

TRABAJO DE GRADO

INFORME FINAL

TEMA:

RECONOCIMIENTO ÓPTICO DE MUSICA IMPRESA EN UN ENTORNO VISUAL

Facultad de Informática

U.N.L.P.

Autoras

Cucchetti Fabiana n° 1013/2

Laborde María Victoria N° 1025/7

Directora

Mg. Claudia Russo

Co-Director

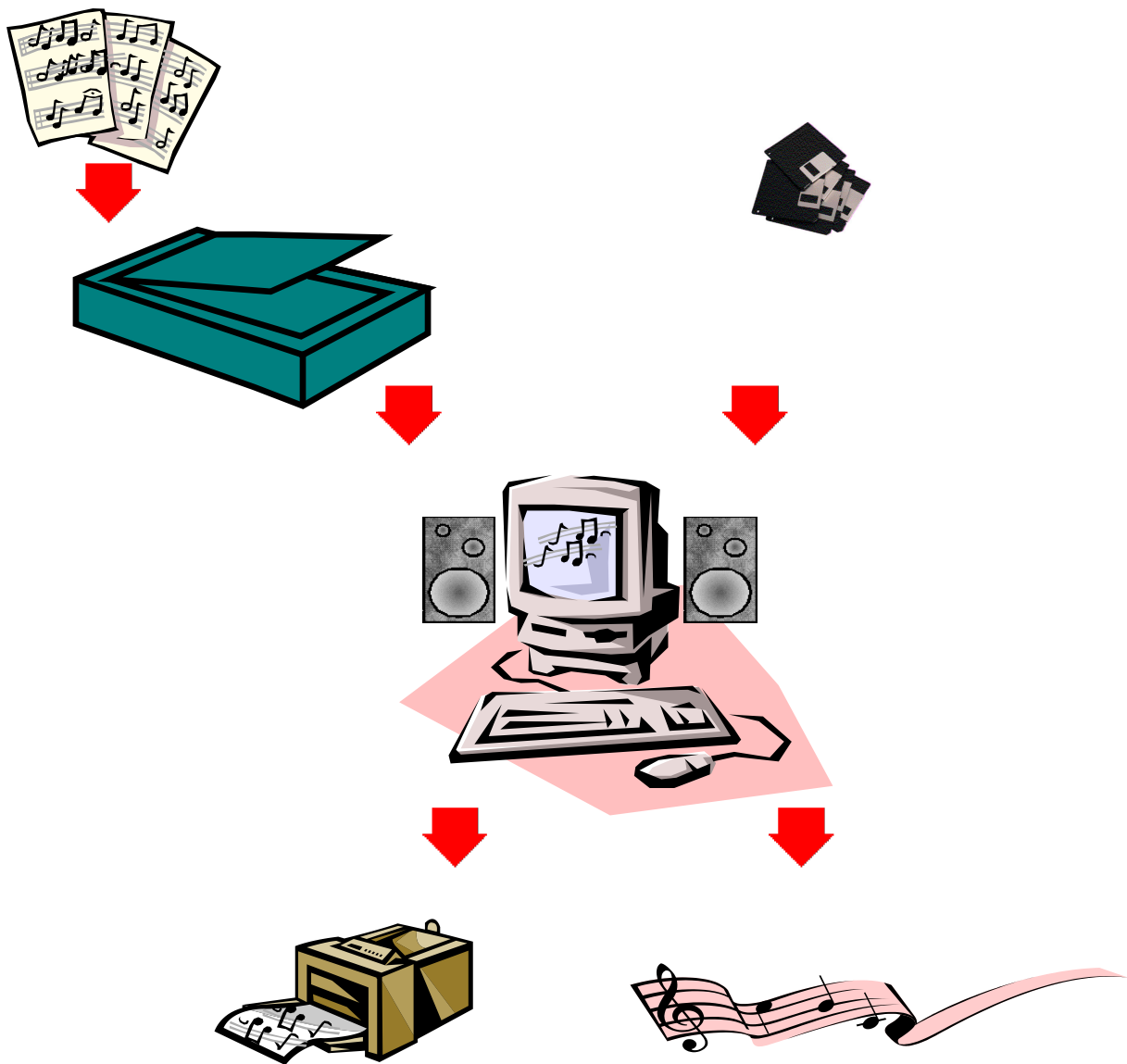
Mg. Hugo Ramón

Manifestamos nuestro reconocimiento hacia nuestra Directora de trabajo, Claudia Russo, y al Co-Director Hugo Ramón por dedicarnos su tiempo y posibilitarnos realizar este trabajo en el tiempo requerido.

Así mismo agradecemos a Xulio Fernández Hermida por su deferencia hacia nosotros.

1 INTRODUCCIÓN

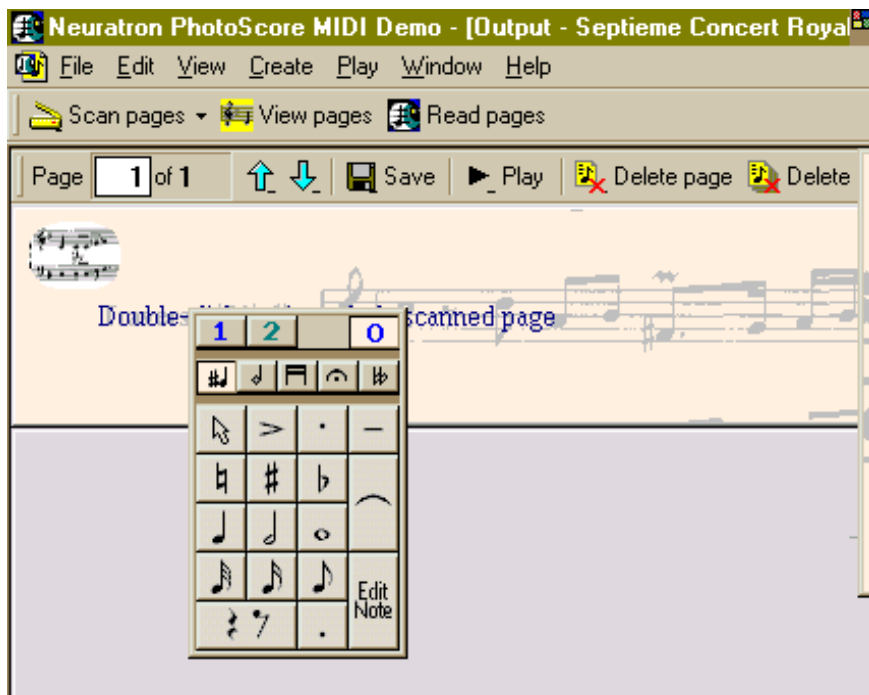
A través de este proyecto desarrollamos un software capaz de leer imágenes de partituras musicales desde un scanner o desde un disco, las cuales son procesadas reconociendo los símbolos musicales que se encuentran en ellas. Estos símbolos son almacenados en un archivo ASCII de formato sencillo, el cual es convertido a un archivo de formato MIDI (Musical Instrument Digital Interface) de modo de poder escuchar la música escrita en la partitura y editarla con algún software de edición.



1.1 Aplicaciones

Mediante la utilización de este software:

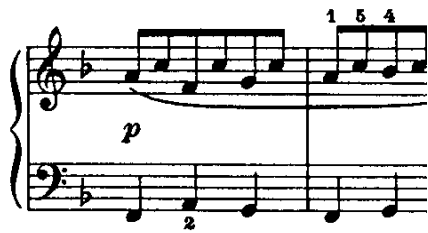
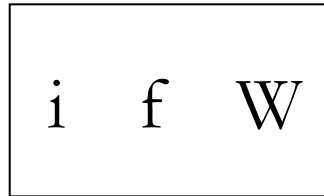
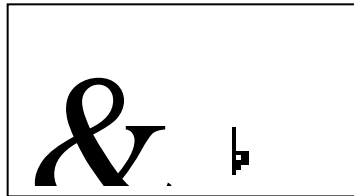
- Las personas sin conocimiento de escritura musical podrán elegir una partitura y escuchar su melodía.
- Los estudiantes de música, ante la duda de cómo interpretar una partitura podrán escucharla en forma correcta.
- Se puede lograr el almacenamiento masivo de partituras en archivos ASCII formando bases bibliográficas, que de otra manera sólo sería posible guardarlas como imágenes con el consiguiente espacio en disco.
- Al almacenarse las partituras en archivos ASCII de formato sencillo, estos pueden convertirse a notación musical Braille o a cualquier otro tipo de archivo.
- Comercialmente, en una casa de ventas de partituras musicales el cliente podría escuchar la partitura antes de comprarla.
- Dado que la salida es convertible a formato MIDI , se facilita la edición musical de las partituras. De otra forma sólo quedaba la posibilidad de copiar en editores de música, símbolo por símbolo, armando así la partitura siendo esto muy engorroso.



- Sirve para ayudar en los ensayos a los músicos, mientras ensayan solos pueden simular con el programa el acompañamiento.

1.2 Comparación con Reconocimiento óptico de caracteres

Los símbolos musicales pueden ser tratados como caracteres en algunas fuentes especiales. Las partituras pueden dividirse en grupos, los cuales son análogos a las filas de caracteres en Reconocimiento óptico de caracteres (O.C.R.). Esto quiere decir que el reconocimiento óptico de caracteres podría adaptarse a la ejecución del reconocimiento óptico de música (con introducción de fuentes para símbolos musicales), pero hay grandes diferencias entre el texto y las partituras que hacen que dicha adaptación sea difícil. La mayor diferencia está en que los símbolos musicales en la partitura son conectados a la vez por el pentagrama. Otra diferencia es que los símbolos musicales sobre la partitura pueden tener gran variación en tamaños relativos.



Un símbolo de clave de sol, puede ser mucho más alto que un símbolo plano (bemo), el cual es mucho más largo que una duración de puntillo. Para texto, los tamaños de los caracteres no varían mucho, además los textos impresos pueden arreglarse, mientras que no hay reglas estándar para la ubicación de símbolos. El espacio entre símbolos adyacentes horizontalmente puede ser arbitrario y la densidad de los símbolos sobre la partitura puede diferir desde editor a editor.

Un símbolo de clave de sol, puede ser mucho más alto que un símbolo plano (bemo), el cual es mucho más largo que una duración de puntillo. Para texto, los tamaños de los caracteres no varían mucho, además los textos impresos pueden

*midl. In H. Bunke, edito
tern Recognition (Process
and Syntactic Pattern R
Machine Perception and
Scientific, 1992.*

Existen dos grandes diferencias entre O.M.R. y O.C.R. : la *superposición* y la *adyacencia*.

Respecto a la primera diferencia, en una imagen que contiene texto no vamos a tener en ningún caso de *superposición* entre los diferentes caracteres que debemos localizar y reconocer (estamos hablando de un reconocedor de texto convencional). En una partitura en cambio, lo más regular que podemos encontrar es el pentagrama mismo (cinco líneas paralelas y equiespaciadas), pero nos vamos a encontrar con el grave problema de la superposición de símbolos. Prácticamente todos los

símbolos musicales van a estar escritos sobre el pentagrama, siendo complicado extraerlos limpiamente de él, especialmente cuando parte de las figuras son tangentes a él.

Por otra parte está el problema de la *adyacencia*. La ubicación del próximo carácter de texto en un OCR convencional puede preverse, si es que lo hay, ya que su posicionado no puede salirse de ciertos límites. Esto no ocurre en una partitura, complicándose aún más dada la falta de algunas reglas básicas respecto a la escritura musical.

Por lo antes mencionado es evidente que el O.M.R. es un problema completamente distinto al O.C.R. Las herramientas empleadas en el reconocimiento de imágenes, aún teniendo puntos de encuentro, difieren.

2 INTRODUCCIÓN A LA MÚSICA

2.1 Elementos del arte musical

Las artes se distinguieron entre sí por la diferente materia estructurada por el artista en la creación de su obra.

El compositor utiliza el sonido, la naturaleza de las leyes físicas del cual fundamentan las normas formales de la música, y sus posibilidades expresivas y estéticas. Es por eso que la teoría de este arte debe comenzar por el estudio del hecho sonoro y de las diversas formas de su producción.

Pero, una composición, no es una mera suma de sonidos simples, simultáneos o sucesivos, sino que entre ellos se establece una estrecha interconexión.

Lógicamente, la primera articulación resultante es la derivada de la relación entre dos sonidos que recibe el nombre de intervalo. A partir de ella se constituyen otras más complejas (escalas, modos, acordes, ritmos, temas, series, etc.) que pueden considerarse como los elementos básicos de la forma musical, y condicionan, por tanto, la verificación de los principios estéticos de este arte.

2.2 Sonido

La música se hace con sonidos. Llamamos sonido a la sensación auditiva que produce en nosotros el fenómeno físico originado por las vibraciones de los cuerpos.

En el sonido distinguimos diversos elementos, como la intensidad o fuerza con que se produce este sonido; la altura que nos hace considerarlo como agudo, medio o grave; el timbre, que es aquella cualidad del sonido gracias a la cual sabemos que lo que escuchamos está producido por uno u otro instrumento o voz; y la duración que nos permite apreciar el tiempo que el sonido está en nuestro oído.

Existe una distinción entre sonido y ruido. El sonido está producido por vibraciones regulares y periódicas, y el ruido por vibraciones irregulares que dan esta sensación confusa, sin entonación determinada.

Tradicionalmente la música se hacía con sonidos y no con ruidos, pero hoy en día esto no se puede afirmar. La música utiliza cualquier sonido o ruido, ya sea natural o artificial.

2.2.1 Cualidades del sonido

2.2.1.1 *Altura o tono*

Cada sonido se caracteriza por su velocidad específica de vibración, que impresiona de manera peculiar al sentido auditivo. Esta propiedad recibe el nombre de tono.

La duración de las vibraciones de instrumentos de un mismo tipo es proporcional a sus dimensiones lineales. (Ver la forma de los instrumentos para comprender mejor esto).

La medida de un sonido se encuentra fácilmente por comparación con otro parecido del cual se pueda determinar la frecuencia de una forma simple.

Los sonidos de mayor o menor frecuencia se denominan respectivamente, agudos o graves; términos relativos, ya que entre los tonos diferentes un de ellos será siempre más agudo que el otro y a la inversa.

La intensidad Es la cualidad que permite distinguir entre sonidos fuertes o débiles. Además de la amplitud en la percepción de la intensidad, influye la distancia a que se encuentra situado el foco sonoro del oyente y la capacidad auditiva de este.

2.2.1.2 *Timbre*

Si el tono permite diferenciar unos sonidos de otros por su frecuencia, y la intensidad los sonidos fuertes de los débiles, el timbre completa las posibilidades de variedades del arte musical desde el punto de vista acústico, porque es la cualidad que permite distinguir los sonidos producidos por los diferentes instrumentos.

Esta cualidad físicamente se llama forma de onda

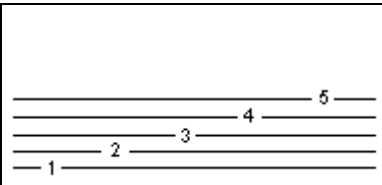
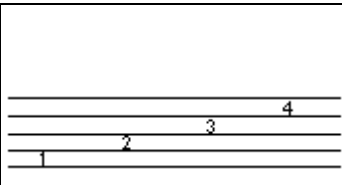
2.3 Intervalos musicales

Un intervalo es la diferencia de tono entre dos notas. Mientras el nombre de la nota expresa su tono o frecuencia de vibración, el intervalo indica la relación entre una nota y otra.

2.4 Pentagrama

La **música** es el arte de los sonidos y ésta puede ser escrita mediante el uso de símbolos musicales. Estos símbolos los escribimos en el **pentagrama**. En el pentagrama descubrimos dos dimensiones importantes:

- Dimensión Temporal: horizontalmente, a lo largo de los pentagramas; se define a la **distancia cronológica** entre eventos sucesivos.
- Dimensión Tonal: verticalmente relativamente a los pentagramas; se determina a **altura** (frecuencia de oscilación) de las notas.

El Pentagrama		
 <p>Líneas</p>	<p>El pentagrama es formado por cinco líneas horizontales, y la distancia entre cada línea se le llama espacio. Las líneas y espacios se cuentan de abajo hacia arriba y sobre éstas escribimos las figuras musicales.</p>	 <p>Espacios</p>

Cuando la cabeza de una nota se sitúa encima de la 5ª línea o abajo de la 1ª es común representar pequeños segmentos que permitan visualizar la posición relativa de la nota. Estos segmentos se denominan **líneas suplementarias**.

Los pentagramas están normalmente divididos, por barras verticales, en **compases** o sea divisiones temporales de igual duración. Los pentagramas pueden surgir agrupados en **sistemas**, unidos por barras de división de compás, e por veces por una llave en el margen izquierda, abarcando todos los pentagramas de cada sistema.

2.5 Notas musicales

El nombre de las notas musicales tiene su origen en un himno a San Juan. En concreto, en la primera sílaba de cada verso.

El himno es el siguiente:

Ut queant laxis
 Resonare libris
 Mira gestorum
 Famuli tuorum
 Solve polluti
 Labii reatum
 Sancte Joannes



Posteriormente la nota Ut, se transformó en nuestro Do.

Actualmente también se utilizan las letras A, B, C, D, E, F, G para designar las notas musicales. La letra A corresponde al La, y la letra G a la nota Sol. A partir de aquí, los símbolos se repiten y corresponden a las mismas notas una octava más alta.

Estas son las 7 notas de la escala diatónica. De cualquier forma, en una octava se utilizan 12 notas (las de la escala cromática). Las 5 notas restantes se simbolizan añadiendo a la derecha el carácter # (sostenido) o b (bemol).

Todos los sonidos en la música son representados por los siete nombres de notas: **DO - RE - MI - FA - SOL - LA - SI.**

Cada línea y espacio en el pentagrama representa un sonido musical y adquiere el nombre de una de las siete notas mediante otro símbolo musical llamado **clave**.

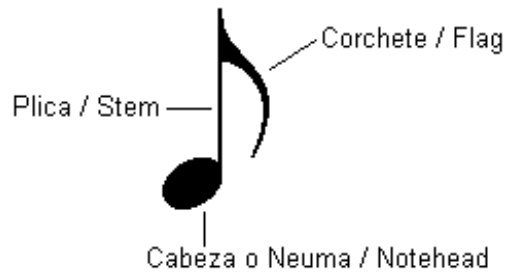
Las **claves** se colocan al principio del pentagrama y le asignan nombre a las líneas y espacios en el pentagrama.

Son 7: Clave de Sol, Clave de Fa en 3ª y 4ª línea y Clave de Do en 1ª, 2ª, 3ª y 4ª línea.



SOL	FA	DO 3ª línea	DO 4ª línea
Nombre de las Líneas		Nombre de los Espacios	
en Clave de Sol		en Clave de Sol	

2.5.1 Partes de la Nota



Para representar la duración de las notas musicales usamos las **figuras de notas**:

	Redonda		Corchea
	Blanca		Semicorchea
	Negra		Fusa

Para representar el silencio en la música usamos otros símbolos llamados **silencios**:

	Silencio de Redonda		Silencio de Corchea
	Silencio de Blanca		Silencio de Semicorchea



Silencio de Negra



Silencio de Fusa

1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
Redonda	Blanca	Negra	Corchea	Semi-corchea	fusa	Semi-fusa

2.6 Grupos de notas

Las notas que tienen corchetes pueden agruparse cuando esto ocurre se las representa unida por barra o barras según la cantidad de corchetes con que la nota se representa.



2.7 Acordes y arpeggios

Se llama **acorde** a la combinación de notas que suenan simultáneamente. Los acordes toman su nombre de la raíz o nota inicial y de los intervalos que contiene.



Cuando las mismas notas les hacemos sonar separadamente en orden ascendente o descendente se llaman **arpeggios**.



2.8 Puntillos y puntos de aumento

Tanto las notas como los silencios pueden ser prolongadas la mitad de su duración colocando un punto a la derecha de cabeza de la nota o a la derecha del silencio.

$$\text{♩.} = \text{♩} + \text{♩}$$



2.9 Accidentes musicales, tonalidades y armaduras

Los accidentes alteran la altura de las notas. El efecto de los accidentes sólo prevalece dentro de un mismo compás. Los accidentes o alteraciones pueden también surgir al inicio de un pentagrama, formando una tonalidad, agrupados inmediatamente después de la clave, a esto llamamos **armazón** o **armadura de clave**. En este caso su efecto se prolonga a lo largo de todo el pentagrama siendo posible de su anulado con la aparición de becuadros y/o con la aparición de una nueva armazón de clave.

2.9.1 Sostenido:



Sube o eleva medio tono



♯ 2.9.2 Doble sostenido

Sube o eleva un tono

♭ 2.9.3 Bemol

Baja o disminuye medio tono

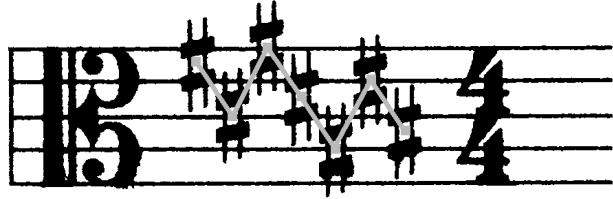
♭♭ 2.9.4 Doble bemol

Baja o disminuye un tono

2.9.5 Becuadro

Anula cualquier alteración anterior

Las tonalidad como vimos arriba es el grupo de alteraciones que puede aparecer tras la clave. Este grupo de alteraciones está formado como máximo por 7 bemoles o por 7 sostenidos, teniendo siempre el mismo tipo



de alteración (nunca habrá sostenidos y bemoles juntos). Para cada una de las posiciones y tipos de claves posibles habrá 15 combinaciones de alteraciones que puedan formar una tonalidad. Los sostenidos siempre se colocan siguiendo el orden Fa-Do-Sol-Re-La-Mi-Si. Esto quiere decir que si una armadura tuviese 3 sostenidos, serían necesariamente, y en este orden: Fa-Do-Sol. Con los bemoles ocurre lo mismo pero siguiendo la secuencia inversa: Si-Mi-La-Re-Sol-Do-Fa.

2.10 Ligaduras

Las ligaduras, surgen normalmente sobre las notas, y tal como o nombre lo indica, va ligando las notas que abarca.



2.11 Indicadores de compás

Al principio del pentagrama, se indica la dimensión (valor temporal) que deben tener los compases, delimitados por sucesivas barras verticales. Esta indicación se representa normalmente en forma de una fracción; por ejemplo a indicación $3/8$ nos dice que cada compás tendrá una duración equivalente a $3 \cdot 1/8$ de redonda, o sea $3 \cdot$ corcheas.

Los tiempos de compás mas frecuentes son representados por la letra C.

