Optimization-based Advanced Image Processing

Master VIBOT, Le Creusot, France

1 Study of a Haar wavelet filter

1.1 Analysis of an unknown signal

Load in Scilab/Octave/Matlab signal s `'greasy.mat'`. This signal corresponds to an audio signal sampling at the frequency of 16 kHz.

Display the signal in the time and frequency domains. Extract information from these two representations.

1.2 Haar filter

We can find in the literature many examples of digital filters associated with wavelet bases with their own properties. They are generally defined as a transfer function. We will study the Haar basis which is basically the easiest to handle. The associated filter is defined by the following transfer function:

$$H_0(z) = \frac{1}{\sqrt{2}}\left(1 + z^{-1}\right) \tag{1}$$

Plot the digital filter frequency response $|H_0(f)|$ according to the normalized frequencies.

What is the type of filter? Measure *graphically* the cutoff frequency of this filter.

This filter will be our reference filter to build a filter bank that will allow to study the different frequency bands of the test signal.

1.3 Complementary Filter

We want to define a filter to retrieve deleted information with the filtering with the low-pass filter of the previous paragraph.

1. What should be the nature of the second filter $G_0(z)$ and its cutoff frequency to be complementary to the first filter? To get it, we apply the transformation

```
    g0 = -( (-1).^(1:length(h0)) ).*h0;
```

   Define the transfer function of this new filter.

   View the frequency response of this filter and check that the gain satisfies the constraint.

2. We want to achieve the decomposition and reconstruction following the principle of the filter bank on one scale. Compute this decomposition. For this you have at your disposal:

   - The instruction `c=pconv(b,s)` for calculating the filtering of signal s by a periodic convolution product following a FIR filter defined by $H(z) = b(z)$
   - The extraction of a part of a vector is computed with `s2=s(1:n2:N)` where `s2` is signal s on which we extracted a point on `n2`.

3. Display the filtered signals c_1 and d_1. Comment the information carried by these two components.

1.4 Reconstruction

We want to define the filters H_1 and G_1 to reconstruct the signal from c_1 and d_1 as indicated in the previous block diagram.

1. What are the natures of filters $G_1(z)$ and $H_1(z)$. To define them, we apply the transformation $H_1(z) = H_0(z^{-1})$ and $G_1(z) = G_0(z^{-1})$.

2. Define the transfer function of these new filters. Plot the frequency response of these filters and check the gains.
   We want to realize the reconstruction for an unique scale following the filter bank principle.

3. Compute this reconstruction. For vector interpolation, we can use the command `s2=zeros(1,n2*N); s2(1:n2:n2*N)=s(1:N)` where `s2` corresponds to signal s interpolated by a factor `n2`.

4. Compare the reconstructed signal with the starting signal.

2 Image denoising

2.1 Image restoration by wavelet

You add white Gaussian noise on loaded images with `randn()`. You can load color images, noise the 3 color components and marginally restore the 3 components. Images can be loaded and rescaled as following:

```
name = 'lena';
n = 256;
f0 = load_image(name);
f0 = rescale(crop(f0,n));
clf;
imageplot(f0);
```

You have a function to decompose an image along a bank of filters:

```
res=fwtor2d(sens,s,L,h0);
```

Parameters

- Variable `s` is a matrix which contains the image.
- Variable `L` is the scale numbers.
- Variable `h0` contains the transfer function coefficients of the low-pass filter.
- Variable `direction` indicates if it is a decomposition or a reconstruction. If `direction =0` it is a decomposition, else it is a reconstruction.

Results

- Variable `res` is a matrix of same size as `s` that contains either the result of the decomposition (`direction=0`) or the one of the reconstructed image.

