Bézier Spline Simplification Using Locally Integrated Error Metrics

Siqi Wang
New York University

Chenxi Liu
University of British Columbia

Daniele Panozzo
New York University

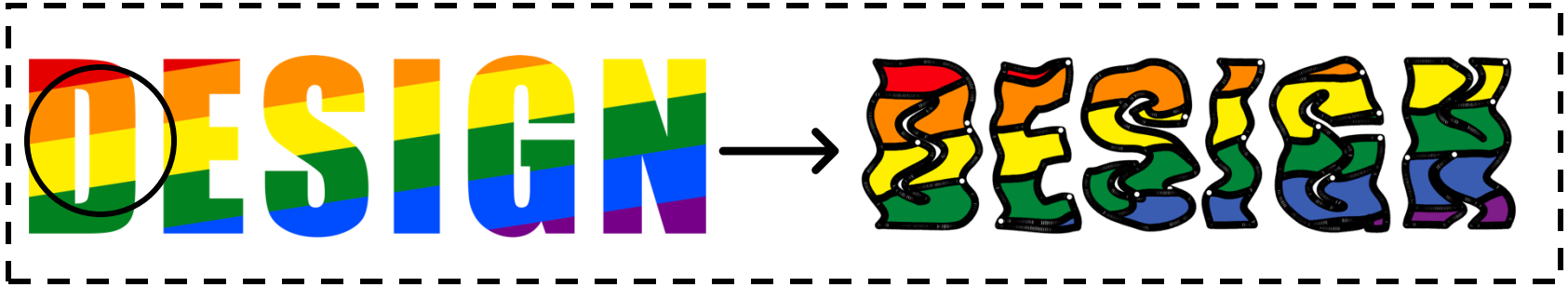
Denis Zorin
New York University

Alec Jacobson
University of Toronto
Adobe Research

SIGGRAPH ASIA 2023
SYDNEY | 12 – 15 DEC

Background

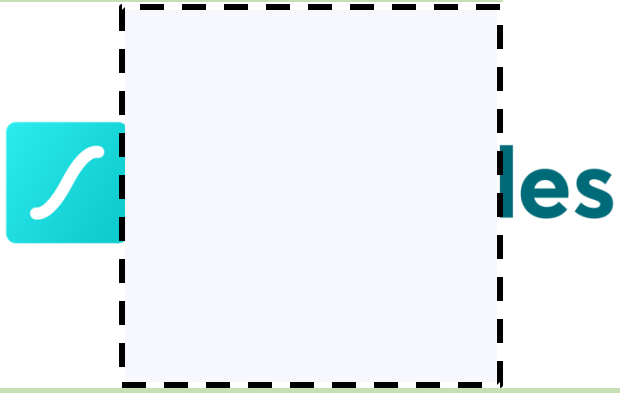
Vector graphics editing tool



Background

Vector graphics editing tool

Vector graphics animation format

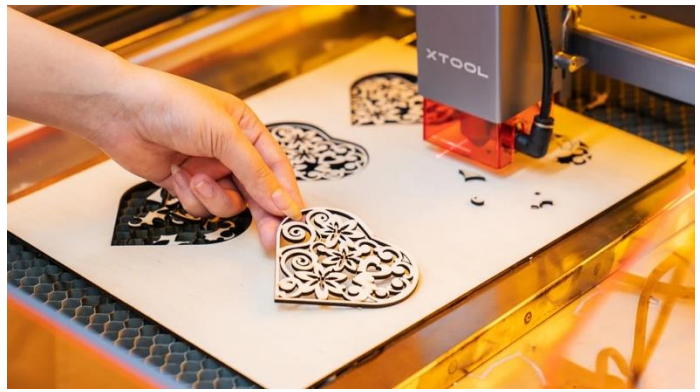


Background

Vector graphics editing tool

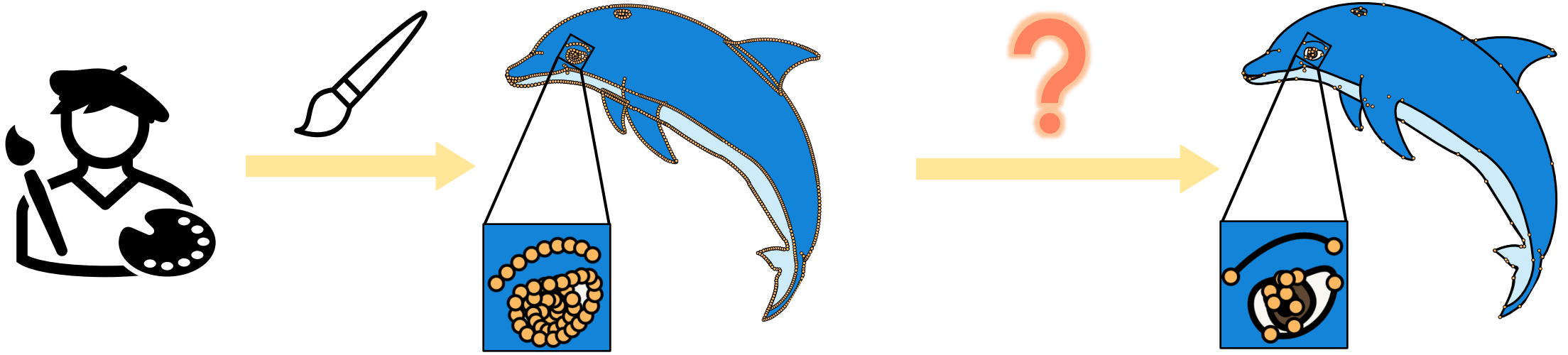
Vector graphics animation format

CNC machines



Problem

Simplifying ASAP while remaining *exceptionally accurate* to *preserve* artist's intention



Dense SVG

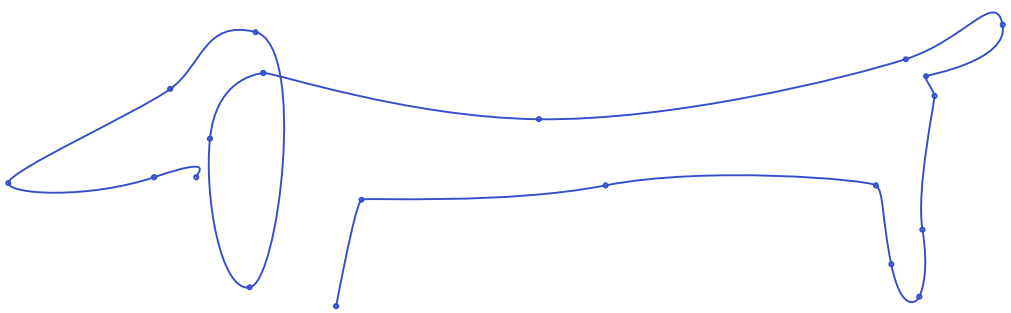
More splines (control points)

Simplified SVG

Fewer splines (control points)

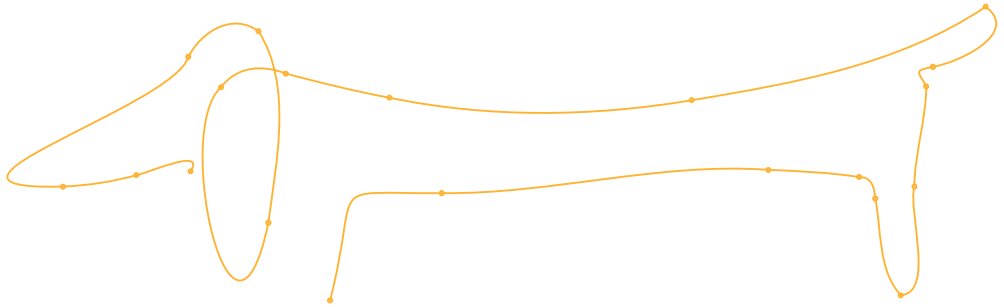
Lossless simplification?

State-of-the-art way of editing curves…



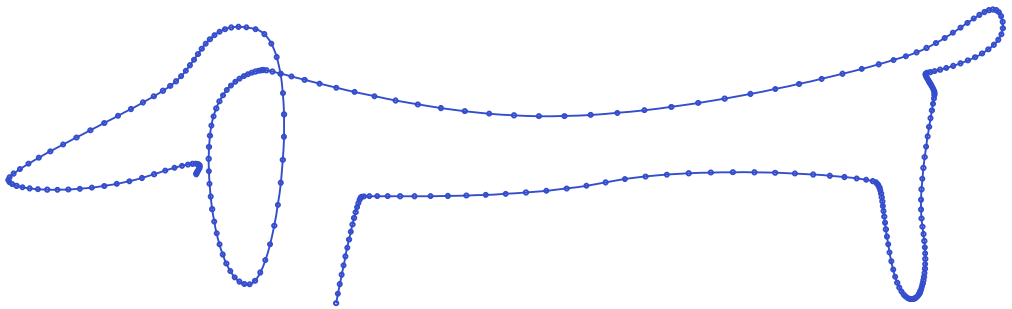
#segments = 19

≠



#segments = 19

Up-sample in place

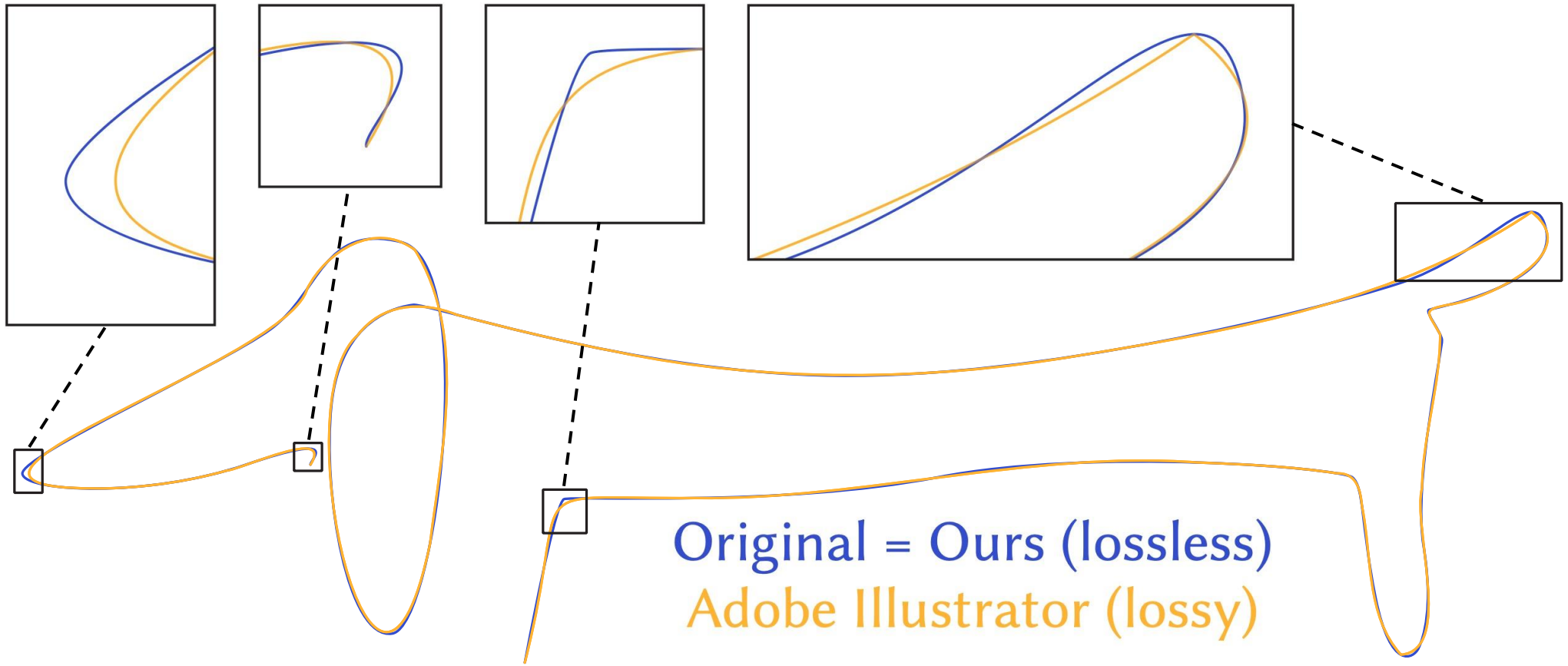


#segments = 304

Simplify by Ai

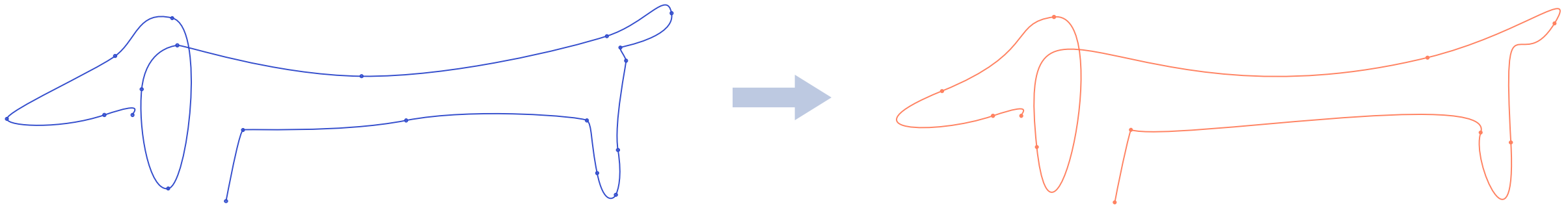
Lossless simplification?

