A Guide to Producing An A Cappella CD and Development of a Pitch Detection Program

A Senior Project submitted in partial fulfillment of requirements for the Bachelor of Arts Degree in Liberal Arts and Engineering Studies

By

Jacob Ray Stringfellow

Liberal Arts and Engineering Studies Department

College of Engineering

California Polytechnic State University

San Luis Obispo

Spring Quarter, 2012

Abstract

An in-depth look at the steps required to produce a CD for an a cappella group. From what microphone and preamplifiers to use, to what steps to take during the editing, mixing, and mastering processes. Finished with a look at pitch detection algorithms and how they work, and a little bit of experimentation with my own algorithm and program.

1. Introduction

We live in a world that is filled with music. This music comes in different varieties ranging from hip-hop and rap to country and classical. One genre of music that seems to be on the rise now a day is that of a cappella, or singing without instruments. With shows such as *America's Got Talent* showcasing collegiate a cappella groups, and shows like the *Sing Off* that focus on a cappella singing, it seems that collegiate a cappella groups are reaching a peak in popularity.

With a cappella groups being formed, and music being performed, there is also a rising demand to record their music. This presents an interesting challenge. Getting your a cappella group's song recorded professionally can be costly, and on college student budgets it can be hard to afford. That being said, I suggest that there is a solution to this problem. That solution is recording the CD yourself. You do not have to be a professional to get some quality sounding tracks these days. You just need a good ear, and some time to get everything recorded. Because the task of recording an a cappella group's CD might seem like a little too much to handle, I have created a guide to help push you in the right direction. With everything from what equipment to chose, to how to get the vocal percussion sounding just right, it should help tremendously when attempting to record your groups CD. With a little bit of effort and a read through the guide, you should have a CD recorded in no time.

2. CD

During the production of an a cappella group's CD there are three key steps. The first being recording, followed by editing, and ending with mixing/mastering. The following paper details how I went about the production of a four song CD for Cal Poly's a cappella group Take it SLO.

2.1 Recording

Recording is the first step taken during the production process. Things to think about before you start recording include microphone choice, pre-amplifier and audio interface choice, location of recording, microphone placement, and how to record each member of the a cappella group.

2.1.1 Microphone Choice

Because a cappella deals exclusively with human voices and noises made my human mouths you will want to pick a microphone that does well with detail and clarity. Condenser microphones do well at capturing sounds accurately and in detail so this will be the best choice when recording vocals. Condenser microphones require phantom power to charge them and record sound, so you have to keep that in mind when picking up a pre-amplifier or audio interface. Try to stay away from the low-end cheap condenser microphones and pick one that has plenty of good reviews for doing what you want it to do. I used the AKG C-214 for my project because it is a large diaphragm condenser microphone that got good reviews for recording vocals. I would also suggest microphones such as Blue's Spark or Bluebird, AKG's C-414, and if by some chance you can afford it, Neumann's U87.

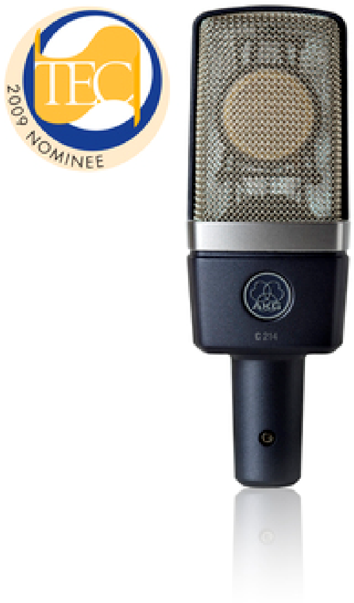


Figure 1. The AKG C-214.
(http://www.akg.com/site/products/powerslave,id,1080,nodeid,2,_language,EN.html)

2.1.2 Pre-Amplifier and Audio Interface Choice

The microphone pre-amplifiers you use are as important, if not more important, than the actual microphone you choose. Again you will want to choose pre-amplifiers that are not in the low-end cheap range, and look for pre-amplifiers and audio interfaces that have numerous good reviews. Most pre-amplifiers and interfaces have phantom power, but because a condenser microphone is being used make sure the pre-amplifier or audio interface has phantom power. I used the Motu 828 mk3 for my project and would suggest brands such as Presonus, Focusrite, and Avid (M-Box). I would advise to stay away from Behringer.

Figure 2. The MOTU 828mk3.
(http://www.motu.com/products/motuaudio/828mk3)

2.1.3 Choosing a Space to Record

When recording vocals for an a cappella CD you want to make sure you have as much isolation as possible. You do not want any background noises, and want as pure of a vocal track as you can get. Due to wanting just the raw sound of the voice, you want to make sure the space you are recording in is not too reverberant. I recorded all the vocalists in a small 5’X5’ box, made from walls covered in foam cones and pyramids that absorbed the sound vibrations of their voices and did not allow for any reflections. Other choices would be recording in a room with padded walls. The key is to make sure the vocals do not bounce off anything and come back to the microphone a second or third time. That can cause phase problems and reverb that is unwanted.

2.1.4 Microphone Placement

You will want to position the microphone directly in front of the singer's mouth. Tell them to stand with their mouth about 3-5" away from the microphone. In some instances a position farther from the microphone might be necessary due to the volume the singer is singing at. To get rid of any unwanted plosives due to gusts of air hitting the microphone diaphragm, you will want to place a windscreen or pop-filter between the singer and the microphone. The windscreen or pop-filter should be placed roughly 2" away from the microphone.

