Project Lightgreen:
Digital Needle

Final Report

2E1366 - Project course in Signal Processing and Digital Communication

Ulf Kalla Niklas Jaldén Niklas Lithammer
Markus Eriksson Eduardo Pérez

June 2, 2003

Abstract

The purpose of the project was to implement a solution for extracting sound information from a scanned image of a LP record onto a Digital Signal Processing board (DSP-board). The report gives a short description about how records are made and how records can be played without an ordinary record player. The program developed in the project is called Scan'n'Play.

Keywords: digital needle, Old 78 records

Preface

This is the final report for the project group Lightgreen in the Project course in Signal Processing and Digital Communication, 2E1366, of S3 at the Royal Institute of Technology, Stockholm, spring 2003. The project group would like to thank our project assistant, Tomas Andersson, for all help during the project. We would also like to thank the project group Lightblue for not putting up a fight in pool.

Contents

List of Figures

1 Introduction

1.1 Background

Today the traditional record player is out of date, but still you can buy ordinary records (LPs) and play them on a record player. And still some DJ:s prefer the sound from LP:s. The predecessor to today's LP player, the Old gramophone, were able to play back the Old 78 records, with a speed of about 78 rpm (see section 2.1 for more details regarding the speed). Record players of today often lack the possibility of playing records at a speed of 78 rpm. This capability is not really necessary, since there are no new records today that require that speed.

If you have some Old 78 records but your player is out of order, you can still be able to play your favourite records. You can simply buy a laser record player. This player has a laser beam that follows the groove on the record and outputs music. One of the advantages of this player, is that the player reads the groove optically and therefore the record is not damaged by a needle. Unfortunately these players are rather expensive and probably affordable only by the most devoted music lovers.

1.2 Objective

The task was to implement a fully automatic system for extracting sound information from one or several scanned images of an Old 78 record, on a PC with a DSP board. The final product should not use Matlab in any way and all sound enhancement and signal processing should be done by the DSP board. The output from the final product should be of better or of equal quality compared to [1].

1.3 The project

1.3.1 Overview

The Project Lightgreen Digital Needle is a successful project. All course deadlines were held and both the goals for the project and the objectives for the project were fullfilled. However, the project planning could have been better and the project deadlines were moved several times. The time estimated for the project was not enough, which resulted in overtime work the group. The total time for the project was ∼1 500 hours. (See also enclosed and updated Time schedule for deadlines and delays).

1.3.2 Personnel and equipment

The Project Lightgreen group consists of the following people:

Name	**Areas**
Ulf Kalla	Project manager, Project Plan, Poster Presentation, Final report
Niklas Jaldén	Algorithm 1, Matlab, C code
Niklas Lithammer	DSP, GUI, C code
Markus Eriksson	Algorithm 2, C code
Eduardo Peréz	Scanning, Webmaster, GUI, Presentation
Tomas Andersson	Project assistant

The list below shows the most important equipment used in the project.

- Three PC:s with different hardware configurations
- One Texas Instrument C6701 EVM DSP board
- One USB scanner CanScan 5000, Canon, with software
- Loudspeakers
- Old 78 records and LP:s
- Code Composer Studio 2.0
- Photoshop 6.0
- Visual Studio C++ 6.0
- Open CV [6]

1.4 Project Lightgreen on the web

Since the start of the project there has been a web page where interested people have been able to follow the progress of the project. The web page offers the Project plan, progress reports, sound files from the latest test and more to download.

http://www.s3.kth.se/signal/edu/projekt/students/03/lightgreen/

Both versions, the DSP version and the sound card version, are downloadable from the web page of the project.

2 Theory

2.1 Making a record

The system for making a record can easily be described as an amplifier connected to a cutting-head. The stylus of the cutting-head is connected to a coil, which lies in a strong magnetic field [2], [3]. When the sound signal is applied to the coil, the stylus moves and engraves the sound wave into the blank disc. The cut is adjusted to give an avarage depth of 60 μm. The sound wave can easily be seen on a record when looking close enough, Figure 1. The groove is orientated in a helix with a counter clock-wise direction. On a mono record the sound information is "carried" by lateral movements of the groove. On stereo records is the information carried by lateral and vertical movements of the groove. The Digital Needle Scan'n'Play uses only the lateral movements of the groove on the record, hence Scan'n'Play is a mono player.

