Poisson Image Editing

Supervisor :
Thomas Möllenhoff

Implemented By :
Saion Chatterjee
Gaurav Krishna Joshi

Agenda

- Introduction
- Technical Overview
- Functional Implementation
- Optimizing Techniques
- Performance comparison
  - GPU vs. CPU
  - Gauss-Seidel vs. SOR Red-Black Scheme
- Conclusion
- Live Demo

Introduction- The Need

- **Using Classic tools**
  - seams are clearly visible.
  - Can only be partly hidden using classic tools.

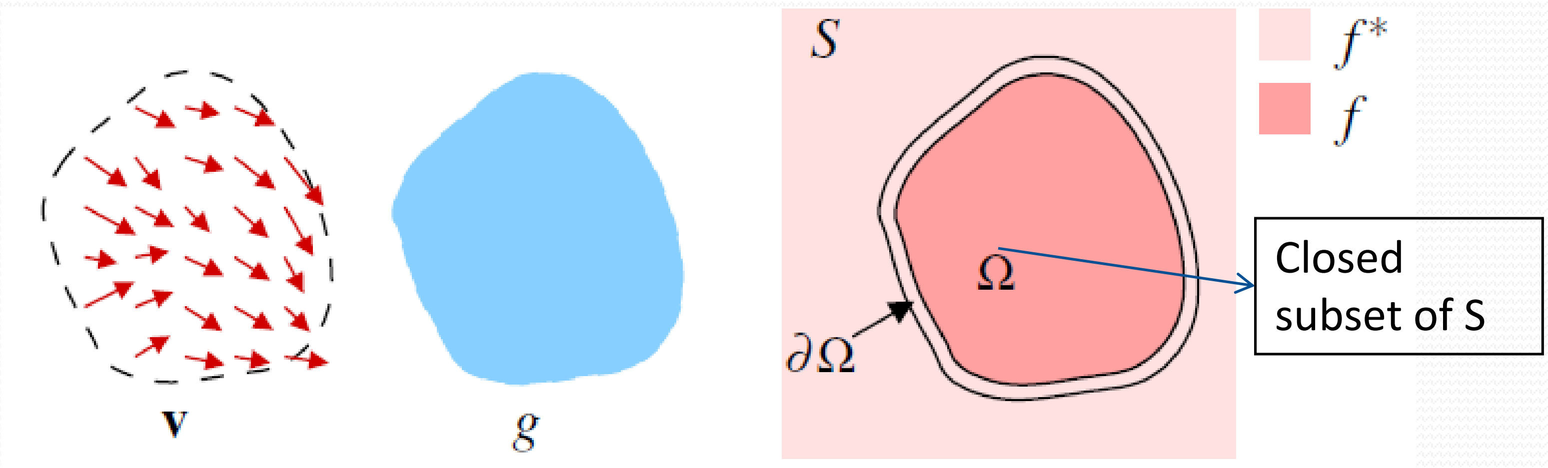
Poisson Image Editing

- A technique for "seamless blending" of one image's selected region into another image.
- Mathematical tool used:-
  - Poisson's equation
    - Laplacian of unknown function over the domain of interest
  - Dirichlet boundary conditions
    - Known function value along the boundary

Poisson Solution to Guided Interpolation

- Guided Interpolation

S: Image domain



Guidance vector field

Gradient field of a source function

f*:Destination function

f: Unknown function

Properties of the Poisson's Equation

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \ \textit{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

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

$$\operatorname{div}\mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$

- Second-order variations extracted by Laplacian operator are the most significant "perceptually"
- Scalar function on a bounded domain is uniquely defined by its values on the boundary and its Laplacian in the interior
  - Poisson equation therefore has a unique solution

With no guidance vector field

- Membrane inter-polant

$$\min_f \iint_\Omega |\nabla f|^2 \ with \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

