Automatic Discernment System of Country and Rap Music Genres

Linda Sun and Amrita Mazumdar
ELEN 4810 Digital Signal Processing – Fall 2012

1. Intro

Musical genres are subjective labels to categorize music based on the style and culture of the time. The interpretation of genres and what defines one changes over time, which thus makes the problem of automatic genre classification more difficult. Determining appropriate features characteristic of a genre of music is in itself a complex and subjective problem, but these features can be used to great benefit in music information retrieval schemes. This project explores the problem of classifying music from two different styles of popular music, rap and country. Certain characteristics of each of the two genres of music were exploited to develop a signature of testing features, which could then be used to automatically classify other future music samples. The features chosen were rhythmic structure, noisiness of timbre, and frequency-component distribution. This implementation considers each sample as a uniform selection of a certain genre, and can only classify whole samples.

2. Problem Specification & Feature Determination

Our goal was to have the program be able to automatically distinguish between the two genres. We first did research and came upon one of the most influential papers in the area: “Musical Genre Classification of Audio Signals” by George Tzanetakis and Perry Cook. We decided to use multiple features to compare country and rap music rather than testing just one characteristic. Based on our initial analysis of spectrograms and algorithms used by Tzanetakis et al., we picked out three aspects we felt were the most important to analyze: RMS power, beat histograms, and zero crossings.

2.1 Training Data & Selection

We chose two songs in each genre, taking care to choose different genders and tempos. We picked “Beer Money” by Kip Moore, “Redneck Woman” by Gretchen Wilson, “Can I Get A...” by Jay-Z, and “Did It On’em” by Nicki Minaj. Moore and Wilson’s songs were more instrumented with many of background guitars in a typical country style. Jay-Z and Minaj’s raps were mostly spoken word against a beat or quiet melody. We used 15 second samples of each song in order to cut out parts of irregularity, such as random screams in Minaj’s song.

2.2 Signal Analysis & Feature Determination

We used the Matlab function specgram to create spectrograms of all four songs.

