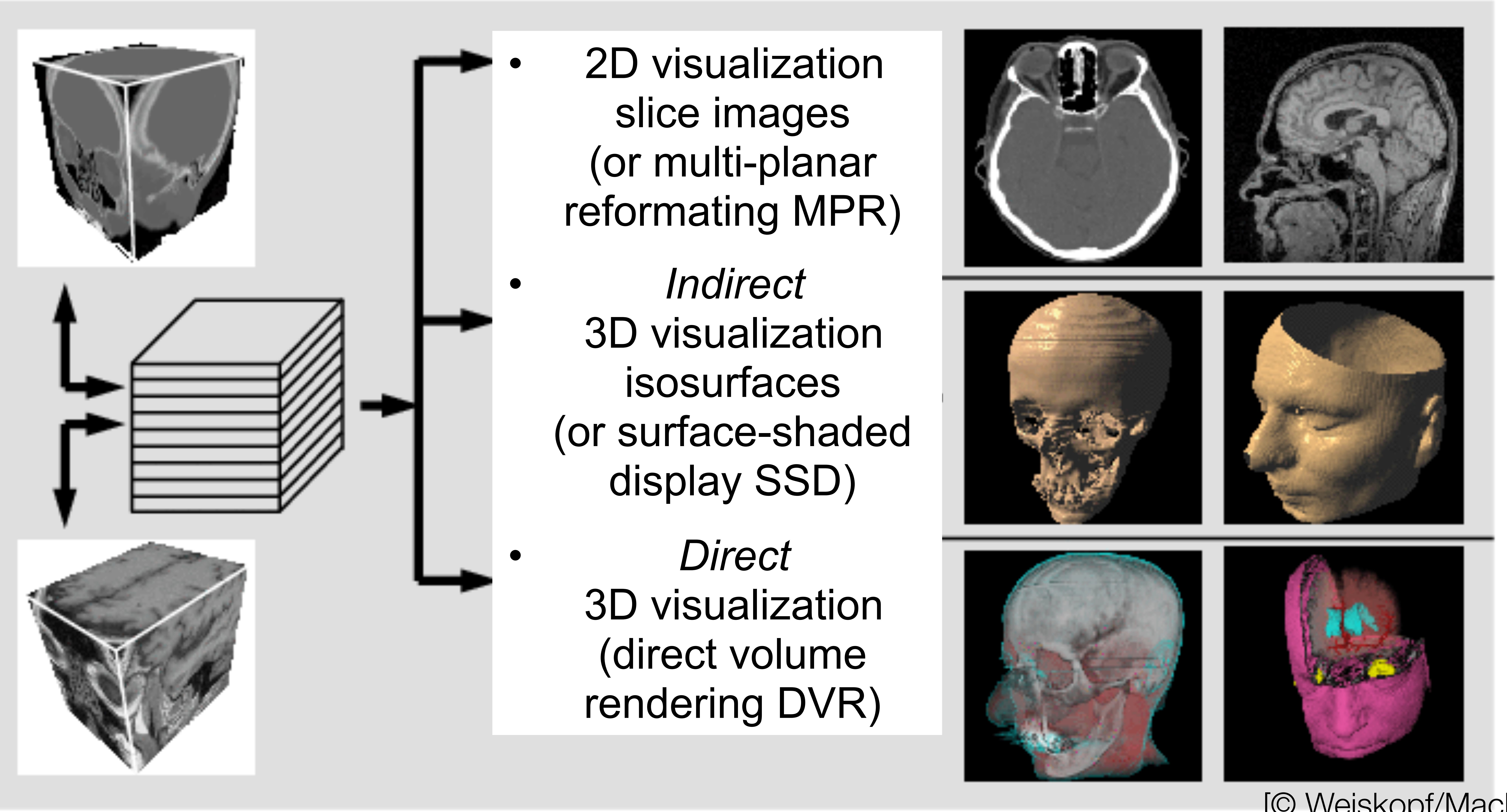
Data Visualization (CSCI 490/680)

Volume Rendering & Vector Fields

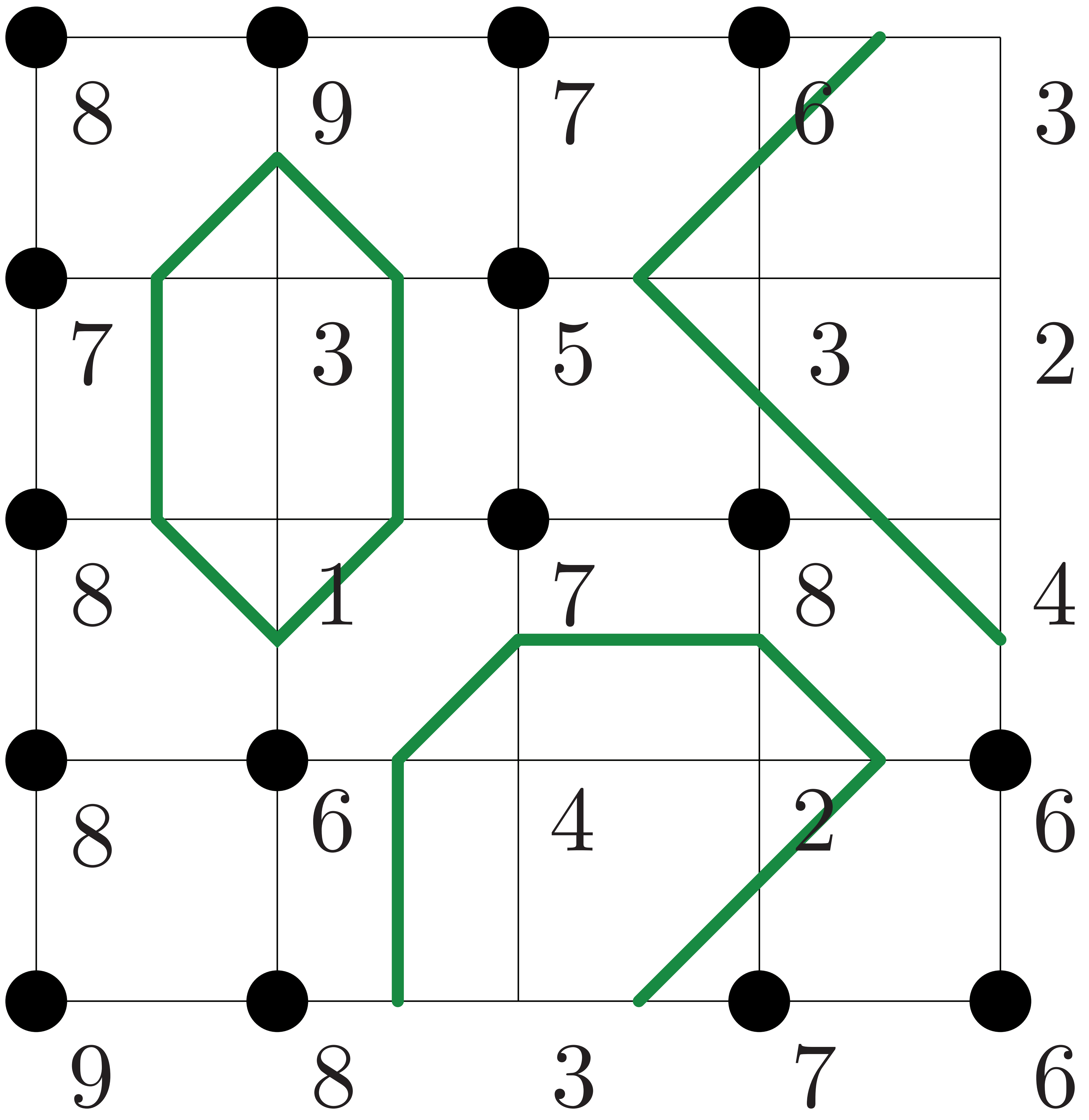
Dr. David Koop

Visualizing Volume (3D) Data



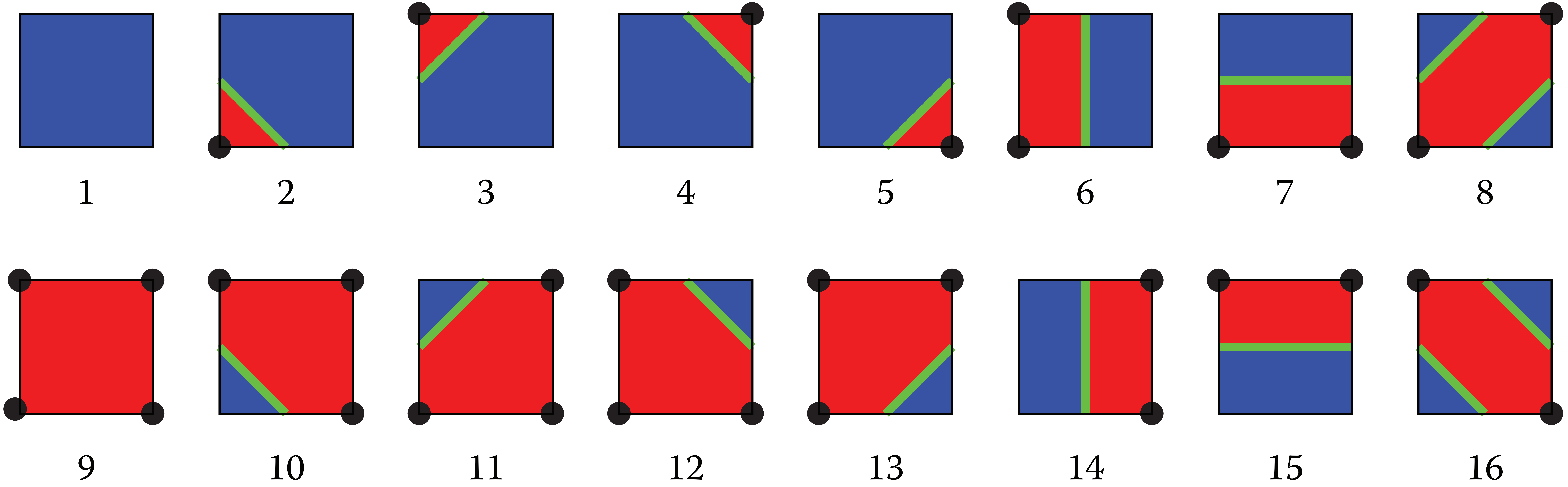
[© Weiskopf/Machiraju/Möller]

Generating Isolines (Isovalue = 5)



[R. Wenger, 2013]

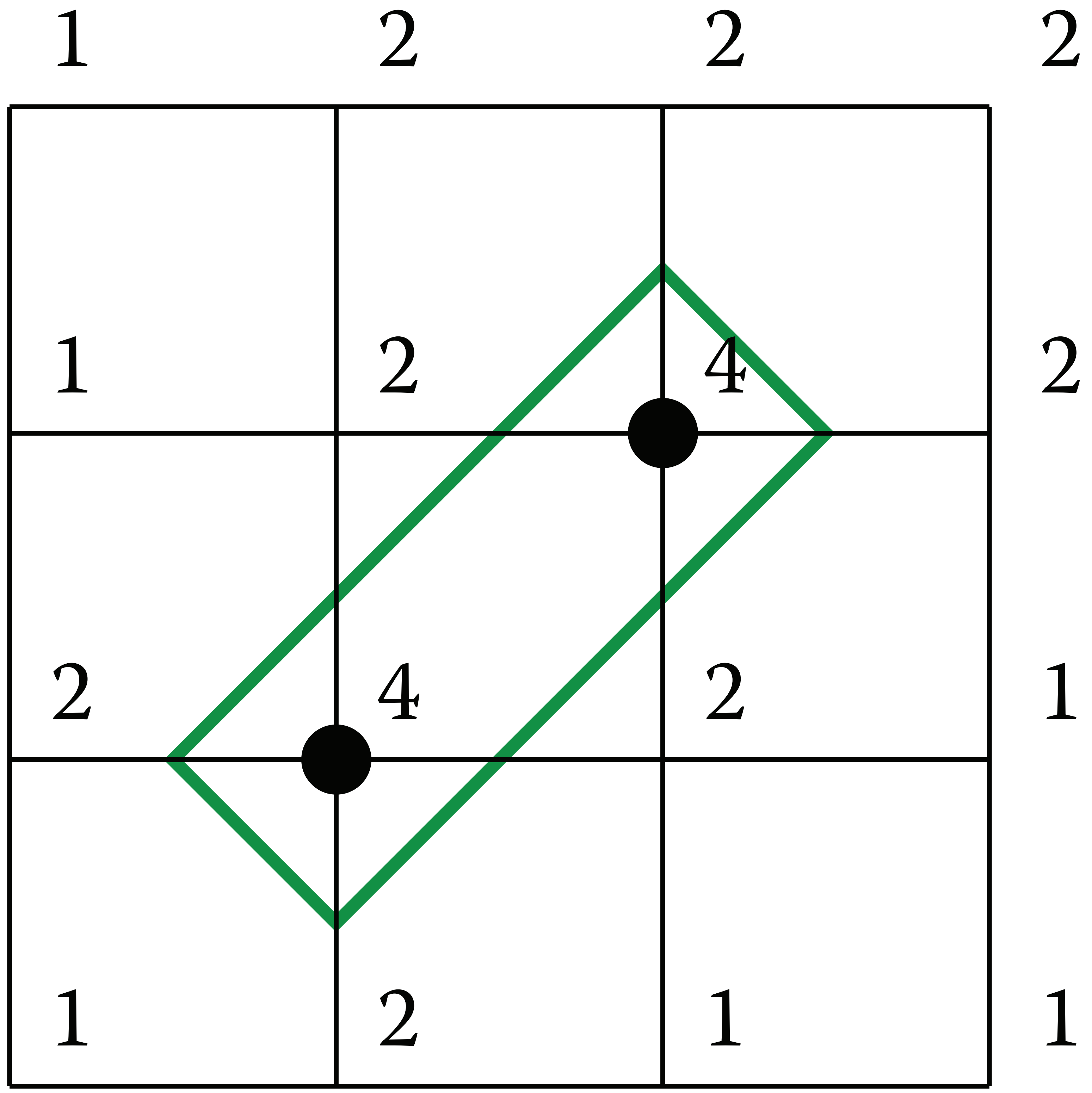
Marching Squares

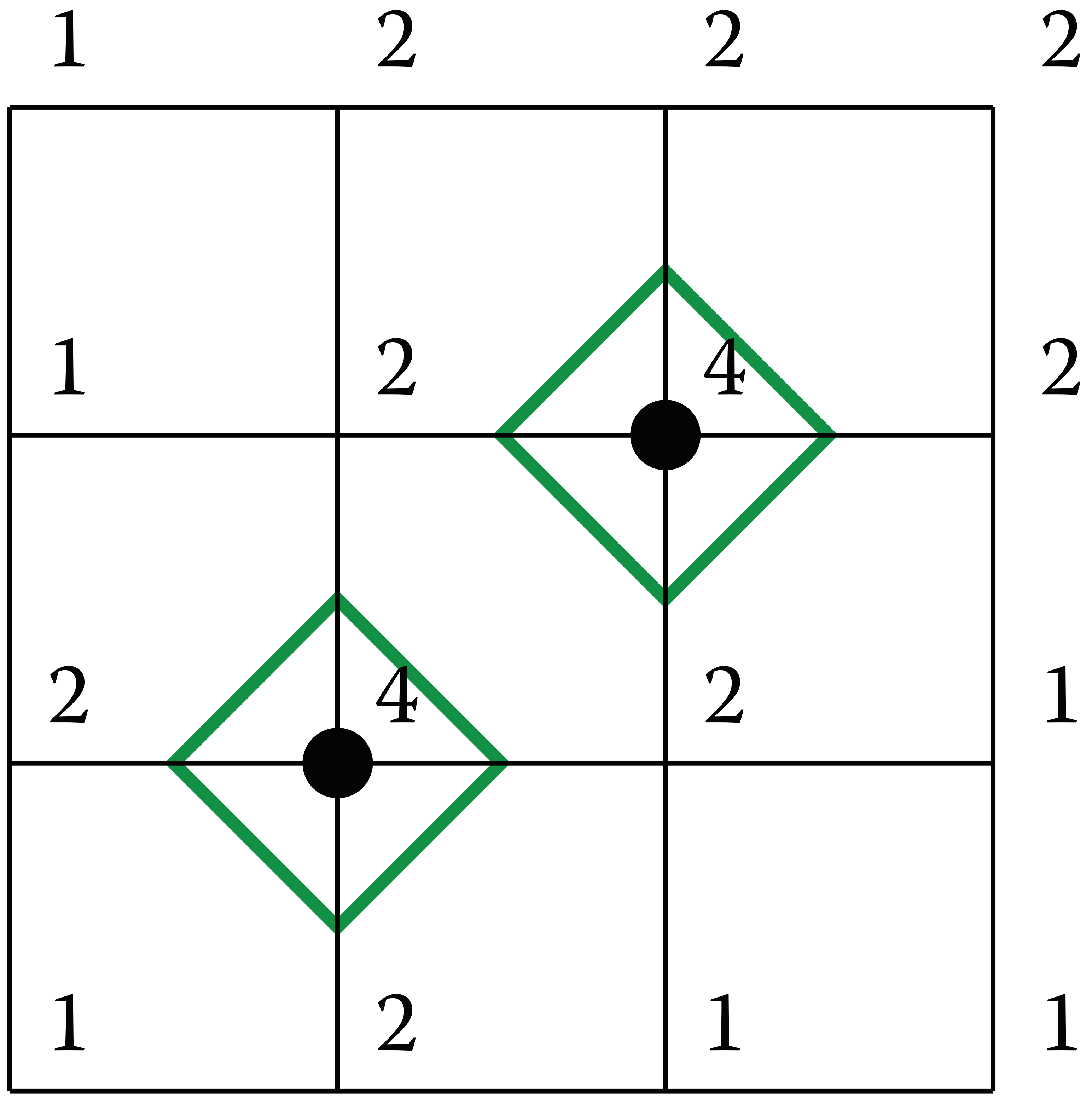


[R. Wenger, 2013]

Ambiguous Configurations

- Either works for marching squares, this isn't the case for 3D

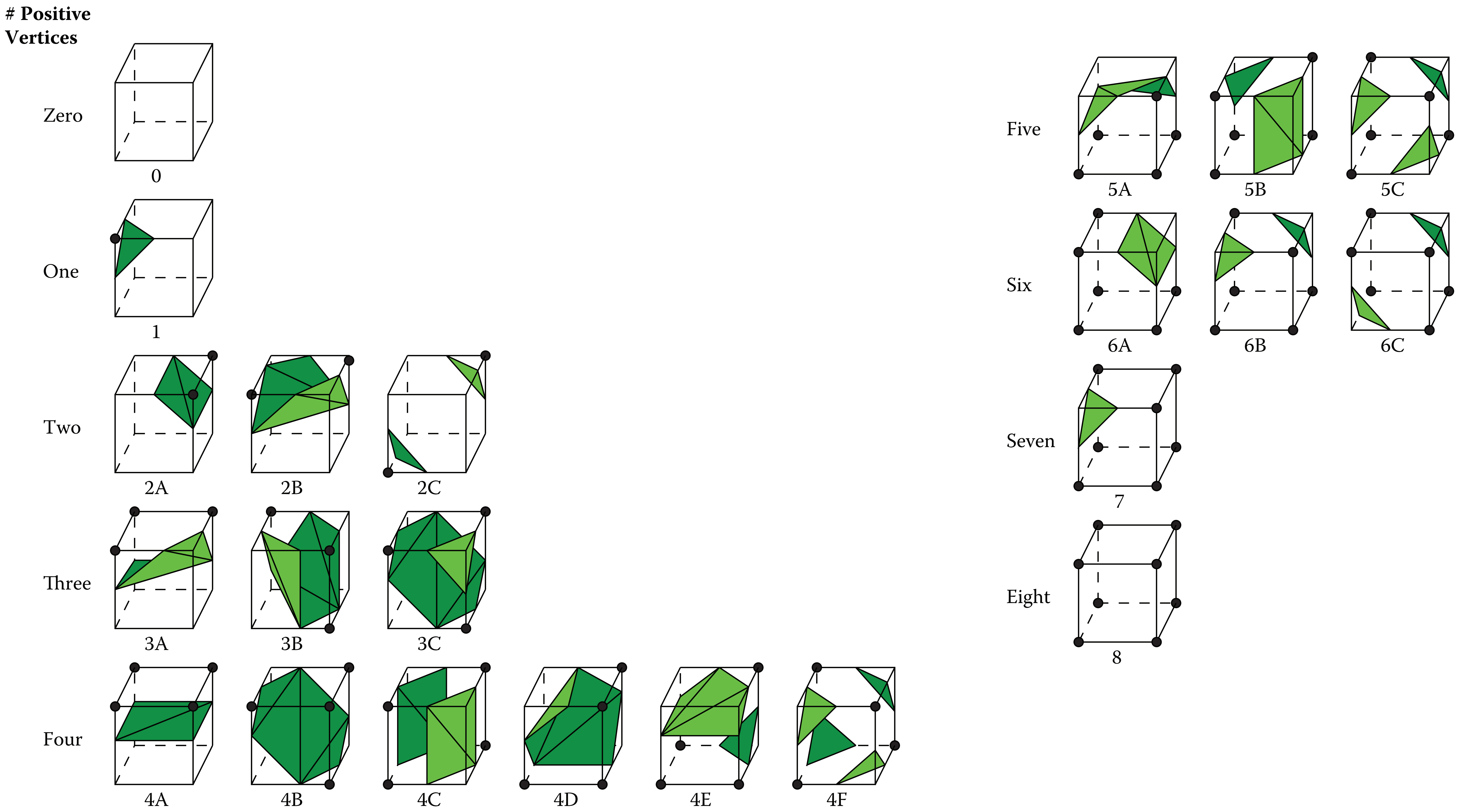




[R. Wenger, 2013]