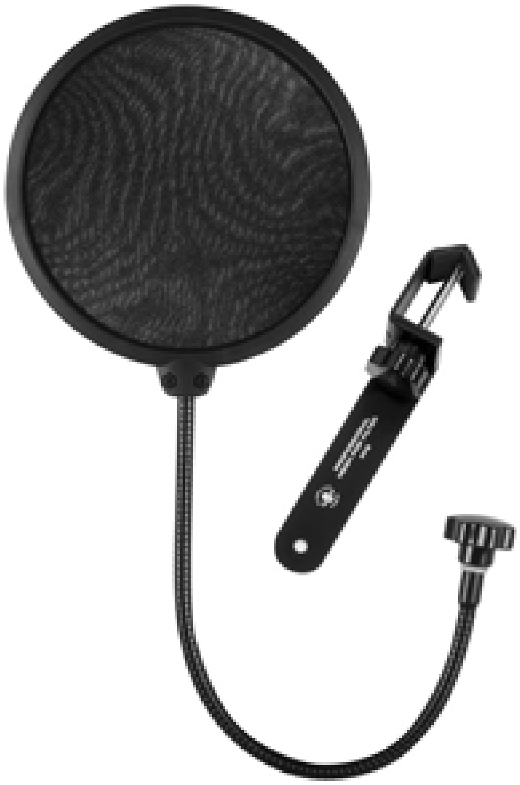


Figure 3. Microphone Pop filter.
(http://www.sterlingaudio.net/products/index.cfm?product=PF2&modId=0)

2.1.5 Recording Each Voice Isolated

When recording a cappella, it is beneficial, and relatively standard to record each voice in the group separately. This gives you the ability in the later stages of editing and mixing, to edit and add effects to each voice individually. This also makes the editing process much easier and provides the opportunity to give each individual his or her own sound.

2.1.6 Double Tracking

During my project, I double tracked each of the background singers. This just means that each singer is recorded twice for each part of the song. This creates a more full sound because it is, in essence, doubling the size of the group.

2.1.7 Recording the Soloist

When you record the soloist, you want to make sure that they are comfortable. You need them to feel like they can sing their heart out to you, and not be judged. You also need to be honest with them and let them know if they hit a wrong note. At the same time you always want to be positive with them though. You do not want them to get down on themselves. You also want them to give you as much emotion in the lyrics as they can. You can fix pitch and timing errors in the editing process, you cannot however fix problems with not emoting the lyrics and feel of a song.

2.1.8 Recording Vocal Percussion (Beat box)

Just like each singer is recorded individually, the vocal percussionist or beat boxer will be recorded separately as well. The majority of a cappella vocal percussion is recorded by a process known as sampling, or in other words recording a set of sounds that are later used to build the drum patterns. To do this the vocal percussionist will record a noise for each of the pieces of a drum kit. For example the kick drum, snare drum, tom toms, hi-hat, and cymbals. Other noises or sound effects might be desired. This is the process I used in my project to record the vocal percussion:

- Have the VP record straight through the song a couple times, until you feel like you have a couple good takes.
- Next have the VP make individual drum sounds you think you might need for the percussion in the song. Examples include: kick drum, snare drum, high hat, toms, and cymbals.
- Once you have a collection of different sounds from the vocal percussionist, have them lay down drum fills that they normally use in the songs you are recording.
- Once you have all the drum sounds and drum fills, sit down with the vocal percussionist and talk about the feel they were going for in each song, so you can make each song sound good, without loosing too much of the vocal percussionist original goal with the vocal percussion.
- Your goal in the end is to take the individual drum tracks (kick, snare, etc.) and build what the vocal percussionist laid down on his runs straight through the song.

2.2 Editing

The second step in the production of an a cappella CD is the editing. During editing, there are two main phases to complete. The first is making sure all voices are in tune, followed by making sure that all of the voices and percussion are aligned properly.

2.2.1 Tuning

The first step is tuning all the background singers (instruments). I use the program Melodyne. I suggest using Melodyne in its standalone form, because using it as a plug-in in your Digital Audio Workspace tends to take up a lot of CPU, and causes frequent crashes. To tune each track you must open each audio file in Melodyne and check to make sure that each note is correct. If a note is incorrect, you simply change it using the drag tool in Melodyne. After you are sure that each note is correct save the audio file as a name that makes sense to you so it is easy to go back in and insert each tuned file. I saved my files as follows: singersname-songtitle#offile.aiff. Once you have all the tuned files back into your DAW, you should listen back to each part individually to check for any errors or dissonance. If all the parts sound okay together, then listen to all parts together. After the background sounds like it is all tuned then you can look at the soloist. I always tend to do as little tuning on the soloist as possible. That is so the song sounds relatively natural. If there are trouble spots, where the soloist sounds overly auto-tuned it might be beneficial to have them come back in to re-record. After the tuning of the soloist is done, it is time to move on to alignment.

