Deep Sketch Vectorization via Implicit Surface Extraction

Chuan Yan[1], Yong Li[2,1], Deepali Aneja[3], Matthew Fisher[3], Edgar Simo-Serra[4], Yotam Gingold[1]

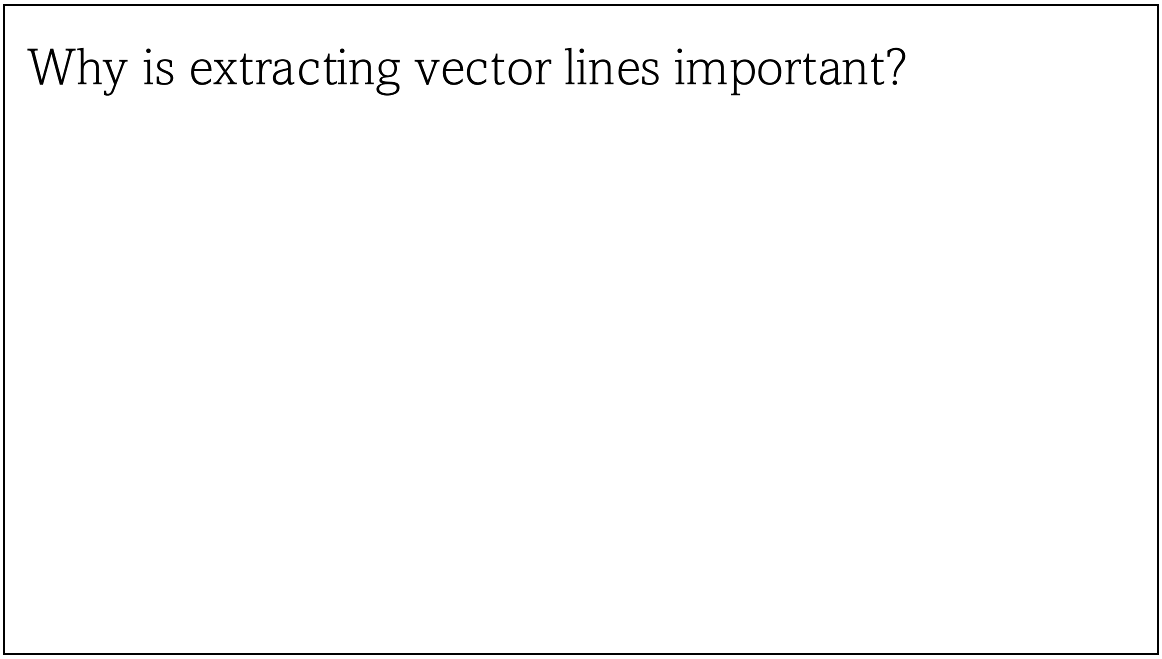
1. GEORGE MASON UNIVERSITY CraGL Computational Reality Creativity and Graphics Lab

2. SOUTH CHINA UNIVERSITY OF TECHNOLOGY

3. Adobe

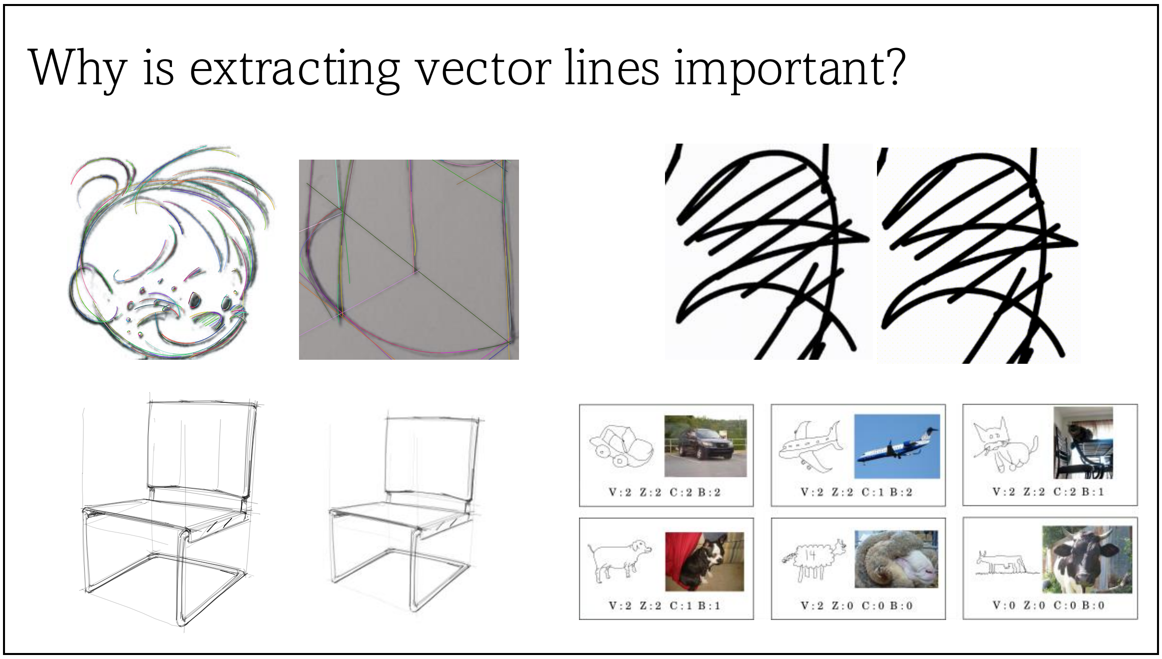
4 WASEDA University

Hi everyone, my name is Chuan Yan and today I will introduce our work: Deep Sketch Vectorization via Implicit Surface Extraction. This work is a collaboration with Yong Li from south China university of technology, Deepali Aneja. Matthew Fisher from Adobe, Edgar Simo-serra from Waseda university and Yotam Gingold from George Mason university



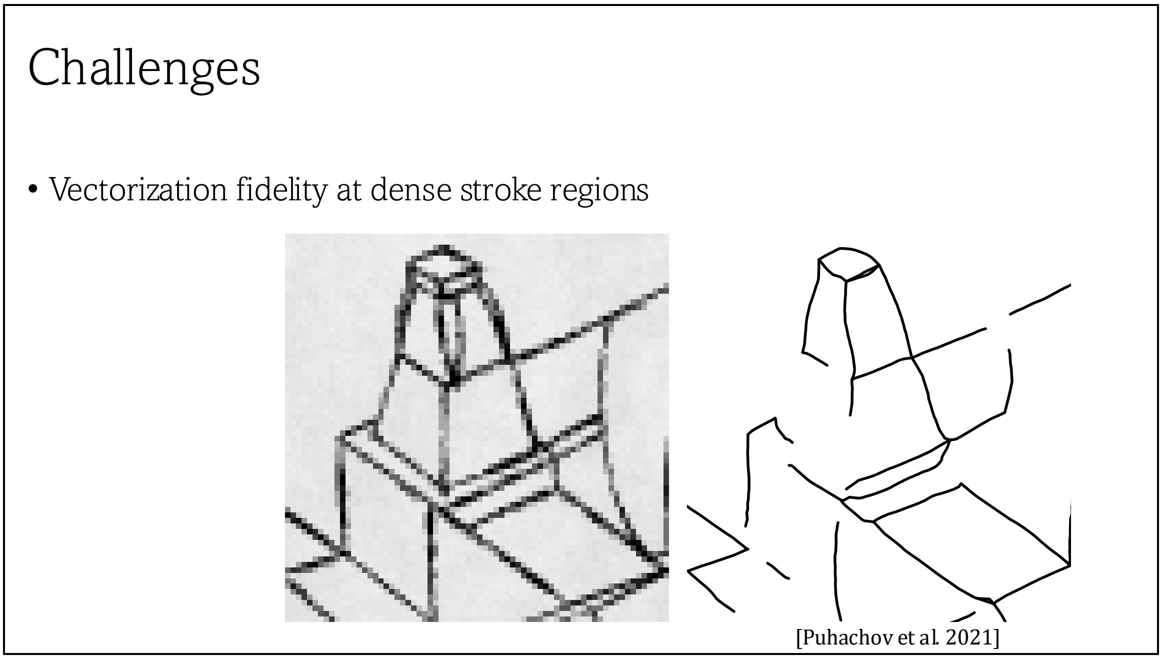
Why is extracting vector lines important?

Accurately extract the vector lines from raster sketches is a beneficial step for many aspects.
First, vector lines and junctions could preserve important information that is difficult to extract from bitmaps only such as the artist’s drawing intent, the 2D or 3D shape .
Also, it is easy to only choose the target strokes and adjust them without touching other neighbor strokes.
A lot of downstream sketch processing applications usually requires vector lines as input.
At last, vector lines are necessary for many human vison abstraction and expression researches such as sketch retrieval, sketch recognition, etc.



Why is extracting vector lines important?

Accurately extract the vector lines from raster sketches is a beneficial step for many aspects.
First, vector lines and junctions could preserve important information that is difficult to extract from bitmaps only such as the artist's drawing intent, the 2D or 3D shape .
Also, it is easy to only choose the target strokes and adjust them without touching other neighbor strokes.
A lot of downstream sketch processing applications usually requires vector lines as input.
At last, vector lines are necessary for many human vison abstraction and expression researches such as sketch retrieval, sketch recognition, etc.

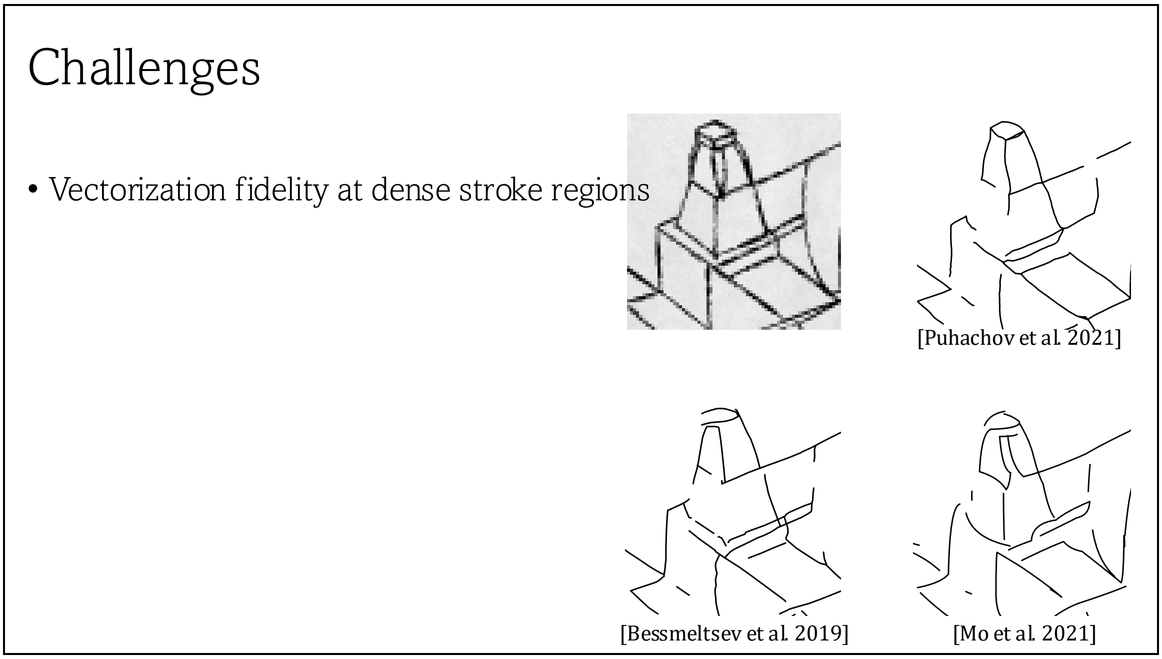


Challenges

- Vectorization fidelity at dense stroke regions

[Puhachov et al. 2021]

However, there are still challenges for existing sketch vectorization methods. The frame field-based methods or fully end-to-end data driven based methods are all struggle at extracting accurate vector paths on the sketch regions where contains dense strokes.



Challenges

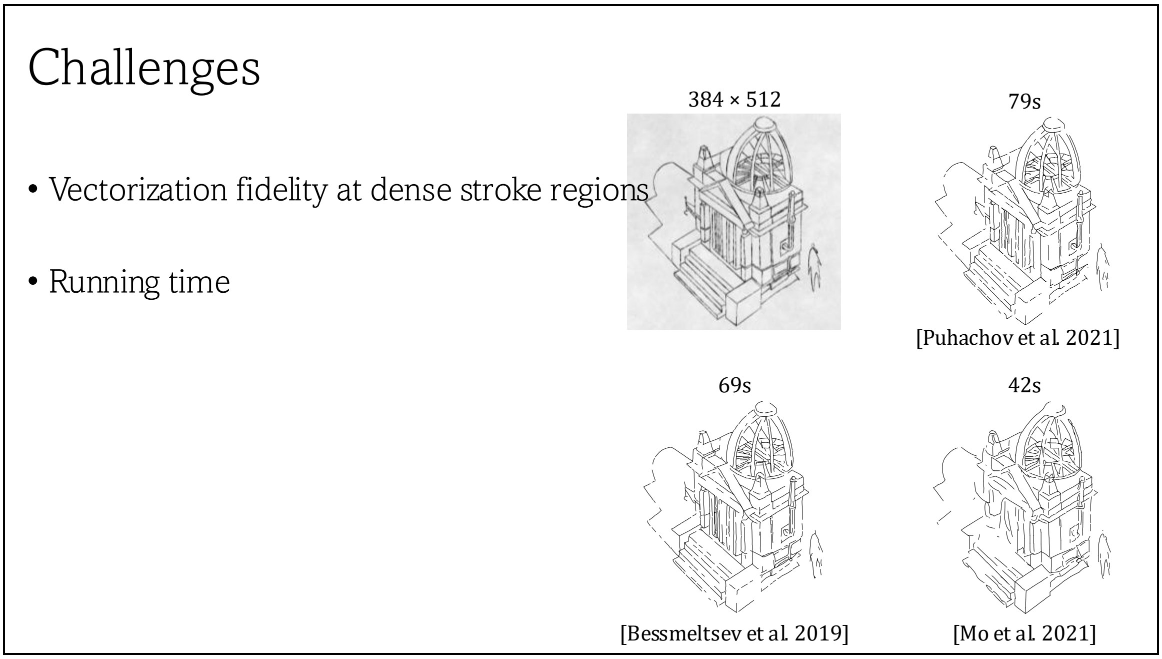
- Vectorization fidelity at dense stroke regions

[Puhachov et al. 2021]

[Bessmeltsev et al. 2019]

[Mo et al. 2021]

However, there are still challenges for existing sketch vectorization methods. The frame field-based methods or the fully end-to-end data driven based methods all struggle to extract accurate vector paths on the sketch regions containing dense strokes.



