XX Congreso de Ecuaciones Diferenciales y Aplicaciones
X Congreso de Matemática Aplicada
Sevilla, 24-28 septiembre 2007
(pp. 1–8)

Non-local regularization of the reverse heat equation for digital images enhancement

A. Buades[1], B. Coll[1], J.M. Morel[2]

[1] *Dpt Matematiques i Informatica Univ. Illes Balears*
Ctra Valldemossa km 7.5, 07122 Palma de Mallorca, Spain email: email{vdmiabc4, tomeu.coll}@uib.es.
[2] *Centre de Mathematiques et Leurs Applications, ENS Cachan*
61, Av du President Wilson, 94235 Cachan, France

Palabras clave: filtrado y realce de imágenes, EDP's, problemas inversos.

Resumen

In 1955 Kovasznay et al. proposed to enhance an image by reversing the heat equation. This process is highly unstable and blows up the noise. Thus an efficient deblurring depends crucially on accurate denoising. In this paper we investigate the use of neighborhood filters and of a recent variant, NL-means, to stabilize the reverse heat equation. We shall prove that adding to the heat equation a term involving the self-similarities of the image stabilizes the reverse heat equation. By experiments on good quality images, not artificially blurred, we will illustrate the feasibility of the method to reveal image details. In contrast with the various PDE's for image enhancement, the non-local reverse heat equation thus introduced is linear.

1. Introduction

Let $u(t, \mathbf{x})$ denote the solution of the heat equation

$$\frac{\partial u}{\partial t} = \Delta u, \quad u(0, \mathbf{x}) = u_0(\mathbf{x}).$$

If u_0 is sufficiently smooth, namely C^2 and bounded, then we can write

$$u(t, \mathbf{x}) - u(0, \mathbf{x}) = t\Delta u_0(\mathbf{x}) + o(t). \tag{1}$$

Now the difference between the original u_0 and a blurred version image of it, $k * u_0$, is roughly proportional to its Laplacian. We can write this relation as

$$k_h * u_0(\mathbf{x}) - u_0(\mathbf{x}) = h\Delta u_0(\mathbf{x}) + o(h). \tag{2}$$

for $h \to 0$ and k a positive radial kernel.

The comparison of equations (2) and (1) shows that blurring u_0 with a kernel k_h is for small h equivalent to applying the heat equation to u_0 at scale h. These equations have led to an image restoration method: We read in the paper [10] by Lindenbaum, Fischer, and Bruckstein that Kovasznay and Joseph [7] introduced in 1955 the notion that a slightly blurred image could be deblurred by subtracting a small amount of its Laplacian. Numerically, this amounts to subtracting a fraction λ of the Laplacian of the observed image from itself:

$$u_{\text{restored}} = u_{\text{observed}} - \lambda \Delta u_{\text{observed}}.$$

Dennis Gabor studied this process and determined that the best value of λ was the one that doubled the steepest slope in the image [10]. In other terms one can to some extent enhance an image by reversing time in the heat equation:

$$\frac{\partial u}{\partial t} = -\Delta u, \quad u(0) = u_{observed}.$$

Numerically, this amounts to iterating substraction of its Laplacian from the observed image. This operation can be repeated several times with some small values of h, until it blows up. Indeed, the reverse heat equation is extremely ill-posed.

Since then one can list several attempts to stabilize the time-reverse heat equation or to emulate it by another more stable partial differential equations

The Osher-Rudin "shock filter". The term of shock filter itself was framed by Rudin in his PhD dissertation [13], inspired from the use of nonlinear filters in shock simulation for P.D.E.'s. Osher and Rudin [12] proposed to sharpen a blurred image u_0 by applying the following equation:

$$\frac{\partial u}{\partial t} = -sign(\Delta u)|Du|, \text{ with } u(0, \mathbf{x}) = u_0(\mathbf{x})$$

where Du is the spatial gradient of u at $\mathbf{x}$ and $|Du|$ its euclidean norm $\sqrt{u_x^2 + u_y^2}$. This equation can be seen as a pseudo reverse heat equation, where the propagation term Du is tuned by the sign of the laplacian.

The Kramer Algorithm. In [8], Kramer defined a filter for sharpening blurred images. The filter replaces the gray level value at a point $\mathbf{x}$ by either the minimum or the maximum of the gray level values in a circular neighborhood $B(\mathbf{x}, h)$. This choice depends on which is the closest to the current value. It was proved in [15] that the Kramer and the Osher-Rudin filters share the same asymptotic behavior for regular 1D signal. In other terms, they are infinitesimally identical in 1D. However this is not true in the case of images. In that case, Guichard and Morel [5] proved that the PDE underlying the Kramer filter is

$$\frac{\partial u}{\partial t} = -sign(D^2u(Du, Du))|Du|.$$

Thus, the laplacian in the Rudin-Osher equation is replaced by a directional second derivative of the image, $D^2u(Du, Du)$.

