Skip to content

KDTree Utilities (mathutils.kdtree)

Generic 3-dimensional kd-tree to perform spatial searches.

```
import mathutils

# create a kd-tree from a mesh
from bpy import context
obj = context.object

mesh = obj.data
size = len(mesh.vertices)
kd = mathutils.kdtree.KDTree(size)

for i, v in enumerate(mesh.vertices):
    kd.insert(v.co, i)

kd.balance()


# Find the closest point to the center
co_find = (0.0, 0.0, 0.0)
co, index, dist = kd.find(co_find)
print("Close to center:", co, index, dist)

# 3d cursor relative to the object data
co_find = obj.matrix_world.inverted() @ context.scene.cursor.location

# Find the closest 10 points to the 3d cursor
print("Close 10 points")
for (co, index, dist) in kd.find_n(co_find, 10):
    print("    ", co, index, dist)


# Find points within a radius of the 3d cursor
print("Close points within 0.5 distance")
for (co, index, dist) in kd.find_range(co_find, 0.5):
    print("    ", co, index, dist)
```

class mathutils.kdtree.**KDTree**

KdTree(size) -> new kd-tree initialized to hold `size` items.

> Note
>
> `KDTree.balance` must have been called before using any of the `find` methods.

balance()

Balance the tree.

> Note
>
> This builds the entire tree, avoid calling after each insertion.

find(co, filter=None)

Find nearest point to `co`.

PARAMETERS:

- **co** (*Sequence[float]*) – 3D coordinates.
- **filter** (*Callable[[int], bool]*) – function which takes an index and returns True for indices to include in the search.

RETURNS:

Returns (position, index, distance).

RETURN TYPE:

tuple[`Vector`, int, float]

find_n(co, n)

Find nearest `n` points to `co`.

PARAMETERS:

- **co** (*Sequence[float]*) – 3D coordinates.
- **n** (*int*) – Number of points to find.

RETURNS:

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

RETURN TYPE:

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

find_range(co, radius)

Find all points within `radius` of `co`.

PARAMETERS:

- **co** (*Sequence[float]*) – 3D coordinates.
- **radius** (*float*) – Distance to search for points.

RETURNS:

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

RETURN TYPE:

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

insert(co, index)

Insert a point into the KDTree.

PARAMETERS:

- **co** (*Sequence[float]*) – Point 3d position.
- **index** (*int*) – The index of the point.