INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Digital Image Processing

Monsoon 2018

Poisson Image Editing

Submitted by
Manish Rao (201530228) Jyotish P (20161217)

Contents

1 Introduction

Image editing tasks can be global or local operations. In this project, we will try to explore local image editing operations that are performed by solving Poisson differential equations. The extent of changes ranges from slight distortions to complete replacement of the image content. The classical methods for such image editing operations involve applying filters to a confined selection for slight changes and interactive cut-and-paste with cloning tools for complete replacements. But these methods result in visible seams in the selected regions. The seams are usually applied with feathering which partly hides them.

This project tries to implement a generic machinery from which different tools for seamless editing and cloning of a selected region can be derived. At the heart of this generic method involves solving Poisson differential equation with Dirichlet boundary conditions. The advantage with this method is that we get the Laplacian of an unknown function over the domain of interest along with the unknown function values.

2 Motivation

According to psychologists, the slow gradients of intensity can be superimposed on an image with a barely noticeable effect. The second-order variation in intensity are perceptually most significant. Laplacian suppresses the first order variations and extracts the second order variations. Now, consider an image blending case. We have a source, destination and mask. A patch of pixels on the target should be replaced by some values (this is the unknown function) such that the modified destination image looks natural. The known function is rest of the destination image. We will pose this as a Poisson differential equation for filling the values of known function and boundary conditions being derived from the pixel values on the boundary of the patch.

3 Poisson Solution to Guided Interpolation

3.1 Patch filling

Let's first try to solve for the interpolation of a patch using Poisson equation.

S = closed subset of R^2

Ω = domain of the destination image

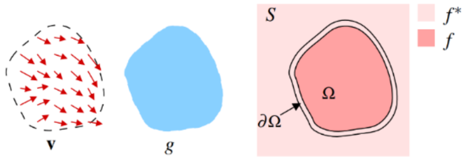
*f** = known function over S-Ω

f = an unknown function in the interior of Ω

∂Ω = boundary of Ω

g = gradient of the source image

v = vector field defined over Ω and it may or may not be same as *g*



Our goal is to fill the f such that the gradient from the boundary of the patch to the values in f is as smooth as possible. The smoothness of the gradient can be obtained from,

$$min \iint_{\Omega} ||\nabla f||^2$$

This expression minimizes the values of the gradients in the patch. But we also need the pixel values in the patch to match the values of the target image at the boundary. So, the boundary constraints will be,

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

Let's try to see what happens when we just want to fill the values of f given that the boundary values for $f*$ are the same (constant color at the boundary). If the values of f are the same color as on boundary, the gradient values will be zero and also the cost function above gives the minimum value. Now, consider a case where the pixel values on the boundary are high on one side and low on the other. Then the cost function tries to achieve a smooth transition from the intensity values at one side to the values on the other side.

Now, we will try to solve for the above cost function. From Euler-Lagrangian equation,

$$J = \int F(x, f, f_x)dx$$

and for J to be stationary, $\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f_x} = 0$.

In our case, $F = ||\nabla f||^2 = (f_x^2 + f_y^2)$. Hence we can write it as,

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

$$\frac{\partial F}{\partial f} = 0, \quad \frac{d}{dx}\frac{\partial F}{\partial f_x} = \frac{d}{dx}2f_x = 2\frac{\partial^2 f}{\partial x^2}$$

Finally,

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \Delta f = 0$$

Now, let's repose our initial minisation problem. The Euler-Lagrangian equation, $argmin \iint\limits_{\Omega} |\nabla f|^2 \quad s.t.\ f|_{\partial\Omega} = f^*|_{\partial\Omega}$ is minimized when $\Delta f = 0 \ \ over \ \ \Omega \ \ s.t. \ \ f|_{\partial\Omega} = f^*|_{\partial\Omega}$.