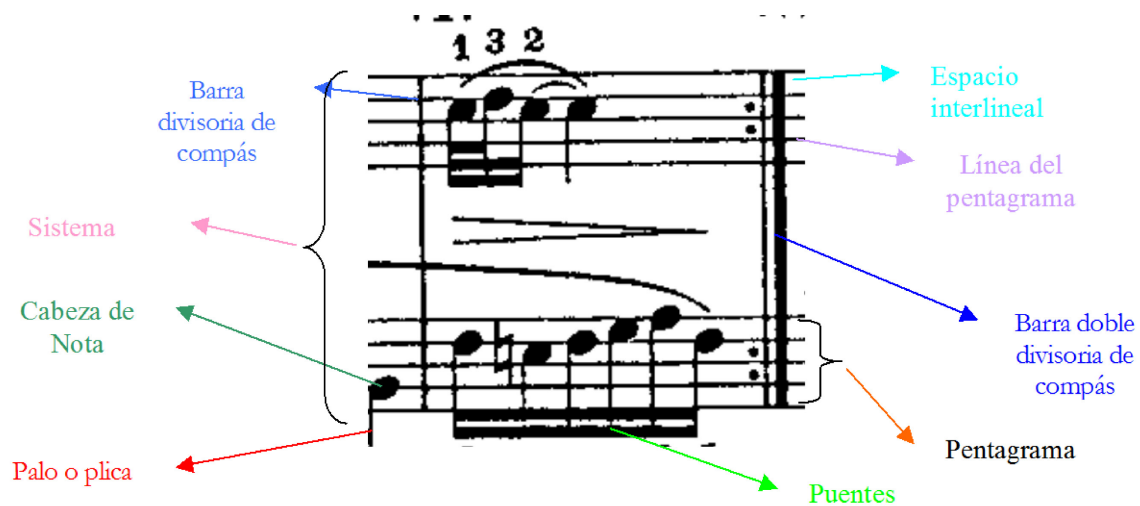
$$C = \frac{4}{4}$$

2.12

2.13 Barras divisorias

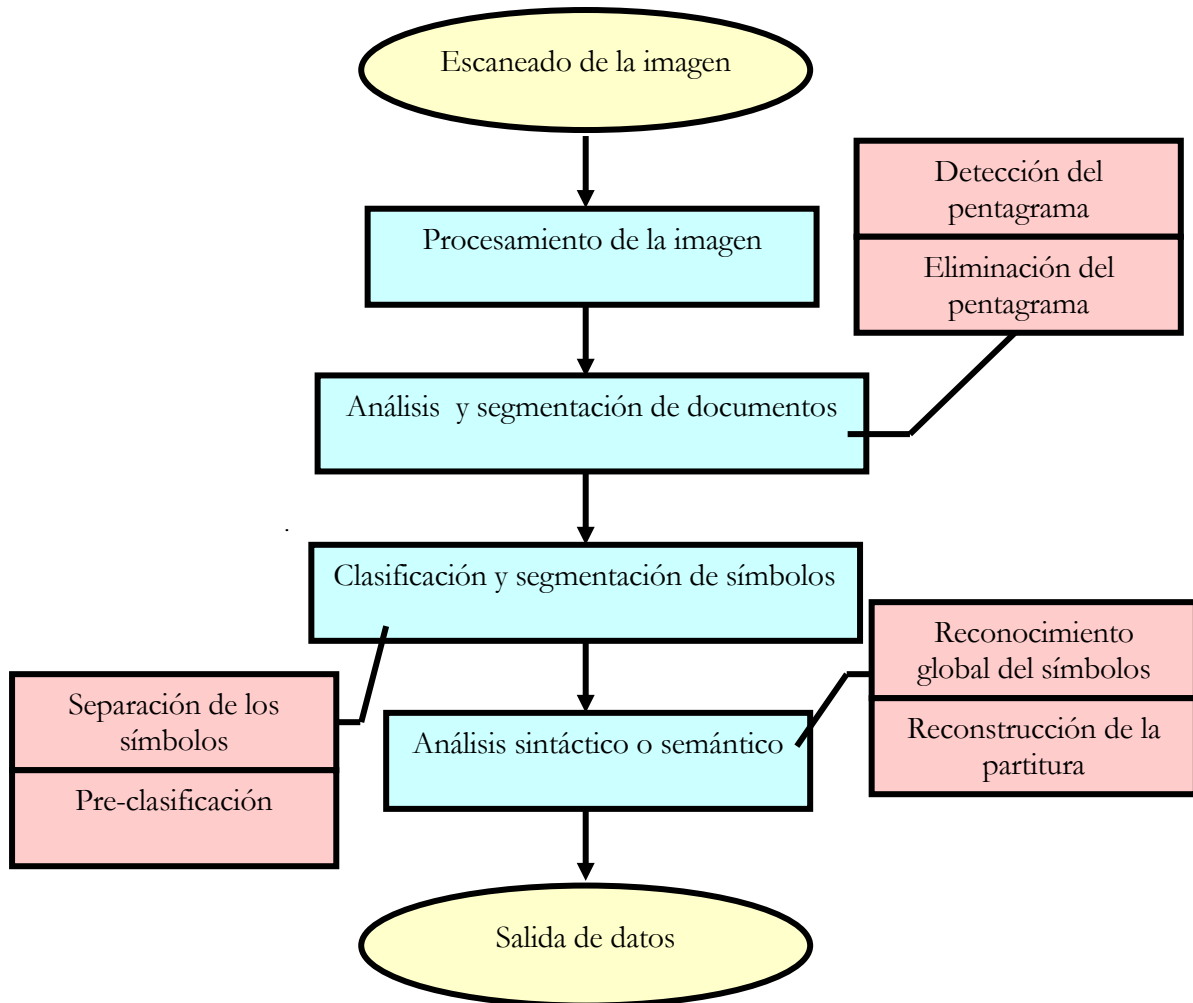
Las barras divisorias delimitan los compases. En el caso de las **barras dobles** indican la repetición de la sección.

Las **barras de compás** agrupan un número de pentagramas componentes determinando un sistema.



3 RECONOCIMIENTO DE PARTITURAS

Las etapas de reconocimiento son:



Procesamiento

El procesamiento puede involucrar casi cualquier operación de procesamiento de imagen incluyendo extracción de ruido, imágenes desdibujadas, inclinación, ajuste de contraste, afinamiento, binarización. Cualquier número de operaciones puede ser necesaria para tomar una imagen de entrada en bruto y preparar ésta para reconocimiento, pero la salida de este paso debe ser una imagen binaria para el resto del proceso de reconocimiento.

Análisis y segmentación de documentos

Antes que el símbolo de documento pueda ser clasificada, un análisis completo de la estructura del documento es necesaria. En el caso de reconocimiento de música, esto es necesario para identificar y remover las líneas del pentagrama para que sea capaz de separar apropiadamente el símbolo individual. La correcta identificación de las líneas y el grupo de ellas en pentagramas y sistemas es esencial para la clasificación de símbolos y posterior interpretación de los mismos.

Clasificación y segmentación de símbolos

Una vez que fue removido el pentagrama, los objetos son segmentados y se procede a su clasificación.

Análisis sintáctica o semántica

Estos procesos reconstruyen un documento dentro de la representación semántica desde el símbolo individual. Ejemplos incluyen combinación de barras, corchetes, y cabeza de nota en notas musicales. Este proceso depende enteramente del tipo de documento que sea procesado.

Salida

Convierte la salida de símbolos en bruto o de datos de interpretación post estructural en un formato conveniente para almacenamiento.

3.1 Problemas

3.1.1 Decoloración del documento

- Hace difícil la binarización y frecuentemente requiere algoritmos localmente-adaptables.

3.1.2 Fragmentación y pliegues en la imagen

- Líneas rotas causan problemas en la segmentación de símbolos.
- Separación o fallas en partes de algunos símbolos



3.1.3 Ruidos de alta frecuencia

- Provocado por puntos oscuros

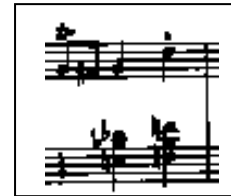
- Manchas en el papel, irrelevante en el proceso de reconocimiento.

3.1.4 Rotación o distorsión global de la imagen

- Introducida generalmente por la digitalización: el movimiento de la hoja, o la hoja mal colocada, etc.

3.1.5 Intersección de símbolos

- Símbolos pegados a otros
- Símbolos auxiliares de texto
- Símbolos que no contienen información musical pero que se pueden encontrar en la partitura.



3.1.6 Dificultades en el reconocimiento

Similitud entre símbolos de significado diferente

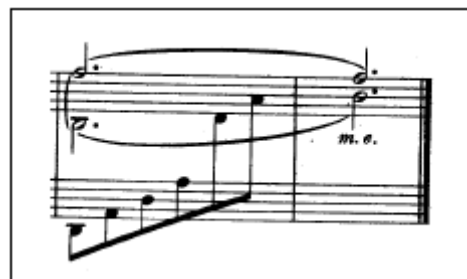
- El puntillo y el punto de aumento.
- El sostenido y el becuadro.



3.1.7 Variación de tamaño de los símbolos

3.1.8 Dificultades de Interpretación

- Voces concurrentes
- Cruzamiento de pentagramas



3.1.9 Métrica incompleta



4 TÉCNICAS USADAS POR DISTINTOS SISTEMAS EXISTENTES

Principales proyectos

Xulio Fernández Hermida y Carlos Sánchez Barbudo Vargas (1998) Reconocimiento óptico de Música Impresa. Proyecto fin de carrera. E.T.S.I.T. de la Universidad de Vigo. El objetivo fue el de conseguir un O.C.R.. específico para símbolos musicales, la finalidad no era únicamente la de reconocer las partituras sino la de poder escuchar lo que está expresado en ellas.

M. Ferrand, J.A.Leite y A. Cardoso. (1996) RIEM, (Reconhecimento e Interpretação de Escrita Musical) Desarrollado en la Universidad de Coimbra (Portugal). El objetivo del sistema fue lograr un soporte informático para la representación musical escrita. Posteriormente los autores continuaron desarrollando técnicas que mejoraran la performance en los sistemas de OMR.

Carter (1992) Propone un sistema de reconocimiento de fuentes independientemente de su tamaño y permite con una cierta tolerancia la rotación del documento inicial. Es un proyecto muy leído y sus técnicas aplicadas en varios proyectos posteriores realizado en la Universidad de Surrey (Gran Bretaña)

Fujinaga (1992) su proyecto fue desarrollado en la Universidad de Mc Gill cuyo objetivo fue un sistema de reconocimiento de partituras rápido y de precio moderado. Se integra con el editor de partituras Nutation. El sistema utiliza correcciones por parte de un humano.

Bertránd Couäsnon de la Universidad de Rennes I en 1996 realizó su Tesis para doctorado. En búsqueda de lograr un reconocimiento más efectivo. Aplicando el método DMOS (Descripción de documentos y Modificaciones de la Segmentación) que aplica un formalismo gramatical de posición que permite la descripción de documentos a reconocer y también un análisis a fin de corregir la segmentación propuesta durante la descripción del documento .

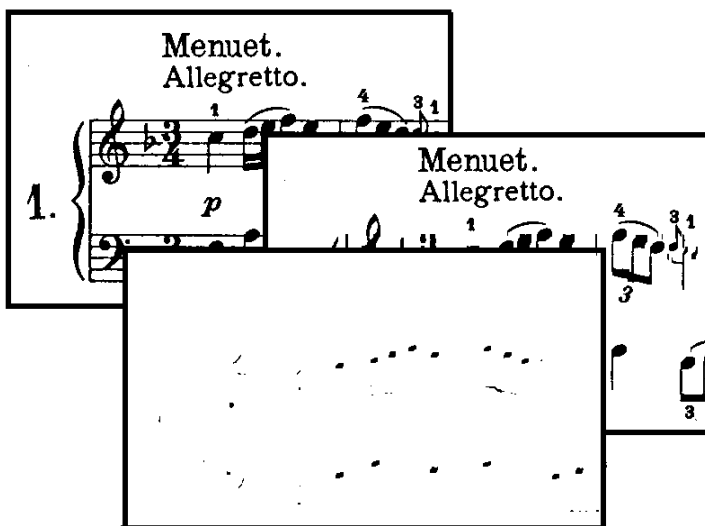
Lee Sau Dan (199) Reconocimiento óptico de música automático. Reporte de proyecto final. Se estudia la conversión de una imagen simple en forma óptica de una partitura musical en una

representación que puede ser convenientemente almacenada en computadora y recuperada para varios propósitos.

4.1 Análisis y segmentación de documentos – Detección del pentagrama

4.1.1 Xulio Fernandez Hermida y Carlos Sánchez-Barbudo Vargas

Todo el procesado de la imagen está sustentado en un entorno basado en “capas” de imágenes diferentes. La filosofía de este método se basa en mantener en memoria diferentes imágenes procesadas convenientemente, además de la original. Una de las ventajas de este modo de trabajo es que aunque se procese la imagen destructivamente siempre se puede consultar la fuente original o procesados intermedios.



El primer parámetro fundamental que se busca es el espaciado interlineal (EIL), este espaciado da la información acerca de la separación y tamaño de los símbolos que se buscan más tarde. Lo que se hace es procesar varias columnas de la imagen generando un histograma de saltos entre zonas negras de cada columna y escogiendo como EIL el espaciado o salto más común en la página. De la misma manera se busca el grosor de las líneas del pentagrama. El siguiente paso es encontrar las posiciones de las líneas del pentagrama que son localizadas recorriendo una columna de la imagen y buscando 4 zonas negras separadas verticalmente EIL, independientemente de lo que haya entre esos saltos. Se sabe así que existen P líneas de pentagramas pero podría haber ciertos errores debido al ruido en la imagen procesada. Para evitar esto se hace una búsqueda inicial en varias columnas de la imagen y se determina el número real de líneas y se halla el máximo encontrado.



Otro dato fundamental que se determina en este procesado inicial de la imagen es la inclinación de la misma. Para hallar esta inclinación se buscan dos columnas a izquierda y a derecha de la imagen que contengan el número real de líneas encontrado. Comprobando las posiciones de los primeros inicios de las líneas del pentagrama en cada una de esas columnas, comenzando por el borde superior de la página, se obtiene la inclinación de la partitura.

4.1.1.1 *Eliminación de las líneas del pentagrama*

En este punto se conocen el EIL, el grosor de las líneas, la inclinación y el número de pentagramas presentes. Se conoce dónde comienza cada pentagrama, siguiendo el algoritmo de Bresenham y basándose en un punto centrado horizontalmente

de cada una de las líneas del pentagrama, se recorren las líneas a izquierda y a derecha de ese punto, eliminando toda zona por la que pase tal que su anchura vertical sea comparable al grosor..

Existe un problema que es el de las tangencias (zonas en las que la línea que se está borrando toca tangencialmente a una figura que se superpone a ella, y cuyo grosor en esa zona es comparable al de la línea).

4.1.1.2 *Localización de las zonas a procesar*

Una vez eliminado el pentagrama, se usa una copia de este trabajo. Lo primero que se hace es conocer para cuantos instrumentos o voces está escrita la partitura, así como la delimitación de las zonas en las que se van a buscar los símbolos. Los pentagramas pueden agruparse en sistemas, de forma que varía totalmente la forma de interpretar la partitura, entonces se estudia el espaciado vertical entre cada par de pentagramas, además de si existe una o varias líneas verticales uniendo los pentagramas que forman los sistemas. En función de la diferencia de espaciados y de esa unión vertical, se puede indicar que existen N voces o líneas musicales diferentes. Luego se acota a los compases. Se analizan 2 casos:

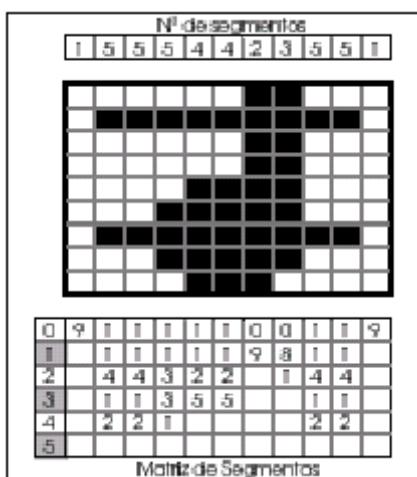
- Si existen sistemas, se realiza una proyección sobre el eje x de toda la zona que incluye cada sistema y se localizan los máximos que superan un cierto threshold. Esos puntos serán las posiciones de las barras divisorias.
- En caso de no existir sistemas, se localizan las barras divisorias tras comprobar que su proyección se halla dentro de los márgenes establecidos y que cerca de la zona candidata no se supera unos valores máximos muy bajos. Con ello se evita confusiones con las plicas de las notas.

En cualquiera de los casos se obtiene también la posición de las dobles barras. Luego se elimina de la copia de trabajo, las barras divisorias.

4.1.2 RIEM

El proyecto RIEM realiza una codificación de la imagen donde se busca reducir la cantidad de información a manipular durante el proceso de reconocimiento, permitiendo la reconstrucción sin que se pierda información. Aumento de la performance de los algoritmos.

La matriz de puntos constituida por la imagen binaria original, es analizada columna por columna. En cada columna se memorizan temporalmente la secuencia de bits encontrados que forman segmentos del mismo color, alternadamente serán blancos y negros. Para cada columna se registra el número de segmentos encontrados.



Esta segmentación permite reducir significativa la información en memoria. Esta reducción es inversamente proporcional a la complejidad de la imagen.

El primer segmento de cada columna será obligatoriamente de blancos o sea que en caso de que sea un segmento negro se pondrá 0.

4.1.2.1 Detección de las líneas del pentagrama

La detección de las líneas debe ser rigurosa e inequívoca por 3 razones:

- Las líneas perturban el proceso de reconocimiento de símbolos que se le superponen, siendo por eso necesario preservarlos. Distinguir los símbolos de las líneas y donde se superponen no permitir la fragmentación los símbolos.
- La localización de las líneas tiene un papel fundamental en la interpretación de los símbolos cuyo significado depende directamente de su posición relativa en las líneas.
- La distancia entre líneas y la espesura de las mismas permanecen sensiblemente constantes a lo largo de la partitura, siendo medidas constantes dependientes del contexto, utilizadas en los algoritmos de reconocimiento, evitando así constantes independientes o “números mágicos”.

Las líneas del pentagramas surgen en grupos de 5, equidistantes unidas por líneas verticales llamadas divisores de compás y por una llave en el margen izquierdo. Estas condiciones pueden ser levemente alteradas en el proceso de digitalización si se introduce ruido o se produce inclinación o curvatura en la imagen.

4.1.2.2 *Localización de fragmentos de las líneas*

La matriz de segmentos verticales es recorrida una única vez por columnas, procurando localizar segmentos horizontalmente contiguos que sea aprox. o inferior a un máximo establecido (MAXCOMP) que podría constituir fragmentos de líneas. Una lista mantiene, todos fragmentos “abiertos”, que son actualizados con la lectura de cada nueva columna de la imagen. Cada entrada en esa lista mantiene un registro de segmento que cumplan que su espesura media y de sus coordenadas de inicio a fin. Cada nuevo fragmento encontrado es colocado en una lista que está ordenada por orden creciente de posición vertical de los segmentos.

Una fragmentos es “fichado” es retirado de la lista cuando en una columna dada no existe un segmento contiguo (fin o interrupción de una línea), o el segmento contiguo a ese fragmento cumple que es superior a MAXCOMP (posiblemente la intersección de la línea con un objeto). Veremos mas adelante el seguimiento dado a esos fragmentos retirados de la lista.

El valor MAXCOMP es en un comienzo, muy superior al valor del espesor de las líneas, y consecuentemente de sus fragmentos constituyentes. No obstante él es ajustado dinámicamente, en función de la espesura media de los fragmentos que van siendo encontrados (mayoritariamente

fragmentos de líneas) de tal forma que su valor sea siempre sensiblemente superior a la espesura media de las líneas, localmente.

4.1.3 Proyecciones

Consiste en sumar los puntos negros horizontalmente, y de sus proyecciones observar 5 picos que indicarían la posición del pentagrama. Este método fue un comienzo utilizado por Fujinaga.

Las proyecciones Y localizadas son aplicadas para la ubicación con precisión de las líneas, de este modo: las series de proyecciones Y son tomas comenzando por el margen derecho moviéndose hacia la izquierda hasta que surgen los 5 picos claramente.



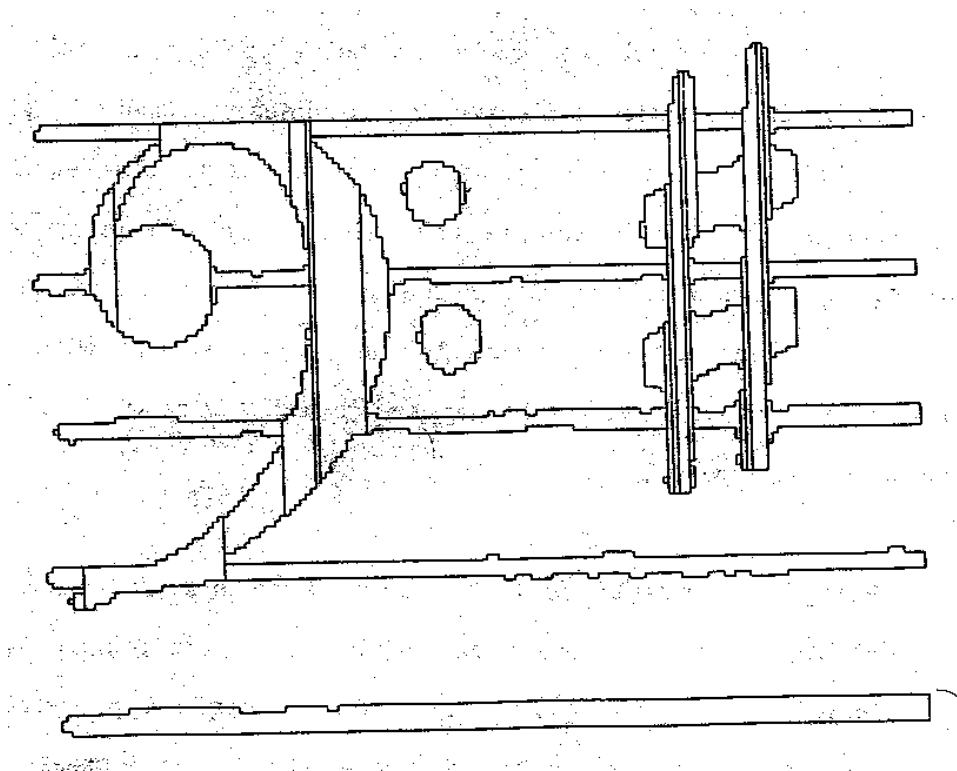
Dado que con la digitalización es muy frecuente una mínima inclinación o rotación en la imagen, y este método no era aplicable en esos casos entonces se fueron desarrollando técnicas que tuvieran en cuenta este problema tan frecuente. El valor medio de una proyección completa se utiliza como valor de threshold. En una proyección Y, el pico de la línea individual no puede ser resuelto si el pentagrama está sesgado tal que Δy es para la línea mucho mayor que la mitad del espacio entre líneas. Fujinaga acepta 5 o más picos como un pentagrama ya que las líneas sobrantes pueden ser líneas suplementarias, la barra de un grupo de notas o sesgo.

Cada grupo de picos debe estar separado de otro agrupamiento de picos por una cierta distancia para que pueda ser reconocido como un pentagrama.

Este método ubica sucesivamente pentagramas, excepto aquellos que son para percusión dada son formados por una línea simple.

4.1.4 Gráfico de líneas adyacentes (LAG)

Carter es uno de los autores que aplica la técnica de LAG la cual obtiene directamente de una codificación run-length de una imagen binaria. Dos pasadas son hechas sobre los datos durante la segmentación. La primera para producir la codificación run-length de la imagen, con la corrida de píxeles (segmentos) orientados verticalmente. La segunda pasada construye la transformada de LAG. El proceso es de izquierda a derecha recorriendo la imagen y toma pares de columnas, los segmentos son agrupados formando secciones (nodos de la transformada de LAG). Si un segmento existe en la columna de la mano derecha y no se superpone al segmento de la columna de la mano izquierda, se crea una nueva sección. Si un par de segmentos son de conexión única entonces el segmento de la mano derecha fue alocado en la sección del segmento de la mano izquierda.



La otra posibilidad es una conexión multicaminos, donde un único segmento en una columna superpone más que un segmento en la otra columna. En ésta situación la sección o secciones son terminadas (columna mano izquierda) o inicializadas (columna mano derecha). Cuando se realiza la transformada de LAG sobre la imagen de música impresa, algunos nodos del gráfico corresponden a los componentes estructurales de símbolos musicales. Para mejorar esta correspondencia se suma

una regla. Los requerimiento para que la sección procesada sea terminada es si la relación de sus espesores promedios (altura del segmento) excede 2.5. Si fuese el caso, entonces el segmento siguiente será alocado en una nueva sección.

Usando ésta técnica de segmentación resulta un seccionamiento consistente de la imagen sin hacer caso de la rotación limitada del original. El tratamiento de los datos darán un incremento significativo en la velocidad de ésta técnica sobre las técnicas de operación sobre un píxel individual.

Varios atributos de la sección son almacenados juntos tales como una entrada en una estructura de datos. Algunos de los incluidos, por ej. x-mínimo, x-máximo, y-mínimo, y-máximo, un puntero a lista de segmentos (contando las y-ordenadas del comienzo y fon de cada corrida de píxeles), el área (número del conjunto de píxeles), punteros a listas de conexiones hacia delante y hacia atrás con otras secciones, y varios totales los cuales son utilizados en los cálculos de ajustes de mínimos cuadrados.

4.1.4.1 *Extracción del ruido*

Todas las secciones con una área pequeña (λ píxeles o menos, donde λ es comúnmente 5 para imágenes de 300 o 400 d.p.i.) las cuales no están conectadas o tienen una conexión (cualquiera adelante o atrás) será removido como ruido. Como resultado de remover el ruido

4.1.4.2 *Reconocimiento de filamentos y pentagramas*

Para encontrar secciones de potenciales pentagramas se hace una búsqueda de las secciones a encontrar, utilizando el siguiente criterio:

1. Aspecto del radio (longitud / grosor promedio) $> \alpha$
2. La sección con conexión hacia delante y atrás
3. La curvatura $< \beta$

($\alpha=10$ y $\beta= 1$)

1. Las secciones que son relativamente largas y finas, consecuentemente serían incluidas como secciones de pentagramas, pero algunas de esas secciones son barras de grupos (beams), entonces para filtrarlas se forma un histograma con el grosor de los filamentos sobre la página completa y se obtiene el punto de grosor permitido. El threshold utilizado es “el medio +3 por la desviación estándar”.

