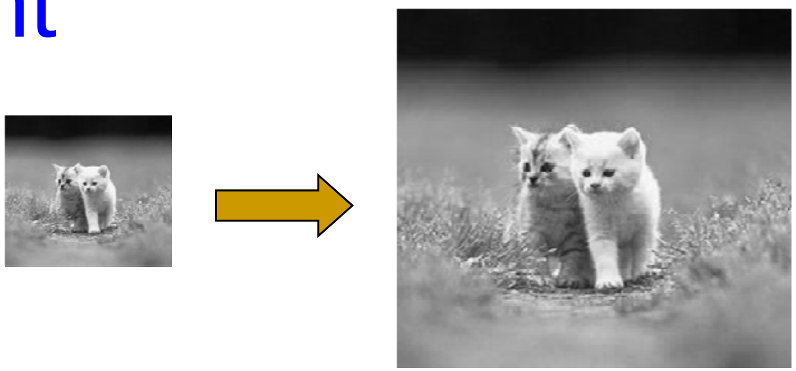
Multimedia Computing

Image/Video Resolution Enhancement



Topics

- Interpolation: basic single-frame resolution enhancement
  - Concepts
  - Techniques
  - Applications
- Super-resolution: advanced single/multi-frame resolution enhancement

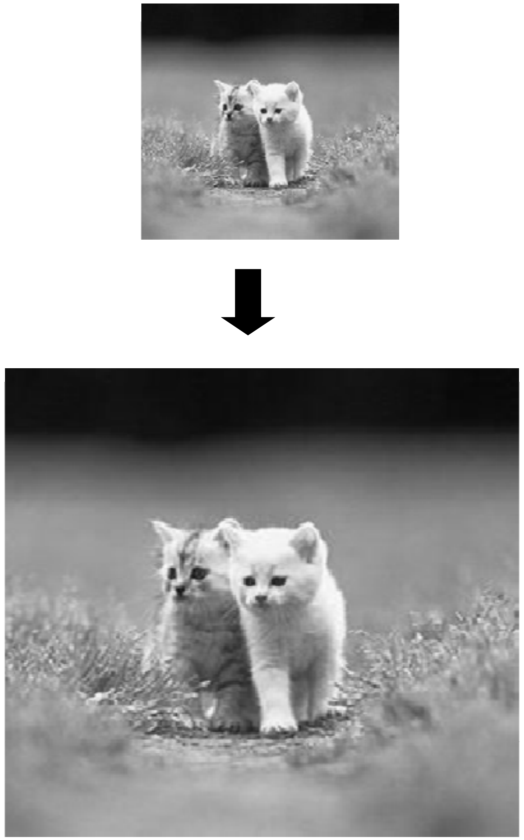
Image interpolation

- What is image interpolation?
  - Suppose there is a continuous image f(x,y) and we only have the intensity values at the integral lattice locations, i.e. when x and y are both integers.
  - Image interpolation refers to the estimation of intensity values at other missing locations, i.e., x and y can be arbitrary.

Image interpolation

- Why do we need image interpolation?
  - We want BIG images
    - When we view a picture or watch a movie on a PC, we prefer to view/watch it in the full screen mode.
  - We want GOOD images
    - If some parts of an image get damaged during the transmission, we want to repair it.
  - We want COOL images
    - Manipulate images digitally can render fancy artistic effects as we often see in movies.

Low resolution vs. High resolution



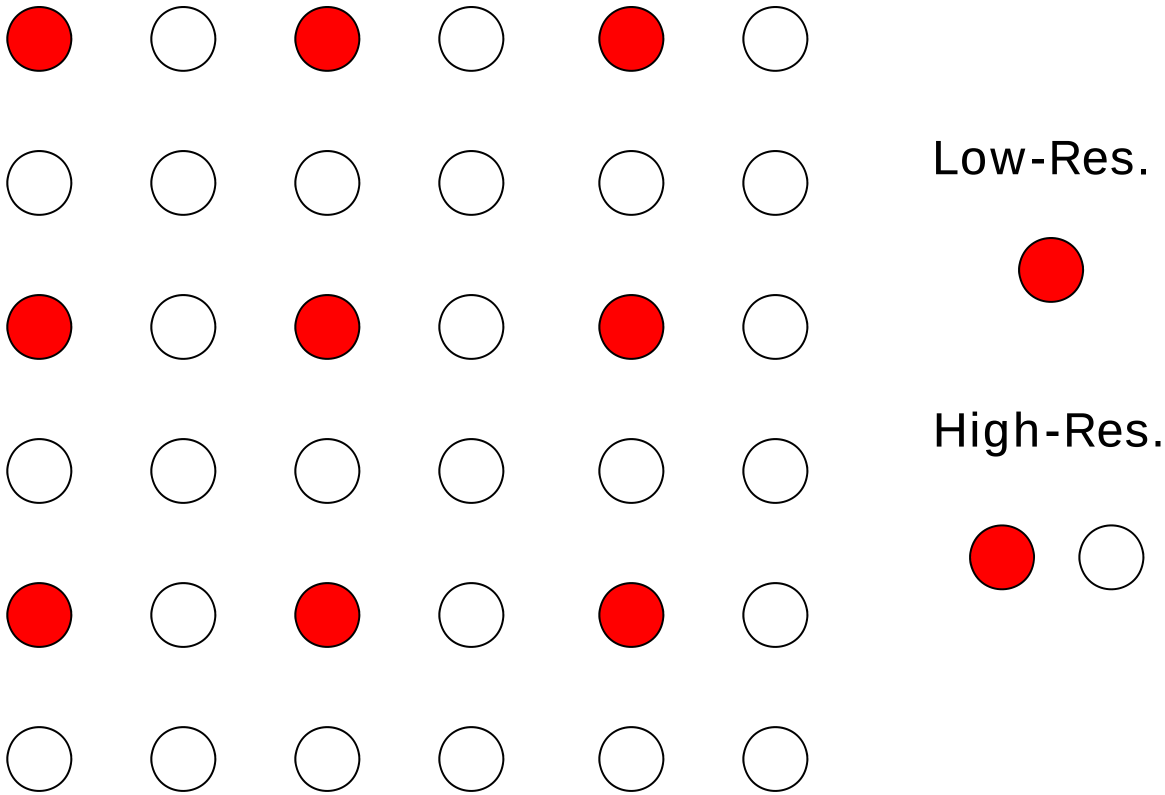
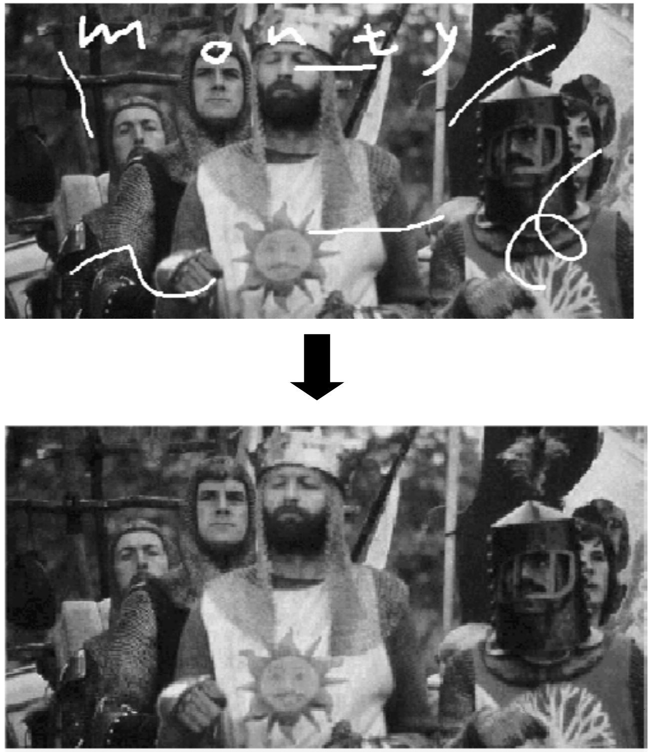
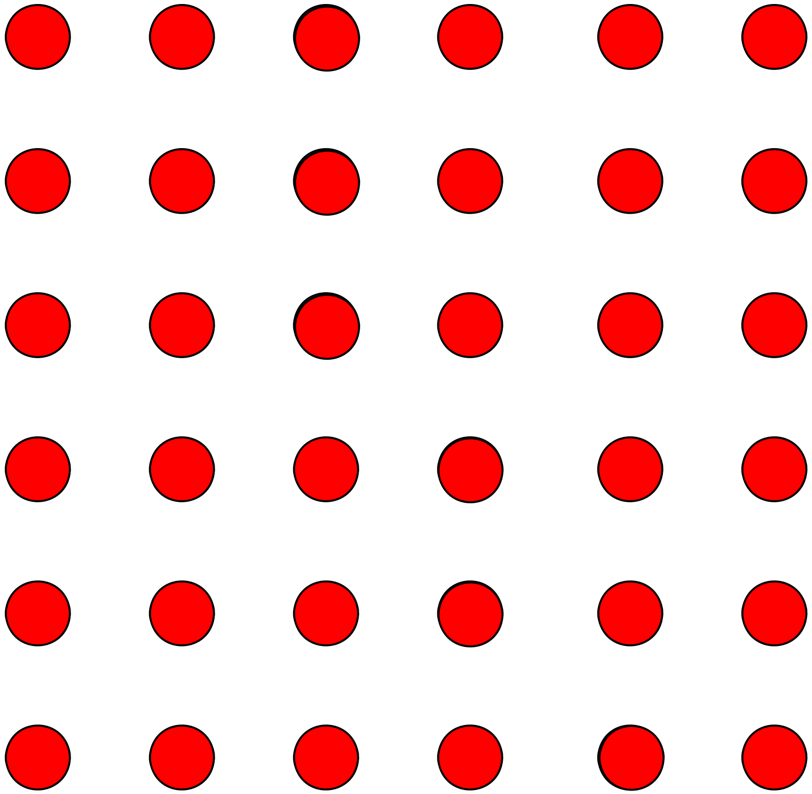


