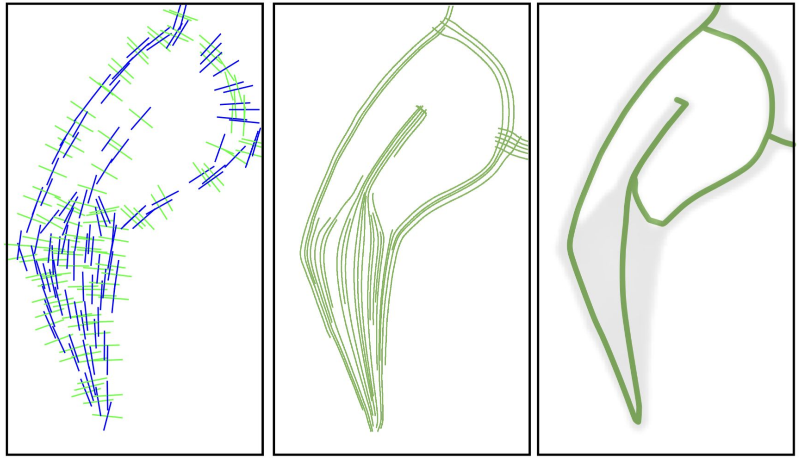
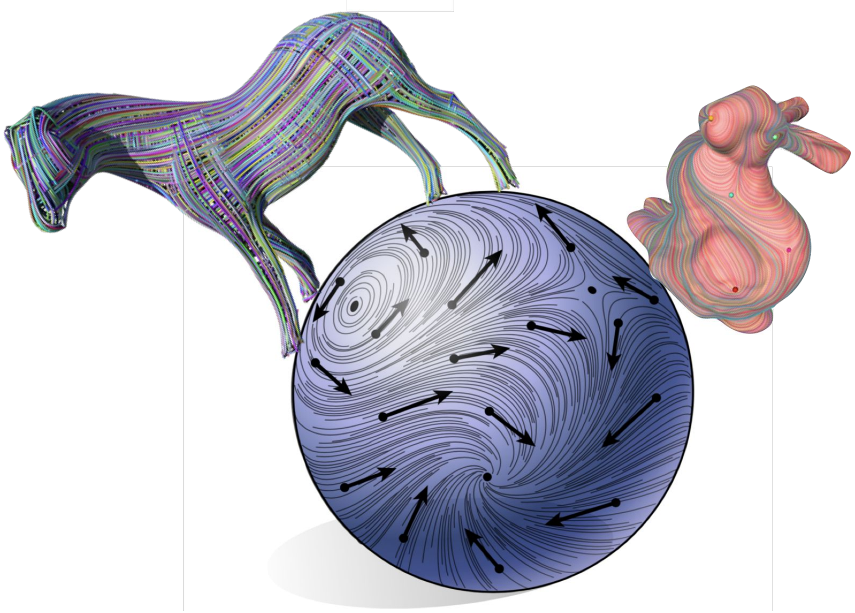
Singularity-Free Frame Fields for Line Drawing Vectorization

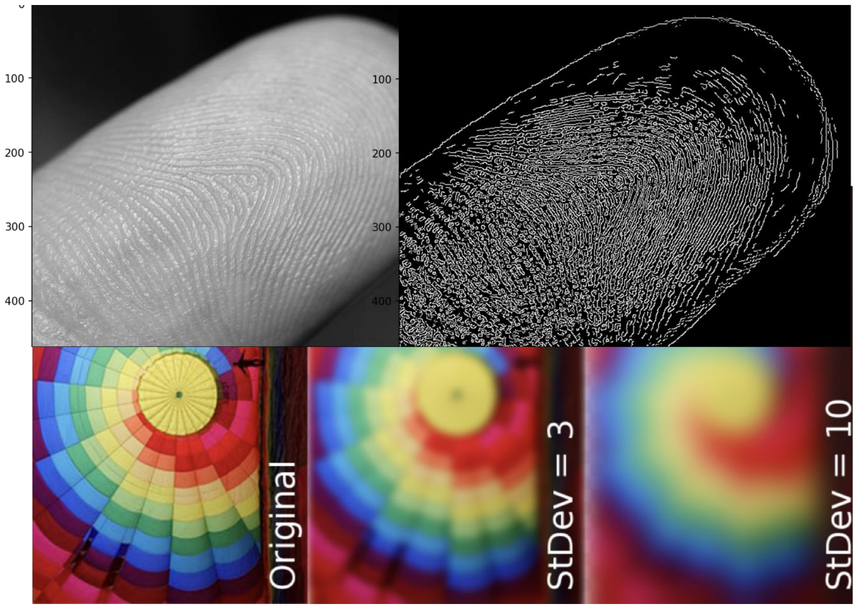


Olga Guțan*, Carnegie Mellon University
Shreya Hegde*, Amherst College
Erick Jimenez Berumen, Caltech
Mikhail Bessmeltsev, Université de Montréal
Edward Chien, Boston University

Symposium on Geometry Processing 2023 | July 3, 2023



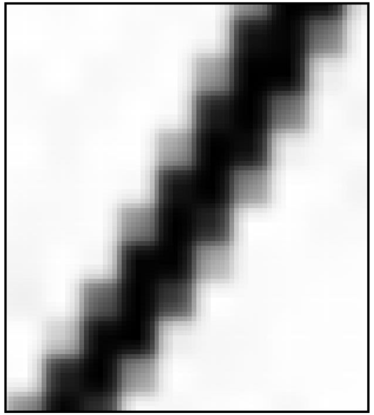
Methods from
Discrete Differential Geometry



2D Image Processing

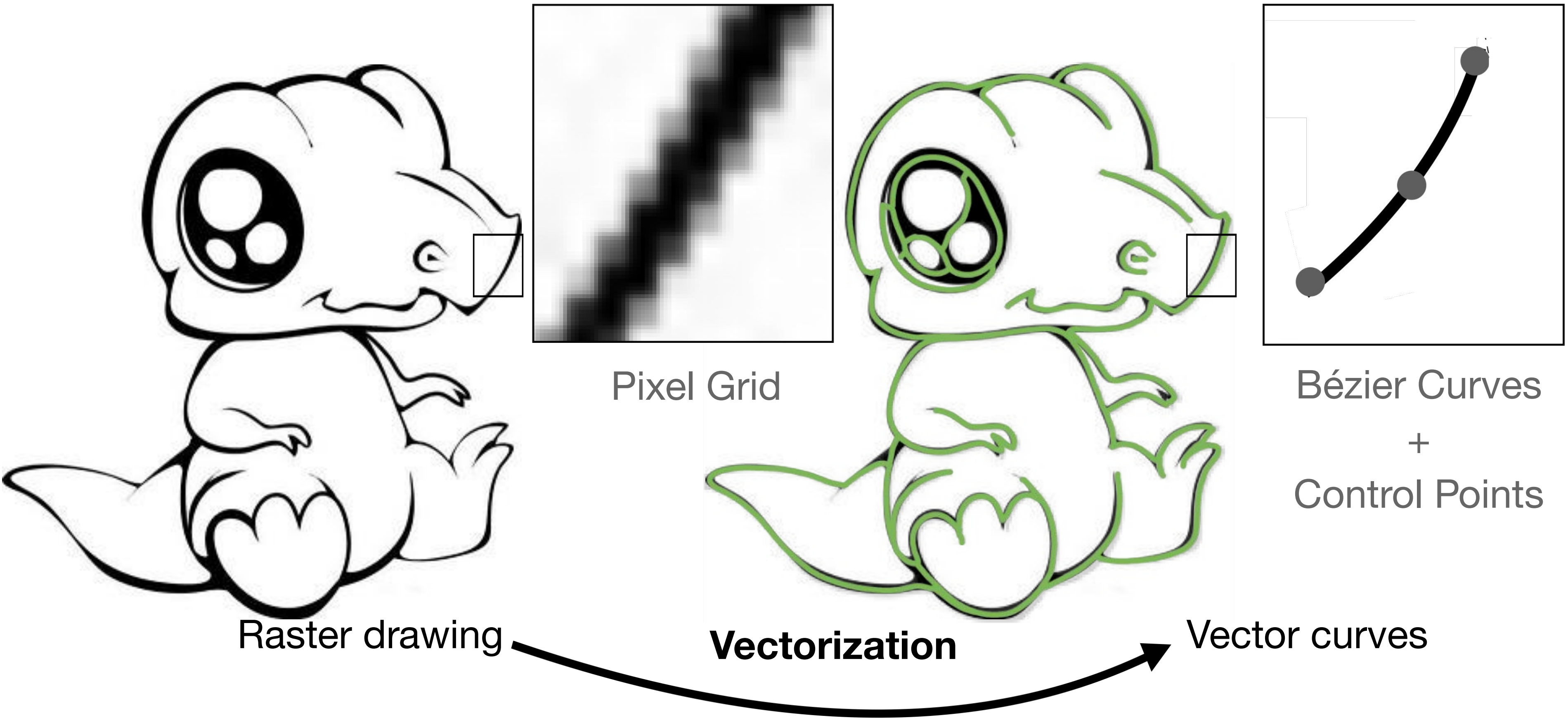
Vectorization of Line Drawings

Raster drawing



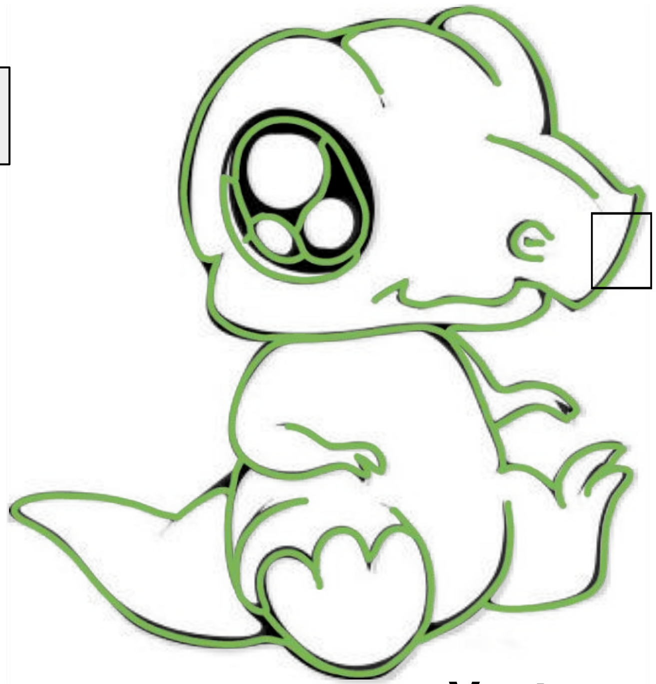
Pixel Grid

Vectorization of Line Drawings

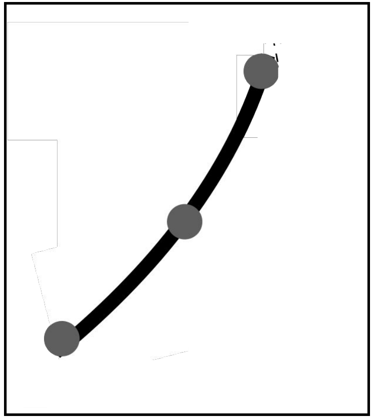


Why Vectorize?

Resolution Independent



Vector curves

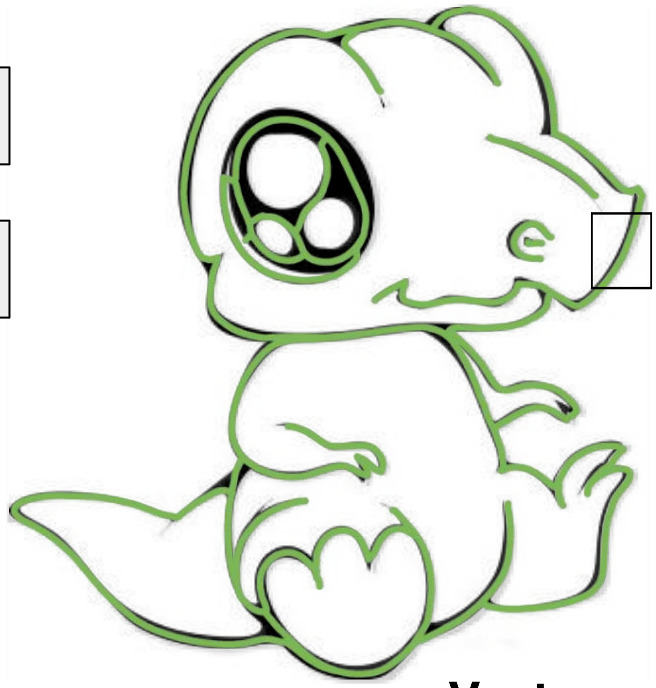


Bézier Curves
+
Control Points

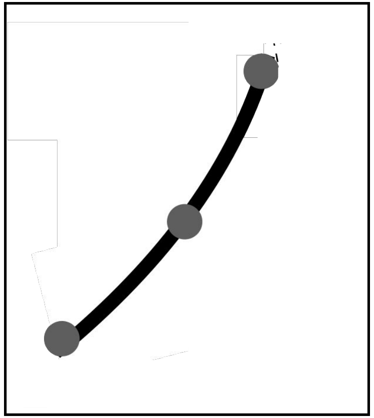
Why Vectorize?

Resolution Independent

Compact Size



Vector curves



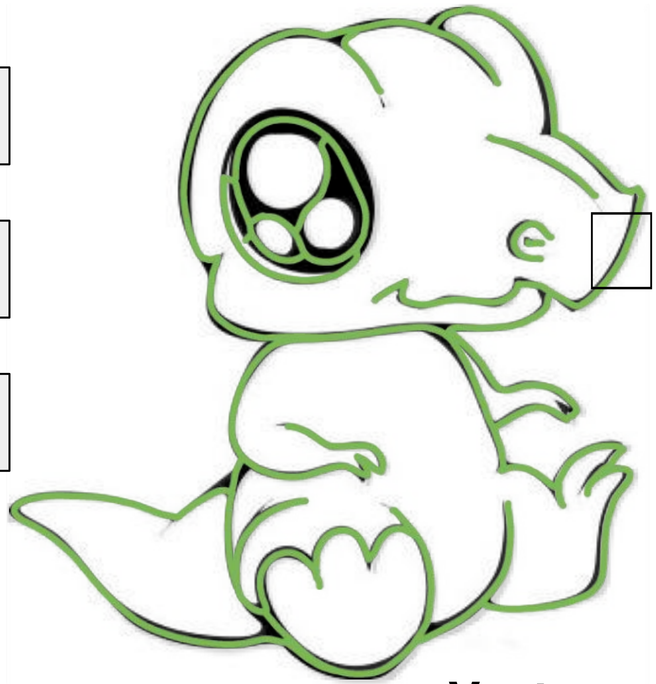
Bézier Curves
+
Control Points

Why Vectorize?

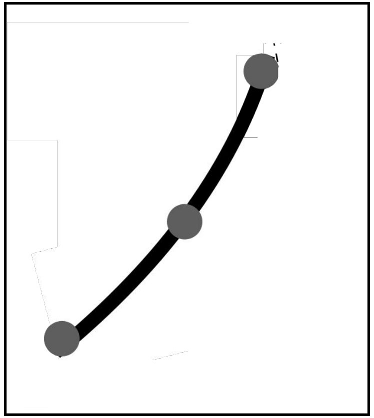
Resolution Independent

Compact Size

Easy to Edit



Vector curves



Bézier Curves

+

Control Points

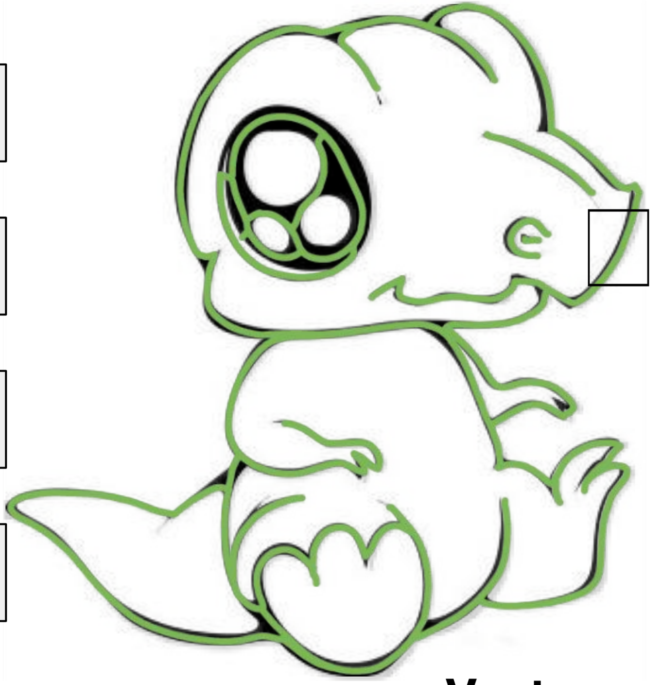
Why Vectorize?

Resolution Independent

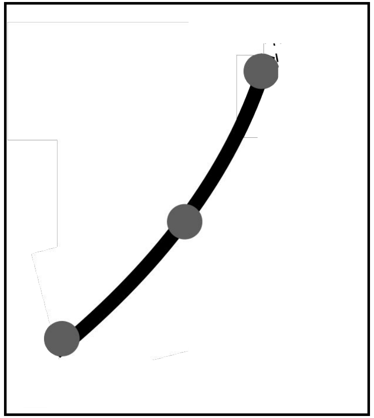
Compact Size

Easy to Edit

Input to Algorithms



Vector curves



Bézier Curves

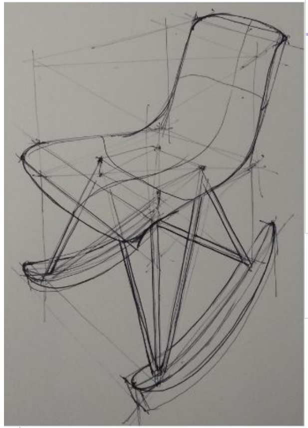
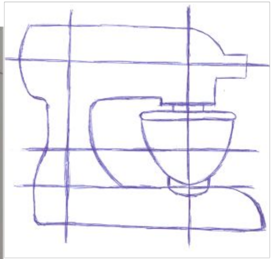
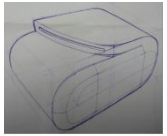
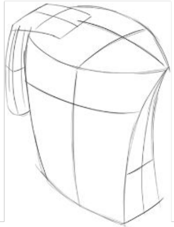
+

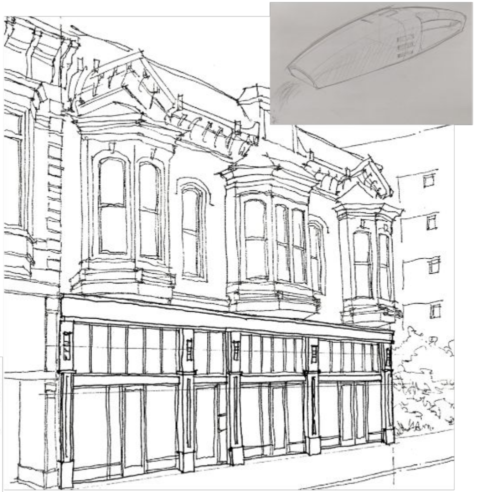
Control Points

Vectorization is Very Useful!

Architecture

Vectorization is Very Useful!

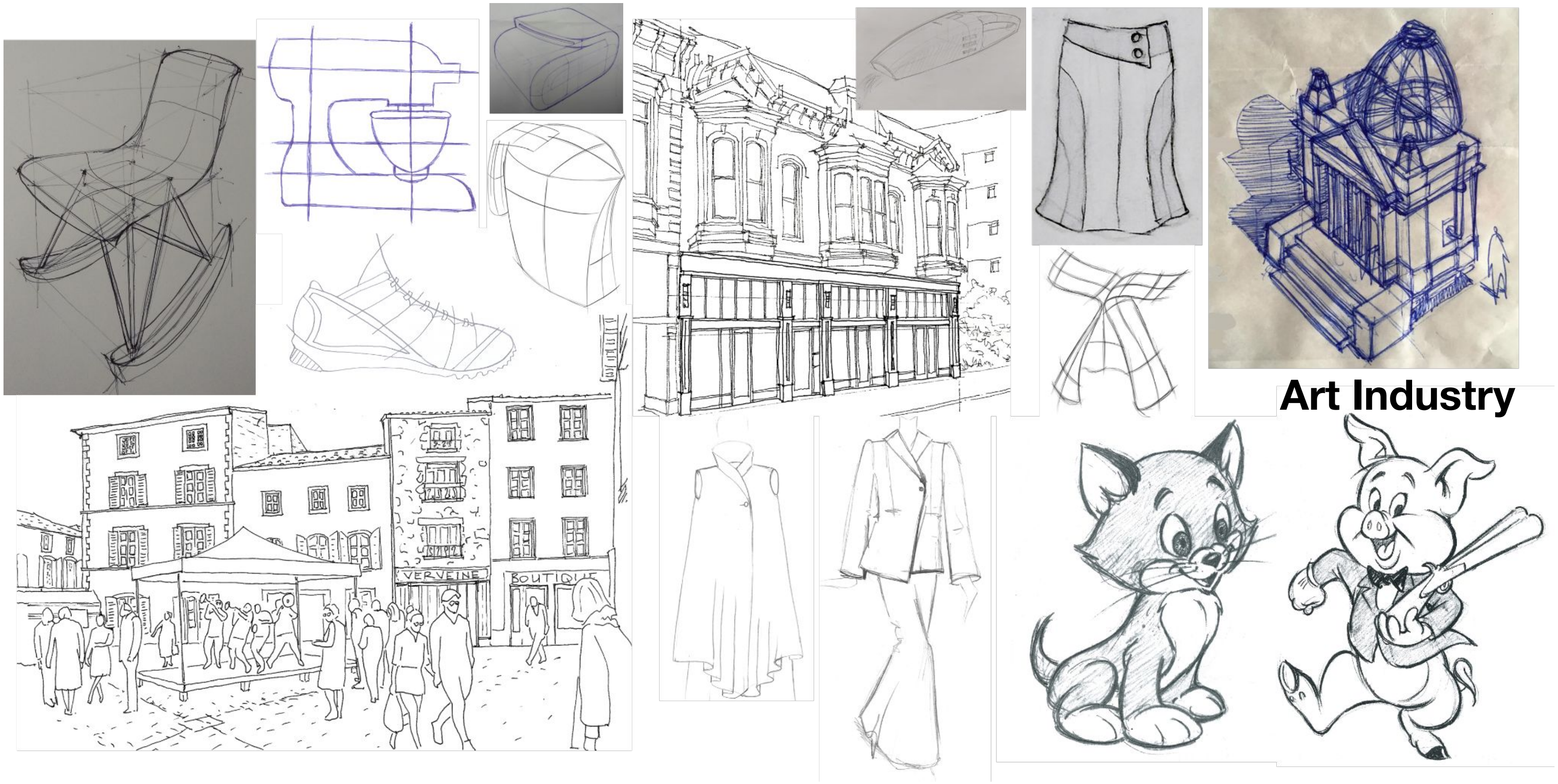



Industrial Product Design

Vectorization is Very Useful!

Fashion Industry

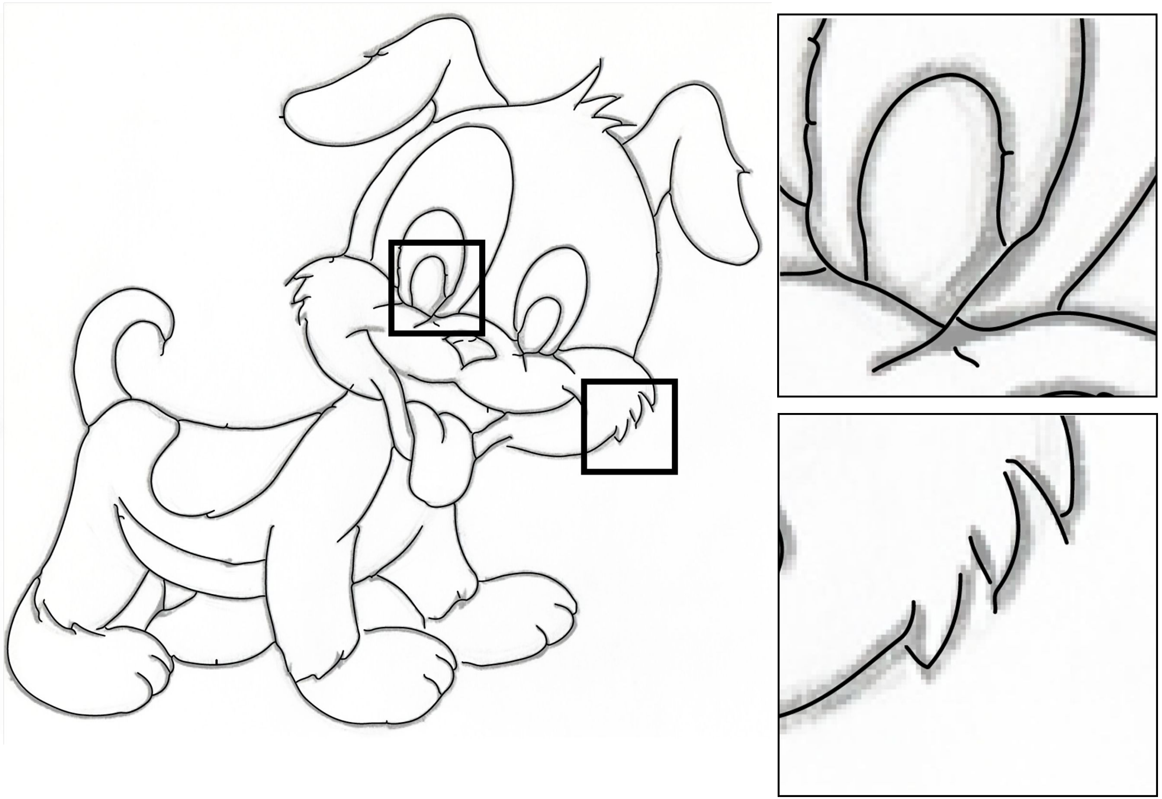
Vectorization is Very Useful!



