Computational Methods for Inverse Problems and Applications in Image Processing

Lecture 7
Separable Nonlinear Problems

James Nagy

Emory University
Atlanta, GA USA

Introduction

Here we consider the *separable* nonlinear problem

$$\mathbf{b} = \mathbf{A}(\mathbf{y}_{\rm true})\mathbf{x}_{\rm true} + \boldsymbol{\eta}$$

where

- $\mathbf{A}(\mathbf{y})$ is a matrix, defined by (unknown) parameters:

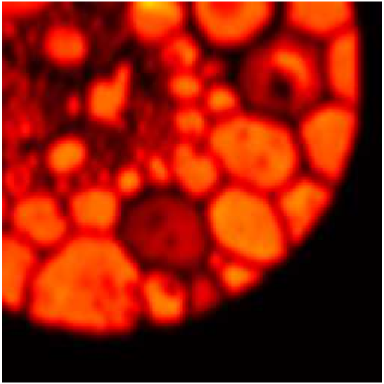
$$\mathbf{y} = \left[\begin{array}{cccc} y_1 & y_2 & \cdots & y_p \end{array} \right]^T$$

- The goal is to find approximations of $\mathbf{x}_{\rm true}$ and $\mathbf{y}_{\rm true}$.
- In imaging applications, $\mathbf{x}_{\rm true}$ is typically an (unknown) image of an object.
- Note that the forward problem, $\mathbf{b} = F(\mathbf{x}, \mathbf{y}) + \boldsymbol{\eta}$, depends
  - linearly on $\mathbf{x}$
  - nonlinearly on $\mathbf{y}$

Application: Blind Deconvolution

- $\mathbf{b} = \mathbf{A}(\mathbf{y})\,\mathbf{x} + \boldsymbol{\eta}$ = observed image where **y** describes blurring function
- Given: **b** and an estimate of **y**
- Standard Image Deblurring: Compute approximation of **x**
- Better approach: Jointly improve estimate of **y** and compute approximation of **x**.
- See project 4.

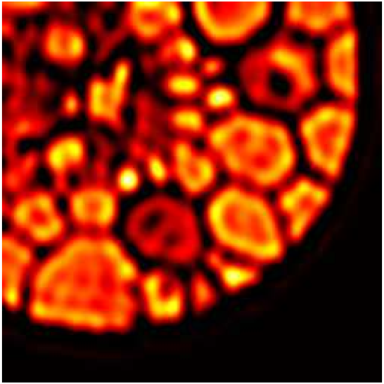
Observed Image



Application: Blind Deconvolution

- $\mathbf{b} = \mathbf{A}(\mathbf{y})\,\mathbf{x} + \boldsymbol{\eta}$ = observed image
  where $\mathbf{y}$ describes blurring function
- Given: $\mathbf{b}$ and an estimate of $\mathbf{y}$
- Standard Image Deblurring:
  Compute approximation of $\mathbf{x}$
- Better approach:
  Jointly improve estimate of $\mathbf{y}$
  and compute approximation of $\mathbf{x}$.
- See project 4.

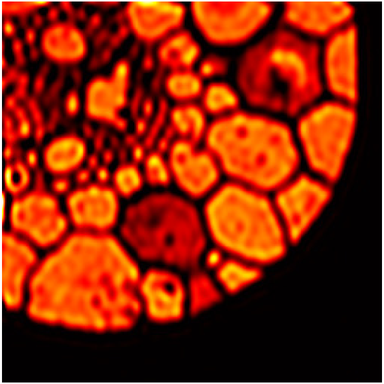
Reconstruction using initial PSF



Application: Blind Deconvolution

- $\mathbf{b} = \mathbf{A}(\mathbf{y})\,\mathbf{x} + \boldsymbol{\eta}$ = observed image where $\mathbf{y}$ describes blurring function
- Given: $\mathbf{b}$ and an estimate of $\mathbf{y}$
- Standard Image Deblurring: Compute approximation of $\mathbf{x}$
- Better approach: Jointly improve estimate of $\mathbf{y}$ and compute approximation of $\mathbf{x}$.
- See project 4.