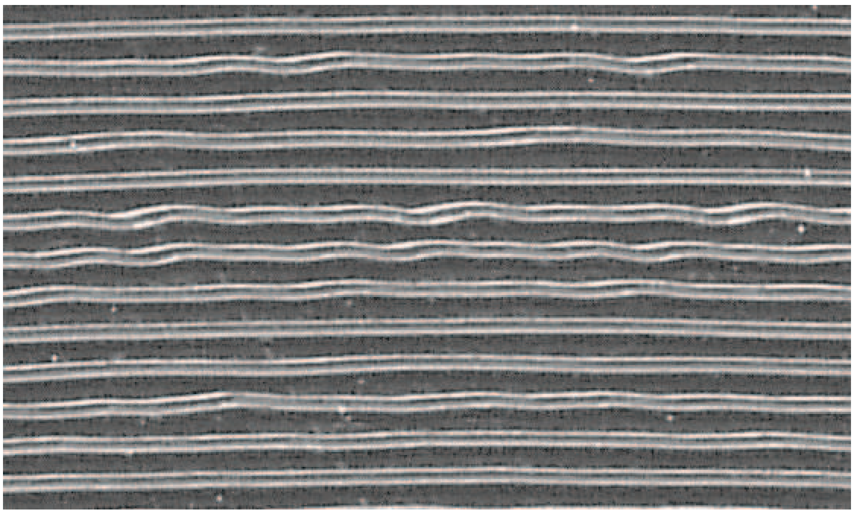


Figure 1: Tracks of an Old 78 record

The signal being cut into the master disc can not have a flat frequency response. If it had a flat frequency response the resulting cut would be "constant velocity". This is due to the fact that the cutter is a velocity type transducer. The effect of this is that the cut will have a large amplitude for bass frequencies and a low amplitude for high frequencies. These two effects restrict the playing time and cause signal to noise problems. Therefore, their signal is processed through a equalization network before it is sent to the cutter. When the disc is played, the amplifier runs the signal through a reproduction filter, which naturally is an inverted recording filter. One standard for this type of filter is the RIAA curve (see Figure 2).

Regarding the speed of the Old 78 records there is no actual standard, even though it should be 78 rpm. The play back speed of the records can differ between 68 rpm up to 84 rpm [4].

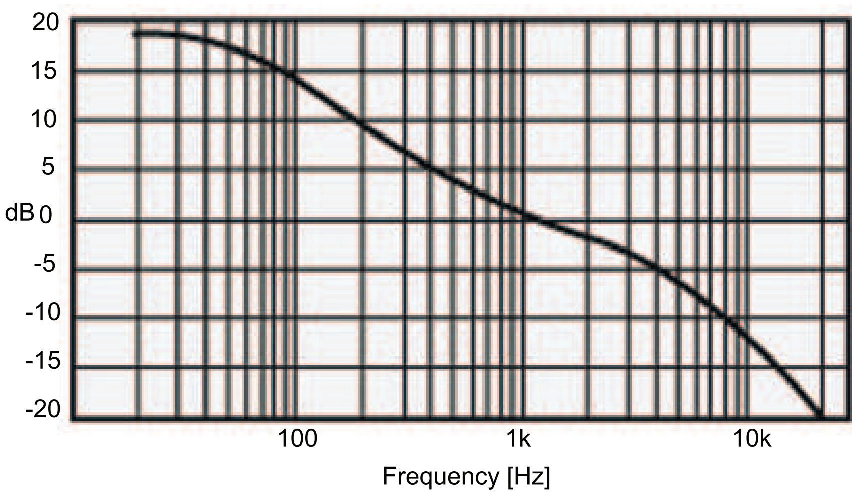


Figure 2: A reproduction filter, RIAA type [2]

2.2 Short facts of records

- Play back speeds $33\frac{1}{3}$rpm, 45 rpm and ~78 rpm
- Recorded in mono or stereo
- The track is a helix running counter clockwise
- The Old 78 records and the 45 rpm singles have just one track on each side

3 Algorithm and solution

In the beginning of the project there were two different algorithms discussed, the Polar algorithm and the the Straighten-out algorithm. In an early stage the Polar algorithm was chosen as the main algorithm and the Straighten-out algorithm became the backup algorithm. At the time of the decision, the Polar algorithm was the most developed one and it was also chosen because it requires less and easier calculation steps. The Polar algorithm makes almost the same operations as the Straighten-out algorithm but in one step.

3.1 Limitations, restrictions

Due to light reflections and scanner resolution, only the top 90 degrees can be used. Therefore the record must be scanned in four parts and in each scan, half the record must be scanned. There are only two types of records that the system is able to "play" due to limitaions in the scanner. The two types of records that the system can use are Old 78 records, with a play back speed of ∼78 rpm and 45 rpm records, with a play back speed of 45 rpm. See also section 4.4.1, "How to scan". In the project a resolution of 2 400 dpi has been used for the records.

3.1.1 Exposure settings for 45 rpm records

When scanning an Old 78 record the default settings for the scanner have been used. For some 45 rpm records the exposure settings need to be modified. This has to be done in order to "see" the tracks since they are ingraved much tighter than on an Old 78 record. See Figure 3 for correct exposure settings. Also see section 4.4.1 for how to scan. Some Old 78 records may alos need this correction. However, this gives a noiser result.

