Problem Session 3

Jan. 22th 2016

Orly Liba
A few announcements

• Poster presentation on Wednesday, March 9, from 4-6 pm in the Packard atrium. Poster easels will be provided. More details to come.

• “Buddy system” for using Packard 001: Always bring someone with you that could call for help if something goes wrong (flood, earthquake etc...). We don’t want you to be alone in Packard basement. Can be anyone:

  a classmate,
  a friend,
  Mom or Dad,
  a CA,
  a date,
  etc.
  but they need to be able to use a telephone and summon help using the English language.

  For example, “rescue” dogs would NOT be suitable ....

  Thank you for helping keep folks safe!    (Steven Clark)

• We’re also working on getting you remote access.
Topics

• Image filtering:
  • Spatial domain vs. Fourier domain
  • Low pass and high pass

• Image restoration: sharpen a noisy image
  • Straight forward
  • With damping (Wiener)
  • ADMM
Task 1: Image filtering

Primal domain vs. Fourier domain

- Primal: $I(x, y) \rightarrow I(x, y) \ast PSF(x, y)$

  Point spread function

- Fourier domain: $\tilde{I}(\omega_x, \omega_y) \rightarrow \tilde{I}(\omega_x, \omega_y) \times OTF(\omega_x, \omega_y)$

  Optical transfer function
Task 1: Image filtering

Primal domain vs. Fourier domain

• Helpful functions: fft2, ifft2, pst2otf, fspecial, conv2
• Notice that the time of the convolution increases as the PSF becomes larger. The time of the Fourier domain computation remains similar and independent of the kernel size.
• Don’t forget to normalize the filter so the sum over all elements will be 1.
• You can implement high pass as $\tilde{I} \times (1 - OTFLP)$ in the Fourier domain or $(I - I \ast PSF_{LP})$ in the primal domain.
Task 1: Image filtering

Example of results

• Primal and dual results look similar
Task 1: Image filtering

Example of results

Why are we seeing this behavior?
Task 2: Image restoration

What is the best $h'$ (or $H'$)?

Simply using $H' = 1/H$ will (usually) amplify noise and destroy the image. Why?
Task 2: Image restoration

For HW:

• First, blur the image with a Gaussian kernel (primal or Fourier domain)
• Add random noise: \( I = I + \text{sigma} \cdot \text{randn(size(I))} \)
• Reconstruct the image by
  1. Dividing by the blur kernel (OTF) in Fourier domain (inverse filtering)
  2. Wiener deconvolution, which is almost the same as inverse filtering, but uses a damping factor in the Fourier domain that depends on the noise

\[
H' = \frac{1}{H} \cdot \frac{|H|^2}{(|H|^2 + 1/\text{SNR})} \quad \text{SNR} = \frac{\bar{I}}{\sigma_{\text{noise}}}
\]
Task 2: Image restoration

Calculate the mean squared error (MSE) and the peak signal-to-noise ratio (PSNR):

\[
MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [I_{\text{original}}(i, j) - I_{\text{restored}}(i, j)]^2
\]

\[
PSNR = 10 \log_{10} \left( \frac{\text{max}(I_{\text{original}})^2}{MSE} \right)
\]
Task 2: Image restoration

Example of results

Blurred image with noise, $\sigma = 0.001$

Inverse filtering
Image after inverse filtering, PSNR = -158.5462 dB

Wiener deconvolution
Image after Wiener deconvolution, PSNR = 26.4985 dB
Task 3: Deconvolution using ADMM

- We want to use an assumption (prior) to reconstruct the image.
- Our assumption is that the image (without noise and blurring) has **sparse gradients**. Minimizing the L1 norm of the gradients would result in that.
- This prior is inserted as a **regularization** term in an optimization problem.
From EE364a (lecture 6), histogram of residuals for penalties (regularization)

\[
\phi_L(u) = |u|, \quad \phi_L(u) = u^2, \quad \phi_L(u) = \max\{0, |u| - a\}, \quad \phi_L(u) = -\log(1 - u^2)
\]

Sparse gradients:
Many are zero
The rest are spread out

shape of penalty function has large effect on distribution of residuals
Task 3: Deconvolution using ADMM

The reconstruction problem becomes:

Find image $x$ that minimizes

$$
\frac{1}{2} \|Cx - b\|^2_2 + \lambda \|Dx\|_1
$$

- $C$ is the blur operator
- $b$ is the noisy and blurred image
- $X$ is the image we want to find
- $D$ is the gradient operator
- Regularization term

Not differentiable...
Therefore, solution is not analytical
Task 3: Deconvolution using ADMM

How can we find $x$?

