www.crs4.it/vic/

vcg.isti.cnr.it/

Fast and robust techniques for 3D/2D registration and photo blending on massive point clouds

R. Pintus, E. Gobbetti, M.Agus, R. Combet
CRS4 Visual Computing

M. Callieri
Visual Computing Group ISTI-CNR

Goal

- **Fast and robust techniques for creating accurate colored models**
- **Acquisition**
  - 3D – laser scanners
  - Color – digital cameras
- **Point-based rendering pipeline**
- **Mapping photo-to-geometry**
  - Fast and Robust Semi-Automatic Registration of Photographs to 3D Geometry
- **Photo blending**
  - A Streaming Framework for Seamless Detailed Photo Blending on Massive Point Clouds

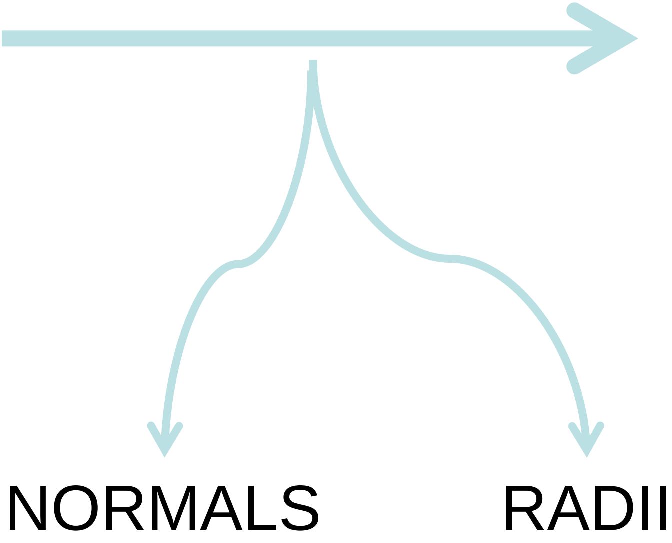
www.crs4.it/vic/

vcg.isti.cnr.it/

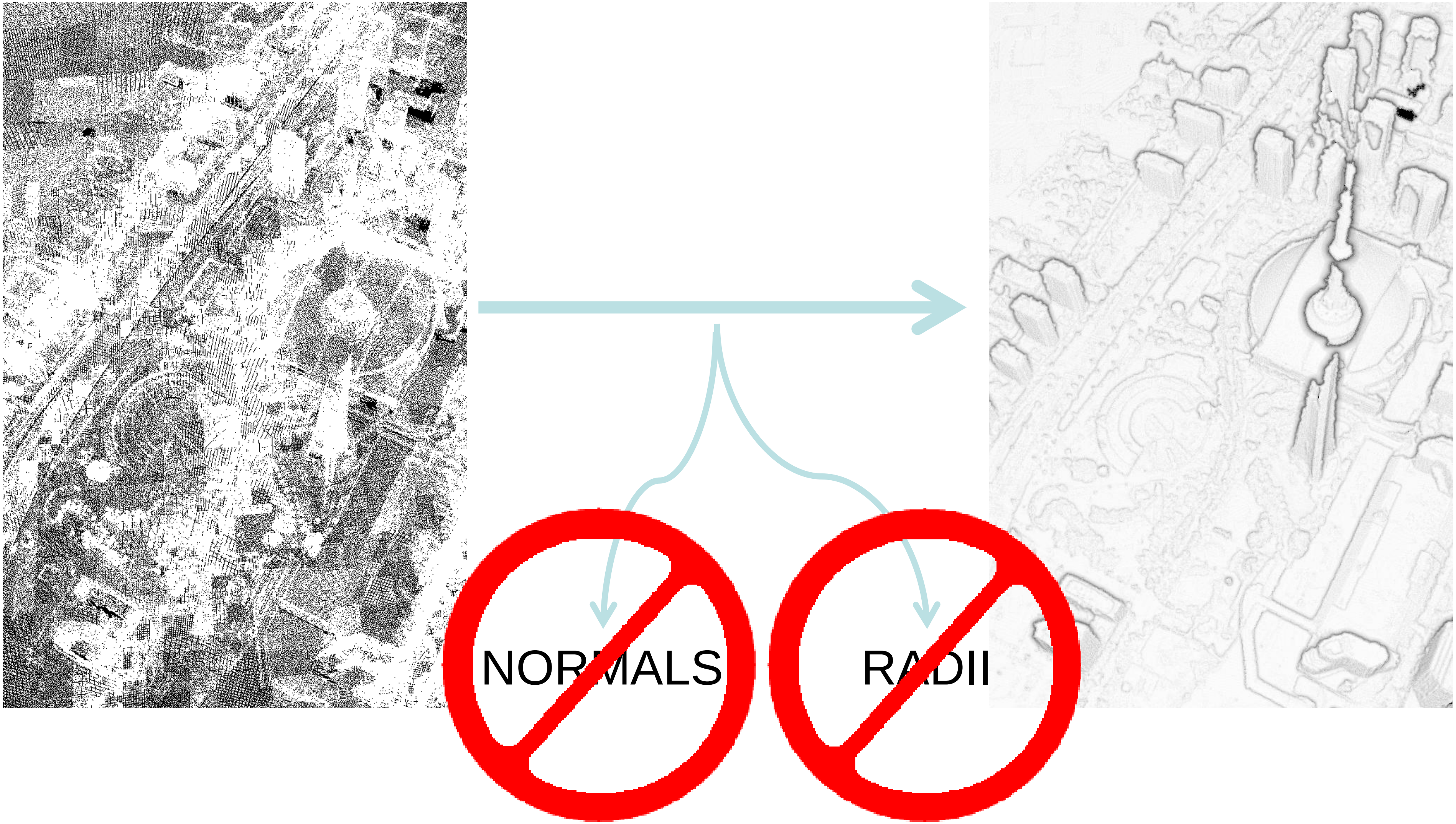
Point-based Rendering

Ruggero Pintus, Enrico Gobbetti, and Marco Agus. "Real-time Rendering of Massive Unstructured Raw Point Clouds using Screen-space Operators". In The 12th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, October 2011.

Problem Statement



Problem Statement



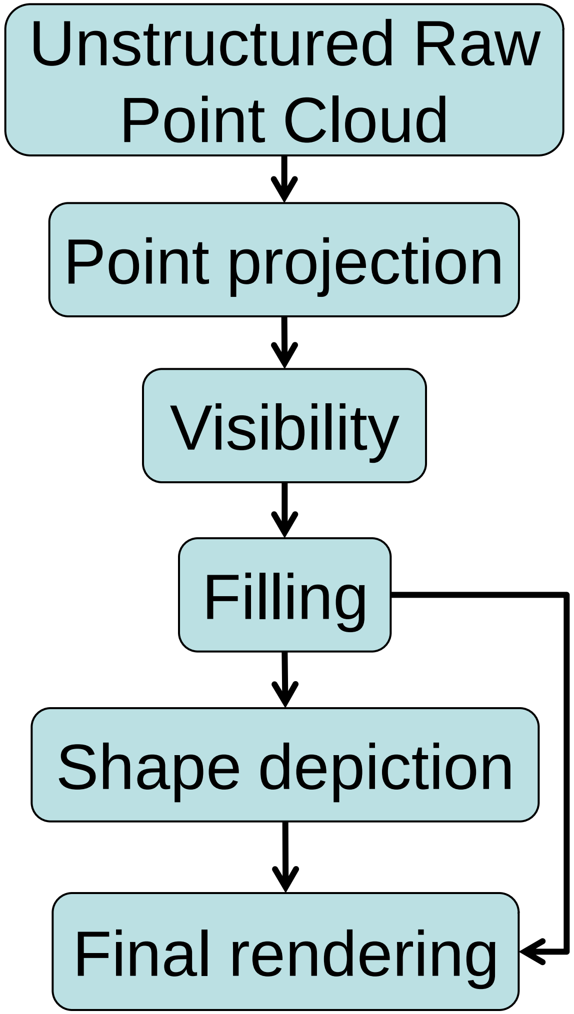
Related work

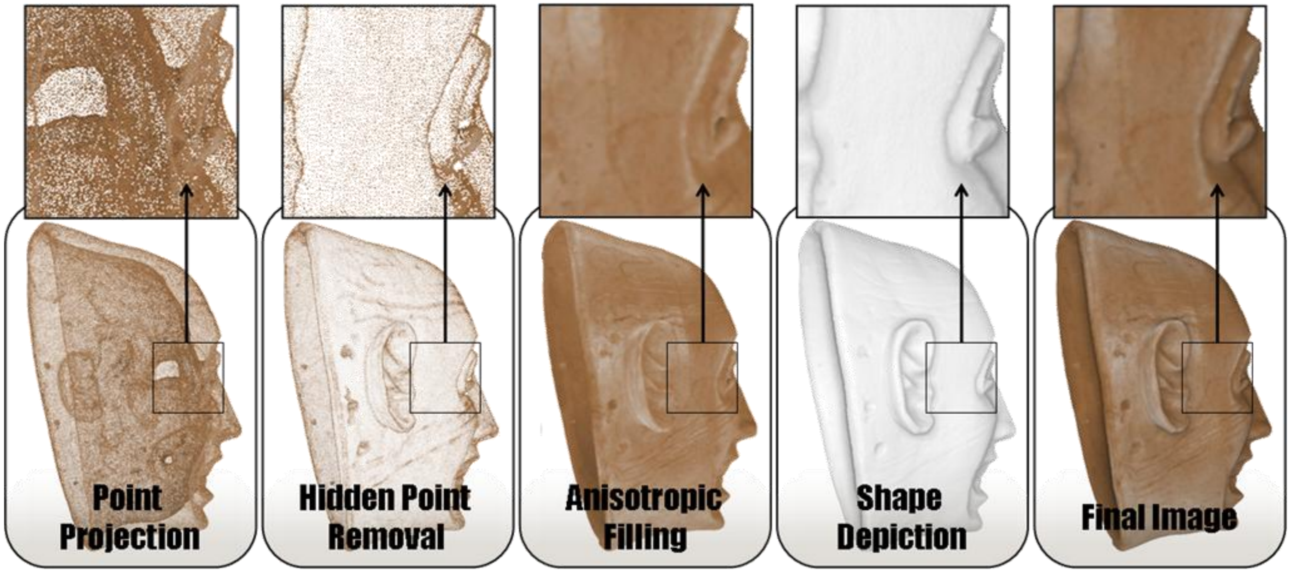
- **Point based rendering (PBR) techniques**
- **Continuous surface, solving visibility**
- **Object space vs Screen space**
  - Point attributes (normals and radii) needed
- **Require only radii – [Grottel et al. 2010]**
- **No attributes – [Wimmer et al. 2006]**
  - Visibility not solved
- **Direct visibility algorithms – [Katz et al. 2007] and [Mehra et al. 2010]**
  - Compute point set convex hull – not feasible for a real-time application

Our method

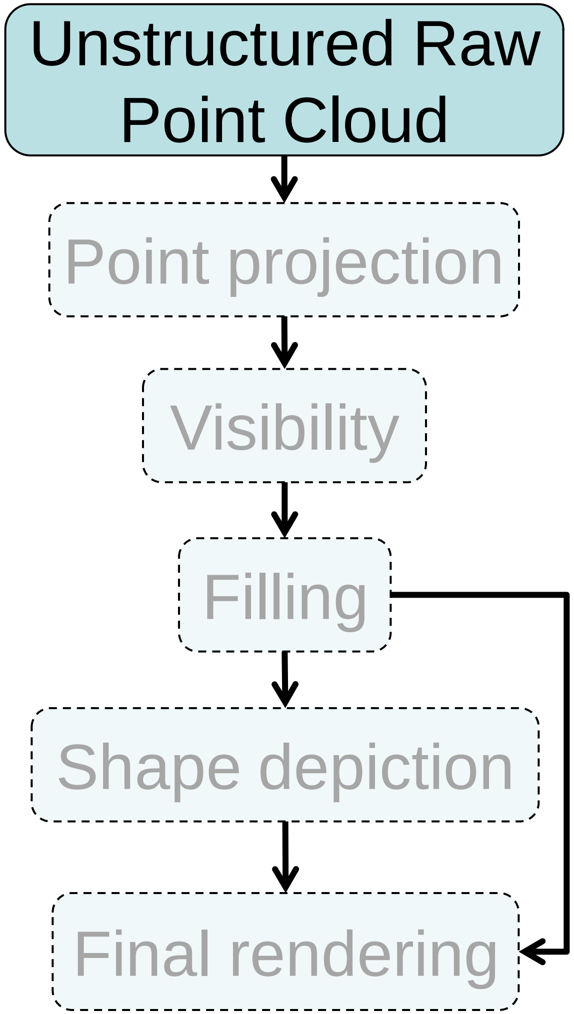
- **Point based rendering pipeline**
- **Unstructured raw point clouds (no attributes)**
- **Screen-space operators**
  - Direct visibility, Anisotropic filling, Shading
- **Real-time with massive models**
  - Multi-resolution out-of-core structure
- **Minimizing pre-computation**
- **Suitable for quick on-site visualizations**

Pipeline - Overview



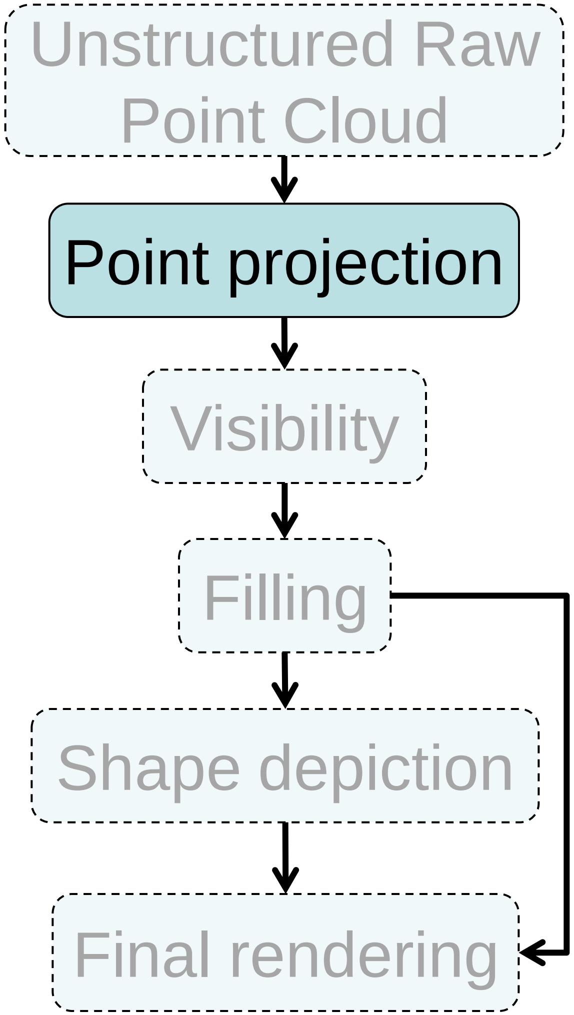


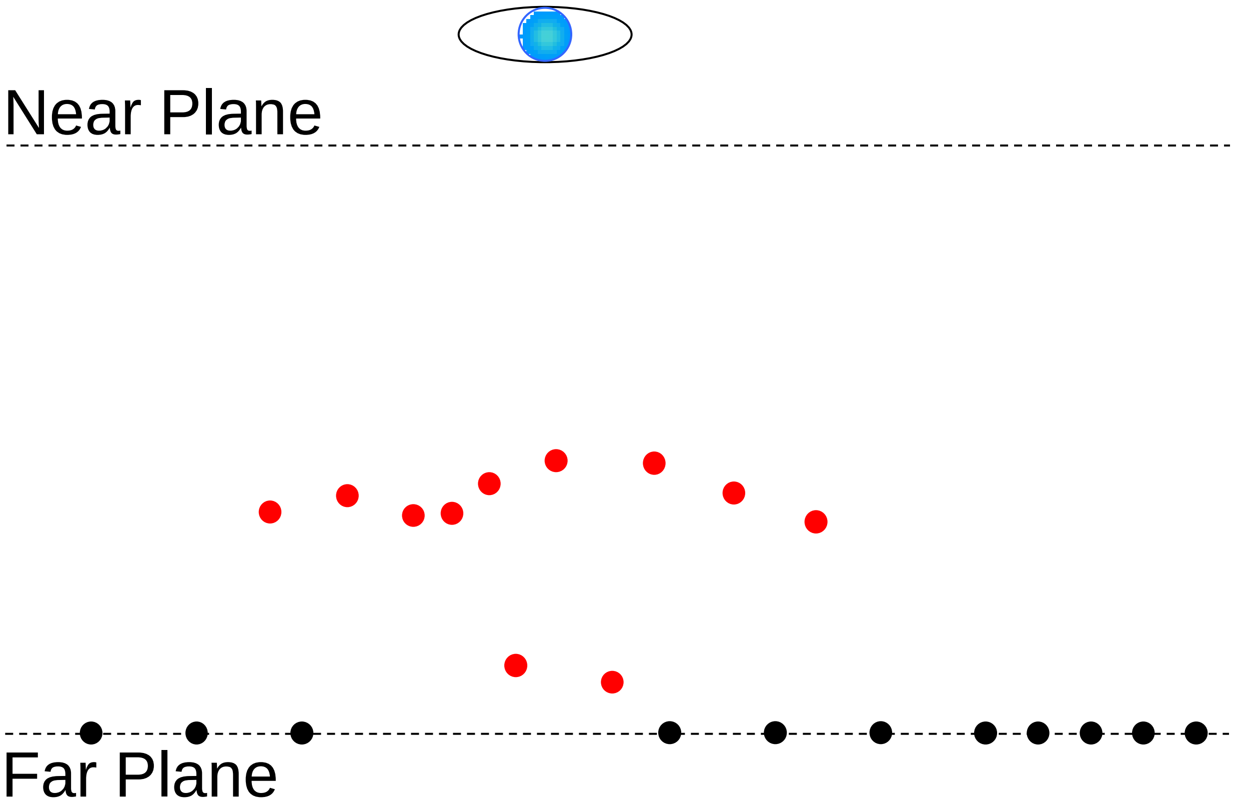
Input data



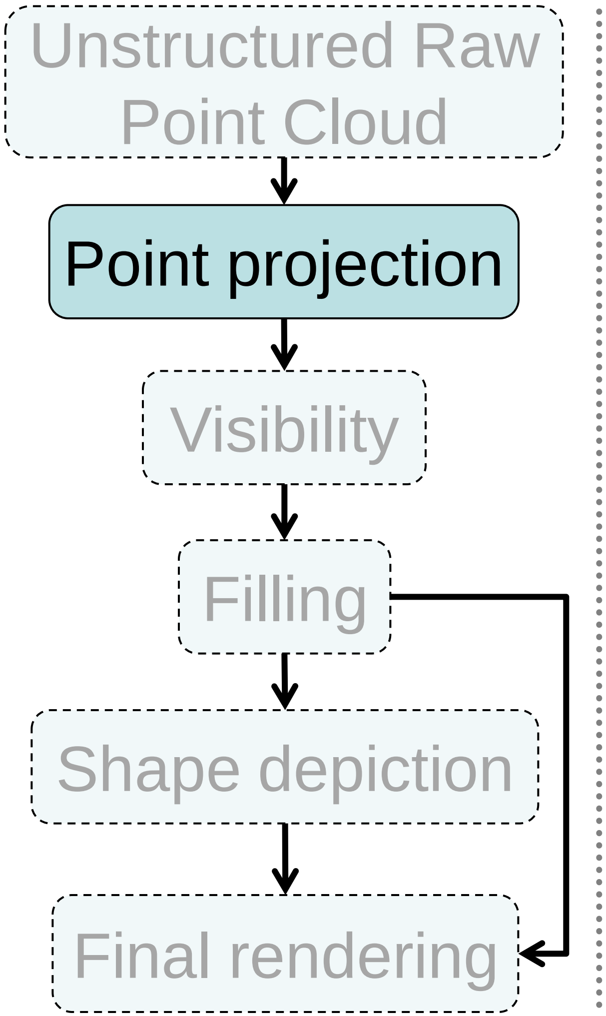
- **Unstructured raw point cloud**
  - Position
  - (Color?)
  - No normals
  - No influence radii
- **Multi-resolution out-of-core structure**
  - Fast pre-processing

