Paper Review

Im2vec: Synthesizing vector graphics without vector supervision

YeongHyeon Park

Department of Electrical and Computer Engineering

SungKyunKwan University

Im2Vec: Synthesizing Vector Graphics without Vector Supervision

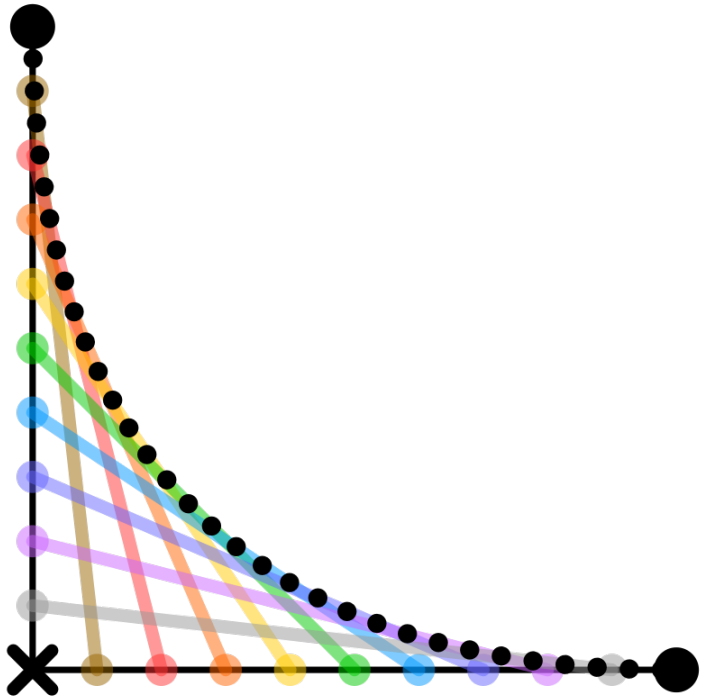
Pradyumna Reddy[1] Michaël Gharbi[2] Michal Lukáč[2] Niloy J. Mitra[1,2]

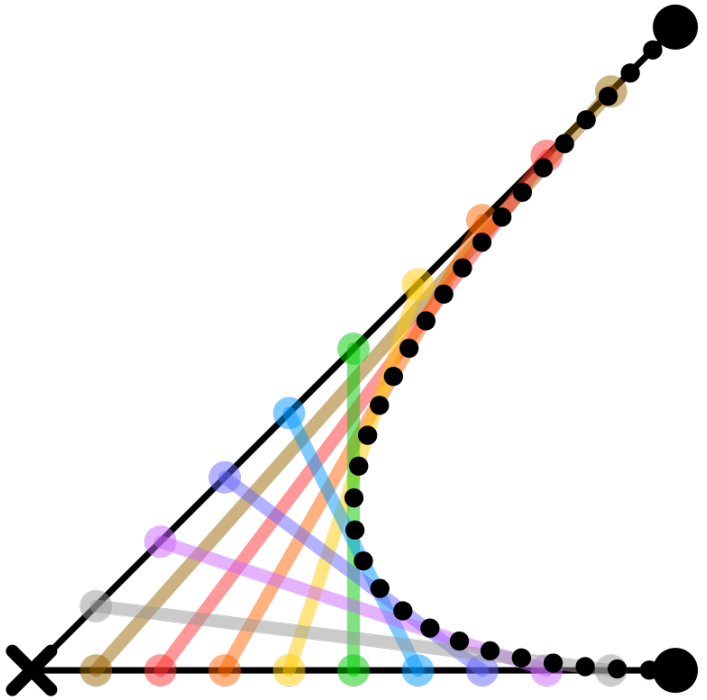
[1]University College London [2]Adobe Research

Reddy, Pradyumna, et al. "Im2vec: Synthesizing vector graphics without vector supervision." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

Warm-Up

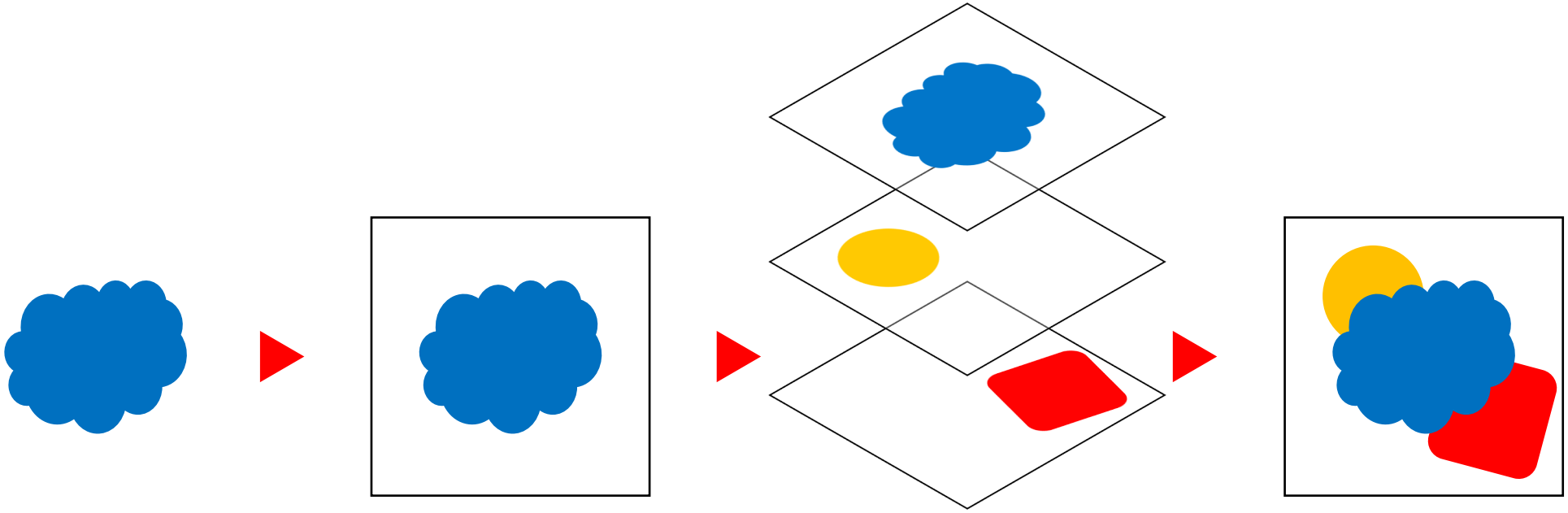
Bézier Curve





https://en.wikipedia.org/wiki/B%C3%A9zier_curve

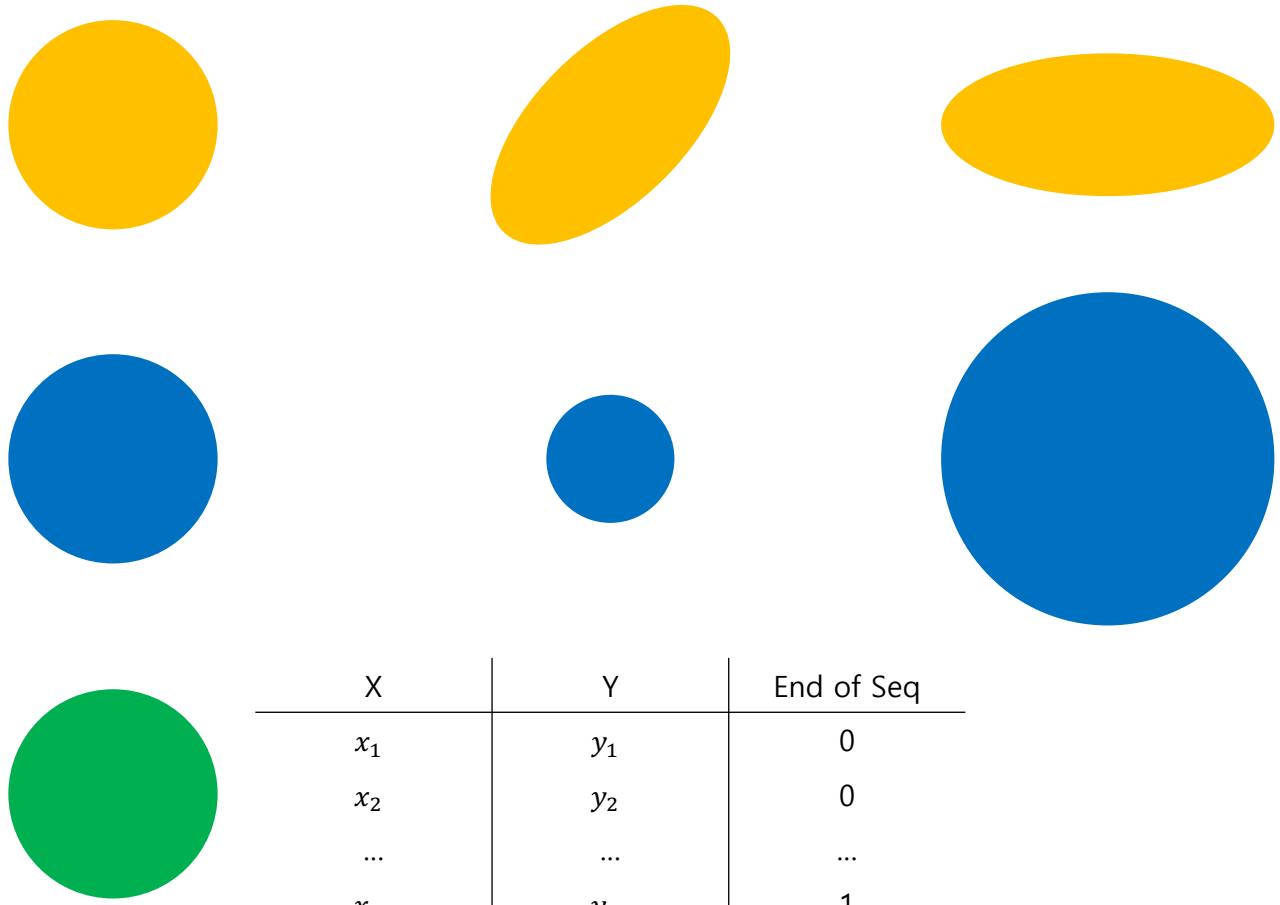
Layer Basics



Component Layer Layers Image

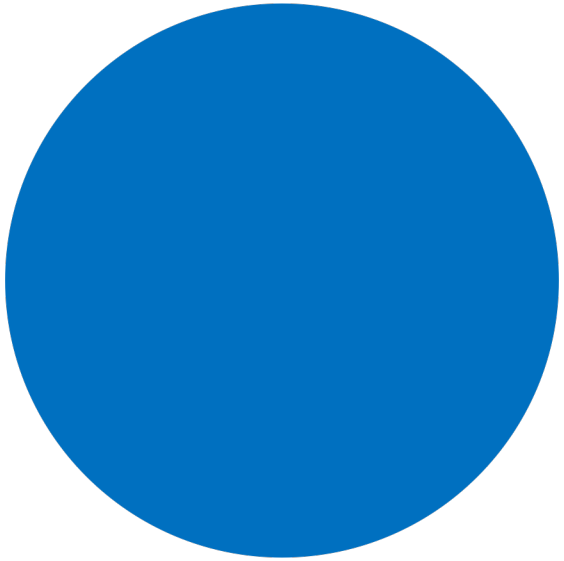
Why Vector?

