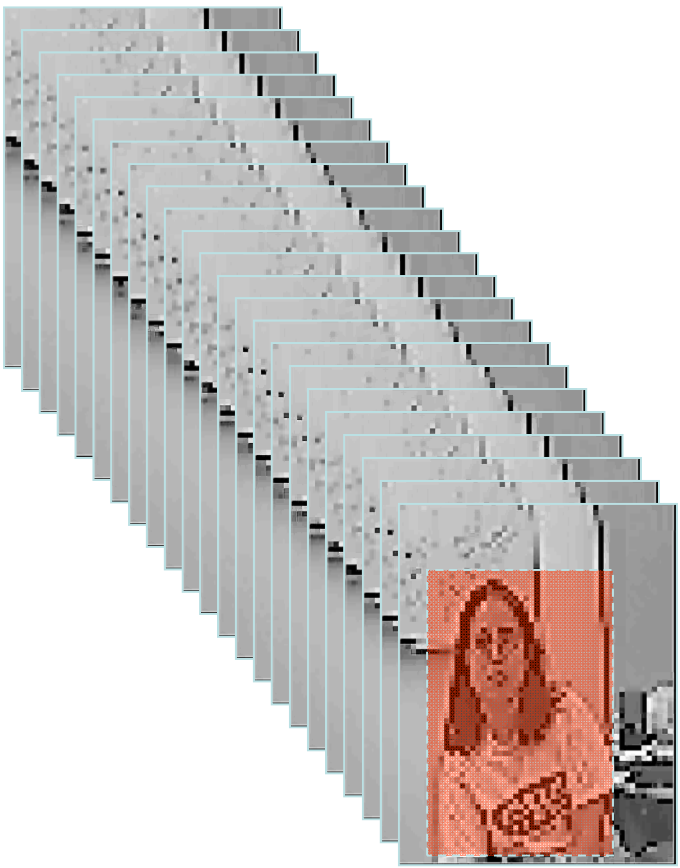
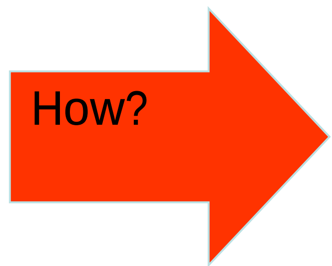
Super-Resolution from Multiple Images

Slides from Miki Elad, Yossi Rubner

Basic Super-Resolution Idea

Given: A set of low-quality images:



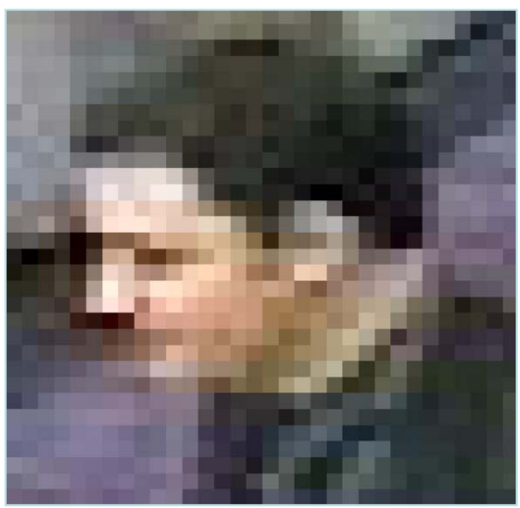


Required: Fusion of these images into a higher resolution image

Comment: This is an actual super-resolution reconstruction result

Example – Surveillance

40 images
ratio 1:4



Example – Enhance Mosaics

Work In Super Resolution

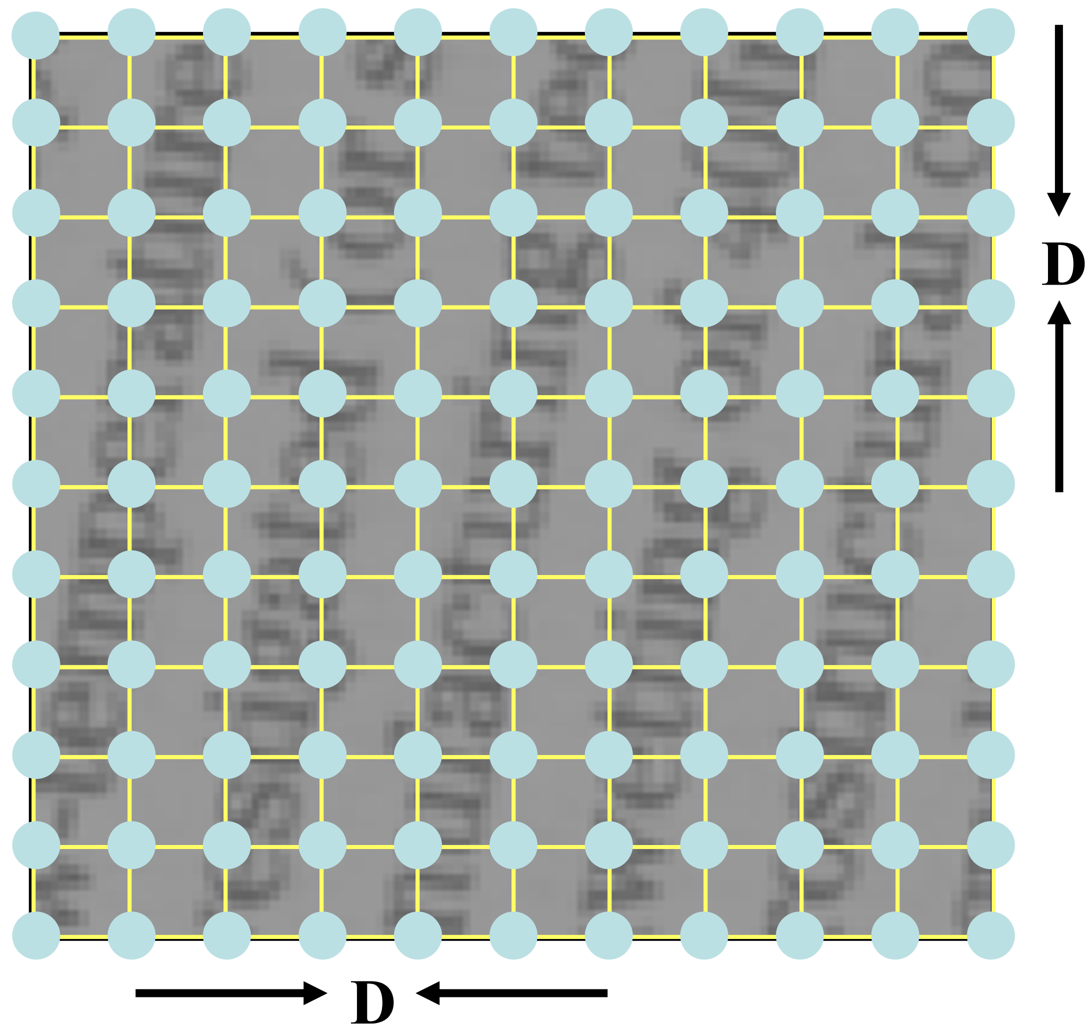
People	Place	Years
Peleg, Irani, Werman, Keren, Schweitzer	**HUJI**	**1987-1994**
Kim, Bose, Valenzuela	Penn. State	1990-1993
Patti, Tekalp, Zesan, Ozkan, Altunbasak	Rochester	1992-1998
Morris, Cheeseman, Smelyanskiy, Maluf	NASA-AMES	1992-2002
Ur & Gross	TAUI	1992-1993
Elad, Feuer, Sagi, Hel-Or	Technion	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.

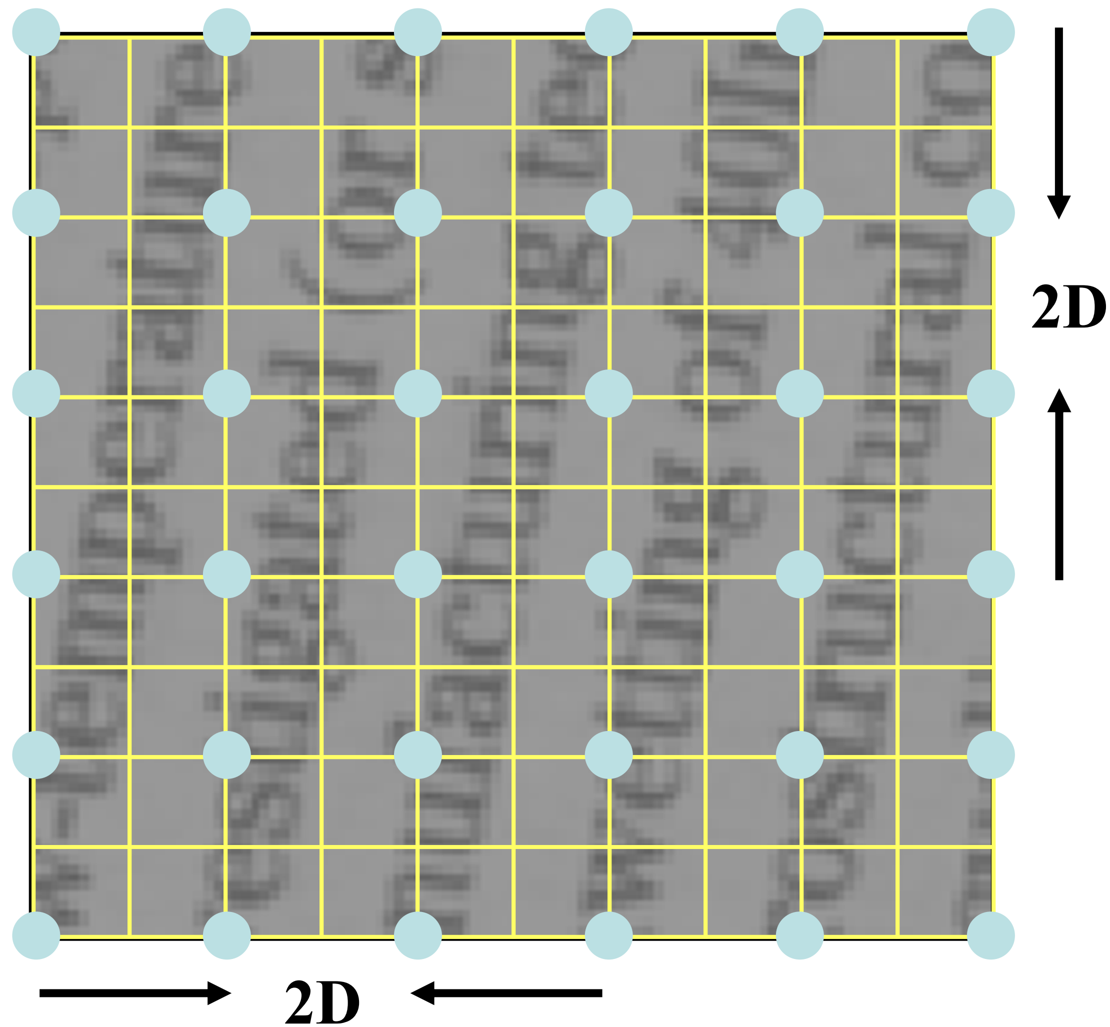
Intuition

We would like to get a uniform sampling with sampling distance $\leq$**D**



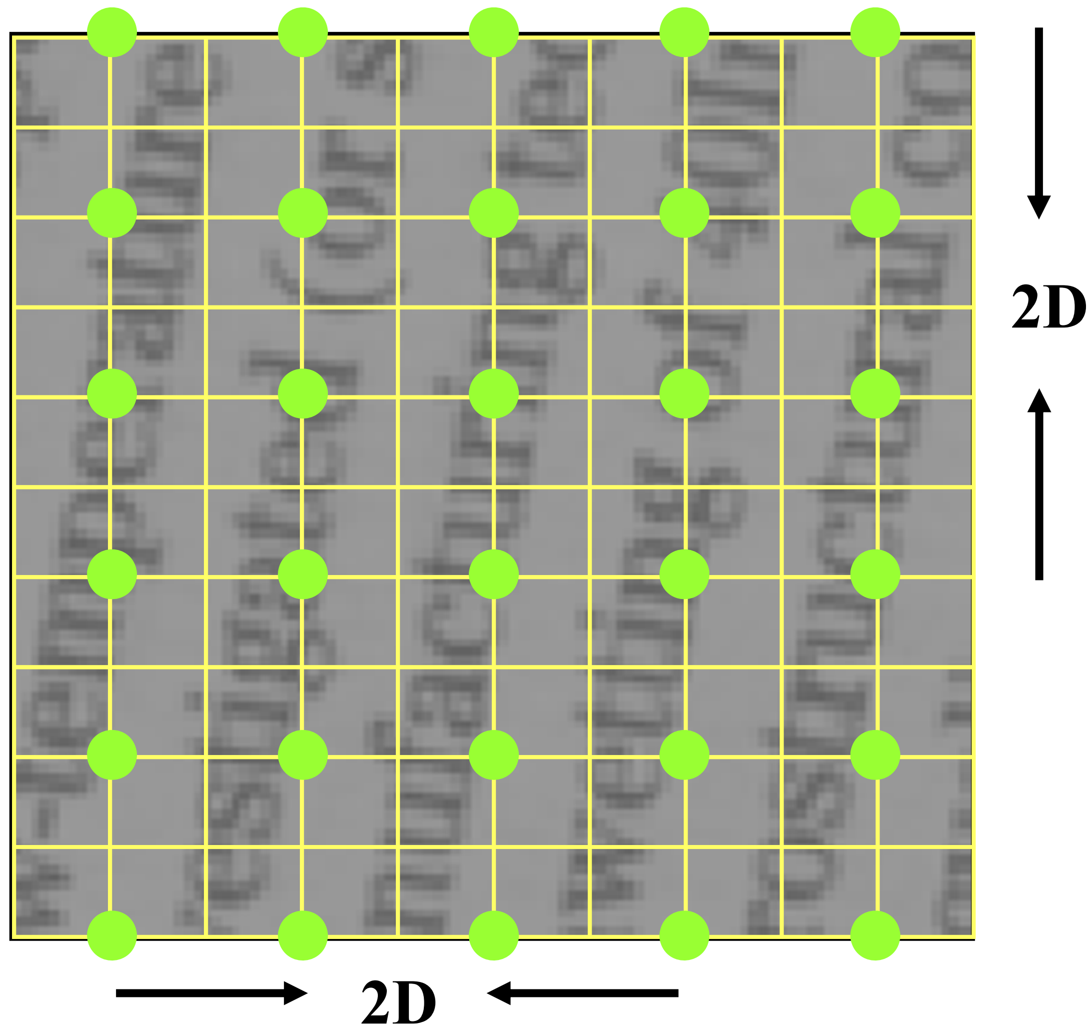
Intuition

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



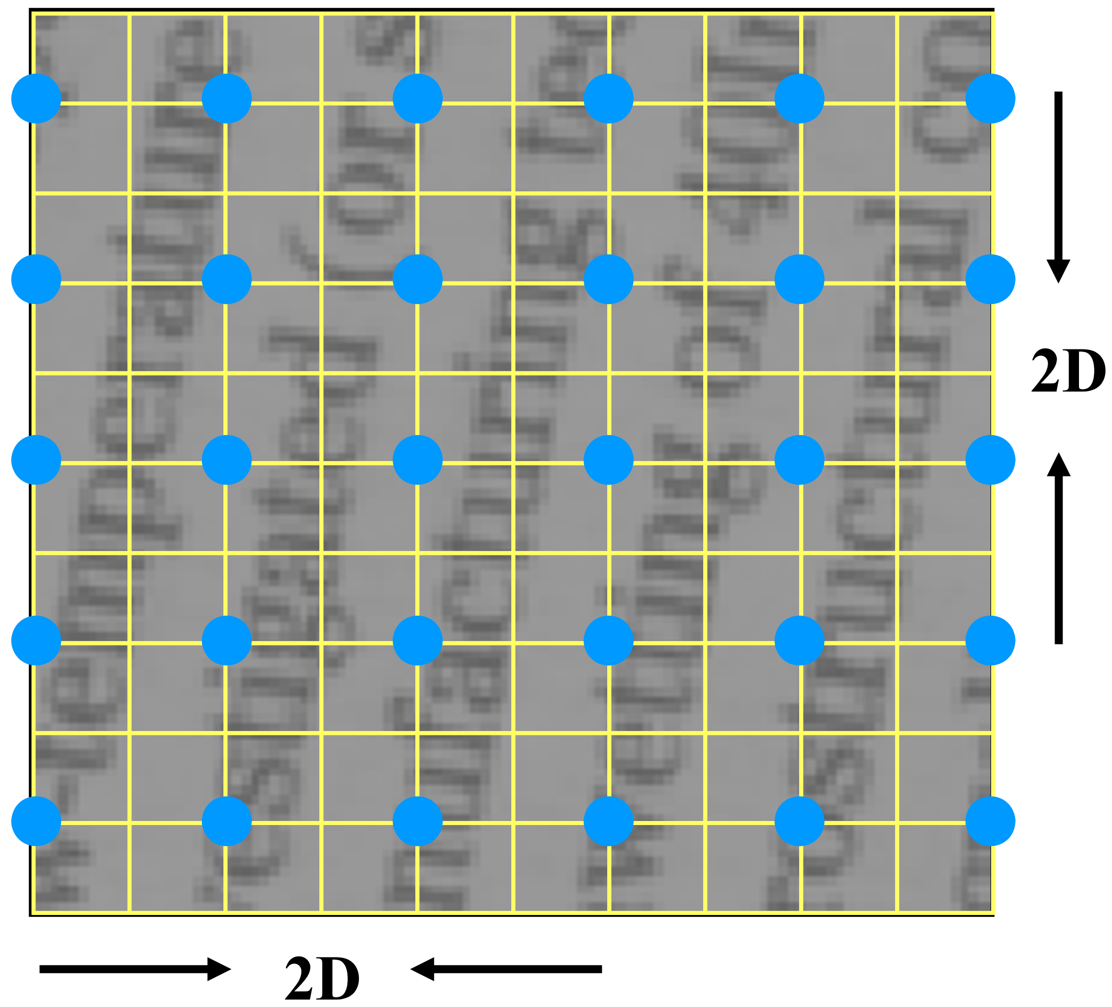
Intuition

However, if we take a second picture, shifting the camera 'slightly to the right' we obtain:



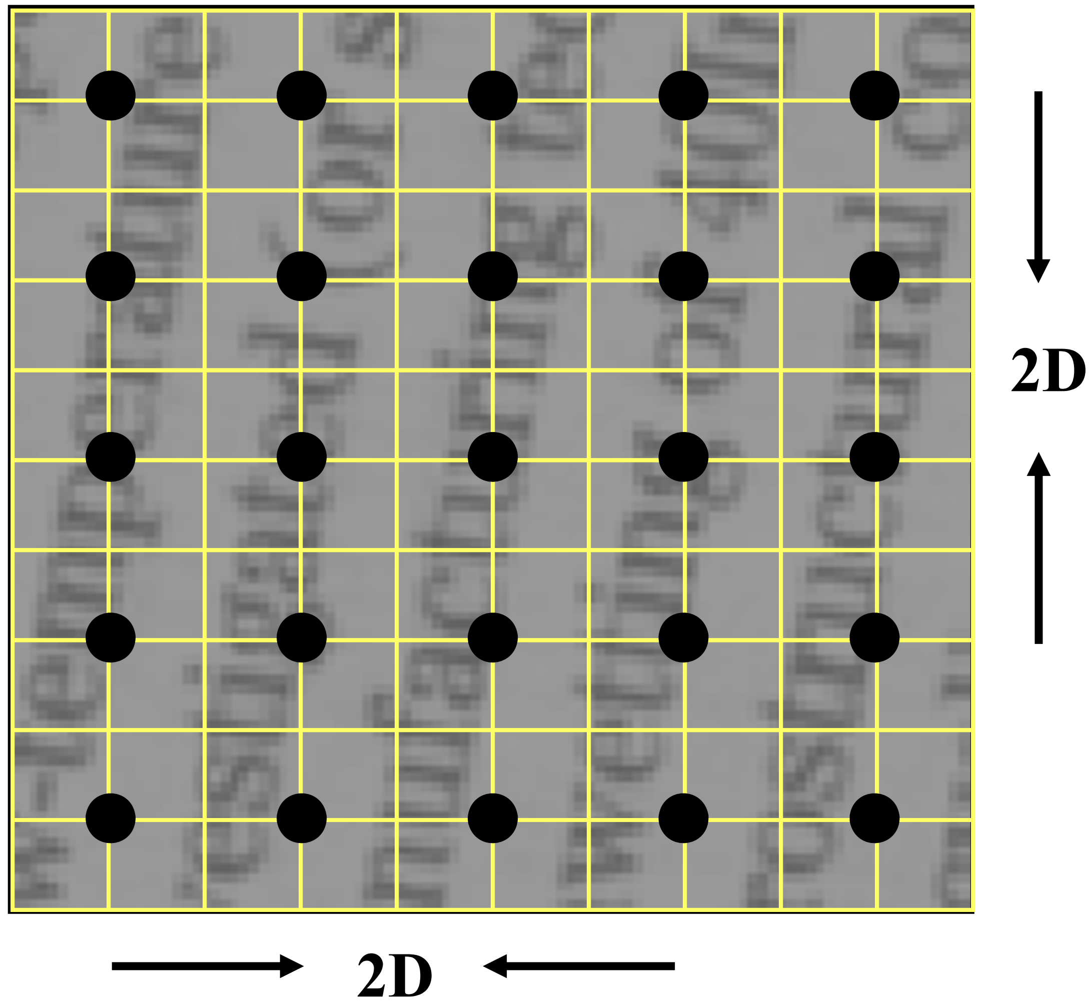
Intuition

Similarly, by shifting down we get a third image:



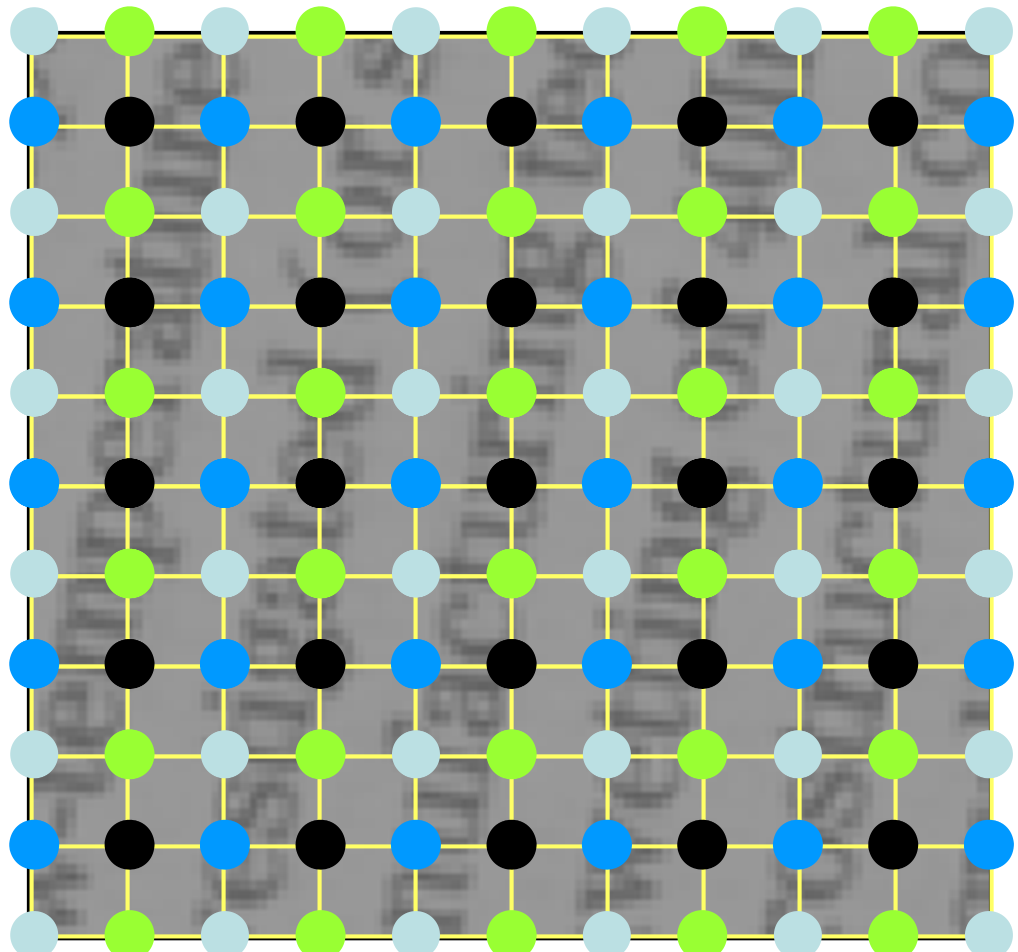
Intuition

And finally, by shifting down and to the right we get the fourth image:

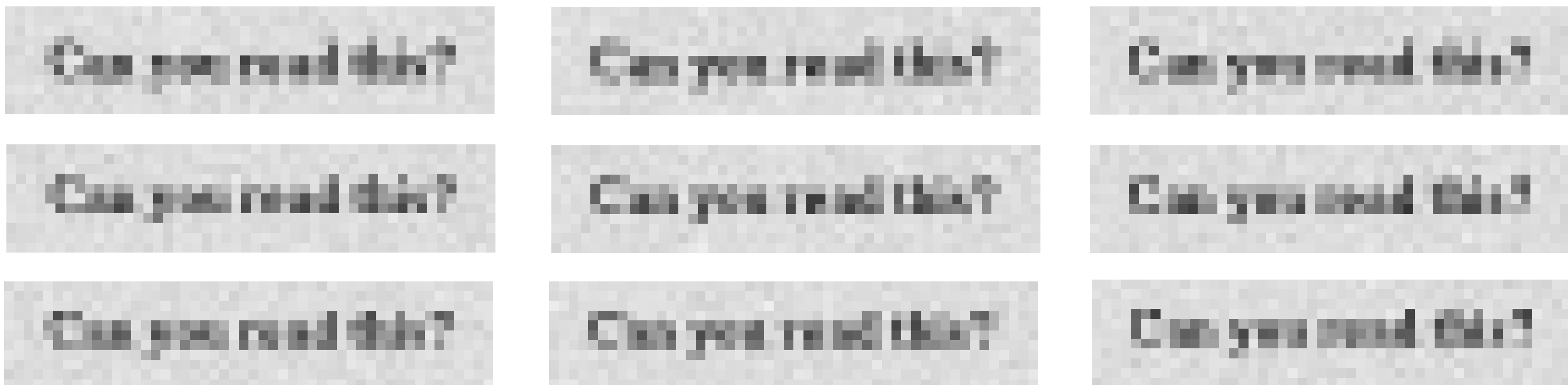


Intuition

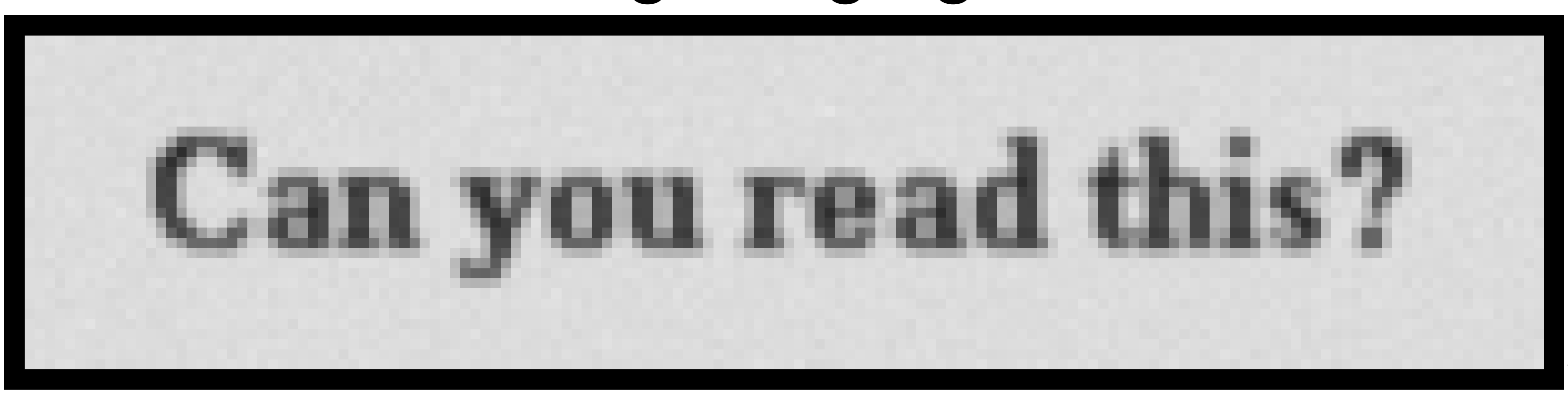
It is trivial to see that interlacing the four images, we get that the desired resolution is obtained, and thus perfect reconstruction is guaranteed.



A Small Example

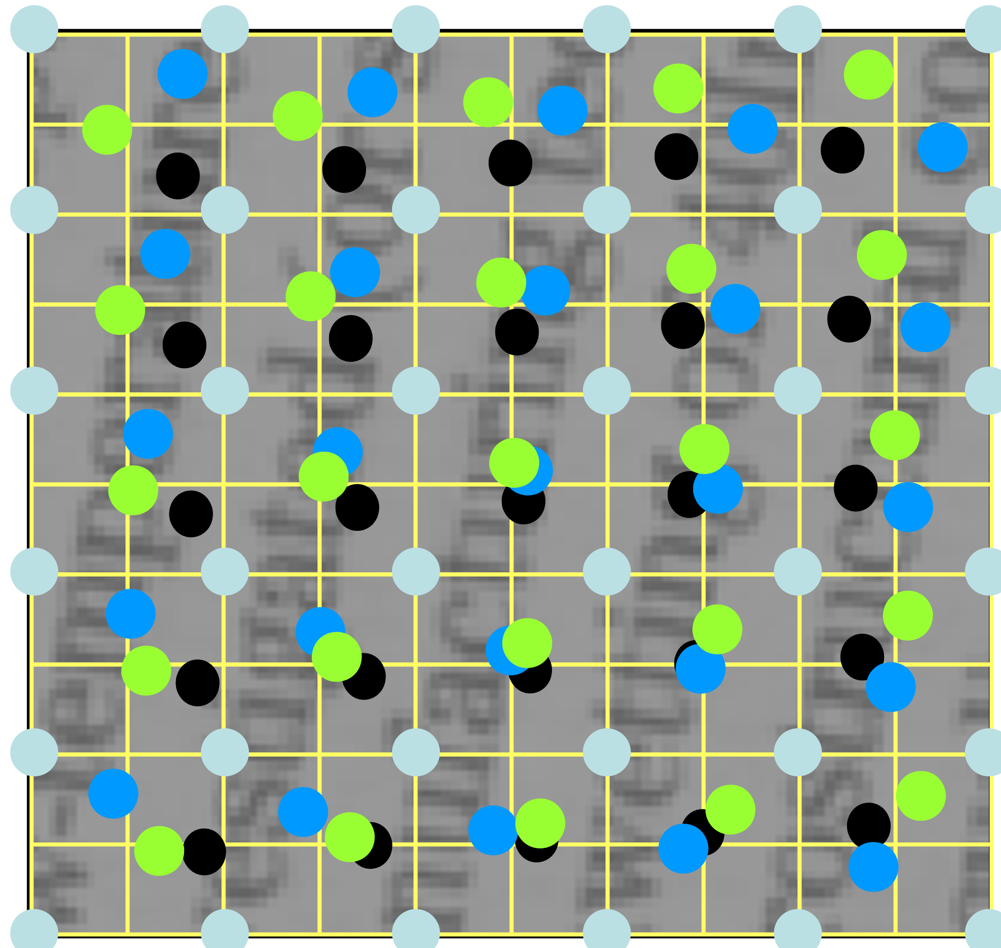


3:1 scale-up in each axis using 9 images
Translation of 1/3 pixel between images
Uniform blur during imaging



Rotation/Scale/Disp.

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



But Life is more Interesting…

- Motion is not what we want:
  – Non-uniform translations
  – Rotations, perspective, moving objects
- Blur: Optical & Sensor
  – Spatially variant blur
  – Temporally variant blur
- Noise
  – Sensor Noise, Quantization Noise
  (Simulations with float or char…)

Image Formation

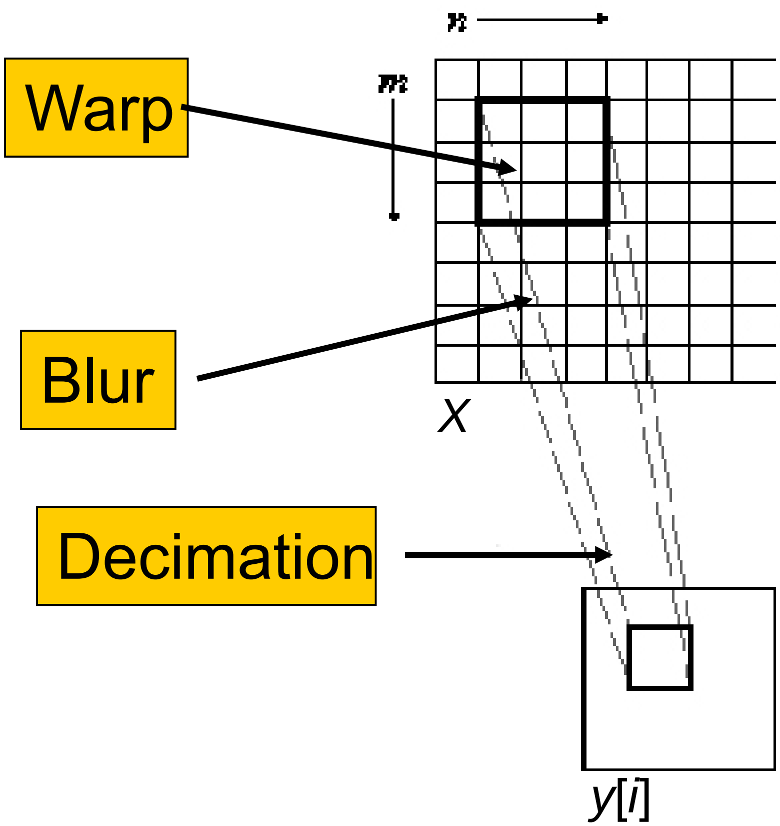
Scene	Geometric transformation	Optical Blur	Sampling	Noise
HR	$\mathbf{F}_k$	$\mathbf{H}_k$	$\mathbf{D}_k$	LR

Can we write these steps as linear operators?

$$\mathrm{LR} = \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \cdot \mathrm{HR}$$

Super resolution by Simulating the Imaging process

Assumption: The low resolution images *y*[1]…*y*[N] are the result of projection and sampling of a high resolution image *X*.

