INRIA
RHÔNE-ALPES

CONTRIBUTIONS TO SPATIAL AND TEMPORAL 3D RECONSTRUCTION FROM MULTIPLE CAMERAS

by Andrei Zaharescu

Supervisors: Radu Horaud & Edmond Boyer
Perception Team

3D SPATIAL RECONSTRUCTION

Input

Output

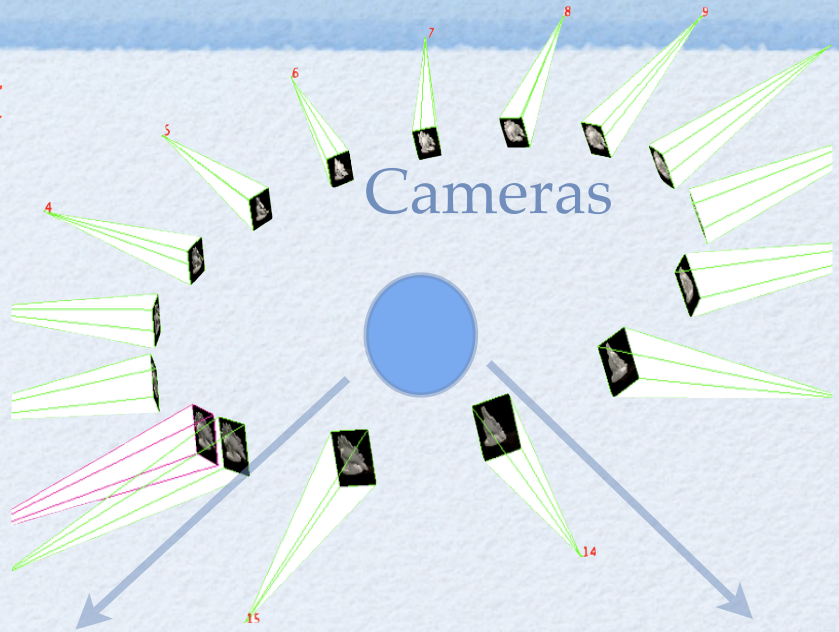
Image Source: © Middleburry University

3D SPATIAL RECONSTRUCTION

Input

Image Source: © Middleburry University

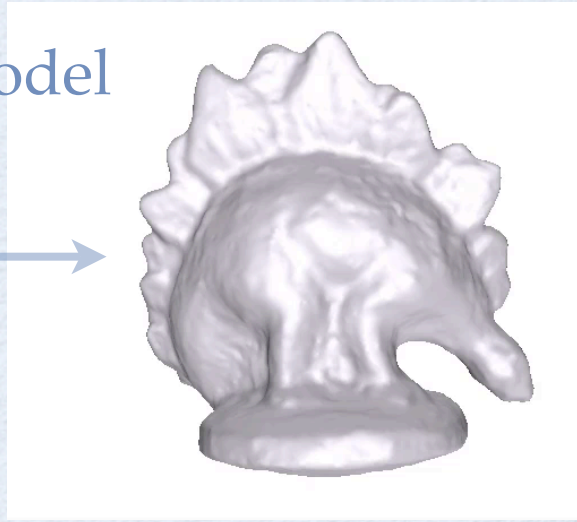
Output



Sparse

[Zaharescu & al - IJCV 2009]

Model



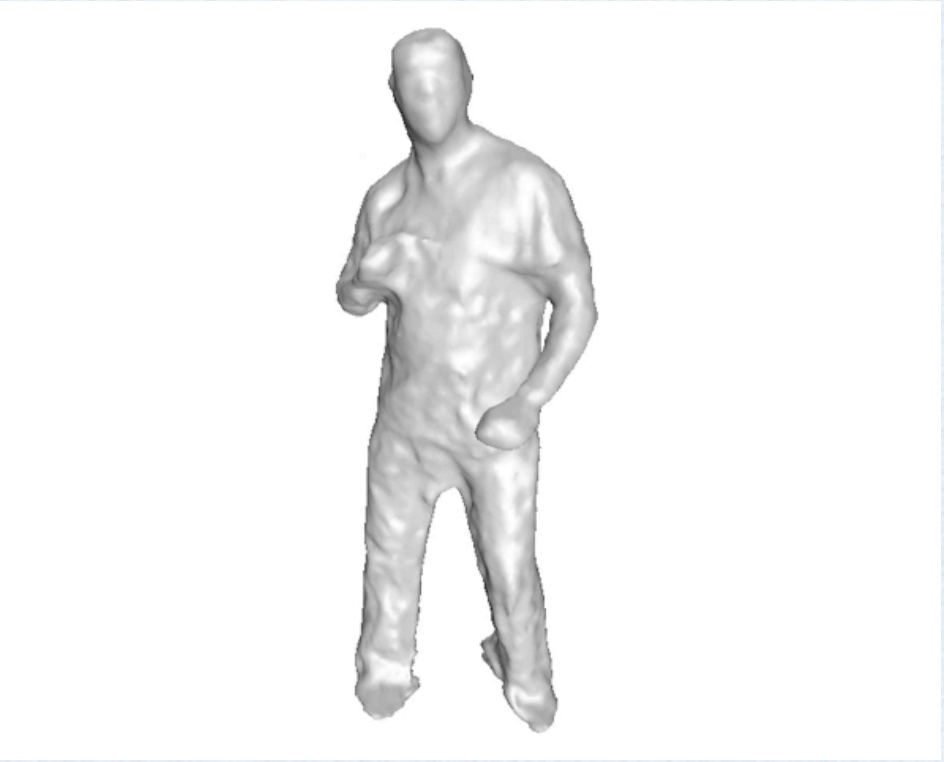
Dense

[Zaharescu & al - ACCV 2007]

3D TEMPORAL RECONSTRUCTION

Input

Output



reconstructions
across time

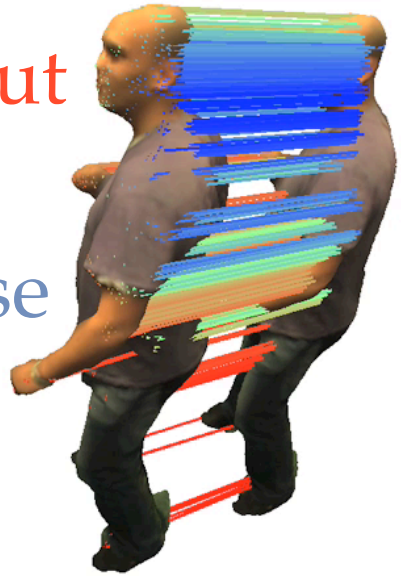
3D TEMPORAL RECONSTRUCTION

Input

reconstructions
across time

Output

sparse

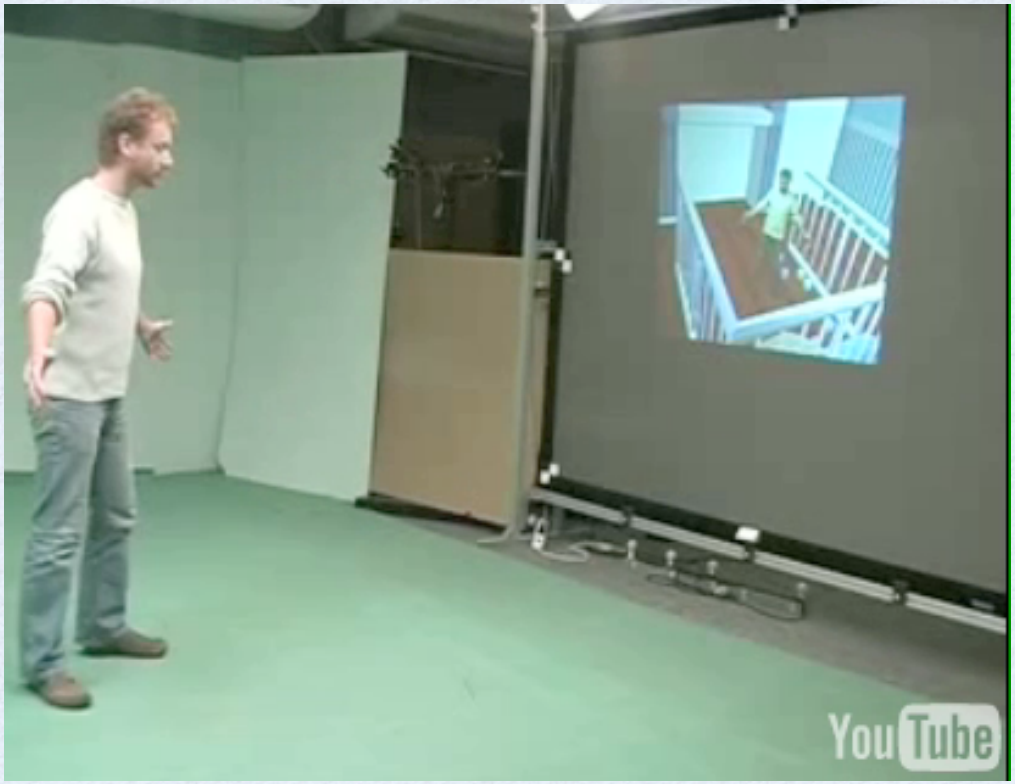


[Zaharescu & al - CVPR 2009 -submitted]

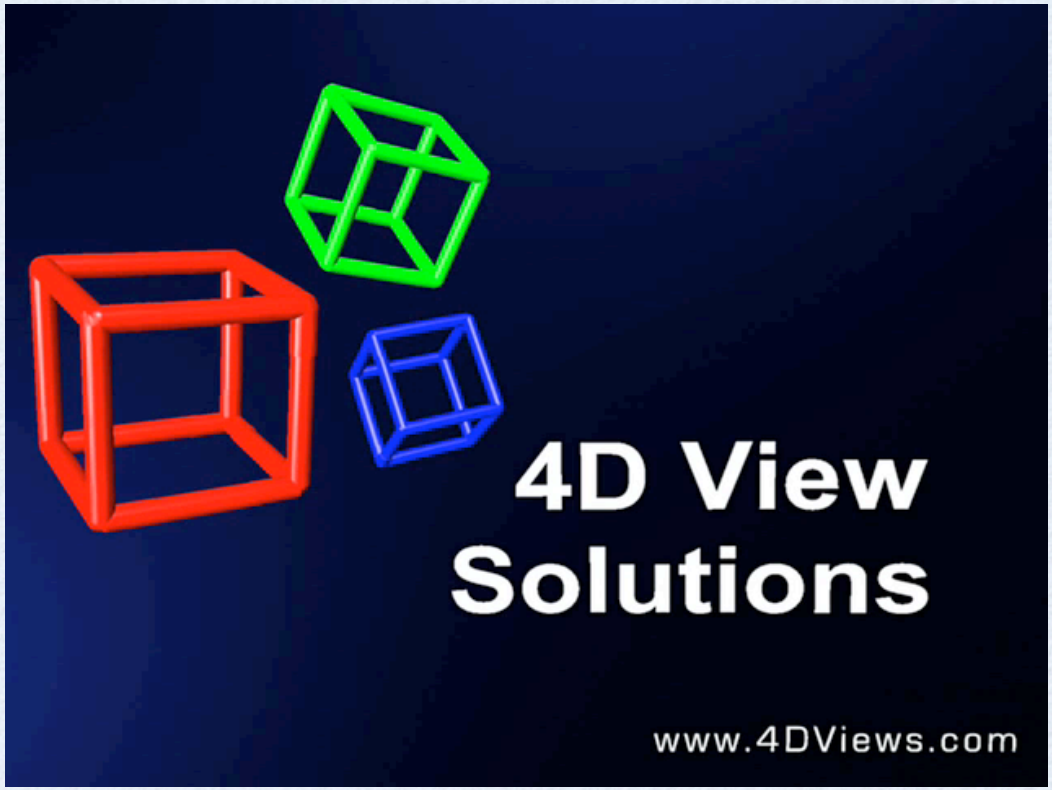
dense

[Varanasi, Zaharescu & al - ECCV 2008]

Applications



Source: © 4D View Solutions



Source: © 4D View Solutions

- Virtual Reality and Human Computer Interactions (immersive environments)
- TV & entertainment (free viewpoint rendering)