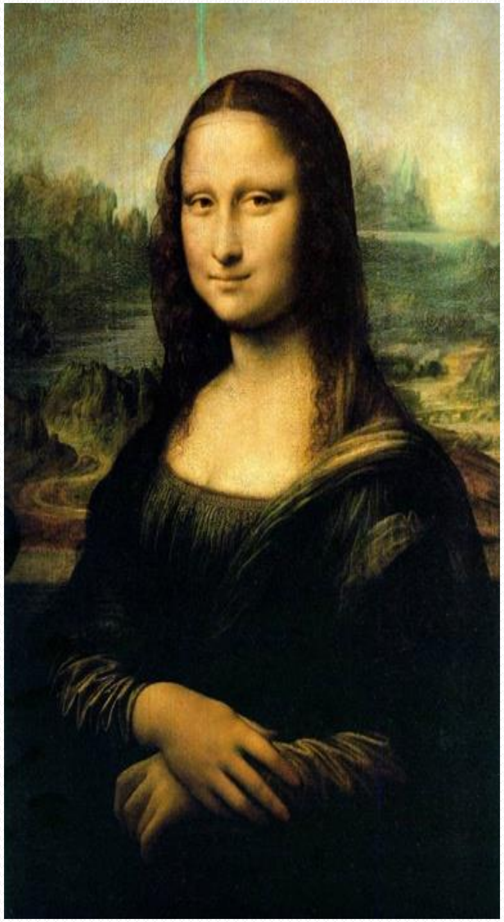
$$\text{gradient operator } \nabla . = \left[\frac{\partial .}{\partial x}, \frac{\partial .}{\partial y} \right]$$

The minimizer must satisfy the associated Euler-Lagrange equation

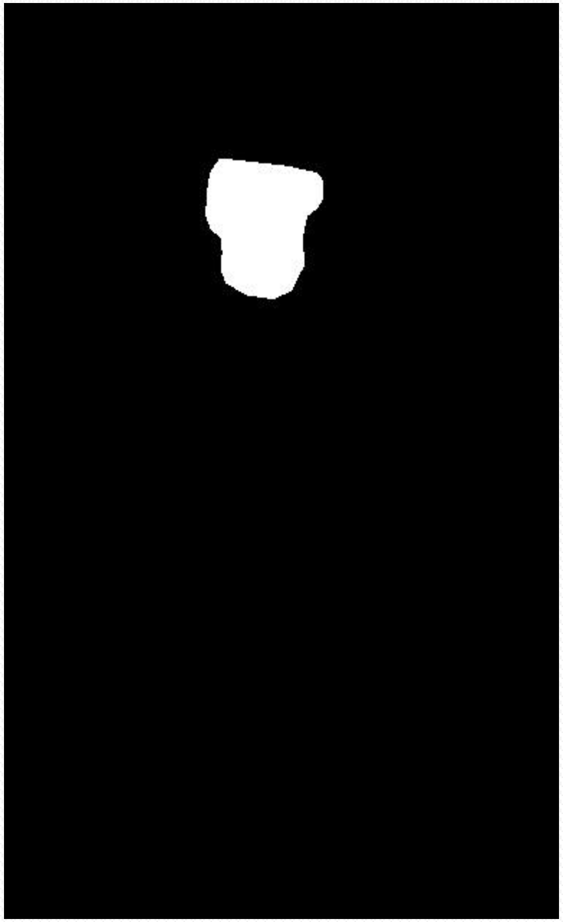
$$\Delta f = 0 \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$$\text{Laplacian operator } \Delta . = \left[\frac{\partial^2 .}{\partial x^2}, \frac{\partial^2 .}{\partial y^2} \right]$$

