

**RECONOCIMIENTO DE PARTITURAS MUSICALES POR MEDIO DE VISIÓN
ARTIFICIAL**

**MANUEL ENRIQUE ALDANA SÁNCHEZ
JUAN SEBASTIAN PERDOMO MÉNDEZ**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA – HUILA, 2013**

**RECONOCIMIENTO DE PARTITURAS MUSICALES POR MEDIO DE VISIÓN
ARTIFICIAL**

**MANUEL ENRIQUE ALDANA SÁNCHEZ
JUAN SEBASTIAN PERDOMO MÉNDEZ**

**Proyecto de grado presentado
Para optar al título de
Ingeniero Electrónico**

**Director del proyecto
JOSE JESUS SALGADO
Ingeniero Electrónico Msc.**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA – HUILA, 2013**

Nota de aceptación:

Firma del representante

Firma jurado

Firma jurado

Neiva, 7 de Junio de 2013

A mi madre, padre y hermano
Manuel Enrique Aldana Sánchez

A mi madre y hermanos
Juan Sebastián Perdomo Méndez

AGRADECIMIENTOS

Agradezco a Dios y a todas aquellas personas que fueron parte de este ciclo de mi vida y me apoyaron de una u otra forma; en especial a mi madre, padre, hermano, novia y suegros quienes fueron una parte esencial en la formación del profesional integro que soy en día.

Gracias.

MANUEL ENRIQUE ALDANA SÁNCHEZ

Gracias a Dios por permitirme llegar hasta este punto determinante en mi vida, por darme la salud y la fuerza para sobrellevar y superar los obstáculos que se me han presentado a lo largo de mi vida y que me han permitido estar hoy acá.

A mi madre por estar siempre a mi lado, por sus sabios consejos, por tenderme la mano en los buenos y malos momentos, por entender mis ausencias y mi mal genio, por inculcar en mi los valores y motivación que me hicieron una persona de bien, pero más que nada por su grande y tierno amor.

Gracias a mis hermanos, familiares y personas cercanas, que siempre tuvieron una palabra de aliento, y que aunque en algunas ocasiones nos encontremos lejos, sé que puedo contar con ellos. A todos y cada uno de los profesores que me dieron clase; por su dedicación y su tiempo, por todas las buenas enseñanzas que me han brindado y sé que serán de bien para el futuro.

Por ultimo quiero agradecer a mis compañeros, que me permitieron entrar en sus vidas estos cinco largos años, compartiendo buenos y malos momentos, dentro y fuera del salón de clase, apoyándonos mutuamente durante nuestra formación. Fueron días en el que el tiempo parecía pasar lentamente, y que sin darnos cuenta nos encontramos en el final de la carrera y vemos que no solo es importante aprender de libros, sino también de las personas. Me siento orgulloso de haber conocido gente maravillosa, con la que he pasado gratos momentos y que hasta el día de hoy seguimos siendo amigos.

Gracias.

JUAN SEBASTIAN PERDOMO MÉNDEZ

TABLA DE CONTENIDO

	pág.
1. OBJETIVOS	15
2. DESARROLLO DEL ARTE.....	16
2.1 TEORÍA MUSICAL.....	16
2.2 LA PARTITURA	16
2.2.1 Notas.....	18
2.2.2 Clave.....	18
2.2.3 Figuras.	20
2.2.4 Escalas Musicales.	23
2.2.5 Tonalidad.....	24
2.3 RECONOCIMIENTO ÓPTICO DE CARACTERES MUSICALES.....	25
2.4 TRABAJO DE OTROS AUTORES	26
2.4.1 Aoyama y Tojo (1982).....	26
2.4.2 Maenaka y Tadokuro.....	28
2.4.3 Mcgee y Merkley.	28
2.4.5 ETIENNE SICARD	29
2.4.6 Miyao.....	29
2.5 TÉCNICAS USADAS EN EL PROCESAMIENTO DE IMÁGENES	29
2.5.1 Métodos de Preprocesamiento.	30
2.5.2 Segmentación.....	30
3. DESARROLLO Y DESCRIPCION DEL SISTEMA.....	32
3.1 FUNCIONALIDAD.....	32
3.2 DESCRIPCIÓN GENERAL DEL SISTEMA	33
3.3 BLOQUE DE PREPROCESAMIENTO	34
3.3.1 Conversión Escala de Grises, Filtrado y Conversión A Binario.....	34
3.3.2 Detección y Corrección de Líneas Horizontales	35
3.3.3 Extracción de Características del Pentagrama.....	38
3.4 BLOQUE CLASIFICADOR	41
3.4.1 Imagen con Símbolos Aislados.....	41
3.4.2 Imagen Binaria Negada.....	42
3.4.3 Etiquetado.	42
3.4.4 Comparación.	43
3.4.5 Asignación de Símbolos.	43
3.4.6 Símbolos Aislados.	45
3.4.7 Organización de Símbolos y Almacenamiento en Estructura.....	45
3.5 BLOQUE ANALIZADOR.....	45
3.5.1 Eliminación Líneas Verticales.	46
3.5.2 Factor Forma y Eliminación Falsas Cabezas de Nota.....	47
3.5.3 Identificación de Líneas o Espacios de la Figura.....	48
3.5.4 Asignación y Adecuación del Pentagrama y Sonido a la Nota.	49

	pág.
3.6 APLICADOR DE ATRIBUTOS	53
3.6.1 Aplicador de Clave	54
3.6.2 Aplicador de Alteraciones	54
3.7 GENERADOR DE AUDIO	58
3.7.1 Almacenamiento de Ondas Sinusoidales.....	58
3.7.2 Conversor y Exportador de Audio.....	59
3.8 BLOQUE SELECTOR DE FIGURAS	59
3.8.1 Selección Recuadro	60
3.8.2 Verificación Sector-Figuras	60
3.8.3 Aplicación de Atributos y Conformación de Canción.....	62
4. RESULTADOS.....	63
4.1 INTERFAZ GRAFICA DEL SOFTWARE.	66
4.2 SOFTWARE.....	68
5. CONCLUSIONES.	69
6. RECOMENDACIONES.....	71
BIBLIOGRAFIA.....	72
ANEXOS.....	74

LISTA DE FIGURAS

	pág.
Figura 1. Partitura musical.	17
Figura 2. El pentagrama.	17
Figura 3. Numeración de las líneas y espacios del pentagrama.	17
Figura 4. Líneas adicionales.	18
Figura 5. Clave de Sol.	18
Figura 6. Ubicación de notas en clave de Sol.	19
Figura 7. Notas con líneas adicionales en escala de Sol.	19
Figura 8. Clave de Fa.	19
Figura 9. Ubicación de notas en clave de Fa.	20
Figura 10. Formas de ubicación de las notas en el pentagrama.	21
Figura 11. Representación de notas agrupadas.	21
Figura 12. Ligaduras.	21
Figura 13. El puntillo.	22
Figura 14. Forma de denotar las figuras de alteración.	23
Figura 15. Tonalidad.	24
Figura 16. Imagen después de una burda eliminación de las líneas del pentagrama.	26
Figura 17. Clasificación.	27
Figura 18. Escaneo para determinar tipo de figura musical.	28
Figura 19. Notación musical antigua.	28
Figura 20. Vecindad de 4.	31
Figura 21. Vecindad de 8.	31
Figura 22. Esquema general del sistema.	33
Figura 23. Bloque preprocesamiento.	34
Figura 24. Imagen o partitura.	34
Figura 25. Partitura filtrada y binarizada.	35
Figura 26. a) Imagen binaria negada, b) Imagen etiquetada, c) Codificación o etiqueta asignada.	36
Figura 27. Ejemplo del funcionamiento de regionprops.	36
Figura 28. Algoritmo corrección de ángulo.	37
Figura 29. a) Imagen con ángulo de inclinación, b) Imagen sin ángulo de Inclinación.	38
Figura 30. Pentagramas sin ruido.	38
Figura 31. Características del pentagrama.	39
Figura 32. Diagrama de bloques de bloque clasificador.	41
Figura 33. Imagen binaria con solo símbolos.	41
Figura 34. Imagen binaria negada.	42
Figura 35. Imagen segmentada.	42
Figura 36. Base de datos e imágenes no reconocidas.	44
Figura 37. Bloque analizador.	45
Figura 38. Recorrido y determinación de líneas a eliminar.	46

Figura 39. Ejemplo de figuras antes de eliminar líneas verticales.	46
Figura 40. Ejemplo de figuras después de eliminar líneas verticales.....	47
Figura 41. Factor Forma.	47
Figura 42. Búsqueda de ubicación de la figura.	49
Figura 43. Rango de frecuencias, tomando como referencia el piano.	50
Figura 44. Asignación para frecuencias de las notas en base a “La”.....	51
Figura 45. Normalización de notas a “La”.	51
Figura 46. Asignación con notas sostenidas.....	52
Figura 47. Bloque aplicador de atributos.	53
Figura 48. Valores de Sonido para Clave.	54
Figura 49. Saltos de Tonalidad a las notas, a) impares, b) pares.	57
Figura 50. Armadura y figuras de Alteración.....	57
Figura 51. Bloque generador de audio.....	58
Figura 52. Bloque selector de figuras.	59
Figura 53. Recuadro selector de figuras.	60
Figura 54. Ejemplo recuadro por fuera del último pentagrama.	61
Figura 55. Ejemplo recuadro dentro de los pentagramas.	61
Figura 56. Búsqueda de figuras dentro del recuadro de selección.	62
Figura 57. Partitura Noche de Paz con inclinación de -0,8 grados.	63
Figura 58. Extraído de Himno de la Alegría de Beethoven con inclinación de un grado.....	63
Figura 59. Partitura Campanas Sobre Campanas con inclinación de 1,2 Grados.	64
Figura 60. Ejemplo 4. (Extraído de Himno de la Alegría de Beethoven).....	65
Figura 61. Ejemplo 5. (Extraído de Chiquita de Abba).....	66
Figura 62. Interfaz gráfica del software.....	67
Figura 63. Programa ejecutado.....	67

LISTAS DE CUADROS

	pág.
Cuadro 1. Símbolos y nombres.....	20
Cuadro 2. Figuras de alteración.....	22
Cuadro 3. Otras figuras de alteración.	23
Cuadro 4. Nivel del proceso.....	29
Cuadro 5. Tipos de tonalidad.....	55
Cuadro 6. Tiempo de preprocesamiento.....	64
Cuadro 7. Tiempo de rotación.....	65

GLOSARIO

Armadura: es la cantidad de alteraciones propias a las notas de la partitura, que se escriben al comienzo de un pentagrama y después de la clave. También permite saber en qué tonalidad está escrita la partitura.

Base de datos: es un conjunto de información organizada que pertenece a un mismo contexto, generalmente se organizan de forma sistemática para tener un acceso fácil y rápido en caso de necesitarlo.

Binarización: es el proceso que convierte una imagen digital en blanco y negro.

Centroide: es el punto centro o promedio de una figura geométrica en el cual se encuentra su centro de masa o de gravedad.

Frecuencia: es una magnitud que mide la cantidad de veces que se repite un determinado fenómeno o suceso en una unidad de tiempo, su relación es ciclos por segundos (Hz).

Frecuencia de muestreo: la frecuencia de muestreo es la cantidad de muestras que se tienen de una señal en una unidad de tiempo.

Normalización: es ajustar el tamaño, la forma o posición con el objetivo de minimizar la variación entre imágenes de la misma clase.

Píxel: es un único punto en una imagen gráfica digital, la sucesión de estos píxeles en miles o millones, construyen una imagen.

Preprocesamiento: se refiere a las técnicas existentes, para la reducción del ruido, realce del contraste, detalles o características de la imagen.

Reconocimiento: es el proceso que asocia un significado a un conjunto de objetos reconocidos.

Resolución: se refiere a cuan nítida o visualmente se ve mejor una imagen, está definida por la densidad de píxeles en una imagen, a mayor cantidad de píxeles mayor resolución de la imagen.

Ruido: cualquier tipo de señal no deseada.

Segmentación: es el proceso de dividir la imagen en objetos que son de interés.

Umbral: es la cantidad mínima que debe tener una señal para existir dentro de un sistema, en otras palabras es el punto de inicio o de ingreso a determinado hecho.

WAV: formato de audio digital creado en 1991 por Microsoft e IBM, se utiliza para almacenar sonidos.

INTRODUCCIÓN

Uno de los sentidos más importantes del hombre es la visión, siendo este el más versátil y el que mayor información proporciona del mundo físico. A pesar que el hombre ha estado interesado en resolver como él percibe el exterior a través de la vista, los primeros experimentos computacionales en el campo de visión por computador no llegaron sino hasta finales de los años cincuenta.

La visión artificial o también conocida como visión por computador, es uno de los muchos campos de la inteligencia artificial, y puede entenderse como la programación de un computador para que “comprenda” una escena o características en una imagen, típicamente el reconocimiento de objetos, tipos de objetos, o modelos de “estructura espacial”. La mayor dificultad depende esencialmente del grado de control que se posee sobre las “condiciones” del tema a tratar. Una aplicación de visión artificial puede simplificarse tanto como se desee, restringiendo el tipo de escenas a procesar; por tanto, uno de los objetivos de la visión artificial es obtener sistemas cada vez más abiertos.

Actualmente existen varias aplicaciones que trabajan con el reconocimiento de caracteres conocidos como OCR (Reconocimiento Óptico de Caracteres), también con reconocimiento de voz y palabras, pero el reconocimiento musical no es muy común, y de hecho hay pocos programas enfatizados en esta área. Es aquí donde comienza a surgir una nueva rama llamada OMR, conocida también por sus siglas en español como Reconocimiento Óptico de Música, que da a un computador la capacidad de reconocer y comprender los contenidos de una partitura.

Se propone desarrollar un sistema de reconocimiento de notas musicales por medio de visión artificial, usando Matlab como herramienta de programación para reconocer las figuras musicales en una partitura digital, además de poder reproducir y generar el archivo de audio de dicha partitura.

ABSTRACT

One of the most important senses of man is the vision, which is the most versatile and the largest provider of information from the physical world.

Despite the fact that man has been interested in solving questions about how the outside world is perceived using the sense of sight, the first computational experiments in the field of computer vision were not introduced until the end of the fifties.

Machine vision, also known as computer vision, is one of the many fields of artificial intelligence. It is meant to be understood as the computer programming in order to the machine "understands" a scene or features in an image. This "understanding" typically covers such issues as object recognition, object types and "spatially-structured" models. The main difficulty depends crucially on the degree of control over the conditions of the subject in question.

A machine vision application can be simplified as much as desired by restricting the type of scenes; therefore one of the aims of artificial vision is to obtain increasingly open systems.

There are a series of applications that currently work with character recognition known ORC and Speech Recognition and word, however, the music recognition is uncommon and there are few dedicated programmes designed especially for this purpose.

This is where a new branch called OMR is beginning to appear, it is also known by its Spanish acronym as Optical Music Recognition which gives the computer the ability to recognize and understand the musical contents of the score.

The project aims to develop a musical notes recognition system by artificial vision through the use of MATLAB as a programming tool for recognizing musical figures on a musical score in digital form. In addition to that, the audio file of this score may be generated and reproduced.

1. OBJETIVOS

Generales:

- Desarrollar un sistema que permita reproducir una partitura mediante visión artificial, mediante el software Matlab Versión 7.11.0.584 (R2010b).

Específicos:

- Estudio e interpretación de partituras musicales.
- Aplicar las técnicas de preprocesamiento para adquirir la información deseada a la imagen.
- Aplicar las técnicas de reconocimiento para la correcta identificación y posterior reproducción de los tonos o melodía a partir de la información obtenida del análisis de la partitura.
- Diseño de una interfaz gráfica en Matlab v. 7.11 amigable para el usuario y la elaboración de un manual para un correcto manejo del programa.
- Profundizar los conocimientos y técnicas utilizadas en visión artificial para el reconocimiento y la correcta identificación de notas musicales.

2. DESARROLLO DEL ARTE

2.1 TEORÍA MUSICAL

Todas las materias o todas las ciencias tienen su idioma o formas y expresiones, por ejemplo el lenguaje de las matemáticas es formal y abstracto, acompañada de números y símbolos, al igual que la física, la química, la propia lengua como el inglés o el latín. Se dice que la música es un lenguaje universal, puesto que se lee de igual forma en cualquier parte del mundo, pero deben existir algunas reglas, organización o sintaxis, es aquí donde aparece la importancia de la correcta interpretación de la partitura.

Antes de empezar a hablar de la partitura (que es el campo de interés), se hará mención al sonido.

El sonido, es la sensación percibida por el oído al recibir las variaciones de presión generadas por el movimiento vibratorio de los cuerpos sonoros.¹ Así mismo el sonido tiene cuatro características principales.

- La “altura”, que es el resultado de la frecuencia que produce el cuerpo sonoro. Debido a esto se encuentran sonidos “graves” o sonidos “agudos”. Cuanto mayor sea la frecuencia, más agudo es el sonido, en caso contrario será grave.
- La “duración”, que corresponde al tiempo que duran las vibraciones que producen un sonido.
- La intensidad es la fuerza con que se produce un sonido y depende de la energía, en una onda vendría representada por la amplitud.
- Cuarto el “timbre”, el cual permite identificar el instrumento, a pesar que se produzcan con la misma altura, duración e intensidad. Los sonidos que se escuchan son complejos y son el resultado de un conjunto simultáneos de sonidos.

2.2 LA PARTITURA

Es un documento en donde se indica cómo debe interpretarse una composición musical.

¹“Sonido” [en línea], disponible en: <http://es.wikipedia.org/wiki/Sonido>, recuperado: 8 de febrero de 2013.

El pentagrama es donde se centra toda la grafía musical. En éste se colocan los símbolos y notas, está compuesto por cinco líneas horizontales paralelas y equidistantes.

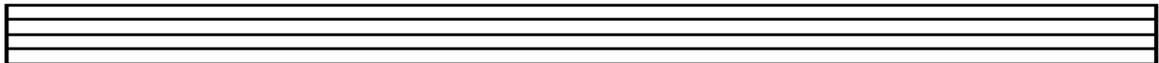
Figura 1. Partitura musical.

ROCKY Arr.: Roberto Reina

Flauta

Fuente: <http://mariajesusmusica.files.wordpress.com/2009/05/rocky.jpg>

Figura 2. El pentagrama.



Fuente de los autores.

En los espacios formados por las líneas del pentagrama, también se ubican las notas y símbolos. Las líneas y espacios se nombran de abajo hacia arriba como se muestra a continuación.

Figura 3. Numeración de las líneas y espacios del pentagrama.



Fuente de los autores.

En algunas ocasiones las notas exceden el ámbito del pentagrama y es necesario el uso de líneas adicionales. Sin embargo no es recomendable usar más de 3 líneas adicionales.

Figura 4. Líneas Adicionales.



Fuente: <http://www.aprende-gratis.com/teoria-musical/imagenes/lineas-adicionales.png>

Los sonidos más graves se escriben en la parte inferior del pentagrama, mientras que los agudos en la superior.

2.2.1 Notas. Son la representación de los sonidos en el pentagrama. Estas son siete y se representan de la siguiente forma.

1	2	3	4	5	6	7
LA	SI	DO	RE	MI	FA	SOL
A	B	C	D	E	F	G

2.2.2 Clave. Son símbolos utilizados como referencia para indicar el nombre o ubicación de las notas en el pentagrama. Por ejemplo, ¿Cómo saber que sonido produce una nota que se encuentra en la segunda línea del pentagrama?, no hay forma de saber a menos que se indique por medio de un símbolo gráfico. Este símbolo es la clave y sirve para discernir las notas en el pentagrama.

Clave de Sol. Indica que la nota Sol se encuentra ubicada en la segunda línea. Es la clave más utilizada.

Figura 5. Clave de Sol.



Fuente: <http://www.aprende-gratis.com/teoria-musical/imagenes/clave-sol.png>

Por lo tanto el resto de notas se ubican en la forma que se observa en la Figura 6.

Figura 6. Ubicación de notas en clave de Sol.



Fuente: http://www.teoria.com/referencia/media/lec_04a.gif

La ubicación de las notas en las líneas adicionales se observa en la Figura 7.

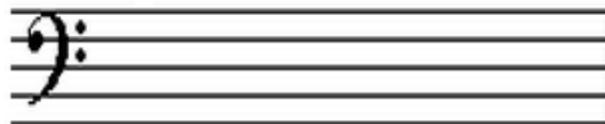
Figura 7. Notas con líneas adicionales en escala de Sol.



Fuente: http://www.teoria.com/referencia/media/lec_04a.gif

Clave de Fa. Esta es utilizada por instrumentos graves como el contrabajo, violonchelo, tuba o el trombón.

Figura 8. Clave de Fa.



Fuente: <http://www.aprende-gratis.com/teoria-musical/imagenes/clave-fa4.png>

También se conoce como clave de Fa en cuarta ya que va sobre la cuarta línea, Fa en tercera no es muy común. Las notas en el pentagrama con esta clave vienen dadas como se ve en la Figura 9.

Figura 9. Ubicación de notas en calve de Fa.



Fuente: http://www.teoria.com/referencia/media/lec_05a.gif

Hay otras claves como Do, pero son aún más escasas que Fa, por lo que se centrará en Sol que es las más conocida y usada.

2.2.3 Figuras. En la música existen dos tipos de figuras, de duración y alteración.

2.2.3.1 Figuras de duración. Son los símbolos que representan la duración de las notas. Se debe tener en cuenta que el tiempo de duración de las figuras es relativo según el género de la música de la partitura.

SILENCIOS. Indican la ausencia momentánea de sonido, quedando así para cada figura un silencio determinado.

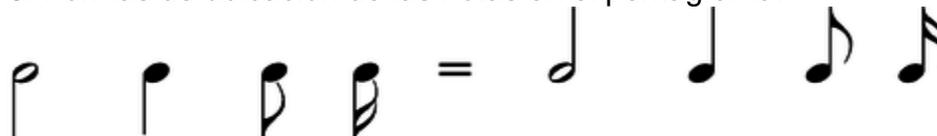
Cuadro 1. Símbolos y Nombres

NOMBRE	FIGURA	SILENCIO	TIEMPO RELATIVO
REDONDA			4
BLANCA			2
NEGRA			1
CORCHEA			1/2
SEMICORCHEA			1/4

Fuente: de los autores.

Como se observa en el Cuadro 1 la figura musical indica su duración, además estas figuras pueden cambiar su orientación dependiendo la posición de la cabeza de la figura. Por ejemplo, si la cabeza se encuentra por encima de la línea central del pentagrama, la plica se orienta hacia abajo desde el lado izquierdo de la cabeza y si es el caso contrario la plica va hacia arriba del lado derecho de la cabeza como se muestra en la Figura 10.

Figura 10. Formas de ubicación de las notas en el pentagrama.



Fuente de los autores.

