Poisson Image Editing

Patric Perez, Michel Gangnet, and Andrew Black
(SIGGRAPH 2003)

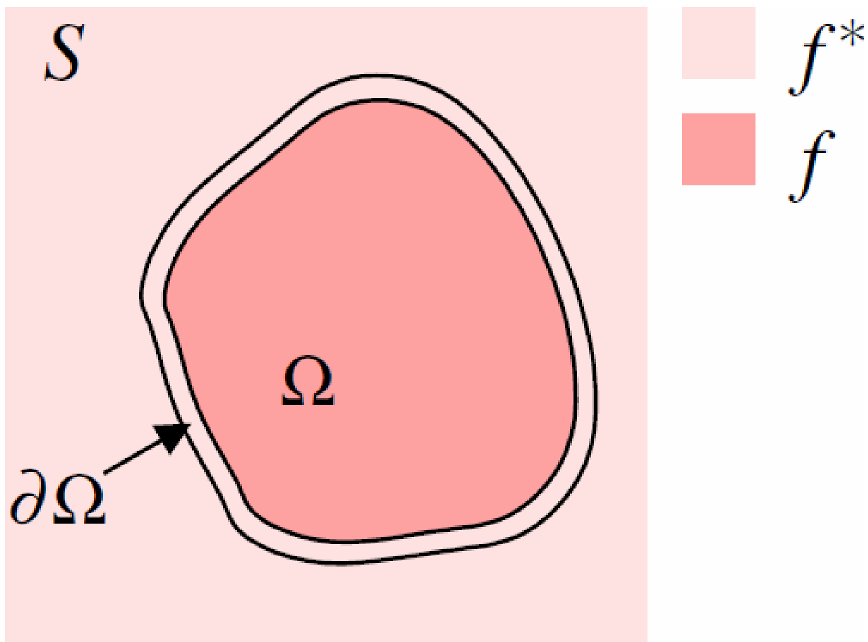
Seminar Talk by

Tim Weyrich

Overview

- Guided Image Interpolation
- Discretized Solution
- Editing Operations
- Discussion

Interpolation Problem

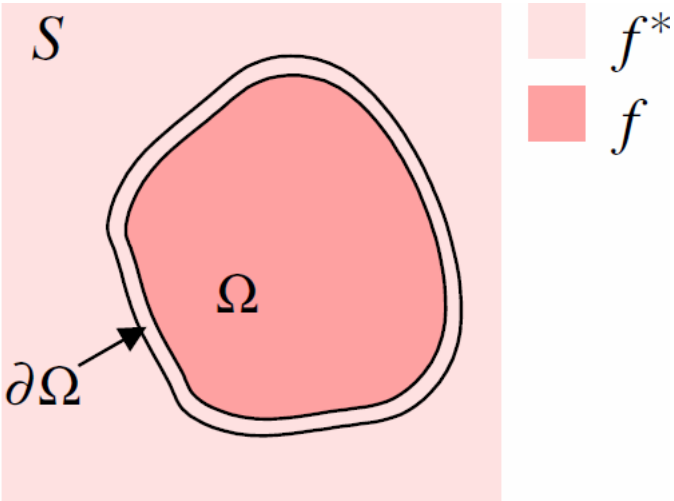


- f^*: known image values
- f : unknown values over region Ω
- Assuming scalar image values

