Image restoration using partial differential equations

Introduction

Equations originating from physics have recently founds their way to other areas. One, possibly surprising, application is that of restoring art or images (image restoration, or, image inpainting).[1, 2]

To understand how partial differential equations (PDEs) from physics can help in image restoration, consider some grayscale image of your own choice. Due to graffiti painters, let us assume that a piece is missing (black region). Can we fill in the missing region without any information of what is missing? This may seem like a hopeless task, but partial differential equations (PDEs) are here to help!

PDE-based methods for image restoration are based on propagating the information (typically, intensity values and gradients) at the boundaries of the missing region inwards. The propagation is performed by solving a partial differential equations with specified boundary conditions.

The simplest case of PDE-based image restoration would be to simply use the Laplace equation for inward propagation of intensities:

$$\Delta I(\mathbf{x}) = 0$$

Here, $\mathbf{x} = (x, y)$ labels different pixels in the image, $I(\mathbf{x})$ is the intensities at these pixels and $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$. The equation above is complemented by Dirichlet boundary conditions, i.e., the known intensities at the boundary of the missing region. Intuitively, it might be helpful to recall that the equation above describes the long-time behavior of the concentration of a set of independent random walkers (solution of the diffusion equation), where the concentration of random walkers at the boundaries is kept fixed. Using the solution to Laplace equation is the simplest way of restoring an image and is referred to as *harmonic inpainting* by scientists in the field.

Scientist in the field noticed that, whereas using the method above works well for small missing regions, for larger regions the solution of the diffusion equation yields visually "too blurred" images. To deal with this, several different more sophisticated PDE-based methods has been developed. For instance, one can use an anisotropic diffusion constant in the diffusion equation [3], or use the Navier-Stokes equation from fluid dynamics [4] or other non-linear equations [5] in order in order to incorporate "fluid" flow besides diffusion.

Project description

Basic version

1. Download or create a grayscale image from a photo. A grayscale image is nothing but a matrix of values, and using image software such as Matlab, Python, Image J etc, you can export your image to text-file in the form of a matrix. The matrix has dimension $M \times N$ and the intensity values are $I(x, y)$, where x denotes rows and y denotes columns.

2. Create a mask (dimensions $M \times N$), i.e. a matrix with values 1 or 0, where 0 corresponds to a missing pixel ("graffiti-sprayed" region) and 1 corresponds to a non-missing pixel. The number of missing pixels is n.

3. Implement a code which solves numerically the Laplace equation for the missing region. Denote by $I_{\text{restored}}(x, y)$, the restored image.

4. To quantify how successful the image restoration was introduce a "discrepancy score", χ^2, between the graffiti-sprayed regions from the original image and the restored images. For instance,

$$\chi^2 = \frac{\frac{1}{n}\sum_{x,y}[I_{\text{restored}}(x, y) - I(x, y)]^2}{\sigma^2} \tag{1}$$

   Here, the sum is over the pixels in the missing region and

$$\sigma^2 = \frac{1}{n-1}\sum_{x,y}[I(x, y) - I_{\text{mean}}]^2 \tag{2}$$

   where I_{mean} is the mean of the original data of the missing region. Investigate how χ^2 depends on the size and shape of the missing region. Are some images more difficult than others?

Advanced version (for higher grade)

Try to improve upon the method above. For instance, you could try to include a force term into Laplace equation, use an anisotropic or spatially-varying diffusion constant, change the boundary condition etc. Or, you may want to consult one of the methods available in the literature (see [3]-[6]). Or, even better, come up with some method of your own!

A few instructions

You need to submit your code(s) by e-mail to Tobias, Victor and Jens (tobias.ambjornsson@thep.lu.se, victor.olariu@thep.lu.se, jens.krog@thep.lu.se) before the deadline. The deadline is provided in the schedule, see http://home.thep.lu.se/˜tobias/FYTN03/Schedule.html. Also please attach your input image and a mask image (a representative example mask, if you are testing different ones) to the e-mail together with enough instruction so that we can reproduce your output if needed.

The oral presentations are strictly limited to 15 minutes per group (see schedule for dates). Sign-up sheets will be provided on Tobias's office door (K345) or through Doodle. During the presentation, we expect to see the input image, its "grafitti-painted" counter part(s) and the restored image. If you did only the basic version of the project we expect only one brief "Methods" powerpoint-slide. If you tackled an advanced version of the project we would like a bit more elaboration on your choice of method.

GOOD LUCK!

References

[1] [1] C.B. Schönlieb, Mathematics can make you fly?

[2] [2] I. Peterson, Filling in blanks: Automating the restoration of a pictures missing pieces

[3] [3] Bertalmio, "Processing of flat and non-flat image information on arbitrary manifolds using Partial Differential Equations", PhD Thesis, 2001.

[4] [4] Bertalmio, Marcelo, Andrea L. Bertozzi, and Guillermo Sapiro. "Navier-stokes, fluid dynamics, and image and video inpainting." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.

[5] [5] Bertozzi, Andrea L., Selim Esedoglu, and Alan Gillette. "Inpainting of binary images using the Cahn-Hilliard equation." IEEE Transactions on image processing 16.1 (2007): 285-291.