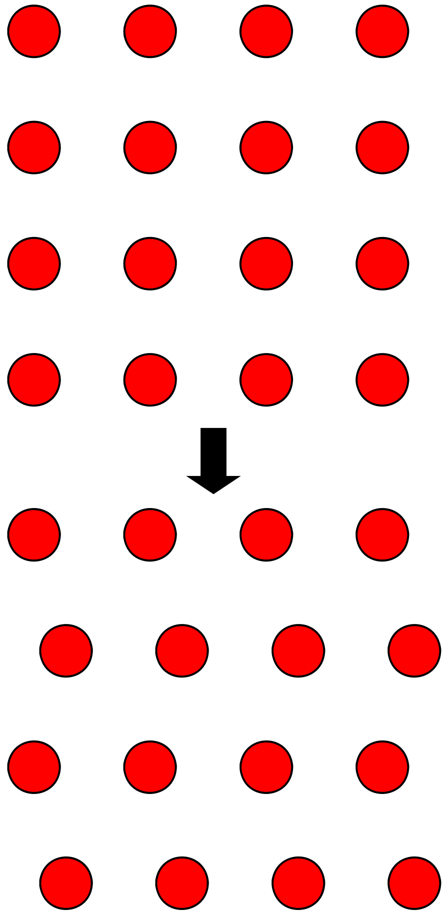
Image inpainting



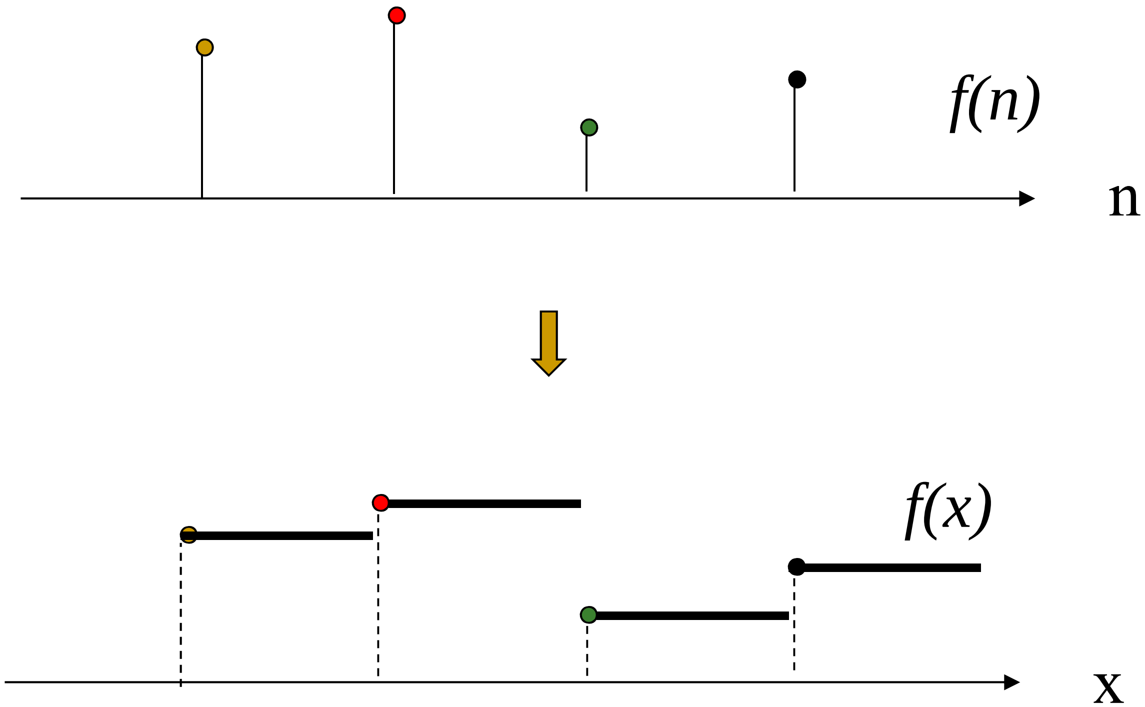


Non-damaged pixels   Damaged pixels

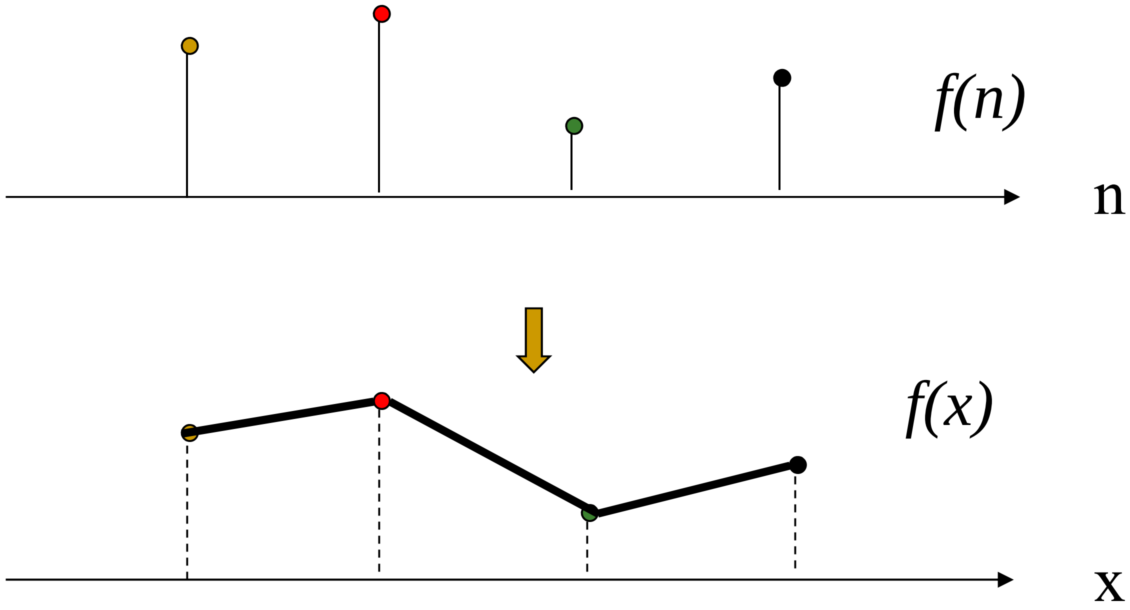
Image warping



1D interpolator: replication



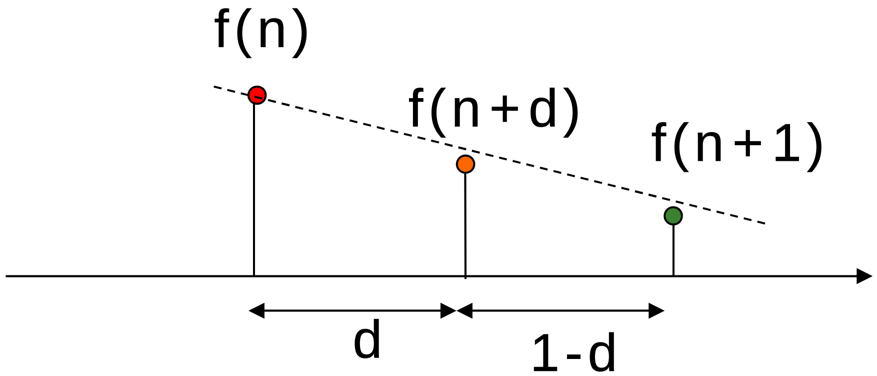
1D interpolation: linear



- Interpolation principle: use two neighboring points to fit a one-order (linear) polynomial f(x)=ax+b; then the value at any position can be calculated.

Linear interpolation: formula

The closer to a pixel, the higher weight is assigned



$$\boxed{f(n+d)=(1-d)\times f(n)+d\times f(n+1),\ 0<d<1}$$

When d=0.5, we simply have the average of two neighbors.

Examples

f(n)=[0,120,180,120,0]

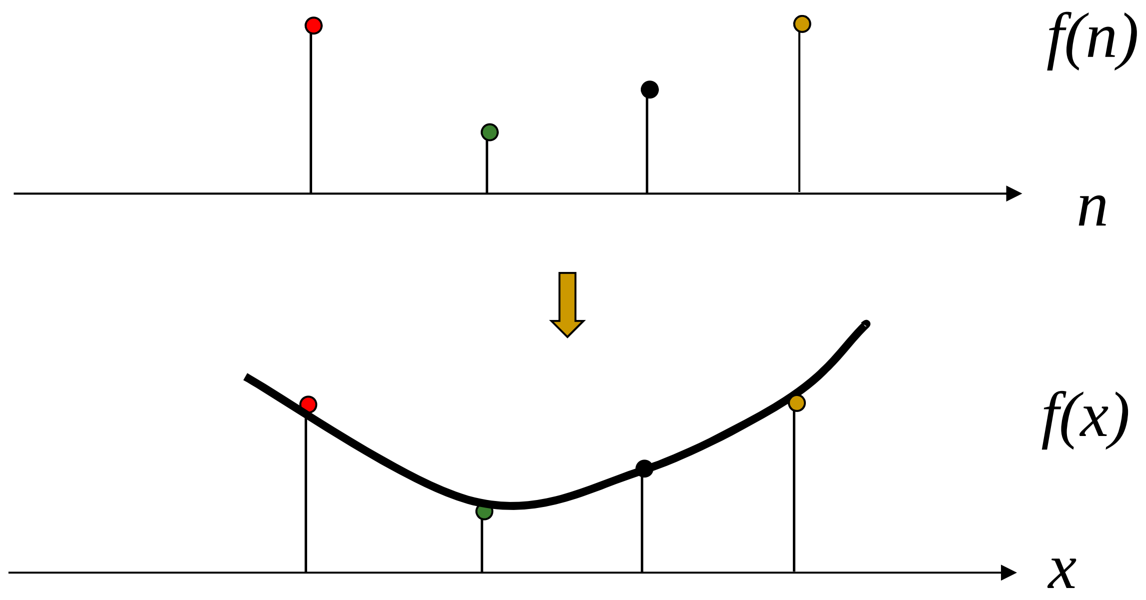
↓ Interpolate at 1/2-pixel

f(x)=[0,60,120,150,180,150,120,60,0]

↓ Interpolate at 1/3-pixel

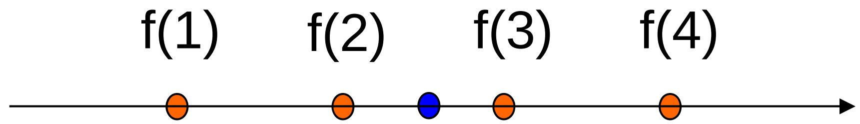
f(y)=[0,20,40,60,80,100,120,130,140,150,160,170,180,...]

1D interpolation: cubic



- Interpolation principle: use the four points to fit a third-order (cubic) polynomial $f(x)=ax^3+bx^2+cx+d$; then the value at any position can be calculated.

1D cubic interpolation at half-pel position



$$f(2.5)=(-1*f(1)+9*f(2)+9*f(3)-1*f(4))/16$$

Example:

f(n)=[10,60,120,100,80,60]

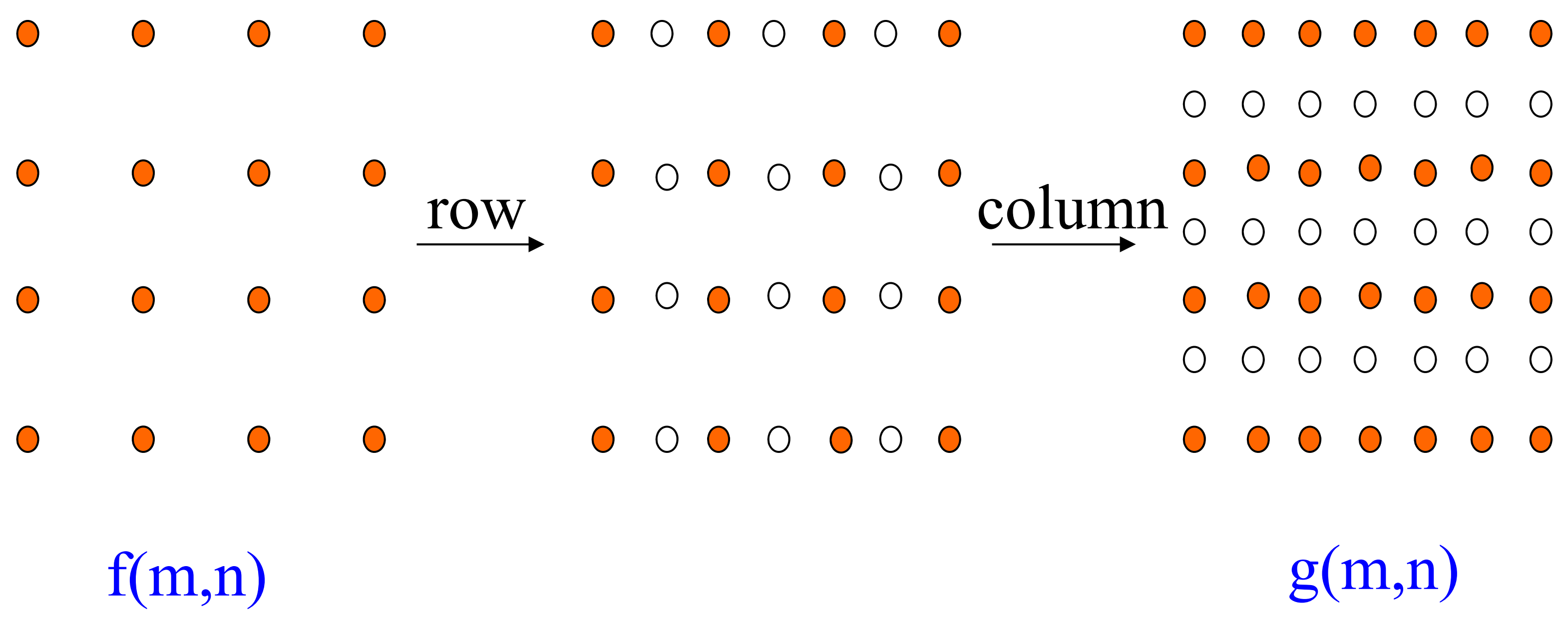
Interpolate at 1/2-pixel

f(x)=[10,_,60,94.375,120,115,100,90,80,_,60]

From 1D to 2D

- 2D interpolation can be implemented by two sequential 1D interpolators along row and column direction respectively. In other word, 2D interpolation can be decomposed into two sequential 1D interpolations.
- The row and column interpolation order does not matter (row-column = column-row).
- So linear→bilinear; cubic→bicubic.
- Such separable implementation is not optimal but enjoys low computational complexity.

Illustration of interpolation at half-pel



Numerical example at half-pel position

a	b
c	d

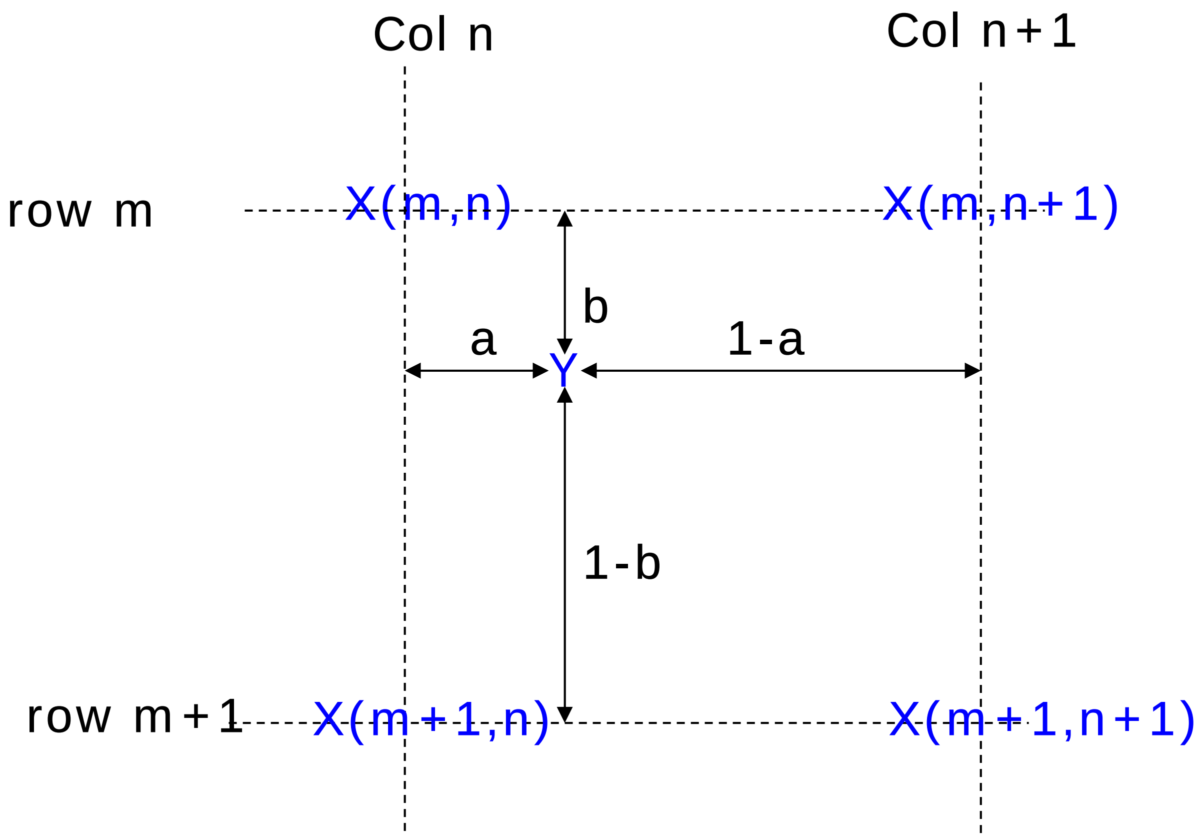
replication

a	a	b	b
a	a	b	b
c	c	d	d
c	c	d	d

linear

a	(a+b)/2	b
(a+c)/2	(a+b+c+d)/4	(b+d)/2
c	(c+d)/2	d

Linear interpolator at arbitrary position

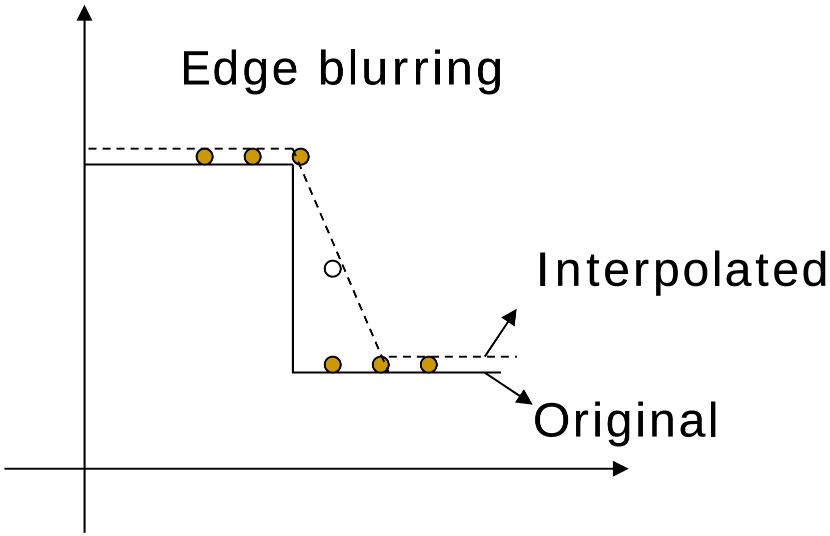


Q: what is the interpolated value at Y?