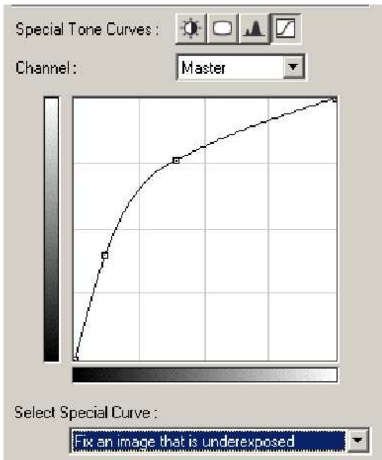
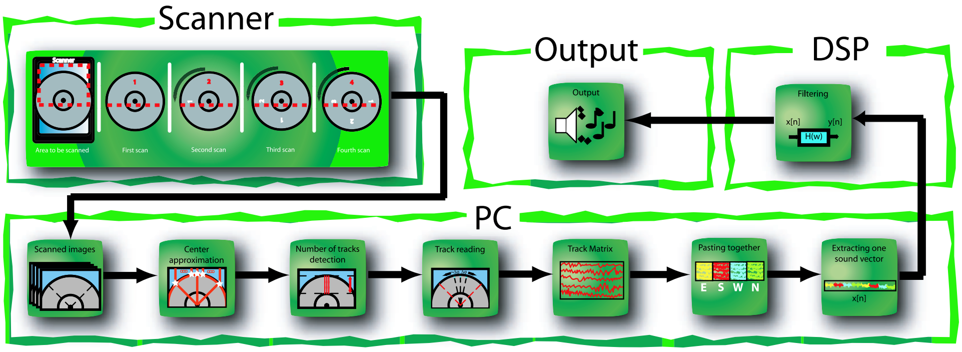


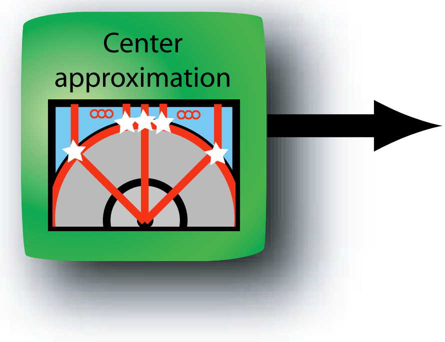
Figure 3: Exposure curves. Exposure settings for scanning a 45 rpm record

3.2 Polar algorithm

1. The center of the record is detected.
2. The numbers of tracks are detected along a vertical line, which goes from the center of the record to the edge of the record. The length of the vector is equal to the number of tracks.
3. The highest radius value in the vector received from the second step forms a start value for the farthest track.
4. Two new searches are done with $\pm\Delta\varphi$, where $\Delta\varphi$ is an angle from the vertical line mentioned in step 2. Along these two new lines, the startvalue $\pm$P ($1 \leq P \leq 3$) pixels are considered and the one on each line with highest intensity is saved. P can be set in the settings window, Figure 9. The pixel is saved with its radius value. The next start value is an avarage of the N latest picked radii values.
5. The algorithm is then repeated from step four until the algorithm has reached $\pm 45^o$. All values are collected in a row in a matrix. The lower the number of the row, the farther out the track.
6. The next value in the vector from step two is picked and the algorithm is repeated from step four. Building up the matrix with each track corresponding to a row in the matrix.



3.2.1 Center detection



The edge of the record is detected. This is done by stepping down from the top of the image and looking for a specified threshold value, i.e. where the intensity is changed dramatically. First the step is about 100 pixels, then when the intensity is changed, the algorithm returns to the point before the change and the step is decreased to 10 pixels. When a change is discovered once more, the algorithm returns again to the point before the change and the step is decreased to one pixel. When the change in intensity is discovered, the x and y coordinates for this pixle are saved in a vector. This proceedur is then repeated for up to 6 000 lines. Each line gives the x and y coordinates for the records edge. Due to reflections and shadows an area of $\pm$N points from the vertical line are not used. N is $\sim$1 500 pixles for a 45 rpm single and $\sim$4 000 pixels for an Old 78 record, Figure 4.