We use the Daubechies'D4 low pass filter. This filter is defined by the following transfer function:

$$H_0(z) = \frac{326}{675} + \frac{1095}{1309}z^{-1} + \frac{648}{2891}z^{-2} - \frac{675}{5216}z^{-3} \tag{2}$$

You will test the denoising with a 2D wavelet transform. The hard thresholding is defined as $h_T(x) = 0$ if $-T < x < T$ and $h_T(x) = x$ otherwise. It thus defines a thresholding operator of wavelet coefficients as $H_T(a)_m = h_T(a_m)$.

```
HardThresh = @(x,t)x.*(abs(x)>t);
```

We see that we threshold the decomposition. In general the threshold is defined by t=3σ where σ is the noise standard deviation. We then reconstruct.

Apply this denoising processing on color noisy images using wavelet decomposition. The appendix explains the threshold computation. Display the reconstructed image.

Change the noise magnitude and evaluate the SNR of the resulting images.

3 Inpainting using Sparse Regularization Algorithm

3.1 Sparse Regularization Principle

Let measurements be $y = \Phi f_0$ where Φ is a masking operator.

Image is recovered using sparsity from the measurements y. It considers a synthesis-based regularization, that compute a sparse set of coefficients $(a_m^\star)_m$ in a frame $\Psi = (\psi_m)_m$ that solves

$$a^\star \in \mathrm{argmin}_a \frac{1}{2}\|y - \Phi\Psi a\|^2 + \lambda J(a),$$

where λ should be adapted to the noise level $\|w\|$.

Ψa indicates the synthesis operator, and $J(a)$ is the ℓ_1 sparsity prior:

$$J(a) = \sum_m \|a_m\|.$$

3.2 Work: Inpainting using Sparse Regularization

3.2.1 Damaged Image

In the framework of an inpainting problem, we consider a linear imaging operator $\Phi : f \mapsto \Phi(f)$ that maps high resolution images to low dimensional observations. A masking operator Φ has to be computed from a random mask Λ:

```
rho = .8;
Lambda = rand(n,n)>rho;
Phi = @(f)f.*Lambda;
```

Parameter ρ ajusts the damage. Observations $y = \Phi f_0$ have computed as following:

```
y = Phi(f0);
imageplot(y);
```

3.2.2 Work: Inpainting process

The soft thresholding operator is the simplest ℓ_1 minimization schemes. It can be applied to coefficients a, or to an image f in an ortho-basis. The soft thresholding is a 1-D functional that shrinks the value of coefficients.

$$s_T(u) = \max\left(0, 1 - \frac{T}{|u|}\right) u.$$

```
SoftThresh = @(x,T)x.*max(0, 1-T./max(abs(x),1e-10));
```

We use an orthogonal wavelet basis Ψ and we set the following parameters:

```
Jmax = log2(n)-1;
Jmin = Jmax-3;
options.ti = 0; % use orthogonality.
Psi = @(a)perform_wavelet_transf(a, Jmin, -1,options);
PsiS = @(f)perform_wavelet_transf(f, Jmin, +1,options);
```

The soft thresholding operator in the basis Ψ corresponds to applying the transform Ψ^*, thresholding the coefficients using S_T and then undoing the transform using Ψ:

$$S_T^{\Psi}(f) = \Psi \circ S_T \circ \Psi^*$$

This process on $f0$ can be computed and displayed with:

```
SoftThreshPsi = @(f,T)Psi(SoftThresh(PsiS(f),T));
imageplot(clamp(SoftThreshPsi(f0,.1)) );
```

Implement the algorithm with a static threshold and analyse the evolution of the restored images according to the iterations.

We can improve the reconstruction result by using a soft thresholding 1-D functional.

```
T = linspace(Beg,End,Nb);
plot(T, SoftThresh(T,.5));
```

Implement the algorithm and analyse the evolution of the restored images according to the iterations. Use different images and different values of the damage parameter ρ.

4 Inpainting using Primal-Dual Total Variation Algorithm

4.1 Douglas-Rachford Algorithm

The Douglas-Rachford algorithm is an iterative scheme to minimize functionals of the form

$$\min_x F(x) + G(x),$$

where F and G are convex functions, of which one is able to compute the proximity operators. This algorithm was is a generalization of an algorithm introduced by Douglas and Rachford in the case of quadratic minimization.