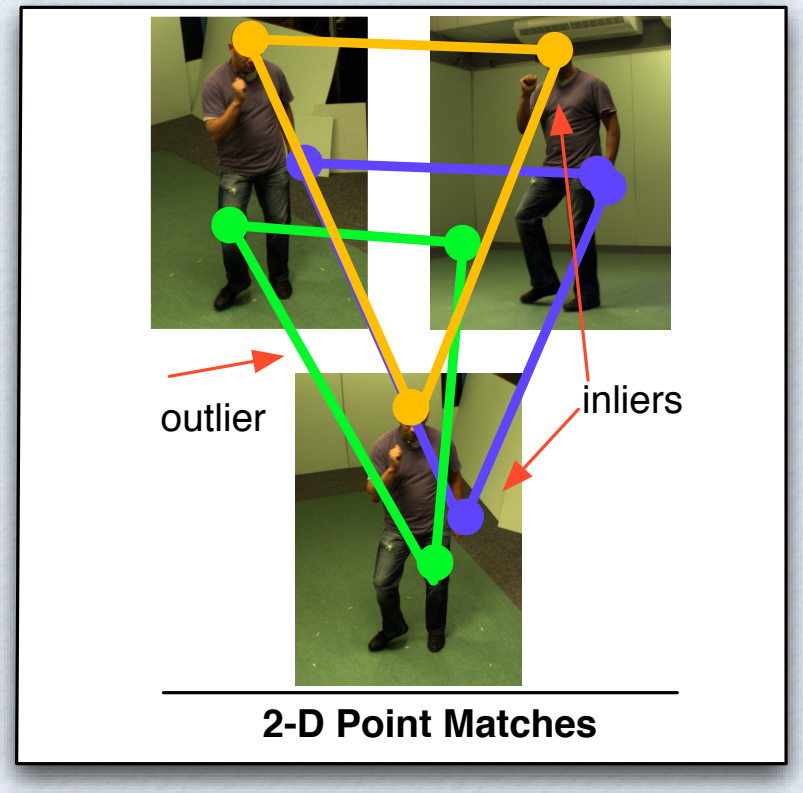
CONTRIBUTIONS

Input

Output

CONTRIBUTIONS

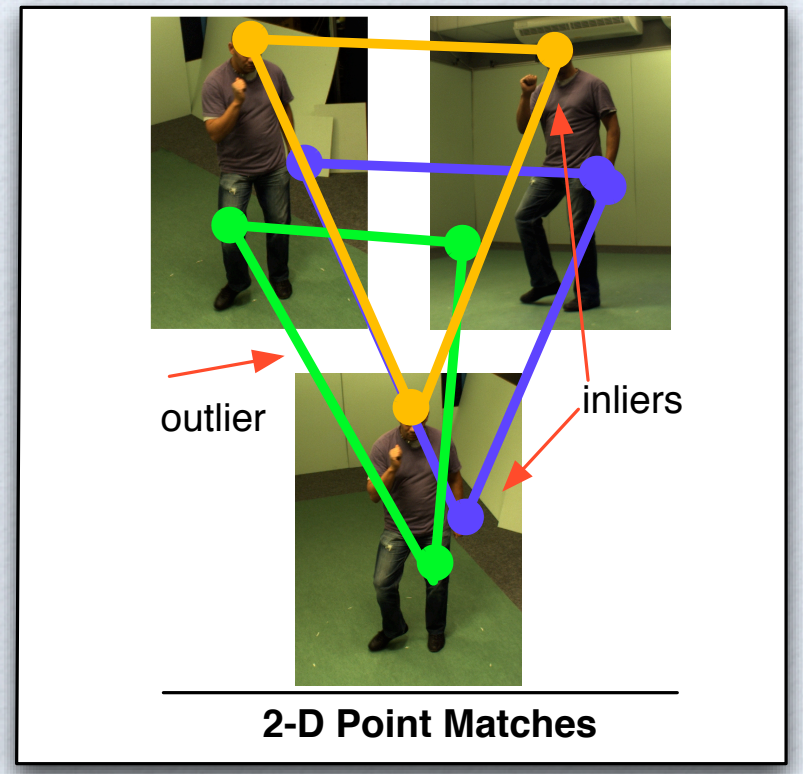
Input



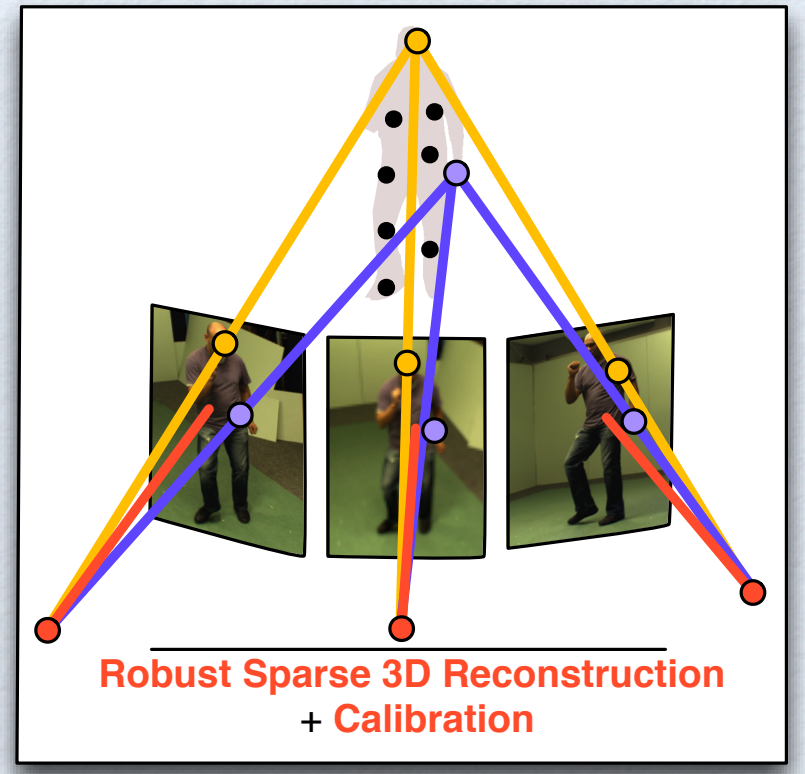
Output

CONTRIBUTIONS

Input

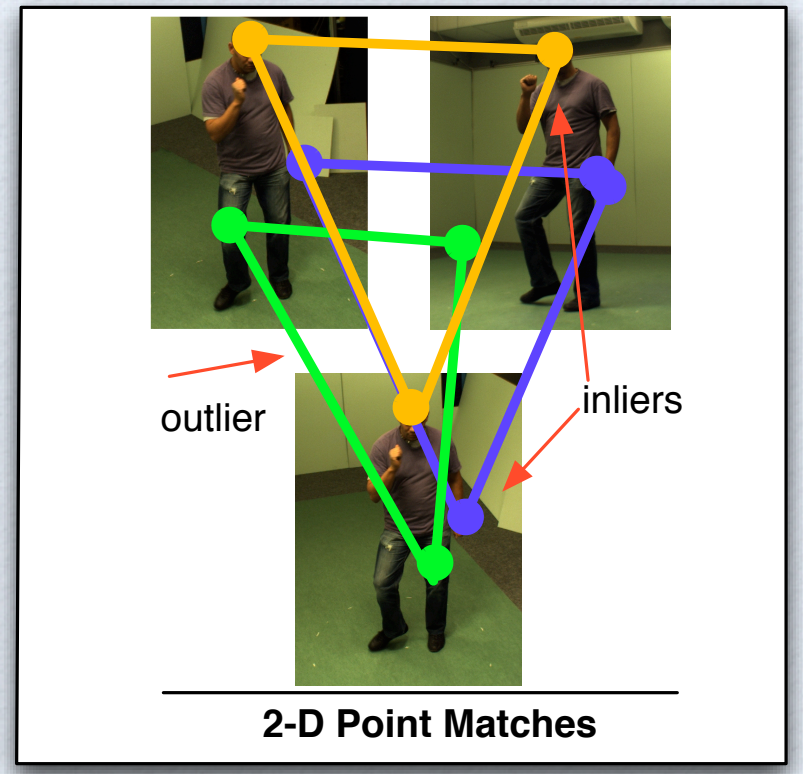


Output

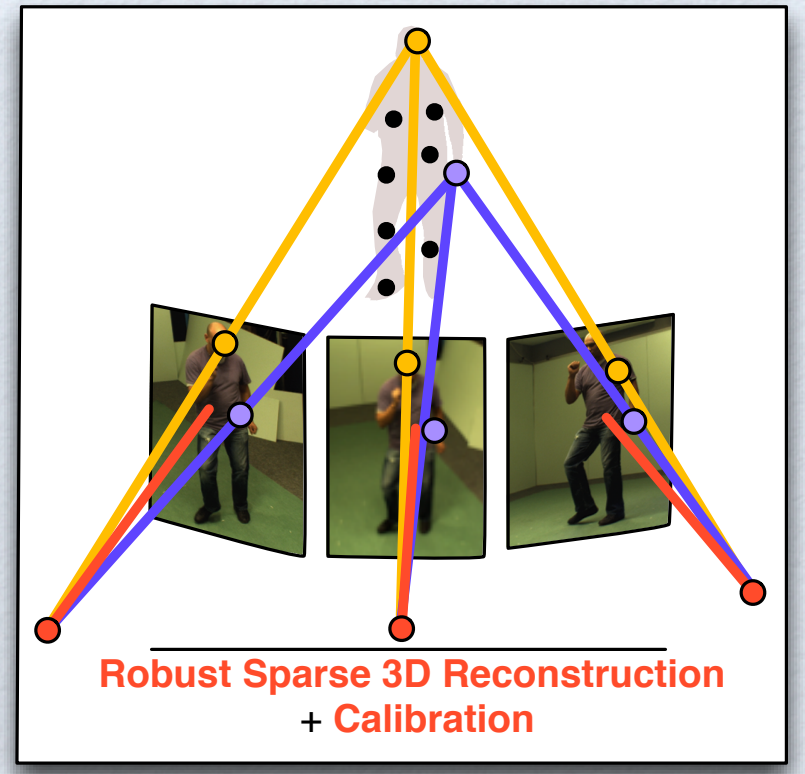


CONTRIBUTIONS

Input



Output



1. A Robust Probabilistic Factorization Framework

CONTRIBUTIONS

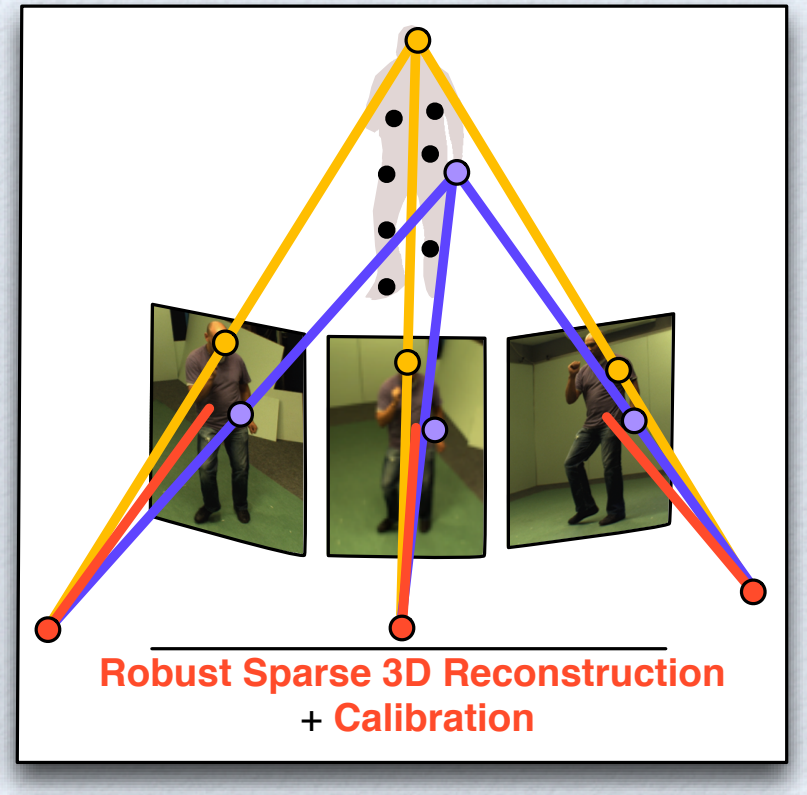
Input

Output

CONTRIBUTIONS

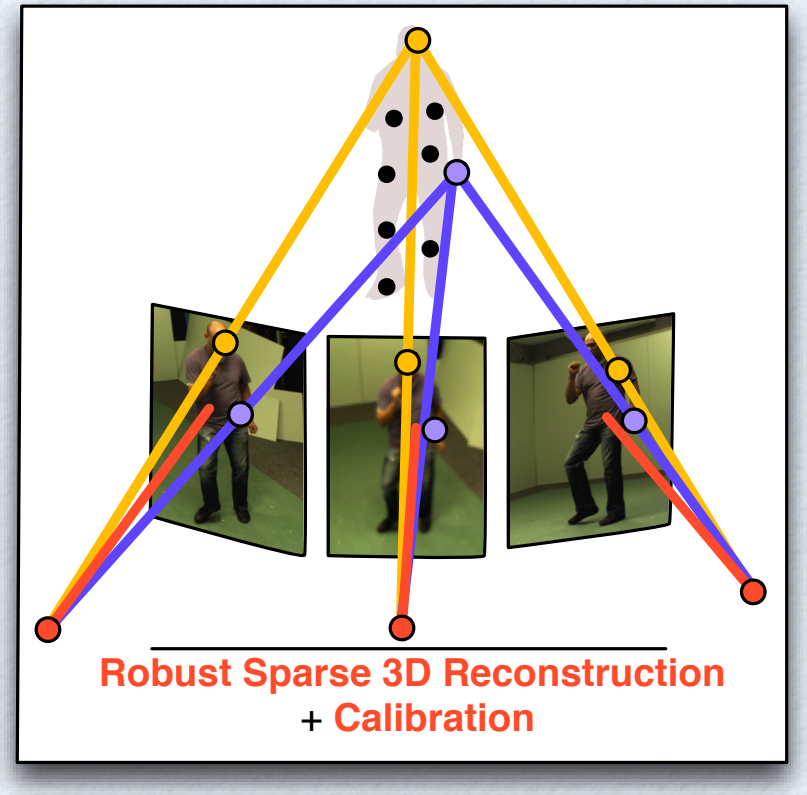
Input

Output

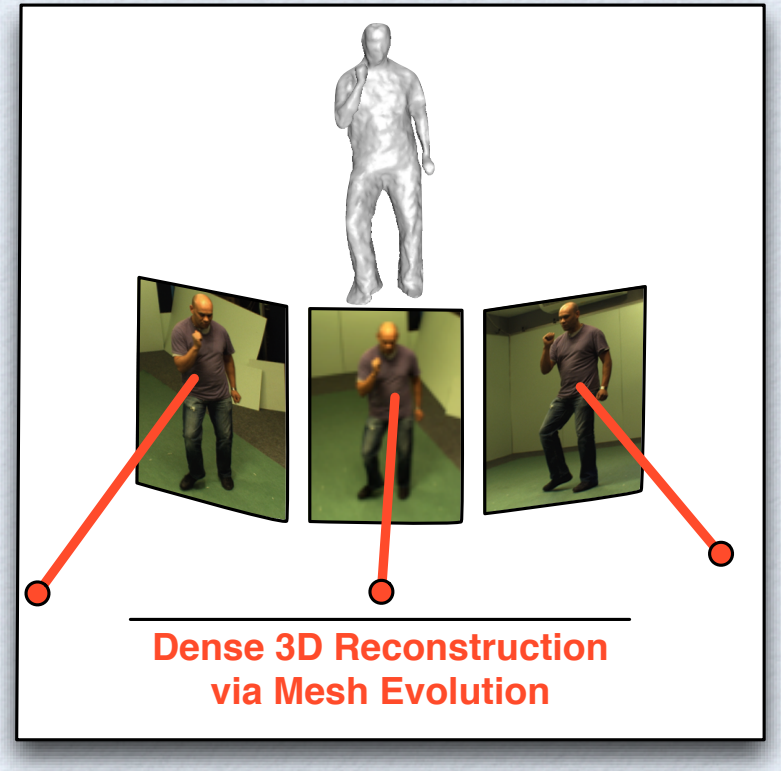


CONTRIBUTIONS

Input



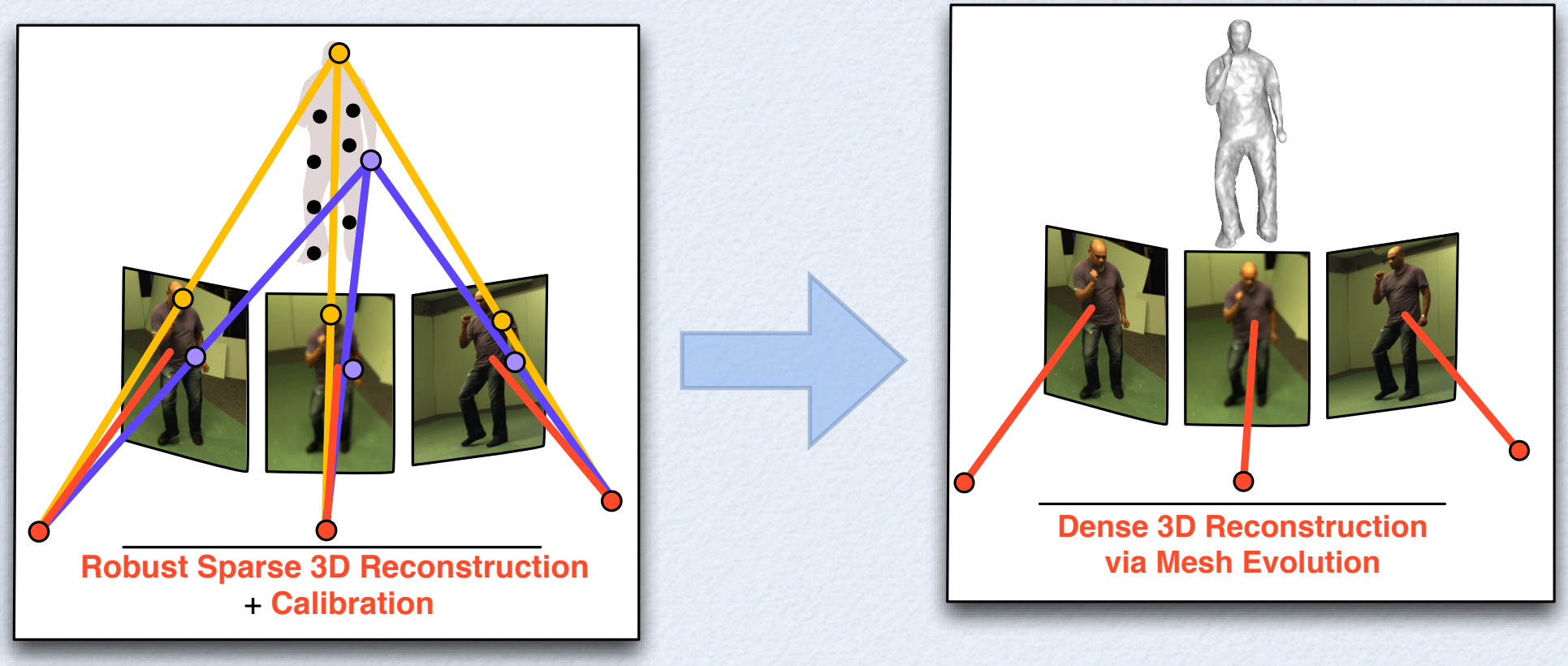
Output



CONTRIBUTIONS

Input

Output



2. A Mesh Surface Evolution Algorithm

3. Content-Aware Camera Clustering

CONTRIBUTIONS

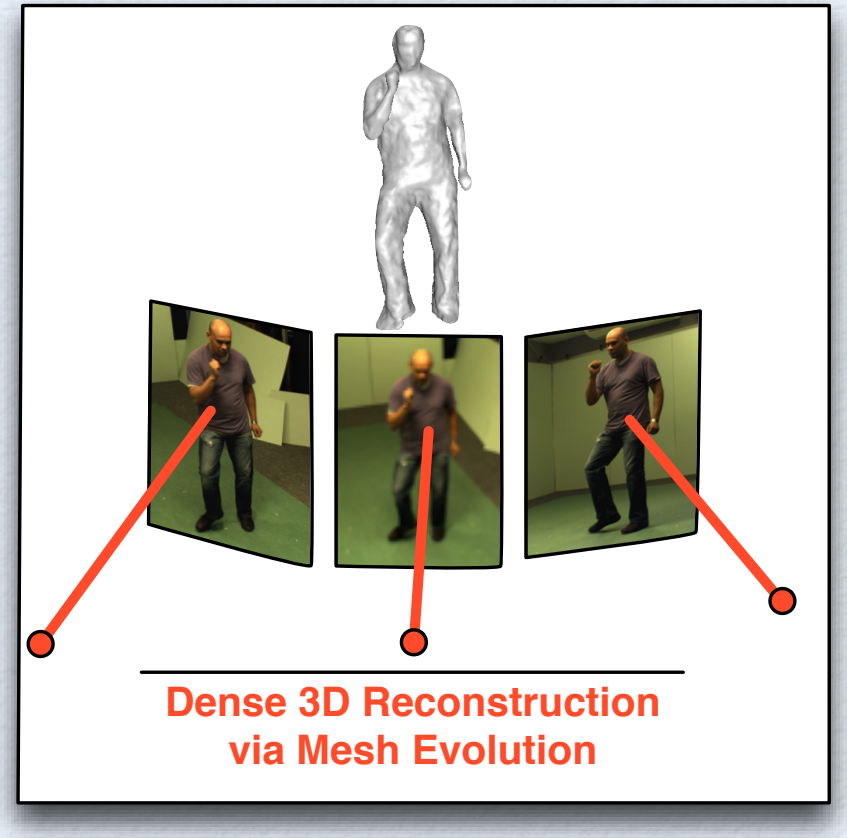
Input

Output

CONTRIBUTIONS

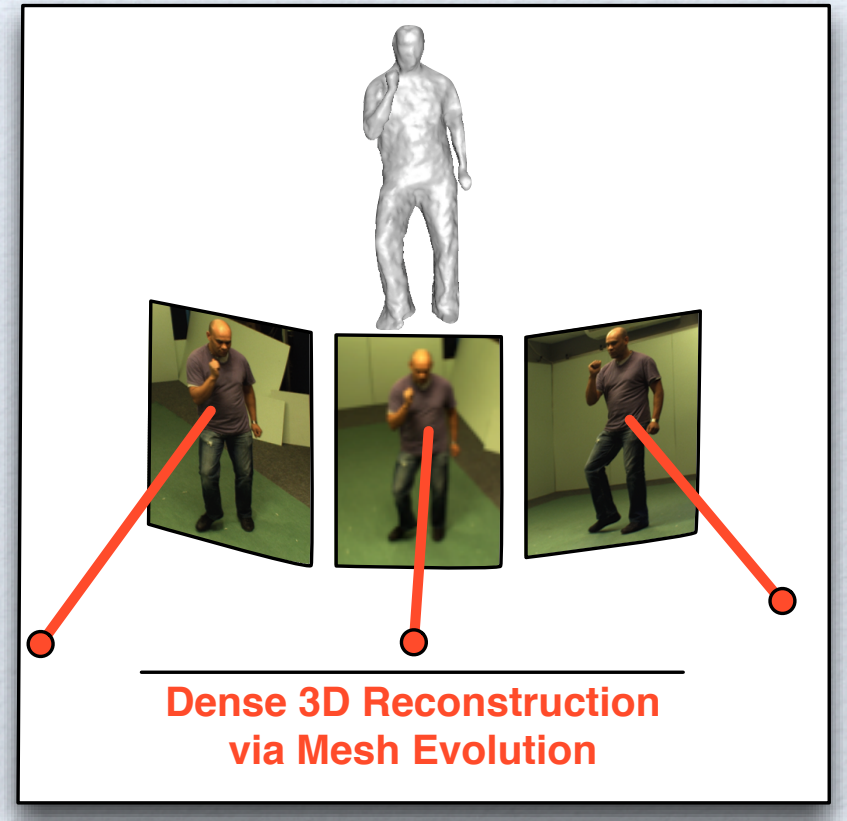
Input

Output



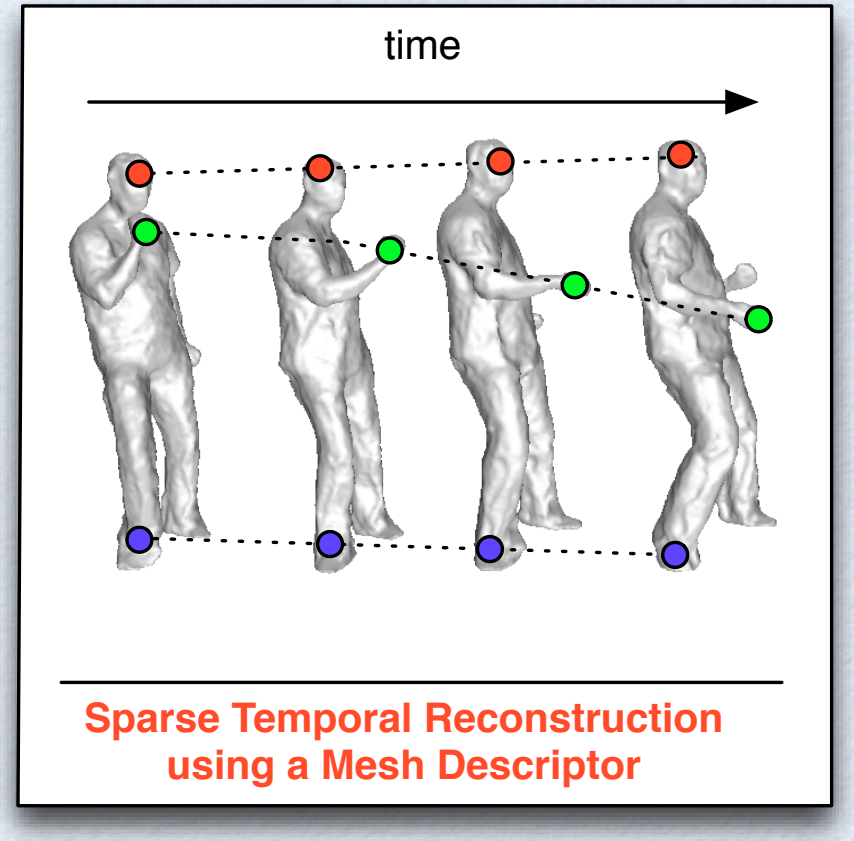
CONTRIBUTIONS

Input



Dense 3D Reconstruction
via Mesh Evolution

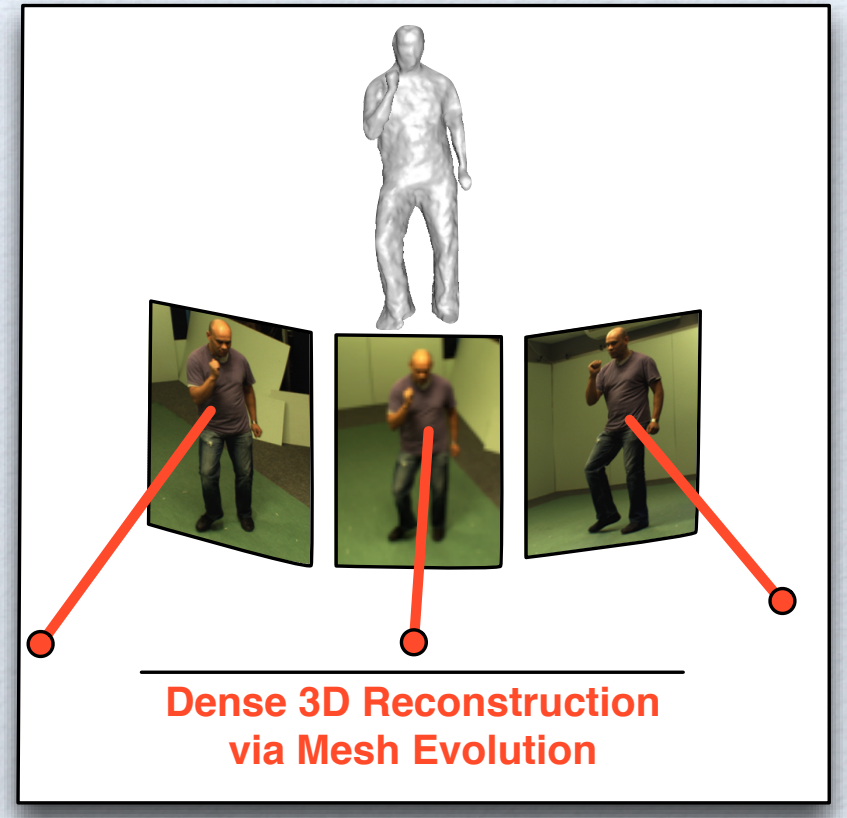
Output



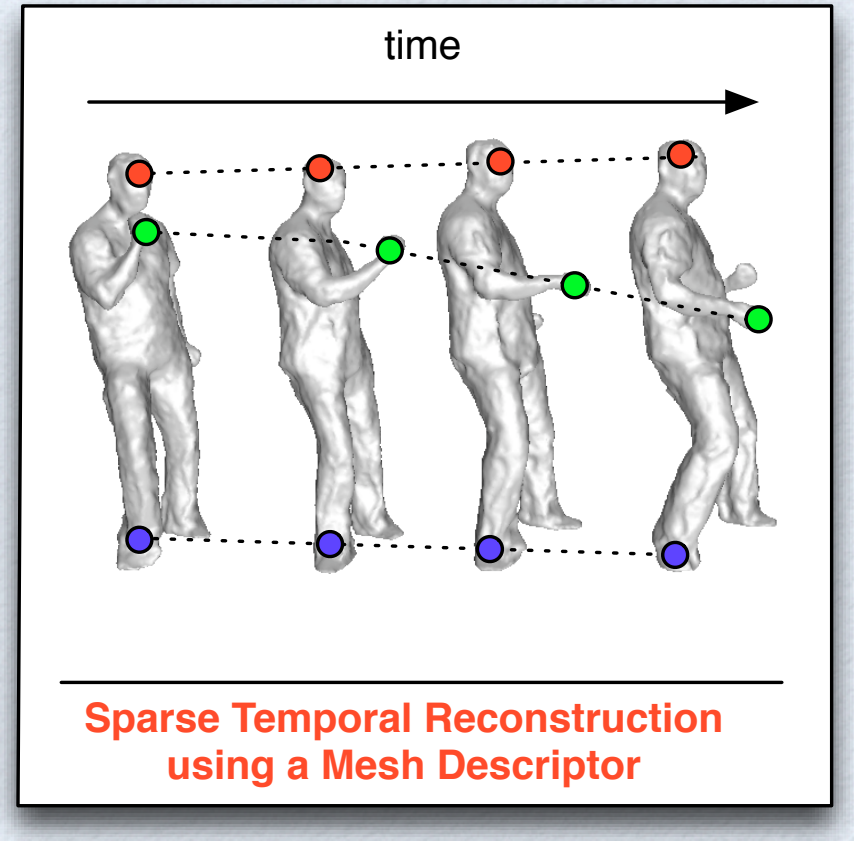
Sparse Temporal Reconstruction
using a Mesh Descriptor

CONTRIBUTIONS

Input



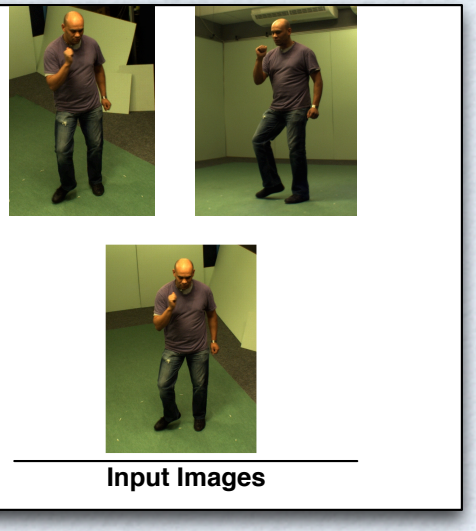
Output



4. 3-D Mesh Detector & Descriptor

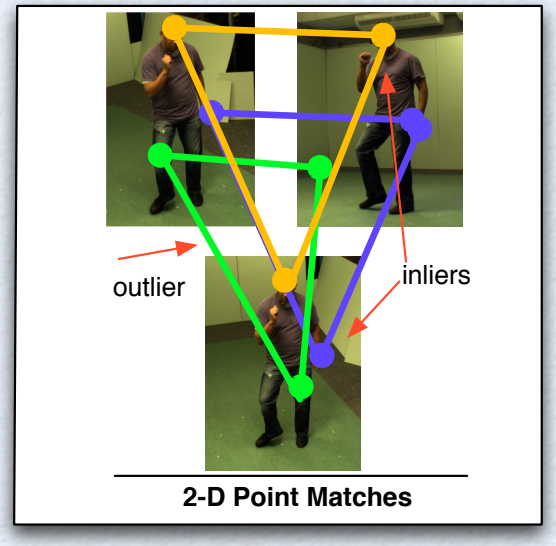
RECONSTRUCTION FRAMEWORK

RECONSTRUCTION FRAMEWORK

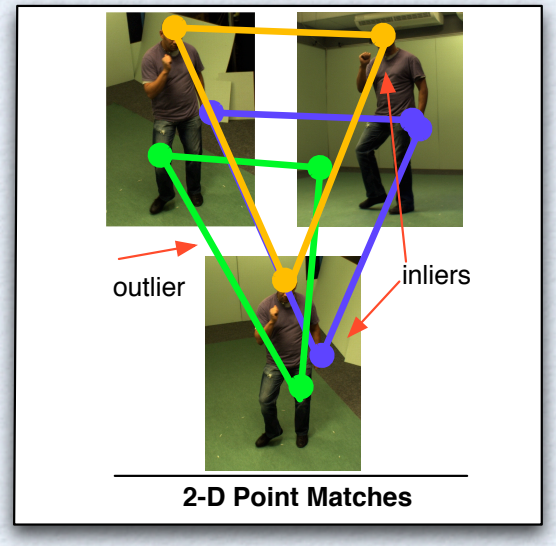


Input Images

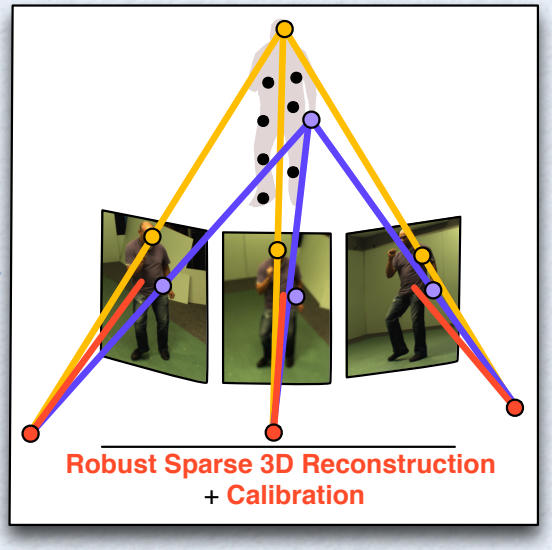
RECONSTRUCTION FRAMEWORK



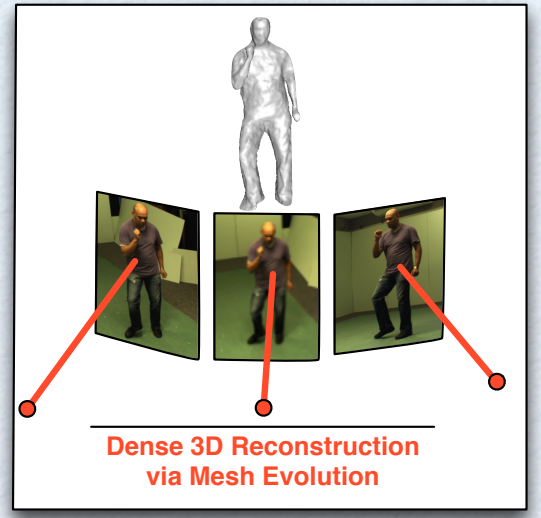
RECONSTRUCTION FRAMEWORK



C1

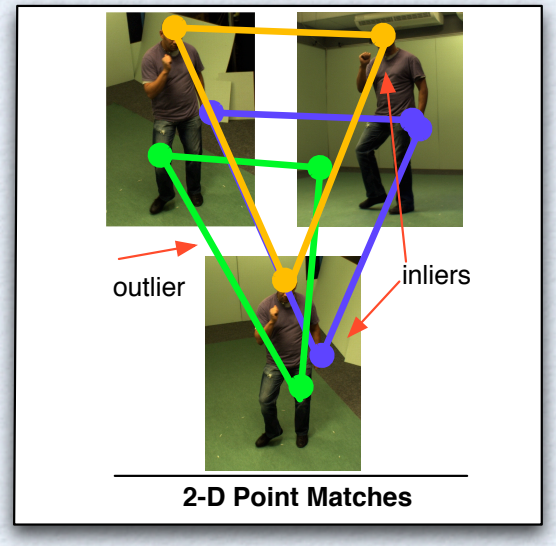


RECONSTRUCTION FRAMEWORK

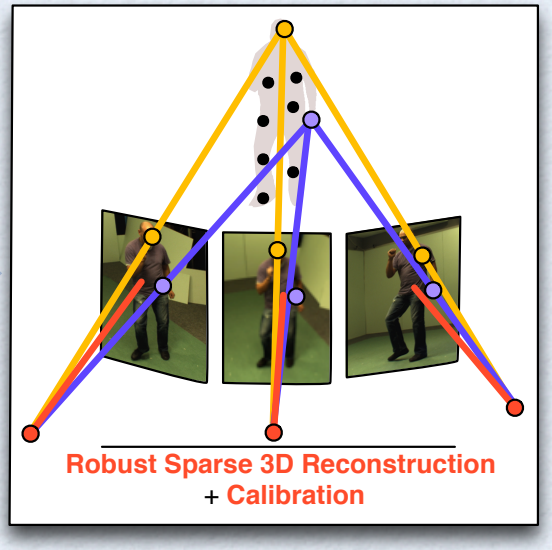


C2

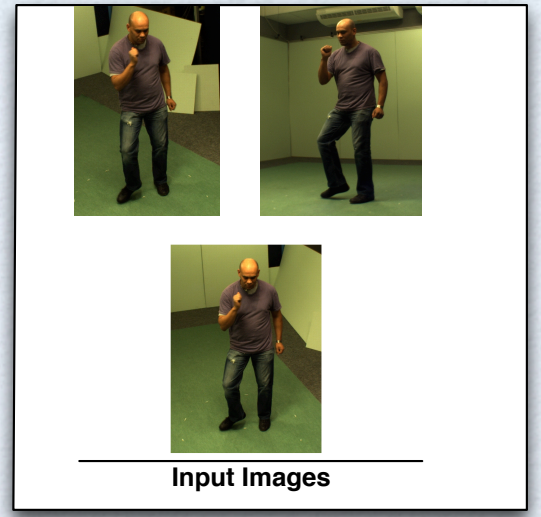
C3

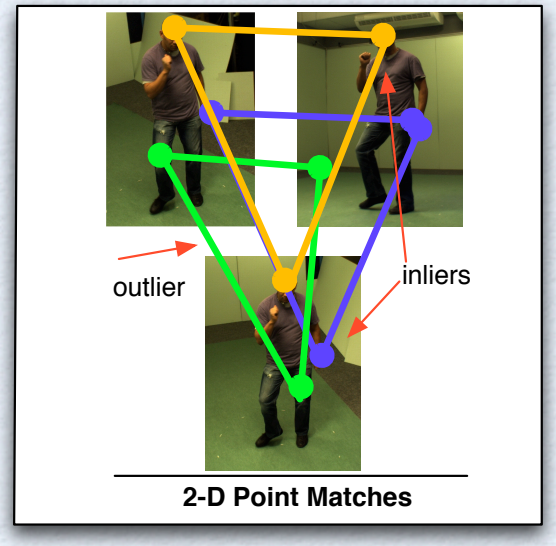


C1

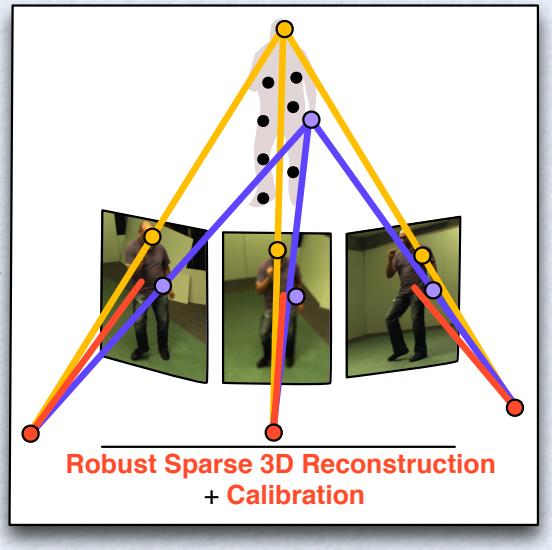


RECONSTRUCTION FRAMEWORK

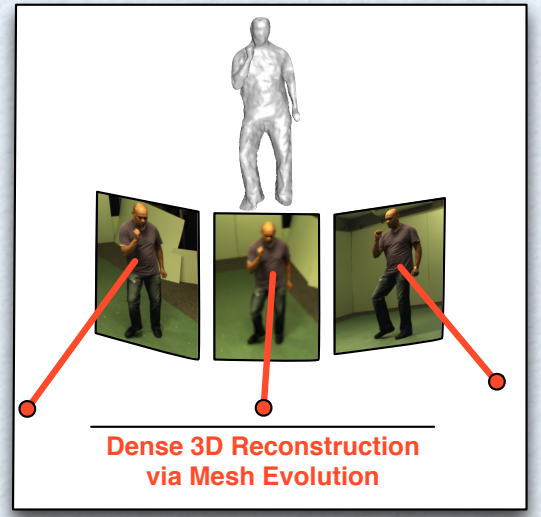




C1

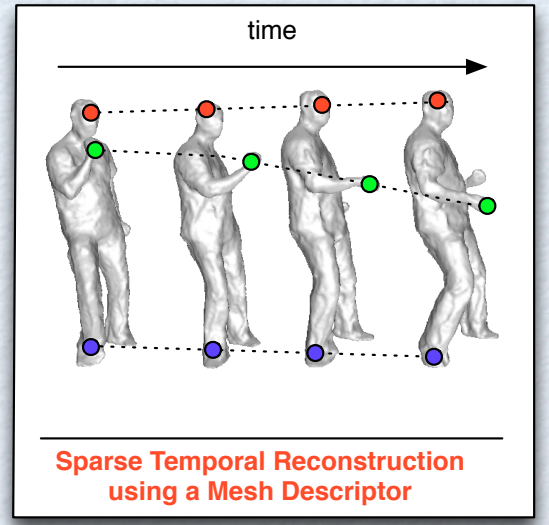


C2

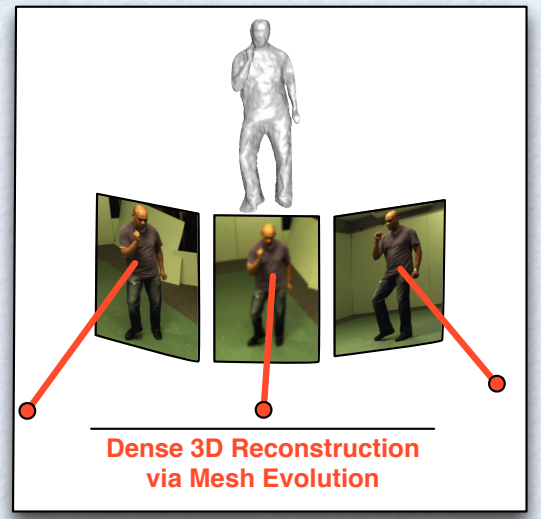


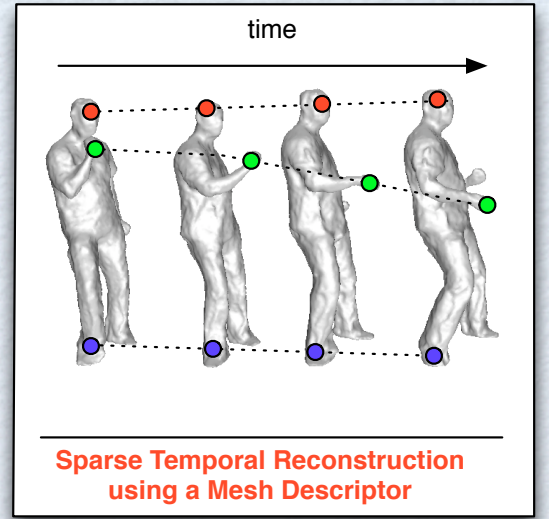
C3

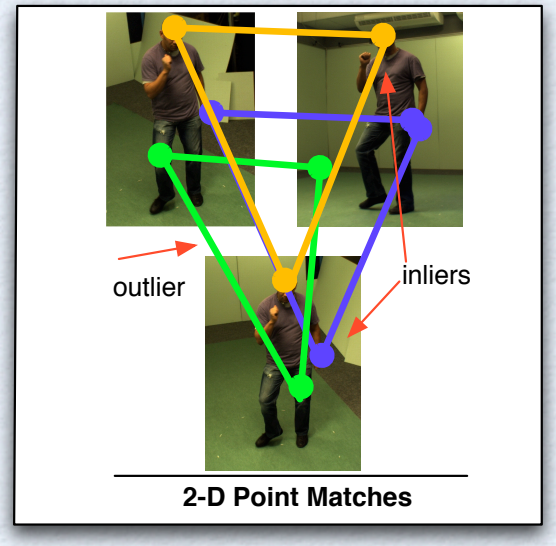
C4

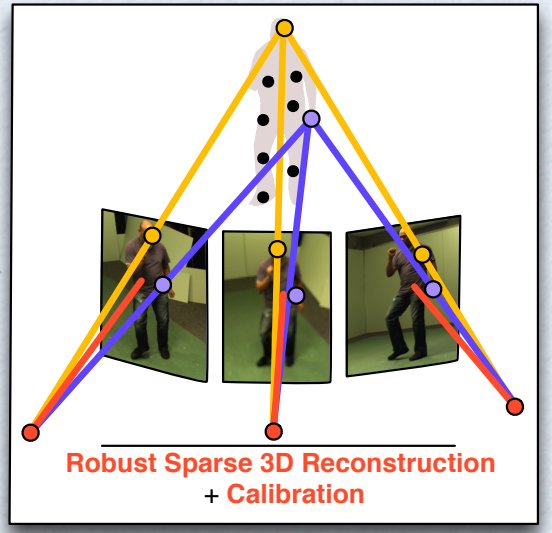


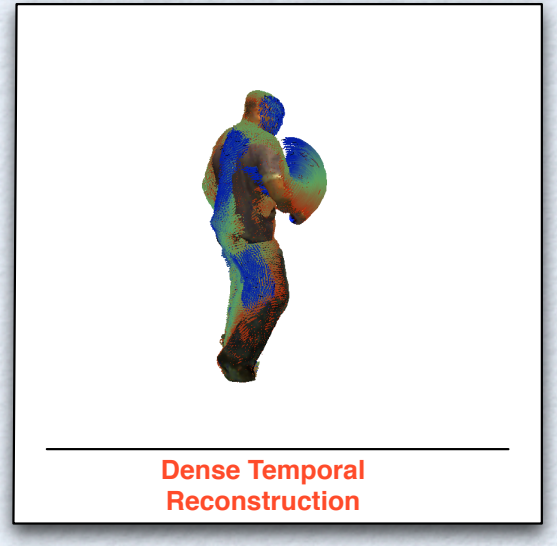
RECONSTRUCTION FRAMEWORK











C1

C2

C3

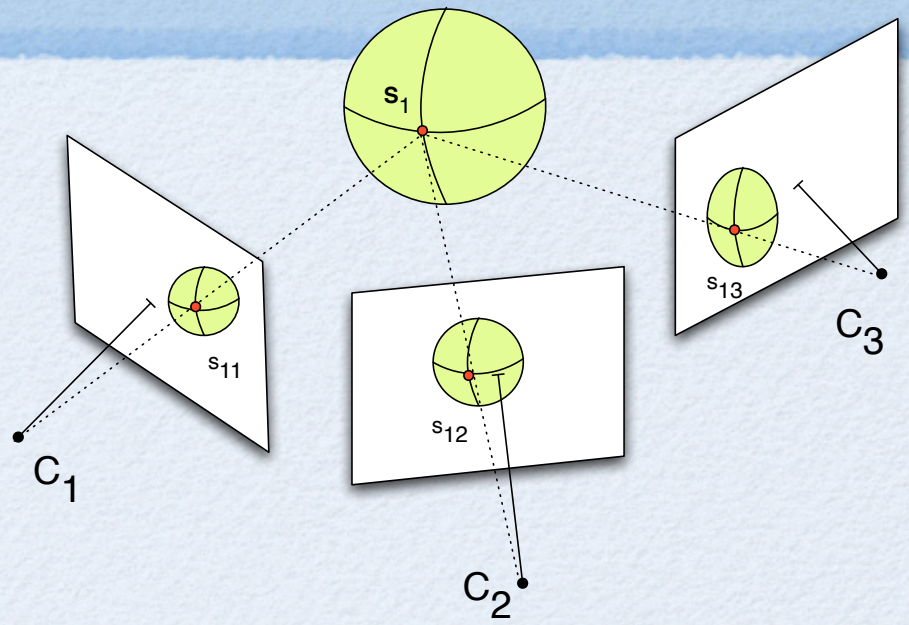
C4

1. A ROBUST PROBABILISTIC FACTORIZATION FRAMEWORK

[Zaharescu & al - 3DPVT 2006]

[Zaharescu & al - IJCV 2009]

FACTORIZATION



- **Factorization** = method that starts from 2-D observations and recovers 3-D points + cameras
- Given point correspondences $\mathbf{s}_{ij}$, form the measurement matrix **S**
- We seek a factorization **S**=**M P**,

  M - affine cameras (linearization of the perspective model)

  P - 3D points.

