

# **Neural Strokes: Stylized Line Drawing of 3D Shapes**

Difan Liu<sup>1</sup>      Matthew Fisher<sup>2</sup>      Aaron Hertzmann<sup>2</sup>      Evangelos Kalogerakis<sup>1</sup>  
<sup>1</sup>University of Massachusetts Amherst      <sup>2</sup>Adobe Research

# What is the goal?

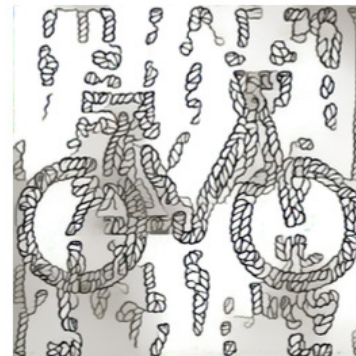
- Image translation/stylization: copy the style of an exemplar image on a target image (image)



[CycleGAN – 2017 – Zhu et al]

# What is the goal?

- Same but with line drawings: problem of keeping the structure of the lines



target

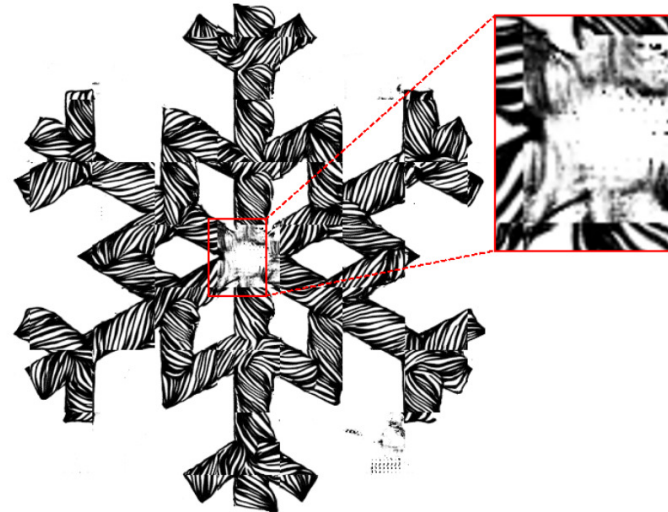
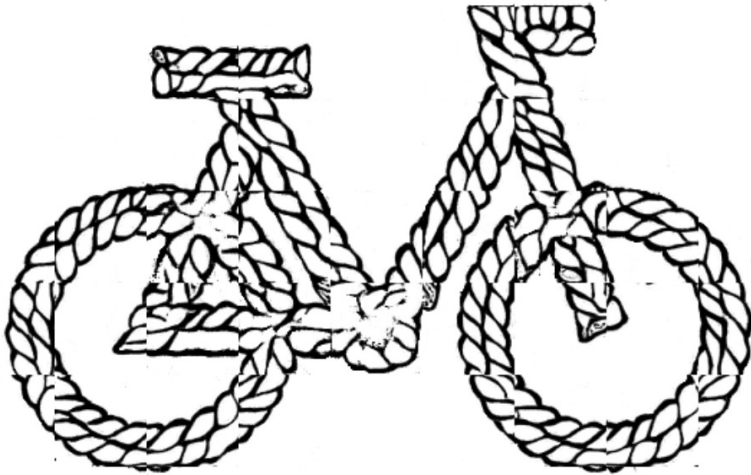
exemplar

Gatys et al.[2016]

Liao et al.[2017]

