

---

# Fondamenti di Grafica Tridimensionale

Paolo Cignoni

[p.cignoni@isti.cnr.it](mailto:p.cignoni@isti.cnr.it)

<http://vcg.isti.cnr.it/~cignoni>

# Obiettivi

---

- ❖ Conoscere gli aspetti fondamentali riguardo alle nozioni teoriche e algoritmiche per la modellazione geometrica e il rendering di scene tridimensionali
  - ❖ Strutture e algoritmi per la rappresentazione di geometrie
  - ❖ Architetture per la grafica interattiva
  - ❖ Algoritmi per il rendering
  - ❖ Modellazione
  - ❖ Modelli avanzati di illuminazione
- ❖ Requisiti
  - ❖ Conoscenza delle basi della grafica 3D
  - ❖ Conoscenza del C++

# Corsi collegati

---

- ❖ Costruzione di interfacce
  - ❖ Fortemente (molto fortemente) consigliato che lo abbiate già fatto prima di seguire. Durante il corso assumo che sappiate bene:
    - ❖ C++; un minimo di grafica 3D
    - ❖ Sappiate costruire un app. interattiva con una gui non banale.
- ❖ Laboratorio di computer games (LCG)
  - ❖ E' sensato aver fatto prima CI e FGT prima di LCG
- ❖ Matematica computazionale: Geometria computazionale (MCG)
  - ❖ Per approfondire gli aspetti piu' teorici

# Strumenti

---

- ❖ Un po' di analisi, algebra lineare e un po' di geometria computazionale
- ❖ Linguaggio C++
- ❖ Libs, Toolkits, ecc
  - ❖ Tutto quel che ci serve in modo utilitaristico
    - ❖ Qt per le interfacce
    - ❖ SDL per i primi esperimenti in opengl
    - ❖ STL per non riscrivere i soliti contenitori

# Esame

---

- ❖ Basato su progetto
  - ❖ Piu' algoritmico di quello di CI
  - ❖ dettagli in seguito
- ❖ Compitino/orale
  - ❖ più influenti sul voto rispetto a CI

# MeshLab

---

- ❖ Generic mesh processing tool
- ❖ Open source GPL avail for win and linux
  - ❖ The system relies on a large gpl library for mesh processing
- ❖ Aimed to the (almost) automatic processing of large unstructured 3d models
  - ❖ acquired 3D models are different from human built models
    - ❖ No structure, no particular need of a scene graph
    - ❖ Large datasets : millions of primitives

# MeshLab tasks

## ❖ Filtering

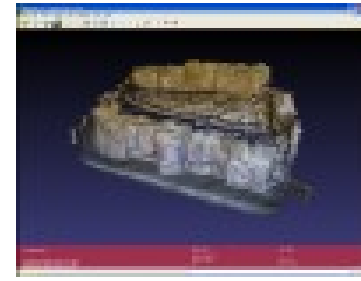
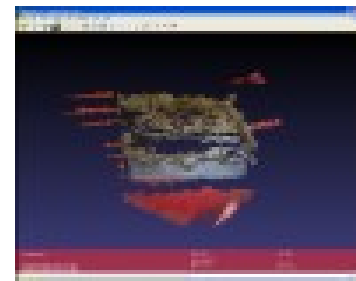
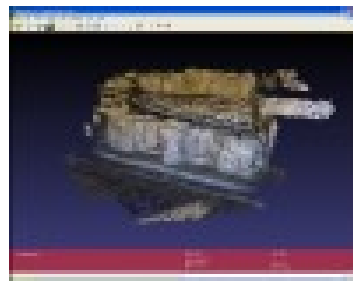
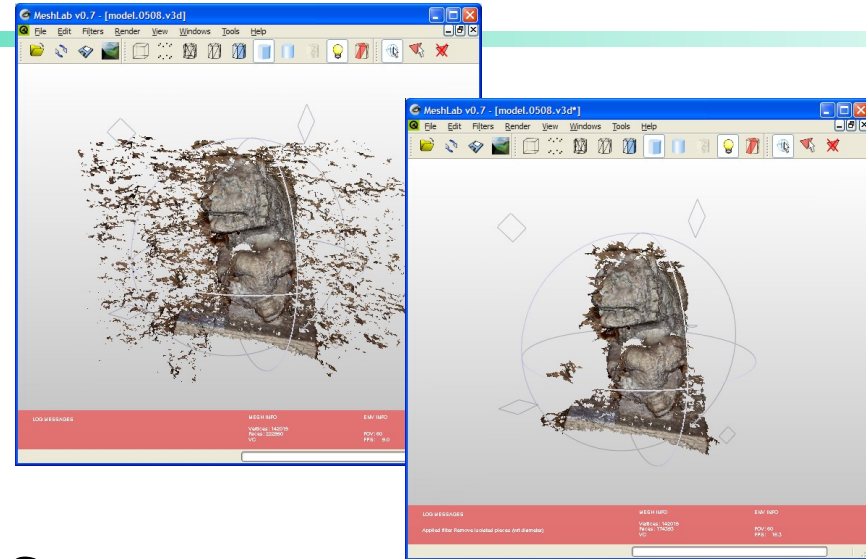
- ❖ Removal of outliers according to various heuristics

