IMAGE DENOISING USING PDE

ZHENGWEI SHEN
DEPARTMENT OF MATHEMATICS OF UNIVERSITY SCIENCE AND TECHONOLGY, BEIJING, CHINA

1. WHAT IS IMAGE(HEAT) DIFFUSION

The concept of diffusion emerged from physical sciences. Diffusion is the movement of particles from an area of high concentration to an area of low concentration. The diffusion result of some kind of particles(e.g. heat, or pixels intensity) is to obtain the mean of the density of particles. Imagine the the evolution of the temperature in a room or the evolution of the concentration of chemical components in a solution. Thus, the diffusion of the image intensity of pixels is equivalent to image smoothing.

In heat transfer, conduction (or heat conduction) is the transfer of thermal energy between neighboring molecules in a substance due to a temperature gradient(or there exists the temperature difference between neighboring molecules). Heat transfer always goes from a region(or a point) of higher temperature to a region(or a point) of lower temperature, and acts to equalize the temperature differences, i.e. achieve the mean of the temperature.

2. Diffusion Equation

Let function $I(x, y, z; t)$ be the temperature at point (x, y, z) and time t. How to describe the asymmetry of temperature distribution? The answer is to use the temperature gradient $\nabla I(x, y, z; t)$. Thus, the reason why the heat will diffusion, just because of the push of the temperature gradient $-\nabla I(x, y, z; t)$, where the minus " $-$ " represents the *push* direct to the temperature decreasing direction at point (x, y, z) and time t.

If the transferring medium is *isotropic*, by the push of $-\nabla I(x, y, z; t)$, then it will result in the flux density, i.e. the fluid passing the unit area that is perpendicular to temperature gradient vector per unit time. That is the flux density can be defined as

$$\overrightarrow{f} = -a \times \nabla I(x, y, z; t)$$

Notice that flux density also is a vector.

Theorem 1. ***Gaussian theorem or Divergence theorem***

Suppose V is a subset of $\mathbb{R}^3$ (i.e. V represents a volume in $3D$ space) which is compact and has a piecewise smooth boundary S (also indicated with $\partial V = S$, i.e. S is the surface

Date: Oct. 8, 2019.

of V). If f is a continuously differentiable vector field defined on a neighborhood of V, then we have

$$\iiint_V (\nabla \cdot \overrightarrow{f})dv = -\oiint_S (\overrightarrow{f} \cdot \overrightarrow{n})dS \tag{1}$$

where the ∂V is a closed manifold quite generally the surface of volume V oriented by outward-pointing normals, and $\overrightarrow{n}$ is the outward pointing unit normal field of the S.

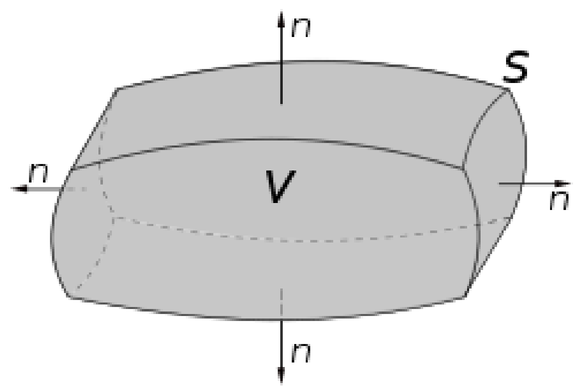

FIGURE 1. A region V bounded by the surface $S = \partial V$ with the surface normal n.