Challenges

- Vectorization fidelity at dense stroke regions
- Running time

384 × 512

79s

[Puhachov et al. 2021]

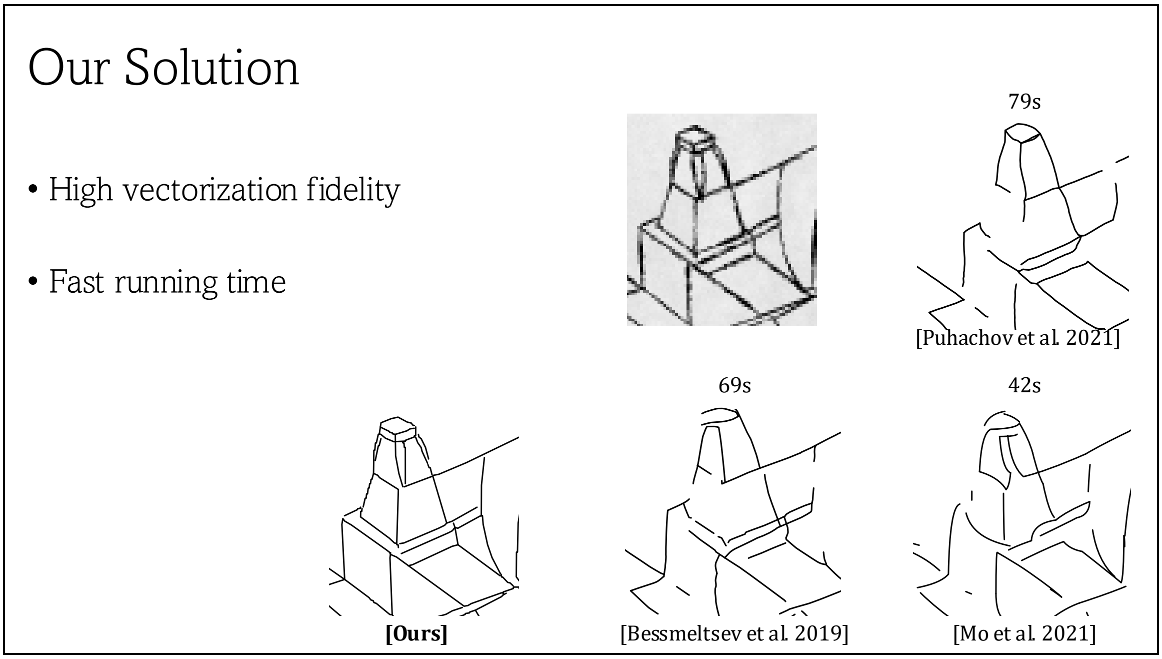
69s

[Bessmeltsev et al. 2019]

42s

[Mo et al. 2021]

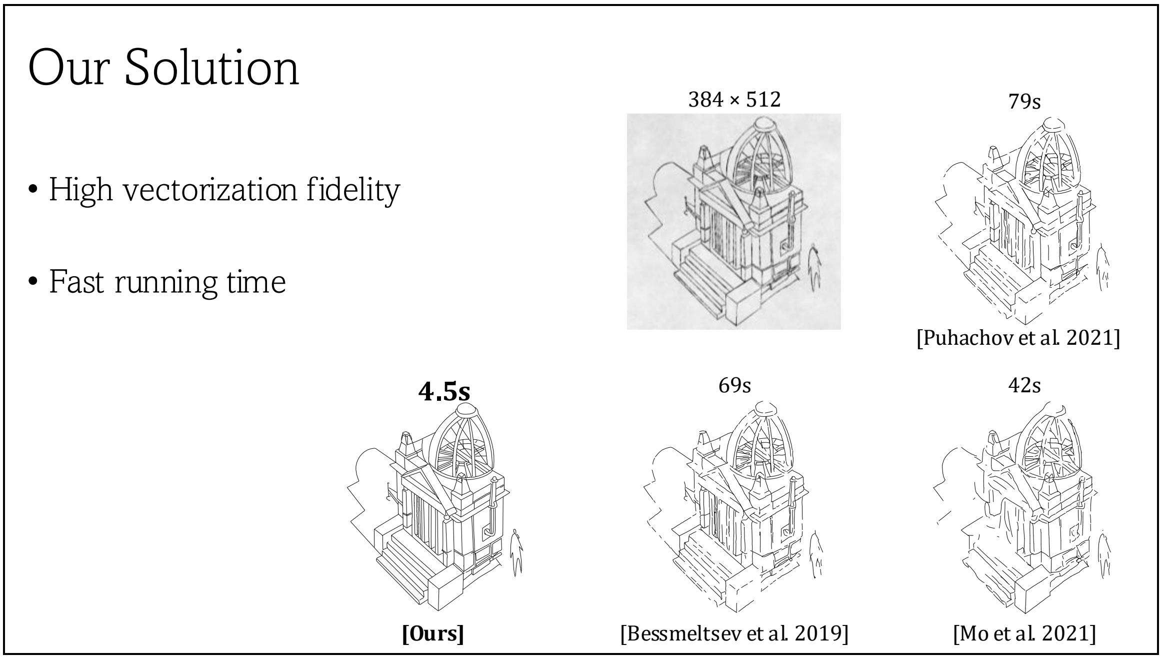
And it will be difficult to further reduce the runtime if the raster sketches contain complex structures with numerous junctions.
For example, for a 384 x 512 raster sketch, the current methods will take at least more than 40 seconds.



Our Solution

- High vectorization fidelity
- Fast running time

We have proposed a new sketch vectorization method that provides much higher vectorization fidelity and faster execution time.



Our Solution

- High vectorization fidelity
- Fast running time

384 × 512

79s

[Puhachov et al. 2021]

4.5s

69s

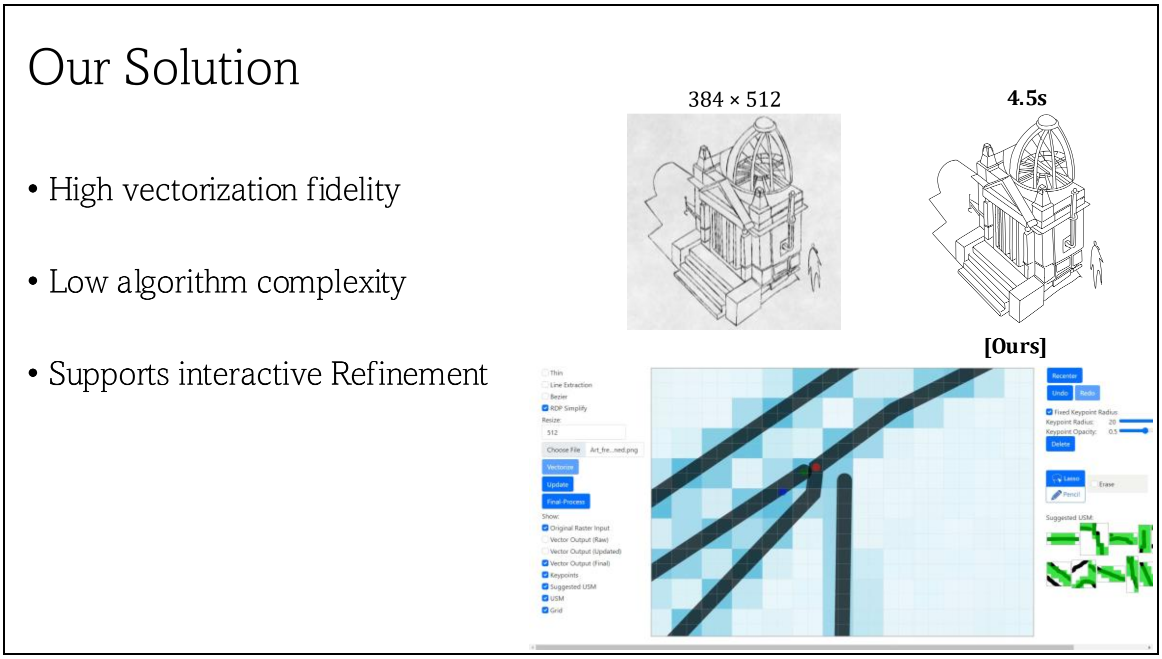
42s

[Ours]

[Bessmeltsev et al. 2019]

[Mo et al. 2021]

We have proposed a new sketch vectorization method that provides much higher vectorization fidelity and faster execution time.



Our Solution

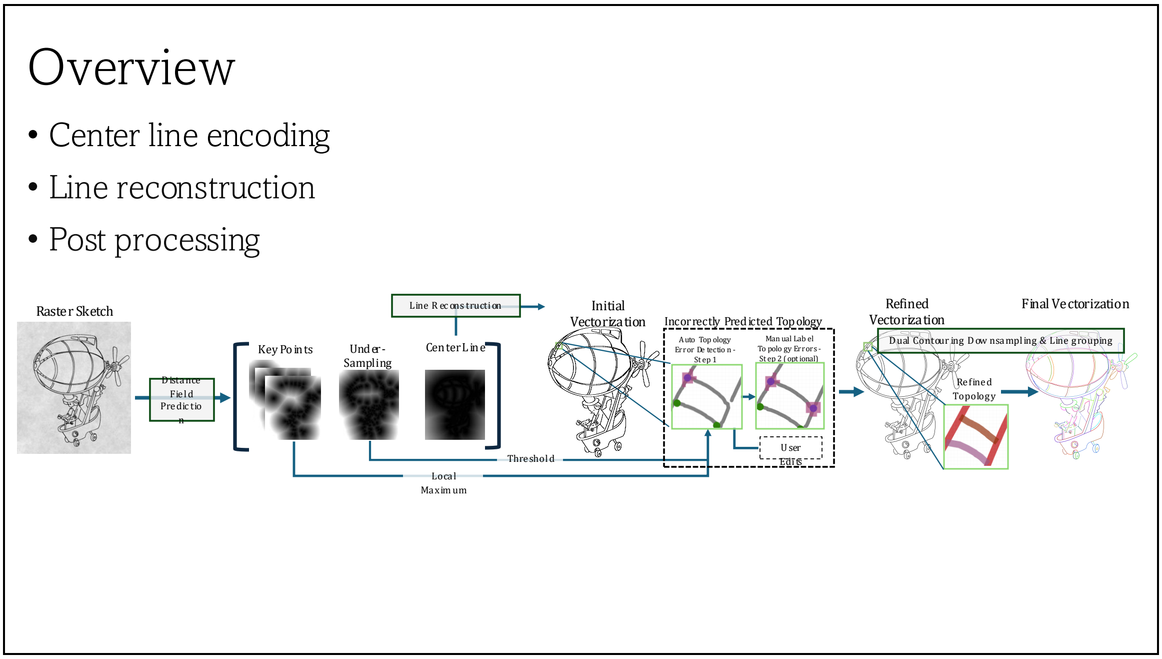
- High vectorization fidelity
- Low algorithm complexity
- Supports interactive Refinement

☐ Thin
☐ Line Extraction
☐ Bezier
☑ RDP Simplify
Resize: 512
Choose File Art_fre...ned.png
Show:
☑ Original Raster Input
☐ Vector Output (Raw)
☐ Vector Output (Updated)
☑ Vector Output (Final)
☑ Keypoints
☑ Suggested USM
☑ USM
☑ Grid

☑ Fixed Keypoint Radius
Keypoint Radius: 20
Keypoint Opacity: 0.5
☐ Erase
Suggested USM:

Additionally, our intermediate vector path representation naturally support further interactive topology Refinement if necessary.

But, all results shown in this slides are fully automatic unless stated otherwise

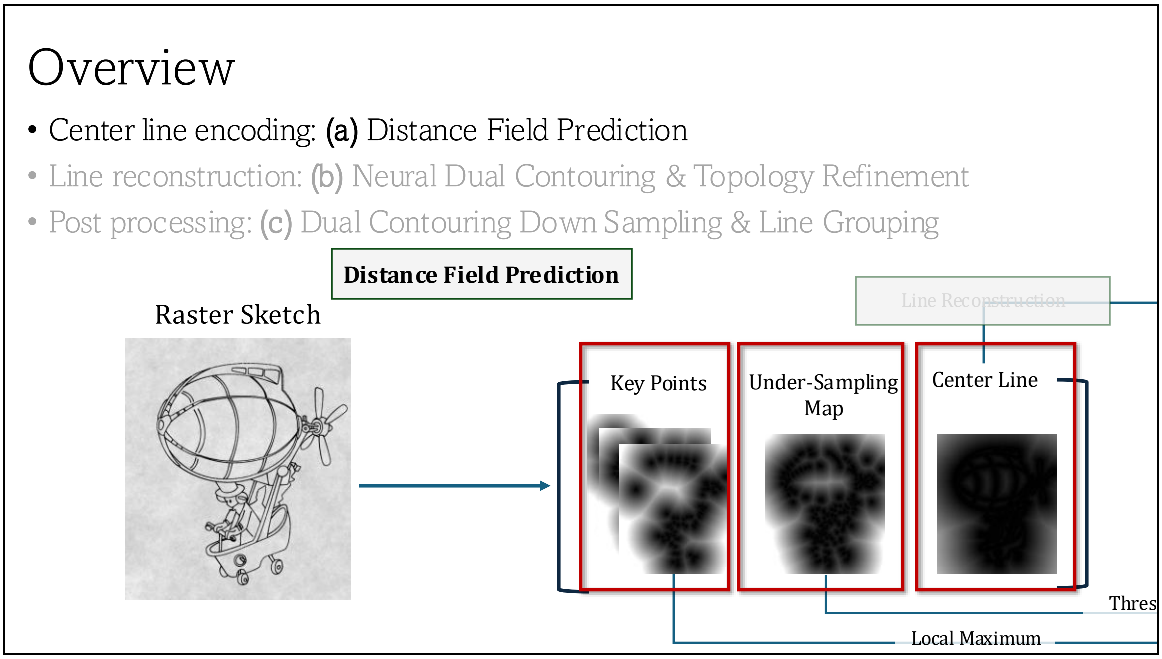


Overview

- Center line encoding
- Line reconstruction
- Post processing

Raster Sketch
Distance Field Prediction
Key Points
Under-Sampling
Center Line
Line Reconstruction
Initial Vectorization
Incorrectly Predicted Topology
Auto Topology Error Detection - Step 1
Manual Label Topology Errors - Step 2 (optional)
User Edits
Threshold
Local Maximum
Refined Vectorization
Final Vectorization
Dual Contouring Downsampling & Line grouping
Refined Topology

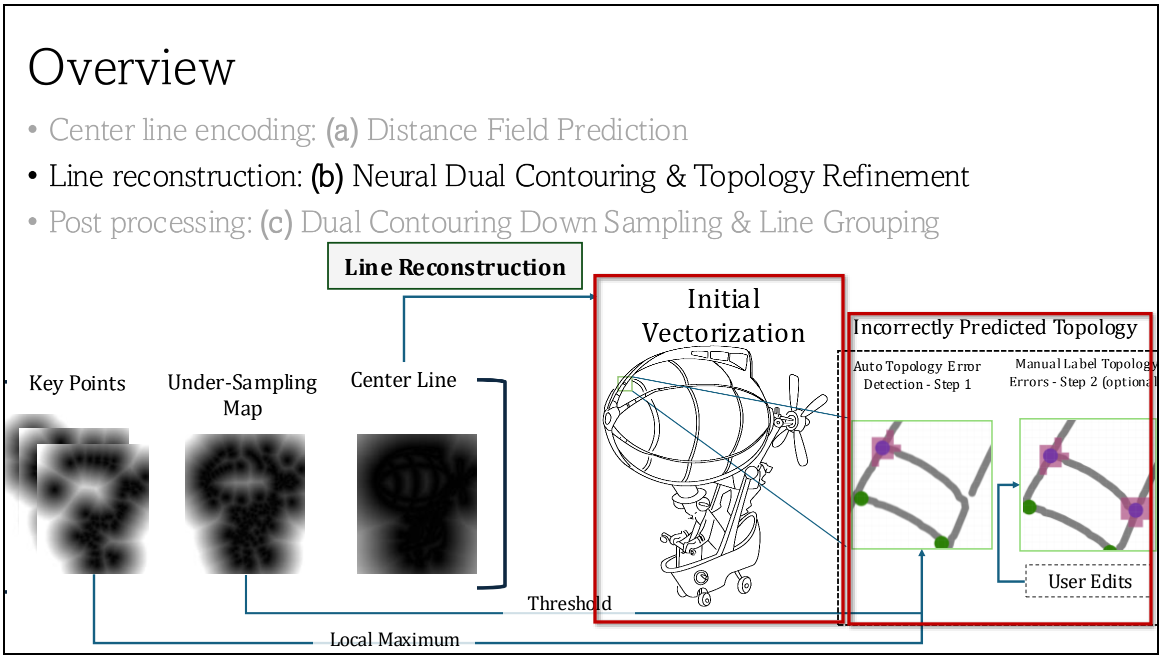
We designed our vectorization framework into 3 stages: center line encoding, vector line reconstruction and post Refinements



Overview

- Center line encoding: (a) Distance Field Prediction
- Line reconstruction: (b) Neural Dual Contouring & Topology Refinement
- Post processing: (c) Dual Contouring Down Sampling & Line Grouping

At the first center line encoding stage, we trained a distance field prediction network that extract information such as key point, under sampling map and center line from the given raster sketch.
All this information are encoded as Unsigned Distance Fields.



