Image processing
Coursework 4: Raluca-Cristina Cocioban

1. **Poisson Image Editing by Patrick Perez, Michel Gangnet, Andrew Blake, Microsoft Research UK**

Task 1 a. Paper Summary

Introduction

The paper introduces novel sets of tools used for seamless editing of manually selected regions (concerned with local changes in an image rather than global changes). The technique behind these tools involves interpolation based on solving Poisson equations. To be more specific it uses Poisson partial differential equation with Dirichlet boundary conditions. This will lead to the Laplacian of an unknown function over the area of interest as well as its unknown values along the boundary of the domain/area. The reason for approaching seamless editing tasks in this manner is because it is known that a function is uniquely defined by its values on the boundary and its Laplacian in the interior. Thus, solving the Poisson equation will fill the domain of interest in a seamless manner. For colorful images, it is important to apply the same technique for all the color channels. Alternatively, solving the Poisson equation can be seen as a minimization problem which will be presented in detail in the following section.

The tools introduced in this paper are: **seamless cloning** which means the importation of a region in the source image into a destination region, **mixed seamless cloning** which relies on mixing gradients from the source and destination for a better blending and **appearance modification** of a region in an image (affecting the texture, illumination, colors etc).

Guided interpolation

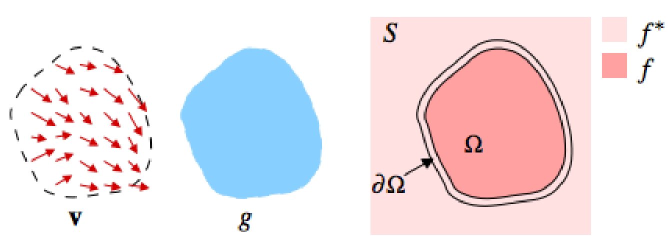


Figure 1: Unknown function f interpolates in domain Ω the destination function f∗, under guidance of vector field v (in the seamless cloning case, v is the gradient field of a source function g).

This paper proposes a guided interpolation framework which means computing the function whose gradient is the closest to some guidance vector field under the boundary conditions. As a consequence, by interpolating f in Ω the destination function f*, it will respect the spacial variations as much as possible leading thus to better results and to more accurate blending.

Notation:

S = image definition domain
Ω = subset of S with boundary ∂Ω (area of interest where changes will take place).
f* = known scalar function defined over S\Ω
f = unknown scalar function inside of Ω (this needs to be found)
v = vector field defined over Ω

In this paper, the interpolation process involves finding the solution for the extended minimization problem which takes into account the guidance field (vector field v):

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad \text{Eq 1}$$

This variational problem has an associated Poisson equation with Dirichlet boundary conditions that needs to be solved. For discrete images, the problem relies on the underlying discrete pixel grid. Having said this, the optimization problem is presented in equation (2) and the idea is to minimise its cost. In practice, this boils down to solving the simultaneous linear equations obtained from (3).

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

where v_{pq} is the projection of $\mathbf{v}(\frac{p+q}{2})$ on the oriented edge $[p,q]$.

$$\text{for all } p \in \Omega, \quad |N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}. \quad \text{Eq 3}$$

The unknowns here are the values of the intensities of the pixels in the region Ω (so the values of each $f_p \in \Omega$).Here N_p are the 4-neighbours of each pixel p. $|N_p|$ is the number of neighbours of p that belong to Ω, v_{qp} is the gradient of the source image g at point p (approximated by $g_p - g_q$ in seamless cloning).

Applications of solving Poisson equation

1. **Seamless Cloning**

This can be used to delete undesirable image features or to insert elements from the source image into the destination image. The tool ensures better results due to the fact that it works with the gradients themselves. It therefore can insert objects with complex outlines into the new background of the destination image. In some situations, we need to transfer just parts of the source image (case of texture transfer between pictures). In this case, we need to transfer just the intensity pattern from the source image; problem solved by turning the source monochrome before applying the algorithm. Fig. 1.