Simple Interpolation

- Maximize smoothness

$$\min_f \int_\Omega \|\nabla f\|^2$$



- Boundary constraints

$$f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

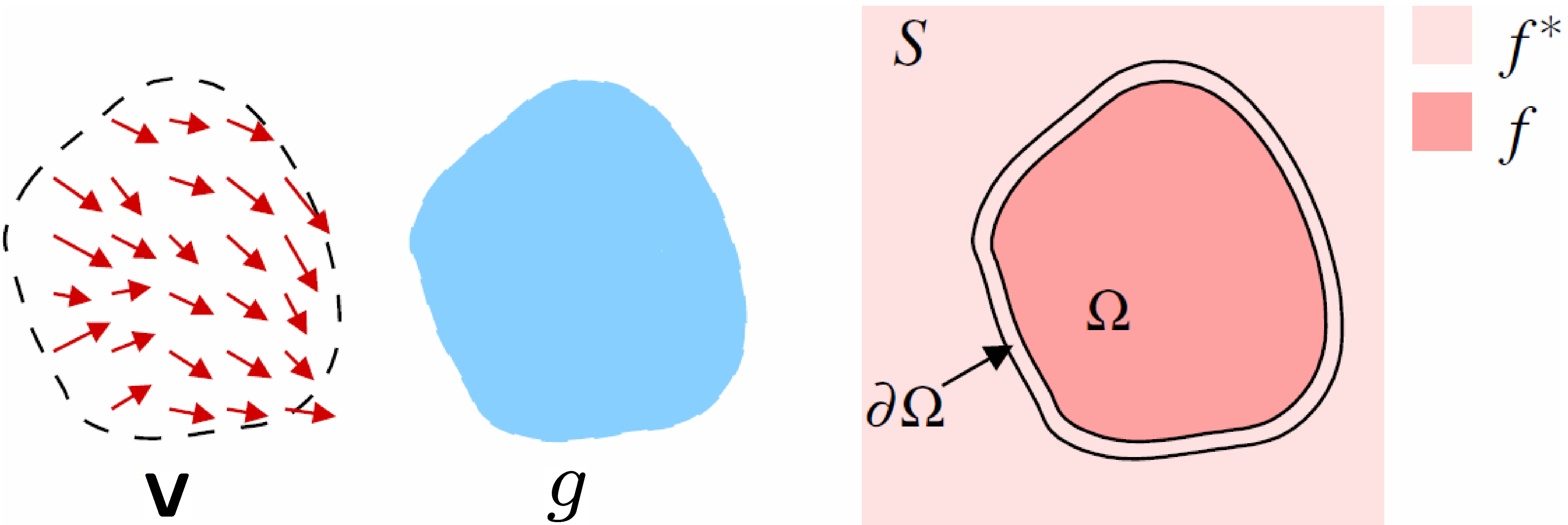
Simple Interpolation

- Solution: *Laplace Equation* with Dirichlet boundary conditions

$$\nabla^2 f = 0, \qquad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

- Membrane solution
- Unsatisfactory due to over-blurring

Guided Interpolation

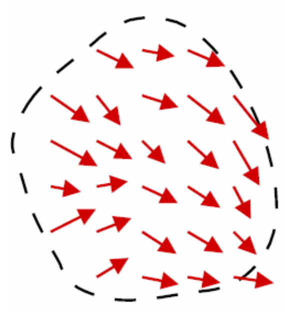


- **v**: guided field
- **v** may be gradient of a function *g*

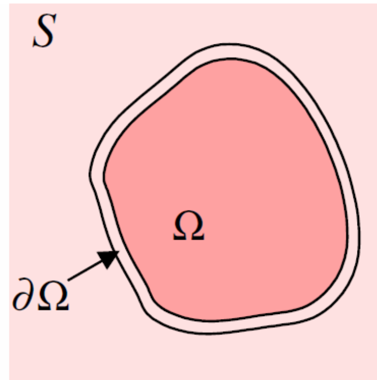
Guided Interpolation

- Minimize difference of gradient fields

$$\min_f \int_\Omega \|\nabla f - \mathbf{v}\|^2$$



$\mathbf{v}$



- Solution: *Poisson Equation* with Dirichlet boundary conditions

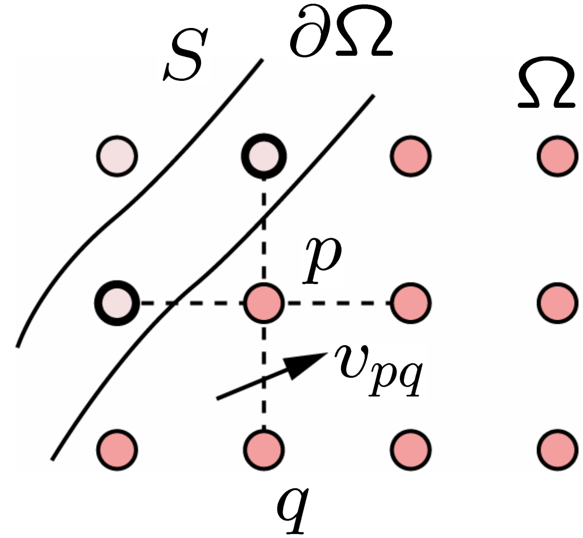
$$\nabla^2 f = \operatorname{div} \mathbf{v}, \qquad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