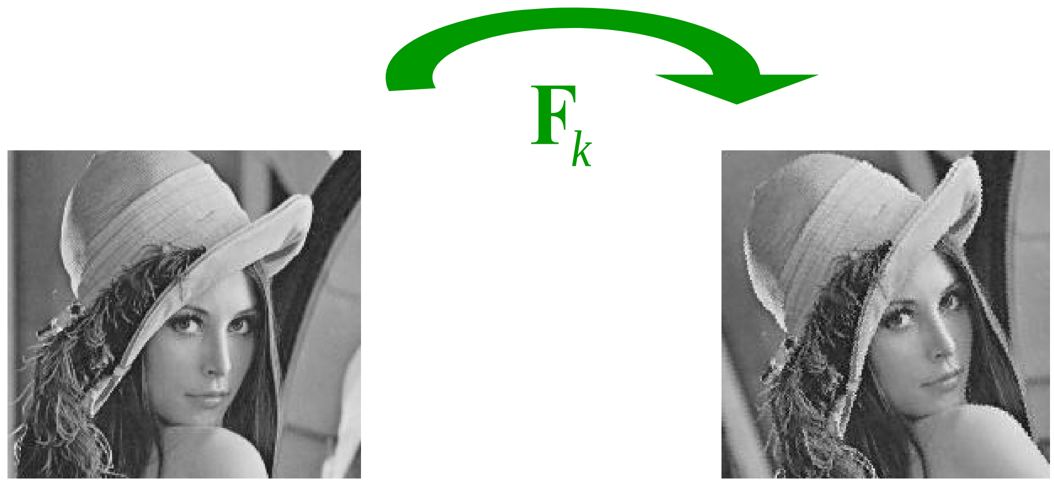


Find the high resolution image $\underline{X}$, which minimizes the Error:

$$E(\underline{X}) = \sum_{i=1}^{N} \left\| P_i(\underline{X}) - y[i] \right\|_2^2$$

Where $P_i(\underline{X})$ is the simulated projection of image $\underline{X}$ onto the grid of image *y*[*i*].

Geometric Transformation



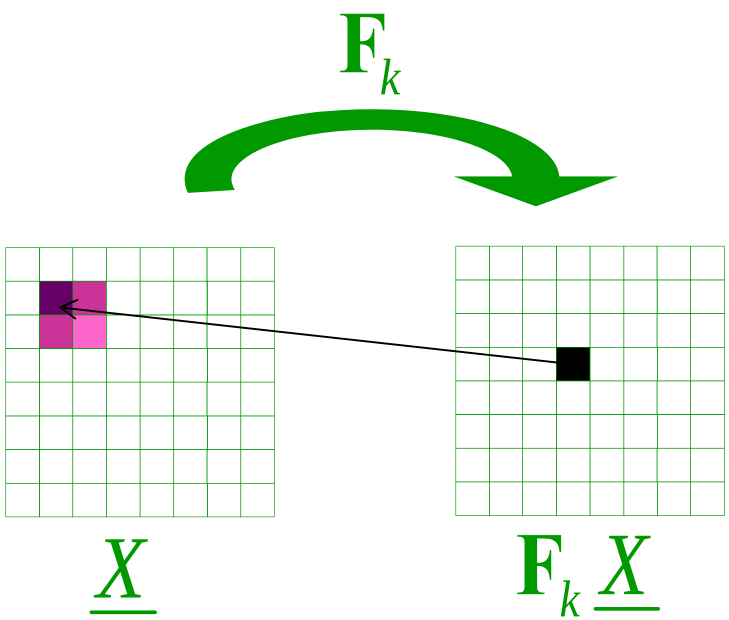
Scene

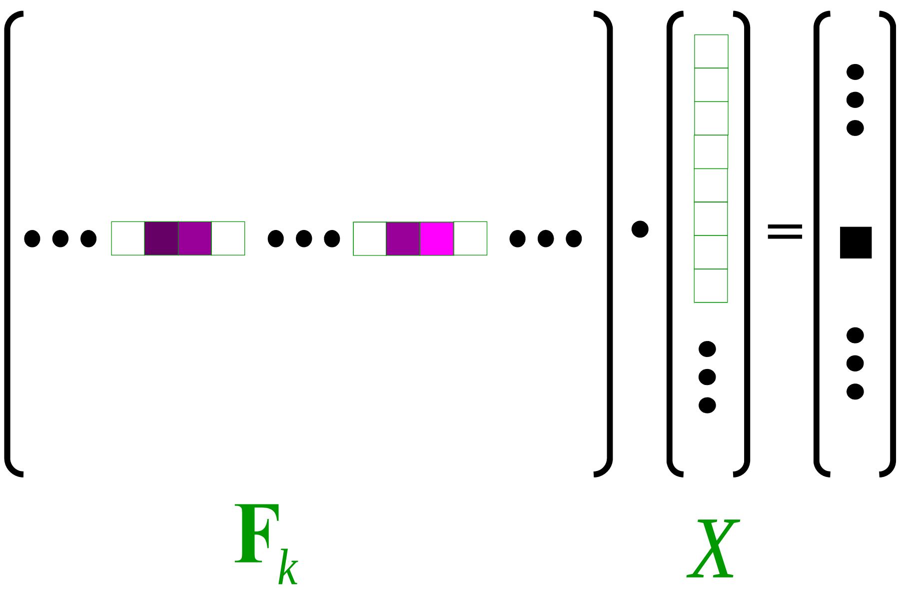
Geometric transformation

- Any appropriate motion model
- Every frame has different transformation
- Usually found by a separate registration algorithm

Geometric Transformation

Can be modeled as a <u>linear</u> operation $\mathbf{F}_k \underline{X}$

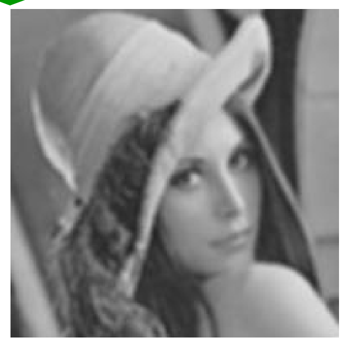




Translation can even be modeled as a <u>convolution</u>

Blur

$\mathbf{H}_k$

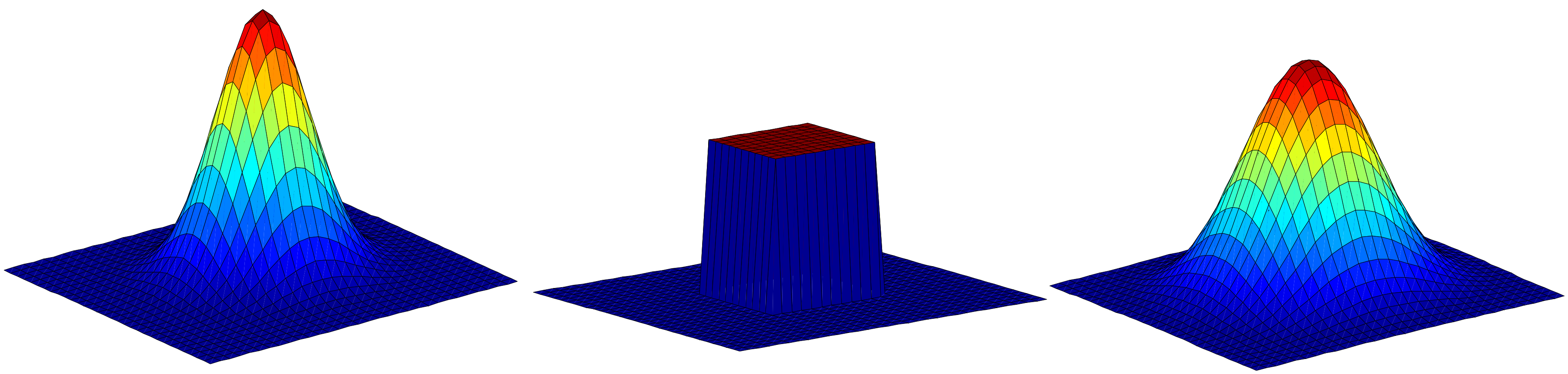


Geometric transformation

Blur

- Due to the lens PSF and pixel integration
- Usually $\mathbf{H}_k = \mathbf{H}$: All images have same blur

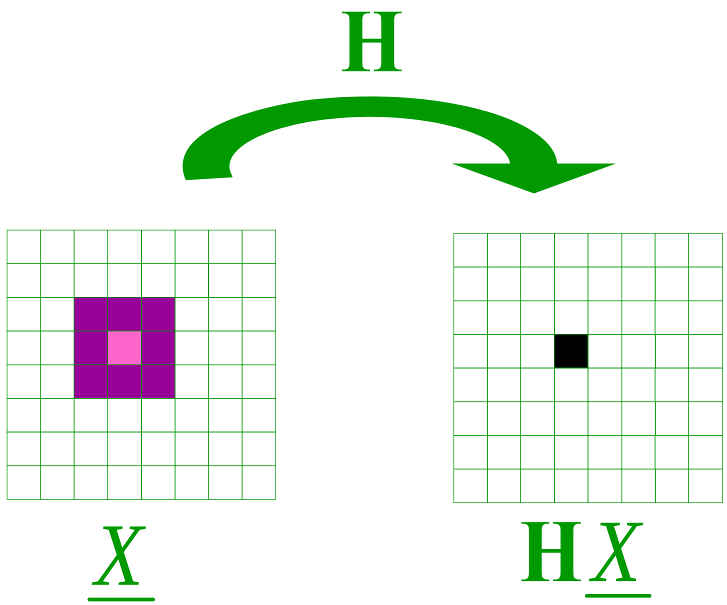
H

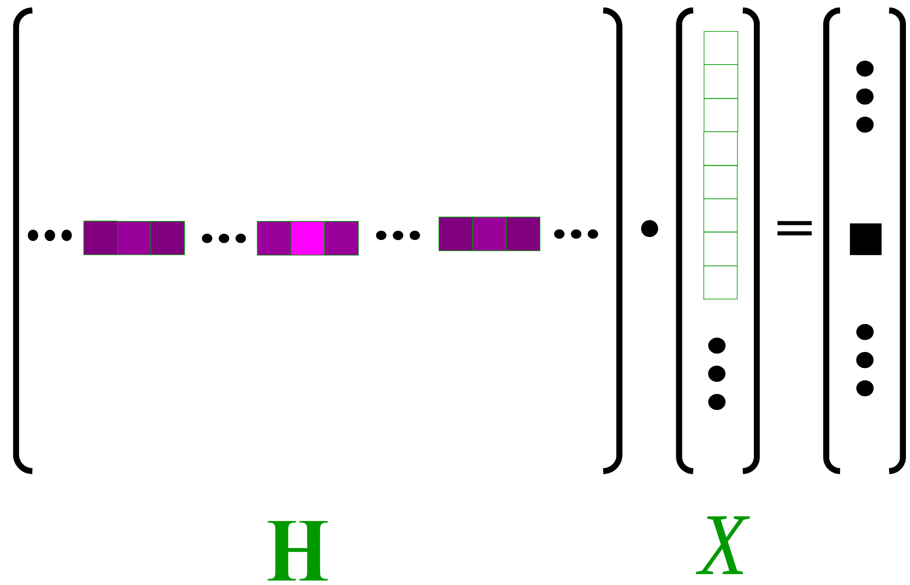


Optical PSF * Sensor Area = H

Blur

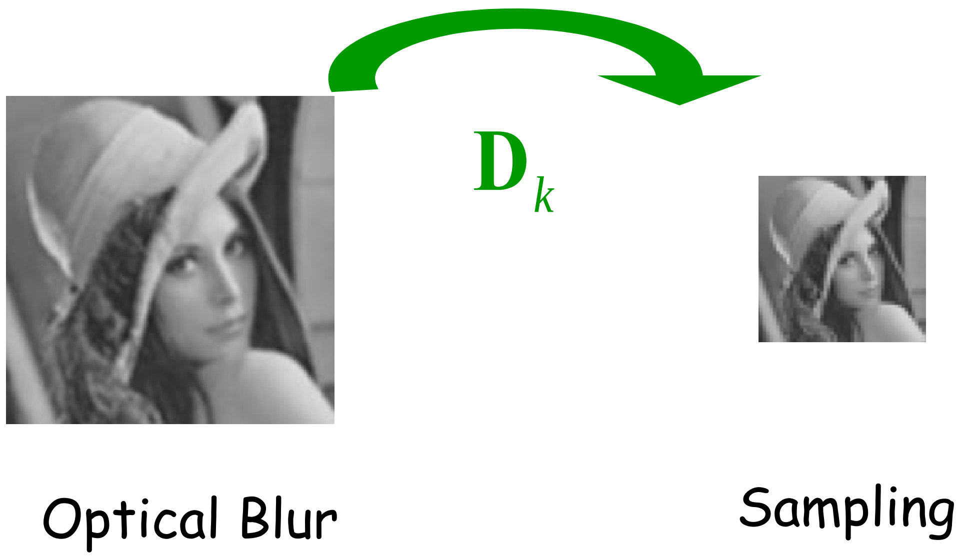
Can be modeled as a linear operation $\mathbf{H}\underline{X}$