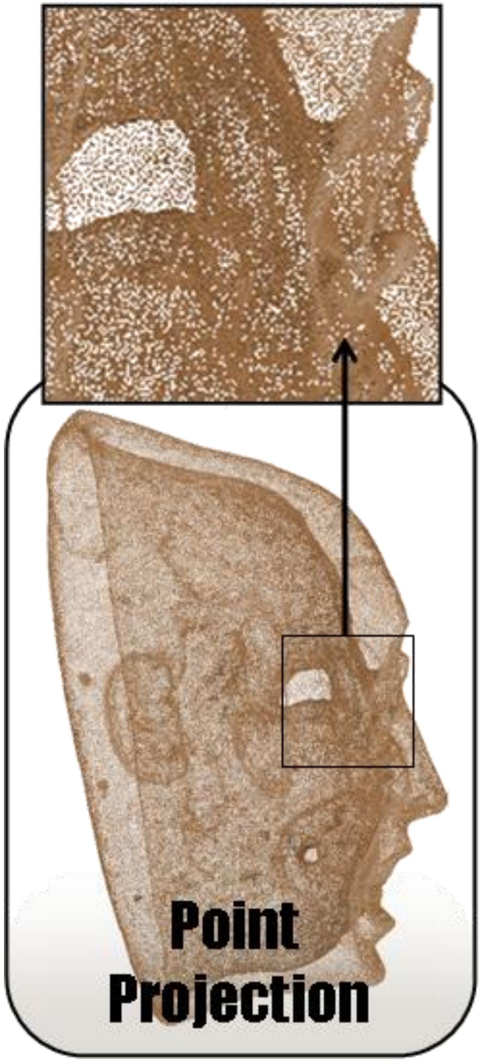
Point projection



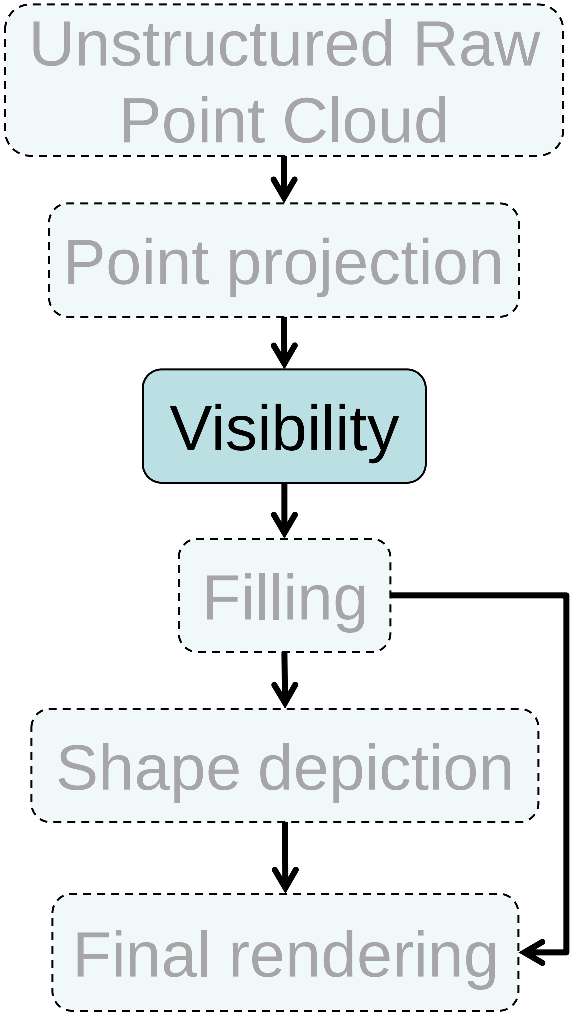


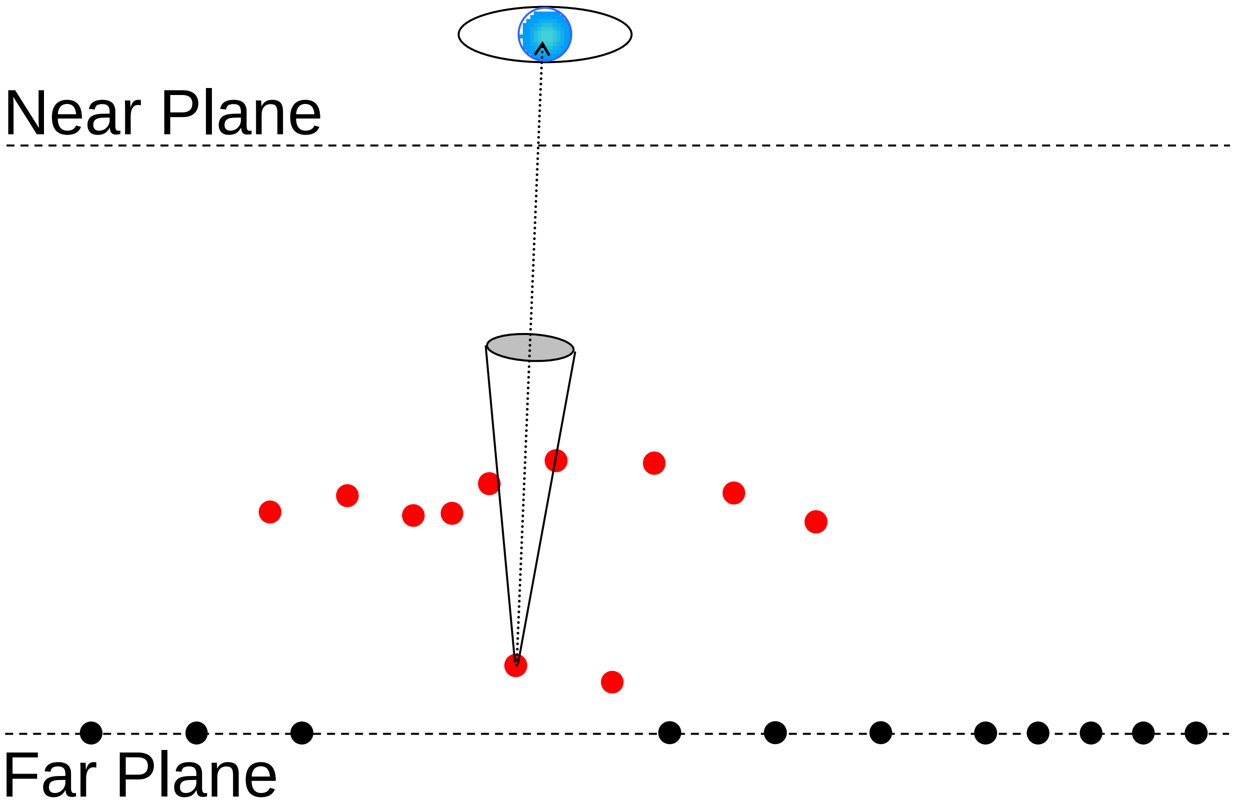
Point projection



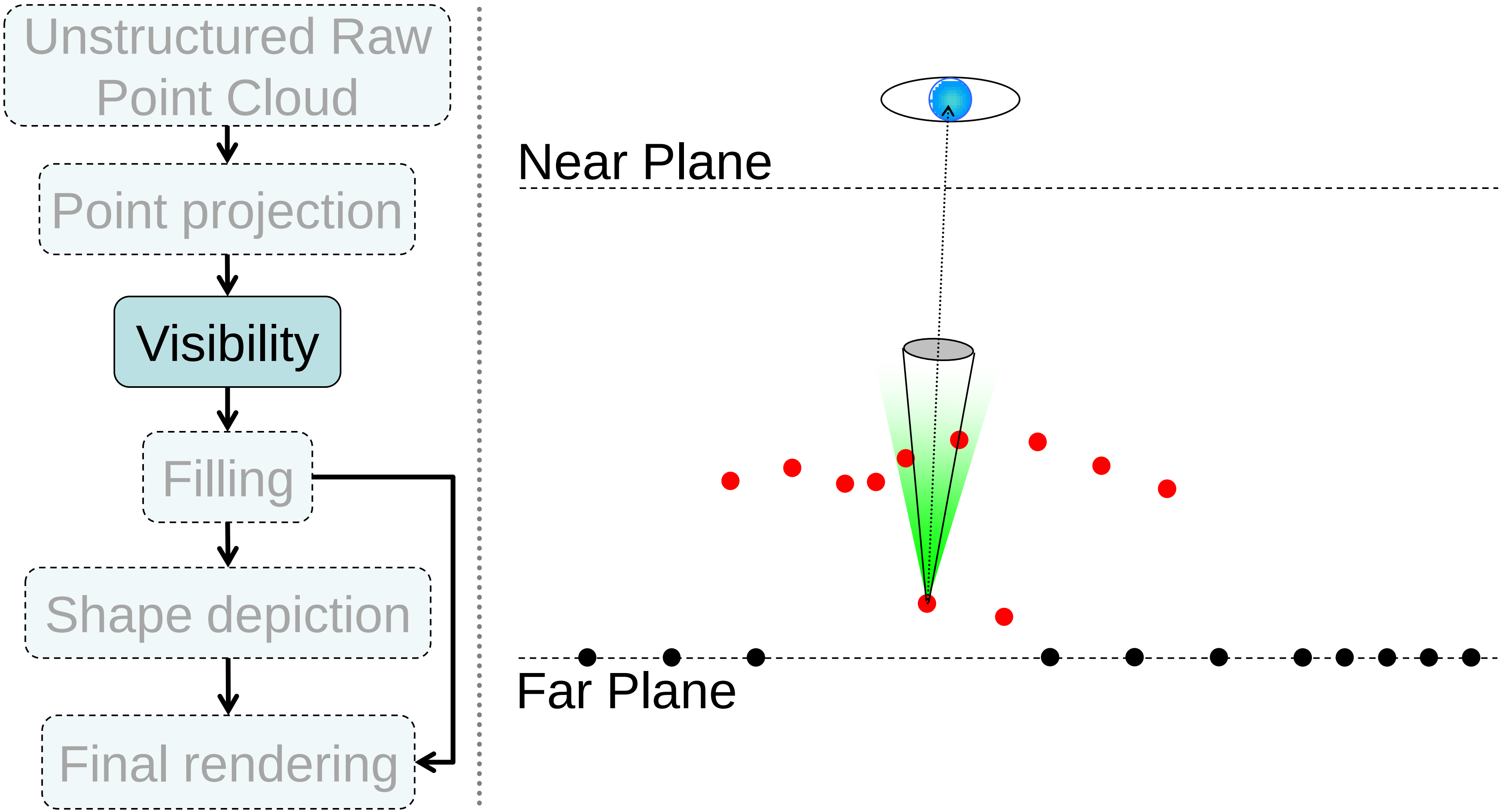


Visibility

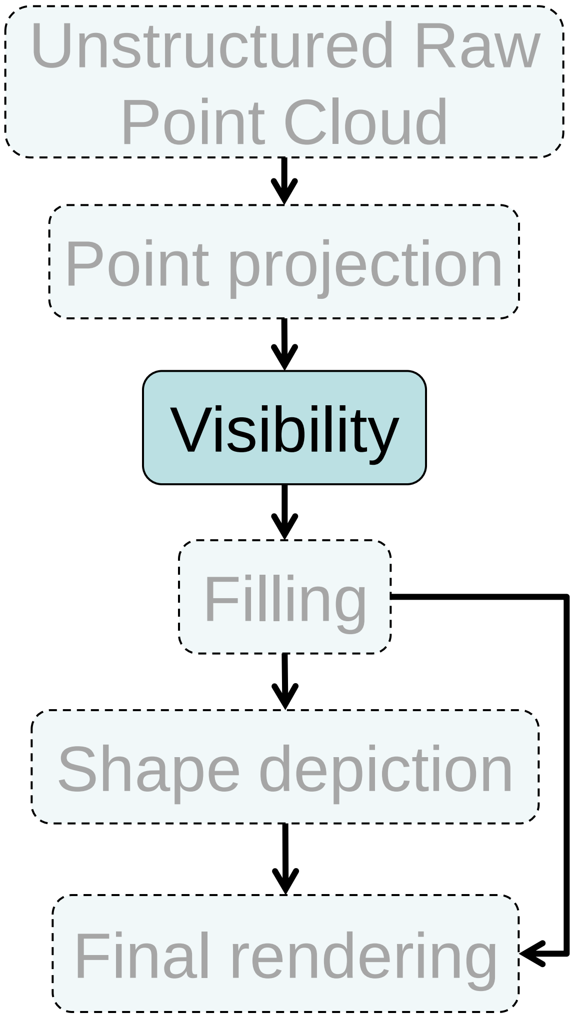

Unstructured Raw Point Cloud
Point projection
Visibility
Filling
Shape depiction
Final rendering

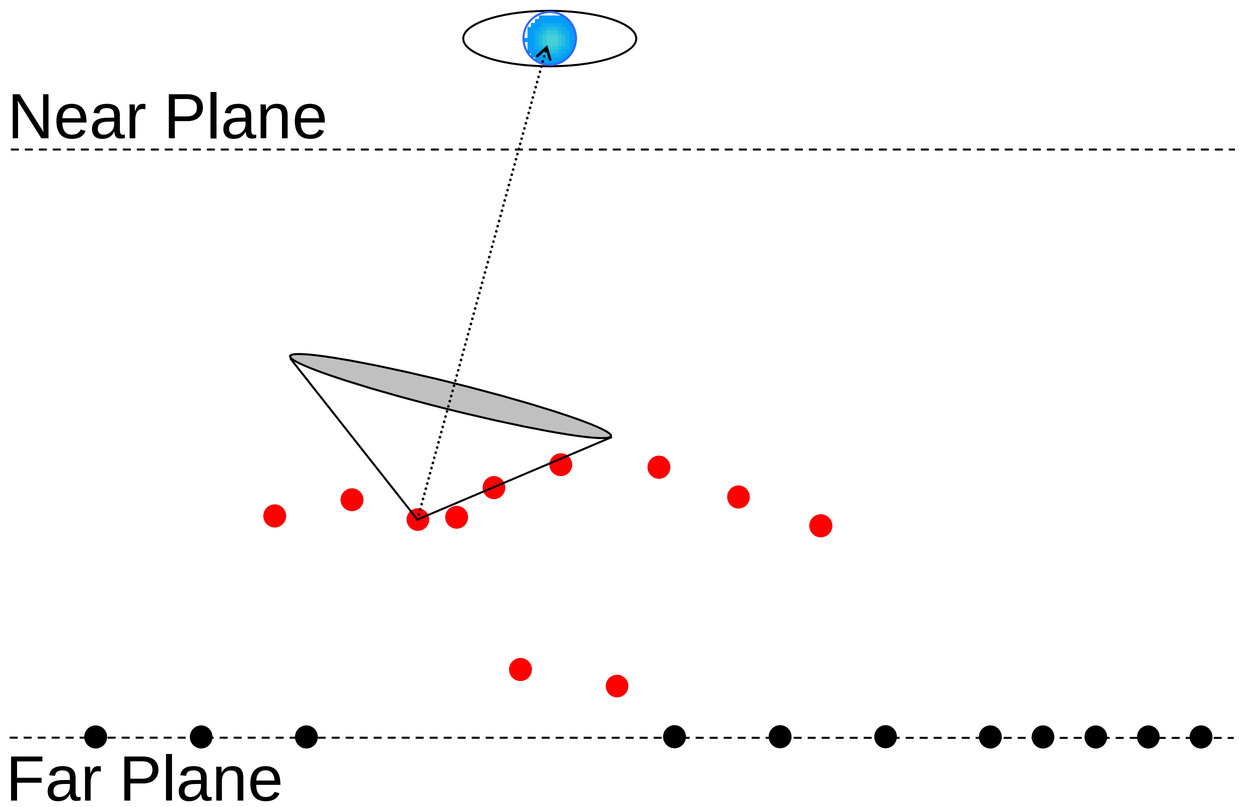

Near Plane
Far Plane

Visibility

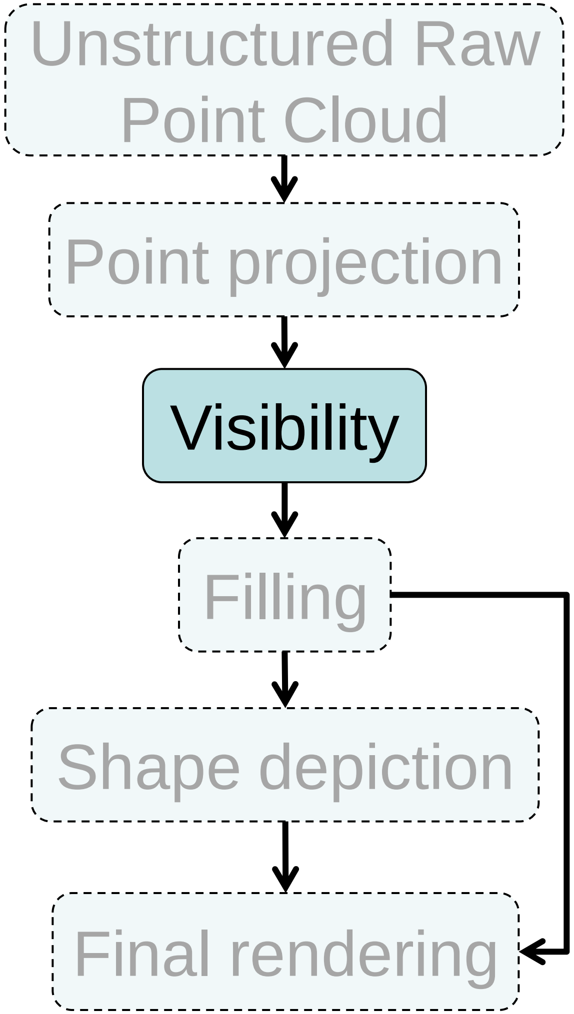

Unstructured Raw Point Cloud
Point projection
Visibility
Filling
Shape depiction
Final rendering
Near Plane
Far Plane

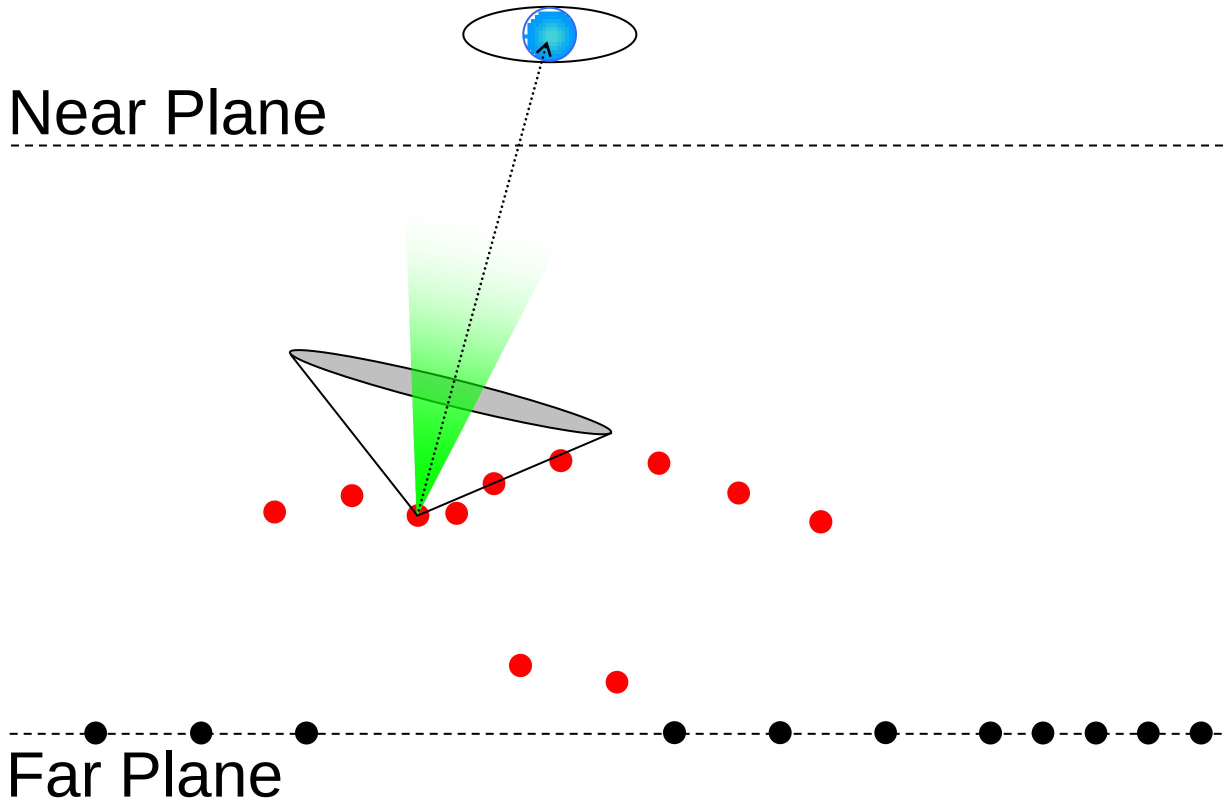
Visibility


Unstructured Raw Point Cloud
Point projection
Visibility
Filling
Shape depiction
Final rendering

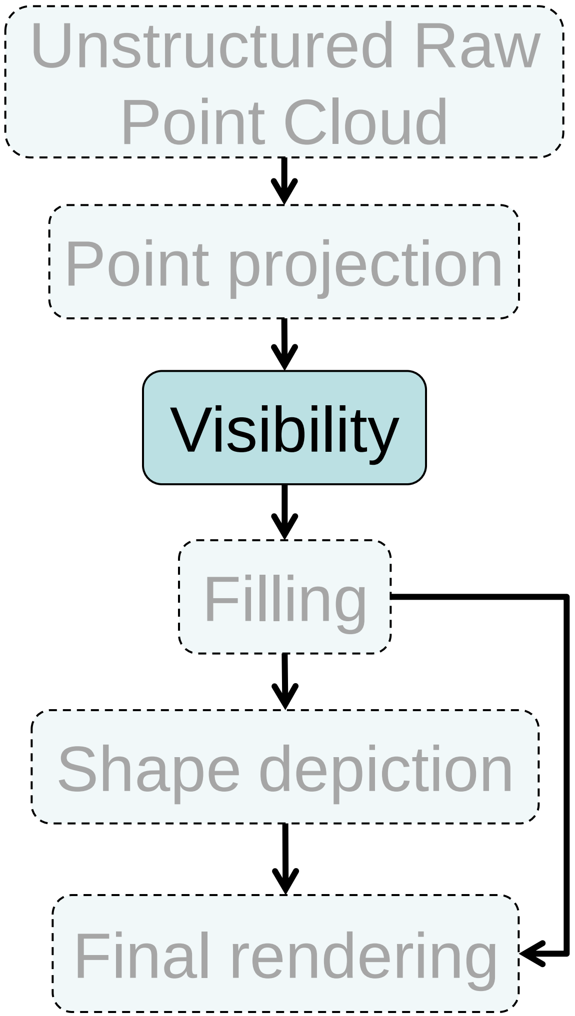

Near Plane
Far Plane

Visibility

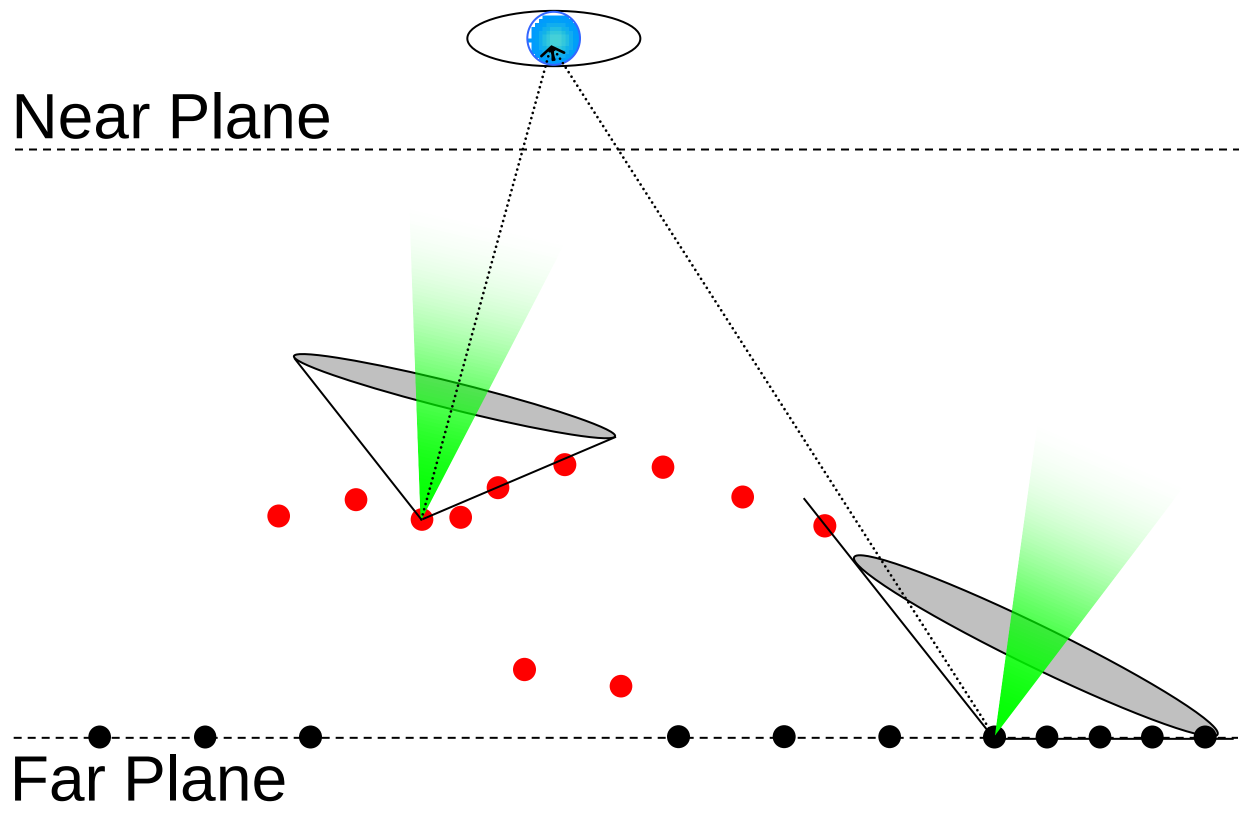

Unstructured Raw Point Cloud
Point projection
Visibility
Filling
Shape depiction
Final rendering


