Poisson Image Editing

193079014 and 193074015

December 2020

1 Introduction

Here we intend to attain local changes in a given target image (ie, changes confined to small manually selected region) in a seamless and effortless manner.If classic tools like cut and paste or filtering were used, then seams would still be visible.Here we use Poisson's PDE with Drichlet boundary condition to solve this problem. Authors were motivated by the following facts,that second order variations (extracted by Laplacian) are more significant perceptually and scalar function on a bounded domain can be uniquely defined by its values on the boundary and its Laplacian at the interior.So we can solve Poisson equation so as to get unique solution.This same procedure is repeated for all three channels (R,G,B) independently. This method can be used remove or add objects seamlessly, add transparent objects convincingly,add objects with holes ,texture flattening etc.

2 Poisson solution to guide interpolation

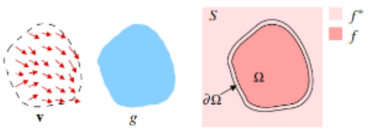

Figure 1.0 Guided Interpolation notations

Here we are dealing with single channel (this can be extended to other channels too).Let's consider target domain ,S to be closed subset of R^2 and Ω be a closed subset of S.Let boundary be defined as $\partial\Omega$. Let f^* be a known function defined over the domain S-Ω and f be unknown scalar function defined over the interior of Ω and $\mathbf{v}$ be a vector field define over Ω. The optimization turns out to be a minimization problem.

$$min\, wrt\, to\, f \int\int_{\Omega} |\nabla f - \mathbf{v}|^2$$

$$with\, f|\partial\Omega = f^*|\partial\Omega$$

Solution of above minimization problem is given by

$$\nabla^2 f = \nabla.\mathbf{v} \tag{1}$$

$$with\ f|\partial\Omega = f^*|\partial\Omega$$

3 Discrete Poisson equation

Above equation both objective and the solution in terms of Laplacian operator (ie Poisson equation) can be discretized and solved. Let S denotes all pixels in image. For each pixel pϵS, let N_p be set of four connected neighbors which are in S, this can be denoted as pixel pair $< p, q >$ such that qϵN_p. The boundary of Ω is $\partial\Omega$ ={ p $\epsilon S - \Omega$: N$_p \cap$ $\Omega \neq$ ϕ } and f_p is the value of f at p, our task is to compute f$|\Omega$={f$_p, p\epsilon\Omega$ }.On discretizing the above optimization problem and the solution of the corresponding discrete optimization problem ,will result in the following simultaneous linear equations .For all p $\epsilon\Omega$

$$|N_p|f_p\text{-} \sum_{q\epsilon N_p \cap \Omega} f_q \text{ - } \sum_{q\epsilon N_p \cap \partial\Omega} f^*_q \text{–} \sum_{q\epsilon N_p} v_{pq} = 0 \tag{2}$$

Here $|N_p|$ is the number of neighbours of pixel p (in our case we had taken it to be 4). For N such pixels inside Ω, equation (2) can give rise to N linear equations ,in matix notation as Ax=b. A is a symmetric and positive definite matrix. Solving these equations by inverting A matrix can be computationally inefficient and requires large memory for large regions of Ω, but since the matrix is of symmetric and positive definite, we can use iterative solvers, ie gradient descent or fast convergent versions of it.

4 Choice of guidance field

Usually the gradient of the source image is taken as the guidance field. If g is source image

$$\mathbf{v} = \nabla g \tag{3}$$

So equation(4) becomes

$$\nabla^2 f \ = \nabla^2 g \quad over\ \Omega, with \quad f|\partial\Omega = f^*|\partial\Omega \tag{4}$$

If we follow this notion, then

$$v_{pq} = g_p - g_q \ \ for\ \ all < p, q > \tag{5}$$

Equation 5 can be plugged into equation 2

5 Mixing Gradients

At times we'll have to incorporate information about target image inside Ω. For example, when we have a source that has got holes in it or when it's transparent or when the source is placed on top of textured or cluttered background. Image 4 and 5 is exactly this case. To get good editing , we tend to take the stronger of the gradient of , either f^* or g as $\mathbf{v}$ in Ω

6 Test results

6.1 Insertion of new elements to target image

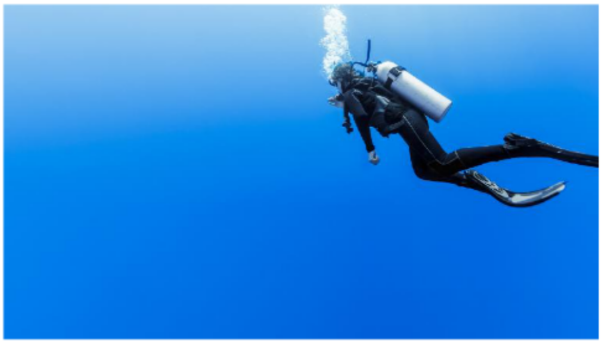

Figure 1.a Target Image

Figure 1.b Source Image

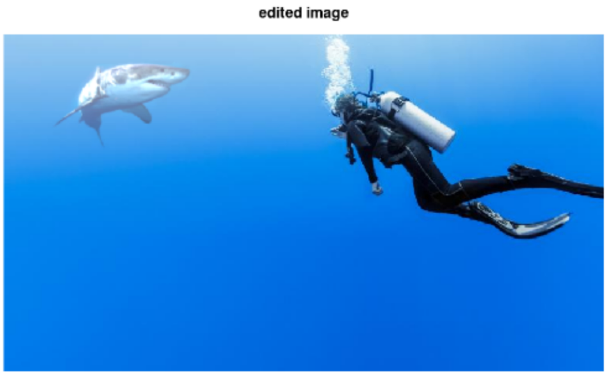

Figure 1.c Edited Image

Insertion of an object to target image seems pretty reasonable,there are no visible seems and we didn't take much effort other than coding the algorithm and locating the source on target