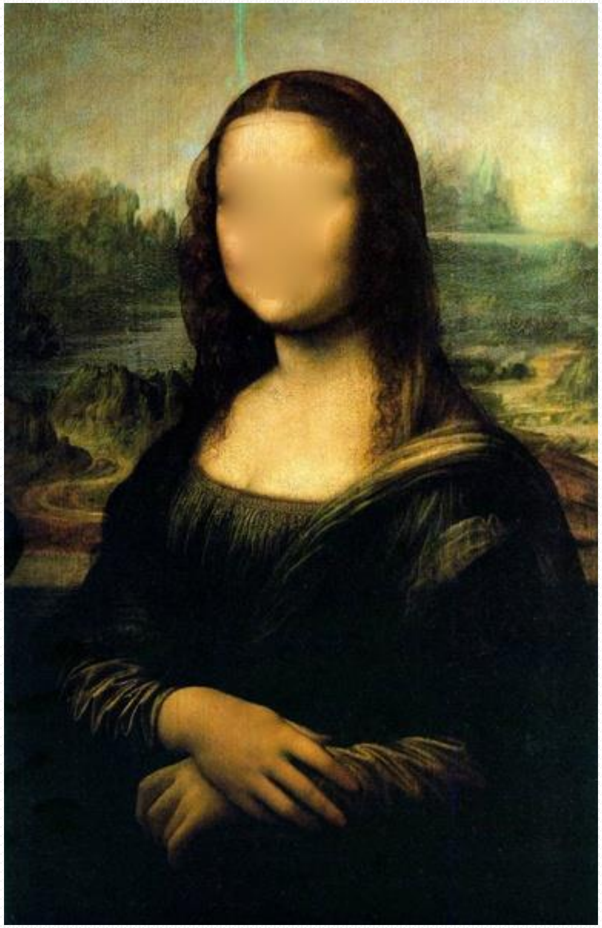
With no guidance vector field



Target



Mask



Output

With source guiding gradient

Source (That's Me)

Target

Output

Discrete Poisson solver

- S , Ω become finite point sets defined on a discrete grid
  - For each pixel p in S,
    Np is its 4-connected neighbors in S
  - The boundary $\partial\Omega = \{ p \in S \setminus \Omega : N_p \cap \Omega \}$
  - Pixel pair $< p, q >, \quad q \in N_p$

The task is to compute the pixel values over the region:

$$f\,|_{\Omega} = \{ f_p, p \in \Omega \}$$

$$\min_{f|_{\Omega}} \sum_{<p,q> \cap \Omega \neq \Phi} \left(f_p - f_q - v_{pq} \right)^2 \textit{ with } \quad f_p = f_p^*, \text{ for all } p \in \partial\Omega$$

Discrete Poisson solver

- Solution: linear equations

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

- For all pixels p interior to Ω , no boundary terms

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

This is an iterative solver which can be solved by Jacobian method.

