EL5123 --- Image Processing

Image Sampling and Resizing

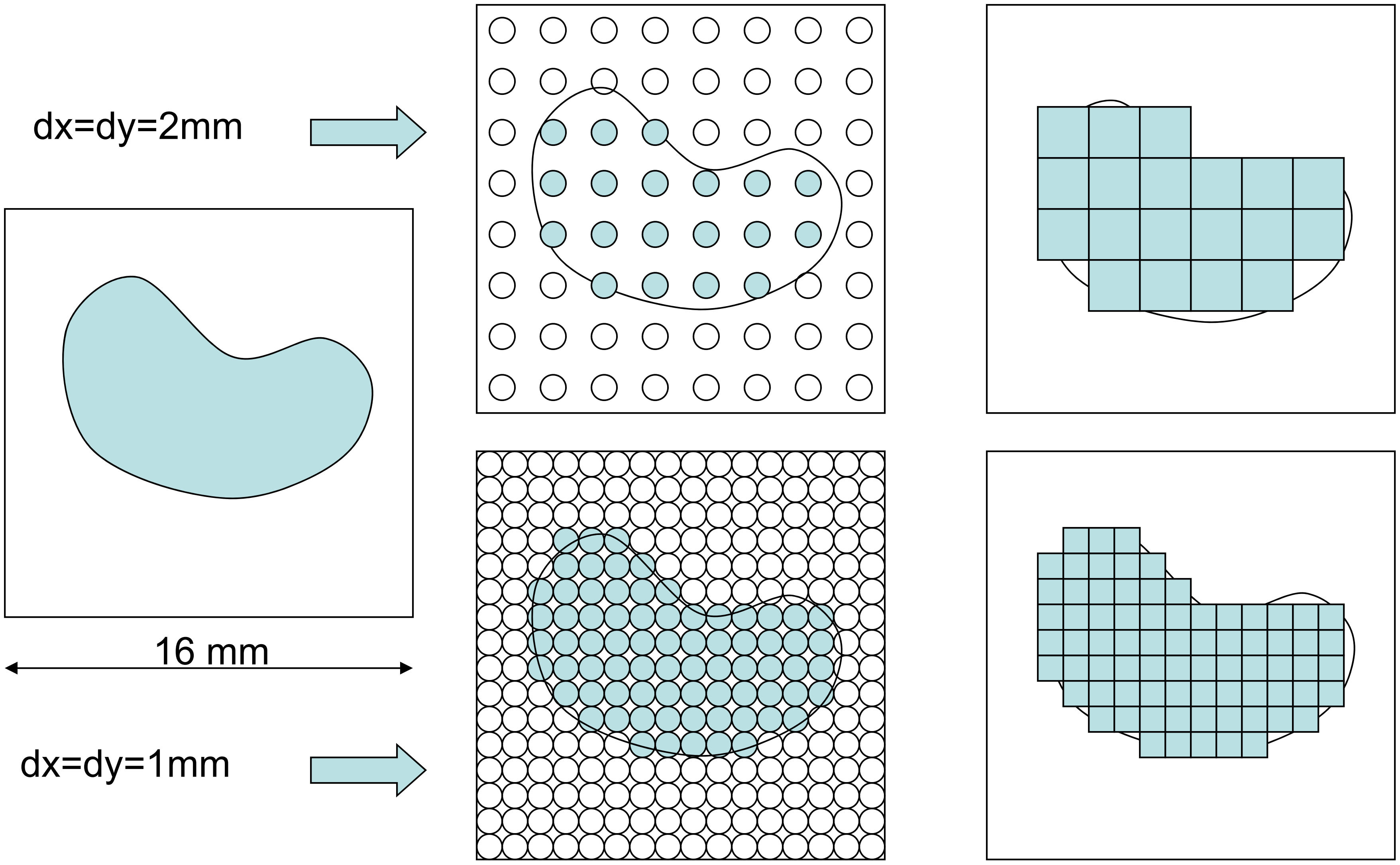
Yao Wang
Polytechnic Institute of NYU, Brooklyn, NY 11201

With contribution from Zhu Liu
Partly based on A. K. Jain, Fundamentals of Digital Image Processing

Lecture Outline

- Introduction
- Nyquist sampling and interpolation theorem
- Common sampling and interpolation filters
- Sampling rate conversion of discrete images (image resizing)

Illustration of Image Sampling and Interpolation



How to choose dx, dy to reach a good trade off between accuracy and cost of storage, transmission, processing?

Uniform Sampling

- *f(x,y)* represents the original continuous image, $f_s(m,n)$ the sampled image, and $\hat{f}(x,y)$ the reconstructed image.
- Uniform sampling

$$f_s(m,n) = f(m\Delta x, n\Delta y),$$
$$m = 0,\ldots,M-1; n = 0,\ldots,N-1.$$

  - Δx and Δy are vertical and horizontal sampling intervals. $f_{s,x}=1/\Delta x$, $f_{s,y}=1/\Delta y$ are vertical and horizontal sampling frequencies.

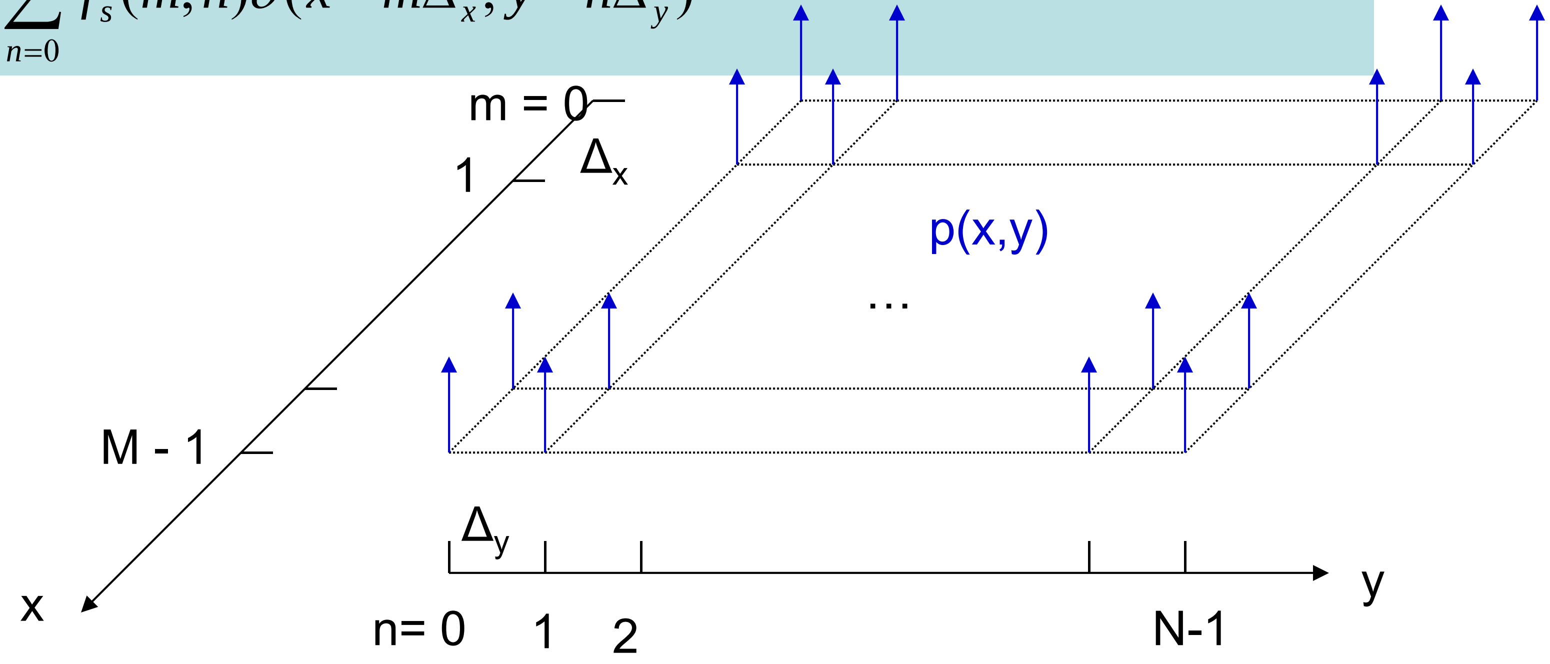
Image Sampling as Product with Impulse Train

- Periodic impulse sequence

$$p(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\delta(x-m\Delta_x,y-n\Delta_y)$$

$$\tilde{f}_s(x,y)=f(x,y)\times p(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m\Delta_x,n\Delta_y)\delta(x-m\Delta_x,y-n\Delta_y)$$

$$=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f_s(m,n)\delta(x-m\Delta_x,y-n\Delta_y)$$



Fourier Transform of Impulse Train

- 1D

$$p(t) = \sum_{m,n} \delta(t - n\Delta t) \Leftrightarrow P(u) = \frac{1}{\Delta t} \sum_{n} \delta(u - nf_s)$$

$$where \quad f_s = \frac{1}{\Delta t}$$

- 2D

$$p(x,y) = \sum_{m,n} \delta(x - m\Delta x, y - n\Delta y) \Leftrightarrow P(u,v) = \frac{1}{\Delta x \Delta y} \sum_{m,n} \delta(u - mf_{s,x}, v - nf_{s,y})$$

$$where \quad f_{s,x} = \frac{1}{\Delta x}, f_{s,y} = \frac{1}{\Delta y}$$

Frequency Domain Interpretation of Sampling

- Sampling is equivalent to multiplication of the original signal with a sampling pulse sequence.

$$f_s(x,y) = f(x,y)\,p(x,y)$$

$$\textit{where} \quad p(x,y) = \sum_{m,n} \delta(x - m\Delta x, y - n\Delta y)$$

- In frequency domain

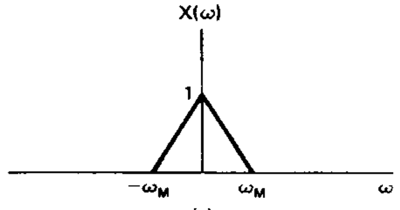
$$F_s(u,v) = F(u,v) * P(u,v)$$

$$P(u,v) = \frac{1}{\Delta x \Delta y} \sum_{m,n} \delta(u - m f_{s,x}, v - n f_{s,y}) \quad \Rightarrow \quad F_s(u,v) = \frac{1}{\Delta x \Delta y} \sum_{m,n} F(u - m f_{s,x}, v - n f_{s,y})$$

$$\textit{where} \quad f_{s,x} = \frac{1}{\Delta x},\ f_{s,y} = \frac{1}{\Delta y}$$

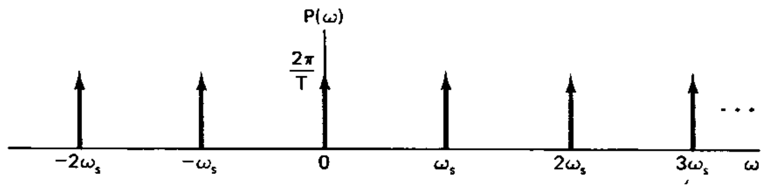
Frequency Domain Interpretation of Sampling in 1D

Original signal



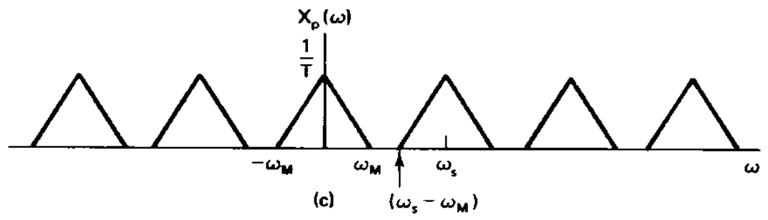
(a)

Sampling impulse train



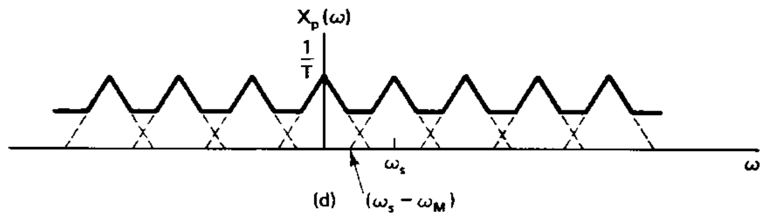
(b)

Sampled signal $f_s > 2f_m$



(c)

Sampled signal $f_s < 2f_m$ (Aliasing effect)



(d)

The spectrum of the sampled signal includes the original spectrum and its aliases (copies) shifted to $k\ f_s$, k=+/- 1,2,3,…

When $f_s < 2f_m$, aliases overlap with the original spectrum -> aliasing artifact