Discrete Poisson Solver

- Discretize $\min_f \int_\Omega \|\nabla f - \mathbf{v}\|^2$ directly by

$$\min_{f|_\Omega} \sum_{\langle p,q \rangle \cap \Omega \neq 0} (f_p - f_q - v_{pq})^2$$

$$f_p = f_p^*, \ \forall p \in \partial\Omega$$



for neighbors *p* and *q* with

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

Discrete Poisson Solver

- Minimum satisfies linear system of equations

If neighborhood N_p overlaps boundary:

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

For interior points:

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

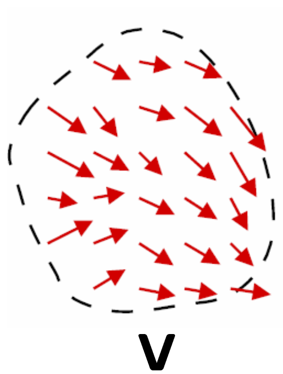
Discrete Poisson Solver

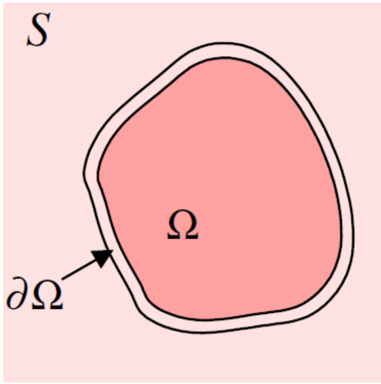
- Linear system of equations
  - sparse (banded)
  - symmetric
  - positive-definite
- Irregular shape of boundary requires general solver, such as
  - Gauss-Seidel iteration
  - Multi-grid
- System can be solved at interactive rates