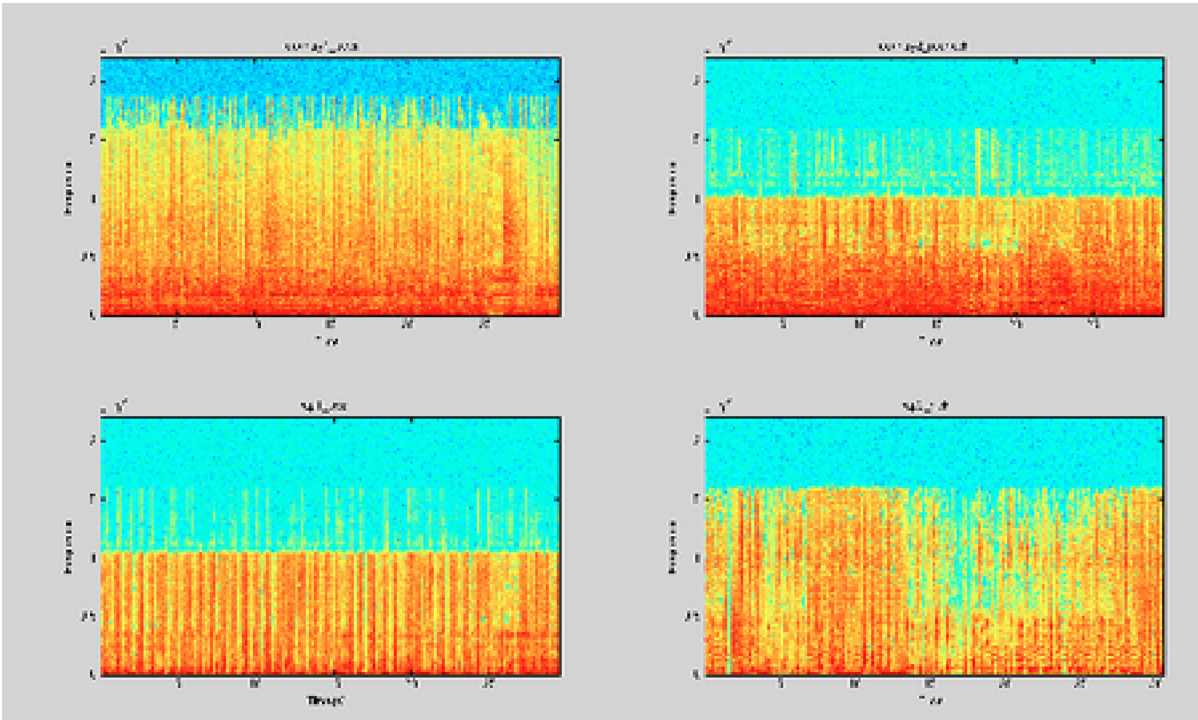


Figure 1: Spectrograms of our testing samples

Zero crossings are used in speech and music discrimination because they occur more often in speech-based signals. Since rap is similar to speech, but with a musical component, we decided to test this field to distinguish it from country music.

Beats per minute (BPM) are a standard feature of a song's tempo, and it has been used in previous methods of discerning musical genres. We felt that rap music would have a higher BPM than country music, especially because modern rap music tends to use computer-generated electronic samples that would keep the BPM steady. For these reasons, we decided to analyze the beats.

The country spectrograms show more red concentrated at the bottom compared to the rap spectrograms. We hypothesized this meant that the country songs had greater RMS power and aimed to test this.

3. Feature Extraction

For each feature, we tried to develop a graph exploiting that feature to let us clearly differentiate between the two genres. We then tried to develop a quantifiable measure using that feature to use in testing.

3.1 Zero Crossing Rate

First we summed up the zero crossings across each frame and plotted the sum of the zero crossings against the frame number. We used 50 frames per second.

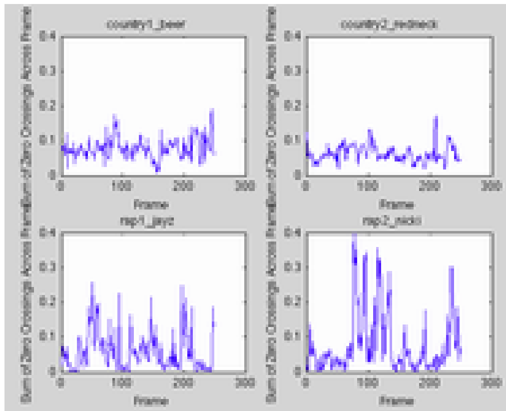


Figure 2: Plot of Zero Crossings vs Frame Rate

We observed that the graphs for country music were higher on the y axis than the graphs for the rap music. The rap music's graphs were noticeably closer to 0. To emphasize this difference, we decided to cut out all signals above a certain threshold y value. Below this threshold, if signals exist, the music is determined to be rap.

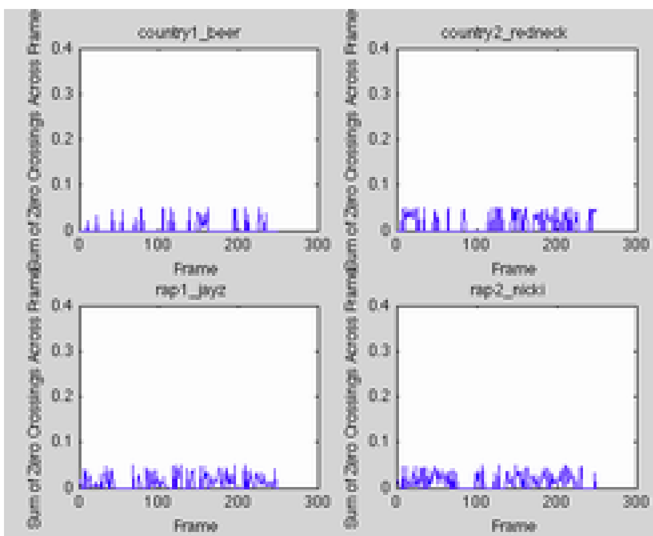


Figure 3: Zero Crossings Using a .05 Threshold

.05 is a poor threshold value because all of the signals look very similar and the country signals are not cut out.