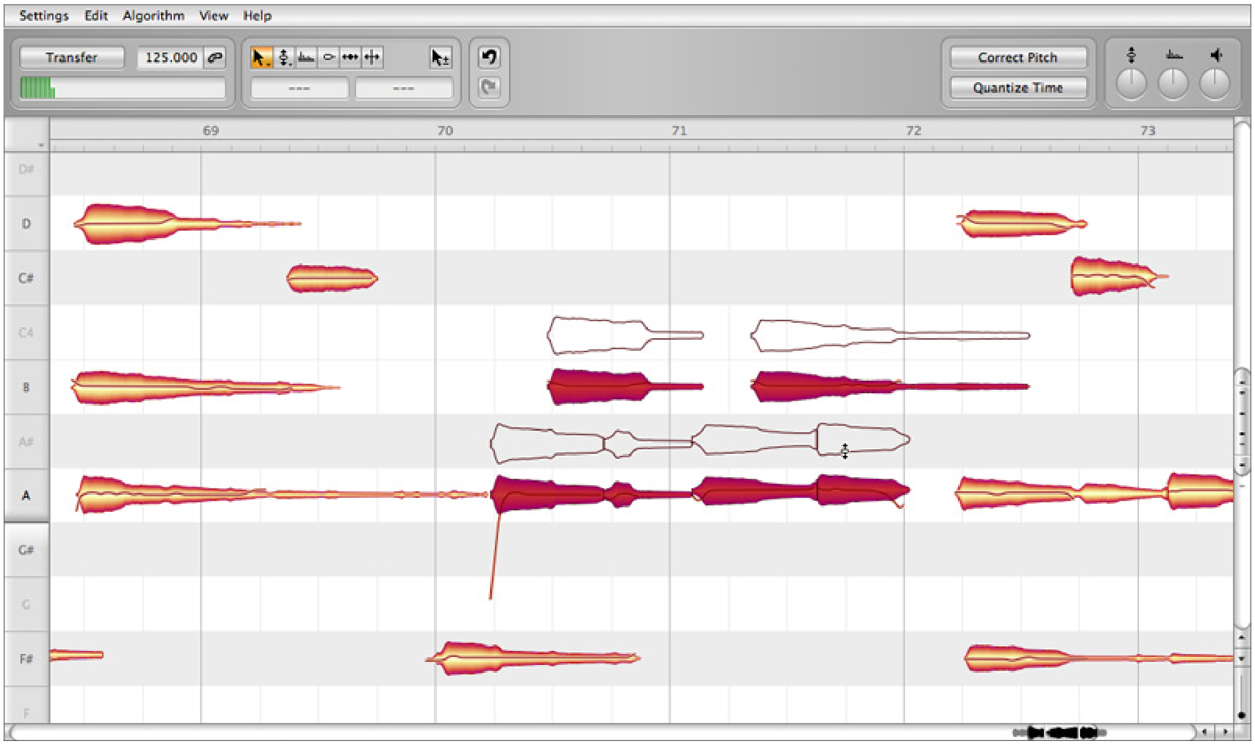


Figure 4. Melodyne Editor tuning software.
(http://www.celemony.com/cms/index.php?id=products_editor&L=0)

2.2.2 Alignment

Alignment is the second step I take when editing. This step makes sure that all rhythms in the song are correct and that all consonants are lined up throughout the group. Sometimes if the rhythms are simple enough, you can use the quantize function in Melodyne to help you line everything up. If that is not the case I use the flex-time function in Logic Pro to help line up all consonants and rhythms. If you do not have access to flex-time there is other software and software packages that help with alignment. The main purpose of this step is to make sure that everyone in the group sounds like they are singing at the same tempo and singing together rather than individually like they recorded. Once you are satisfied with the cohesiveness of the song you can move on to the last step of the editing process, the vocal percussion.

2.2.3 Building Vocal Percussion

The final step I take in the editing process is building the vocal percussion. To do this I usually listen heavily to the rough takes that the vocal percussionist did. I then take the sampled sounds they made and insert them underneath the rough takes. During this process I usually have different tracks for the kick, snare, hi-hat, tom toms, and the overhead (cymbals). Once all the samples have been placed in throughout the whole song, listen back and make sure all of it is in tempo and sounds correct in regards to the rough tracks. Once that is done, the editing process is relatively finished.

2.2.4 Letting Other Ears Help

I highly recommend letting other people, either from the group or in the field of recording, listen to what you have after you are done editing and getting their feedback. They can often hear things that you missed and help you make sure your track is perfect. Once you get their feedback, you can go back and re-evaluate each step of the editing process.

2.3 Mixing

During the mixing stage of the production of an a cappella CD there are a few things to take care of, including making sure all the levels are right, adding cool effects to the vocal tracks, and making the vocal percussion sound more like real percussion. After you finish those things, it is always a good idea to let other people listen to your mixes and give you feedback.

2.3.1 Levels

During the levels portion of mixing, you want to make sure that each vocal part is loud enough, but not too loud. Make sure it is balanced overall and that you can hear each part. Parts that are important include the percussion, the bass, and the soloist. Make sure that each of those is heard clearly. The other parts usually just fill in the rest of the chord, which is also very important, but without the foundation, the top notes don't have any context.

2.3.2 Effects

After all the levels are set, the next step is to start adding effects. Effects include everything from equalization to reverb and even distortion. In this stage you make the voices sound more like instruments. To add these effects you must have some sort of plug-ins. All DAWs (digital audio workspaces) come with a few plug-ins for the basics, such as EQ, reverb, phasing, chorus. If you are not satisfied with the plug-ins that come as part of the DAW you have to purchase other plug-in packages or look for free software online. I suggest Waves plug-ins. The first thing to get done is your equalization, or EQ. During portion of mixing, you take different frequencies and either boost or cut them. Usually you want to make sure that you do not boost too many frequencies, or the song will just explode. The best technique is to apply subtractive EQ, and take out frequencies that you do not want. Certain instruments have natural frequencies that should be boosted or cut, and during the EQ portion you should treat each voice like it is the instrument it is imitating. If you do not know what these frequencies are, you can easily search for them online or get a book on recording arts that will tell you all the fundamental frequencies for different instruments. After all the EQ is done, move on to adding other effects like reverb, delay, distortion, phase shifting, etc. Remember to again treat each voice like the instrument it is imitating and to add other effects as if it were that instrument. Also keep in mind during this portion of mixing what sound you are aiming for overall. If you want your songs to sound more natural and like it is a group of human voices try not to add too many effects. If you want it to sound as close to real instruments as possible that is perfectly fine too.

