18-799: Applied Computer Vision

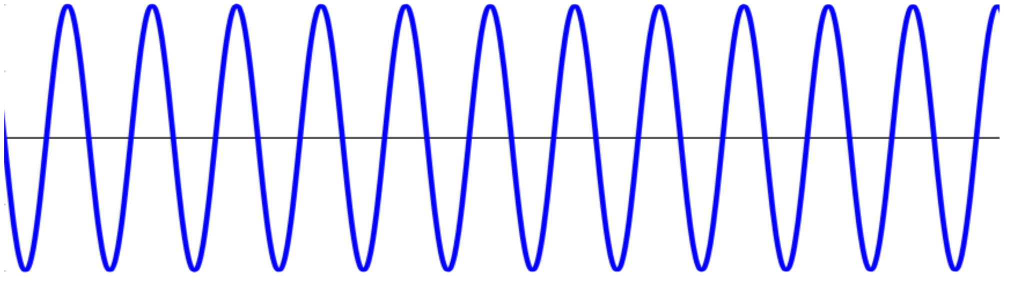
Spring 2026

Image Processing:
Image Resizing

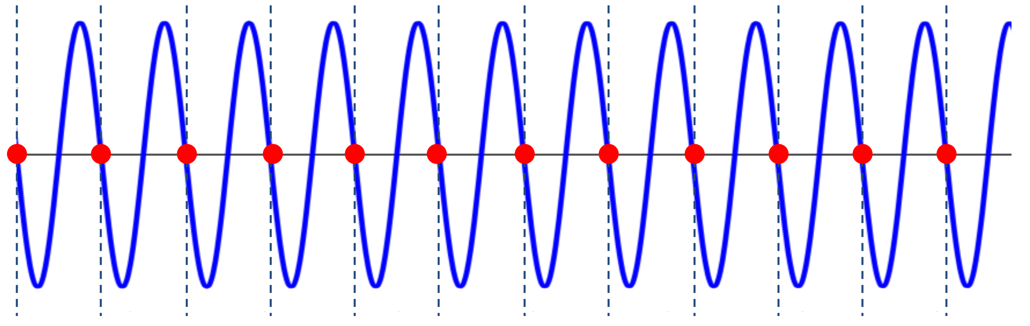
Objectives

In today's class we will learn

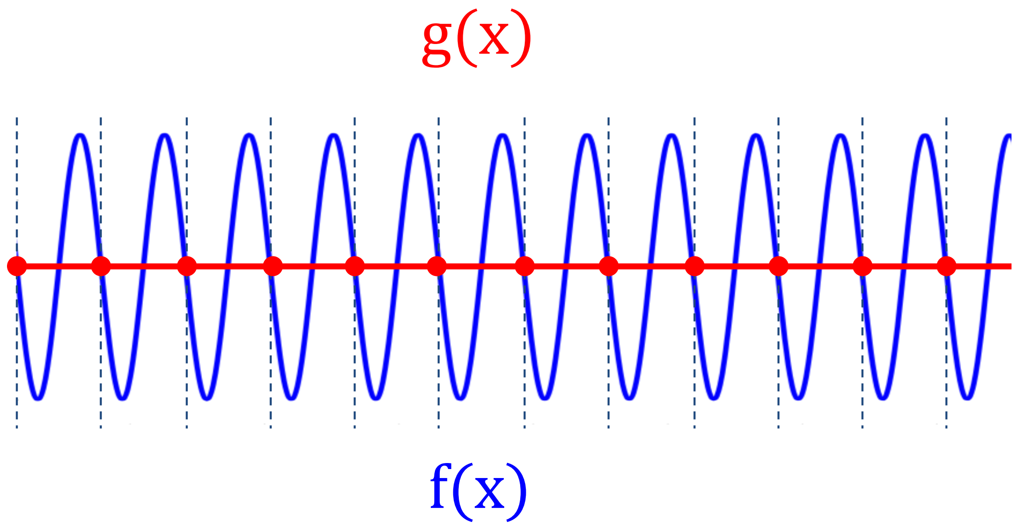
- What is Image Resizing
- What is image pyramid
- Application of Image pyramid

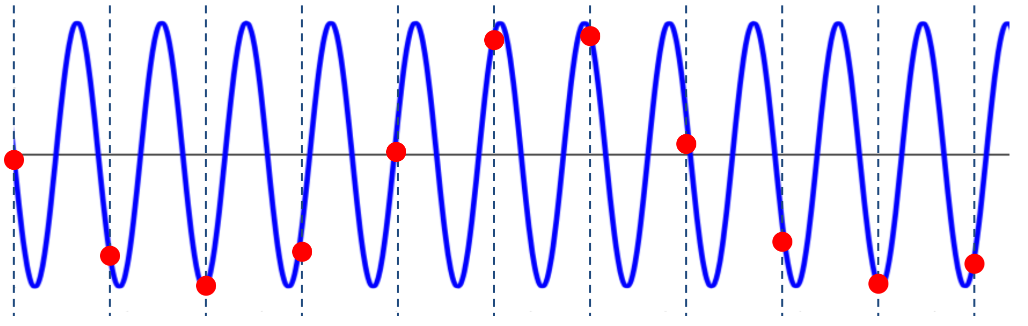


f(x)



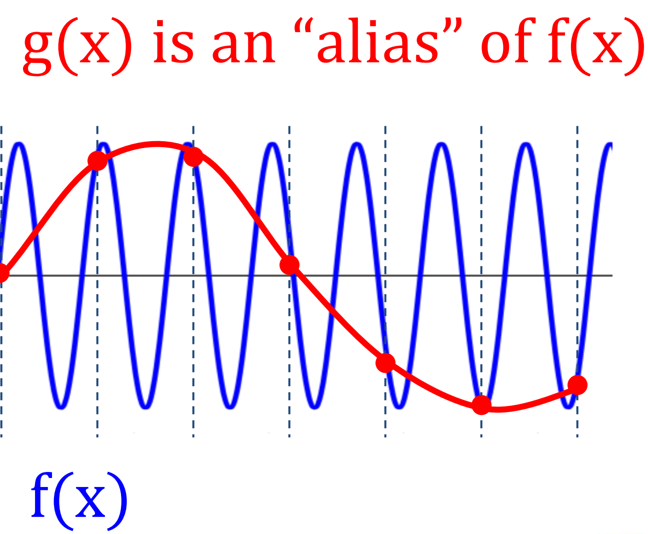
f(x)


g(x)
f(x)

