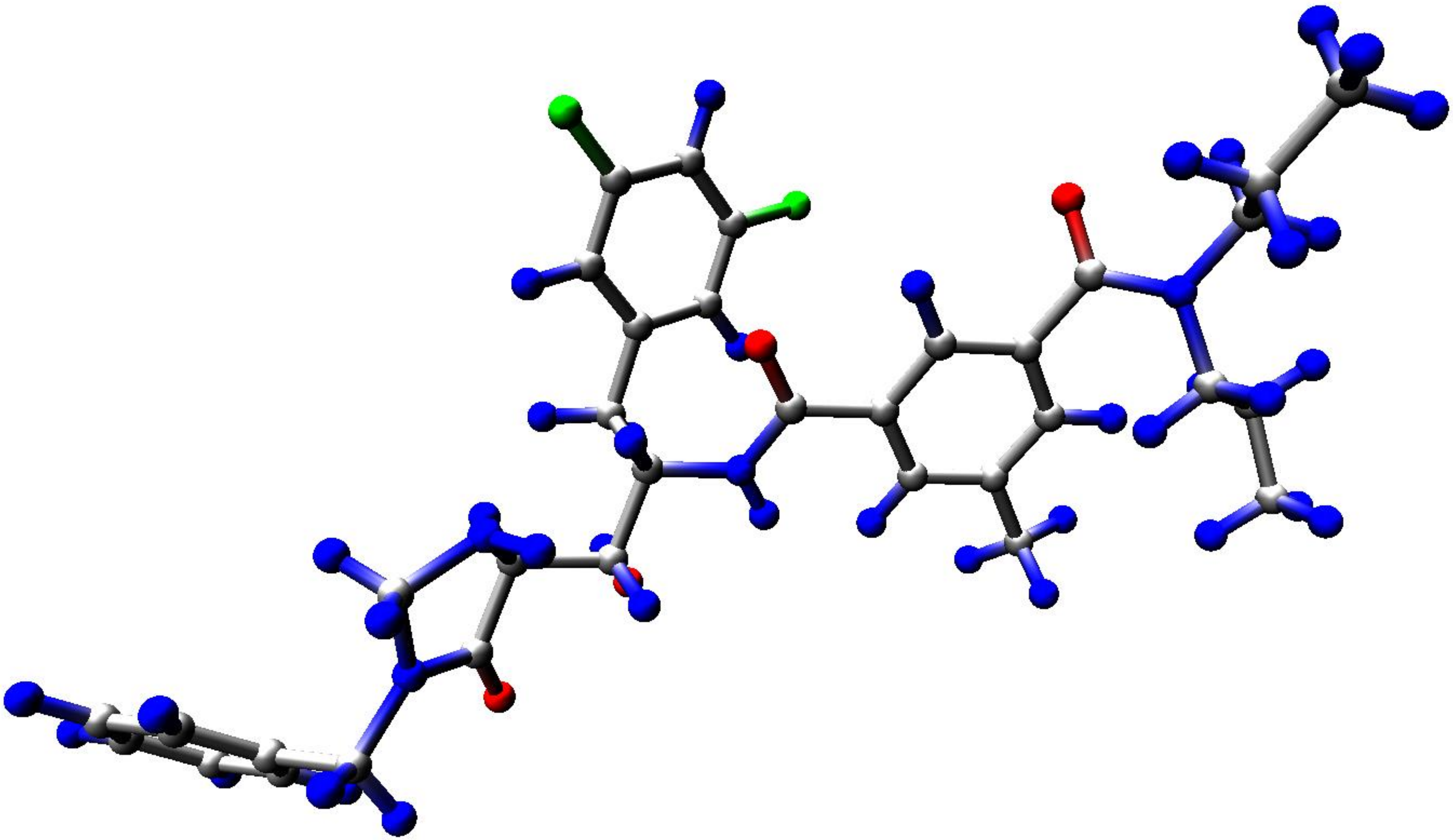
Exercise 2

Particle

Exercise 2 – Particle