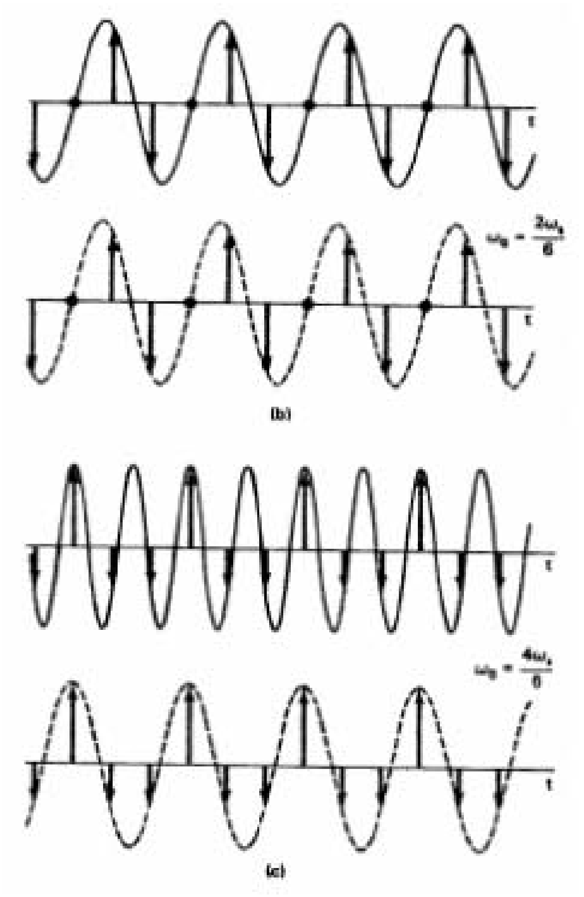
Sampling of 1D Sinusoid Signals

Sampling above Nyquist rate
$\omega_s=3\omega_m>\omega_{s0}$

Reconstructed =original

Sampling under Nyquist rate
$\omega_s=1.5\omega_m<\omega_{s0}$

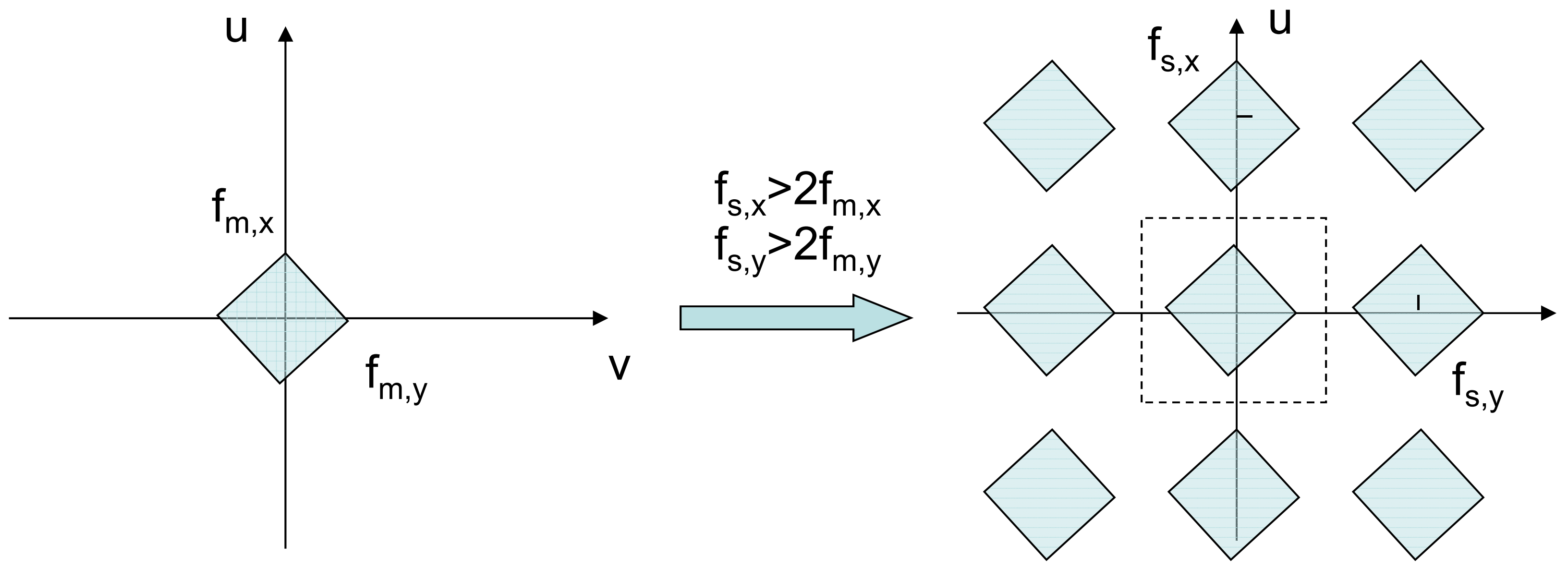
Reconstructed != original



Aliasing: The reconstructed sinusoid has a lower frequency than the original!

Frequency Domain Interpretation of Sampling in 2D

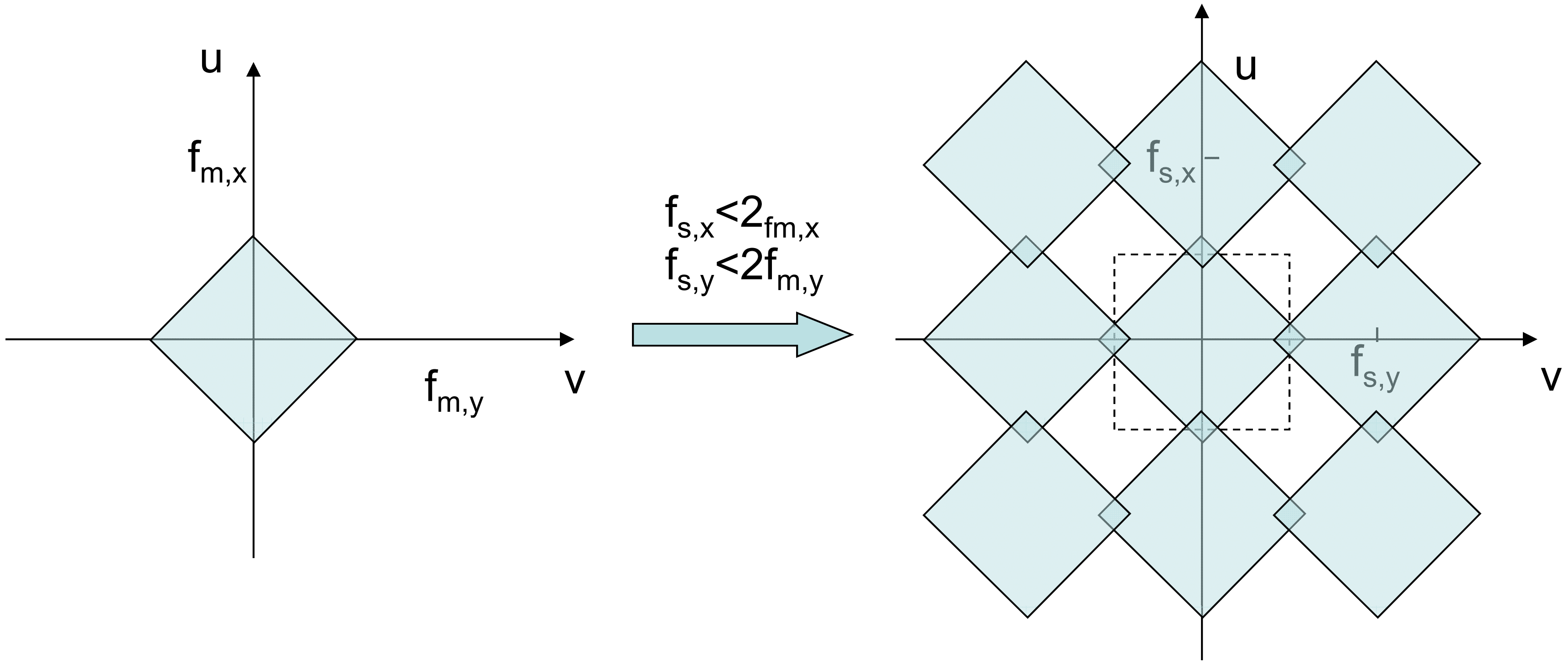
- The sampled signal contains replicas of the original spectrum shifted by multiples of sampling frequencies.



Original spectrum F(u,v)

Sampled spectrum $F_s(u,v)$

Illustration of Aliasing Phenomenon



Original spectrum F(u,v)

Sampled spectrum $F_s(u,v)$

Nyquist Sampling Rate

- Nyquist sampling rate
  - To be able to preserve the original signal in the sampled signals, these aliasing components should not overlap with the original one. This requires that the sampling frequency $f_{s,x}$, $f_{s,y}$ must be at least twice of the highest frequency of the signal, known as *Nyquist sampling rate*.
- Interpolation
  - Remove all the aliasing components

Nyquist Sampling and Reconstruction Theorem

- A band-limited image with highest frequencies at $f_{m,x}$, $f_{m,y}$ can be reconstructed perfectly from its samples, provided that the sampling frequencies satisfy: $f_{s,x} > 2f_{m,x}$, $f_{s,y} > 2f_{m,y}$
- The reconstruction can be accomplished by the ideal low-pass filter with cutoff frequency at $f_{c,x} = f_{s,x}/2$, $f_{c,y} = f_{s,y}/2$, with magnitude $\Delta x \Delta y$.

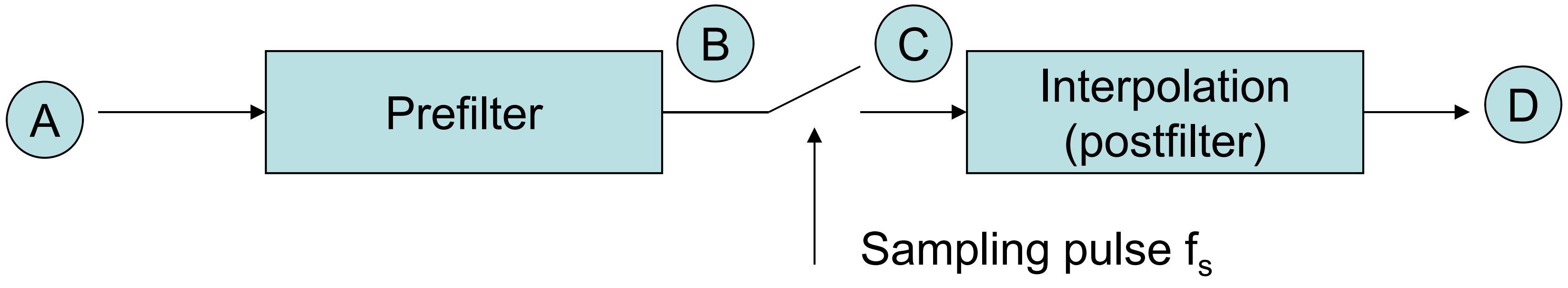
$$H(u,v) = \begin{cases} \Delta x \Delta y & |u| \le \frac{f_{s,x}}{2}, |v| \le \frac{f_{s,y}}{2} \\ 0 & otherwise \end{cases} \quad \Leftrightarrow \quad h(x,y) = \frac{\sin \pi f_{s,x} x}{\pi f_{s,x} x} \cdot \frac{\sin \pi f_{s,y} y}{\pi f_{s,y} y}$$

- The interpolated image

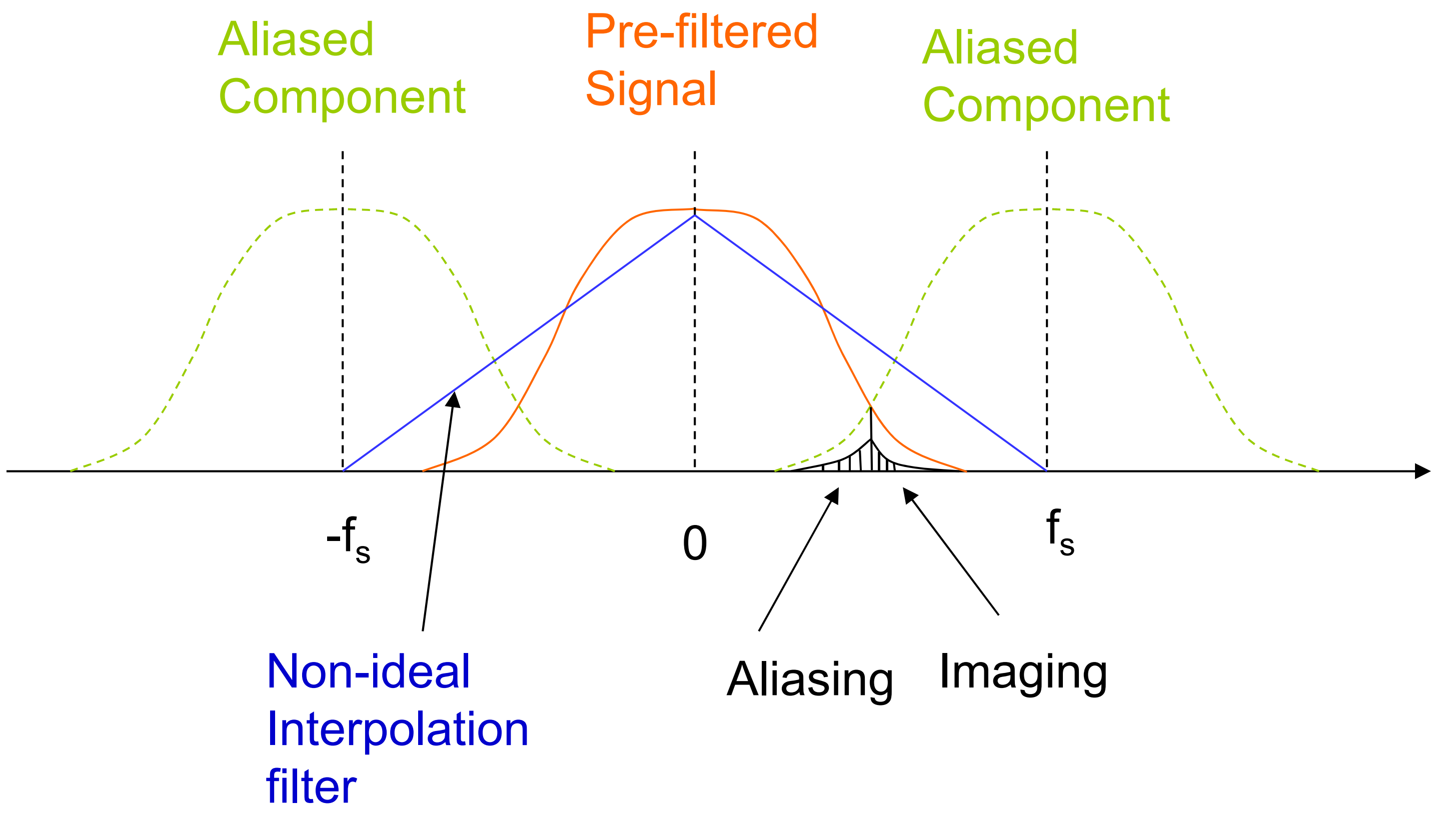
$$\hat{f}(x,y) = \sum_m \sum_n f_s(m,n) \frac{\sin \pi f_{s,x}(x - m\Delta x)}{\pi f_{s,x}(x - m\Delta x)} \frac{\sin \pi f_{s,y}(y - m\Delta y)}{\pi f_{s,y}(y - m\Delta y)}$$