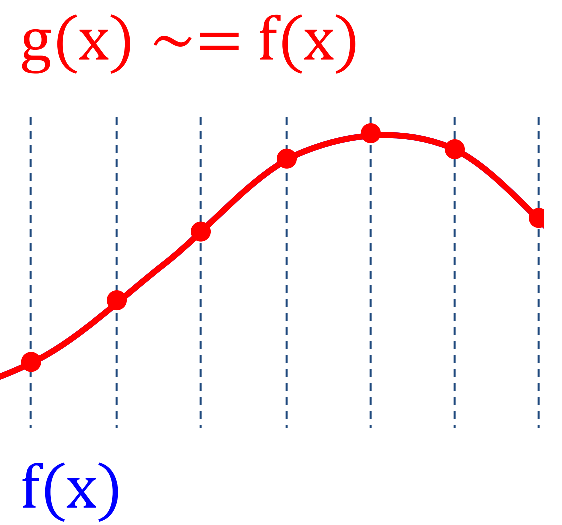


f(x)

Aliasing

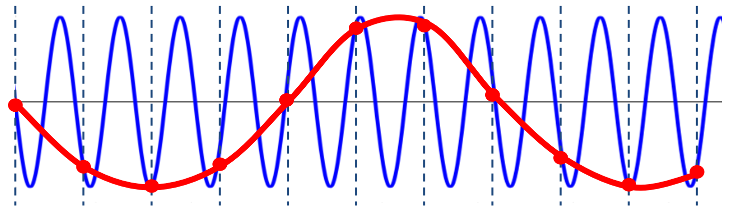


Aliasing

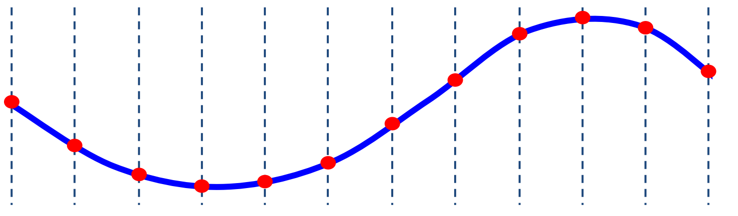


No aliasing in this case

Sampling and the **Nyquist** rate



< 1 sample per cycle



> 10 samples per cycle

To avoid aliasing

- **≥ two samples per cycle**

This minimum sampling rate is called the **Nyquist rate**

But this only works for sine waves, right?

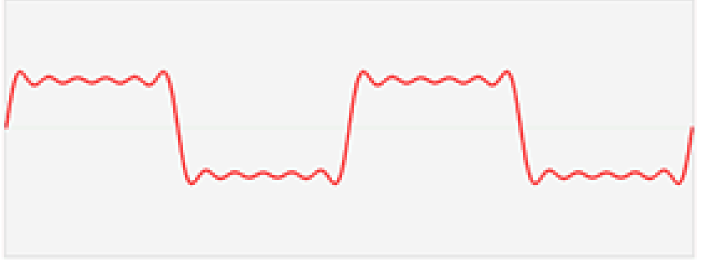
Fourier transform

- Decomposes any signal or image into weighted sum of sines and cosines

To avoid aliasing

- sampling rate ≥ 2 * max frequency present in the image (Nyquist rate)

