

Estimation of groove position and correction of defects

Philippe Ballestraz and Raynald Seydoux

Project type: Diploma project 2009
Department of information and communication technologies
Telecommunication

Supervisors: Carl Haber, Earl Cornell
Responsible professors: Ottar Johnsen, Frédéric Bapst
External expert: Luciano Sbaiz

Start/end dates : 05/20/2009 – 08/16/2009

1. INTRODUCTION	6
1.1 HISTORY	6
1.2 2D IMAGING	7
1.2.1 VISUAL AUDIO PROJECT	7
1.2.2 IRENE	7
1.3 3D IMAGING	7
1.3.1 DICTATION BELT	7
1.3.2 PRISM 2005	7
1.3.3 3DSEVEN	7
1.4 GOAL OF THE PROJECT (OFFICIAL DEFINITION)	8
1.5 ACHIEVED WORK	8
1.6 REPORT ORGANIZATION	8
2. TASK LIST	9
2.1 WORKING ON THE EXISTING CONFIGURATION	9
2.1.1 UNDERSTAND PRISM 2005 CODE	9
2.1.2 WRITE A SHORT USER'S GUIDE IN 2-3 PAGES	9
2.1.3 STUDY THE EFFECT ON SOUND OF VARIOUS QUALITY CUTS INTRODUCED BY HEZEL AND NOWAK	9
2.2 IMPROVEMENTS ON PRISM	9
2.2.1 IMPLEMENT HISTOGRAMS AND CUTS	9
2.2.2 IMPROVE THE METHOD FOR FINDING GROOVE'S CENTER	9
2.2.3 COMPARE AND COMBINE THE WAV OF THE DIFFERENT DISC-PROCESSING METHODS	10
2.2.4 MODIFY THE PRISM'S CODE ARCHITECTURE	10
2.3 TILTED METHOD OF DATA ACQUISITION AND PROCESSING	10
2.3.1 CONFIGURE THE SCANNER FOR TILTED PROBE MEASUREMENT	10
2.3.2 TAKE DATA ON A VARIETY OF DISCS	11
2.3.3 ADAPT PRISM 2005 TO ANALYZE THIS NEW TYPE OF DATA	11
2.3.4 EVALUATE THE RESULTS	11
3. 78 RPM DISCS	11
3.1 78 RPM	11
3.2 STORED SOUND	12
3.3 READING METHODS	12
3.3.1 MECHANICAL READING	12
3.3.2 ELECTRICAL READING	12
3.4 RECORDING METHODS	12
4. GLOBAL DESCRIPTION 3D SOUND EXTRACTION SYSTEM	13
4.1 ACQUISITION OF A RECORDED SUPPORT'S PROFILE	13
4.1.1 THE CONFOCAL MICROSCOPE (MPLS 180)	13
4.1.2 THE XPS MOTION CONTROLLER AND THE THREE STAGES	14
4.2 STORAGE OF ACQUIRED DATA INTO A FILE	14
4.3 ANALYZE AND PROCESS OF THE DATA	14
4.4 MEASURING, PROCESSING, FILTERING THE SOUND	14

5.	ACQUISITION OF DATA WITH THE TILTED PROBE	15
5.1	THE METHOD	15
5.1.1	THE ANGLE BETWEEN THE BEAM AND THE GROOVE	15
5.1.2	THE SCANNING DIRECTION	15
5.1.2.1	Along the groove	15
5.1.2.2	Across the groove	16
5.1.3	THE ANGLE BETWEEN THE BEAM AND THE SURFACE	16
5.1.4	THE MIRROR	16
5.1.5	OUR CHOICE	17
5.2	THE AUTOFOCUS	17
5.3	Z STAGE	18
5.4	SCANNING BOTH EDGES	19
5.5	SAMPLING FREQUENCY AND ANGULAR SCANNING VELOCITY	20
5.6	DATA ACQUISITION SOFTWARE	21
5.6.1	MODIFICATIONS ON THE SOFTWARE	21
5.6.1.1	Two rotations' directions	21
5.6.1.2	Dark position	21
5.6.1.3	Start and end radius points	21
5.6.2	INPUT PARAMETERS	22
5.6.2.1	File parameters	22
5.6.2.2	Acquisition parameters	22
5.6.2.3	mechanical parameters	22
5.6.2.4	captured data parameters	22
5.6.2.5	Dark position parameters	22
5.6.2.6	Delay parameters	23
5.6.2.7	Tracking parameters	23
5.6.3	SEQUENCE FOR A DISC DATA ACQUISITION	23
5.6.3.1	User's tasks list	23
5.6.3.2	Acquisition application steps	24
5.6.4	DATA FILE FORMATS	24
5.6.4.1	The *.pri file	24
5.6.4.2	The *.bri File	25
6.	READING THE ACQUIRED DATA INTO PRISM	26
6.1	TILTED RAW DATA TO FLAT DATA	26
6.1.1	RAW DATA	26
6.1.2	SUBTRACTING THE STRAIGHT	27
6.1.3	MOVE THE TOP LINE TO ZERO	28
6.2	MATCHING THE PASSES	30
6.2.1	SPACES BETWEEN TWO PASSES	30
6.2.2	MATCHING LINE	31
7.	PROCESSING THE DATA TO GET A SOUND	32
7.1	TRACKING	32
7.1.1	BINNING THE IMAGE	32
7.1.2	FINDING THE BEGINNING OF THE TRACK	32
7.1.3	TRACKING METHOD	33

7.1.3.1	Top of the groove tracking (inbounds tracking)	33
7.1.3.2	Out points tracking (outbounds tracking)	34
7.2	PROCESS	34
7.2.1	FITTING A STRAIGHT	34
7.2.1.1	Top straight	35
7.2.1.2	Slope straight	35
7.2.1.3	Intersection of the two straights	35
7.2.1.4	Error sources	36
7.2.2	TAKING THE TOP AS A REFERENCE	36
7.2.2.1	Top straight	36
7.2.2.2	Slope middle point	36
7.2.2.3	Slope straight	36
7.2.2.4	Intersection of the two straights	36
7.2.2.5	Error sources	36
7.2.3	TAKING THE TOP AS A HORIZONTAL LINE REFERENCE	37
7.2.3.1	Error sources	37
7.2.4	NO REFERENCE	38
7.2.4.1	Storing the variation	38
7.2.4.2	Sorting and throwing out bad variations	38
7.2.4.3	Storing the average of the variation	39
7.2.4.4	Height variations due to the autofocus	39
7.3	FINDING BAD POINTS	39
7.4	CREATING THE SOUND BY DERIVATIVE	39
7.5	INTERPOLATION OF THE BAD POINTS	39
7.6	RESAMPLING THE DATA	40
7.6.1	LINEAR INTERPOLATION	40
7.6.2	UPSAMPLING	40
7.6.3	ACQUIRING AT THE RIGHT SAMPLE RATE	41
7.7	TREATING THE NORMAL AND REVERSE DIRECTION ACQUIRED DATA	41
7.8	COMPARISON OF THE DIFFERENT PROCESS METHODS	42
8.	PRISM 2005 DESCRIPTION	43
8.1	GENERAL DESCRIPTION OF THE GUI	43
8.1.1	WHOLE TREATED DATA RAW IMAGE	44
8.1.2	DETAILS OF A PART OF THE TREATED DATA	44
8.1.3	AUDIO PANEL	44
8.1.4	ANALYSIS PANEL	44
8.2	WORKING DESCRIPTION	45
8.3	ALGORITHMS	47
8.3.1	LOAD THE FILE	47
8.3.2	CLEAN	48
8.3.3	TRACK	49
8.3.4	PROCESS	52
8.3.4.1	Cylinder	52
8.3.4.2	Disc	53
9.	CUTS ON 3DSEVEN	54
9.1	HISTOGRAMS AND CUTS	54
9.2	STUDY OF 3DSEVEN CUTS	55

9.3 MEASURES	56
9.3.1 TEST DISC	56
9.3.1.1 Interleaved acquisition: Disc-testsound-G7_10_907_6_40000_400.pri	56
9.3.1.2 Standard acquisition (one pass): Disc-testsound-G6_10_1814_3_40000_400.pri	58
9.3.2 MUSIC DISC: DISC-MOON-G1_5_907_4_40000_400.PRI	60
10. COMPARISON PRISM—3DSEVEN	61
10.1 TEST DISC: DISC-TESTSOUND-G7_10_907_6_40000_400.PRI	61
10.2 MUSIC DISC: DISC-MOON-G1_5_907_4_40000_400.PRI	62
11. IMPROVEMENTS ON PRISM 2005	64
11.1 HISTOGRAMS AND CUTS	65
11.1.1 FIRST CUT'S TEST WITH A MUSIC DISC: « MOON LIGHT »	66
11.2 IMPROVED DISC FIT METHOD	68
11.2.1 DETECT BAD POINTS	68
11.2.2 NEW SLOPE DETECTION	69
11.3 ALTERNATIVE TRACKING ALGORITHM	71
11.3.1 BY FOLLOWING THE GROOVE'S PATH	71
11.3.2 FIND EACH TRACK'S POINT INDEPENDENTLY	72
11.4 OBJECT-ORIENTED PROGRAMMING	73
11.4.1 CHOICE	73
11.4.2 CLASS DIAGRAM	74
11.4.3 DESCRIPTION OF CLASSES	75
11.4.4 CODE'S BEHAVIOR	77
11.4.4.1 Launch PRISM2009	77
11.4.4.2 Open a file	77
11.4.4.3 Painting events	78
11.4.4.4 Histograms and cuts	78
11.5 MODIFICATIONS ON THE GRAPHICAL USER INTERFACE (GUI)	78
11.5.1 PROCESSING METHOD CHOICE	79
11.5.2 STATISTICS PANEL	80
11.5.3 OTHERS	81
12. FURTHER DEVELOPMENT	81
12.1 INCLUDE VARYING FIBER SPACING	81
12.2 MERGING BOTH EDGES	82
12.3 RESAMPLING TECHNIQUE	82
12.4 LAUNCH, COMPARE AND MIX THE DIFFERENT DISC METHODS	83
12.5 FIX THE PAINTING METHOD	84
12.6 REDUCE THE NOISE ON THE DISC METHODS	85
13. CONCLUSION	86
14. ACKNOWLEDGMENTS	86
15. BIBLIOGRAPHY	87

1. INTRODUCTION

1.1 HISTORY

On March 25, 1857, the printer Edouard-Leon Scott de Martinville invented the phonautograph. This device made a visual image of sound waves on a cylinder, but did not play or reproduce any sounds. Scott used a horn to collect sound, a diaphragm at the end of the horn that vibrated from the sound, a stiff brush bristle attached to the diaphragm, and a rotating cylinder covered with lampblack or blackened paper that recorded the wavy lines from the vibrating diaphragm and bristle. [1] [6]

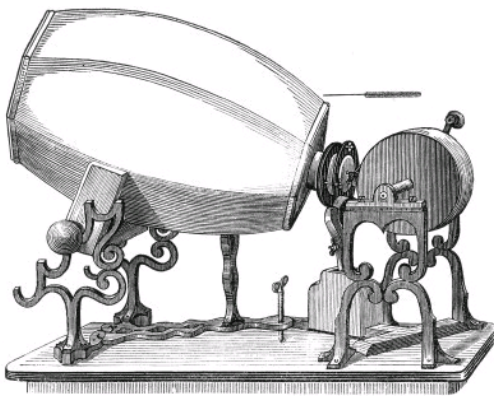


FIGURE 1: PHONAUTOGRAPH

The sound files Scott phonautograms released in 2008 by the First Sounds collaborative were created using Lawrence Berkeley National Laboratory's virtual stylus technology. One of the first recordings was of "Au clair de la lune."

The mechanical cylinder phonograph invented in 1877 by Thomas Edison was the first practical sound recording and reproduction device that signified the beginning of commercial recording, distribution and sale of sound records.

Invented by Emile Berliner in 1887, the phonographic disc overtook the cylinders by 1910 and became the major recording format in the 20th century. Different kinds of material on reading method have been used over the years. In 1942, we have a standardization of the disc recording format. The most standard format became the 78 rpm (rotation per minute) discs. A large part of the world's audio heritage is stored on 78 rpm discs. This format is the one on which we are working on this project. The next chapter will explore in detail the characteristics of this format.

In 1963, PHILIPS imposed its new format of the audio magnetic tape by depositing their patent for free.

In 1982 appears the digital revolution with the invention of the 5-inch CD. The sound is now stored digitally. This opens the way to store the same digital information into different mediums. One of the most commercial revolutions was from Apple in 2001, when it introduced the iPod portable music player.

1.2 2D IMAGING

The two-dimensional scanning process basically consists of acquiring a high-resolution monochromatic picture of the medium. This picture contains information about the slope at each point, whereas a steep section is represented by dark pixels and a flat section by bright pixels. [5]

1.2.1 VISUAL AUDIO PROJECT

The 2D method was employed by Carl Haber and Earl Cornell from LBNL to obtain sound from discs in their so-called project I.R.E.N.E. They were able to extract sound from phonographic records, which in some cases was comparable to the quality of analogue playback systems.

1.2.2 IRENE

A very similar project called VisualAudio is being investigated at the University of Applied Sciences in Fribourg in Switzerland, to retrieve the audio data from damaged phonographic records that either were cracked or whose lacquer layer has shrunk over the years. By taking high-resolution photographs on a film and then scanning these films, the team at EIA FR was able to save an amount of unplayable records digitally.

1.3 3D IMAGING

The three-dimensional scanning process consists of acquiring a height map of the sound carrier instead of a picture representing the slopes. The depth of each point is accurately measured by a confocal microscope to represent the surface of the support. By digital processing, the stylus movement is simulated virtually to extract the sound.

1.3.1 DICTATION BELT

The 3D method was applied in a former diploma project by two students of University of Applied Sciences Fribourg at LBNL to extract sound from dictation belts with a single-point color-coded confocal microscope [4].

1.3.2 PRISM 2005

The single probe was used to extract sound from cylinders. Unlike for discs and the dictation belts, where sound is stored as a lateral modulation of the groove, cylinders have vertical groove modulation, which requires the 3D method. One application called PRISM was developed by Earl Cornell to extract sound from cylinder. The main problem of the single-point confocal microscope was the long duration of data acquisition. This is the reason why LBNL acquired a multiple-point line sensor (MPLS180), which can measure the depth of 180 points simultaneously, obviously promising much more rapid acquisition. The PRISM application has been modified and adapted by Earl Cornell to acquire data from cylinders and discs with the 180 points probe.

1.3.3 3DSEVEN

Last year, Sven Hezel and Severin Nowak were able to use the new MPLS180 optical sensor to extract the 3D shape of a phonographic record, and to make some simple sound extractions. The probe was positioned vertically above the disc. This caused some problems with the bright discs because of the reflection of the probe's light by the groove's edges. [5]

1.4 GOAL OF THE PROJECT (OFFICIAL DEFINITION)

On the 3DSEVEN application or PRISM 2005, there are many problems causing defects. They are due partly to the sensor, in particular some depth points cannot be measured, and partly due to the record itself, in particular scratches, dust and other deteriorations.

In this project, the goal is to improve the sound extraction. This will be obtained by:

- a) Getting the best estimation of the groove position, which corresponds to the instantaneous sound value.
- b) Correct defects in the sound where the groove position is incorrect or lacking.

In order to reach the goals, it is necessary to understand the experimental setup and the characteristics of phonographic records, in particular when they are old and deteriorating.

1.5 ACHIEVED WORK

The work that has been done can be separated into four main points. The first part is a study of the two actual software programs (PRISM 2005 and 3DSEVEN) which can extract sound from the mat surface's discs. This part has been undertaken by both of us.

Seydoux worked then on 3DSEVEN (application written by Sven Hezel and Severin Nowak in 2008) to study the effect on sound of various quality cuts. Taking the best quality of sound from 3DSEVEN, he then compared the three different methods of PRISM and the 3DSEVEN to find out which method was the best according to different discs.

The 3D acquisition of the profile of a disc with the 180 points probe had until now worked well only with mat discs. To avoid the problem of reflection of the disc, Ballestraz changed the acquisition method by tilting the probe above the disc. He had to modify some part of the acquisition process. He also worked on the different algorithms of PRISM to treat the acquired data and extract the sound.

Seydoux also worked on the improvement of PRISM on the existing methods. He added some algorithms and modified the whole application into an oriented object application.

Those different tasks bring us to the end of the project with a new application for acquiring the data, a new oriented object application (PRISM 2009) to analyze and extract the sound from the data, and a method that allows us to extract data from bright discs.

1.6 REPORT ORGANIZATION

The report is organized in 12 chapters. Chapter 2 describes the task list of the project. Chapter 3 presents the 78 rpm discs, their specifications and the different process for recording and reading them. Chapter 4 describes the whole sound extraction system. Chapters 5 to 7 detail the process of 3D sound extraction with the tilted probe above the disc. Chapter 8 describes the data analysis software PRISM 2005. Chapter 9 presents the effect on sound of various quality cuts applied on 3DSEVEN data. Chapter 10 compares different extracting methods to find out which is the best according to the disc and Chapter 11 discusses the improvement done on PRISM 2005. In Chapter 12, we will discuss further development that can be done in future projects. This will be followed by a conclusion concerning the two first months of the project.

2. TASK LIST

This chapter describes the different objectives that we will work on during the project. The duration of the project is 55 working days.

2.1 WORKING ON THE EXISTING CONFIGURATION

2.1.1 UNDERSTAND PRISM 2005 CODE

As discussed with Professor Haber and Earl Cornel, it is more efficient to continue to develop the code done by Earl (the one they are using) than Hezel and Nowak's code. Before adding to it, we have to understand how it works. There is almost no comment in it. To understand the code, we will read it, insert comments, draw some flow graph of the code and explain some of the algorithms.

2.1.2 WRITE A SHORT USER'S GUIDE IN 2-3 PAGES

After studying the code in general, we will write a short user guide of the PRISM's application. This user guide will contain the description of the different GUI's parts and a short description of the options that we can choose when opening a *.pri file. The *.pri file is the file containing the data acquired by the probe. The data stored on the file represent the depth of each point acquired by the probe.

2.1.3 STUDY THE EFFECT ON SOUND OF VARIOUS QUALITY CUTS INTRODUCED BY HEZEL AND NOWAK

There are a lot of differences between the PRISM program and the one of Hezel and Nowak (3DSEVEN). One of the main differences is that the PRISM application bin the data before treating it and the 3DSEVEN one apply different algorithms on the whole acquired data. This task concerns some cuts on the whole acquired data. A cut consists of putting data into various histograms and cutting some part of that histogram by throwing out the data. Hezel and Nowak wrote some cut's code but didn't have enough time to experiment with them. We will first study the cuts they wrote, apply them on some bad quality discs and analyze them.

2.2 IMPROVEMENTS ON PRISM

2.2.1 IMPLEMENT HISTOGRAMS AND CUTS

Many histograms and statistics can be done during PRISM execution; the first goal is to show them graphically in PRISM. The second is to be able to apply cuts on them and analyze the effects on the output sound.

2.2.2 IMPROVE THE METHOD FOR FINDING GROOVE'S CENTER

In PRISM, many different algorithms for finding the grooves' center are implemented. One of these computes one straight on each groove's slope, the intersection of the slopes gives the groove's center. In some cases, many bad points interfere with the calculation and alter the result. We have to find a way to avoid the bad points and compute two right straights.

2.2.3 COMPARE AND COMBINE THE WAY OF THE DIFFERENT DISC-PROCESSING METHODS

As mentioned in the previous section, many disc processing methods are implemented in PRISM. We are going to launch these ones and compare them. PRISM has to be able to display either the result of the launched method separately or altogether.

2.2.4 MODIFY THE PRISM'S CODE ARCHITECTURE

PRISM is coded in C#, a computer language created by Microsoft and part of "The Microsoft .NET Framework." The original version of PRISM is essentially contained in a main file (~6000 line code). Adding more functions or options becomes very complicated and in some cases impossible. The behavior's application is sequential, that is, not very flexible. As C# is an Object Oriented (OO) programming language, it is useful to take all the OO's advantages and apply them on PRISM.

2.3 TILTED METHOD OF DATA ACQUISITION AND PROCESSING

2.3.1 CONFIGURE THE SCANNER FOR TILTED PROBE MEASUREMENT

The LabView program is now configured to capture data from the cylinder or to capture data from a disc. The probe is now placed 90° above the disc. As explained into Hezel and Nowak's final report, their technique works well with mat discs. If the disc is reflecting, the main part of the beam is reflected at 45°.

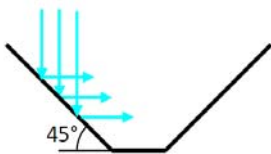


FIGURE 2: REFLECTION ON THE EDGE

To improve the acquisition of reflecting discs, the idea is to direct the beam at 45° of the groove with a mirror. This task consists of finding a way to capture data at 45° by adding the mirror and modifying the LabView program to capture the data as required.

2.3.2 TAKE DATA ON A VARIETY OF DISCS

Capture data from mat and reflecting discs.

2.3.3 ADAPT PRISM 2005 TO ANALYZE THIS NEW TYPE OF DATA

PRISM 2005 is written to treat the data captured on a cylinder or disc when the probe goes to the direction of the groove. The way we want to capture the data from the disc is not following the groove. If we want the beam to be at 45° of the groove during the whole capture, some parts of the disc will be scanned more than one time, because the disc is round and the beam is straight. We first have to modify the PRISM program to rearrange the data. Then, depending on the data, we might use some parts that Earl Cornell has developed and write some others. (Cuts, cleaning, tracking, sound processing...). The different tests will be done during the development.

2.3.4 EVALUATE THE RESULTS

To evaluate the result of our program, we will compare the quality of the sound between the different techniques for acquiring data.

3. 78 RPM DISCS

Any flat disc record made between about 1898 and the late 1950s and playing at a speed around 78 revolutions per minute is called a "78" by collectors. Generally 78s are made of a brittle material which uses a shellac resin (thus they are also referred to as shellac records).[11]

3.1 78 RPM

Although the speeds of the earliest discs can vary considerably from 78 rpm they are still called "78s." The recording speed chosen was a compromise between playing time, groove and needle size, wear on the record and needle, and fidelity of reproduction. At first the speeds were not standardized, but ranged between 65 and over 100 rpm, but in the 1900s an average speed was around 78rpm. The first disc recording machines were weight driven, with the speed controlled by a governor. Different calibrations between machines could lead to speed differences. When electrical recording was first used in 1925, it was used for the soundtrack of films, and so a precise speed was needed to keep in sync with the picture. In the United States using 60Hz mains, the speed of electric motors is either 3600 or 1800 rpm. Using a 3600 rpm motor with a 46 : 1 reduction gives a speed of 78.26. A similar calculation for England using 50 Hz mains gives a speed of 77.92 rpm. These speeds became the standard for the rest of the 78 rpm era. [11]

3.2 STORED SOUND

The sound recorded on the disc is stored on a groove spiral from outside to the center of the disc. When playing back, the sound is produced by the lateral velocity of the needle going through the groove. The lateral variation of the groove's position represents the data stored on the disc.

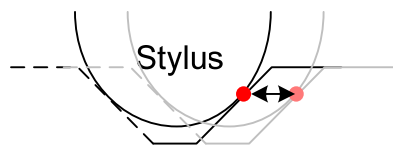


FIGURE 3: STYLUS

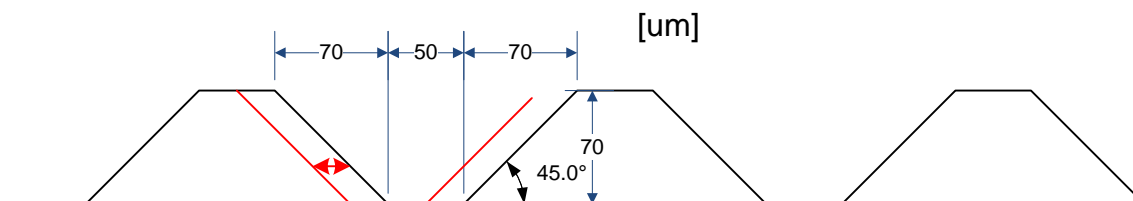


FIGURE 4: DIMENSIONS OF A GROOVE

The figure above shows the typical dimensions of a groove. The angle of the edge is usually around 45°. Those dimensions can be variable according the recording.

3.3 READING METHODS

The principle of reading is to transform the needle variation horizontal position into an audible signal. The first method used was mechanical.

3.3.1 MECHANICAL READING

Going through the groove, the needle transmitted the vibrations to a very thin plate vibrating at the rate of turning. As the power obtained was very low, the sound was amplified with a flag.

3.3.2 ELECTRICAL READING

Gradually, through electronic, vibration, the vibrations were applied to a thin quartz plate which has the particularity of producing a voltage when twisted. This voltage was then amplified with tube amplifiers, and finally reproduced by a loudspeaker.

3.4 RECORDING METHODS

Before 1925, all 78s were recorded by means of the artist singing or speaking into a horn, the power of their voice directly vibrating the recording stylus and thus cutting the wax of the master disc. Collectors call these discs "acoustic" recordings.

After about 1925, 78s were recorded by the artist singing or speaking into a microphone and amplifier which then cut the master record. This allowed a wider range of sound to be recorded. Records recorded by this process are called "electrical" recordings. Collectors can identify these discs by either by listening or by means of small marks in the record surface close to the label. [11]

One of the major disadvantages of all the mechanical carriers was that, as the discs were fragile, and as they were played, the grooves deteriorated.

4. GLOBAL DESCRIPTION 3D SOUND EXTRACTION SYSTEM

This chapter presents the whole system, from the physical audio support to the extracted and processed sound.

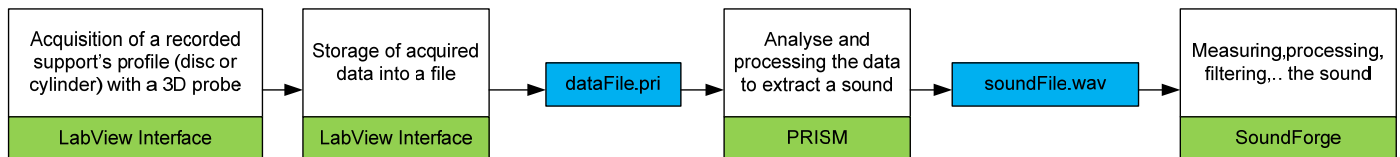


FIGURE 5: SOUND EXTRACTION SEQUENCE

4.1 ACQUISITION OF A RECORDED SUPPORT'S PROFILE

The first step of the process is to acquire the profile of the support. In order to do this, the hardware system is composed of different parts. The main part is a confocal microscope MPLS180 which calculates the depth of 180 points on a surface. We will call it the probe. To go through the whole surface of the disc with the probe, we have a system with three stages. There are two stages to move the disc under the probe (rotation and translation) and one vertical focusing stage to keep the probe in range above the disc. To drive those three stages, we also have an XPS motion controller. There is also an Autofocus Height Detector, which drives the vertical stage.

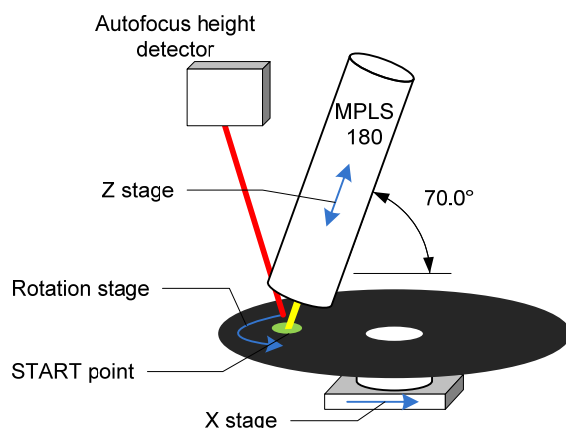


FIGURE 6: ACQUIRING SYSTEM

4.1.1 THE CONFOCAL MICROSCOPE (MPLS 180)

The MPLS 180 can capture 180 points of a 3D profile at the same time. The range of the data that can acquire the MPLS180 is around 400 [um]. This is the difference between the highest and the lowest point that can be acquired by the probe at the same time. The MPLS 180 returns the depth and the brightness of each captured point.

The following figure explains this range on a random profile:

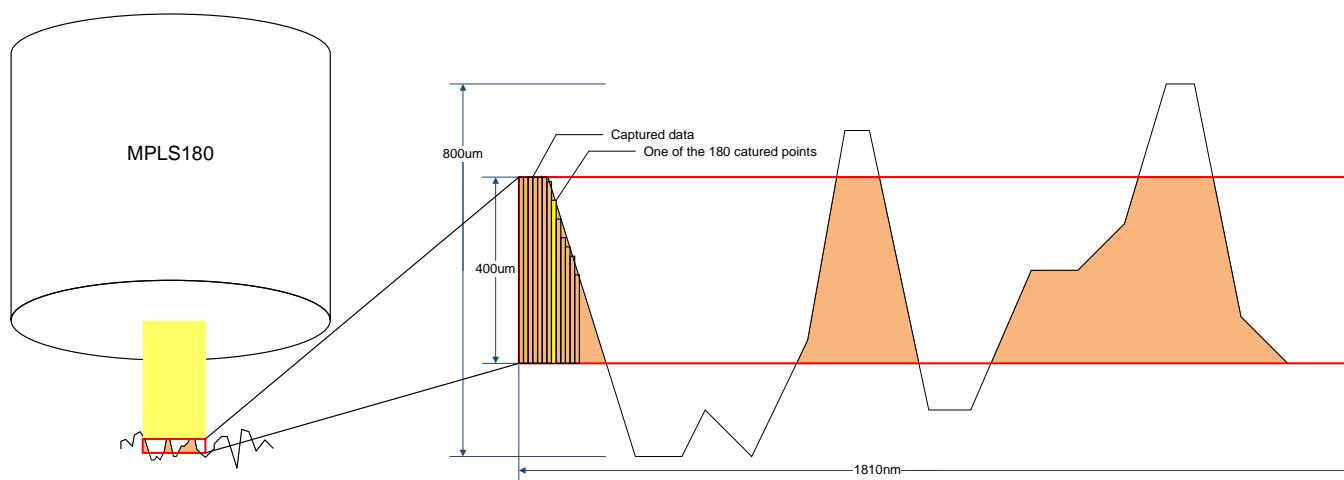


FIGURE 7: MPLS' RANGE

4.1.2 THE XPS MOTION CONTROLLER AND THE THREE STAGES

To acquire the required surface of the support, we use three different stages, which are controlled by a machine called an XPS motion controller.

A more precise description of the probe and the XPS can be found in the document “Extraction of Sound using 3D” [ref.]

To control the capture of the data by the probe and the three stages, we use a LabView interface. This interface will be detailed in chapter “Acquisition of data with the tilted probe.”

4.2 STORAGE OF ACQUIRED DATA INTO A FILE

When reading the data, the program stores them in a *.pri file. This file contains a header which describes the acquired data followed by the whole data. This part will be explained in the chapter “Acquisition of data with the tilted probe.”

4.3 ANALYZE AND PROCESS OF THE DATA

We use the application PRISM developed by Earl Cornell to analyze and process the data. Chapters 8 and 11 describe the application and the improvements we have made to it. The goal of this application is to return a wave sound file from the previous acquired data of the best quality possible.

4.4 MEASURING, PROCESSING, FILTERING THE SOUND

When we get the sound file from PRISM, we can measure; apply some filters or process directly on the sound with application SoundForge 9.0 (Sony).

5. ACQUISITION OF DATA WITH THE TILTED PROBE

This chapter explains why we chose to capture the data with a tilted probe and then how this acquisition is done.