3D: Marching Cubes

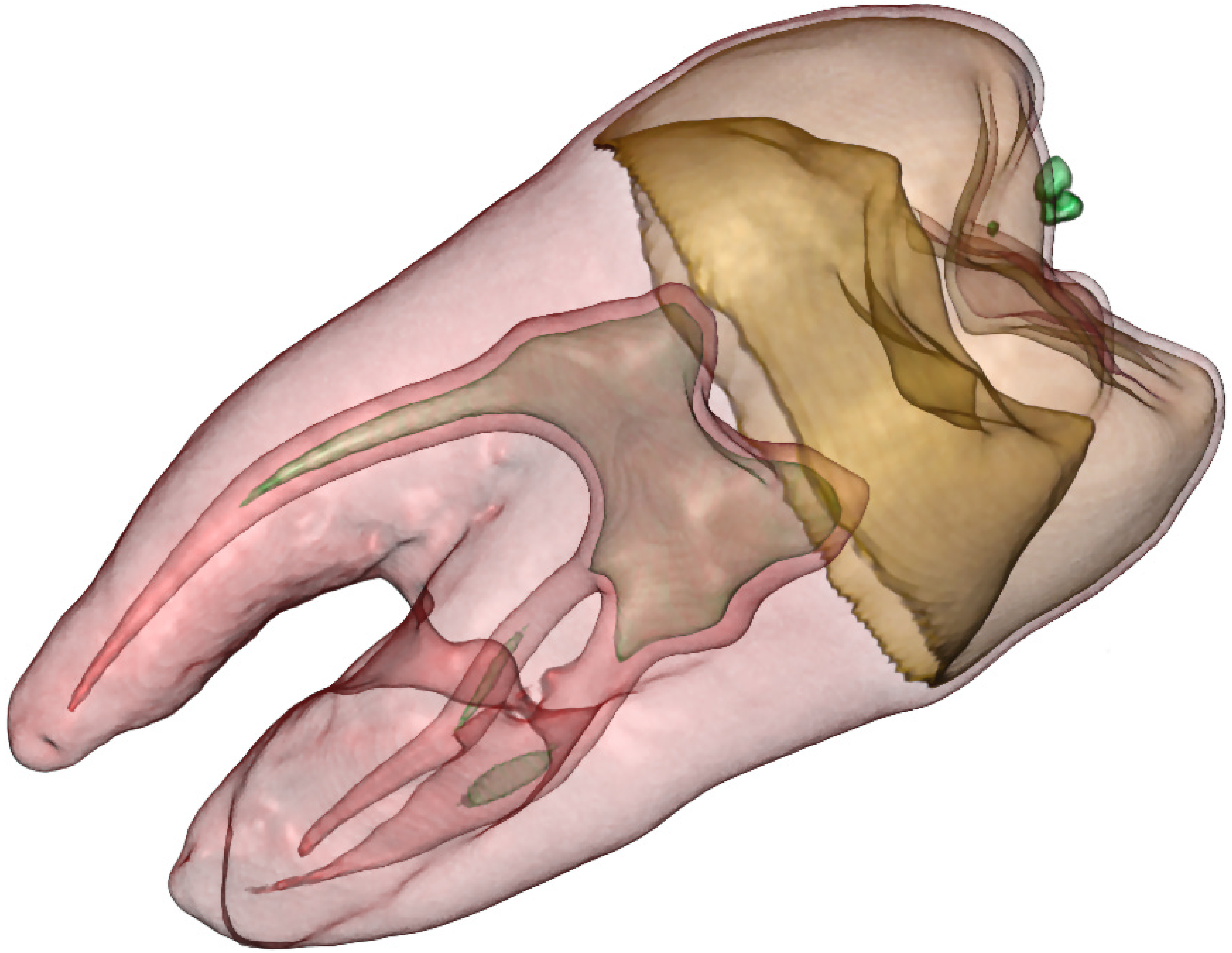
- Same idea, more cases [Lorensen and Cline, 1987]



[R. Wenger, 2013]

Multiple Isosurfaces

- Topographical maps have multiple isolines to show elevation trends
- Problem in 3D? **Occlusion**
- Solution? Transparent surfaces
- Issues:
  - Think about color in order to make each surface visible
  - Compositing: how do colors "add up" with multiple surfaces
  - How to determine good isovalues?



[J. Kniss, 2002]

Volume Rendering vs. Isosurfacing

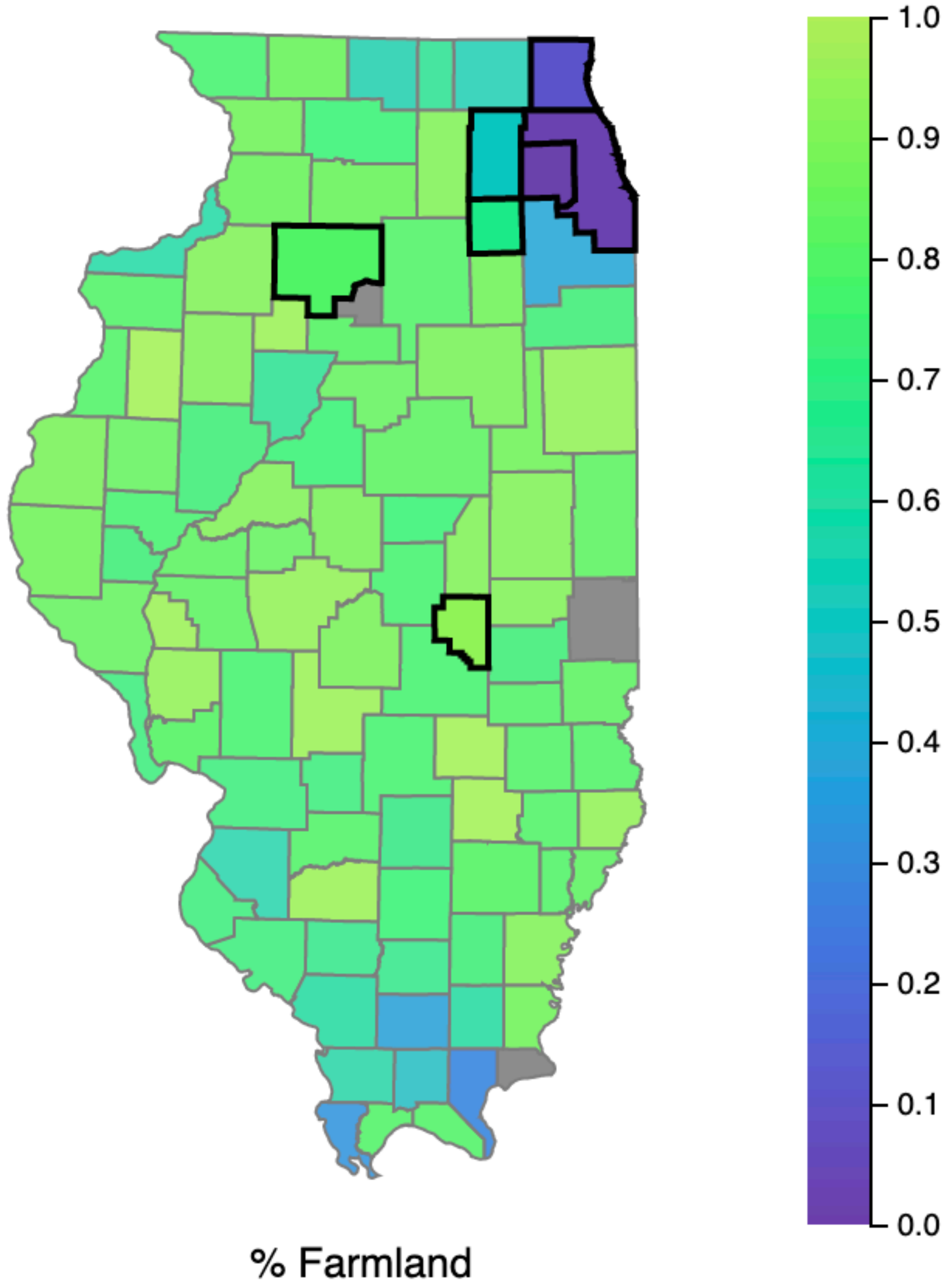
(a) Direct volume rendered

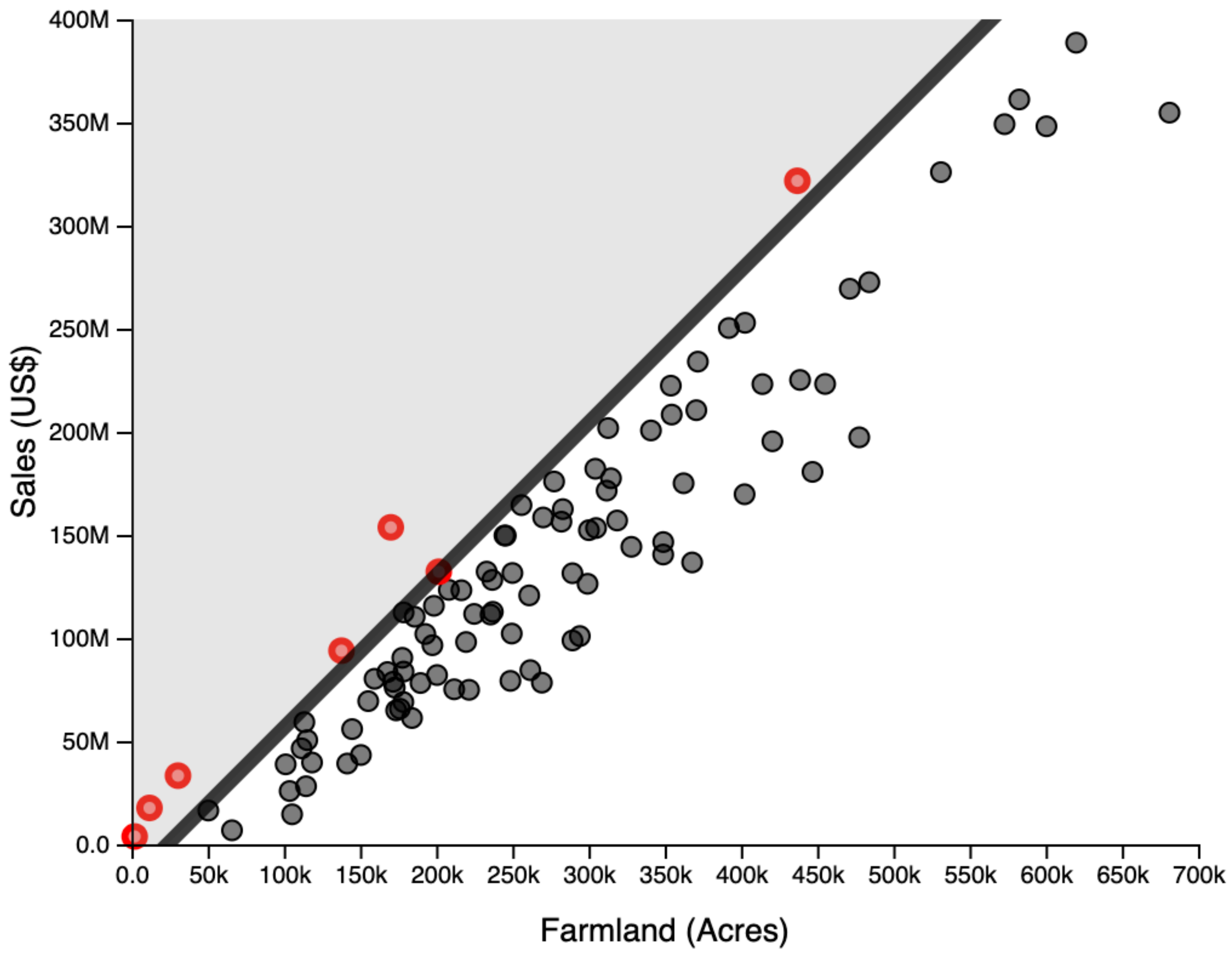
(b) Isosurface rendered

[Kindlmann, 1998]

Assignment 5

- Multiple Views and Interaction using Linked Highlighting
- Due Tomorrow

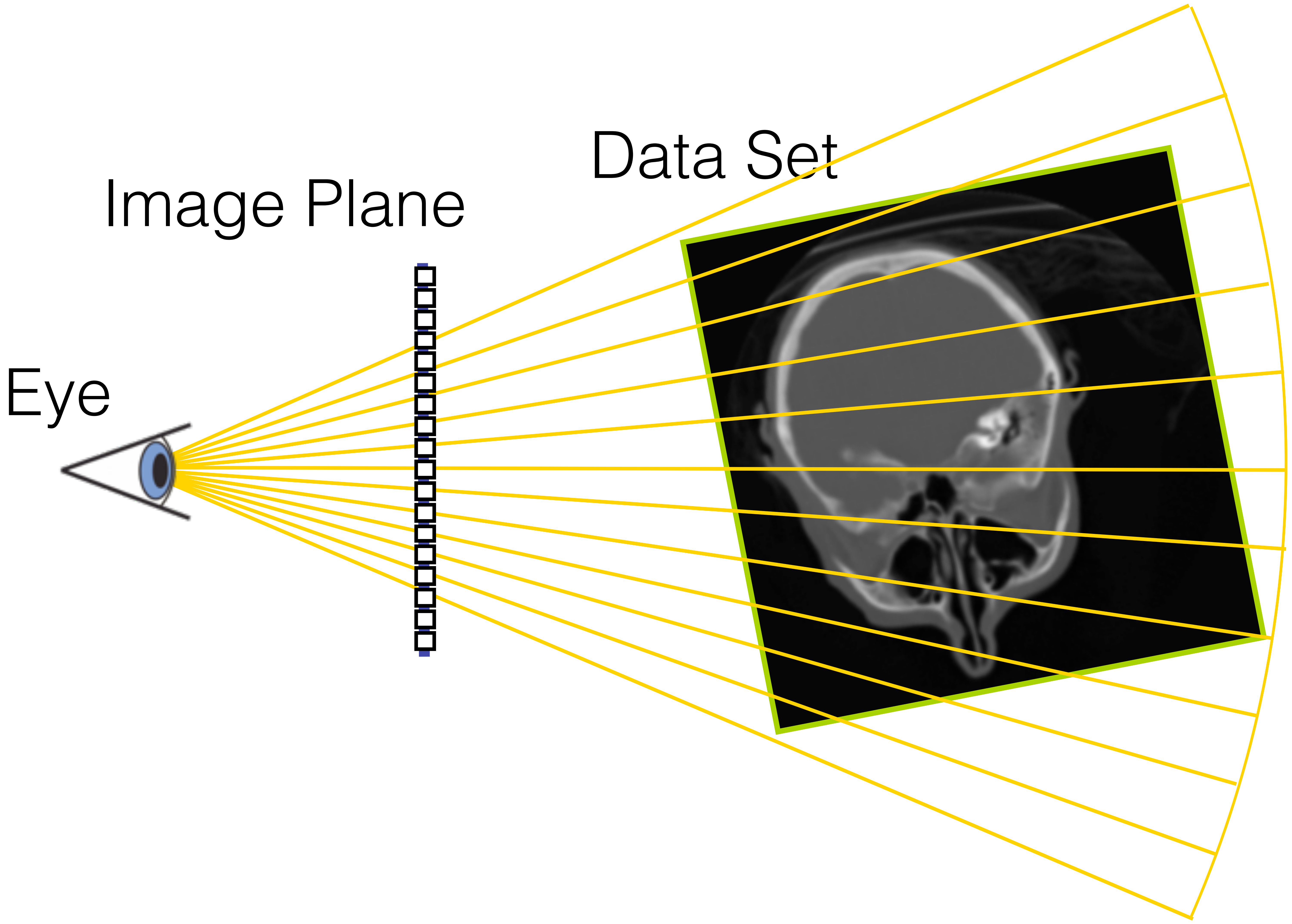




Project

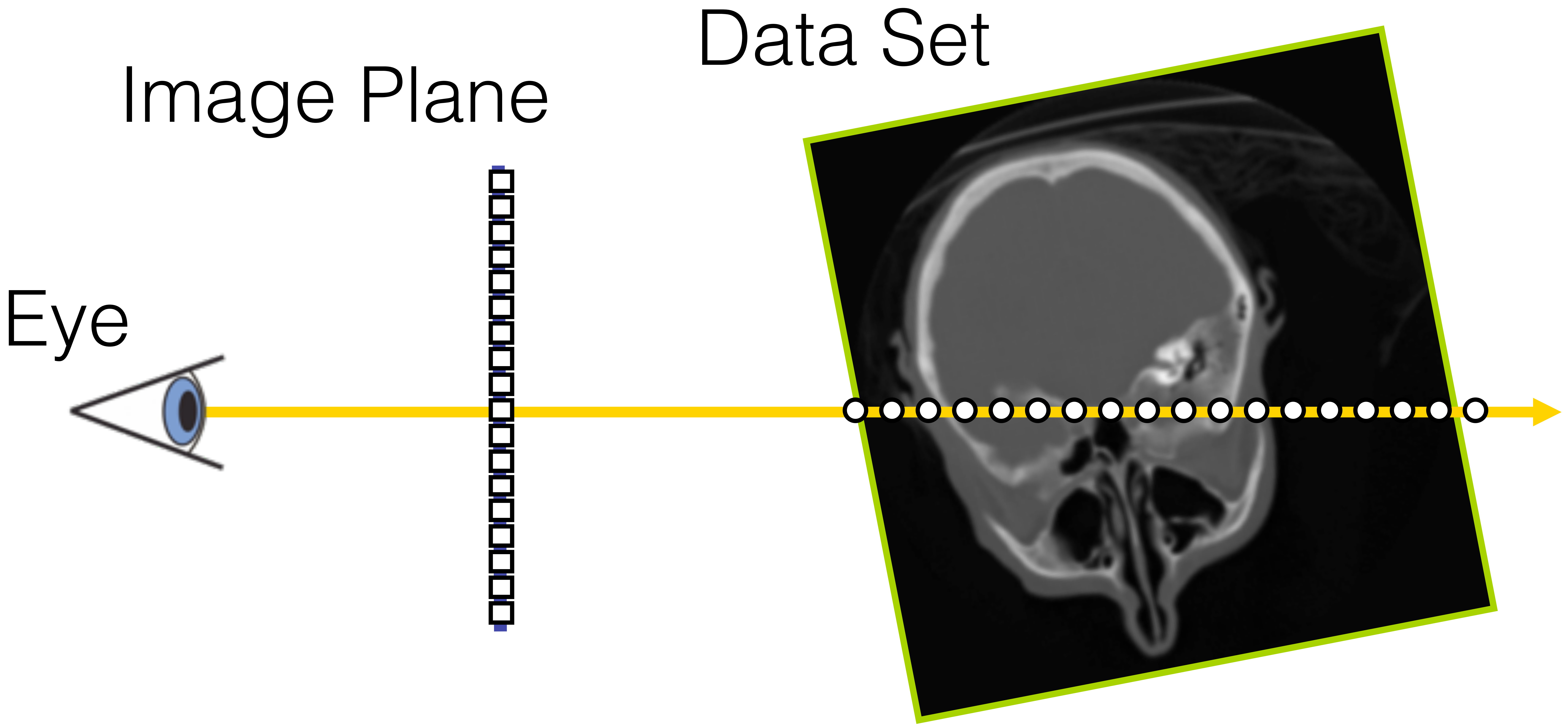
- Incorporate feedback from Blackboard comments
- Continue to be creative but also remember expressiveness and effectiveness
- Looking forward to presentations on Dec. 5
- Have until Dec. 6 to turn in final code and report

Volume Ray Casting



[Levine]