Otra forma en que las notas pueden variar su representación, es cuando varias semicorcheas o figuras como corcheas, fusas y semifusas se encuentran de forma consecutiva y se encuentran dentro del mismo compas², sus corchetes se unen convirtiéndose en una o mas barras gruesas que pueden o no ser horizontales según la dirección general de las notas a unir, como se observa en la Figura 11.

Figura 11. Representación de notas agrupadas.



Fuente de los autores.

LIGADURA: es una línea curva que une a dos simbolos continuos de la misma nota, haciendo que el sonido del segundo simbolo sea la continuacion del primero.

Figura 12. Ligaduras.



Fuente de los autores.

²“Compas” [en línea], disponible en: [http://es.wikipedia.org/wiki/Comp%C3%A1s_\(m%C3%BAsica\)](http://es.wikipedia.org/wiki/Comp%C3%A1s_(m%C3%BAsica)), recuperado: 10 de febrero de 2013.

PUNTILLO: se ubica a la derecha de la NOTA o SILENCIO y los prolonga a la mitad de su valor, por ejemplo.

Figura 13. El puntillo.



Fuente de los autores.

2.2.3.2 Figuras de alteración. Son los símbolos que se ubican en la parte izquierda de una nota (en la misma altura) y que modifican su nota en un semitono cromático³ de forma ascendente o descendente haciendo que suene la nota más aguda o más grave. El Cuadro 2, ilustra los principales símbolos de alteración:

Cuadro 2. Figuras de alteración.

SIMBOLO	NOMBRE	TIPO DE ALTERACION
#	Sostenido	Aumenta el sonido en un semitono cromático
b	Bemol	Reduce el sonido en un semitono cromático
♮	Becuardo	Cancela una alteración por lo que puede aumentar o reducir un sonido en un semitono cromático, según sea el caso

Fuente: de los autores

Existen más símbolos de alteración que son derivados de los anteriores y casi no son usados, el Cuadro 3 muestra otros símbolos de alteración.

³“Semitono cromático” [en línea], disponible en: http://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io2/public_html/escalas.html, recuperado: 11 de febrero de 2013.

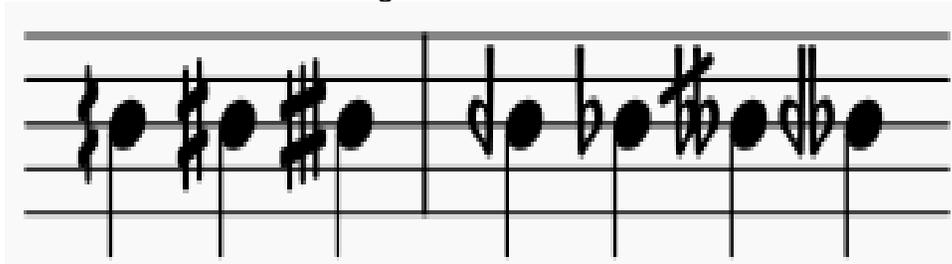
Cuadro 3. Otras figuras de alteración.

SIMBOLO	NOMBRE	TIPO DE ALTERACION
	Medio sostenido	Aumenta el sonido en medio semitono cromático
	Medio bemol	Reduce el sonido en medio semitono cromático
	Sostenido y medio	Aumenta el sonido en un semitono y medio cromático
	Bemol y medio	Reduce el sonido en un semitono y medio cromático
	Doble sostenido	Aumenta el sonido en dos semitonos cromáticos
	Doble bemol	Reduce el sonido en dos semitonos cromáticos

Fuente: de los autores

La Figura 14 ilustra la forma como se denotan estas figuras en un pentagrama:

Figura 14. Forma de denotar las figuras de alteración.

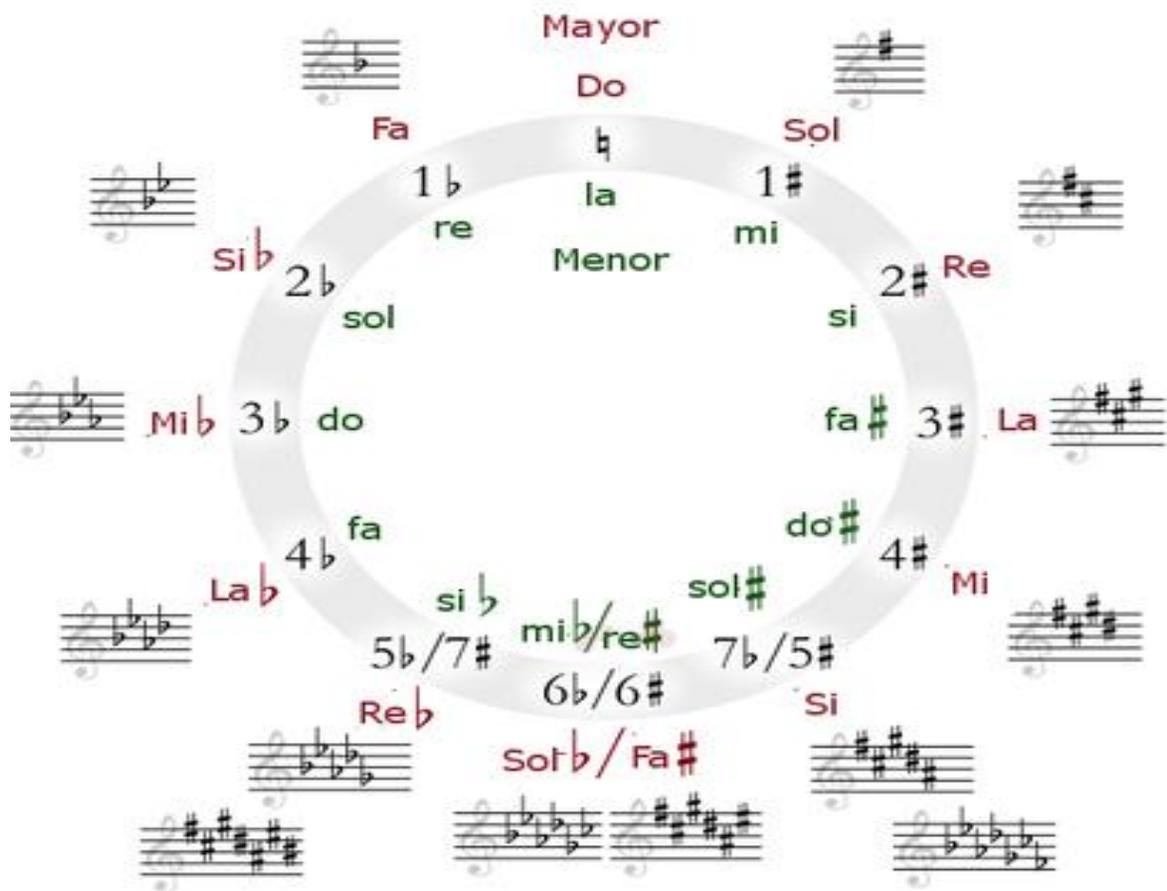


Fuente: [http://es.wikipedia.org/wiki/Tonalidad_\(m%C3%BAsica\)](http://es.wikipedia.org/wiki/Tonalidad_(m%C3%BAsica))

2.2.4 Escalas musicales. La escala musical es la sucesión de sonidos que por lo general se da de forma ascendente o descendente, con relacion a un solo tono. Las escalas se usan para aprender a armonizar, componer, e improvisar en cualquier instrumento. En la actualidad hay más de 500 escalas diferentes, de jazz, blues, pop, country, exóticas, etc. pero con sólo algunas ya es posible dominar gran parte de la música.

2.2.5 Tonalidad. Es la melodía que se encuentra escrita en una partitura principalmente por medio de las notas (sonidos naturales) y sus alteraciones (sostenidos, bemoles); la tonalidad se puede encontrar de dos formas, la primera y tal vez la más usada es cuando aparecen sostenidos o bemoles en la encabezadura del pentagrama las cuales definen la tonalidad general de la partitura y la segunda cuando aparece un sostenido o bemol en determinada nota la que cambia la tonalidad de esa nota a partir de ese momento hasta que se encuentre con un becuadro. En caso de no encontrarse un becuadro la nota guardara su tonalidad hasta el final de la partitura. La Figura 15 ilustra los diferentes tipos de tonalidades que hay.

Figura 15. Tonalidad



Fuente: http://upload.wikimedia.org/wikipedia/commons/thumb/5/5b/Circle_of_fifths_deluxe_4-ES.png/450px-Circle_of_fifths_deluxe_4-ES.png

2.3 RECONOCIMIENTO ÓPTICO DE CARACTERES MUSICALES

El ser humano a través de la historia ha venido implementando con mayor frecuencia máquinas para facilitar las tareas del día a día, es así como empieza el hombre a introducirse en el campo de la visión artificial. Las primeras investigaciones en el campo no llegaron sino hasta finales de los años 50, desde aquel entonces se han desarrollado sistemas de visión por computador más complejos y algoritmos aplicados a diversas áreas como, detección remota, seguridad, robótica, automatización industrial, análisis de imágenes, reconocimiento de objetos y caracteres, entre otros.

El campo de reconocimiento de objetos, caracteres o símbolos es relativamente amplio, ya que se puede enfocar los sistemas de reconocimiento hacia cualquier cosa. Los sistemas se han venido desarrollando para detectar y reconocer símbolos bajo ciertas condiciones, en el caso del reconocimiento de caracteres generalmente se requiere la identificación directa del objeto, mientras que los caracteres musicales plantean un problema de reconocimiento en un espacio bidimensional. Por una parte, la dirección horizontal de dicho espacio puede estar asociada con la duración temporal de la nota, mientras que la dirección vertical puede estar asociada con el tono.⁴

Otra complicación en la música es que los símbolos en ocasiones se superponen en el pentagrama, cosa que hace difícil aislarlos, además los símbolos están formados por componentes que pueden combinarse entre ellos de distintas formas y algunos símbolos pueden escribirse de diferentes formas. Esto aumenta la complejidad del OMR, obligando a que se incluya muchas normas sobre lo que se puede encontrar en la notación musical. Hay sistemas OMR que tienen prefijadas estas normas y otros en los que pueden ser definidas o adaptadas por el usuario, aportando así mucha flexibilidad.

Debido a los problemas que afronta el reconocimiento de caracteres musicales, se hace necesario una nueva rama o extensión del OCR (Reconocimiento Óptico de Caracteres) conocida como OMR (Reconocimiento óptico musical).

Utilizando estos sistemas se puede, a partir de una partitura digital obtener un archivo que contenga información sobre esa partitura, lo que permite después reproducirlo en una computadora, cambiar la afinación a la que se debe tocar la canción automáticamente o realizar hacer cualquier otra modificación con facilidad gracias a programas especiales para la edición de partituras o simplemente almacenarlas e imprimirlas en cualquier momento.

⁴“OMR” [en línea], disponible en: http://www.informatics.indiana.edu/donbyrd/N560Site-Fall06/B+B_ChallengeOfOMR.pdf, recuperado: 11 de febrero de 2013.

2.4 TRABAJO DE OTROS AUTORES

Hasta hace poco la investigación basada en el reconocimiento de música impresa, se ha limitado a los trabajos de Pruslin y Prearau, dos trabajadores del instituto tecnológico de Massachusetts, sus trabajos fueron el punto de partida para muchos otros autores.

Desde entonces se han desarrollado diversos sistemas, como por ejemplo, se diseñó un sistema para el robot musical WABOT-2, que facilitaba la reproducción de una partitura. En 1978, Nakamura, desarrollo un sistema para analizar canciones del folclore japonés. Por otro lado en 1982, Aoyama y Tojo, diseñaron un sistema de lectura de partituras, para almacenar en una base de datos.⁵

Como se puede observar, aunque todos los trabajos traten sobre el reconocimiento óptico de música, cada uno se enfoca en forma distinta para conseguir un objetivo determinado.

Los autores enfrentaron el problema del reconocimiento de distintas formas. Algunos se centraron en reconocer una cantidad limitada de símbolos, para aplicaciones de ámbito general, mientras que otros se limitaron al problema a bajo nivel es decir, reconociendo los componentes de los símbolos.

A continuación se describen algunos de los trabajos desarrollados en el área de reconocimiento de partituras musicales.

2.4.1 Aoyama Y Tojo (1982). Estos dos autores propusieron un sistema dividido en tres etapas: Entrada, Segmentación y Reconocimiento. En la primera etapa, se binariza la imagen y se encuentra las líneas del pentagrama, luego se halla la altura entre las líneas y la anchura de las mismas. En la segunda etapa, se segmentan los símbolos usando componentes conectados. En la última etapa los símbolos se segmentan y se clasifican.⁶

Se asume que la entrada es una partitura limpia, es decir libre de símbolos rotos, pero permitiendo cualquier tamaño. El sistema remueve la mayoría de las líneas del pentagrama, pero para evitar una segmentación excesiva y que se pierda información relevante, se conservan las líneas cercanas al área de los símbolos, como se muestra en la Figura 16.

⁵ REVIEW OF OMR RESEARCH. [en línea]. Disponible en:http://evirtual.lasalle.edu.co/info_basica/nuevos/guia/GuiaClaseNo.3.pdf, recuperado: 8 de febrero de 2013. p. 7.

⁶ Ibíd, p. 8-9.

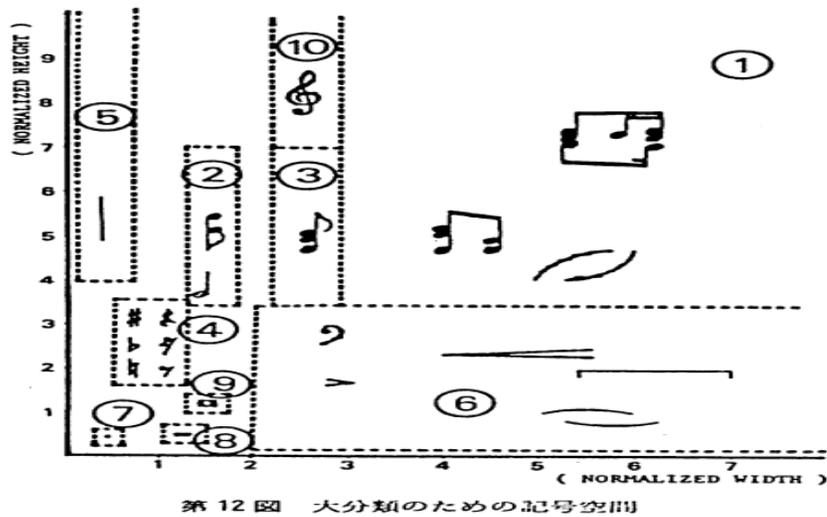
Figura 16. Imagen después de una burda eliminación de las líneas del pentagrama.



Fuente: <http://www.music.mcgill.ca/~ich/research/diss/2.%20Reviews.pdf>.

Luego se buscan las cabezas de notas con una plantilla para extraerlas temporalmente. Esto tiene como objetivo principal encontrar huecos dentro de las notas, una vez encontrados los símbolos se marcan, de forma que al eliminar el resto de líneas estos no sufran una fragmentación. La imagen resultante se segmenta por análisis de componentes conectados, para etiquetar los componentes en diez grupos como se muestra en la Figura 17.

Figura 17. Clasificación.



Fuente: <http://www.music.mcgill.ca/~ich/research/diss/2.%20Reviews.pdf>.

En el grupo con notas y uniones de notas, corchetes, entre otros, se extrae la cabeza de la nota. Luego, se extraen características como ancho, alto, centroide y área, estos parámetros servirán para que el clasificador cumpla su función.

2.4.2 Maenaka Y Tadokuro. El algoritmo usado por estos autores para detectar una figura musical la entre dos líneas, se basó en escanear de izquierda a derecha la cabeza de la figura musical como se observa en la Figura 18, de este modo se puede verificar si hay un cambio de color del pixel negro a blanco y viceversa en la cabeza de las figuras musicales y de esta forma saber si trata de una negra o blanca. En el caso que la figura musical este sobre una línea del pentagrama se verifica si hay más de un cambio color de pixel.⁵

Figura 18. Escaneo para determinar tipo de figura musical.

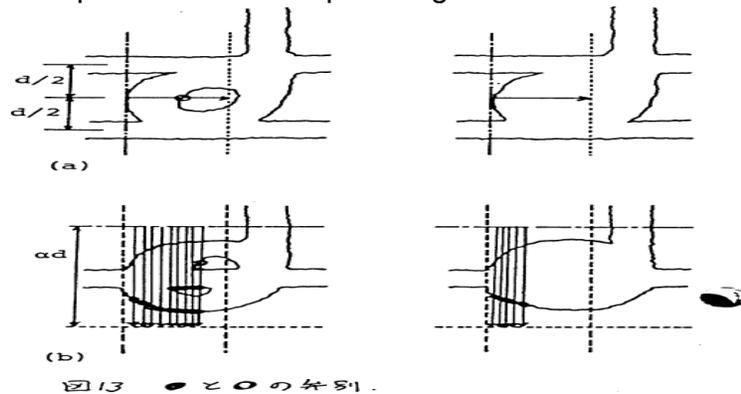
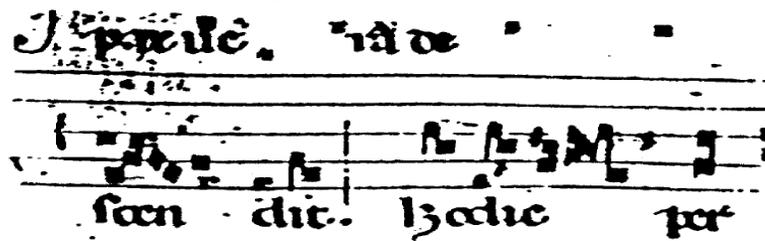


図13 ●と○の判別.

Fuente: <http://www.music.mcgill.ca/~ich/research/diss/2.%20Reviews.pdf>

2.4.3 Mcgee Y Merkley. El sujeto de esta investigación fue el reconocimiento de notas musicales cuadradas procedentes de notaciones musicales antiguas (ver Figura 19). Para eliminar las líneas del pentagrama, se buscan líneas horizontales delgadas y de una determinada longitud. Luego son enderezadas y clasificadas usando un conjunto de rectángulos que abarcan el área de una nota.⁷

Figura 19. Notación musical antigua.



Fuente: <http://www.music.mcgill.ca/~ich/research/diss/2.%20Reviews.pdf>.

⁷ REVIEW OF OMR RESEARCH. [en línea]. Disponible en: http://evirtual.lasalle.edu.co/i/nfo_basica/nuevos/guia/GuiaClaseNo.3.pdf, recuperado: 8 de febrero de 2013. p. 19-21.

2.4.4 Etienne Sicard. El autor utiliza la proyección en el eje Y para la detección de líneas del pentagrama, falla si la partitura esta inclinada más de 10 grados. Luego se eliminan las líneas del pentagrama y son extraídas. Diferentes algoritmos fueron desarrollados para distinto símbolos, por ejemplo para las líneas verticales como compases se usa la proyección en el eje X para su detección. Otros algoritmos como detección de bordes y diámetro son usadas para la identificación.⁶

2.4.5 Miyao. En este trabajo se hizo énfasis en dos puntos, el primero en la incorporación de una gramática musical, para ayudar al reconocimiento, y el segundo en las líneas del pentagrama que son eliminadas después de extraer los símbolos. El tamaño de los símbolos va acorde a la altura del espacio entre las líneas del pentagrama. Las notas son buscadas y removidas, luego se eliminan las líneas del pentagrama, los símbolos restantes son toscamente agrupados, por tamaño y posición, los símbolos son clasificados utilizando características estructurales o mediante la comparación con ejemplos. Los símbolos candidatos a notas encontrados fuera del pentagrama se verifican mediante la búsqueda de líneas adicionales. Si no se encuentran estas líneas, el símbolo es rechazado. Para ello, las líneas se buscan mirando a los bordes de la izquierda y de la derecha del símbolo. Este sistema llega a conseguir una eficacia de alrededor del 93% con un tiempo de ejecución de 20 minutos por página aproximadamente.⁶

2.5 TÉCNICAS USADAS EN EL PROCESAMIENTO DE IMÁGENES

Un sistema de reconocimiento está formado generalmente por procesos de obtención, caracterización, interpretación de información de imágenes. Estos procesos están divididos en cinco áreas y agrupados acorde a su complicación. Debido a que este proyecto realiza procesos de interpretación se encuentra en el nivel alto como se muestra en el Cuadro 4.

Cuadro 4. Nivel del proceso.

PROCESOS.	NIVEL
Preprocesamiento	Bajo
Segmentación	Medio
Descripción	
Reconocimiento	
Interpretación.	Alto

Fuente: los autores.

⁶REVIEW OF OMR RESEARCH. [en línea]. Disponible en:http://evirtual.lasalle.edu.co/info_basica/nuevos/guia/GuiaClaseNo.3.pdf, recuperado: 8 de febrero de 2013. p. 21.

2.5.1 Métodos de preprocesamiento. Antes de empezar a trabajar con la imagen, esta debe ser pasada por una serie de etapas con el fin de dejarla lo mejor posible para ser analizada. Estas etapas incluyen un filtrado, corrección de la orientación, eliminación de zonas de no interés, entre otras.⁹

Filtrado de una imagen.

El objetivo principal de las técnicas de filtrado es obtener, a partir de la imagen original, una cuyo resultado sea el más adecuado para ser procesada, acorde a una aplicación específica mejorando características de interés. Los filtros más comunes son usados para:

- Suavizar la imagen. Reduce la variación de intensidad en los píxeles vecinos.
- Eliminación de ruido. Busca eliminar los cambios bruscos, de valores o intensidad de píxeles vecinos.
- Realzar Bordes. Como su nombre lo indica, pronuncia los bordes en una imagen.
- Detección de Bordes. Encuentra los píxeles que tiene un cambio brusco en la función de intensidad.

2.5.2 Segmentación. Es el proceso de dividir una imagen en regiones o partes, asignándoles una etiqueta a los píxeles que comparten ciertas características, y que se puede identificar como un objeto dentro de la imagen.¹⁰

Este algoritmo puede trabajar en imágenes binarias o en formato de escala de grises.

Para poder determinar esas componentes conexas hay que hacer uso de algoritmos básicos, que analizan la vecindad del píxel (Vecindad en ocho o Vecindad en cuatro).

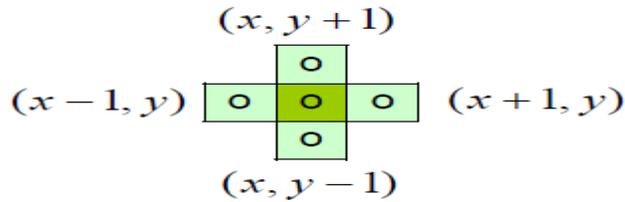
Vecindad de un Pixel.