Related work

- **Affine** factorization methods: Tomasi & Kanade 1992, Aanaes et al. 2002, Hartley & Schaffalitzky 2003, Buchannan & Fitzgibbon 2005, etc
- **Projective/Perspective** factorization methods: Sturm & Triggs 1996, Christy & Horaud 1996, etc
- **Bundle Adjustment** - non linear minimization scheme - needs a good initialization

OUTLIER TREATMENT

- **Random sampling** (i.e. RANSAC - [Fischler and Bolles, 1981]) - slow when dealing with lots of variables
- **Robust Estimators** (i.e. M-estimators - [Huber, 1981]) - hand tuning of threshold parameter

IDEA

- Propose a method that works in conjunction with any affine factorization algorithm and makes it robust to outliers!
- Proof of convergence to a local minima

AFFINE FACTORIZATION WITH EM

Problem formulation

Maximum Likelihood with Missing Data

solved via EM:

E-Step

Posterior Inlier Estimation using a Gaussian/ Uniform mixture model

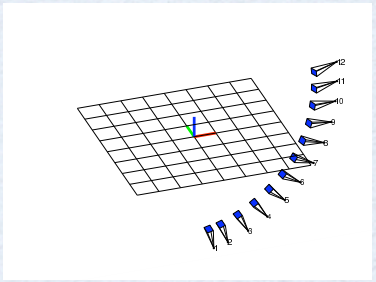
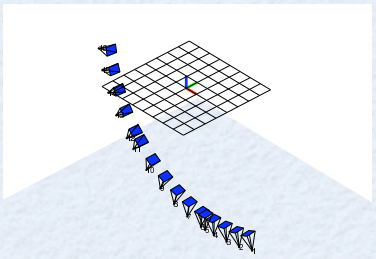
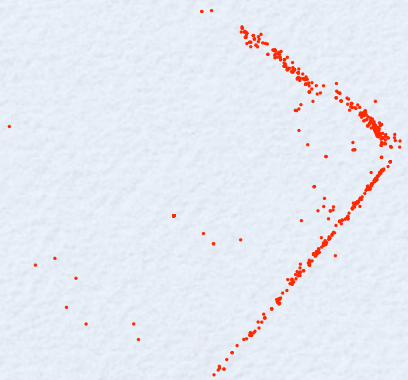
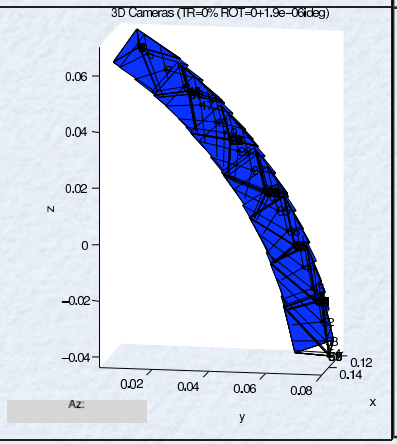
M-Step

Affine Factorization (PowerFactorization)

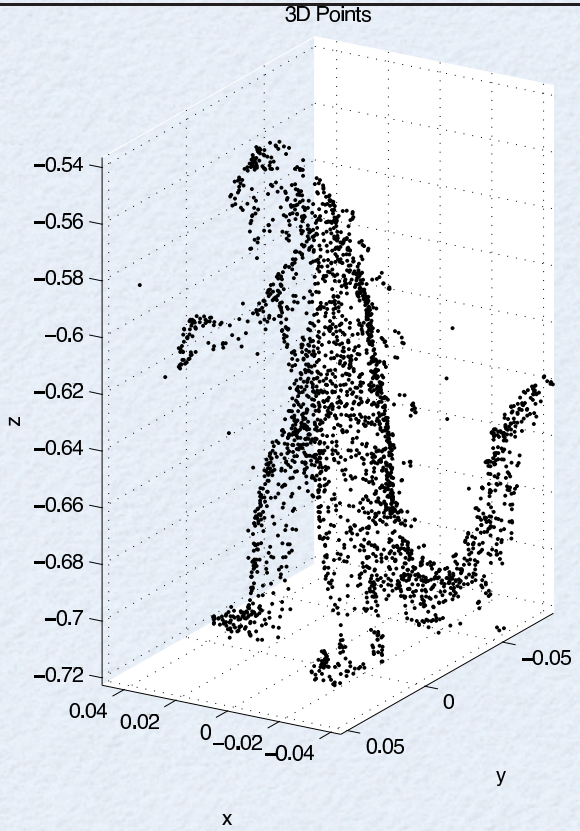
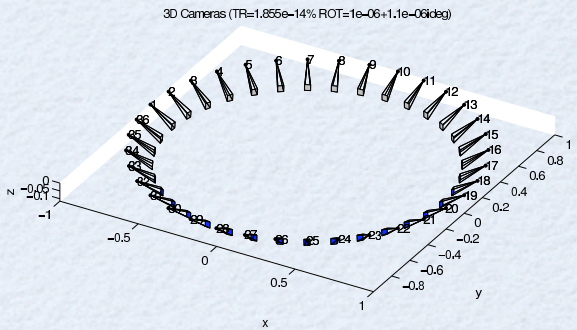
Error Variance Estimation

The error variance is estimated, not like in robust estimators!

3-D REC. RESULTS

	Input Image	3-D Points	Cameras	2-D Reprojection Error
Middleburry Dino				0.82 pixels
Middleburry Temple				0.93 pixels
INRIA Box				0.69 pixels

3-D REC. RESULTS

	Input Image	3-D Points	Cameras
Oxford Dinausor			

2-D Reprojection Error: 0.33 pixels

2. TOPOLOGY ADAPTIVE MESH SURFACE EVOLUTION

[Zaharescu & al - ACCV 2007]

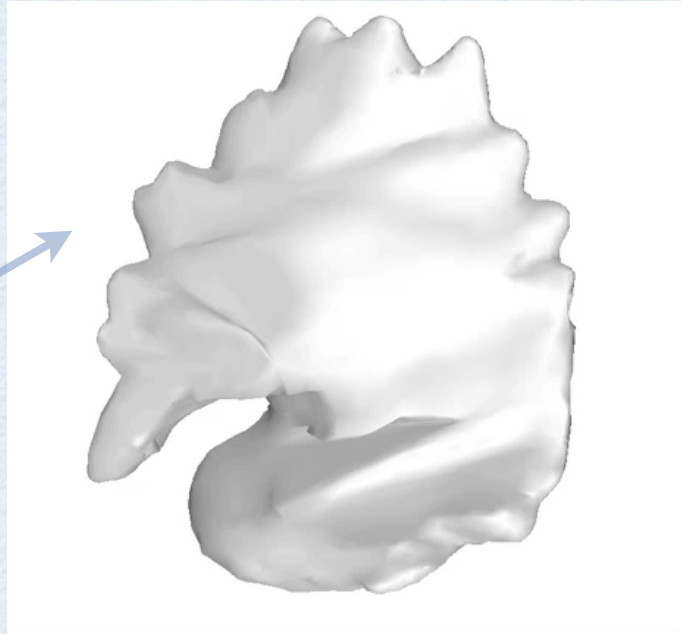
[Zaharescu & al - TVCG -accepted, subject to revisions]

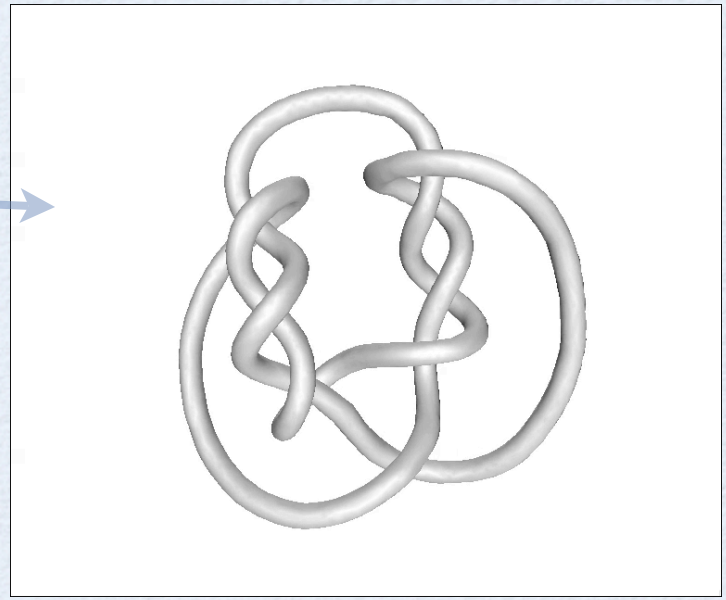
[Varanasi & al - ECCV 2008]

Motivation

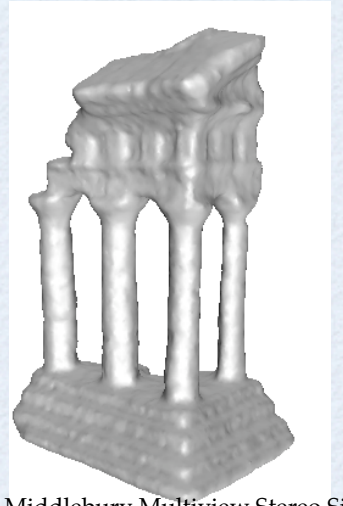
Surface evolution methods are used in a large number of problems, including :

- variational methods for 3D reconstruction
- mesh morphing
- physics based simulations





EXISTING METHODS



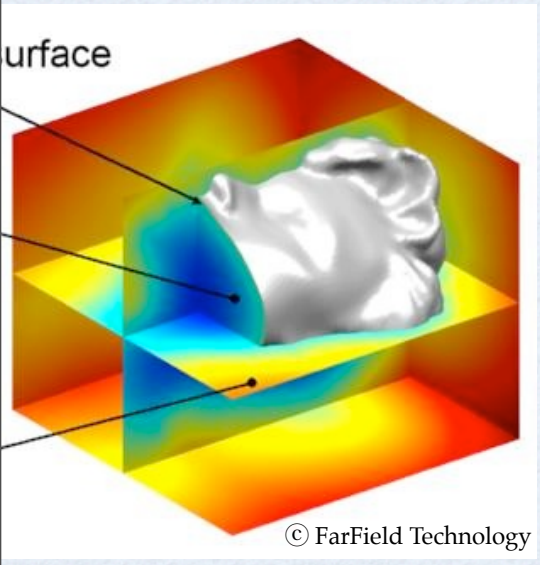
© Middlebury Multiview Stereo Site

Lagrangian methods:

- Adopt a more natural point of view : use meshes and deform them
- Provides compact representation, adaptive resolution
- Lachaud et al. (2000), Pons (2007)

Problems:

- Self-Intersections
- Topology changes (splits and unions)



© FarField Technology

Eulerian methods:

- Level Sets - Embed the mesh within an implicit function
- Solves mesh issues by discretizing space
- Osher & Senthian (1988), many extensions

Problems:

- It loses the mesh resolution up to the grid cell size!
- The mesh has to be recovered at each iteration
- Not suitable to track mesh properties easily (colour, motion, etc)

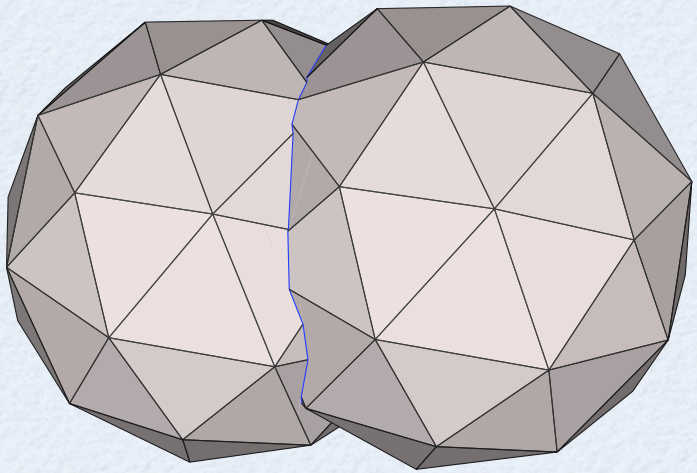
- Wouldn't be great to have a MESH based solution that gracefully handles self-intersections and topology changes ?

TRANSFORMESH

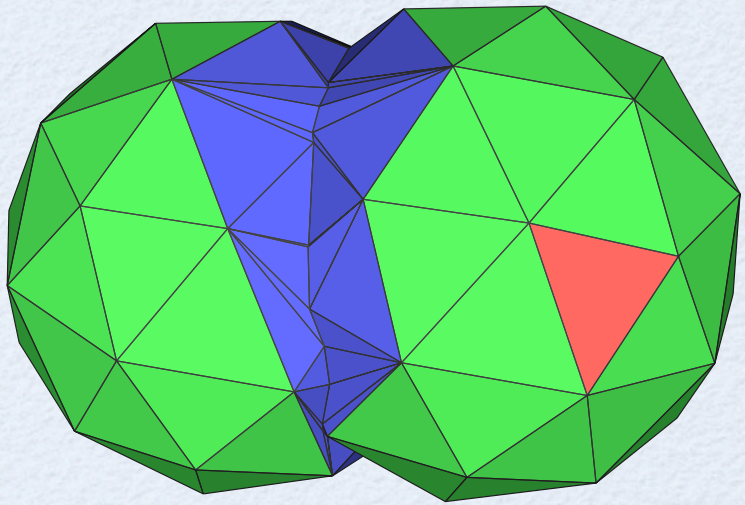
Input: Compact oriented 2D manifold

Output: Compact oriented 2D manifold

Compute Self-Intersections



Valid Region Growing



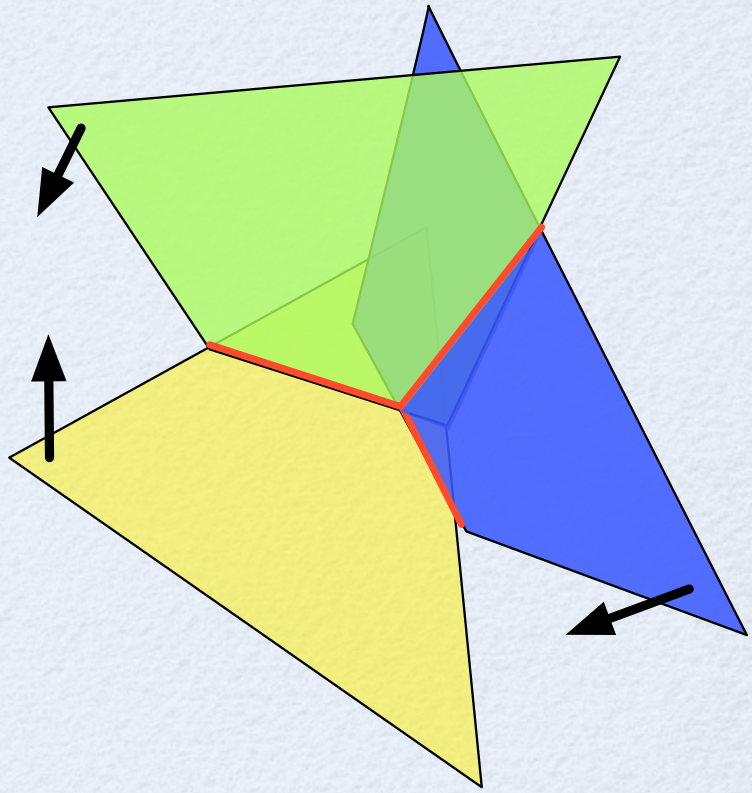
Seed triangle = a triangle without inters. **outside**

Valid triangle = a triangle without intersections

Partial triangle = a triangle with intersections

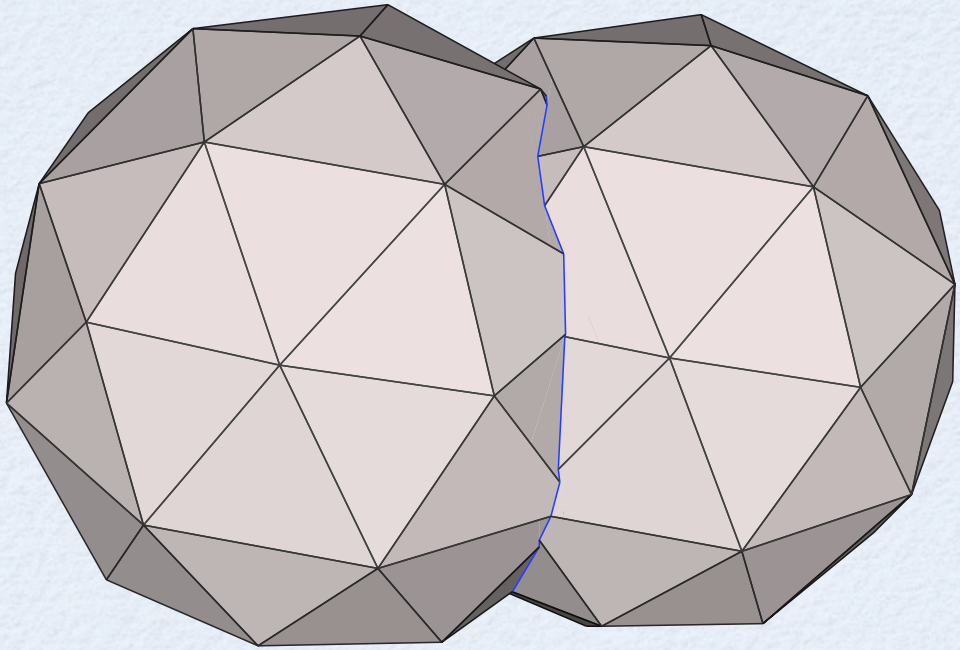
SELF-INTERSECTIONS

- Compute all triangle intersections
  - Use a fast implementation using Box Intersection
  - We obtain intersection segments



VALID REGION GROWING

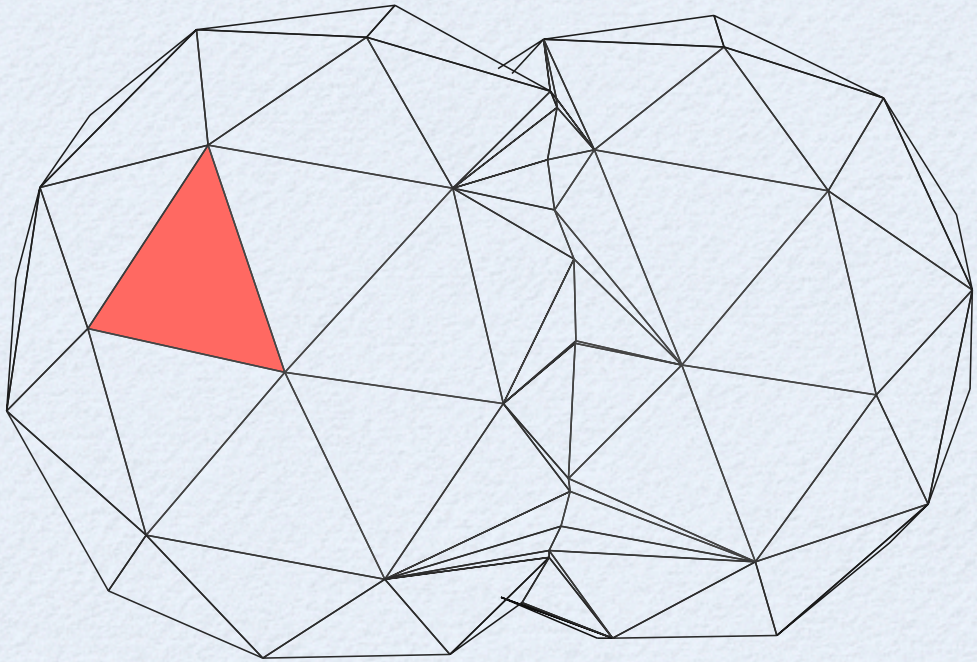
Mark all triangles as Unvisited.



SEED TRIANGLE FINDING

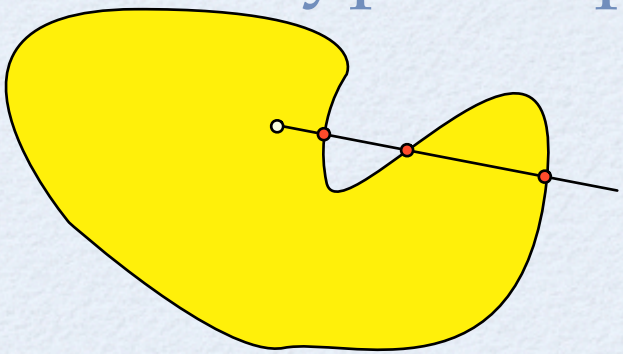
Find a seed triangle to initialize the region growing.

How to find a seed triangle ?



SEED TRIANGLE FINDING

Would the typical "point in polygon" work?

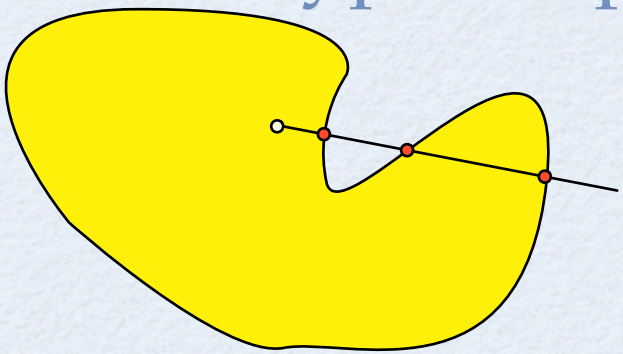


intersections even = outside

intersections odd = inside

SEED TRIANGLE FINDING

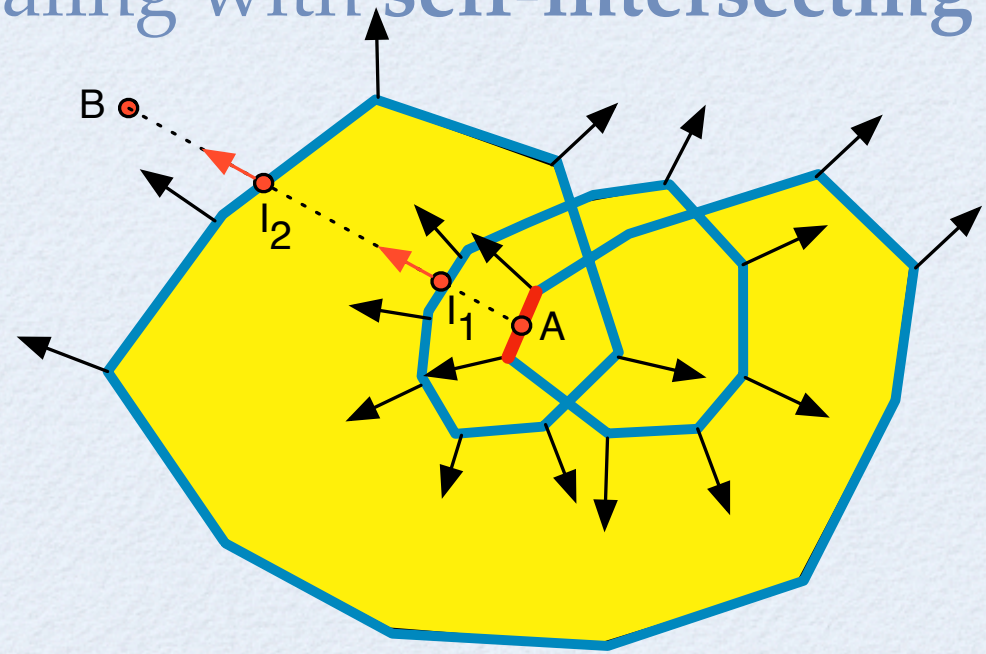
Would the typical "point in polygon" work?



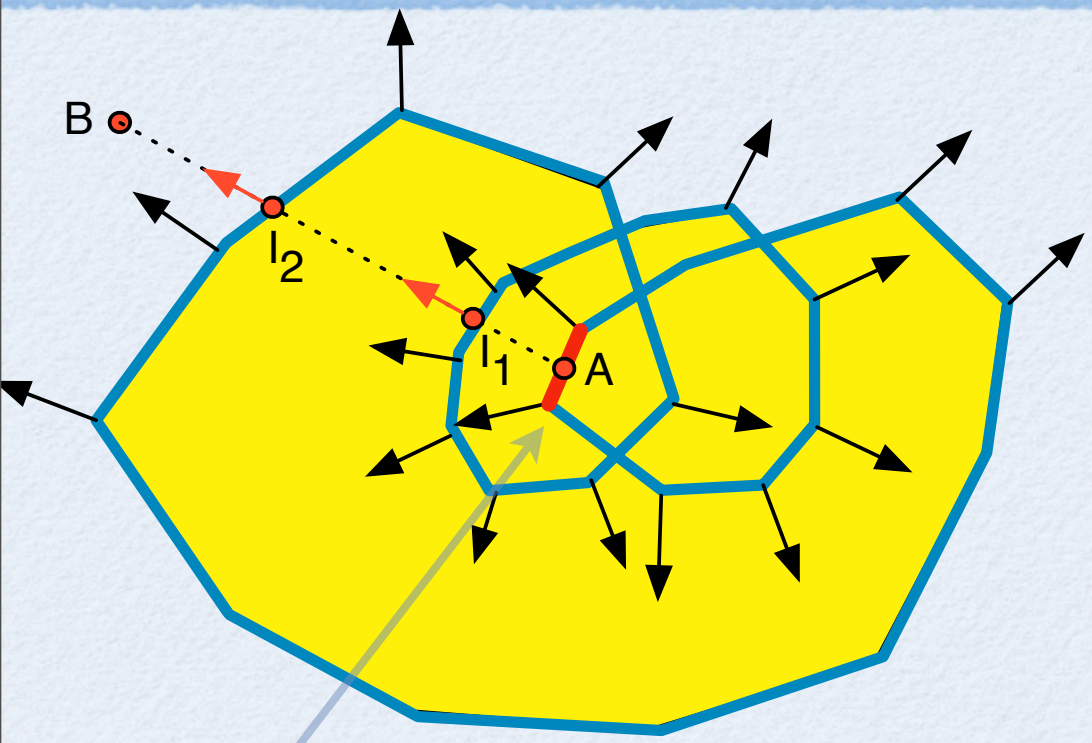
intersections even = outside

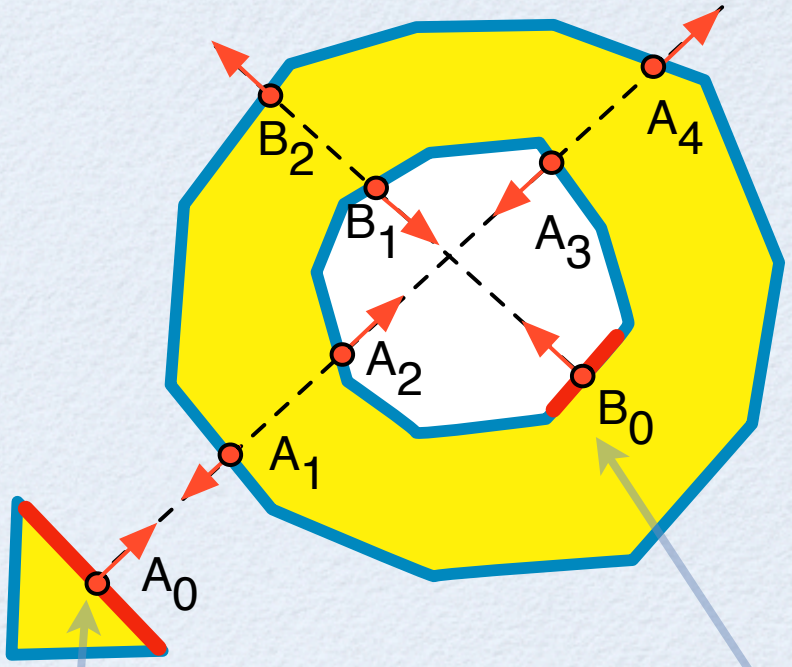
intersections odd = inside

No, the typical "point in polygon" test does not work when dealing with **self-intersecting** polygons.