2. Para eliminar las líneas que no son fragmentos de pentagramas y son líneas de (texto)píxeles prolongadas, marcas para disminuyendo y crescendo . Estas secciones están conectadas en forma simple, en cambio las secciones de pentagramas verdaderas normalmente están conectadas. Las roturas ocurridas en los pentagramas causadas por malas impresiones son bastante frecuente. Esto produce que las secciones tengan conexión simple, lo que causaría la eliminación de la sección.
3. Elimina las ligaduras que son largas y finas, pero ellas tienen un valor alto de curvatura. El threshold de curvatura es bastante alto para incluir pentagramas curvados y el valor de β dado fue encontrado para satisfacer esta meta.

4.1.4.3 *Reconocimiento de strings de filamentos y pentagramas*

Para establecer la presencia de pentagramas hay que encontrar 5 superposiciones horizontales y filamentos igualmente espaciados. Esto involucra pasar de principio a fin la lista de filamentos y compararlos mediante testeos con otros filamentos para determinar si forma o no parte del pentagrama. Los filamentos que cumplen el testeo son marcados para en un futuro eliminarlos.

Luego se mide el espaciado en los testeos de strings de filamentos y en una lista de espaciados el que mejor se ajusta es el elegido y si es posible 4 del resto de la lista con un threshold de espaciado máximo.

Después de que los pentagramas fueron encontrados, una reconstrucción de la transformada de LAG es ejecutada. Esto involucra recorrer la lista y categorizar secciones de conexiones de adelante y atrás en secciones de líneas de pentagramas y secciones de no líneas de pentagrama. Consecuentemente el registro de datos de secciones contiene punteros a las lista de conexiones, adelante y atrás, de secciones de líneas de pentagramas y fusionando INTA unir nuevamente las secciones de no líneas de pentagrama que fueron conectadas de forma simple, (ignorando conexiones de secciones de líneas de pentagrama) y que pasaron el testeo de espesor de radio (punto 1). Los símbolos musicales son así aislados del pentagrama.

4.1.4.4 *Simplificación de la transformada de LAG*

Aún después de eliminar el ruido, la transformada de LAG contiene numerosas sección que no son líneas de pentagrama, las cuales no representan estructuralmente parte significativa de la imagen.

Algunas de estas secciones las cuales realizaron el siguiente criterio fueron removidas desde la transformada de LAG.

- I. El aspecto vertical del radio $\geq \delta$
- II. Las conectadas en forma simple
 - a. Tiene un rectángulo más largo o igual que el testeado de la sección.
 - ó
 - b. Tiene un aspecto de radio vertical $\geq \delta$ y ésta área es mayor que la del testeado de la sección.

El valor de δ empleado fue 5.

/* esto es escrito por Cuasnon

El método de líneas adyacentes propuesto por Carter, trata de distinguir las partes de líneas del pentagrama rotas y las que se superponen con un símbolo. La dificultad más grande es la detectar los símbolos que son tangentes a las líneas del pentagrama. La técnica utilizada permite reconocer las hojas que han sido rotadas hasta 10° , para lograr esto Carter desarrolló un gráfico de líneas adyacentes. Esto es construir segmentos y reagrupar los segmentos en secciones y en funciones. Una sección esta constituida de una serie de segmentos verticales conexos que tienen un grosor relativamente parecidos. Una función es creciente cuando varios segmentos son verticalmente conexos a una columna precedente, o cuando es una ruptura en el grosor. Dentro de las secciones que pueden pertenecer a partes del pentagrama se ven partes potenciales del pentagrama llamadas filamentos. El máximo en el histograma de grosores de los filamentos se calcula a fin de tener una estimación del grosor de las líneas del pentagrama. Esta información le permite suprimir en los filamentos, las barras de grupos paralelas a las líneas del pentagrama. Los filamentos son luego concatenados en cadenas modificadas a fin de la integridad de las líneas del pentagrama. Las secciones conexas y no etiquetadas de líneas son reagrupadas para formar los objetos. Las secciones etiquetadas son marcadas en el gráfico. Obteniendo así una segmentación de los objetos musicales. Esta segmentación altera un poco menos los objetos que los otros métodos, la clave de Fa por ejemplo, subsiste más las separaciones cuando los objetos están tangentes a las líneas del pentagrama. */

4.1.5 Lee Sau Dan

Se realiza un preprocesamiento en donde se determina el grosor y el espaciado de la línea del pentagrama antes de construir el LAG¹. La ventaja de este preprocesamiento es que el valor del Threshold se basa en estos dos valores. El grosor de la línea será el valor más frecuente de conjuntos de píxeles negros ocurridos luego de una corrida vertical. De la misma manera se obtiene el espaciado de las líneas, teniendo en cuenta los píxeles blancos en lugar de negros.

4.2 Clasificación y segmentación de símbolos

4.2.1 Xulio Fernández Hermida y Carlos Sánchez-Barbudo Vargas

Se realiza un proceso por compases, de modo que hasta que no se han completado el reconocimiento de todos los símbolos implicados en un compás, no se procesa el siguiente. No todos los compases necesitan el mismo estudio. Por ejemplo el primer compás de cada pentagrama contiene información que no está en las demás como la clave y la armadura.

Lo primero que se hace es localizar la clave. Se proyecta sobre el eje X el compás en el que se busca la clave, se localiza una proyección de una determinada anchura al comienzo del compás. Posteriormente se acota la figura encontrada verticalmente, tras lo que se comparan ciertos parámetros de la ventana segmentada con datos almacenados previamente, tales como tamaño (relativo al EIL), posicionado respecto al pentagrama y zonas de la ventana en blanco. Así, se separan las tres claves y también su posición.

Seguidamente se localiza la armadura, se consideran las quince armaduras diferentes, se procede a localizar la primer alteración y si existe la armadura se sigue la secuencia de posiciones que deberían ocupar las siguientes alteraciones, hasta finalizar las que están a continuación de la clave. La segmentación de las alteraciones se consigue mediante proyecciones sobre los ejes X e Y.

El siguiente paso consiste en encontrar las cabezas negras de las notas y todos los símbolos asociados a cada una de ellas. Se realiza una erosión de profundidad dos veces el grosor hallados para las líneas del pentagrama, así se elimina gran parte de los símbolos y quedan los centros de las

¹ La construcción del LAG se realiza como se describió anteriormente en Carter

cabezas negras. Corrigiendo la posición que ocupan, por la inclinación del pentagrama, se conoce qué nota es cada centro.

Para extraer la duración de cada nota, es buscada en la capa de trabajo correspondiente, la plica de esa cabeza en las posiciones posibles según las reglas de escritura musical. A continuación con la ayuda del adelgazamiento (thinning) y del seguimiento de líneas se localiza el final de la plica, generando una ventana de ancho próximo al EIL que contiene la plica y a los corchetes que dan la información de la duración. A esa ventana se le hace una erosión horizontal de profundidad el grosor del trabajo con lo que se queda con una serie de transiciones verticales (blanco, negro, blanco) que son los corchetes. Sólo con contarlos se sabe si es corchea, fusa, etc.

Por otra parte estudiando la zona que está delante de cada una de las cabezas, se puede averiguar si existe un conjunto de píxeles en negro, de ciertas dimensiones, susceptibles de ser una alteración. Del mismo modo se encuentran los puntillos, pero la zona de búsqueda será después de la cabeza.

De una nueva copia de trabajo, se irán eliminando gran parte de los símbolos reconocidos en esta fase. A partir de aquí se buscan los símbolos que faltan, como las blancas y las redondas.

4.2.2 RIEM

Según el proyecto RIEM, separar o aislar todos los objetos / símbolos de la imagen, que en su mayoría están superpuestos al pentagrama a las líneas del pentagrama, utilizando para esto la información sobre la localización de las líneas. Lo que busca es definir el contorno de cada objeto individual. Para esto necesita remover las líneas del pentagrama y para esto se define que se admite que un segmento pertenece a un pentagrama si cumple que:

El segmento $<$ espesura media de la línea * MAX_VAR_ESPES_LÍNEA

Distancia de punto medio del segmento a la línea $<$ espesura media de la línea

4.2.2.1 *Construcción de contornos*

La separación de objetos ocurre en simultaneo con la creación de los contornos envolventes de esos objetos. Algunos símbolos musicales no pueden ser identificados sólo por el contorno, presentan agujeros internos que corresponden a contornos internos.

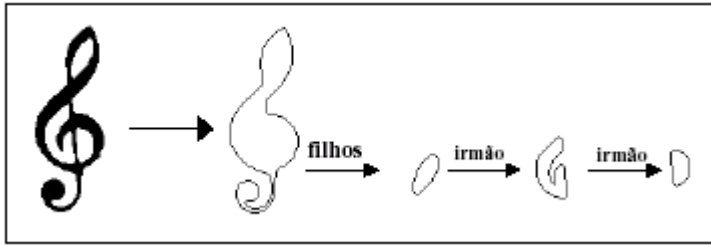
Esta información adicional tornará mas complejo el modelo de análisis de reconocimiento, no obstante es utilizada como auxiliar en la pre-clasificación de símbolos, y de forma determinante en la interpretación de esos símbolos.

La relación entre los distintos contornos de un objeto se representa en árboles binarios. Los padres son los contornos externos y los hijos los contornos internos.

La matriz de segmentos de la imagen es leída columna por columna, de izquierda a derecha procurando segmentos verticales adyacentes horizontalmente que formen los objetos. Son considerados los segmentos no pertenecientes a las líneas del pentagrama.

4.2.2.2 *Abertura de un contorno*

Cuando surge una columna en un segmento vertical que no tiene ningún segmento contiguo de la columna anterior, se inicia un nuevo contorno externo, etiquetado con un número secuencial (identificador) y son almacenados sus extremos abiertos en una lista de contornos abiertos.



Los extremos de un contorno son actualizados a medida que nuevas columnas de segmentos son leídas, y en función de la posición de los nuevos segmentos contiguos. La apertura de un contorno interno se procede de forma análoga, y sus extremos abiertos son igualmente insertados en una lista de acuerdo a su posición relativa.

4.2.2.3 *Fichado de un contorno*

Cuando se ficha un contorno, sus extremos son unidos y retirados de la lista de contornos abiertos. Antes, se inspecciona cual es el contorno padre que lo envuelve, y se registra el número de identificador. El contorno es entonces colocado en una lista de contornos fichados, y recorre de esa lista todos los contornos ya existentes cuyo identificador de padre es el de contorno insertado. Es importante aclarar que ningún contorno puede fichar sin que todos sus hijos se hayan encontrado.

4.2.2.4 *Unión de dos contornos*

Si ocurre la unión de las extremidades de dos contornos distintos, ellas son unidas y retiradas de la lista de contornos abiertos. Las restantes extremidades serán las extremidades de un nuevo contorno único que adoptará el identificador del primer contorno abierto.

La clave de este sistema es que asienta una correcta actualización de las listas de contornos (abiertos y fichados) después de la unión. Después de la unión podrán existir contornos fichados, cuyos padres es un contorno que deja de existir, pero que actualiza todos los identificadores respectivos.

Por convención todos los contornos exteriores corren en sentido antihorario y los interiores en sentido horario, de forma de facilitar la ligación entre sus cadenas de vectores, de una unión de dos contornos.

Un nuevo objeto estará completamente formado cuando el identificador de padre de contorno que ficha, es cero, o sea la raíz de todos los objetos de la imagen.

4.2.2.5 *División de contornos en primitivas gráficas*

Divide los contornos de cada objeto en segmentos de curvatura constante (que serán clasificados en primitivas gráficas tipificadas), permitiendo descubrir claramente su geometría. Se toma también más fácil la descomposición de algunos objetos compuestos (en especial notas) posibilitando su reconocimiento genérico.

La curvatura se calcula con la siguiente fórmula:

$$k(t) = \frac{x'(t).y''(t) - x''(t).y'(t)}{(x'(t)^2 + y'(t)^2)^{\frac{3}{2}}}$$

Dado el valor discreto de los valores (x,y) de un contorno introducen grandes variaciones en el valor de las derivadas, por lo que se realiza un pre-alisamiento de contornos que reduce las oscilaciones.

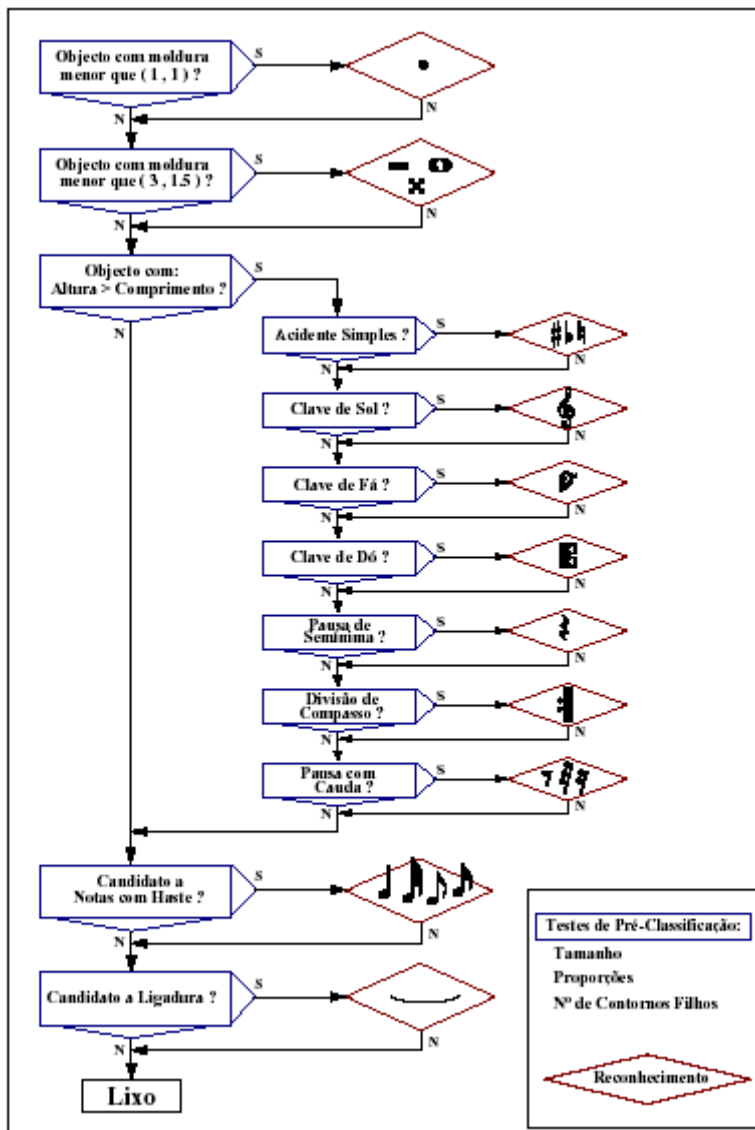
Los segmentos son registrados en una lista circular (respetando el orden natural de contorno), y son etiquetados como primitivas gráficas de tipo ARCO, SEGMENTO VERTICAL, SEGMENTO HORIZONTAL O SEGMENTO OBLICUO de acuerdo con sus características. Cada primitiva en esa lista registrará también otros parámetros característicos como: curvatura media, moldura, declive y sentido (en relación al eje cartesiano de mayor variación).

4.2.2.6 *Localización de objetos*

Después de separar los objetos de las líneas, cada objeto es localizado en relación a los pentagramas. Se pretende determinar a qué pentagrama pertenece el objeto. Existen algunos casos en que esta localización requiere un tratamiento particular:

- Barras de división de compás
- Objetos no pertenecientes a la imagen musical (números de página, títulos, etc.)

4.2.2.7 Pre-clasificación de objetos



La clasificación o reconocimiento de objetos musicales no recurre a un método genérico único. La existencia de múltiples dimensiones y de objetos compuestos obliga a la utilización de diversos procedimientos con algoritmos complejos.

Se procura dividir los objetos en grupos o clases con características globales semejantes. Los criterios de pre-clasificación son los siguientes:

- Área de moldura envolvente del objeto
- Proporciones de moldura envolvente (Largo vs. Alto)
- Números de contornos interiores (o agujeros)

Las medidas utilizadas son: El espacio medio entre líneas y el espesor medio de las líneas.

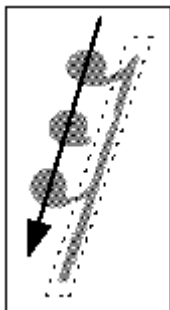
4.2.2.8 Reconocimiento de objetos

En la elaboración de procedimiento de reconocimiento distinguimos dos dominios:

- Conocimiento de notación musical: para reconocer cualquier símbolo es necesario conocer las variaciones de forma, tamaño o posición que él puede tomar.
- Conocimiento de efectos de ruido y de procesamiento de imágenes: algunas fases de procesamiento de imagen pueden resultar en deformación y/o fragmentación de algunos objetos. Es sabido cómo y cuáles son afectados algunos símbolos. El objetivo será prever los casos más frecuentes de distorsión para cada tipo de símbolo de forma de analizar sus particularidades en un contexto de cada caso posible.

4.2.2.9 Técnicas de reconocimiento utilizadas

- Identificación de secuencias de primitivas gráficas particulares: permite detectar formas geométricas elementales de los símbolos mediante la secuencia de primitivas.
- Análisis del tamaño y posición relativa entre varias primitivas: permite reconstruir partes elementales de los objetos cuando son interceptadas por otros elementos del mismo objeto.
- Análisis de localización y tamaño de contornos internos: permite distinguir objetos con contorno exterior semejante tales con la blanca y la negra.
- Proyecciones localizadas: con el propósito de reparar objetos frecuentemente fragmentados (especialmente debido a la superposición en líneas), se recurre a proyecciones de pequeñas áreas localizadas de la imagen. De acuerdo a cada caso particular se analiza:



- Área proyectada vs. Área de la proyección
- Números de bloques distintos y en cada bloque: área, largo y alto de los bloques
- Números de picos para un máximo dado
- Números de máximos o mínimos locales

4.2.3 Fujinaga

Las proyecciones X e Y son suficientes para identificar símbolos musicales aunque ellas pueden solamente establecer la forma y el tamaño aproximado de los símbolos. La estrategia básica empleada por Fujinaga es ubicar símbolos usando proyecciones o conocimiento sintáctico, entonces calcula proyecciones locales para detallar la clasificación del símbolo.

Luego de localizar el pentagrama y determinar la posición de las líneas con proyecciones Y, ahora se usa una proyección X del pentagrama para ubicar individualmente los símbolos de música. Esta proyección es del núcleo del pentagrama, es decir, el área entre la primera y última línea del pentagrama. Mientras ésta proyección corta los símbolos que se proyectan por encima o por debajo del pentagrama, esto es suficiente para ubicar símbolos. Las líneas pentagrama dan en sí mismas un valor de proyección de fondo en la proyección X; los símbolos son identificados en cualquier momento en que el valor de la proyección X exceda el valor de fondo por un espacio de pentagrama. En este punto realiza una proyección local Y, cubriendo la altura completa del pentagrama mejor que el pentagrama núcleo. Esta proyección Y se emplea para determinar la extensión vertical del símbolo; finalmente se toma una proyección X usando este límite vertical. Entonces las características del símbolo son calculadas: ancho, alto, área, número de picos en la proyección X. Estas características son usadas en conjunto con el conocimiento sintáctico para la clasificación del símbolo.

Algunas clasificaciones de símbolos tal como los indicadores de compás no son reconocidos usando proyecciones y la distinción entre sostenido y becuadro no está hecha.

En trabajos posteriores con proyección basada sobre clasificación en un espacio de características.

Por lo tanto este trabajo demuestra que las proyecciones proveen una eficiente medida para la ejecución de la clasificación inicial.

4.2.4 Carter

Usa la transformada de LAG para separar líneas de símbolos musicales, y describe objetos que corresponden a símbolos musicales o componentes conectadas de símbolos musicales. Esta segmentación resultante es interpretado por el sistema de reconocimiento. Los objetos resultantes de la segmentación son clasificados de acuerdo al tamaño de su “caja” y de acuerdo al número de organización de sus secciones. Carter nota que los símbolos superpuestos deben ser tratados por un algoritmo específico.

4.2.4.1 *Formación del objeto*

Un objeto es considerado un grupo de secciones conectadas, sus detalles se almacenan en una estructura de datos. Los objetos se encuentran recorriendo la lista de secciones, ignorando secciones pentagrama, y se realiza siguiendo el método del primero en profundidad a través de la transformada de LAG. A cada sección que no ha sido incluida hasta ahora en un objeto. Por cada objeto se establecen varias estadísticas y son almacenadas como miembros de la estructura de datos del objeto. Esto incluye las coordenadas del rectángulo, el área (suma de las áreas de las secciones) y el número de secciones que comprenden al objeto.

4.2.4.2 *Ordenación de símbolos*

Cada objeto en la lista de objetos es ordenado con respecto a la página completa. Cada objeto también está asociado con uno o más número de pentagramas, usando estos números los símbolos deben ser organizados por pentagrama y ordenados de izquierda a derecha.

Un grupo separado de objetos contiene aquellos objetos que no están en contacto con las líneas del pentagrama, por ejemplo algunas ligaduras.

4.2.4.3 *Reconocimiento*

Símbolos simples tales como negras, líneas de compás, accidentes y silencios de corcheas deben ser reconocidos por el tamaño de su rectángulo y el número y organización de sus secciones. El tono de la nota debe ser encontrado examinando el número de pentagrama de las secciones de líneas de pentagrama conectadas a la sección de la cabeza de nota o la posición vertical del centro de la sección de la cabeza de nota con respecto a la sección local de la línea de pentagrama. Este caso es considerando la posibilidad de rotura entre la cabeza de nota y la línea de pentagrama.

Existen modelos que son usados en conjunto con reglas los cuales especifican dimensiones relevantes u organización de espacio para las secciones. Cada modelo contiene el número de secciones las cuales constituyen la forma apropiada del símbolo, conjuntamente con el número de conexiones hacia delante y atrás para cada sección. Esta forma de modelo solamente garantiza una correspondencia uno a uno entre el modelo y el tipo de símbolo.

Un objeto que consiste de dos secciones se clasifica como una negra . Si la sección más a la izquierda tuvo un aspecto vertical del radio alto, se estima que es un tallo (barra de nota) y la negra

tiene el tallo hacia abajo en caso contrario donde la sección más a la derecha es el tallo entonces la dirección del tallo es hacia arriba.

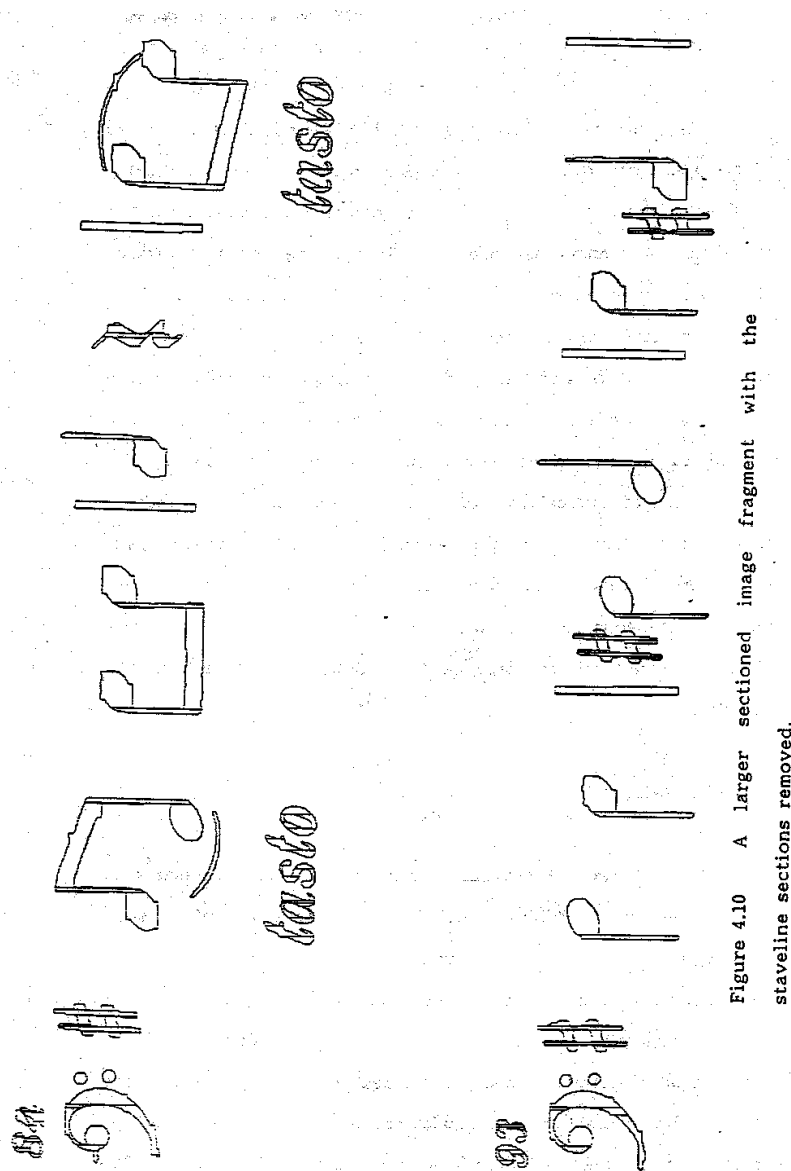


Figure 4.10 A larger sectioned image fragment with the stave line sections removed.

