MLSEB: Edge Bundling using Moving Least Squares Approximation

Jieting Wu, Jianping Zeng, Feiyu Zhu, Hongfeng Yu

Department of Computer Science & Engineering

University of Nebraska-Lincoln, Lincoln, Nebraska

Outline

- Motivation
- Background
- Approach
- Results
- Conclusion

Motivation

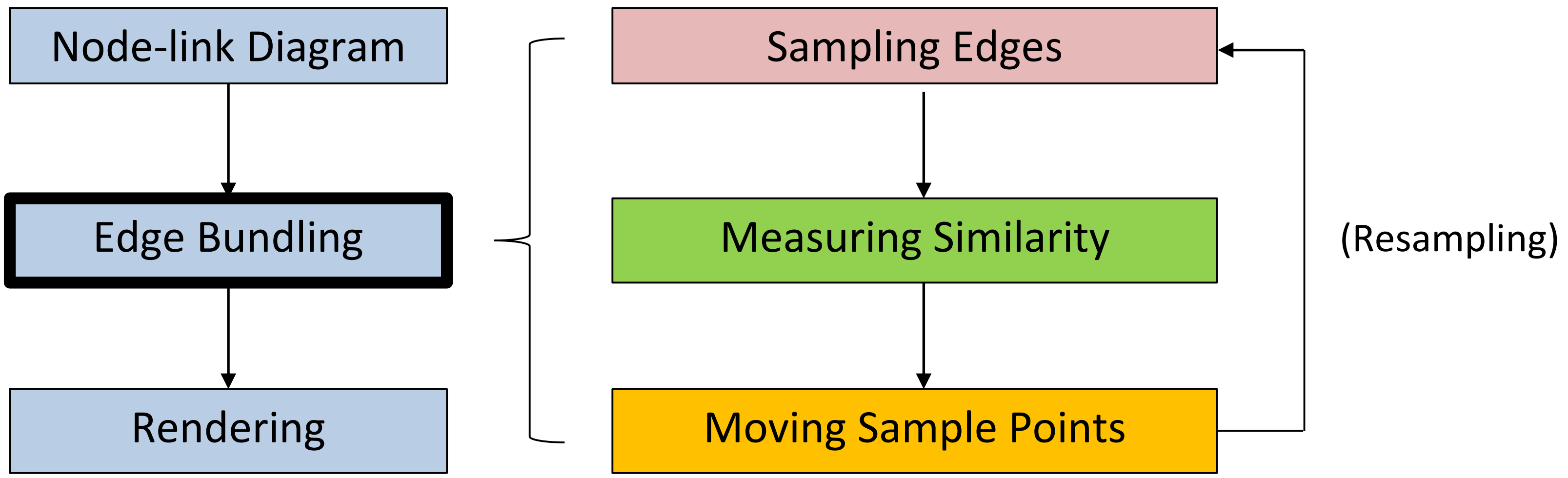
- State-of-the-art graph visualization
  - Node-Link diagram
    - Pro
      - Simple and intuitive
    - Con
      - Easily incur visual clutter

  - Edge bundling
    - Pros
      - Effectively remedy visual clutter
      - Reveal high-level graph structures
    - Cons
      - High complexity
      - Non-trivial quality evaluation

FFTEB [Lhuillier2017]

Background

- Edge bundling algorithms
  - Visually merge edges based on similarity measurements
    - Iterative refinement

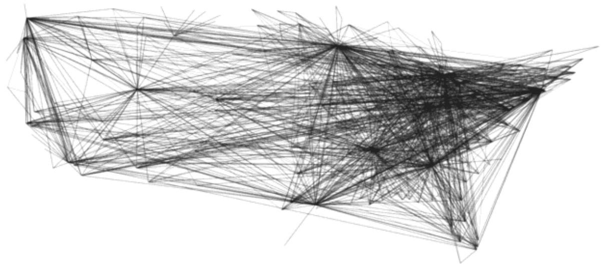


Background

- Force-directed edge bundling [Holten2010]
- Kernel density estimation (KDE) based methods
  - KDEEB: Graph Bundling by Kernel Density Estimation [Hurter2012]
  - CUBu: CUDA Universal Bundling [Matthew van der Zwan2016]
  - FFTEB: Fast Fourier Transform Edge Bundling [Lhuillier2017]

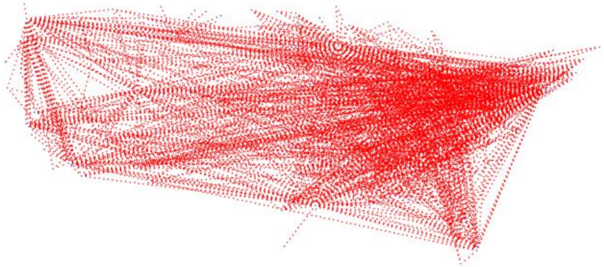
Background

- Kernel density estimation (KDE) based methods



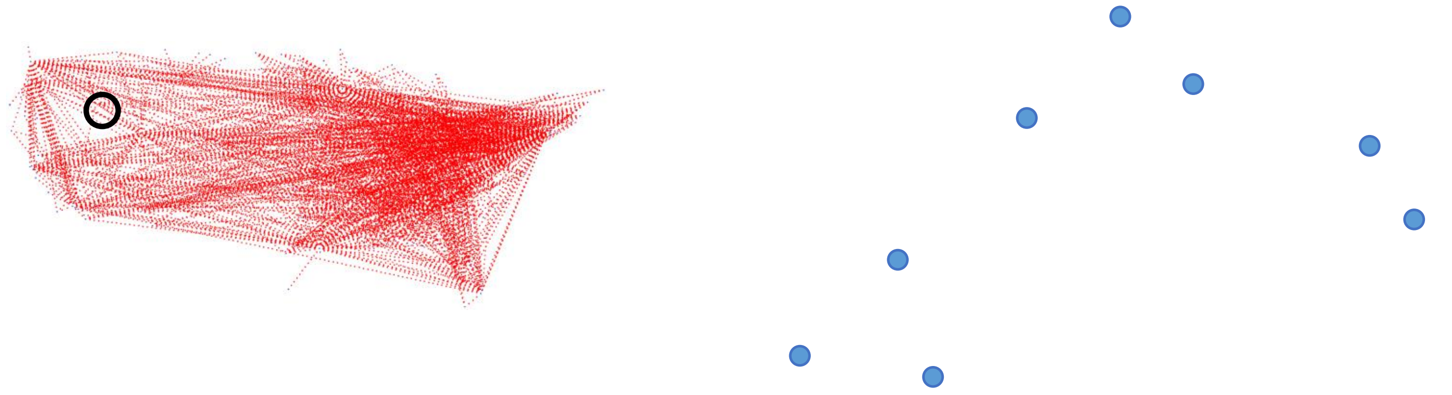
Background

- Kernel density estimation (KDE) based methods
  - Image-based sampling



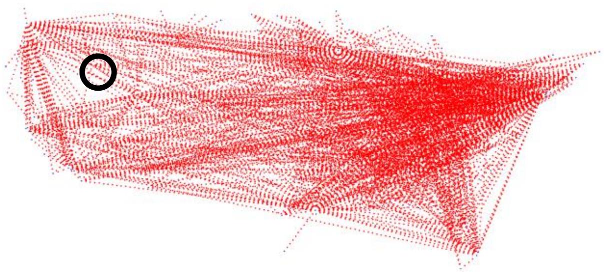
Background

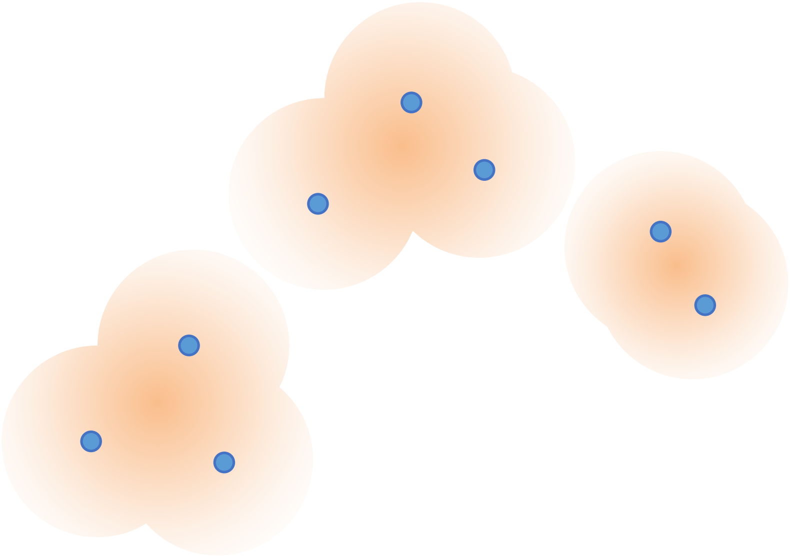
- Kernel density estimation (KDE) based methods
  - Mean-shift clustering