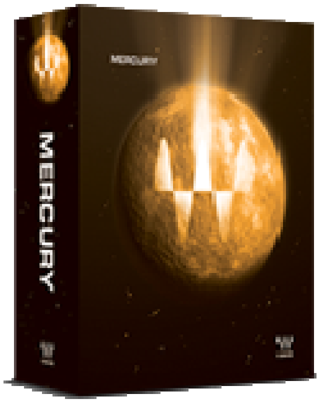


Figure 5. Waves Mercury Plug-in Bundle. Helps with hundreds of different plug-in suites. (http://www.waves.com/Default.aspx?l=4)

2.3.3 Vocal Percussion

Mixing the vocal percussion might be one of the hardest parts of the entire process. Make sure to leave a lot of time to get it sounding exactly how you want it to. To make sure that your vocal percussionist sounds like a real drum set, you will want to have separate tracks for each part of the drum set the percussionist is imitating. The different tracks include

Kick, Snare, Hi-Hat, hi-Tom, mid-Tom, low-Tom, and one or two overhead tracks; one for right and one for left. The kick drum should sound nice and punchy with a nice low end. To get this you will want to enhance the lower frequencies and cut some of the higher ones. One way to get this done is to apply a low-pass filter. Applying the low pass filter around 8kHz will usually get the job done. Then you should boost lower frequencies around 50-80Hz. Though you are cutting a lot of the upper frequencies, make sure that you keep some around the 1-3.5kHz ranges. That gives the kick drum a little definition and keeps it from sounding too boomy. Next is the snare drum. For the snare, it is better to do the opposite and apply a hi-pass filter. Usually cutting around 82Hz and below works well. You can even cut up to 150Hz. Just like the high end on the kick drum giving it some definition, you will want to make sure and keep some of the low frequencies on the snare drum, around 180-250Hz, to make sure that you still get the definition of the initial attack. You'll also want to boost a little around 3.3kHz and 6.8kHz to give the snare the snappy sound an actual snare has. Also adding a little bit of reverb to a snare can help it sound a bit more real, allowing for the sound to ring for a little bit. Moving on to the hi-hat, you will again want to apply a hi-pass filter, to get rid of all the low frequencies. For this low pass filter you can cut around 360Hz and below. Other than that you will want to make sure to boost around 10kHz, and maybe cut a little around 2.5-3kHz. Next piece of the drum set is the toms. The mixing of the toms is highly dependent on what song you are mixing and your personal preference. If you are mixing a rock song you are definitely going to want some beefy toms that can be heard. If you are mixing a slower song, you might not even need any toms. For songs you do need toms, it is good to start of by thinking of them kind of as pitched kick drums. You can apply a low-pass filter, and boost some of the lower frequencies. Next slightly boost some of the frequencies around 800-1000Hz. After that adding some reverb can help to let the tom hit ring out a little, just like the snare drum. After that it is just experimentation with different boosts and cuts in EQ and adding different effects. The last part of the drum kit to mix is the cymbals or overheads. This part is pretty simple, just like the hi-hat. Apply a hi-pass filter at around 360-400Hz. Boost frequencies of about 15-16kHz. You really want your overheads to sizzle, just like a real cymbal would. Sometimes adding a little reverb will help. Make sure after you have mixed each sound of the drum individually, that you listen to them all together and that they sound good as a whole.

2.3.4 Letting Other Ears Help

Just like in the editing section, once you believe that you are done, you probably are not. Let other people take a listen to your mixes and see what they think. Have them give you some advice and use that in a second mix. After your second mix, let them listen again. And make a third mix, and a fourth, and a fifth, until you get everything just right.

2.4 Mastering

Mastering is the last step to producing an a cappella CD, or any CD for that matter. In this final step you give the songs that last coat of sheen and sparkle. You make sure all the volumes are consistent throughout the CD, and that each song has a good balance.

2.4.1 Overall Volume/Levels

For each song, you will want to make your mix as "hot" as possible, or make it as loud as it can be without clipping or distorting. You can use mastering plug-ins to do this. Wave-Arts has some good mastering plug-ins. After you have maximized the volume of each individual track, you will want to listen to all of them and make sure that no individual song is louder, or softer than the other ones.

2.4.2 Final Polish

The final polish includes making sure that the balances of all the songs are correct and also could include an overall EQ of the songs. This last step is a step to make your mixes shine.

3. Pitch Detection Algorithm and Program

A pitch detection algorithm is an algorithm that is used to detect the pitch of a recorded sound.

There are three main kinds of pitch detection algorithms. These include algorithms in the time domain, the frequency domain, and finally a mix of the two. Pitch detection algorithms are used for a multitude of applications, but most the common are in speech recognition programs, video games, or in music. My look at pitch detection algorithms concentrated on use in music and was focused in the frequency domain. I used the numerical computing environment and fourth-generation programming language MATLAB to produce my pitch detection algorithm.

3.1 Algorithm

The first step to coming up with an algorithm for pitch detection is to think about the steps you need to take to detect a pitch. First and foremost, you need an audio sample to detect the pitch of. Next you need to analyze that pitch. And finally you need to identify the pitch.

