Image Editing
Poisson Reconstruction

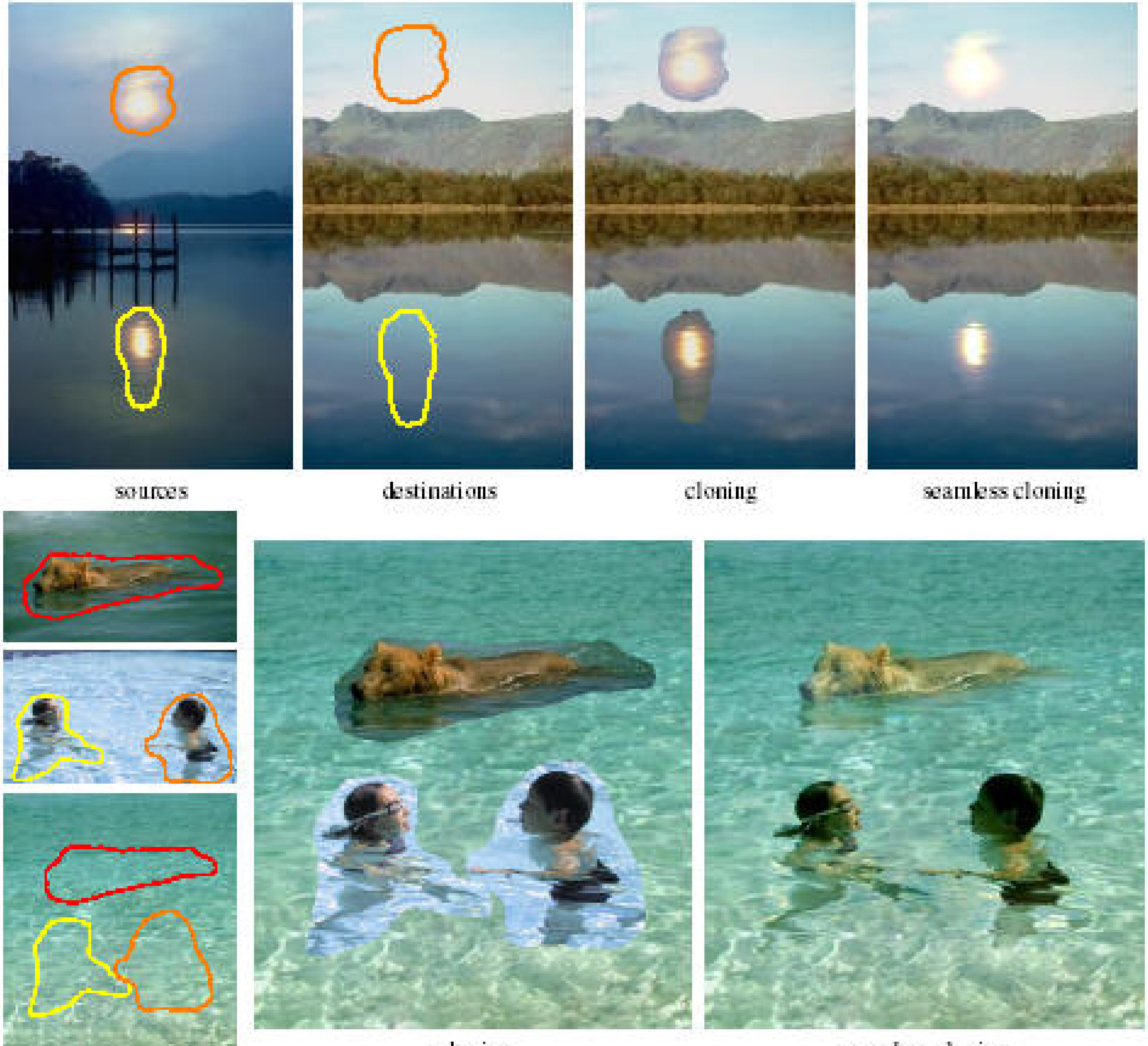
CVFX @ NTHU

7 May 2015

Poisson Reconstruction

- *Poisson Image Editing*
  - Patrick Perez, Michel Gangnet, and Andrew Blake
  - SIGGRAPH 2003

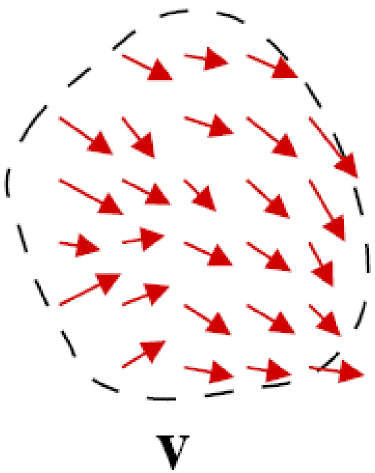
- *Drag and Drop Pasting*
  - Jia *et al*., SIGGRAPH 2006

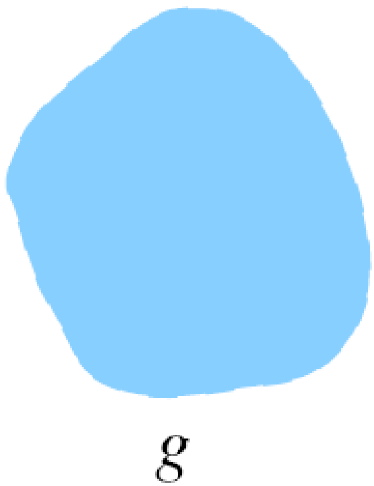


sources destinations cloning seamless cloning

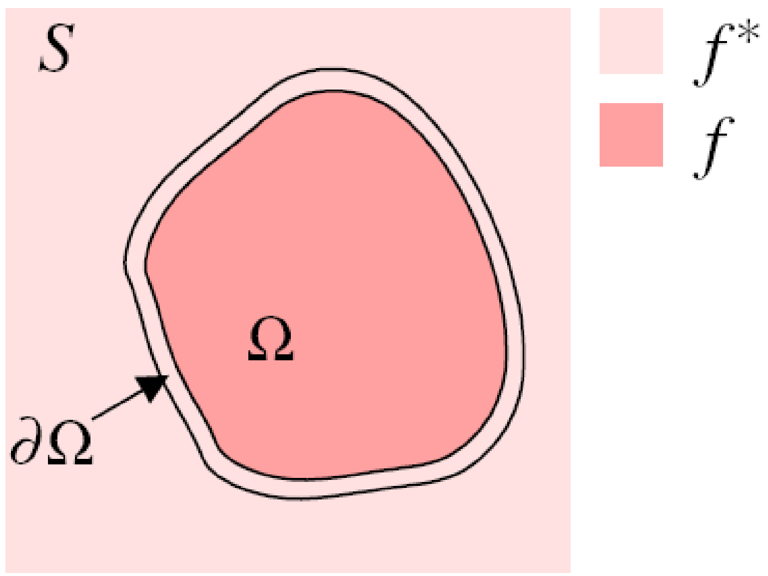
sources/destinations cloning seamless cloning

Guided Interpolation





under the guidance of vector field $\mathbf{v}$



interpolating the unknown scalar function f

Simple Interpolation

- Smoothness assumption

$$\min_f \int\int_\Omega \|\nabla f\|^2 \, dx \, dy \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]^T$$

Optimization

- Minimize the functional

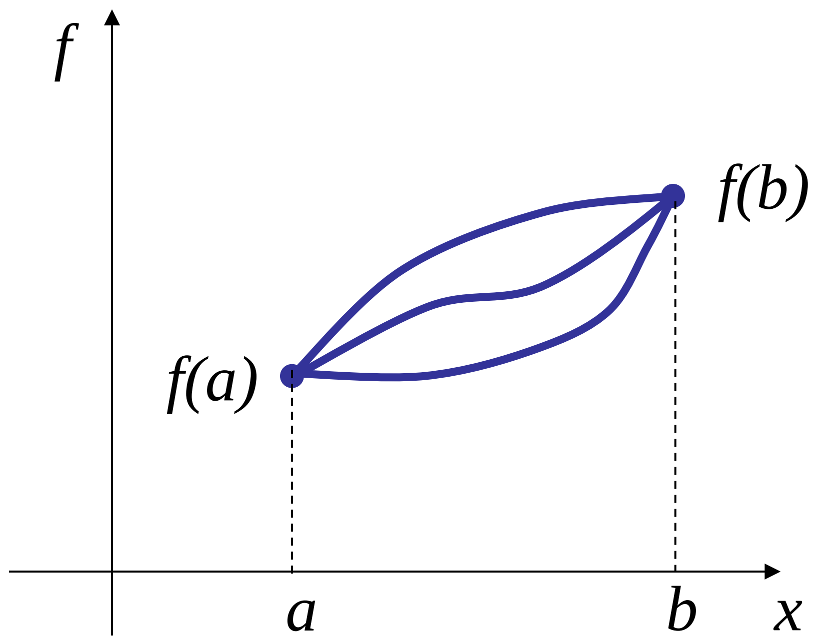
$$\int\int_{\Omega} F(\nabla f)\, dx\, dy$$

where $F(\nabla f) = \|\nabla f\|^2 = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2$

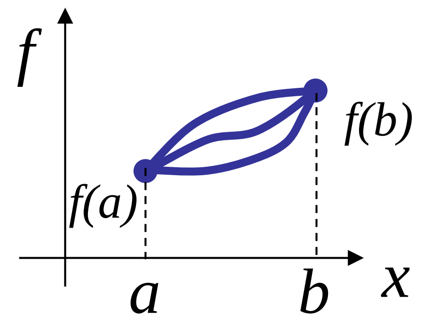
Calculus of Variations

Minimize a functional $I[f(x)] = \int_a^b F(f, \frac{df}{dx}, x)\, dx$

$$\min_f \int_a^b F(f, \frac{df}{dx}, x)\, dx$$



Minimize a functional $I[f(x)] = \int_a^b F(f, \frac{df}{dx}, x)\, dx$



$\eta(a) = \eta(b) = 0$

$f(x) \rightarrow f(x) + \alpha\eta(x)$

α is small and $\eta(x)$ arbitrary

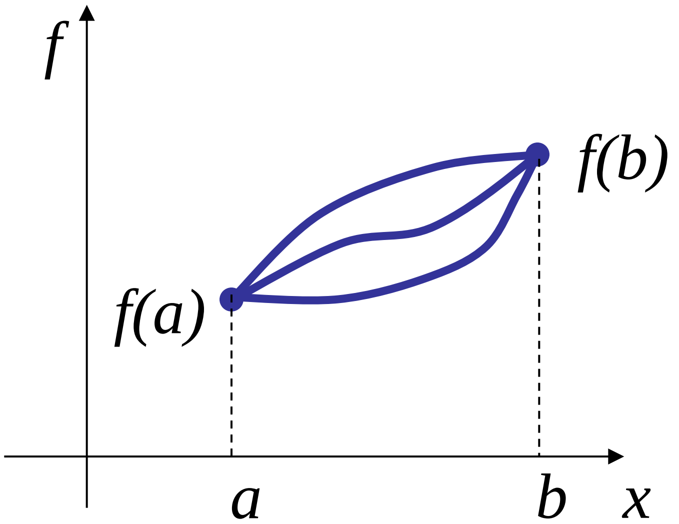
if the functional is to be stationary, then
we must have $\frac{dI}{d\alpha}|_{\alpha=0} = 0$ for all $\eta(x)$

$$I(\alpha) = \int_a^b F(f + \alpha\eta, f' + \alpha\eta', x) dx$$

$$= I(0) + \alpha \underbrace{\int_a^b \left(\frac{\partial F}{\partial f}\eta + \frac{\partial F}{\partial f'}\eta' \right) dx}_{=0} + O(\alpha^2)$$

$$
\begin{aligned}
0 &= \int_a^b \left(\frac{\partial F}{\partial f}\eta + \frac{\partial F}{\partial f'}\eta' \right) dx \\
&= \frac{\partial F}{\partial f'}\eta \bigg|_a^b + \int_a^b \left(\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f'} \right) \eta \, dx
\end{aligned}
$$

$$
\Rightarrow \boxed{\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f'} = 0}
$$



$$\eta(a) = \eta(b) = 0$$

integration by part $\displaystyle \int_a^b \frac{\partial F}{\partial f'}\eta' \, dx = \frac{\partial F}{\partial f'}\eta \bigg|_a^b - \int_a^b \frac{d}{dx}\frac{\partial F}{\partial f'}\eta \, dx$

Euler-Lagrange Equation

- Minimize the functional

$$\int\int_{\Omega} F(\nabla f)\, dx\, dy$$

where $F(\nabla f) = \|\nabla f\|^2 = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2$

The solution *f* satisfies

$$\boxed{\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f_x} - \frac{d}{dy}\frac{\partial F}{\partial f_y} = 0}$$

$$\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f_x} - \frac{d}{dy}\frac{\partial F}{\partial f_y} = 0$$

$$F(\nabla f) = \|\nabla f\|^2 = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 = f_x^2 + f_y^2$$

$$f_x = \frac{\partial f}{\partial x}$$

$$f_y = \frac{\partial f}{\partial y}$$

$$\frac{\partial F}{\partial f} = 0 \qquad \frac{\partial F}{\partial f_x} = 2f_x \qquad \frac{\partial F}{\partial f_y} = 2f_y$$

$$-2\frac{d}{dx}f_x - 2\frac{d}{dy}f_y = 0$$

$$\frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2} = 0$$

Example: Let's Consider a 1D Case

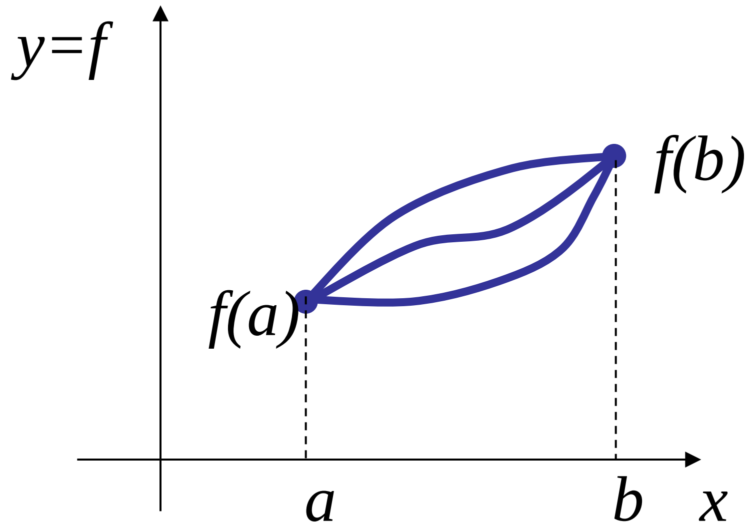
- Minimize the functional

$$\int\int_{\Omega} F(y, y', x)\, dx$$

- where $F(y, y', x) = F(y') = \left(y'\right)^2 + 1$

$$\boxed{\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f'} = 0}$$

What does the solution *f* look like?