Applying Nyquist Theorem

- Two issues
  - The signals are not bandlimited.
    - A bandlimiting filter with cutoff frequency $f_c=f_s/2$ needs to be applied before sampling. This is called *prefilter* or *sampling filter*.
  - The sinc filter is not realizable.
    - Shorter, finite length filters are usually used in practice for both prefilter and interpolation filter.
- A general paradigm



Non-ideal Sampling and Interpolation



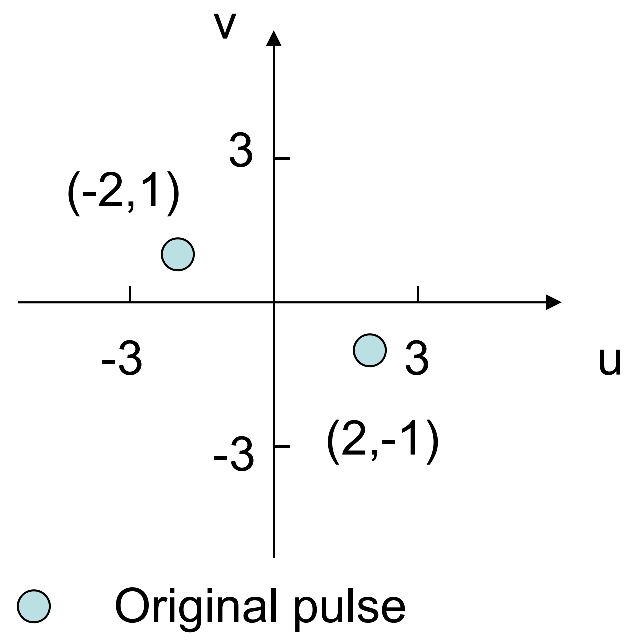
Non ideal prefiltering causes Aliasing
Non ideal interpolation filter causes Imaging

Sampling a Sinusoidal Signal

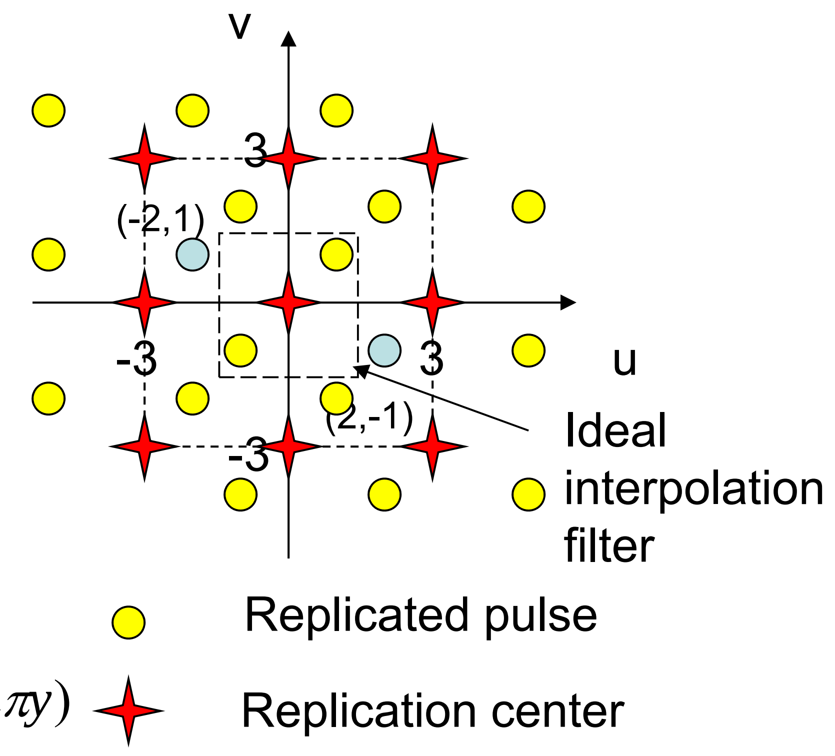
$$f(x,y) = \cos(4\pi x - 2\pi y) \quad \Leftrightarrow \quad F(u,v) = \frac{1}{2}\left[\delta(u-2, v+1) + \delta(u+2, v-1)\right]$$

Sampled at $\Delta x = \Delta y = 1/3 \quad f_{s,x} = f_{s,y} = 3$

Original Spectrum

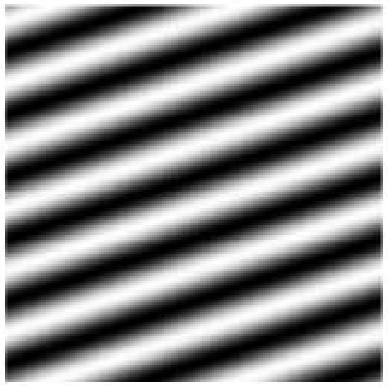


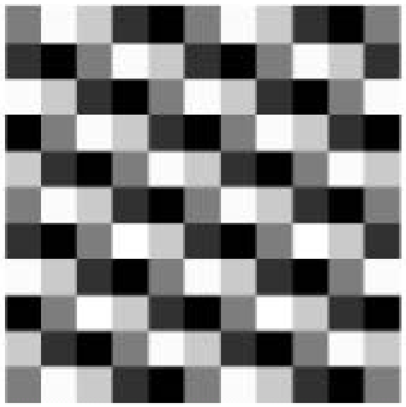
Sampled Spectrum



$$\hat{f}(x,y) = \cos(2\pi x + 2\pi y)$$

Sampling in 2D: Sampling a 2D Sinusoidal Pattern



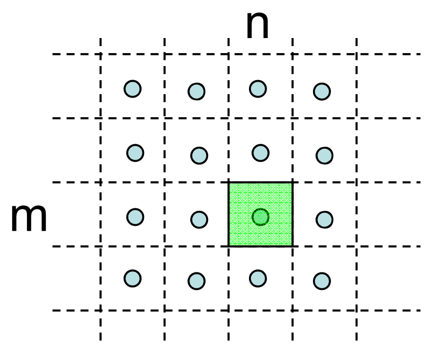


f(x,y)=sin(2*π*(3x+y))
Sampling: dx=0.01,dy=0.01
Satisfying Nyquist rate
$f_{x,max}=3$, $f_{y,max}=1$
$f_{s,x}=100>6$, $f_{s,y}=100>2$

f(x,y)=sin(2*π*(3x+y))
Sampling: dx=0.2,dy=0.2
(Displayed with pixel replication)
Sampling at a rate lower than Nyquist rate

A Simple Prefilter – Averaging Filter

- Each sampling value is the mean value of the original continuous function in a rectangular region of dimension Δx and Δy, i.e:



$$f_s(m,n) = \frac{1}{\Delta x \Delta y} \int_{(x,y)\in D_{m,n}} f(x,y)\,dx\,dy$$

where

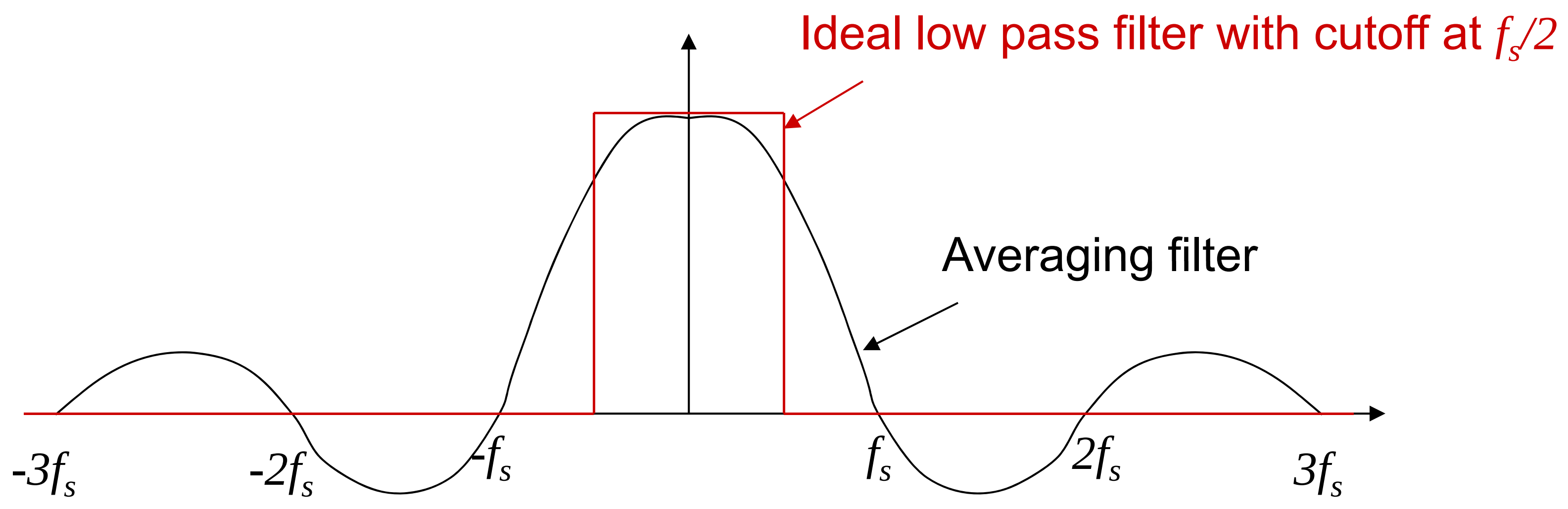
$$D_{m,n} = \{(m-1/2)\Delta x < x < (m+1/2)\Delta x, (n-1/2)\Delta y < y < (n+1/2)\Delta y\}$$

The equivalent prefilter is

$$h(x,y) = \begin{cases} \dfrac{1}{\Delta x \Delta y} & |x| < \Delta x/2, |y| < \Delta y/2 \\ 0 & otherwise \end{cases} \Leftrightarrow H(u,v) = \Delta x \Delta y \frac{\sin \pi \Delta x u}{\pi \Delta x u} \frac{\sin \pi \Delta y v}{\pi \Delta y v}$$

How good is the averaging Filter?