Art Industry

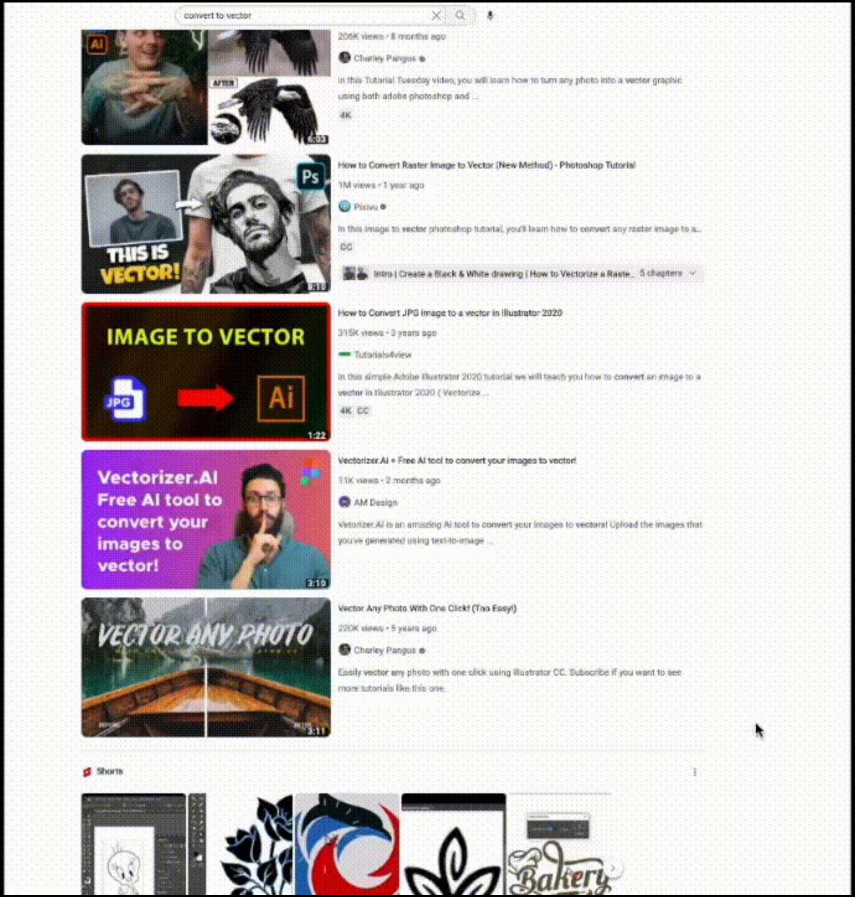
Commercial Methods are not Accurate Enough

Input



[Adobe Illustrator 2021]

Vectorization is Difficult (and Popular!)



Vectorization is Difficult (and Popular!)

YouTube
https://www.youtube.com › watch

Convert Any Photo Into Vector Graphics! (Photoshop ...

in this Tutorial Tuesday video, you will learn how to turn any **photo** into a **vector graphic** using both adobe photoshop and illustrator ...

YouTube · Charley Pangus · Sep 20, 2022

10 key moments in this video

Sticker Mule
https://www.stickermule.com › blog › how-to-convert...

6 ways to convert any image to vector (free & paid) | Blog

Apr 3, 2023 — 4) **Convert images to vector** in Illustrator · Open a new document in Adobe Illustrator. · On the menu bar, click on File, and then Place. Find the ...

Sponsored

Vector Design US
https://www.vectordesign.us › vectorize-image

$5 Vectorizing an Image - Same Day Pricing Starts at $5

Pro **Vector** Redraw Service Starts $5. We Vectorize by Hand - No Auto Tracing - Let's Starts

Get Free Quote · Works Sample · Best Vectorize Services · Contact US

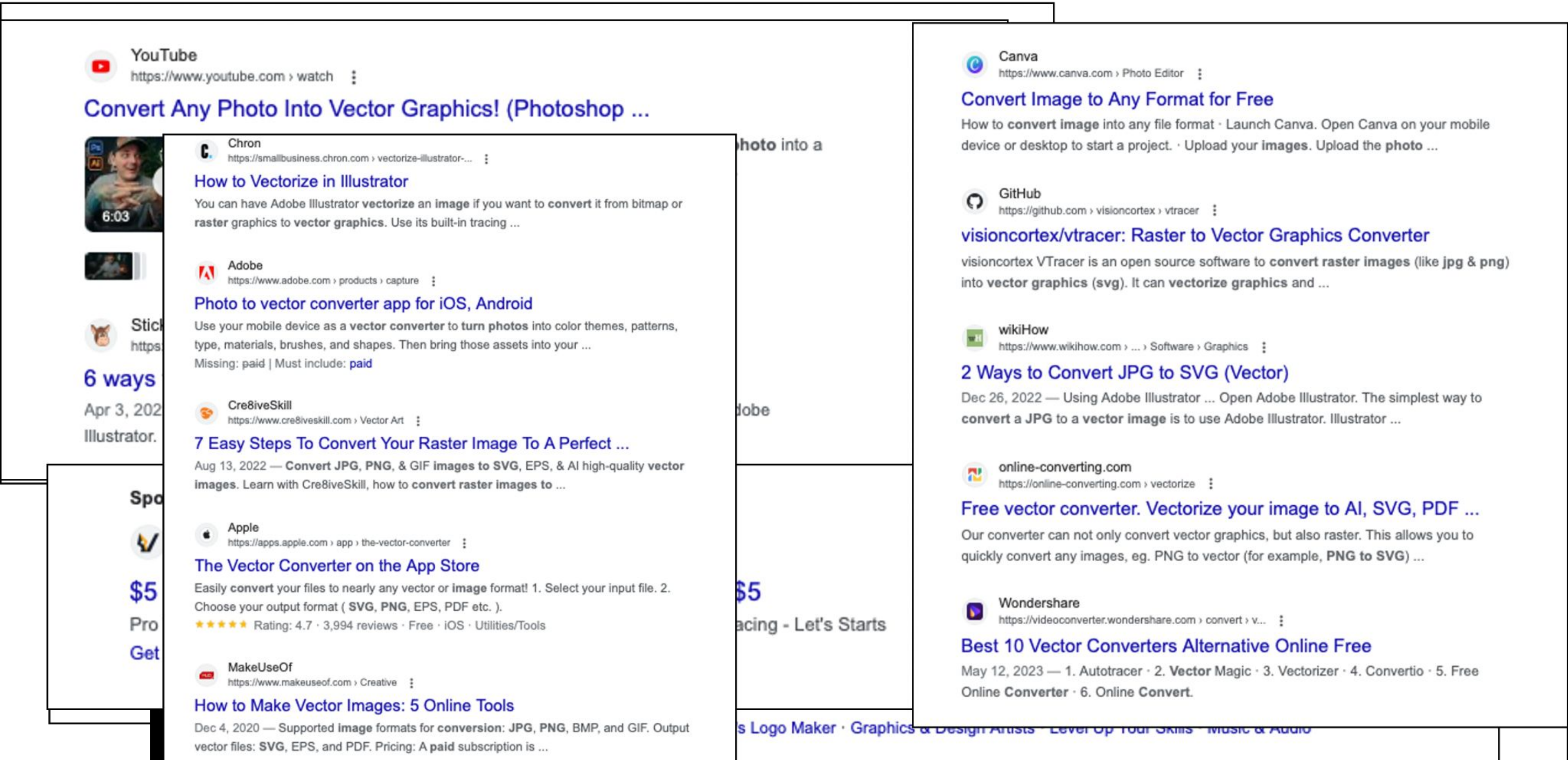
ine | Fiverr

Get the Job Done,

Task Possible....

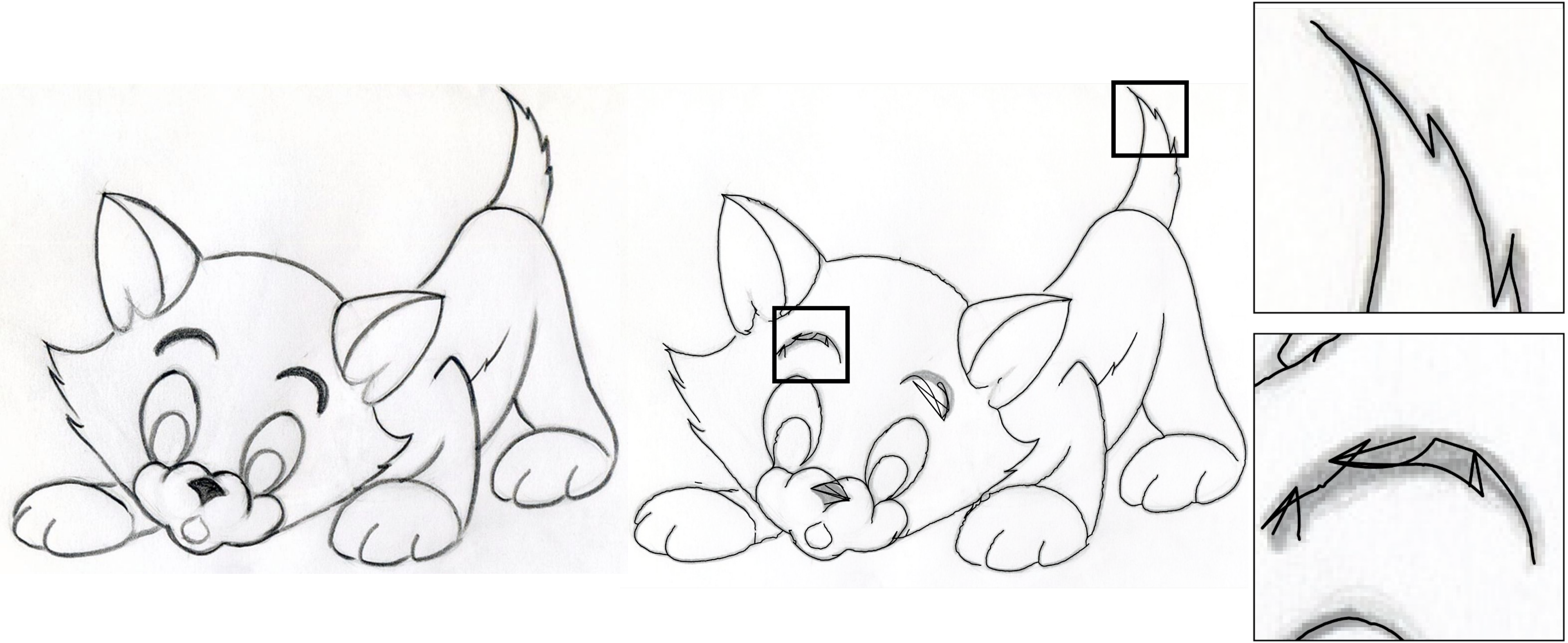
Try Fiverr's Logo Maker · Graphics & Design Artists · Level Up Your Skills · Music & Audio

Vectorization is Difficult (and Popular!)



(Recent!) Past Work

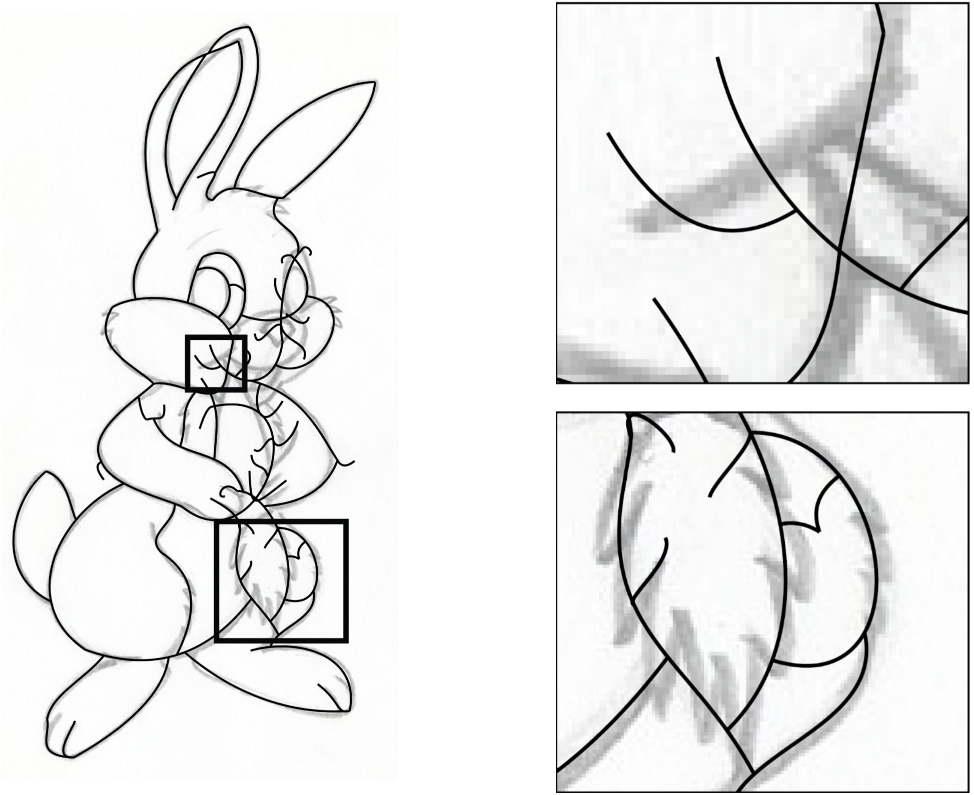
1-Skeleton Methods: Suitable for Clean Digital Input Only



Input [Noris et al. 2013]

Region-Based Skeleton: Oversimplifies Vectorization

Input

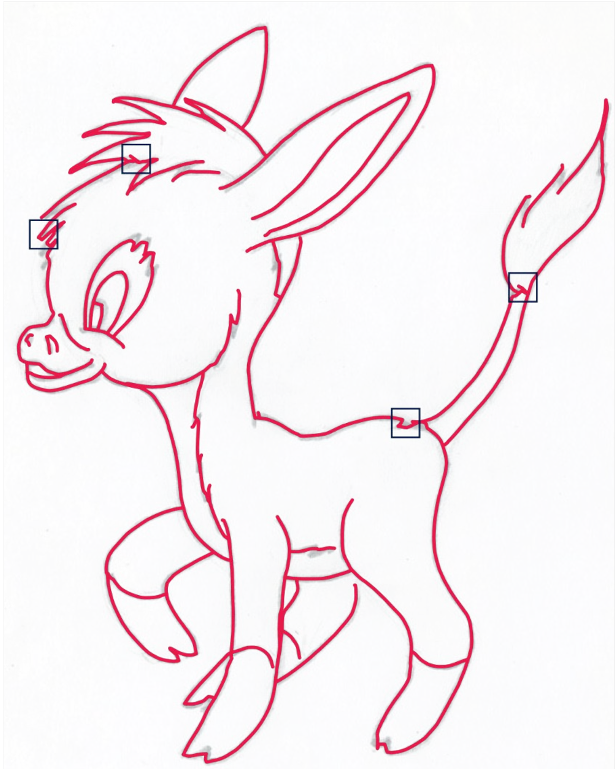


[Favreau et al. 2016]

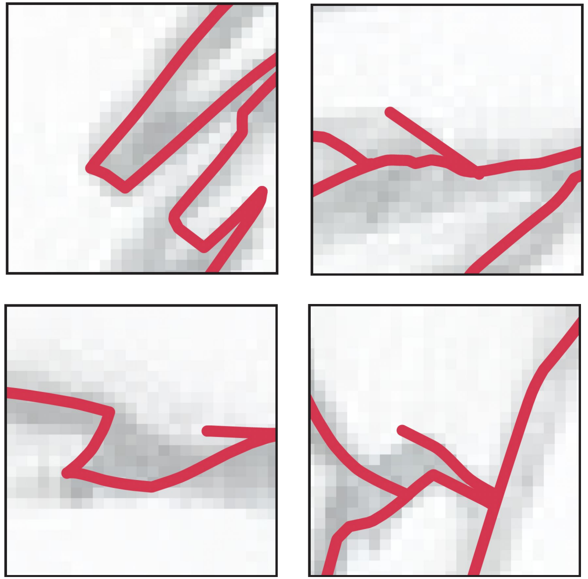
Frame-Fields–Based Methods

Polyvector Fields Vectorization: Does Not Capture Details

Input

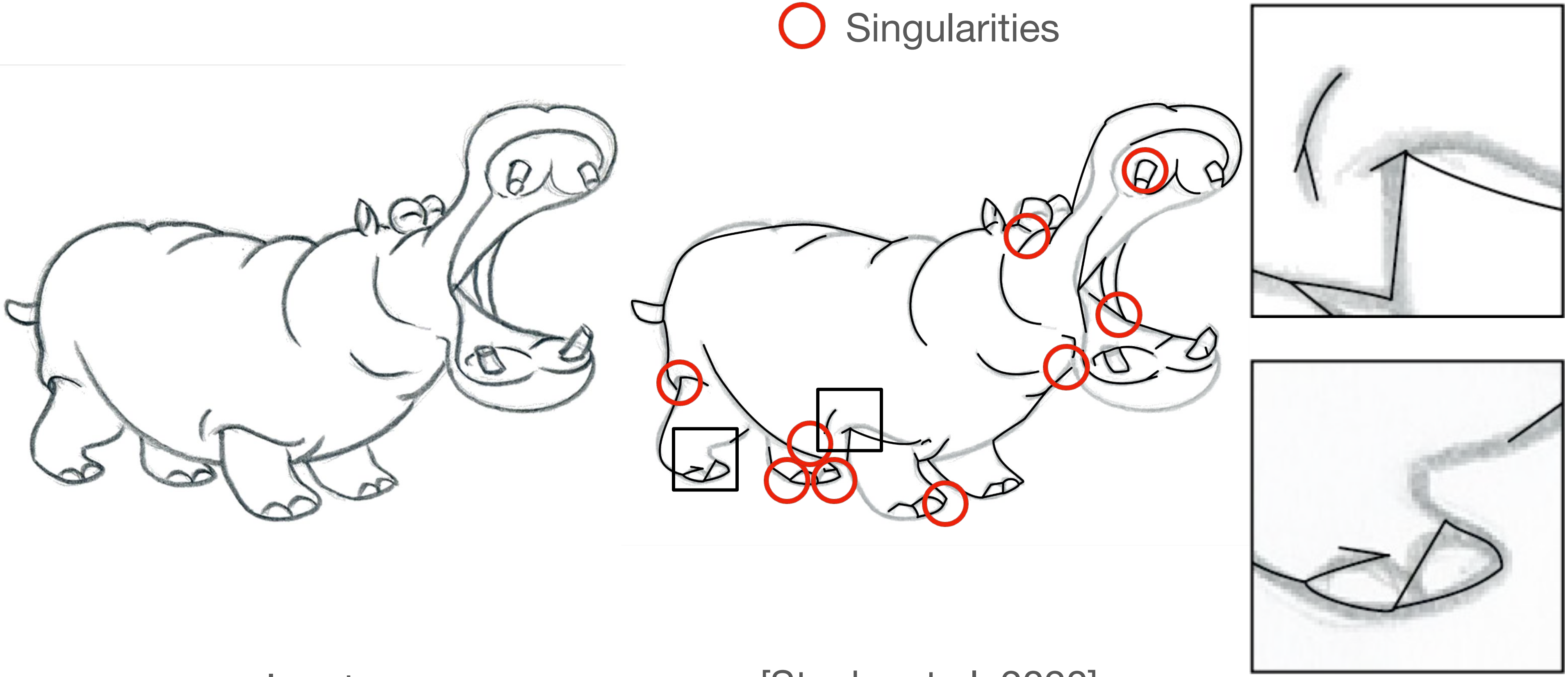


[Bessmeltsev and Solomon 2019]



- **Dull corners**
- **Spurious lines**
- **Interrupted tracing**

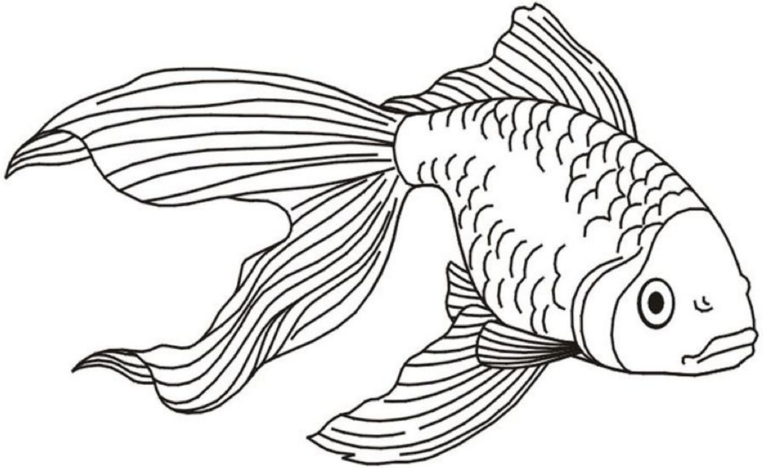
Parameterization: Relies on the Absence of Singularities



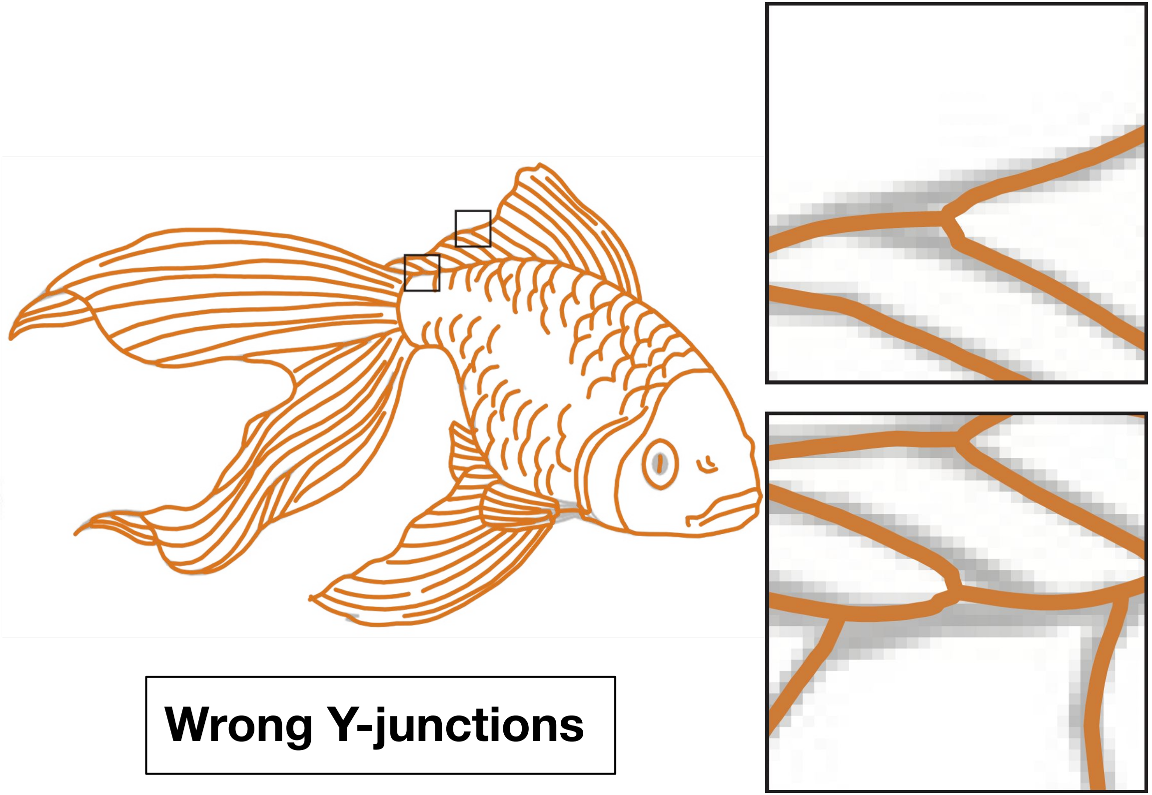
Input

[Stanko et al. 2020]

Polyvector Flow: Incorrectly Resolves Junctions



Input



[Puhachov et al. 2021]

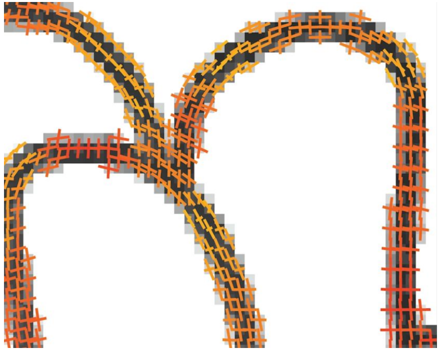
Frame-Fields–Based Vectorization Pipeline