Se llaman píxeles vecinos a aquellos que rodean al píxel de interés. En un píxel de coordenadas (x,y) , se ubican dos píxeles vecinos horizontalmente y otros dos verticales, a esto se le conoce como vecindad cuatro como se ve en la Figura 20.

⁹ PEREZ, JOSE IGNACIO. Segmentación y Búsqueda de Patrones en Partituras Musicales. Proyecto fin de carrera. Madrid, España: Universidad Rey Juan Carlos. Ingeniería Técnica en Informática de Sistemas. 2005, p. 7.

¹⁰ *Ibíd*, p 10-26.

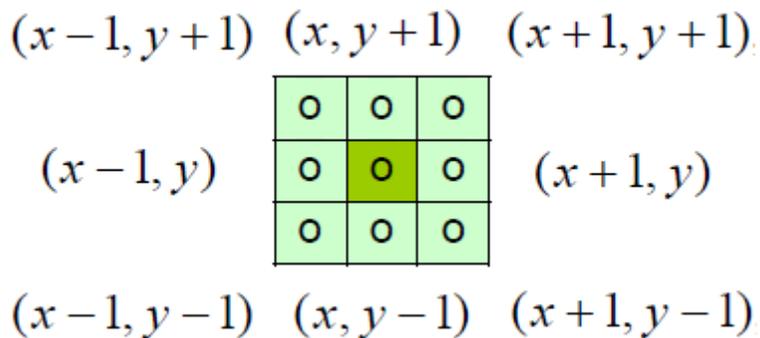
Figura 20. Vecindad de cuatro.



Fuente de los autores.

Ahora, si se tienen en cuenta los píxeles diagonales al píxel de interés, se encuentran otros cuatros que junto a los anteriores forman la vecindad de ocho como se observa en la Figura 21.

Figura 21. Vecindad de ocho.



Fuente de los autores.

Conociendo estos datos, se puede establecer distintas relaciones de conectividad.

Conectividad.

Para saber si dos píxeles están conectados, solo debe conocerse si pertenecen a alguno de los conjuntos, vecindad de cuatro u ocho, y si sus niveles de grises satisfacen un criterio de similaridad. En el caso de imágenes binarias con píxeles de valores de 0 o 1, por ejemplo, dos píxeles pueden ser vecinos, pero se dicen que están conectados solo si tienen el mismo valor.

3. DESARROLLO Y DESCRIPCIÓN DEL SISTEMA

Una vez definido los obstáculos del reconocimiento óptico de caracteres musicales y ver algunos métodos para resolverlos, en este capítulo se describe el programa desarrollado, así como su estructura, las decisiones al momento del diseño y las funciones desarrolladas para lograr los objetivos.

3.1 FUNCIONALIDAD

El sistema desarrollado es capaz de analizar los elementos de partituras no manuscritas. Debido a que es un proyecto de fin de carrera no cuenta con un grado de desarrollo equivalente al de un proyecto de tipo comercial. Esto implica ciertas limitaciones. A continuación se describirá lo que es capaz de reconocer el programa.

- Partituras de uno o más pentagramas.
- Símbolos no manuscritos.
- Claves de SOL y FA.
- Notas musicales como: Redonda, blanca, Negra, Corchea (unidas hasta cuatro), Semicorchea (Unidas hasta cuatro), Unión de corchea y semicorchea, identificación del separador, Puntillo, Silencios y Alteraciones.
- Capacidad para cambiar el tiempo de las figuras musicales de la partitura.
- Guardar símbolos u objetos no identificados para ser agregados luego a la base de datos.
- Notas cuya ubicación sea de hasta tres líneas adicionales.

Por otra parte, las limitaciones son.

- No acepta partituras de piano.
- No se permite notación musical distinta a la notación moderna (A no ser que se agregue posteriormente a la base de datos para ser identificada como tal).
- Se rechaza todo aquello que no sea pentagramas, como títulos o nombre de las partituras.

- Partituras rotadas máximo entre 1 y 1.5 grados.

3.2 DESCRIPCIÓN GENERAL DEL SISTEMA

Figura 22. Esquema general del sistema.



Fuente de los autores.

La figura 22 muestra el esquema general del sistema, el cual consiste en cinco bloques. El bloque de preprocesamiento se encarga de limpiar y filtrar la imagen, seguidamente el bloque clasificador agrupa las figuras musicales según su clase en conjunto con el bloque analizador que se ocupa de identificar cada elemento con su respectiva ubicación y nota correspondiente, después el bloque aplicador de atributos se encarga de las alteraciones de las notas, continuamente el bloque generador de audio permite reproducir y guardar el audio generado en formato WAV. Y el último bloque Selector de Figuras permite reproducir sectores de la partitura y permite cambiar algunas características de la partitura como tiempo de duración, clave y tonalidad.

A continuación se describirá detalladamente cada bloque y las decisiones de diseño aplicadas.

3.3 BLOQUE DE PREPROCESAMIENTO

Este bloque se encarga de adaptar las características significativas en información de la imagen de la partitura para el óptimo funcionamiento de los siguientes bloques. El bloque de Preprocesamiento trabaja con una imagen cargada como se muestra en la Figura 24 para continuar con el proceso o dar paso a las funciones internas del bloque como se observa en la Figura 23.

Figura 23. Bloque Preprocesamiento.



Fuente de los autores.

Figura 24. Imagen o Partitura.



Fuente: <http://www.teoria.com/aprendizaje/escalas/02-tonos-semitonos.php>

3.3.1 Conversión Escala de Grises, Filtrado y Conversión a Binario.

Inmediatamente la imagen se convierte en escala de grises para aplicar inicialmente el filtro “Gaussian”, para corregir ruidos, suavizando la imagen y el filtro “unsharp” para evitar desenfoques de la imagen, en pocas palabras realza la información de interés.¹¹ Seguidamente la imagen se pasa de escala de grises a un imagen binaria (negro y blanco), lo que sirve como un tercer filtro para descartar gran cantidad de información como se ve en la Figura 25.

Figura 25. Partitura filtrada y binarizada.



Fuente: <http://www.teoria.com/aprendizaje/escalas/02-tonos-semitonos.php>

3.3.2 Detección y corrección de líneas horizontales. Este paso busca la detección y corrección de las líneas de los pentagramas con ángulos diferentes a cero, dado que si no se parte de una imagen con sus pentagramas bien definidos sería muy poco probable reconocer las notas musicales debido a que los pentagramas alterarían su forma y el espacio que estas ocuparían. Para esto se elimina todo tipo de notación en la partitura diferente a las líneas de los pentagramas, esto se logra eliminando las partes en donde se encuentren más de tres píxeles negros unidos de forma vertical, dejando una imagen con líneas horizontales similar a la Figura 30, seguidamente se niega la imagen binaria y se etiqueta las líneas con la función de Matlab “bwlabel”.¹²

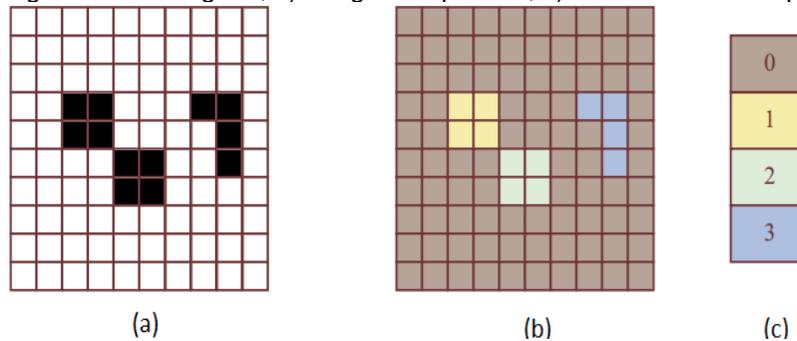
¹¹GONZALES AGUILERA, DIEGO. Procesamiento de Imágenes. Universidad de Salamanca. Master De Geotecnologías Cartográficas En Ingeniería Y Arquitectura. Castilla y León. 2008, p 3-12.

¹² Ibíd, p 19-23.

La función `bwlabel`, retorna una matriz o imagen con las etiquetas, como se puede observar en la figura 26.

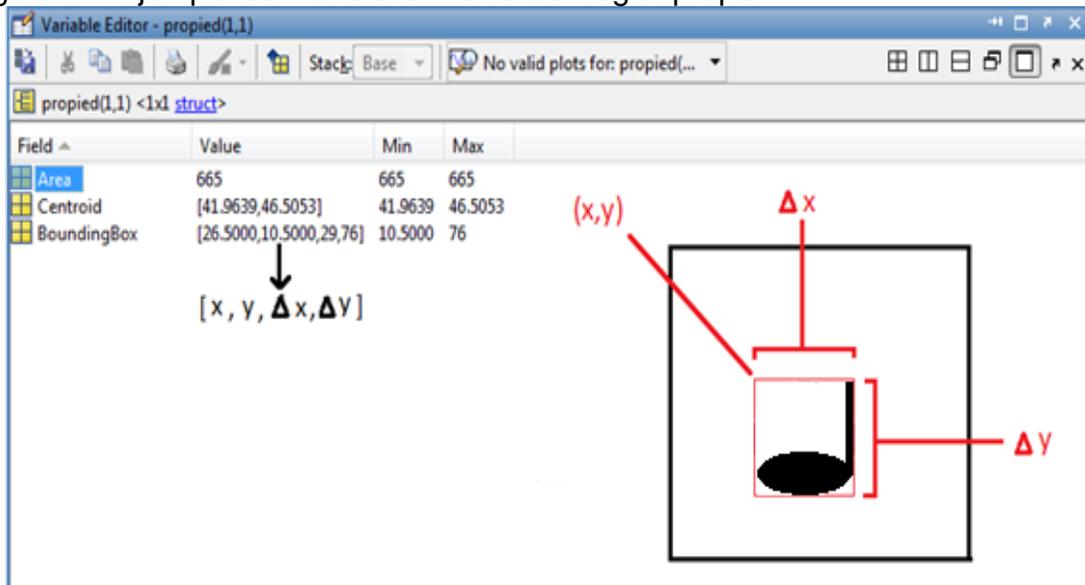
Ahora se necesita saber las propiedades de todos los elementos (líneas), esto con el fin de conocer parámetros como la ubicación en el eje Y, centroide y áreas; para realizar lo anterior se usa la función “`regionprops`”¹³, que recibe como parámetro de entrada la imagen etiquetada y devuelve un array de estructuras, tantas como la cantidad de objetos que fueron identificados, donde cada estructura contiene las propiedades de Área, Centro y Boundingbox (Cajón Imaginario), como se muestra en la figura 27.

Figura 26. a) Imagen Binaria Negada, b) Imagen Etiquetada, c) Codificación o Etiqueta Asignada.



Fuente de los autores.

Figura 27. Ejemplo del Funcionamiento de Regionprops.



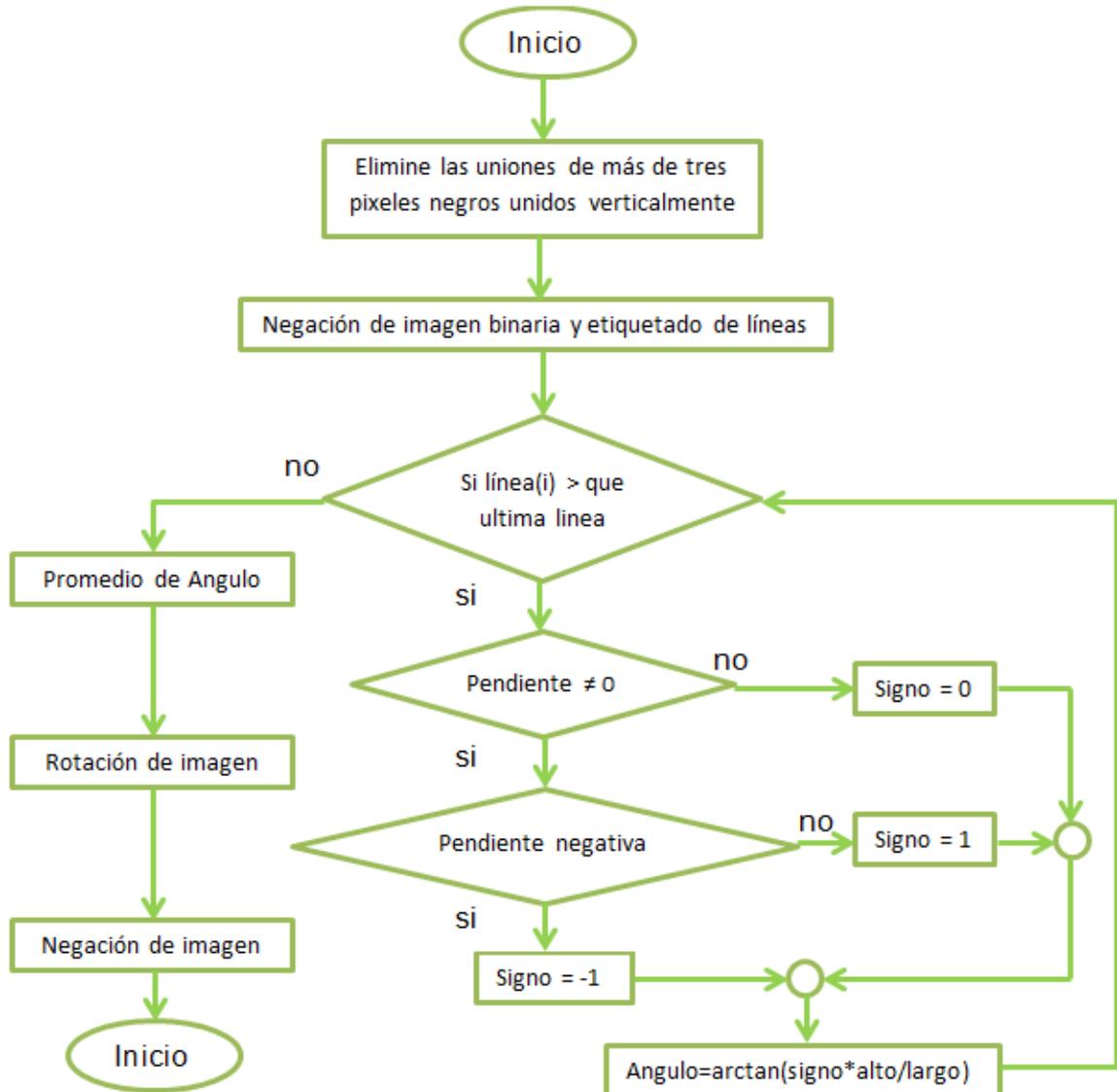
Fuente de los autores.

¹³“Regionprops”. [en línea]. Disponible en: <http://www.mathworks.com/help/images/ref/regionprops.html>, recuperado: 15 de febrero de 2013.

Algoritmo para detección y corrección del ángulo.

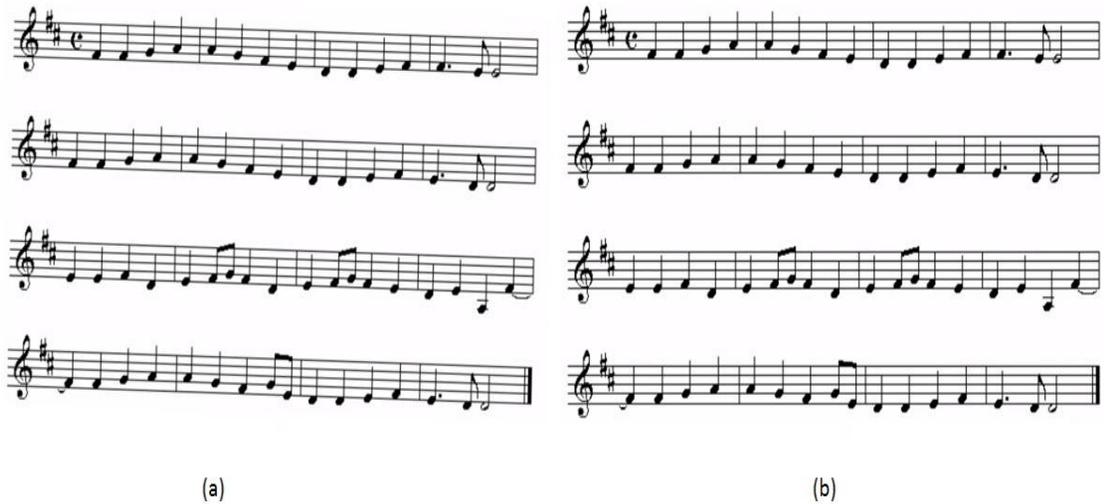
Por medio de la propiedad del “BoundingBox”, se verifica los valores del recuadro tales como el alto y ancho para conocer el ángulo poseen las líneas y también que valores tienen las esquinas el recuadro para conocer la orientación de estas. El algoritmo mostrado en la Figura 28, hace un representación del proceso que realiza en esta etapa. La Figura 29, ilustra el resultado final de una partitura inclinada y otra partitura con la inclinación corregida.

Figura 28. Algoritmo corrección de ángulo.



Fuente de los autores.

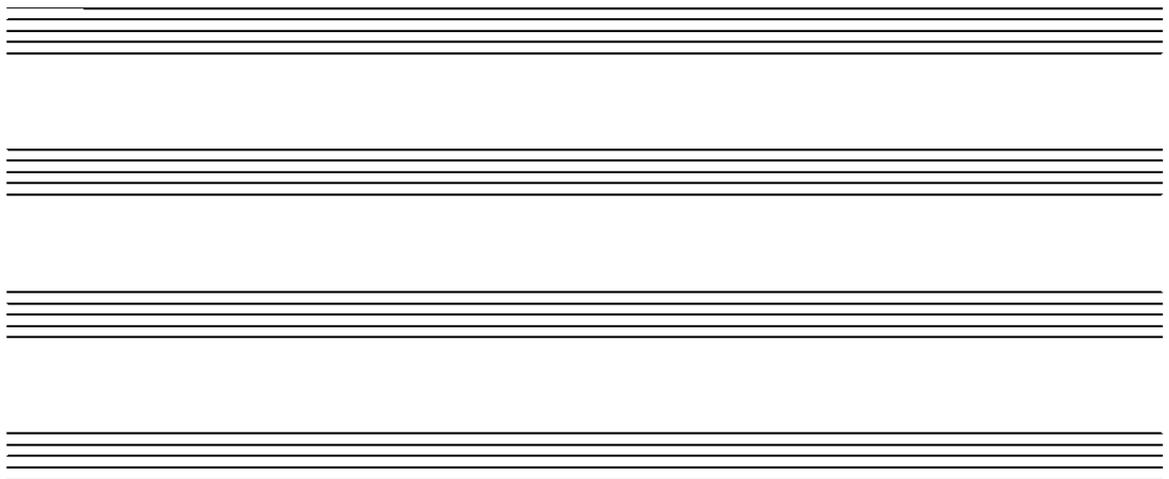
Figura 29. a) Imagen con ángulo de inclinación, b) Imagen con corrección de ángulo.



Fuente: <http://www.teoria.com/aprendizaje/escalas/02-tonos-semitonos.php>

3.3.3 Extracción de características del pentagrama. Para este paso es necesario concentrarse exclusivamente en los pentagramas debido a que hay una buena parte de información inscrita intrínsecamente en ellos. Inicia de la imagen empleada por la Figura 30 que entrega la etapa anterior con pentagramas bien definidos.

Figura 30. Pentagramas sin ruido.

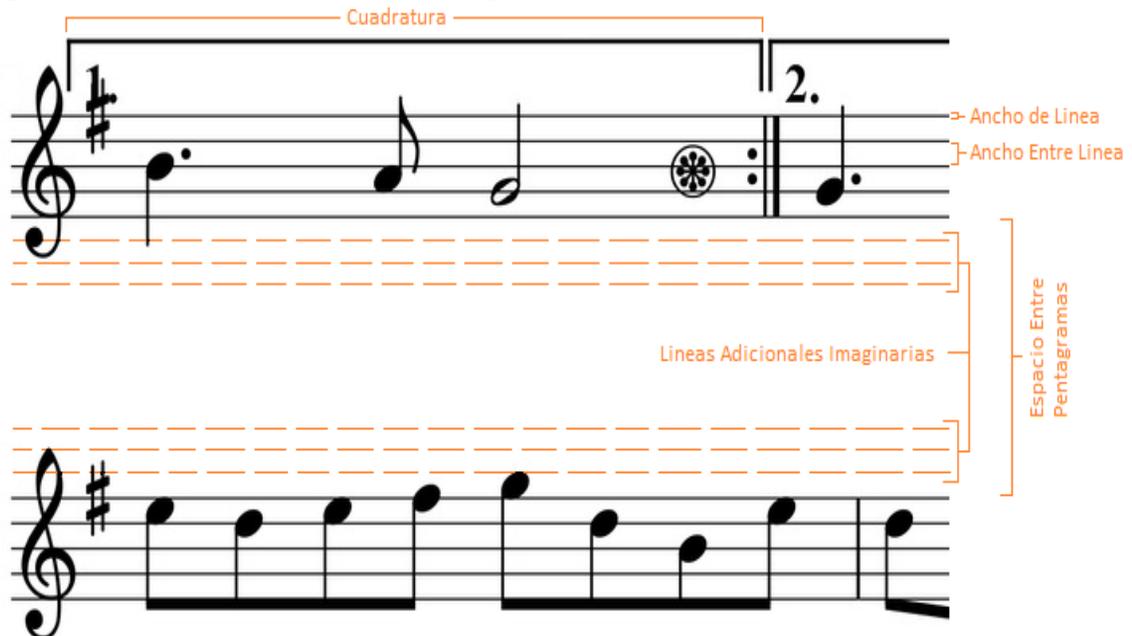


Fuente de los autores.

Ahora se procede a encontrar el ancho de cada línea de los pentagramas, por ende se realizan barridos verticales en diferentes puntos de la Figura 30 buscando cuantos pixeles negros se encuentran unidos, los resultados obtenidos se guardan en un vector y se verifica el número que más se repite, hallando de esa forma el ancho que se utilizara para todas las líneas, continuo a este proceso se encuentran las ubicaciones de las líneas de los pentagramas en la partitura de forma similar a la anterior pero ahora se guardan las posiciones de los pixeles negros en un vector. Una vez obtenidos el ancho de las líneas y las ubicaciones de estas, se puede encontrar el ancho entre líneas. Estas y otras características del pentagrama se pueden observar en la Figura 31.

Es posible que aun existan algunas líneas falsas como las cuadraturas que son líneas horizontales muy prolongadas, como se conoce el ancho entre líneas de pentagrama, se halla la diferencia de la distancia entre la línea siguiente y la presente, si es igual al ancho entre líneas entonces la línea pertenece al pentagrama, de lo contrario significa que es una cuadratura y se elimina, este proceso se repite para todos los pentagramas. Ahora se puede saber con seguridad cuántos pentagramas hay en la partitura.

Figura 31. Características del pentagrama.