3.1.1 Recording An Audio Sample

This first step is fairly simple. You can either record your audio sample outside of MATLAB and then import it in, or you can use the record function in MATLAB. To use the record function inside of MATLAB, simply name your audio file anything you like. For example you can name it recObj. Then you set the name you want for your audio sample equal to audiorecorder. It looks like this:

```
recObj = audiorecorder
```

After you have done that you need to set the time to record your audio sample. To do this you use the recordblocking function in MATLAB. For this function you need to tell it what the name of the file is, and how long it will be recording. It will look like this:

```
Recordblocking(recObj, 2)
```

After it has recorded, you can check and make sure that you have your audio file by using the play function in MATLAB. This function simply plays an audio sample back to you. It will look like this:

```
Play(recObj)
```

Your other option is to read in a file that you created outside of MATLAB. The best way to do this is to have a .wav file. To read in your .wav file you simply use the wavread function in MATLAB. It will look something like this:

```
wavObj = wavread(filename)
```

Once you have either recorded your sample, or imported into MATLAB you are ready to move onto the next step, analyzing the audio sample.

3.1.2 Analyzing Audio Sample

The first step in analyzing your audio file is converting the audio file into a format that the computer can understand. To do this, you are going to want to store the audio signal in a numeric array. You can do this by using the getaudiodata function in MATLAB. It will look something like this:

```
myRec = getaudiodata(recObj)
```

After you have run this function you should see a bunch of numbers that the computer now can read as information about the audio file. The next step is to take the data from the audio file, which is currently in the time domain, and convert it to the frequency domain. This is fairly simple because there is a function in MATLAB that does just that. It is the fast Fourier Transform, or fft function. It will look something like this:

```
myFreq = fft(myRec)
```

Now that your audio file has been converted into the frequency domain you need to trim off unnecessary data. To do this, you simply run an absolute value function on the data. It will look like this:

```
realFreq = abs(myFreq)
```

Now you have the data of your audio file in the frequency domain, and it is ready to be analyzed. You can look at the frequencies by simply plotting the array using the plot function. It would look like this:

```
Plot(realFreq)
```

Now that you have all the data ready for analysis, it is time to identify what the pitch is.

3.1.3 Identifying Pitch

Unless you are identifying a sine wave of one frequency, you will have a bunch of frequencies present in your audio file. Also, unless you record in complete silence you are going to have some noise. To get rid of some of your noise, you can apply a noise filter in your program. To figure out what pitch is being played, you have to find the fundamental, or lowest frequency in the file that has a big peak. The peaks of other frequencies after the lowest frequency represent what are called overtones. The overtones present are part of an overtone series, and the amplitude of each of these overtones can give you insight to the timbre or color of the pitch you are identifying. To find this fundamental frequency, look for the first peak on the data that you plotted. In MATLAB you can zoom in until you find out what frequency the first big peak is at. You can also use the function findpeaks to find all peaks, and the frequency they occur at. It would look like this:

```
[pks, locs] = findpeaks(realFreq)
```

This will return an array pks and an array locs. The array pks will consist of the max amplitudes of the peaks, while the array locs will consist of the frequencies at which these peaks occurred. Because there might be quite a number of peaks in your audio file that you are not interested in, you might want to ask your program to only find peaks over certain amplitudes. To do this, you apply a minimum peak height to your findpeaks function. It will look like this:

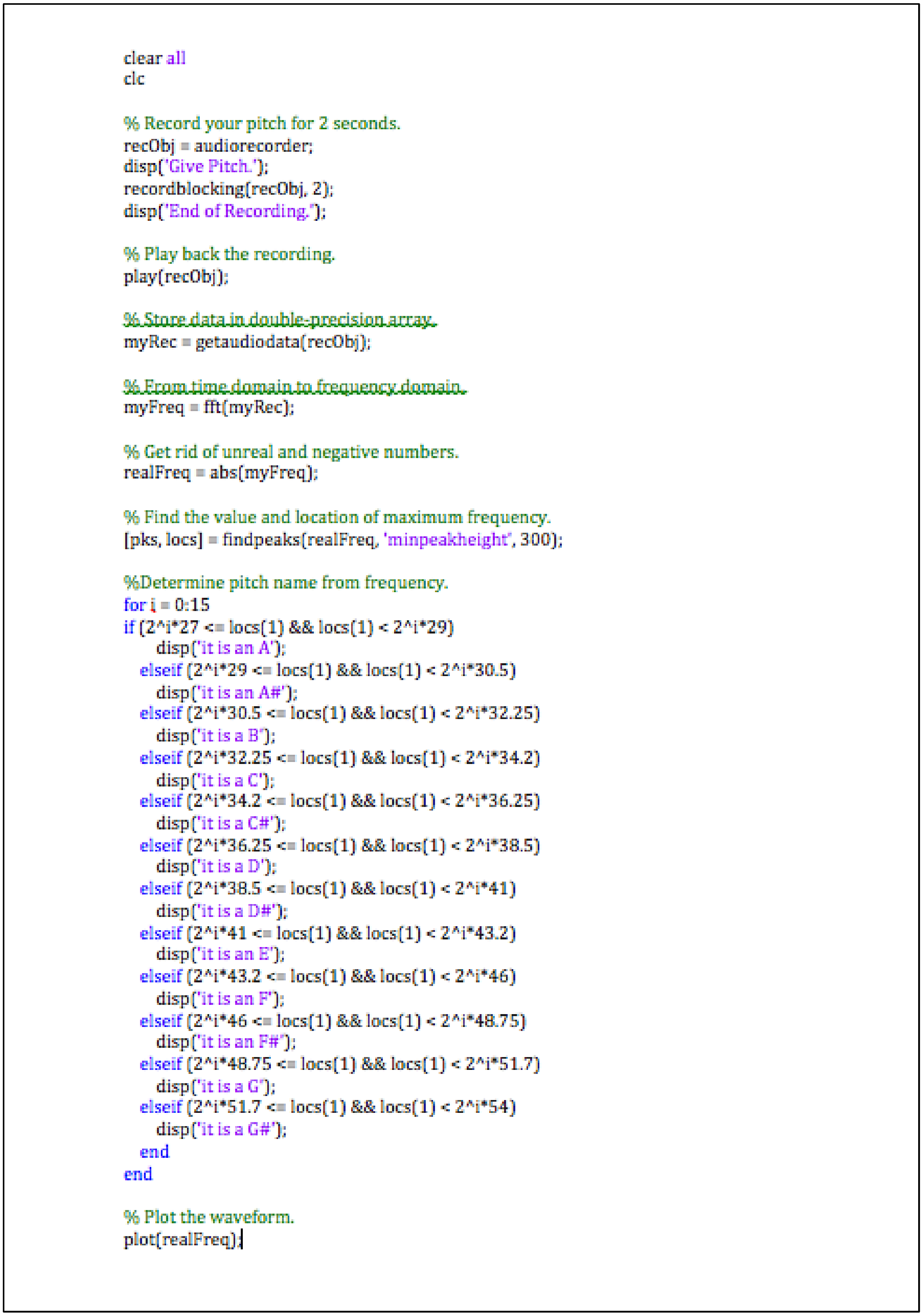
```
[pks, locs] = findpeaks(realFreq, ‘minpeakheight’, 300)
```