A circle is approximated to the achieved edge points of the record. The center of the approximated circle is supposed to coincide with the center of the record. However, due to reflections and other errors this is not the case. Therefore each received point is compared to the radius of the approximated

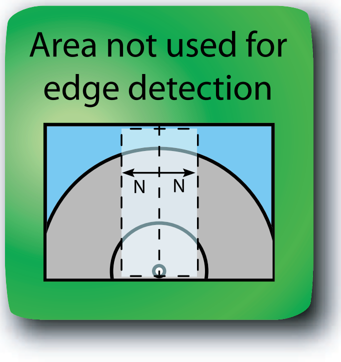
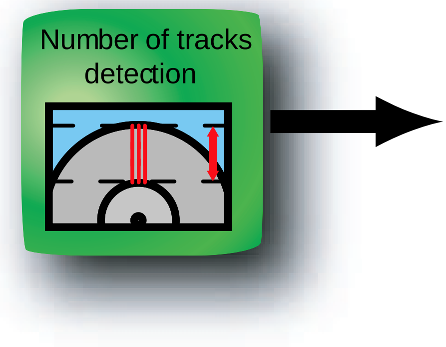


Figure 4: Area not used for edge detection

circle and the circle itself. The 30% of the points that gives the biggest difference between the radius and the approximated circle are deleted. A new circle is approximated to the points that are left. The new center of the circle gives a more accurate approximation of the center of the record.

3.2.2 Number of tracks detection



A line is vertically "drawn" from the center of the record to the edge of the record. Along this line the number of tracks will be detected. The area for this is between the label of the record and the edge of the record. A FFT is calculated along the line to receive the distance between two adjecent tracks, and the width of one track.

This line results in an intesity image, Figure 5. A threshold value (the horisontal line i Figure 5) is calculated as the average value of the vector times a constant $\frac{1}{k}$, $0.5 \leq k \leq 2$. However, some of the tracks, may not actually be tracks. These can be hair or dust on the record. The algorithm checks if the distance between tracks is to big, if so, the algorithm "looks" for the brightest value, the highest peak in Figure 5 between the two latest detected tracks and sets this as a track. If the distance is to small it removes the wrong detected "track".

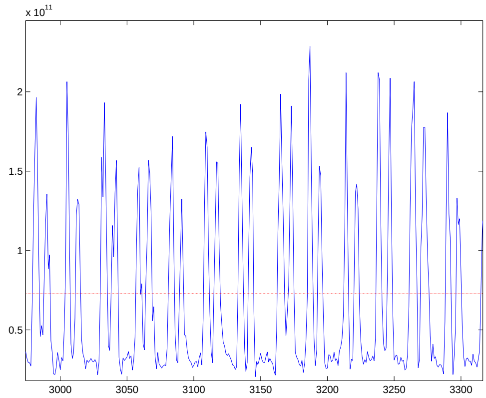


Figure 5: The number of tracks are detected

3.2.3 Track reading

The highest radius value, the first element, in the vector received from the "Number of tracks detection", is used as the start coordinates for the outmost track. Two new searches are done with $\pm\Delta\varphi$, where $\Delta\varphi$ is an angle from the vertical line mentioned in the "Number of tracks detection". Along these two new lines, the start value $\pm$P ($1 \leq P \leq 3$) pixels are considered and the one on each line with highest intensity is saved with its radius value. The following start coordinate is an aveage of the $\sim$14 saved coordinates.

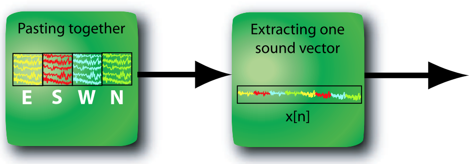
When one track has been read up to $\pm\Delta\varphi = 45^o$ the procedure is repeated for the next track and so on, until all tracks are extracted. Each track is saved as a row in a matrix.

3.2.4 Correlating the matricies

- image A = "north" image
- image B = "west" image
- image C = "south" image
- image D = "east" image

For the first image, image A, being scanned, a pixel vector is extracted along a line $\varphi = -45^o$ from the vertical line. This vector is saved. When the next image, image B, is scanned and processed, a pixel vector is extracted along a line $\varphi = +45^o$ from the vertical line. This vector is also saved. Then these two pixel vectors are correlated in an attempt to find how the two matricies should be pasted together to get corresponding tracks from each image. The result of the correlation is a constant which tells how much image B should be moved in a vertical direction in comparison to image A. This constant is added to radius values of image B. Image A is the norm, so when image C is compared to image B, the constant from this comparison is added to the AB constant. This gives that the tracks of C will be correct "connected" with corresponding tracks in both A and B. Same procedure is also done for image D.

3.2.5 Extracting one vector from the matricies



Now the matricies can be "pasted" together with the correction from the previous correlation step. In the matricies, the number of rows are equal to the number of tracks, if the algorithm has worked as expected. The first row in matrix A will be the beginning of the music. The end of this row will be matched with all rows in matrix B and the best match, if it is not larger than the average lap distance, will be appended to row 1 in matrix A. Then the end of the chosen row of matrix B, is matched with all rows in matrix C. The procedure is repeated for matrix C to D. Now the first track of music is extracted. The procedure is repeated for the rest of the tracks until all music is extracted..