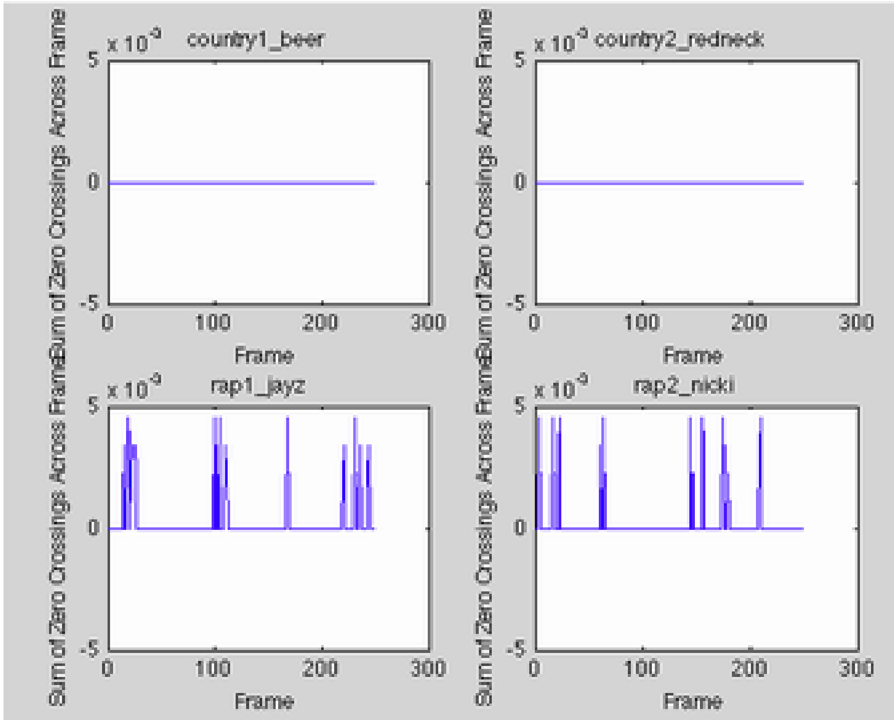


Figure 4: Zero Crossings Using a .005 Threshold

The country music signals are completely cut out. From the rap plots a considerable amount of signal remains. From this, we decided if the sum of zero crossings across a frame is nonzero below values of .005, the music signal is rap.

3.2 Beat Analysis

We saw that there were more emphasized beats for rap music than for country, so we tused the algorithm for beat histogram analysis from Tzanetakis et al. to develop our own distinguisher based on the beat features.

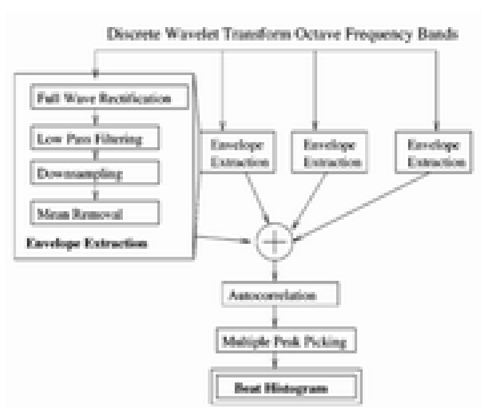


Figure 5: Beat Analysis Flow (adapted from Tzanetakis)

After implementing the steps envelope extraction and autocorrelation, we were left with a signal that was very dense and concentrated for country music, and a less dense, more spread-out signal for rap music. Repeated autocorrelation simply made the envelope smaller and more dense, for each signal.