Exercise 2 – Particle

1. Sphere Raycasting

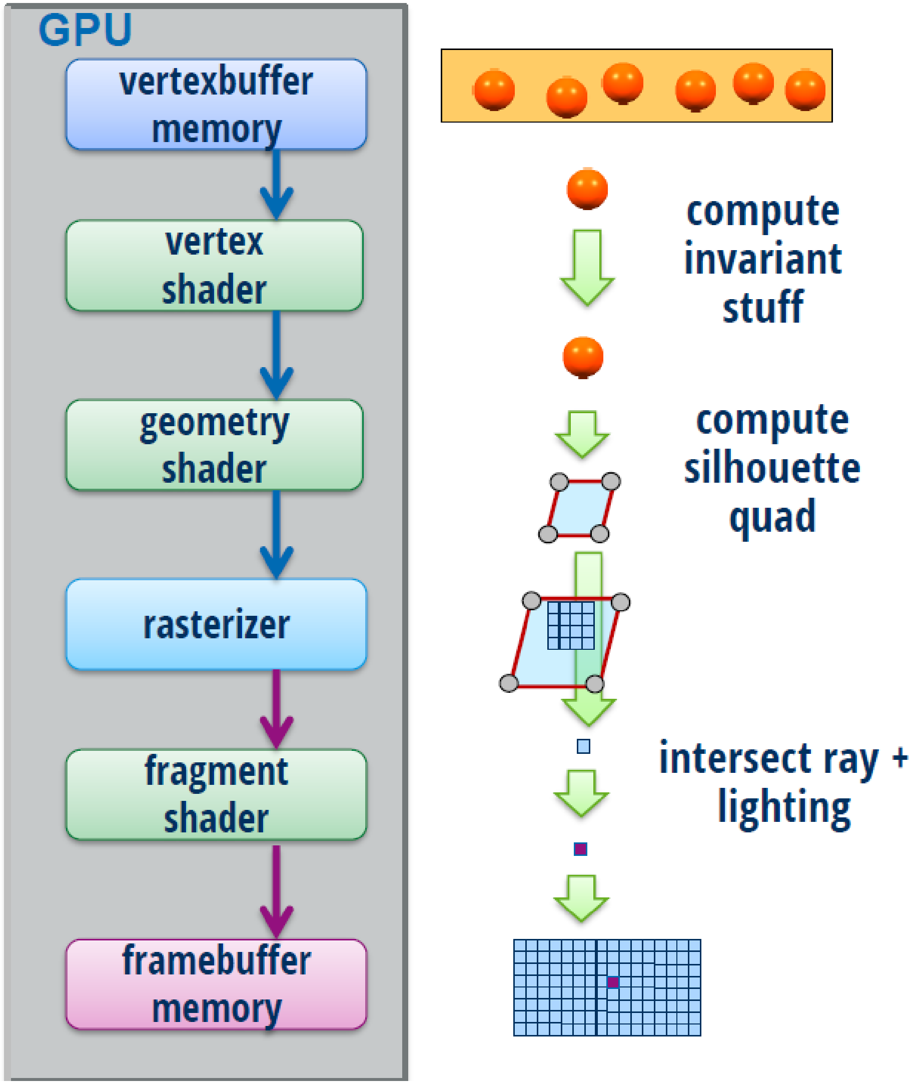
Tasks:

a) Render silhouette quad

b) Discard fragment

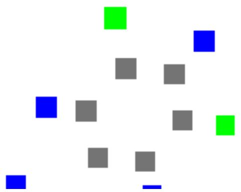
c) Ray intersection with sphere

d) Depth test

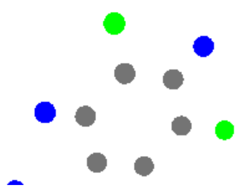


Intermediate Results

a) Render silhouette quad

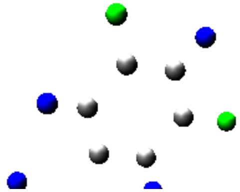


b) Discard fragment not on sphere



c) Ray intersection with sphere

d) Depth test



1. Sphere Raycasting

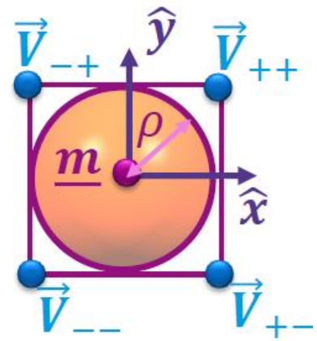
Computergraphik
und Visualisierung

a) Vertex Shader *sphere_raycast.glvs*

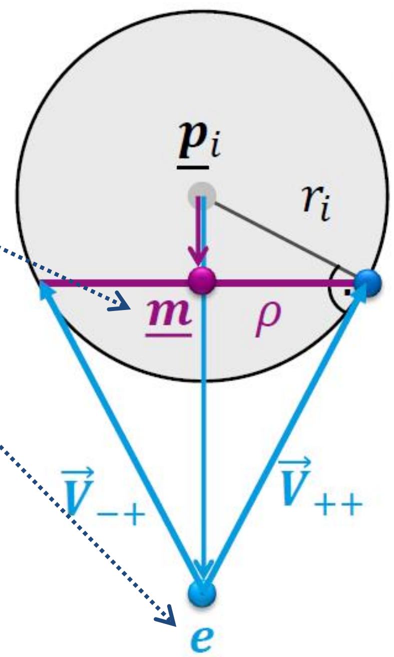
- Compute silhouette parameter *sps* in **parameter space**:
  - orthogonal directions of silhouette quad with length ρ:
    - y_tilde = ρ · $\hat{y}$
    - x_tilde = ρ · $\hat{x}$

$$\vec{V}_{\pm\pm} = \underline{\boldsymbol{m}} \pm \rho\hat{x} \pm \rho\hat{y}$$

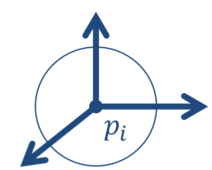
$$\rho^2 = 1 - \frac{r_i^2}{e^2}$$



  - silhouette center m_tilde $\tilde{\boldsymbol{m}} = \frac{r_i^2}{e^2}\tilde{e}$
  - eye point e_tilde $\tilde{\boldsymbol{e}} = \frac{1}{r_i}(\underline{\boldsymbol{e}} - \underline{\boldsymbol{p}_i})$



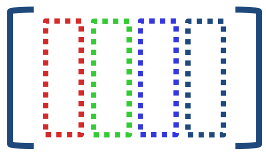
Parameter space:
Sphere center at origin

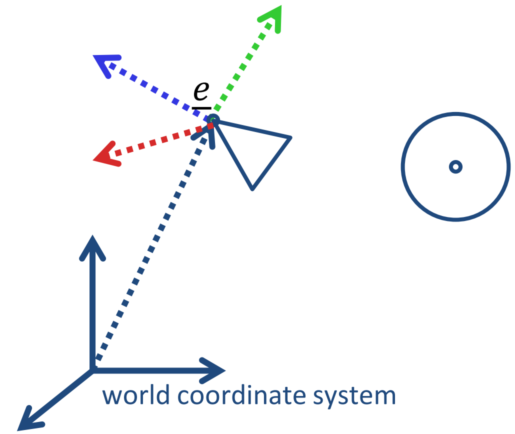


$\underline{e}$ ... eye point in world coordinates
$\underline{p_i}$ ... sphere center in world coordinates
r_i ... sphere radius
e ... length of vector ($\underline{e}$- $\underline{p_i}$)

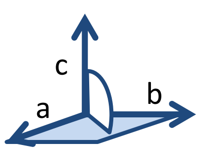
1. Sphere Raycasting

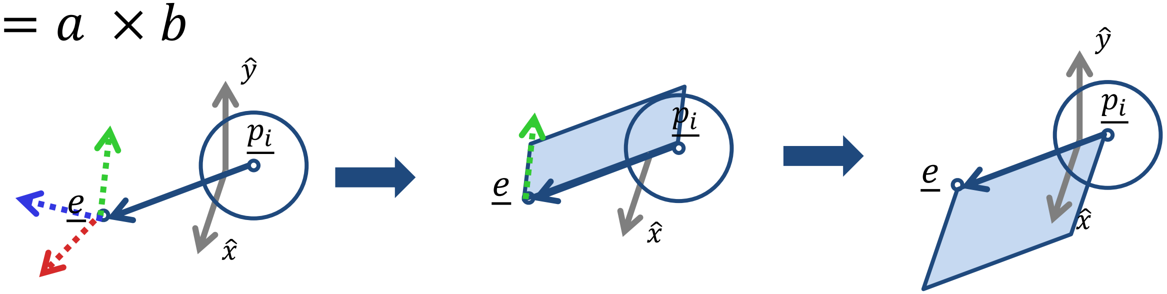
Hints for a)

- Inverse of the Model View Matrix
  - $iMV =$ 
  - Each column describes a vector in world coordinates