3.3 Straighten-out algorithm

1. The center of the record is detected
2. Converting the first "row" closest to the label radius to a polar coordinate image. This is done with angles$\pm 45^o$ from the vertical line from the center of the record.
3. Increase the radius and repeat step three. Each "circle line" will be a straight line in the new image.
4. When the whole record, from the label to the edge of the record, has been converted into a polar image, the tracks are detected and followed.

3.4 Straighten-out algorithm

3.4.1 Center detection

The Straighten-out alogrithm uses the same Center detection function as the Polar algorithm. See section 3.2.1 Center detection.

3.4.2 Converting to polar image

After the center has been detected, the conversion from polar to cartesian coordinates begins. Starting from the radius where the label has its edge, all pixels on that radius in the angle interval, -45 degrees to$+45$ degrees from the vertical line, are copied to a new image and placed on the first row. Then the radius is increased and all the pixels with this new radius are copied to the second row in this new picture. This will be repeated with all radiuses until the whole upper quarter of the record is transformed to polar coordinates. This polar image will have a number of rows that is calculated as ("radius of record" - "radius of label") x "resolution of scanner", and the number of columns will depend on the desired sample rate. Typically this will be around 10 000 rows and 5 000 - 8 000 columns. This whole operation is repeated with all four images of the record. This polar conversion will "bend" the tracks, so that each track will go from one side to the other in one image and then continue in the next image. At the end of the fourth image the track will continue in the first image slightly below where it started.

3.4.3 Track reading

Following the track is fairly straightforward. Assume that the track will vary around a specific average radius. This assumption is not entirely correct since the track spirals in towards the center, but it is good enough. Next, assume that the track will be the brightest pixel in the image. The final assumption is that the track will only vary very little around the average radius. This is true because the needle of a real record player can only follow the track if it varies very little. Now the first thing to do is to find out where the track starts. The easiest approach is to just start looking somewhere slightly above where the track usually starts and just search downwards until a pixel is found that is bright enough to be part of the track. If this method finds a pixel that is not part of any track, that is ok. The only effect is that it will take a while before the music produced is actually part of the track. But since this time is significantly less than one second, this is not a major flaw.

With the previous stated assumptions and a pixel that is part of the track is found, the best thing to do to follow the track, is just to go to the next column in the polar image and look for the brightest pixel at a radius near the radius where the track was in the last column.

If the center had been found perfectly, it would now be easy to just follow the track trough the different images. However, since the center might not be perfectly found, it is possible that the track will be placed at different radiuses in the different images of the record. To compensate for this a correlation between the edges is calculated to move the images up or down until the tracks are well aligned.

Another problem is that sometimes the track is distorted, so that it is not possible to find a good pixel to represent the track. This might cause the "needle" to go off from the track. But if this happens for too long, about five bad columns, a special find-track function is run that looks up and down until the closest pixel that is a good pixel, is found, and continues to follow the track from there. A problem that is closely related to this is that sometime when the tracks lie close together, it skips to the next or the previous track. To compensate for this, the found radius values are continuously checked with the previous values to make sure they do not deviate too much from the earlier ones. If the difference is too large, it is assumed that the "needle" has skipped and it is moved back to where it is expected that the track really is at this time.

In this manner the track is followed until the whole song is found. What is found is actually the radius of the track, and what is desired is the variations in the radius. To calculate these variations the assumption that the track will vary around the average radius is used. Just calculate the local average, ± 100 pixels is enough, and subtract this from the calculated radius. This will very closely approximate the variations in the track.

4 System

The C code used for loading images has been downloaded from [6]. Most of the knowledge for the DSP is from [5].

4.1 DSP

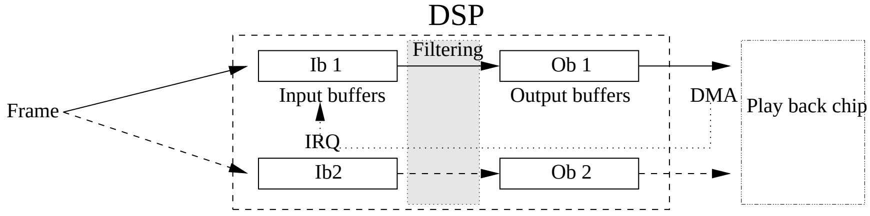
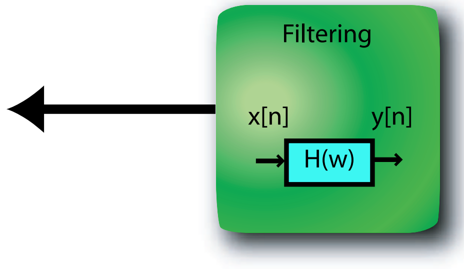


Figure 6: Frame processing in the DSP

The output from Scan'n'play is a sound vector with unprocessed samples. This vector is split into frames, where each frame is 8 000 samples long. One frame is sent from the PC to the input buffer 1 of the DSP board (see Figure 4.2 for all parts). When the input buffer is full the buffer is filtered through the FIR filter and stored in corresponding output buffer. When the filtering is done and the output buffer is full, the content is sent via DMA to the play back chip. While output buffer 1 is played, input buffer 2 receives a new frame which is filtered and stored in output buffer 2. When output buffer 1 is empty, an IRQ is sent and swithes output buffer from 1 to 2.