Seamless Cloning

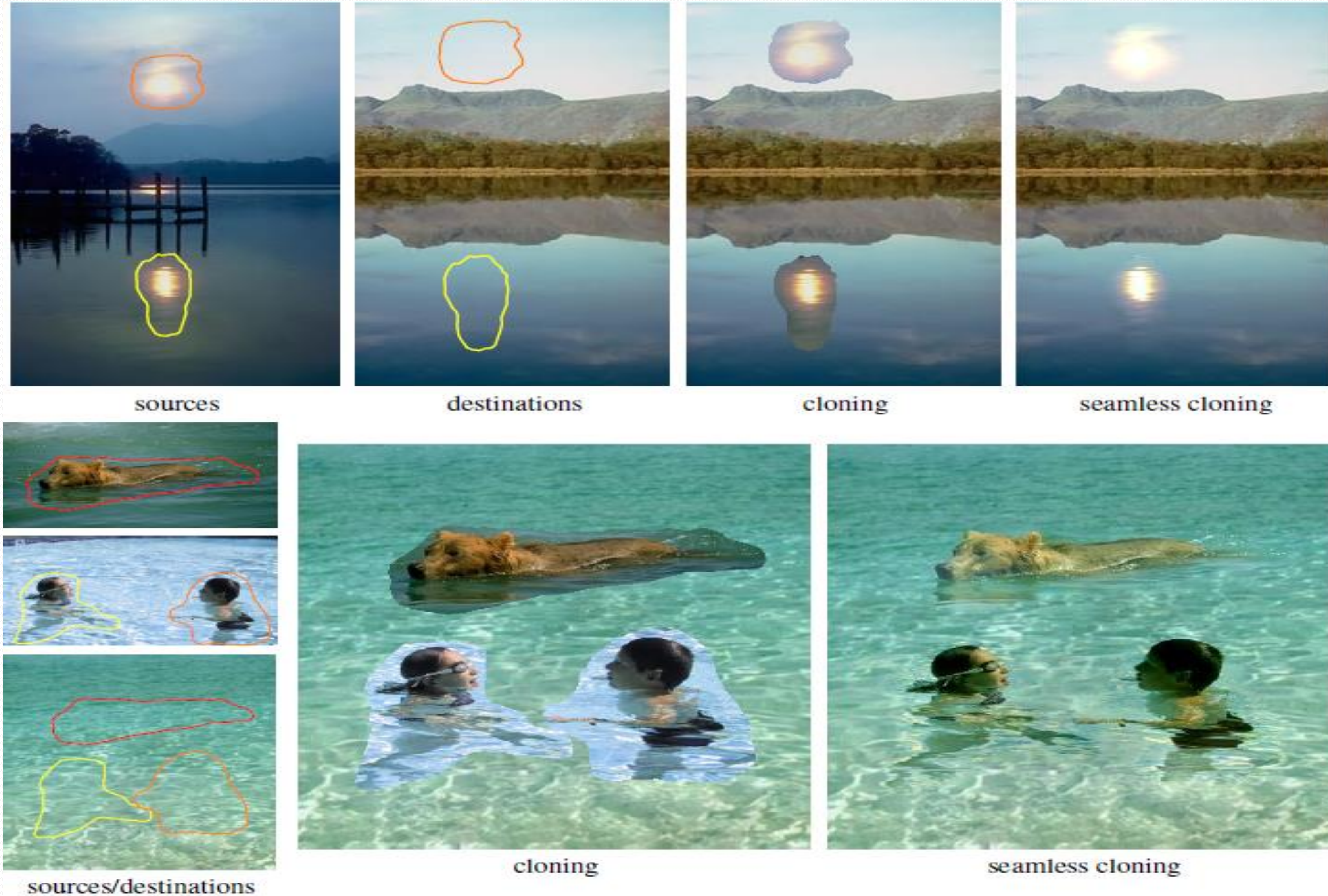
- Importing gradients
  - Gradient field directly from source image g

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

