

Diffusion Curves: A Vector Representation for Smooth-Shaded Images

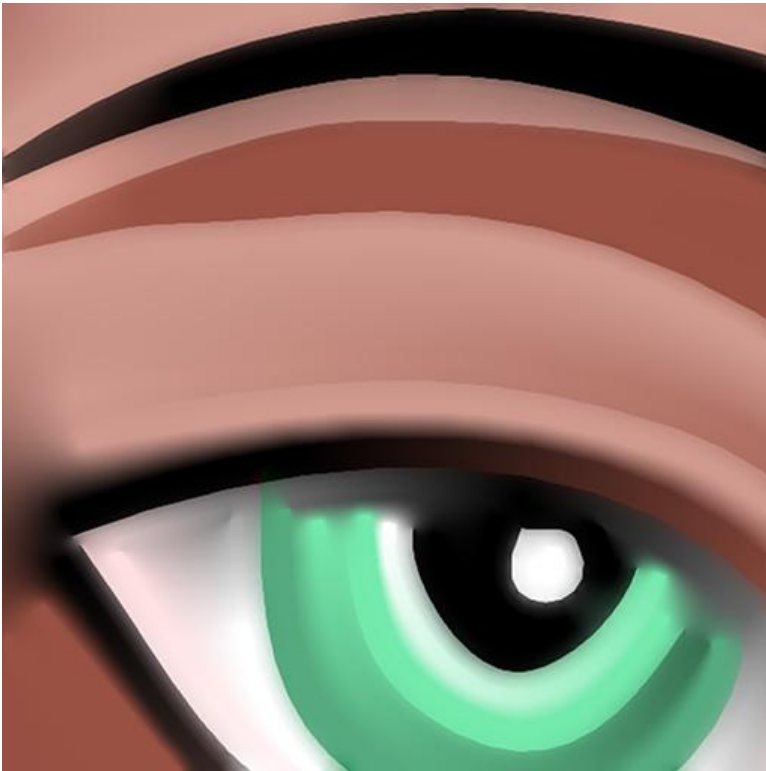
报告人：黄学真

学 号：11121004

Vector Graphics V.S. Raster Graphics



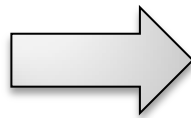
Vector Graphics V.S. Raster Graphics



- Compact representation and resolution-independence
- geometric editability and keyframe animation