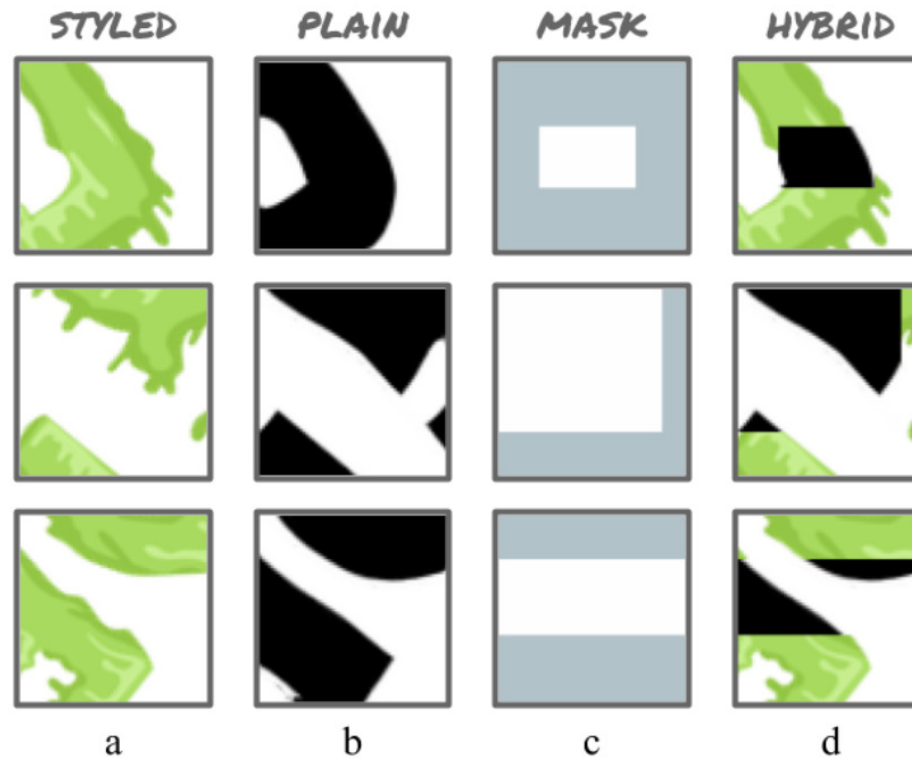
# SketchPatch

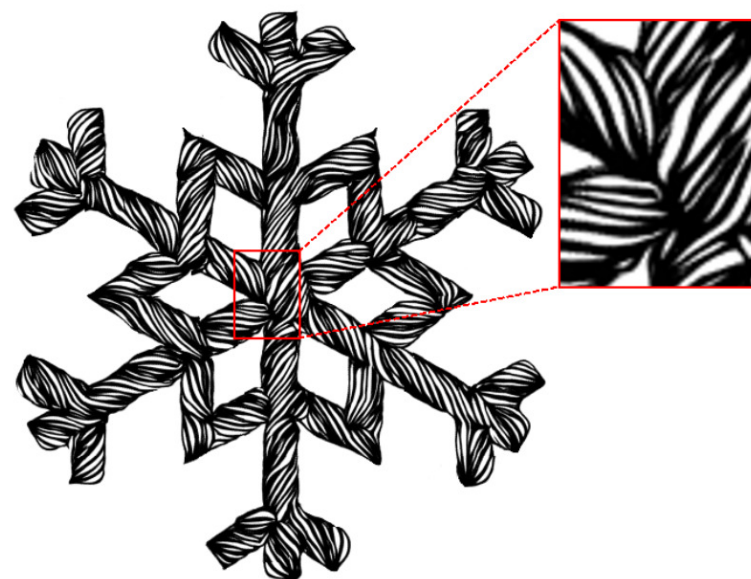
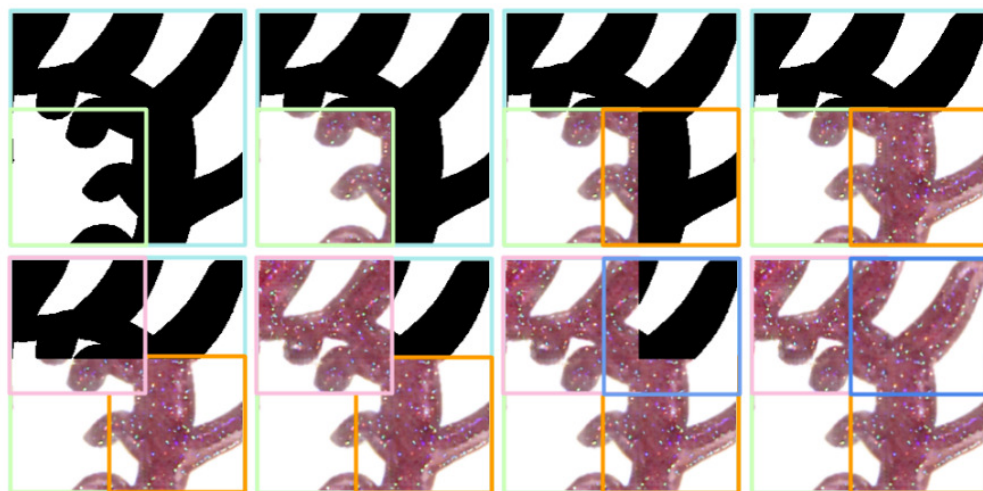
- Global idea: use existing networks (pix2pix) but with clever additions
- 1st idea: use patches of images (more detail, more data)
- Good but visible artifacts (seams)