This works for the continuous domain. But our images are discrete. So, we need to discretise the functions.

$$\Delta f = 0 \ \ over \ \ s.t. \ \ f|_{\partial\Omega} = f^*_{\partial\Omega}$$

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial f}{\partial x} \cong f_{x+1,y} - f_{x,y}$$

$$\frac{\partial^2 f}{\partial x^2} \cong f_{x+1,y} + f_{x-1,y}$$

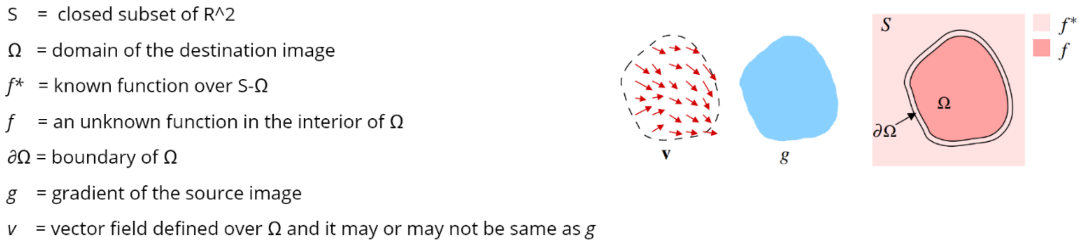
$$\begin{aligned}\Delta f(x,y) &\cong f_{x+1,y} - 2f_{x,y} + f_{x-1,y} - 2f_{x,y} + f_{x,y-1} \\ &= f_{x+1,y} + f_{x-1,y} + f_{x,y+1} + f_{x,y-1} - 4f_{x,y} = 0\end{aligned}$$

These equations can be arranged as a set of linear equations. Let's assume there are n pixels in f. Each pixel in f can be denoted using $x_1, x_2, \ldots, x_n$. If we apply the previous conditions on every pixel, we will get a set of equations which can be arranged into the following matrix form.

$$Au = B$$

A is a sparse matrix containing $(0, 1, -4)$ or $(0, -1, 4)$.
B is a non-zero vector with all values equal to zero except for the edge pixels in this case.
u is what we want to find.

3.2 Patch filling with values from another image



In the previous case, we didn't bother about the vector fields v and g. These gradients from the pixel values of the source image. Now, instead of simple patch filling, we try to blend the gradients from the source image into the target image while having the same cost and boundary constraints.

The cost function can be re-written as,

$$min \iint_{\Omega} |\Delta f - v|^2 \quad with \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

The above cost function can be interpreted as – instead of trying to bring the total change in gradients to the smallest possible value (as in previous case), we will try to bring it as close as possible to the gradients of the source image.

From Euler-Lagrangian equation,

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}$$

$$\Delta f = div\ v$$

$$f_{x+1,y} + f_{x-1,y} + f_{x,y+1} + f_{x,y-1} - 4f_{x,y} = div\ v$$

Now, we need to discretize the functions. The equations can be arranged as a linear set of equations. Let's assume that there are n pixels in f. Each pixel in f can be denoted using $(x_1.x_2, \ldots, x_n)$. If we apply the above conditions on every pixel, we get a set of equations which can be arranged into the following matrix form.

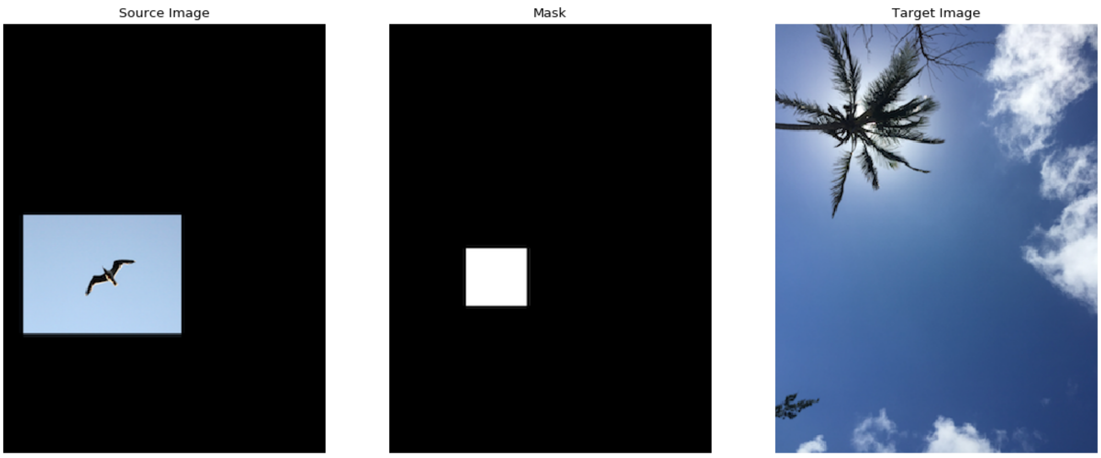
$$Au = B$$

A is a sparse matrix containing $(0, 1, -4)$ or $(0, -1, 4)$.
B is a non-zero vector whose values are given by $div\ v$ in this case.
u is what we want to find.