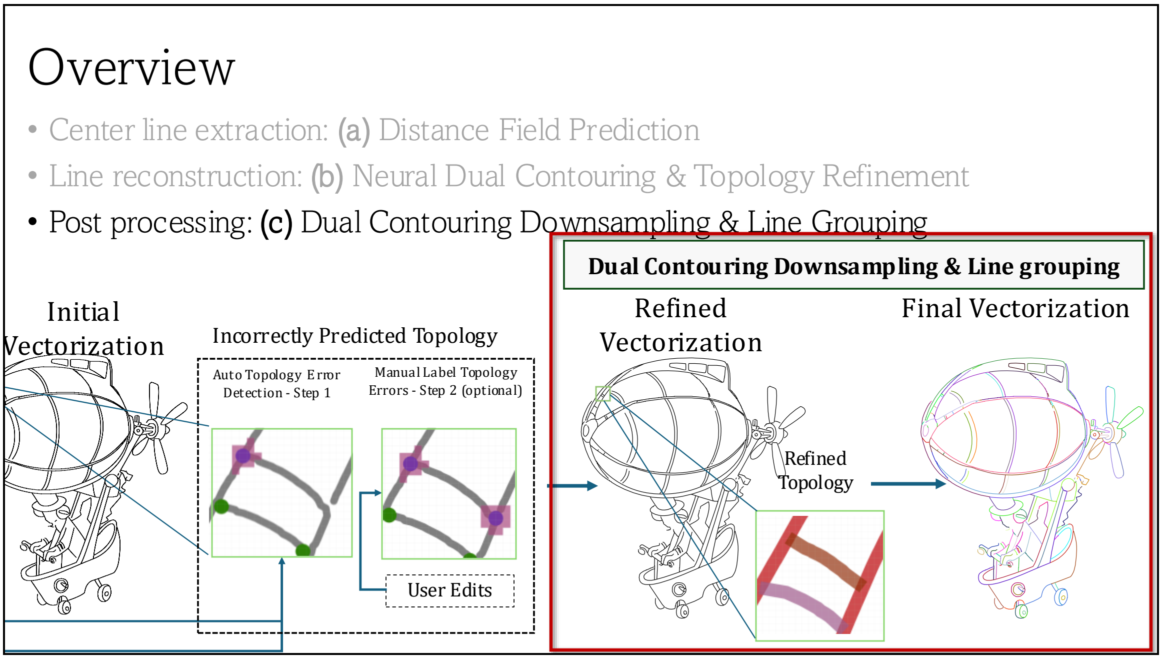
Overview

- Center line encoding: (a) Distance Field Prediction
- Line reconstruction: (b) Neural Dual Contouring & Topology Refinement
- Post processing: (c) Dual Contouring Down Sampling & Line Grouping

Line Reconstruction
Key Points
Under-Sampling Map
Center Line
Initial Vectorization
Incorrectly Predicted Topology
Auto Topology Error Detection - Step 1
Manual Label Topology Errors - Step 2 (optional
User Edits
Threshold
Local Maximum

Then we trained another network to reconstruct initial vector paths from the encoded center line which is predicted in our first stage.

After getting the initial vector lines, its junctions will be automatically refined based on the predicted key point and under sampling maps.



Overview

- Center line extraction: (a) Distance Field Prediction
- Line reconstruction: (b) Neural Dual Contouring & Topology Refinement
- Post processing: (c) Dual Contouring Downsampling & Line Grouping

Initial Vectorization

Incorrectly Predicted Topology

Auto Topology Error Detection - Step 1

Manual Label Topology Errors - Step 2 (optional)

User Edits

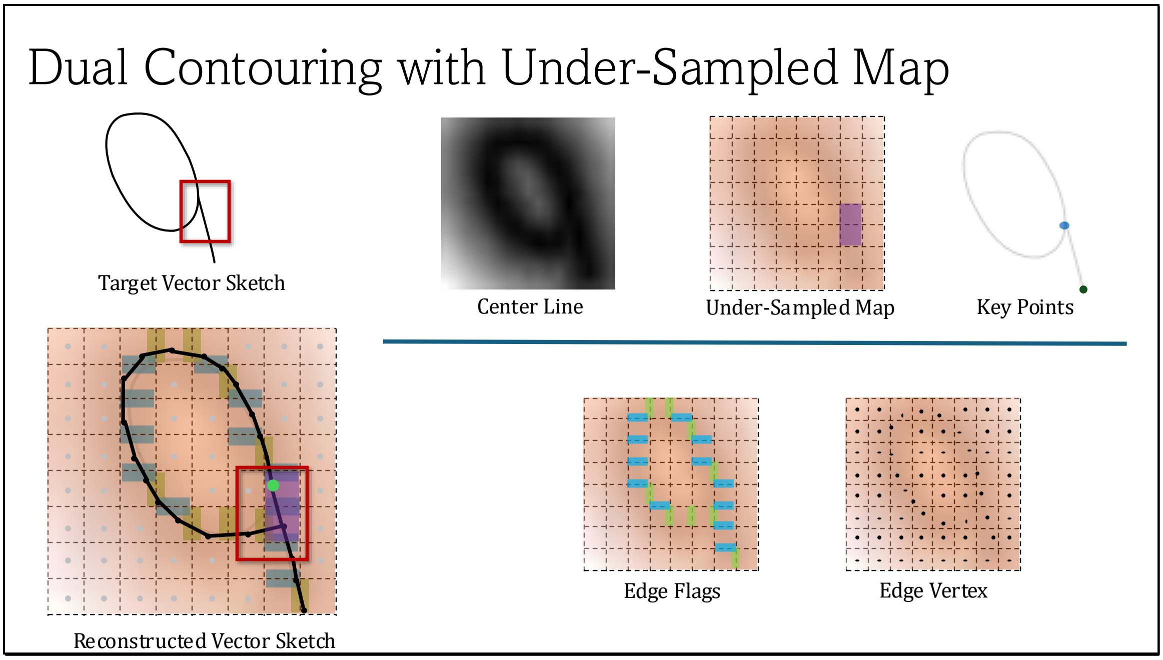
Dual Contouring Downsampling & Line grouping

Refined Vectorization

Refined Topology

Final Vectorization

At last, to further refine the output.
we proposed a dual contouring down sampling method and a simple line grouping method to turn vector segments into vector strokes with optimized file size.
Now let’s dive into the technical details



Dual Contouring with Under-Sampled Map

One of our core contribution is that we formulate the sketch vectorization as an implicit surface reconstruction task which is Dual Contouring in our method. Therefore, we will briefly review its reconstruction logic in 2D first.

Given a target vector sketch, we sampled the canvas into a bunch of evenly distributed grids.

For each grid, we compute the closest distance from the grid center to the vector paths as an Unsigned Distance field.
Since this field implicitly encoded the vector path as a 2D surface, we can use dual contouring to reconstruct it as vector segments based on this field.

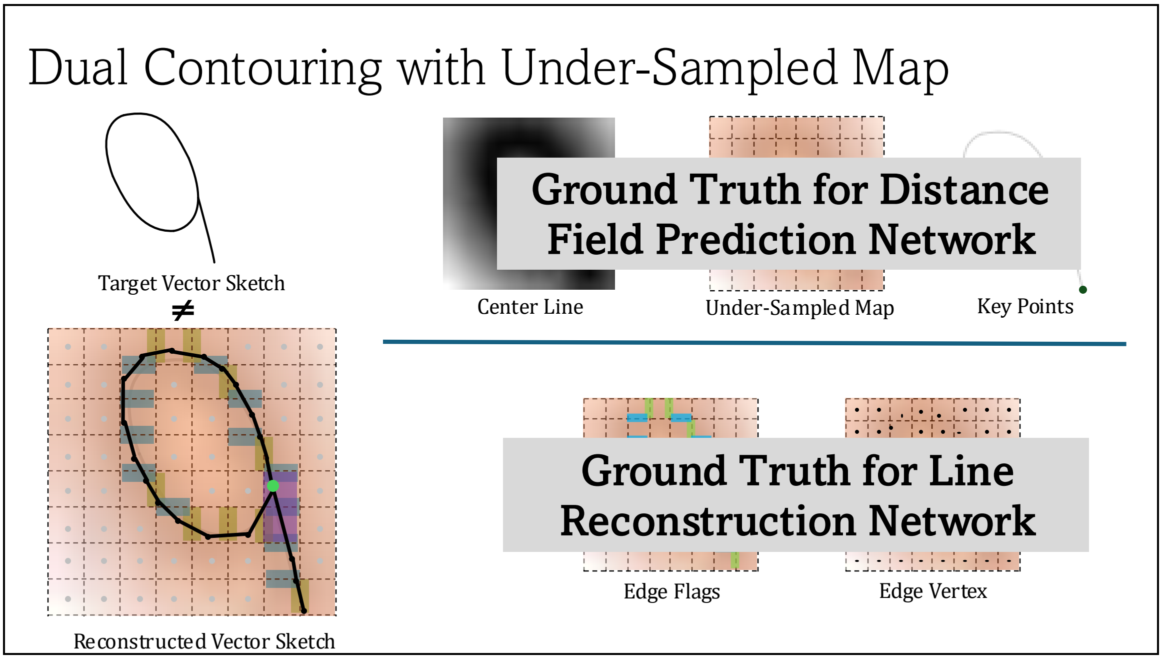
To be more specific, we could infer the edge flags and edge vertex from it.
The edge flag records the intersection between the target vector sketch and grid edge.
And the edge vertex in each grid is set onto the vector sketch if the grid contains single vector path.
Each grid could only have one edge vertex.

Then the reconstruction will be simply connecting any 2 neighbor vertices if their shared grid edge has a flag.

However, this reconstruction could be inaccurate since our sampling rate in the vector space is not enough.

And it is theoretically impossible to reconstruct the high-valence junctions with dual contouring. We termed those regions as under sampled regions.

To solve this problem, we detect those under sampled regions and extract the junction key points



Dual Contouring with Under-Sampled Map

Ground Truth for Distance Field Prediction Network

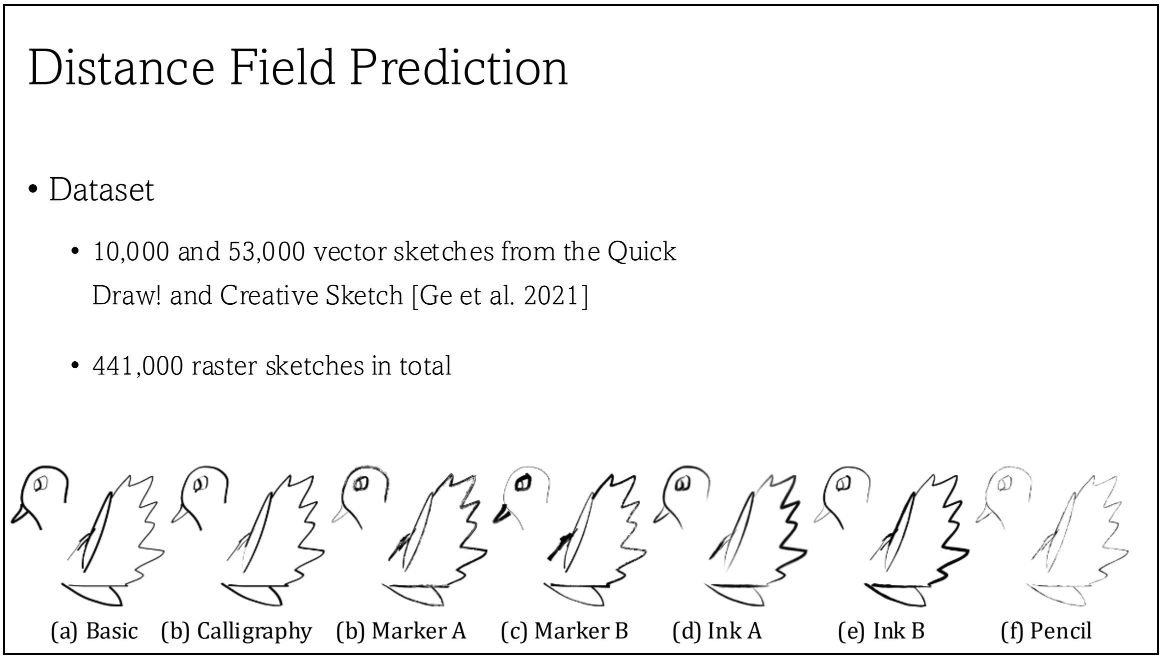
Ground Truth for Line Reconstruction Network

Then we can make correction by dropping all the line segments in the under sampled regions and rebuild the junction by connecting the key point to the all the truncations.

To sum up, to get the reconstructed vector sketch, we need derive this 5 information which are our training ground truth from a given target vector sketch.

The upper 3 are the ground truth for our distance filed prediction network, the lower 2 is the ground truth for our line reconstruction network.

It is also worth noting that although the reconstructed vector sketch is not equal to the target vector sketch, this design of intermediate vector representation could greatly reduce the training difficulty and naturally support interactive junction Refinement.