El número de conexiones hacia delante y hacia atrás de cada sección son usadas para distinguir entre el sostenido y el grupo de 3 corcheas. Si el aspecto vertical del radio es alto en la sección más a la izquierda del grupo de 3 corcheas indicado, entonces el tallo de las notas es hacia abajo. Si ningún conjunto de conexiones fue apropiado y el ancho del rectángulo fue igual a la altura del espacio del pentagrama en +/- 20 % mientras que la altura fue entre la suma de 3 y 4 alturas del espacio del pentagrama, el objeto se clasifica como un silencio de negra.

El bemol o el becuadro provee su dimensión vertical la suma entre 2 y 3 altura de espacio de pentagrama. El radio de la altura promedio de la sección mas a la izquierda y la sección más a la derecha se usó para distinguir los dos símbolos.

Para distinguir entre el silencio de corchea y un grupo de dos corcheas se usa la lista que contiene el número de ce conexiones hacia delante y hacia atrás para cada sección. Para detectar la dirección del tallo del grupo de dos corcheas se usa el mismo método que el descrito para el grupo de tres corcheas, y un chequeo opcional para detectar la presencia de cabeza de notas, usando el promedio de grosor.

Los grupos de notas son examinados por un algoritmo distinto debido a los proceso de impresión.

Los símbolos superpuestos o sobrepuestos también serán examinados por un algoritmo específico.

4.2.5 Lee Sau Dan

Una vez aislados los símbolos a través de LAG, se buscan las barras de compás examinando el tamaño de su bounding box. Una componente con un bounding box descrito como tres veces el grosor de una línea y 4 veces el espaciado del pentagrama se clasifica como una barra de compás.

Una vez detectadas las barras de compás se examina cada compás. Esto permite trabajar más rápidamente y requiere menos memoria. Se realiza una proyección horizontal de cada componente, donde se buscan los picos con un ancho no mayor que la mitad del espaciado. Esos picos sugieren la presencia de notas con plica. Se examinan los lados izquierdo y derecho, comenzando con el izquierdo. Mediante una proyección horizontal se chequea si existe la presencia de una nota cabeza. En este punto se fija si el bounding box es aproximadamente cuadrado, con el longitud del lado aproximadamente igual al espacio entre líneas. Luego se examina el área excede las tres cuartas partes del bounding box. Si todas estas condiciones se satisfacen entonces es una nota cabeza. Habiendo encontrado una nota cabeza se desplaza al otro extremo de la plica para encontrar un corchete.

4.3 Análisis sintáctico o semántico

4.3.1 Xulio Fernández Hermida y Carlos Sánchez-Barbudo Vargas

Un módulo gestiona la correcta sintaxis de los símbolos reconocidos. Este módulo realimenta ciertos datos a otras partes de la aplicación avisando por ejemplo si en un compás faltan figuras o silencios que reconocer (por sumas de tiempos).

4.3.2 RIEM

Los símbolos reconocidos necesitan un tratamiento intermedio entre la imagen de entrada y un archivo de salida. Teniendo en cuenta su significado musical surgen así dos lenguajes.

1. Representación orientada a primitivas
2. Representación orientada a eventos

4.3.2.1 *Representación orientada a primitivas*

Se divide en dos grupos desde el punto de vista gráfico:

- Símbolos elementales: son los que alteran la característica de una nota (accidentes, puntillos, ligaduras, etc. También incluyen silencios de redonda, blanca y negra).
- Símbolos compuestos: son los silencios de menor duración y las notas o grupos de notas.

Como forma de obtener una descripción por objeto se construyen árboles de primitivas que llevan orden jerárquico y dirección unívoca. Luego cada árbol pasará a formar parte de una red de árboles como forma de descripción de los compases.

Este método es relativamente eficiente para partituras simples, lo que implica que cuando se aplica a casos más complejos se introducen problemas imposibles de resolver y consecuentemente se estanca el proyecto.

4.3.2.2 *Representación orientada a eventos*

Desde el punto de vista de su significado musical, una partitura representa una secuencia de acontecimientos que son caracterizados por dos atributos fundamentales:

- La altura
- La duración

Estos acontecimientos serán designados por eventos. De cada objeto se extraen los eventos que los constituye, se determina su localización en la partitura que representa el instante en que el evento tuvo lugar y se detectan sus características en términos de altura y duración.

A partir de esta separación será necesario mantener información que relacione todos los eventos extraídos de un mismo objeto, para que posteriormente puedan ser reagrupados por el editor de música.

Este método posibilita una representación de voces concurrentes de una forma simple. Facilita el proceso de reconocimiento pues las características de un evento se hace de una forma mucho más localizada, con un grado de certeza mayor.

Para realizar un punto intermedio entre la imagen de entrada y el fichero de salida se escoge este método.

4.4 Salida de datos

4.4.1 Xulio Fernández Hermida y Carlos Sánchez-Barbudo Vargas

Aunque la salida final del programa es un fichero standard MIDI, se implementó la generación de un archivo ASCII intermedio, con un formato sencillo. Esto permite una conversión de éste formato a música Braille o cualquier otro tipo de fichero.

4.4.2 RIEM

Este proceso está dividido en las siguientes etapas:

- Clasificación de notas: determina la posición relativa de una nota en el pentagrama en lo que se refiere a línea o espacio entre líneas al que pertenece. Esta clasificación está hecha en función de las coordenadas de la nota y de las coordenadas que constituyen el pentagrama.
- Clasificación de alteraciones: los accidentes y puntillos se consideran alteraciones y tienen la atribución de estar situados en la misma línea/espacio que la nota a la cual alteran.

- Atribución de alteraciones: sirve para asociar las alteraciones a las notas correspondientes y para separar los accidentes que forman parte de una armadura de clave.
- Ordenación de eventos: la finalidad es la transformación de una lista desordenada de eventos en una secuencia de eventos ordenada temporalmente. Debido a la existencia de varias voces en el pentagrama se prevé la existencia de eventos simultáneos y por eso se crea una estructura bidimensional.
- Preprocesamiento de eventos:
 1. Determinación de la duración de un evento
 2. Determinación del número de voces
- Separación de voces: se optó por hacer una separación que no siendo correcta en términos de concepto musical de voz, permite pasar toda la información necesaria a edición y reproducción musical del pentagrama. En particular, un formato de editor de música SCORE impone que cada pentagrama sea descrito como una secuencia de eventos, permitiendo una superposición de estas secuencias, representando voces concurrentes. Entonces será necesario introducir nuevas reglas:
 1. Atribuir una misma voz de eventos que pertenezcan a un grupo de notas común.
 2. Atribuir una misma voz de eventos que contengan ligaduras entre sí.

Es importante destacar que el formato SCORE apenas permite ligar y agrupar notas pertenecientes a una misma voz.

- Interpretación de voces: es el proceso por el cual una lista de eventos es leída y se ejecutan las siguientes tareas:
 1. Determinación de la altura de las notas.
 2. Elaboración de armadura de clave.
 3. Tratamiento de las barras de compás.
 4. Creación de campos del fichero en formato SCORE.
- Generación de fichero de salida:
 1. En formato SCORE: es generado a partir de las cadenas caracteres construidas en el paso anterior.
 2. En formato MIDI: este formato es el usado en la mayoría de las aplicaciones, no obstante algunas consideraciones son hechas.
 - Pérdida de información.

- Ausencia de indicación de compás.
- Ausencia de edición previa de las partituras reconocidas.

4.4.3 Lee Sau Dan

La salida de datos es en ASCII, en un formato de diseño propio, que permite su conversión a otro formato fácilmente.

5 EL PROCESAMIENTO EN EL RECONOCIMIENTO DE MÚSICA IMPRESA

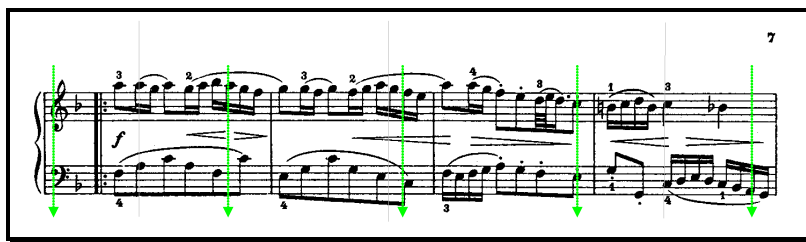
(Tema de realización de la publicación entregada)

En esta primera etapa describiremos los problemas encontrados y las soluciones implementadas a través de distintos métodos.

La información leída, nos permitió implementar métodos con las técnicas que consideramos más eficientes.

5.1 Búsqueda de características del pentagrama

Conociendo a priori las componentes de un pentagrama, en partituras de música impresa, tales como: el espaciado entre líneas y el grosor de las líneas, desarrollamos un método que nos permita hallar sus valores. El método consiste de hacer corridas verticales de la imagen en columnas dispersas proporcionalmente al ancho de la imagen. Las corridas se realizan en cinco columnas para obtener un resultado más exacto, ya que durante el proceso de digitalización de la imagen, el pentagrama pierde la precisión en el patrón de líneas equiespaciadas y el grosor constante.

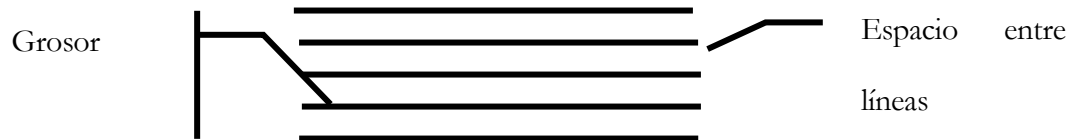


La imagen de una partitura nunca va a constar de un pentagrama vacío, es decir, sin símbolos musicales. El método que obtiene las componentes del pentagrama contempla la presencia de los símbolos superpuestos y/o adyacentes a las líneas del pentagrama.

Como dijimos anteriormente consta de cinco corridas verticales. En cada una de ellas vamos buscando grupos de píxeles alternadamente blancos y negros en forma secuencial. A estos grupos

de píxeles los denominaremos segmentos. Por cada segmento se guarda su longitud, la posición de comienzo y si el segmento es de blancos o negros. Luego obtenemos:

- Espaciado entre líneas o interlineado: La cantidad de píxeles blancos que hay entre 2 líneas del pentagrama. Obtenemos el espaciado entre líneas como promedio de los segmentos blancos más frecuentes en cada una de las columnas mencionadas anteriormente.



- Grosor de las líneas: Similarmente se designa el grosor como el promedio de las longitudes de los segmentos negros más frecuentes en cada una de las cinco columnas en las que se realizan las corridas. Para agilizar este proceso durante la búsqueda de los segmentos negros más frecuentes, se establece una condición que restringe la cantidad de segmentos negros. La condición es que aquellas longitudes que sean superiores a la tercera parte del interlineado, que ya conocemos, se descartan, ya que formarían parte de otros objetos².

5.2 Localización de los pentagramas

Utilizamos una modificación realizada a la técnica utilizada por Xulio Fernandez, que consiste en no trabajar sobre la imagen píxel a píxel sino sobre una tabla compuesta de: segmentos blancos y negros, y la posición y la longitud de ambos segmentos. Dicha tabla almacena los datos obtenidos durante el proceso de cálculo de espaciado entre líneas y el grosor. Esto agiliza la búsqueda de pentagramas notablemente, ya que el desplazamiento es a nivel segmento y no a nivel píxel.

Para encontrar todos los pentagramas de las partituras con una mayor precisión, buscamos en 3 columnas distintas. Una vez encontradas las posiciones iniciales de cada pentagrama se almacenan en una estructura de datos. Musicalmente hablando se guardarán las coordenadas del comienzo de la quinta línea de cada pentagrama.

En cada una de las columnas nos posicionamos en el primer segmento negro, luego buscamos 4 segmentos negros consecutivos que se encuentren distanciados por un espacio interlineal

² Cabe destacar que no se tendrán en cuenta en este trabajo las partituras que contienen lírica.

aproximadamente. Para localizar cada uno de los 4 segmentos negros lo que se hace es sumar a la coordenada Y, el total del interlineado más el grosor de línea.

Si no es posible encontrar los restantes cuatro segmentos equidistantes consecutivos seguimos probando con el siguiente segmento negro.

Luego de hallados los cinco segmentos consecutivos, nos fijamos que en los cuatro interlineados siguientes no se encuentren segmentos negros equidistantes. Si se cumple esta condición es determinado como pentagrama.

En el caso de no poder determinar el pentagrama por algún factor que causa una indeterminación como por ejemplo una línea adicional por encima o por debajo del pentagrama, una ligadura, una barra de un grupo de notas, etc (Como vemos en la figura). Entonces se realiza el método utilizado por Xulio Fernandez ubicándonos en la imagen según la posición que obtenemos de la tabla..



Una vez encontrados todos los pentagramas en las tres columnas nos quedamos con el número máximo de pentagramas encontrados en una columna.

5.3 Detección de la inclinación de la imagen

Al haber obtenido nuestras imágenes desde la digitalización con un scanner es posible que no se haya colocado la hoja lo suficientemente derecha. Esto provoca complicaciones en la localización y reconocimiento de los símbolos musicales, por esta razón se averiguará el ángulo de giro que tiene la imagen fuente.

Limitamos este giro a +/- 5 grados, lo que es apreciable por el usuario.



Si se da el caso que la inclinación es mayor, se pedirá que se repita el proceso de digitalización intentando así ajustar mejor la imagen.

Lo que hacemos es buscar en la parte izquierda y derecha de la imagen, el primer pentagrama que aparece. Una vez ubicado el pentagrama como se describió anteriormente en el punto 5.2, calculamos la pendiente entre los dos puntos iniciales del pentagrama y nos dará como resultado el número de grados en que se halla girada la imagen.

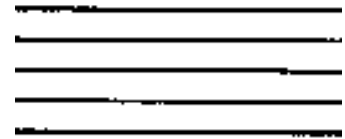
Cabe destacar que se realiza aplicando el Algoritmo de Bresenham, el cual realiza la proporción entre píxeles en horizontal y vertical como referencia del giro (Ver Técnicas). Se basa en ir calculando cada cuantos píxeles en horizontal se debe bajar o subir en vertical.

5.4 Eliminación del pentagrama

Una vez ubicados los pentagramas, de manera de procesar la imagen en cuantos al reconocimiento de los símbolos musicales, eliminaremos las pentagrama, ya que conocemos su posición por lo que no nos aporta ninguna otra información.

Implementamos un método que logra que la imagen quede más limpia y manejable que la original y prácticamente sin pérdida de información.

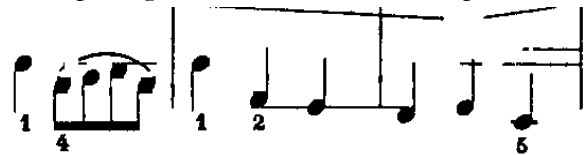
La digitalización convierte esta tarea en algo bastante complicado, ya que las líneas no van a tener un grosor uniforme y además algunas de ellas atraviesan figuras o símbolos musicales que no debemos modificar borrando píxeles que pertenezcan a ellos.



Comúnmente se detectan dos problemas:

- La eliminación incompleta de las líneas. (pérdida de la dirección de seguimiento)
- Eliminación incompleta del grosor de las líneas.

Aplicamos un algoritmo que siguiera una línea del pentagrama. Tomamos cuatro puntos de cada una de las líneas superiores de los pentagramas encontrados, siempre teniendo en cuenta la inclinación de la imagen que ya es conocida. La división nos permite trabajar con tramos cortos esto implica que la desviación es menor y en caso de existir algún error éste también será de menor proporción. De este modo evitamos la eliminación incompleta de líneas.



Para cada tramo recorrido medimos el grosor en píxeles (en dirección vertical), a veces nos encontraremos con que el grosor medido es bastante mas grande que el grosor (GRO) calculado anteriormente., lo cual implica que estamos en presencia de una figura.

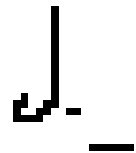
La idea en principio es eliminar aquellos tramos verticales que midan lo mismo que el grosor (GRO), pero si aplicamos esto tan estrictamente nos pueden aparecer errores como se muestra en la Figura.



Como se ven en la figura quedaron restos de líneas sin borrar, este error se debe a que existen valores de grosor de pentagrama mayores que el valor de la media. Por lo tanto para solucionar este problema duplicamos el valor de threshold para la eliminación de las líneas.

Un problema que surge después de borrar las líneas del pentagrama, es la pérdida de partes de símbolos, en aquellos que son tangentes a las líneas como vemos en la Figura.

Este problema se soluciona posteriormente en la etapa de reconocimiento de símbolos debido a la metodología de trabajo donde la copia original de la imagen puede consultarse, si es necesario.



8

Rondo.
Poco Allegretto, quasi Andantino.

(Fr. Couperin)

espr.

7. *p* *mf*

espr. *p* *cresc.*

p *cresc.*

f *p*

Fine.

cresc. *f* *p*

cresc. *f* *espr.* *poco f* *p*

Edition Peters. 10890

Imagen Original

8

Rondo.
Poco Allegretto, quasi Andantino. (Fr. Couperin)

7.

p *mf* *espr.*

espr. *p* *cresc.*

p *cresc.*

f *p*

cresc. *f* *poco f*

p

Fine.

Edition Peters. 10890

Imagen luego de eliminar los pentagramas

5.5 Definición de sistemas

Nos interesa analizar lo que hemos dado en llamar definición de sistemas. Consiste en encontrar si están o no agrupados los pentagramas en la partitura. Los pentagramas están unidos por una barra divisoria en los márgenes izquierdo y derecho y delimitando los compases como veremos más adelante en 5.6, además el espaciado entre los pentagramas de un sistema es menor que el que los separa del siguiente sistema de la partitura.

Definimos a n conjunto de pentagramas agrupados como un sistema. Los sistemas tienen la característica de que todos los eventos musicales escritos sobre los pentagramas que lo componen se interpretan simultáneamente.

Generalmente, salvo excepciones, cada uno de los pentagramas del sistema corresponden a

The image shows a snippet of a musical score titled "Septième Concert Royale". It features three staves. The top staff is labeled "Traverso / Violin" and contains a melodic line with various notes and rests. The middle staff is labeled "Basse Continue" and contains a bass line with notes and rests. The bottom staff is a continuation of the melodic line. A red vertical line is drawn between the first and second staves, indicating a system boundary. The score includes musical notation such as clefs, notes, rests, and dynamic markings like "F".

instrumentos musicales o a voces diferentes, pero todos deberán ejecutar la partitura a la vez. Excepciones como en el caso de las partituras de piano, los sistemas son de dos pentagramas, esto indica que un pentagrama se ejecuta con la mano

izquierda y el otro con la derecha.

Para averiguar el tipo de sistema que existe en la partitura nos basamos en el espaciado entre pentagramas.

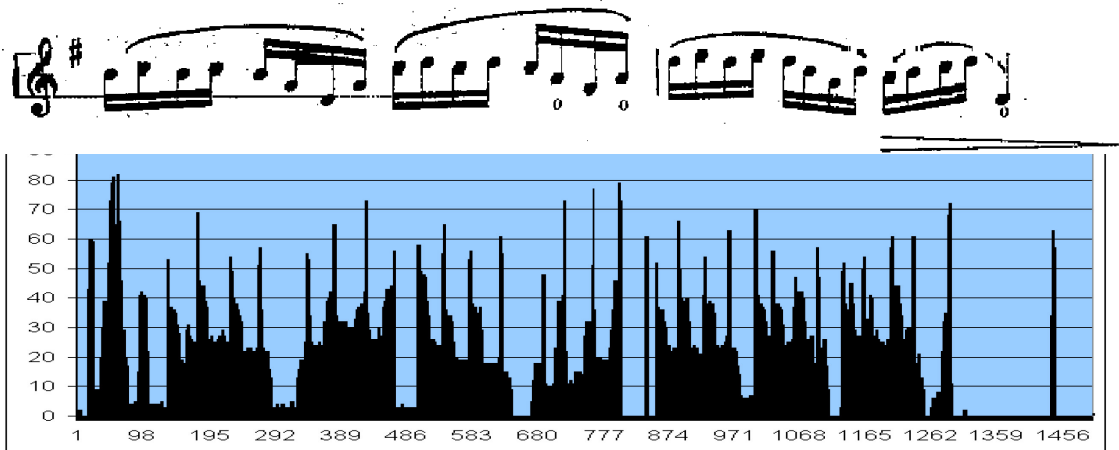
Tomamos como base el espaciado entre los dos primeros pentagramas y a continuación los comparamos con el tercer pentagrama. Si el tamaño de los espaciados es similar (teniendo en cuenta un margen de diferencia) decimos que ese pentagrama pertenece al sistema anterior. Y en caso que los espaciados sean muy diferentes, suponemos que comienza un nuevo sistema.

5.6 Delimitación de compases

Las barras verticales que cruzan el sistema se llaman barras divisorias, tiene la función de delimitar espacios temporales iguales llamados compases.

Dividimos en dos casos el tratamiento de las barras divisorias:

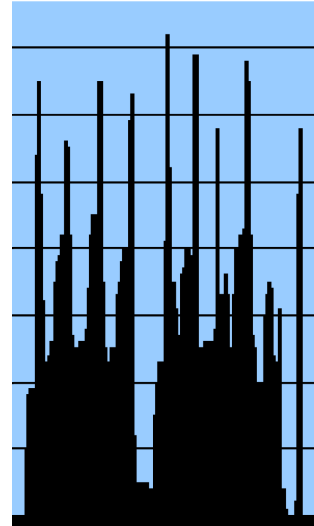
- a) Pentagrama único
 - b) Sistemas
- a) Para detectar las barras divisorias en pentagramas únicos el proceso es complejo.



Si intentáramos localizar los pentagramas por medio de proyecciones nos encontraríamos con que, al no destacar el tamaño de las barras frente al resto de los símbolos, no obtendríamos ninguna información:

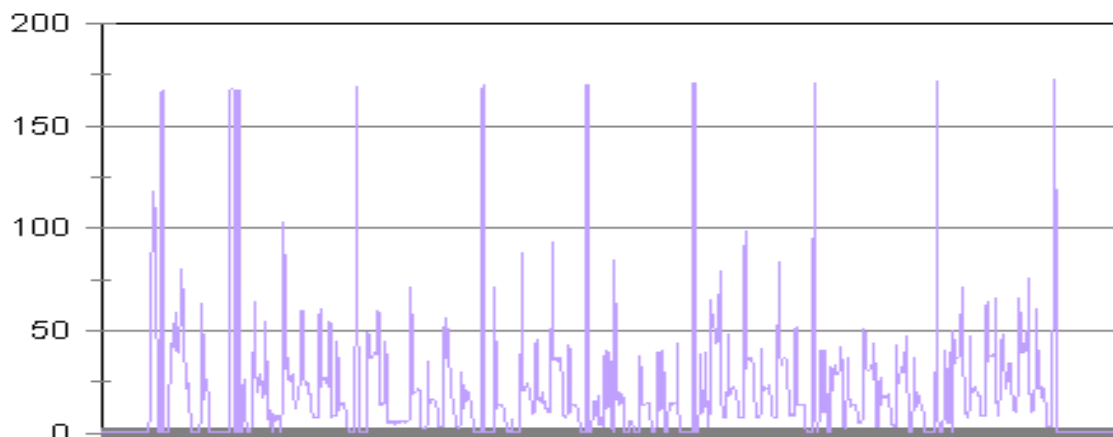
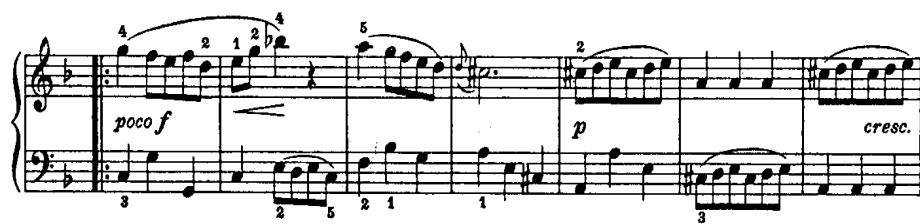
Esto nos ha ocurrido porque, principalmente, las plicas de las notas son del mismo orden de magnitud o incluso mayores que las barras divisorias, por lo que en la proyección son indistinguibles. En la figura siguiente aparece un ejemplo.