It is well known that the totally volume of the fluid in V is

$$\iiint_V I(x, y, z; t)dv \tag{2}$$

Then then flus rate is defined by the change rate with respect to the time t, that is

$$\frac{\partial \iiint_V I(x, y, z; t)dv}{\partial t} = \iiint_V \frac{\partial I(x, y, z; t)}{\partial t} dv \tag{3}$$

By Eq.(1) and (3), we have

$$\iiint_V \frac{\partial I(x, y, z; t)}{\partial t} dv = \iiint_V (\nabla \cdot \overrightarrow{f})dv \tag{4}$$

That is

$$\frac{\partial I(x, y, z; t)}{\partial t} = -\nabla \cdot \overrightarrow{f} \tag{5}$$

In reminding of that $\overrightarrow{f} = -a \times \nabla I(x, y, z; t)$, we obtain the diffusion equation

$$\frac{\partial I(x,y,z;t)}{\partial t} = a\nabla^2 I(x,y,z;t) \tag{6}$$

where ∇^2 represents the Laplacian operator.

The above process can be seen as the $1D$ diffusion sum of 3 perpendicular directions, with the identical diffusion coefficients. since

$$\frac{\partial I(x,y,z;t)}{\partial t} = a\nabla^2 I(x,y,z;t) = a\frac{\partial^2 I(x,y,z;t)}{\partial x^2} + a\frac{\partial^2 I(x,y,z;t)}{\partial y^2} + a\frac{\partial^2 I(x,y,z;t)}{\partial z^2}$$

So, it is isotropic diffusion.

3. Image denoising using PDE

We consider the degraded image u_0 to be $u_0(x,y) = u(x,y) + \eta$, where η is gaussian noise with zero means and variation σ^2, and $u = u(x,y)$, $(x,y) \in \Omega$, where Ω is the given bounded domain, i.e. the size of image.

The classical denoising procedure is via linear heat equation(i.e. linear diffusion filter)

$$\begin{cases} \dfrac{\partial u(x,y;t)}{\partial t} = a\nabla^2 u(x,y;t) \\ u(x,y;0) = u_0(x,y) \end{cases} \tag{7}$$

That is $u_0(x,y)$ is the initial image at time $t = 0$, and a is the diffusivity coefficient.

Notice that from the PDE theory and the Dirichlet boundary condition $u = u_0 = u(x,y;0)$ on $\partial\Omega$ or the Neumann boundary condition $\frac{\partial u}{\partial \nu} = \nabla u \cdot \nu = 0$ on $\partial\Omega$.

The *fundamental solution* (7), also called a *heat kernel*, is a solution of the heat diffusion equation corresponding to the initial condition of an initial point source of heat at a known position. These can be used to find a general solution of the heat equation over certain domains. In one variables, the Green's function is a solution of the initial value problem, i.e. with initial value $u = \delta(x)$ and $u_t(x;t) = au_{xx}(x;t)$, with $(x,t) \in R \times (0,\infty)$.

$$G(x,t) = \frac{1}{2\sqrt{a\pi t}} e^{-\frac{x^2}{4at}}$$

Then, one can obtain the general solution of the one variable heat equation with initial condition $u_0 = u(x,0)$ by applying a convolution:

$$u(x,y;t) = (G(t) * u_0)(x,y) \tag{8}$$

where $G(x,y;t) = \frac{1}{(2\sqrt{\pi t})^2} e^{(-\frac{x^2+y^2}{4t})}$. From this point view, we can see that the linear diffusion filter is equivalent to the Gaussian filter smoothing with variation $\sigma_G^2 = 2t$.

Next, we will show by using Fourier Transfromation(FT) that the solution of (7) of 1-dimensional on R(infinity domain) is (8). That is, consider the following 1D Heat diffusion equation

$$
\begin{cases} u_t = au_{xx} & (x,t) \in R \times (0,\infty) \\ u(x;0) = u_0(x) & x \in R \\ |u(x;0)|, |u_x| \to 0 \text{ as } x \to 0 \text{ Boundaryverconstions} \end{cases} \tag{9}
$$

has solution given by:

$$
u(x;t) = \frac{1}{2\sqrt{a\pi t}} e^{(-\frac{x^2}{4at})} * u_0(x) = \frac{1}{2\sqrt{a\pi t}} \int_{-\infty}^{\infty} e^{-\frac{(x-y)^2}{4at}} u_0(y)dy, \quad t > 0 \tag{10}
$$