Background

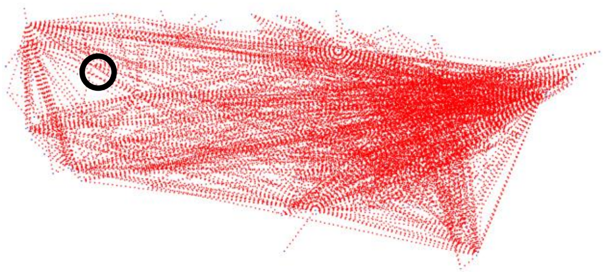
- Kernel density estimation (KDE) based methods
  - Mean-shift clustering
    - Kernel density estimation

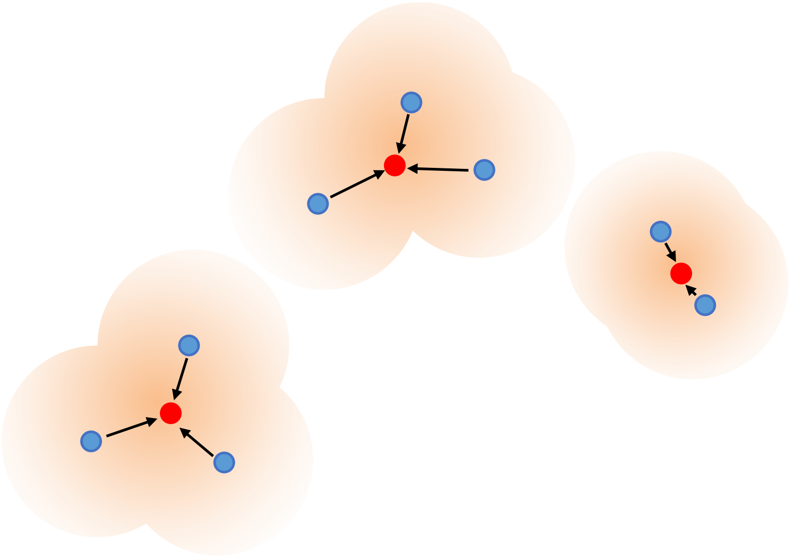




Background

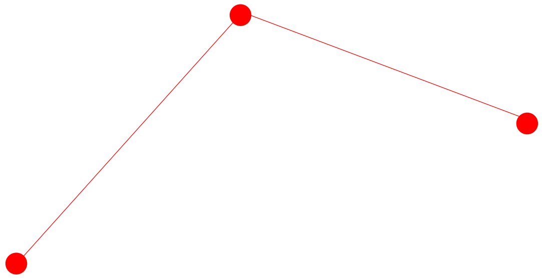
- Kernel density estimation (KDE) based methods
  - Mean-shift clustering
    - Kernel density estimation
    - Gradient-based advection





Background

- Kernel density estimation (KDE) based methods
  - Incur excessive convergence artifact
    - Require resampling to avoid excessive convergence

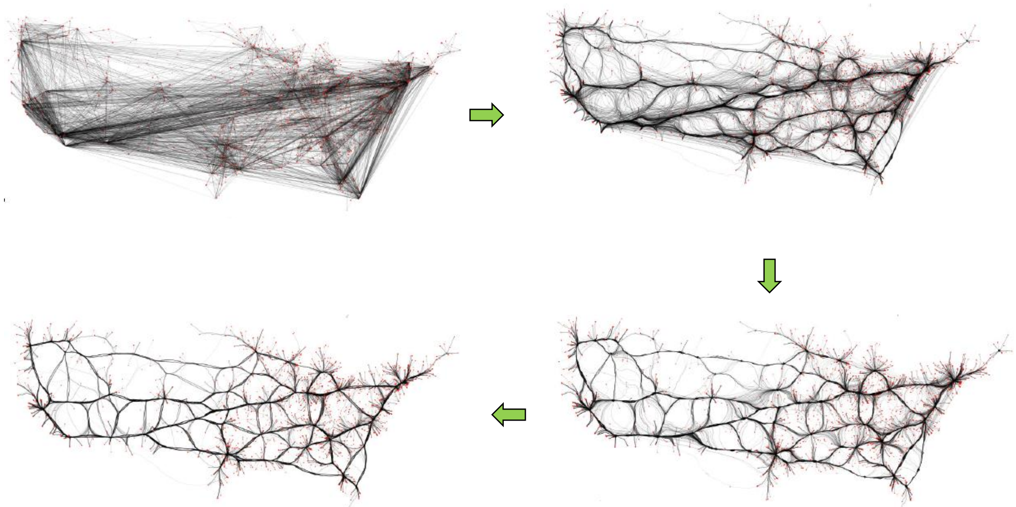


Background (Complexity)

- Kernel density estimation (KDE) based methods
  - Image-based sampling
  - Mean-shift clustering
  - Iterative refinement (resampling)

Complexity: O(SNI + IE)

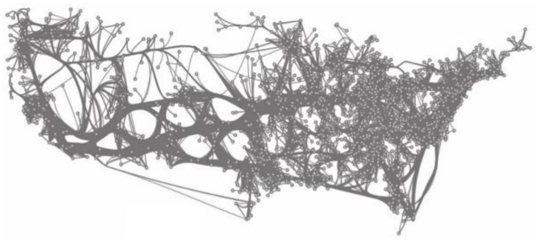
S: sample points
N: image pixel number
I: iteration number
E: edge number



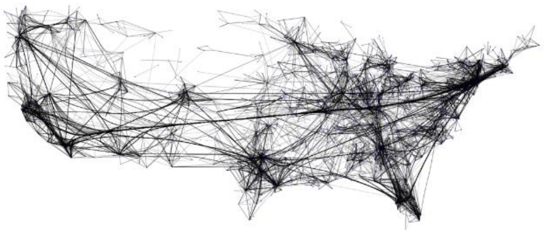
Graph Bundling by Kernel Density Estimation [Hurter2012]

Examples of Existing Edge Bundling Methods

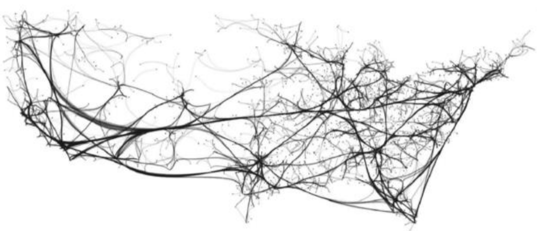
Original graph



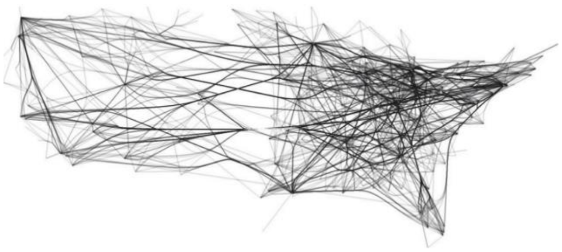
GBEB [Cui2008]



