Introduction to Computer Vision - Final Report

Title: Image Blending

20182010, Elkhan Ismayilzada

GITHUB LINK

Introduction

1) My goal in this project is smooth blending between the pictures. Cropping an object in the image and pasting it on a location of another image is not sufficient due to the difference in the intensities of these images. After pasting it, some techniques should be applied so that the object looks natural in another image.

2) Image composition is an important operation to create a visual content and seamless image blending is one of the necessary methods for image composition. In real life, it is often used in film industry. Usually in movies there is a need to blend the scenes and to make it natural, seamless blending is crucial.

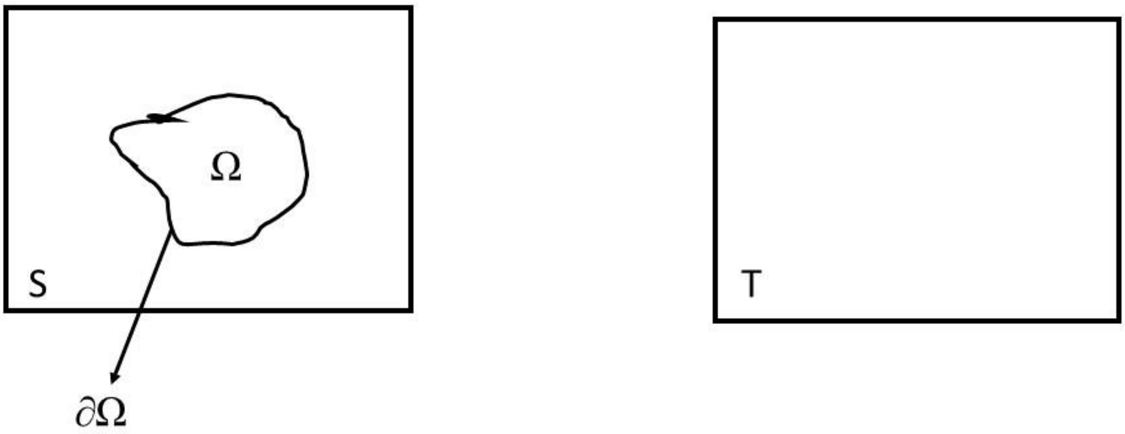
3) Image blending is not an easy problem to solve. This is because the cropped region might not be precisely described. As a result, blending process should not only adjust the appearance of the cropped object to be compatible with the new background but also make the cropping boundary appear seamless.

4) Currently, techniques can be classified into three categories: Laplacian pyramid blending, Gradient domain blending, Exposure compensation. However, the most famous one among them is Poisson image editing [1] which is a technique that belongs to Gradient domain blending. Authors of the paper present two reasons why their approach should work. The first one is that according to well-known psychologists, the second-order variations extracted by Laplacian operator that they use are the most significant perceptually and the second one is that since their function is uniquely defined, it must have a unique solution hence it is a sound algorithm. This is the technique I am going to employ in my project.

Methods

1) The input and output of my program is crystal clear. You will be asked to provide two images: Source and target image. In source image you will crop a region of your wish and in target image you will choose a location to put the cropped region onto. After selection procedure, my program is going to blend the cropped image with target image seamlessly by using the technique I mentioned in **Introduction** section and show you the corresponding result as an image.

2) To describe the method given in paper, let's define following variables first. Suppose we have a source image S, a target image T to blend with, a cropped region Ω, and a boundary of this region $\partial\Omega$ visualized in following picture.



The idea is that we want to reduce color mismatch between source and target by creating composite in gradient domain. More explicitly, we want the gradient of composite inside Ω to look like the source image as close as possible whereas it has to match with gradient of target image on the boundary ∂Ω. In a mathematical notation we can write this as follow:

$$min_{I(x,y)\in\Omega} = \iint_{\Omega} \left|\left|\nabla I(x,y) - \nabla S(x,y)\right|\right|^{2}\ dx\,dy$$

$$such\ that\ I(x,y) = T(x,y)\ on\ \ \partial\Omega$$

And the solution to this problem would be to make the Laplacian of composite be equal to the Laplacian of source image in Ω and make the composite be equal to the target image on ∂Ω. In a mathematical notation:

$$\nabla^2 I(x,y) = \nabla^2 S(x,y) \quad in \quad \Omega$$

$$I(x,y) = T(x,y) \quad on \quad \partial\Omega$$

And this is basically a Poisson Equation.