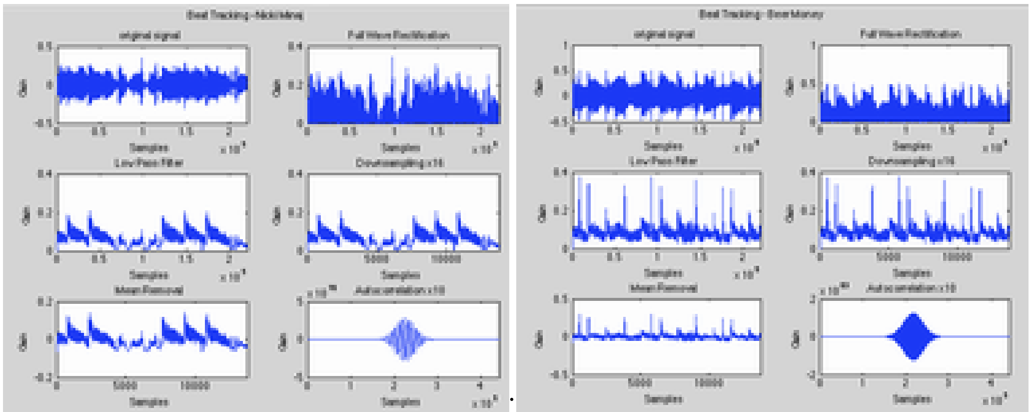


Figure 6: Beat Analysis for Nicki sample (rap, left) and Beer sample (country, right)

We determined that by averaging the top 5% of the peak values, we could come up with a “peak amplitude” for the signal. In theory, this peak amplitude would be much higher for signals with emphasized beats, because their more periodic nature would be emphasized.

In practice, however, computing autocorrelation over 10x simply became too computationally challenging, and we could not find a reasonable trend in the data after autocorrelation for the beat peaks to be a reasonable solution. Examining the autocorrelated graph more closely, we realized that the “higher density” in the signal we detected earlier was not a function of the peak size or strength, but really just the high frequency of zero crossings in the auto-correlated signal.

Looking at the graphs generated from each step of the beat histogram, we tried to make use of the earlier steps to find another trend, because we could clearly see variations between the rap and the country music signals in the beats. We found variations after the mean removal stage, where the rap signals varied more continuously while the country signals had very brief and distinct spikes aside from the DC component. We did a brief statistical analysis to determine whether we could quantify these changes simply based on the envelope-extracted signal alone, but the results of this test were also inconclusive.

Genre	Sample	Variance	Standard Deviation
Rap	Jay-Z	.0153	.0356
	Nicki	.0013	.1237
Country	Beer Money	.0013	.0360
	Redneck Woman	.0022	.0470

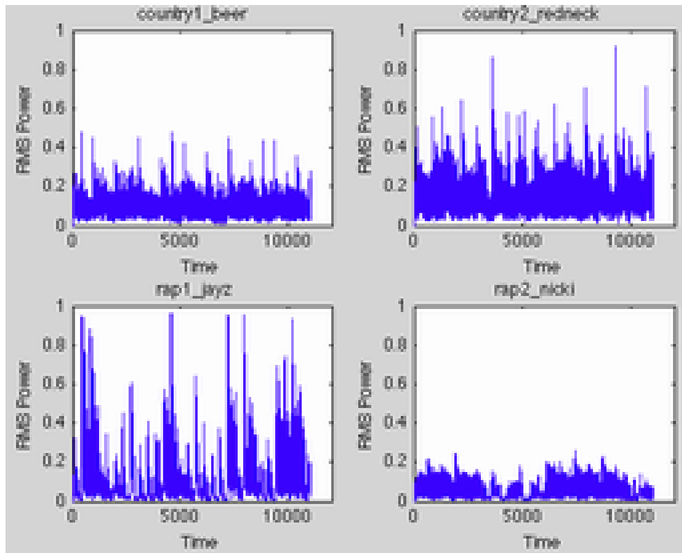
Table 1: Statistical Analysis of Signal Variance after Beat Analysis

What was curious in our statistical tests was that although the signals were clearly different for the Nicki and Beer Money samples, they demonstrated the same statistical variance.

