Skip to content

BVHTree Utilities (mathutils.bvhtree)

BVH tree structures for proximity searches and ray casts on geometry.

class mathutils.bvhtree.**BVHTree**

classmethod FromBMesh(bmesh, epsilon=0.0)

BVH tree based on `BMesh` data.

PARAMETERS:

- **bmesh** (`BMesh`) – BMesh data.
- **epsilon** (*float*) – Increase the threshold for detecting overlap and raycast hits.

classmethod FromObject(object, depsgraph, deform=True, render=False, cage=False, epsilon=0.0)

BVH tree based on `Object` data.

PARAMETERS:

- **object** (`Object`) – Object data.
- **depsgraph** (`Depsgraph`) – Depsgraph to use for evaluating the mesh.
- **deform** (*bool*) – Use mesh with deformations.
- **cage** (*bool*) – Use modifiers cage.
- **epsilon** (*float*) – Increase the threshold for detecting overlap and raycast hits.

classmethod FromPolygons(vertices, polygons, all_triangles=False, epsilon=0.0)

BVH tree constructed geometry passed in as arguments.

PARAMETERS:

- **vertices** (*Sequence[Sequence[float]]*) – float triplets each representing `(x, y, z)`
- **polygons** (*Sequence[Sequence[int]]*) – Sequence of polygons, each containing indices to the vertices argument.
- **all_triangles** (*bool*) – Use when all **polygons** are triangles for more efficient conversion.
- **epsilon** (*float*) – Increase the threshold for detecting overlap and raycast hits.

find_nearest(origin, distance=1.84467e+19)

Find the nearest element (typically face index) to a point.

PARAMETERS:

- **co** (`Vector`) – Find nearest element to this point.
- **distance** (*float*) – Maximum distance threshold.

RETURNS:

Returns a tuple: (position, normal, index, distance), Values will all be None if no hit is found.

RETURN TYPE:

tuple[`Vector` | None, `Vector` | None, int | None, float | None]

find_nearest_range(origin, distance=1.84467e+19)

Find the nearest elements (typically face index) to a point in the distance range.

PARAMETERS:

- **co** (`Vector`) – Find nearest elements to this point.
- **distance** (*float*) – Maximum distance threshold.

RETURNS:

Returns a list of tuples (position, normal, index, distance)

RETURN TYPE:

list[tuple[`Vector` , `Vector` , int, float]]

overlap(other_tree)

Find overlapping indices between 2 trees.

PARAMETERS:

other_tree (`BVHTree`) – Other tree to perform overlap test on.

RETURNS:

Returns a list of unique index pairs, the first index referencing this tree, the second referencing the **other_tree**.

RETURN TYPE:

list[tuple[int, int]]

ray_cast(origin, direction, distance=sys.float_info.max)

Cast a ray onto the mesh.

PARAMETERS:

- **origin** (`Vector`) – Start location of the ray in object space.
- **direction** (`Vector`) – Direction of the ray in object space.
- **distance** (*float*) – Maximum distance threshold.

RETURNS:

Returns a tuple: (position, normal, index, distance), Values will all be None if no hit is found.

RETURN TYPE:

tuple[`Vector` | None, `Vector` | None, int | None, float | None]