- Look at its frequency response, how far is it from the ideal low pass filter



Reconstruction of continuous image from Samples

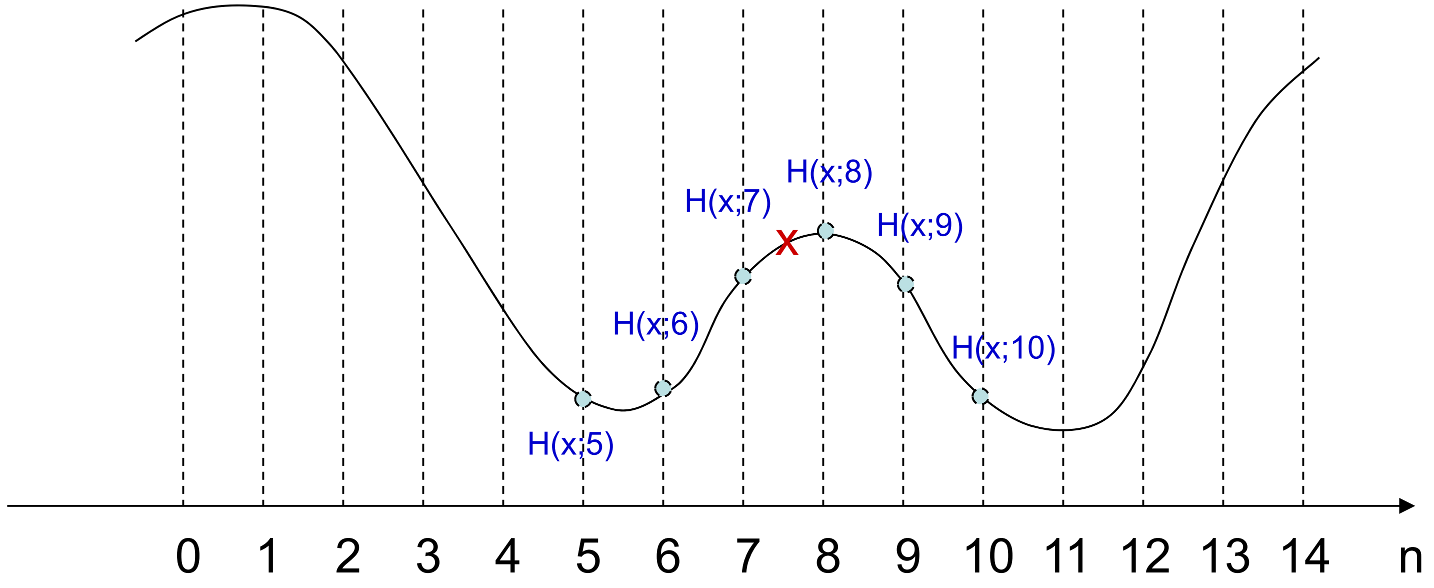
- The interpolation problem
- Interpretation as weighted average of sample values
- Interpretation as filtering
- What filter should we use?
  - Ideal interpolation filter: remove the replicated spectrum using ideal low-pass filter with cutoff frequency at fs/2 (the sinc function)

$$\hat{f}(x,y) = \sum_m \sum_n f_s(m,n) \frac{\sin \pi f_{s,x}(x - m\Delta x)}{\pi f_{s,x}(x - m\Delta x)} \frac{\sin \pi f_{s,y}(y - n\Delta y)}{\pi f_{s,y}(y - n\Delta y)}$$

Interpretation as a weighted average of sample values

- The value of a function at arbitrary point (x, y) is estimated from a weighted sum of its sample values in the neighborhood of ([x/Δx], [y/Δy]):
  - Let h(x,y; m,n) specifies the weight assigned to sample m,n, when determining the image value at x,y

$$\hat{f}(x,y)=\sum_{m,n} h(x,y;m,n)\,f_s(m,n)$$



Desirable Properties of the Weight Function

- The weighting function h(x,y;m,n) should depend only on distance between (x,y) and the spatial location of (m,n), i.e.

$$h(x, y; m, n) = h(x - m\Delta x, y - n\Delta y).$$

- Should be a decreasing function of the distance
  - Higher weight for nearby samples
- Should be an even function of the distance
  - Left neighbor and right neighbor of same distance have the same weight
  - $h1(x)=h1(-x)$
- Generally Separable:
  - h2(x,y)=h1(x) h1(y)
- To retain the original sample values, should have
  - h(0,0)=1, h(mΔx,nΔy)=0

Interpretation as Filtering

- The weighted average operation is equivalent to filtering fs(x,y) with h(x,y)
- Usually, h(x,y) is separable h(x,y) = $h_x(x)h_y(y)$
- To retain the original sample values, should have
  - h(0,0)=1, h(mΔx,nΔy)=0
  - Nyquist filter

$$\hat{f}(x,y)=\sum_{m,n} h(x-m\Delta x, y-n\Delta y) f_s(m,n)$$

$$\tilde{f}_s(x,y)=\sum_{m,n} f_s(m,n)\delta(x-m\Delta x, y-n\Delta y)$$

$$h(x,y)*\tilde{f}_s(x,y)$$

$$=\int h(x-\alpha, y-\beta)\,\tilde{f}_s(\alpha,\beta)d\alpha d\beta$$

$$=\int h(x-\alpha, y-\beta)\sum_{m,n} f_s(m,n)\delta(\alpha-m\Delta x, \beta-n\Delta y)d\alpha d\beta$$

$$=\sum_{m,n} f_s(m,n)\int h(x-\alpha, y-\beta)\delta(\alpha-m\Delta x, \beta-n\Delta y)d\alpha d\beta$$

$$=\sum_{m,n} f_s(m,n)h(x-m\Delta x, y-n\Delta y)$$

$$\Rightarrow \quad \hat{f}(x,y)=h(x,y)*\tilde{f}_s(x,y)$$

A Simple Interpolation Filter: Sample-And-Hold (pixel replication)

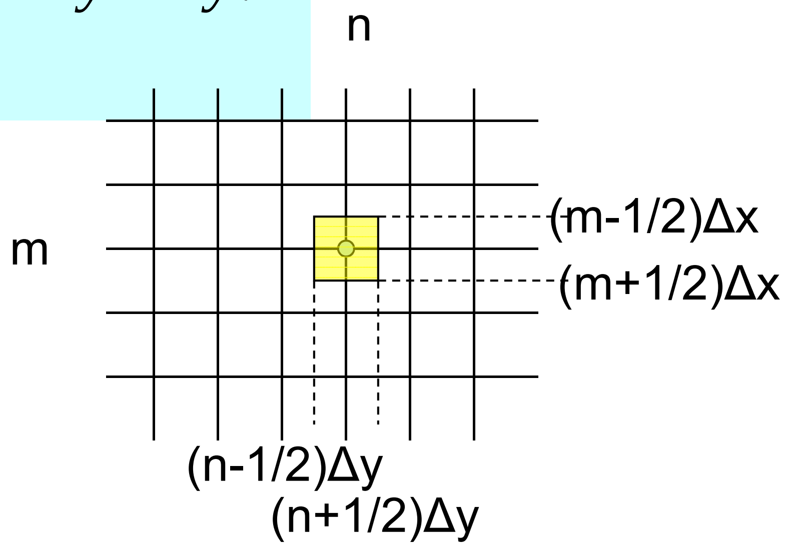
- The interpolated value at a point is obtained from that of its nearest sample

$$\hat{f}(x,y) = f_s(m,n) \quad (m-1/2)\Delta x \le x < (m+1/2)\Delta x, (n-1/2)\Delta y \le y < (n+1/2)\Delta y$$

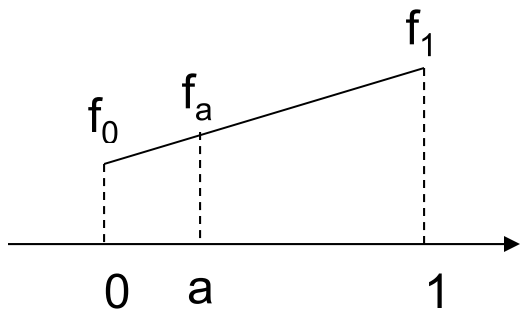
- Corresponding interpolation filter is

$$h(x,y) = \begin{cases} 1 & -\Delta x/2 \le x < \Delta x/2, -\Delta y/2 \le y < \Delta y/2 \\ 0 & otherwise \end{cases}$$

0th order interpolation filter

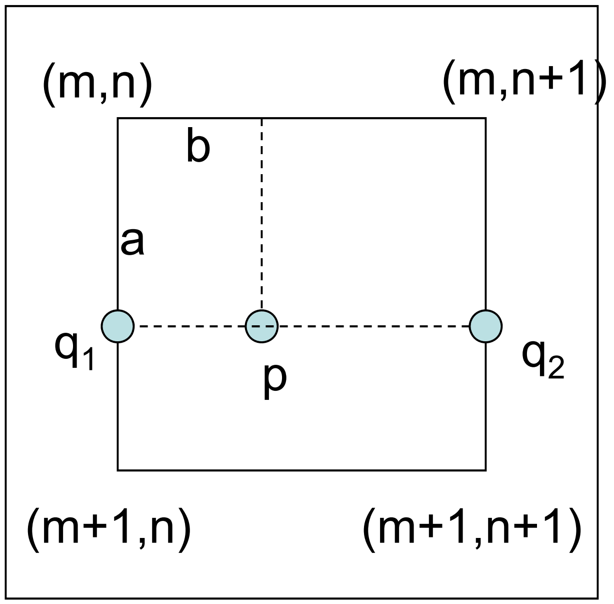


Bilinear Filter



$$f_a = (1-a) f_0 + a f_1$$

1D Linear interpolation



Step 1:

$$f(q_1) = (1-a) f(m,n) + af(m+1,n)$$
$$f(q_2) = (1-a) f(m,n+1) + af(m+1,n+1)$$

Step 2:

$$f(p) = (1-b) f(q_1) + bf(q_2)$$

2D bilinear interpolation

Corresponding interpolation filter

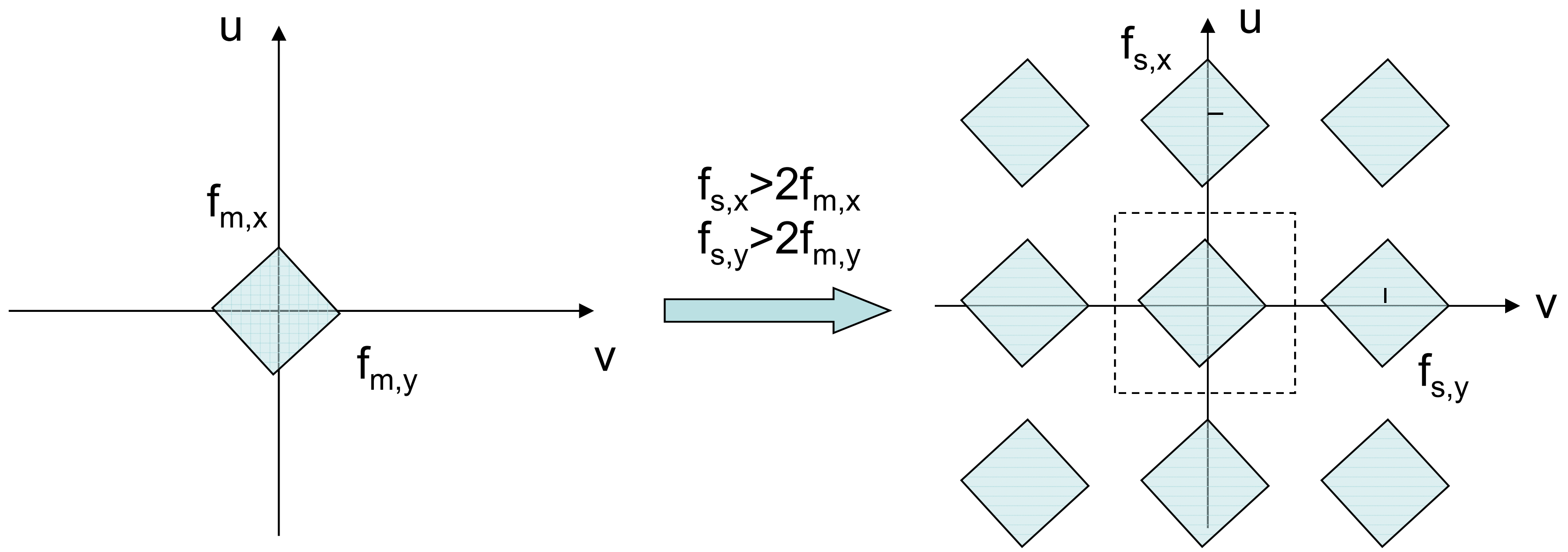
$$h(x,y) = \begin{cases} \left(1-\frac{|x|}{\Delta x}\right)\left(1-\frac{|y|}{\Delta y}\right) & -\Delta x \le x < \Delta x, -\Delta y \le y < \Delta y \\ 0 & otherwise \end{cases}$$

Which filter is better?

- Recall what happened in the frequency domain when we sample an image
- Ideal filter: ½ band ideal low pass filter
- Quantitatively we can evaluate how far is the filter from the ideal filter
- But we should also look at visual artifacts

Frequency Domain Interpretation of Sampling in 2D

- The sampled signal contains replicas of the original spectrum shifted by multiples of sampling frequencies.