- Deformable
- Scalable
- Compactness



X	Y	End of Seq
x_1	y_1	0
x_2	y_2	0
...	...	...
x_n	y_n	1

Comparison



SVG

PNG

Related Works

Related Works

Discovering Pattern Structure Using Differentiable Compositing

PRADYUMNA REDDY, University College London
PAUL GUERRERO, Adobe Research
MATT FISHER, Adobe Research
WILMOT LI, Adobe Research
NILOY J. MITRA, Adobe Research and University College London

DiffComp

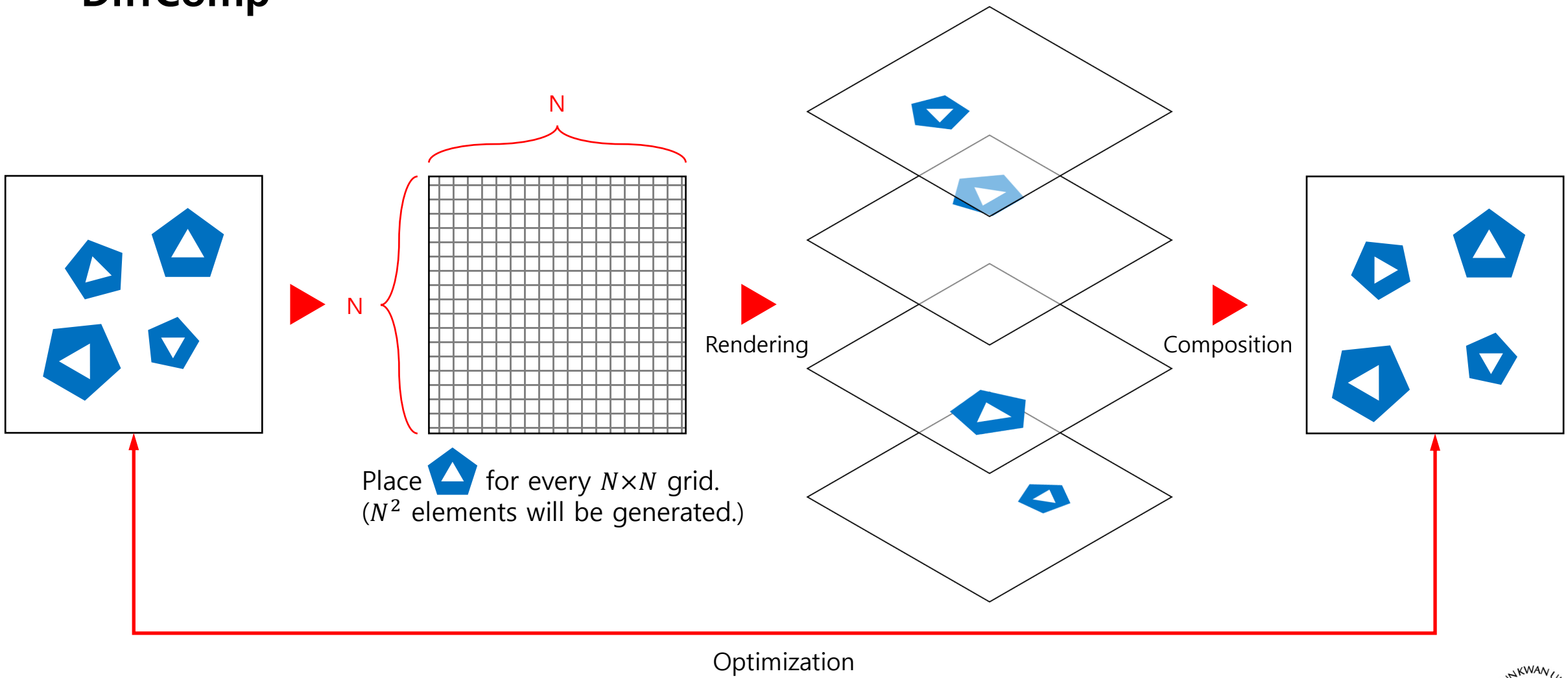
Differentiable Vector Graphics Rasterization for Editing and Learning

TZU-MAO LI, MIT CSAIL
MICHAL LUKÁČ, Adobe Research
MICHAËL GHARBI, Adobe Research
JONATHAN RAGAN-KELLEY, MIT CSAIL

DiffVG

Reddy, Pradyumna, et al. "Discovering pattern structure using differentiable compositing." *ACM Transactions on Graphics*. 2020.
Li, Tzu-Mao, et al. "Differentiable vector graphics rasterization for editing and learning." *ACM Transactions on Graphics*. 2020.

DiffComp



Reddy, Pradyumna, et al. "Discovering pattern structure using differentiable compositing." *ACM Transactions on Graphics*. 2020.

DiffComp

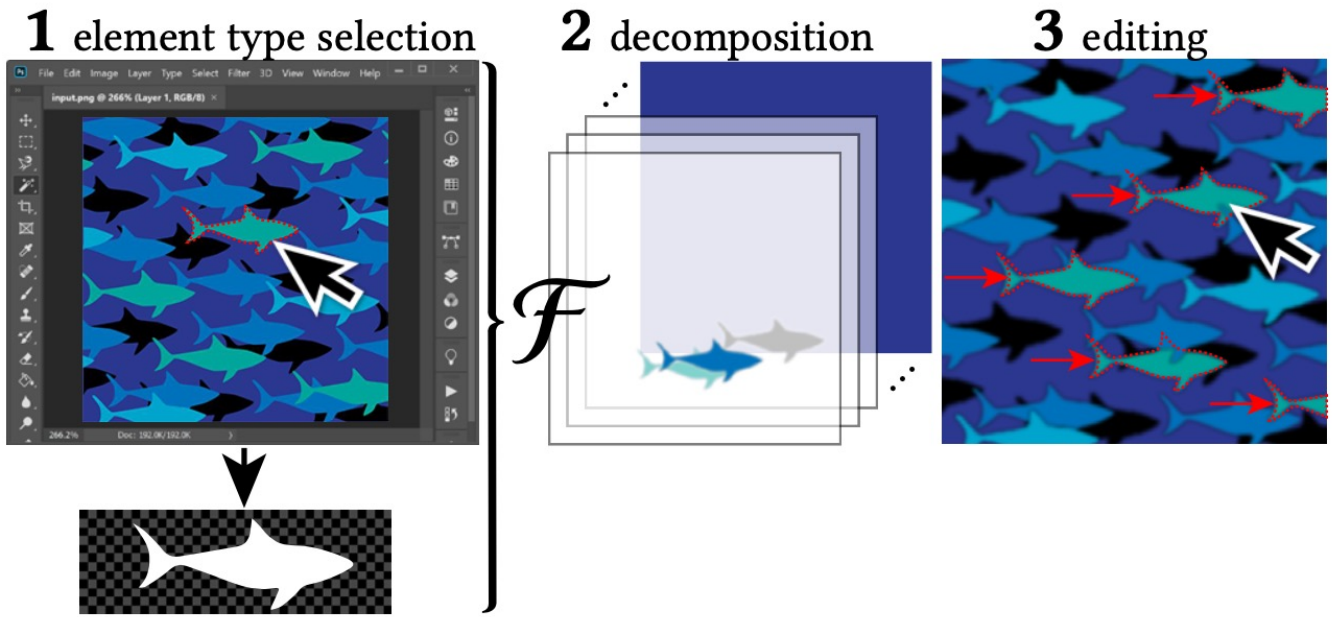
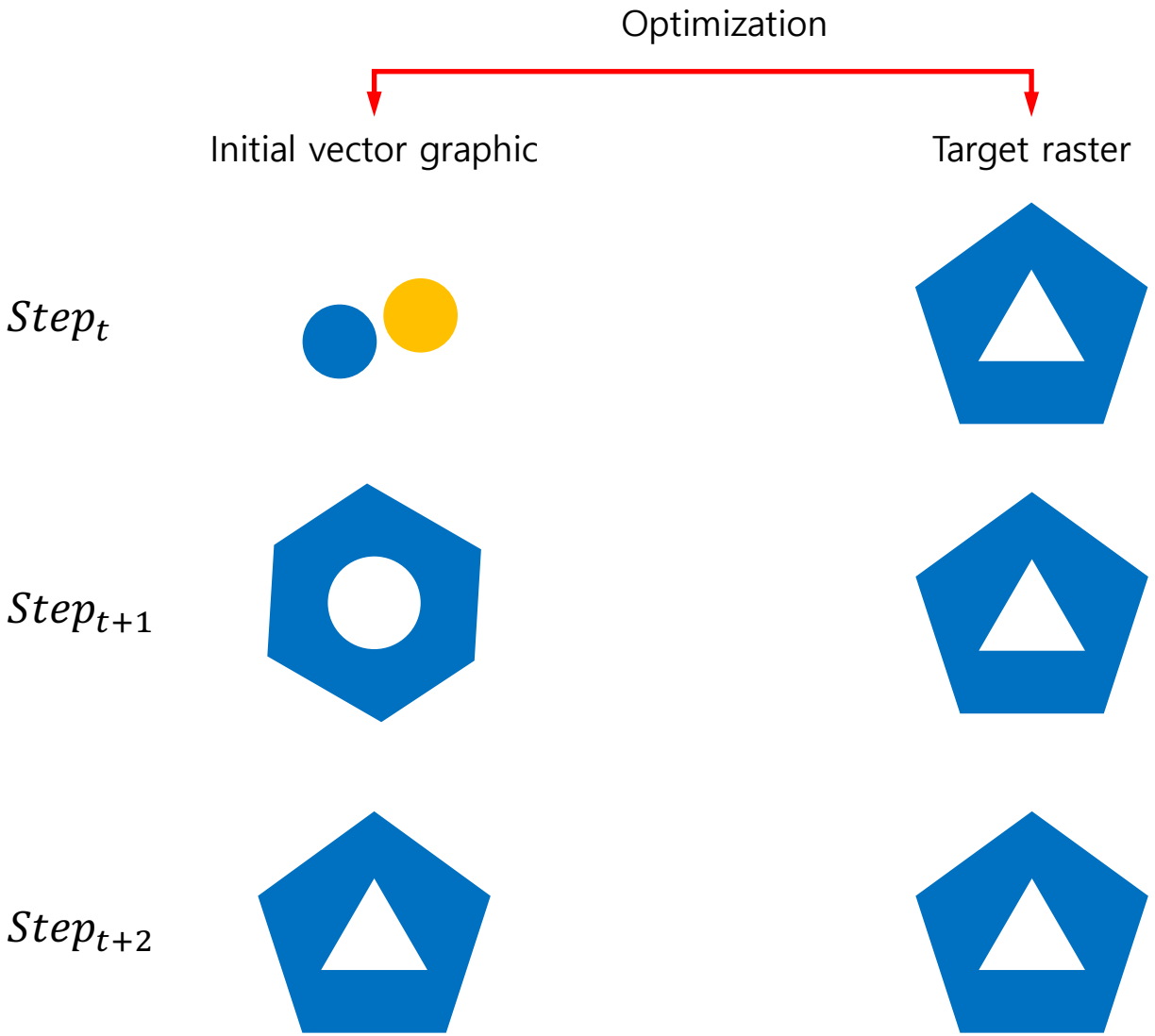


Fig. 6. Pattern decomposition workflow. Starting from a pattern image, the user selects one example of each element type. Here, we only select a single element since we optimize over element colors. The input image and the element types form the input to our differentiable compositor, resulting in discrete elements that we can edit directly or use in down-stream tasks.

Reddy, Pradyumna, et al. "Discovering pattern structure using differentiable compositing." *ACM Transactions on Graphics*. 2020.

DiffVG



Li, Tzu-Mao, et al. "Differentiable vector graphics rasterization for editing and learning." *ACM Transactions on Graphics*. 2020.