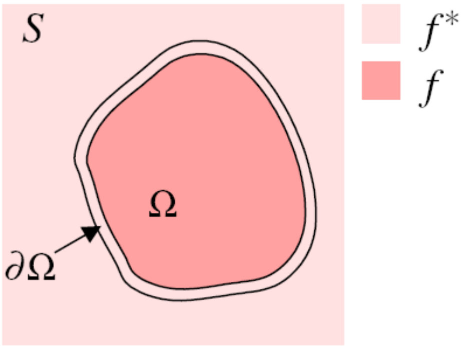


Simple Interpolation

- Smoothness assumption

$$\min_f \int\int_\Omega \|\nabla f\|^2 \, dx\, dy \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]^T$$



$$\frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2} = 0 \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$$\Downarrow$$

$$\Delta f = 0 \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

Dirichlet boundary condition

Guidance Field

$$\min_f \int\int_\Omega \|\nabla f - \mathbf{v}\|^2 \, dx \, dy \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$$\Delta f = \operatorname{div} \mathbf{v} \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$$\mathbf{v} = [u, v]^T \qquad \operatorname{div} \mathbf{v} = \nabla \cdot \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$

$$\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f_x} - \frac{d}{dy}\frac{\partial F}{\partial f_y} = 0$$

$$F(\nabla f) = \|\nabla f - \mathbf{v}\|^2 = \left(\frac{\partial f}{\partial x} - u\right)^2 + \left(\frac{\partial f}{\partial y} - v\right)^2$$
$$= (f_x - u)^2 + (f_y - v)^2$$

$$f_x = \frac{\partial f}{\partial x}$$

$$f_y = \frac{\partial f}{\partial y}$$

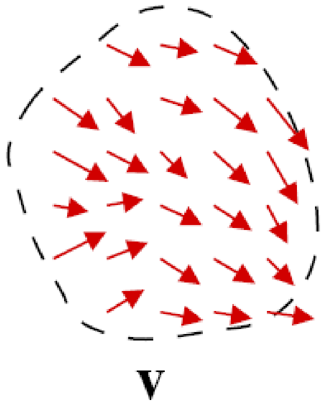
$$\frac{\partial F}{\partial f} = 0 \qquad \frac{\partial F}{\partial f_x} = 2(f_x - u) \qquad \frac{\partial F}{\partial f_y} = 2(f_y - v)$$

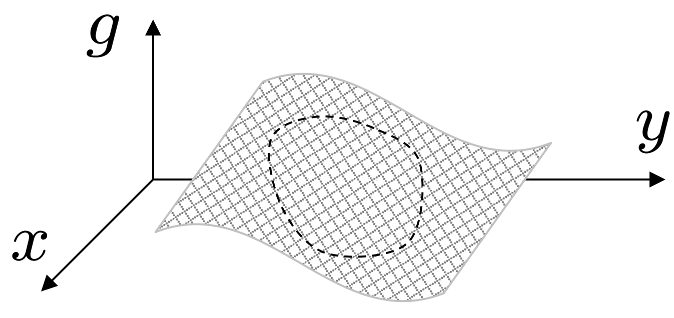
$$-2\frac{d}{dx}(f_x - u) - 2\frac{d}{dy}(f_y - v) = 0$$

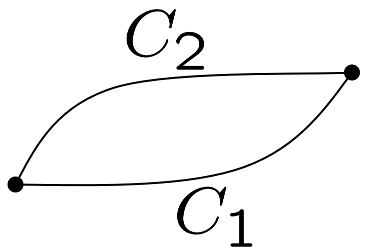
$$\frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \quad \Rightarrow \quad \Delta f = \operatorname{div} \mathbf{v}$$

Conservative Guidance Field

$$\mathbf{v} = \nabla g$$







$$\oint_C \mathbf{v} \cdot ds = 0$$

$$\int_{C_1} \mathbf{v} \cdot ds = \int_{C_2} \mathbf{v} \cdot ds$$

When the Guidance Field Is Conservative

$$\mathbf{v} = \nabla g$$

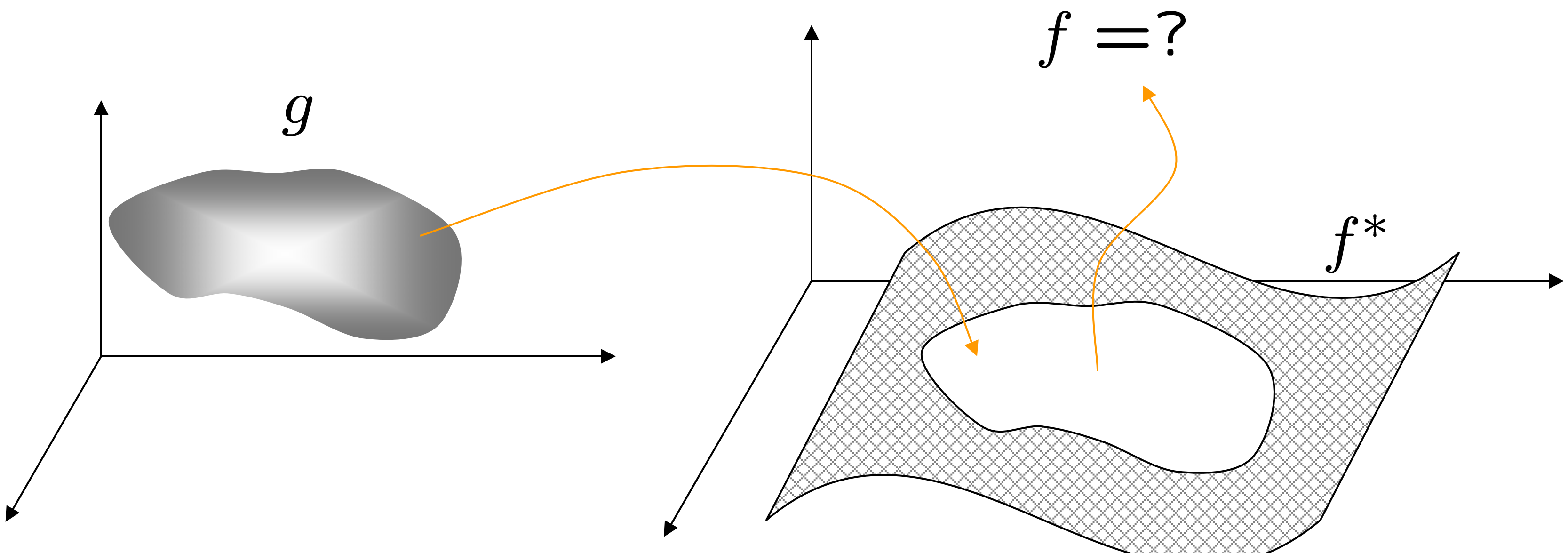
$f = g + \tilde{f}$ $\qquad$ $\tilde{f}$ is the correction funciton

$$\Delta f = \operatorname{div} \mathbf{v} \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$$\Delta f = \nabla \cdot \nabla g \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

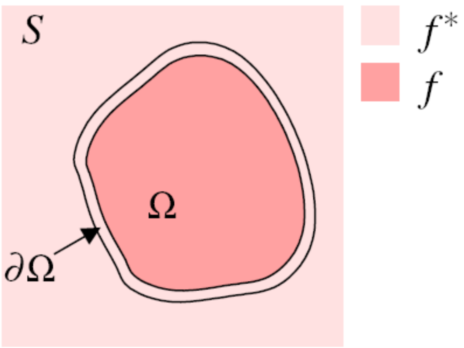
$$\Delta(g + \tilde{f}) = \Delta g \text{ over } \Omega \text{ with } (g + \tilde{f})|_{\partial\Omega} = f^*|_{\partial\Omega}$$

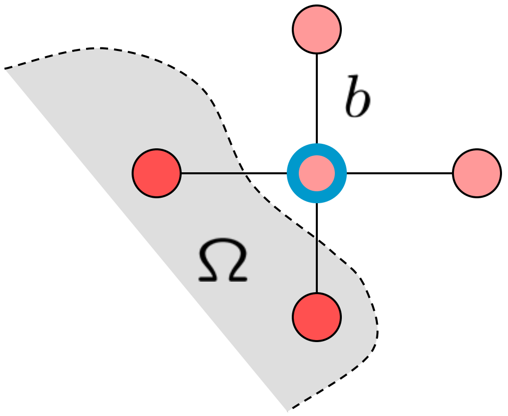
$$\Delta \tilde{f} = 0 \text{ over } \Omega \text{ with } \tilde{f}|_{\partial\Omega} = (f^* - g)|_{\partial\Omega}$$


g
$f = ?$
f^*

Discrete Poisson Equation

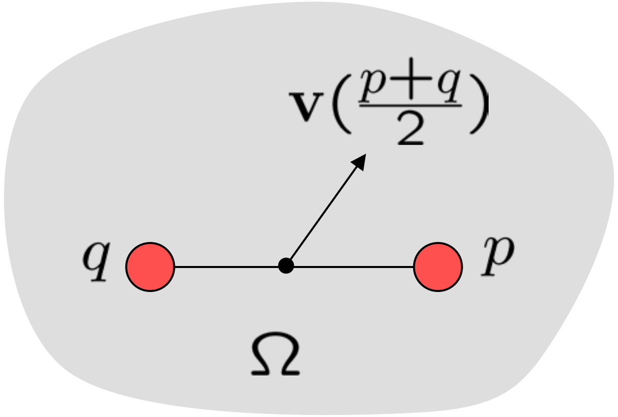
$$\partial\Omega = \{b \in S \setminus \Omega : N_b \cap \Omega \neq \emptyset\}$$





$$\min_{\{f_p, p\in\Omega\}} \sum_{(p,q)\cap\Omega\neq\emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_b = f_b^*, \text{ for all } b \in \partial\Omega$$

$$v_{pq} = \mathbf{v}(\tfrac{p+q}{2}) \cdot \vec{pq}$$



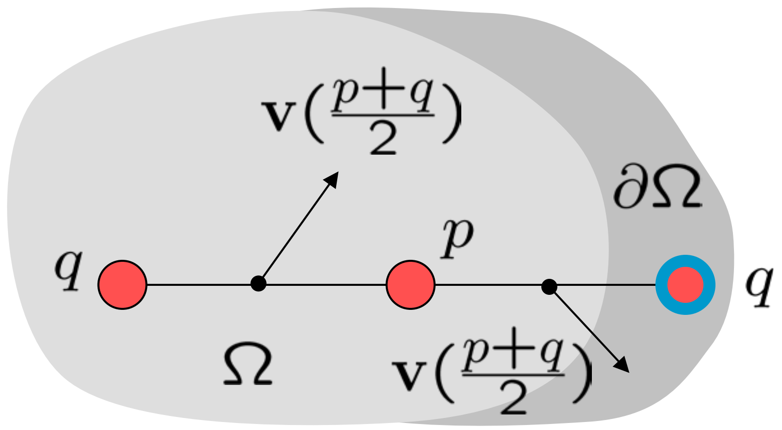
Discrete Poisson Solver

$$\min_{\{f_p, p\in\Omega\}} \sum_{(p,q)\cap\Omega\neq\emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_b = f_b^*, \text{ for all } b \in \partial\Omega$$

for each p: $$2 \sum_{(p,q)\cap\Omega\neq\emptyset} (f_p - f_q - v_{pq}) = 0 \qquad f_q = f_q^* \text{ if } q \in \partial\Omega$$

for all $p \in \Omega$,

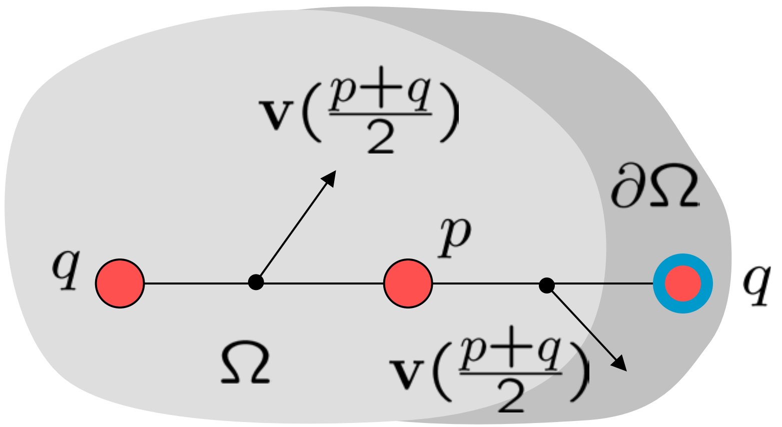
$$|N_p| f_p - \sum_{q\in N_p\cap\Omega} f_q - \sum_{q\in N_p\cap\partial\Omega} f_q^* - \sum_{q\in N_p} v_{pq} = 0$$



Discrete Poisson Solver

for all $p \in \Omega$, $|N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}$

sparse (banded), symmetric linear system



$$|N_p| f_p - \sum_{q \in N_p} f_q = \sum_{q \in N_p} v_{pq}$$

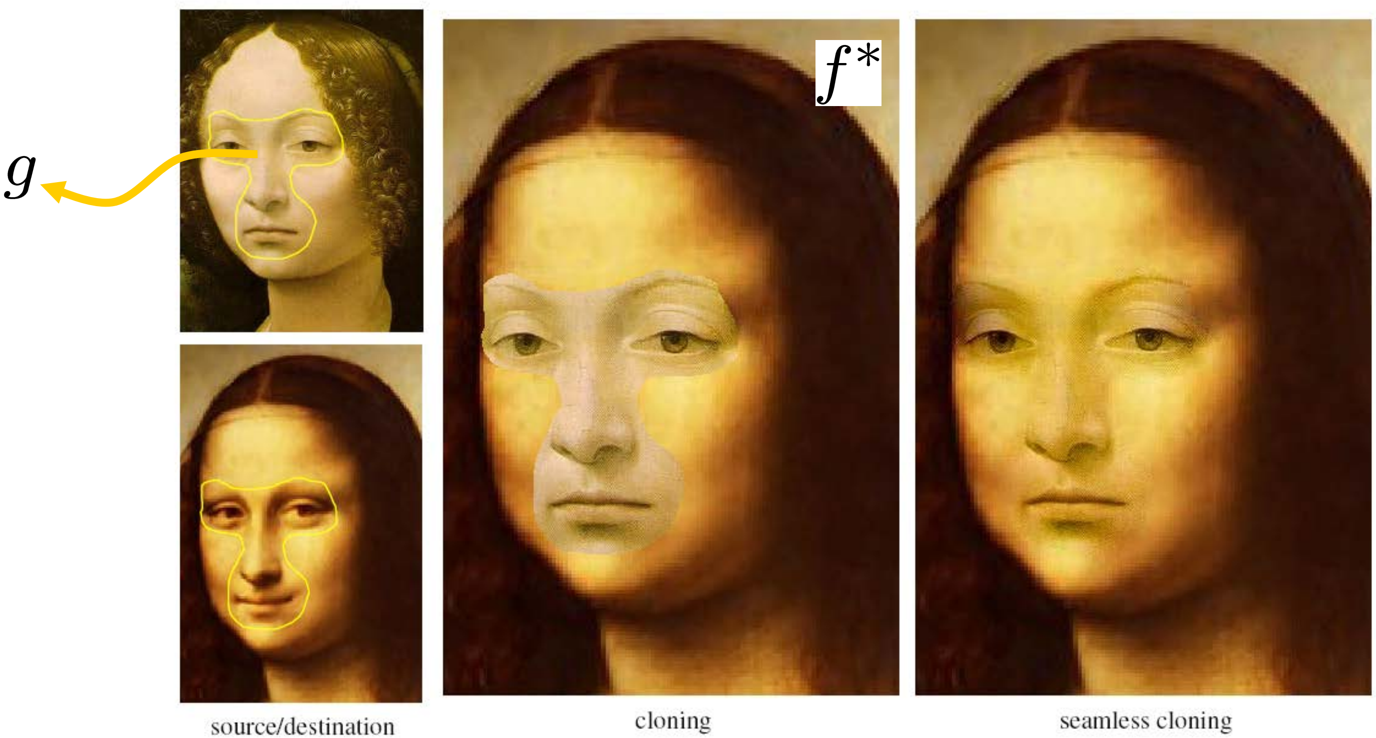
Seamless Cloning

- Importing gradients

  $\mathbf{v} = \nabla g$

  $\Delta f = \Delta g$ over Ω with $f|_{\partial\Omega} = f^*|_{\partial\Omega}$

  for all pairs (p, q), $v_{pq} = g_p - g_q$ (finite difference)