Perona and Malik model Perona and Malik in 1987 [11] proposed to smooth what needs to be smoothed, namely, the irrelevant homogeneous regions, and to deblur the boundaries. With this in mind, the diffusion should look like the heat equation when $|Du|$ is small, but it should act like the inverse heat equation when $|Du|$ is large. Here is an example of a Perona–Malik equation:

$$\frac{\partial u}{\partial t} = \operatorname{div}(g(|Du|)Du), \tag{3}$$

where $g(s) = 1/(1 + \lambda^2 s^2)$. It is easily checked that we have a diffusion equation when $\lambda|Du| \leq 1$ and an inverse diffusion equation when $\lambda|Du| > 1$. Thus the model mixes the heat equation and the reverse heat equation.

Variational deblurring (case where the blurring kernel is known). Another class of attempts covers the so called "variationalrestoration methods. Their principle is to search for a function $u_{restored}$ which, once blurred by the heat equation or another blurring kernel k, gives the original image $u_{observed}$. The Rudin–Osher–Fatemi algorithm [14] is efficient when the observed image u_0 is of the form $k*u+n$, *where k and the statistics of the noise n are known.* Given the observed image u_0, one tries to find a restored version u that minimizes the functional

$$E_{TV} := \int_\Omega \left(|Du(\mathbf{x})| + \lambda(k * u(\mathbf{x}) - u_0(\mathbf{x}))^2\right) \mathrm{d}\mathbf{x}, \tag{4}$$

where Ω denotes the image domain and the parameter λ controls the oscillation in the restored version u. If λ is large, the restored version will closely satisfy the equation $k * u = u_0$, but it may be very oscillatory. If instead λ is small, the solution is smooth but inaccurate. This parameter can be computed in principle as a Lagrange multiplier.

2. Neighborhood filters and NL-means

The principle of most denoising methods is quite simple: they replace the color of a pixel with an average of the nearby pixels color. The variance law in probability theory ensures that if N^2 pixels with the same color plus some decorrelated noise are averaged, then the noise in the average is divided by N. We shall not discuss all denoising methods but concentrate on those which create the least artifacts. Among those, neighborhood filters seem to be the most adequate as pointed out by several perceptual and structure criteria in [2] and [3]. We shall call *neighborhood filters* all image filters which reduce the noise by averaging similar pixels.

2.1. Local neighborhood filters

In order to denoise a pixel, it is better to average the color of this pixel with the nearby pixels with similar colors and only them. This is exactly the technique of the sigma-filter. This famous algorithm is generally attributed to J.S. Lee [9] in 1983 but can be traced back to L. Yaroslavsky and the Soviet Union image processing school [18]. The idea is to average neighboring pixels which also have a similar color value. The filtered value by this strategy can be written as

$$YNF_{h,\rho}\, u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{B_\rho(\mathbf{x})} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} u(\mathbf{y}) d\mathbf{y}, \tag{5}$$

where $u(\mathbf{x})$ is the color at $\mathbf{x}$ and $NF_{h,\rho}\, u(\mathbf{x})$ its denoised version. Only pixels inside $B_\rho(\mathbf{x})$ are averaged, h controls the color similarity and $C(\mathbf{x})$ is the normalization factor. SUSAN [16] and the bilateral filter [17] make this process more symmetric by involving a "bilateral"gaussian depending on both space and grey level. This leads to

$$SNF_{h,\rho}\, u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{\Omega} e^{-\frac{|\mathbf{x}-\mathbf{y}|^2}{\rho^2}} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} u(\mathbf{y}) d\mathbf{y}.$$

2.2. Non local averaging

The most similar pixels to a given pixel have no reason to be close to it. Think of periodic patterns, or of the elongated edges which appear in most images. In 1999 Efros and Leung [4] used non local self-similarities to synthesize textures and to fill in holes in images. Their algorithm scans a vast portion of the image in search of all the pixels that resemble the pixel in restoration. The resemblance is evaluated by comparing a whole window around each pixel, not just the color of the pixel itself. Applying this idea to neighborhood filters leads to a generalized neighborhood filter, the non-local means (or NL-means) filter [2][3]. NL-means has a formula quite similar to the sigma-filter,

$$NLu(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{\Omega} e^{-\frac{(G_\rho * |u(\mathbf{x}+.)-u(\mathbf{y}+.)|^2)(0)}{h^2}}\ u(\mathbf{y})\ d\mathbf{y}, \tag{6}$$

where G_ρ is the Gauss kernel with standard deviation ρ, $C(\mathbf{x})$ is the normalizing factor, h acts as a filtering parameter and

$$(G_\rho * |u(\mathbf{x}+.) - u(\mathbf{y}+.)|^2)(0) = \int_{\mathbb{R}^2} G_\rho(\mathbf{t}) |u(\mathbf{x}+\mathbf{t}) - u(\mathbf{y}+\mathbf{t})|^2 d\mathbf{t}.$$