5.1 THE METHOD

5.1.1 THE ANGLE BETWEEN THE BEAM AND THE GROOVE

The method used by Hezel and Nowak in the 3DSEVEN project was to acquire the data vertically with the probe. As the probe reads the light coming back from the surface to measure its height, if the surface is reflective and at 45° from the probe direction, no light will be returned. Some measurements on reflective discs have been tested by Hezel and Nowak.

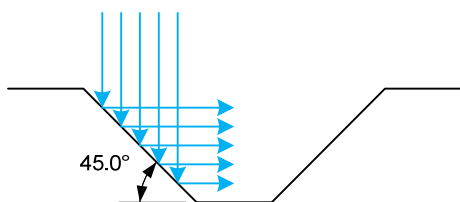


FIGURE 8: VERTICAL BEAM

To avoid this problem and to improve the accuracy of the acquired data, the beam should reach the disc at 45° from the surface.

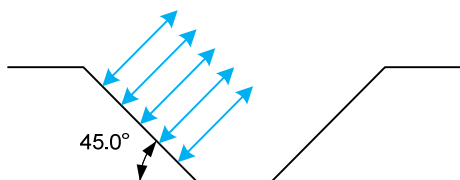


FIGURE 9: TILTED BEAM

5.1.2 THE SCANNING DIRECTION

Independently from the angle of the beam, we have two main options concerning the direction of the scan: along the groove or across the groove.

5.1.2.1 ALONG THE GROOVE

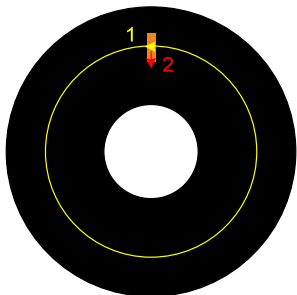


FIGURE 10: ALONG THE GROOVE

The advantage of that direction is that the information of one acquired point and the next one is the same as the sound direction. As explained in point 4.3, if we tilt it at 45° degrees with the mirror, a part of the 180 points will be useless.

5.1.2.2 ACROSS THE GROOVE

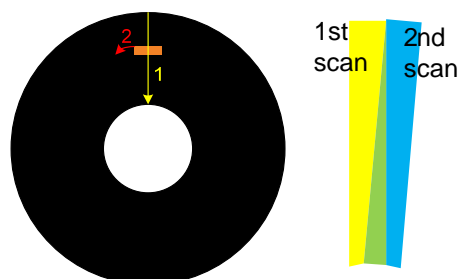


FIGURE 11: ACROSS THE GROOVE

By acquiring bands like that, the information of the 180 points measured is useful, and there is redundancy (which is good) near the centre of the disc, but the reconstruction of the image and the groove become very complicated.

Of those two options, we prefer to capture the data in the groove direction.

5.1.3 THE ANGLE BETWEEN THE BEAM AND THE SURFACE

If we capture a surface of a mat disc vertically, the range of the probe is big enough to cover the profile of the disc.

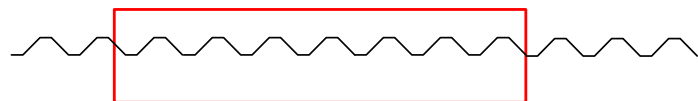


FIGURE 12: VERTICAL CAPTURE

5.1.4 THE MIRROR

We added a mirror tilted at 22.5° to read the data at an angle of 45° . The data received are good, but we get a lot of useless points on the 180 points of the probe.

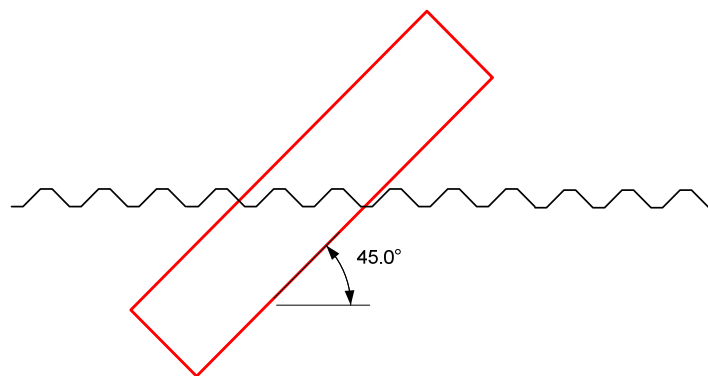


FIGURE 13: TILTED CAPTURE

Another problem is the losses due to the mirror.

5.1.5 OUR CHOICE

According to the advantages and disadvantages described above, we chose to tilt the probe without using the mirror at its maximum position: 20° from the surface. With that compromise, we get a good signal from the edges and more than half of the 180 points are still useful. The chosen direction is the groove direction.

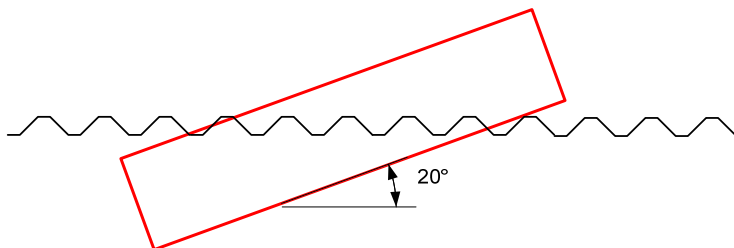


FIGURE 14: 20° CAPTURE

5.2 THE AUTOFOCUS

To keep the acquired data within the range of the probe, we use an autofocus, which sends the information of the height to the XPS, which moves the vertical motor to adjust the probe.

The point of impact of the autofocus touches the disc before the beam of the probe. That is why we need to insert a delay between the measured point and the correction of the probe's height.

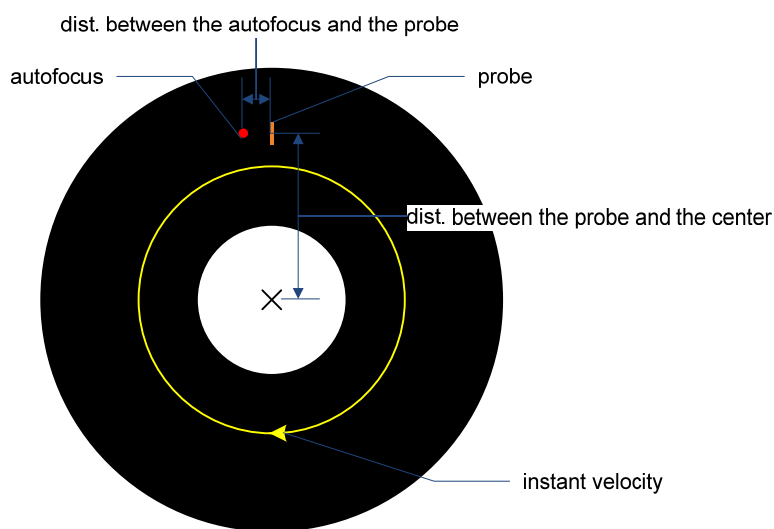


FIGURE 15: AUTOFOCUS PARAMETERS

The delay between the autofocus point and the probe depends on the velocity (deg/sec) and the angle at the center between the autofocus point and the probe. We coded the function, which calculates in TCL. With this code we can read and drive the XPS, and read the value of the autofocus height. The function takes as the main parameters: the distance between the autofocus and the probe and the offset distance between the center of the disc and the probe. The sequence of the function is described below.

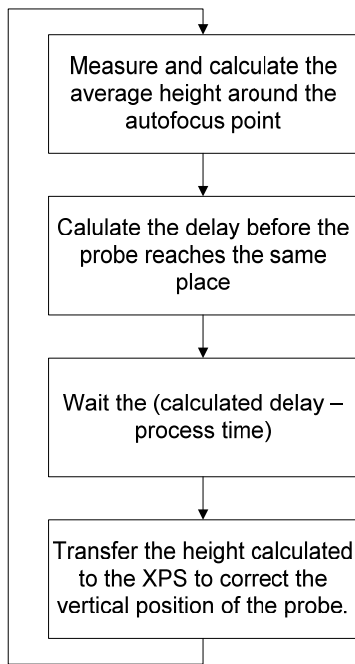
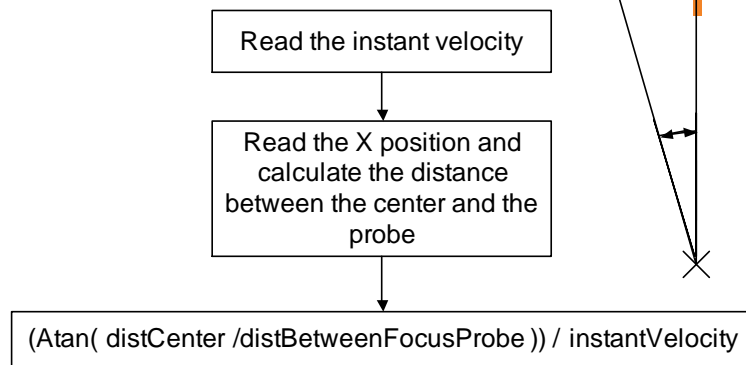
Main function**Calculation of the delay function**

FIGURE 16: AUTOFOCUS FLOW GRAPHS

5.3 Z STAGE

Because the probe is tilted, the modification of the height of the probe during the acquisition process (according to the autofocus) implies a horizontal movement of the acquired data. To avoid that problem, we place the vertical motor in the direction of the probe.

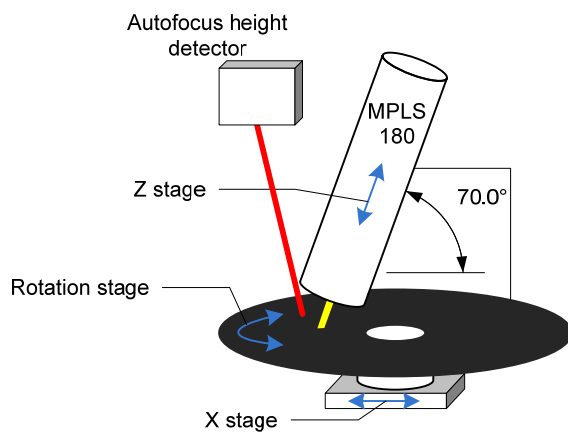


FIGURE 17: TILTED PROBE AND Z STAGE

5.4 SCANNING BOTH EDGES

By acquiring data with the probe tilted, we get only one of the two edges of the groove. To get more redundancy, it would be better to get both edges. We modified the LabView in this way.

Here are the different steps to capture both sides.

- a. Move the disc to the start position

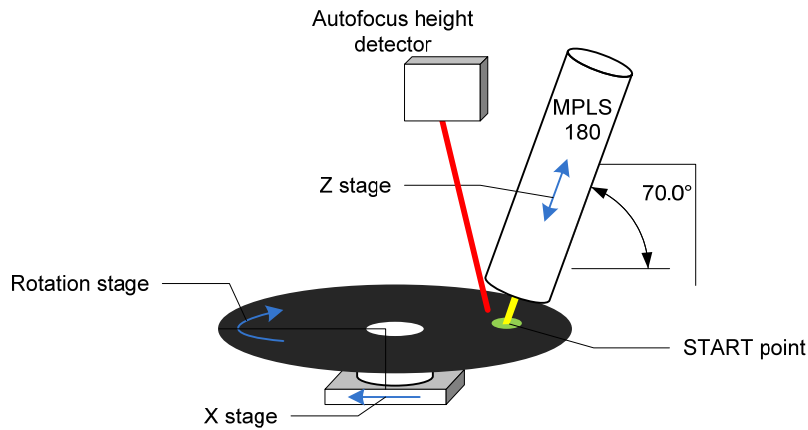


FIGURE 18: HORIZONTAL MOVE

- b. Capture one edge of the groove using the “Along the groove” method described above.

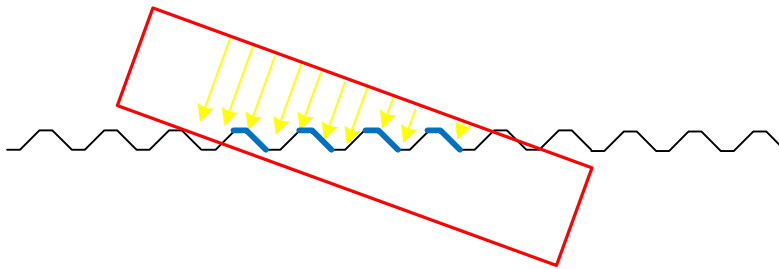


FIGURE 19: FIRST EDGE CAPTURE

- c. Rotate the disc 180°

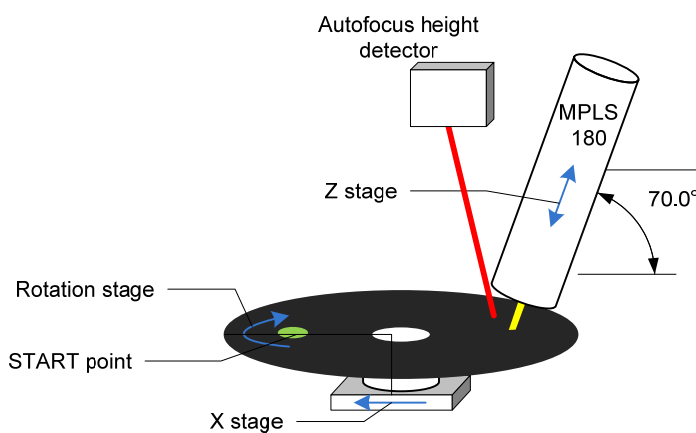


FIGURE 20: 180° ROTATION

- d. Move the X stage to above the start position with the probe.

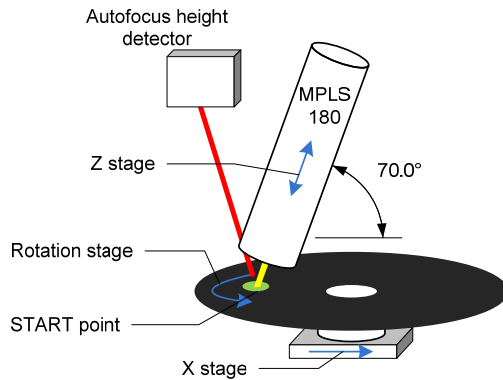


FIGURE 21: REVERSE DIRECTION SCANNING

- e. Capture one edge of the groove using the “Along the groove” method described above. Turning in the opposite direction.

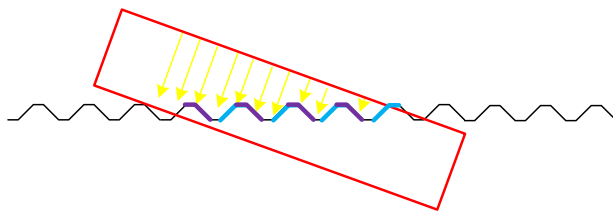


FIGURE 22: SECOND EDGE CAPTURE

- f. The two edges will be taken in two different files.

5.5 SAMPLING FREQUENCY AND ANGULAR SCANNING VELOCITY

As explain by Hezel and Novak, the sampling frequency depends on the MPLS180's acquisition frequency and the rotation speed of the disc. The disc spins at 78 revolutions per minute, which comes to a frequency of:

$$f = \frac{60s}{78rpm} = \underline{\underline{0.769Hz}} \Rightarrow T = \frac{78rpm}{60s} = \underline{\underline{1.3s}}$$

Humans can hear frequencies up to 20 kHz. To respect the sampling theorem, we need at least a sampling frequency of 40000 kHz. Because filtering of spectral repetitions due to sampling is never perfect, we want to sample at a higher frequency. To keep some tolerance, we sample at 52 kHz. This frequency corresponds to: Number of samples on one disc revolution:

$$\# = \frac{f}{T} = \frac{52kHz}{1.3s} = \underline{\underline{40000}}$$

Sample spacing:

$$dphi = \frac{360^\circ}{40000} = \underline{\underline{0.009s}}$$

With an exposure time of 2.5 ms, which gave best data acquisition, angular scanning velocity comes to:

$$\omega = dphi * \frac{1}{\text{exposure time}} = 0.009^\circ * \frac{1}{2.5 \text{ ms}} = \underline{\underline{3.6^\circ/s}}$$

5.6 DATA ACQUISITION SOFTWARE

To acquire the data with a tilted probe, we modified some parts of the LabView interface adapted for the discs by Hezel and Nowak in September 2008.

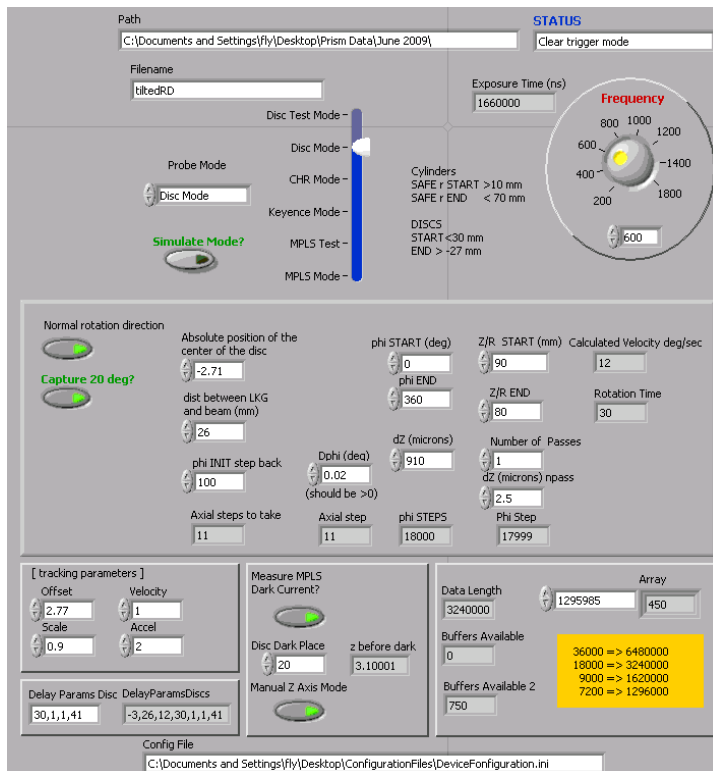


FIGURE 23: LABVIEW GUI

5.6.1 MODIFICATIONS ON THE SOFTWARE

The modifications must be carried out because of the acquisition with the tilted probe. They are mainly due to the capture of both sides of the groove into two different files.

5.6.1.1 TWO ROTATIONS' DIRECTIONS

To be sure to start both acquisitions at the same place on the disc, we need to know the center of the disc. We calculate it manually and introduce it as an input parameter.

We modified the LabView code to read data in both directions. Because the autofocus is fixed regarding to the probe position and the captured data has to pass under the autofocus before the probe, we need to rotate the disc in the opposite direction to capture the second slope of the groove. The data will be in the wrong side for each pass.

5.6.1.2 DARK POSITION

To calibrate the MPLS180, we need to bring the probe out of range. Because we want to capture both sides, we cannot go out of range by moving the X stage or the rotation stage. We need to move the probe vertically. Because the Z stage was command by the autofocus (tracking) before launching the capture process, we could not move it either. We modified the LabView code to start the tracking at the right position during the capture process after the calibration of the MPLS.

5.6.1.3 START AND END RADIUS POINTS

The user can now enter the radius start and end position from the disc's center. The application calculates everything and starts the acquisition at the right place, taking account of the absolute center parameter.

5.6.2 INPUT PARAMETERS

The following describes the main input parameters of the application:

5.6.2.1 FILE PARAMETERS

Path: Destination's folder for the written file
Filename: Written file's name
Mode: Only the disc mode is used to capture data with the tilted probe

5.6.2.2 ACQUISITION PARAMETERS

Frequency: Acquisition frequency
Capture 20 deg: True, if the probe is tilted
False for a normal acquisition (probe vertically above the disc)
Normal direction: True, if we want to capture the first edge in the normal direction
False, if we want to capture the second edge (reverse acquisition)

5.6.2.3 MECHANICAL PARAMETERS

Absolute position of the
center of the disc: X absolute position when the beam is on the disc's center
Dist between LKG
and beam (mm): Distance between the autofocus' beam and the probe's beam on the disc's surface.

5.6.2.4 CAPTURED DATA PARAMETERS

Phi START: angular scan start position (in degrees)
Phi END: angular scan end position (in degrees)
R START: radial start position (from the center of the disc)
R END: radial end position (from the center of the disc)
phi INIT step back: angle before the start position to avoid the effect of the acceleration
dZ (microns): difference between two radial steps (depends on the tilted probe to get the whole data) (in our case 910)
Number of passes: Number of passes on the same data ring
Axial steps: Calculated total number of axial step

5.6.2.5 DARK POSITION PARAMETERS

The dark position is the position to make a calibration of the probe.

Measure MPLS

Dark Current?: True, to calibrate the MPLS, (when acquiring the first edge)
False, acquisition without calibration (if the MPLS is already calibrated, for the second edge)
Disc dark Place: Absolute Z position for calibration
Z before dark: Absolute Z position before the calibration. After the calibration the probe comes back to this position.

5.6.2.6 DELAY PARAMETERS

Parameters for the TCL application that introduces a delay between the autofocus and the probe according to the rotation's velocity and the distance between them.

5.6.2.7 TRACKING PARAMETERS

Parameters to track the Z position according to the autofocus data

Offset: Offset Z position of the autofocus. This position can be read from the autofocus.
Scale: Scale correction parameter

5.6.3 SEQUENCE FOR A DISC DATA ACQUISITION

This point is separated into two parts. The first one is the steps that the user must take to acquire the data of both sides into two different files with the tilted probe. The second one describes the sequence done by the LabView application to acquire the data.

5.6.3.1 USER'S TASKS LIST

- a. Find the absolute disc's center X position with the probe and the "LineSensorExample" application.
- b. Put the probe at the start position and check to be in range.
- c. Close "LineSensorExample" and open the VI "3D-Control-TEST-V3-DISCTilted.vi" with LabView.
- d. Insert the parameters to acquire the first edge on the normal direction.

FileName and Path:	user choice ND (normal direction)
Mode:	DiscMode (4)
Normal rotation direction:	True
Capture 20 deg?:	True
Abs. pos. of the center:	Found on point a
dist between LKG and beam in mm	
Offset:	Read on the autofocus reader
Delay parameters:	30,1,1,41
Phi Init step back:	100 (degrees)
Phi START, phi END:	user choice
R START, R END:	user choice
Measure MPLS Dark Cur.:	True
- e. Start the acquisition of the first edge of the groove.
- f. When finished, change the following parameters:

FileName:	user choice RD (reverse direction)
Normal rotation direction:	False
Measure MPLS Dark Cur.:	False
- g. Start the acquisition of the second edge of the groove.
- h. Close the application.

5.6.3.2 ACQUISITION APPLICATION STEPS

Once the VI is running in disc scanning mode (mode 4), it executes the following sequence:

- a. initialize MPLS180
 - set exposure time
 - save the start position
- b. If dark measurement == True
 - dark current calibration
 - back at start position
 - start the TCL delay code and the tracking mode
- c. Move to starting position.
- d. Set pulse mode for RGV (rotation stage).
- e. Loop for data acquisition in jog mode
 - RGV to start position
 - start RGV JOG
 - read data from MPLS180
 - end RGV jog
 - shift translation stage XM (for multiple axial steps and multiple scan passes)
- f. End of acquisition and application. (The tracking mode is still running to capture the second edge without doing the dark measurement.)

5.6.4 DATA FILE FORMATS

In September 2008 Hezel and Nowak described those files as follows [ref]:

For further treatment of the data, the file format has to be known. The LabView program writes two files containing height and intensity data.

5.6.4.1 THE *.PRI FILE

This file contains the height data of the scanned sample. The structure of this file is very simple, with a header at the beginning of the file, followed by all the data.

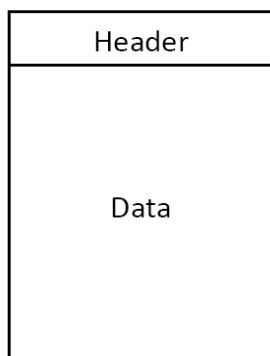


FIGURE 24: GENERAL STRUCTURE OF *.PRI FILE

Header

All information concerning the scanning method is written in the header. Its size is fixed to 10 floating point values containing the input parameters before scanning and information about obtained data size. The values indicate exactly the same as the LabView input parameters and output indicators. ID is the identifier of the file. Its default value is 99999.

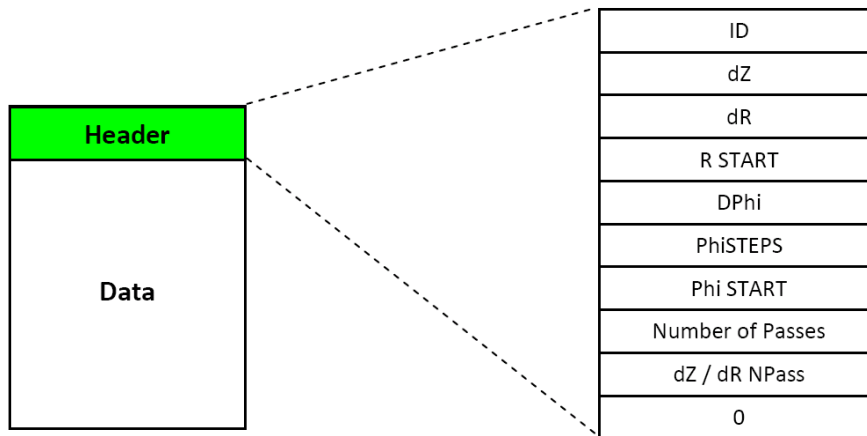


FIGURE 25: DETAILED VIEW OF THE *.PRI FILE HEADER

Data

Each scan returns values for 180 points. These values will be written consecutively into the file. Finally there is a data containing the height of all the points for all axial steps and scan passes.

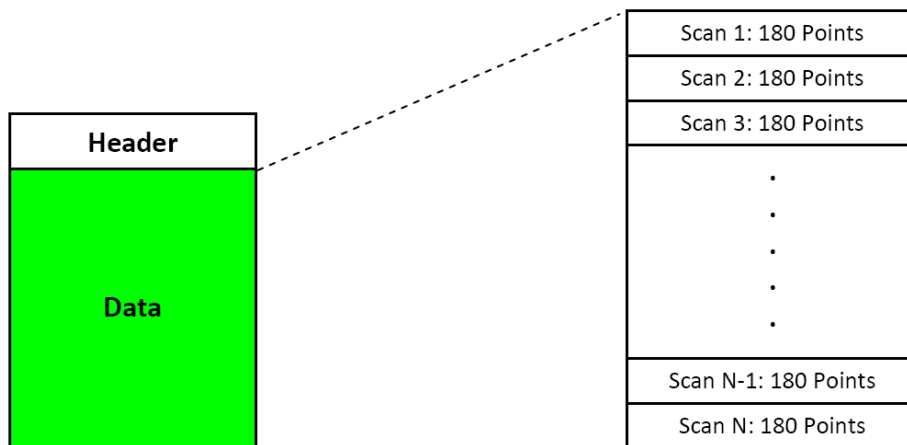


FIGURE 26: DETAILED VIEW OF THE *.PRI FILE DATA

5.6.4.2 THE *.BRI FILE

MPLS180 not only measures heights, it also indicates the intensity of each point. The intensity is saved in the *.bri file. The structure and the header are exactly the same as for the *.pri file. Instead of the height value, the intensity value is written into the file.

6. READING THE ACQUIRED DATA INTO PRISM

Once the data is acquired in a file, we will adapt the PRISM application to read the data as required. This chapter explains the main steps when reading the *.pri file from the acquisition.

6.1 TILTED RAW DATA TO FLAT DATA

6.1.1 RAW DATA

The raw acquired data resemble the following picture:

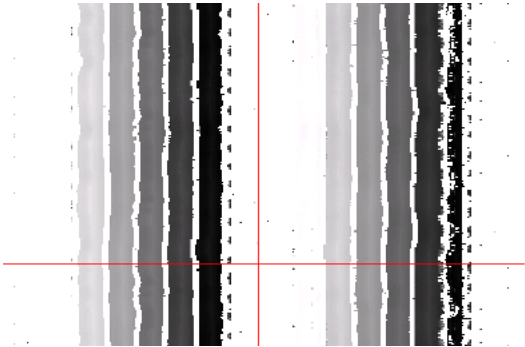


FIGURE 27: RAW DATA

This picture represents only a small part of the acquired data. The dark parts are the deepest parts the white are the highest. If we cut this image horizontally, we get the profile of the data:

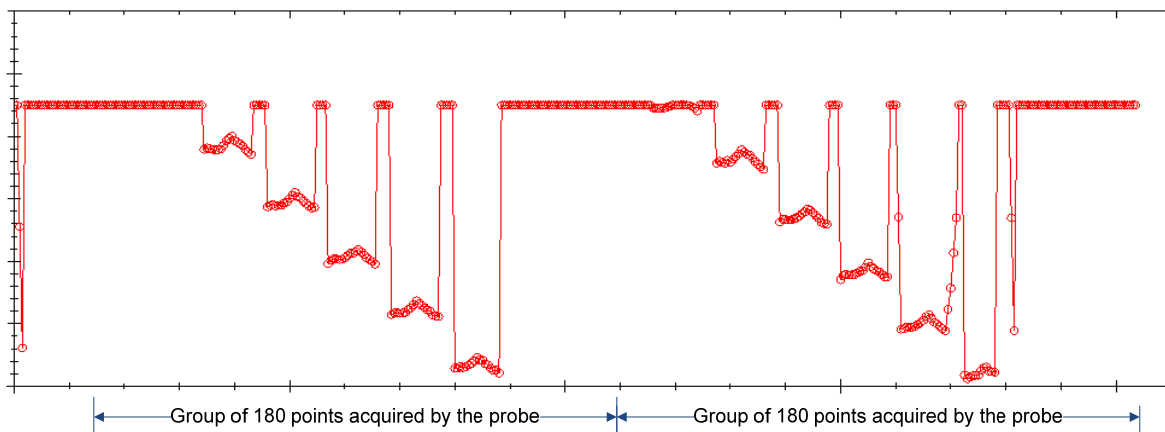


FIGURE 28: RAW DATA PROFILE

The next picture represents the whole acquired data binned vertically by a big factor.

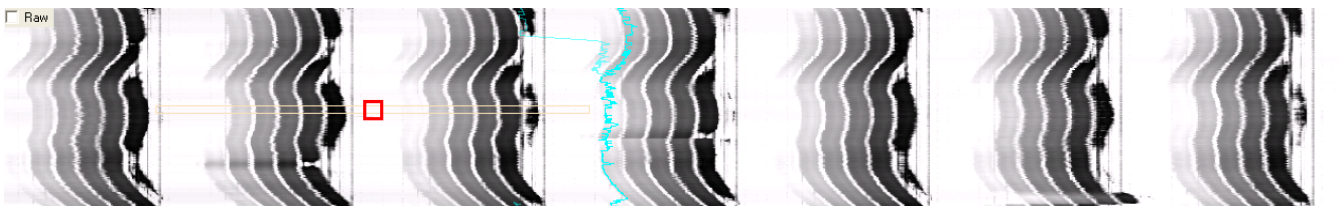


FIGURE 29: BINNED RAW DATA

The 180 points taken at the same time by the probe are drawn on the next graph.

