TV Based Denoising model

Yuyang Zhou 1953793

Date: June, 2020

1 Introduction

TV is the abbreviation of Total variation, a method to deal with functional, instead of function. In image restoration area, the target is not a simply a value, but a function about position(rows and columns). In this case, functional are more widely used.

To set up a TV function, we first need to construct a energy function. After all mathematical methods, our termination is the function that minimizes the energy.

One of the crucial part to figure out the TV model is E-L function, which points out the stationary points of the functional and thereby indicate the extreme point.

2 [Model]restoration model

The core of Total variation regularization is the image restoration model.

Consider to construct a energy functional to evaluate the result. This formula should include two basic function.

2.1 Fidelity term

First, it should evaluate the degree of restoration. The closer to original image, the better the result. Therefore , we can construct a "error term" to evaluate the distance between result and original image and minimize it. Specifically we use the L^2 norm $\iint (u - u_0)^2 dxdy$. We call this term fidelity term.

2.2 Regularization term

Second, we should avoid overfitting, which,in this case, means the result appears abnormally close to the original image. For example, if we only consider and minimize the distance to the original image, ideally, the original image itself should become the optimal solution. However, the image includes some noises, driving it away to reality. That is the so-called overfitting. To avoid overfitting and noise interference, we can turn to the regularization method. A new term is used to evaluate the smoothness of the image.

Specifically in the TV model, we add the L^1 norm to regularize the image.(Typically we use L^1 norm in this model. According to some documents, TV model is the first to consider L^1 norm.) The L^1 norm is decided by both the gradient in x direction and y direction. $|\nabla u| = |\nabla_x u| + |\nabla_y u|$. In discrete cases, or the digital image, gradient has a particular mathematical calculation formula.Take the ∇u_x for example. Derived from the definition $\nabla u_{x_0} = \lim_{x \to x_0} \frac{u(x,y_0)-u(x_0,y_0)}{x-x_0}$, the formula is simply $u(x_0+1, y_0) - u(x_0, y_0)$, since the nearest pixel in x direction is its neighbor and we could approximately consider it infinitely close.

2.3 more explanations

Therefore, the problem ahead is to minimize the functional

$$\min_u |\nabla u|_E + \frac{\mu}{2} \|u - f\|^2$$

Here, $|\nabla u|_E$ means the summation of L^1 norm of all pixels. Attention should be paid to that, μ is a hyperparameter we need to set, which prominently influence the result. It reflects the ratio of fidelity and regularization.

3 [Algorithm]Splitting method E-L function

Then the energy becomes

$$\min_u |\nabla u_x|_E + |\nabla u_y|_E + \frac{\mu}{2} \|u - f\|^2$$

Generally, to find the minimum of the functional, we need to firstly find those points whose first-order derivative equals 0.

3.1 Why choose splitting method

To find those stationary points, we usually use gradient descent method. When it comes to computing the derivative, or gradient, of the functional, the gradient vector appears in the denominator,

leading to instability.

In consideration of that, we turn to a calculation technique called ”the splitting method”.Try to replace those troublesome terms with another variation. In TV model, the splitting version is

$$\min_u |d_x|_E + |d_y|_E + \frac{\mu}{2} \|u - f\|^2$$
$$d_x = \nabla_x u, \quad and d_y = \nabla_y u$$

The new model is equal to the former one. So we can focus on the new one for solution instead.

3.2 Unconstrained version

To make the problem more soluble, we can introduce a penalty term $\frac{\beta}{2} \|d_x - \nabla_x u\| + \frac{\beta}{2} \left\|d_y - \nabla_y u\right\|$. The corresponding unconstrained version is

$$\min_{u,d_x,d_y} |d_x|_E + |d_y|_E + \frac{\mu}{2} \|u - f\|^2 + \frac{\beta}{2} \|d_x - \nabla_x u\| + \frac{\beta}{2} \left\|d_y - \nabla_y u\right\|$$

β is also a crucial hyperparameter. If β is infinitely large, then $\nabla_x u$ is infinitely close to d_x. Theoretically, it has been proved that a not-so-large β will not destroy the astringency of this functional.