The formula (6) means that $u(\mathbf{x})$ is replaced by a weighted average of $u(\mathbf{y})$. The weights are significant only if a gaussian window around $\mathbf{y}$ looks like the corresponding gaussian window around $\mathbf{x}$. Thus the non-local means algorithm uses image self-similarity to reduce the noise.

3. Non-local deblurring

3.1. A non-local deblurring energy

Our proposition for deblurring follows from the discussion of the preceding sections. Then, the new hypothesis we wish to introduce to justify a nonlocal deblurring method is the following:

Hypothesis The deblurred image must maintain the same similarities as the blurry image.

It is clear that this hypothesis is a strong limitation since it forbids recreating by deblurring new structures. On the other hand it permits to rebuild all structures provided they left a trace, even tiny, in the blurry image. Thus given a slightly blurry image u_0 we want to define a deblurred version u which has the same self-similarities as u_0. In order to do so, define the new operator associated with u,

$$NL_0u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_\Omega w_0(\mathbf{x},\mathbf{y})\ u(\mathbf{y})\ d\mathbf{y}, \tag{7}$$

with

$$w_0(\mathbf{x},\mathbf{y}) = e^{-\frac{(G_\rho * |u_0(\mathbf{x}+.)-u_0(\mathbf{y}+.)|^2)(0)}{h^2}}.$$

This means that NL_0u is obtained by applying the NL-means algorithm, but instead of computing the weights with u they are computed with u_0.

We are now in a position to define a non-local deblurring functional. Then we propose

$$E_{NL}(u) := \int \left((u(\mathbf{x}) - NL_0u(\mathbf{x}))^2 + \lambda(k * u(\mathbf{x}) - u_0(\mathbf{x}))^2\right) \mathrm{d}\mathbf{x}. \tag{8}$$

The first term can be viewed as a integral of the Laplacian of u with respect to a new metrics defined on the image by the internal similarities of u_0. Thus it forces u to maintain the same coherence as u_0 had. For this reason it can be viewed as a regularization term. Let us see what link we can make for this term with more classical regularity term. If for instance we had $w_0(\mathbf{x},\mathbf{y}) = e^{-|\mathbf{x}-\mathbf{y}|^2/2h^2}$ (up to a normalization constant) and h is large enough, the first term would boil down to the integral $\int |\Delta(\mathbf{x})|^2 d\mathbf{x}$. The derivation of the NL-means from such a variational principle was also commented in [6].

The second term introduces the hypothesized convolution kernel k. In the case where no particular information is available, k is taken to be a gaussian. The parameter λ is the usual Tikhonov weight parameter between the fidelity to the data term and the regularity term.

3.2. The nonlocal reverse heat equation

The evolution equation associated to the deblurring process writes as a non locally stabilized version of the reverse heat equation, namely

$$\frac{\partial u}{\partial t} = -\Delta u + \lambda NL_0 u, \tag{9}$$

where λ plays the same role as before and therefore has the same name. This linear equation can be by Chernoff's principle implemented algorithmically by an *alternate* scheme. One step of reverse heat equation is alternated with one step of nonlocal regularization. Thus the numerical scheme can be described by the discrete iteration

$$u_{n+1} = NL_0(u_n) - c * NL_0(\Delta u_n) = NL_0(u_n - c\Delta u_n) \tag{10}$$

However, the NL-means filter is not fully symmetric and one can have $w(\mathbf{x},\mathbf{y}) \neq w(\mathbf{y},\mathbf{x})$ because of the normalization factor $C(\mathbf{x})$ in the definition of $w(\mathbf{x},\mathbf{y})$. Thus the consistency of the above scheme with (9) is not guaranteed unless one symmetrizes the filter NL. However, the alternate scheme (10) makes sense by itself and this is why it was used in all experiments. The parameter c must be small enough to maintain stability CFL conditions. Notice finally that the reverse heat equation and its implementation are *linear*, in contrast with all above mentioned PDE's.

4. Experiments

Figure 1 applies the alternate scheme to the Lena image. This figure shows how controlling the growth of the Laplacian avoids a blow up of the edges.