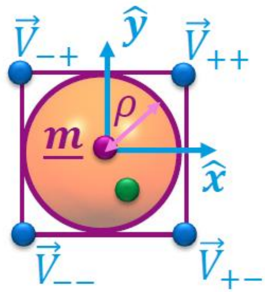
- Relation to orthogonal directions of the silhouette quad:
  - Crossproduct: $c = a \times b$





Hints for b)

- Silhouette Quad – easy sphere intersection check
  - Attach texture coordinates q_tilde $\in [-1,1]^2$ to the quad corners (struct *sphere_quad_info* in *sphere_raycast.glgs*)



$\underline{q} \in [-1,1]^2$

  - This simplifies the ray intersection to:

$$\|q\_tilde\| \leq 1$$

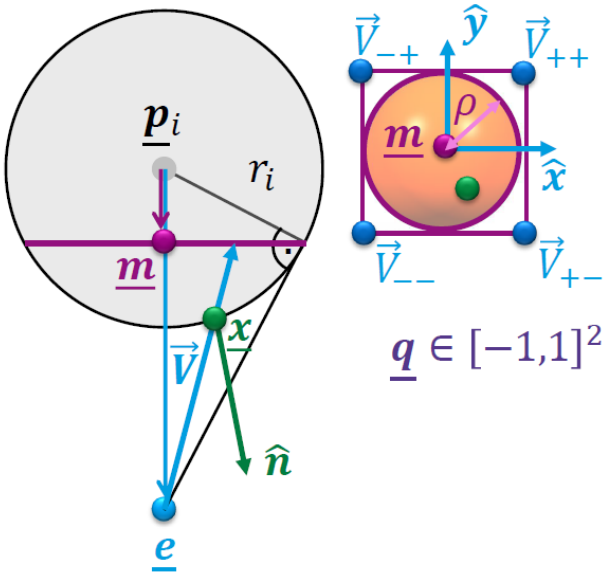
c) Rayintersection with Sphere

- special form where λ_+ is first intersection along ray:

$$\underline{x}_\pm = \underline{e} + \lambda_\pm \vec{V}, \lambda_\pm = \frac{1}{1 \pm \beta}$$

$$\beta = r_i \sqrt{1 - \|\underline{q}\|^2} / \|\underline{e} - \underline{p}_i\|$$

Hint for calculation in eye coordinates:
$\underline{e}$ ... is origin
$\underline{x}$ ... intersection point in eye coordinates
$\vec{V}$ ... ray in eye coordinates



Normal:

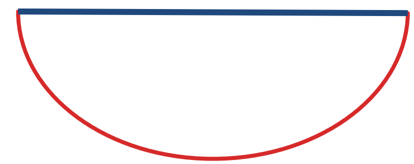
$$n = normalize((NM * \tilde{e}) + \lambda * (NM * (\tilde{v} - \tilde{e})))$$

NM ... Normal Matrix
$\tilde{v}$ ... point on silhouette in parameter space

invariant parts:
- could be already computed in the vertex shader (blue dashed box)
- could be already computed in the geometry shader (pink dashed box)

1. Sphere Raycasting

Hints for d) Depth Correction



- In order to correct the depth we only need the deph-value (z) and w
- Therefore compute two 2D-vectors containing the z and w component of the clip coordinates of the eye and the current point on the silhouette
- Compute the intersection point with the previous vectors as eye position and for ray calculation:

$$\underline{x}_{\pm} = \underline{e} + \lambda_{\pm} \vec{V},$$

- Perform a w-clip and remapped to range of window coordinates

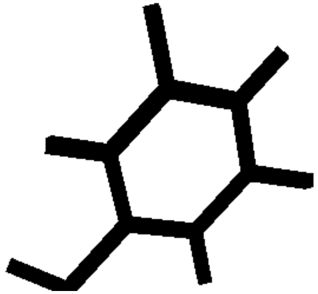
```
// vertex/geometry shader
out vec2 zw_cl;
:
    zw_cl = (MVP*position).zw;
:
```

```
// fragment shader
in vec2 zw_clip;
:
    float z_NDC = zw_cl.z/zw_cl.w;
    gl_FragDepth = 0.5*(z_NDC+1.0);
:
```