We ultimately had to discard the beat tracking test, as we could not come up with a quantifiable measure of how the signals' beats varied.

3.3 RMS Power Analysis

We implemented a function that plotted the RMS power of the music against the length of the signal.



Plot of RMS Power over time

There were no clear trends in this data so we discarded the RMS test.

4. Execution & Decision System

The function FinalTest runs the zero crossing test on a given music sample. If the sample displays has any non-zero value in its y-value vector, it is rap. If it has only zero values as its y values after the cutoff has been in place, it is country.

4.1 Decision System

We had intended on using a k-nearest neighbor decision system. After eliminating RMS power and the beat histogram analyses as testing fields, we simply used the zero-crossing rate test to determine a song's genre. Since we are doing a binary categorization, a given sample will be categorized as either rap or country music, and never both.

4.2 Test Songs Used

We tested our system using 10 samples, with equally distributed 5 each of qualitatively chosen country and rap music samples. The songs were chosen from Grammy award winners in the genre, under the assumption that the most popular songs in a given genre are most representative of that genre's styling. Also, effort was taken to ensure any sample was not significantly older than the training data, as recording processes and outdated instrumentation may skew the results of such data.

Country songs: Last Name, Carrie Underwood; Sweet Thing, Keith Urban; How Long, Eagles; Not Ready to Make Nice, Dixie Chicks; Bless the Broken Road, Rascal Flatts

Rap songs: Not Afraid, Eminem; A Milli, Lil' Wayne; Crack A Bottle, Eminem; Money Maker, Ludacris; What You Know About That, T.I.

5. Results & Analysis

All songs were correctly classified. We found that our zero-crossings test had a 100% succes rate tested against songs.

6. Conclusion

Our test exceeded our expectations with 100% accuracy. However, we were only able to test ten songs. We were stringent about the songs we picked, so they had to be converted from youtube to mp3 and then clipped, which is a long process. We used 15 second clips because using the entire song caused the Matlab function to be very slow though the results were the same.

While our system seems effective, it tests only one trait instead of the three we intended on.

7. Future Work

In the future we will explore if a country music and rap music mp3 database exists so we can run tests more efficiently, without having to physically clip the songs ourselves.

We plan on reading more on past work with RMS power and beat histograms to determine whether we made mistakes in our analysis.

From qualitatively analyzing the spectrograms, we saw that the rap songs had the majority of its frequencies in a much lower range than the country songs. We aim to implement a test for this trait.

Once we have implemented these tests, we will use a k-nearest neighbor decision system to combine the results and determine whether the signal is rap or country.

8. Sources

"Musical Genre Classification of Audio Signals" *George Tzanetakis and Perry Cook*
IEEE Transactions on Speech and Audio Processing, 10(5), July 2002
Music Emotion Recognition, Yi-Hsuan Yang, Homer H. Chen, CRC Press, 2011
Signal RMS. Bolu Ajiboye. Matlab File Exchange. www.mathworks.com. 7 December 2012.