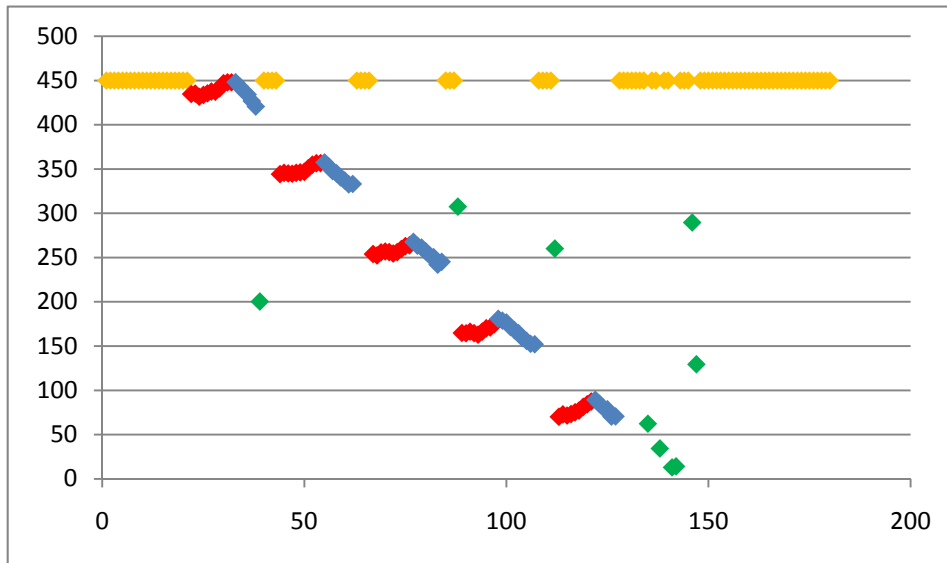


FIGURE 30: SAMPLE OF 180 POINTS

The yellow points are the points that the probe could not measure. Their value is 450. The blue points are the points of the top of the grooves. The red ones are the points of the edges where the information for the music is stored. The green points are bad measured points (we will call them bad points in this document).

6.1.2 SUBTRACTING THE STRAIGHT

The blue points representing the tops of the groove are on a horizontal straight on the real disc. To assemble the 180 points with their 180 neighbors, we need to make that top horizontal. As the tilted straight is due to the tilted probe, the slopes of whole groups of the 180 points' straights are the same. We only need to take few groups of 180 points to know the slope for all the data.

To find the slope of the straight due to the tilted probe, we build a histogram of slopes taken in 500 hundred groups of 180 points and look for the maximum. Those slopes are calculated as follows: We take the out points as a reference to build straights between the points. The slope of each straight is stored in the histogram.

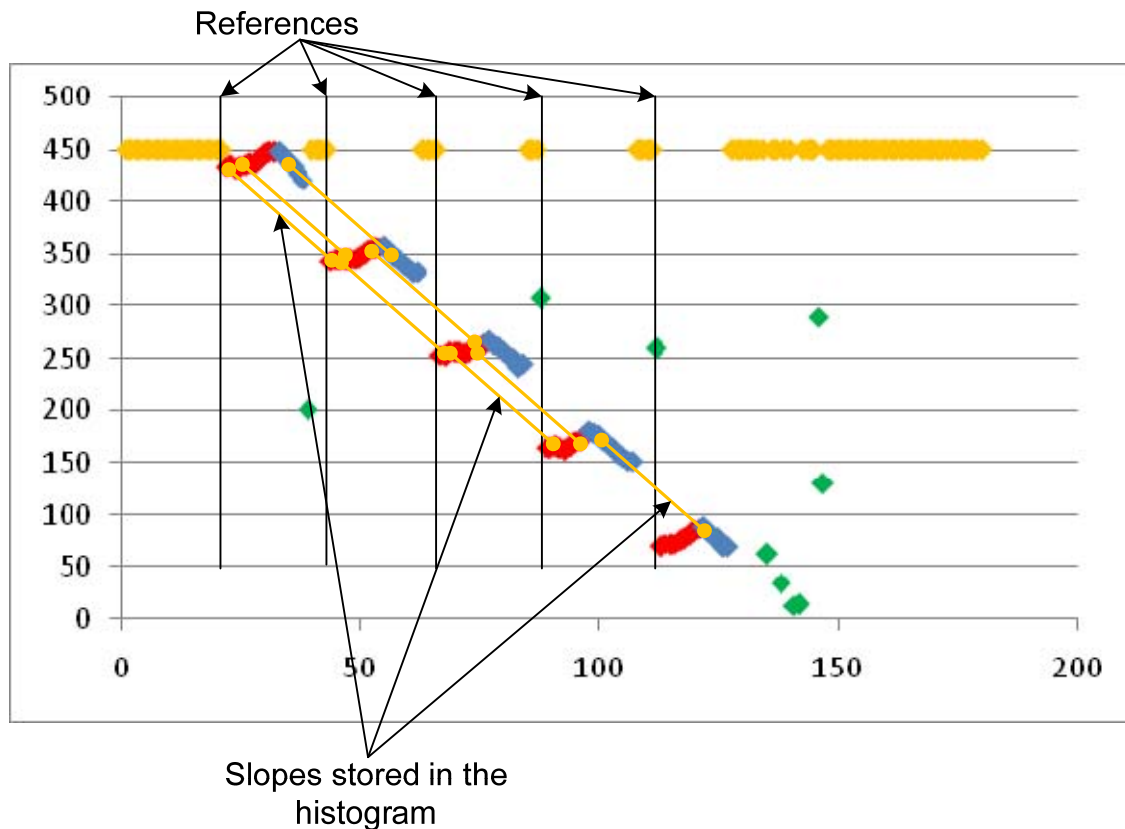


FIGURE 31: SLOPE FINDING

The formula to subtract the straight is: $\text{NewPointY} = \text{OriginalPointY} - \text{Slope} * X$

We do not take the out points (yellow) into account.

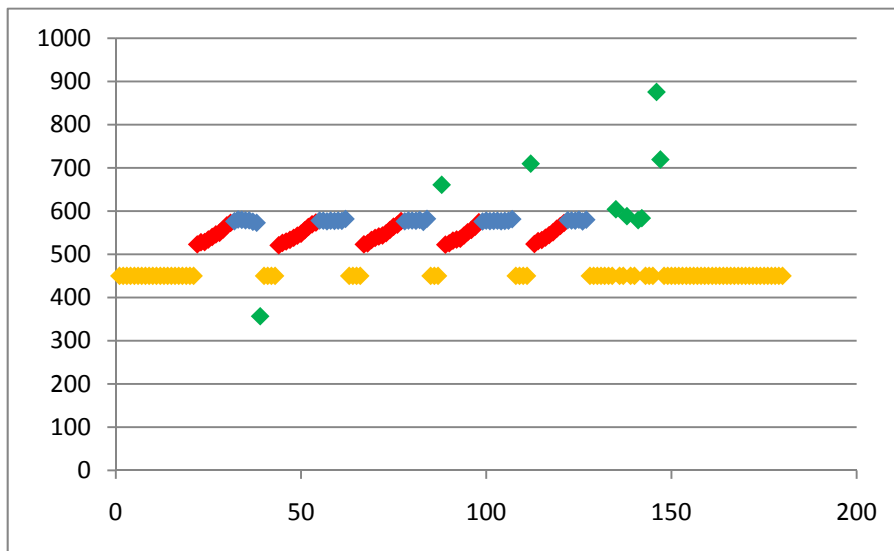


FIGURE 32: STRAIGHT SUBTRACTED

6.1.3 MOVE THE TOP LINE TO ZERO

Because of the z automatic tracking during the acquisition of the data, the height of the blue line can be different from one group of points to the other. We must detect the height of the blue line of each group of 180 points and bring all the blue points to the same height (zero).

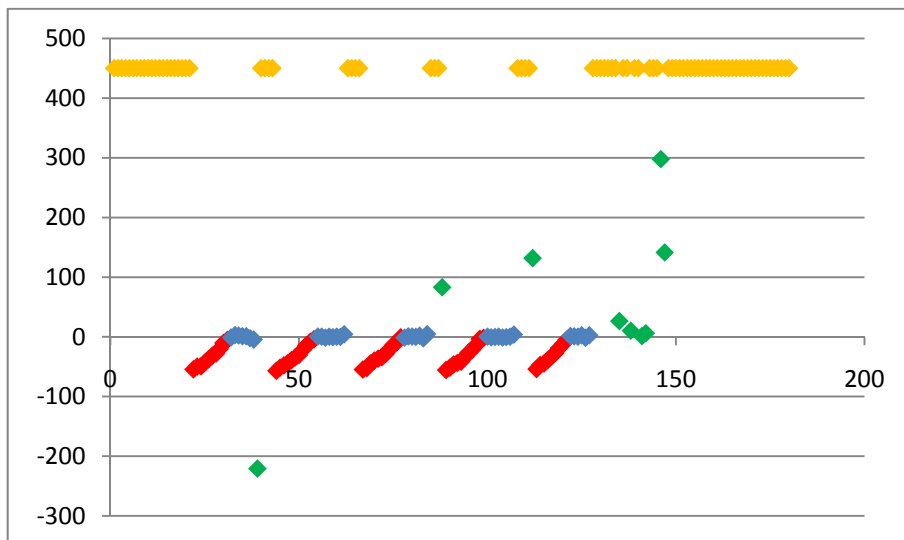


FIGURE 33: TOP SUBTRACTED

In the following picture, we can see that the color of the grooves is now the same. The white lines are the out points.

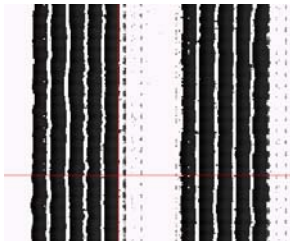


FIGURE 34: RESULTING IMAGE

6.2 MATCHING THE PASSES

As explained in the previous chapter, because of the range of the probe and because the probe is tilted, some of the border points on the 180 are not useful. When we acquire the data, we must take this into account. Between two passes, we cannot move the full 180 points if we want to capture all the information.

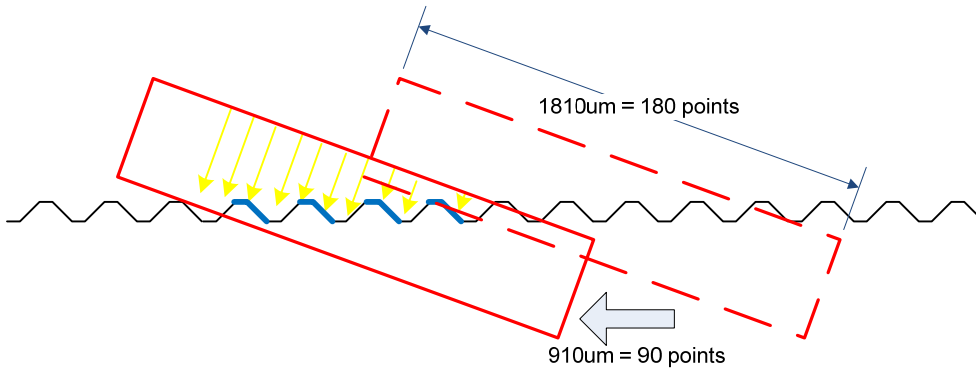


FIGURE 35: MATCHING PASSES

To be sure to obtain enough data to match between two passes, we chose to move the horizontal stage of 910μm = 90 points. When we read the raw data, the out points according to the tilted probe are in the borders. On the following picture, they are the big white spaces between the passes.

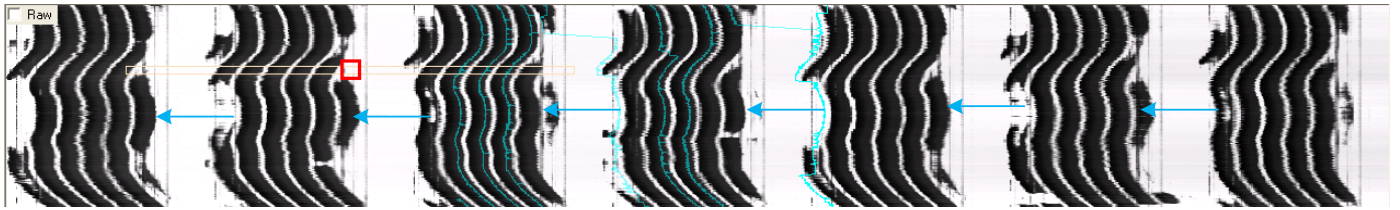


FIGURE 36: IMAGE BEFORE MATCHING

6.2.1 SPACES BETWEEN TWO PASSES

The size of the space that we need to throw out is constant and is a function of the slope and the size of the horizontal stage between the passes.

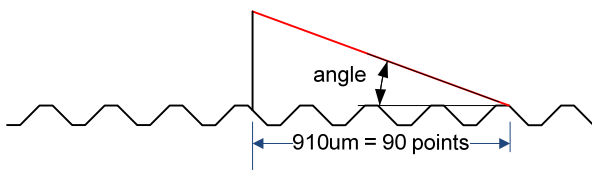


FIGURE 37: SPACE BETWEEN TWO PASSES

$$Space = Xmove / (\cos(\frac{\pi}{2} + \arctan(Slope)))$$

6.2.2 MATCHING LINE

The space is always the same, but we must find where the matching line of the two passes is. The algorithm used for discarding the points of the spaces is explained below.

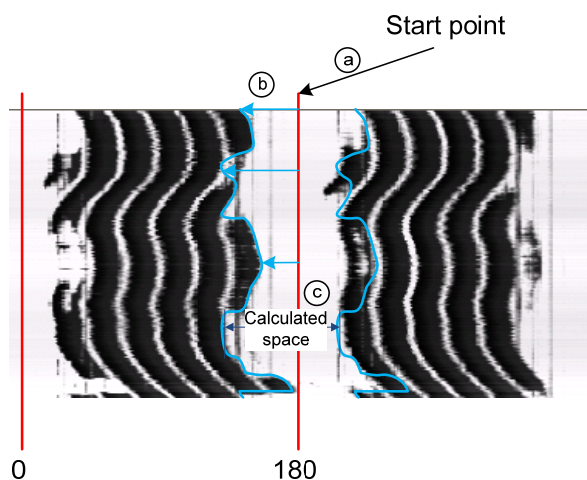


FIGURE 38: MATCHING LINE

- Start from the last point of one 180-point block.
- Go backwards to find the first good data. This operation draws the matching line.
- Discard the data of the calculated space.

After these operations, the data look like a horizontal profile of the disc, containing the information of the top of the groove and one of the two edges. Each black line represents one groove revolution.



FIGURE 39: MATCHED PASSES

The vertical cut of the red line is represented on the figure below. We can easily see the top and the edge of each groove. The points with the 450 value are the out points.

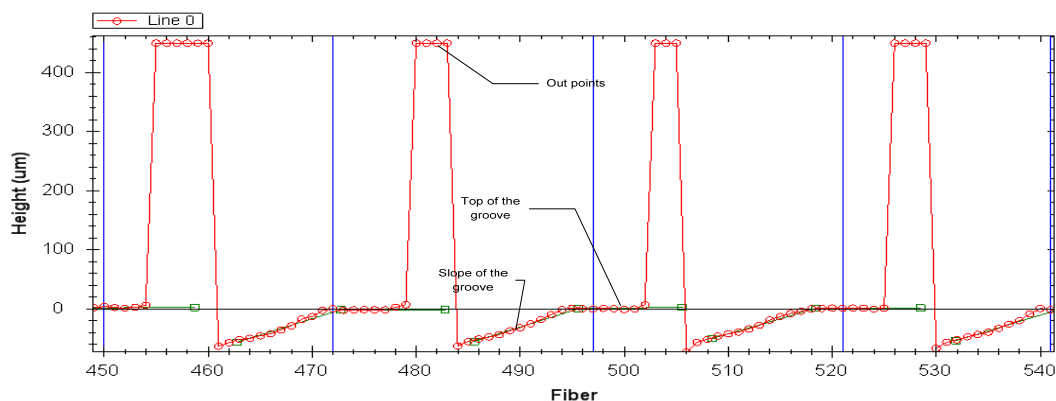


FIGURE 40: PROFILE DESCRIPTION

7. PROCESSING THE DATA TO GET A SOUND

From the read points, the processing of the data is separated into four tasks. The first one is tracking, which consists of finding approximately the path of the whole groove from the beginning to the end of the data. This process finds the precise information of the sound on the groove following the track. Some calculated points will be outside the track. We will call those points the bad points. Those bad points can be found by histograms. All the calculated points put together build the track. To get the sound, we then derivate track function. Finally, we discard the bad points and make an interpolation with their neighbors to replace them.

7.1 TRACKING

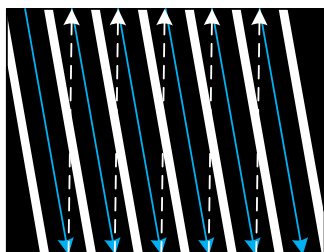
7.1.1 BINNING THE IMAGE

To reduce the calculation time and avoid errors due to a small number of large variations, we first bin the image vertically by a chosen factor. We will then do the tracking on the binned image. The binning is done by averaging n vertical points (n is the binning factor).

7.1.2 FINDING THE BEGINNING OF THE TRACK

Because the disc is not perfectly centered, tracks on the image are not straight. If the disc was centered, the image of six revolutions would resemble the following diagram:

Start point of the
captured groove



End point of the
captured groove

FIGURE 41: TRACK

To find the beginning of the track, we backtrack from the center to the beginning of the image. The backtracking uses the same method as the tracking explained in the following section. We start on the centre of the image to be sure to find a groove. At the beginning of the image, there is some time without groove, as we can see on the following image:

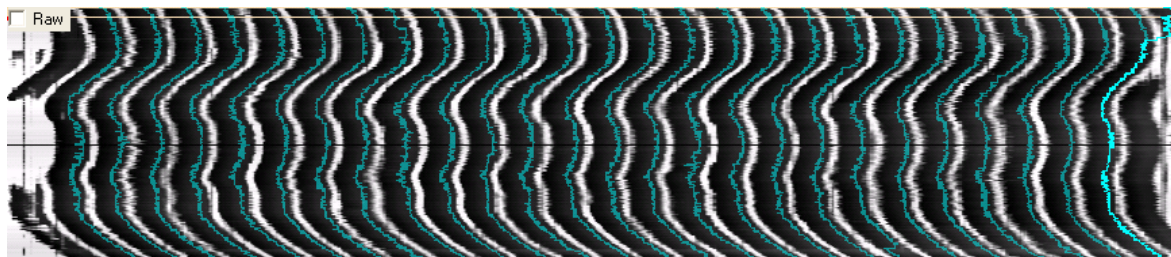


FIGURE 42: TRACKING RESULT

7.1.3 TRACKING METHOD

The goal of the tracking is not to find the sound information, but the groove. We chose to find the top of the groove for the tracking, because finding a horizontal line is faster and easier to code than finding a slope. We then found that this first tracking did not work on some files with this method and developed another one that uses the out points to find the track. Both methods are described below.

The data that we will treat look like this:

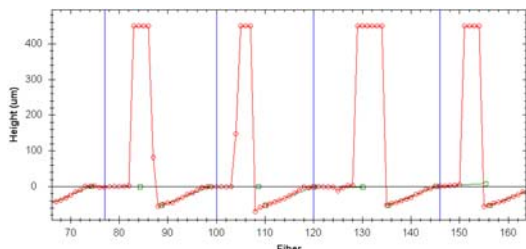


FIGURE 43: DATA PROFILE

7.1.3.1 TOP OF THE GROOVE TRACKING (INBOUNDS TRACKING)

The following algorithm finds the flat tops and returns a point that is near the middle of the top. The average number of point of a top is 7. To be sure to have some top's following points, we will look at 5 points every time.

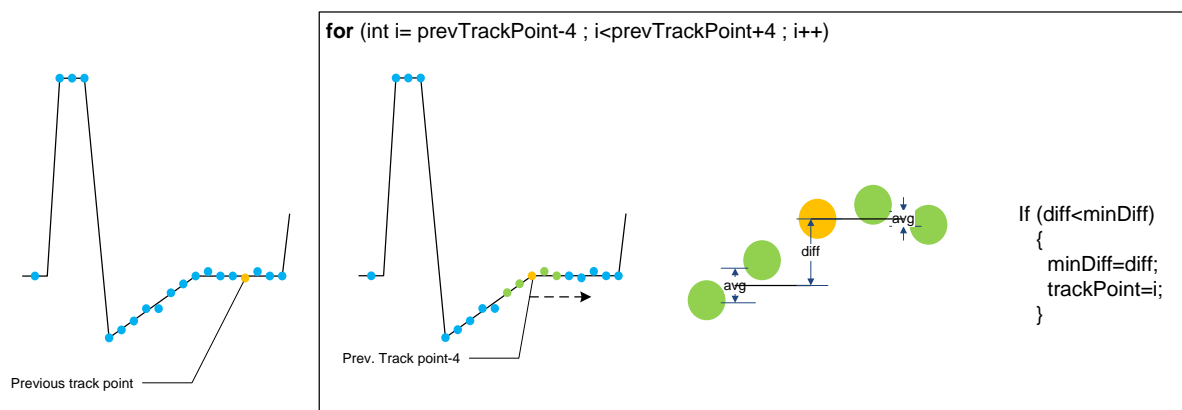


FIGURE 44: INBOUNDS TRACKING ALGORITHM

The algorithm is looking for the minimum difference between the next five points. We do not take into account one of the five points, to avoid some possible bad points.

Once the tracking is done, PRISM shows it on the binned image. It is easy to check visually if the tracking is following the groove. In the following example, we can see a good tracking, except at one place.

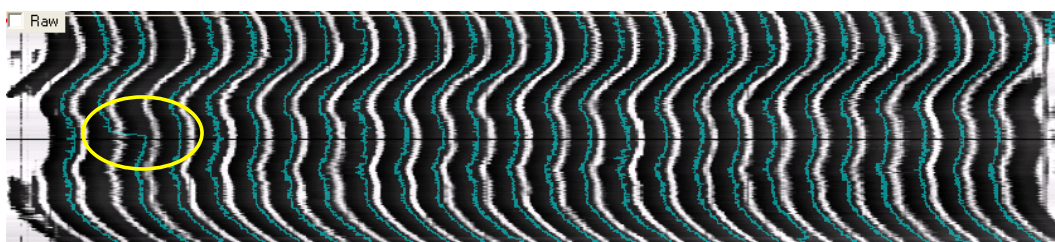


FIGURE 45: ERROR IN TRACKING

7.1.3.2 OUT POINTS TRACKING (OUTBOUNDS TRACKING)

At some points of the binned image, because of the acquisition or because of the quality of the disc itself, a slope can be more flat than a top and the tracking can be affected. This is why we developed a tracking method that takes the out points as reference.

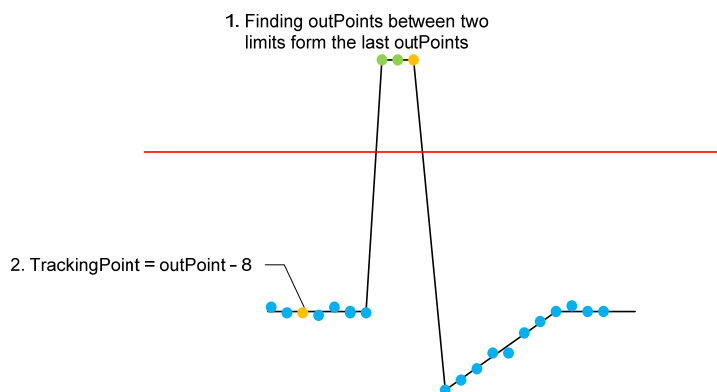


FIGURE 46: OUTBOUNDS TRACKING ALGORITHM

7.2 PROCESS

To process the data, we developed different processes. We then opened the data and processed them with those different methods, keeping the best one. As a further development, it would be interesting to combine those results to obtain the best part of each. The three first methods are based on the horizontal position of the groove, and the last one is based on the height difference between two following points.

7.2.1 FITTING A STRAIGHT

The goal of the process is to build the track that contains the sound information. If we compare it with mechanical disc reading, the information points are the points of the groove in contact with the groove.



FIGURE 47: SOUND VARIATION ON GROOVE

The sound information is the horizontal difference between the red points. As the information is only read horizontally, we can also build the track from any points of the slope, as shown here.



FIGURE 48: SOUND VARIATION ON GROOVE 2

We chose to build the track with the intersection of the top and the slope (orange points above). To find the intersection, we look for the two straights from the tracking point.

7.2.1.1 TOP STRAIGHT

To find the top straight, we use nearly the same method as the tracking with five and six points.

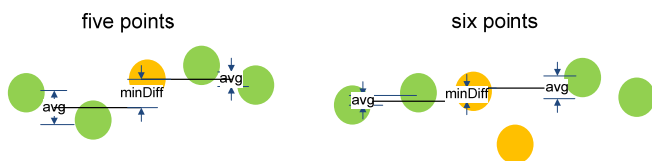


FIGURE 49: MINIMUM HEIGHT FINDING

When the minimum difference is found, we fit a straight on the four side points (Least Squares Method). This straight is the top of the groove.



FIGURE 50: TOP STRAIGHT

7.2.1.2 SLOPE STRAIGHT

There are two main parts to find the straight of the groove's slope. The first one is to find out the average groove's slope's tilt (not developed yet). The second one is to match a straight that has this slope to the bigger number of points. We fit then a straight to the found points. This is the slope straight.

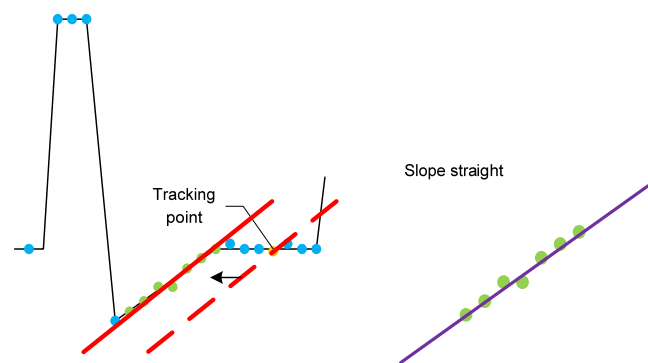


FIGURE 51: SLOPE STRAIGHT

7.2.1.3 INTERSECTION OF THE TWO STRAIGHTS

When we have the two straights, we find their intersection and store the horizontal position of the point into the track.

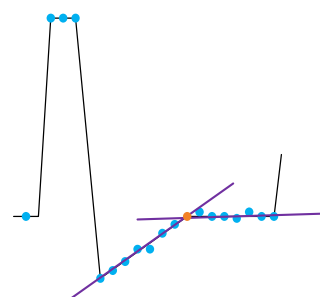


FIGURE 52: INTERSECTION OF THE TWO STRAIGHTS

7.2.1.4 ERROR SOURCES

The main source of error of this method is that the slope of the groove is not constant and might be wrong enough to take some top point into account for the calculation of the slope straight.

7.2.2 TAKING THE TOP AS A REFERENCE

7.2.2.1 TOP STRAIGHT

In this method, the same algorithm as in the first method is used to find the top straight.

7.2.2.2 SLOPE MIDDLE POINT

We then take this top straight as a reference to find a point of the slope. We are looking for the point in the middle of the groove, which means near the half depth from the top. The groove's depth parameter can be entered by the user.

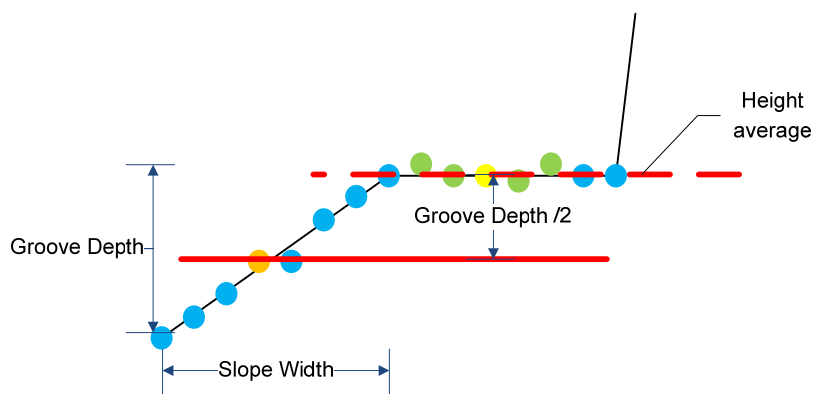


FIGURE 53: TOP REFERENCE

7.2.2.3 SLOPE STRAIGHT

From that middle point, we take its neighbors and itself into account to fit a straight. The number of neighbor points is defined by the slope width, which can be entered by the user.

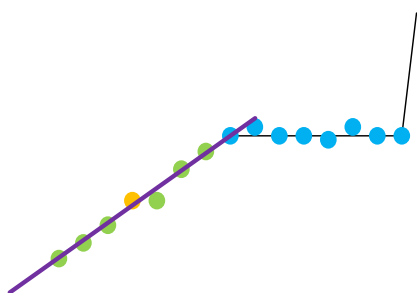


FIGURE 54: SLOPE STRAIGHT

7.2.2.4 INTERSECTION OF THE TWO STRAIGHTS

As in the first method, we then store the horizontal position of the intersection of the top and the slope into the track.

7.2.2.5 ERROR SOURCES

As we take the horizontal position variation of the groove to build the sound, we suppose the top horizontal as well. We can see that the top is never completely horizontal. We should take the variation position between two points in the direction of the top in state of horizontal.

7.2.3 TAKING THE TOP AS A HORIZONTAL LINE REFERENCE

To compensate for a part of the error explained above, we store the intersection of the slope straight and the horizontal straight into the track.

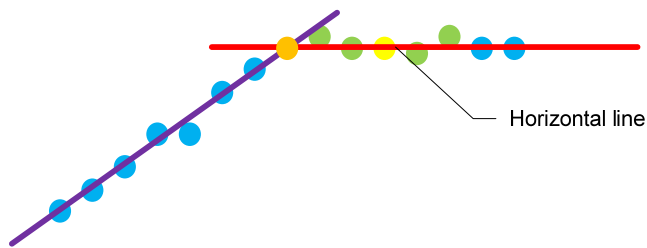


FIGURE 55: TOP HORIZONTAL REFERENCE

7.2.3.1 ERROR SOURCES

In the three previous methods, we take the top of the groove as a reference to find the horizontal position of the intersection point. This reference can be incorrect because of the difference in height between two passes.

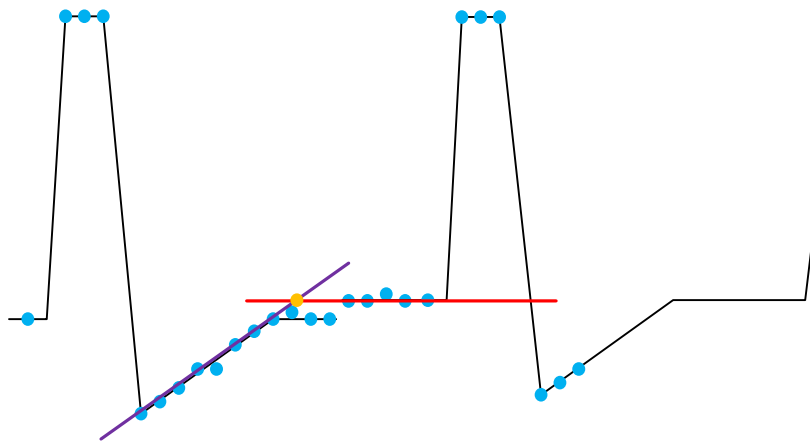


FIGURE 56: HEIGHT ERROR

7.2.4 NO REFERENCE

To avoid the problem due to the reference, instead of storing the horizontal position of the groove, as in the previous method, this method consists of storing the vertical variation between two consecutive points. We directly store the sound.

As described above, the sound, on a disc is produced by the horizontal position variation of the groove. As we can see on the following figure, a horizontal variation of the groove also produces a depth difference measured by the same fiber. We will store this variation to build the sound.