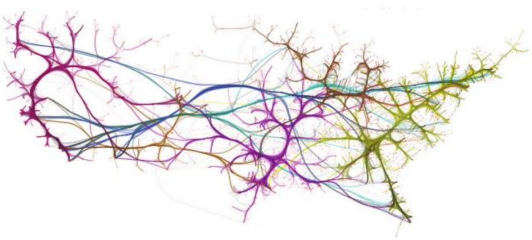
FDEB [Holten2010]



WR [Lambert2010]

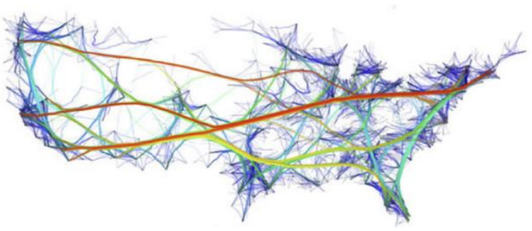


MINGLE [Ganser2011]



SBEB [Ersoy2012]

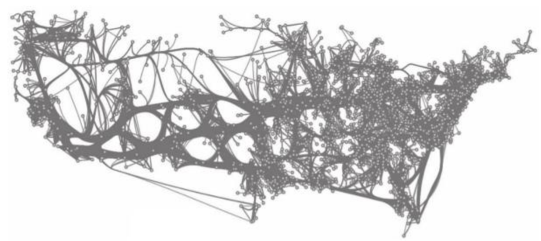
KDEEB [Hurter2012]



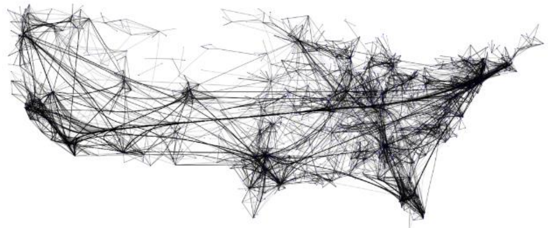
CUBu [Matthew van der Zwan2016]

FFTEB [Lhuillier2017]

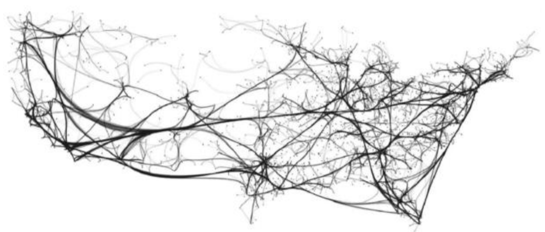
Evaluation?



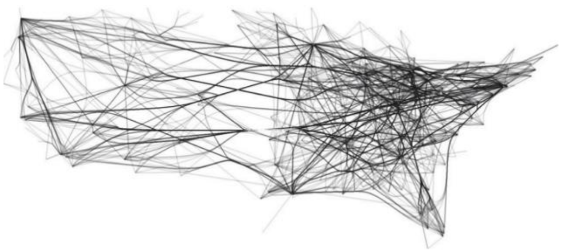
GBEB [Cui2008]



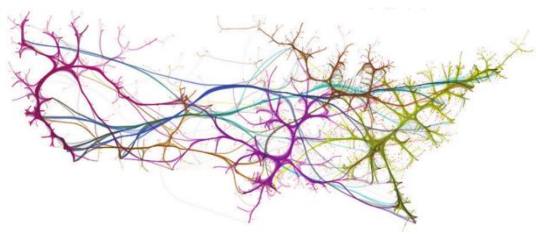
FDEB [Holten2010]



WR [Lambert2010]



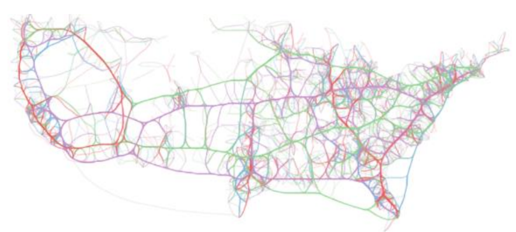
MINGLE [Ganser2011]



SBEB [Ersoy2012]

KDEEB [Hurter2012]

CUBu [Matthew van der Zwan2016]



FFTEB [Lhuillier2017]

Evaluation

- Quality of edge bundling
  - Lhuillier et al. [Lhuillier2017] suggested to use the ratio of clutter reduction to amount of distortion to quantify the quality of a bundled graph

$$Q = \frac{C}{T}$$

- C: clutter reduction
- T: amount of distortion

Evaluation

- Quality of edge bundling
  - T: The distortion is measured by computing the distance between original edge drawings and the bundled edge drawings
  - C: The calculation of clutter reduction has not been fully concluded in the existing work

Evaluation

- Quality of edge bundling
  - We propose to employ the reduction of the used pixel number in a graph drawing to measure C

$$C = \Delta P = P - P'$$

  - We also propose to use the average distortion $\bar{T}$, instead of the total distortion of all the sample points

$$\bar{T} = \frac{T}{S}$$

T is the total distortion generated

S is the number of sample points

Evaluation

- Quality of edge bundling
  - We have a quality metric to quantify the quality of edge bundling

$$Q = \frac{\Delta P}{\bar{T}}$$

- ΔP: reduced pixels ↑
- $\bar{T}$: average distortion ↓

Evaluation

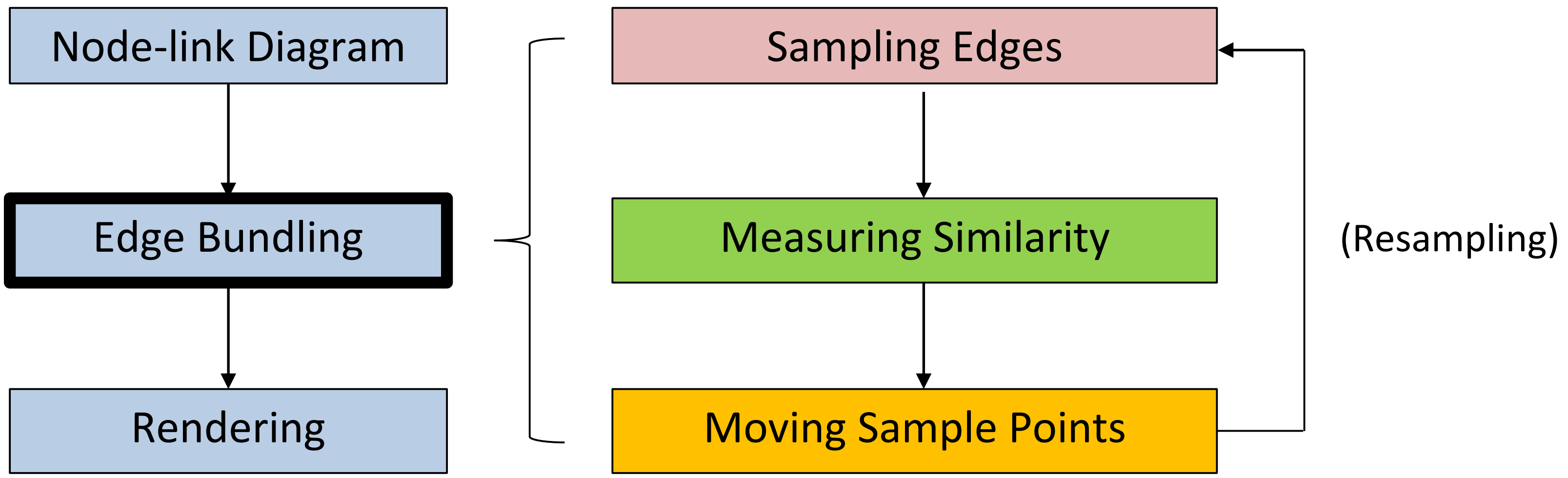
- Quality of edge bundling
  - The pros and cons of the existing methods
    - Pros
      - Create visually appealing edge bundles that reduce clutter
    - Cons
      - Resampling is required in iterative refinement
      - Does not take distortion into their methods

Contribution

- We present MLSEB, a novel method to generate edge bundles based on moving least squares (MLS) approximation
  - Introduce MLS into edge bundling
    - Simplify the edge bundling pipeline
    - Generate **better quality** results compared to other methods
      - Based on the aforementioned quality metric
  - Ensure **scalability** and **efficiency**
    - A set of graphs that range from ten thousand to a half million edges
    - A GPU implementation