Existing literature and commercial software *fail* this simple test…



Lossy simplification

Once lossless removals are exhausted, subsequent lossy simplification starts to introduce error…



Lossless
#segments = 19

Lossy
#segments = 10

Related work

Sampling and refitting

- **Introduce error while sampling**
- **Consistent endpoint tangent not guaranteed**

[Schneider 1990]

Related work

Sampling and refitting

[Schneider 1990]

Top-down algorithm by subdividing and fitting

[Levien 2009]

Adobe Illustrator

Related work

Sampling and refitting

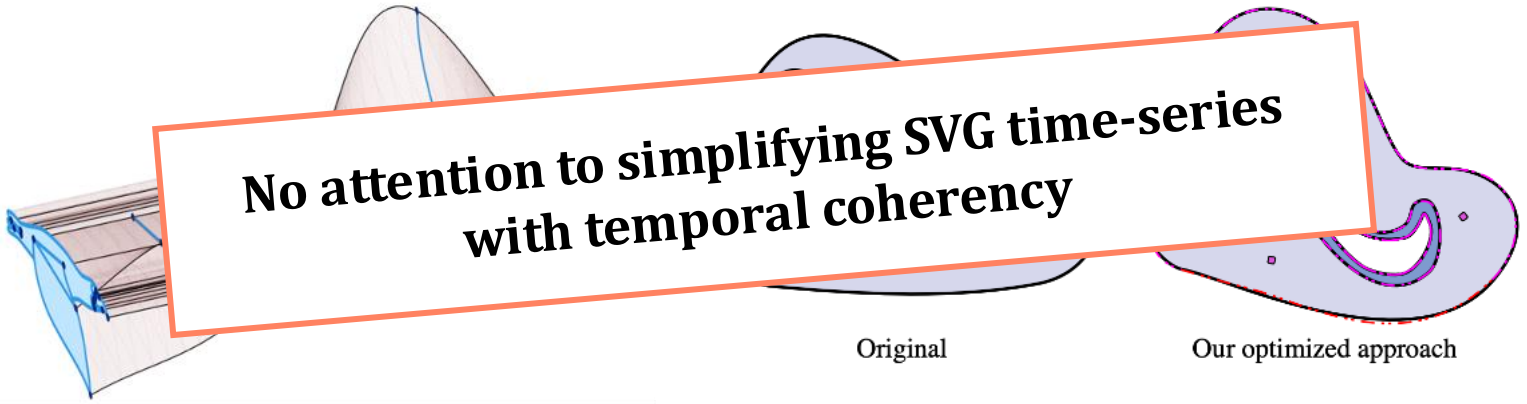
[Schneider 1990]

Top-down algorithm by subdividing and fitting

[Levien 2009]

Adobe Illustrator

Vector graphics animation



No attention to simplifying SVG time-series with temporal coherency

[Dalstein et al. 2015]

[Liu et al. 2014]

Cartoon Animator 5

Contributions

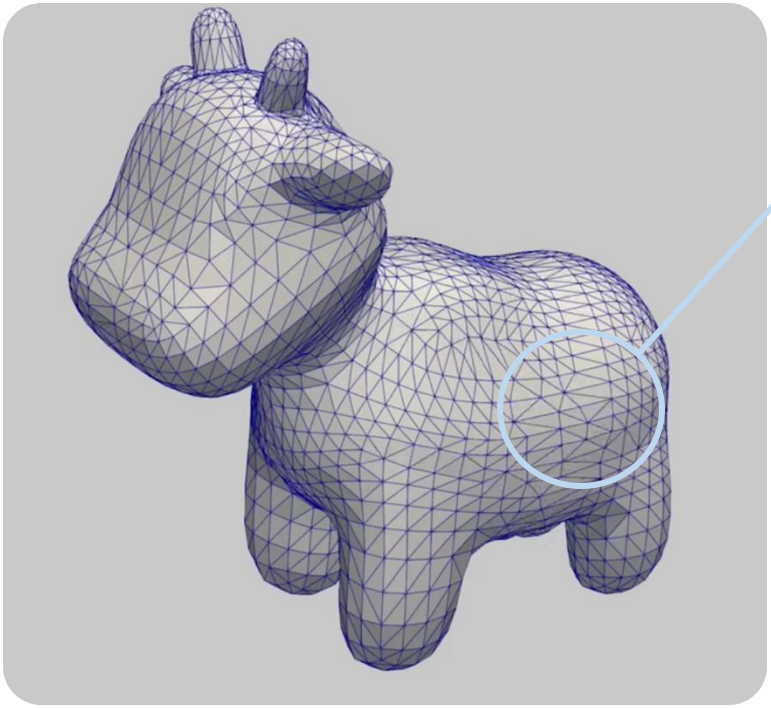
01 Recover *lossless* simplification

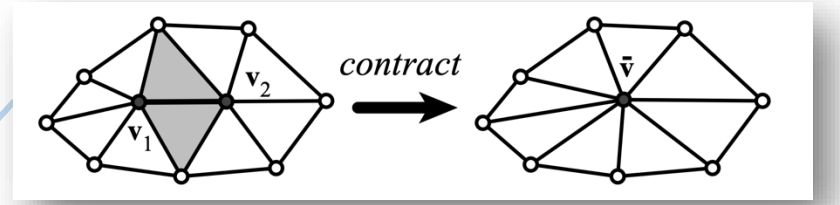
02 Improve *lossy* simplification

03 Conduct a large-scale benchmark comparison

04 Simplify vector graphics time-series with temporal coherency

Inspired by mesh simplification…

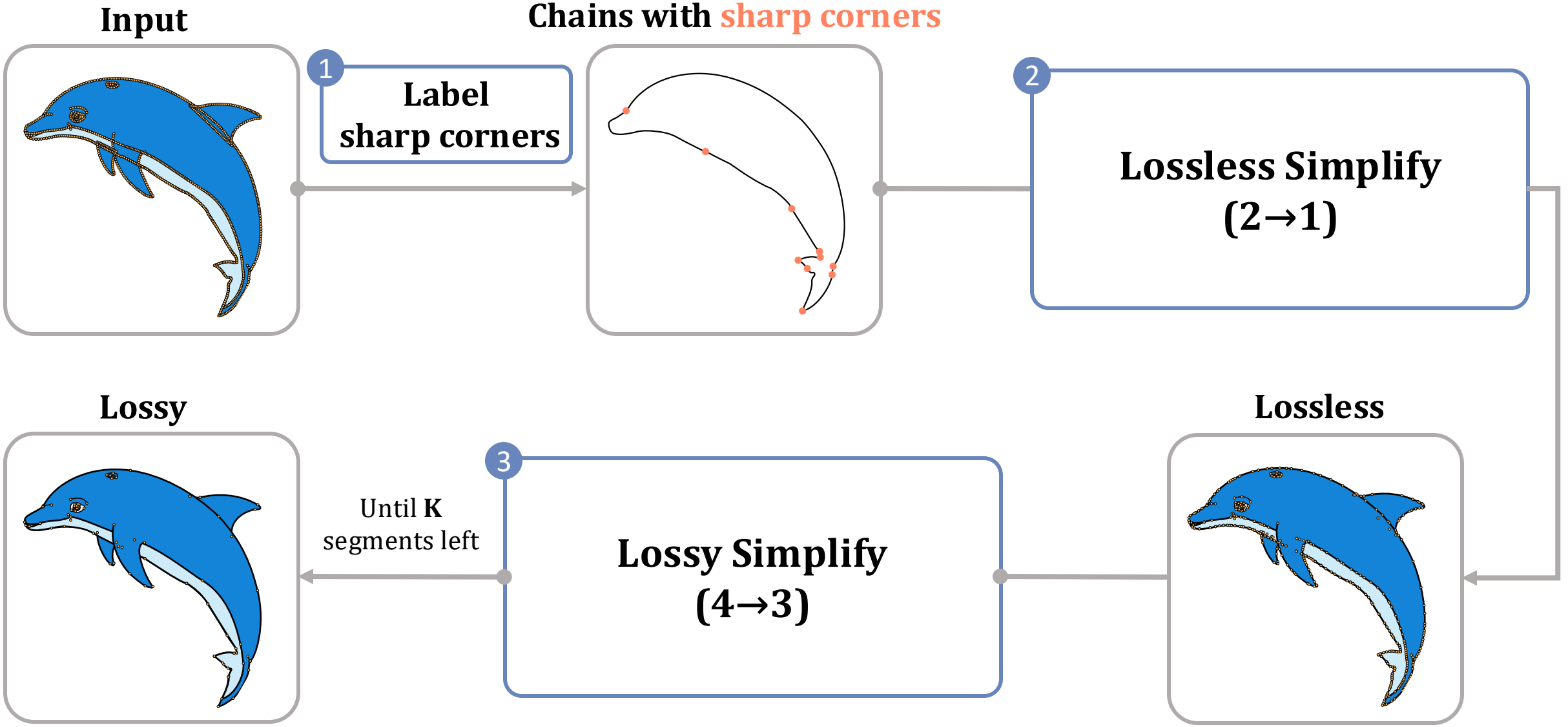




1. Simplify with local operations
2. Process in a greedy manner

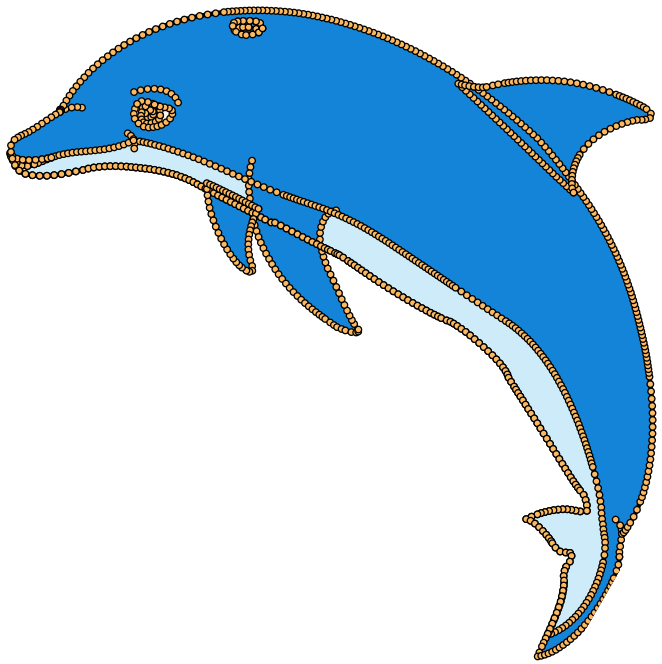
Surface Simplification using Quadric Error Metrics (QEM)
[Garland and Heckbert 1997]

Method


Input
1
Label
sharp corners
Chains with sharp corners
2
Lossless Simplify
(2→1)
Lossless
3
Lossy Simplify
(4→3)
Until K
segments left
Lossy