Near Plane
Far Plane

Visibility

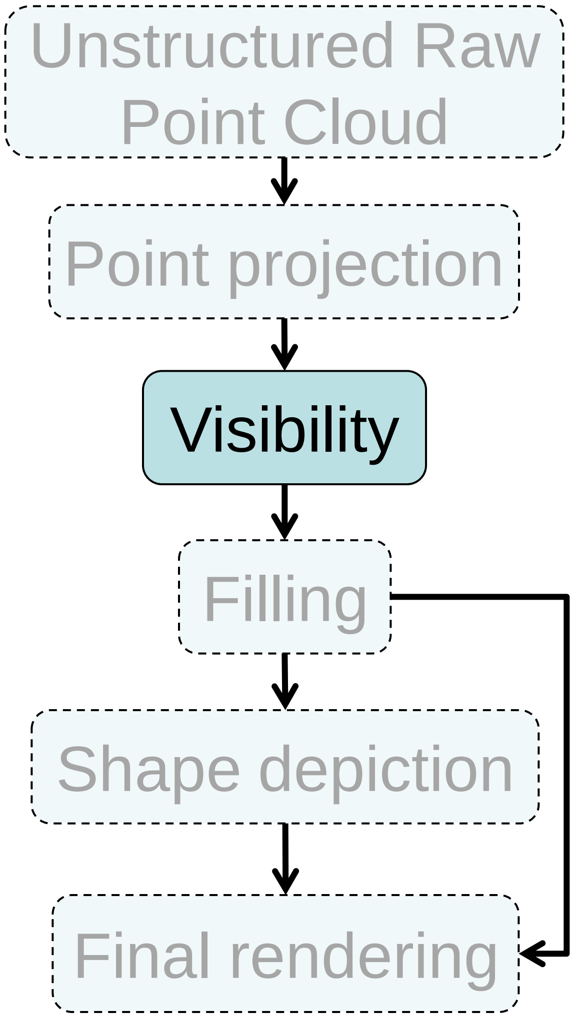


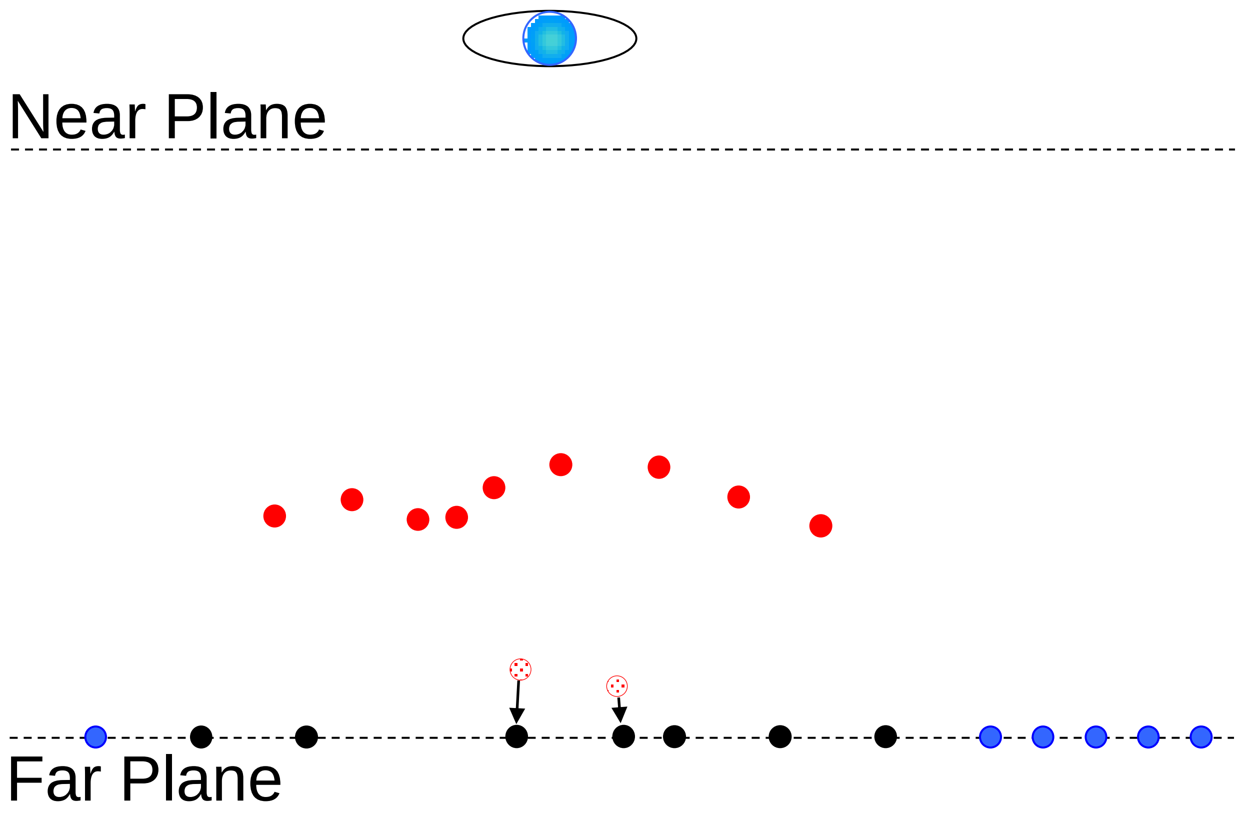
Unstructured Raw Point Cloud
Point projection
Visibility
Filling
Shape depiction
Final rendering



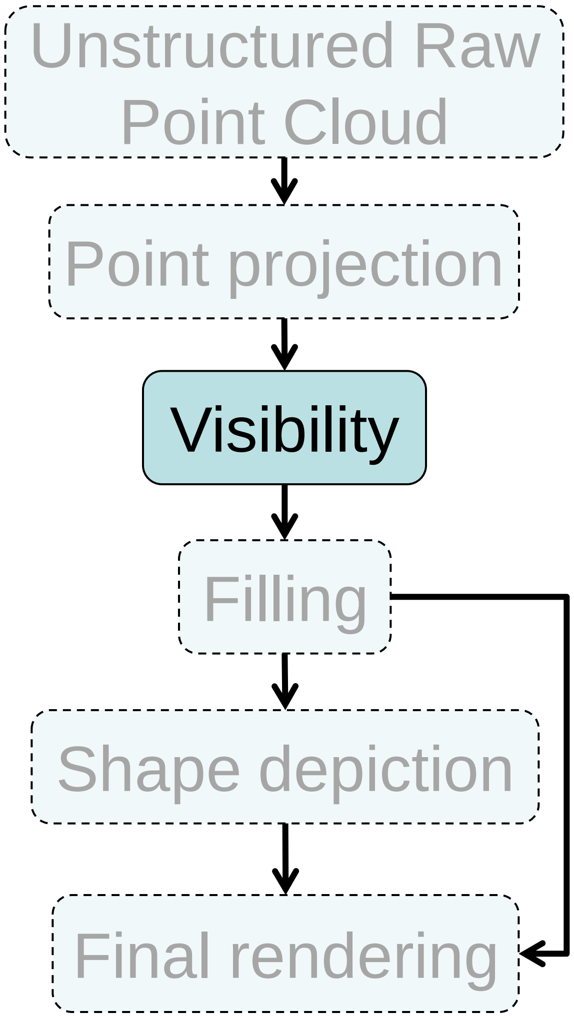
Near Plane
Far Plane

Visibility

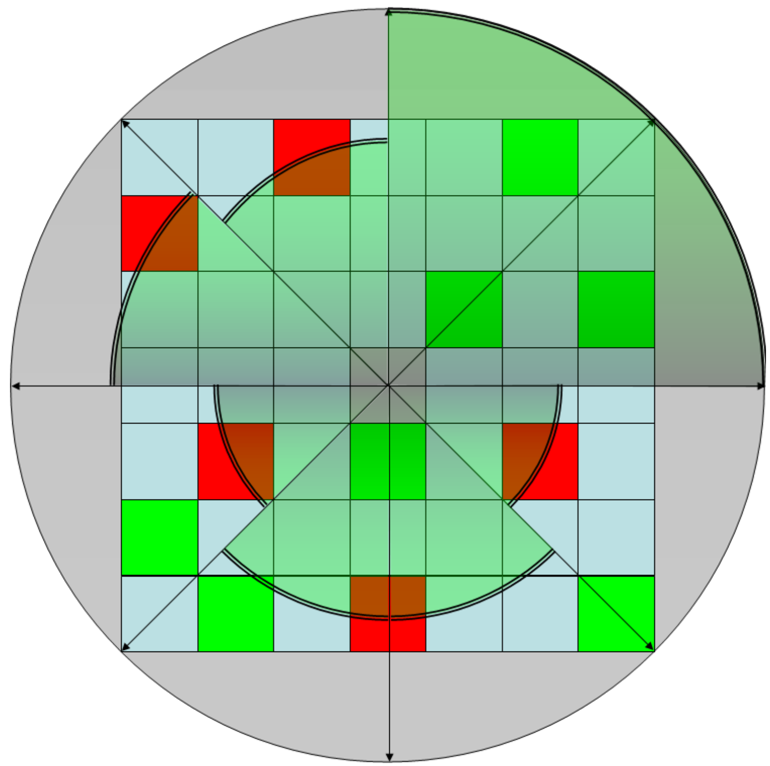

Unstructured Raw Point Cloud
Point projection
Visibility
Filling
Shape depiction
Final rendering


Near Plane
Far Plane

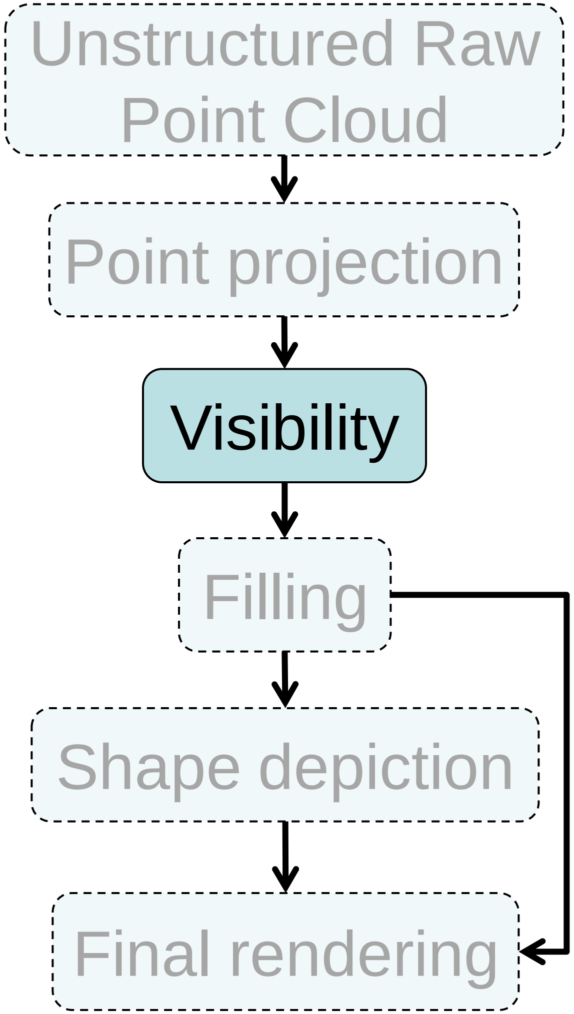
Visibility

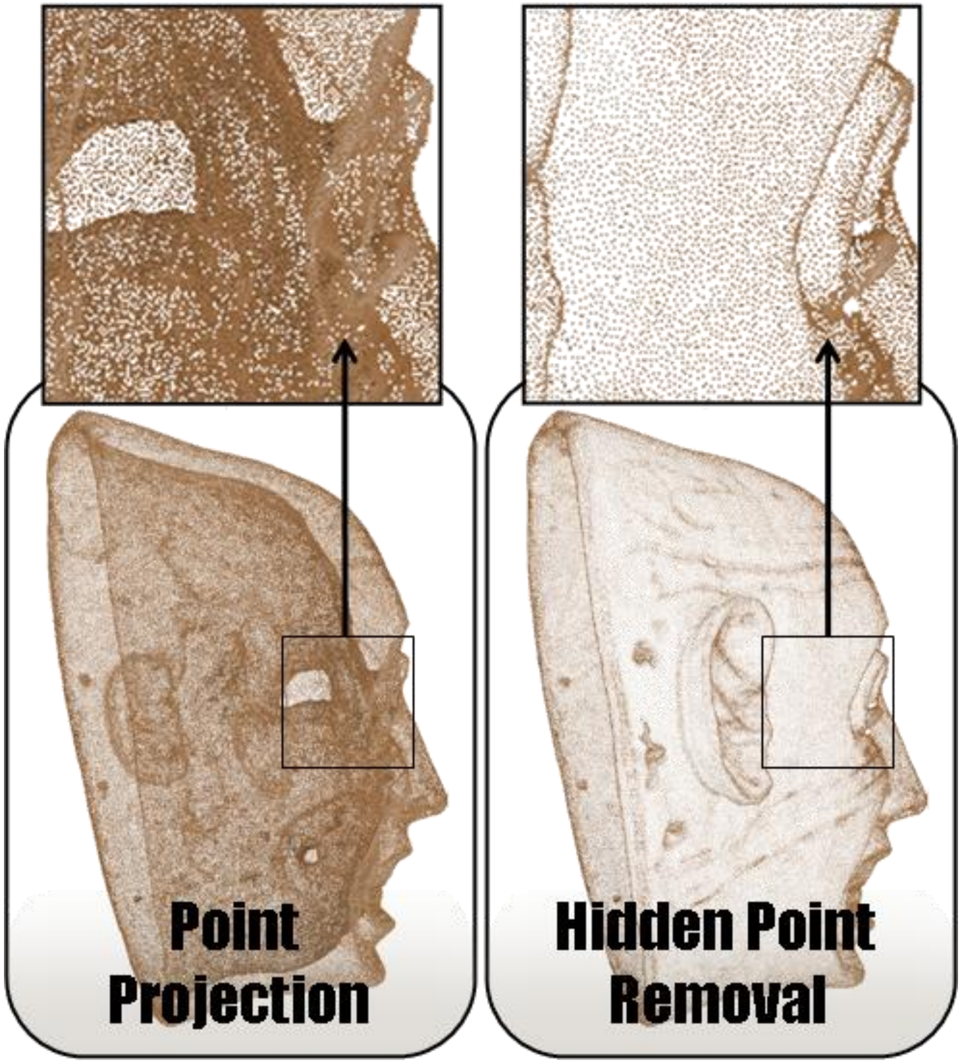


- **Screen-space implementation**

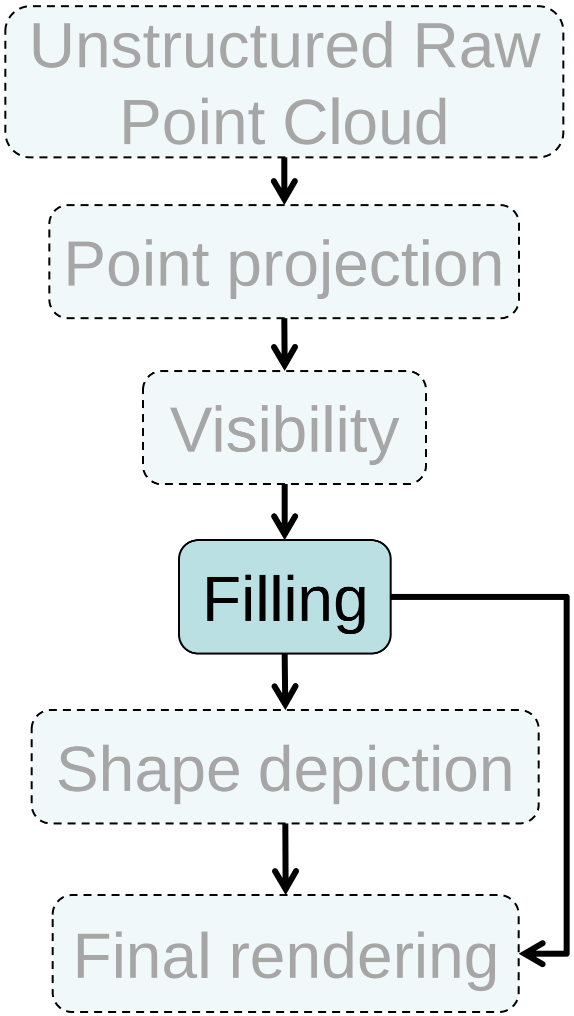


Visibility


Unstructured Raw Point Cloud
Point projection
Visibility
Filling
Shape depiction
Final rendering


Point Projection
Hidden Point Removal

Anisotropic filling

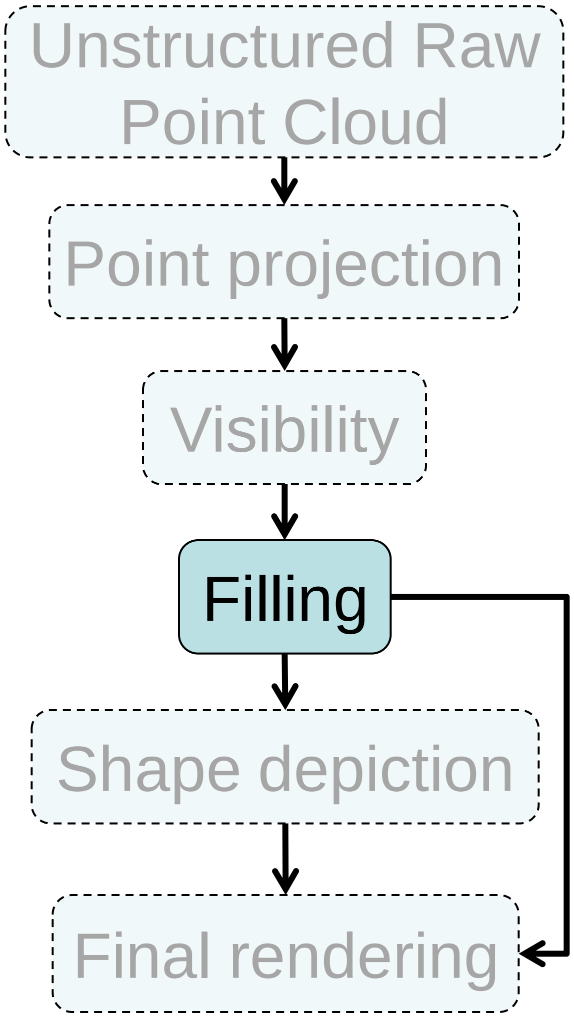


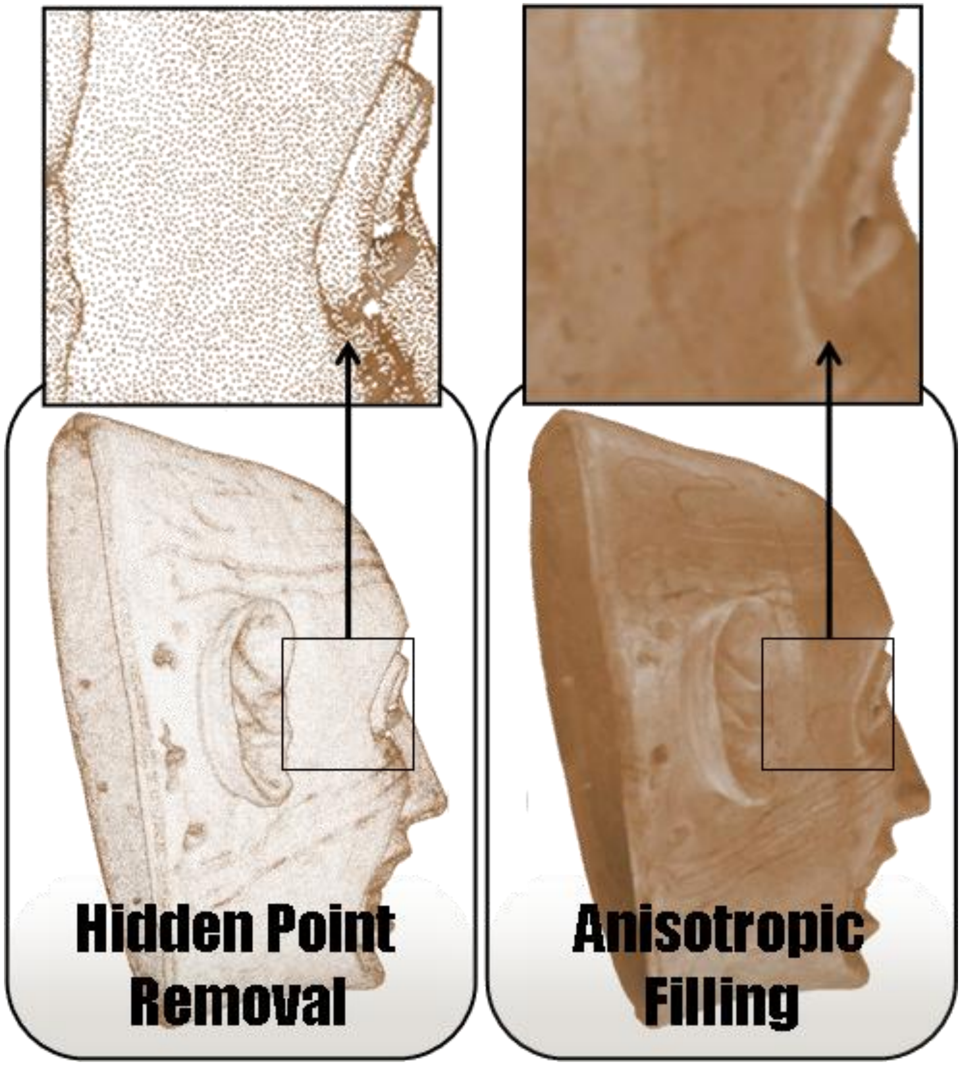
- **3D surface reconstruction →2D infilling**
- **Multi-pass GPU shader**

$$w_i = \left(1 - \frac{r_i}{2}\right)\left[1 - \min\left(1, \frac{\left|z_i - z_0\right|}{\delta}\right)\right]$$

