Poisson Image Editing

Team Name : Pixel Army
Team Members : Sairam kolla, Sharfuddin Mohammed

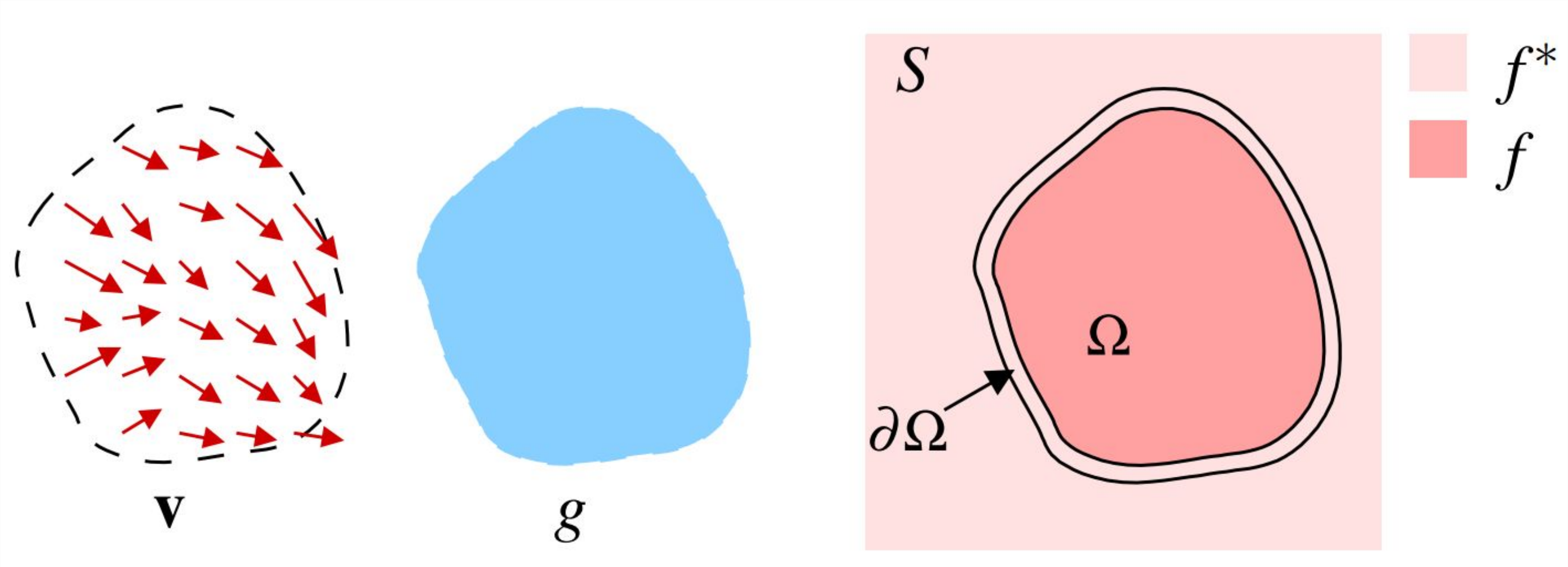
Motivation

- Image editing tasks concerns global changes like color/intensity corrections or image filtering or local changes in a selected region of the image.
- We are interested in achieving local changes in a region manually selected, in a seamless and generic manner.
- Classic tools to achieve this include image filtering and feathering along the seam of the composite image.This results in the seam between the images being only partially hidden.
- Poisson editing allows for seamless image editing and cloning of a selected region.

Introduction

In normal cut and paste algorithms,naive methods like feathering are used which do not completely hide the boundaries between the merged images.Also saliency of the composite image is also affected.Hence the paper proposes a stronger method for blending two images.Here we try to minimize the integral of the difference between the gradient of the source Image and a guidance vector field with boundary conditions.The solution to the obtained equation is a poisson equation.

Single gradient Approach



Single Gradient Approach

The composite image must preserve all the sharp features and edges of the original image.So we try to minimize the following integral.

$$min \iint_{\Omega} |\nabla I(x,y) - \nabla S(x,y)|^2 \, dA$$

The solution to the minimization problem is a poisson equation :

$$\Delta I = \Delta S$$

Also to preserve continuity the intensity values on the boundaries of the composite image must be equal to those of the target image.So:

$$I|_{d\Omega} = T|_{d\Omega}$$

Single Gradient Approach

In discrete domain, this equation translates to a system of linear equations such that:

$$Y = \begin{bmatrix} 0 & 1 & 0 & 0 \cdots -4 & \ldots 0 & 1 & \ldots \\ 0 & 0 & 1 & 0 \cdots -4 & \ldots 0 & 1 & \ldots \\ 0 & 1 & 0 & 0 \cdots -4 & \ldots 0 & 1 & \ldots \\ \vdots & & & & & & \\ 0 & 1 & 0 & 0 \cdots -4 & \ldots 0 & 1 & \ldots \end{bmatrix} \begin{bmatrix} I_{1,1} \\ I_{1,2} \\ I_{1,3} \\ \vdots \\ I_{M,N} \end{bmatrix}_{MxN}$$

Here A is a sparse matrix so unknown can be solved easily.