Label sharp corners

Input

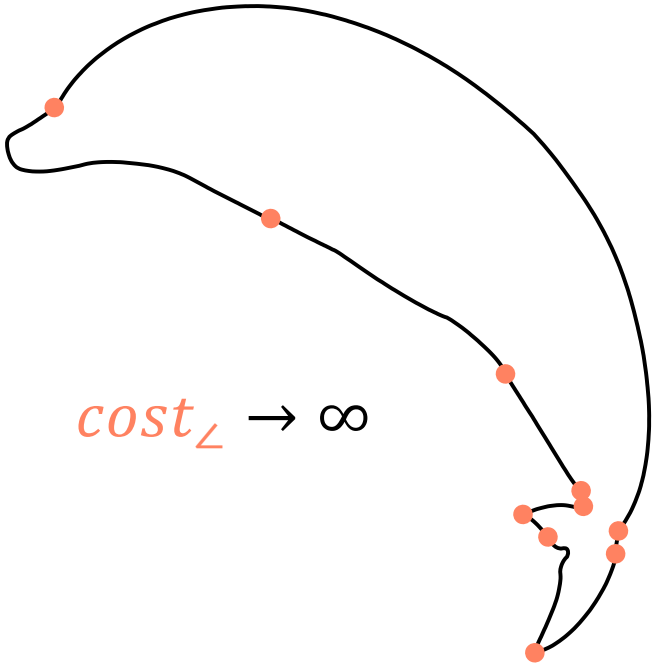


1

Label sharp corners

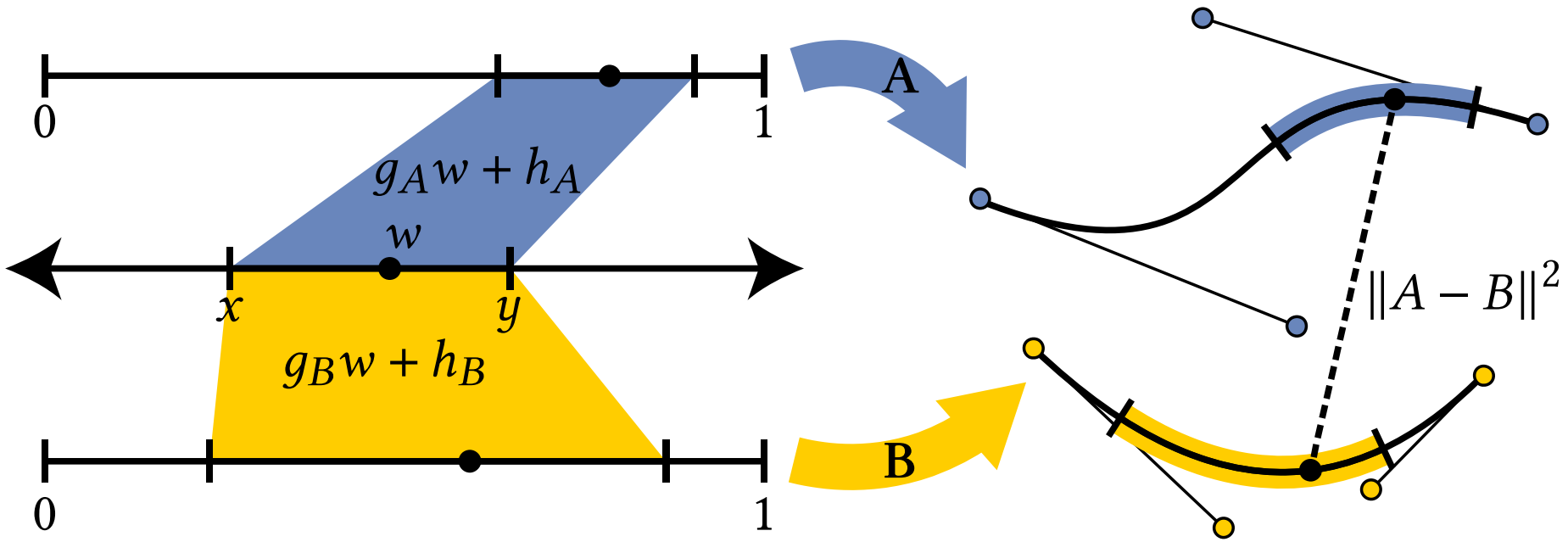
With tangent direction change
$\theta > \theta_0$

Chains with sharp corners



Distance between two segments

- **Segment**: a single cubic Bézier curve

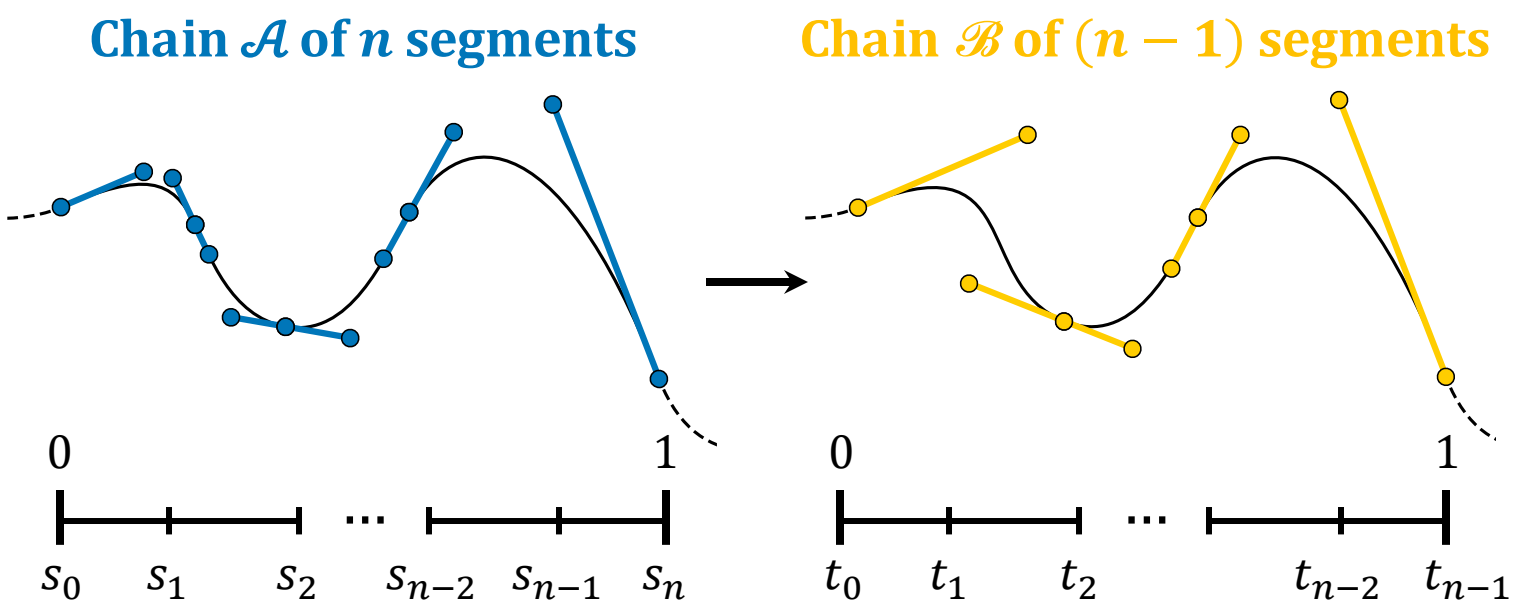


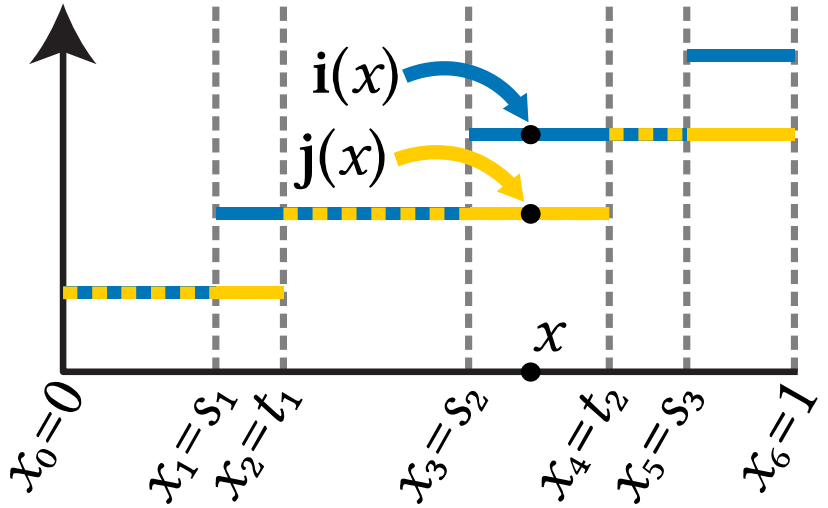
- Integrated distance over an arbitrary parametric domain:

$$\mathcal{E}(x, y) = \int_x^y \|A(g_A w + h_A) - B(g_B w + h_B)\|^2 dw$$

Distance between two chains

- **Chain**: G^1 continuous sequence of one or more segments

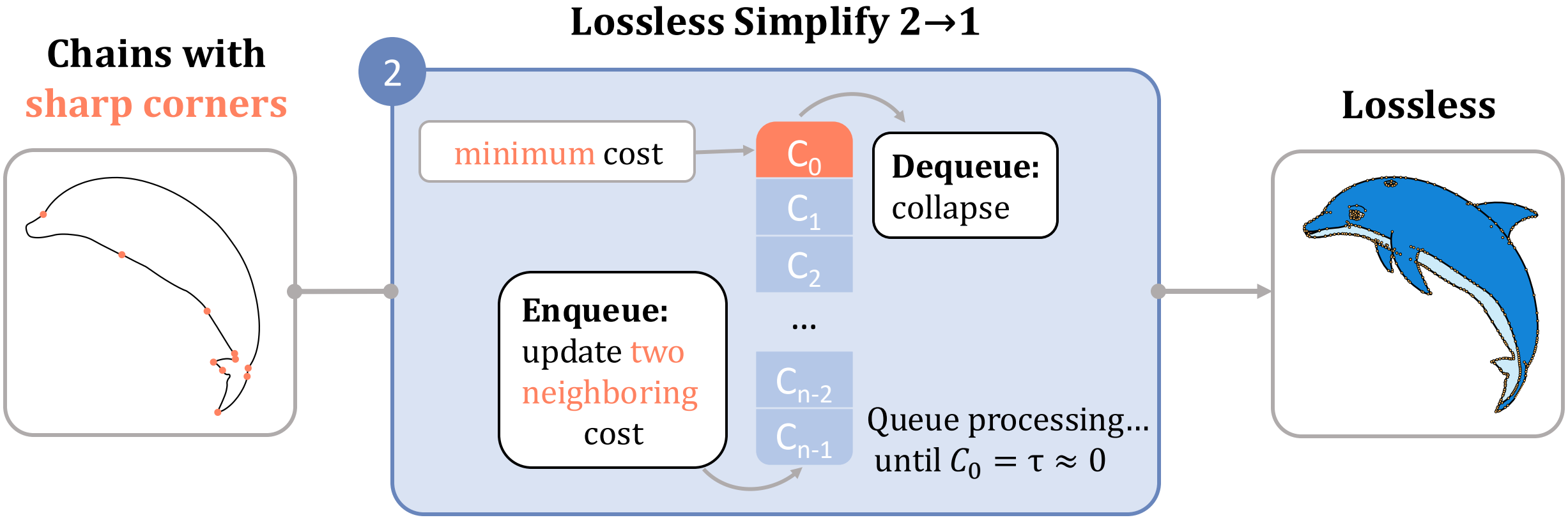




MIN $E(\boldsymbol{P}, \{s_i\}, \boldsymbol{Q}, \{t_j\}) = \sum_{k=1}^{2n-2} \omega_k \int_{x_{k-1}}^{x_k} \left\| dist(\boldsymbol{A}^{seg\_i(x)} - \boldsymbol{B}^{seg\_j(x)}) \right\|^2 dx$