Recall definitions of FT to function $u(x;t)$ with respect to x

- $\hat{u}(\xi;t) = \mathcal{F}(u(x;t)) = \int_{-\infty}^{\infty} u(x;t)e^{-i\xi x}dx$
- $u(x;t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{u}(\xi;t)e^{i\xi x}d\xi$
- $\hat{u}_x(\xi;t) = \mathcal{F}(u_x) = i\xi\mathcal{F}(u) = i\xi\hat{u}$
- $\hat{u}_{xx}(\xi;t) = \mathcal{F}(u_x x) = -\xi^2\mathcal{F}(u) = -\xi^2\hat{u}$
- $\hat{u}_t = \mathcal{F}(\frac{\partial u}{\partial t}) = \frac{\partial}{\partial t}\mathcal{F}(u)$
- $\mathcal{F}((f * g)(x)) = \mathcal{F}(f)\mathcal{F}(g) = \hat{f}\hat{g}$

Now, we perform FT to transform PDEs $u_t = au_{xx}$ to ODEs. By the FT to u_t and u_{xx}, we obtain

$$
\hat{u}_t = \frac{\partial}{\partial t}\hat{u} = -a\xi^2\hat{u}
$$

which is an ODEs and its solution is

$$
\hat{u}(\xi;t) = \hat{u}(\xi;0)e^{-a\xi^2 t}
$$

where $\hat{u}(\xi;0)$ is the FT of initial value $u_0(x)$. Indeed, by

$$
\frac{\partial}{\partial t}(e^{a\xi^2 t}\hat{u}) = 0
$$

we have,

$$
e^{a\xi^2 t}\hat{u} = c_1(\xi)
$$

and

$$
\hat{u}(\xi;t) = c_1(\xi)e^{-a\xi^2 t}
$$

Apply FT to IC $u(x;0) = u_0$ and $t = 0$, we have get $c_1(\xi) = \hat{u}(\xi;0)$

So, obtain the solution to 1D heat equation on R

$$
\hat{u}(\xi;t) = \hat{u}_0(\xi)e^{-a\xi^2 t}
$$

That is, the FT of solution to (9) is the product of the FT to initial value and a Gaussian function $\hat{g}(\xi;t) = e^{-a\xi^2 t}$. That is

$$
\hat{u}(\xi;t) = \hat{u}_0(\xi)\hat{g}(\xi;t)
$$

By the last definitions of FT, we have

$$u(x;t) = u_0(x) * g(x;t) = \int_{-\infty}^{\infty} g(x-y;t)u_0(y)dy \tag{11}$$

So, we just need to obtain inverse FT to the Gaussian function $\hat{g}(\xi;t) = e^{-a\xi^2 t}$

$$\begin{aligned}
g(x;t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-a\xi^2 t}e^{i\xi x}d\xi \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-at[\xi^2-\frac{ix}{at}\xi]}d\xi \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-at[(\xi-\frac{ix}{2at})^2+\frac{x^2}{4a^2t^2}]}d\xi \\
&= \frac{1}{2\pi}e^{-\frac{x^2}{4at}}\int_{-\infty}^{\infty} e^{-at(\xi-\frac{ix}{2at})^2}d\xi \\
&= \frac{1}{2\pi}e^{-\frac{x^2}{4at}}\int_{-\infty}^{\infty} e^{-aty^2}dy \\
&= \frac{1}{2\pi}e^{-\frac{x^2}{4at}}\frac{1}{\sqrt{ct}}\int_{-\infty}^{\infty} e^{-\omega^2}d\omega \\
&= \frac{1}{2\sqrt{a\pi t}}e^{-\frac{x^2}{4at}}
\end{aligned}$$

Then, we obtain the solution to (9).

4. Numerical implementation

In order to numerical solve the (7), we need to discrete the time t, x and y, that is, we need use the discrete difference to replace the one order derivative and the two-order derivative.