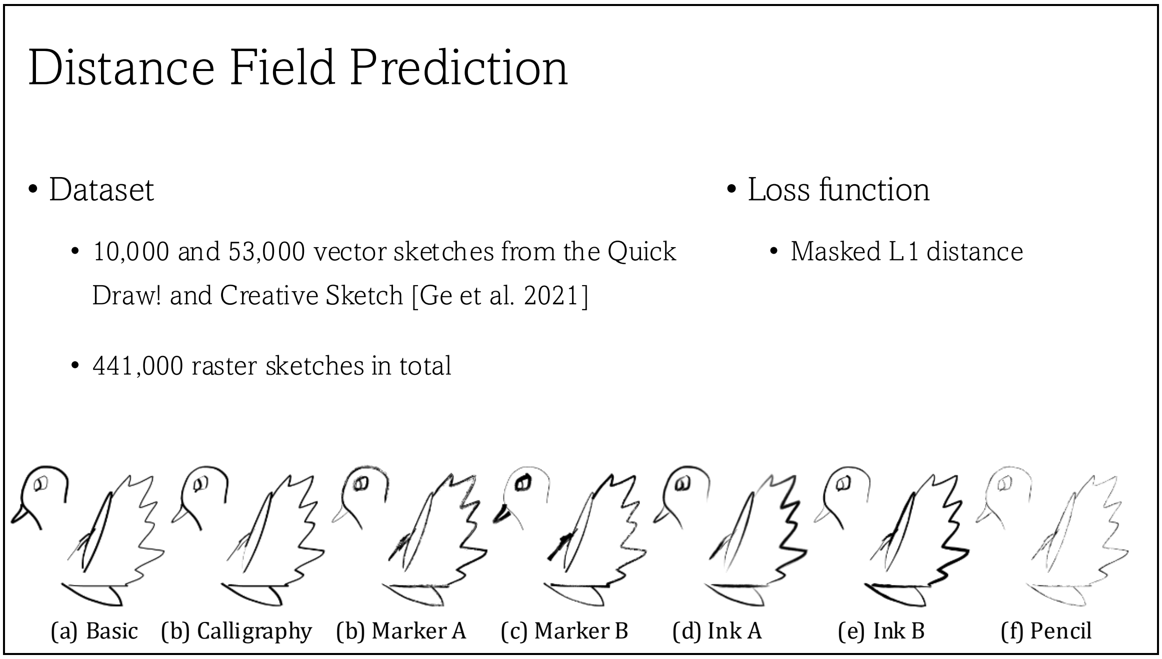


Distance Field Prediction

- Dataset
  - 10,000 and 53,000 vector sketches from the Quick Draw! and Creative Sketch [Ge et al. 2021]
  - 441,000 raster sketches in total

(a) Basic (b) Calligraphy (b) Marker A (c) Marker B (d) Ink A (e) Ink B (f) Pencil

We then create our dataset with 63k vector sketches from public dataset then rasterize them with 7 different brush styles. This ends up with 441k raster sketches as our training data. One group of the raster sketch examples is shown below.

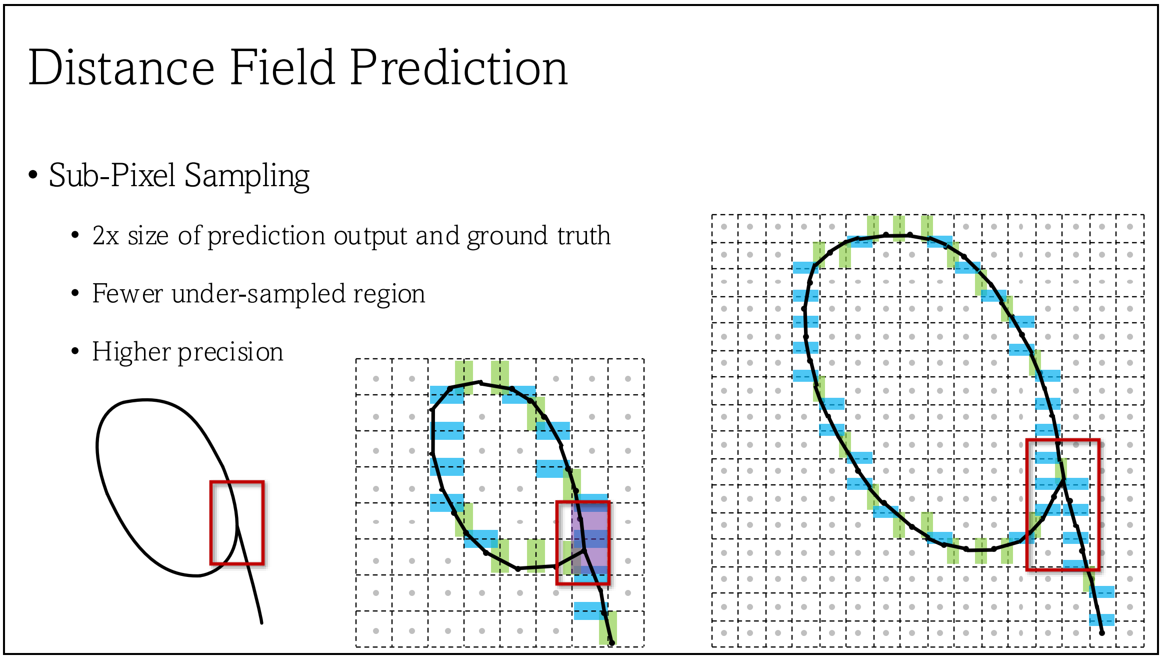


Distance Field Prediction

- Dataset
  - 10,000 and 53,000 vector sketches from the Quick Draw! and Creative Sketch [Ge et al. 2021]
  - 441,000 raster sketches in total
- Loss function
  - Masked L1 distance

(a) Basic (b) Calligraphy (b) Marker A (c) Marker B (d) Ink A (e) Ink B (f) Pencil

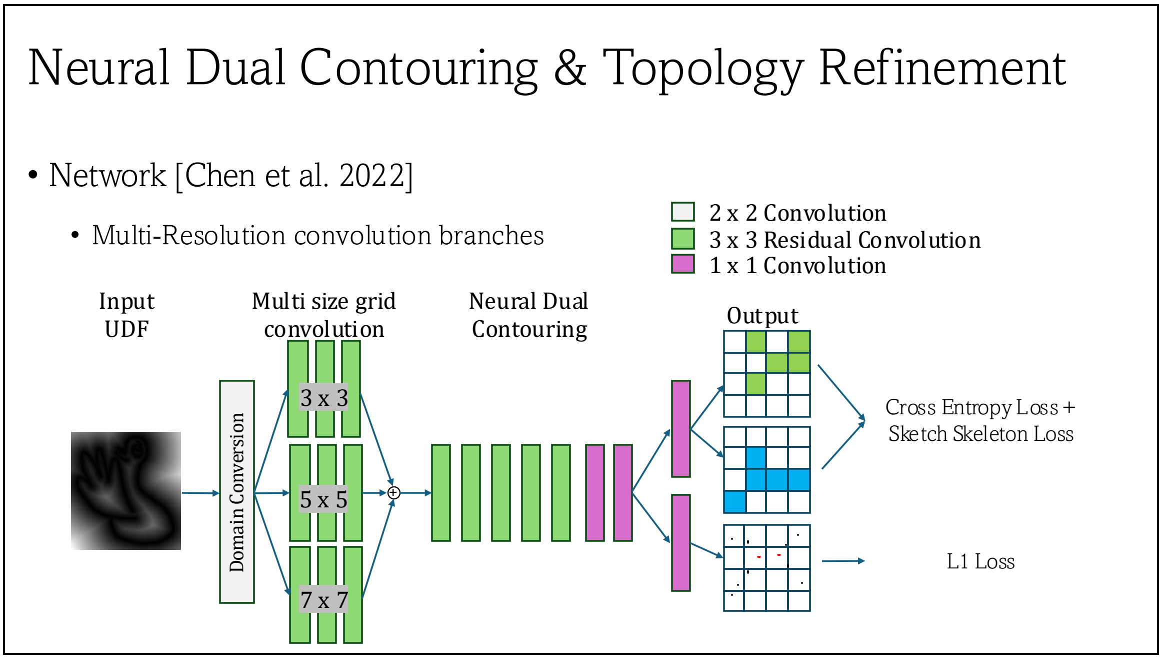
we formulated this distance field prediction as an image-to-image translation task and trained a U-Net like network based on the created data.
We using the masked L1 distance as our training loss to predict 3 type of unsigned distance fields that encoding the centerlines, under sampled maps and key points, respectively.



Distance Field Prediction

- Sub-Pixel Sampling
  - 2x size of prediction output and ground truth
  - Fewer under-sampled region
  - Higher precision

To reduce the number of under sampled regions and improve the smoothness of the reconstructed curve, we doubled the resolution of the output unsigned distance field which equivalent to double the sampling rate in the vector domain. This will also lead the resolution increase for both corresponding edge flags and edge vertex.

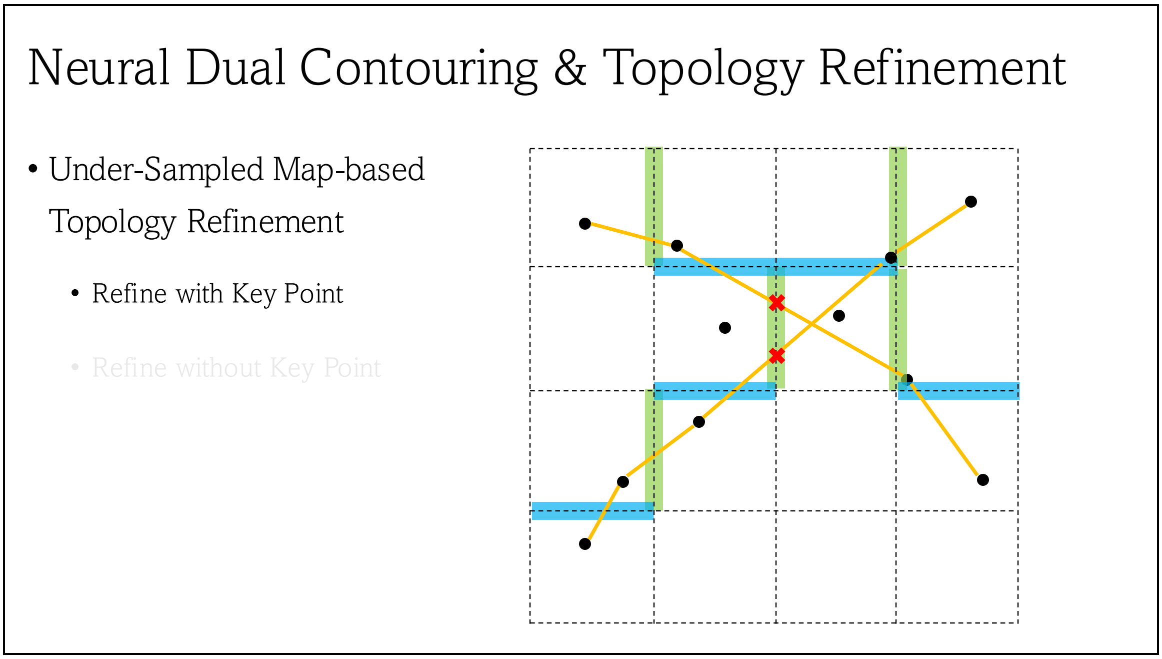


Neural Dual Contouring & Topology Refinement

- Network [Chen et al. 2022]
  - Multi-Resolution convolution branches

☐ 2 x 2 Convolution
🟩 3 x 3 Residual Convolution
🟪 1 x 1 Convolution

Then we adapt recent Neural Dual Contouring and applied a series of modifications which particularly optimized for 2D vectorization task. Please see our paper for more details.

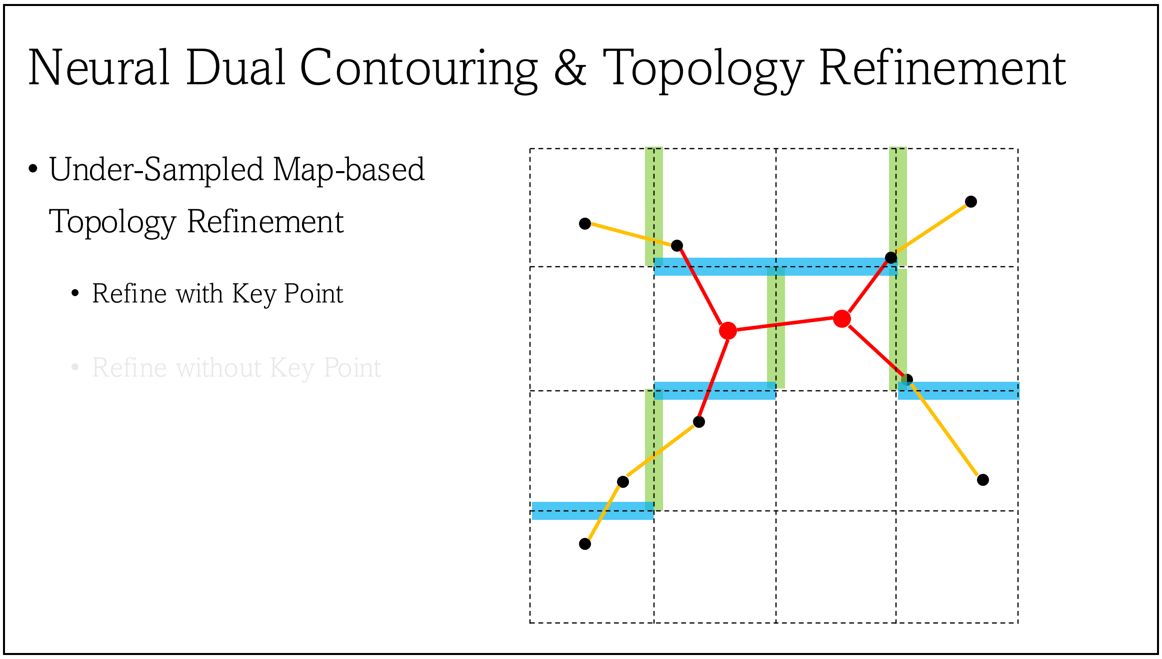


Neural Dual Contouring & Topology Refinement

- Under-Sampled Map-based Topology Refinement
  - Refine with Key Point
  - Refine without Key Point

Now I’m going to give more details about the logic how we refine the topology.

Given a target vector sketch with sampled 4 by 4 grids, we could easily find that the reconstructed vector lines can’t preserve the x junction even all edge flags are correctly predicted. Because there exists under sampled issue

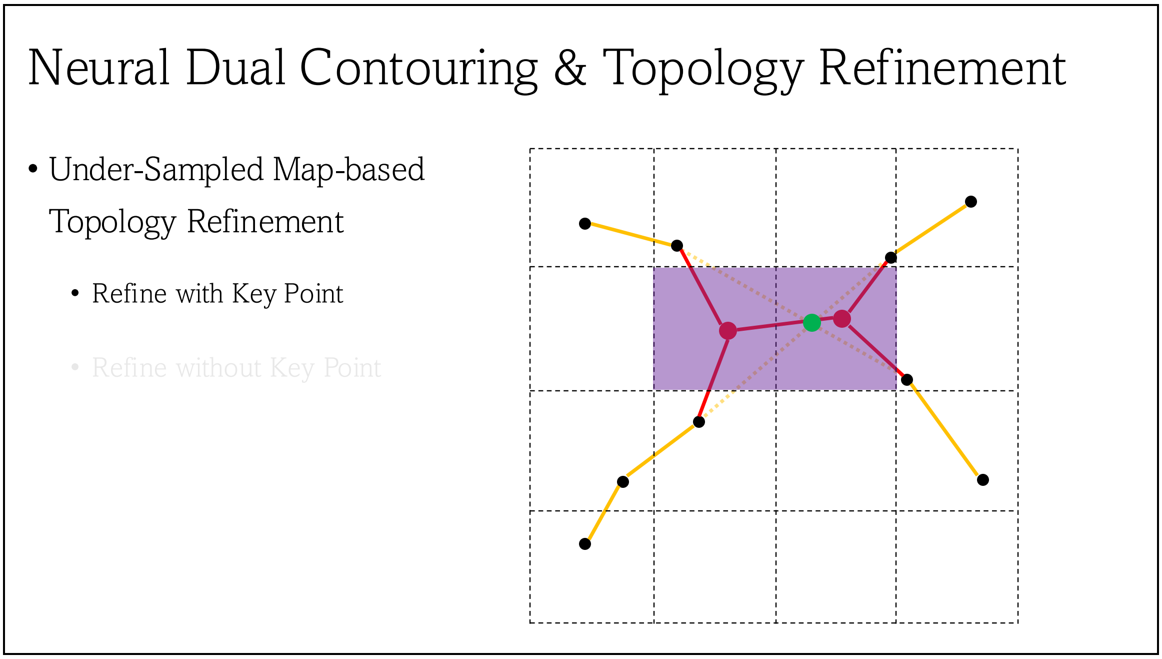


Neural Dual Contouring & Topology Refinement

- Under-Sampled Map-based Topology Refinement
  - Refine with Key Point
  - Refine without Key Point

Now I’m going to give more details about the logic how we refine the topology.