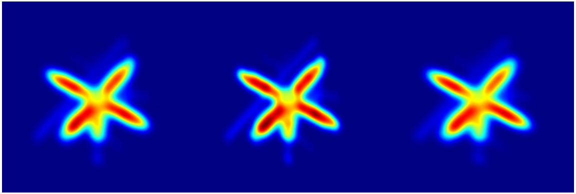
Reconstruction after BD method



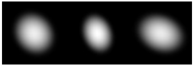
Multi-Frame Blind Deconvolution (MFBD)

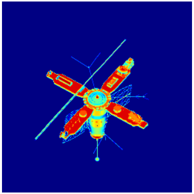
Multi-Frame Blind Deconvolution:

- Given blurred images:



- Compute estimates of true PSFs and image





Multi-Frame Blind Deconvolution (MFBD)

$$\underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} \mathbf{A}(\mathbf{y}_1) \\ \vdots \\ \mathbf{A}(\mathbf{y}_m) \end{bmatrix}}_{\mathbf{A}(\mathbf{y})} \mathbf{x} + \underbrace{\begin{bmatrix} \eta_1 \\ \vdots \\ \eta_m \end{bmatrix}}_{\eta}$$

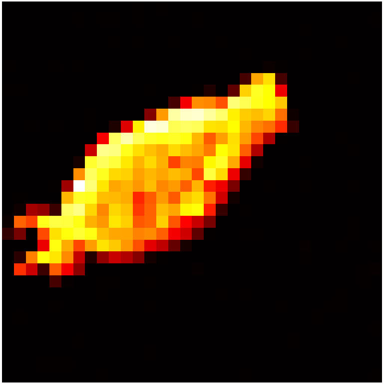
- Known: $\mathbf{b}_j$ = observed image, frame j
- Unknown: $\mathbf{y}_j$ = parameters defining PSF for frame j
- Unknown: $\mathbf{x}$ = image to reconstruct
- Note: If we assume $\mathbf{x}, \mathbf{b}_j \in \mathcal{R}^n$, and $\mathbf{y}_j \in \mathcal{R}^p$, then
  - Number of measured data: mn
  - Number of unknowns: $mp + n$
  - If $p = \frac{n}{2}$ and $m = 1$ (single frame), then $mn = n < \frac{3}{2}n = mp + n$
  - If $p = \frac{n}{2}$ and $m = 4$ (four frames), then $mn = 4n > 3n = mp + n$

Super-resolution/Data Fusion

- $\mathbf{b}_j = \mathbf{A}(\mathbf{y}_j)\,\mathbf{x} + \eta_j$
  (collected low resolution images)

$$\underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} \mathbf{A}(\mathbf{y}_1) \\ \vdots \\ \mathbf{A}(\mathbf{y}_m) \end{bmatrix}}_{\mathbf{A}(\mathbf{y})} \mathbf{x} + \underbrace{\begin{bmatrix} \eta_1 \\ \vdots \\ \eta_m \end{bmatrix}}_{\eta}$$

1−th low resolution image



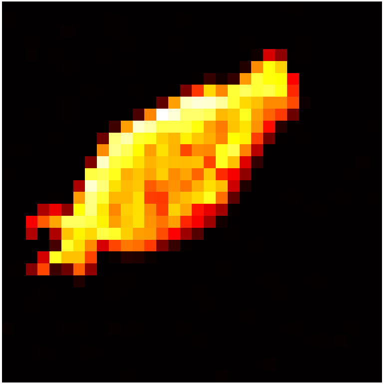
- $\mathbf{y}$ = registration, blurring, etc., parameters
- Goal: Improve parameters $\mathbf{y}$ and compute $\mathbf{x}$