The seamless cloning tool thus obtained ensures the compliance of source and destination boundaries

Solution to the poisson equation

$$\min_{f} \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega},$$

In the discrete domain the equations translates to:

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

So the solution to the poisson equation is

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

Seamless cloning with single gradient

sources destinations cloning seamless cloning

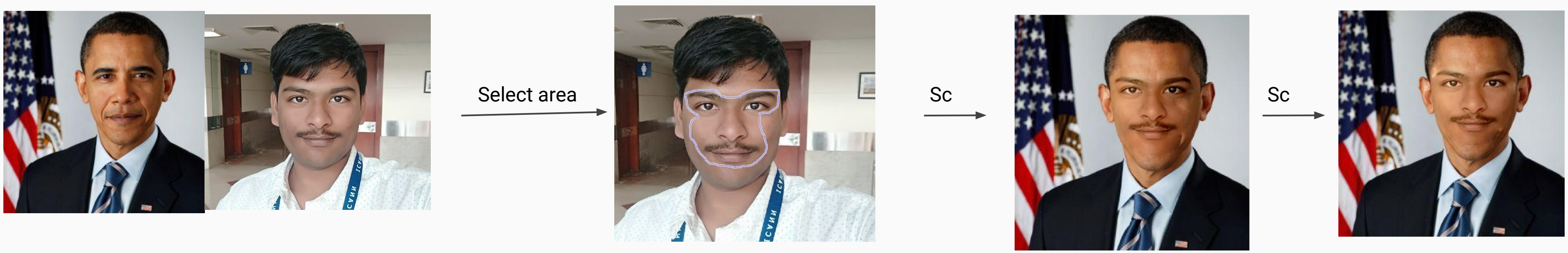
Seamless cloning with single gradient

Inputs

Select area

Seamless cloning

Another Example



Inputs

Feature Exchange

swapped textures

Guided Interpolation

In single gradient approach we only reduced the mismatch between the gradient source image and the gradient Composite image in the Region of interest so we minimize:

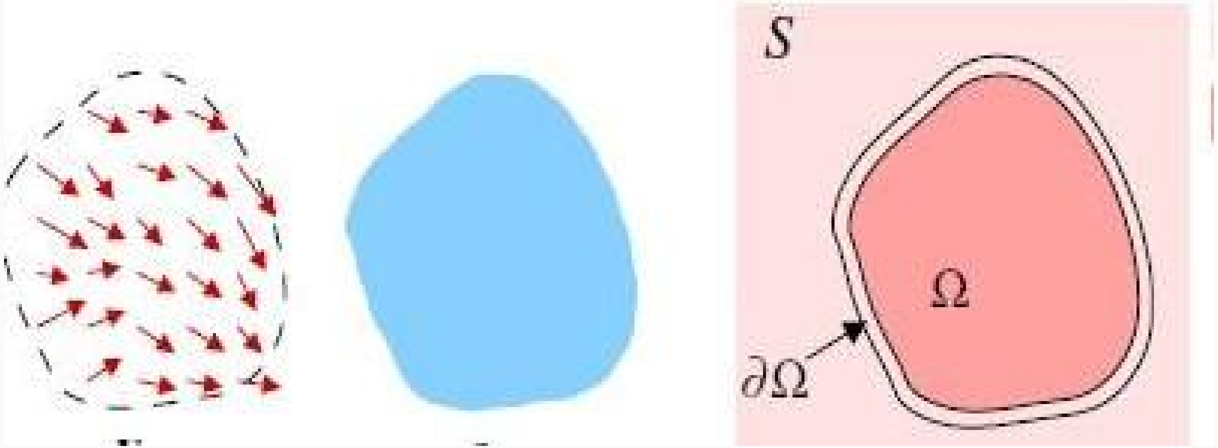
$$min \iint_{\Omega} |\nabla I(x,y) - \vec{v}|^2 \, dA$$

This is the general poisson equation:

$$\Delta I = \vec{\nabla} . \vec{v}$$

Depending on how the composite image should look like we choose,the vector field.

Guided Interpolation

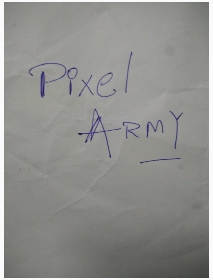


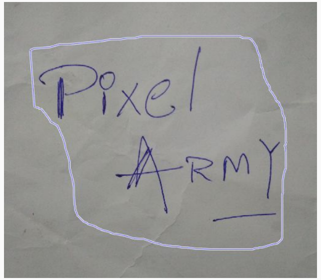
Mixing Gradients

One possible approach is to define the guidance field v as a linear combination of source and destination gradient fields.Consider the vector field:

Here Edges and sharp features of both the Source image and the target image are preserved reducing the mismatch in the region of interest.