Original spectrum F(u,v)

Sampled spectrum $F_s(u,v)$

Ideal Interpolation Filter

- The ideal interpolation filter should be a low-pass filter with cutoff frequency at $f_{c,x} = f_{s,x}/2$, $f_{c,y} = f_{s,y}/2$, with magnitude $\Delta x \Delta y$

$$H(u,v) = \begin{cases} \Delta x \Delta y & |u| \le \frac{f_{s,x}}{2}, |v| \le \frac{f_{s,y}}{2} \\ 0 & otherwise \end{cases} \quad \Leftrightarrow \quad h(x,y) = \frac{\sin \pi f_{s,x} x}{\pi f_{s,x} x} \cdot \frac{\sin \pi f_{s,y} y}{\pi f_{s,y} y}$$

The sinc filter

- The interpolated image

Weight function h(x,y;m,n)

$$\hat{f}(x,y) = \sum_m \sum_n f_s(m,n) \underbrace{\frac{\sin \pi f_{s,x}(x - m\Delta x)}{\pi f_{s,x}(x - m\Delta x)} \frac{\sin \pi f_{s,y}(y - n\Delta y)}{\pi f_{s,y}(y - n\Delta y)}}$$

Comparison of Different Interpolation Filters

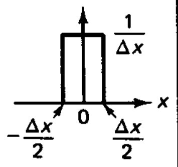
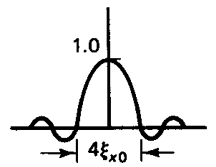
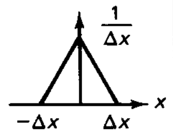
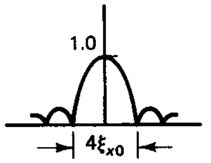
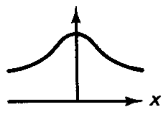
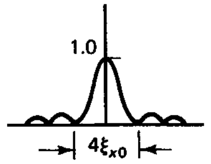
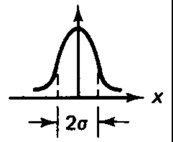
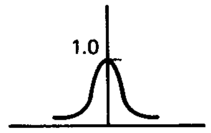
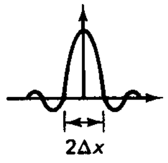
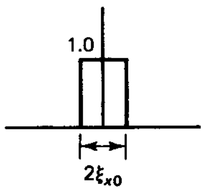
One-dimensional interpolation function	Diagram	Definition $p(x)$	Two-dimensional interpolation function $p_d(x, y) = p(x)p(y)$	Frequency response $P_d(\xi_1, \xi_2)$	$P_d(\xi_1, 0)$
Rectangle (zero-order hold) ZOH $p_o(x)$	 $\frac{1}{\Delta x}$ x $-\frac{\Delta x}{2}$ 0 $\frac{\Delta x}{2}$	$\frac{1}{\Delta x}\,\text{rect}\left(\frac{x}{\Delta x}\right)$	$p_o(x)p_o(y)$	$\text{sinc}\left(\frac{\xi_1}{2\xi_{x0}}\right)\text{sinc}\left(\frac{\xi_2}{2\xi_{y0}}\right)$	
Triangle (first-order hold) FOH $p_1(x)$		$\frac{1}{\Delta x}\,\text{tri}\left(\frac{x}{\Delta x}\right)$ $p_o(x) \circledast p_o(x)$	$p_1(x)p_1(y)$	$\left[\text{sinc}\left(\frac{\xi_1}{2\xi_{x0}}\right)\text{sinc}\left(\frac{\xi_2}{2\xi_{y0}}\right)\right]^2$	
nth-order hold $n = 2$, quadratic $n = 3$, cubic splines $p_n(x)$		$p_o(x) \circledast \cdots \circledast p_o(x)$ n convolutions	$p_n(x)p_n(y)$	$\left[\text{sinc}\left(\frac{\xi_1}{\xi_{x0}}\right)\text{sinc}\left(\frac{\xi_2}{\xi_{y0}}\right)\right]^{n+1}$	
Gaussian $p_g(x)$		$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{x^2}{2\sigma^2}\right]$	$\frac{1}{2\pi\sigma^2}\exp\left[-\frac{(x^2+y^2)}{2\sigma^2}\right]$	$\exp\left[-2\pi^2\sigma^2(\xi_1^2+\xi_2^2)\right]$	
Sinc		$\frac{1}{\Delta x}\,\text{sinc}\left(\frac{x}{\Delta x}\right)$	$\frac{1}{\Delta x \Delta y}\,\text{sinc}\left(\frac{x}{\Delta x}\right)\text{sinc}\left(\frac{x}{\Delta y}\right)$	$\text{rect}\left(\frac{\xi_1}{2\xi_{x0}}\right)\text{rect}\left(\frac{\xi_2}{2\xi_{y0}}\right)$	

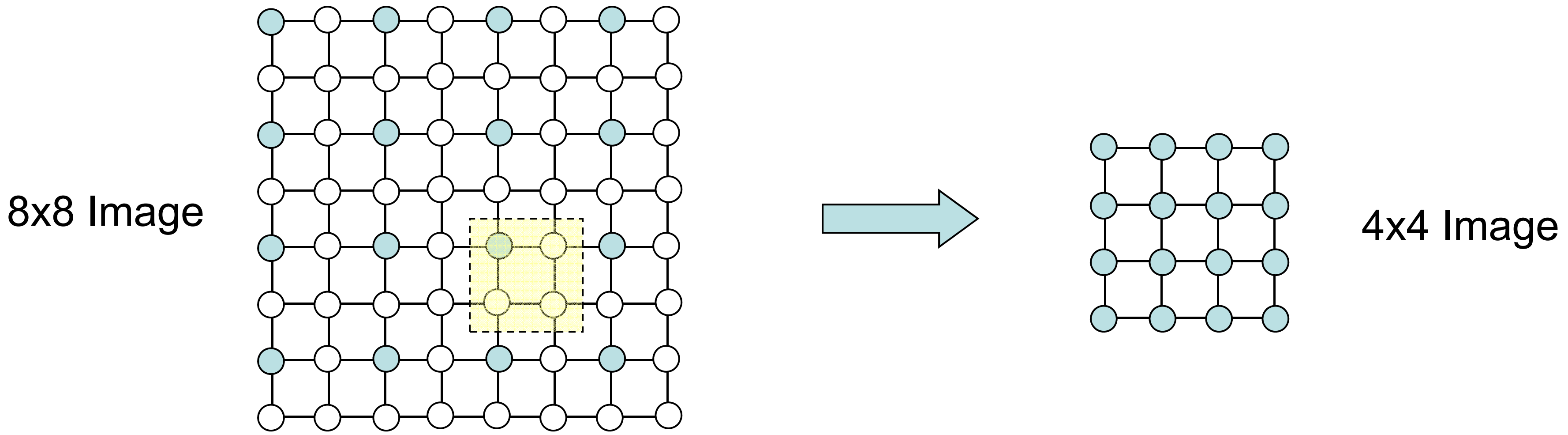
Image Resizing

- Image resizing:
  - Enlarge or reduce the image size (number of pixels)
  - Equivalent to
    - First reconstruct the continuous image from samples
    - Then Resample the image at a different sampling rate
  - Can be done w/o reconstructing the continuous image explicitly
- Image down-sampling (resample at a lower rate)
  - Spatial domain view
  - Frequency domain view: need for prefilter
- Image up-sampling (resample at a higher rate)
  - Spatial domain view
  - Different interpolation filters
    - Nearest neighbor, Bilinear, Bicubic

Image Down-Sampling

- Example:
  - reduce a 512x512 image to 256x256 = factor of 2 downsampling in both horizontal and vertical directions
  - In general, we can down-sample by an arbitrary factor in the horizontal and vertical directions
- How should we obtain the smaller image ?

Down Sampling by a Factor of Two



- Without Pre-filtering (simple approach)

$$f_d(m,n) = f(2m,2n)$$

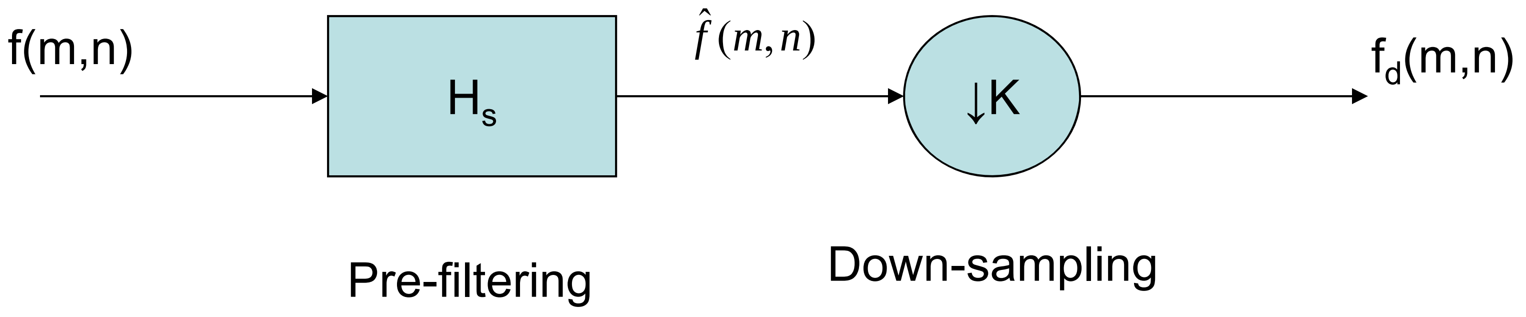
- Averaging Filter

$$f_d(m,n) = [f(2m,2n) + f(2m,2n+1) + f(2m+1,2n) + f(2m+1,2n+1)]/4$$

Problem of Simple Approach

- Aliasing if the effective sampling rate is below the Nyquist sample rate = 2 * highest frequency in the original continuous signal
- We need to prefilter the signal before down-sampling
- Ideally the prefilter should be a low-pass filter with a cut-off frequency half of the new sampling rate.
  - In digital frequency of the original sampled image, the cutoff frequency is ¼.
- In practice, we may use simple averaging filter

Down Sampling by a Factor of K

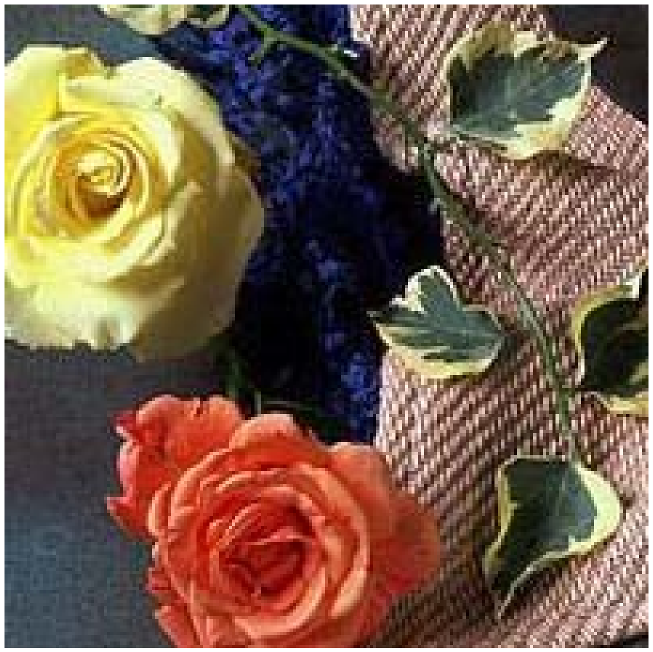


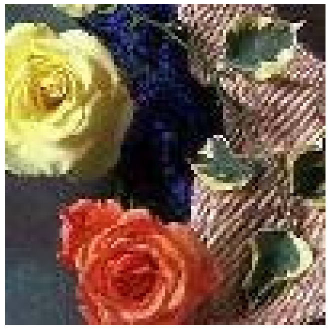
$$f_d(m,n) = \hat{f}(Km, Kn)$$

For factor of K down sampling, the prefilter should be low pass filter with cutoff at fs/(2K), if fs is the original sampling frequency

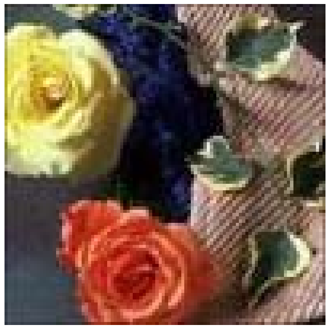
In terms of digital frequency, the cutoff should be 1/(2K)

Example: Image Down-Sample





Without prefiltering



With prefiltering (no aliasing, but blurring!)