DiffVG

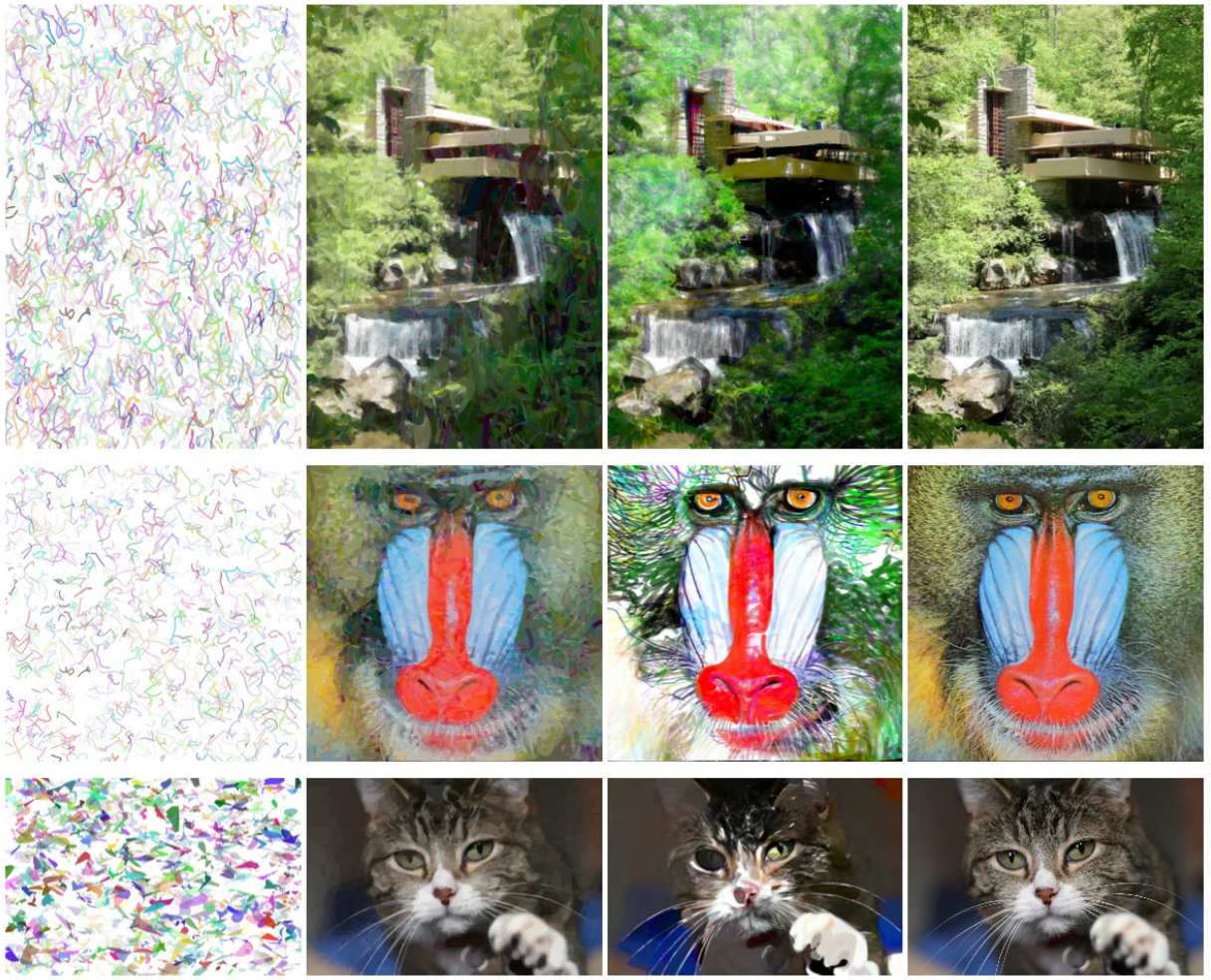


Fig. 13

Li, Tzu-Mao, et al. "Differentiable vector graphics rasterization for editing and learning." *ACM Transactions on Graphics*. 2020.

Im2Vec

Summary of Im2vec

Purpose
- Generate vector graphics from raster images.
- Improve computational efficiency.
  - Computing the same component on every grid (DiffComp).
  - Limitation of active control of the number of components according to input (DiffVG).

Contributions
- Mathematical approach
  - Eliminate the cost of preparing a vector corresponding to a raster (from DiffVG).
- Neural Network Structural Approach
  - Enable active control of the number of components via RNN.
  - Ensure connectivity between Bezier components via Circular CNN.
  - Reduce the computationof redundant components via cartesian coordinate prediction.
  - Improve reconstruction accuracy within a few Bezier segments via adaptive resampler.
- Outperform the vector supervision model without vector supervision.

Limitations
- Small features are sometimes missed.
- Unnecessary features are produced among entangled vector paths.

End-to-End Variational Auto-Encoder

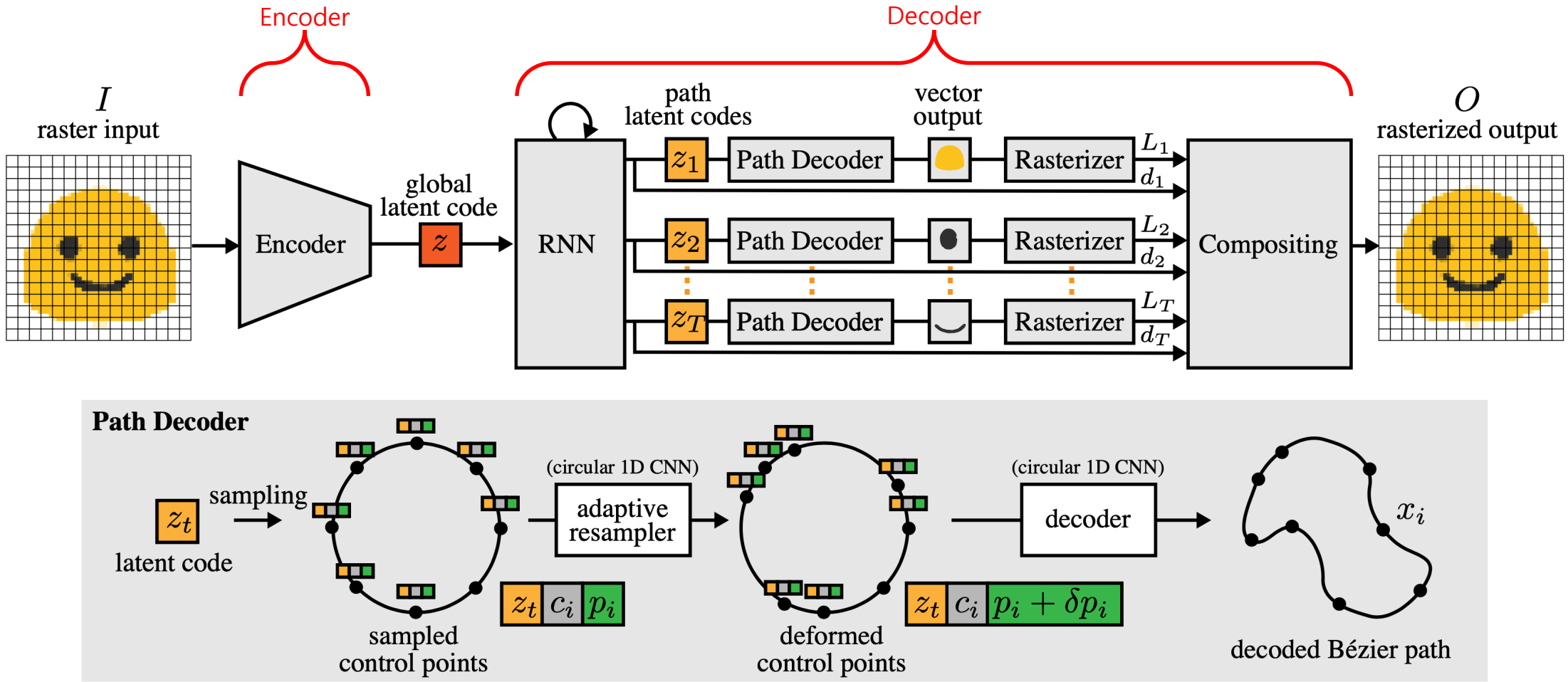
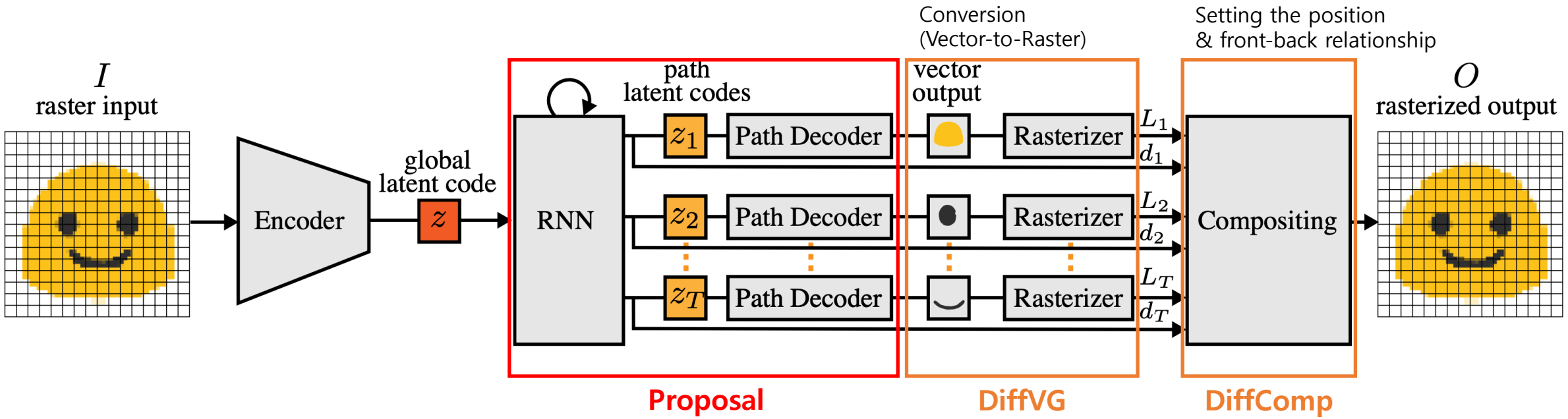
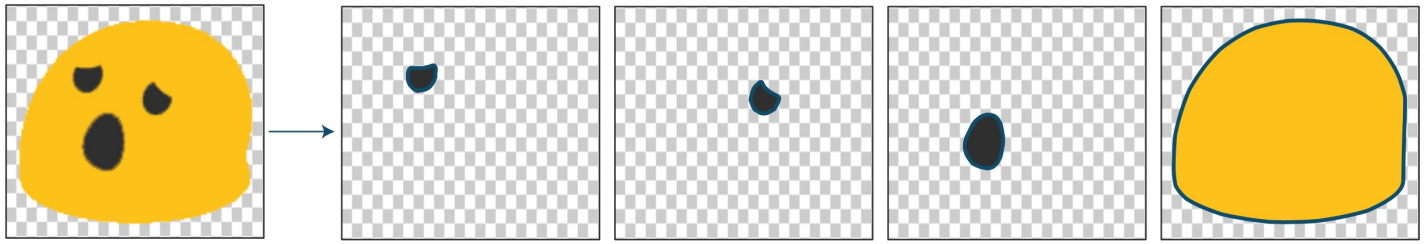