To discretize and solve this problem we need to consider two cases.

a) For a pixel p inside Ω, there are five unknowns. e.g.

$$\begin{aligned} I(x+1,y) + I(x-1,y) + I(x,y-1) + I(x,y+1) - 4I(x,y) \\ = S(x+1,y) + S(x-1,y) + S(x,y-1) + S(x,y+1) \\ - 4S(x,y) \end{aligned}$$

b) For a pixel p on ∂Ω or neighborhood pixels of p are on ∂Ω, there are 3 unknowns. e.g.

$$\begin{aligned} -4I(x,y) + I(x,y+1) + I(x+1,y) + T(x,y-1) + T(x-1,y) \\ = S(x+1,y) + S(x-1,y) + S(x,y-1) + S(x,y+1) \\ - 4S(x,y) \end{aligned}$$

Notice that right hand side of both equations are same. Therefore, we can represent this for all needed pixels in one linear system as follows.

$$AX = B$$

Where A is a sparse Laplacian matrix where there are mostly zeros (for pixels outside Ω) and either 1 or -4 for the interested pixels, x is the all the unknown pixels and b is S and T pixel values corresponding to it. In MATLAB, after constructing A and the b matrix we can solve for x by

$$X = A\backslash B$$

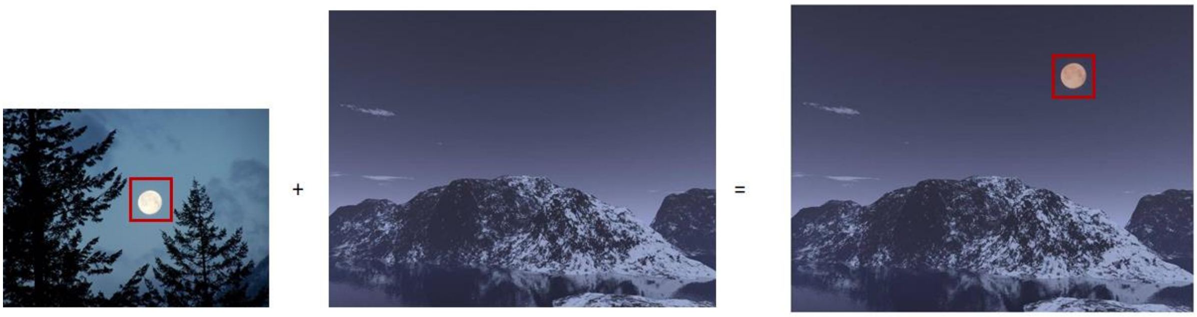
This might not be sufficient, however, if there is a distinct object or figure on the ∂Ω so the paper goes further and tries to mix gradients. In order to do that, there was one example given in the paper as follows.

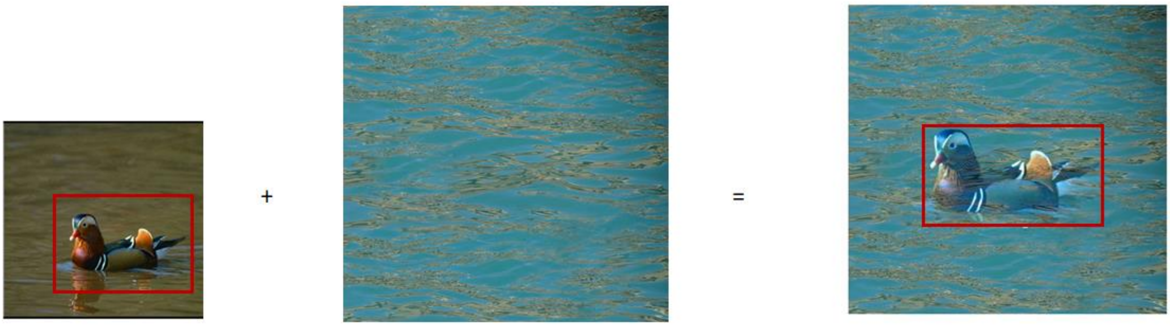
$$\nabla^2 I(x,y) = \begin{cases} T(x,y) & if \quad \|\nabla T(x,y)\| > \|\nabla S(x,y)\| \\ S(x,y) & otherwise \end{cases}$$

