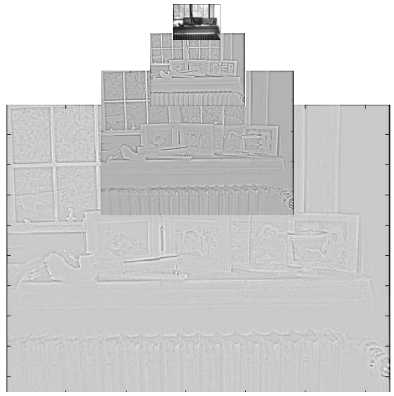
CSE397/497-011 Real-time Image Processing
for Autonomous Robot Systems



Lecture 13: Pyramids in Image Processing

LEHIGH UNIVERSITY

Department of
Computer Science & Engineering

P.C. Rossin
College of Engineering
and Applied Science

Class Objectives/Announcements

- Objectives
  - Examine pyramid representation of images
  - Review procedures for generating Gaussian and Laplacian pyramids
  - Investigate applications for pyramid processing
- Announcements
  - Homework 4 is posted. Due date is 1 week from today
  - Lab sessions to resume Thursday

References

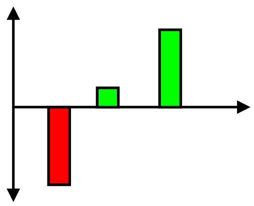
- Gonzalez and Woods, "Wavelets and Multiresolution Processing," Chapter 7
- Adelson *et al*, "Pyramid Methods in Image Processing," 1984
- CVOnline, "Image Pyramids and Scale Reduction," http://homepages.inf.ed.ac.uk/cgi/rbf/CVONLINE/entries.pl?TAG3

One Motivation: 2D Pattern Recognition

- Let's assume we are given a pattern template of interest (a target, face, etc.) and must locate it within the video image
- This template can be reduced to a series of numbers corresponding to its pixel values
- We can compare these pixel values to similar size regions in the video image
- The video image position of *maximum* correlation to the template corresponds to the location of our pattern in the video image

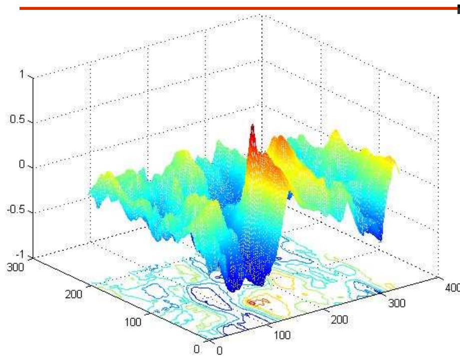
2-D Pattern Recognition Example

Template



Search Region

2D Pattern Recognition Example



Actual Template

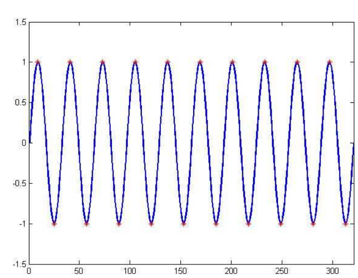
Max Correlation Template

Complexity

- For an m x n image…
- For a p x q template…
- The complexity of the 2D pattern recognition task is O(mnpq) ☹
- This gets even worse for a family of templates (*e.g.*, to address scale and/or rotational effects)
- To reduce actual running time (not complexity) we can
  - Reduce the image size
  - Reduce the template size
- Why not just do this?

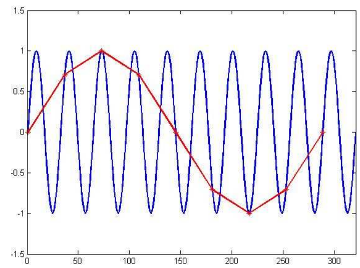
ISSUE: Sampling Limits

- *Shannon's Sampling Theorem* states that if a function *f* is sampled at a rate ≥ twice its highest frequency, *f* can be completely recovered from its samples
- This is known as the *Nyquist Frequency Limit*
- The sampling rate in images is the pixels/unit distance



Aliasing

- For an image sampled spatially, the shortest "wavelength" that be represented is 2 pixels
- When a signal is under-sampled, aliasing can result
- Aliasing is when a high frequency signal masquerades as a lower frequency signal



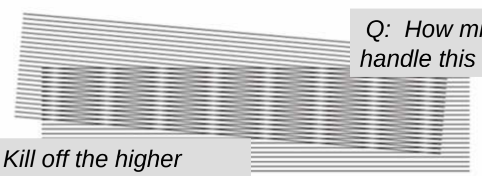
CSE397/497 Real-time Image Processing

Aliasing Examples

Original

Scanned Image

Moire Pattern



Q: How might we handle this effect?