Figure 3: **Architecture overview.**

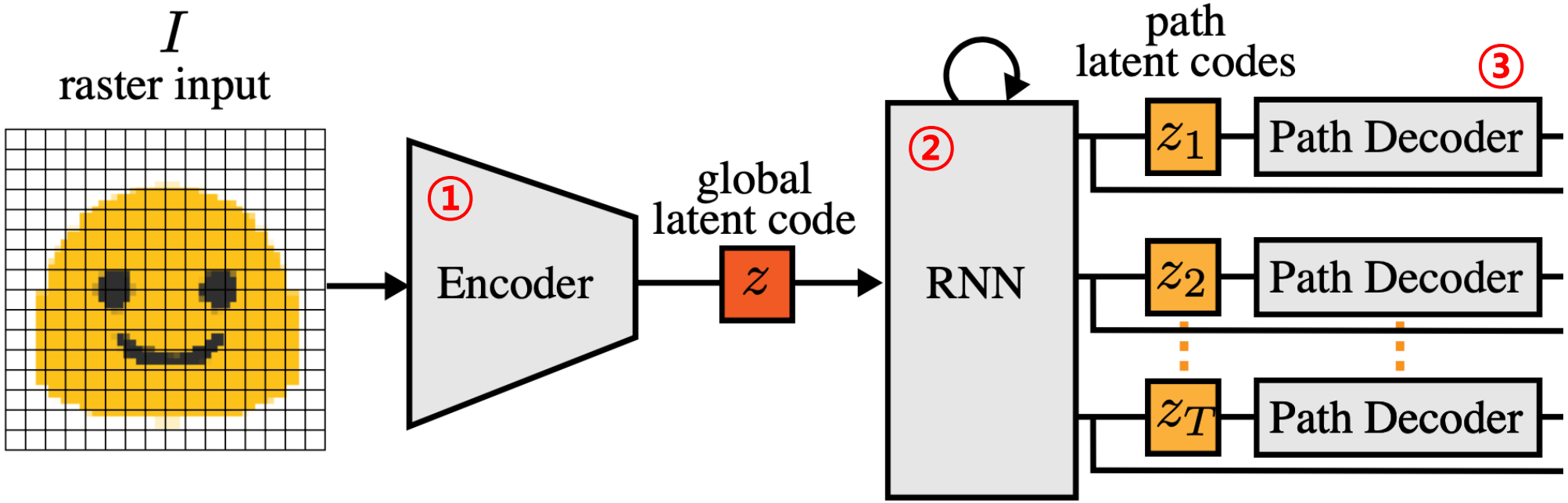
End-to-End Variational Auto-Encoder



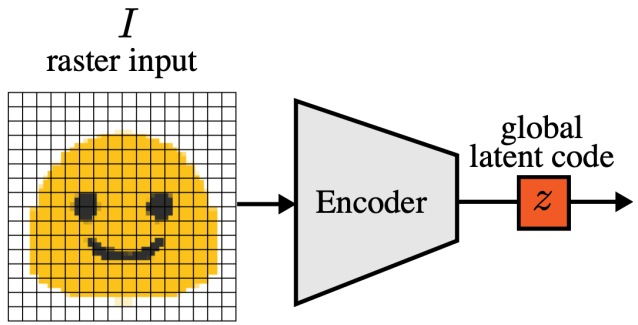


Reddy, Pradyumna, et al. "Discovering pattern structure using differentiable compositing." *ACM Transactions on Graphics*. 2020.
Li, Tzu-Mao, et al. "Differentiable vector graphics rasterization for editing and learning." *ACM Transactions on Graphics*. 2020.

Vector Generation



Step 1 – Convolutional Encoder



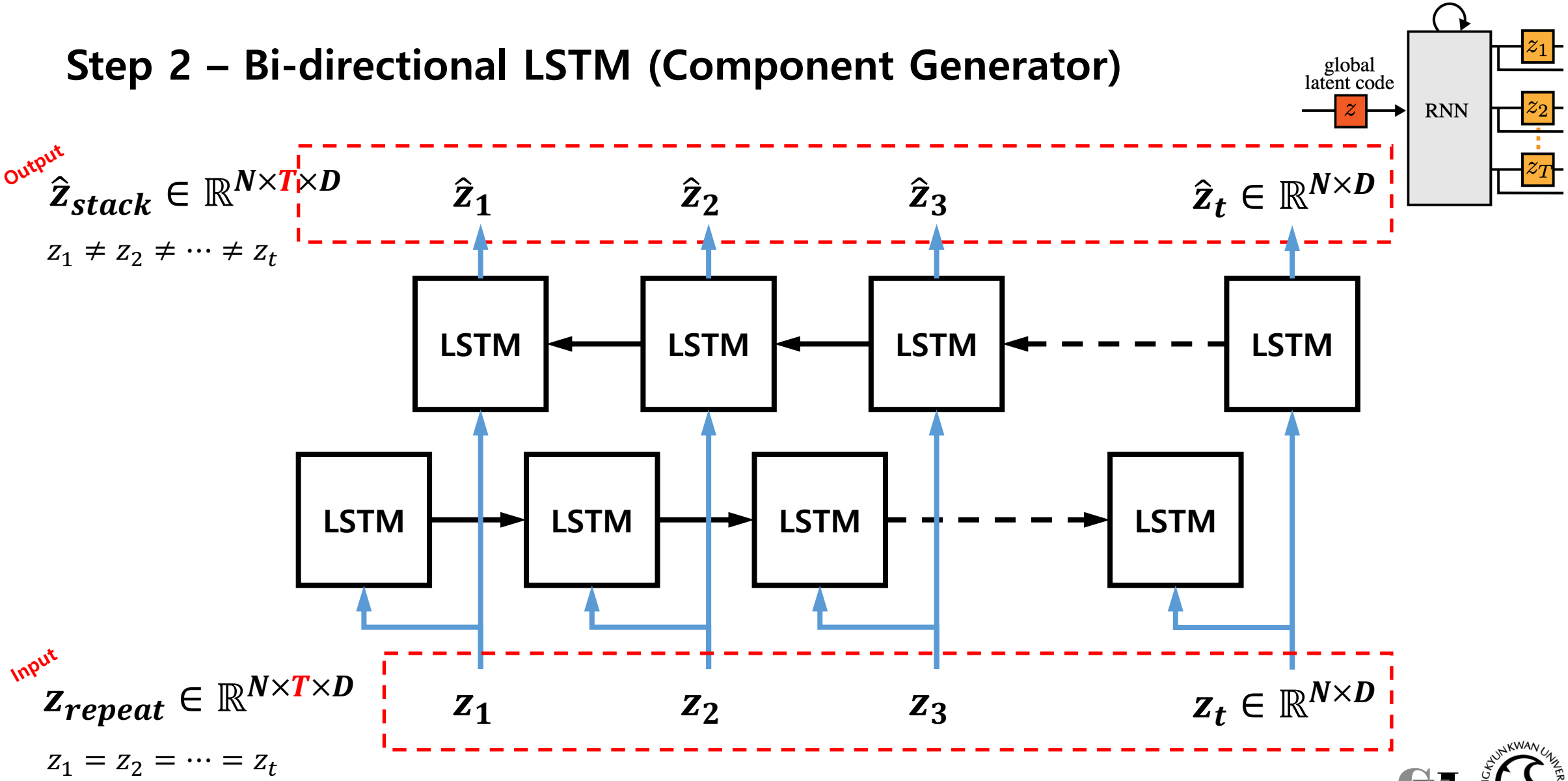
$$\boldsymbol{I} \in \mathbb{R}^{\boldsymbol{N}\times\boldsymbol{H}\times\boldsymbol{W}\times\boldsymbol{C}} \rightarrow \textbf{Encoder} \rightarrow \boldsymbol{z} \in \mathbb{R}^{\boldsymbol{N}\times\boldsymbol{D}}$$

Global Latent Code

D: latent space dimension

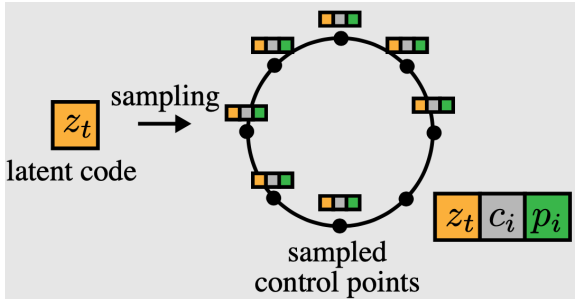
T: number of vector components

Step 2 – Bi-directional LSTM (Component Generator)



B: number of Bezier curve

Step 3 – Path Decoder (1)



$\hat{z} \in \mathbb{R}^{N \times T \times D}$

extract component & repeat (Bx3)

$\hat{z}_t \in \mathbb{R}^{N \times (B \times 3) \times D}$

$c \in \mathbb{R}^{3 \times 2}$

$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$ start point of Bezier curve

repeat (NxB)

$c_i \in \mathbb{R}^{N \times (B \times 3) \times 2}$

concat

$z_{base} \in \mathbb{R}^{N \times (B \times 3) \times (D+2)}$

concat

$z_{angle} \in \mathbb{R}^{N \times (B \times 3) \times (D+2+2)}$

$\theta \in \mathbb{R}^{B \times 3}$

$sin(\theta) \in \mathbb{R}^{B \times 3}$

repeat N

$sin_{repeat}(\theta) \in \mathbb{R}^{N \times B \times 3}$

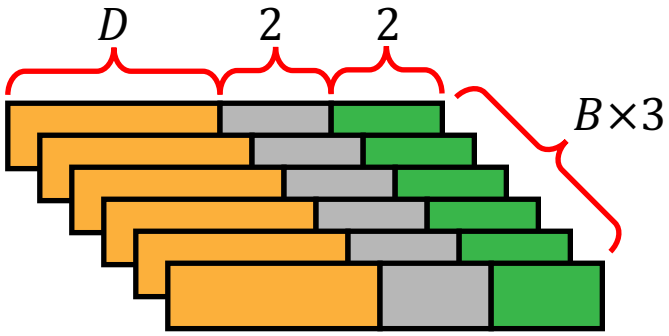
$cos(\theta) \in \mathbb{R}^{B \times 3}$

repeat N

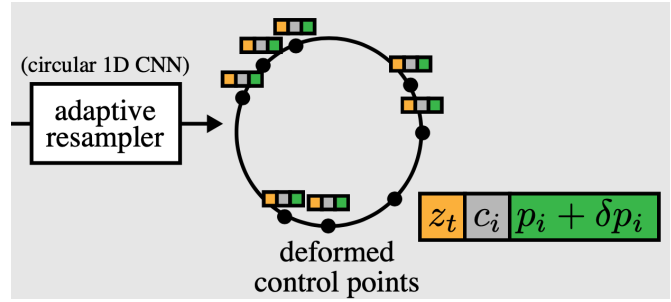
$cos_{repeat}(\theta) \in \mathbb{R}^{N \times B \times 3}$

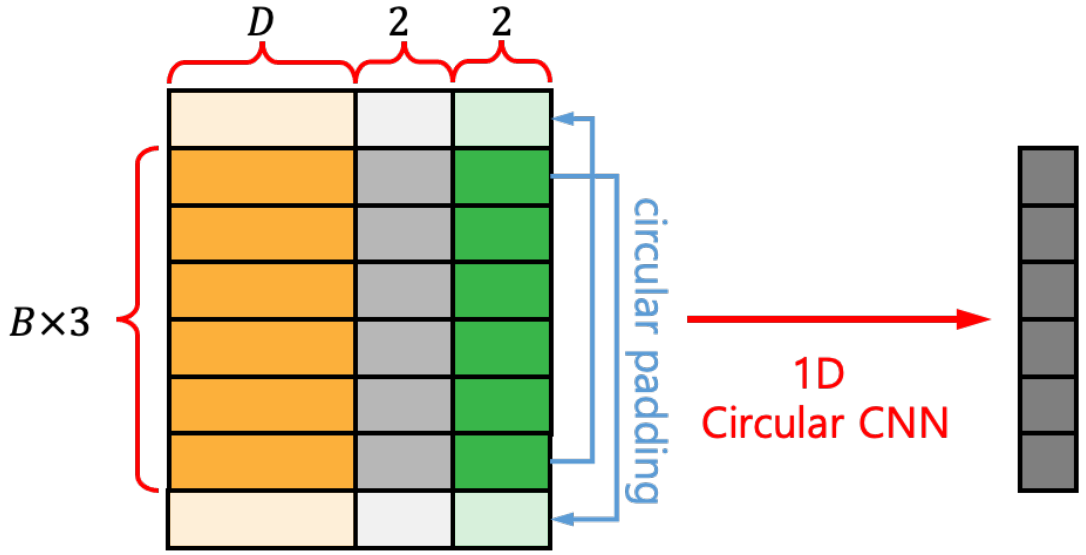
concat

$p_i \in \mathbb{R}^{N \times (B \times 3) \times 2}$



Step 3 – Path Decoder (2) / Adaptive Resampler





1D Circular CNN

$\boldsymbol{z_{angle}} \in \mathbb{R}^{N\times(B\times3)\times(D+2+2)}$

$\boldsymbol{\delta} \in \mathbb{R}^{N\times(B\times3)}$

$\boldsymbol{\theta + \delta} \in \mathbb{R}^{B\times3}$

sin, cos, repeat, concat

$\hat{\boldsymbol{p}}_{\boldsymbol{i}} \in \mathbb{R}^{N\times(B\times3)\times2}$