Now your pks array will only consist of peaks that are over 300, which significantly cuts down the number of peaks returned. Now that you have the important peaks it is time to assign the first peak as the fundamental. The first peak amplitude will be located at pks(1) and the first peak frequency will be located at locs(1). For the purposes of this program, only the first peak frequency is important. So now that you know your fundamental frequency is equal to locs(1), you have to compare that frequency to frequencies on an existing frequency chart that tells you what pitch corresponds with the given frequency. Due to the fact that when a frequency is doubled, it is the same pitch name an octave higher, it makes the coding for determining pitch a little simpler. Instead of having to go through each note one at a time from the bottom of human hearing around 20Hz, and going all the way up to the upper ranges of human hearing at 16,000Hz, you can just apply code that basically states, if the fundamental is x frequency, or any number of doublings of that frequency, it belongs to a certain pitch name. This coding will look something like this:

```
for i = 0:15
        if (2^i*27<= locs(1) && locs(1) < 2^i*29)
```

```
                                        disp (‘it is an A’)
```

This piece of code basically says, for the variable i being equal to 0-15, if the fundamental frequency is between 27, or any doubling of 27, and 29, or any doubling of 29, the pitch assigned to that frequency is an A. You will write code like this for every semitone from A all the way to G#. Once your fundamental frequency is matched with the correct pitch equivalent, the program should display the pitch name, i.e. A or F#.



```
clear all
clc

% Record your pitch for 2 seconds.
recObj = audiorecorder;
disp('Give Pitch.');
recordblocking(recObj, 2);
disp('End of Recording.');

% Play back the recording.
play(recObj);

% Store data in double-precision array.
myRec = getaudiodata(recObj);

% From time domain to frequency domain.
myFreq = fft(myRec);

% Get rid of unreal and negative numbers.
realFreq = abs(myFreq);

% Find the value and location of maximum frequency.
[pks, locs] = findpeaks(realFreq, 'minpeakheight', 300);

%Determine pitch name from frequency.
for i = 0:15
if (2^i*27 <= locs(1) && locs(1) < 2^i*29)
      disp('it is an A');
   elseif (2^i*29 <= locs(1) && locs(1) < 2^i*30.5)
      disp('it is an A#');
   elseif (2^i*30.5 <= locs(1) && locs(1) < 2^i*32.25)
      disp('it is a B');
   elseif (2^i*32.25 <= locs(1) && locs(1) < 2^i*34.2)
      disp('it is a C');
   elseif (2^i*34.2 <= locs(1) && locs(1) < 2^i*36.25)
      disp('it is a C#');
   elseif (2^i*36.25 <= locs(1) && locs(1) < 2^i*38.5)
      disp('it is a D');
   elseif (2^i*38.5 <= locs(1) && locs(1) < 2^i*41)
      disp('it is a D#');
   elseif (2^i*41 <= locs(1) && locs(1) < 2^i*43.2)
      disp('it is an E');
   elseif (2^i*43.2 <= locs(1) && locs(1) < 2^i*46)
      disp('it is an F');
   elseif (2^i*46 <= locs(1) && locs(1) < 2^i*48.75)
      disp('it is an F#');
   elseif (2^i*48.75 <= locs(1) && locs(1) < 2^i*51.7)
      disp('it is a G');
   elseif (2^i*51.7 <= locs(1) && locs(1) < 2^i*54)
      disp('it is a G#');
   end
end

% Plot the waveform.
plot(realFreq);
```

Figure 6. The MATLAB code for my pitch detection program. Refer back to earlier sections for explanation of code.