As we have known that, the Laplacian operator

$$a\nabla^2 u(x,y;t) = a\frac{\partial^2 u}{\partial x^2} + a\frac{\partial^2 u}{\partial y^2} \tag{12}$$

on the right side in (7) can be discrete, with fixed time t, as

$$\frac{\partial^2 u}{\partial x^2} \approx u(i,j+1) - u(i,j) - (u(i,j) - u(i,j-1)) = u(i,j+1) + u(i,j-1) - 2u(i,j)$$

$$\frac{\partial^2 u}{\partial y^2} \approx u(i+1,j) + u(i-1,j) - 2u(i,j)$$

Thus, $a\nabla^2 u(x,y;t)$ approximate the following difference

$$a\nabla^2 u(x,y;t) \approx a(u(i,j+1) + u(i,j-1) + u(i+1,j) + u(i-1,j) - 4u(i,j)) \tag{13}$$

FIGURE 2. Linear diffusion with different time

Fix the space position (i, j),i.e. for each pixel at position (i, j) the time discrete can be represented as

$$\begin{aligned} \frac{\partial u(x, y; t)}{\partial t} &= \frac{u(i, j; (n+1)\Delta t) - u(i, j; n\Delta t)}{\Delta t} \\ &= \frac{u^{(n+1)}(i, j) - u^n(i, j)}{\Delta t} \end{aligned} \tag{14}$$

Combing with (13) and (14), we have the image *iterative* diffusion equation

$$\frac{u^{(n+1)}(i, j) - u^n(i, j)}{\Delta t} = a(u(i, j+1) + u(i, j-1) + u(i+1, j) + u(i-1, j) - 4u(i, j)) \tag{15}$$

That is

$$u^{(n+1)}(i, j) = u^n(i, j) + a\Delta t \times (u(i, j+1) + u(i, j-1) + u(i+1, j) + u(i-1, j) - 4u(i, j)) \tag{16}$$

Notice the stability of the discrete solution to the original diffusion problem:

- If stepsize Δt are too large, the $u(i, j; n\Delta t)$ is a very poor approximation of the true solution;
- If it is too small, the gain on the approximation is marginal, whereas the computational cost and time increase and becomes much too large.

5. How to implement edge preserving for image deposing using PDE?

5.1. Anisotropic diffusion equation. The intuitive motivation to edge preserving for image deposing using PDE is to *stop* diffusion on the edge, how to implement that idea? The idea of a naturally is the diffusion coefficient depends on the local features of the image. Specifically, in the flat region of the image, where the magnitude of the gradient is small, the DC should be large. On the contrast, the DC should small.

The well known scheme is proposed firstly by Perona and Malik in 1990, named as *nonlinear* diffusion filter or the *anisotropic* diffusion, this is in fact is one kind of additive image denoising method.

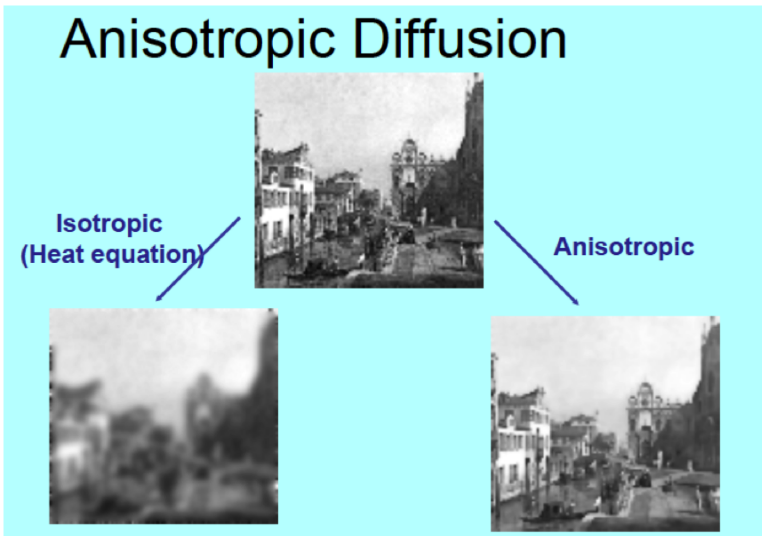

Figure 3. Comparison of isotopic and anisotropic diffusion

$$
\begin{aligned}
&\frac{\partial u}{\partial t} = div(c(x,y)\nabla u) \\
&u(x,y;0) = u_0(x,y)
\end{aligned}
\tag{17}
$$

