Pacific Graphics 2021

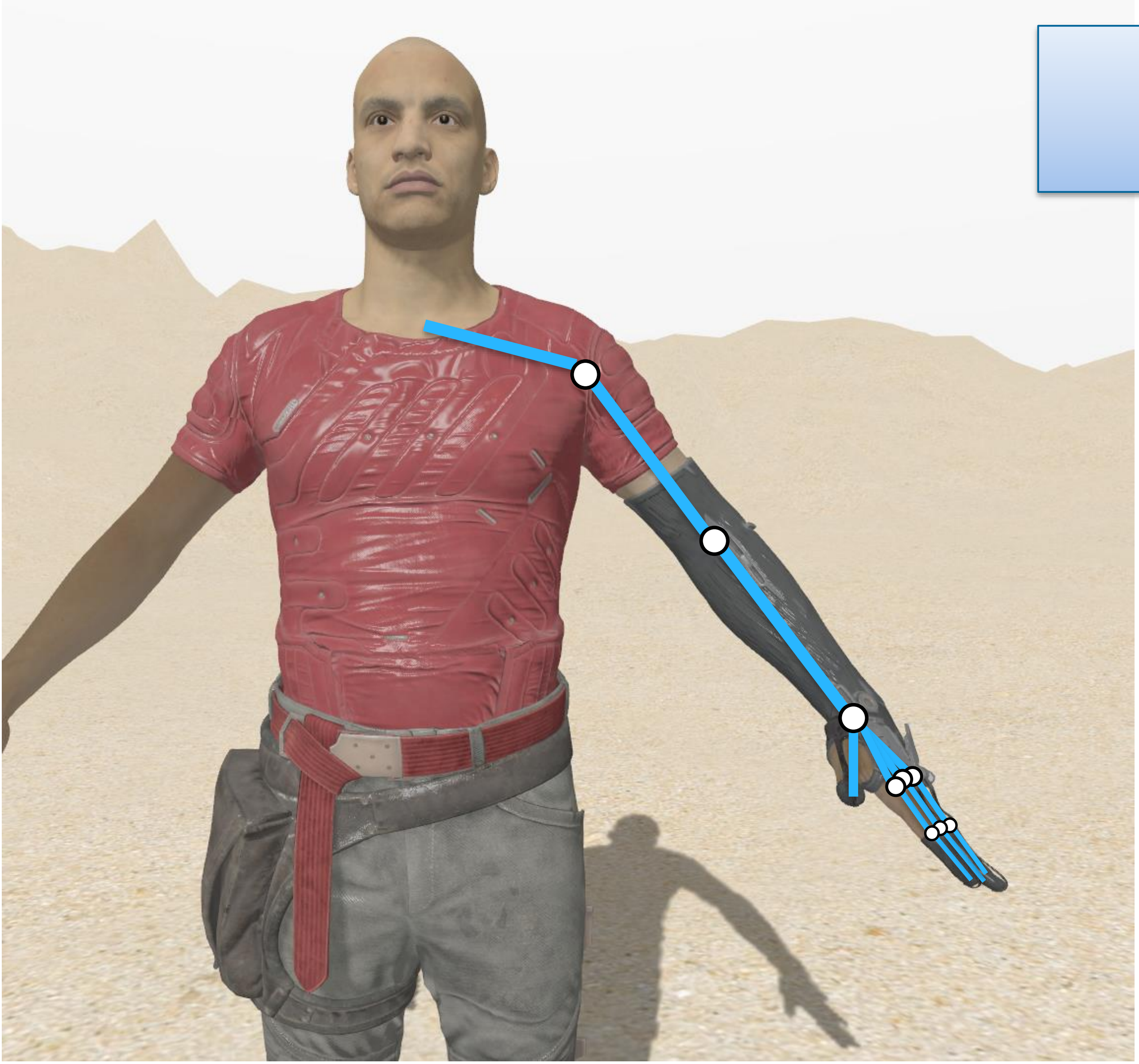
Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Johannes Unterguggenberger, Bernhard Kerbl,
Jakob Pernsteiner, and Michael Wimmer

TU Wien, Institute of Visual Computing &
Human-Centered Technology, Austria

Clusters!

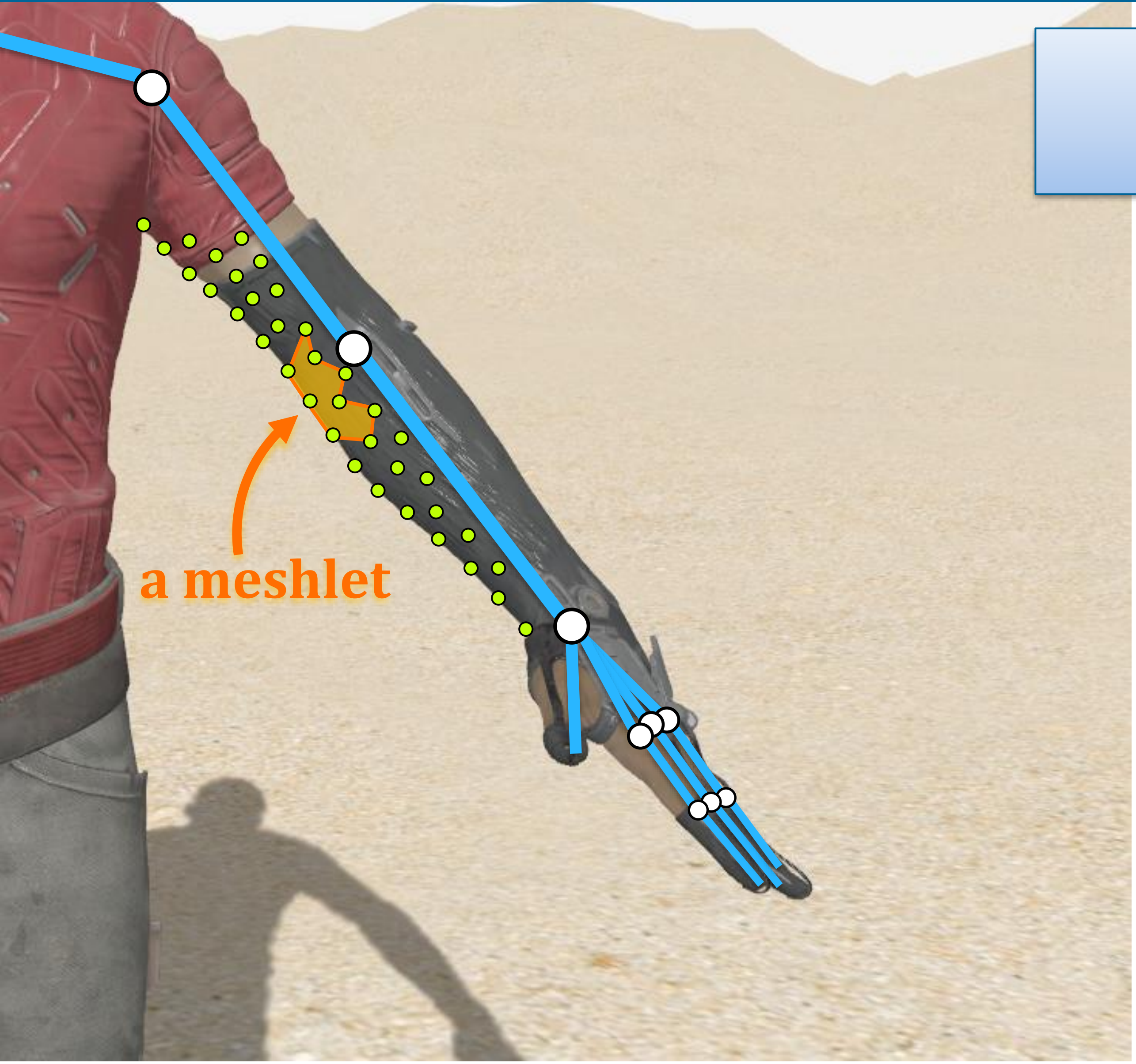
Skinned Meshes



Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

- Skinned, animated models
- Underlying skeleton
  - Bone hierarchy
  - Hierarchical transformation

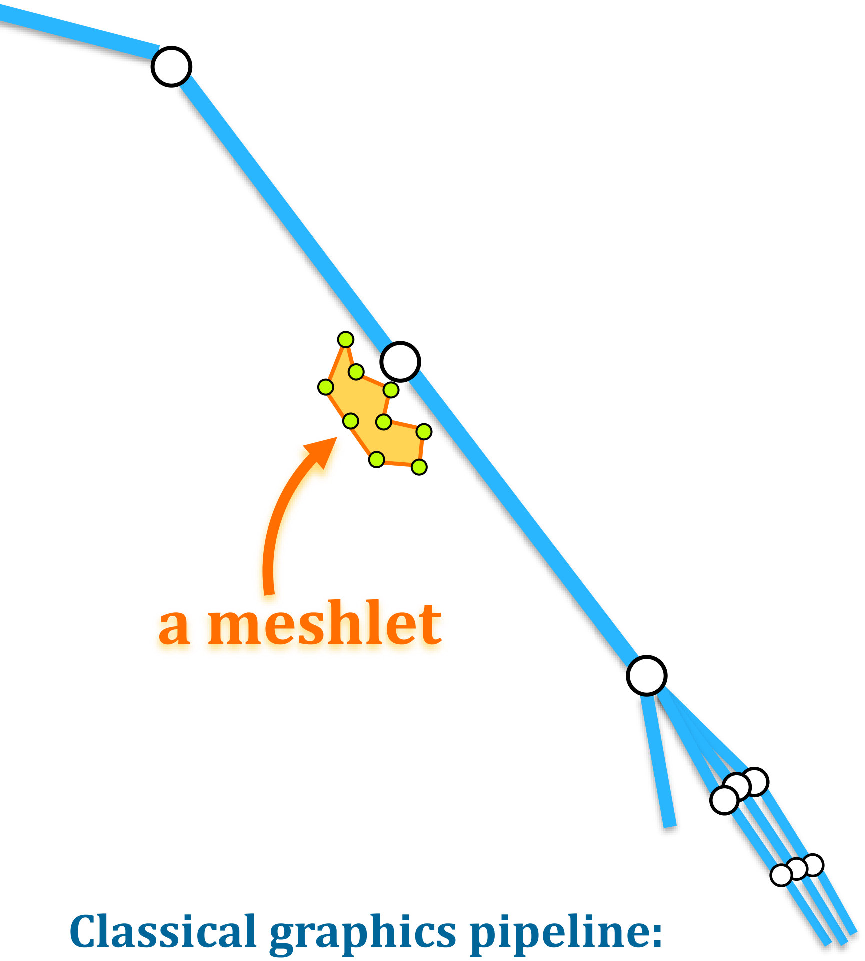
Skinned Mesh…lets



Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

- Skinned, animated models
- Underlying skeleton
  - Bone hierarchy
  - Hierarchical transformation
- Vertices (the skin)
  - Assigned to one or multiple bones
  - Transformed by bones (weighted)

Meshlets!



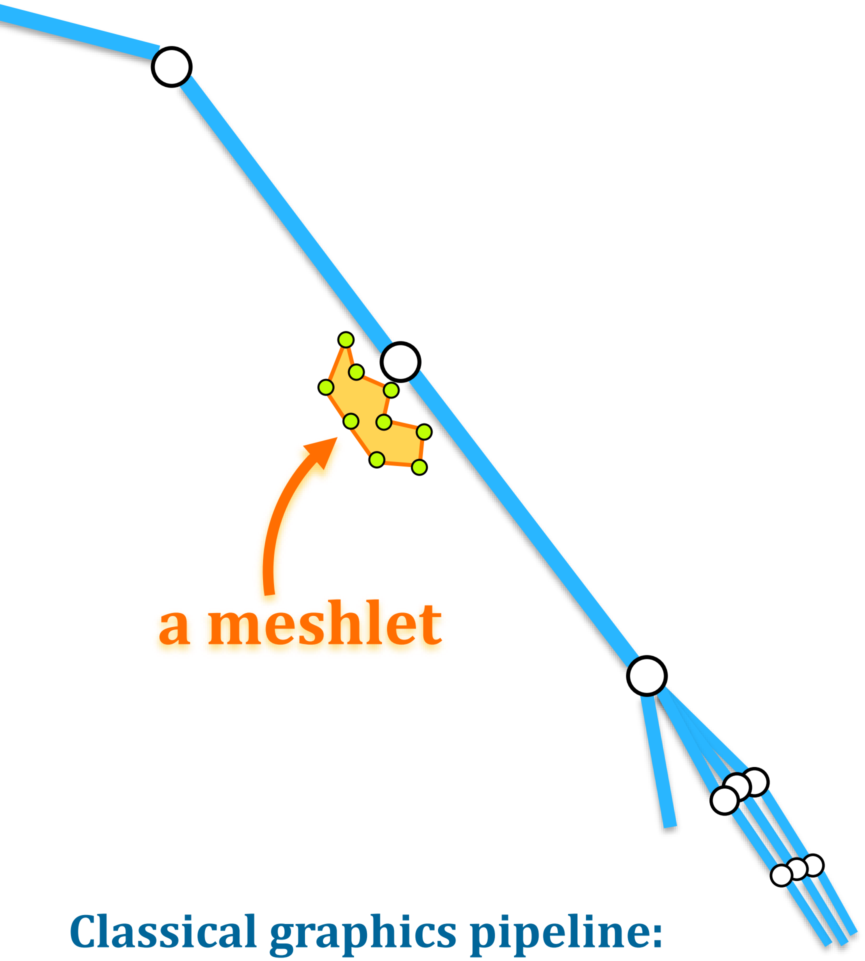
Conservative **Meshlet** Bounds for Robust Culling of Skinned Meshes

- Meshlets
  - Small clusters of geometry
  - E.g., small patch of indexed geometry
- New shader stages (`task` and `mesh`) for efficient processing
  - *within* rasterization pipelines
  - Using Vulkan with Nvidia extension
    - Max. 64 vertices and 126 triangles

Classical graphics pipeline:

Input Assembly → Vertex Shader → Tess.Ctrl. Shader → Tessellator → Tess.Eval. Shader → Geometry Shader → Rasterizer → Fragment Shader

Meshlets!

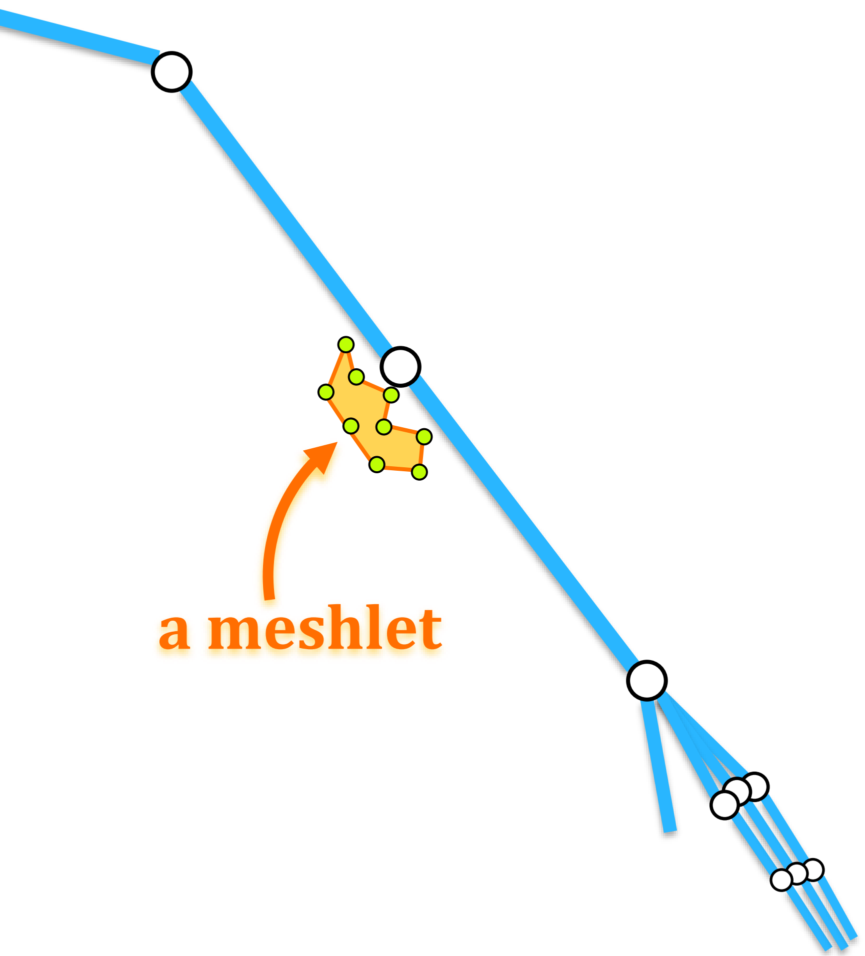


Conservative **Meshlet** Bounds for Robust Culling of Skinned Meshes

- Meshlets
  - Small clusters of geometry
  - E.g., small patch of indexed geometry
- New shader stages (`task` and `mesh`) for efficient processing
  - *within* rasterization pipelines
  - Using Vulkan with Nvidia extension
    - Max. 64 vertices and 126 triangles

Classical graphics pipeline:

Mesh Shader-Based Graphics Pipeline



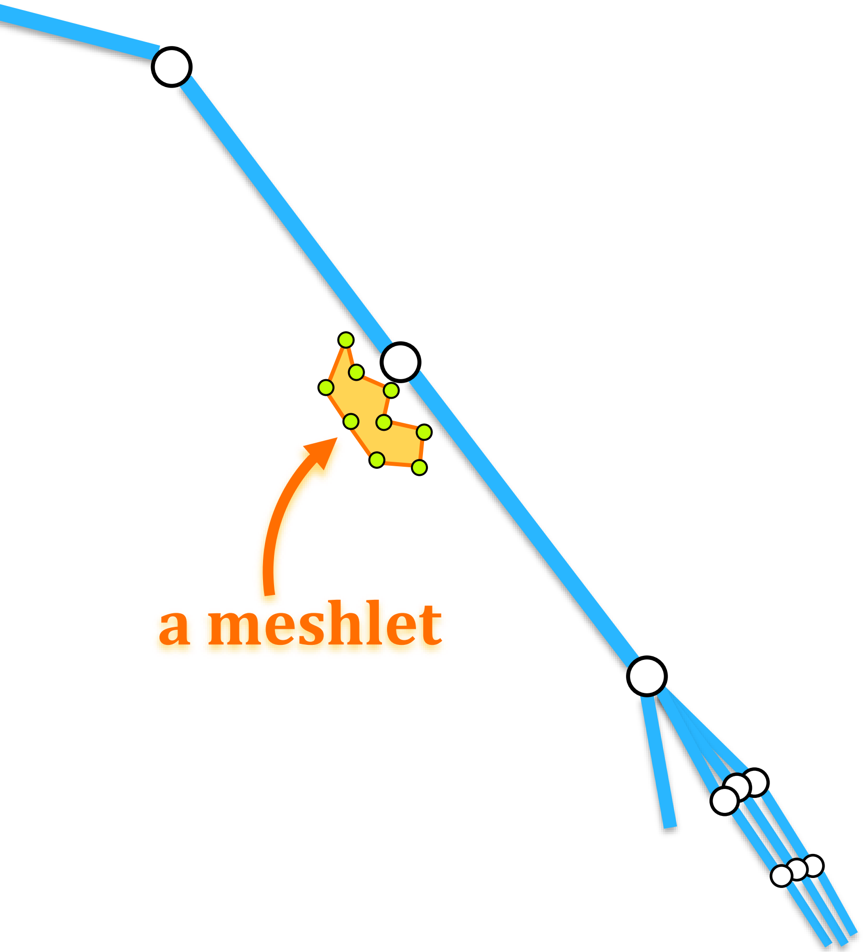
Conservative **Meshlet** Bounds for Robust Culling of Skinned Meshes

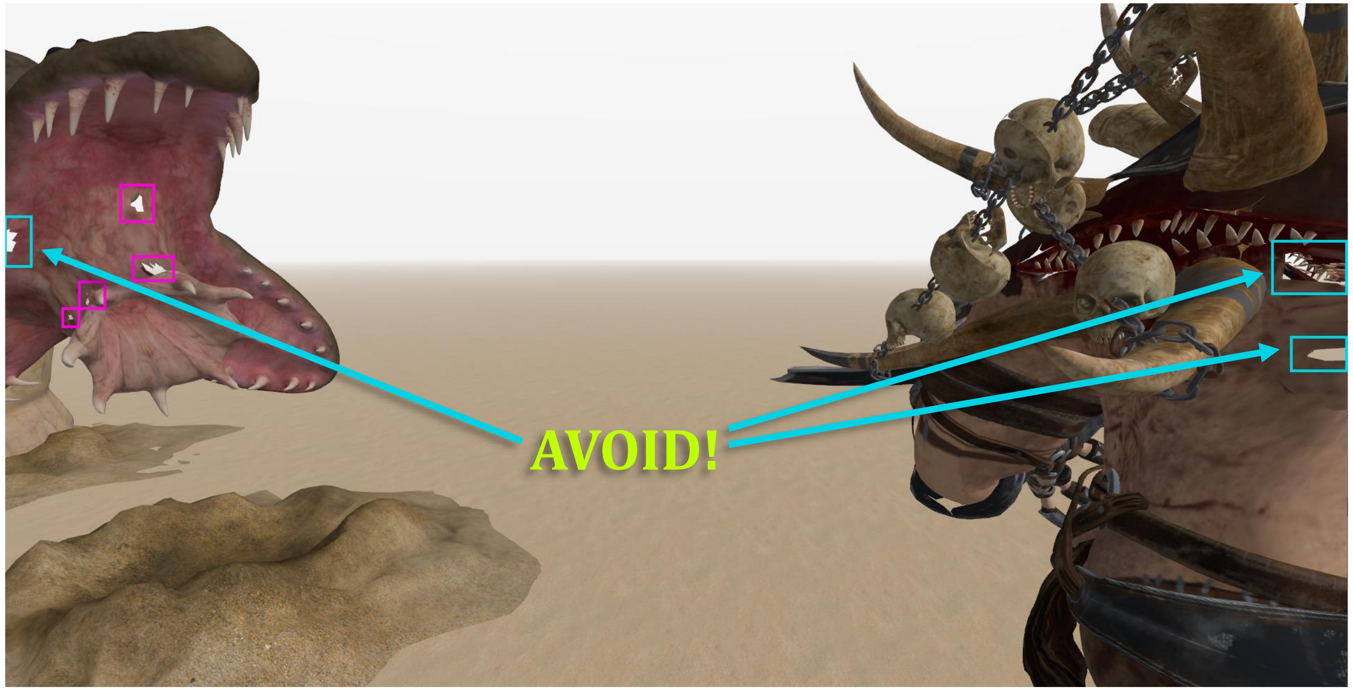
- Meshlets
  - Small clusters of geometry
  - E.g., small patch of indexed geometry
- New shader stages (`task` and `mesh`) for efficient processing
  - *within* rasterization pipelines
  - Using Vulkan with Nvidia extension
    - Max. 64 vertices and 126 triangles

Task and mesh shader pipeline:

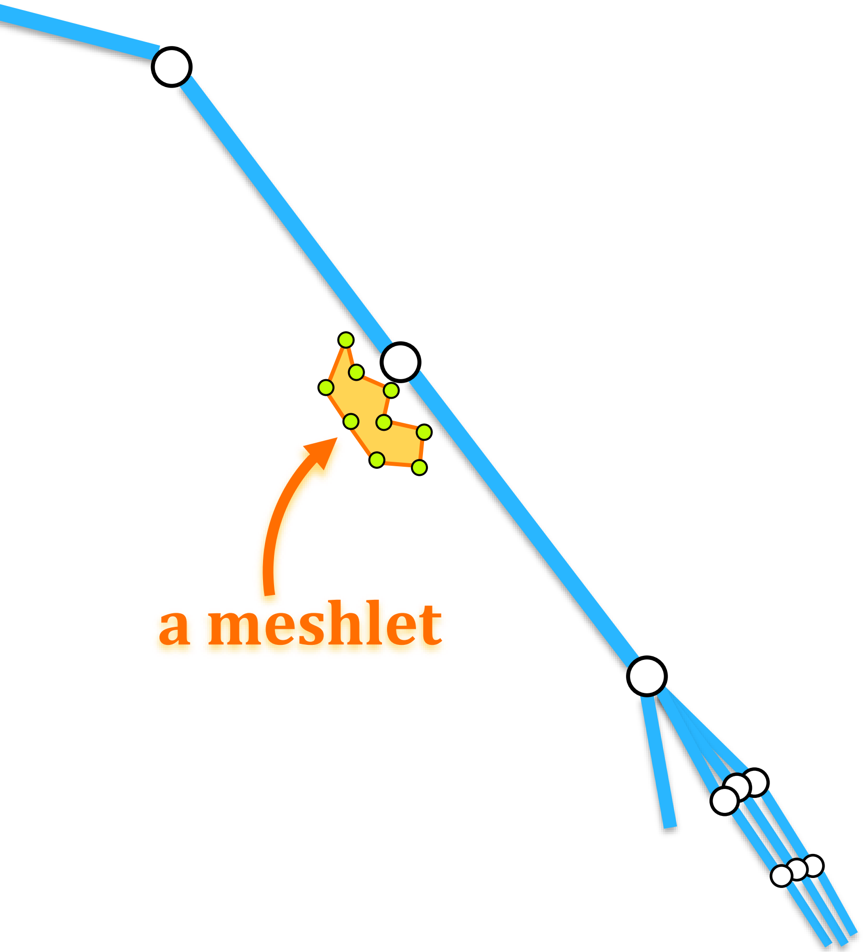
- Efficient processing of meshlets
- Culling in `task` shader stage