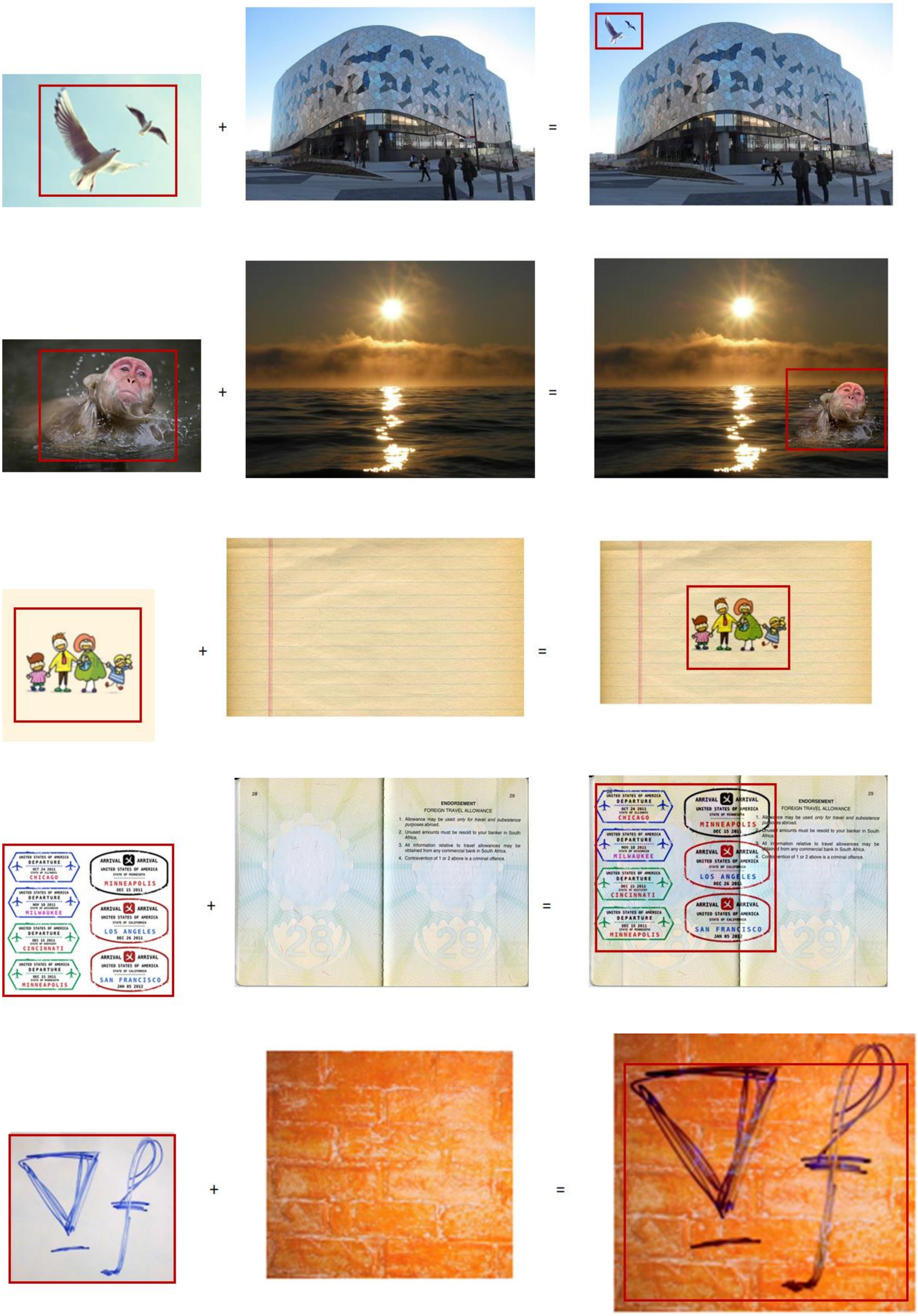
What it means is basically we want to keep distinct edges or features of the target image (gradient is higher in this case).