SEED TRIANGLE FINDING





→ $\{I_1, I_2\}$
← $\{\}$

inside

→ $\{A_0, A_2, A_4\}$
← $\{A_1, A_3, A_5\}$

outside

→ $\{B_1\}$
← $\{B_2\}$

outside

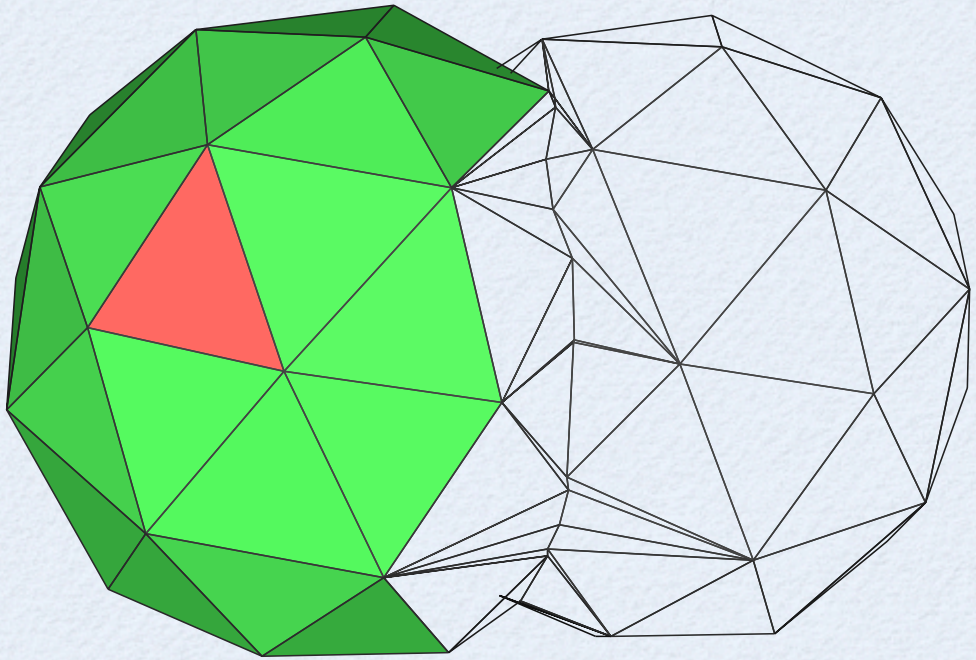
Solution: Keep track of normal orientations using two counters

[# → ≠ # ←] if inside

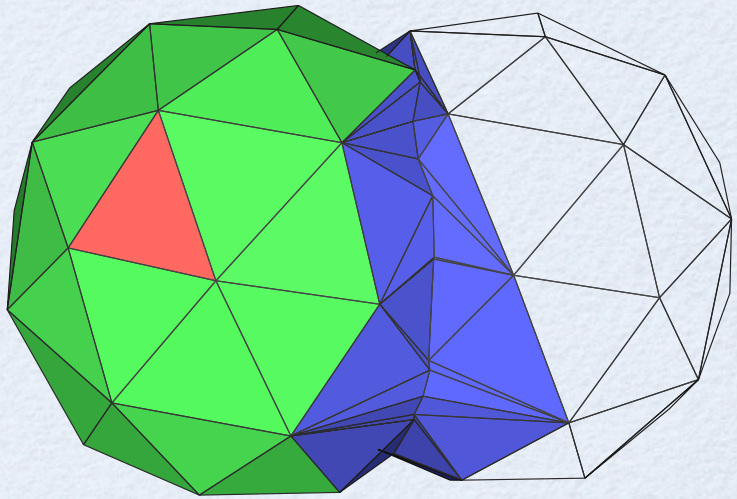
[# → = # ←] if outside

VALID TRIANGLE EXPANSION

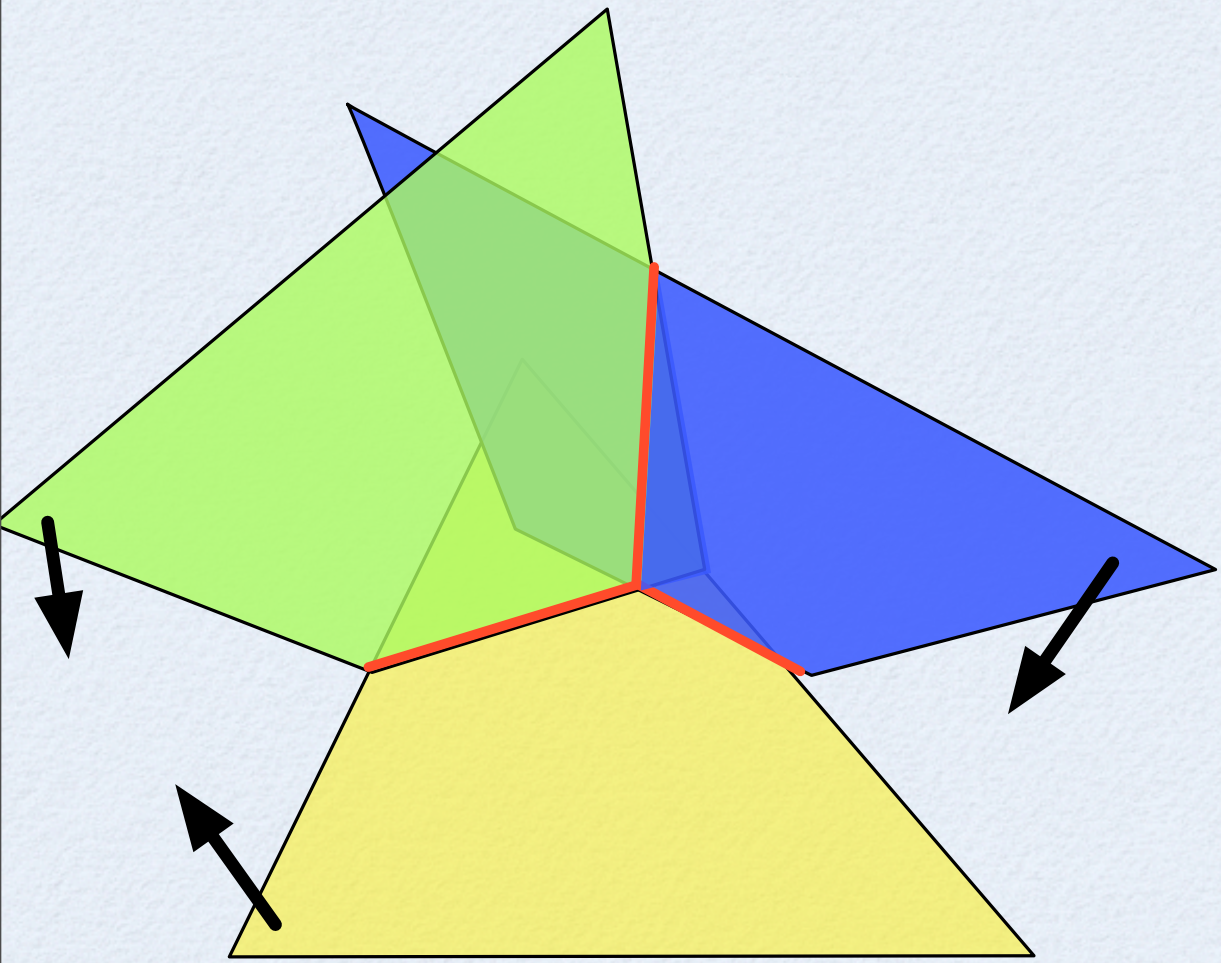
Expand on the valid triangles

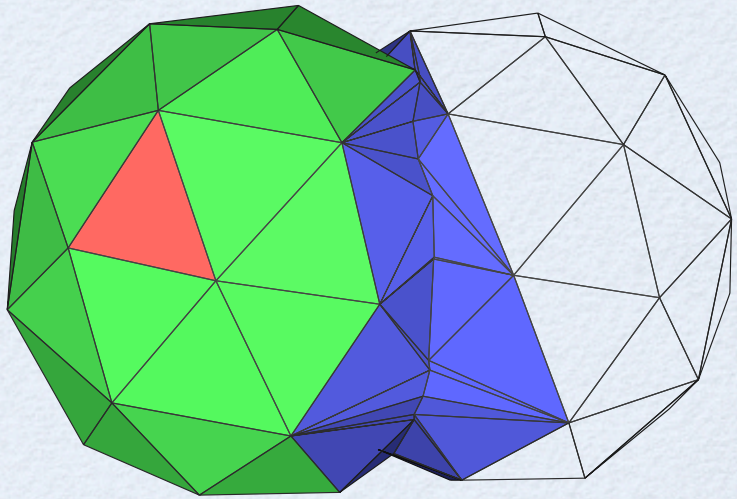


PARTIAL TRIANGLE EXPANSION



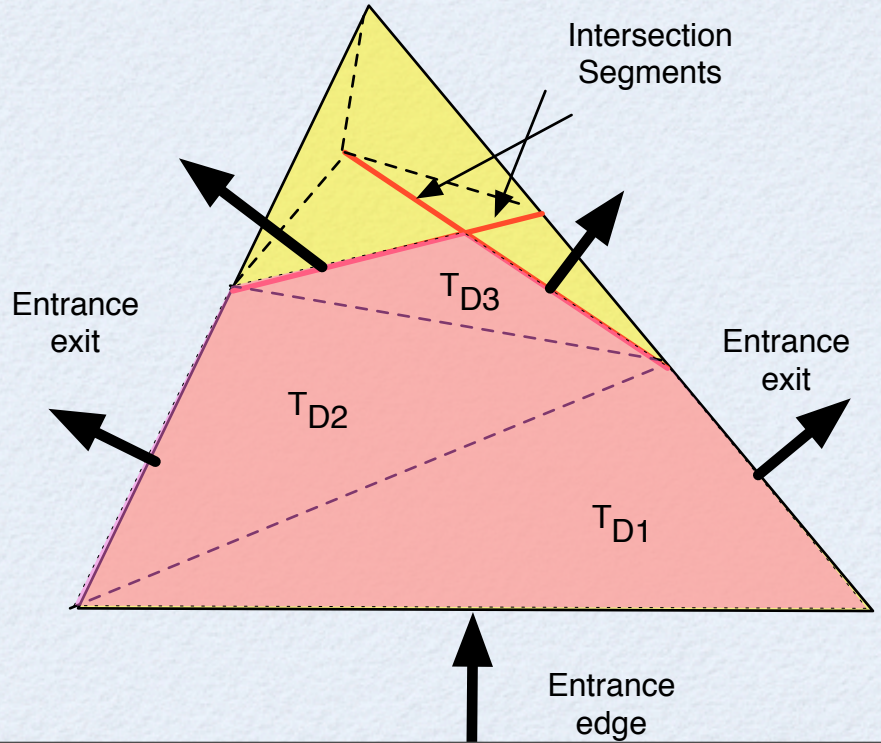
PARTIAL TRIANGLE EXPANSION

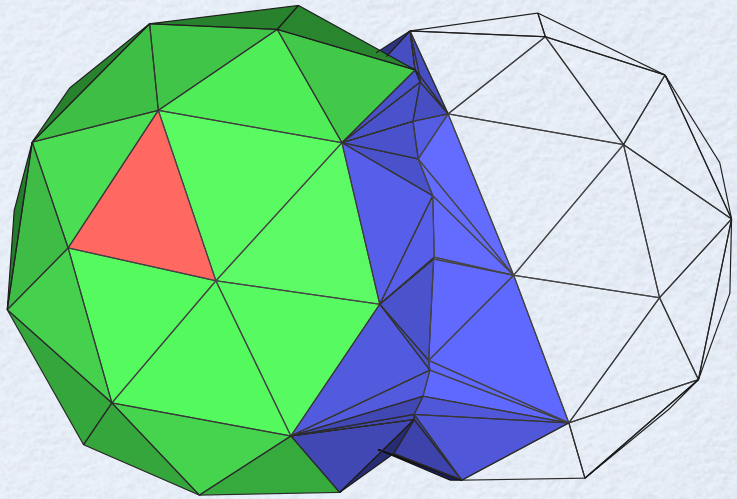




PARTIAL TRIANGLE EXPANSION

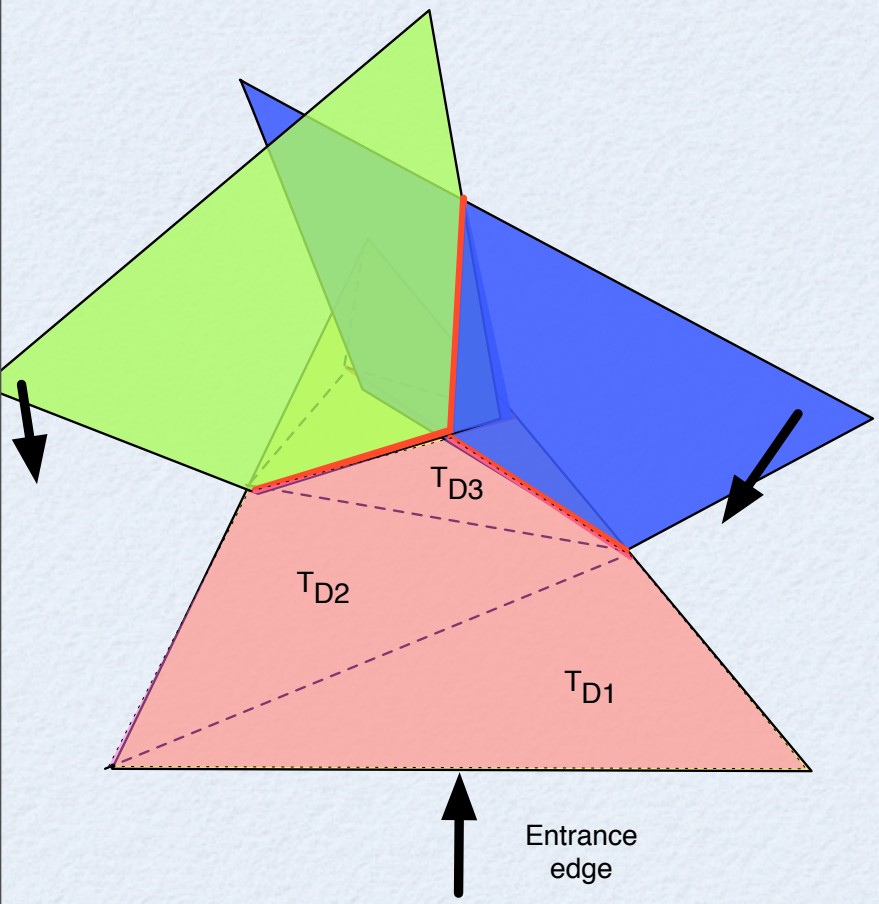
Redefine local geometry using a constrained 2D Delaunay Triangulation





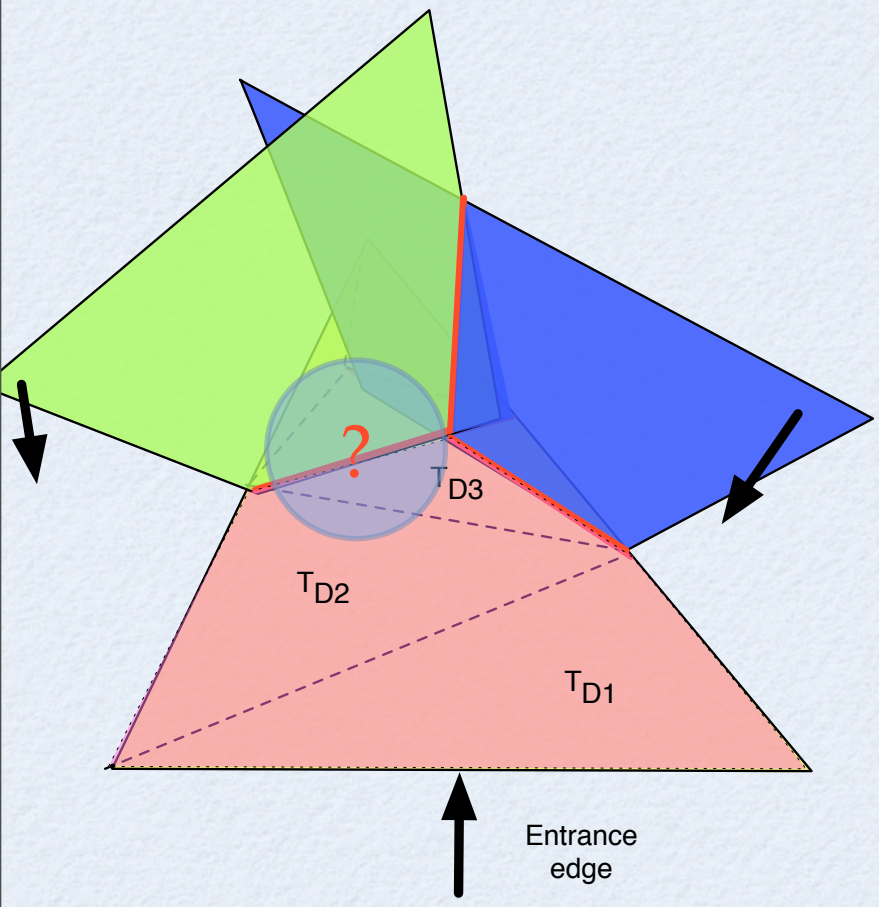
PARTIAL TRIANGLE EXPANSION

"Crossing the river": traverse the partially valid triangles: crossing to the next triangle method is CRUCIAL



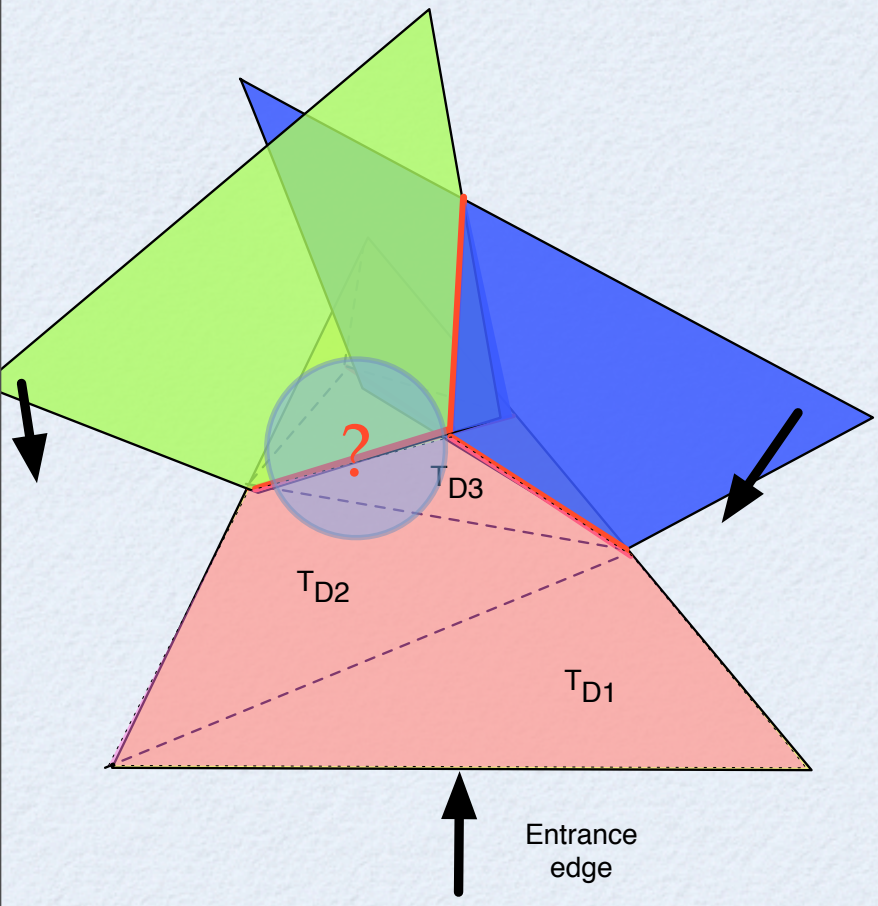
PARTIAL TRIANGLE EXPANSION

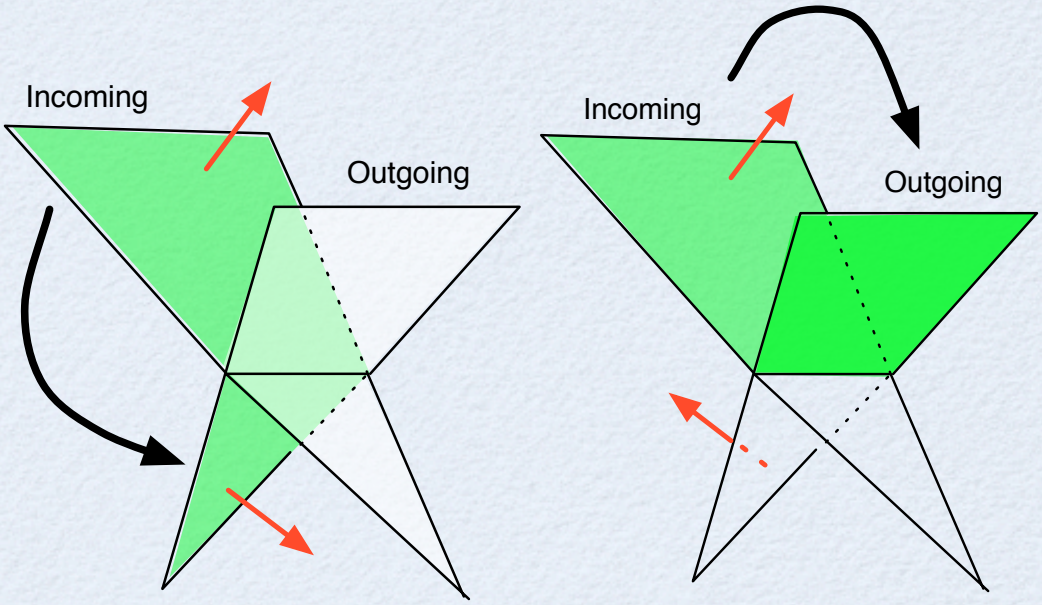
"Crossing the river": traverse the partially valid triangles: crossing to the next triangle method is CRUCIAL



PARTIAL TRIANGLE EXPANSION

"Crossing the river": traverse the partially valid triangles: crossing to the next triangle method is CRUCIAL



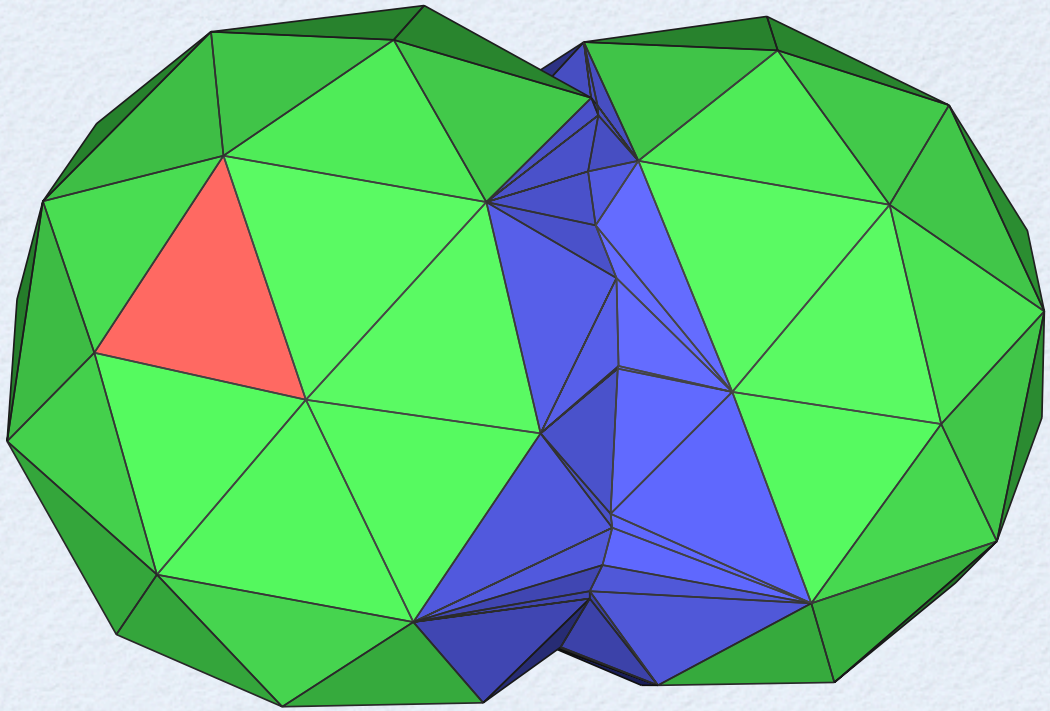


two cases

Choose the triangle side whose normal is consistent with the one of the incoming triangle

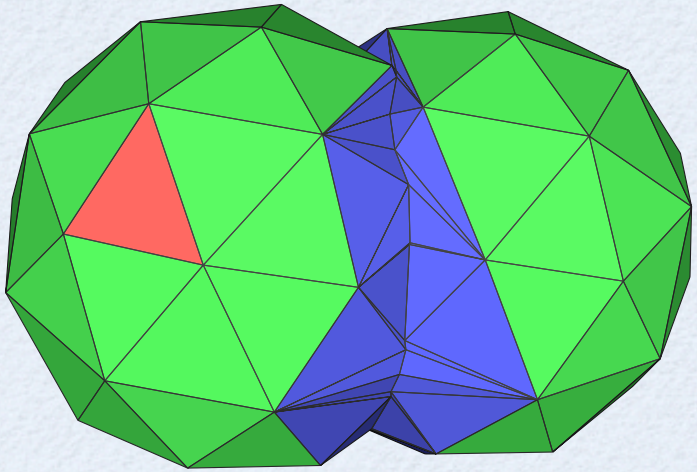
VALID REGION GROWING

Expand until the current region finished growing.

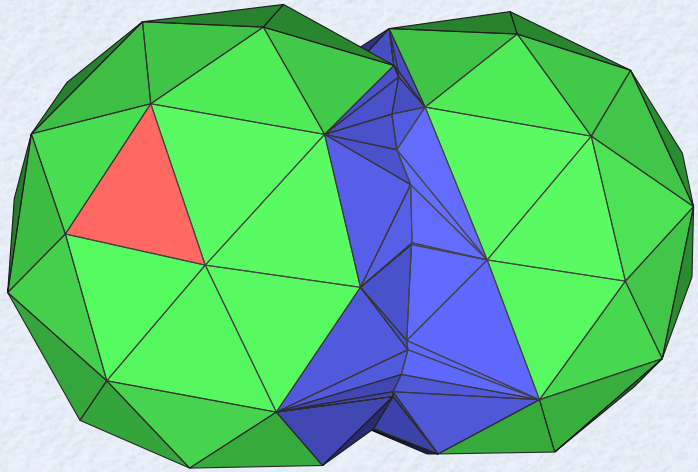


If another unused seed triangles is found, start another region growing .

