[ASIDE: I did not have time to discuss noise last lecture, so I will add it in here.]

**Noise**

Often signals contain additive noise. We would like to know the result when the noisy signal is convolved with some filter \( h(x) \). Because of the distributive property of convolution, convolving a noisy image with \( h(x) \) gives the same result as convolving the image and noise separately with \( h(x) \) and then adding the result together. So, it is meaningful to examine the noise on its own. In particular, what can we say about the frequency components of a noise signal \( n(x) \)?

If the noise \( n(x) \) at different pixels is independent\(^1\) then one can show that its Fourier transform satisfies:

\[
| F n(x) |^2 = | \hat{n}(k) |^2 = \text{constant}
\]

where the constant depends on \( \sigma_n^2 \). Rather than proving this mathematically, I will just give an example.

See the code: [http://www.cim.mcgill.ca/~langer/546/MATLAB/plot1DWhiteNoise.m](http://www.cim.mcgill.ca/~langer/546/MATLAB/plot1DWhiteNoise.m)

**Filtering and bandwidth**

Suppose we convolve an image \( I(x) \) with a function \( h(x) \). We have referred to \( h(x) \) as an impulse response function. \( h(x) \) is also called a linear filter. Recall that the Fourier transform of the filter \( h(x) \) can be written

\[
\hat{h}(k) = |\hat{h}(k)| e^{i\phi(k)}
\]

where \( |\hat{h}(k)| \) is called the amplitude spectrum and \( \phi(k) \) is called the phase spectrum. By the convolution theorem,

\[
F \{ I(x) \} = F \{ I(x) \ast h(x) \} = \hat{I}(k) |\hat{h}(k)| e^{i\phi(k)}
\]

\(^1\)In fact, one only requires the noise is uncorrelated.
and so the amplitude $|\hat{h}(k)|$ of the filter amplifies frequency component and the phase $\phi(k)$ of the filter shifts each frequency component.

We can characterize filters by how they affect different frequencies. We will concern ourselves mainly with the amplitude spectrum for now. Let’s first address the case of “ideal” filters. We say:

- $h(x)$ is an ideal low pass filter if there exists a frequency $k_0$ such that
  
  $$
  \hat{h}(k) = \begin{cases} 
  1, & 0 \leq k \leq k_0 \\
  0, & k_0 < k \leq \frac{N}{2}
  \end{cases}
  $$

- $h(x)$ is an ideal high pass filter if there exists $k_0$ such that
  
  $$
  \hat{h}(k) = \begin{cases} 
  0, & 0 \leq k < k_0 \\
  1, & k_0 \leq k \leq \frac{N}{2}
  \end{cases}
  $$

- $h(x)$ is an ideal bandpass filter if there exists two frequencies $k_0$ and $k_1$ such that
  
  $$
  \hat{h}(k) = \begin{cases} 
  0, & 0 \leq k < k_0 \\
  1, & k_0 \leq k \leq k_1 \\
  0, & k_1 < k \leq \frac{N}{2}
  \end{cases}
  $$

Note that these definitions above only concern $k \in \{0, \ldots, \frac{N}{2}\}$. Frequencies above $k = \frac{N}{2}$ are ignored in the definition because the values of $\hat{h}(k)$ of these frequencies are determined by the conjugacy property. (Recall that the conjugacy properties applies only if the filter is real valued function. )

**Non-ideal filters**

We typically work with filters that are not ideal i.e. filters that only approximately satisfy the above definitions. If we have an approximately bandpass filter, then we would like to describe the width of this filter i.e. the range of frequencies that it lets through. One often does this by considering the frequencies at which $|\hat{h}(k)|$ reaches half its maximum value. The bandwidth at half-height is defined to be $k_1 - k_0$, where $k_0 < k_1$ and

$$
|\hat{h}(k_0)| = |\hat{h}(k_1)| = \frac{1}{2} \max_{k \in [0, \frac{N}{2}]} |\hat{h}(k)|
$$
Bandwidth can also be defined in terms of the ratio of \( k_1 \) to \( k_0 \), specifically, the octave bandwidth at half height is:

\[
\log_2 \left( \frac{k_1}{k_0} \right) = \log_2(k_1) - \log_2(k_0)
\]

For example, a filter with a bandwidth of one octave means that the \( k_1 \) frequency is twice the \( k_0 \) frequency.