Uniform blur can be modeled as a <u>convolution</u>

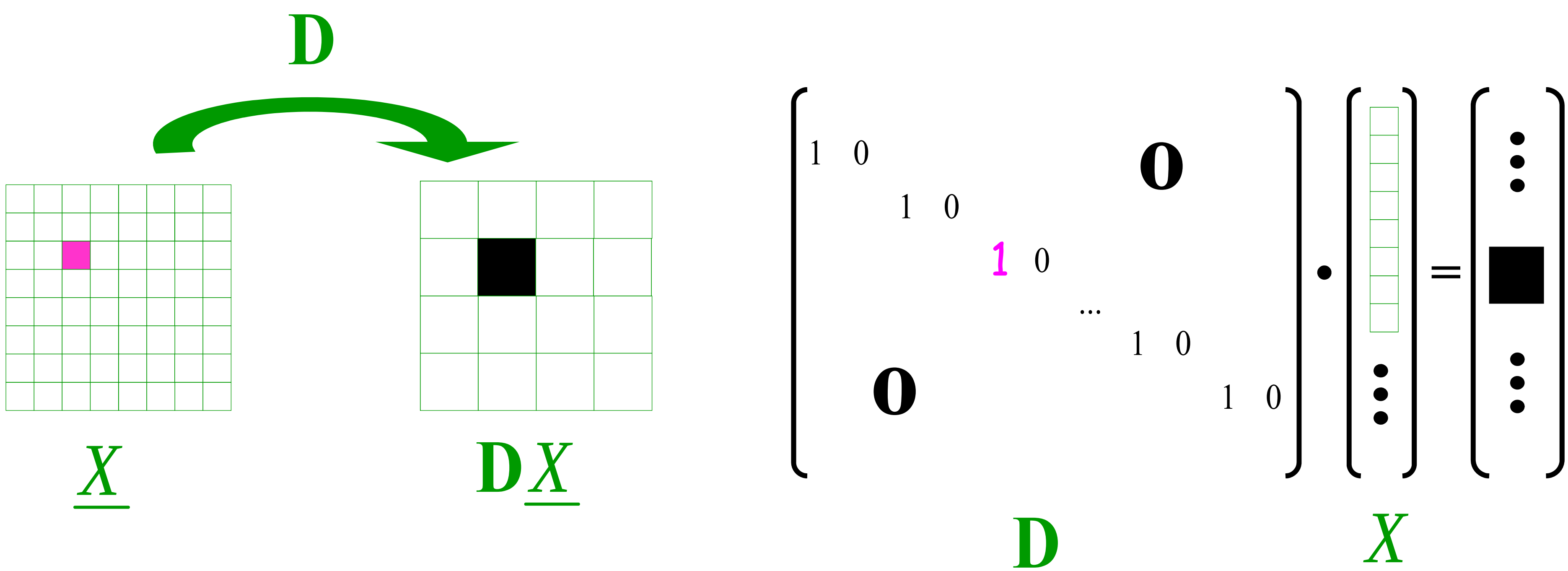
Sampling



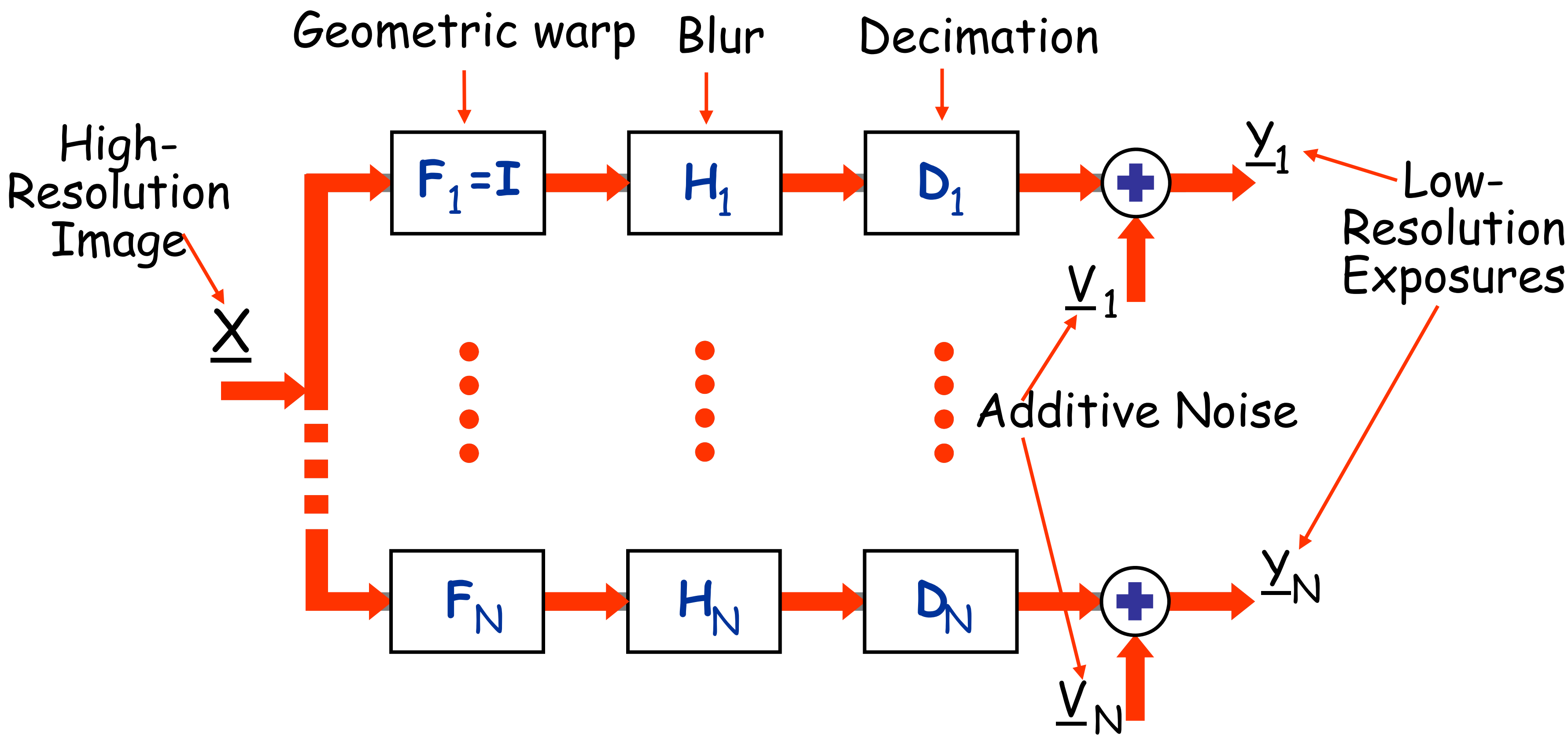
- Pixel operation consists of area integration followed by decimation
- D is the decimation only
- Usually $\mathbf{D}_k = \mathbf{D}$

Sampling / Decimation

Can be modeled as a linear operation $\mathbf{D}\underline{X}$

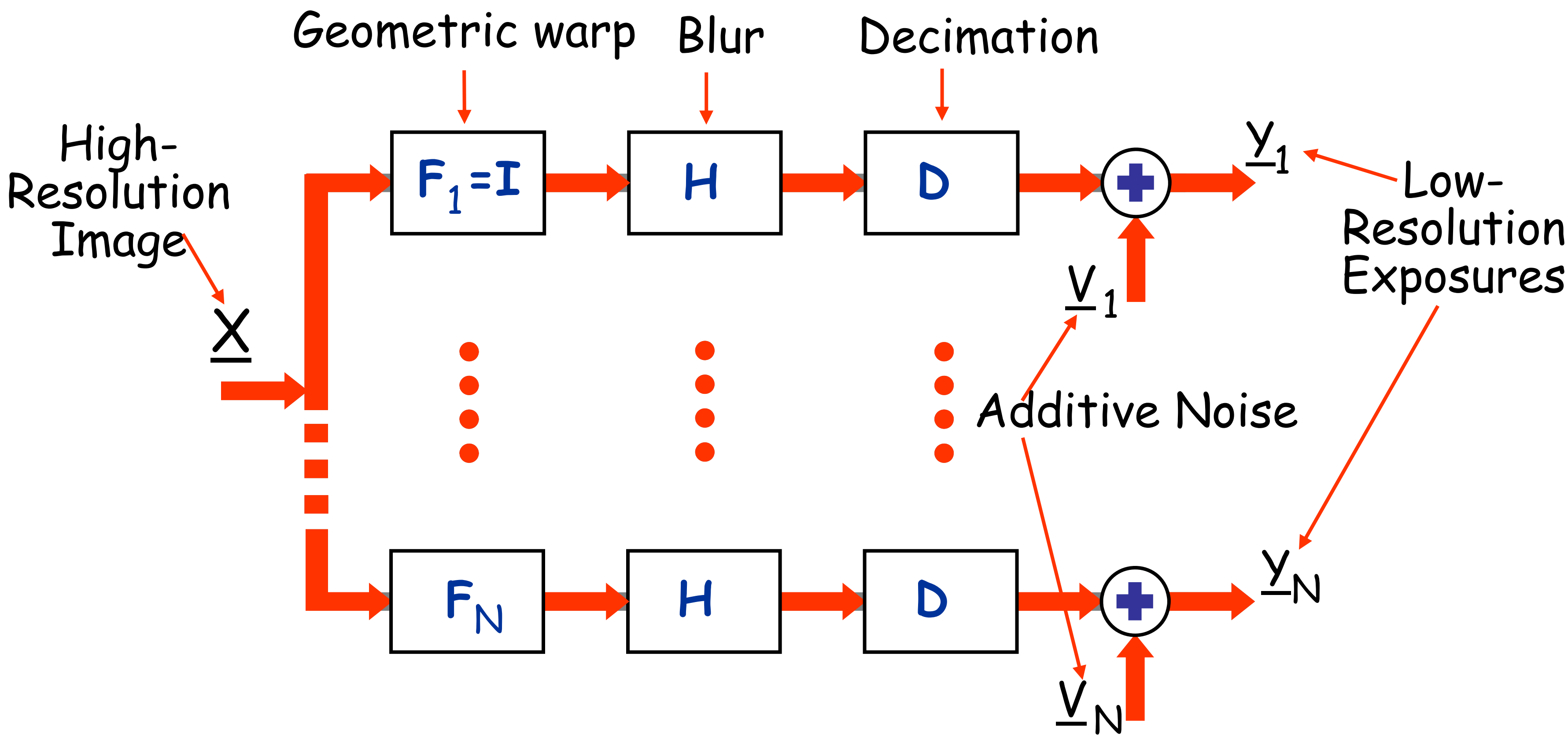


Super-Resolution - Model



$$\left\{ \underline{Y}_k = \mathbf{D}_k \mathbf{H}_k \mathbf{F}_k \underline{X} + \underline{V}_k, \quad \underline{V}_k \sim \mathbf{N}\left\{0, \sigma_n^2\right\} \right\}_{k=1}^{N}$$

Simplified Model



$$\left\{ \underline{Y}_k = \mathbf{D}\ \mathbf{H}\ \mathbf{F}_k \underline{X} + \underline{V}_k,\quad \underline{V}_k \sim \mathbf{N}\left\{0, \sigma_n^2\right\} \right\}_{k=1}^{N}$$

The Super-Resolution Problem

$$\underline{Y}_k = \mathbf{DHF}_k \underline{X} + \underline{V}_k, \quad \underline{V}_k \sim \mathbf{N}\left\{0, \sigma_n^2\right\}$$

- Given

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

  H – The blur can be extracted from the camera characteristics

  D – The decimation is dictated by the required resolution ratio

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

  $\sigma_{\mathbf{n}}$ – The noise can be extracted from the camera / image

- Recover

  $\underline{X}$ – HR image

The Model as One Equation

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

r = resolution factor = 4
M×M = size of the frames = 1000×1000
N = number of frames = 10

$\underline{Y}$ of size $[NM^2 \times 1]$ =[10Meg×1]
$\mathbf{G}$ of size $[NM^2 \times r^2M^2]$ =[10Meg×16Meg]
$\underline{X}, \underline{V}$ of size $[r^2M^2 \times 1]$ =[16Meg×1]

Linear algebra notation is intended only to develop algorithm

SR - Solutions

- Maximum Likelihood (ML)
  - Find X giving highest probability to y_k

$$\underline{X} = \arg\min_{\underline{X}} \sum_{k=1}^{N} \left\| \mathbf{DHF}_k \underline{X} - \underline{Y}_k \right\|^2$$

Often ill posed problem!

- Maximum Aposteriori Probability (MAP)
  - Find the most likely X given y_k

$$\underline{X} = \arg\min_{\underline{X}} \sum_{k=1}^{N} \left\| \mathbf{DHF}_k \underline{X} - \underline{Y}_k \right\|^2 + \lambda A\{\underline{X}\}$$

Smoothness constraint regularization

ML Reconstruction (LS)

Minimize:
$$\varepsilon_{ML}^2\left(\underline{X}\right)=\sum_{k=1}^{N}\left\| \mathbf{DHF}_k\,\underline{X}-\underline{Y}_k\right\|^2$$

Thus, require:
$$\frac{\partial\,\varepsilon_{ML}^2\left(\underline{X}\right)}{\partial\,\underline{X}}=2\sum_{k=1}^{N}\mathbf{F}_k^T\mathbf{H}^T\mathbf{D}^T\left(\mathbf{DHF}_k\,\underline{\hat{X}}-\underline{Y}_k\right)=0$$

$$\Rightarrow\quad \sum_{k=1}^{N}\underbrace{\mathbf{F}_k^T\mathbf{H}^T\mathbf{D}^T}_{\mathrm{A}^{\mathrm{T}}}\underbrace{\mathbf{DHF}_k}_{\mathbf{A}}\cdot\underline{\hat{X}}=\sum_{k=1}^{N}\underbrace{\mathbf{F}_k^T\mathbf{H}^T\mathbf{D}^T}_{\mathrm{A}^{\mathrm{T}}}\underline{Y}_k$$

$$\mathrm{A}^T\mathrm{A}\,\underline{\hat{X}}=\mathrm{A}^T\,\underline{Y}$$

LS - Iterative Solution

- Steepest descent

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n - \beta \sum_{k=1}^{N} \underbrace{\mathbf{F}_k^T \mathbf{H}^T \mathbf{D}^T}_{\substack{\text{Back projection} \\ A^T}} \underbrace{\left(\mathbf{D}\mathbf{H}\mathbf{F}_k \underline{\hat{X}}_n - \underline{Y}_k \right)}_{\substack{\text{Simulated} \\ \text{error}}}$$

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

There is no actual need to use the Matrix-Vector notations as shown here.

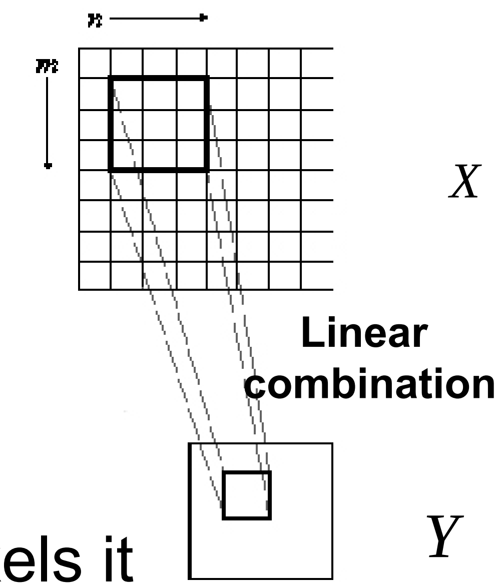
Transpose – Back Projection

•If X is a vector representing f, and $Y=AX$ represents the image g, created by the operator A. (warp, blur, etc.) how to implement the operator A^T?

•Y_i represent the *(x,y)* pixel in g, is derived by linear combination of the values of the pixels X_j in the image f:

$$y_i = \sum_j A_{ij} X_j$$

and $A_{ij} \neq 0$ if pixel i has influence on pixel j.



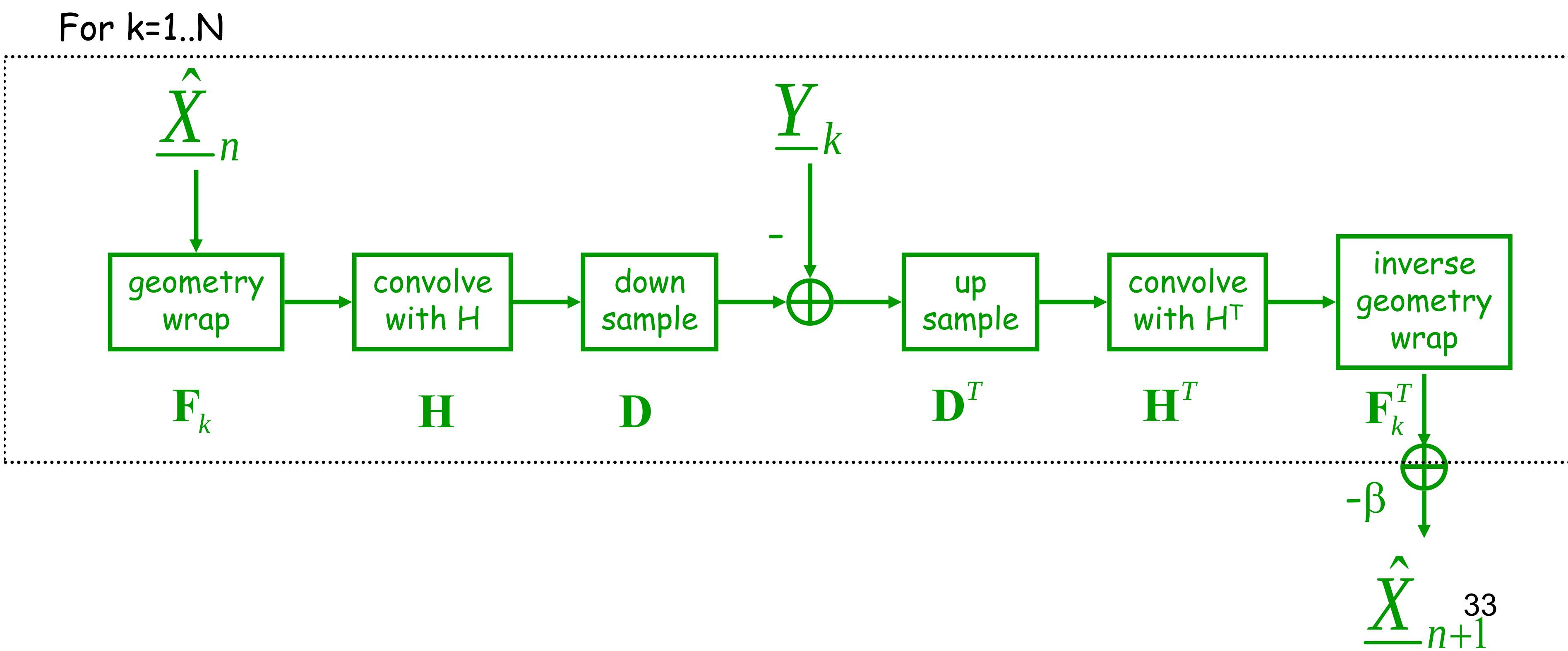
•Therefore. Multiplying with A^T is simply:

$$X_i = \sum_j A_{ij} Y_i$$

The pixel in f is a linear combination of the pixels it influenced.