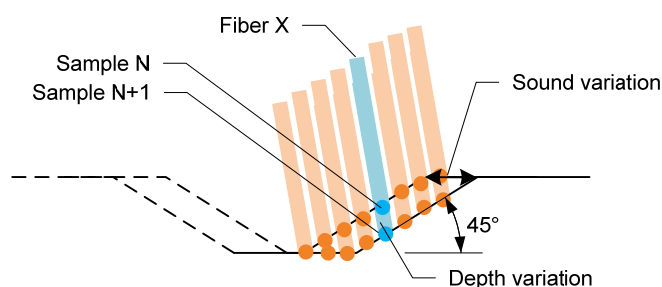


FIGURE 57: TWO FOLLOWING SAMPLES

The variation between two samples for the points on the slope should be similar. To find a point near the middle of the slope, this method uses the algorithm described in section 7.2.2.2.

7.2.4.1 STORING THE VARIATION

From the middle point, we then store the variations of two following samples. The number of points that the method will take in account is defined by the slope width, which can be entered by the user.

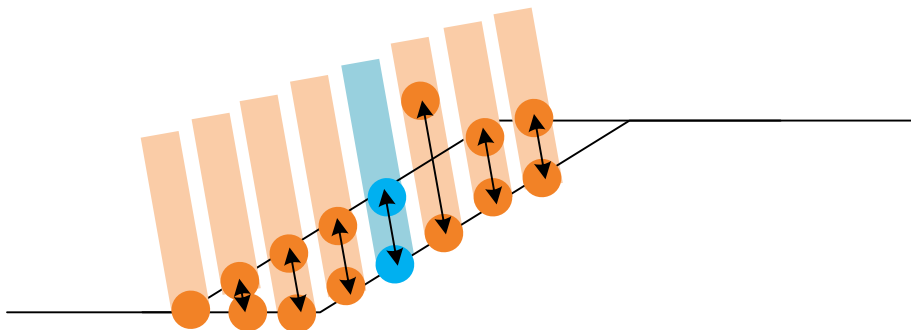


FIGURE 58: VARIATIONS BETWEEN TWO SAMPLES

7.2.4.2 SORTING AND THROWING OUT BAD VARIATIONS

Because of the border and some bad measured points, we will throw out some points before calculating the average variation. We first sort the variation and then throw out the extremes.

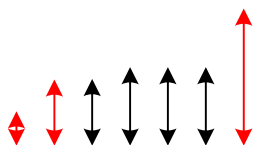


FIGURE 59: VARIATIONS SORT

The number of throwing out points can be entered by the user.

7.2.4.3 STORING THE AVERAGE OF THE VARIATION

We then store the average of the kept variation as the sound. We do not need to do step 7.4 because the stored information is not the position, but already the derivative.

7.2.4.4 HEIGHT VARIATIONS DUE TO THE AUTOFOCUS

The measured depth by one fiber does not only change because of the variation of sound, but also because of the height variation due to the autofocus during the acquisition. The variation velocity of the autofocus is slower than 20Hz. Those frequencies are not audible to the human ear, and those variations are consequently not a problem for the creation of the sound.

7.3 FINDING BAD POINTS

Finding bad points consists mainly of making different histograms of parameters corresponding to the track and defining as bad points the points that are outside the normal Gaussian distribution on the histogram. This choice can be done by the user. To find the bad points, we use the classes Histogram and Cuts of PRISM.

7.4 CREATING THE SOUND BY DERIVATIVE

As explained above in section 7.2, the sound is produced by the difference between two consecutive samples. To get the sound of the track, we will make the derivative of the track points. This method corresponds to the physical effect when reading a disc with a stylus. The sound is produced by the lateral velocity, which is the derivative of the stylus' position.

7.5 INTERPOLATION OF THE BAD POINTS

In PRISM 2005, the bad points are replaced by the linear interpolation of their good neighbor points before applying the derivative to create the sound. This produces a straight between following points, which means a constant slope. When taking the derivative of a constant slope, we obtain 0. We modified it into the version of PRISM 2009 by doing the interpolation after the derivative.

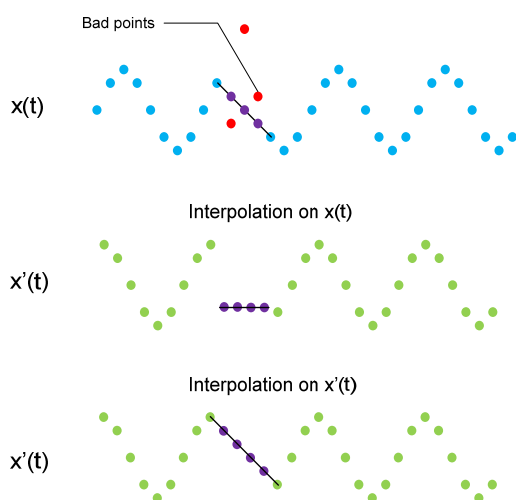


FIGURE 60: INTERPOLATION AFTER DERIVATIVE

We can see on the figure above that the interpolation before the derivative creates some steep variation on the sound, which translates into a “click” sound.

7.6 RESAMPLING THE DATA

7.6.1 LINEAR INTERPOLATION

The actual technique used in PRISM to resample uses the linear interpolation. A ratio is first calculated between the acquisition sampling rate and the output sampling rate desired.

$ratio = speed/60 * samplingAcquisitionRate / outputRate;$

In our case : $ratio = 78rpm/60 * 40,000 / 44,100$

```
// linear
for (int i=1; i<fdt.Length; i++) {
    double d1 = i*ratio + end;
    int i1 = (int)Math.Floor(d1);
    fdt[i] = fdt2[i1]*(i1+1-d1) + fdt2[i1+1]*(d1-i1);
}
```

Depending on the ratio, the code above applies a linear interpolation on the data. The result signal is a sum of triangles in the time domain. This corresponds to a sinc^2 filter in the frequencies domain.

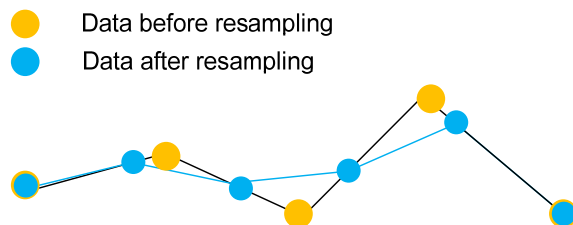


FIGURE 61: LINEAR INTERPOLATION RESAMPLING

The problem of the linear interpolation is that this process introduces a sinc^2 filter.

$$g(t) = \text{tri} \left[\frac{t - T_e}{T_e} \right]$$

$$G(f) = T_e \text{sinc}^2(fT_e) e^{-j2\pi fT_e}$$

Depending on the ratio, this filter can cut some useful frequencies.

7.6.2 UPSAMPLING

To avoid this effect, it would be better to resample using upsampling by rational fraction, which consists of upsampling by a factor M low-filtering and decimate by a factor N.

As explained by R. Riedi, in « Signaux et Codage 2, EIA-FR, 2009 11 » some few rules have to be followed:

- First upsample, then downsample (preserve signal quality)
- Low-pass filters after upsampling and before downsampling can be combined into the one filter with the smaller pass-band.
- Up- or downsampling by large factors should be done in steps, keeping N and M below or at 6 at most.

The advantage of this method is that the filters chosen between upsampling and downsampling can be better than a sinc^2 filter.

7.6.3 ACQUIRING AT THE RIGHT SAMPLE RATE

To avoid the resampling, it would be better to acquire the data at the right sample rate.

$$\text{number of samples} = \frac{44100\text{Hz}}{1.3\text{s}} = 33923$$

7.7 TREATING THE NORMAL AND REVERSE DIRECTION ACQUIRED DATA

All the different algorithms above are described for the normal direction acquisition. As described in section 5.4, both edges are captured in two different files. To differentiate those two files, the ID number is different in the header: 99998 for the normal direction and 99997 for the reverse direction. For the moment they are treated separately as two different objects in PRISM 2009. Some parts of the code are different depending on the direction. The slope of the edge is negative compared to the other. The tracking is done on the other side from the out bounds points. The process is evaluated from the tracking in the opposite direction.

We can see thereafter the two sounds extract from the normal and reverse captured directions from the same disc. We can easily see that the bad points, mainly the peaks, are not always at the same place on the sound. This means that combining the two edges by doing some smart choices will improve the final sound.

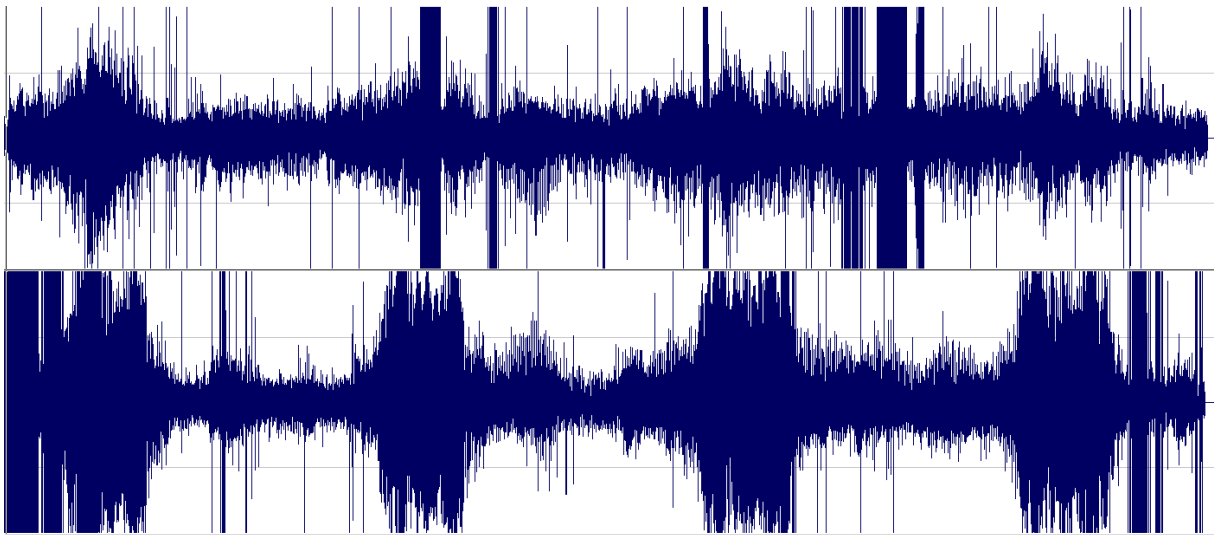


FIGURE 62: SOUNDS OF NORMAL AND REVERSE DIRECTION

7.8 COMPARISON OF THE DIFFERENT PROCESS METHODS

To compare these four processing methods, we are going to use the same *.pri file and the same slope parameters to extract four different sounds. We will then compare the frequency spectrum of those sounds. On a music disc it is impossible to compute a true Signal to Noise Ratio (SNR), because we cannot measure the noise level alone. We will arbitrarily choose the highest peak compared to the lowest point for each method. The file used to make those measurements is: tiltedND_100_910_7_18000_600.pri. To compare those methods without the effect of the resampling, we kept the original sampling done during the acquisition.

Method	Input parameters	Peak [dB]	LowPoint [dB]	Ratio [dB]
Fit	Slope width: 70 Slope depth: 60	-60	-73	13
TopRef	Slope width: 70 Slope depth: 60	-57	-78	21
TopFlatRef	Slope width: 70 Slope depth: 60	-57	-77	20
NoRef	Slope width: 70 Slope depth: 60 Outliers: 2, 2	-48	-76	28

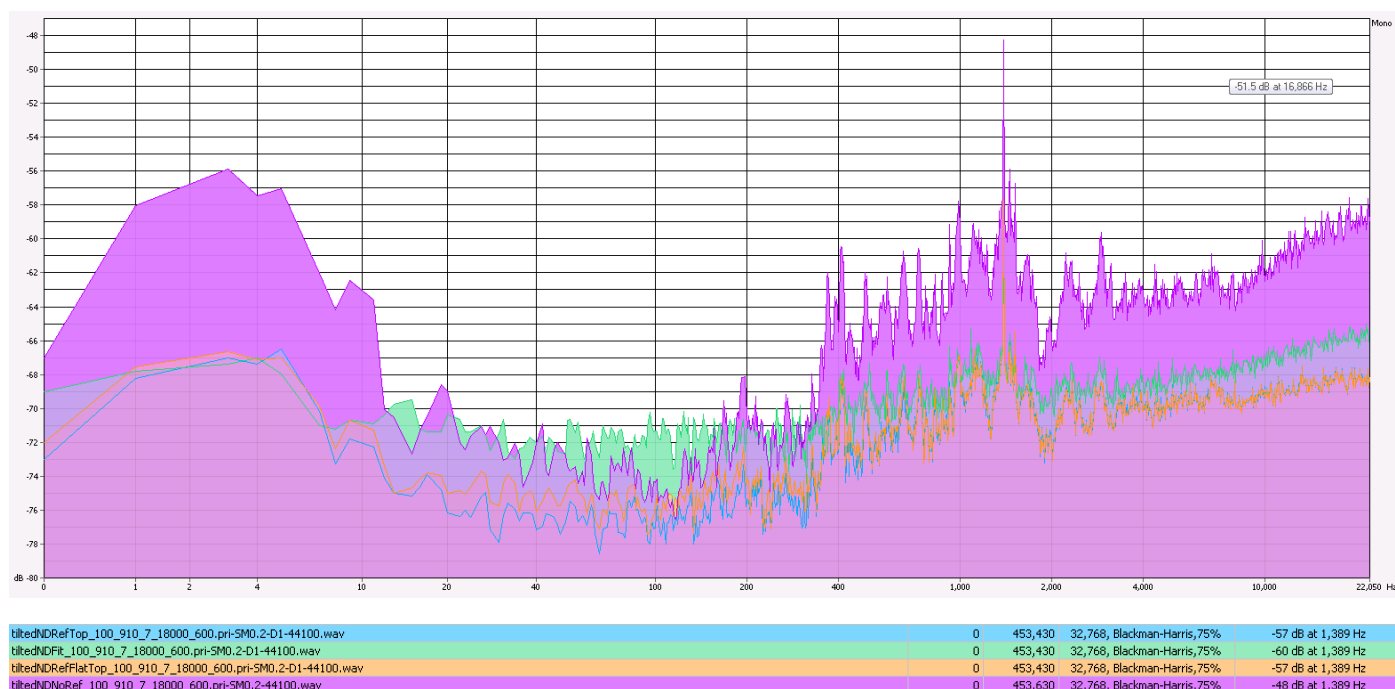


FIGURE 63: COMPARISON BETWEEN THE FOUR TILTED METHODS

We can see that the No Reference method is the one that seems to have the best ratio. The fit method, the first we developed, is the worst, and the two Top References are comparable. This represents the improvement achieved during the development between the different methods, and trying to ensure that each new method avoids the sources of errors involved in the previous method.

The track and process methods described in this chapter will provide the sound to which we can apply cuts as described in Chapter 10.

8. PRISM 2005 DESCRIPTION

The goal of PRISM 2005 is to extract the sound of cylinders or discs from a file containing the surface of those supports. The file contains the depth of each point of the surface measured by a multi-point line sensor (MPLS180, which measures the depth of 180 points simultaneously).

One of our first tasks was to understand the PRISM program on which we were going to work and to modify it to improve the sound tracking of a *.pri file. The program had no comment, and nearly all the methods were written into the same file. We started to read it and comment on each method and each part of the code to understand it. We drew some flow diagrams to explain the program and some algorithms associated with it.

8.1 GENERAL DESCRIPTION OF THE GUI

The GUI of the application is as following. The first main bar contains the different parameters we can choose to acquire, clean, track and process the image. Below that bar there is the bar to read the sound.

The horizontal image represents all the acquired data. The image on the right is a detailed part of the global image. We can see different kinds of images on the right according the parameters (flat, sub flat, cleaned, blobs...) we can see different graphs on the last frame.

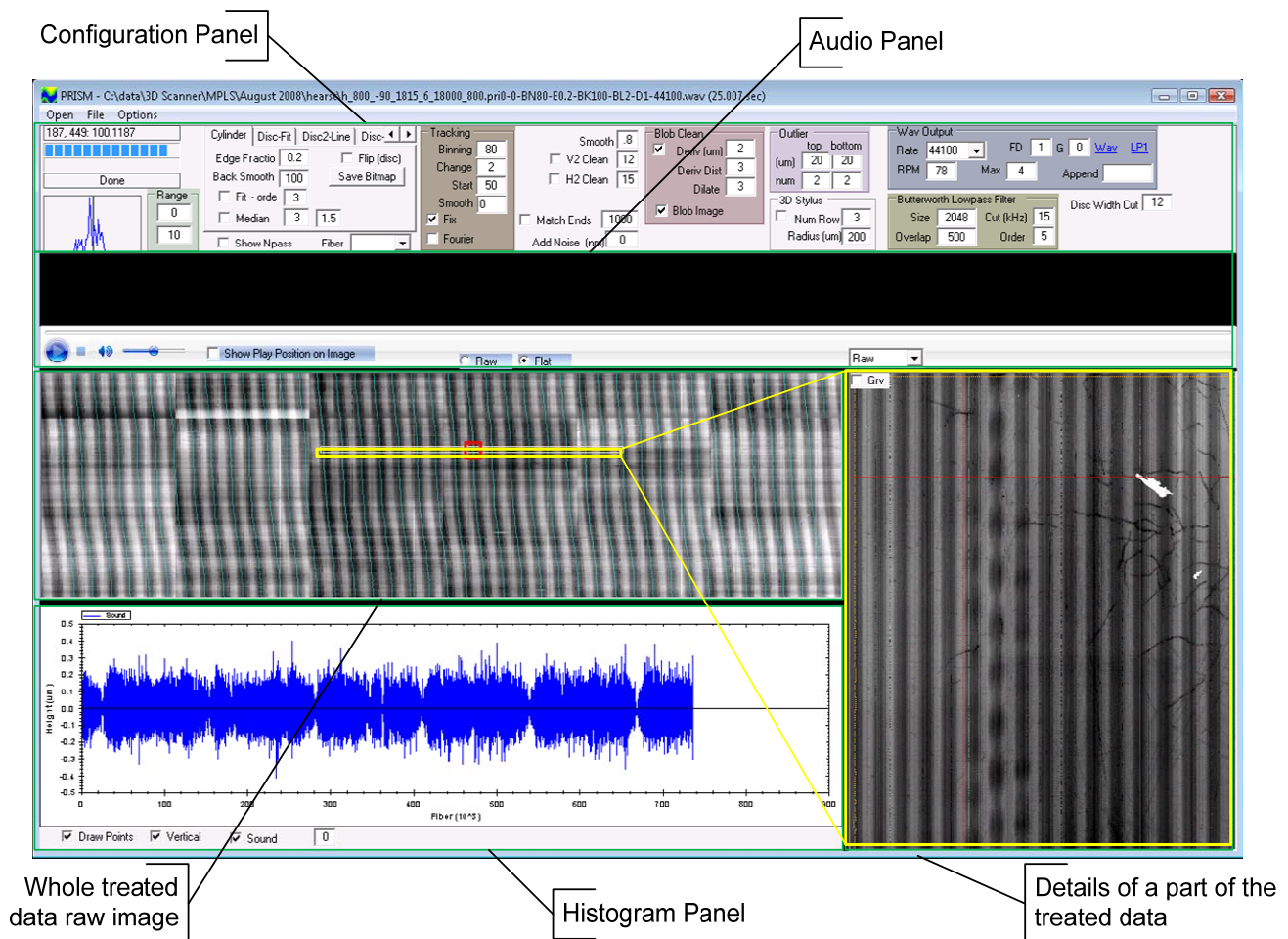


FIGURE 64: PRISM 2005 GUI

8.1.1 WHOLE TREATED DATA RAW IMAGE

This image is the illustration of all the treated data. We can choose which data to treat by changing the **Range** options (starting point and ending point). Only the data between those will be treated by the application. The light blue lines illustrate the tracking of the groove.

8.1.2 DETAILS OF A PART OF THE TREATED DATA

This image is the detailed image representation of the yellow frame on the raw image. According to the combo box's option, this image illustrates the data after different treatments.

Raw	Raw data without treatment
Flat	Average of the groove
Sub-Flat	Raw-Flat. Data without the average groove. (Audio and blobs)
CleanH	Image cleaned horizontally
Fourier	Image filtered along groove by a Butterworth Low pass filter
Thresh	According to the Blob's options, the Thresh image represents in BW the bad (blobs) and the good points
Blob Clean	Image after blob cleaning (the blob's cleaning is done from the "thresh" image on the sub-Flat image) (Option used for cylinder)

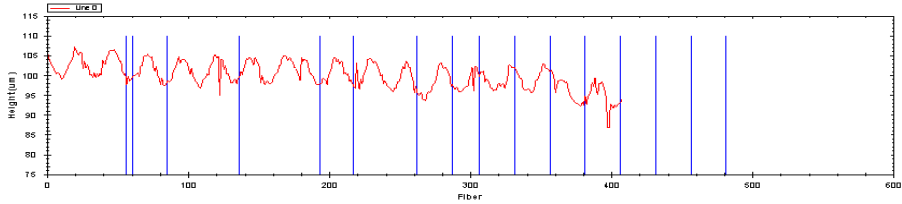
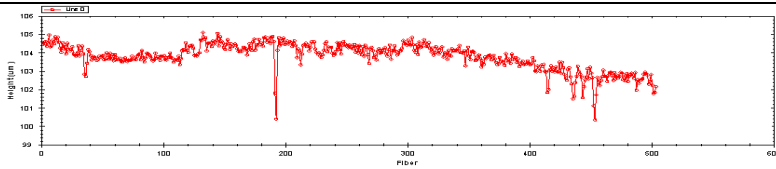
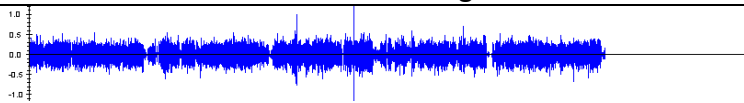
If the **Grv** checkbox is checked, the groove is added on the image. The entire detailed image can be saved in its mode (Raw, Flat, Sub-Flat...) by using the "Save Bitmap" button. The image is saved into the folder of the *.pri file.

8.1.3 AUDIO PANEL

The sound extracted from the file can be played into this panel. If the **Show Play Position on Image** is checked, the red square follows the light blue track on the raw image.

8.1.4 ANALYSIS PANEL

Different histograms can be shown on this panel. We can switch between the histograms by checking or not checking the checkboxes below the panel. ☒ Draw Points ☐ Vertical ☐ Sound

Everything unchecked	 <p>The vertical cut of the detailed image in the direction of the horizontal red line.</p>
Draw Points	Draw the acquired data points on the histogram.
Vertical	 <p>The vertical cut of the detailed image in the direction of the vertical red line.</p>
Sound	 <p>Representation of the extracted sound</p>

8.2 WORKING DESCRIPTION

Basically, the program follows sequentially the flow graph below.

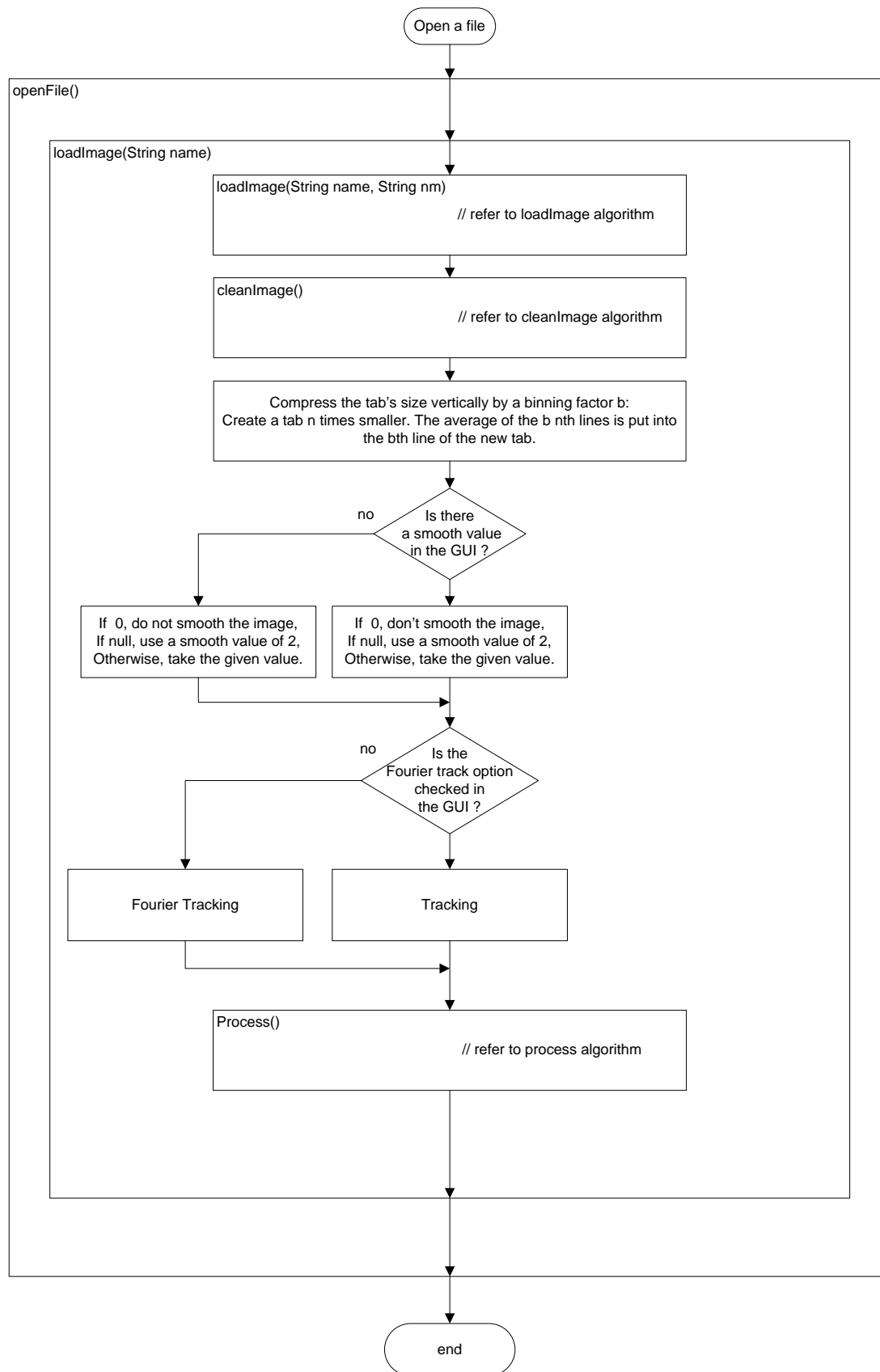


FIGURE 65: FLOW GRAPH OPEN A FILE

The program does four main tasks:

1. *Reads and orders the acquired data into a table (builds an image)*

The data can be acquired in many ways:

1. Standard acquisition, each set of data is unique.
2. Overlapped acquisition, each set of data is taken n times.
3. Interleaved acquisition, increase resolution (by 2 in the picture below).

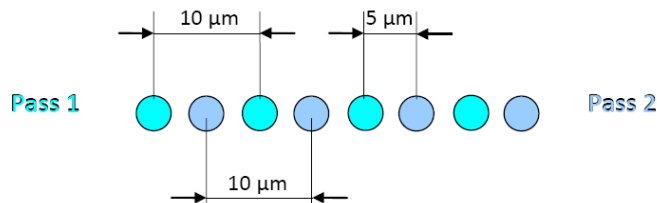


FIGURE 66: INTERLEAVED ACQUISITION

4. One fiber acquisition (CHR mode): instead of using 180 fibers, use just one.

2. *Cleans and filters the image*

Most of the cleaning algorithms are developed for cylinders (flat algorithm, blob detections, etc.), the only one useful for discs are smoothing, “clean H2” (horizontal cleaning) and “clean V2” (vertical cleaning).

In the next chapter we will explain how “clean H2” works. The other one is available in the appendix.

3. *Track the sound*

To reconstruct the sound (from cylinders or discs), we need to determine the middle of the grooves. In PRISM, the groove’s center research is done in two steps: tracking and processing.

The tracking researches the bottom of the grooves. It operates on binned data. This means it looks for the groove’s bottom on only each n line (where n : binning factor). The gain of time is very significant because we do not go through the whole image.

4. *Process*

From the tracking’s information, the processing finds precisely the middle (discs) or the depth (cylinders) of the grooves. As we know approximately where the centers are, the computational time is not long. Various processing algorithms exist on PRISM; they will be described in the next chapter.

8.3 ALGORITHMS

PRISM has many algorithms to treat the four main tasks. Below we show only the main algorithms. The other ones are noted in the appendix.

8.3.1 LOAD THE FILE

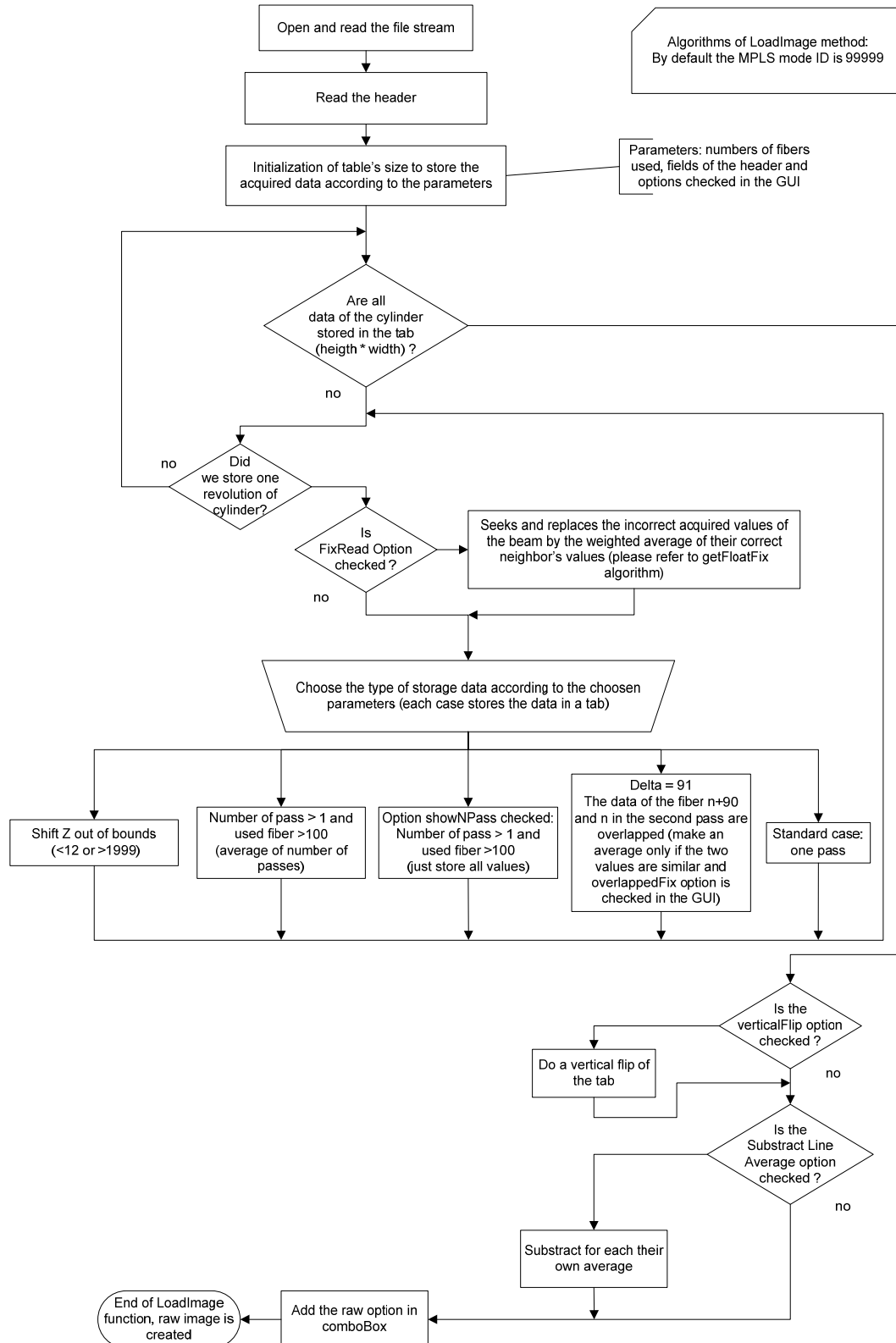


FIGURE 67: FLOW GRAPH LOAD *.PRI FILE

8.3.2 CLEAN

This algorithm is able to detect and correct bad values from the raw image. It reads the image line per line.