**Example: Blur**

Low pass filters attenuate the high frequency components of a signal. An example of a non-ideal low pass filter is the blur function \( B(x) \) that we saw earlier. What happens if we convolve an image repeatedly – say \( m \) times – with \( B(x) \)? Check for yourself, for example, that for \( m = 2 \),

\[
B(x) * B(x) = \begin{cases} 
\frac{1}{16}, & x \pm 2 \\
\frac{1}{4}, & x = \pm 1 \\
\frac{3}{8}, & x = 0 \\
0, & \text{otherwise}
\end{cases}
\]

From the convolution theorem, for the case of general \( m \), we have

\[
\hat{F}(I(x) * B(x) * \cdots * B(x)) = \hat{I}(k) \hat{B}(k)^m = \left( \frac{1}{2(1 + \cos(\frac{2\pi}{N}k)))} \right)^m
\]

The case of \( m = 1, 2 \) are sketched in the slides. As \( m \) is increased, more frequencies become attenuated and only the very low frequencies are left in the signal. See the Matlab code:

[http://www.cim.mcgill.ca/~langer/546/MATLAB/plotB_1D.m](http://www.cim.mcgill.ca/~langer/546/MATLAB/plotB_1D.m)

which generates the figure


which shows \( B(x) * \cdots * B(x) \) and its Fourier transform, for various \( m \).

**Example 2: Gaussian, DOG**

A more common way to blur is to use a Gaussian function

\[
G(x, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}
\]

If we sample this function on the integers, and take the Fourier transform of the sampled (discretized) function, we get the following approximation:

\[
\hat{G}(k, \sigma) \approx e^{-\frac{1}{2} \left( \frac{2\pi}{N} \right)^2 \sigma^2 k^2}
\]

This approximation becomes exact in the limit as \( N, \sigma \to \infty \), with \( \frac{\sigma}{N} \) held constant. (This amounts to taking the continuous instead of discrete Fourier transform. The proof of these claims are beyond the scope of this course.)

If you wish to see this approximation for yourself, run the Matlab script

last updated: 12th Mar, 2017
A few key properties to notice are:

- If the standard deviation of the Gaussian in the space \((x)\) domain is \(\sigma\) then the standard deviation of the Gaussian in the frequency \((k)\) domain is proportional to \(\frac{1}{\sigma}\).

- \(\hat{G}(k, \sigma)\) has a Gaussian shape, but it does not integrate to 1, namely there is no scaling factor present. The max value occurs at \(k = 0\) and the max value is always 1.

- The Fourier transform is periodic, with period \(N\). This is always true.

Since the Fourier transform of a Gaussian always has the value 1 at \(k = 0\), a difference of two Gaussians (DOG), \(G(x, \sigma_1) - G(x, \sigma_2)\), where \(\sigma_1 \neq \sigma_2\) will have a Fourier transform that is 0 at \(k = 0\) but is non-zero for \(k \neq 0\). See sketch below. This should make sense intuitively because this frequency component is just a constant image, and blurring a constant image has no effect. Finally, note that DOG’s are (non-ideal) bandpass filters.

Example 3: Gabor

Recall a Gabor function is defined by multiplying a cosine function and a Gaussian:

\[
\cosGabor(x, k_0, \sigma) \equiv G(x, \sigma) \cos\left(\frac{2\pi}{N} k_0 x\right)
\]

we define a \(\sin\) Gabor similarly:

\[
\sinGabor(x, k_0, \sigma) \equiv G(x, \sigma) \sin\left(\frac{2\pi}{N} k_0 x\right)
\]

What is the Fourier transform of a Gabor? To answer this question, we use property which is similar to the convolution theorem:

\[
\mathbf{F}(I(x)h(x)) = \frac{1}{N} \mathbf{F} I(x) \ast \mathbf{F} h(x).
\]
Thus the Fourier transform of a cosine Gabor is the convolution in the frequency domain of the Fourier transforms of a Gaussian and the Fourier transform of a cosine:

\[ F \text{cosGabor}(x, k_0, \sigma) = F \{ G(x, \sigma) \ast \cos\left(\frac{2\pi}{N}k_0x\right) \} \]

\[ = e^{-\frac{1}{2} \left(\frac{2\pi \sigma}{N}k\right)^2} \ast \frac{N}{2}(\delta(k_0-k) + \delta(k_0+k)) \]

\[ = e^{-\frac{1}{2} \left(\frac{2\pi \sigma}{N}(k-k_0)\right)^2} + e^{-\frac{1}{2} \left(\frac{2\pi \sigma}{N}(k+k_0)\right)^2} \]

which is the sum of two Gaussians, centered at \( k = \pm k_0 \). The Fourier transform of a sine Gabor can be calculated similarly. (See Exercises.)

Convolving a function \( I(x) \) with a cosine or sine Gabor filters the function such that only a band of frequencies remains, namely the frequencies near the center frequency \( k_0 \) of the Gabor. The width of the band depends on the \( \sigma \) of the Gaussian. We will return to this idea in the 1D case when we discuss audio signals.