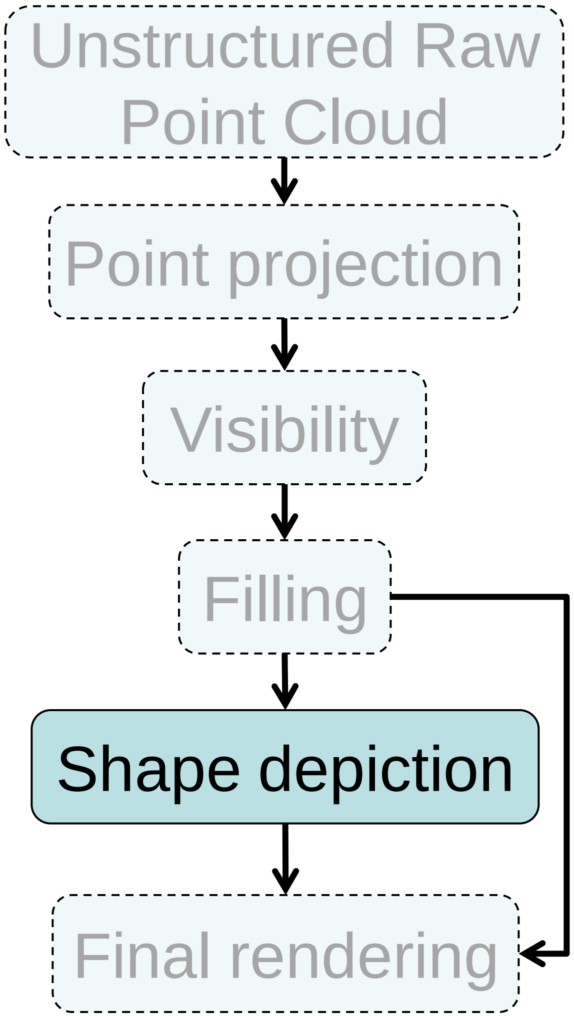
- r_i: Neighbor distance from Kernel center (pixel)
- z_i: Depth value of neighbor *i*
- δ: Depth filter dimension
- z_0: Depth value of current pixel

Anisotropic filling





Shape depiction

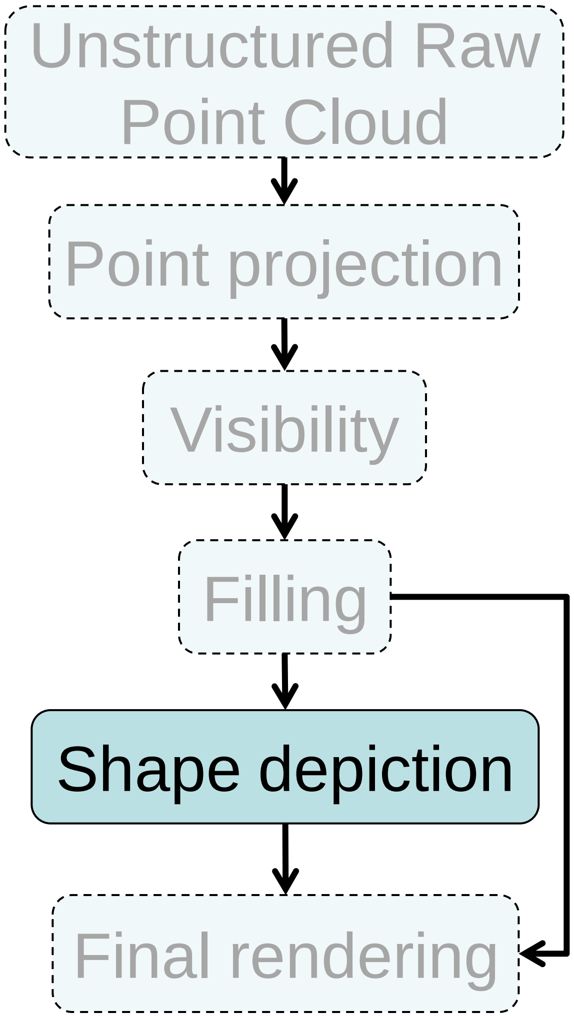


- **Deferred shading**
- **Multi-level shading term**
  - Similar to [Rusinkiewicz et al. 2006]
- **We start from D^0 (3x3 kernel)**

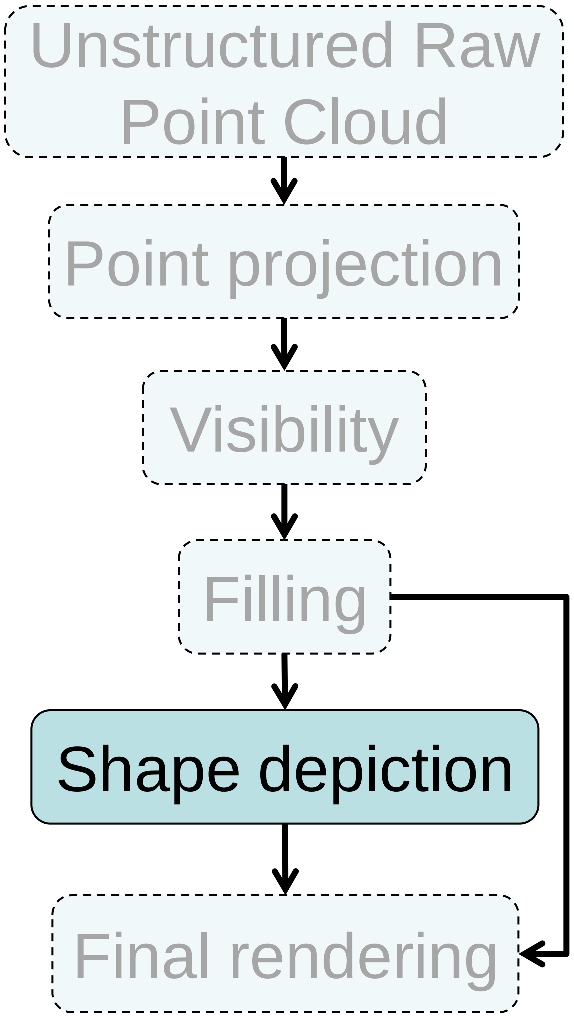
$$\begin{cases} d_{i_{\min}}^0 = \mu^0 - \sigma^0 \\ d_{i_{\max}}^0 = \mu^0 + \sigma^0 \end{cases}$$

$$\omega_i^0 = clamp_{[0\ldots1]}\left(1 - \frac{\left|d_i^0 - d_{i_{\min}}^0\right|}{d_{i_{\max}}^0 - d_{i_{\min}}^0 + \varepsilon}\right)$$

Shape depiction



Shape depiction



- **For a level** D^k → $d_i^{k+1} = \frac{\mu^k + d_{i_{\min}}^k}{2}$

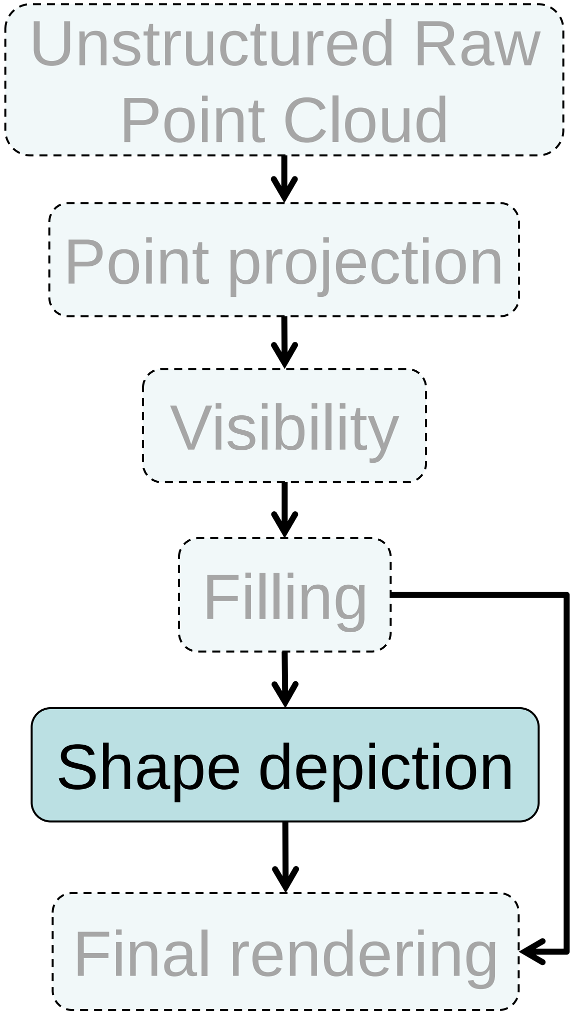
Shape depiction

Unstructured Raw Point Cloud
↓
Point projection
↓
Visibility
↓
Filling
↓
Shape depiction
↓
Final rendering

- **For a level** D^k → $d_i^{k+1} = \frac{\mu^k + d_{i_{\min}}^k}{2}$

$k = 0$

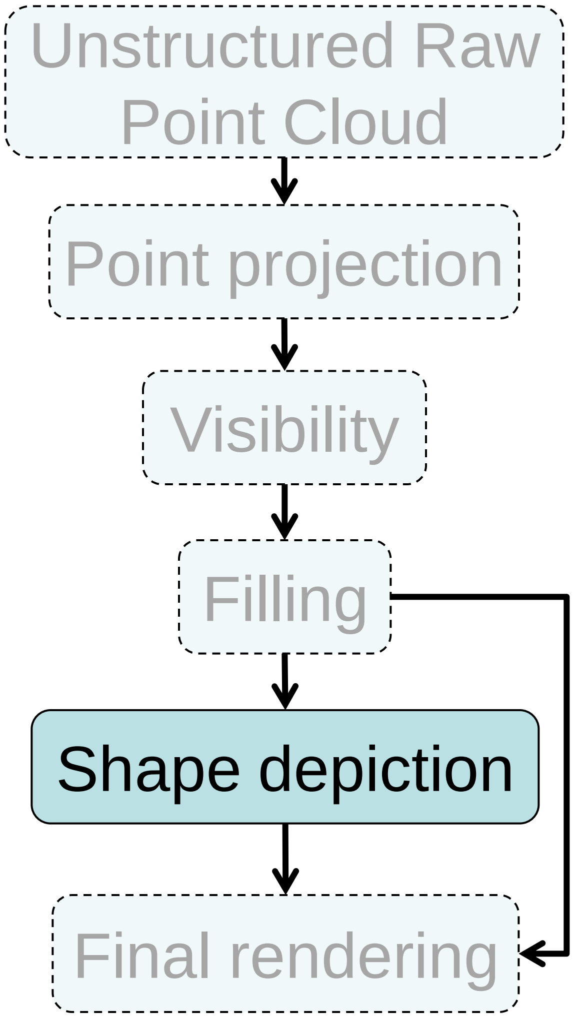
Shape depiction



- **For a level** D^k → $d_i^{k+1} = \frac{\mu^k + d_{i_{\min}}^k}{2}$

$k = 10$

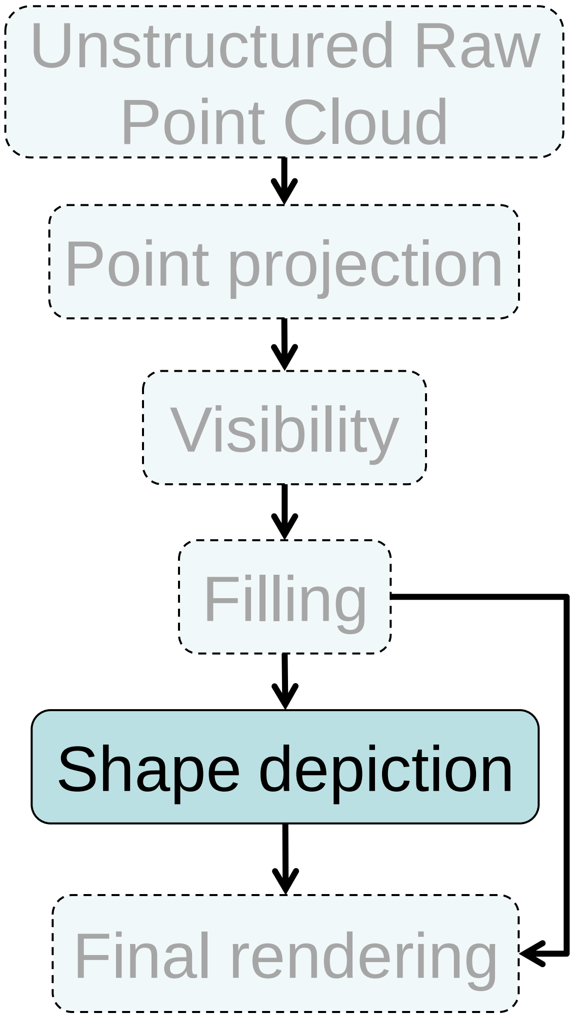
Shape depiction



- **For a level** D^k → $d_i^{k+1} = \frac{\mu^k + d^k_{i_{\min}}}{2}$

$k = 20$

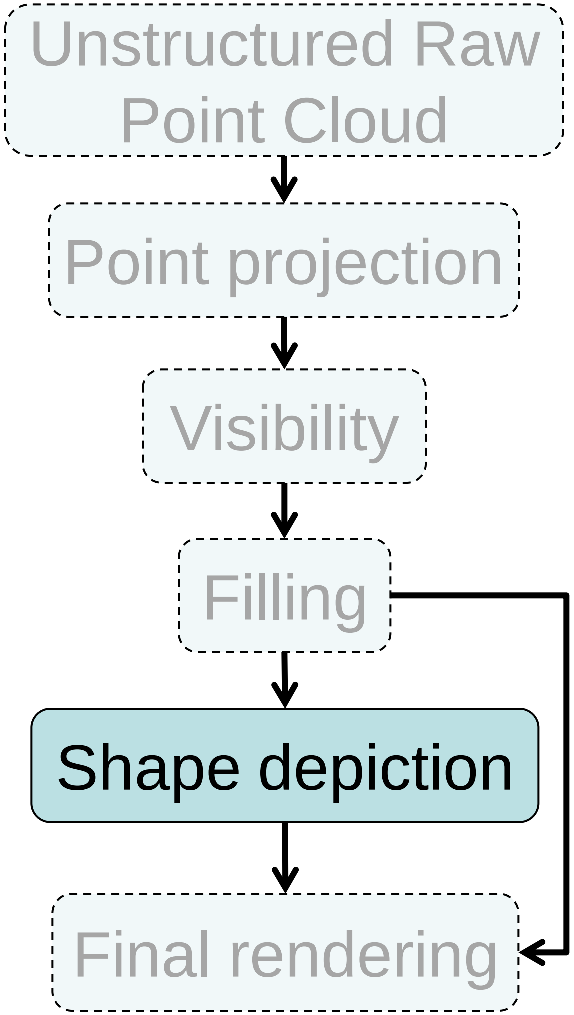
Shape depiction



- **For a level** D^k → $d_i^{k+1} = \dfrac{\mu^k + d_{i_{\min}}^k}{2}$

$k = 30$

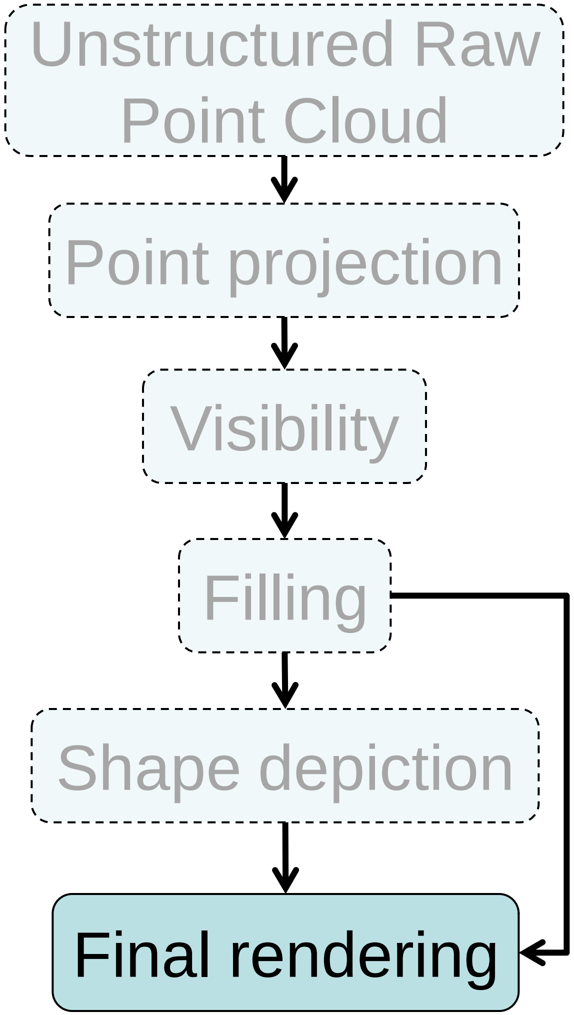
Shape depiction

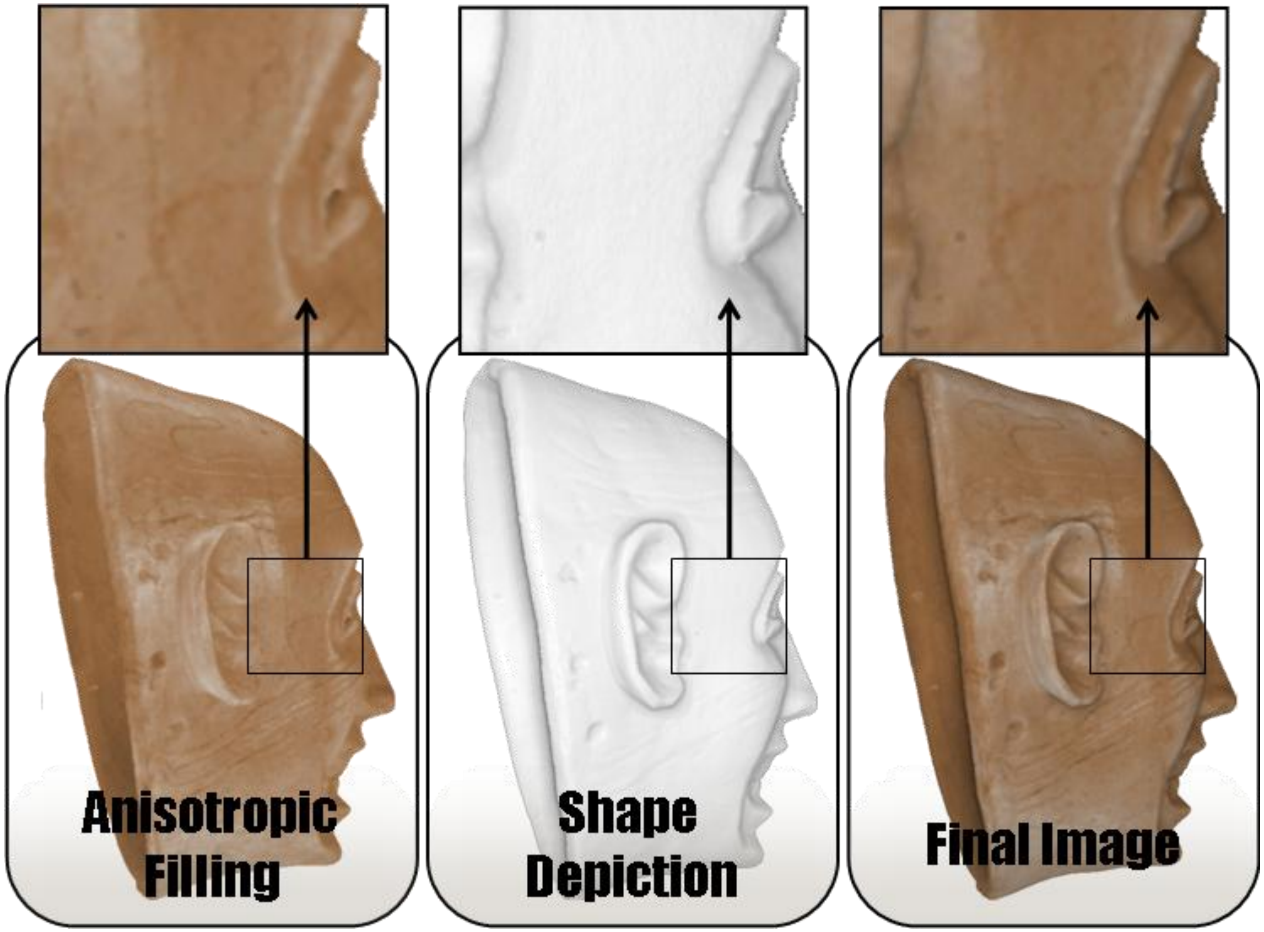


- **Merging levels →** $\omega_i = \sum_{k=0}^{K-1} \frac{\omega_i^k}{(k+1)^{\xi}}$

$$\xi = \frac{3}{2}$$

Final rendering





Results

- **Massive unstructured raw point clouds**
- **Integrated in a pipeline for real-time out-of-core rendering**
- **Datasets**
  - Sirmione – 100Mpts
  - Toronto – 260Mpts
  - Loggia (Brescia) – 110Mpts