LS - Iterative Solution

- Steepest descent $$\hat{\underline{X}}_{n+1} = \hat{\underline{X}}_n - \beta\sum_{k=1}^{N}\mathbf{F}_k^T\mathbf{H}^T\mathbf{D}^T\left(\mathbf{D}\mathbf{H}\mathbf{F}_k\hat{\underline{X}}_n - \underline{Y}_k\right)$$



oise

u at al.
sing, 04

Robust Reconstruction

- Cases of measurements outlier:
  - Some of the images are irrelevant
  - Error in motion estimation
  - Error in the blur function
  - General model mismatch

Robust Reconstruction

$$b_k(\underline{X}) = \left(\mathbf{DHF}_k \underline{X} - \underline{Y}_k \right)$$

Minimize 1: $$\varepsilon^2(\underline{X}) = \sum_{k=1}^{N} \left\| b_k(\underline{X}) \right\|^2$$

Minimize 2: $$\varepsilon^1(\underline{X}) = \sum_{k=1}^{N} \left\| b_k(\underline{X}) \right\|$$

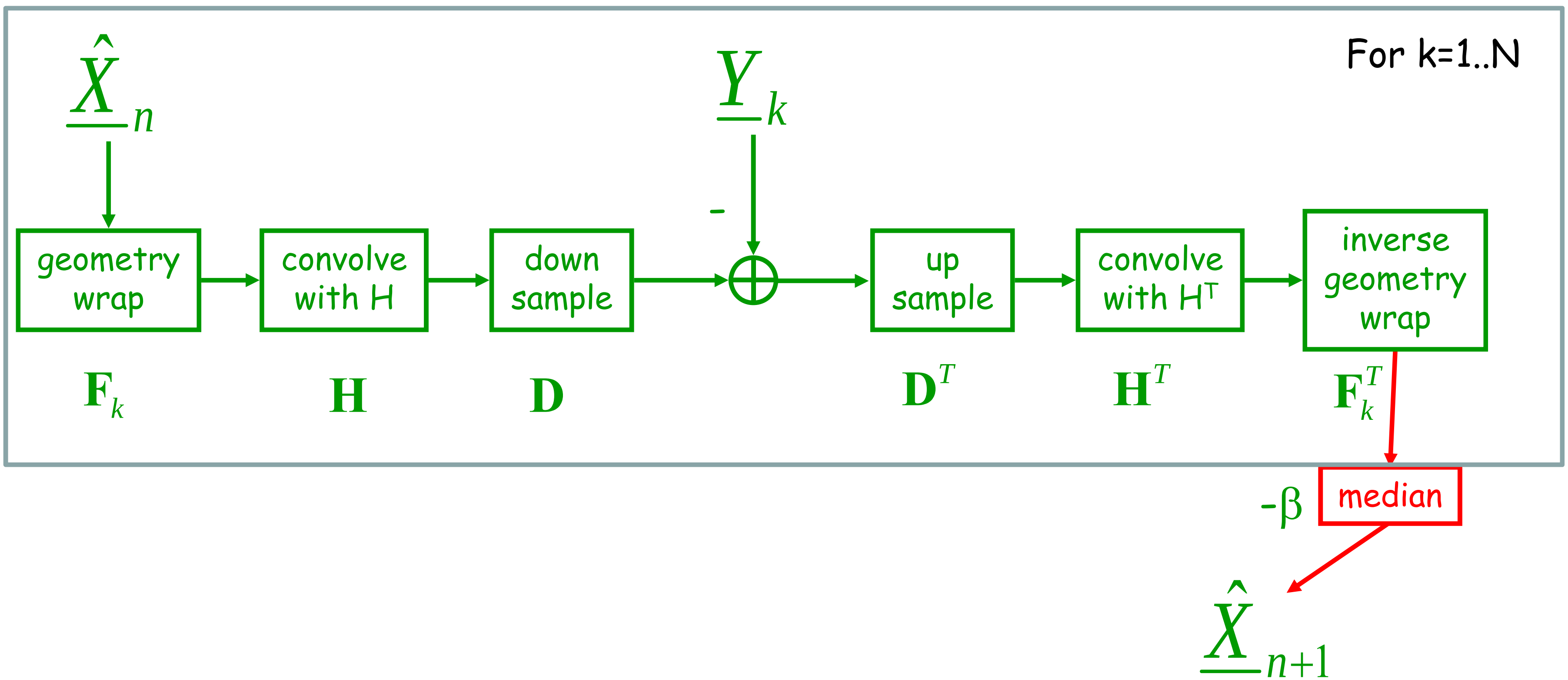
$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n - \beta \cdot Median\left\{ b_k(\underline{X}) \right\}$$

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n - \beta \cdot \sum Sign\left\{ b_k(\underline{X}) \right\}$$

Robust Reconstruction

- Steepest descent

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n - \beta \cdot Median\left\{b_k\left(\underline{X}\right)\right\}$$



Example - Outliers

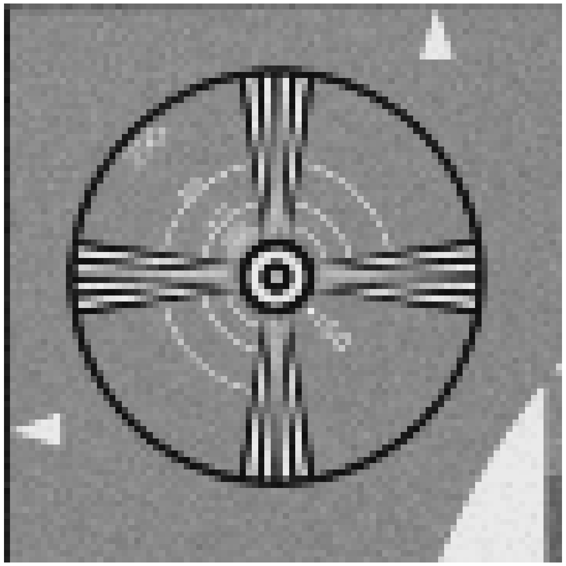
HR image

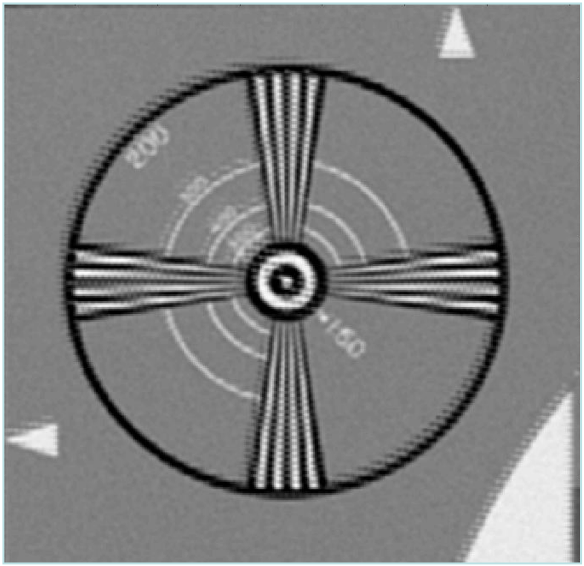
LR + noise
X4

Least squares

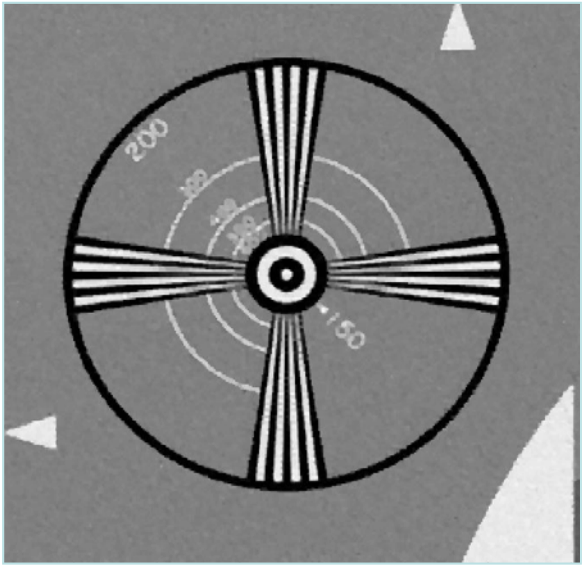
Robust Reconstruction

Example – Registration Error





L_2 norm based



L_1 norm based

20 images, ratio 1:4

MAP Reconstruction

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

- Regularization term:
  - Tikhonov cost function

$$A_T\left\{\underline{X}\right\} = \left\|\Gamma \underline{X}\right\|^2$$

  - Total variation

$$A_{TV}\left\{\underline{X}\right\} = \left\|\nabla \underline{X}\right\|_1$$

  - Bilateral filter

$$A_B\left\{\underline{X}\right\} = \sum_{l=-P}^{P} \sum_{m=-P}^{P} \alpha^{|l|+|m|} \left\| \underline{X} - S_x^l S_y^m \underline{X} \right\|_1$$

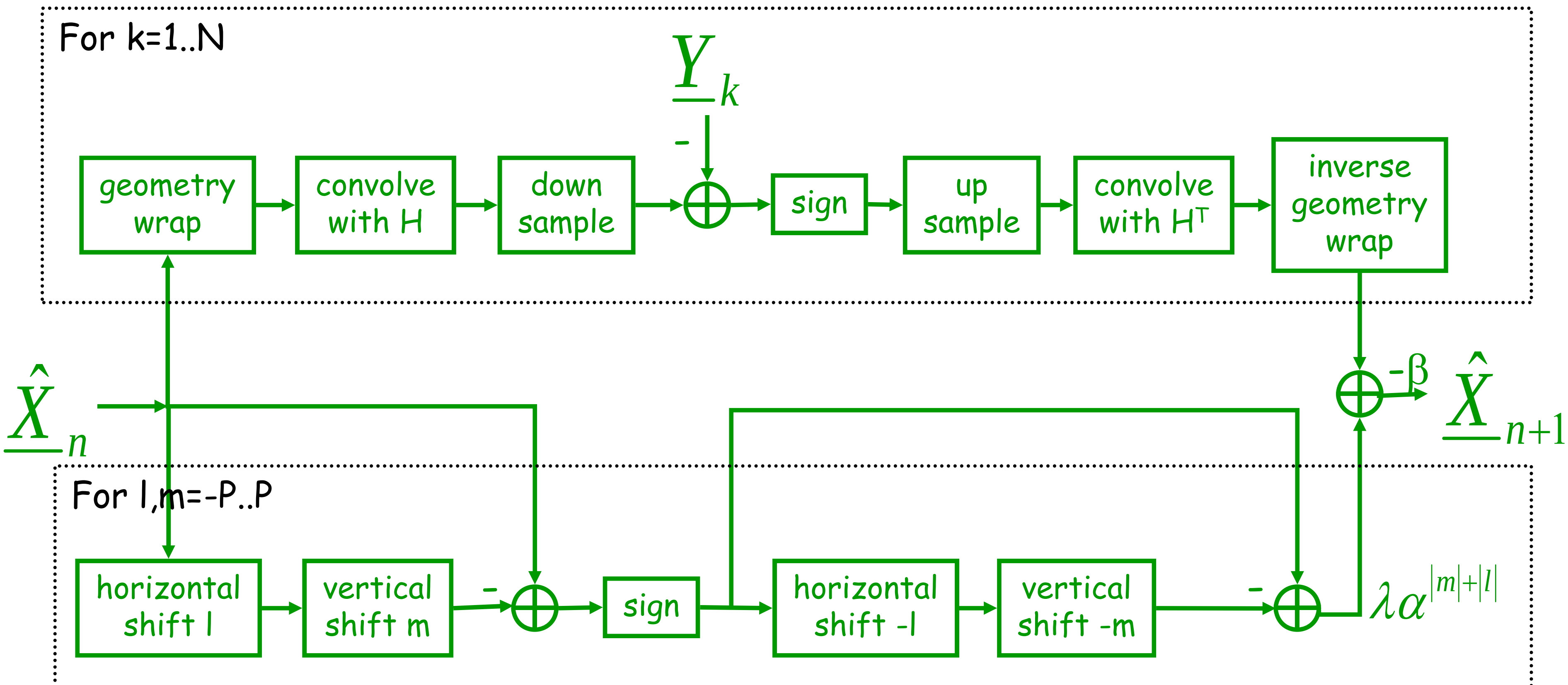
Robust Estimation + Regularization

Minimize: $$\varepsilon^2(\underline{X}) = \sum_{k=1}^{N} \left\| \mathbf{DHF}_k \underline{X} - \underline{Y}_k \right\|_1 + \lambda \sum_{l=-P}^{P} \sum_{m=-P}^{P} \alpha^{|l|+|m|} \left\| \underline{X} - S_x^l S_y^m \underline{X} \right\|_1$$

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n - \beta \left\{ \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}^T \mathbf{D}^T \operatorname{sign}\left(\mathbf{DHF}_k \underline{\hat{X}}_n - \underline{Y}_k \right) + \lambda \sum_{l=-P}^{P} \sum_{m=-P}^{P} \alpha^{|l|+|m|} \left[I - S_x^{-l} S_y^{-m} \right] \operatorname{sign}\left(\underline{\hat{X}}_n - S_x^l S_y^m \underline{\hat{X}}_n \right) \right\}$$

Robust Estimation + Regularization

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n - \beta\left\{\sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{H}^T \mathbf{D}^T \operatorname{sign}\left(\mathbf{D}\mathbf{H}\mathbf{F}_k \underline{\hat{X}}_n - \underline{Y}_k\right) + \lambda \sum_{l=-P}^{P} \sum_{m=-P}^{P} \alpha^{|l|+|m|}\left[I - S_x^{-l} S_y^{-m}\right] \operatorname{sign}\left(\underline{\hat{X}}_n - S_x^l S_y^m \underline{\hat{X}}_n\right)\right\}$$



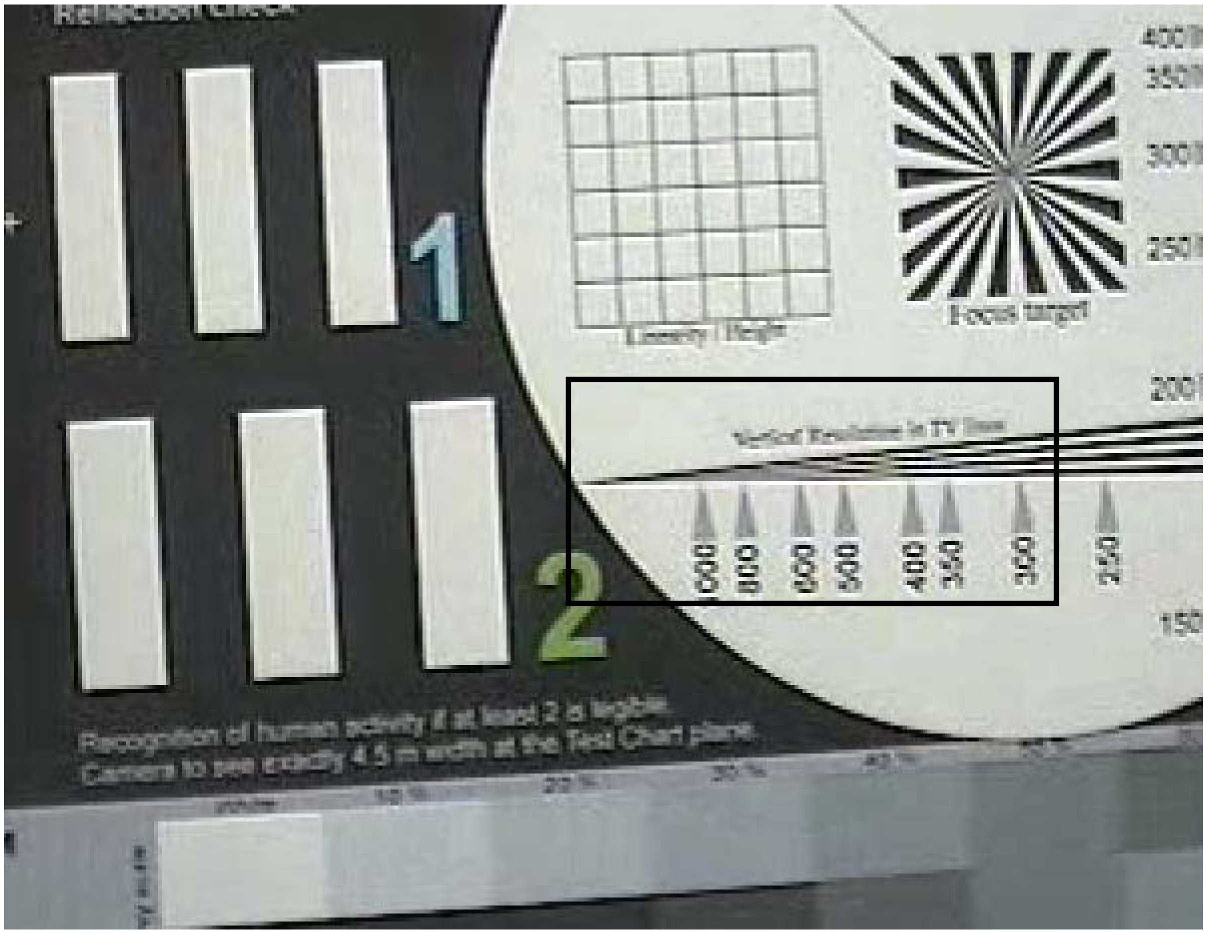
From Farisu at al. IEEE trans. On Image Processing, 04

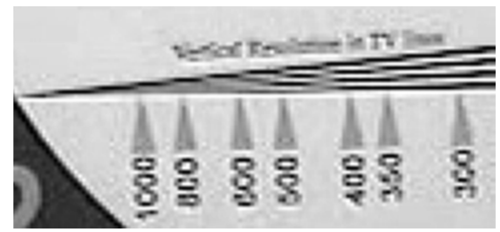
Example

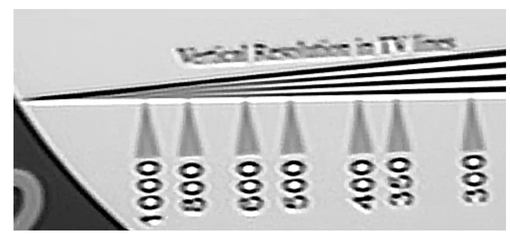
- 8 frames
- Resolution factor of 4

From Farisu at al. IEEE trans. On Image Processing, 04

Example







Images from Vigilant Ltd.