9. Matlab Code

a. TrainingData.m

```
% 15 Second Clips %

% COUNTRY %
% [d1a, sr1a] = mp3read('country1_beer.mp3');
[d1, sr1] = mp3read('country1_beerCUT.mp3');
[d1b, sr1b] = mp3read('country1_beerCUT.mp3', sr1*5);
subplot(2,2,1);
specgram(d1b(:,1),1024,sr1b);
title('country1\_beer');
ylim([0 4000]);

% [d2, sr2] = mp3read('country2_redneck.mp3');
[d2, sr2] = mp3read('country2_redneckCUT.mp3');
[d2b, sr2b] = mp3read('country2_redneckCUT.mp3', sr2*5);
subplot(2,2,2);
specgram(d2b(:,1),1024,sr2b);
title('country2\_redneck');
ylim([0 4000]);


% RAP %
% [d3, sr3] = mp3read('rap1_jayz.mp3');
[d3, sr3] = mp3read('rap1_jayzCUT.mp3');
[d3b, sr3b] = mp3read('rap1_jayzCUT.mp3');
subplot(2,2,3);
specgram(d3b(:,1),1024,sr3b);
xlabel('Time (s)');
title('rap1\_jayz');
ylim([0 4000]);


% [d4, sr4] = mp3read('rap2_nicki.mp3');
[d4, sr4] = mp3read('rap2_nickiCUT2.mp3');
[d4b, sr4b] = mp3read('rap2_nickiCUT2.mp3', sr4*5);
subplot(2,2,4);
specgram(d4b(:,1),1024,sr4b);
title('rap2\_nicki');
ylim([0 4000]);
```

b. FinalTest

```
function FinalTest(mp3signal)
[d, sr] = mp3read(mp3signal); %try to precut the signal

figure;
%%-----------------------Draw a spectogram
subplot(2,2,1);
specgram(d(:,1),1024,sr);
colorbar;
str1 = sprintf('Spectrogram: %s',mp3signal);
title(str1);
ylim([0 4000]);

%%------------------------------ZERO CROSSING
% find length of wav file
```

```
len_samp = length(d);
% Length of frame
frame_size = .02;
frame_length = round(sr*frame_size);
frames_per_sec = round(1/frame_size); % 50 frames per second
% Calculate number of zero-crossings in each frame
zcr = [];
n=1;
for frame = 1:frame_length:len_samp-frame_length
    frameData = d(frame:frame+frame_length-1);
    % Sum up zero crossings accross frame
    zcr(n) = 0;
    for i = 2:length(frameData)
    zcr(n) = zcr(n) + abs(sign(frameData(i)) - sign(frameData(i-1)));
    end
    zcr(n) = zcr(n)/(2*frame_length);
    n=n+1;
end
num_frames = length(zcr);
zcr(zcr>.005)=0;
subplot(2,2,2);
plot(1:1:num_frames,zcr)
%axis([0 300 -.005 .005]) %for cutoff
str2 = sprintf('Zero Crossings: %s',mp3signal);
title(str2);
xlabel('Frame');
ylabel('Sum of Zero Crossings Across Frame');

if any(zcr)
    signal_type = 'This is a rap music signal';
end
if ~any(zcr)
    signal_type = 'This is a country music signal';
end

disp(signal_type);
```

c. rms.m (from Matlab File Exchange)

```
%% DECLARATIONS AND INITIALIZATIONS

% Calculates windowed (over- and non-overlapping) RMS of a signal using the specified windowlength
% y = rms(signal, windowlength, overlap, zeropad)
% signal is a 1-D vector
% windowlength is an integer length of the RMS window in samples
% overlap is the number of samples to overlap adjacent windows (enter 0 to use non-overlapping windows)
% zeropad is a flag for zero padding the end of your data...(0 for NO, 1 for YES)
% ex. y=rms(mysignal, 30, 10, 1).  Calculate RMS with window of length 30 samples, overlapped by 10
samples each, and zeropad the last window if necessary
% ex. y=rms(mysignal, 30, 0, 0).  Calculate RMS with window of length 30 samples, no overlapping samples,
and do not zeropad the last window
%
% Author: A. Bolu Ajiboye

function y = rms(signal, windowlength, overlap, zeropad)

delta = windowlength - overlap;

%% CALCULATE RMS
```

```
indices = 1:delta:length(signal);
% Zeropad signal
if length(signal) - indices(end) + 1 < windowlength
    if zeropad
        signal(end+1:indices(end)+windowlength-1) = 0;
    else
        indices = indices(1:find(indices+windowlength-1 <= length(signal), 1, 'last'));
    end
end

y = zeros(1, length(indices));
% Square the samples
signal = signal.^2;

index = 0;
for i = indices
    index = index+1;
    % Average and take the square root of each window
    y(index) = sqrt(mean(signal(i:i+windowlength-1)));
end
```