$\boldsymbol{z_{base}} \in \mathbb{R}^{N\times(B\times3)\times(D+2)}$

concat

$\boldsymbol{z_{deform}} \in \mathbb{R}^{N\times(B\times3)\times(D+2+2)}$

Step 3 – Path Decoder & Raterizer

$$z_{deform} \in \mathbb{R}^{N\times(B\times3)\times(D+2+2)}$$

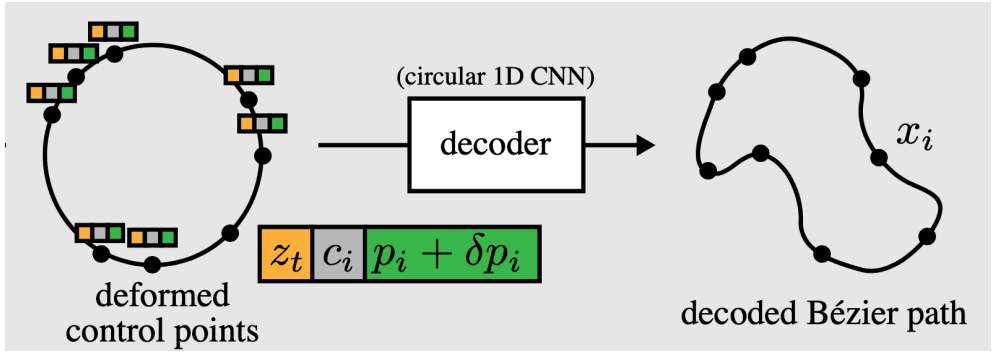
1D
Circular CNN

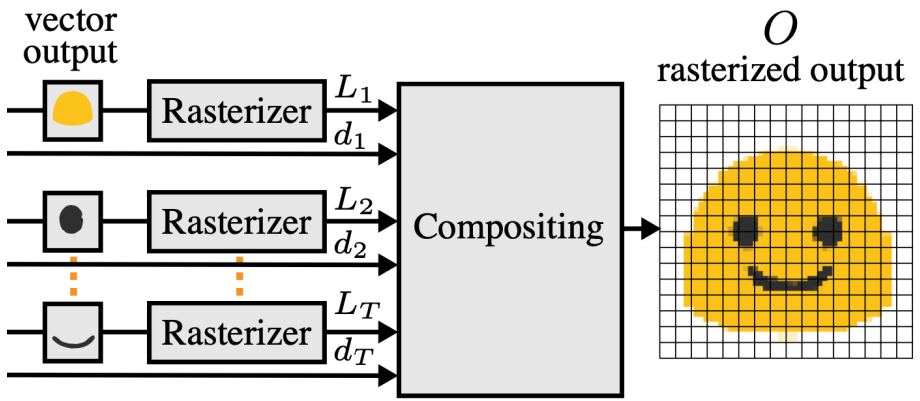
$$x_t \in \mathbb{R}^{N\times(B\times3)\times2}$$

Vector component

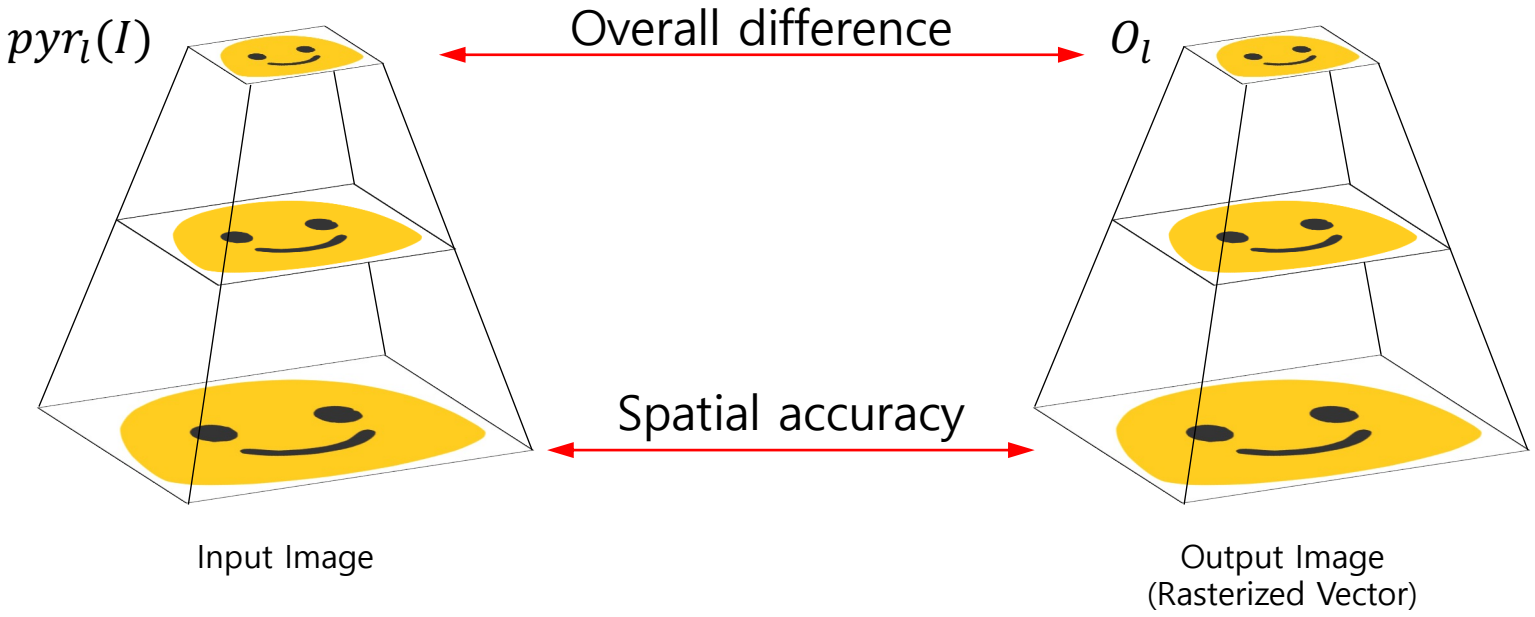
Rasterizing

Compositing





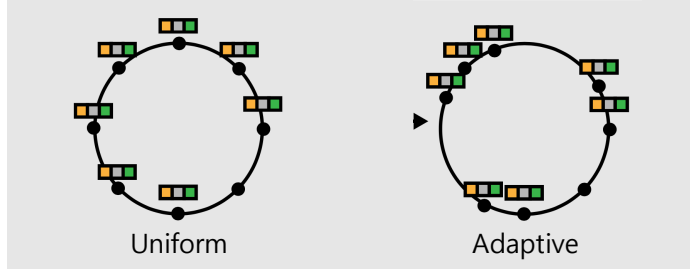
Loss Function / Multi-resolution Raster Loss