Handling Color in SR

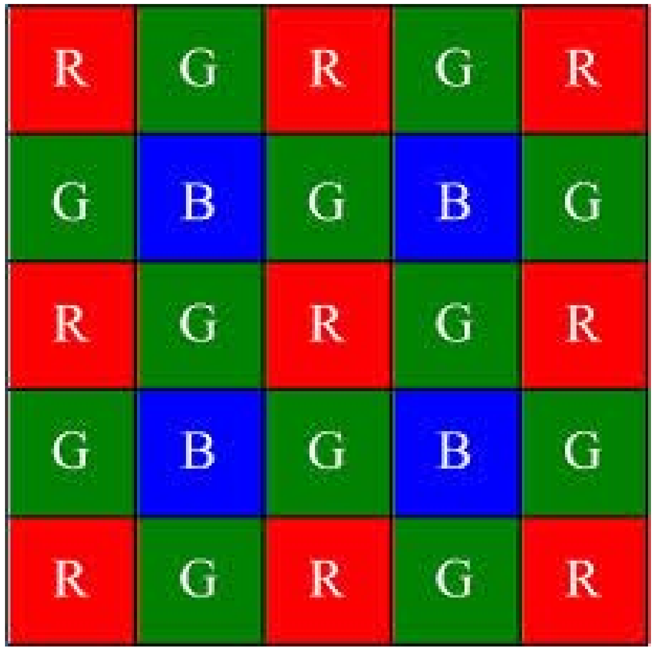
$$\varepsilon_{MAP}^{2}\left(\underline{X}\right)=\sum_{k=1}^{N}\left\| \mathbf{DHF}_k \underline{X - \underline{Y}_k} \right\|^2 + \lambda A\left\{\underline{X}\right\}$$

Handling color: the classic approach is to convert the measurements to YCbCr, apply the SR on the Y and use trivial interpolation on the Cb and Cr.

Better treatment can be obtained if the statistical dependencies between the color layers are taken into account (i.e. forming a prior for color images).

In case of mosaiced measurements, demosaicing followed by SR is sub-optimal. An algorithm that directly fuse the mosaic information to the SR is better.

Color Sensor: Bayer Pattern

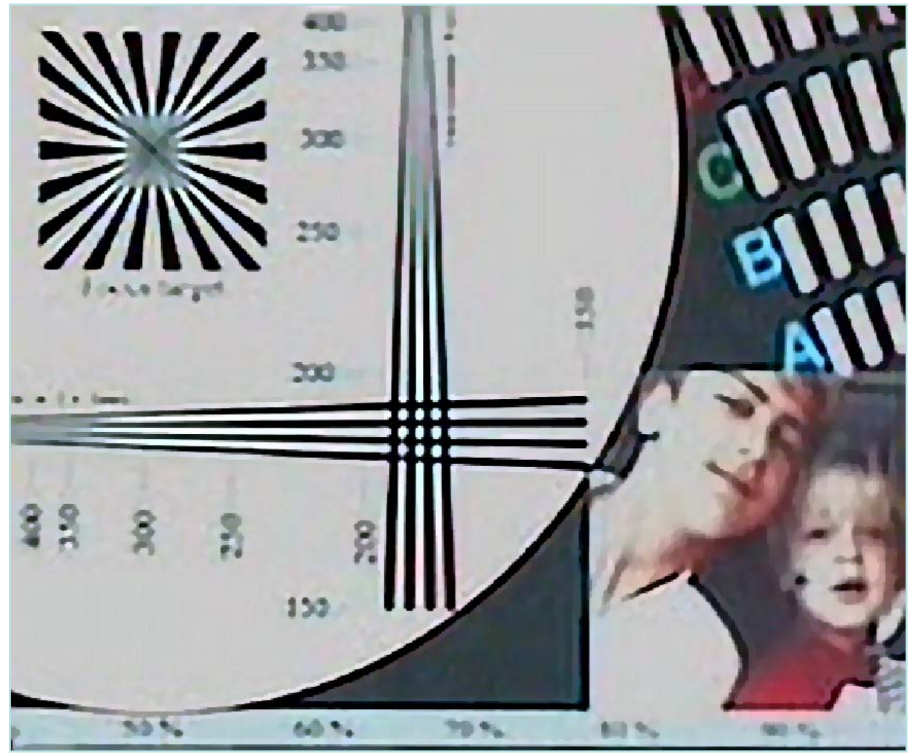


In each pixel only one color component is measured.

Two color component are “hallucinated” (demosaicking)

SR for Full Color

20 images, ratio 1:4



SR+Demosaicing

20 images, ratio 1:4

Mosaiced input

Mosaicing and then SR

Combined treatment

Super-Resolution - Agenda

- The basic idea
- Image formation process
- Formulation and solution
- **Special cases and related problems**
- Limitations of Super-Resolution
- SR in time

Special Case – Translational Motion

- In this case H and F commute:

$$\mathbf{H}\mathbf{F}_k = \mathbf{F}_k\mathbf{H} \qquad \mathbf{H}^T\mathbf{F}_k^T = \mathbf{F}_k^T\mathbf{H}^T$$

$$\begin{aligned} \underline{Y}_k &= \mathbf{D}\mathbf{H}\mathbf{F}_k\underline{X} + \underline{V}_k \\ &= \mathbf{D}\mathbf{F}_k\mathbf{H}\underline{X} + \underline{V}_k \\ &= \mathbf{D}\mathbf{F}_k\underline{Z} + \underline{V}_k \qquad \underline{Z} = \mathbf{H}\underline{X} \end{aligned}$$

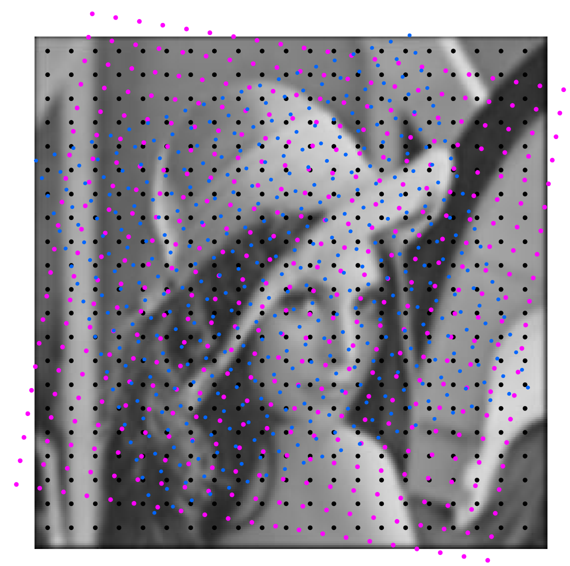
- SR is decomposed into 2 steps
  1. Find blur HR image from LR images ➔ non-iterative
  2. Deconvolve the result using H ➔ iterative

Intuition

$$\underline{Y}_k = \mathbf{D}\mathbf{F}_k \underline{Z} + \underline{V}_k \qquad Z = \mathbf{H}\underline{X}$$

X

PSF*X



Z=PIXEL*PSF*X

- Using the samples can, at most, reconstruct Z
- To recover X, need to deconvolve Z

Step I – Find Blurred HR

Minimize: $$\varepsilon_{ML}^{2}\left(\underline{Z}\right)=\sum_{k=1}^{N}\left\| \mathbf{DF}_k\,\underline{Z}-\underline{Y}_k\right\|^{2}$$

- L_2 ➔ For all frames, copy registered pixels to HR grid and average [Elad & Hel-Or, 01]
- L_1 ➔ For all frames, copy registered pixels to HR grid and use median [Farisu, 04]

Solution for L$_2$

Minimize: $$\varepsilon_{ML}^2(\underline{Z}) = \sum_{k=1}^{N} \| \mathbf{D}\mathbf{F}_k \underline{Z} - \underline{Y}_k \|^2$$

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

$$\Rightarrow \quad \mathbf{R}\hat{\underline{Z}} = \underline{P} \qquad \left\{ \begin{array}{l} \mathbf{R} = \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{D}^T \mathbf{D} \mathbf{F}_k \\ \underline{P} = \sum_{k=1}^{N} \mathbf{F}_k^T \mathbf{D}^T \underline{Y}_k \end{array} \right\}$$

Diagonal, number Of occurrences per HR grid

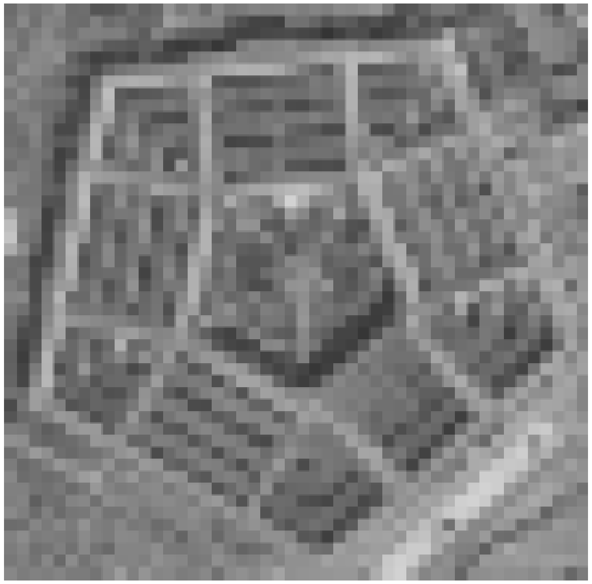
Sum of HR grid

Step II - Deblur

Minimize: $$\varepsilon^2\left(\underline{X}\right) = \left\| \mathbf{H}\underline{X} - \underline{Z} \right\|_1 + \lambda A\left\{\underline{X}\right\}$$

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n - \beta\left\{\mathbf{H}^T \operatorname{sign}\left(\mathbf{H}\underline{\hat{X}}_n - \underline{Z}_k\right) + \lambda \frac{\partial}{\partial \underline{\hat{X}}_n} A\left\{\underline{\hat{X}}_n\right\}\right\}$$

Example



64X64 LR

256X256
Before deblur

256X256
After deblur

From Pham at al. Proc. Of SPIE-IS&T, 05. Simulated.

Related Problems

- Denoising (multiple frames)

$$\underline{Y}_k = \underline{X} + \underline{V}_k, \quad \underline{V}_k \sim \mathbf{N}\left\{0, \sigma_n^2\right\}$$

- Denoising (single frame)

$$\underline{Y} = \underline{X} + \underline{V}, \quad \underline{V} \sim \mathbf{N}\left\{0, \sigma_n^2\right\}$$

- Deblurring

$$\underline{Y} = \mathbf{H}\underline{X} + \underline{V}, \quad \underline{V} \sim \mathbf{N}\left\{0, \sigma_n^2\right\}$$

- Interpolation – "single-image super-resolution"

$$\underline{Y} = \mathbf{DH}\underline{X} + \underline{V}, \quad \underline{V} \sim \mathbf{N}\left\{0, \sigma_n^2\right\}$$

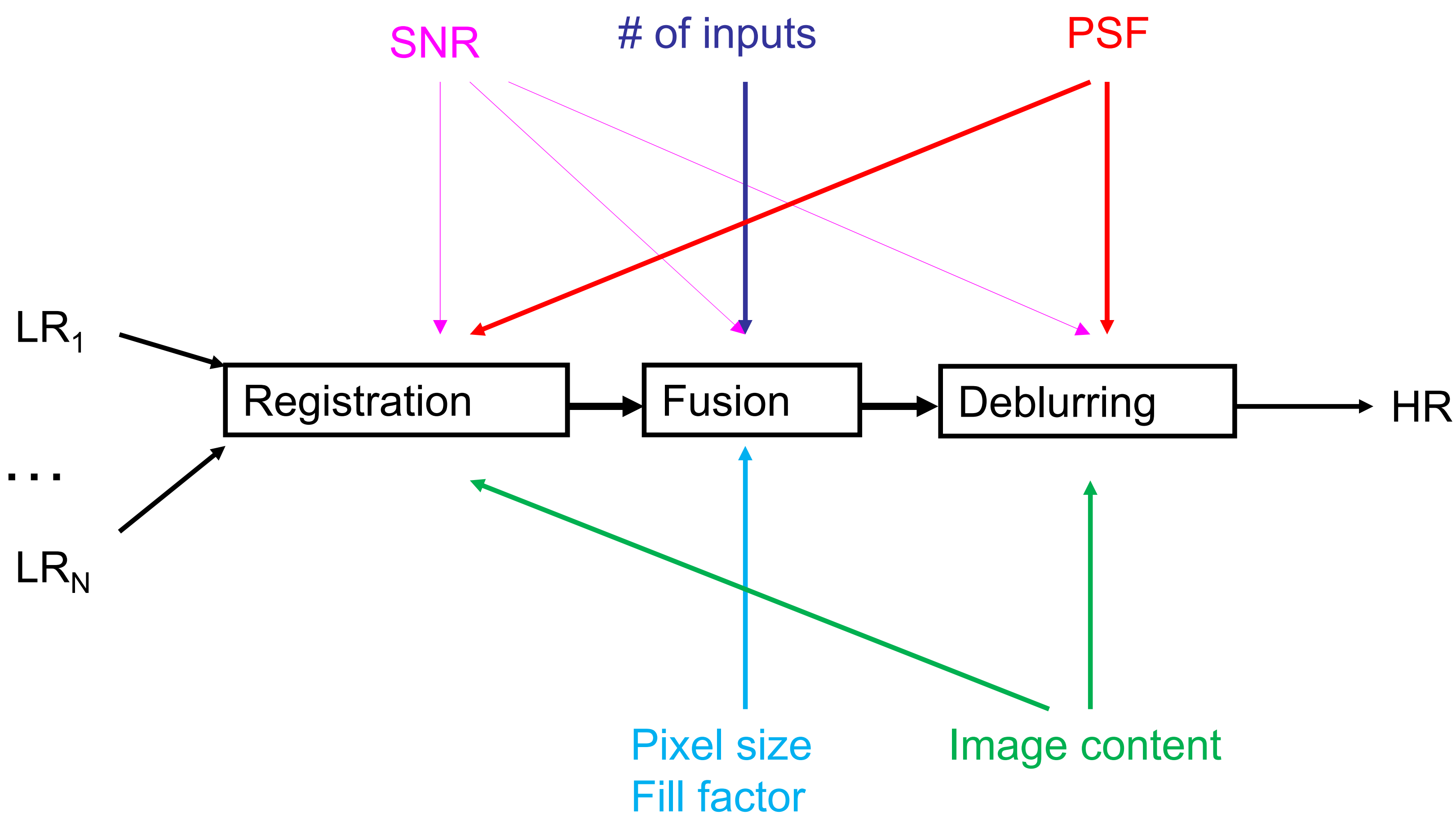
Super-Resolution - Agenda

- The basic idea
- Image formation process
- Formulation and solution
- Special cases and related problems
- **Limitations of Super-Resolution**
- SR in time

Limiting Factors

- Main factors
  - SNR
  - PSF (optical+pixel)
  - Number of inputs
  - Pixel size (sampling rate)
  - Fill factor
  - Image content