4.2 Filtering and output



For play back of the soundvector see section 4.4.1 "Play music". If the check box FIR-filtering is marked, the sound vector is processed through a FIR-filter with 64 taps. The charateristics of the filter can be seen in Figure 7. This filter is basically built on a RIAA reproduction filter, Figure 2. There are no mathematical calculations for this filter. The filter has been produced by testing, which gave the best sound. This filter reduces the effect of the noise called "the train" (see section 5.1). There are two different filters. One for the Old 78 record option and one for the 45 rpm record option. The filters differ because the two options use different sampling rates. The Old 78 record option uses 44.2 kHz (but is played with 44.1 kHz) and the 45 rpm record option uses 32 kHz.

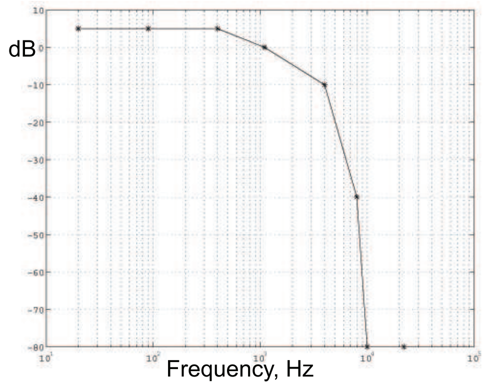
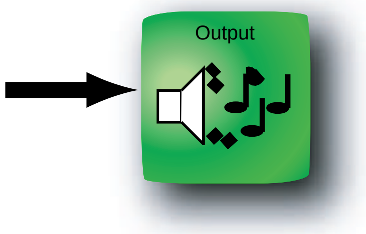


Figure 7: The frequency response of the modified filter



4.3 Testing the system

The system and the progress of the system has been tested with an audible method. No "technical" measurements, like SNR, have been done. All improvements have been done from the audical testing of the system. It has been easy to compare if one extracted sound were better than an other sound.

4.4 GUI

There are two system parts in the Digital Needle. One is the Graphical User Interface (GUI) with which the user "controls" the system and the other part consists of the program which the user "actually" does not see and the DSP board.

In next section there is a description on how the program Scan'n'Play work. This description can also be found in the help files of "Scan'n'Play". The main window of the GUI can be seen in Figure 8. There is also a play window, Figure 10 and a settings window, Figure 9.

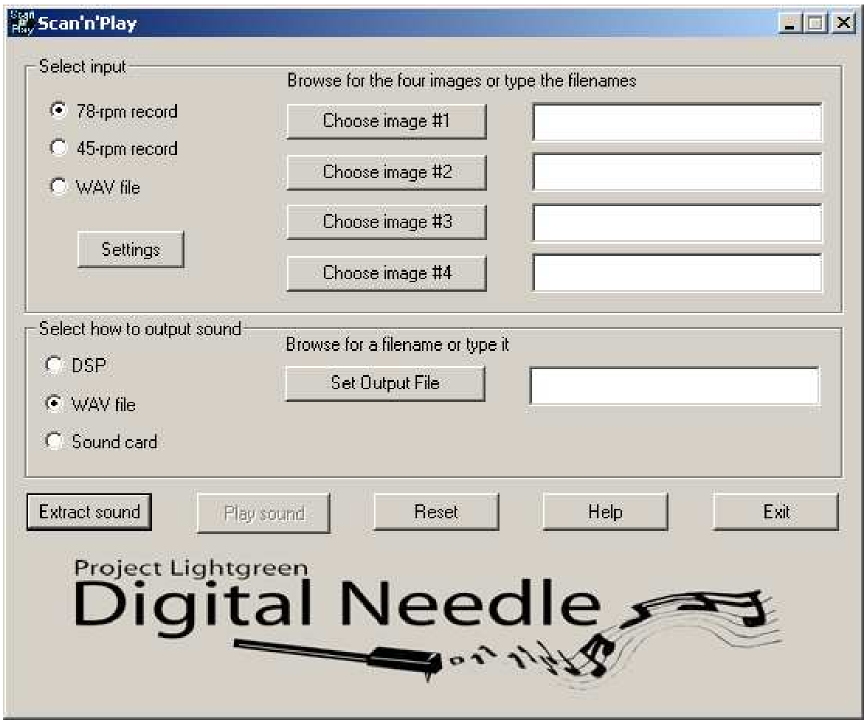


Figure 8: The main window of the GUI

4.4.1 How does the program Scan'n'Play work?

How to scan

Before you can start to create music you have to scan a whole record. Due to the limitations of our scanner we only read the top quarter of the record. This means that you´ll have to rotate the record four times. See section 3.1 and 3.1.1 for limitations and settings for scanning.