Fuente de los autores.

Por último en esta etapa de preprocesamiento, se agregan las tres líneas adicionales imaginarias del pentagrama por encima y por debajo para las notas más agudas y más graves respectivamente. Esta parte se realiza si la imagen

correspondiente a la partitura tiene el espacio suficiente entre pentagramas para introducir las líneas adicionales imaginarias.

Eliminación de las líneas de los pentagramas.

Una vez obtenidas todas las características de los pentagramas, se continúa con la eliminación de las líneas de los pentagramas teniendo en cuenta varias condiciones para no eliminar píxeles de la imagen (figuras) que aportan información relevante para su posterior reconocimiento. Esto le agrega un grado más dificultad, debido a que todas las figuras musicales están inscritas entre las líneas de los pentagramas.

Para no afectar la forma de las notas al eliminar las líneas horizontales de los pentagramas, se procede de la siguiente manera.

Primero, como ya se conoce la posición de las líneas, se recorre pixel a pixel por la línea evaluando un pixel por encima y por debajo del pixel de interés de tal forma que si encuentra un punto blanco arriba o debajo de él, convierte este pixel en blanco. Esto para el caso idóneo, en que el ancho de línea sea de un pixel.

Segundo, sigue el mismo proceso con la particularidad de evaluar el pixel por encima y abajo teniendo en cuenta el ancho de la línea que se obtuvo en la extracción de características del pentagrama.

Se debe tener en cuenta al momento de evaluar un pixel si hace parte de una figura (blanca y redonda) para no eliminarlo debido a que puede fragmentar la figura musical, entorpeciendo el siguiente proceso que es la segmentación por componentes conexos de las figuras.

Tercero, evalúa por encima y por abajo del pixel de interés, el equivalente al ancho de línea más un par de píxeles, si estos dos puntos son blancos entonces se elimina el ancho de línea alrededor del pixel.

Cuarto, se sigue el mismo procedimiento con la diferencia de evaluar los píxeles negros por encima y abajo, con el fin de detectar figuras como blancas y redondas.

Al realizar el proceso descrito, se obtiene una imagen como la mostrada en la Figura 33, en la cual se observan solo los símbolos musicales, que son los siguientes elementos de interés.

3.4 BLOQUE CLASIFICADOR

El bloque clasificador se encarga de seleccionar y agrupar según el tipo o clase de figura musical, la Figura 32 muestra las funciones empleadas en este bloque.

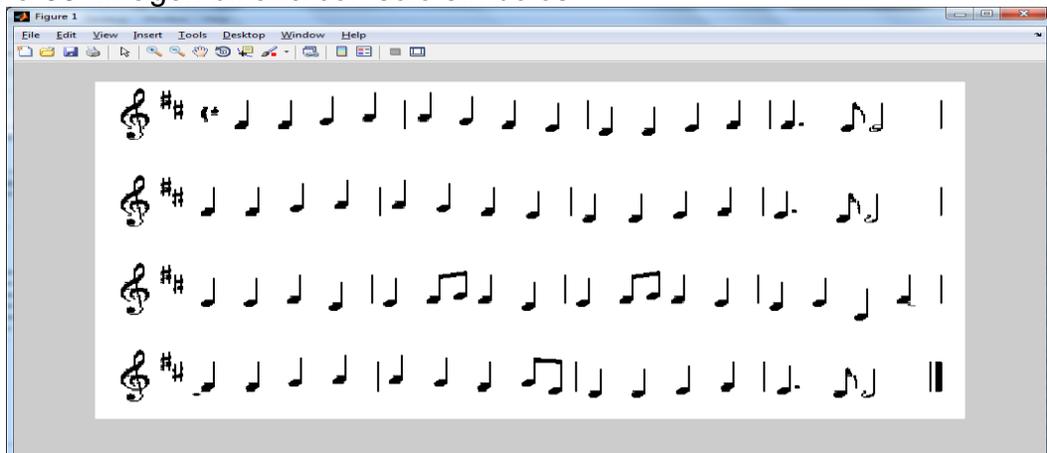
Figura 32. Diagrama de bloques de bloque clasificador.



Fuente de los autores.

3.4.1 Imagen con símbolos aislados. Una vez obtenidas las posiciones y eliminadas las líneas de los pentagramas, se tiene solo las figuras musicales como se ve en la Figura 33.

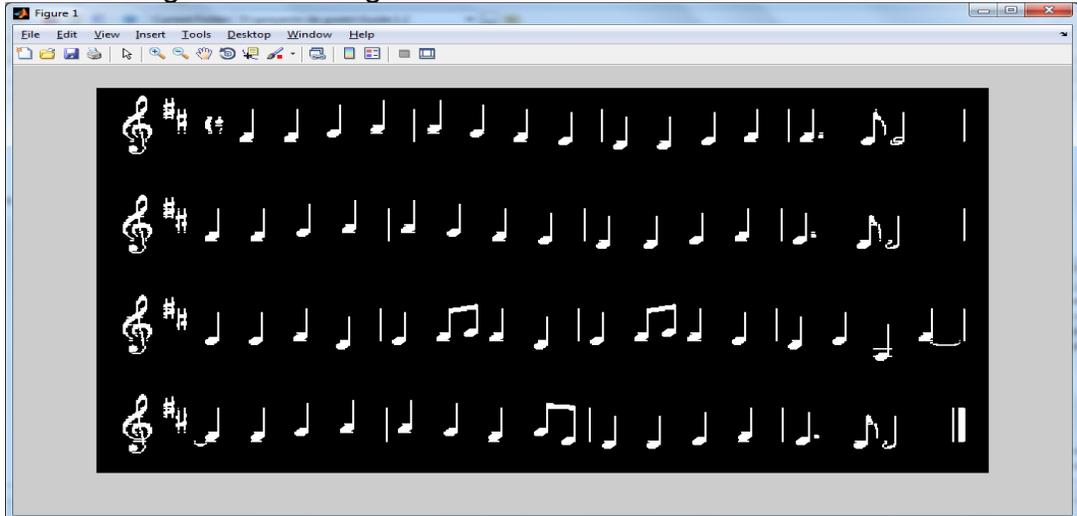
Figura 33. Imagen binaria con solo símbolos.



Fuente de los autores.

3.4.2 Imagen binaria negada. Ahora se procede a negar la imagen, con el fin de etiquetar los símbolos, es decir lo negro se vuelve blanco y viceversa como se muestra en Figura 34.

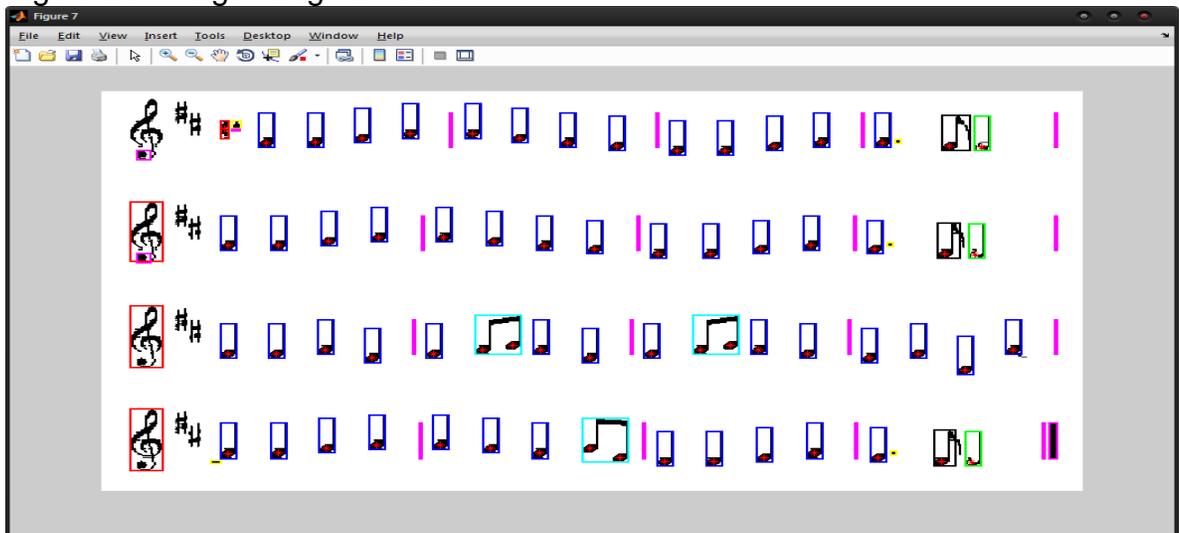
Figura 34. Imagen binaria negada.



Fuente de los autores.

3.4.3 Etiquetado. Se etiqueta la imagen que contiene solo los símbolos para poder analizar cada uno de ellos con las funciones "bwlabel" y "regionprops" logrando una segmentación como muestra la Figura 35.

Figura 35. Imagen segmentada.



Fuente de los autores.

3.4.4 Comparación. Una vez segmentadas las figuras musicales se toma una a una para compararla con cada figura musical que se encuentra almacenada en la base de datos, esta comparación se realiza hasta que encuentre un porcentaje de correlación superior al 85%, en caso de no superar el porcentaje se crea un vector con los porcentajes de correlación hallados para tomar el máximo valor de correlación y asignarlo al grupo de interés para ser analizado.

Correlación. La correlación se utiliza generalmente para indicar la correspondencia o la relación recíproca que se da entre dos o más cosas, en probabilidad y estadística indica la fuerza y la dirección lineal que se establece entre dos variables aleatorias.

Se considera que dos variables están correlacionadas cuando los valores de una de ellas varían sistemáticamente con respecto a los valores de la otra: si se tiene dos variables (A y B) existe correlación si al aumentar los valores de A lo hacen también los de B y viceversa.

Para calcular esto, se usa la función “corr2” de Matlab, que calcula el coeficiente mediante la siguiente ecuación donde m y n son las posiciones de los píxeles y A y B son los volares de los píxeles en dichas posiciones.

Ecuación 1. Coeficiente de Correlación

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2) * (\sum_m \sum_n (B_{mn} - \bar{B})^2)}}$$

Fuente. Matlab.

3.4.5 Asignación de Símbolos. Para identificar, se ha creado unos grupos para clasificar más rápidamente los símbolos de interés.

Los grupos son:

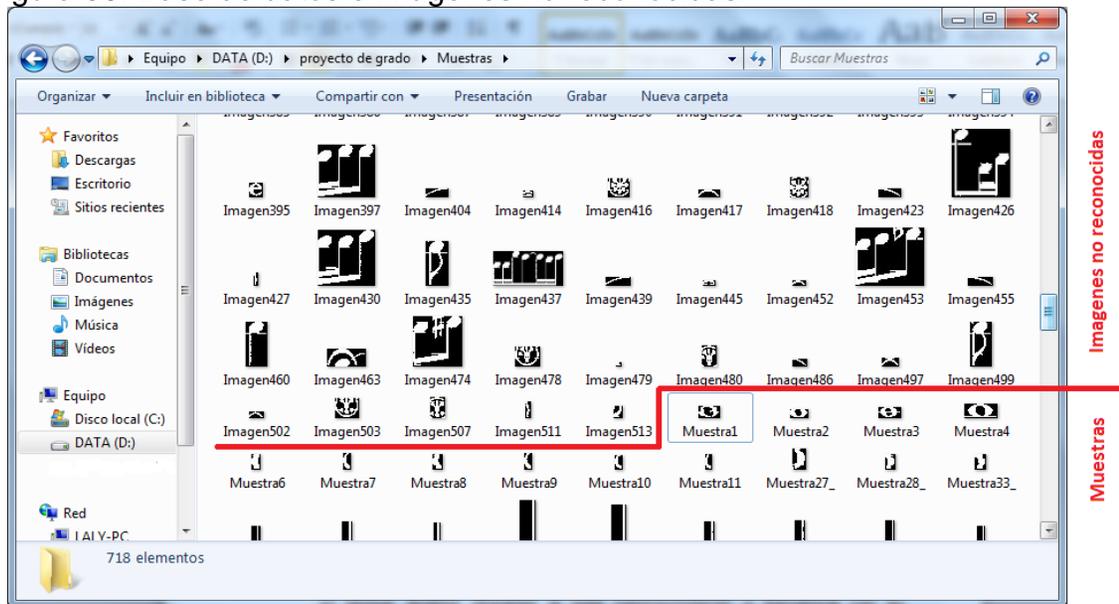
- Blancas.
- Negras.
- Redondas.
- Corcheas.
- Semicorcheas.
- Uniones Corchea-Semicorchea.

- Claves.
- Bemol-Becadro-Sostenido-Puntillo (Figuras de alteración).
- Silencios.
- Separador-Otros Símbolos.

Para asignar o conocer a que grupo pertenece, se halla el número de la muestra con la que presento el mayor coeficiente de correlación.

Con una serie de condicionales, se encuentra el rango en que esta la muestra en la base de datos, con que se encontró la mayor correlación al momento de comparar con el símbolo de interés. Por ejemplo, si se tiene mil muestras, las primeras 100 son muestras de la figura redonda de 101 a 200 son negras, las siguientes 100 serán blancas y así sucesivamente con diferentes figuras. De esta forma se asigna el símbolo de interés al grupo que pertenece para luego ser analizado, pero en caso que el símbolo no tenga al menos 68% de correlación con alguna muestra, se guarda la imagen del símbolo en la carpeta donde están las muestras con el nombre de Imagen (i).jpg como se ve en la Figura 36, debido a que si no supera ese porcentaje de correlación no se puede asegurar con certeza que la imagen pertenece a determinada figura musical, esto se hace con la finalidad de agregarla luego a la base de datos.

Figura 36. Base de datos e imágenes no reconocidas.



Fuente de los autores.

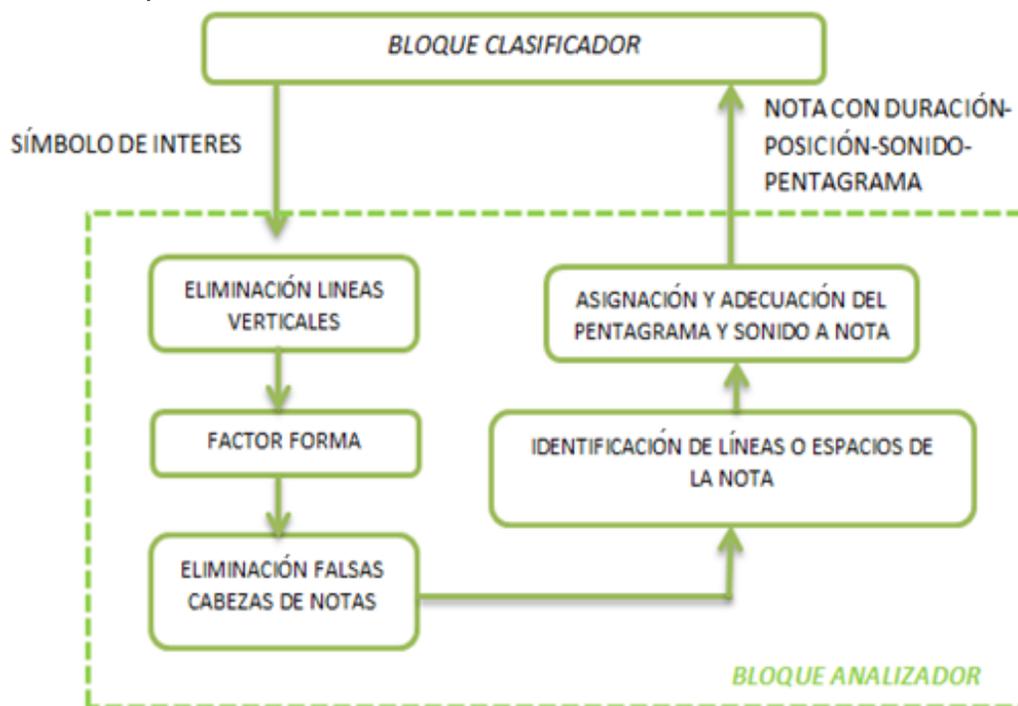
3.4.6 Símbolos aislados. Como se ha explicado en apartados anteriores, se ha de sacar y aislar el símbolo para ser analizado con la función desarrollada “Análisis”, que se explicara más adelante en el bloque Analizador.

3.4.7 Organización de símbolos y almacenamiento en estructura. El bloque Analizador devuelve la nota correspondiente o frecuencia con su posición, y el número del pentagrama al que corresponde. Es importante conocer estos valores: posición en x, duración, nota o sonido y pentagrama. La posición en x hace referencia al centroide de la figura dentro de la imagen que sirve para ubicar y organizar de forma horizontal las notas que se irán reproduciendo en orden. La duración es el tiempo que tiene la nota y es dada por el bloque “Asignación de símbolos”, como se sabe no todas las notas tienen el mismo valor. Una vez conocidos estos valores, se van almacenando en una variable de 2000 estructuras ya que se ha previsto que se tendrá un máximo de 2000 símbolos identificados por partitura.

3.5 BLOQUE ANALIZADOR

Este bloque se encarga de hallar a que notas pertenecen las figuras musicales halladas anteriormente, la Figura 37 ilustra las funciones del bloque.

Figura 37. Bloque analizador.

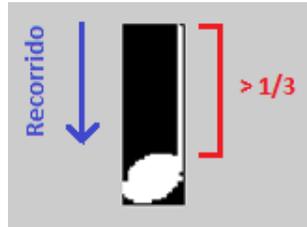


Fuente de los autores.

Como ya se ha clasificado el símbolo, es decir, se conoce la figura que representa, ahora es de interés ubicarla dentro del pentagrama, conocer su nota (Do, Re, Mi...), por lo cual se ha de concentrar en analizar las cabezas de nota de las figuras y se debe realizar un segundo “preprocesamiento” y “segmentación”, con el fin de trabajar solo con la cabeza de nota, el resto de información como la plica o el corchete se puede obviar o en otras palabras no es requerido.

3.5.1 Eliminación líneas verticales. Se recorre la imagen de la figura musical en sentido vertical y se encuentran aquellas líneas verticales (plica) mayores a un tercio de la altura total de la imagen y se eliminan, la Figura 38 muestra el recorrido y el tamaño de la plica en la imagen.

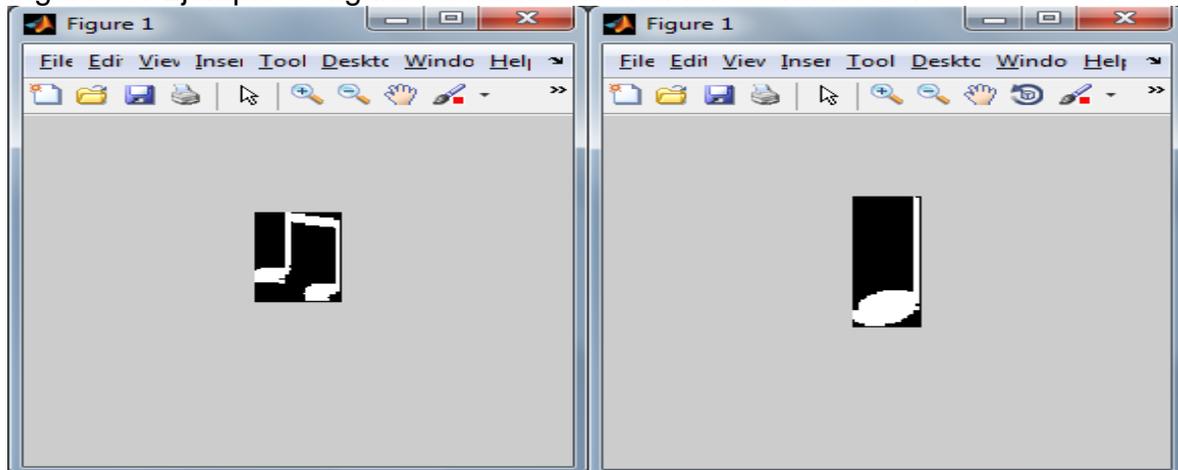
Figura 38. Recorrido y determinación de líneas a eliminar.



Fuente de los autores.

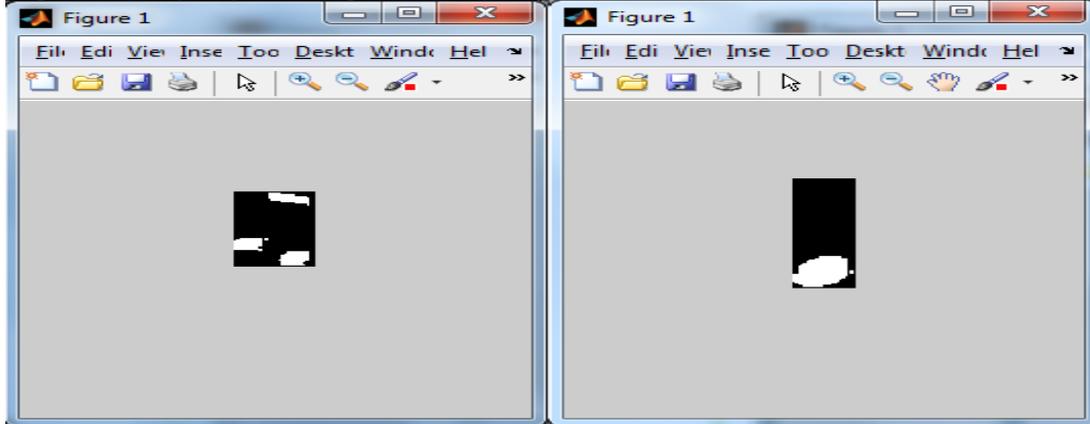
Se comprobó empíricamente con varias muestras de notas, que las líneas verticales o plicas siempre superan $1/3$ de la altura total de la imagen, de ahí su elección. La Figura 39 ilustra dos figuras musicales con sus respectivas plicas.

Figura 39. Ejemplo de figuras antes de eliminar líneas verticales.



Fuente de los autores.

Figura 40. Ejemplo de figuras después de eliminar líneas verticales.



Fuente de los autores.

En el caso del ejemplo, la corchea de la Figura 40 se le han eliminado las plicas pero la imagen resultante tiene una línea horizontal y dos cabezas de notas por lo cual es necesario realizar un filtrado para eliminar esa línea que no aporta información relevante y centrarse en la cabeza de nota. Antes de ir al siguiente paso, se ha de eliminar áreas menores a 5 píxeles, esto con el fin de borrar el ruido dejado al eliminar las líneas verticales o plicas como se puede observar en la Figura 40 han quedado unos pequeños puntos cerca a la cabeza de nota.

3.5.2 Factor forma y eliminación falsas cabezas de nota. Se ha denominado el factor forma a la división del ancho sobre el alto de la cabeza de nota. Con esta relación, se eliminará información no relevante para el análisis de la figura.

Figura 41. Factor Forma.



Fuente de los autores.