Fig 1: Monochrome transfer : dealing with textures

However, this simple seamless cloning method cannot deal with complex insertions of objects; especially when the destination image contain pronounced textures. Therefore, another method needs to be introduced to solve the case when source and destination have drastic differences.

1.1 Mixed Seamless Cloning

Sometimes, it is necessary to keep some features of the destination image when performing cloning. Cases when its better to mix the gradients of the source and destination images are mostly when the objects that we need to insert contain holes, are slightly transparent and the background of the destination is highly textured.

The solution for this problem involves using mixed gradients (variation of guidance vector). At each point in Ω, we keep the stronger of the variations in f∗ (destination) or in g (source), using the following guidance field:

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

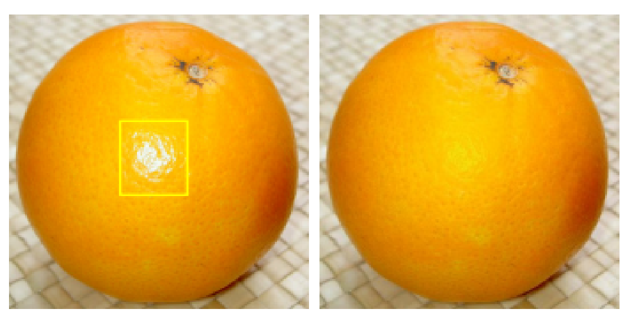
As mentioned before, this technique facilitates the transfer of slightly transparent objects or when adding an object close to another one in the destination picture (in this case, it needs more precise gradient selection to delimit the edges between the objects).

2. **<u>Appeareance change for selected region</u>**

It is possible to perform in-place image transformations by using a guidance field based only on the original image (it can be seen as source and destination coincides). Therefore, it facilitates texture flattening, illumination changes, background or foreground color modifications, and seamless tiling.

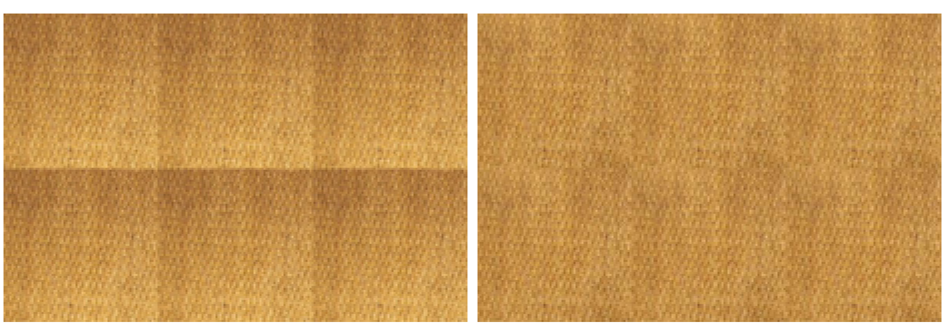
<u>2.1 Texture flattening:</u> involves non-linear modifications of the original gradient field in the selected area. Texture flattening can be achieved by keeping only the gradients around edges, before using the Poisson solver. The effect is then accomplished by washing out small details (content is flattened) and preserving the original structure.

<u>2.2 Local illumination changes</u>: covers applications such as correcting under-exposed objects or reducing specular reflections. This is achieved by transforming the gradient field inside the selected area before integrating back with Poisson solver.



2.3 Local color changes: involves using two differently colored images and mixing them together in a seamless manner to produce color changes. One version of the image will provide the function that deals with the outside area of Ω whereas the other version will provide the source function g that needs to be modified inside of Ω and give some new color results. In general, the technique keeps the object of interest unchanged and turns everything else monochrome. Poisson method provides an easier way of achieving this without the need of precisely selecting the object of interest. In the following figure, the left part represents the original image with the object of interest being selected, the middle part is the same image with the background decolourised (set g to the original color image and $f*$ to the luminance of g) and the right image shows the end result where the local colors are modified by multiplying the RGB channels of the selected area in the original image.