1 – Isolate Algorithm Domain

1. Preprocessing

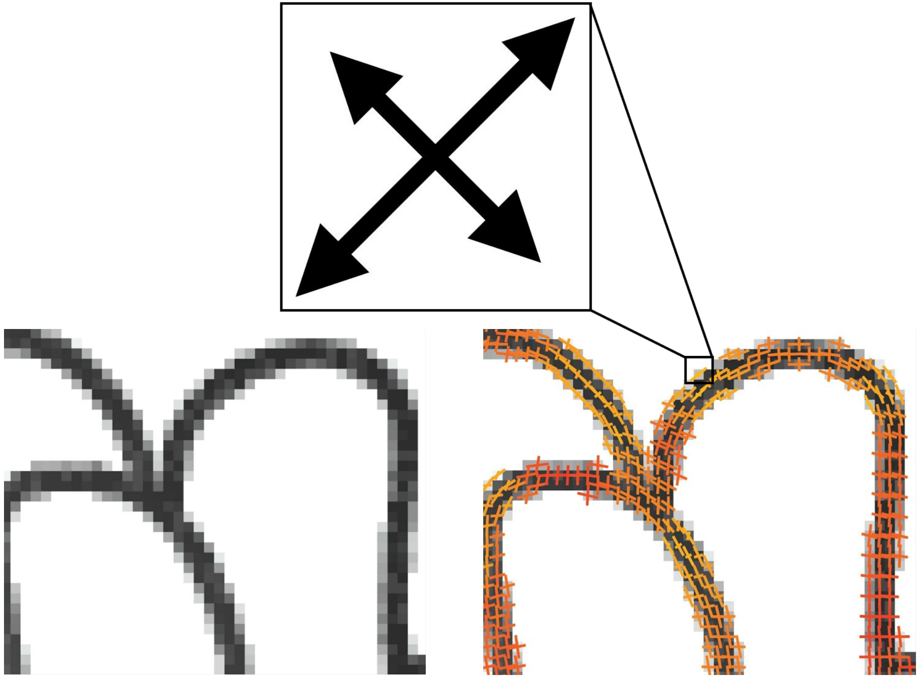
2 – Compute Frame Fields for Each Domain Pixel

1. Preprocessing



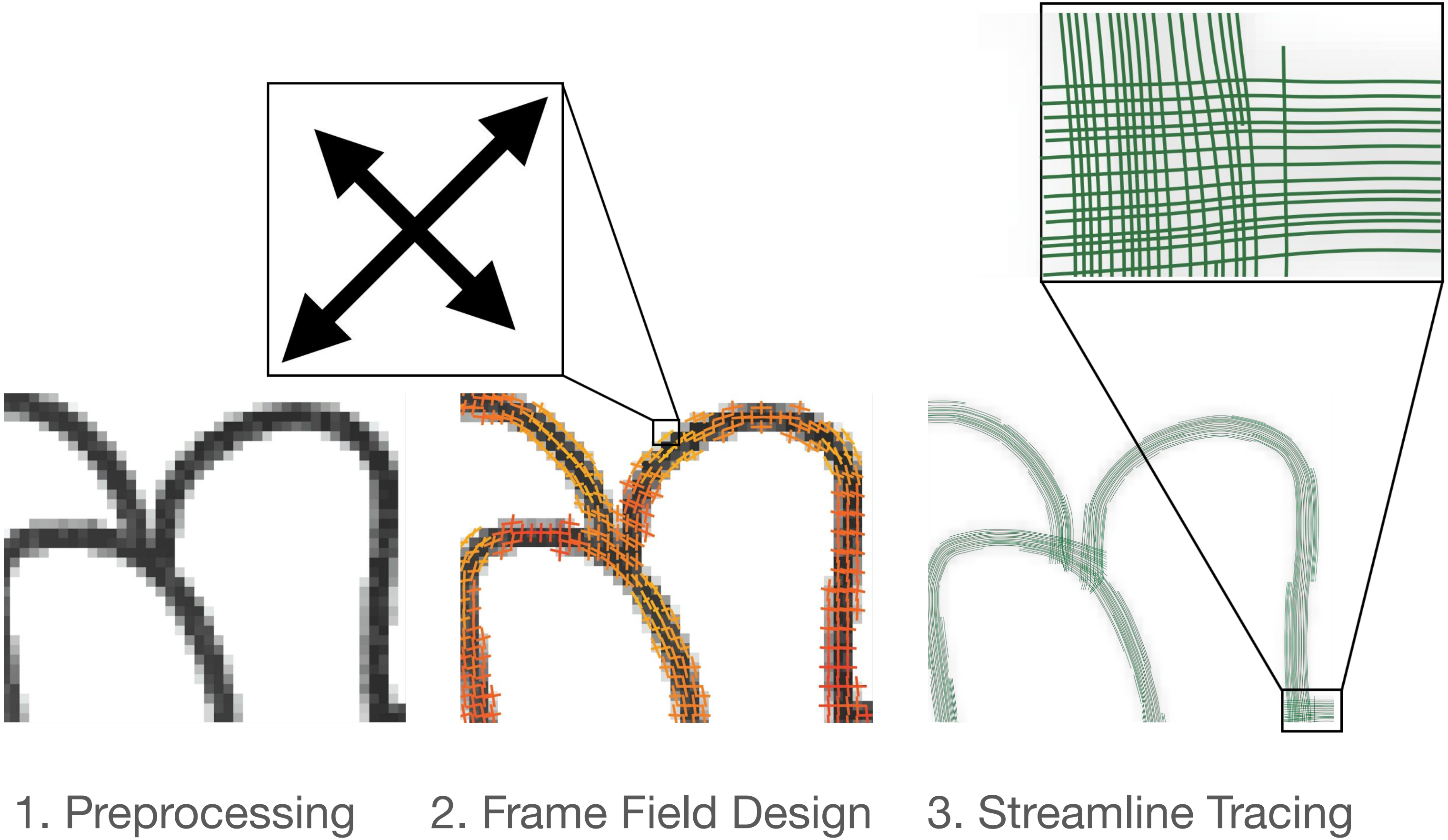
2. Frame Field Design

2 – Frame Fields Indicate Directions

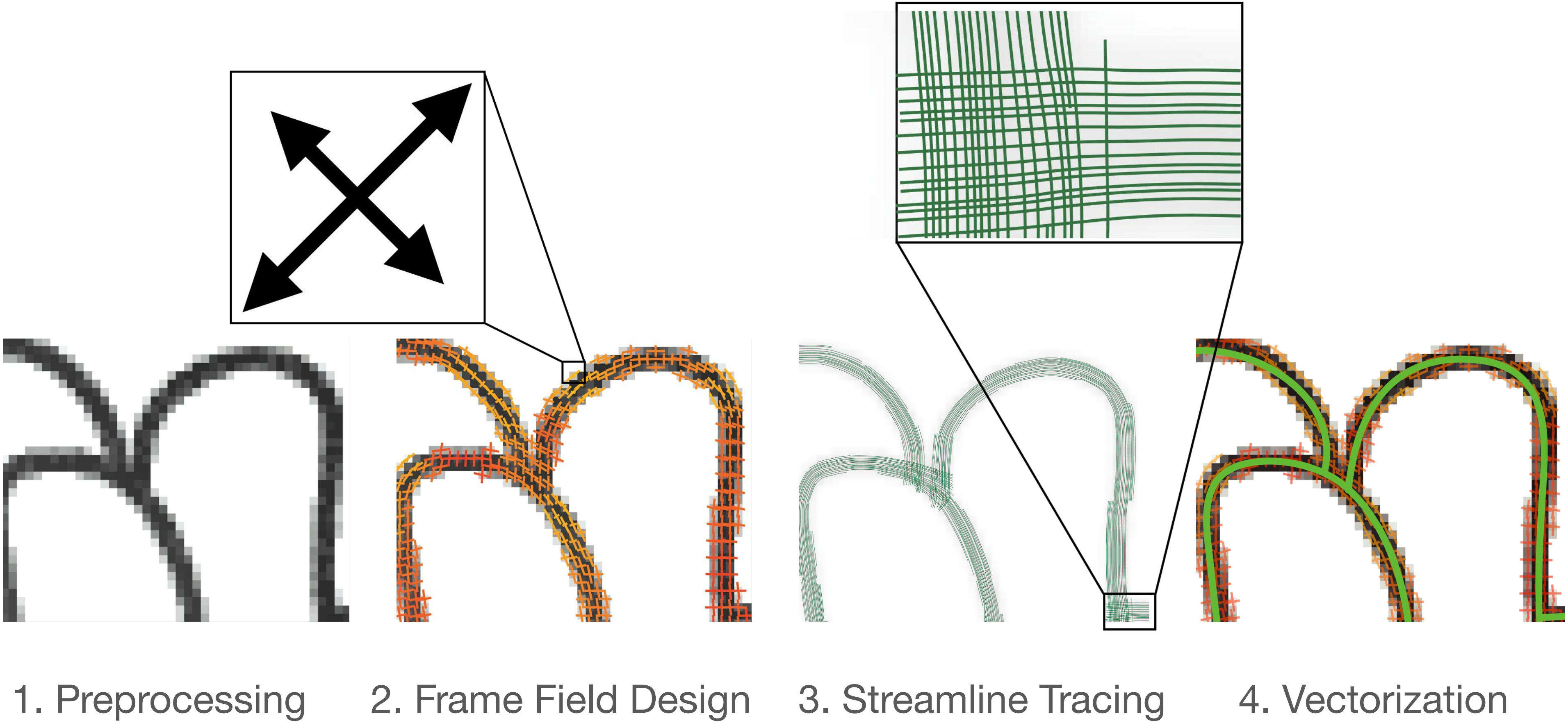


1. Preprocessing 2. Frame Field Design

3 – Compute Streamlines from Directions



4 – Compute Vector Curves from Streamlines

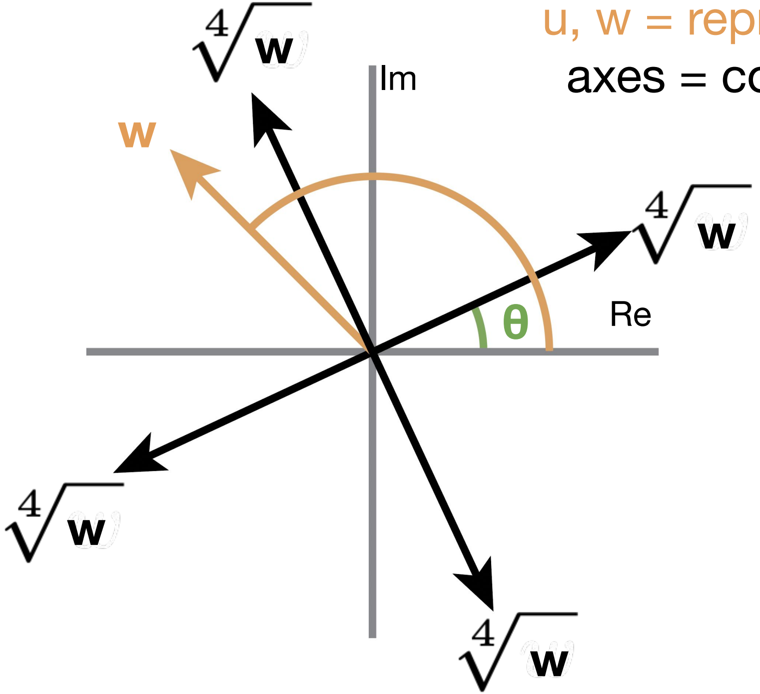


How to Construct Frame Fields?

Orthogonal Cross Fields

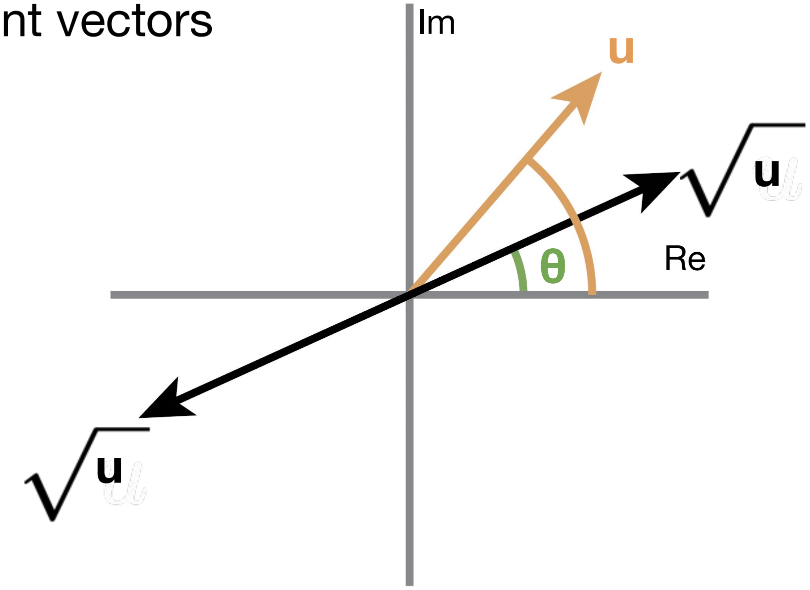
u, w = representative vectors

axes = component vectors



Cross Field

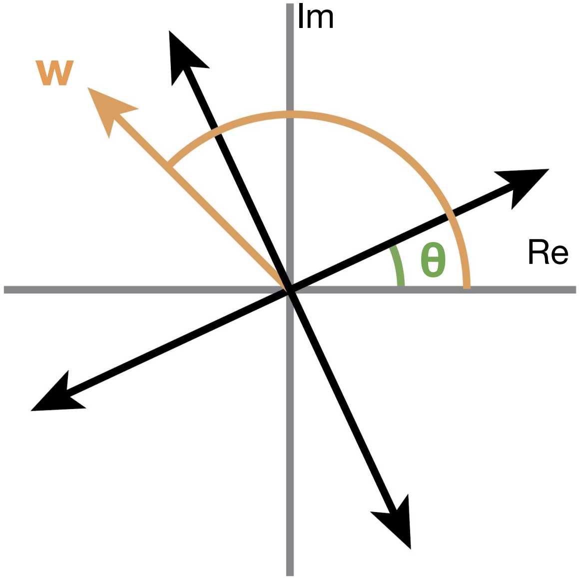
arg (w) = 4θ



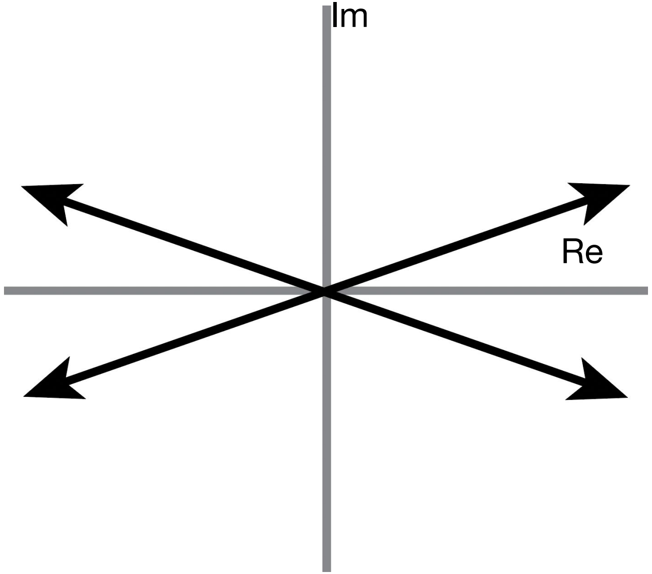
Line Field

arg (u) = 2θ

Nonorthogonal Frame Fields



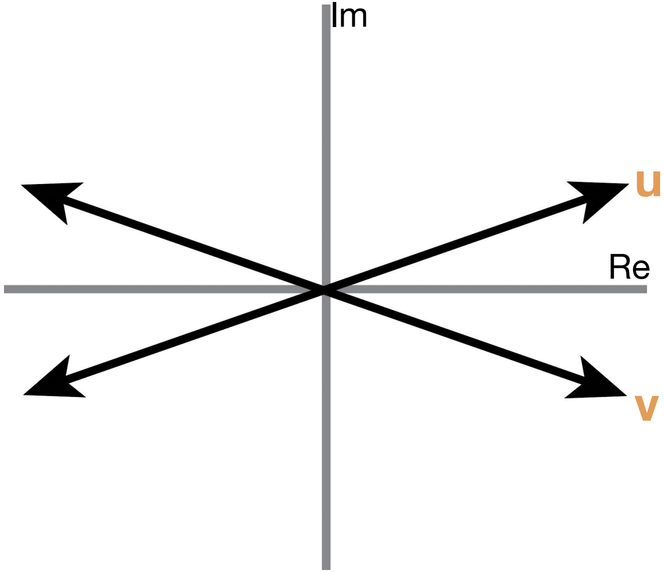
Orthogonal



Nonorthogonal

Nonorthogonal Frame Fields

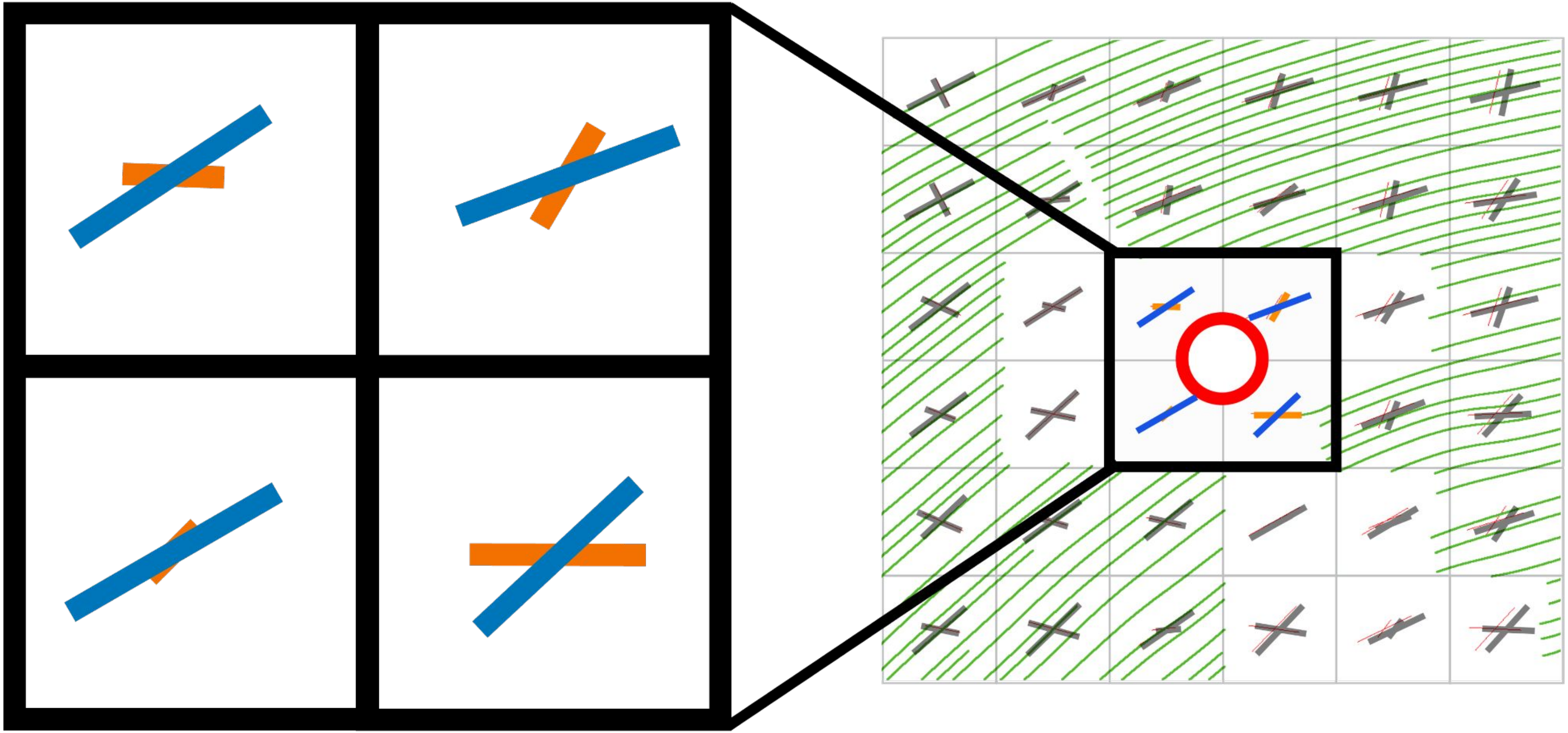
[Bessmeltsev and Solomon 2019]



[Puhachov et al. 2021]

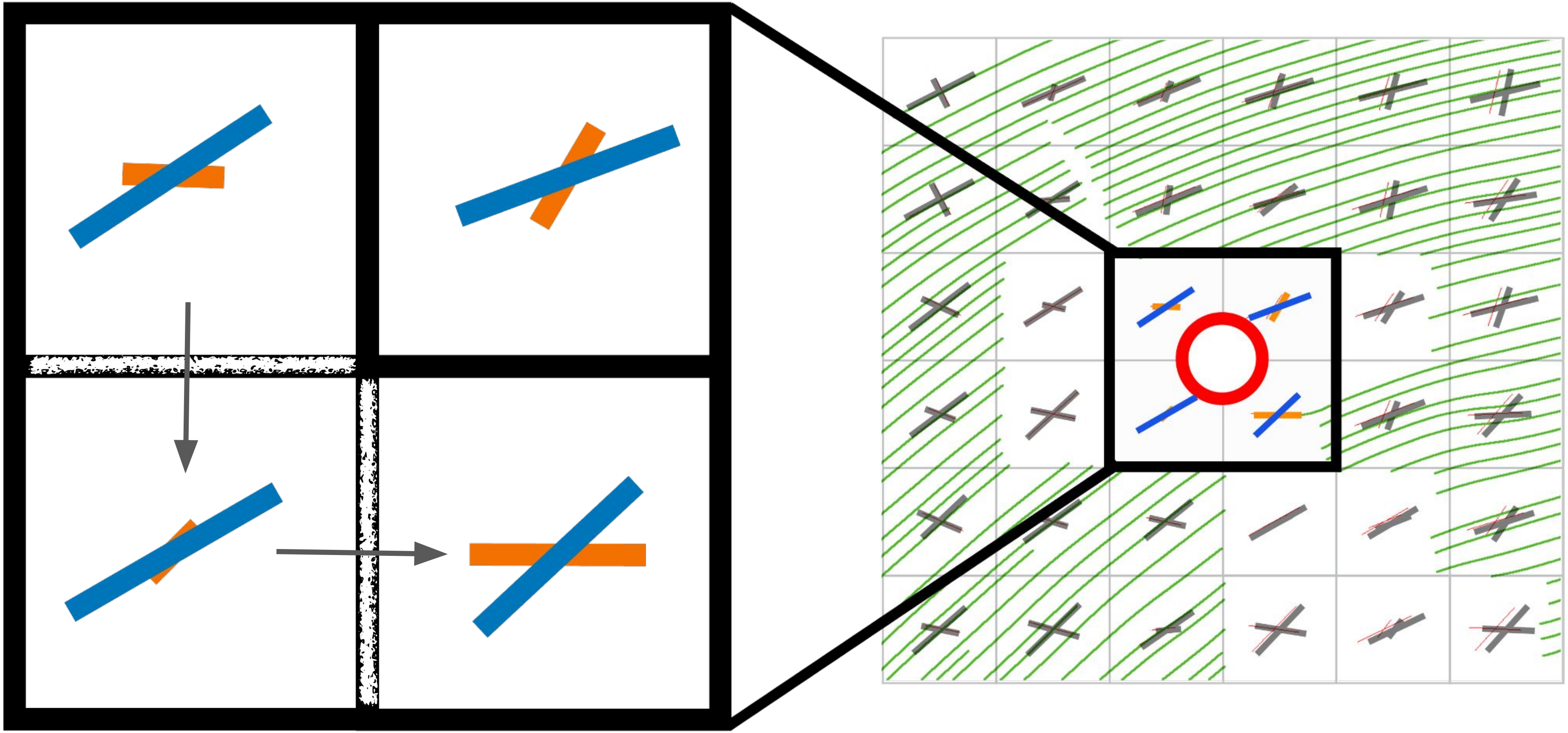
$$f(z) = (z^2 - u^2)(z^2 - v^2) = z^4 + c_2 z^2 + c_0$$

