9.35 Sensation And Perception

Spring 2009

```
function [spect, time, freq] = mySpectrogram(y, Fs, takeLog)
% [spect, time, freq] = mySpectrogram(y, Fs)
%   Creates an image of the spectrogram of sound vector y

if ~exist('takeLog', 'var')
   takeLog = false;
end
if size(y,2)==2
   y = mean(y,2);
end

%% 1a - properties
window = round(Fs/20);     % Number of samples required for 1 cycle at 20Hz
% Minimum Fs to sample at 20kHz = 2*20kHz = 40kHz

%% 1b - spectrogram
timestep = round(window/2);             % Indeces to jump by
steps = 1:timestep:(length(y)-window);  % Indeces at start of each slice
time = (steps+window/2)/Fs;             % Time at center of each slice (s)
% Cycles/s = cycles/sample * samples/s
freq = Fs*(0:(window-1))/window;        % Frequencies returned by fft

% Index of the valid frequencies
valid = freq>=20 & freq <= min(5000, Fs/2);
freq = freq(valid);

% Creates empty spect matrix
spect = zeros(length(freq), length(steps));   % Output

% Runs the spectrogram
for i = 1:length(steps)
   myslice = y(steps(i):(steps(i)+window-1));
   f = abs(fft(myslice));
   spect(:, i) = f(valid);
end

if takeLog
   sim = log(spect);
else
   sim = spect;
end
imagesc(time, freq, sim);
```

```
xlabel('Time (s)')
ylabel('Frequency (Hz)')
set(gca, 'YDir', 'Normal')
colorbar
end
```