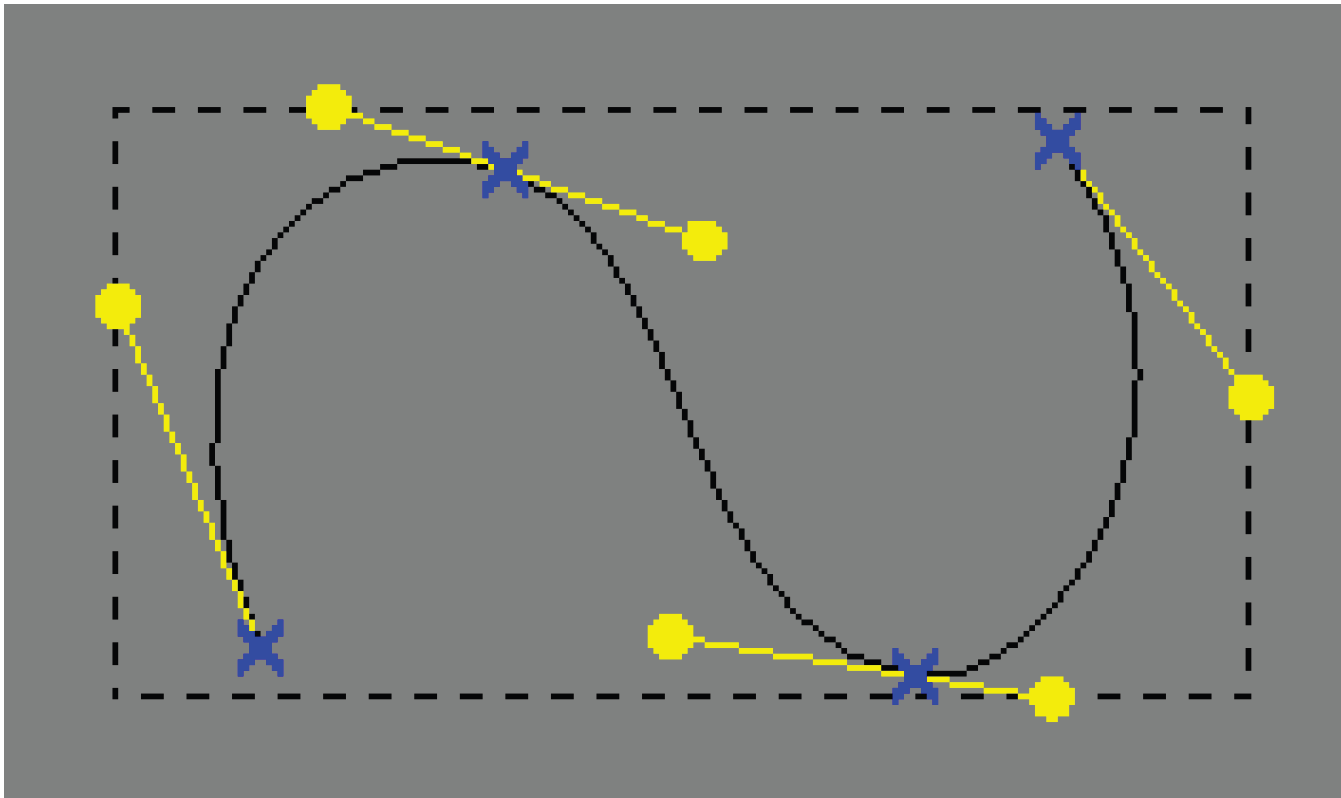
Diffusion Curves

- diffuses colors on both sides
- supports freehand drawing
- the same level of visual as complex as gradient meshes



