

Генерация шрифта

Автор: Линич А.А.

Научный руководитель: Матвеев И.А.

МФТИ

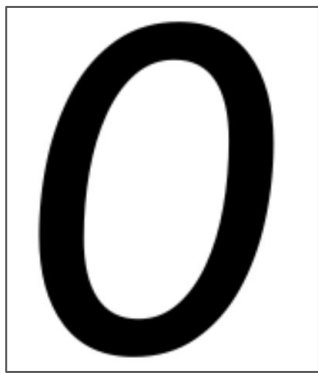
Март, 2024г.

Предмет исследования

Шрифт (упрощенно) – набор глифов (графических представлений символов).

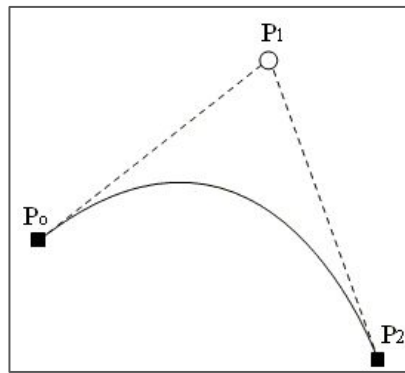
Глиф – список команд различных типов ("moveTo" – переместить перо, "qCurveTo" – провести кривую Безье, "closePath" – оторвать перо и др.) с указанием координат.

Кривая Безье порядка n :
$$f(t) = \sum_{k=0}^n \mathbf{P}_k \binom{n}{k} t^k (1-t)^{n-k}.$$



```
[('moveTo', ((278, -12),)),  
( 'qCurveTo', ((204, -12), (96, 48), (37, 155), (37, 223))),  
( 'qCurveTo', ((37, 310), (108, 450), (236, 532), (323, 532))),  
( 'qCurveTo', ((397, 532), (505, 472), (564, 366), (564, 297))),  
( 'qCurveTo', ((564, 211), (494, 70), (365, -12), (278, -12))),  
( 'closePath', ()),  
( 'moveTo', ((281, 68),)),  
( 'qCurveTo', ((341, 68), (426, 127), (472, 227), (472, 289))),  
( 'qCurveTo', ((472, 335), (436, 408), (368, 452), (320, 452))),  
( 'qCurveTo', ((261, 452), (175, 393), (129, 293), (129, 231))),  
( 'qCurveTo', ((129, 186), (165, 112), (233, 68), (281, 68))),  
( 'closePath', ())]
```

Начертание буквы «о» шрифтом "ABeeZee_italic.ttf" и соответствующий список команд.



Пример кривой Безье третьего порядка.

Актуальность темы

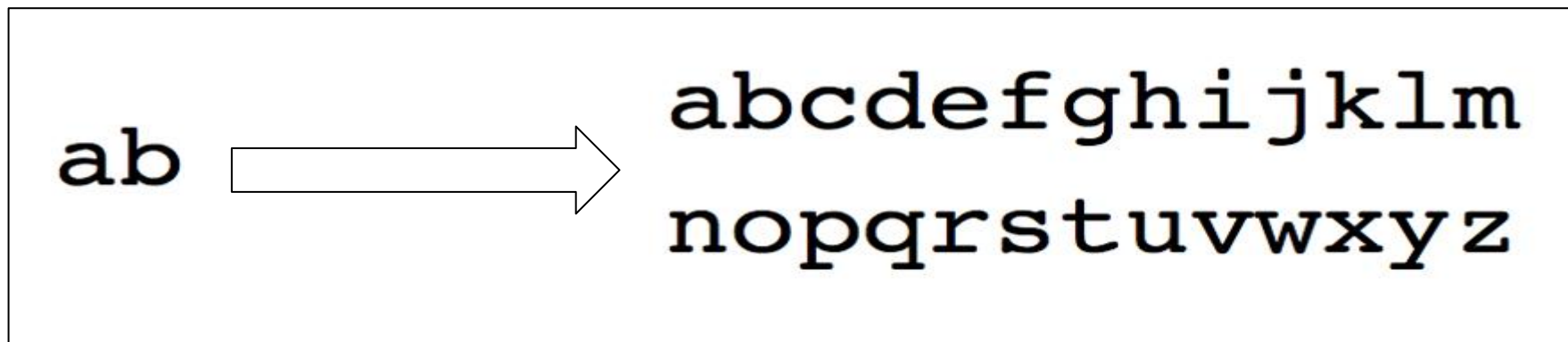
- Трудоемкость создания шрифта для дизайнеров
- Использование неполных шрифтов по набору глифов
- Недостаточное развитие моделей создания векторных изображений



Пример использования одинакового шрифта в надписях различных размеров.

Постановка задачи

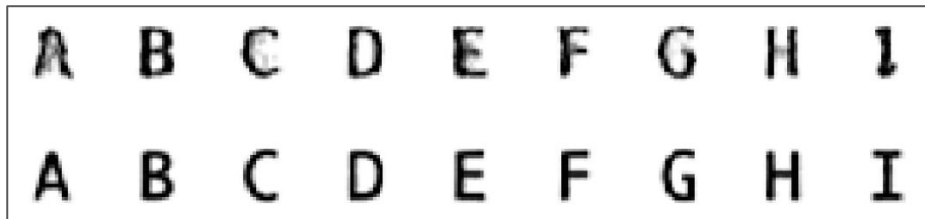
Научиться генерировать стилистически выдержанное векторное графическое представление всех строчных латинских букв по одному или нескольким глифам.