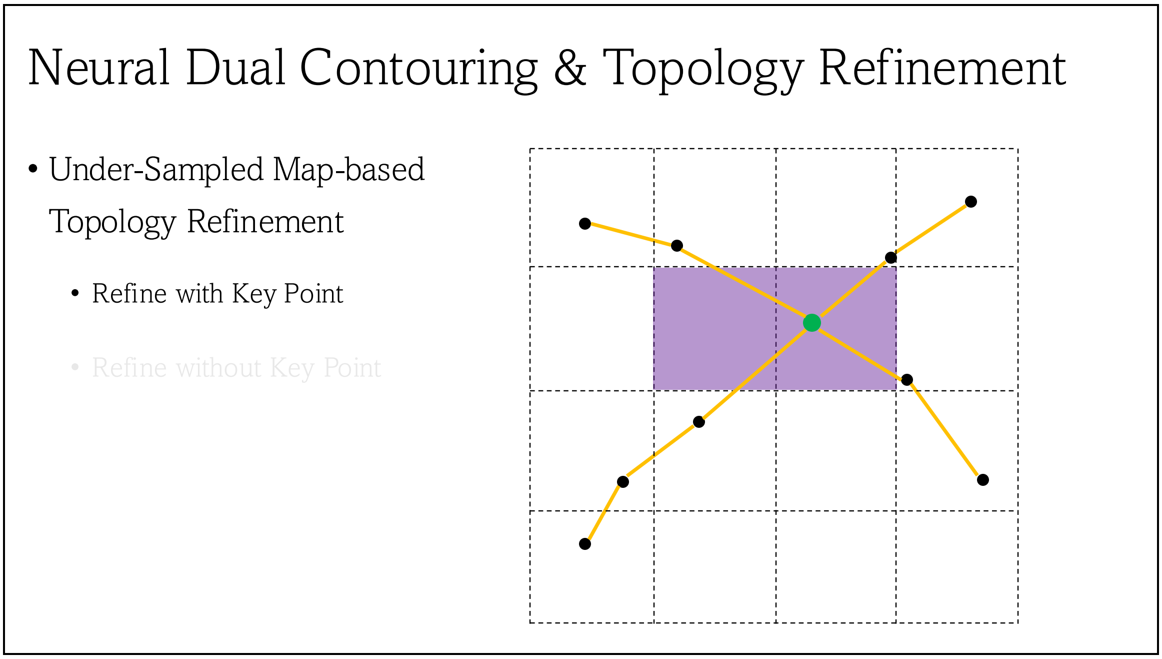
Given a target vector sketch with sampled as 4 by 4 grids, we could easily find that the reconstructed vector lines can’t preserve the x junction even all edge flags are correctly predicted.



Neural Dual Contouring & Topology Refinement

- Under-Sampled Map-based Topology Refinement
  - Refine with Key Point
  - Refine without Key Point

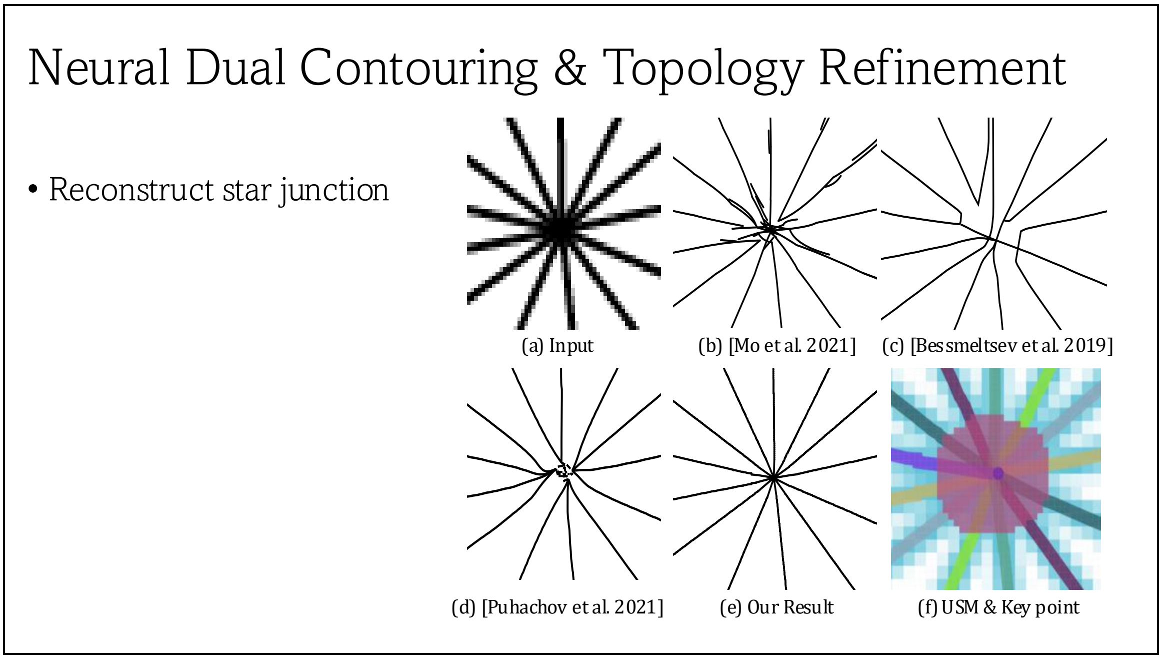
But we can still fix this by predicting a under sampling map along with the x junction key point.



Neural Dual Contouring & Topology Refinement

- Under-Sampled Map-based Topology Refinement
  - Refine with Key Point
  - Refine without Key Point

We remove all the line segments that inside the under-sampling map, then connect the all truncations to the predicted key point



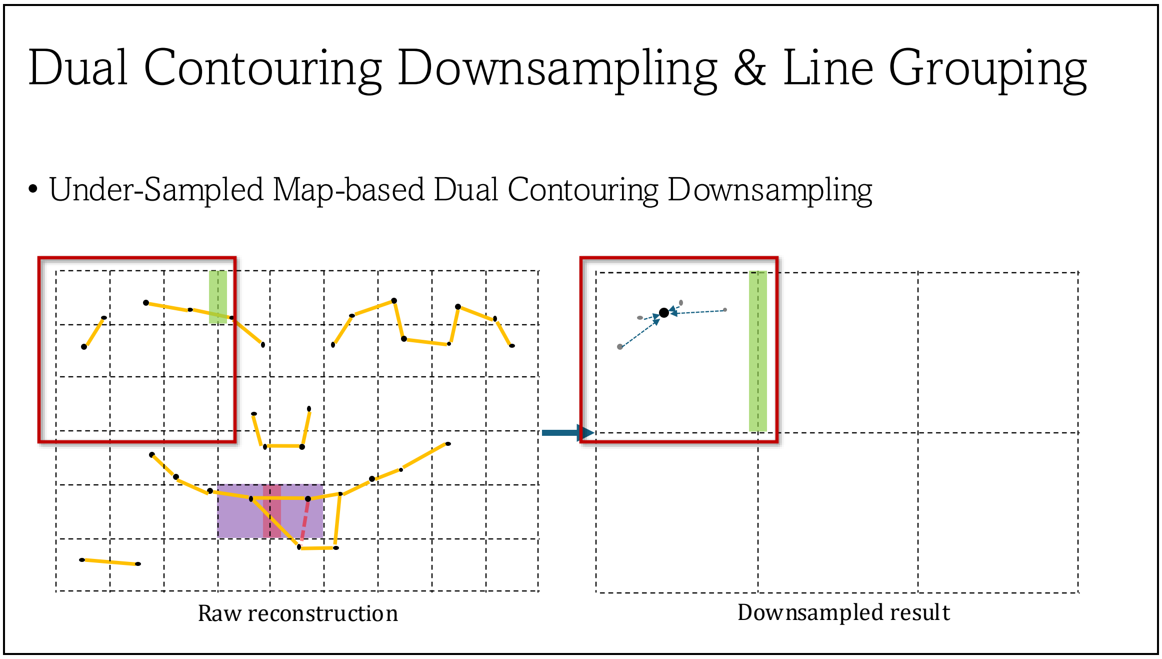
Neural Dual Contouring & Topology Refinement

- Reconstruct star junction

(a) Input (b) [Mo et al. 2021] (c) [Bessmeltsev et al. 2019]

(d) [Puhachov et al. 2021] (e) Our Result (f) USM & Key point

While reconstructing the high valence junction, or star junction in other word, is a historically challenging problem.
Our Refinement method provides one simple solution to it.



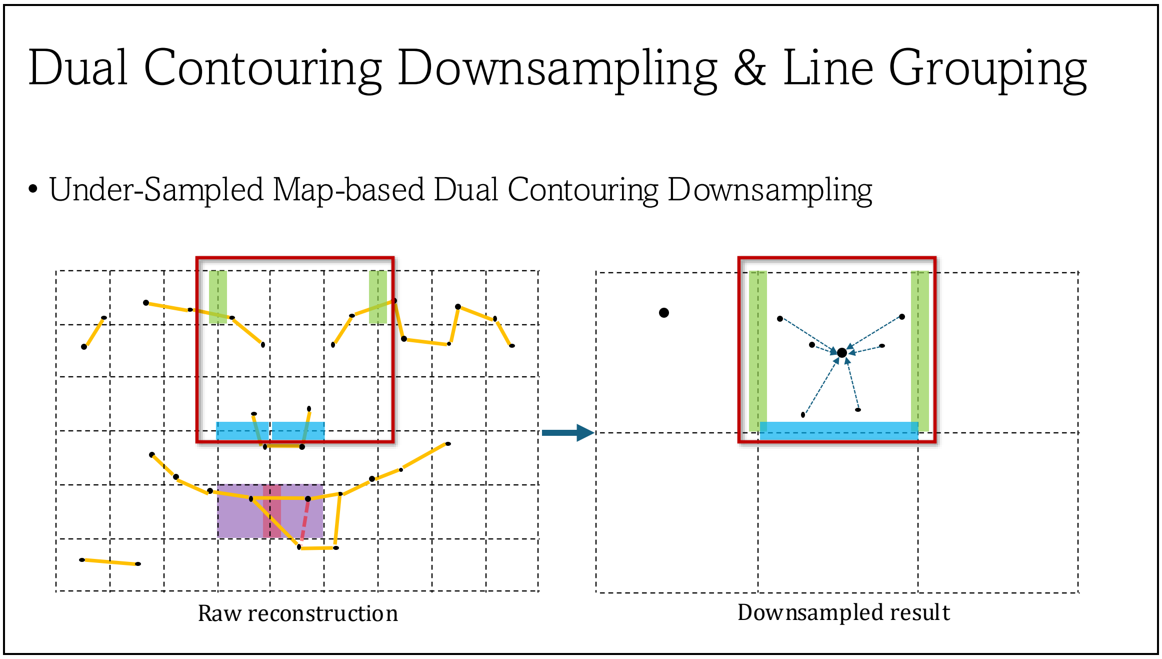
Dual Contouring Downsampling & Line Grouping

- Under-Sampled Map-based Dual Contouring Downsampling

Raw reconstruction

Downsampled result

Then to further reduce the number of unnecessary vector segments, we proposed an under-sampled map based dual contouring down sampling method
To turn vector segments into strokes, we proposed a simple line grouping method that iteratively fining shortest paths.
For more details, please check out our paper



Dual Contouring Downsampling & Line Grouping

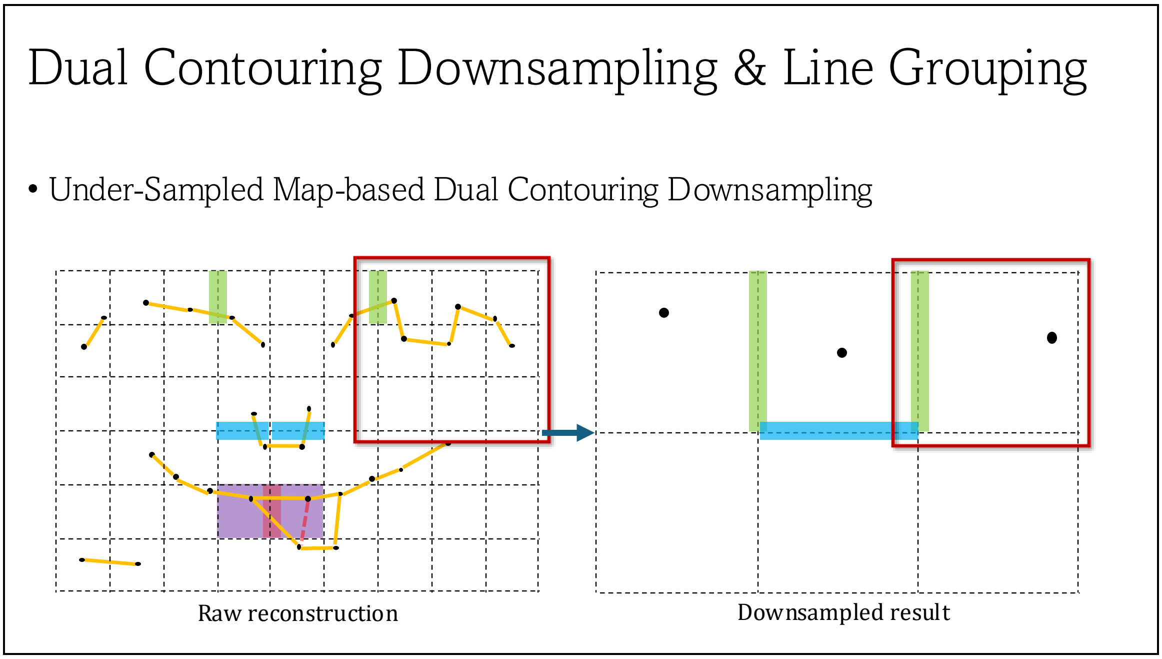
- Under-Sampled Map-based Dual Contouring Downsampling

Raw reconstruction

Downsampled result

Then to further reduce the number of unnecessary vector segments, we proposed an under-sampled map based dual contouring down sampling method To turn vector segments into strokes, we proposed a simple line grouping method that iteratively fining shortest paths.