Mixing Gradients

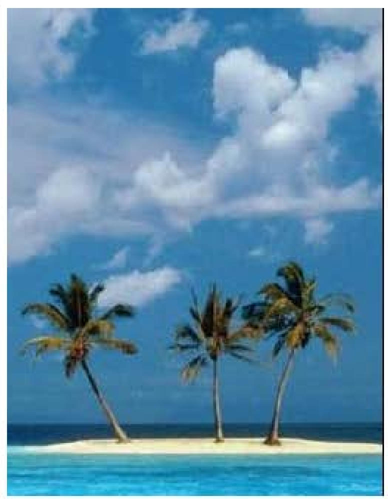
$$\text{for all } \mathbf{x} \in \Omega,\ \mathbf{v}(\mathbf{x}) = \begin{cases} \nabla f^*(\mathbf{x}) & \text{if } |\nabla f^*(\mathbf{x})| > |\nabla g(\mathbf{x})|, \\ \nabla g(\mathbf{x}) & \text{otherwise.} \end{cases}$$

$$v_{pq} = \begin{cases} f_p^* - f_q^* & \text{if } |f_p^* - f_q^*| > |g_p - g_q|, \\ g_p - g_q & \text{otherwise,} \end{cases}$$

non-conservative

source

g



destination

f^*

Texture Flattening

for all $\mathbf{x} \in \Omega$, $\mathbf{v}(\mathbf{x}) = M(\mathbf{x}) \nabla f^*(\mathbf{x})$ binary mask/edge detector

$$v_{pq} = \begin{cases} f_p - f_q & \text{if an edge lies between } p \text{ and } q, \\ 0 & \text{otherwise,} \end{cases}$$

Local Illumination Change

$$\mathbf{v} = \alpha^{\beta} |\nabla f^*|^{-\beta} \nabla f^*$$

$$\beta = 0.2.$$

$$\mathbf{v} = \left(\frac{0.2 \langle \nabla f^* \rangle}{|\nabla f^*|} \right)^{0.2} \nabla f^*$$

Local Color Change

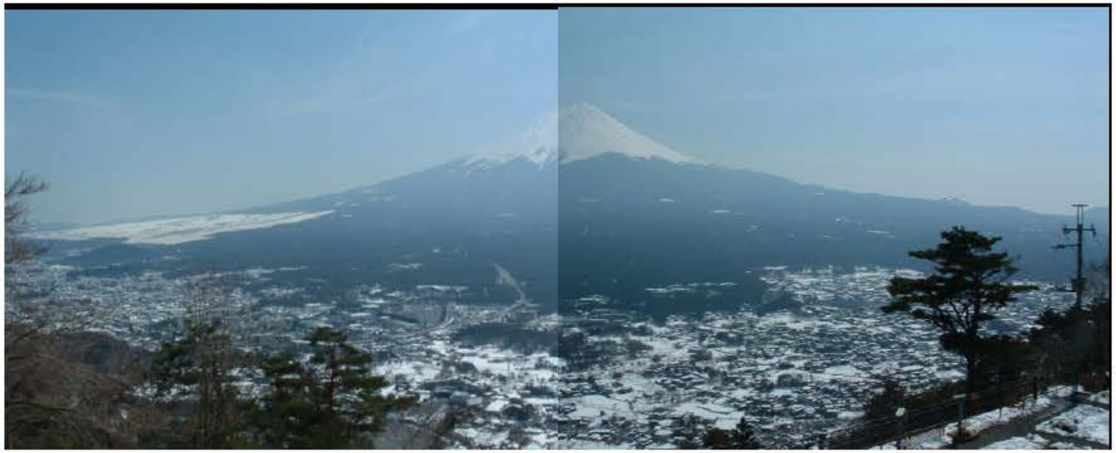
background
de-colorization

re-coloring

Image Stitching

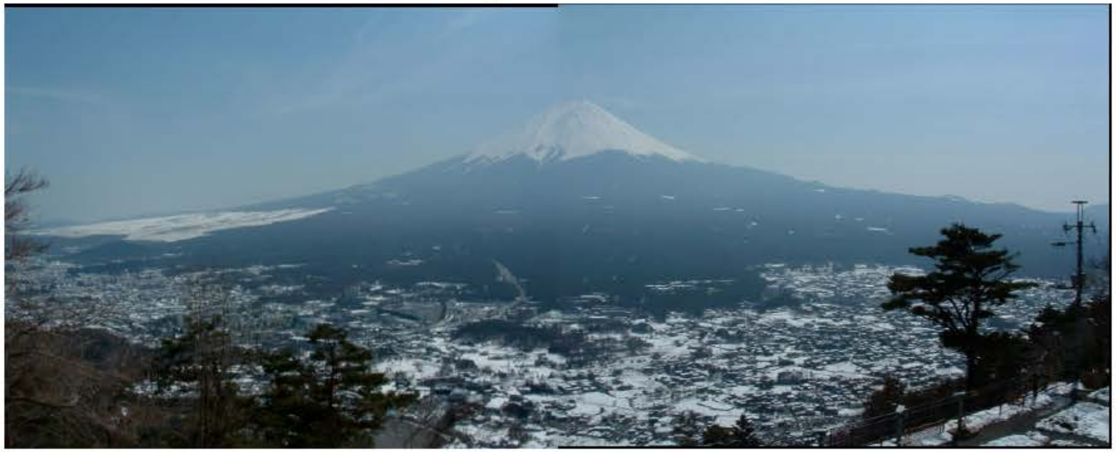
- Levin *et al.*, ECCV 2004

Input image 1_1



Pasting of I_1 and I_2

Input image I_2



Stitching result

Alpha Interpolation

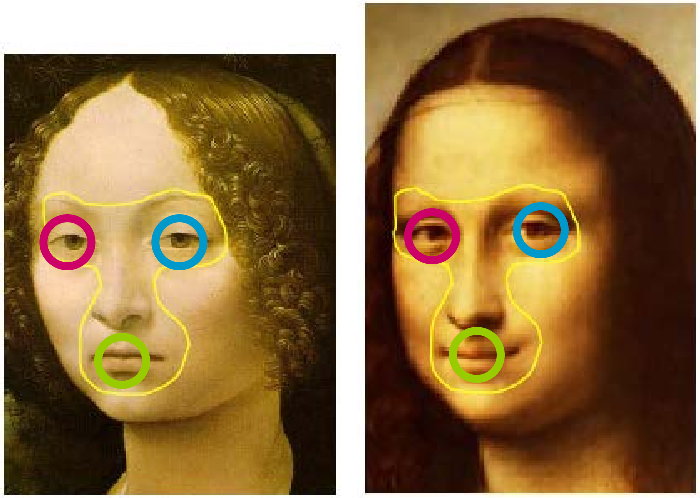
$$\forall x \in \Omega, v(x) = \begin{cases} \nabla f^*(x) & \text{if } \|\nabla f^*(x)\| > \alpha \|\nabla g(x)\| \\ \alpha \nabla g(x) & \text{otherwise} \end{cases}$$

Leventhal *et al.*

Discussion

- Fast enough for interactive editing
  - 0.4s for a region of 60,000 pixels
  - Gauss-Seidel method
- Arbitrary shape

- Automatic alignment?
- Automatic deformation?

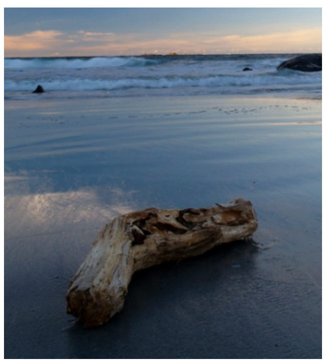


Poisson Reconstruction

- *Poisson Image Editing*
  - Patrick Perez, Michel Gangnet, and Andrew Blake
  - SIGGRAPH 2003
- *Drag and Drop Pasting*
  - Jia *et al*., SIGGRAPH 2006
    - » http://www.cse.cuhk.edu.hk/~leojia/all_project_webpages/ddp/drag-and-drop_pasting.html
  - Slides created by Jia *et al.*
    - » *http://www.cse.cuhk.edu.hk/~leojia/all_project_webpages/ddp/ddp_v3.ppt*