1. Firstly seeks in the image all points out of the tolerance maxDiff .
2. Secondly corrects all bad points.

The algorithm in the `getFloatFix` method uses this algorithm to detect incorrect value of the points. The `cleanV2` algorithm uses the same process but reads and makes corrections column per column.

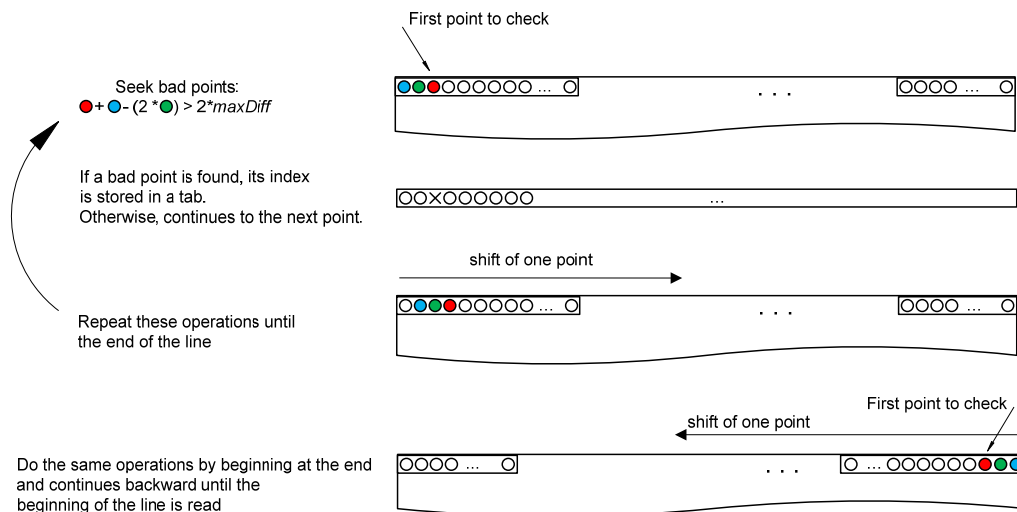


FIGURE 68: DETECT BAD POINTS

Now all indexes of the row's bad points are stored in an array. To correct these bad points, a weighted average is made using its good neighbors.

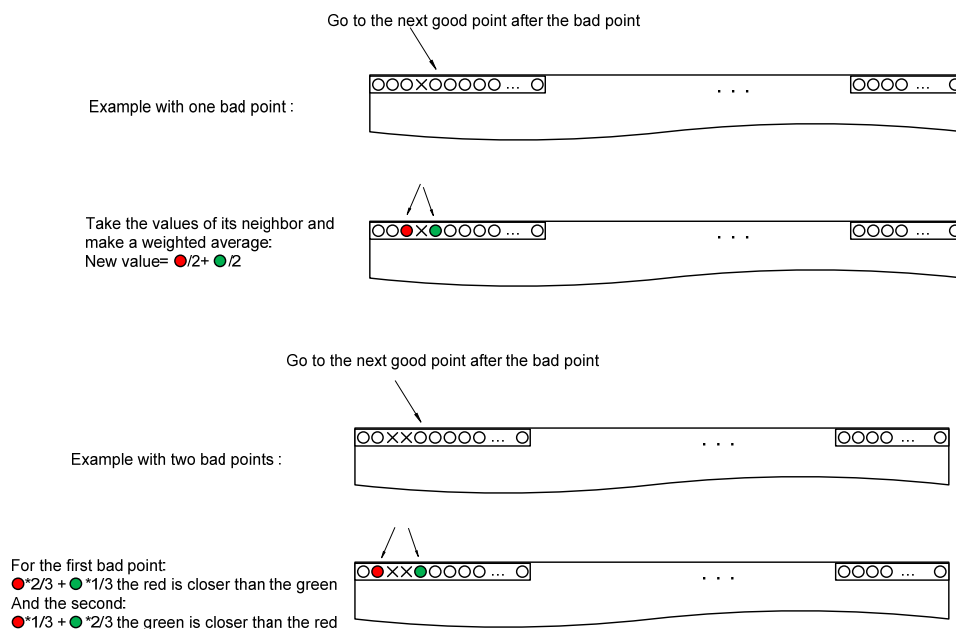


FIGURE 69: INTERPOLATE BAD POINTS

With more consecutive bad points, the same process is used.

8.3.3 TRACK

Step by step; here is how PRISM 2005 tracks.

Original image without tracking

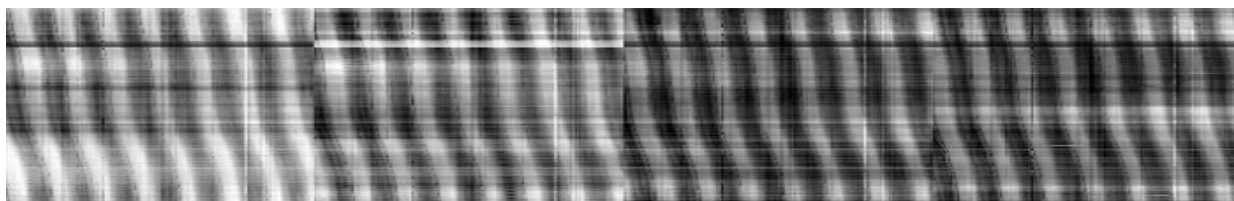
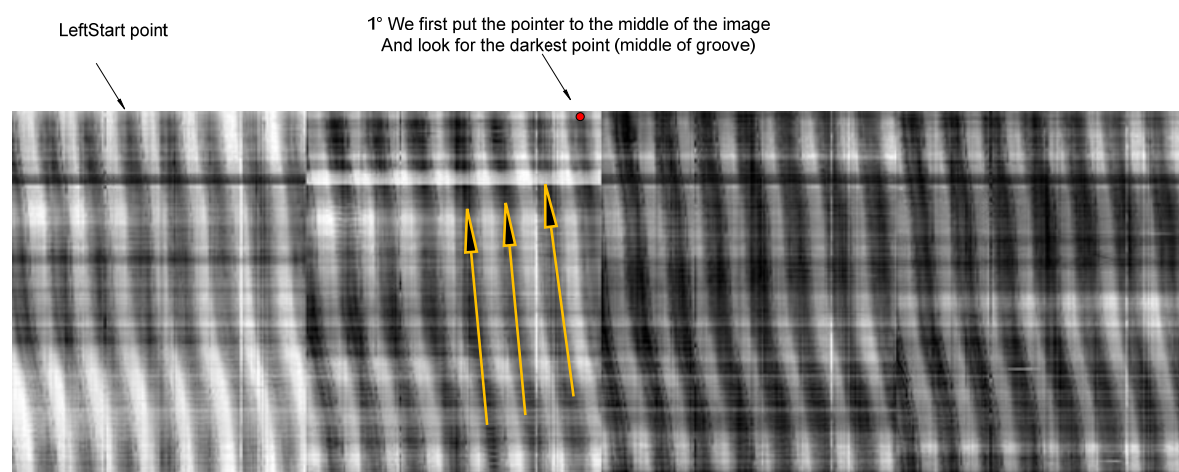


FIGURE 70: BINNED IMAGE

1. Find starting point (backtracking)

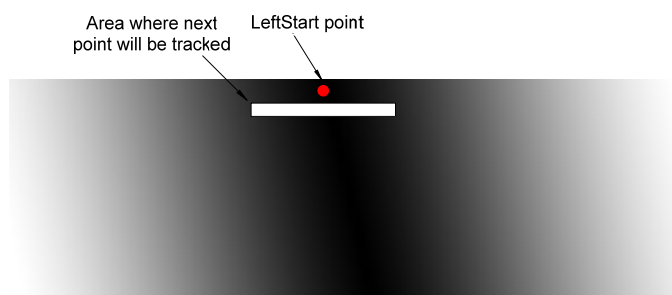


2° Then go back through the groove until reaching the LeftStart point.
Always follow the darkest points.
If the LeftStart is 0, reach the start of the image

FIGURE 71: BACKTRACKING

This technique makes it possible to detect when the groove goes out of the image and to stop beforehand.

2. Track



The points discovered by tracking are stored in a tab as shown below

table of groove points	y pos	x pos
1st point		
2nd point		
.		
.		
.		
nth point		

FIGURE 72: TRACKING DETAIL

To find the next point of the groove:

1° go to next line below and begin on the x position of the last groove point found

2° take an area that extends over the groove

3° Look for the deepest point.

4° Continue 1. 2. 3. points to the bottom of the image.

5° Continue 1. 2. 3. 4. points across the whole image.

3. *Raw track done*

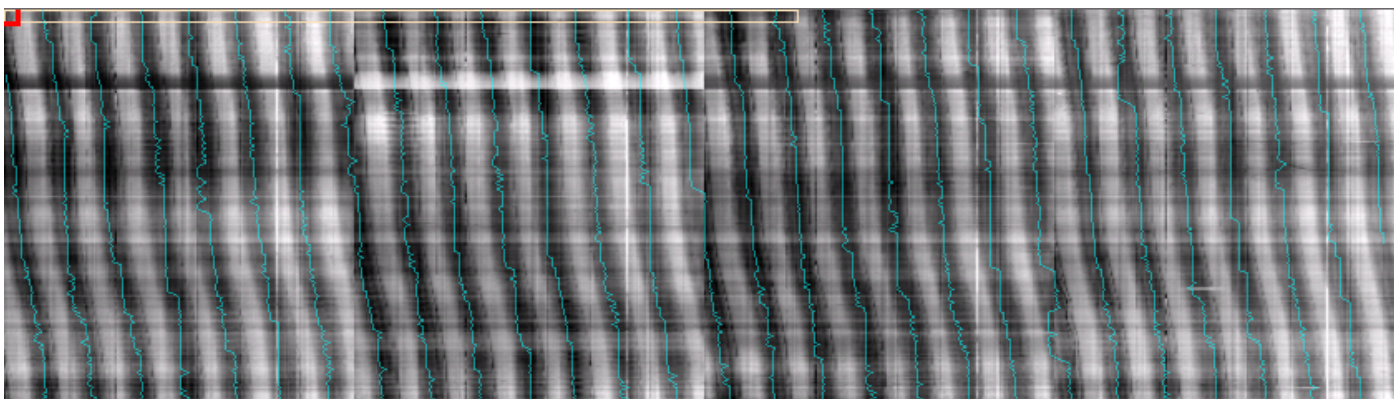


FIGURE 73: RAW TRACKING

4. *Fix the tracking*

The blue line represents the tracking without the « Fix » checked

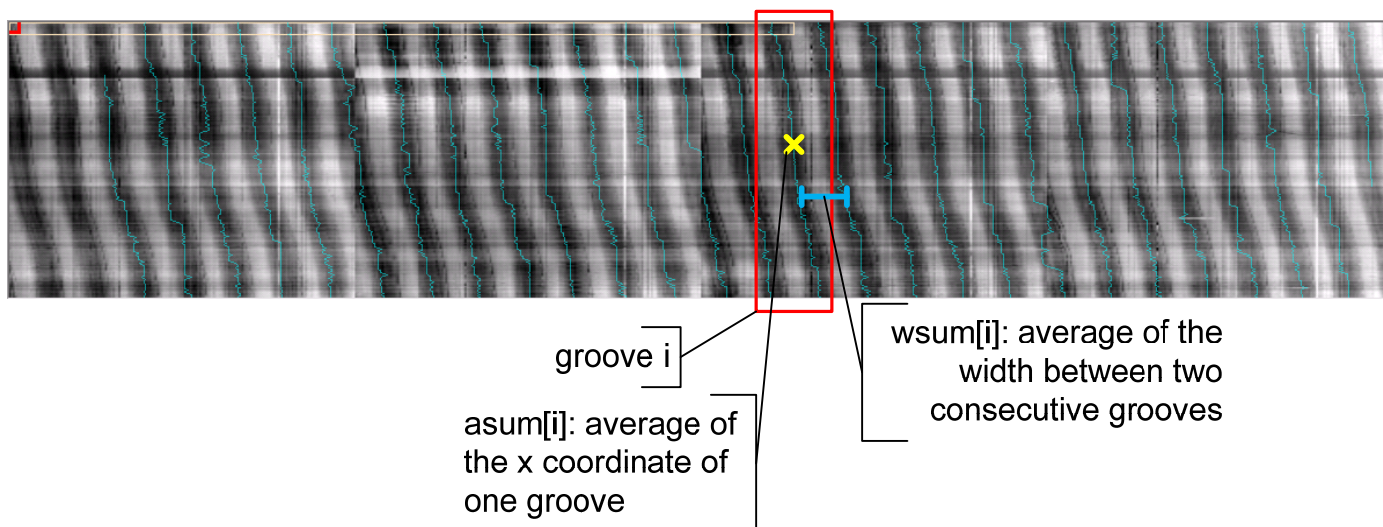


FIGURE 74: FIX THE TRACKING

- a. Build an average profile from the good points of each profile

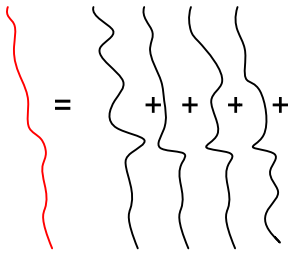


FIGURE 75: BUILD AN AVERAGE PROFILE

- b. By interpolation, find a straight that best fits the average of the each groove.

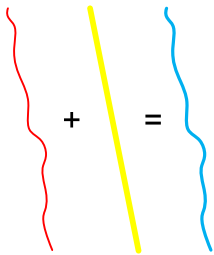
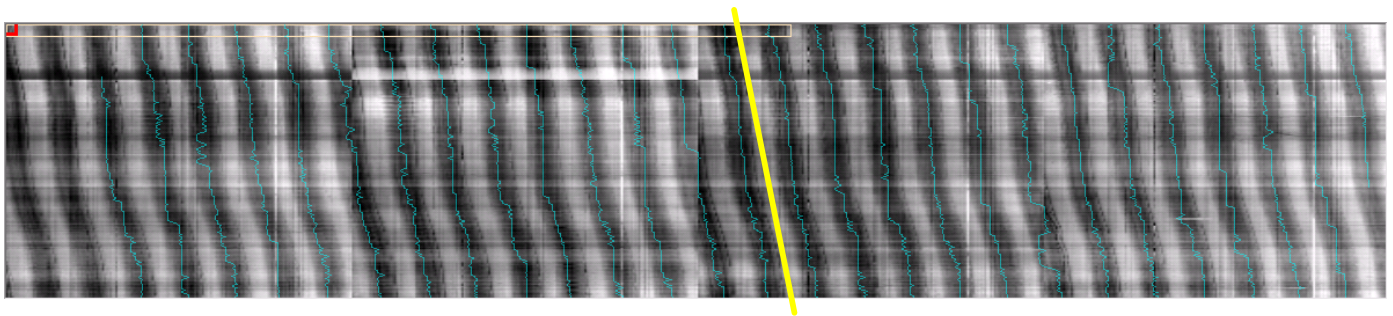


FIGURE 76: FIND AN AVERAGE TRACKING

- c. place the straight at the right place and add the profile to it.

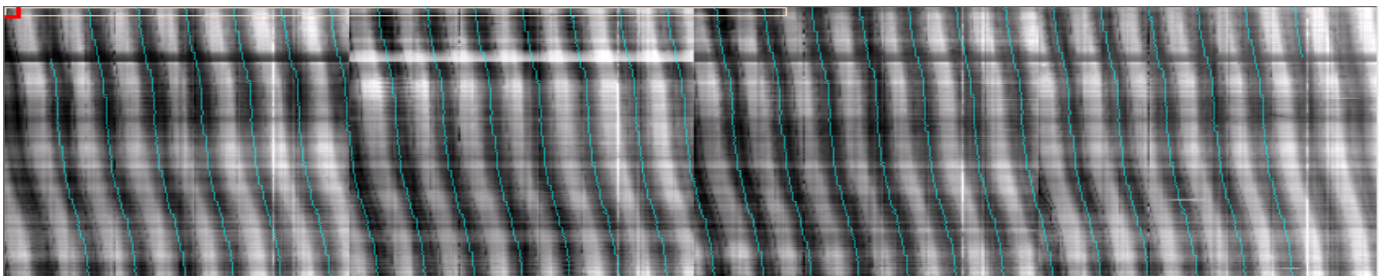


FIGURE 77: TRACKING RESULT

8.3.4 PROCESS

The tracking was done on a binned image. By this means, we will follow the groove of the whole image.

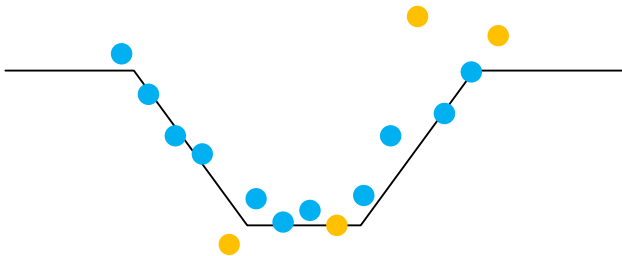
- Take the next track point.
- Seek the groove's center in the following n lines (n is the bin factor).

Complete these two tasks until the end of the non-binned image.

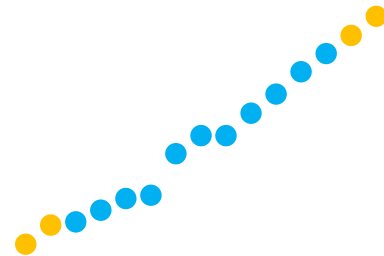
This time the cylinders and the discs are treated independently. Indeed, the sound information in cylinders stays in the depth difference, and the one in discs stay in the horizontal displacement, so we need to determine exactly the groove's center.

8.3.4.1 CYLINDER

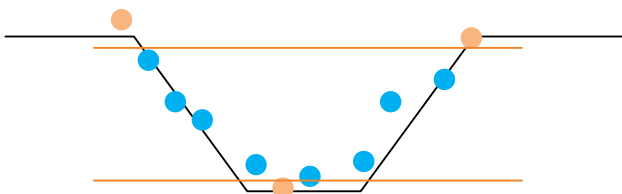
1. all the tracked points of the profile



2. sorting and discarding the extreme points



3. top and bottom limits



4. calculate the average depth with the rest of the points

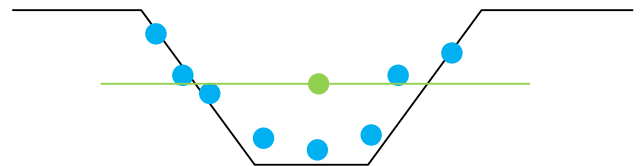


FIGURE 78: CYLINDER PROCESSING METHOD

8.3.4.2 Disc

The two first methods are the two better ones, giving the best sound and the best SNR.

Disc Fit method

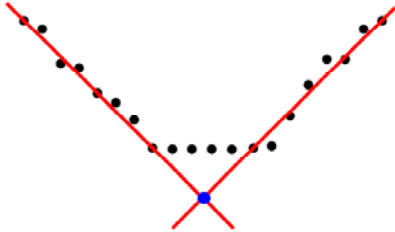


FIGURE 79: FIT

Realize a polynomial interpolation of the two slopes.

The intersection of these straights gives the position of the groove's center.

Disc Fit-line method

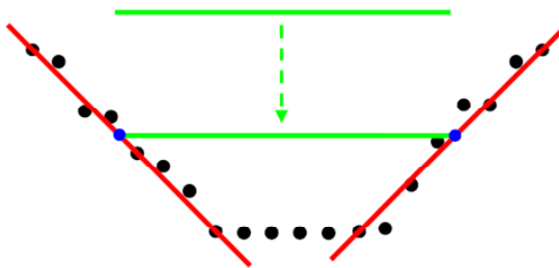


FIGURE 80: FIT-LINE

Drop a horizontal line into the groove.

Once both these ends intersect with the linear functions of the slopes, register the left and right intersection.

The average between the two points gives the groove's center.

Disc Quad method:

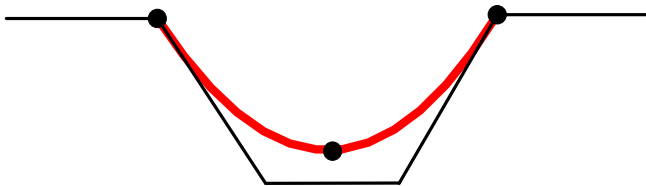


FIGURE 81: QUAD

Trade the two points at the beginning of the slope and compute a quadratic function where the peak of this one gives the groove's center. This one does not work well.

The last method "Disc LR" is used with only one fiber and takes the data with a tilted probe. The two slopes are acquired and the groove's center is computed by an average of the two profiles.

This analysis was done at the beginning of the project, but many improvements in the GUI and the Disc Fit method were made subsequently.

9. CUTS ON 3DSEVEN

Our predecessors Sven Hezel and Severin Nowak developed some cut algorithms implemented in their program 3DSEVEN. First, we have to study the efficiency of their cuts on the sound quality and the signal-to-noise ratio (SNR) of some discs.

9.1 HISTOGRAMS AND CUTS

Cuts play a very important role in the sound reconstruction process. Some are applied at the beginning, when we take the raw data and we look for the value of anomalous points. When we search all grooves' centers, the values used to compute them are not always good, and we need to detect and fix them. By applying good cuts in the best places, we can decrease the noise level and increase the sound level.

The cuts are based on histograms. The histograms illustrate a statistical distribution of the measured data. On these, we can apply cuts and reject the values outside the limits.

For example below, we check on the histogram all values in the Gaussian distribution section. The rest is the bad part. Based on this information, we can detect the bad data and correct or drop it.

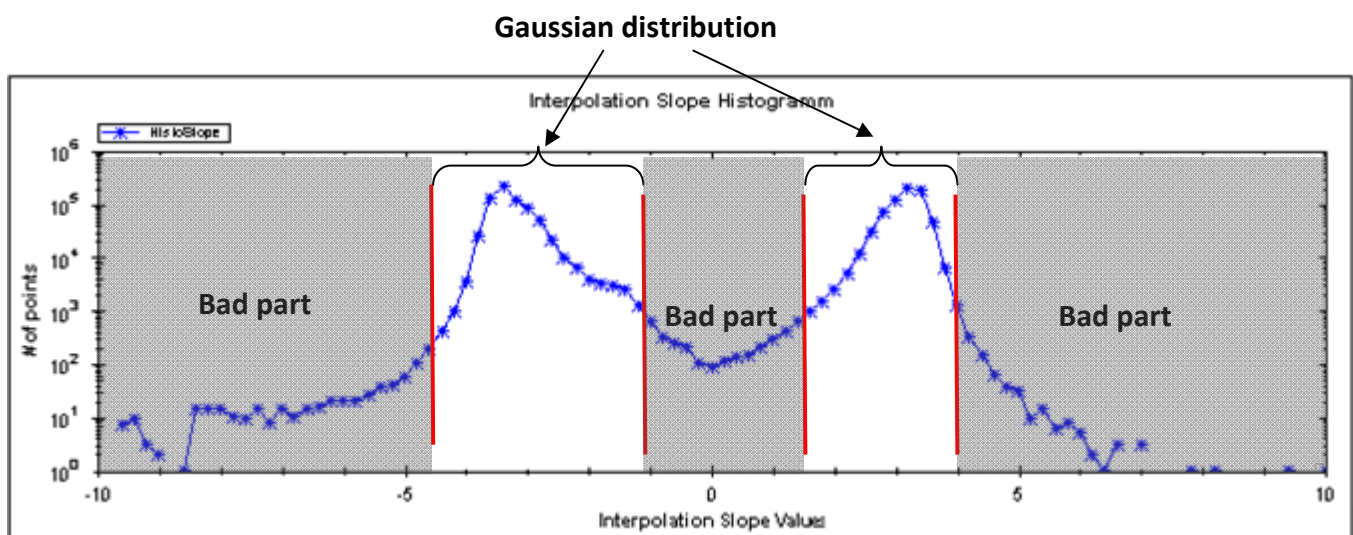


FIGURE 82: HISTOGRAM AND CUT EXAMPLE ON 3DSEVEN

One way to realize cuts is to make an interpolation between two good nearby points. However, if too many points are not available, many bad points will be detected, and enormous interpolations must be made. This will degrade the sound information.

Many histograms and cuts are possible to improve sound reconstruction algorithms.

9.2 STUDY OF 3DSEVEN CUTS

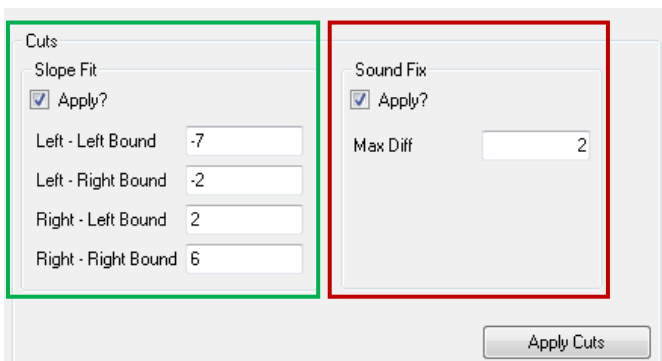


FIGURE 83: CUTS; GUI 3DSEVEN

- **Green cuts:** define the limits of useful data. (Refer to previous sub chapter).
- **Red cuts:** define the maximum difference between two consecutive groove centers. The main effect of this cut is to increase or decrease the volume.

$\text{MaxDiff} < 1$, increase the difference between two grooves. Useful if the sound is too low.

i.e. $\text{MaxDiff}=0.5$ a low factor can amplify the sound too.

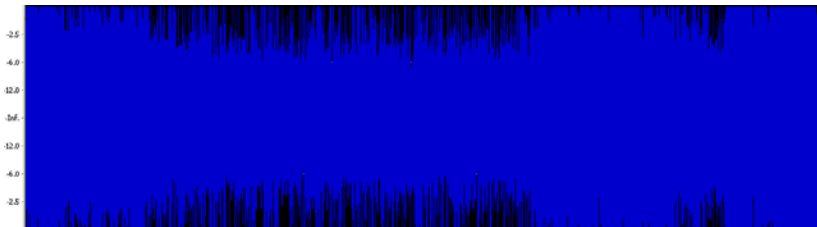


FIGURE 84: SOUND TOO LOUD

$\text{MaxDiff} > 2$, decrease the difference between two grooves. Useful if the sound is too loud.

i.e. $\text{MaxDiff}=6$ a large factor can reduce the sound too.

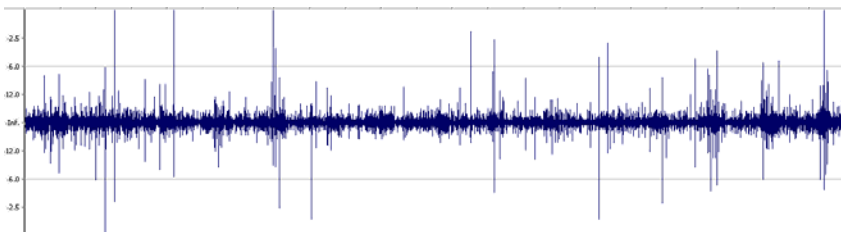


FIGURE 85: SOUND TOO LOW

9.3 MEASURES

We are going to show the measurements of two discs. The first (test disc) has only sinusoids engraved. The second one is a music disc (“Moonlight”) with many scratches and much dust.

On the test disc’s measure we can compute the SNR. We can easily measure the level of the fundamental frequency. The noise is measured in a silence in the wave, indeed a silence in the test disc means “without engraved sound information.”

On a music disc, it is impossible to compute a true SNR, because we cannot measure the noise level alone as is possible in test sound. In music, all frequencies have sound information.

9.3.1 TEST DISC

Two files are tested, the first with interleaved acquisition and the second without interleaved acquisition.

9.3.1.1 INTERLEAVED ACQUISITION: DISC-TESTSOUND-G7_10_907_6_40000_400.PRI

Sound spectrum without cuts

Sound spectrum with slope’s cuts

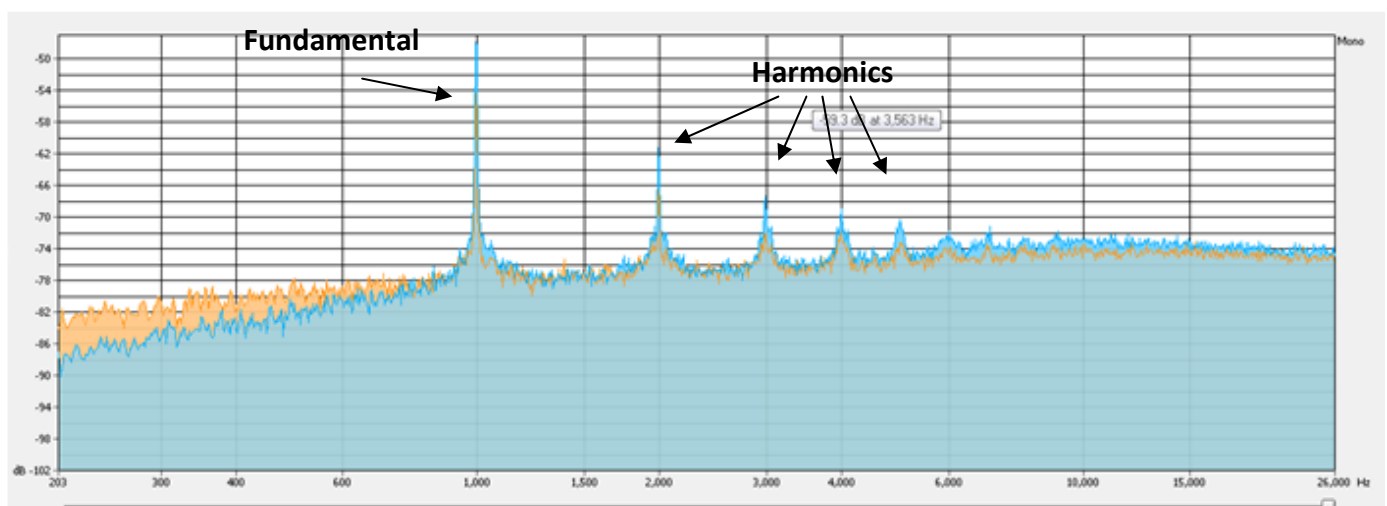


FIGURE 86: CUTS ON A TEST TONE

By applying cuts, the spectrum becomes flatter, and noise increases at low frequencies and decreases at high frequencies.

Noise measurement: The dark highlighted part is a silence.