Como se puede observar en la Figura 41, el factor forma de las cabezas de nota esta alrededor de 1.12, y es lógico pensar que siempre estará alrededor de 1 debido a que la cabeza de nota tiende a ser redonda, por lo cual la medida del ancho y el alto será muy similar, mientras que las líneas o corchetes tienden a cero o mayores a 1.5. Luego de varios ensayos se escogió un rango entre 0.5 y

1.5 para excluir los elementos cuyos factores de forma estén fuera de este rango y de esta forma trabajar solo con las cabezas de notas.

3.5.3 Identificación de líneas o espacios de la figura. Se debe recordar que la posición de la cabeza de nota dentro del pentagrama indicara la nota a la que corresponde la figura a analizar, por lo cual es uno de los pasos más importantes o de mayor atención.

Por lo anterior se debe encontrar si la cabeza de nota se encuentra en una línea o en un espacio, además de saber el número de la línea o del espacio.

Una vez despejada la cabeza de nota es de interés trabajar con su centroide y de esta forma saber la ubicación de la figura musical en el pentagrama.

Primero se ubica el centroide, la posición Y, y se busca la posición de la línea de pentagrama que sea igual en posición del centroide de la figura.

En el caso que la figura no este sobre una línea, se empieza a indagar hacia abajo, pixel por pixel hasta encontrar la línea del pentagrama siguiente, teniendo en cuenta las líneas adicionales.

En el ejemplo de la Figura 42, se busca la posición de la línea siguiente al centroide de la cabeza de nota hasta encontrar que la siguiente línea está en el pixel 20.

Debido a todo el preprocesamiento y como se tiene en cuenta es el centroide de la cabeza de la figura musical, es muy poco probable que su centroide se encuentre ubicado sobre una línea del pentagrama, es decir, si la posición de la línea es 20 continuando con el ejemplo, probablemente la posición en Y del centroide sea 21, 22, o 19. Por ende se deben establecer dos umbrales para ubicar la figura en la línea o en el espacio, un umbral mínimo y uno máximo, como se muestra en la siguiente porción de código.

```
humbrala=floor(ancho/3.5);%%rango a
humbralb=floor(ancho/3.3)+ancho/2;%%rango b

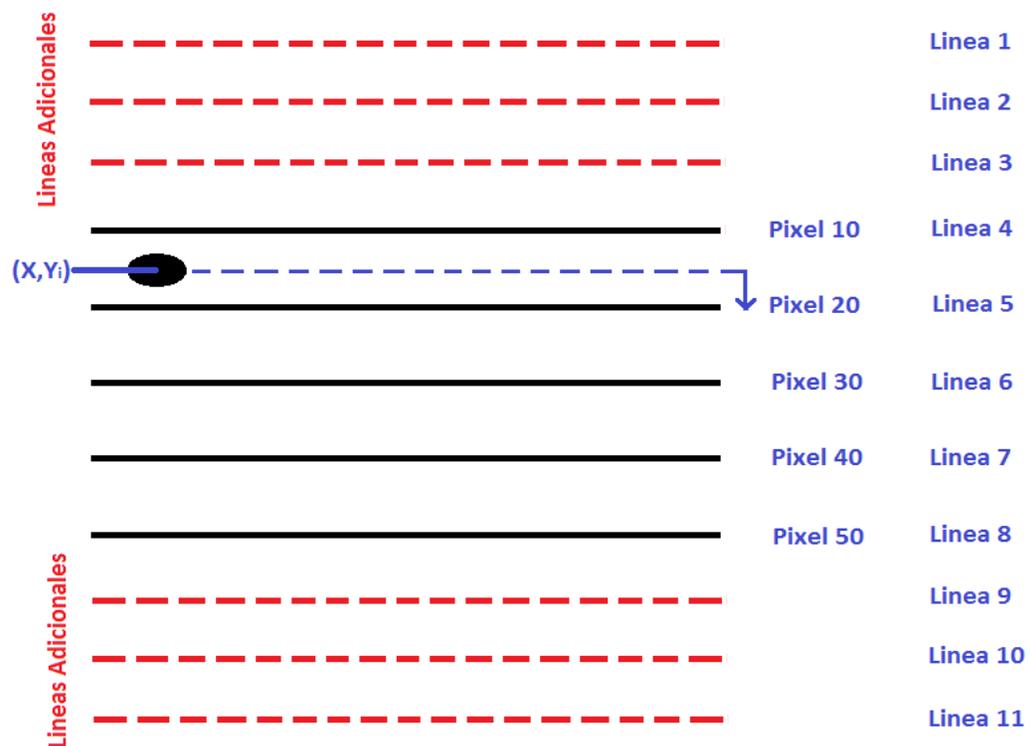
if dif<=humbrala
    sonido=linea; %si dif es menor a humbrala, entonces el sonido
es igual a la linea próxima encontrada

    elseif dif<=humbralb
        sonido=linea-0.5; %Si dif es menor al humbral b, el
sonido pertenece al espacio y no a la linea (-0.5)
    else
        sonido=linea-1; %De lo contrario sonido pertenece
a la linea anterior
end
```

En este caso la variable “dif” vista en código anterior, es el valor absoluto de la diferencia entre la posición en Y del centroide de la Figura 42 y la posición de la línea siguiente. Por ejemplo, si la posición en Y es 16 y la siguiente línea está en la posición 20, su diferencia será 4. Ahora si el ancho es 10, los umbrales a y b serán 2 y 7 respectivamente. Como la diferencia es menor al umbral b ($4 < 7$), se asigna a la variable “sonido”, la línea de la posición encontrada menos 0.5 debido a que pertenece al espacio y no una línea.

Las líneas son representadas por números enteros por ejemplo, si hay 5 pentagramas y cada pentagrama tiene 6 líneas adicionales se tendrá un total de 55 líneas y los espacios serán representados por números decimales que irán desde 0.5 hasta 55.5 en aumento de uno en uno, es decir 0.5, 1.5, 2.5, 3.5....., hasta 55.5

Figura 42. Búsqueda de ubicación de la figura.



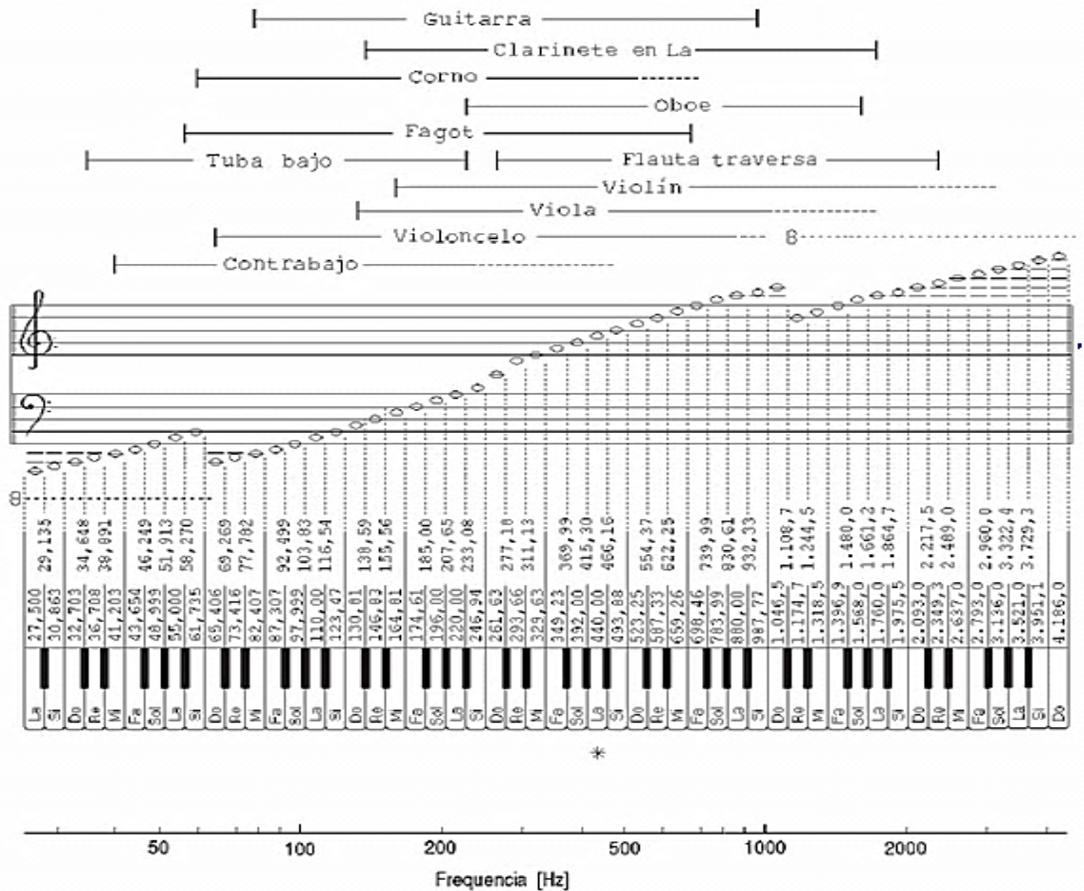
Fuente de los autores.

3.5.4 Asignación y adecuación del pentagrama y sonido a la nota. En esta etapa se asignarán las frecuencias a cada nota para posteriormente ser reproducidas, a continuación se explica de donde salen las frecuencias correspondientes a cada nota.

Rango de frecuencias para algunos instrumentos.

En la Figura 43 se muestra el rango de frecuencias para algunos instrumentos musicales, tomando como base o referencia la escala de un piano con sus correspondientes frecuencias.

Figura 43. Rango de frecuencias, tomando como referencia el piano.



Fuente: http://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io2/public_html/escalas.html

Para calcular las frecuencias, se toma como base la nota “La” ubicada con un asterisco en la imagen anterior y se utiliza la siguiente expresión.

Ecuación 2. Ecuación para hallar frecuencias de las notas.

$$f_i = 440 * 2^{i/12}$$

Fuente: http://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io2/public_html/escalas.html

Donde “f” es la frecuencia de la nota, i es el desplazamiento o distancia de “La” hasta la nota de interés, siendo “La” igual a cero como se muestra en Figura 44.

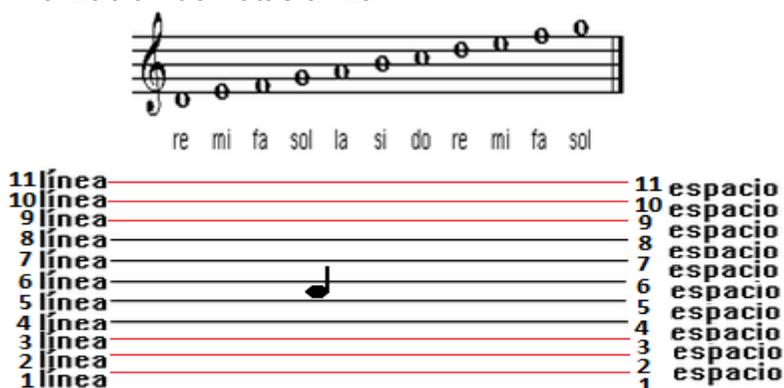
Figura 44. Asignación para frecuencias de las notas en base a “La”.



Fuente de los autores.

Para poder aplicar la formula se debe “normalizar” todas las notas encontradas a “La”, por lo cual se resta 5.5 a todas las notas.

Figura 45. Normalización de notas a “La”.

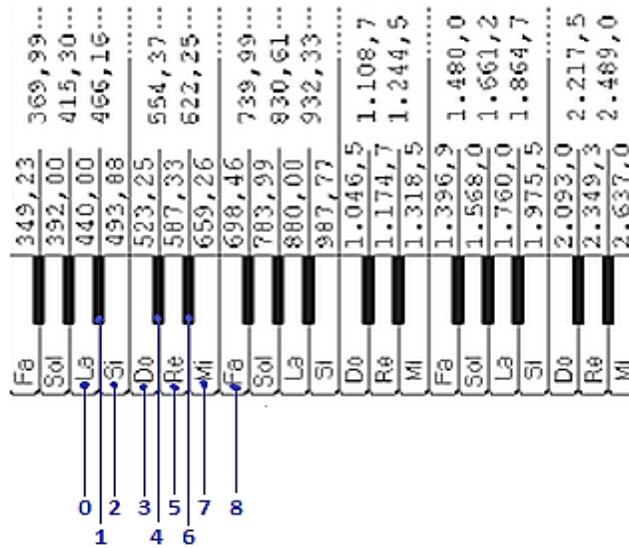


Fuente de los autores.

Como se puede observar La en clave de Sol se encuentra en el segundo espacio, que hace referencia a la variable “sonido” igual a 5.5, al restarle el valor 5.5 da 0, por lo cual se ha de restar a todas las notas este valor, de esta forma el resto tomara el valor correspondiente para ser introducido en la expresión 2.

Ahora se debe tener en cuenta las notas sostenidas, pues hasta ahora no se han tocado las figuras de alteración.

Figura 46. Asignación con notas sostenidas.



Fuente de los autores.

Ha de tenerse en cuenta que para el caso de “La Sostenido”, en la ecuación 2, la variable i estaría representada por 1 como se ve en la Figura 46, pero en el programa 1 sería en realidad la nota “Si”, por lo cual se debe buscar la forma de “saltar”. Para solucionar esto se hace uso de los siguientes condicionales.

```

sonido=sonido-5.5;%Normalizo a La=0 =LA(440)
sonido=sonido*2; %1*Nota sonidos deben ser enteros

    %Saltar los espacios para Sostenidos según teclado
    %#Escala Cromatica#
if sonido>0 && sonido<=2
    sonido=sonido+1;
elseif sonido>2 && sonido<=4
    sonido=1+sonido+(sonido-2);
elseif sonido==5
    sonido=sonido+3;
elseif sonido>5 && sonido<=8
    sonido=3+sonido+(sonido-5);
elseif sonido==9
    sonido=sonido+6;
elseif sonido>9 && sonido<=11
    sonido=6+sonido+(sonido-9);

```

Para el caso de la nota “Si” en el programa sería 1 pero para la ecuación 2 sería el valor 2, con el ajuste del código anterior cuando la variable sonido tenga un valor entre 1 y 2 se suma 1. Para el caso del ejemplo “Si” corresponde a 1 al sumar otra unidad queda 2, por lo cual al reemplazar en la ecuación 1 da una frecuencia de 493.8 Hz correspondiente a la nota “Si” en la distribución del piano.

La función que cumplen los condicionales del código, es adecuar los resultados encontrados en el programa a la fórmula, para hallar la frecuencia de las notas naturales. Más adelante se explicará la forma en que se relacionan las notas sostenidas o figuras de alteración. Lo mismo ocurre para las notas por debajo de “La”.

Para finalizar, el bloque Analizador devuelve la nota con el pentagrama que corresponde, la posición y la frecuencia al bloque Clasificador.

Para el caso de las Claves solo se requiere saber la posición y el número del pentagrama al que pertenece y para los silencios además de la posición y número del pentagrama, se necesita la duración, que viene dada por el bloque clasificador.

3.6 APLICADOR DE ATRIBUTOS

El bloque aplicador de atributos se encarga de aplicar los atributos de las figuras de duración y de alteración que son muy importantes para obtener una melodía armónica. También crea una variable dedicada al almacenamiento de las señales sinusoidales, que son las que convertirán en un formato de audio posteriormente en el bloque generador de audio. La Figura 47 muestra las funciones que ejecuta el bloque aplicador de atributos.

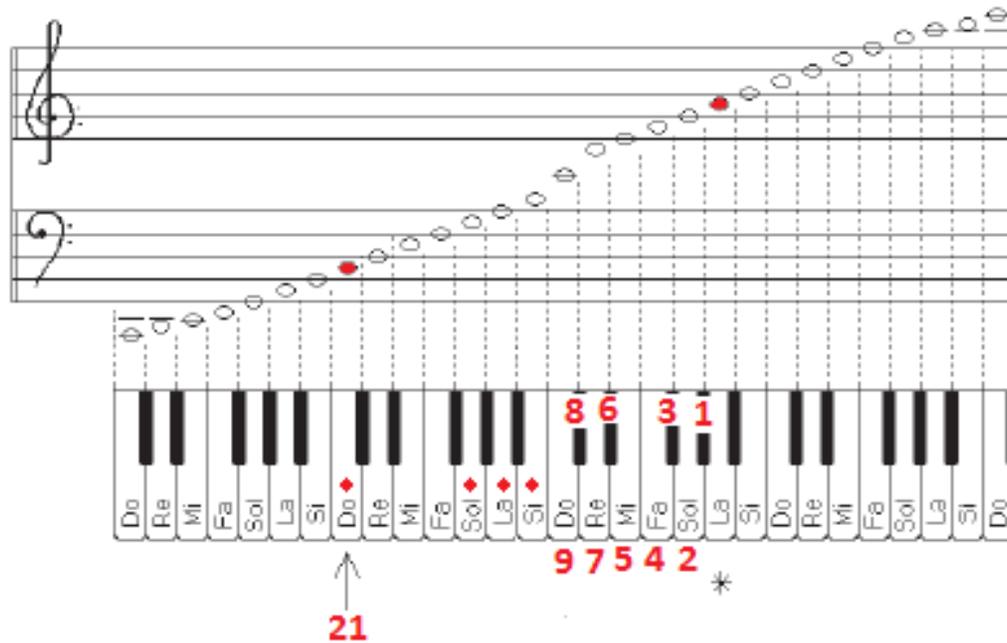
Figura 47. Bloque Aplicador de Atributos.



Fuente de los autores.

3.6.1 Aplicador de Clave. Lo primero es identificar en que clave está escrita la partitura, que es la primera pauta para saber cuál es la frecuencia natural de las notas; como se ha mencionado anteriormente las notas del pentagrama están normalizadas en la nota “La” dentro las frecuencias de la Clave de “Sol”, razón por la cual cuando una partitura tiene como clave a “Sol” no modifica la variable sonido, mientras la clave de “Fa” hace necesario restar veintiuno a la variable sonido.¹⁴

Figura 48. Valores de Sonido para Clave.



Fuente de los autores.

3.6.2 Aplicador de Alteraciones. Seguidamente se evalúan y aplican los sostenidos y bemoles que alteran la frecuencia natural de las notas. La particularidad de estas alteraciones es que siempre están ubicadas al lado izquierdo de una nota, por lo que al identificar una alteración se le aplica a la nota siguiente a esta. Otro caso de alteración de frecuencias esta dado en la armadura del pentagrama, a lo que se define como tonalidad, y aplica sus respectivas alteraciones de frecuencias a las correspondientes notas. Según sea la tonalidad, las notas que son alteradas son de la siguiente manera.

¹⁴ Acústica Musical [en línea]. Recuperado: 20 de Octubre de 2012. Disponible: http://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io_2/public_html/escalas.html

Cuadro 5. Tipos de tonalidad.

Armadura	Tonalidad mayor	Tonalidad menor
<i>sib, mib, lab, reb, solb, do_b, fab</i>	<i>do</i> bemol mayor	<i>la</i> bemol menor
<i>sib, mib, lab, reb, solb, do_b</i>	<i>sol</i> bemol mayor	<i>mi</i> bemol menor
<i>sib, mib, lab, reb, solb</i>	<i>re</i> bemol mayor	<i>si</i> bemol menor
<i>sib, mib, lab, reb</i>	<i>la</i> bemol mayor	<i>fa</i> menor
<i>sib, mib, lab</i>	<i>mi</i> bemol mayor	<i>do</i> menor
<i>sib, mib</i>	<i>si</i> bemol mayor	<i>sol</i> menor
<i>sib</i>	<i>fa</i> mayor	<i>re</i> menor
	<i>do</i> mayor	<i>la</i> menor
<i>fa_#</i>	<i>sol</i> mayor	<i>mi</i> menor
<i>fa_#, do_#</i>	<i>re</i> mayor	<i>si</i> menor
<i>fa_#, do_#, sol_#</i>	<i>la</i> mayor	<i>fa</i> sostenido menor
<i>fa_#, do_#, sol_#, re_#</i>	<i>mi</i> mayor	<i>do</i> sostenido menor
<i>fa_#, do_#, sol_#, re_#, la_#</i>	<i>si</i> mayor	<i>sol</i> sostenido menor
<i>fa_#, do_#, sol_#, re_#, la_#, mi_#</i>	<i>fa</i> sostenido mayor	<i>re</i> sostenido menor
<i>fa_#, do_#, sol_#, re_#, la_#, mi_#, si_#</i>	<i>do</i> sostenido mayor	<i>la</i> sostenido menor

Fuente: [http://es.wikipedia.org/wiki/Tonalidad_\(m%C3%BAsica\)](http://es.wikipedia.org/wiki/Tonalidad_(m%C3%BAsica))

Como se observa en el Cuadro 5, la tonalidad puede ser mayor y menor pero en este proyecto solo se tendrá en cuenta la mayor, también se nota que la tonalidad se halla con el número de sostenidos y bemoles que se encuentre en la armadura de la clave. Ahora, el orden de las notas alteradas por la tonalidad es: *Fa-Do-Sol-Re-La-Mi-Si con sostenidos* y *Si-Mi-La-Re-Sol-Do-Fa*, diferente del orden natural de las notas que es *Do-Re-Mi-Fa-Sol-La-Si* siendo necesario hacer ajustes para la tonalidad de una partitura como se muestra en la siguiente función del código.

```
function nsonido=evalalter(sonido,ton)

nsonido=sonido;
salto=0;
if ton>=0      %Para sostenidos
    for k=1:ton  %(El numero de sostenidos, afecta a las nota
        hasta el correspondiente numero. Sost=FA DO SOL RE LA MI SI )

                if (sonido==-16+salto||sonido==-4+salto||sonido==8+salto
                    ||sonido==20+salto) %NOTA acorde a la modificacion hecha por los
                    sostendios, a estas.

                nsonido=sonido+1;      % Suma a todas las notas un
sonido.
```

```

        break
    end
    if (mod(k,2)==0)           %si es par o impar el numero de
sostenidos en el encabezado
        salto=salto-5; %cuando par entonces reste 5 a salto
    else
        salto=salto+7; %cuando impar sume 7 a salto
    end
end
end

else

    ton=abs(ton); %Cuado la tonalidad es bemol

    for k=1:ton % (El numero de bemoles, afecta a las nota hasta
el correspondiente numero. Sost=SI MI LA RE SOL DO FA)

        if (sonido==-10+salto||sonido==2+salto ||sonido==14+salto)
            %
            nsonido=sonido-1;

            break
        end

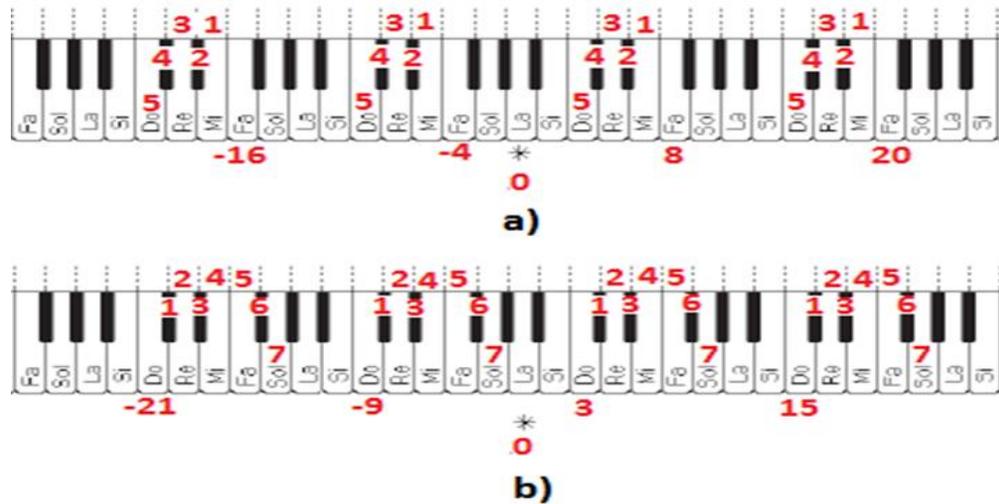
        if (mod(k,2)==0)
            salto=salto+5;
        else
            salto=salto-7;
        end
    end
end

end

```

Este código busca las notas que son alteradas por la tonalidad de la partitura mediante saltos y agregando o quitando tonos a la variable *sonido*, debido a que una nota es equidistante a sí misma se puede modificar los tonos que genera cada nota por ejemplo, si una partitura tiene la tonalidad de *Sí mayor*, “*Fa*” es la primera nota alterada por la tonalidad de la partitura (mirar Cuadro 5) y tiene las ubicaciones de -16, -4, 8 y 20 en la escala cromática del piano como se observa en la parte a) de la Figura 49, estas posiciones se les altera su tono en uno a la variable *sonido*. La segunda nota alterada es “*Do*” que se encuentra a cinco posiciones de “*Fa*” como se observa en la parte b) de la Figura 49, por ende se resta 5 posiciones a las anteriores teniendo como resultado -21, -9, 3 y 15 y se les aplica la tonalidad correspondiente, para la siguiente nota se suman 7 posiciones y este proceso se repite cuantas veces sea necesario para cumplir con las alteraciones de los tonos de las notas involucradas en la tonalidad de la partitura.