Entonces trataremos de averiguar cuales son las barras divisorias de entre todos los picos que superen un cierto threshold inferior (por ejemplo el 95% de $4*EIL$), pero que no superen un segundo threshold (superior) dado que esto significaría que la barra es más alta de lo posible.



De este modo hemos resuelto parcialmente el problema. Al menos ahora no confundimos plicas largas con barras divisorias, pero seguimos teniendo problemas con las demás plicas. Luego calculamos el grosor de las proyecciones cuyos máximos están dentro de unos ciertos márgenes, estudiamos la caída por la izquierda y derecha de esos picos para ver “cuanto tarda” en bajar de un cierto umbral. Caso de cumplirse esta serie de condiciones, se decide que es una barra divisoria.

- b) Para detectar las barras divisorias en un sistema realizamos una proyección horizontal de la imagen en la cual las líneas pentagrama fueron eliminadas. En esta proyección se distinguen claramente las barras divisorias como se muestra en la figura:



Localizamos el máximo de los valores proyectados, definimos como threshold al 80% de este máximo. Decimos entonces que todos los valores que superen este threshold son las posiciones de las barras divisorias.

5.7 Doble barras

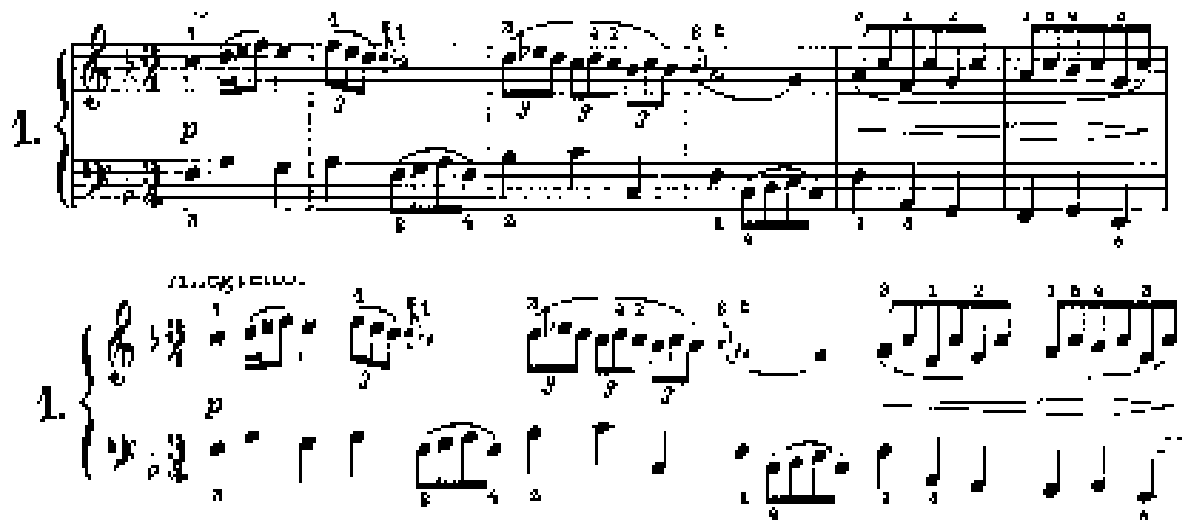
Medimos en cada sistema o pentagrama la distancia en horizontal entre cada par de barras. De esta manera encontramos las barras divisorias dobles que no son más que barras divisorias simple muy cercanas. Luego son analizadas en una zona pequeña para determinar si son signo de repetición o cambio de clave, etc.



5.8 Eliminaciones de barras divisorias

La eliminación de las barras divisorias se realiza exactamente igual que la eliminación de las líneas del pentagrama, excepto que son líneas verticales más cortas.

Estas barras las eliminaremos para seguir limpiando la imagen.



En la figura vemos el aspecto de la imagen antes y después de este proceso.

En el caso que haya inclinación hay que corregir el valor de la fila inicial, a la real, utilizando el conocimiento de la inclinación ya mencionado anteriormente.

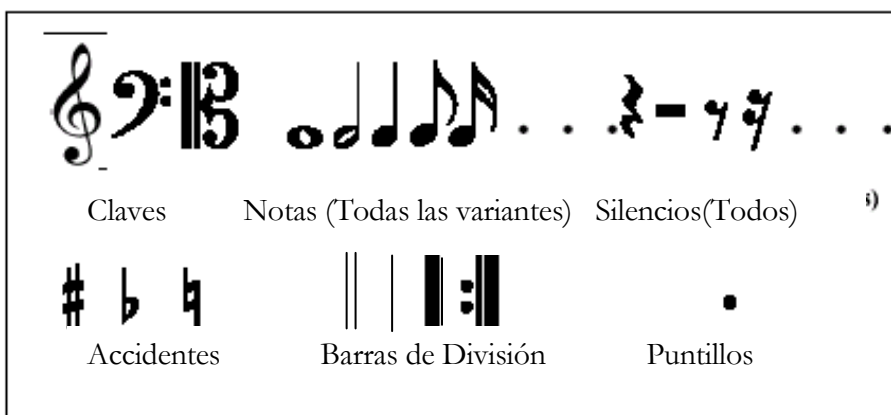
Continuando con el proceso de eliminación de las barras divisorias, lo que se hace posteriormente es recorrer la barra verticalmente hacia arriba y hacia abajo eliminando el tramo horizontal en cada uno de los puntos de la línea. Hay que decir que el recorrido en ambas direcciones de esta línea “gruesa” es bastante básico, limitándose a ver si el píxel de la línea siguiente³, y de la misma columna C , está “a negro” o no, (o en su defecto, cualquiera de sus vecinos de la línea siguiente (en las columnas $C-1$ y $C+1$)). Así se repite el procedimiento hasta no poder continuar. El método así programado funciona correctamente dado que las inclinaciones permitidas en la imagen son leves, por lo que no afectan al seguimiento de estas líneas verticales.

³ Hacia arriba o hacia abajo.

6 SEGMENTACIÓN Y RECONOCIMIENTO DE SÍMBOLOS

6.1 Símbolos reconocidos por la aplicación

Los símbolos y sus características que son reconocidos en esta aplicación:



Localizaremos las 3 claves en sus 7 posiciones, todas las notas con la plica adelante o atrás, hacia arriba o hacia abajo, todos los silencios, los accidentes expuestos en forma individual o como componentes de una tonalidad, las barras de división y los puntillos.

No se localizaran, las ligaduras y los calderones.

6.2 Metodología de trabajo ‘

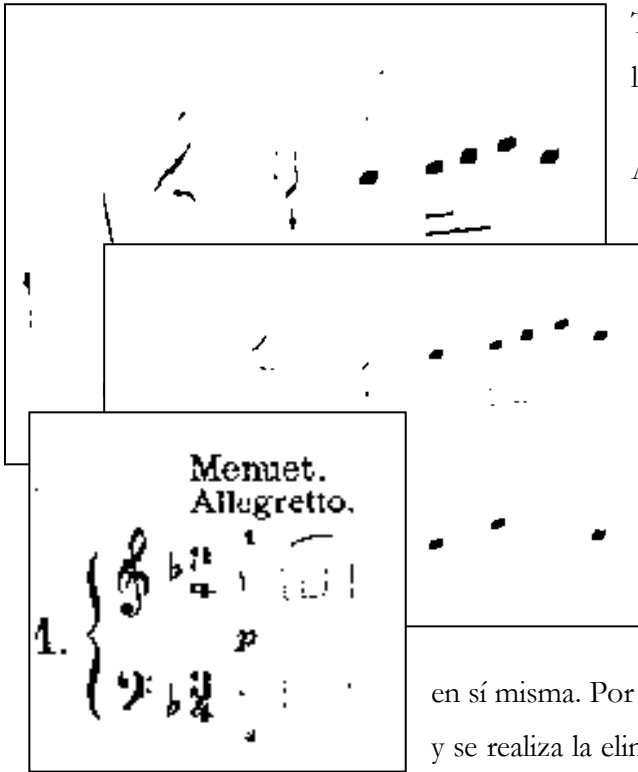
Trabajo en Capas

Elegimos el método de trabajo en capas. Consiste en mantener en memoria diferentes imágenes que se procesan convenientemente, además de la original. La elección estuvo



basada en varias de sus ventajas como que aunque durante el procesamiento de la imagen

malogremos parte ella, siempre podemos volver a consultar la original. Como por ejemplo, durante la eliminación se pierde parte de los símbolos tangentes a las líneas del pentagrama, lo que nos resulta un problema en la etapa de reconocer el símbolo, como solución simplemente será necesario consultar la imagen original.



Trabajaremos sobre pequeñas zonas o ventanas de la imagen de manera de aprovechar la memoria.

Asumimos que cada capa que se procesa, en las que usamos métodos como la erosión o el thinning, podremos consultar sus características de zona o ventana de la imagen con facilidad, ya que sólo habrá que consultar el mismo punto en otra capa de la imagen en las diferentes capas.

Como parte de la metodología de trabajo incluimos poder almacenar el resultado el resultado de procesar una imagen en otra capa o en sí misma. Por ejemplo localizamos las cabezas negras en una capa y se realiza la eliminación de las cabezas en la capa que se encuentra sin pentagrama y sin barras de división.

En nuestra aplicación trabajaremos los compases como zonas o ventanas. Esto significa que tomaremos un compás y trabajaremos con él lo cual implica que la ventana será mucho más pequeña que la imagen original. Recorremos todos los compases de la partitura. Trabajaremos de arriba hacia abajo, comenzando por el primer compás del primer pentagrama localizado, hasta llegar al último compás del último pentagrama.

A partir de este punto tomaremos la imagen que ya tiene eliminados los pentagramas y las barras divisorias. Ya tenemos en memoria la localización de los pentagramas y de las barras divisorias, con lo que tenemos perfectamente delimitado el espacio de búsqueda de los símbolos.

La primera tarea que realiza este código es erosionar convenientemente la ventana en la que está contenido el compás (erosión de profundidad $2 * GRO$, discutida en el apartado correspondiente). Comentar nada más cómo se usan las copias de trabajo manejadas por $g1$ y $g2$ para, tras erosionar una zona de $g1$, dejar el resultado en la misma zona de $g2$ y, posteriormente, copiar esa zona

erosionada de g^2 a g^1 para repetir el proceso. No es posible realizar esta tarea almacenando los resultados de la erosión en la copia de trabajo original, sin alterar el resultado.

Posteriormente ya entramos en el control del proceso de reconocimiento. Es fundamental entender el significado de la variables *com*. Cada vez que llamemos a una función que intente localizar y reconocer algún tipo de figura (cabezas negras, blancas, silencios, etc...), esta nos devolverá un valor que almacenaremos en la variable *com*. El resultado puede ser:



- La suma de los tiempos que ocupan los símbolos localizados del compás (todos), ocupan exactamente lo esperado.
- La suma mencionada es menor que lo esperado. Deberíamos seguir buscando símbolos.
- La suma es mayor de lo esperado. Esto puede implicar algún error durante el reconocimiento o matices en algunos de ellos (por ejemplo grupos de valoración especial: tresillos, dosillos, etc...)

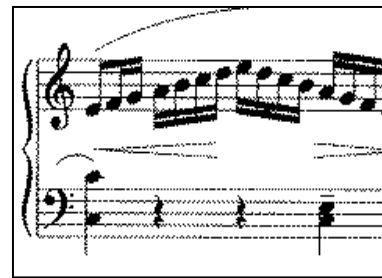
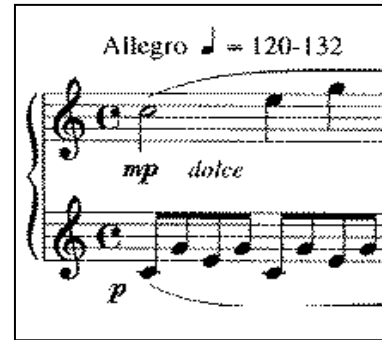
Es decir, con esta variable controlaremos el reconocimiento a nivel de compás. Los pasos que se dan dentro de la función son los siguientes:

1. Buscar las cabezas negras (y luego sus plicas y corchetes para conocer sus duraciones) y sumar los tiempos de los símbolos encontrados. Devolver el valor del *com* actual.
2. Si no es 0, localizar las blancas del compás (en una copia de trabajo no erosionada) y añadir la suma de tiempos de las blancas encontradas al *com* actual, devolviendo el estado de esta suma.
3. De nuevo repetir el procedimiento pero buscando esta vez los silencios.
4. Lo mismo para las redondas.

6.3 Casos de excepción

Destacaremos dos situaciones en la que los compases se procesan de manera diferente:

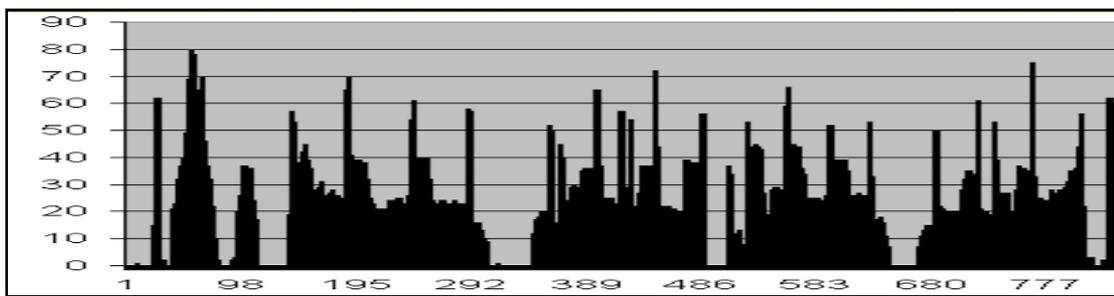
- Primer compás de la partitura: aquí aparecerán la clave, la tonalidad, el compás y posiblemente el Aire o Tempo. Tendremos que localizar y reconocer la clave y la tonalidad. Se pedirá al usuario que ingrese el compás mediante el teclado. En el caso que hubiere Aire o Tempo no lo consideraremos ya que no trabajamos con O.C.R.
- Primer compás de los pentagramas localizados: aparecerá la clave y la tonalidad (si tiene) como vemos en la figura que sigue. Entonces reconoceremos y localizaremos ambos símbolos.



6.4 Reconocimiento de las claves

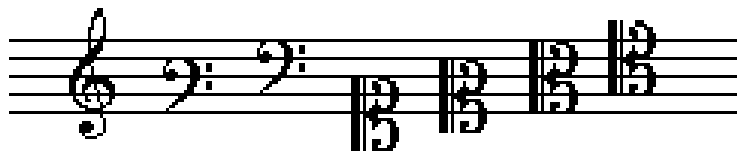


Lo primero que hacemos para recorrer una clave es una proyección del compás completo, después buscamos el primer tramo de la proyección que supere dos veces el grosor (para no confundir con una barra divisoria), y además debe superar una anchura que sea 3 veces el espaciado; de esta manera acotamos la clave horizontalmente.



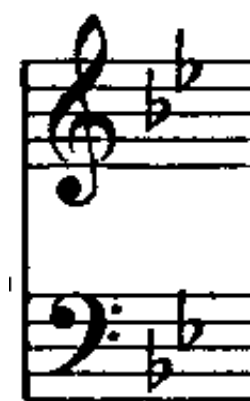
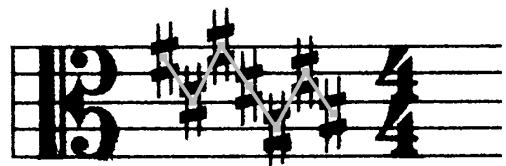
Para acotarla verticalmente lo que hacemos es partir del centro de donde estaría el pentagrama, buscamos dentro del ancho encontrando líneas vacías por encima y por debajo de ese centro.

El reconocimiento se realiza por comparación de tamaños (altura). Por ejemplo la Clave de Sol tiene una altura aproximada de 8*EIL, mientras que la Clave de Fa 3.5 *EIL y las Claves de Do unos 4.5 *EIL . Además como se verá posteriormente, al localizar y reconocer la tonalidad (conjunto de alteraciones que sigue a la clave), podemos usarla a modo de comprobación para ver si es sintácticamente correcta la correspondencia entre clave y tonalidad. Es tan importante la sintaxis como el estricto reconocimiento de las formas.



6.5 Reconocimiento de tonalidades

La tonalidad es el grupo de alteraciones que aparece después de la clave. Sabemos por definición que los sostenidos siguen el orden Fa-Do-Sol-Re-La-Mi-Si y los bemoles Si-Mi-La-Re-Sol-Do-Fa.



En nuestra aplicación almacenamos los 14 valores para las 7 claves con el objeto de evitar la búsqueda ciega de las alteraciones de la tonalidad, ya que sabemos por donde comenzar y por donde seguir la búsqueda.

En las figuras se muestran algunas de las secuencias de las tonalidades más usadas.

The image displays a musical score with eight rows of staves, each containing four staves. The first two columns use treble clefs, and the last two columns use bass clefs. The key signature changes from C major to D major, then to D minor, and finally to E-flat major across the rows.

Row	Staff 1 (Treble)	Staff 2 (Treble)	Staff 3 (Bass)	Staff 4 (Bass)
1	C Major	C Major	C Major	C Major
2	D Major	D Major	D Major	D Major
3	D Major	D Minor	D Major	D Minor
4	D Major	D Minor	D Major	D Minor
5	D Major	D Minor	D Major	D Minor
6	D Major	D Minor	D Major	D Minor
7	D Major	D Minor	D Major	D Minor
8	D Major	D Minor	D Major	D Minor

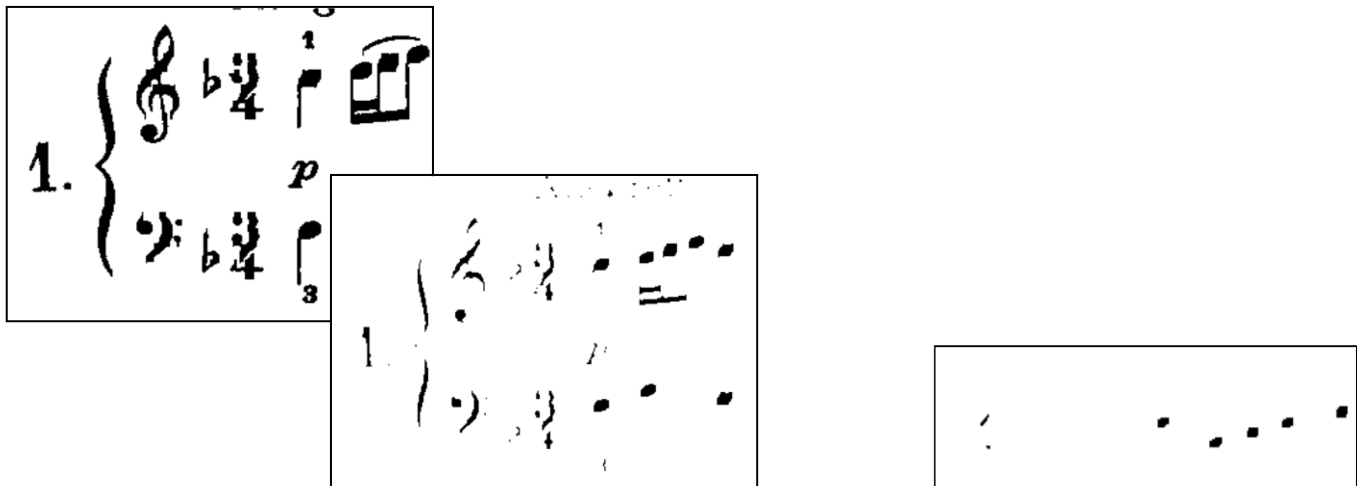
6.6 Localización y reconocimiento de notas con cabeza negra

Observando una partitura podemos ver que una de las características comunes es que casi todos los símbolos musicales tienen la cabeza negra.

Una vez que hayamos localizado todas las cabezas negras junto con sus símbolos asociados (sostenido #, bemol y puntillo .) habremos reconocido gran parte de la partitura.

6.6.1 Localización de notas con cabeza negra

Para localizar las notas con cabeza negra aplicamos erosión, porque es un método morfológico, rápido y eficiente. Primero localizamos el centro de las cabezas negras realizando una erosión a cierta profundidad en la zona de la imagen que tratamos.



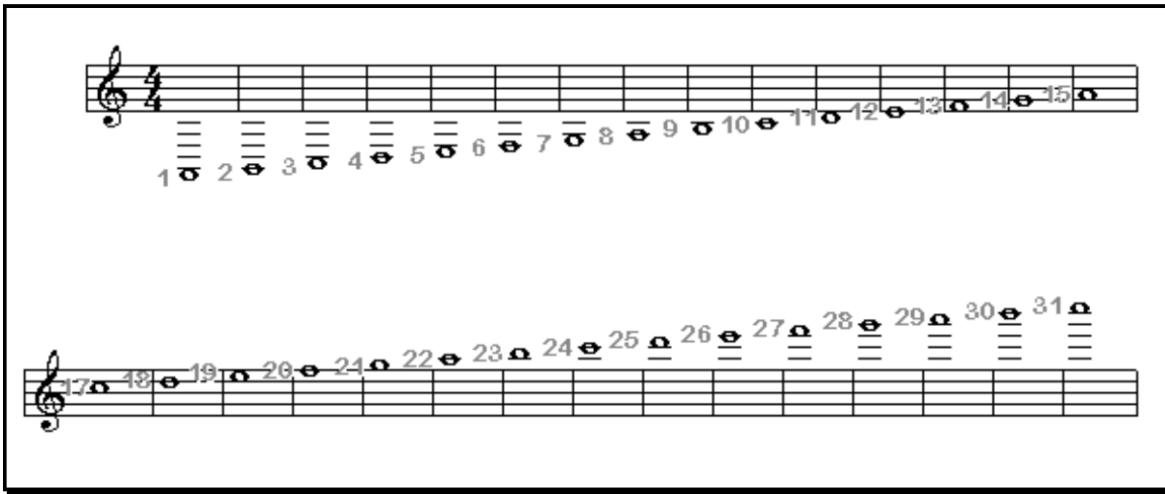
Lo que hacemos a continuación es buscar las dimensiones del rectángulo donde se encuentran las cabezas negras. Controlamos que cumplan ciertos thresholds mínimos y máximos en forma horizontal y vertical. En caso que se reconozca como cabeza negra entonces se guarda el centro del rectángulo como el centro de la cabeza negra.

Una vez que todas las cabezas negras fueron encontradas y guardada su posición, éstas son borradas evitando así procesar el resto de las cabezas ahorrando tiempo.

Para el proceso de erosión elegimos como profundidad dos veces el grosor, ésta elección se basa en que con esa profundidad se deja solamente los núcleos de las cabezas negras al erosionar. Podríamos haber elegido una profundidad menor pero al haber más conjuntos de puntos conexos se tardaría más en procesar.

6.6.2 Reconocimiento de notas de cabeza negra

Una vez localizadas las cabezas negras y antes de ocuparnos del resto de las figuras lo que hacemos es calcular la posición relativa de las cabezas negras en el pentagrama al que pertenecen. Para ello usamos lo que conocemos que son las coordenadas (x,y) en la imagen. Averiguaremos entonces su posición relativa en la partitura. Mediante una función se devuelve el número de la nota dentro del pentagrama, la numeración es la asignada en la figura a continuación. Es importante destacar que esta numeración no es la de MIDI.



Nuestra función realiza dos operaciones:

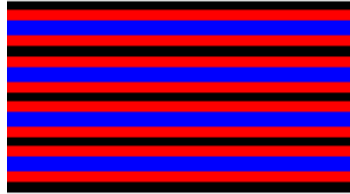
- La primera es encontrar la coordenada vertical del punto respecto al pentagrama. Conocemos la columna (colA) y el primer punto (pA) de esa columna en el pentagrama, los cuales se almacenaron al detectas las líneas del pentagrama.

Como tenemos que averiguar en qué posición (A') del pentagrama está la nota debemos, en caso de haber inclinación, hacer la corrección de pA al A' , utilizando los valores de la inclinación conocidos.

