

# ShaderNodeTexPointDensity(ShaderNode)

base classes — [bpy\\_struct](#), [Node](#), [NodeInternal](#), [ShaderNode](#)

**class** bpy.types.**ShaderNodeTexPointDensity**(**ShaderNode**)

Generate a volumetric point for each particle or vertex of another object

## interpolation

Texture interpolation

- `Closest` Closest – No interpolation (sample closest texel).
- `Linear` Linear – Linear interpolation.
- `Cubic` Cubic – Cubic interpolation.

## TYPE:

enum in ['Closest', 'Linear', 'Cubic'], default 'Linear'

## object

Object to take point data from

## TYPE:

[Object](#)

## particle\_color\_source

Data to derive color results from

- `PARTICLE_AGE` Particle Age – Lifetime mapped as 0.0 to 1.0 intensity.
- `PARTICLE_SPEED` Particle Speed – Particle speed (absolute magnitude of velocity) mapped as 0.0 to 1.0 intensity.
- `PARTICLE_VELOCITY` Particle Velocity – XYZ velocity mapped to RGB colors.

## TYPE:

enum in ['PARTICLE\_AGE', 'PARTICLE\_SPEED', 'PARTICLE\_VELOCITY'], default 'PARTICLE\_AGE'

## particle\_system

Particle System to render as points

## TYPE:

[ParticleSystem](#)

## point\_source

Point data to use as renderable point density

- `PARTICLE_SYSTEM` Particle System – Generate point density from a particle system.
- `OBJECT` Object Vertices – Generate point density from an object's vertices.

## TYPE:

enum in ['PARTICLE\_SYSTEM', 'OBJECT'], default 'PARTICLE\_SYSTEM'

## radius

Radius from the shaded sample to look for points within

## TYPE:

float in [0.001, inf], default 0.0

## resolution

Resolution used by the texture holding the point density

**TYPE:**

int in [1, 32768], default 0

**space**

Coordinate system to calculate voxels in

**TYPE:**

enum in ['OBJECT', 'WORLD'], default 'OBJECT'

**vertex\_attribute\_name**

Vertex attribute to use for color

**TYPE:**

string, default '', (never None)

**vertex\_color\_source**

Data to derive color results from

- VERTEX\_COLOR Vertex Color – Vertex color layer.
- VERTEX\_WEIGHT Vertex Weight – Vertex group weight.
- VERTEX\_NORMAL Vertex Normal – XYZ normal vector mapped to RGB colors.

**TYPE:**

enum in ['VERTEX\_COLOR', 'VERTEX\_WEIGHT', 'VERTEX\_NORMAL'], default 'VERTEX\_COLOR'

**classmethod is\_registered\_node\_type()**

True if a registered node type

**RETURNS:**

Result

**RETURN TYPE:**

boolean

**classmethod input\_template(index)**

Input socket template

**PARAMETERS:**

**index** (*int in [0, inf]*) – Index

**RETURNS:**

result

**RETURN TYPE:**

[NodeInternalSocketTemplate](#)

**classmethod output\_template(index)**

Output socket template

**PARAMETERS:**

**index** (*int in [0, inf]*) – Index

**RETURNS:**

result

**RETURN TYPE:**

[NodeInternalSocketTemplate](#)

**cache\_point\_density(\*, depsgraph=None)**

Cache point density data for later calculation

**calc\_point\_density(\*, depsgraph=None)**

Calculate point density

**RETURNS:**

RGBA Values

**RETURN TYPE:**

float array of 1 items in [-inf, inf]

**calc\_point\_density\_minmax(\*, depsgraph=None)**

Calculate point density

**RETURNS:**

*min*, min, `mathutils.Vector` of 3 items in [-inf, inf]

*max*, max, `mathutils.Vector` of 3 items in [-inf, inf]

**RETURN TYPE:**

(`mathutils.Vector` of 3 items in [-inf, inf], `mathutils.Vector` of 3 items in [-inf, inf])

**classmethod bl\_ma\_get\_subclass(id, default=None)**

**PARAMETERS:**

**id** (*str*) – The RNA type identifier.

**RETURNS:**

The RNA type or default when not found.

**RETURN TYPE:**

`bpy.types.Struct` subclass

**classmethod bl\_ma\_get\_subclass\_py(id, default=None)**

**PARAMETERS:**

**id** (*str*) – The RNA type identifier.

**RETURNS:**

The class or default when not found.

**RETURN TYPE:**

type

## Inherited Properties

- `bpy_struct.id_data`
- `Node.type`
- `Node.location`
- `Node.location_absolute`
- `Node.width`
- `Node.height`
- `Node.dimensions`
- `Node.name`
- `Node.label`
- `Node.inputs`
- `Node.outputs`
- `Node.internal_links`
- `Node.select`
- `Node.show_options`
- `Node.show_preview`
- `Node.hide`
- `Node.mute`
- `Node.show_texture`
- `Node.bl_idname`
- `Node.bl_label`
- `Node.bl_description`
- `Node.bl_icon`
- `Node.bl_static_type`
- `Node.bl width default`

- `Node.parent`
- `Node.warning_propagation`
- `Node.use_custom_color`
- `Node.color`
- `Node.color_tag`
- `Node.bl_width_min`
- `Node.bl_width_max`
- `Node.bl_height_default`
- `Node.bl_height_min`
- `Node.bl_height_max`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.id_properties_clear`
- `bpy_struct.id_properties_ensure`
- `bpy_struct.id_properties_ui`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_overridable_library`
- `bpy_struct.is_property_readonly`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.pop`
- `bpy_struct.property_overridable_library_set`
- `bpy_struct.property_unset`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Node.socket_value_update`
- `Node.is_registered_node_type`
- `Node.poll`
- `Node.poll_instance`
- `Node.update`
- `Node.insert_link`
- `Node.init`
- `Node.copy`
- `Node.free`
- `Node.draw_buttons`
- `Node.draw_buttons_ext`
- `Node.draw_label`
- `Node.debug_zone_body_lazy_function_graph`
- `Node.debug_zone_lazy_function_graph`
- `Node.poll`
- `Node.bl_rna_get_subclass`
- `Node.bl_rna_get_subclass_py`
- `NodeInternal.poll`
- `NodeInternal.poll_instance`
- `NodeInternal.update`
- `NodeInternal.draw_buttons`
- `NodeInternal.draw_buttons_ext`
- `NodeInternal.bl_rna_get_subclass`
- `NodeInternal.bl_rna_get_subclass_py`
- `ShaderNode.poll`
- `ShaderNode.bl_rna_get_subclass`
- `ShaderNode.bl_rna_get_subclass_py`