where $c(x,y)$ is the diffusion coefficient depends on different pixel point (x,y) makes the equation to be nonlinear. Typically, $c(x,y)$ taken as

$$c(x,y) = g(\|\nabla u(x,y)\|)$$

where $\|\nabla u(x,y)\|$ is the gradient magnitude and $g(\|\nabla u\|)$ is an edge-stopping function. This function is chosen to satisfy

$$g(x) \to 0 \quad as \quad x \to \infty$$

For small gradient norm (homogeneous regions) large values of the diffusivity are expected, to perform stronger smoothing. In regions with big gradient norm (inhomogeneity) smaller diffusivity is expected, to slow down the diffusion process and protect delicate image features. So that the diffusion is stopped across edges. Thus, it is anisotropic diffusion or adaptive smoothing.

Two functions to $g(x)$ are suggested by Perona and Malik

$$g(x) = \frac{1}{1 + (x/\lambda)^p}; \quad p = 1 \quad or \quad 2 \tag{18}$$

and

$$g(x) = e^{(-\frac{x^2}{2\lambda^2})} \tag{19}$$

where λ is a constant to be tuned for a particular application.

5.2. **Investigation of anisotropic diffusion equation.** Unfortunately, it has been widely noted that anisotropic diffusions with diffusion coefficients given by (18) and (19) are ill posed in the sense that images close to each other are likely to diverge during the diffusion process. For example, the presence of noise, especially when the gradient generated by noise is comparable to that by image features, can drive the diffusion process to undesirable results. Even without noise, *staircasing* effects can arise around smooth edges.

Several discussions[1] about the ill-posedness of the anisotropic diffusion based on the work of Hollig *etal.*, which states that 1-D anisotropic diffusion is well posed iff

$$\phi'(x) \geq 0$$

where $\phi(x)$ is a flux function(or, influence function) defined as

$$\phi(x) = xg(x)$$

How does the influence function come from? Consider the 1D anisotropic diffusion

$$\begin{aligned} u_t &= \frac{\partial}{\partial x}[g(|u_x|)u_x] \\ &= g'(|u_x|)u_x u_{xx} + g(|u_x|)u_{xx} \\ &= (\underbrace{g'(|u_x|)u_x + g(|u_x|)}_{\phi'(|u_x|)})u_{xx} \end{aligned} \tag{20}$$

By this equation, we find that

$$\phi'(x) = g'(x)x + g(x)$$

For the $g(x)$ defined in (18), for the case $p = 1$, it can be proved that $\phi'(x) \geq 0$ corresponding to the forward(or positive) diffusion; however, for the case $p = 2$, we have

[1] Y.You, W.Xu, A.Tannenbaum and M.Kaveh. Behavioral analysis of anisotropic diffusion in image processing, IEEE Image processing, 5(11): 1539-1553, 1997

$$\begin{cases} \phi'(x) \geq 0, & 0 \leq x \leq k \\ \phi'(x) < 0, & x > k \end{cases} \tag{21}$$

This indicates that , for the case of $x > k$(i.e. the edges in the image), $g(x)$ will lead to backward diffusion. Backward diffusion are well known to be highly unstable and smooth initial data may lead to singularities in arbitrarily short time.

5.3. **Numerical implementation.** Despite of this discouraging analytical facts, Perona and Malik, and many others, found that numerical approximations of (17) do not exhibit significant instabilities. Even better: if computations are carried on for a sufficiently large time interval, the approximations seem to produce piecewise constant solutions giving a simplified image of u_0 preserving sharp boundaries of large brightness variation.