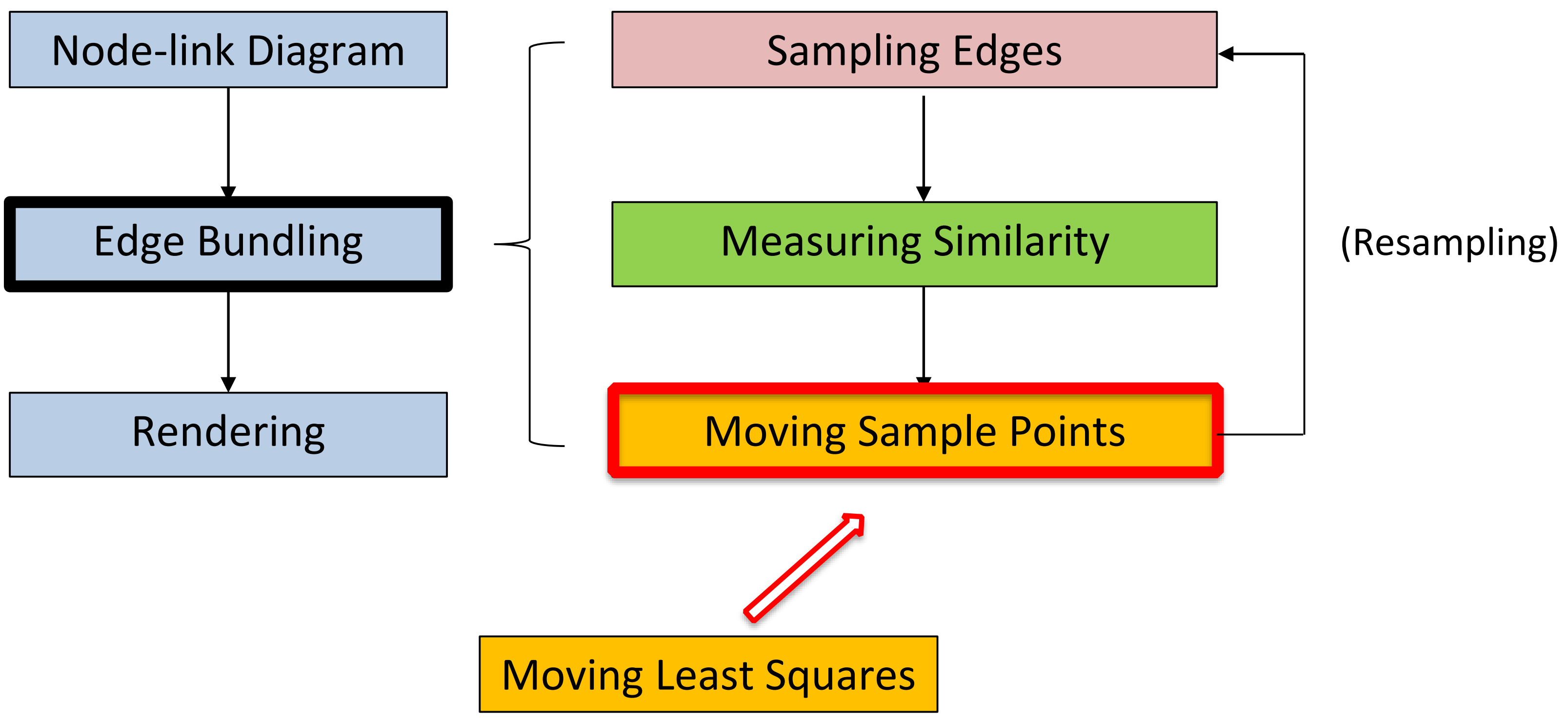
Approach

- The pipeline of moving least squares edge bundling



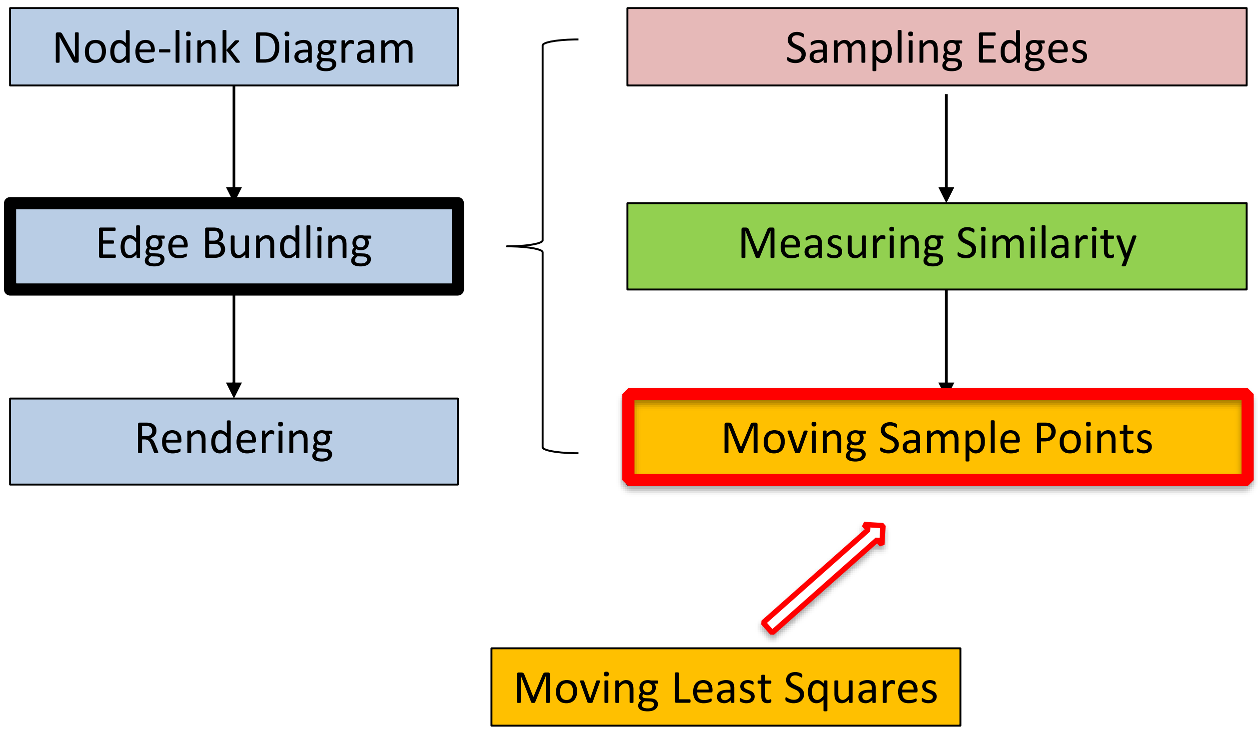
Approach

- The pipeline of moving least squares edge bundling



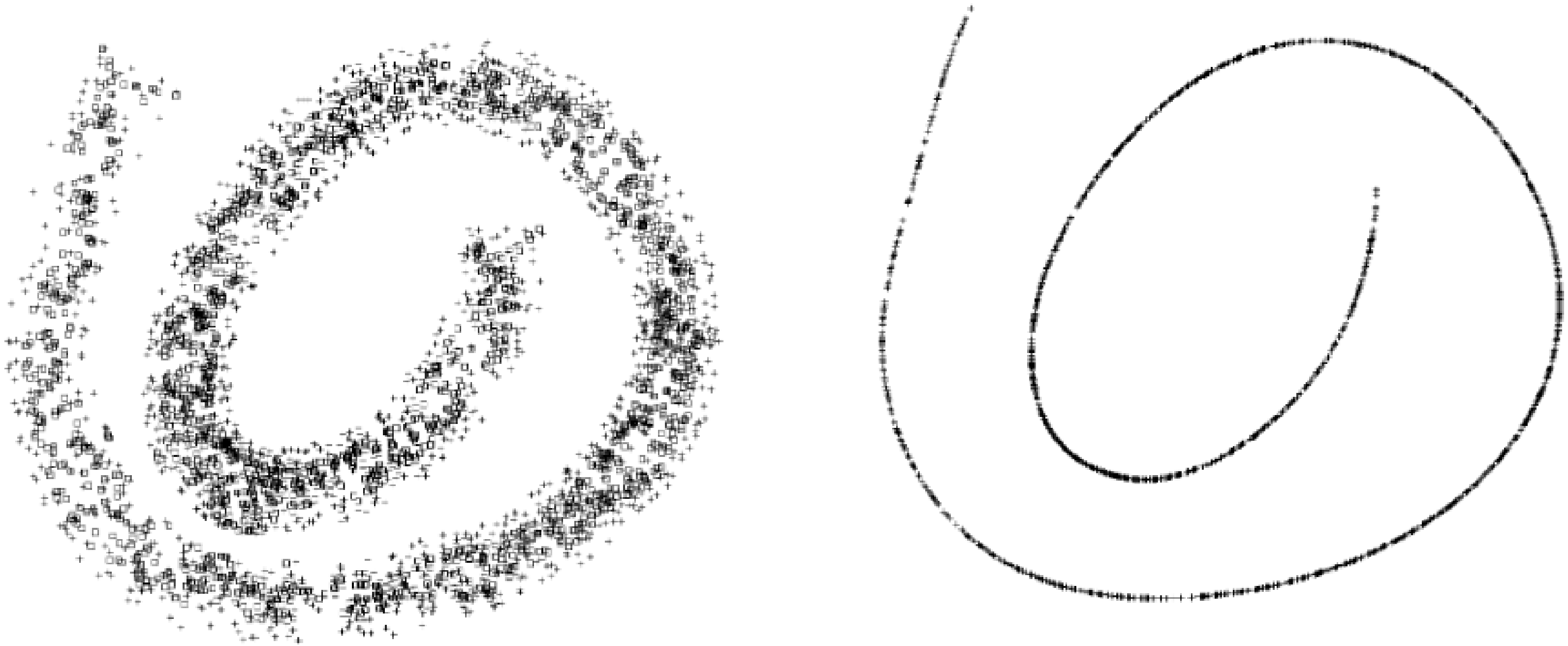
Approach

- The pipeline of moving least squares edge bundling



Approach

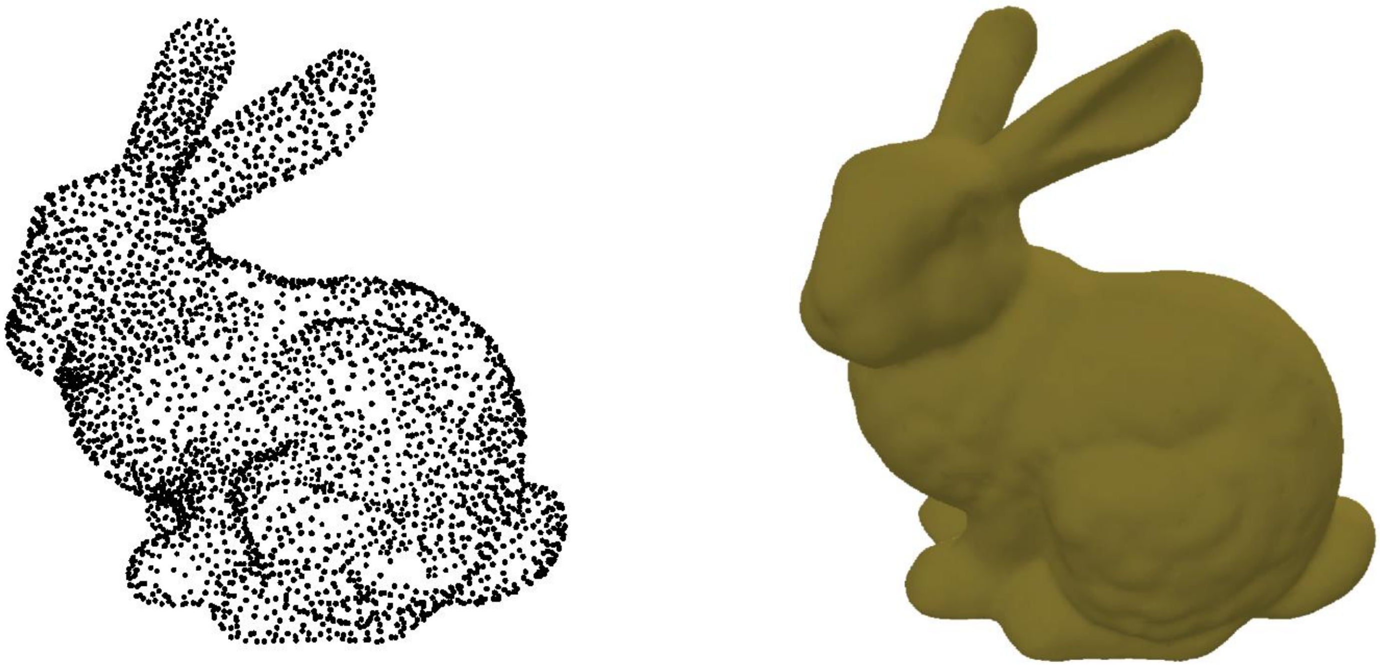
- Moving least squares application
