Vector sketch generation using differentiable rasterization and GAN

Ivan Puhachov
UdeM
ivan.puhachov@umontreal.ca

Abstract

In this project we use differentiable rasterizer recently proposed by Li et al. [2020] to generate artistic vector-graphics images, validating proposed pipeline on a more complex dataset. Vector graphics data representation makes the problem challenging, as it stores stroke information. This makes common image generation techniques unusable, as strokes do not form a linear space. At the same time, vector data allows editing and scaling to any resolution, which is essential for art production (printing and distribution).

1 Overview

Drawings are an important part of creative processes - we draw to discuss ideas, express emotions, outline a prototype. However, drawings are hard to process by machine as they use high-level abstractions to convey shapes and depth information, support structures and auxiliary lines (see Figure 4 for reference). But that's what make sketches interesting, as they are inexact depiction of familiar objects.

Sketch generation is a known problem, however less popular than general image generation, because less training data is available. Another challenging problem lies in data representation: high-quality digital art is often stored in SVG (scalable vector graphics) format, where each stroke and shape is stored as mathematical expressions. This allows for scaling and editing without loss in quality, opposed to raster format (storing pixels - fixed resolution). To visualize vector graphics (and compare it with target image, for example) we need a rasterizer, and for deep learning applications we need a differentiable rasterizer (to pass gradients backward to stroke representations). Recent advancement Li et al. [2020] introduces such a differentiable rasterizer capable of working with different tools and shapes.

1.1 Related works

Most sketch generation works completely discards possible vector graphics structure of drawings and generated images at pixel-level. Recent DoodlerGAN by Ge et al. [2020] was trained on a newly collected expressive dataset "Creative Birds" (see samples on Figure 2) and models drawing as layered bodyparts. The generation is then done by 2-stage process: the first net decides what bodypart to redraw, the second net (GAN) generates and updates the corresponding layer. Although results are gerat, this is hardly different to any other image generation problem.

The first successful vector-graphics sketch generation was done by Ha and Eck [2018]. They collected a massive dataset (QuickDraw) of 50 million of simple drawings, and modelled the process of drawing as sequential decision making. Under the hood, Sketch-RNN has Sequence-to-Sequence Variational Autoencoder. Given the sketch sequence (encoded as a sequence of points + pen drawing status at each point) bidirectional RNN encodes it to latent distribution. Decoder part is an autoregressive RNN, conditioned by a sampled latent vector. Other NLP techniques were succesfully applied to sketch generation, such as Sketch-BERT Lin et al. [2020], where Transformer net was asked to restore corrupted image having the same point-wise encoding. The limitation in NLP approach is in the dataset requirements. Only QuickDraw is big enough to train large models, but the samples are rather primitive (see Figure 1), which affects the images generated by the model.

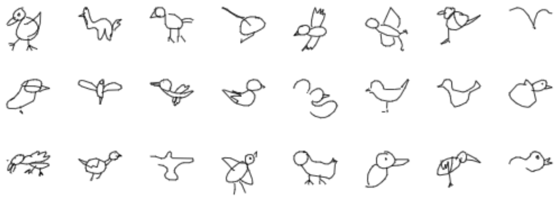


Figure 1: Samples of QuickDraw dataset, by Ha and Eck [2018]

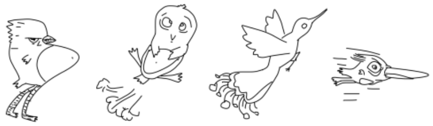


Figure 2: Samples from Creative Birds dataset by Ge et al. [2020]

Reinforcement learning was also applied to drawing generation, following the idea that drawing is a sequential decision-making process - see SPIRAL by Ganin et al. [2018] and SPIRAL++ Mellor et al. [2019]. These systems are expensive to train, as reinforcement agent receives feedback only in the end of drawing process, thus a lot of training attempts is required.

Finally, in the work Li et al. [2020] differentiable rasterizer was proposed and applied in different scenarios, including image generation with VAE and GAN. Wasserstein GAN was trained with a classification discriminator (real or fake), on MNIST dataset and cats from QuickDraw (see results on Figure 5). In this project, we aim to train a generational model for more expressive Creative Birds dataset.

1.2 Idea

Generate artistic images using vector-graphic representation of strokes, differentiable rasterizer and GANs.

Hypothesis We want to confirm that approach described in Li et al. [2020] works for slightly more complex sketches - mainly we want to train WGAN for birds generation with differential rasterizer. See reported results from the paper on Figure 5 and samples from desired dataset on Figure 2.