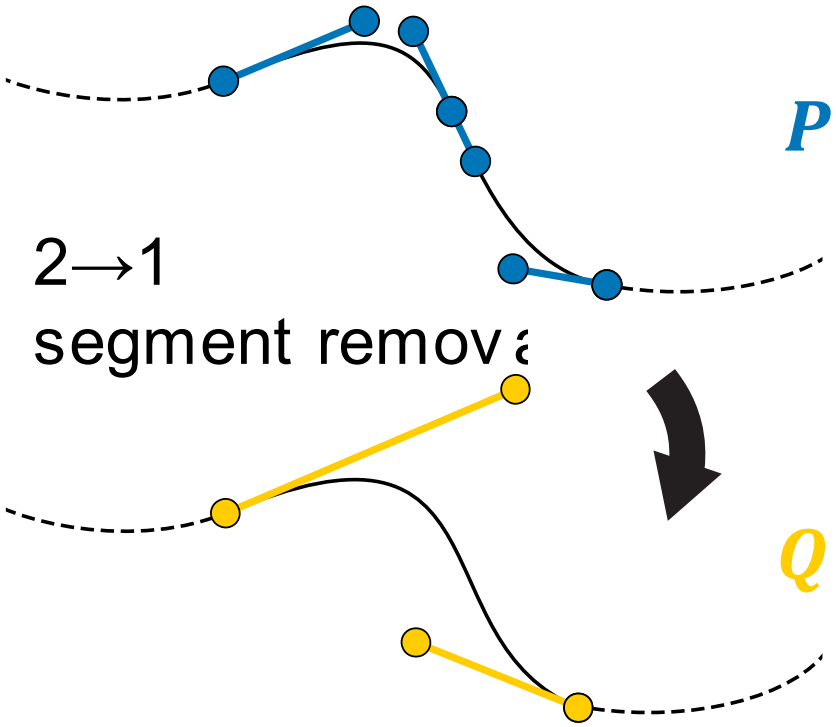
$$\omega_k = \frac{1}{s_i - s_{i-1}} + \frac{1}{t_j - t_{j-1}}$$

Greedy algorithm (2→1)



Why 2→1 lossless?

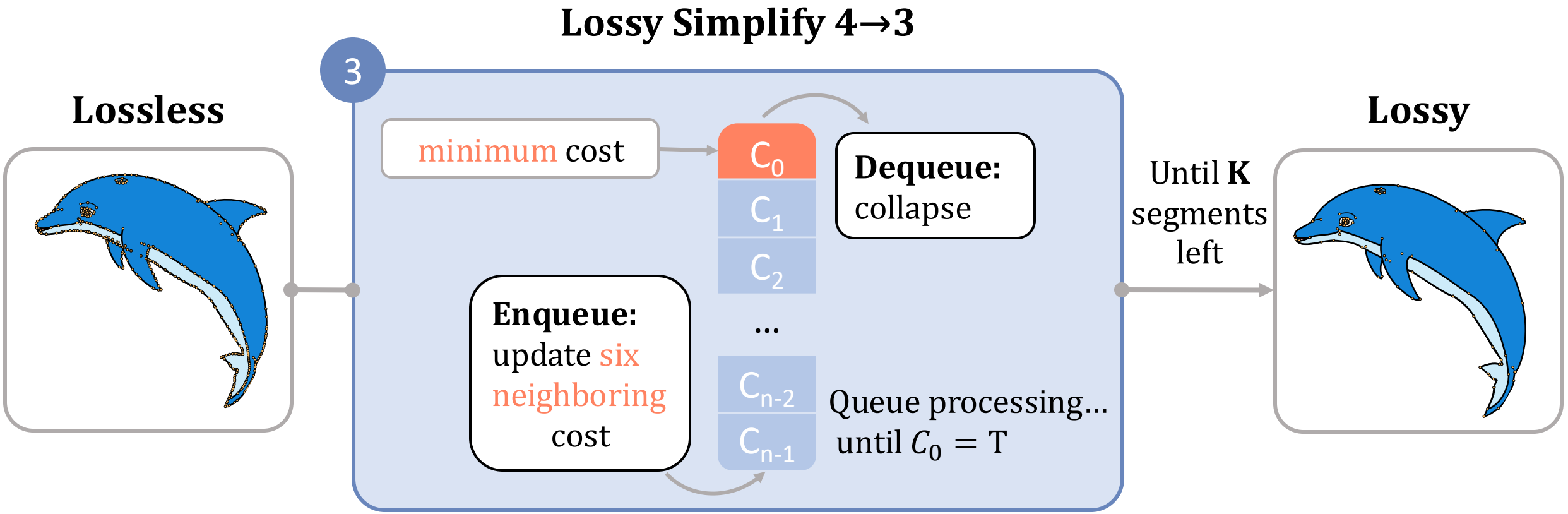
$$cost = \min_{\{s_i\},\boldsymbol{Q},\{t_j\}} E\left(\boldsymbol{P},\{s_i\},\boldsymbol{Q},\{t_j\}\right)$$



Exact solution is just a root finding!

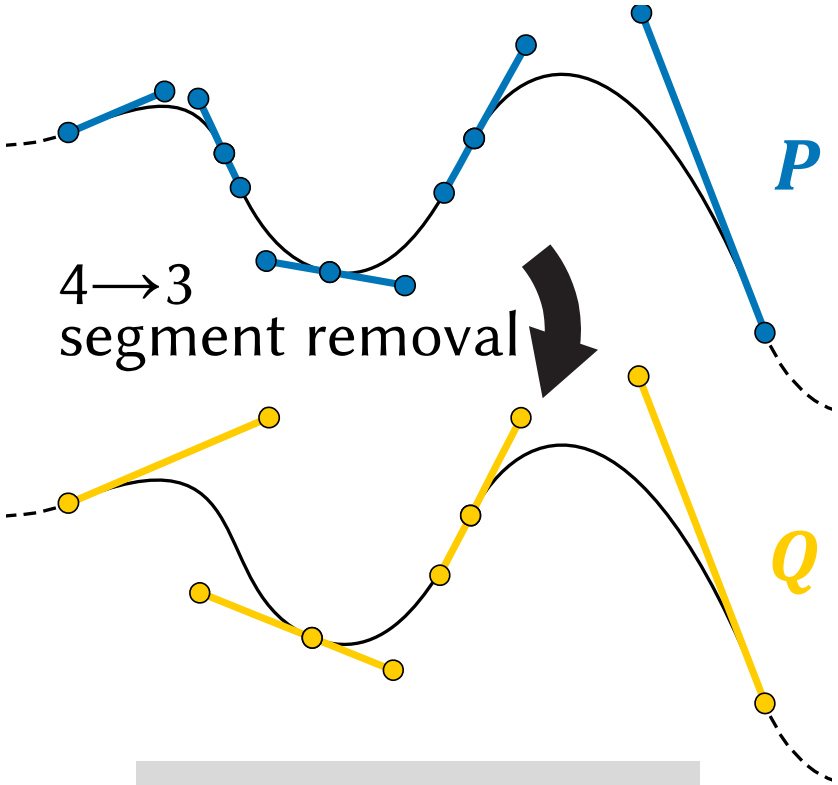
$$-(1+r)t^3 + 3rt^2 - 3rt + r = 0$$

Greedy algorithm (4→3)



4→3 segment removal operation

$$cost = \min_{\{s_i\},\boldsymbol{Q},\{t_j\}} E\left(\boldsymbol{P},\{s_i\},\boldsymbol{Q},\{t_j\}\right)$$



G^1 & C^0continuity

G^1 ratio and relative parameterization ***constant***

Quadratic Form

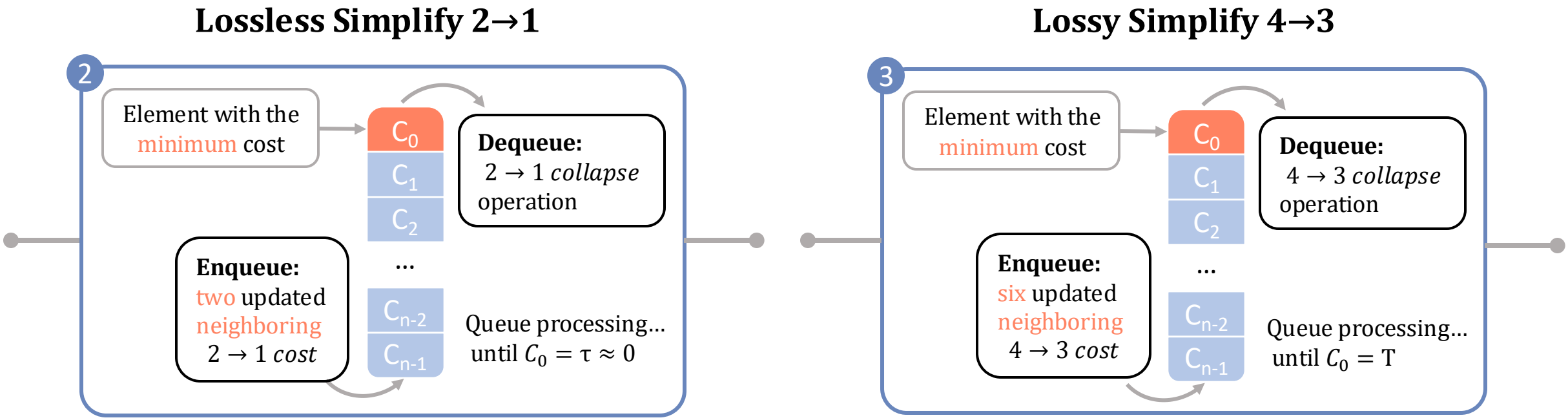
Linear solve

Relax G^1 ratio and relative parameterization

Non-linear optimization

Gauss-Newton method

Time complexity



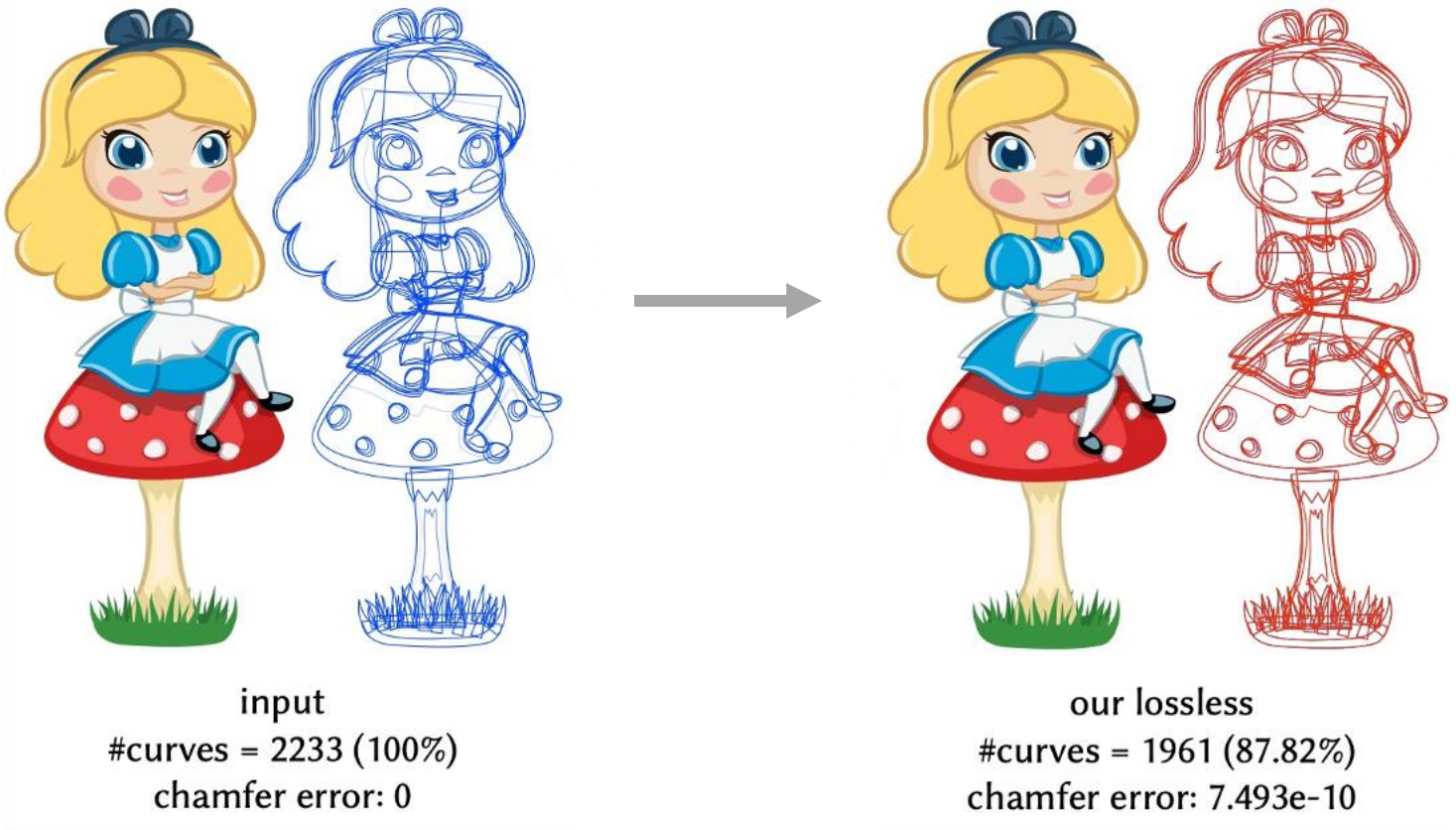
- All local operations have $O(1)$ complexity.
- Total processing complexity is $O(NlogN)$.