Seamless Cloning

- Importing Gradients from a *Source Image g*

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

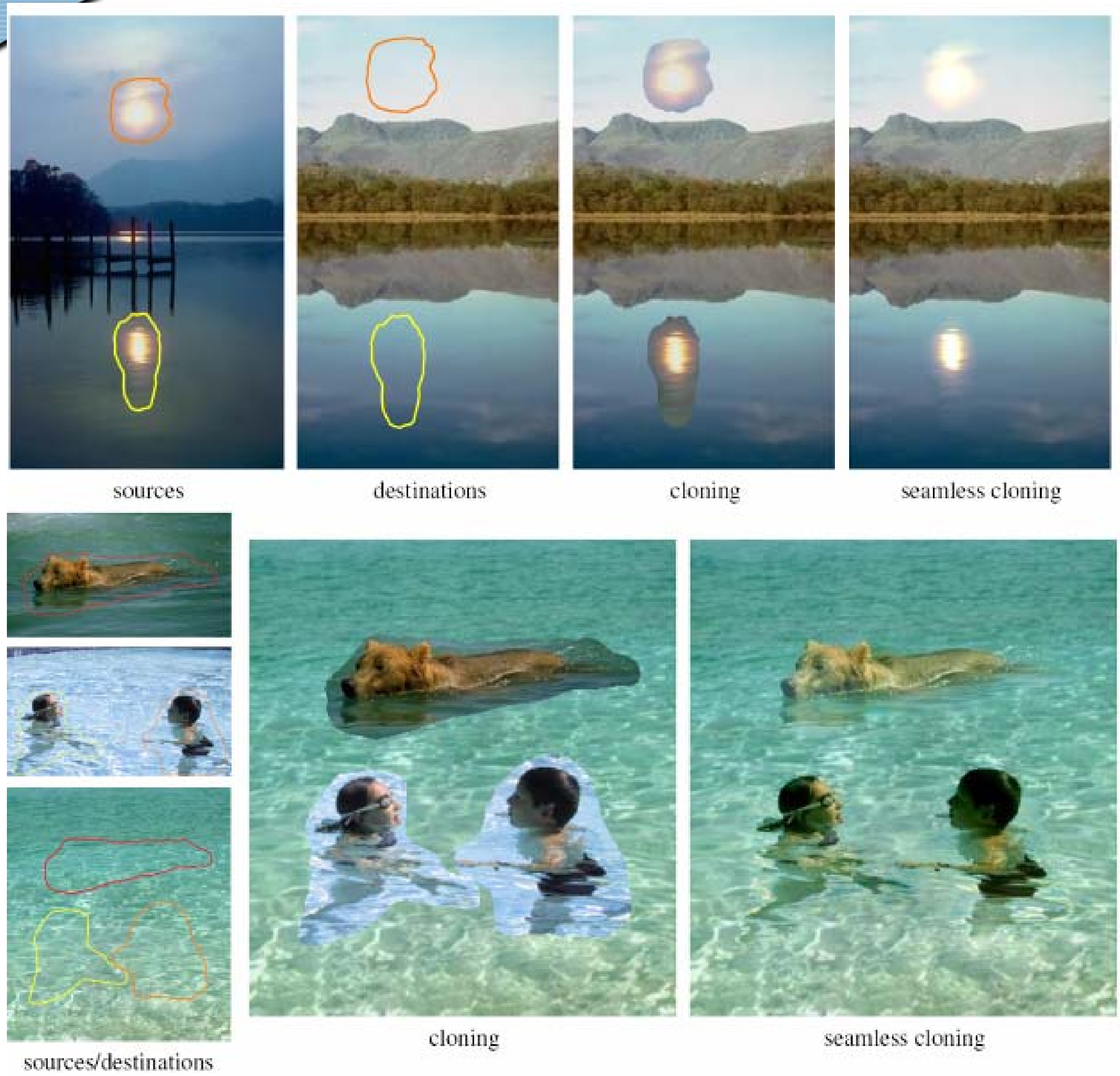


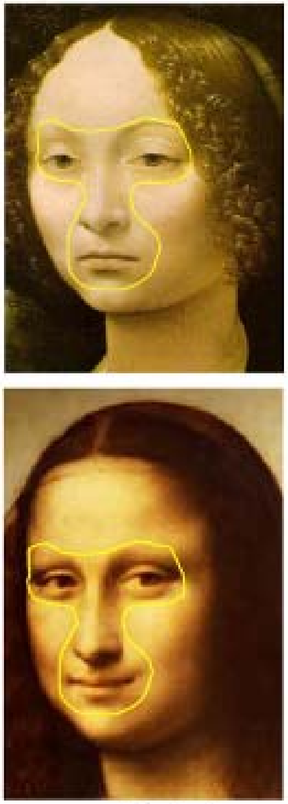


- Discretize

$$v_{pq} := g_p - g_q, \qquad \forall \langle p, q \rangle$$

Seamless Cloning Results


sources
destinations
cloning
seamless cloning
cloning
seamless cloning
sources/destinations