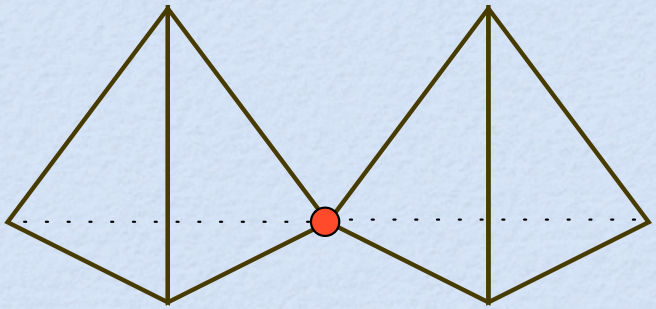
STITCHING



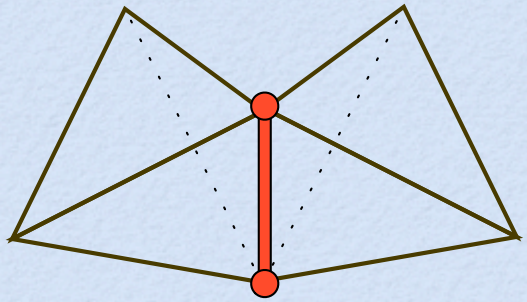
STITCHING



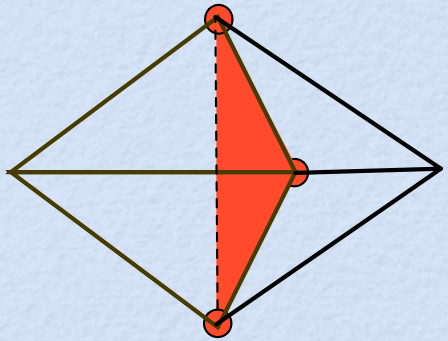
Degenerate configurations:



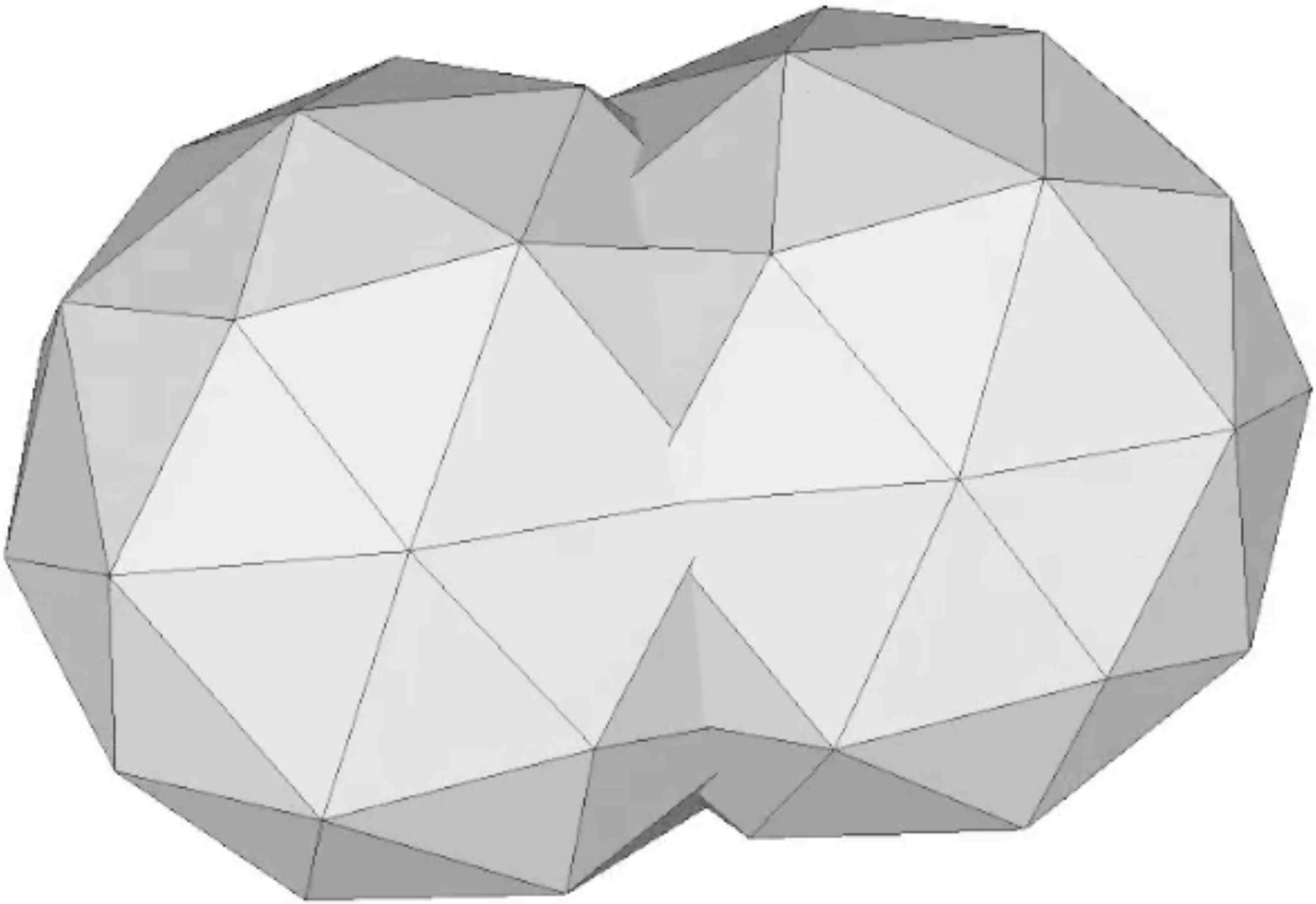
Singular Vertex



Singular Edge



Singular Facet



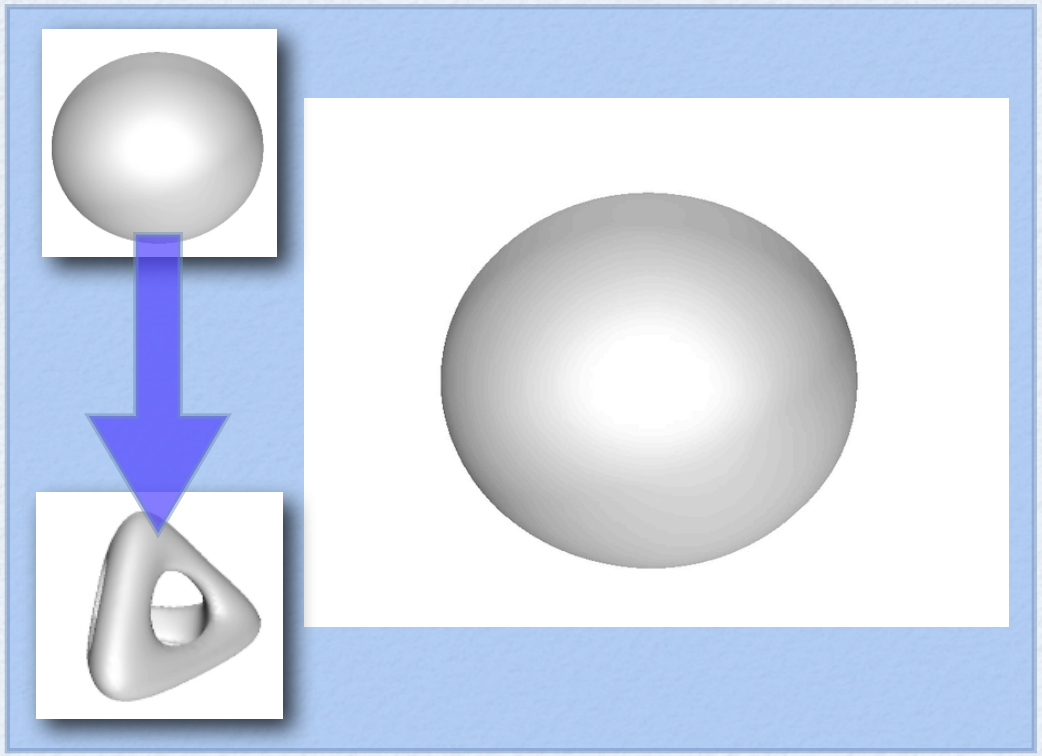
Applications

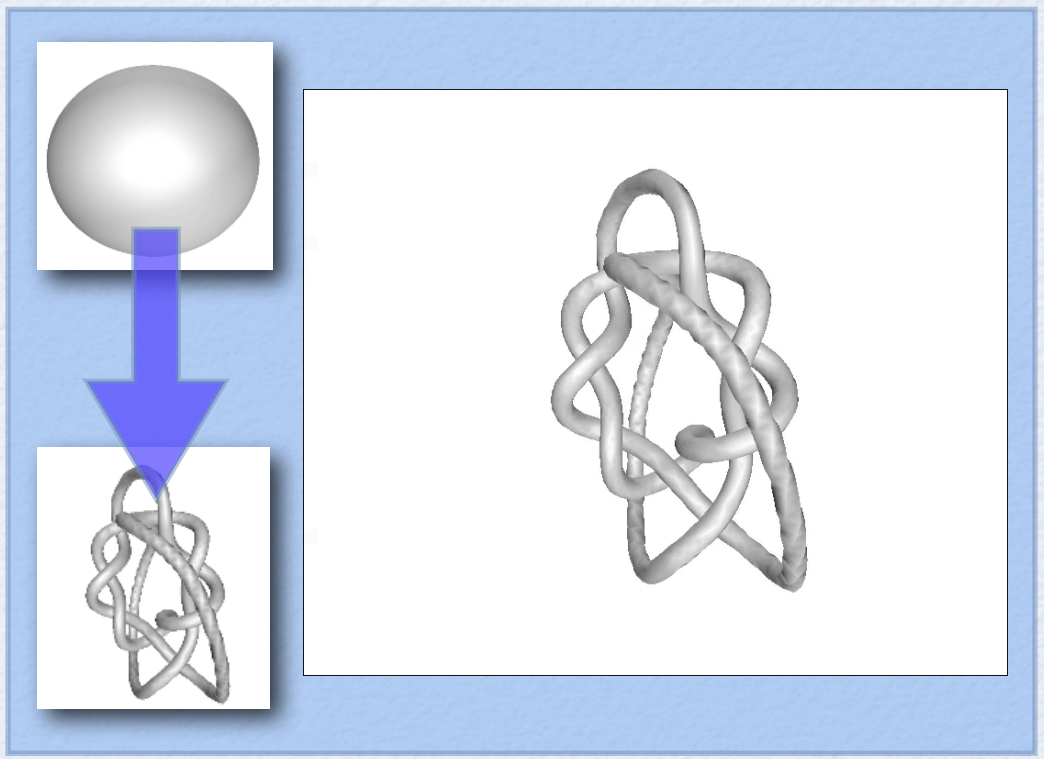
- 3-D Mesh Morphing
- Multiple Camera 3-D Reconstruction

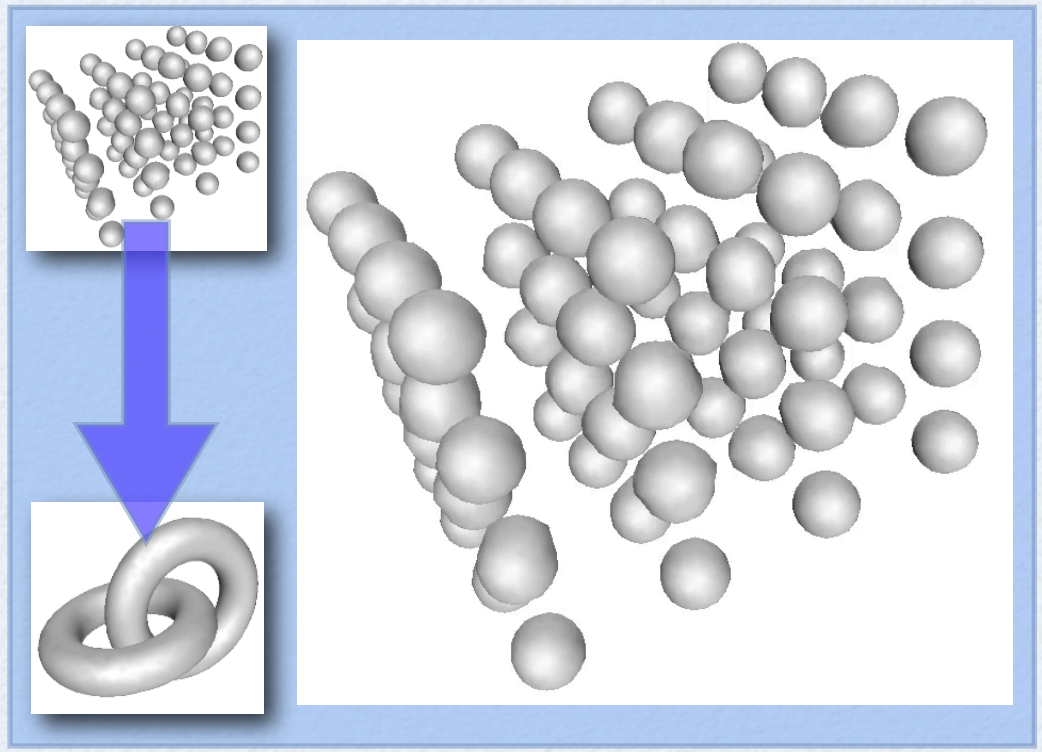
Mesh Morphing

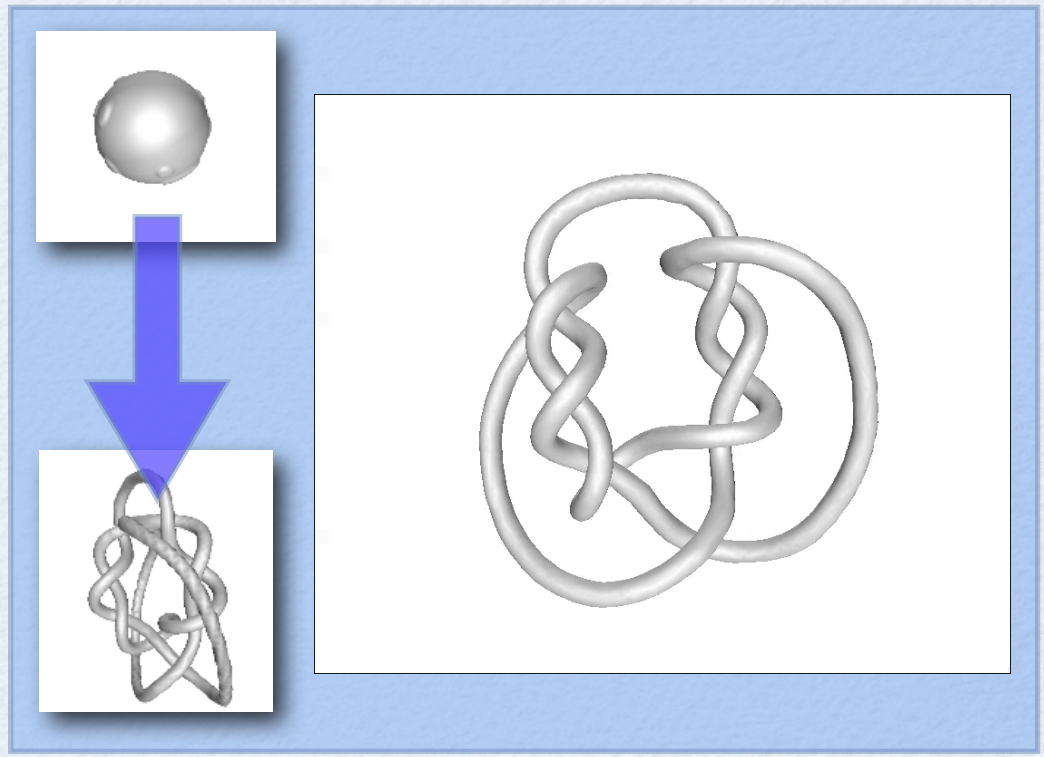
Mesh Morphing

- Morph a source mesh into a destination mesh based on a signed distance function
- Allows to easily test various complicated case

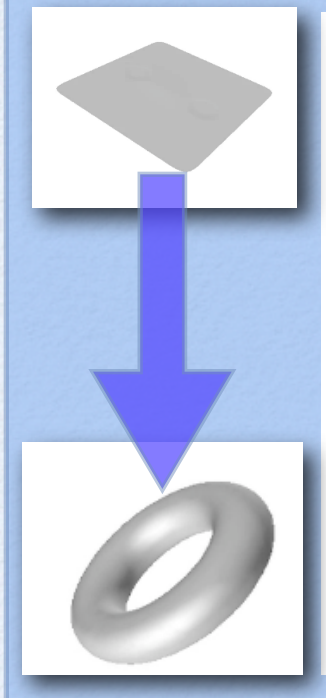


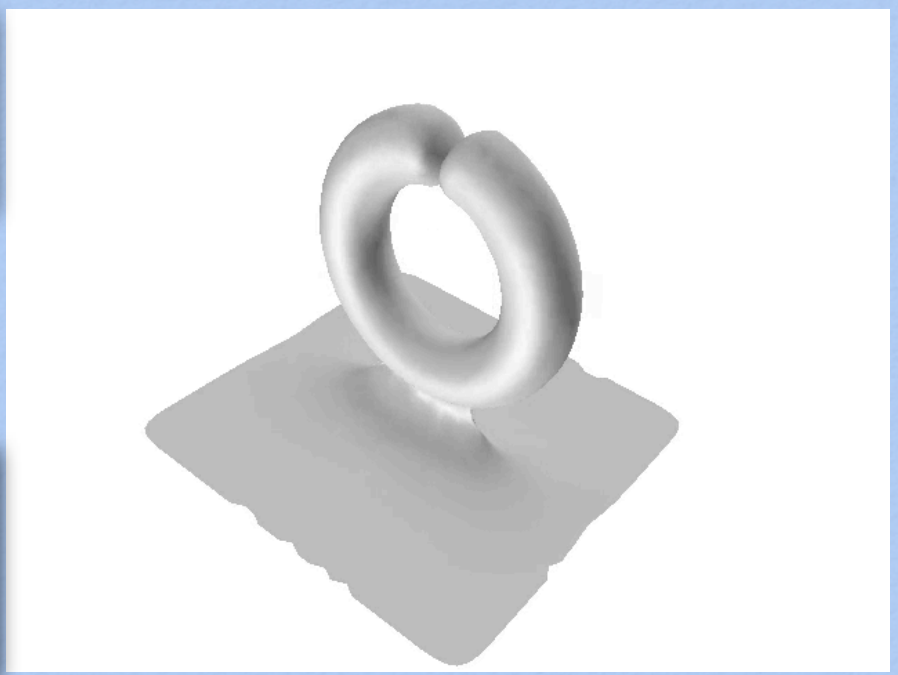


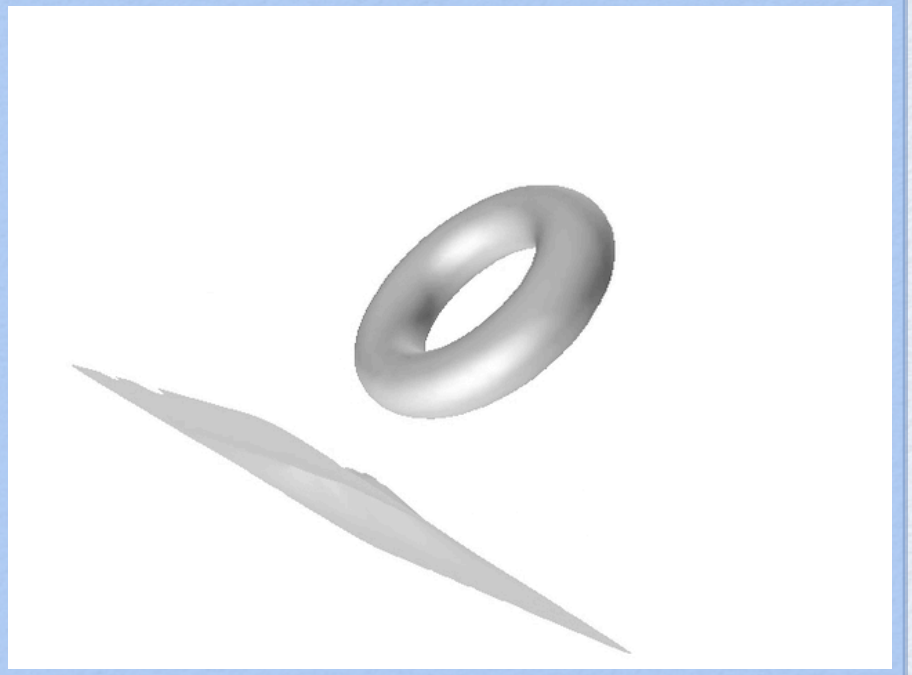




OPEN SURFACES





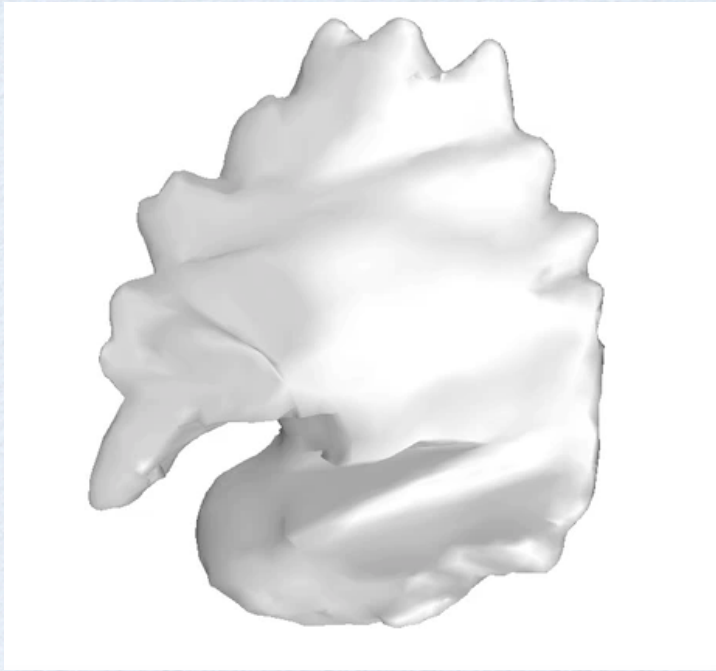


TransforMesh can be **easily extended** to deal with open surfaces

3-D RECONSTRUCTION

3-D RECONSTRUCTION

- Start from an initial surface:
  - visual hulls (EPVH - Franco PAMI 2008)
  - triangulated 3-D points (Powercrust - Amenta CGTA 2001)

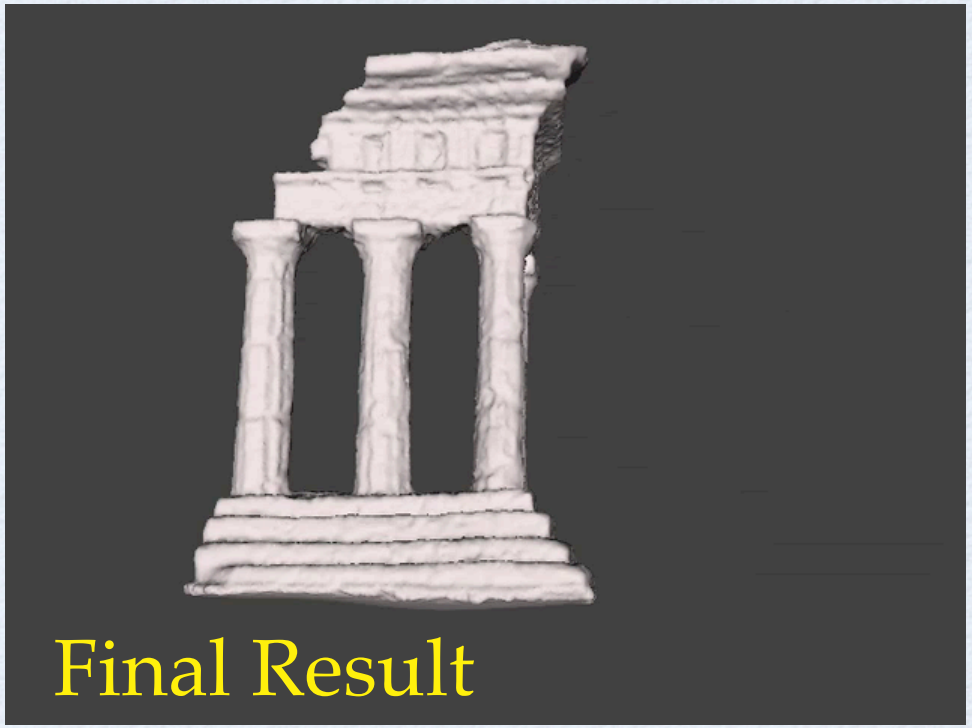


- Evolve the mesh by imposing photometric constraints (Pons, Keriven & Faugeras IJCV 2007)

Visual Hull

Final Result

Visual Hull



Final Result

TransforMesh ranks in top 1-3 on the Middleburry Eval. Site

Visual Hull

Final Result

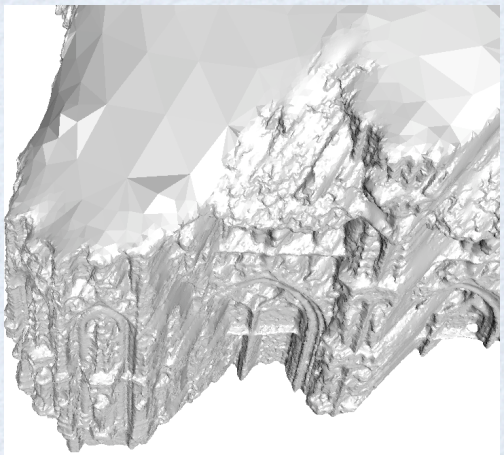
Visual Hull

Final Result

TransforMesh ranks in top 1-3 on the Middleburry Eval. Site

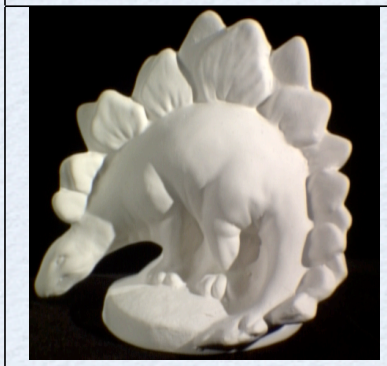
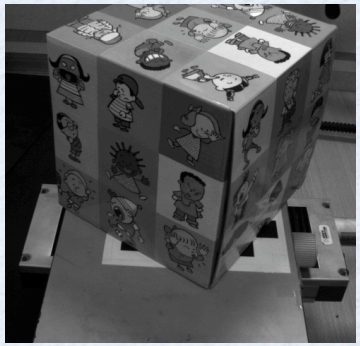
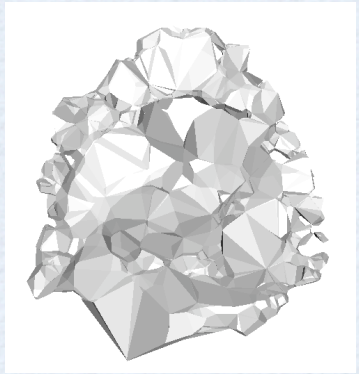
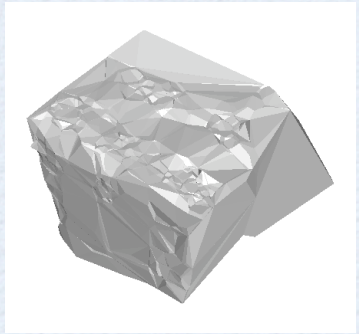
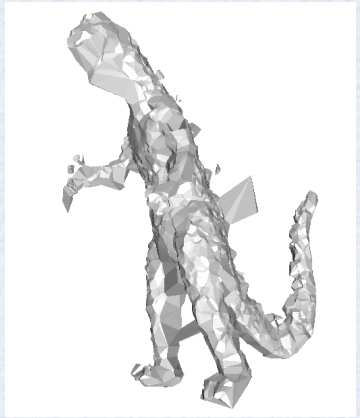
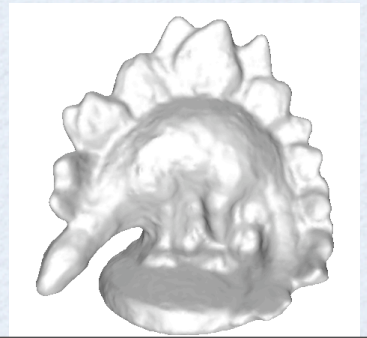
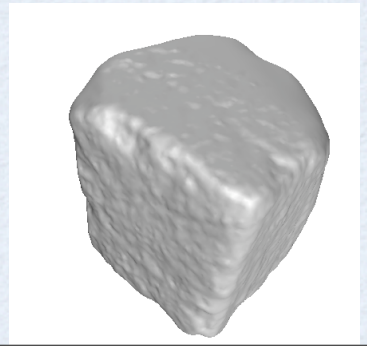
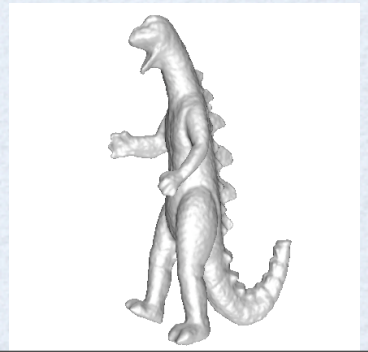
LEUVEN CITY HALL (EPFL)

Input: 3000x2000



Over 1.5 million triangles

FROM SPARSE TO DENSE REC.

Dino	Temple	Box	Dinausor
			
			
			

RECAP

- TransforMesh
  - Algorithm that computes **EXACTLY** all the intersections
  - Handles **TOPOLOGY** changes **NATURALLY**
  - It easily extends to open surfaces
  - **A library will shortly be available for download** (C++, uses CGAL)

3. CAMERA CLUSTERING

[Zaharescu & al - ECCV Workshop on Multiple Cameras 2008]

MOTIVATION

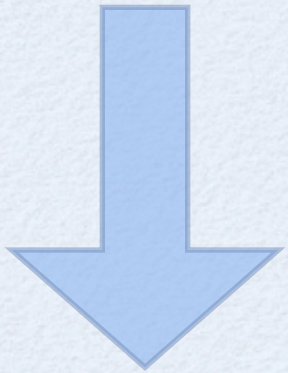
Large 3-D reconstruction tasks with 100s of cameras become intractable (memory-wise and time-wise)

How to split large reconstruction tasks?

MOTIVATION

Large 3-D reconstruction tasks with 100s of cameras become intractable (memory-wise and time-wise)

How to split large reconstruction tasks?