Rendering Artefacts To Be Avoided

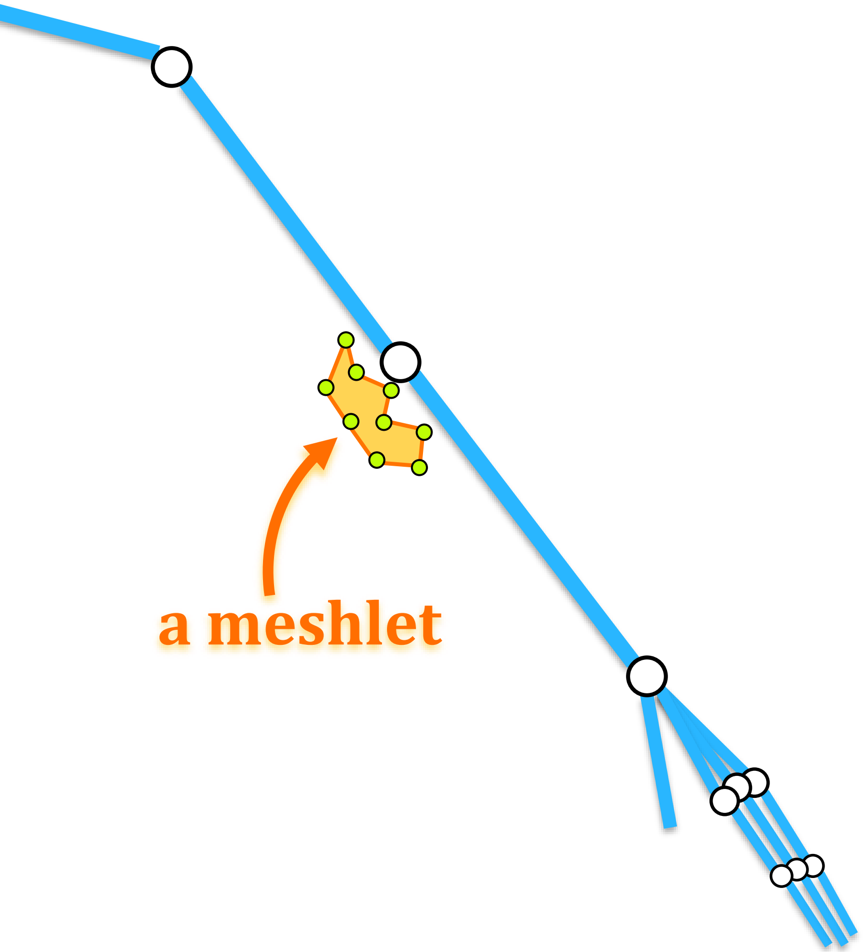


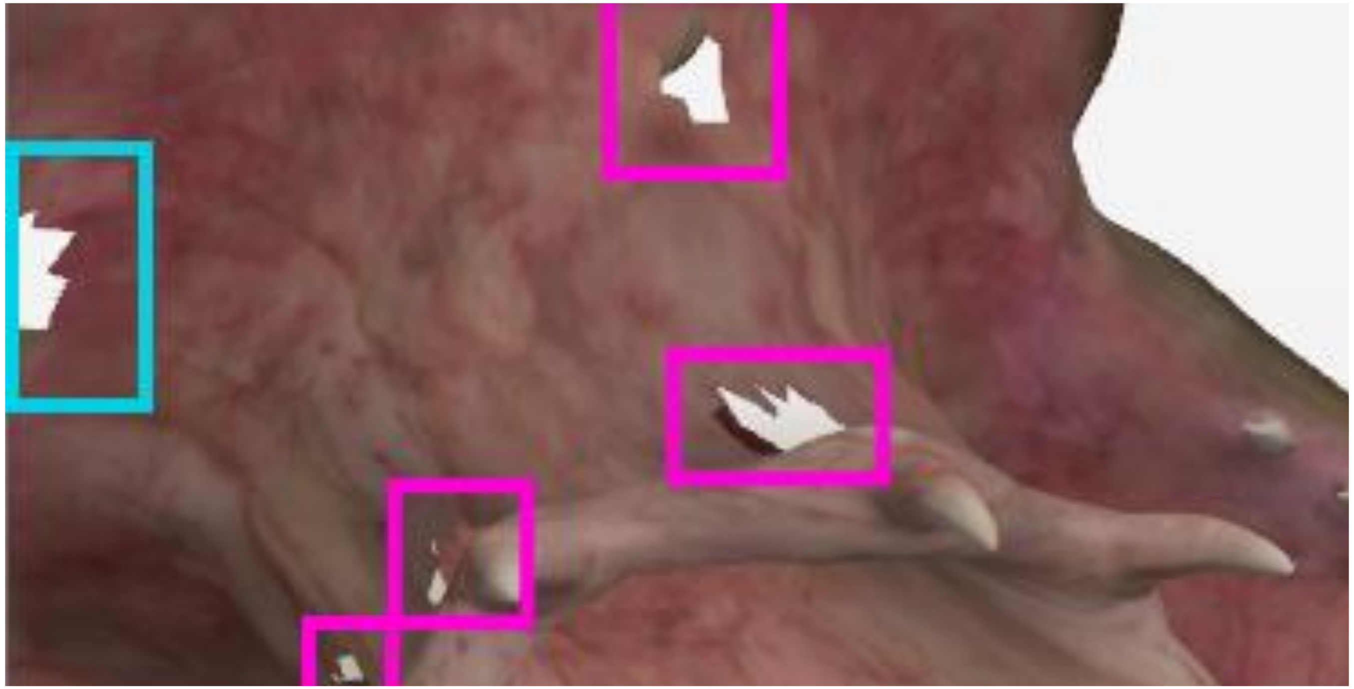


Rendering Artefacts To Be Avoided

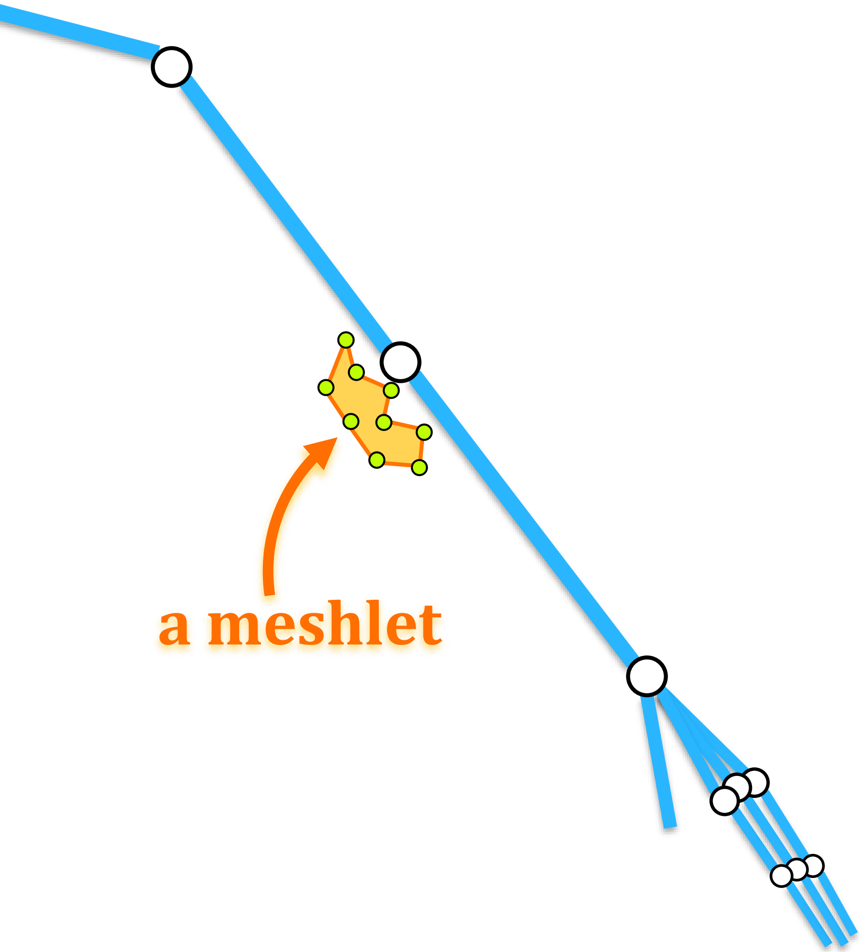


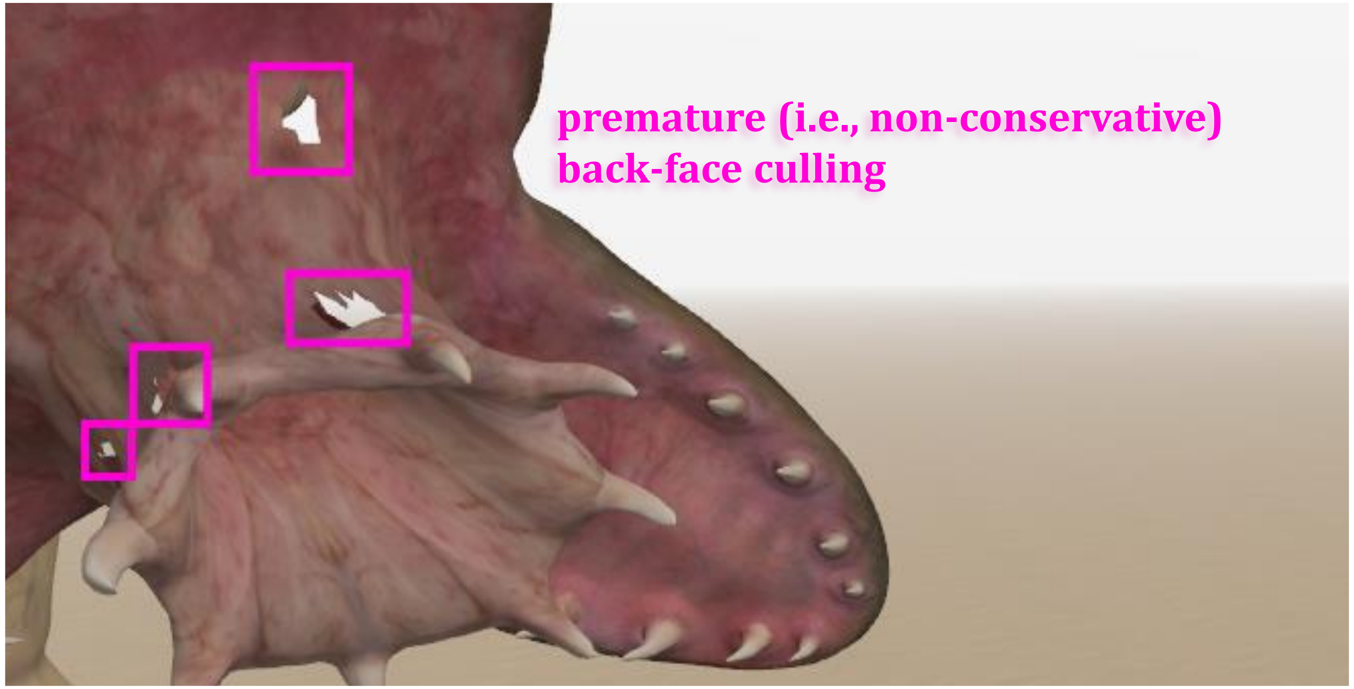
Rendering Artefacts To Be Avoided



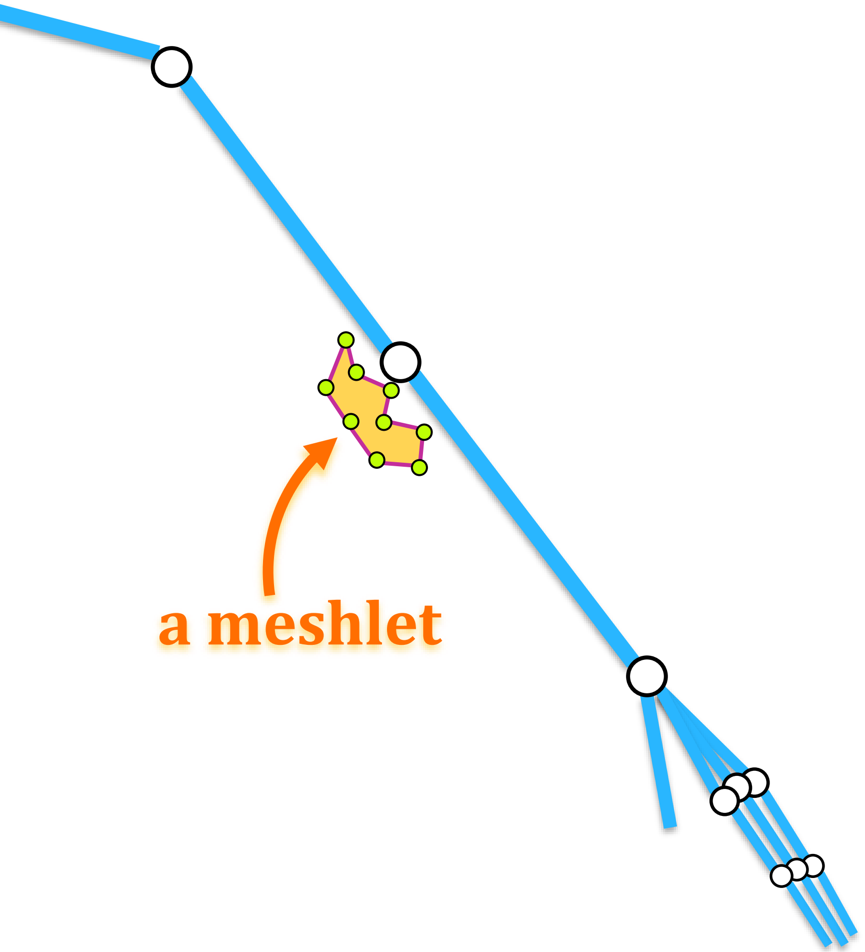


Rendering Artefacts To Be Avoided

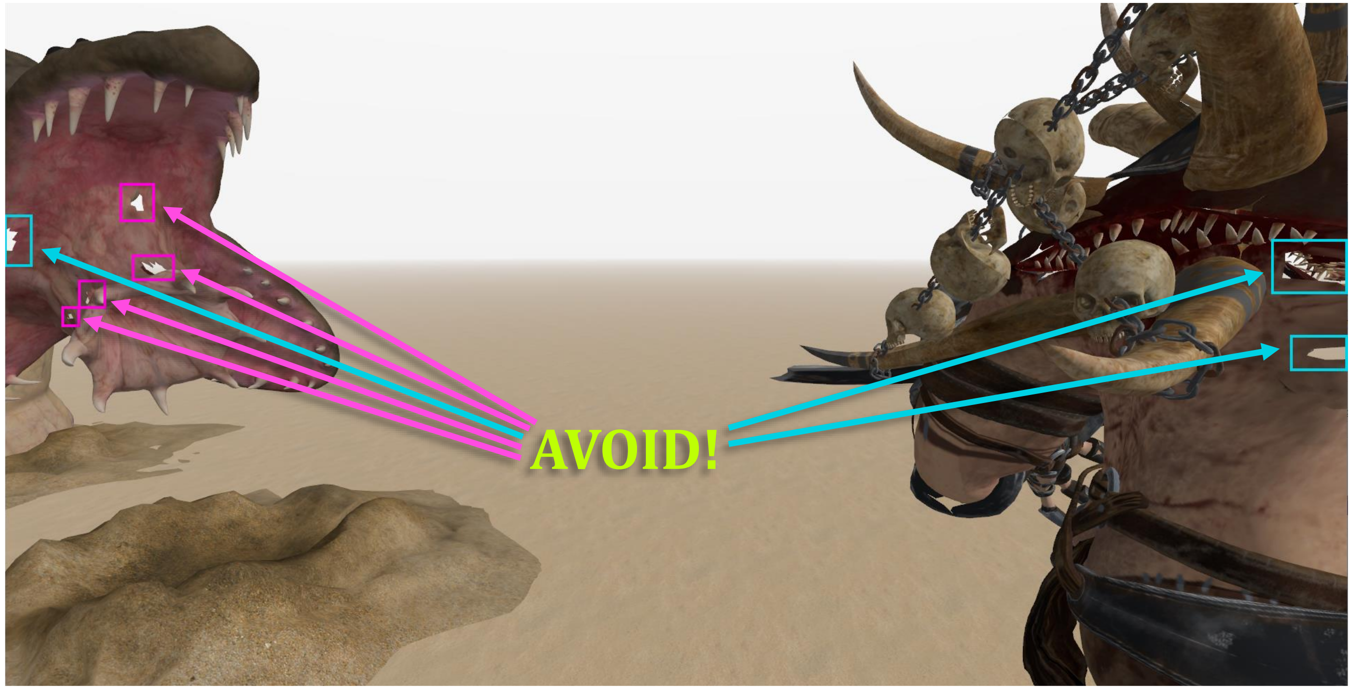




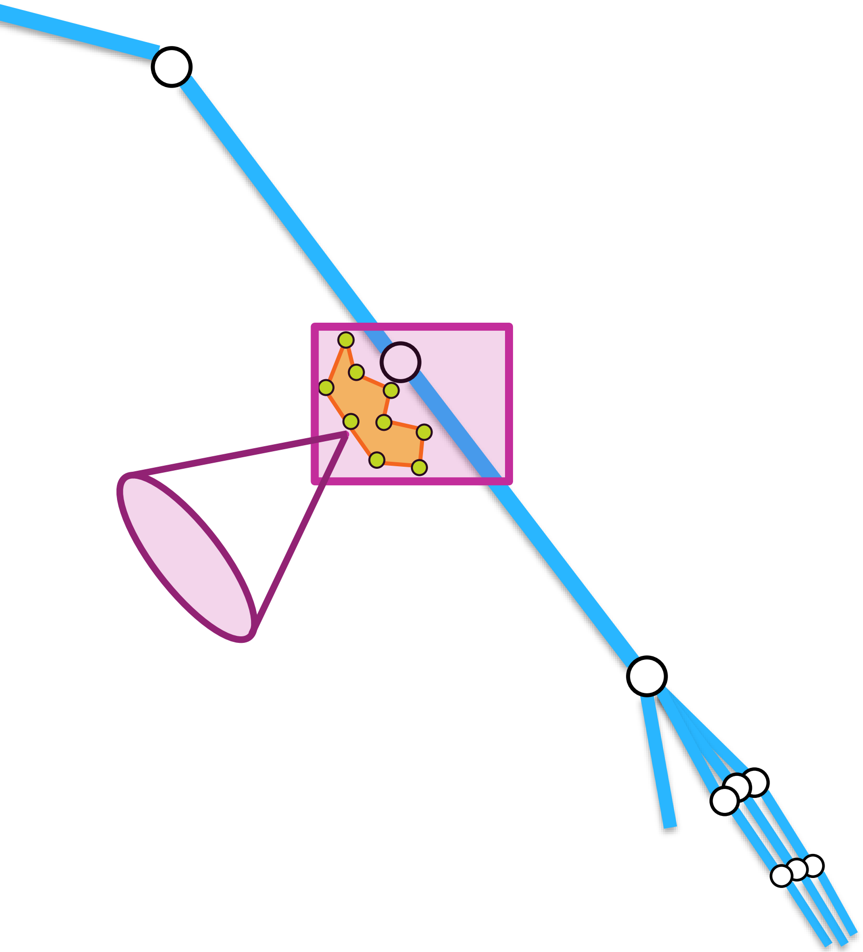
Rendering Artefacts To Be Avoided



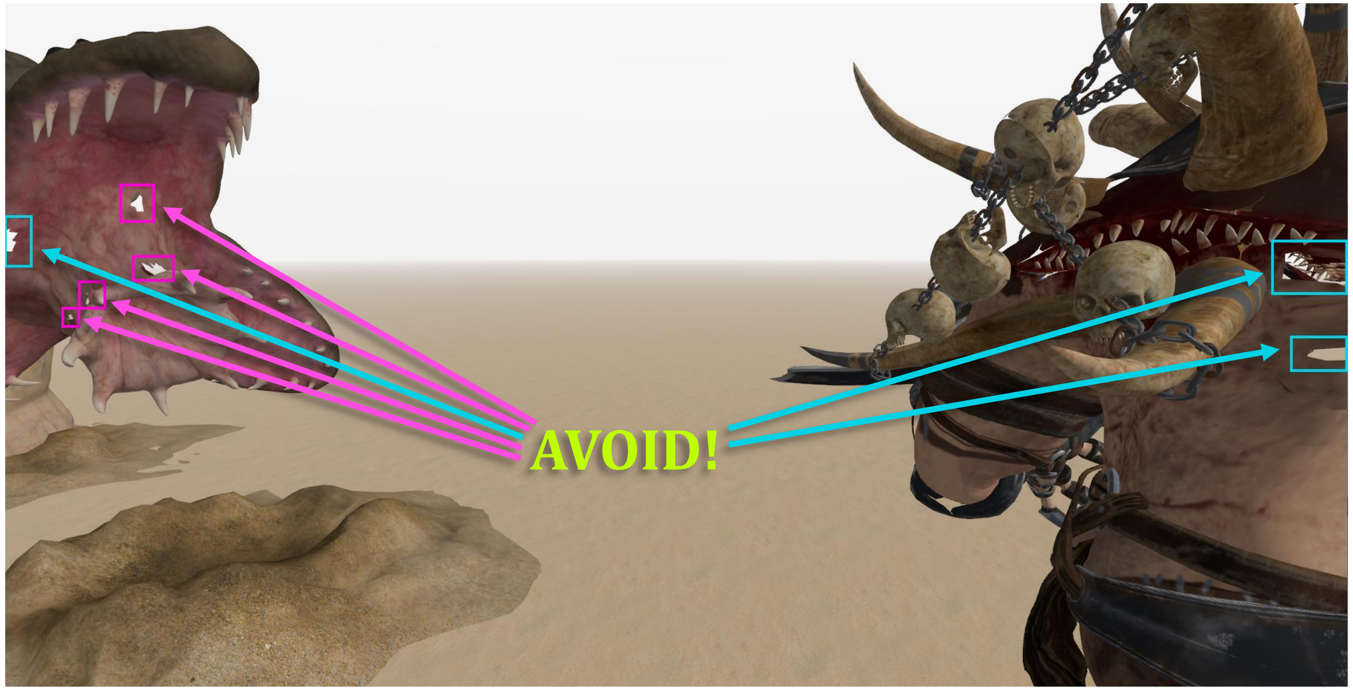
Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes



Rendering Artefacts To Be Avoided

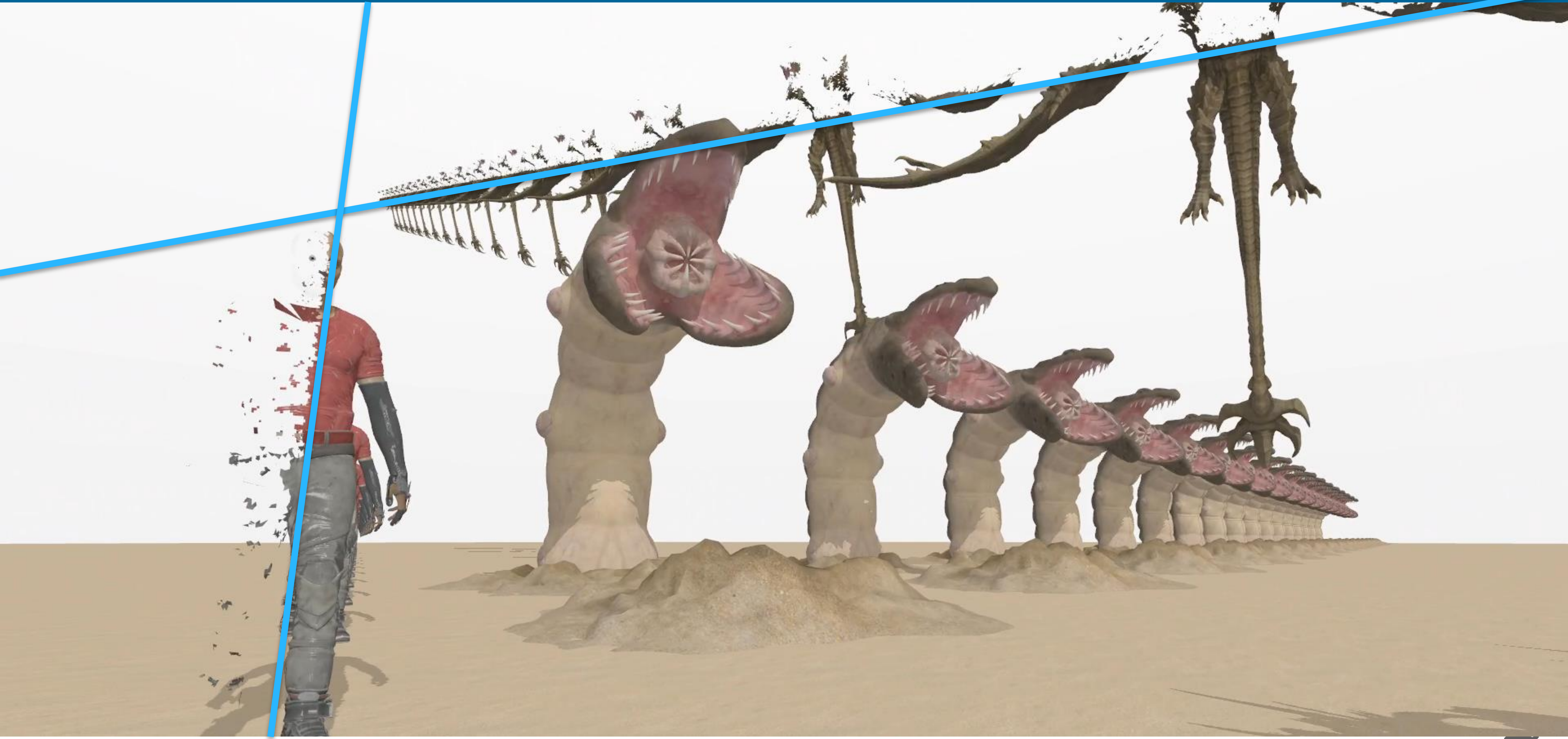


Conservative Meshlet Bounds for Robust Culling of Skinned Meshes



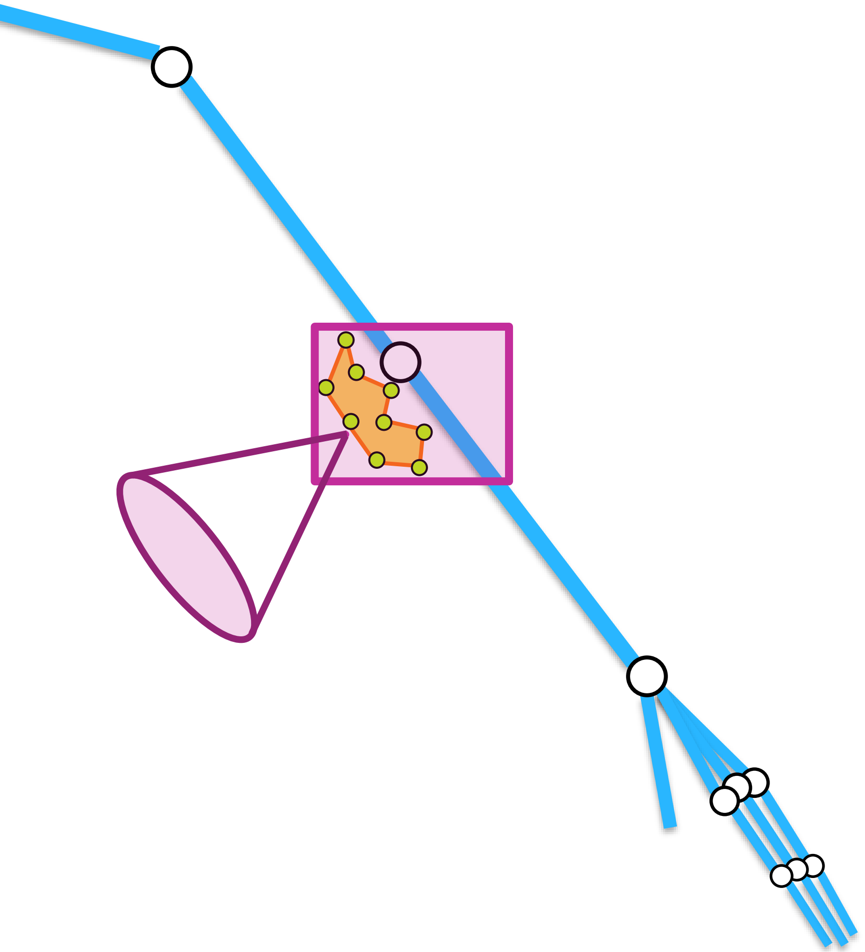
Fine-Grained Culling of Meshlets

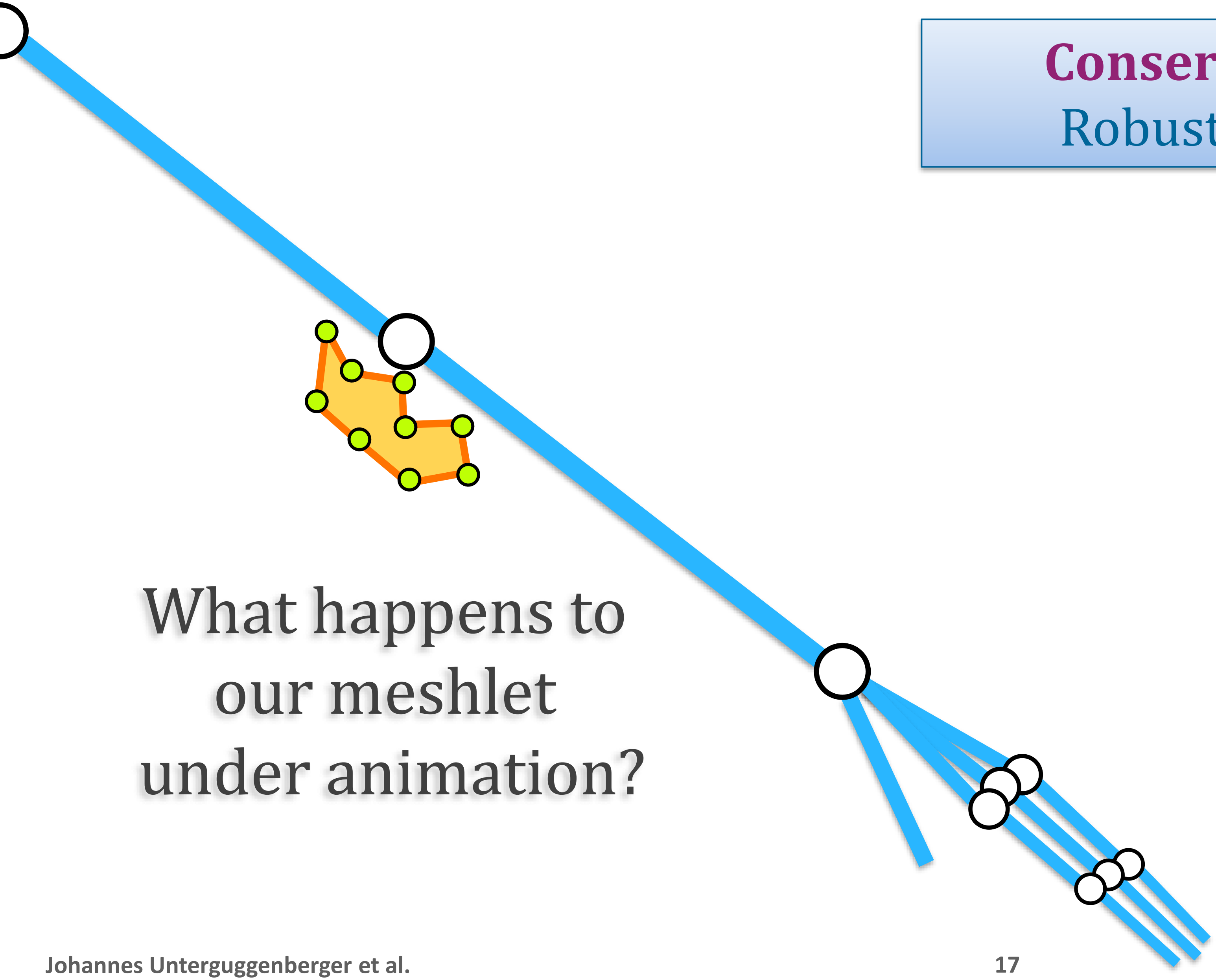
Fine-Grained Culling of Meshlets



Rendering Artefacts To Be Avoided

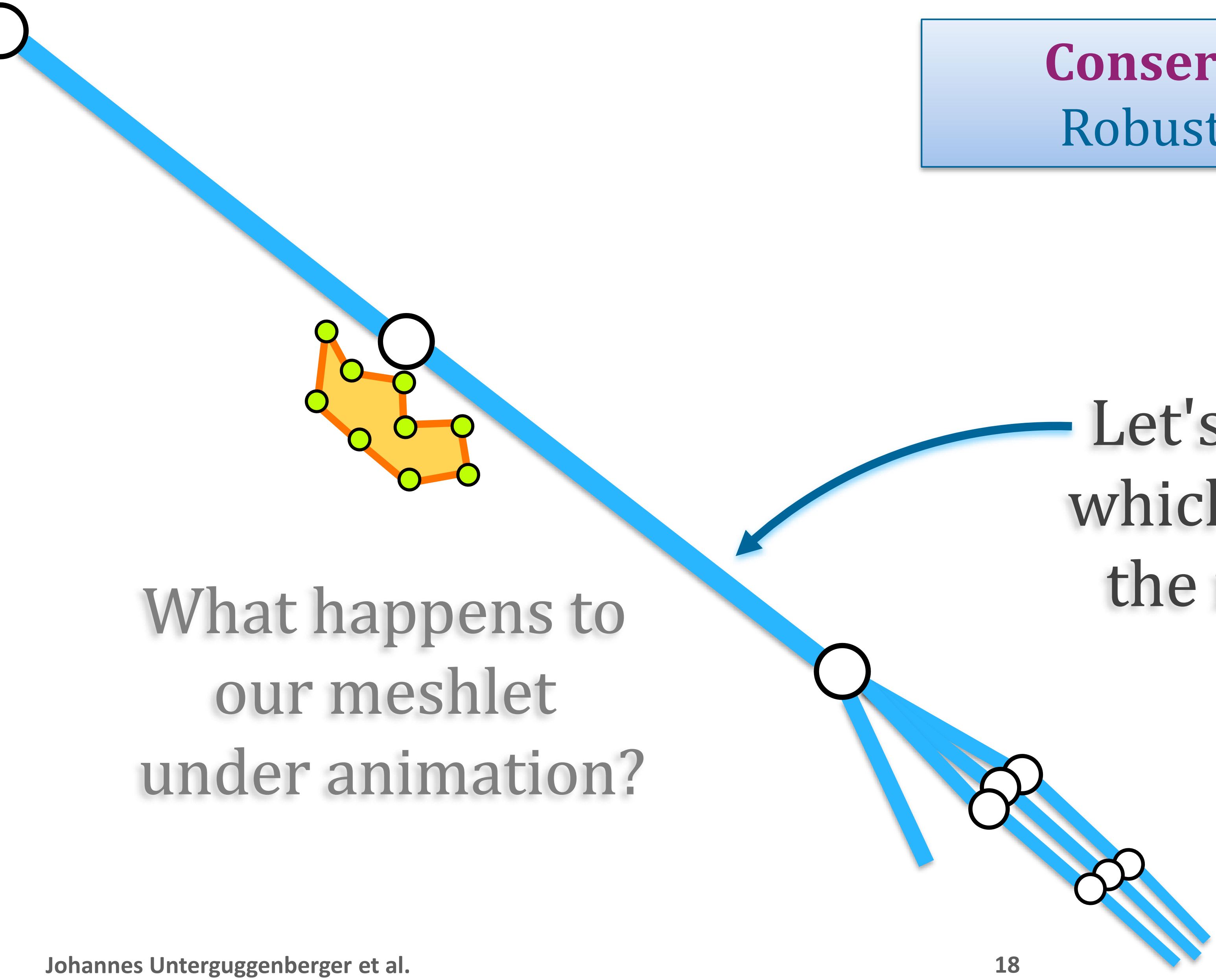
Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes





What happens to our meshlet under animation?