2.4 Seamless tiling: relies on modifications outside the domain in the original image. This will provide new boundary conditions allowing the tiling effect to occur. As an example, we can set periodic boundary values on the border of a rectangular region before integrating with the Poisson solver.



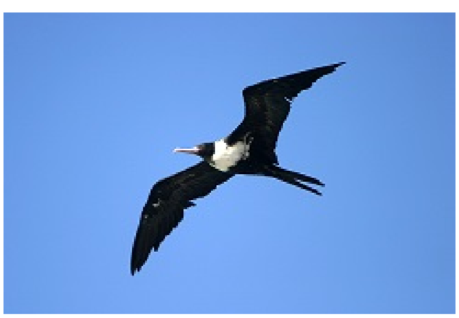
Conclusion

The paper introduced a variety of tools used in seamless and effortless editing of selected regions of images. Important applications of using guided interpolation with Poisson solver include cloning used for adding or removing objects seamlessly, mixed cloning which allows dealing with transparent or complex objects and modifying the appearance of an image within a restricted domain (color, illumination, texture, tiling effect of rectangle images). It is also possible to combine the cloning facilities with the local changes in the domain. The major advantage of this Poisson method is that the area of interest does not need to be precisely selected since the algorithm knows to automatically blend the source/destination based on the boundary information.

Task 1 b: General observations. Cases of failure

At the first iteration, the results are very surprising and it seems like Poisson editing is a stable and robust approach (especially for seamless cloning). However, going through the paper many times and analysing the algorithm, some weak points can be identified. The cases are are very likely to fail represent situations when the source and the destination images are very different in terms of texture.

As a general observation, the algorithm seems to perform very well when the background of the selected region in the source and destination pictures is as homogeneous as possible (similar textures). This is mainly because the algorithm can blend its content in the destination image with no problem without needing to use mixed gradients for good results. (as seen in the following pictures):



+

The simple seamless cloning is therefore adequate for pictures with similar features in the selection (no drastic differences). However, it doesn't produce a good result when it needs to add objects close to other objects (the area where these two objects connect becomes blurry when using this simple cloning technique). This is because it doesn't take the strongest gradient variation between source and destination as presented in the following examples:

+

Mixed seamless cloning seems promising especially for adding transparent objects (like rainbows for example) or when it needs to deal with different textures, objects with holes etc. Due to its tendency to adjust the two images, some content from the source becomes too transparent as a consequence of the mixed gradient. The effect is undesired in many cases (in the next example the background of both the source and the target picture have a different structure, illumination conditions and texture leading to visible edges of the source image):