As a result, the two formula is absolutely equal.

3.3 Alternating minimization

There are 3 parameter that affect the result. Consider an Alternating method to solve it.

Specifically, for d_x-subproblem, we fix d_y and u, turning it into a uni-variate problem

$$\min_{d_x} |d_x| + \frac{\beta}{2} \|d_x - \nabla_x u\|$$

Solution is the shrinkage operator

$$d_x^* = shrink(\nabla_x u, \frac{1}{\beta})$$

for d_y-subproblem, we fix d_x and u, turning it into a uni-variate problem

$$\min_{d_y} |d_y| + \frac{\beta}{2} \left\|d_y - \nabla_y u\right\|$$

Solution is the shrinkage operator

$$d_x^* = shrink(\nabla_y u, \frac{1}{\beta})$$

for $u-$subproblem, we fix d_x and d_y, turning it into a uni-variate problem

$$\min_u \frac{\mu}{2} \|u - f\|^2 + \frac{\beta}{2} \|d_x - \nabla_x u\| + \frac{\beta}{2} \left\|d_y - \nabla_y u\right\|$$

Solution comes from the E-L equation

$$(\beta \nabla_x^T \nabla_x + \beta \nabla_y^T \nabla_y + \mu I) u = \mu f + \beta \nabla_{xx}^d + \beta \nabla_y^T d_y$$

By split the problem into 3 parts, we can solve each problem by given closed-from solution, making it easier for computers to figure it out.

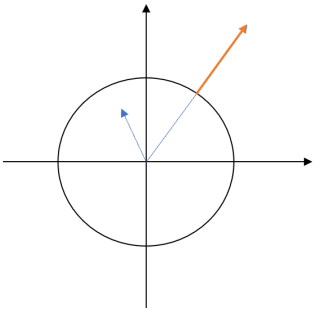

Figure 1: shrinkage sketch map

3.4 shrinkage operator

In fact, shrinkage is a soft thresholding.

$$shrink(x,\gamma) = \frac{x}{|x|} * \max(|x| - \gamma, 0)$$

As shown in **??**, for a vector x, if x is longer than the threshold γ, then we save the direction information of x, and take the bigger part $|x| - \gamma$ as result. Else, the result is 0 instead.

3.5 Eula-Lagrange equation

3.5.1 Mathematical process

Derivate the function

$$\min_u \frac{\mu}{2} \|u - f\|^2 + \frac{\beta}{2} \|d_x - \nabla_x u\| + \frac{\beta}{2} \|d_y - \nabla_y u\|$$

The minimum subject to

$$-div(\frac{\nabla u}{|\nabla u|} + \mu(u - f) = 0$$

It is clear that the gradient emerges on the denominator, that's why we choose the splitting method. For u-subproblem

$$\min_u \frac{\mu}{2} \|u - f\|^2 + \frac{\beta}{2} \|d_x - \nabla_x u\| + \frac{\beta}{2} \|d_y - \nabla_y u\|$$

Attention that u here is processed from a matrix to a vector, simplifying the calculation.$|\nabla_x u|$can be seen as a matrix $|\nabla_x|$ plus the vector u.

Compute the derivation

$$\mu(u - f) + \beta\nabla_x^*(\nabla_x u - d_x) + \beta\nabla_y^*(\nabla_y u - d_y) = 0$$

Arrange the equation and obviously we get a set of liner equations

$$(\mu I + \beta\nabla_x^*\nabla_x + \beta\nabla_y^*\nabla_y)u = \mu f + \beta\nabla_x^* d_x + \beta\nabla_y^* d_y$$

3.5.2 Fourier transform

This is quite a huge liner system. To over relief, the coefficient matrix has some special quality. It is symmetric and thus possibly to be diagnolized by orthogonal matrix. If we take the cycle boundary when calculating the gradient, the coefficient matrix can be diagnolized by Fourier matrix, which means we can solve this problem by Fourier transform.