Results – Video

Sirmione – 100Mpts

Point projection

Sirmione – 100Mpts

Pull-push method

Sirmione – 100Mpts

Our method

Sirmione – 100Mpts

Pull-push method

Sirmione – 100Mpts

Our method

Sirmione – 100Mpts

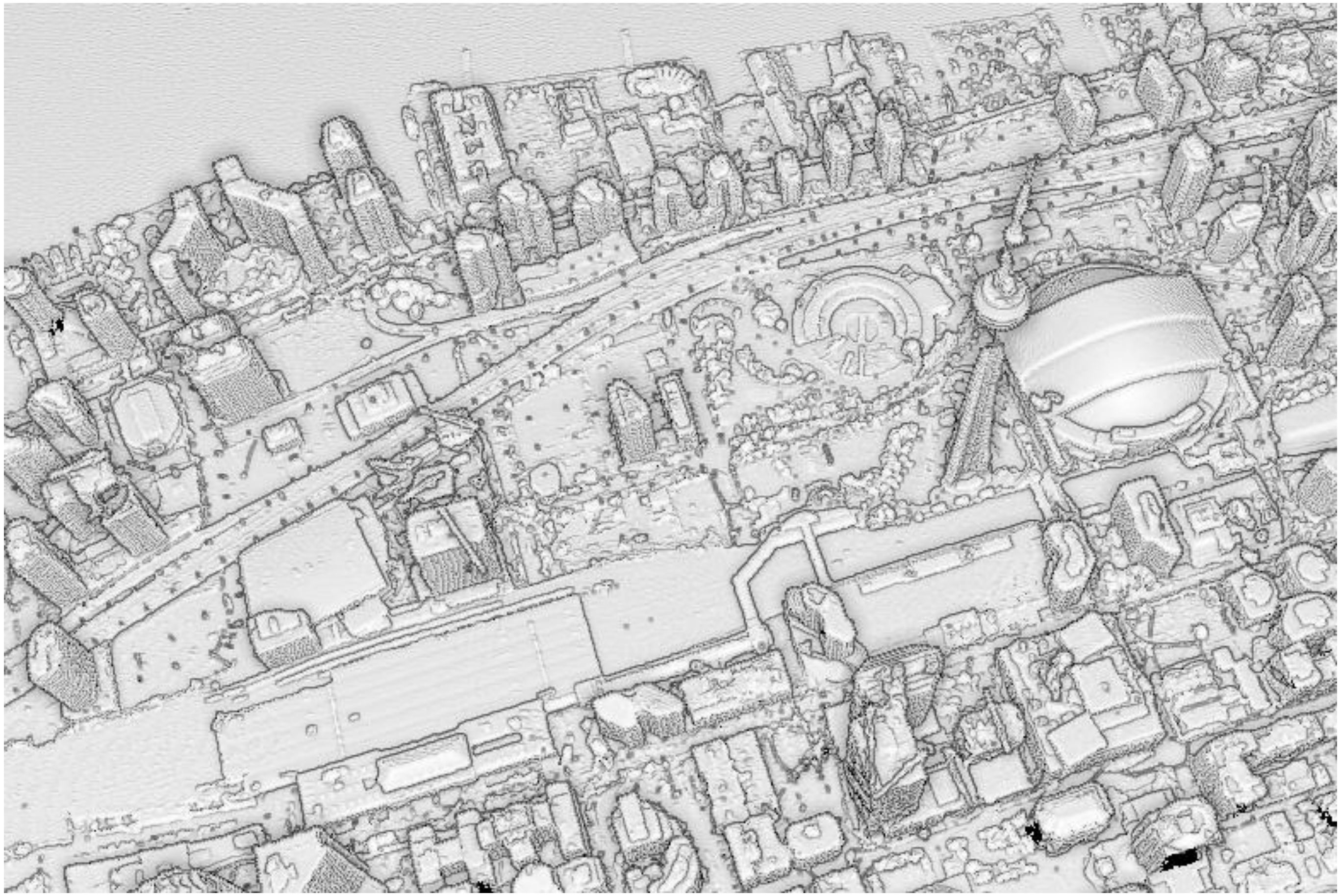
Our method with shape depiction

Toronto – 260Mpts

Point Splat – Noisy dataset

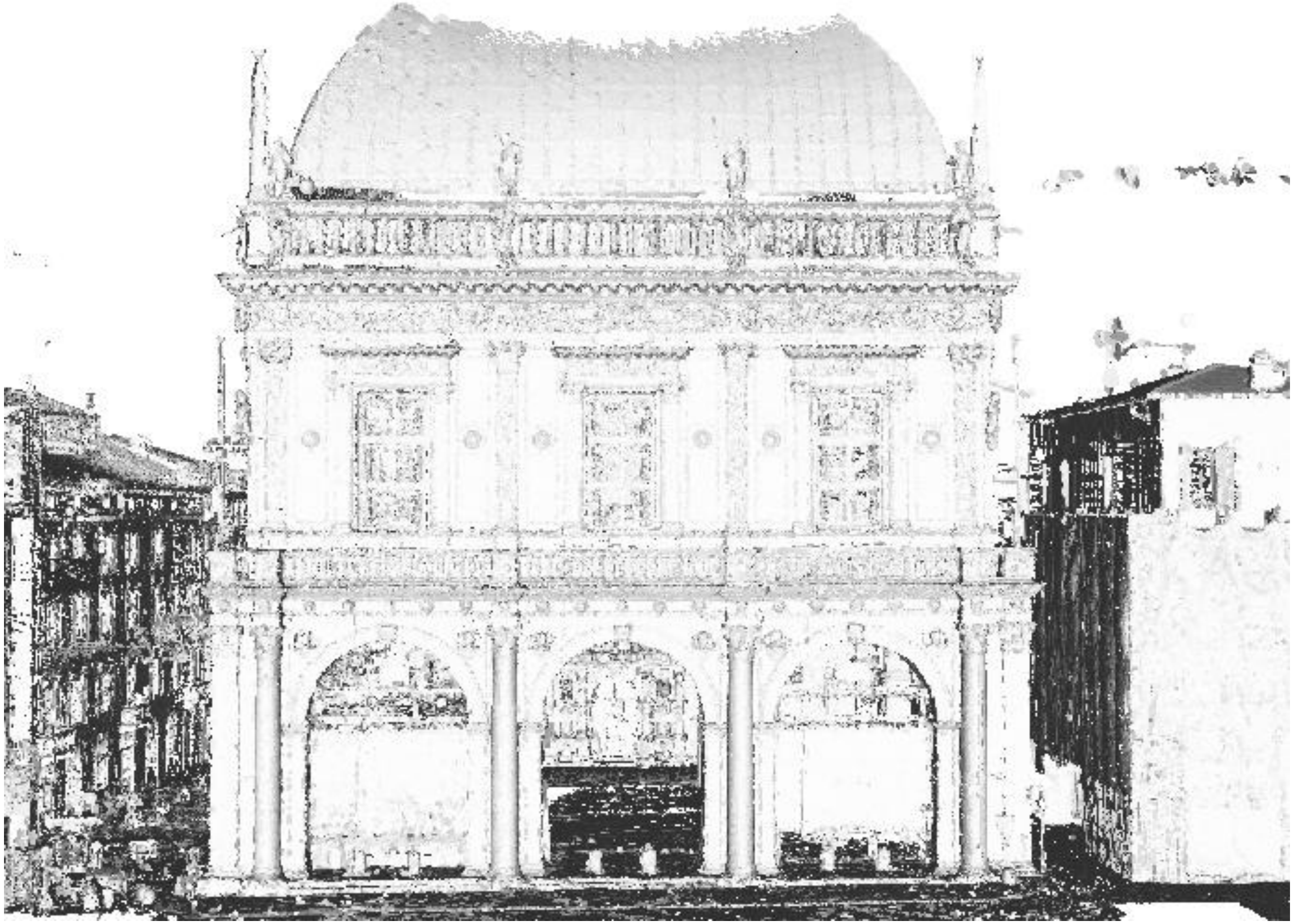
Toronto – 260Mpts

Our Rendering – Noisy dataset



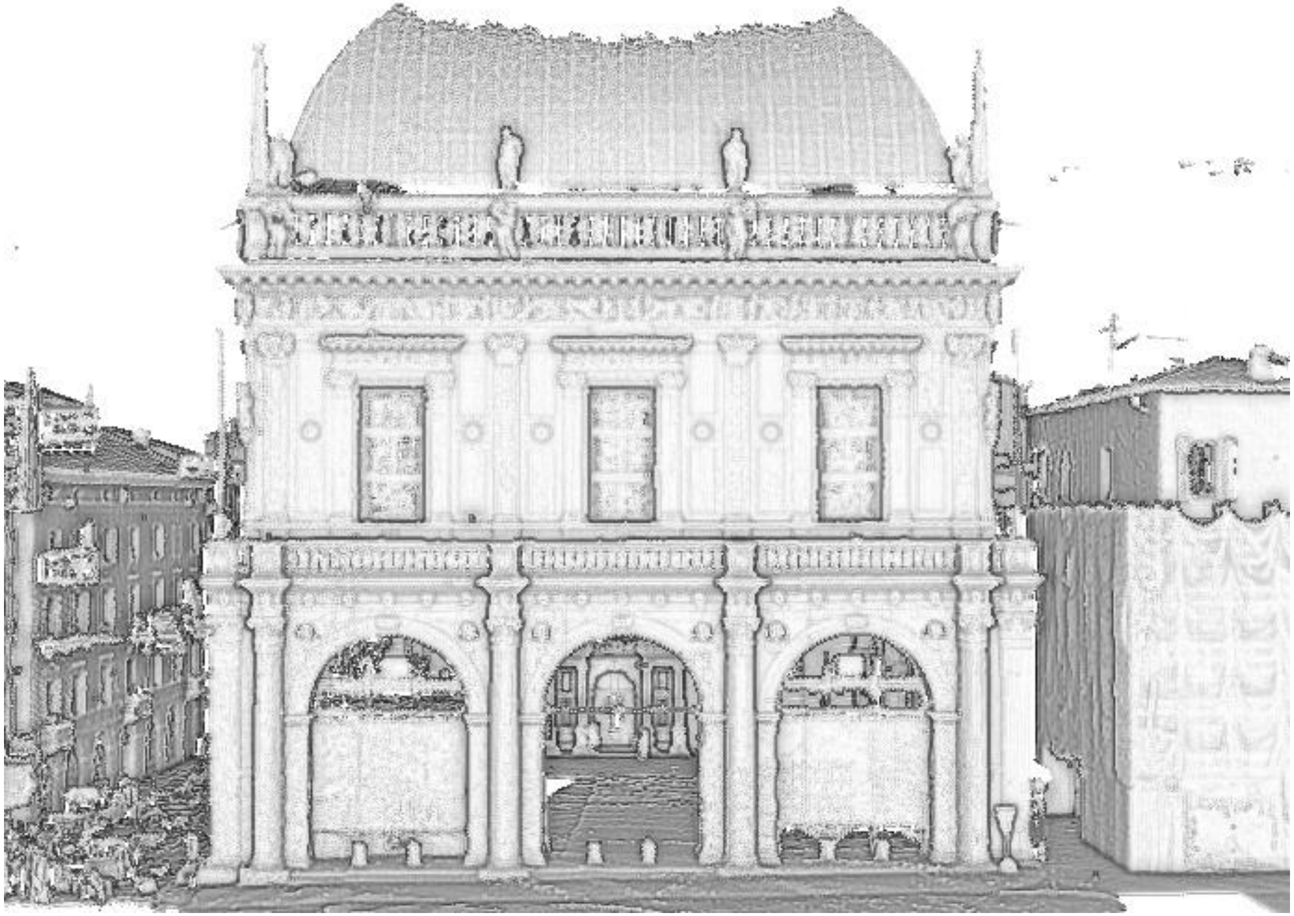
Loggia (Brescia) – 110Mpts

Point Splat – Noisy dataset



Loggia (Brescia) – 110Mpts

Our Rendering – Noisy dataset



Conclusion

- **PBR pipeline comparable to state-of-the-art techniques**
- **Unstructured raw point clouds**
- **Screen-space operators**
- **Real-time with massive models**
  - Multi-resolution out-of-core structure
  - Minimizing pre-computation
- **Robust to noise**
- **Suitable for quick on-site visualizations**

www.crs4.it/vic/

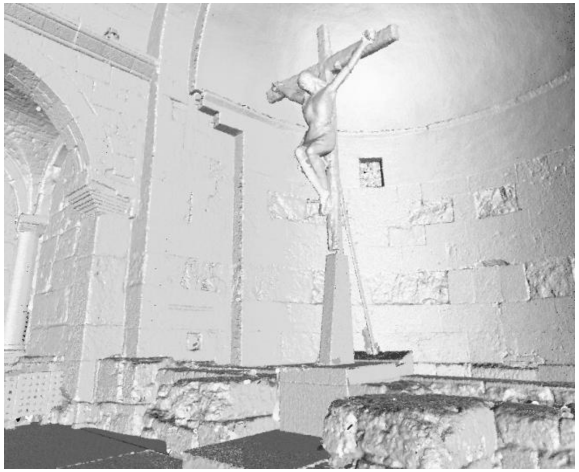
vcg.isti.cnr.it/

Photo Mapping

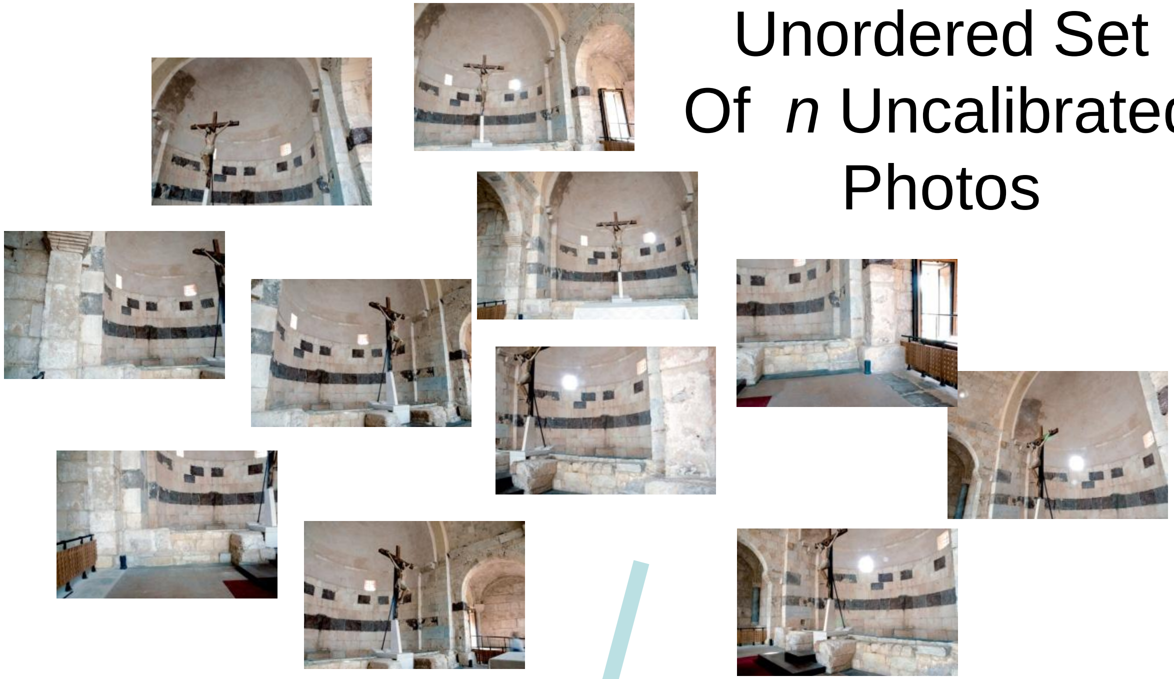
Ruggero Pintus, Enrico Gobbetti, and Roberto Combet. "Fast and Robust Semi-Automatic Registration of Photographs to 3D Geometry". In The 12th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, October 2011.

Problem Statement

3D Geometry



Unordered Set
Of *n* Uncalibrated
Photos

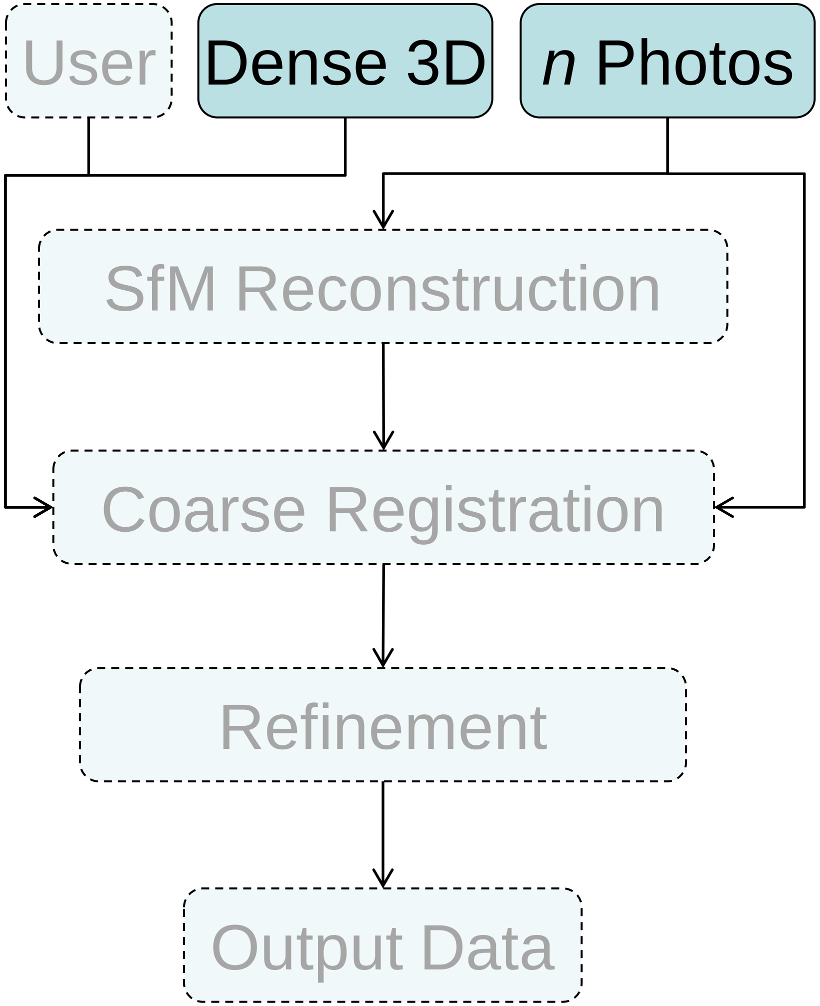


n Camera Poses
(2D/3D Registration)

Related work

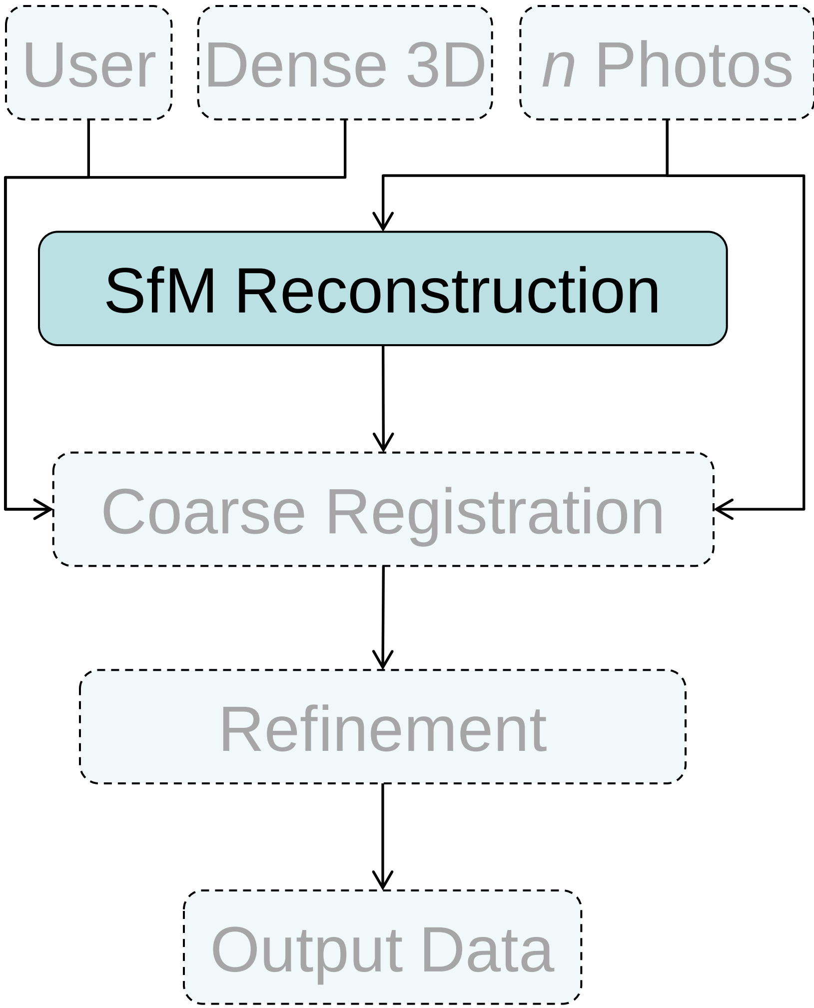
- **Manual selection of 2D-3D matches**
  - Massive user intervention – Tiring and time-consuming
- **Automatic feature matching**
  - Not robust enough for a generic dataset
- **Semi-automatic statistical correlation**
  - Point cloud attributes not always provided
- **Geometric multi-view reconstruction**
  - 2D-3D problem → 3D-3D registration task
  - dense and ordered frame sequence
- **Our contribution**
  - Minimize user intervention / Large datasets / Semi-automatic / Multi-view based approach / No Attributes

Input Data



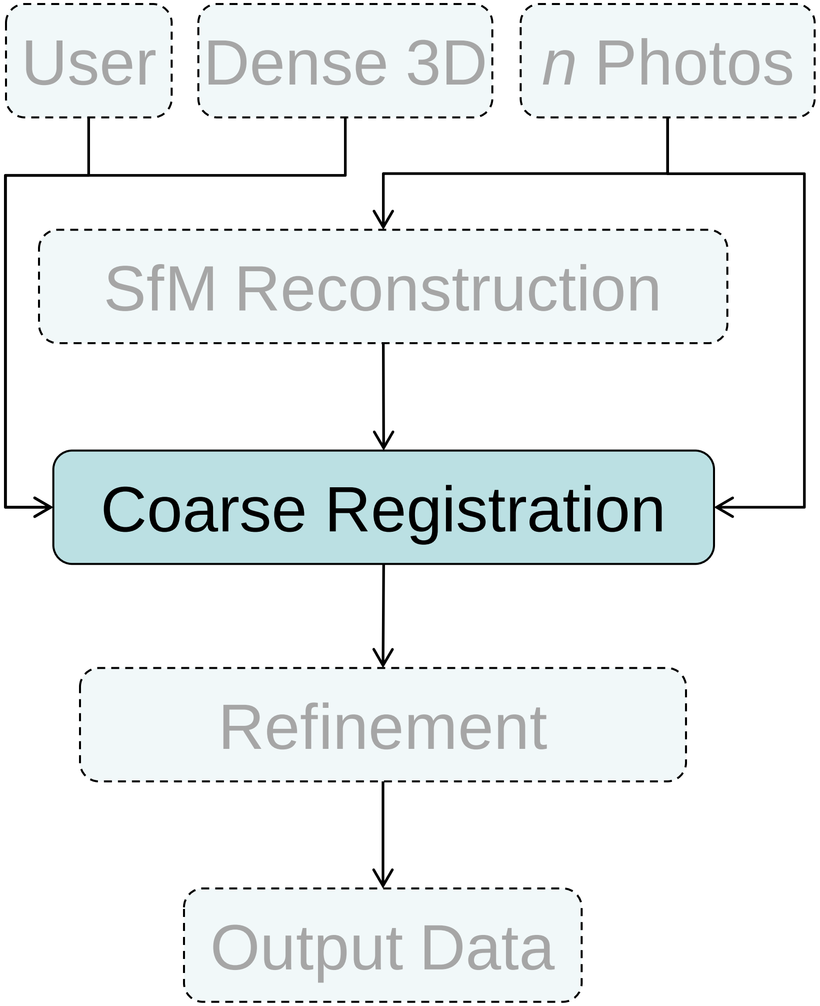
- **Dense Geometry**
  - Point cloud, triangle mesh, etc.
  - No attributes
  - No particular features
- **n photos**
  - Naïve constraints:
    - Blur, Noise, Under- or over-exposured
  - Sufficient overlap