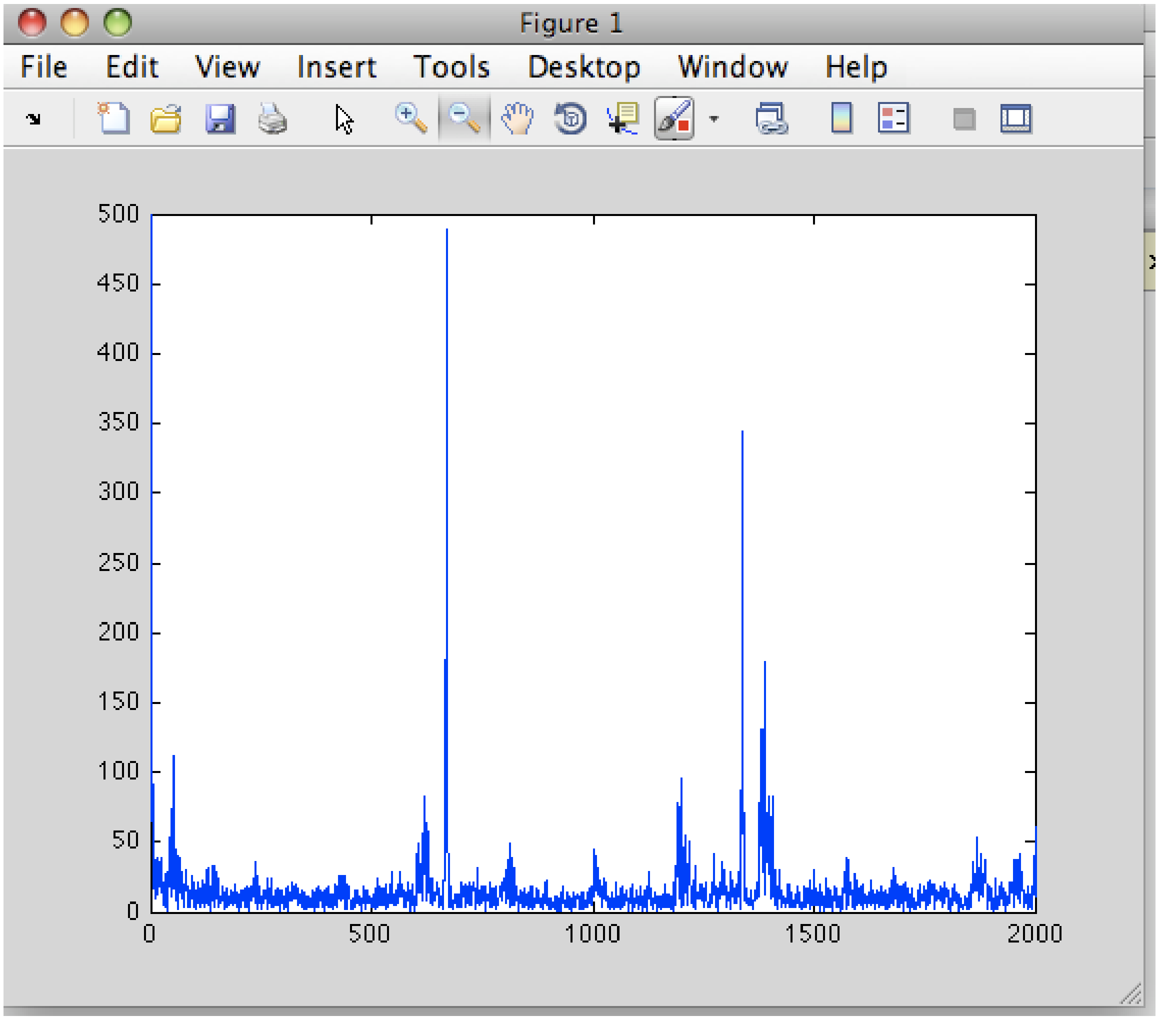


Figure 7. An example of the graph output in MATLAB at the end of the pitch detection program.

4. Future Work

Due to time constraints, I was not able to finish all that I was hoping to on this project. There is still some editing to be done on the four songs, and also a bit of mixing. The pitch detection program is almost finished, but I need to figure a way to run the program outside of MATLAB.

4.1 CD

For the CD, I still have a bit of work to do on the vocal percussion, the soloists, and getting the alignment of the background parts exactly right. The biggest part of the vocal percussion that needs to be finished is the toms and cymbals or overheads. The soloists need to be tuned, and in some spots be lined up better with the background parts. As for aligning the background, there are some cut offs that still are not together. I would like to submit my work to be evaluated, and possible put on a compilation album of the best collegiate a cappella music around the world.

4.2 Pitch Detection

With a bit more time to work on the pitch detection programming, I hope to discover a way to get the program to run outside of MATLAB. Maybe make a phone application out of it. I also want to develop some kind of noise filter, so that when the recorded audio file plays back, it sounds cleaner. I also might look into taking the pitch detection program, and adding a correction function.

5. Conclusion

In this paper, I have explained step-by-step instructions on how to record, mix, and produce an a cappella CD. I have also briefly explained the methods used to create a pitch detection algorithm and program. Major concepts to take away from this paper include the following:

- When recording an a cappella CD, or any CD for that matter, there are three important steps: Recording, Editing/Mixing, and Mastering.
- In the first step, recording, make sure you do not rush into anything without thoroughly thinking about the end goal of the recording. Many choices along the way play a huge role in the final product.
- In the second step, editing/mixing, make sure to have a critical ear. Try to be as precise as you can, and make sure to get other people’s opinions on your work. Also, remember to take breaks from your work and come back with fresh ears at a later time.
- In the final step, mastering, again have a critical ear. This is the part of the process that really makes your production shine. Again, use other people and their opinions whenever possible.
- When trying to identify a pitch from something that is not a perfect sine wave of one frequency, remember you are looking for the fundamental, or lowest frequency. That frequency will correspond to the pitch name.
- When a frequency is doubled, it is the same pitch name, just an octave higher.
- When recording sound, there will always be some noise, so only pay attention to the important frequency peaks.