### 2D Fourier transforms

As in the 1D case, the 2D Fourier transform allows one to express an image \( I(x, y) \) as a sum of 2D sines and cosines. For any \( N \times N \) function \( I(x, y) \) with \( x \) and \( y \) in \( \{0, 1, \ldots, N-1\} \), the 2D Fourier transform is defined:

\[ \hat{I}(k_x, k_y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I(x, y) e^{-i\frac{2\pi}{N}(k_xx + k_yy)} \]

It is the inner product of an image \( I(x, y) \) with a set of 2D cosines and sines of different frequencies \( (k_x, k_y) \).

The 2D inverse Fourier transform is defined similarly:

\[ I(x, y) = \frac{1}{N^2} \sum_{k_x=0}^{N-1} \sum_{k_y=0}^{N-1} \hat{I}(k_x, k_y) e^{i\frac{2\pi}{N}(k_xx + k_yy)} \]

The 2D Fourier and inverse Fourier transforms have the same properties as the 1D Fourier transform which you can prove on your own if you are interested. The **convolution theorem** holds:

\[ F (I(x, y) \ast h(x, y)) = \hat{I}(k_x, k_y) \hat{h}(k_x, k_y) \]

and similarly the second version of the convolution theorem mentioned earlier in the lecture today:

\[ F (I(x, y)h(x, y)) = \frac{1}{N^2} \hat{I}(k_x, k_y) \ast \hat{h}(k_x, k_y) \]

The periodicity and the conjugacy properties also hold. The proofs of are exactly the same as in the 1D case. Assuming \( h(x, y) \) is a real value function defined on an \( \{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, N-1\} \), then

\[ \hat{h}(k_x, k_y) = \hat{h}(k_x + m_xN, k_y + m_yN) \quad \text{(periodicity)} \]

and

\[ \hat{h}(k_x, k_y) = \overline{\hat{h}(N - k_x, N - k_y)} \quad \text{(conjugacy)} \]
2D Filtering

Let’s look at some examples of filtering. In the first example, we will filter a small $64 \times 64$ image with *ideal* bandpass filters that are approximately one octave. The bands are illustrated below on the left. The filter $\hat{h}(k_x, k_y)$ has value 1 over the band and value 0 outside the band. The one on the bottom right is non-zero for frequencies $|(k_x, k_y)| \in 16, ..., 31$.

The plot on the right shows $I(x,y) * h(x,y)$ for the filters defined above. The images are the sums of 2D sines and cosines for the ranges of frequencies shown above.

Let’s next look at non-ideal 2D filters. First consider a 2D Gaussian filter. Just as in the 1D case, the Fourier transform of a 2D Gaussian is a 2D Gaussian in the frequency domain. (See Exercises.) For a large Gaussian, say $\sigma = 16$, we get a small Gaussian in the frequency domain and for a small Gaussian $\sigma = 1$ we get a large Gaussian in the frequency domain. This can be understood with the convolution theorem. Convolving an image with a Gaussian blurs the image, so if a big Gaussian is used then this will average over a wide image region. For high frequency sine and cosine functions, the max and min will sum and cancel out. In the frequency domain, all components of the original image will be attenuated to near 0, except for the low frequency sine and cosines. That is why there is only a small Gaussian shown below for the $\sigma = 16$ case. For the $\sigma = 1$ case, there is little blurring, and only the very high frequency sine and cosine components will be attenuated. For medium and low frequency sinusoids, the convolution with the Gaussian has little effect.

The filtered image $I(x,y) * G(x,y, \sigma)$ are shown on the right below. One subtle point about these figures is how they were computed in Matlab using the convolution theorem. The proof of the convolution theorem treats both functions as *periodic*. For the image, this means that to blur beyond the image boundary one does *not* pad with zeros. Instead, one ‘tiles’ the infinite image plane with the same image. For example, in the blurred images shown below, the intensities of points shown near the left boundary are averages of intensities of points near the left *and right* image boundaries. Of course this gives ‘invalid’ results. But are they any less invalid than if we were to pad with zeros? Arguably not.

The last example shows 2D non-ideal bandpass filters, namely difference of Gaussians. The
DOGs were defined for $\sigma$ and $\sigma \times 1.1$ for a range of $\sigma$ values indicated in the subpanels. The filtered images $I(x, y) \ast DOG(x, y)$ are similar to the ones for the ideal bandpass filters on the previous page, but there are differences owing to slight differences in the filter details.

Finally, in the lecture, I showed the same figures from lecture 5 where an image was cross-correlated with Gabors of different orientations. Recall that cross-correlation is very similar to convolution, and convolution will produce similar figures. I plan to have a question on Assignment 4 that deals with convolving oriented filters with Gabors, so I will not provide details here.