By using alternating direction method of multipliers (ADMM)!

ADMM splits the objective into a weighted sum of two independent functions $f(x)$ and $g(z)$ that are only linked through the constraints.

$$\text{minimize}_{\{x\}} \frac{1}{2} \|Cx - b\|^2_2 + \lambda \|z\|_1$$

subject to $Dx - z = 0$

(Equivalent to previous slide)
Task 3: Deconvolution using ADMM

Following the general ADMM strategy, the Augmented Lagrangian of Equation 9 is formed as

$$L_\rho (x, z, y) = f(x) + g(z) + y^T (Dx - z) + \frac{\rho}{2} \|Dx - z\|_2^2$$

(10)

As discussed in more detail in Chapter 3.1 of [Boyd et al. 2001], using the scaled form of the Augmented Lagrangian, the following iterative updates rules can be derived:

$$x \leftarrow \text{prox}_{f,\rho} (v) = \arg \min_{\{x\}} L_\rho (x, z, y) = \arg \min_{\{x\}} f(x) + \frac{\rho}{2} \|Dx - v\|_2^2, \quad v = z - u$$

$$z \leftarrow \text{prox}_{g,\rho} (v) = \arg \min_{\{z\}} L_\rho (x, z, y) = \arg \min_{\{z\}} g(z) + \frac{\rho}{2} \|v - z\|_2^2, \quad v = Dx + u$$

(11)

$$u \leftarrow u + Dx - z$$

where $u = (1/\rho)y$. The $x$ and $z$-updates are performed with what is known as proximal operators $\text{prox}_{\cdot,\rho} : \mathbb{R}^N \rightarrow \mathbb{R}^N$. The interested reader is referred to [Boyd et al. 2001] for more details.
Task 3: Deconvolution using ADMM

Pseudo code (what you need to implement):

```
Algorithm 1 ADMM for TV-regularized deconvolution
1: initialize $\rho$ and $\lambda$
2: $x = \text{zeros}(W, H)$
3: $z = \text{zeros}(W, H, 2)$
4: $u = \text{zeros}(W, H, 2)$
5: for $k = 1$ to maxIters do
6: \hspace{1cm} $x = \text{prox}_{f,\rho} (z - u) = \text{arg min}_{\{x\}} \frac{1}{2} \|Cx - b\|_2^2 + \frac{\rho}{2} \|Dx + u\|_2^2$
7: \hspace{1cm} $z = \text{prox}_{g,\rho} (Dx + u) = S_{\lambda/\rho} (Dx + u)$ (19)
8: \hspace{1cm} $u = u + Dx - z$
9: \hspace{1cm} end for
```

We provided a template.
Run on 2 images, Lenna and art.
Present the resulting images and the convergence of the algorithm (the residual for each iteration). Also report the PSNR (compared to image without noise and blurring).
Task 3: Deconvolution using ADMM

Hints:

• Read the lecture notes. Use them as your guide for this implementation.
• All operations are in the Fourier domain.
• Calculate the gradient operators in Fourier domain as:
  \[ \text{dx} = [-1, 1]; \quad \text{dxFT} = \text{psf2otf(dx)}; \quad (\text{similarly on y}) \]
  And then \( D_x^T x \) can be calculated as \( \text{real}(\text{ifft2}(\text{fft2}(x) .* \text{dxFT})) \)
• Convolution with the transpose kernel is implemented as a Fourier multiplication with the conjugate kernel: \( \text{conj} (\text{psf2otf(psf)}) \).
• To update \( x \) use equation 17 in the notes. Use division in Fourier domain.
• The residual is the function we want to minimize (eq. 8 in the notes)
• Norm of vector ≠ Norm of matrix \( \rightarrow \) vectorize before norm
Task 3: Deconvolution using ADMM

- The residual should always decrease.
- The PSNR should always increase.
- Works better when gradients are very sparse.
- On more natural images, the result is patchy
- Try different $\lambda$. As $\lambda$ increases, more noise is reduced, but images become patchier too.
Task 3: Deconvolution using ADMM

PSNR = 25.51dB $\sigma_n=0.001 \lambda=0.005 \rho=10$
Have a nice weekend!
And good luck with the homework!