Limiting Factors



SR Limits Analysis

- Noise
  - Registration noise
  - Fusion noise

- SR factor
  - Point-Spread-Function (PSF)
    - Optical Transfer Function (OTF)
  - Sensor pixel size
    - Sensor Transfer Function (STF)

Registration Noise

- Using Cramer Rao Lower Bound (CRLB)
  - Lower bounds for shift estimation:

$$\mathrm{var}(u) \geq \sigma_n^2 \frac{\sum \mathbf{I}_y^2}{\sum \mathbf{I}_x^2 \sum \mathbf{I}_y^2 - \left(\sum \mathbf{I}_x \mathbf{I}_y\right)^2}$$

$$\mathrm{var}(v) \geq \sigma_n^2 \frac{\sum \mathbf{I}_x^2}{\sum \mathbf{I}_x^2 \sum \mathbf{I}_y^2 - \left(\sum \mathbf{I}_x \mathbf{I}_y\right)^2}$$

$$\Rightarrow \sigma_{reg}^2 \approx \sigma_n^2 \frac{\sum \mathbf{I}_x^2 + \sum \mathbf{I}_y^2}{\sum \mathbf{I}_x^2 \sum \mathbf{I}_y^2 - \left(\sum \mathbf{I}_x \mathbf{I}_y\right)^2}$$

- Better registration accuracy by:
  - Less noise
  - Higher derivatives in image
  - Bigger registration area
  - Narrow PSF

Fusion Noise

$$\sigma^2_{fusion} \approx \frac{r^2}{N}\sigma^2_n$$

- r = super-resolution factor
- N = number of images

Registration + Fusion Noise

$$\sigma_{total}^2 \approx \frac{\sum \mathbf{I}_x^2 + \sum \mathbf{I}_y^2}{\sum \mathbf{I}_x^2 \sum \mathbf{I}_y^2 - \left(\sum \mathbf{I}_x \mathbf{I}_y\right)^2} \sigma_n^2 + \frac{r^2}{N} \sigma_n^2$$

- If N $\rightarrow\infty$ then
  - Fusion error vanishes
  - Registration error is equivalent to Gaussian blur

Super-Resolution - Agenda

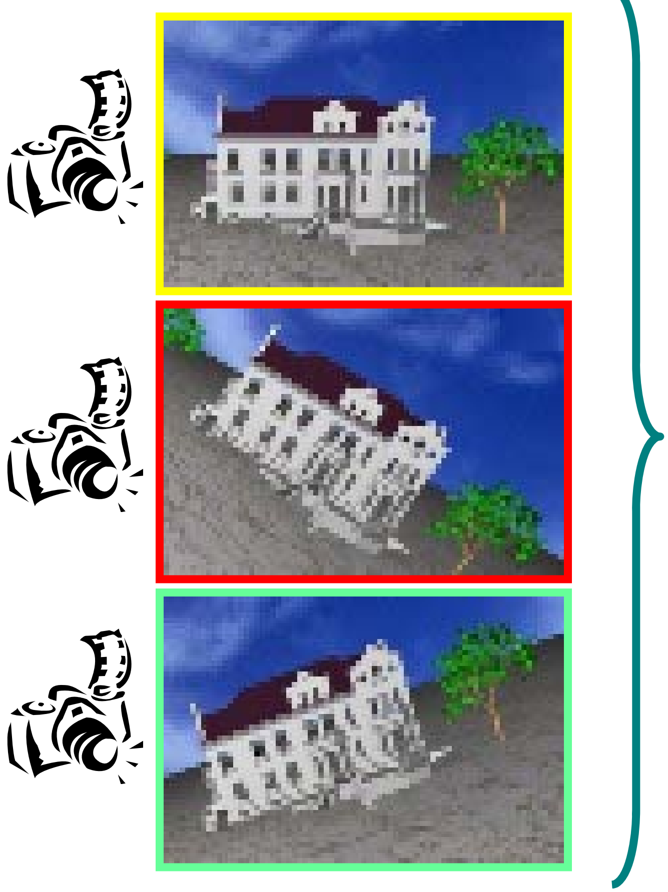
- The basic idea
- Image formation process
- Formulation and solution
- Special cases and related problems
- Limitations of Super-Resolution
- **SR in time**

Work and slides by Michal Irani & Yaron Caspi (ECCV'02)

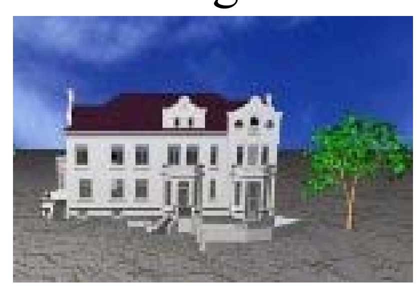
“Classical” Image Super-Resolution

Scene:

Low-resolution images:

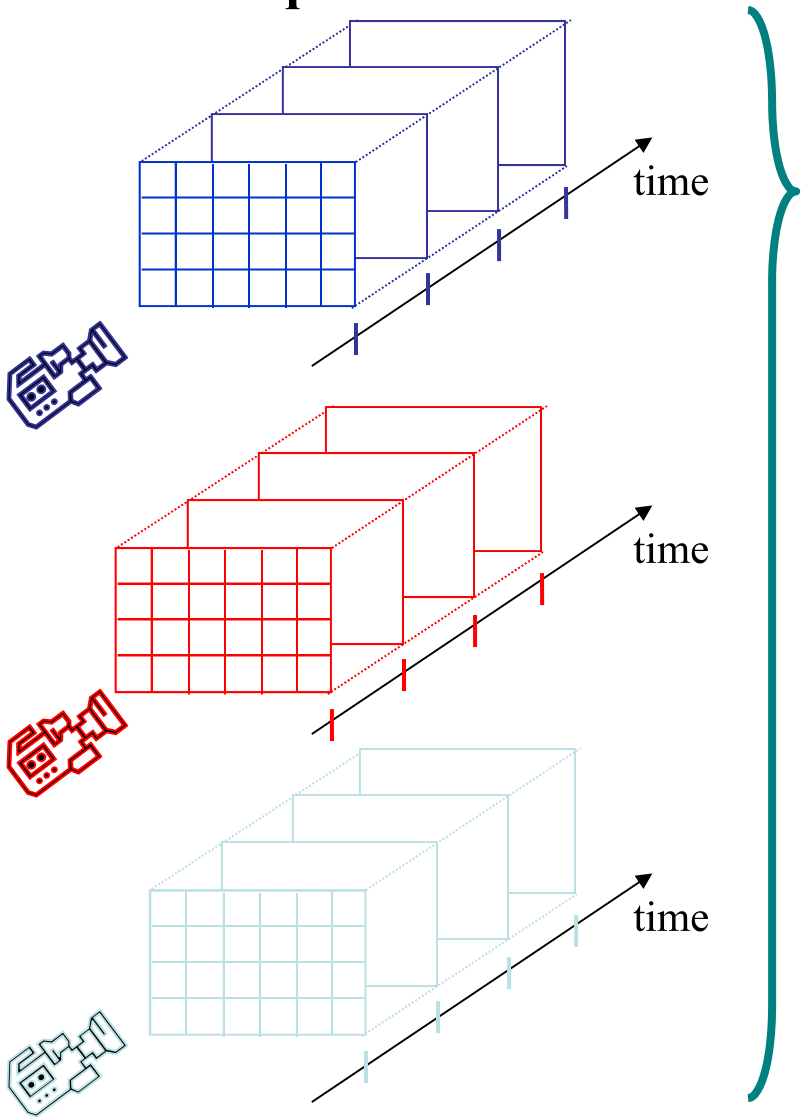


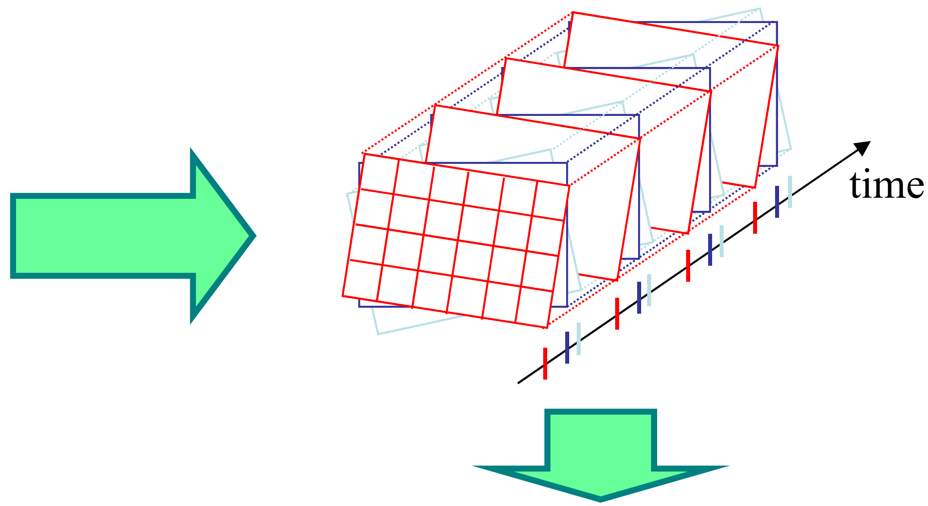
High-resolution image:



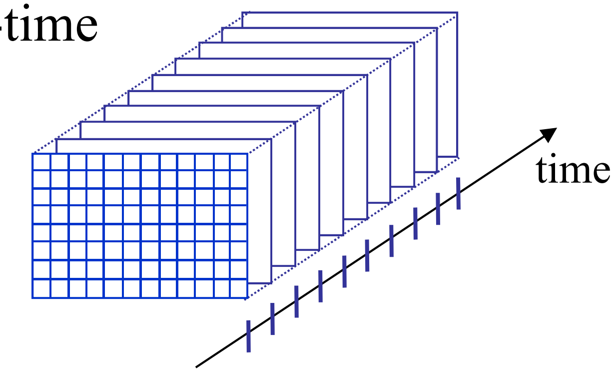
Space-Time Super-Resolution

Low-resolution ~~images~~ video **sequences**:





High space-time resolution sequence:



Super-resolution in **space** and in **time**.

What is Super-Resolution in Time?

Observing events “faster” than frame-rate.

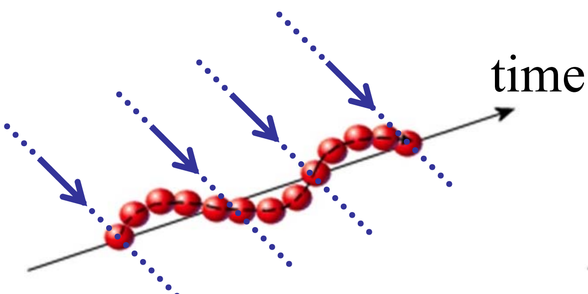
- **Handles:**
  - (1) Motion aliasing
  - (2) Motion blur

- **Application areas:**
  - - sports scenes
  - - scientific imaging
  - - etc...

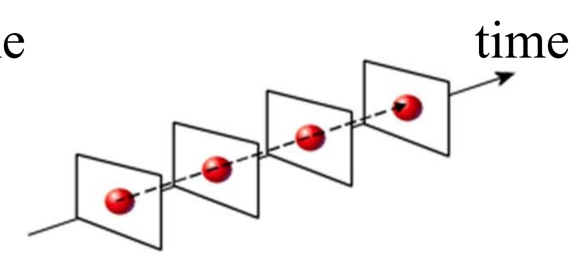
(1) Motion Aliasing

The “Wagon wheel” effect:

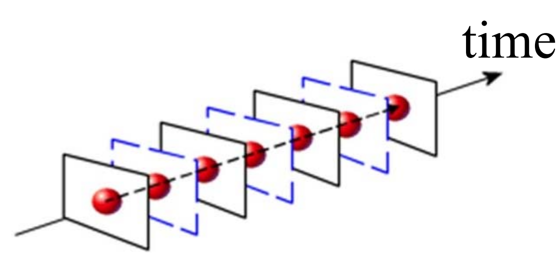
Slow-motion:



Continuous signal



Sub-sampled in time

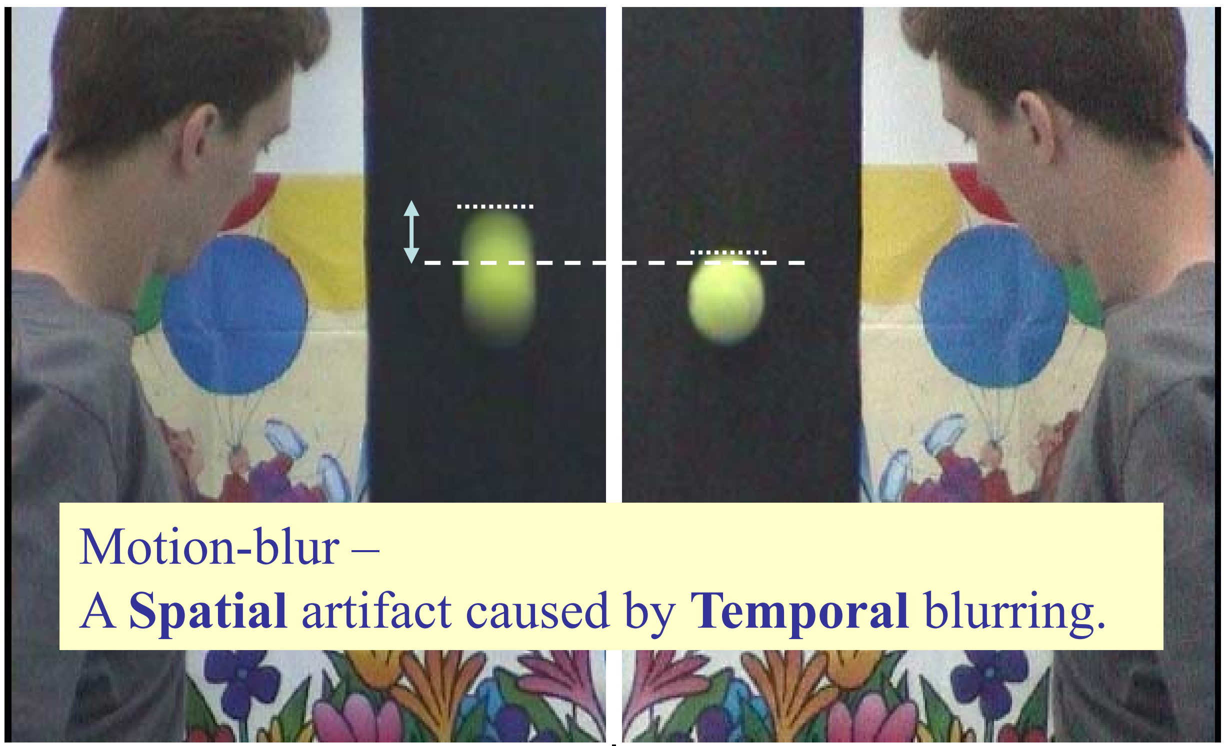


“Slow motion”