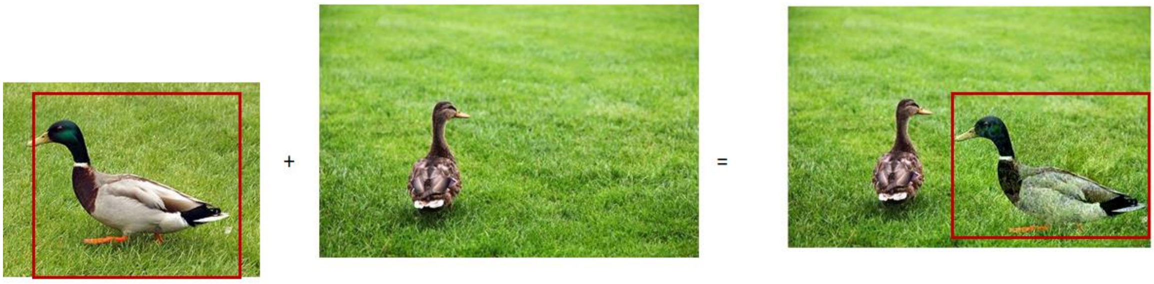
Results

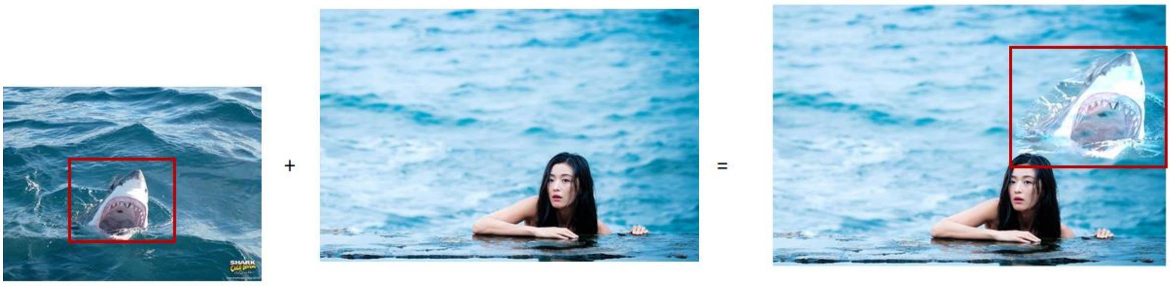
For each figure, there are 3 pictures (Source image, Target image, Blended image from left to right). The cropped region is highlighted in red box. As you can see from the results, it works as expected.

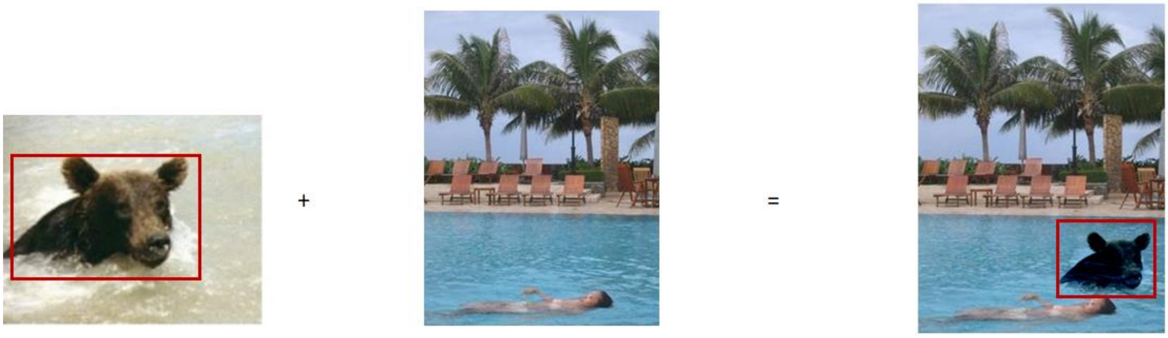












Discussion

As you noticed, in some pictures, the cropped source image looks transparent in the target image. This is because the algorithm reduces the color mismatch between the cropped source image and target image. On the other hand, the algorithm works best when the cropped source image has almost no background. This project helped me understand important equations (e.g., Poisson equation) that are widely used in computer vision.

References

[1] Pérez, P., Gangnet, M., & Blake, A. (2003). Poisson image editing. In ACM SIGGRAPH 2003 Papers (pp. 313-318).