source/destination

cloning

seamless cloning

swapped textures

Mixing Gradients

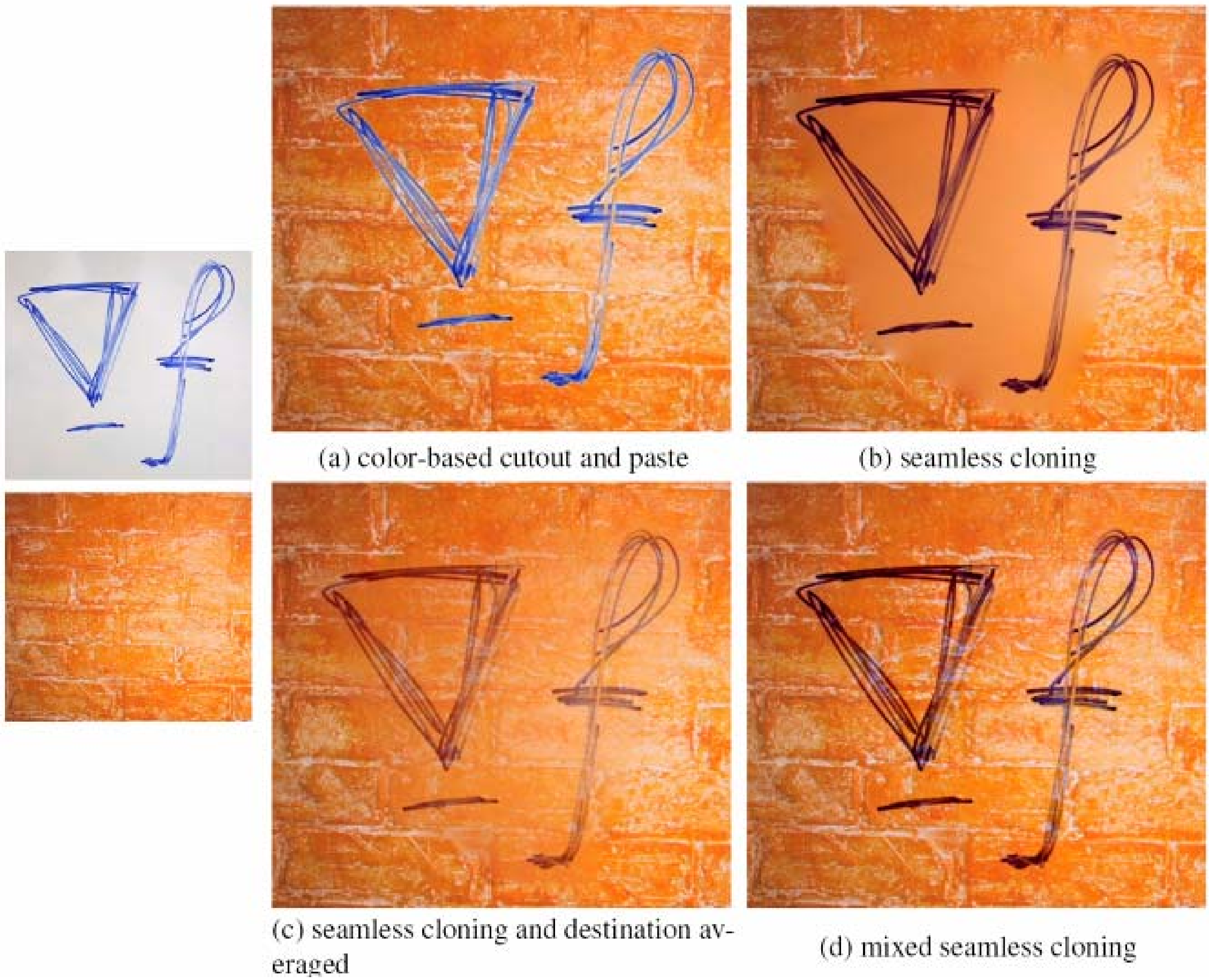
- Two Variants
  - **v** averaged from source and destination gradients ⇒ transparency
  - Select stronger one from source and destination gradients:

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

Discretization:

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

Mixing Gradients Results



(a) color-based cutout and paste

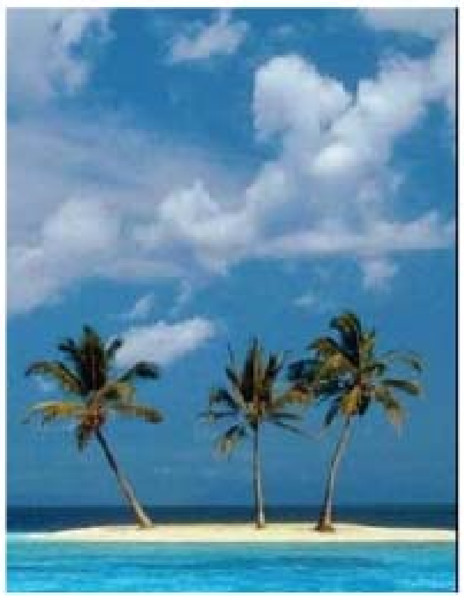
(b) seamless cloning

(c) seamless cloning and destination averaged

(d) mixed seamless cloning

Mixing Gradients Results

source



destination

Mixing Gradients Results

source/destination seamless cloning mixed seamless cloning

Texture Flattening

- Preserve only salient gradients

$$\mathbf{v}(\mathbf{x}) = M(\mathbf{x})\nabla f^*(\mathbf{x})$$

with masking function $M(\mathbf{x})$ so that

$$v_{pq} = \begin{cases} f_p - f_q & \text{if } \overline{pq} \text{ crosses an edge} \\ 0 & \text{otherwise} \end{cases}$$

Texture Flattening

Local Illumination Changes

- Approximate tone mapping transformation after Fattal et al. 2002:

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

$$\alpha = 0.2 |\nabla|_{\text{avg}}$$

$$\beta = 0.2$$

- Attenuating large gradients

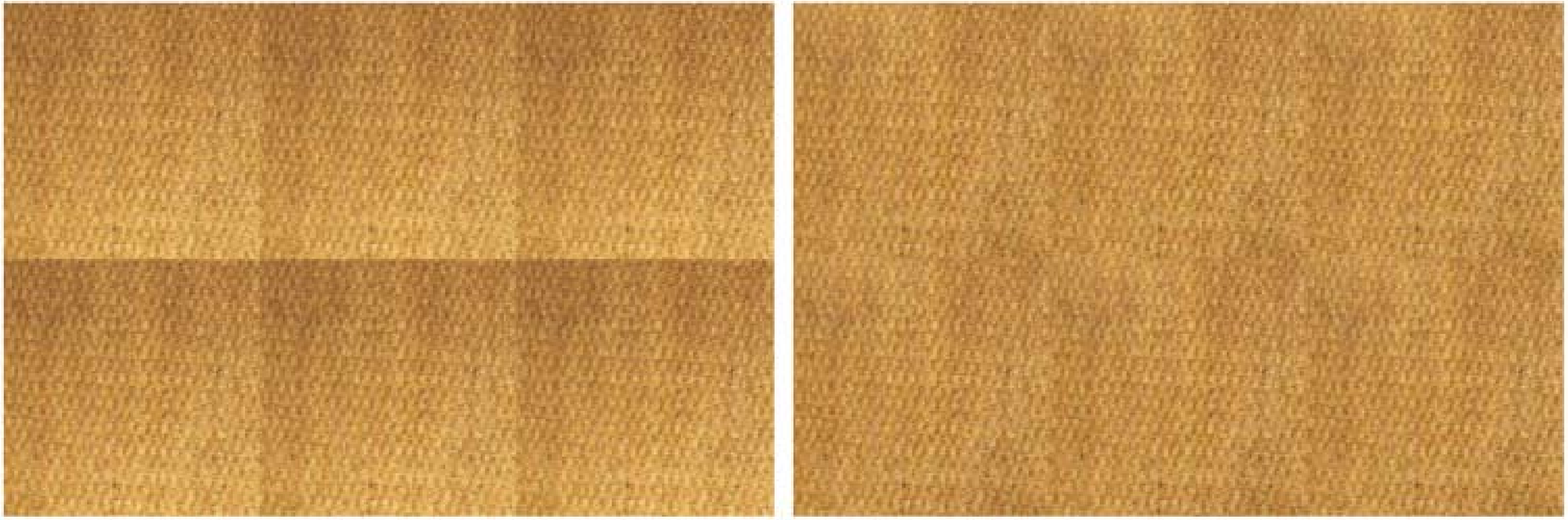
Local Color Changes

- Mix two differently colored version of original image
  - One provides f^* outside
  - One provides g inside

