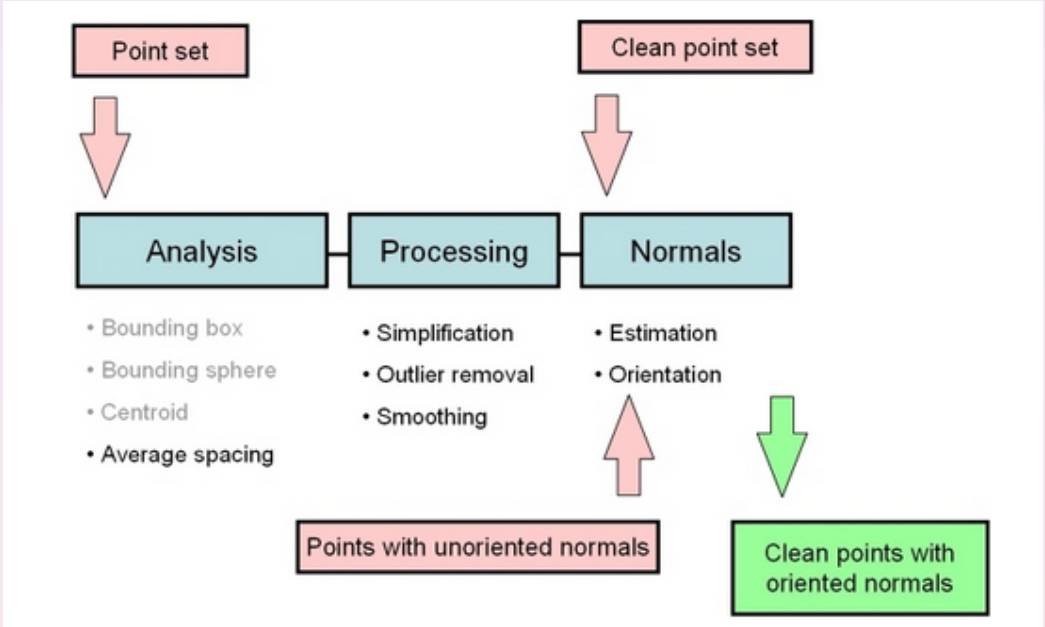


Figure 12. The simplification effect comparison using the bunny data set: **(a)** uniform-based simplification algorithm; **(b)** grid-based simplification algorithm; **(c)** curvature-based simplification algorithm; **(d)** FPPS.