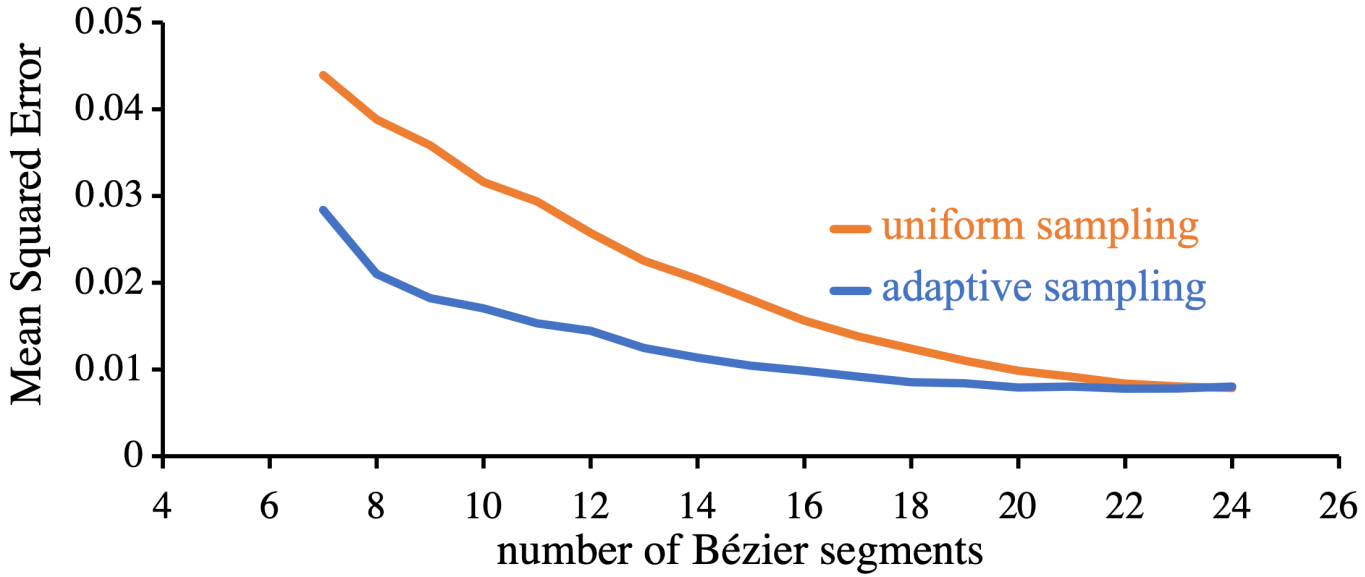


$$\mathbb{E}_{I\sim\mathcal{D}}\sum_{l=1}^{L}\|pyr_l(I)-O_l\|^2$$

Experiments

Comparison – Sampling Method





(b) error vs. number of segments

Figure 4: **Uniform vs. adaptive sampling.**

Comparison – Font / Reconstruction

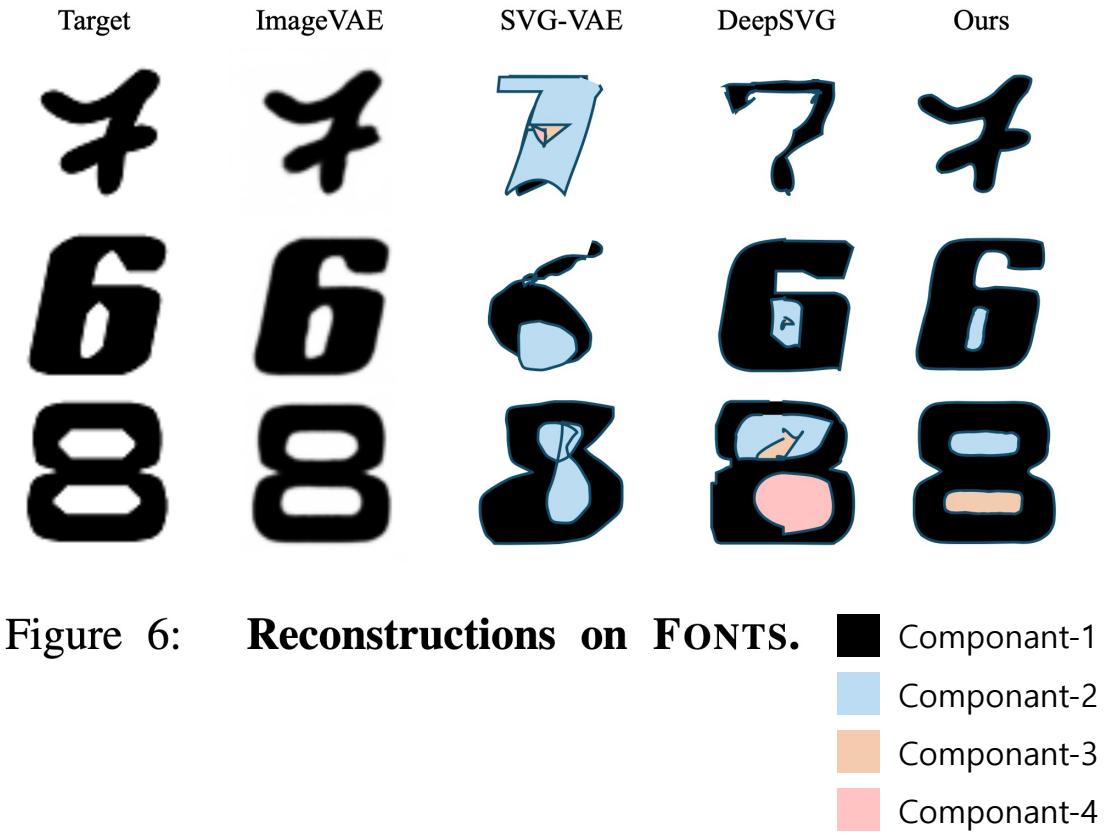


Figure 6: **Reconstructions on FONTS.**

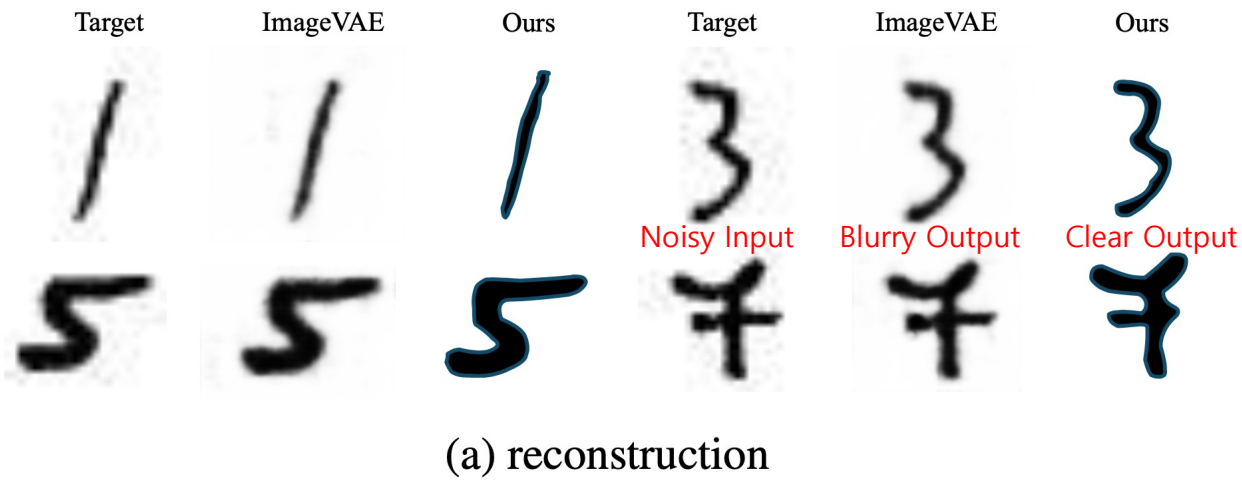
Table 1: **Reconstruction quality.** Comparison of pixel-space reconstruction losses for various methods and datasets. Note that neither SVG-VAE nor DeepSVG operate on datasets without vector supervision.

	FONTS	MNIST	EMOJIS	ICONS
ImageVAE	0.0116	0.0033	0.0016	0.0002
SVG-VAE	0.1322	×	-	-
DeepSVG	0.0938	×	-	-
Im2Vec (Ours)	0.0284	0.0036	0.0014	0.0003

