Methods description

Task – separate the free and flat models

step 01 - Bounding Box Ratio = Measuring Thickness

What is bounding box?

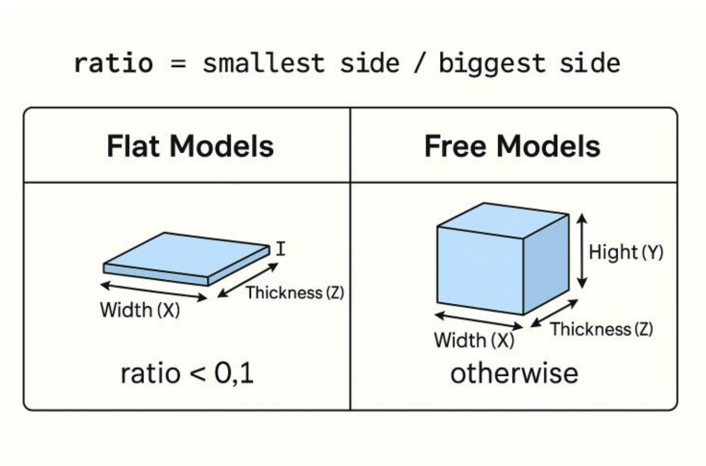
- A bounding box is the smallest 3D box that completely encloses a 3D object.
- Just like wrapping the 3D model tightly in a rectangular box.

How does it work?

1. For each 3D model we should calculate its **bounding box**.
2. Means we measure **length, width**, and **height(thickness)** of the box.
3. Calculate the **ratio** of the **smallest dimension to the largest dimension.**

ratio=largest side / smallest side

4. If **ratio = small,** then it is **thin and likely flat**.
5. If **ratio = large**, then it is **more volumetric → free** model.



Example:

Flat model: A 3D printed coaster might have dimensions

100×100×2 mm → ratio = 2/100 = 0.02 → very flat.

Free model: A figurine of a cat might have dimensions

50×30×80 mm → ratio = 30/80 = 0.375 → not flat.

```
if not isinstance(mesh, trimesh.Trimesh):
    continue  # skip multi-part meshes

# --- Step 1: Bounding box ratio ---
extents = mesh.bounding_box.extents
min_dim = min(extents)
max_dim = max(extents)
ratio = min_dim / max_dim
```

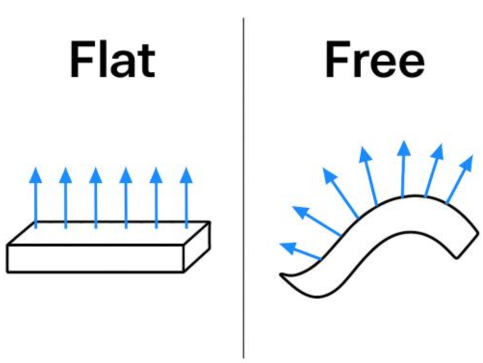
extends → gives the **length, width, and height (Thickness)** of the bounding box.

Example: If a model's box is 10×2×1 (length×width×height), extents = [10, 2, 1].

min_dim / max_dim → calculates the **ratio of the smallest dimension to the largest dimension**.

Step 02 - Surface Normal Alignment – Measuring Flatness of Surfaces

Every 3D model is made of **tiny flat triangles**.



What is a Normal?

- Every small triangle that makes up a 3D model has a **normal**.
- A **normal is like a tiny arrow sticking straight out** from the triangle.
- It shows **which direction that part of the surface is facing**

1. Average Normal

In this take the mean of all normals

$$\text{avg\_normal} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{n_i}$$

- n_i = normal vector of each triangle.
- Normalize it (make it length = 1) → gives a **reference direction**. [A reference direction is a single, consistent direction used as a baseline to compare other directions.]

2. Dot Product

Measure similarity between vecotrs.

$$\mathbf{A} \cdot \mathbf{B} = |A||B| \cos\theta$$

- θ = angle between vectors.
- If $\cos\theta \approx 1$ → vectors point almost the same direction.
- If $\cos\theta \approx 0$ → perpendicular.
- If $\cos\theta \approx -1$ → opposite.

3. Alignment Check

aligned_ratio = np.mean(alignment > normal_threshold)

alignment > normal_threshold → counts triangles whose normals are mostly aligned with the average normal.

np.mean(...) → calculates the fraction of aligned triangles.

Example:

- aligned_ratio = 0.95 → 95% of faces point in the same direction → flat surface.
- aligned_ratio = 0.3 → many faces point different ways → free surface.

How it works?

1. Each triangle in the mesh has a **normal vector** perpendicular to its surface.
2. Calculate the **average normal** for the whole model.

- Values close to **1** → triangle is aligned with the average → flat region.
- Values close to **0** → triangle is perpendicular → curved surface.

4. Calculate the **aligned ratio**: the fraction of triangles aligned with the average normal.
5. If **most triangles are aligned** (>80%), the model is considered **flat**.

Step 3: Combined

- **Bounding box ratio** → finds thin models.
- **Surface normal alignment** → finds flat surfaces.
- **Both together** → robust classification: thin and flat models vs free-form models.