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

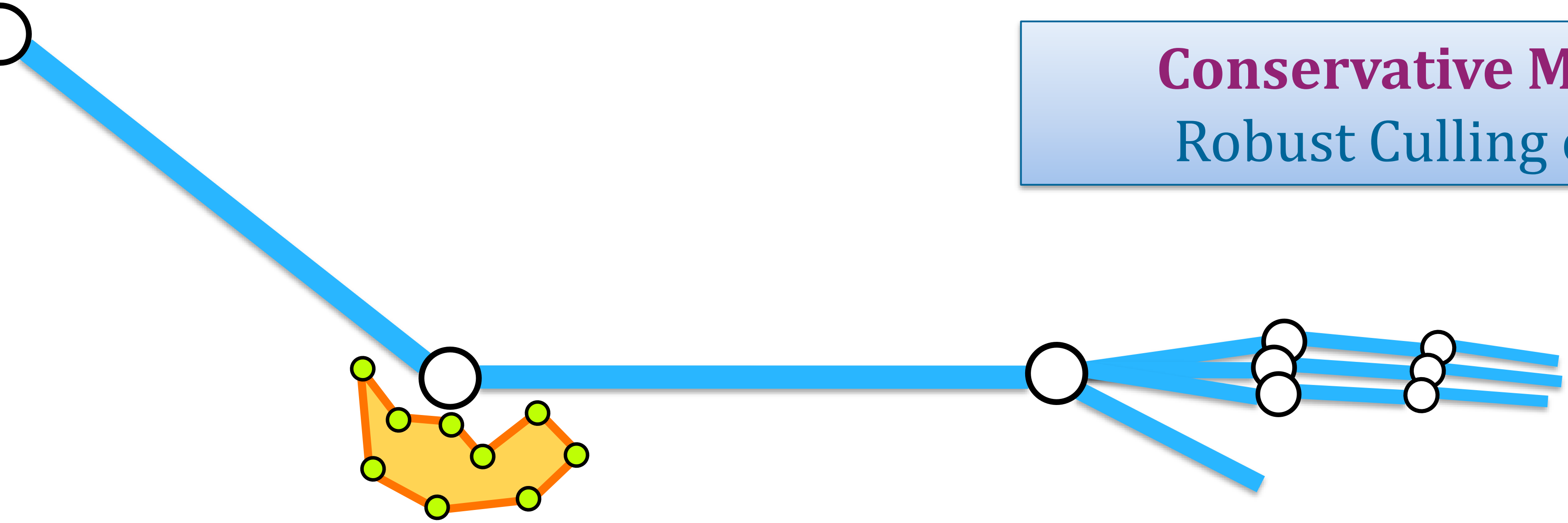


Let's animate that bone,
which in turn transforms
the meshlet's **vertices**.

What happens to
our meshlet
under animation?

Meshlet Under Animation

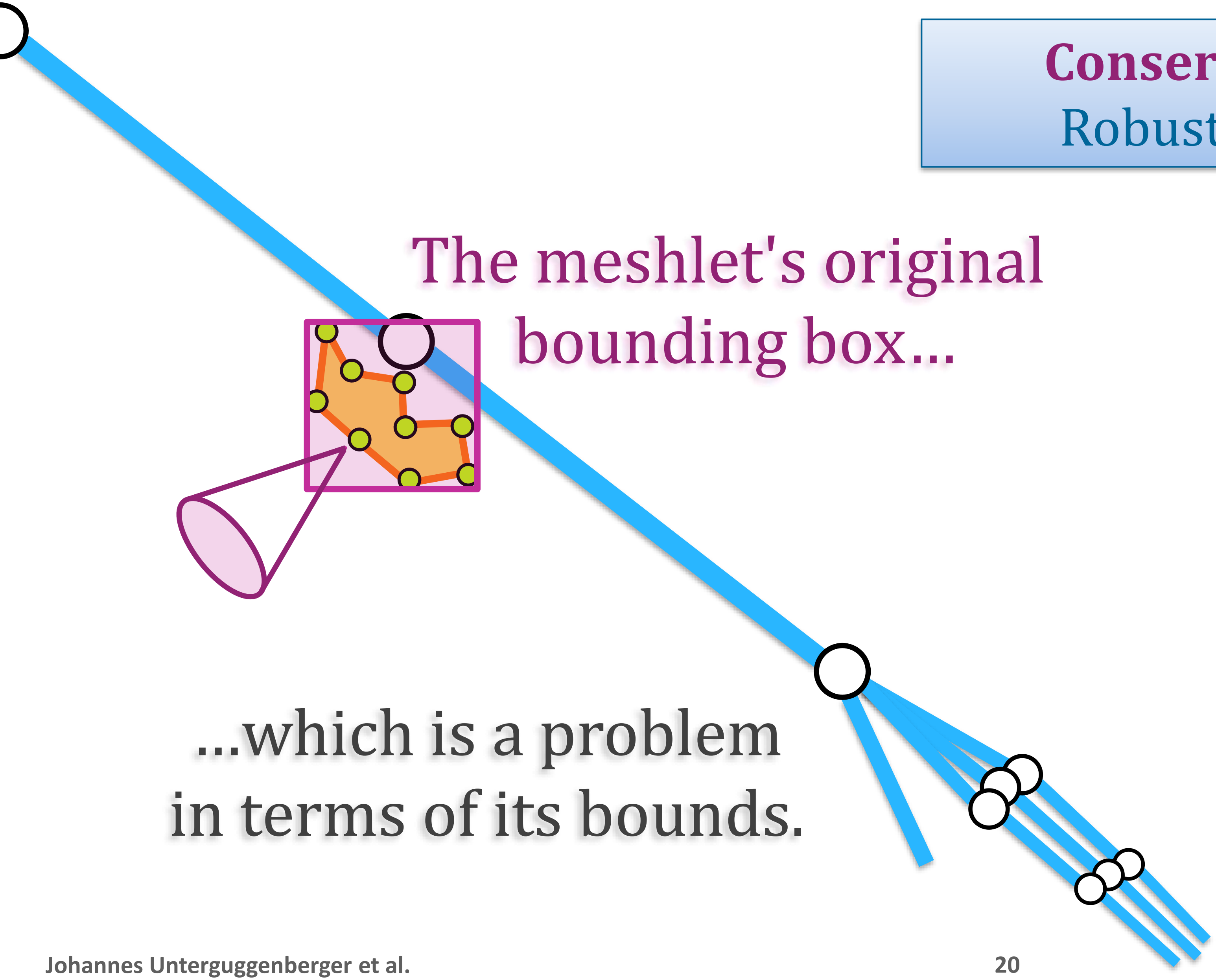
Conservative Meshlet Bounds for Robust Culling of Skinned Meshes



The meshlet's **shape** changed...

Meshlet Bounds

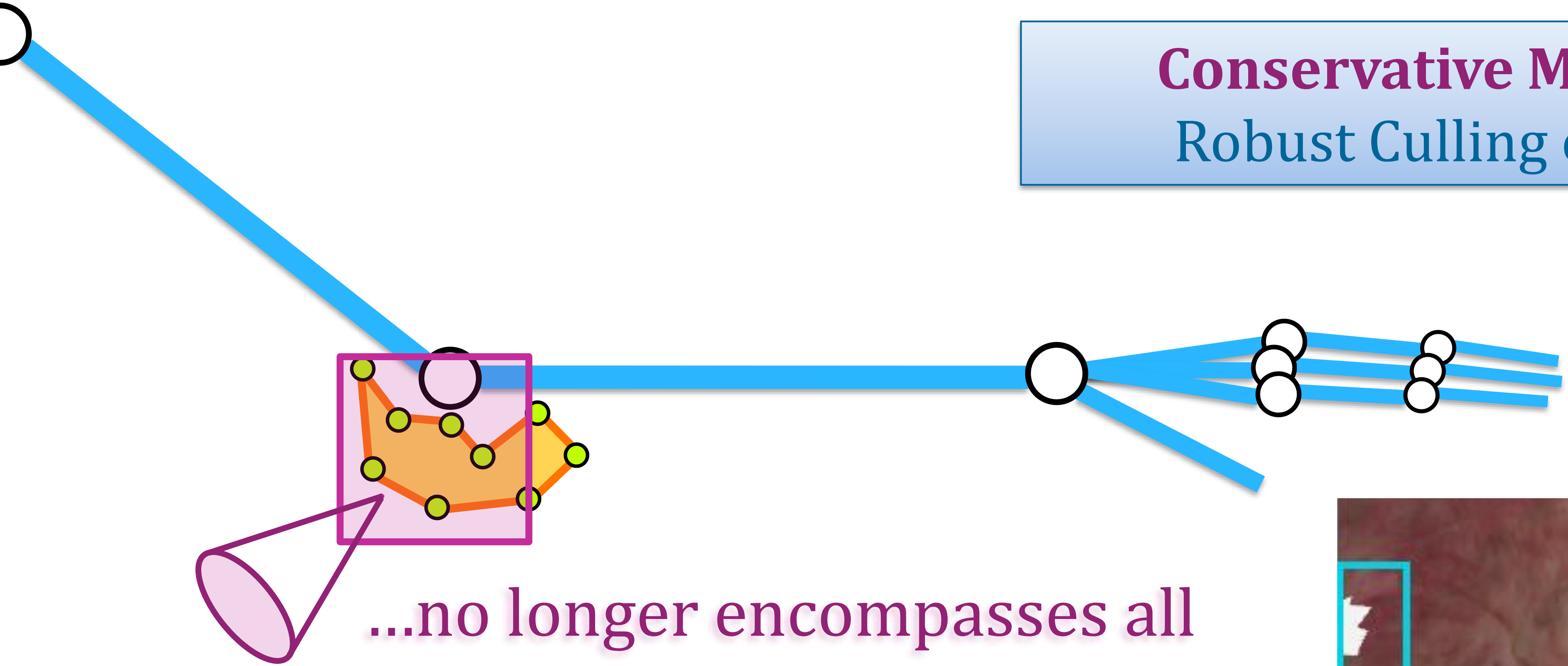
Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes



The meshlet's original bounding box...

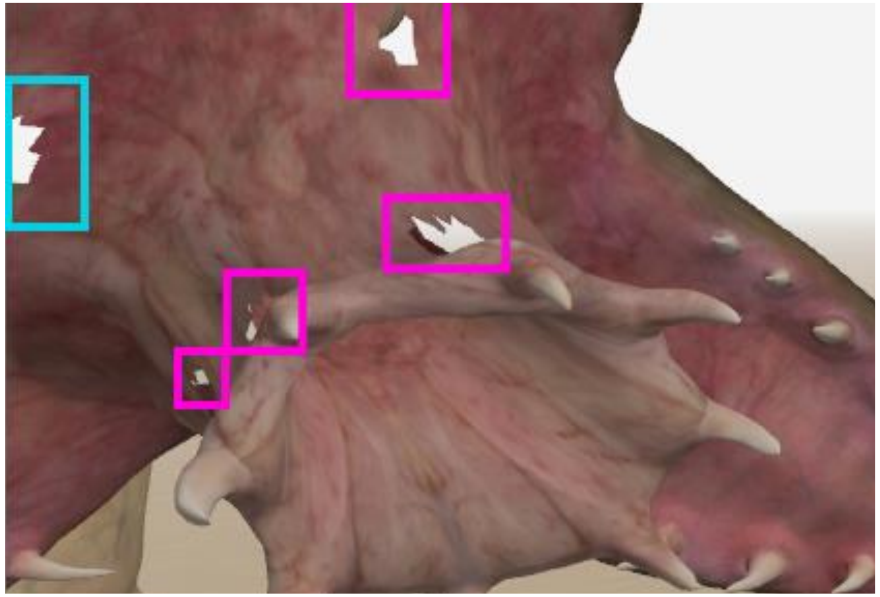
...which is a problem
in terms of its bounds.

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

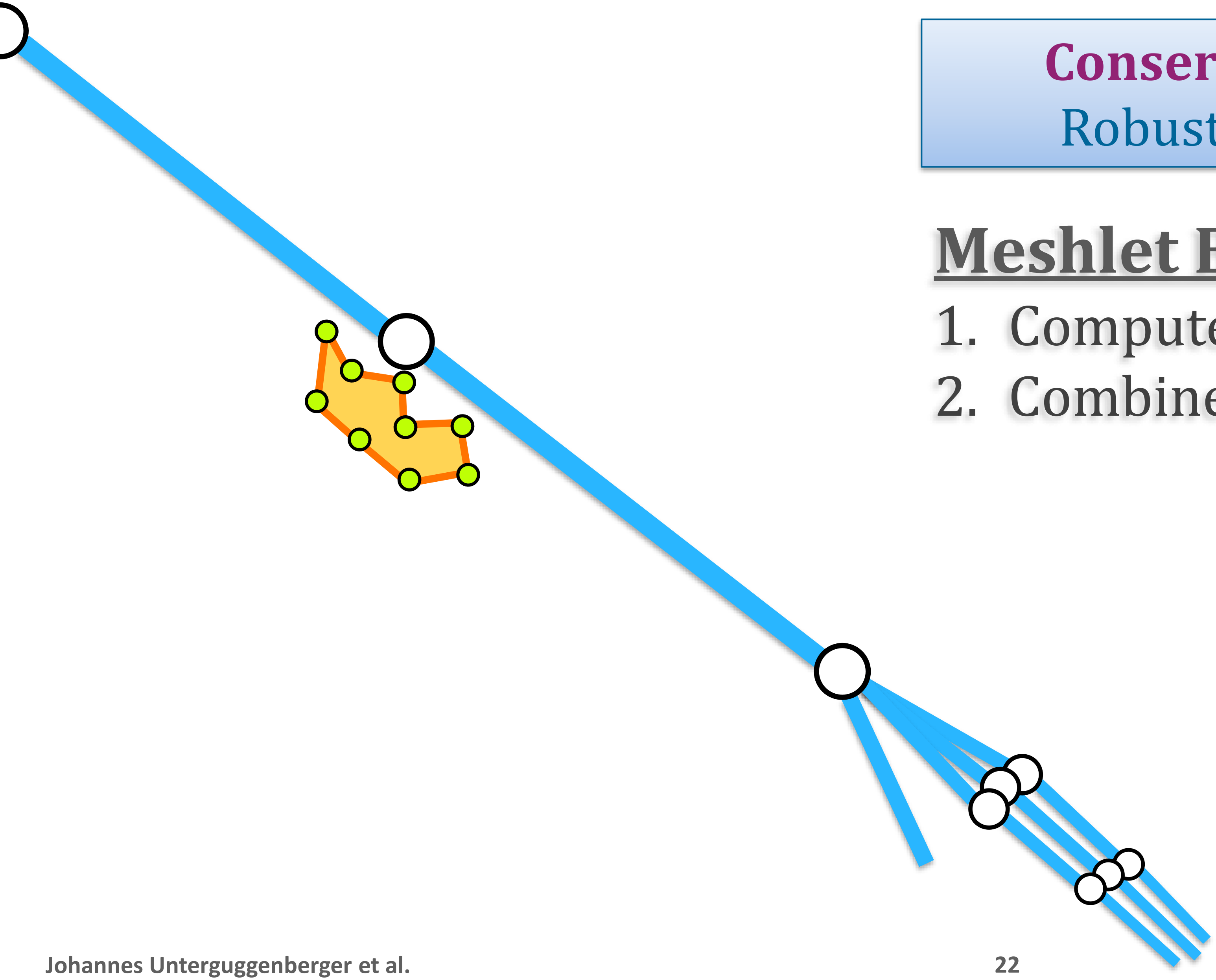


…no longer encompasses all **vertex positions** in the transformed state.

Also, the face normals distribution has changed under animation.



Meshlet Bounds



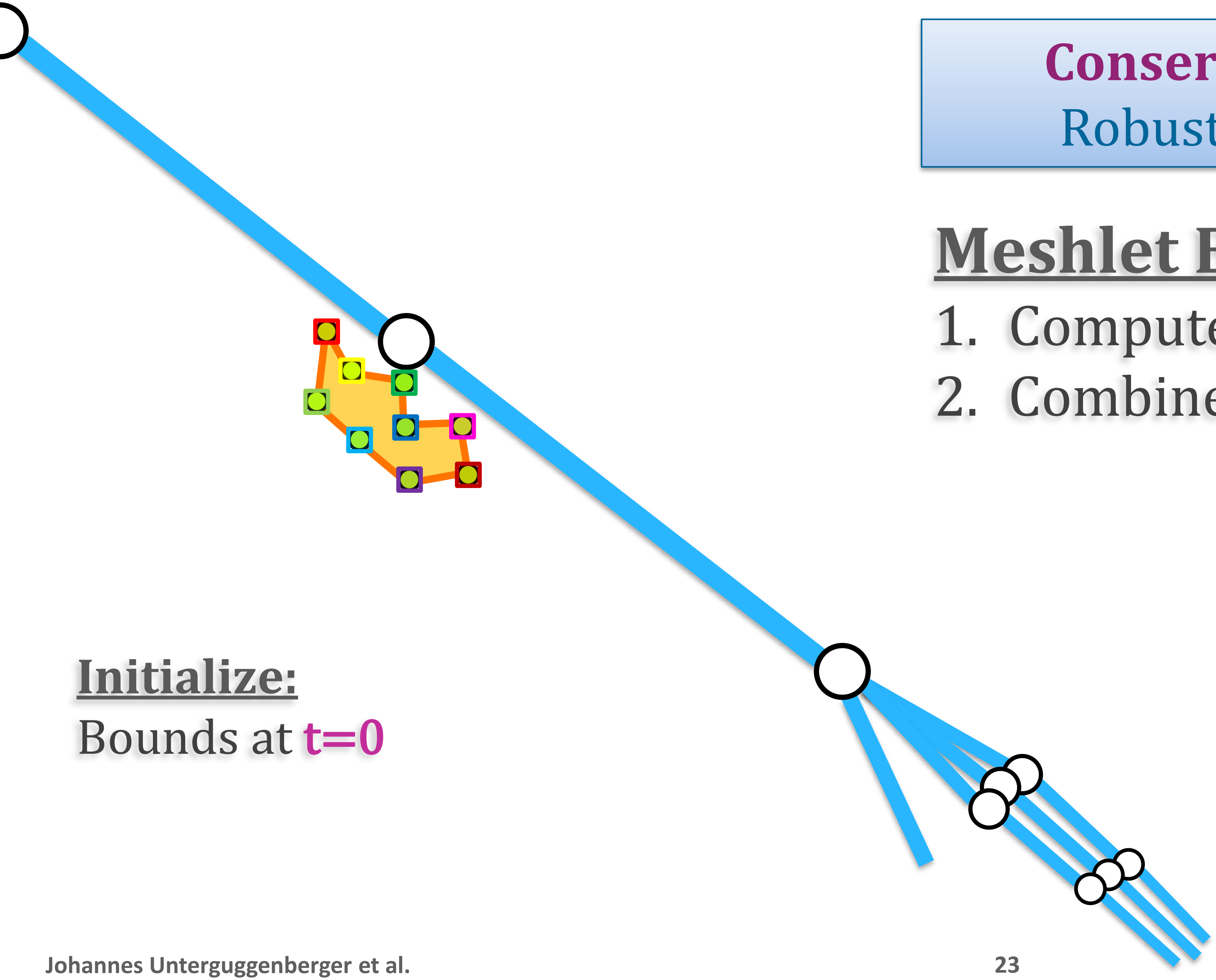
Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

Meshlet Bounds Algorithm:

1. Compute all *vertex bounds*
2. Combine into *meshlet bounds*

Vertex Bounds

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes



Meshlet Bounds Algorithm:

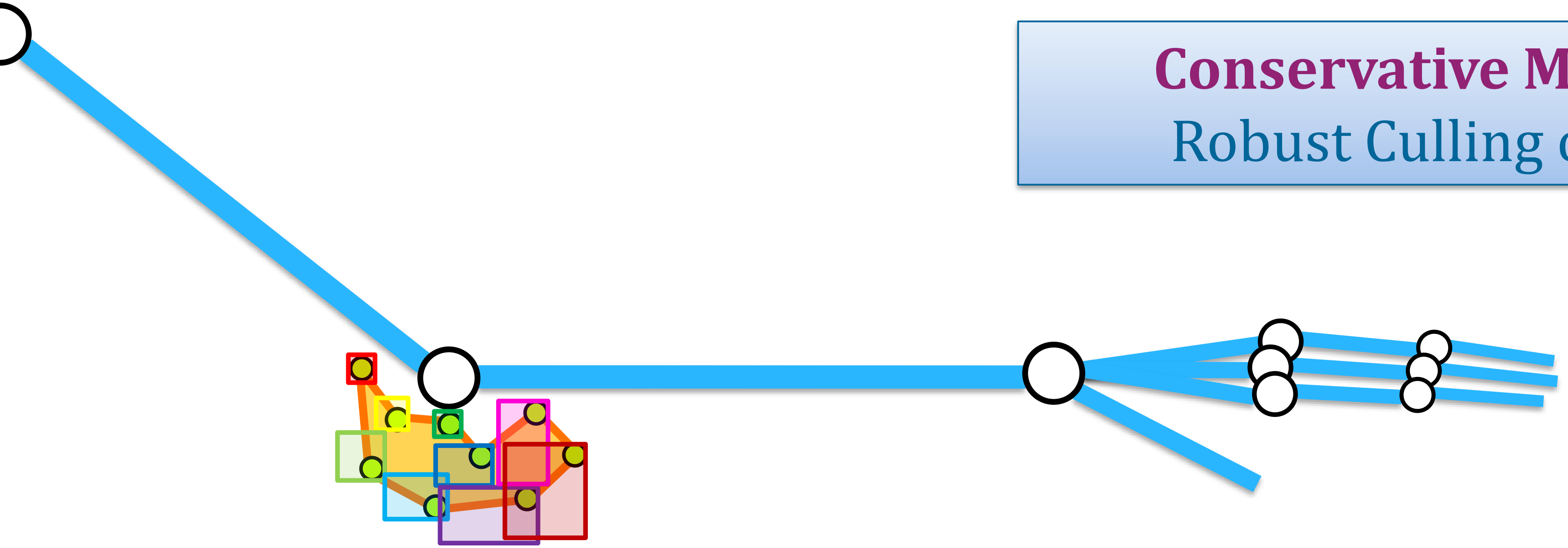
1. Compute all *vertex bounds*
2. Combine into *meshlet bounds*

Initialize:

Bounds at **t=0**

Vertex Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

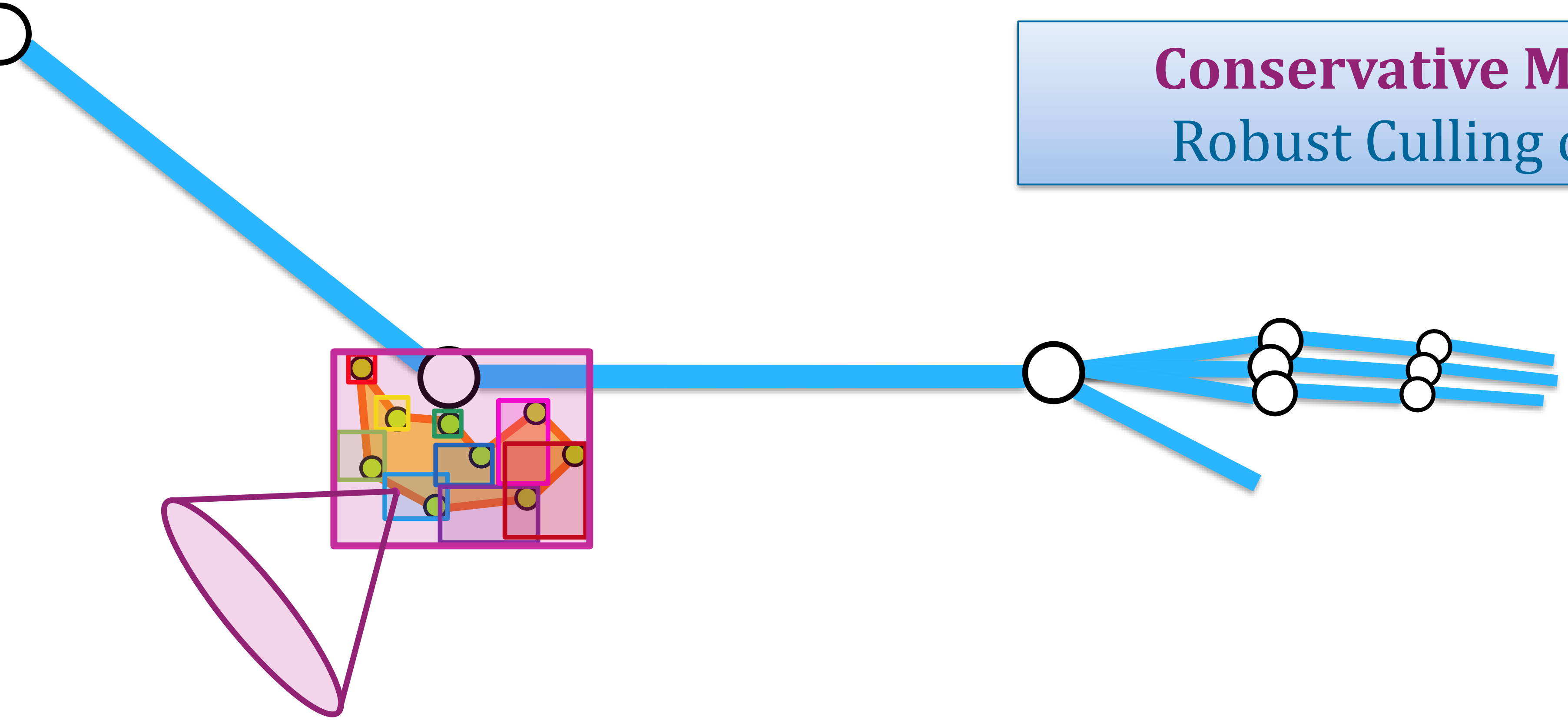


Calculate:

1. Bounds from t=0 to t=1 per vertex

Vertex Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

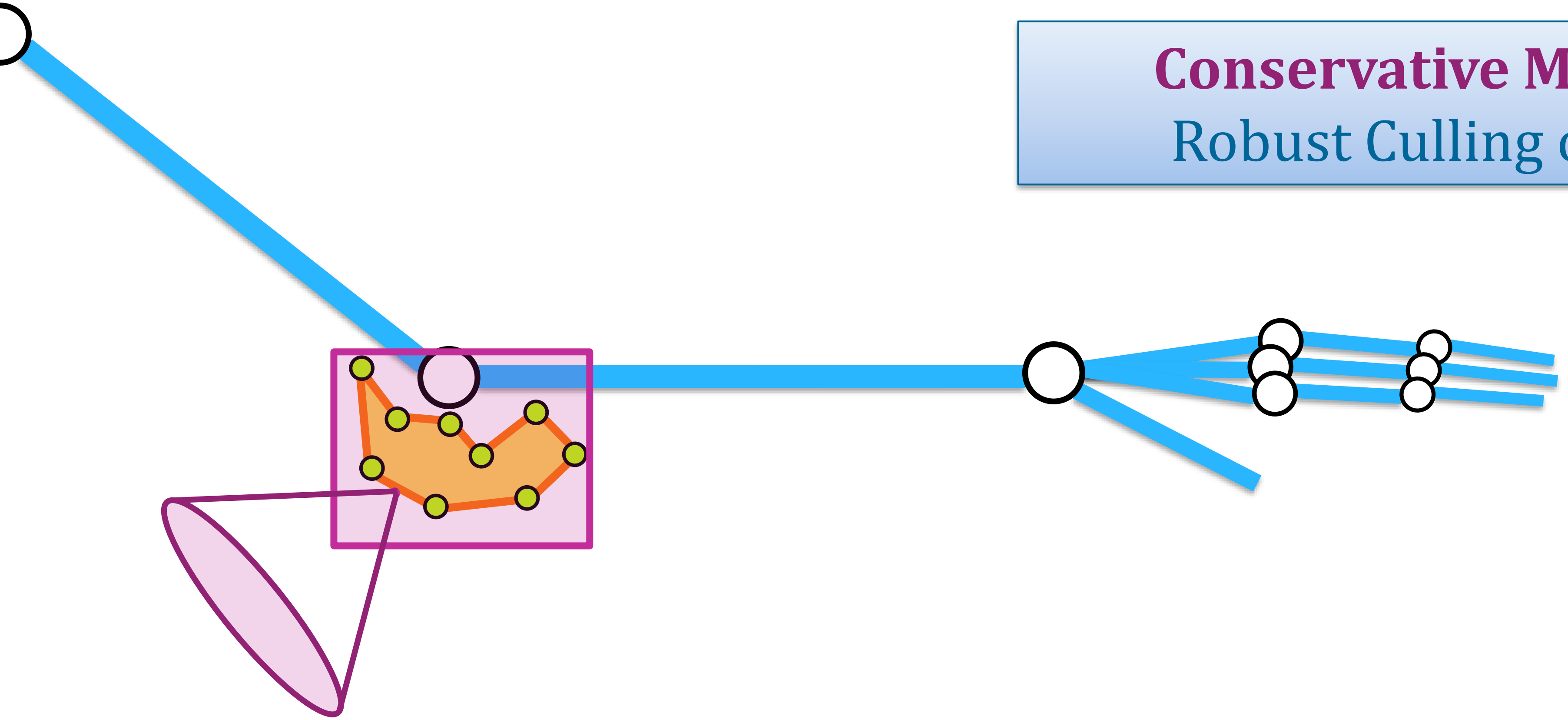


Calculate:

1. Bounds from **t=0** to **t=1** per vertex
2. Meshlet bounds = min and max bounds
   Normals distribution based on vertex bounds, too

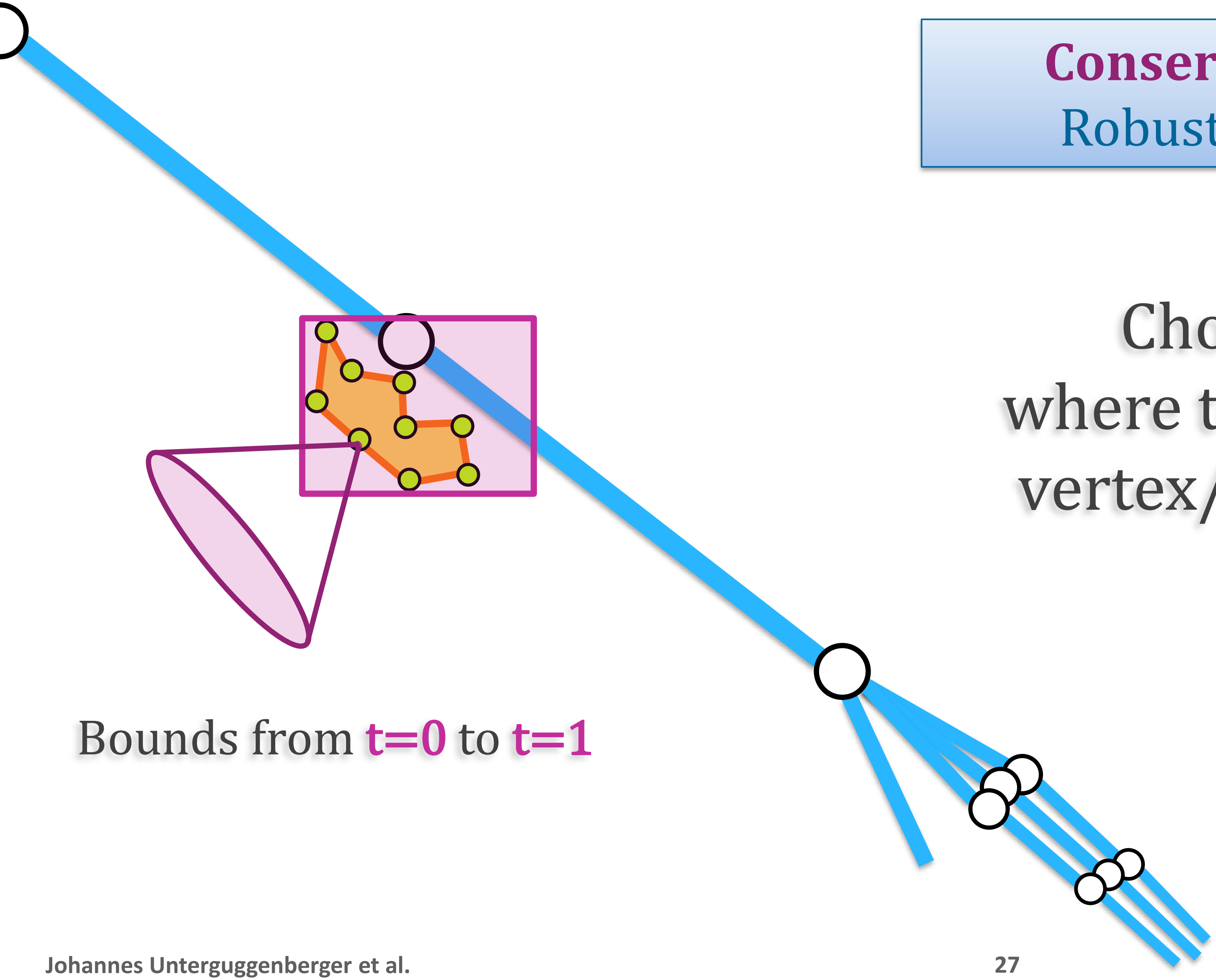
Vertex Bounds

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes



Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

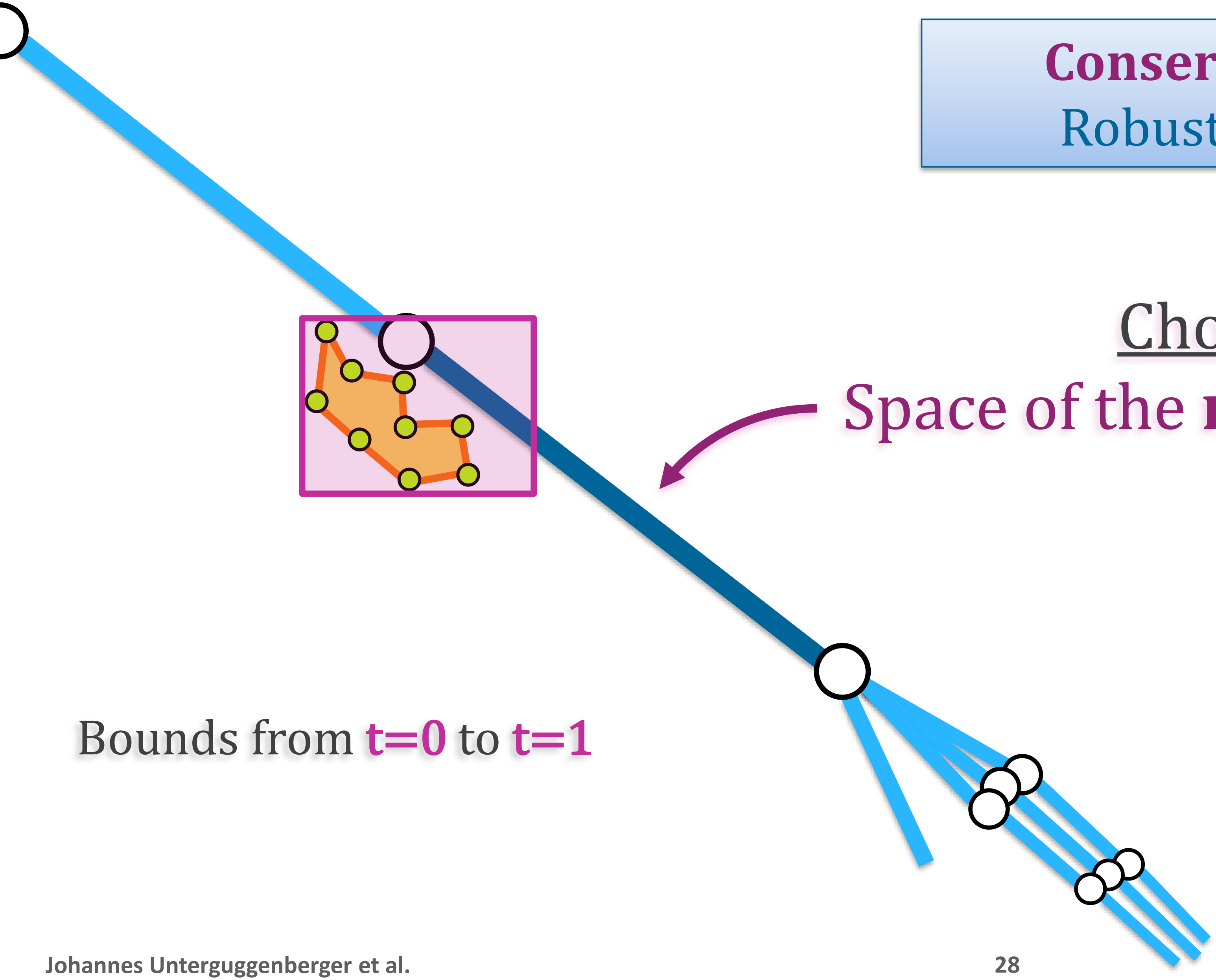


Bounds from **t=0** to **t=1**

Choice of **space**
where to store/calculate
vertex/meshlet bounds

Meshlet Bounds

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes



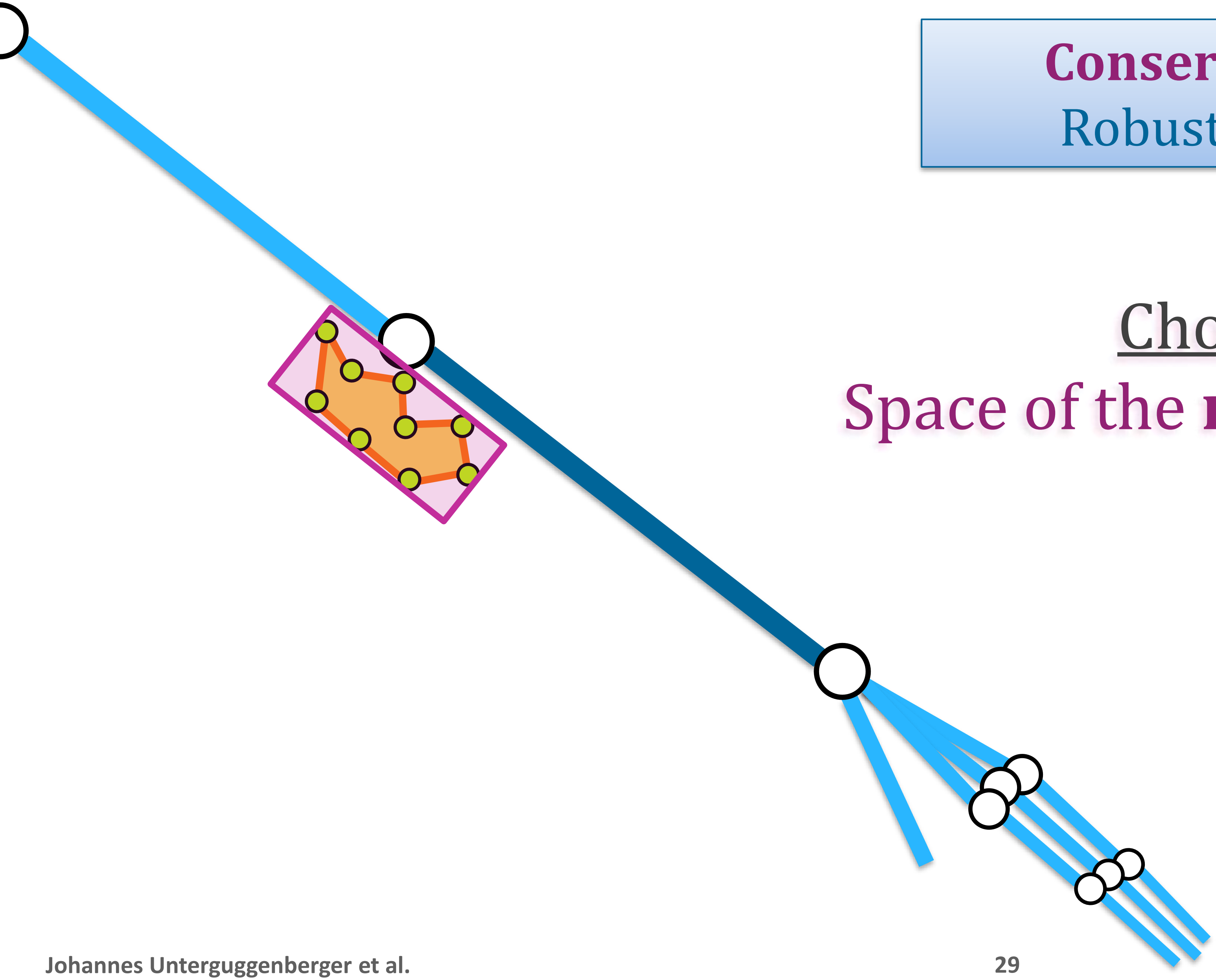
Choice of **space:**

Space of the **most influential bone**

Bounds from t=0 to t=1

Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes



Choice of **space:**

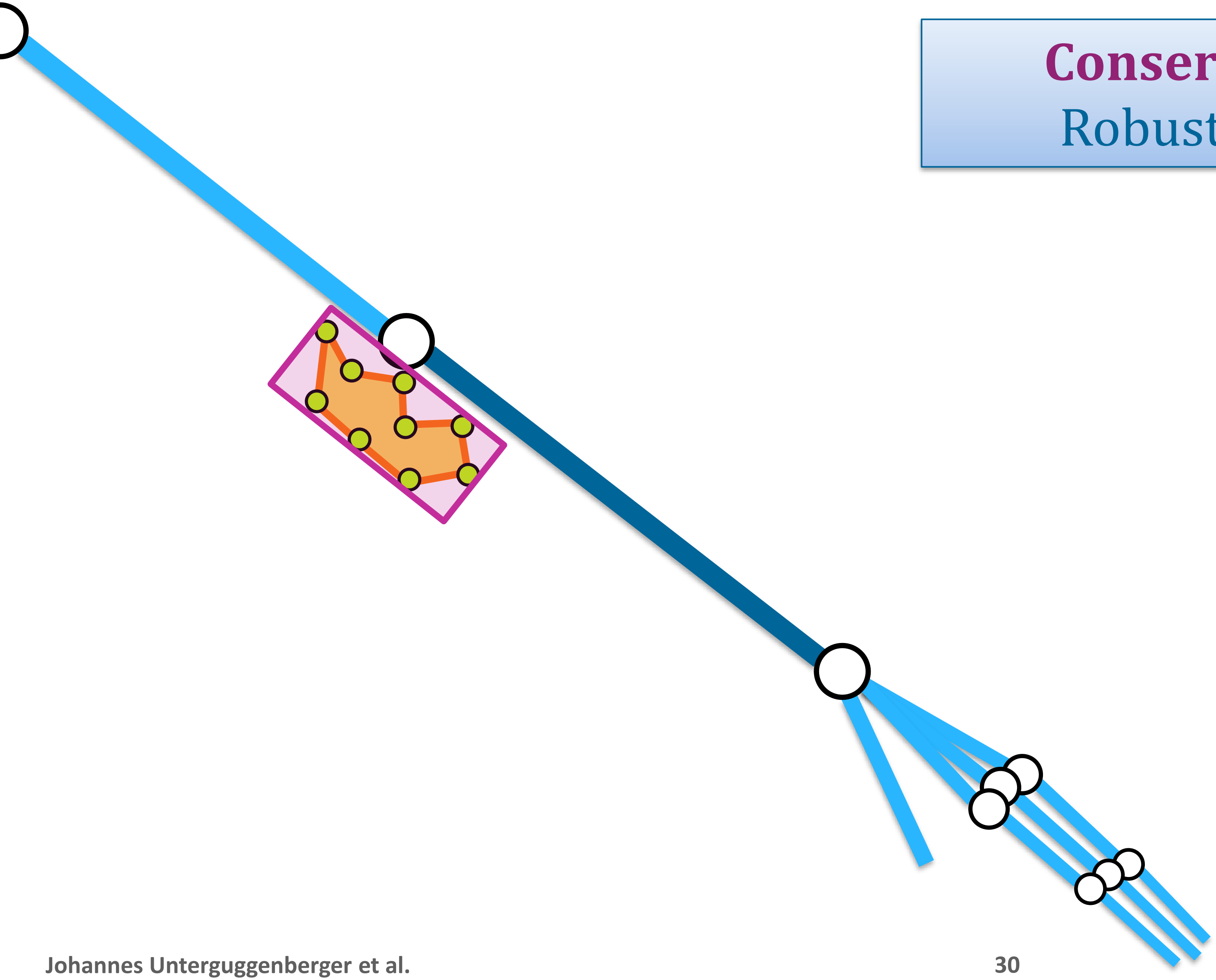
Space of the **most influential bone**

We are going to**:**

- Compute **initial bounds** in that space
- Transform bounds **with** that space

Meshlet Bounds

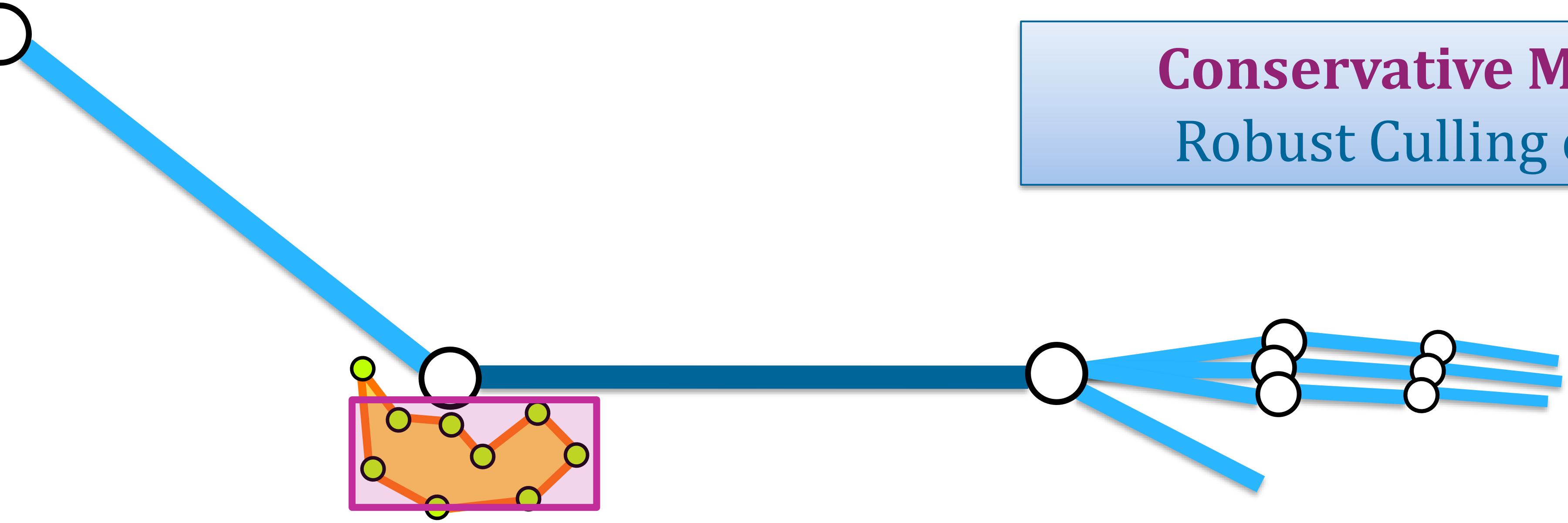
Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes



- Transform bounds **with** that space

Meshlet Bounds

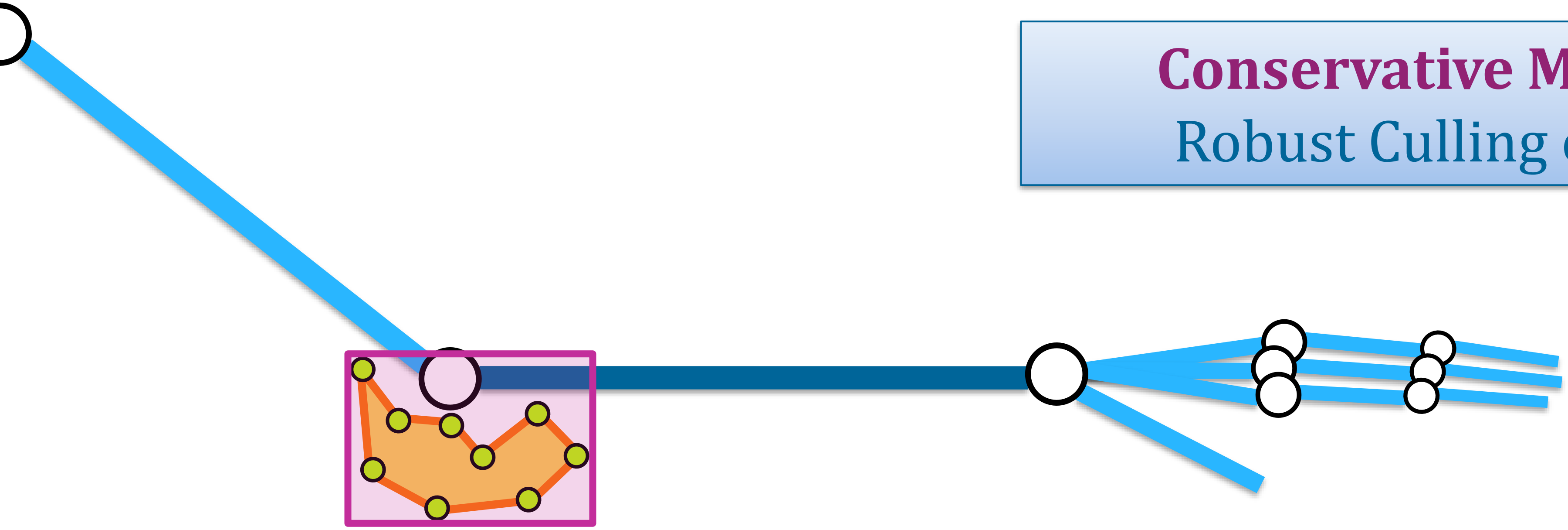
Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes



Calculate bounds for all animation intervals of interest!

- Transform bounds **with** that space

Meshlet Bounds



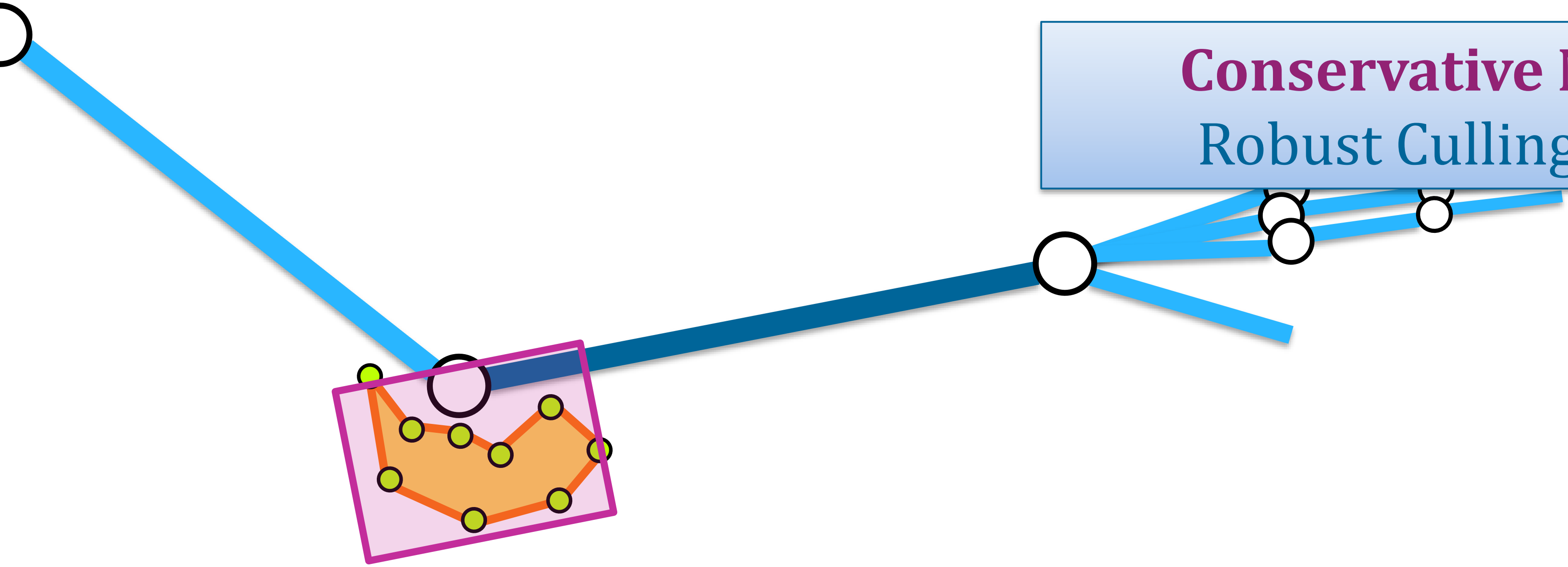
Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

Calculate bounds for all animation intervals of interest!

from **t=0** to **t=1**
from **t=1** to **t=2**

Meshlet Bounds

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes



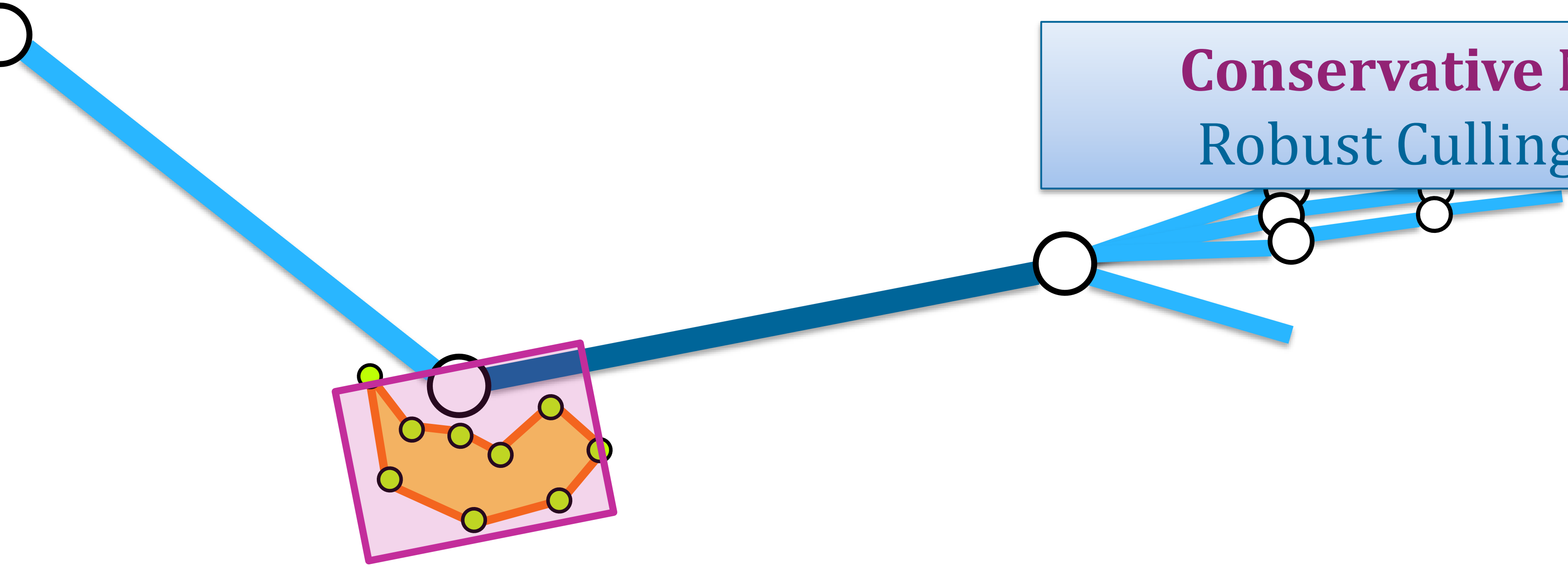
Calculate bounds for all animation intervals of interest!