Local Color Changes

Seamless Tiling

- Select original image as g
- Boundary condition:
  - $f^*_{north} = f^*_{south} = 0.5\,(g_{north} + g_{south})$
  - Similarly for the east and west



Discussion

Pros

- Very general framework
- No parameter tuning required
- Method does not require precise selection
- Versatile method
  - Seamless cloning, mixing gradients
  - Texture flatening
  - Local changes of illumination and color
  - Seamless tiling

Discussion

Cons

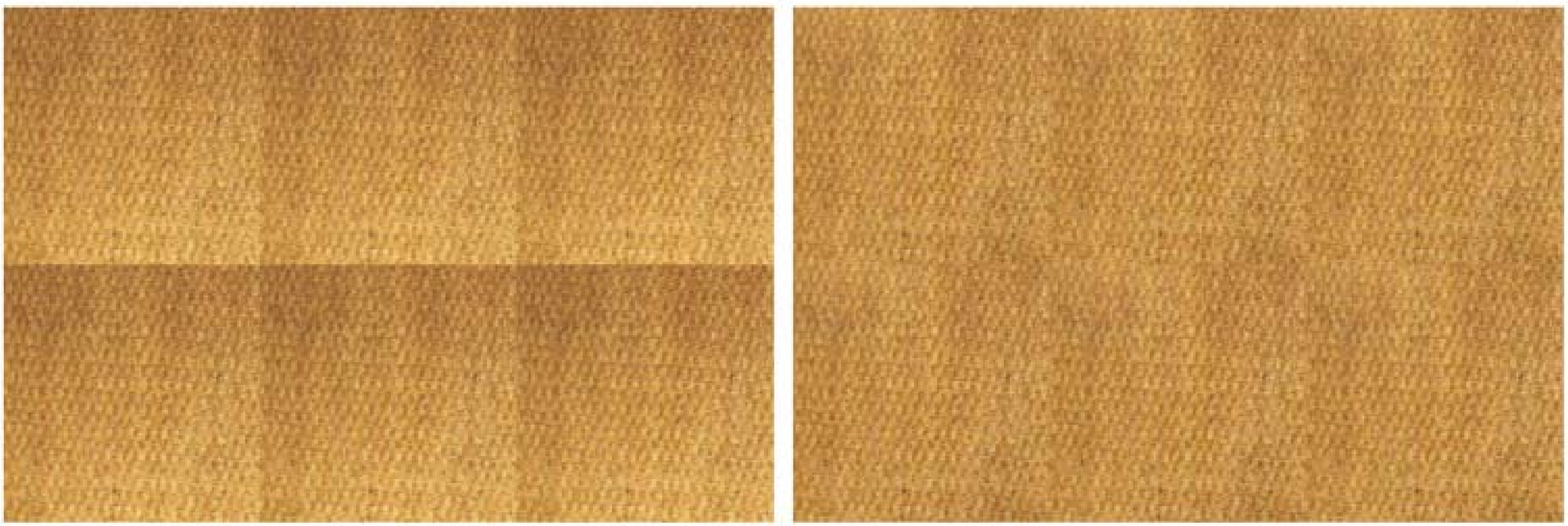
- Cloning requires either of the images to be smooth
- No refined selection is returned
- Minimization only adapts low-frequency content
  - Potential color shift / re-coloring difficult to control
  - Dissatisfactory tiling
  - Cloning requires careful placement of prominent features

Outlook

- More image editing operators
  - Combinations (insert while flattening)
  - Other non-linear operations on gradients
  - More than one source images
- Poisson editing of triangle meshes
  - Feature transfer
  - Detail preserving deformations
- Other editing domains possible?

Thanks

Seamless Tiling



Mixing Gradients Results

source/destination

color transfer

monochrome transfer