Perform **Content-Aware** Camera Clustering

IDEA

- Use both scene geometry (content) and camera geometry to perform the clustering:

  1. Obtained matched 2-D points in images using:

     a. 3-D reconstruction at a **coarse** resolution
     b. point detectors + descriptors (i.e. SIFT, HOG)

  2. Use visibility information to cluster the scene

Method

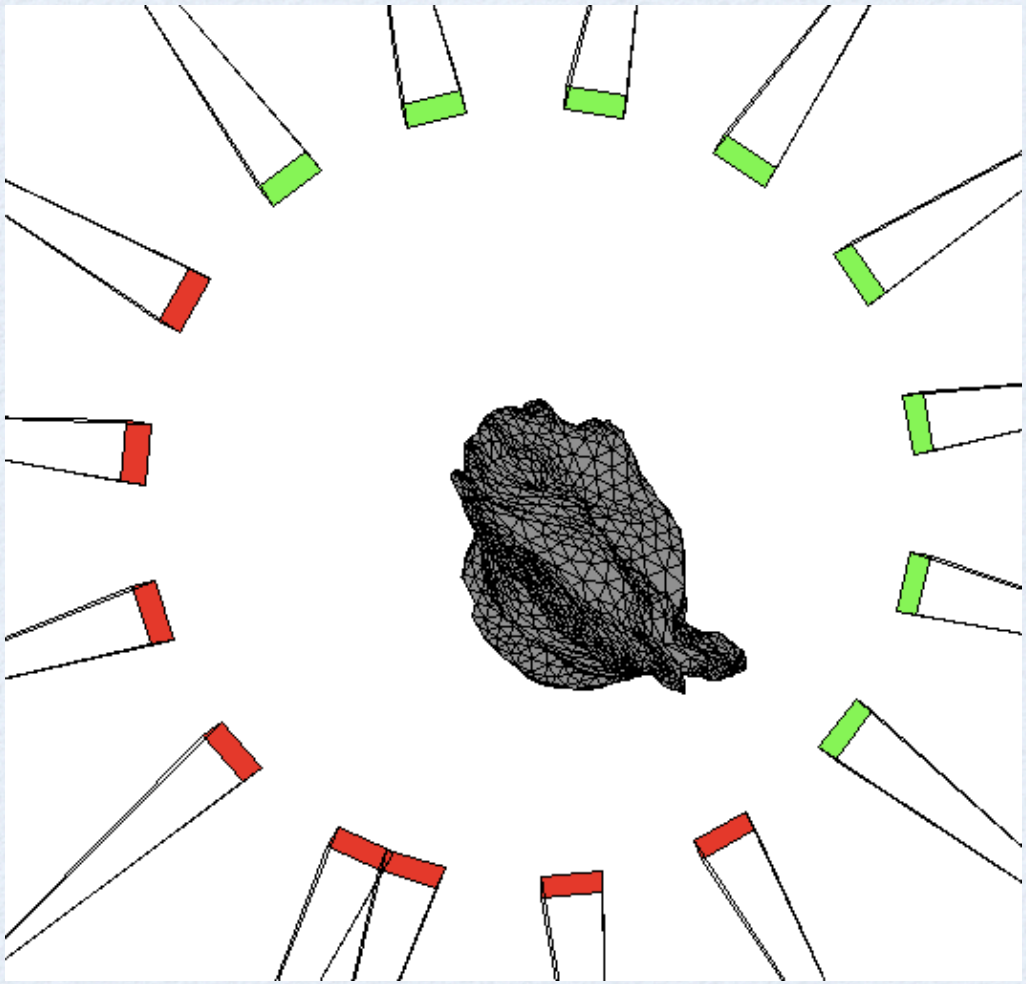
Visibility matrix (1 - visible; 0 otherwise)

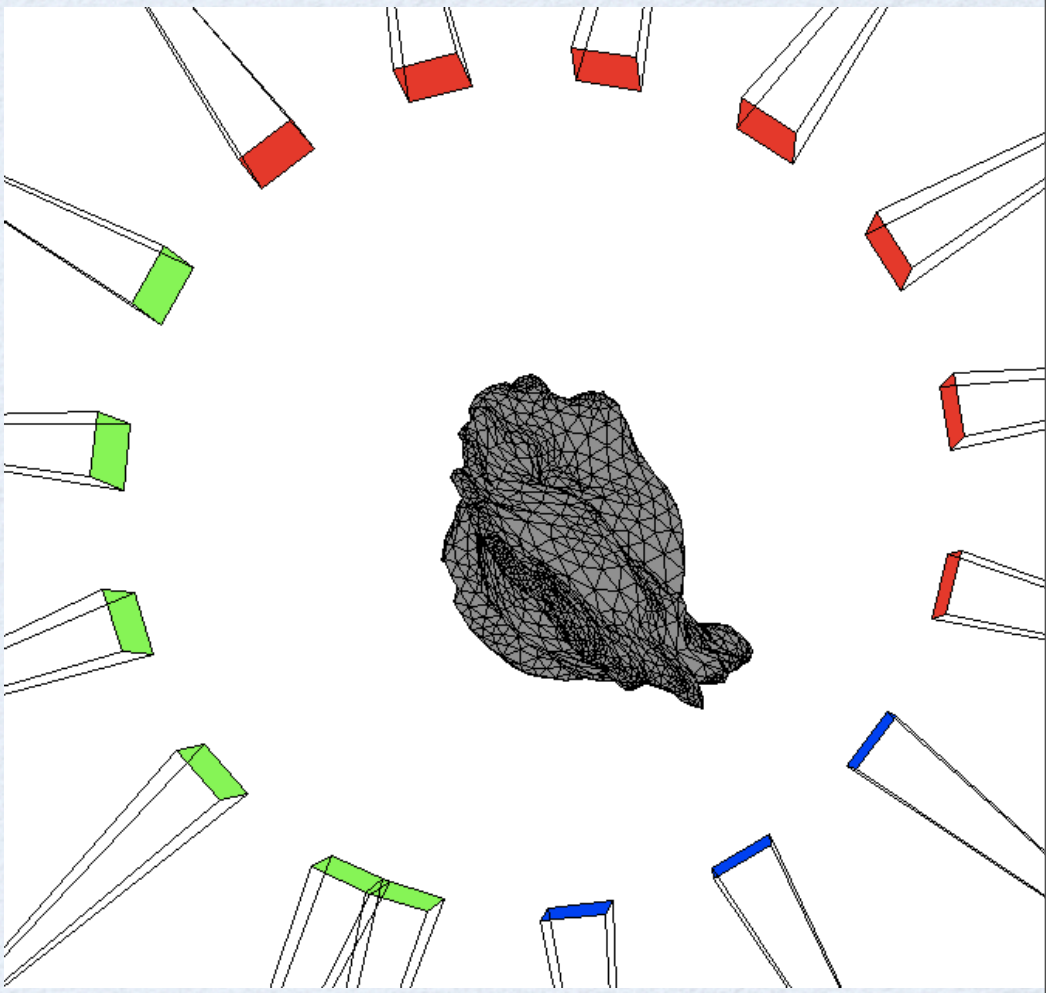
$$\Delta = \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \cdots & \beta_{1,N_c} \\ \beta_{2,1} & \beta_{2,2} & \cdots & \beta_{2,N_c} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{N_v,1} & \beta_{N_v,2} & \cdots & \beta_{N_v,N_c} \end{bmatrix}$$

Perform k-means clustering:

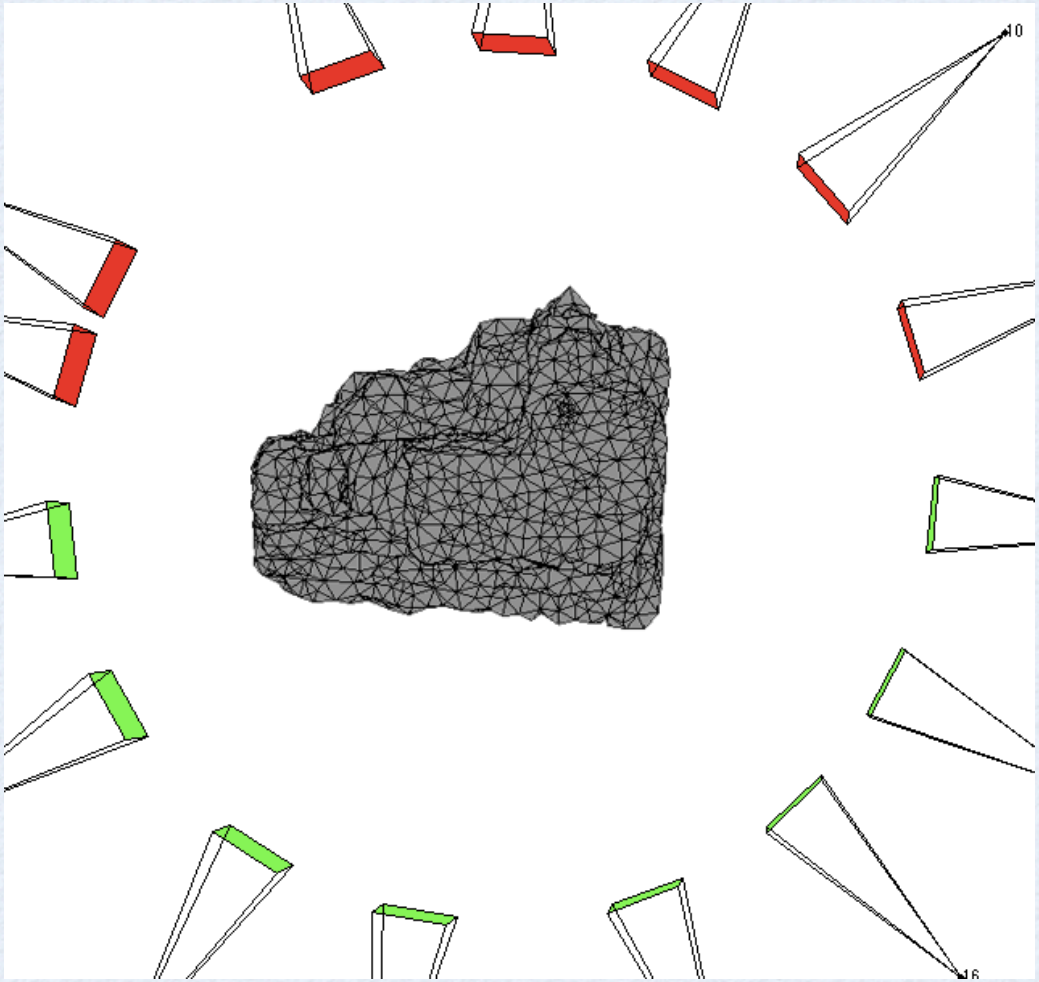
- *camera-based* using the columns of Δ
- *geometry-based* using the rows of Δ (involves a 2nd step of selecting the most discriminative cameras for each geometry cluster - one tuning param.)

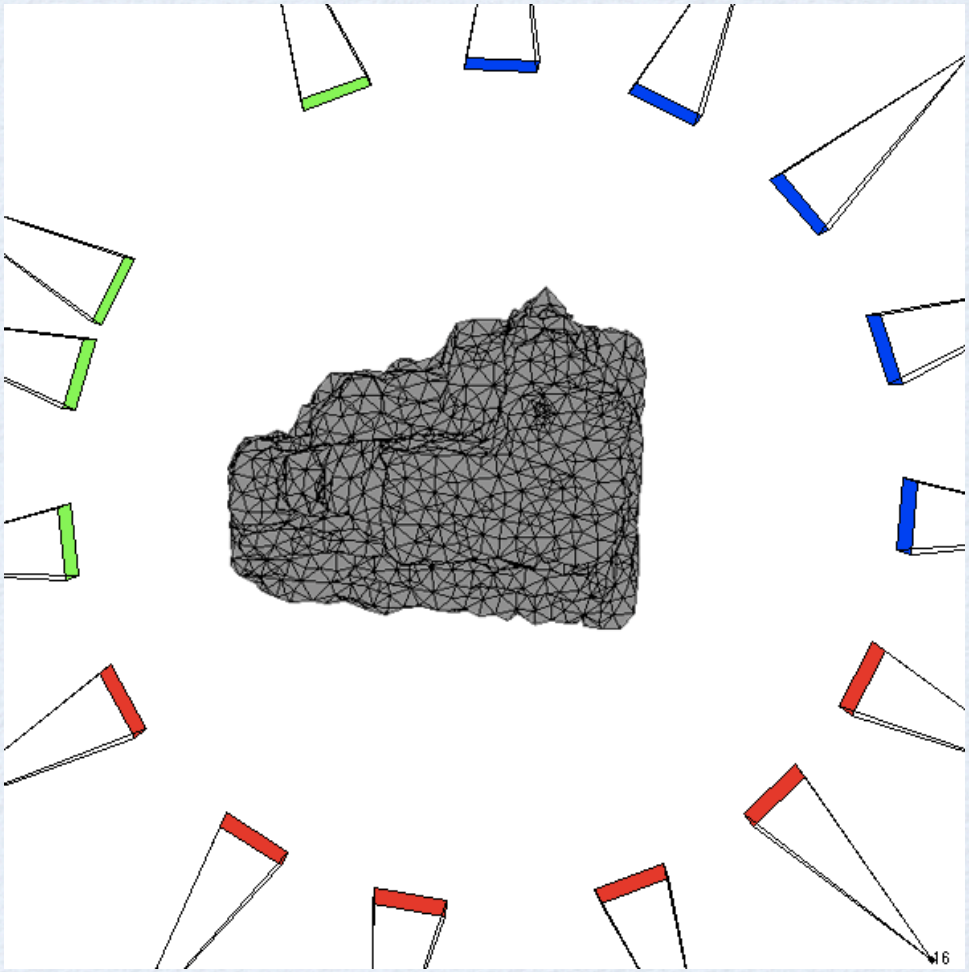
CAMERA-BASED CLUSTERS



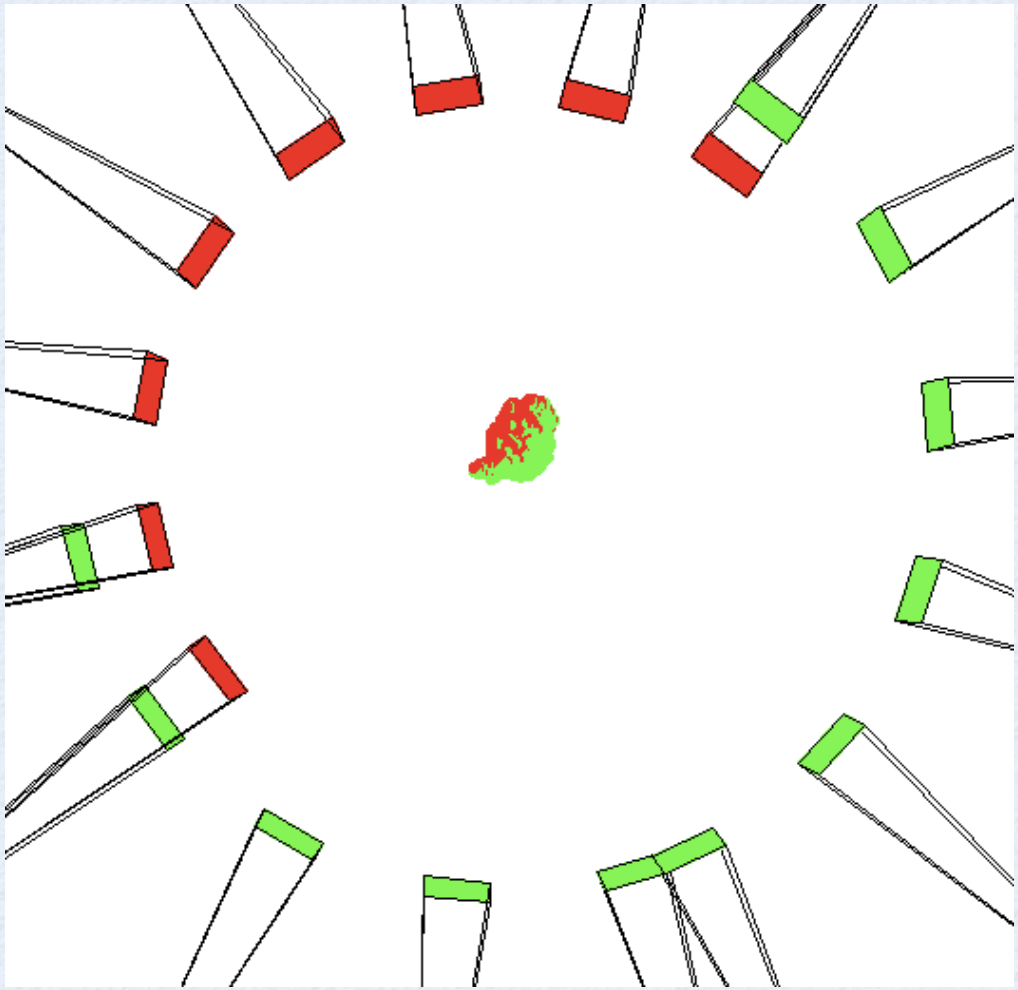


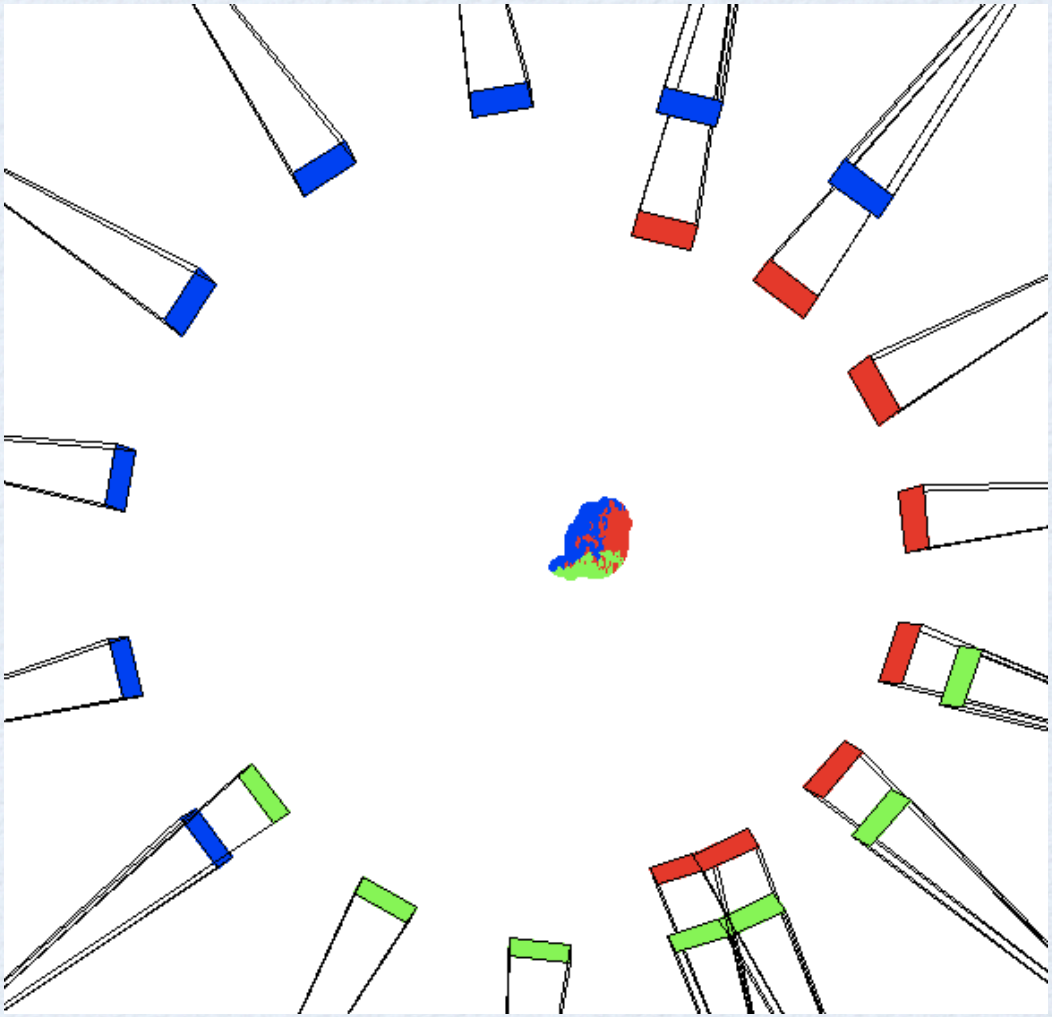
CAMERA-BASED CLUSTERS



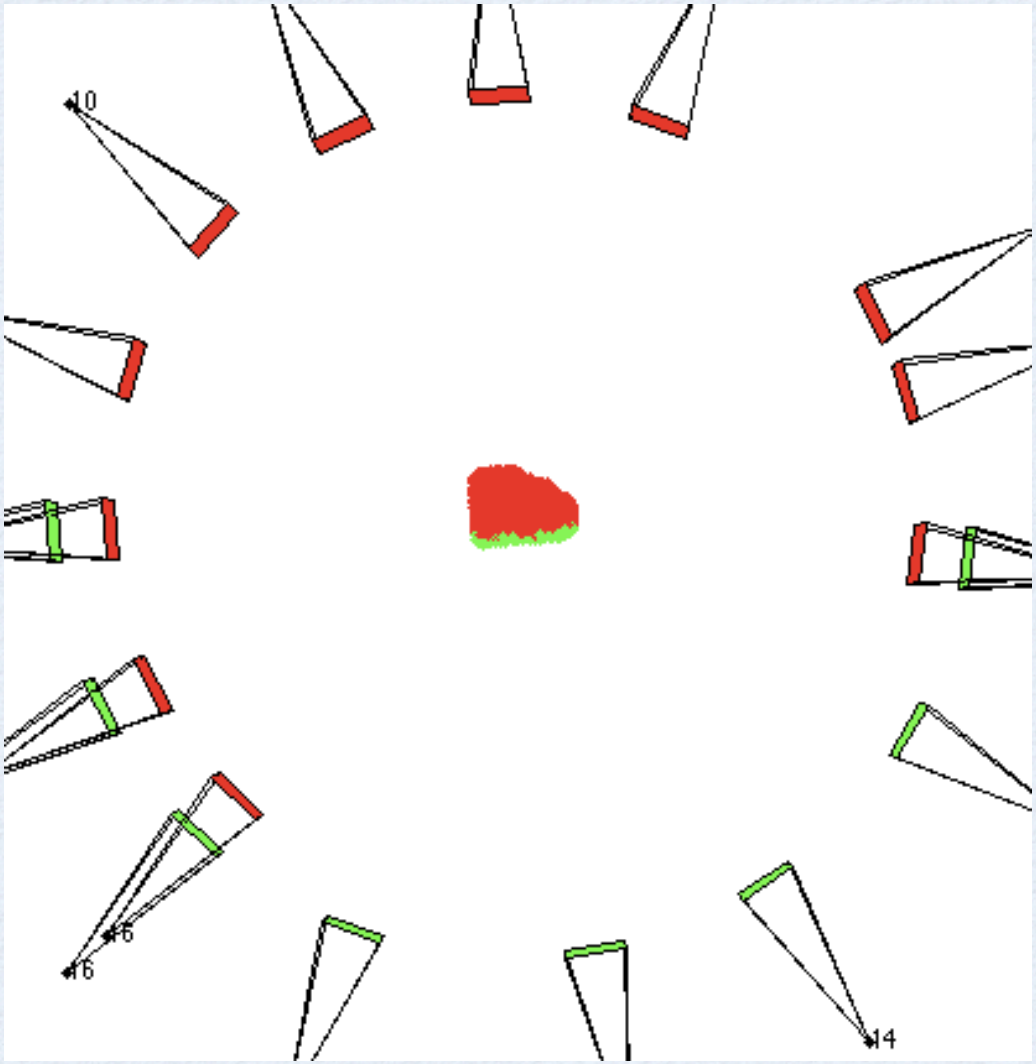


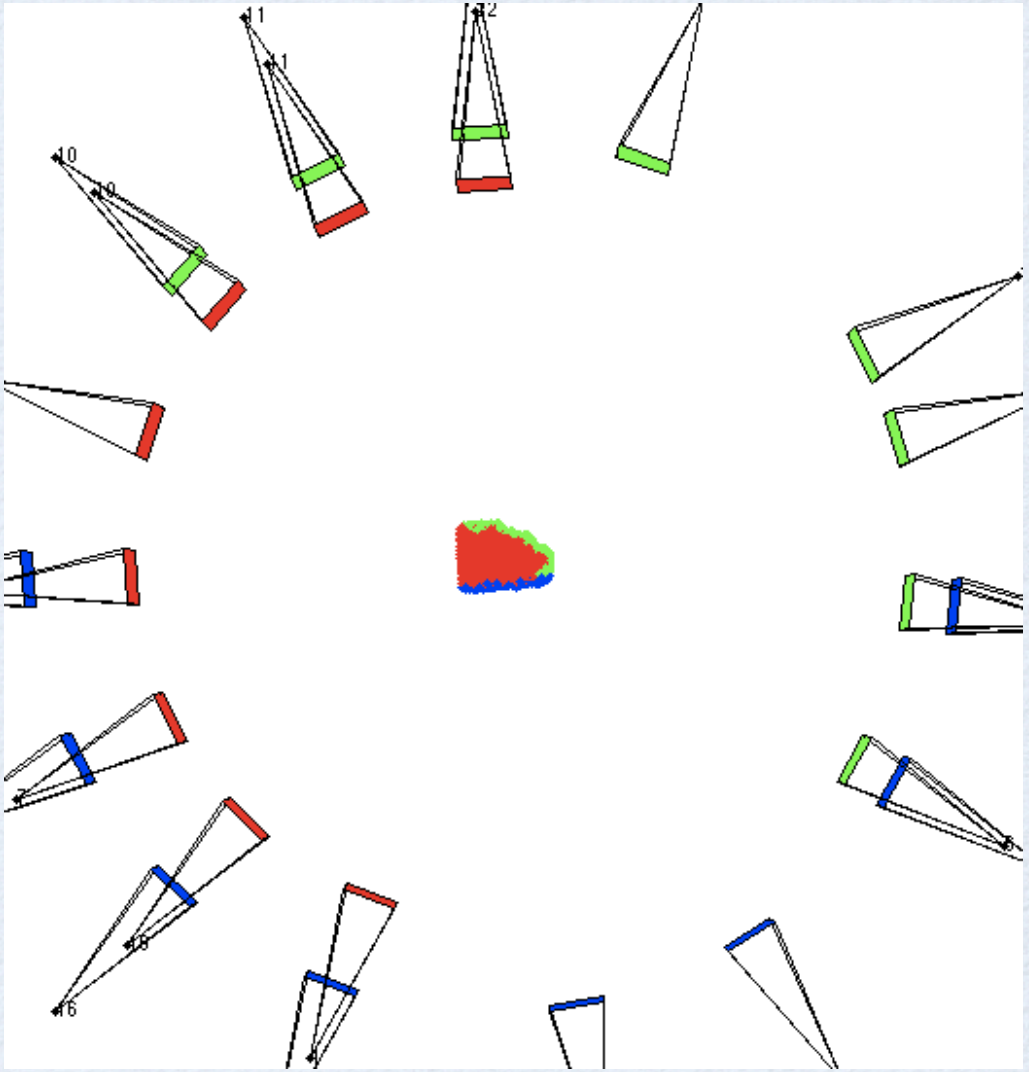
GEOMETRY-BASED CLUSTERS



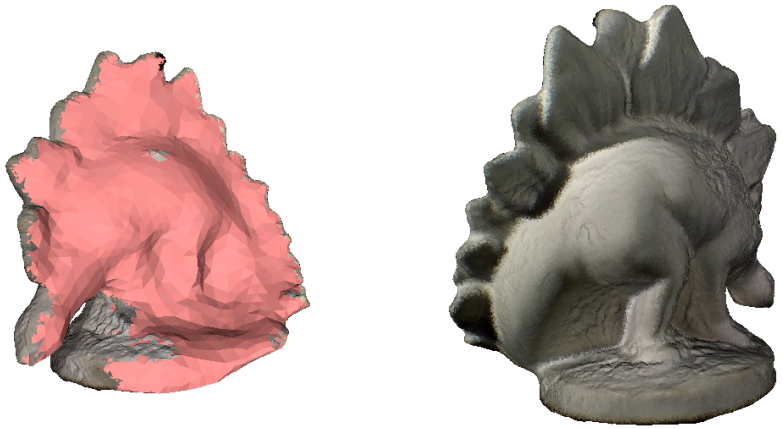


GEOMETRY-BASED CLUSTERS



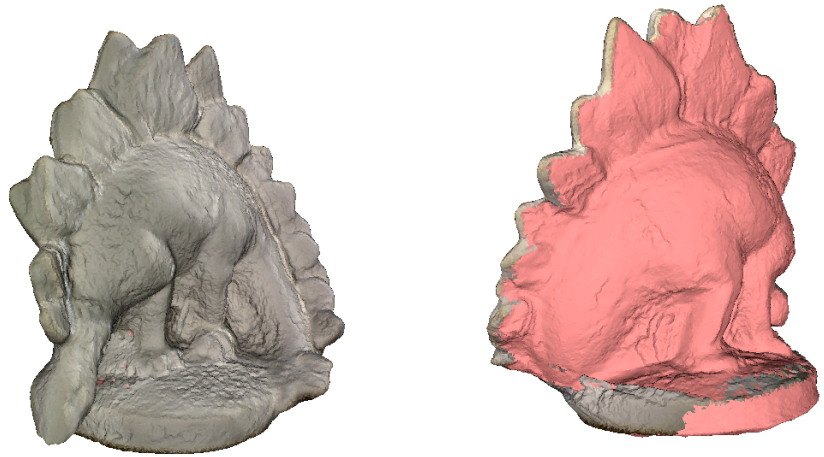


1st cluster - Temple



1st cluster - Dino

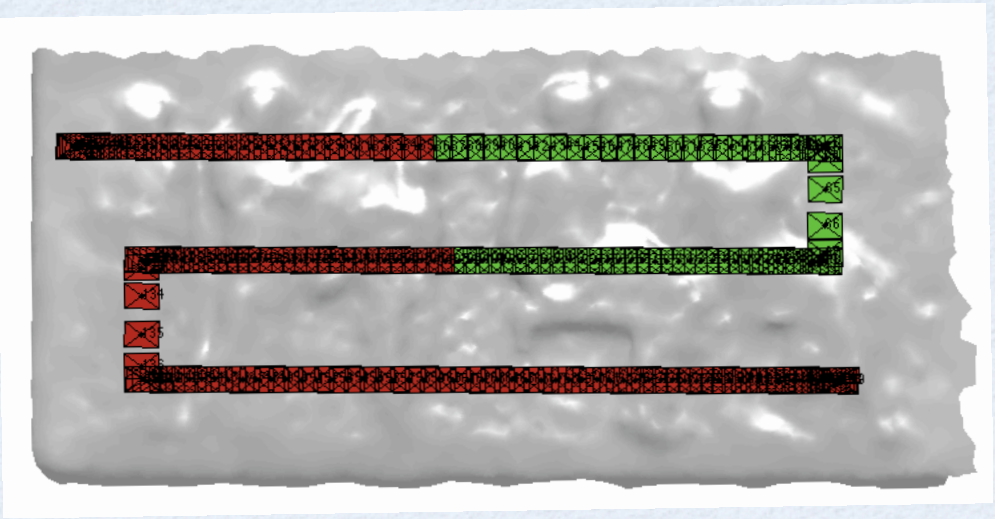
2nd cluster - Temple



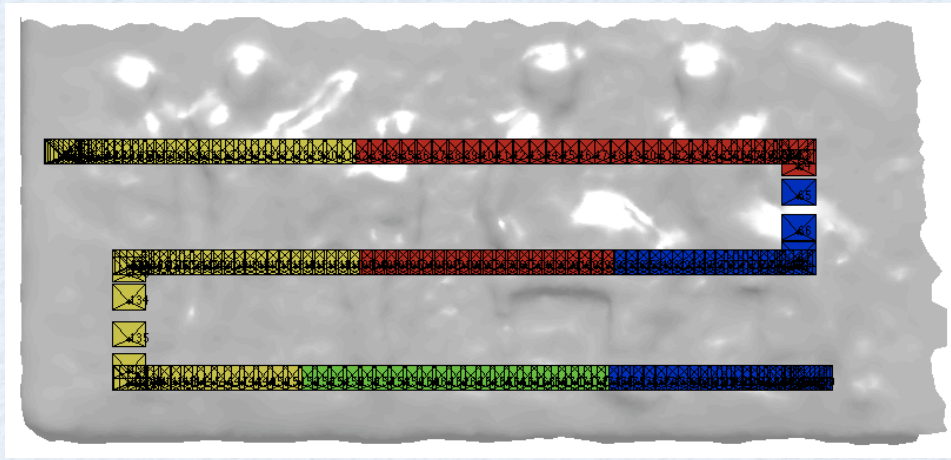
2nd cluster - Dino

Paper \ Dataset	Temple Ring				Dino Ring			
	Acc.	Compl.	Mem.	Time	Acc.	Compl.	Mem.	Time
Zaharescu et al [169]	0.55mm	99.2%	1031MB	60min	0.42mm	98.6%	962MB	43min
Our method - cluster 1	0.62mm	98.5%	468MB	36 min	0.5mm	98.5%	483MB	33min
Our method - cluster 2			472MB	42 min			476MB	35min