# Hybrid patches

- Specifically train the network to do seamless transitions between overlapping patches

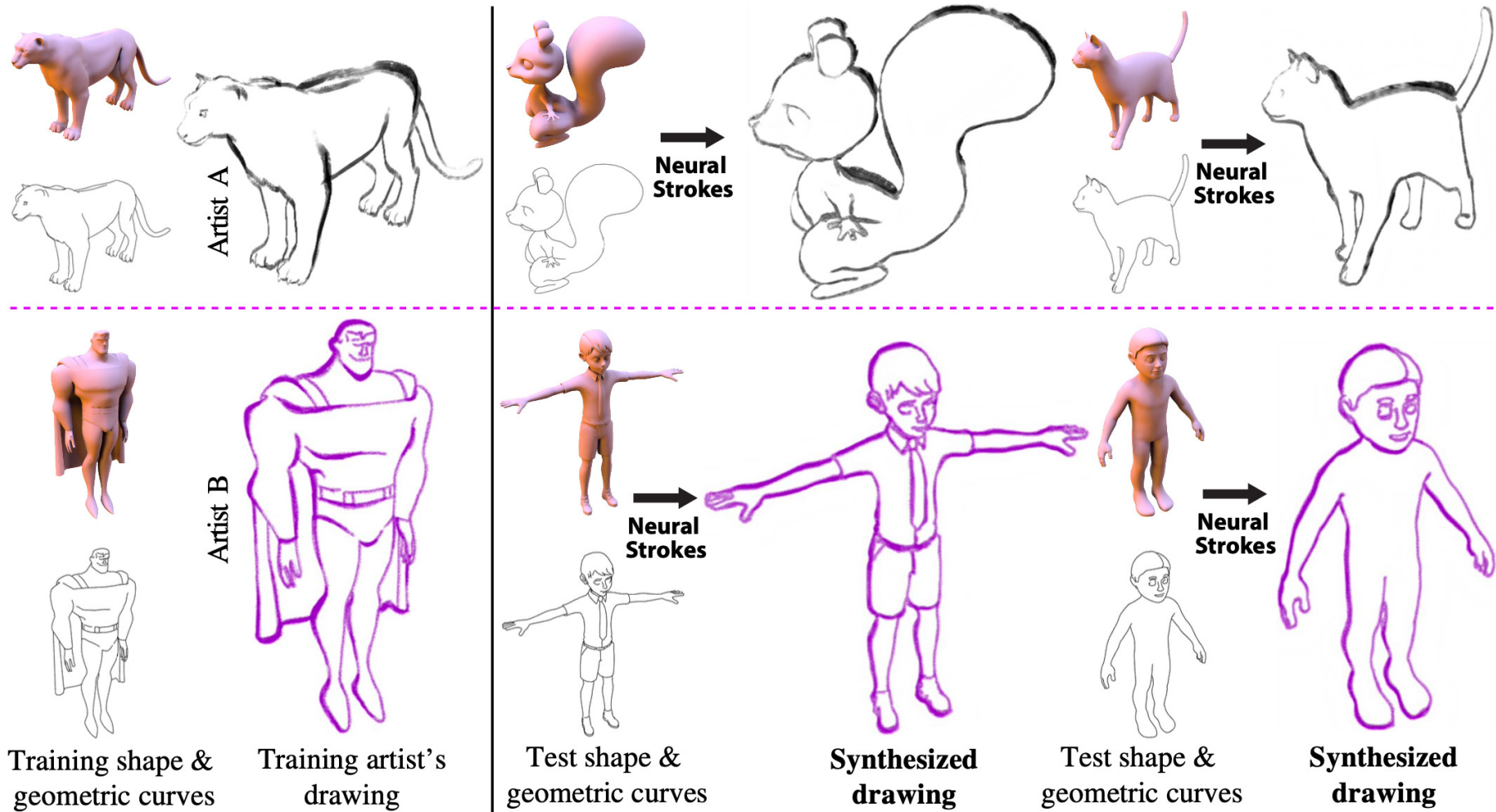




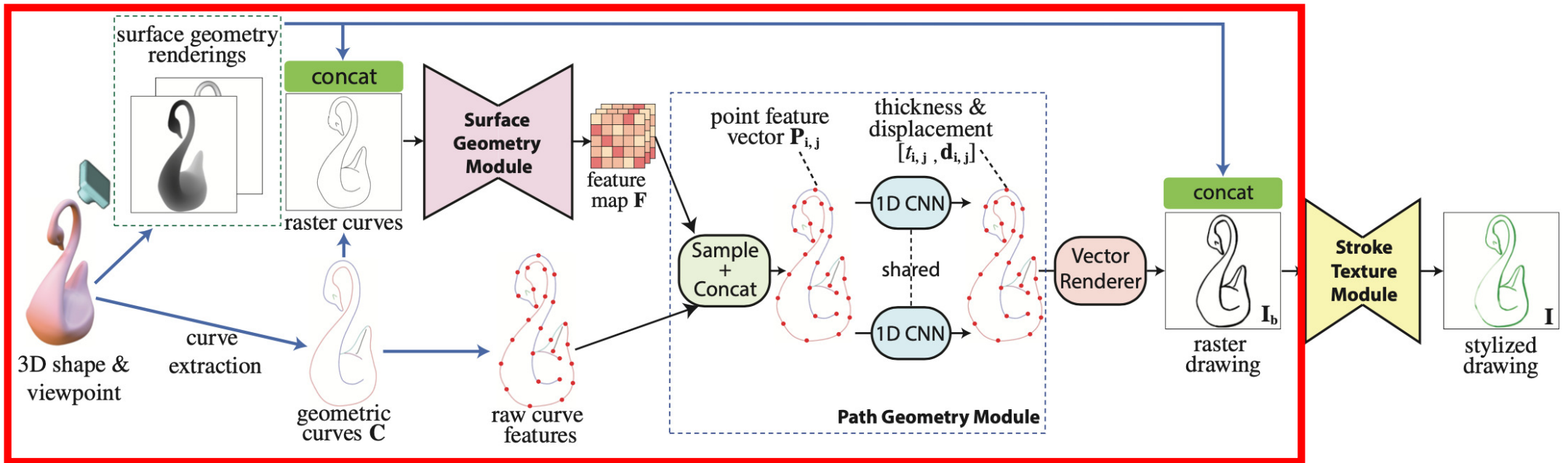
# To conclude on SketchPatch

- Uses existing image translation technique
- Still pixel to pixel
- What if we use vectorized data? More precisely, a 3D shape