Super-resolution/Data Fusion

- $\mathbf{b}_j = \mathbf{A}(\mathbf{y}_j)\,\mathbf{x} + \boldsymbol{\eta}_j$
  (collected low resolution images)

$$\underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} \mathbf{A}(\mathbf{y}_1) \\ \vdots \\ \mathbf{A}(\mathbf{y}_m) \end{bmatrix}}_{\mathbf{A}(\mathbf{y})} \mathbf{x} + \underbrace{\begin{bmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_m \end{bmatrix}}_{\boldsymbol{\eta}}$$

8−th low resolution image



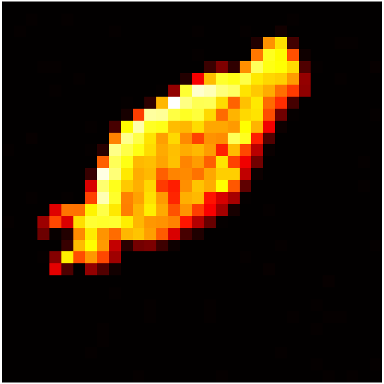
- $\mathbf{y}$ = registration, blurring, etc., parameters
- Goal: Improve parameters $\mathbf{y}$ and compute $\mathbf{x}$

Super-resolution/Data Fusion

- $\mathbf{b}_j = \mathbf{A}(\mathbf{y}_j)\,\mathbf{x} + \eta_j$
  (collected low resolution images)

$$\underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} \mathbf{A}(\mathbf{y}_1) \\ \vdots \\ \mathbf{A}(\mathbf{y}_m) \end{bmatrix}}_{\mathbf{A}(\mathbf{y})} \mathbf{x} + \underbrace{\begin{bmatrix} \eta_1 \\ \vdots \\ \eta_m \end{bmatrix}}_{\eta}$$

15−th low resolution image

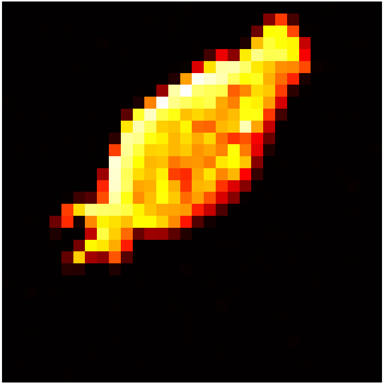


- $\mathbf{y}$ = registration, blurring, etc., parameters
- Goal: Improve parameters $\mathbf{y}$ and compute $\mathbf{x}$

Super-resolution/Data Fusion

- $\mathbf{b}_j = \mathbf{A}(\mathbf{y}_j)\,\mathbf{x} + \eta_j$
  (collected low resolution images)

$$\underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} \mathbf{A}(\mathbf{y}_1) \\ \vdots \\ \mathbf{A}(\mathbf{y}_m) \end{bmatrix}}_{\mathbf{A}(\mathbf{y})} \mathbf{x} + \underbrace{\begin{bmatrix} \eta_1 \\ \vdots \\ \eta_m \end{bmatrix}}_{\eta}$$



- $\mathbf{y}$ = registration, blurring, etc., parameters
- Goal: Improve parameters $\mathbf{y}$ and compute $\mathbf{x}$

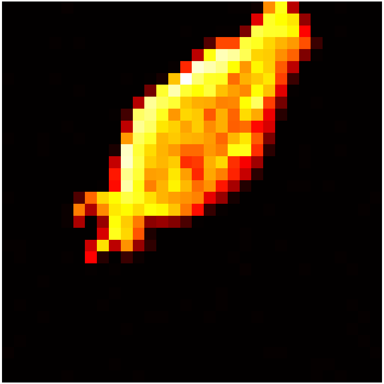
Super-resolution/Data Fusion

- $\mathbf{b}_j = \mathbf{A}(\mathbf{y}_j)\,\mathbf{x} + \eta_j$
  (collected low resolution images)

$$\underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} \mathbf{A}(\mathbf{y}_1) \\ \vdots \\ \mathbf{A}(\mathbf{y}_m) \end{bmatrix}}_{\mathbf{A}(\mathbf{y})} \mathbf{x} + \underbrace{\begin{bmatrix} \eta_1 \\ \vdots \\ \eta_m \end{bmatrix}}_{\eta}$$