d. BeatTracking.m

```
figure;

[d2, sr2] = mp3read('rap1_jayzCUT.mp3');
[d2b, sr2b] = mp3read('rap1_jayzCUT.mp3', sr2*5, 1);
% [d2, sr2] = mp3read('rap2_nickiCUT2.mp3');
% [d2b, sr2b] = mp3read('rap2_nickiCUT2.mp3', sr2*5, 1);
% [d2, sr2] = mp3read('country1_beerCUT.mp3');
% [d2b, sr2b] = mp3read('country1_beerCUT.mp3', sr2*5, 1);
% [d2, sr2] = mp3read('country2_redneckCUT.mp3');
% [d2b, sr2b] = mp3read('country2_redneckCUT.mp3', sr2*5, 1);

x = d2b;
subplot(3,2,1)
plot(x);
xlabel('Samples');
ylabel('Gain');
xlim([0 length(x)]);
title('original signal');
disp('sig done');

%% full wave rectification
y1 = abs(x);
subplot(3,2,2);
plot(y1);
xlim([0 length(y1)]);
xlabel('Samples');
ylabel('Gain');
title('Full Wave Rectification');
disp('full rect done');

%% low pass filtering

% simple low pass filter with one pole and alpha val of .99
% used to smooth envelope
% y[n] = (1-a)x[n] + a*y[n-1]
```

```
% y[n] - a*y[n-1] = (1-a)x[n]
% (1-az^-1)y[n] = (1-a)x[n]
% (1-a)/(1-az^-1) = H(z)
% a = .99 -> .01/(1-.99*z^-1) = H(z)

y2 = filter(.01, [1 -.99], y1);
subplot(3,2,3);
plot(y2);
xlim([0 length(y2)]);
title('Low Pass Filter');
xlabel('Samples');
ylabel('Gain');
disp('lp done');

%% downsampling -> for computational efficiency

y3 = downsample(y2, 16);
subplot(3,2,4);
plot(y3);
xlim([0 length(y3)]);
xlabel('Samples');
ylabel('Gain');
title('Downsampling x16');
disp('downsp done');

%% mean removal - center signal at zero
y4 = y3 - mean(y3);
subplot(3,2,5);
plot(y4);
xlim([0 length(y4)]);
xlabel('Samples');
ylabel('Gain');
title('Mean Removal');
disp('mean rem done');
% std(y4)

% %% autocorrelation
y5 = conv(y4, fliplr(y4));
a = conv(y5, fliplr(y5));
b = conv(a, fliplr(a));
c = conv(b, fliplr(b));
y7 = conv(c, fliplr(c));
% subplot(3,2,6);
% plot(y7);
% xlabel('Samples');
% ylabel('Gain');
% xlim([0 length(y7)]);
% title('Autocorrelation x10');
disp('autocr done');

ha = axes('Position',[0 0 1 1],'Xlim',[0 1],'Ylim',[0
1],'Box','off','Visible','off','Units','normalized', 'clipping' , 'off');

text(0.5, 1,'\bf Beat Tracking - Beer Money','HorizontalAlignment','center','VerticalAlignment', 'top');

%% fourier plot
fftsig = fft(y5);
% fftsig = fftshift(fftsig);
fs = 1/(length(fftsig));
f =  fs/2*linspace(-1,1,length(fftsig));
```

```
subplot(3,2,6);
plot(f,abs(fftsig));
xlabel('\omega');
ylabel('Gain');
title('FFT');

%% get max values
% sort, get top 10%, avg
% downsample first

y8 = downsample(y7, 256);
y9 = sort(y8, 'descend');
sortlen = floor(.1*length(y9));
y10 = zeros(sortlen);
for K = 1:sortlen
    y10(K) = y9(K);
end

disp('sort done');
maxval = mean(y9)
```