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

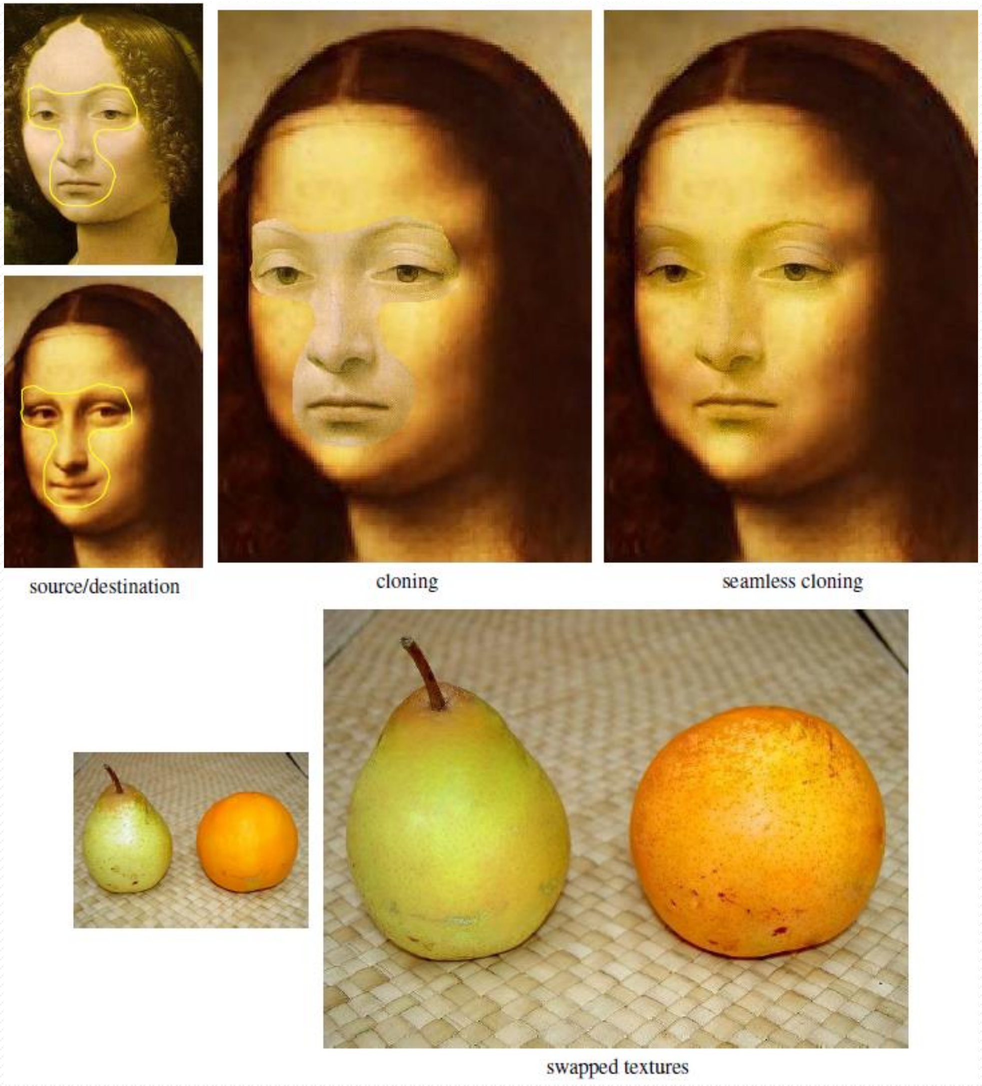
$$\text{for all } \langle p, q \rangle, v_{pq} = g_p - g_q$$