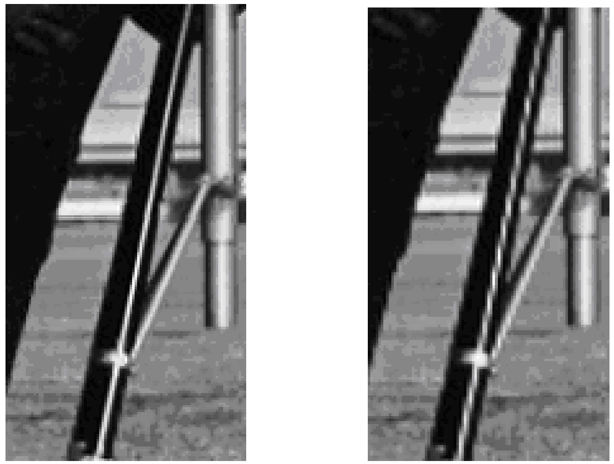
Ans.: (1-a)(1-b)X(m,n)+(1-a)bX(m+1,n)
+a(1-b)X(m,n+1)+abX(m+1,n+1)

Limitation with bilinear/bicubic

- Edge blurring
- Jagged artifacts



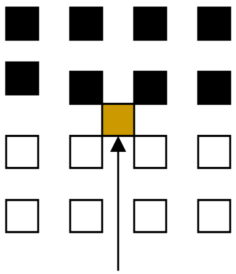
Jagged artifacts



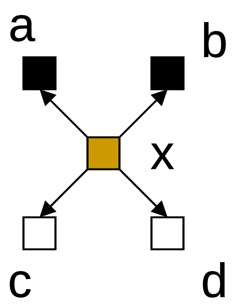
Original Interpolated

A simple directional linear interpolator

Step 1: interpolate the missing pixels along the diagonal



The pixel to be interpolated



If $|a-d| \approx |b-c|$

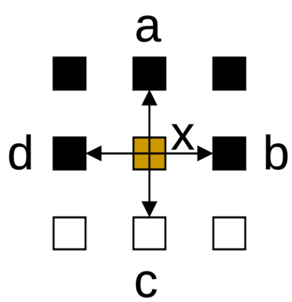
$x = (a+b+c+d)/4$

If $|a-d| > |b-c|$

$x = (b+c)/2$

otherwise $x = (a+d)/2$

Step 2: interpolate the other missing pixels



For example, since $|a-c| > |b-d|$

$x = (b+d)/2$

Applications

- Interpolation Applications
  - Resolution enhancement: zooming
  - Image inpainting (error concealment)
  - Others: geometric transformation

Matlab function “interp2.m”

```
% read the image into MATLAB
I=imread('cameraman.tif');
[n,m]=size(I);

%%We focus on part of the image
I=I(71:150,71:150);

% display the image
figure(1),clf;
imshow(I,[0 255]);

I=double(I);

% use Matlab built-in function
%"interp2" to zoom I
```

```
% 1. replication
A=interp2(I,2,'nearest');
% "2" means enlarge 2 times, i.e. 2^2=4 times
% the original image's size
figure(2),clf;
imshow(A,[0 255]);

% 2. bilinear interpolation
B=interp2(I,2,'linear');
figure(3),clf;
imshow(B,[0 255]);

% 3. bicubic interpolation
C=interp2(I,2,'cubic');
figure(4),clf;
imshow(C,[0 255]);
```

Pixel replication

Small image

4×4 times enlarged

Bilinear interpolation

Small image

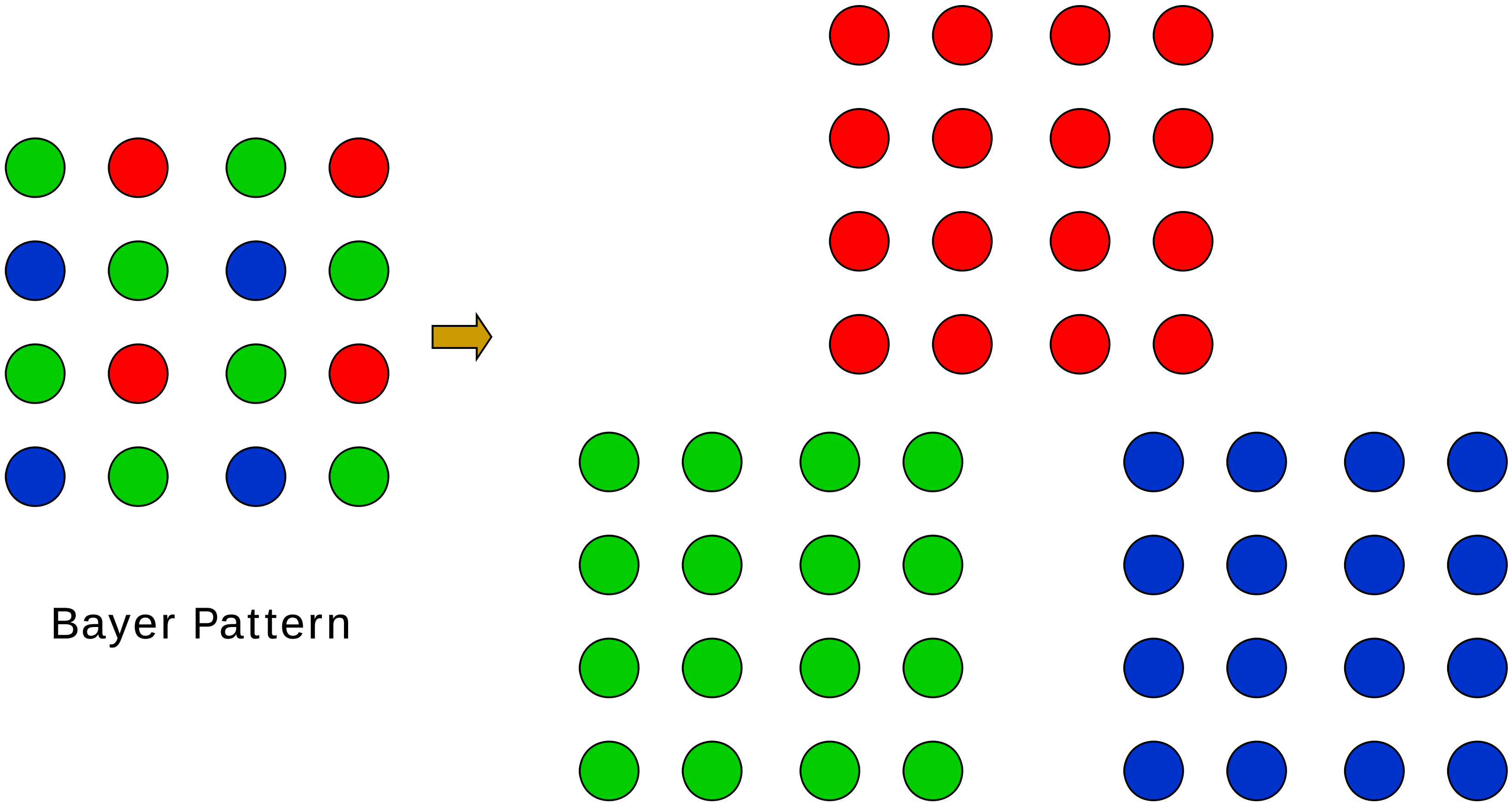
4×4 times enlarged

Bicubic interpolation

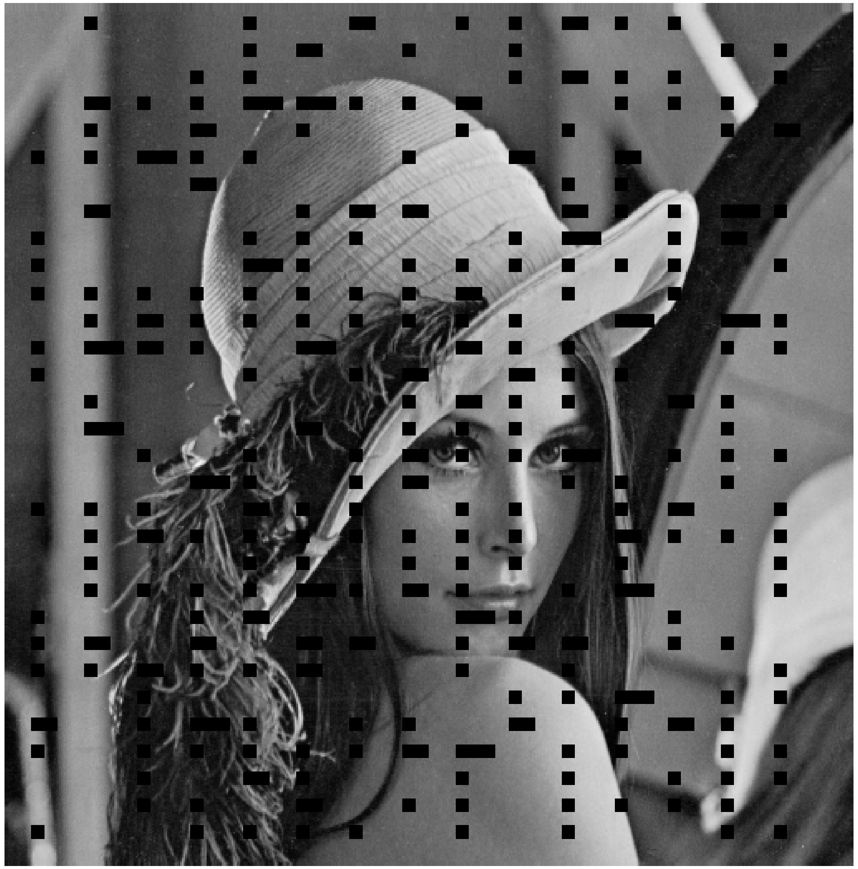
Small image

4×4 times enlarged

Image color demosaicing
(Color-Filter-Array Interpolation)