A: Kill off the higher frequencies before resampling

** Wikipedia*

Resampling Operations: Decimation

- Decimation corresponds to subsampling from every other row/column in the image
- This reduces the image size by a factor of 4
- To eliminate potential aliasing effects, we first convolve the image with a low pass filter (*e.g.*, a Gaussian kernel) to kill off high frequency signal components

Decimation Example

*

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

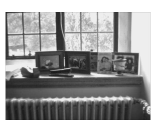
Filter

Subsample

Resampling Operations: Expansion

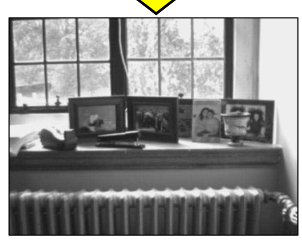
- Expansion corresponds to increasing the image size by inserting a new row/column between each pixel
- This increases the image size by a factor of 4
- To fill in the new pixels, the values are interpolated from the current image
  - Nearest neighbor
  - Bilinear
  - Bicubic, etc.

Expansion Example



Expand Image

Interpolation



Interpolation Techniques

- Nearest Neighbor (in this case pixel duplication)

20	200	0	210	0	211	11
20	0	0	0	0	0	11
20	205	0	215	0	215	15
20	0	0	0	0	0	15
20	206	0	221	0	219	19

Bilinear Interpolation

200	0	210	0	211
0	0	0	0	0
205	0	215	215	215
0	0	0	0	0
206	0	221	0	219

- Linear Interpolation
  - (215 + 215)/2 = 215

Bilinear Interpolation

200	0	210	0	211
0	0	0	0	0
205	0	215	0	215
0	0	218	0	0
206	0	221	0	219

- Linear Interpolation
  - (215 + 221)/2 = 218

Bilinear Interpolation

200	0	210	0	211
0	0	0	0	0
205	0	215	0	215
0	0	0	217	0
206	0	221	0	219

- Bi-Linear Interpolation
  - (215 + 221 + 215 + 219)/4 = 217

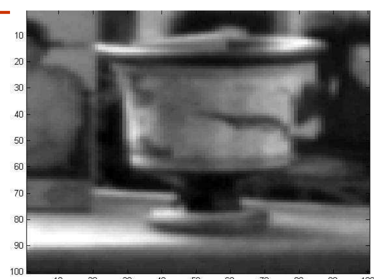
Bilinear Interpolation Implementation

$$\begin{bmatrix} 200 & 0 & 210 & 0 & 211 \\ 0 & 0 & 0 & 0 & 0 \\ 205 & 0 & 215 & 0 & 215 \\ 0 & 0 & 0 & 0 & 0 \\ 206 & 0 & 221 & 0 & 219 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} / 4$$

- Bilinear interpolation can be compactly implemented through a convolution operation with an appropriate averaging filter

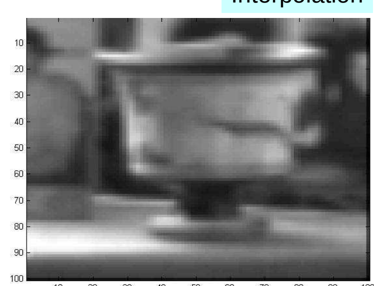
Interpolation Results

Original Image



Nearest Neighbor

Bilinear Interpolation



The 2-D Discrete Fourier Transform Revisited

- Since our images are nothing more than 2D discrete functions, we are interested in the 2D DFT

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

for $u=0,\ldots,M-1$ and $v=0,\ldots,N-1$ and the iDFT is defined as

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$$

Spatial Domain

- In the spatial domain, a single point corresponds to the integration of all contributing frequencies at that position

