Super-Resolution Reconstruction of Images - Static and Dynamic Paradigms

February 2002

Michael Elad*

The Scientific Computing and Computational Mathematics Program - Stanford

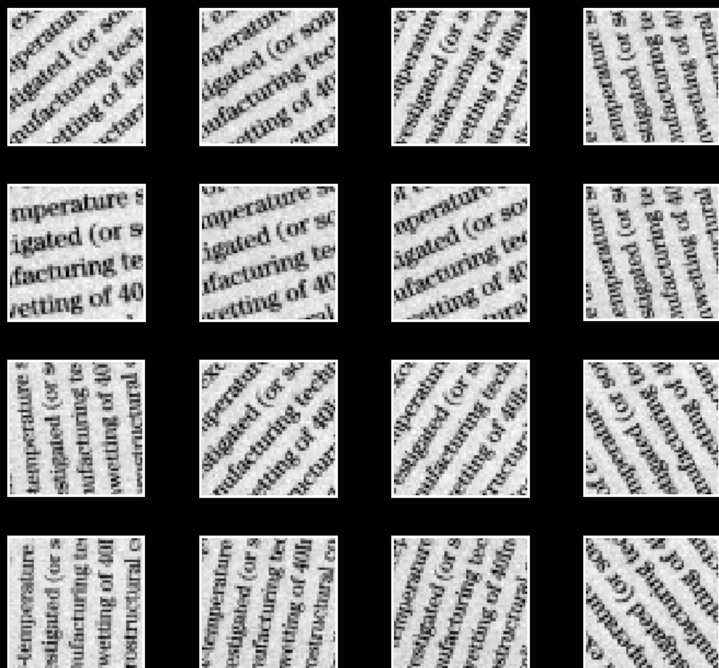
* Joint work with Prof. Arie Feuer – The Technion, Haifa Israel,
Prof. Yacob Hel-Or – IDC, Herzelia, Israel,
Tamir Sagi - Zapex-Israel.

Static Versus Dynamic Super-Resolution

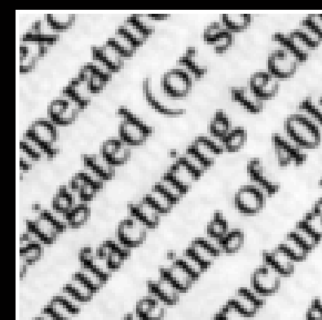
Definitions and Activity Map

Basic Super-Resolution Idea

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

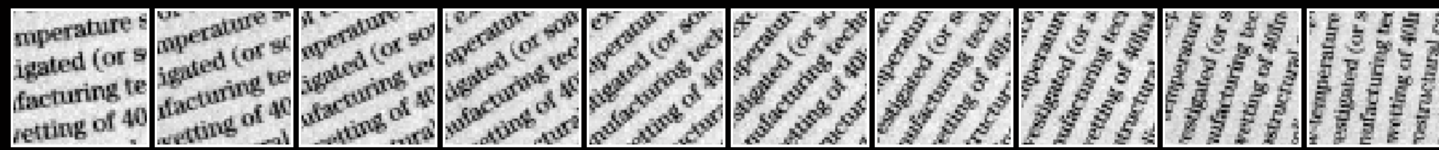


Required: Fusion of the measurements into a higher resolution image/s



Static Super-Resolution (SSR)

$\underline{Y}_1$ $\underline{Y}_2$ $\underline{Y}_3$ •••• $\underline{Y}_N$

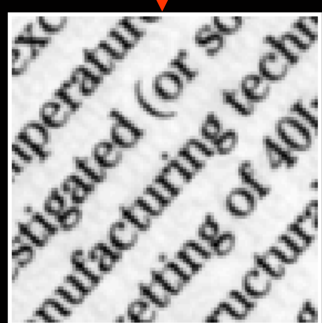


t

Low Resolution Measurements

Static Super-Resolution Algorithm

$$\boxed{\hat{\underline{X}} = f\left\{\underline{Y}_1, \underline{Y}_2, \underline{Y}_3, \ldots, \underline{Y}_N\right\}}$$

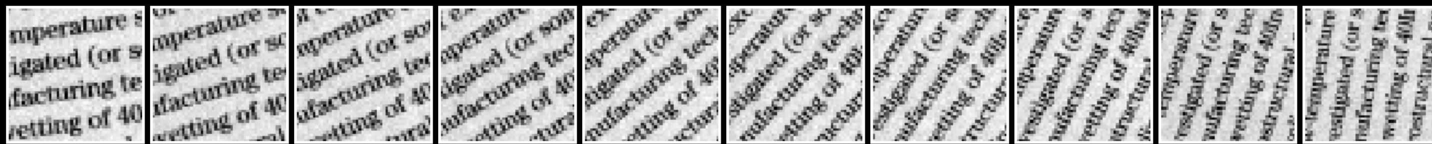


$\hat{\underline{X}}$

High Resolution Reconstructed Image

Dynamic Super-Resolution (DSR)

Low Resolution Measurements

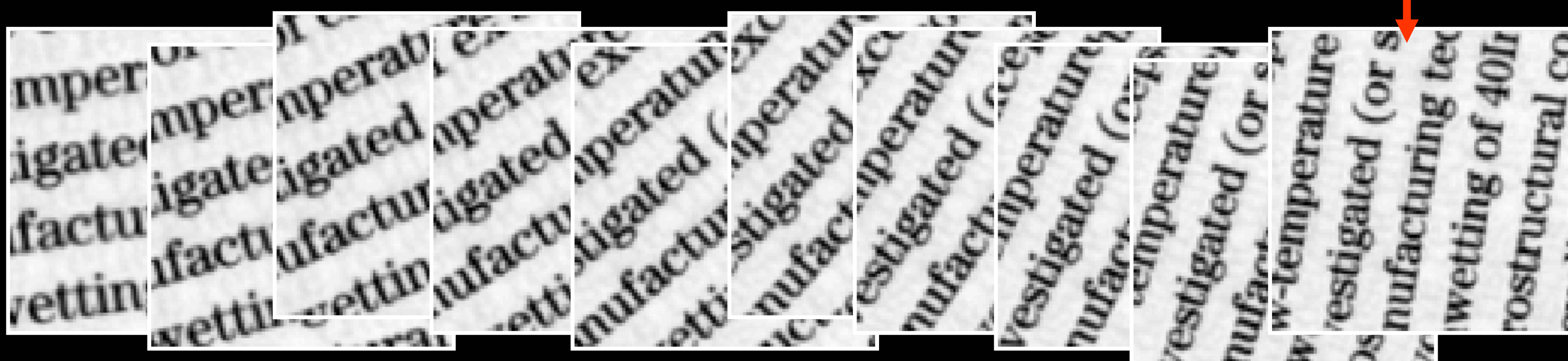


$\{\underline{Y}(t)\}_t$

$$\hat{\underline{X}}(t) = f\{\underline{Y}(t), \underline{Y}(t-1), \ \ldots\}$$

Dynamic Super-Resolution Algorithm

High Resolution Reconstructed Images



$\{\hat{\underline{X}}(t)\}_t$

Other Work In this Field

People	Place	Years
Peleg, Irani, Werman, Keren, Schweitzer	HUJI	1987-1994
Kim, Bose, Valenzuela	Penn. State	1990-1993
Patti, Tekalp, Zesan, Ozkan, Altunbasak	Rochester	1992-1998
Morris, Cheeseman, Smelyanskiy, Maluf	NASA-AMES	1992-2002
Ur & Gross	TAUI	1992-1993
Elad, Feuer, Sagi, Hel-Or	Technion	1995-2001
Schutlz, Stevenson, Borman	Notre-Dame	1995-1999
Shekarforush, Berthod, Zerubia, Werman	INRIA-France	1995-1999
Katsaggelos, Tom, Galatsanos	Northwestern	1995-1999
Shah, Zachor	Berkeley	1996-1999
Nguyen, Milanfar, Golub	Stanford	1998-2001
Baker, Kanade	CMU	1999-2001

* 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

- M. Elad and A. Feuer, “Restoration of Single Super-Resolution Image From Several Blurred, Noisy and Down-Sampled Measured Images”, the IEEE Trans. on Image Processing, Vol. 6, no. 12, pp. 1646-58, December 1997.
- M. Elad and A. Feuer, “Super-Resolution Restoration of Continuous Image Sequence - Adaptive Filtering Approach”, the IEEE Trans. on Image Processing, Vol. 8. no. 3, pp. 387-395, March 1999.
- M. Elad and A. Feuer, “Super-Resolution reconstruction of Continuous Image Sequence”, the IEEE Trans. On Pattern Analysis and Machine Intelligence (PAMI), Vol. 21, no. 9, pp. 817-834, September 1999.
- M. Elad and Y. Hel-Or, “A Fast Super-Resolution Reconstruction Algorithm for Pure Translational Motion and Common Space Invariant Blur”, Accepted to the IEEE Trans. on Image Processing, March 2001.
- T. Sagi, A. Feuer and M. Elad, “The Periodic Step Gradient Descent Algorithm - General Analysis and Application to the Super-Resolution Reconstruction Problem”, EUSIPCO 1998.

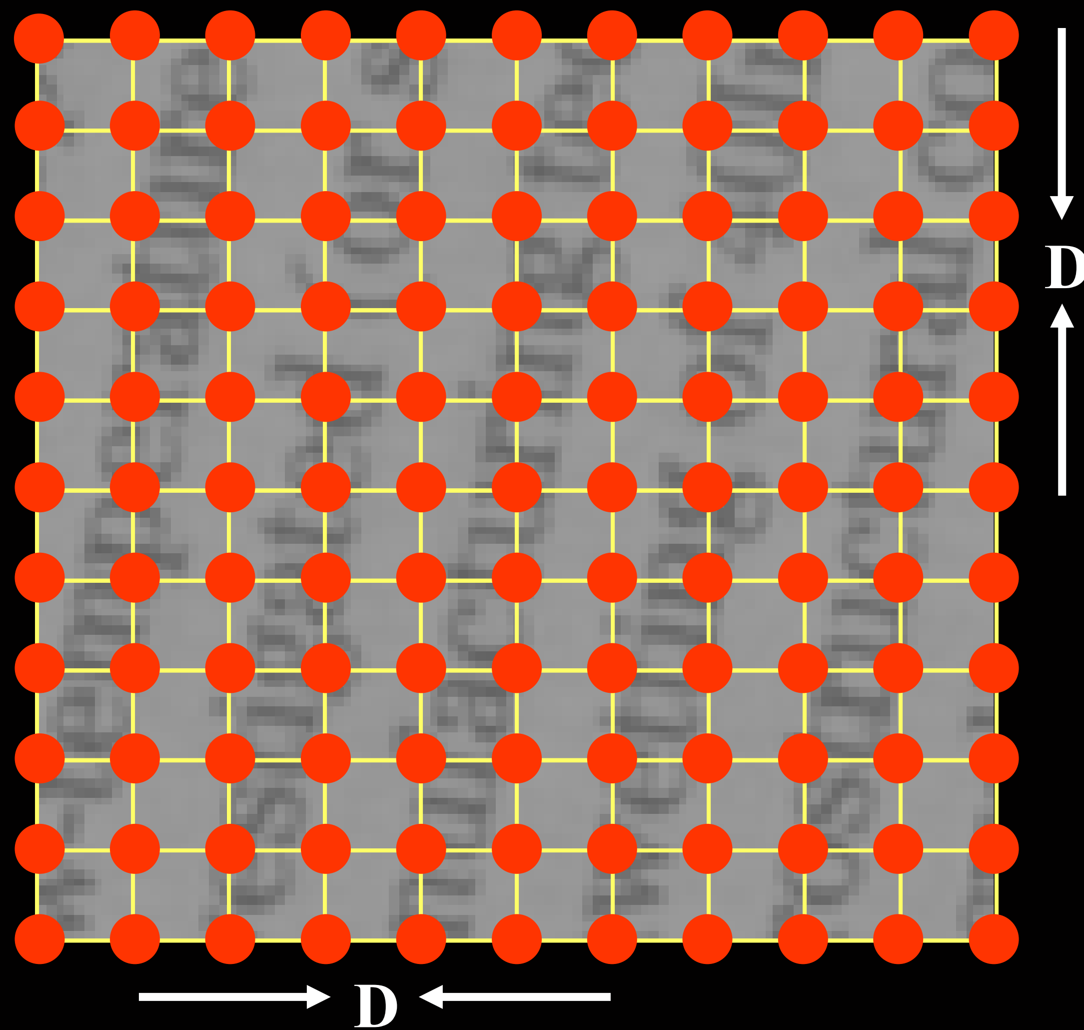
All found in http://sccm.stanford.edu/~elad

Super-Resolution Basics

Intuition and Relation to Sampling theorems

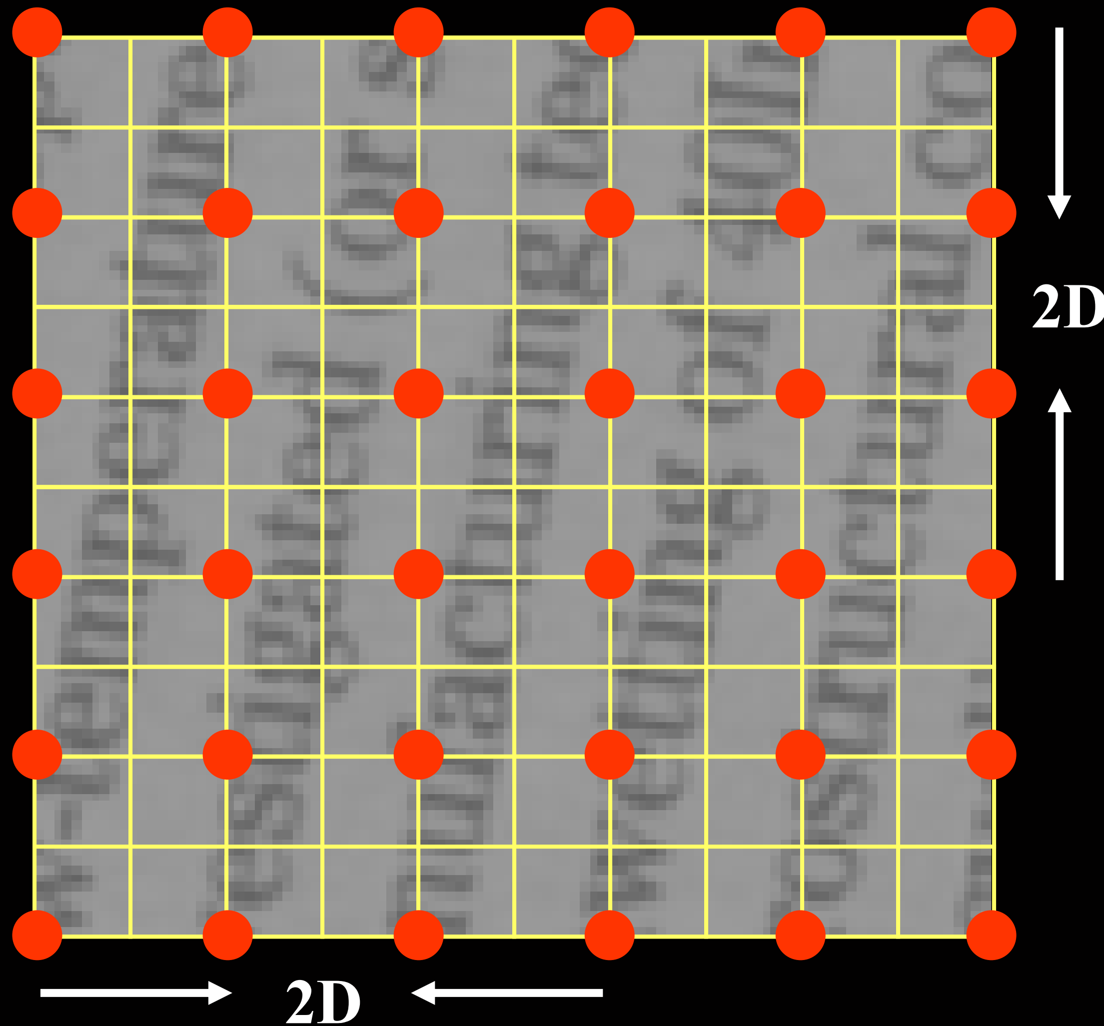
Simple Example

For a given band-limited image, the Nyquist sampling theorem states that if a uniform sampling is fine enough (≥**D**), perfect reconstruction is possible.