For more details, please check out our paper



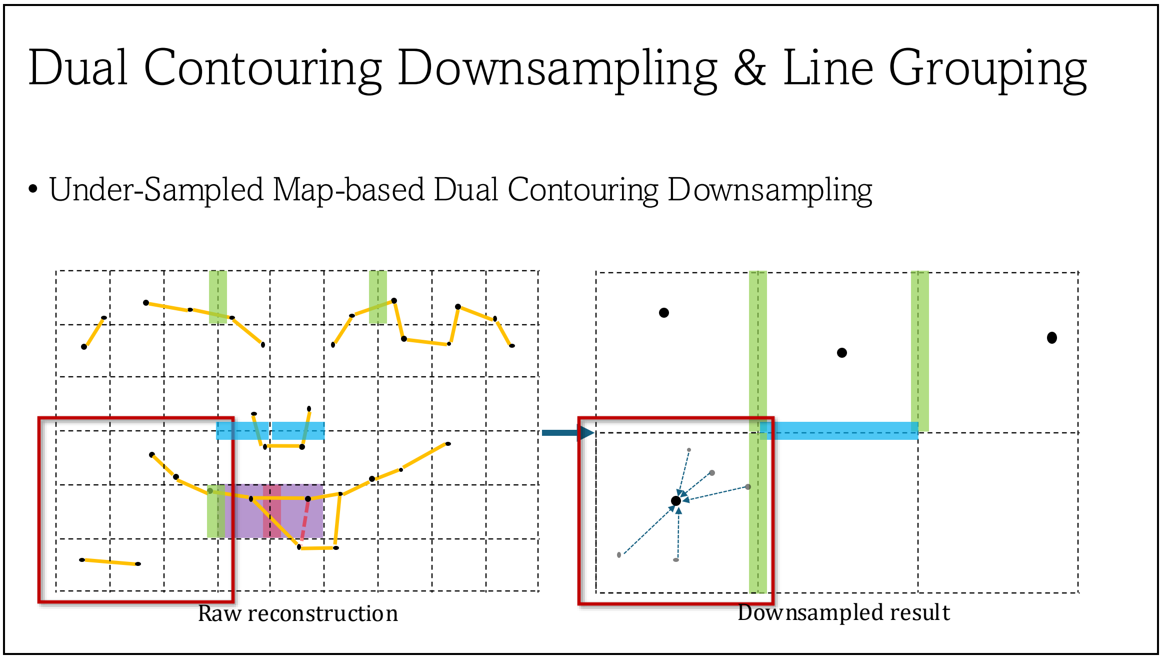
Dual Contouring Downsampling & Line Grouping

- Under-Sampled Map-based Dual Contouring Downsampling

Raw reconstruction

Downsampled result

Then to further reduce the number of unnecessary vector segments, we proposed an under-sampled map based dual contouring down sampling method
To turn vector segments into strokes, we proposed a simple line grouping method that iteratively fining shortest paths.
For more details, please check out our paper



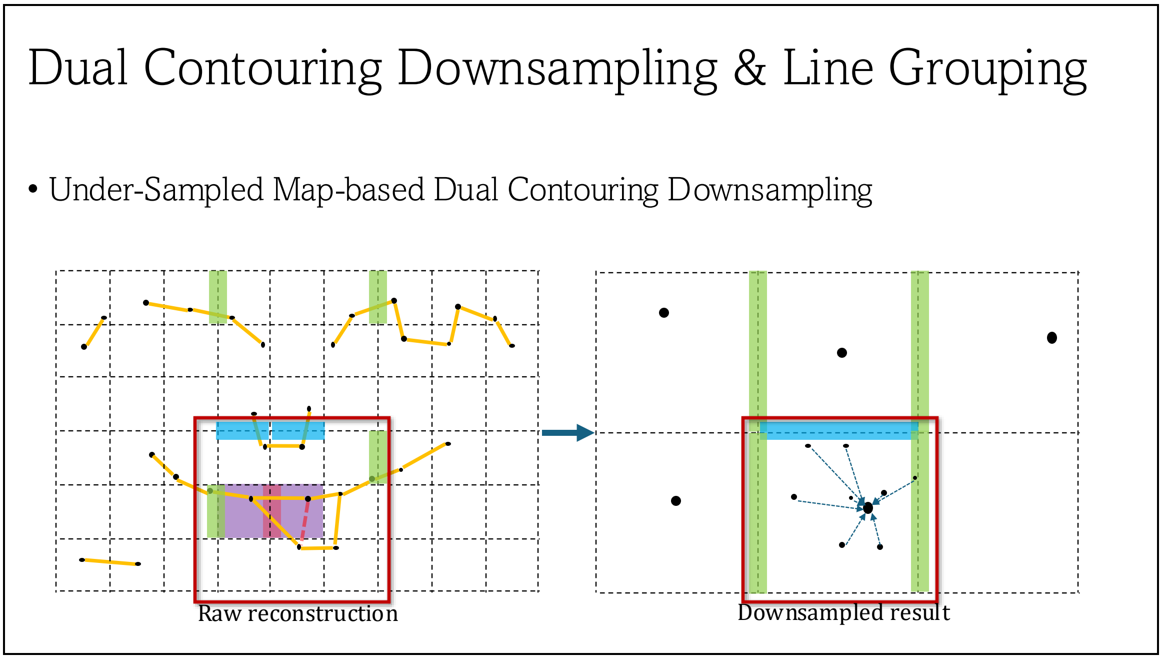
Dual Contouring Downsampling & Line Grouping

- Under-Sampled Map-based Dual Contouring Downsampling

Raw reconstruction

Downsampled result

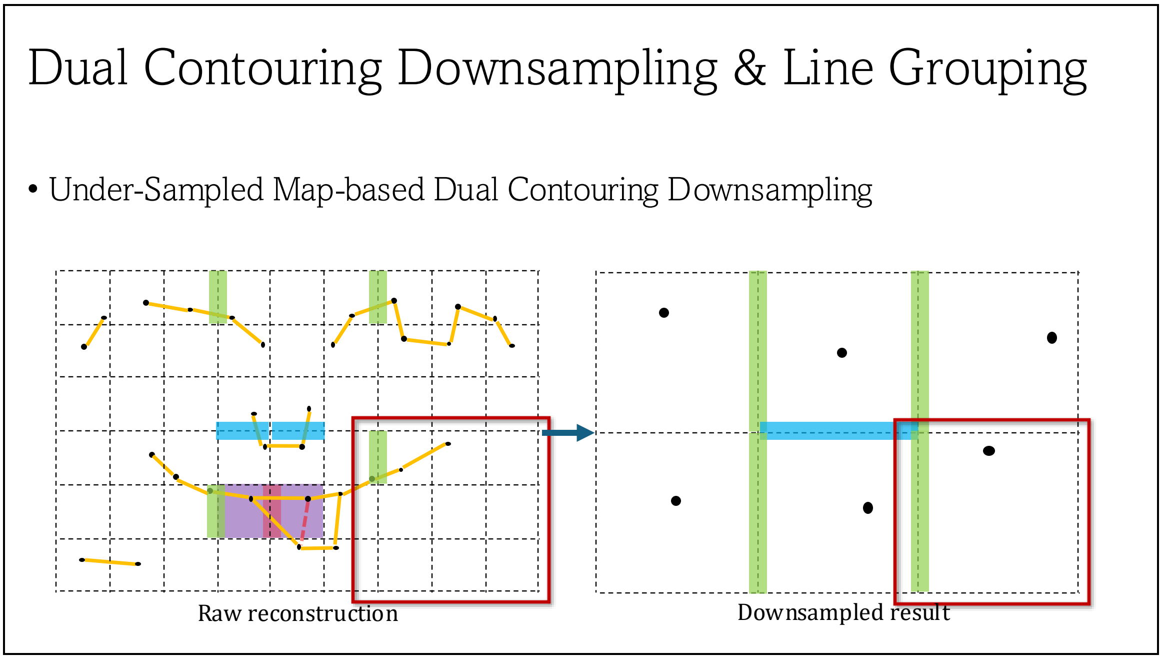
Then to further reduce the number of unnecessary vector segments, we proposed an under-sampled map based dual contouring down sampling method
To turn vector segments into strokes, we proposed a simple line grouping method that iteratively fining shortest paths.
For more details, please check out our paper



Dual Contouring Downsampling & Line Grouping

- Under-Sampled Map-based Dual Contouring Downsampling

Then to further reduce the number of unnecessary vector segments, we proposed an under-sampled map based dual contouring down sampling method
To turn vector segments into strokes, we proposed a simple line grouping method that iteratively fining shortest paths.
For more details, please check out our paper



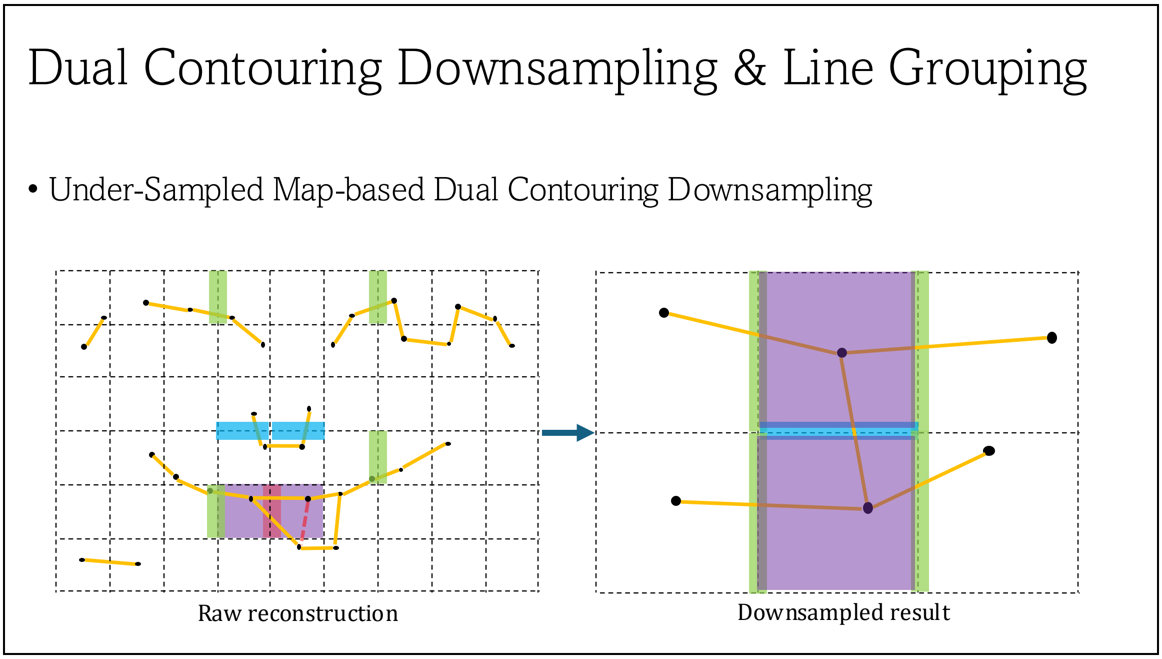
Dual Contouring Downsampling & Line Grouping

- Under-Sampled Map-based Dual Contouring Downsampling

Raw reconstruction

Downsampled result

Then to further reduce the number of unnecessary vector segments, we proposed an under-sampled map based dual contouring down sampling method
To turn vector segments into strokes, we proposed a simple line grouping method that iteratively fining shortest paths.
For more details, please check out our paper



Dual Contouring Downsampling & Line Grouping

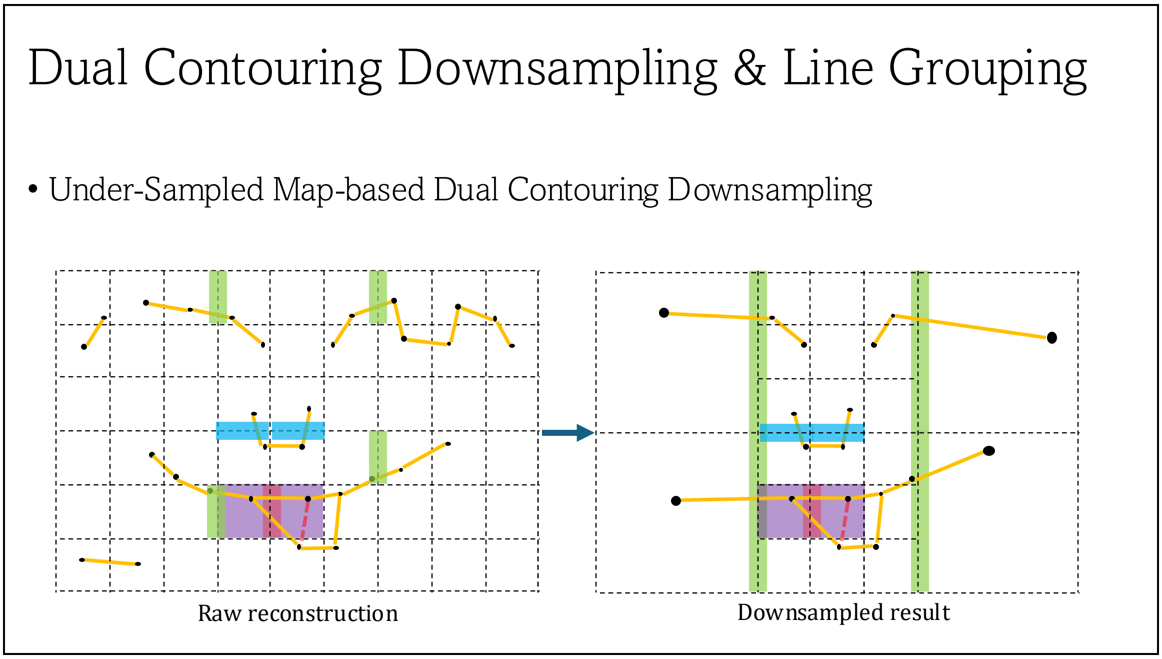
- Under-Sampled Map-based Dual Contouring Downsampling

Raw reconstruction

Downsampled result

Then to further reduce the number of unnecessary vector segments, we proposed an under-sampled map based dual contouring down sampling method To turn vector segments into strokes, we proposed a simple line grouping method that iteratively finding shortest paths.