Mesh decimation

RESULT

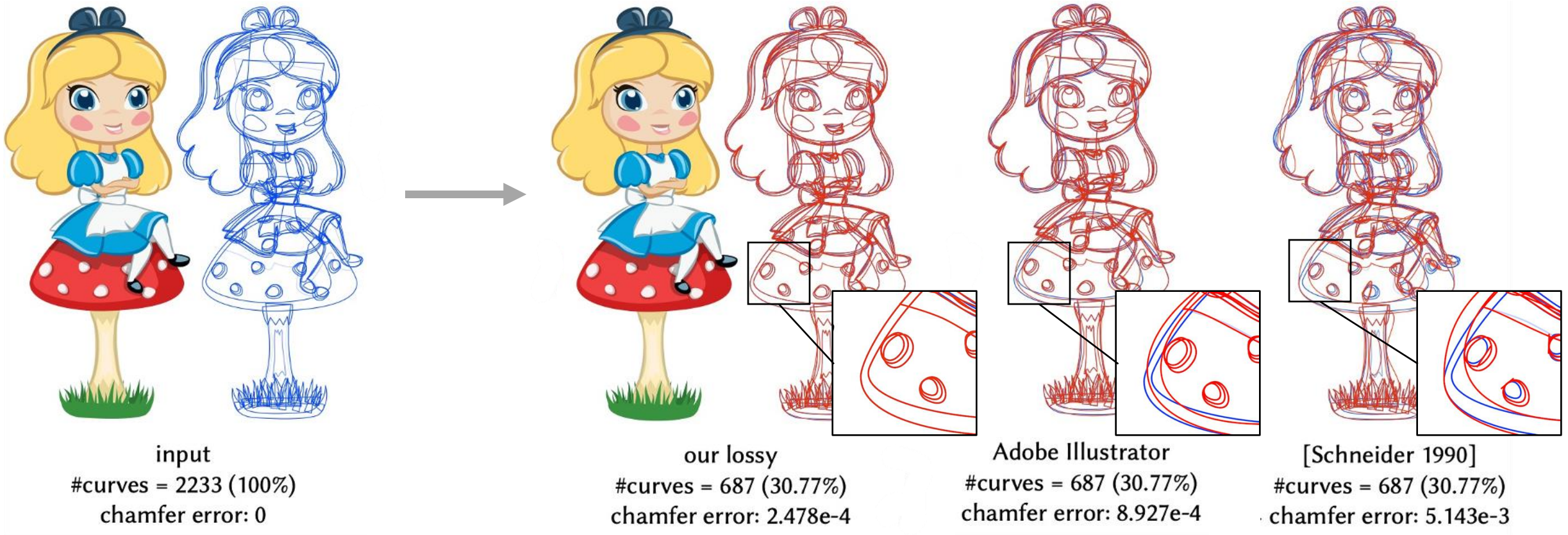
Qualitative evaluation

- Lossless simplification removes ~13% redundancy.



Qualitative evaluation

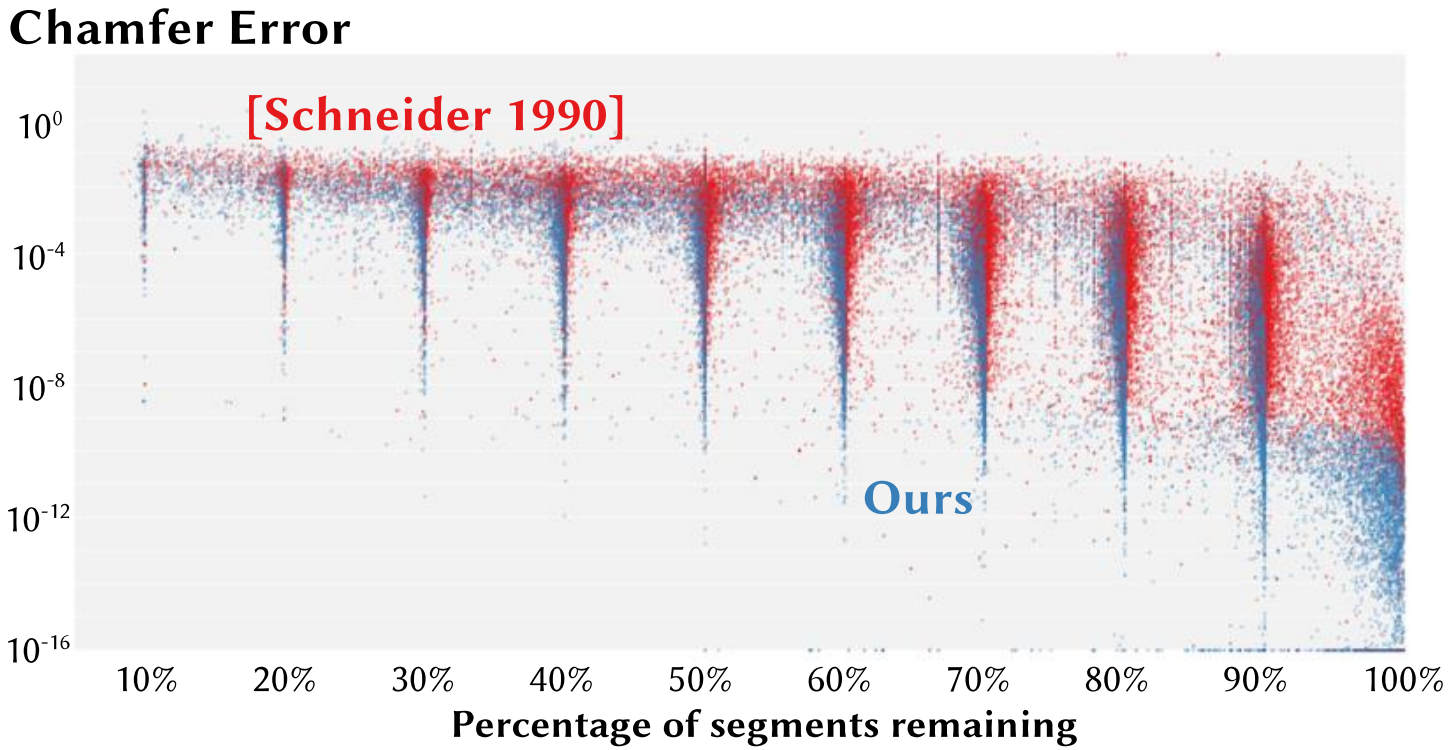
- Our lossy simplification outperforms Adobe Illustrator and Inkscape [Schneider 1990].



Quantitative evaluation

openclipart

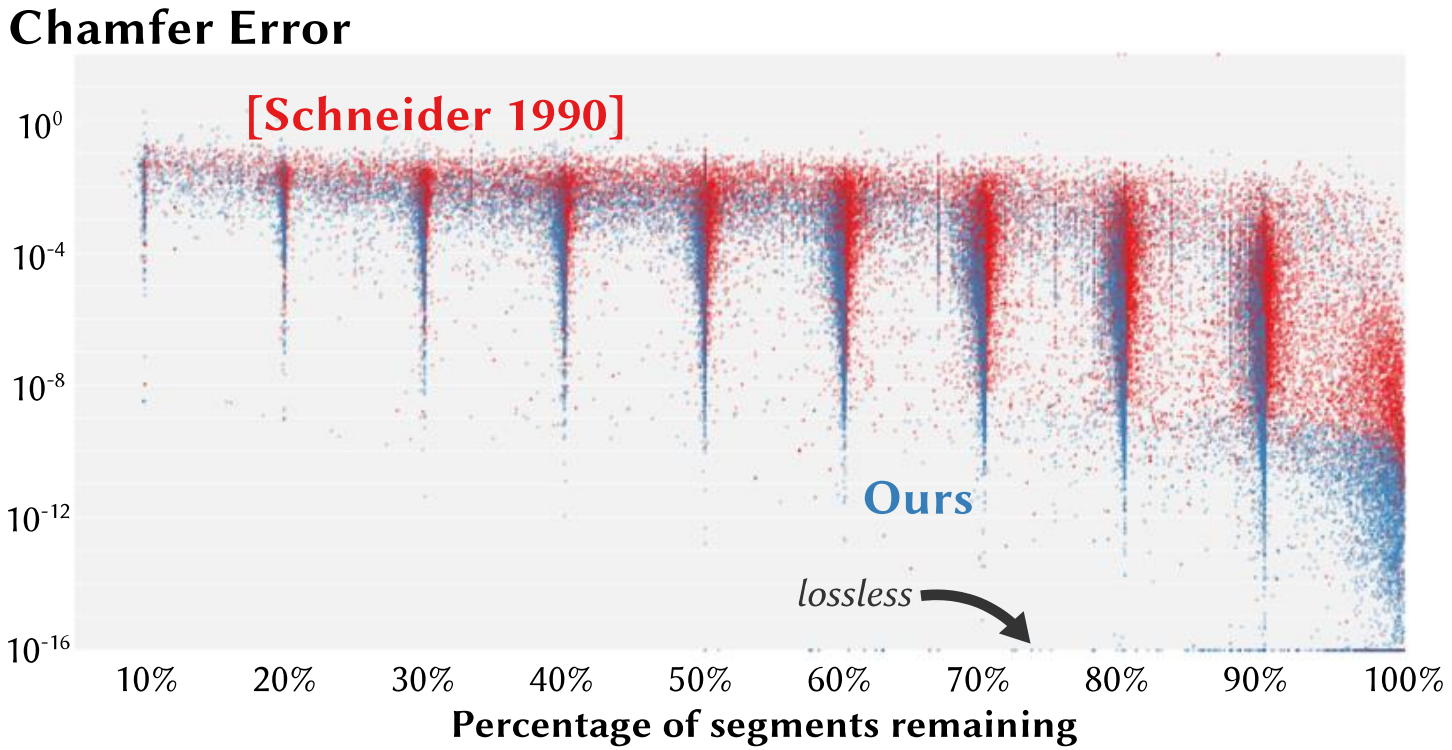
Large-scale benchmark of vector graphics simplification on **20K OpenClipArts .svg files**:



Quantitative evaluation

Large-scale benchmark of vector graphics simplification on **20K OpenClipArts .svg files**:

- Around 75% SVGs in the wild have at least some *fully-redundant* curves!