  - Reconstructing continuous functions from a set of unorganized point samples
    - 2D curve reconstruction



Curve Reconstruction from Unorganized Points [Lee00]

Approach

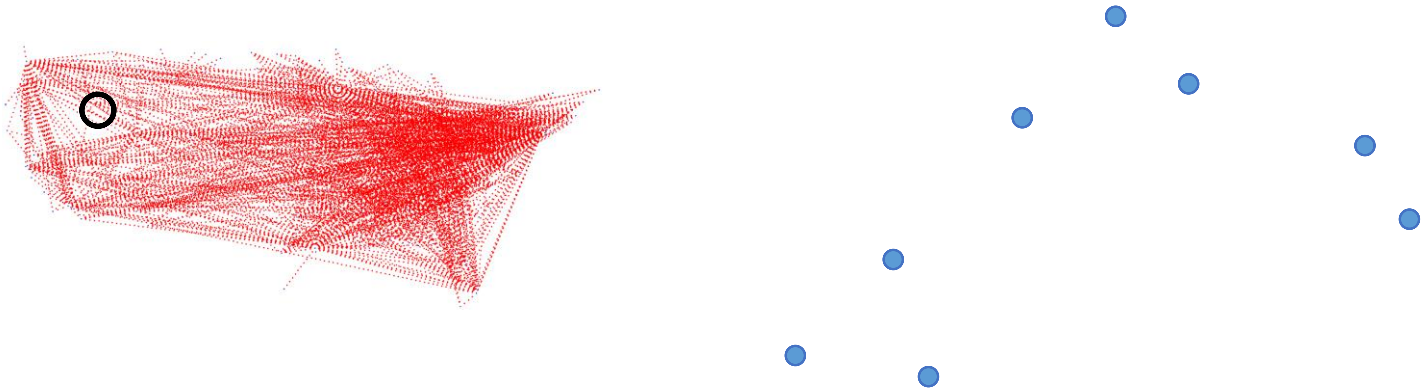
- Moving least squares application
  - Reconstructing continuous functions from a set of unorganized point samples
    - 3D surface reconstruction



Moving Least Squares Multiresolution Surface Approximation [Mederos03]