Figura 49. Saltos de Tonalidad a las notas, a) impares, b) pares.

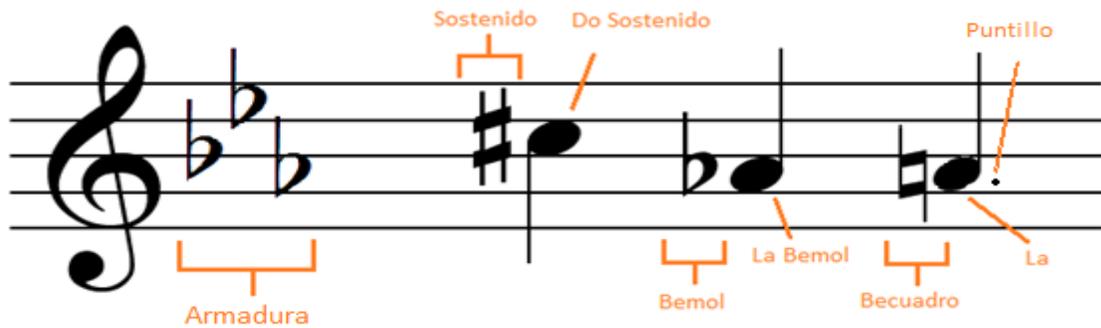


Fuente de los autores.

Las alteraciones para las tonalidades con bemoles en la armadura, se realizan de la misma forma solo que ahora se suma cinco para las notas pares y se restan siete para las impares, y a estas posiciones se le restan uno en la variable *sonido*.

También al encontrar un becuadro retorna el tono natural de una nota para el restante de la partitura.

Figura 50. Armadura y figuras de Alteración.



Fuente de los autores.

Además el bloque Aplicador de Atributos incluye las figuras de duracion como el puntillo que contrario a las figuras de alteracion, estas se encuentran ubicados al lado derecho de la nota como muestra en la Figura 50 y altera la duracion de la nota anterior a su ubicacion en medio tiempo.

3.7 GENERADOR DE AUDIO

El bloque Generador de Audio se encarga de crear el archivo de audio que se encuentra inscrito en la partitura, la Figura 51 muestra las funciones empleadas.

Figura 51. Bloque Generador de Audio.



Fuente de los autores.

3.7.1 Almacenamiento de ondas sinusoidales. Esta etapa es la encargada de crear la variable llamada “song” que almacena todos los tonos con sus respectivas notas. Para ello se usa primero la duración, la velocidad de la partitura y una frecuencia de muestreo para obtener el tiempo de duración de cada una de las notas. Para este proyecto la frecuencia de muestreo seleccionada fue ocho mil muestras por segundo debido a que permite reproducir señales de hasta 3,5 kHz la cual es suficiente para reproducir el tono de mayor frecuencia con 1,318 kHz en la escala del piano, por otra parte reduce el costo computacional en comparación con otras frecuencias de muestreo. También se agrega un tiempo muerto entre notas para que los tonos se escuchen de una forma natural y armónica para el oído humano.

Por otra parte se aplica la Ecuación 2 a la variable *sonido* para obtener la frecuencia de las notas respectivas y se usan los datos extraídos para obtener una sucesión de ondas senos mediante la Ecuación 3

Ecuación 3. Ecuación sinusoidal

$$seno(t) = \sin(2 * \pi * frec * t/8000)$$

Fuente: http://es.wikipedia.org/wiki/frecuencia_de_muestreo

donde "t" es el tiempo de duración de nota, "frec" es la frecuencia de la nota y 8000 es la frecuencia de muestreo. Esta sucesión de ondas senos se guardan en la variable "song" mencionada anteriormente.

3.7.2 Conversor y exportador de audio. En esta función se reproduce el audio de la partitura, mediante la expresión

```
cancion=audioplayer(song,8000);
play(cancion);
```

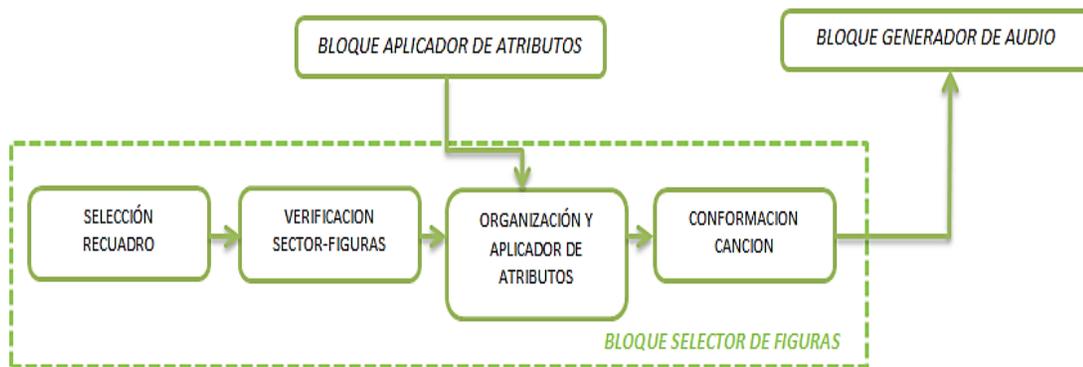
donde "audioplayer" es una función de Matlab que devuelve el manejo de un objeto para reproducción de audio. En este caso la señal a reproducir es "song", que contiene todas las sinusoides de las notas organizadas con una tasa de muestreo de 8 KHz, luego se reproduce con el comando "play".

Por último para exportar o guardar el archivo de audio se usa una función especial de matlab llamada "wavwrite" la cual tiene como parámetros de entrada la variable de almacenamiento de las ondas sinusoidales y frecuencia de muestreo para generar el audio, esto se realiza en las dos líneas de código que se observan a continuación.

```
[FileName,PathName] = uiputfile('*.wav','Indique la ruta de Guardado');
wavwrite(song,8000,[PathName FileName]);
```

3.8 BLOQUE SELECTOR DE FIGURAS

Figura 52. Bloque selector de figuras.



Fuente de los autores.

El bloque Selector de Figuras permite reproducir sectores específicos de la partitura o alguna nota en particular, además permite cambiar la clave del sector

que se quiera escuchar, al igual que la tonalidad. La figura 52 muestra las funciones empleadas en este bloque.

3.8.1 Selección recuadro. En la Figura 53 se puede observar el recuadro de color azul, el cual al seleccionar una parte de la imagen, automáticamente se reproducirá el sector.

Para poder dibujar el recuadro se usa la función “imrect” del Toolbox de procesamiento de imagen de Matlab, el cual permite crear un recuadro dibujable (con el puntero del mouse).

Figura 53. Recuadro selector de figuras.



Fuente de los autores.

Las siguientes dos líneas de código permiten lo anterior

```
hrect=imrect;  
sector = getPosition(hrect);
```

donde “hrect” es la variable o el objeto que contiene las propiedades del rectángulo (“imrect”). Una de las propiedades de mayor interés en este proyecto es la posición que se consigue con la función “getPosition” que arroja un vector con cuatro valores [Xmin Ymin Ancho Alto].

3.8.2 Verificación sector-figuras. Una vez se han conocido los puntos o la ubicación del sector, ha de analizarse donde se encuentra el sector, el pentagrama que ocupa y que figuras involucra.

Se inicia buscando si la posición en el eje Y es menor a la última línea del último pentagrama, en caso de seleccionar fuera del último pentagrama como se observa en la Figura 54 se entregara un vector vacío que no reproducir nada. En caso contrario indica que se ha seleccionado algo para ser reproducido como se muestra en la Figura 55.

Figura 54. Ejemplo recuadro por fuera del último pentagrama.

Chiquitita

ABBA diegosax.es
 directoriopax.com tubeparatitura.com
ABBA

Saxo Tenor
Soprano Sax

The image shows a musical score for 'Chiquitita' by ABBA. It features two staves: Saxo Tenor and Soprano Sax. The key signature is three sharps (F#, C#, G#) and the time signature is 4/4. The score consists of two staves. The second staff has a blue rectangular box drawn around a portion of the music, specifically the notes on the second and third lines of the staff. A blue line with the label 'x,y' points to the top-left corner of this box. The box is positioned such that its bottom edge is below the bottom line of the staff.

Fuente de los autores.

Figura 55. Ejemplo recuadro dentro de los pentagramas.

Chiquitita

ABBA diegosax.es
 directoriopax.com tubeparatitura.com
ABBA

Saxo Tenor
Soprano Sax

The image shows the same musical score for 'Chiquitita' as in Figure 54. In this version, a blue rectangular box highlights a section of the second staff that is positioned entirely within the staff lines. A blue line with the label 'x,y' points to the top-left corner of the box, and another blue line with the label 'x,y2' points to the bottom-left corner. The box encloses a sequence of notes on the second and third lines of the staff. The score continues to a third staff, which ends with the word 'Fin'.

Fuente de los autores.

Ahora se halla la línea por encima del eje Y, y la línea que se encuentre por debajo de Y_2 , para conocer en que pentagrama se encuentra el sector seleccionado. Ahora se seleccionan las figuras musicales cuyos pentagramas correspondan al encontrado y además cuya posición este entre X y X_2 como se muestra en la Figura 56.

Figura 56. Búsqueda de figuras dentro del recuadro de selección.



4. RESULTADOS.

En este capítulo se analizan los resultados obtenidos de forma que se pueda determinar si los objetivos propuestos en el proyecto se cumplen, además de evaluar la eficiencia y eficacia de las diferentes etapas del sistema en general. En este caso, el objetivo principal será obtener la melodía (audio) correspondiente, a partir de una imagen digital de una partitura musical. A continuación se analizan los resultados.

Bloque de Preprocesamiento.

El bloque de preprocesamiento de imagen además de encargarse de filtrar, eliminar ruido y la conversión a formato binario, se encarga de un proceso a un más importante, el detectar la inclinación de la imagen (partitura) y corregirla. En las Figuras 57, 58 y 59 se muestran algunos de los resultados obtenidos con partituras que tienen cierto grado de inclinación.

Figura 57. Partitura Noche de Paz con inclinación de -0,8 grados.



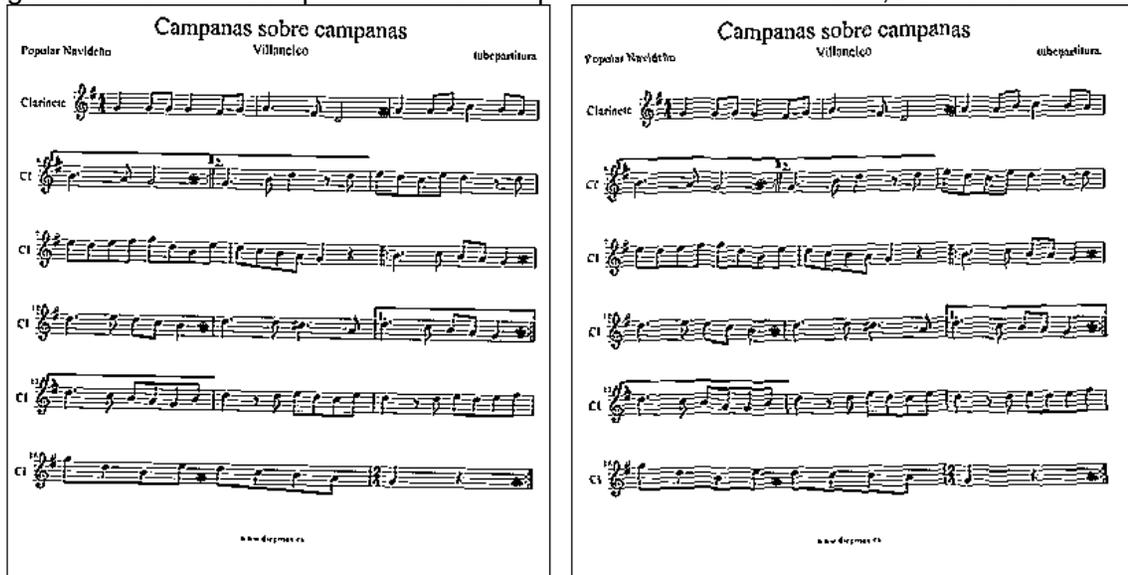
Fuente de los autores.

Figura 58. Extraído de Himno de la alegría de Beethoven con inclinación de un grado.



Fuente de los autores.

Figura 59. Partitura Campanas Sobre Campanas con inclinación de 1,2 Grados.



Fuente de los autores.

Vale aclarar que no se ha tenido en cuenta las distorsiones de la partitura, dado a que al rotar la imagen es posible que los pentagramas no queden perfectamente rectos, el cual abre la posibilidad de que el bloque analizador no reconozca adecuadamente el tono de las notas en determinadas posiciones. Si bien el código de corrección de inclinación no lo hace a la perfección, si logra una inclinación casi nula por debajo de las dos décimas de grado para las imágenes con un grado de inclinación considerable. En general esta etapa cumple con las expectativas de forma satisfactoria aunque fuese una de las etapas más complejas y difíciles de resolver en el proyecto.

El tiempo de ejecución de rotación es relativamente corto, del orden de los segundos, aunque este factor es directamente proporcional al tamaño de la imagen debido a que la función de Matlab “*imrotate*” requiere de más costo computacional a mayor tamaño de la imagen. El Cuadro 6 muestra el tiempo de procesamiento del bloque de preprocesamiento excluyendo la etapa de corrección de inclinación.

Cuadro 6. Tiempo de Preprocesamiento.

FIGURA	TAMAÑO (pixeles)	TIEMPO DE EJECUCION (seg)
58	1672 x 914	5.90
59	403 x 402	1.84
60	864 x 107	1.42

Fuente de los autores.

Si bien estos son los tiempos de la primera fase del bloque preprocesamiento como el filtrado, la eliminación de ruido y la conversión a binario, se continúa con la corrección de inclinación registrando los siguientes tiempos de procesamiento.

Cuadro 7. Tiempo de rotación.

FIGURA	TAMAÑO (píxeles)	TIEMPO DE EJECUCION (seg)
58	1672 x 914	72.35
59	403 x 402	18.12
60	864 x 107	13.86

Fuente de los autores.

Se puede observar de los Cuadros 6 y 7 que el mayor tiempo de procesamiento del bloque de preprocesamiento se encuentra en la fase de detección y corrección de líneas horizontales, además dicho tiempo depende del tamaño de la imagen.

Bloque Clasificador y Bloque Analizador.

Para evaluar estos bloques es necesario recordar que finalidad tienen estos. El objetivo del bloque Clasificador es el de reconocer claramente que figura musical es cada elemento en la imagen y del bloque Analizador es encontrar a que nota pertenece la figura musical. Como estos bloques parten de la extracción de características de los pentagramas y la imagen limpia de las figuras sin pentagramas como se ve en la Figura 60.

Figura 60. Ejemplo 4. (Extraído de Himno de la alegría de Beethoven).



Fuente de los autores.

El bloque Clasificador tiene una eficiencia muy alta en el momento de distribuir las figuras musicales según sea el grupo de pertenencia, debido a la limpieza meticulosa de los pentagramas y de la información no relevante o no aportante a

la melodía escrita en la partitura, además este bloque es eficiente por la amplia base de datos que existe y la posibilidad de seguirla ampliando por el usuario en cualquier momento. Por eso es considerado un bloque muy eficiente y eficaz.

Finalmente, el bloque Analizador que tiene una gran eficiencia y eficacia de encontrar la nota a la que pertenece la figura musical. Para ello realiza un filtrado minucioso, para encontrar el espacio en que se ubica una figura musical y por ello su margen de error es prácticamente nulo. En la Figura 61 se puede observar algunos recuadros que sirven para reconocer visualmente algunas figuras, por ejemplo los recuadros rojos encierran las claves, los azules las alteraciones, los verdes los silencios, etc. También se pueden observar los asteriscos que no se mas que el indicador del centroide de la cabeza de cada nota.

Figura 61. Ejemplo 5. (Extraído de Chiquita de Abba).



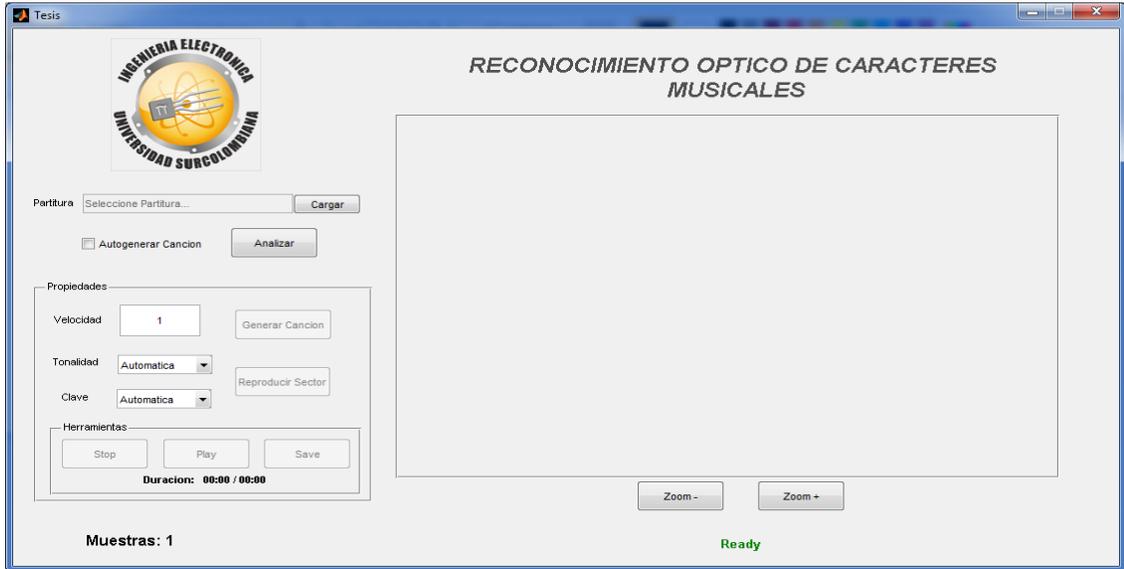
Fuente de los autores.

Los tiempos de ejecución son proporcionales al número de figuras musicales inscritas en la partitura. En general el costo computacional es relativamente bajo debido a la eficiencia y eficacia de los algoritmos desarrollados en este proceso.

4.1 INTERFAZ GRÁFICA DEL SOFTWARE.

Esta es una de las etapas más relevantes del proyecto, debido a que se desea que cualquier persona pueda hacer uso del software. Es por ello que se desarrolló una interfaz muy sencilla y amigable para el usuario como se observa en la Figura 62.

Figura 62. Interfaz Gráfica del Software.



Fuente de los autores.

Dado a que el software es semiautomático el usuario con solo cargar la imagen, seleccionar la opción de “Autogenerar Canción” y presionar el botón “Analizar”, ya puede generar el audio que está inscrito en la partitura musical. Además cuenta con herramientas útiles y sencillas de manejar que se explican en el manual del software (ver Anexos), por estas razones se considera que es una interfaz eficiente y eficaz debido a la sencillez de su manejo.

Figura 63. Programa ejecutado.



Fuente de los autores.

4.2 SOFTWARE

Este software fue diseñado para la reproducción semiautomática de partituras musicales digitales, con una interfaz sencilla y fácil de utilizar por el usuario.

Uno de los objetivos de la aplicación es que el usuario pueda usarlo como medio de aprendizaje musical, dado a que posee herramientas muy útiles y sencillas de manejar, entre las cuales se puede cambiar la tonalidad, la clave y la velocidad de la partitura. Estas son herramientas que permiten armonizar la melodía.

Además cuenta con otra funcionalidad muy útil que le permite al usuario seleccionar una nota o sector que desee reproducir de la partitura, teniendo varios beneficios. Por ejemplo, si el usuario desea entender cómo debe sonar y que tanto tiempo debe durar el sonido de determinada figura musical o si se desea escuchar determinado compás o pentagramas o sector de la partitura. Esta herramienta facilita el estudio de una persona que por ejemplo se le dificulte tocar en su instrumento o cierto sector de una partitura, entonces podría seleccionar solo ese fragmento y escucharlo.

Finalmente este software busca ser una herramienta adicional para las personas, en el uso académico y de aprendizaje en lo relacionado a música, más en particular en cómo puede sonar o ser interpretadas diferentes partituras musicales.