For more details, please check out our paper



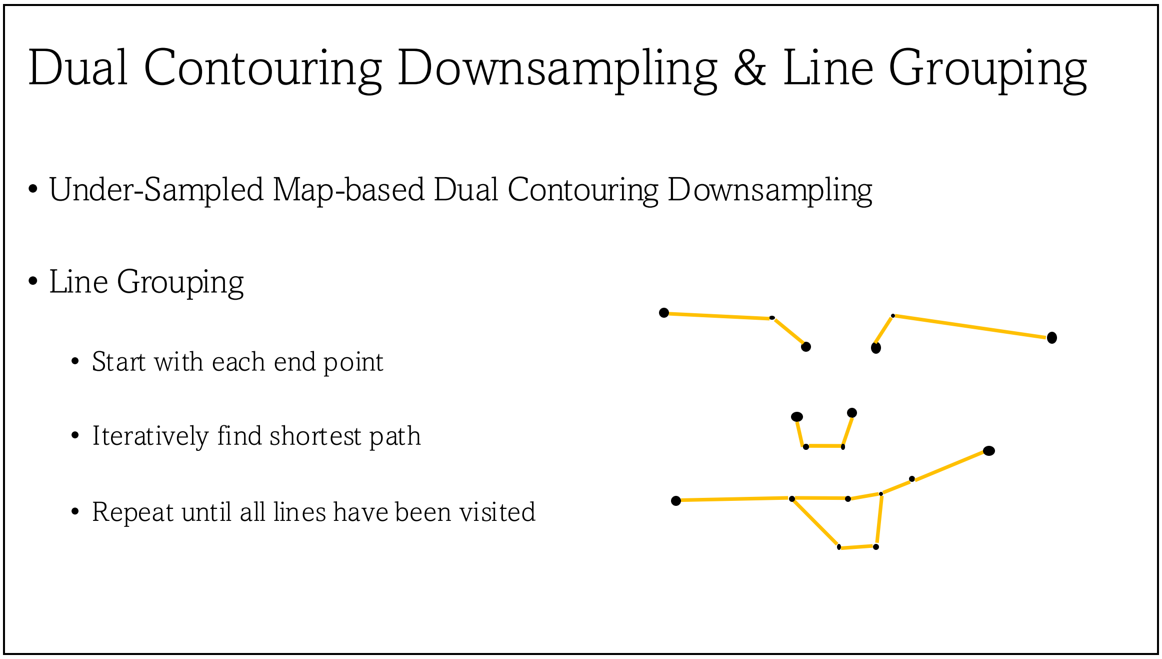
Dual Contouring Downsampling & Line Grouping

- Under-Sampled Map-based Dual Contouring Downsampling

Raw reconstruction

Downsampled result

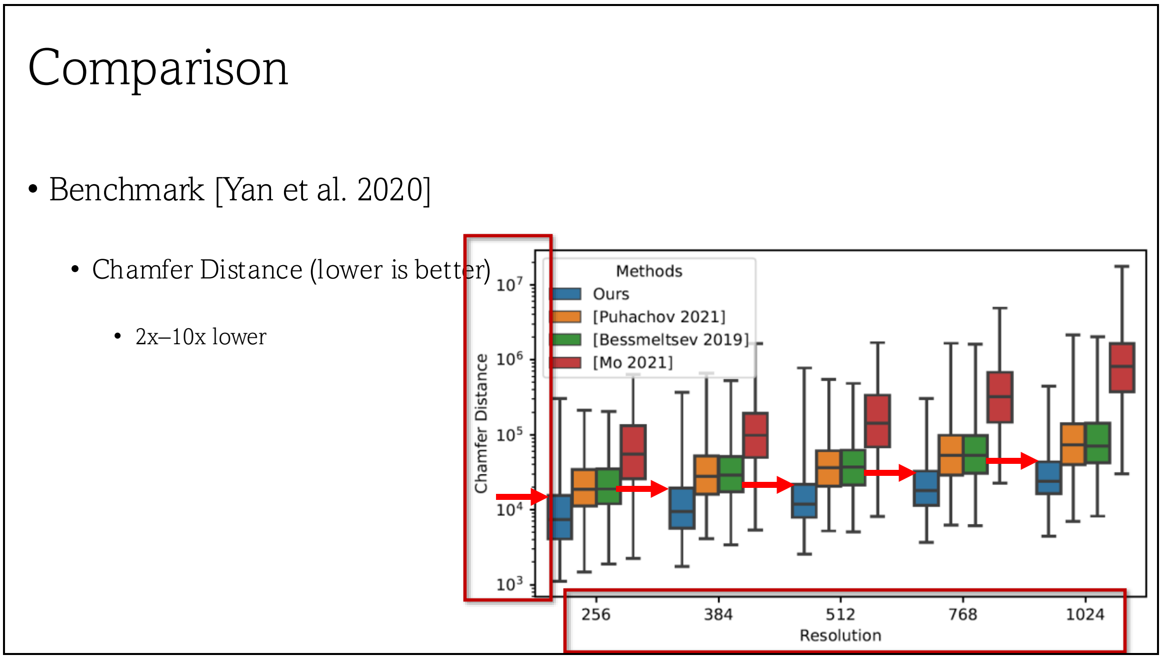
Then to further reduce the number of unnecessary vector segments, we proposed an under-sampled map based dual contouring down sampling method
To turn vector segments into strokes, we proposed a simple line grouping method that iteratively finding shortest paths.
For more details, please check out our paper



Dual Contouring Downsampling & Line Grouping

- Under-Sampled Map-based Dual Contouring Downsampling
- Line Grouping
  - Start with each end point
  - Iteratively find shortest path
  - Repeat until all lines have been visited

Then to further reduce the number of unnecessary vector segments, we proposed an under-sampled map based dual contouring down sampling method
To turn vector segments into strokes, we proposed a simple line grouping method that iteratively finding shortest paths.
For more details, please check out our paper



Comparison

- Benchmark [Yan et al. 2020]
  - Chamfer Distance (lower is better)
    - 2x–10x lower

We used our sketch clean up benchmark to create a vectorization test set. The test set contains 369 clean raster sketches with their vector ground truth.

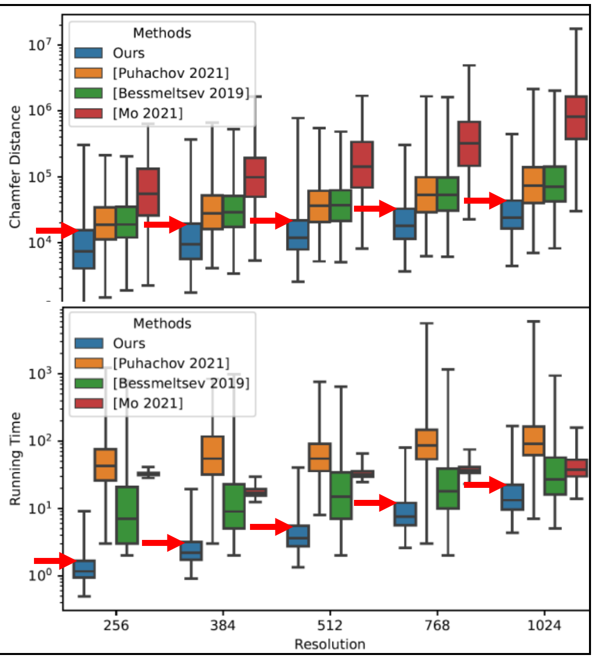
Then we compared our method with others using metrics such as chamfer distance, running time under 6 different input sketch resolution levels.

Please note that all metrics along the y–axis are log scaled

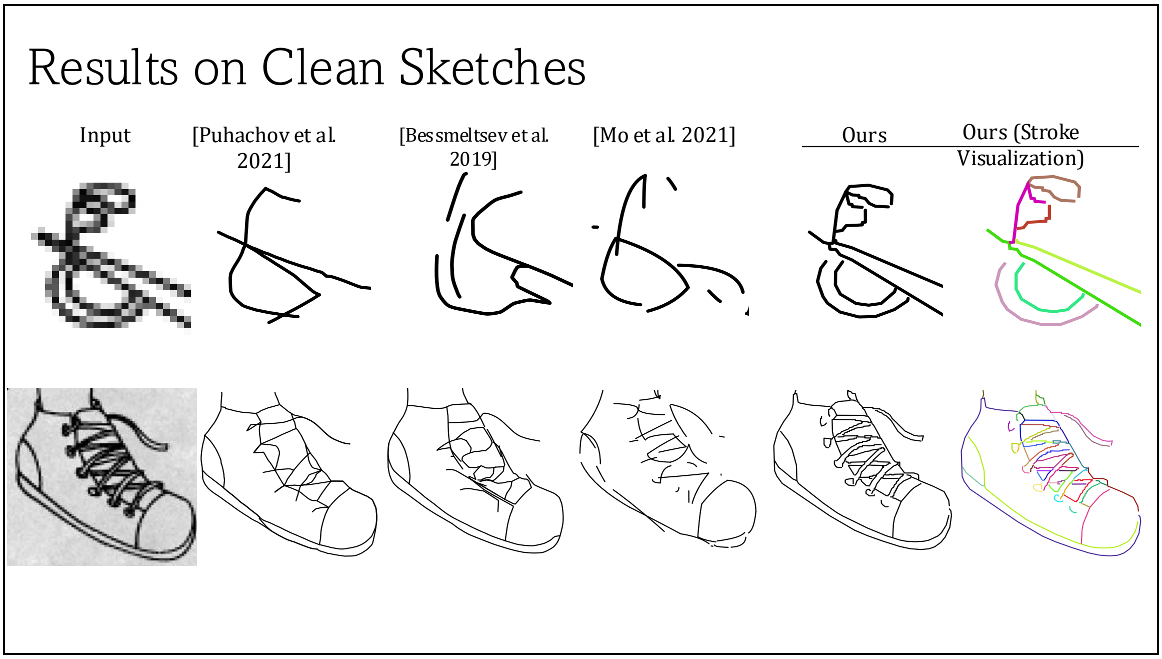
The experiment show our method consistently performs better and run faster than all other methods.

Comparison

- Benchmark [Yan et al. 2020]
  - Chamfer Distance (lower is better)
    - 2x–10x lower
  - Running Time
    - 2x–10x faster

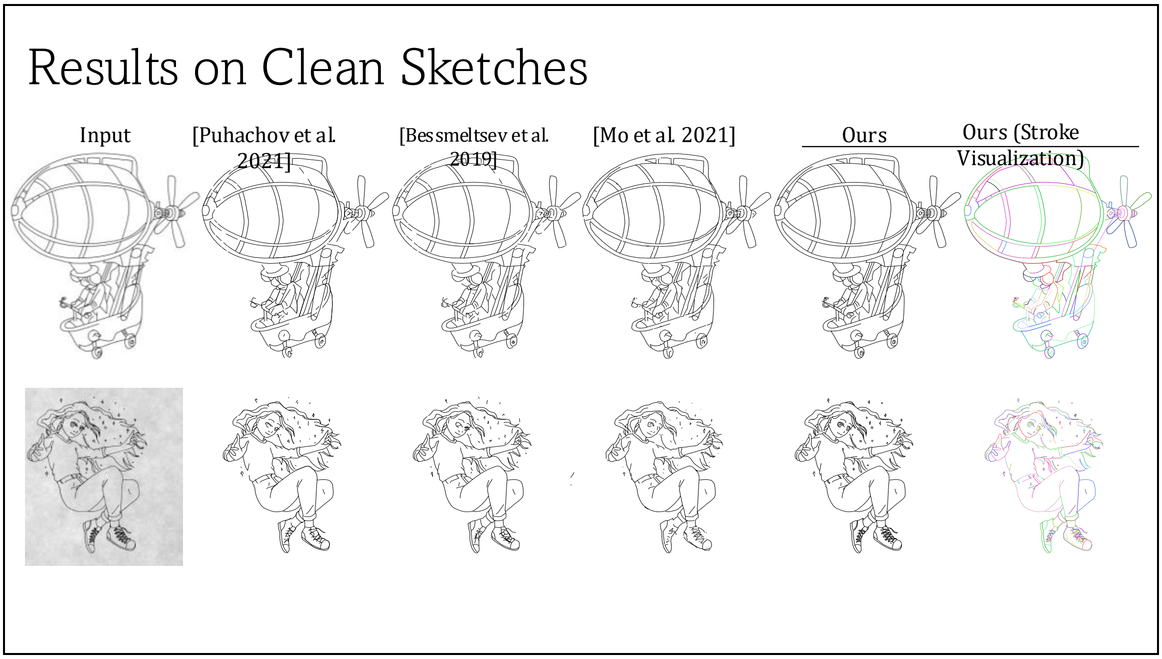


We used our sketch clean up benchmark to create a vectorization test set. The test set contains 369 clean raster sketches with their vector ground truth.
Then we compared our method with others using metrics such as chamfer distance, running time under 6 different input sketch resolution levels.
The experiment show our method consistently performs better and run faster than all other methods.



Results on Clean Sketches

Here we show a comparison of vectorization results
As we can see our method faithfully captured much more detail for even a tiny portion of the raster sketch



Results on Clean Sketches

Input

[Puhachov et al. 2021]

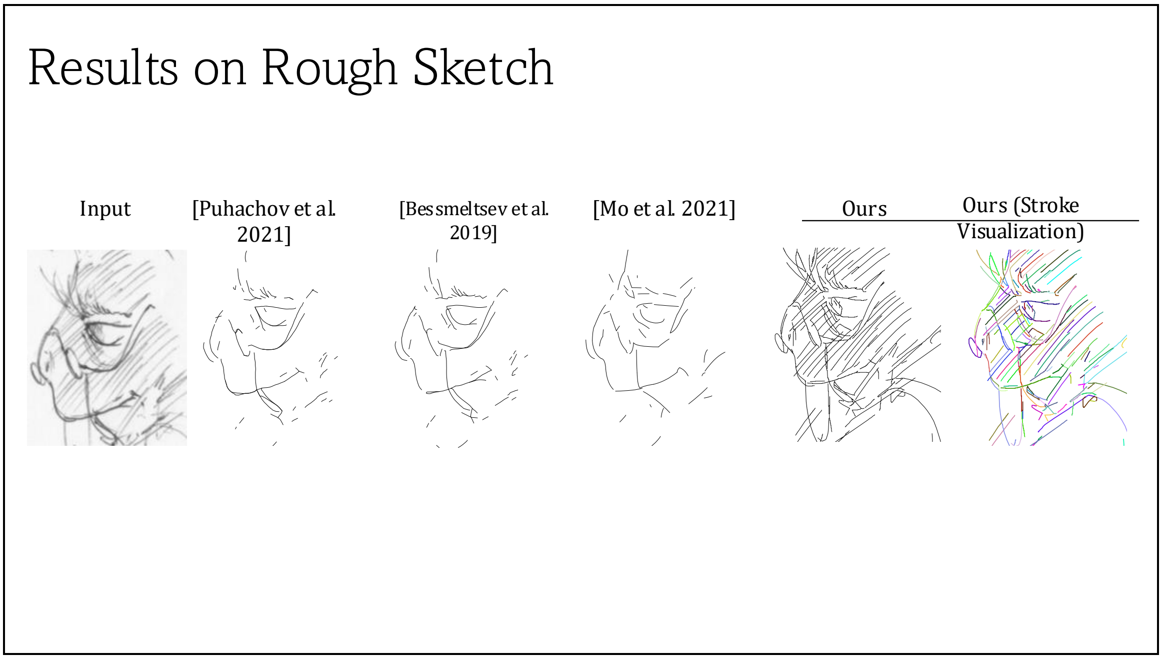
[Bessmeltsev et al. 2019]

[Mo et al. 2021]

Ours

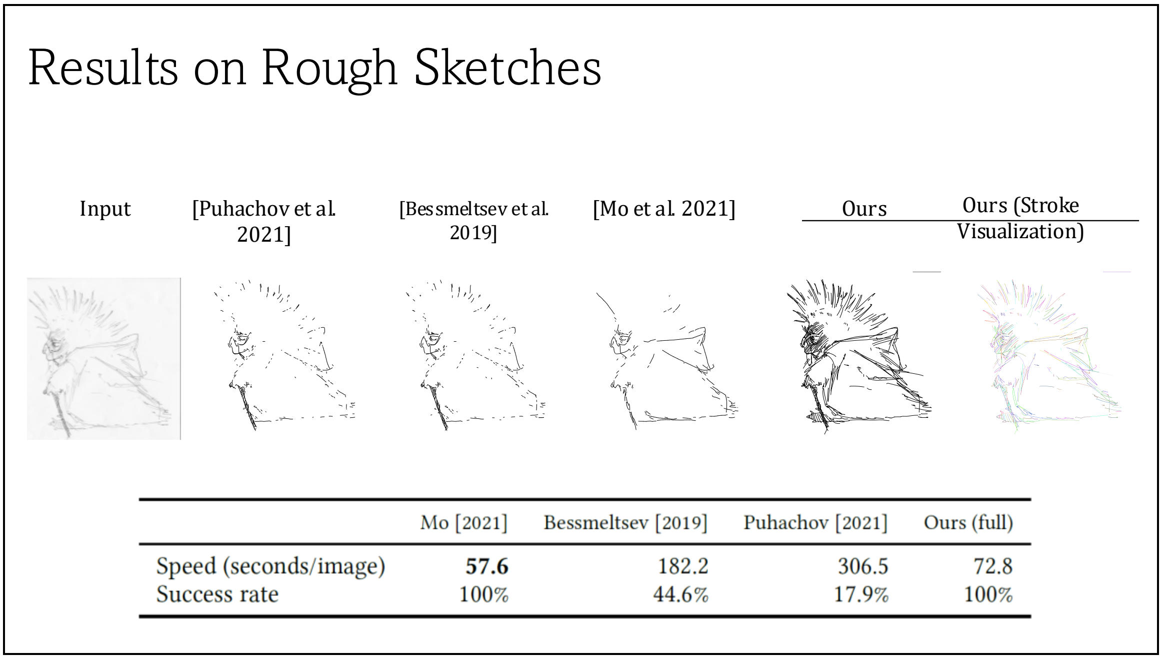
Ours (Stroke Visualization)