Noise spectrum without cuts

Noise spectrum with slope's cuts

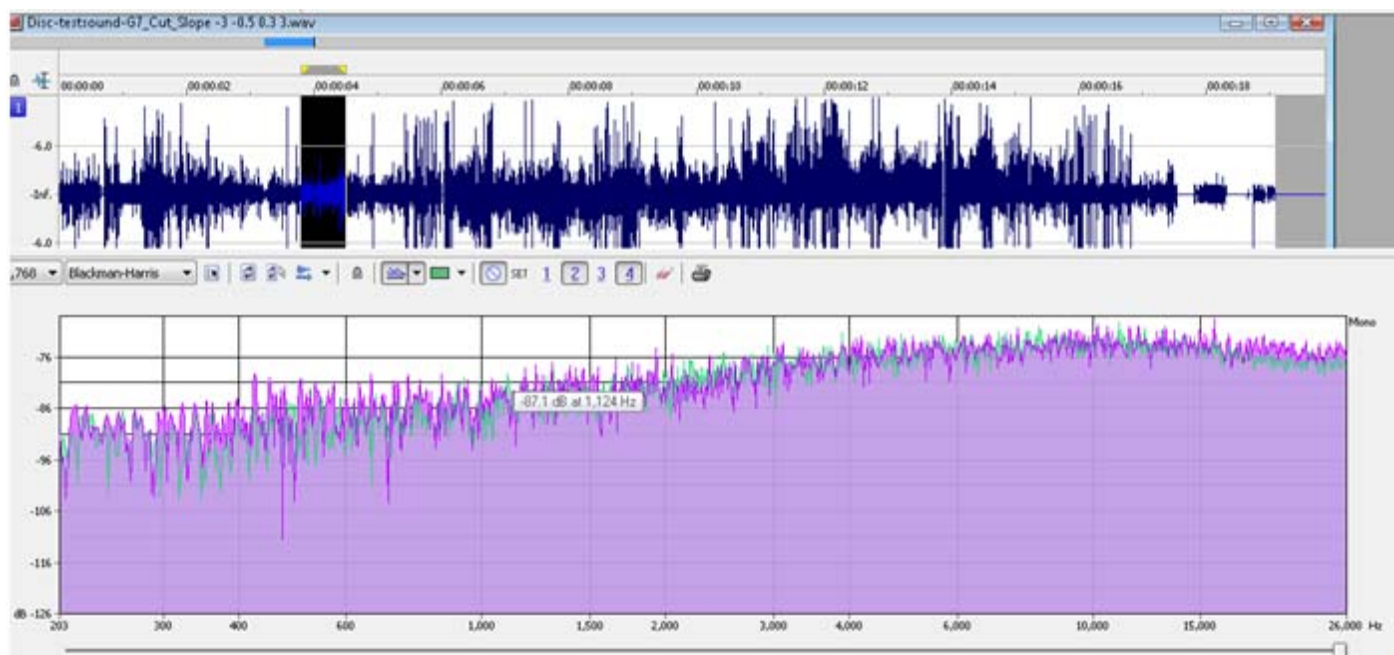


FIGURE 87: NOISE MEASUREMENT ON A SILENCE

Parameters	Peak [dB]	Noise at 1kHz [dB]	SNR [dB]
Without cuts	-47	-87	40
Slope fix: LL: -3 LR: -0.5 RL: 0.5 RR: 3	-54	-88	34

The SNR on the fundamental frequency is more important when we are applying no cuts.

9.3.1.2 STANDARD ACQUISITION (ONE PASS): DISC-TESTSOUND-G6_10_1814_3_40000_400.PRI

As the sound volume was too low, we increased the volume of the sound fix parameter. Below, we show the comparison before and after this parameter was changed.

Sound spectrum without cuts

Sound spectrum with sound's cuts

By comparison with the previous measure (9.3.1.1), more parasite frequencies appear. All the waves analyzed in the current chapter were acquired by Hezel and Nowak, and we do not know their measure's conditions.

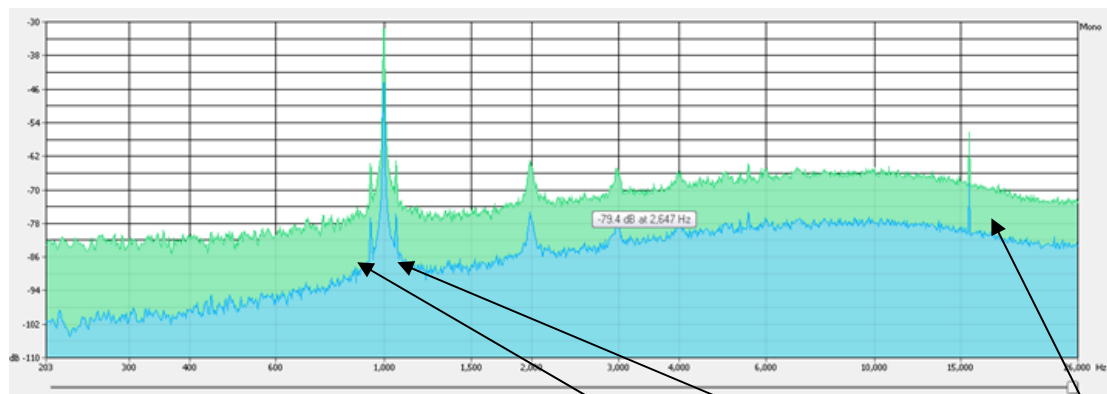


FIGURE 88: SOUND CUTS ON A TEST TONE WITHOUT INTERLEAVING

Inter-modulation between the fundamental and the line frequency: 1 KHz - 60 Hz and 1 KHz + 60 Hz

Horizontal frequency of standard television (cathodic): 15'625 Hz

Noise spectrum without cuts

Noise spectrum with sound's cuts

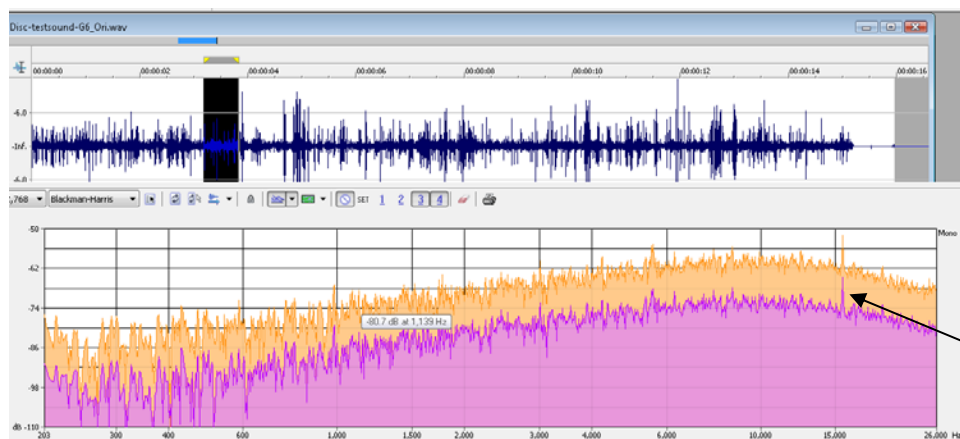


FIGURE 89: NOISE MEASUREMENT WITHOUT INTERLEAVING

This frequency is present everywhere in the .wav file

Parameters	Peak [dB]	Noise at 1kHz [dB]	SNR [dB]
Without cuts	-44	-87	43
With cut: sound fix at 2	-31	-75	44

The SNR is a bit better 1dB, but the noise is still audible.

Next, we try to apply cuts on the interpolated slope's histograms:

Sound spectrum without cuts

Sound spectrum with sound's cuts

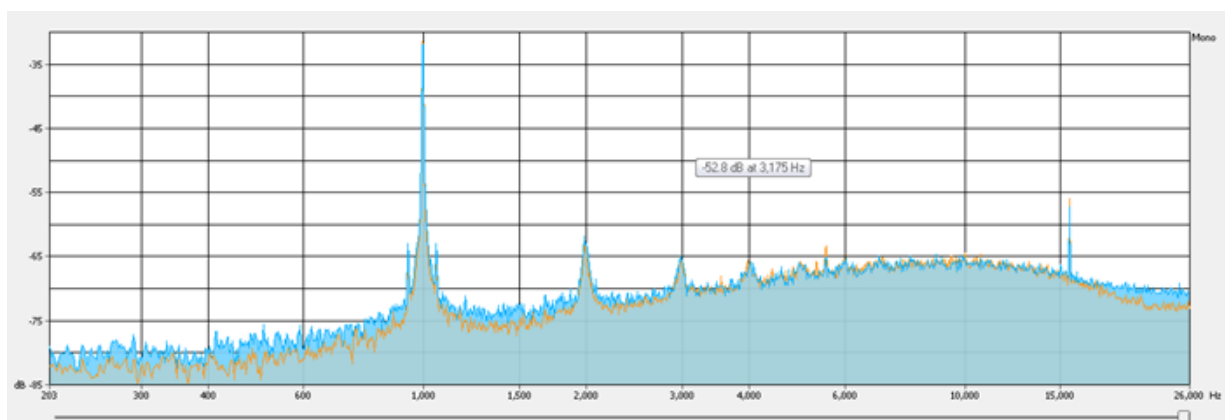


FIGURE 90: SPECTRUM WITH THE ADDITION OF SLOPE CUTS

We observe that the global noise level increases on the whole spectrum. But if we see on the noise's measure below (during a silence), the **noise level with cuts** (in blue) is a little lower than the **noise level without cuts** (in violet).

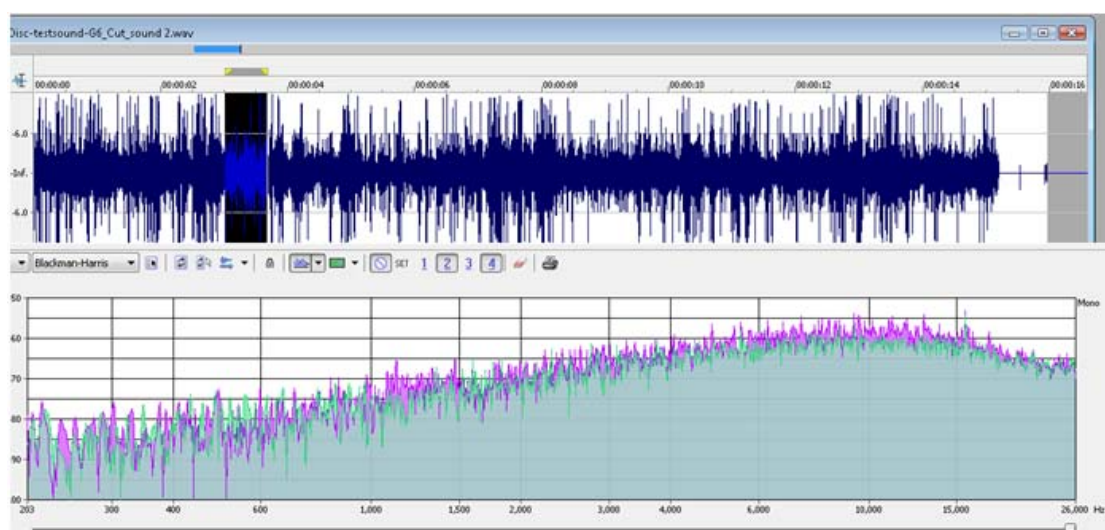


FIGURE 91: SILENCE MEASUREMENT

Parameters	Peak [dB]	Noise at 1kHz [dB]	SNR [dB]
Sound cut at 2	-31	-76	45
Sound cut at 2 + Slope fix: LL: -5.5 LR: -1.2 RL: 1.2 RR: 5	-31	-78	47

At the fundamental frequency, the SNR is better, but on the whole range the noise level is higher, which is not satisfactory.

9.3.2 MUSIC DISC: DISC-MOON-G1_5_907_4_40000_400.PRI

Sound spectrum without cuts

Sound spectrum with slope's cuts

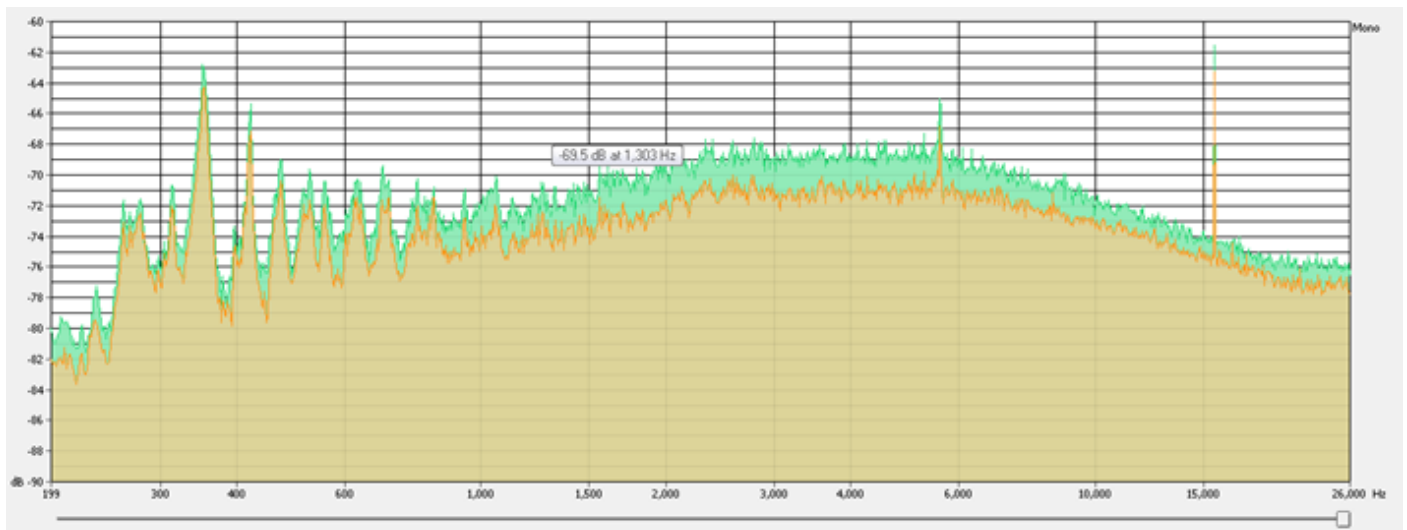


FIGURE 92: CUTS ON A RECORDED DISC

With recorded discs, the result is exactly inverted, by a difference of 2dB. This measure demonstrates that the cuts developed in 3DSEVEN work on music. But the result is still noisy; lots of other cuts or improvements have to be brought in to obtain an excellent output wave.

10. COMPARISON PRISM—3DSEVEN

Actually for four algorithms work to treat matte discs numerically: one implemented in 3DSEVEN and three others in PRISM:

- Method for reading cylinders applied on discs (cylinder for disc method)
- Method to cross two straight -> the intersection gives the position of the groove's center. (Disc Fit method)

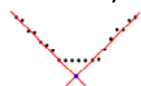


FIGURE 93: DISC FIT METHOD

- Method by dropping a horizontal line into the groove and the average between the two intersection's point gives the groove's center (Disc Fit-line method)

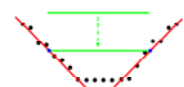


FIGURE 94: DISC FIT-LINE METHOD

We will compare them by using the test sound disc and a scratched music disc.

10.1 TEST DISC: DISC-TESTSOUND-G7_10_907_6_40000_400.PRI

As noted in sections 9.3.1.1 and 9.3.1.2, we calculate the signal-to-noise ratio by measuring the fundamental's peak and the noise level during a silence.

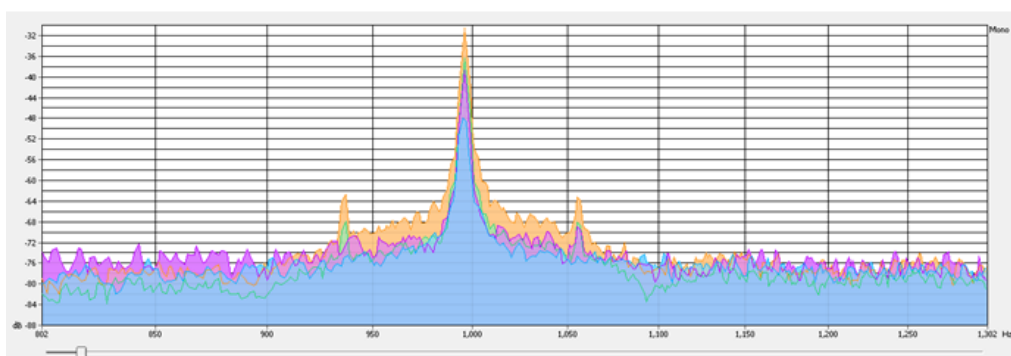


FIGURE 95: COMPARISON OF ALL DISC METHODS ON A TEST TONE

	Parameters	Peak [dB]	Noise [dB]	SNR [dB]
	3DSEVEN	-47	-87	40
	PRISM Cylinder method	-38	-78	40
	PRISM Disc fit	-36	-85	49
	PRISM Disc fit-line	-30	-82	52

Clearly, we see that "Disc fit" and "Disc fit-line" are the best. On the three PRISM methods we distinguish the intermodulation at 1 kHz \pm 60Hz. In the 3DSEVEN one, we believe that they are covered by the noise.

10.2 MUSIC DISC: DISC-MOON-G1_5_907_4_40000_400.PRI

To listen, the four results are similar. The 3DSEVEN's one with cuts seems to be the one with least noise.

It is interesting to analyze each spectrum separately:

This is the wave's spectrum obtained with 3DSEVEN. Each peak is very sharp and distinct. One defect is the parasite component at 15'625 Hz, which disrupts the sound.

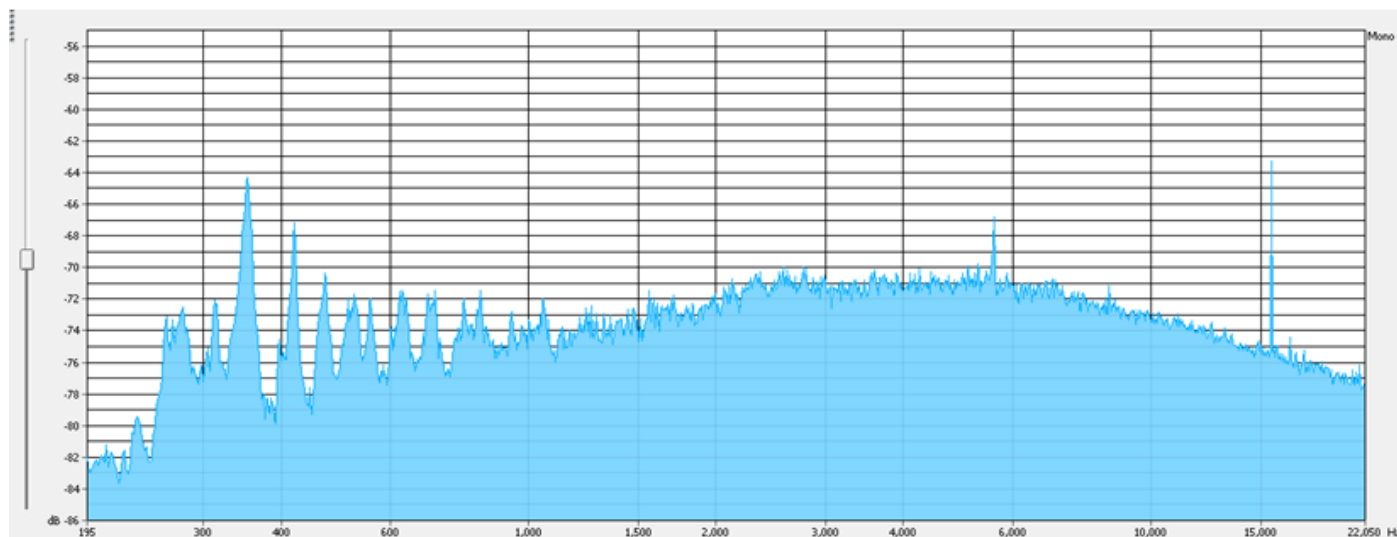


FIGURE 96: MUSIC RECORDING, 3DSEVEN SPECTRUM

Disc Fit method of PRISM. The global noise level is higher than the previous and the peaks are less distinct. However, this method is the only one that suppresses the parasite component of 15'625 Hz.

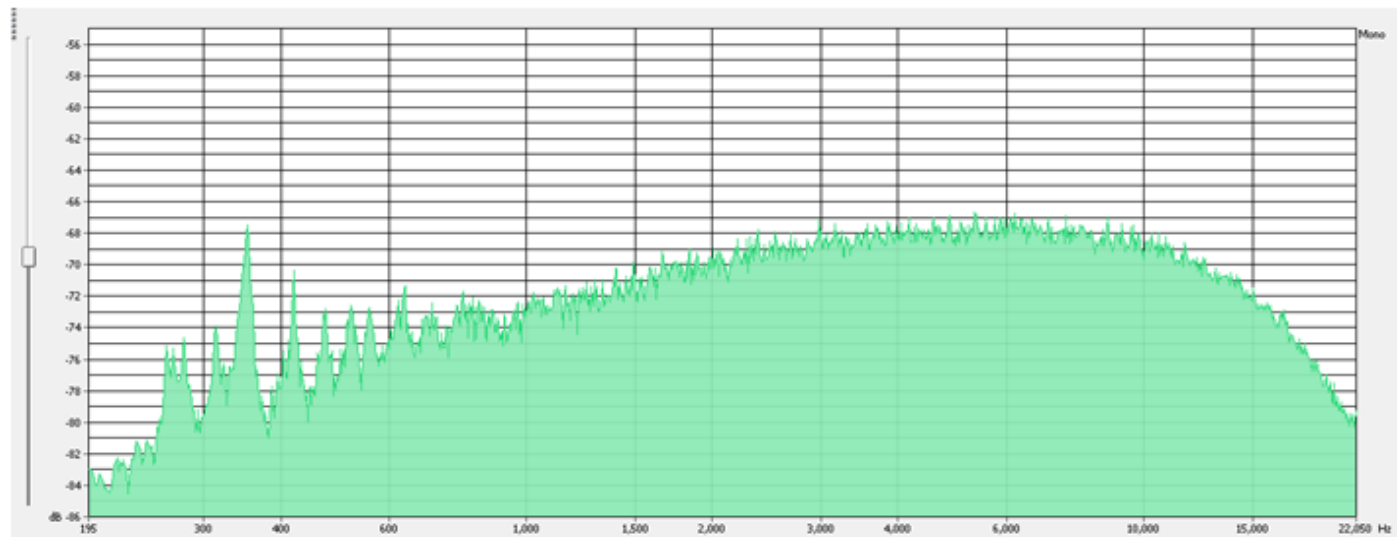


FIGURE 97: MUSIC RECORDING, PRISM, FIT METHOD

Disc Fit-line method of PRISM. The peaks are bigger than the Disc fit ones. The sound is a bit better. Unfortunately, the parasite component is present.

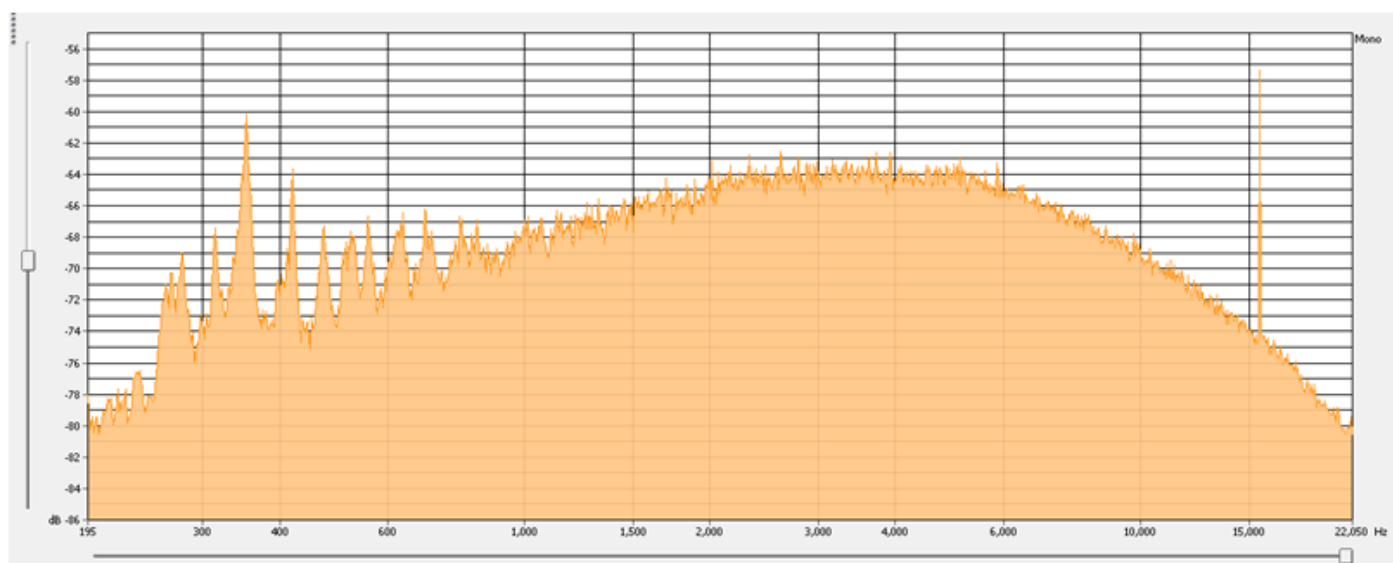


FIGURE 98: MUSIC RECORDING, PRISM, FIT-LINE METHOD

The last one is the cylinder for disc method. Here the spectrum is extremely disrupted, and the peaks over 1 kHz seem to be products of intermodulation with harmonics.

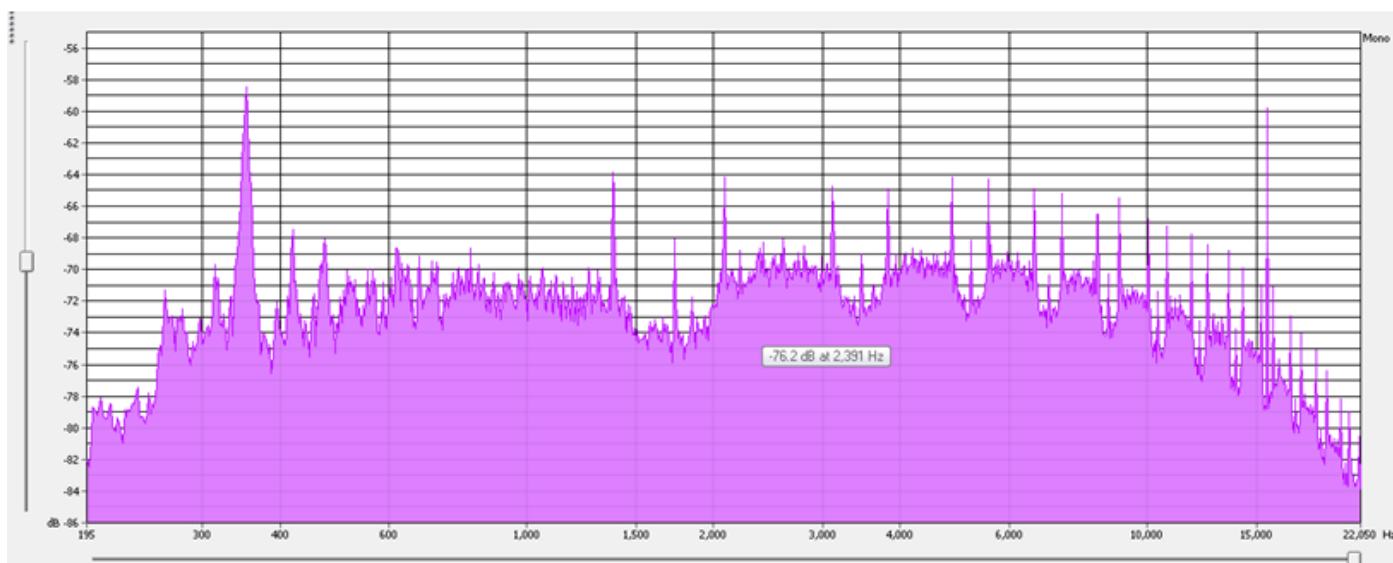


FIGURE 99: MUSIC RECORDING, PRISM, CYLINDER METHOD

These measures (11.2) show the usefulness of the cuts. 3DSEVEN is the only one to have this and gives better results in disc processing. Below a short analysis of this application is presented.

3DSEVEN advantages:

- Good slope detection for discs -> good tracking
- Possibility of applying the cuts

3DSEVEN disadvantages:

- Very lengthy period needed to treat *.pri file (15-20 minutes)
- Not robust (an inexperienced user frequently crashes the application!)

PRISM is the program on which we have to bring improvements; it is most developed, has far less bugs and can read cylinders. We will take the 3DSEVEN's advantages and implement them in PRISM.

11. IMPROVEMENTS ON PRISM 2005

One constraint of the project was to work on PRISM and add new functions. A powerful one is to see various statistics and distributions from computed parameters (slopes, inter-grooves space, interpolated slopes, sound difference...) and to apply cuts to it.

The previous chapter shows that the disc Fit method does not work well, so some modifications must be made.

Once the three main disc processing methods (cylinder for disc, disc fit and disc fit-line) work well, we have to run the three ones, compare sound quality/histograms and be able to mix them.

Follow the description of the code's migrations from sequential architecture into object-oriented architecture.

Finally, we will present the new Graphical User Interface (GUI) and will only explain the difference from PRISM 2005 and what the new functions bring to it.

11.1 HISTOGRAMS AND CUTS

We have added a new tab to display all histograms calculated at the various tasks of the program. Each histogram can be selected with the combo box at the bottom-left panel. Often the histograms resemble the example below: where the points are correct, they form a Gaussian distribution, beyond the points are considered bad and have to be corrected. To apply cuts on them, the user has to click twice to mark the good points and the bad points.

Once the bad points are identified, they are interpolated.

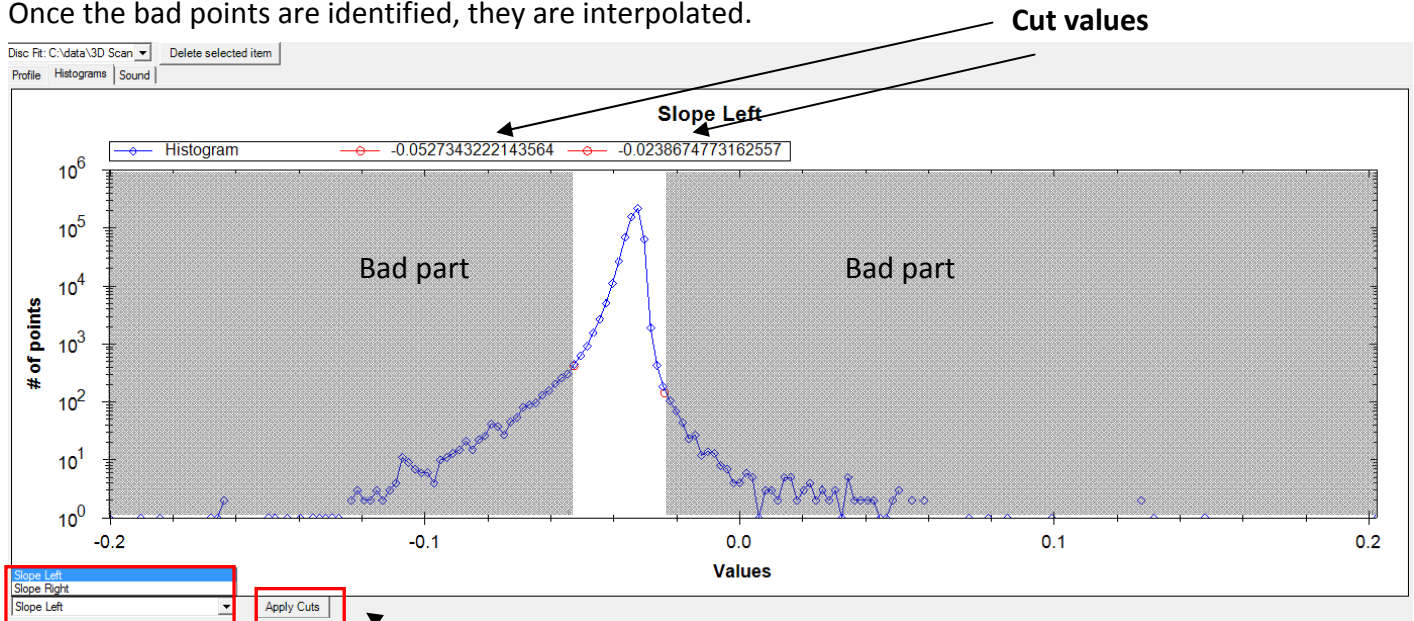


FIGURE 100: CUTS ON PRISM 2009

Combo Box contains all existing histograms

Once the cut values are defined, apply them by clicking on this button

11.1.1 FIRST CUT'S TEST WITH A MUSIC DISC: « MOON LIGHT »

The blue spectrum is the result with the “disc fit” method without cuts applied.