5. CONCLUSIONES

- La aplicación cuenta con una serie de bloques que realizan cada una de las tareas definidas por los objetivos iniciales. Se logró implementar un método eficaz para corregir el problema de la extracción adecuada de los pentagramas y las notas.
- Una de las etapas más importantes del proyecto es el preprocesamiento, ya que entrega la imagen en condiciones ideales para luego ser tratada. La forma de afrontar esta etapa depende de cada autor y de los alcances o finalidad del trabajo.
- Cuando se analizan o se quiere reproducir partituras en clave de Fa es difícil escuchar frecuencias tan bajas con los parlantes que traen incorporados los equipos de cómputo portátiles, por lo cual se hace necesario contar con un sistema de sonido de buena calidad.
- Como se ha mencionado, el mundo de la música es “infinito” por lo cual es importante que al no reconocer símbolos, estos sean almacenados para ser agregados a la base de datos posteriormente por el usuario. Esto por supuesto implica dos cosas, la primera de ellas, que a futuro se reconozcan más símbolos y se mejore el sistema de reconocimiento e identificación. La segunda implica un coste computacional, debido al incremento en el número de muestras con las que se debe comparar para encontrar el grupo perteneciente de la nota analizada.
- Hasta el momento no existe un sistema completamente automático de OMR. Debido al grado de complejidad de la música y sus distintas formas de representación e interpretación, como se pudo observar por ejemplo con el tiempo de reproducción. No todas las partituras tiene el mismo ritmo o velocidad, por lo cual es difícil conocer cuando se deben manejar unos tiempos y en qué casos otros, además de las escalas utilizadas por diversos autores.

Alguno de los objetivos alcanzados en este trabajo de grado ha llevado una gran dedicación. Cabe destacar el trabajo realizado para dar solución a.

1. Análisis de figuras y asignación de notas.
2. Asignación de tonos o figuras de alteración.

Para el primero se debió estudiar y analizar detenidamente todas las figuras musicales y de esta forma agruparlas para minimizar el costo computacional. Lo anterior está estrechamente relacionado con el bloque “Analizador”, debido que no todas las figuras musicales se deben examinar de la misma forma y no todas dan

la misma información. Para esto se desarrollaron algoritmos que arrojan un resultado óptimo para considerar cumplido el objetivo.

Para el segundo, se debió implementar un algoritmo basado en diversas reglas acorde a la teoría musical, para asignar las diferentes figuras de alteración a las notas y que correspondieran con los resultados esperados.

6. RECOMENDACIONES

Aunque se han conseguido los objetivos propuestos y se ha implementado un sistema OMR funcional, aún se pueden realizar algunas mejoras que acerquen al programa en uno ideal.

- Mejorar el algoritmo de corrección del ángulo de inclinación y rotación de la imagen, en cuanto a tiempo de ejecución y capacidad de trabajar con imágenes que tengan un alto grado de rotación.
- Reducir problemas de segmentación que existe cuando hay figuras continuas y una puede afectar a la otra. Una segmentación excesiva traerá como consecuencia un alto coste computacional y una muy pobre acarrea inconvenientes a la hora de clasificar e identificar.
- Modificar la aplicación para que sea capaz de analizar varias hojas de la partitura y luego ser reproducidas como un solo archivo de audio, debido a que las obras musicales por lo general se componen por más de una hoja.
- Mejorar el diseño del sistema de forma que admita partituras más complejas como las del piano que producirán sonidos más armónicos.
- Diseñar y entrenar una red neuronal para el reconocimiento de partituras manuscritas, debido a que no todas las personas tienen la misma forma de escribir y es un método apropiado para este tipo de partituras musicales, ya que una base de datos de símbolos manuscritos sería extensa a causa de las infinitas formas y diferencia en la escritura de los símbolos.
- Mejora en la interfaz del usuario permitiéndole corregir malas clasificaciones.
- Incluir la opción de reproducir los sonidos de diferentes instrumentos musicales para generar sonidos más acorde a la partitura a reproducir.
- Agregar una herramienta para usuario en la cual pueda modificar o alterar determinada nota en la partitura.
- Creación de un módulo que sea capaz de interactuar con dispositivos de entrada como escáneres o cámaras.
- Implementar el programa en lenguaje C debido a su eficiencia que permite utilizar sus características de bajo nivel para implementaciones óptimas.

BIBLIOGRAFIA

- UNIVERSIDAD DE VALLADOLID. Acústica Musical [en línea]. Actualizada 2005. [Fecha de Consulta: 20 de Octubre de 2012]. Disponible en: <http://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io2/public_html/escalas.html>
- ARMINGOL, JOSE MARIA. Preprocesamiento de Imágenes. En: Sistemas de Percepción. Madrid. Agosto 16, 2010.
- BAINBRIDGE, DAVID. The Challenge of Optical Music Recognition. En: Computers and the Humanities. 2001, p. 95-121.
- CARRETERO DE LA HOCHA, DAVID. Sistema De Reconocimiento De Partituras Musicales. Proyecto fin de carrera. Madrid, España: Universidad Carlos III De Madrid. Ingeniería Técnica De Telecomunicación. Especialidad: Sonido E Imagen. 2009. 203 p.
- GONZALES AGUILERA, DIEGO. Procesamiento de Imágenes. Universidad de Salamanca. Master De Geotecnologías Cartográficas En Ingeniería Y Arquitectura. Castilla y Leon. 2008, p 3-23.
- JOHANSEN, LINN SAXRUD. Optical Music Recognition. Tesis para el grado de Maestría en Ciencias de Modelización y Análisis de Datos. Universidad de Oslo. Facultad de Matemáticas y Ciencias Naturales. Departamento de Matemáticas. 2009, 101 p.
- López Espejo, Iván. OMR y Síntesis de Partitura. En: Propuesta de Proyecto [en línea]. Actualizada en Octubre de 2012. [Fecha de Consulta: 14 de Noviembre de 2012]. Disponible en: <http://es.scribd.com/doc/110692597/6/OMR-y-sintesis-de-partituras>
- PALOMINO, NORA. Técnicas de Segmentación en Procesamiento Digital de Imágenes. En: Revista de Ingeniería de Sistemas e Informática VOL 6, No 2, Julio-Diciembre, 2009, p. 10-16.
- PEREZ, JOSE IGNACIO. Segmentación y búsqueda de patrones en partituras musicales. Proyecto fin de carrera. Madrid, España: Universidad Rey Juan Carlos. Ingeniería Técnica en Informática de Sistemas. 2005, p 64.
- P. BELLINI, I. BRUNO Y P. NESI. Assessing Optical Music Recognition Tools. Computer Music Journal Spring 2007, p 68-93

RAPHAEL, CHRISTOPHER. New Approaches To Optical Music Recognition. En. Sociedad Internacional para la Conferencia de Recuperación de Información Musical [en línea]. 2011, p 305-310. [Fecha de Consulta: 14 de Noviembre de 2012]. Disponible en: < <http://ismir2011.ismir.net/papers/OS3-3.pdf> >

RODRIGUEZ ALVIRA, J. ¿Qué es una escala? [en línea]. Actualizada 2011. [Fecha de Consulta: 20 de Octubre de 2012]. Disponible en: < <http://www.teoria.com/aprendizaje/escalas/> >

REVIEW OF OMR RESEARCH. Documentación [en línea]. Disponible en: < http://evirtual.lasalle.edu.co/info_basica/nuevos/guia/GuiaClaseNo.3.pdf >

Roger B Dannenberg. Review Of OMR Research, [en línea]. [Fecha de Consulta: 26 de agosto de 2012]. Disponible en: <http://www.music.mcgill.ca/~ich/research/diss/2.%20Reviews.pdf>. ISMIR, Octubre 2006, p 2-25.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACION. Norma Técnica Colombiana 1486 Documentación: Presentación de tesis, trabajos de grado y otros trabajos de investigación. Sexta actualización. Bogotá: ICONTEC, 2008.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACION. Norma Técnica Colombiana 5613. Referencias bibliográficas, contenido, forma y estructura. Sexta actualización. Bogotá: ICONTEC, 2008.

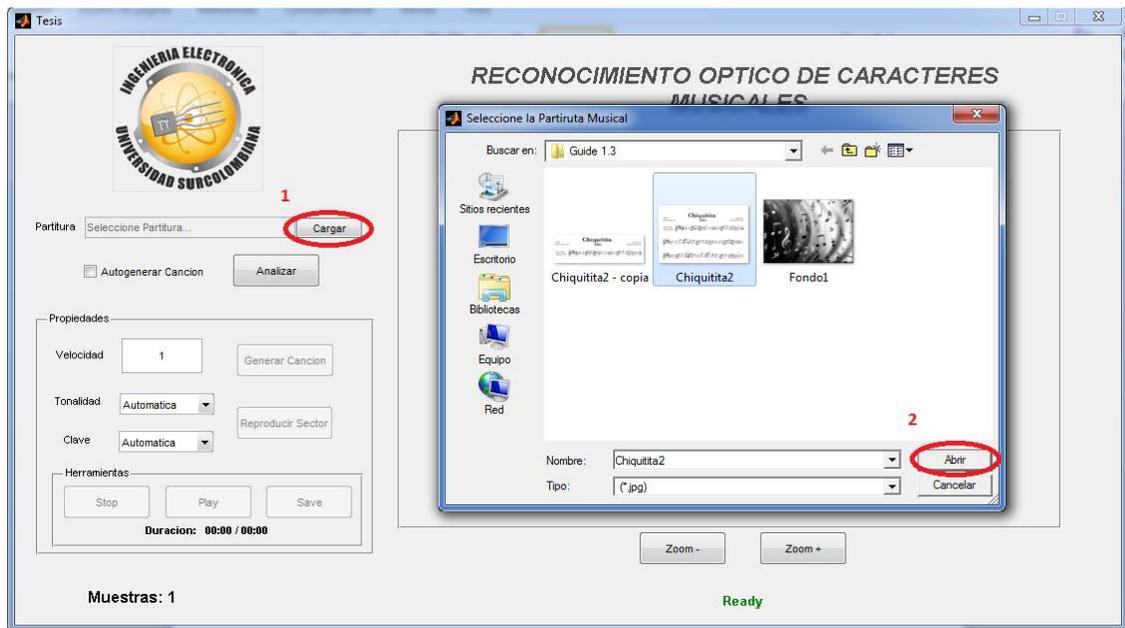
VARIEDAD DE CURSOS GRATIS. Teoría Musical [en línea]. Actualizada 2007. [Fecha de Consulta: 26 de agosto de 2012]. Disponible en: < <http://www.aprende-gratis.com/teoria-musical/> >

ANEXOS

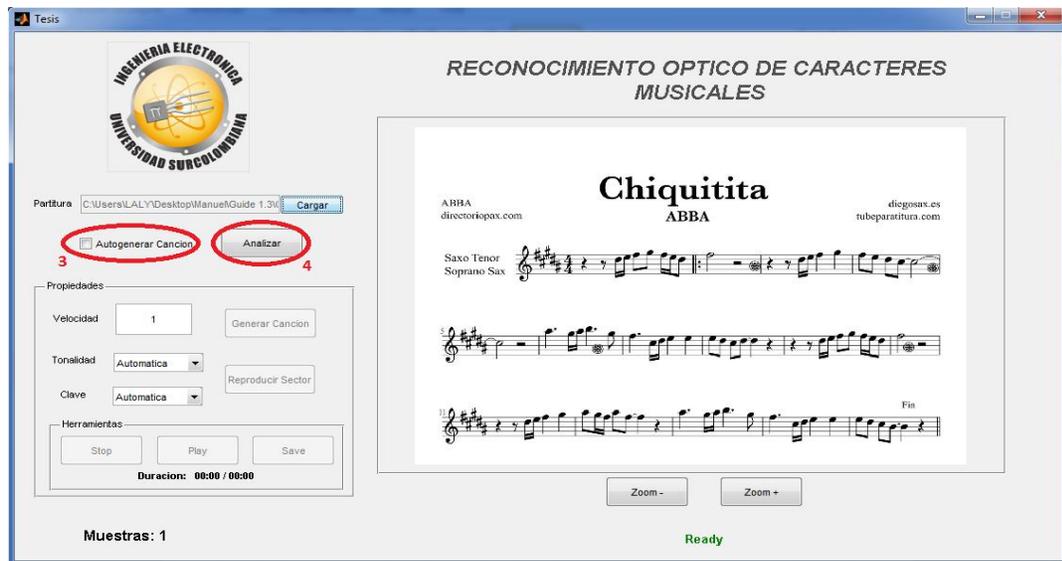
A. Manual de uso para el sistema “OMR”



El sistema OMR es un programa muy sencillo de usar, dado a que el sistema es prácticamente automático. Lo primero que se debe hacer es cargar una imagen (partitura) presionando el botón “Cargar” el cual despliega la siguiente ventana.



Seleccione la ruta de la imagen que desea reproducir y presiona el botón de “Abrir” para cargar la partitura en el programa. Seguidamente debe presionar el botón “Analizar” teniendo la posibilidad de que inmediatamente al terminar el sistema sus procesos, genere automáticamente el audio de la canción inscrita en la partitura con solo seleccionar (marcar) la opción de “Autogenerar” antes de presionar el botón “Analizar”.



El paso número 3 es opcional, después de presionar el botón “Analizar” el programa comienza a procesar la imagen para generar el audio respectivo.



El proceso de análisis de la imagen tardara varios segundos, la barra azul en la parte inferior indica cuanto ha transcurrido del proceso total y debajo de esta se

indica específicamente en que proceso del código del programa se está ejecutando.



Se recomienda esperar a que el programa termine completamente el proceso para seguir con los ajustes o parámetros de entrada para la partitura.



Cuando el programa haya finalizado el análisis, se reconoce porque la imagen ha sido marcada y señalada las cabezas de las figuras musicales, y el indicador de procesos dirá “Ready”.



Si, se hubiese seleccionado la opción de Autogenerar Canción el audio de la partitura sonaría de inmediato, sino podemos modificar el valor numérico de la opción Velocidad que puede aumentar o disminuir la velocidad de reproducción del audio, por ejemplo 0.6. Al presionar el botón “Generar Canción” se reproduce el audio.



Otra funcionalidad del programa muy útil es la opción de “Reproducir Sector” que una vez presionado se puede dirigir hacia la partitura y con un clic sostenido seleccionamos la región que se desea reproducir.



Existen otras opciones o parámetros que se puede modificar tal como la tonalidad y la clave de la partitura, los que modifican el tono de la canción.



Por ultimo existe un comando de Herramientas que permite correr (Play), pausar y detener (Stop) el audio. Y los botones de zoom para acercar o alejar la partitura.

Tesis



RECONOCIMIENTO OPTICO DE CARACTERES MUSICALES

Chiquitita

ABBA
irectoriopax.com

diegosax.e:
tubeparatitura.com

Partitura: D:\proyecto de grado\Guide 1.3\Chiquitita2. Cargar

Autogenerar Cancion Analizar

Propiedades

Velocidad: 0.6 Generar Cancion

Tonalidad: SI Mayor Reproducir Sector

Clave: SOL

Herramientas: Stop Play Save Duracion: 00:00 / 00:34

Saxo Tenor
Soprano Sax



Zoom - Zoom +

Ready

Muestras: 381

RECONOCIMIENTO DE PARTITURAS MUSICALES POR MEDIO DE VISIÓN ARTIFICIAL

Recognition Of Musical Scores through artificial vision

Jose Jesus Salgado¹, Manuel Enrique Aldana² y Juan Sebastián Perdomo³

Resumen

Actualmente existen varias aplicaciones que trabajan con el reconocimiento de caracteres conocidos como OCR (Reconocimiento Óptico de Caracteres), también con reconocimiento de voz y palabras, pero el reconocimiento musical no es muy común, y de hecho hay pocos programas enfatizados en esta área. Es aquí donde comienza a surgir una nueva rama llamada OMR, conocida también por sus siglas en español como Reconocimiento Óptico de Música, que da a un ordenador la capacidad de reconocer y comprender los contenidos de una partitura, siendo el campo en donde se dedica este trabajo[9].

Se propone el desarrollo de un sistema de reconocimiento de notas musicales por medio de visión artificial, usando el software Matlabv. 7.11, que permita digitalizar una partitura musical y reconocer los caracteres musicales en ella, además de poder reproducir y generar el archivo de audio con la melodía o información obtenida de la partitura antes mencionada.

Palabras claves: OMR; Partitura; MATLAB.

Abstract

Currently there are several applications that work with character recognition known as OCR (Optical Character Recognition), also with voice recognition and words, but musical recognition is not very common, and in fact there are few programs emphasized in this area. It is here that begins to emerge a new branch called OMR, also known by its Spanish acronym as Optical Recognition of Music, which gives a computer the ability to recognize and understand the contents of a score.

It was chose to develop a recognition system of musical notes through vision, using Matlab software, which allows recognize musical recognize characters in it, besides being able to reproduce and create the audio file with the melody or information obtained from the score above.

Keywords: OMR; Partitura; MATLAB.

1. Introducción

Todas las materias o todas las ciencias tienen su idioma o formas y expresiones, por ejemplo el lenguaje de las matemáticas es formal y abstracto, acompañada de números y símbolos, al igual que la física, la química, la propia lengua como el inglés o el latín. Se dice que la música es un lenguaje universal, puesto que se lee de igual forma en

1 Ingeniero Electrónico. Universidad Surcolombiana. Neiva Av. Pastrana Borrero – Carrera 1. Jose.salgado@usco.edu.co

2 Ingeniero Electrónico. Universidad Surcolombiana. Neiva Av. Pastrana Borrero – Carrera 1. Manuel.sanchez@hotmail.com

3 Ingeniero Electrónico. Universidad Surcolombiana. Neiva Av. Pastrana Borrero – Carrera 1. Sebastian.perdomo@hotmail.com

cualquier parte del mundo, pero deben existir algunas reglas, organización o sintaxis, es aquí donde aparece la importancia de la correcta interpretación de la partitura.

La partitura es un documento en donde se indica cómo debe interpretarse una composición musical compuesto por uno o más pentagramas. El pentagrama es donde se centra toda la grafía musical, en éste se colocan los símbolos y notas, está compuesto por cinco líneas horizontales paralelas y equidistantes.



Fig. 1 Partitura musical

La finalidad de este proyecto es el desarrollo de un sistema que sea capaz de entender e interpretar una partitura previamente digitalizada que será de gran utilidad y apoyo como medio o alternativa de aprendizaje para aquellas personas inmersas en el infinito mundo de la música.

Desde el ámbito de la ingeniería electrónica el estudio, profundización y combinación de las diferentes técnicas de visión artificial, aportará en el desarrollo y avance de sistemas OMR más “sólidos”, que en un futuro puedan llegar a reemplazar la intervención del hombre y reformar las aplicaciones informáticas de la música.

2. Metodología

El sistema o aplicativo software trabaja con imágenes en formato jpg previamente digitalizadas y sin tener en cuenta el tamaño. Una vez leída la imagen debe pasar por una serie de bloques que se encargan de limpiar, extraer las características y e información de la partitura para ser reproducida y exportada en formato de audio WAV.

El sistema desarrollado es capaz de analizar los elementos de partituras no manuscritas. Debido a que es un proyecto de fin de carrera no cuenta con un grado de desarrollo equivalente al de un proyecto de tipo comercial. Esto implica ciertas limitaciones. A continuación se describirá lo que es capaz de reconocer el programa.

- Partituras de uno o más pentagramas.
- Símbolos no manuscritos.
- Claves de SOL y FA.
- Notas musicales como: Redonda, blanca, Negra, Corchea (unidas hasta cuatro), Semicorchea (Unidas hasta cuatro), Unión de corchea y semicorchea, identificación del separador, Puntillo, Silencios y Alteraciones.
- Capacidad para cambiar el tiempo de las figuras musicales de la partitura.
- Guardar símbolos u objetos no identificados para ser agregados luego a la base de datos.
- Notas cuya ubicación sea de hasta tres líneas adicionales.

Por otra parte, las limitaciones son.

- No acepta partituras de piano.

- No se permite notación musical distinta a la notación moderna (A no ser que se agregue posteriormente a la base de datos para ser identificada como tal).
- Se rechaza todo aquello que no sea pentagramas, como títulos o nombre de las partituras.
- Partituras rotadas máximo entre 1 y 1.5 grados

Se emplearon seis bloques, uno de PRE-PROCESAMIENTO que se encarga de limpiar y filtrar la imagen, seguidamente el bloque CLASIFICADOR agrupa las figuras musicales según su clase en conjunto con el bloque ANALIZADOR que se ocupa de identificar cada elemento con su respectiva ubicación y nota correspondiente, después el bloque APLICADOR DE ATRIBUTOS se encarga de las alteraciones de las notas, continuamente el bloque GENERADOR DE AUDIO permite reproducir y guardar el audio generado en formato WAV. Y el último bloque SELECTOR DE FIGURAS permite reproducir sectores de la partitura y permite cambiar algunas características de la partitura como tiempo de duración, clave y tonalidad.



Fig. 2 Diagrama en bloques del proyecto.

2.1 BLOQUE PRE-PROCESAMIENTO

Este bloque se encarga de adaptar las características significativas en información de la imagen de la partitura para el óptimo funcionamiento de los siguientes bloques.

2.1.1 Conversión Escala de Grises, Filtrado y Conversión a Binario. Inmediatamente la imagen se convierte en escala de grises para aplicar inicialmente el filtro “Gaussian”, para corregir ruidos, suavizando la imagen y el filtro “unsharp” para evitar desenfoques de la imagen, en pocas palabras realza la información de interés.¹¹ Seguidamente la imagen se pasa de escala de grises a un imagen binaria (negro y blanco), lo que sirve como un tercer filtro para descartar gran cantidad de información.

2.1.2 Detección y corrección de líneas horizontales. Este paso busca la detección y corrección de las líneas de los pentagramas con ángulos diferentes a cero, dado que si no se parte de una imagen con sus pentagramas bien definidos sería muy poco probable reconocer las notas musicales debido a que los pentagramas alterarían su forma y el espacio que estas ocuparían. Para esto se elimina todo tipo de notación en la partitura diferente a las líneas de los pentagramas, esto se logra eliminando las partes en donde se encuentren más de tres pixeles negros unidos de forma vertical, dejando una imagen con líneas horizontales, seguidamente se niega la imagen binaria y se etiqueta las líneas con la función de Matlab “bwlabel”. Se verifica la posición en Y del inicio y fin de una línea, si es el mismo valor

significa que la línea no tiene inclinación y por lo tanto la imagen no está rotada, por el contrario si este valor difiere indica una inclinación, se haya por Pitágoras y se saca un promedio de inclinación con todas las líneas.



Fig. 4 a) Imagen con ángulo de inclinación, b) Imagen con corrección de ángulo.