# Neural strokes

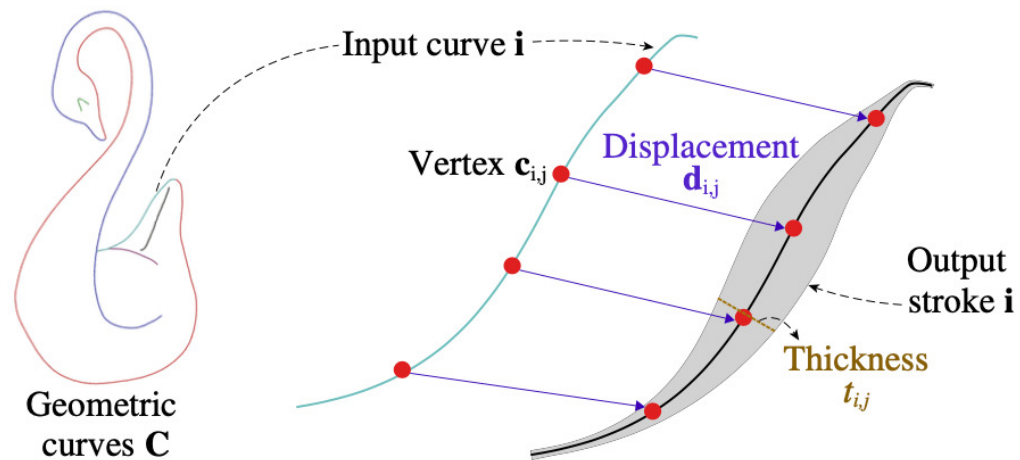


- General idea:
  - Learn how to move and change the thickness of the points (simple black curve)
  - Then add the texture => back to image translation (like SketchPatch)
- Having the geometry helps (stroke may depend on depth or curvature)



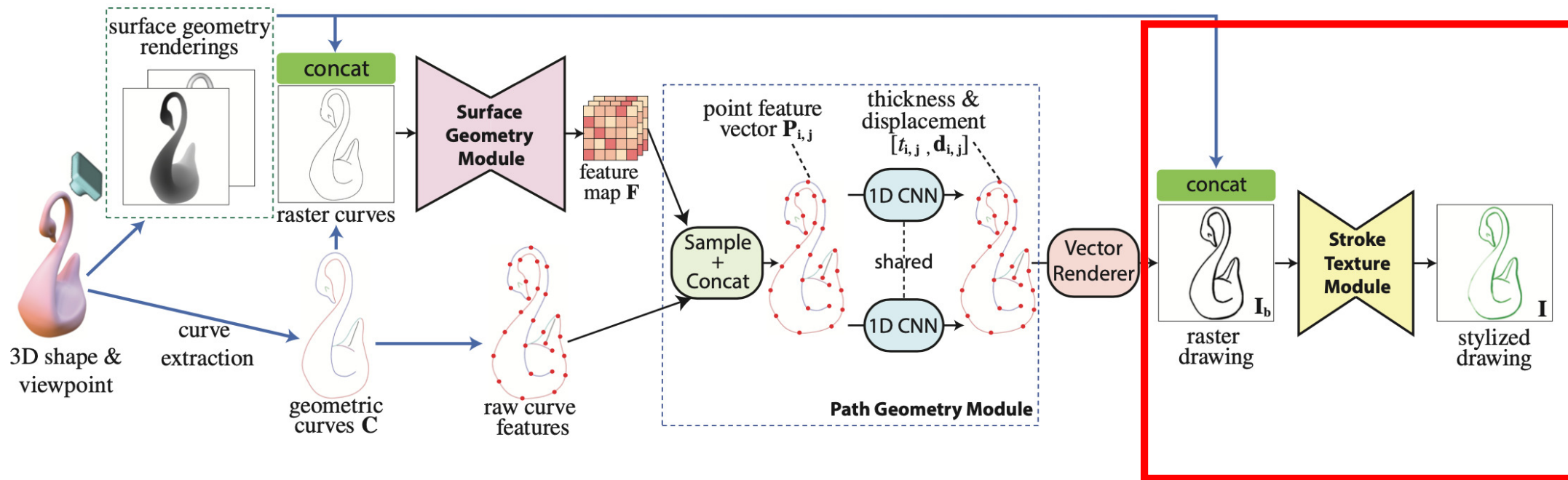
- Goal of the Surface geometry module?

- For each point: geometric + curve features



- Loss : L1+regularization

- Classical image translation => SketchPatch (but simpler)



# To conclude

- Never need to explicitly treat the neighbourhood because it's encoded in the geometric features