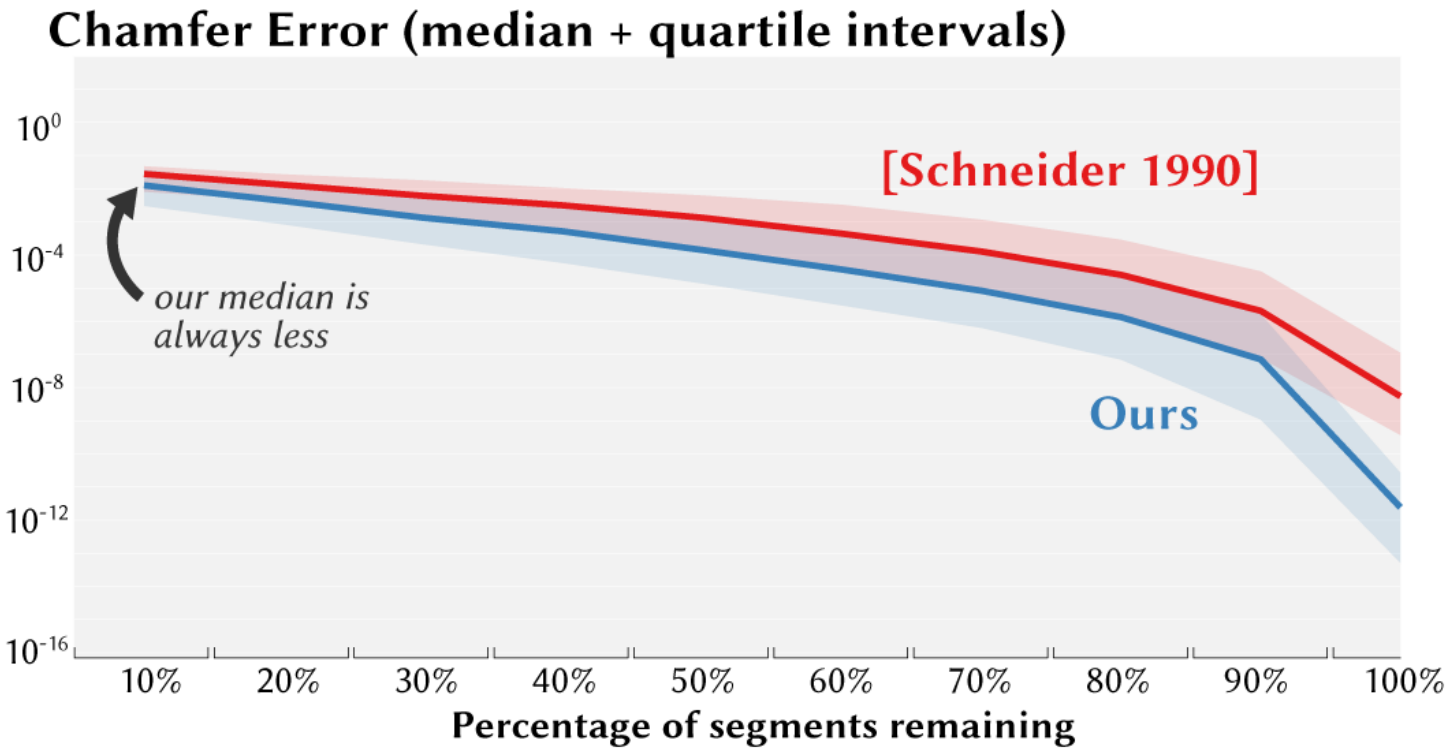


Quantitative evaluation

openclipart

Large-scale benchmark of vector graphics simplification on **20K OpenClipArts .svg files**:

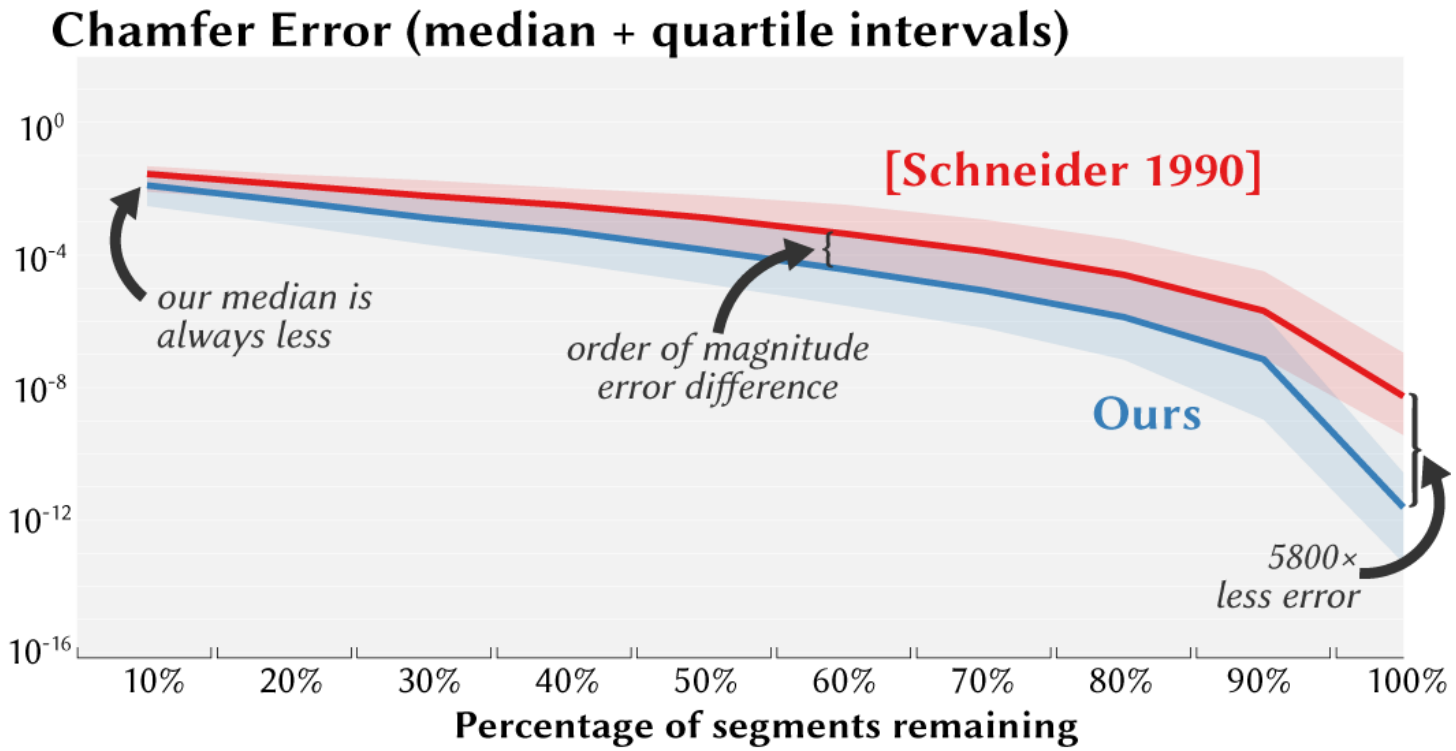
- Our method's median is always *smaller*.



Quantitative evaluation

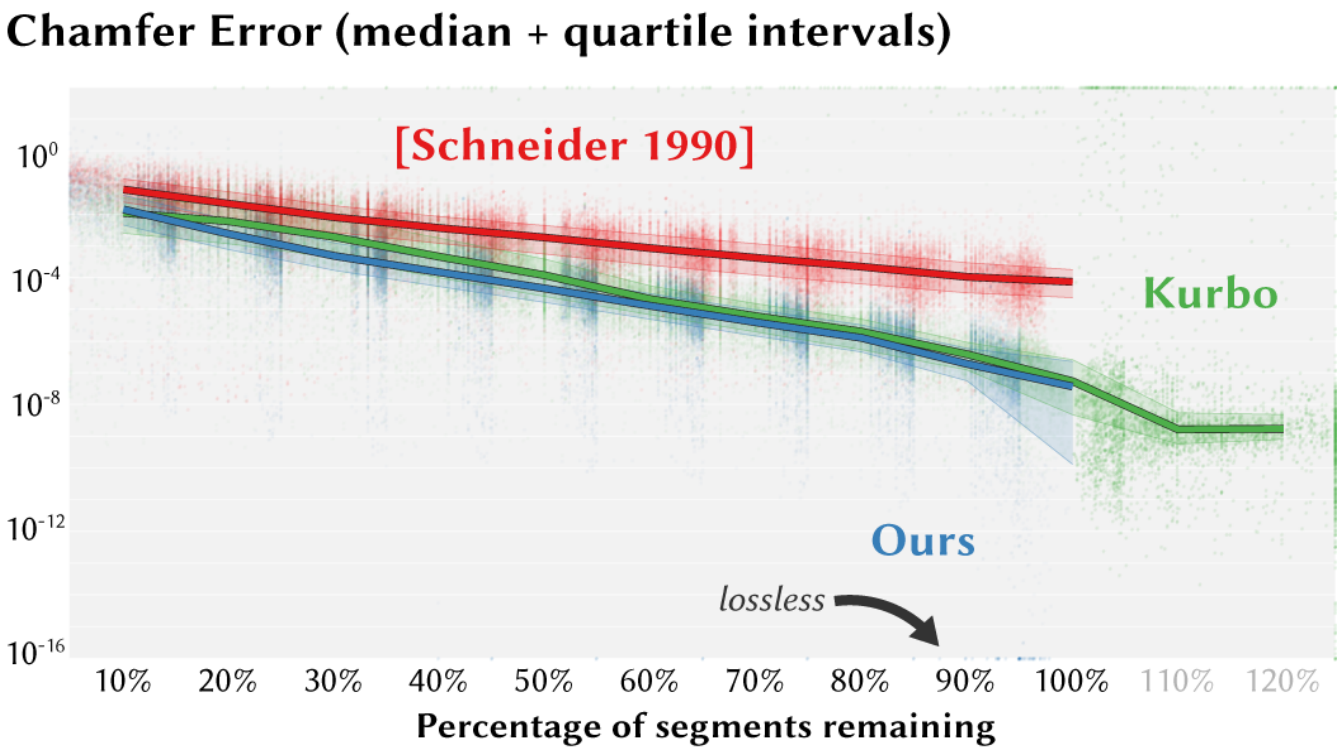
Large-scale benchmark of vector graphics simplification on **20K OpenClipArts .svg files**:

- Our method's median is always *smaller*.
- *Order of magnitude* improvement and best improvement for *larger percentages*.



Quantitative evaluation

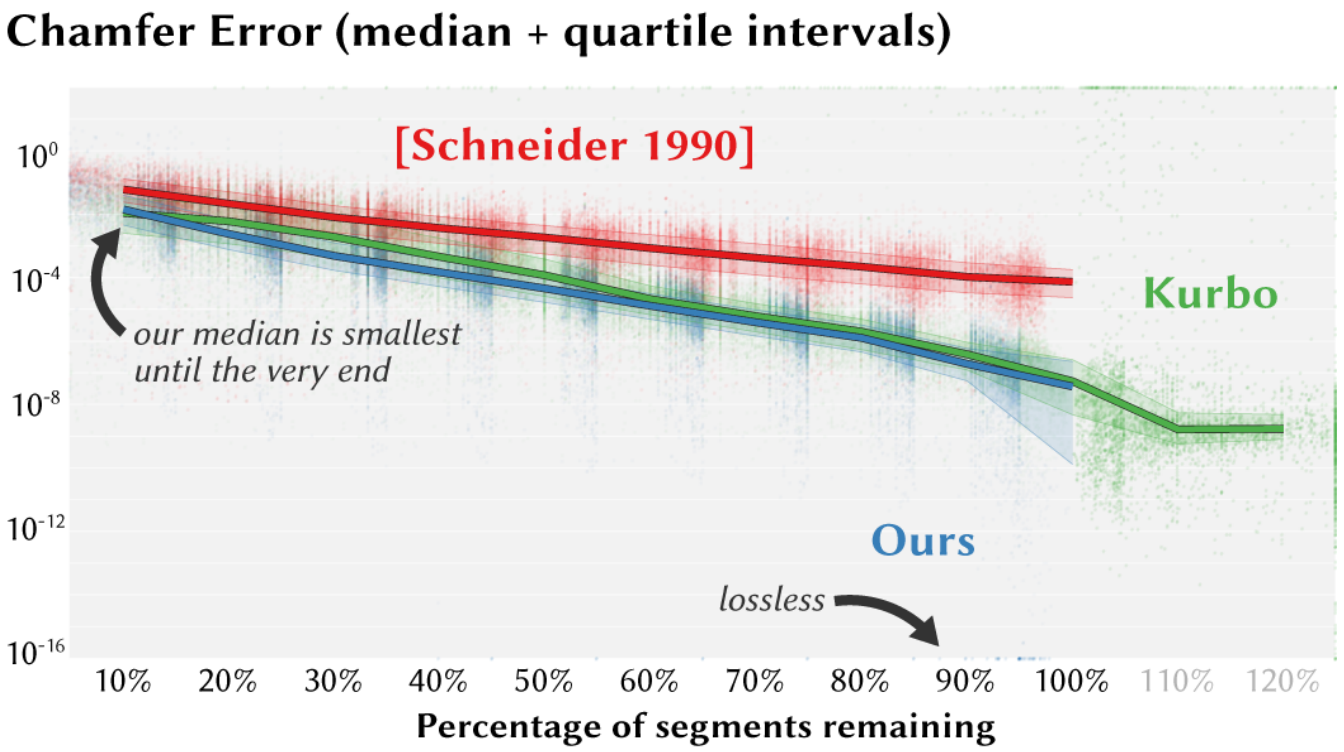
5000 randomly-sampled smooth chains from the dataset of length $N \geq 20$:



Quantitative evaluation

5000 randomly-sampled smooth chains from the dataset of length $N \geq 20$:

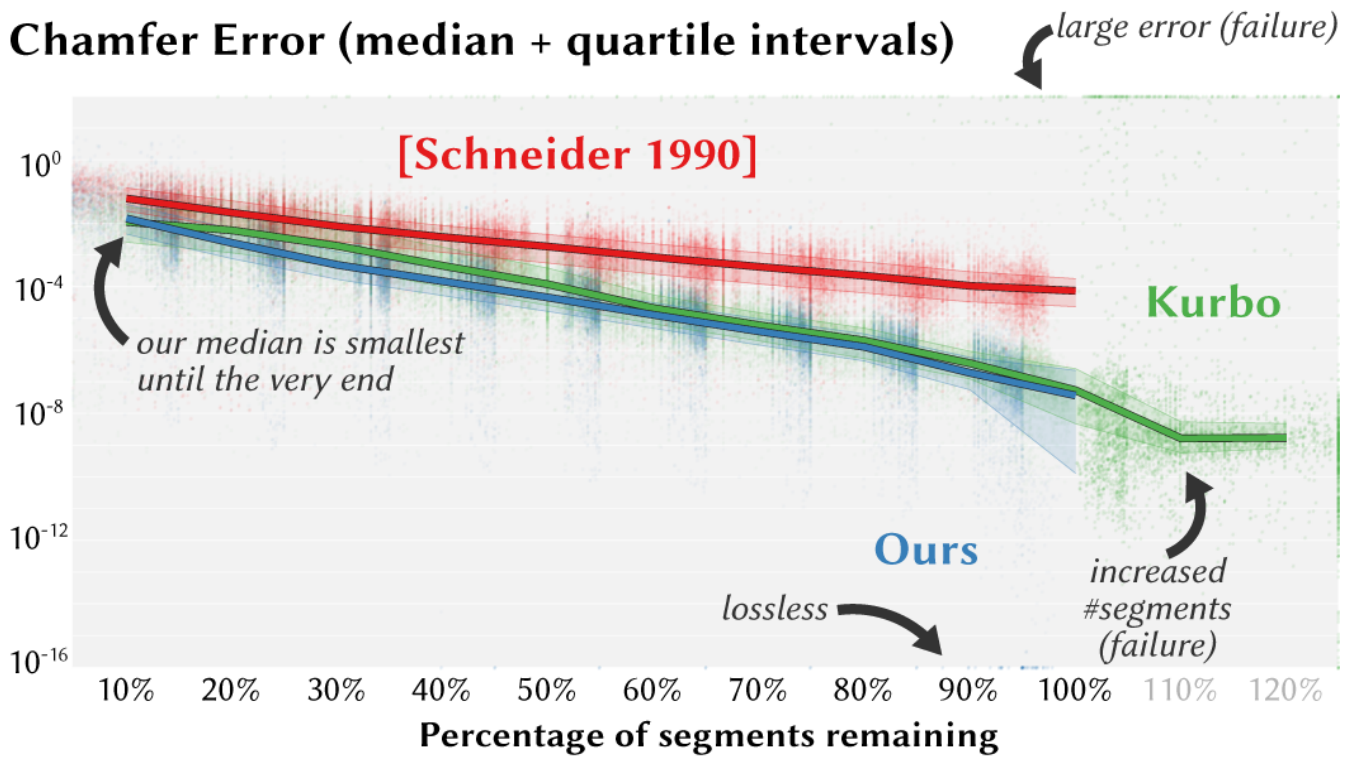
- Our median is the *smallest* until the very end.



Quantitative evaluation

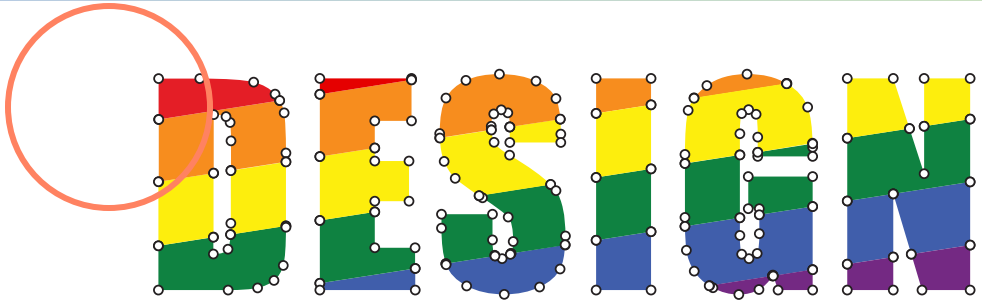
5000 randomly-sampled smooth chains from the dataset of length $N \geq 20$:

- Our median is the *smallest* until the very end.
- Kurbo [Levien 2009] fails ~34% of the time.