- $\mathbf{y}$ = registration, blurring, etc., parameters
- Goal: Improve parameters $\mathbf{y}$ and compute $\mathbf{x}$

29−th low resolution image



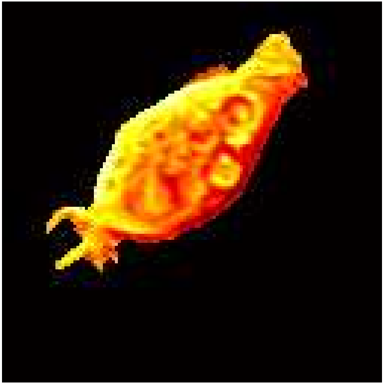
Super-resolution/Data Fusion

- $\mathbf{b}_j = \mathbf{A}(\mathbf{y}_j)\,\mathbf{x} + \boldsymbol{\eta}_j$
  (collected low resolution images)

$$\underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} \mathbf{A}(\mathbf{y}_1) \\ \vdots \\ \mathbf{A}(\mathbf{y}_m) \end{bmatrix}}_{\mathbf{A}(\mathbf{y})} \mathbf{x} + \underbrace{\begin{bmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_m \end{bmatrix}}_{\boldsymbol{\eta}}$$

- $\mathbf{y}$ = registration, blurring, etc., parameters
- Goal: Improve parameters $\mathbf{y}$ and compute $\mathbf{x}$

Reconstructed high resolution image



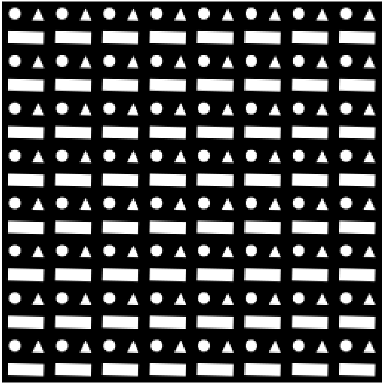
Introduction to Super Resolution

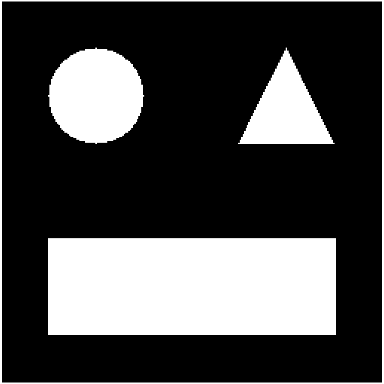
Super resolution ideas have been around for many years.
Some references:

- Andrews (Proc. IEEE, 1972)
- Park, Park, Kang (IEEE Signal Process. Mag., 2003)
- Farsiu, Robinson, Elad, Milanfar (J. Image. Systems Tech., 2004)
- Matson and Tyler (Optics Express, 2006)
- Chung, Haber, N. (Inverse Problems, 2006)

Introduction to Super Resolution

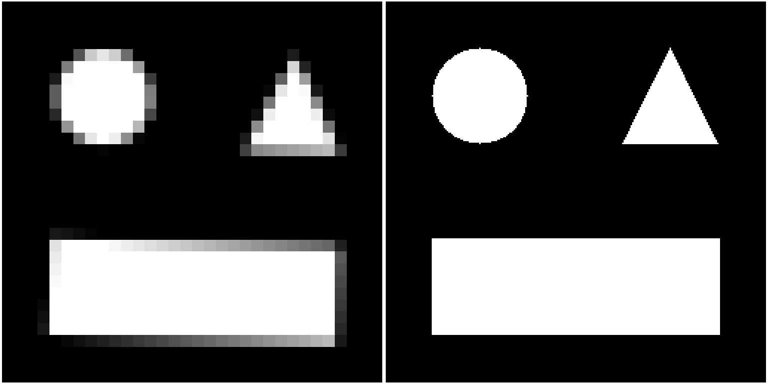
Basic idea: Combine a given set of low resolution images to get a high resolution image.