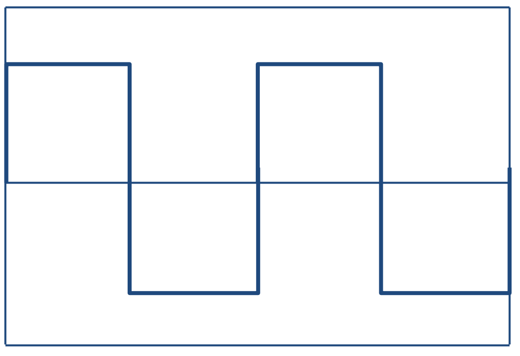
Fourier Transform



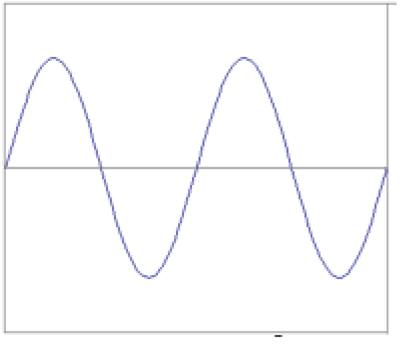
Fourier transform

- Decomposes any signal or image into weighted sum of sines and cosines

Fourier Series



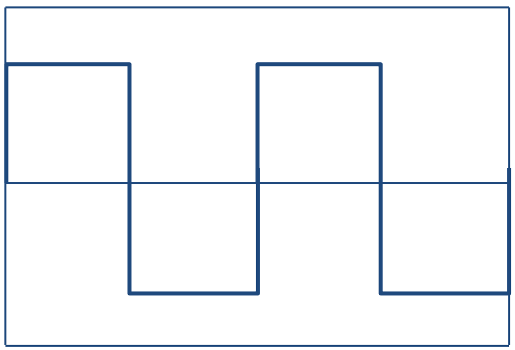
$\approx$



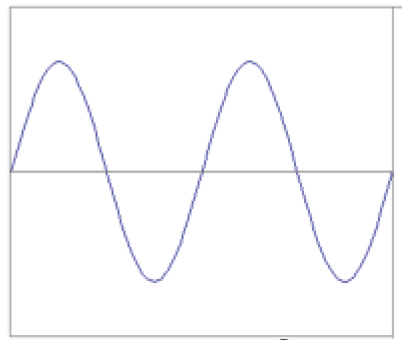
We want to get this function

Slides: Alyosha Efros

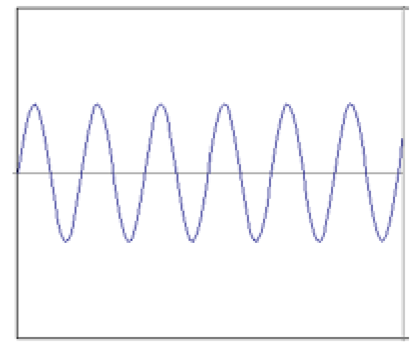
Fourier Series



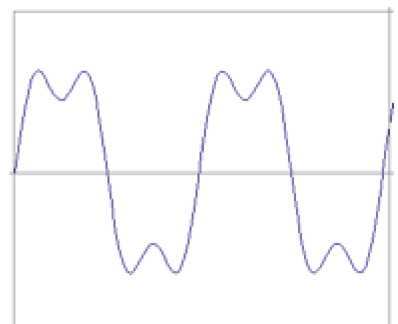
≈



+



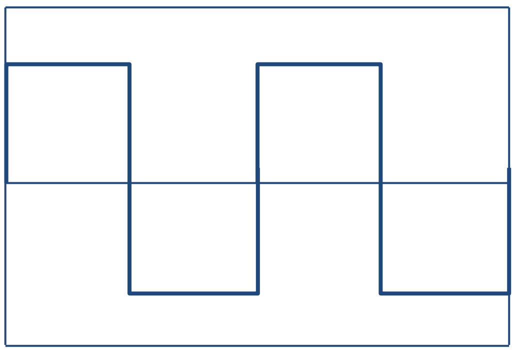
=



We want to get this function

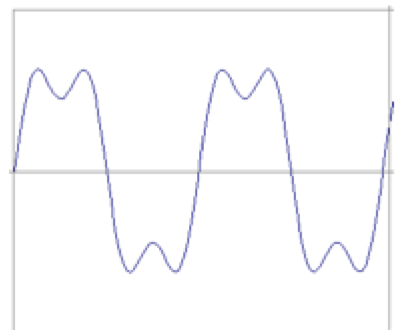
Following slides from Alyosha Efros

Fourier Series

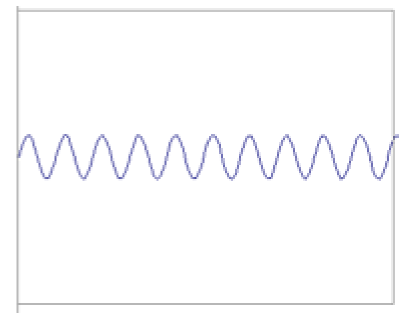


We want to get this function

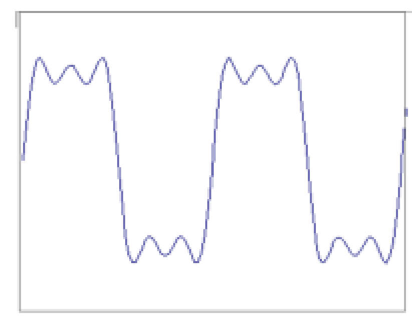
$\approx$



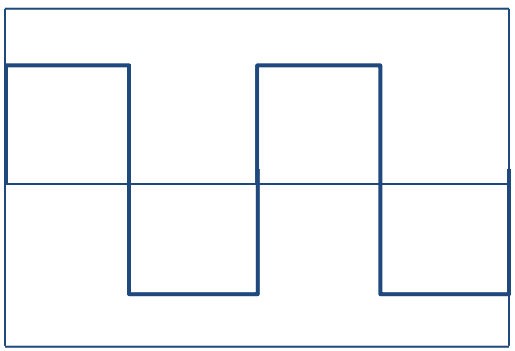
$+$



$=$

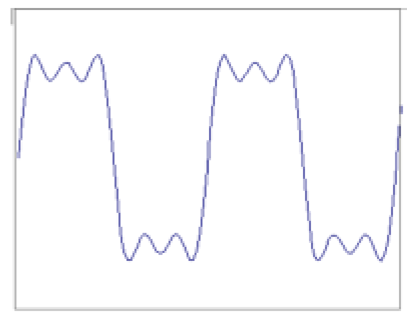