Poisson Equations in Images

- A case study

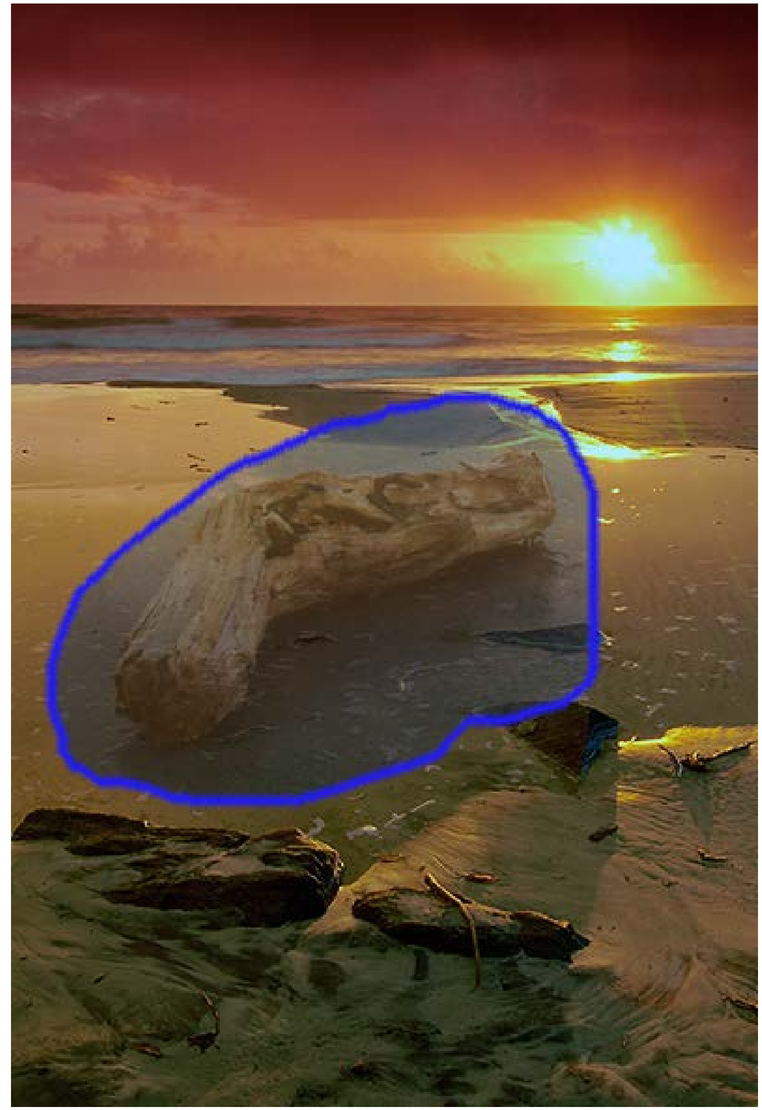


f_s

+

f_t

$\longrightarrow$



Slides of Jia *et al.*

Poisson Equations in Images

- A case study

f_s

f_t

Slides of Jia *et al.*

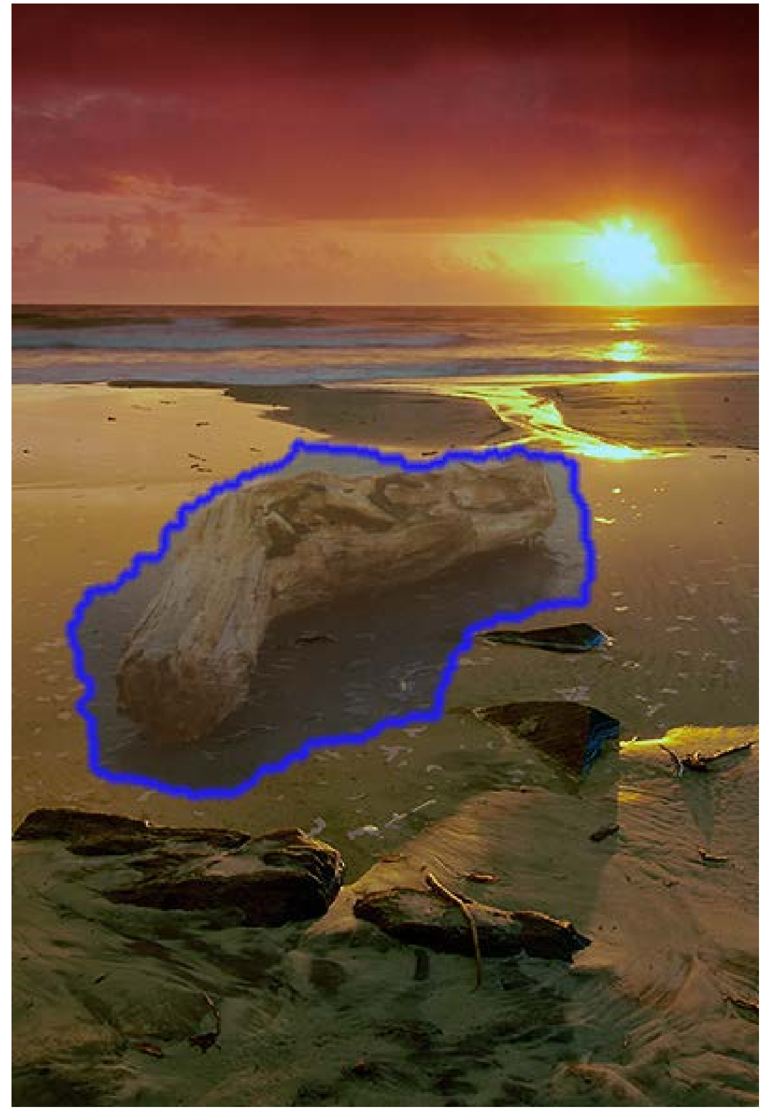
Poisson Equations in Images

- A case study

f_s

+

f_t



Slides of Jia *et al.*

Poisson Equations in Images

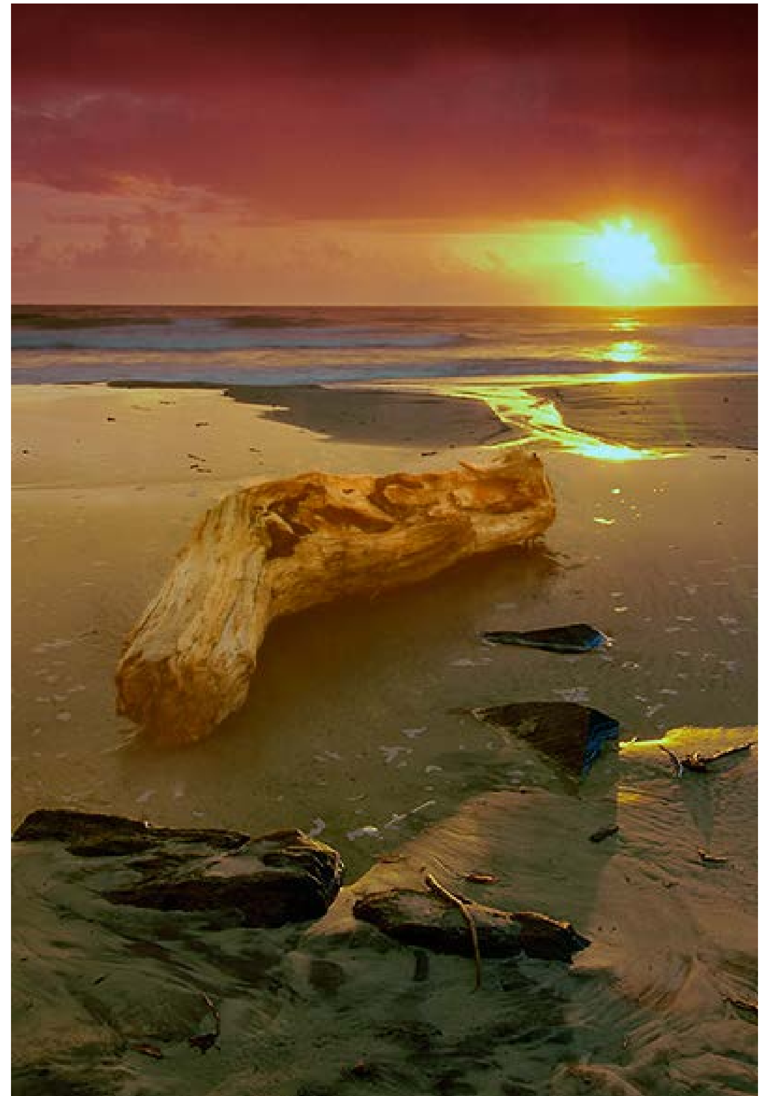
- A case study

f_s

+

f_t

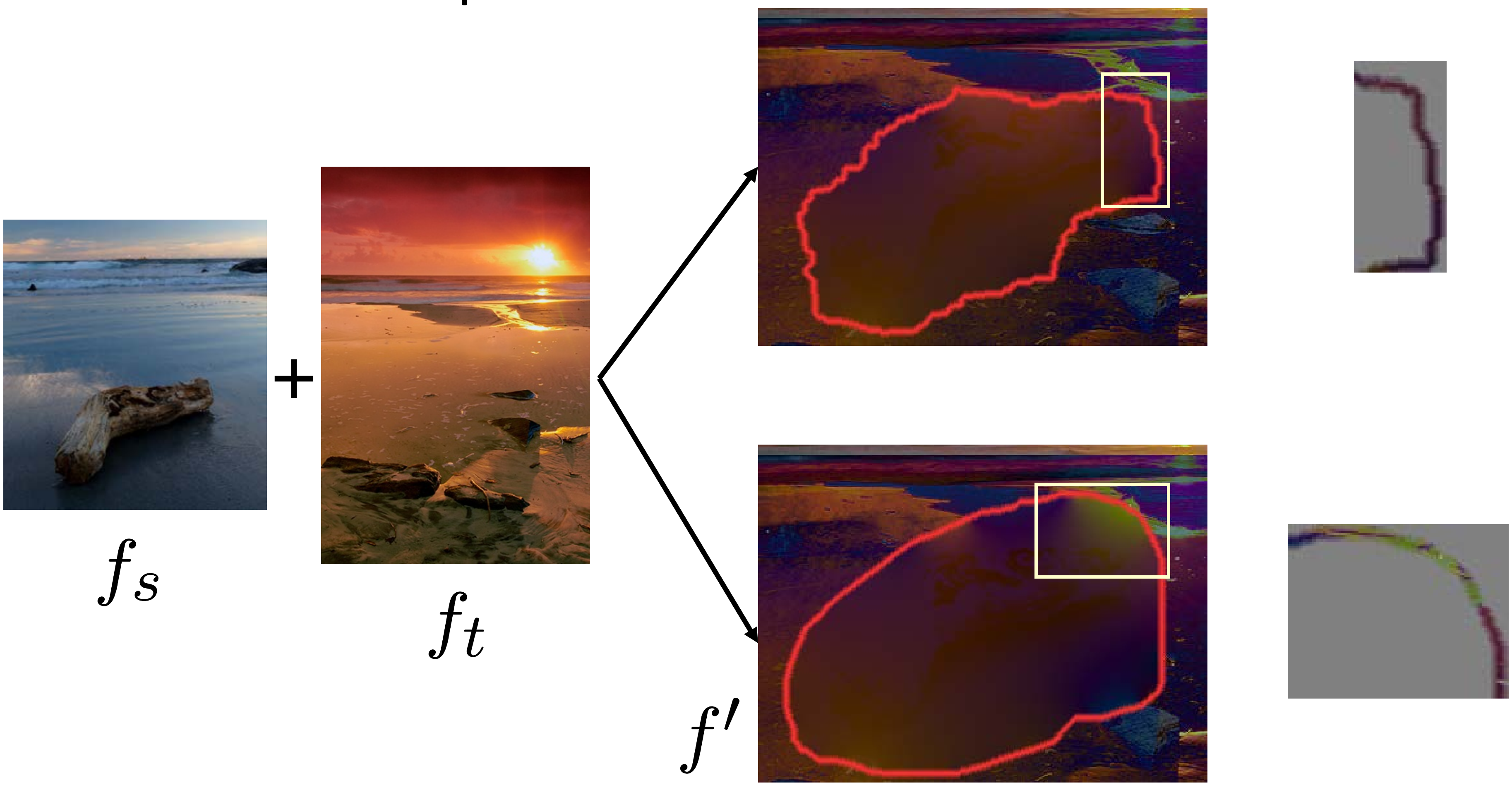
$\longrightarrow$



Slides of Jia *et al.*

Poisson Equations in Images

- The same example



Slides of Jia *et al.*

Poisson Equations in Images

- Where is the optimal boundary $\partial\Omega$?
  - Inside the user drawn region
  - Outside the object of interest

- How to optimize it?
  - Minimum color variance

$$\min \sum_{p \in \partial\Omega} ((f_t(p) - f_s(p)) - k)^2, \text{ s.t. } \partial\Omega \in \text{ blue}$$

Slides of Jia *et al.*

Boundary Optimization

$$E(\partial\Omega, k) = \sum_{p \in \partial\Omega} ((f_t(p) - f_s(p)) - k)^2, \text{ s.t. } \partial\Omega \in \text{blue}$$

- $\partial\Omega$ and k are all unknowns
- An iterative optimization
  - Initialize $\partial\Omega$ as the user drawn boundary.
  - Given new $\partial\Omega$, the optimal k is computed:

$$\frac{\partial E(\partial\Omega, k)}{\partial k} = 0$$

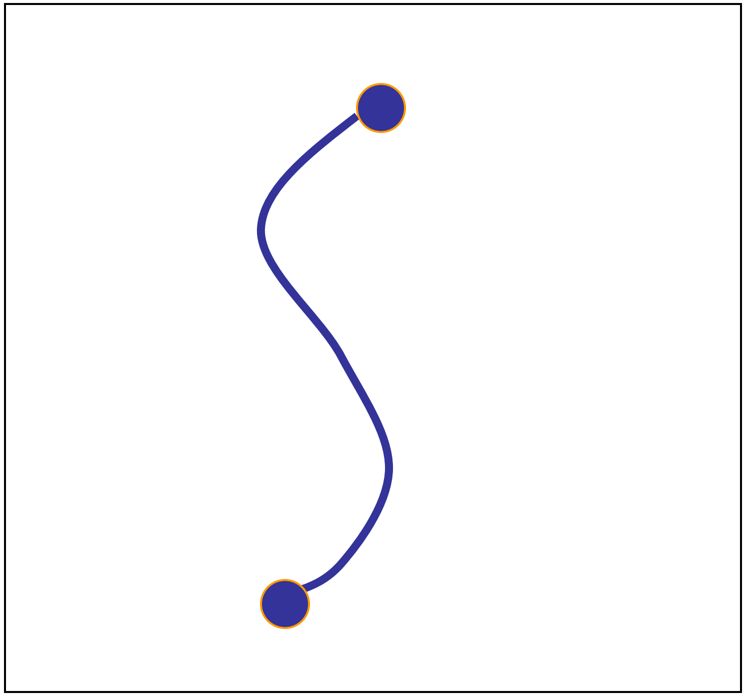
Shortest path problem

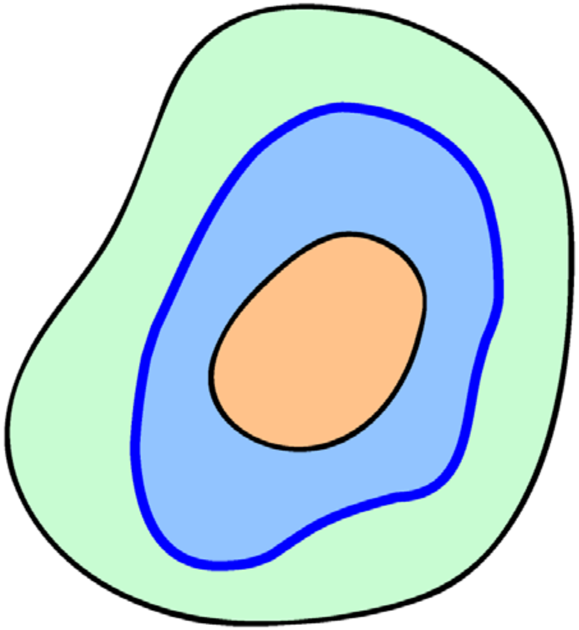
  - Given new k, optimize the boundary $\partial\Omega$.
  - Repeat the previous two steps until convergence.

Slides of Jia *et al.*

Boundary Optimization

- In 2D graph, computing the shortest path between any two points: Dynamic Programming
- Our problem is to compute a closed path

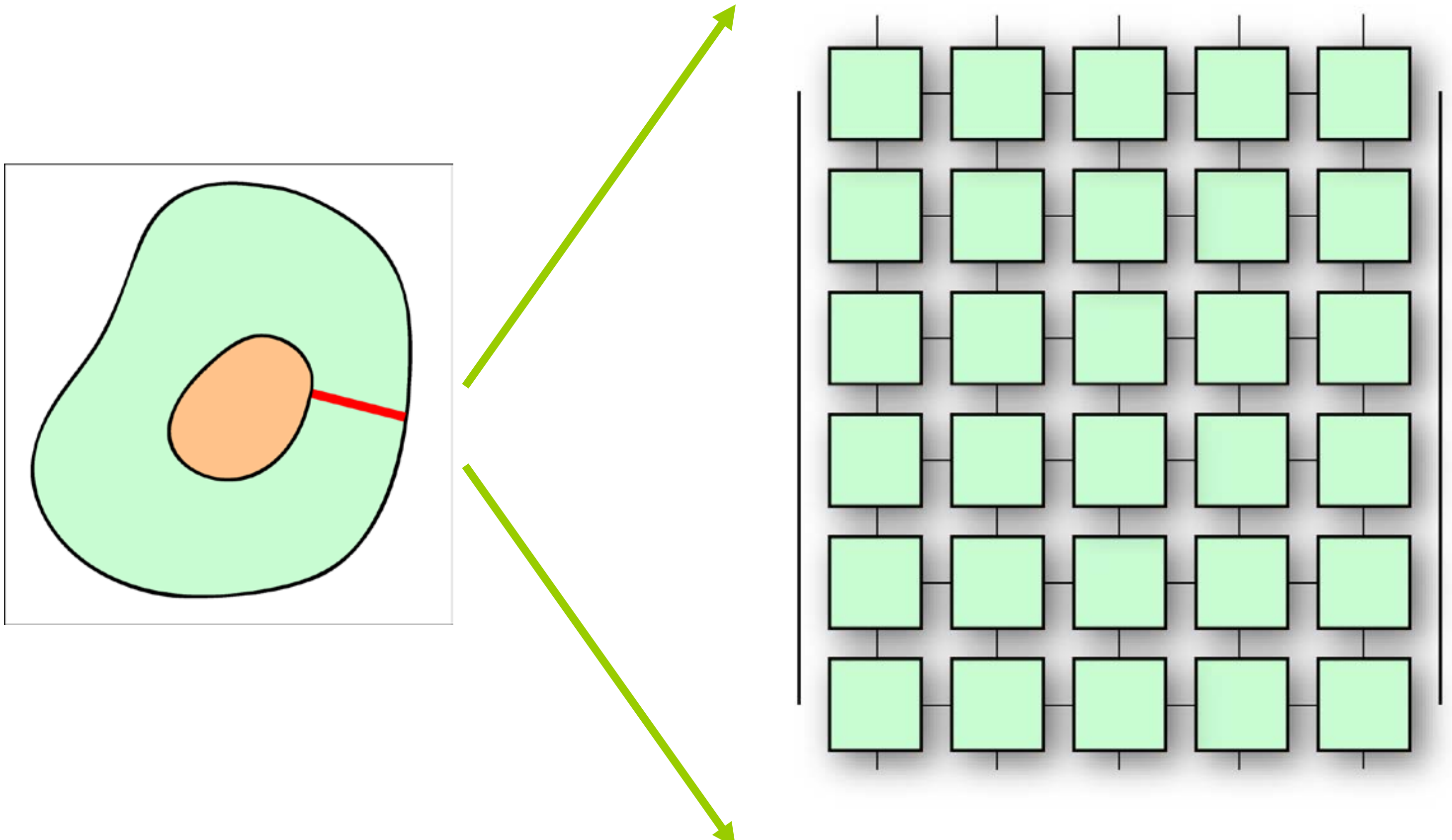




Slides of Jia *et al.*

Boundary Optimization

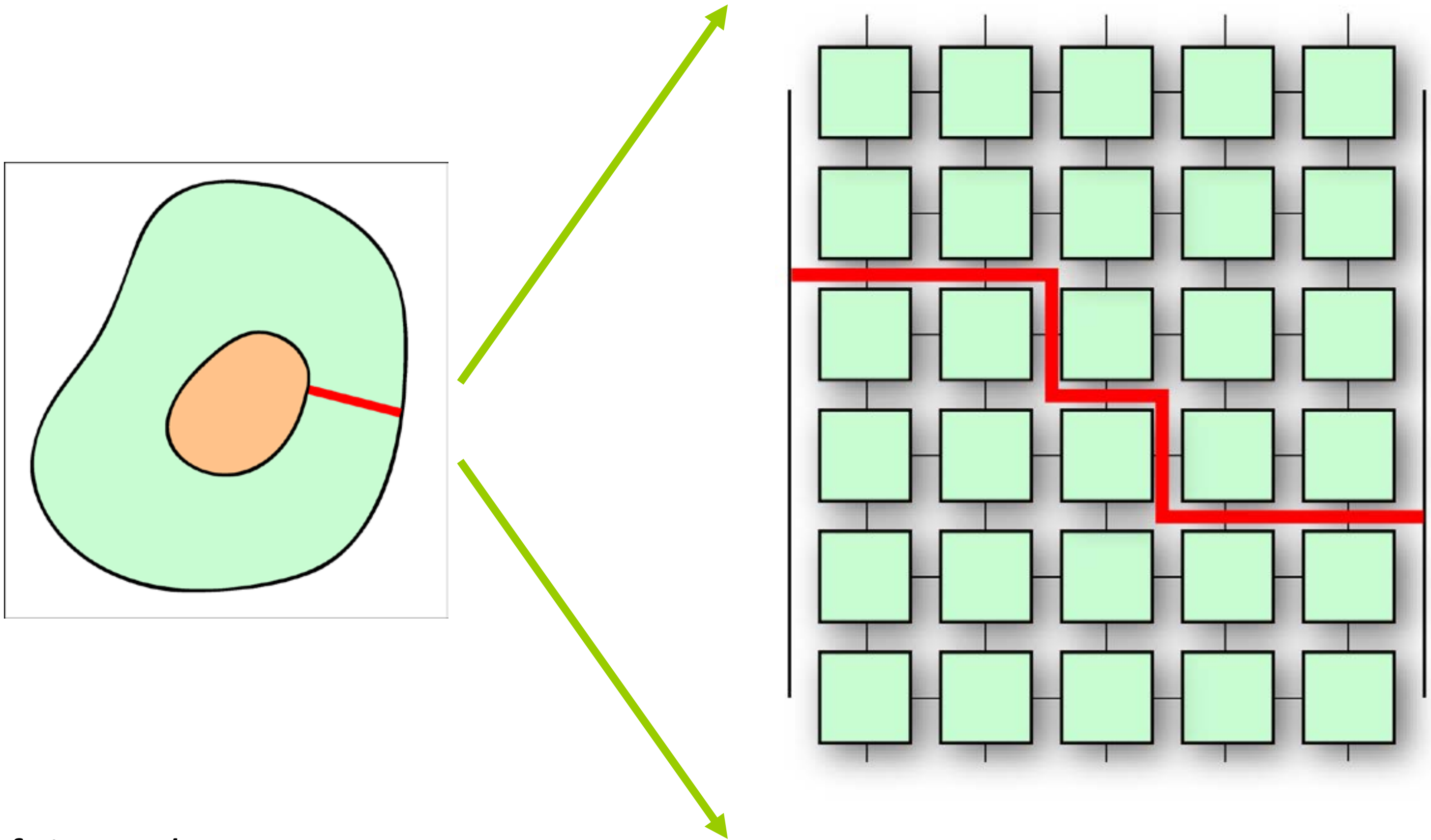
- A shortest closed-path algorithm
  - Breaking closed boundary