Seamless Cloning

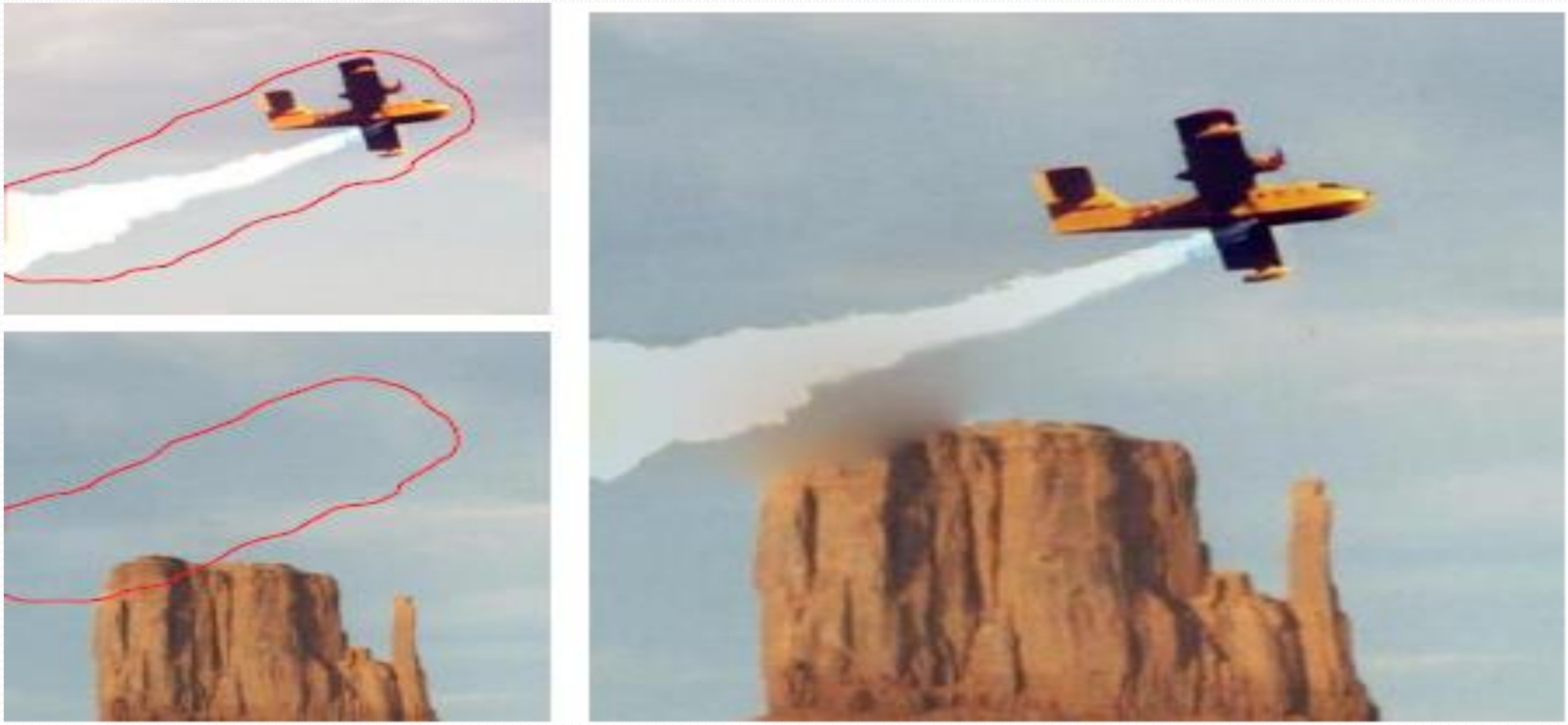


Feature Exchange

- A need to draw precise boundary for specific object.
- Doesn’t sounds that good?