The green is the sound after cutting.

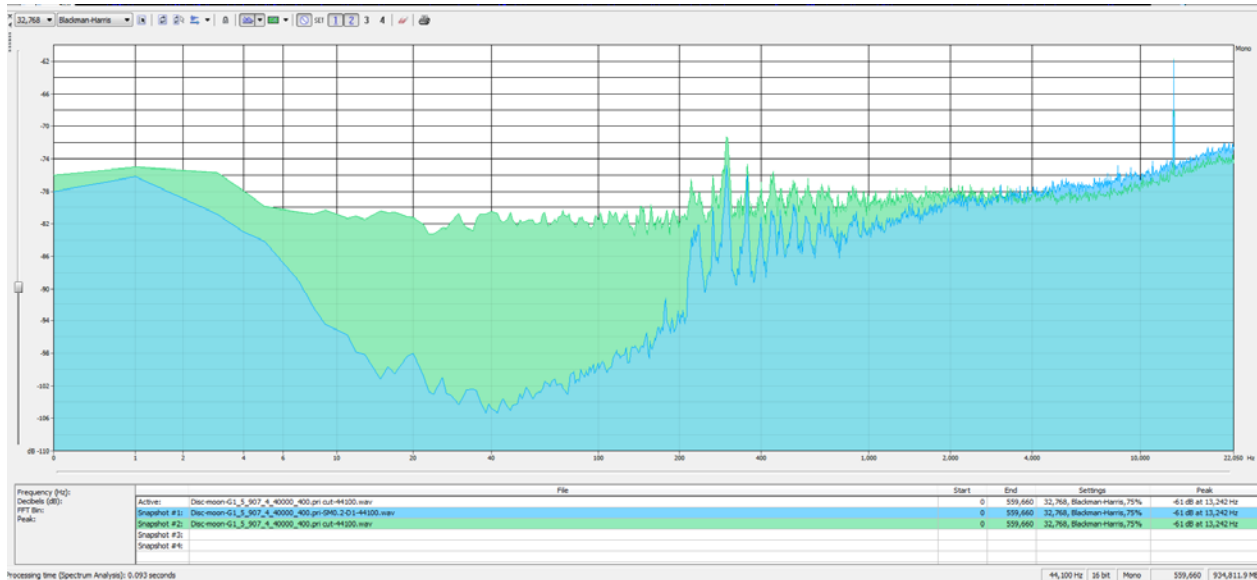


FIGURE 101: SPECTRUM COMPARISON WITH PRISM CUTS

These results are unexpected. Instead of having less noise overall, we see much more noise at low frequencies (0-1000Hz). At high frequencies, less noise is observed.

By hearing, the sound quality is affected by these cuts.

Below, we can see the sound visualization before and after cuts. The third one shows the difference between them.

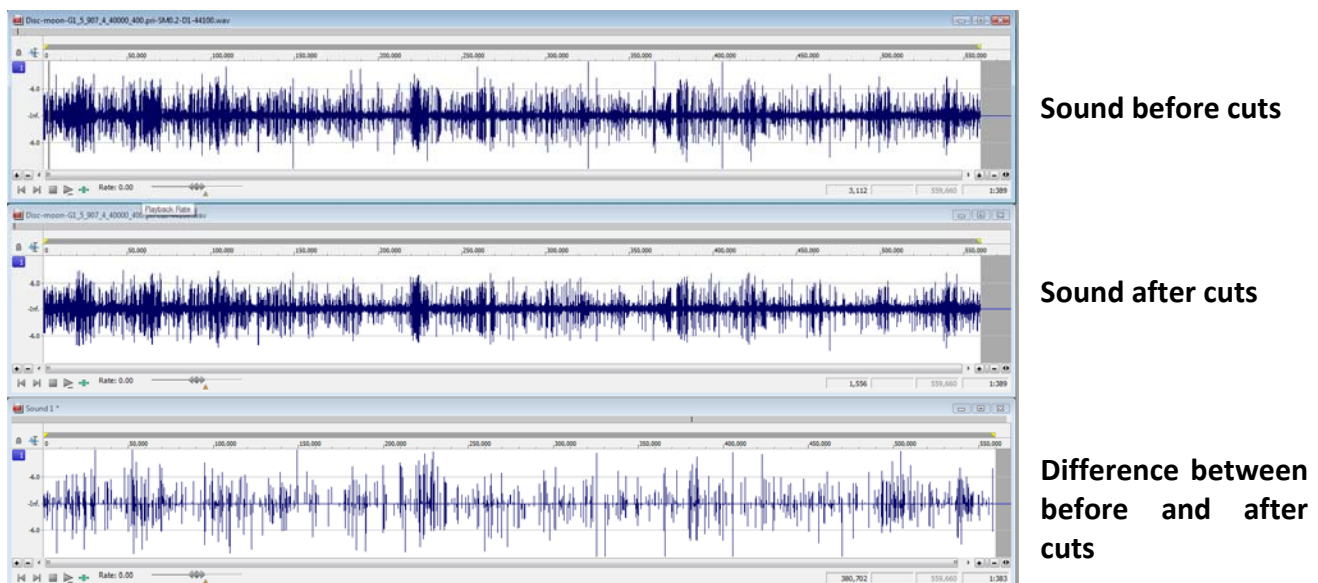


FIGURE 102: SOUND COMPARISON IN SONY SOUNDFORGE

A detailed view of a specific part of the extracted sound is represented below.

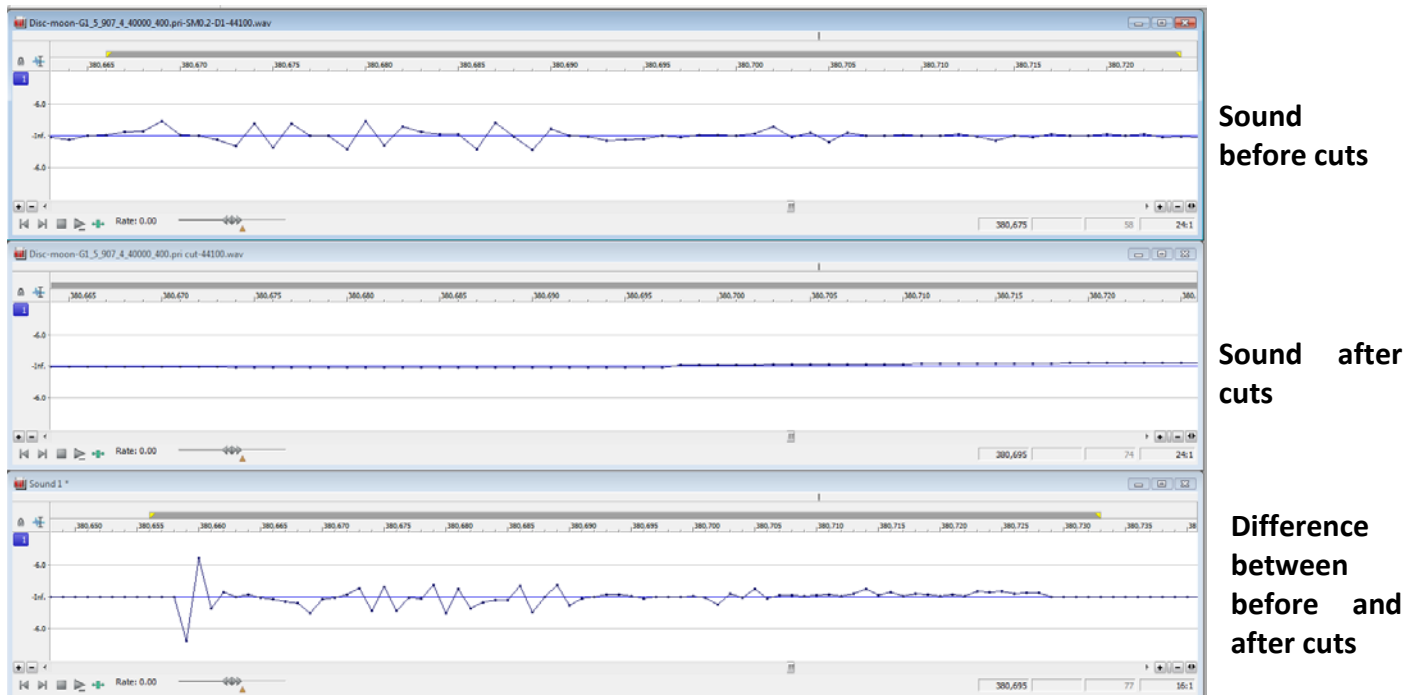


FIGURE 103: ZOOM IN A SPECIFIC PART OF THE SOUND WAVE

Sometimes, there is a long sequence of bad points. Those points will be interpolated. The result is a long straight replacing some useful information. This explains the addition of many low-frequency components.

It could be useful to display a statistic of the sequence of these bad points and in some cases, as shown above, not to interpolate them at all, to avoid this addition of noise. Only short sequences (less than 4-5 points) should be interpolated.

Some improvements, as shown above, will be added to the cut's algorithm.

11.2 IMPROVED DISC FIT METHOD

The disc fit method gave completely wrong results because of the bad points in the slopes. The method computes the straights with all the points of the slopes, which is why bad point detection is useful. Considering the number of bad points, the slope detection can be improved too.

11.2.1 DETECT BAD POINTS

Shown below is an example of incorrect computation of the groove's center. Many slope points have the wrong value, and the computation of the line does not work.

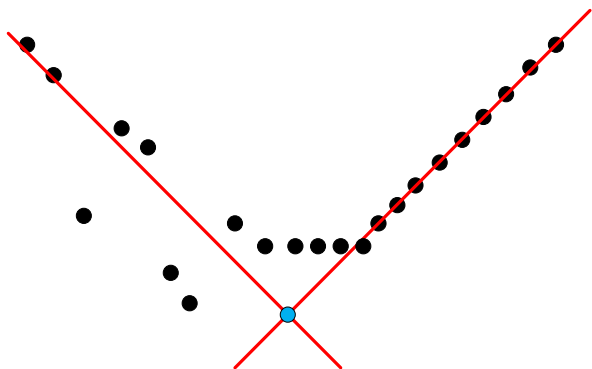


FIGURE 104: ERROR BECAUSE OF BAD POINTS

The first task is to compute a new line, ignoring the bad points. The bad points are detected as follows: the user enters a value that defines a threshold distance. Next, for each point, we compute the distance on the Y axis. If this distance is greater than the threshold the point is bad.

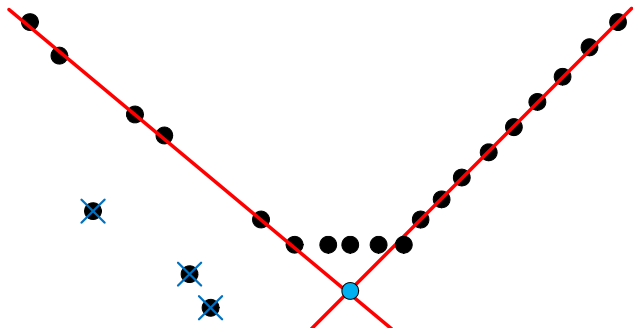


FIGURE 105: RESULT OF CORRECTING THIS ERROR

This method improves the sound quality, but each time the actual slope detection algorithm misses a slope, this algorithm is inefficient. A better slope detection is required.

11.2.2 NEW SLOPE DETECTION

The 3DSEVEN's slope detection algorithm seems to work better than the PRISM one. After the study of 3DSEVEN, we incorporated it into PRISM, but the result has not changed. In fact, the slope detection's algorithms of 3DSEVEN and PRISM are similar.

Detail of 3DSEVEN algorithm:

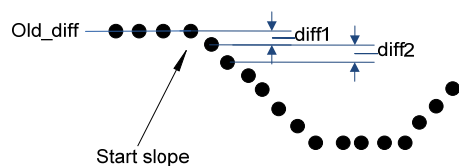


FIGURE 106: LOOKING FOR START SLOPE POINT

If $\text{diff1} \geq \text{CONST}$ && $\text{diff2} \geq \text{CONST}$ && $\text{old_diff} < \text{CONST}$
Then we find a start point slope
where $\text{CONST} \rightarrow$ constant value to detect slope.

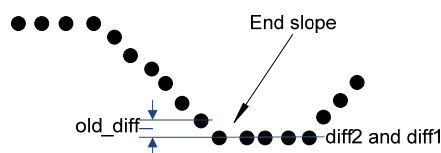


FIGURE 107: LOOKING FOR END SLOPE POINT

If $\text{diff1} < \text{CONST}$ && $\text{diff2} < \text{CONST}$ && $\text{old_diff} \geq \text{CONST}$
Then we find an end point slope.

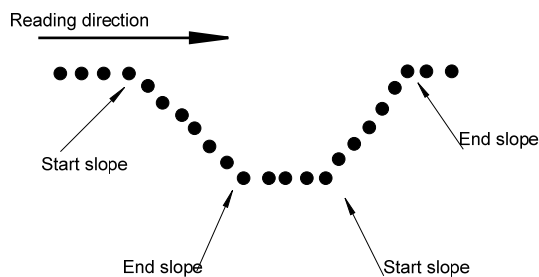


FIGURE 108: REGISTER THE FOUR POINTS

The position of the four points is saved

PRISM does not save the start and end, but when it detects a start, it registers the point's slope until the end is reached.

Our proposal: the sliding line method

In fact, we can know the mean slope's angle of the disc.

- First, read all the points of one line.

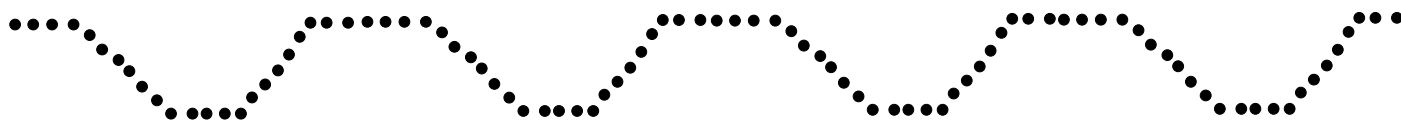


FIGURE 109: OVERVIEW OF A LINE OF DATA

- Create a histogram of these points, the result gives two peaks (one for the top and one for the bottom). These peaks represent the mean top and bottom values. The difference gives the mean depth.
- For each line of the image, we total the new mean depth.
- Then we divide the sum of the depths by the number of lines, and we have the mean depth of the whole acquired data.
- The user has to pass thru the GUI the width (um) of the current disc.
- By dividing the number of points per slope by the mean depth, we have the mean slope of the disc.

Then we slide a line at this tilt until reach the real slope and we store the line's coefficients. For finding the other groove's side, we invert the tilt.

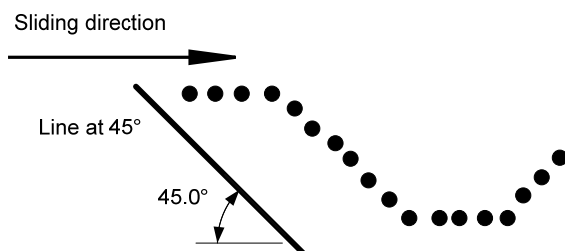


FIGURE 110: SLIDING LINE METHOD

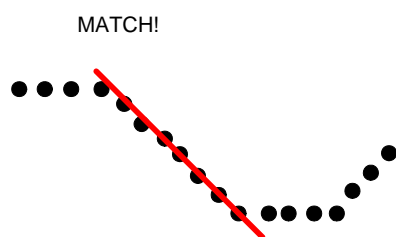


FIGURE 111: SLIDING LINE MATCH

This method replaces the actual PRISM slope detection and the bad point's detection described in Section 11.2.1. If we combine this one with bad point detection, the final result is worse.

In blue the sliding line method

In orange the combination of the two

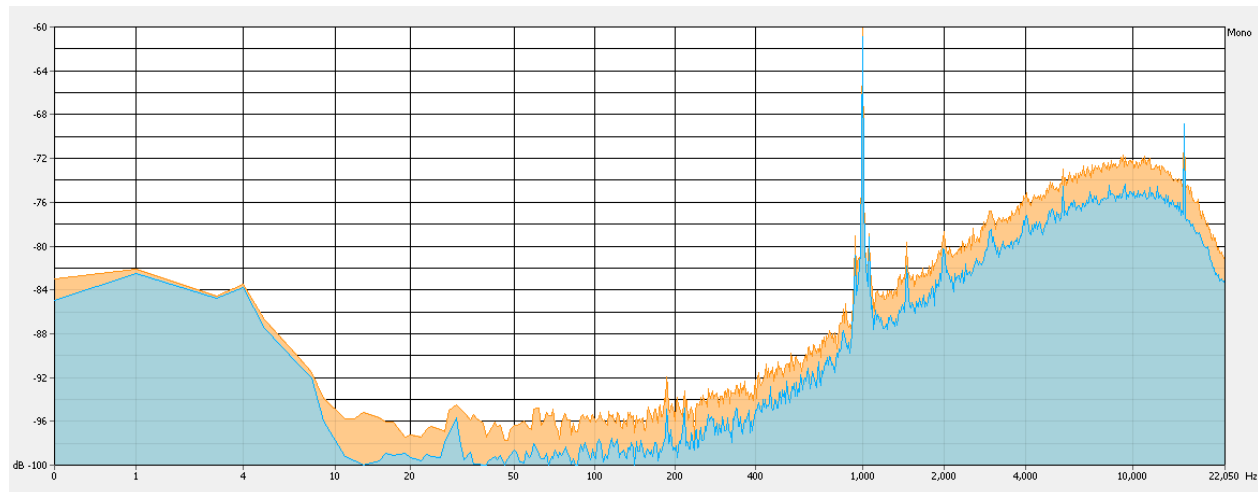


FIGURE 112: COMPARISON BEFORE AND AFTER SLIDING LINE METHOD

By using only the sliding line method, we realize a gain of 3dB.

11.3 ALTERNATIVE TRACKING ALGORITHM

The purpose of this new tracking is not to detect the lowest point of the groove but approximately the middle point of the groove. We kept the same method to find the starting point as the original track. We begin at the first line with the starting point.

11.3.1 BY FOLLOWING THE GROOVE'S PATH

Show in Section 11.2.2 we can detect the top and bottom level on each line of the image. Then we find a level exactly between the top and the bottom.

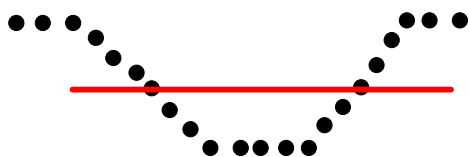


FIGURE 113: FIND A LEVEL BETWEEN THE TOP AND THE BOTTOM

We keep all the points below this level.

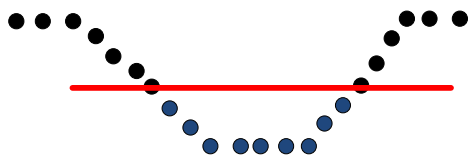


FIGURE 114: STORE ALL POINTS UNDER THE LINE

We take the first and the last blue point. The track point (yellow) is exactly at the middle of the first and last blue point.

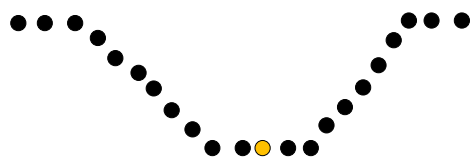


FIGURE 115: COMPUTE THE MIDDLE POINT

To find the next track point, jump to the next line and take as a reference the previous track point found.

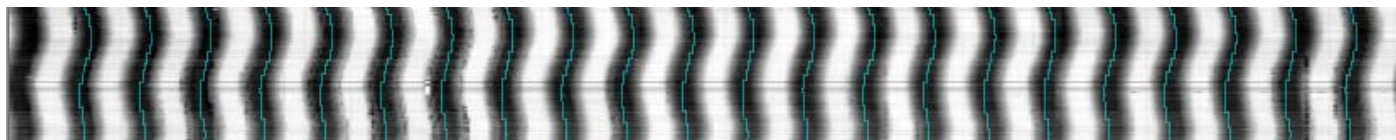


FIGURE 116: TRACKING RESULT

11.3.2 FIND EACH TRACK'S POINT INDEPENDENTLY

With the technique above, we can find all the track's points of one line at the same time.

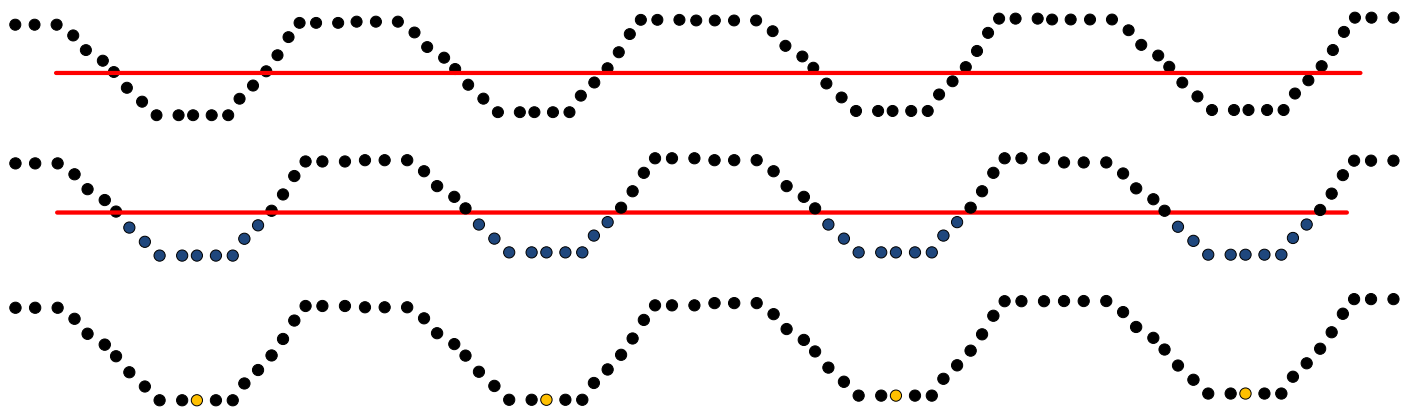


FIGURE 117: FIND ALL THE TRACK POINTS ON A LINE AT THE SAME TIME

This way is more efficient, but is trickier, because sometimes one groove is missing or some bad points can cause computing errors. Such events produce the result seen below.

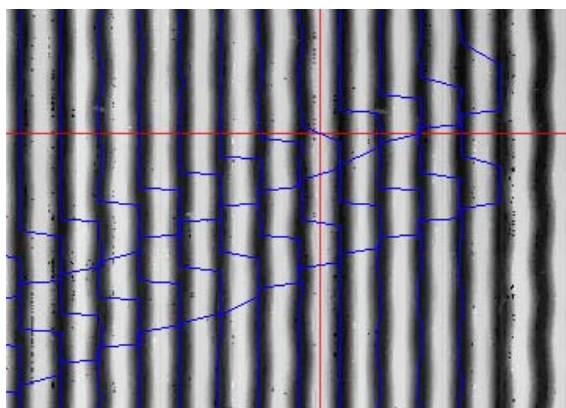


FIGURE 118: ERROR OBSERVED

This method needs much improvement and testing to be wholly efficient.

11.4 OBJECT-ORIENTED PROGRAMMING

Many additions have to be implemented in PRISM. At the beginning of our project, the code was essentially concentrated in one huge file (~6,000 lines). We began by understanding this code and adding some comments to it. We implemented some new functions, but it was very difficult and time-consuming. After consultation with Earl Cornell, we decided to fully change the application's architecture into object-oriented (OO).

Previous class diagram of the application PRISM2005

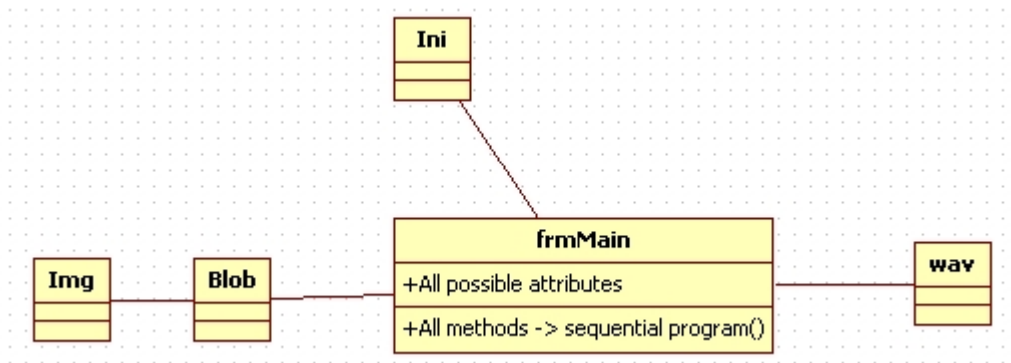


FIGURE 119: ORIGINAL CLASS DIAGRAM

11.4.1 CHOICE

Some advantages of OO programming: [10]

- **Simplicity:** software objects model real world objects, so the complexity is reduced and the program structure is very clear.
- **Modularity:** each object forms a separate entity whose internal workings are decoupled from other parts of the system.
- **Modifiability:** it is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program.
- **Extensibility:** adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones.
- **Maintainability:** objects can be maintained separately, making locating and fixing problems easier.
- **Re-usability:** objects can be reused in different programs.

11.4.2 CLASS DIAGRAM

Because of the large number of attributes in the Hardware, Cylinder, Disc, Disc_Tilted and all child classes, these one are not visible in the class diagram below but commented in the code.

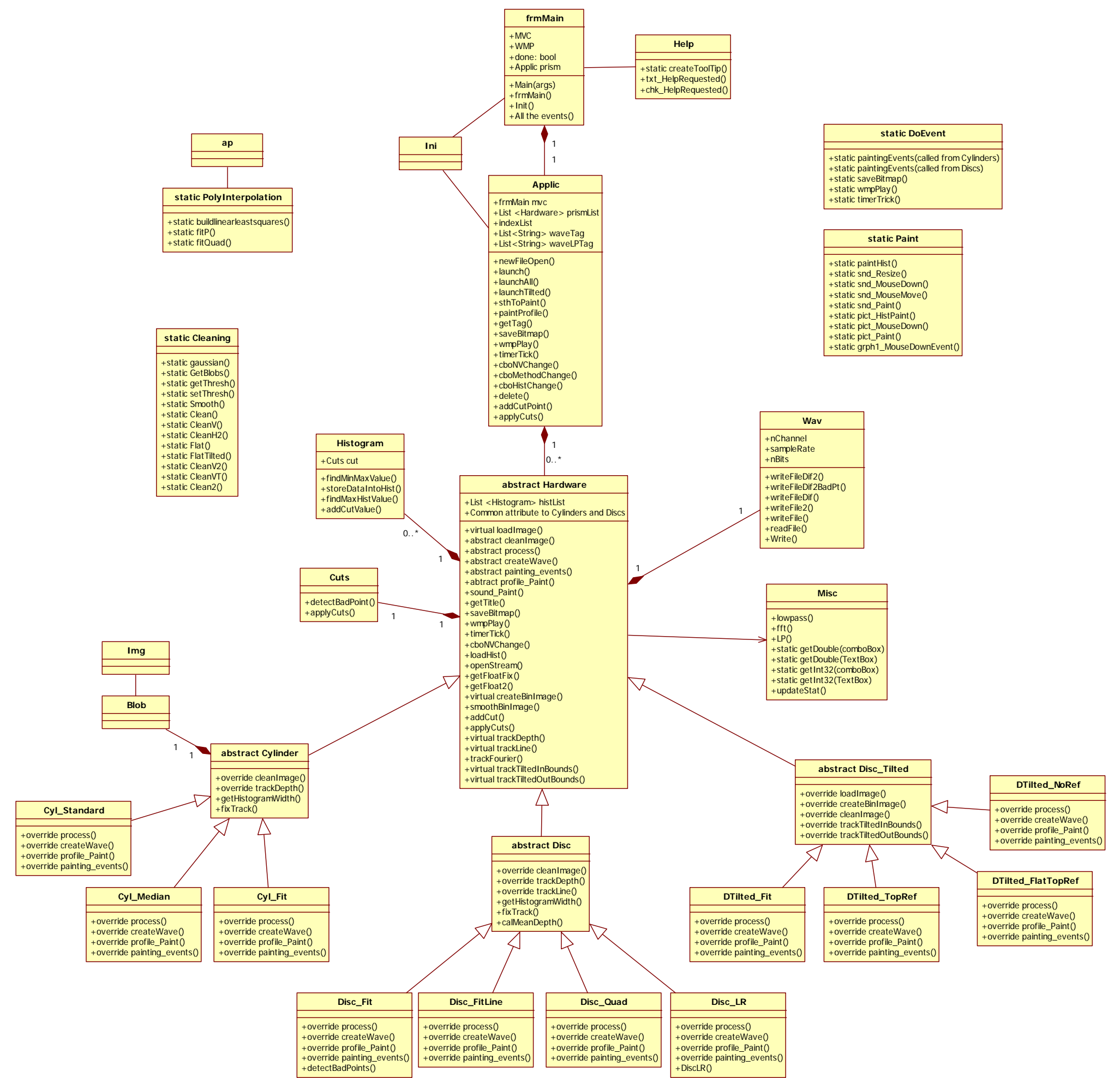


FIGURE 120: NEW CLASS DIAGRAM

11.4.3 DESCRIPTION OF CLASSES

frmMain

This is the starting class of PRISM. It contains the *Main* method, which is the entry point of the program. This class contains Model and the View (automatically generated with the Visual Studio's Designer). All the controls and events are managed in it. We are aware that the best solution would be to follow the MVC (Model – View – Controller) model, but we did not have the time to accomplish that in addition to all our other tasks.

Ini

Initialize the default parameter during the loading of the application.

Applic

The system's brain: According to the GUI's parameters, create right type's object (the derivate classes from "Cylinder", "Disc" and "Disc_Tilted"...) and launch the sequence Load Image – Clean – Track – Process. These objects are put into a List that manages the "Hardware" list and the GUI parameters. For each event, apply them on the current object treated by the application. All .wav files' path directories created (with and without low pass filter) are stored into lists of "String." Even the object has been deleted in the application; the user can still hear all the sounds created.

Hardware

Generic abstract class, which contains attributes and methods common to the discs and the cylinders. Read the stream and load the image. Implement some of events that are executed in the same way by all the derived Hardware's classes. All the public method of the derived classes are prototyped with either the keyword "abstract" (which mean all children must implement this method) or "virtual" (which means this method can be overridden).

Cylinder

Abstract class, which contains attributes and methods common to the cylinders' type processing, essentially manages cleaning and tracking. Inherit attributes and methods from Hardware class.

Disc

Abstract class. Realizes the same functions as Cylinder, but for the discs. Provides more tracking. Inherits attributes and methods from Hardware class.

Disc_Tilted

Abstract class. More than Cylinder and Disc, this class provide new "loadImage" cleaning and tracking algorithms. Inherit attributes and methods from Hardware class.

Cyl_Standard, Cyl_Fit, Cyl_Median, Disc_Fit, Disc_FitLine, Disc_Quad, Disc_LR, DTilted_Fit, DTilted_TopRef, DTilted_FlatTopRef, DTilted_NoRef

These classes inherit attributes and methods from respectively the Cylinder, Disc and Disc_Tilted classes. Each of them implements their own process technique, create their .wav file and draw their profile with their own specificity into the GUI.