Error concealment

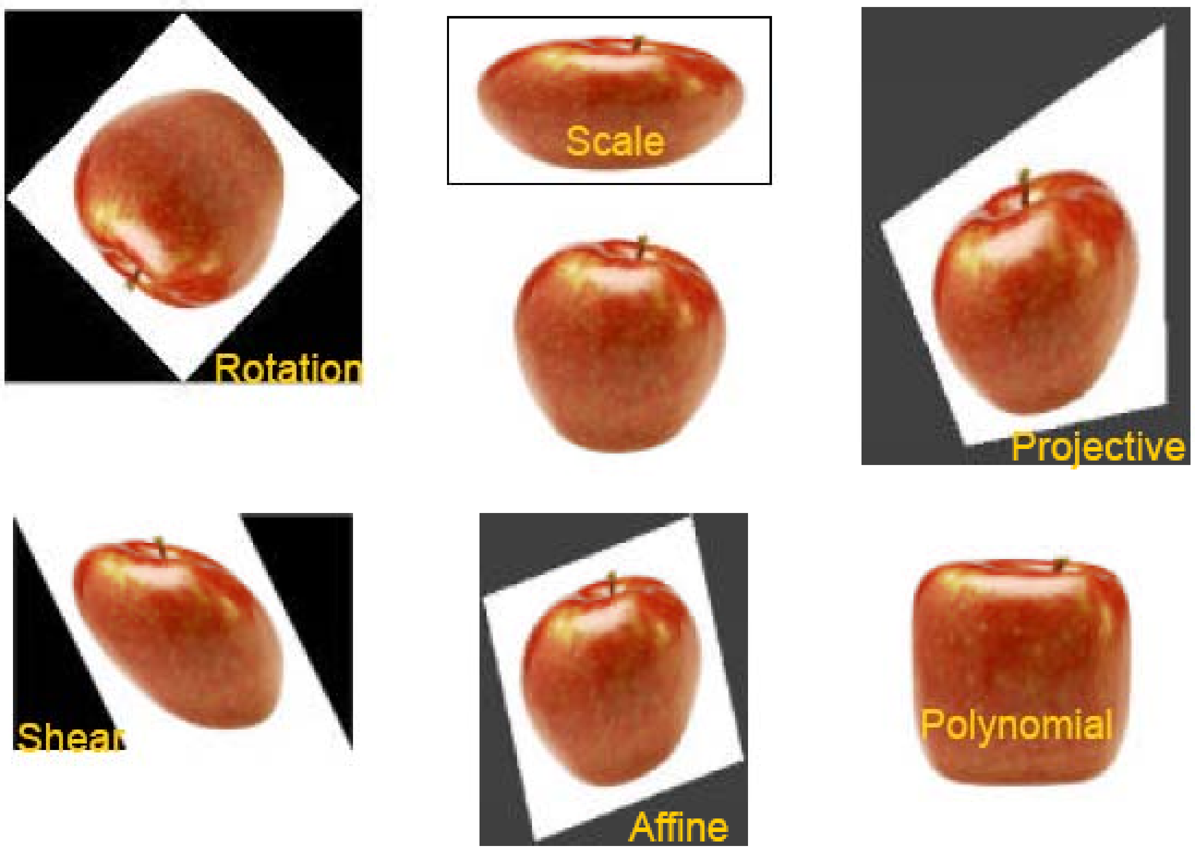


damaged

interpolated

Image inpainting

Geometric transformation



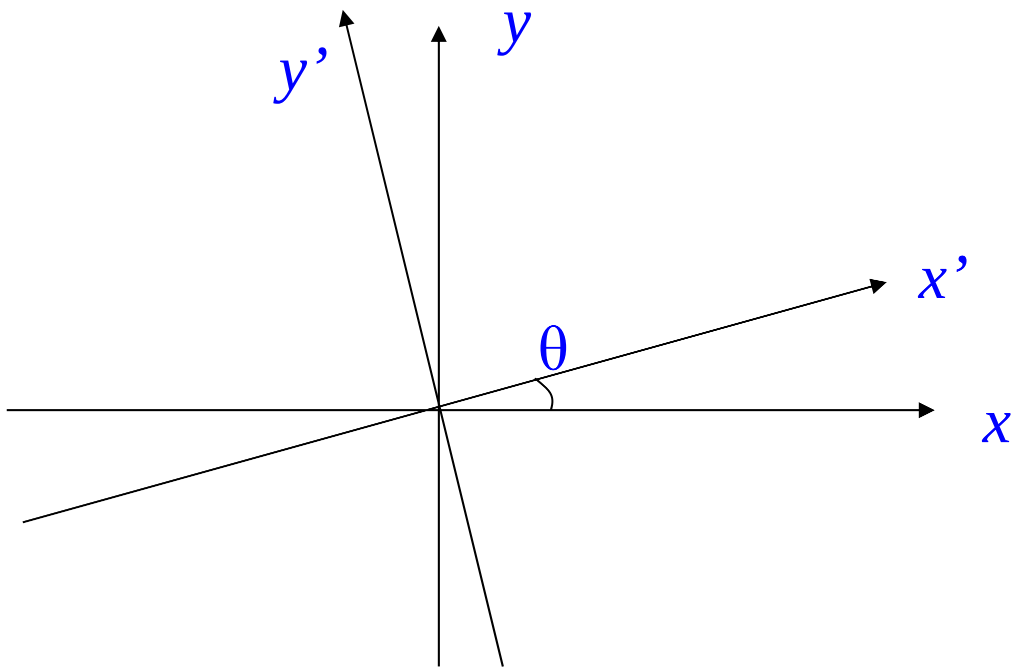
Widely used in computer graphics to generate special effects

MATLAB functions: griddata, interp2, maketform, imtransform

Scale

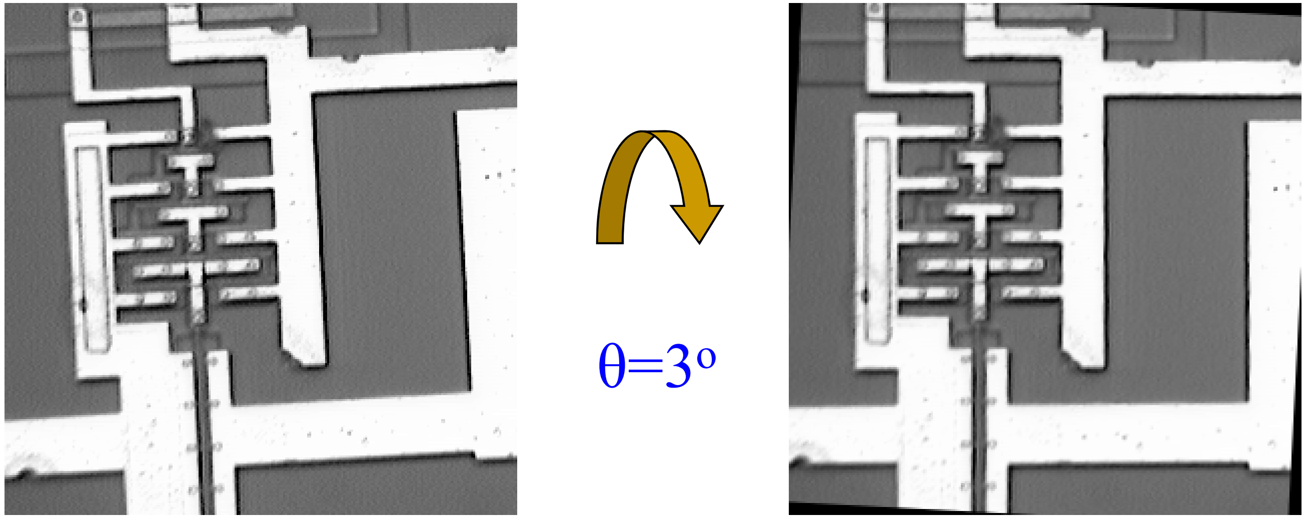
a=2

Rotation

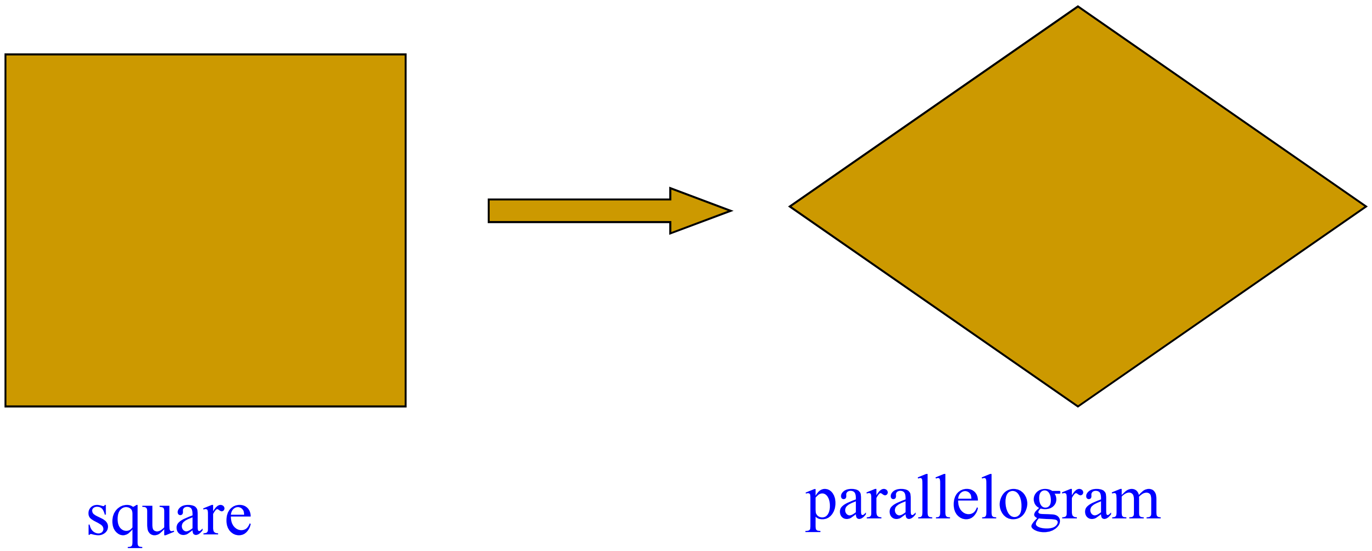


$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation example



Affine transform

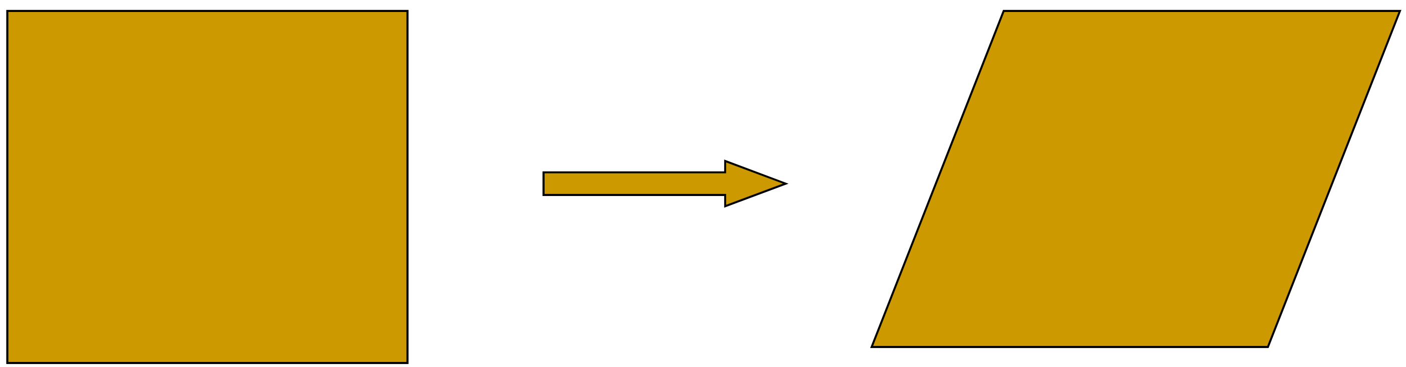


$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

Affine transform example

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} .5 & 1 \\ .5 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Shear



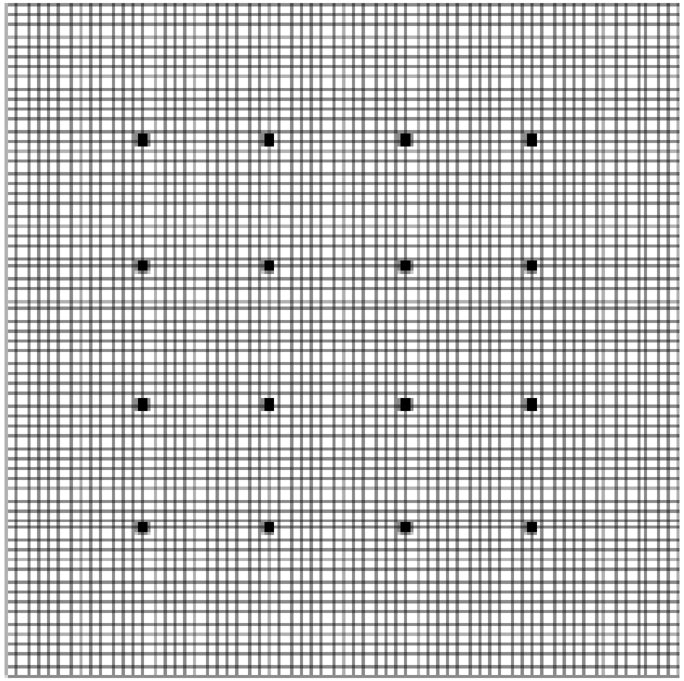
square　　　　parallelogram

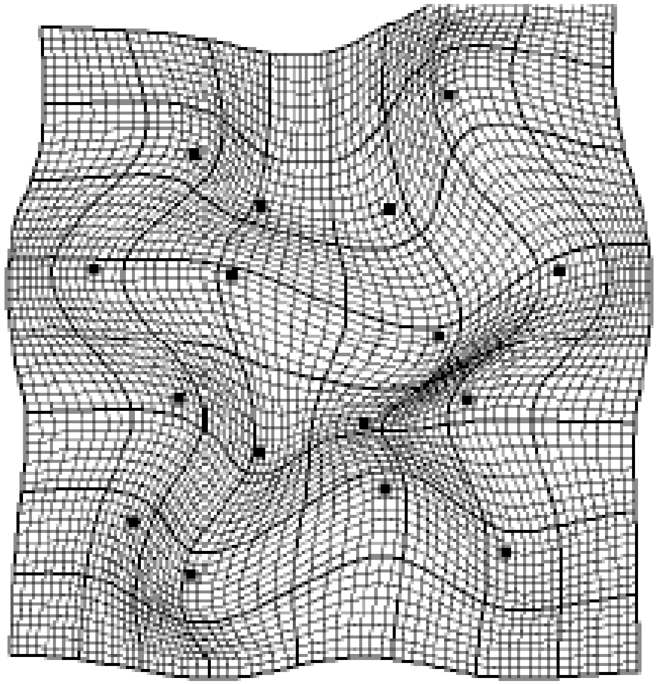
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

Shear example

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ .5 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