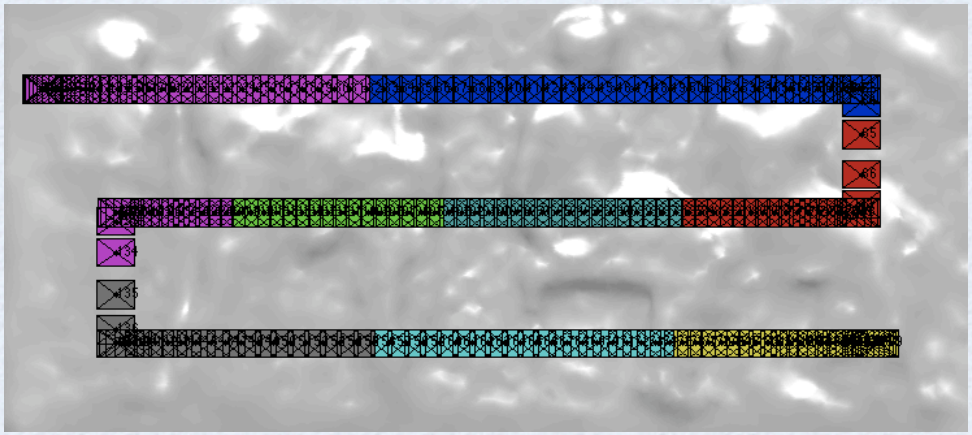
PARTHENON - 200 CAMERAS



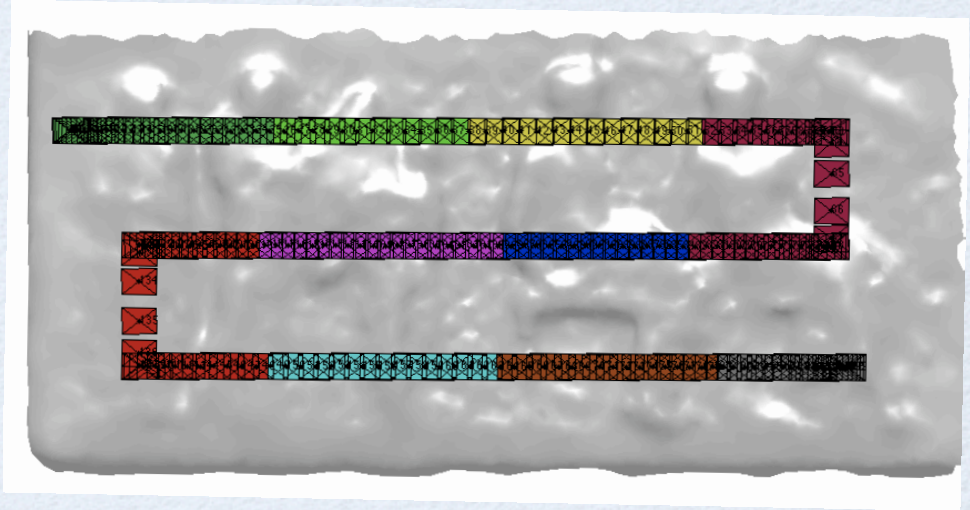
2 clusters



4 clusters

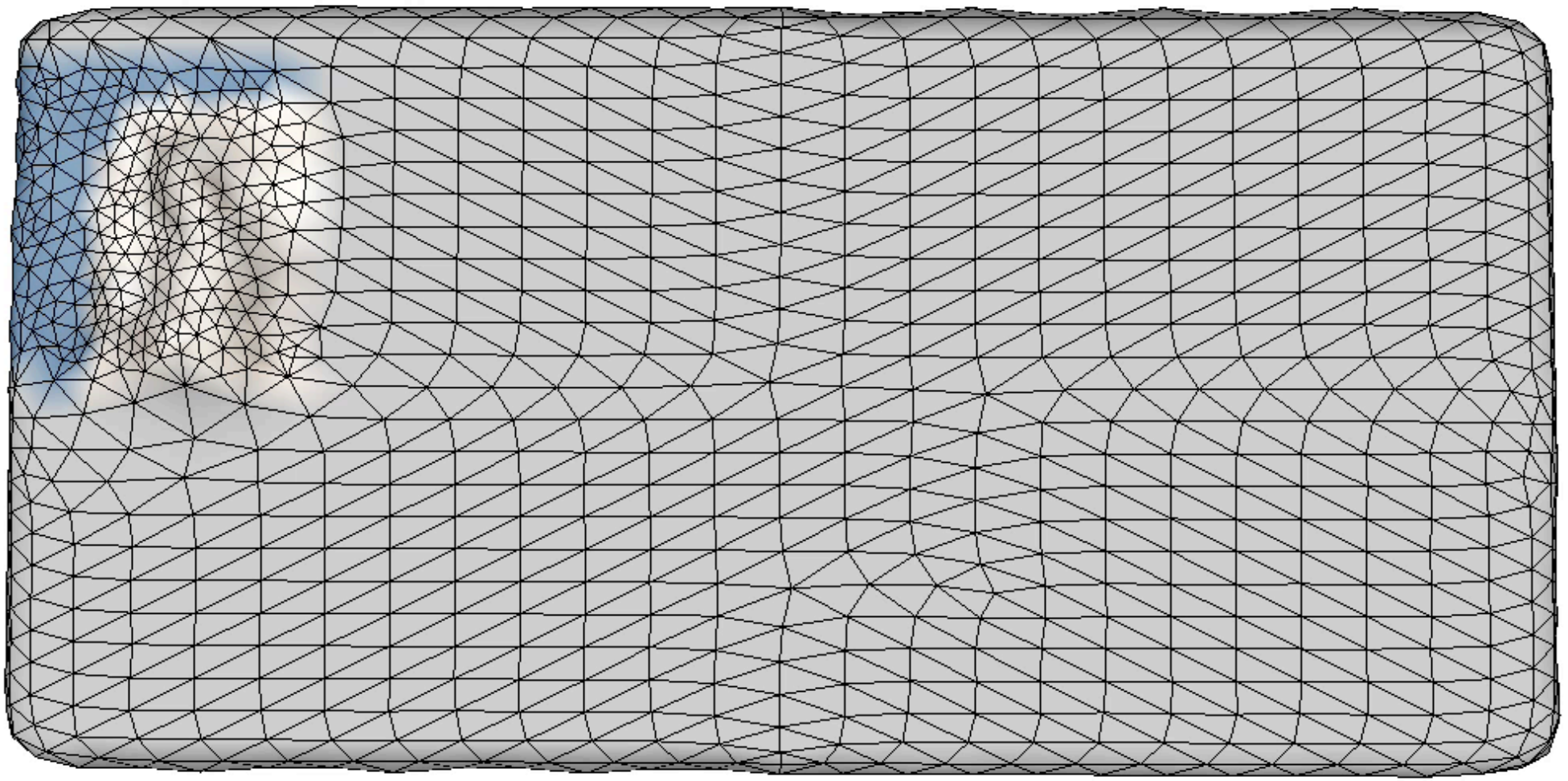


8 clusters



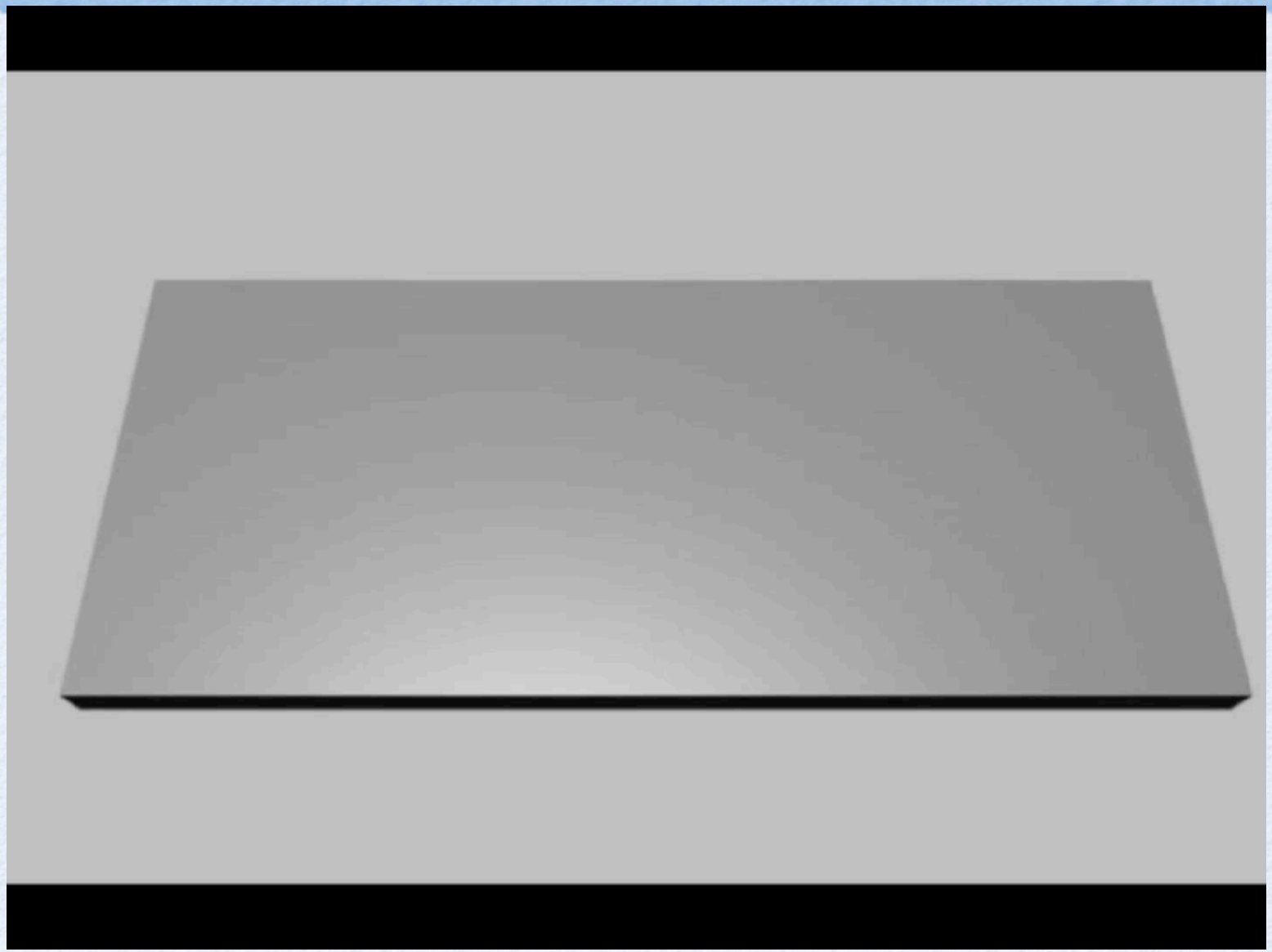
10 clusters

PARTHENON - 200 CAMERAS

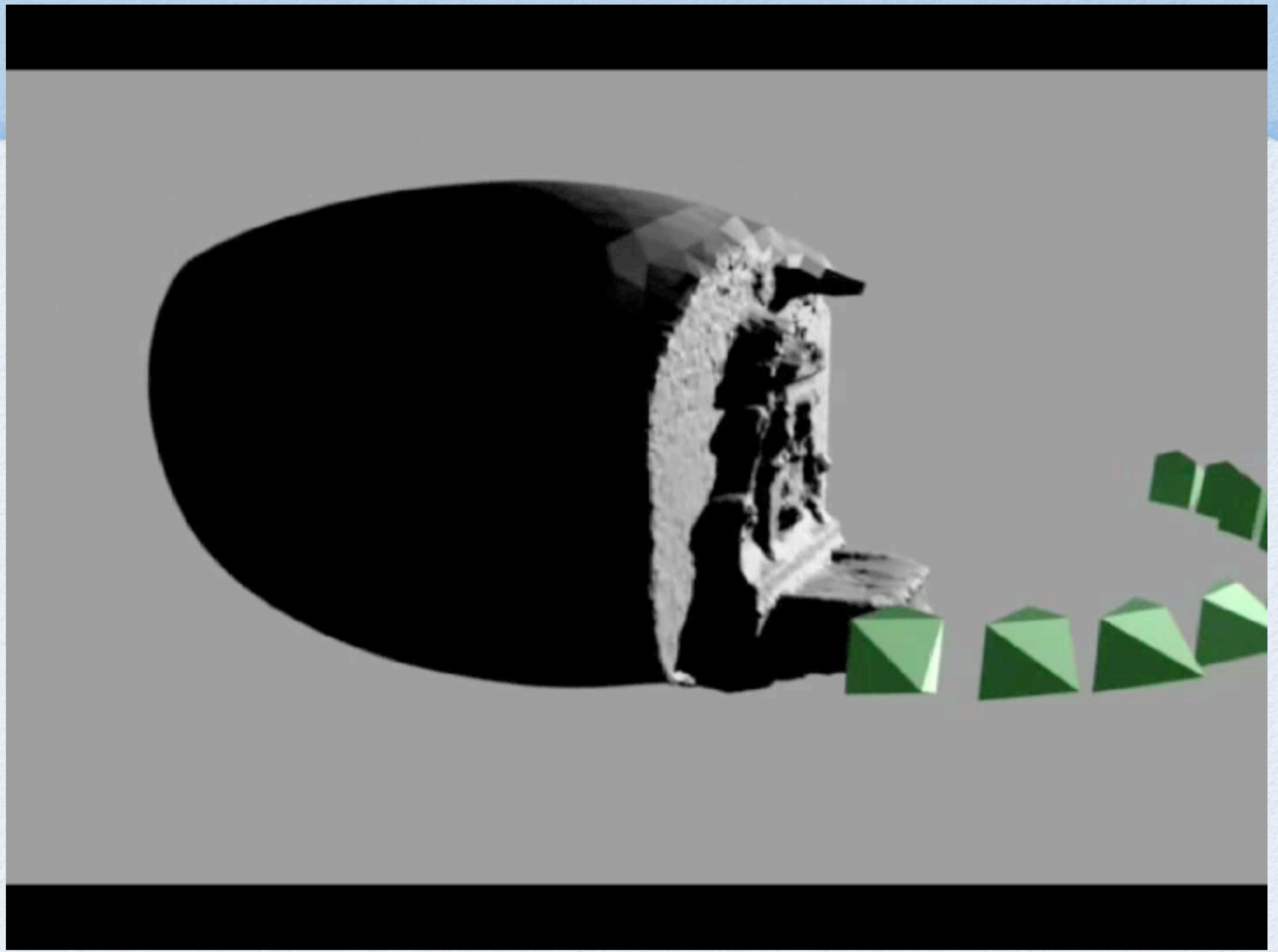


1st Level Resolution with 20 camera clusters

PARTHENON - 200 CAMERAS

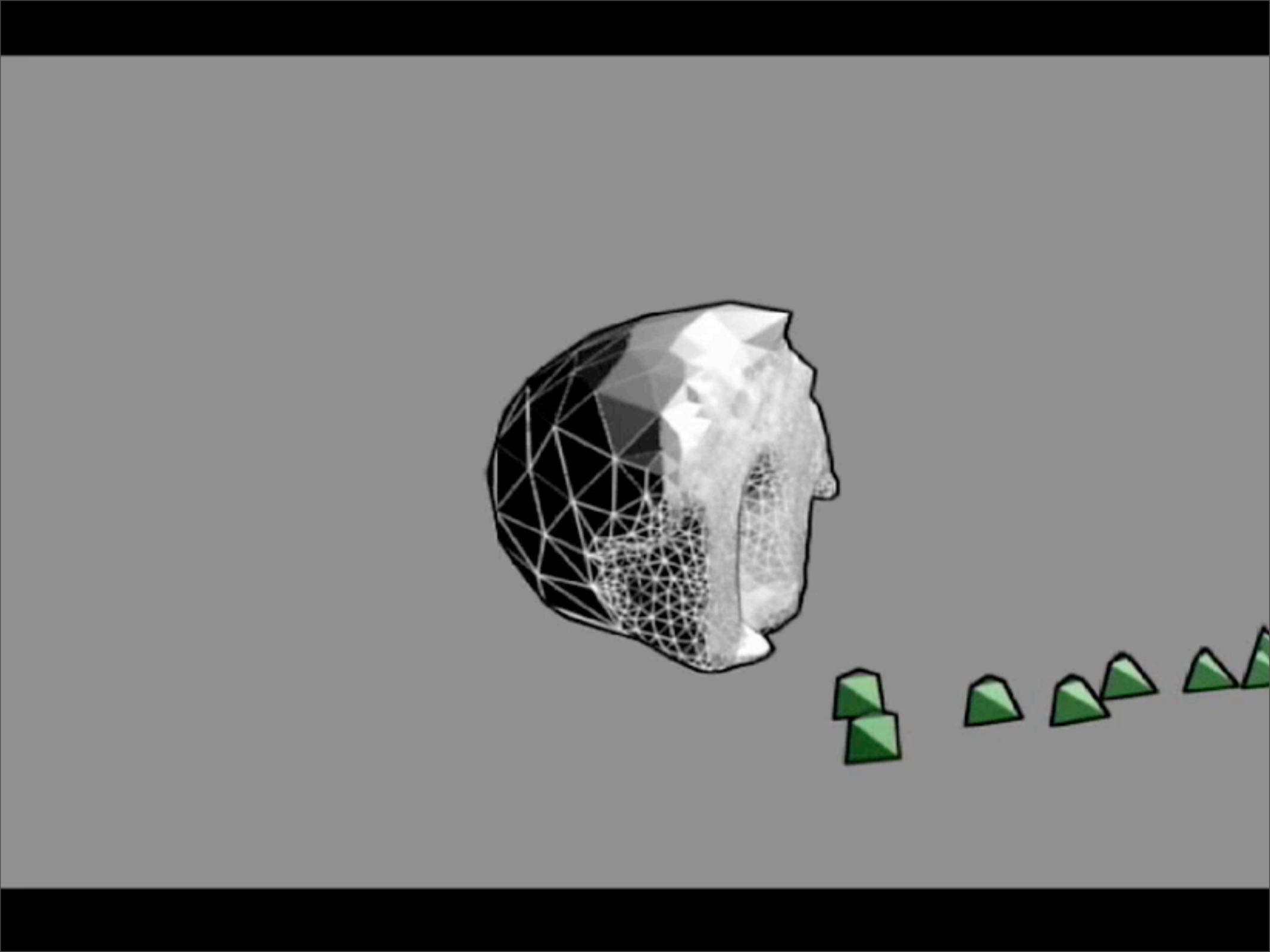


2nd Level Resolution with 20 camera clusters



MORE VIRTUAL CAMERAS

- User guided adaptive resolution
- Only a region of interest will be reconstructed in higher resolution
- Virtual cameras will be generated based on the interactive user selection of the region of interest



4. A 3-D MESH DETECTOR &DESCRIPTOR

[Zaharescu & al - submitted to CVPR 2009]

INTRODUCTION

- **Matching problem**
- **2-D Image Descriptors (photometric)**: SIFT [Lowe IJCV'04], HOG [Dalal et al. CVPR'05], etc
- **Mesh Descriptors (geometric):** SpinImages[Johnson et al, PAMI'92], ShapeContexts[Frome et al ECCV'04], etc

INTRODUCTION

- Recently, an increasingly larger number of coloured meshes available in the vision community (advancements in 3D Rec)

- **Augmented Mesh Descriptors (geometric + photometric)**: VIP (SIFT + normals on meshes [Wu et al - CVPR2008]), [Stark et al, ICCV2007]

MOTIVATION

A **photometric+geometric mesh detector & descriptor** would be **better than an image-based one**:

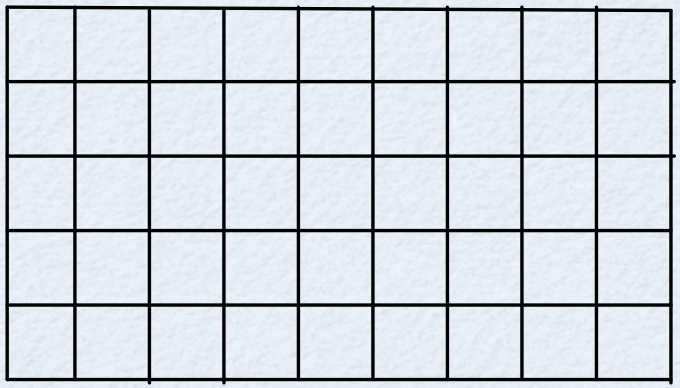
- *no false contour edges*
- *no perspective distortions*
- *meshes incorporate geometric information*

A recent number of **dense mesh tracking** approaches **use back-projected image features** as an initial set of matches - [Varanasi, Zaharescu & al, ECCV08], [Ahmed. et al, CVPR08], [deAguiar et al - HM-UMCA07]

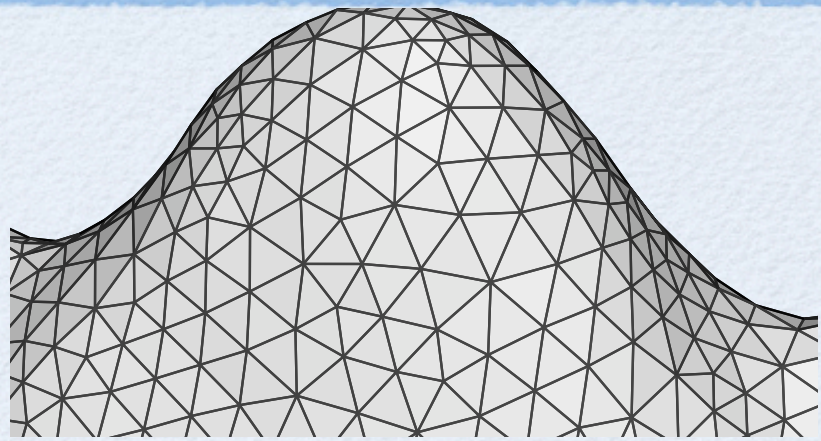
IDEA

- Extend image descriptors to 2-manifolds (meshes)
- We propose:
  - MeshDoG(Mesh Difference of Gaussians) interest point detector
  - MeshHOG(Histogram of Gradient) descriptor

REQUIREMENTS



regular 2D grid



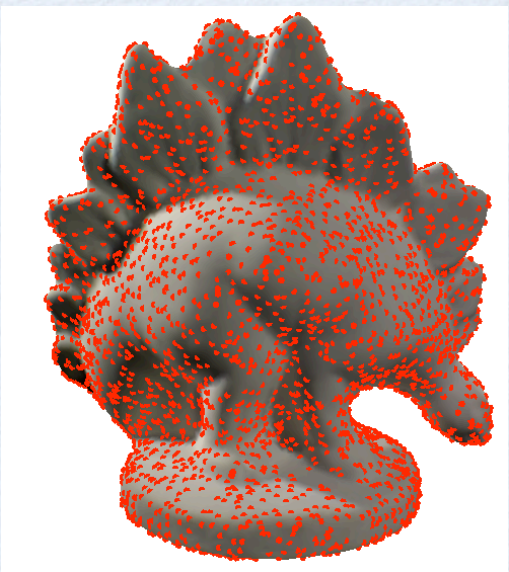
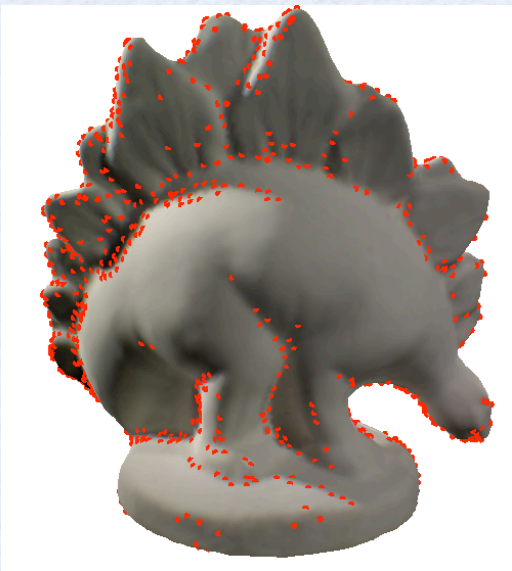
2D manifold

- We need equivalent image based **operators for scalar functions on 2D manifolds**: directional gradient, gradient, convolution
- The scalar function on 2D manifold can be of any type, e.g. colour, curvature, geodesic intregral.

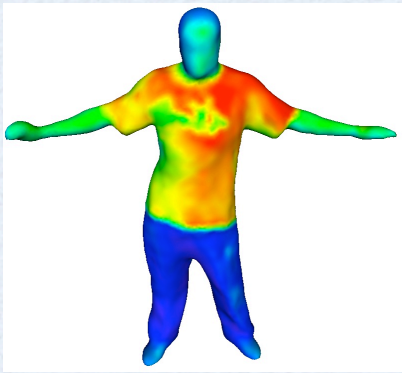
FEATURE DETECTOR

MeshDoG(Difference of Gaussian):

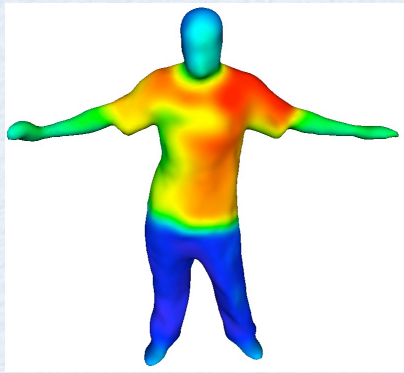
- extrema of the scale space (use convolutions with Gaussian kernel)
- thresholding
- non stable feature elimination (corner detection)