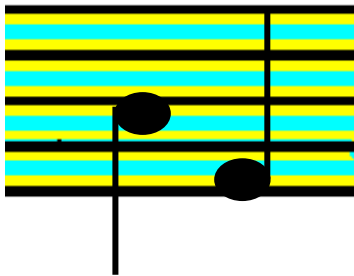
Entonces ya sabemos en qué coordenadas comienza el pentagrama verticalmente cuanto dista este punto al centro de pA, queda por averiguar en qué posición relativa está, es decir qué numeración le corresponde⁴.

⁴ Tenemos una tabla donde a cada nota en el pentagrama se le asigna un número

- b) La segunda operación consiste en averiguar a cual de las líneas o espacios del pentagrama va asociado ese centro de la nota entonces lo que hacemos es dividir cada espacio en cuatro zonas de igual anchura.



Si el centro de la cabeza que localizamos en el punto a) se encuentra dentro de los dos tramos centrales, asignamos la nota a ese espacio.



Si está dentro de los dos tramos más cercanos a una línea, asignamos la posición a esa línea.

Todo este procedimiento nos servirá para averiguar la posición de cualquier otra figura que procesemos.

A este punto sabemos el nombre de las notas. Ahora localizaremos las plicas.

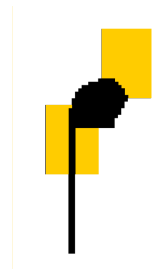
6.7 Localización y Reconocimiento de plicas

Lo primero que hacemos es localizar la plica que acompaña a la cabeza negra y seguidamente localizar los corchetes que salen de ella, que indican la duración de la nota.

Vamos a trabajar con la copia en la cual se eliminaron las líneas de pentagrama, las barras divisorias y las cabezas negras de las notas.

Sabemos que existen reglas musicales que indican a dónde y cómo colocar las plicas en las notas:

- Si la nota está por debajo de la tercera línea, la plica se dibujará hacia arriba y la cabeza estará por delante de la plica
- Si la nota está justo en la tercera línea la plica puede ir para arriba o para abajo.
- Si la nota esta por encima de la tercera línea, la plica va hacia abajo con la cabeza por detrás de la plica.



Estas reglas no son validas cuando se trata de grupos de notas. Hicimos una función que estudia dos zonas que están en el posible comienzo de la plica e intenta recorrer, si es posible, un tramo vertical.

Como resultado de la función obtendremos la dirección de la plica.

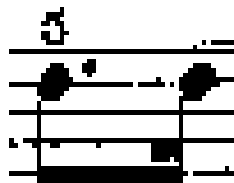
Paso siguiente localizaremos el otro extremo.

6.7.1 Localización de corchetes y barras de grupo

Nos falta por determinar la duración de las notas que depende del número de corchetes o salientes que aparecen en el extremo de la plica de la nota.

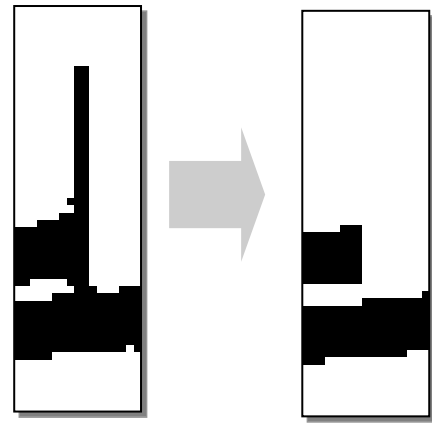


Estos corchetes pueden aparecer unidos a las notas o a un grupo de notas unidos por lo que llamamos barra de grupo.



El primer paso que se realiza, es el de tomar una ventana rodeando a la plica bajo estudio (recordemos que ya conocemos sus extremos y que hemos eliminado la cabeza negra). Por ejemplo, si extraemos esa pequeña ventana de la figura central de la última imagen, vemos que por el extremo inferior izquierdo de la plica aparecen dos corchetes mientras que por la derecha aparece uno. Para realizar el conteo automático de esos corchetes o salientes, lo primero que haremos será eliminar la propia plica de esa ventana. Esto se realiza mediante una erosión únicamente en la dirección horizontal, de profundidad $3 \cdot \text{GRO} / 2$ (1.5 veces el grosor estimado para el trazado básico de la partitura). Aplicando esto a la pequeña ventana segmentada, obtenemos una serie de bandas prácticamente horizontales, de una determinada longitud. Así, por ejemplo, en la figura de la izquierda, se obtienen

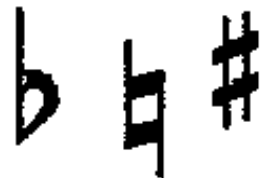
dos bandas. Lo que se hace a continuación es contar para cada columna de esa ventana, el máximo número de transiciones verticales blanco/negro y negro/blanco, que cumplan una serie de reglas mínimas de grosor. Ese número máximo es el número de corchetes que se le asigna a la cabeza negra que estoy tratando (en el ejemplo de esta figura, ese número máximo de transiciones o bandas es 2, que se dan en las primeras columnas). Así, atendiendo al número de transiciones las figuras serán las siguientes:



<i>0 transiciones</i>	<i>Negra</i>
<i>1 transición</i>	<i>Corchea</i>
<i>2</i>	<i>Semicorchea</i>
<i>3</i>	<i>Fusa</i>
<i>4</i>	<i>Semifusa</i>

6.8 Localización y Reconocimiento de alteraciones

Las alteraciones modifican el tono de la nota que se ha localizado. Trataremos las alteraciones accidentales (sostenido, bemol y becuadro) que son las que encontramos en cualquier lugar de la partitura antecediendo a una nota, no las que forman parte de la tonalidad.



Los símbolos que buscamos están asociados a notas de cabeza negra pero esos procedimientos se usarán de igual manera con las redondas y blancas aún no localizadas.

Sabemos que la distancia entre la alteración y la nota no es constante.

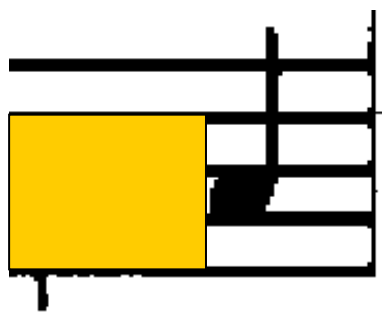
Trabajamos con la copia en la cual no están las cabezas negras, ni las plicas, ni los corchetes o barras de grupos. El proceso que realizamos es el siguiente:

1. Buscamos en una zona que puede ser entre la última nota estudiada y la actual o una zona cuya distancia sea menor que $3 * EIL$, este último caso mencionado se da cuando la nota actual no está precedida por ninguna nota.



2. Para cada zona encontrada se hace una proyección vertical con una altura aproximada a $3 * EIL$.

En este momento tenemos acotada la posible alteración ente dos columnas. Como conocemos la posición de entre las 31 posibles, de la cabeza o nota que estamos tratando, sabemos dónde debe

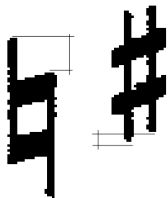
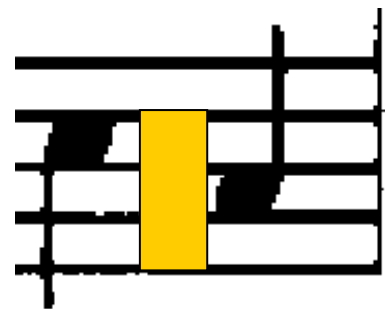


estar localizado el centro de la alteración si es que la hay. Recorriendo hacia arriba y hacia abajo la banda vertical segmentada, a partir de ese centro, localizamos los extremos de la alteración, encontrando así la bounding-box.

a la función de reconocimiento

Una vez segmentada la posible alteración, pasamos esta ventana

Esta función de reconocimiento realmente no decide por la forma de la alteración, como sería de esperar. Se basa en los picos correspondientes a las barras verticales izquierda y derecha (si la hay). Estrictamente hablando no es un reconocimiento de alteraciones, pero en la práctica funciona correctamente.



Realizamos una extracción de los picos de la proyección de la ventana segmentada, comprobando si es uno y está a la izquierda, o si son dos (a izquierda y derecha). Admitimos como “pico” un cierto tamaño en la proyección que tenga

una altura mínima, y que además no sobrepase un *threshold* de anchura⁵. Este es un primer filtrado. Como segundo paso, si sólo existe un pico a la izquierda, puede ser un bemol. Para comprobarlo se observa el resto de la proyección⁶. El problema más delicado sería el de discernir entre el sostenido y el becuadro. Como sus proyecciones son prácticamente idénticas en la dirección vertical y muy parecidas en dirección horizontal, lo que se hace es medir las diferencias de altura entre los picos de las barras izquierda y derecha (tanto en la zona superior como en la inferior: ver *Figura* . Es decir, se toma la mitad superior y medimos la diferencia de alturas entre los extremos izquierdo y derecho. De igual modo se hace en la zona inferior. Tomamos la mayor de esas diferencias y comparamos con un cierto *threshold* que decide entre sostenido y becuadro (fijado en $\frac{1}{4}$ de EIL).

6.9 Localización y Reconocimiento de puntillos

Los puntillos son los puntos que están colocados a la derecha de la nota o silencio y aumenta en un medio su duración.

Localizaremos los puntillos asociados a cabezas negras, blancas, redondas y silencios. Nos basamos en las reglas básicas de la escritura musical para lograr su ubicación.



Estas reglas son:

- Cuando la figura está sobre un espacio, el puntillo se escribe sobre el mismo espacio a la derecha.
- Cuando la figura está sobre una línea, el puntillo se escribe indistintamente sobre el espacio superior inferior y también a la derecha.

Por lo tanto nuestra función buscará el puntillo siempre entre dos líneas. La función que lo localiza recibirá las coordenadas del centro de la cabeza y la posición de la misma entre las 31 posibles. Si el número de la posición es impar significa que



⁵ Estos *threshold* fueron decididos en las pruebas.

⁶ Es importante darse cuenta de que lo que se le pasa a la función de reconocimiento puede no ser exactamente el bemol (con forma de "b"), sino una versión sesgada por las tangencias. Esto suele ocurrir con los bemoles colocados en los espacios. Con los sostenidos y becuadros puede ocurrir algo parecido. Siempre puede recurrirse a imágenes de procesados anteriores.

la nota está sobre un espacio por lo que buscaremos directamente en ese espacio. Ahora, si fuese para significa que la nota está sobre una línea entonces se buscará primero en el espacio superior y si no se localiza se buscará en el inferior.

Luego recorreremos el espacio hacia la derecha por su punto medio, busco primero un cambio de blanco a negro y luego de negro a blanco. A continuación comprobamos la anchura de la mancha. Si cumple la anchura mínima localizamos el bounding box y entonces controlamos que este dentro del threshold que será que sea menor que $EIL/2$ tanto de alto como de ancho. Si esto se cumple decimos que es un puntillo, guardamos su posición y luego lo eliminamos.



6.10 Localización y Reconocimiento de blancas

Es más difícil que reconocer figuras con cabeza negra. No podemos usar erosión dado que al ser cabezas “huecas” se borrará completamente el contorno.

Lo que hacemos es tener en cuenta el compás, que es la metodología de nuestro trabajo, entonces sumamos las duraciones de los símbolos localizados y de esta manera sabemos si tenemos que continuar buscando o no.



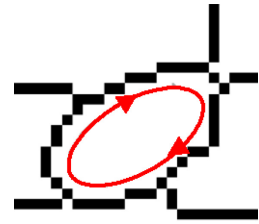
Trabajamos con la copia en la que se ha eliminado todo lo visto hasta este punto. La característica que usaremos para localizar y es conocer las notas blancas es la plica, ya que son los únicos tramos rectos y verticales que nos quedan.

Entonces generamos una proyección horizontal del compás, separando los que sobrepasen $3 \cdot EIL$.

Para diferenciar el pico correspondiente a una plica se estudian todos los picos que cumplen la condición formando un bounding box con la altura es la del threshold y la anchura $2.5 \cdot GRO$ dejando el pico centrado en la ventana.

Hacemos thinging de la ventana, si detectamos que hay una línea continua que es una posible plica cuya longitud es el ramo recorrido (L), siempre que L supere el threshold. A continuación vemos si en el extremo de esta posible plica se encuentra una cabeza blanca, debemos buscar en el punto superior a la derecha y del punto inferior a la izquierda.

Tomaremos una pequeña ventana en cada uno de esos dos extremos, y la función de recorrido del contorno, devuelve si ha podido recorrer un contorno cerrado alrededor del centro de la ventana que se le pasó. Seguidamente comprobaremos las dimensiones y decidiremos ya si es una cabeza blanca o no lo es.



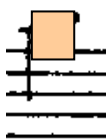
6.10.1 Problemas y resolución de los mismos

Existen dos problemas que aparecen luego del reconocimiento de figuras blancas al utilizar los métodos de thinning y contorno:



- El primer problema aparece cuando aplicamos thinning a cabezas cuyo interior no está totalmente vacío, como es el caso de cabezas superpuestas a líneas adicionales o propias del pentagrama.

La idea es borrar la línea superpuesta por la figura. Para ello averiguamos si la posición del centro de la cabeza blanca es una posición correspondiente a una línea de pentagrama o a una línea adicional. Consiste en enviarle a una cierta función las coordenadas del centro del bounding box que contiene la cabeza, esta función nos devolverá la numeración de la posición ya corregida por la inclinación. Si dicha numeración es una posición en la que sabemos existe una línea cruzando la figura, realizaremos un corte en la línea interior de la cabeza eliminando un píxel en la columna central. Posteriormente se eliminarán de la ventana todos aquellos píxeles tales que sólo tengan uno o ninguno de sus ocho vecinos negros, este proceso termina cuando no se ha podido eliminar ningún píxel. Llegado este punto podemos usar la función del recorrido de contorno de cabeza.



- El segundo problema es localizar con exactitud el final de la plica, dado que puede darse el caso en que no delimitemos correctamente la cabeza dentro del bounding box como se ve en la figura.

La solución a este problema es tomar una ventana de forma rectangular de forma que acotemos horizontalmente toda la posible cabeza. Realizamos una proyección vertical de la ventana y calculamos centro vertical de la cabeza y la coordenada vertical será la que corresponda al ancho de la ventana.

6.11 Localización y reconocimiento de silencios

La localización de los silencios sigue la misma metodología de trabajo que llevamos hasta ahora, es decir, comprobamos la suma de las duraciones de los símbolos encontrados hasta el momento en el compás. Sólo se buscarán los silencios en aquellos tramos donde haga falta completar el compás.

Para localizar los silencios nos basaremos en las reglas básicas de la escritura musical:

- Los silencios de redonda y blanca deben estar colocados en el tercer espacio del pentagrama. Lo que haremos es recorrer dos líneas dentro del tercer espacio hasta encontrar algún grupo de píxeles negros de dimensión determinada. Estas líneas estarán a una distancia $EEL/4$ de las líneas 4ª y 3ª como se muestra en la figura.
- Para el resto de los silencios recorreremos la zona correspondiente a la 3ª línea del pentagrama buscando un conjunto de píxeles negros y comprobando que la ventana segmentada tenga unas mínimas dimensiones.

El reconocimiento de los silencios se hará de diferente manera atendiendo a dos bloques:

- En el caso de los silencios de redonda y blanca, de forma rectangular y situados en el tercer espacio del pentagrama, se decidirá por la situación y las dimensiones de la ventana segmentada.
- En el caso de los silencios desde la negra a la semifusa, donde la forma del símbolo es mucho menos básica, se usará el mismo procedimiento que se usó para el reconocimiento de las claves, esto es, la comparación con máscaras o patrones almacenados previamente.



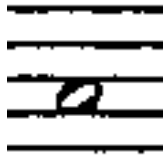
Una vez localizada la eliminamos de la copia de trabajo actual. De la misma forma que hacíamos con las figuras anteriores se localizan los símbolos asociados (alteraciones y puntillos), pasando a una función las coordenadas del centro de la figura y dejando que se encargue de localizar éstos símbolos.

6.12 Localización y reconocimiento de redondas

La redonda tiene aproximadamente las mismas proporciones que la cabeza de la blanca. Haremos una búsqueda ciega, limitada por la ausencia de todos los demás símbolos y por la suma de duraciones del compás.

A diferencia de la localización de los otros símbolos, la redonda puede estar en cualquier posición. El centro de la redonda siempre va a estar sobre la línea o sobre el centro de un espacio, con esto acotamos su búsqueda. También conocemos características de su tamaño: su altura es aproximadamente igual al espaciado entre líneas y su ancho un poco mayor que el alto.

Buscamos puntos negros en el compás, cada vez que encontramos uno estudiamos una ventana a su derecha obtenida de la copia original del trabajo para asegurarnos la figura completa. A partir de acá utilizamos los mismos métodos descritos para localizar las cabezas blancas.



Este proceso es costoso porque se hace para cada píxel negro encontrado en la zona de trabajo, aunque la frecuencia de encontrar una redonda es menos probable.



Una vez localizada la eliminamos de la copia de trabajo actual. De la misma forma que hacíamos con las figuras anteriores se localizan los símbolos asociados (alteraciones y puntillos), pasando a una función las coordenadas del centro de la figura y dejando que se encargue de localizar éstos símbolos.

7 ARCHIVO DE SALIDA

El proceso de reconocimiento produce el almacenamiento de los datos en un archivo de formato LEM, que luego de pasarlo por el convertidor LEM2MIDI, realizado por Carlos Sánchez Barbudo dirigido por Xulio Fernández produce la conversión a formato MIDI.

El archivo LEM es un archivo en formato ASCII, fácil de editar que permite realizar la conversión a MIDI pero también podría utilizarse como base para una conversión a Braille o a otro formato compatible con algún editor de partituras.

El archivo está formado por la secuencia de símbolos reconocidos ordenados, de fácil interpretación. Conteniendo también mensajes de advertencia o de error si estos ocurriesen.

7.1 Especificación de los archivos LEM (Lista de Eventos Musicales)

Los bloques integrantes del archivo LEM, podemos definirlos como:

- Encabezado del archivo
- Cabecera de partitura
- Pista 1
 - Pentagrama 1
 - .
 - .Pentagrama N
- Pista 2
- .
- .
- Pista N
- Fin del archivo

7.1.1 Encabezado del archivo

Contiene los componentes fundamentales en la formación de melodías, tales como:

El número de instrumentos que o voces definidos en la partitura, llamados **pistas** o **tracks**.

El **tempo** con el que corresponde interpretar la partitura. Una tabla contendrá los valores de los parámetros necesarios.

El **compás** que corresponde justamente al compás que indique la partitura y que marcará el valor del mismo.

El archivo debe comenzar con \CP y el el exacte orden en que exponemos debe crearse el archivo:

trk N

Número de pistas

tmp FIG NNN

FIG

NNN

com N D

N: numerador

D: denominador

7.1.2 Pistas

Cada bloque de pista debe comenzar por los caracteres “\TR”. Terminará donde comience el bloque correspondiente a la siguiente pista (“\TR”) o cuando se llegue al final del archivo (“\END”).

Lo primero que debe aparecer en cada bloque de pista son tres líneas, en cualquier orden, que contengan las especificaciones para la clave, alteraciones propias (armadura), y patch de ese bloque de pista. Su formato será el siguiente:

- **cla CLAVE (POS):** Codifica tanto la clave que es como su posición. La clave podrá ser “SOL”, “DO” o “FA”, mientras que la posición será un número de entre “1”, ”2”, ”3” y ”4”. En caso de ser clave de SOL, la posición se deja en blanco. El parámetro POS hace referencia a la línea sobre la que está colocada la clave, por lo que habrá pares de parámetros no aceptados, como SOL 4 o FA 2, ya que no existen como tales en la escritura musical.
- **arm N ALT:** Armadura. ALT se refiere al tipo de alteraciones que la forman y N es el número de ellas que aparecen. Así, para una armadura con tres sostenidos escribiríamos “arm 3 SOS”. ALT pueden ser únicamente las cadenas “SOS” (sostenidos), y “BEM” (bemoles). Caso de ser N=0, nos estamos refiriendo a una armadura sin alteraciones (Tonalidad de Do Mayor).
- **pat P:** Patch. Se trata del patch de la tarjeta de sonido que debería cargarse a la hora de interpretar la melodía. “P” es un número que corresponde al patch. Generalmente coincidirá con el número asignado por el estándar General MIDI.

Posteriormente se codifican los eventos musicales propiamente dichos (notas, silencios, alteraciones, etc...).

Al final de cada compás debe aparecer un punto: “.”. Es estrictamente necesario para facilitar el proceso sintáctico en el conversor y no tener que dotarlo de un sistema que esté pendiente de cuando acaba cada compás en función de los tiempos de las figuras leídas, y actualizar la tonalidad global de la partitura. Por otra parte el incluir esta separación hace mucho más cómoda la visualización del archivo LEM.

Respecto a la codificación de eventos, se definen diferentes etiquetas⁷ con parámetros asociados que son las que siguen:

⁷A efectos de sencillez en los futuros programas de conversión LEM-MIDI o LEM-Braille, se ha tomado la norma de dar a todas las etiquetas (excepto la de fin de fichero), una longitud de tres caracteres. Es lo bastante larga como para ser inteligible, y lo suficientemente corta como para no ocupar demasiado. Así “LEM”, “\CP”, “\TR”, “tmp”, “trk”, “imp”, etc... tienen todas igual longitud.

- **not NO OCT FI AL VE PU LI:** Nota. Se codificará el nombre de la nota (NO), la octava a la que pertenece (OCT), la figura que es (FI), la alteración si la tiene (ALT), si forma parte de un grupo de valoración especial (VE), si existen puntillos (PU), y si existe ligadura (LI). En el campo “NO” serán válidos: DO, RE, MI, FA, SO, LA, SI. En el parámetro ALT serán válidas las etiquetas SOS, BEM, BEC y NUL (sostenido, bemol, becuadro y Nulo). VE será un dígito que indicará si la figura forma parte de un grupo de valoración especial (3= Tresillo, 2= Dosillo, 6= Seisillo, etc...). En el campo de las figuras serán válidas: RED, BLA, NEG, COR, SCO, FUS, SFU (Redonda, blanca, negra, corchea, semicorchea, fusa y semifusa). PU indicará si existen 0, 1 o 2 puntillos (valores válidos), y LI estará a 1 si esa nota está ligada.). Los campos PU y LI son opcionales, pero si LI debe ser 1, PU y VE deberán existir y, en caso de no tener otro valor, ser 0. De igual modo, caso de existir VE y no haber alteración, se rellenará este campo con la etiqueta NUL.
- **sil FI PU LI:** Silencio. Los campos FI, PU y LI tienen idéntico significado que en el apartado anterior.
- **ccl CLAVE (POS):** Cambio de Clave. Habrá que insertar este evento cuando exista un cambio de clave en la partitura. El significado de los parámetros es igual que en el del evento inicial de la pista.
- **car N alt:** Cambio de armadura. Se insertará este evento cuando exista un cambio de tonalidad (generalmente tras una doble barra). El formato de los parámetros es igual que en el del evento inicial de la pista.
- **dob DIR:** Doble barra. Se codificará en DIR la dirección (“I”, “D” o “F”: Izquierda, derecha o fin de pista). En cualquier caso, deberá aparecer tras el punto indicativo de final de compás.

El aspecto de la pista sería algo parecido a:

\TR

cla SOL

arm 4 SOS

pat 1

///

not DO 3 RED 1

not FA 3 COR ⁸

sil RED

.

not SO 4 COR

not SI 3 SCO

sil COR

.

7.1.3 Fin de archivo

El archivo LEM debe acabar con los caracteres “\END”.

\END

⁸Estos valores no tienen sentido “musical”, sólo están aquí para mostrar el aspecto del formato de pista.

7.2 EJEMPLO

A continuación se expone el principio de una partitura con dos pistas, así como el archivo LEM que se obtendría de ella.



```

LEM 1.0
///

///
/// Archivo LEM de salida
Mayo del 2002
///

/// Encabezado de Partitura...
\CP
trk 2
tmp NEG 92
com 3 4

///          PISTA 1
\TR
cla SOL
arm 2 BEM
pat 0

/// pentagrama 1
.
not DO 4 COR NUL 0 1 0
not SI 4 SCO NUL 0 0 0
not DO 4 SFU NUL 0 1 0
not SI 4 FUS NUL 0 0 0
not DO 4 SCO NUL 0 1 0
not RE 4 SCO NUL 0 0 0

.
not DO 4 SCO NUL 0 0 0
not SI 4 SCO NUL 0 0 0
not DO 4 SCO NUL 0 0 0

```

not RE 4 SCO NUL 0 0 0
not DO 4 SCO NUL 0 0 0
not RE 4 SCO NUL 0 0 0

8 TÉCNICAS Y ALGORITMOS APLICADOS

8.1 Algoritmo de Bresenham (Algorithm for Computer Control of a Plotter Digital – IBM Journal 1965)

Este es un algoritmo preciso y efectivo para la generación de líneas de rastreo desarrollado por Bresenham, el cual convierte mediante rastreo las líneas al utilizar sólo cálculos incrementales con enteros que se pueden adaptar para desplegar circunferencias y otras curvas. Este algoritmo es un poco largo para explicarlo aquí, así que será resumido. Para trazar una línea con una pendiente positiva menor que 1.