Polyvector Singularities: Axis Switch

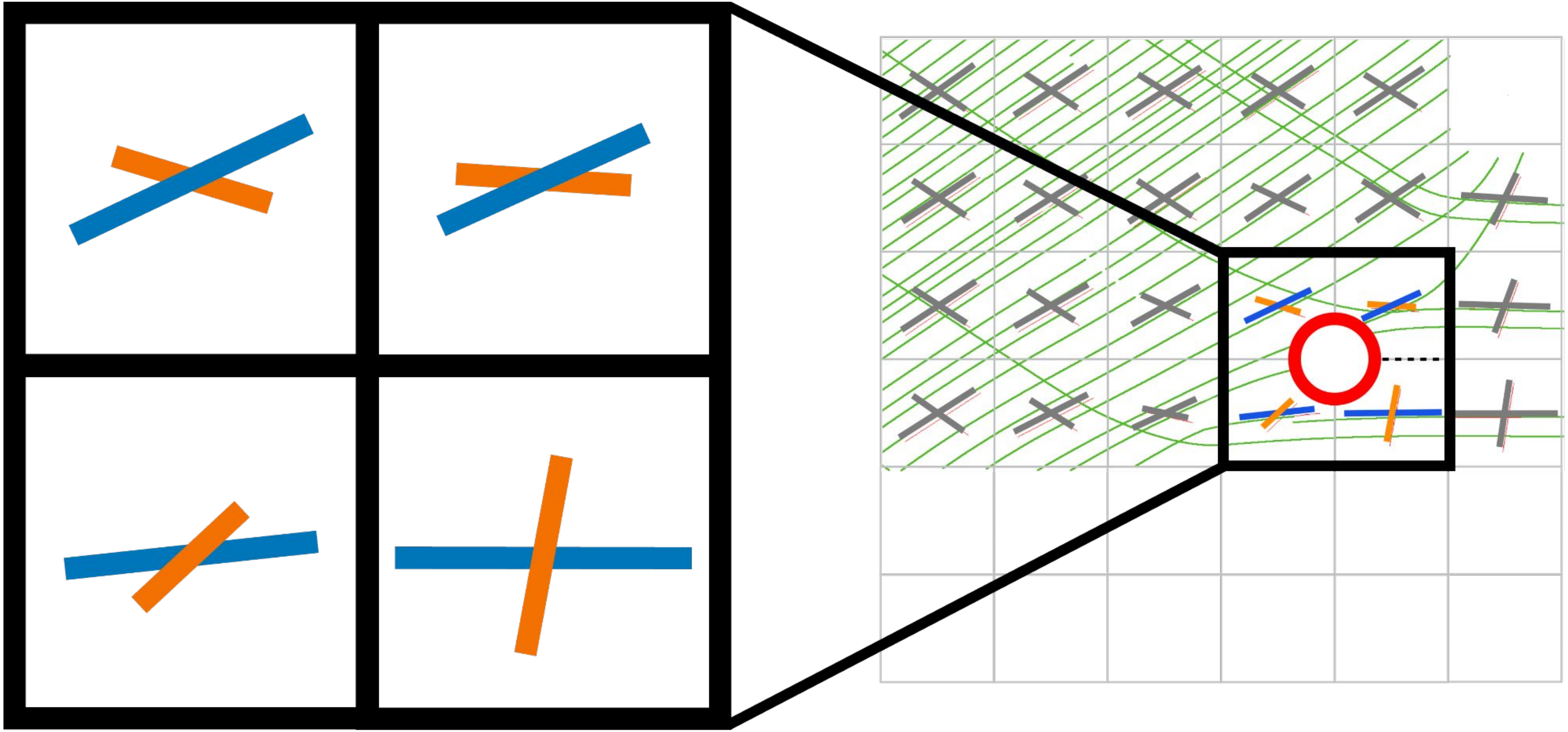


Polyvector Singularities: Axis Switch Details

Smaller orange axis vanishes + switches with the blue one

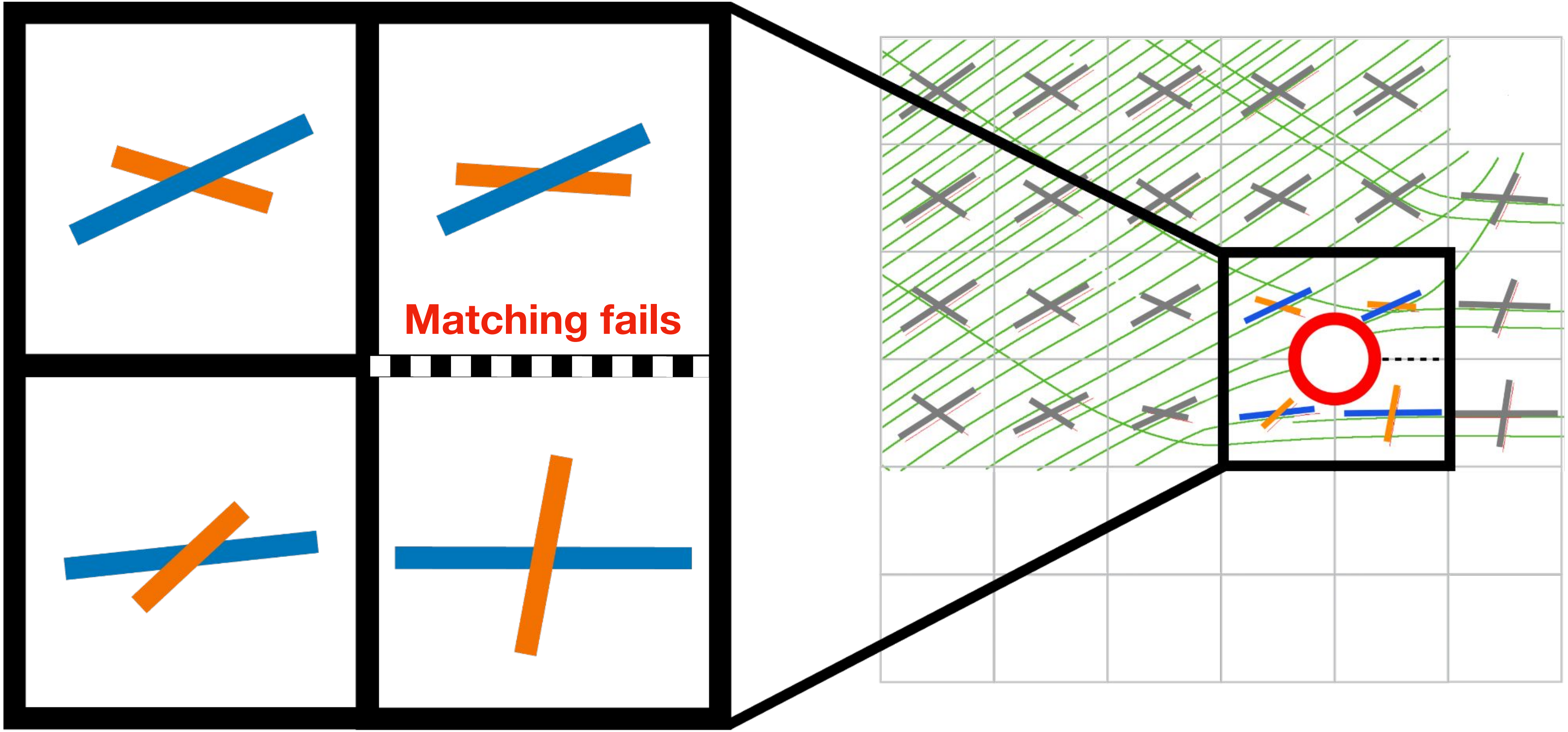


Polyvector Singularities: Zero Singularity

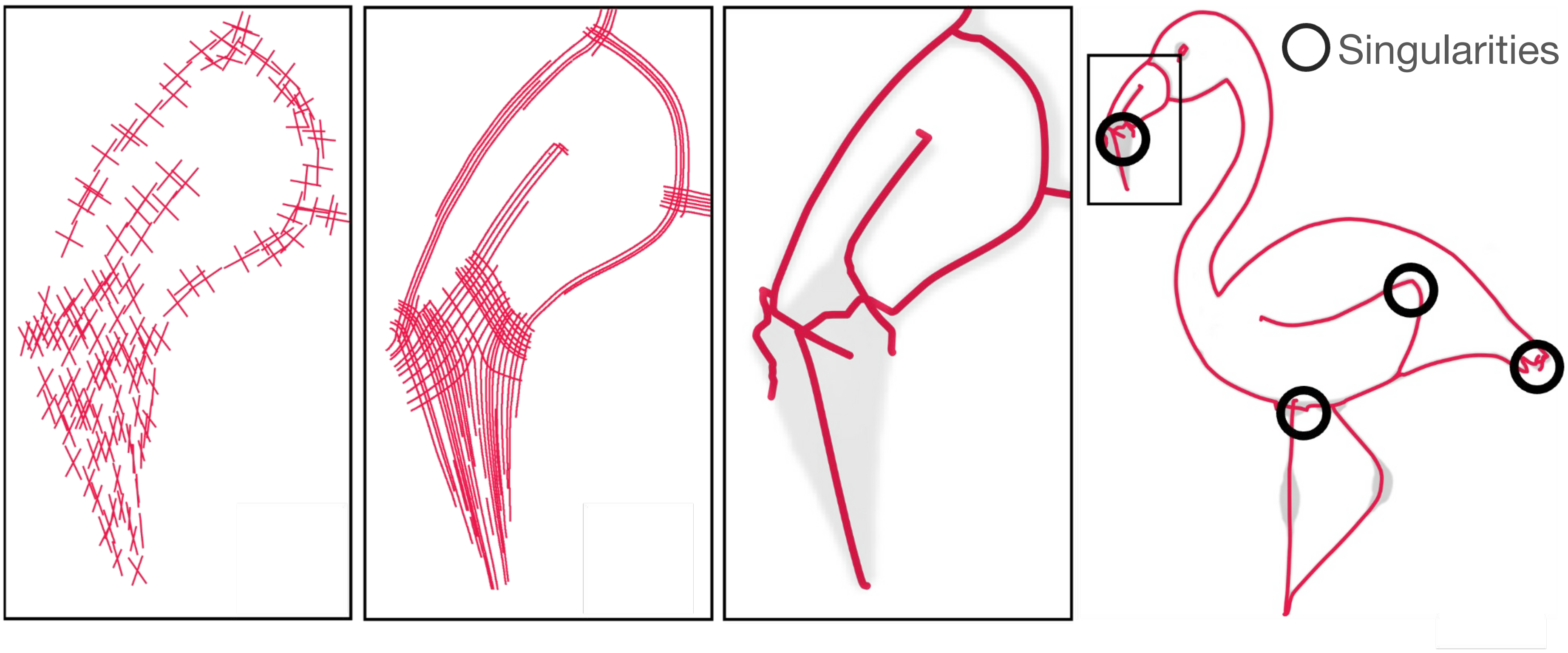


Polyvector Singularities: Zero Singularity Details

Representative vector: nonzero counterclockwise rotation



Tracing Errors due to Singularities

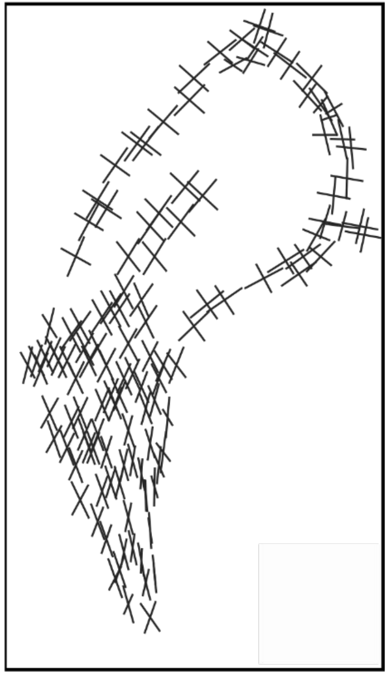


[Bessmeltsev and Solomon 2019]

Contribution

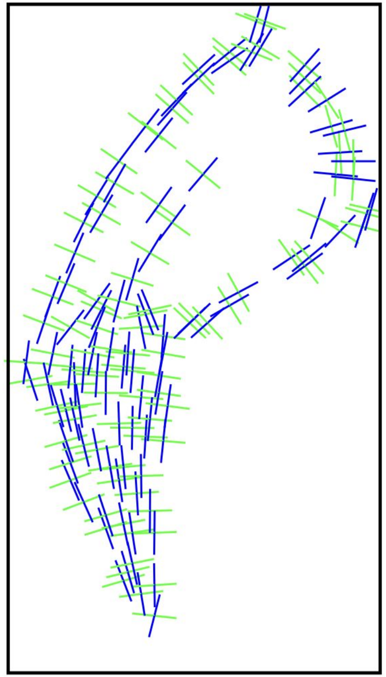
Contribution

Past Work



Frame Fields with Singularities

Our Method

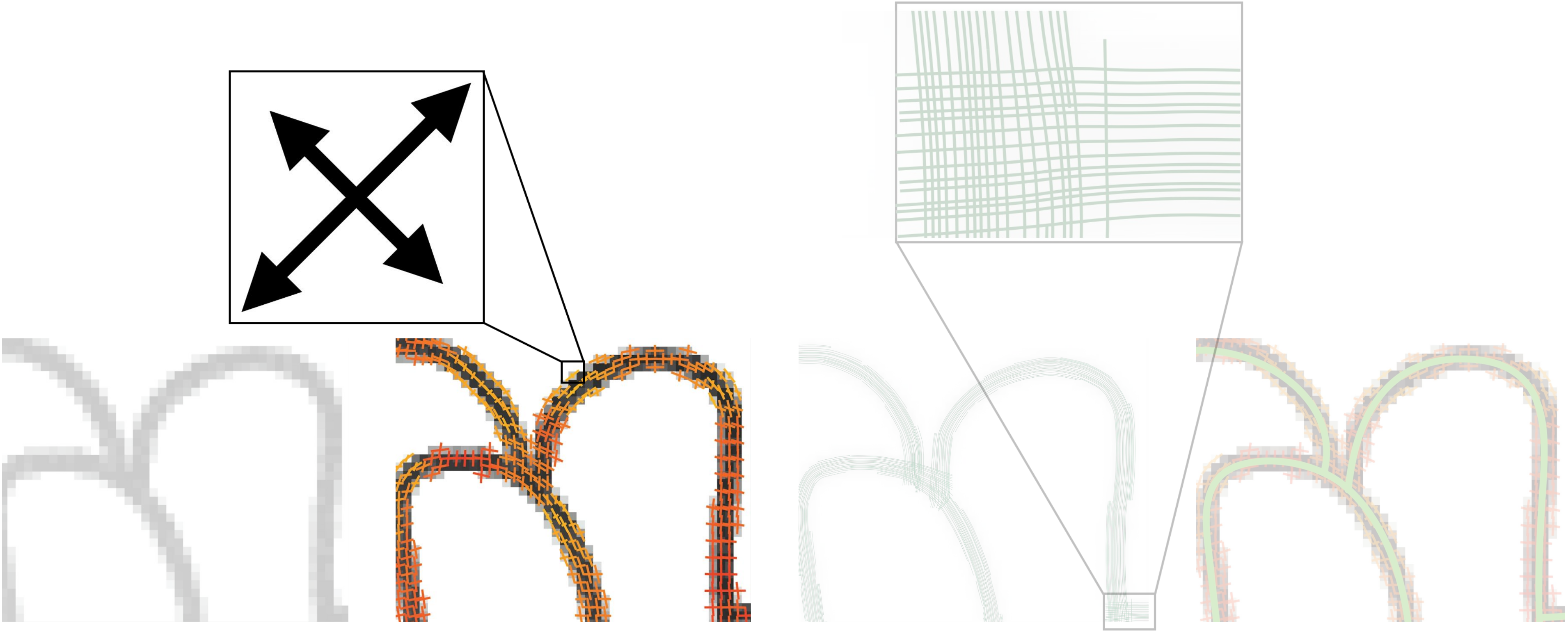


Singularity-Free Frame Fields

Guarantees:

- **No singularities!**
- **Qualitatively Improved Fields**
  – Even when starting singularity-free

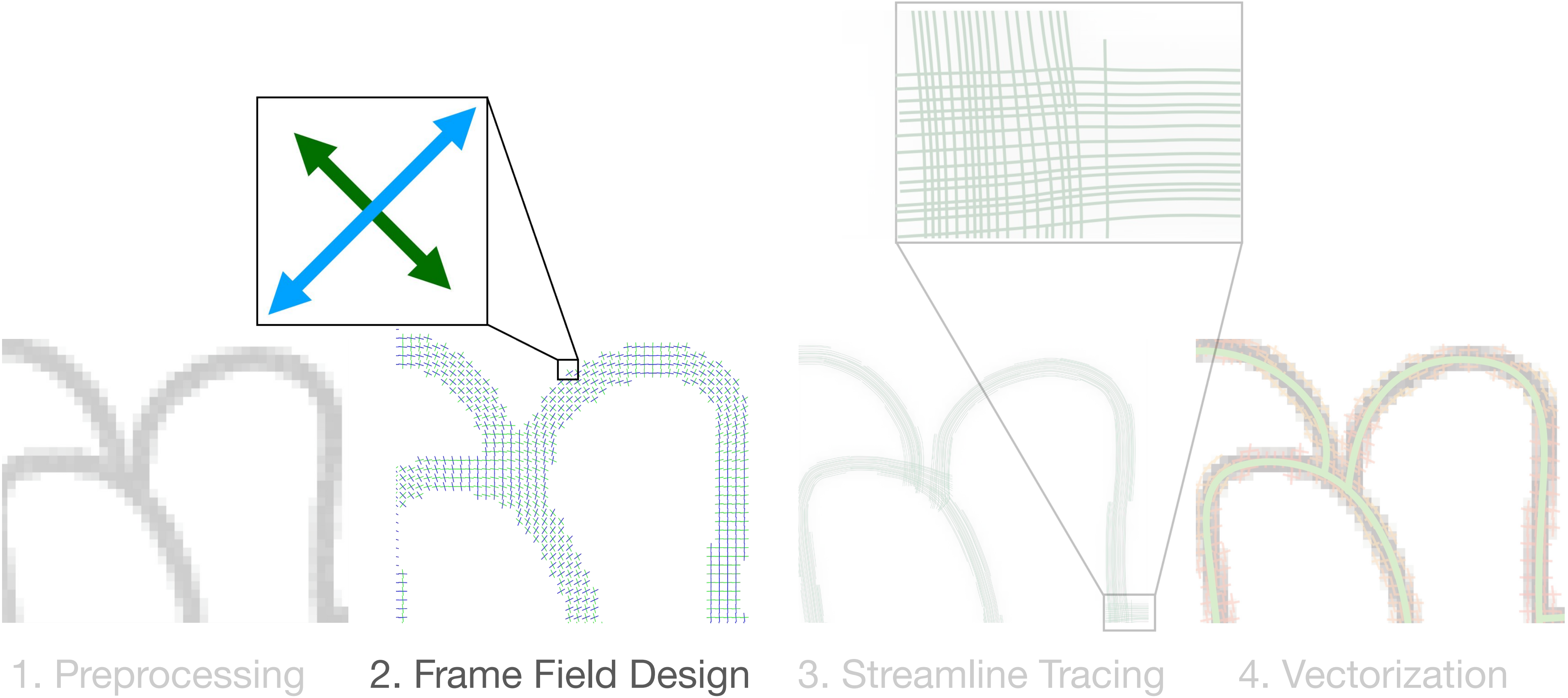
Current Vectorization Pipeline: Polyvector Fields



1. Preprocessing 2. Frame Field Design 3. Streamline Tracing 4. Vectorization

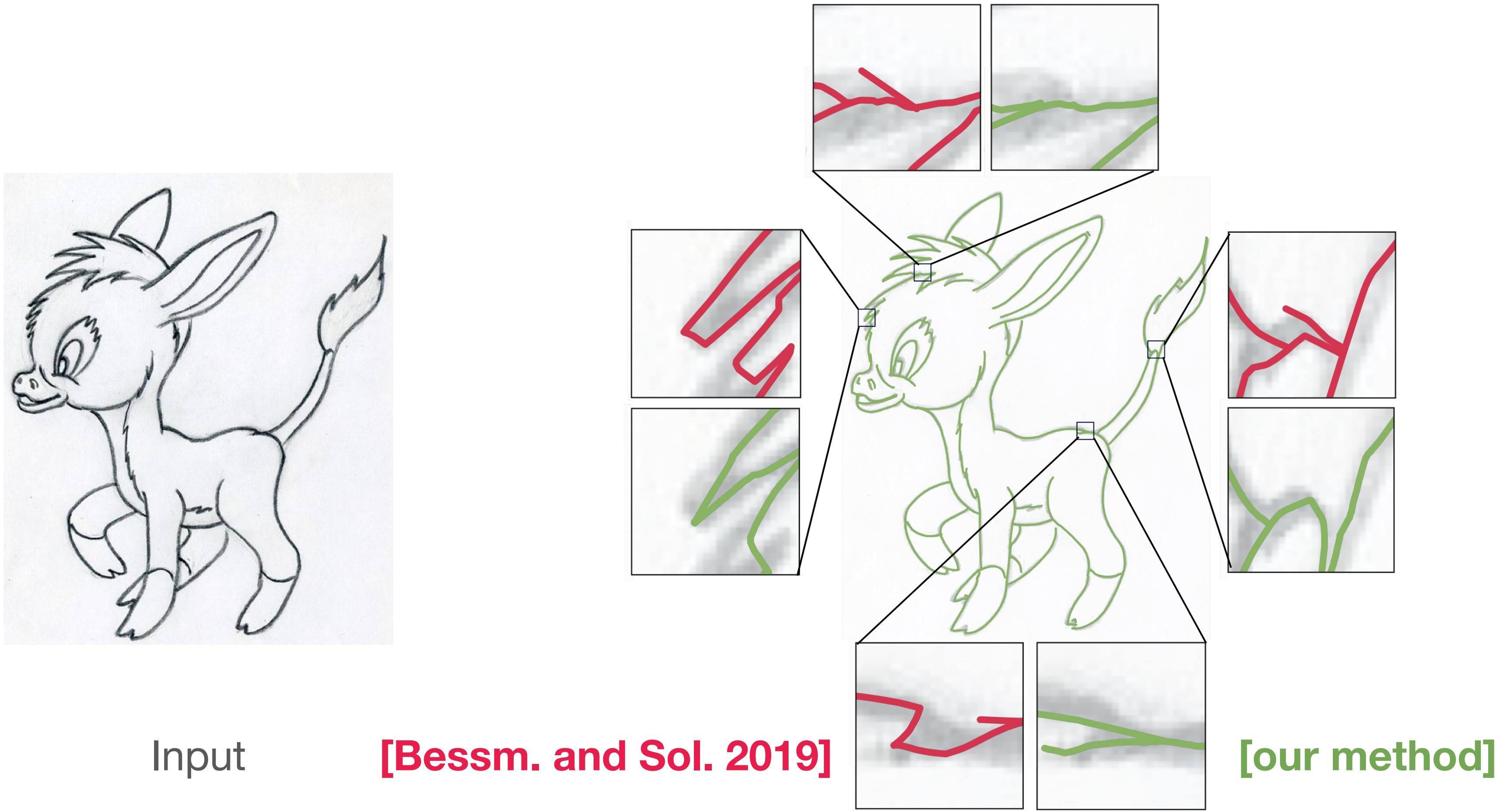
With singularities

Our Vectorization Pipeline: Singularity-Free Fields

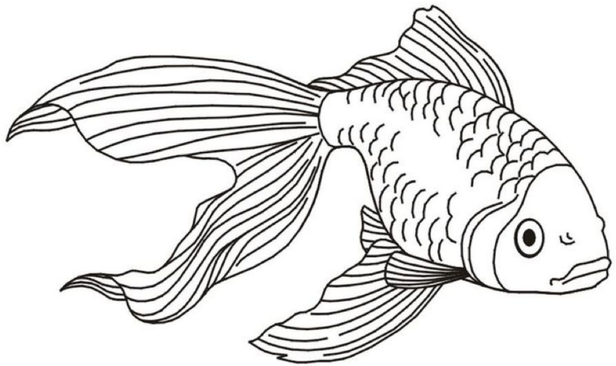


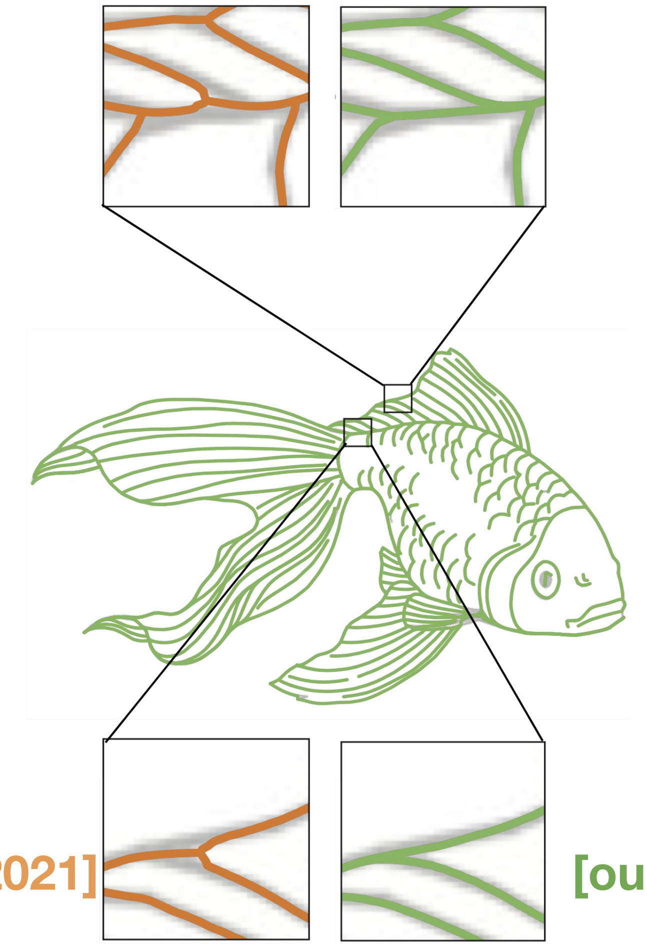
With NO singularities

Our Method: Vectorizations Closer to Input Drawings



Our Method: Correctly Resolves Y-Junctions



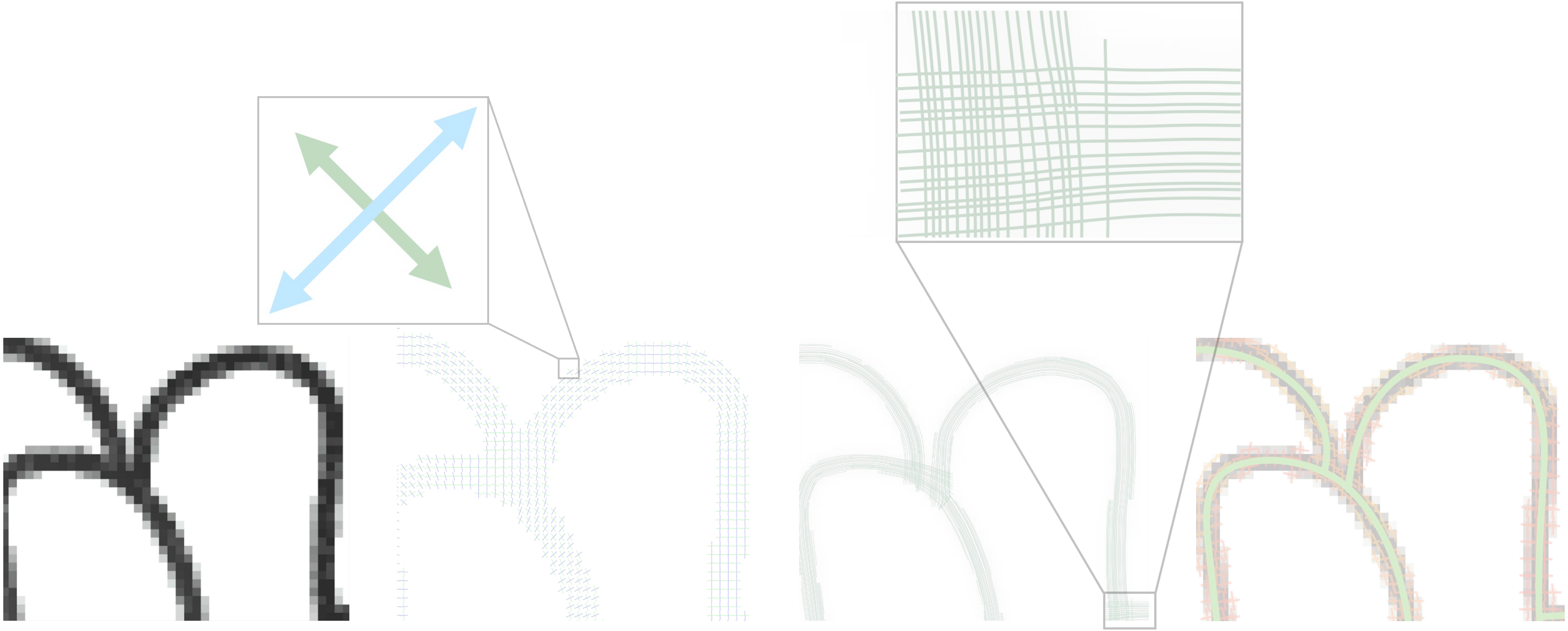


Input [Puhachov et al. 2021] [our method]