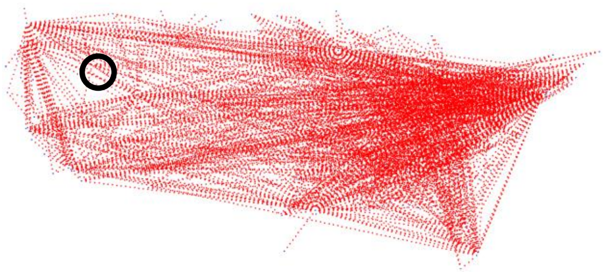
Approach

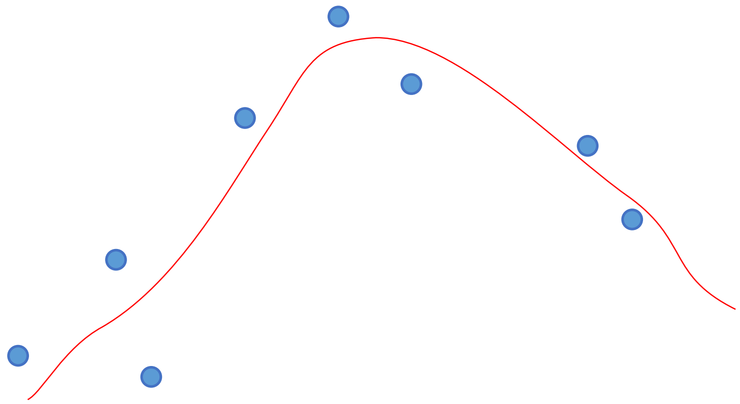
- MLSEB
  - Image-based sampling



Approach

- MLSEB
  - Assume there is an implicit skeleton that is a suitable place to gather sample points and form bundles
    - Skeleton can be interpreted as a curve



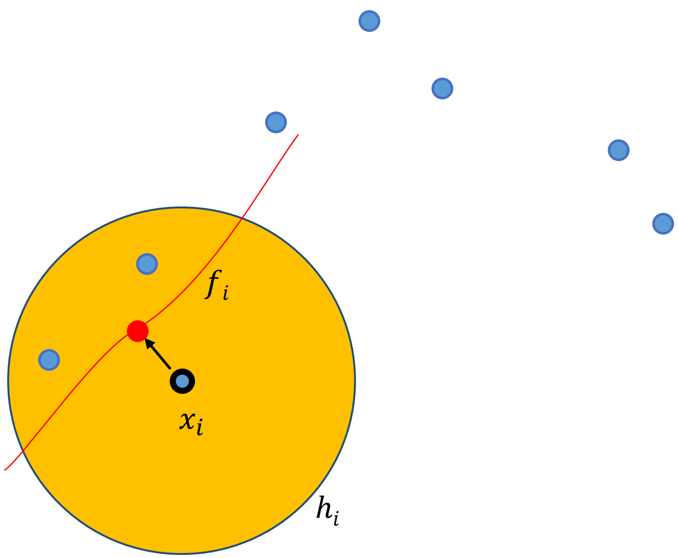


Approach

- MLSEB
  - Skeleton can be interpreted as a piece-wise polynomial curve
    - Calculate f_i by minimizing a weighted least squares error $\boldsymbol{\epsilon}$
      - Within a radial neighborhood h_i of x_i

$$\boldsymbol{\epsilon} = \sum_{j=1}^{h_i} (|x_j - f_i|)^2 \boldsymbol{\theta}(|x_j - x_i|)$$

Least squares approximation

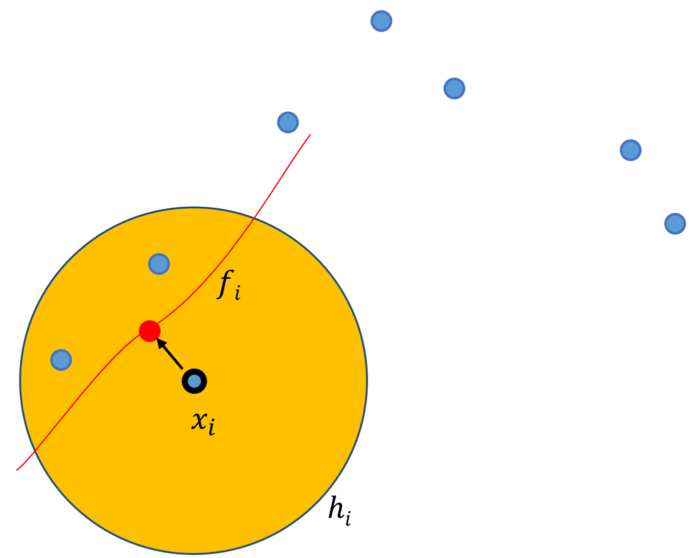


Approach

- MLSEB
  - Skeleton can be interpreted as a piece-wise polynomial curve
    - Calculate f_i by minimizing a weighted least squares error $\boldsymbol{\epsilon}$
      - Within a radial neighborhood h_i of x_i

$$\boldsymbol{\epsilon} = \sum_{j=1}^{h_i} (|x_j - f_i|)^2 \boldsymbol{\theta}(|x_j - x_i|)$$

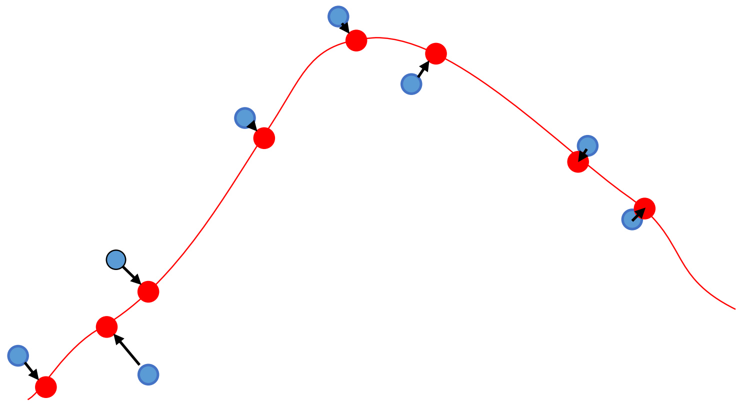
Weighting function:
Gaussian function



Approach

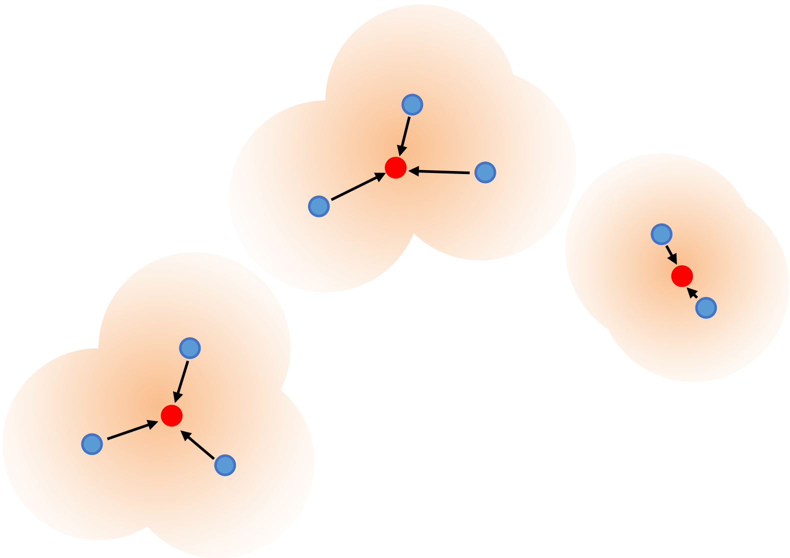
- MLSEB
  - Skeleton can be interpreted as a piece-wise polynomial curve
    - Calculate f_i by minimizing a weighted least squares error $\boldsymbol{\epsilon}$
      - Within a radial neighborhood h_i of x_i
      - Project x_i into f_i

$$\boldsymbol{\epsilon} = \sum_{j=1}^{h_i} (|x_j - f_i|)^2 \boldsymbol{\theta}(|x_j - x_i|)$$