- Finally, if at first you do not succeed, try again. Nothing is going to be perfect the first time you try it, or the hundredth time you try it for that matter. You will get better experience.

6. Further reading

If you are interested in learning more about either producing CD's or pitch detection algorithms or programs, the following sources might be worth looking into.

6.1 CD Recording

If you are interested in recording, you might benefit from looking at these books and articles. They are more general to recording anything, not just a cappella but still can be beneficial.

Anker, Kevin, and Orren Merton. *Logic Pro 9 power! the comprehensive guide*. Boston, MA: Course Technology/Cengage Learning, 2011. Print.

Bartlett, Bruce, and Jenny Bartlett. *Practical recording techniques: the step-by-step approach to professional audio recording*. 5th ed. Burlington, MA: Focal Press, 2009. Print.

Kefauver, Alan P.. *The audio recording handbook*. Middleton, Wis.: A-R Editions, 2001. Print.

Stanyek, J. , & Piekut, B. (2010). Deadness: Technologies of the intermundane. *Volume 54, Number 1, Spring 2010 (T 205)*, *54*(1), 14-38.

6.2 Pitch Detection Algorithms

For more information on pitch detection algorithms you can take a look at these articles.

Cai, R., Shi, S., & Zhu, Y. (2010). A modified pitch detection method based on wavelet transform. , *2*, 246-249.

Grell, A. , Grell, A. , Sundberg, J. , Ternstrom, S. , Ptok, M. , et al. (2009). Rapid pitch correction in choir singers. *The Journal of the Acoustical Society of America*, *126*(1), 407-413.

Hasan, M. , & Kabir, M. (2008). Minimization of error in pitch detection algorithm using discrete fractional cosine transform. , 403-406.

Hsu, C. , Wang, D. , & Jang, J. (2011). A trend estimation algorithm for singing pitch detection in musical recordings. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *ISSU*(PAGE), 393-396.

Lane, J. (1990). Pitch detection using a tunable iir filter. *Computer Music Journal*, *14*(3), 46-59.

Prukkanon, N. , Chamnongthai, K. , & Miyanaga, Y. (2011). Vt-amdf pitch detection algorithm and thai tone recognition system. , 958-961.

Rabiner, L. , Cheng, M. , McGonegal, C. , & Rosenberg, A. (1976). Comparative performance study of several pitch detection algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *ASSP-24*(5), 399-418.

Shahnaz, C. , Zhu, W. , & Ahmad, M. (2007). A robust pitch detection algorithm for speech signals in a practical noisy environment. *Midwest Symposium on Circuits and Systems*, *ISSU*(PAGE), 385-388.

References

PreSonus . (n.d.). *PreSonus* . Retrieved June 7, 2012, from http://www.presonus.com/

Apple - Logic Pro 9 - Everything you need to make great music.. (n.d.). *Apple*. Retrieved June 7, 2012, from http://www.apple.com/logicpro/

Apple - Logic Pro 9 - How To. (n.d.). *Apple*. Retrieved June 7, 2012, from http://www.apple.com/logicpro/how-to/

Avid | Mbox Pro. (n.d.). *Avid | Home*. Retrieved June 7, 2012, from http://www.avid.com/US/products/Mbox-Pro

Blue Microphones. (n.d.). *Blue Microphones*. Retrieved June 7, 2012, from http://www.bluemic.com/

C 214. (n.d.). *AKG START*. Retrieved June 7, 2012, from http://www.akg.com/site/products/powerslave,id,1080,nodeid,2,_language,EN.html

C 414 XLS. (n.d.). *AKG START*. Retrieved June 7, 2012, from http://www.akg.com/site/products/powerslave,id,1128,pid,1128,nodeid,2,_language,EN.html

celemony_ :: News. (n.d.). *celemony_ :: News*. Retrieved June 7, 2012, from http://www.celemony.com/cms/

MATLAB - Documentation. (n.d.). *MathWorks - MATLAB and Simulink for Technical Computing*. Retrieved April 5, 2012, from http://www.mathworks.com/help/techdoc/

MOTU.com - Overview . (n.d.). *MOTU.com - Welcome to MOTU Audio*. Retrieved June 7, 2012, from http://www.motu.com/products/motuaudio/828mk3

NeumannUSA - Homepage. (n.d.). *NeumannUSA - Homepage*. Retrieved June 7, 2012, from http://www.neumannusa.com/

Saffire PRO 40 Audio Interfaces Professional 20 In / 20 Out Firewire interface with eight Focusrite Pre-amps. (n.d.). *Audio Interfaces, EQ & Compression, Mic Pres - Focusrite*. Retrieved June 7, 2012, from http://www.focusrite.com/products/audio_interfaces/saffire_pro_40/

:: Sterling Audio ::. (n.d.). *:: Sterling Audio ::*. Retrieved June 11, 2012, from http://www.sterlingaudio.net/products/index.cfm?product=PF2&modId=0

The Ultimate EQ Guide. (n.d.). *Electronic Music & Music Production Forum - IDM Forums*. Retrieved June 7, 2012, from http://www.idmforums.com/showthread.php?t=18237

Waves Audio. Audio Plugins for Mixing, Mastering and Audio Recording. (n.d). From http://www.waves.com

Take it SLO – Official Website. <www.takeitslo.com>