Investigating Image Inpainting via the Alternating Direction Method of Multipliers (ADMM)

Jonathan Tuck
Stanford University

Motivation
- In many imaging applications, there exists potential for corruption of the images by sources of noise that completely lose original pixel information, such as degradation over time [1]:
  - Image inpainting allows for the estimation and restoration of missing pixels
  - The inpainting process leaves some artifacts behind, and is usually not perfect
  - Accuracy of the inpainting is related to the prior used

ADMM & Image Inpainting
- The Alternating Direction Method of Multipliers (ADMM) [2] solves the following optimization problem:
  \[
  \begin{align*}
  \text{minimize} & \quad f(x) + g(z) \\
  \text{subject to} & \quad Ax + Bz = c
  \end{align*}
  \]
- ADMM is general, simple, and parallelizable [2]
- Solved using the following update procedure [2] [3]:
  \[
  \begin{align*}
  x^{k+1} &= \text{argmin}_x L_\rho(x, z^k) = \text{prox}_{\frac{1}{\rho} f}(x - \frac{1}{\rho} z^k) \\
  z^{k+1} &= \text{argmin}_z \frac{1}{2\rho} L_\rho(x^k, z) = \text{prox}_{\frac{1}{\rho} g}(z - \frac{1}{\rho} L_\rho(x^k, z))
  \end{align*}
  \]
- Image inpainting can be formulated in the ADMM framework:
  \[
  \begin{align*}
  \text{minimize} & \quad \|Kx - b\|_2^2 + \lambda \Gamma(z) \\
  \text{subject to} & \quad x = z = 0
  \end{align*}
  \]
  - b is observed, K is the diagonal inpainting mask, and \( \Gamma(z) \) is the prior information on the image

Prior Selection
- Total Variation [4]
  - Used when image has sparse gradients
  - Makes use of finite difference matrix D
- Non-Local Means [5]
  - Used when image has self-similar structure
  - Weighted sum of differences in neighborhoods
- Recursive Filter [6]
  - Takes an image, transforms it into a new domain (right), filters the transformed signal with a Gaussian filter, and inverts the transformation.
- BM3D [7]
  - Used when image has a locally sparse transform
  - Performs block matching, denoising in 3D transform domain

Methodology and Results
- Formulate image inpainting as an ADMM problem and solve using ADMM
- Use "Plug-and-Play" ADMM [8] as an open source ADMM solver
- Measure PSNR and SSIM of inpainted images, using different corruption masks

Example results:
- Original Image
- Corrupted Image
- Inpainting
- Reconstructed Image

Results / Discussion
- In most cases, corruption mask determined which prior gave highest quality output
- Some images will perform better with a particular prior regardless of the mask type

Future Work
- Quantify relationship between mask type (corruption model) and priors, if any
- Machine learning applications:
  - An algorithm might be able to use other test images to determine best prior, \( \lambda \) to use for a given image.

Conclusion
- ADMM is an efficient and effective way of implementing image inpainting
- Not only does the image itself affect prior selection, but the corruption model affects which prior to use

References