Background

Domain | Poincaré-Hopf Theorem | Boundary Indices | Discrete Trivial Connections

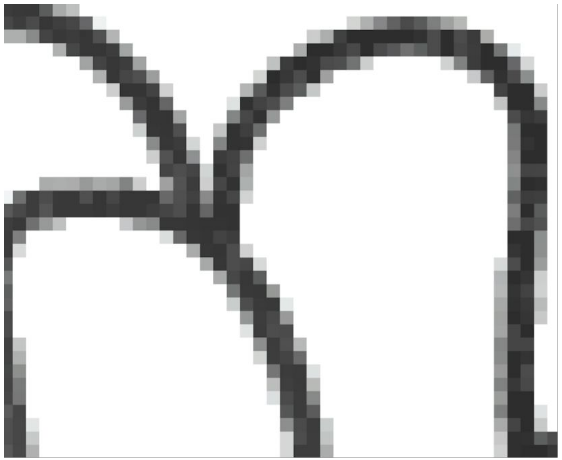
Vectorization Pipeline



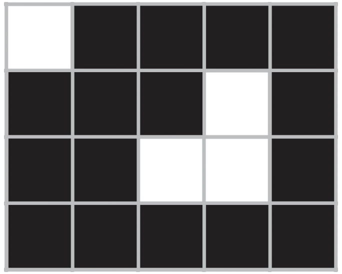
1. Preprocessing   2. Frame Field Design   3. Streamline Tracing   4. Vectorization

Background: **Domain** | Poincaré-Hopf Theorem | Boundary Indices | Discrete Trivial Connections

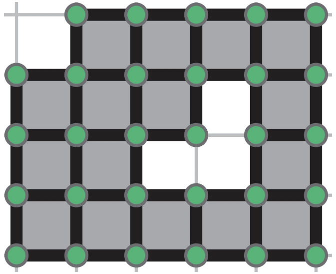
Primal and Dual Mesh



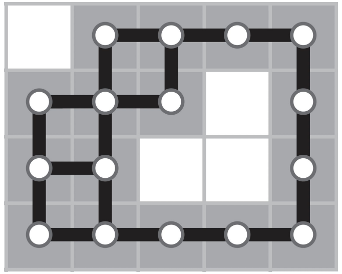
Preprocessing



Narrowband

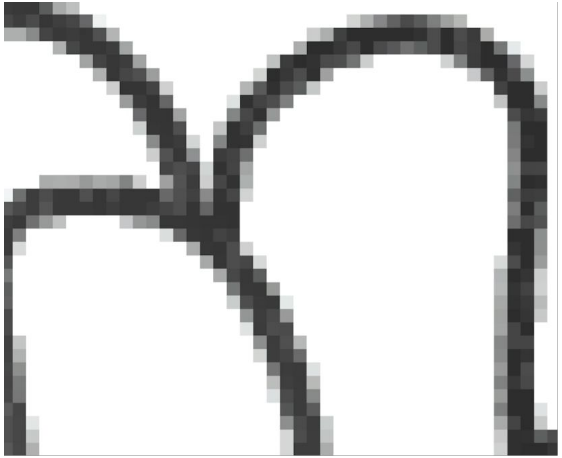


Primal Mesh

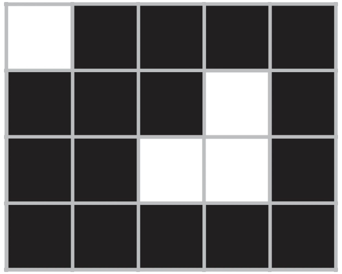


Dual Mesh

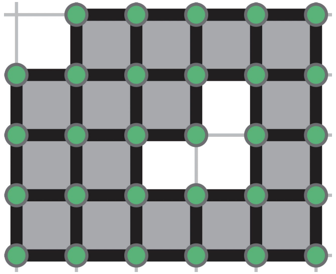
Primal and Dual Mesh + Discrete Differential Operators



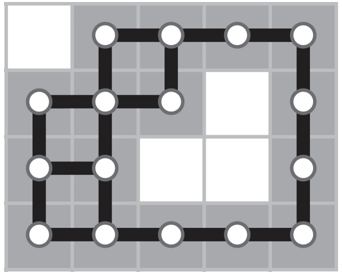
Preprocessing



Narrowband



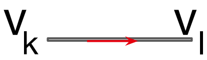
Primal Mesh



Dual Mesh

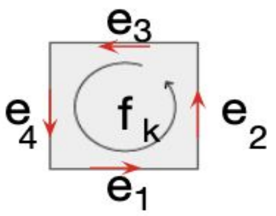
Discretized gradient

$d_0 = v_l - v_k$

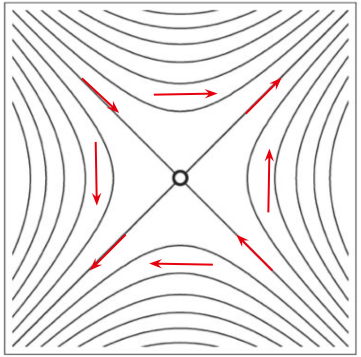


Discretized curl

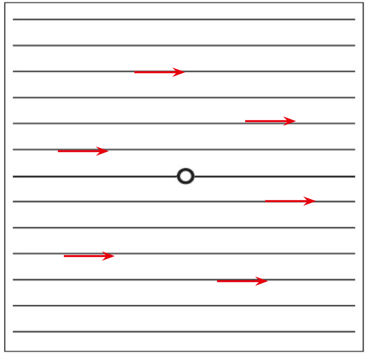
$d_1 = e_1 + e_2 - e_3 - e_4$



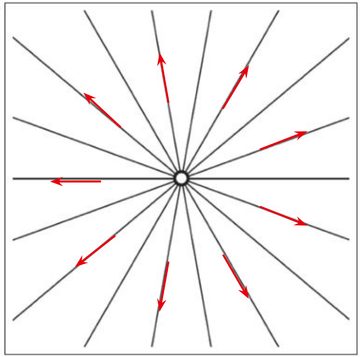
Indices of Singularities



-1

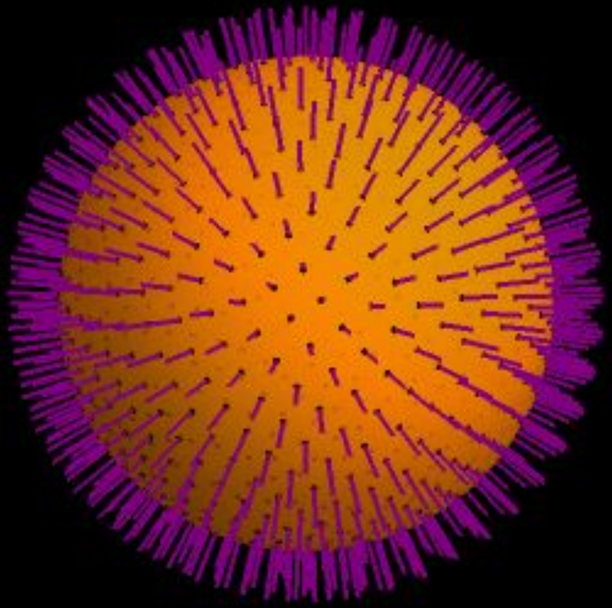


0



+1

Background: Poincaré-Hopf Theorem (χ = 2)



Source: Jacopo Bertolotti

Poincaré-Hopf Theorem: Broadly

Theorem:

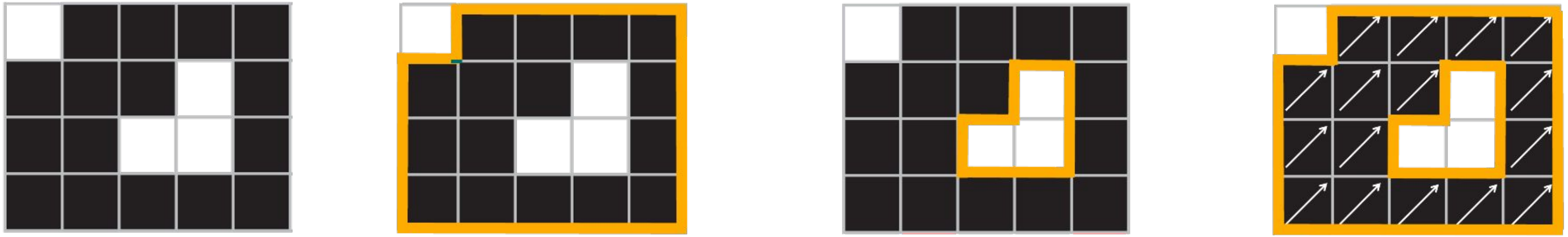
A smooth vector field on a manifold satisfies the following index formula:

$$\sum I_p + \sum I_{\partial_i} = 2\pi\chi = 2\pi(2 - 2g - b)$$

Singularity indices

Winding around the boundaries

Poincaré-Hopf Theorem: 2D + Boundaries



Narrowband | Boundaries | No singularities

Poincaré-Hopf Theorem: 2D + Boundaries

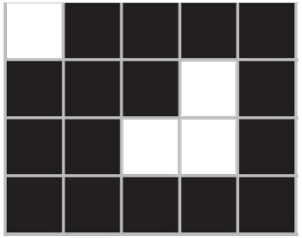
Sum of Boundary Indices

Euler characteristic

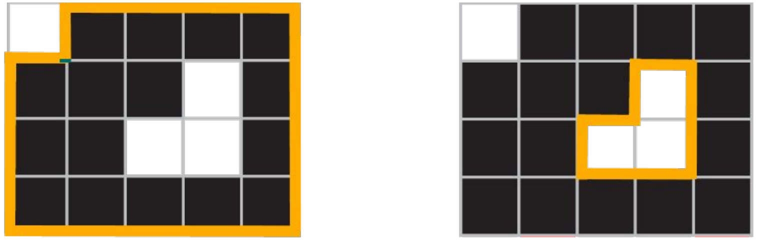
Number of Boundaries

$$\sum_{i=1}^{n} \mathrm{ind}_\alpha((\partial\Omega)_i) = \chi(\Omega) = 2 - n$$

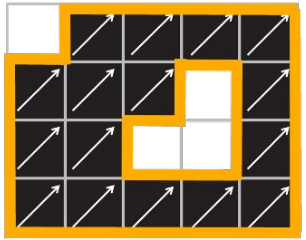
Push winding to boundaries



Narrowband



Boundaries



No singularities

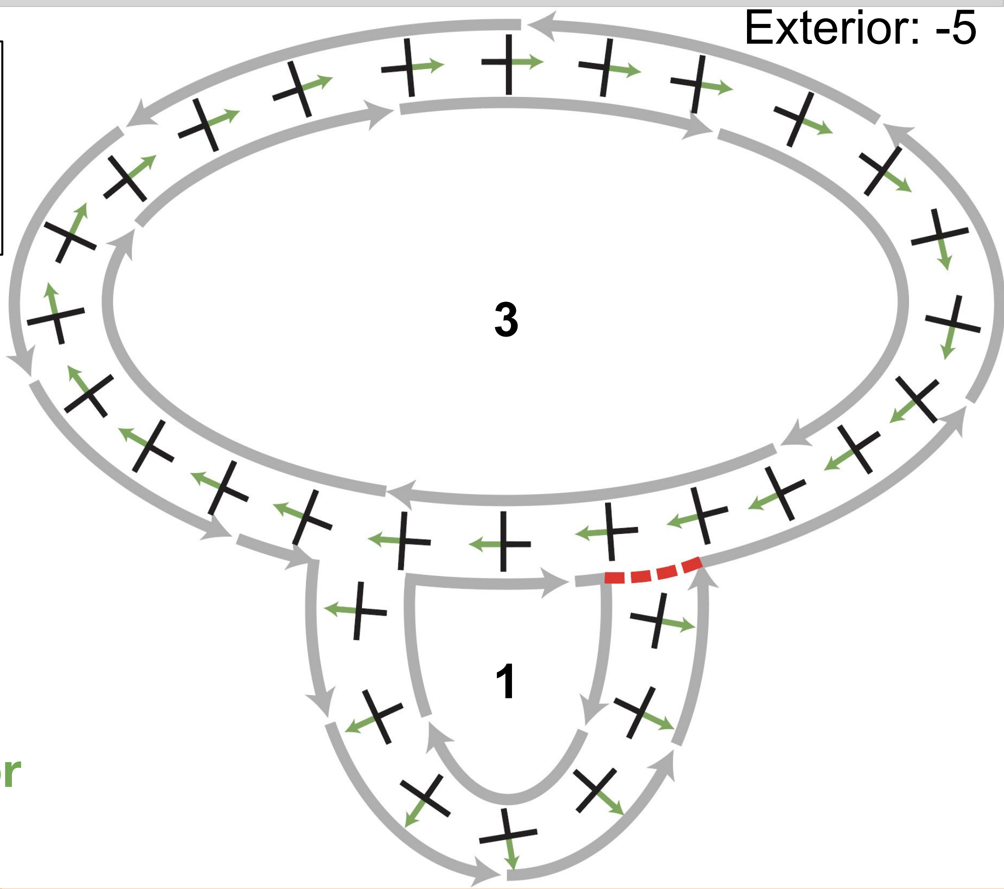
Boundary Indices Examples

In our work, integer indices only,
because:
rotations of <u>representative vector</u>

Process:

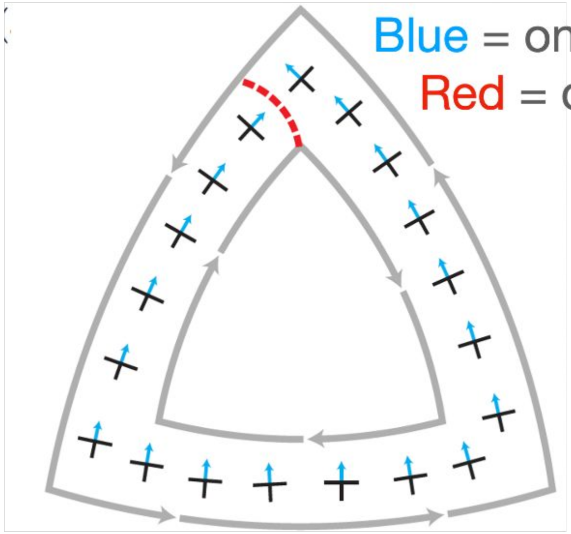
1. Find component vector rotation
2. Multiply by 4
3. Obtain representative vector rotation

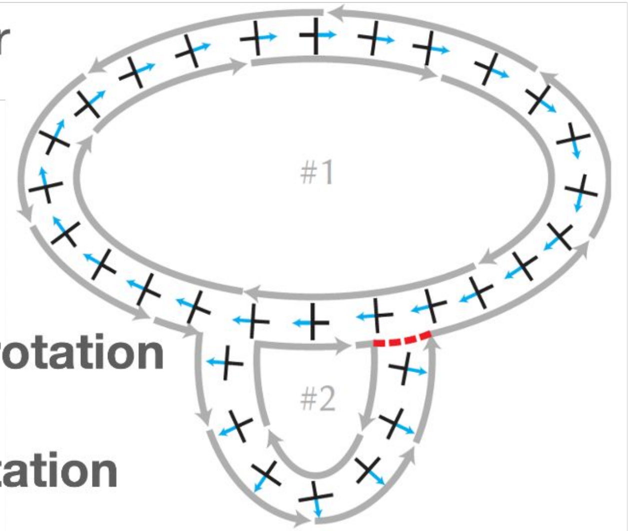
$1 + 3 - 5 = 2 - 3$ ⟷ Number of Boundaries



Green: one particular component vector
Red: discontinuity in the matching

Boundary Indices Framework (If needed)





Blue = one particular component vector
Red = discontinuity in the matching

Process:

1. **Find component vector rotation**
2. **Multiply by 4**
3. **Obtain the rep vector rotation**

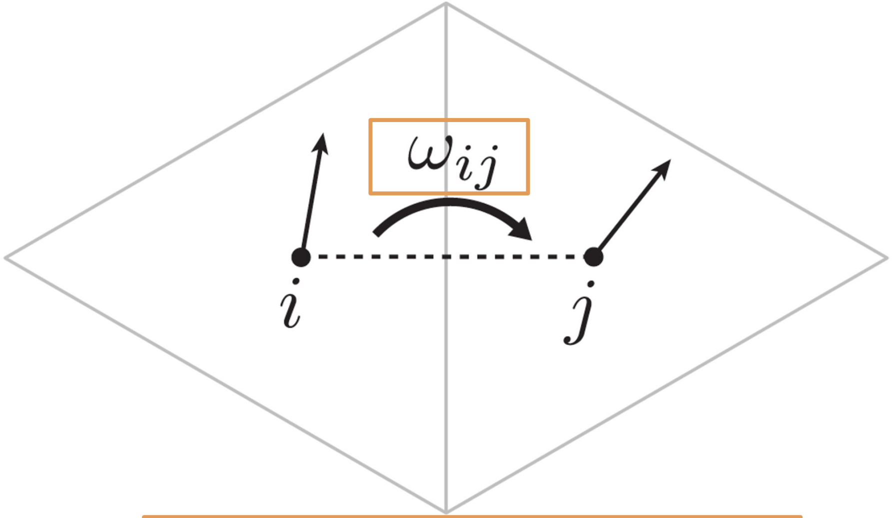
$\chi = 0$	Ext	Int
Cpt rotation	$\frac{\pi}{2}$	$-\frac{\pi}{2}$
Rep rotation	2π	-2π
Index	0	0

In our work, integer indices only

$\chi = 1$	Ext	Int_1	Int_2
Cpt rotation	3π	-2π	$-\pi$
Rep rotation	12π	-8π	-4π
Index	-5	3	1

Trivial Connections: On Triangle Mesh

Intuition: change in angle of field along each edge

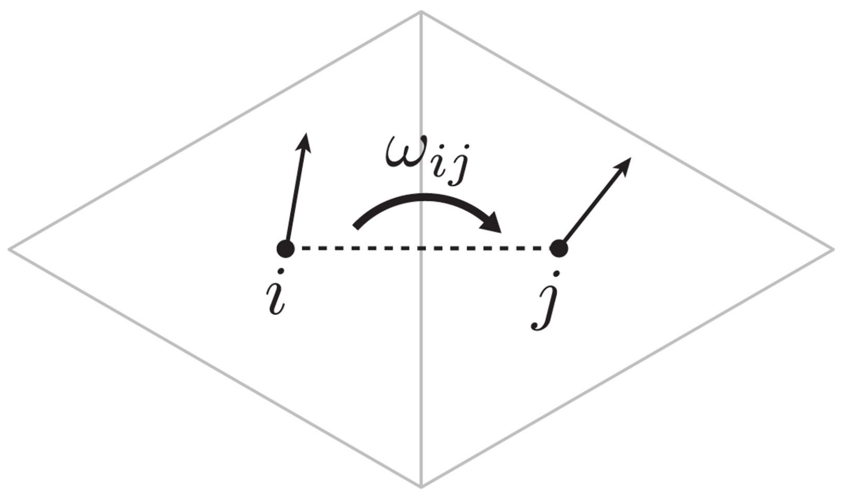


Rotation angle on dual edge

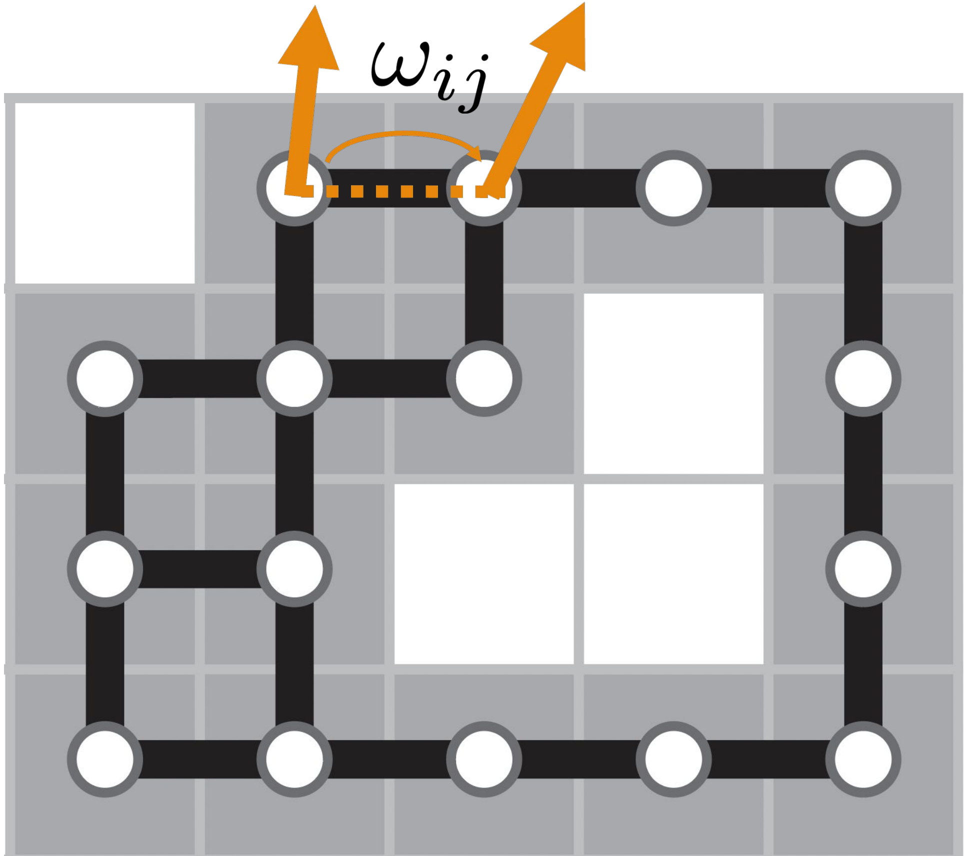
Work with **derivative** of field, rather than field itself ⇒ index constraints become linear

[Crane et al. 2013]

Trivial Connections: On Pixelated Planar Domain



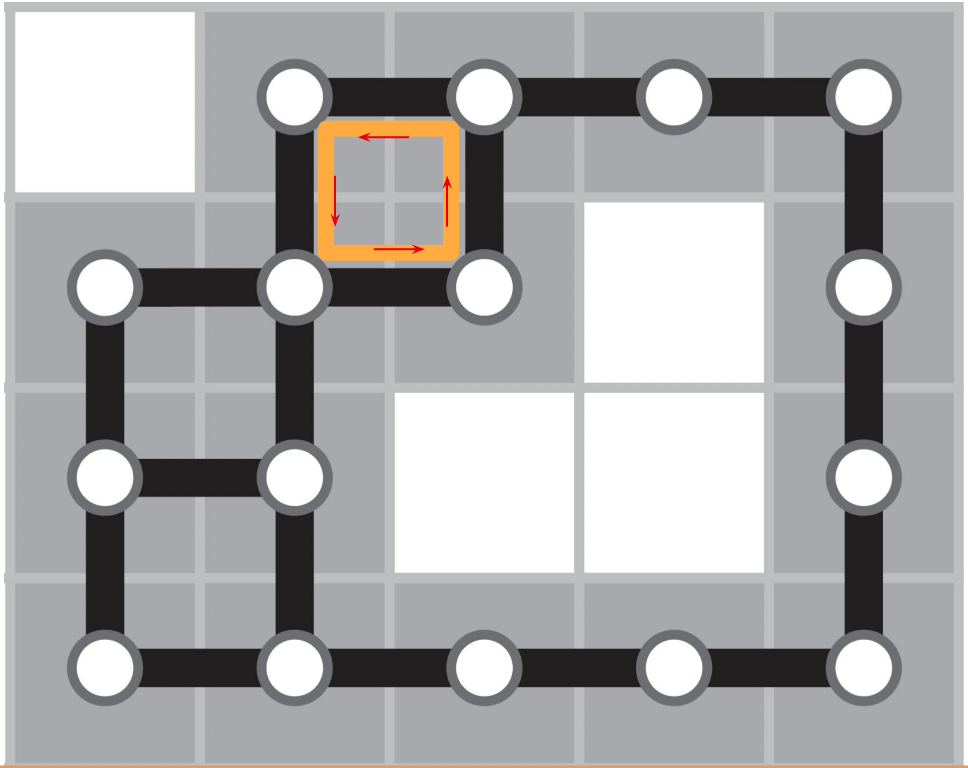
[Crane et al. 2013]



Express Index Constraints as Linear System

(Submatrix of)
Discrete Gradient Operator

$$A = \begin{bmatrix} d_0^{\text{int}} \\ H \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \mathbf{k} \end{bmatrix}$$