Slides of Jia *et al.*

Boundary Optimization

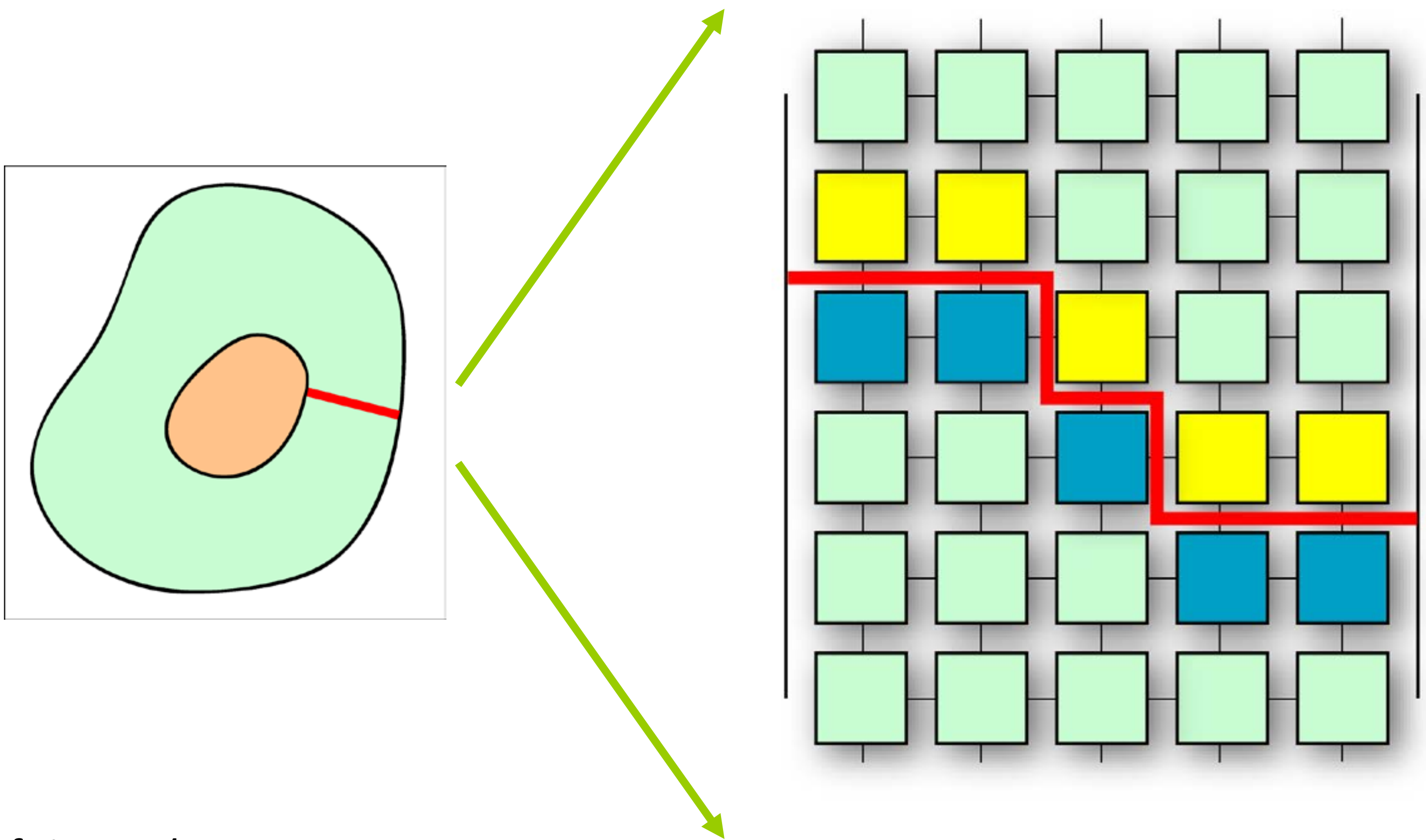
- A shortest closed-path algorithm
  - Breaking closed boundary



Slides of Jia *et al.*

Boundary Optimization

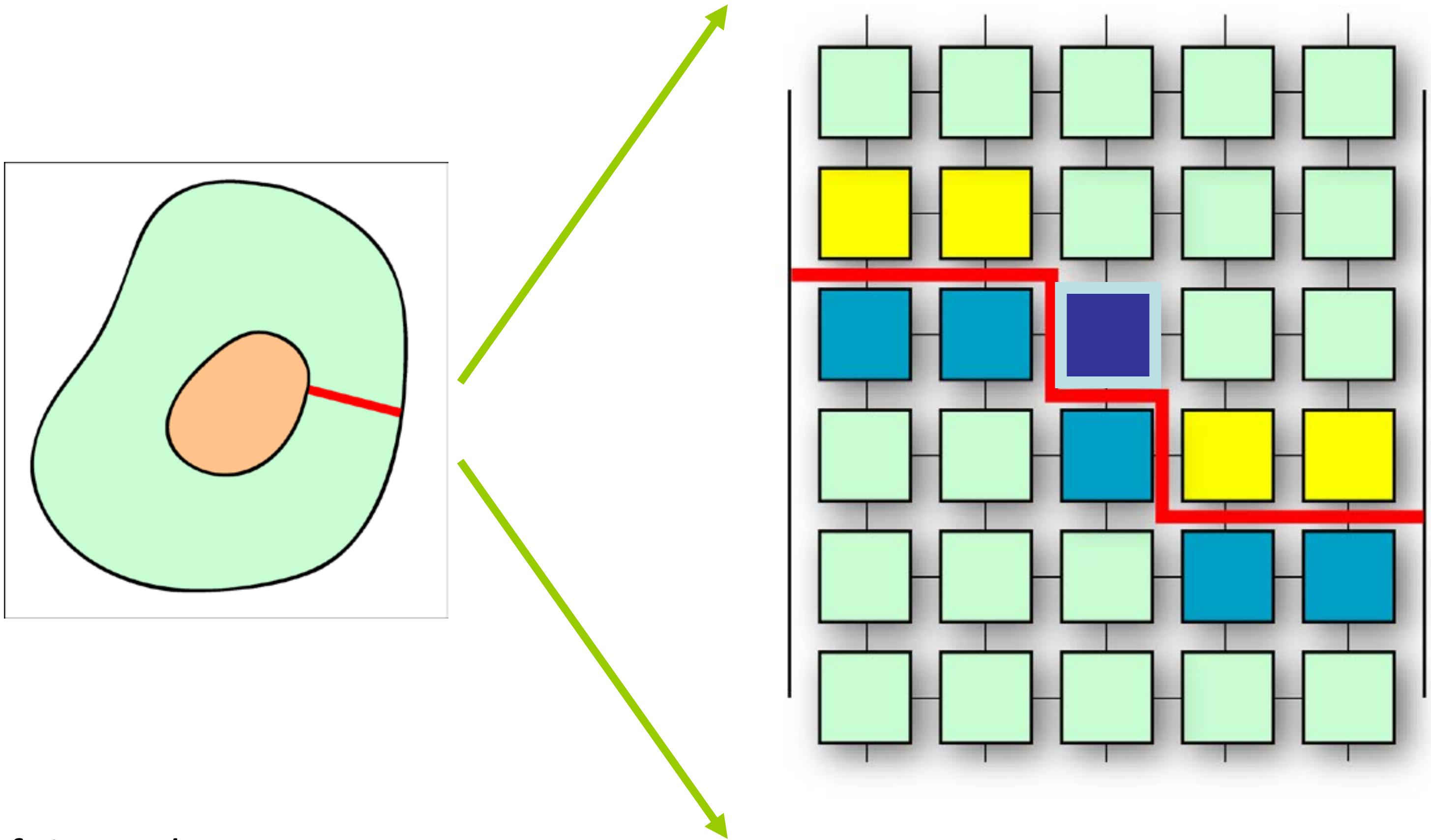
- A shortest closed-path algorithm
  - Breaking closed boundary