1. Se capturan los dos extremos de la línea y se almacena el extremo izquierdo en (x_0, y_0) .
2. Se carga (x_0, y_0) en el búfer de estructura, es decir, se traza el primer punto.
3. Se calculan las constantes A_x , A_y , $2A_y$ y $2A_y - 2A_x$ y se obtiene el valor inicial para el parámetro de decisión como

$$p_0 = 2A_y - A_x$$

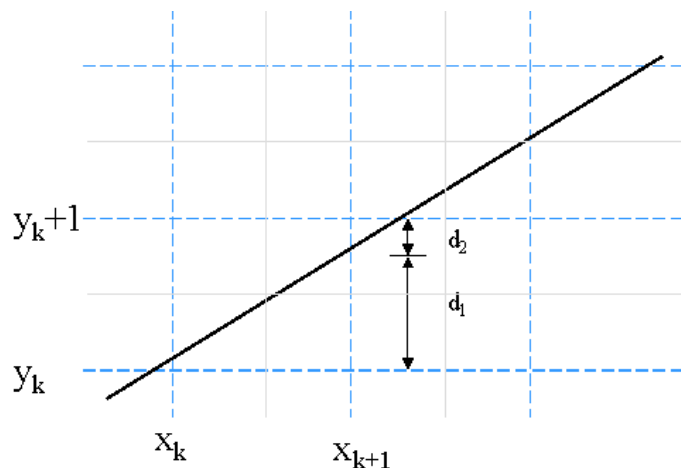
4. En cada x_k a lo largo de la línea, que inicia en $k = 0$, se efectúa la prueba siguiente: si $p_k < 0$, el siguiente punto que se debe trazar es $(x_k + 1, y_k)$ y

$$p_{k+1} = p_k + 2A_y$$

De otro modo, el siguiente punto debe trazarse en $(x_k + 1, y_k + 1)$ y

$$p_{k+1} = p_k + 2A_y - 2A_x$$

5. Se repite el paso 4 A_x veces.



8.2 Thining

El algoritmo de thinning se realiza en la ventana delimitada por las coordenadas: filas inicial y final, y columnas inicial y final.

Nombrando los píxeles vecinos a X de la siguiente manera:

P1	P2	P3
P4	X	P5
P6	P7	P8

Para cada uno de los puntos activos (a negro) de la zona tratada, deben cumplirse una serie de condiciones para eliminar el punto:

1. El número de píxeles vecinos activos debe ser mayor o igual a 2 y menor o igual a 6. Para comprobar esta condición
2. Se recorren los 8 píxeles vecinos al tratado en orden contrario al sentido horario, deberá haber una única transición blanco/negro.
3. Si $P2 * P4 * P8 = 0$ y $P2 * P4 * P6 = 0$ (alguno de esos píxeles está en blanco), se puede eliminar X si se han cumplido (1) y (2).
4. Si $P2 * P4 * P8 \neq 0$, pero el número de transiciones en el vecindario de P2 es distinto de 1, y además $P2 * P4 * P6 = 0$, se puede eliminar X si se han cumplido (1) y (2).
5. Si $P2 * P4 * P8 = 0$ y $P2 * P4 * P6 \neq 0$, pero el número de transiciones en el vecindario de P4 es distinto de 1, y además $P2 * P4 * P6 = 0$, se puede eliminar X si se han cumplido (1) y (2).

8.3 Erosión

Es un método que aplica un proceso de filtrado de píxeles negros según un umbral dado, devolviendo como resultado una imagen en la que sólo quedarán las concentraciones de píxeles negros de mayor extensión.

La profundidad de la erosión está dada por el umbral. A mayor umbral mas desgaste de la imagen.

La erosión que aplicamos se realiza de modo que para cada punto negro de la imagen a erosionar, se estudia el conjunto de puntos conexos al que pertenece, eliminando el punto si la suma de los vecinos sobrepasa el umbral sugerido.



9 DESARROLLO DEL PROGRAMA

“PARTITURAS”

9.1 Descripción del trabajo

9.1.1 Requerimientos Mínimos

- PC con capacidad para Windows 95
- Escáner
- Tarjeta de sonido

9.1.2 Consideraciones generales

El trabajo se desarrolló sobre una plataforma Windows. Está escrito en C++ Builder 4 profesional.

La adquisición de la imagen se realiza mediante el control del scanner con compatibilidad TWAIN.

Las imágenes reconocidas por el programa Partituras deben ser de tipo BitMap (BMP), en blanco y negro. Las partituras a reconocer deben ser impresas, de ningún modo reconocerá partituras manuscritas, debido a que el reconocimiento de los símbolos musicales está basado en las reglas de música impresa. Tales como, la armadura al principio del pentagrama, el equiespaciado entre las líneas de pentagrama, entre pentagramas y entre sistemas. Consideramos que los sistemas se componen de igual número de pentagramas en toda la partitura. El programa no hace reconocimiento de O.C.R. por lo tanto los números de Compás y el Tempo serán ignorados durante el proceso de reconocimiento pero luego serán requeridos para el armado del archivo de salida.

Al escanear, para lograr un reconocimiento óptimo de la imagen, la precisión de adquisición de la misma debe estar entre 200 y 300 píxeles por pulgada(ppp).

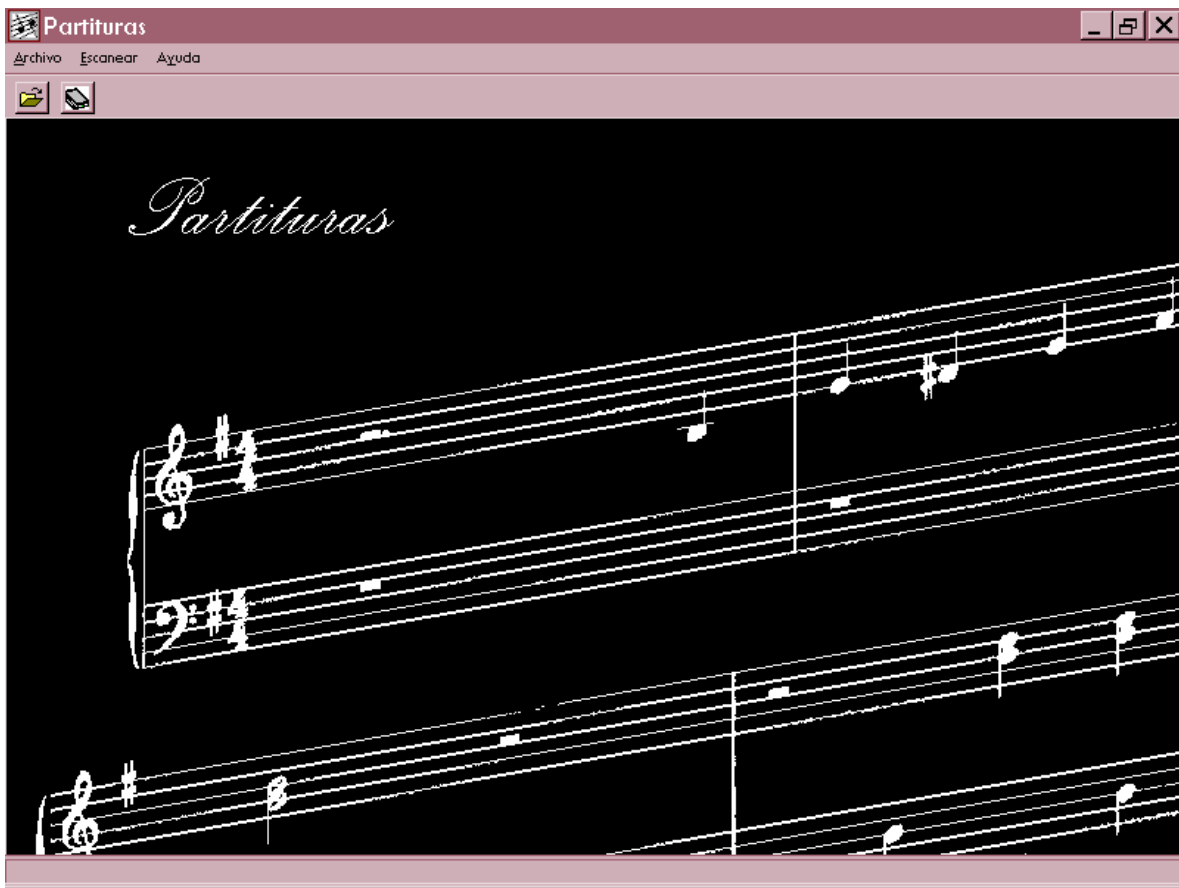
9.2 Entorno visual

Como ya hemos mencionado el programa fue realizado utilizando el Buidex C++ 4.0.

Consta de cuatro pantallas en las que se espera el ingreso de pedido de ejecución de las tareas y otras en las despliega la información.

La pantalla principal consta de un menú desde el cual se puede acceder a las siguientes opciones:

- Abrir un archivo existente que contenga una imagen BMP.
- Seleccionar el escáner con el que se trabajará.
- Ejecutar el escaneo de una imagen que será almacenada en un archivo temporal para ser procesada inmediatamente o que puede ser guardada en un archivo para su posterior procesamiento.



Una segunda pantalla en la que se despliega en una sector de la misma la partitura a procesar y en otro se informan el resultado del proceso a medida que este va transcurriendo. Consta también con un menú principal que permite guardar la imagen si es que ésta se encuentra en un archivo temporal luego del escaneado, como así también consta con la opción de procesar la imagen y el submenú de Utilidades.

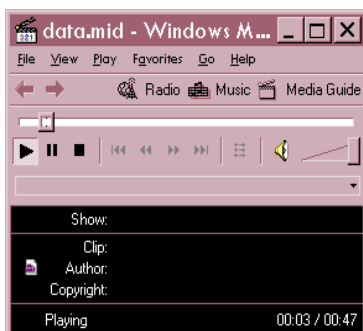
Cabe destacar que durante el procesado de la imagen la imagen de la partitura irá cambiando mostrando los distintos tratamientos que se le va realizando con el fin de extraer de ella la información buscada. Paralelamente se va comentando por escrito los resultados que arrojan dichos tratamientos. Por ejemplo la eliminación del pentagrama, la erosión de la imagen para hallar las cabezas negras, el borrado de los símbolos a medida que se van reconociendo, etc...



Las tercera y cuarta pantallas se desplegarán durante el proceso con el fin de solicitar que el ingreso del Tempo, el Compás y el Patch, variables indispensables para el armado del archivo de salida. El ingreso de Patch, indica con que instrumento se ejecutará la partitura y se indica por cada voz de la partitura. Por ejemplo si la partitura agrupa sus pentagramas en sistemas de 2, se pedirá 2 veces el ingreso de un Patch.

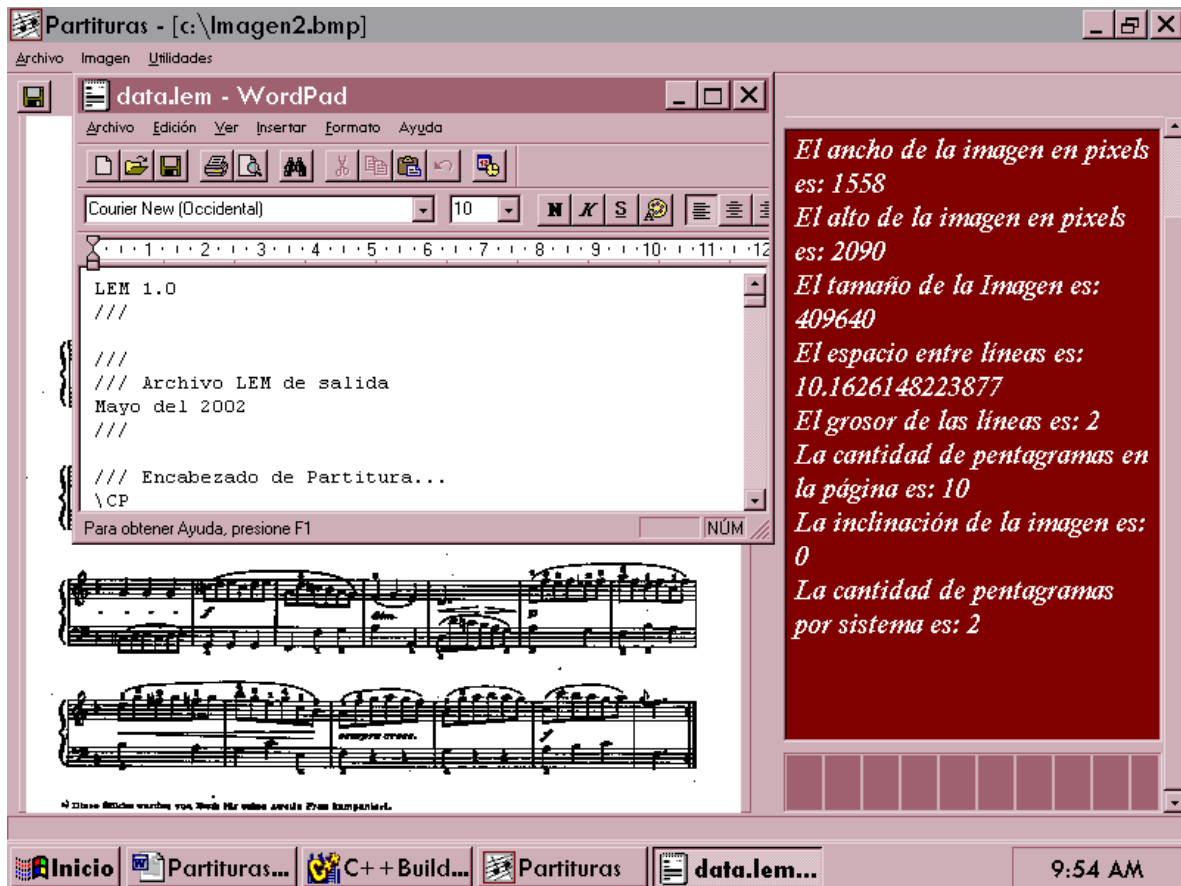


El submenú Utilidades consta de dos opciones, una de ellas es Escuchar la partitura y la otra es visualizar el archivo LEM.



Escuchar la partitura activa el reproductor Media Player ejecutando el archivo MIDI correspondiente a la partitura en proceso.

Visualizar LEM activa el WordPad editando el archivo LEM generado en el reconocimiento.



9.3 Conocimiento a priori

Como explicamos anteriormente el reconocimiento de música está muy sujeto al conocimiento de las reglas de escritura de música impresa. Del conocimiento de estas reglas surge la aplicación de gran parte de las técnicas usadas.

Algunas de estas reglas se detallan a continuación:

- La posición de las notas.(do ,re,..)⇒ trabajo con tablas
- El formato de las notas. (redonda, blanca,..) ⇒ Erosión para las cabezas negras, Thinning en las blancas y redondas, Erosión y trabajo con segmentos para los corchetes o puentes, etc
- Equiespaciado entre las líneas de los pentagramas, entre los pentagramas y entre los sistemas. ⇒ trabajo con histogramas

- Igual grosor de las líneas en los pentagramas. \Rightarrow trabajo con histogramas.
- Al principio de cada pentagrama la existencia de la armadura (Clave más tonalidad.) \Rightarrow Búsqueda de la clave y posterior tonalidad al principio de cada pentagrama.
- La existencia de tonalidades. \Rightarrow trabajo con tablas
- La posición hacia arriba o hacia debajo de las plicas o palos de las notas. \Rightarrow Buonding box de la zona exacta para su análisis
- La posición del puntillo. \Rightarrow Buonding box de la zona exacta para su análisis
- La posición de las alteraciones accidentales. \Rightarrow Buonding box de la zona exacta para su análisis
- El cumplimiento de los tiempos de valor del compás. \Rightarrow Permite controlar la búsqueda.
- El tamaño de las calves proporcional al interlineado. \Rightarrow Reconocimiento por tamaño
- El tamaño de las barras indicando el tamaño de los sistemas. \Rightarrow Trabajo con Erosión y Thinning en la zona específica.
- El tamaño de las barras de compás indicando fin del compás. \Rightarrow Realizar el guardado de la información por compás.

9.4 Descripción del programa

9.4.1 Ingreso de la imagen

9.4.1.1 *Imagen tomada del escáner*

Las funciones utilizadas son las predefinidas para el trabajo con escáner, una vez agregada la librería TWAIN que habilita la posibilidad de agregar el Active X correspondiente. El archivo con la imagen queda en un archivo temporal que se tratará al igual que se explica en el punto siguiente.

9.4.1.2 *Imagen tomada de un archivo*

Simplemente un *CreateMDIChild(OpenDialog->FileName)* que permite guardar el nombre del archivo que abrirá la imagen en otra pantalla.

Archivo: Main.cpp

9.4.2 Procesamiento

9.4.2.1 Extracción de datos de la imagen y almacenamiento de la imagen en memoria

Para la extracción de los datos la función *LeeBMP(LPSTR nombrefich)* hace la apertura del archivo *nombfich* en binario, extrae los datos como ancho, tamaño, alto, etc de la BITMAPINFOHEADER.

Luego aloca el lugar de la imagen en memoria, se copia la imagen a memoria y la función retorna el puntero a la misma.

Archivo: E_s_BMP.cpp

9.4.2.2 Cálculo del espaciado interlineal y el grosor de las líneas

En la función *fija_EIL*, los segmentos blancos y negros son almacenados en la estructura *línea* y luego se separan en blancos en la estructura *histoB* de la que surgirá del histograma la variable *EIL* que contiene el interlineado y los segmentos negros se tratarán de forma similar en *histoB* de donde surgirá *GRO*. que contiene el grosor de las líneas.

Archivo: Eil.cpp

9.4.2.3 Búsqueda de la cantidad y posición de los pentagramas

La función *busca_pentagramas*, utilizando la estructura *línea*, se realiza el proceso dejará guardado en la estructura *pentagramas* las posiciones del principio de cada pentagrama de. Las posiciones de las primeras líneas de todos los pentagramas de las tres columnas son almacenadas, los punteros al principio de cada columna están en el vector *columnas*. La columna que contenga el máximo de pentagramas encontrado será la que *PENTA* y *Col_Pentagrama* guardará el número de la columna.

Archivo: Pentagramas.cpp

9.4.3 Búsqueda del formato de la partitura

La función *busca_formato_thin* actualiza *SIS* que es el número de pentagramas por sistema y en *CANT_SIS* el número de sistemas. Tal como se explica la descripción, se trata básicamente de tomar una parte de la imagen y realizar sobre ella varios procesos de erosión y thinning. Las variables que delimitan la zona de la imagen con la que se trabaja son *lim_izquierdo*, *lim_derecho*, *lim_superior* y *lim_inferior*. La longitud máxima se almacena en la variable *Lmax*.

Archivo: Pent_Sistemas.cpp

9.4.4 Verificación de la inclinación de los pentagramas

La función *calcula_inclinación* devuelve la inclinación encontrada en la partitura, en grados. Además devuelve en las variables *dx* y *dy*, los valores de *Delta_X* y *Delta_Y* que se mencionaban al hablar del algoritmo de Bresenham.

La función usa las variables *columna_izquierda* y *columna_derecha* para marcar las columnas a izquierda y derecha de la imagen en donde extraeremos la posición del primer pentagrama almacenados en la estructura *pentagramas* durante la búsqueda de la cantidad de pentagramas.

Archivo:Inclinación.cpp

9.4.5 Eliminación de los pentagramas

Mediante la función *eliminar_pentagramas*, se elimina los pentagramas en la imagen manejada por *gin*, y deja una copia procesada en *gout*.

Tal como se explica en la descripción correspondiente, se usan cinco columnas para que nos sirvan de guía mientras se eliminan los pentagramas por segmentos. Las posiciones en horizontal de estas columnas están dadas por las variables *v*, *vi*, *vd*, *vim* y *vdm*, y corresponden a las cinco posiciones espaciadas 1/6 del ancho de la imagen. Se buscan las posiciones de las primeras líneas de los pentagramas en las cinco columnas como guías para producir el borrado en los seis tramos. Esos tramos son las zonas horizontales en las que queda dividida la imagen, si la segmentamos por las columnas: *col*, *coli*, *cold*, *colim*, *coldm* (parámetros que se pasan a la función *quita_pentagrama_delta* (UINT *col*,UINT *coli*,UINT *cold*,UINT *colim*,UINT *coldm*,Punt_imag *gin*, Punt_imag *gout*).

Las variables *iniup* e *inidown* controlarán el grosor del tramo en vertical, hacia arriba y hacia abajo respectivamente.

Archivo:Eliminar_pentagramas.cpp

9.4.6 División por compases

9.4.6.1 *Compases simples*

En la función *busca_compases_simple* se realiza la búsqueda de las barras divisorias (delimitación de compases), en el caso que se haya detectado previamente que no existen sistemas. Es más

complicada que la función análoga que se encarga de hacer lo mismo en partituras con sistemas, ya que en este caso no podemos hacer proyecciones como fue explicado en la descripción.

Un parámetro importante a tener en cuenta es la variable *tramo*, que denota la altura de la banda horizontal proyectada sobre el eje X. La variable *umbral* está fijada en el 95% del máximo esperado en la altura de la barra divisoria, esto es, $4*EIL$.

Se observa que proyección de las zonas inmediatamente anterior y posterior a la proyección de la barra divisoria. Esto es crucial para poder distinguir entre auténticas barras divisorias y plicas pertenecientes a figuras blancas.

9.4.6.2 *Compases de sistemas*

En el caso de la *busca_compases_sistema* se realiza la búsqueda de las barras divisorias (delimitación de compases), en el caso que se haya detectado previamente que existen sistemas. La función buscará las barras divisorias en la imagen apuntada por el puntero que se le entrega (*punt*), basándose en los datos almacenados previamente en la estructura *pentagramas*.

Se hace una proyección por el ancho de la imagen, que se almacena en la lista *proyeccion_penta* de donde se seleccionaran las posibles barras (aprox. el 80% del valor máximo).

Se fue almacenando las barras divisorias encontradas no en el orden “natural” en cuanto a la lectura musical se refiere sino en otro distinto. Esto provoca que la lista enlazada en donde tenemos los datos de las barras divisorias no esté ordenada: lo que se hace es ordenar esa lista de datos *info_compás*.

Cuando la distancia en horizontal entre dos barras divisorias consecutivas es menor que un cierto umbral (fijado en dos veces el espaciado entre líneas ($2*EIL$)), se decide que se trata de una doble barra y hacia qué lado es doble barra., almacenando ese dato la lista enlazada

Archivo: Busca_compas

9.4.7 **Eliminación de barras divisorias**

Mediante la función *elimina_barras_verticales* se ejecuta la eliminación de todas las barras divisorias presentes en la partitura (referenciada por el handle *gin*), resultando la imagen referenciada por *gout*.

Se basa en los datos almacenados en la lista enlazada de estructuras tipo *info_compas*, en donde están las coordenadas de las barras divisorias encontradas. Para cada una de estas barras, la función llamará a *borra_barra_vertical()* para que esta última se encargue de la eliminación de esa barra divisoria que realiza un proceso hacia arriba y hacia abajo del punto medio aproximado, y que consiste en recorrer la barra vertical eliminando todos los píxeles de cada fila.

Archivo: *eliminar_barras.cpp*

9.4.8 Reconocimiento de símbolos

La función *reconocimiento_basico(g_or,gin)* es la que controla todo el procesado a nivel de compás.

Requiere que se ingresen las características necesarias para reconocer los símbolos musicales y poder escuchar la partitura reconocida. Las características solicitadas son:

- Número de Compás
- Tempo/Aire
- Patch

A continuación se ejecuta la función *fija_control_compas* que mediante el cálculo de la variable *control_compás* devuelve el número de partes de tiempo de la duración de la semifusa, que deberíamos esperar encontrar en un compás completo. Este valor devuelto se usará para hacer comprobaciones y decidir si debemos seguir buscando símbolos o no dentro de un determinado compás.