Here we show a comparison of vectorization results
As we can see our method faithfully captured much more detail for even a tiny portion of the raster sketch



Results on Rough Sketch

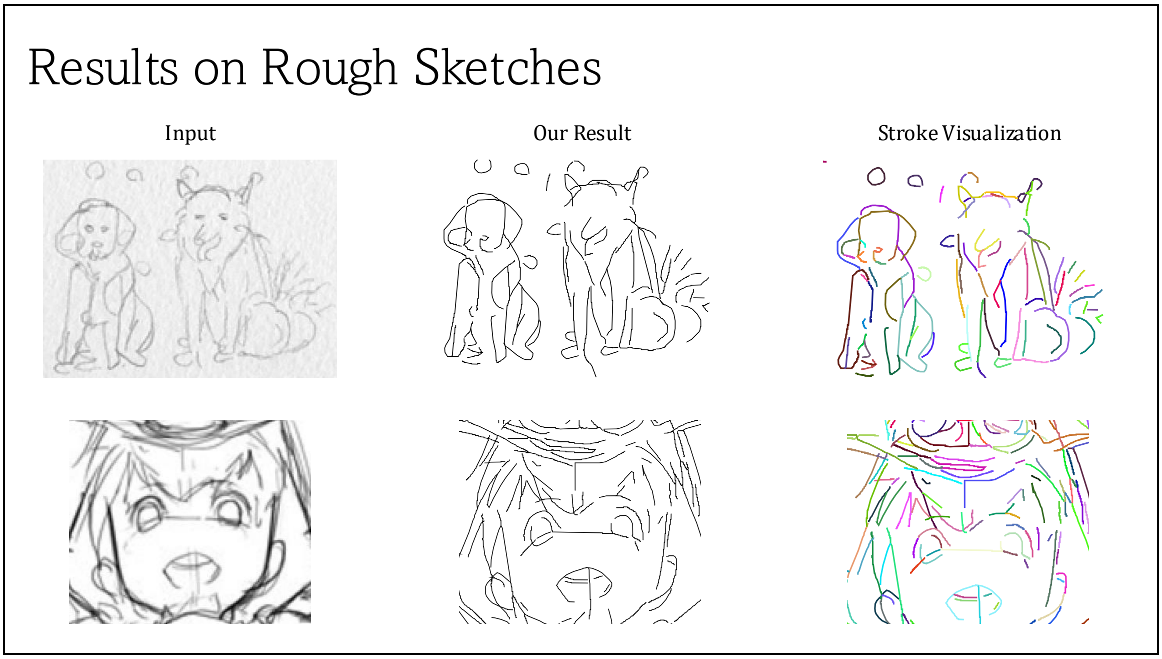
We also tested all methods with another test set which contains 112 rough sketches in the wild, this comparison further shows our method's capacity of capturing stroke details.



Results on Rough Sketches

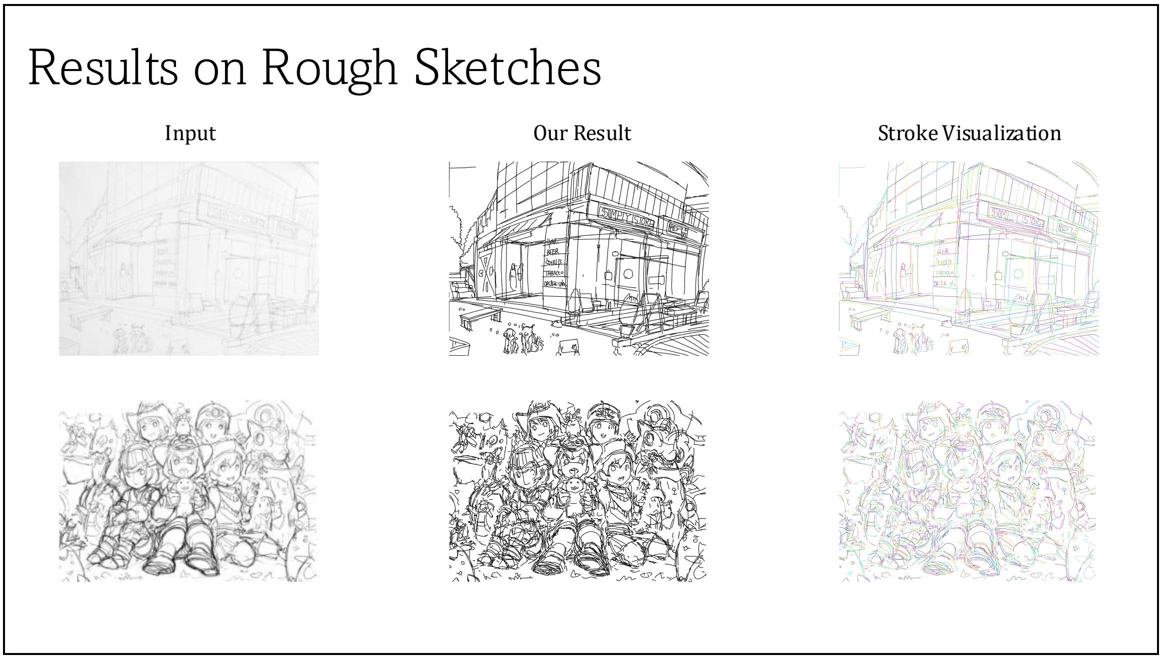
	Mo [2021]	Bessmeltsev [2019]	Puhachov [2021]	Ours (full)
Speed (seconds/image)	**57.6**	182.2	306.5	72.8
Success rate	100%	44.6%	17.9%	100%

It is also worth mentioning that our method could successfully vectorize all rough sketches with faster speed compared to the frame filed based methods



Results on Rough Sketches

Here are two more vectorization results on complex rough sketches that only our method could vectorize. Please note that the zoom in region is only about 1/50 to 1/25 of the full sketch.



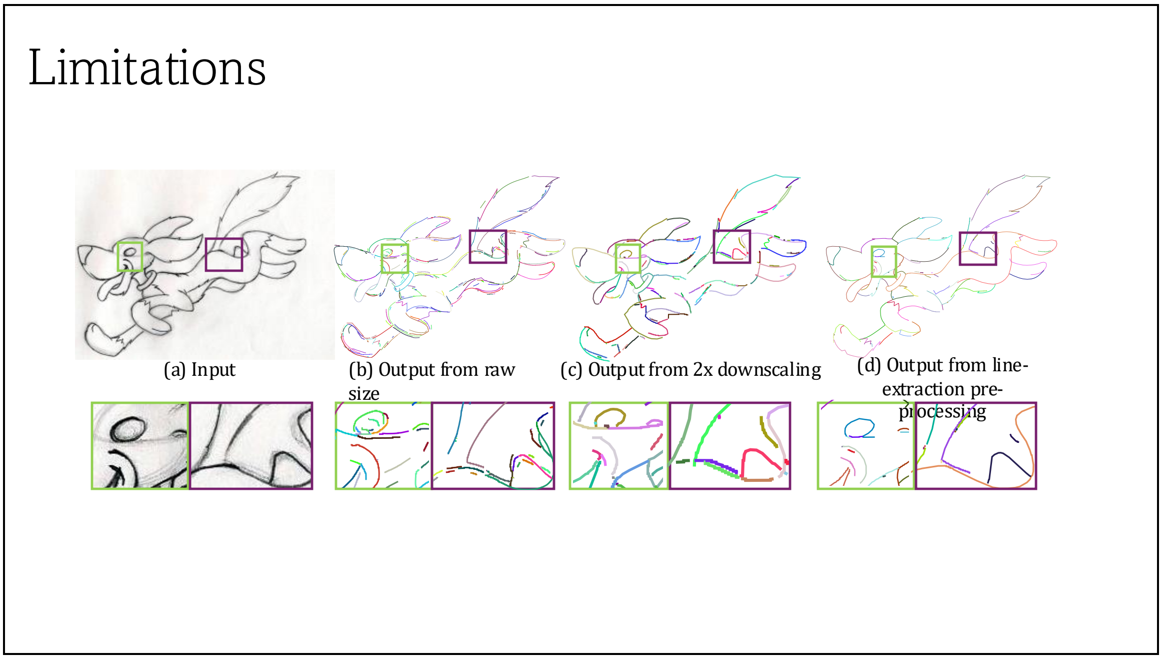
Results on Rough Sketches

Input

Our Result

Stroke Visualization

Here are two more vectorization results on complex rough sketches that only our method could vectorize. Please note that the zoom in region is only about 1/50 to 1/25 of the full sketch.



Limitations

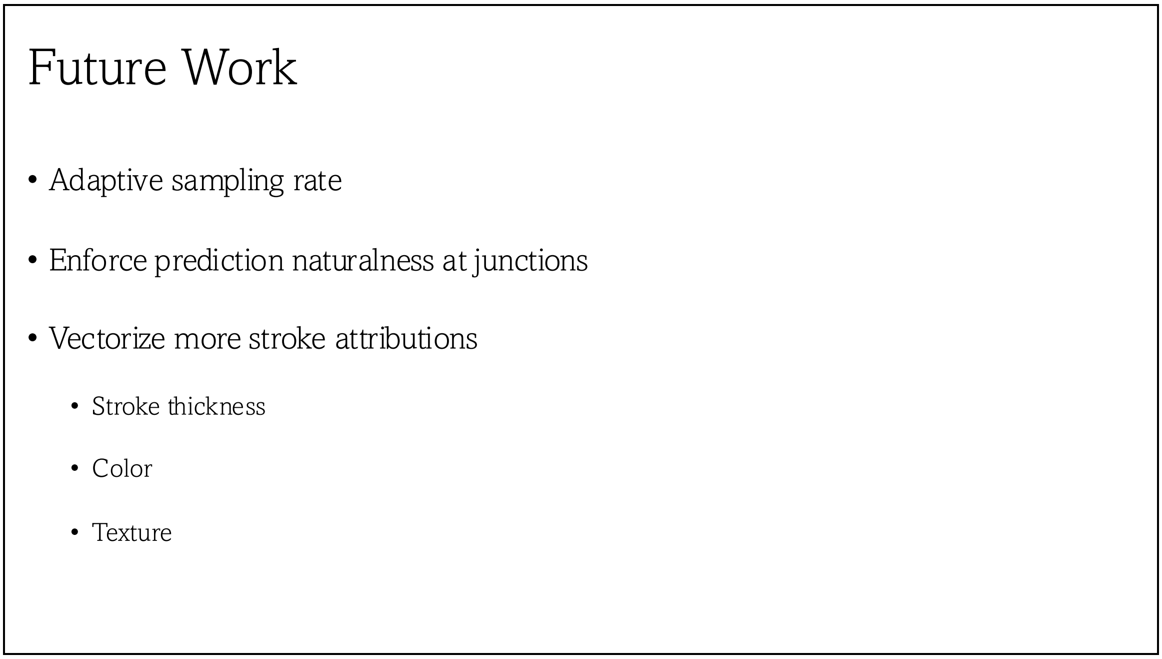
(a) Input

(b) Output from raw size

(c) Output from 2x downscaling

(d) Output from line-extraction pre-processing

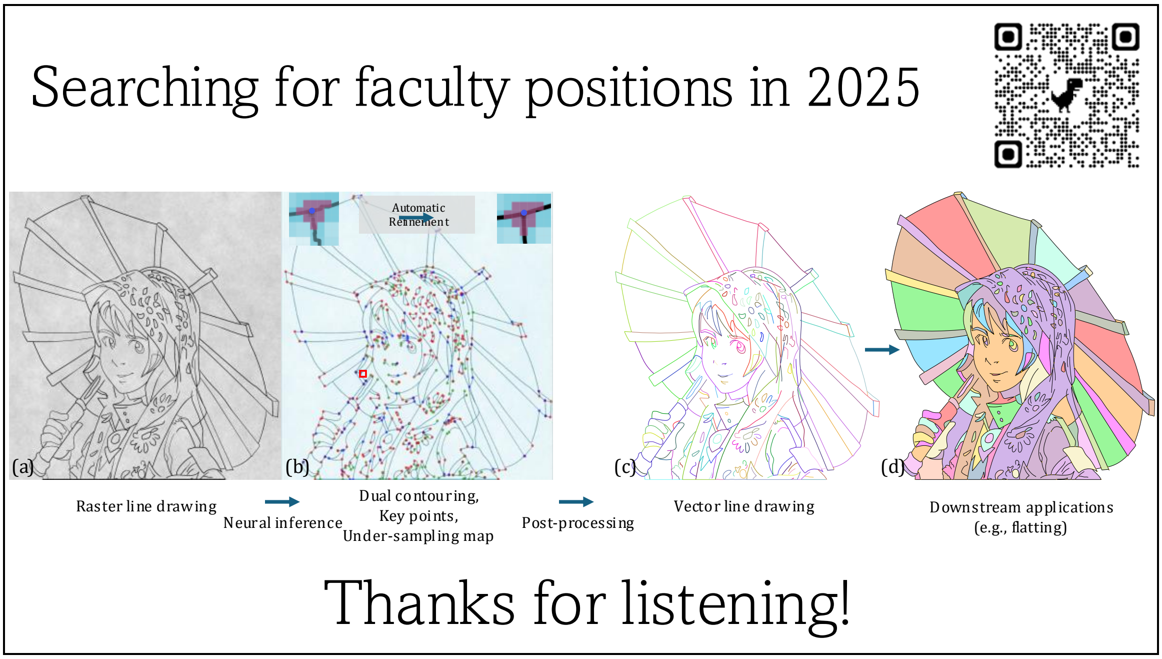
Here is one failure case that shows the limitations of our method.
Our methods will output broken and messy lines when the input sketch contains densely repeated strokes or very thick strokes.
Although this vectorization result could be improved by either down sample the input raster sketch or apply pre-processing such as line extraction.
We believe the fundamental reason for this limitation should be our pair-wised training strategy which makes our network struggling when predicting the center lines and junctions for those type of strokes.



Future Work

- Adaptive sampling rate
- Enforce prediction naturalness at junctions
- Vectorize more stroke attributions
  - Stroke thickness
  - Color
  - Texture

In the future, we can consider an adaptive sampling rate, enforce more natural junctions, and vectorize more stroke attributes.



Searching for faculty positions in 2025

Raster line drawing → Neural inference → Dual contouring, Key points, Under-sampling map → Post-processing → Vector line drawing → Downstream applications (e.g., flatting)

Thanks for listening!

At last, I want to say thanks for all my collaborators and thank you for your listening.