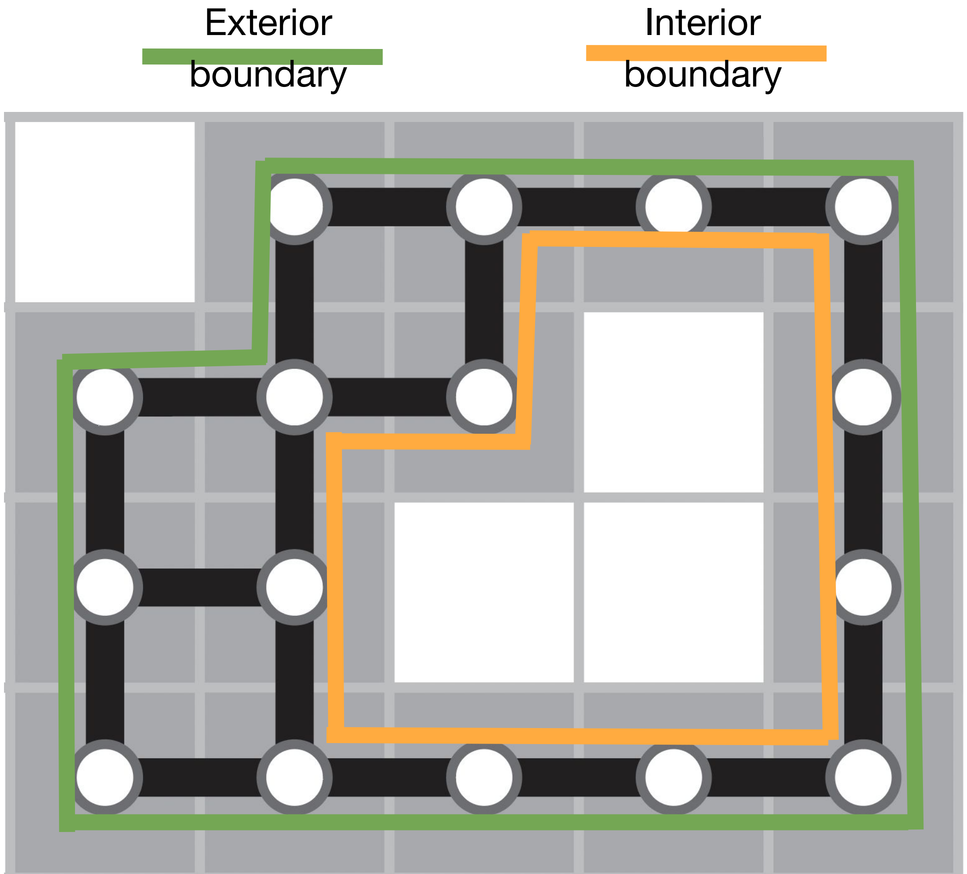
No singularities on the domain interior

Express Index Constraints as Linear System

(Submatrix of)
Discrete Gradient Operator

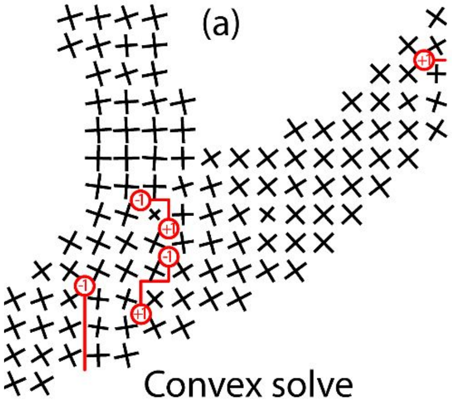
$$A = \begin{bmatrix} d_0^{\text{int}} \\ H \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \mathbf{k} \end{bmatrix}$$

Describes the boundary winding



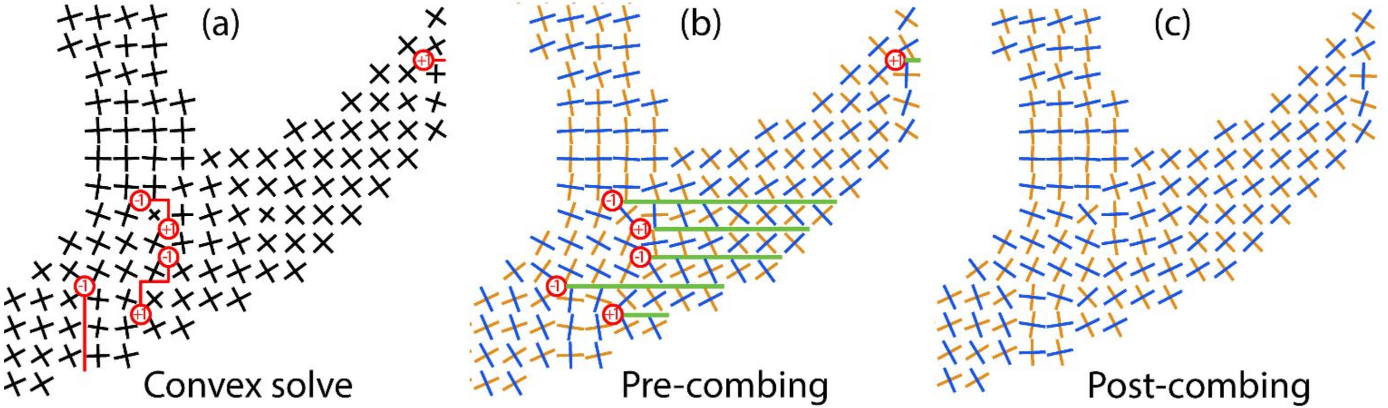
Next: Method

Method Overview



An initial quadratic solve for a roughly-aligned cross field with singularities

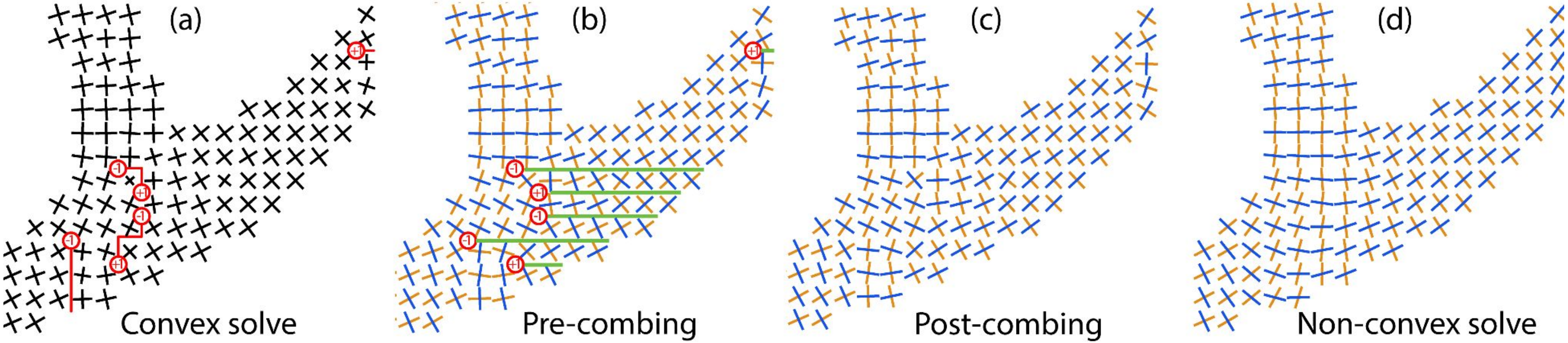
Method Overview



An initial quadratic solve for a roughly-aligned cross field with singularities

An identification and combing of the singularities

Method Overview



An initial quadratic solve for a roughly-aligned cross field with singularities

An identification and combing of the singularities

A non-convex solve to obtain a smoother, better-aligned non-orthogonal singularity-free field

Convex Initialization

- Linear solve to optimize a quadratic energy
- Produce a cross field aligned with the drawing, but likely having singularities

$$\min_{f \in \mathbb{C}^{|F|}} \left(\frac{1}{A_S} f^\top \Delta \bar{f} + \frac{w}{A_A} \sum_{i : g_i \neq 0} |f_i - g_i^4|^2 \right)$$

Convex Initialization

- Linear solve to optimize a quadratic energy
- Produce a cross field aligned with the drawing, but likely having singularities

$$\min_{f \in \mathbb{C}^{|F|}} \left(\frac{1}{A_S} f^\top \Delta \overline{f} + \frac{w}{A_A} \sum_{i: g_i \neq 0} |f_i - g_i^4|^2 \right)$$

Dirichlet energy to measure field smoothness

Convex Initialization

- Linear solve to optimize a quadratic energy
- Produce a cross field aligned with the drawing, but likely having singularities

$$\min_{f \in \mathbb{C}^{|F|}} \left(\frac{1}{A_S} f^{\top} \Delta \overline{f} + \frac{w}{A_A} \sum_{i : g_i \neq 0} |f_i - g_i^4|^2 \right)$$

Dirichlet energy to measure field smoothness

Expresses alignment to the image gradient

Convex Initialization

- Linear solve to optimize a quadratic energy
- Produce a cross field aligned with the drawing, but likely having singularities

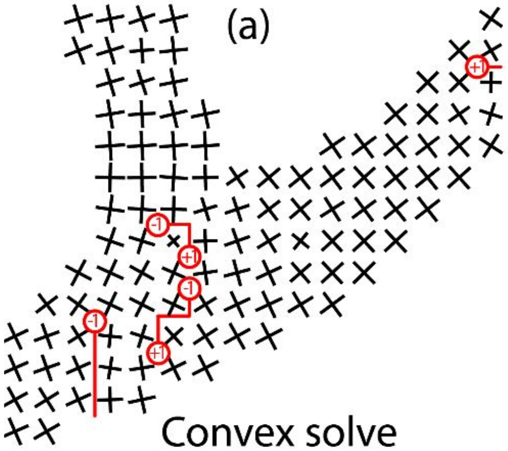
$$\min_{f \in \mathbb{C}^{|F|}} \left(\frac{1}{A_S} f^\top \Delta \overline{f} + \frac{w}{A_A} \sum_{i: g_i \neq 0} |f_i - g_i^4|^2 \right)$$

Dirichlet energy to measure field smoothness

Expresses alignment to the image gradient

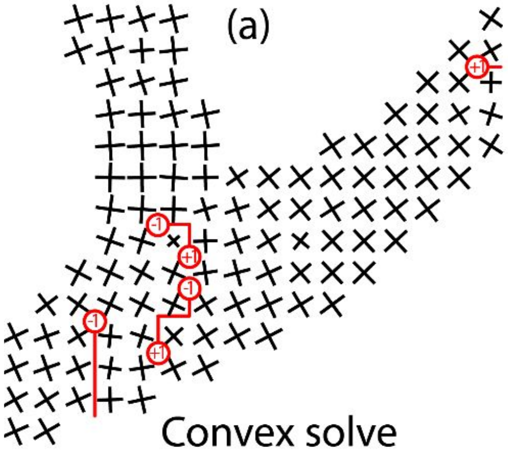
- Quadratic optimization results in a linear KKT system which is solved via conjugate gradients

Convex Initialization



Aligns well in most regions

Convex Initialization

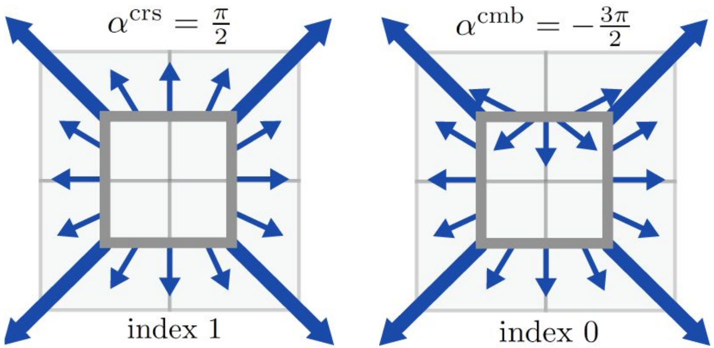


Aligns well in most regions

But many internal singularities, often arising at junctions and curve endpoints

How to comb singularities?

- Push singularities by modifying interpolation



- When we push a singularity over an edge *i j*

$$\alpha_{ij}^{cmb} = \begin{cases} \alpha^{crs} \mp 2\pi & \text{if } ij \text{ is with edge orientation} \\ \alpha^{crs} \pm 2\pi & \text{if } ij \text{ is against edge orientation} \end{cases}$$

How much does pushing singularities cost?

- Push of a positive/negative singularity leads to an increase in smoothness by

  $(\alpha^{crs} \mp 2\pi)^2 - (\alpha^{crs})^2 = 4\pi\,(\pi \mp \alpha^{crs})$

- Two directed graphs for pushing positive and negative singularities:

$$w_{ij}^{\pm} = \begin{cases} (|f_i| + |f_j|)(\pi \mp \alpha^{crs}) & \text{if } ij \text{ interior, with orientation} \\ (|f_i| + |f_j|)(\pi \pm \alpha^{crs}) & \text{if } ij \text{ interior, against orientation} \\ \infty & \text{if } ij \text{ boundary} \end{cases}$$

- Push +/- singularities to either match each other or go the boundaries
- These weights give us the costs for our optimal transport problem

Optimal transport problem

- We have a transport matrix $T \in \mathbb{R}^{(|\mathcal{S}^-|+1)\times(|\mathcal{S}^+|+1)}$
- Unbalanced transport problem due to the boundary acting as infinite source and sink
- The linear program solves for T

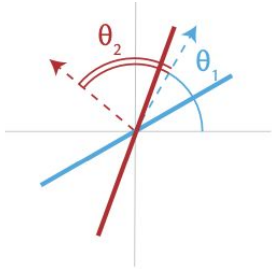
$$
\begin{aligned}
&\min_T \langle C, T\rangle \\
&s.t. \sum_j T_{ij} = 1, \text{ for } 1 \le i \le |\mathcal{S}^-| \\
&\quad \sum_i T_{ij} = 1, \text{ for } 1 \le j \le |\mathcal{S}^+| \\
&\quad T_{ij} \ge 0, \ \forall i, j.
\end{aligned}
$$

- This optimization problem has a binary solution $T_{ij} \in \{0,1\}^{(|\mathcal{S}^-|+1)\times(|\mathcal{S}^+|+1)}$
- Partially inspired by Justin Solomon and Amir Vaxman's work[1]

1. Solomon and Vaxman 2019

Nonconvex field optimization

- Our representation: two line fields
- Encode the frame field as three unknowns
  - ϕ: absolute rotation of the root for each component
  - α: connection angle per dual edge
  - θ_2: relative angles between the representative vectors



- Linear index constraints: $Ax=b$

$$A = \begin{bmatrix} d_0^{\text{int}} \\ H \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \mathbf{k} \end{bmatrix}$$

Objective function

$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} \mathrm{E}_{\text{Alignment}}$$

Objective function

$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} \mathrm{E}_{\text{Alignment}} + \frac{w_B}{A_B} \mathrm{E}_{\text{Barrier}}$$

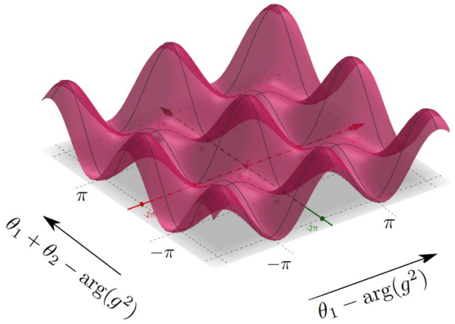
Objective function

$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} \mathrm{E}_{\mathrm{Alignment}} + \frac{w_B}{A_B} \mathrm{E}_{\mathrm{Barrier}} + \frac{w_S}{A_S} \mathrm{E}_{\mathrm{Smoothness}}$$

Alignment

$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} E_{\text{Alignment}} + \frac{w_B}{A_B} E_{\text{Barrier}} + \frac{w_S}{A_S} E_{\text{Smoothness}}$$

Alignment



Local minimum achieved when one of the line fields is:

- Parallel to the local tangent, or
- Perpendicular to the normalized image gradient g

Smoothness

$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} \mathrm{E}_{\mathrm{Alignment}} + \frac{w_B}{A_B} \mathrm{E}_{\mathrm{Barrier}} + \frac{w_S}{A_S} \mathrm{E}_{\mathrm{Smoothness}}$$

Smoothness

$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} \mathrm{E}_{\text{Alignment}} + \frac{w_B}{A_B} \mathrm{E}_{\text{Barrier}} + \frac{w_S}{A_S} \mathrm{E}_{\text{Smoothness}}$$

$$\mathrm{E}_{\text{Smoothness}} = \alpha^\top \alpha + \beta^\top \beta$$

Smoothness

$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} \mathrm{E}_{\mathrm{Alignment}} + \frac{w_B}{A_B} \mathrm{E}_{\mathrm{Barrier}} + \frac{w_S}{A_S} \mathrm{E}_{\mathrm{Smoothness}}$$

$$\mathrm{E}_{\mathrm{Smoothness}} = \alpha^\top \alpha + \beta^\top \beta$$

$$\beta_{ij} = (\theta_2)_j - (\theta_2)_i + \alpha_{ij}$$

⇩

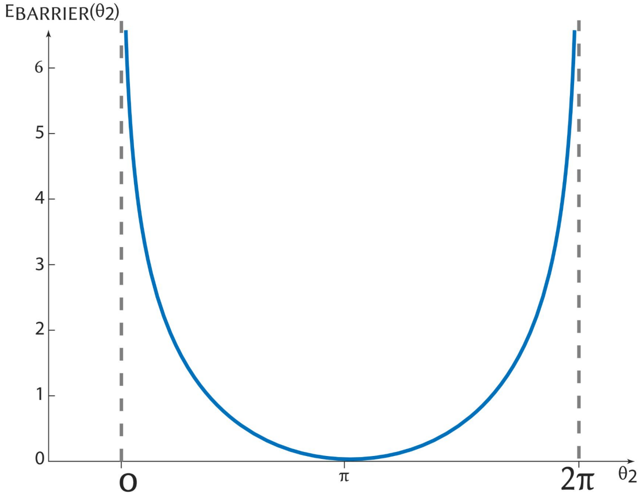
Change in the second line field over each dual edge

Barrier

$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} \mathrm{E}_{\text{Alignment}} + \frac{w_B}{A_B} \mathrm{E}_{\text{Barrier}} + \frac{w_S}{A_S} \mathrm{E}_{\text{Smoothness}}$$

Barrier

$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} E_{\text{Alignment}} + \frac{w_B}{A_B} E_{\text{Barrier}} + \frac{w_S}{A_S} E_{\text{Smoothness}}$$



Optimization method and setting

- Here are some of the parameters we used: $w_B = 0.5$, $w_S = 5$, $w = 0.125$

$$\min_{f \in \mathbb{C}^{|F|}} \left(\frac{1}{A_S} f^\top \Delta \bar{f} + \frac{w}{A_A} \sum_{i: g_i \neq 0} |f_i - g_i^4|^2 \right)$$

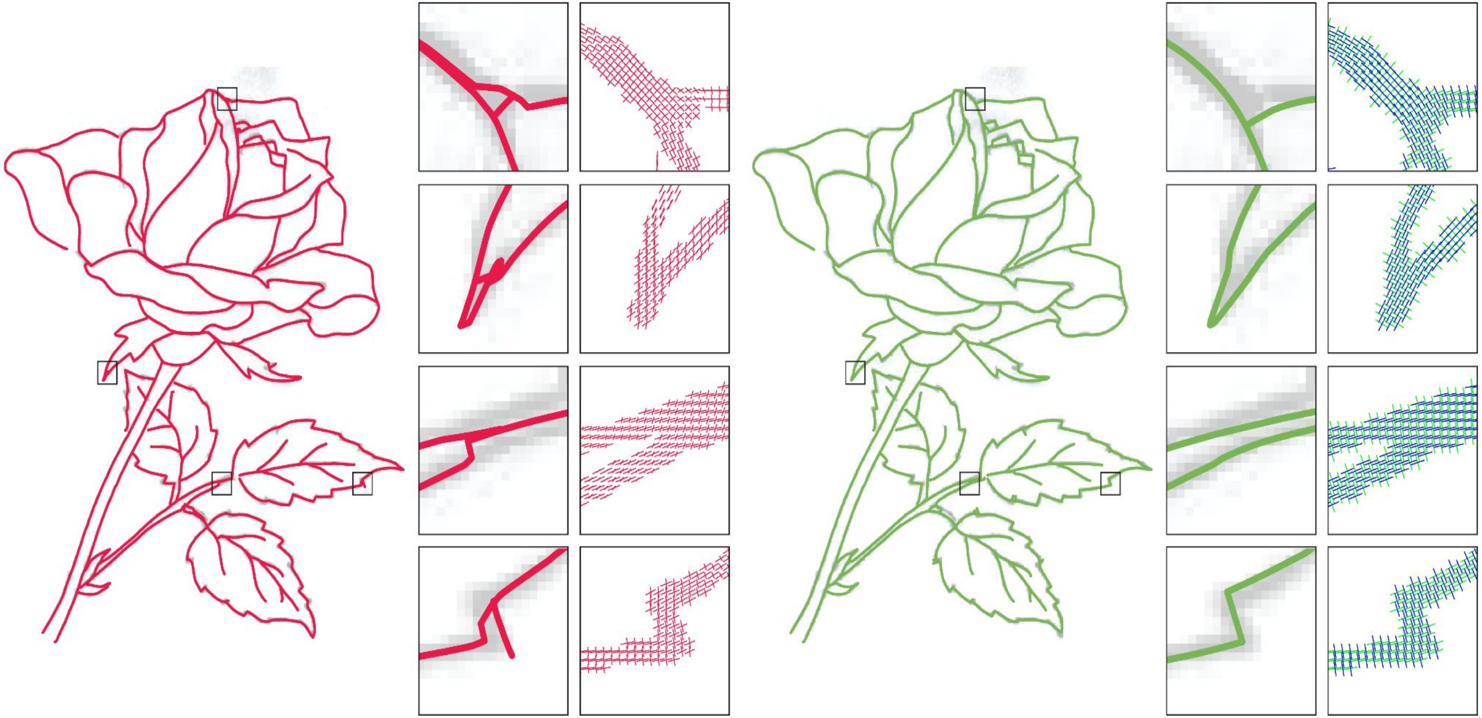
$$f(\phi, \alpha, \theta_2) = \frac{1}{A_A} \mathrm{E}_{\mathrm{Alignment}} + \frac{w_B}{A_B} \mathrm{E}_{\mathrm{Barrier}} + \frac{w_S}{A_S} \mathrm{E}_{\mathrm{Smoothness}}$$

- Optimize the non-convex energy using Newton's method as implemented in IPOPT

Results

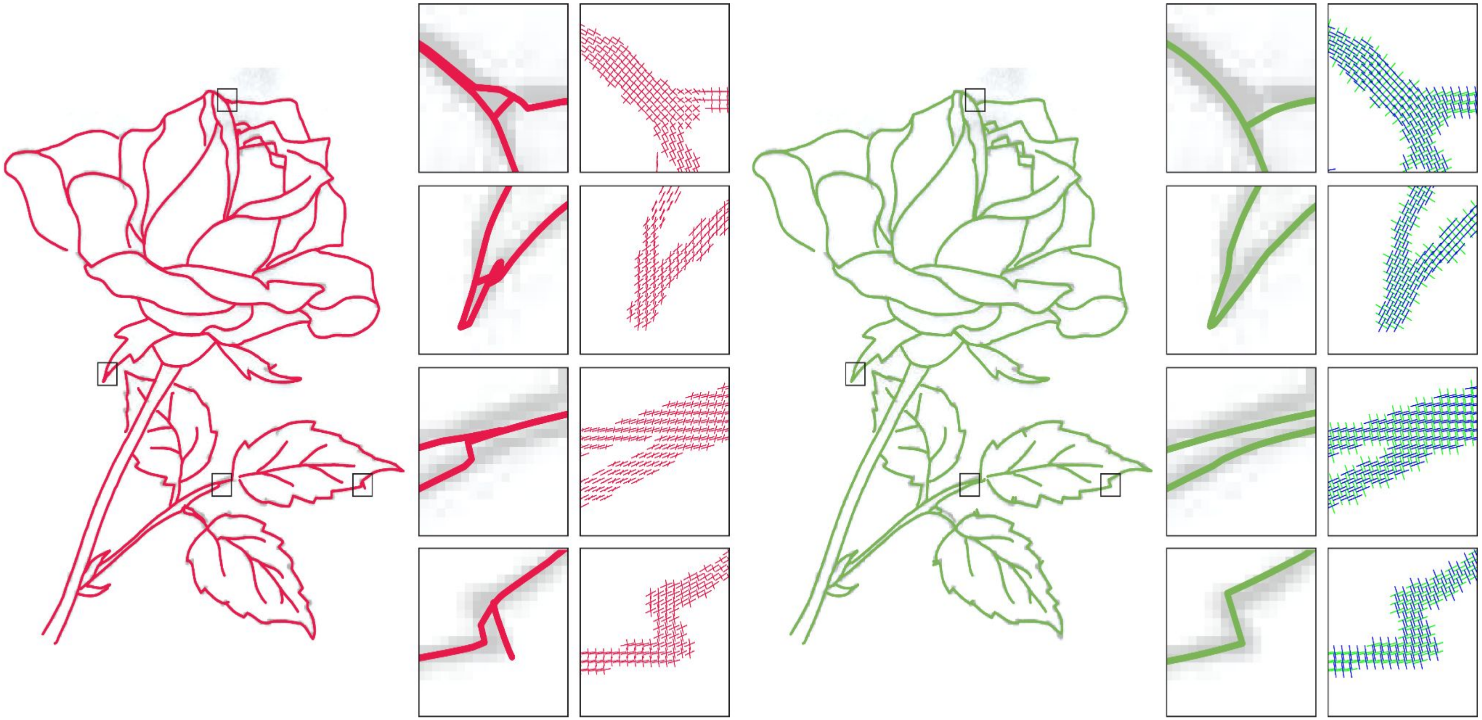
PolyVector and PolyFlow comparison

PolyVector comparison



Bessmeltsev and Solomon 2019

PolyVector comparison



Due to zero singularities:

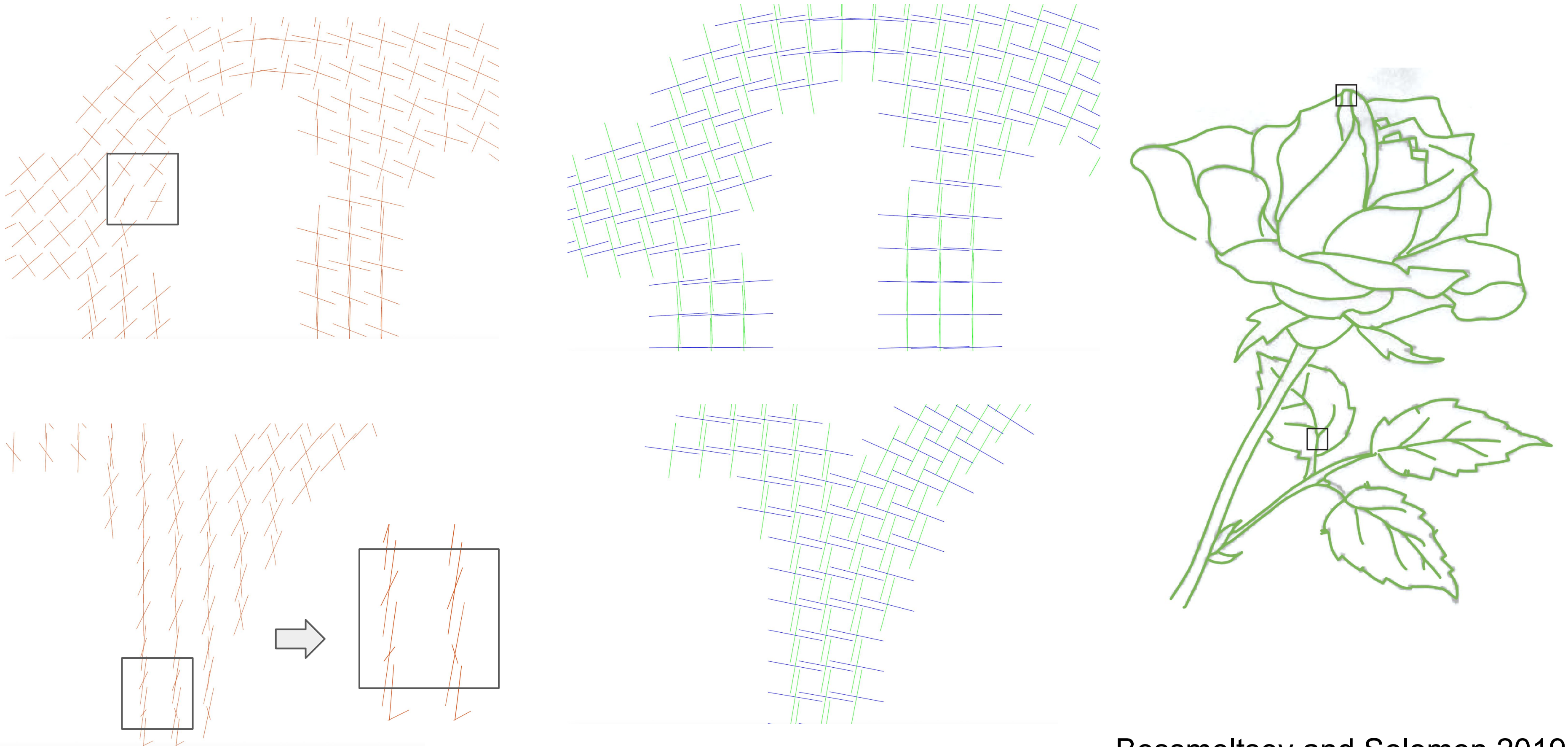
Tracing produces continuous streamlines

We no longer need heuristics

Fewer artifacts

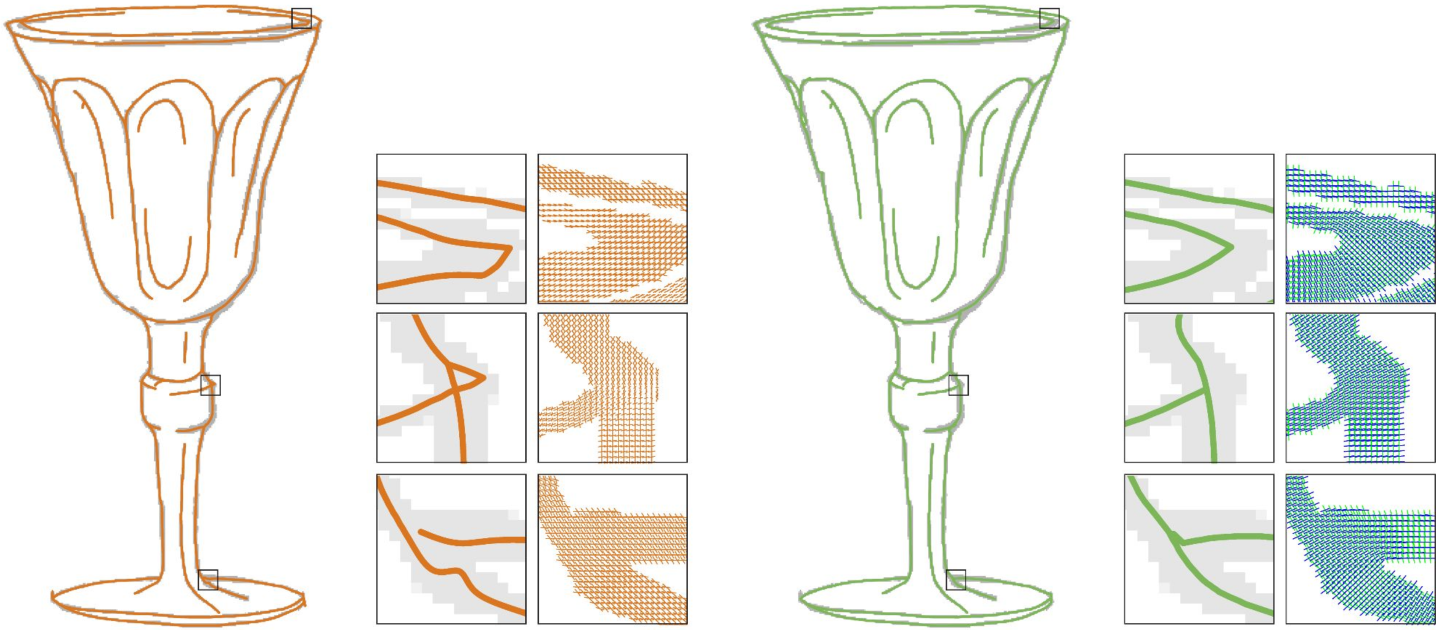
Bessmeltsev and Solomon 2019

PolyVector comparison: Field quality



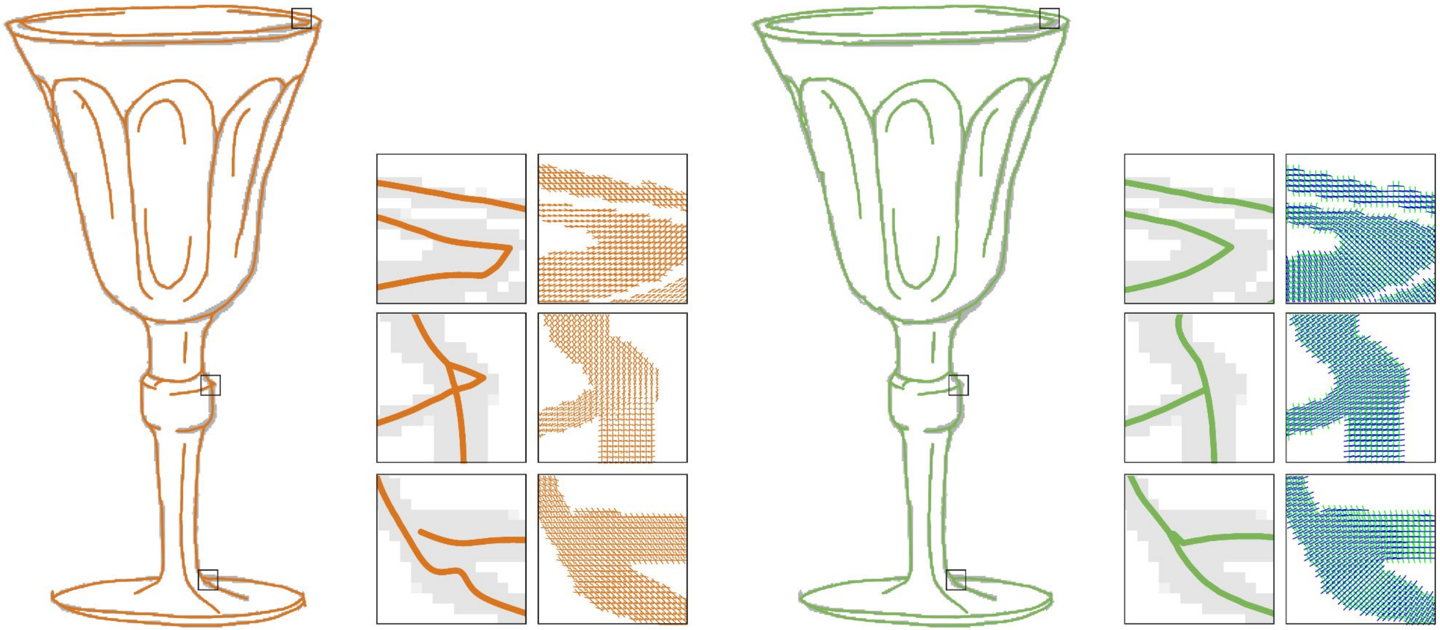
Bessmeltsev and Solomon 2019

PolyFlow comparison



Puhachov, Neveu, Chien, Bessmeltsev 2021

PolyFlow comparison

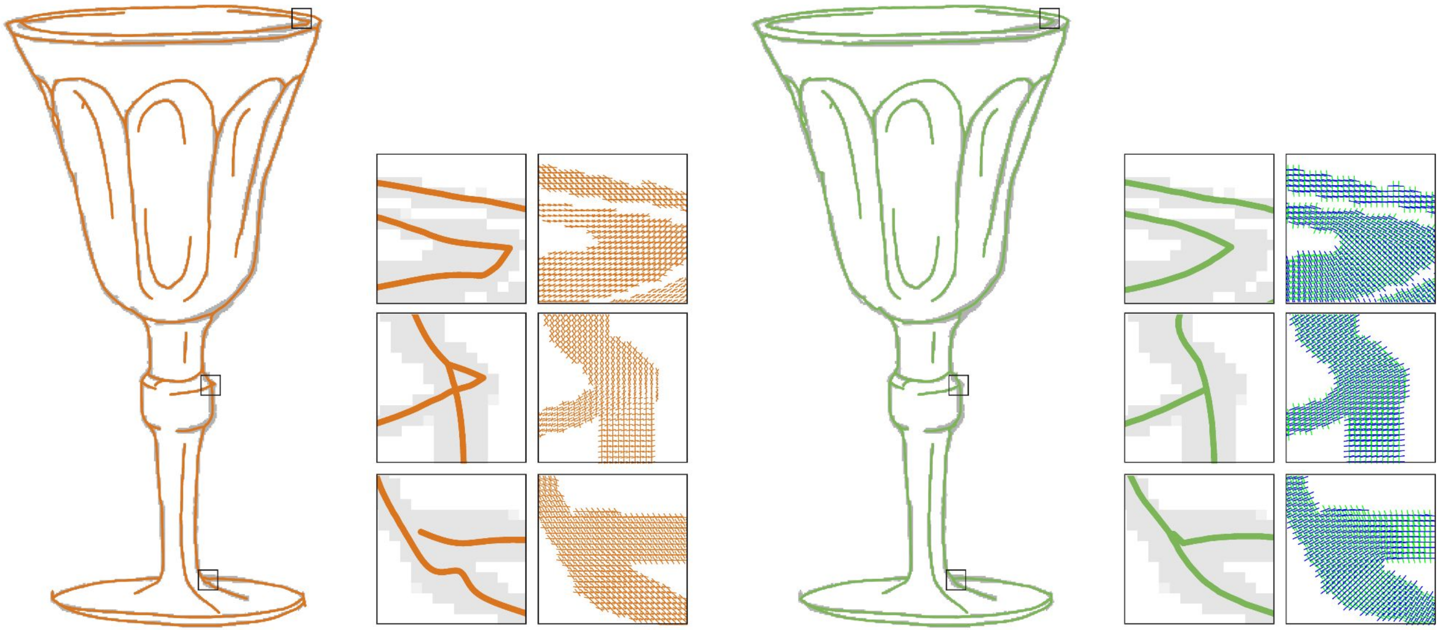


Our frame fields produce vectorizations more precise at:

- Junctions
- Thick strokes
- Fills

Puhachov, Neveu, Chien, Bessmeltsev 2021

Results: PolyFlow comparison



Also:

Improves resolution of Y-junctions

Correctly resolve ambiguous connectivity

Puhachov, Neveu, Chien, Bessmeltsev 2021

Processing time

	Fig.	[BS19] Sings	# Dark Pixels	Cvx Opt (s)	OT (s)	Ncvx Opt (s)	Vect time (s)
Cup	6, 8	14	41k	12	5	259	46
Sketch1	6	8	25k	8	11	125	24
Sketch2	7	24	51k	48	447	1217	260
Donkey	6	8	30k	13	3	778	45
Elephant	S	12	35k	18	8	465	41
Goldfish	8	4	39k	12	10	461	53
Hippo	S, 8	24	25k	9	10	208	27
Horse	S	34	103k	280	192	618	132
Kitten	S	0	29k	8	13	668	40
Pig	S	43	28k	10	10	389	144
Rose	7	0	19k	4	3	194	50
Trumpet	S	12	14k	2	1	148	12

Conclusions

Limitations and Future Work

- We work best on clean line drawings
- Improve relatively long runtimes with more sophisticated optimization techniques
- Improve upon the tracing pipelines
- Extend our work to different domains requiring higher-valence junctions
- Work on vectorization of complex hatching styles and fills
- Develop meaningful quantitative quality measures[1]
  – Chamfer and Hausdorff distances are insufficient

1. Puhachov, Neveu, Chien, Bessmeltsev 2021

Conclusions

- We generate singularity-free frame fields, missing from state-of-the-art frame field–based vectorization algorithms
- Make use of trivial connections + simple frame field representation + convex solve for the initialization
- No singularities inside the domain and improved frame fields → quality vectorizations

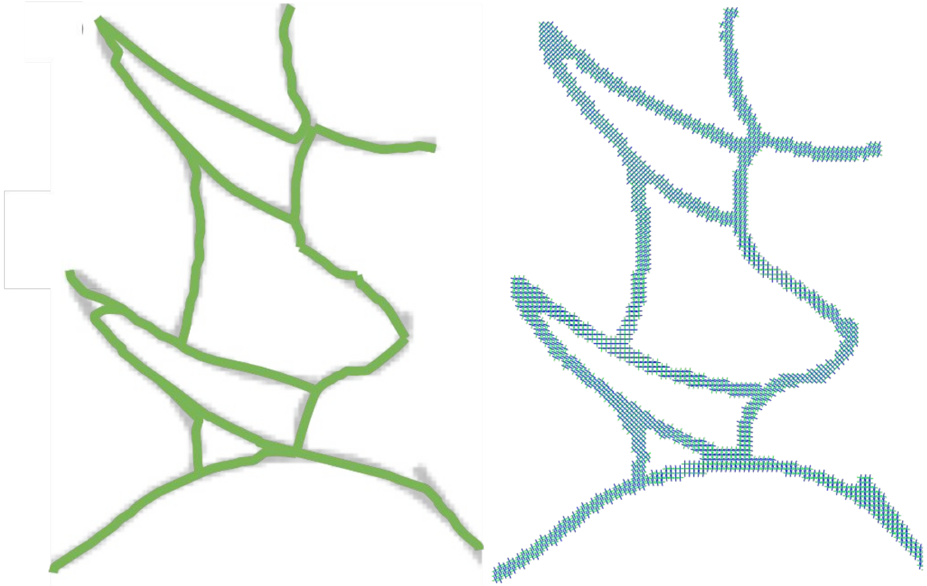
Thank you! Any questions?

Ablation Study (Optional) | Alignment

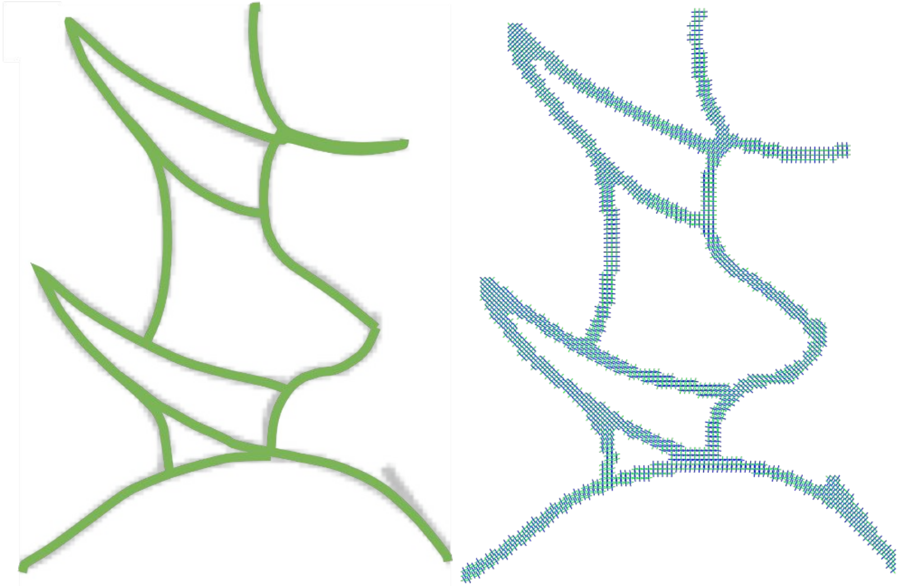
$$E_{\text{Alignment}} = \sum_{i:g_i \neq 0} \left[\nu E_\nu + \left[\cos((\theta_1)_i - \arg(g_i^2)) + 1\right] \times \left[\cos((\theta_1)_i + (\theta_2)_i - \arg(g_i^2)) + 1\right] \right]$$

Vanishes if the first line field is ⊥ to image gradient

Vanishes if the second line field is ⊥ to image gradient



Removing the Alignment Term



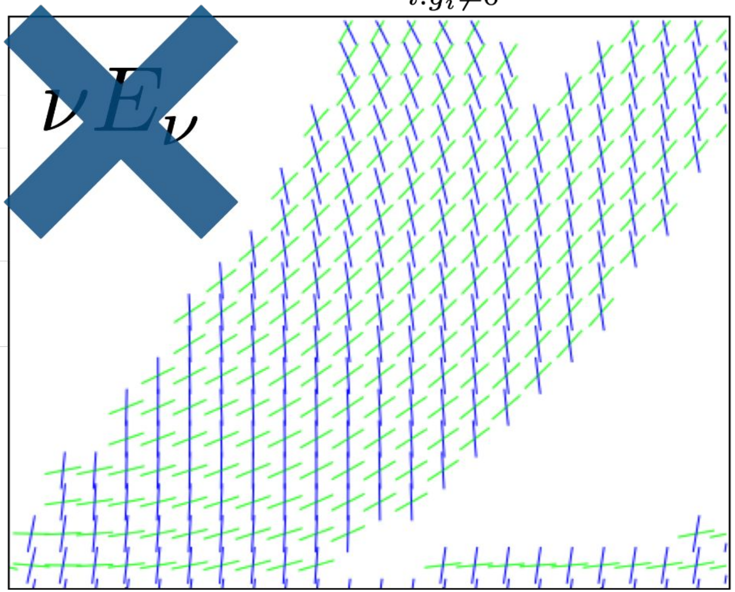
Objective Function with All Terms

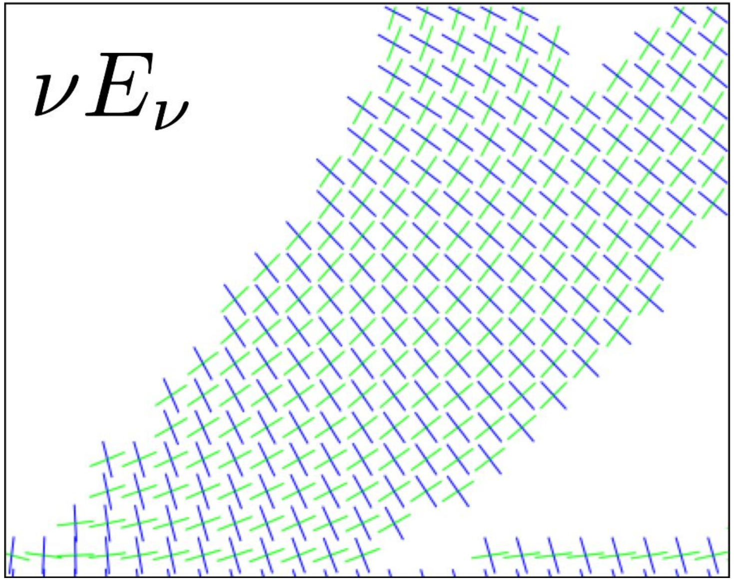
Ablation Study (Optional) | Anti-Align

Frame Field Objective: "Anti"-Alignment

$$E_{\text{Alignment}} = \sum_{i:g_i \neq 0} \left[\nu E_\nu + \left[\cos((\theta_1)_i - \arg(g_i^2)) + 1\right] \times \left[\cos((\theta_1)_i + (\theta_2)_i - \arg(g_i^2)) + 1\right]\right]$$

$$\sum_{i:g_i \neq 0} [-\cos((\theta_1)_i - g_i^2) + 1] \times [-\cos((\theta_1)_i + (\theta_2)_i - g_{\theta_i}^2) + 1]$$



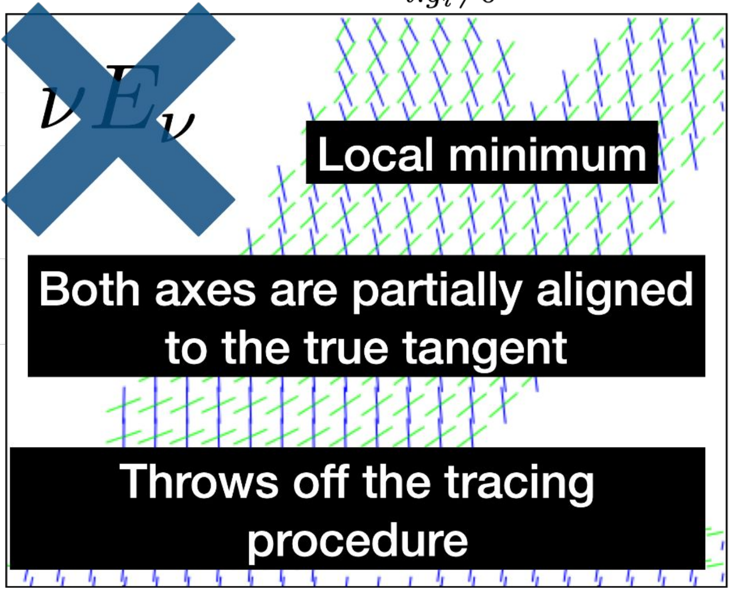


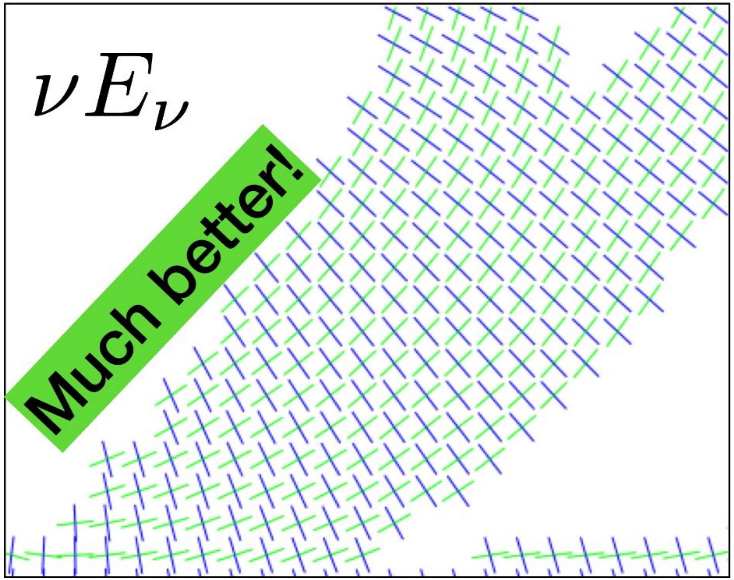
Ablation Study (Optional) | Anti-Align

Frame Field Objective: “Anti”-Alignment

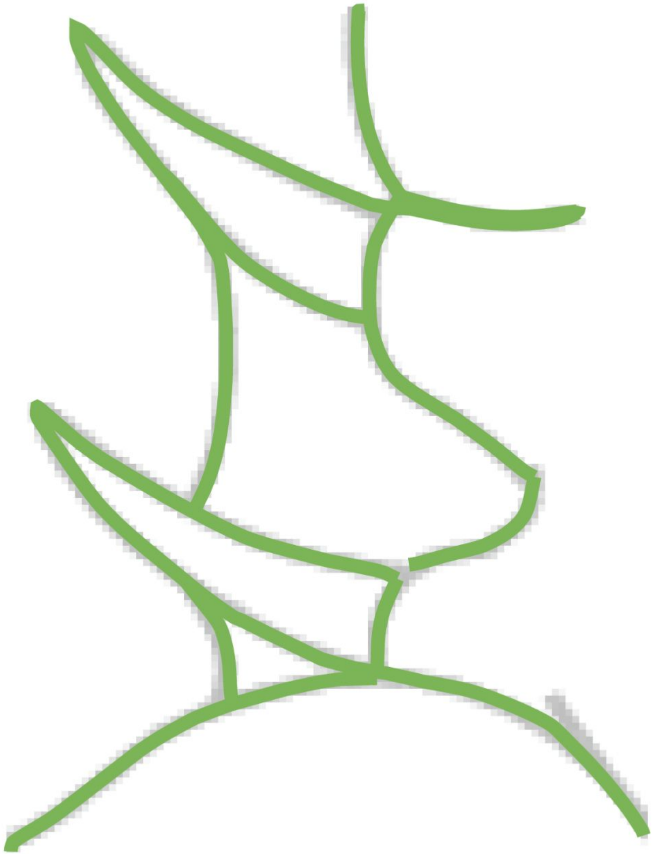
$$\mathrm{E}_{\text{Alignment}} = \sum_{i:g_i \neq 0} \Big[\nu \mathrm{E}_\nu + \big[\cos((\theta_1)_i - \arg(g_i^2)) + 1\big] \times \big[\cos((\theta_1)_i + (\theta_2)_i - \arg(g_i^2)) + 1\big] \Big]$$