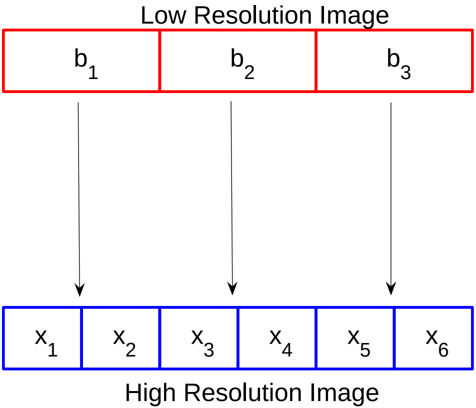
Introduction to Super Resolution

Basic idea: Combine a given set of low resolution images to get a high resolution image.



Matrix Models for Super Resolution

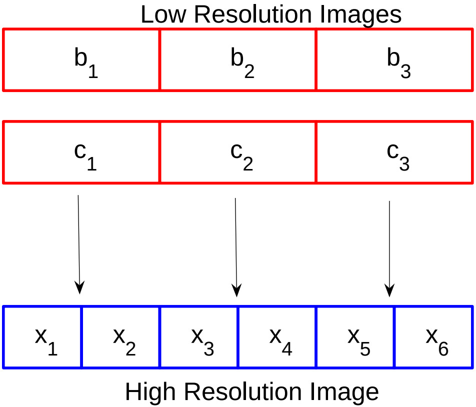
How are low and high resolution images related?



$$
\begin{aligned}
b_1 &= 0.5\, x_1 + 0.5\, x_2 \\
b_2 &= 0.5\, x_3 + 0.5\, x_4 \\
b_3 &= 0.5\, x_5 + 0.5\, x_6
\end{aligned}
$$

Matrix Models for Super Resolution

With multiple images, we can have no new information

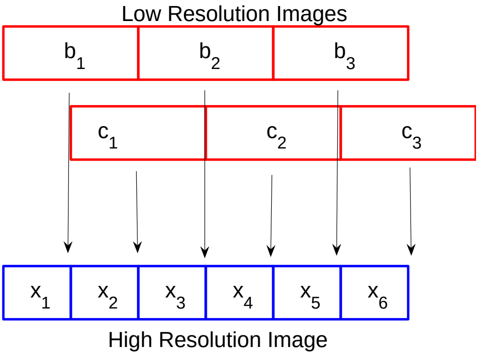


$$
\begin{aligned}
b_1 &= 0.5\,x_1 + 0.5\,x_2 \\
b_2 &= 0.5\,x_3 + 0.5\,x_4 \\
b_3 &= 0.5\,x_5 + 0.5\,x_6
\end{aligned}
$$

$$
\begin{aligned}
c_1 &= 0.5\,x_1 + 0.5\,x_2 \\
c_2 &= 0.5\,x_3 + 0.5\,x_4 \\
c_3 &= 0.5\,x_5 + 0.5\,x_6
\end{aligned}
$$

Matrix Models for Super Resolution

By offsetting multiple images, we get new information

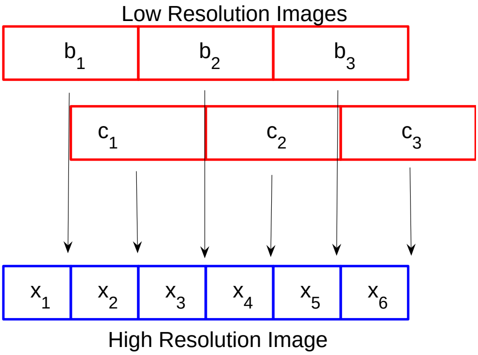


$$
\begin{aligned}
b_1 &= 0.5\,x_1 + 0.5\,x_2 \\
b_2 &= 0.5\,x_3 + 0.5\,x_4 \\
b_3 &= 0.5\,x_5 + 0.5\,x_6
\end{aligned}
$$

$$
\begin{aligned}
c_1 &= 0.5\,x_2 + 0.5\,x_3 \\
c_2 &= 0.5\,x_4 + 0.5\,x_5 \\
c_3 &= 0.5\,x_6 + 0.5\,\square
\end{aligned}
$$

Matrix Models for Super Resolution

Note that ordering of equations is arbitrary



$$
\begin{aligned}
b_1 &= 0.5\,x_1 + 0.5\,x_2 \\
c_1 &= 0.5\,x_2 + 0.5\,x_3 \\
\\
b_2 &= 0.5\,x_3 + 0.5\,x_4 \\
c_2 &= 0.5\,x_4 + 0.5\,x_5 \\
\\
b_3 &= 0.5\,x_5 + 0.5\,x_6 \\
c_3 &= 0.5\,x_6 + 0.5\,\square
\end{aligned}
$$

Matrix Models for Super Resolution

By alternating the order like this, we have the linear system:

$$\begin{bmatrix} 0.5 & 0.5 & & & & & \\ & 0.5 & 0.5 & & & & \\ & & 0.5 & 0.5 & & & \\ & & & 0.5 & 0.5 & & \\ & & & & 0.5 & 0.5 & \\ & & & & & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ \hline \square \end{bmatrix} = \begin{bmatrix} b_1 \\ c_1 \\ b_2 \\ c_2 \\ b_3 \\ c_3 \end{bmatrix}$$

where $\square$ is a boundary condition variable.

That is, we have a basic convolution model!

Matrix Models for Super Resolution

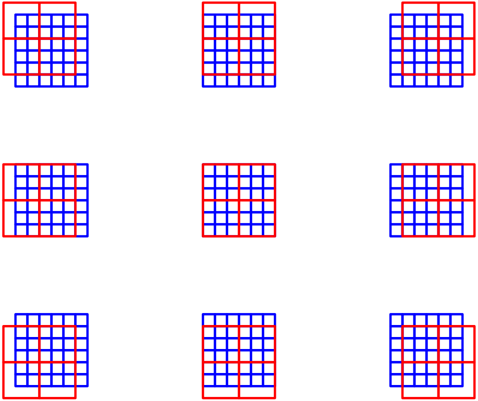
In this 1-D example, we can have:

- More low resolution images.
- Shift left, or shift right.
- Lower resolution $\Rightarrow$ more overlap of pixels. (i.e., larger PSF).
- Exact sub-pixel shifts, and with appropriate ordering, the super resolution problem is equivalent to a spatially invariant deblurring problem.
- Nonuniform and inexact shifts result in a spatially variant deblurring problem.

Matrix Models for Super Resolution

The 2-D problem is similar:

Low res. pixels (red) overlay high res. pixels (blue)



- Here we have spatially invariant, separable convolution.
- "PSF" contains weights, e.g.

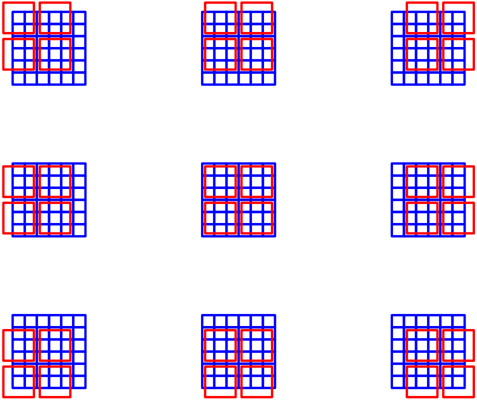
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

- Can use filtering methods for image deblurring.

Matrix Models for Super Resolution

Some practical considerations:

"active" pixel size $\neq$ actual pixel size



- Here we have spatially invariant, separable convolution.
- "PSF" contains weights, e.g.

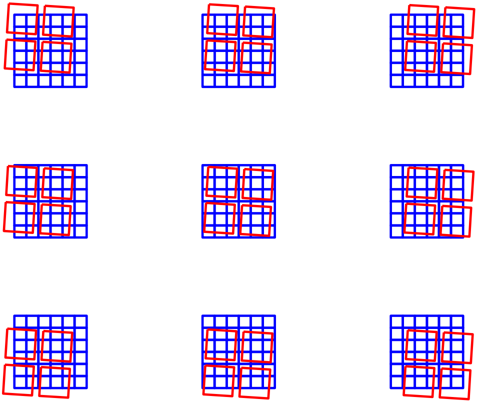
$$\begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$

- Can use filtering methods for image deblurring.

Matrix Models for Super Resolution

Some practical considerations:

linear, uniform perturbations

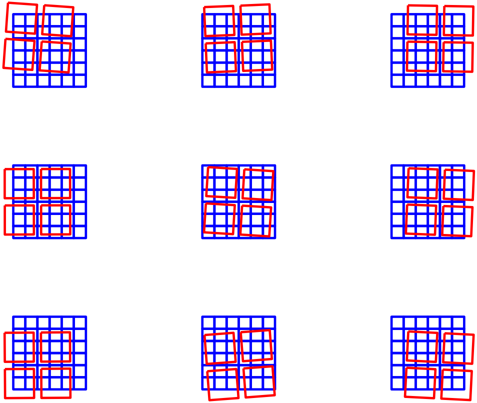


- Here we have a slightly spatially variant problem.
- Can construct a sparse matrix for $\mathbf{A}$.
- Can use iterative methods for image deblurring.

Matrix Models for Super Resolution

Some practical considerations:

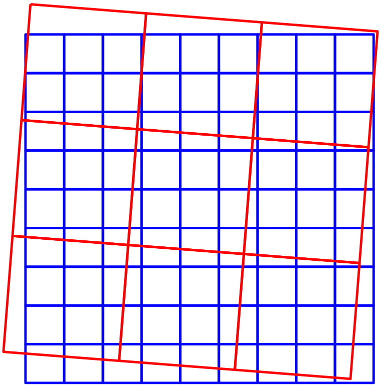
linear, nonuniform perturbations



- Here we have more variance.
- Can construct a sparse matrix for **A**, but more difficult.
- Can use iterative methods for image deblurring.

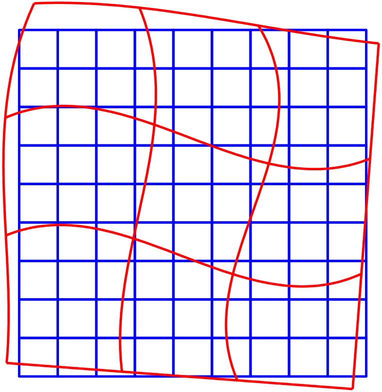
Matrix Models for Super Resolution

A closer look at pixel overlap for linear perturbations



Matrix Models for Super Resolution

We could also have nonlinear perturbations



Super Resolution Inverse Problem

Each low resolution image can be modeled as:

$$\mathbf{b}_k = \mathbf{D}\mathbf{S}(\mathbf{y}_k)\mathbf{x} + \boldsymbol{\eta}_k$$

where

- $\mathbf{x}$ is the unknown high resolution image
- $\mathbf{S}(\mathbf{y}_k)$ model distortion operation (e.g., shifting, rotation, etc.)
- $\mathbf{D}$ models the subsampling (decimation) operation
- $\mathbf{b}_k$ are the given low resolution images

Super Resolution Inverse Problem

The inverse problem is:

$$\mathbf{b} = \mathbf{A}(\mathbf{y})\mathbf{x} + \boldsymbol{\eta}$$

where

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix}, \quad \mathbf{A}(\mathbf{y}) = \begin{bmatrix} \mathbf{D}\mathbf{A}(\mathbf{y}_1) \\ \mathbf{D}\mathbf{A}(\mathbf{y}_2) \\ \vdots \\ \mathbf{D}\mathbf{A}(\mathbf{y}_m) \end{bmatrix}, \quad \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \vdots \\ \boldsymbol{\eta}_m \end{bmatrix}$$

Important Considerations

The inverse problem is:

$$\mathbf{b} = \mathbf{A}(\mathbf{y})\mathbf{x} + \boldsymbol{\eta}$$

The main goal is to reconstruct $\mathbf{x}$.

However, $\mathbf{y}$ is also generally not known exactly.

- Good reconstruction of $\mathbf{x}$ depends on good approximation of $\mathbf{y}$.
- Calibration may used to approximate $\mathbf{y}$.
- $\mathbf{y}$ has significantly fewer parameters than does $\mathbf{x}$
  - horizontal/vertical shifts $\Rightarrow$ each $\mathbf{y}_k$ defined by two parameters
  - more complicated movement might be defined by linear affine transformation $\Rightarrow$ each $\mathbf{y}_k$ defined by six parameters

Separable Inverse Problems

Consider the forward problem:

$$\mathbf{b} = \mathbf{A}(\mathbf{y}_{\text{true}})\mathbf{x}_{\text{true}} + \boldsymbol{\eta}$$

and suppose we consider the simple regularized objective function:

$$F(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{b} - \mathbf{A}(\mathbf{y})\mathbf{x}\|_2^2 + \frac{\alpha^2}{2}\|\mathbf{x}\|_2^2$$

Three basic approaches to attack this problem:

- Decoupled approach.
- Fully coupled approach.
- Partially coupled approach (Variable Projection).

Separable Inverse Problems

Decoupled approach: Block Coordinate Descent:

$$
\begin{array}{l}
\mathbf{y}^{(0)} = \text{initial guess} \\
\texttt{for} \quad k = 0, 1, 2, \ldots \\
\quad \bullet\ \mathbf{x}^{(k)} = \quad \min_{\mathbf{x}} F\left(\mathbf{x}, \mathbf{y}^{(k)}\right) \\
\quad \bullet\ \mathbf{y}^{(k+1)} = \quad \min_{\mathbf{y}} F\left(\mathbf{x}^{(k)}, \mathbf{y}\right) \\
\texttt{end}
\end{array}
$$

Separable Inverse Problems

Fully coupled approach: Joint Optimization:

- Let $\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$
- Define $F(\mathbf{z}) = F\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}\right) = F(\mathbf{x}, \mathbf{y})$
- Use an optimization method (e.g., Gauss-Newton, nonlinear CG, ...) to solve

$$\min_{\mathbf{z}} F(\mathbf{z})$$

Separable Inverse Problems

Partially coupled approach: Variable Projection:

- Let $\mathbf{x}(\mathbf{y}) = \mathbf{A}^{\dagger}(\mathbf{y})\mathbf{b}$
  - e.g., $\mathbf{A}^{\dagger}(\mathbf{y}) = \left(\mathbf{A}^T(\mathbf{y})\mathbf{A}(\mathbf{y}) + \alpha^2\mathbf{I}\right)^{-1}\mathbf{A}^T(\mathbf{y})$
- Consider the reduced cost functional:

$$F_r(\mathbf{y}) = \frac{1}{2}\|\mathbf{b} - \mathbf{A}(\mathbf{y})\mathbf{A}^{\dagger}(\mathbf{y})\mathbf{b}\|_2^2 + \frac{\alpha^2}{2}\|\mathbf{A}^{\dagger}(\mathbf{y})\|_2^2$$

- Use an optimization method (e.g., Gauss-Newton, nonlinear CG, ...) to solve

$$\min_{\mathbf{y}} F(\mathbf{y})$$

Remark: Although this has advantages:

- Exploits linear dependence on $\mathbf{x}$
- Reduces the number of explicit unknowns (e.g., in MFBD from $mp + n$ to mp).