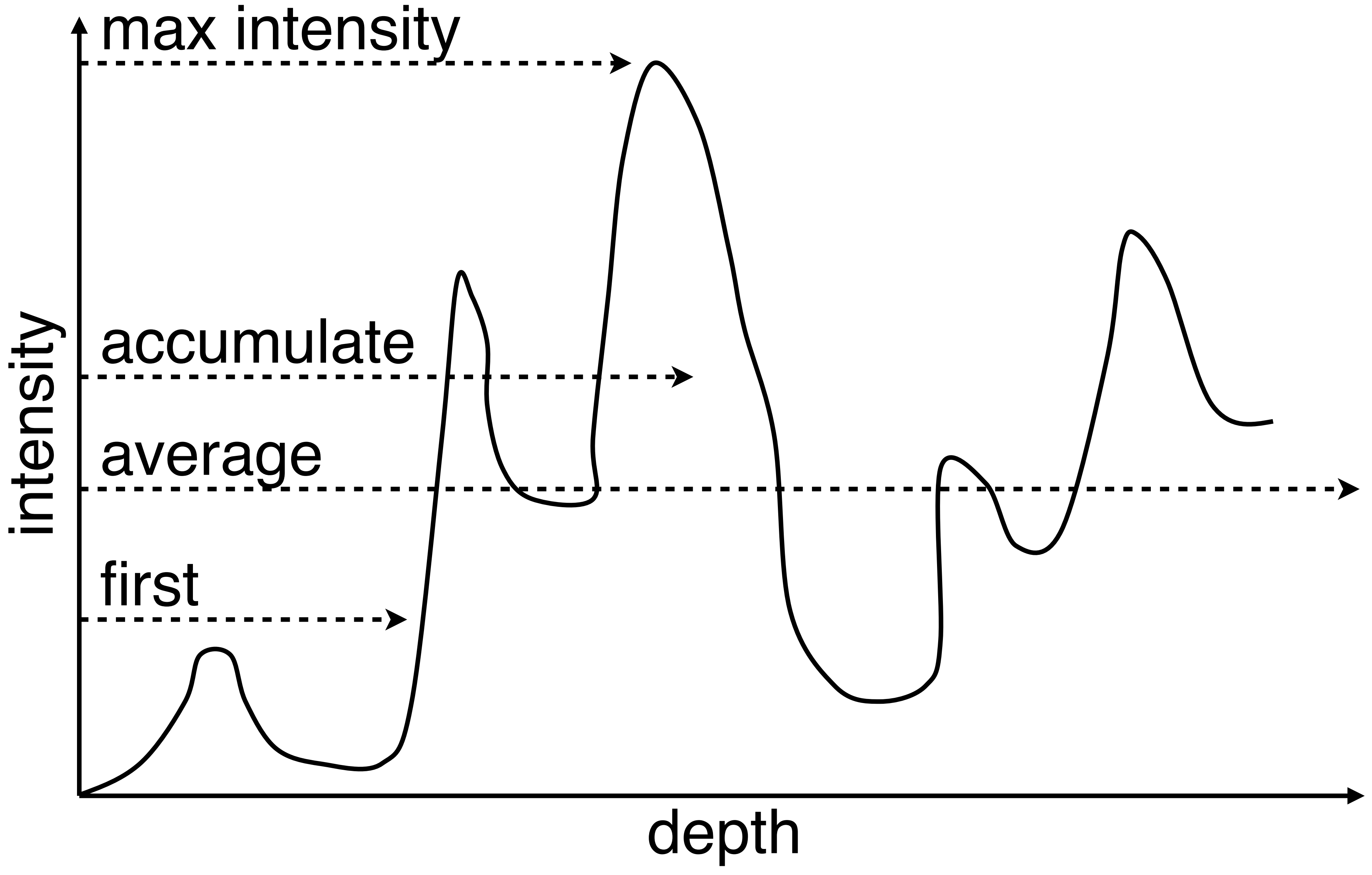
Volume Ray Casting



[Levine]

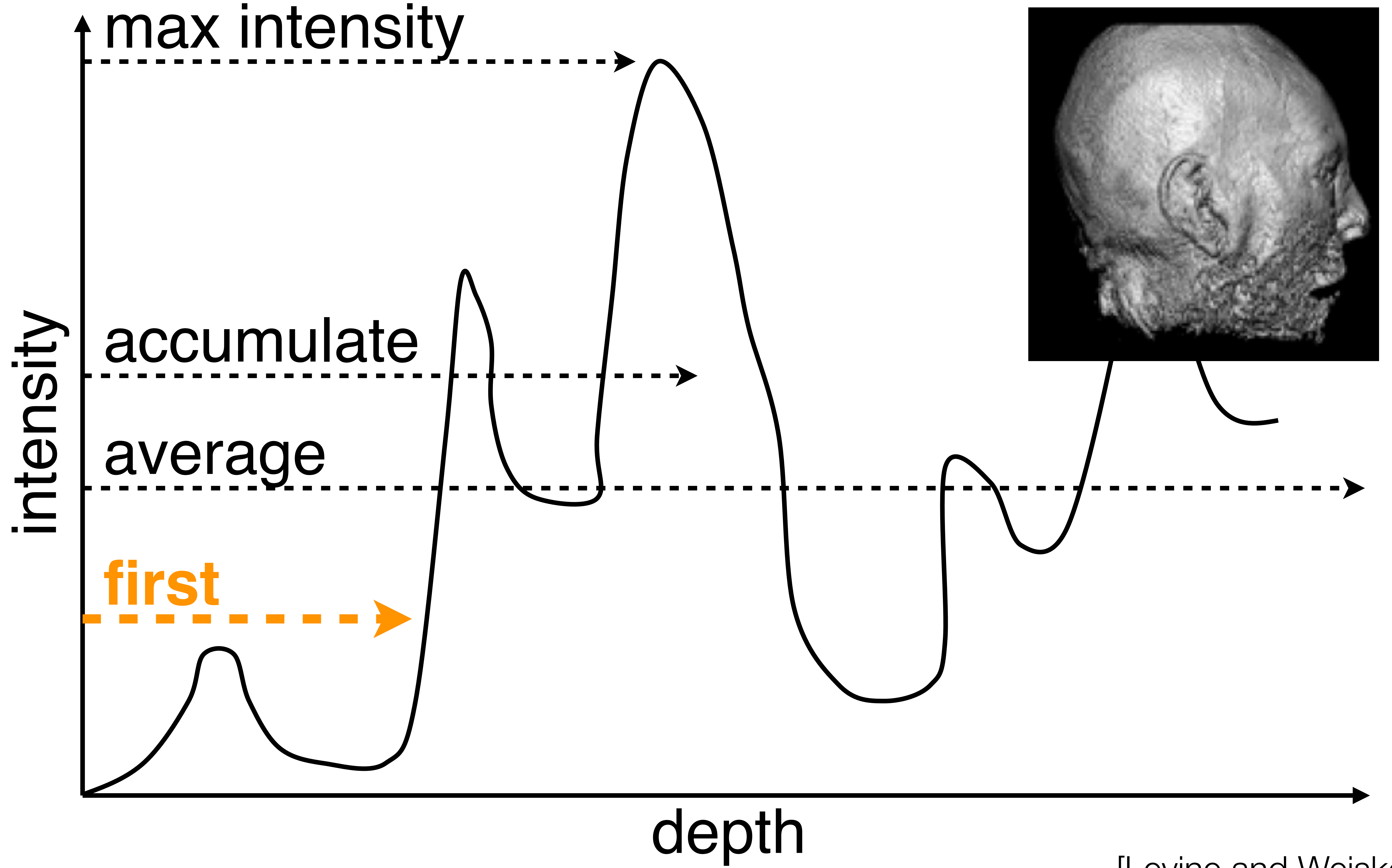
Compositing

- Need **one pixel** from all values along the ray
- Q: How do we "add up" all of those values along the ray?
- A: Compositing!
- Different types of compositing
  - First: like isosurfacing, first intersection at a certain intensity
  - Max intensity: choose highest val
  - Average: mean intensity (density, like x-rays)
  - Accumulate: each voxel has some contribution



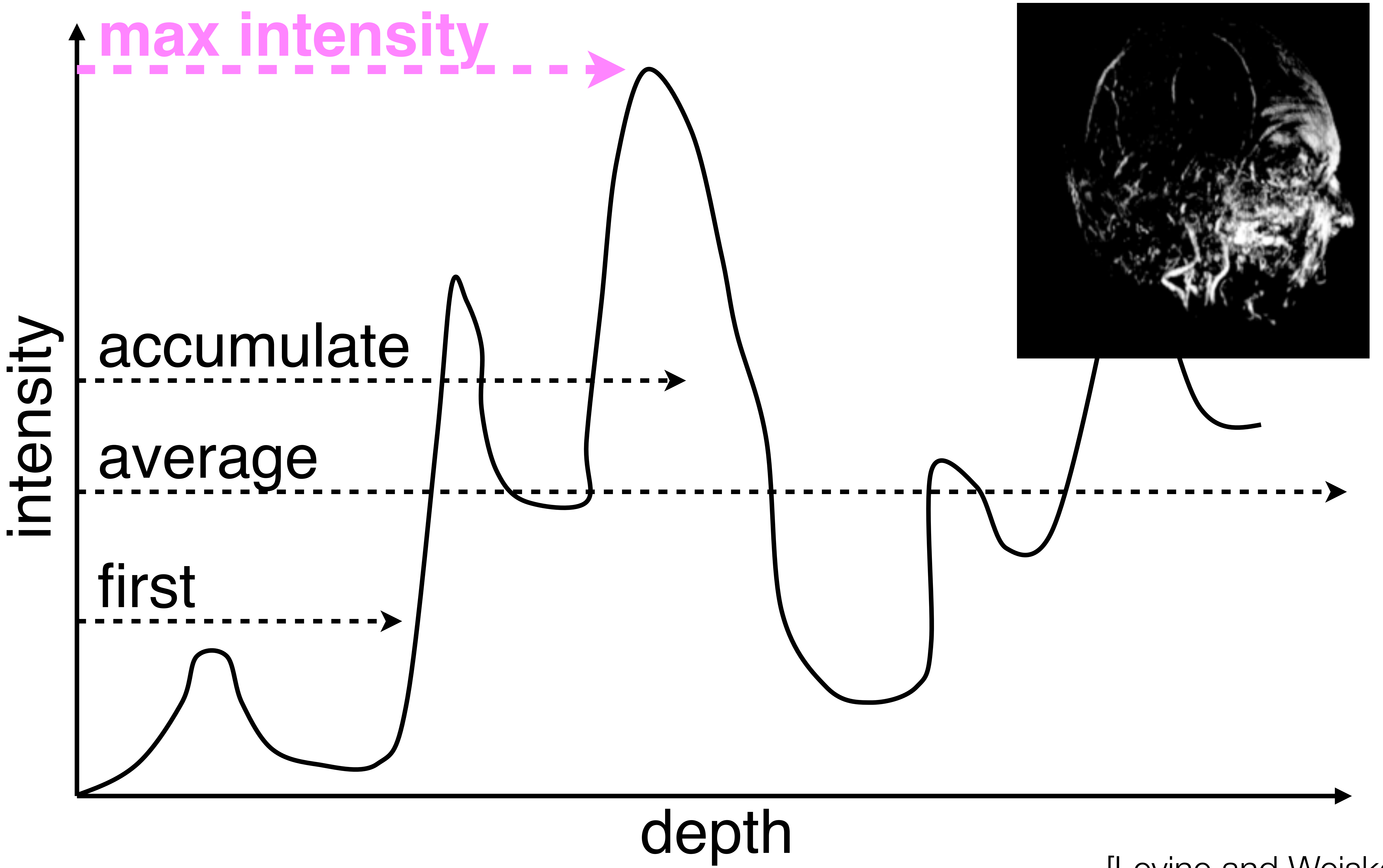
[Levine and Weiskopf/Machiraju/Möller]

Types of Compositing



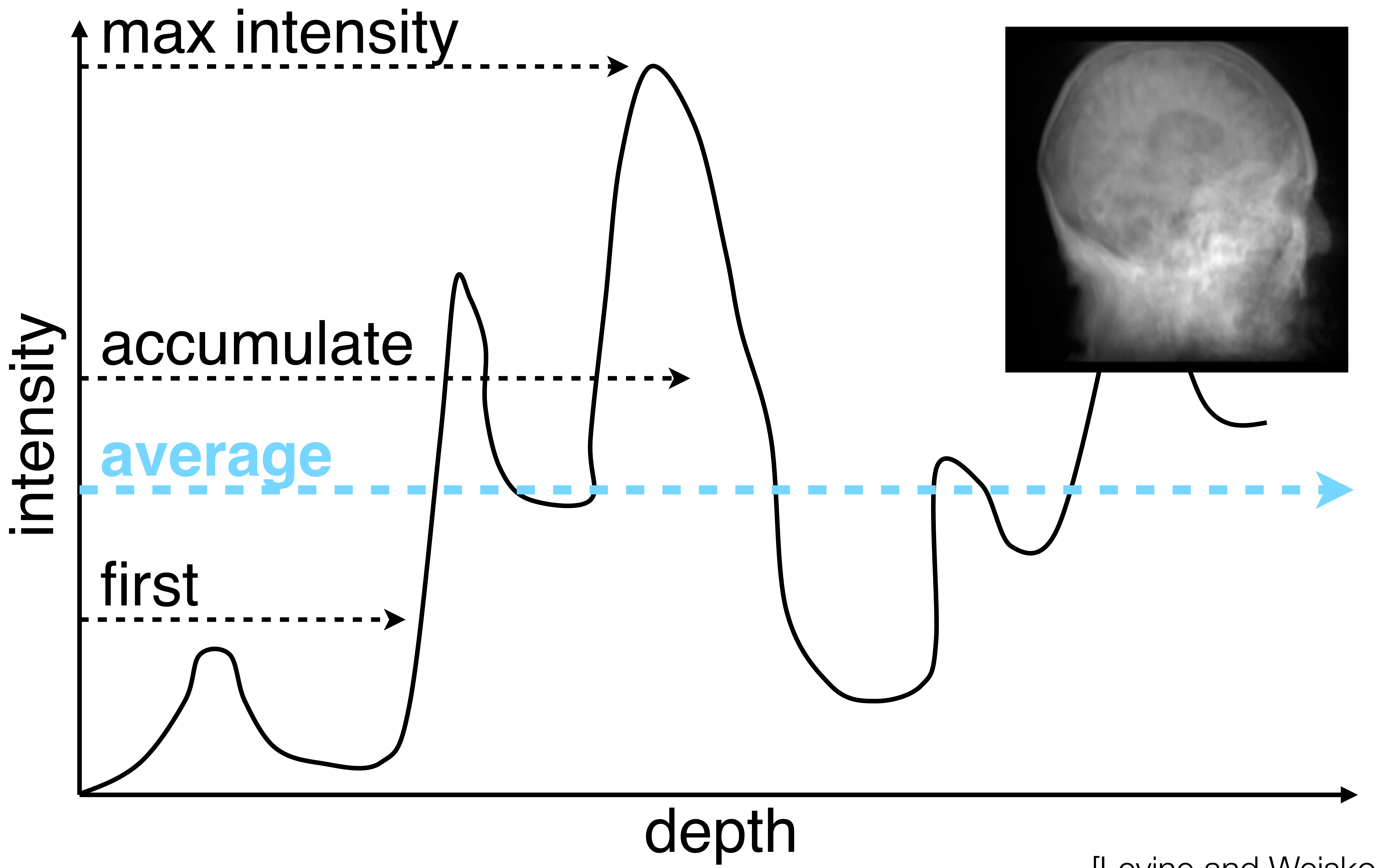
[Levine and Weiskopf/Machiraju/Möller]

Types of Compositing



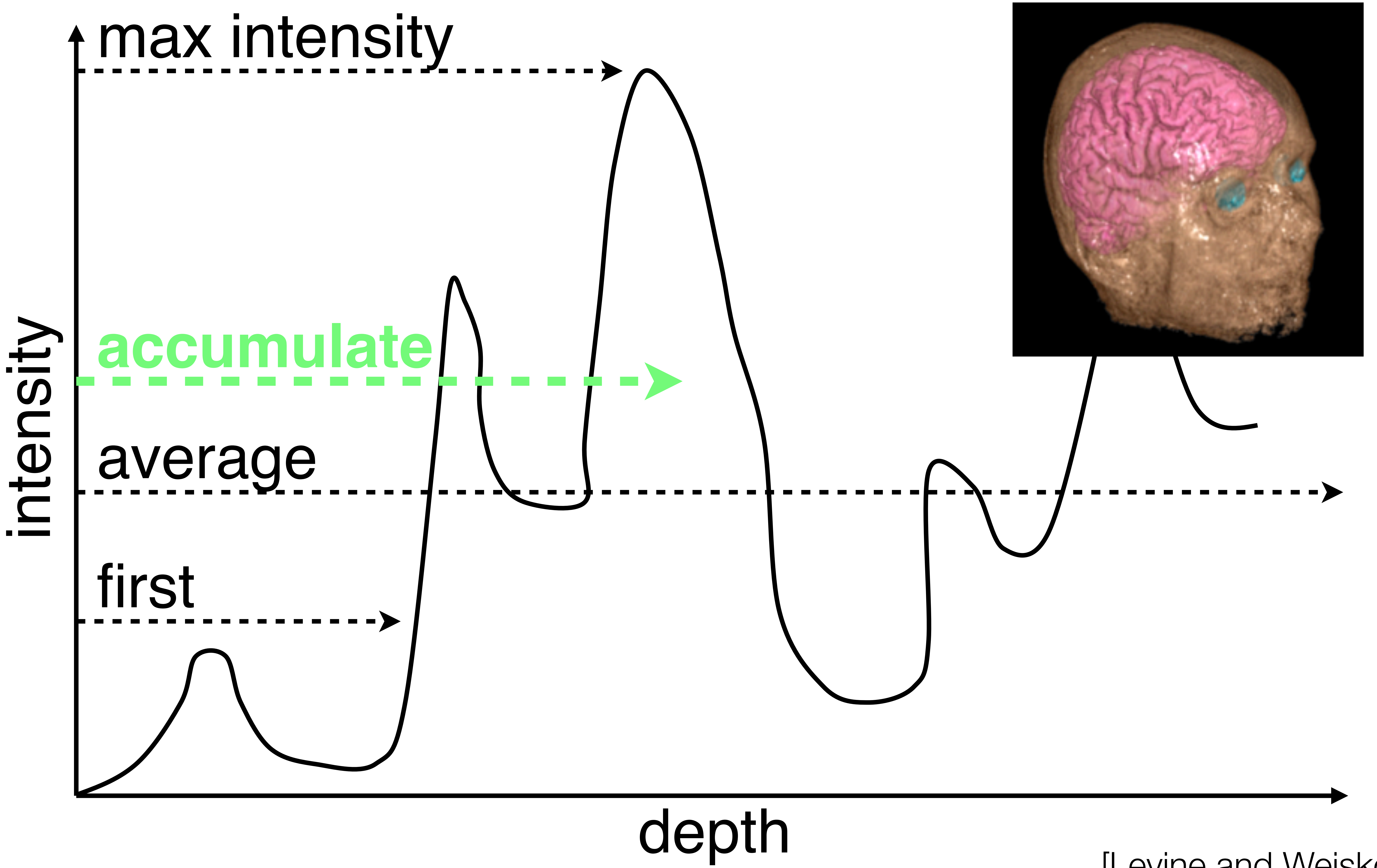
[Levine and Weiskopf/Machiraju/Möller]

Types of Compositing



[Levine and Weiskopf/Machiraju/Möller]