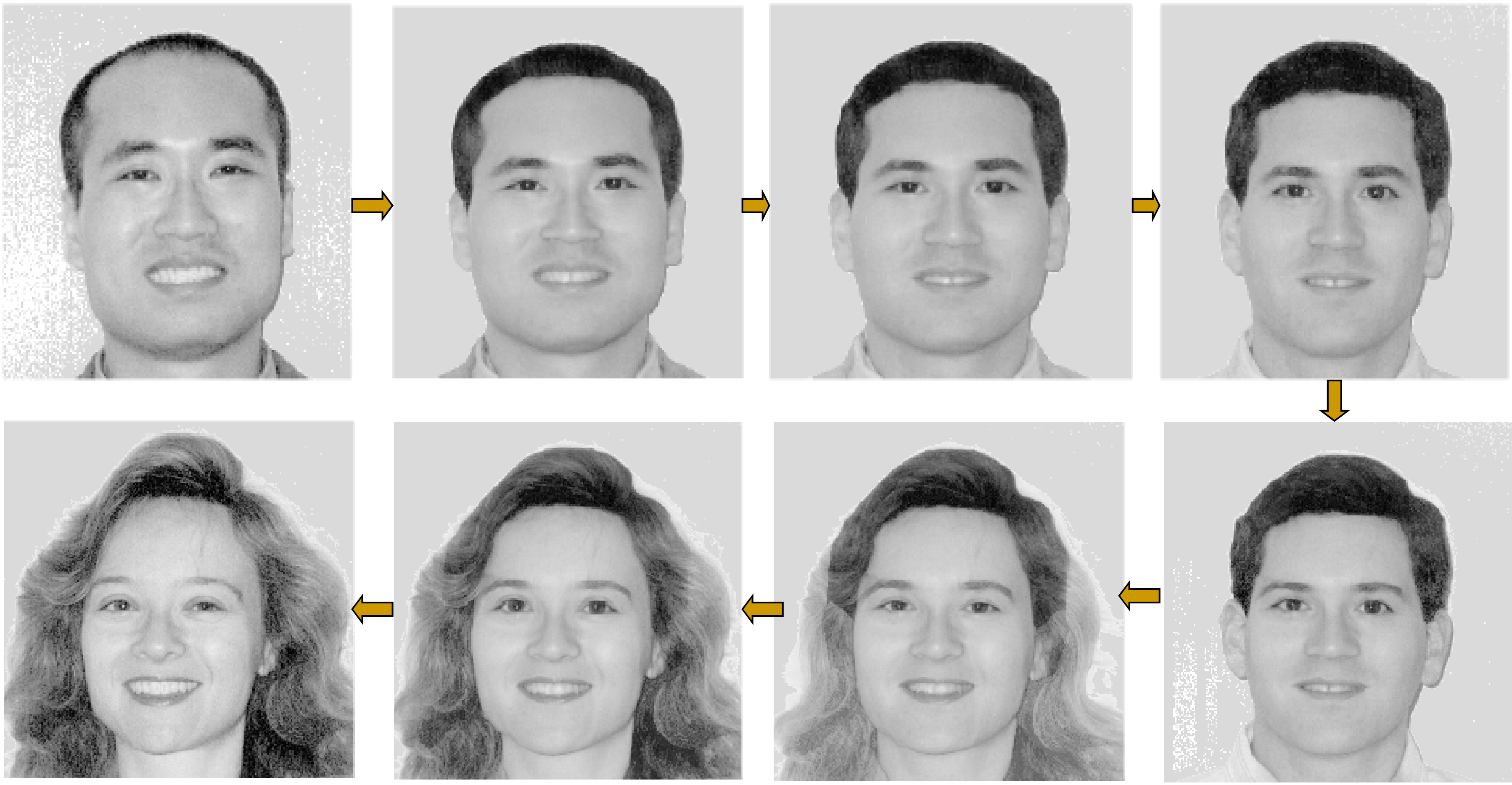
Free form deformation





Seung-Yong Lee et al., “Image Metamorphosis Using Snakes and Free-Form Deformations,”*SIGGRAPH’1985,* Pages 439-448

Application to image metamorphosis



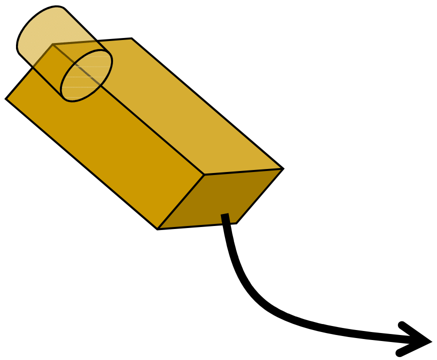
Topics

- Interpolation: basic single-frame resolution enhancement
  - Concepts
  - Techniques
  - Applications
- Super-resolution: advanced single/multi-frame resolution enhancement

Imaging system

High resolution scene

Resolution is reduced during imaging process due to physical constraints of sensor, nonzero aperture time, optical blurring, motion, sensor noise, etc.