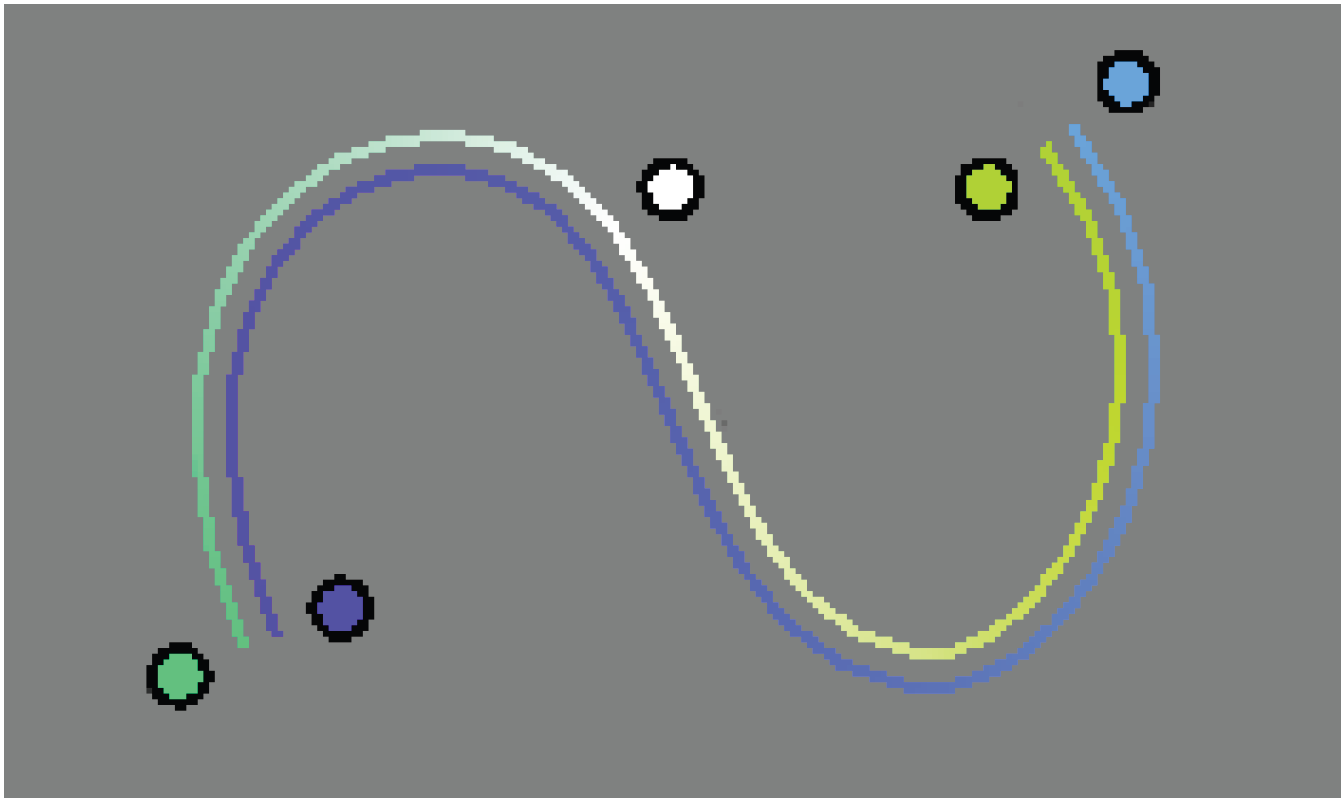
Diffusion Curves Model

- a geometric curve described by a Bezier spline



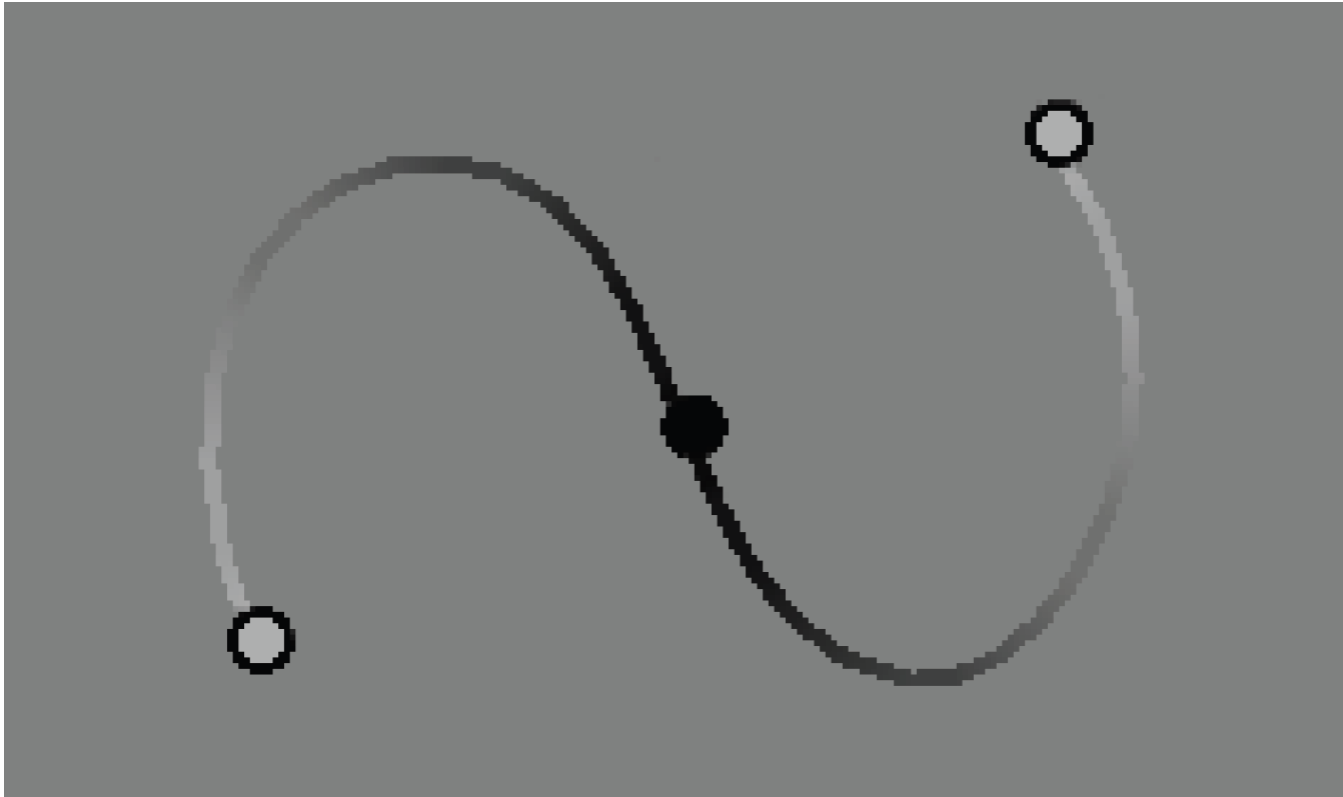
Diffusion Curves Model

- arbitrary colors on either side, linearly interpolated along the curve



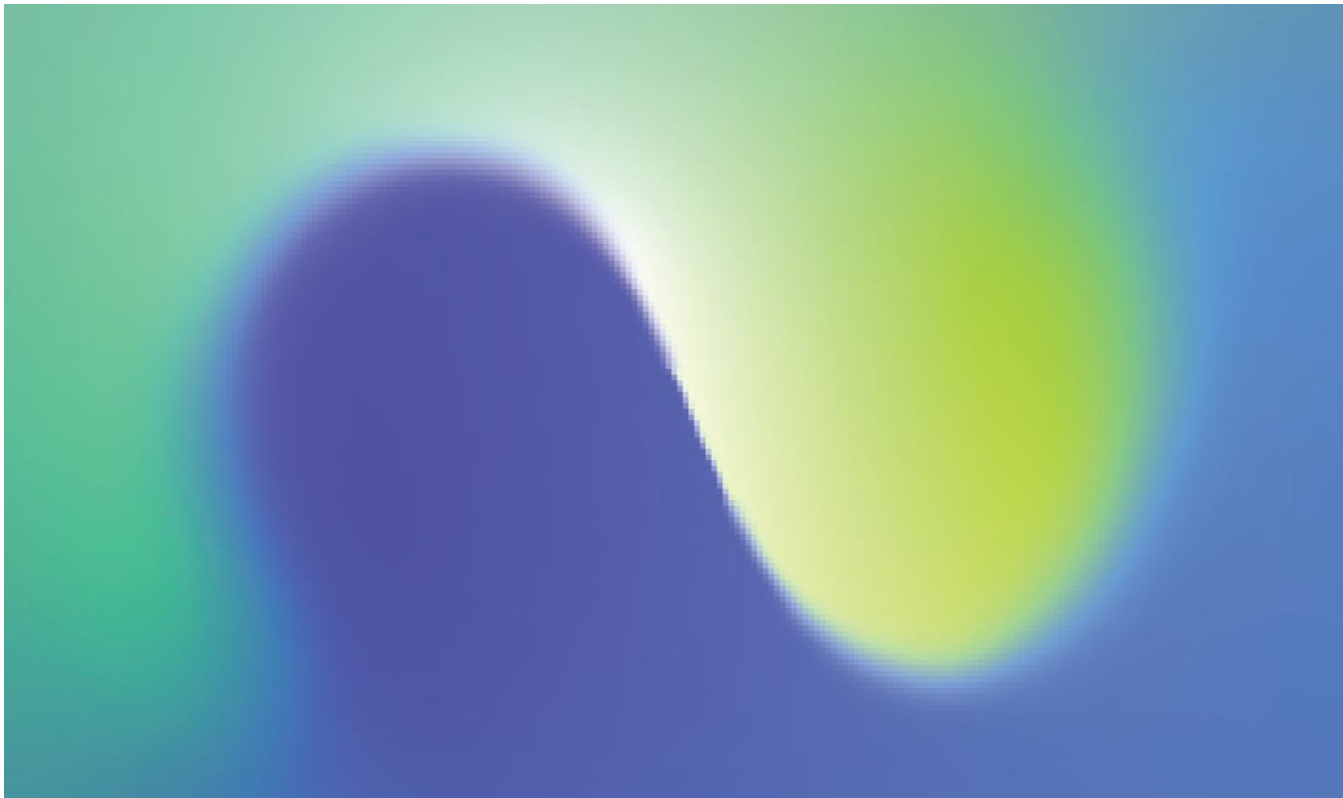
Diffusion Curves Model

- a blur amount linearly interpolated along the curve



Diffusion Curves Model

- The final image is obtained by diffusion and reblurring.




Diffusion Curves Solver

- Curves diffuse color similarly to heat diffusion
- Express diffusion as the solution to a Poisson equation
- add color gradient constraint to maintains the sharp color transition