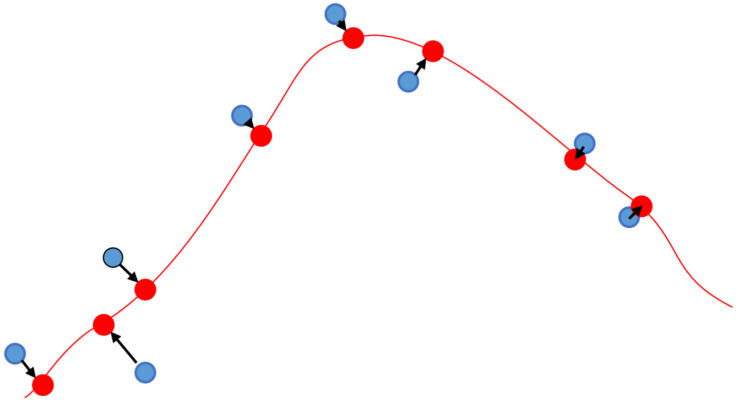


Approach

- MLS vs. KDE



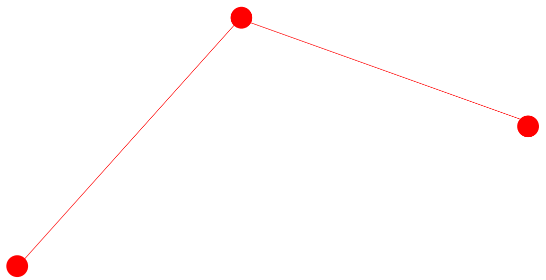
KDE

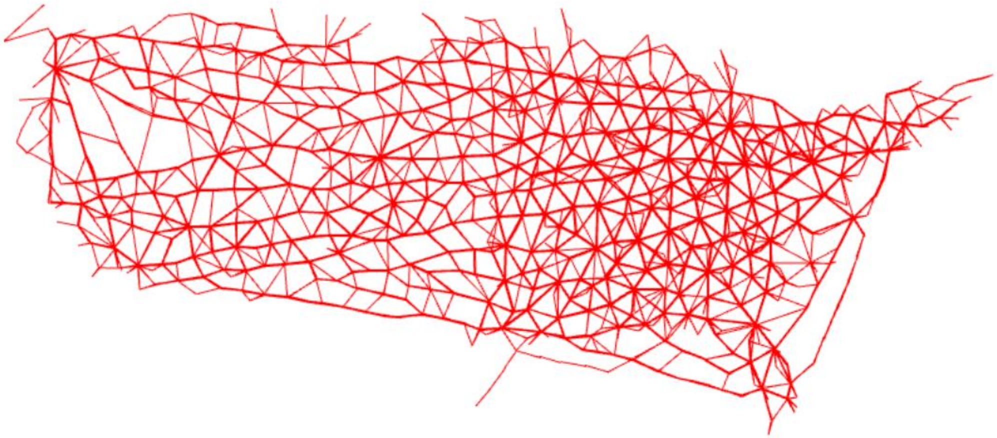


MLS

Approach

- MLS vs. KDE
  - KDE-based methods incur excessive convergence
    - Resampling is required to generate better bundling results

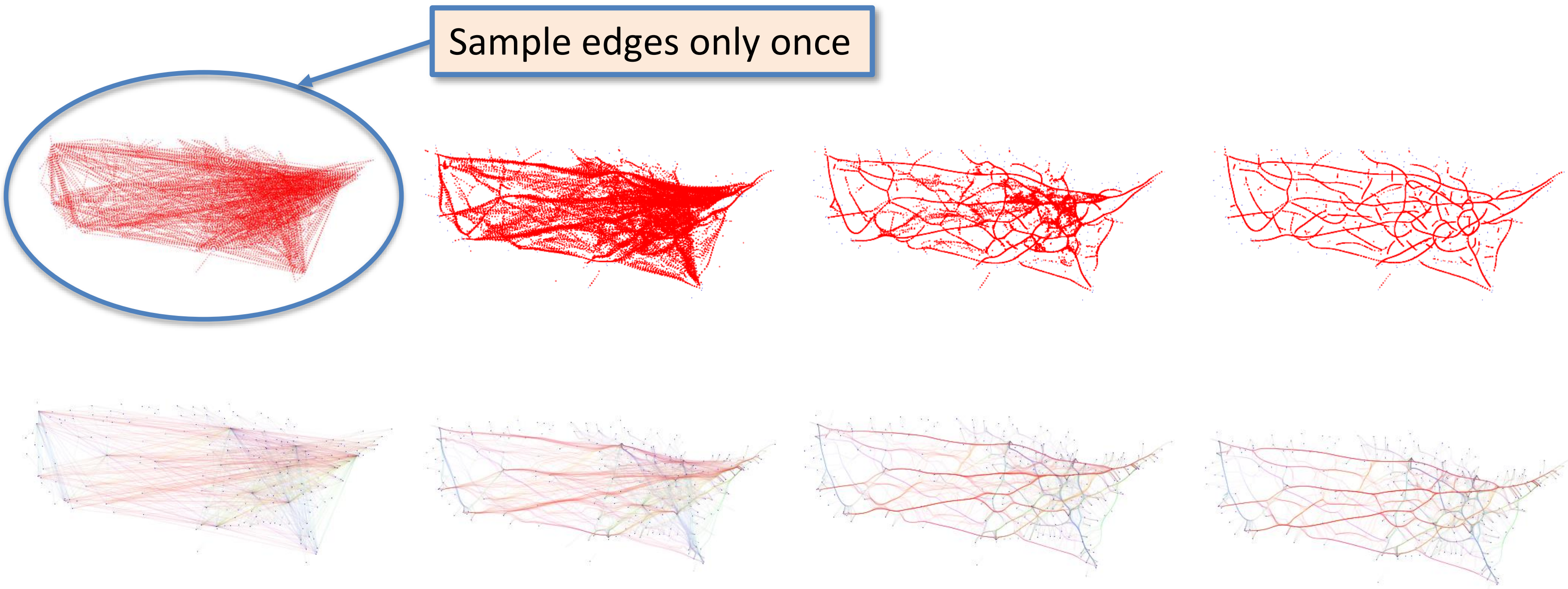




KDE

Approach

- MLS vs. KDE
  - MLS method only samples edges in the initial step, and it **doesn't** incur excessive convergence in the following iterations



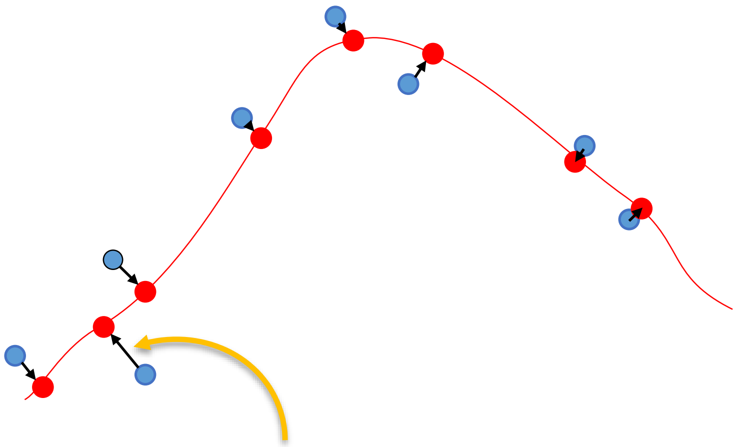
Iteration 0 Iteration 2 Iteration 5 Iteration 10

MLS

Approach

- Moving least squares edge bundling
  - Project a sample point x_i into its local regression curve f_i
    - f_i is locally approximated
      - Within a radial neighborhood of x_i
    - The distortion of x_i is locally minimized

$$\boldsymbol{\epsilon} = \sum_{j=1}^{h_i} (|x_j - f_i|)^2 \boldsymbol{\theta}(|x_j - x_i|)$$