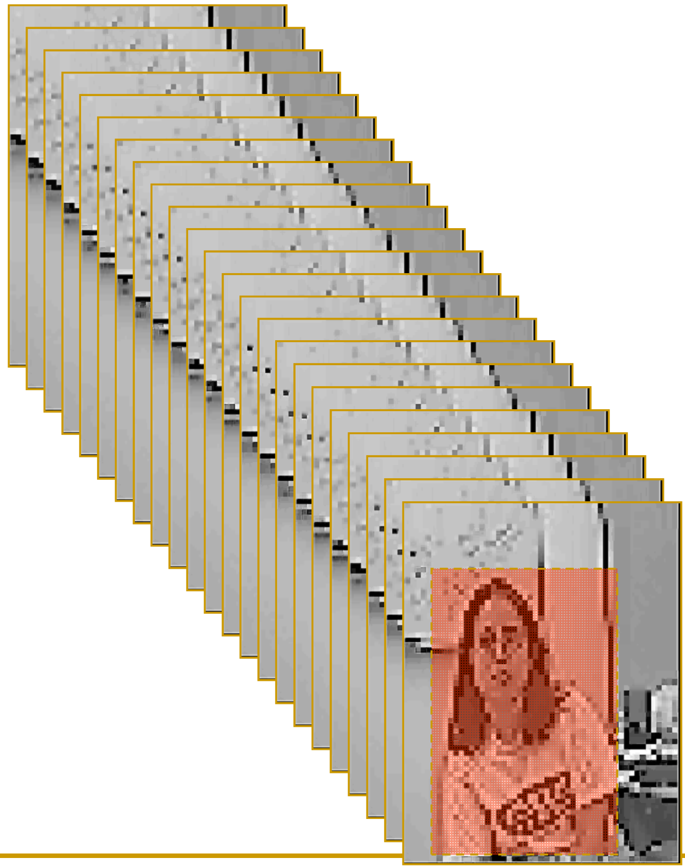


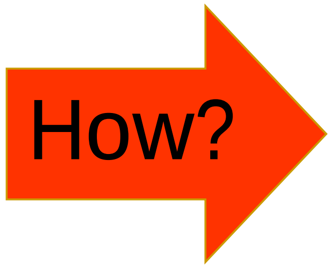
Low Resolution image

Super-resolution: idea

Given: A set of low-resolution images

Goal: Fusion of these images into a higher resolution image

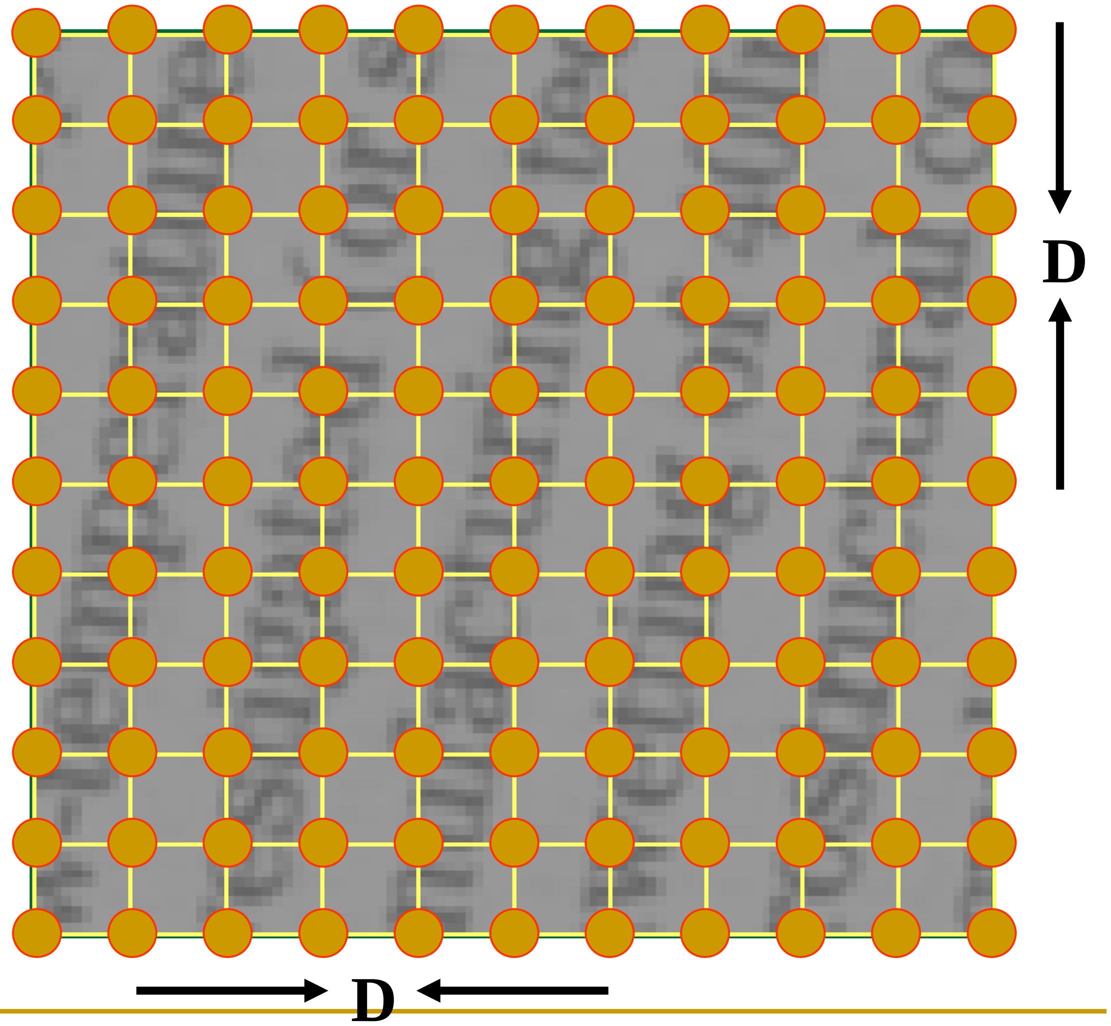




Comment: This is an actual super-resolution reconstruction result

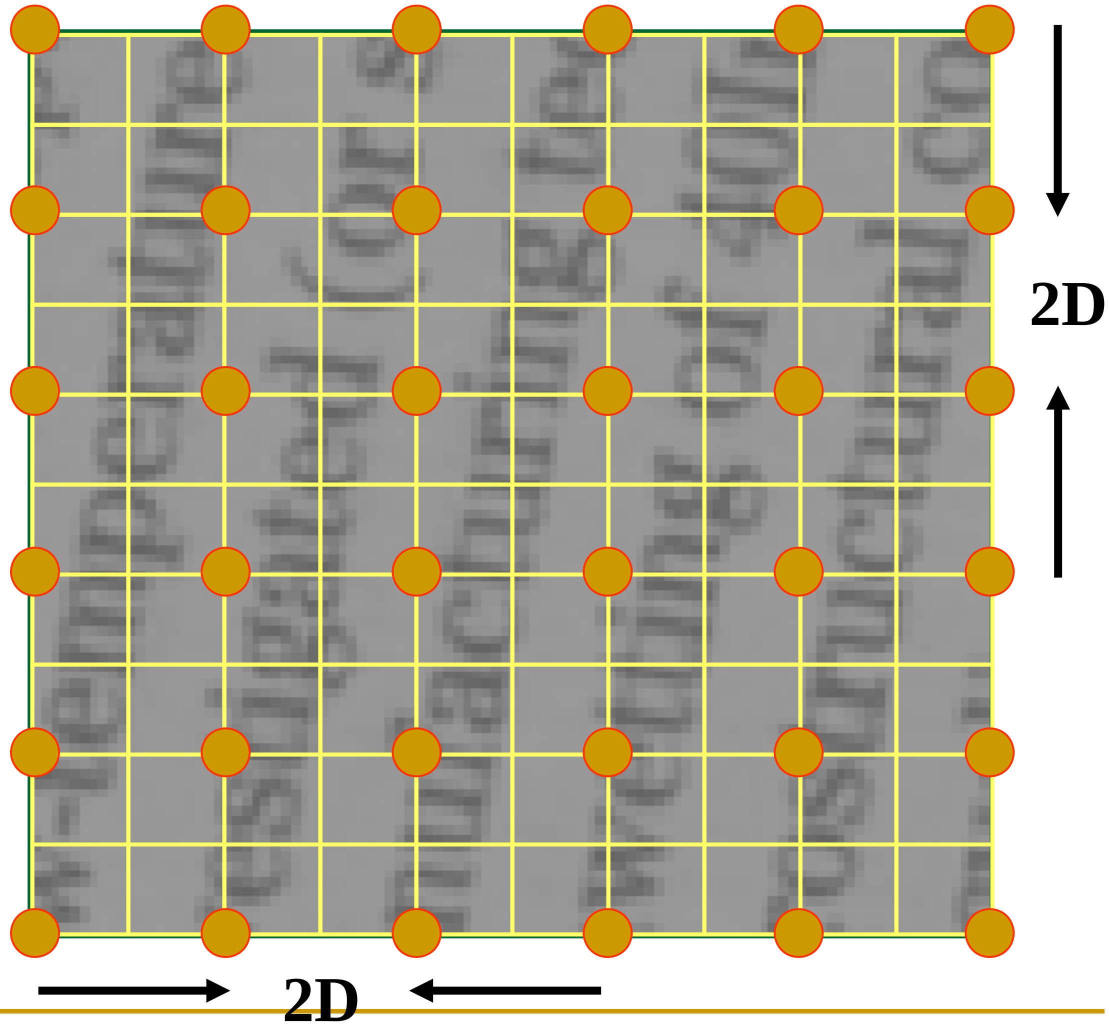
Simple example

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



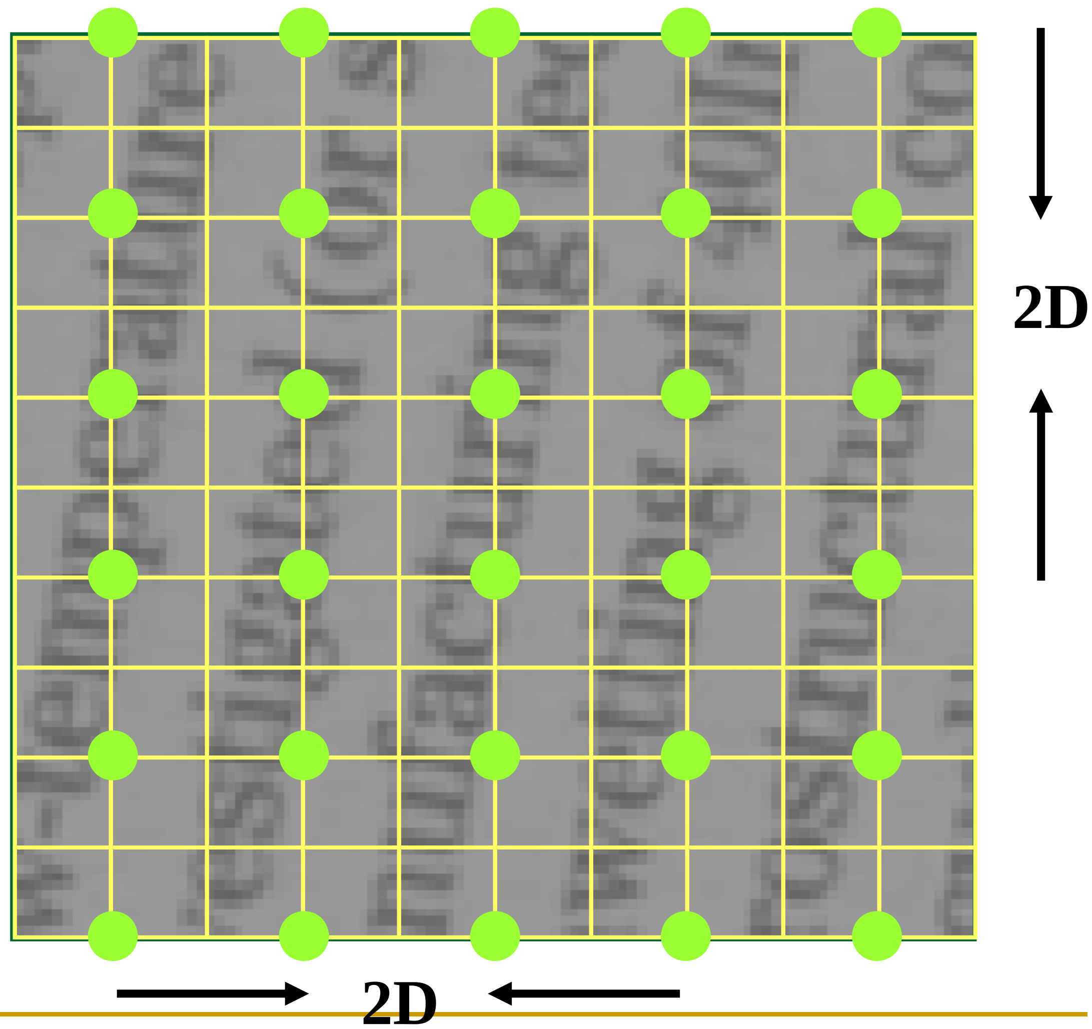
Simple example

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



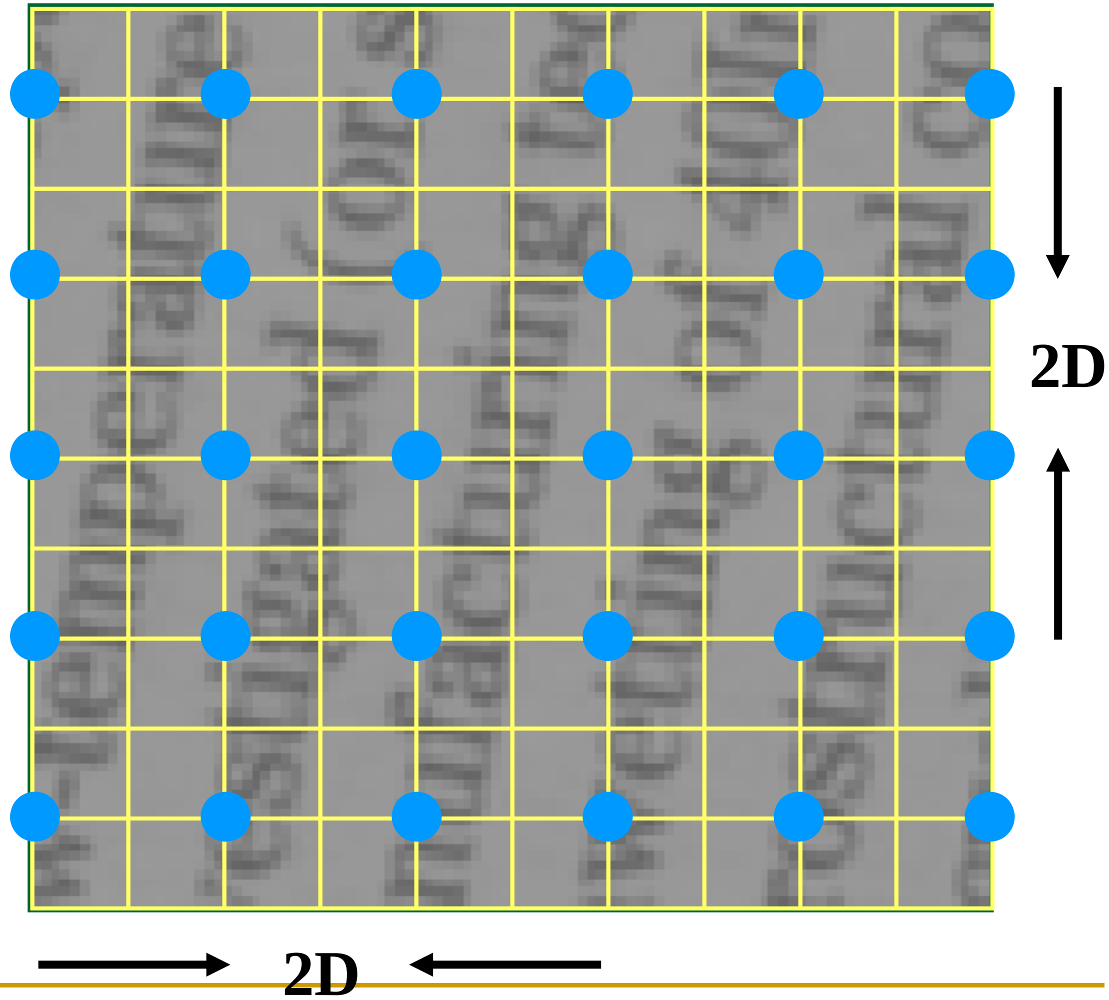
Simple example

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



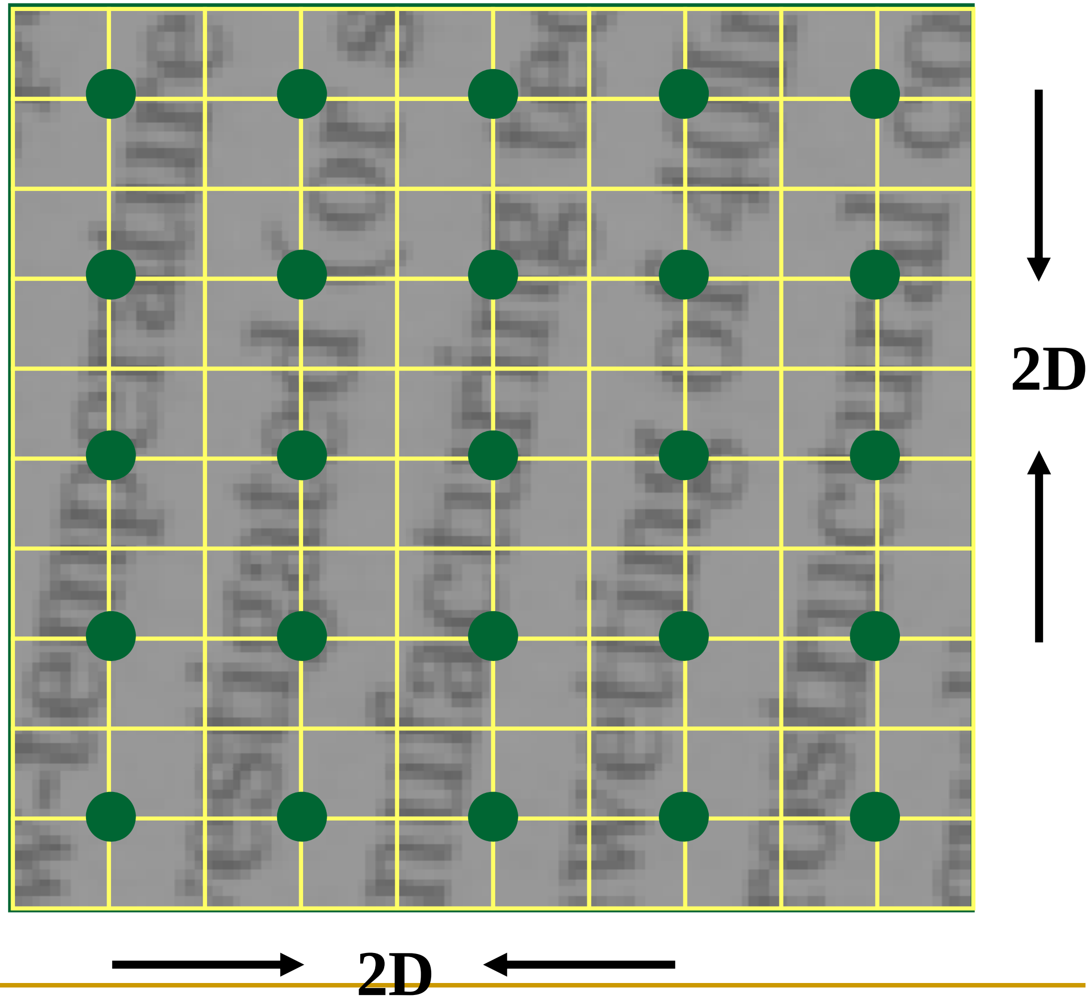
Simple example

Similarly, by shifting down we get a third image



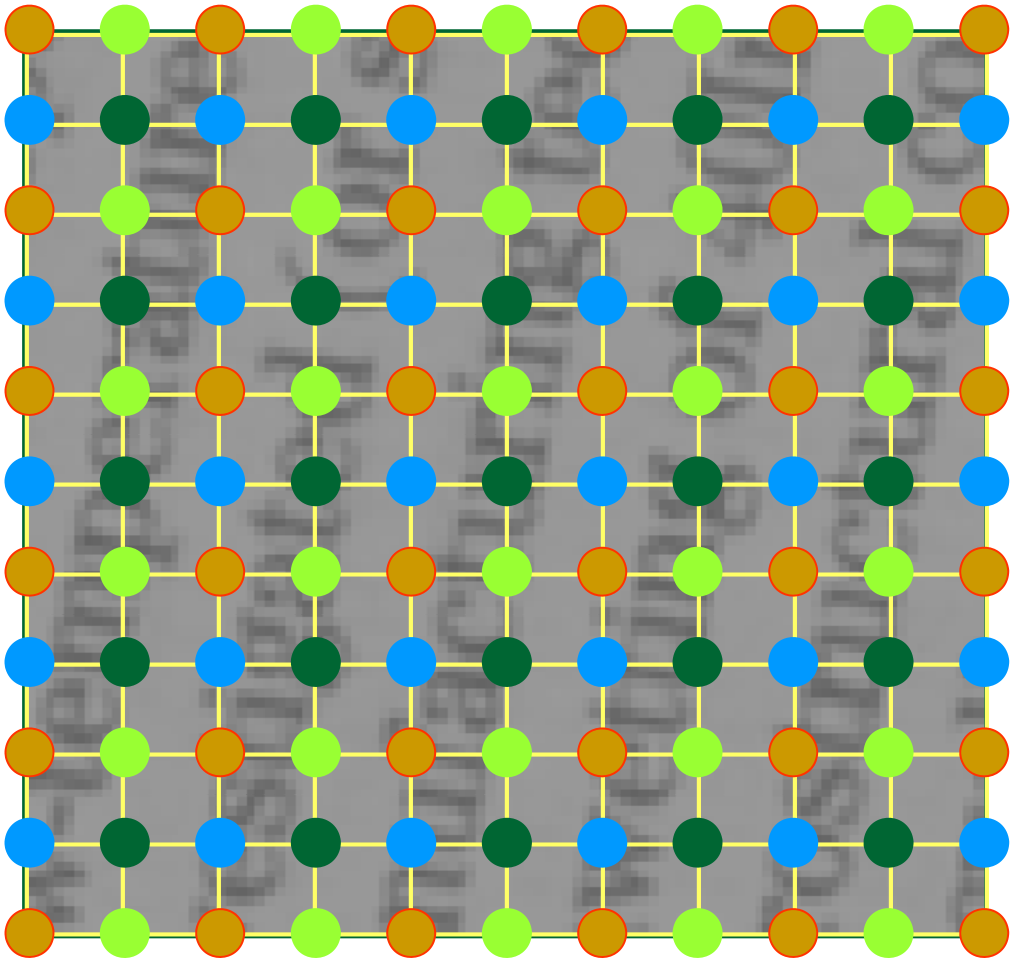
Simple example

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



Simple example - finally