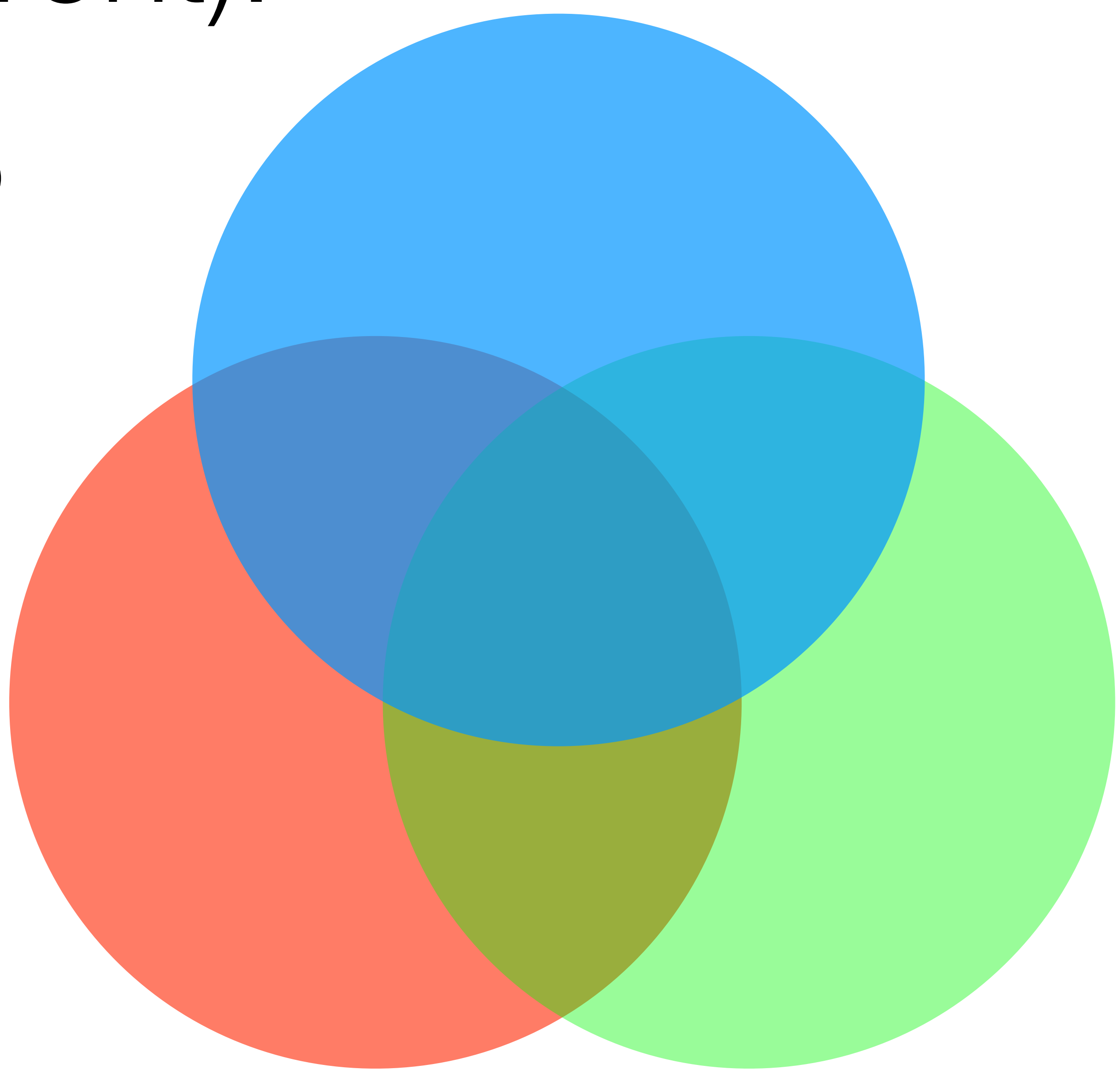
Types of Compositing



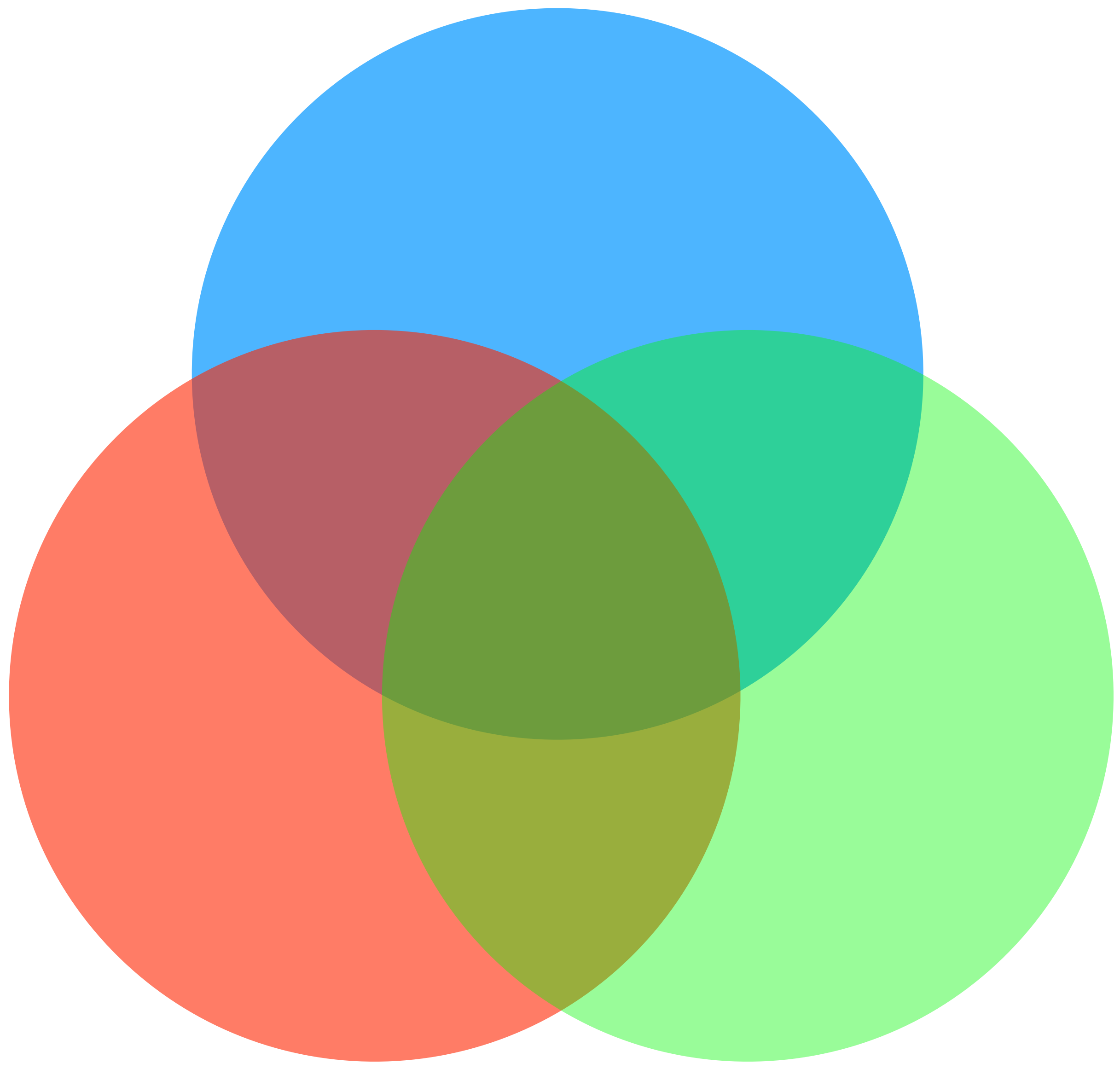
[Levine and Weiskopf/Machiraju/Möller]

Accumulation

- If we're not just calculating a single number (max, average) or a position (first), how do we determine the accumulation?
- Assume each value has an associated color (c) and opacity (α)
- Over operator (back-to-front):
  - $c = \alpha_f \cdot c_f + (1-\alpha_f) \cdot \alpha_b \cdot c_b$
  - $\alpha = \alpha_f + (1-\alpha_f) \cdot \alpha_b$
- Order is important!



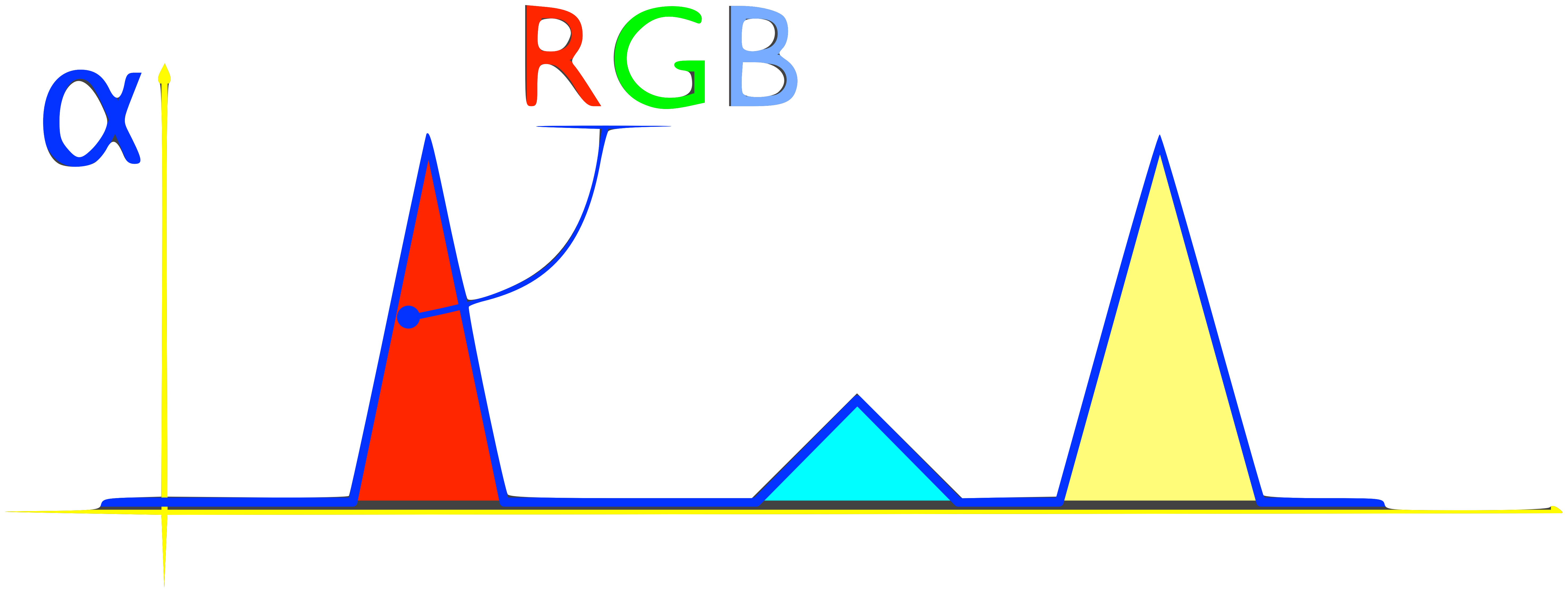
Blue Last



Blue First

Transfer Functions

- Where do the colors and opacities come from?
- Idea is that each voxel emits/absorbs light based on its scalar value
- …but users get to choose how that happens
- x-axis: color region definitions, y-axis: opacity



[Kindlmann]

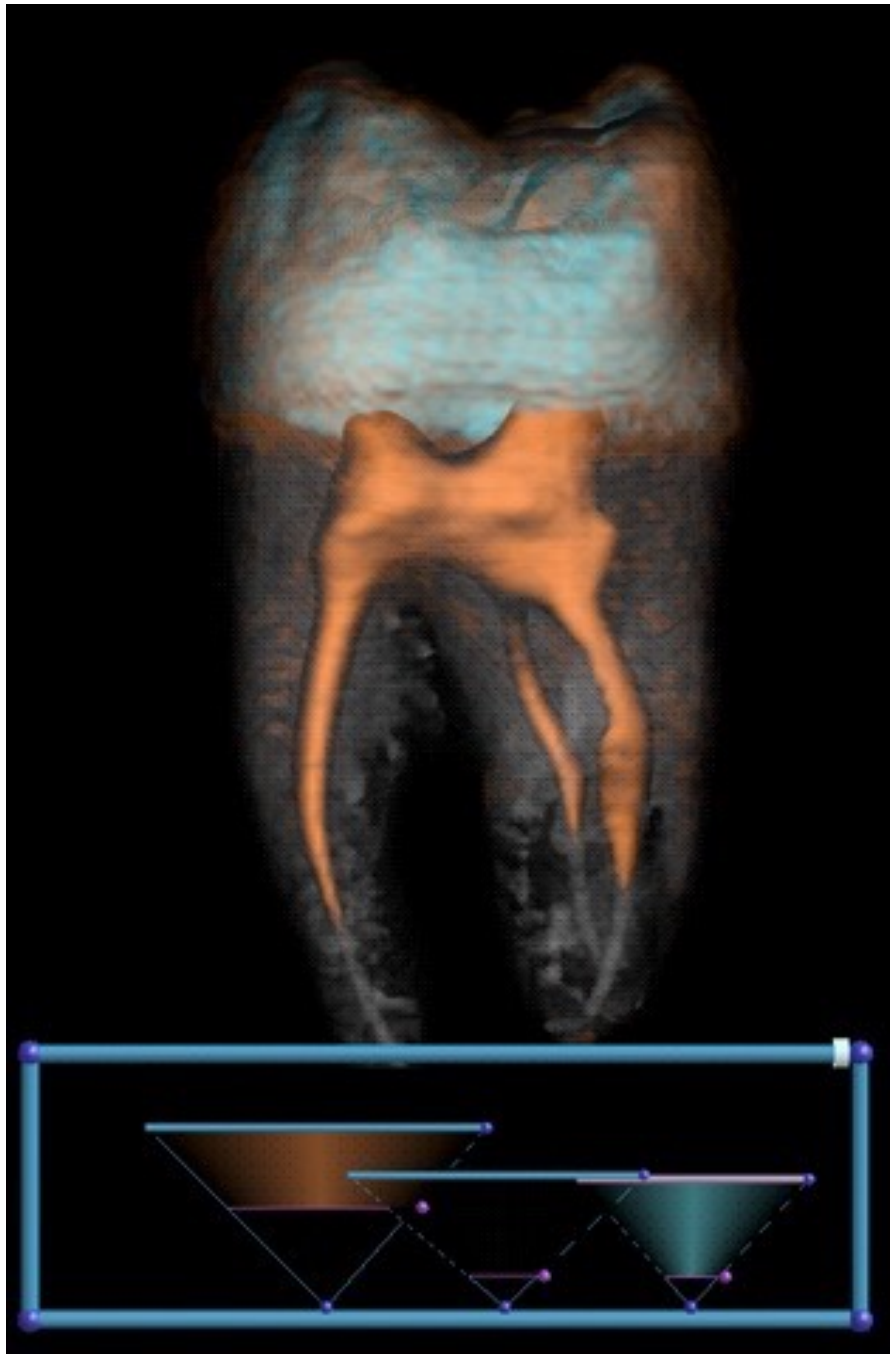
Transfer Function Design

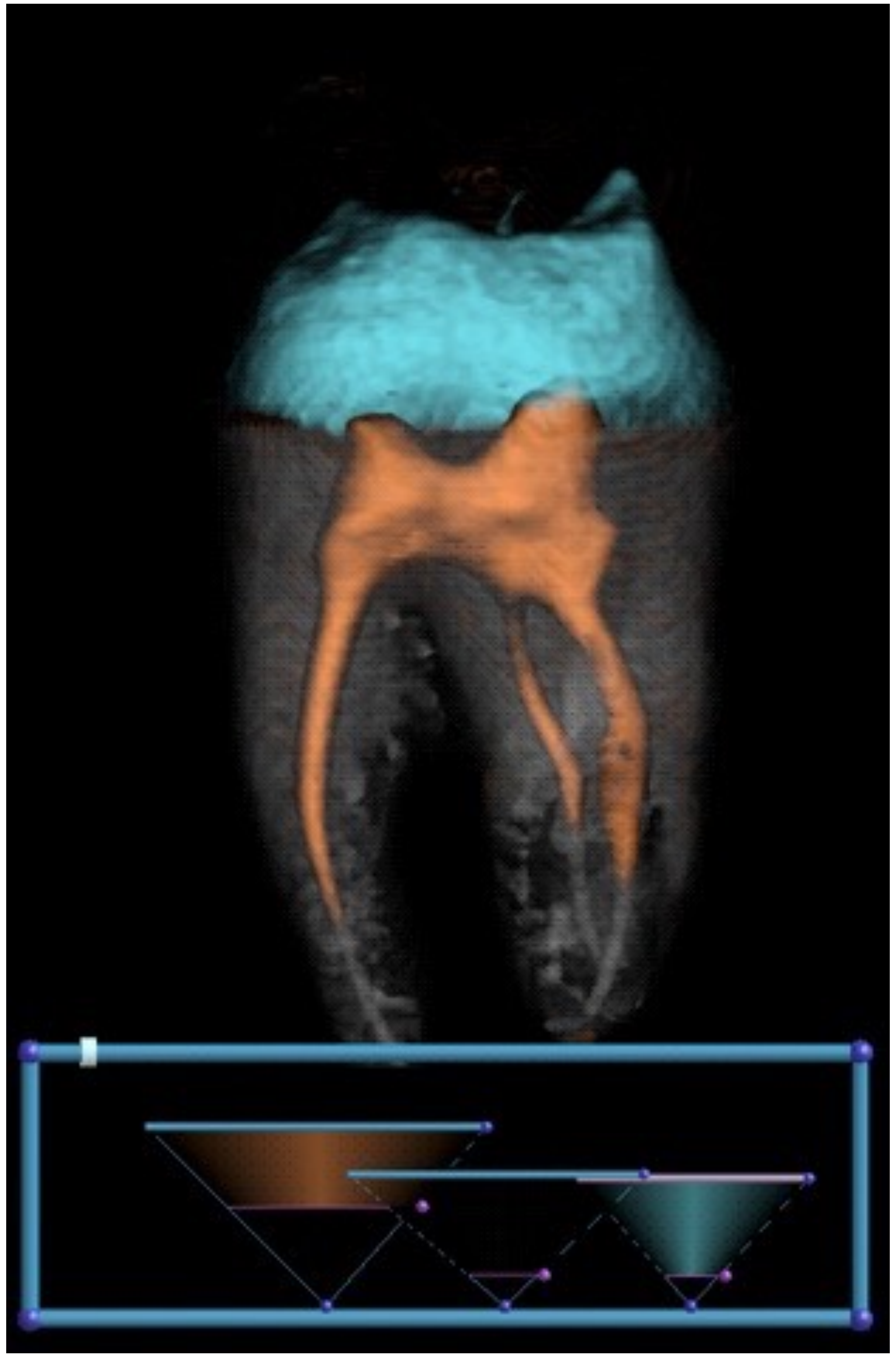
- Transfer function **design** is non-trivial!
- Lots of tools to help visualization designers to create good transfer functions
- Histograms, more attributes than just value like gradient magnitude

Multidimensional Transfer Functions

[J. Kniss]

Multidimensional Transfer Functions



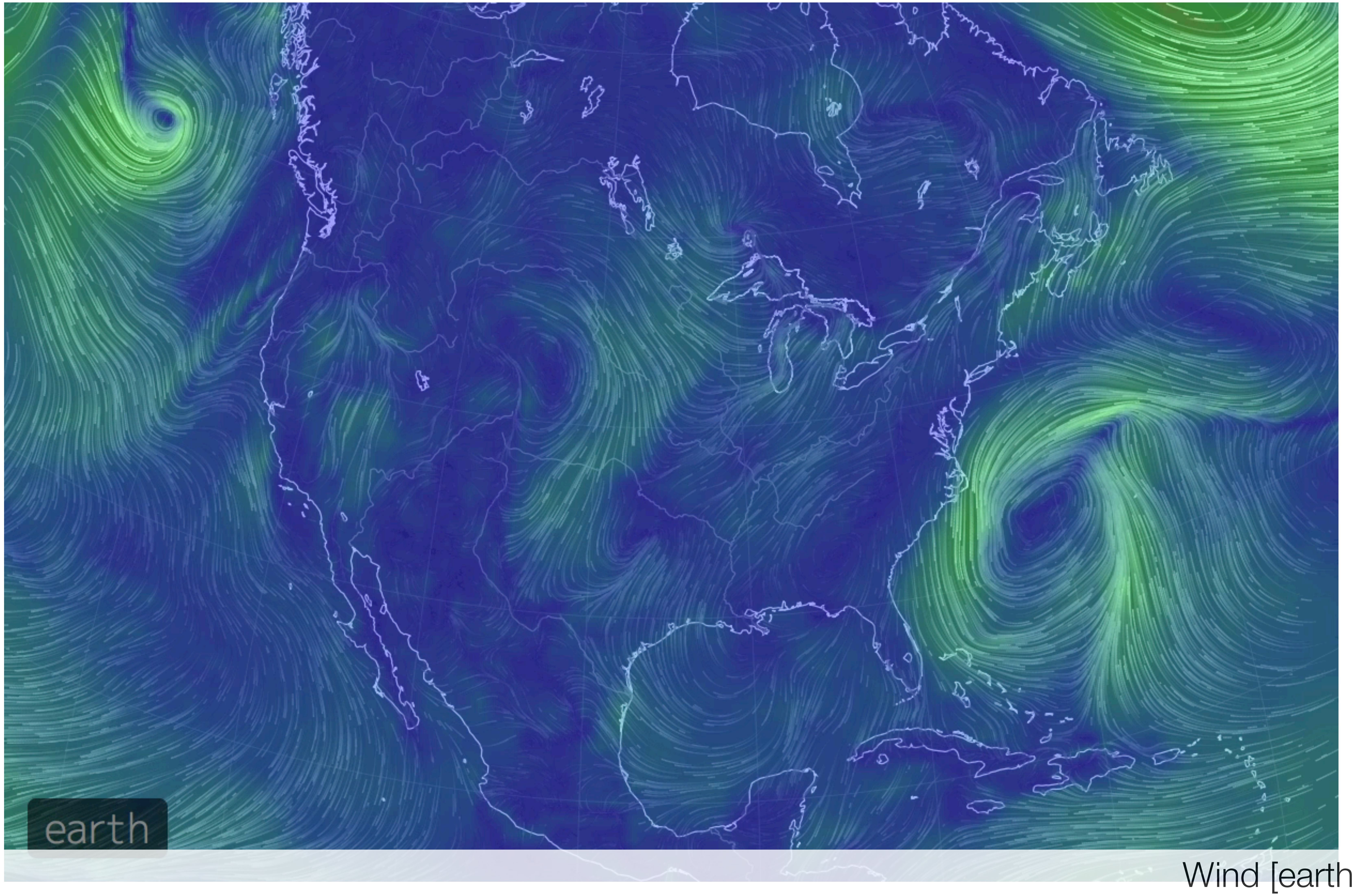


[J. Kniss]