CGAL Point Set Processing

CGAL 6.1 - Point Set Processing

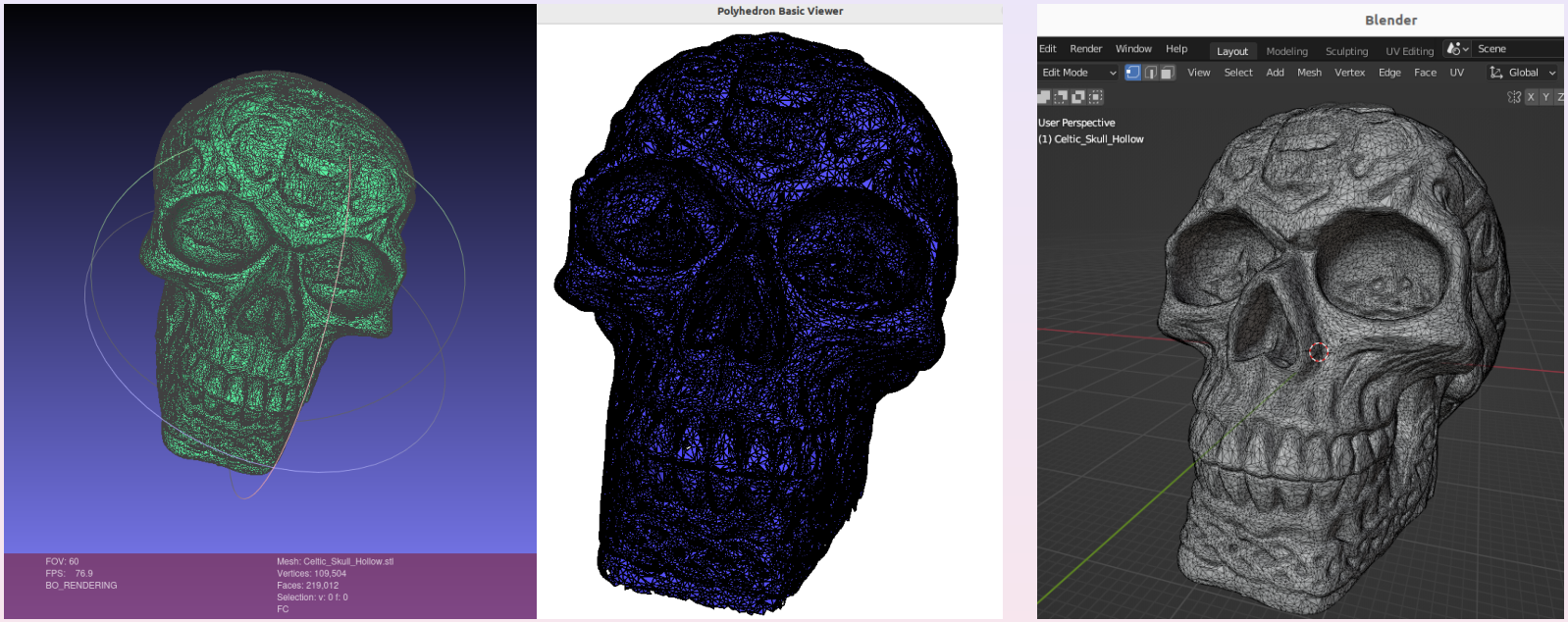


CGAL Point Set Processing

CGAL 6.1 - Point Set Processing

1. *random_simplify_point_set()*
2. *grid_simplify_point_set()*
3. *hierarchy_simplify_point_set()*
4. *wlop_simplify_and_regularize_point_set()*
5. *poisson_eliminate()*

Thingi10K



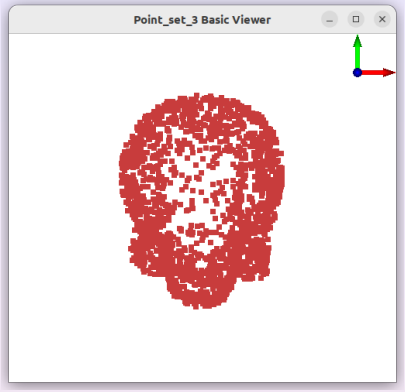
(a) Meshlab (b) CGAL Polyhedron (c) Blender

Figure – Thingi10K *Celtic_Skull_Hollow* (STL)

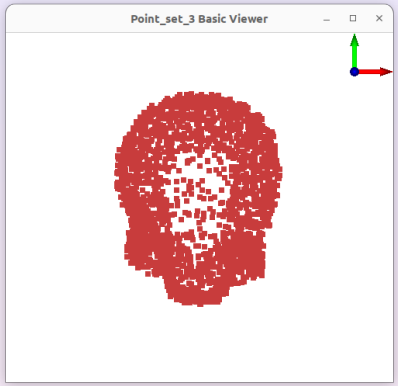
Simplification Aléatoire

```
#include <CGAL/Exact_predicates_inexact_constructions_kernel.h>
#include <CGAL/random_simplify_point_set.h>
#include <CGAL/IO/read_points.h>
#include <CGAL/IO/write_points.h>
#include <vector>
#include <fstream>
typedef CGAL::Exact_predicates_inexact_constructions_kernel Kernel;
typedef Kernel::Point_3 Point;
int main(int argc, char*argv[]){
  const std::string fname =
    (argc>1) ? argv[1] : CGAL::data_file_path("data/oni.xyz");
  std::vector<Point> points;
  if(!CGAL::IO::read_points(fname, std::back_inserter(points))){
    std::cerr << "Error: cannot read file " << fname << std::endl;
    return EXIT_FAILURE;}
  unsigned int removed_percentage = 98;
  auto iterator_to_first_to_remove
    = CGAL::random_simplify_point_set(points, removed_percentage);
  points.erase(iterator_to_first_to_remove, points.end());
  points.shrink_to_fit();
  if(!CGAL::IO::write_points("out.xyz",points,
                             CGAL::parameters::stream_precision(17))){
      std::cerr << "Error: cannot write file " << std::endl;
      return EXIT_FAILURE;}
  return EXIT_SUCCESS;
}
```