$$\Delta I = \operatorname{div} \mathbf{w}$$

$$I(x, y) = C(x, y)$$

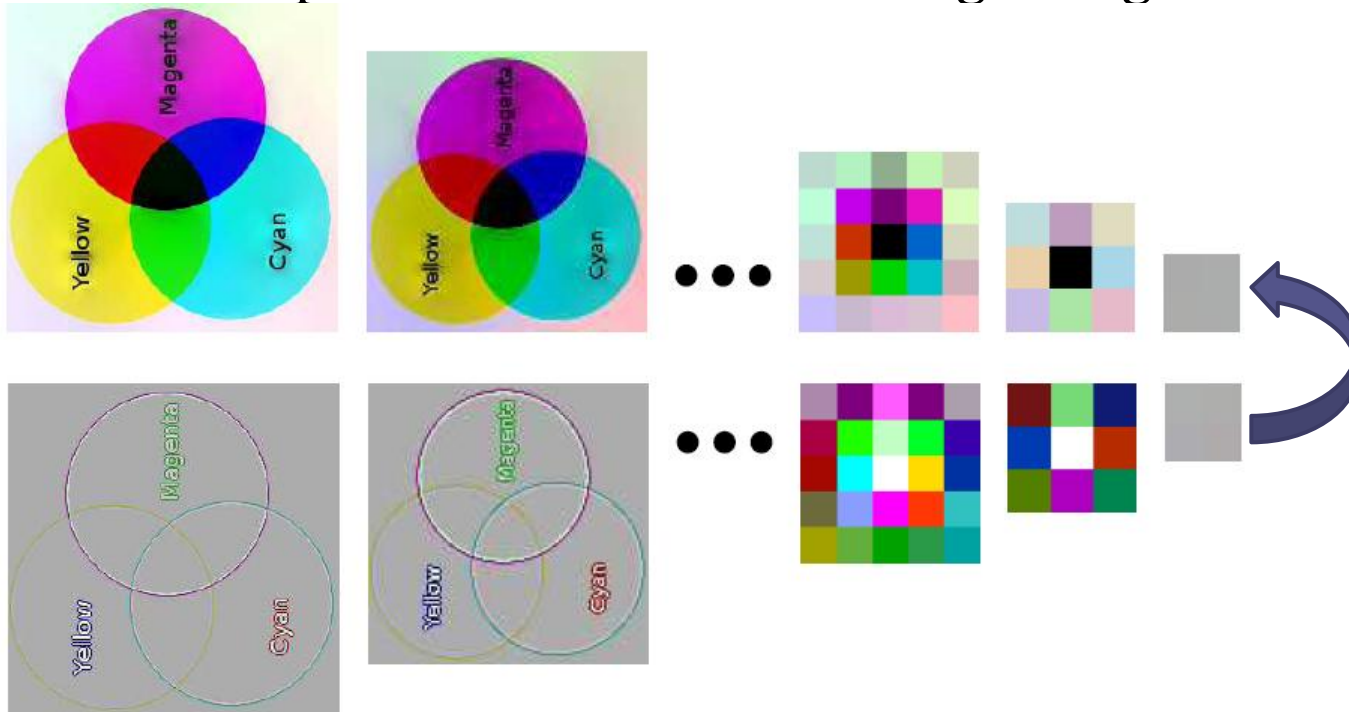
Discrete Poisson equation


$$\Delta I = \operatorname{div} \mathbf{w}$$
$$I_{x,y} = \frac{[I_{x-1,y} + I_{x+1,y} + I_{x,y-1} + I_{x,y+1} + \operatorname{div}(w)_{x,y}]}{4}$$