Fourier Series

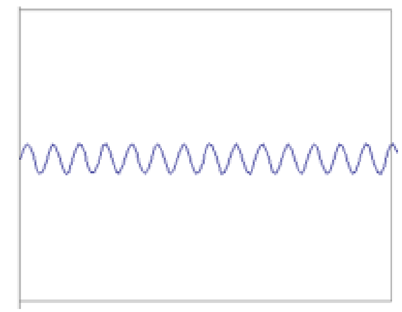


We want to get this function

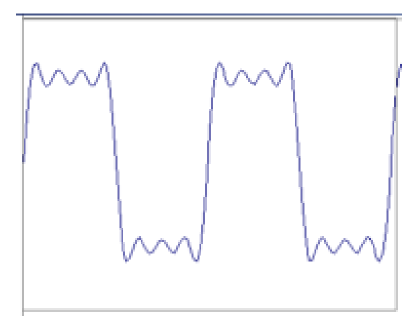
≈



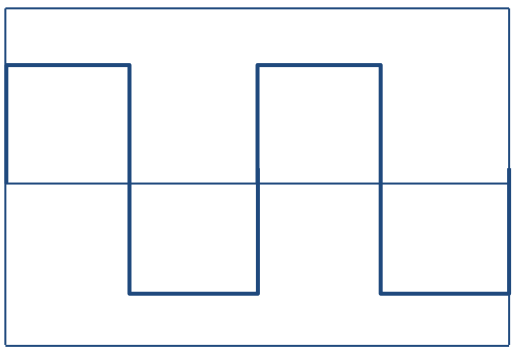
+



=

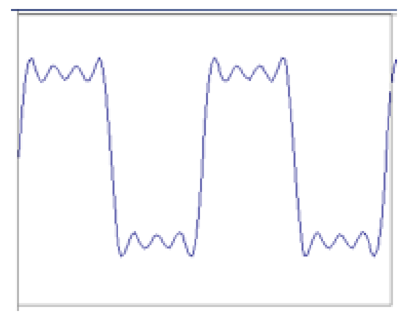


Fourier Series

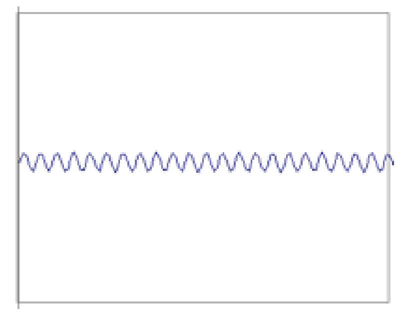


We want to get this function

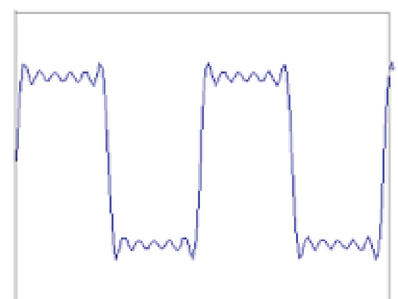
≈



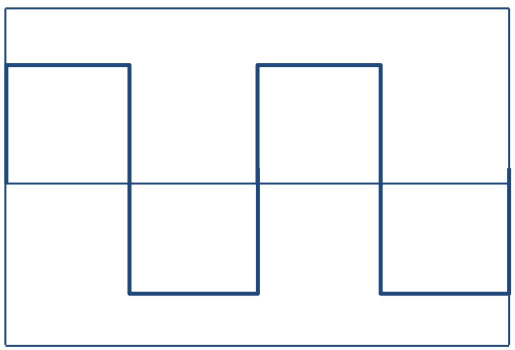
+



=

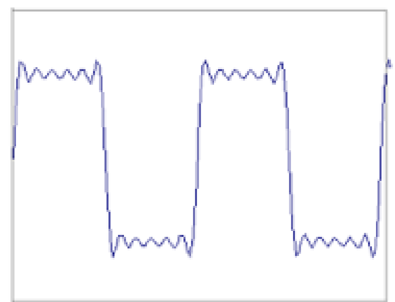


Fourier Series

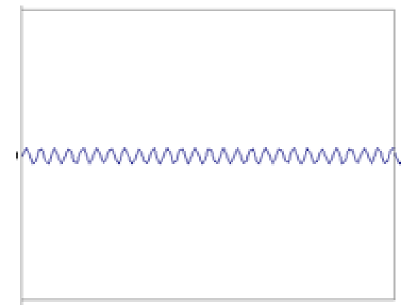


We want to get this function

≈

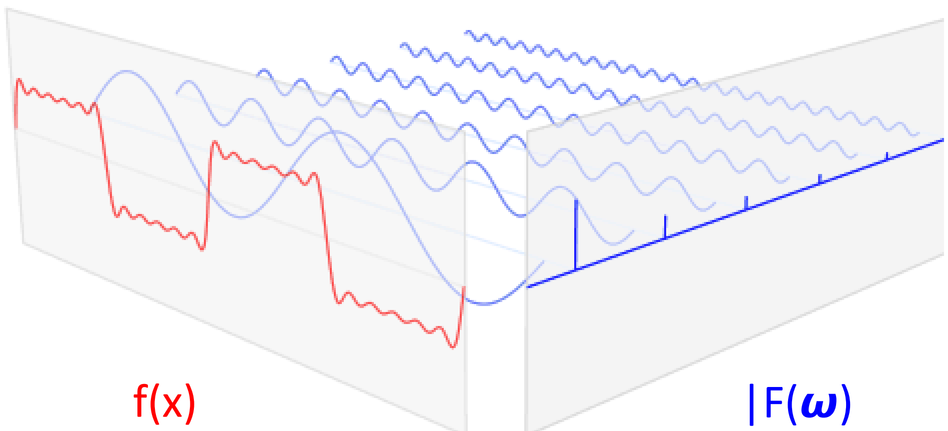


+



=

Fourier transform



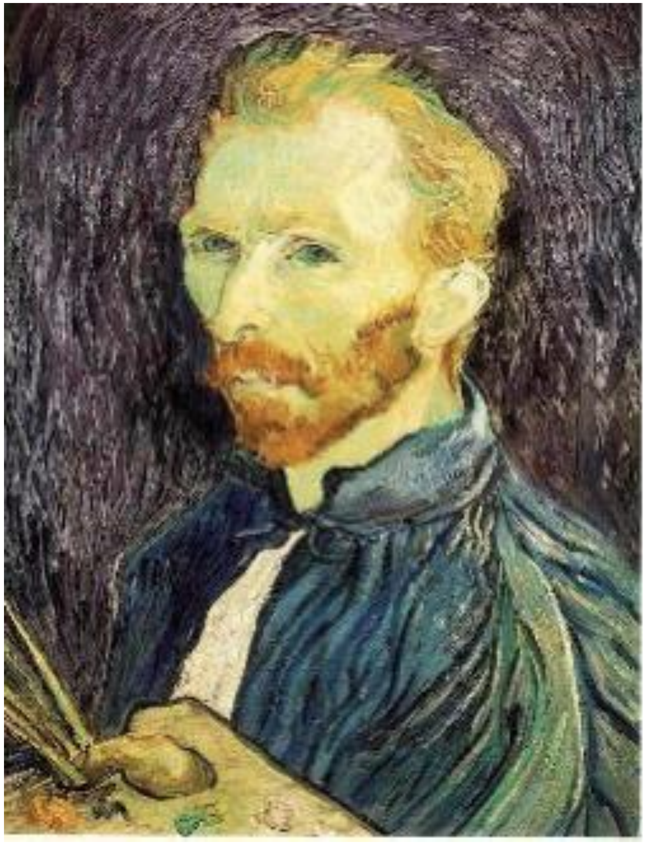
$f(x)$

signal

$|F(\boldsymbol{\omega})|$

Fourier transform of signal

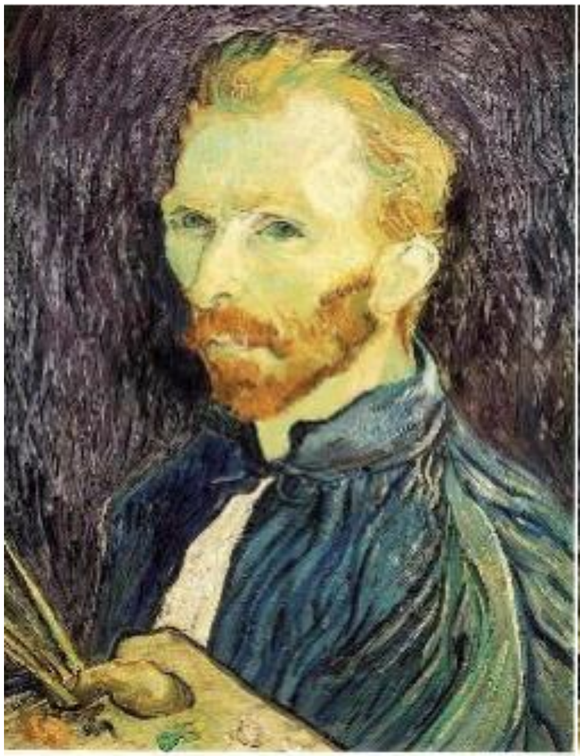
Image Resizing



1/4

1/8

Image Resizing



1/2

1/4 (2x zoom)

1/8 (4x zoom)