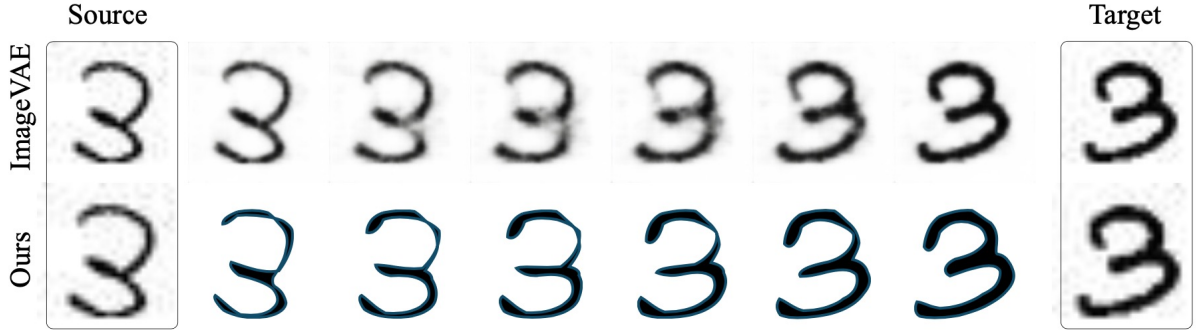
Comparison – Font / Interpolation

(b) Comparison to baselines on FONTS

Comparison – MNIST



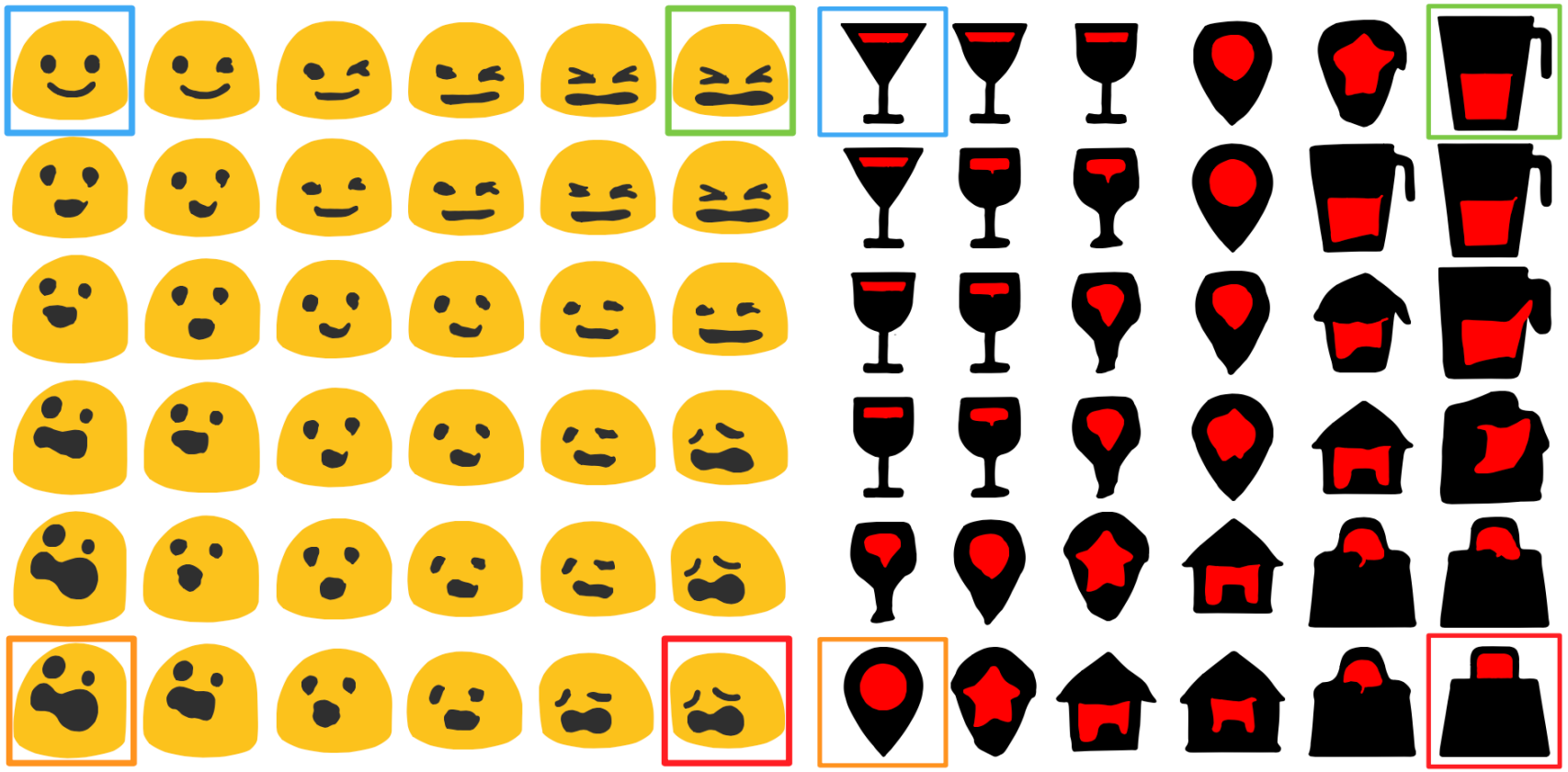
(a) reconstruction



(b) interpolation

Figure 7: **MNIST results.**

Comparison – Emoji & Icons



(a) EMOJIS and ICONS interpolations using Im2Vec

Limitations

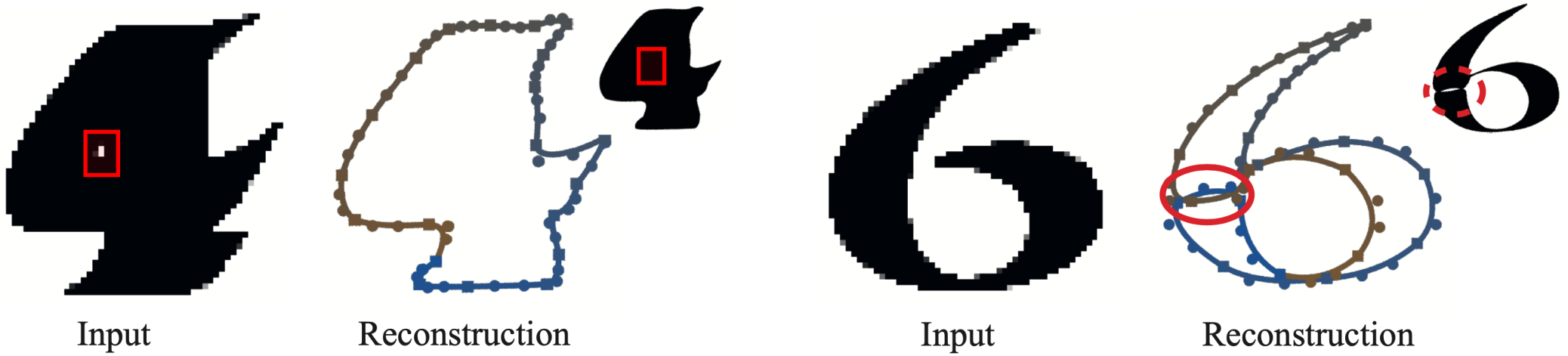


Figure 11: **Limitations.**

Recap. Im2vec

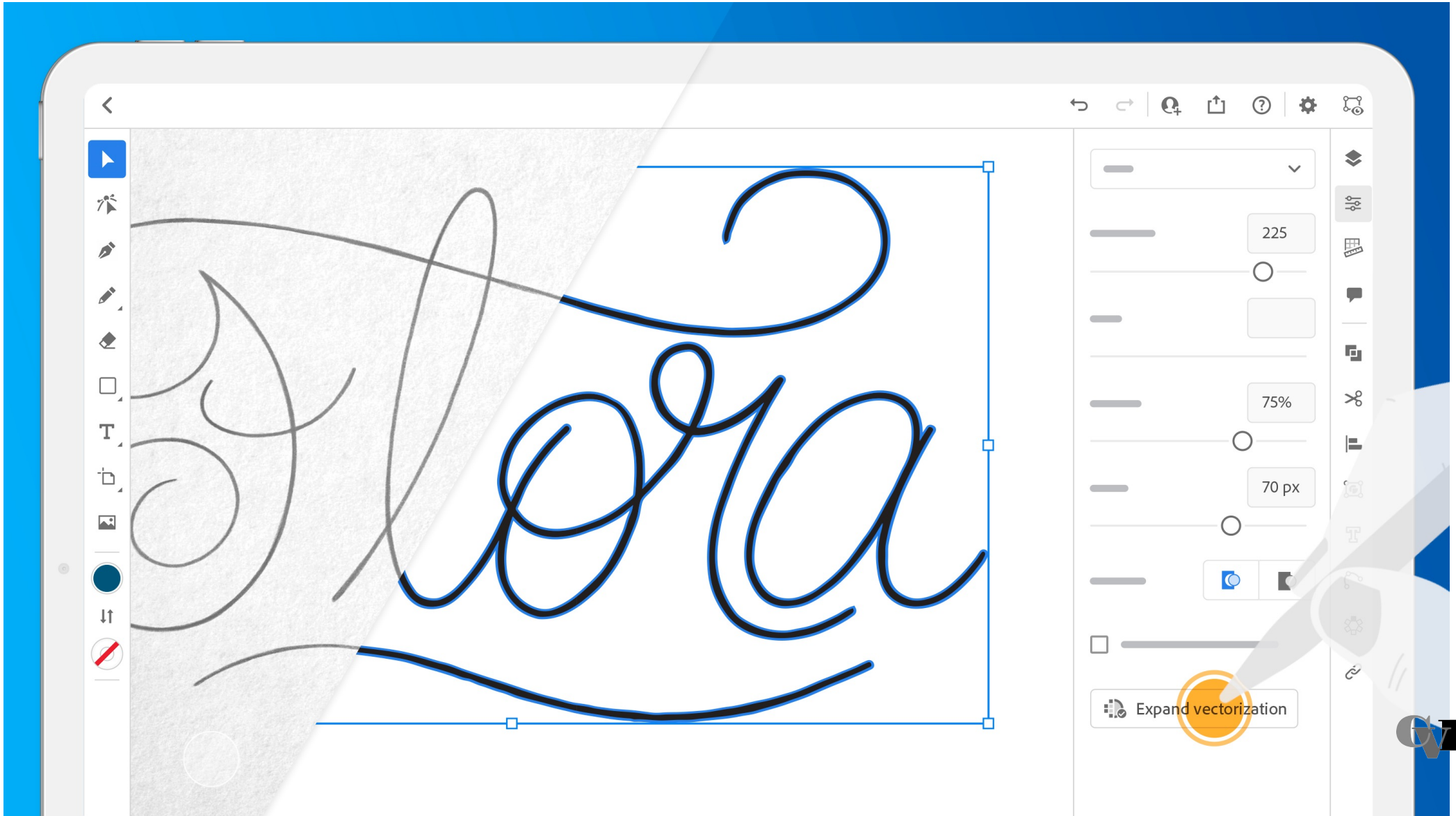
Purpose
- Generate vector graphics from raster images.
- Improve computational efficiency.
  - Computing the same component on every grid (DiffComp).
  - Limitation of active control of the number of components according to input (DiffVG).

Contributions
- Mathematical approach
  - Eliminate the cost of preparing a vector corresponding to a raster (from DiffVG).
- Neural Network Structural Approach
  - Enable active control of the number of components via RNN.
  - Ensure connectivity between Bezier components via Circular CNN.
  - Reduce the computationof redundant components via cartesian coordinate prediction.
  - Improve reconstruction accuracy within a few Bezier segments via adaptive resampler.
- Outperform the vector supervision model without vector supervision.

Limitations
- Small features are sometimes missed.
- Unnecessary features are produced among entangled vector paths.

Illustrator on the iPad, Adobe, Oct 2022.

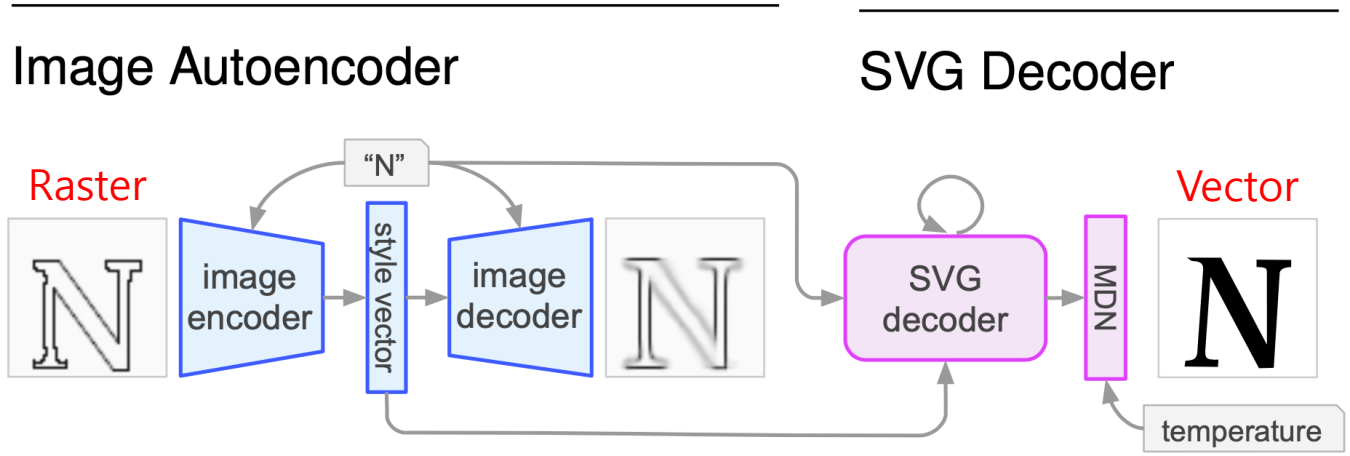
Where to Use?



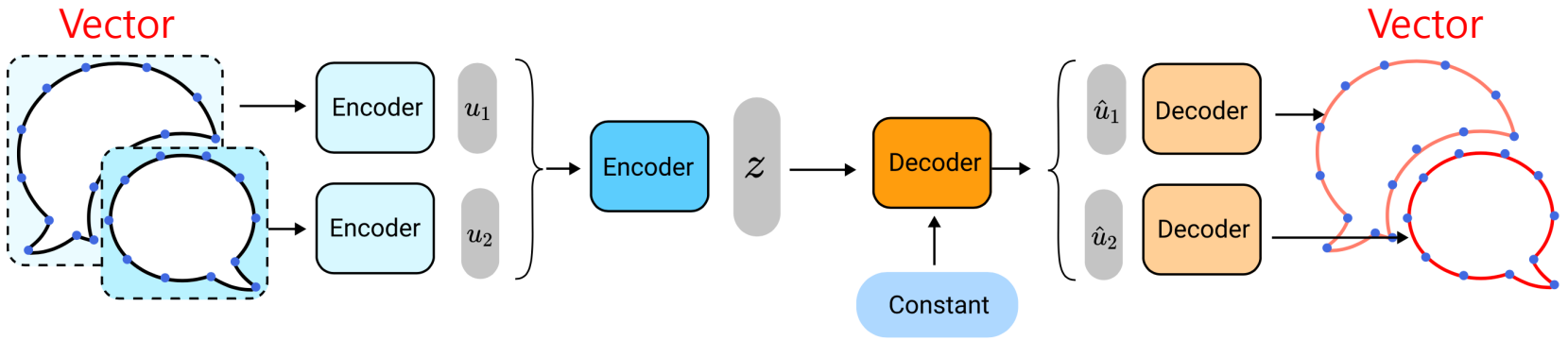
Appendix-A

Vector Supervised Models

SVG-VAE



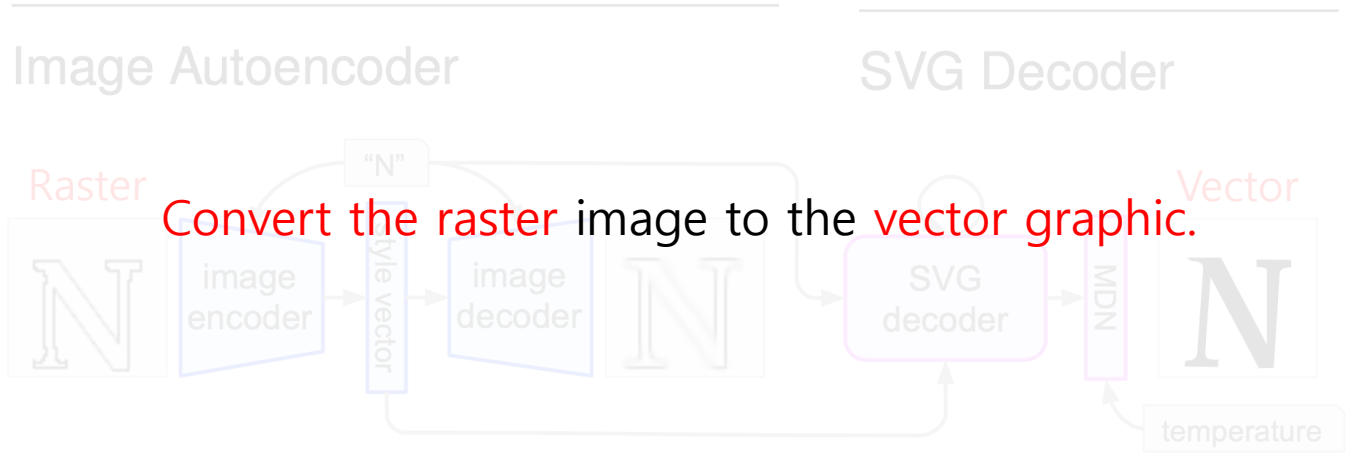
DeepSVG



Lopes, Raphael Gontijo, et al. "A learned representation for scalable vector graphics." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.
Carlier, Alexandre, et al. "Deepsvg: A hierarchical generative network for vector graphics animation." *Advances in Neural Information Processing Systems*. 2020.

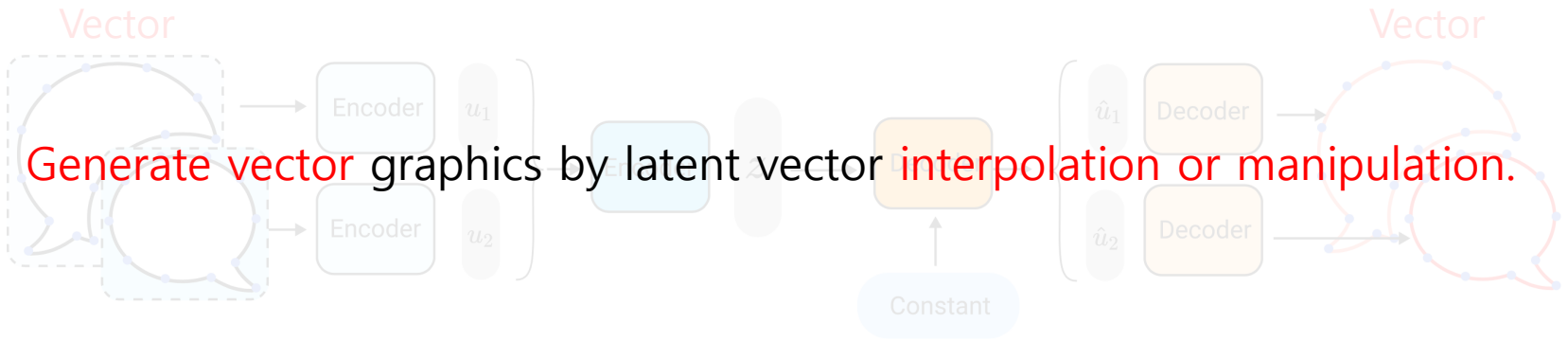
Vector Supervised Models

SVG-VAE



Convert the raster image to the vector graphic.

DeepSVG

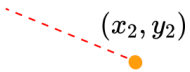
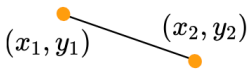
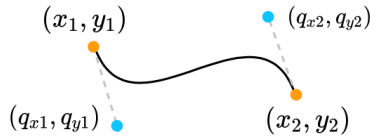
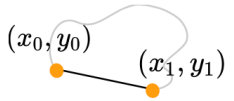


Generate vector graphics by latent vector interpolation or manipulation.