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+vy)}dudv$$

- Position is known well, but contributed frequency is "unlimited"

Frequency Domain

- In the frequency domain, a single point corresponds to the strength of a single frequency

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy$$

- Each point is influenced by the intensity of ALL points in space
- Frequency is known well, but spatial contributions are "unlimited"

Pyramids

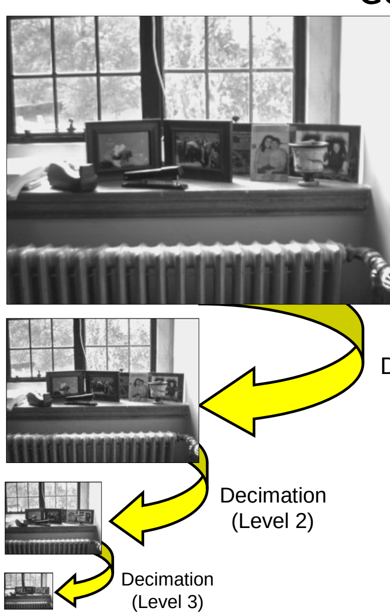
- Pyramids are an example of a *multi-resolution* representation of the image
- Pyramids separate information into frequency bands
- In the case of images, we can represent high frequency information (textures, etc.) in a finely sampled grid
- Coarse information can be represented in a coarser grid (lower sampling rate acceptable)
- Thus, coarse features can be detected in the coarse grid using a small template size
- This is often referred to as a multi-resolution or multi-scale resolution

Wavelength Domain

- Pyramid correspond to a mixture of the spatial and frequency domains
- Position information at each level is known to the accuracy of that grid resolution
- Contributing frequencies are bandwith limited at each grid resolution

CSE397/497 Real-time Image Processing

Generating the Gaussian Pyramid



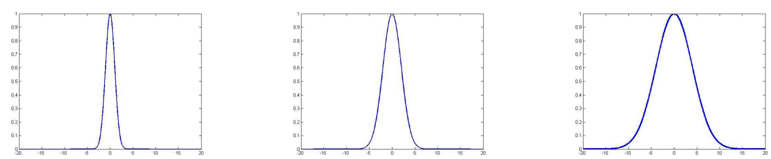
The Number of levels is a function of the application (3- 5 is typical)

Gaussian Pyramid Generation

- For each decimation, we convolve with the same size Gaussian kernel, *e.g.*

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

- While the width of this filter stays fixed, its effective width actually doubles at each pyramid level



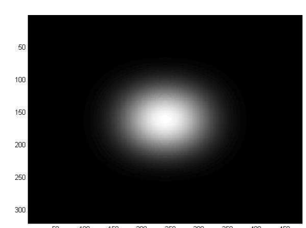
Level 1 Level 2 Level 3

Recall our Procedure for Filtering in the Frequency Domain

3. Multiply F(u,v) by a filter G(u,v)

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

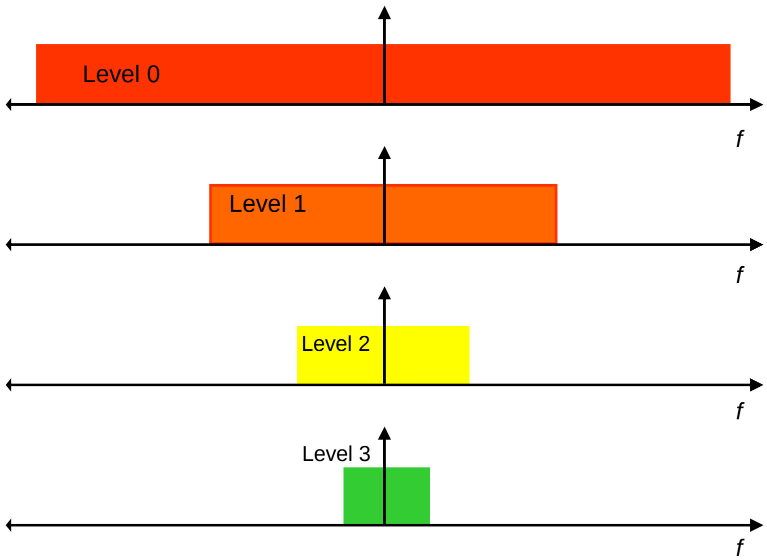
g(x,y)