Exercise 2 – Particle

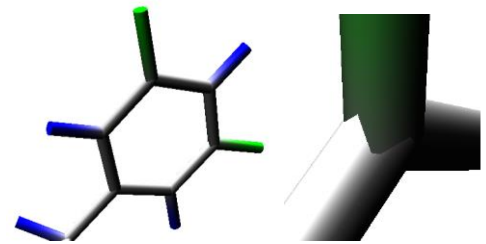
2. Cylinder Raycasting

- Render silhouette as bounding box

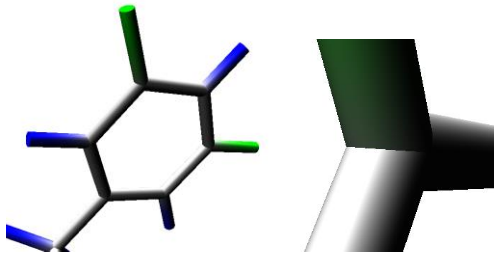


- Ray intersection with cylinder
  - Test for planar sides of cylinder

  - Intersection with cylinder mantle

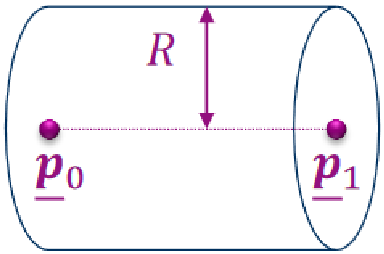


- Depth correction



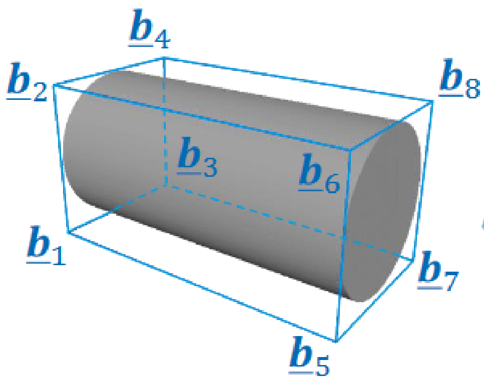
Cylinder Raycasting

Definition: start and end points $\underline{\boldsymbol{p}}_0$ and $\underline{\boldsymbol{p}}_1$ plus radius R.



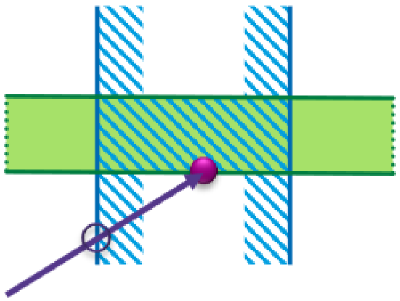
Silhouette Cover

- tessellate object oriented bounding box (OOBB)



Ray Intersection

- represent cylinder as intersection of two planar half-spaces and cylinder barrel
- Intersection is first ray point that is inside of all three parts



Cylinder Raycasting – OOBB

OOBB Tessellation

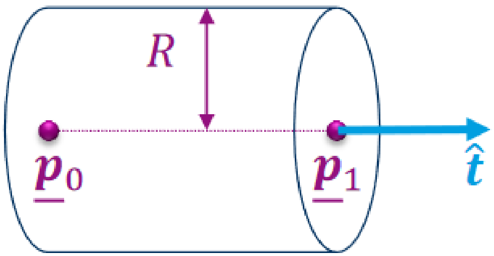
- compute tangent vector $\hat{\boldsymbol{t}}$ from $\underline{\boldsymbol{p}}_{0|1}$
- extent to orthonormal object coordinate system $\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{t}}$:

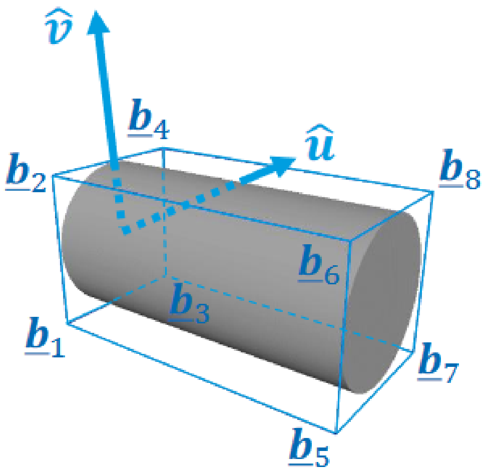
$$\hat{\boldsymbol{u}} = \text{normalize}\left(\hat{\boldsymbol{t}} \times \begin{cases} \hat{\boldsymbol{z}} & t_x^2 + t_y^2 > \epsilon \\ \hat{\boldsymbol{y}} & \text{sonst} \end{cases}\right)$$