from t=0 to t=1

from t=1 to t=2

Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

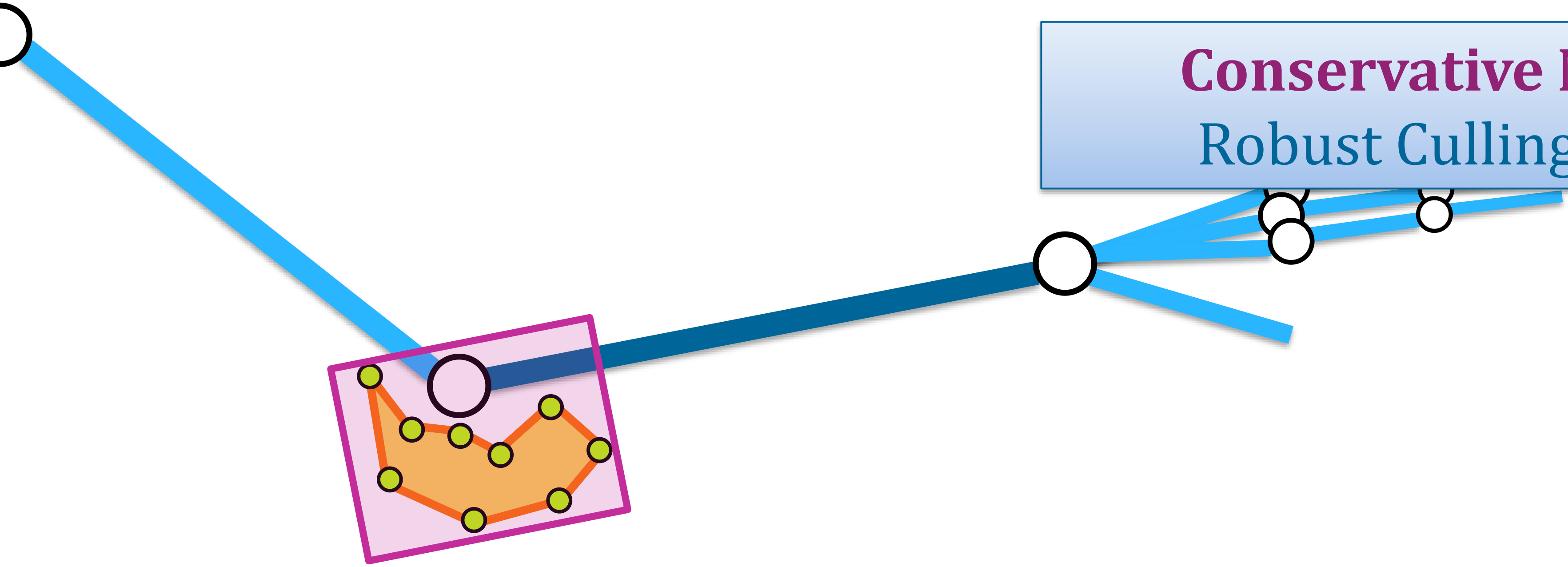


Calculate bounds for all animation intervals of interest!

from t=0 to t=1
from t=1 to t=2

Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

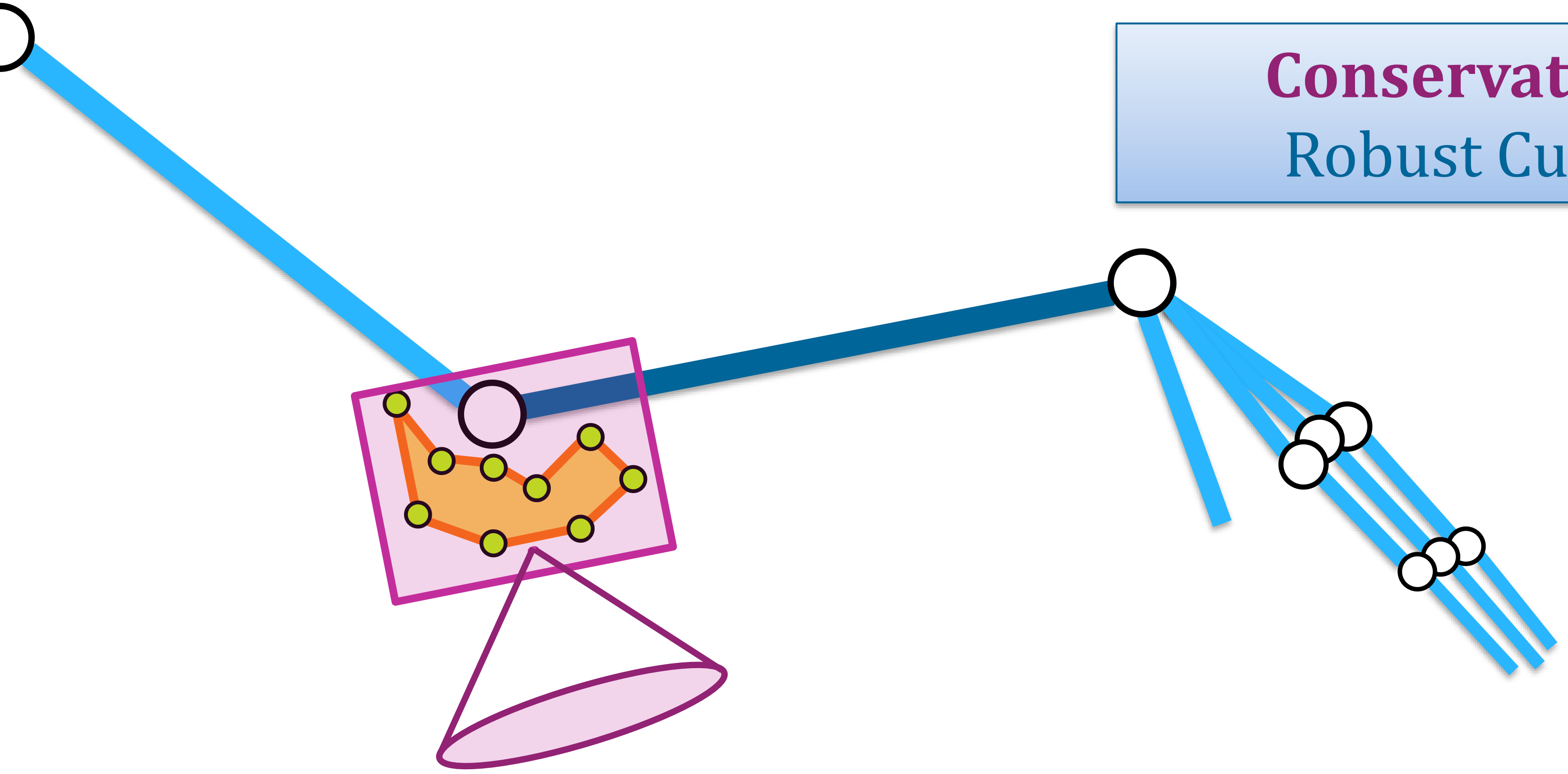


Calculate bounds for all animation intervals of interest!

from t=0 to t=1
from t=1 to t=2

Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes



Calculate bounds for all animation intervals of interest!

from **t=0** to **t=1**

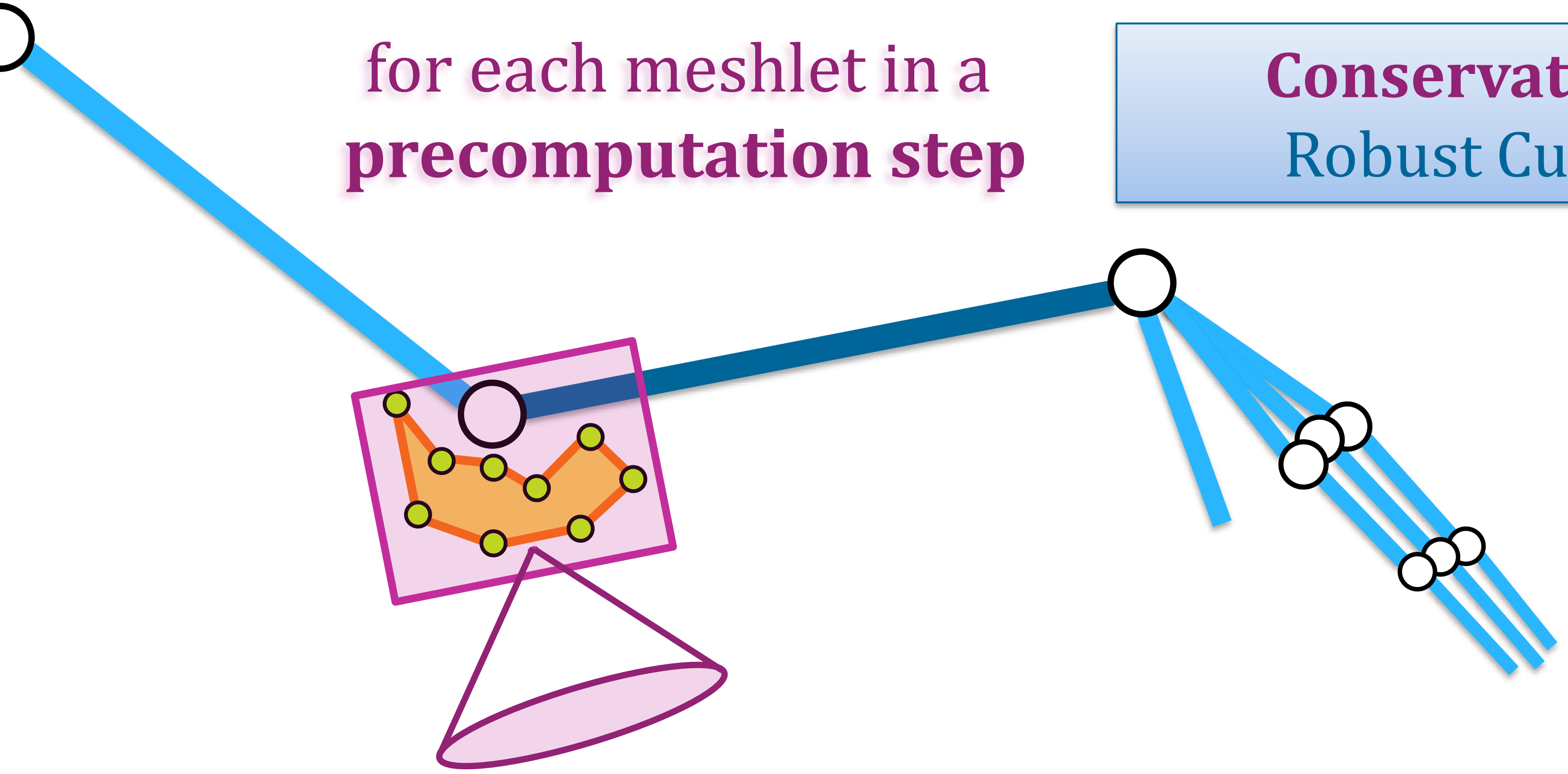
from **t=1** to **t=2**

from **t=2** to **t=3**

Task Shader → Mesh Shader → Rasterizer → Fragment Shader

Meshlet Bounds

for each meshlet in a **precomputation step**

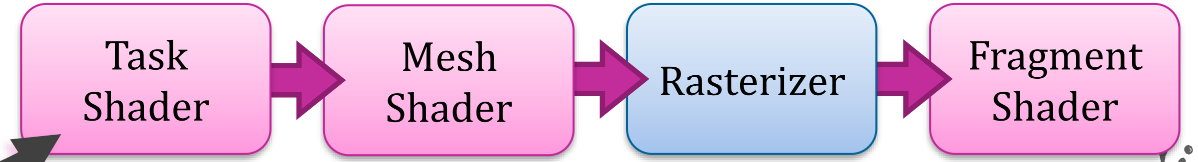


Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

- Low-overhead culling in task shaders
- Low memory consumption
- Low memory bandwidth

from t=0 to t=1
from t=1 to t=2
from t=2 to t=3

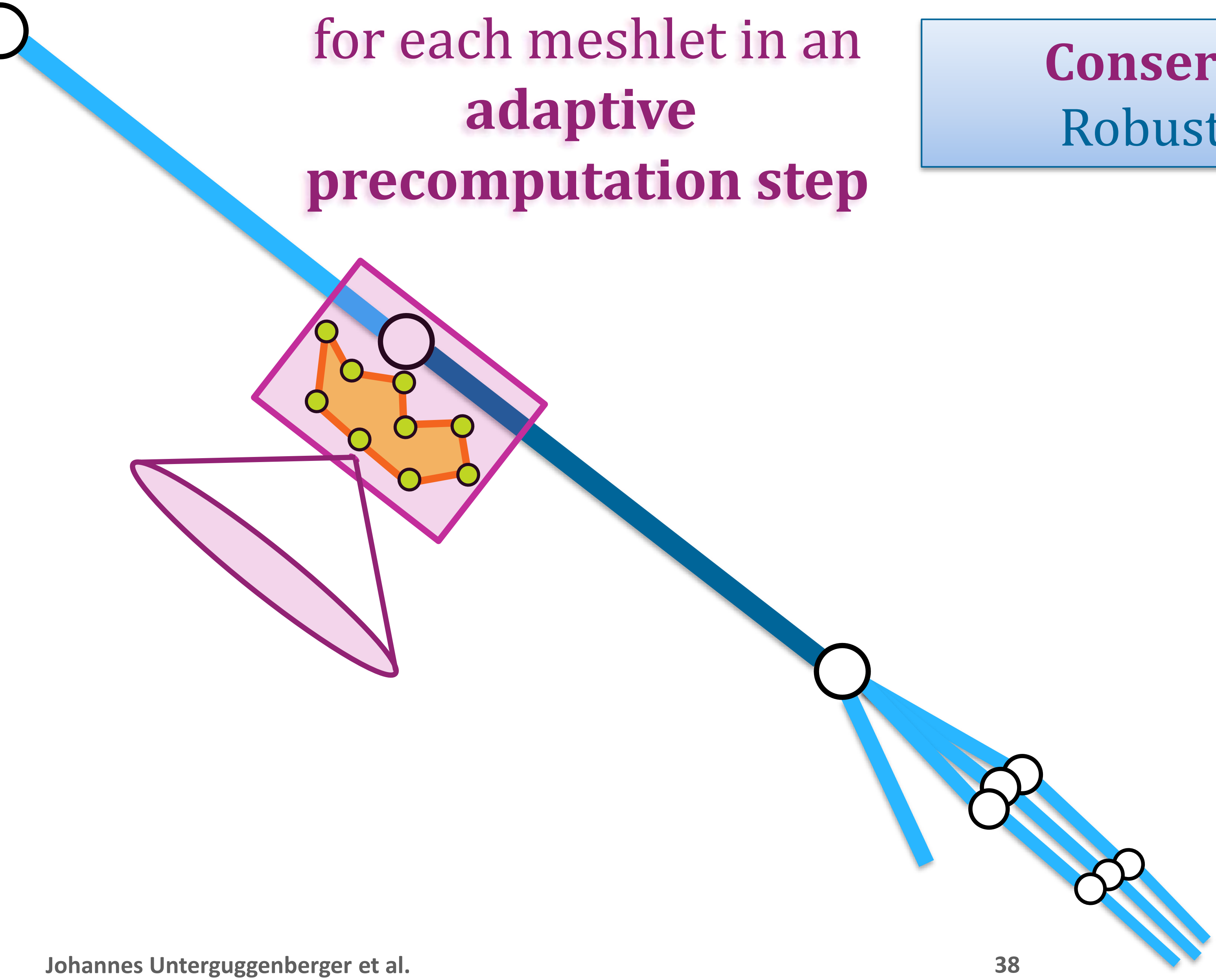
Store 1 bounding box and 1 normals distribution "cone"



Meshlet Bounds

for each meshlet in an **adaptive precomputation step**

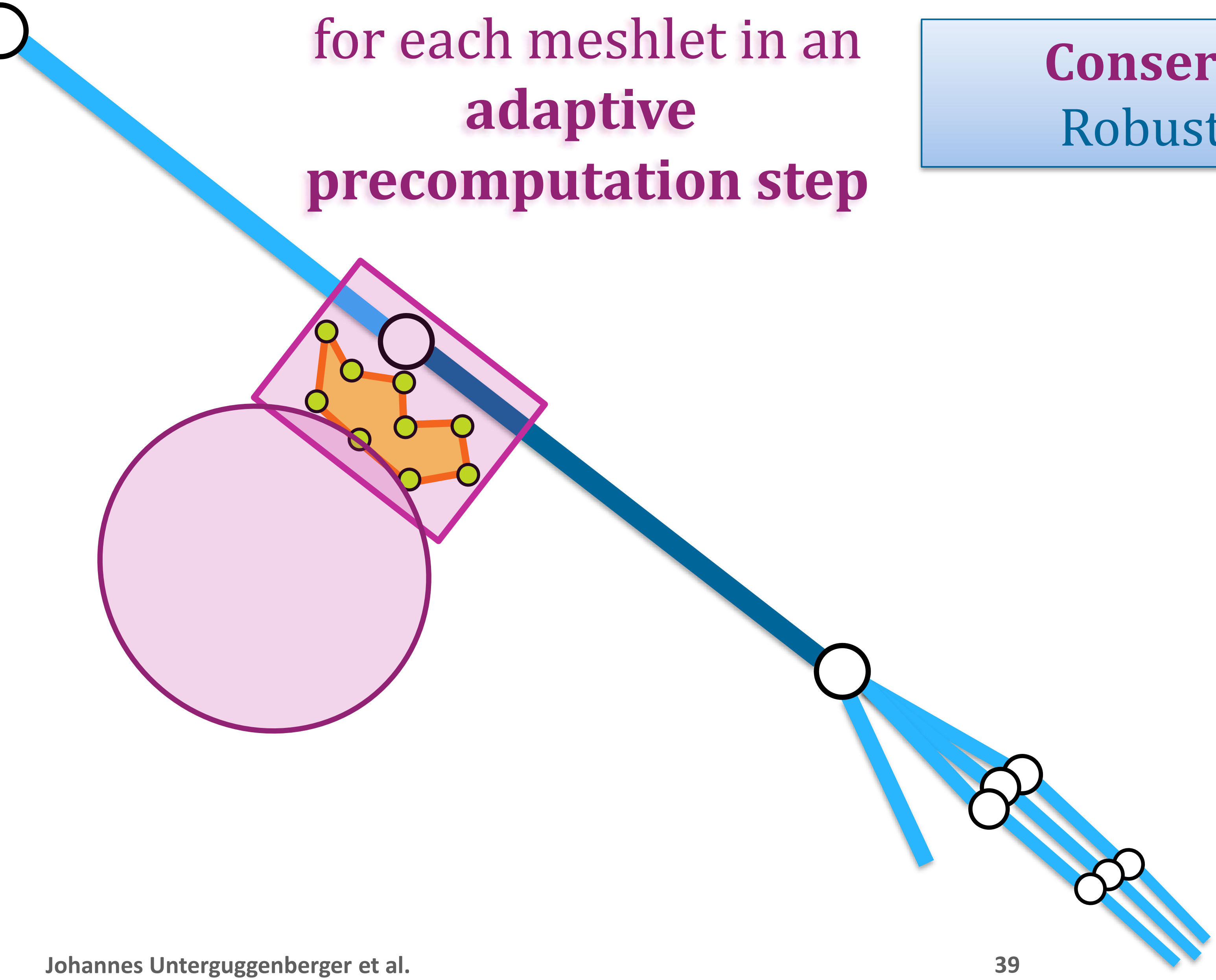
Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

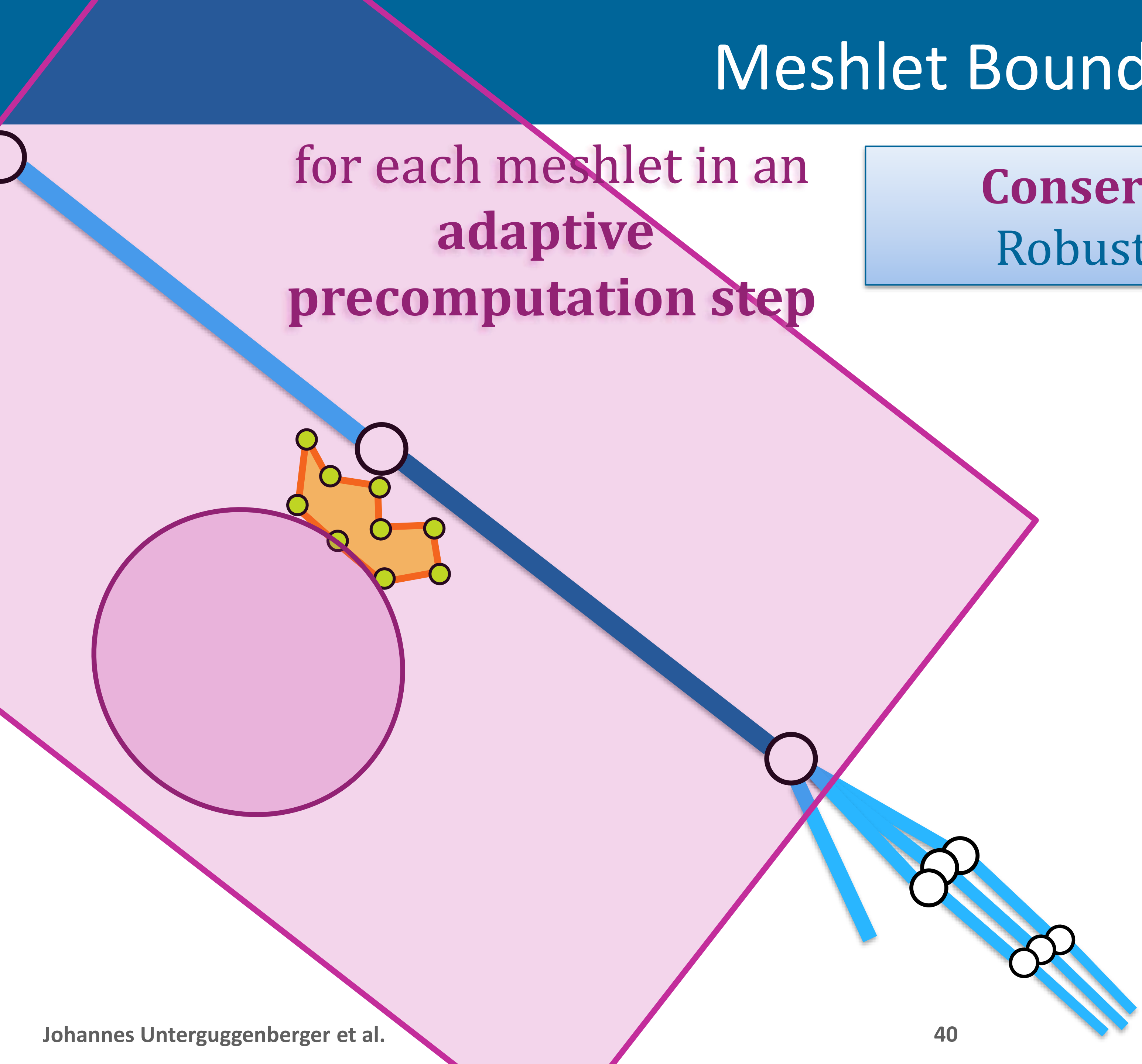


Meshlet Bounds

for each meshlet in an **adaptive precomputation step**

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes





for each meshlet in an **adaptive precomputation step**

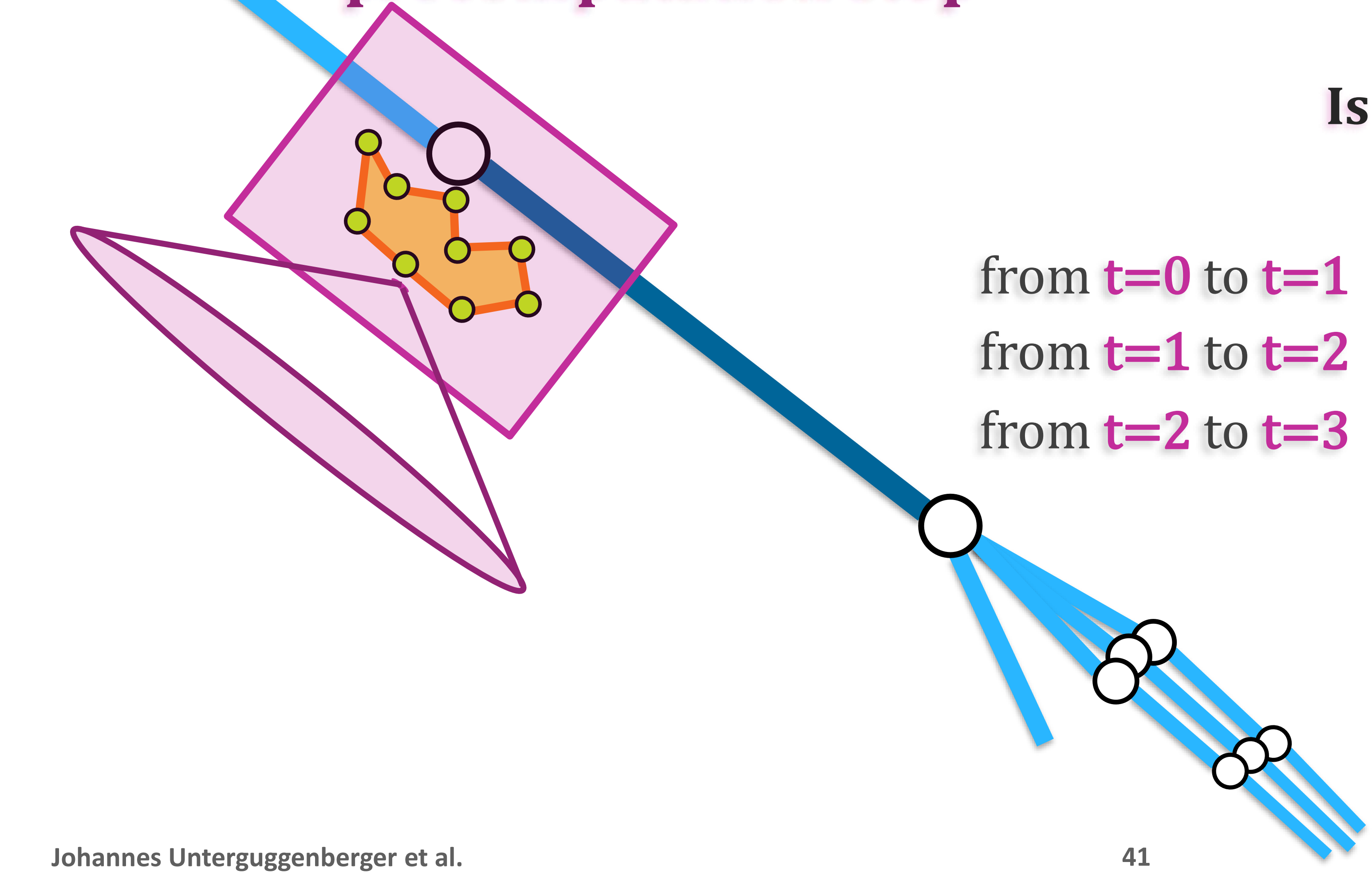
Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Meshlet Bounds

for each meshlet in an **adaptive precomputation step**

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Is this a "well-~~b~~ehaved" meshlet?