Histogram

Attached to the Hardware class, can create and paint histograms from various data.

Cuts

Attach to histogram class, according to values of cuts; detect bad points on the histogram's input data and cut the array previously passed.

Cleaning

Catalog of various cleaning methods. Static class -> no attributes, no instance, only static methods.

Wav

Build the sound wave file on 8 or 16 bits according to the GUI's parameters. Compute derivative and do resampling if needed.

Paint

This class has to paint the different panel of PRISM. Static class.

DoEvent

Treat the events. According to the type of event, call the right paint method or execute the request. Static class.

Misc

Catalogue of generic methods used by all classes derived from Disc or Cylinder.

Blob

Contains blob's characteristics and various useful methods for dealing with them. Depends on Cylinder.

Img

Collaborate with blob class.

PolyInterpolation

Catalog of least square method and higher degree's polynomial interpolation. Static class.

ap

Collaborate with PolyInterpolation class.

11.4.4 CODE'S BEHAVIOR

This section shows a sample of examples illustrating how the code works and in which sequence the classes are called.

In referring to derived classes, we mean Cyl_Standard, Cyl_Fit, Cyl_Median, Disc_Fit, Disc_FitLine, Disc_LR, Disc_Quad, DTilted_Fit, DTilted_TopRef, DTilted_FlatTopRef or DTilted_NoRef.

The .cs means **c** sharp (C#)

11.4.4.1 LAUNCH PRISM2009

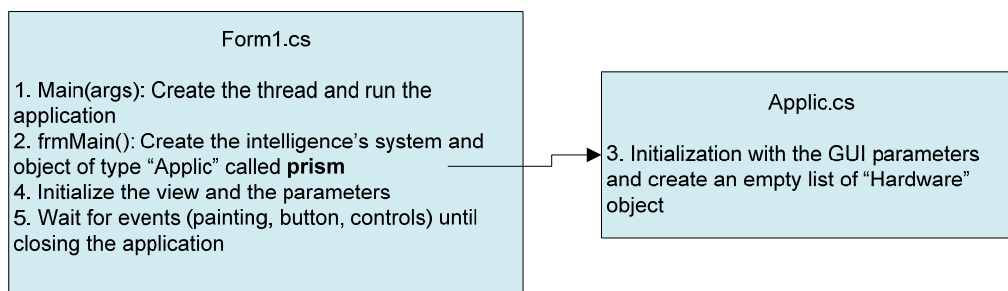


FIGURE 121: LAUNCH PRISM 2009 SEQUENCE

11.4.4.2 OPEN A FILE

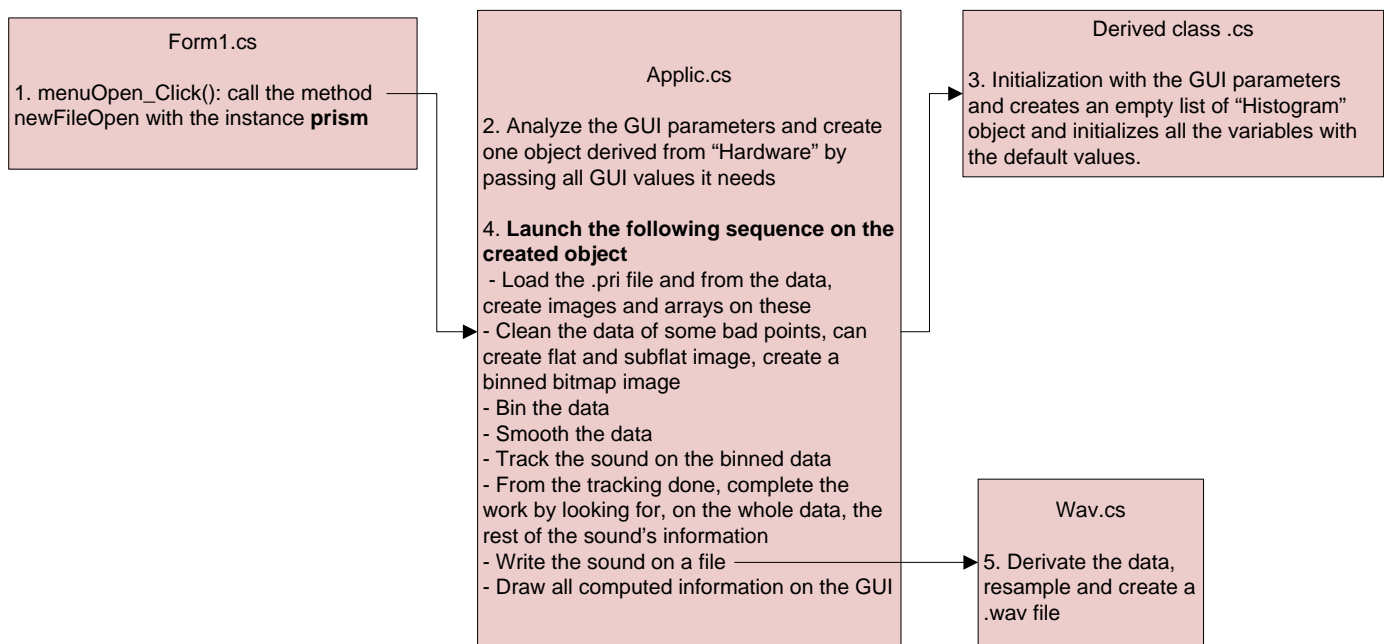


FIGURE 122: OPEN A FILE SEQUENCE

11.4.4.3 PAINTING EVENTS

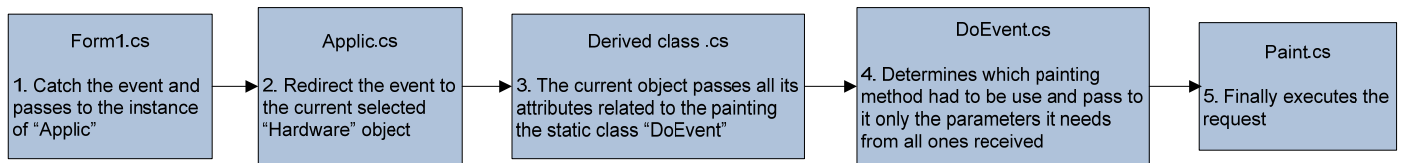


FIGURE 123: PAINTING EVENTS SEQUENCE

11.4.4.4 HISTOGRAMS AND CUTS

Below is the sequence followed when clicking on a histogram to define cuts.

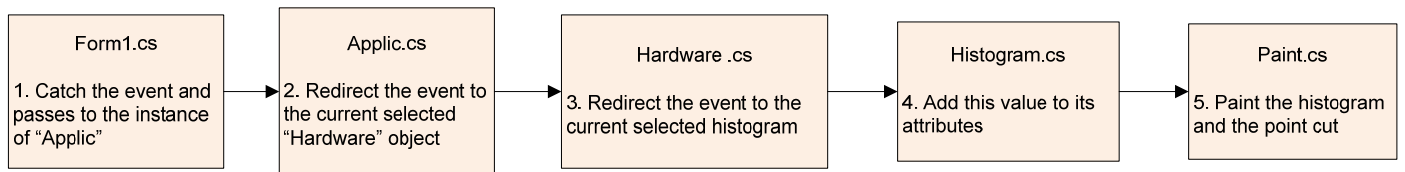


FIGURE 124: CLICK ON THE HISTOGRAM SEQUENCE

11.5 MODIFICATIONS ON THE GRAPHICAL USER INTERFACE (GUI)

In red; parts added or modified,

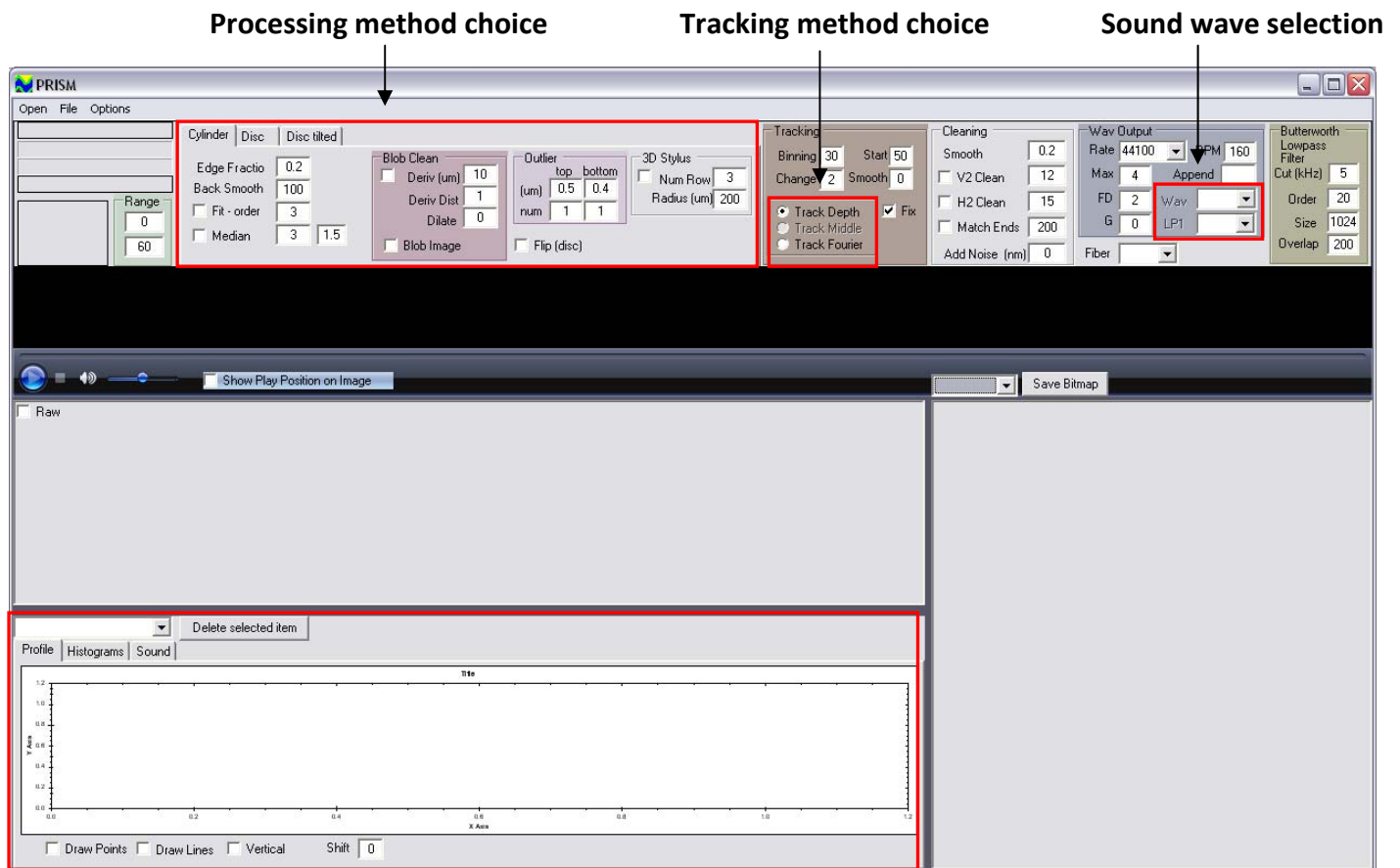


FIGURE 125: PRISM 2009 GUI

Statistics panel

11.5.1 PROCESSING METHOD CHOICE

First we split and sort the parameters that have common functionalities into three tabs: Cylinder, Disc and Disc Tilted. Any parameters in the selected tab can interfere with the two others. Below, an overview of the options available in each tab. For each option, a tooltip exists. The user moves the mouse over the chosen textbox and a tip appears with an explanation.

FIGURE 126: CYLINDER PANEL

FIGURE 127: DISC PANEL

FIGURE 128: DISC TILTED PANEL

Figure 129 shows the configuration panel on our arrival. Earl Cornell began by reorganizing this, and we continued this process until the result shown at the beginning of the present chapter.

FIGURE 129: ORIGINAL PARAMETERS PANEL

11.5.2 STATISTICS PANEL

With the OO programming, we can keep all computed data for each opened file. The combo box at the top left shows the list of the opened file and the processing method used. A button “Deleted selected item” removes the selected object from the list and releases memory space. This button is very useful if the user wants to launch several processing methods or to open some *.pri file because it avoids the “OutOfMemoryException.”

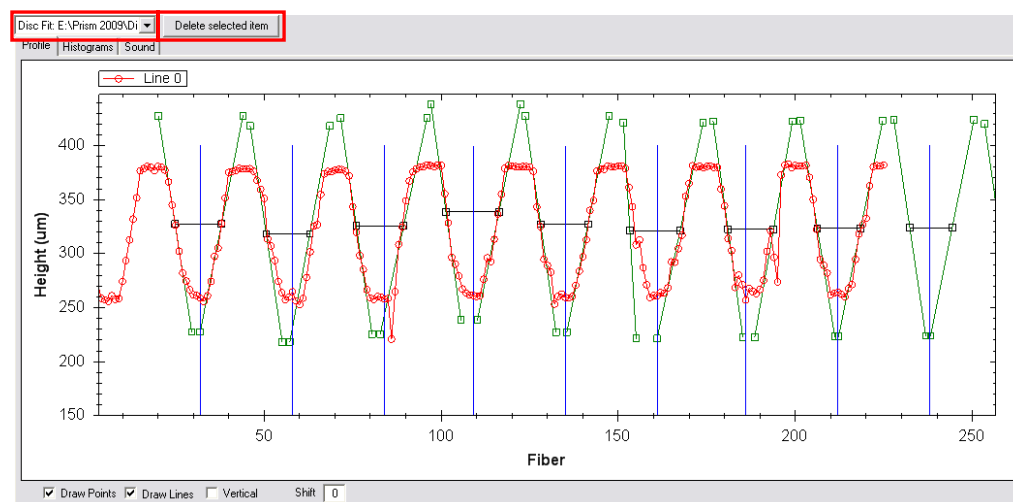


FIGURE 130: PROFILE GRAPH

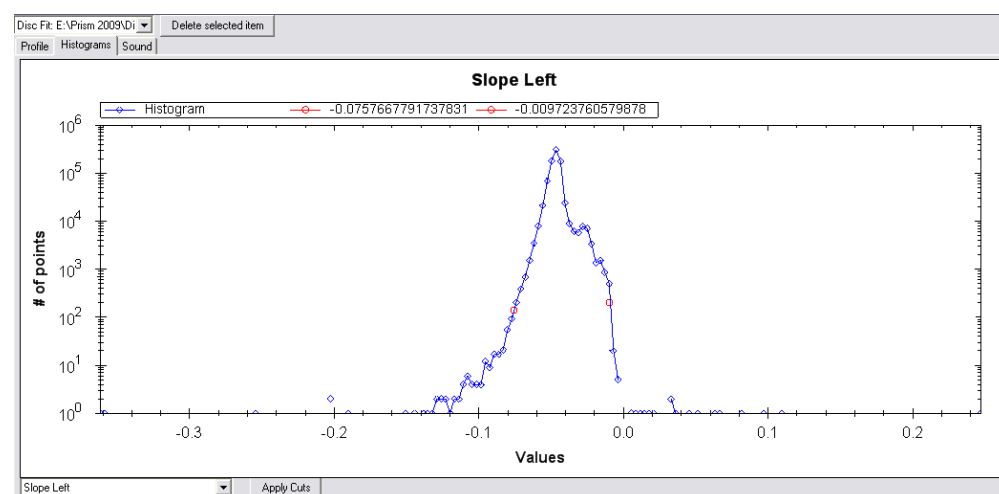


FIGURE 131: HISTOGRAM GRAPH

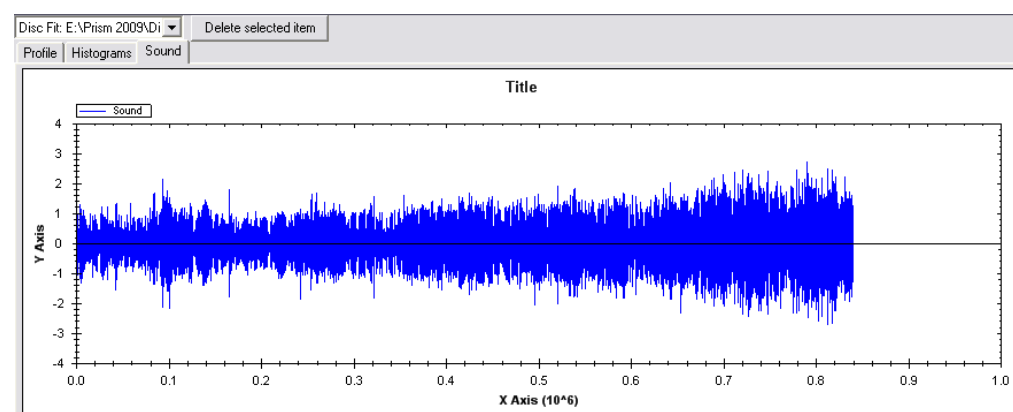


FIGURE 132: SOUND GRAPH

11.5.3 OTHERS

The other changes are minor: the tracking options have been transposed into radio buttons.

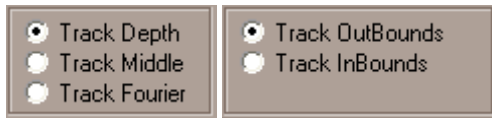


FIGURE 133: TRACK TYPE OPTIONS

Two combo boxes have been added to the “Wav” and “LP1” buttons. These contain a list of all .wav files created. Even though the “delete selected item” button described earlier can delete the item, the sound will be preserved.

FIGURE 134: PART OF WAV GROUP OPTIONS

At present the program automatically adapts some parameters on the GUI when the user selects a processing method. For example: changes from tab Cylinder to tab Disc and select Disc Fit -> blobs have to be deactivated, NoFlip180 selected, Track Middle becomes available, etc. These small functions allow a user who is not familiar with the program to run it without crashing it.

12. FURTHER DEVELOPMENT

At the conclusion of the project, more could be done to improve some parts of the acquisition, the processing and the PRISM 2009’s GUI to improve the quality of the extracted sound.

12.1 INCLUDE VARYING FIBER SPACING

As explained by Hezel and Novak in “Extraction of Sound Using 3D,” the space between two points depends on the fiber number and the height of the measured sample. This is due to the spread of the MPLS’ beam. Instead of taking only the height of a sample as input data, we should also take a relative lateral position. This relative lateral position can be calculated by given function. This function is a characteristic of the MPLS 180.

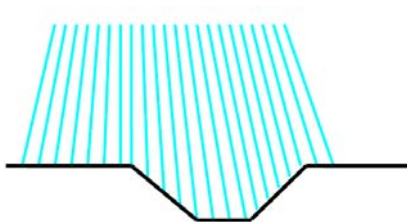


FIGURE 135: VARYING FIBER SPACING

12.2 MERGING BOTH EDGES

As explained above, because of the tilted acquisition, both edges of the groove are stored in two different files and treated separately. To improve the redundancy of data and the quality of the sound, each corresponding sample of both edges should be compared and a smart choice made. If both edges correspond to each other, they should be averaged to reduce the noise (3dB if independent). If one of the edges is bad, only the good one should be used.

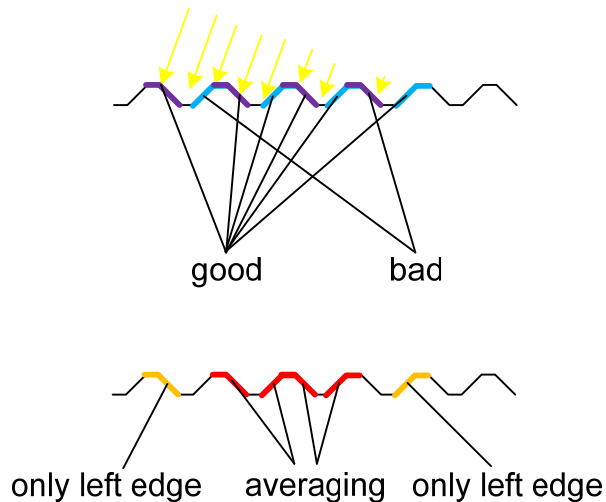


FIGURE 136: EXAMPLE OF MERGING BOTH EDGES

12.3 RESAMPLING TECHNIQUE

As explained above, to avoid the errors due to the resampling, the acquisition should be done at a sample rate corresponding to the required output signal. The exposure time parameter and the rotation velocity can be modified in the LabView's acquisition code.

12.4 LAUNCH, COMPARE AND MIX THE DIFFERENT DISC METHODS

As PRISM has different methods to find the center of the groove, it is useful to compare them. The user must check “Run all the discs methods” and choose his *.pri file.

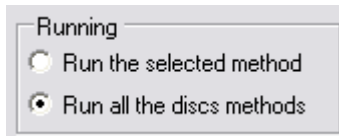


FIGURE 137: GUI PART OF DISC TAB

Once the program has finished, each result can be read and analyzed separately by picking the selection box above the paint panel.

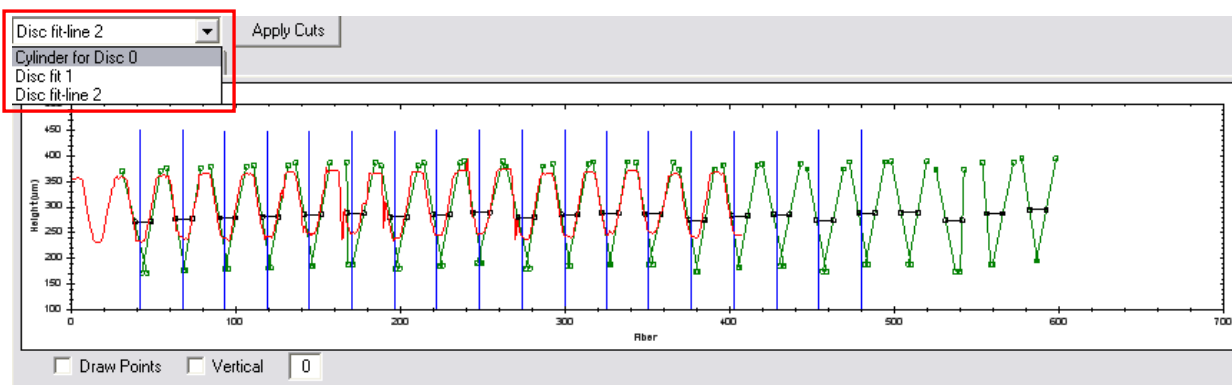


FIGURE 138: THE THREE DISC METHODS ARE STORED AND MUST NOW BE COMPARED

Another function will be added to read the three reconstructed sounds simultaneously and superimpose the various histograms created.

If some of these return similar results, we will combine them (do an average), hoping to decrease the global noise level. An intelligent average will give the best effect. Indeed, instead of making just an average of all the points, it is preferable to detect the inconsistent points and to average only the good points.

12.5 FIX THE PAINTING METHOD

There is a small defect of drawing concerning the tracking and the processing function. Indeed, when the yellow rectangle touches or is beyond the left side of the binned image, the tracing of the green lines works well.

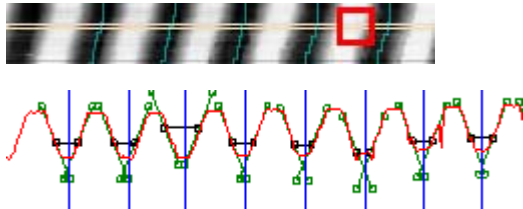


FIGURE 139: CORRECT DRAWING

When the yellow rectangle is farther than the left side of the binned image, the tracing of the green lines is shifted.

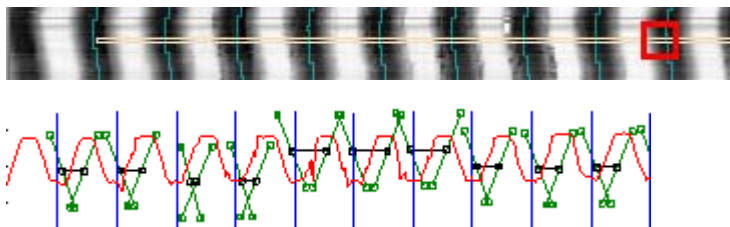


FIGURE 140: INCORRECT DRAWING

Before considering a painting error, we studied what we seemed to be a problem of tracking: at the left of the image, the tracking is correctly centered, but more we go to the right, the tracking tends to the left of the groove. This is why we developed another tracking algorithm.



FIGURE 141: AT THE END THE TRACKING TENDS TO THE LEFT

12.6 REDUCE THE NOISE ON THE DISC METHODS

The disc method used with the straight probe works well but involves some imperfections. The example below was realized with the “Disc Fit” method. The green lines must fit on the slopes of the groove, but in some cases, there is a small difference. These differences are a source of error and noise.

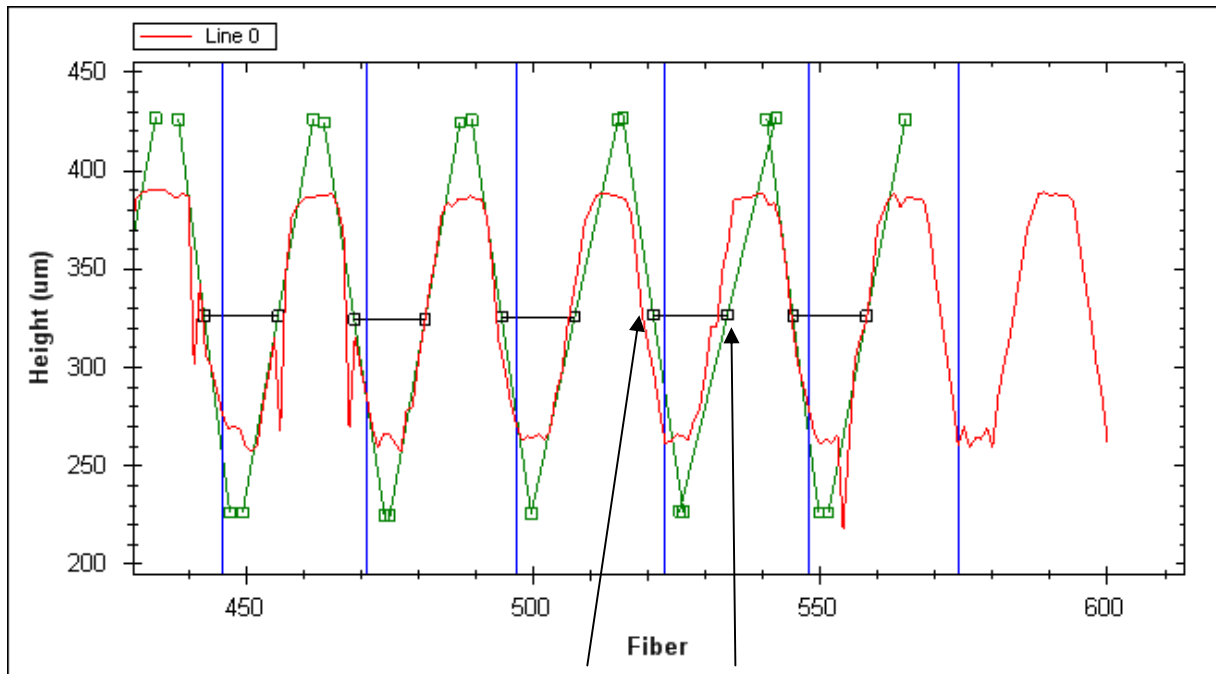


FIGURE 142: EXAMPLE OF INACCURACIES

The blue lines represent the sound. We can see lots of small peaks that disturb the sound.

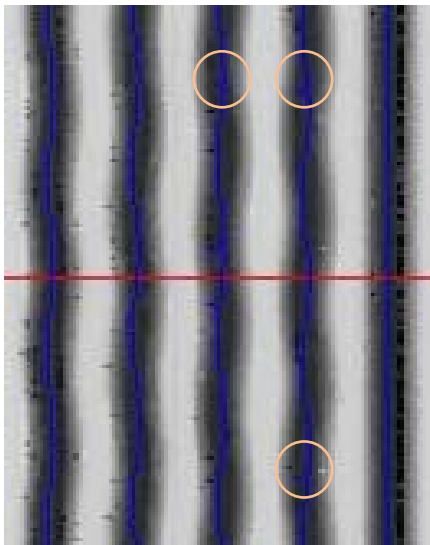


FIGURE 143: SOURCE OF NOISE

13. CONCLUSION

This project has been a new step in the process of “3D sound extraction” from 78 rpm. We can now read some bright reflective discs. This was not possible with the probe positioned vertically above the disc. Our work on the existing application is part of the extraction of sound project led by Professor Carl Haber at LBNL.

We made some improvements in the acquisition of the data by tilting the 3D probe, on the processing by improving and developing some new algorithms, on the analyze software (PRISM 2009) by transposing it in an object-oriented way. This improvement allows the user to open different files and compare the different methods of processing.

At this point, as mentioned previously, the acquisition process and the analyzing software could be improved in several ways to minimize the noise in the extracted sound. At each step of the whole extracting process, imprecision in the calculations adds some noise to the final sound. The acquisition is subject to external disturbances like parasite frequencies (TV line, line 60Hz) and incidental vibration near the acquisition system. The tracking and the processing are the most sensitive part during the analysis. Some of these improvements could be carried out either by Earl Cornell or by future students.

During this project, we learned a great deal about the application of signal processing and acquired some coding skills in C#, TCL, G (graphical language of LabView). We covered the history of recording by learning about cylinders, 78rpm and precursors in the domain, such as Bell and Edison. We also applied our knowledge in object-oriented programming and algorithms.

We found this project particularly interesting because of the breadth of the domain covered and the variety of work (history of sound, signal processing, coding, mechanical work and so on). We would highly recommend this project to other students.

14. ACKNOWLEDGMENTS

We would like in particular to thank Carl Haber for his advice, help, support, patience and especially for giving us the opportunity to do this work at the Lawrence Berkeley National Laboratory. We also thank Professor Ottar Johnsen for making it possible for us to go abroad for our diploma work and for visiting and supporting us during the project. We would also like to thank Earl Cornell for his indispensable assistance on a wide variety of issues during the entire project.

Thanks also go to the University of Applied Science of Fribourg, which has partially funded our stay in Berkeley and to Vicky Elliott for correcting the paper.

Berkeley, August 15, 2009

Philippe Ballestraz

Raynald Seydoux

15. BIBLIOGRAPHY

- [1] <http://history.sandiego.edu/GEN/recording/scott.html>
- [2] Vitaliy Fadeyev and Carl Haber, "Reconstruction of Mechanically Recorded Sound by Image Processing," Berkeley, LBNL Report 51983, Physics division, March 2003.
- [3] Sylvain STOTZER, "Phonographic Record Sound Extraction by Image Processing," PhD Thesis, Fribourg, 2006.
- [4] Noé Lutz, Michel Yerly, "Studies of Mechanical Recording Media with 3D Surface Profiling Methods: Data Collection and Analysis", Berkeley, LBNL, October 2005.
- [5] Sven Hezel and Severin Nowak, "Extraction of sound using 3D", Berkeley, LBNL, November 2008.
- [6] <http://www.firstsounds.org>
- [7] Carl Haber, "MPLS Scanner Status Report", LBNL, September 2008.
- [8] Ottar Johnsen, "Digital Filters," Fribourg, EIA FR, September 2006.
- [9] Ottar Johnsen, "Digital signal processing and theory », Fribourg, EIA FR, October 2005.
- [10] M. Melly, Department of Electronics and Computer Science, University of Southampton, "An object-oriented linkbase for Microcosm," <http://eprints.ecs.soton.ac.uk/857/3/html/node3.html>
- [11] C. Bill, 78rpm Beginner's Page, <http://www.bill78.btinternet.co.uk/beginer.htm>