Slides of Jia *et al.*

Boundary Optimization

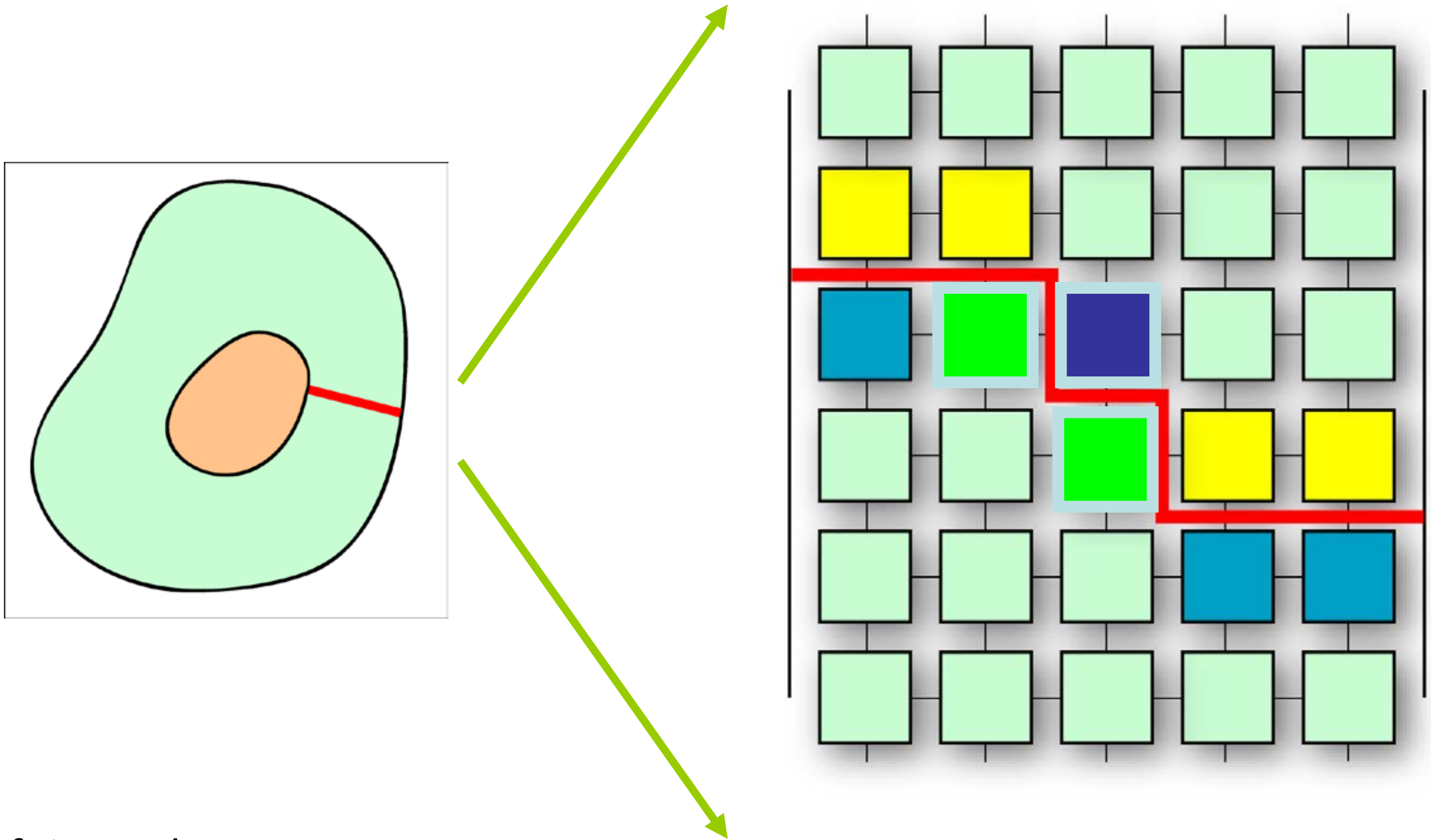
- A shortest closed-path algorithm



Slides of Jia *et al.*

Boundary Optimization

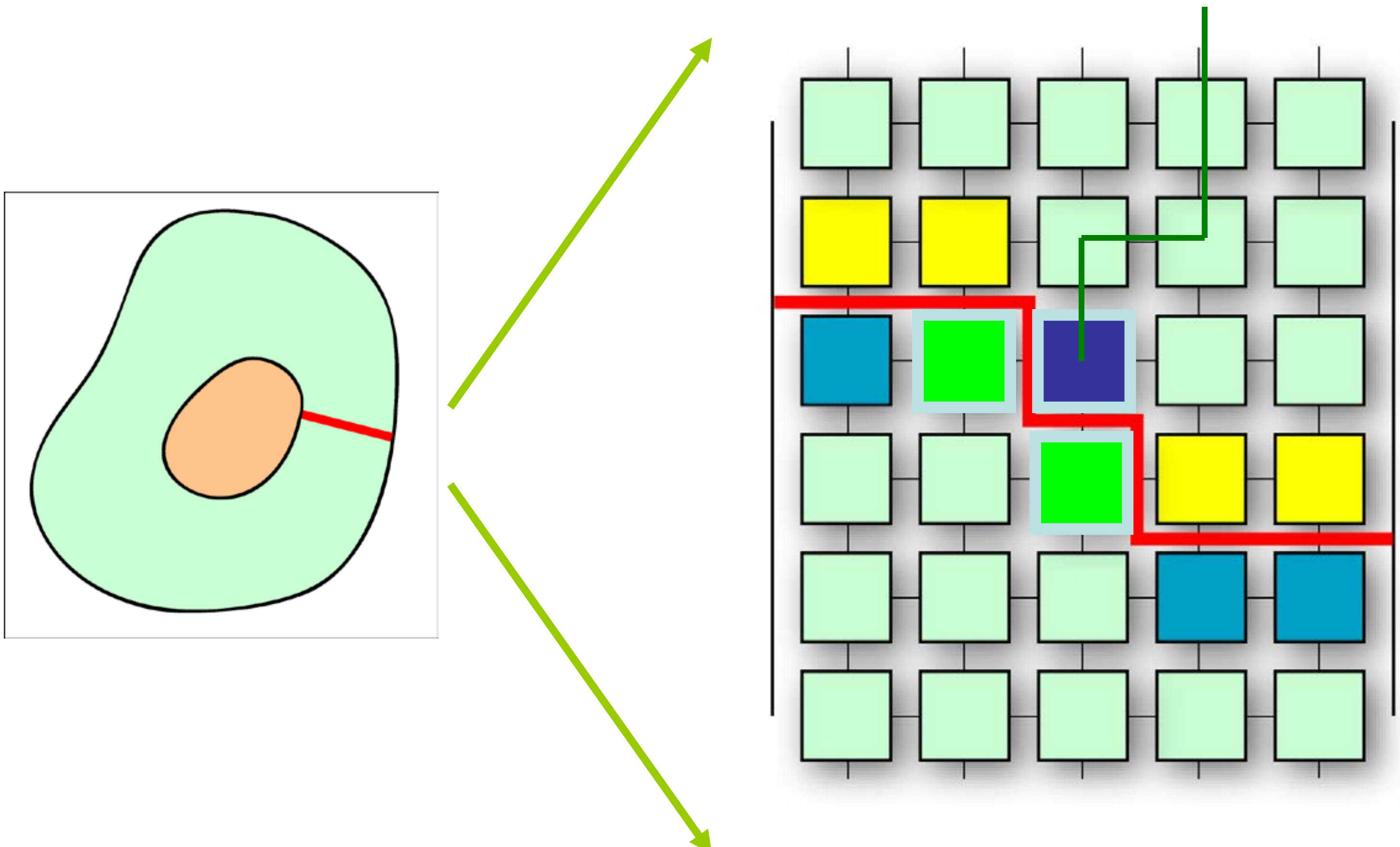
- A shortest closed-path algorithm



Slides of Jia *et al.*

Boundary Optimization

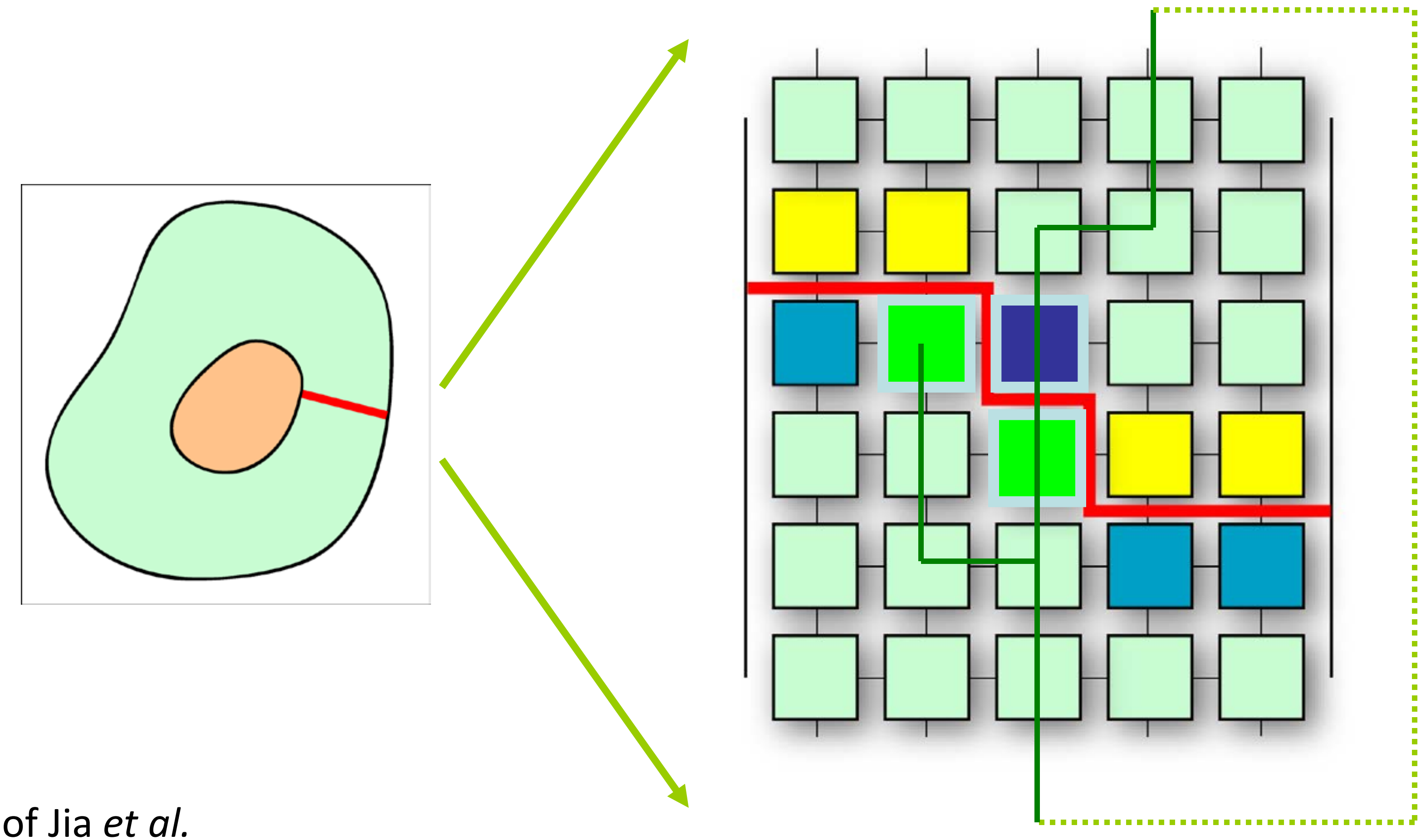
- A shortest closed-path algorithm



Slides of Jia *et al.*

Boundary Optimization

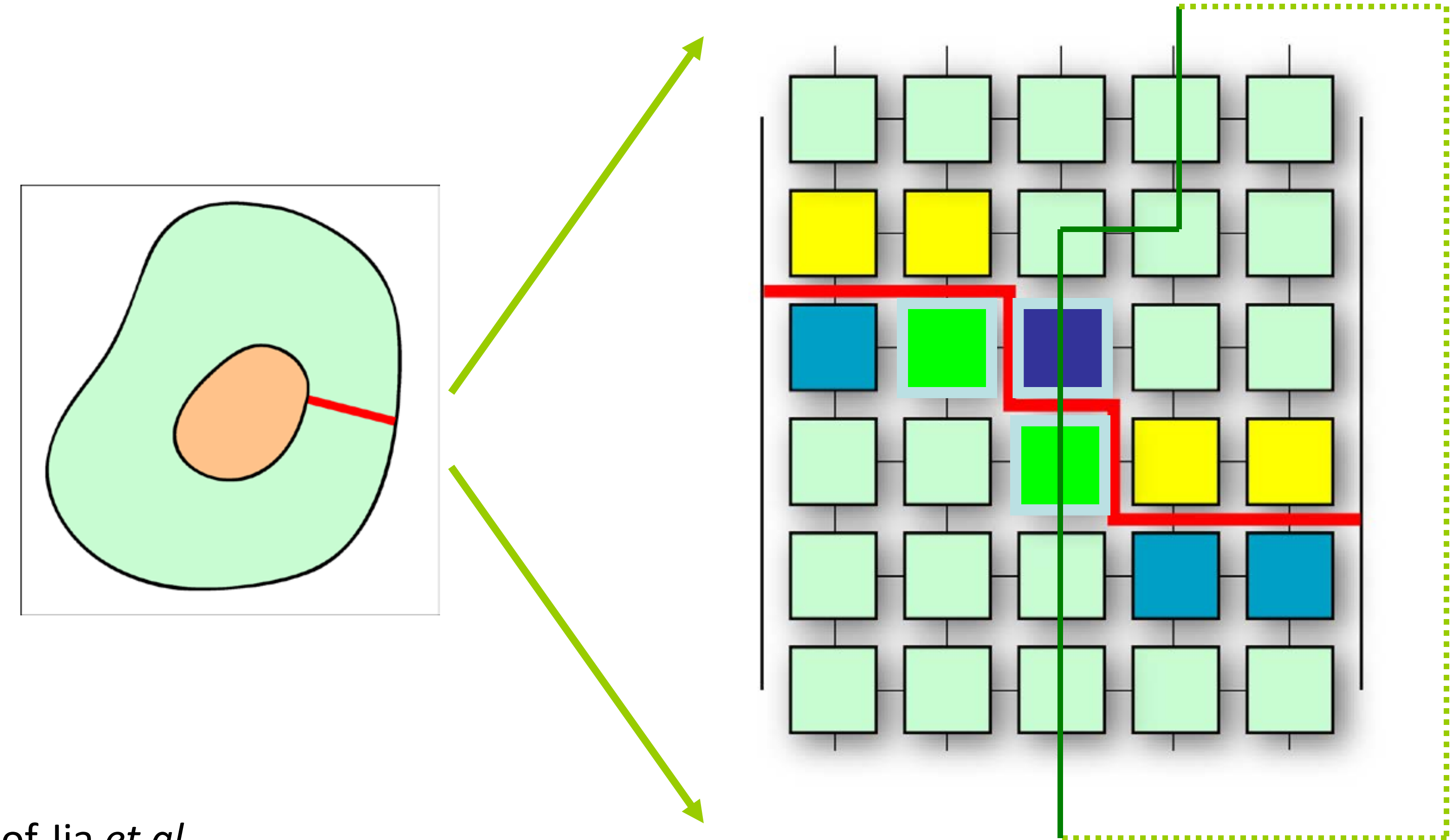
- A shortest closed-path algorithm



Slides of Jia *et al.*

Boundary Optimization

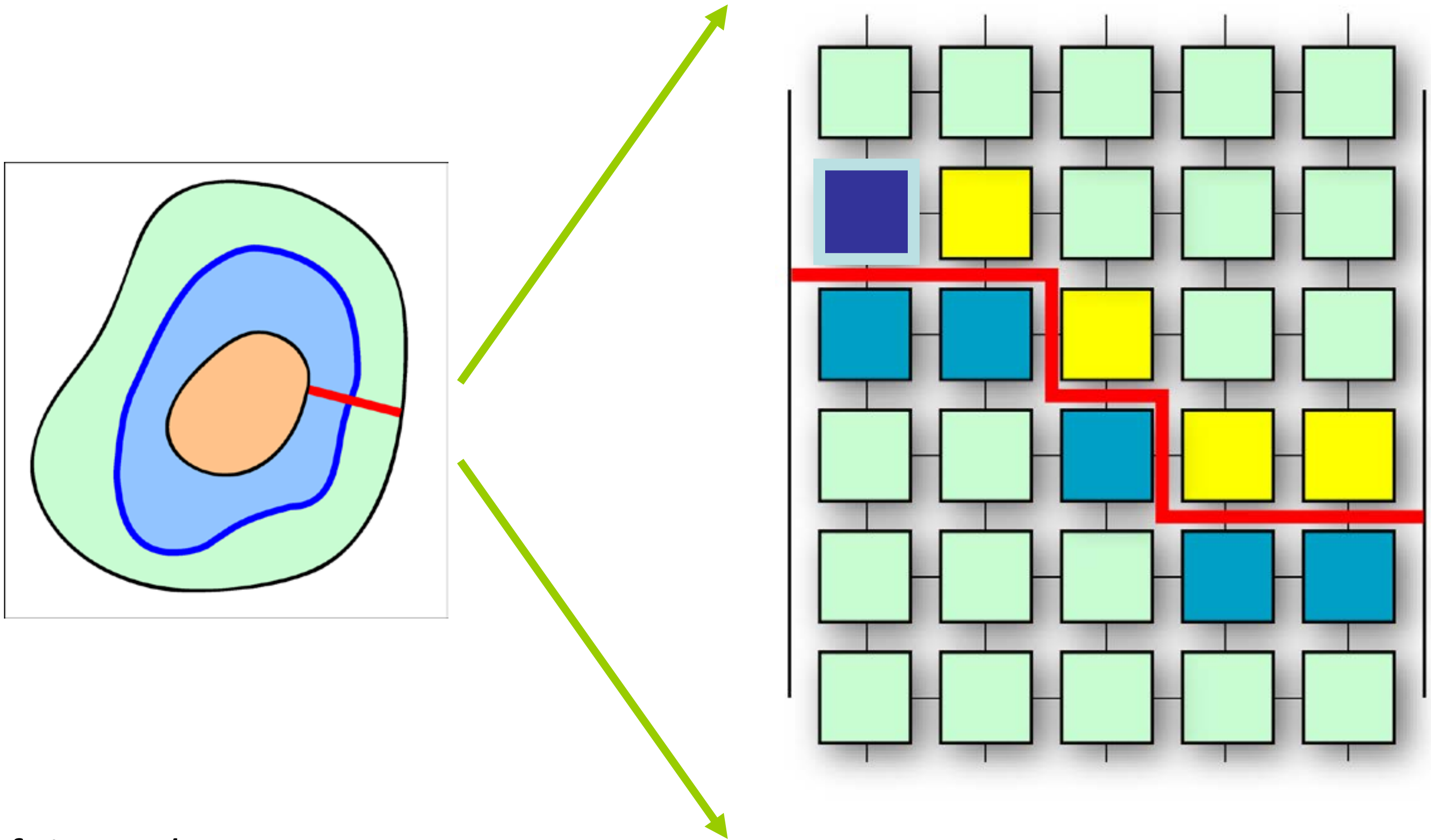
- A shortest closed-path algorithm
  - Computation complexity $O(N)$