Scanning

1. Insert the record into the scanner with the side you want to listen to downwards.
2. Try to match the centre of the record as close as possible with the centre of the scanner.
3. Set the scanning to grayscale and 2 400 dpi. If you have a 45-rpm record which has unsharp tracks, you should also configurate the scanner so that it fixes an underexposed record. (See Figure 3 for an exposure curve).
4. Narrow down the scanning area so that it includes just a bit more than half the record, make sure that you include the whole centrehole (see scanningprocedure).
5. Press scan
6. Save the image. (TIP: give the saved images appropiate names. For example [name]_north, _west, _south and _east)
7. Rotate the record 90 degrees counterclockwise and repeat steps 2 through 5.

When you have the four images, representing each quarter of the record, you are ready to start the program. See Figure 8 for the main window of Scan'n'Play.

Loading the images

1. Choose record speed depending on what kind of record you have scanned.
2. Open the images in correct order.
   (a) image #1 = "north" image
   (b) image #2 = "west" image
   (c) image #3 = "south" image
   (d) image #4 = "east" image
3. Choose an output
   (a) WAV file - Set the name you want the music file to have
   (b) DSP - This option saves the music file temporarily on your hard drive and you are able to play the music
   (c) Sound card - same as DSP but the music is output from your sound card and you can not filter

4. Press "Extract sound"
   - Extracts the music from the images and writes the wav file, if chosen
   - Enables the "Play sound" button
5. Press "Play sound"
   - Takes you to the play music window

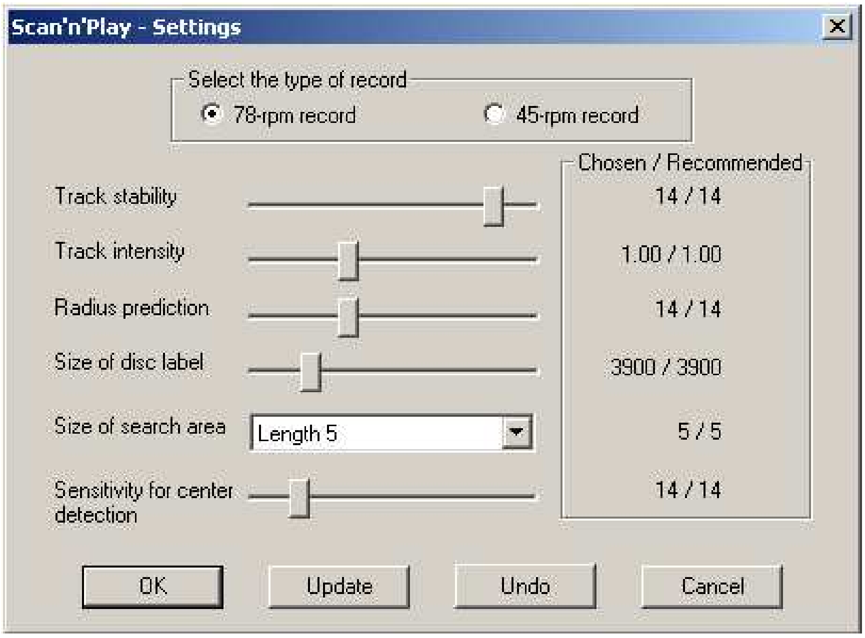


Figure 9: The setting window of the GUI

Settings

1. Select the type of record you have chosen
2. Set the new settings by draging the different slide bars
3. Press one of the following:

- "Update" - Updates the program on current settings
- "Undo" - Resets all the values to the recommended (default)
- "Cancel" - Returns to the main window

- Track stability - A variable between 1-16, that determines the track-reading algorithms sensitivity to noise, where 1 is the most sensitive. A large number makes the algorithm more stable, but makes the algorithm follow the track more slowly which can result in loss of high frequencies.

- Track intensity - A variable between 0.5-2 with 0.5 standard for Vinyl, and 1 for old 78rpm records. This variable determines the threshold used for detecting tracks. Larger values makes the algorithm more sensitive, bur might detect scratches as tracks. Larger values are often needed for discs scanned with light correction for pictures that are underexposed.

- Radius prediction - A variable between 6-30, that determines the number of old values used for estimating the radius for the next sample to read. Low numbers can make the algorithm follow the track faster and therefore detect rapid changes in the music, but on the other hand it can make the algorithm loose track for records with scratches.