$$\hat{\boldsymbol{v}} = \hat{\boldsymbol{t}} \times \hat{\boldsymbol{u}}$$

- box corners: $\underline{\boldsymbol{b}}_{1..8} = \underline{\boldsymbol{p}}_{0|1} \pm R\hat{\boldsymbol{u}} \pm R\hat{\boldsymbol{v}}$
- For more efficient transformation encode information in a single 4x4-matrix:

$$\widetilde{\boldsymbol{B}}^{\text{world}} = \begin{pmatrix} \underline{\boldsymbol{p}}_0 & \underline{\boldsymbol{p}}_1 & R\hat{\boldsymbol{u}} & R\hat{\boldsymbol{v}} \\ 1 & 1 & 0 & 0 \end{pmatrix}$$



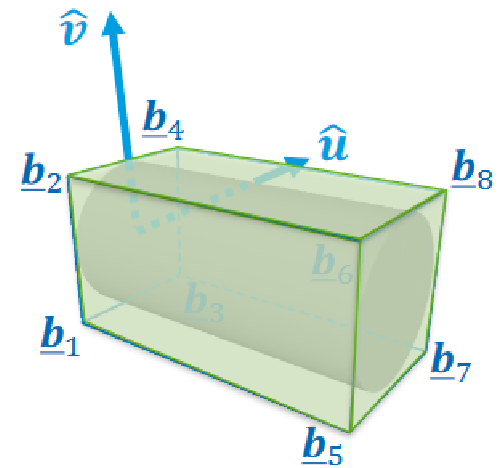


Cylinder Raycasting

Computer Graphics and Visualization

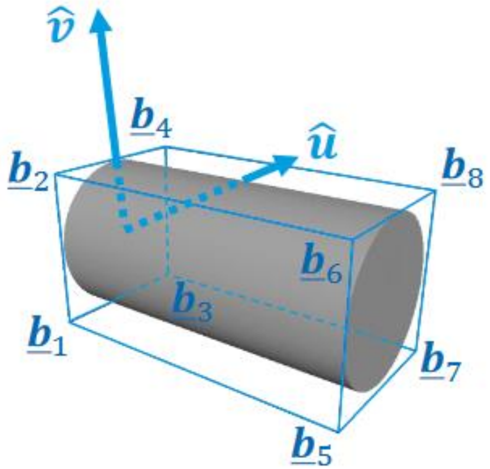
OOBB Tessellation

- already in vertex shader transform to clip coordinates
$$\widetilde{\boldsymbol{B}}^{\text{clip}} = \boldsymbol{MVP} \cdot \widetilde{\boldsymbol{B}}^{\text{world}}$$
- Pass $\widetilde{\boldsymbol{B}}^{\text{clip}}$ to geometry shader, recover clip space corners and emit length 2 triangle strip per OOBB face
$$\widetilde{\boldsymbol{b}}_{1..8}^{\text{clip}} = \widetilde{\boldsymbol{B}}_{0|1}^{\text{clip}} \pm \widetilde{\boldsymbol{B}}_{2}^{\text{clip}} \pm \widetilde{\boldsymbol{B}}_{3}^{\text{clip}}$$
- Careful: this pre-transformation only works with points/vectors that have either 1 or 0 in the w-component.
- Optionally perform culling of backfacing facettes



Exercise 2 – 2. Cylinder Raycasting

Hints for b) Ray Intersection



- Check if fragment on planar sides of cylinder:
  - $\|pos_{ucyl}.uv\|^2 < 1$ (fragment inside cylinder)
  - $pos_{ucyl}.t < \epsilon$ or $> \epsilon$ (fragment on planar parts)
- Compute ray intersection in unit cylinder coordinates

$$x = e + \lambda v$$
$$r^2 = 1 = x^2 = (e + \lambda v)^2 = e^2 + 2\lambda(e \cdot v) + \lambda^2 v^2$$
$$\Rightarrow 0 = e^2 - 1 + 2(e \cdot v)\lambda + v^2\lambda^2$$

$$\Rightarrow \lambda_\pm = \left(-(e \cdot v) \pm \sqrt{\underbrace{(e \cdot v)^2 - v^2(e^2 - 1)}_{> 0}}\right) / v^2$$