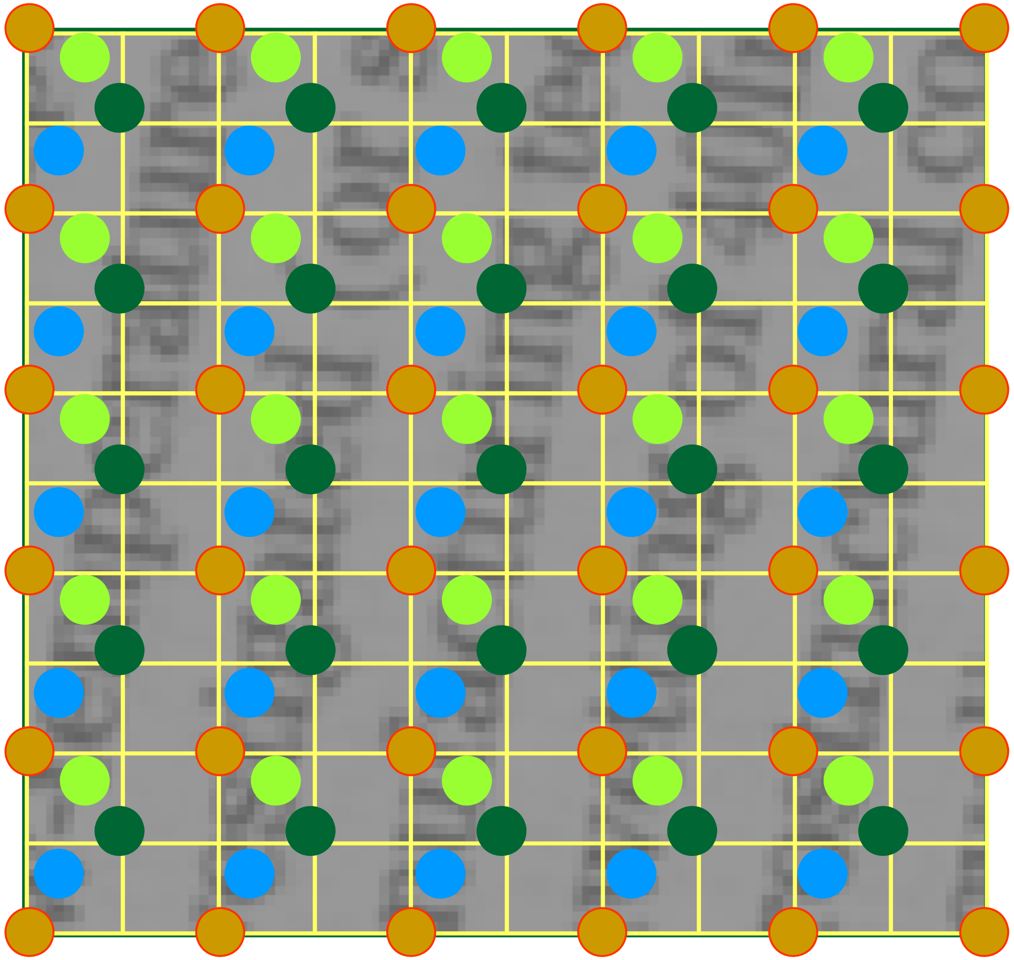
It is trivial to see that interlacing the four images, we get that the desired resolution is obtained, and thus perfect reconstruction is guaranteed.



This is Super-Resolution in its simplest form

Uncontrolled displacements

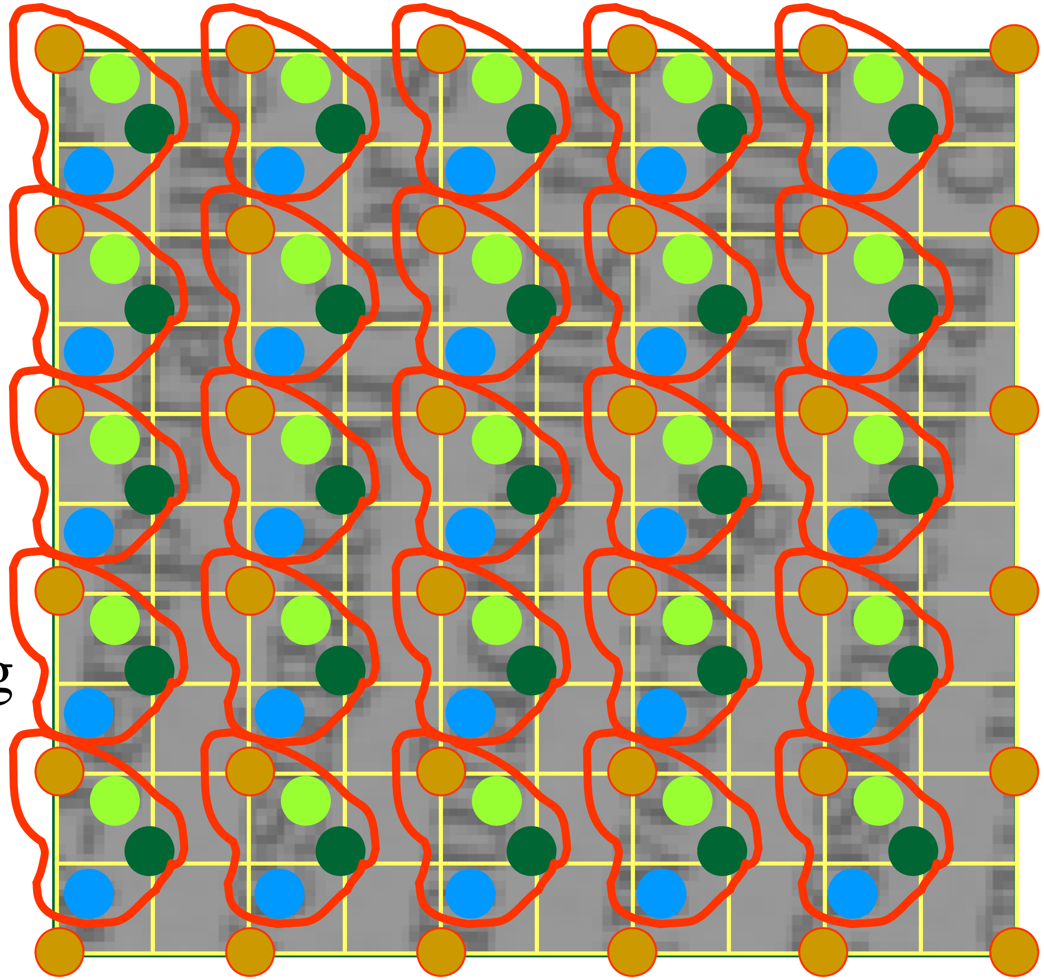
In the previous example we assumed the camera moves exactly as what we want.



What if the camera displacement is uncontrolled?

Uncontrolled displacements

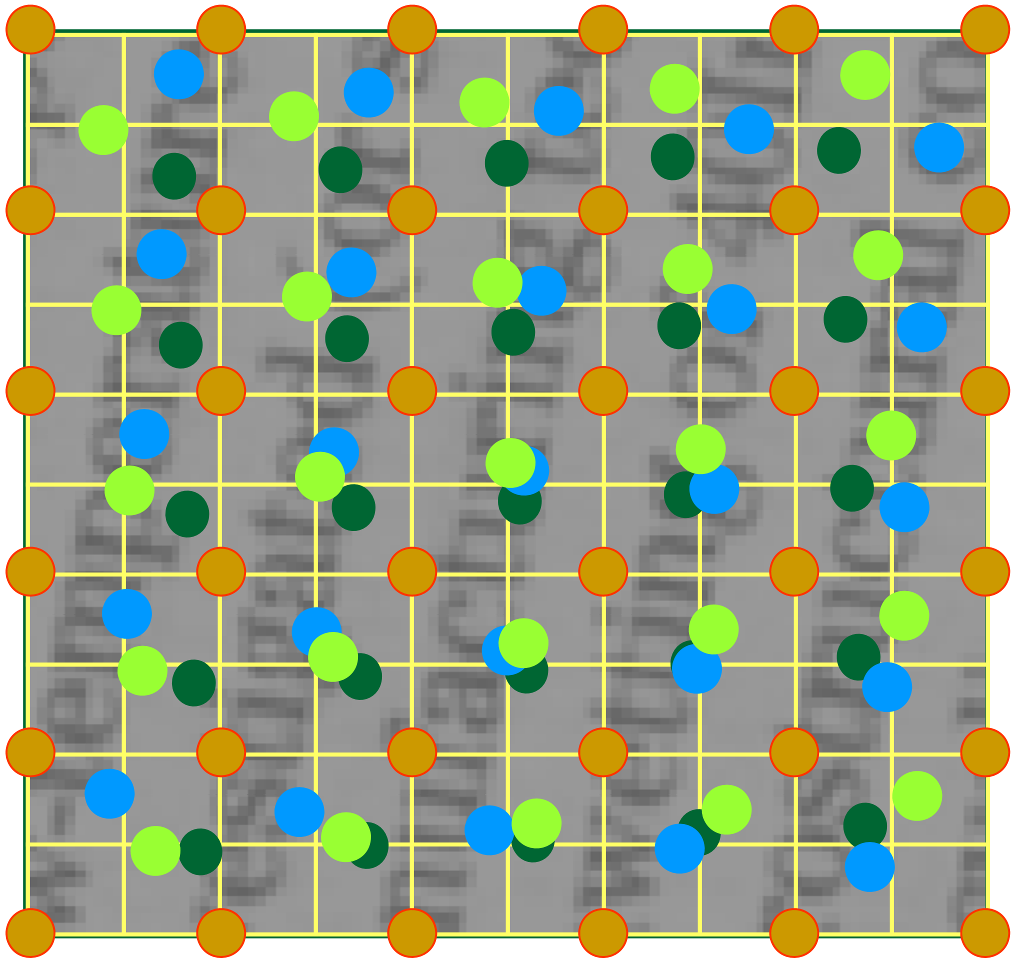
It turns out that there is a sampling theorem due to Yen (1956) and Papulis (1977) covering this case, guaranteeing perfect reconstruction for periodic uniform sampling if the sampling density is high enough (1 sample per each D-by-D square).



Uncontrolled rotation/scale/displacement.

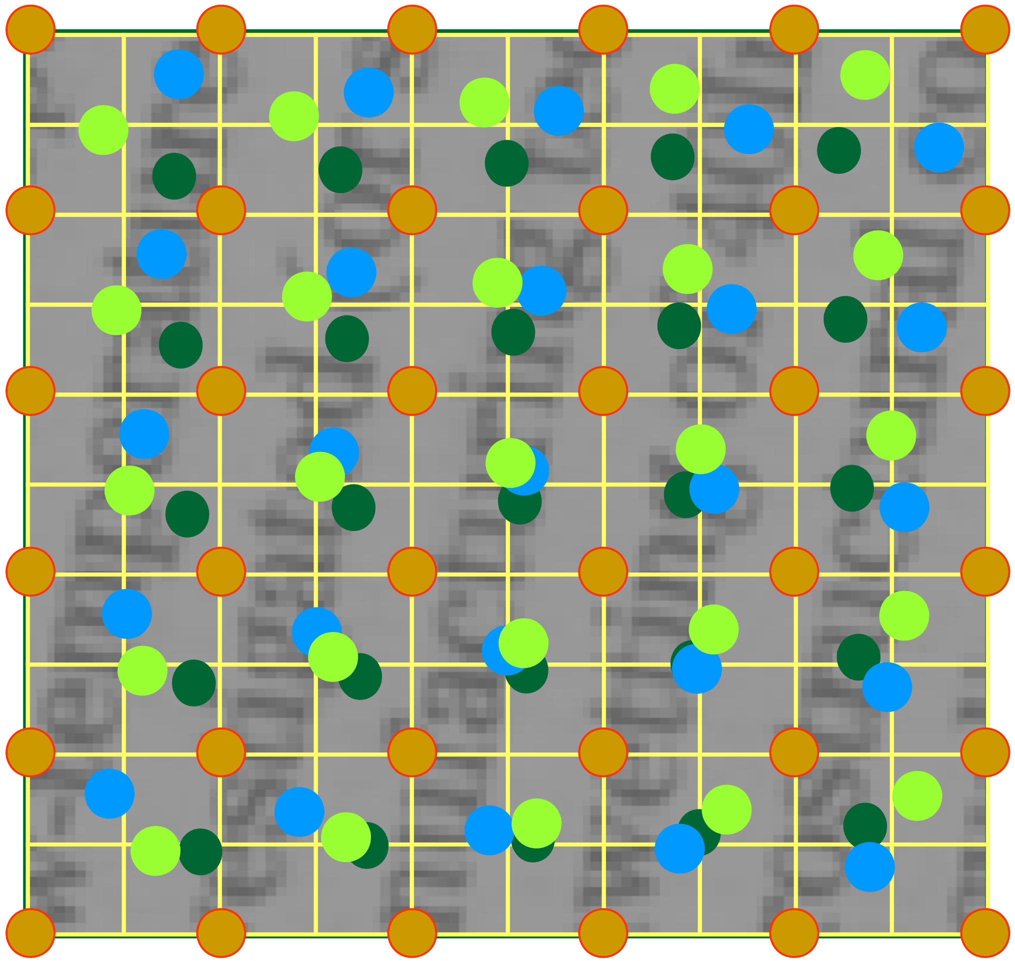
In the previous examples we restricted the camera to move horizontally/vertically parallel to the photograph object.

