Geonode + Gflownet Thesis Back Ground

JayCe Leonard

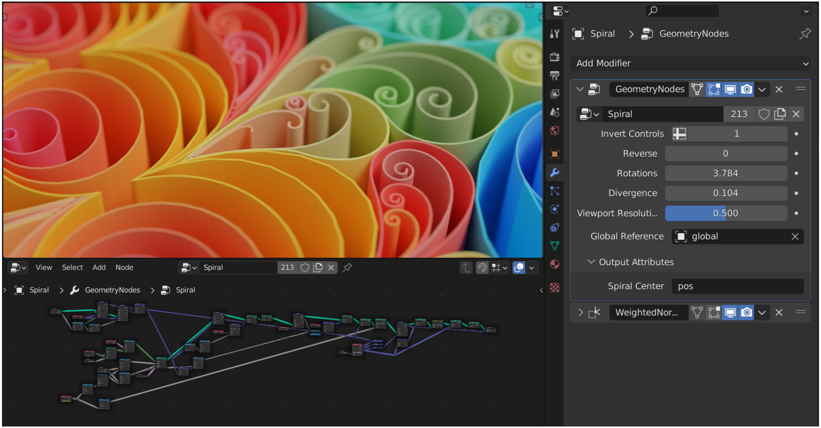
2021

What are Geometry Nodes in Blender?

Geometry Nodes is a powerful, node-based tool in Blender that allows for procedural modeling, animation, and effects through a visual interface.

- **Node-Based Workflow:** Users connect different nodes (such as math, transformations, and geometry modifiers) to build complex, parametric models.
- **Procedural Generation:** Geometry Nodes enable the creation of assets that can be easily modified by adjusting node parametersideal for non-destructive workflows.
- **Key Use Cases:**
  - Procedural asset creation (trees, buildings, terrains).
  - Parametric design for flexible, customizable models.
  - Animation and special effects, such as particle systems.

Geomatry nodes



Invert Controls 1

Reverse 0

Rotations 3.784

Divergence 0.104

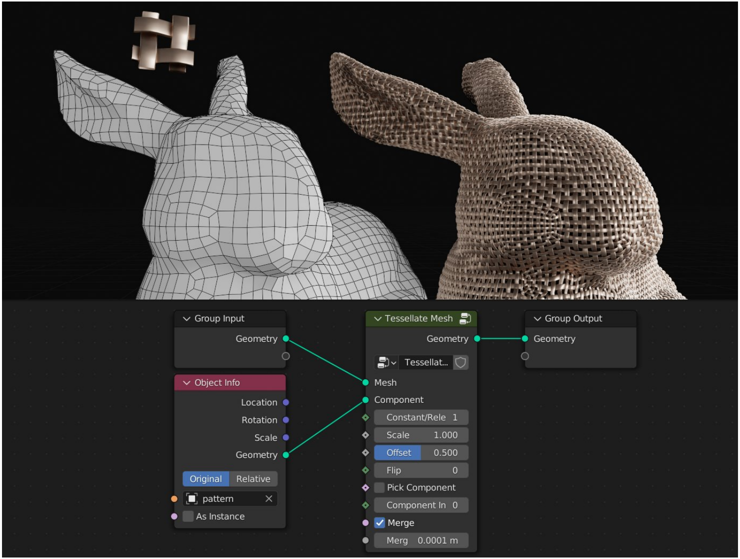
Viewport Resoluti... 0.500

Global Reference global

Spiral Center pos

Figure: click here for link to intro page [2]

Frame Title



Constant/Rele 1

Scale 1.000

Offset 0.500

Flip 0

□ Pick Component

Component In 0

☑ Merge

Merg 0.0001 m

Figure: source

Modular city with Geometry Nodes

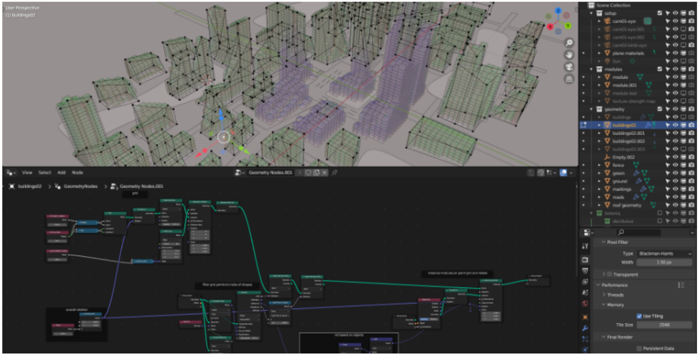


Figure: [4]link

Benefits of Geometry Nodes

- **Non-Destructive**: Adjust models on-the-fly by tweaking node parameters.
- **Reusability**: Node setups can be saved, reused, and shared across different projects.
- **Flexibility**: Suitable for everything from small assets to complex scene generation.

What are GFlowNets?

[1] **Generative Flow Networks (GFlowNets)** are probabilistic models that learn to generate diverse and structured outputs by sampling from a probability distribution over possible solutions.

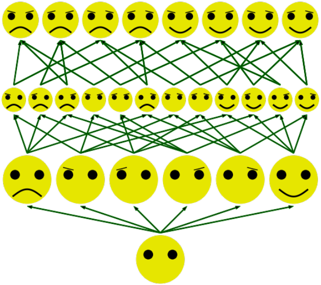
- **Flow-based Model:** GFlowNets define a flow through a directed graph where each node represents a state and edges represent actions.
- **Goal:** The model learns to sample sequences of actions (or paths) that lead to desirable outcomes.
- **Key Feature:** Unlike traditional generative models, GFlowNets aim to sample multiple diverse solutions, rather than just the most likely one.
- **Applications:** Drug discovery, combinatorial optimization, scientific discovery, and in this case, generating compositions in Blenders Geometry Nodes.