For some images, using $v = \nabla g$ won't work. In such cases, the maximum of the gradients of source and target images are taken. This is called mixed gradients method.

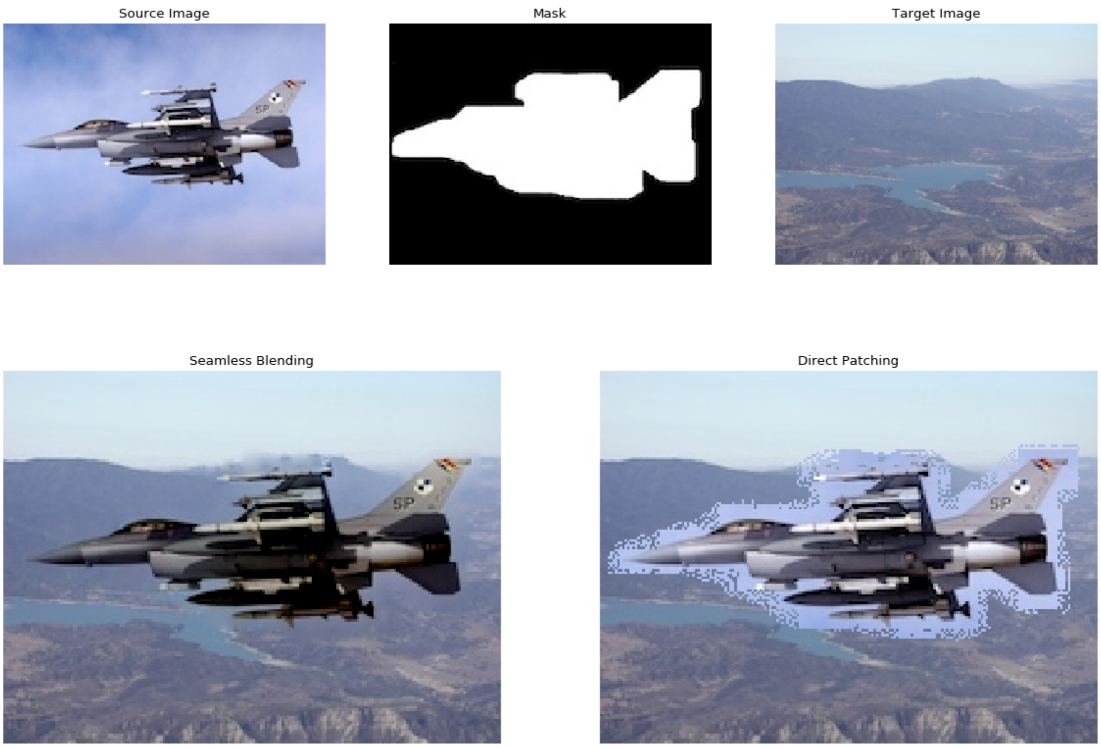
4 Results

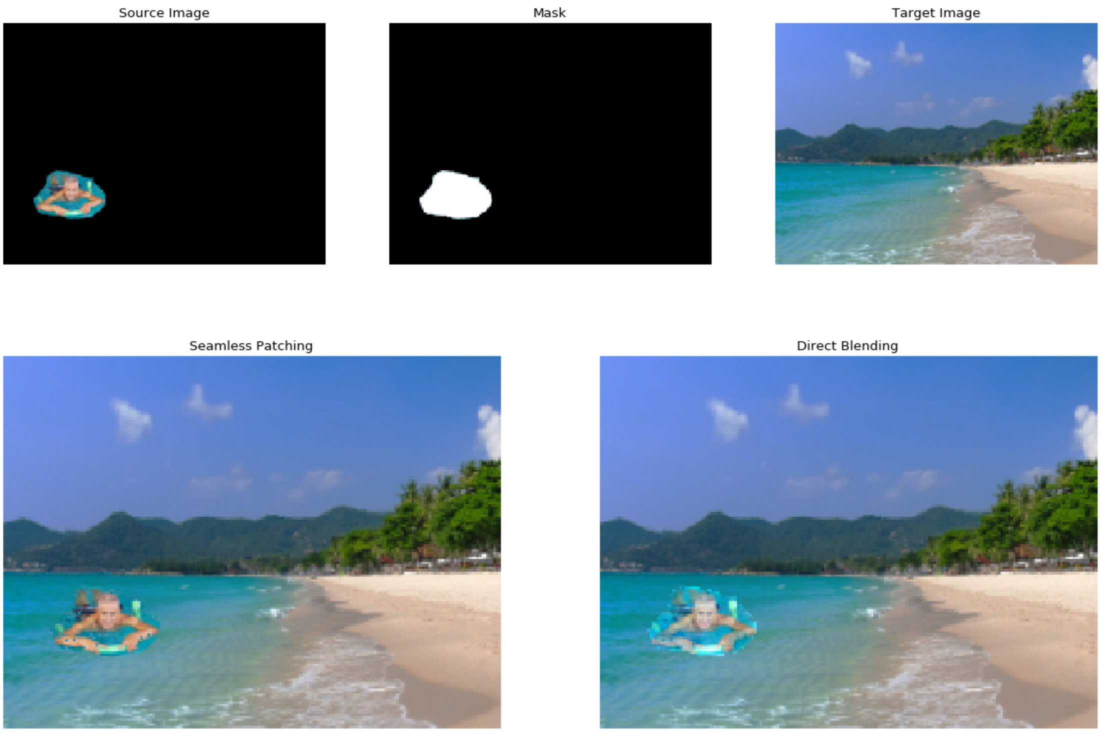
For simple image blending or cloning applications, the vector field v is same as ∇g, where g is the gradient of the source image. The results of the following operation are given below.