Lopes, Raphael Gontijo, et al. "A learned representation for scalable vector graphics." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.
Carlier, Alexandre, et al. "Deepsvg: A hierarchical generative network for vector graphics animation." *Advances in Neural Information Processing Systems*. 2020.

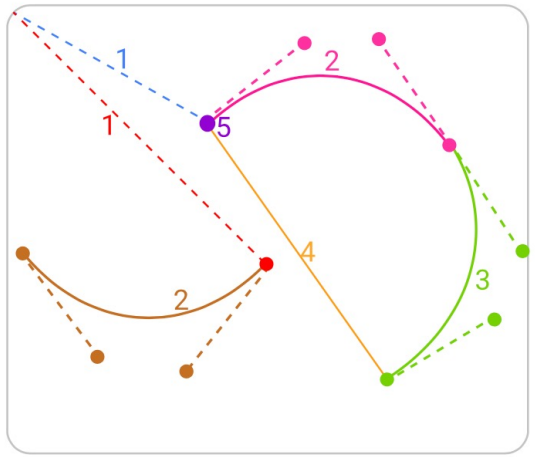
Scalable Vector Graphics (SVG) Representation

Table 1: List of the SVG draw-commands, along with their arguments and a visualization, used in this work. The start-position (x_1, y_1) is implicitly defined as the end-position of the preceding command.

Command	Arguments	Description	Visualization
<SOS>	$\varnothing$	'Start of SVG' token.	
M (MoveTo)	x_2, y_2	Move the cursor to the end-point (x_2, y_2) without drawing anything.	
L (LineTo)	x_2, y_2	Draw a line to the point (x_2, y_2).	
C (Cubic Bézier)	q_{x1}, q_{y1} q_{x2}, q_{y2} x_2, y_2	Draw a cubic Bézier curve with control points (q_{x1}, q_{y1}), (q_{x2}, q_{y2}) and end-point (x_2, y_2).	
Z (ClosePath)	$\varnothing$	Close the path by moving the cursor back to the path's starting position (x_0, y_0).	
<EOS>	$\varnothing$	'End of SVG' token.	

Carlier, Alexandre, et al. "Deepsvg: A hierarchical generative network for vector graphics animation." *Advances in Neural Information Processing Systems*. 2020.

SVG Visualization



P_1

Index	Command	Arguments
0	<SOS>	-1 -1 -1 -1 -1 -1
1	m	-1 -1 -1 -1 95 54
2	c	143 16 180 14 215 66
3	c	255 118 244 152 191 181
4	l	-1 -1 -1 -1 95 54
5	z	-1 -1 -1 -1 -1 -1
6	<EOS>	-1 -1 -1 -1 -1 -1

P_2

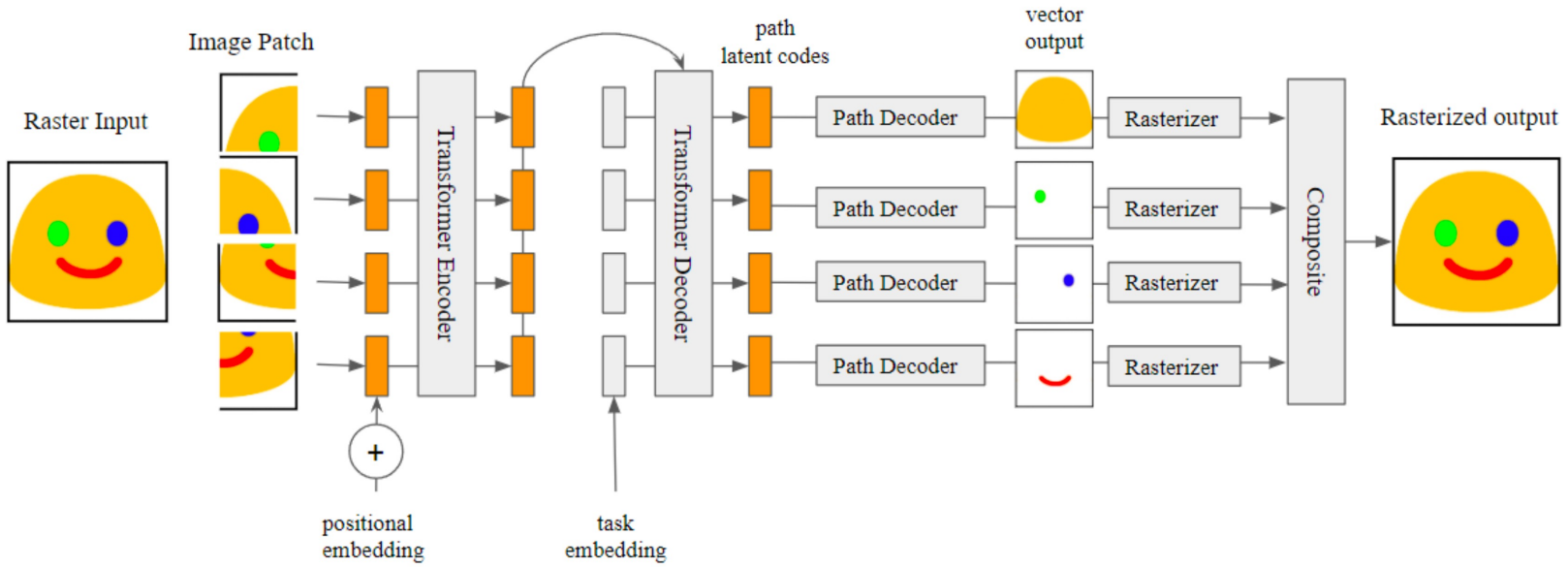
Index	Command	Arguments
0	<SOS>	-1 -1 -1 -1 -1 -1
1	m	-1 -1 -1 -1 132 130
2	c	93 183 48 176 11 125
3	<EOS>	-1 -1 -1 -1 -1 -1
4	<EOS>	-1 -1 -1 -1 -1 -1
5	<EOS>	-1 -1 -1 -1 -1 -1
6	<EOS>	-1 -1 -1 -1 -1 -1

Figure 10: Example of SVG representation. Left: Input SVG image. Right: Corresponding tensor representations with 2 paths and 7 commands ($N_P = 2, N_C = 7$). Commands in the image and the corresponding tensor are color-coded for a better visualization. The arguments are listed in the order q_{x1}, q_{y1}, q_{x2}, q_{y2}, x_2 and y_2. Best viewed in color.

Carlier, Alexandre, et al. "Deepsvg: A hierarchical generative network for vector graphics animation." *Advances in Neural Information Processing Systems*. 2020.

Appendix-B

트랜스포머를 이용한 래스터 이미지 벡터화 자기 지도학습



장점

- CNN 대신 ViT로 latent code 형성 (global feature)

단점

- Latent code의 수 (컴포넌트 수)가 패치 개수에 의존성을 가짐

Appendix-C

Appendix

Official Source Code

- Im2Vec: https://github.com/preddy5/Im2Vec PyTorch
- DiffComp: https://github.com/preddy5/DiffCompositing PyTorch
- DiffVG: https://github.com/BachiLi/diffvg PyTorch

DiffComp

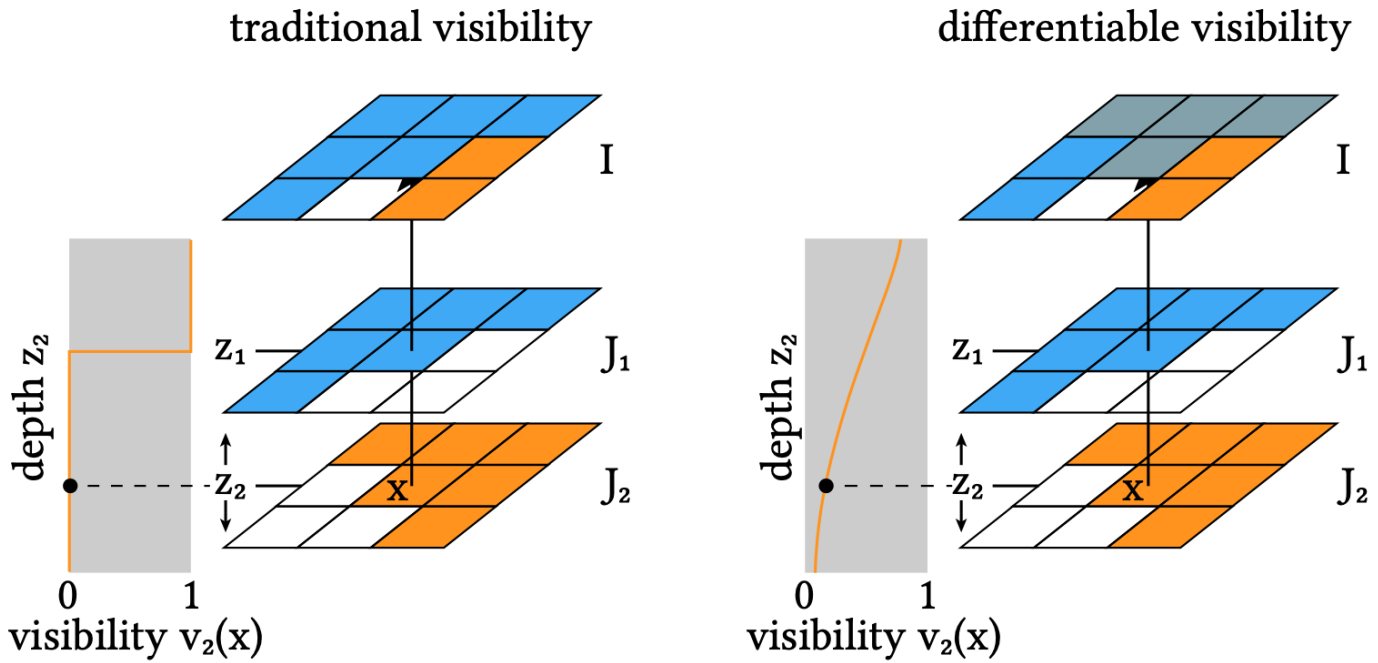


Fig. 5. Differentiable visibility. The traditional visibility function is not differentiable due to C^0 discontinuities at disocclusions. In our differentiable formulation, each layer contributes to the final image with a weight that decreases exponentially with distance from the top-most layer.

Reddy, Pradyumna, et al. "Discovering pattern structure using differentiable compositing." *ACM Transactions on Graphics*. 2020.

DiffVG

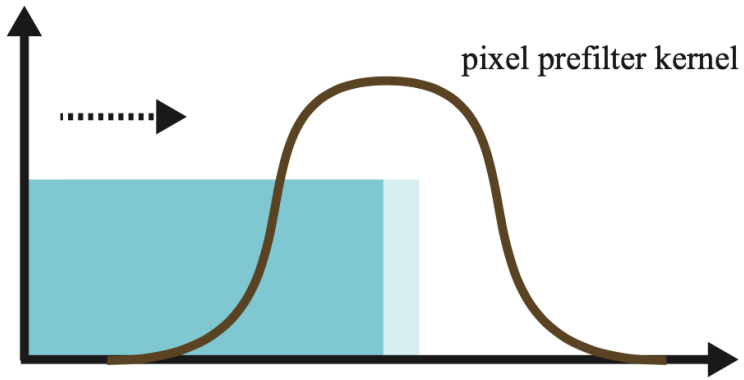


Fig. 2. Pixel prefiltering using a convolution makes a discontinuous function smooth. We illustrate this using a box function in 1D. To rasterize this discontinuous box function with anti-aliasing, we convolve the box with a kernel centered at the pixel center. Moving the box causes a continuous change of the area under the filtering kernel, and therefore, of the anti-aliased signal.

Li, Tzu-Mao, et al. "Differentiable vector graphics rasterization for editing and learning." *ACM Transactions on Graphics*. 2020.