Multi-view



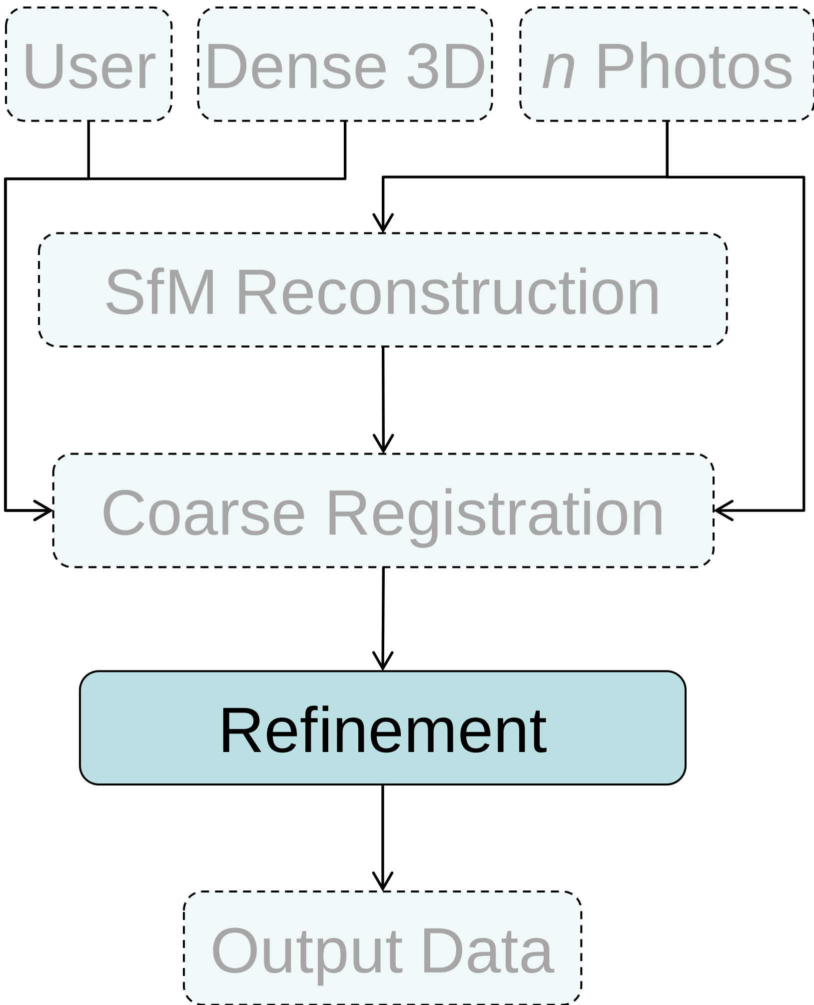
- **Bundler [Snavely et al. 2006]**
  - SfM system for unordered image collections
  - http://phototour.cs.washington.edu/bundler/
- **Output**
  - A sparse point cloud
  - n camera poses
  - SIFT keypoints (projections of sparse 3D points)

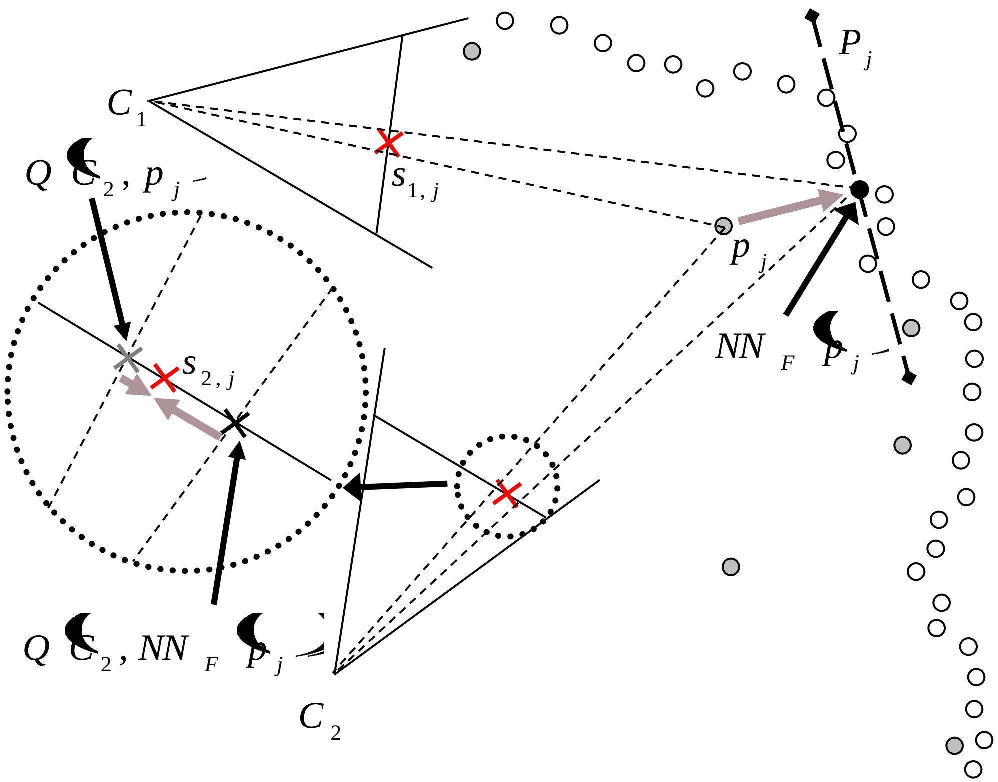
Coarse registration



- **Register two point clouds with different:**
  - scales
  - reference frames
  - resolutions
- **Automatic methods are not robust and efficient enough**
- **User aligns few images (one or more) to the dense geometry**
- **Affine transformation is applied to all cameras and sparse points**

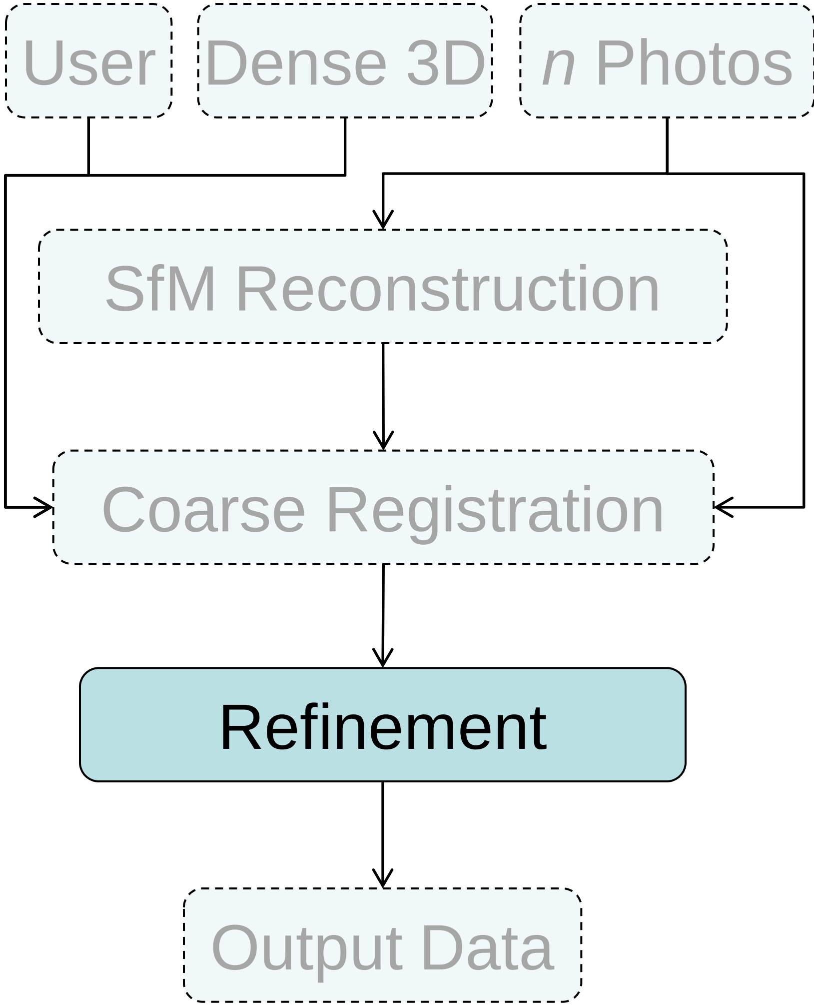
Refinement





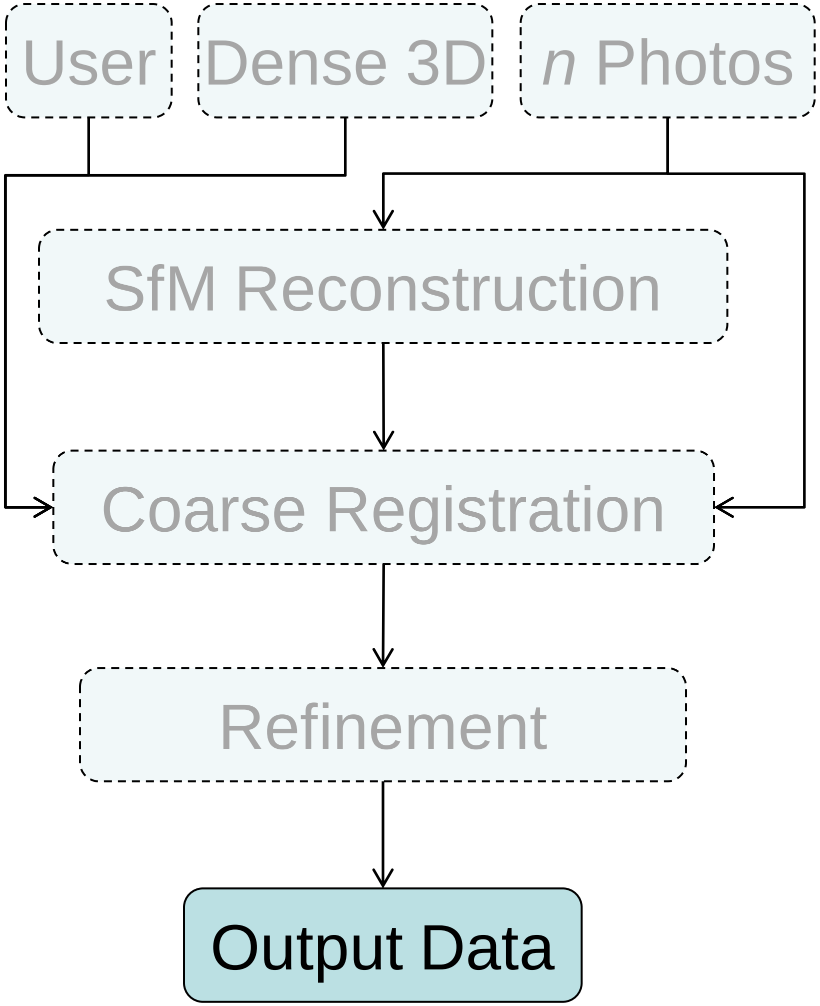
$$E(C, P) = \sum_{j=1}^{N_P} \sum_{i=1}^{N_C} v_{ij} \left\| Q(C_i, NN_F(p_j)) \; s_{i,j} \right\|^2$$

Refinement



- **Sparse Bundle Adjustment (SBA)**
  - Constants – SIFT keypoints, dense 3D points
  - Variables – Camera poses, sparse 3D points
  - SBA
    - A Generic SBA C/C++ Package Based on the Levenberg-Marquardt Algorithm
    - http://www.ics.forth.gr/~lourakis/sba/

Output data



- **n camera poses**
- **Input of photo blending**
  - n photos
  - n camera poses
  - Dense 3D geometry

Results – Photo mapping

Model	3D (# Points)	Images (wxh)	SfM (# Points)	Manual 2D/3D	Coarse	Fine	Total
Grave	8.3M	21 (1936x1296)	17m36s(19K)	3m/1 photo	4s	18m20s	40m
Church's Apse	14M	40 (1936x1296)	10m50s(7.6K)	7m/2 photos	6s	13m40s	32m
Church's Detail	4.7M	49 (1936x1296)	28m14s(17K)	4m/1 photo	8s	22m50s	55m

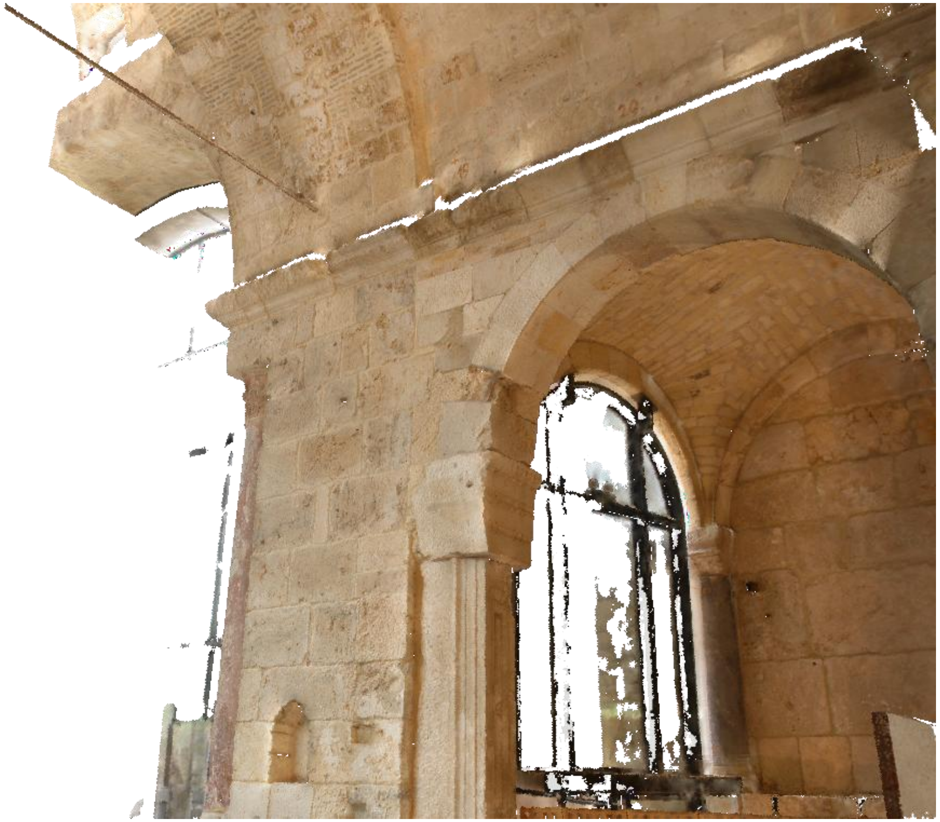
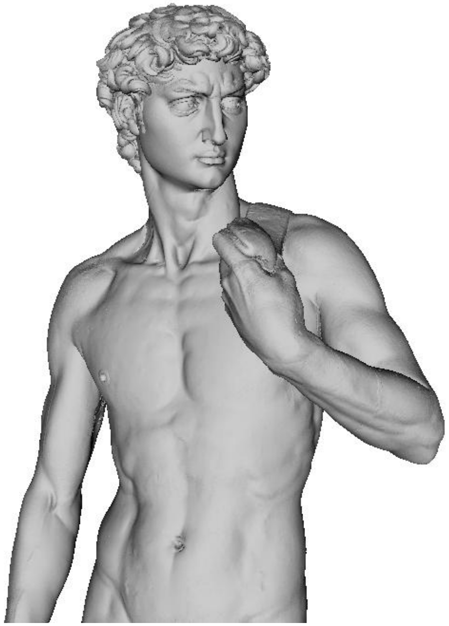


Photo Blending

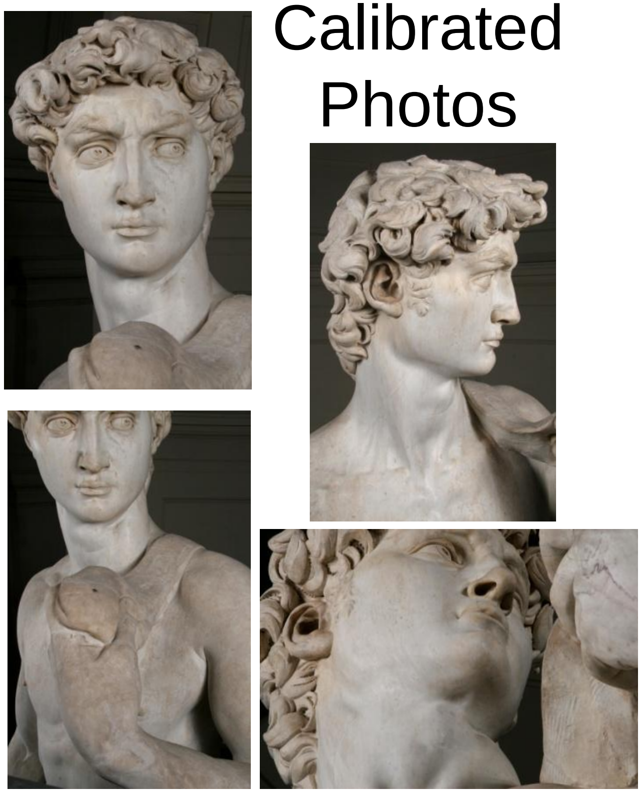
Ruggero Pintus, Enrico Gobbetti, and Marco Callieri. A Streaming Framework for Seamless Detailed Photo Blending on Massive Point Clouds. In Proc. Eurographics Area Papers. Pages 25- 32, 2011.