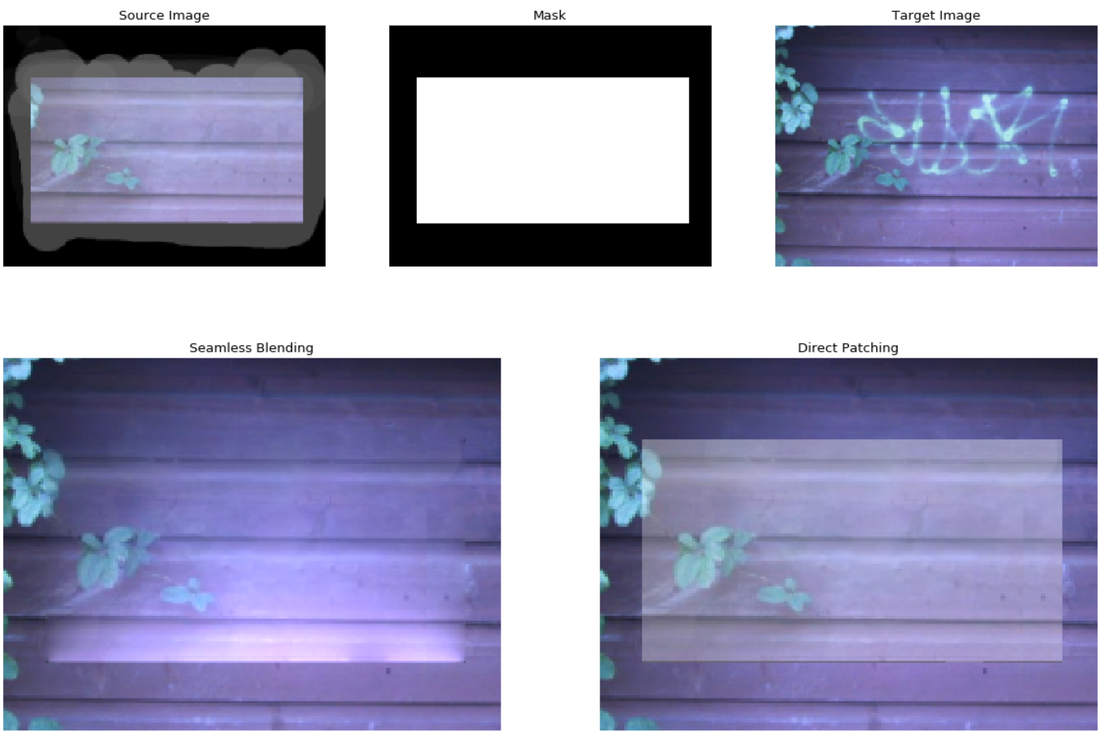


Code and demo notebook are at https://github.com/jyotishp/Poisson-Image-Editor


Seamless Blending
Direct Patching


Source Image
Mask
Target Image
Seamless Blending
Direct Patching


Source Image
Mask
Target Image
Seamless Patching
Direct Blending


Source Image
Mask
Target Image
Seamless Blending
Direct Patching


Source Image
Target Image
Seamless blend
Mixed Gradient Image