Image Pyramids

- I've motivated the DFT as a transformation that allows you to see different aspects of the data
- Today we will look at a different transformation that gives you easier access to different kinds of information
- Specifically we will look at scale information
Why Scale Matters

• In any algorithm, you will only be able to examine a certain number of pixels
  – Let's say 64x64

• Task 1: What is this?
Can you tell now?
How about now?
New Task

• How many whiskers does the zebra have?
Now you need high-resolution info
Image Pyramids

- Represent the image using a collection of images
- Called a pyramid because the resolution of the images decreases
- Most common pyramid: Gaussian pyramid
  - At each level, generate the next level by blurring the image with a Gaussian, then downsampling
Each level of the pyramid represents the image if it were blurred with a Gaussian. The levels vary by the std. dev. of the Gaussians.
When would it be useful?

- Useful whenever you need to work at multiple scales
  - Looking for an object that could be close or far
- Can eliminate distractions
What's wrong with the Gaussian pyramid?

- It is redundant
- Each level contains all of the low-frequencies that are available at the lower levels.
Another Pyramid: The Laplacian pyramid

Original Image

Reduce

Enlarge

Original Image

Shrunk and Enlarged Version
What is this image?

- This image represents everything in the high-resolution image that cannot be represented in a low-resolution image.
Each level captures a band of spatial frequencies
What about orientation?
If you're interested

- Good tutorial on image pyramids at
  - http://www.cns.nyu.edu/~lcv/software.html
- Look for matlabPyrTools
Application: Texture Synthesis

- Basic Problem
Heeger and Bergen: SIGGRAPH95

• Basic Assumption
  – Take two texture images
  – Decompose them into a steerable pyramid
  – If the histogram of the pyramid coefficients are similar, then the textures will appear similar
    • Based on studies of human vision, see (Bergen and Adelson 88) or (Malik and Perona 89)
Basic algorithm

- Decompose current image into steerable pyramid
- Modify each pyramid image so that its histogram matches the histogram of the corresponding image from the sample
- Invert the pyramid to recover the current texture image
- Repeat until the image converges
Start with Noise

From (Heeger and Bergen 95)
Results

Input  Synthesized

From (Heeger and Bergen 95)
Results

From (Heeger and Bergen 95)
Failures

From (Heeger and Bergen 95)
Texture beyond histograms

- This approach, while interesting, ended up not being used widely in the graphics community
- Researchers found that they could generate more visually pleasing textures by replicating patches of texture in a smooth fashion
  - Efros and Leung – 1999
  - Efros and Freeman – 2003
- This approach is still very interesting from an analysis point of view
Application 2: Removing Noise

(From Adelson and Simoncelli - 1996)
Two Key Questions

- How do we characterize what makes these image look different?
- How do we use these differences to remove the noise?

(From Adelson and Simoncelli - 1996)
We can use the steerable pyramid

Figure 1. Histograms of a mid-frequency subband in an octave-bandwidth wavelet decomposition for two different images. Left: The “Einstein” image. Right: A white noise image with uniform pdf.

(From Adelson and Simoncelli - 1996)
What do these histograms tell us?

- Pyramid coefficients from images: Usually zero, but big sometimes
- Noise coefficients: Usually close to zero, very rarely big

(From Adelson and Simoncelli - 1996)
Question

- What would my estimate of the coefficient be if:
  - It had a big value?
  - It had a small value

(From Adelson and Simoncelli - 1996)
Coring

- Can express that as a function

This estimator minimizes the squared error in the estimate of the coefficients (From Adelson and Simoncelli - 1996)

Figure 2. Bayesian estimator (symmetrized) for the signal and noise histograms shown in figure 1. Superimposed on the plot is a straight line indicating the identity function.
Basic Algorithm

- Decompose Noisy Image into Steerable Pyramid
- Run each pyramid image through the non-linearity
- Reconstruct the pyramid
Results

(From Adelson and Simoncelli - 1996)
Denoising

- One of the state-of-the-art algorithms (Portilla, Strela, Wainwright, and Simoncelli) extends this basic idea
  - Uses a better estimator of derivative values
Fast Filtering

- All of these operations require a lot of filtering
- The FFT is one trick to making it faster.
- We can do more by choosing the filters correctly
- Technique called separable filtering
Let's count operations

- Let's filter with a very big filter – an 81x81 Gaussian filter
- How many multiplications and additions?
Let's see if we can construct this filter in a more efficient fashion

- Construct a 1-D filter
- Can think of this as the middle row from the last filter
Now, let's do a convolution
The result

Result from 2 1-D convolution

Original Filter
What does this mean?

- Using associativity, we can get the same result by filtering with a 81x81 filter as with convolving by two 1x81 filters
- 81x81 filter=81*81=6561 additions and multiplications
- 2 1x81 filters = 2*81=162 operations
- Much faster
When can we use this trick?

- Not all filters can be decomposed in this way.
- We can look at the Singular Value Decomposition (We'll talk more about this later)

\[
K = \sum_i \sigma_i u_i v_i^T
\]

- We can approximate any filter as the summation of separable filters