Figure 2.a Target Image

Figure 2.b Source Image

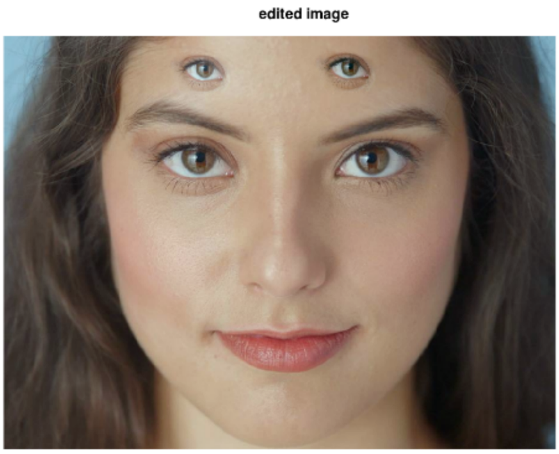

Figure 2.c Edited Image

This looks pretty realistic, there are no visible seams

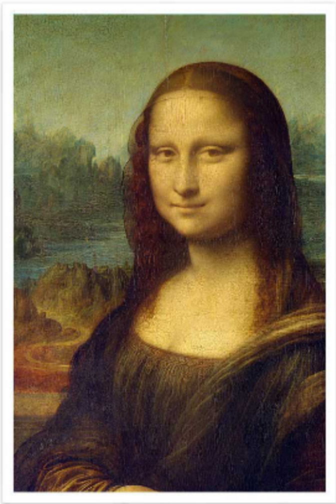

Figure 3.a Target Image

Figure 3.b Source Image

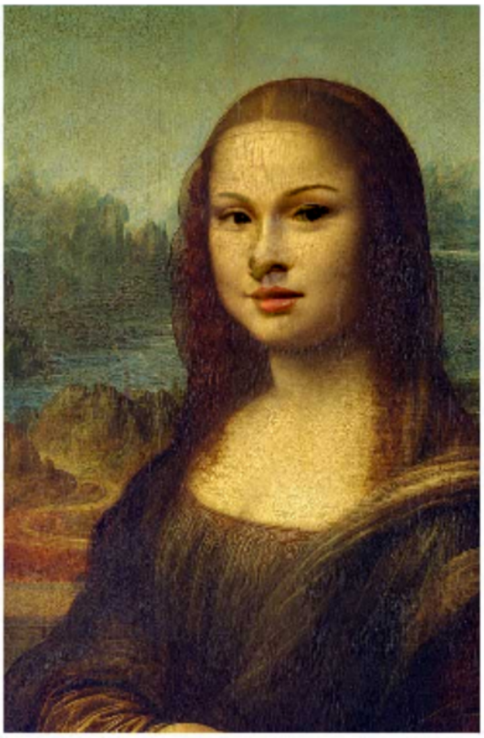

Figure 3.c Edited Image

That looks cool,the texture,color etc is matched between target and source, only problem is in the nose region that is due to improper alignment between source face and target face.

6.2 Insertion of transparent objects to target image

Figure 4.a Target Image

Figure 4.b Source Image

Figure 4.c Edited Image

Here the source image have holes present in it ,we only wanted to copy the signature on the wooden plank and we did it.

Figure 5.a Target Image

Figure 5.b Source Image

Figure 5.c Edited Image

Here transparent source ,ie rainbow is added to the target image (island sky).We used the mixing gradient technique.

6.3 Removal of objects in scene

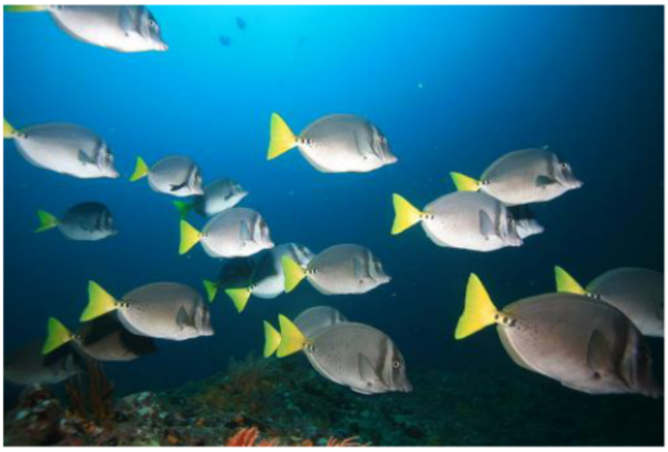

Figure 6.a Target Image

Figure 6.b Source Image

Figure 6.c Edited Image

Here a fish in the target image is removed with any marks.

7 References

Patrick Perez, Micheal Gangnet, Andrew Blake. Poisson Image Editing