Problem Statement

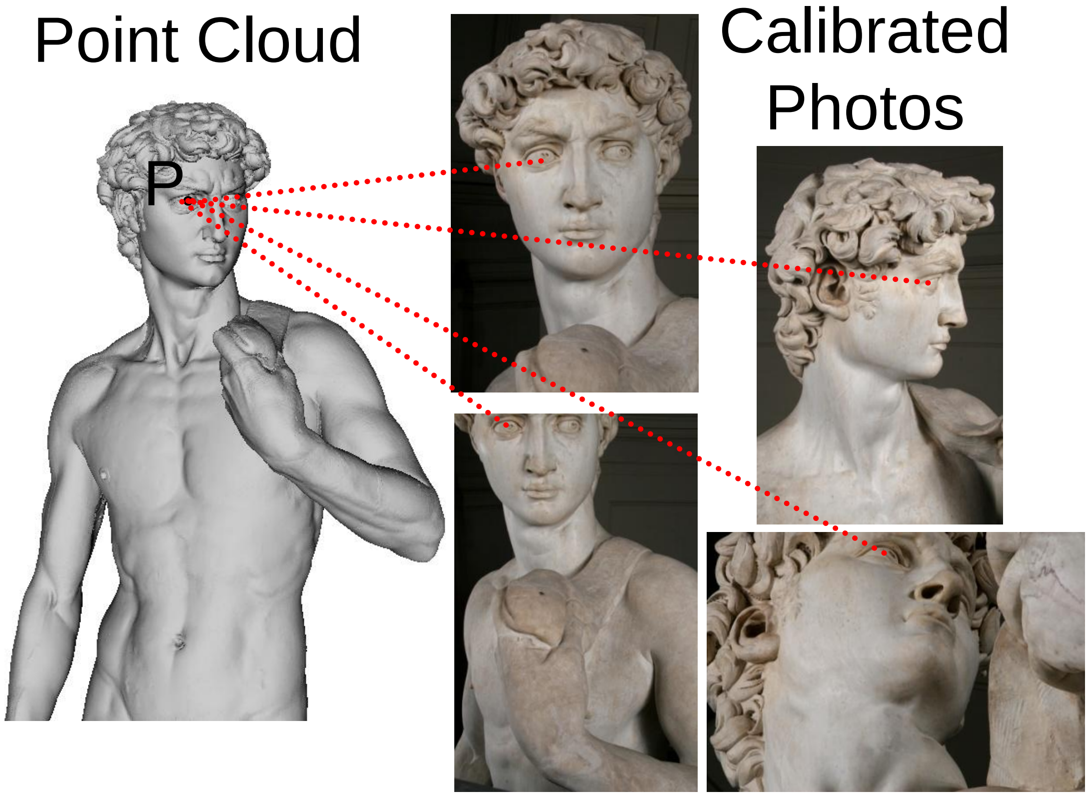
Point Cloud



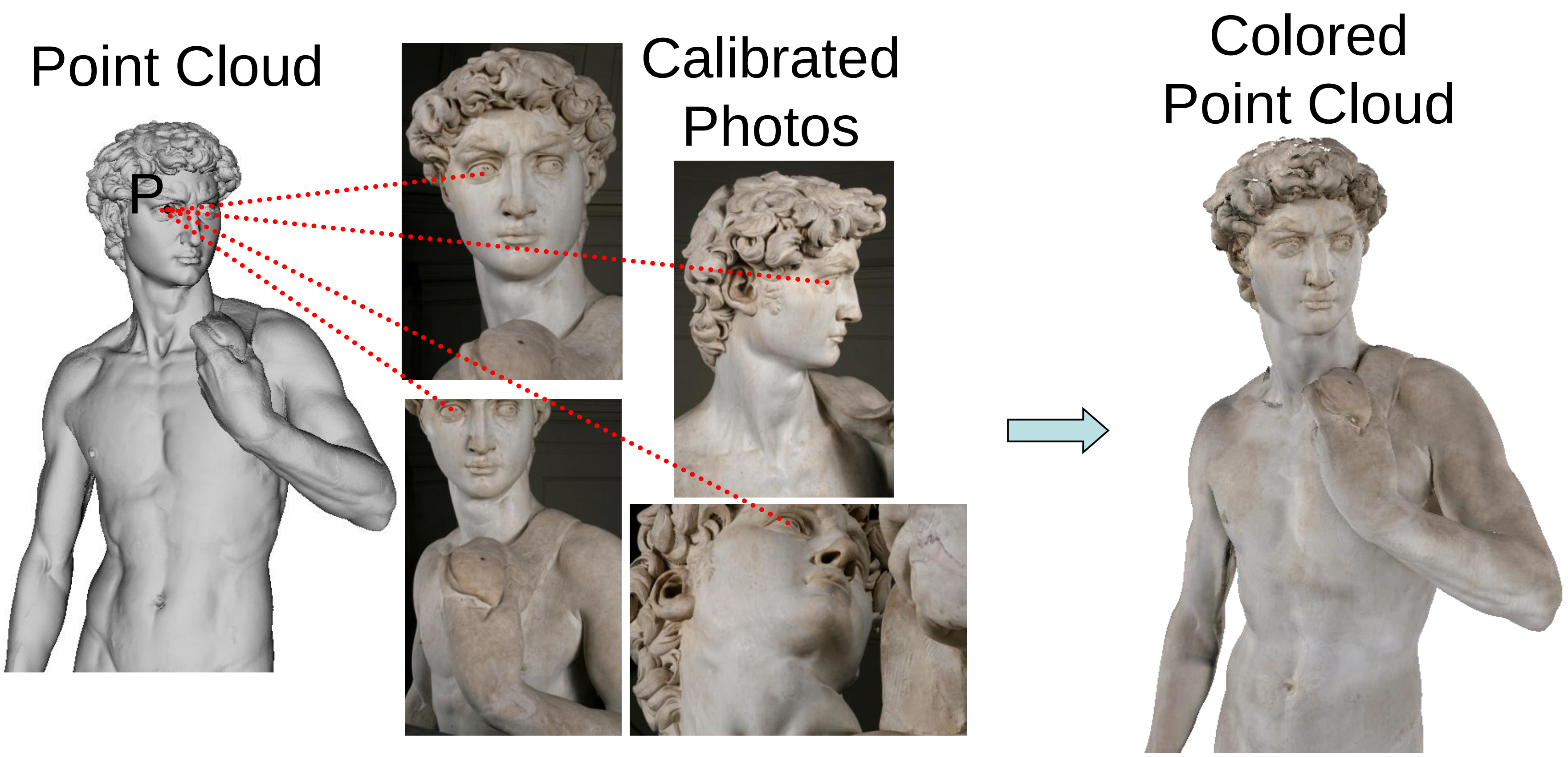
Calibrated Photos



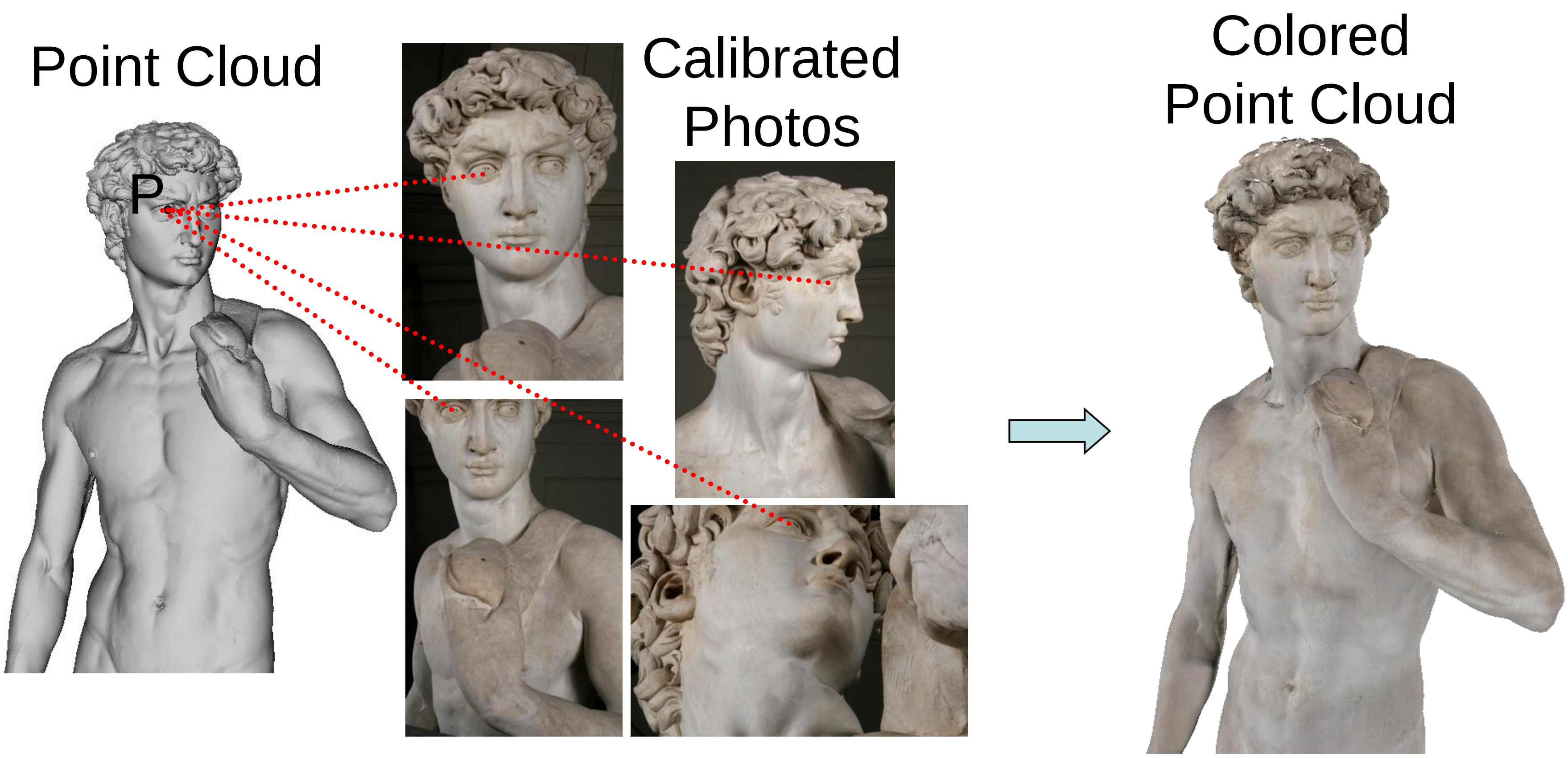
Problem Statement



Problem Statement



Problem Statement



- **Problem → Unlimited size of 3D model (Gpoints) and unlimited number of images**

Related work

- **State-of-the-art techniques**
  - Image quality estimation
  - Stitching or blending
- **Data representation**
  - Triangle meshes – exploit connectivity
  - Meshless approaches
    - Both triangle meshes and point clouds
- **Memory settings**
  - All in-core – no massive geometry/images
  - 3D in-core and images out-of-core – no massive geometry
  - All out-of-core – Low performances
- **Our contribution**
  - Blending function / Streaming framework / Massive point cloud / Adaptive geometry refinement

Pipeline

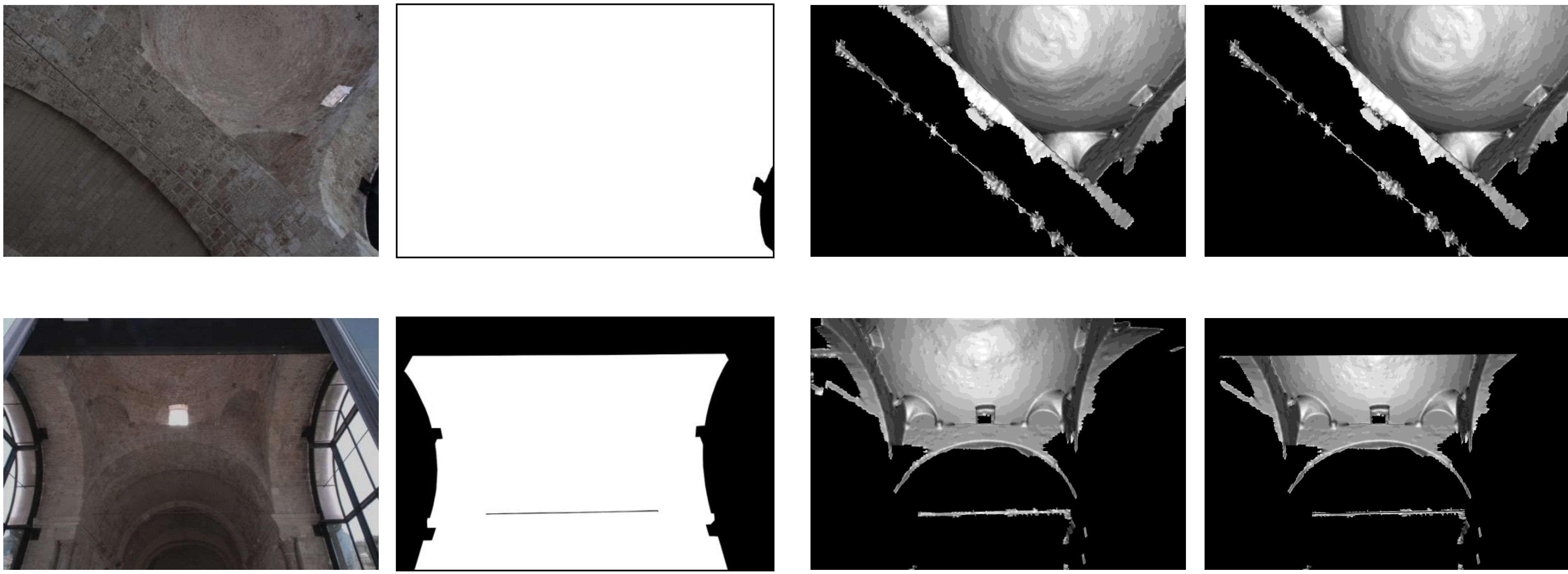
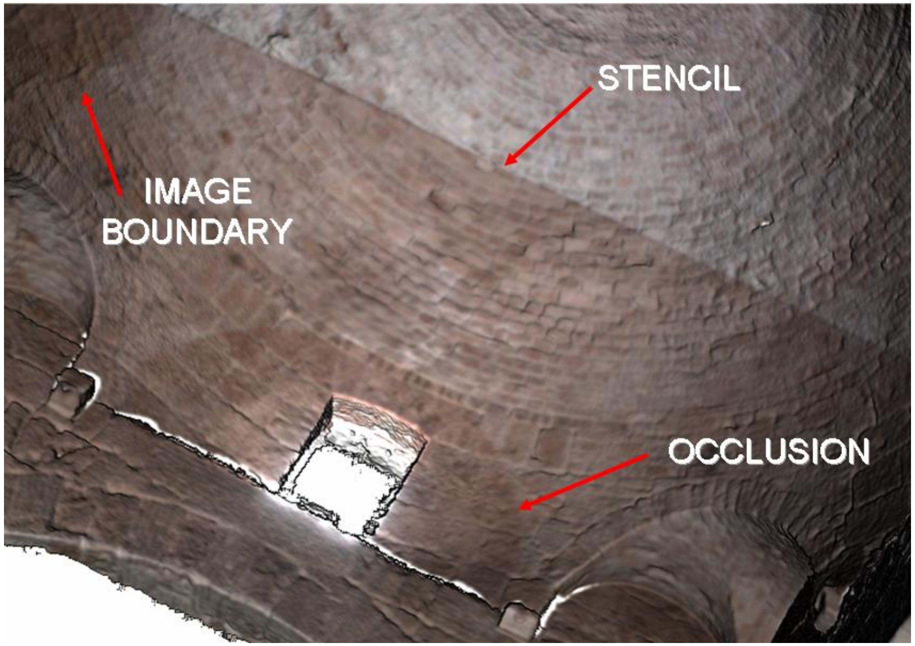
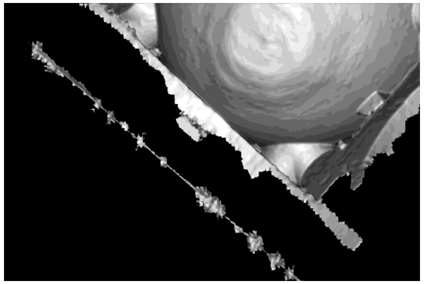


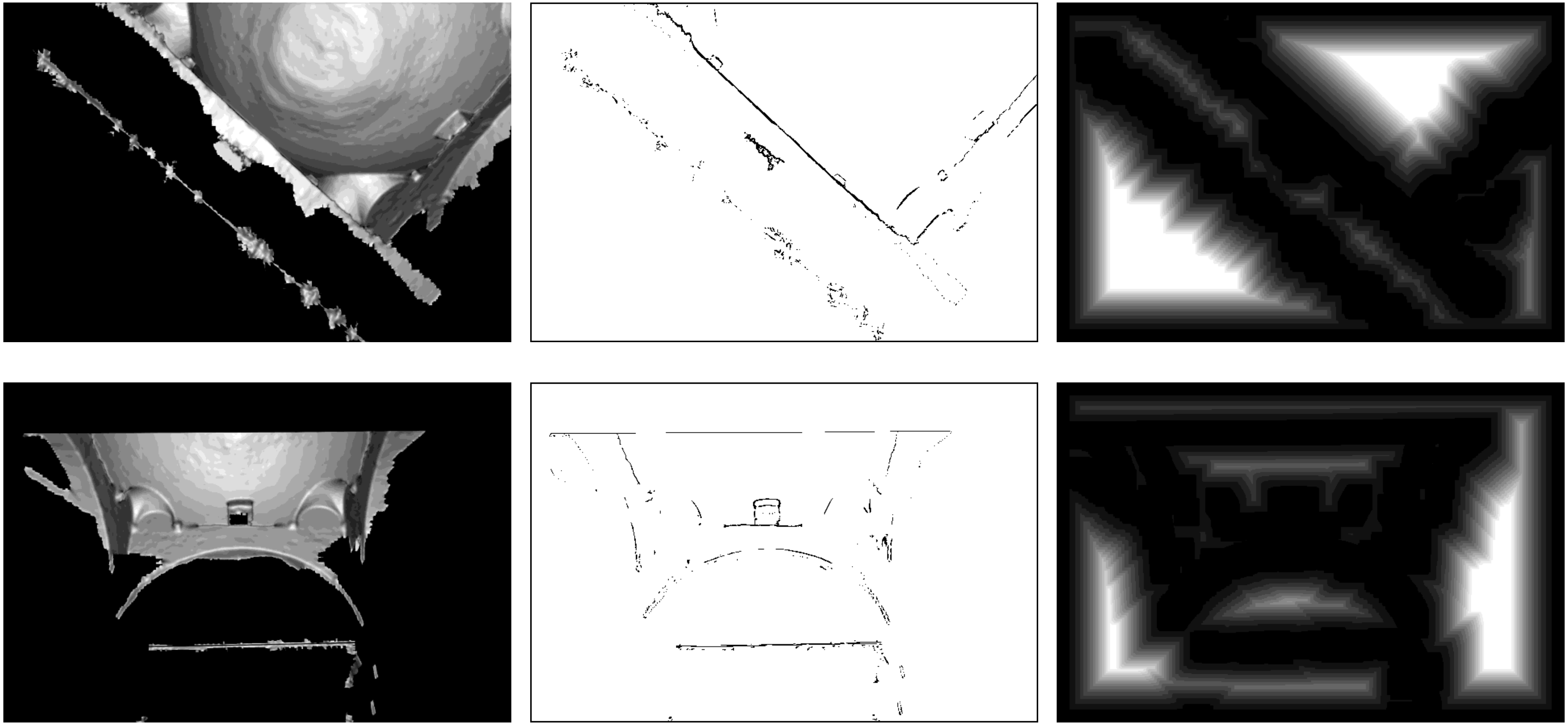
Photo | Stencil | Per-pixel Weight | Masked Per-pixel Weight

Simple blending

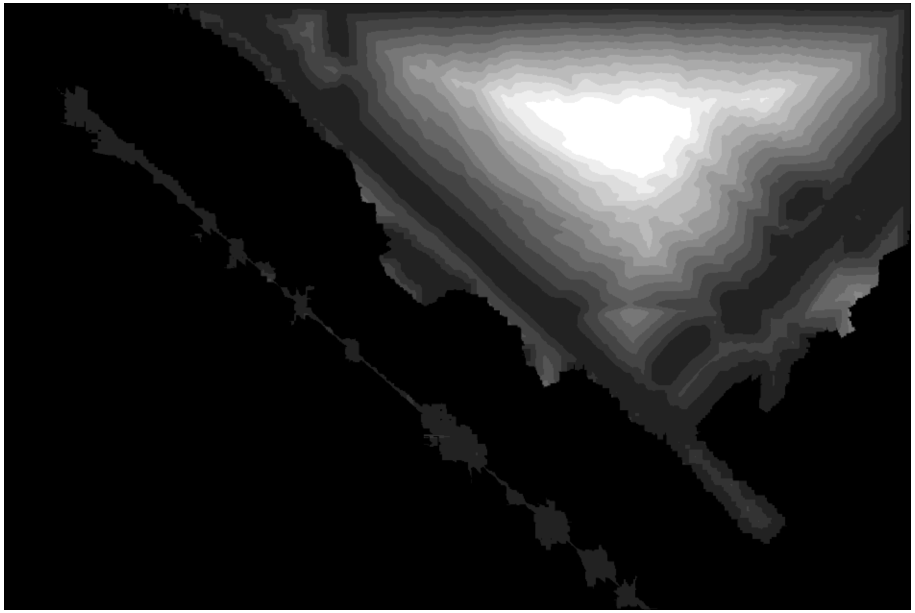


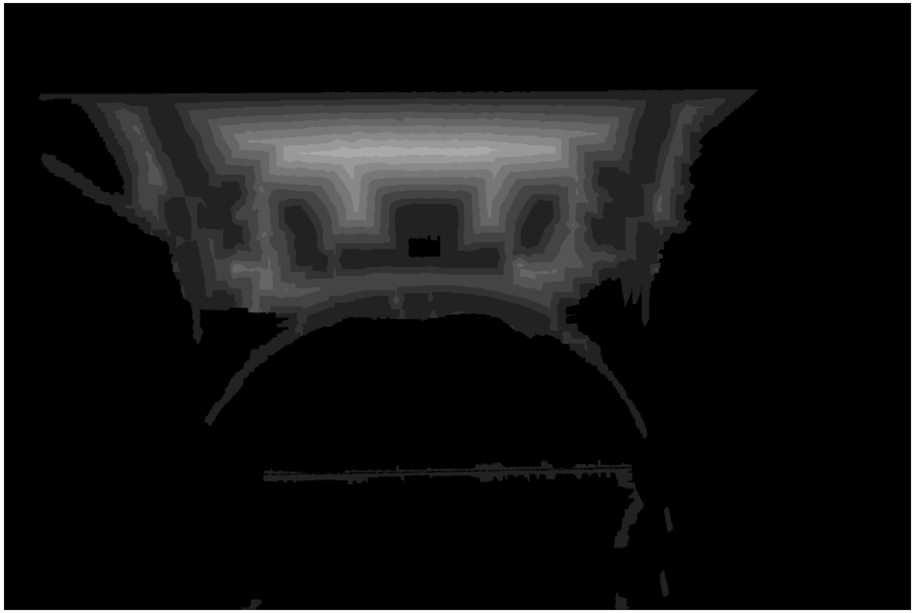


Edge extraction and Distance Transform

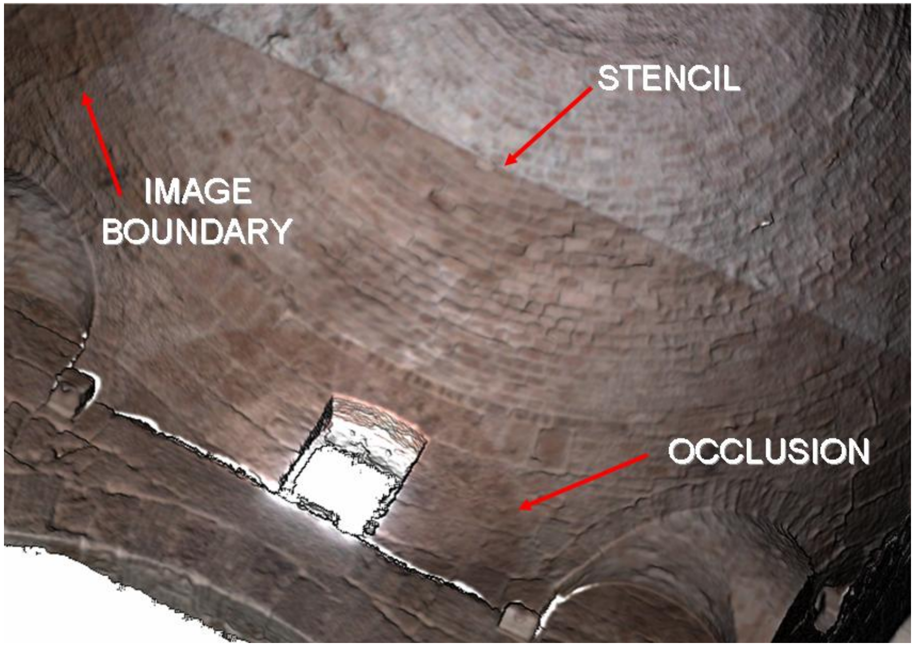


Smooth weight





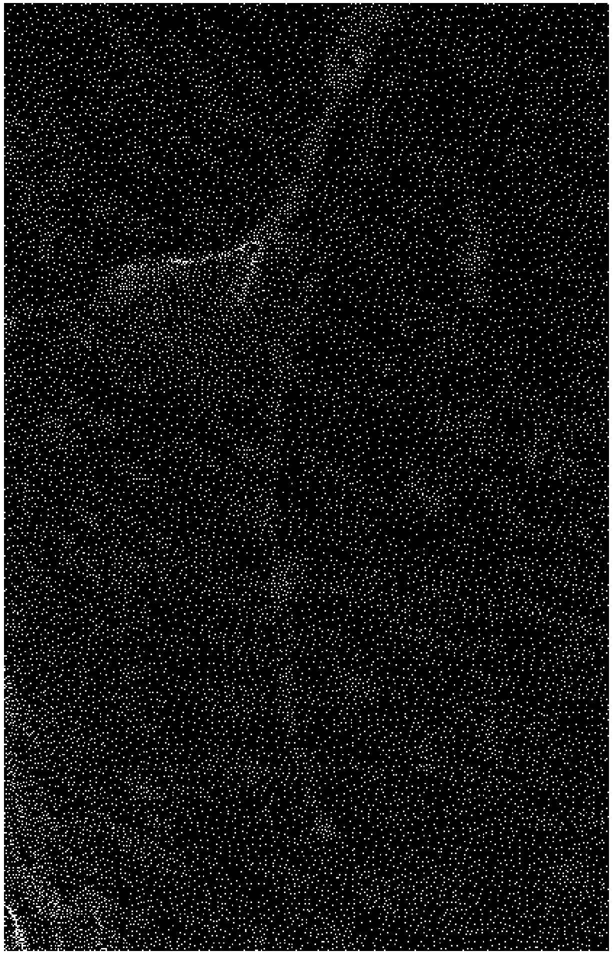
Smooth weight



Single band blending

Multi band blending

Adaptive point refinement



Adaptive point refinement

Adaptive point refinement

Adaptive point refinement

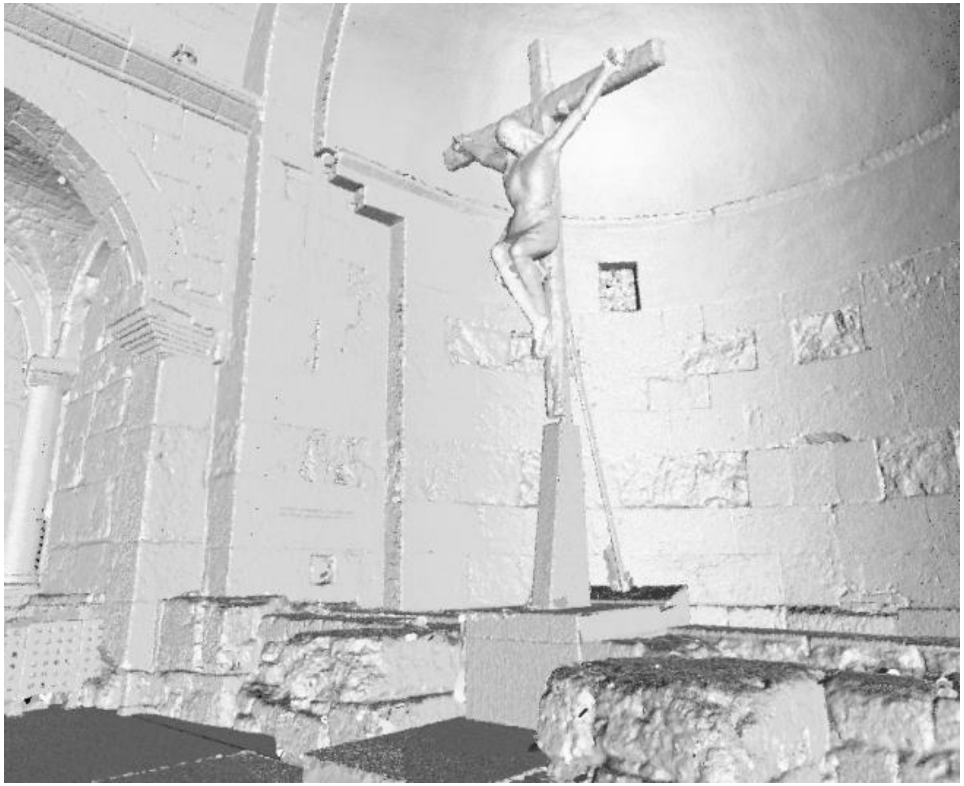
Results

Model	Resolution (# Points)	Images (width x height)	M pixels	No Morton	Morton	Tmp Disk Space
David	28M	67 (2336x3504)	548	5m21s	5m12s	960MB
Church	72M	162 (3872x2592)	1625	4h30m	47m30s	2.2GB
Arch. Site	133M	19 (3872x2592)	19	2h55m	1h43m	4GB
David	470M	67 (2336x3504)	548	21h6m	4h40m	14GB