Down-Sampling Using Matlab

- Without prefiltering
  - If f(,) is an MxN image, down-sampling by a factor of K can be done simply by

```
>> g=f(1:K:M,1:K:N)
```

- With prefiltering
  - First convolve the image with a desired filter
    - Low pass filter with digital cutoff frequency 1/(2K)
      - In matlab, 1/2 is normalized to 1
  - Then subsample

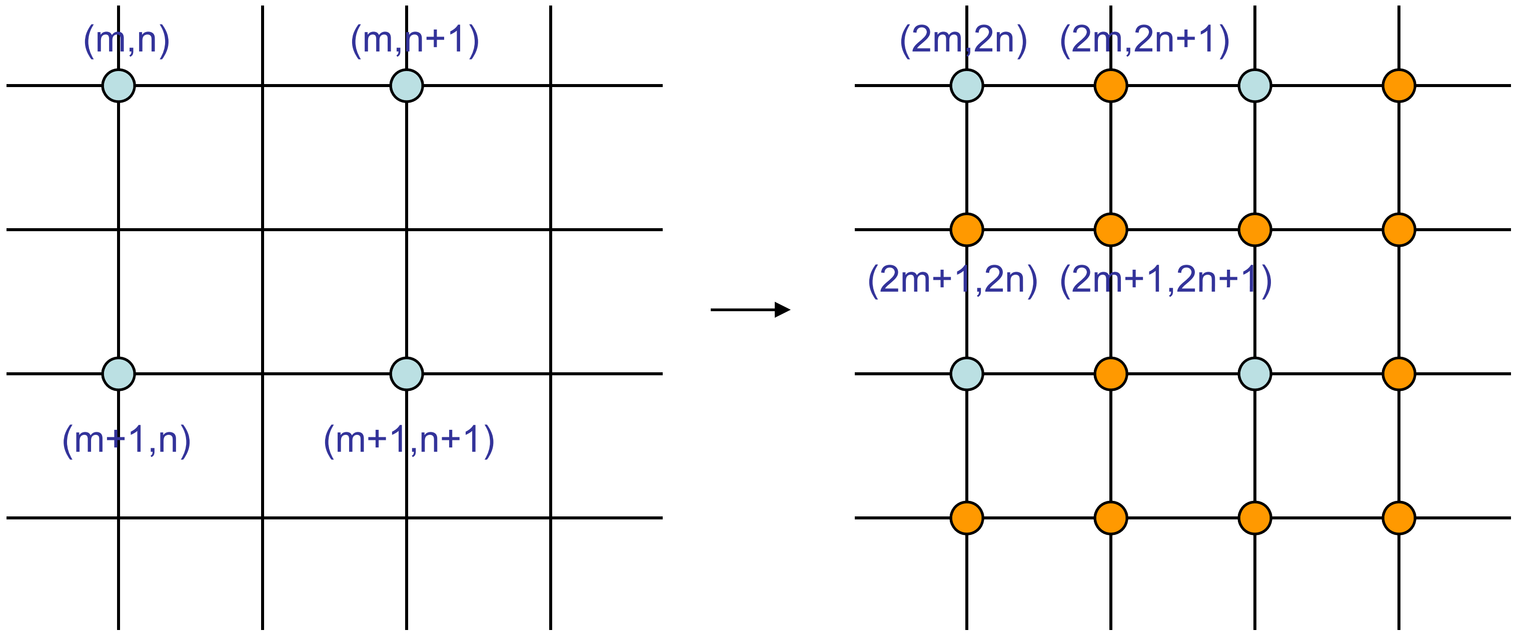
```
>> h=fir1(N, 1/K)
   %design a lowpass filter with cutoff at 1/K.
>> fp=conv2(f,h)
>> g=fp(1:K:M,1:K:N)
```

Image Up-Sampling

- Produce a larger image from a smaller one
  - Eg. 512x512 -> 1024x1024
  - More generally we may up-sample by an arbitrary factor L
- Questions:
  - How should we generate a larger image?
  - Does the enlarged image carry more information?
- Connection with interpolation of a continuous image from discrete image
  - First interpolate to continuous image, then sampling at a higher sampling rate, L*fs
  - Can be realized with the same interpolation filter, but only evaluate at x=mΔx', y=nΔy', Δx'=Δx/L, Δy'=Δy/L
  - Ideally using the sinc filter!

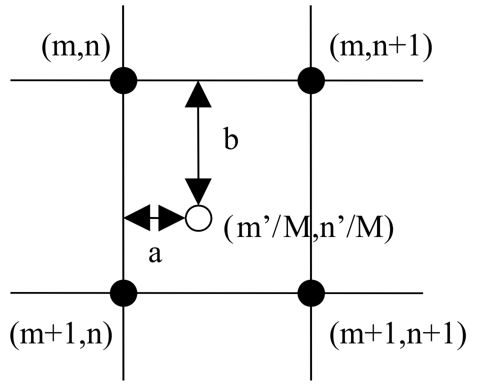
$$\hat{f}(x,y)=\sum_m \sum_n f_s(m,n)\frac{\sin \pi f_{s,x}(x-m\Delta x)}{\pi f_{s,x}(x-m\Delta x)}\frac{\sin \pi f_{s,y}(y-m\Delta y)}{\pi f_{s,y}(y-m\Delta y)}$$

Example: Factor of 2 Up-Sampling



Green samples are retained in the interpolated image;
Orange samples are estimated from surrounding green samples.

Nearest Neighbor Interpolation (pixel replication)



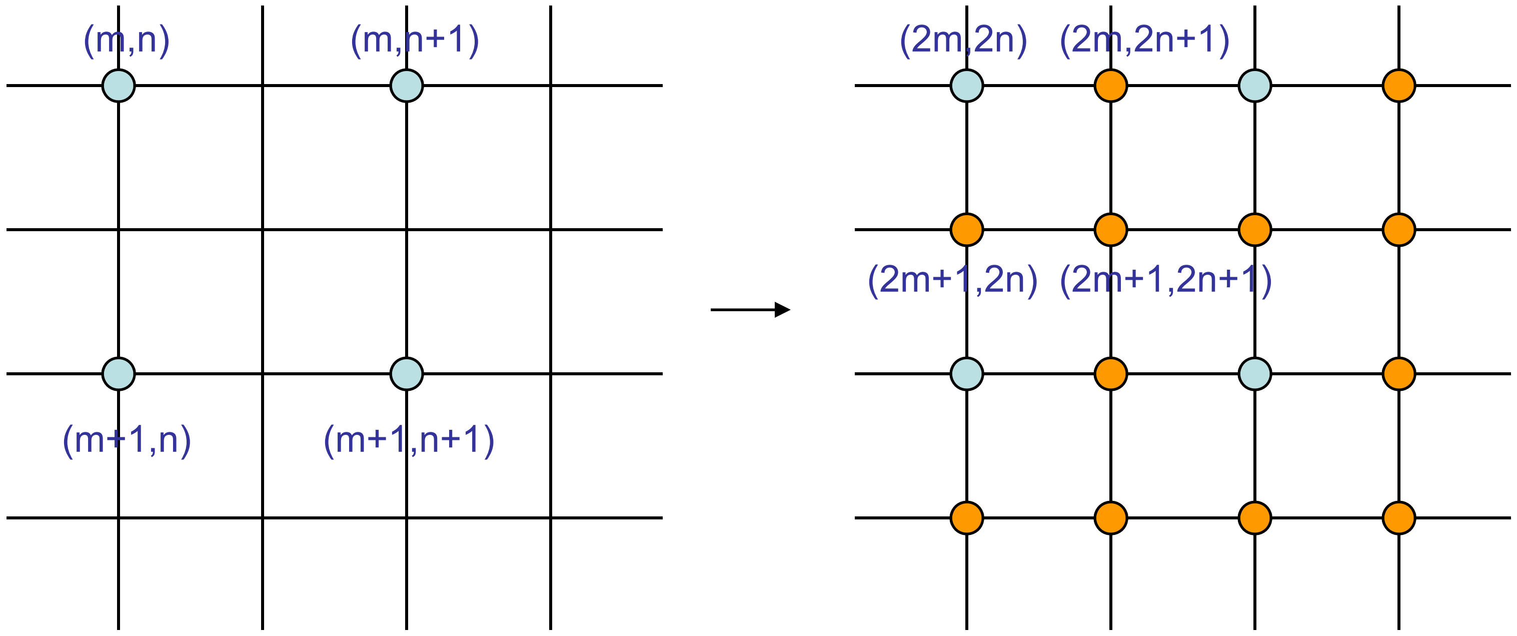
O[m',n'] (the resized image) takes the value of the sample nearest to (m'/M,n'/M) in I[m,n] (the original image):

$$O[m',n'] = I[(\mathrm{int})\,(m+0.5), (\mathrm{int})\,(n+0.5)]\,,\; m = m'/M,\; n = n'/M.$$

Also known as pixel replication: each original pixel is replaced by MxM pixels of the sample value

Equivalent to using the sample-and-hold interpolation filter.

Special Case: M=2



Nearest Neighbor:

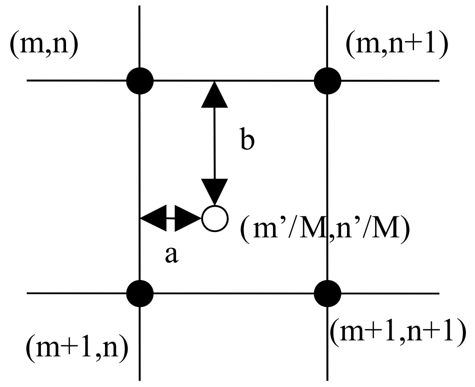
O[2m,2n]=I[m,n]

O[2m,2n+1]= I[m,n]

O[2m+1,2n]= I[m,n]

O[2m+1,2n+1]= I[m,n]

Bilinear Interpolation



- O(m',n') takes a weighted average of 4 samples nearest to (m'/M,n'/M) in I(m,n).

- Direct interpolation: each new sample takes 4 multiplications:

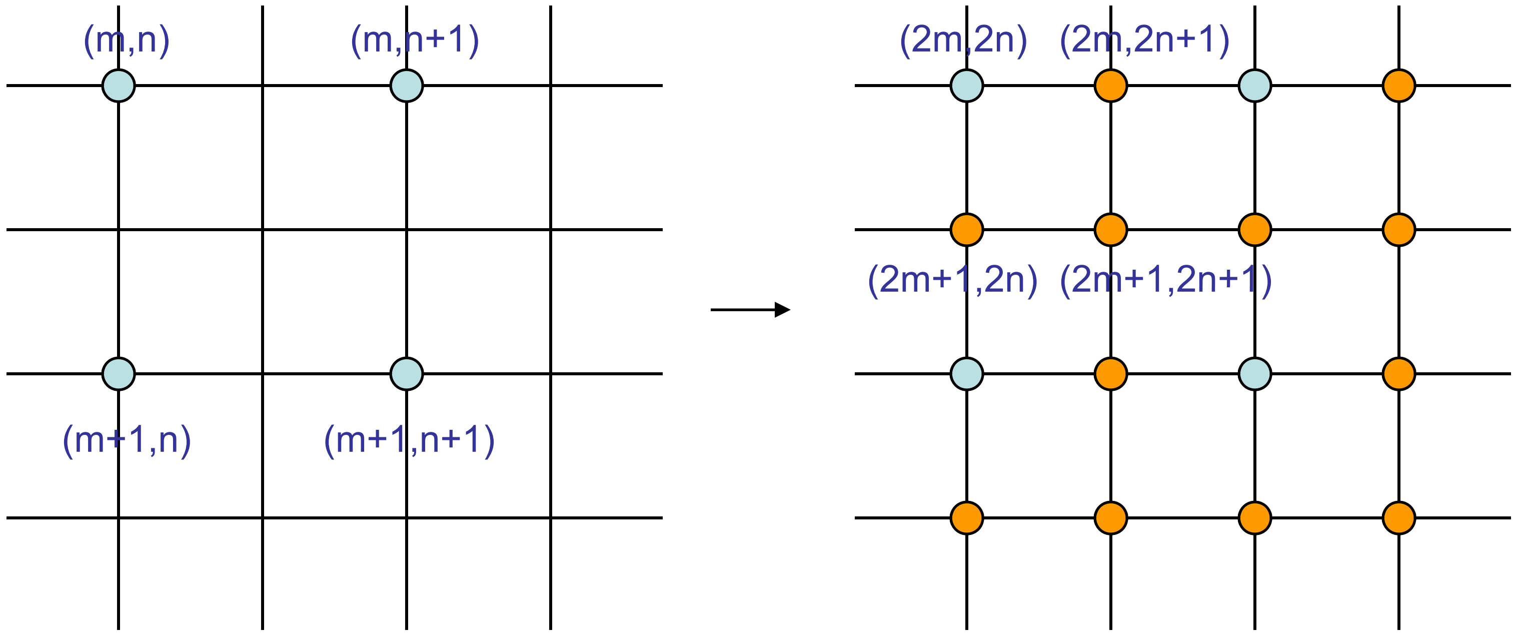
  O[m',n']=(1-a)*(1-b)*I[m,n]+a*(1-b)*I[m,n+1]+(1-a)*b*I[m+1,n]+a*b*I[m+1,n+1]

- Separable interpolation:

  i) interpolate along each row y: F[m,n']=(1-a)*I[m,n]+a*I[m,n+1]

  ii) interpolate along each column x': O[m',n']=(1-b)*F[m',n]+b*F[m'+1,n]

Special Case: M=2



Bilinear Interpolation:

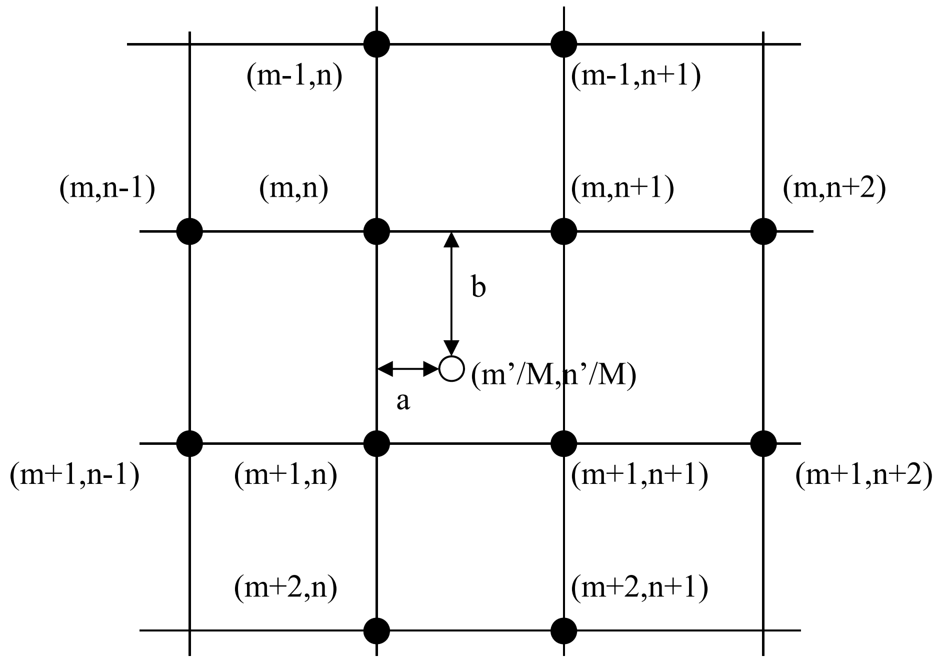
O[2m,2n]=I[m,n]