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



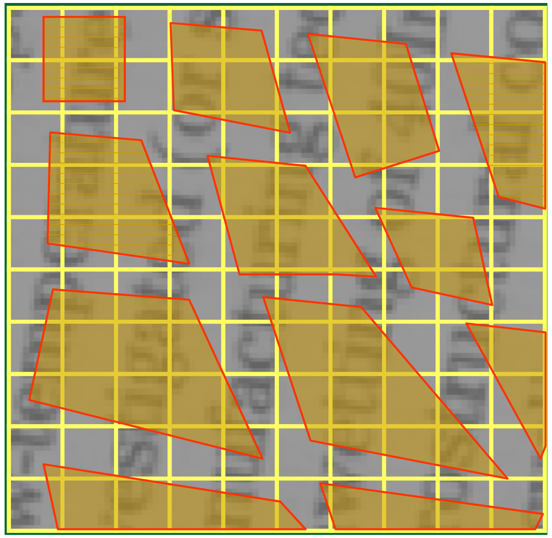
Uncontrolled rotation/scale/displacement.

So far, there is no sampling theorem covering this case!

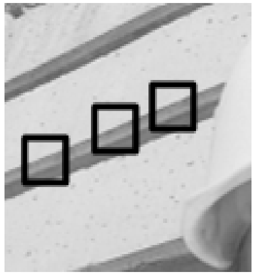
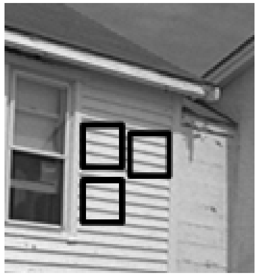
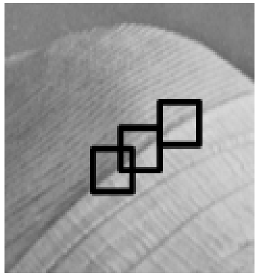


In practice, there are more difficulties …

- Sampling is not a point operation – there is a blur
- Motion may include perspective warp, local motion, etc.
- Samples may be noisy – any reconstruction process must take that into account.



Single frame super-resolution

- There are many repetitive patterns/structures in even one single natural image.
- For each given local patch, we can find many similar patches to it, which may not be in the local neighborhood of the given patch.
- Those “nonlocal” neighbors can be used to enhance the given patch.
- In terms of nonlocal, the ingle and multi-frame super-resolution can be unified.

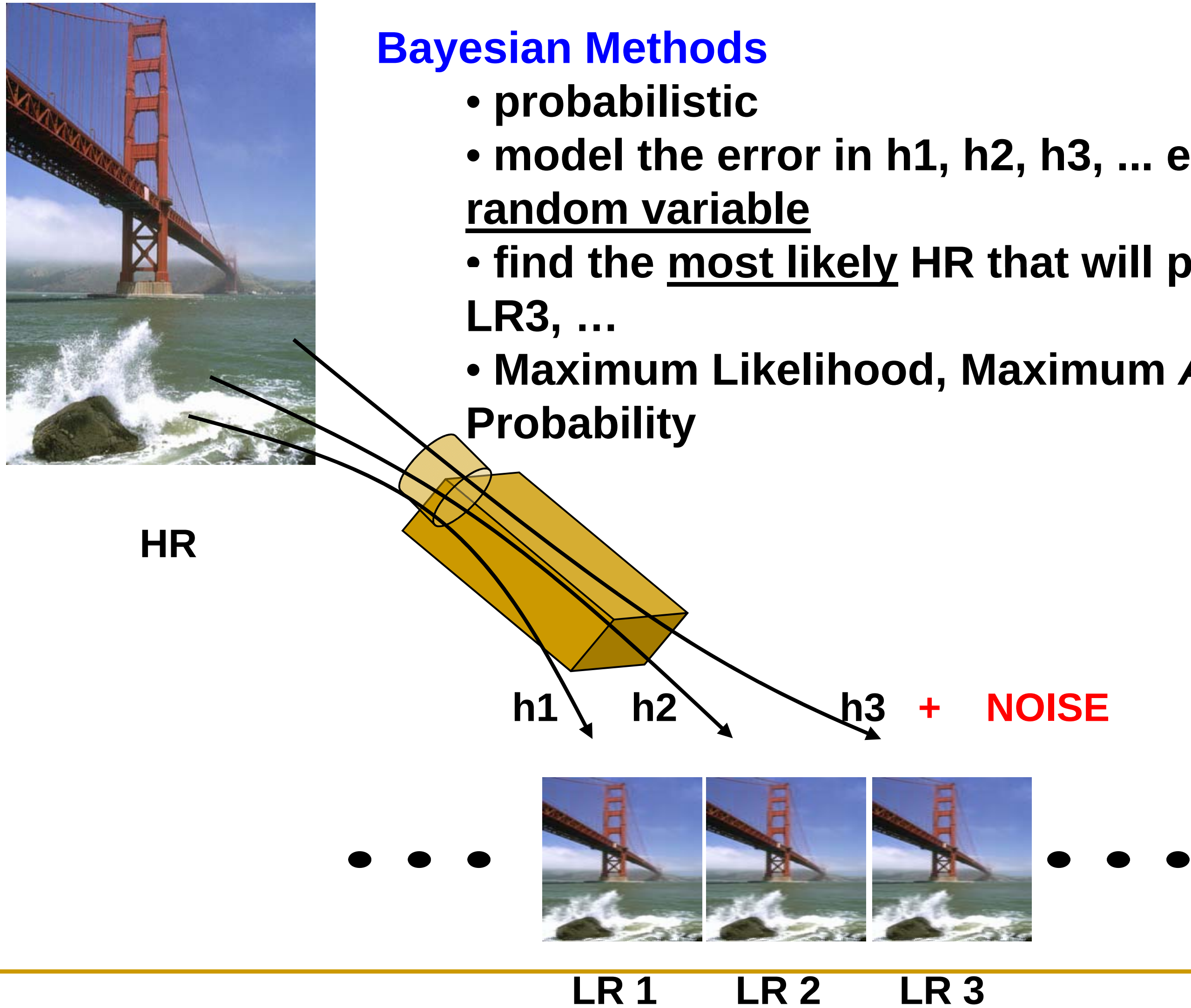
Applications of super-resolution

- Aerial/Satellite Imaging
- Medical Imaging
- High-Definition TV (HDTV) Displays
- Digital Camera
- Scanner resolution enhancement
- Extracting/Printing still image from video sequence
- Security/Surveillance systems, forensic science, ...

Super-resolution methods

Bayesian Methods

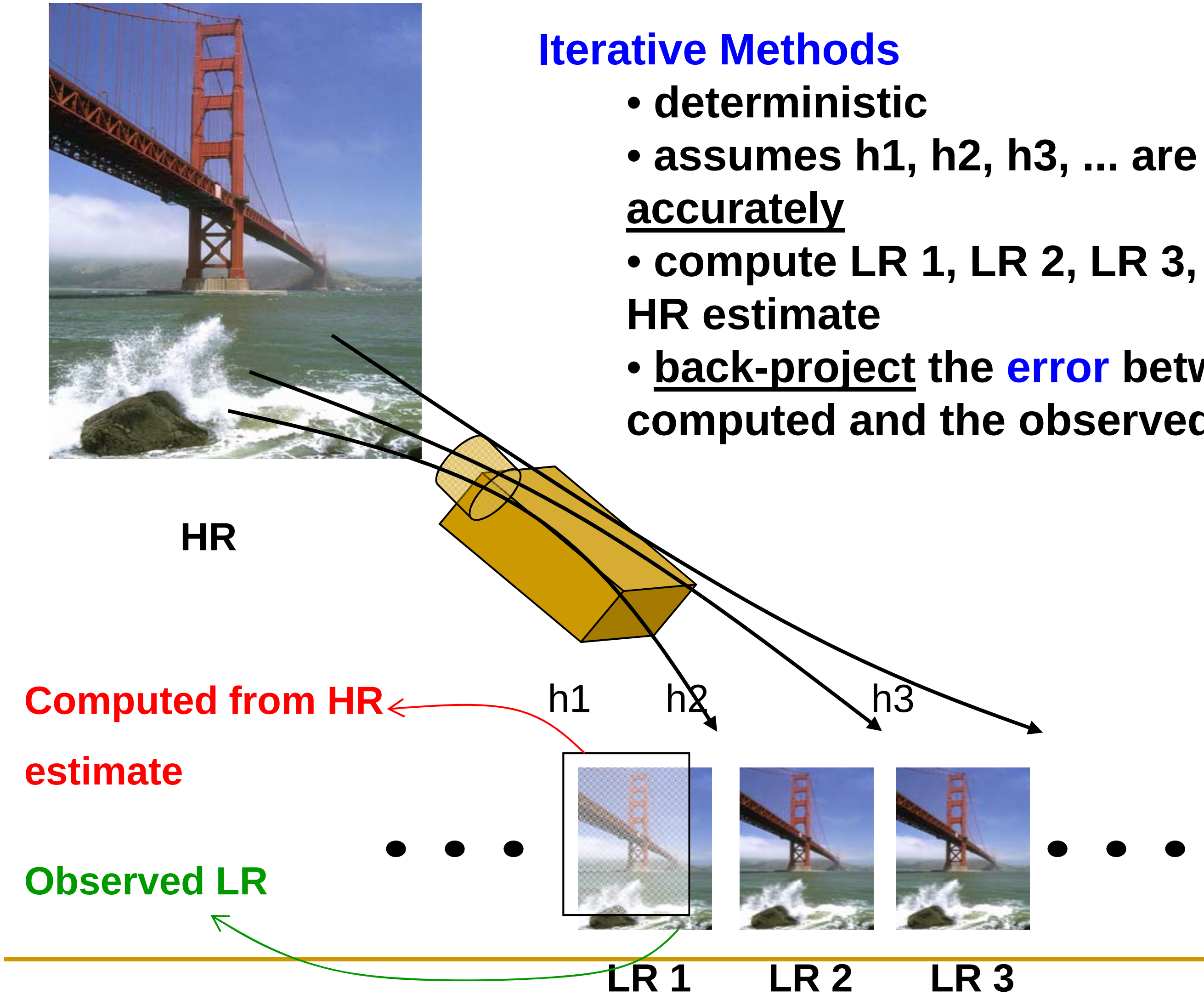
- **probabilistic**
- **model the error in h1, h2, h3, ... estimation as a <u>random variable</u>**
- **find the <u>most likely</u> HR that will produce LR1, LR2, LR3, …**
- **Maximum Likelihood, Maximum *A Posteriori* Probability**



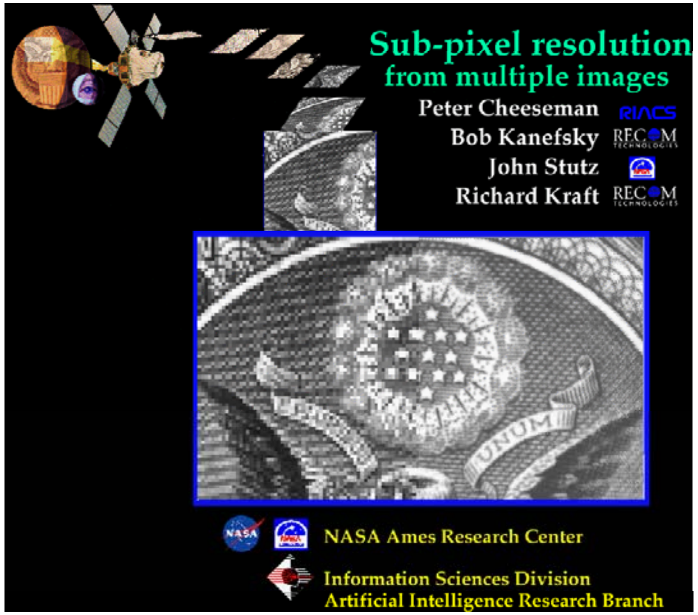
Super-resolution methods

Iterative Methods

- **deterministic**
- **assumes h1, h2, h3, ... are estimated <u>accurately</u>**
- **compute LR 1, LR 2, LR 3, … from an initial HR estimate**
- **<u>back-project</u> the error between the computed and the observed LR images**

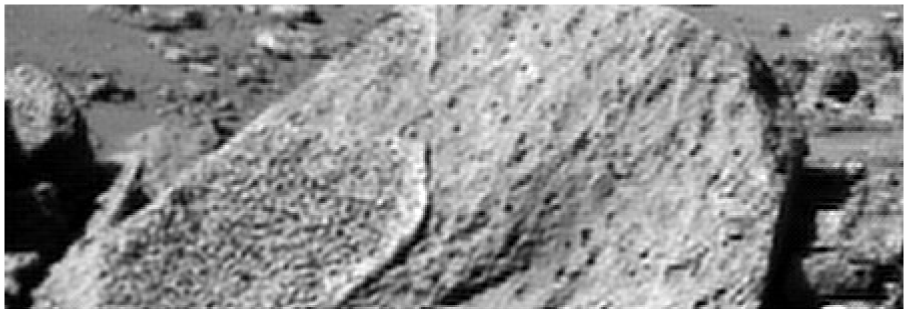


An example



- LR images sent from Mars Pathfinder

- Superresolution applied



Some comments on super-resolution

- Super-resolution is still an active research area. We still don't have fast and robust algorithms.
- Many theoretical problems are not solved yet.
- It's performance strongly depends on the set of images. If no enough information in the images, it doesn't work.
- Recently, the sparse representation methods have been successfully used in super-resolution, yet the complexity is very high.

References

- IEEE Signal Processing Magazine *special issue on Super-resolution Image Reconstruction*, vol. 20, no. 5, May 2003.
- http://www.cs.technion.ac.il/~elad/talks/2002/Super-Resolution_All.ppt