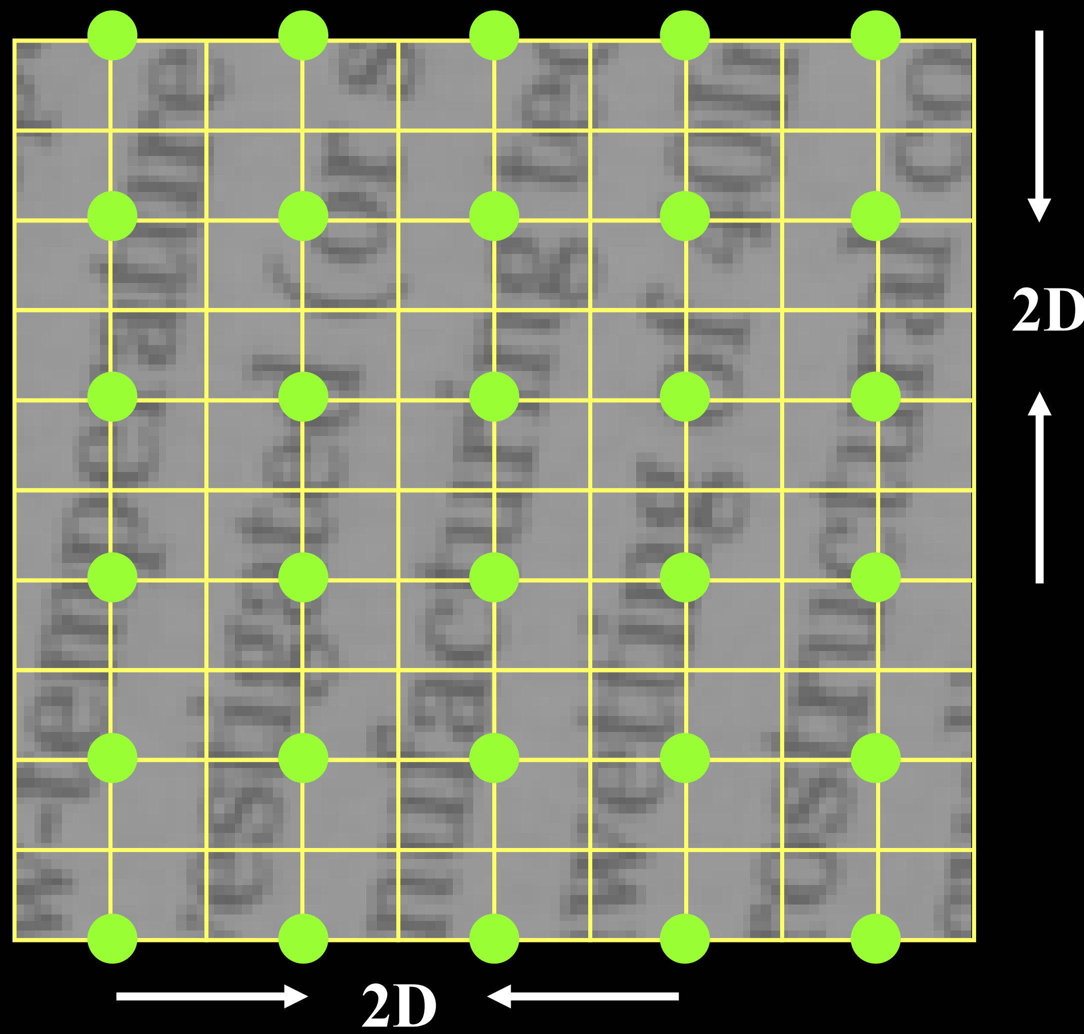
Simple Example

Due to our limited camera resolution, we sample using an insufficient 2D grid



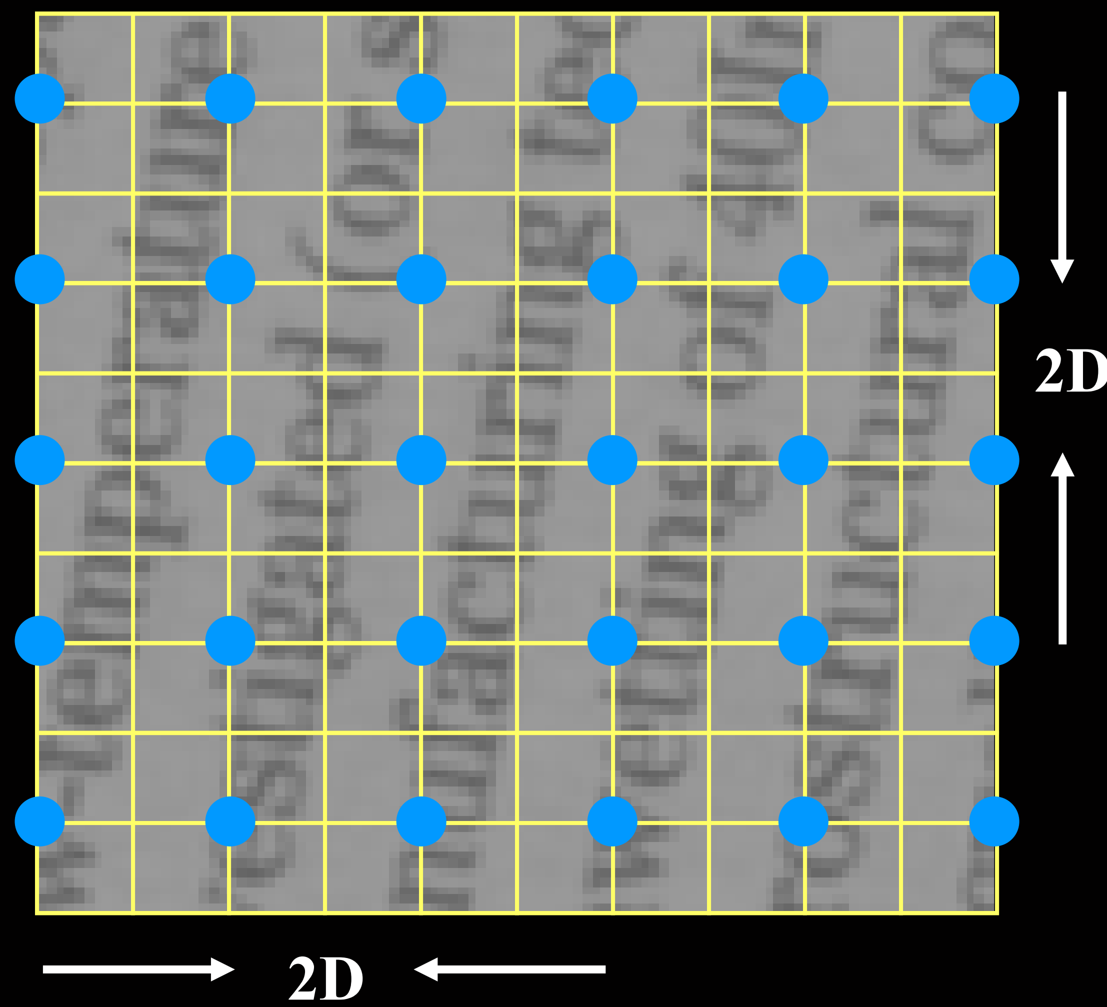
Simple Example

However, we are allowed to take a second picture and so, shifting the camera 'slightly to the right' we obtain



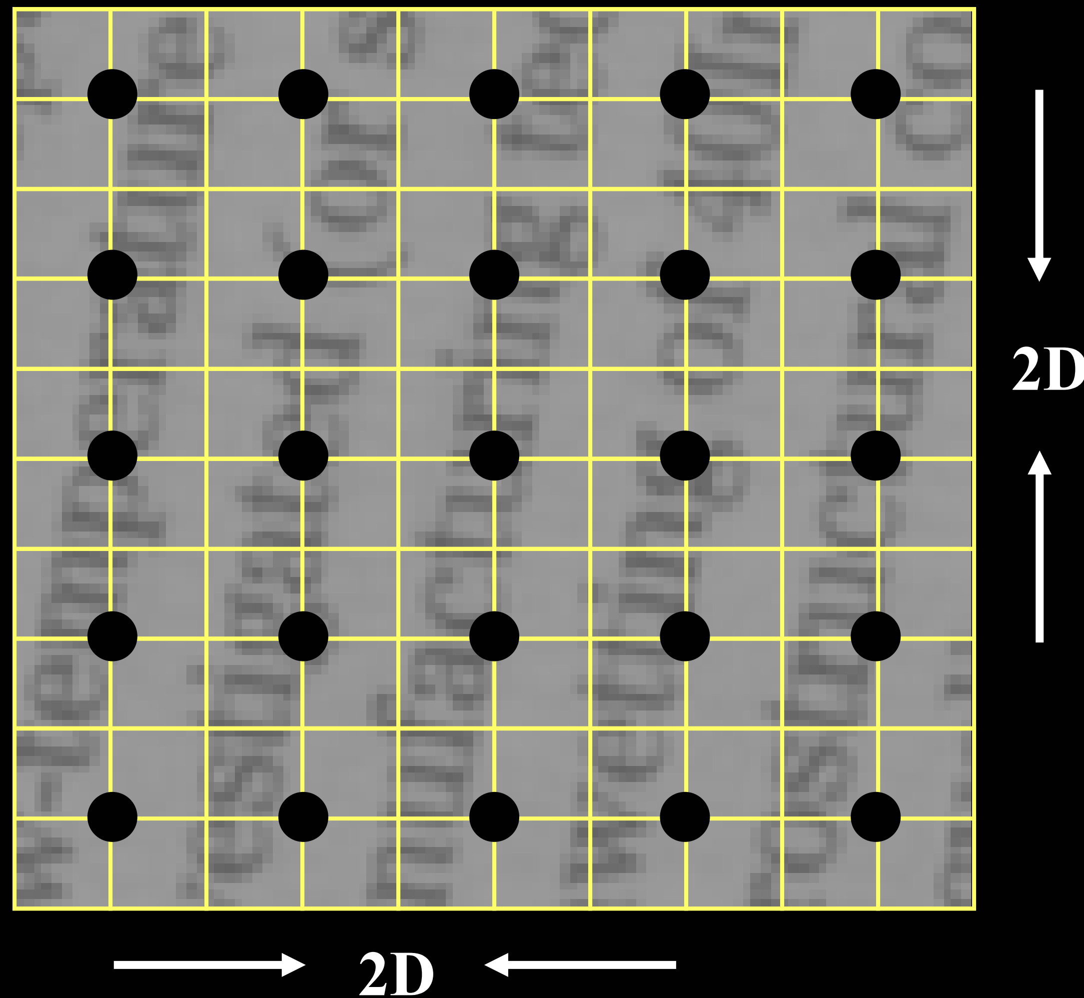
Simple Example

Similarly, by shifting down we get a third image



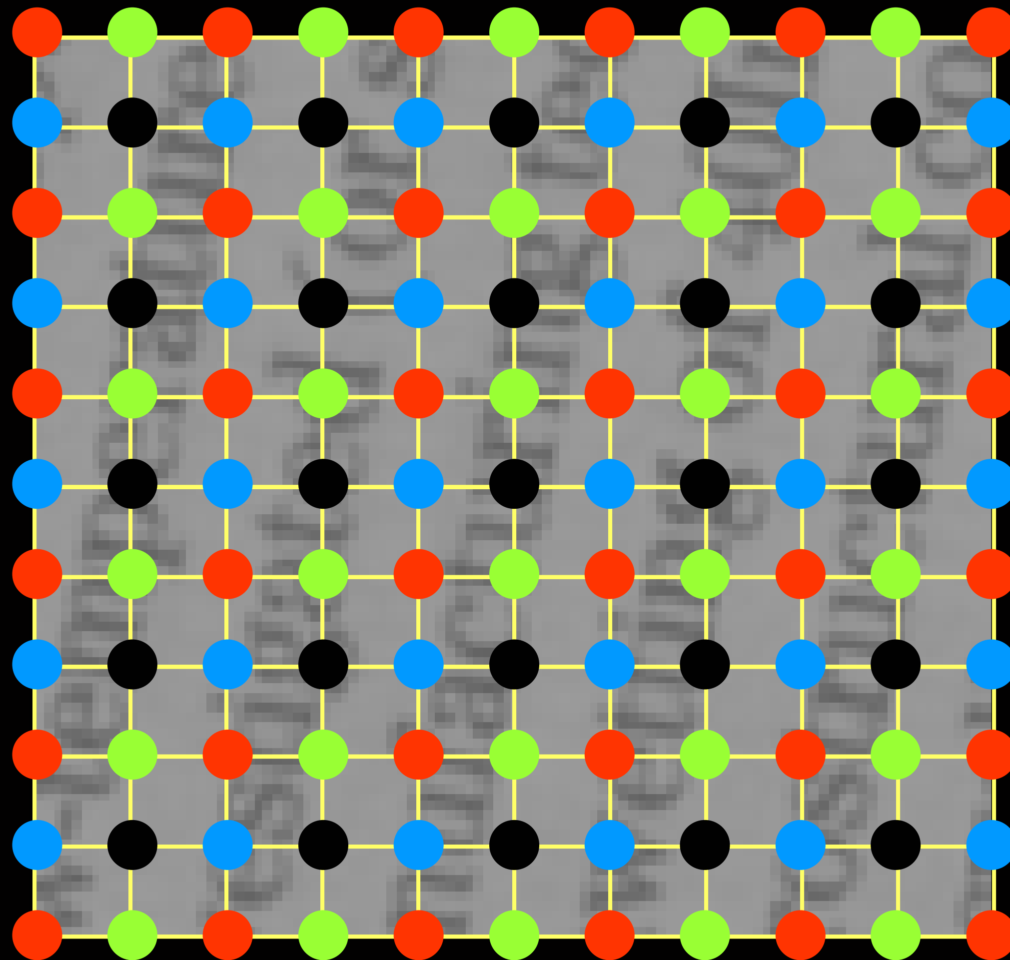
Simple Example

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.

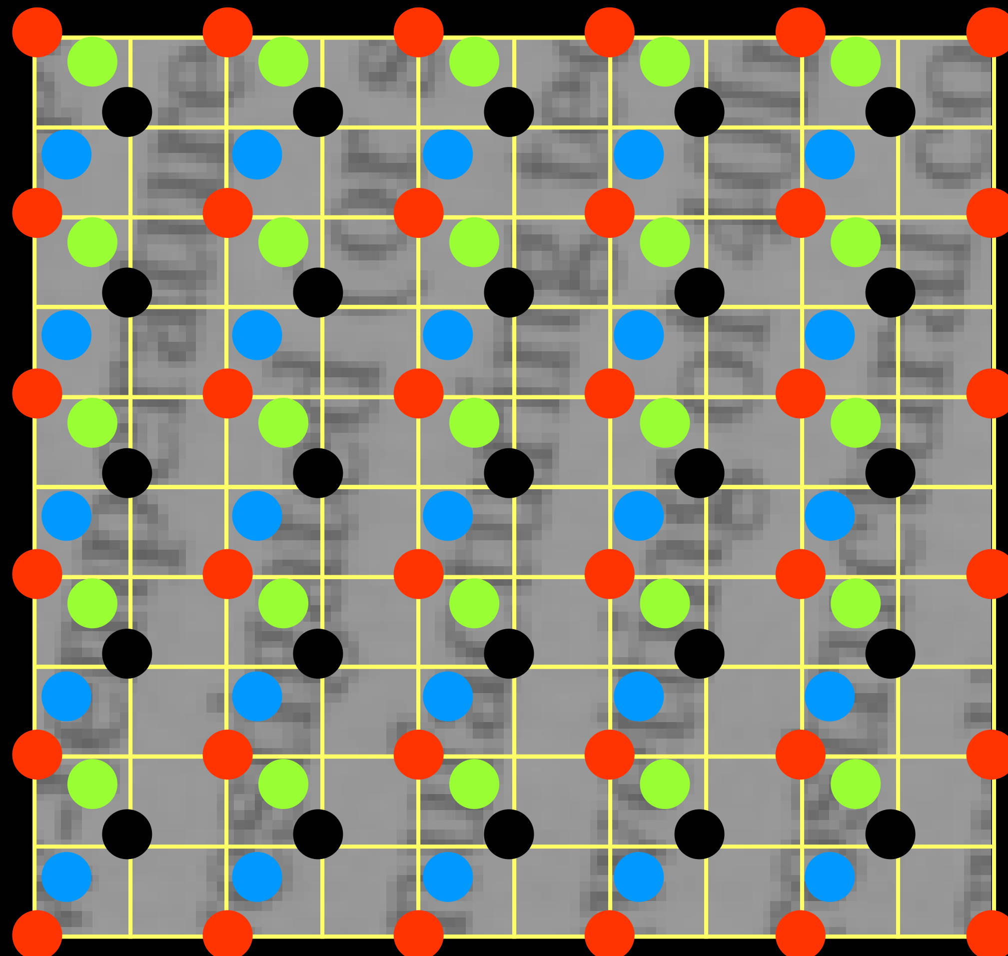


This is Super-Resolution in its simplest form

Uncontrolled Displacements

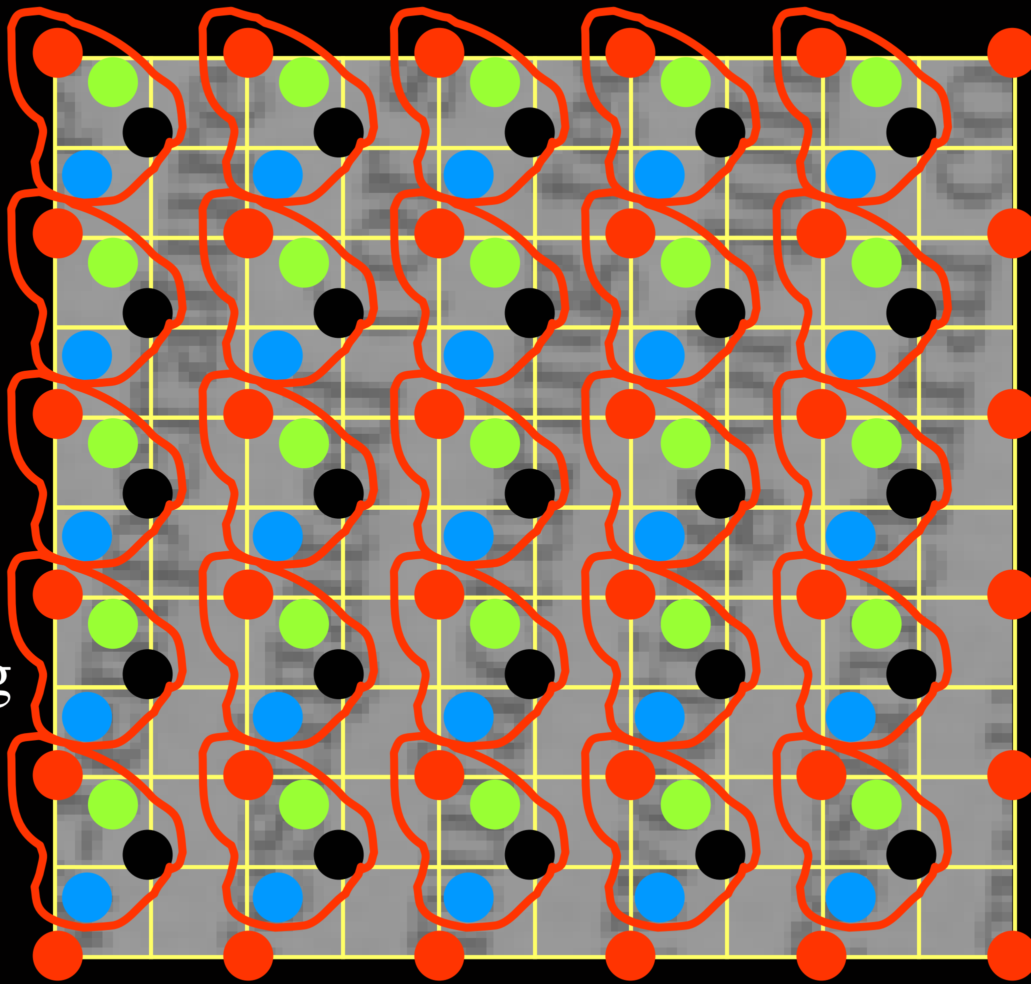
In the previous example we counted on exact movement of the camera by D in each direction.

