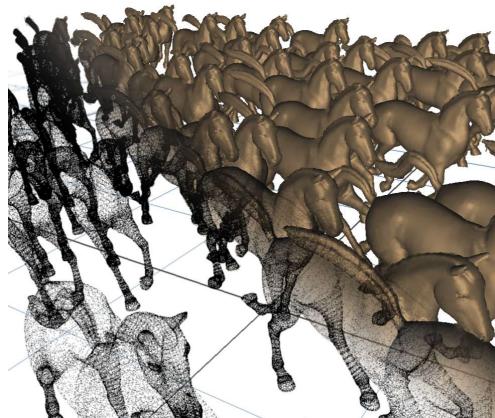


Point cloud: rendering

- ✓ Attraverso «point splats»:
- ✓ «splat» = una regione di schermo che viene disegnata per rappresentare il punto



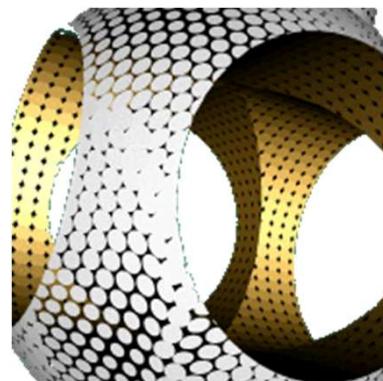
img by Michael Wimmer 2012



16

Point cloud: rendering

- ✓ dimensione dello splat dipende dalla densità dei punti, la distanza, etc
- ✓ disegnabili anche come pezzetti di superficie



img by Mario Botsch, 2004

possible
grazie
alle normali!



17

Point cloud: file formats

- ✓ <filename>.xyz
- ✓ davvero molto semplice...

```
-93.588310 384.453247 8.672122 -0.750216 0.605616 0.265339
-93.365311 384.456879 9.228838 -0.766408 0.558148 0.317947
-92.981407 384.595978 9.903000 -0.771328 0.544261 0.329898
-93.032166 384.662994 9.664095 -0.753935 0.579665 0.309145
-92.670418 384.707367 10.423333 -0.773636 0.510460 0.375390
-92.432343 384.866455 10.697584 -0.775882 0.504696 0.378535
-92.233391 384.935974 11.023732 -0.770233 0.538780 0.341258
-91.959816 384.952209 11.629505 -0.765533 0.534614 0.357975
-91.684883 385.122528 11.965501 -0.783344 0.499727 0.369656
-90.981750 385.299408 13.283755 -0.797000 0.474158 0.374119
-91.017151 385.362061 13.132144 -0.801652 0.457642 0.384600
-90.481400 385.523560 13.933455 -0.691992 0.589880 0.416160
-90.241745 385.597351 14.216219 -0.664425 0.625032 0.409725
-89.772568 385.683197 14.820392 -0.626528 0.653291 0.425056
-89.167023 385.833191 15.455617 -0.654831 0.640269 0.401562
-88.530830 386.009369 16.353712 -0.737686 0.498695 0.455108
-87.759621 386.192017 17.259428 -0.681486 0.572231 0.456211
-86.996521 386.357880 18.085392 -0.653592 0.574349 0.492890
-86.823006 386.214783 18.469635 -0.665109 0.533859 0.522135
-85.953079 386.573792 19.228466 -0.682286 0.524974 0.508810
...eccetera
```



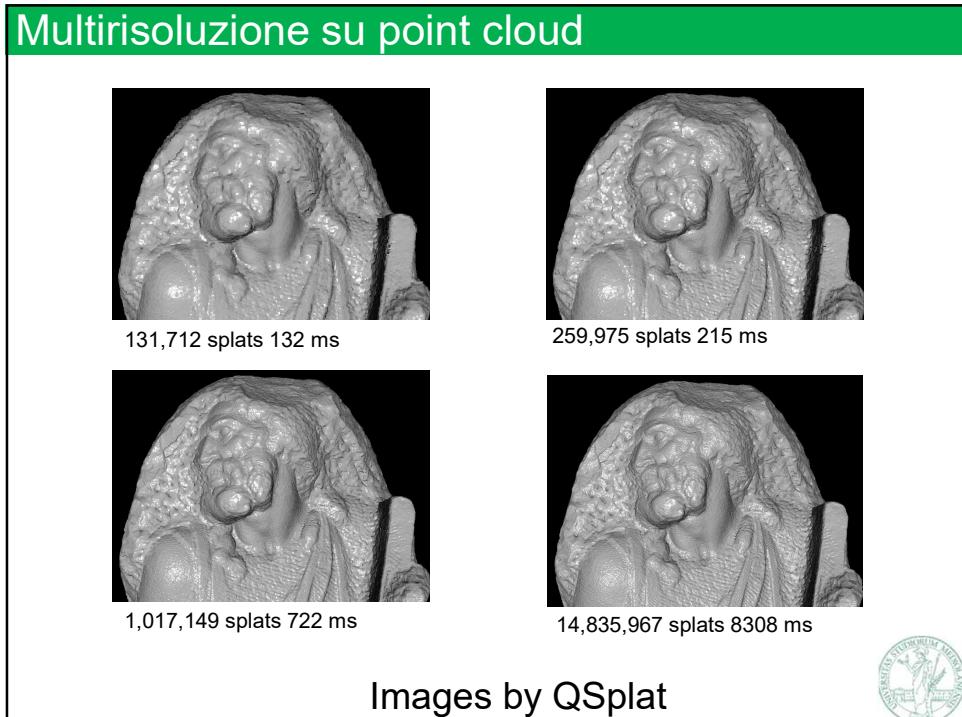
18

Multirisoluzione su point cloud

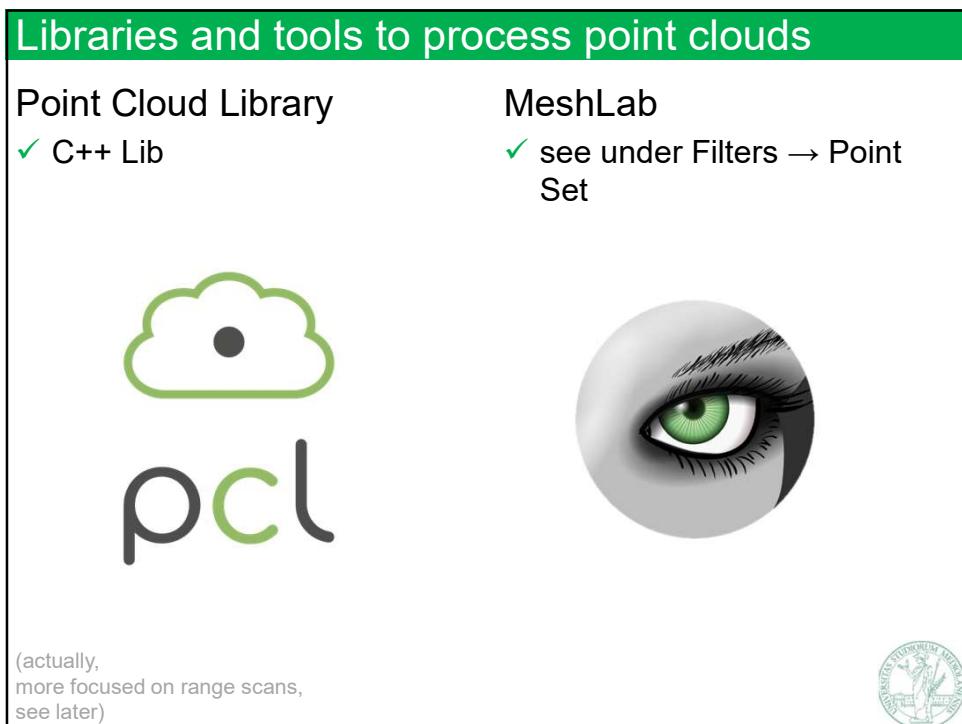
- ✓ La point cloud è semplice da gestire perché ogni punto è indipendente dagli altri
- ✓ Ad esempio,
una semplice **multirisoluzione continua**
si ottiene semplicemente prenderendo un
sottoinsieme
- ✓ Strategia: (adottata ad es da «qSplat» - Stanford uni)
 - ⇒ preordinare la point cloud di N vertici
in modo che i primi M<N vertici siano ben distribuiti
 - ⇒ (mischiarli a caso = buona approssimazione!)
 - ⇒ in fase di rendering disegnare solo i primi M vertici
 - ⇒ fast preview



19



22



23

Geometry Processing on point clouds

Examples of typical tasks: **creation!**

- ✓ Extract point cloud from images
 - ⇒ Shape From Motion, studied in Computer Vision
 - ⇒ see lecture on 3D acquisition
- ✓ Define normals (whenever they are missing)

The availability of easy method to create point clouds (from standard images) is the element making them increasingly popular.



24

Geometry Processing on point clouds

Let's try a simple task first:
offsetting a surface with a point cloud.

- ✓ Surface offsetting:
add a small «offset» to the surface,
e.g. simulating as if a small layer of paint over the surface
- ✓ How to: displace every point
in the cloud P by a small offset k in the normal direction

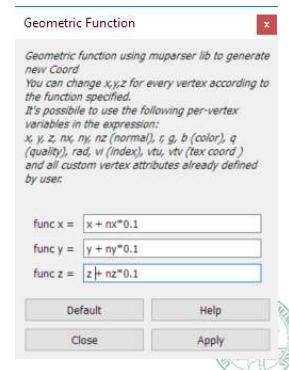
$$\forall \mathbf{p}_i \in P : \mathbf{p}_i \leftarrow \mathbf{p}_i + k \cdot \hat{\mathbf{n}}$$



25

Esperimento in aula: surface offsetting

- ✓ Scaricare una point cloud dal sito del corso
 - ⇒(da ariel)
- ✓ Scaricare, installare ed eseguire MeshLab (sito del corso per il link)
 - ⇒ File => import mesh...
 - => selezionare la pointcloud
 - ⇒ Filters => Smoothing, fairing and defor.
 - => Geometry Function
 - ⇒ Imettere i valori per applicare la formula
 - il punto p_i ha coordinate (x,y,z)
 - la normale \hat{n} ha coordinate (nx,ny,nz)
 - come offset usiamo, in questo esempio, 0.1



26

Geometry Processing on point clouds

Examples of typical tasks: processing

- ✓ Common subtask:
 - find spatial neighbors of a point in the cloud
 - ⇒ es, find its **k nearest neighbors** (k-NN)
 - ⇒ remember: each point is independent from other
 - ⇒ can be expensive (linear with resolution!)
 - ⇒ see: spatial indexing structures, later
- ✓ Cleaning-up
 - ⇒ Remove «outliers»
 - ⇒ Reduce noise
- ✓ «registration»: Alignment to another point could



27

Geometry Processing on point clouds

Examples of typical tasks: processing

- ✓ Most common sub-task:
find spatial neighborhood of a point in the cloud
 - ⇒ es, find its ***k* nearest neighbors** (k-NN)
 - ⇒ can be expensive (linear with resolution!)
 - ⇒ remember: each point independent from all others
 - ⇒ see: spatial indexing structures, later

- ✓ A preliminary needed by any other task!

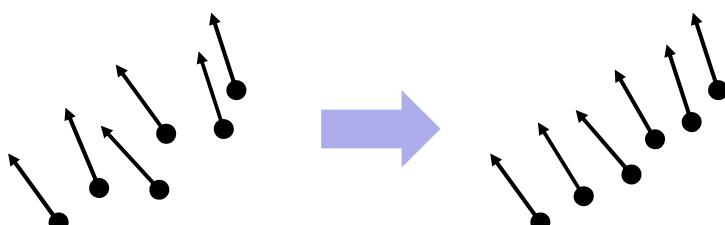


28

Geometry Processing on point clouds

Examples of typical tasks: processing

- ✓ Cleaning-up: noise reduction
 - ⇒ on position and/or normals

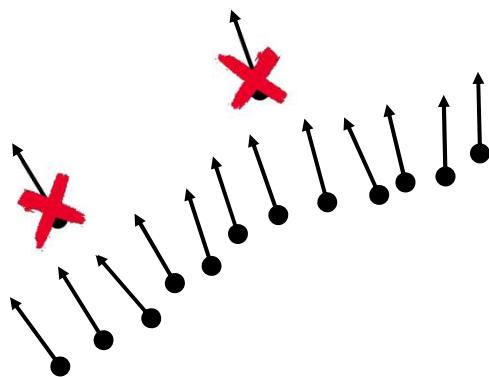


29

Geometry Processing on point clouds

Examples of typical tasks: processing

- ✓ Cleaning-up: remove «outliers»

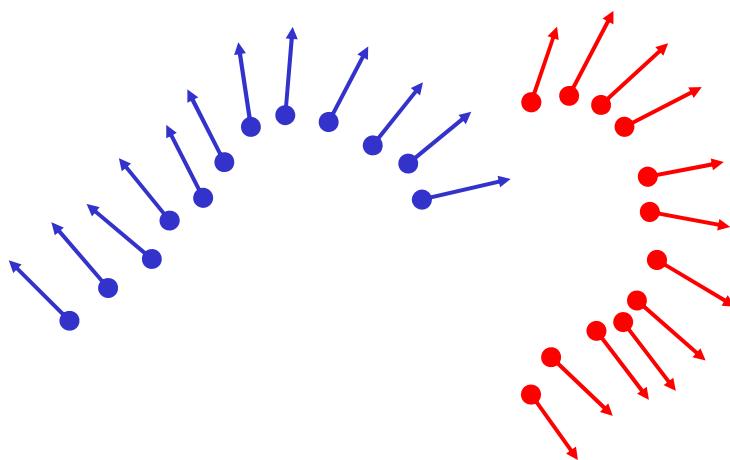


30

Geometry Processing on point clouds

Examples of typical tasks: processing

- ✓ «Registration»: alignment to another point could

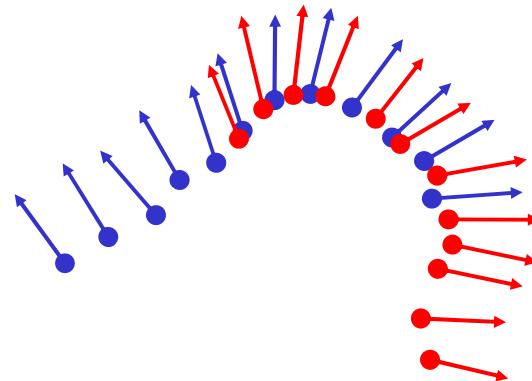


31

Geometry Processing on point clouds

Examples of typical tasks: processing

- ✓ «Registration»: alignment to another point could

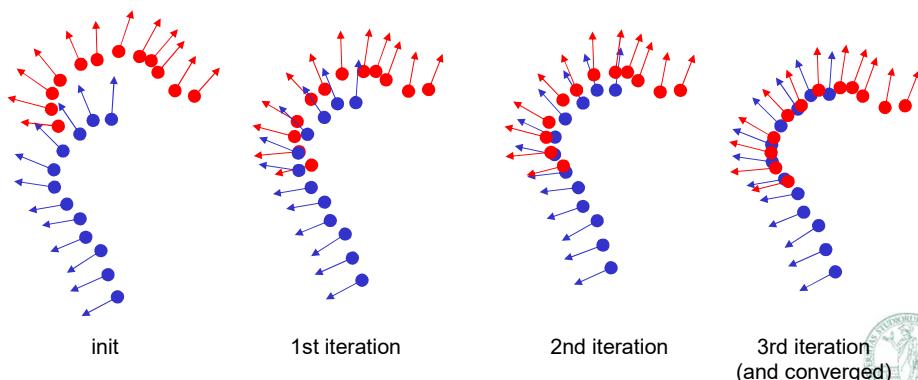


34

Geometry Processing on point clouds

Registering one point cloud over another

- ✓ Step 1: Coarse Registration: difficult (e.g. manual)
- ✓ Step 2: Fine registration: automatic.
ICP («Iterative Closest Points») algorithm



35

Esperimento in aula: allineare due point cloud

Come ricrearlo:

1. Scaricare le due point cloud in formato xyz
(dal sito ariel)
2. Scaricare meshlab (se non lo si è ancora fatto)
3. Importare entrambe le point cloud
4. Aiutarsi con questo tutorial:
<https://www.youtube.com/watch?v=4g9Hap4rX0k>



36

Geometry Processing on point clouds

Examples of typical tasks: processing

- ✓ Most typical task: convert to ... a mesh
- ✓ see next lecture!



37