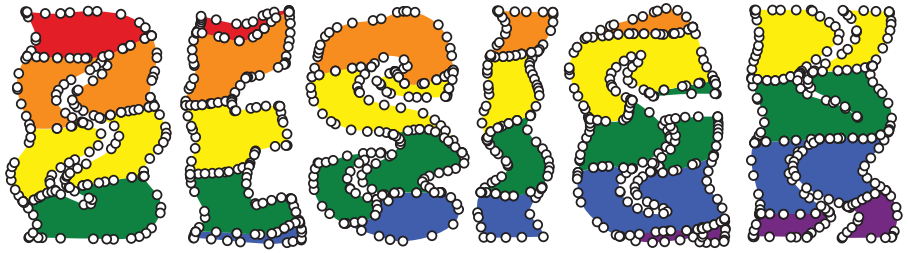
APPLICATIONS

Integrate with brushwork

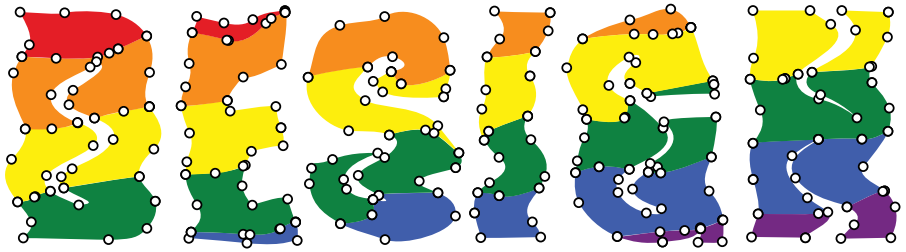


original
#segments: 231

densely up-sampled DESIGN + WARP Brushwork
#segments: 7224



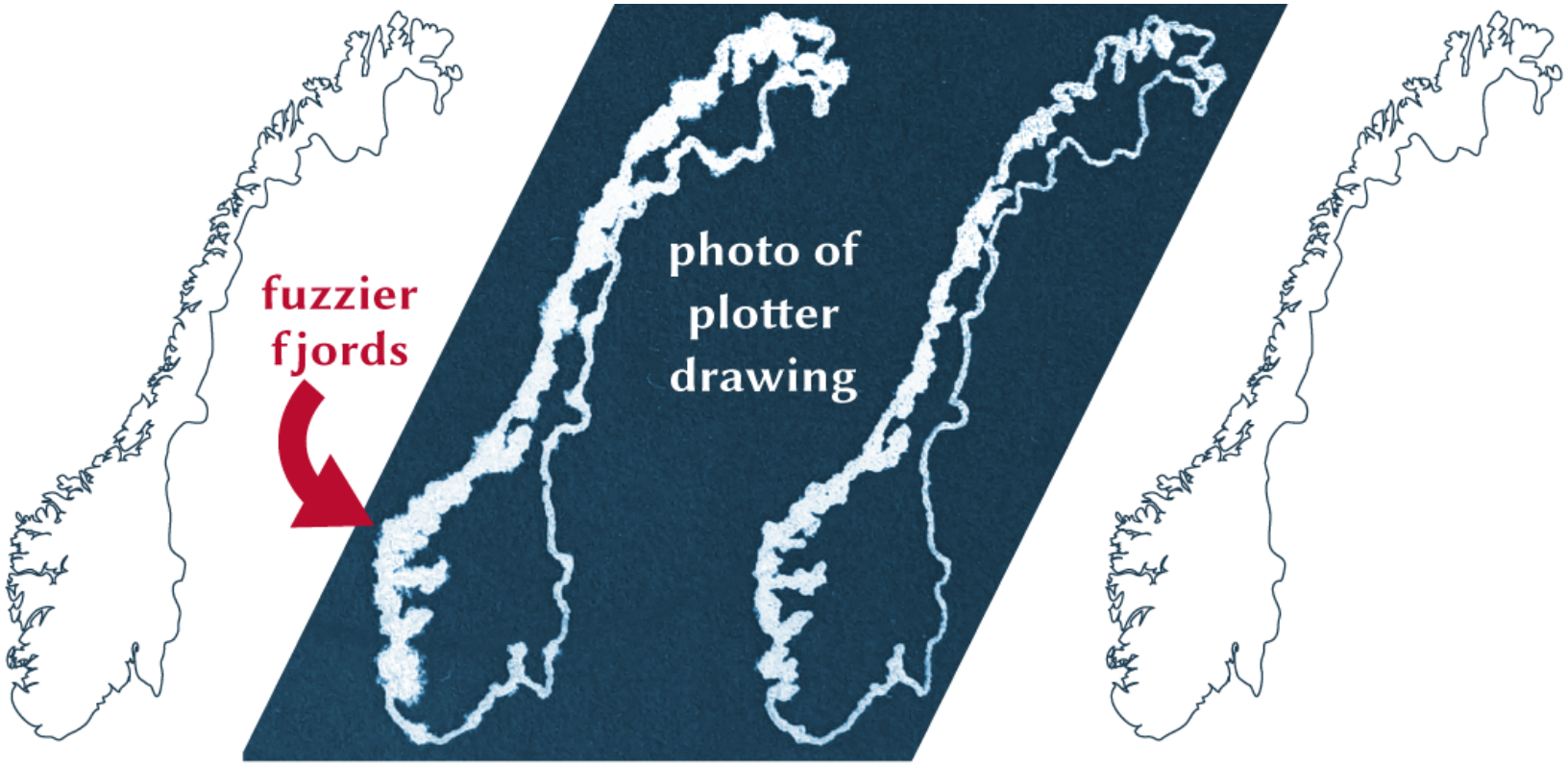
our lossless
#segments: 1164



our lossy
#segments: 300

Plotter & Laser cutter

Simplified result produces less ink bleeding



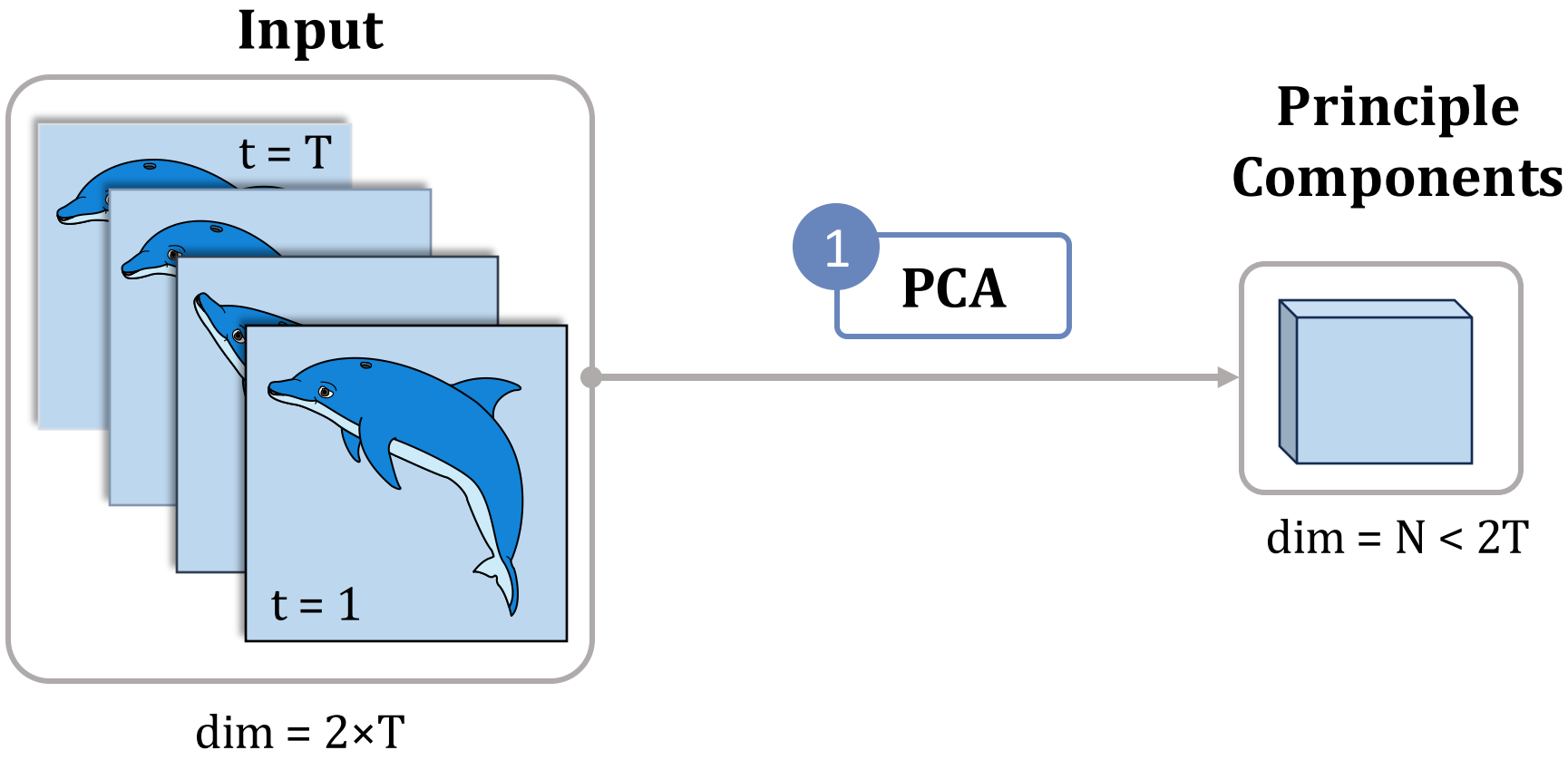
“3D” curve

Simplify “3D” curve by treating the stroke width as an extra dimension

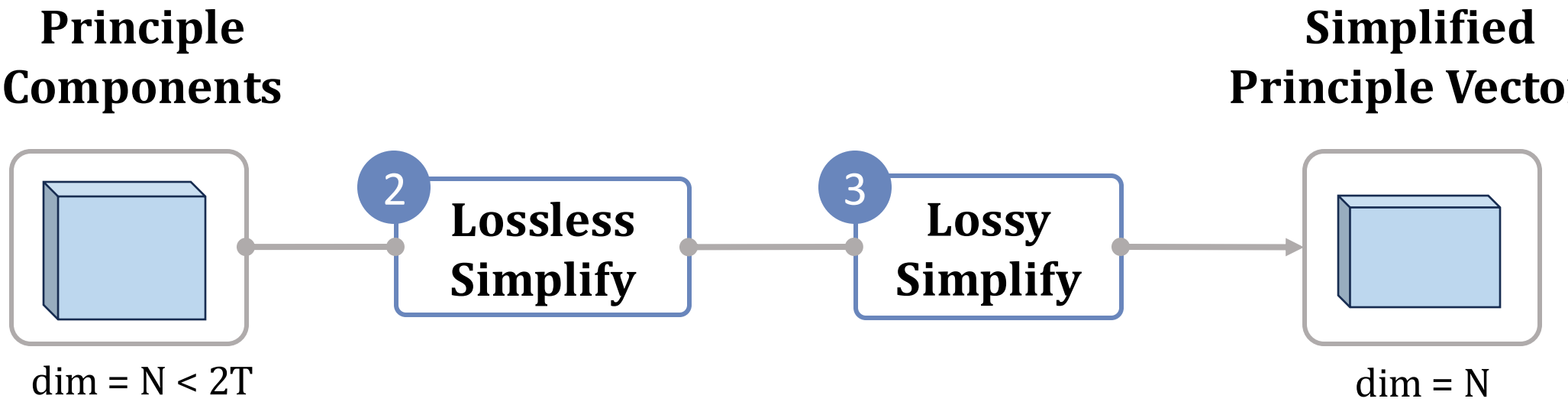
original
#segments: 1000

ours
#segments: 100

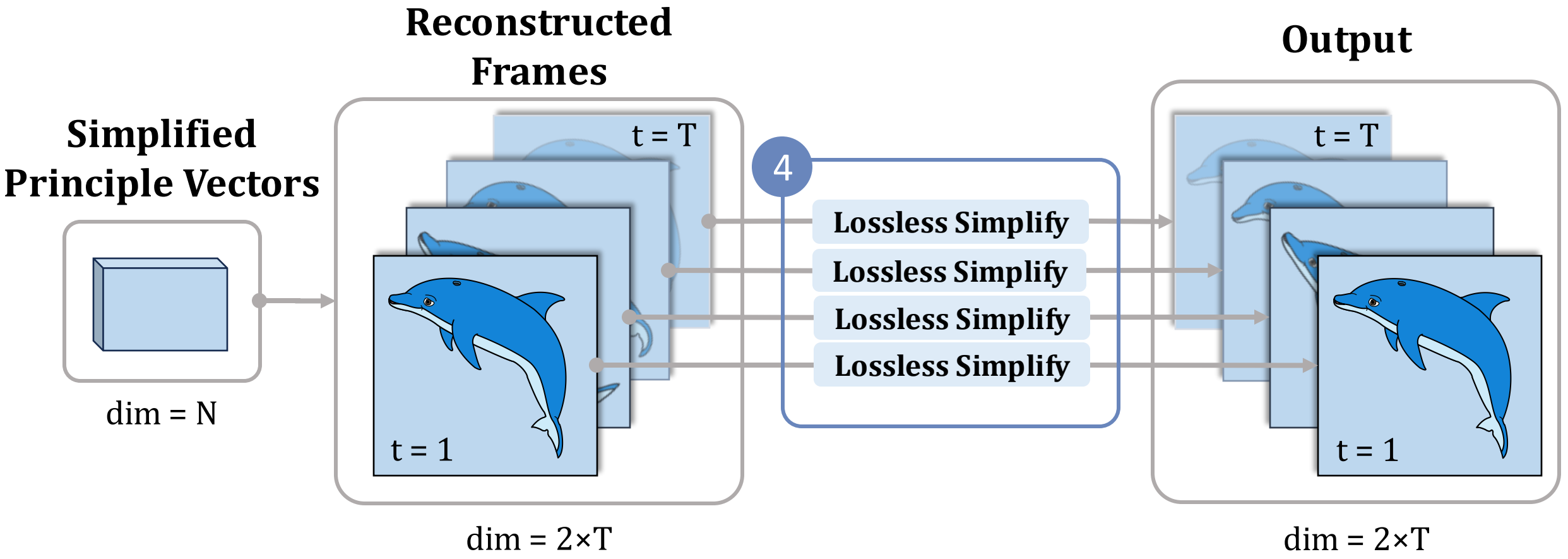
Vector graphics animation



Vector graphics animation

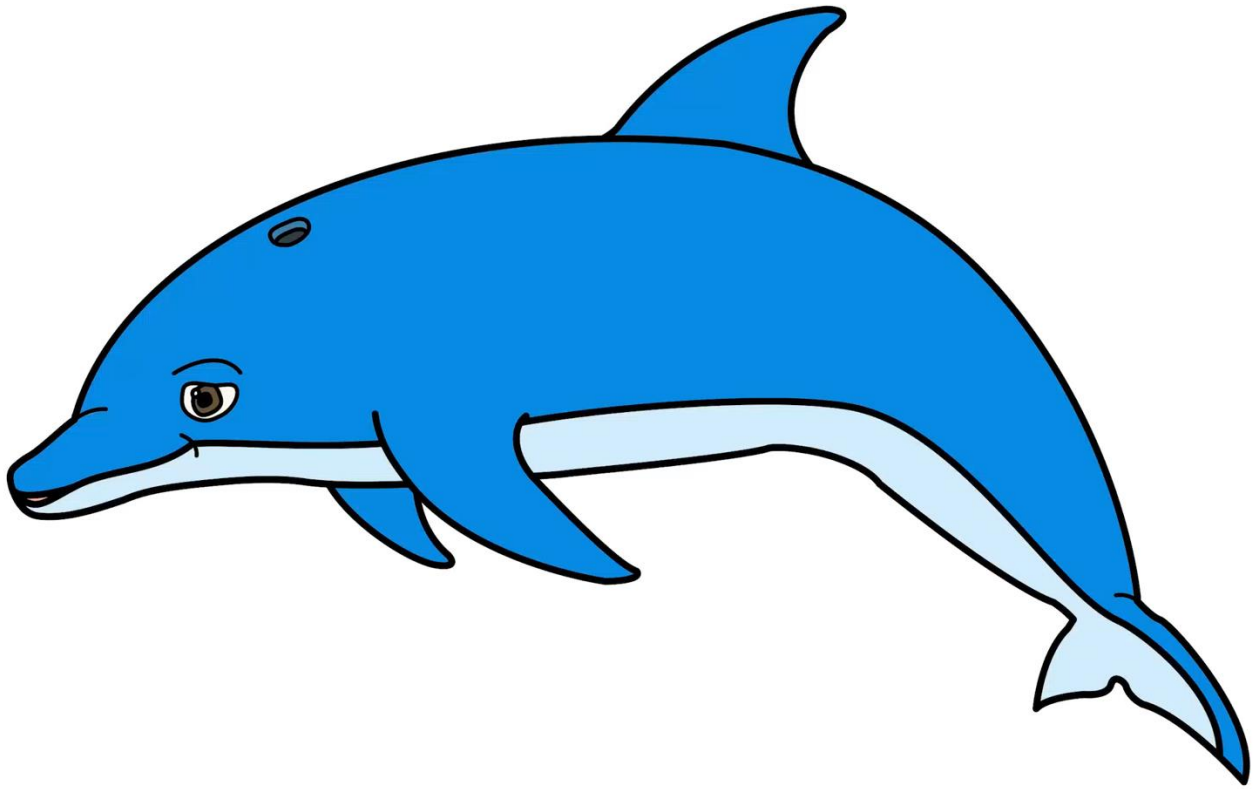


Vector graphics animation



Vector graphics animation

Input: Dense Animation



#segments: 1130

Limitations and Future work

- ❑ Parallelism analogous to surface mesh simplification in parallel to speed up.
- ❑ Our method does *not* conduct topological simplification.
- ❑ More perceptually accurate corner detection.
- ❑ Extension to spline surface simplification.

Code: MATLAB (complex-step numerical differentiation)

C++ (automatic differentiation)

Waiting for release approval
Coming out soon!

https://cs.nyu.edu/~sw4429/ https://github.com/rachael-wang sw4429@nyu.edu