Name

$$A = \mu I + \beta \nabla_x^* \nabla_x + \beta \nabla_y^* \nabla_y$$

$$b = \mu f + \beta \nabla_x^* d_x + \beta \nabla_y^* d_y$$

The equation comes to a familiar form

$$Au = b$$

And there is a certain Fourier matrix F subject to

$$A = F^T \Lambda F$$

Since F is Fourier matrix, a kind of orthogonal matrix,

$$FF^T = I$$

$$u = F^{-1} \Lambda^{-1} F b$$

Which means only a Fourier transform and a Fourier inverse transform are needed to gain the answer. Specifically,

$$u = F^{-1}\left(\frac{F(h)^* \circ F(f) + F(\partial_1^+)^* \circ F(\omega_1) + F(\partial_2^+)^* \circ F(\omega_2)}{F(h)^* \circ F(h) + F(\partial_1^+)^* \circ F(\partial_1) + F(\partial_2^+)^* \circ F(\partial_2)}\right)$$

4 Experiment

In the unconstrained version of energy function,

$$\min_{u,d_x,d_y} |d_x|_E + |d_y|_E + \frac{\mu}{2} \|u - f\|^2 + \frac{\beta}{2} \|d_x - \nabla_x u\| + \frac{\beta}{2} \left\|d_y - \nabla_y u\right\|$$

there are two hyperparameters need to be set artificially. One of them is μ. Its test results are shown below. The other is β. A value of 0.1 could ensure the convergence.

As we can see, the perfect value for μ is around 0.1. When μ is too small, the fidelity term is nearly neglected. Also, when μ is too big, the regularization term is nearly neglected, and the image appears quite close to the image with noise.

Figure 2: The original image

Figure 3: The image with Gaussian noise

Figure 4: $\mu = 0.001$

Figure 5: $\mu = 0.01$

Figure 6: $\mu = 0.1$

Figure 7: $\mu = 1$

A splitting method for TV

Input matlab source:

```
function R = SplitROF(s,beta,gamma)
[rows,cols,dim] = size(s);
     dx = zeros(rows,cols,dim);
     dy = zeros(rows,cols,dim);
for iter = 1:50
R = FFTsolutionofL2norm(dx,dy,s,beta/gamma);
x = Dx(R); y = Dy(R);
[dx,dy] = shrink2anisotropic(x,y,1/gamma);
end
imshow(R);
end
```

B FFT

Input matlab source:

```
function f = FFTsolutionofL2norm(dx,dy,v,lambda)
%the solution of min{1/2int(gradient u-d)^2+lambda/2int(u-v)^2}
    sizev = size(v);
    otfDx = psf2otf([-1,1],sizev);
    otfDy = psf2otf([-1;1],sizev);
 Nomin = lambda*fft2(v)+conj(otfDx).*fft2(dx)+conj(otfDy).*fft2(dy);
%  size(lambda*fft2(v))
%  size(conj(otfDx))
%  size(dx)
%  size(fft2(dx))
Denom = abs(otfDx).^2 + abs(otfDy ).^2+lambda;
            s = Nomin./Denom;
             f = real(ifft2(s))/255;
end
```

C shrinkage operator

Input matlab source:

```
function [xs,ys] = shrink2anisotropic(x,y,lambda)
xs = max(0, abs(x)-lambda).*sign(x);
ys = max(0, abs(y)-lambda).*sign(y);
```

D gradient operator

Input matlab source:

```
function d = Dy(u)
[rows,cols,dim] = size(u);
d = zeros(rows,cols);
d(2:rows,:,dim) = u(2:rows,:,dim)-u(1:rows-1,:,dim);
d(1,:,dim) = u(1,:,dim)-u(rows,:,dim);
end

function d = Dx(u)
[rows,cols,dim] = size(u);
d = zeros(rows,cols,dim);
d(:,2:cols,dim) = u(:,2:cols,dim)-u(:,1:cols-1,dim);
d(:,1,dim) = u(:,1,dim)-u(:,cols,dim);
end
```