## ❖ Smoothing out noise

- ❖ Various fairing algorithms

## ❖ Editing

- ❖ Selection of parts and removal of non interesting portions of the acquired data



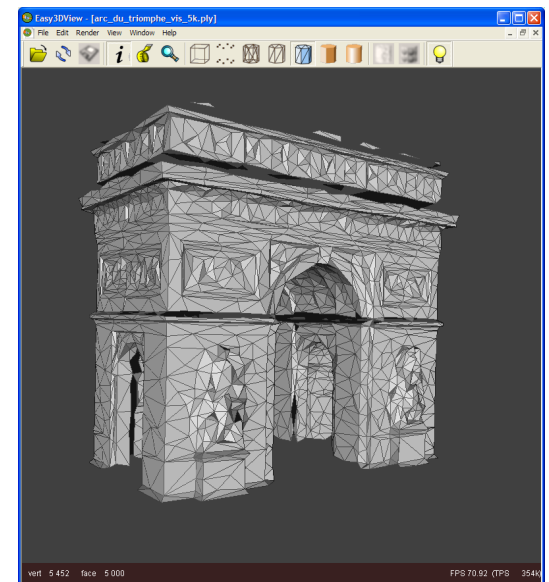
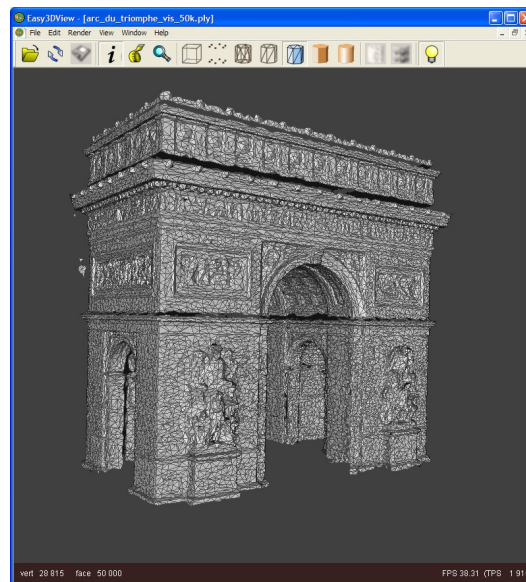
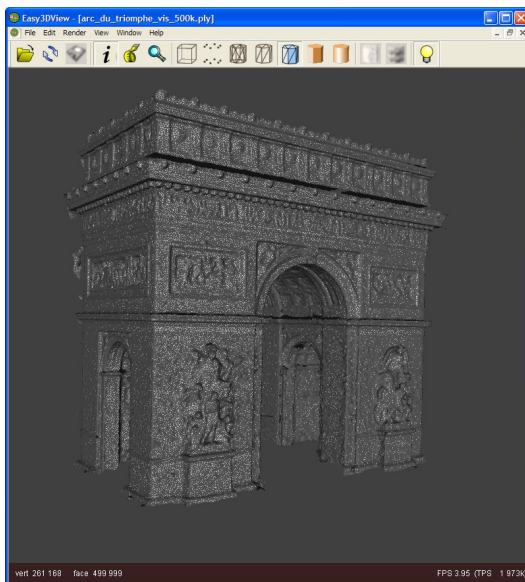
# Remeshing

- ❖ Subdivision Surface (loop and butterfly)
- ❖ Reducing complexity through simplification
  - ❖ Both fast approximate and slower high quality algorithms

❖ 500k

50k

5k





# Checking

- ❖ Visual inspection
  - ❖ See through filters (opengl shaders)
- ❖ Quality evaluation
  - ❖ Curvature dependent
- ❖ Automatic marking of mesh inconsistencies
  - ❖ topological: non manifoldness)
  - ❖ Geometric: self intersection
  - ❖