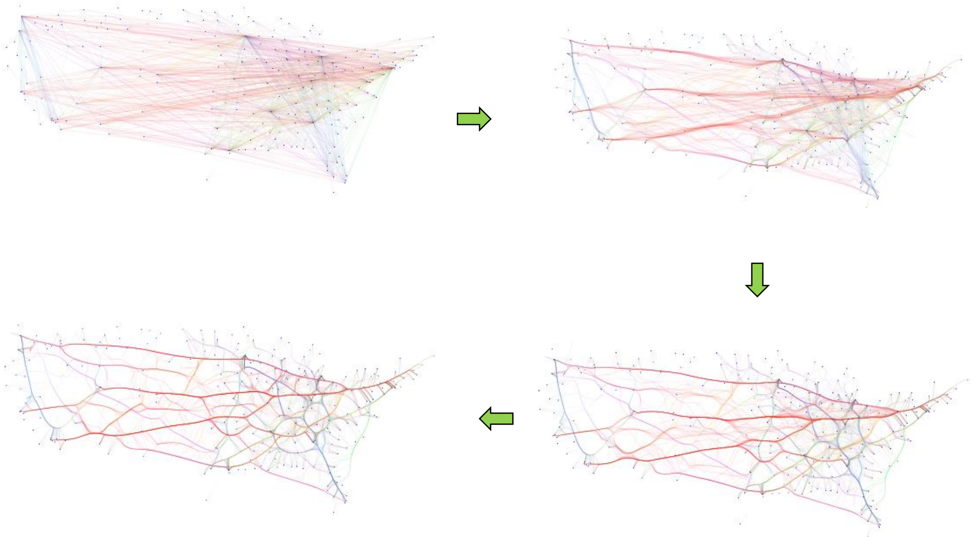


Approach

- Moving least squares edge bundling
  - Image-based sampling (sample edges in the initial step)
  - Moving least squares approximation and projection
  - Iterative refinement

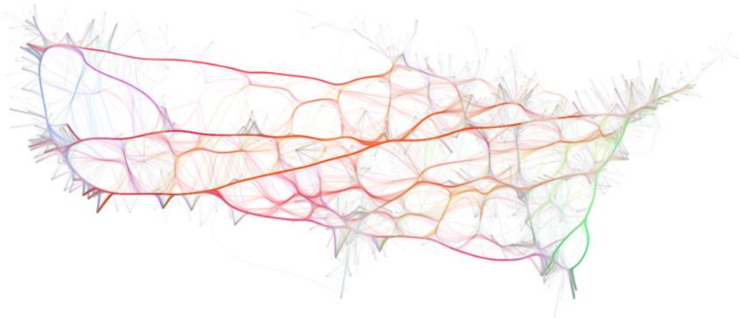
Complexity: O(SNI + E)

S: sample points
N: image pixel number
I: iteration number
E: edge number



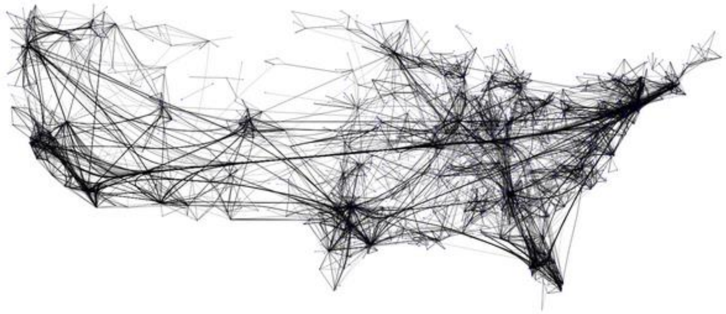
Results

- Dataset 1: a small US migrations graph (9780 edges)



MLSEB (our method)

FFTEB

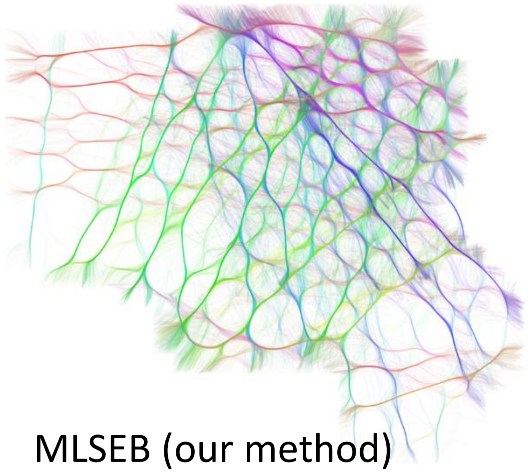


FDEB

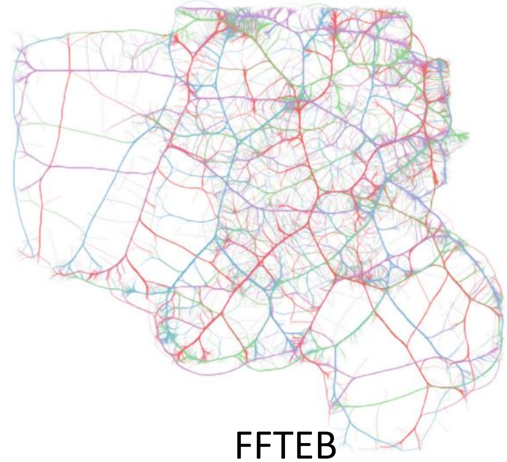
	Samples	Time (ms) / iteration	Iterations	Quality
FDEB	3785K	80	300	8.9
FFTEB	489K	48	262	7.60
MLSEB	207K	38	10	9.20

Results

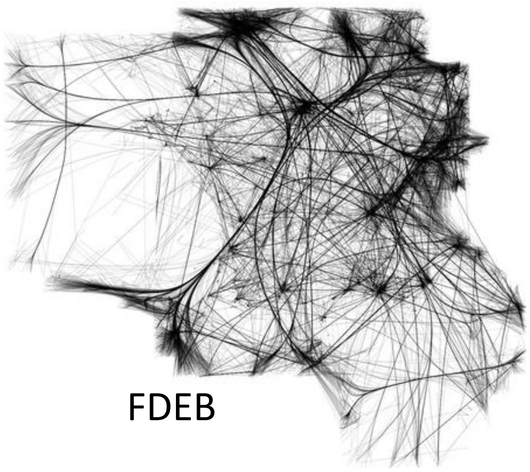
- Dataset 2: a France airlines graph (17274 edges)



MLSEB (our method)



FFTEB



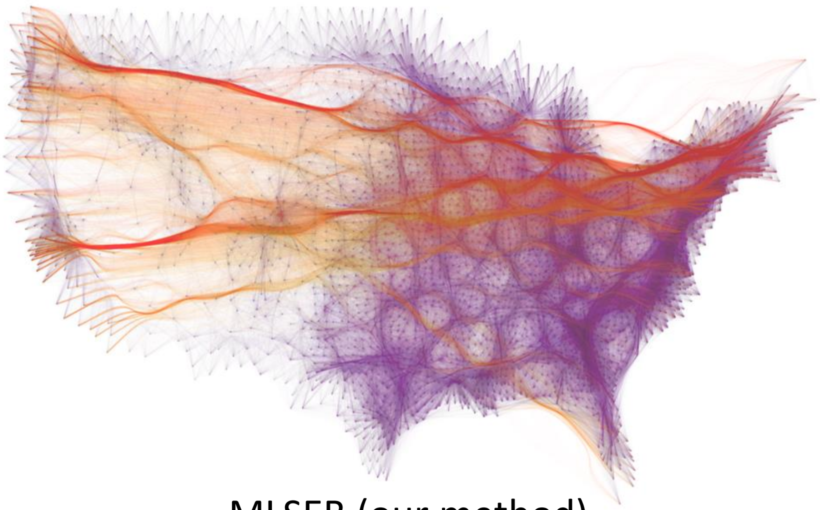
FDEB

	Samples	Time (ms) / iteration	Iterations	Quality
FDEB	6685K	110	300	3.7
FFTEB	864K	70	244	21.3
MLSEB	990K	94	10	26.0

Results

- Dataset 3: a large US migrations graph (545881 edges)

FFTEB



MLSEB (our method)

	Samples	Time (ms) / iteration	Iterations	Quality
FFTEB	6.4M	123	390	13.28
MLSEB	5.8M	554	20	13.30

Conclusion

- Moving Least Squares Edge Bundling (MLSEB)
  - A simple and efficient method for constructing edge bundles of large graphs using MLS projection
    - Only sample edges once, and avoid resampling in the following iterations
    - Achieve better visualization results based on a quality metric
    - Ensure scalability and efficiency

Acknowledgement

- This research has been sponsored by the National Science Foundation through grants IIS-1652846, IIS-1423487, and ICER-1541043.

Thank You!


Q&A