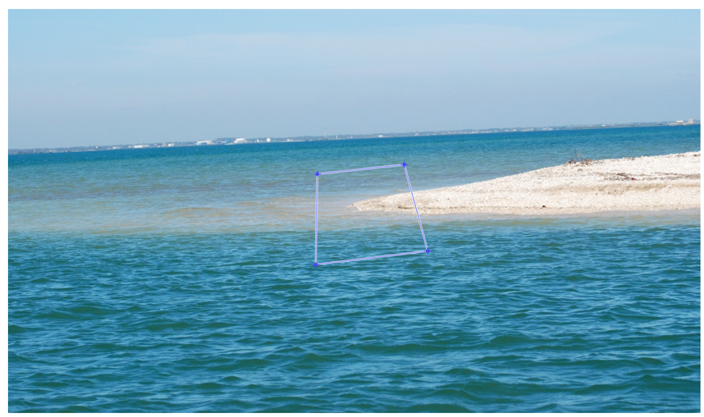
+

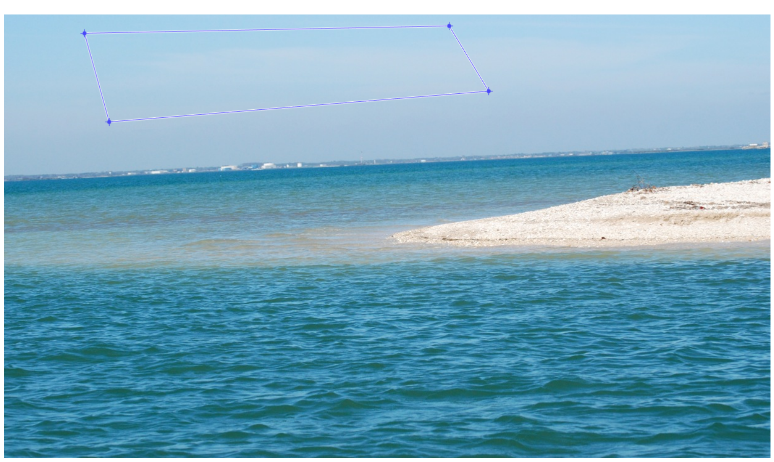
As a conclusion, Poisson based solver used in image editing has proved to be successful in most of the cases. However, artefacts can be identified in both simple cloning and mixed one. Usually, it is likely that the algorithm fails when huge discrepancies exist between source and destination. The main factors that trigger poor results are: the background of both the source and the target picture having extremely different structures, colors, ambient light conditions and textures that cannot be blended.

Task 2: Filling a selected region within an image without guidance vector

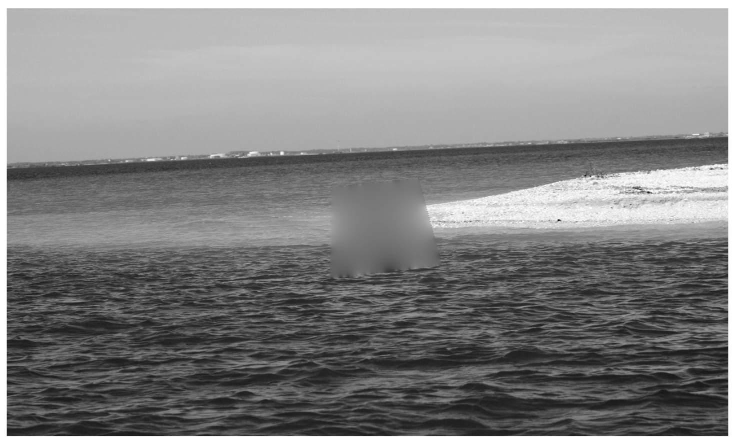
Solving the linear equation system produced by (3) will fill the selected region. In this case, there is no gradient field. This technique works differently in smooth area as opposed to regions with edges. The main reason for this is the fact that it tries to fill the region based on the boundary pixels. This means that in homogeneous parts, it will blur the selected area and whenever it encounters edges (areas of high frequencies) it gets information from that edge and blends it in the selection. The more homogeneous the selected region is, the more undetectable filling occurs. If the edges are prominent (drastic differences), the area of interest becomes more blurry. The following pictures highlight these differences. Also, the algorithm was implemented for both grayscale and RGB images.

Selected regions for smooth and high frequency areas:



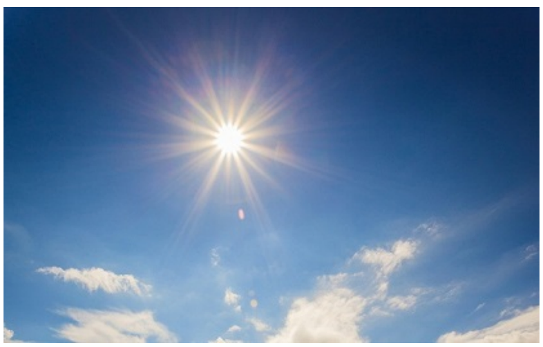


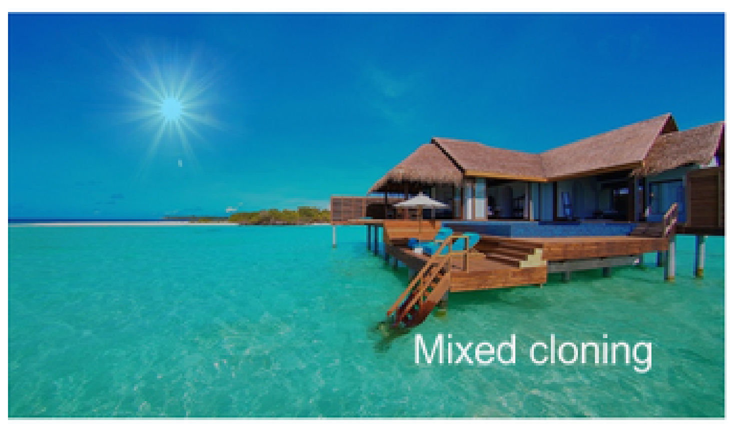

In this case, for the colored image, the filling is almost unnoticeable (due to the homogenous selected area in the sky).

In this case, both images intersect areas with high frequencies and thus it takes some details from the edges and blends them into the interior of the selection.

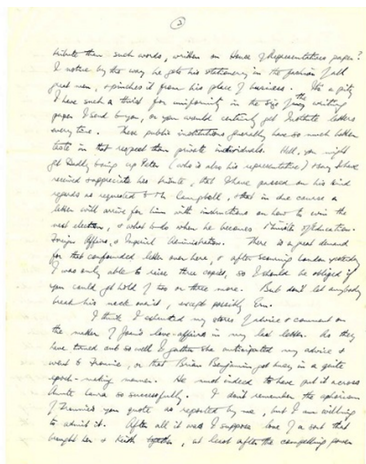

Additional section: Interesting effects. Simple Poisson cloning vs Mixed Poisson Cloning