Slides of Jia *et al.*

Boundary Optimization

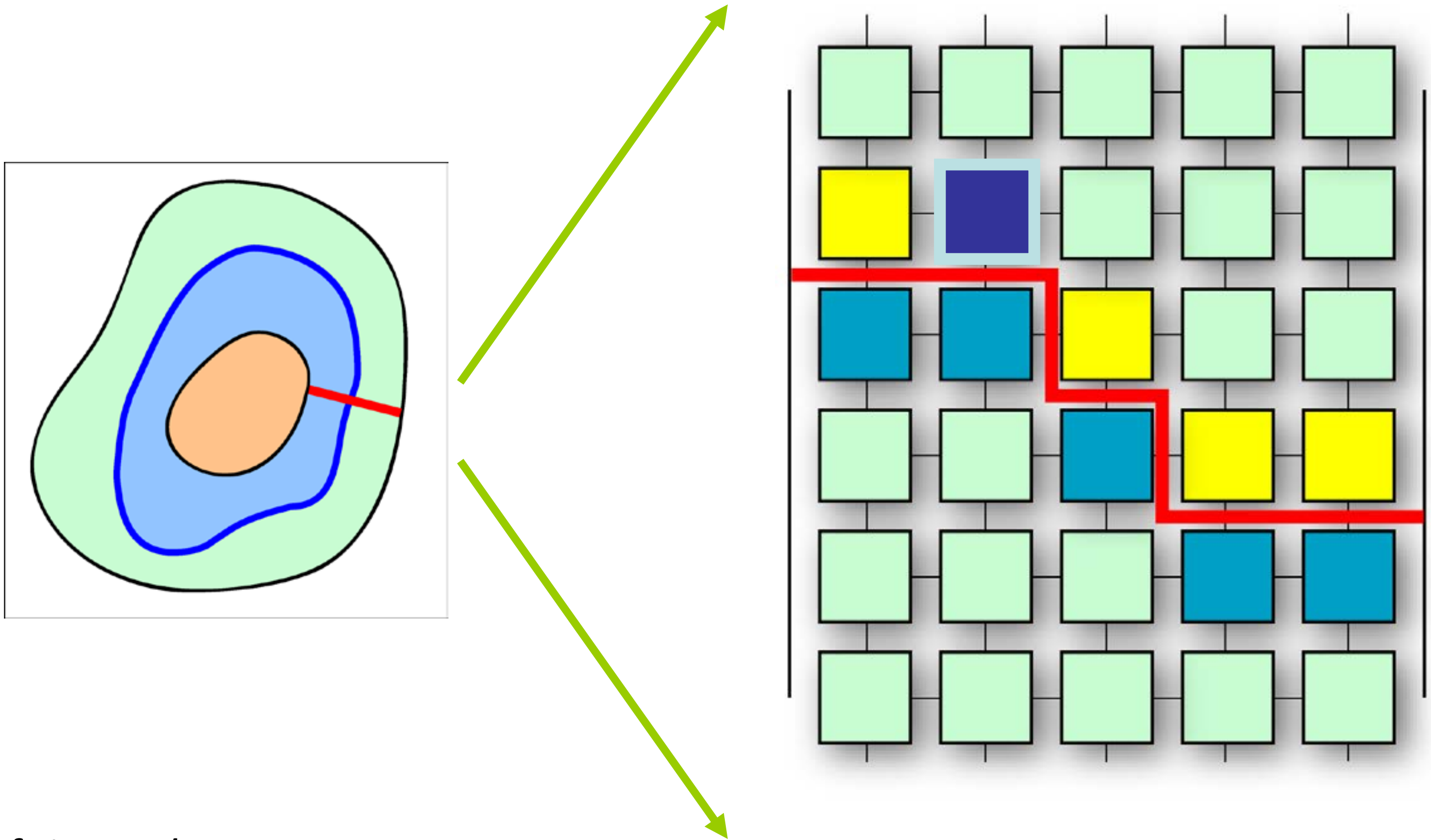
- A shortest closed-path algorithm



Slides of Jia *et al.*

Boundary Optimization

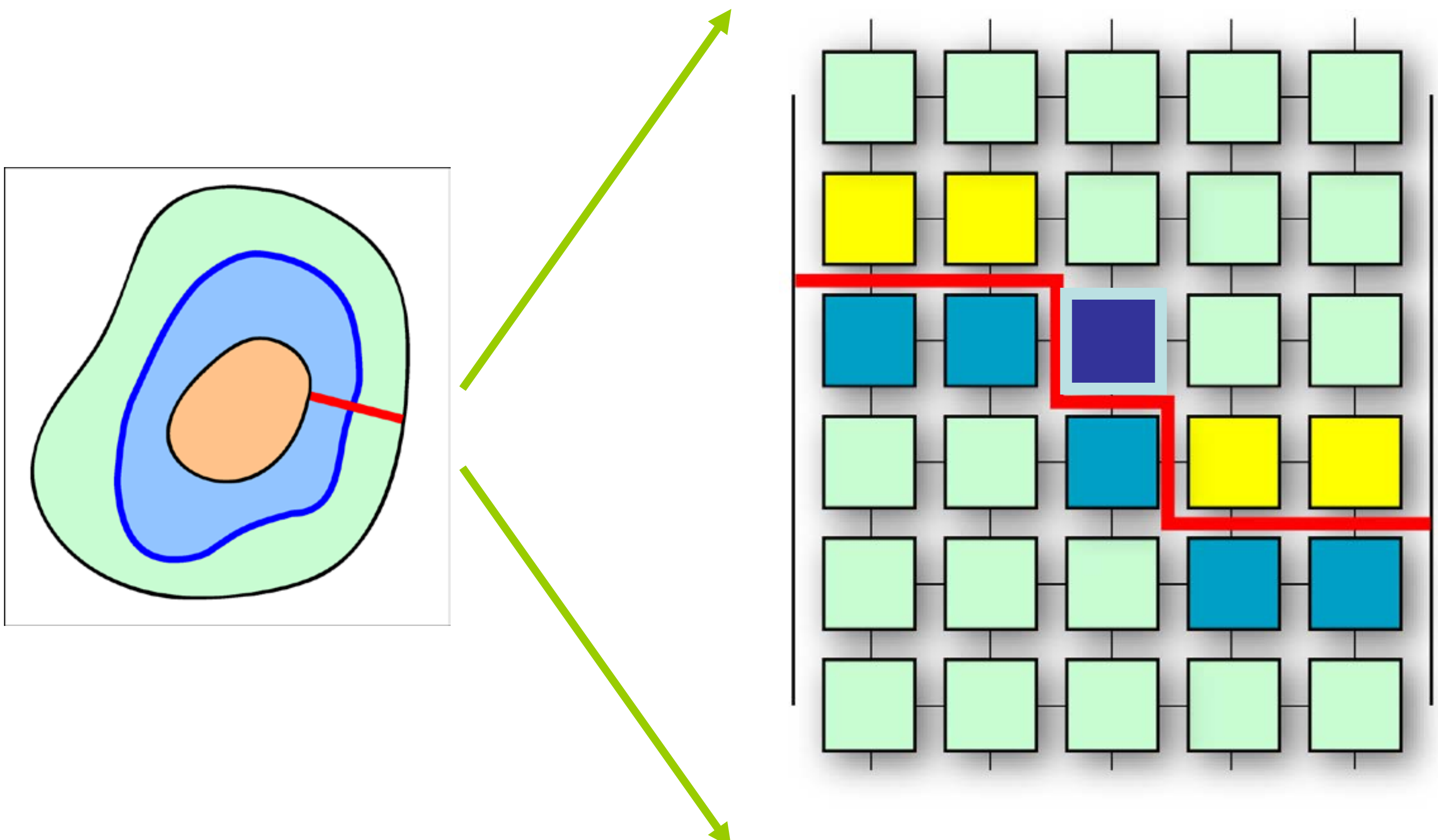
- A shortest closed-path algorithm



Slides of Jia *et al.*

Boundary Optimization

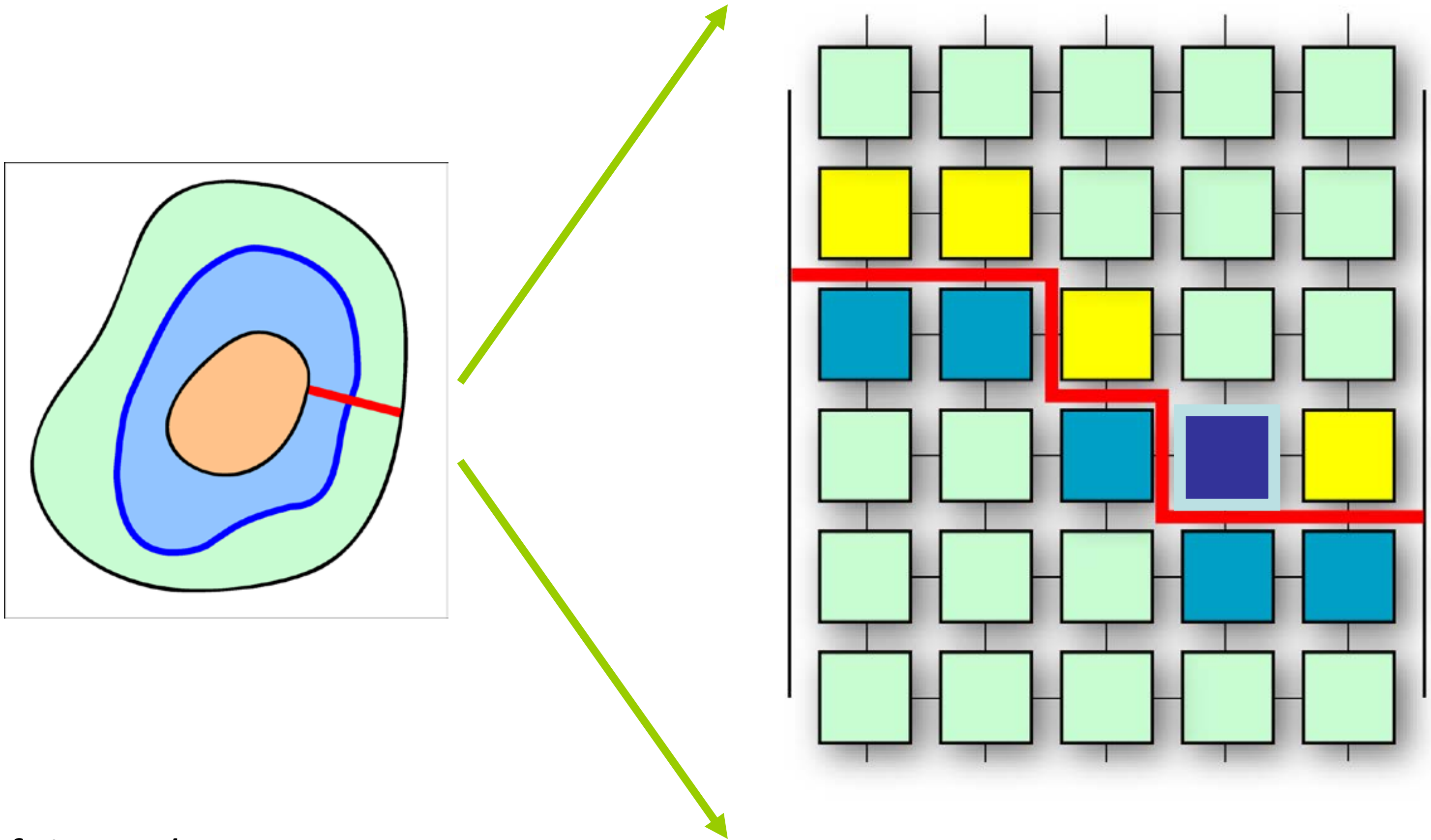
- A shortest closed-path algorithm



Slides of Jia *et al.*

Boundary Optimization

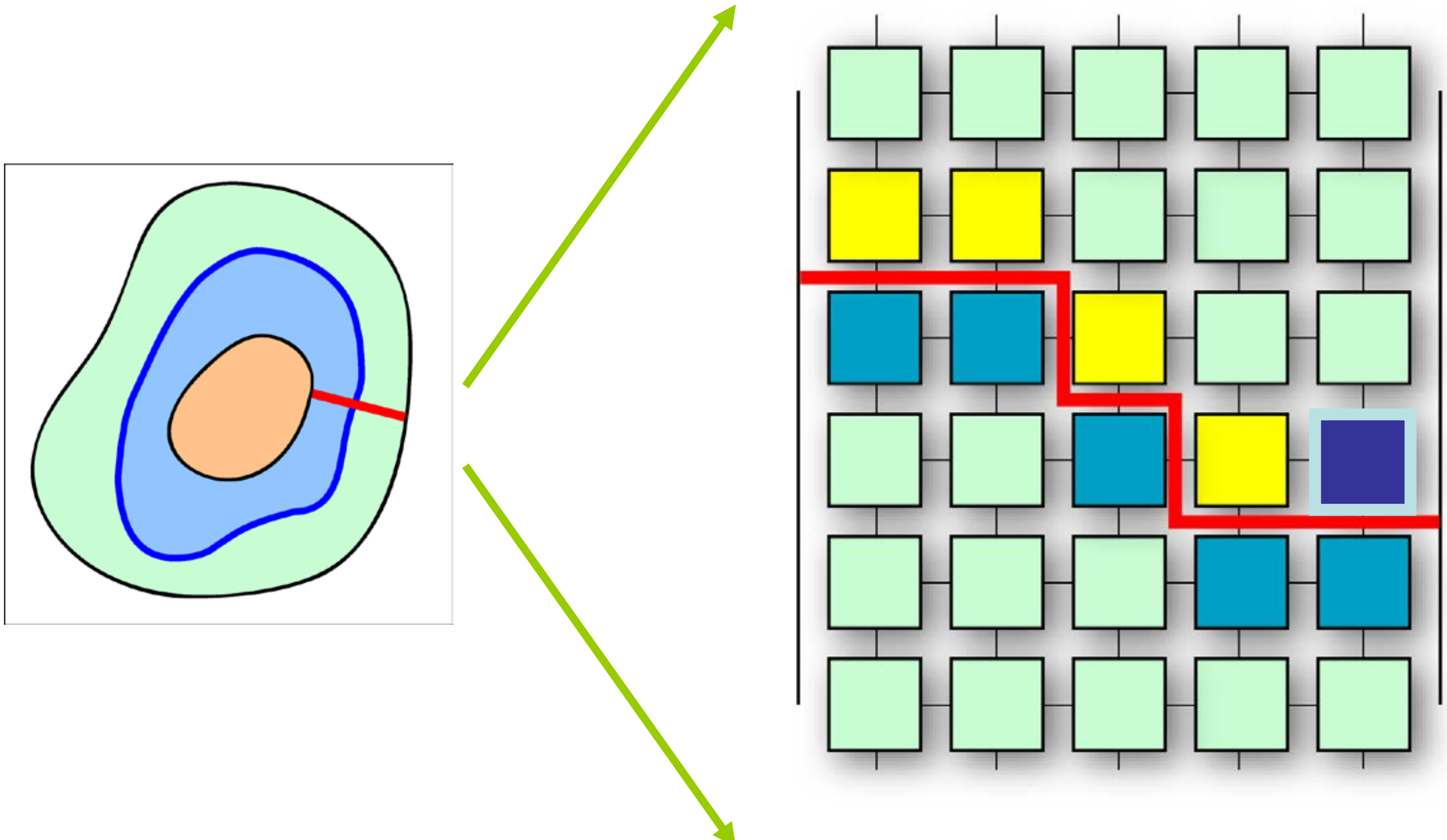
- A shortest closed-path algorithm



Slides of Jia *et al.*

Boundary Optimization

- A shortest closed-path algorithm
  - Total computation complexity $O(NM)$

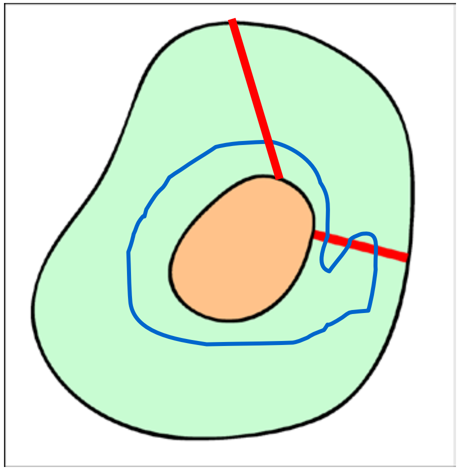


Slides of Jia *et al.*

Boundary Optimization Discussion

- Optimality
  - Avoiding that the path twists around the cut by selecting the initial cut position.

- How to select the initial cut?
  - Making it short to reduce $O(MN)$
  - Passing smooth region



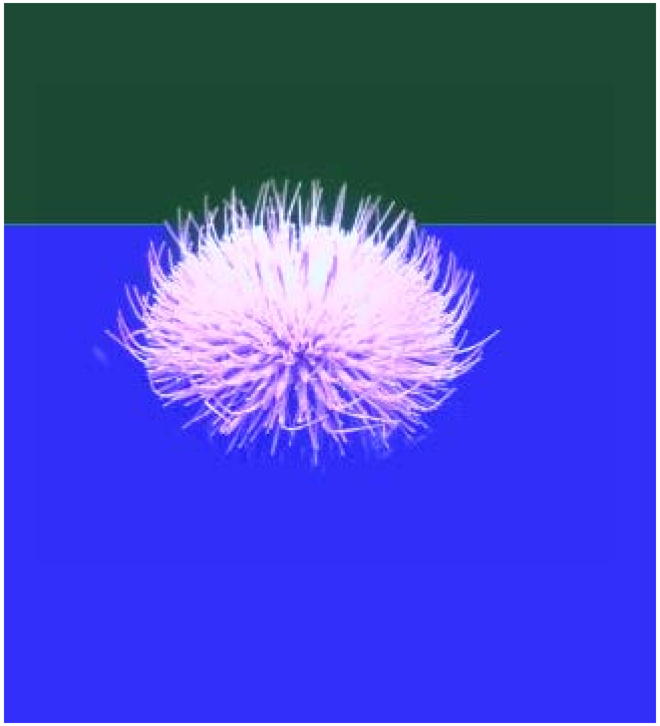
Slides of Jia *et al.*

Integrating Fractional Boundary

- The alpha blending and Poisson blending are two separated methods in previous work.
  - Alpha blending maintains fractional boundary but cannot modify the color of the source object.
  - Poisson blending can modify the color of the source object but only uses a binary boundary.
  - They are integrated in our method.