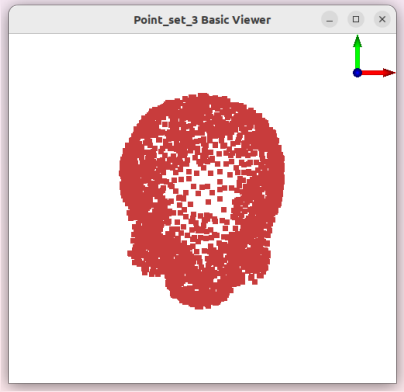
CGAL Point Set Processing



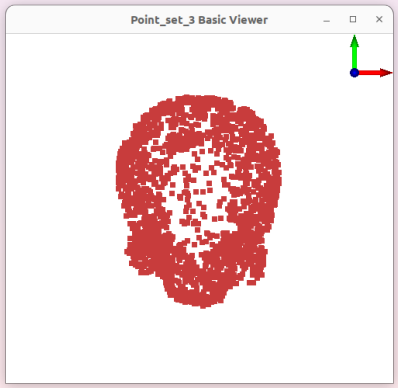
(a) aléatoire



(b) grille



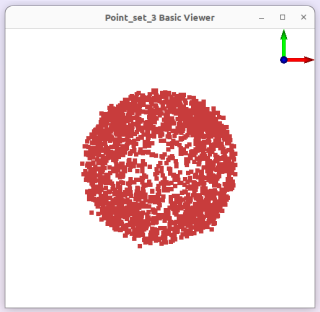
(c) hiérarchie



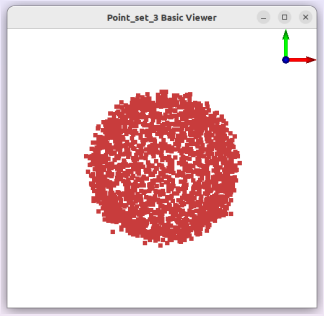
(d) wlop

Figure – Simplification de *Thingi10K, Celtic_Skull_Hollow.xyz*

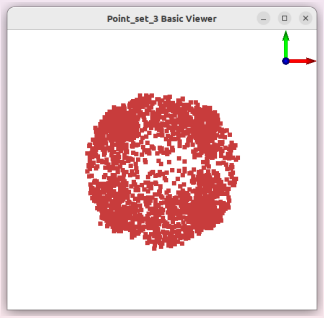
CGAL Point Set Processing



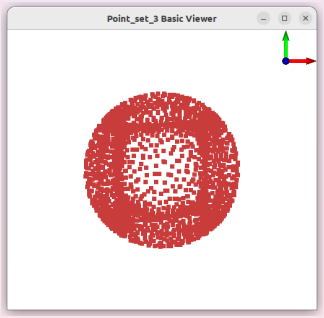
(a) aléatoire, fraction de points à supprimer= 90



(b) grille, taille d'un voxel= 0.03, min pnts par voxel= 3



(c) hiérarchie, taille max d'un cluster= 100, variation= 0.12



(d) wlop, points préservés= 10%, rayon de voisinage= 0.5

Figure – Simplification de *sphere_20k.xyz*