2.1.3 Extracción de características del pentagrama. Para este paso es necesario concentrarse exclusivamente en los pentagramas debido a que hay una buena parte de información, inscrita intrínsecamente en ellos. Primero se procede a encontrar el ancho de cada línea de los pentagramas, se hacen barridos verticales en diferentes puntos, buscando cuantos pixeles negros se encuentran unidos, luego se guardan en un vector para encontrar posteriormente el valor que se repite con mayor frecuencia, ese será el ancho de todas las líneas, continuo a este proceso se encuentran las ubicaciones de las líneas de los pentagramas en la partitura. Es posible que hayan quedado algunas líneas falsas como las cuadraturas, que son líneas horizontales muy prolongadas. Entonces como se conoce el ancho entre líneas de pentagrama, se halla la diferencia entre la línea siguiente y la presente, si es igual al ancho entre líneas entonces la línea pertenece al pentagrama, de lo contrario significa que es una cuadratura y se elimina, ese proceso para todos los pentagramas. Ahora se puede saber cuántos pentagramas hay en la partitura.

Por ultimo en esta etapa de preprocesamiento se agregan las tres líneas adicionales imaginarias del pentagrama, por encima y por debajo, para las notas más agudas y más graves respectivamente. Esta parte se realiza si la imagen (partitura) tiene el espacio suficiente entre pentagramas para introducir las líneas adicionales imaginarias.

Eliminación de las líneas de los pentagramas.

Una vez obtenidas todas las características de los pentagramas, se continúa con la eliminación de las líneas de los pentagramas teniendo en cuenta varias condiciones, para no eliminar pixeles de la imagen (figuras) que aportan información relevante para su posterior reconocimiento. Esto le agrega un grado más dificultad, debido a que todas las figuras musicales están inscritas entre las líneas de los pentagramas.

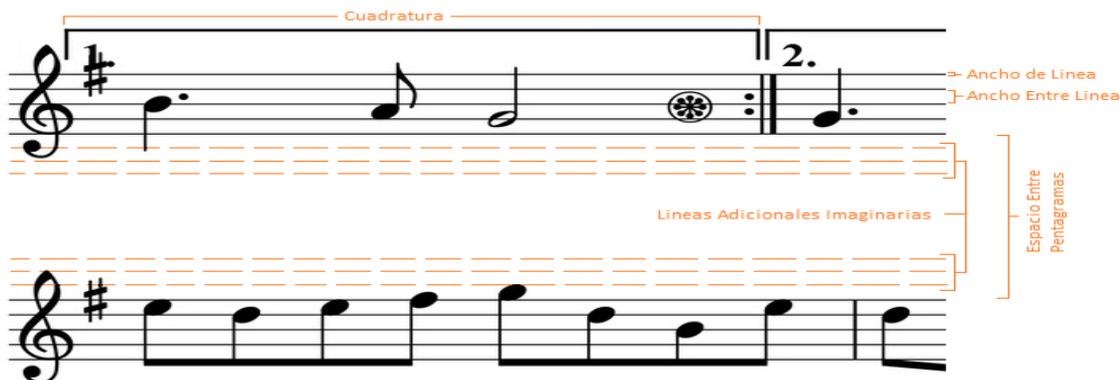


Fig. 7 Características del pentagrama

2.2 BLOQUE CLASIFICADOR

Este bloque se encarga de agrupar los objetos identificados dentro de grupos de notas (Redondas, Blancas, Negras, Corcheas, etc).

2.2.1 Imagen con símbolos aislados. Una vez obtenidas las posiciones y eliminadas las líneas de los pentagramas, se tiene solo los símbolos que son objeto de interés por ahora.

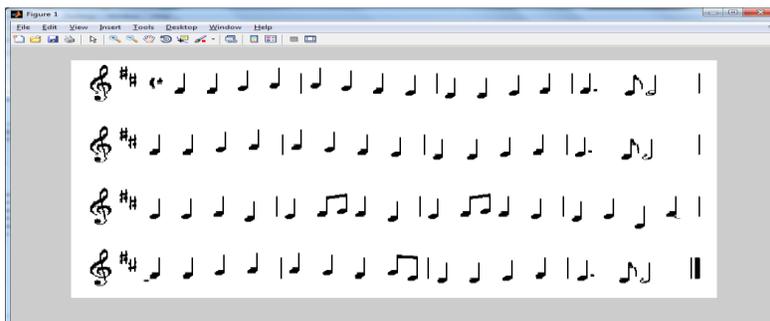


Fig. 12 Imagen con símbolos aislados.

2.2.2 Imagen Binaria y Etiquetada. Ahora se procede a negar la imagen, con el fin de etiquetar los símbolos, es decir lo negro se vuelve blanco y viceversa. Se etiqueta la imagen que contiene solo los símbolos para poder analizar cada uno de ellos con las funciones “bwlabel” y “regionprops”.

2.2.3 Comparación. Para esto se redimensiona el símbolo al tamaño de las muestras a comparar, se realiza esto hasta que encuentre un porcentaje de correlación superior al 85%, en caso de no encontrarlo se crea un vector con los porcentajes de correlación encontrados, al final se encuentra el mayor valor que corresponderá al número de la muestra y entrara al grupo de interés para ser analizado. En caso que el símbolo no tenga al menos 68% de correlación con alguna muestra, entonces se guarda la imagen del símbolo en la carpeta donde están las muestras con el fin de agregarla luego a la base de datos.

2.3 BLOQUE ANALIZADOR

Ahora es de interés ubicarla dentro del pentagrama, conocer su nota (Do, Re, Mi...), por lo cual se ha de concentrar en analizar las cabezas de nota de las figuras y se debe realizar un segundo “preprocesamiento” y “segmentación”, con el fin de trabajar solo con la cabeza de nota, el resto de información como la plica o el corchete se puede obviar o en otras palabras no es requerido. Para finalizar, el bloque Analizador devuelve la nota con el pentagrama que corresponde, la posición y la frecuencia, al bloque Clasificador.

2.3.1 Eliminación líneas verticales. Se recorre la imagen (la figura) en sentido vertical y se encuentran aquellas líneas mayores a un tercio de la altura total y se eliminan.

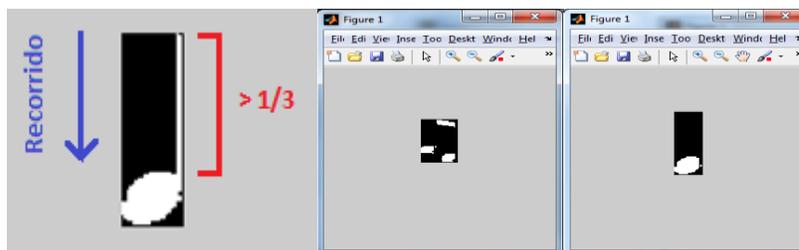


Fig. 13 Eliminación de líneas verticales o plica.

2.3.4 Factor forma y eliminación falsas cabezas de nota. Se ha denominado el factor forma a la división del ancho sobre el alto de la cabeza de nota. Con esta relación, se eliminará información no relevante para el análisis de la figura.



Fig. 14 Factor forma y eliminación de corchetes.

2.3.5 Identificación de líneas o espacios de la figura. Se debe recordar que la posición de la cabeza de nota dentro del pentagrama indicara la nota a la que corresponde la figura a analizar, por lo cual es uno de los pasos más importantes o de mayor atención. Por lo anterior se debe encontrar si la cabeza de nota se encuentra en una línea o en un espacio, además de saber el número de la línea o del espacio. Una vez despejada la cabeza de nota es de interés trabajar con su centroide y de esta forma saber la ubicación (vertical) de la figura. Primero se ubica el centroide, la posición Y, y se busca la posición de la línea de pentagrama que sea igual en posición del centroide de la figura. En el caso que la figura no este sobre una línea, se empieza a indagar hacia abajo, pixel por pixel hasta encontrar la línea del pentagrama siguiente.

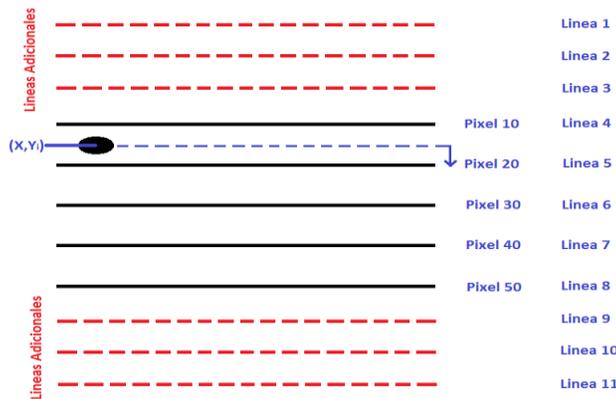


Fig. 15 Identificación de líneas o espacios de la figura.

2.3.6 Asignación y adecuación del pentagrama y sonido a la nota. En esta etapa se asignarán las frecuencias a cada nota, para posteriormente ser reproducidas. En la siguiente figura 16 se muestra el rango de frecuencias para algunos instrumentos musicales, tomando como base o referencia la escala de un piano con sus correspondientes frecuencias. Para calcular las frecuencias, se toma como base la nota “La” y se utiliza la siguiente expresión de la figura 16.

Donde f es la frecuencia de la nota, i es el desplazamiento o distancia de “La” hasta la nota de interés, siendo “La” igual a cero.

$$f_i = 440 * 2^{i/12}$$

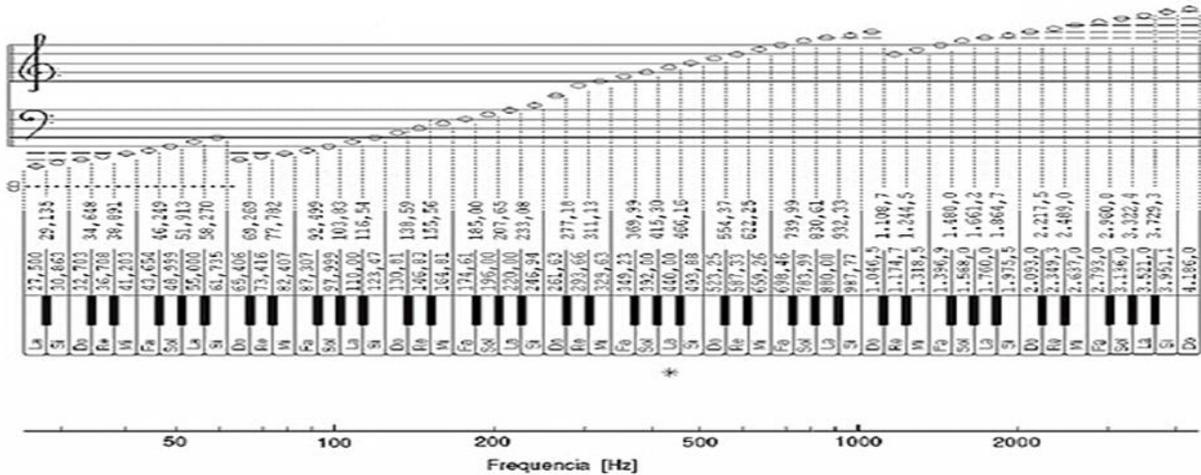


Fig. 16 Rango de frecuencias, tomando como referencia el piano

2.4 BLOQUE APLICADOR DE ATRIBUTOS

Este bloque se encarga de aplicar los atributos de las figuras de duración y de alteración que son muy importantes para obtener una melodía armónica. Si las alteraciones se encuentran acompañando la clave se modifica las notas según el número de alteraciones y de esta forma hallar la tonalidad de la canción como se muestra en la figura 17, si la alteración acompaña a una sola nota entonces la modificación será únicamente a dicha nota. Además este bloque se encarga de crear una variable dedicada al almacenamiento de las señales sinusoidales, que son las que convertirán en un formato de audio posteriormente en el bloque generador de audio.

Armadura	Tonalidad mayor	Tonalidad menor
<i>si b, mi b, la b, re b, sol b, do b, fa b</i>	do bemol mayor	la bemol menor
<i>si b, mi b, la b, re b, sol b, do b</i>	sol bemol mayor	mi bemol menor
<i>si b, mi b, la b, re b, sol b</i>	re bemol mayor	si bemol menor
<i>si b, mi b, la b, re b</i>	la bemol mayor	fa menor
<i>si b, mi b, la b</i>	mi bemol mayor	do menor
<i>si b, mi b</i>	si bemol mayor	sol menor
<i>si b</i>	fa mayor	re menor
	do mayor	la menor
<i>fa #</i>	sol mayor	mi menor
<i>fa #, do #</i>	re mayor	si menor
<i>fa #, do #, sol #</i>	la mayor	fa sostenido menor
<i>fa #, do #, sol #, re #</i>	mi mayor	do sostenido menor
<i>fa #, do #, sol #, re #, la #</i>	si mayor	sol sostenido menor
<i>fa #, do #, sol #, re #, la #, mi #</i>	fa sostenido mayor	re sostenido menor
<i>fa #, do #, sol #, re #, la #, mi #, si #</i>	do sostenido mayor	la sostenido menor

Fig. 17 Tonalidad

2.5 BLOQUE GENERADOR DE AUDIO

Este bloque es el encargado de almacenar todos los tonos con sus respectivas notas. Para ello se usa primero la duración, la velocidad de la partitura y una frecuencia de muestreo para obtener el tiempo de duración de cada una de las notas. Para este proyecto la frecuencia de muestreo seleccionada fue ocho mil

muestras por segundo debido a que permite reproducir señales de hasta 3,5 kHz la cual es suficiente para reproducir el tono de mayor frecuencia con 1,318 kHz en la escala del piano, por otra parte reduce el costo computacional en comparación con otras frecuencias de muestreo. También se agrega un tiempo muerto entre notas para que los tonos se escuchen de una forma natural y armónica para el oído humano. Por último para exportar o guardar el archivo de audio se usa una función especial la cual tiene como parámetros de entrada la variable de almacenamiento de las ondas sinusoidales y frecuencia de muestreo para generar el audio.

2.6 BLOQUE SELECTOR DE FIGURAS

El bloque Selector de Figuras permite reproducir sectores específicos de la partitura o alguna nota en particular, además permite cambiar la clave del sector que se quiera escuchar, al igual que la tonalidad.



Fig. 17 Selector de figura

2.7 INTERFAZ GRAFICA

Esta es una de las etapas más relevantes del proyecto, debido a que se desea que cualquier persona pueda hacer uso del software. Es por ello que se desarrolló una interfaz muy sencilla y amigable para el usuario como se observa en la Figura 18.



Fig. 18 Selector de figura

3. Conclusiones

La aplicación cuenta con una serie de bloques que realizan cada una de las tareas definidas por los objetivos iniciales. Se logró implementar un método eficaz para corregir el problema de la extracción adecuada de los pentagramas y las notas.

Una de las etapas más importantes del proyecto es el preprocesamiento, ya que entrega la imagen en condiciones ideales para luego ser tratada. La forma de afrontar esta etapa depende de cada autor y de los alcances o finalidad del trabajo.

Cuando se analizan o se quiere reproducir partituras en clave de Fa es difícil escuchar frecuencias tan bajas con los parlantes que traen incorporados los equipos de cómputo portátiles, por lo cual se hace necesario contar con un sistema de sonido de buena calidad.

Como se ha mencionado, el mundo de la música es “infinito” por lo cual es importante que al no reconocer símbolos, estos sean almacenados para ser agregados a la base de datos posteriormente por el usuario. Esto por supuesto implica dos cosas, la primera de ellas, que a futuro se reconozcan más símbolos y se mejore el sistema de reconocimiento e identificación. La segunda implica un coste computacional, debido al incremento en el número de muestras con las que se debe comparar para encontrar el grupo perteneciente de la nota analizada.

Hasta el momento no existe un sistema completamente automático de OMR. Debido al grado de complejidad de la música y sus distintas formas de representación e interpretación, como se pudo observar por ejemplo con el tiempo de reproducción. No todas las partituras tiene el mismo ritmo o velocidad, por lo cual es difícil conocer cuando se deben manejar unos tiempos y en qué casos otros, además de las escalas utilizadas por diversos autores.

Alguno de los objetivos alcanzados en este trabajo de grado ha llevado una gran dedicación. Cabe destacar el trabajo realizado para dar solución a.

1. Análisis de figuras y asignación de notas.
2. Asignación de tonos o figuras de alteración.

Para el primero se debió estudiar y analizar detenidamente todas las figuras musicales y de esta forma agruparlas para minimizar el costo computacional. Lo anterior está estrechamente relacionado con el bloque “Analizador”, debido que no todas las figuras musicales se deben examinar de la misma forma y no todas dan la misma información. Para esto se desarrollaron algoritmos que arrojan un resultado óptimo para considerar cumplido el objetivo.

Para el segundo, se debió implementar un algoritmo basado en diversas reglas acorde a la teoría musical, para asignar las diferentes figuras de alteración a las notas y que correspondieran con los resultados esperados.

El diseño del sistema electrónico posee algunas ventajas frente al sistema mecánico como lo son: permite el almacenamiento de las variables medidas (Bases de Datos), posee alarmas de Velocidad de Línea y Tensión de Línea acorde a los rangos óptimos de trabajo de la Unidad Slickline, tiene una salida de control para la recogida del cable

la cual se ajusta con relación a la Velocidad de Línea y al diámetro del Spooler o Carrete donde se almacena el cable de acero liso.

Debido a la escasez de ofertas similares al sistema electrónico diseñado a nivel nacional y/o regional, hace que nuestro sistema se presente como una alternativa llamativa frente a las opciones que actualmente se encuentran en el mercado, las cuales requieren de importación de equipos, la posibilidad de traer personal calificado para su implementación o adquirir una unidad completamente automatizada.

Una de las ventajas de implementación de este sistema electrónico, es la gran facilidad de acople y operación en la unidad, no afectando al sistema actual en la medida que no se requieren modificaciones al ya establecido, haciéndolo más económico frente a los sistemas, presentándose como una alternativa llamativa frente a las opciones que actualmente se encuentran en el mercado.

El diseño del sistema electrónico implementado en el prototipo a escala (Maqueta) operó correctamente durante el proceso de ensayo, demostrando el éxito funcional y adaptación del diseño sobre las variables que encierran este proceso.

Las poderosas herramientas gráficas de Labview permitieron el desarrollo de un entorno de visualización amigable, tanto de las variables que intervienen en el proceso como de los registros generados en las bases de datos, permitiendo una fácil comprensión para los usuarios.

Este sistema se encamina hacia el mejoramiento de la toma de decisiones del operador, ya que la obtención de un registro de proceso facilita bajo qué condiciones exactas de presión, tiempo de trabajo y cantidad de cable usado se encuentra operando la unidad, garantizando la información de manera detallada y estructurada del procesos. El reporte generado facilita el consolidado final de la operación sobre el pozo, convirtiéndolo en una ayuda de suma importancia para el operador.

4. Referencias bibliográficas

UNIVERSIDAD DE VALLADOLID. Acústica Musical [en línea]. Actualizada 2005. [Fecha de Consulta: 20 de Octubre de 2012]. Disponible en: <http://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io2/public_html/escalas.html>

ARMINGOL, JOSE MARIA. Preprocesamiento de Imágenes. En: Sistemas de Percepción. Madrid. Agosto 16, 2010.

BAINBRIDGE, DAVID. The Challenge of Optical Music Recognition. En: Computers and the Humanities. 2001, p. 95-121.

CARRETERO DE LA HOCHA, DAVID. Sistema De Reconocimiento De Partituras Musicales. Proyecto fin de carrera. Madrid, España: Universidad Carlos III De Madrid. Ingeniería Técnica De Telecomunicación. Especialidad: Sonido E Imagen. 2009. 203 p.

GONZALES AGUILERA, DIEGO. Procesamiento de Imágenes. Universidad de Salamanca. Master De Geotecnologías Cartográficas En Ingeniería Y Arquitectura. Castilla y Leon. 2008, p 3-23.

JOHANSEN, LINN SAXRUD. Optical Music Recognition. Tesis para el grado de Maestría en Ciencias de Modelización y Análisis de Datos. Universidad de Oslo. Facultad de Matemáticas y Ciencias Naturales. Departamento de Matemáticas. 2009, 101 p.

López Espejo, Iván. OMR y Síntesis de Partitura. En: Propuesta de Proyecto [en línea]. Actualizada en Octubre de 2012. [Fecha de Consulta: 14 de Noviembre de 2012]. Disponible en: <http://es.scribd.com/doc/110692597/6/OMR-y-sintesis-de-partituras>

PALOMINO, NORA. Técnicas de Segmentación en Procesamiento Digital de Imágenes. En: Revista de Ingeniería de Sistemas e Informática VOL 6, No 2, Julio-Diciembre, 2009, p. 10-16.

PEREZ, JOSE IGNACIO. Segmentación y búsqueda de patrones en partituras musicales. Proyecto fin de carrera. Madrid, España: Universidad Rey Juan Carlos. Ingeniería Técnica en Informática de Sistemas. 2005, p 64.

P. BELLINI, I. BRUNO Y P. NESI. Assessing Optical Music Recognition Tools. Computer Music Journal Spring 2007, p 68-93

Guerriero V. et al. (2012) "A permeability model for naturally fractured carbonate reservoirs". *Marine and Petroleum Geology* (Elsevier) 40: 115-134 doi. Web: <http://www.sciencedirect.com/science/article/pii/S0264817212002310>

Guerriero V. et al. (2011). "Improved statistical multi-scale analysis of fractures in carbonate reservoir analogues". *Tectonofísica* (Elsevier), 504: 14-24. Doi. Web: <http://www.sciencedirect.com/science/article/pii/S0040100047>

RAPHAEL, CHRISTOPHER. New Approaches To Optical Music Recognition. En. Sociedad Internacional para la Conferencia de Recuperación de Información Musical [en línea]. 2011, p 305-310. [Fecha de Consulta: 14 de Noviembre de 2012. Disponible en: < <http://ismir2011.ismir.net/papers/OS3-3.pdf> >

RODRIGUEZ ALVIRA, J. ¿Qué es una escala? [en línea]. Actualizada 2011. [Fecha de Consulta: 20 de Octubre de 2012]. Disponible en: < <http://www.teoria.com/aprendizaje/escalas/> >

REVIEW OF OMR RESEARCH. Documentación [en línea]. Disponible en: < http://evirtual.lasalle.edu.co/info_basica/nuevos/guia/GuiaClaseNo.3.pdf >

Roger B Dannenberg. Review Of OMR Research, [en línea]. [Fecha de Consulta: 26 de agosto de 2012]. Disponible en: <http://www.music.mcgill.ca/~ich/research/diss/2.%20Reviews.pdf>. ISMIR, Octubre 2006, p 2-25.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACION. Norma Técnica Colombiana 1486 Documentación: Presentación de tesis, trabajos de grado y otros trabajos de investigación. Sexta actualización. Bogotá: ICONTEC, 2008.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACION. Norma Técnica Colombiana 5613. Referencias bibliográficas, contenido, forma y estructura. Sexta actualización. Bogotá: ICONTEC, 2008.

VARIEDAD DE CURSOS GRATIS. Teoría Musical [en línea]. Actualizada 2007. [Fecha de Consulta: 26 de agosto de 2012]. Disponible en: < <http://www.aprende-gratis.com/teoria-musical/> >