Figure 2 compares the non local reverse heat equation with two classical equations, the shock filter and the Perona Malik equation. The shock filter enhances the noise and oscillations of the edges making the result unconvincing. The shock filter can be combined with a mean curvature motion to avoid this effect as proposed in [1]. However, the mean curvature motion filters out details and texture and rounds off corners. The Perona-Malik enhances the main edges but filters the rest of the image. This makes this filter suitable for edge detection but not for enhancement.

5. Conclusion

In this paper we made a quick review of PDE's emulating a reverse heat equation for image enhancement, when the blurring kernel is not known. In fact these PDE's combine a smoothing term with a reverse equation term but the reverse equation is not a real reverse heat. In continuation we have shown that the smoothing term could be the NL-means algorithm which preserves and enhances all image self-similarities. Thus alternating this filter with a real reverse heat equation permits to control the noise and yields a enhancement equation. Further work will concentrate on the mathematical analysis of the consistency of a Chernoff iteration.

Referencias

[1] L. Alvarez and L. Mazorra, *Signal and image restoration using shock filters and anisotropic diffusion*, SIAM J. Numer. Anal. 31 (1994), 590–605.

[2] A. Buades, B. Coll and J.M. Morel, *A review of image denoising methods, with a new one*, Multiscale Modeling and Simulation, vol 4 (2005), 490-530.

[3] A. Buades, B. Coll and J.M. Morel, *A non-local algorithm for image denoising*, IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2005.

[4] A. Efros and T. Leung, *Texture synthesis by non parametric sampling,* Proc. Int. Conf. Computer Vision (ICCV 99), vol. 2 (1999), 1033-1038.

[5] Guichard, F. and Morel, J.M., *A Note on Two Classical Enhancement Filters and Their Associated PDE's*, International Journal of Computer Vision, vol. 52 (2003), 153–160.

[6] Kindermann S, Osher S and Jones PW, *Deblurring and denoising of images by nonlocal functionals*, Multiscale Modeling and simulation, 4 (2005), 1091-1115.

[7] L. S. G. Kovasznay and H. M. Joseph, *Image processing*, Proc. IRE, 43 (1955).

[8] H. P. Kramer and J. B. Bruckner, *Iterations of a non-linear transformation for enhancement of digital images*, Pattern Recognition, 7 (1975).

Figura 1: Experiment comparing the reverse heat equation and the modified version alternating with the NL-means algorithm. Top: Application of the reverse heat equation for time $0{,}4, 0{,}5$. Bottom: Modified NL-means reverse heat equation for the same time and $\lambda = 0{,}25$. The combination with NL-means performs a better enhancement than the reverse heat equation since pixels with a similar structure are uniformly enhanced.

[9] J.S. Lee, *Digital image smoothing and the sigma filter*, Computer Vision, Graphics and Image Processing, vol. 24 (1983), 255-269.

[10] M. Lindenbaum, M. Fischer, and A. M. Bruckstein, *On gabor's contribution to image enhancement*, Pattern Recognition, 27 (1994), 1–8.

[11] P. Perona and J. Malik, *Scale space and edge detection using anisotropic diffusion*, IEEE Trans. Patt. Anal. Mach. Intell., 12 (1990), 629–639.

[12] S. Osher and L. I. Rudin, *Feature-oriented image enhancement using shock filters*, SIAM J. Numer. Anal., 27 (1990), 919–940.

[13] L. I. Rudin, *Images, Numerical Analysis of Singularities and Shock Filters*, PhD thesis, California Institute of Technology, Pasadena, CA, 1987.

[14] L. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D 60 (1992) , 259-268.

[15] Schavemaker, J.G.M., Reinders, M.J.T., Gerbrands, J.J. and Backer, E. *Image sharpening by morphological filtering*, Pattern Recognition, vol. 33, n6 (2000), 997–1012.

[16] S.M. Smith and J.M. Brady, *Susan - a new approach to low level image processing*, International Journal of Computer Vision, Volume 23 (1997), 45-78.

Figura 2: Comparison of different enhancement algorithms. From top to bottom and left to right: reverse heat equation ($t = 0{,}6$), shock filter, the Perona-Malik equation and the non local reverse heat equation ($t = 0{,}6$) with $\lambda = 0{,}25$. The shock filter enhances noise and edge irregularities. The Perona-Malik equation enhances the main edges but blurs the rest of the image, thus loosing many details.

[17] C. Tomasi and R. Manduchi, *Bilateral filtering for gray and color images,* Sixth International Conference on Computer Vision (1998), 839-46.

[18] L.P. Yaroslavsky, *Digital Picture Processing - An Introduction*, Springer Verlag, 1985.