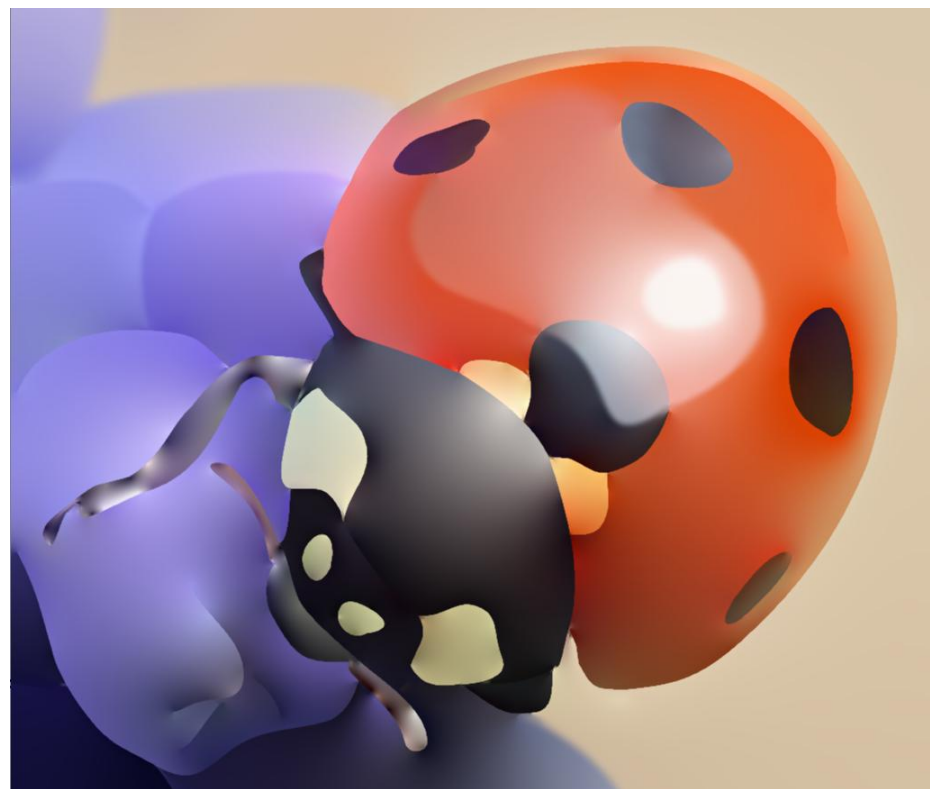
- requires solving a large, sparse, linear system,
- very time consuming if implemented naively.

Multigrid Acceleration

- Downsample to efficiently reach a coarse solution
- Upsample and refine the solution
- Use Jacobi relaxations in each level
- a GPU implementation of the multigrid algorithm



Results1 (Extracting colors)