Image Resizing

- Downsampling

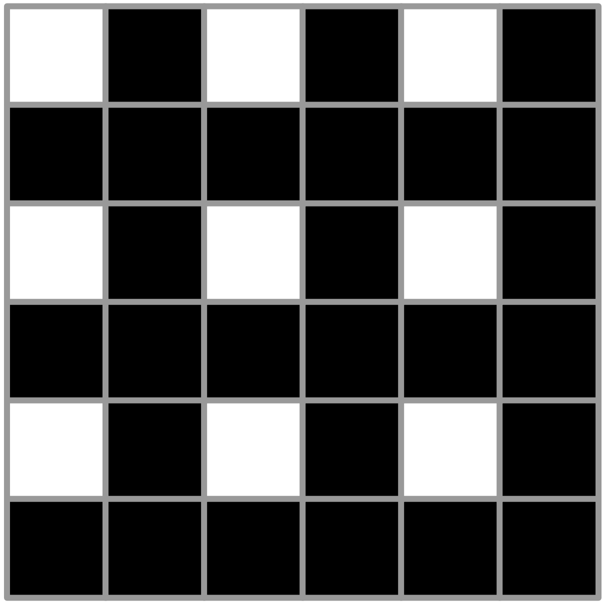
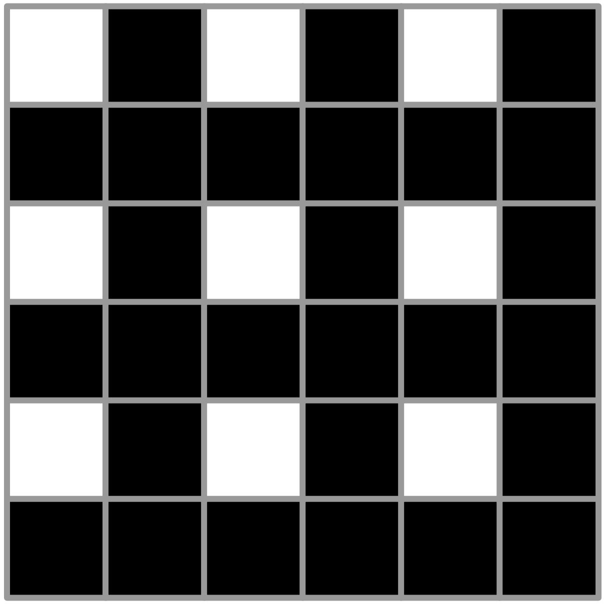


Image Resizing

- Upsampling



Carnegie Mellon University Africa

Image Resizing

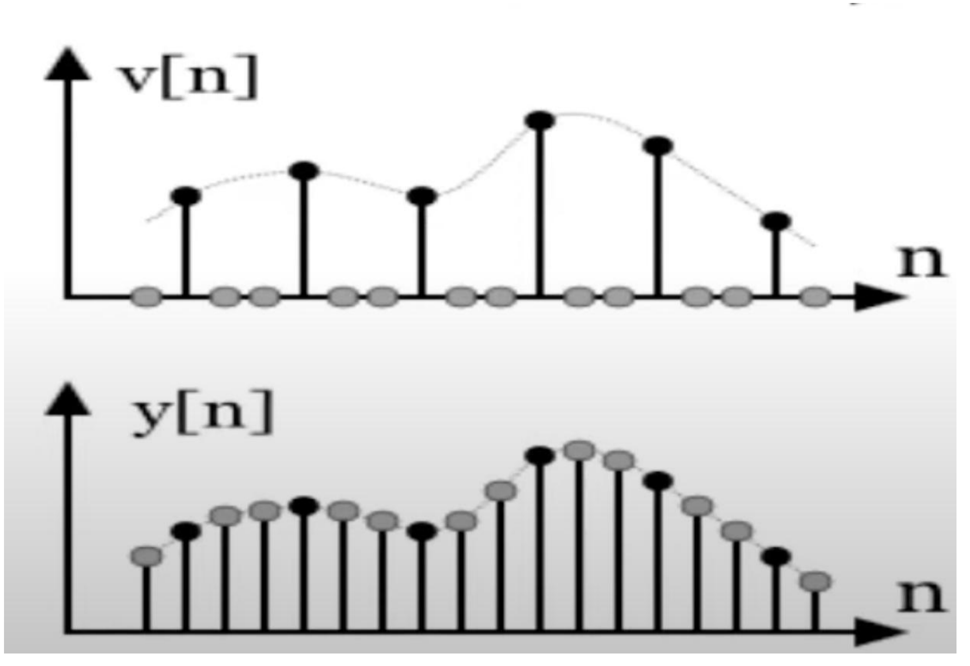


Image resizing

- Upsampling (Nearest Neigbor interpolation)

5	7	8
4	7	8
3	2	1

5		7		8
4		7		8
3		2		1

- Upsampling (Linear Interpolation)

5	7	8
4	7	8
3	2	1

5		7		8
4		7		8
3		2		1

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$$

- Upsampling (Linear Interpolation)

5	7	8
4	7	8
3	2	1

5	?			8
4				8
3				1

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$$

- Upsampling (Linear Interpolation)

5	7	8
4	7	8
3	2	1

5				8
	?			
4				8
3				1

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$$

Multi-resolution image pyramids

- Commonly used in coarse-to-fine matching, optical flow, stereo, blending, …
- … deep neural networks