ParaView Examples

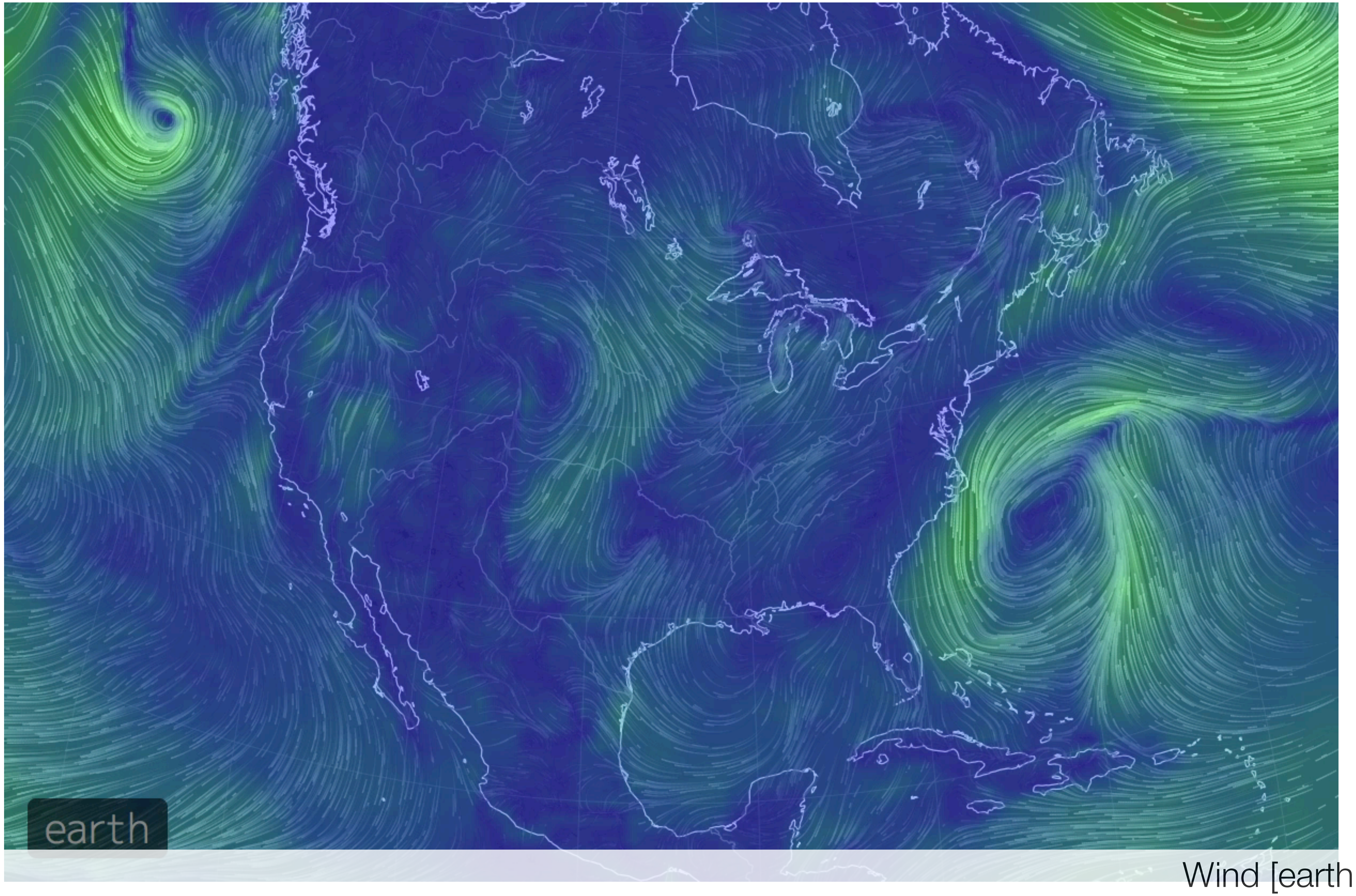
Vector Field Visualization

Examples of Vector Fields



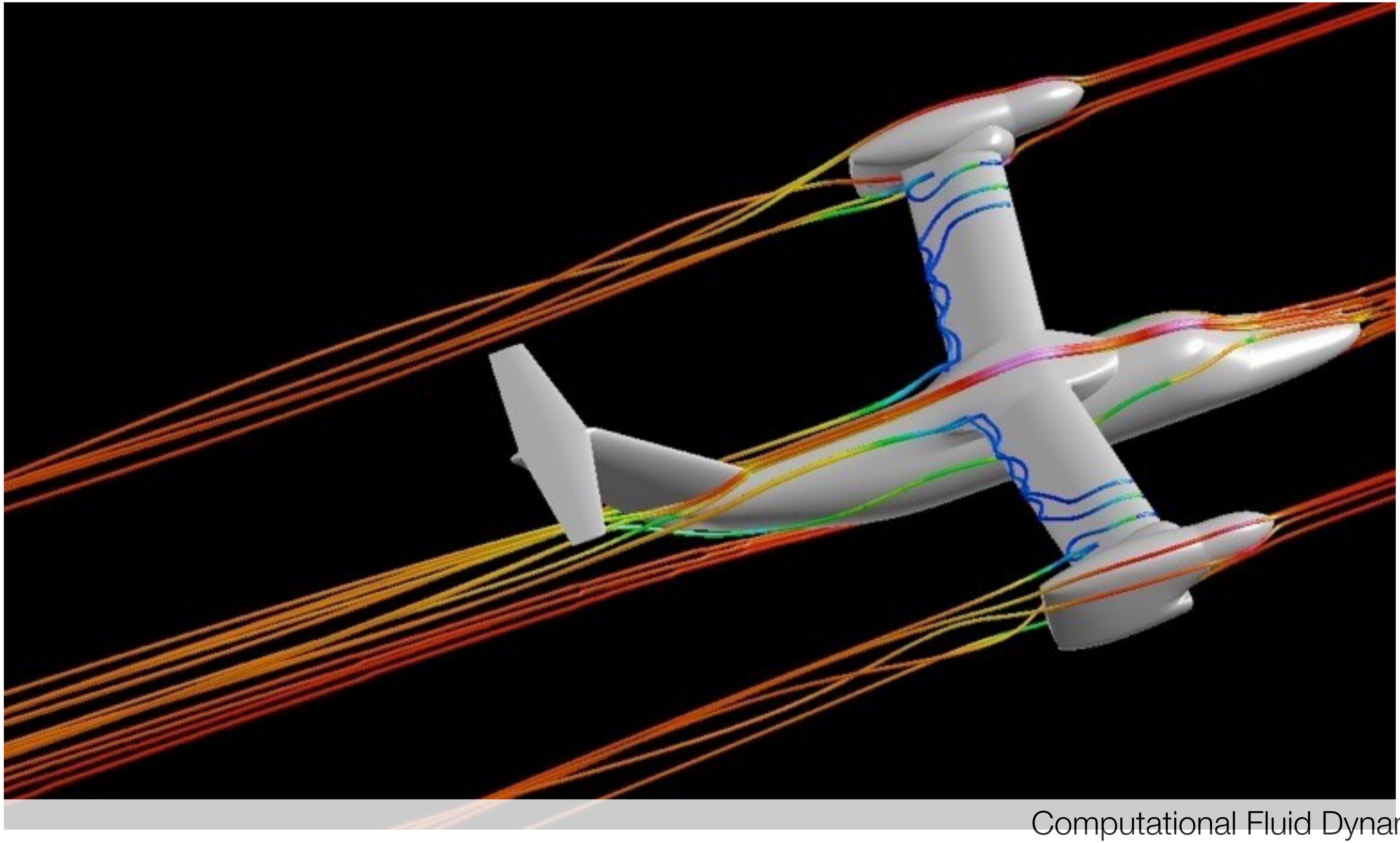
Wind [earth.nullschool.net, 2014]

Examples of Vector Fields



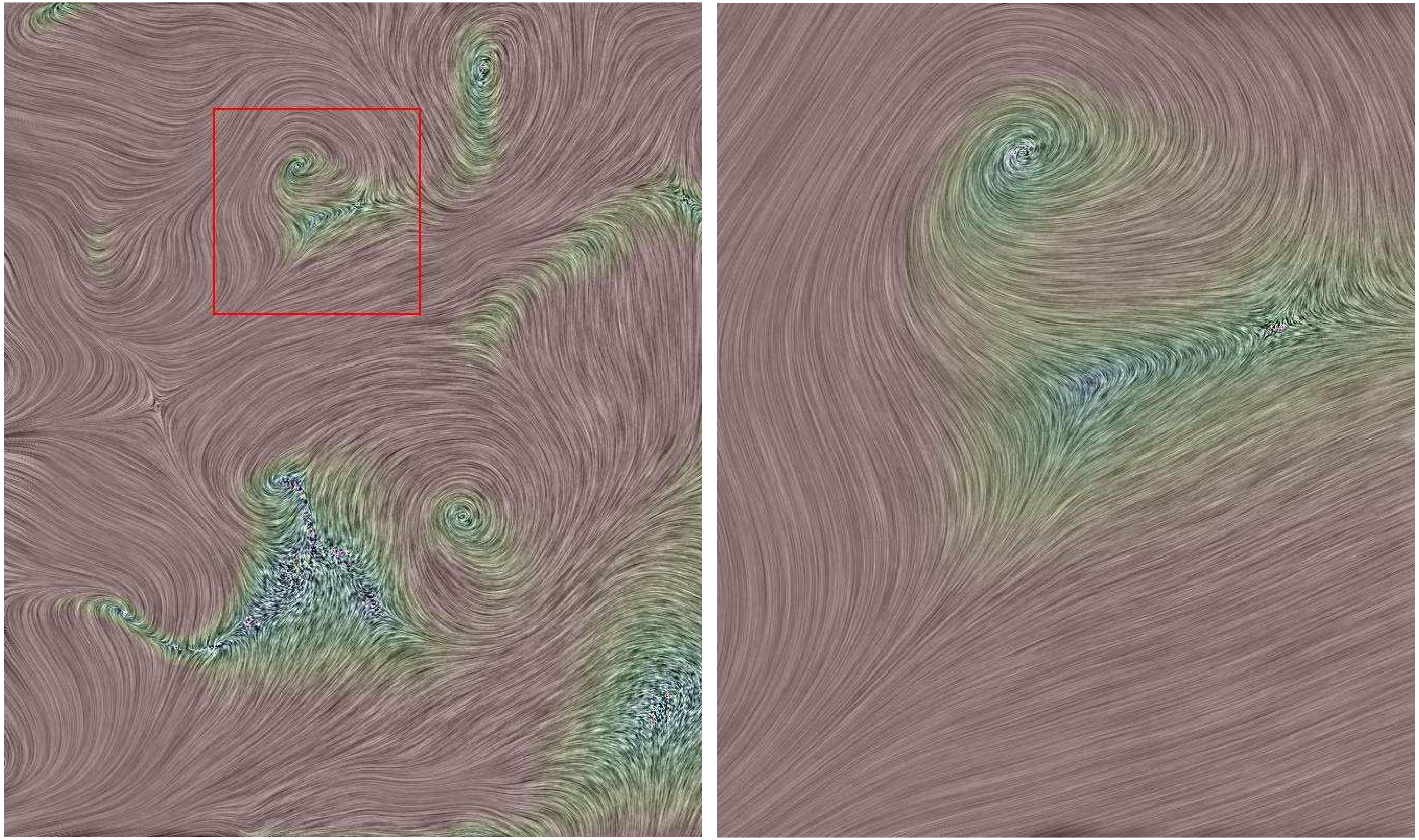
Wind [earth.nullschool.net, 2014]

Examples of Vector Fields



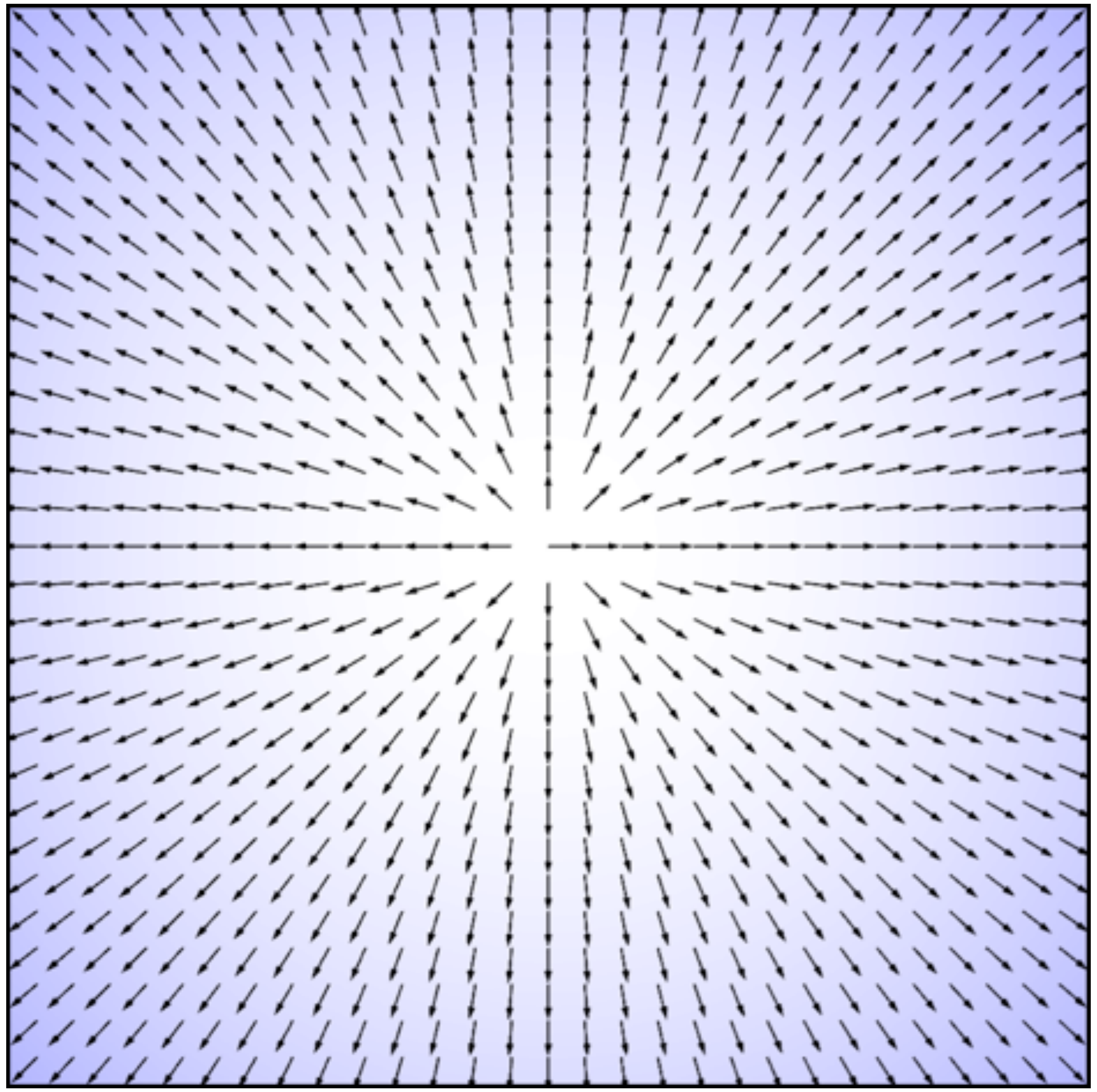
Computational Fluid Dynamics [newmerical]

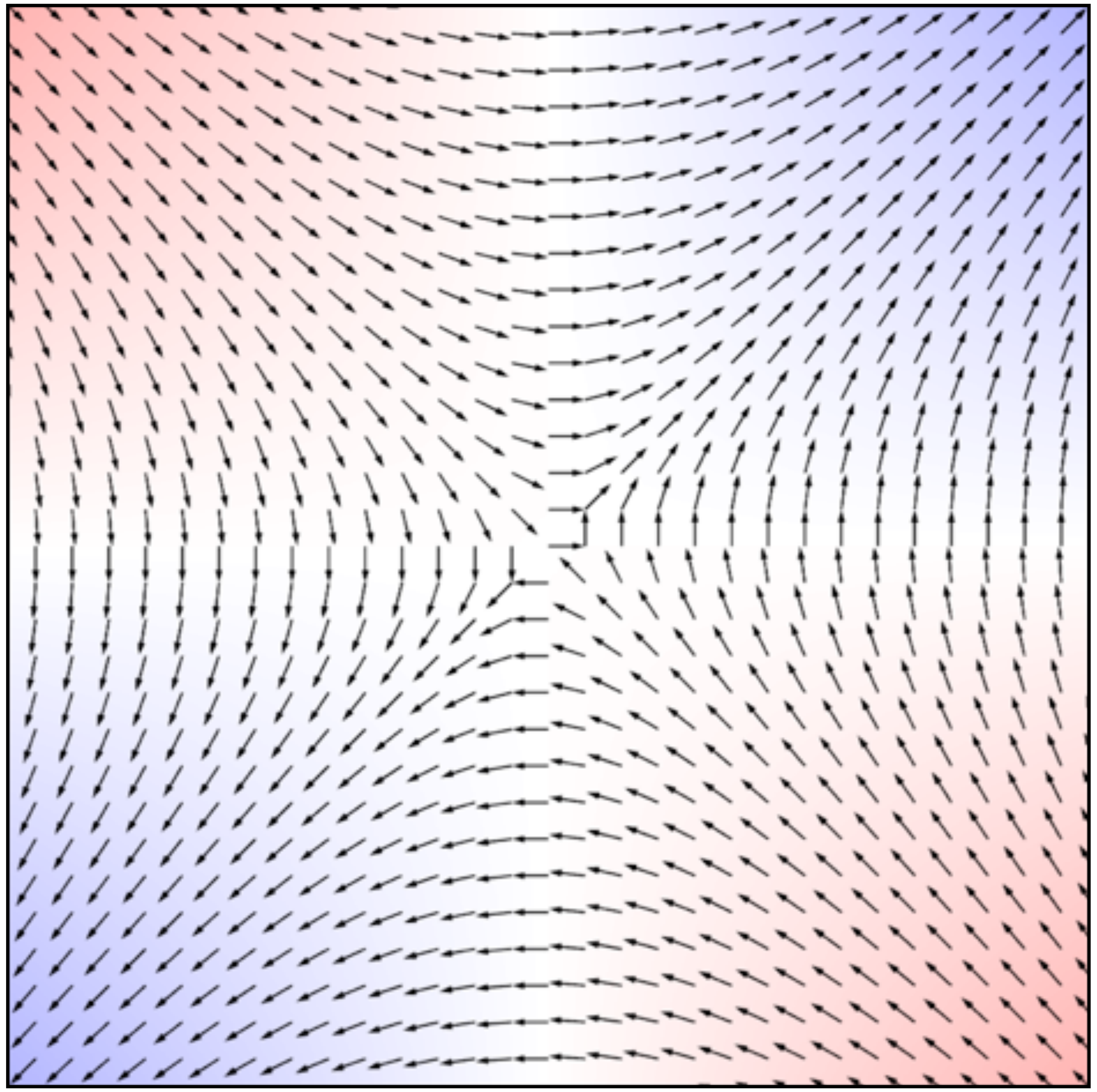
Examples of Vector Fields



Earthquake Ground Surface Movement [H. Yu et. al., SC2004]