5.3.1. *Semi-explicit discrete scheme.* The explicit discrete scheme proposed by Perona and Malik

$$u_s^{(n+1)} = u_s^{(n)} + \frac{\tau}{|\eta_s|} \sum_{p \in \mathcal{N}_s} g(\|\bar{\nabla} u_{s,p}^{(n)}\|) \bar{\nabla} u_{s,p}^{(n)} \tag{22}$$

where $|\eta_s|$ is the number of neighboring points, here is 4, $\mathcal{N}_s$ is 4-connection neighbor centered at s, and

$$\bar{\nabla} u_{s,p}^{(n)} = u_p^{(n)} - u_s^{(n)}$$

Please notice that here $\bar{\nabla}$ is not the gradient operator! In fact, (22) can be regarded as *semi-explicit discrete* to the $div[g(\|\nabla u\|)\nabla u]$, that is

$$\begin{aligned} div[g\nabla u] &= \frac{\partial}{\partial x}(g u_x) + \frac{\partial}{\partial y}(g u_y) \\ &\approx g_{i,j+1/2}(u_{i,j+1} - u_{i,j}) - g_{i,j-1/2}(u_{i,j} - u_{i,j-1}) \\ &+ g_{i+1/2,j}(u_{i+1,j} - u_{i,j}) - g_{i-1/2,j}(u_{i,j} - u_{i-1,j}) \end{aligned} \tag{23}$$

where $g_{i\pm1/2,j} = g(|u_{i\pm1}, j - u_{i,j}|)$, $g_{i,j\pm1/2} = g(|u_{i,j\pm1} - u_{i,j}|)$.

The *semi-explicit discrete* here indicates that it just simplify the parameter $\|\nabla u\|$ in the function $g(\cdot)$, that is just to take the semi-value on the horizontal and vertical directions , i.e. $|u_x|$ and $|u_y|$. In this case, $\sum_{p\in\mathcal{N}_s} g(\|\bar{\nabla} u_{s,p}^{(n)}\|)\bar{\nabla} u_{s,p}^{(n)}$ can be interpreted as the total heat flux from the neighboring points (p_1, p_2, p_3, p_2) to the centered points s, in a τ time. A large of experiments shows that, if take the stepsize $\tau < 0.25$, even for the case $p = 2$, we will obtain the stable solution.

5.3.2. *Semi-implict discrete scheme.* Semi-implict discrete scheme suggested by Weickert as follows

$$u_s^{(n+1)} = u_s^{(n)} + \tau \sum_{p \in \mathcal{N}_s} \frac{g_p^{(n)} + g_s^{(n)}}{2} (u_p^{(n+1)} - u_s^{(n+1)}) \tag{24}$$

where $g_s^{(n)} := g(|\bar{\nabla} u_{\sigma,s}^{(n)}|)$ with $u_\sigma^{(n)} = u^{(n)} * G_\sigma$ is the regularized image by a Gaussian function.

Since (24) can be expressed linear equations as

$$u^{n+1} = u^n + \tau A^n u^{n+1}$$

That is

$$u^n = [I - \tau A^n] u^{n+1}$$

.

However, in this case for each time step we need to solve a system of equations, and the matrix A, although sparse, would have 256×10^{10} elements for picture of size 400×400 pixels.

So the Jacobi iteration or Gauss-Seidel iteration method can be exploited to solve this system of equations. Furthermore, the multiplicative operator splitting(MOS) or the Peaceman-Rachford splitting method which belongs to alternating direction implicit scheme or the Additional operator splitting(AOS) and the Additional and multiplicative operator splitting(AMOS).

6. Some problems

(1), The isotopic diffusion is equivalent to Gaussian smoothing filter, how about the anisotropic diffusion, is it equivalent to the bilateral smoothing?

(2) How do you regard the adaptive the image smoothing methods?

(3) How to understand the influence function $\phi(x) = xg(x)$?