Art and Machine Learning | Prof. Eunsu Kang & Prof. Barnabas Poczos | TA Jonathan Dinu | Carnegie Mellon Univ. 2018 Spring

Evolving Method of Perception

Jacqui Fashimpaur, Tatyana Mustakos, Char Stiles, Tait Wayland

Concept

We wanted to explore how computers visualize and perceive the world.

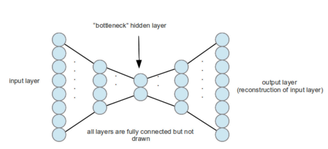
Embracing naïvety, we imagine what a neural net might dream on some random night during its training to become more aware of its surroundings.

In the process of creating and exploring this VR experience, we asked ourself how this computer perception relates to our own, and whether or not there is something universal about the nature of perception.

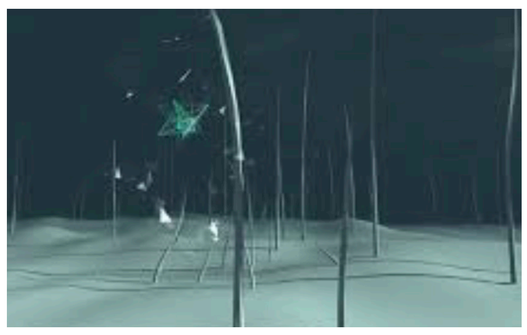
Related Work / Background Research / Literature Review

From a conceptual standpoint, we explored Polyfauna and the interactive experience it affords.

From a machine learning standpoint, we researched autoencoders and looked specifically at architecures that work at representing pointclouds.



Above: AutoEncoder architecture.
Right: Screenshot from Polyfauna.



PointClouds to Unity

PolyFauna is interactive and evolves, but does not emphasize our concept: there is no machine learning involved in this app. Our pipeline is as follows:
1. Train encoder on dancing figure pointclouds.
2. Train encoder on screens pointclouds
3. interpolate, draw from latent space of both.
4. bring interpolated pointclouds into unity
5. create an environment where pointclouds randomly spawn.
6. allow user to interact with pointclouds based on camera orientation.

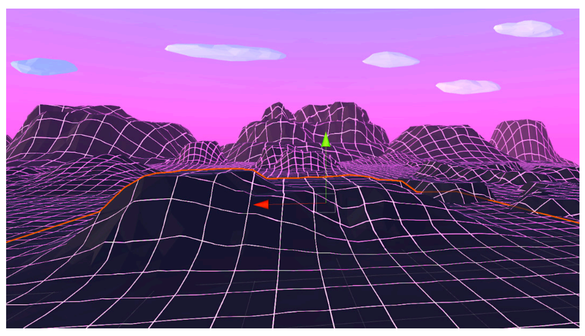
References

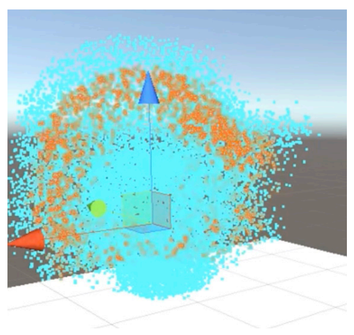
Learning Representations And Generative Models For 3D Point Clouds [https://arxiv.org/pdf/1707.02392.pdf]

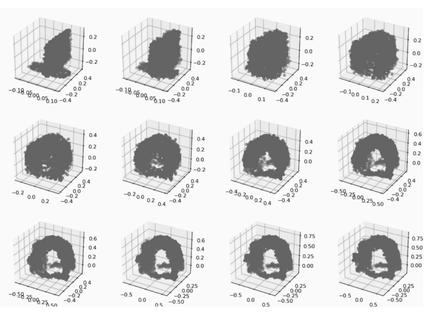
Method / Pipeline / Algorithm / Process

We created a VR experience where the user will interact with point clouds, these point clouds spawn and evolve. The nature of how they transform, and at what speed they transform is affected by the user's camera orientation.

This involved several pipelines and points of coordination among the team. Moving parts involved training an autoencoder on custom point clouds, and then finding an appropriate method to interpolate from the latent space.





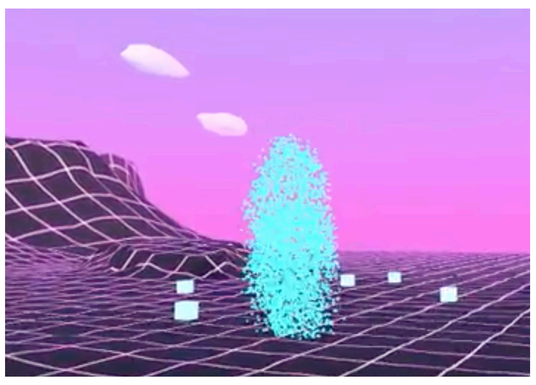


Left: Unity Environment for rendering point clouds. *Middle:* Rendering point clouds in unity. *Right:* Dancers pulled from latent space, generated by autoencoder.

A normal autoencoder was applied on a large point cloud set of a dancing figure. We chose a computer screen as an ideal object to interpolate against, particularly to compliment our concept of computer vision.

On the Unity side, the team used PCX plugin to load point clouds into the engine. Interpolation frames were precomputed and loaded into the scene to prevent any lag issues with loading pointclouds in real time as new objects are spawned.

Some final touches included adding new objects and particle emittors to make the environment richer. Also, UI was added to give the user a more clear sense that they are a computer processing the world in this experience. Finally, this was ported to Android.



Results

While our work intends to study the nature of computer vision, this could be explored in many different ways, and with more focus on specific nuances as they relate to our own perception.

Future work may focus not on the ability of a neural net to see and process the world, but to generate based on its experience. An example project in this space might be to adapt DeepDream, formatted for images in its current state, to build on 3D pointclouds instead.