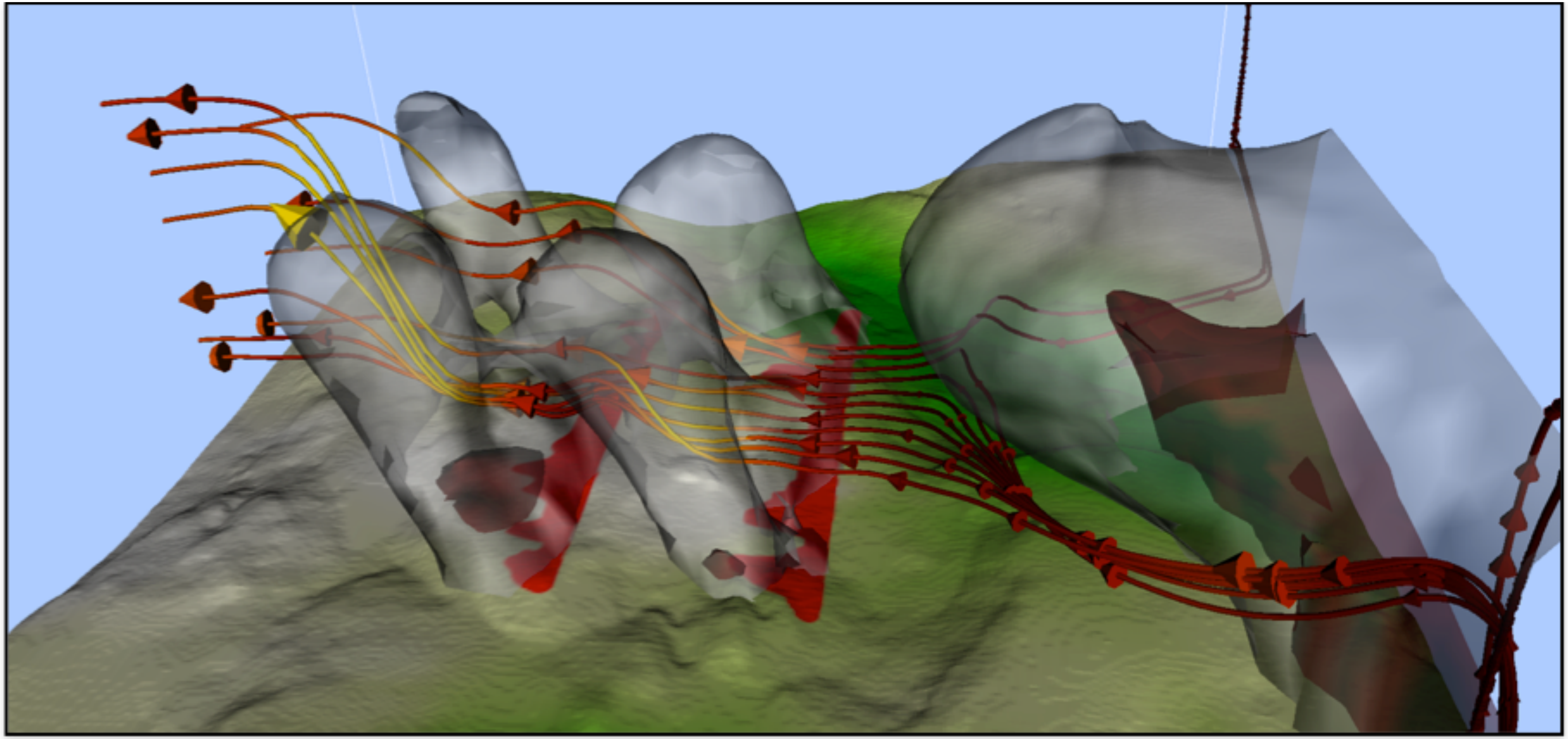
Examples of Vector Fields





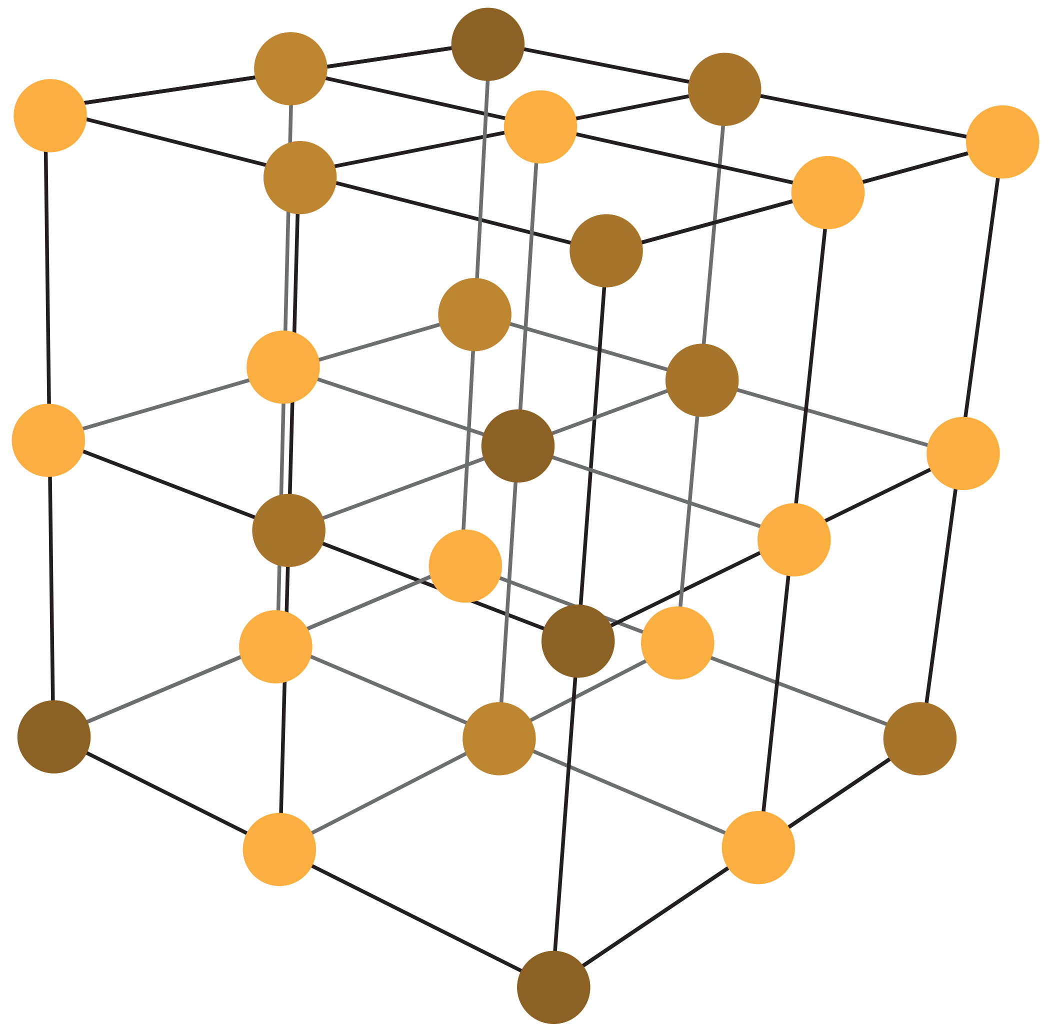
Gradient Vector Fields

Examples of Vector Fields



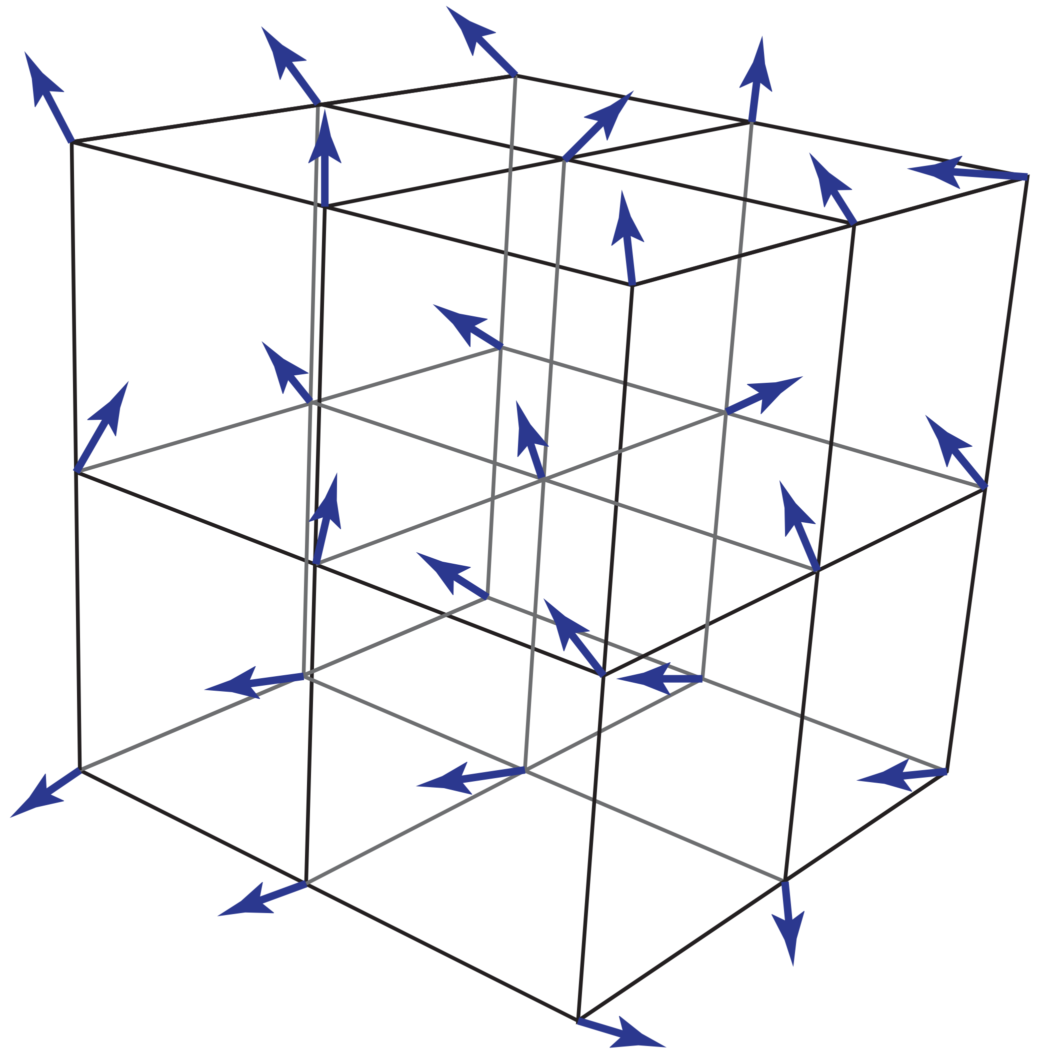
Wildfire Modeling [E. Anderson]

Fields in Visualization



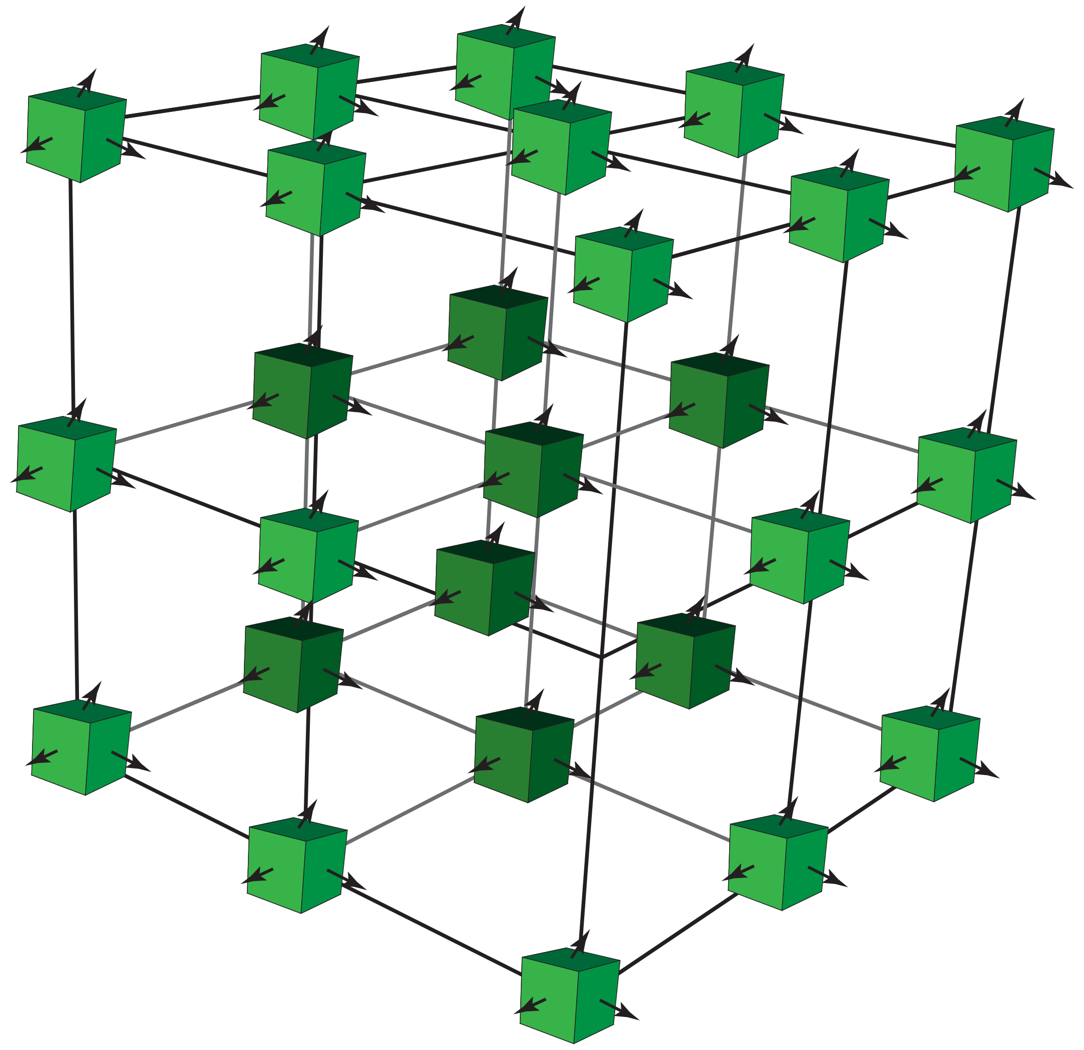
Scalar Fields

(Order-0 Tensor Fields)



Vector Fields

(Order-1 Tensor Fields)



Tensor Fields

(Order-2+)

Each point in space has an associated...

$$s_0$$

Scalar

$$\begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}$$

Vector

$$\begin{bmatrix} \sigma_{00} & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_{11} & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_{22} \end{bmatrix}$$

Tensor

Visualizing Vector Fields

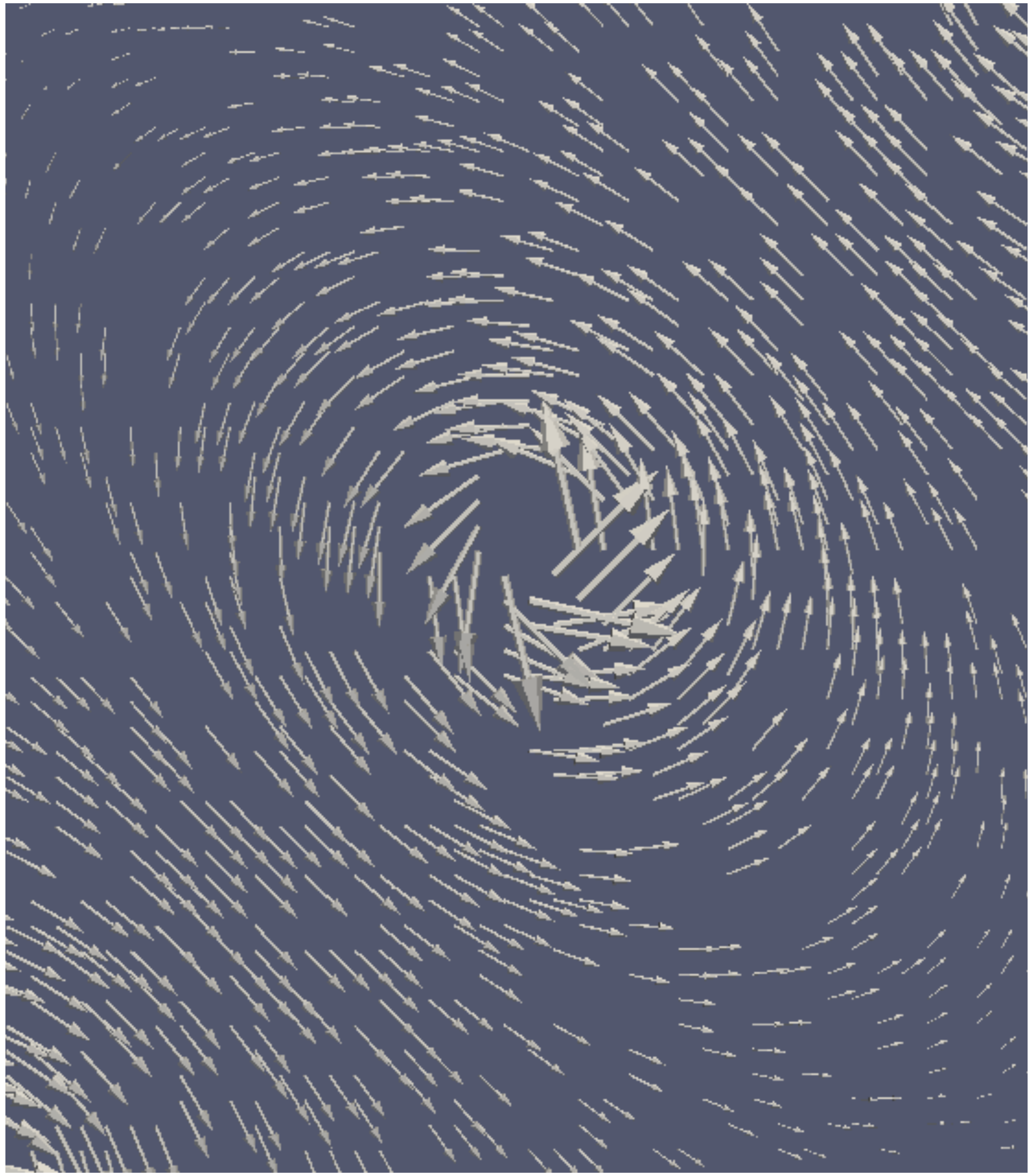
- Direct: Glyphs, Render statistics as scalars
- Geometry: Streamlines and variants
- Textures: Line Integral Convolution (LIC)
- Topology: Extract relevant features and draw them

Glyphs

- Represent each vector with a symbol
- Hedgehogs are primitive glyphs (glyph is a line)
- ParaView Example

Glyphs

- Represent each vector with a symbol
- Hedgehogs are primitive glyphs (glyph is a line)
- Glyphs that show direction and/or magnitude can convey more information
- If we have a separate scalar value, how might we encode that?
- Clutter issues

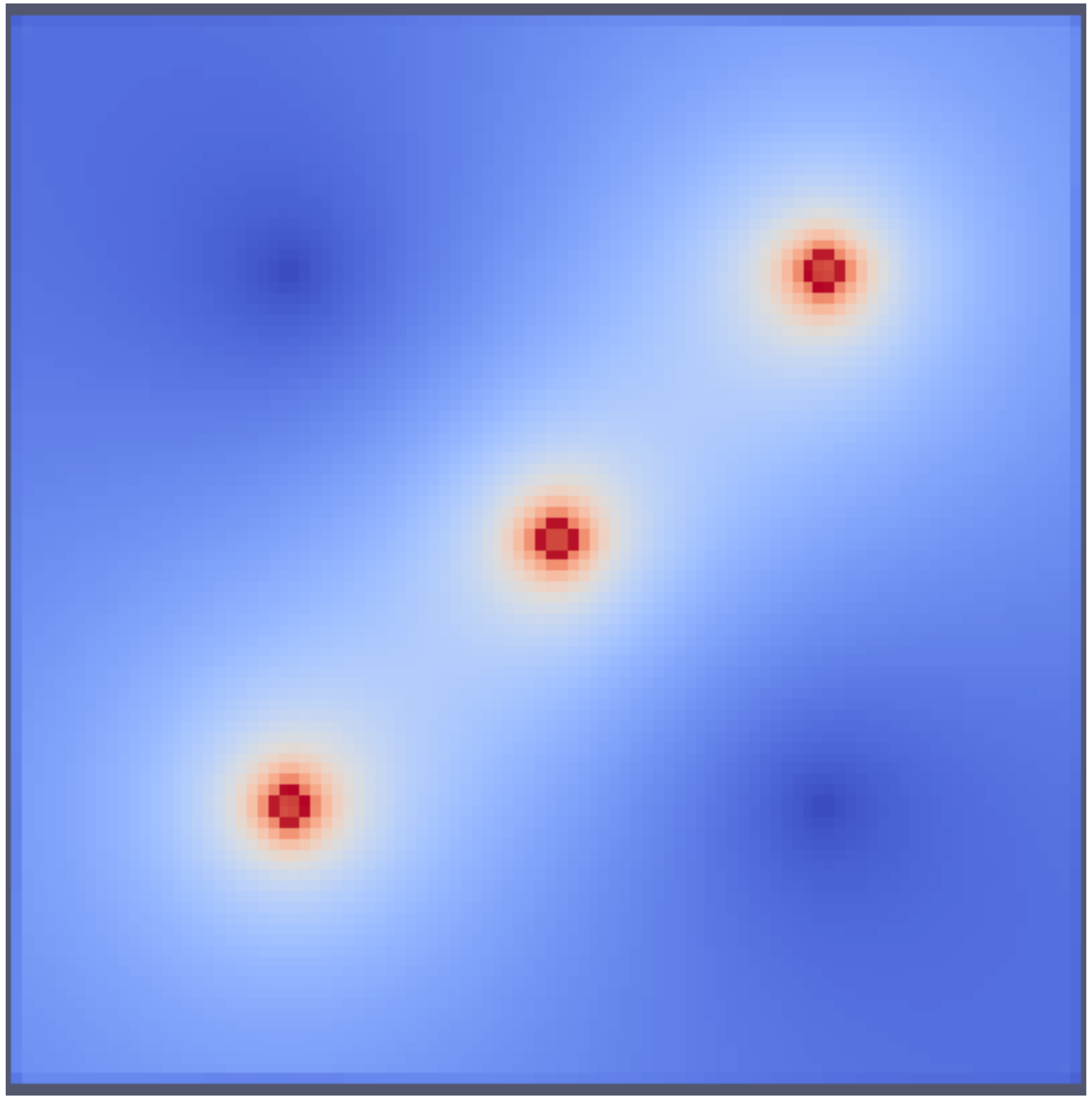


Glyphs

- For vector fields, can encode direction, magnitude, scalar value
- Good:
  - Show precise local measures
  - Can encode scalar information as color
- Bad:
  - Possible sampling issues
  - Clutter (Occlusion): Can remove some points to help
  - Clutter is worse in higher dimensions

Rendering Vector Field Statistics as Scalars

- Many statistics we can compute for vector fields:
  - Magnitude
  - Vorticity
  - Curvature
- These are scalars, can color with our scalar field visualization techniques (e.g. volume rendering)



[Color indicates vector magnitude]

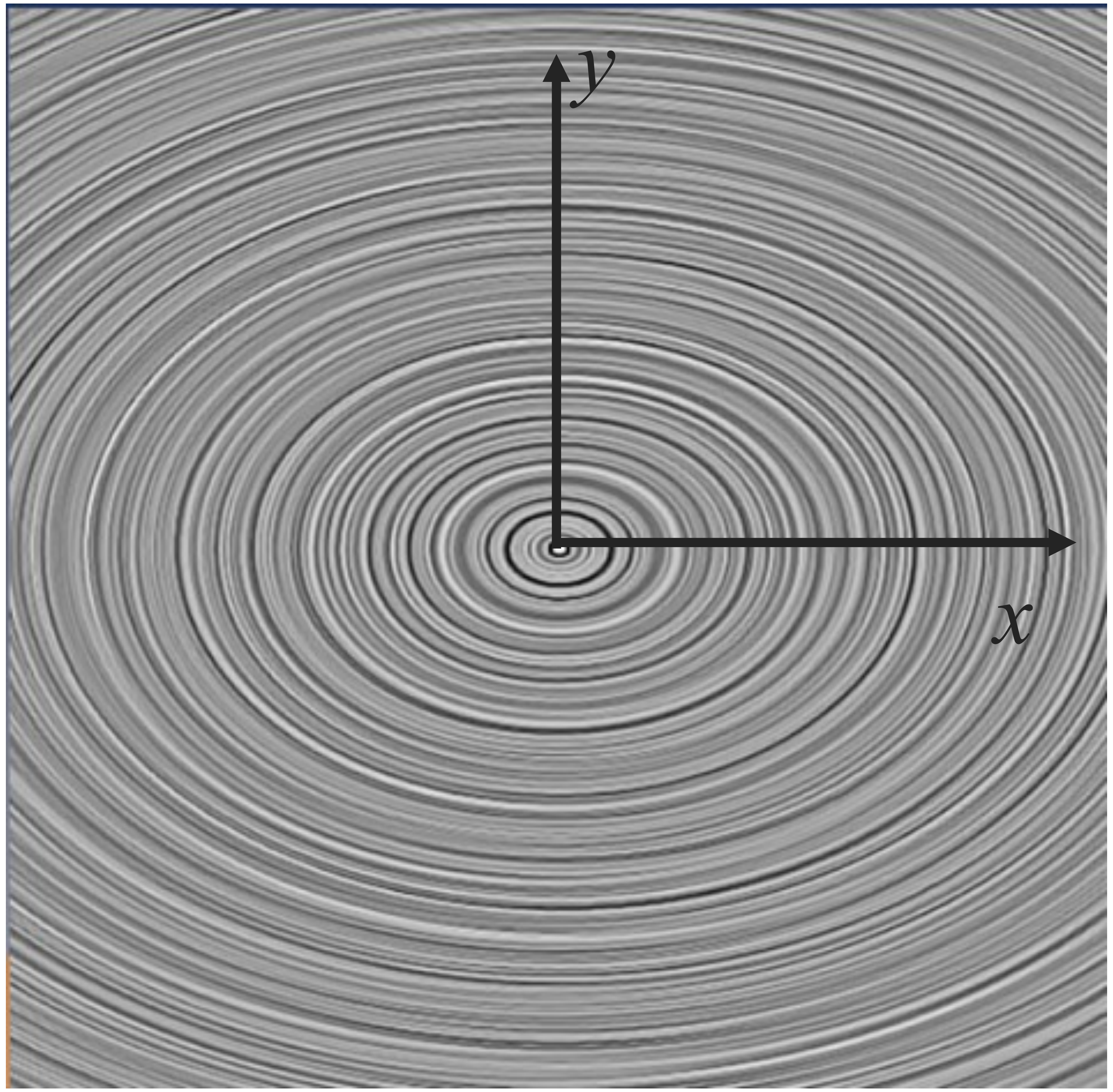
Streamlines & Variants

- Trace a line along the direction of the vectors
- Streamlines are always tangent to the vector field
- Basic Particle Tracing:

1. Set a starting point (seed)
2. Take a step in the direction of the vector at that point
3. Adjust direction based on the vector where you are now
4. Go to Step 2 and Repeat

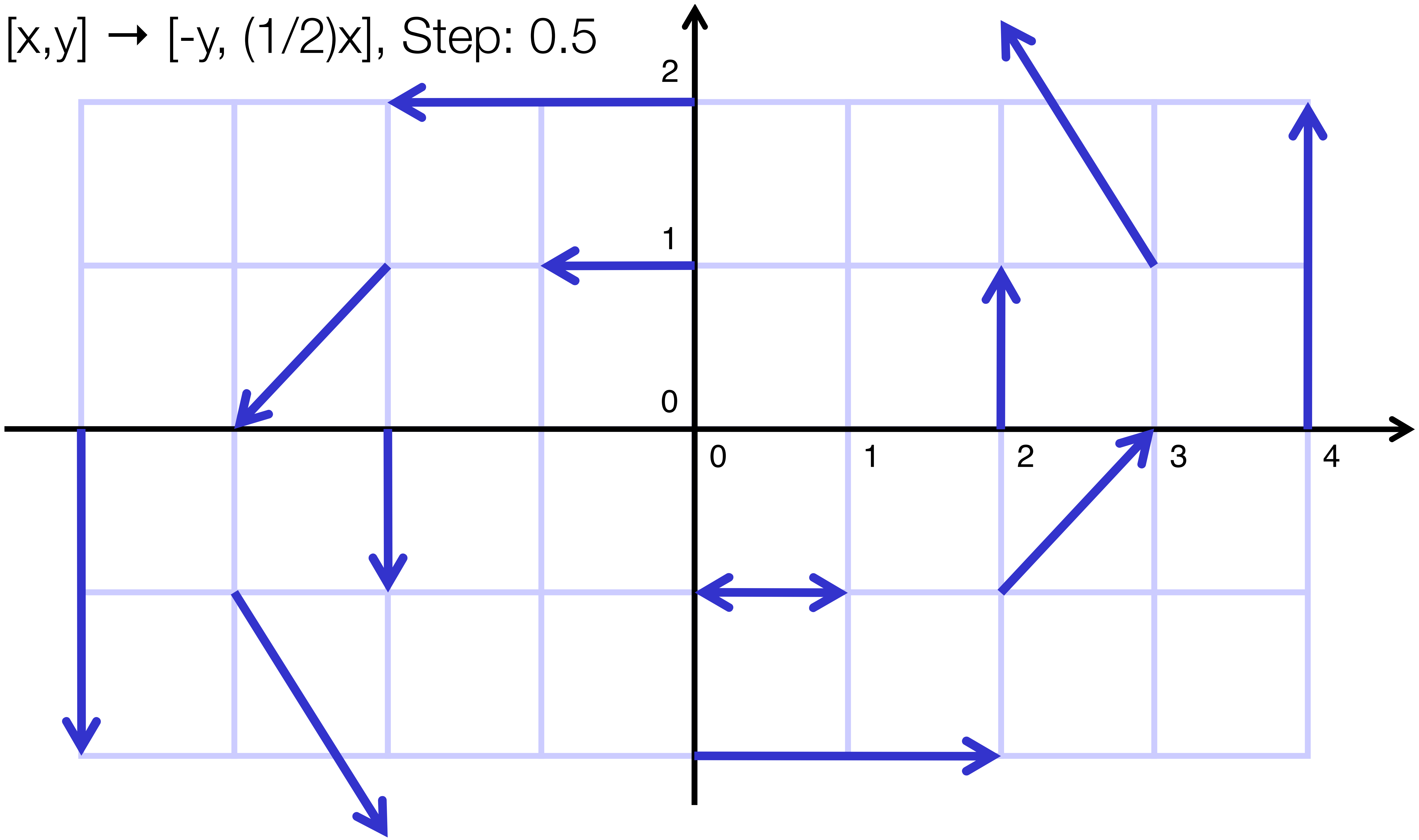
Example

- Elliptical path
- Suppose we have the actual equation
- Given point (x,y), the vector is at that point is $[v_x, v_y]$ where
  - $v_x = -y$
  - $v_y = (1/2)x$
- Want a streamline starting at (0,-1)



[LIC (not streamlines!) via Levine]

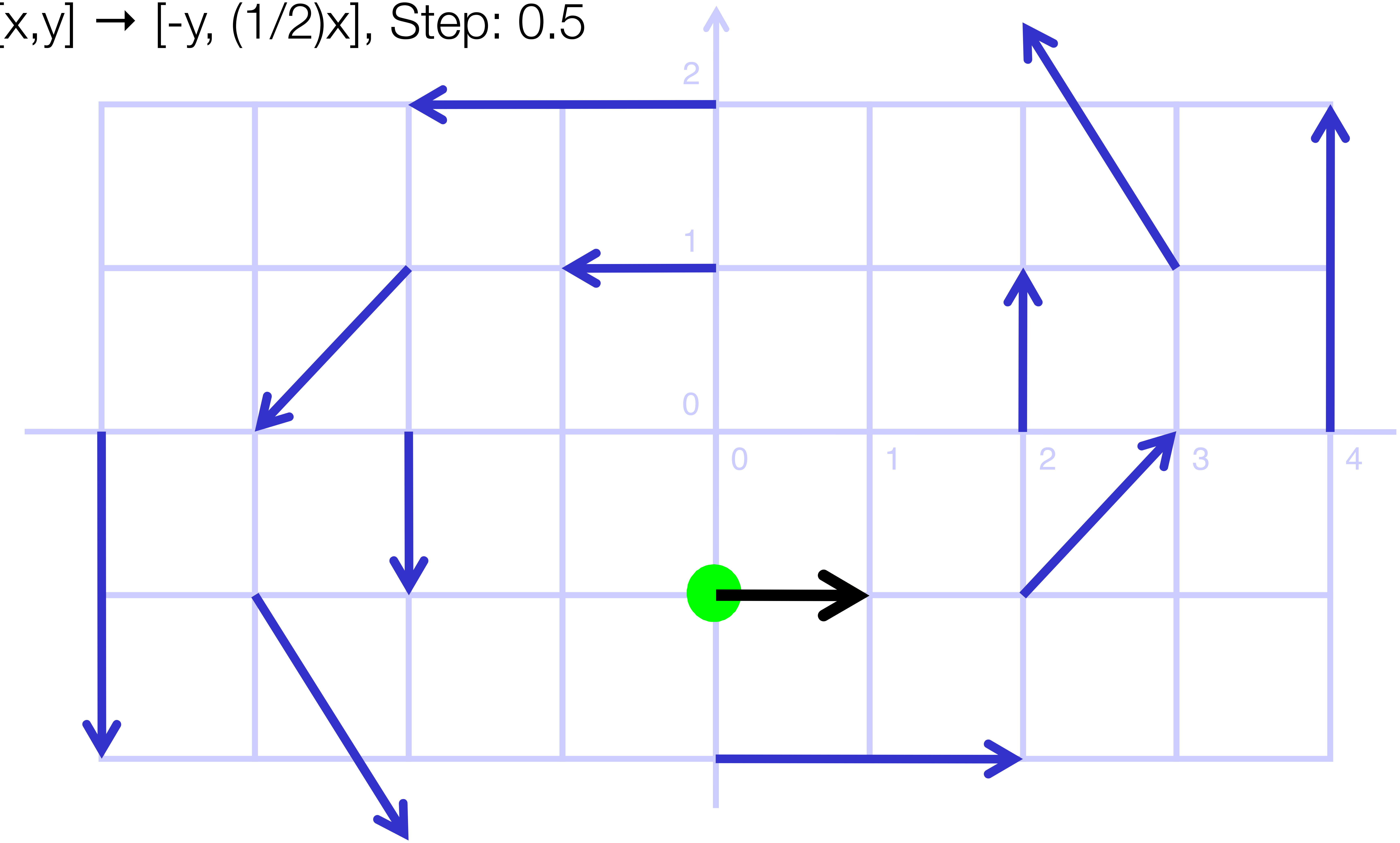
Some Glyphs



[via Levine]

Streamlines (Step 1)

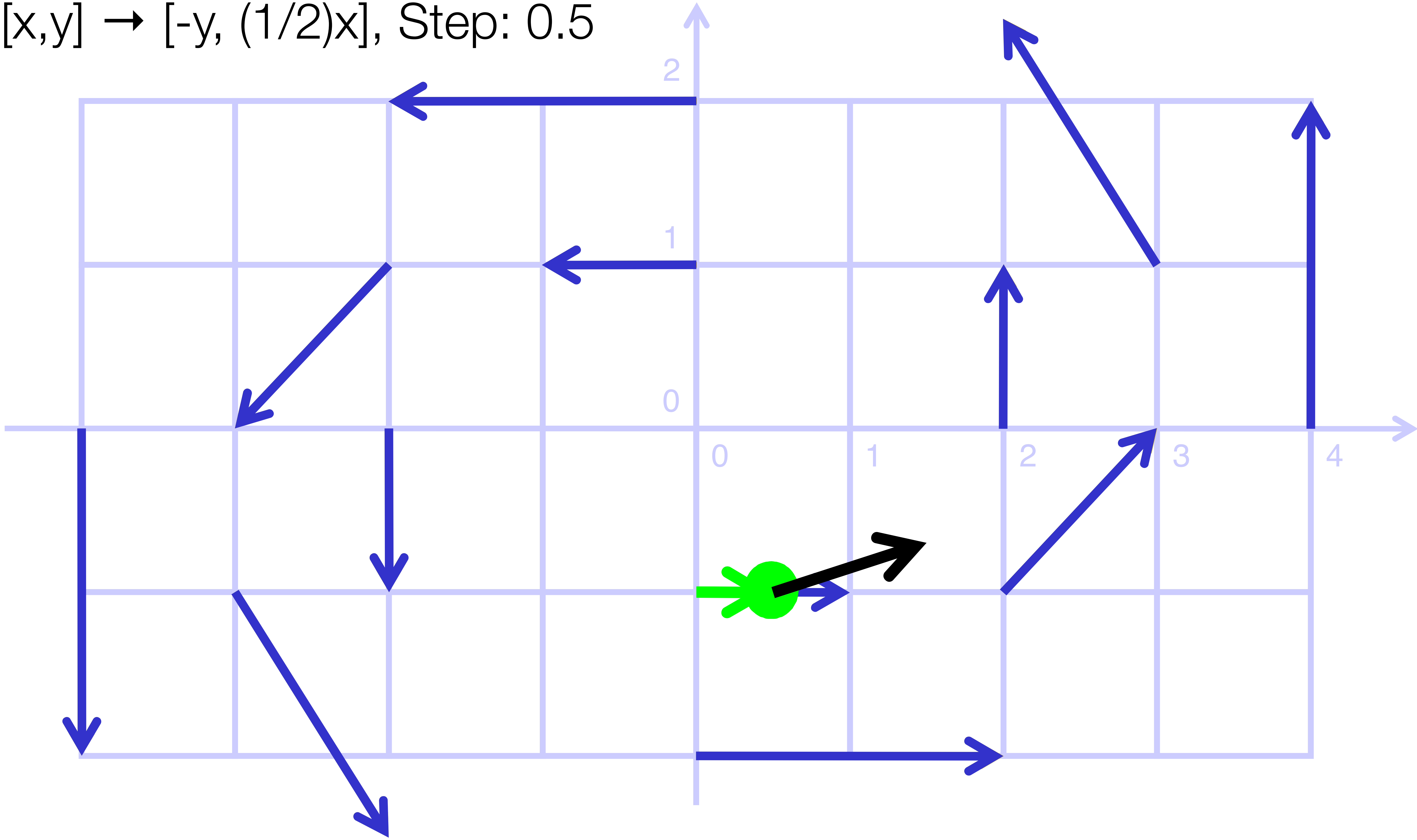
$[x,y] \rightarrow [-y, (1/2)x]$, Step: 0.5



[via Levine]

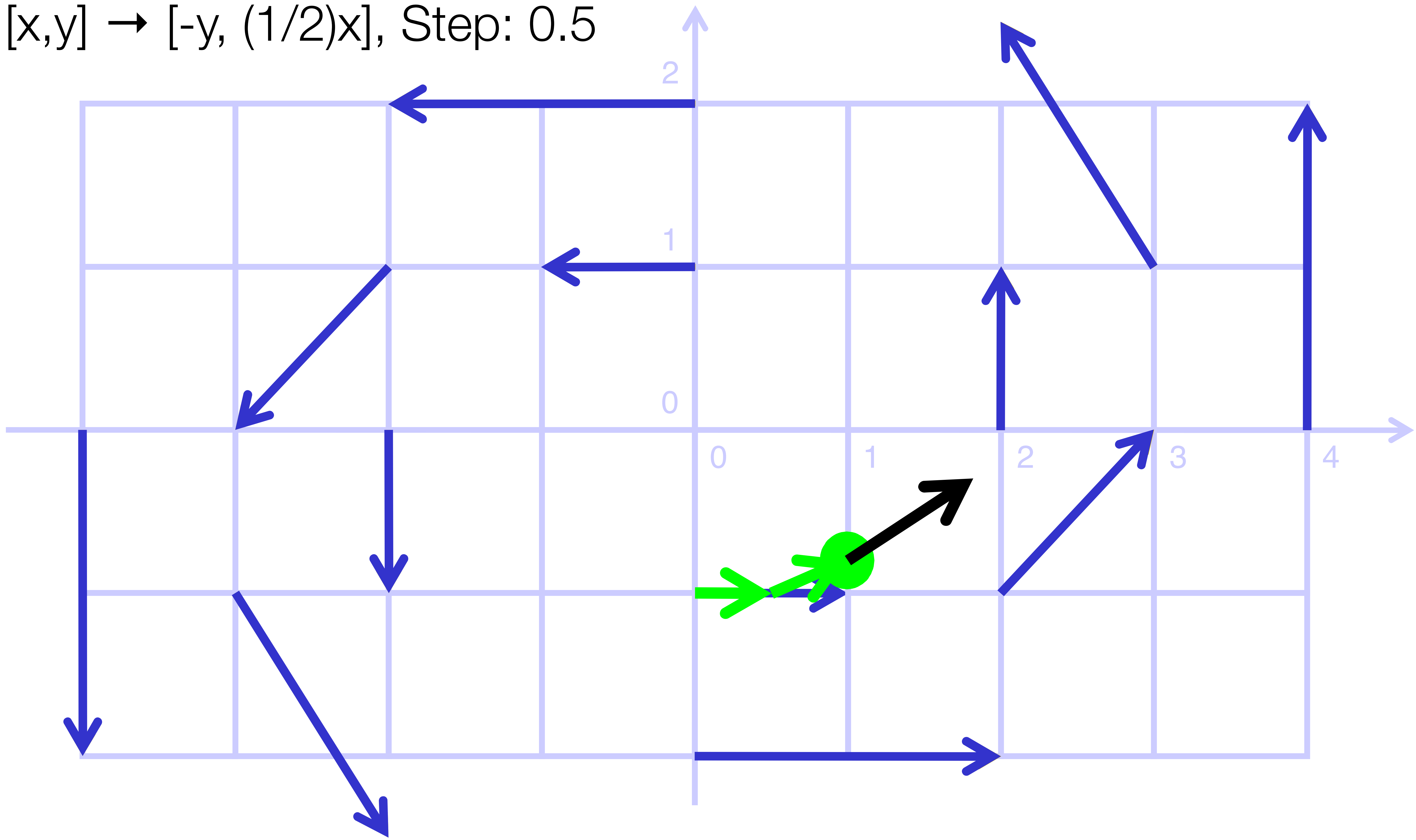
Streamlines (Step 2)

$[x,y] \to [-y, (1/2)x]$, Step: 0.5



[via Levine]

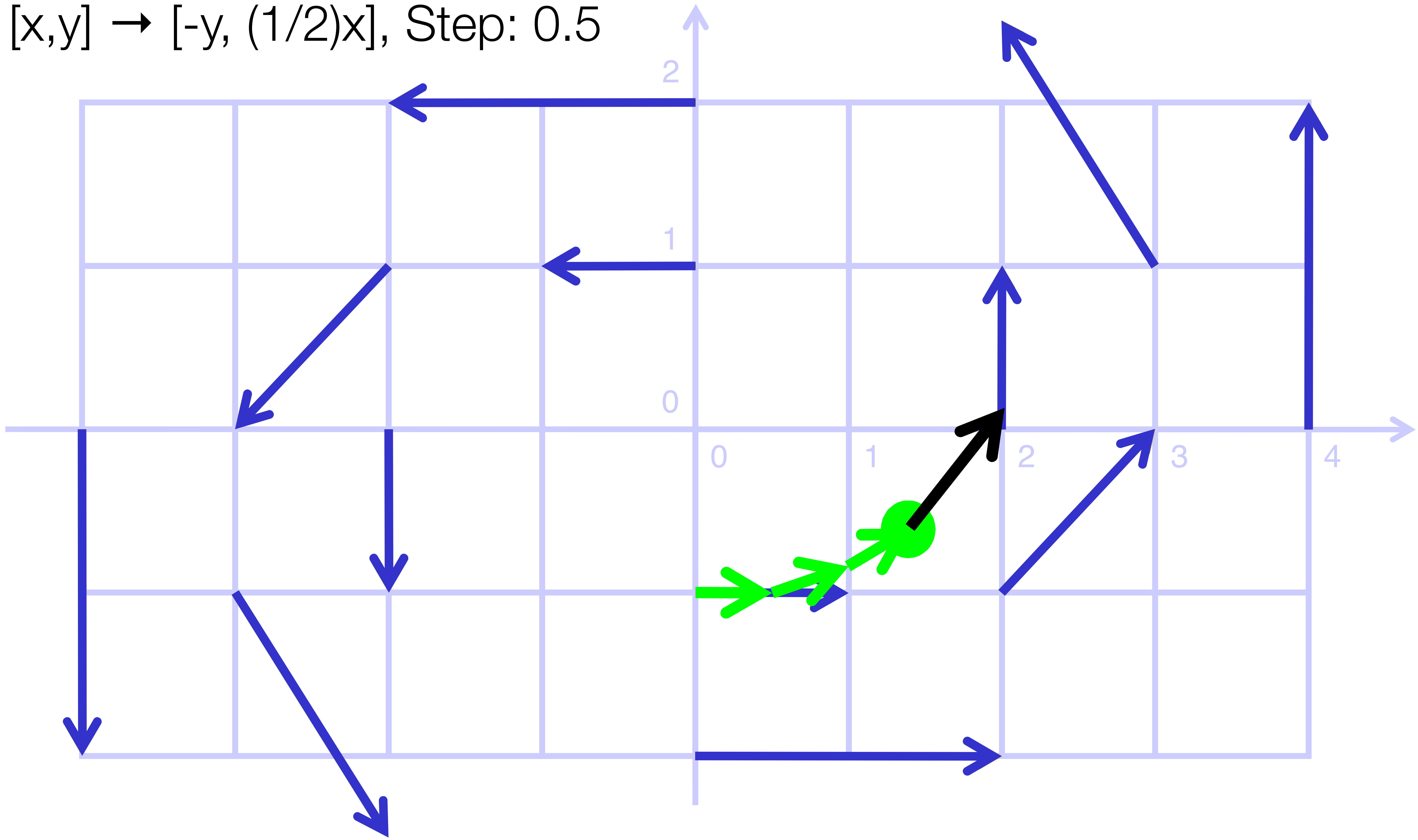
Streamlines (Step 3)



[via Levine]

Streamlines (Step 4)