(2) Motion Blur

(2) Motion Blur

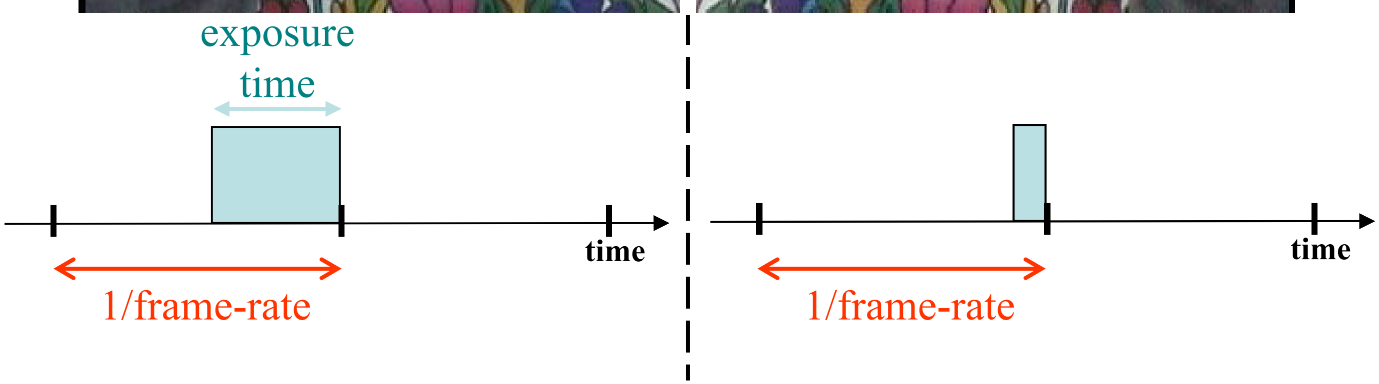
Camera 1:
long
exposure-time



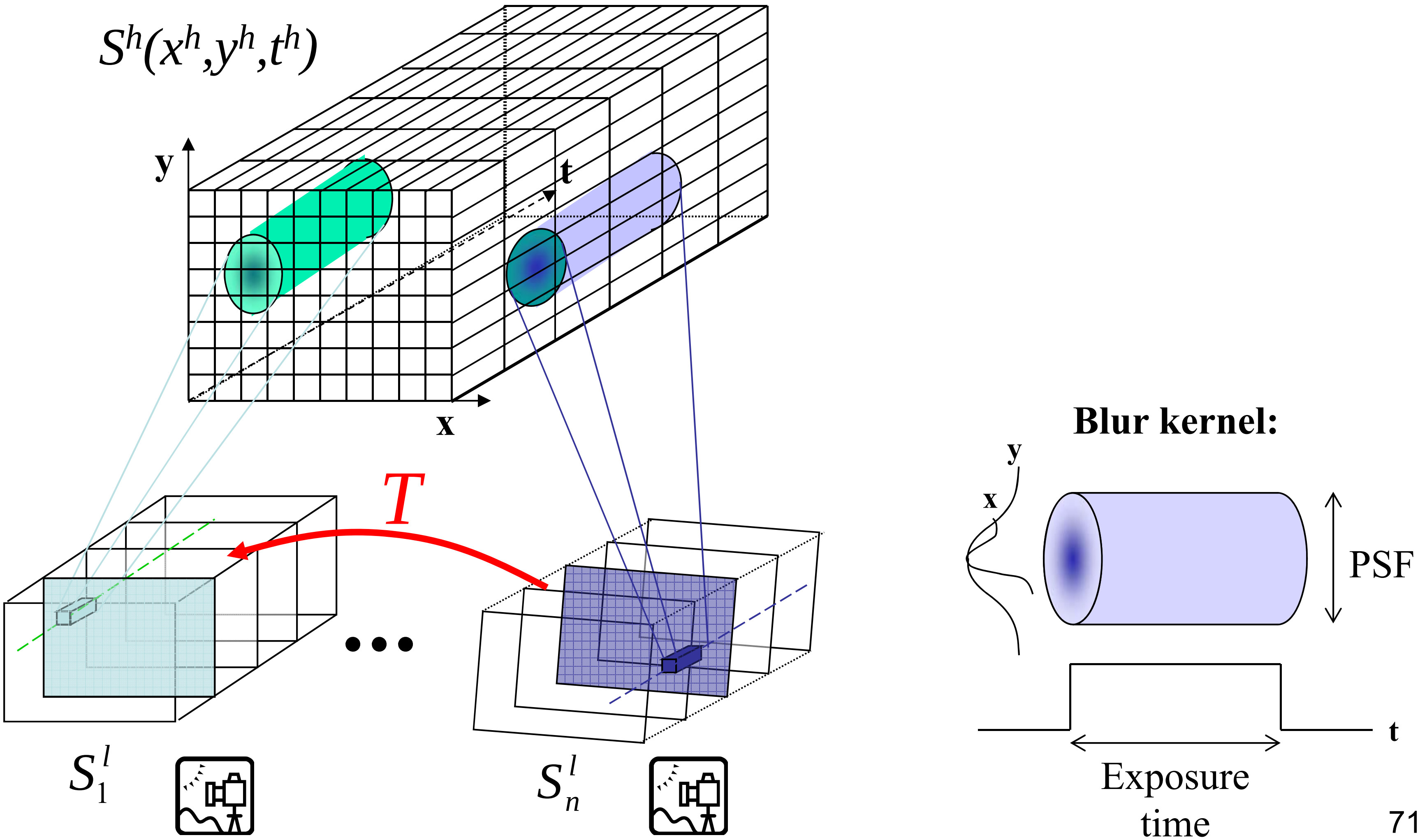
Camera 2:
short
exposure-time

Motion-blur –
A **Spatial** artifact caused by **Temporal** blurring.

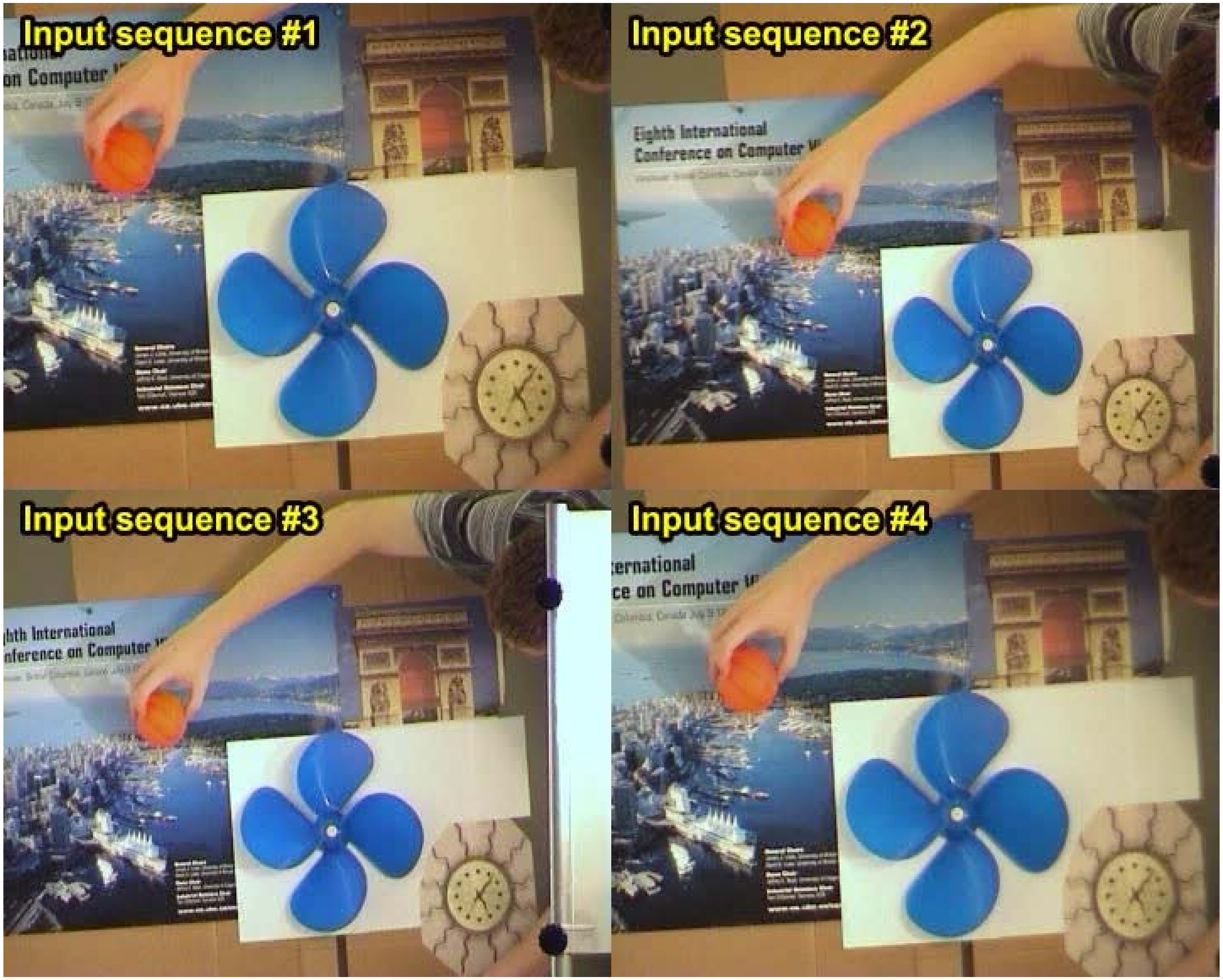
Operation of camera:



Space-Time Super-Resolution



Example: Motion-Aliasing

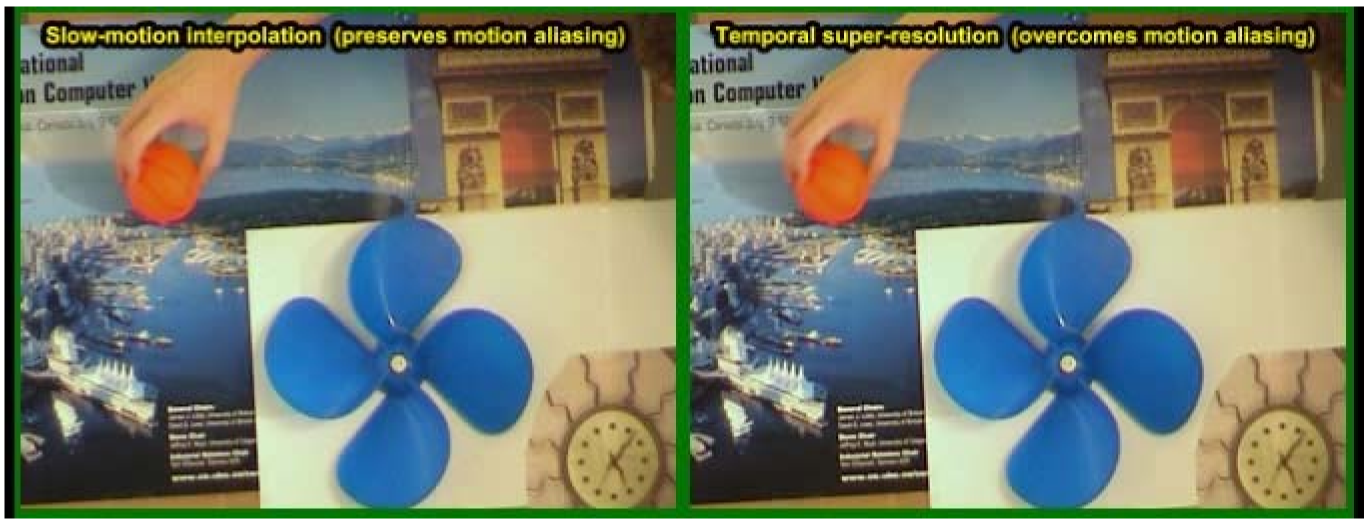


25 [frames/sec]

Example: Motion-Aliasing

Input sequence in slow-motion (x3):

Super-resolution in time (x3):



75 [frames/sec]

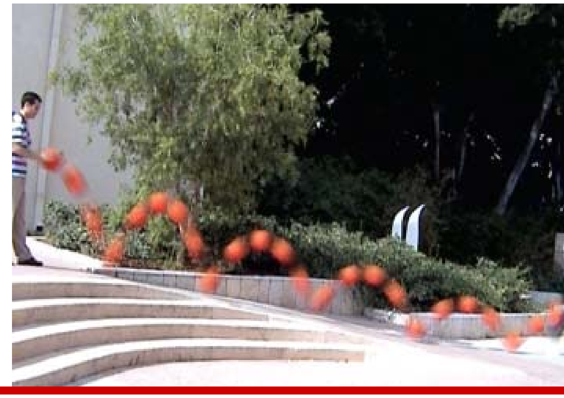
75 [frames/sec]

Deblurring:

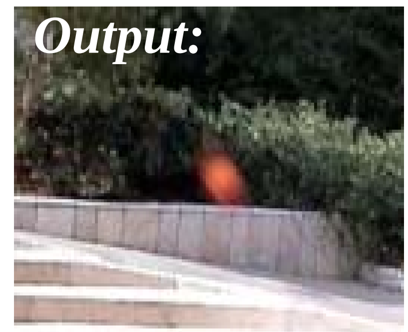
Output sequence:

(x15 frame-rate)

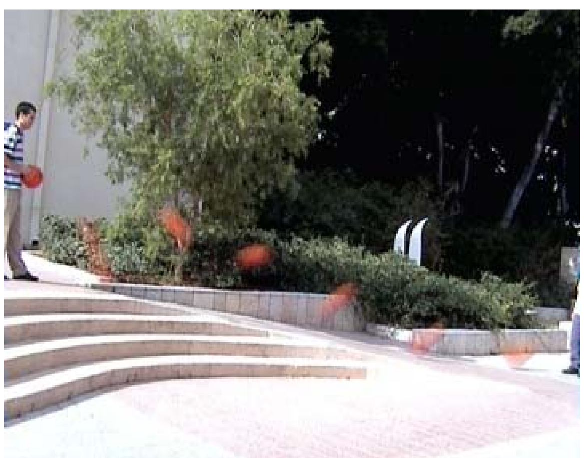
Output trajectory:

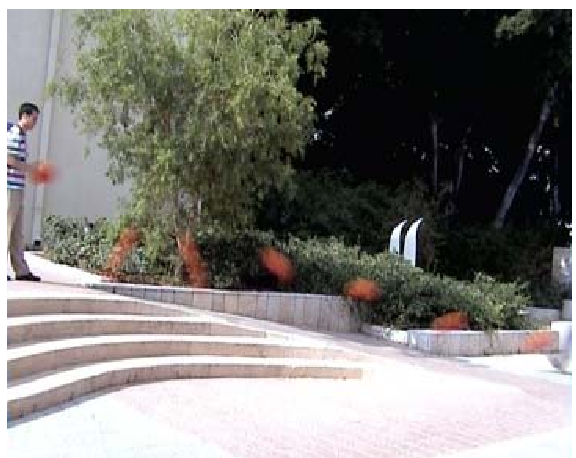


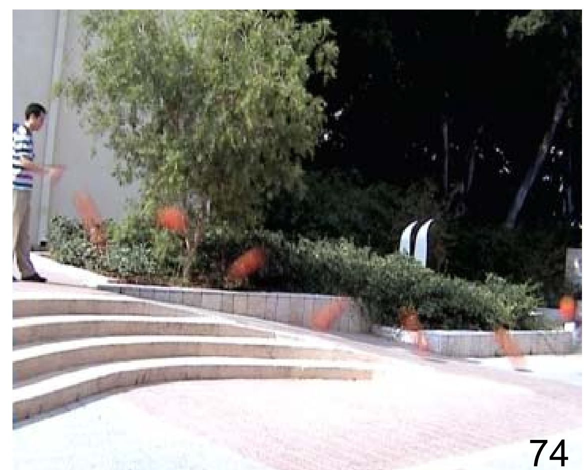
Without estimating motion of the ball!



3 out of **18** low-resolution **input sequences** (frame overlays; trajectories):







...

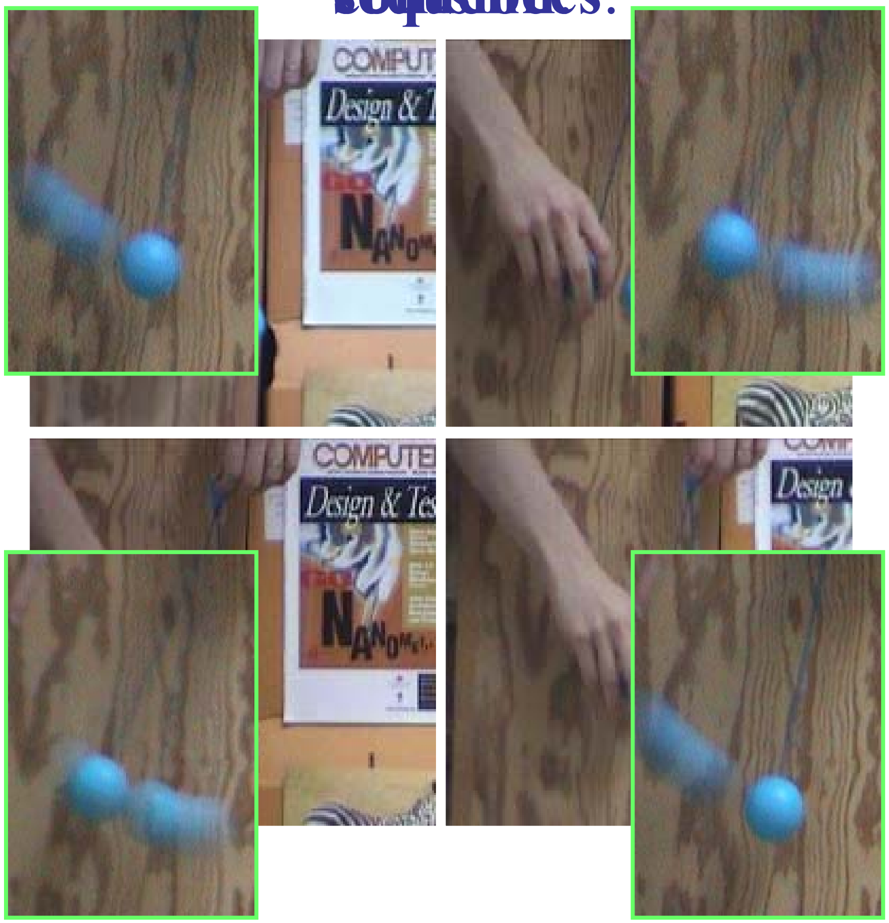
Example: Motion-Blur (real)

4 input sequences:

Frames at collision:

Video 1

Video 2



Video 3

Video 4

Output frame at collision: