Super-Resolution Reconstruction of Images - Static and Dynamic Paradigms

February 2002

Michael Elad*

The Scientific Computing and Computational Mathematics Program - Stanford

* Joint work with Prof. Arie Feuer – The Technion, Haifa Israel, Prof. Yacob Hel-Or – IDC, Herzelia, Israel, Tamir Sagi - Zapex-Israel.
Static Versus Dynamic Super-Resolution

Definitions and Activity Map
**Basic Super-Resolution Idea**

**Given:** A set of degraded (warped, blurred, decimated, noised) images:

**Required:** Fusion of the measurements into a higher resolution image/s
Static Super-Resolution (SSR)

\[ \hat{X} = f\{Y_1, Y_2, Y_3, \ldots, Y_N\} \]

High Resolution
Reconstructed Image
Dynamic Super-Resolution (DSR)

Low Resolution Measurements

\[
\hat{X}(t) = f\{Y(t), Y(t-1), \ldots\}
\]

Dynamic Super-Resolution Algorithm

High Resolution Reconstructed Images
## Other Work In this Field

<table>
<thead>
<tr>
<th>People</th>
<th>Place</th>
<th>Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peleg, Irani, Werman, Keren, Schweitzer</td>
<td>HUJI</td>
<td>1987-1994</td>
</tr>
<tr>
<td>Kim, Bose, Valenzuela</td>
<td>Penn. State</td>
<td>1990-1993</td>
</tr>
<tr>
<td>Ur &amp; Gross</td>
<td>TAUI</td>
<td>1992-1993</td>
</tr>
<tr>
<td>Elad, Feuer, Sagi, Hel-Or</td>
<td>Technion</td>
<td>1995-2001</td>
</tr>
<tr>
<td>Schutlz, Stevenson, Borman</td>
<td>Notre-Dame</td>
<td>1995-1999</td>
</tr>
<tr>
<td>Shekarforush, Berthod, Zerubia, Werman</td>
<td>INRIA-France</td>
<td>1995-1999</td>
</tr>
<tr>
<td>Katsaggelos, Tom, Galatsanos</td>
<td>Northwestern</td>
<td>1995-1999</td>
</tr>
<tr>
<td>Shah, Zachor</td>
<td>Berkeley</td>
<td>1996-1999</td>
</tr>
<tr>
<td>Baker, Kanade</td>
<td>CMU</td>
<td>1999-2001</td>
</tr>
</tbody>
</table>

* This table probably does mis-justice to someone - no harm meant

Methods which relate also to DSR paradigm. All others deal with SSR.
Our Work In this Field


All found in http://sccm.stanford.edu/~elad
Super-Resolution Basics

Intuition and Relation to Sampling theorems
For a given band-limited image, the Nyquist sampling theorem states that if a uniform sampling is fine enough (≥D), perfect reconstruction is possible.
Simple Example

Due to our limited camera resolution, we sample using an insufficient 2D grid
However, we are allowed to take a second picture and so, shifting the camera ‘slightly to the right’ we obtain
Similarly, by shifting down we get a third image.
And finally, by shifting down and to the right we get the fourth image.
Simple Example - Conclusion

It is trivial to see that interlacing the four images, we get that the desired resolution is obtained, and thus perfect reconstruction is guaranteed.

This is Super-Resolution in its simplest form
Uncontrolled Displacements

In the previous example we counted on exact movement of the camera by $D$ in each direction.

What if the camera displacement is uncontrolled?
Uncontrolled Displacements

It turns out that there is a sampling theorem due to Yen (1956) and Papulis (1977) covering this case, guaranteeing perfect reconstruction for periodic uniform sampling if the sampling density is high enough (1 sample per each D-by-D square).
In the previous examples we restricted the camera to move horizontally/vertically parallel to the photograph object.

What if the camera rotates? Gets closer to the object (zoom)?
Uncontrolled Rotation/Scale/Disp.

There is no sampling theorem covering this case.
Further Complications

1. Sampling is not a point operation – there is a blur

2. Motion may include perspective warp, local motion, etc.

3. Samples may be noisy – any reconstruction process must take that into account.
Static Super-Resolution

The creation of a single improved image, from the finite measured sequence of images
SSR - The Model

\[
\begin{align*}
Y_k &= D_k H_k F_k X + V_k, \quad V_k \sim N\left(0, W_k^{-1}\right) \\
Y_1 &= D_1 H_1 F_1 X + V_1 \\
Y_N &= D_N H_N F_N X + V_N
\end{align*}
\]
The Warp As a Linear Operation

Per every point in X find a matching point in Z

\[
\begin{bmatrix}
  x_1 \\
  \vdots \\
  x_j \\
  \vdots \\
  x_N \\
\end{bmatrix}
\begin{bmatrix}
  1 \\
  0 \\
  0 \\
  \vdots \\
  0 \\
  1 \\
\end{bmatrix}
\begin{bmatrix}
  z_1 \\
  \vdots \\
  z_j \\
  \vdots \\
  z_N \\
\end{bmatrix}
\]

\( F[j,i] = 1 \)
Model Assumptions

We assume that the images $Y_k$ and the operators $H_k$, $D_k$, $F_k$, & $W_k$ are known to us, and we use them for the recovery of $X$.

$Y_k$ – The measured images (noisy, blurry, down-sampled ..)

$H_k$ – The blur can be extracted from the camera characteristics

$D_k$ – The decimation is dictated by the required resolution ratio

$F_k$ – The warp can be estimated using motion estimation

$W_k$ – The noise covariance can be extracted from the camera characteristics
The Model as One Equation

\[
\begin{cases}
Y_k = D_k H_k F_k X + V_k, \\
V_k \sim N\{0, W_k^{-1}\}
\end{cases}
\]

\[
\left[ \begin{array}{c}
Y_1 \\
Y_2 \\
\vdots \\
Y_N \\
\end{array} \right] = 
\left[ \begin{array}{c}
D_{1H_1F_1} \\
D_{2H_2F_2} \\
\vdots \\
D_{NH_NF_N} \\
\end{array} \right] X + 
\left[ \begin{array}{c}
V_1 \\
V_2 \\
\vdots \\
V_N \\
\end{array} \right] \sim N\left( \left[ \begin{array}{cccc}
w_1 & 0 & \cdots & 0 \\
0 & w_2 & \cdots & 0 \\
0 & 0 & \cdots & w_N \\
\end{array} \right]^{-1} \right) 
\]
A Thumb Rule on Desired Resolution

In the noiseless case we have

\[
\begin{bmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_N
\end{bmatrix} = 
\begin{bmatrix}
D_1 H_1 F_1 \\
D_2 H_2 F_2 \\
\vdots \\
D_N H_N F_N
\end{bmatrix} \begin{bmatrix}
X
\end{bmatrix}
\]

Clearly, this linear system of equations should have more equations than unknowns in order to make it possible to have a unique Least-Squares solution.

Example: Assume that we have N images of M-by-M pixels, and we would like to produce an image X of size L-by-L. Then – \( L \leq \sqrt{N \cdot M} \)
The Maximum-Likelihood Approach

Which \( X \) would be such that when fed to the above system it yields a set \( Y_k \) closest to the measured images?
SSR - ML Reconstruction (LS)

Minimize: \[ \varepsilon_{ML}^2(X) = \sum_{k=1}^{N} \| Y_k - D_k H_k F_k X \|_W^2 \]

Thus, require: \[ \frac{\partial \varepsilon_{ML}^2(X)}{\partial X} = 0 \]

\[ \begin{cases} R = \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k D_k H_k F_k \\ P = \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k Y_k \end{cases} \]

\[ R \hat{X} = P \]
SSR - MAP Reconstruction

Add a term which penalizes for the solution image quality

\[ \varepsilon_{MAP}^2(X) = \sum_{k=1}^{N} \left\| Y_k - D_k H_k F_k X \right\|_{W_k}^2 + \lambda A\{X\} \]

Possible Prior functions - Examples:

1. \( A\{X\} = X^T S^T W(X_0) S X \) - simple spatially adaptive,
2. \( A\{X\} = \rho\{S X\} \) - M estimator (robust functions),

Note: Convex prior guarantees convex programming problem
Iterative Reconstruction

Assuming the prior $A\{X\} = X^T S^T WS X$ is used

$$
\begin{align*}
R &= \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k D_k H_k F_k + \lambda S^T WS \\
P &= \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k Y_k
\end{align*}
$$

For $\hat{X} : [1000 \times 1000]$, the matrix $R$ is sparse $R \in \mathbb{M}^{10^6 \times 10^6}$

OPTION: Using the SD algorithm (10-15 iterations are enough)

$$
\hat{X}_{j+1} = \hat{X}_j - \mu \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k \left[ Y_k - D_k H_k F_k \hat{X}_j \right] - \mu \lambda S^T WS \hat{X}_j
$$
Image-Based Processing

All the above operations can be interpreted as operations performed on images.

AND THUS

There is no actual need to use the Matrix-Vector notations as shown here. This notations is important for the development of the algorithm

SD* Iteration:  \[ \hat{X}_{j+1} = \hat{X}_j - \mu \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k [Y_k - D_k H_k F_k \hat{X}_j] - \mu \lambda S^T W S \hat{X}_j \]
SSR – Simpler Problems

\[ \hat{X} = \left[ \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k D_k H_k F_k + \lambda S^T W S \right]^{-1} \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k Y_k \]
## SSR – Simpler Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single image de-noising</td>
<td>$\hat{X} = [I + \lambda S^T W S]^{-1} Y$</td>
</tr>
<tr>
<td>Single image restoration</td>
<td>$\hat{X} = [H^T H + \lambda S^T W S]^{-1} H^T Y$</td>
</tr>
<tr>
<td>Single image scaling</td>
<td>$\hat{X} = [D^T D + \lambda S^T W S]^{-1} D^T Y$</td>
</tr>
<tr>
<td>Motion compensation average</td>
<td>$\hat{X} = \left[ \sum_{k=1}^{N} F_k^T F_k + \lambda S^T W S \right]^{-1} \sum_{k=1}^{N} F_k^T Y_k$</td>
</tr>
</tbody>
</table>

Using $A\{\hat{X}\} = \hat{X}^T S^T W S \hat{X}$
Example 1

Synthetic case:
From a single image create 9 3:1 images this way
Example 1

Synthetic case:
9 images, no blur, 1:3 ratio

One of the low-resolution images
The higher resolution original
The reconstructed result
Example 2

16 images, ratio 1:2, PSF - assumed to be Gaussian with $\sigma=2.5$

Taken from one of the given images

Taken from the reconstructed result
Dynamic Super-Resolution

Low Quality Movie In – High Quality Movie Out
Dynamic Super-Resolution (DSR)

Low Resolution Measurements

\[ \{Y(t)\}_t \]

High Resolution Reconstructed Images

\[ \{\hat{X}(t)\}_t \]

\[ \hat{X}(t) = f\{Y(t), Y(t-1), \ldots\} \]
Modeling the Problem

\[ \{Y(t)\}_t \]

\[ Y(t-k) = M(t,k)X(t) + N(t,k) \]

\[ \{\hat{X}(t)\}_t \]
DSR – Proposed Model

\[ Y(t - k) = M(t, k)X(t) + N(t, k) \]

\[
\begin{cases}
Y(t - k) = D\tilde{F}(t, k)X(t) + N(t, k) \\
N(t, k) \sim N\left(0, \lambda^{-k}W^{-1}\right) \text{ where } 0 < \lambda < 1 \\
\text{and } \tilde{F}(t, k) = F(t - k + 1) \cdots F(t - 1)F(t) \\
\end{cases}
\]
DSR – From Model to ML

- The DSR problem is referred to as a long sequence of SSR problems.
- Thus, Our model is
  \[ Y(t - k) = DHF(t, k)X(t) + N(t, k) \]

  \[ N(t, k) \sim N\left(0, \lambda^{-k}W^{-1}\right) \]

  where \(0 < \lambda < 1\)

  and \(\hat{F}(t, k) = F(t - k + 1) \cdots F(t - 1)F(t)\)

- Using ML approach

  \[ \varepsilon^2(X(t), t) = \sum_{k=0}^{t-1} \lambda^k \left\| Y(t - k) - DHF(t, k)X(t) \right\|_W^2 \]

  and this function should be minimized per each \(t\).
Minimizing $\varepsilon^2(X(t), t) = \sum_{k=0}^{t-1} \lambda^k \| Y(t-k) - DH\hat{F}(t, k)X(t) \|^2_w$

amounts to solving the linear set of equations $L(t)\hat{X}(t) = Z(t)$

where

$L(t) = \sum_{k=0}^{t-1} \lambda^k \left[ DH\hat{F}(t, k) \right]^T W \left[ DH\hat{F}(t, k) \right]$

$Z(t) = \sum_{k=0}^{t-1} \lambda^k \left[ DH\hat{F}(t, k) \right]^T W Y(t-k)$

Note that (apart from the need to solve the linear set), one has to compute $L$ and $Z$ per each $t$ all over again, and the summations length grow linearly in $t$. 
Recursive Representation

\[
L(t) = \sum_{k=0}^{t-1} \lambda^k \left[ DH\tilde{F}(t, k) \right]^T W \left[ DH\tilde{F}(t, k) \right]
\]

\[
Z(t) = \sum_{k=0}^{t-1} \lambda^k \left[ DH\tilde{F}(t, k) \right]^T W Y(t-k)
\]

Simplifies to (Using \( \tilde{F}(t, k) = F(t-k+1) \ldots F(t) F(t) \))

\[
L(t) = \lambda F^T(t) L(t-1) F(t) + H^T W H
\]

\[
Z(t) = \lambda F^T(t) Z(t-1) + H^T W Y(t)
\]
Alternative Approach

- Instead of continuing with the previous model and recursive representation, we adopt a different point of view.

- The new point of view is based on State-Space modeling of our problems.

- This new model leads to better-understanding of the required algorithmic steps towards an efficient solution.

- The eventual expressions with the alternative method are exactly the same as the ones shown previously.
The System’s Equation

\[ \mathbf{X}(t) = \mathbf{G}(t) \mathbf{X}(t-1) + \mathbf{V}(t) \]

- \( \mathbf{X}(t) \) - High-resolution image
- \( \mathbf{G}(t) \) - Warp operation
- \( \mathbf{V}(t) \) - Sequence innovation

Assumed:\[ N\{0, \mathbf{Q}(t)\} \]
DSR - The Model (2)

The Measurements Equation

\[
\begin{bmatrix}
Y(t) \\
0
\end{bmatrix} = \begin{bmatrix}
DH(t) & S
\end{bmatrix}
\begin{bmatrix}
X(t) \\
U(t)
\end{bmatrix} + \begin{bmatrix}
N(t) \\
U(t)
\end{bmatrix}
\]

\(Y(t)\) - Measured image  
\(H(t)\) - Blur  
\(D\) - Decimation  
\(N(t)\) - additive noise  
\(\sim N\{0, W^{-1}(t)\}\)  
\(S\) - Laplacian  
\(U(t)\) - Non-smooth.  
\(\sim N\{0, R^{-1}(t)\}\)
These two equations form a **Spatio-Temporal Prior** forcing spatial smoothness & temporal motion compensated smoothness.
In order to estimate $X(t)$ in time, we need to apply Kalman Filter (KF)

The model is given in a State-Space form

\[
\begin{align*}
\dot{X}(t) &= G(t)X(t-1) + V(t) \\
Y_A(t) &= H_A(t)X(t) + N_A(t)
\end{align*}
\]

where \( V(t) \sim N\{0, Q^{-1}(t)\} \)
\( N_A(t) \sim N\{0, W^{-1}(t)\} \)

In order to estimate $X(t)$ in time, we need to apply **Kalman Filter (KF)**

**The basic idea:**

1. Since all the inputs are Gaussians, so is $X(t)$
2. We know all about $X(t)$ if its two first moments are known - \( X(t) \sim N\{\hat{X}(t), \hat{P}(t)\} \)
1. We start by knowing the pair \( (\hat{X}(t-1), \hat{P}(t-1)) \)

2. Based on \( X(t) = G(t)X(t-1) + V(t) \) we get the Prediction Equations:

\[
\tilde{P}(t) = G(t)\hat{P}(t-1)G^T(t) + Q^{-1}(t) \\
\tilde{X}(t) = G(t)\hat{X}(t-1)
\]

3. Based on \( Y_A(t) = H_A(t)X(t) + N_A(t) \) we get the Update Equations:

\[
\hat{P}(t) = \left[ \tilde{P}^{-1}(t) + H_A^T(t)W_A(t)H_A(t) \right]^{-1} \\
\hat{X}(t) = \hat{P}(t)\tilde{X}(t) + H_A^T(t)W_A(t)Y_A(t)
\]
KF: Information Pair

Information pair is defined by $\langle \hat{Z}(t), \hat{L}(t) \rangle = \langle \hat{P}^{-1}(t)\hat{X}(t), \hat{P}^{-1}(t) \rangle$

The recursive equations become:

Interpolation:

$$\tilde{L}(t) = \left[ G(t)\hat{L}^{-1}(t-1)G^T(t) + Q^{-1}(t) \right]^{-1}$$
$$\tilde{Z}(t) = \tilde{L}(t)G(t)\hat{L}^{-1}(t-1)\tilde{Z}(t-1)$$

Update:

$$\hat{L}(t) = \tilde{L}(t) + H_A^T(t)W_A(t)H_A(t)$$
$$\hat{Z}(t) = \tilde{Z}(t) + H_A^T(t)W_A(t)Y_A(t)$$

Presumably, there is nothing to gain in using the information pair, over the mean-covariance pair.
Information Pair Is Better!!
(for our application)

1. Experimental results indicate that the information matrix is sparser:
\[
\begin{pmatrix}
\end{pmatrix}^{-1} = Q_{GLGL}
\]

2. We intend to avoid the use of \( Q(t) \). Therefore, it is natural to achieve simplifying the equation
\[
\tilde{L}(t) = \left[ G(t) \tilde{L}^{-1}(t-1) G^T(t) + Q^{-1}(t) \right]^{-1}
\]
while approximating \( Q(t) \).
Avoiding $Q(t)$

Instead of using

$$\tilde{L}(t) = \left[ G(t)\hat{L}^{-1}(t - 1)G^T(t) + Q^{-1}(t) \right]^{-1}$$

Approximate

$$Q^{-1}(t) \approx \alpha(t)G(t)\hat{L}^{-1}(t - 1)G^T(t)$$

and obtain that

$$\tilde{L}(t) = \frac{1}{1 + \alpha(t)} F^T(t)\hat{L}(t - 1)F(t) \quad \left[ G^{-1}(t) = F(t) \right]$$

\[
\hat{L}(t) = \lambda(t)F^T(t)\hat{L}(t - 1)F(t) + H_A^T(t)W_A(t)H_A(t)
\]

\[
\hat{Z}(t) = \lambda(t)F^T(t)\hat{Z}(t - 1) + H_A^T(t)W_A(t)Y_A(t)
\]
The Pseudo-RLS Algorithm

1. Initialize: \( \hat{L}(0) = \varepsilon^2 I, \quad \hat{Z}(0) = 0, \quad \hat{X}(0) = 0 \)

2. For \( t > 0, \)

- Update the information pair

\[
\hat{L}(t) = \lambda(t) F^T(t) \hat{L}(t-1) F(t) + H_A^T(t) W_A(t) H_A(t)
\]
\[
\hat{Z}(t) = \lambda(t) F^T(t) \hat{Z}(t-1) + H_A^T(t) W_A(t) Y_A(t)
\]

- Compute the output by \( \hat{X}(t) = \hat{L}^{-1}(t) \hat{Z}(t) \)

Problem: Need to invert the information matrix
The R-SD Algorithm

1. Initialize: $\hat{L}(0) = \varepsilon^2 I$, $\hat{Z}(0) = 0$, $\hat{X}(0) = 0$

2. For $t > 0$,
   - Update the information pair, as before
   - Compute the output by R-SD iterations:
     \[
     \hat{X}_0(t) = G(t)\hat{X}_R(t-1)
     \]
     and for $k=1,2, \ldots, R$:
     \[
     \hat{X}_{k+1}(t) = \hat{X}_k(t) - \mu [\hat{L}(t)\hat{X}_k(t) - \hat{Z}(t)]
     \]

Adopted from the assumed model

Note: $\hat{X}_R(t) \neq \hat{L}^{-1}(t)\hat{Z}(t)$ but error does not propagate
Dynamic Super-Resolution

Low Resolution Measurements

\[ \hat{X}(t) = f\left\{ Y(t), \hat{X}(t-1) \right\} \]

High Resolution Reconstructed Images

Dynamic Super-Resolution Algorithm
The R-LMS Algorithm

1. Initialize: \( \hat{X}(0) = 0 \)

2. For \( t > 0 \),

\[ \text{_committee output by } R \text{-SD iterations using the intermediate information pair:} \]

\[ \hat{X}_0(t) = G(t) \hat{X}_R(t-1) \]

and for \( k=1,2, \ldots, R \):

\[ \hat{X}_{k+1}(t) = \hat{X}_k(t) - \mu H_A(t)^T W_A(t) \left[ H_A(t) \hat{X}_k(t) - Y_A(t) \right] \]

Also obtained if \( \hat{X}_R(t-1) \approx L^{-1}(t-1) \hat{Z}(t-1) \)

or if \( \lambda(t) \) is set to zero
Under some very reasonable assumptions, it is PROVEN that the information matrix remains SPARSE.

\[ \hat{L}(t) = \lambda(t)F^T(t)\hat{L}(t-1)F(t) + M(t) \]
1. Bounds on the dynamic estimation error for the proposed Kalman Filter approximations (the P-RLS, the R-SD and the R-LMS) are obtained.

2. An important role in these convergence theorems plays the term

\[ \| \hat{\mathbf{X}}_{\text{PRLS}}(t) - \mathbf{G}(t) \hat{\mathbf{X}}_{\text{PRLS}}(t-1) \| \]

which stands for the amount of variation (innovative data) that exists in the sequence. The higher this term, the higher is the expected error.
Results - Part 1

Dynamic Estimation Comparison - Low dimension (N=100) synthetic case

MSE versus iterations - A Comparison
### Results - Part 2

Higher dimension (N=2500) synthetic image sequences

<table>
<thead>
<tr>
<th></th>
<th>1&lt;sup&gt;st&lt;/sup&gt;</th>
<th>25&lt;sup&gt;th&lt;/sup&gt;</th>
<th>50&lt;sup&gt;th&lt;/sup&gt;</th>
<th>75&lt;sup&gt;th&lt;/sup&gt;</th>
<th>100&lt;sup&gt;th&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>The original sequence: Image size: 50 by 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Measured sequence: 3 by 3 uniform blurring, 2:1 decimation, noise $\sigma = 5$.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Bilinear interpolation of the measured sequence</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>The 5-LMS algorithm's output, no regularization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>The 5-LMS algorithm's output, with regularization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>The 5-SD algorithm's output, with regularization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** the motion and blur operations are assumed to be known apriori
Sequence 1 [Displacement+zoom]

Measurements

Bilinear Interpolation

5-LMS no Regularization

5-LMS + Regularization

5-SD + Regularization
Sequence 1
[Pure translation]

Measurements

Bilinear Interpolation

5-LMS no Regularization

5-LMS + Regularization

5-SD + Regularization
Sequence 1
[Pure rotation]

Measurements

Bilinear Interpolation

5-LMS no Regularization

5-LMS + Regularization

5-SD + Regularization

1st    25th    50th    75th    100th
Conclusions

- Both Static and Dynamic super-resolution paradigms are presented, along with their solutions.
- Very simple yet general models are proposed for both problems.
- The SSR problem is presented as a classic inverse problem, and treated as such.
- The DSR problem is shown to require KF for its solution. Due to the dimensions involved, approximations are developed and analyzed.
- Simulations show promising results, both for the SSR and the DSR.
- Motion estimation is a bottleneck in the recovery processes.
Fast SSR (1) - A Special Case

What if the same camera is used and the motion is pure translational?
SSR - The Model

\[
\begin{align*}
\mathbf{Y}_k &= \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \mathbf{X} + \mathbf{V}_k, \\
\mathbf{V}_k &\sim \mathcal{N}\left(\mathbf{0}, \mathbf{W}_k^{-1}\right)
\end{align*}
\]
The Model as One Equation

\[
\begin{align*}
\{ Y_k = D_k H_k F_k X + V_k, \quad V_k &\sim N\{0, W_k^{-1}\} \} \\
_{k=1}^N
\end{align*}
\]
Iterative Reconstruction

\[
R = \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k D_k F_k \\
\]

\[
P = \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k Y_k \\
\]

For \( \hat{X} : [1000 \times 1000] \), the matrix \( R \) is sparse \( R \in M^{10^6 \times 10^6} \)

OPTION: Using the SD algorithm (10-15 iterations are enough)

\[
\hat{X}_{j+1} = \hat{X}_j - \mu \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k \left[ Y_k - D_k H_k F_k \hat{X}_j \right] \\
\]
Basic Assumptions

\( H_k = H \) – The blur operation is the same for all the images and it is a linear-space-invariant operation, i.e., it has a block-Circulant form.

\( D_k = D \) – The decimation operation is the same for all the images and it is a uniform sub-sampling operator.

\( F_k \) – The warps are all pure translations, and thus all have a block-Circulant form. Moreover, we assume a nearest-neighbor representation (one non-zero entry in each row and it is ‘1’).

\( W_k = cI \) – The noise is Gaussian and white and thus the covariance matrix is the identity matrix up to some constant.
Using the Iterative SD

\[ \hat{X}_{j+1} = \hat{X}_j - \mu \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k \left[ Y_k - D_k H_k F_k \hat{X}_j \right] \]

\[ \hat{X}_{j+1} = \hat{X}_j - \mu \sum_{k=1}^{N} F_k^T H^T D^T \left[ Y_k - D H F_k \hat{X}_j \right] = \]

\[ = \hat{X}_j - \mu H^T \sum_{k=1}^{N} F_k^T D^T \left[ Y_k - D F H \hat{X}_j \right] \]

where we use the fact that

block-Circulant matrices commute
Important Shortcut

Define $\hat{Z}_j = H\hat{X}_j$ and get

$$\hat{X}_{j+1} = \hat{X}_j - \mu H^T \sum_{k=1}^{N} F_k^T D^T [Y_k - DF_k H \hat{X}_j]$$

Then

$$\hat{Z}_{j+1} = \hat{Z}_j - \mu HH^T \sum_{k=1}^{N} F_k^T D^T [Y_k - DF_k \hat{Z}_j] =$$

$$= \hat{Z}_j - \mu HH^T \left[ \sum_{k=1}^{N} F_k^T D^T Y_k - \sum_{k=1}^{N} F_k^T D^T DF_k \hat{Z}_j \right] = \hat{Z}_j - \mu HH^T \left( \tilde{P} - \tilde{R} \hat{Z}_j \right)$$

$$= \tilde{P} \quad = \tilde{R}$$
Descent Direction - Theory

- Given the quadratic function* \( f\{\mathbf{x}\} = \frac{1}{2} \mathbf{x}^T \mathbf{R} \mathbf{x} - \mathbf{p}^T \mathbf{x} + c \), it's optimal solution satisfies \( \mathbf{R} \hat{\mathbf{x}}_{opt} = \mathbf{p} \).

- Any algorithm of the form \( \hat{\mathbf{x}}_{j+1} = \hat{\mathbf{x}}_j - \alpha \mathbf{M} (\mathbf{R} \hat{\mathbf{x}}_j - \mathbf{p}) \) converges to \( \hat{\mathbf{x}}_{opt} \) for sufficiently small \( \alpha \) and \( \mathbf{M} > 0 \).

- In our case \( \mathbf{M} = \mathbf{H} \mathbf{H}^T \) (positive semi-definite). It means that the error \( \hat{\mathbf{x}}_j - \hat{\mathbf{x}}_{opt} \) in the null space of \( \mathbf{M} \) cannot converge.

* \( \mathbf{R} \) is assumed to be positive definite

Is it a Problem?
Positive Semi-definite $M$

$$\hat{x}_{j+1} = \hat{x}_j - \alpha M (\tilde{R} \hat{x}_j - \tilde{P})$$

$$(\hat{x}_{j+1} - \hat{x}_{\text{opt}}) = (I - \alpha M \tilde{R})^{j+1} (\hat{x}_0 - \hat{x}_{\text{opt}})$$

If $v$ is in the null-space of $M$, then a vector $u = \tilde{R}^{-1} v$

is in the null-space of $M \tilde{R}$. For such a vector we get

$$(I - \alpha M \tilde{R})^{j+1} u = u$$
Positive Semi-definite M

\[ \hat{\mathbf{x}}_0 - \hat{\mathbf{x}}_{\text{opt}} = \hat{\mathbf{e}}_0 + \hat{\mathbf{f}}_0 \]

In the null-space of \( \mathbf{M\tilde{R}} \)

Orthogonal to the null-space of \( \mathbf{M\tilde{R}} \)

\[ \hat{\mathbf{e}}_{j+1} + \hat{\mathbf{f}}_{j+1} = (\mathbf{I} - \alpha \mathbf{M\tilde{R}})^{j+1} (\hat{\mathbf{e}}_0 + \hat{\mathbf{f}}_0) = (\mathbf{I} - \alpha \mathbf{M\tilde{R}})^{j+1} \hat{\mathbf{e}}_0 + \hat{\mathbf{f}}_0 \]

The null-space of \( \mathbf{M\tilde{R}} \) is characterized by very high frequencies (since \( \mathbf{M} = \mathbf{H}\mathbf{H}^T \) and \( \mathbf{H} \) is a low-pass-filter).

Thus, no-convergence there is of no consequence, and this is especially true if proper initialization is used.
What is P?

\[ \tilde{P} = \sum_{k=1}^{N} F_k^T D_k^T Y_k \]

It turns out that this is a motion-compensated average of the input images.
What is R?

\[ \tilde{R} = \sum_{k=1}^{N} F_k^T D_k^T D F_k \]  

Huge matrix, but due to our assumptions …

A. This matrix is a **diagonal matrix**,  
B. Its main diagonal entries are all integers,  
C. The \([j,j]\) entry represents the count of contributing pixels from the Y-sequence to the j-th pixel in X, and  
D. We hereby assume that sufficient measurements are given and thus \( \forall j, \tilde{R}[j,j] \geq 1 \)
To Conclude

\[ \hat{Z}_{j+1} = \hat{Z}_j - \mu HH^T (\tilde{P} - \tilde{R} \hat{Z}_j) \]

\[ \hat{Z}_{\text{opt}} = \tilde{R}^{-1} \tilde{P} \] and it is easy to compute this solution – One division by integer per pixel !!!!

Having found \( \hat{Z}_{\text{opt}} \), since it is defined by

\[ \hat{Z}_j = H \hat{X}_j \]

We have to apply a classic image restoration procedure to recover \( \hat{X}_{\text{opt}} \) (can be done without iterations).
Should We be Surprised?

Every low-quality image fills some pixels in the higher resolution grid.

Some pixels will be filled more than once – good for noise removal.
Adaptive Non-Iterative Restoration

Using \( \hat{X} = \left( H^T H + \lambda S^T W S \right)^{-1} H^T Y \) is edge preserving but not space-invariant.

Instead use

\[
\hat{X}_1 = \left( H^T H + \lambda_1 S^T S \right)^{-1} H^T Y \\
\hat{X}_2 = \left( H^T H + \lambda_2 S^T S \right)^{-1} H^T Y
\]

where \( \lambda_1 < \lambda_{\text{opt}} < \lambda_2 \).

Thus, \( \hat{X}_1 \) and \( \hat{X}_2 \) can be computed using 2D-FFT. The final result should be obtained using a diagonal weight matrix \( W \) with values in the range \([0,1]\) (1-edge, 0-smooth):

\[
\hat{X}_{\text{Final}} = W \hat{X}_1 + (I - W) \hat{X}_2
\]
Fast SSR (2) - Periodic-Step SD

A numerical method to speed-up convergence
Relation to Super-Resolution

\[
\begin{align*}
\{ Y_k &= D_k H_k F_k X + V_k, \quad V_k \sim \mathcal{N}\{0, W_k^{-1}\} \}_{k=1}^N \\
\begin{bmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_N
\end{bmatrix} &= \begin{bmatrix}
D_1 H_1 F_1 \\
D_2 H_2 F_2 \\
\vdots \\
D_N H_N F_N
\end{bmatrix} X + \begin{bmatrix}
V_1 \\
V_2 \\
\vdots \\
V_N
\end{bmatrix}
\end{align*}
\]
Basic Assumptions

- A sequence of measurements $y(k)$ is obtained sequentially.
- These measurements correspond linearly to an unknown vector $\mathbf{x}$ through $y(k) = C^T(k)\mathbf{x} + n(k)$

$$
\begin{bmatrix}
  y(1) \\
  y(2) \\
  y(3) \\
  \vdots \\
  y(L)
\end{bmatrix} =
\begin{bmatrix}
  \cdots & C^T(1) & \cdots \\
  \cdots & C^T(2) & \cdots \\
  \cdots & C^T(3) & \cdots \\
  \vdots & \vdots & \vdots \\
  \cdots & C^T(L) & \cdots \\
\end{bmatrix}
\begin{bmatrix}
  \mathbf{x} \\
  n(1) \\
  n(2) \\
  n(3) \\
  \vdots \\
  n(L)
\end{bmatrix}
$$
Basic Assumptions

- **Assumption 1** – we have enough measurements, i.e., if we write $y = Cx + n$, $C \in M^{[L \times N]}$, then $L \geq N$ and $C$ is full-rank.

  → If LS (ML) is applied, we get

  $f\{x\} = \|y - Cx\|_2^2 \Rightarrow \text{Min.} \quad \Rightarrow \quad \hat{x} = \left(C^T C\right)^{-1} C^T y$

- **Assumption 2** – $x$ is high dimensional [N elements] and thus the above solution is practically impossible

  **Turn to iterative methods**
Simple Iterative Method - SD

\[ f \{x\} = \|y - Cx\|_2^2 \Rightarrow \text{Min.} \quad \Rightarrow \quad \frac{\partial f \{x\}}{\partial x} = C^T (y - Cx) \]

Using the Steepest-Descend idea we get

\[ \hat{x}_{k+1} = \hat{x}_k - \mu C^T (y - C\hat{x}_k) = \]

\[ = \hat{x}_k - \mu \sum_{j=1}^{L} C(j)[y(j) - C^T(j)\hat{x}_k] \]

So we see that the gradient is built from \( L \) separate contributions, each obtained from a different measurement.
Decomposition of the Gradient

\[ \hat{x}_{k+1} = \hat{x}_k - \mu c^T \left( y - c \hat{x}_k \right) = \]

\[ = \hat{x}_k - \mu \sum_{j=1}^{L} \mathbb{C}(j) \left[ y(j) - c^T(j) \hat{x}_k \right] \]
Periodic-Step SD

Instead of using

\[ \hat{x}_{k+1} = \hat{x}_k - \mu \sum_{j=1}^{L} C(j)[y(j) - C^T(j)\hat{x}_k] \]

update the estimate of x for each SCALAR measurement

\[ \hat{x}_{k+j+\frac{j}{L}} = \hat{x}_{k+j+\frac{j}{L}} - \mu C(j)[y(j) - C^T(j)\hat{x}_{k+j+\frac{j}{L}}] \]

for \( k = 0, 1, 2, 3, \ldots \)

and for each k, sweep \( j = 1, 2, 3, \ldots, L \)
Related Work

This idea of breaking the gradient into several parts and updating the estimate after each of them is well-known, especially in cases where sequential measurements are obtained. Two such classic examples:

- Neural Network training (see Bertsekas’s book)
- Signal Processing (see LMS by Widrow et.al.)

In image restoration and super-resolution problems, we may consider updating our output image after every pixel in the measurements. The benefit is convergence speed-up.
Analysis Results

- Convergence is guaranteed if $0 < \mu < \min_{1 \leq j \leq L} \left\{ \frac{2}{C^T(j)C(j)} \right\}$

- The convergence is to the LS optimal solution only if
  - Infinitesimal step-size $\mu \to 0$,
  - Diminishing step-size $\mu_k \to 0$, or if
  - $C$ is square.

- In all other cases, the convergence is to a deviated solution.

- In the SSR case, we are not interested in exact solution !!!!

- Rate of convergence is dramatically improved (compared to SD, NSD, CG, Jacobi, GS, & SOR)
SSR - Simulation Results

SYNTHETIC CASE

25 images were created from one 100-by-100 pixels image using

• Motion - Affine,
• Blur – 3-by-3 uniform,
• Noise – Gaus. white $\sigma=3$.

These 25 images were fused to create a 200-by-200 pixels output.

This algorithm effectively converges after one iteration.