David
470Mpoints

- **Callieri et. al 2008 – David 28M**
  - Disk space occupancy – 6.2GB
  - Computation time – 15.5 hours

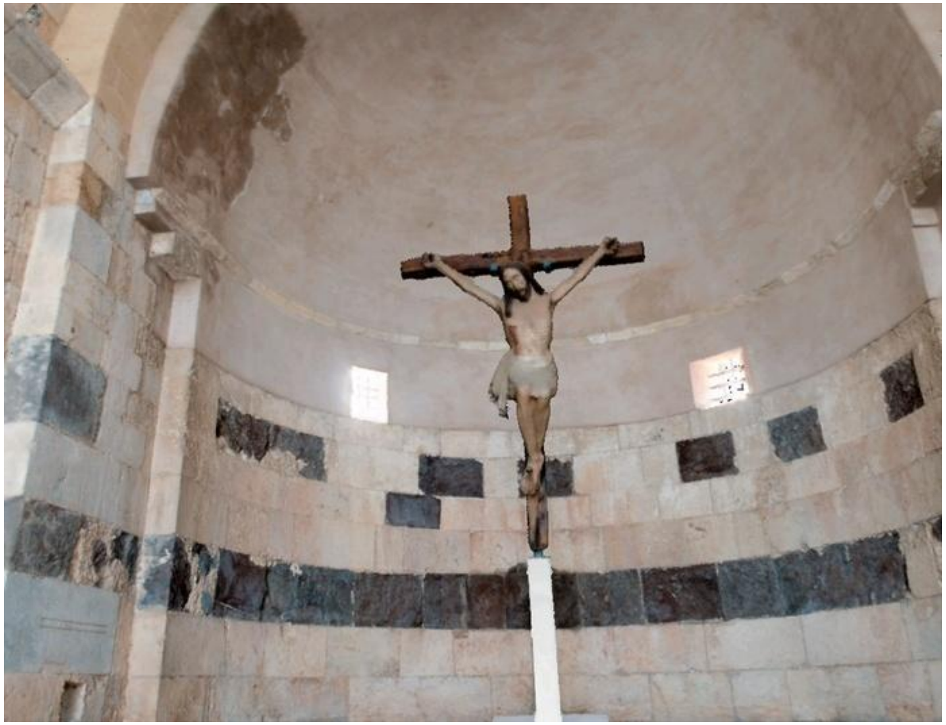
Results – Church's Apse



14 Mpoint Geometry

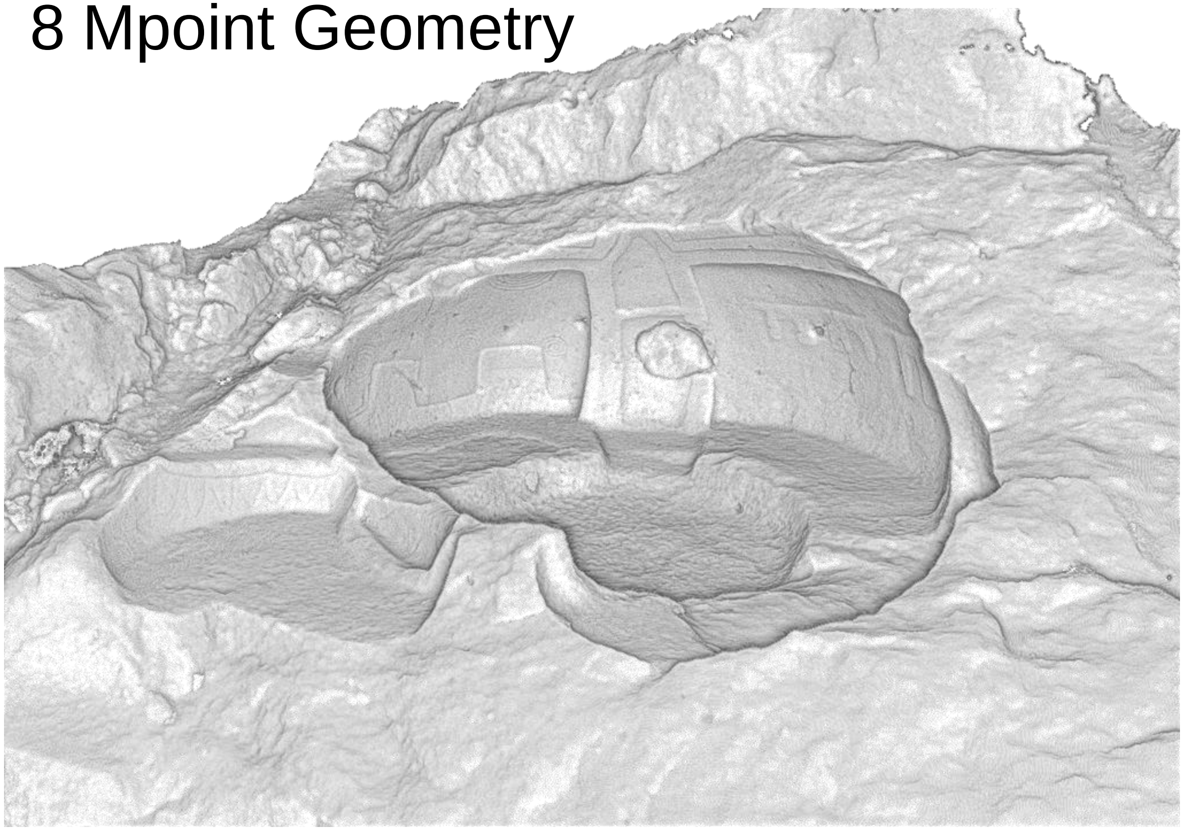
40 photos

Results – Church's Apse



Results – Grave

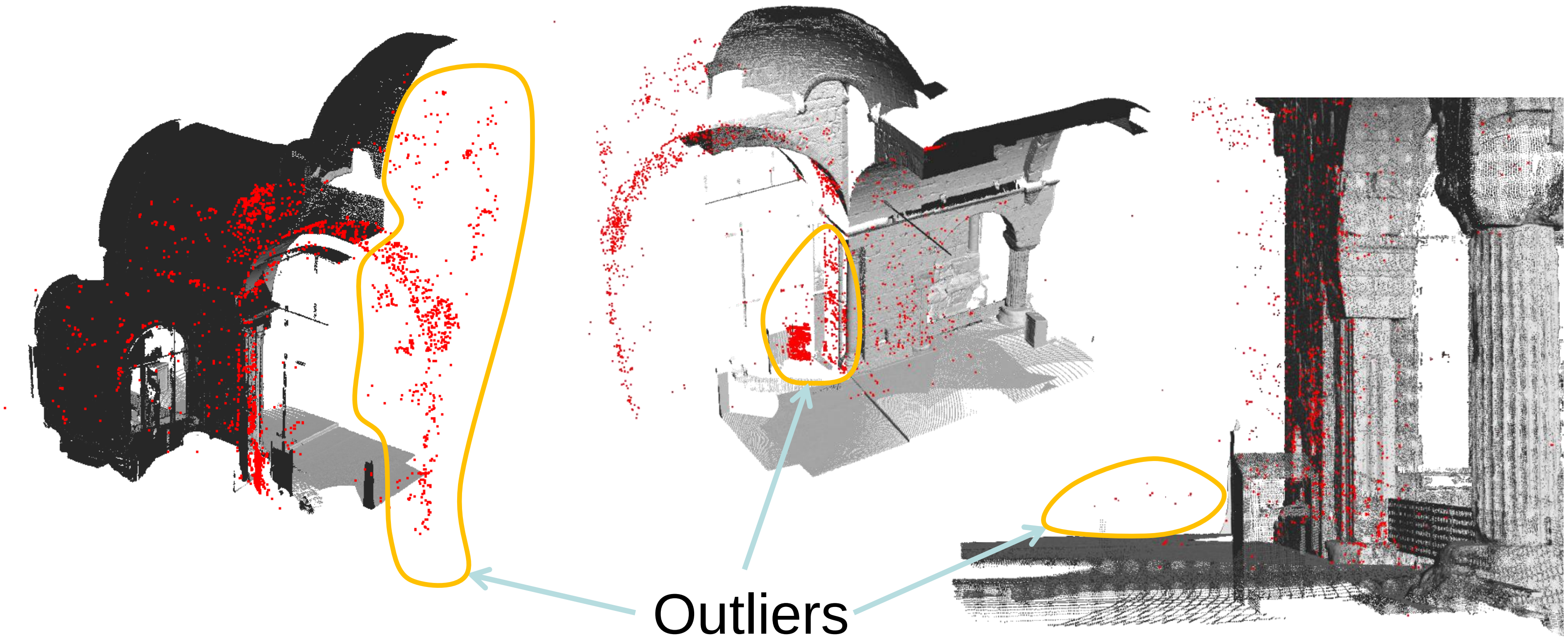
8 Mpoint Geometry



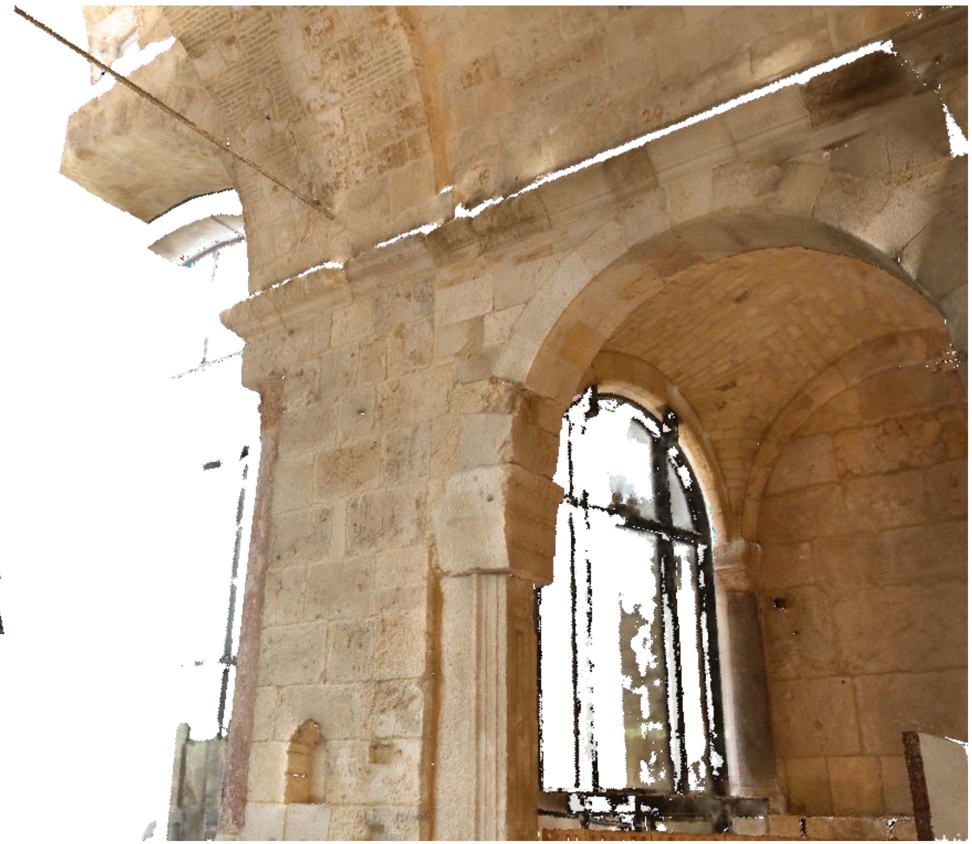
21 photos

Results – Grave

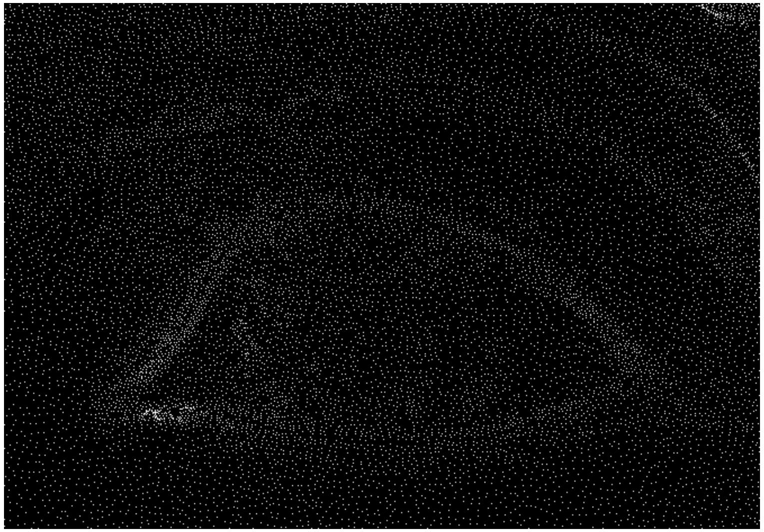
Results – Church's detail



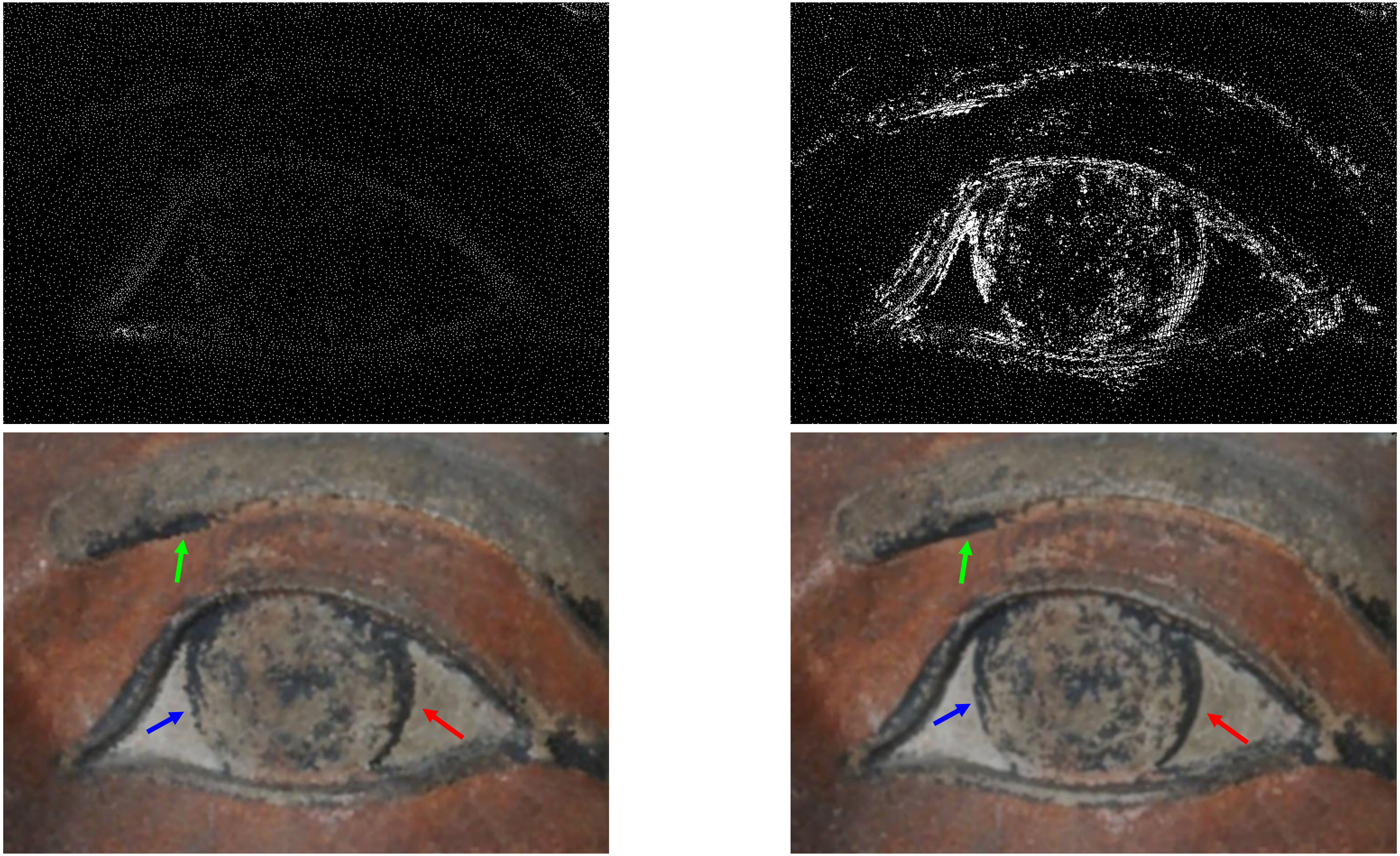
Results – Church’s detail



Results

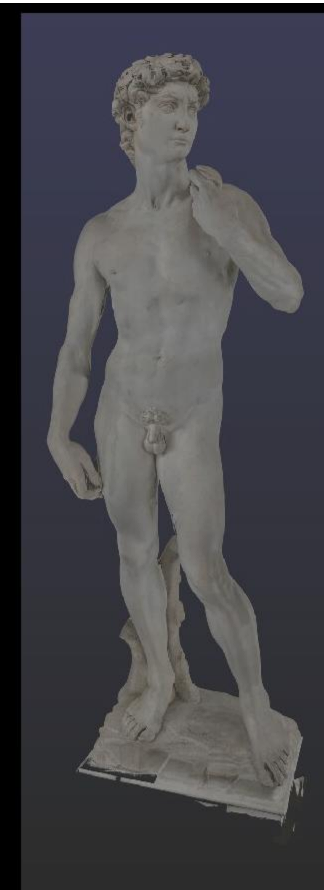


Results



Results

David
470Mpoints
Image size – 19456x53248
1Gpixel



Source data by: The Digital Michelangelo Project.
Alignment, Coloring, rendering and gigapixel image generation by CRS4 Visual Computing and ISTI-CNR Visual Computing
Image size: 19456x53248 - 1Gpixel

Results


Source data by: The Digital Michelangelo Project.
Alignment, Coloring, rendering and gigapixel image generation by
CRS4 Visual Computing and ISTI-CNR Visual Computing
Image size: 19456x53248 - 1Gpixel

Conclusion

- **Point-based rendering pipeline**

- **Image-to-geometry registration approach**
- **Minimum user intervention**
- **No constraints on geometry, attributes and features**
- **Specific robust cost function and SBA**

- **Out-of-core photo blending approach (Point clouds of unlimited size)**
- **Incremental color accumulation (Unlimited number of images)**
- **Smooth weight function (Seamless color blending)**
- **Streaming framework (Performance improvement)**
- **Adaptive point refinement**

- **Future work**
  - Automatic sparse-to-dense geometry registration
  - Interactive blending - adding and removing images in an interactive tool
  - Fast visual check of previous alignment step

Conclusion

- **Low cost**
  - Personal computer
  - Digital camera
  - Decreased manual intervention
- **Open Source / Free Software**
  - Bundler – SfM reconstruction – http://phototour.cs.washington.edu/bundler/
  - Sparse Bundle Adjustment – SBA – Minimization – http://www.ics.forth.gr/~lourakis/sba/
  - Opengl / GLSL shaders – Rendering – http://www.opengl.org/
  - Qt – Interface – http://qt.nokia.com/
  - Opencv – Manual registration – http://opencv.willowgarage.com/wiki/
  - Spaceland Library – Geometric computation – http://spacelib.sourceforge.net/
  - IIPImage – Web-based Viewer – http://iipimage.sourceforge.net/

Questions & Contacts

- **CRS4 – VIC**
  www.crs4.it/vic/

- **Ruggero Pintus**
  ruggero@crs4.it