北京航空航天大學
BEIHANG UNIVERSITY

Ghost Mesh: Cloud-Based Interactive Mesh Editing

Zhong Zhou, Fei Dou, Yi Li, Lin Zhang

State Key Laboratory of Virtual Reality Technology and Systems Beihang University, Beijing, China

Outline

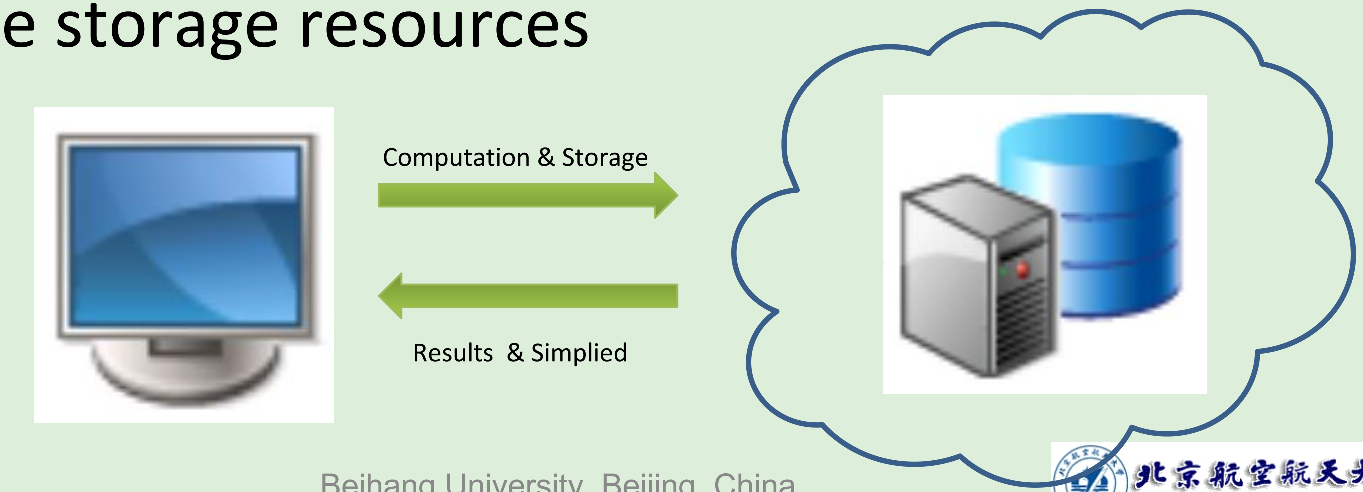
- Ghost Mesh System Pipeline
- User Interface
- Algorithms
- Evaluation
- Editing Examples

Outline

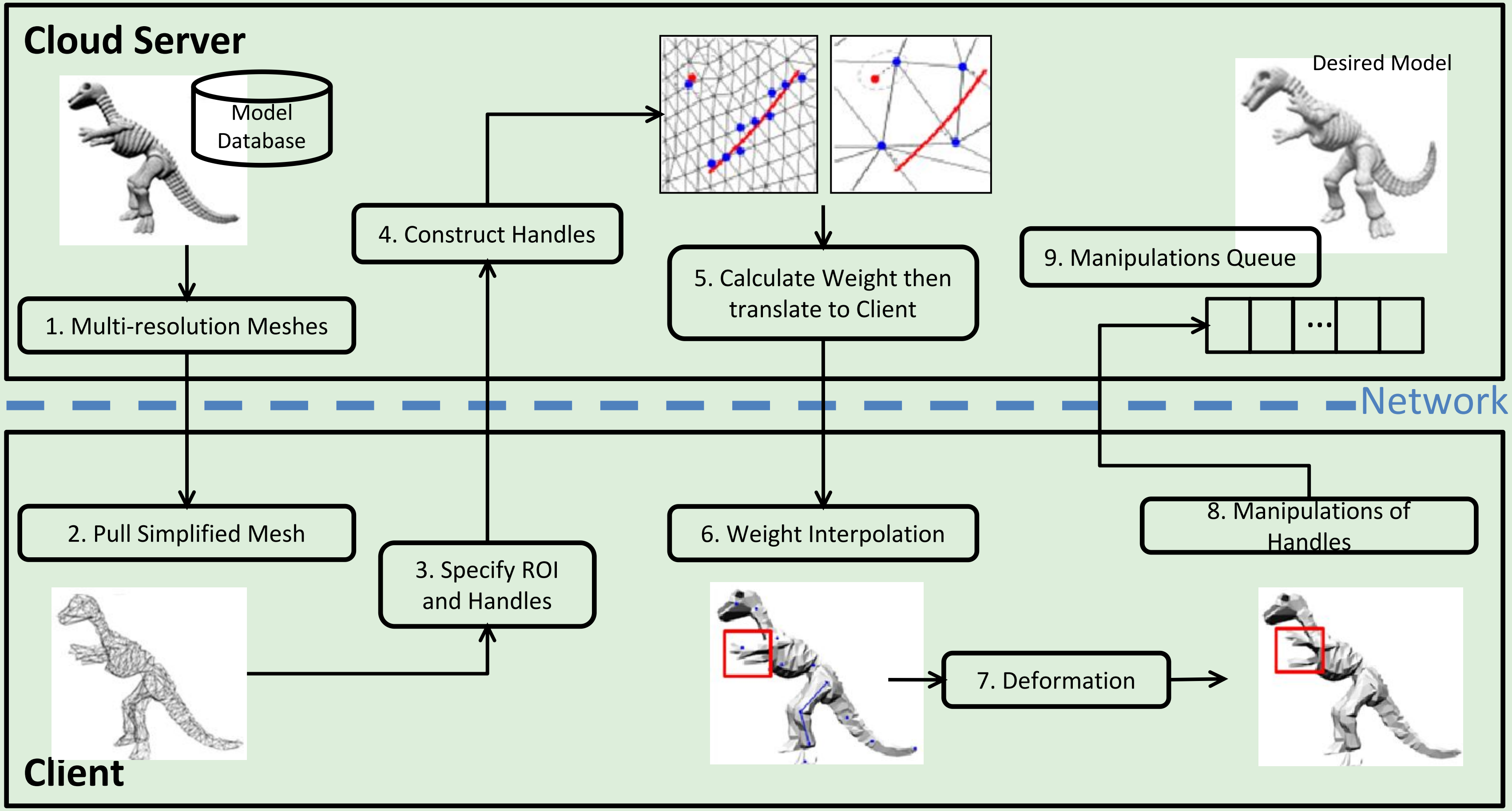
- Ghost Mesh System Pipeline
- User Interface
- Algorithms
- Evaluation
- Editing Examples

Cloud Based

- Effective interactive deformation of complex objects requires:
  - High computation capability
  - Large storage resources
- So …



System Pipeline



Cloud Server

Model Database

Desired Model

1. Multi-resolution Meshes

4. Construct Handles

5. Calculate Weight then translate to Client

9. Manipulations Queue

…

Network

2. Pull Simplified Mesh

3. Specify ROI and Handles

6. Weight Interpolation

8. Manipulations of Handles

7. Deformation

Client

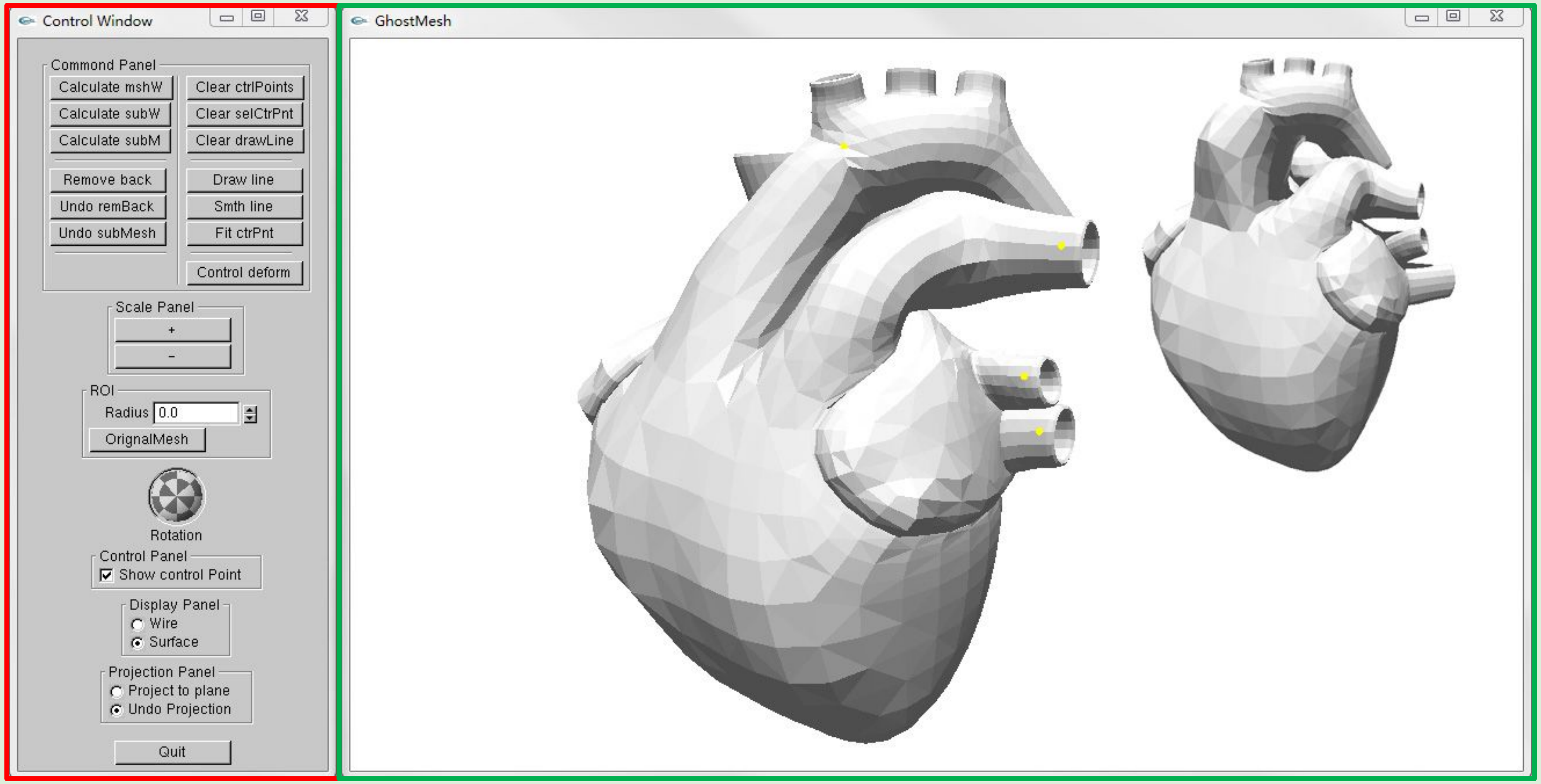
Outline

- Ghost Mesh System Pipeline
- User Interface
- Algorithms
- Evaluation
- Editing Examples

User Interface(1)

Button panel

Rendering window



Radius 0.0

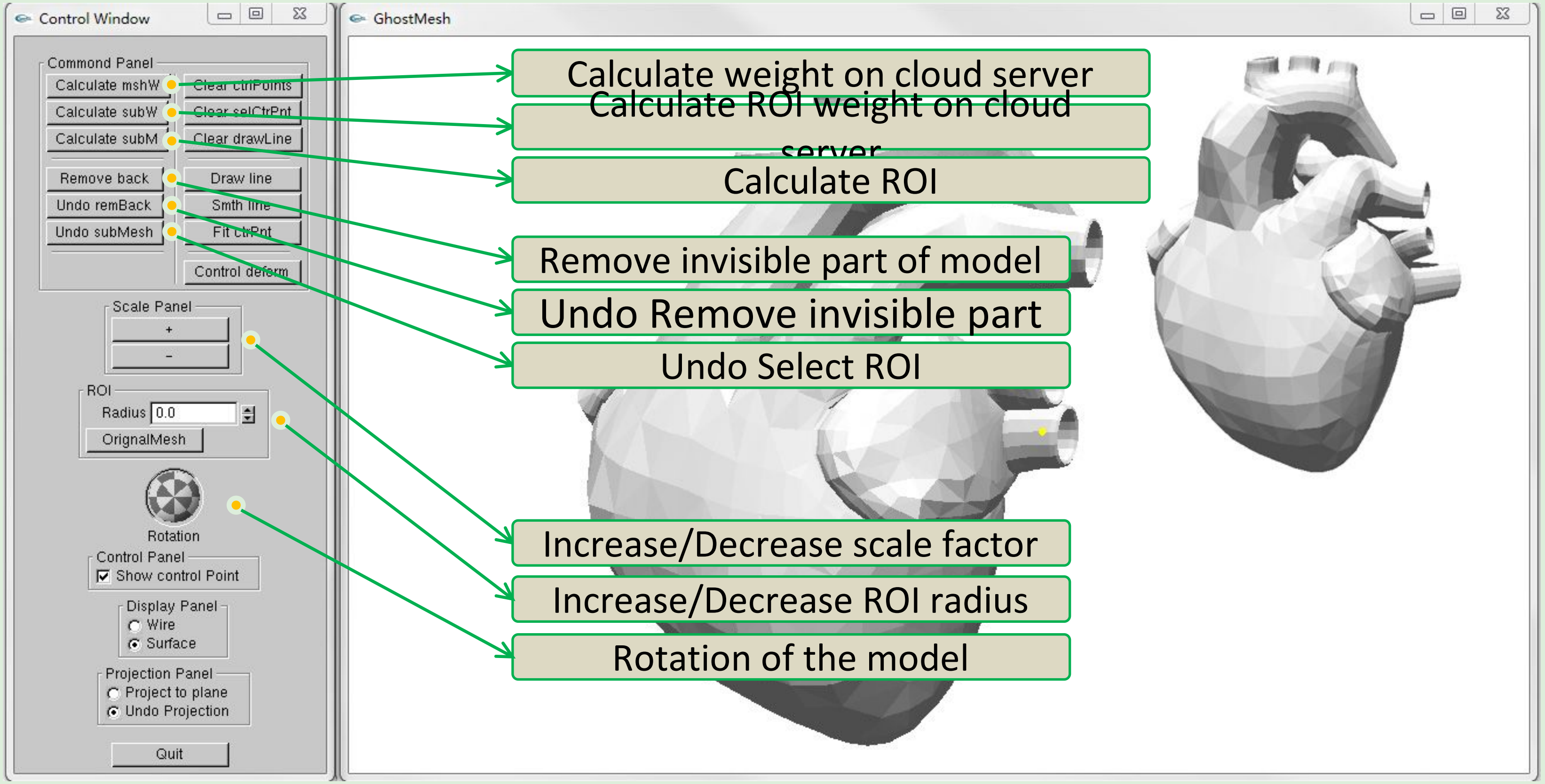
☑ Show control Point

○ Wire
● Surface

○ Project to plane
● Undo Projection

Ghost Mesh user interface

User Interface(2)



Calculate weight on cloud server

Calculate ROI weight on cloud server

Calculate ROI

Remove invisible part of model

Undo Remove invisible part

Undo Select ROI

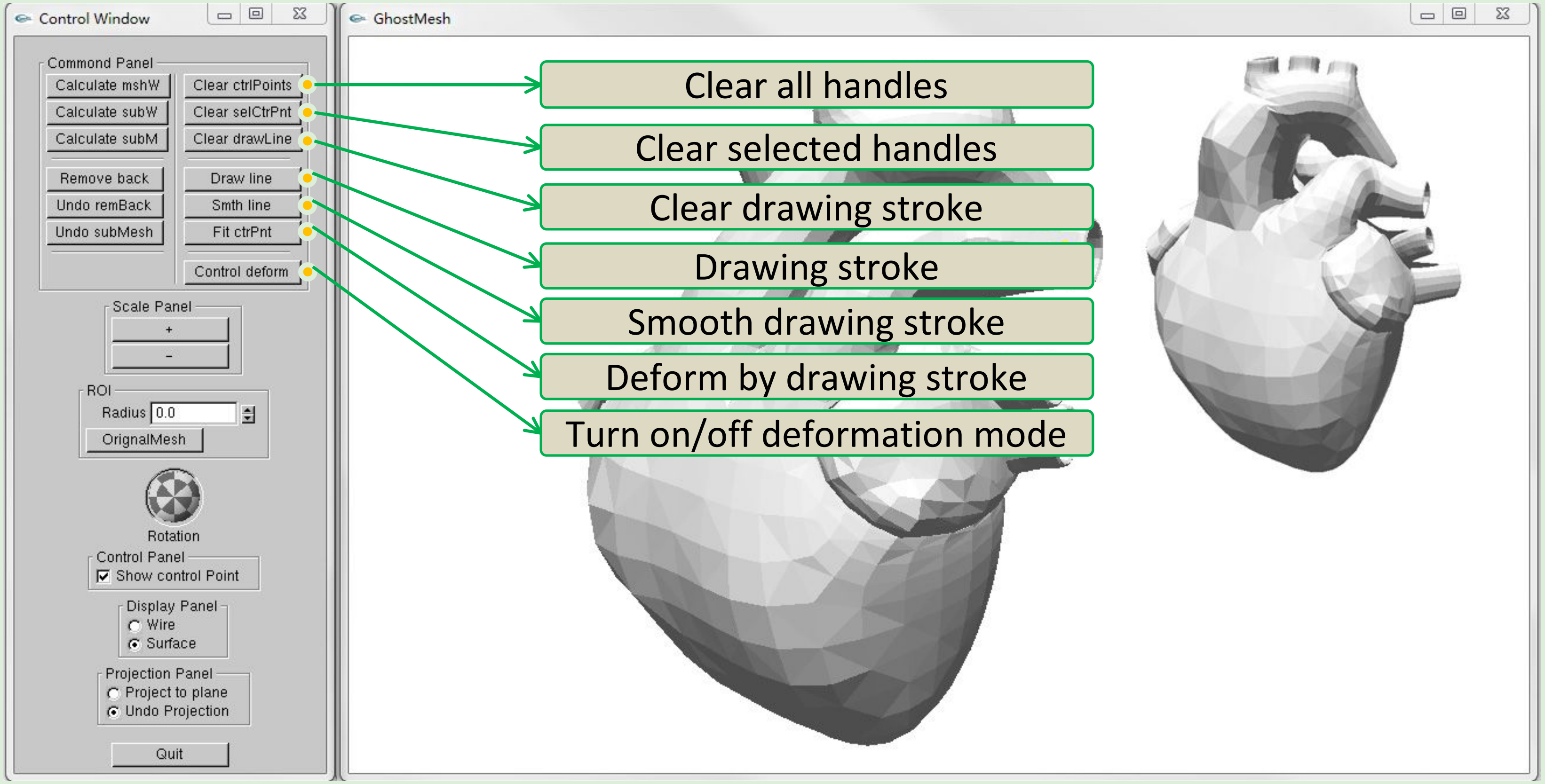
Increase/Decrease scale factor

Increase/Decrease ROI radius

Rotation of the model

Ghost Mesh user interface

User Interface(3)



Clear all handles

Clear selected handles

Clear drawing stroke

Drawing stroke

Smooth drawing stroke

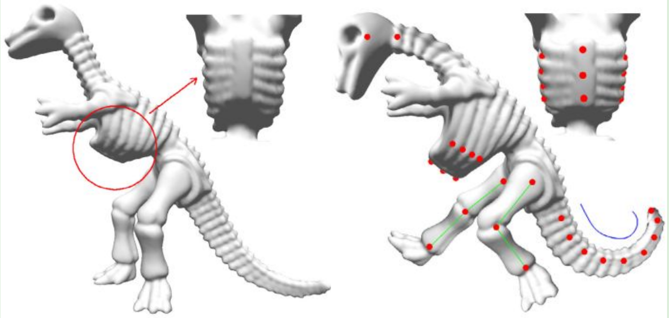
Deform by drawing stroke

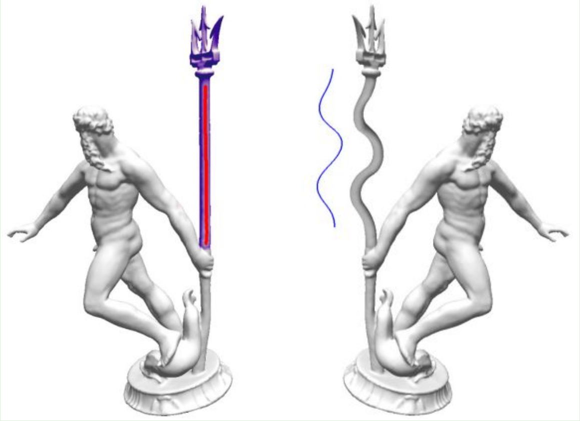
Turn on/off deformation mode

Ghost Mesh user interface

Handle Tool and ROI Tool

- Clicks or drags on the simplified surface to specify a number of handles.





- Point handle
  - Translation, scale, rotation
- Skeleton handle
  - Drag

- Control curve handle
  - Define desired deformation
- ROI tool
  - Specify ROI

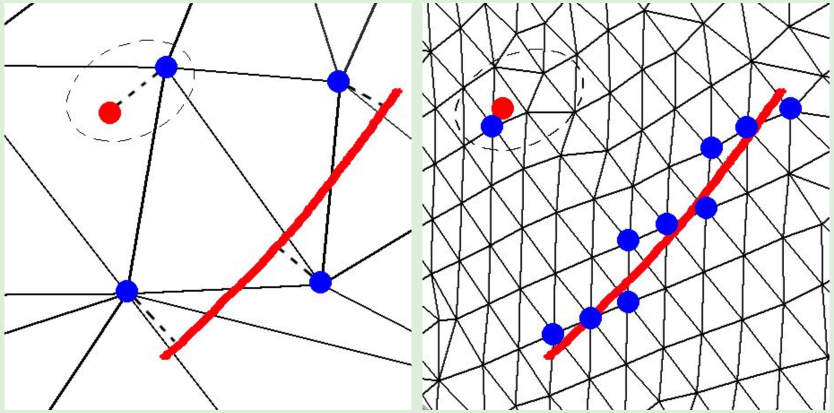
Outline

- Ghost Mesh System Pipeline
- User Interface
- Algorithms
- Evaluation
- Editing Examples

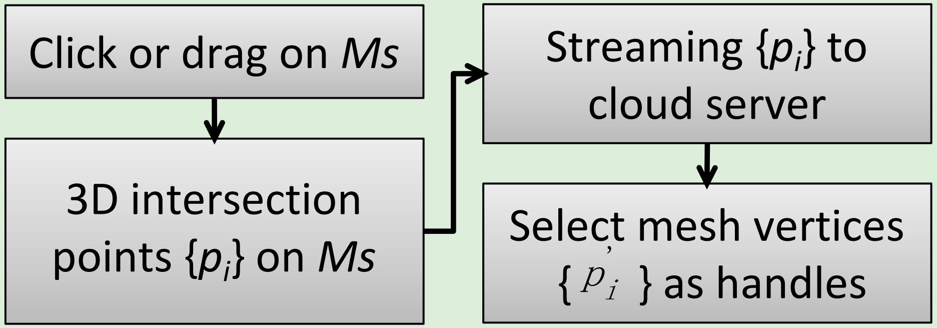
Algorithms

- Handle Selection and Optimization
- Weight Computing
- Data Transmission

Handle Selection and Optimization



Comparison between simplified and original surface



Click or drag on *Ms*

3D intersection points {p_i} on *Ms*

Streaming {p_i} to cloud server

Select mesh vertices {p'_i} as handles

Clien
t

Cloud Server

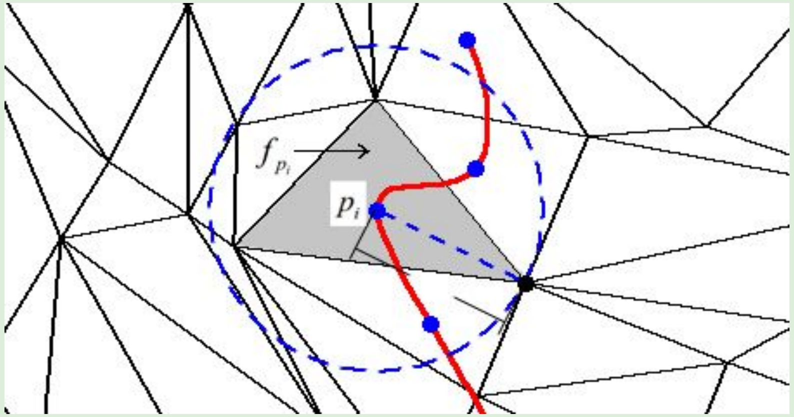
●In current implementation we choose some of the mesh vertices as handle members.

●It is obvious that handles constructed on the original complex mesh are more close to the user intention.

●When the user picks 3D points on the simplified mesh *Ms*, the system streams these points to the cloud server, constructing handles on the complex surface.

Select mesh vertices as handles

$$L = \{p_i \mid \text{intersection points on Ms}\} \xrightarrow{\text{construct}} L' = \{p'_i \mid p'_i \in V\}$$



Initial parameters of collecting p'_i

Subject to

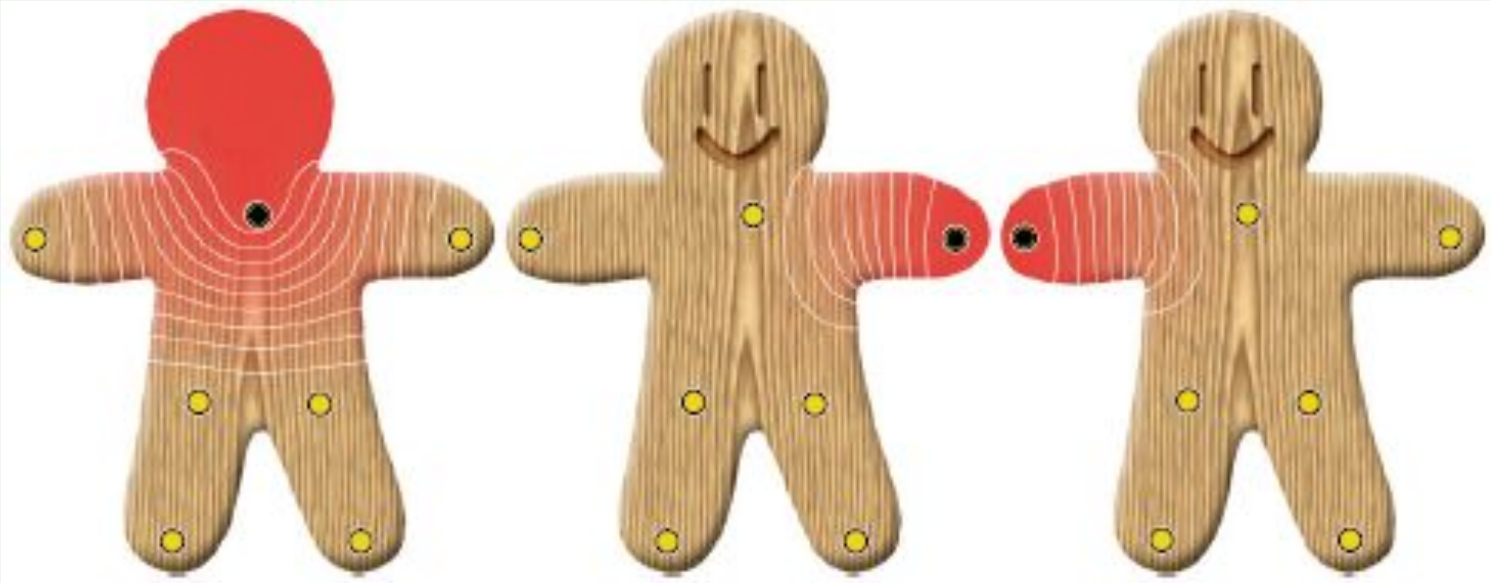
$$\underset{p'_i,\, i=1,\ldots,m}{\arg\min} \sum_{i=1}^{m} \sum_{p_j \in N(i)} \| p'_i - p_j \|_2$$

In practice, we compute the new line L’ by collecting new point *p’*$_i$ through a flood filling process performed on original mesh M, starting at the first vertex of face f(p_i) and restricted to the euclidean distance from it to p_i

- m : is the size of set L = {p_i}
- *N(i)* : is the set of p_i and its neighbors in L

Weight Computing

- Computed by powerful cloud server
- Use BBW [1] for computing weight



1. [Jacobson A, et al. SIGGRAPH 11]

Data Transmission

Data transmission includes:

- ➢Progressive meshes
- ➢Creation of handles
- ➢ROI parameters
- ➢Blending weights
- ➢Manipulations of handles

Manipulations Transmission

Simplified Mesh on Client

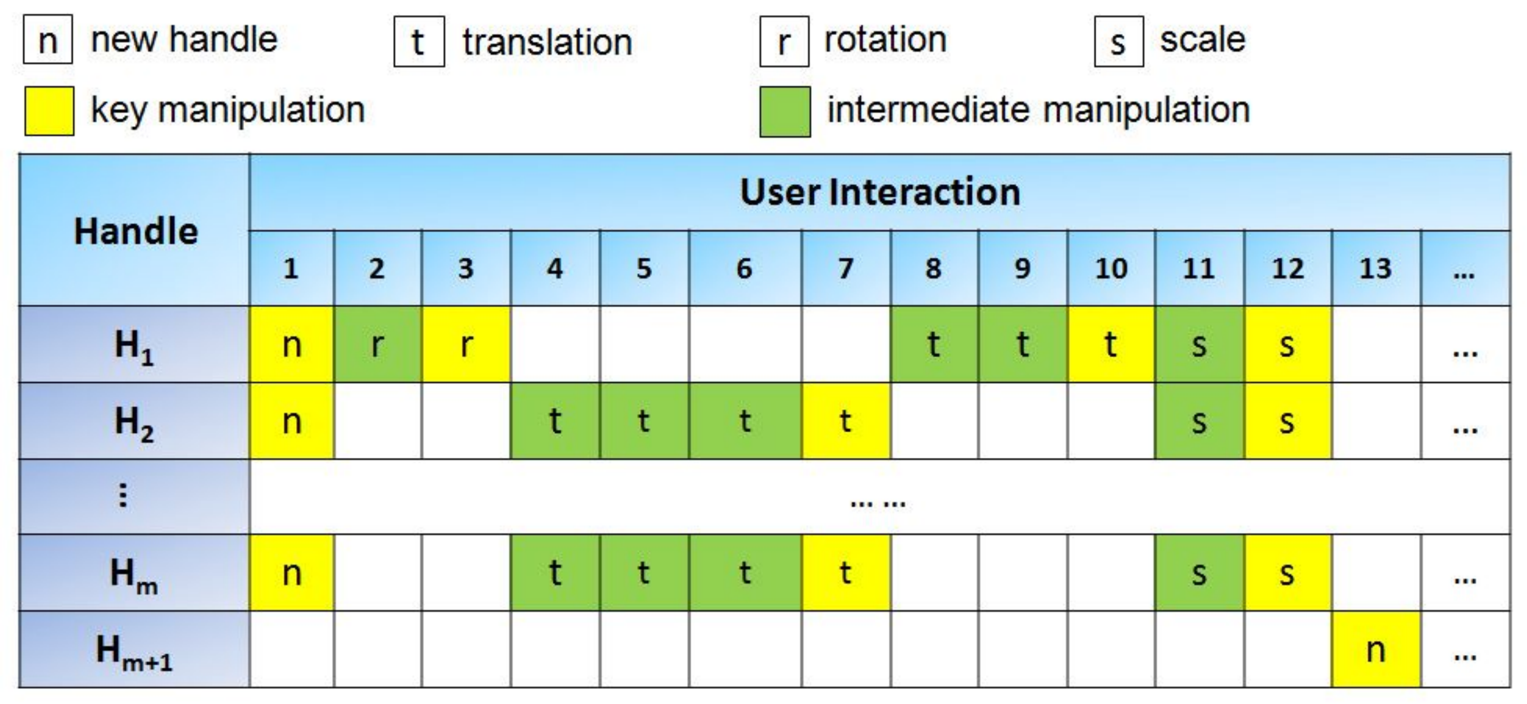
User Manipulations

- Intermediate manipulations
- Key manipulations

Original Mesh on Server

- Two cases producing Intermediate manipulations
  - Specifying a same kind of manipulation(i.e., T., R., S.) *multiple time* to get a desired deformation
  - System records continuous changes of handles by just one user manipulation, such as the trajectory of moving a point by dragging
- Key manipulations
  - Final result of a series of intermediate manipulations
  - A single manipulation

Manipulations Transmission



n new handle   t translation   r rotation   s scale

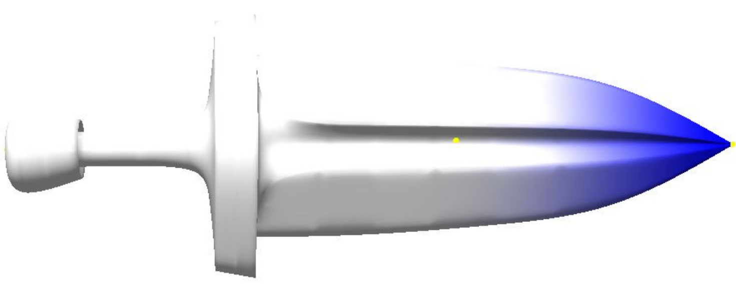
🟨 key manipulation   🟩 intermediate manipulation

Handle	User Interaction													
	1	2	3	4	5	6	7	8	9	10	11	12	13	...
H_1	🟨 n	🟩 r	🟨 r					🟩 t	🟩 t	🟨 t	🟩 s	🟨 s		...
H_2	🟨 n			🟩 t	🟩 t	🟩 t	🟨 t				🟩 s	🟨 s		...
⋮	... ...													
H_m	🟨 n			🟩 t	🟩 t	🟩 t	🟨 t				🟩 s	🟨 s		...
H_{m+1}													🟨 n	...

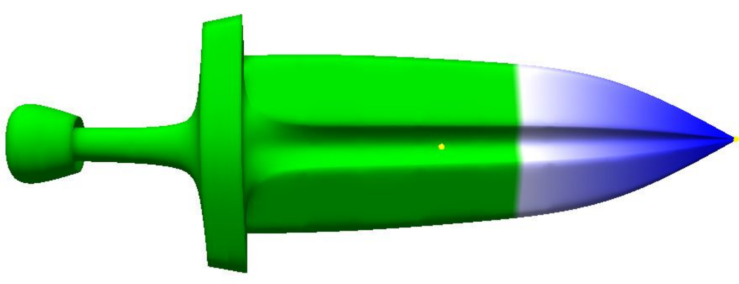
Different types of manipulations

Weights Transmission

- Each handle’s weights decay quickly so:
  - Send weights by TCP if greater than δ
  - Send weights by UDP if smaller than δ



plot of weights



plot of weight groups δ = 0.2

In a typical mesh editing system, each handle has the maximum effect on its immediate region and its influence disappears in distant parts of the object. We think that greater weights are important, should be translate by reliable channel, while the others by unreliable channel for system efficiency.

Weight Estimation

- Estimate lost weight

$$w_j(i) = \sum_{k \in \tilde{N}(i)} \mu_{ik} w_j(k)$$

  - w_j : the weight function associated with handle H$_j$
  - $\overset{\%}{N}(i)$: the 1-ring neighbors of vertex v_i

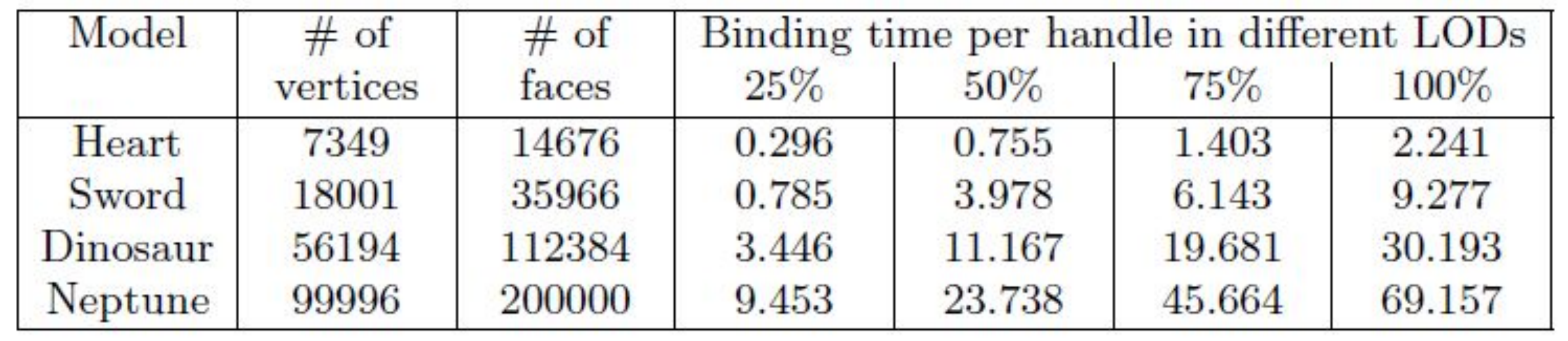
$$m_{ik} = \frac{1}{|\overset{\%}{N}(i)| - 1}\left(1 - \frac{\| v_i - v_k \|_2}{å_{\hat{I}\,\overset{\%}{N}(i)} \| v_i - v_l \|_2}\right)$$

$$å \; m_{ik} = 1$$

Outline

- Ghost Mesh System Pipeline
- User Interface
- Algorithms
- Evaluation
- Editing Examples

Binding Time



Model	# of vertices	# of faces	Binding time per handle in different LODs			
			25%	50%	75%	100%
Heart	7349	14676	0.296	0.755	1.403	2.241
Sword	18001	35966	0.785	3.978	6.143	9.277
Dinosaur	56194	112384	3.446	11.167	19.681	30.193
Neptune	99996	200000	9.453	23.738	45.664	69.157

Statistics of binding time, measured in seconds.

□ The computational overhead of blending weights can be reduced significantly by using a simplified mesh.

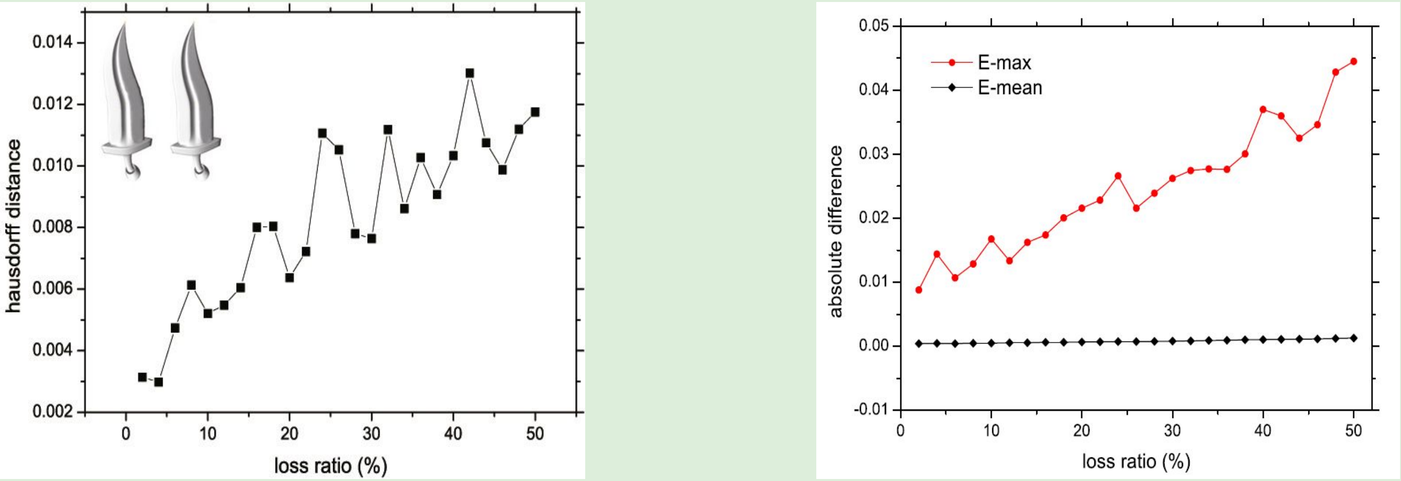
Client:

- Core2 Duo Laptop
- 2.5GHz CPU
- 4GB RAM

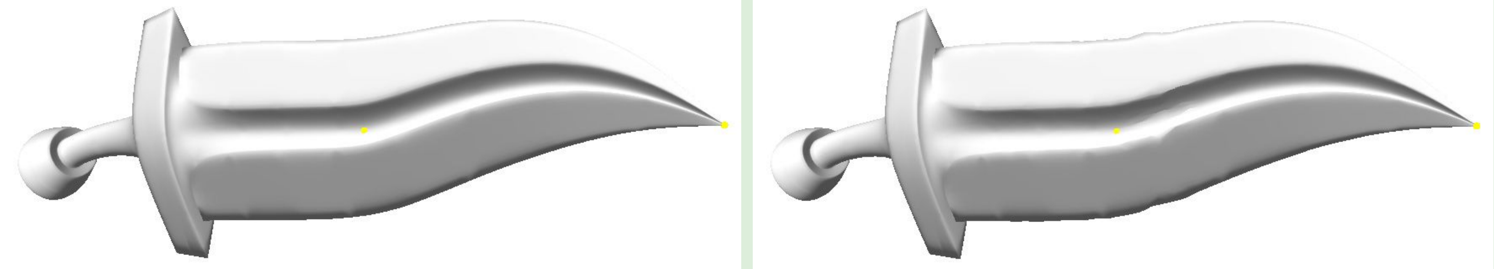
Server:

- Windows XP PC
- Intel Core2 Xeon 3.07GHz CPU
- 4GB RAM

Weights Interpolation

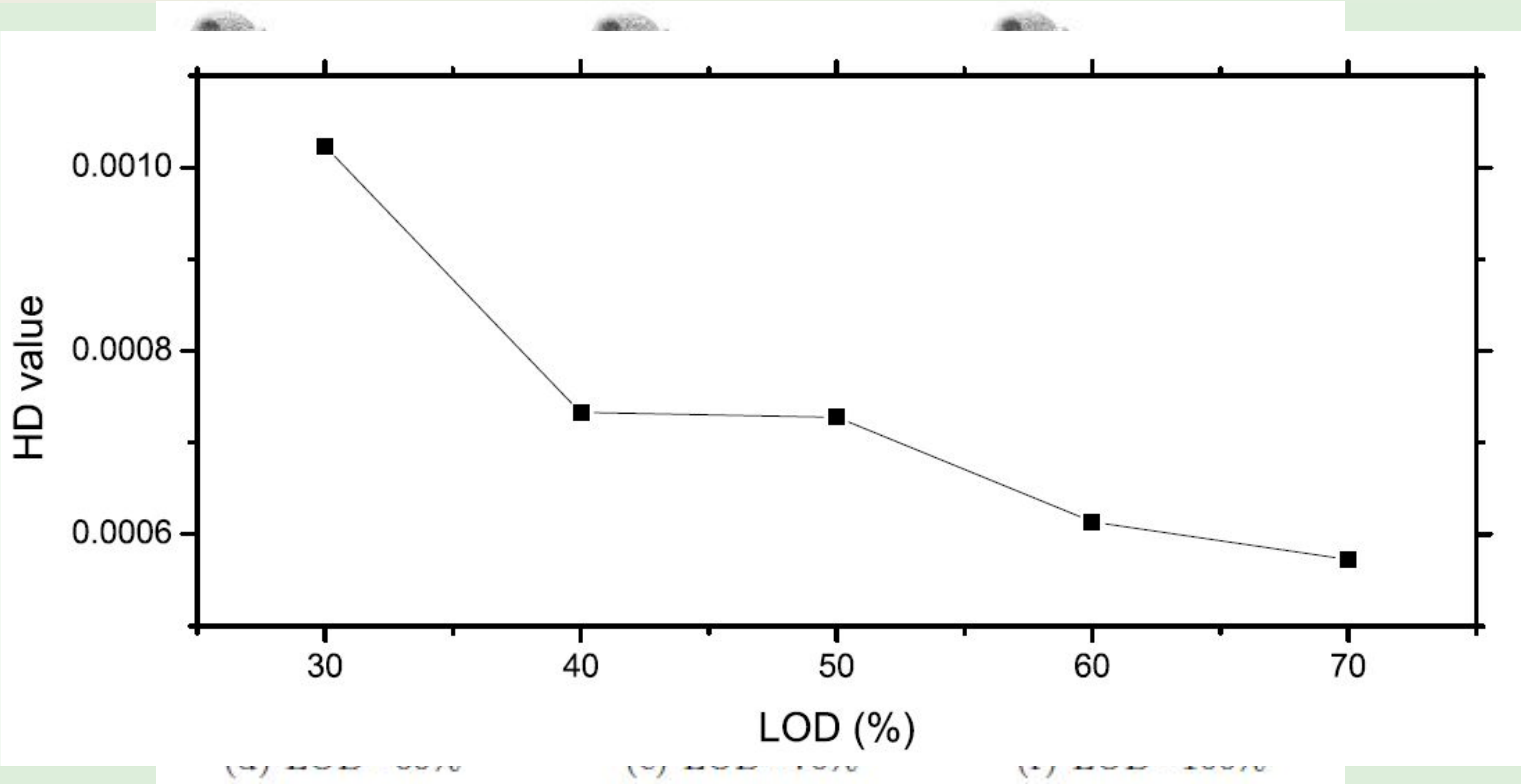


The geometric distortion of the deformed Sword and weight interpolation error with different data loss ratios. The geometric distortion is measured in Hausdorff Distance.



δ = 0.2 and loss ratio is 50%

Simplified vs. Original Results

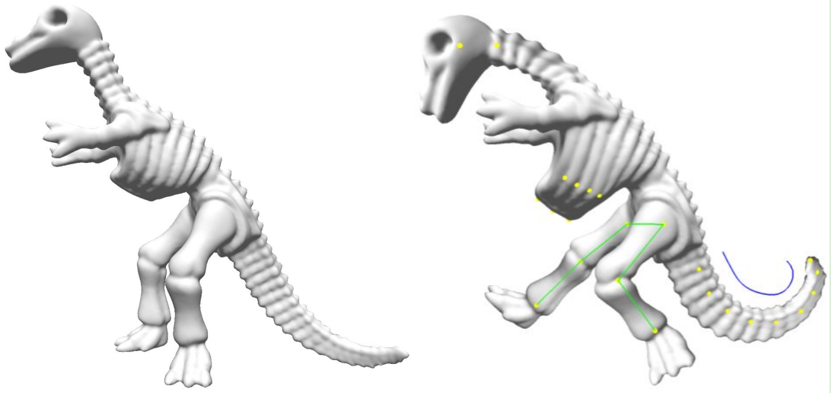


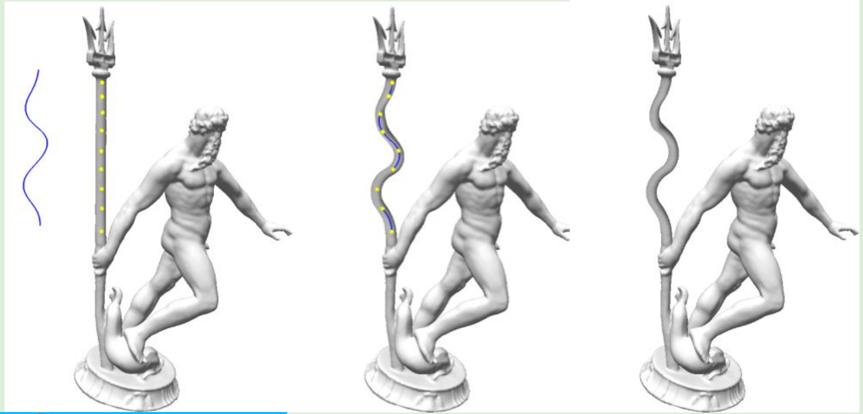
Simplified mesh results and the corresponding results by cloud server

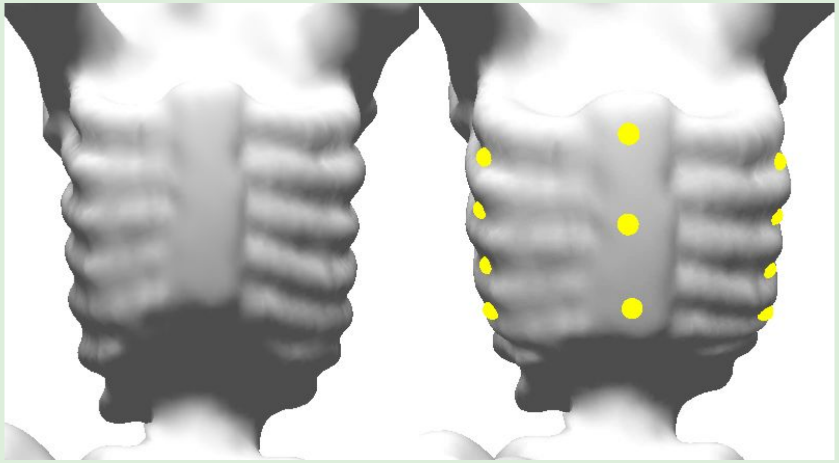
Outline

- Ghost Mesh System Pipeline
- User Interface
- Algorithms
- Evaluation
- Editing Examples

Examples







Conclusion

□ We have presented a system for cloud-based interactive mesh editing, namely GhostMesh.

□ Allowing user to edit a complex mesh that stored in the cloud server by interactively modifying a simplified mesh locally.

□ We proposed efficient strategies to classify and stream blending weights and user manipulations.

Thank You