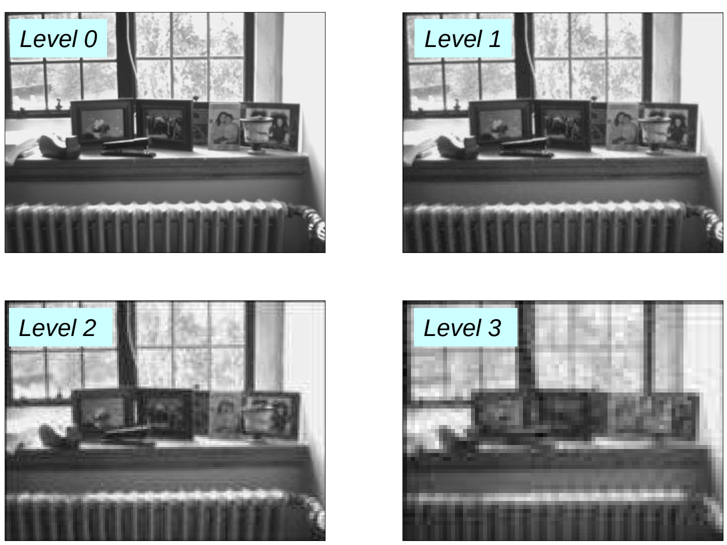
G(u,v)

- As the kernel size in the spatial domain *increases*, it *decreases* in the frequency domain
- As a consequence, only lower frequencies are passed through (low-pass filtering)

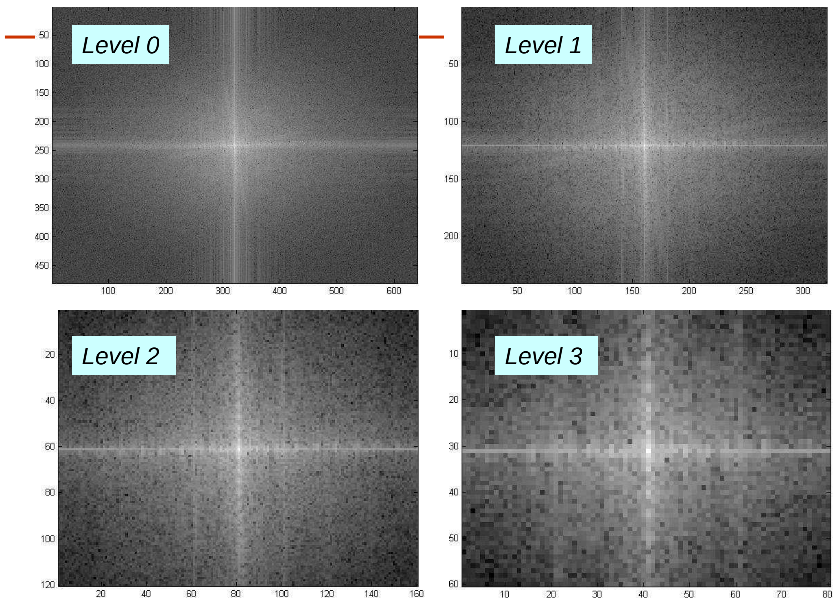
Gaussian Pyramid Frequency Composition



Pyramid Levels at Same Resolution



Discrete Fourier Transforms (Gaussian)



Employing Gaussian Pyramids for our 2D Pattern Recognition Example

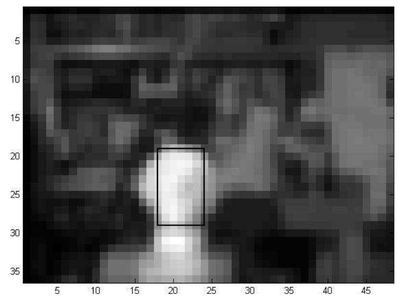
Template

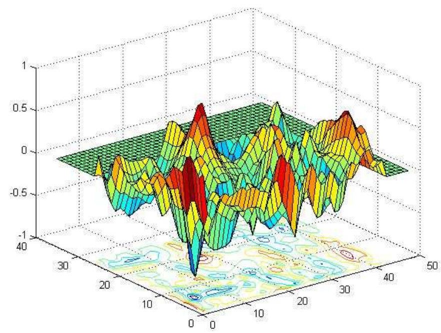
Search Region

Original Image

Level 3 Search

- At the lowest pyramid level, we search the entire image with the correlation template