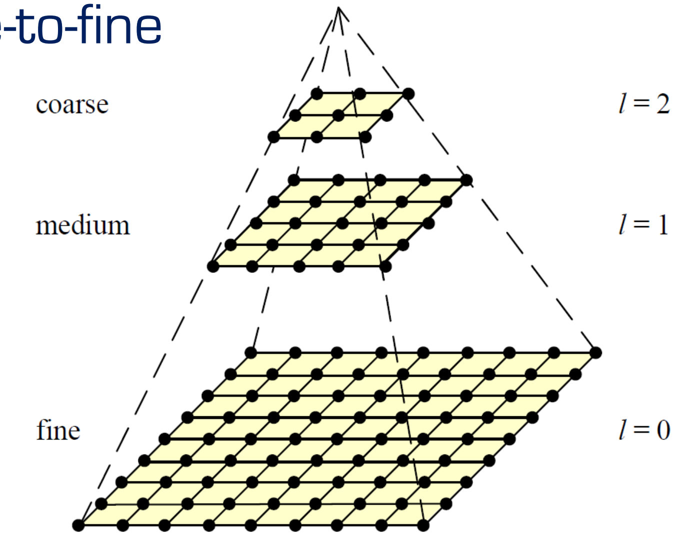
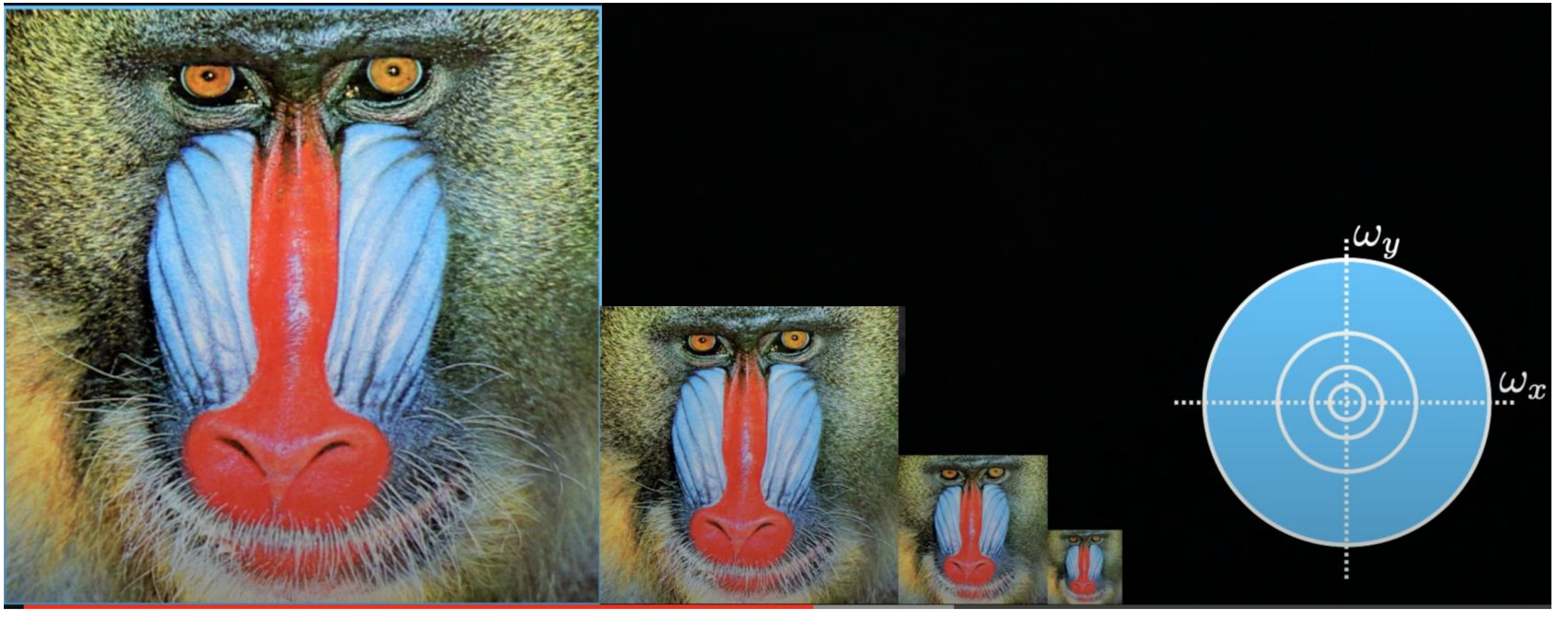


Figure 3.32 A traditional image pyramid: each level has half the resolution (width and height), and hence a quarter of the pixels, of its parent level.

“Gaussian” pyramid



“Gaussian” pyramid

1. **Level 0 (original)**

   Start with the original image G_0.

2. Gaussian smoothing

   Convolve the image with a Gaussian kernel g_σ :

$$\tilde{G}_k = g_\sigma * G_k$$

   This removes high spatial frequencies (anti-aliasing).

3. **Downsampling**

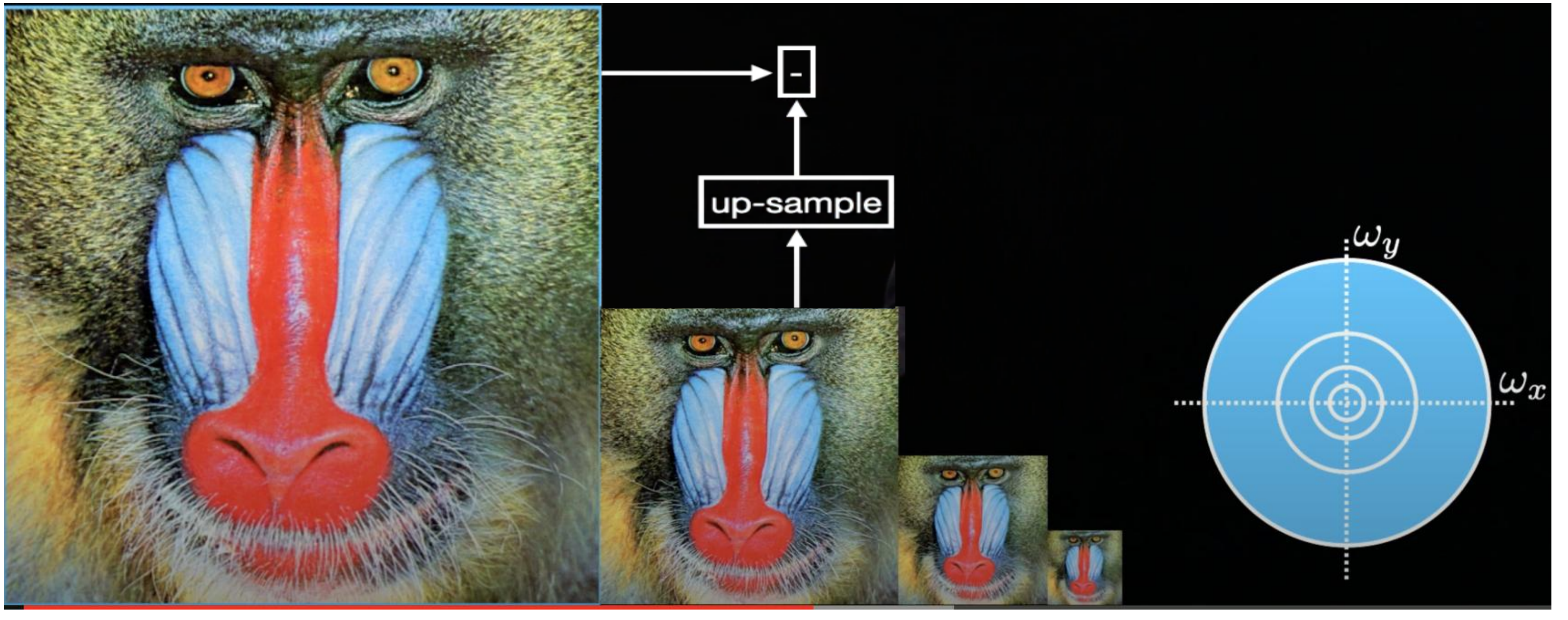
   Subsample by a factor of 2 in each dimension (keep every other pixel):

$$G_{k+1}(x, y) = \tilde{G}_k(2x, 2y)$$

4. **Repeat**

   Apply steps $2 - 3$ to build levels $G_1, G_2, \ldots$.
   Each level is smaller and smoother than the previous one.