The Problem with Source as guiding gradient



Mixing Gradients

Mixing Gradients

Source Image

Target with different texture

Mixing Gradients

Without mixing gradient

Mixing Gradients

After mixing gradient

Mixing gradients

- Linear combination of source and destination gradient fields

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

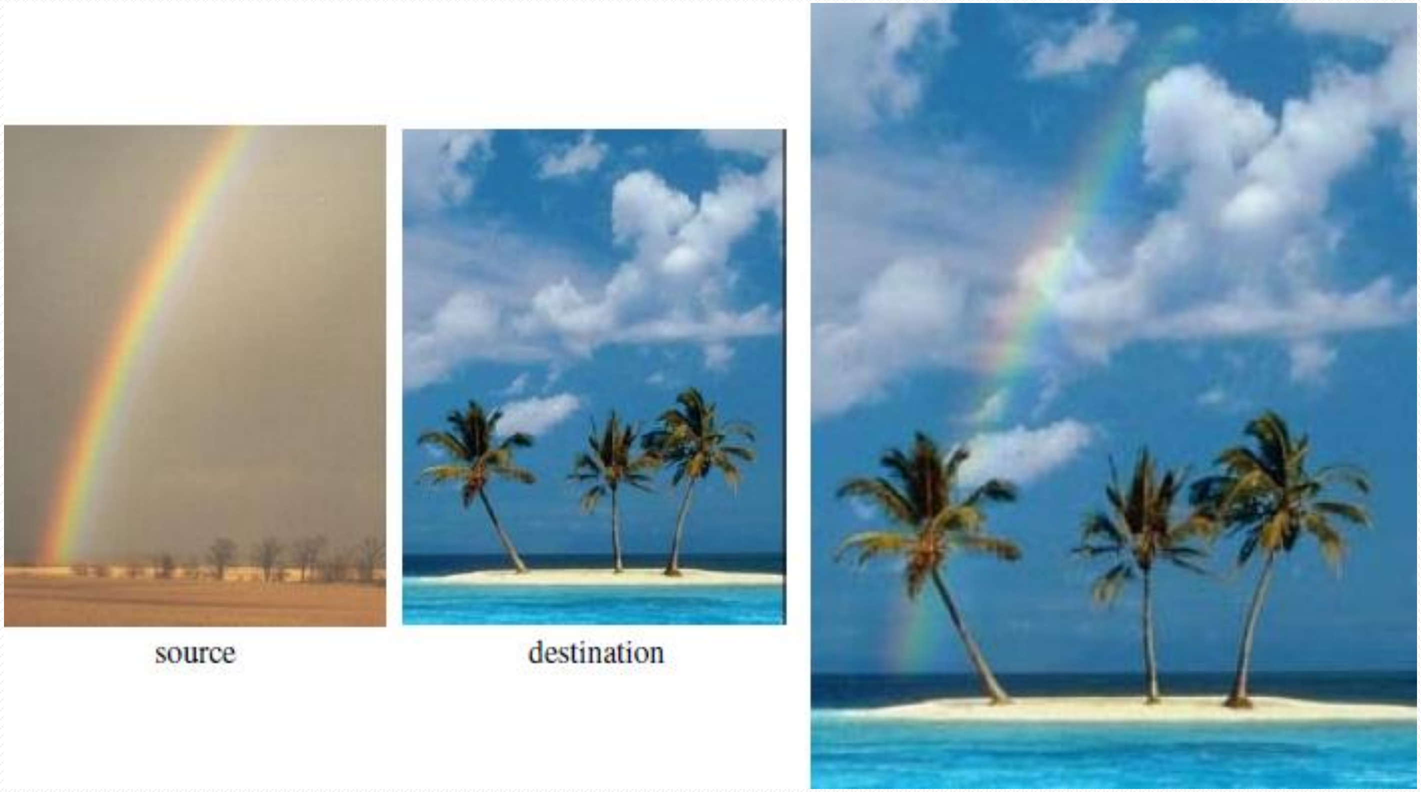
- Non-conservative guidance fields

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

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

Mixing gradients

- Transparent objects



Mixing Gradients

Source Image- Gaurav

The Mask

Mixing Gradients

Target- My Hand ☺

Mixing Gradients

Blended Output

Inserting object with holes

Color-based cut-and-paste

Seamless cloning

Mixed seamless cloning

Functional Implementation

- **ExtractingBoundryPixels**() : Extract the Boundary Pixel from the Mask.
- **Calculate_boundBoxMinMax**() : Extract the Area of Interest in the Target Image.
- **SourceMaskImageMergeinTargetImage**(): Merging the desired portion of the source image to Target Image.
- **Initialize**():Copy the area outside the mask and on Boundry in the Output Image.
- **Evaluate_gradient**(): Compute the gradient of the source image masked region.
- **Poisson_gauss_seidel**(): Blending using Gauss Seidel method.
- **Poisson_sor_redblack**(): Blending using SOR method.
- **Poisson_cpu**() : CPU version of the Poisson implementation

Optimizing Techniques

- Bounding box.
- Shared Memory.
- Registers/local variables
- Different Blocks(32x4,128x4,128x2,64x4).
- Switch case over If else.

Performance Comparison

- CPU vs GPU:

15000 iterations

- CPU run time - 284707.79ms(approx 5 mins)
- GPU run time – 865.193 ms (Gauss-Seidel)(approx 1 sec)

Performance Comparison

- Gauss-Seidel vs. Successive Over Relaxation Red-Black

For 7000 iterations

- Gauss-Seidel- 417.435ms
- SOR- 907.359 ms

Conclusion

- A Generic framework of guided interpolation
- A series of tools to edit in a seamless and effortless manner the contents of an image selection
- The modification can be:
  - Replacement by other images
  - Mixing with other images
  - Alternations of some aspects of the original image, such as texture, illumination or color

Live Demo

Related Work

- Poisson equation has been used extensively in computer vision
  - Rescale gradient field of High Dynamic Range (HDR) image
    - Solving Poisson equation with Neumann boundary
  - Edit image via a sparse set of edge elements
    - Solving a Laplace equation with Dirichlet boundary
  - Spot removing
    - Replace the brightness by harmonic interpolation (solving a Laplace equation)

References

- http://www.cs.jhu.edu/~misha/Fall07/Papers/Perez03.pdf
- http://www.ctralie.com/Teaching/PoissonImageEditing/
- http://www.howardzzh.com/research/poissonImageEditing/

Thank You
For your Attention

Any Questions ??