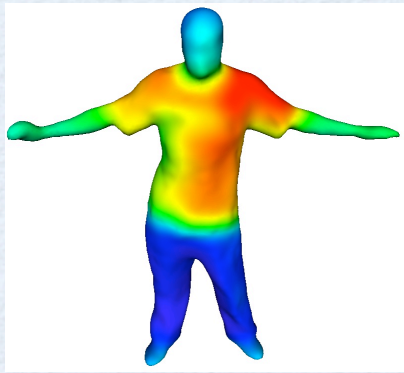

COLOUR SPACE



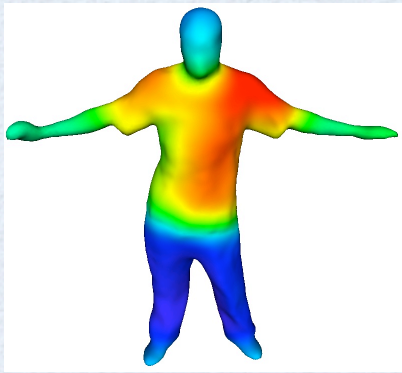
(a) f_2



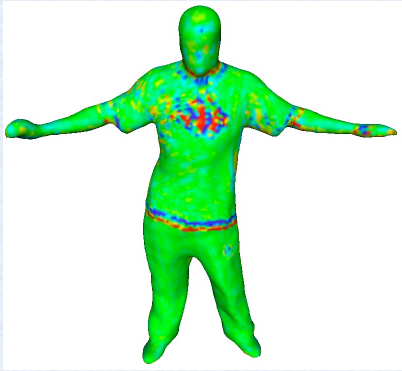
(b) f_{16}



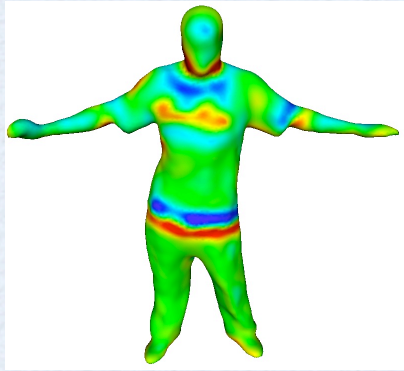
(c) f_{32}



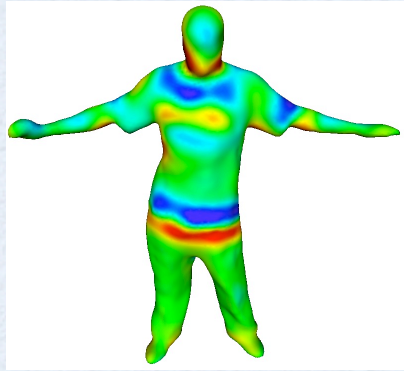
(d) f_{64}



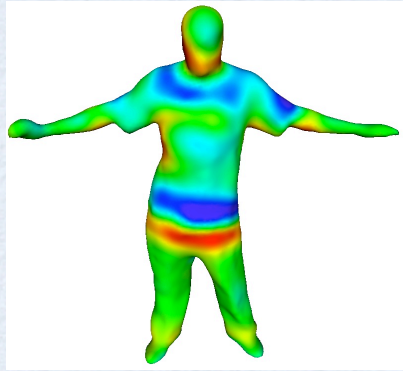
(e) DOG_2



(f) DOG_{16}



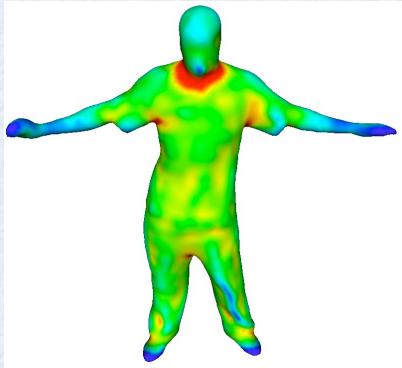
(g) DOG_{32}



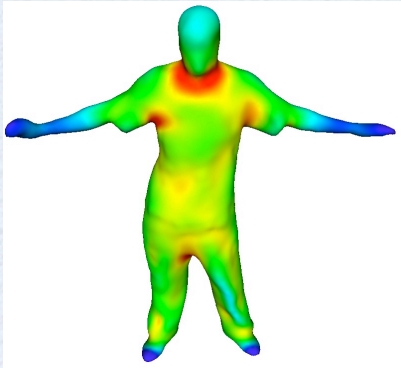
(h) DOG_{64}

f represents colour

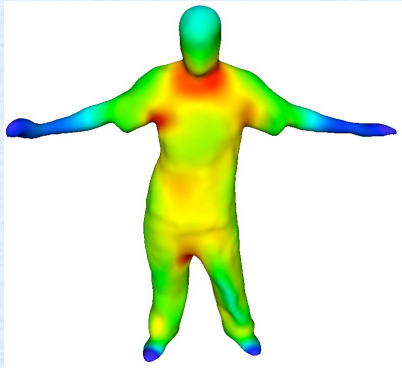
CURVATURE SPACE



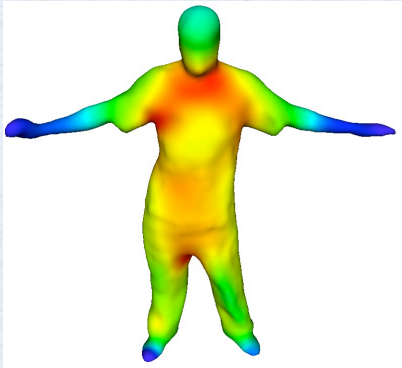
(a) f_2



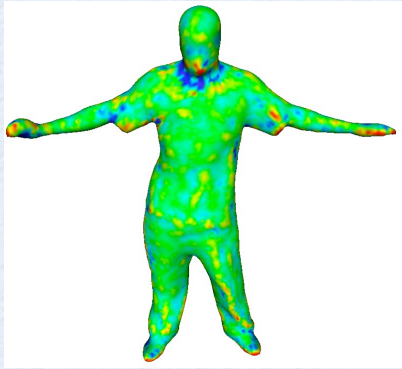
(b) f_{16}



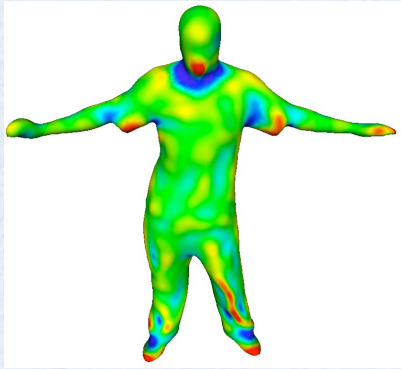
(c) f_{32}



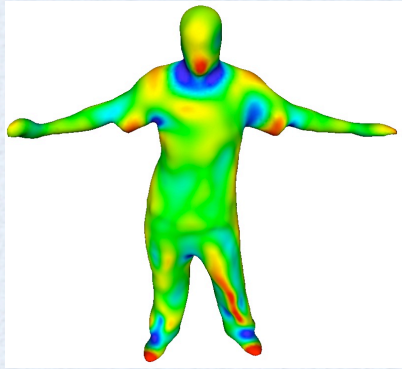
(d) f_{64}



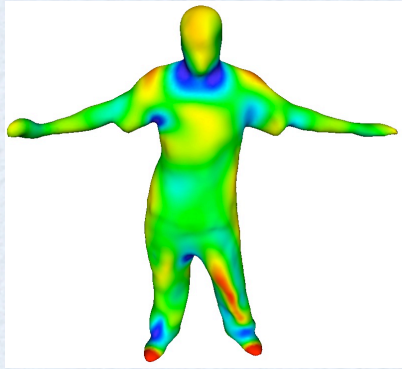
(e) DOG_2



(f) DOG_{16}



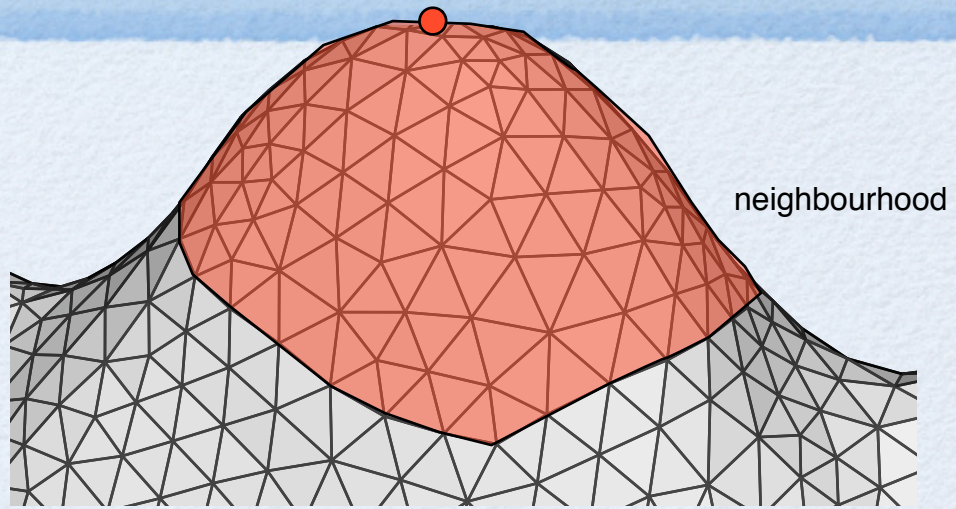
(g) DOG_{32}



(h) DOG_{64}

f represents mean curvature

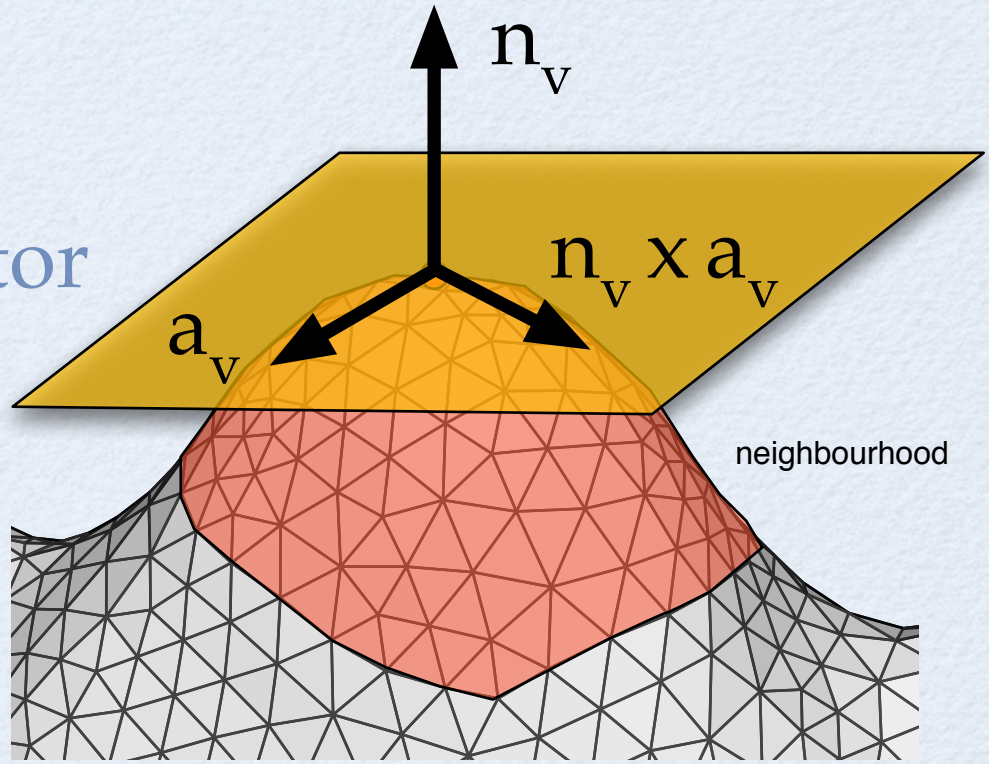
FEATURE DESCRIPTOR



- Around a support region (neighbourhood)
- The neighbourhood size is chosen such that it represents % of the total area of the surface
- Histogram of Gradients
- The gradients are 3-D vectors, so the histograms are 3-D

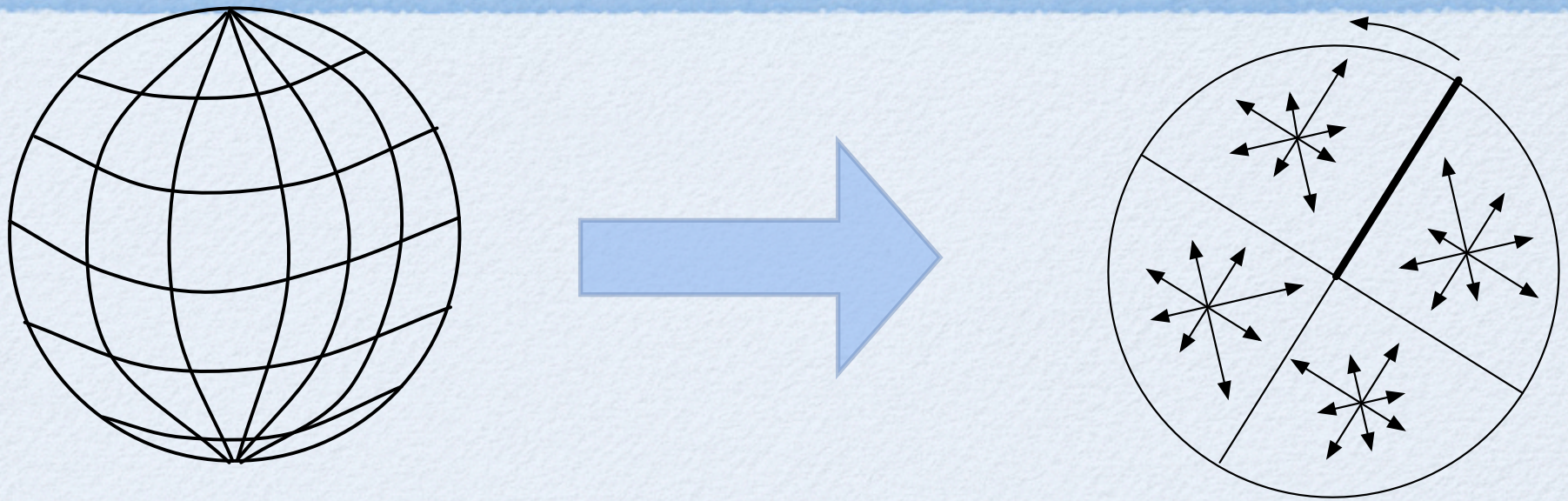
LOCAL COORDINATE SYSTEM

- We need a local coordinate system to make the descriptor **rotation invariant** !



- Project all participating vertices in the tangent plane;
- 2-D polar histogram (b_d=36) of the gradient magnitude;
- Dominant bin chosen as axis a_v.

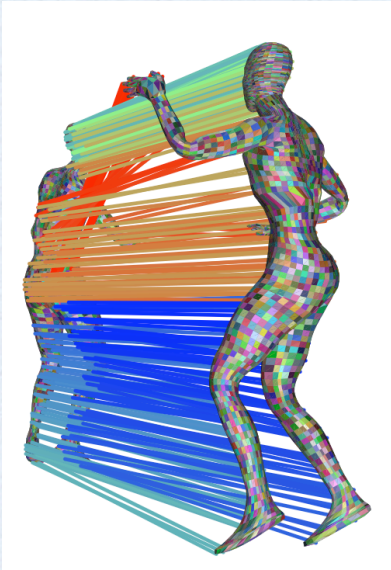
DESCRIPTOR

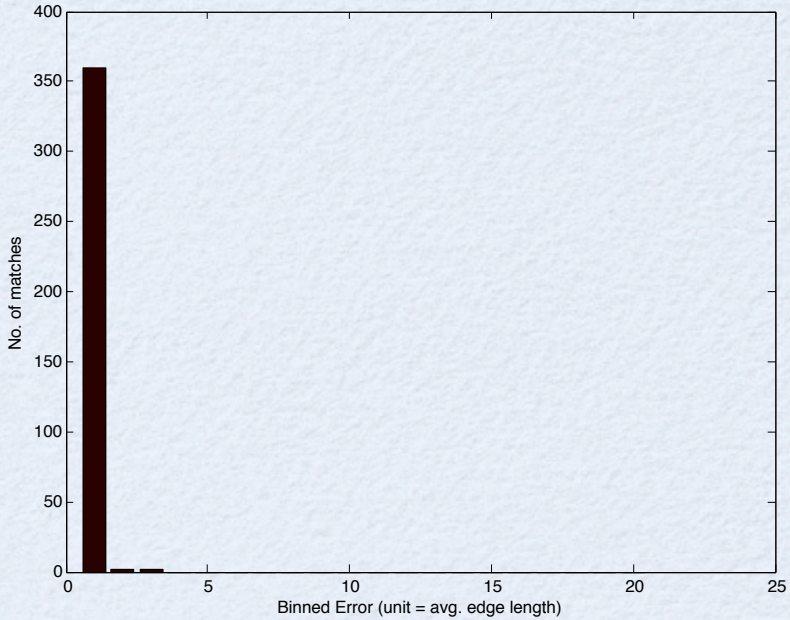


- Two level histogram:
  - 1st level: spatial support (b_s=4)
  - 2nd level: orientation support (b_o=8)

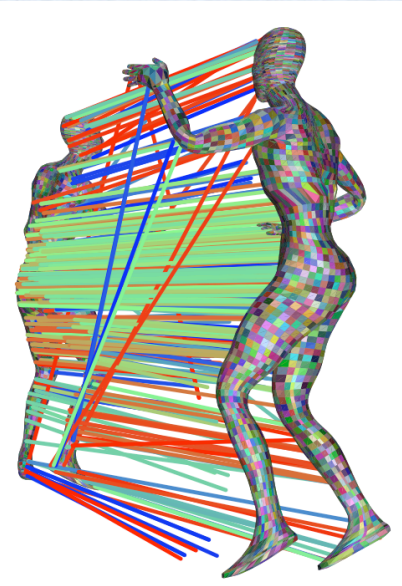
RESULTS

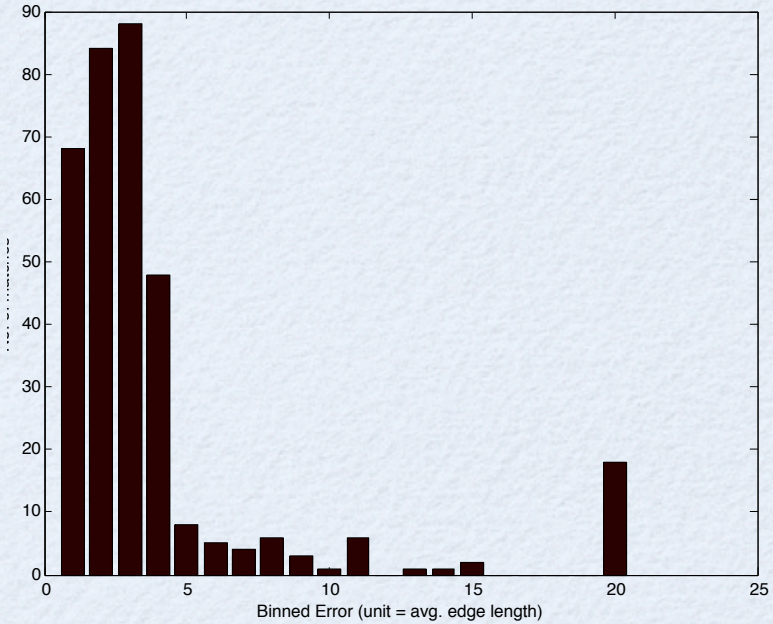
MeshHOG



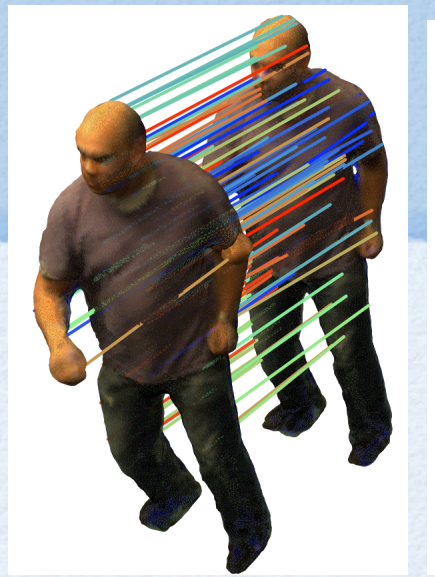


Backprojected SURF



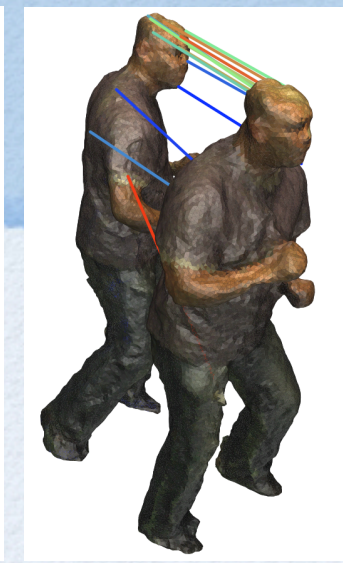


MeshHOG



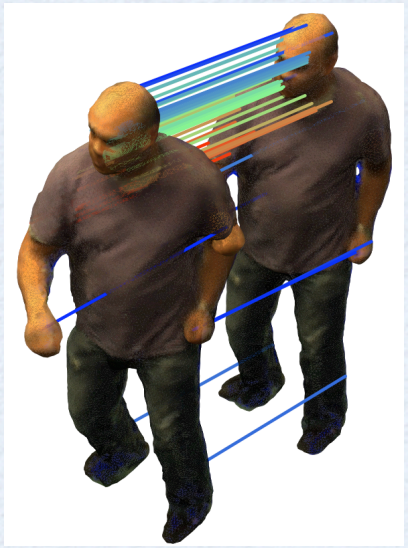
(a) Frames 525-526 MeshHOG (119 matches)

(b) Frames 530-531 MeshHOG (122 matches)

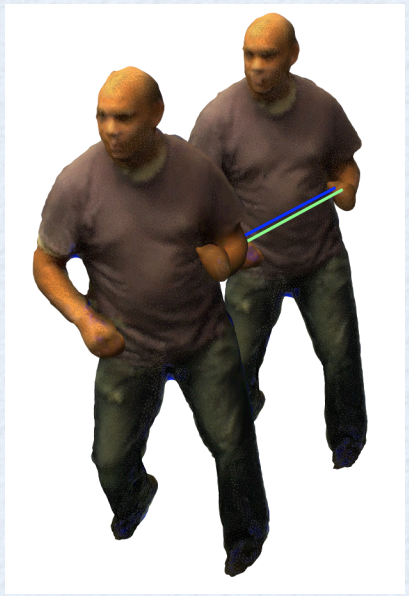


(c) Frames 530-550 MeshHOG (13 matches)

Backprojected SURF

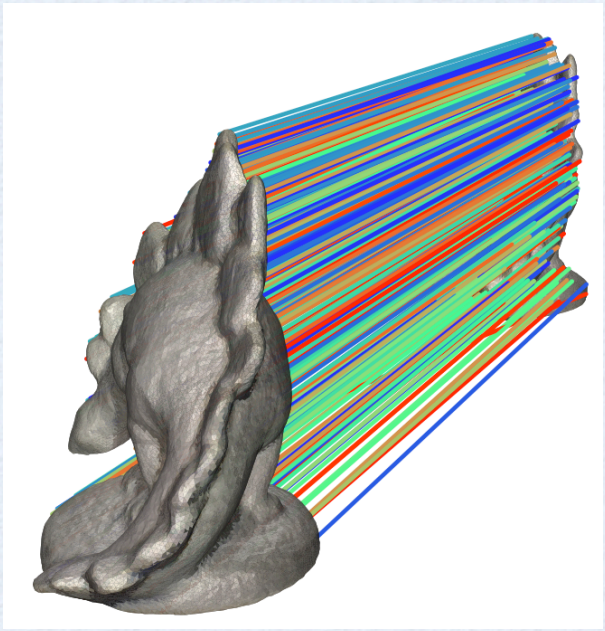


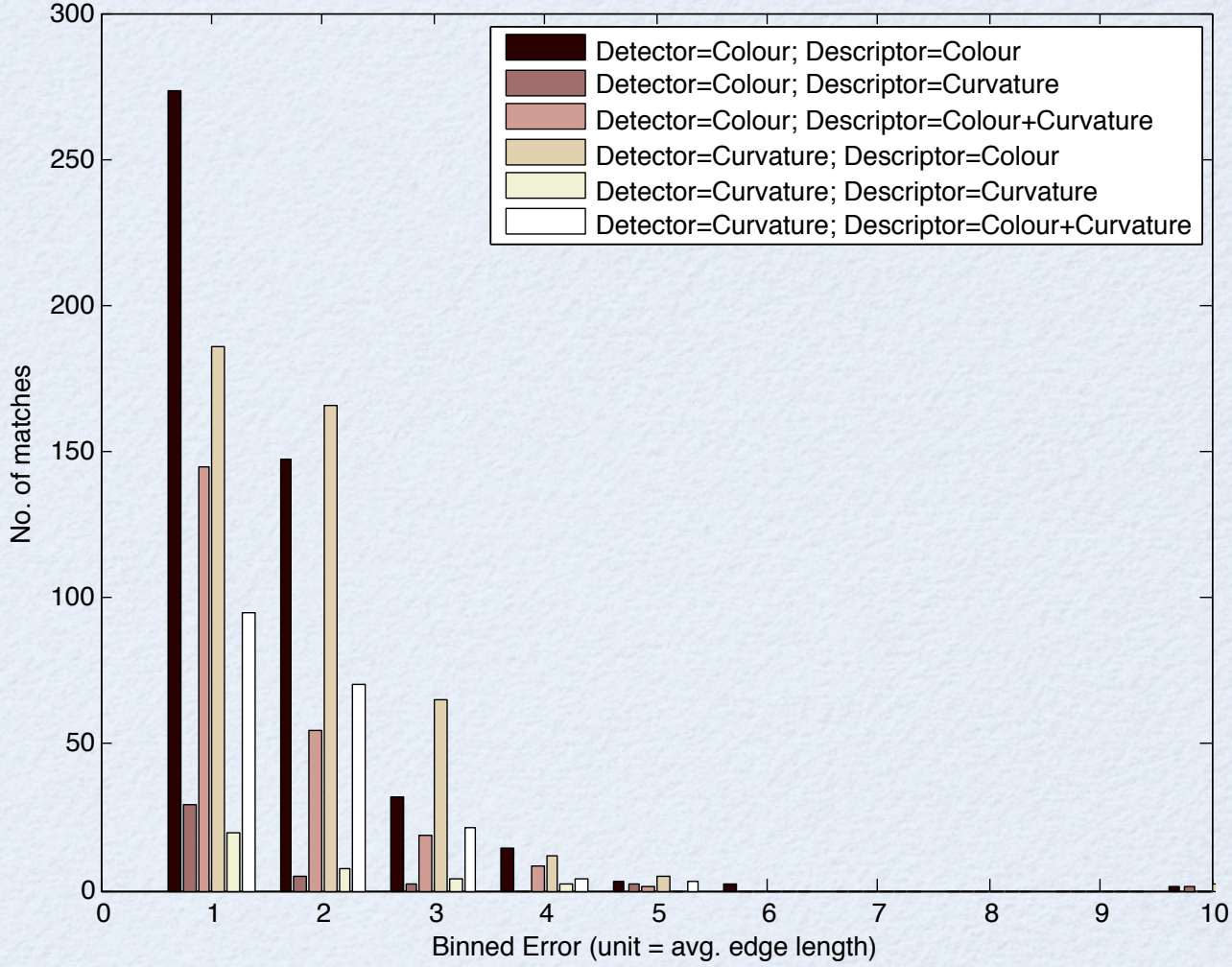
(d) Frames 525-526 SURF (54 matches)



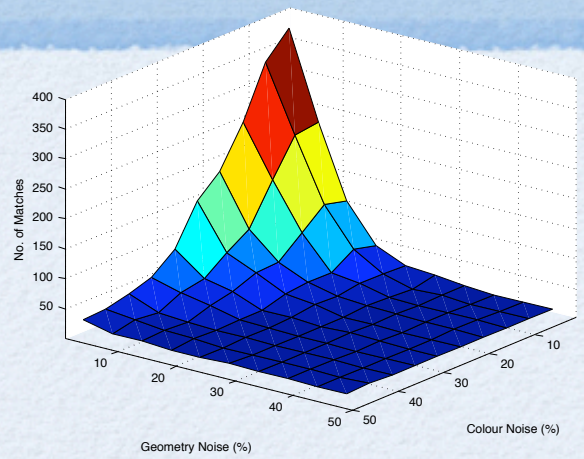
(e) Frames 530-531 SURF (2 matches)

(f) Frames 530-550 SURF (0 matches)

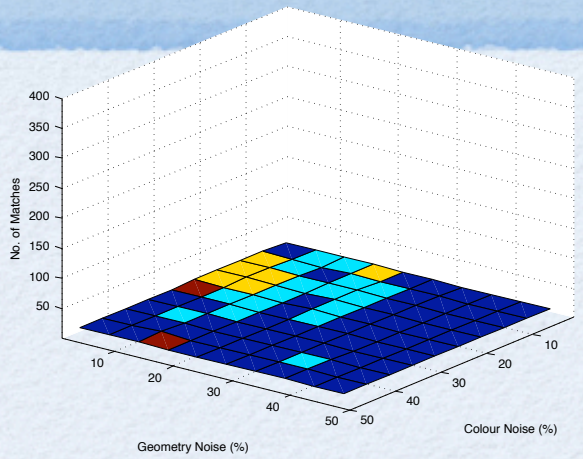



Detector=Colour; Descriptor=Colour
Detector=Colour; Descriptor=Curvature
Detector=Colour; Descriptor=Colour+Curvature
Detector=Curvature; Descriptor=Colour
Detector=Curvature; Descriptor=Curvature
Detector=Curvature; Descriptor=Colour+Curvature
No. of matches
Binned Error (unit = avg. edge length)
0
50
100
150
200
250
300
0
1
2
3
4
5
6
7
8
9
10

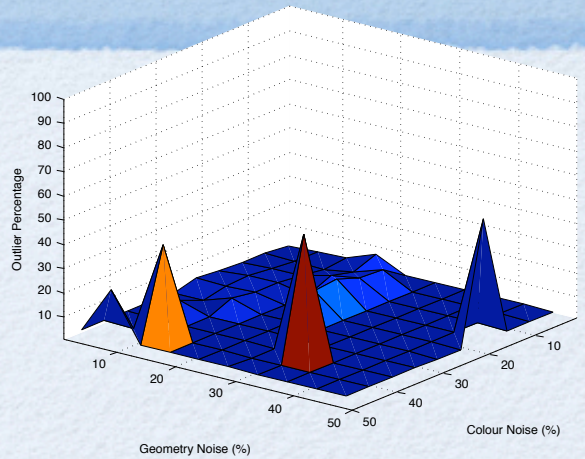
RESILIENCE TO NOISE



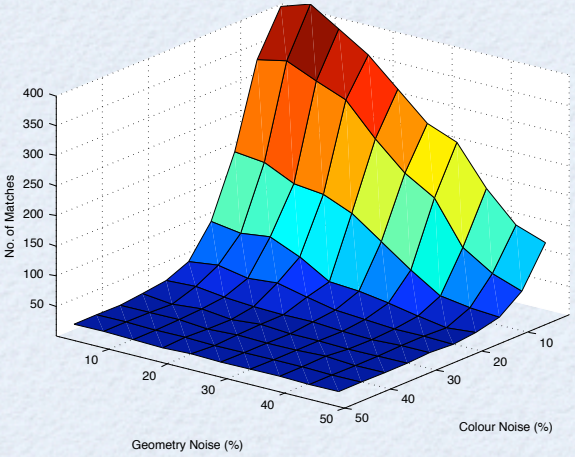
(a) Synth - TP



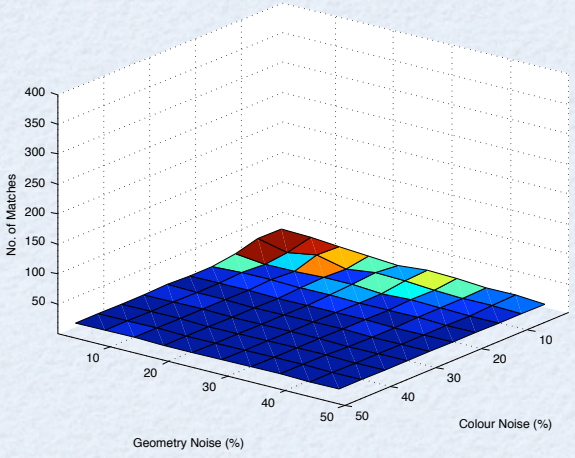
(b) Synth - FP



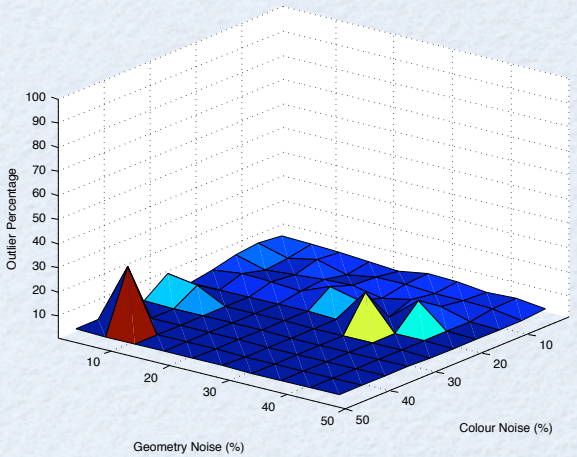
(c) Synth - FP Ratio



(d) Dino - TP



(e) Dino - FP



(f) Dino - FP Ratio

CONCLUSION

Goals: improve 3-D reconstruction methods

Contributions

- A Robust Factorization Framework
- A Mesh Surface Evolution Algorithm
- Content-Aware Camera Clustering
- A Novel 3-D Mesh Detector & Descriptor

FUTURE WORK

Short term:

- Look into parallelising camera clustering
- Use MeshHOG for object recognition
- Look into porting other results form images to manifolds (i.e. MSER, optical flow)

Long term:

- Unified Spatio-Temporal formulation

THAT'S ALL FOLKS!