Integrating Fractional Boundary

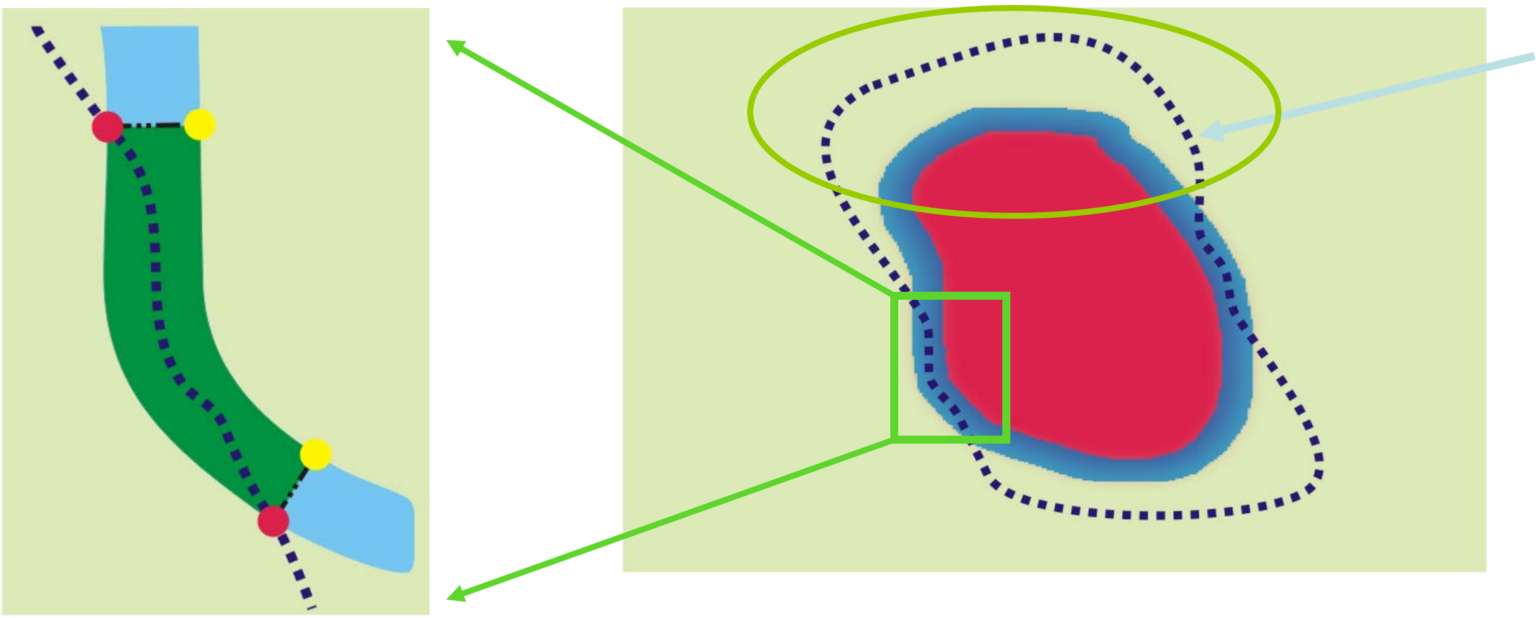
- Fractional boundary is important in image composting:



Slides of Jia *et al.*

Integrating Fractional Boundary

- Where to use the fractional values?
  - only the pixels where the optimized boundary is near *the blue ribbon*



Slides of Jia *et al.*

Integrating Fractional Boundary

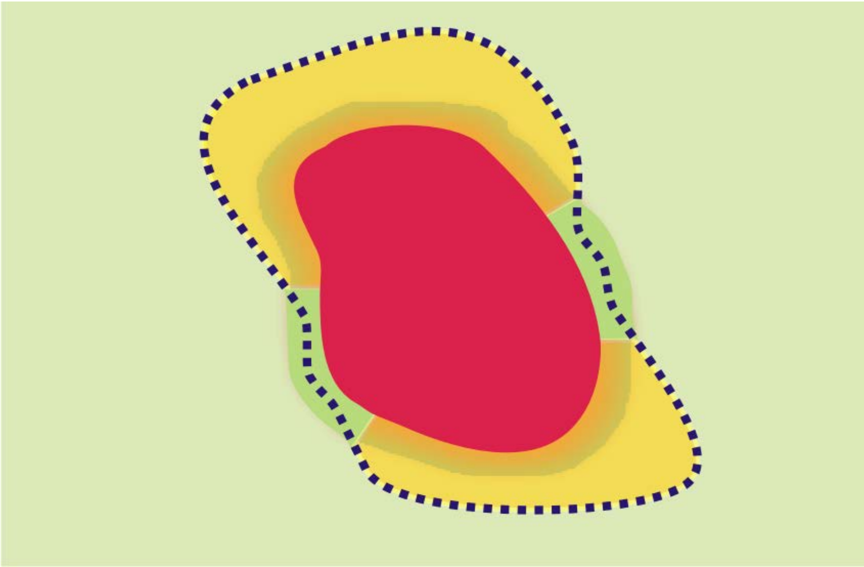
- Where to use the fractional values?
  - only the pixels where the optimized boundary is near *the blue ribbon*

fractional integration:

the green region

otherwise:

the yellow region



Slides of Jia *et al.*

Integrating Fractional Boundary

- How to integrate the fractional values in Poisson blending?
  - A blended guidance field

$$\nabla_x f(x,y) = f(x+1,y) - f(x,y)$$

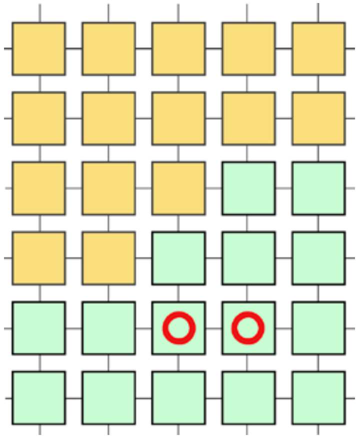
$$v'_x(x,y) = \begin{cases} \nabla_x f_s(x,y), & (x,y),(x+1,y) \in \text{ yellow };\\ \nabla_x(\alpha f_s + (1-\alpha) f_t), & (x,y),(x+1,y) \in \text{ green };\\ 0, & \text{otherwise}. \end{cases}$$

Slides of Jia *et al.*

Integrating Fractional Boundary

- How to integrate the fractional values in Poisson blending?
  - A blended guidance field

$$v'_x(x,y) = \begin{cases} \nabla_x f_s(x,y), & (x,y),(x+1,y) \in \text{ yellow };\\ \nabla_x(\alpha f_s + (1-\alpha) f_t), & (x,y),(x+1,y) \in \text{ green };\\ 0, & \text{otherwise}. \end{cases}$$



Slides of Jia *et al.*

Integrating Fractional Boundary

- How to integrate the fractional values in Poisson blending?
  - A blended guidance field

$$v'_x(x,y) = \begin{cases} \nabla_x f_s(x,y), & (x,y),(x+1,y) \in \text{ yellow };\\ \nabla_x(\alpha f_s + (1-\alpha) f_t), & (x,y),(x+1,y) \in \text{ green };\\ 0, & \text{otherwise}. \end{cases}$$

Slides of Jia *et al.*

Integrating Fractional Boundary

- Final minimization:

$$\min_{f} \int_{p \in \Omega^*} \|\nabla f - v'\|^2 \, dp \ \text{ with } f|_{\partial\Omega^*} = f_t|_{\partial\Omega^*}$$

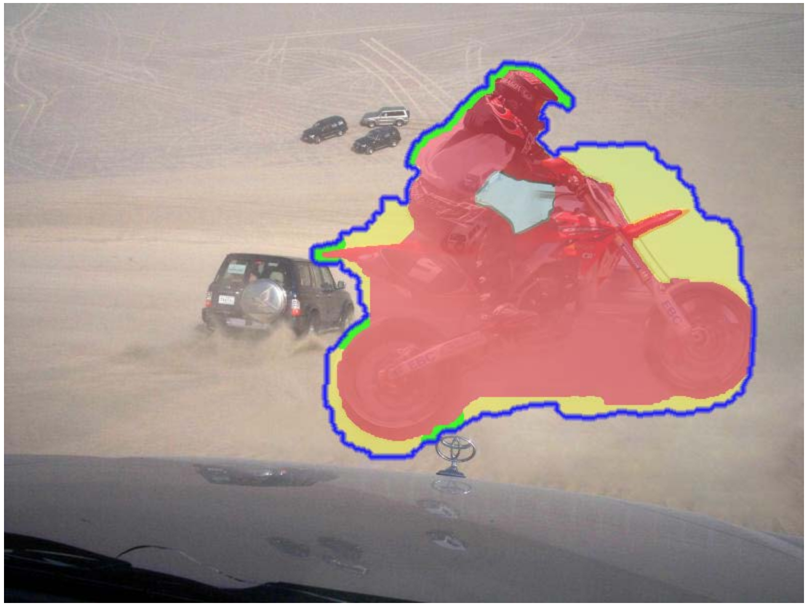
- Solving the corresponding Poisson equation.

Slides of Jia *et al.*

Results and Comparison

Slides of Jia *et al.*

Results and Comparison



Slides of Jia *et al.*

Results and Comparison

Jia *et al.*

Alpha blending

Slides of Jia *et al.*

Results and Comparison

Jia *et al.*

Poisson blending

Slides of Jia *et al.*

Results and comparison

Slides of Jia *et al.*

Results and comparison

Jia *et al.*

Alpha blending

Slides of Jia *et al.*

Results and comparison

Jia *et al.*

Poisson blending

Slides of Jia *et al.*

Results

Slides of Jia *et al.*

Results

Slides of Jia *et al.*

Additional Assignment

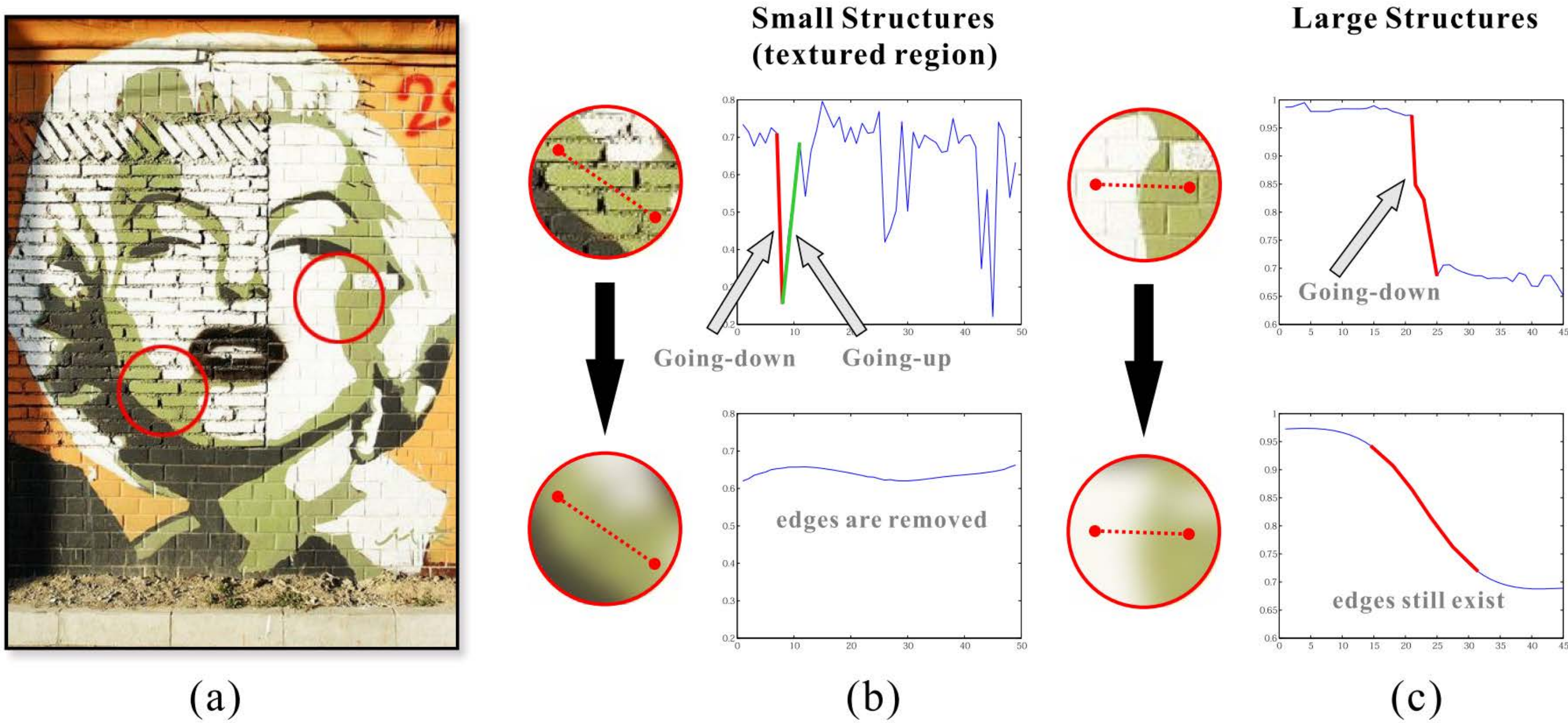
- Image abstraction → video tooning

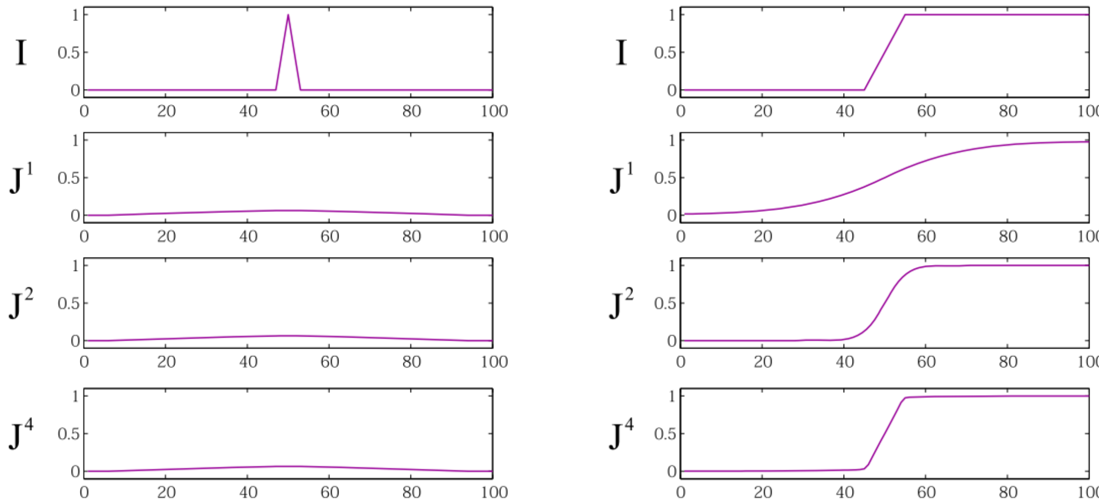
心機掃瞄

- Rolling Guidance Filter, Zhang et al.
  - http://www.cse.cuhk.edu.hk/leojia/projects/rollguidance/
- Video tooning, Wang et al.
  - http://juew.org/publication/VideoTooningFinal.pdf

Rolling Guidance Filter

- Zhang et al., ECCV 2014





(a) (b)

Algorithm 1 Rolling Guidance Using Bilateral Filter

Input: I, σ_s, σ_r, N^{iter}

Output: I^{new}

1: Initialize J^0 as a constant image
2: **for** t:= 1 **to** N^{iter} **do**
3: $\quad J^t \leftarrow JointBilateral(I, J^{t-1}, \sigma_s, \sigma_r)$ {Input: I; Guidance: J^{t-1} }
4: **end for**
5: $I^{\text{new}} \leftarrow J^{N^{\text{iter}}}$