What if the camera displacement is uncontrolled?



Uncontrolled Displacements

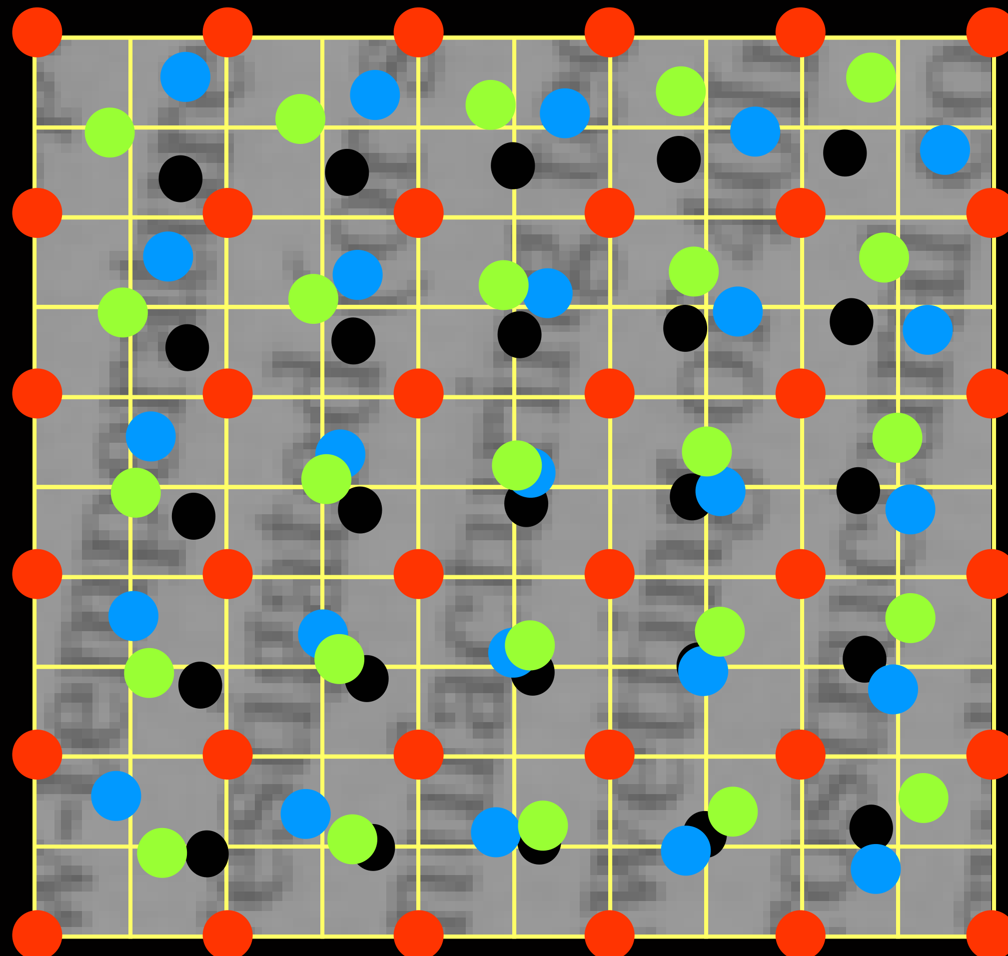
It turns out that there is a sampling theorem due to Yen (1956) and Papulis (1977) covering this case, guaranteeing perfect reconstruction for periodic uniform sampling if the sampling density is high enough (1 sample per each D-by-D square).



Uncontrolled Rotation/Scale/Disp.

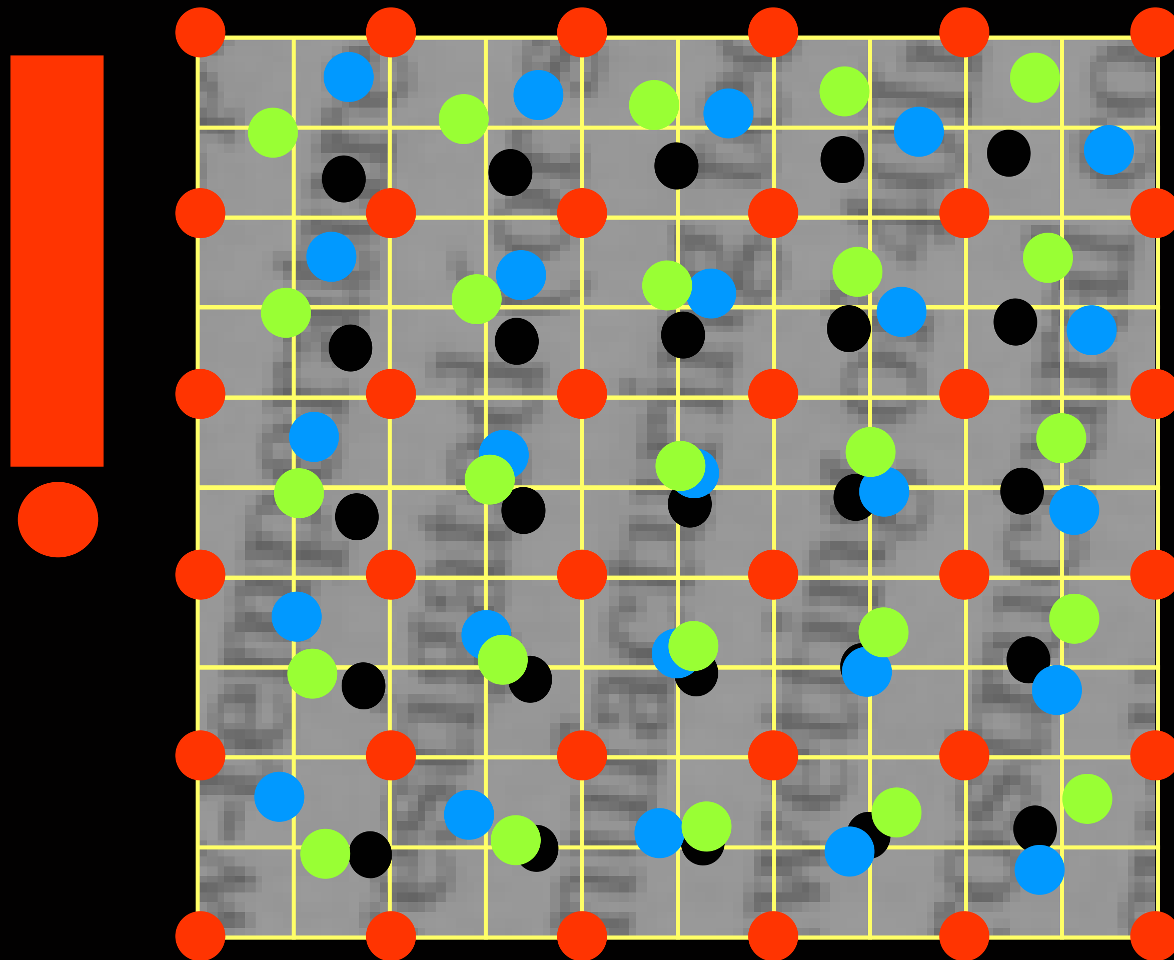
In the previous examples we restricted the camera to move horizontally/vertically parallel to the photograph object.

What if the camera rotates? Gets closer to the object (zoom)?



Uncontrolled Rotation/Scale/Disp.

There is no sampling theorem covering this case

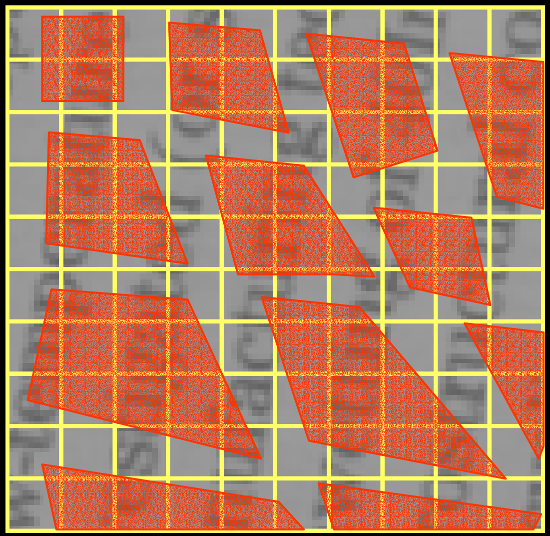


Further Complications

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

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

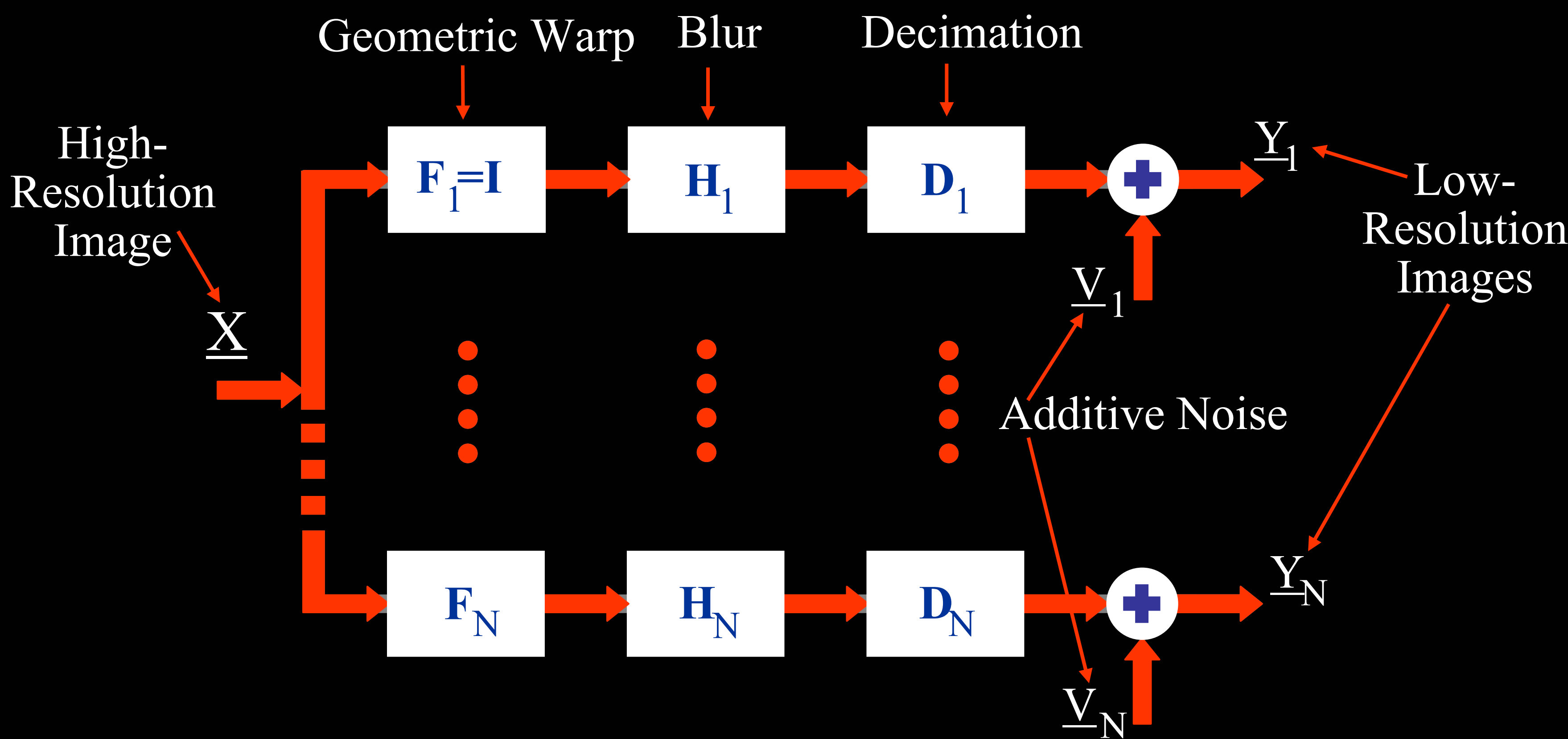
3. Samples may be noisy – any reconstruction process must take that into account.



Static Super-Resolution

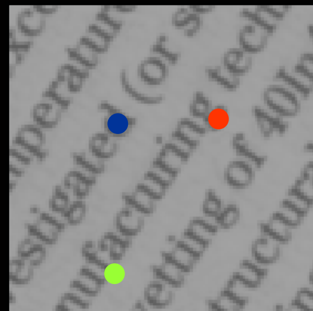
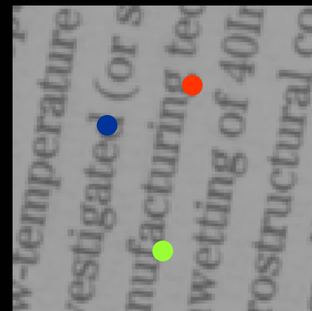
The creation of a single improved image, from the finite measured sequence of images

SSR - The Model



$$\left\{ \underline{Y}_k = \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \underline{X} + \underline{V}_k, \quad \underline{V}_k \sim \mathbf{N}\left\{0, \mathbf{W}_k^{-1}\right\} \right\}_{k=1}^{N}$$

The Warp As a Linear Operation

X   Z

Per every point in X find a matching point in Z

$$\begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & & & \mathbf{0} \\ & & 1 & \\ & \mathbf{0} & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_j \\ \vdots \\ z_N \end{bmatrix}$$

F[j,i]=1

Model Assumptions

We assume that the images $\underline{Y}_k$ and the operators $\mathbf{H}_k$, $\mathbf{D}_k$, $\mathbf{F}_k$,& $\mathbf{W}_k$ are known to us, and we use them for the recovery of $\underline{X}$.

$\underline{Y}_k$ – The measured images (noisy, blurry, down-sampled ..)

$\mathbf{H}_k$ – The blur can be extracted from the camera characteristics

$\mathbf{D}_k$ – The decimation is dictated by the required resolution ratio

$\mathbf{F}_k$ – The warp can be estimated using motion estimation

$\mathbf{W}_k$ – The noise covariance can be extracted from the camera characteristics

The Model as One Equation

$$\left\{ \underline{Y}_k = \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \underline{X} + \underline{V}_k, \quad \underline{V}_k \sim \mathbf{N}\left\{0, \mathbf{W}_k^{-1}\right\} \right\}_{k=1}^{N}$$

$$\begin{bmatrix} \underline{Y}_1 \\ \underline{Y}_2 \\ \vdots \\ \underline{Y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{D}_1\mathbf{H}_1\mathbf{F}_1 \\ \mathbf{D}_2\mathbf{H}_2\mathbf{F}_2 \\ \vdots \\ \mathbf{D}_N\mathbf{H}_N\mathbf{F}_N \end{bmatrix} \underline{X} + \begin{bmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \vdots \\ \underline{V}_N \end{bmatrix} \qquad \begin{bmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \vdots \\ \underline{V}_N \end{bmatrix} \sim \mathbf{N}\left\{ 0, \begin{bmatrix} \mathbf{W}_1 & & & 0 \\ & \mathbf{W}_2 & & \\ & & \ddots & \\ 0 & & & \mathbf{W}_N \end{bmatrix}^{-1} \right\}$$

A Thumb Rule on Desired Resolution

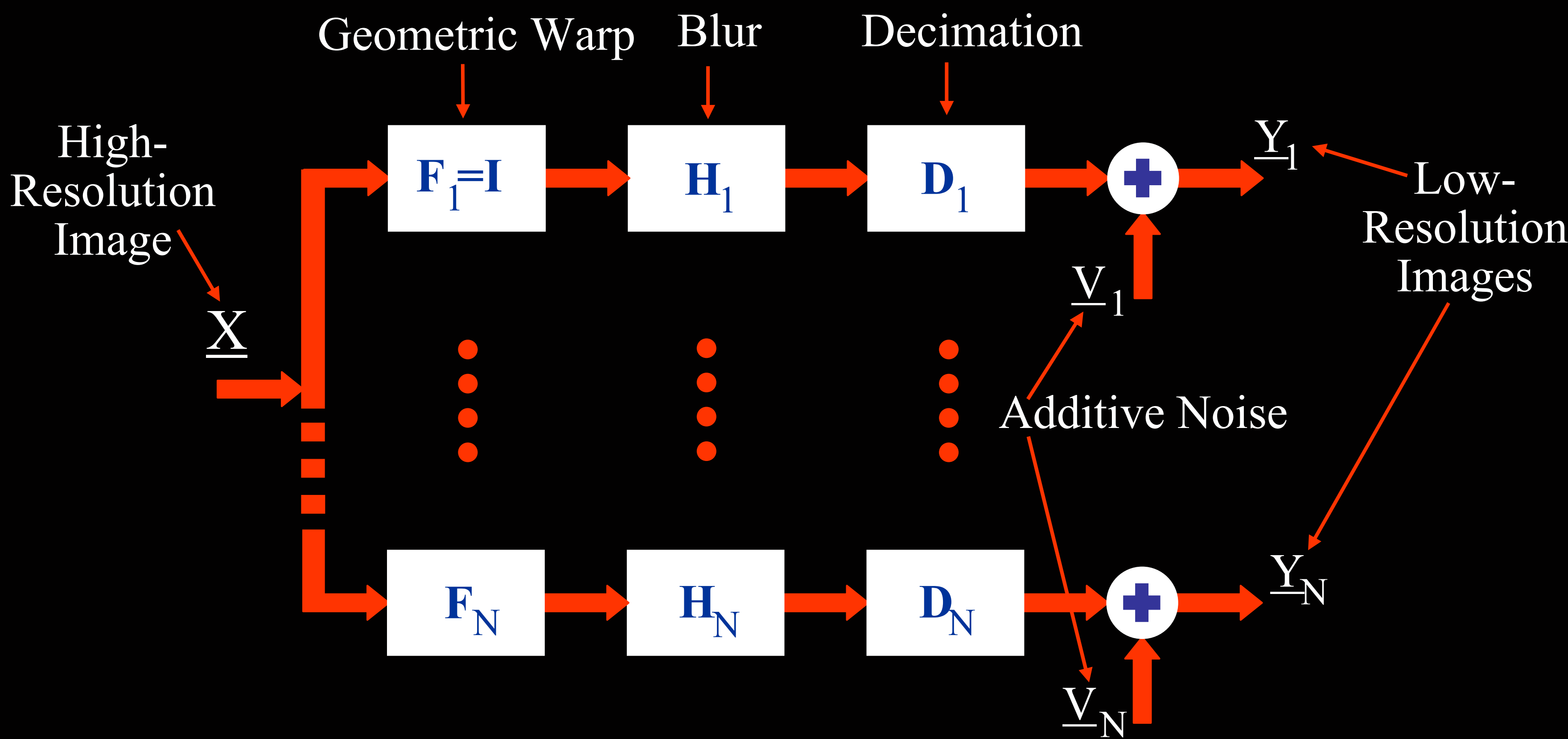
In the noiseless case we have

$$\begin{bmatrix} \underline{Y}_1 \\ \underline{Y}_2 \\ \vdots \\ \underline{Y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{D}_1\mathbf{H}_1\mathbf{F}_1 \\ \mathbf{D}_2\mathbf{H}_2\mathbf{F}_2 \\ \vdots \\ \mathbf{D}_N\mathbf{H}_N\mathbf{F}_N \end{bmatrix} \underline{X}$$

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 – $L \le \sqrt{N} \cdot M$

The Maximum-Likelihood Approach



Which $\underline{X}$ would be such that when fed to the above system it yields a set $\underline{Y}_k$ closest to the measured images ?

SSR - ML Reconstruction (LS)

Minimize: $$\varepsilon_{ML}^2(\underline{X}) = \sum_{k=1}^{N} \left\| \underline{Y}_k - \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \underline{X} \right\|_{\mathbf{W}_k}^2$$

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

$$\left\{ \begin{array}{c} \mathbf{R} = \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \\ \underline{P} = \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k \underline{Y}_k \end{array} \right\} \Rightarrow \mathbf{R}\hat{\underline{X}} = \underline{P}$$

SSR - MAP Reconstruction

Add a term which penalizes for
the solution image quality

$$\varepsilon^2_{MAP}(\underline{X}) = \sum_{k=1}^{N} \left\| \underline{Y}_k - \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \underline{X} \right\|^2_{\mathbf{W}_k} + \lambda A\{\underline{X}\}$$

Possible Prior functions - Examples:

1. $A\{\underline{X}\} = \underline{X}^T \mathbf{S}^T \mathbf{W}(\underline{X}_0) \mathbf{S} \underline{X}$ - simple spatially adaptive,
2. $A\{\underline{X}\} = \rho\{\mathbf{S}\underline{X}\}$ - M estimator (robust functions),

Note: Convex prior guarantees convex programming problem

Iterative Reconstruction

Assuming the prior $A\{\underline{X}\} = \underline{X}^T \mathbf{S}^T \mathbf{W} \mathbf{S} \underline{X}$ is used

$$\left\{ \begin{array}{c} \mathbf{R} = \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k + \lambda \mathbf{S}^T \mathbf{W} \mathbf{S} \\ \underline{P} = \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k \underline{Y}_k \end{array} \right\} \Rightarrow \mathbf{R}\hat{\underline{X}} = \underline{P}$$

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

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

$$\hat{\underline{X}}_{j+1} = \hat{\underline{X}}_j - \mu \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k \left[\underline{Y}_k - \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \hat{\underline{X}}_j \right] - \mu \lambda \mathbf{S}^T \mathbf{W} \mathbf{S} \hat{\underline{X}}_j$$

Image-Based Processing

SD* Iteration: $$\hat{\underline{X}}_{j+1} = \hat{\underline{X}}_j - \mu \sum_{k=1}^{N} \underbrace{\mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k}_{\text{Back projection}} \underbrace{\left[\underline{Y}_k - \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \hat{\underline{X}}_j\right]}_{\text{Simulated error}} \underbrace{- \mu\lambda \mathbf{S}^T \mathbf{W} \mathbf{S} \hat{\underline{X}}_j}_{\text{Weighted edges}}$$

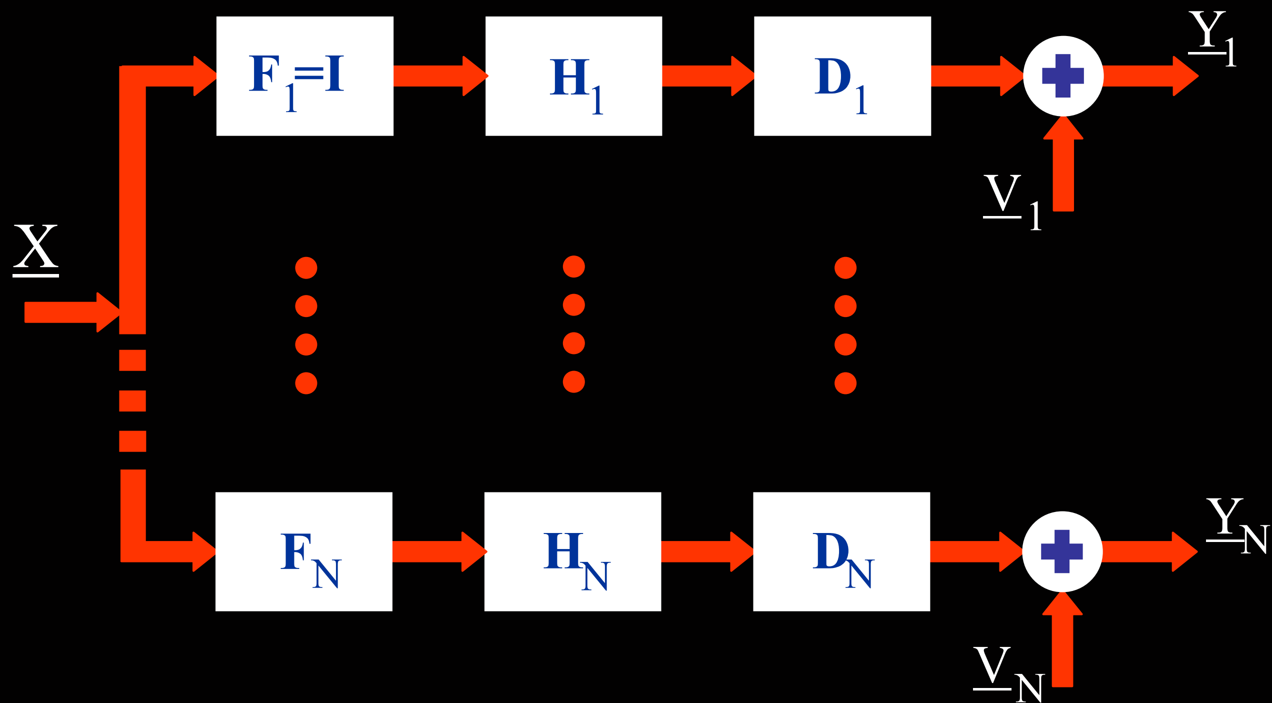
All the above operations can be interpreted as operations performed on images.

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

* Also true for the Conjugate Gradient algorithm

SSR – Simpler Problems



$$\hat{\underline{X}} = \left[\sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k D_k H_k F_k + \lambda S^T W S\right]^{-1} \sum_{k=1}^{N} F_k^T H_k^T D_k^T W_k \underline{Y}_k$$

SSR – Simpler Problems

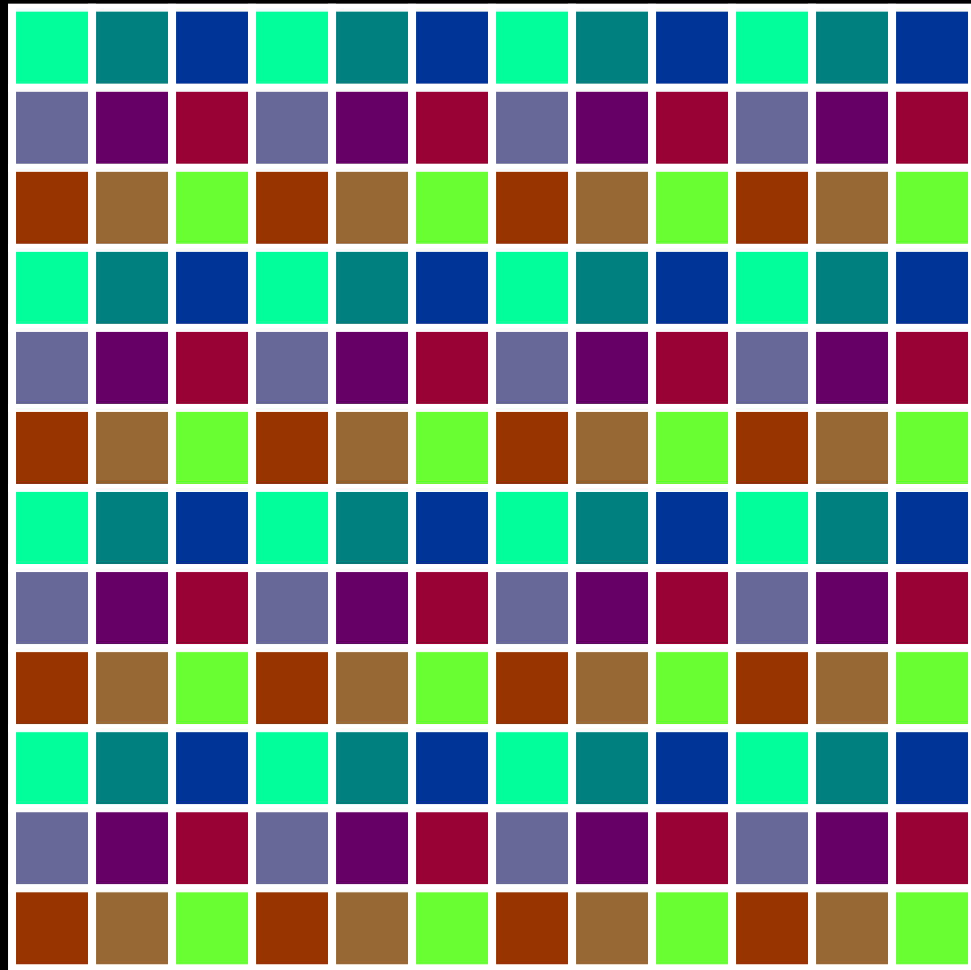
Single image de-noising $\left\{\underline{Y}=\underline{X}+\underline{V}\right\}$	$\hat{\underline{X}}=\left[\mathbf{I}+\lambda\mathbf{S}^T\mathbf{W}\mathbf{S}\right]^{-1}\underline{Y}$
Single image restoration $\left\{\underline{Y}=\mathbf{H}\underline{X}+\underline{V}\right\}$	$\hat{\underline{X}}=\left[\mathbf{H}^T\mathbf{H}+\lambda\mathbf{S}^T\mathbf{W}\mathbf{S}\right]^{-1}\mathbf{H}^T\underline{Y}$
Single image scaling $\left\{\underline{Y}=\mathbf{D}\underline{X}+\underline{V}\right\}$	$\hat{\underline{X}}=\left[\mathbf{D}^T\mathbf{D}+\lambda\mathbf{S}^T\mathbf{W}\mathbf{S}\right]^{-1}\mathbf{D}^T\underline{Y}$
Motion compensation average $\left\{\underline{Y}_k=\mathbf{F}_k\underline{X}+\underline{V}_k\right\}_{k=1}^N$	$\hat{\underline{X}}=\left[\sum_{k=1}^N\mathbf{F}_k^T\mathbf{F}_k+\lambda\mathbf{S}^T\mathbf{W}\mathbf{S}\right]^{-1}\sum_{k=1}^N\mathbf{F}_k^T\underline{Y}_k$

Using $A\left\{\underline{X}\right\}=\underline{X}^T\mathbf{S}^T\mathbf{W}\mathbf{S}\underline{X}$

Example 1

Synthetic case:

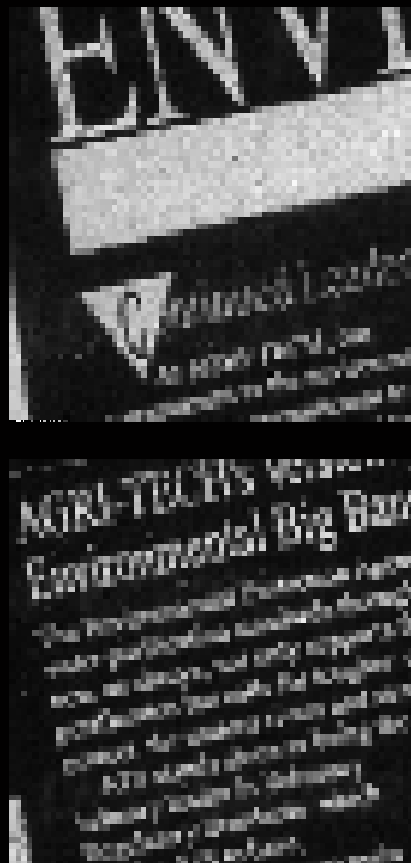
From a single image create 9 3:1 images this way



Example 1

Synthetic case:

9 images, no blur, 1:3 ratio



One of the low-resolution images

The higher resolution original

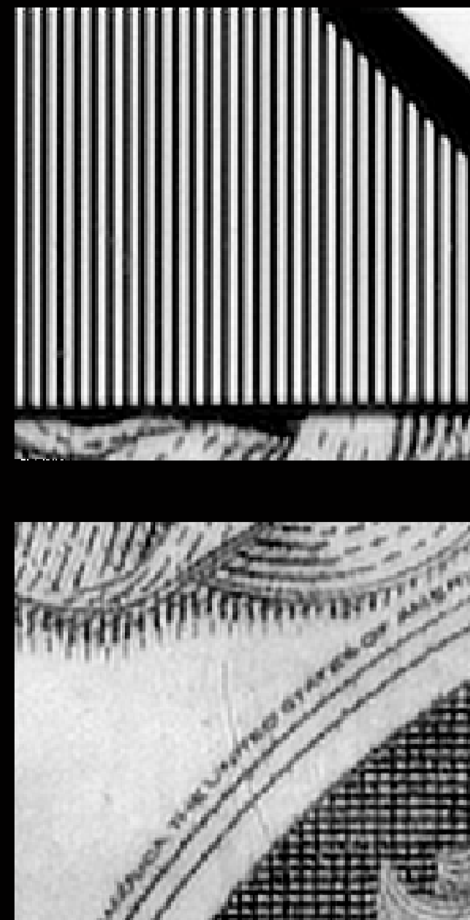
The reconstructed result

Example 2

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

Taken from one of the given images

Taken from the reconstructed result

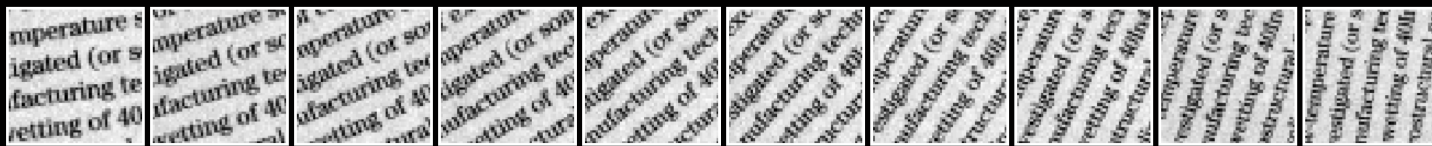


Dynamic Super-Resolution

Low Quality Movie In – High Quality Movie Out

Dynamic Super-Resolution (DSR)

Low Resolution Measurements



$\{\underline{Y}(t)\}_t$

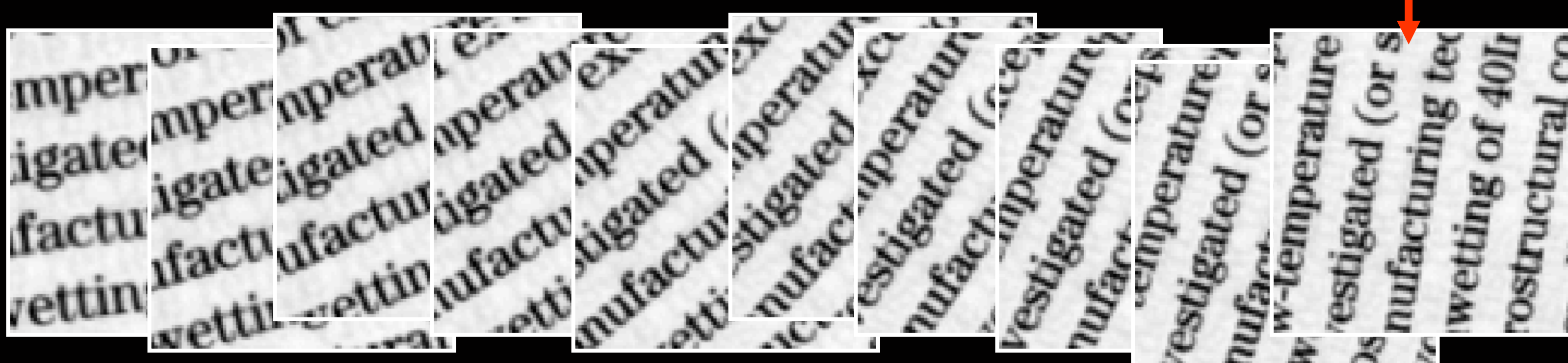
t

$$\hat{\underline{X}}(t) = f\{\underline{Y}(t), \underline{Y}(t-1), \ \ldots\}$$

Dynamic Super-Resolution Algorithm

High Resolution Reconstructed Images

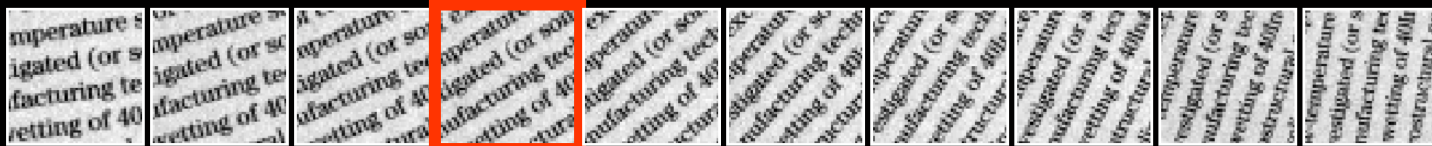
t



$\{\hat{\underline{X}}(t)\}_t$

Modeling the Problem

Low Resolution Measurements



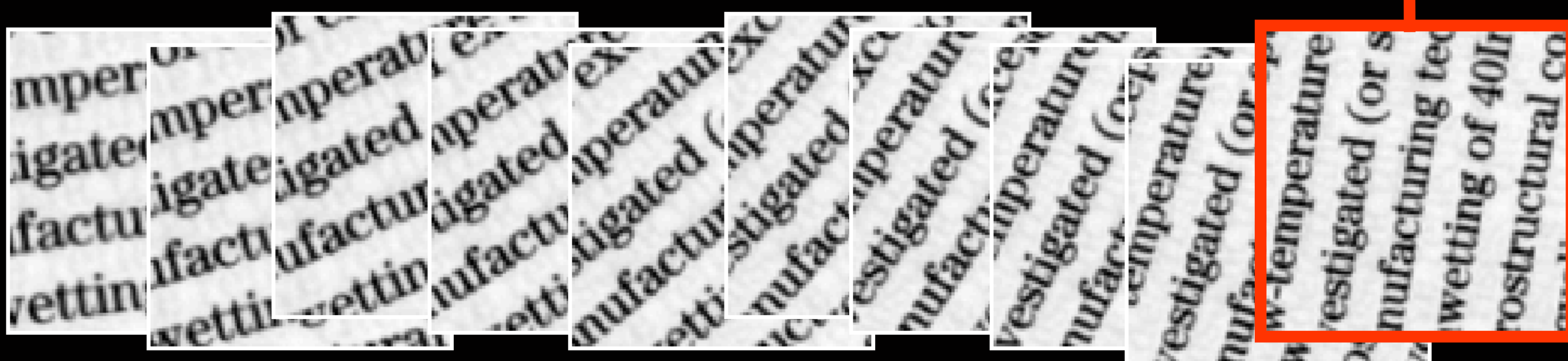
$\left\{\underline{Y}(t)\right\}_t$

t

$$\underline{Y}(t-k) = \mathbf{M}(t,k)\underline{X}(t) + \underline{N}(t,k)$$?

High Resolution Reconstructed Images

t



$\left\{\hat{\underline{X}}(t)\right\}_t$

Bypass

DSR – Proposed Model

$$\underline{Y}(t-k) = \mathbf{M}(t,k)\,\underline{X}(t) + \underline{N}(t,k)$$

$$\left\{ \begin{array}{c} \underline{Y}(t-k) = \mathbf{DH}\tilde{\mathbf{F}}(t,k)\,\underline{X}(t) + \underline{N}(t,k) \\ \\ N(t,k) \sim \mathbf{N}\left\{\underline{0}, \lambda^{-k}\mathbf{W}^{-1}\right\}, \text{ where } 0 < \lambda < 1 \\ \\ \text{and } \tilde{\mathbf{F}}(t,k) = \mathbf{F}(t-k+1)\cdots\mathbf{F}(t-1)\mathbf{F}(t) \end{array} \right\}_{k=0}^{t-1}$$

DSR – From Model to ML

- The DSR problem is referred to as a long sequence of SSR problems.
- Thus, Our model is
$$\left\{\begin{array}{l}\underline{Y}(t-k)=\mathbf{DH\tilde{F}}(t,k)\underline{X}(t)+\underline{N}(t,k)\\ N(t,k)\sim \mathbf{N}\left\{\underline{0},\lambda^{-k}\mathbf{W}^{-1}\right\},\text{ where } 0<\lambda<1\\ \text{and } \mathbf{\tilde{F}}(t,k)=\mathbf{F}(t-k+1)\cdots\mathbf{F}(t-1)\mathbf{F}(t)\end{array}\right\}_{k=0}^{t-1}$$
- Using ML approach

$$\varepsilon^2\left(\underline{X}(t),t\right)=\sum_{k=0}^{t-1}\lambda^k\left\|\underline{Y}(t-k)-\mathbf{DH\tilde{F}}(t,k)\underline{X}(t)\right\|_{\mathbf{W}}^2$$

and this function should be minimized per each t.

Solving the ML

Minimizing $\varepsilon^2\left(\underline{X}(t),t\right)=\sum_{k=0}^{t-1}\lambda^k\left\|\underline{Y}(t-k)-\mathbf{D}\mathbf{H}\tilde{\mathbf{F}}(t,k)\underline{X}(t)\right\|_{\mathbf{W}}^2$

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

where

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

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

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

Recursive Representation

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

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

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

$$\mathbf{L}(t) = \lambda \mathbf{F}^T(t)\mathbf{L}(t-1)\mathbf{F}(t) + \mathbf{H}^T\mathbf{WH}$$

$$\underline{Z}(t) = \lambda \mathbf{F}^T(t)\underline{Z}(t-1) + \mathbf{H}^T\mathbf{W}\underline{Y}(t)$$

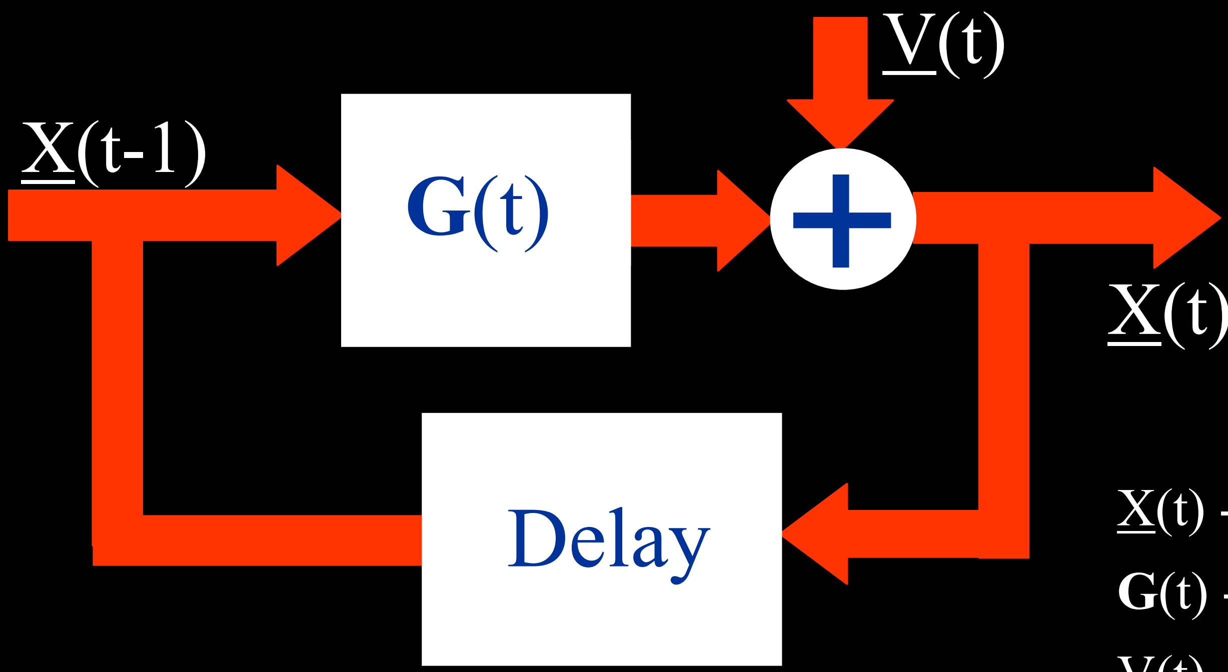
Alternative Approach

- 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.

Bypass

DSR - The Model (1)

The System's Equation



$\underline{X}(t)$ - High-resolution image

$\mathbf{G}(t)$ - Warp operation

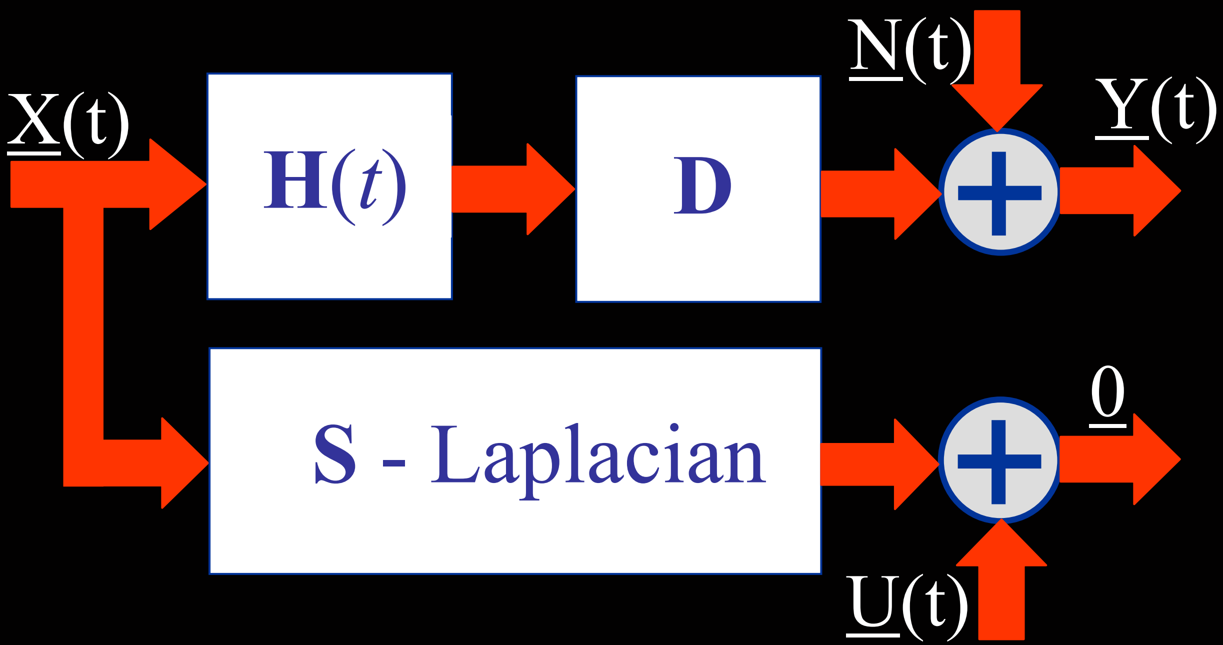
$\underline{V}(t)$ - Sequence innovation

assumed $\sim \mathbf{N}\{0, \mathbf{Q}^{-1}(t)\}$

$$\boxed{\underline{X}(t) = \mathbf{G}(t)\underline{X}(t-1) + \underline{V}(t)}$$

DSR - The Model (2)

The Measurements Equation



$$\begin{bmatrix} \underline{Y}(t) \\ \underline{0} \end{bmatrix} = \begin{bmatrix} \mathbf{DH}(t) \\ \mathbf{S} \end{bmatrix} \underline{X}(t) + \begin{bmatrix} \underline{N}(t) \\ \underline{U}(t) \end{bmatrix}$$

$\underline{Y}(t)$ - Measured image

$\mathbf{H}(t)$ - Blur

$\mathbf{D}$ - Decimation

$\underline{N}(t)$ - additive noise $\sim \mathbf{N}\left\{0, \mathbf{W}^{-1}(t)\right\}$

$\mathbf{S}$ - Laplacian

$\underline{U}(t)$ - Non-smooth. $\sim \mathbf{N}\left\{0, \mathbf{R}^{-1}(t)\right\}$

DSR - The Model (3)

$$\underline{X}(t) = \mathbf{G}(t)\underline{X}(t-1) + \underline{V}(t)$$

&

$$\begin{bmatrix} \underline{\mathbf{Y}}(t) \\ \underline{0} \end{bmatrix} = \begin{bmatrix} \mathbf{DH}(t) \\ \mathbf{S} \end{bmatrix} \underline{X}(t) + \begin{bmatrix} \underline{\mathbf{N}}(t) \\ \underline{U}(t) \end{bmatrix}$$

These two equations form a

Spatio-Temporal Prior

forcing spatial smoothness & temporal motion compensated smoothness

DSR - Reconstruction By KF

The model is given in a State-Space form

$$\underline{X}(t) = \mathbf{G}(t)\underline{X}(t-1) + \underline{V}(t)$$

$$\underline{Y}_A(t) = \mathbf{H}_A(t)\underline{X}(t) + \underline{N}_A(t)$$

where $\underline{V}(t) \sim \mathbf{N}\{\underline{0}, \mathbf{Q}^{-1}(t)\}$

$\underline{N}_A(t) \sim \mathbf{N}\{\underline{0}, \mathbf{W}^{-1}(t)\}$

In order to estimate $\underline{X}(t)$ in time, we need to apply

Kalman Filter (KF)

The basic idea: 1. Since all the inputs are Gaussians, so is $\underline{X}(t)$

2. We know all about $\underline{X}(t)$ if its two first moments are known - $\underline{X}(t) \sim \mathbf{N}\{\hat{\underline{X}}(t), \hat{\mathbf{P}}(t)\}$

KF: Mean-Covariance Pair

1. We start by knowing the pair $\left\langle \hat{\underline{X}}(t-1)\,,\,\hat{\mathbf{P}}(t-1)\right\rangle$

2. Based on $\underline{X}(t)=\mathbf{G}(t)\underline{X}(t-1)+\underline{V}(t)$ we get the

Prediction Equations:

$$\tilde{\mathbf{P}}(t)=\mathbf{G}(t)\hat{\mathbf{P}}(t-1)\mathbf{G}^{T}(t)+\mathbf{Q}^{-1}(t)$$

$$\tilde{\underline{X}}(t)=\mathbf{G}(t)\hat{\underline{X}}(t-1)$$

3. Based on $\underline{Y}_A(t)=\mathbf{H}_A(t)\underline{X}(t)+\underline{N}_A(t)$ we get the

Update Equations:

$$\hat{\mathbf{P}}(t)=\left[\tilde{\mathbf{P}}^{-1}(t)+\mathbf{H}_A^{T}(t)\mathbf{W}_A(t)\mathbf{H}_A(t)\right]^{-1}$$

$$\hat{\underline{X}}(t)=\hat{\mathbf{P}}(t)\left[\tilde{\mathbf{P}}^{-1}(t)\tilde{\underline{X}}(t)+\mathbf{H}_A^{T}(t)\mathbf{W}_A(t)\underline{Y}_A(t)\right]$$

KF: Information Pair

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

The recursive equations become:

Interpolation:

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

$$\tilde{\underline{Z}}(t) = \tilde{\mathbf{L}}(t)\mathbf{G}(t)\hat{\mathbf{L}}^{-1}(t-1)\hat{\underline{Z}}(t-1)$$

Update:

$$\hat{\mathbf{L}}(t) = \tilde{\mathbf{L}}(t) + \mathbf{H}_A^T(t)\mathbf{W}_A(t)\mathbf{H}_A(t)$$

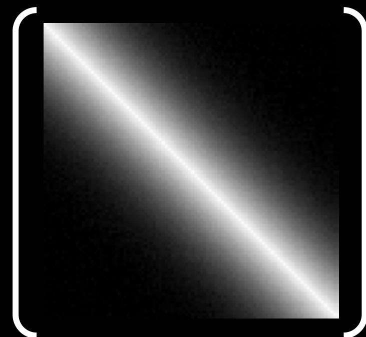
$$\hat{\underline{Z}}(t) = \tilde{\underline{Z}}(t) + \mathbf{H}_A^T(t)\mathbf{W}_A(t)\underline{Y}_A(t)$$

Presumably, there is nothing to gain in using the information pair, over the mean-covariance pair

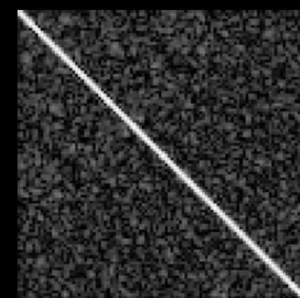
<u>Information Pair Is Better !!</u>

(for our application)

1. Experimental results indicate that the information matrix is sparser:



$^{-1}$ $=$



2. We intend to avoid the use of **Q**(t). Therefore, it is natural to achieve simplifying the equation

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

while approximating **Q**(t).

Avoiding Q(t)

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

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

and obtain that $\tilde{\mathbf{L}}(t) = \boxed{\dfrac{1}{1+\alpha(t)}}\mathbf{F}^T(t)\hat{\mathbf{L}}(t-1)\mathbf{F}(t)$ $\left[\mathbf{G}^{-1}(t) = \mathbf{F}(t)\right]$

$$\boxed{\begin{aligned}\hat{\mathbf{L}}(t) &= \boxed{\lambda(t)}\mathbf{F}^T(t)\hat{\mathbf{L}}(t-1)\mathbf{F}(t) + \mathbf{H}_A^T(t)\mathbf{W}_A(t)\mathbf{H}_A(t)\\ \underline{\hat{Z}}(t) &= \lambda(t)\mathbf{F}^T(t)\underline{\hat{Z}}(t-1) + \mathbf{H}_A^T(t)\mathbf{W}_A(t)\underline{Y}_A(t)\end{aligned}}$$

The Pseudo-RLS Algorithm

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

2. For t > 0,

➲ Update the information pair

$$\hat{\mathbf{L}}(t) = \lambda(t)\mathbf{F}^T(t)\hat{\mathbf{L}}(t-1)\mathbf{F}(t) + \mathbf{H}_A^T(t)\mathbf{W}_A(t)\mathbf{H}_A(t)$$

$$\underline{\hat{Z}}(t) = \lambda(t)\mathbf{F}^T(t)\underline{\hat{Z}}(t-1) + \mathbf{H}_A^T(t)\mathbf{W}_A(t)\underline{Y}_A(t)$$

➲Compute the output by $\boxed{\underline{\hat{X}}(t) = \hat{\mathbf{L}}^{-1}(t)\underline{\hat{Z}}(t)}$

Problem: Need to invert the information matrix !

The R-SD Algorithm

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

2. For t > 0,

   ➲ Update the information pair, as before

   ➲ Compute the output by R-SD iterations:

$$\hat{\underline{X}}_0(t)=\mathbf{G}(t)\hat{\underline{X}}_R(t-1)$$

Adopted from the assumed model

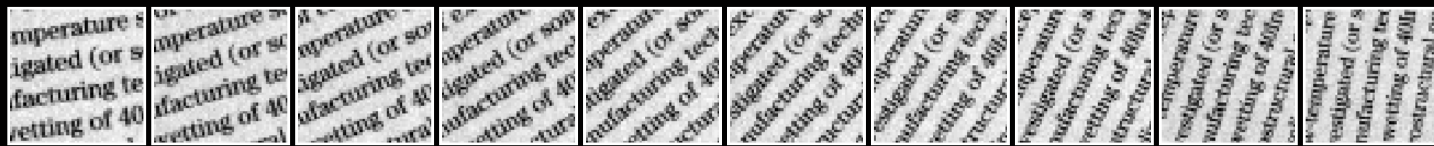
and for k=1,2, … ,R:

$$\hat{\underline{X}}_{k+1}(t)=\hat{\underline{X}}_k(t)-\mu\left[\hat{\mathbf{L}}(t)\hat{\underline{X}}_k(t)-\hat{\underline{Z}}(t)\right]$$

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

Dynamic Super-Resolution

Low Resolution Measurements

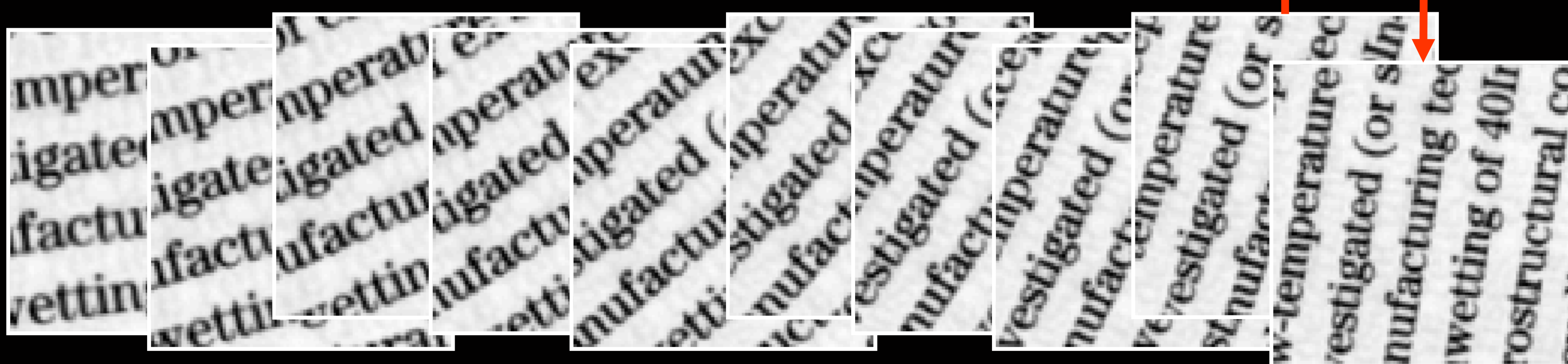


$\left\{\underline{Y}(t)\right\}_t$

$$\boxed{\underline{\hat{X}}(t) = f\left\{\underline{Y}(t),\ \underline{\hat{X}}(t-1)\right\}}$$

Dynamic Super-Resolution Algorithm

High Resolution Reconstructed Images



$\left\{\underline{\hat{X}}(t)\right\}_t$

The R-LMS Algorithm

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

2. For $t > 0$,

➲ Compute the output by R-SD iterations using the **intermediate** information pair:

$$\underline{\hat{X}}_0(t) = \mathbf{G}(t)\underline{\hat{X}}_R(t-1)$$

and for k=1,2, … ,R:

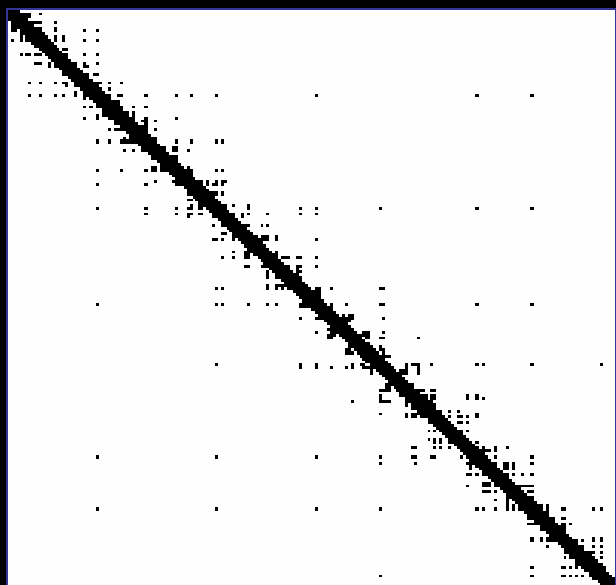
$$\underline{\hat{X}}_{k+1}(t) = \underline{\hat{X}}_k(t) - \mu\mathbf{H}_A^T(t)\mathbf{W}_A(t)\left[\mathbf{H}_A(t)\underline{\hat{X}}_k(t) - \underline{Y}_A(t)\right]$$

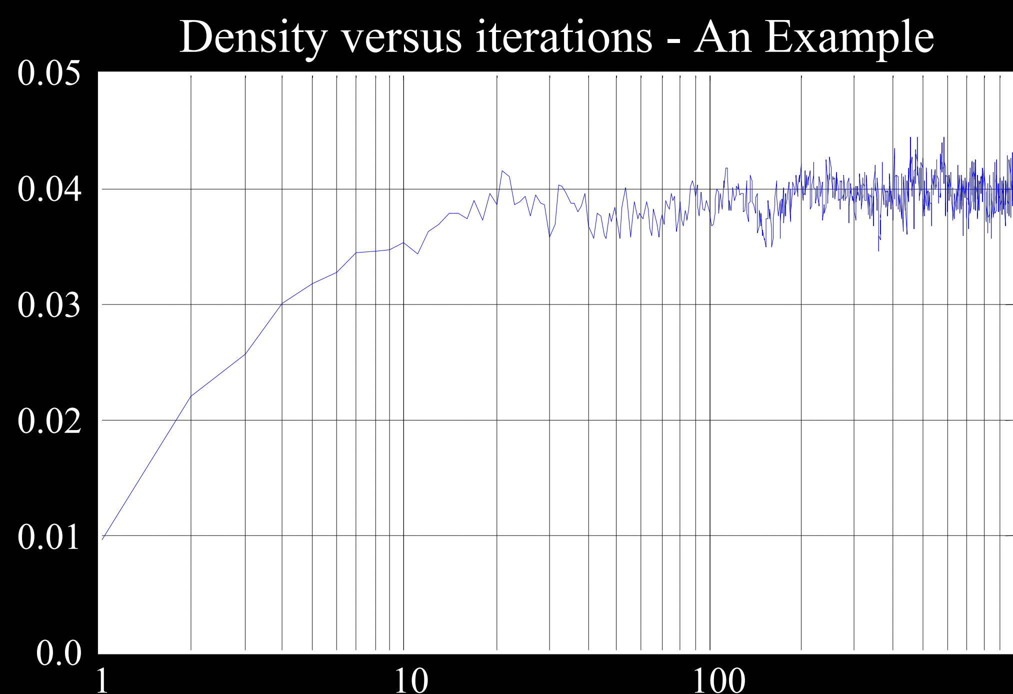
Also obtained if $\underline{\hat{X}}_R(t-1) \cong \hat{\mathbf{L}}^{-1}(t-1)\underline{\hat{Z}}(t-1)$
or if $\lambda(t)$ is set to zero

The Information Matrix

$$\hat{\mathbf{L}}(t) = \lambda(t)\mathbf{F}^{T}(t)\hat{\mathbf{L}}(t-1)\mathbf{F}(t) + \mathbf{M}(t)$$

Under some very reasonable assumptions, it is **PROVEN** that the the information matrix remains **SPARSE**





Convergence Properties

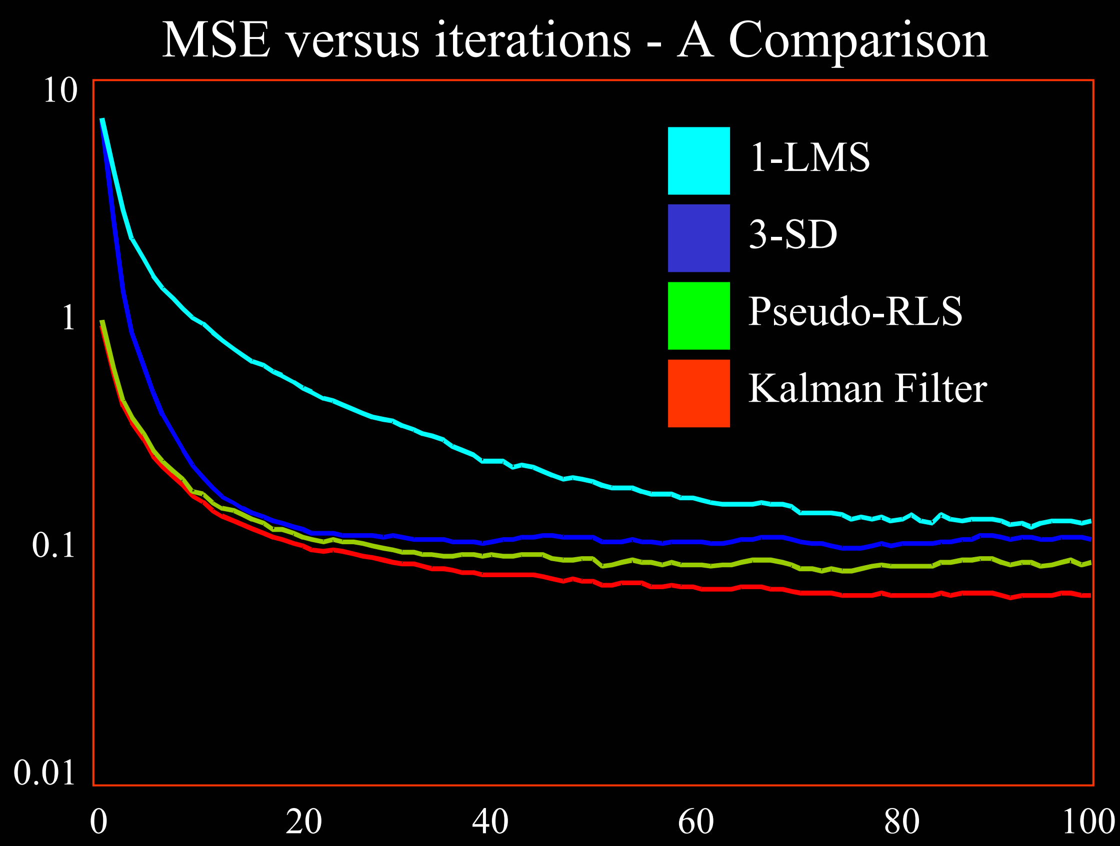
1. Bounds on the dynamic estimation error for the proposed Kalman Filter approximations (the P-RLS, the R-SD and the R-LMS) are obtained.
2. An important role in these convergence theorems plays the term

$$\left\| \underline{\hat{X}}_{PRLS}(t) - \mathbf{G}(t)\underline{\hat{X}}_{PRLS}(t-1) \right\|$$

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

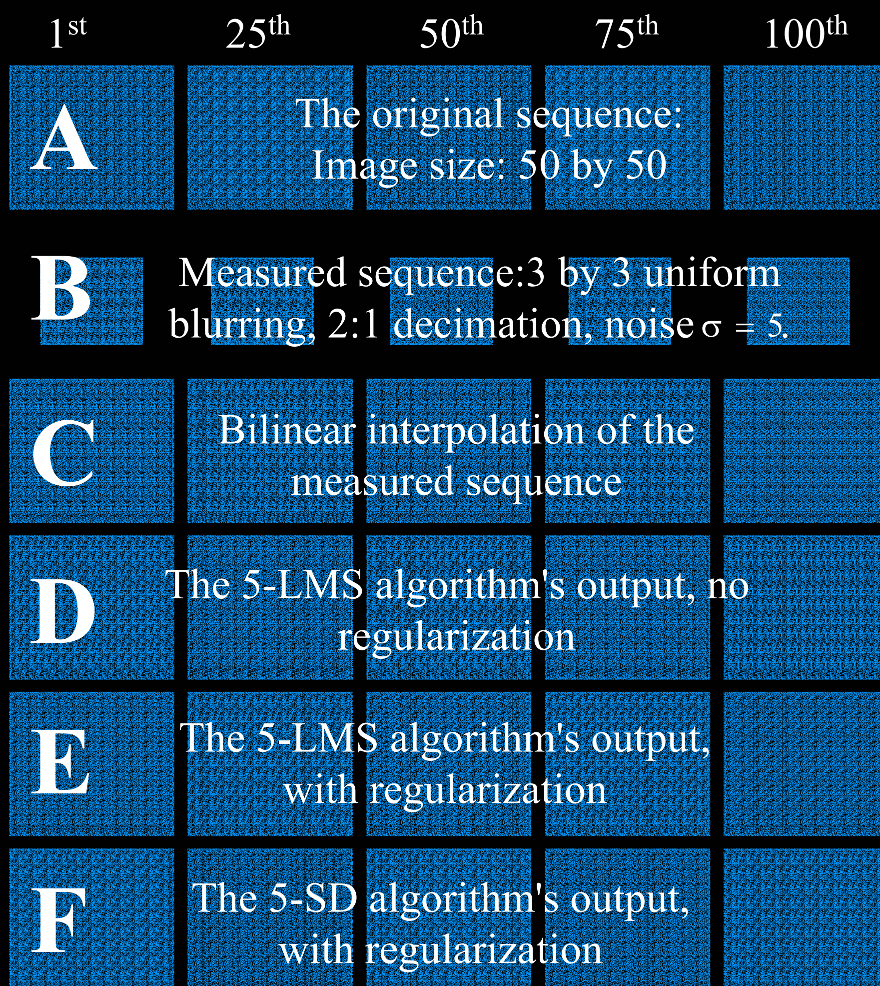
Dynamic Estimation Comparison - Low dimension (N=100) synthetic case



Results - Part 2

Higher dimension (N=2500) synthetic image sequences

Note: the motion and blur operations are assumed to be known apriori



Sequence 1
[Displacement+zoom]

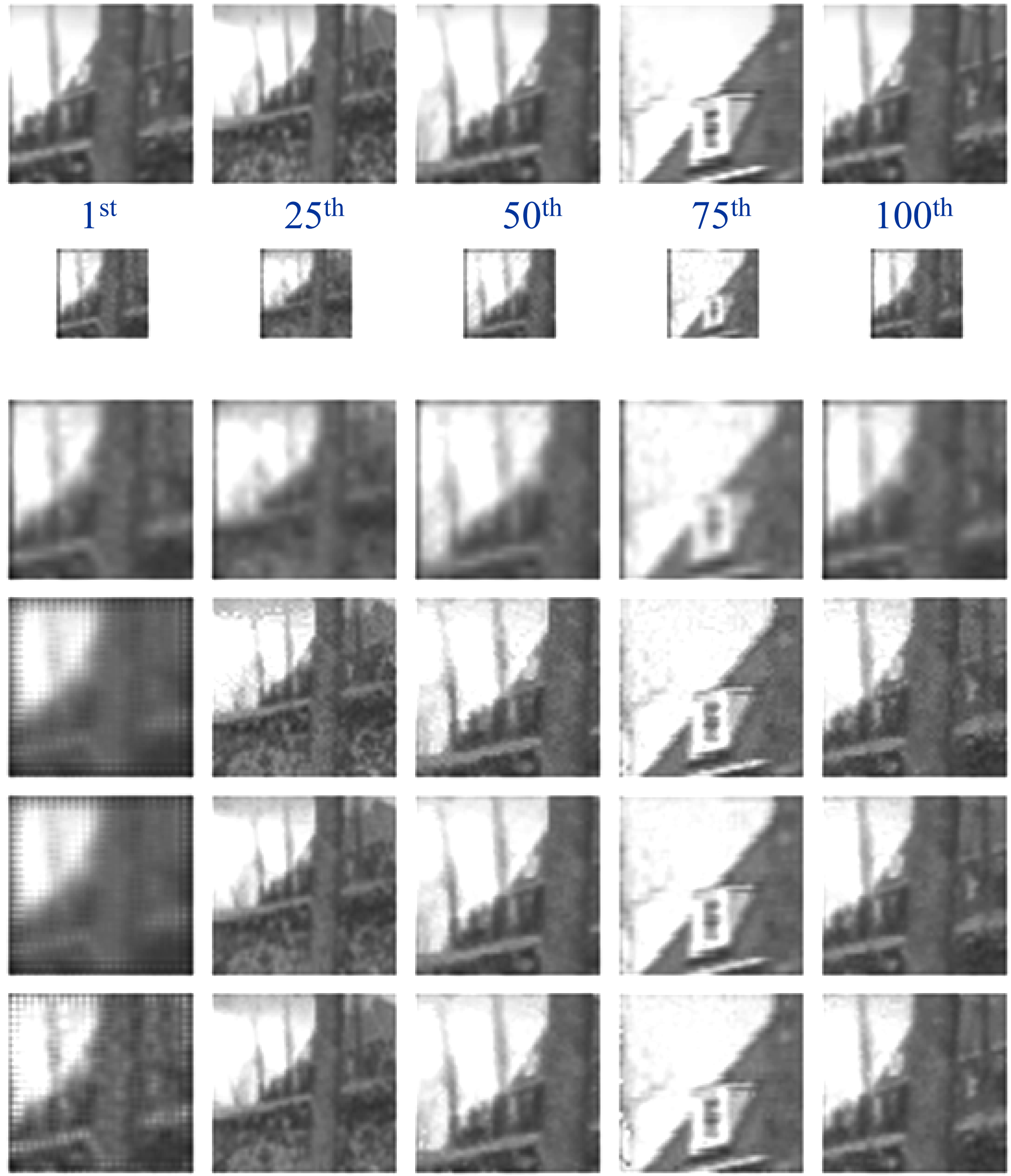
Measurements

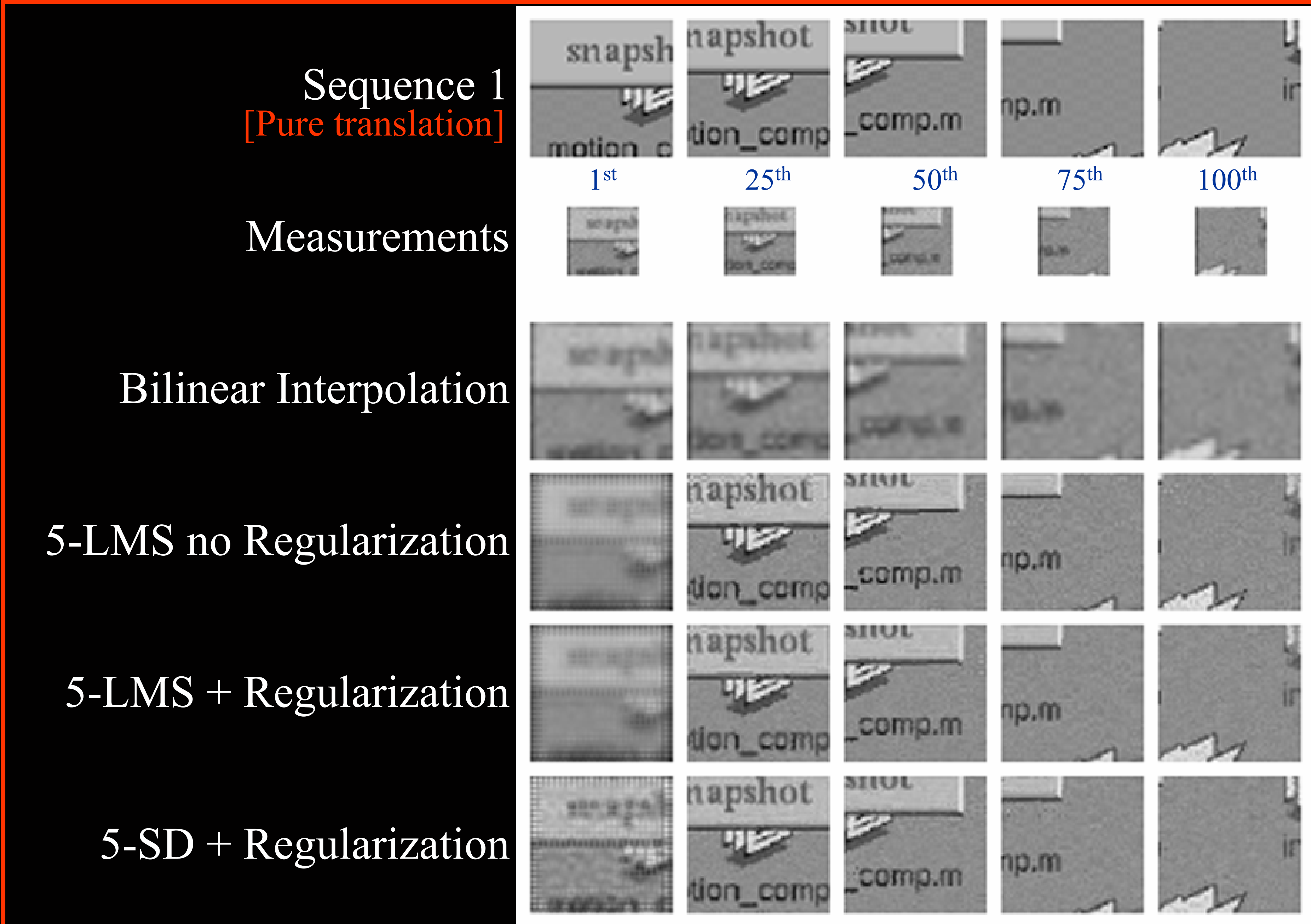
Bilinear Interpolation

5-LMS no Regularization

5-LMS + Regularization

5-SD + Regularization




Sequence 1
[Pure translation]
1st
25th
50th
75th
100th
Measurements
Bilinear Interpolation
5-LMS no Regularization
5-LMS + Regularization
5-SD + Regularization

Sequence 1
[Pure rotation]

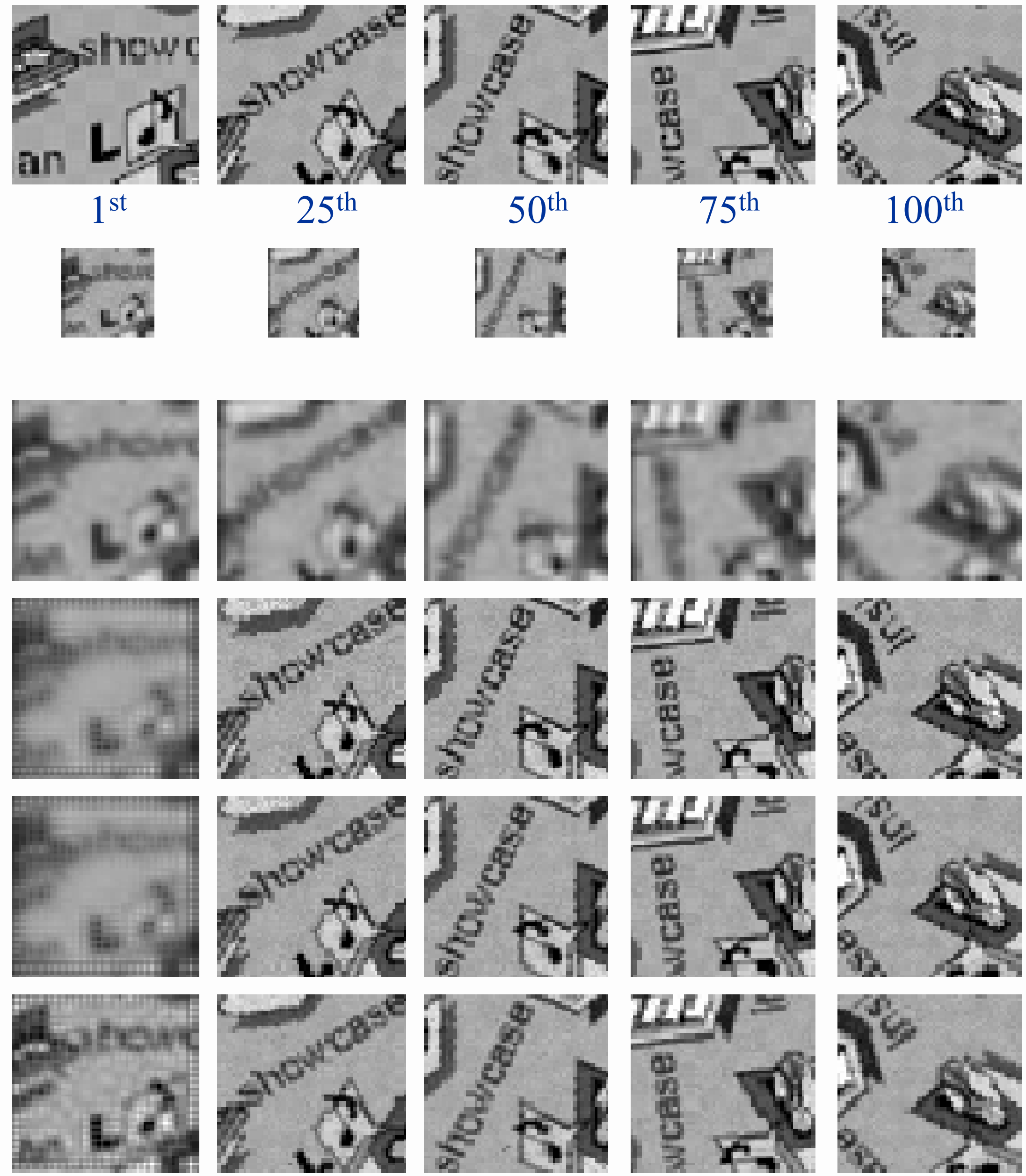
Measurements

Bilinear Interpolation

5-LMS no Regularization

5-LMS + Regularization

5-SD + Regularization



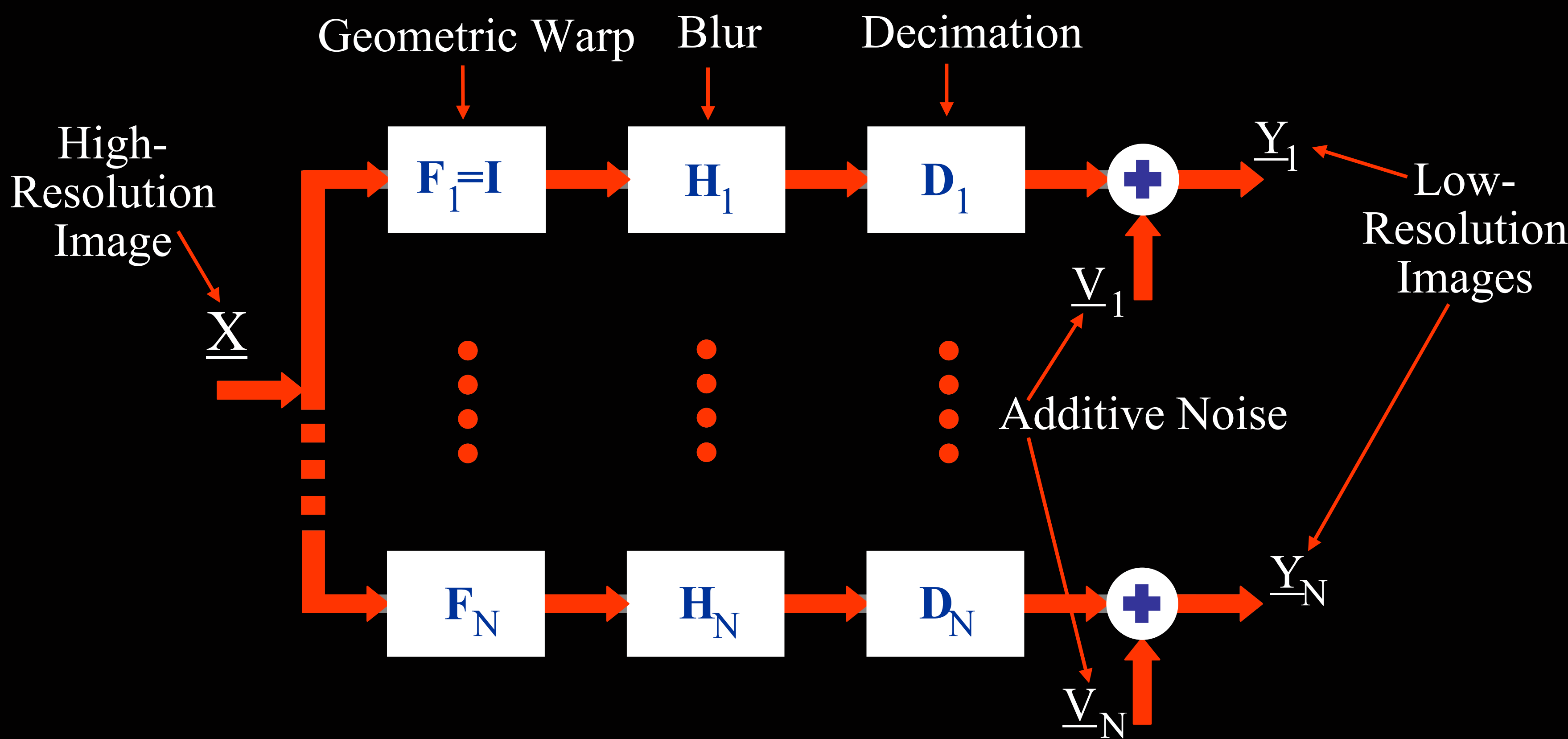
Conclusions

- 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



$$\left\{ \underline{Y}_k = \mathbf{D}_k\mathbf{H}_k\mathbf{F}_k\underline{X} + \underline{V}_k, \quad \underline{V}_k \sim \mathbf{N}\left\{0, \mathbf{W}_k^{-1}\right\} \right\}_{k=1}^{N}$$

The Model as One Equation

$$\left\{ \underline{Y}_k = \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \underline{X} + \underline{V}_k, \quad \underline{V}_k \sim \mathbf{N}\left\{0, \mathbf{W}_k^{-1}\right\} \right\}_{k=1}^{N}$$

$$\begin{bmatrix} \underline{Y}_1 \\ \underline{Y}_2 \\ \vdots \\ \underline{Y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{D}_1\mathbf{H}_1\mathbf{F}_1 \\ \mathbf{D}_2\mathbf{H}_2\mathbf{F}_2 \\ \vdots \\ \mathbf{D}_N\mathbf{H}_N\mathbf{F}_N \end{bmatrix} \underline{X} + \begin{bmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \vdots \\ \underline{V}_N \end{bmatrix} \qquad \begin{bmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \vdots \\ \underline{V}_N \end{bmatrix} \sim \mathbf{N}\left\{ 0, \begin{bmatrix} \mathbf{W}_1 & & & 0 \\ & \mathbf{W}_2 & & \\ & & \ddots & \\ 0 & & & \mathbf{W}_N \end{bmatrix}^{-1} \right\}$$

Iterative Reconstruction

$$\left\{\begin{aligned} \mathbf{R} &= \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \\ \underline{P} &= \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k \underline{Y}_k \end{aligned}\right\} \Rightarrow \mathbf{R}\hat{\underline{X}} = \underline{P}$$

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

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

$$\hat{\underline{X}}_{j+1} = \hat{\underline{X}}_j - \mu \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k \left[\underline{Y}_k - \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \hat{\underline{X}}_j \right]$$

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. More over, 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

$$\hat{\underline{X}}_{j+1} = \hat{\underline{X}}_j - \mu \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}_k^T \mathbf{D}_k^T \mathbf{W}_k \left[\underline{Y}_k - \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \hat{\underline{X}}_j \right]$$

$$\hat{\underline{X}}_{j+1} = \hat{\underline{X}}_j - \mu \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}^T \mathbf{D}^T \left[\underline{Y}_k - \mathbf{D} \mathbf{H} \mathbf{F}_k \hat{\underline{X}}_j \right] =$$

$$= \hat{\underline{X}}_j - \mu \mathbf{H}^T \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{D}^T \left[\underline{Y}_k - \mathbf{D} \mathbf{F}_k \mathbf{H} \hat{\underline{X}}_j \right]$$

where we use the fact that

block-Circulant matrices commute

Important Shortcut

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

$$\underline{\hat{X}}_{j+1} = \underline{\hat{X}}_j - \mu \mathbf{H}^T \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{D}^T \left[\underline{Y}_k - \mathbf{D}\mathbf{F}_k \mathbf{H} \underline{\hat{X}}_j \right]$$

$$\underline{\hat{Z}}_{j+1} = \underline{\hat{Z}}_j - \mu \mathbf{H}\mathbf{H}^T \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{D}^T \left[\underline{Y}_k - \mathbf{D}\mathbf{F}_k \underline{\hat{Z}}_j \right] =$$

$$= \underline{\hat{Z}}_j - \mu \mathbf{H}\mathbf{H}^T \Bigg[\underbrace{\sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{D}^T \underline{Y}_k}_{=\underline{\tilde{P}}} - \underbrace{\sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{D}^T \mathbf{D}\mathbf{F}_k}_{=\tilde{\mathbf{R}}} \underline{\hat{Z}}_j \Bigg] = \underline{\hat{Z}}_j - \mu \mathbf{H}\mathbf{H}^T \left(\underline{\tilde{P}} - \tilde{\mathbf{R}} \underline{\hat{Z}}_j \right)$$

Descent Direction - Theory

- Given the quadratic function* $f\{\underline{x}\} = \frac{1}{2}\underline{x}^T\tilde{\mathbf{R}}\underline{x} - \underline{\tilde{P}}^T\underline{x} + c$,

  it's optimal Solution satisfies $\tilde{\mathbf{R}}\underline{\hat{x}}_{opt} = \underline{\tilde{P}}$.

- Any algorithm of the form $\underline{\hat{x}}_{j+1} = \underline{\hat{x}}_j - \alpha\mathbf{M}\left(\tilde{\mathbf{R}}\underline{\hat{x}}_j - \underline{\tilde{P}}\right)$

  converges to $\underline{\hat{x}}_{opt}$ for sufficiently small α and **M**>0.

- In our case **M=HH**^{T} (positive semi-definite). It means that the

  error $\underline{\hat{x}}_j - \underline{\hat{x}}_{opt}$ in the null space of **M** cannot converge.

* $\tilde{\mathbf{R}}$ is assumed to be positive definite

Is it a Problem?

Positive Semi-definite M

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

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

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

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

$$\left(I - \alpha\mathbf{M}\tilde{\mathbf{R}}\right)^{j+1}\underline{u} = \underline{u}$$

Positive Semi-definite M

$$\underline{\hat{x}}_0 - \underline{\hat{x}}_{opt} = \underline{\hat{e}}_0 + \underline{\hat{f}}_0$$

In the null-space of $\mathbf{M}\widetilde{\mathbf{R}}$

Orthogonal to the null-space of $\mathbf{M}\widetilde{\mathbf{R}}$

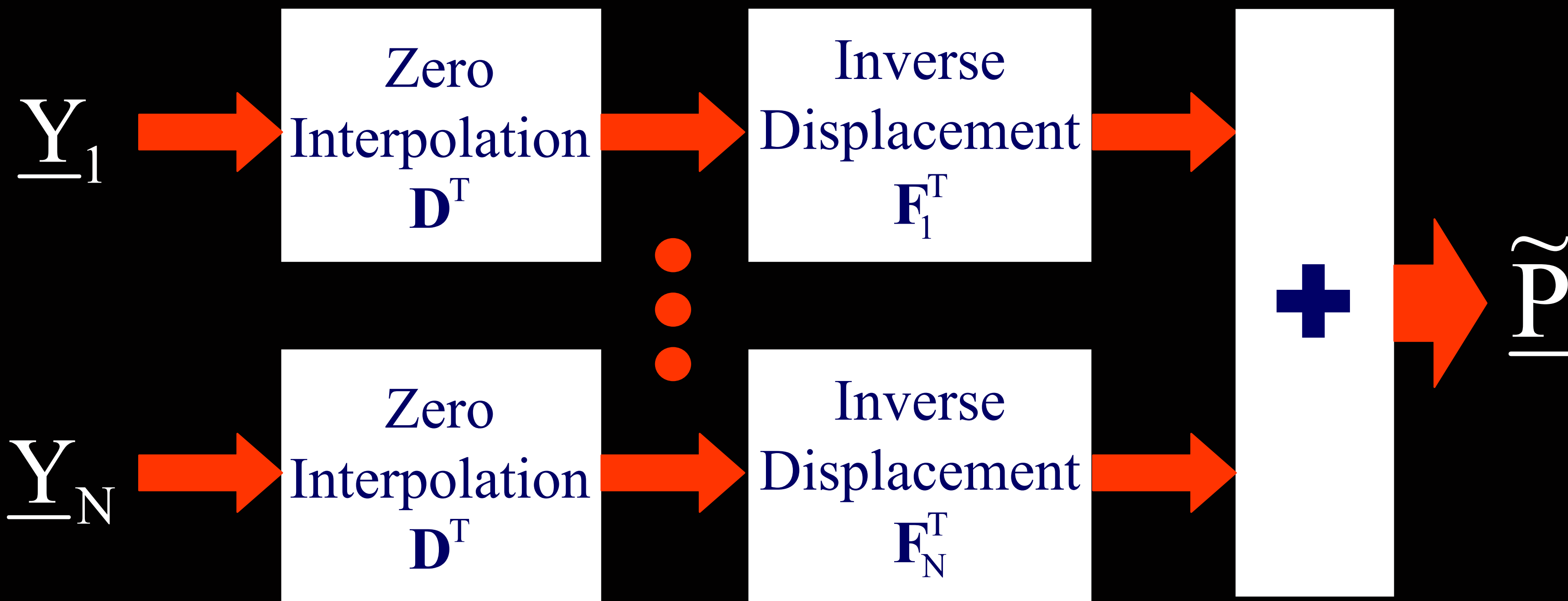
$$\underline{\hat{e}}_{j+1} + \underline{\hat{f}}_{j+1} = \left(\mathrm{I} - \alpha\mathbf{M}\widetilde{\mathbf{R}}\right)^{j+1}\left(\underline{\hat{e}}_0 + \underline{\hat{f}}_0\right) = \left(\mathrm{I} - \alpha\mathbf{M}\widetilde{\mathbf{R}}\right)^{j+1}\underline{\hat{e}}_0 + \underline{\hat{f}}_0$$

The null-space of $\mathbf{M}\widetilde{\mathbf{R}}$ 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 ?

$$\underline{\tilde{P}} = \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{D}^T \underline{Y}_k \ \mathbf{?}$$

It turns out that this is a motion-compensated average of the input images



What is R ?

$$\tilde{\mathbf{R}} = \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{D}^T \mathbf{D} \mathbf{F}_k \ ?$$

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 $\forall j,\ \tilde{\mathbf{R}}[j,j] \geq 1$

To Conclude

$$\underline{\hat{Z}}_{j+1}=\underline{\hat{Z}}_j-\mu\mathbf{H}\mathbf{H}^T\left(\underline{\tilde{P}}-\tilde{\mathbf{R}}\underline{\hat{Z}}_j\right)$$

$$\underline{\hat{Z}}_{opt}=\tilde{\mathbf{R}}^{-1}\underline{\tilde{P}}$$

and it is easy to compute this solution – One division by integer per pixel !!!!

Having found $\underline{\hat{Z}}_{opt}$, since it is defined by

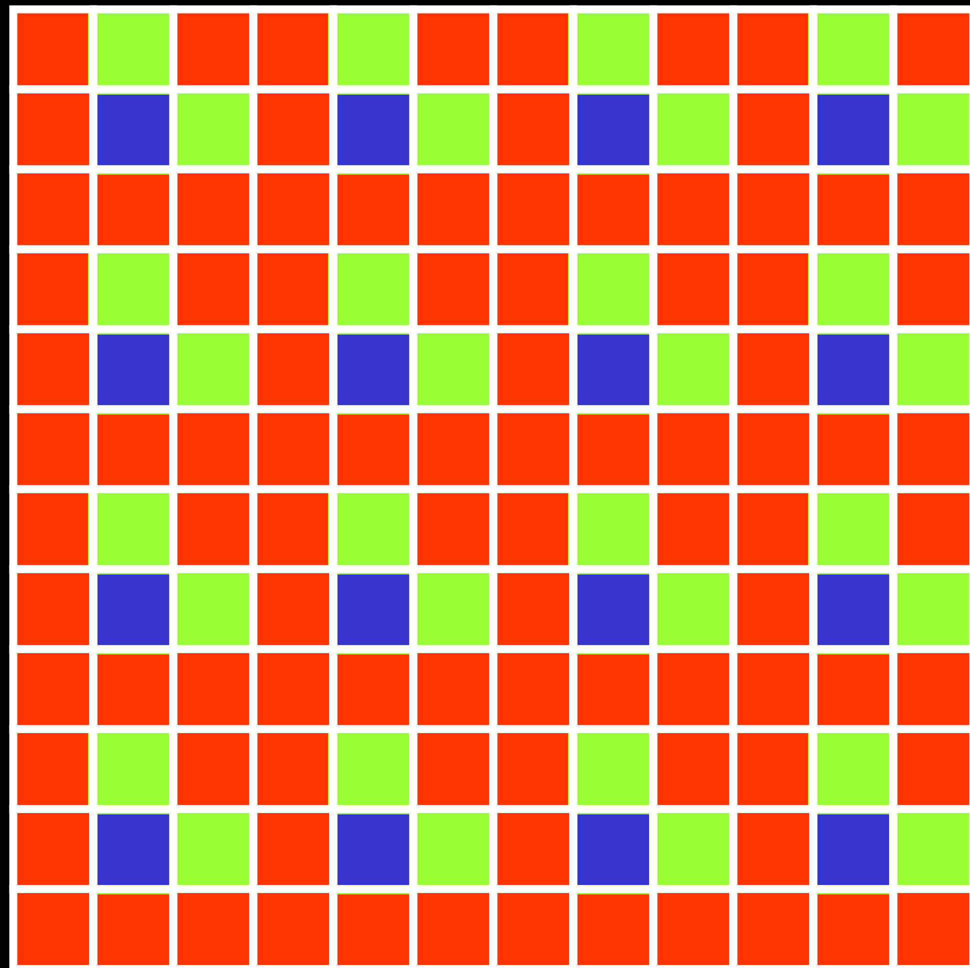
$$\underline{\hat{Z}}_j=\mathbf{H}\underline{\hat{X}}_j$$

We have to apply a classic image restoration procedure to recover $\underline{\hat{X}}_{opt}$ (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 $\underline{\hat{X}} = \left[\mathbf{H}^T\mathbf{H} + \lambda\mathbf{S}^T\mathbf{W}\mathbf{S}\right]^{-1}\mathbf{H}^T\underline{Y}$ is edge preserving but not space-invariant.

Instead use

$$\underline{\hat{X}}_1 = \left[\mathbf{H}^T\mathbf{H} + \lambda_1\mathbf{S}^T\mathbf{S}\right]^{-1}\mathbf{H}^T\underline{Y}$$

$$\underline{\hat{X}}_2 = \left[\mathbf{H}^T\mathbf{H} + \lambda_2\mathbf{S}^T\mathbf{S}\right]^{-1}\mathbf{H}^T\underline{Y}$$

where $\lambda_1 < \lambda_{opt} < \lambda_2$.

Thus, $\underline{X}_1$ and $\underline{X}_2$ 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):

$$\underline{\hat{X}}_{Final} = \mathbf{W}\underline{\hat{X}}_1 + (\mathrm{I} - \mathbf{W})\underline{\hat{X}}_2$$

Fast SSR (2) - Periodic-Step SD

A numerical method to speed-up convergence

Relation to Super-Resolution

$$\left\{ \underline{Y}_k = \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \underline{X} + \underline{V}_k, \quad \underline{V}_k \sim \mathbf{N}\left\{0, \mathbf{W}_k^{-1}\right\} \right\}_{k=1}^{N}$$

$$\begin{bmatrix} \underline{Y}_1 \\ \underline{Y}_2 \\ \vdots \\ \underline{Y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{D}_1\mathbf{H}_1\mathbf{F}_1 \\ \mathbf{D}_2\mathbf{H}_2\mathbf{F}_2 \\ \vdots \\ \mathbf{D}_N\mathbf{H}_N\mathbf{F}_N \end{bmatrix} \underline{X} + \begin{bmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \vdots \\ \underline{V}_N \end{bmatrix}$$

Basic Assumptions

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

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

Basic Assumptions

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

  $\rightarrow$ If LS (ML) is applied, we get

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

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

⇨ Turn to iterative methods

Simple Iterative Method - SD

$$f\{\underline{x}\} = \left\|\underline{y} - \mathbf{C}\underline{x}\right\|_2^2 \Rightarrow \text{Min.} \qquad \Rightarrow \qquad \frac{\partial f\{\underline{x}\}}{\partial \underline{x}} = \mathbf{C}^T\left(\underline{y} - \mathbf{C}\underline{x}\right)$$

Using the Steepest-Descend idea we get

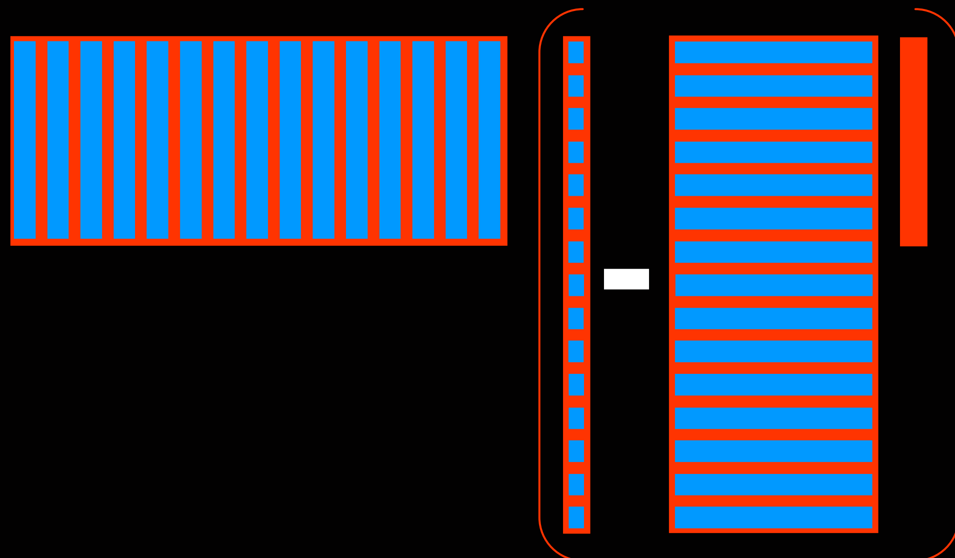
$$\begin{aligned}\hat{\underline{x}}_{k+1} &= \hat{\underline{x}}_k - \mu\mathbf{C}^T\left(\underline{y} - \mathbf{C}\hat{\underline{x}}_k\right) = \\ &= \hat{\underline{x}}_k - \mu\sum_{j=1}^{L}\underline{C}(j)\left[y(j) - \underline{C}^T(j)\hat{\underline{x}}_k\right]\end{aligned}$$

So we see that the gradient is built from L separate contributions, each obtained from a different measurement

Decomposition of the Gradient

$$\underline{\hat{x}}_{k+1} = \underline{\hat{x}}_k - \mu \mathbf{C}^T \left(\underline{y} - \mathbf{C} \underline{\hat{x}}_k \right) =$$

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



Periodic-Step SD

Instead of using $\hat{\underline{x}}_{k+1} = \hat{\underline{x}}_k - \mu \sum_{j=1}^{L} \underline{C}(j)\left[y(j) - \underline{C}^T(j)\hat{\underline{x}}_k\right]$

update the estimate of x for each
SCALAR measurement

$$\hat{\underline{x}}_{k+\frac{j+1}{L}} = \hat{\underline{x}}_{k+\frac{j}{L}} - \mu \underline{C}(j)\left[y(j) - \underline{C}^T(j)\hat{\underline{x}}_{k+\frac{j}{L}}\right]$$

for $\quad k = 0, 1, 2, 3, \quad \ldots.$

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 $0 < \mu < \underset{1 \le j \le L}{\mathrm{Min}} \left\{ 2 / \underline{C}^T(j)\underline{C}(j) \right\}$
- The convergence is to the LS optimal solution only if
  - Infitisimal step-size $\mu \rightarrow 0$,
  - Diminishing step-size $\mu_k \rightarrow 0$, 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 σ=3.

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

This algorithm effectively converges after one iteration

Original Bilinear int.

LS result PSSD Result