“Laplacian” pyramid



“Laplacian” pyramid

“Laplacian” pyramid

1. Start with a Gaussian pyramid

$$G_0, G_1, G_2, \ldots$$

2. Upsample the next Gaussian level

Expand G_{k+1} back to the size of G_k :

$$\hat{G}_k = \mathrm{Expand}(G_{k+1})$$

3. Take the difference

$$L_k = G_k - \hat{G}_k$$

4. Repeat for all levels

The final level is usually:

$$L_N = G_N$$

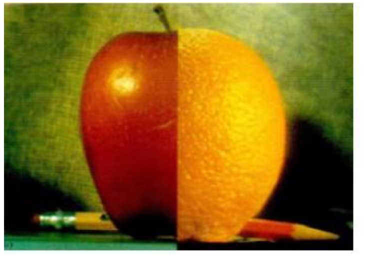
(the coarsest, low-frequency image)

Each L_k is one level of the Laplacian pyramid.

Image blending

Hard compositing:

$$\begin{aligned} I(x,y) \quad &= M(x,y)S(x,y) + (1 - M(x,y))T(x,y) \\ &= \begin{cases} S(x,y) & M(x,y) = 1 \\ T(x,y) & M(x,y) = 0 \end{cases} \end{aligned}$$



Weighted Transition Region:

Laplacian pyramid blending

- Compute Laplacian pyramid for source and target

$$L^S, L^T$$

- Compute gaussian pyramid for mask M

$$G$$

- Laplacian pyramid for composite:

$$= G_i L_i^S + (1 - G_i) L_i^T$$

$$i = 0, \dots N$$

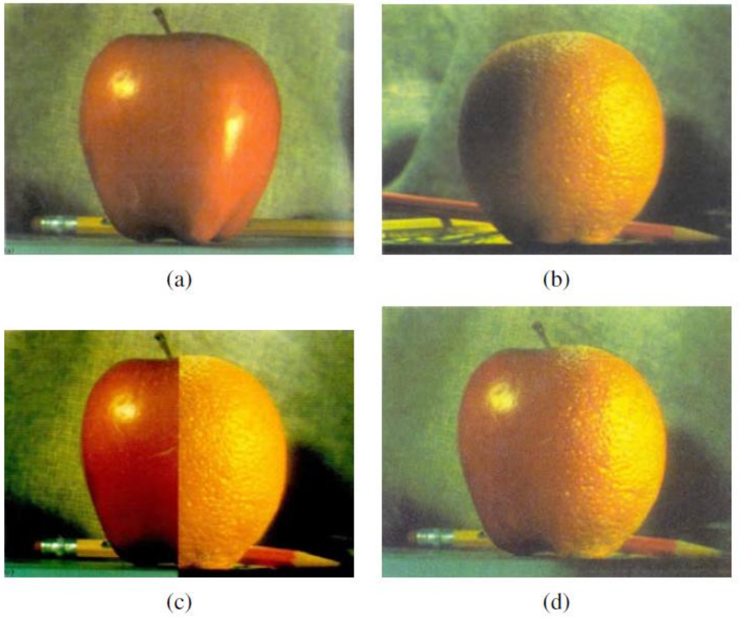
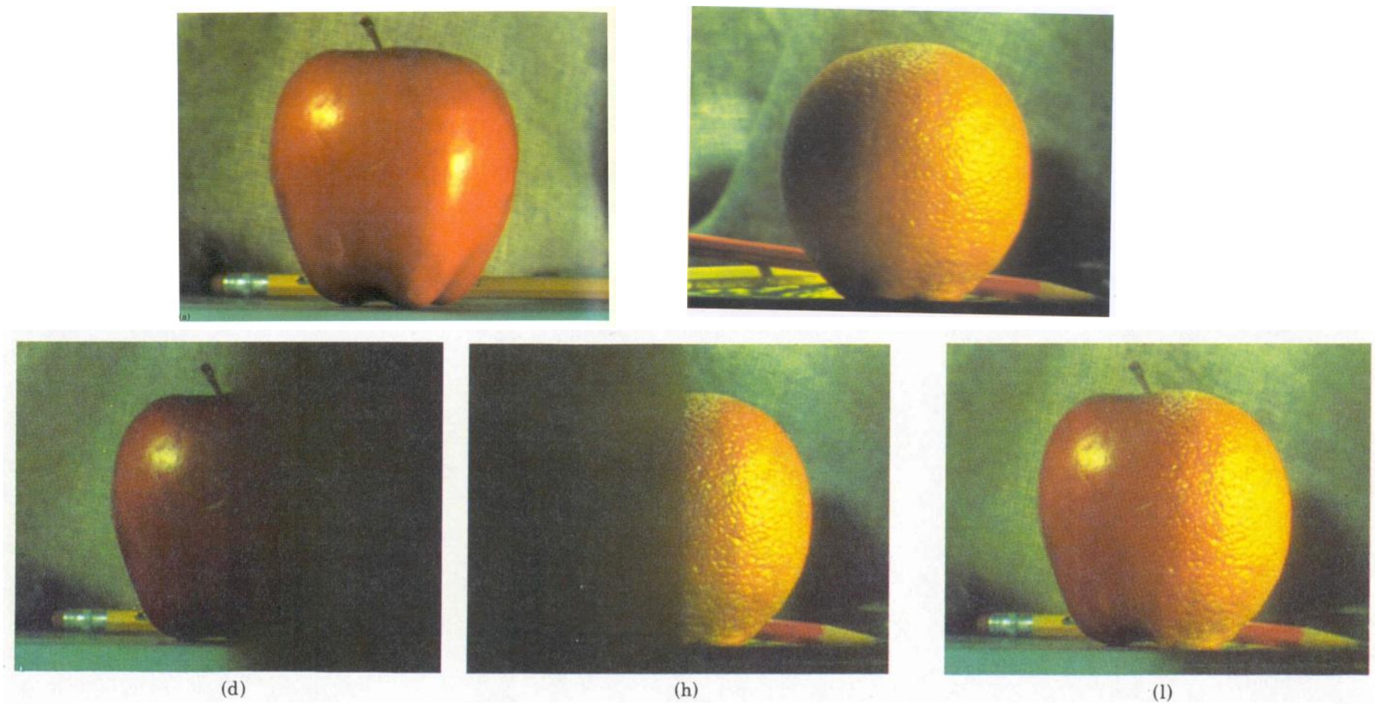


Figure 3.41 Laplacian pyramid blending (Burt and Adelson 1983b) © 1983 ACM: (a) original image of apple, (b) original image of orange, (c) regular splice, (d) pyramid blend.