The Douglas-Rachford algorithm takes an arbitrary element $s^{(0)}$, a parameter $\gamma > 0$, a relaxation parameter $0 < \rho < 2$ and iterates, for $k = 1, 2, \ldots$

$$\left| \begin{array}{l} x^{(k)} = \mathrm{prox}_{\gamma F}(s^{(k-1)}), \\ s^{(k)} = s^{(k-1)} + \rho\big(\mathrm{prox}_{\gamma g}(2x^{(k)} - s^{(k-1)}) - x^{(k)}\big). \end{array} \right.$$

It is of course possible to inter-change the roles of F and G, which defines a different algorithm. The iterates $x^{(k)}$ converge to a solution $x^\star$ of the problem, i.e. a minimizer of $f = F + G$.

4.2 Convex Optimization with a Primal-Dual Proximal Splitting

We consider general optimization problems of the form

$$\min_f F(K(f)) + G(f),$$

where F and G are convex functions and $K : f \mapsto K(f)$ is a linear operator.

To apply the primal-dual algorithm, the proximal mapping of F and G can be computed from:

$$\mathrm{Prox}_{\gamma F}(x) = \operatorname*{argmin}_y \frac{1}{2}\|x - y\|^2 + \gamma F(y),$$

The numerical processes are:

$$g_{k+1} = \mathrm{Prox}_{\gamma F^*}(g_k + \gamma K(\tilde{f}_k)),$$

$$f_{k+1} = \mathrm{Prox}_{\tau G}(f_k - \tau K^*(g_k)),$$

$$\tilde{f}_{k+1} = f_{k+1} + \theta(f_{k+1} - f_k).$$

And the dual functional is defined as following:

$$F^*(y) = \max_x \langle x, y\rangle - F(x).$$

To compute the proximal mapping of F is equivalent to compute the proximal mapping of F^* thanks to Moreau's identity:

$$x = \mathrm{Prox}_{\tau F^*}(x) + \tau \mathrm{Prox}_{F/\tau}(x/\tau).$$

4.3 Work: Inpainting using Primal-dual Total Variation Regularization scheme

The aim is to implement the primal-dual total variation algorithm and inpaint an image from an inverse problem using a total variation regularization:

$$\min_{y=\Phi f} \|\nabla f\|_1$$

For the application, the minimization of $F(K(f)) + G(f)$ can be rewritten as:

$$G(f) = i_H(f), \quad F(u) = \|u\|_1 \quad \text{and} \quad K = \nabla,$$

with i_H the indicator function

These operators can be computed as following:

```
K  = @(f)grad(f);
KS = @(u)-div(u);
Amplitude = @(u)sqrt(sum(u.^2,3));
F = @(u)sum(sum(Amplitude(u)));
```

The proximal operator of the vectorial ℓ_1 norm reads

$$\text{Prox}_{\lambda F}(u) = \max\left(0, 1 - \frac{\lambda}{\|u_k\|}\right) u_k$$

```
ProxF = @(u,lambda)max(0,1-lambda./repmat(Amplitude(u), [1 1 2])).*u;
```

A simple way to compute the proximal operator of the dual function F^*, we make use of Moreau's identity:

$$x = \text{Prox}_{\tau F^*}(x) + \tau \text{Prox}_{F/\tau}(x/\tau)$$

```
ProxFS = @(y,sigma)y-sigma*ProxF(y/sigma,1/sigma);
```

The proximal operator of $G = i_H$ is the projector on H. In our case, since Φ is a diagonal so that the projection is simple to compute

$$\text{Prox}_{\tau G}(f) = \text{Proj}_H(f) = f + \Phi(y - \Phi(f))$$

```
ProxG = @(f,tau)f + Phi(y - Phi(f));
```

We set parameters for the algorithm. In our case, $L = \|K\|^2 = 8$ and we need to have $L\sigma\tau < 1$.

```
L = 8;
sigma = 10;
tau = .9/(L*sigma);
theta = 1;
f = y;
g = K(y)*0;
f1 = f;
```

Example of one iterations.

```
fold = f;
g = ProxFS( g+sigma*K(f1), sigma);
f = ProxG(f-tau*KS(g), tau);
f1 = f + theta * (f-fold);
```

Implement the algorithm for many iterations and analyse the evolution of the restored images according to the iterations. Plot the evolution of the TV energy $F(K(fk))$. Use different images and different values of the damage parameter ρ.