- Size of disc label - A variable between 3400-6000, which is the radius of the disc label in pixels. (scanning in 2400dpi results in 100 pixels 1mm). Its important that this value is larger than the largest radius of the disc label, but smaller than the radius to the inner most groove of the record.

- Size of search area - A variable which can take the values of 3.5 and 7. This value determines the size of the area around the predicted radius where the algorithm searches for the track. Within a large area the track will certainly be found, but it can also result in more noise in the music due to dust and scratches on the record.

- Sensitivity for center detection - A variable between 12-25, which controls the algorithms stability in finding the discs outer most edge, whitch is used for detecting the center. Depending on the characteristics of the disc and the quality of the scanned image, you might need to adjust this variable.

Common trouble (FAQ)
Q: No, or too few tracks are found
A: This could be because the scanned images quality are bad. Try adjusting the track intensity. If this doesn´t help check the image quality or try to rescan the disc

Q: To few tracks are found in one of the images
A1: This problem can occur if you didn´t center the disc during scanning. Try adjusting the sensitivity for center detection. If this doesn´t help, rescan the image you´re having trouble with.
A2: The lable might not be cantered around the hole, and therefore having larger radius on one one side. Try adjusting the size of the disc label, so the value covers the largest radius of the label.

Q: The resulting music is not connected correctly.
A1: This problem can occur if the value of the Radius prediction is either to low or to high. Try adjusting this and make a new calculation.
A2: This problem can occur if the quality of the images are bad, and therefore the center point is detected wrongfully in one of the images. Try adjusting the Sensitivity for center detection.
A3: This problem can occur if the Size of the disc label is set to a value lower than the actual size of the label radius. Try to adjust the value so that the value covers the maximum radius of the label

Play Music See Figure 10 for layout of the play window.

- "Play" - Starts the music and disables
- "Stop" - Stops the music and enables "Play"
- "Pause" - Pauses the music or restarts the music if paused
- "Back to main" - Go to the main window
- "FIR-filtering" - If enabled the music is filtered with a FIR-filter

Figure 10: The play back window of the GUI

4.5 System requirements

Equipment (Recommended)

- PC: Pentium 4, 1.7 GHz, 512 MB RAM (minimum 256 MB RAM for Old 78 records)
- DSP-card, Texas Instrument EVM6701
- Scanner, CanoScan5000F from Cannon
- Soundcard and speakers

5 Conclusion and future

Although, this is a DSP course, the project has not used the DSP as much as wanted. This may depend on the fact that the program has to deal with quite large image files.There are four files in total, each being 100 MB big. However, all signal enhancement, such as filtering and more are done by the DSP board, as required in the Objectives for the project.

If an optimal filtering could be found and the program would work as wanted, the Digtal Needle may have a future as an alternative to ordinary old grammophones. The Digital Needle gives a great opportunity to digitize your Old 78 records and your 45 rpm records and even convert them into MP3 files, which you can carry with you in your MP3 player. No matter how good the filtering ever becomes, you will never lose the good old sound from an Old 78 record.

5.1 Problems

The biggest problem in the project has been the task to detect and define the center of the record. At the start of the project this seemed to be one of the easier tasks but this was a big misstake. Several algorithms and combinations of these has been tested. Some of the algorithms seemed to work in Matlab, but failed or showed not to be good enough when translated to C code. In one algorithm a search for the center hole of the record was made. However, this turned out to be very hard due to vast shadows from the scanning.

Another problem has been the train like noise. This noise occurs due to a bluring effect for larger angles. Some algorithms have been tested for correlating the matricies corrrect. If a "correct" center which is the same for all four images or a very good correlation could be done, then the train like noise would be reduced.

References

[1] Digital Needle - A Virtual Gramophone: *http://www.cs.huji.ac.il/~springer/*

[2] *http://www.airtangent.net/playbackgroove.html*, 2003-05-16

[3] John Watkinson, *The Art of Sound Reproduction*, Biddles Ltd, Guildford and King's Lynn, 1998

[4] *http://www.vadlyd.dk/English/RIAA_and_78_RPM_preamp.html*, 2003-05-16

[5] Naim Dahnounn, *Digital Signal Processing Implementation using the TMS320C6000 DSP Platform*, Prentice Hall, 2000

David Bell, *TMS320C6000 DMA Example Application*,Application Report, SPRA529A, Texas Instruments, April 2002

TMS320C6201/6701 Evaluation Module Technical Reference, SPRU305, Texas Instruments, December 1998

TMS320C6000 Programmer´s Guide, SPRU198F, Texas Instruments, February 2001

TMS320C6000 Peripherals Reference Guide, SPRU190C, Texas Instruments, April 1999

TMS320C6000 Assembly Language Tools User's Guide, SPRU186E, Texas Instruments, February 1999

[6] http://sourceforge.net/projects/opencvlibrary