Why use GFlowNets?

- Encourages **diversity** of solutions.
- Suitable for **exploratory** tasks where multiple correct outputs exist.
- Can optimize complex, structured environments like **Geometry Nodes in Blender**.

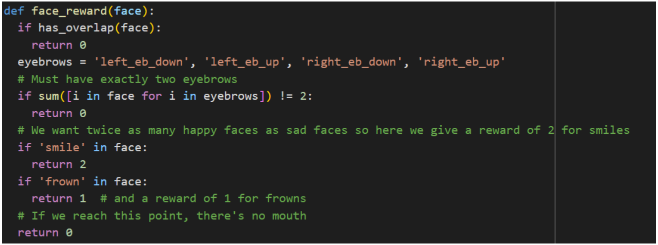
Toy example: state space

- **GFlowNet tutorial** [3]
- here is state space the toy example in the colab uses



- **States**
  1. eyes
  2. right eyebrow
  3. left eyebrow
  4. mouth

Toy example: Loss fn



```python
def face_reward(face):
  if has_overlap(face):
    return 0
  eyebrows = 'left_eb_down', 'left_eb_up', 'right_eb_down', 'right_eb_up'
  # Must have exactly two eyebrows
  if sum([i in face for i in eyebrows]) != 2:
    return 0
  # We want twice as many happy faces as sad faces so here we give a reward of 2 for smiles
  if 'smile' in face:
    return 2
  if 'frown' in face:
    return 1  # and a reward of 1 for frowns
  # If we reach this point, there's no mouth
  return 0
```

- the loss function is saying
  - A make sure there is only 1 of each state in each comosition
  - B reward smiley faces twice as much as frowny faces

Toy example: Result

- notice that the Flow to the smiley face is closer 3 while the flow to the frown is closer to 1.5
  - ≈ 2:1 ratio. just like the loss fn described

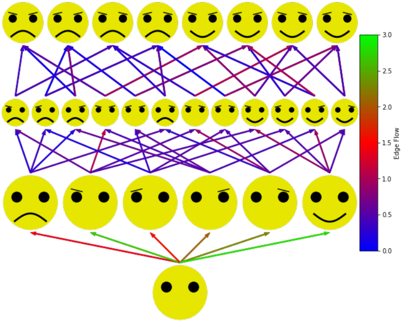
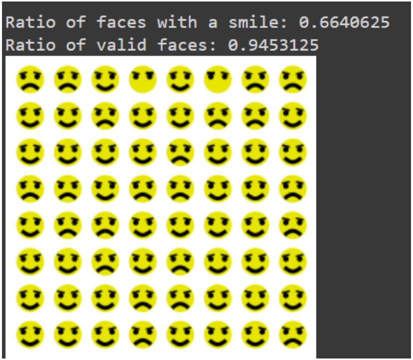


Figure: flow graph



```
Ratio of faces with a smile: 0.6640625
Ratio of valid faces: 0.9453125
```

Figure: Samples

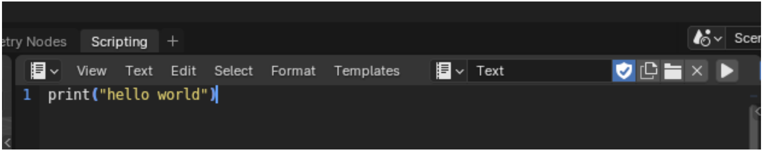
Thesis research

- **Reaserch:** see if gflownets are effective at utilizing Geometry node to meet some desired attribute
  - roughness?
  - a desired frequency of a certain mesh?
  - GPU performance?
  - the list goes on
- **Selling points**
  - Achieves results **without ”stealing”** examples from other artists work
  - Novel. I can’t really find any papers on people who have tried this
  - I think it can be done with the compute resources available

Implementation

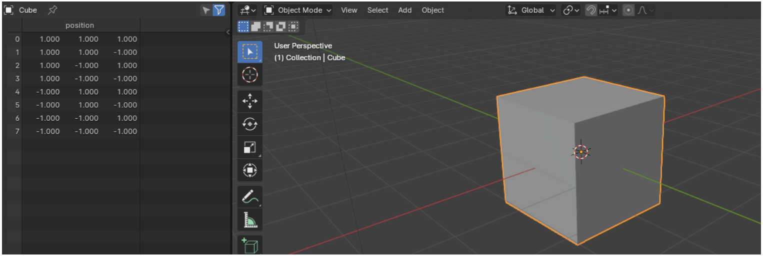
- use blenders scripting library.

  python is literally built into blender



```
print("hello world")
```

- loss fn's will most likely use a point cloud to check their outputs



	position		
0	1.000	1.000	1.000
1	1.000	1.000	-1.000
2	1.000	-1.000	1.000
3	1.000	-1.000	-1.000
4	-1.000	1.000	1.000
5	-1.000	1.000	-1.000
6	-1.000	-1.000	1.000
7	-1.000	-1.000	-1.000

References

[1] Emmanuel Bengio et al. "Flow network based generative models for non-iterative diverse candidate generation". In: *arXiv preprint arXiv:2106.04399* (2021).

[2] Blender Foundation. *Geometry Nodes: Introduction*. Accessed: 2024-09-29. 2023. URL: https://docs.blender.org/manual/en/latest/modeling/geometry_nodes/introduction.html.

[3] Author Name. *Colab Title*. https://colab.research.google.com/drive/1fUMwgu2OhYpQagpzU5mhe9_Esib3Q2VR#scrollTo=w6vut283gILB. Accessed: 2024-09-29. 2023.

[4] UH Studio. *Modular City Geometry Nodes*. https://uhstudio.com/posts/modular-city-geometry-nodes. Accessed: 2024-09-29. 2023.