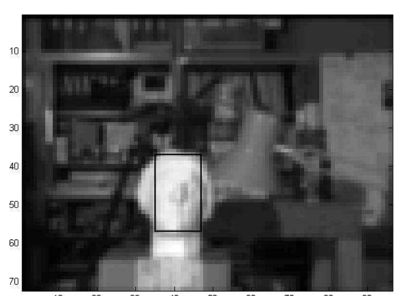


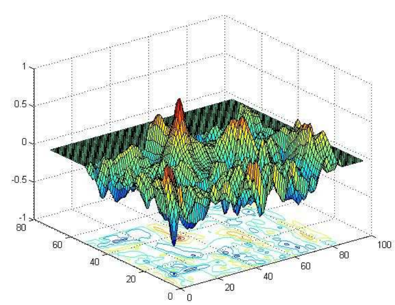


CSE397/497 Real-time Image Processing

Level 2 Search

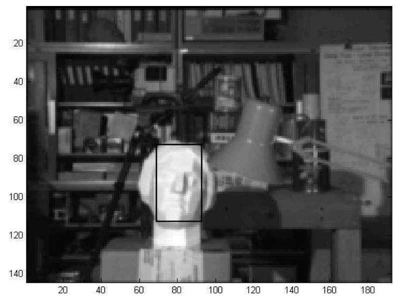
- Subsequent searches are constrained to a neighborhood of only several pixels in the x and y directions

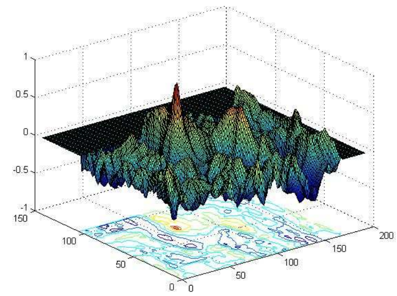




Level 1 Search

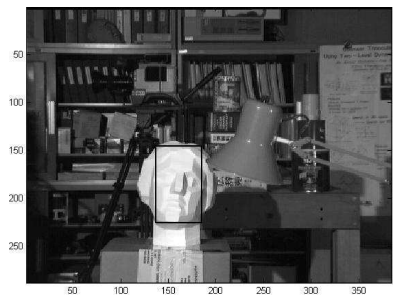
- Subsequent searches are constrained to a neighborhood of only several pixels in the x and y directions

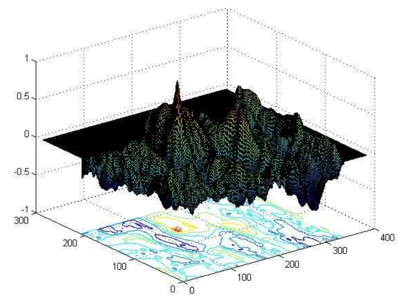




Level 0 Search

- In the end, the total time (in Matlab) was reduced from ≈ 31 seconds to ≈ 0.5 seconds while obtaining the same template match