Appendix

In the case of Gaussian noise, the problem is to determine a real function denoted by f from known measurements x_i such as:

$$x_i = f(t_i) + b_i, \tag{3}$$

with $i = 0...N$ and $b_i \sim \mathcal{N}(0, 1)$ (i.i.d) and > 0.

With a orthogonal discrete wavelet transform, the white noise is composed of a series of random normal centered and decorrelated coefficients. Accordingly the discrete wavelet transform of measures can be written :

$$Wx_i = Wf(t_i) + d_i^b, \tag{4}$$

with d_i^b is a $\mathcal{N}(0, 1)$ (i..i.d). W symbolizes the decimated discrete wavelet transform and d_i^b the noise transform.

To reconstruct the function $f(t)$ from measurements xi, all the coefficients of decomposition Wx_i are shrinked by a thresholding depending on the noise contribution. Donoho and Johnstone have proposed two types of threshold functions (denoted T with a threshold):

- The *Soft thresholding* is defined by:

$$T^S(dl, k) = \mathrm{sgn}(d_{l,k})\left(|d_{l,k}| - \lambda\right)_+ \text{ with } \alpha_+ = \sup(\alpha, 0) \tag{5}$$

- The *Hard thresholding* is defined by:

$$T^H(d_{l,k}) = d_{l,k}.1_{\left[|d_{l,k}| \geq \lambda\right]} \text{ with } 1_A \text{ indicator function of } A \tag{6}$$

It exists a number of methods for estimating the threshold. The first of them called *VisuShrink*, or wavelet shrinkage Universal threshold, has been proposed by Donoho and Johnsone with λ defined by:

$$\lambda = \sigma\sqrt{2\log(n)} \;\; \text{with} \;\; \sigma = \frac{\text{MAD}}{0,6745}. \tag{7}$$

where n is the number of measurements and MAD is the absolute median value of coefficients at the finest scale.

Factor 0.6745 is chosen after a calibration with a Gaussian distribution. Noise dispersion is measured on the first level because it consists essentially of coefficients due to noise.

In summary, from measurements x_i, we use the Mallat algorithm to obtain the detail coefficients, we shrink coefficients by thresholding and we reconstruct.

References

[1] Thomas Blumensath and Mike E. Davies, *Iterative thresholding for sparse approximations*, Journal of Fourier Analysis and Applications, 2008.

[2] Antonin Chambolle and Thomas Pock, *A First-order primal-dual algorithm for convex problems with application to imaging*, Journal of Mathematical Imaging and Vision, 2011

[3] Patrick L. Combettes and Jean-Christophe Pesquet, *Proximal Splitting Methods in Signal Processing*. In: Fixed-point algorithms for inverse problems in science and engineering. Springer New York, 2011.

[4] Laurent Condat, *A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms*, J. Optimization Theory and Applications, 2013

[5] M. Elad and M. Aharon, Image Denoising Via Sparse and Redundant representations over Learned Dictionaries, IEEE Trans. on Image Processing, December 2006.

[6] P. L. Lions and B. Mercier, *Splitting Algorithms for the Sum of Two Nonlinear Operators*, SIAM Journal on Numerical Analysis, 1979

[7] Stéphane Mallat, *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation*, IEEE Transactions on PAMI, 11(7), 1989

[8] Stéphane Mallat, *A wavelet tour of signal processing: the sparse way*, Academic Press, 2009

[9] BA Olshausen, and DJ Field, *Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images*, Nature, 1996.

[10] Gabriel Peyré, *Numerical Tours of Data Sciences*, `http://www.numerical-tours.com/`

[11] P. Tseng, *Convergence of Block Coordinate Descent Method for Nondifferentiable Minimization*, J. Optim. Theory Appl., 109, 2001