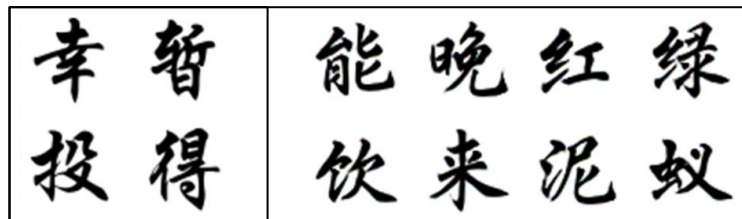
Слева пример входных данных, справа – желательный результат генерации шрифта.

Обзор существующих решений

- Генерация растровых изображений:
 - Обученные с учителем модели ([1]: "Separating Style and Content with Bilinear Models", Джошуа Б. Тененбаум, Уильям Т. Фриман, 1999)
 - Генеративные состязательные сети ([2]: "DG-Font: Deformable Generative Networks for Unsupervised Font Generation", Янчень Се, Синьюань Чень, Ли Сунь, Юэ Лу, 2021)



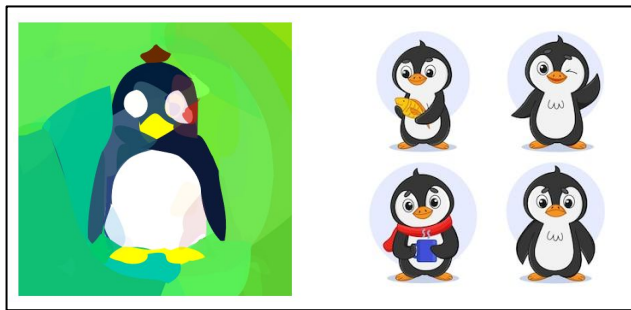
Результаты работы [1]: на нижнем ряду изображены буквы настоящего шрифта, на верхнем – сгенерированного.



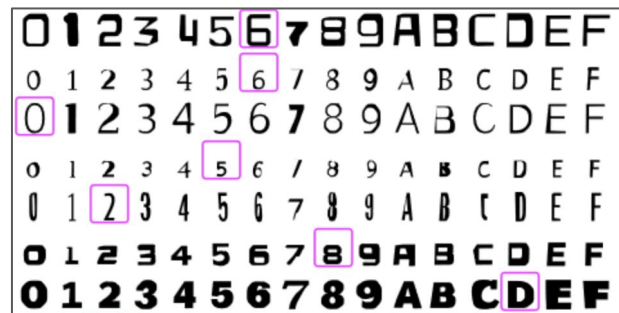
Результаты работы [2]: справа иероглифы настоящего шрифта, слева – сгенерированные.

Обзор существующих решений

- Генерация векторных изображений:
 - Конвертация растровых изображений ([3]: "VectorFusion: Text-to-SVG by Abstracting Pixel-Based Diffusion Models", *Аджай Джейн, Эмбер Се, Питер Аббил*, 2023)
 - Вариационные автоэнкодеры с использованием рекуррентных нейронных сетей ([4]: "DeepSVG: A Hierarchical Generative Network for Vector Graphics Animation", *Александр Карли, Мартин Данелджан, Александр Алахи, Ради Тимофте*, 2019)



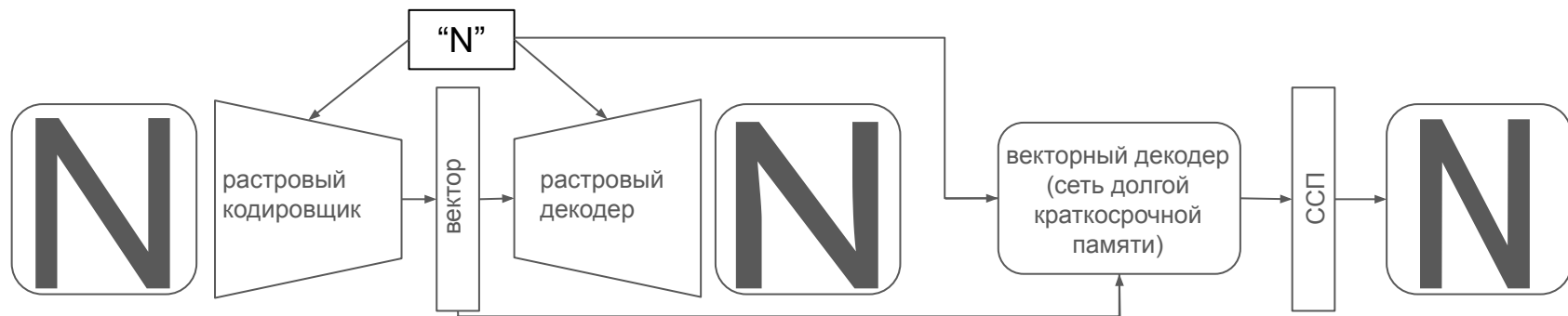
Результат работы [3] слева. Векторное изображение пингвина, сделанное человеком, справа.



Результат работы [4]. В фиолетовой рамке глиф, поданный модели на вход, остальные изображения в ряду – сгенерированные.

Архитектура решения [4]

Автоэнкодер, зависящий от класса



ССП – сеть смеси плотностей.

Функция потерь векторного декодера складывается из кросс-энтропии многопеременной логистической функции для команд и ССП функции потерь, применяемой к координатам команд.

План работ на семестр

1. Взять решение [4] в качестве базового
2. Внести изменения в архитектуру базового решения:
 - Использовать автоэнкодер для последовательностей
 - Перестроить архитектуру векторного декодера, используя механизм внимания
 - Разработать умную функцию потерь векторного декодера
3. Сравнить результаты работы модели с существующими решениями