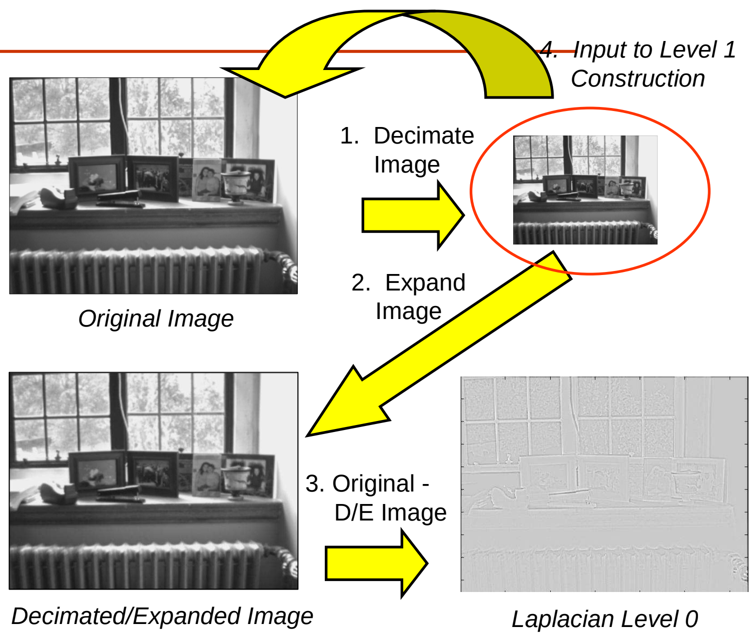
Size of Search Neighborhood

- In theory, each match should be off by no more than 1 pixel in either the x or y direction

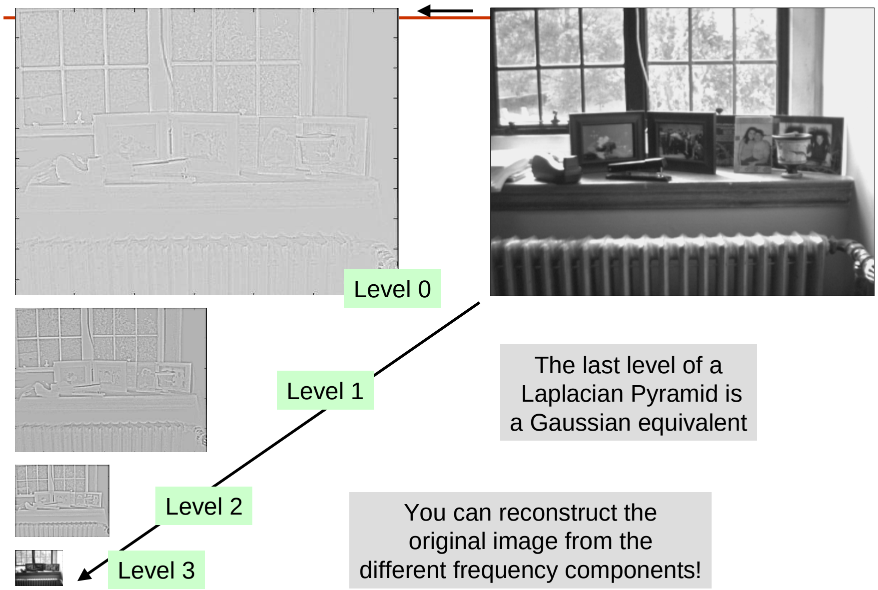
	Predicted	Actual
Level 3		(18,19)
Level 2	(36,38)	(35,37)
Level 1	(70,74)	(69,73)
Level 0	(138,146)	(137,146)

- A larger neighborhood can be used if so desired

Constructing the Laplacian Pyramid



The Laplacian Pyramid

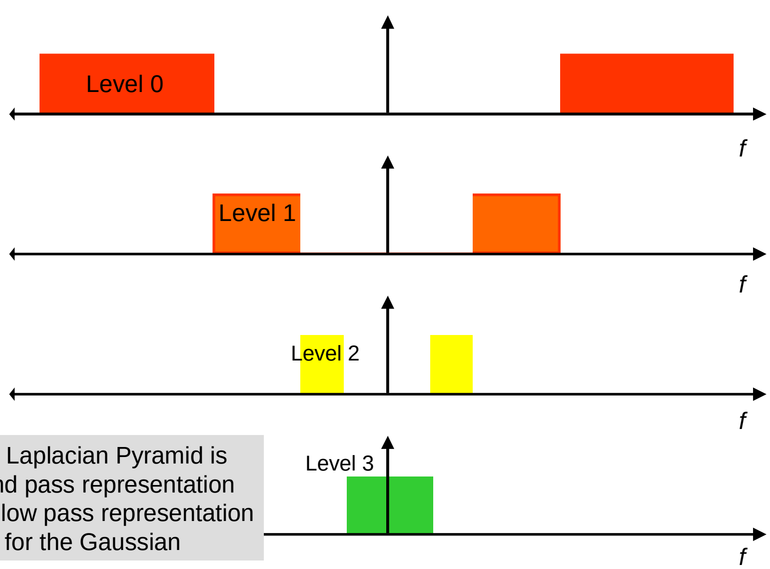


The last level of a Laplacian Pyramid is a Gaussian equivalent

You can reconstruct the original image from the different frequency components!