Future work Inspired by ML-related artwork by Matty Mariansky, we may use differentiable renderer to create adversarial examples for object classification networks, or DeepDream-alike vector images. Also, strokes can be edited for additional coolness of resulting images.

2 Implementation details

We start by reproducing the results from the original paper for a simple MNIST dataset, then moving to the new dataset.

Third-party code We will use differentiable rendering from Li et al. [2020] available at github. We will also use third-party datasets from QuickDraw used in Ha and Eck [2018] and DoodlerGAN from Ge et al. [2020] for experimenting.

Github We will use a personal github repo.

Resources We will use one machine with NVidia 2080Ti GPU (11 GB) and Google Colab to run experiments in parallel.

References

Y. Ganin, T. Kulkarni, I. Babuschkin, S. M. A. Eslami, and O. Vinyals. Synthesizing programs for images using reinforced adversarial learning, 2018.

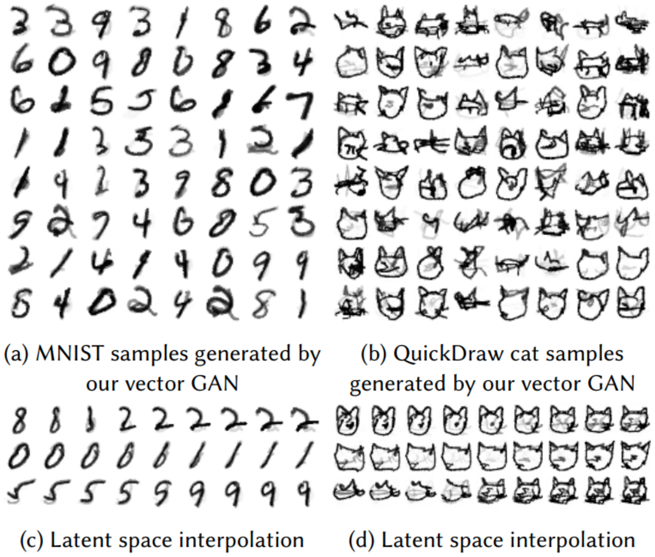
S. Ge, V. Goswami, C. L. Zitnick, and D. Parikh. Creative sketch generation. 2020.

D. Ha and D. Eck. A neural representation of sketch drawings. In *ICLR*, 2018. URL https://openreview.net/pdf?id=Hy6GHpkCW.

T.-M. Li, M. Lukáč, M. Gharbi, and J. Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph.*, 39(6), Nov. 2020. ISSN 0730-0301. doi: 10.1145/3414685.3417871. URL https://doi.org/10.1145/3414685.3417871.

H. Lin, Y. Fu, Y.-G. Jiang, and X. Xue. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6757–6766, 2020.

J. F. J. Mellor, E. Park, Y. Ganin, I. Babuschkin, T. Kulkarni, D. Rosenbaum, A. Ballard, T. Weber, O. Vinyals, and S. M. A. Eslami. Unsupervised doodling and painting with improved spiral, 2019.



(a) MNIST samples generated by our vector GAN

(b) QuickDraw cat samples generated by our vector GAN

(c) Latent space interpolation

(d) Latent space interpolation

Figure 3: Samples generated by GANs, screenshot directly from a paper by Li et al. [2020]

A Additional figures

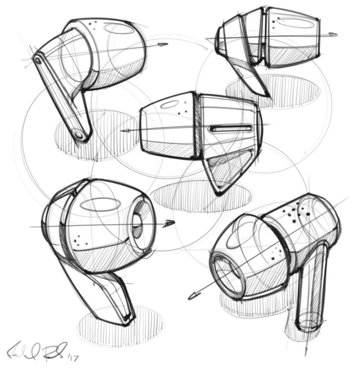


Figure 4: Example of a design sketch. Source: fedriodesign

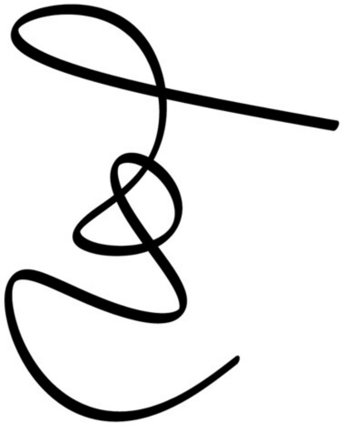


Figure 5: "Evolutionary Faces (2020). A curve generator is pitted against a face recognition algorithm. Using an evolutionary process, the curves thrive to be more face-like. Limiting the curve to just a single or double line forces the generator to remain abstract". Description and image are taken from NeurIPS20 workshop, art by Matty Mariansky