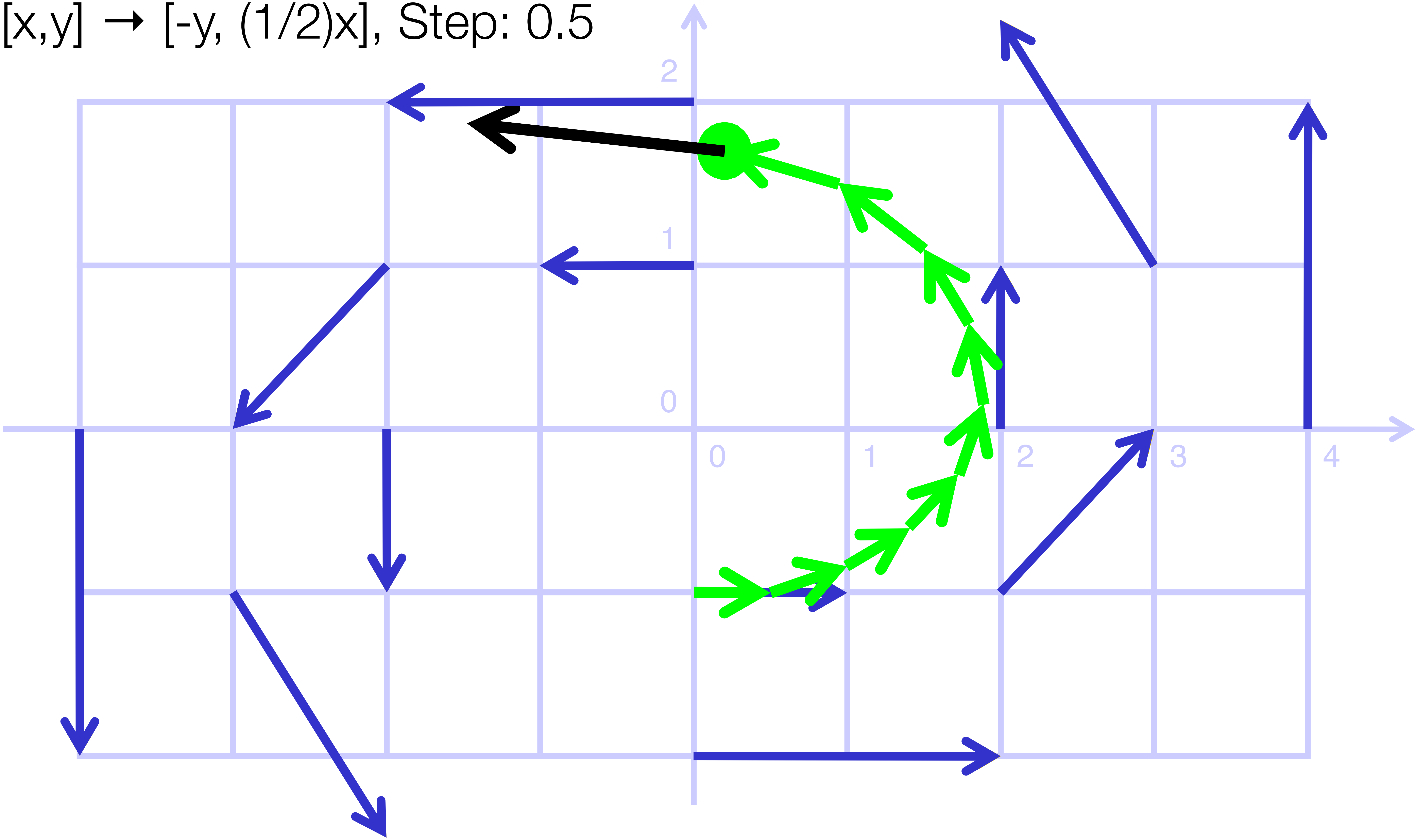
[x,y] → [-y, (1/2)x], Step: 0.5



[via Levine]

Streamlines (Step 10)

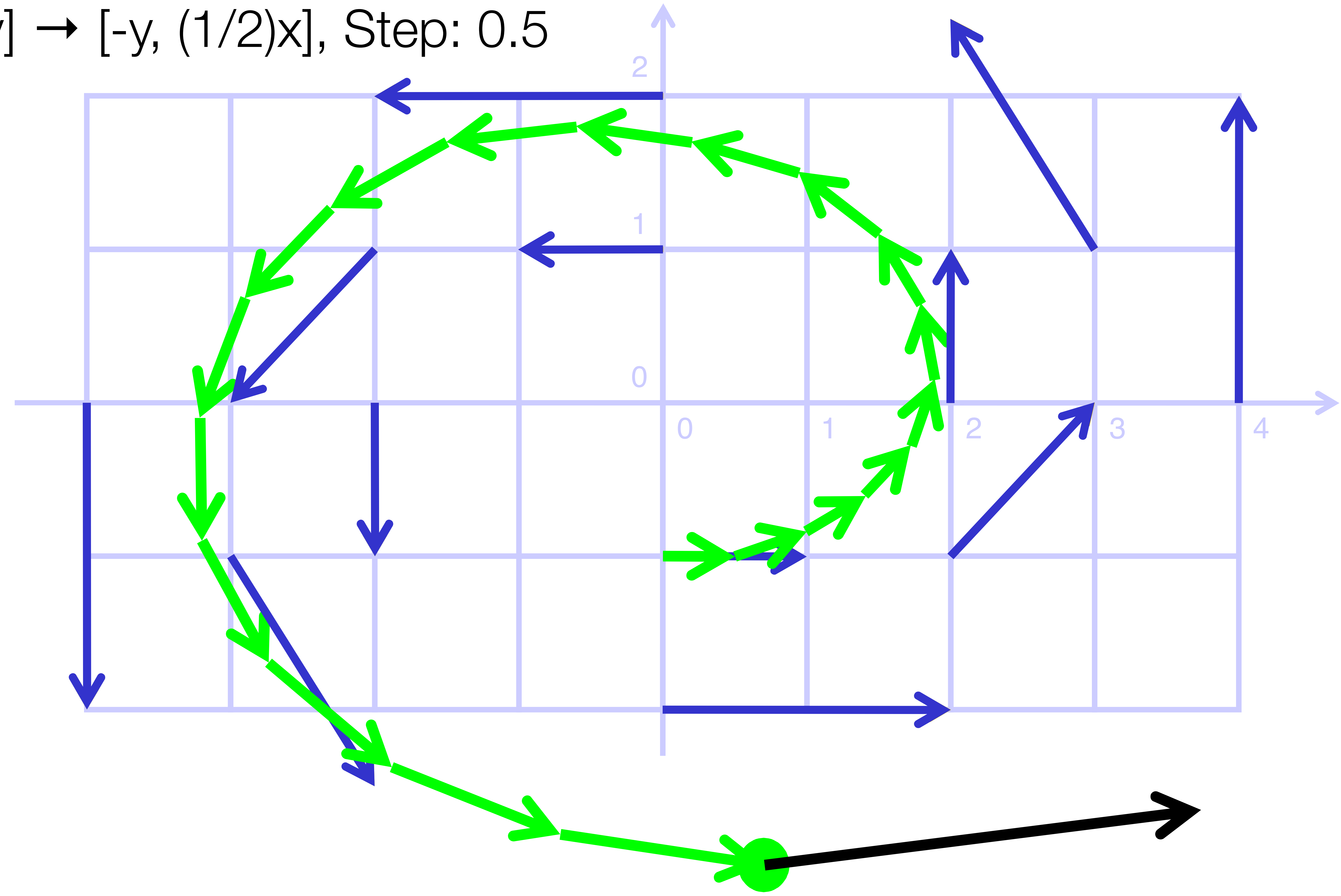
[x,y] → [-y, (1/2)x], Step: 0.5



[via Levine]

Streamlines (Step 19)

[x,y] → [-y, (1/2)x], Step: 0.5



[via Levine]

Euler Method

- Seeking to approximate integration of the velocity over time
- Euler method is the starting point for approximating this
- Problems?

Euler Method

- Seeking to approximate integration of the velocity over time
- Euler method is the starting point for approximating this
- Problems?
  - Choice of step size is important

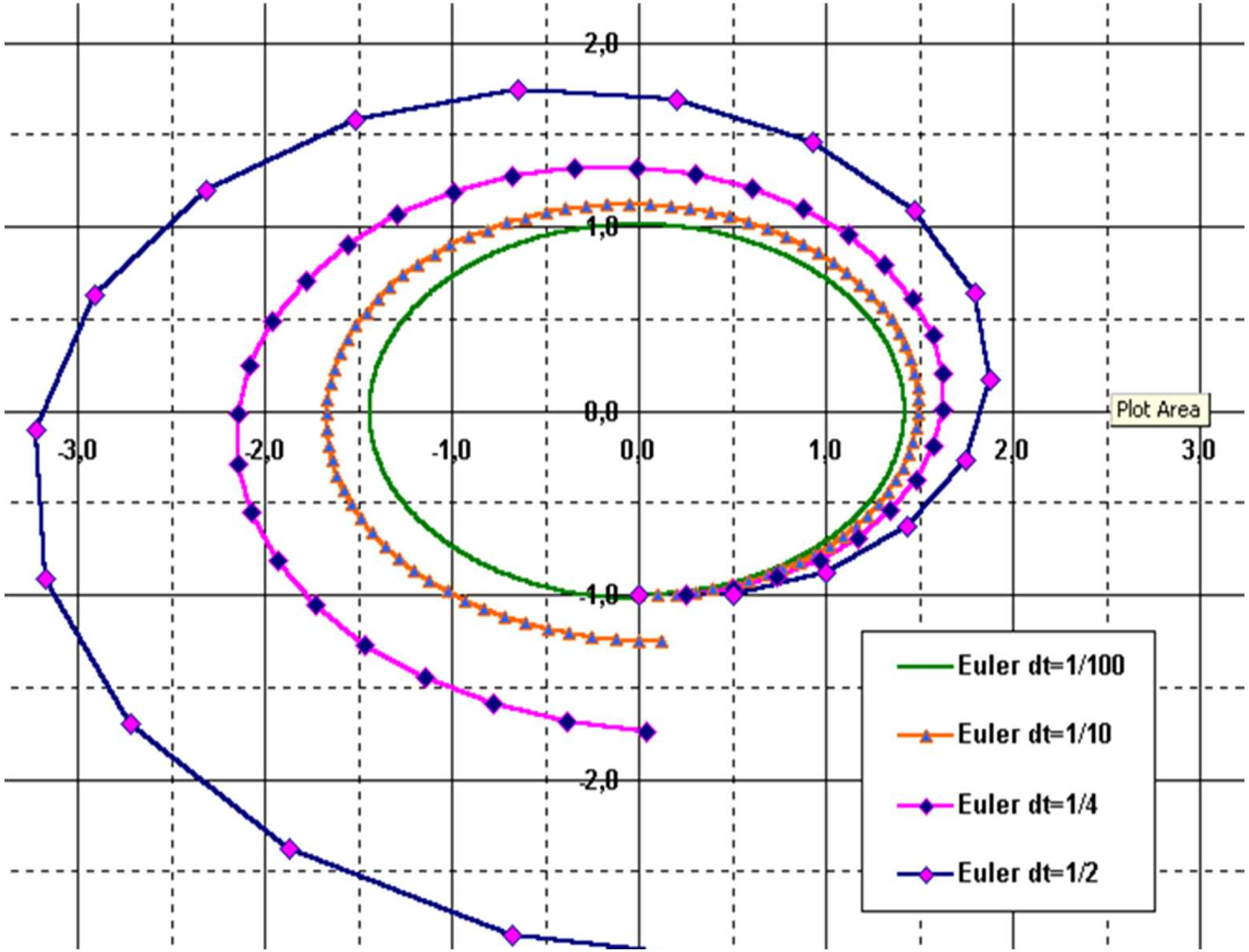
Euler Method

- Seeking to approximate integration of the velocity over time
- Euler method is the starting point for approximating this
- Problems?
  - Choice of step size is important
  - Choice of seed points are important

Euler Method

- Seeking to approximate integration of the velocity over time
- Euler method is the starting point for approximating this
- Problems?
  - Choice of step size is important
  - Choice of seed points are important
- Also remember that we have a field—we don't have measurements at every point (interpolation)

Euler Quality by Step Size



[via Levine]

Numerical Integration

- How do we generate accurate streamlines?
- Solving an ordinary differential equation

$$\frac{dL}{dt} = v(L(t)) \qquad L(0) = L_0$$

where L is the streamline, v is the vector field, and t is “time”

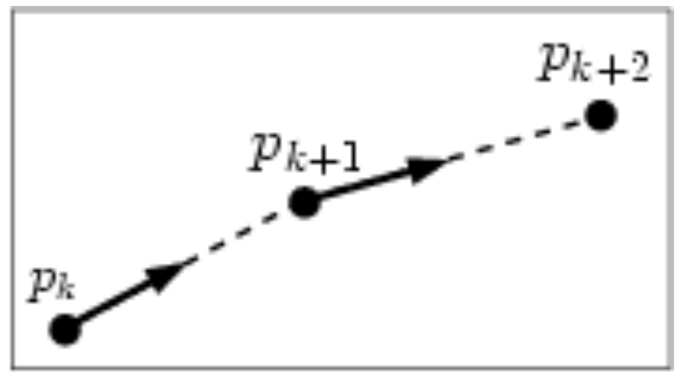
- Solution:

$$L(t + \Delta t) = L(t) + \boxed{\int_t^{t+\Delta t} v(L(t))dt}$$

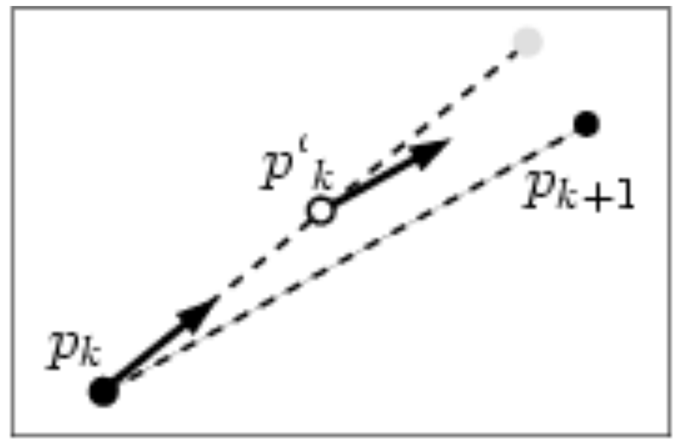
Higher-order methods

$$\int_t^{t+\Delta t} v(L(t))dt$$

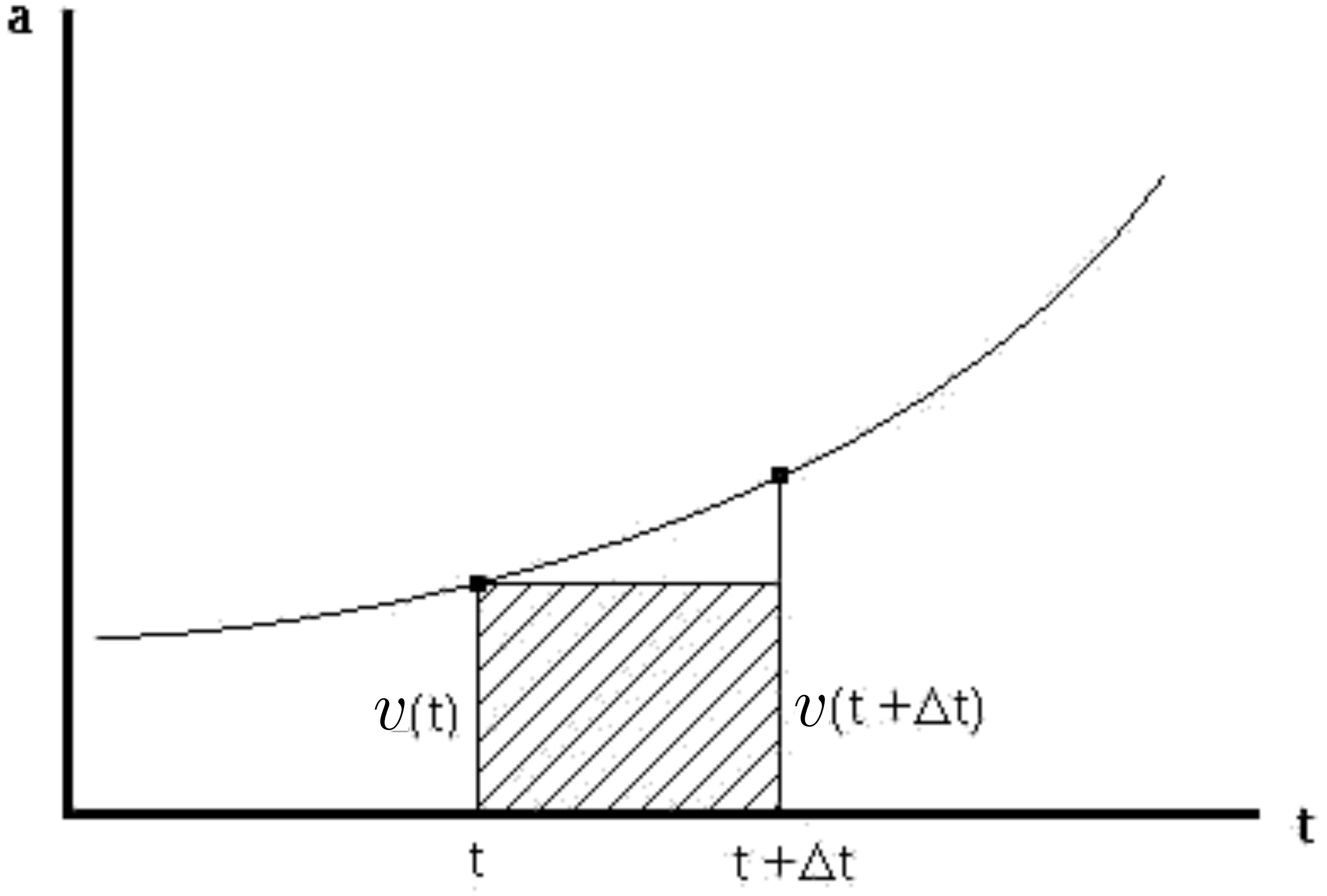
- Euler method (use single sample)



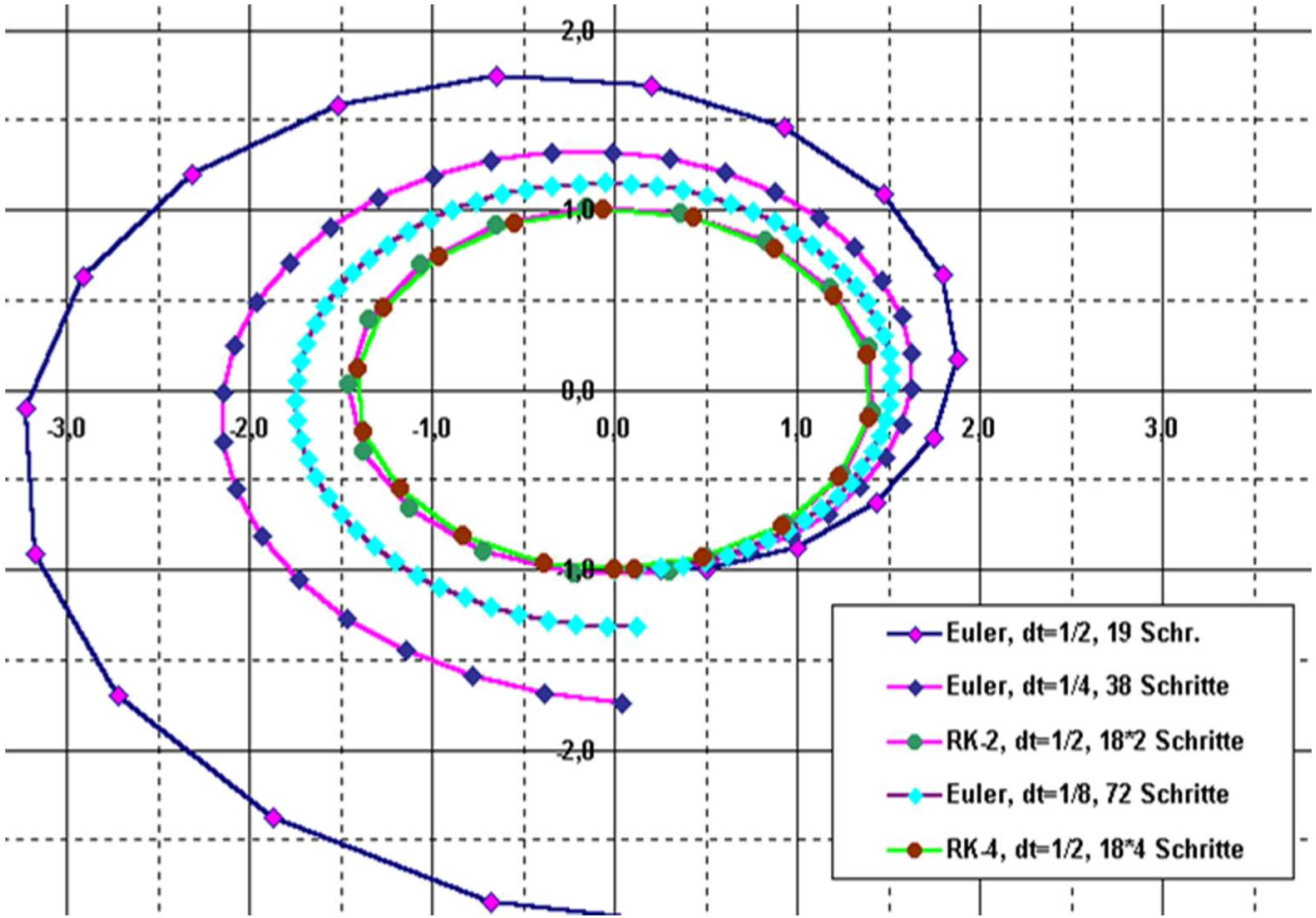
- Higher-order methods (Runge-Kutta) (use more samples)



[A. Mebarki]



Higher-Order Comparison



[via Levine]