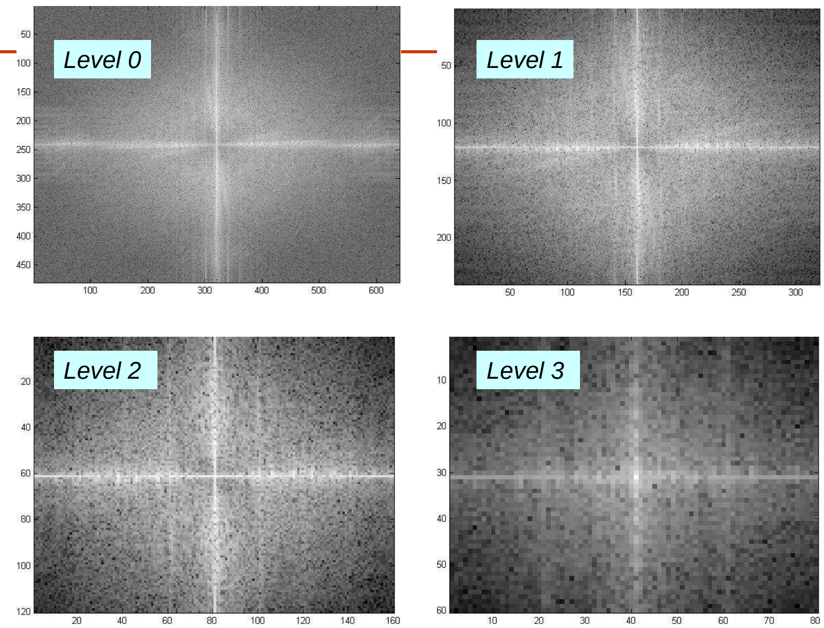
CSE397/497 Real-time Image Processing

Laplacian Pyramid Frequency Composition



The Laplacian Pyramid is a band pass representation vice a low pass representation for the Gaussian

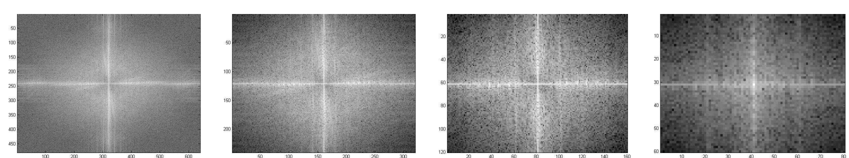
Discrete Fourier Transforms (Laplacian)



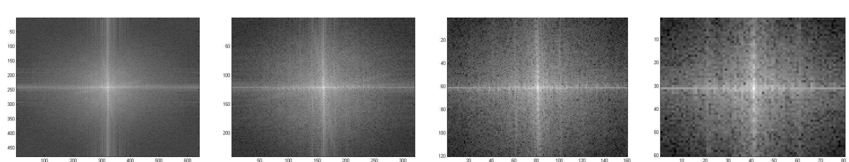
CSE397/497 Real-time Image Processing

Discrete Fourier Transforms

Level 0 *Level 1* *Level 2* *Level 3*



Laplacian



Gaussian

Laplacian Pyramid Application: Image Fusion

- Because Laplacian pyramids are band pass representations, each level represents a different energy component that forms the image
- By maximizing the energy contributions of 2 images taken from different cameras (*e.g.* IR and day cameras), we can fuse the images and improve feature recognition as a result
- http://www.ece.lehigh.edu/SPCRL/

Summary

- Pyramids correspond to a decomposition of an image into spatial/frequency bands
- Higher frequency require a larger image (sample) size to be represented, while lower frequency can be accommodate in a coarse image
- This "coarse to fine" approach can provide tremendous increases in computational efficiency
- Laplacian pyramids correspond to a band pass while Gaussian a low pass representation
- Pyramids are used in many applications beyond target tracking and image fusion
  - Image alignment
  - Mosaicing
  - Blending images
  - Data compression, etc.