Richard Szeliski

Carnegie Mellon University Africa

Pyramid Blending

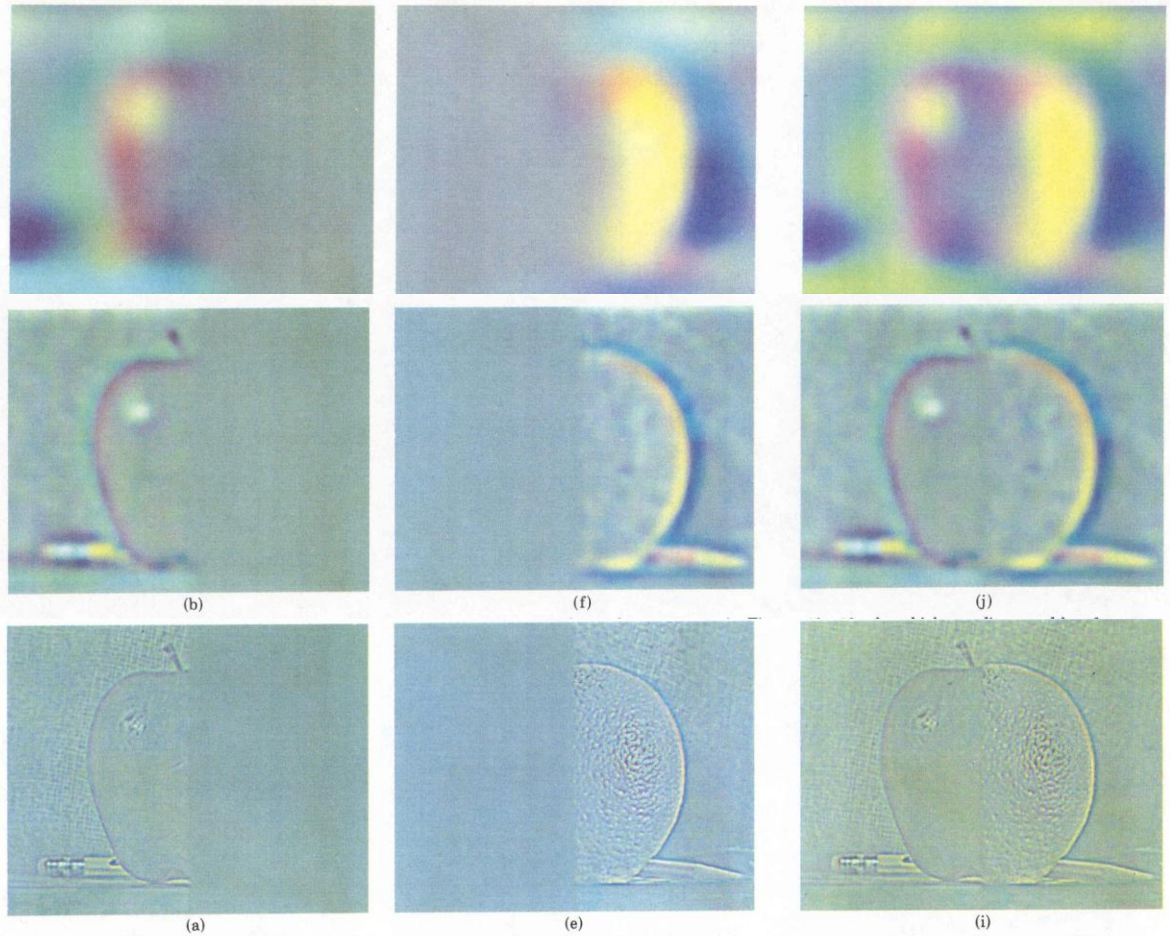


Burt, P. J. and Adelson, E. H., A multiresolution spline with applications to image mosaics, ACM Transactions on Graphics, 42(4), October 1983, 217-236.

Laplacian level 4

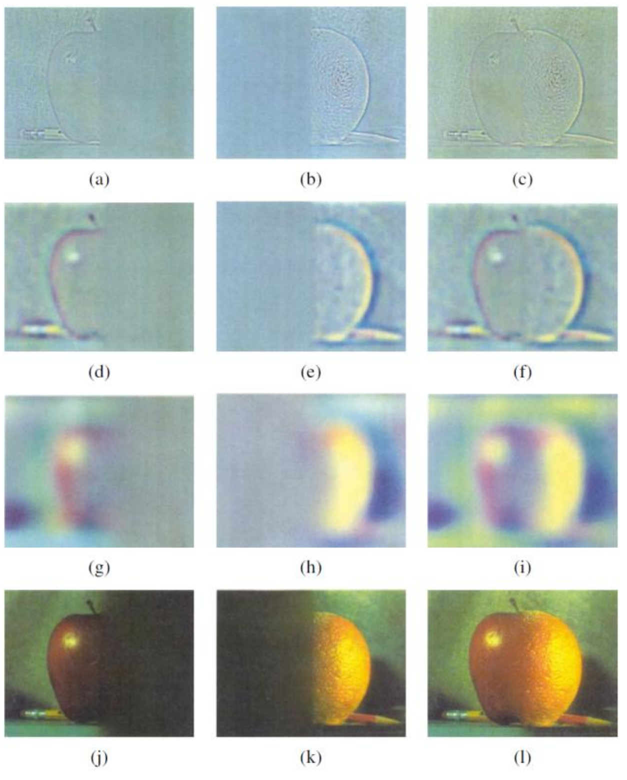
Laplacian level 2

Laplacian level 0



left pyramid | right pyramid | blended pyramid

Laplacian pyramid blending



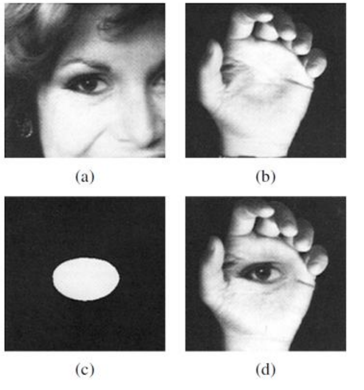


Figure 3.43 Laplacian pyramid blend of two images of arbitrary shape (Burt and Adelson 1983b) © 1983 ACM: (a) first input image; (b) second input image; (c) region mask; (d) blended image.

Richard Szeliski