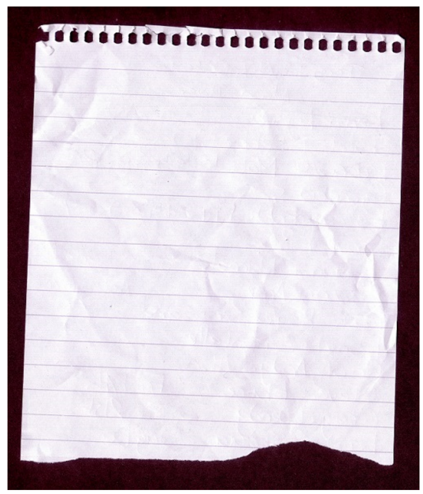


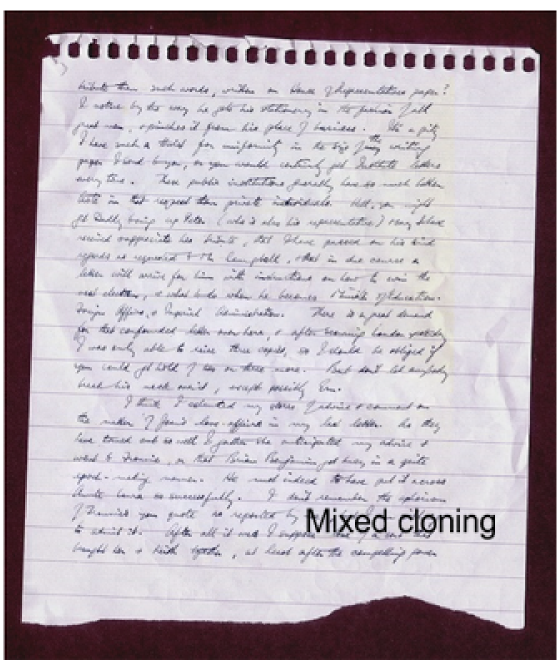


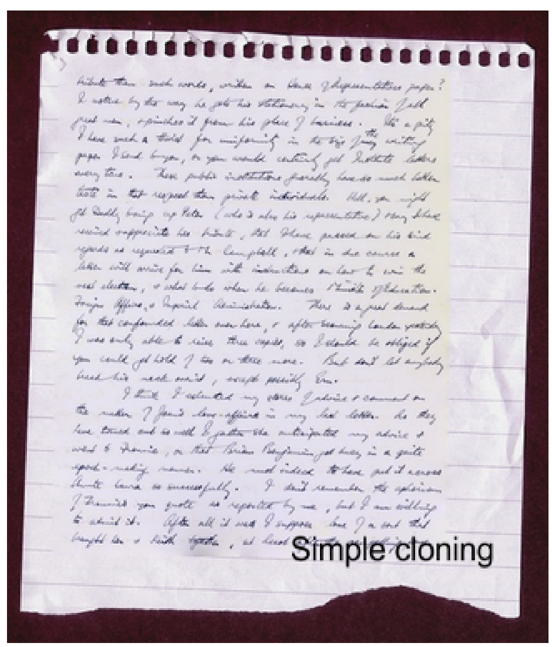

Mixed cloning


Simple cloning





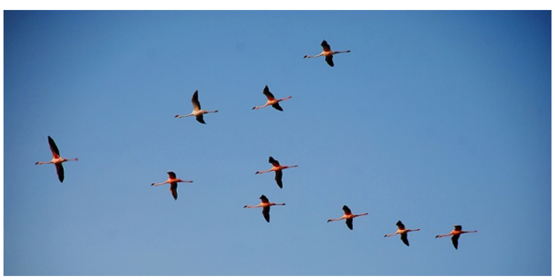

Mixed cloning

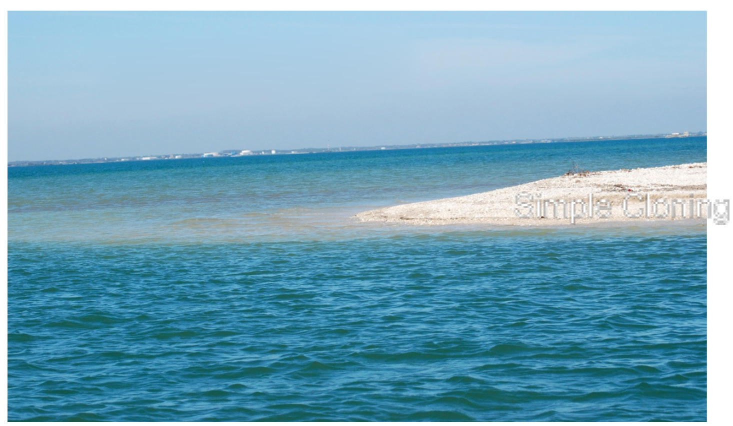

Simple cloning

+

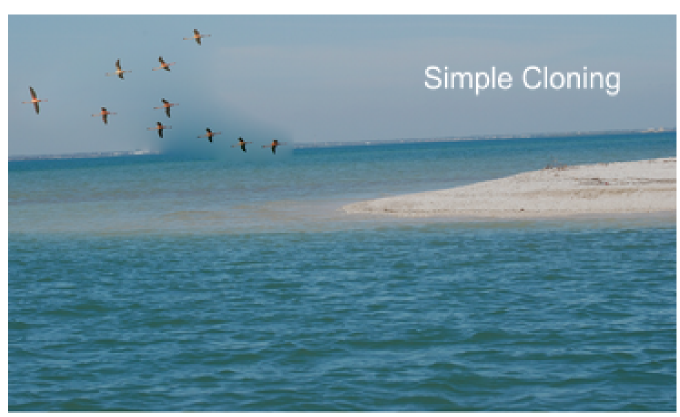

Simple cloning


Mixed cloning




Simple Cloning


Mixed Cloning


Simple Cloning