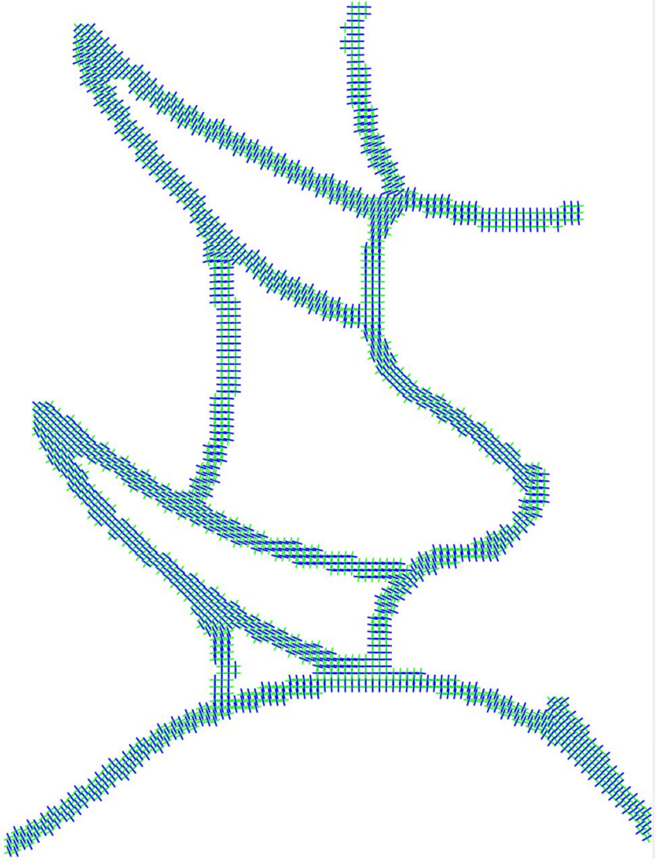
$$\sum_{i:g_i \neq 0} [-\cos((\theta_1)_i - g_i^2) + 1] \times [-\cos((\theta_1)_i + (\theta_2)_i - g_{\theta_i}^2) + 1]$$





Ablation Study (Optional) | Smoothness





SLIDE GRAVEYARD BELOW

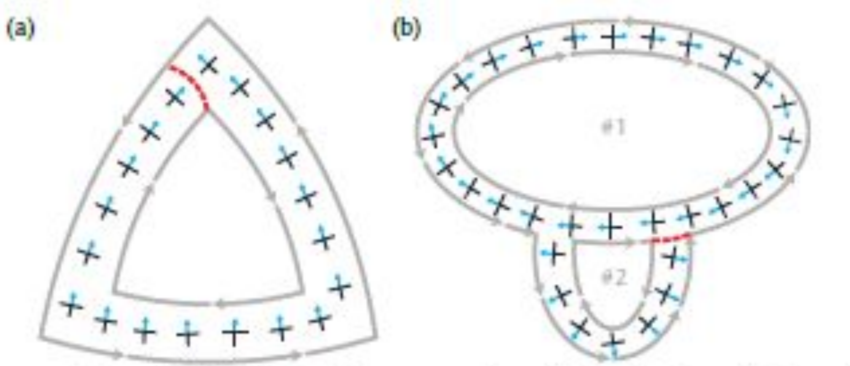
Introduction (perhaps w/ teaser image exemplar)

Related Work (short-ish, 2 slides)

- Some vectorization methods
  - Especially PolyVector & PolyFlow
- Fields stuff (more SGPish)
  - Trivial connections (maybe? Or just discuss later)
  - Lots of other stuff
  - Amir/Justin paper on matching of singularities via optimal transport (SV19)

Cross fields and line fields basics

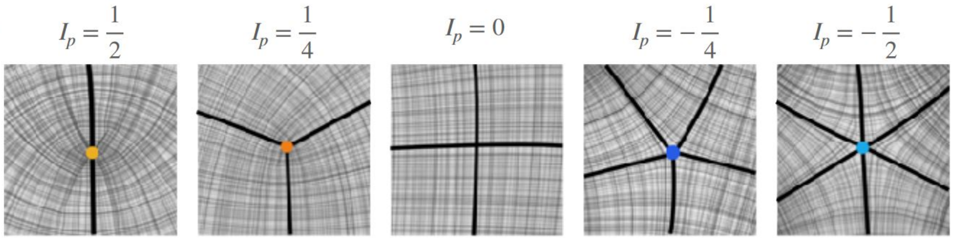
- Representative vectors
- Singularities and indices (and boundary indices, example below)
- Note the different convention
- Poincare Hopf formula

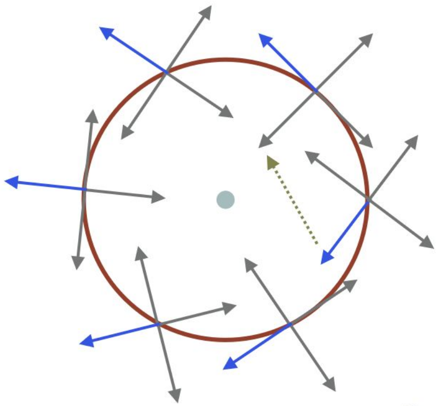


$\chi = 0$	Ext	Int	$\chi = 1$	Ext	Int_1	Int_2
Cpt rotation	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	Cpt rotation	3π	-2π	$-\pi$
Rep rotation	2π	-2π	Rep rotation	12π	-8π	-4π
Index	0	0	Index	-5	3	1

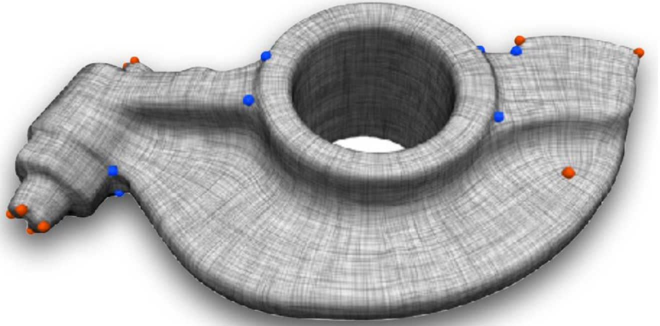
N-FIELD SINGULARITIES

- The same definition, but index can be in multiples of $\frac{1}{N}$.
- Interpretation: field "returns to itself" but with a different *matching* of vectors.





$$I_p = \frac{1}{4}, K_p = \frac{2\pi}{4}$$



Primal and dual mesh description

- Domain description
- D0 and d1 operators briefly, with intuition

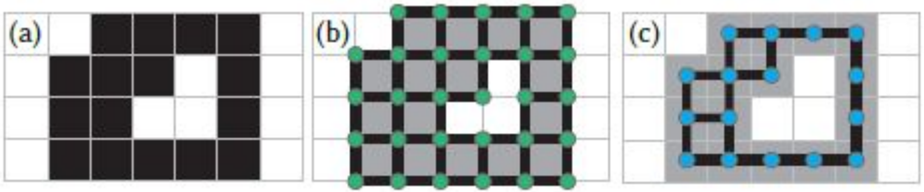
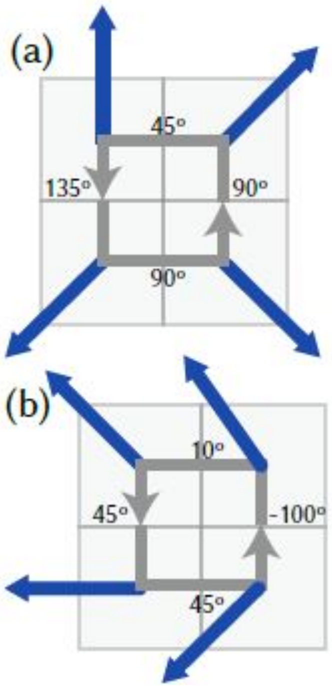


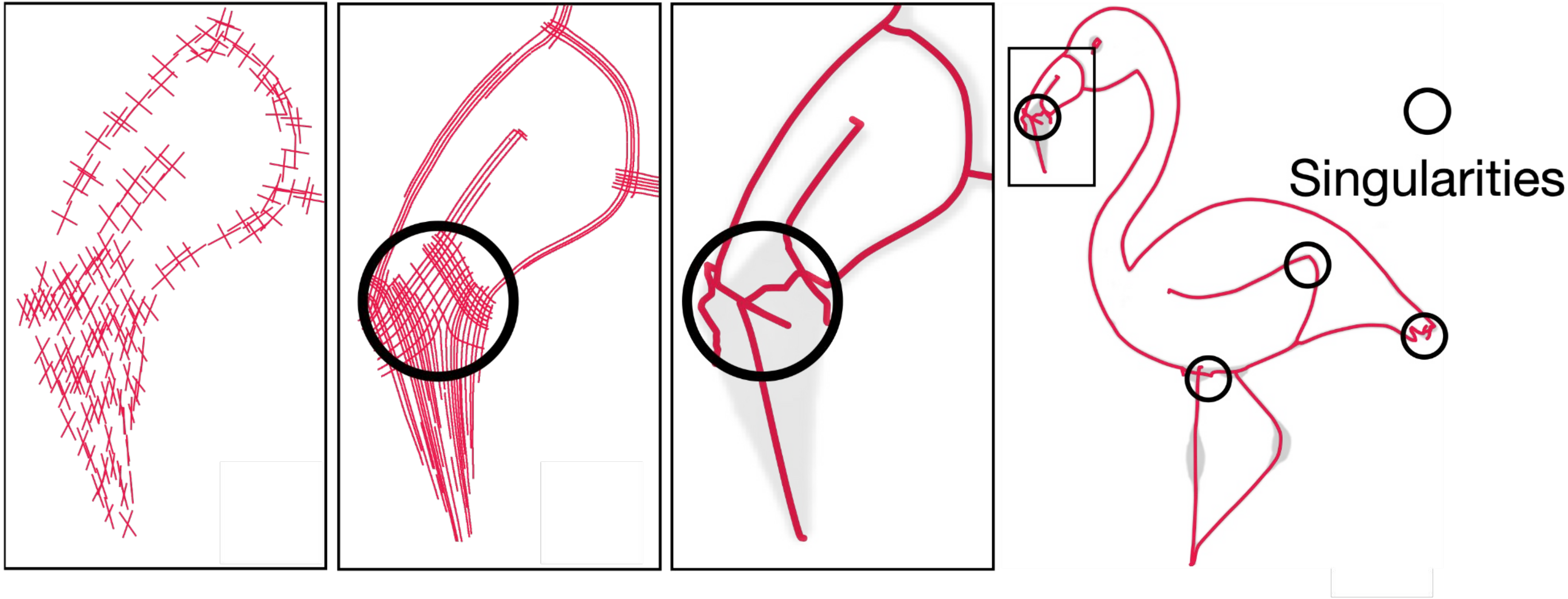
Figure 3: *Narrowband (a), primal mesh (b), and dual mesh (c).*

Putting it all together: trivial connections

- (for a field on a pixel grid)
- Discrete realization of index
- Write out the linear system (Eqn 10)
- Point out how Poincare Hopf shows up



Tracing Errors due to Singularities



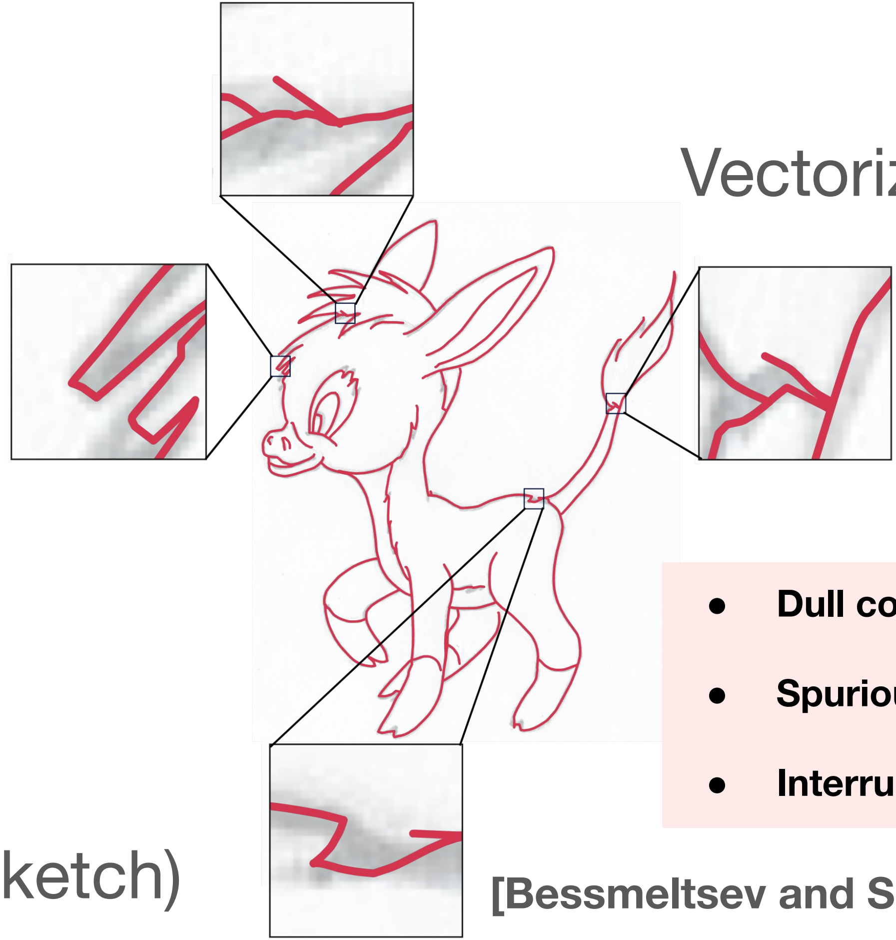
[Bessmeltsev and Solomon 2019]

Existing Methods Produce Topological Artifacts

Input

(Hand-Drawn Pencil Sketch)

Vectorization

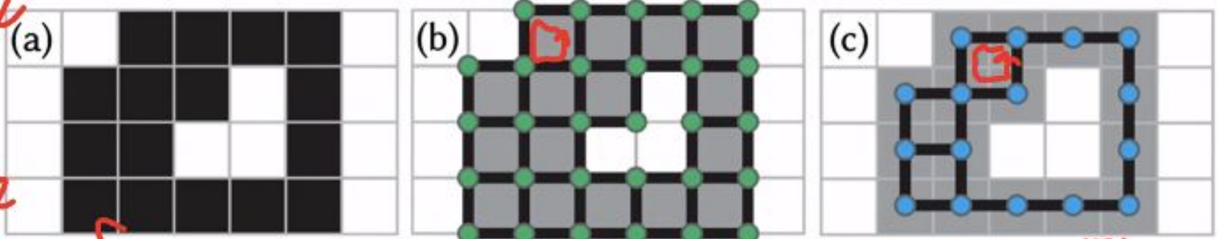


- Dull corners
- Spurious lines
- Interrupted tracing

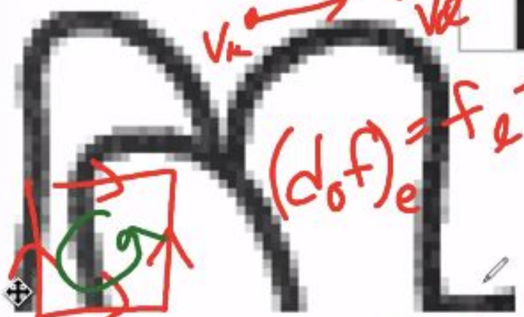
[Bessmeltsev and Solomon 2019]

Primal and Dual Mesh + Discrete Differential Operators

Primal and Dual Mesh + Discrete Differential Operators



(a) narrowband, (b) primal mesh, (c) dual mesh



Domain Processing

$$d_0(i,j) = \begin{cases} 1 & \text{if vertex j is the target of edge i} \\ -1 & \text{if vertex j is the source of edge i} \\ 0 & \text{otherwise} \end{cases}$$

Discretized gradient over each oriented edge

$$d_1(i,j) = \begin{cases} 1 & \text{if edge j is oriented counter-clockwise on face i} \\ -1 & \text{if edge j is oriented clockwise on face i} \\ 0 & \text{otherwise} \end{cases}$$

Discretized curl about primal mesh faces

Remove the formulas with pictures

ver

ns

SHREYA’S SLIDE GRAVEYARD BELOW

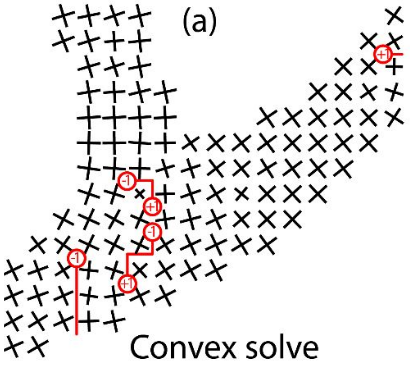
RIP…

(OR) Convex Initialization (if time permits)

- Resulting linear KKT system:

$$\left(\frac{1}{A_S}\Delta + \frac{w}{A_A}P\right) f = \frac{w}{A_A} P g^4$$

- System matrix is sparse, symmetric, and positive semidefinite
- Solve the system via preconditioned conjugate gradients



Aligns well in most regions

But many internal singularities, often arising at junctions and curve endpoints

Field Combing

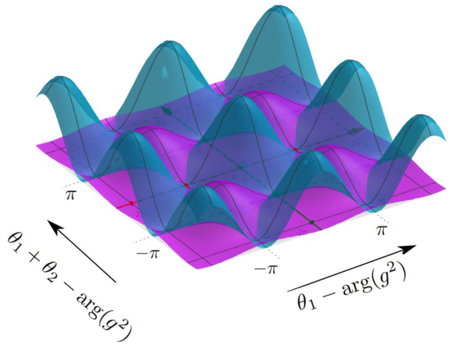
- We consider a cost matrix with d− and d+ that denoting distances on the +1 and -1 directed graphs
- The resulting transport is going to be a binary matching of negative singularities and positive singularities, with some being pushed to boundaries
- The optimization problem has a binary solution $T_{ij} \in \{0,1\}^{(|\mathcal{S}^-|+1)\times(|\mathcal{S}^+|+1)}$

$$
\begin{aligned}
&\min_T \langle C, T\rangle \\
&s.t. \sum_j T_{ij} = 1, \text{ for } 1 \le i \le |\mathcal{S}^-| \\
&\quad \sum_i T_{ij} = 1, \text{ for } 1 \le j \le |\mathcal{S}^+| \\
&\quad T_{ij} \ge 0, \ \forall i, j.
\end{aligned}
$$

SOLOMON J., VAXMAN A.: Optimal transport-based polar interpolation of directional fields. ACM Trans. Graph. 38, 4 (2019)

Alignment (if time permits)

$$\mathrm{E}_\nu = \sum_{i:g_i \neq 0} [-\cos((\theta_1)_i - g_i^2) + 1][-\cos((\theta_1)_i + (\theta_2)_i - g_{\theta_i}^2) + 1]$$



- regularizes alignment term
- ensures that local minima are nondegenerate
- pushes the line fields away from being aligned with each other

$$\mathrm{E}_{\mathrm{Alignment}} = \sum_{i:g_i \neq 0} \left[\nu \mathrm{E}_\nu + \left[\cos\big((\theta_1)_i - \arg(g_i^2)\big) + 1 \right] \times \left[\cos\big((\theta_1)_i + (\theta_2)_i - \arg(g_i^2)\big) + 1 \right] \right]$$

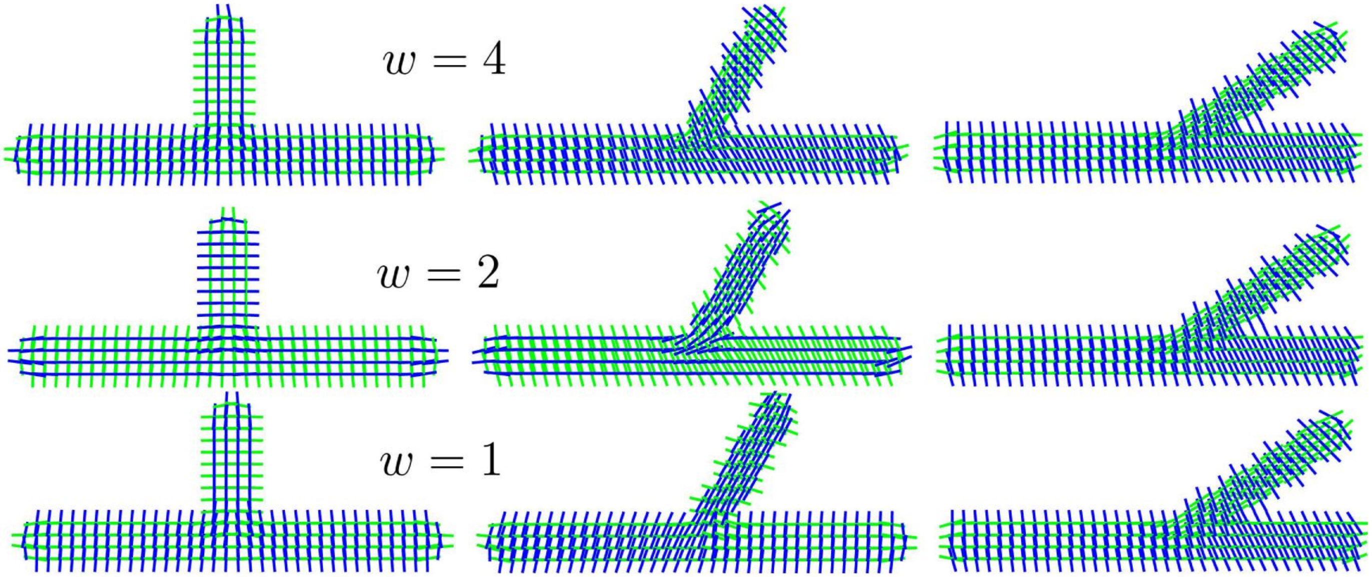
Tangent selection

- Per-pixel selection of one of the axes needs to be the curve tangent
- Pipeline begins by tracing the frame field along the tangent directions
- We perform a simple smooth selection of axis that is most aligned with the tangent

Parameters we used

Effect of parameters (if time permits)

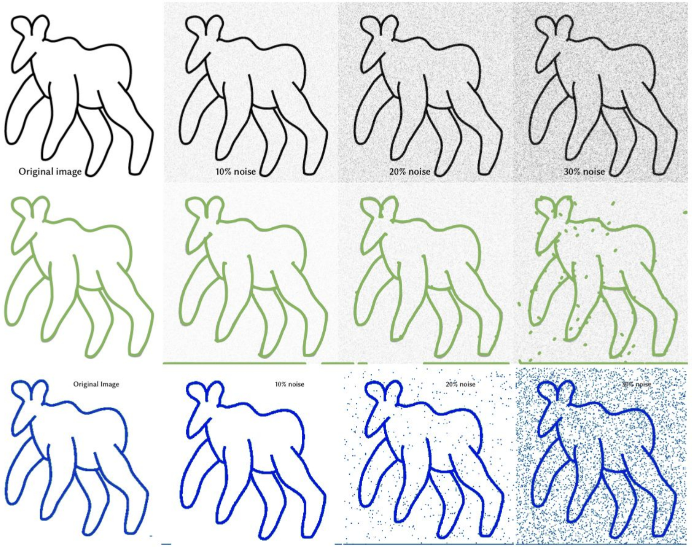
w controls the transition threshold between a T-junction and a Y-junction



*w = 0.125 in our experiments

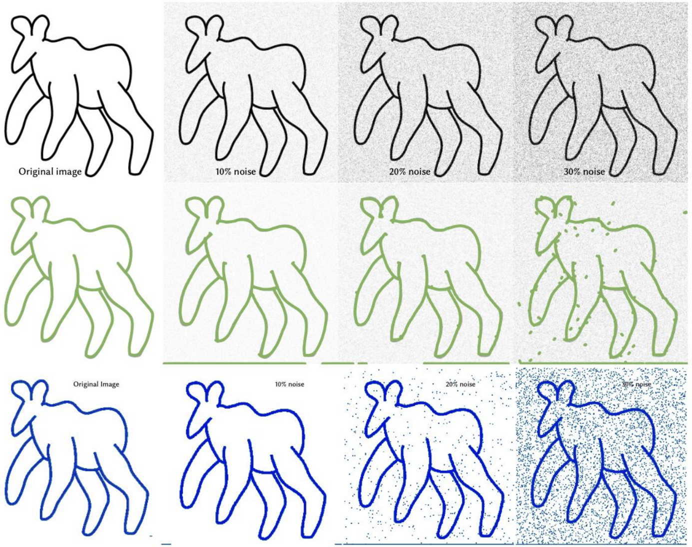
Robustness

Determines correct topology despite a noisy narrowband



Robustness

Determines correct topology despite a noisy narrowband



Robust to changes in resolution