O[2m,2n+1]=(I[m,n]+I[m,n+1])/2

O[2m+1,2n]=(I[m,n]+I[m+1,n])/2

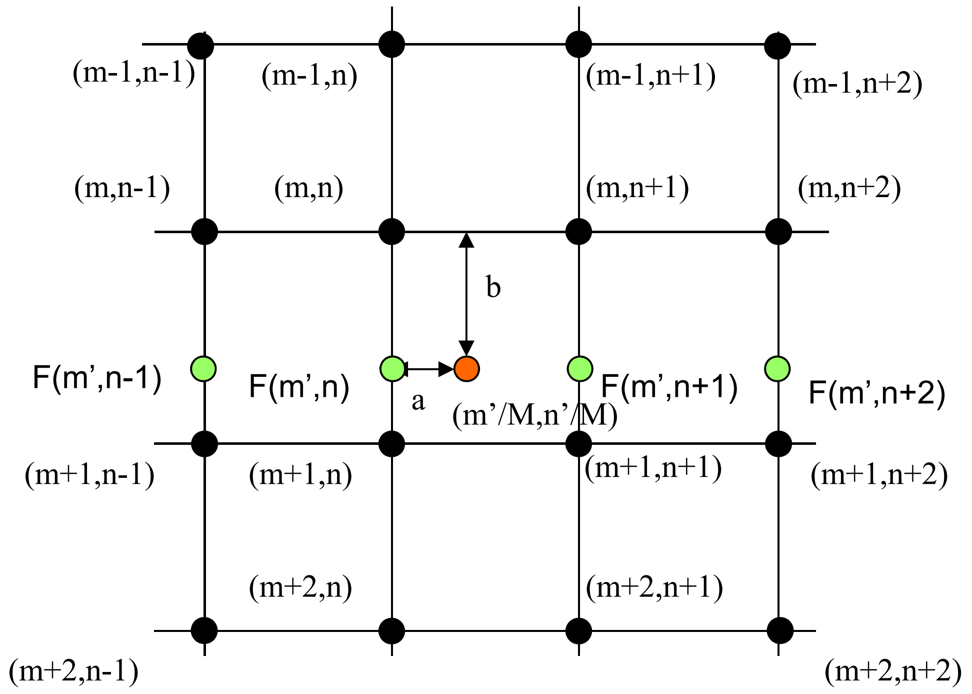
O[2m+1,2n+1]=(I[m,n]+I[m,n+1]+I[m+1,n]+I[m+1,n+1])/4

Bicubic Interpolation



- O(m',n') is interpolated from 16 samples nearest to (m'/M,n'/M) in I(m,n).
- Direct interpolation: each new sample takes 16 multiplications
- Separable interpolation:
  - i) interpolate along each row y: I[m,n]->F[m,n'] (from 4 samples)
  - ii) interpolate along each column x': F[m,n']-> O[m',n'] (from 4 samples)

Interpolation Formula



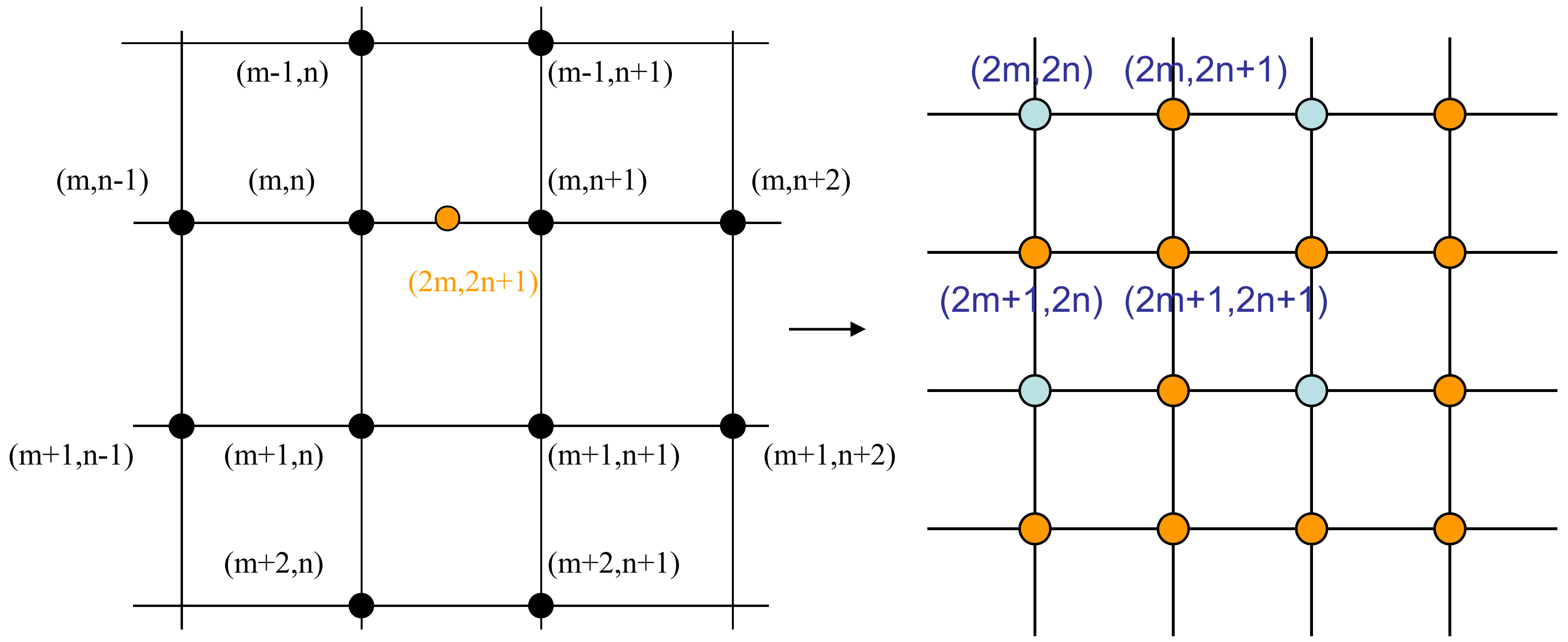
$$F[m',n] = -b(1-b)^2 I[m-1,n] + (1-2b^2+b^3)I[m,n] + b(1+b-b^2)I[m+1,n] - b^2(1-b)I[m+2,n],$$

$$where \quad m = (\text{int})\frac{m'}{M}, b = \frac{m'}{M} - m$$

$$O[m',n'] = -a(1-a)^2 F[m',n-1] + (1-2a^2+a^3)F[m',n] + a(1+a-a^2)F[m',n+1] - a^2(1-a)F[m',n+2],$$

$$where \quad n = (\text{int})\frac{n'}{M}, a = \frac{n'}{M} - n$$

Special Case: M=2

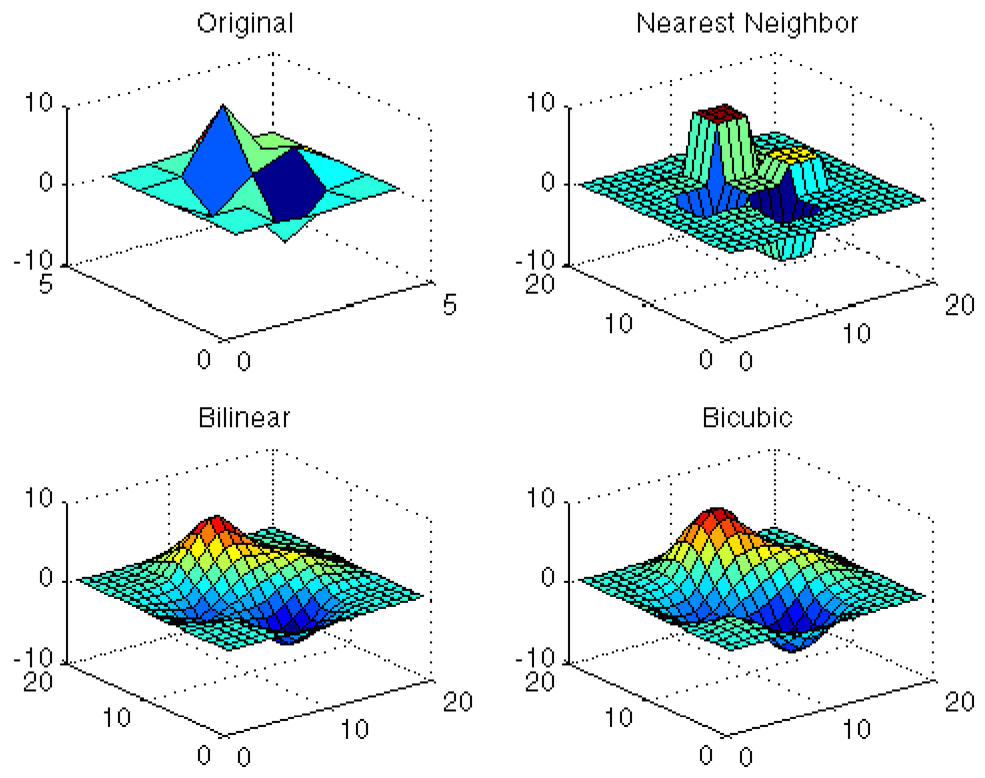


Bicubic interpolation in Horizontal direction

F[2m,2n]=I[m,n]
F[2m,2n+1]= -(1/8)I[m,n-1]+(5/8)I[m,n]+(5/8)I[m,n+1]-(1/8)I(m,n+2)

Same operation then repeats in vertical direction

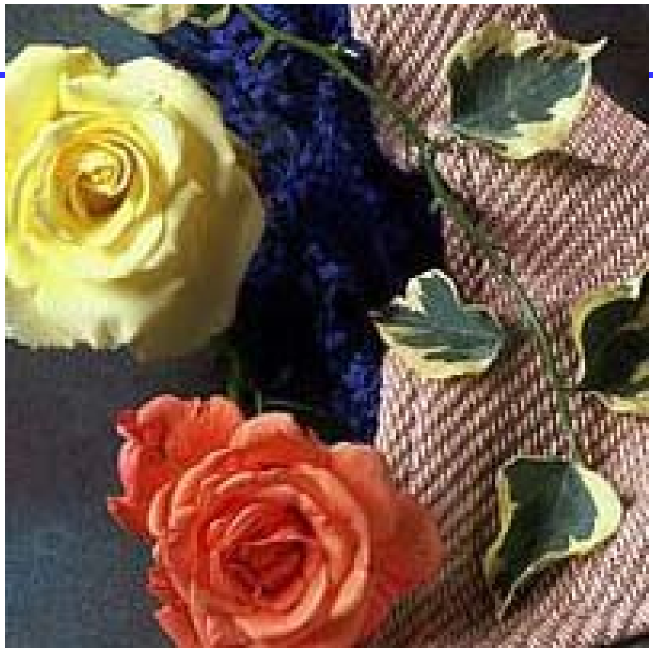
Comparison of Interpolation Methods



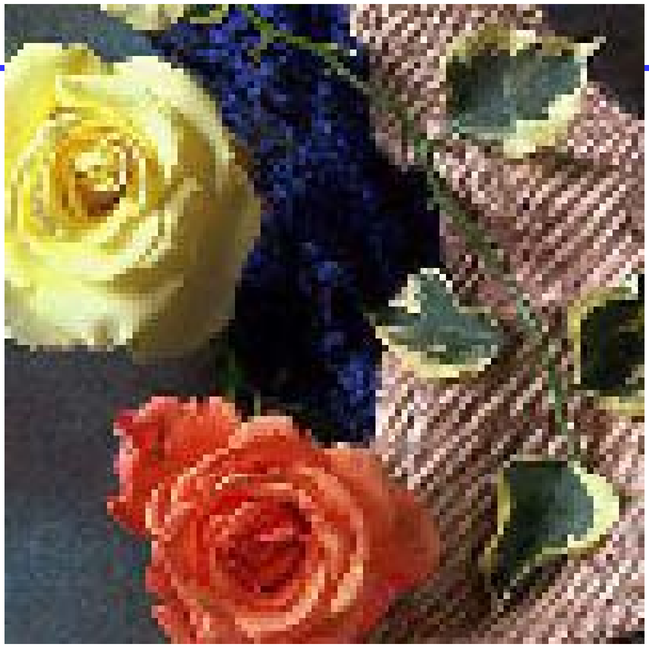
Resize_peak.m

Up-Sampled from w/o Prefiltering

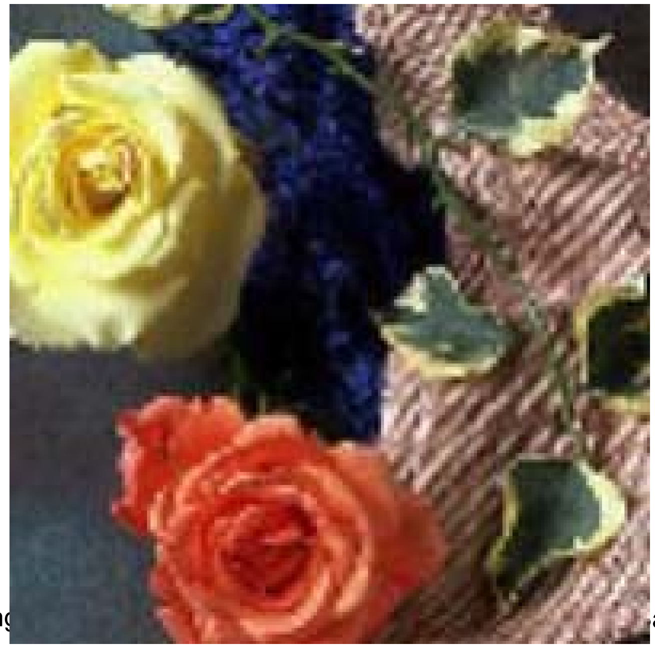
Original



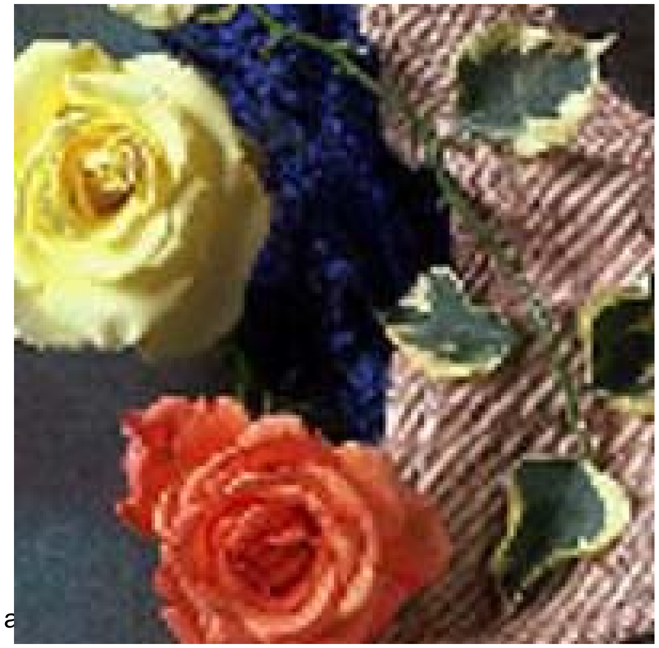
Nearest neighbor



Bilinear

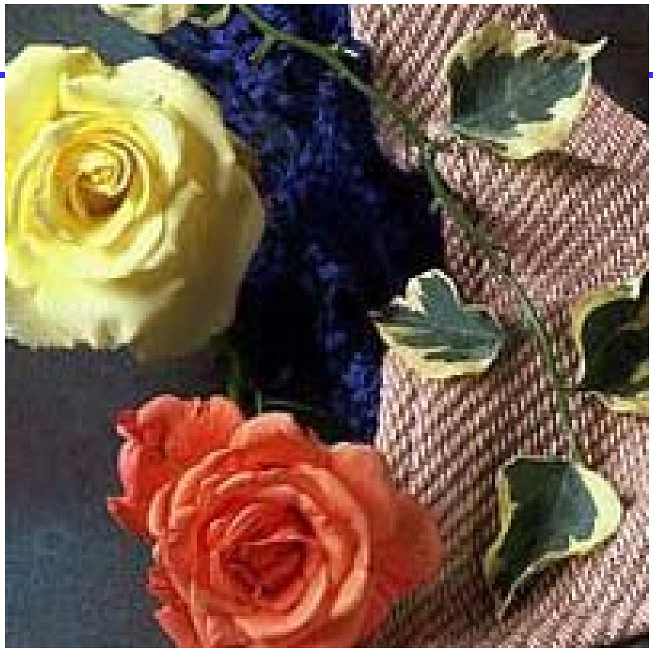


Bicubic

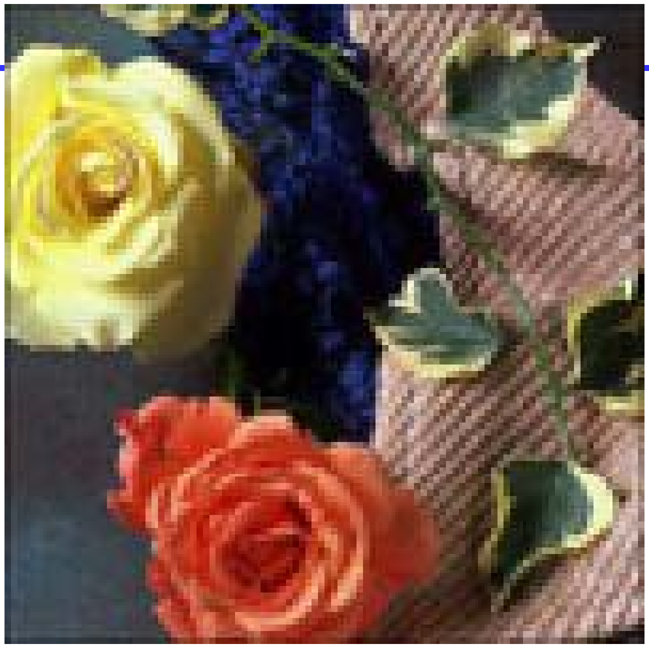


Up-Sampled from with Prefiltering

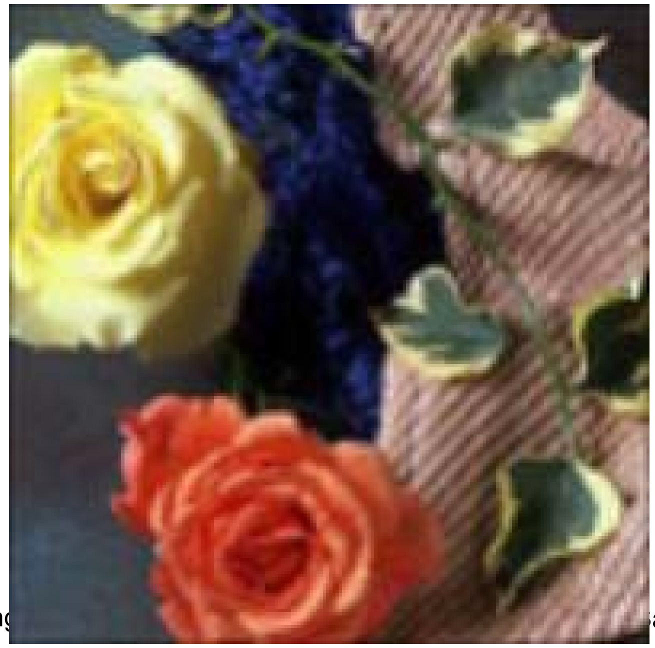
Original



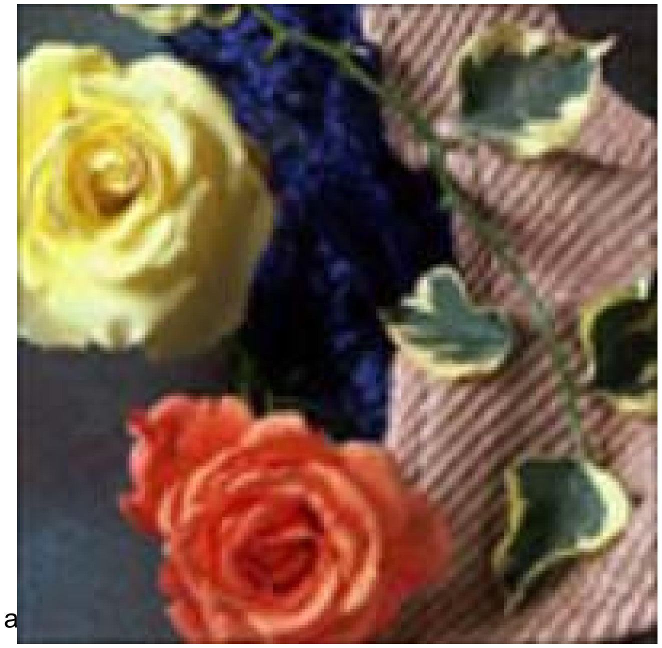
Nearest neighbor



Bilinear



Bicubic

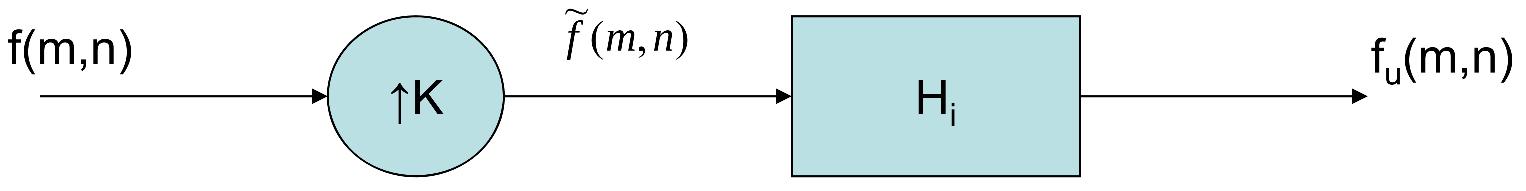


Matlab for Image Resizing

```
[img]=imread('fruit.jpg','jpg');
%downsampling without prefiltering
img1=imresize(img,0.5,'nearest');
%upsampling with different filters:
img2rep=imresize(img1,2,'nearest');
img2lin=imresize(img1,2,'bilinear');
img2cubic=imresize(img1,2,'bicubic');

%down sampling with filtering
img1=imresize(img,0.5,'bilinear',11);
%upsampling with different filters
img2rep=imresize(img1,2,'nearest');
img2lin=imresize(img1,2,'bilinear');
img2cubic=imresize(img1,2,'bicubic');
```

Filtering View: Up Sampling by a Factor of K



Zero-padding Post-filtering

$$\tilde{f}(m,n) = \begin{cases} f(m/K, n/K) & \text{if } m,n \text{ are multiple of } K \\ 0 & \text{otherwise} \end{cases}$$

$$f_u(m,n) = \sum_{k,l} h(k,l)\,\tilde{f}(m-k, n-l)$$

Ideally H should be a low pass filter with cutoff at 1/2K in digital frequency, or fs/2K in continuous frequency

Homework (1)

1. Consider a 1D signal $f(t) = \sin(4\pi t)$. Illustrate the original and the sampled signal $f(n)$ obtained with a sampling interval $\Delta t = 1/3$. Draw on the same figure the interpolated signal from the sampled one using the sample-and-hold and the linear interpolation filter, respectively. Explain the observed phenomenon based on both the Nyquist sampling theorem as well as physical interpretation. What is the largest sampling interval that can be used to avoid aliasing?
2. Consider a function $f(x, y) = \cos 2\pi(4x + 2y)$ sampled with a sampling period of $\Delta x = \Delta y = \Delta = 1/6$ or sampling frequency $f_s = 1/\Delta = 6$.
   a) Assume that it is reconstructed with an ideal low-pass filter with cut-off frequency $f_{cx} = f_{cy} = 1/2 f_s$. Illustrate the spectra of the original, sampled, and reconstructed signals. Give the spatial domain function representation of the reconstructed signal. Is the result as expected?
   b) If the reconstruction filter has the following impulse response:

$$h(x,y) = \begin{cases} 1 & -\Delta/2 < x, y < \Delta/2 \\ 0 & otherwise \end{cases}$$

   Illustrate the spectra of the reconstructed signal in the range $-f_s \le u,v \le f_s$. Give a spatial domain function representation of the reconstructed signal if the reconstruction filter is band-limited to ($-f_s \le u,v \le f_s$). (i.e., this filter remains the same for the frequency range $-f_s \le u,v \le f_s$, and is set to 0 outsize this range.)

Homework(2)

3. (Computer Assignment) Write your own program or programs which can: a) Down sample an image by a factor of 2, with and without using the averaging filter; b) Up-sample the previously down-sampled images by a factor of 2, using the pixel replication and bilinear interpolation methods, respectively. You should have a total of 4 interpolated images, with different combination of down-sampling and interpolation methods. Your program could either directly display on screen the processed images during program execution, or save the processed images as computer files for display after program execution. Run your program with the image *Barbara*. Comment on the quality of the down/up sampled images obtained with different methods.

   Note: you should not use the ”imresize” function in Matlab to do this assignment. But you are encouraged to compare results of your program with ”resize”.

Reading

- R. Gonzalez, “Digital Image Processing,” Section 2.4
- A.K. Jain, “Fundamentals of Digital Image Processing,” Section 4.1-4.4