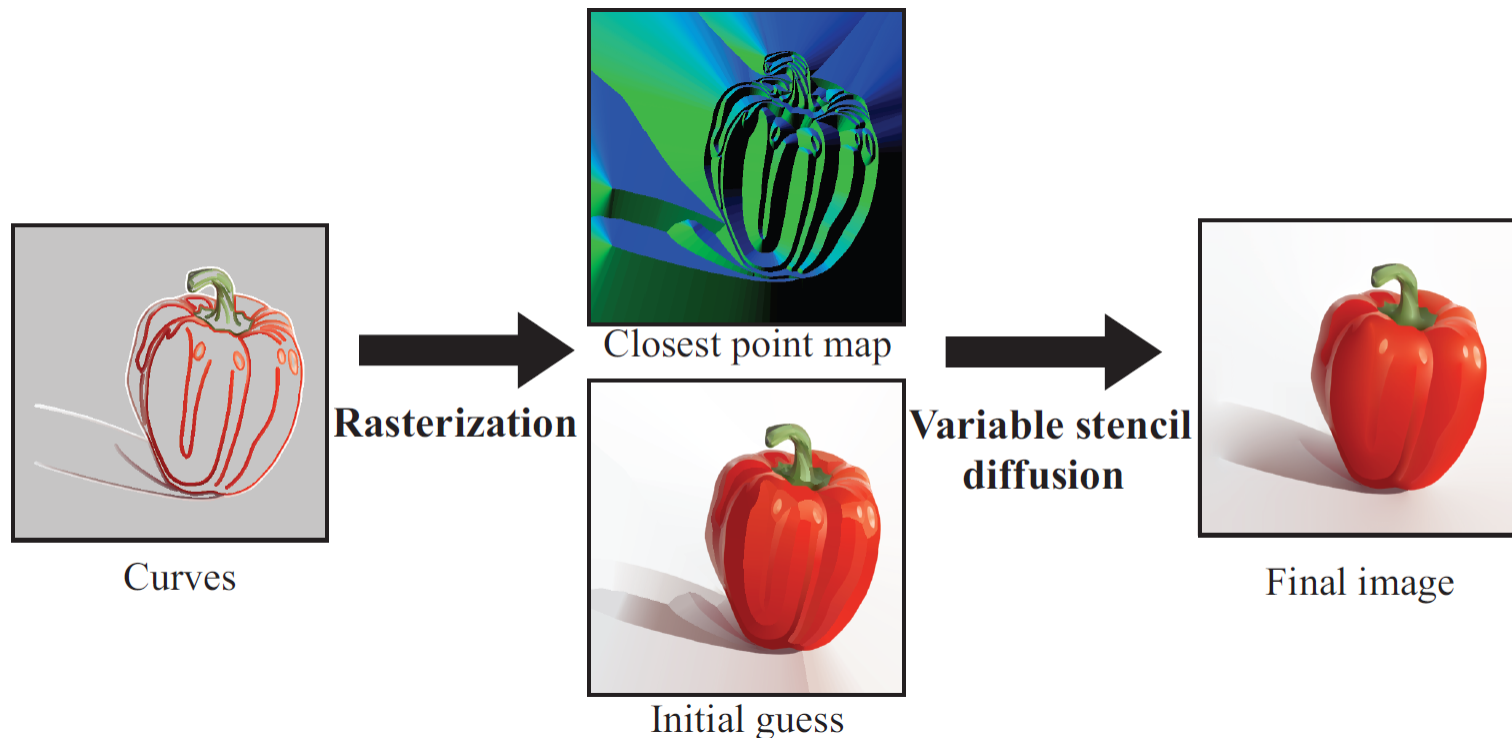


Results2(Automatic extraction)



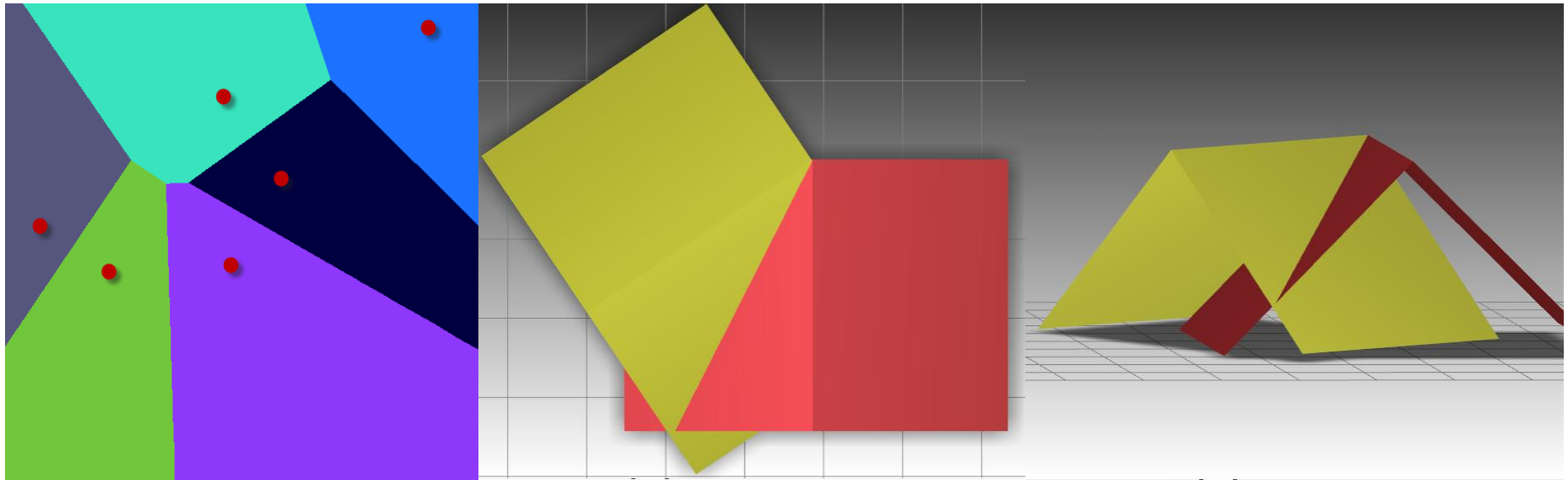
A New Solver

- Create a Voronoi diagram to get a initial guess
- Use variable stencil size diffusion solver



Initial Guess

- creates a Voronoi diagram and initialize pixels to the color of the closest curve point
- For each segment we construct a *tent* as two quads



Variable stencil size diffusion solver

$$\Delta I = \text{div } \mathbf{w} \quad \text{h-neighbor}$$
$$I_{x,y} = \frac{[I_{x-h,y} + I_{x+h,y} + I_{x,y-h} + I_{x,y+h} + \text{div}(\mathbf{w})_{x,y}]}{4}$$



Initial guess



1 step



2 steps



8 steps

- the best choice for h is the distance to the closest boundary curve
- shrink the stencil radius during the diffusion

The New Solver Result



Voronoi Diagram



Final Image(only 8 iterations)

Conclusion

- introduced diffusion curves as a new image representation
- Poisson equation multigrid solver
- A variable stencil size diffusion solver



Thank you!