from t=0 to t=1

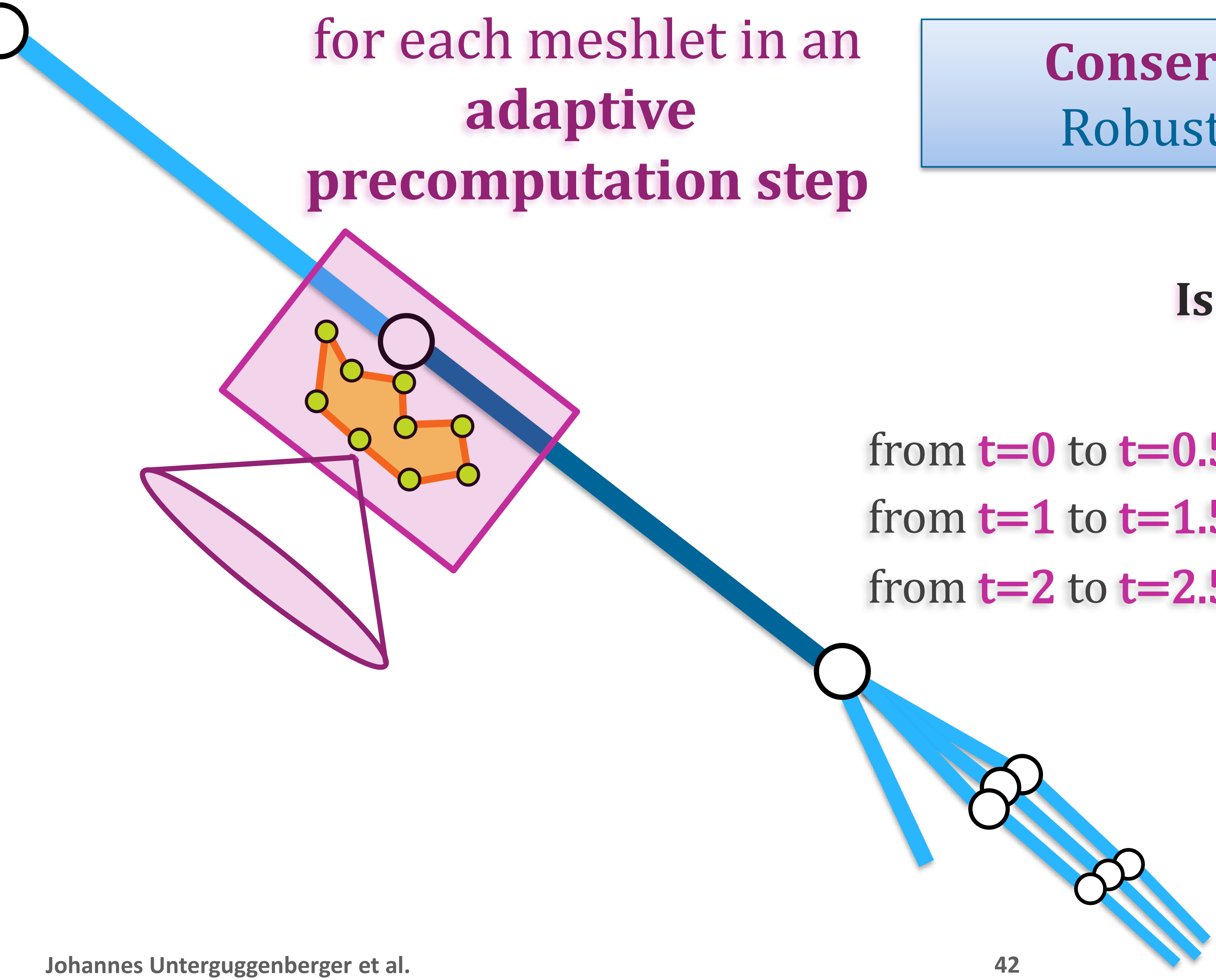
from t=1 to t=2

from t=2 to t=3

Meshlet Bounds

for each meshlet in an **adaptive precomputation step**

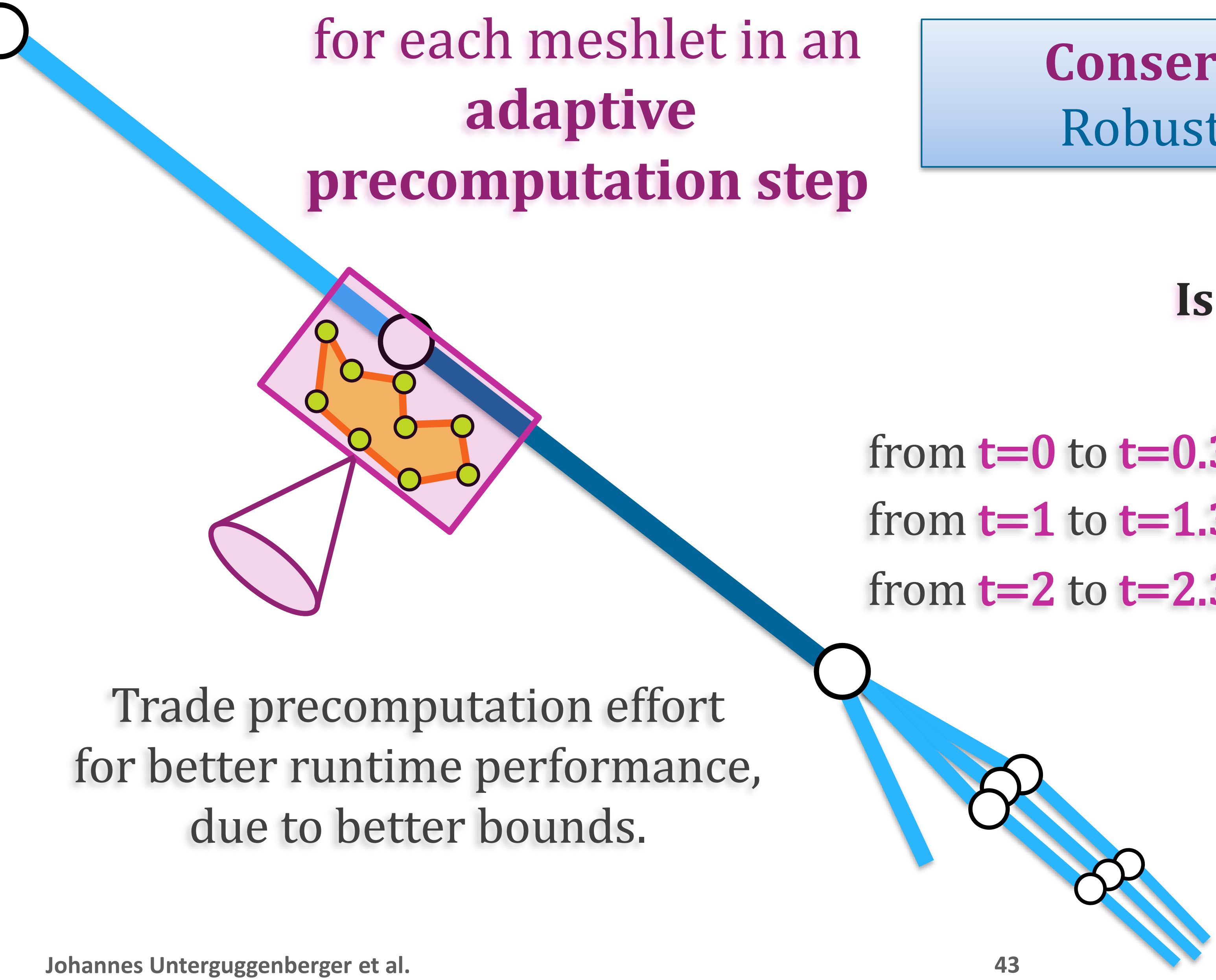
Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes



Is this a "well-~~b~~ehaved" meshlet?

from t=0 to t=0.5 and t=0.5 to t=1
from t=1 to t=1.5 and t=1.5 to t=2
from t=2 to t=2.5 and t=2.5 to t=3

Meshlet Bounds



for each meshlet in an **adaptive precomputation step**

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Is this a "well-~~b~~ehaved" meshlet?

from **$t=0$** to **$t=0.\dot{3}$** and **$t=0.\dot{3}$** to **$t=0.\dot{6}$** and **$t=0.\dot{6}$** to **$t=1$**
from **$t=1$** to **$t=1.\dot{3}$** and **$t=1.\dot{3}$** to **$t=1.\dot{6}$** and **$t=1.\dot{6}$** to **$t=2$**
from **$t=2$** to **$t=2.\dot{3}$** and **$t=2.\dot{3}$** to **$t=2.\dot{6}$** and **$t=2.\dot{6}$** to **$t=3$**

Trade precomputation effort for better runtime performance, due to better bounds.

From our test models:

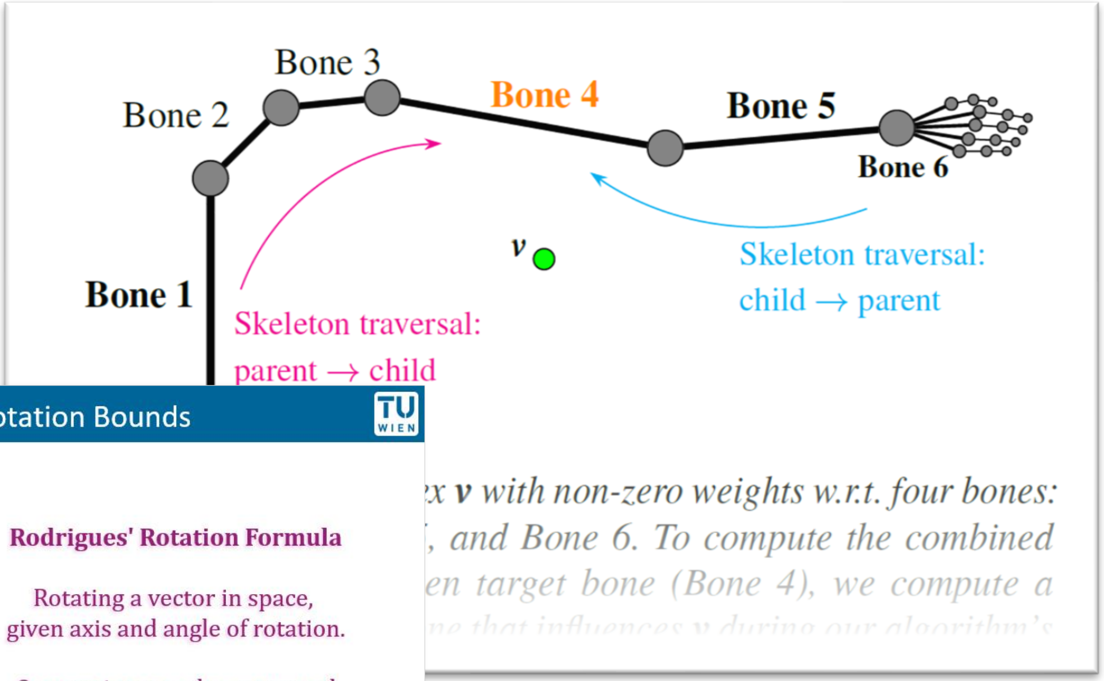
- ≥ 94% view frustum cullable
- ~60%–90% backface cullable

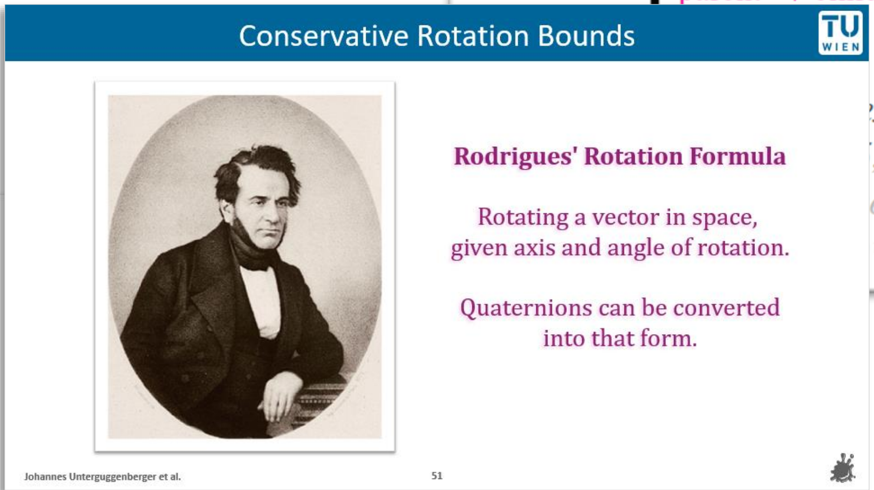
In Our Paper…

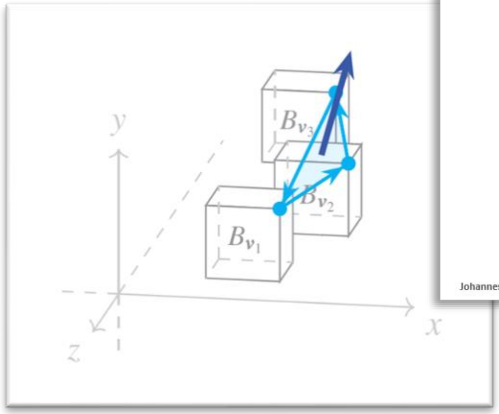
Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Please consult our paper for:

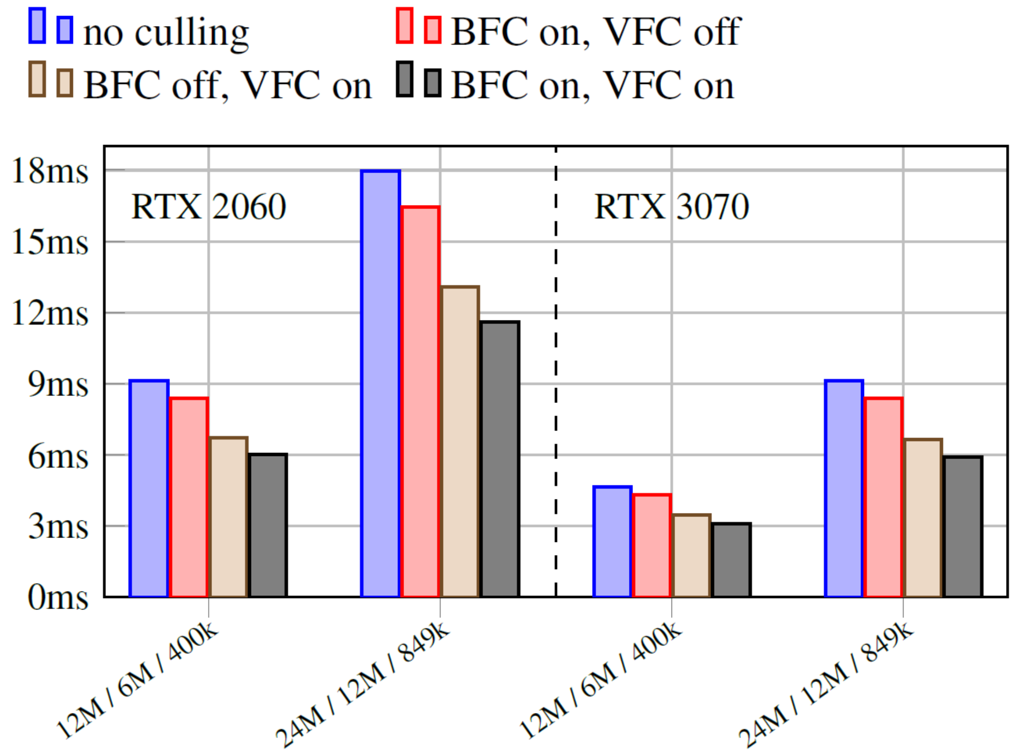
- Vertex bounds **algorithm**
- Conservative rotation bounds extension (Derivative of **Rodrigues' Rotation Formula**)
- Vertex bounds combination for **linear blend skinning**
- Conservative **normal bounds** from vertex bounds
- **Results**, Percentages, …
- Discussion
- Future work





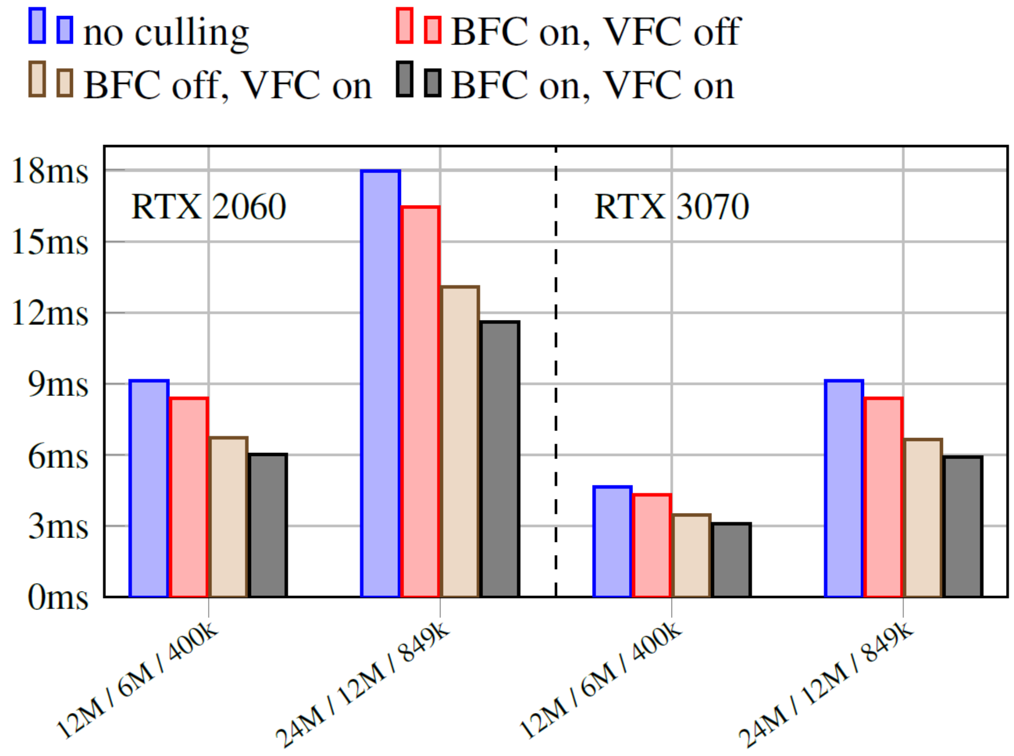


Results



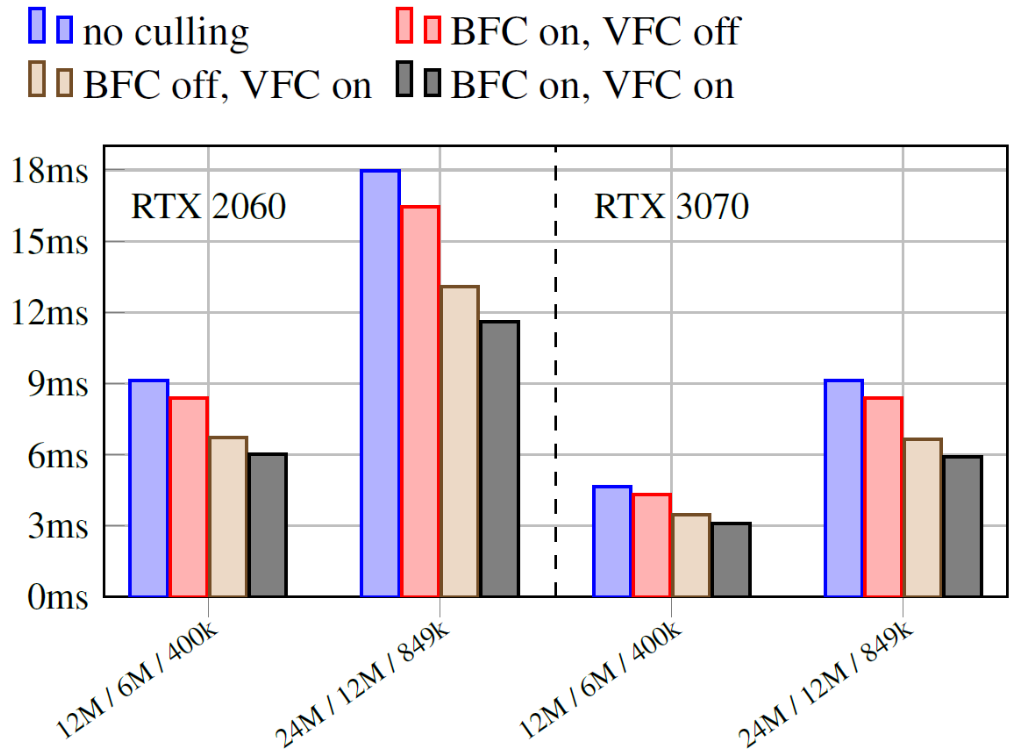
GPU	Scene	BFC only		VFC only		BFC+VFC	
		Culled	Faster	Culled	Faster	Culled	Faster
RTX 2060	400k	11.4%	8.1%	31.3%	26.3%	39.9%	33.8%
	849k	11.5%	8.5%	31.4%	27.1%	39.7%	35.4%
RTX 3070	400k	11.4%	7.8%	31.3%	26.1%	39.9%	34.0%
	849k	11.5%	7.9%	31.4%	27.2%	39.7%	35.4%

Results



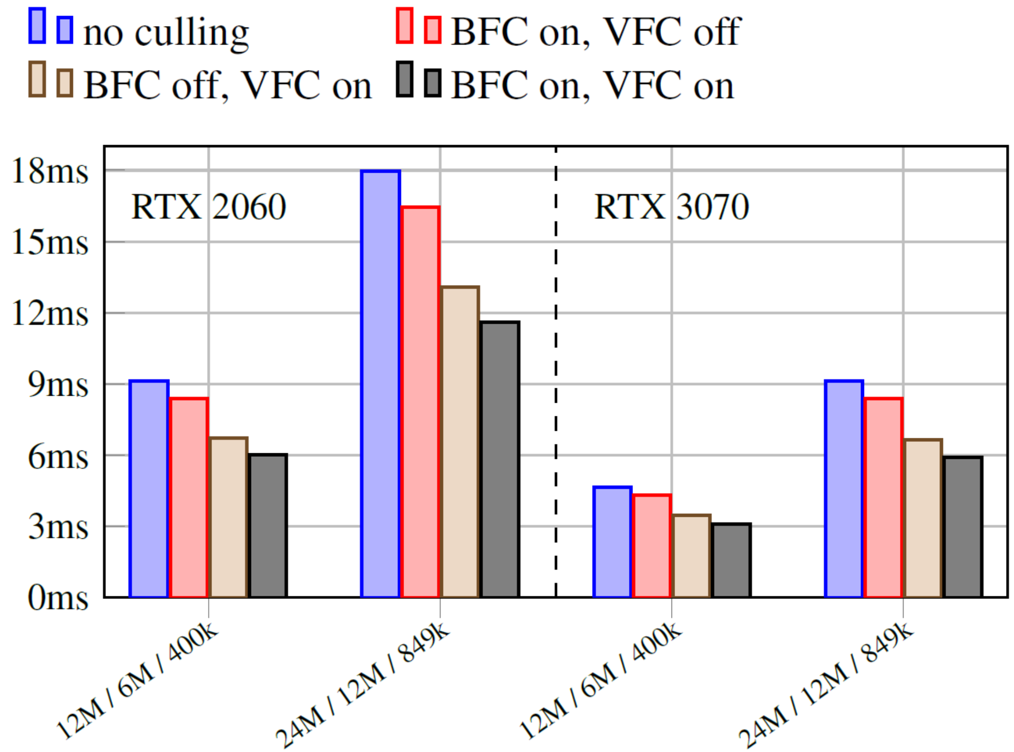
GPU	Scene	BFC only		VFC only		BFC+VFC	
		Culled	Faster	Culled	Faster	Culled	Faster
RTX	400k	11.4%	8.1%	31.3%	26.3%	39.9%	33.8%
2060	849k	11.5%	8.5%	31.4%	27.1%	39.7%	35.4%
RTX	400k	11.4%	7.8%	31.3%	26.1%	39.9%	34.0%
3070	849k	11.5%	7.9%	31.4%	27.2%	39.7%	35.4%

Results



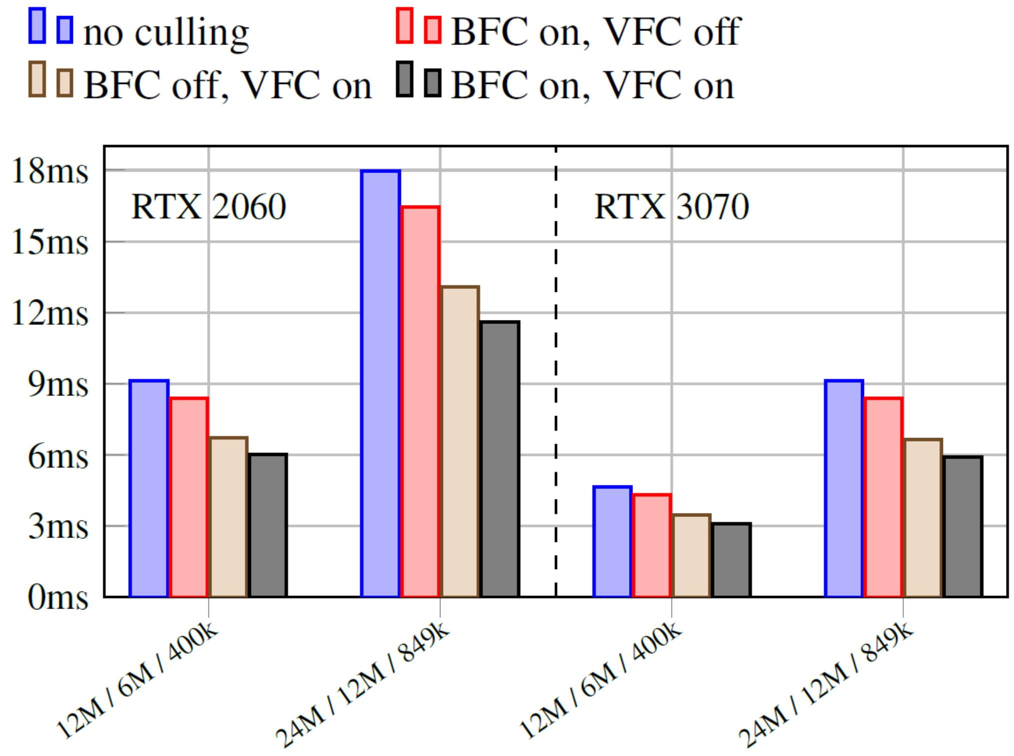
GPU	Scene	BFC only		VFC only		BFC+VFC	
		Culled	Faster	Culled	Faster	Culled	Faster
RTX	400k	11.4%	8.1%	31.3%	26.3%	39.9%	33.8%
2060	849k	11.5%	8.5%	31.4%	27.1%	39.7%	35.4%
RTX	400k	11.4%	7.8%	31.3%	26.1%	39.9%	34.0%
3070	849k	11.5%	7.9%	31.4%	27.2%	39.7%	35.4%

Results



GPU	Scene	BFC only		VFC only		BFC+VFC	
		Culled	Faster	Culled	Faster	Culled	Faster
RTX 2060	400k	11.4%	8.1%	31.3%	26.3%	39.9%	33.8%
	849k	11.5%	8.5%	31.4%	27.1%	39.7%	35.4%
RTX 3070	400k	11.4%	7.8%	31.3%	26.1%	39.9%	34.0%
	849k	11.5%	7.9%	31.4%	27.2%	39.7%	35.4%

Results



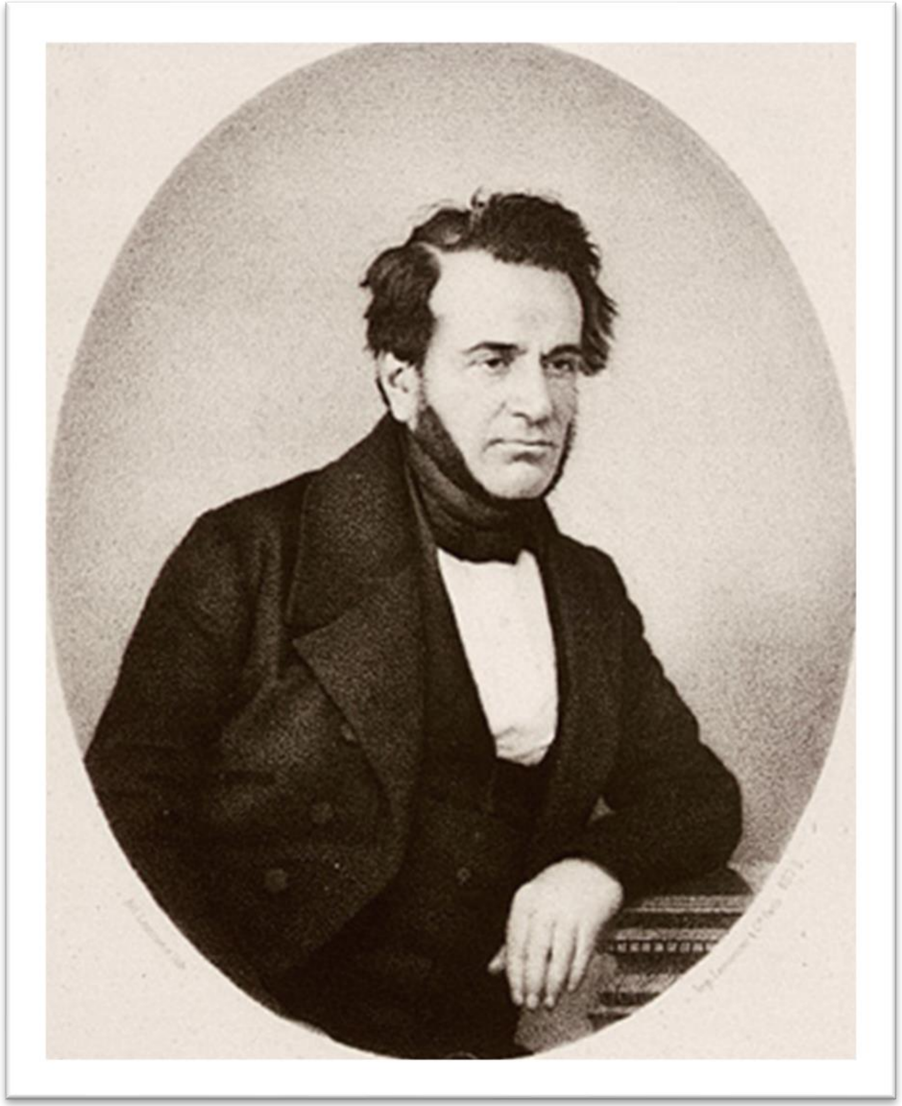
GPU	Scene	BFC only		VFC only		BFC+VFC	
		Culled	Faster	Culled	Faster	Culled	Faster
RTX 2060	400k	11.4%	8.1%	31.3%	26.3%	39.9%	33.8%
	849k	11.5%	8.5%	31.4%	27.1%	39.7%	35.4%
RTX 3070	400k	11.4%	7.8%	31.3%	26.1%	39.9%	34.0%
	849k	11.5%	7.9%	31.4%	27.2%	39.7%	35.4%

Pacific Graphics 2021

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

"Gawain" model © by Unity Technologies, provided through their "The Heretic: Digital Human" package.

Johannes Unterguggenberger, Bernhard Kerbl, Jakob Pernsteiner, and Michael Wimmer
TU Wien, Institute of Visual Computing & Human-Centered Technology, Austria



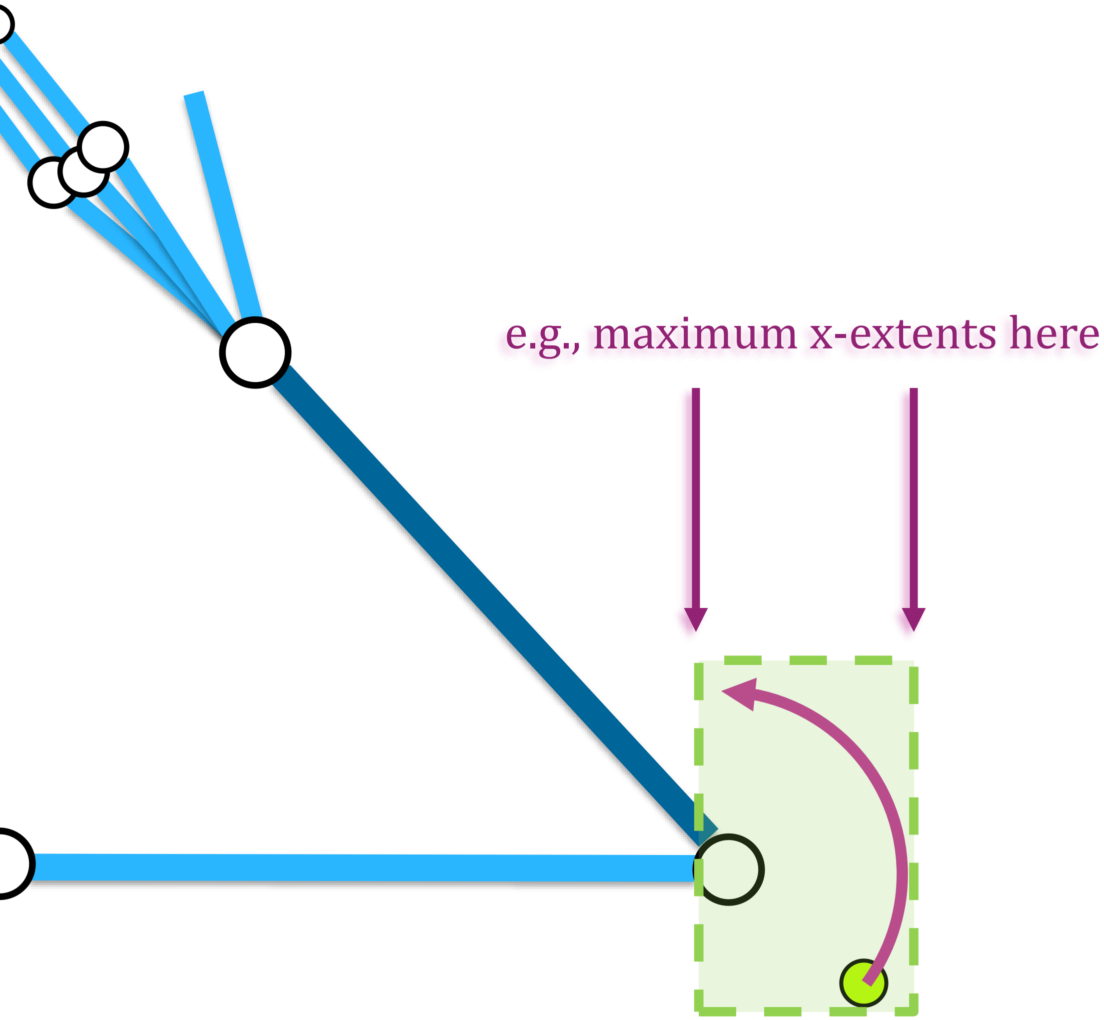
Rodrigues' Rotation Formula

Rotating a vector in space,
given axis and angle of rotation.

Quaternions can be converted
into that form.

Conservative Rotation Bounds

Conservative Rotation Bounds



$$\boldsymbol{v}' = \boldsymbol{v}\cos\theta + (\boldsymbol{n}\times\boldsymbol{v})\sin\theta + \boldsymbol{n}(\boldsymbol{n}\cdot\boldsymbol{v})(1-\cos\theta). \quad (3)$$

We use its first-order derivative by θ to find those angles that lead to maximum extents in each of the principal axes' directions. Setting that first-order derivative of Equation (3) by θ to zero in order to find the extrema results in Equation (4)

$$\boldsymbol{x}_\theta = -\tan^{-1}\frac{\boldsymbol{n}\times\boldsymbol{v}}{\boldsymbol{n}(\boldsymbol{n}\cdot\boldsymbol{v})-\boldsymbol{v}}, \quad (4)$$

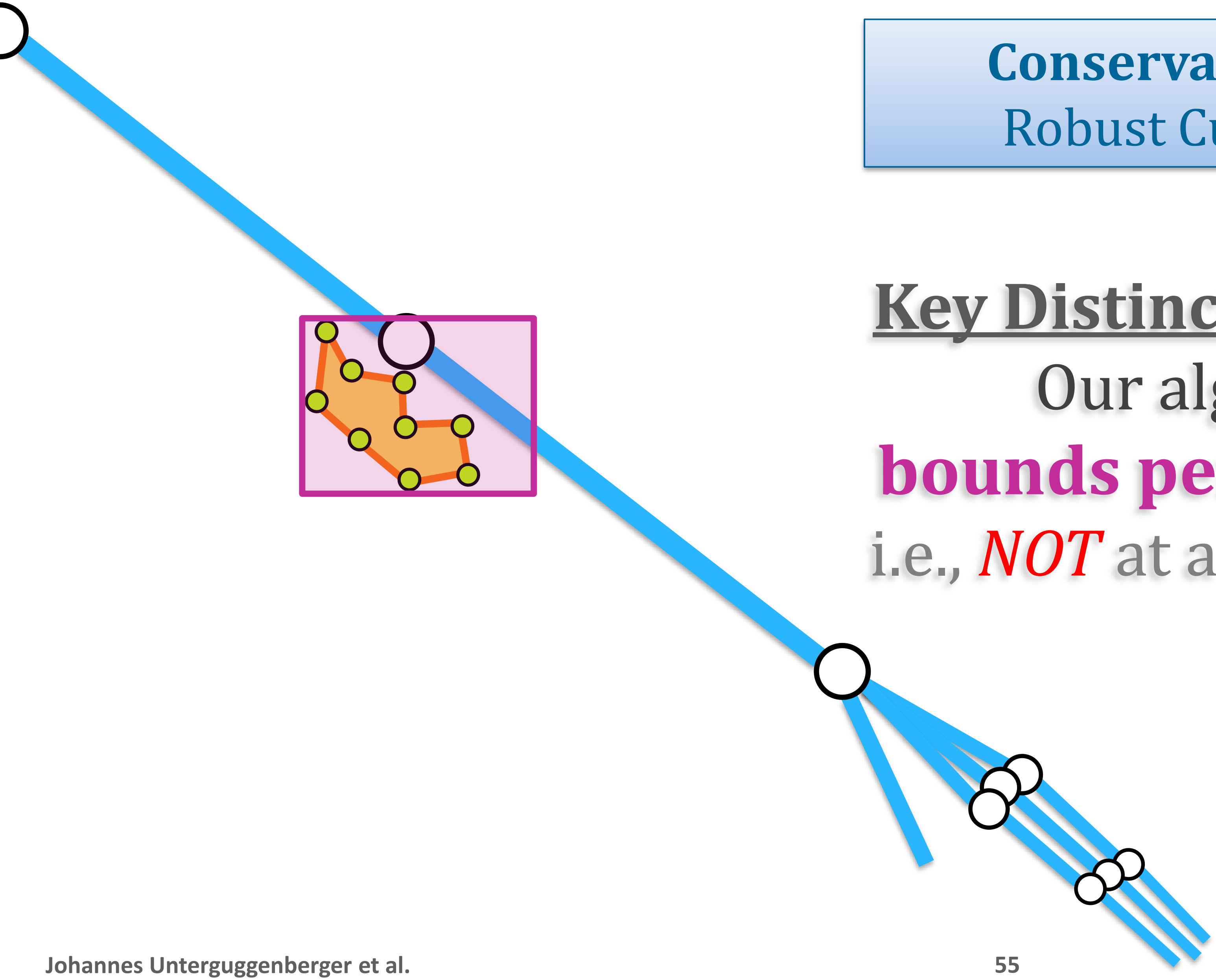
which yields a vector of angles $\boldsymbol{x}_\theta$ in radians that represents the rotation angles which lead to maximum extents in each principal axis direction. Please note that the operations in Equation (4) mean component-wise application of the division and $\tan^{-1}$.

Appendix

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

Johannes Unterguggenberger, Bernhard Kerbl, Jakob Pernsteiner, and Michael Wimmer
TU Wien, Institute of Visual Computing & Human-Centered Technology, Austria

Vertex Bounds

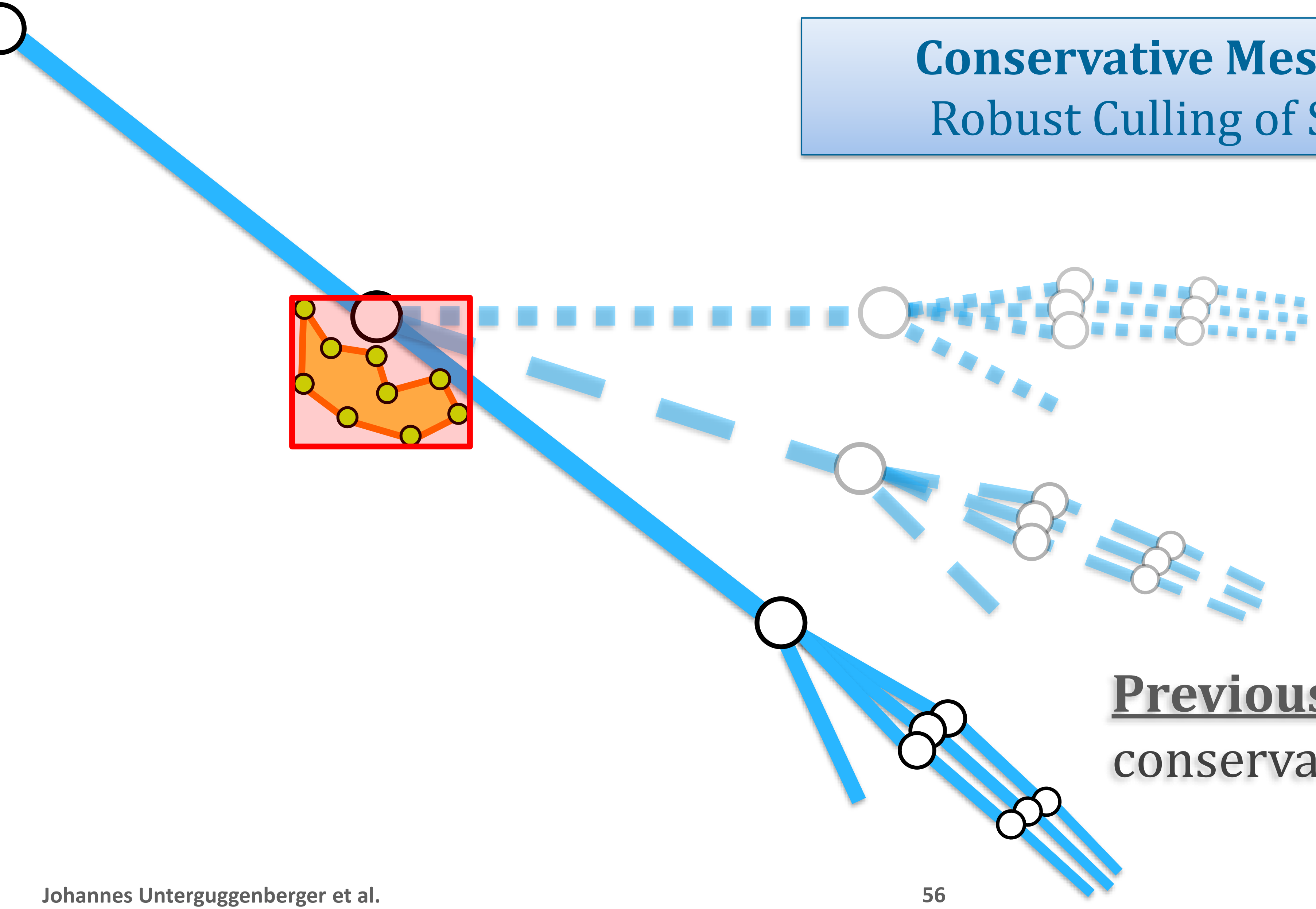


Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

Key Distinction to Previous Work:

Our algorithm computes
bounds per animation interval,
i.e., *NOT* at a specific animation time.

Meshlet Bounds



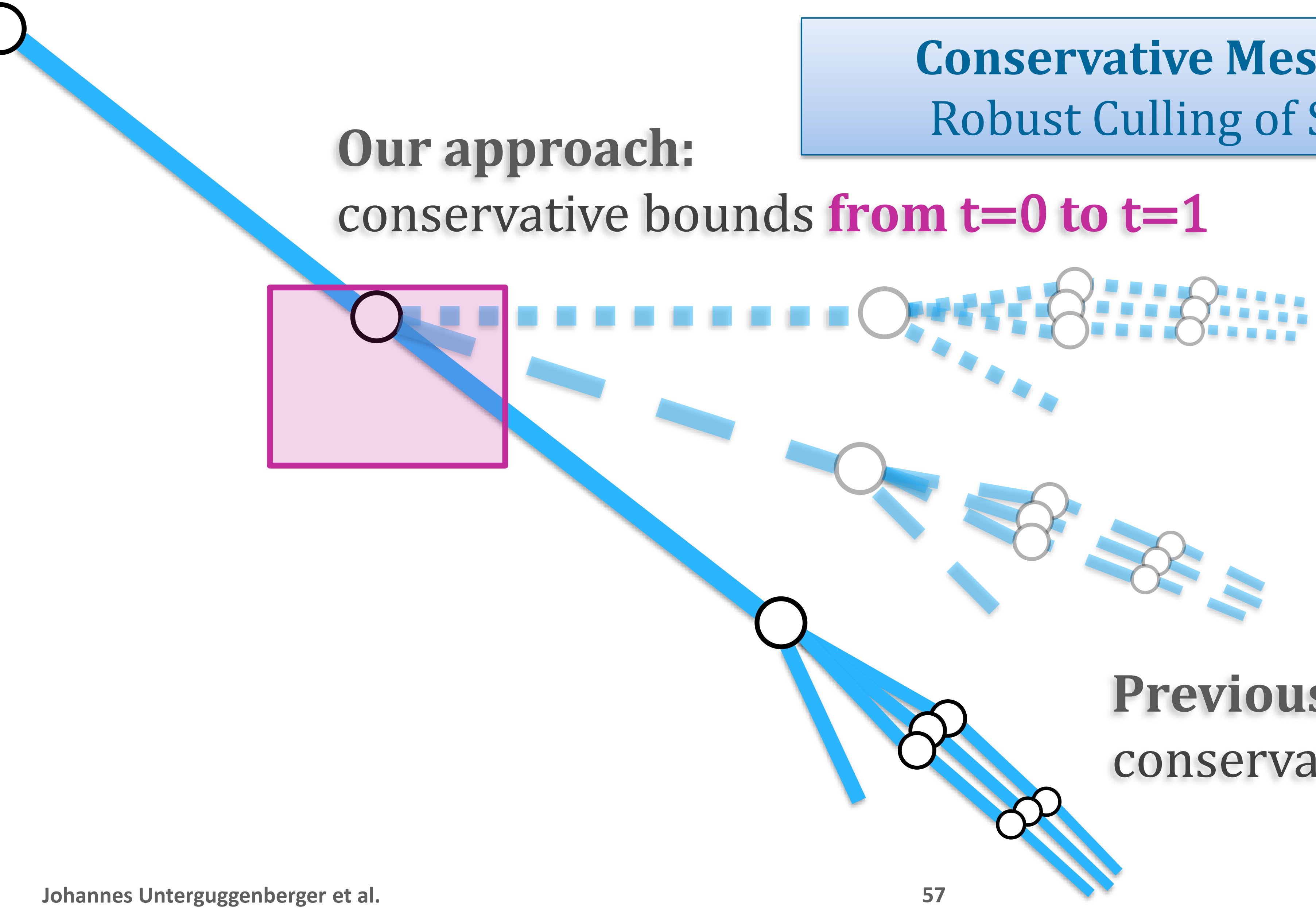
Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Previous approaches:
conservative bounds **at** t=0.5

Meshlet Bounds

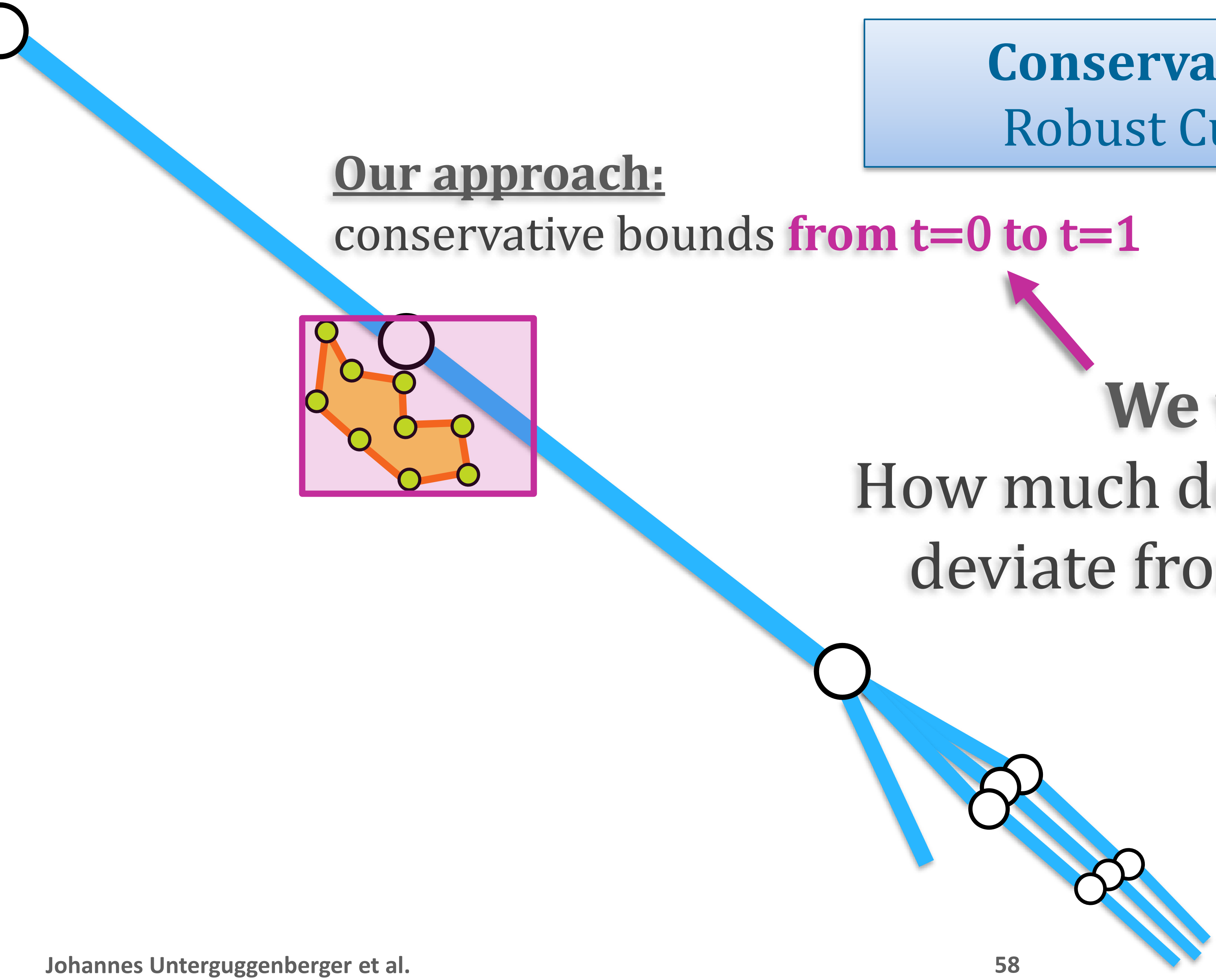
Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

Our approach:
conservative bounds **from t=0 to t=1**



Previous approaches:
conservative bounds **at t=0.5**

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes



Our approach:
conservative bounds **from t=0 to t=1**

We want to know:
How much do **any** animated bounds deviate from an initial position?

Previous approaches:
conservative bounds **at t=0.5**

Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

Our approach:
conservative bounds **from t=0 to t=1**

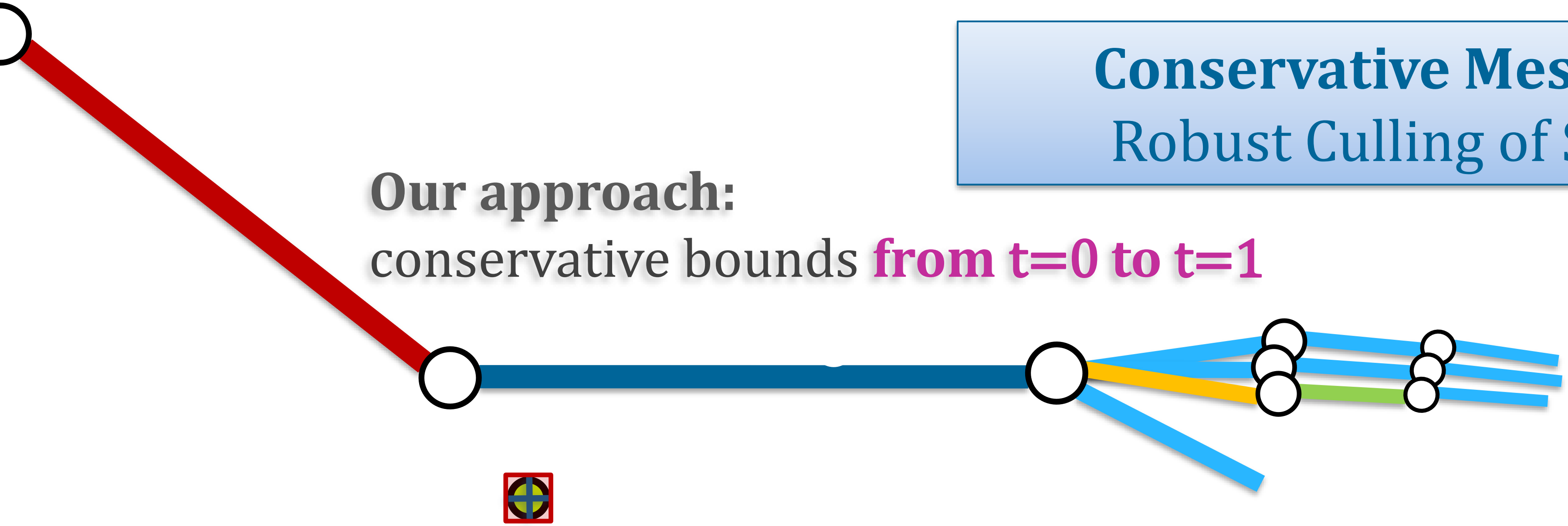
We want to know:
How much do **any** animated bounds deviate from an initial position?

Because if we know
across all animation states
=> very efficient culling in task shader
no computation/memory fetch for different animation states per meshlet with **constant scaling factor**

Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

Our approach:
conservative bounds **from t=0 to t=1**



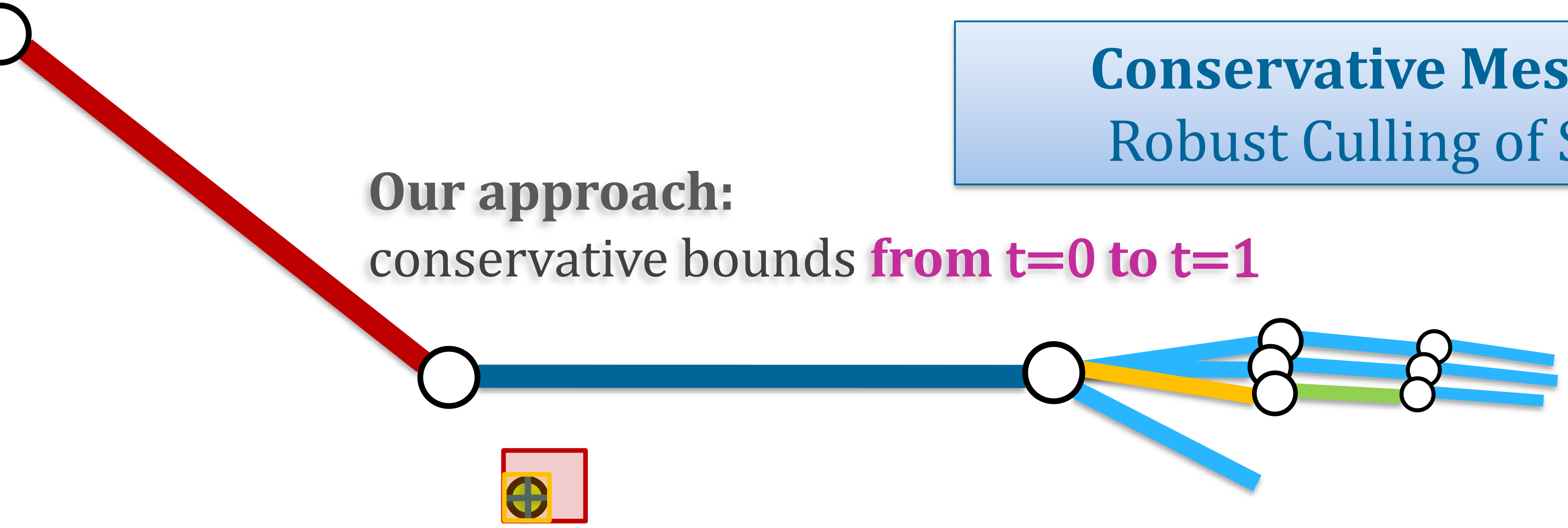
Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Our approach:
conservative bounds **from t=0 to t=1**

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Our approach:

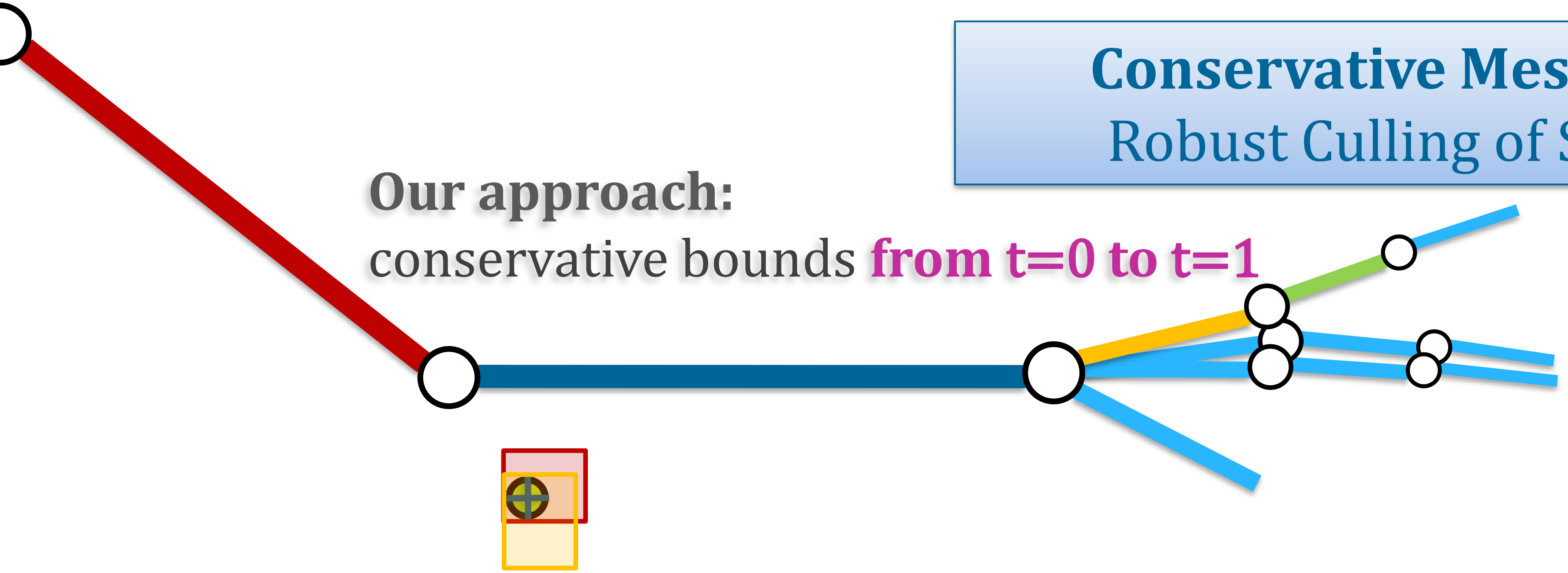
conservative bounds **from t=0 to t=1**



Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

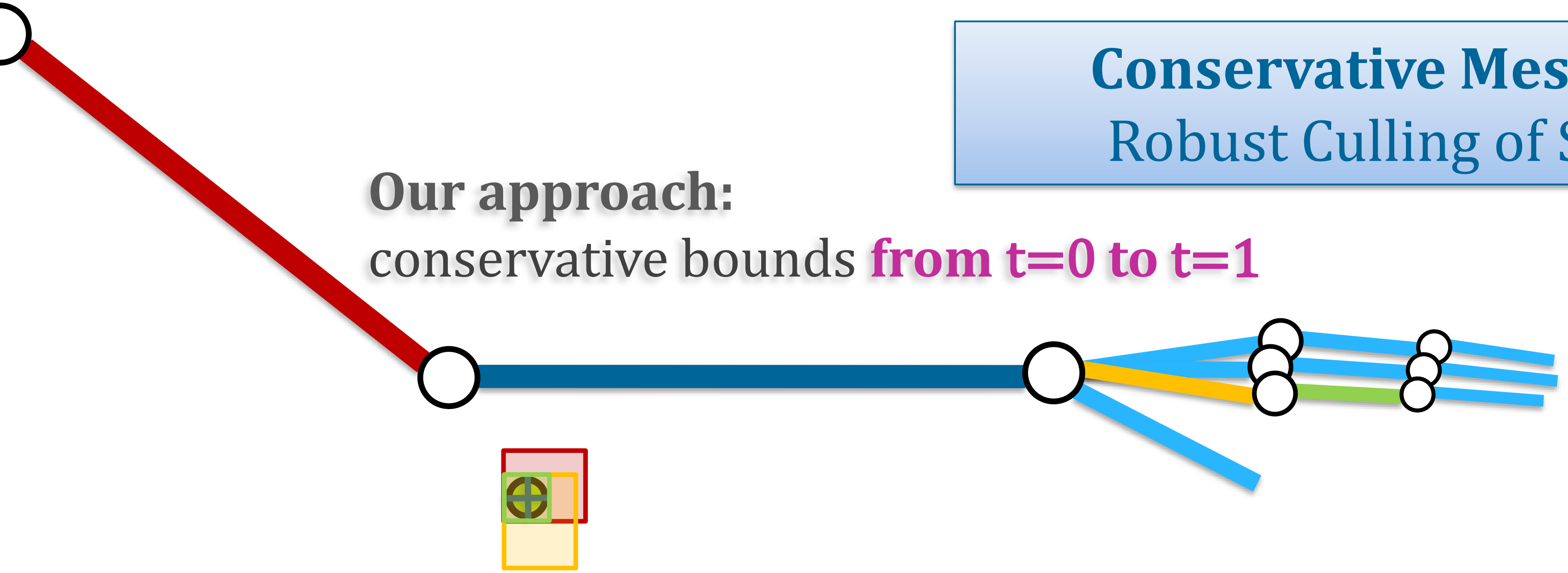
Our approach:
conservative bounds **from t=0 to t=1**



Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

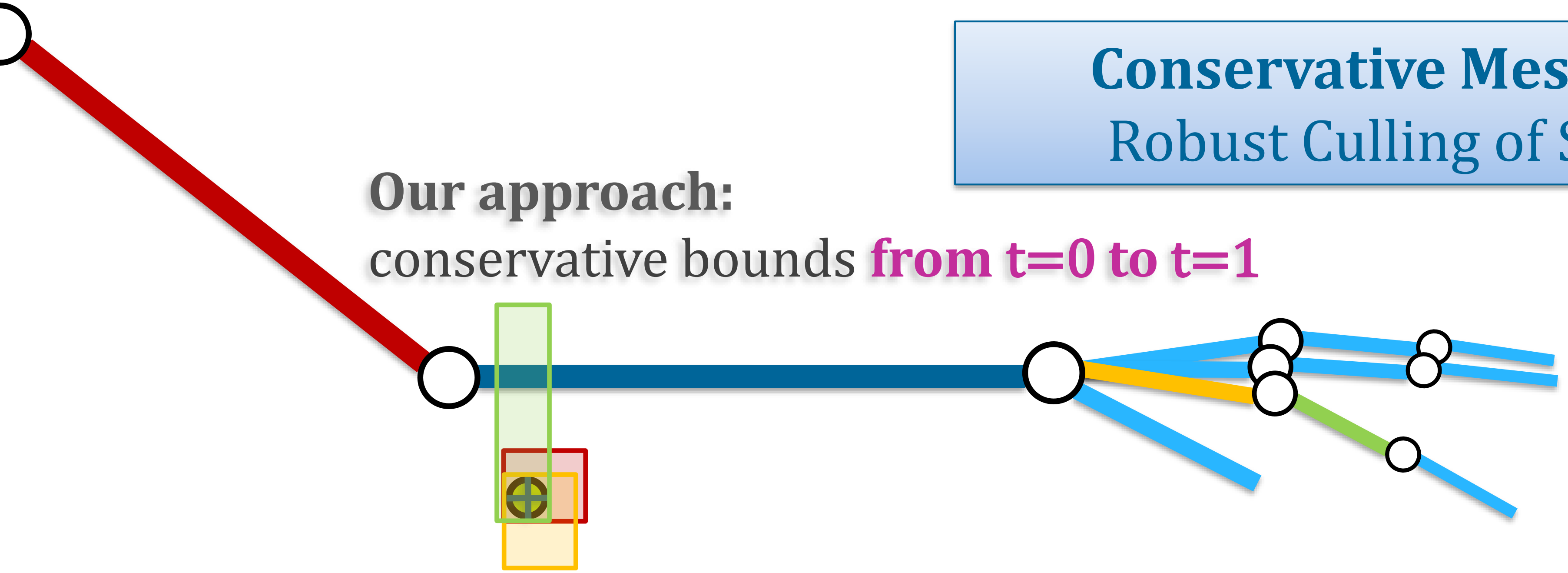
Our approach:
conservative bounds **from t=0 to t=1**



Meshlet Bounds

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

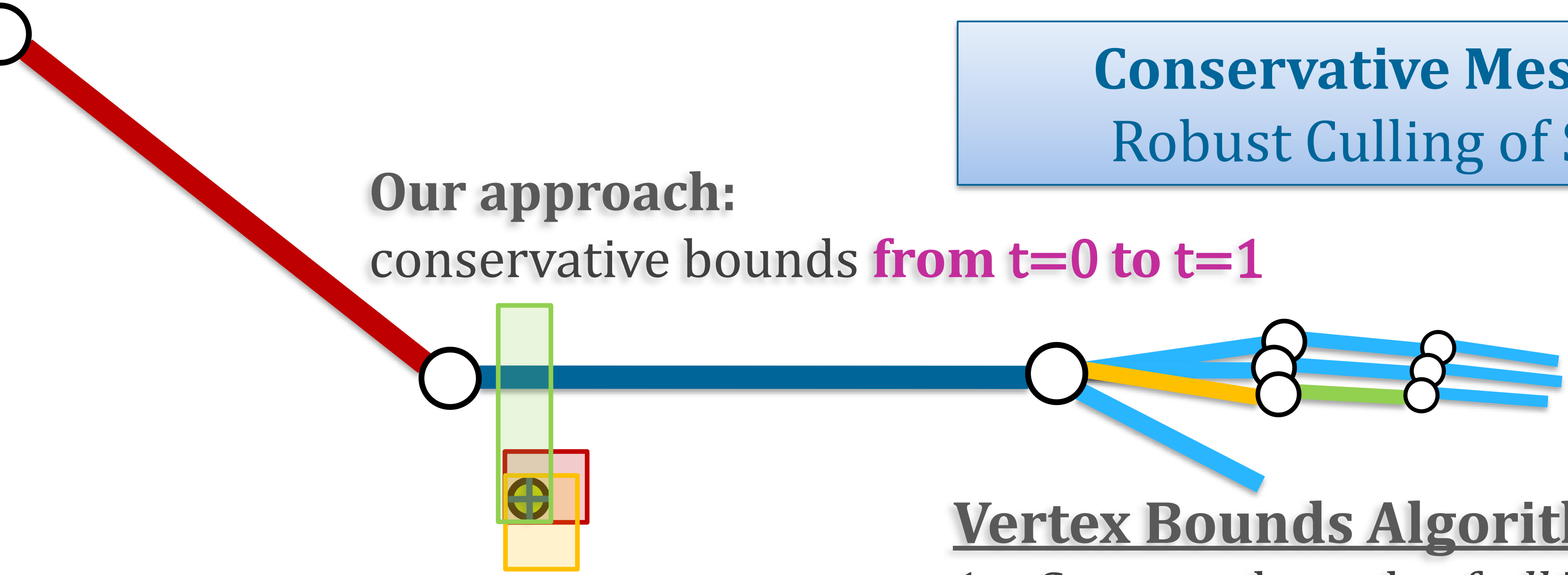
Our approach:
conservative bounds **from t=0 to t=1**



Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Our approach:

conservative bounds **from t=0 to t=1**



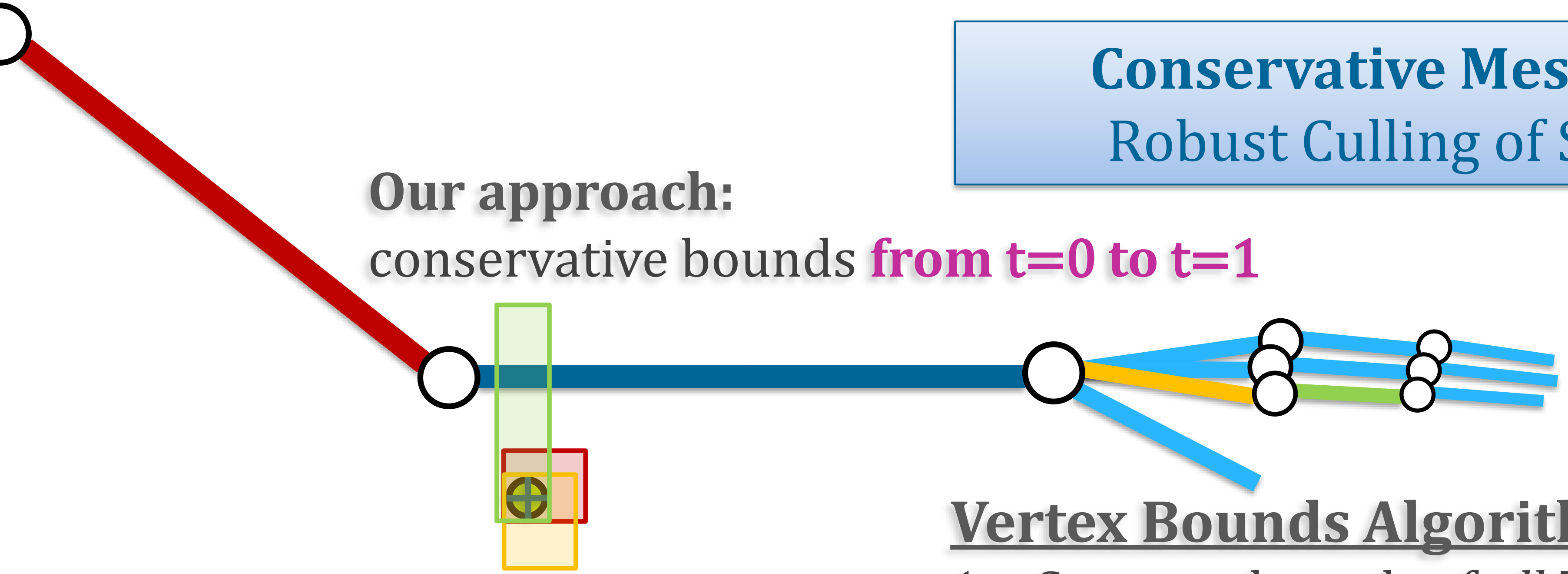
Vertex Bounds Algorithm:

1. Compute bounds of *all* **individual bones** of influence (weight != 0) as if they had weight 1
   a) Initial bounds
   b) Step-wise towards target bone
   c) Conservatively extend
2. Combine weighted into **vertex bounds** according to the skinning method used (LBS)

Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

Our approach:
conservative bounds **from t=0 to t=1**



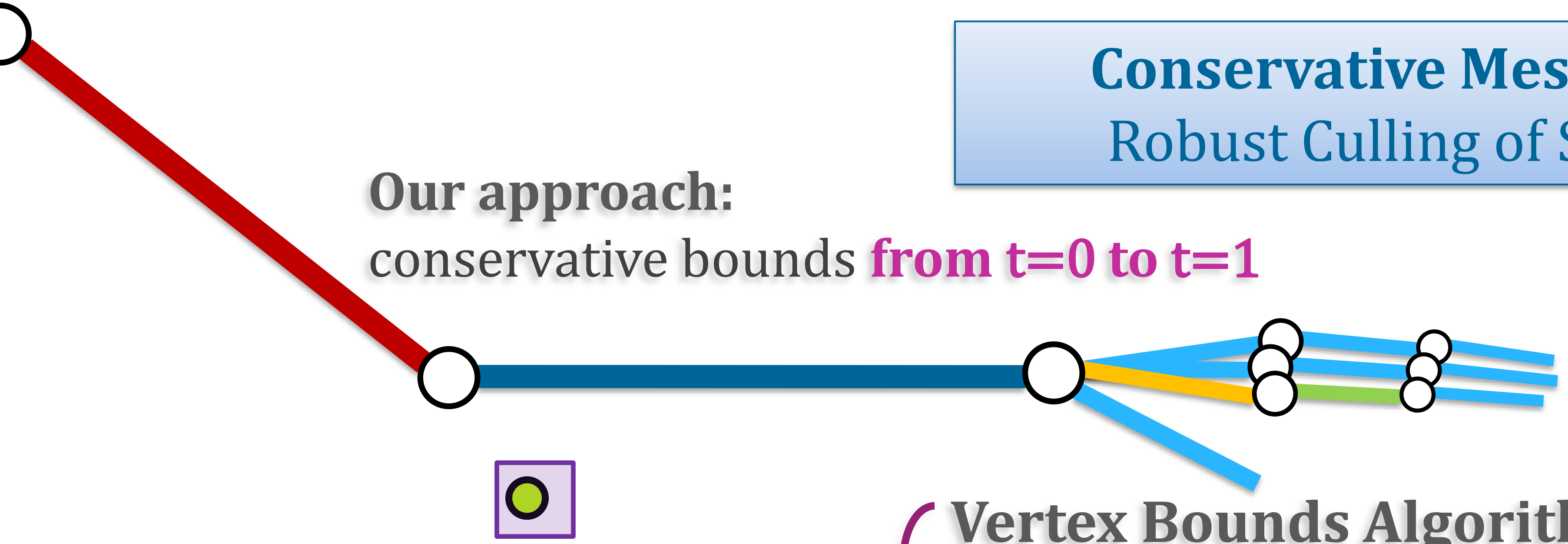
Contribution:
Conservative extension by maximum rotations in x, y, z through a derivative of Rodrigues' Rotation Formula

Vertex Bounds Algorithm:

1. Compute bounds of *all* **individual bones** of influence (weight != 0) as if they had weight 1
   a) Initial bounds
   b) Step-wise towards target bone
   c) Conservatively extend
2. Combine weighted into **vertex bounds** according to the skinning method used (LBS)

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

Our approach:
conservative bounds **from t=0 to t=1**



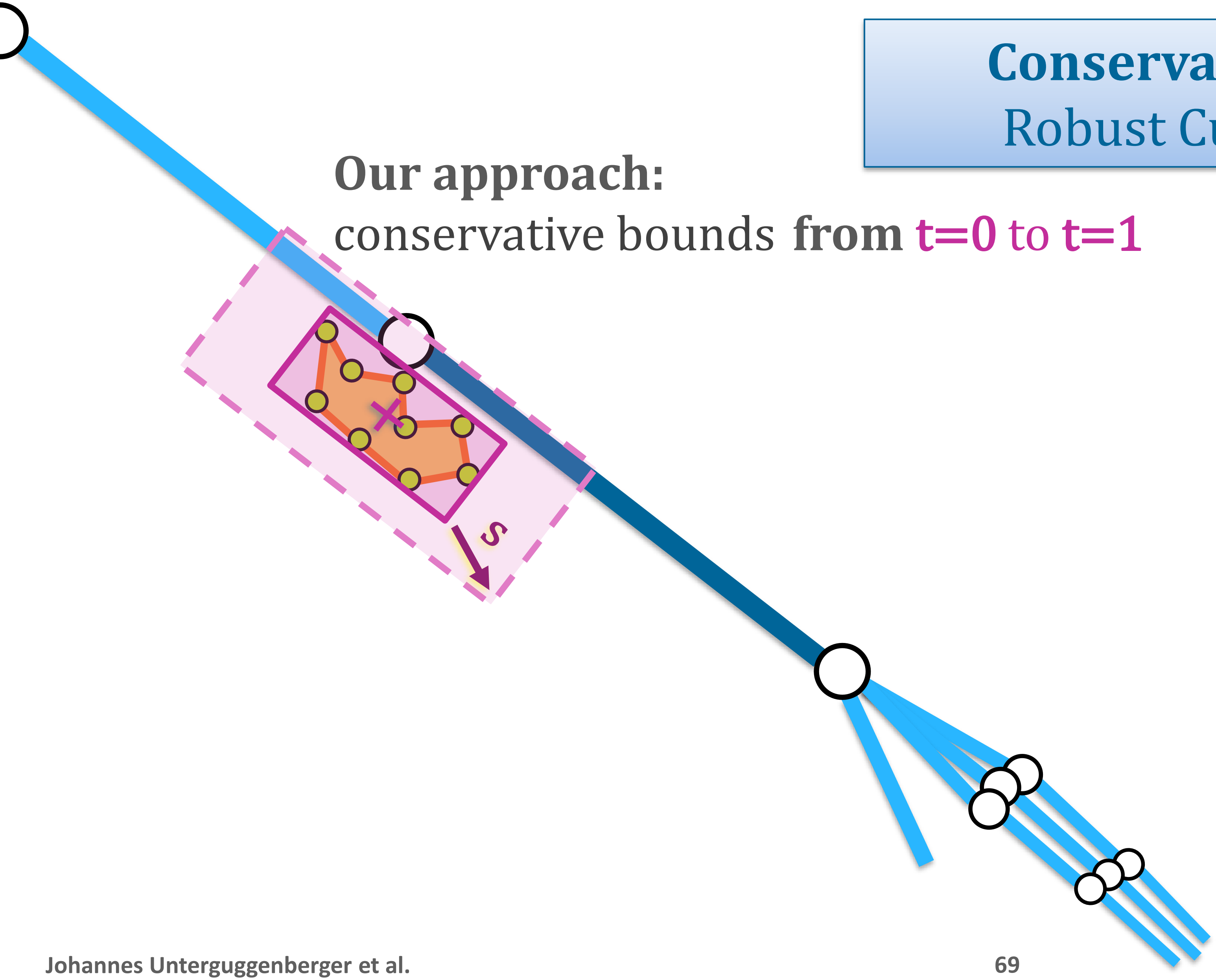
as precomputation step

Vertex Bounds Algorithm:

1. Compute bounds of *all* **individual bones** of influence (weight != 0) as if they had weight 1
   a) Initial bounds
   b) Step-wise towards target bone
   c) Conservatively extend
2. Combine weighted into **vertex bounds** according to the skinning method used (LBS)

Conservative Meshlet Bounds for Robust Culling of Skinned Meshes

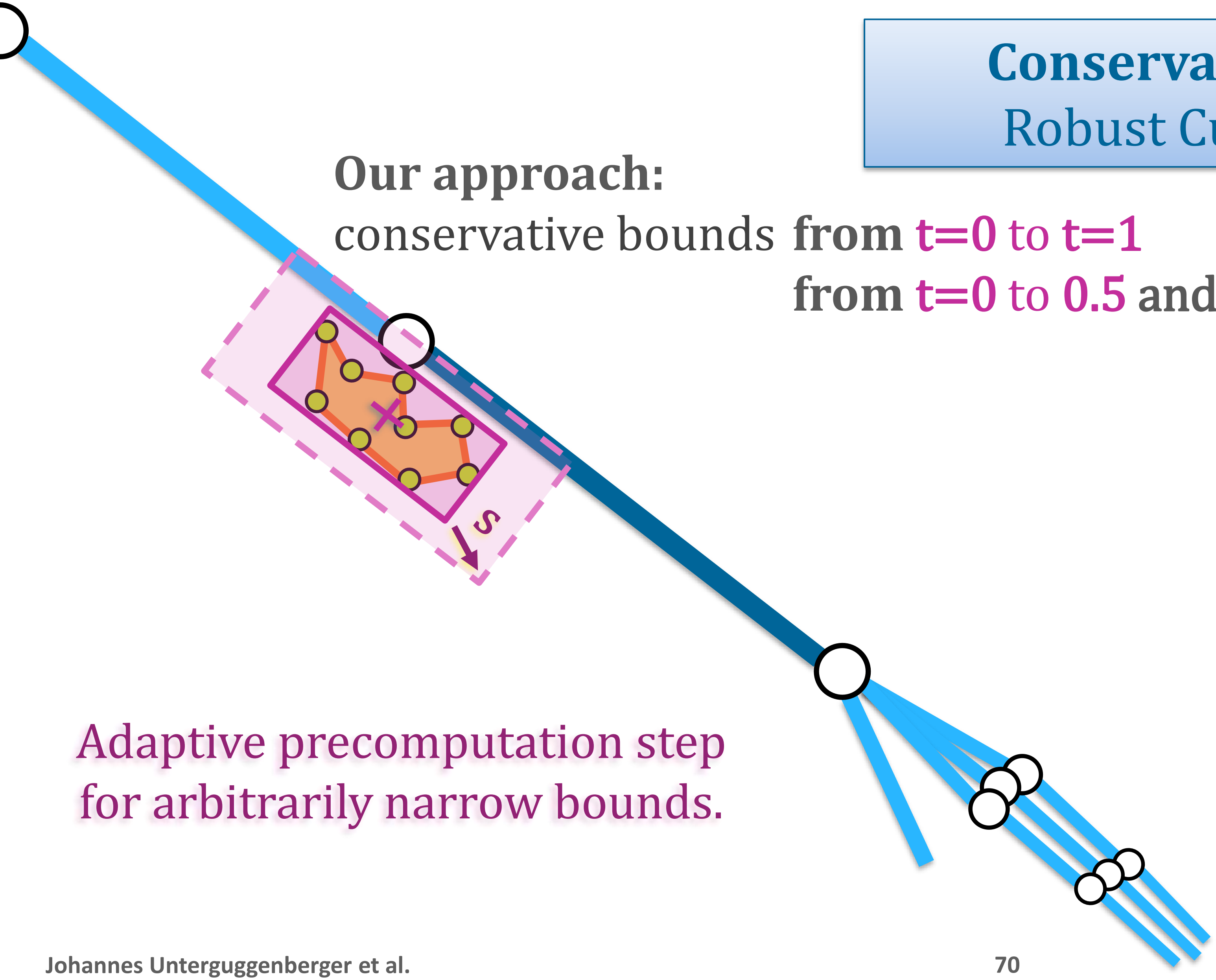
Our approach:
conservative bounds **from t=0** to **t=1**



Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes

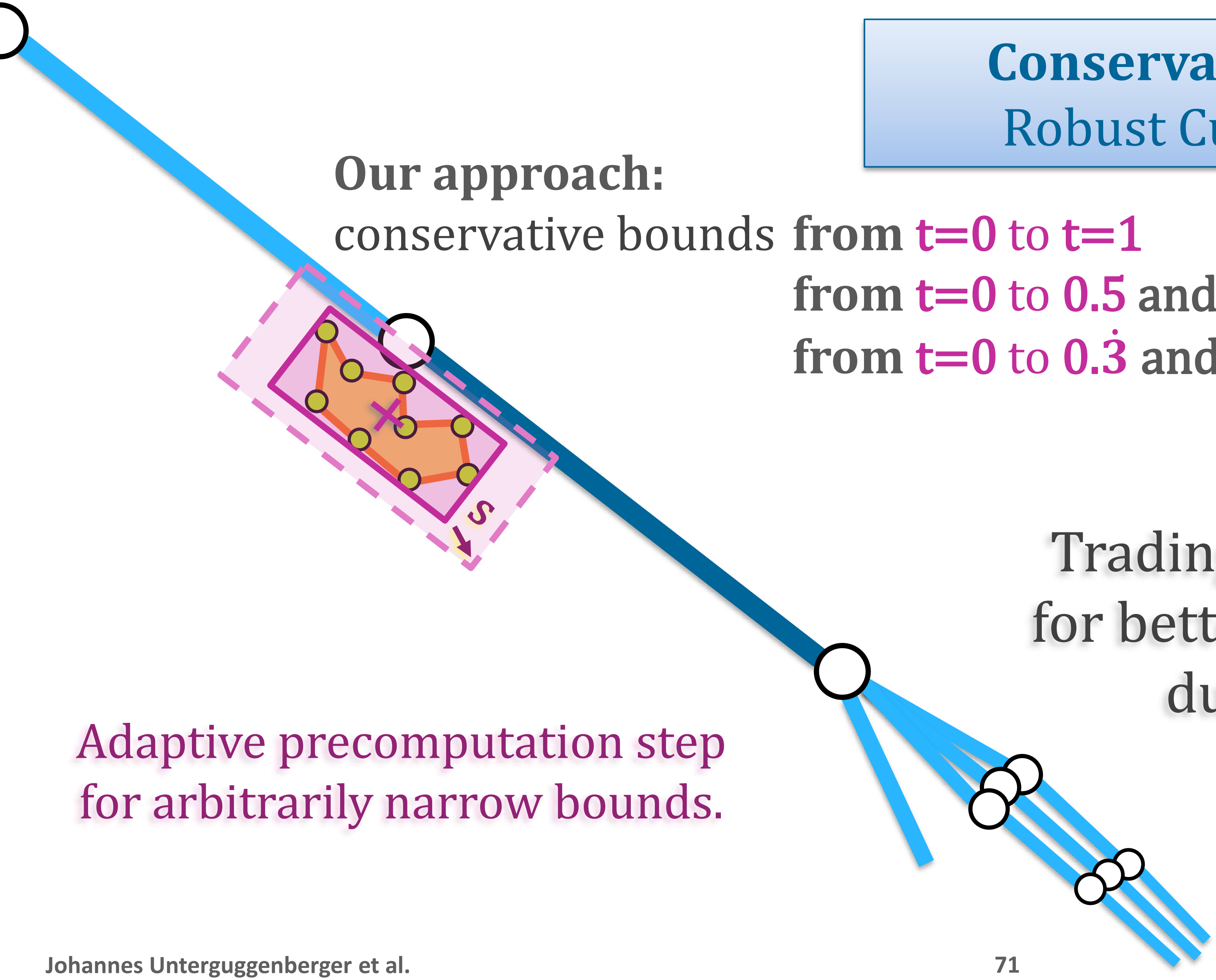
Our approach:
conservative bounds **from t=0 to t=1**
from t=0 to 0.5 and t=0.5 to 1



Adaptive precomputation step
for arbitrarily narrow bounds.

Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes



Our approach:

conservative bounds **from** t=0 to t=1

from t=0 to 0.5 **and** t=0.5 to 1

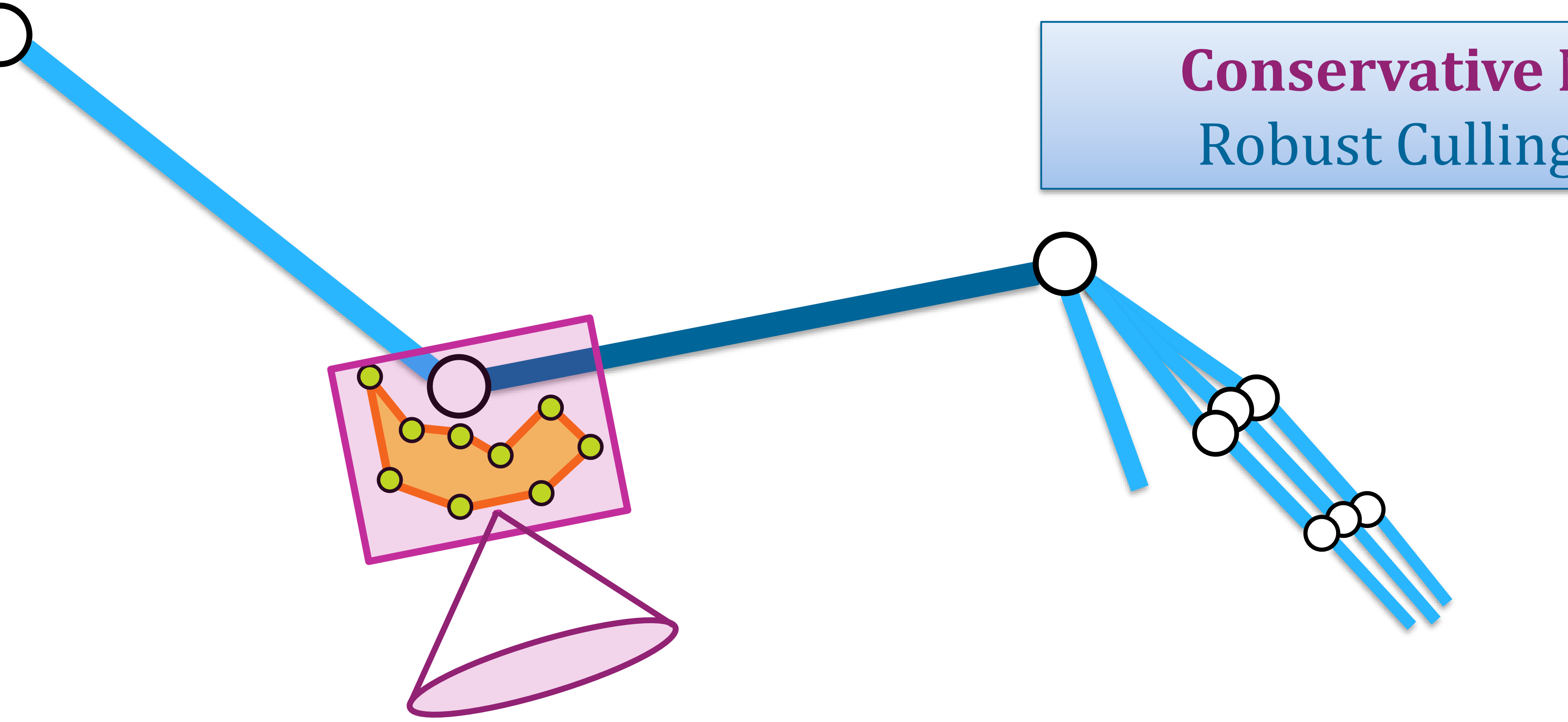
from t=0 to $0.\dot{3}$ **and** t=$0.\dot{3}$ to $0.\dot{6}$ **and and** t=$0.\dot{6}$ to 1

Trading precomputation effort
for better runtime performance,
due to better bounds.

Adaptive precomputation step
for arbitrarily narrow bounds.

Meshlet Bounds

Conservative Meshlet Bounds for
Robust Culling of Skinned Meshes



- High memory consumption
- Increased memory bandwidth
- More cache misses

Calculate bounds for all animation intervals of interest!

from t=0 to t=1
from t=1 to t=2
from t=2 to t=3

~~Store one bounding box per interval?~~