Inserting Objects with holes

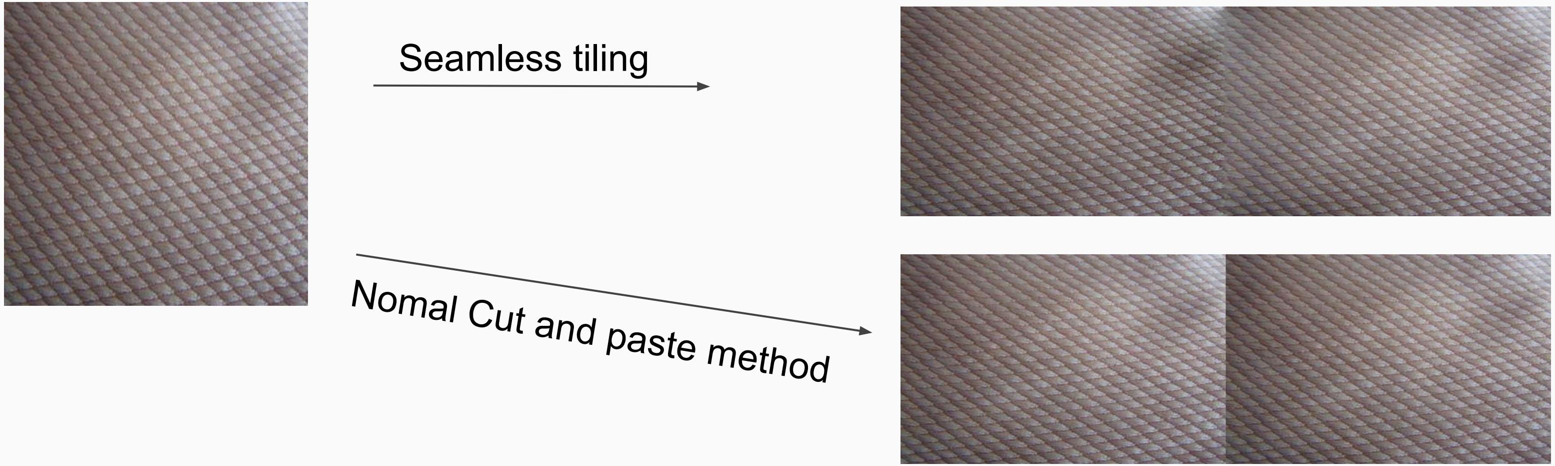




SC with mixed gradient

Sc with single gradient

Seamless tiling



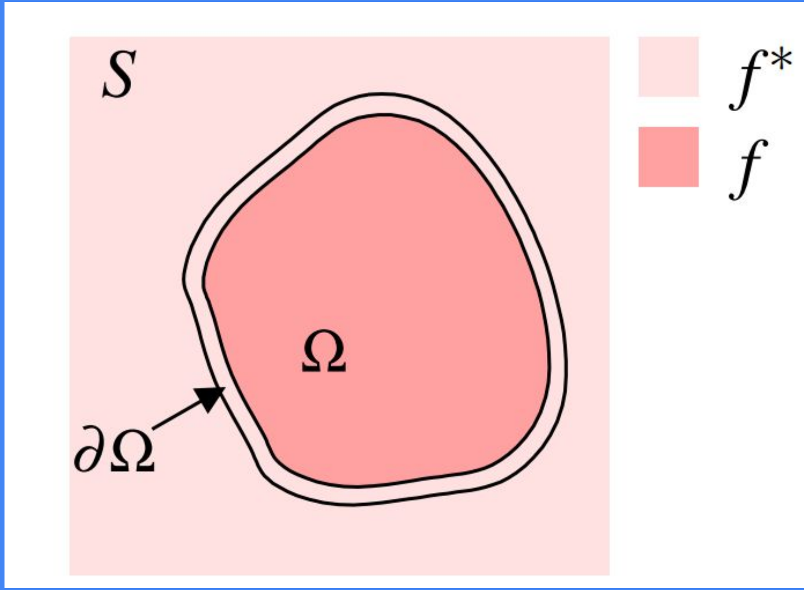
- Average the gradient at the borders and retain the gradient inside the image
- Perform Seamless cloning

Texture Flattening

Here we flatten the texture in the border while we retain the texture at the border

Endless possibilities!!

We just need to carefully construct the gradient image.



Reference paper

http://www.cs.jhu.edu/~misha/Fall07/Papers/Perez03.pdf

Thanks!