Seguidamente, para cada uno de los pentagramas realiza los siguientes pasos:

- Busca la clave
- Reconoce la tonalidad
- Reconoce las figuras musicales

Archivo: *proceso_reconocimiento.cpp*

9.4.8.1 *Busca la clave*

Mediante la función `busca_clave(finí,cini,cfin, lín, *despl,gin)` se encuentra la clave del pentagrama, esta función devolverá un número de acuerdo a la clave que localice:

1. Clave de Sol.
2. Clave de Fa en 4ª línea.
3. Clave de Fa en 3ª línea.
4. Clave de Do en 1ª línea.
5. Clave de Do en 2ª línea.
6. Clave de Do en 3ª línea.
7. Clave de Do en 4ª línea.

Archivo *Claves.cpp*

Lo que hace esta función es acotar horizontal y verticalmente la clave. La acotación horizontal se realiza desde la columna inicial *cini* hasta la columna final *cfin* mediante una proyección vertical en la que se debe superar un umbral fijado en $2 * \text{GRO}$. La acotación vertical se realiza por encima o por debajo de la zona que está entre las columnas obtenidas tras la acotación horizontal, comenzando por la fila *fini*. De esta manera tenemos acotada completamente la clave.

Seguidamente determinamos que tipo de clave es. Las diferenciamos únicamente por el tamaño vertical. En el caso de la clave de Sol sólo puede aparecer en una posición, mientras que la de Fa podría aparecer en dos lugares y la de Do en cuatro. Para las claves de Fa y Do, localizaremos el centro de la figura a partir de la primer línea de pentagrama *lín*, mediante la función `get_pos()`, y decidiremos así la localización exacta de la clave.

En *despl* devuelve la columna a partir de la cual se continuará la búsqueda de la tonalidad luego de hallar la clave.

9.4.8.2 *Reconoce la tonalidad*

La función `reconoce_tonalidad(clav,p,finí,cini,cfin,gin)` devolverá un número entre 0 y 14 dependiendo del tipo de tonalidad que se encuentre:

0: Ninguna alteración

1: Un sostenido

.

7: Siete sostenidos

8: Un bemol

.

14: Siete bemoles

Archivo: *Tonalidad.cpp*.

Lo primero que hace esta función es delimitar una posible alteración comenzando en la fila *fini* y la columna *cini* en el pentagrama *p*, hasta la fila (*fini*+4*EIL) y la columna *cfin*, mediante una proyección vertical en la que se debe superar un umbral fijado en $5 \cdot \text{EIL} / 4$. En el caso que haya encontrado una alteración o algún símbolo, vemos si se trata de un bemol, un sostenido o ninguno de los dos, para ello se observará las posiciones en las que podría esperarse el primer sostenido o bemoles de la tonalidad para la clave *clav* mediante la función *averigua_tipo()*. Si se encontró que es bemol o sostenido se continúa buscando posibles alteraciones.

9.4.8.3 Reconoce las figuras musicales

Primero la función *reconoce_compas(fini,ffin,cini,cfin,gor,source,aux1,aux2,aux3,aux4,aux5,aux6,lin,comp,clav)* erosiona convenientemente la ventana en la que está contenido el compás.

A continuación se buscan las figuras teniendo en cuenta que los tiempos que ocupan las símbolos localizados deben ser exactamente lo esperado, este resultado se guarda en la variable COM.

Se comienza buscando cabezas negras desde la fila *fini* y la columna *cini* hasta la columna *cfin*, en el compás *comp* y el pentagrama *lin* con la clave *clav*. Si la suma de los tiempos de las cabezas negras localizadas no completan el compás o no se encontraron cabezas negras, se buscan cabezas blancas, a continuación si la suma de los tiempos aún no completan el compás o no se encontraron cabezas blancas, se buscan silencios y si la suma de los tiempos sigue sin completar el compás o no se

encontraron silencios se buscan redondas. Si la suma de los tiempos encontrados no completan el compás significa que hubo algún error durante el reconocimiento.

Archivo: *Cabezas_Negras.cpp*.

1.1.1.1 Figuras negras

La función *busca_cabezas_negras_bar(fini,cini,cfin,ffin,fini,g_src,g2,g3,g4,g5,g6,lin,comp,clav)* es la que se encarga de localizar las cabezas negras, comprobar si forman parte de figuras, localizar las plicas y corchetes y gestionar la localización y reconocimiento de los elementos auxiliares (alteraciones, puntillos, etc...).

Lo primero que hace esta función es definir umbrales (*min_an*, *min_al*, *max_an*, *max_al*) que permitirán determinar si es una cabeza negra o no.

A continuación teniendo en cuenta que la copia de trabajo pasada como *g2* es la copia en la que se encuentra la zona a estudiar, ya erosionada, lo que hace la función es obtener el bounding box (ancho y alto) de la cabeza negra, luego se comprueba si existe un cierto porcentaje de píxeles negros dentro del bounding box mediante la función *relleno()* y si es así entonces se calcula el centro de la misma para así conocer su posición en el pentagrama. Una vez que se encontró la cabeza negra se localiza la plica y corchetes, eliminando estos elementos de la capa de imagen una vez localizados.

A partir de este momento localizamos y reconocemos los símbolos accidentales a las figuras con cabeza negra (las alteraciones y los puntillos) y los eliminamos de la capa de imagen una vez localizados.

A continuación en función de la duración de la figura localizada hay que añadir tiempo. Entonces si por ejemplo se encontró una negra, a la suma de tiempos del compás (COMbar) hay que añadirle 16 partes (porque la negra está formada por 16 semifusas) y 8 partes más (la mitad), si existe un puntillo acompañando a la negra (*pu=1*).

Finalmente se elimina la ventana procesada de la copia de trabajo.

Archivo: *Cabezas_Negras.cpp*

1.1.1.2 Figuras Blancas

La función *busca_blancas_bar(finí,cini,cfin,ffin-finí,g_or,g_src,g3,g4,lin,comp,clav)* es la que se encarga del control de la localización y reconocimiento de las figuras blancas dentro de un compás.

Lo primero que hace esta función es definir umbrales (*umbral_ancho*, *umbral_alto*) que permitirán determinar si es una cabeza blanca o no.

A continuación se realiza una proyección vertical de toda la zona de estudio, para obtener una posible plica blanca que supere el umbral *umbral_plica* fijado en $2.5 * EIL$. Luego se obtiene el centro de la posible cabeza blanca y se realiza un thinning de esa ventana. A continuación se realiza un recorrido interno del contorno en sentido horario mediante la función *comprueba_cabeza_blanca()* y sólo si ese recorrido del contorno se ha cerrado, se estudia el tamaño del contorno y se chequean los umbrales antes mencionados. Finalmente se decide si era una cabeza blanca o no.

Si lo es, se estudia la posición de la cabeza y la localización de los posibles símbolos asociados (alteraciones propias y puntillos) mediante la función *procesa_blanca()*. Si no estuviera en esa posición la cabeza se estudiaría el otro extremo de la plica.

En el caso que la cabeza blanca se encuentre sobre una línea, se eliminará el trazo interno de la cabeza mediante la función *quita_linea()* y se continuará con el proceso mencionado.

Archivo: *Blancas.cpp*.

1.1.1.3 Silencios

La función *busca_silencios_bar(finí,cini,cfin,ffin-finí,g_or,g_src,g3,g4,lin,comp)* es la que se encarga de localizar y controlar el reconocimiento de los silencios en un compás.

Lo primero que hace esta función es localizar posibles silencios de redonda y de blanca. Luego entonces se centra en la localización de los restantes tipos de silencios. Para cada caso se fija si hay puntillo.

Archivo: *Silencios.cpp*

1.1.1.4 Redonda

La función *busca_redondas_bar(cini, cfin, g_or, g_src, g3, g4, lin, comp, clav)* es la que se encarga de la localización y reconocimiento de las redondas dentro de un compás. Se recorrerán las 31 líneas horizontales (o ligeramente inclinadas) donde podrían estar el/los centros de las posibles redondas.

Si se llega a un punto a negro, durante el recorrido de una línea, se segmenta una ventana lo suficientemente grande como para poder contener una redonda.

Posteriormente se hace un recorrido interno del contorno en sentido horario. A continuación, y sólo si ese recorrido del contorno se ha cerrado, se estudia el tamaño del contorno. Finalmente se decide si era una cabeza blanca o no.

Si lo es, se estudia la posición de la cabeza como para localizar los posibles símbolos asociados (alteraciones propias y puntillos).

Archivo: *Redondas.cpp*.

9.4.9 Escritura del archivo de Salida

La función *escribe_archivo_LEM* justamente escribe el archivo de salida cumpliendo las especificaciones de los archivos LEM, descargando datos almacenados en la lista *símbolos* que fue creada durante el reconocimiento.

Archivo: *Escribir_LEM.cpp*

9.4.10 Manejo de errores

La función *error* selecciona el error correspondiente al código que se le pasa como parámetro. Los errores aparecerán cuando durante el procesado de imagen desplegados por la función *mostrar_error* que se encuentra en *ChilWin.cpp*, y seguidamente se anulará la operación, como por ejemplo que no hay suficiente memoria, que la inclinación de la partitura sea superior a la permitida o que no puedan detectarse parámetros importantes como el espaciado entre líneas o el grosor.

Archivo: *errores.cpp*

9.5 Cuadro comparativo

El cuadro muestra la secuencia de trabajo que dado los requerimientos musicales, se fue desarrollando el software aplicando las técnicas o métodos que fueron necesarios:

Análisis del Formato de la Imagen		
Partitura a procesar	Escaneo de la imagen o apertura de un archivo Leer la imagen en binario Obtener las dimensiones Rechazo de las no aptas	
Análisis de la ubicación de las grandes componentes de la imagen		
Pentagramas <ul style="list-style-type: none"> • Interlineado • Grosor de las líneas Sistemas	Busca_EIL Busca_pentagramas Busca_sistemas <ul style="list-style-type: none"> • 	Histogramas y trabajo con segmentos Proyecciones Erosión Thinning.

Reconocimiento de símbolos

Separar en compases

Por compás:

Si es el primero

- Reconocer clave
- Reconocer armadura

Los demás y siguiendo hasta completar el valor del compás, reconocer:

- Cabezas negras
- Palos o plicas
- Corchetes o puentes
- Si es blanca
- Si es silencio
- Si es redonda

En todos los casos comprobar la existencia de alteraciones accidentales y puntillos

Busca_compás

Busca_clave
Busca_armadura

Busca_cabezas_negras
Busca_
Busca_corchetes

Busca_blancas
Busca_silencios
Busca_redondas

Busca_alteraciónaccidental
Busca_puntillos

Proyecciones

Histogramas
Tratamiento de Tablas
Conocimiento a priori.

Erosión
Thinnig
Macheo

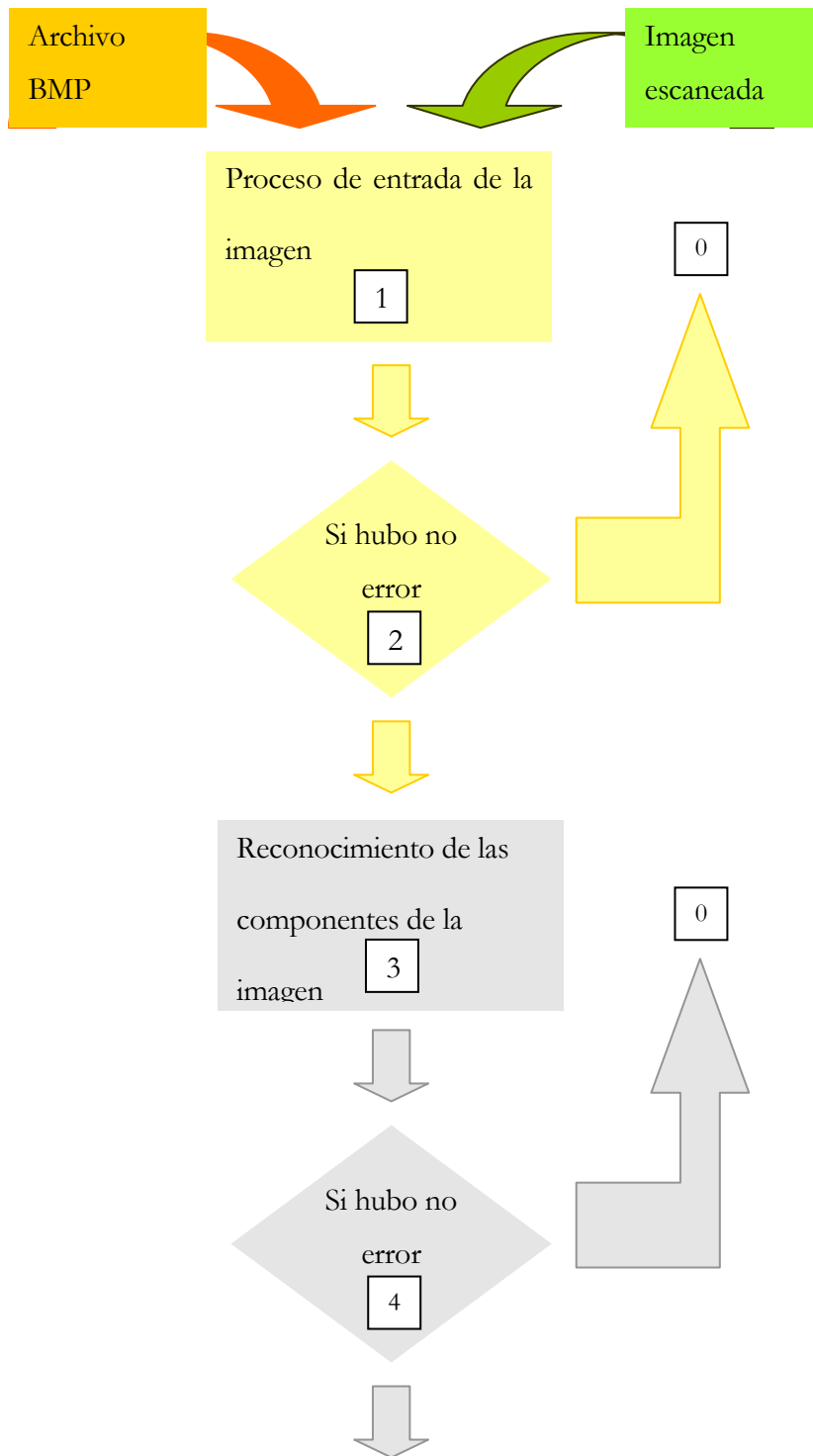
Archivo de Salida LEM

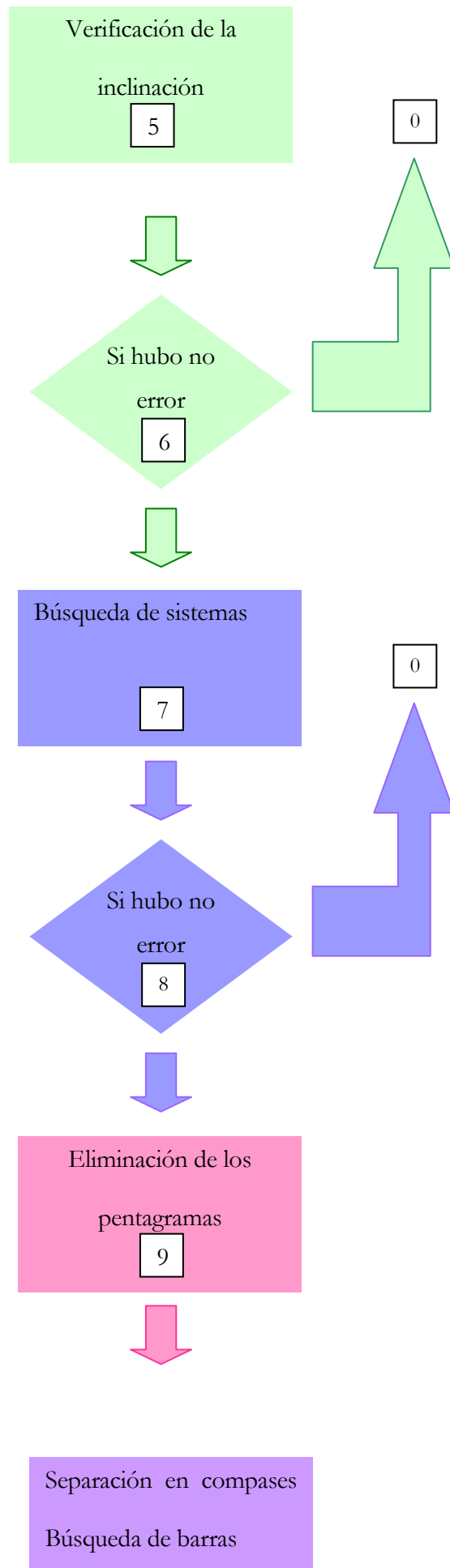
Obtención de un archivo MIDI que puede oírse y editarse..

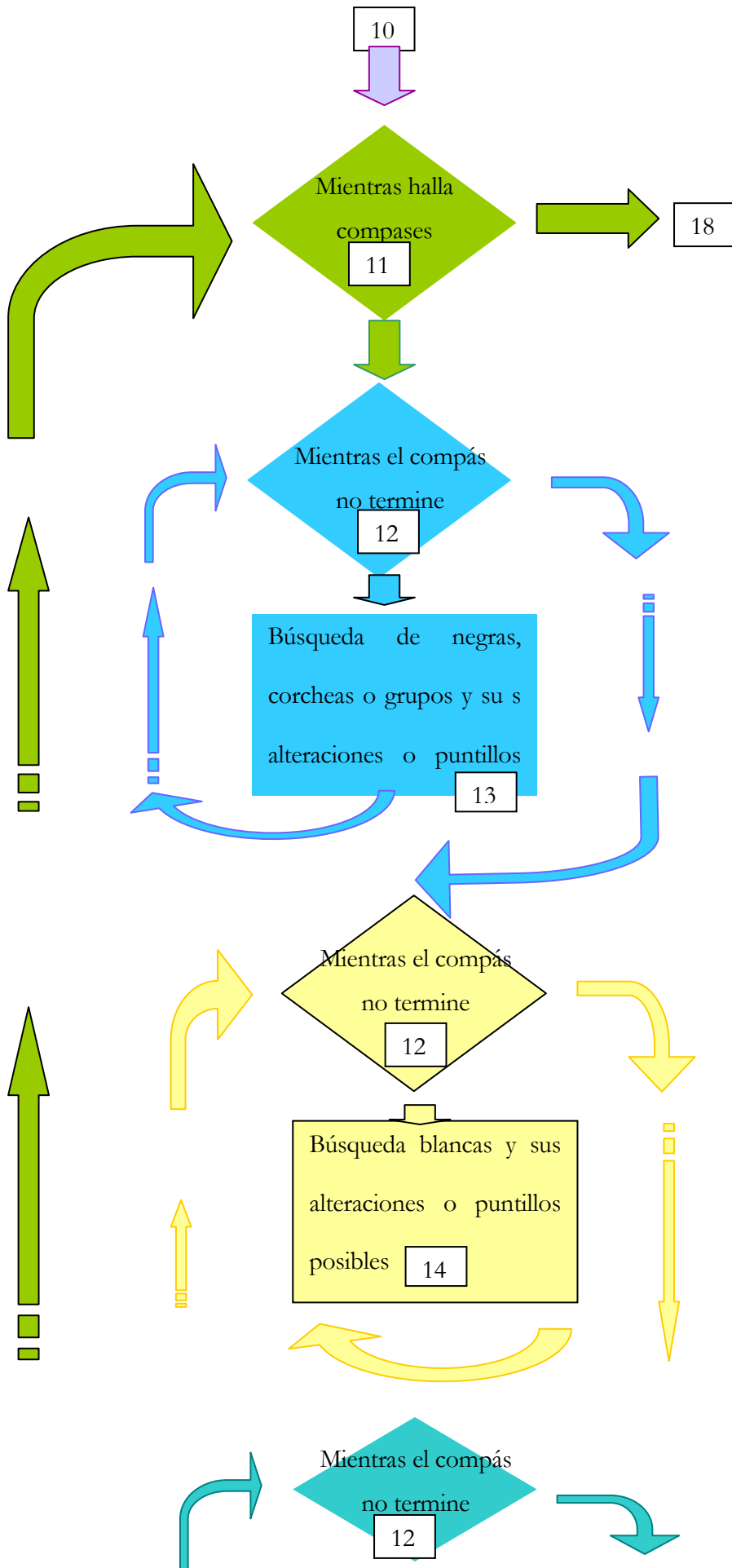
Escribe_LEM donde se escribe la secuencia de símbolos reconocidos en la partitura en un orden que cumpla las especificaciones LEM. Ejecutar el conversor LEM2MIDI para obtener la salida de un archivo .MID.

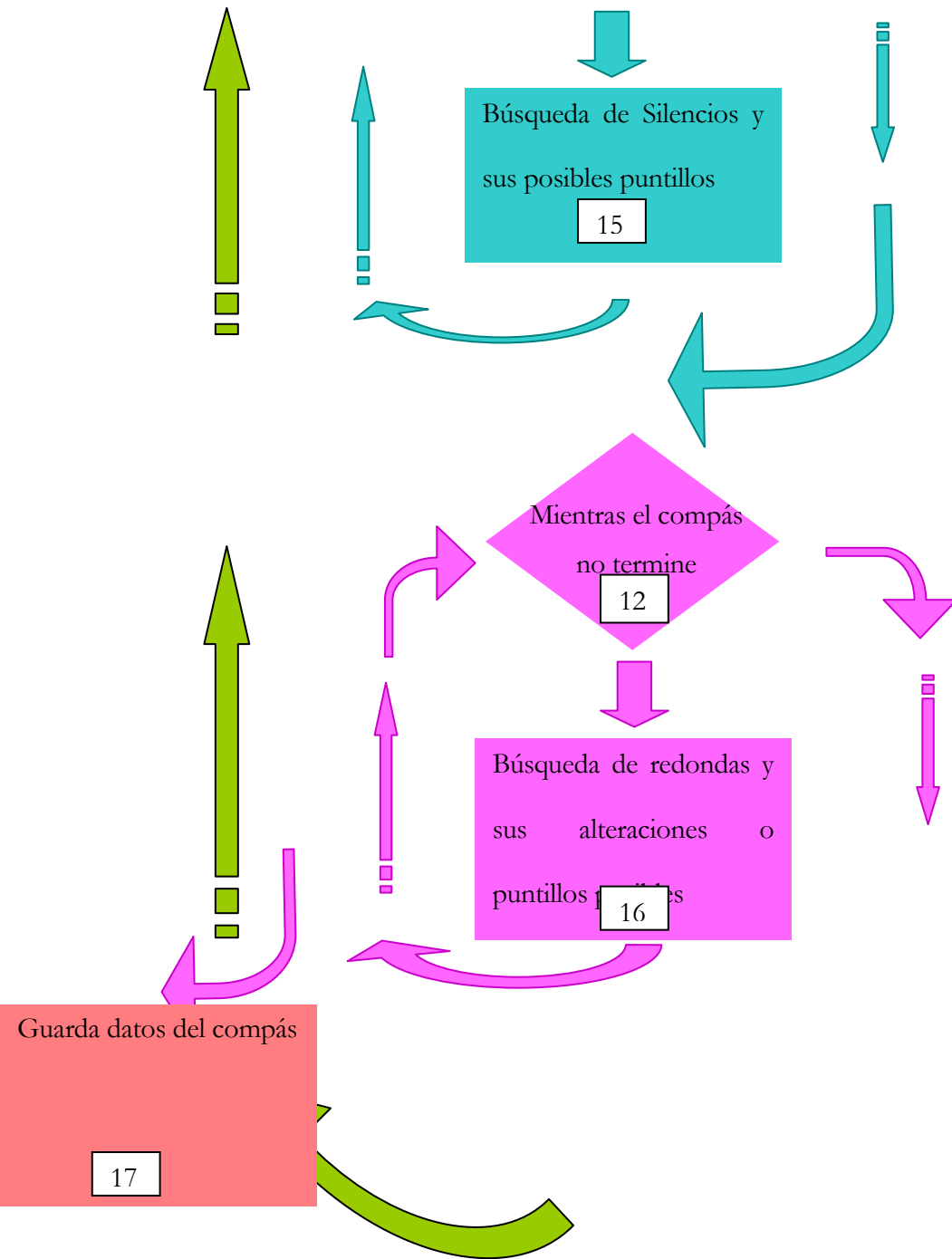
Conocimientos de especificaciones LEM y MIDI.

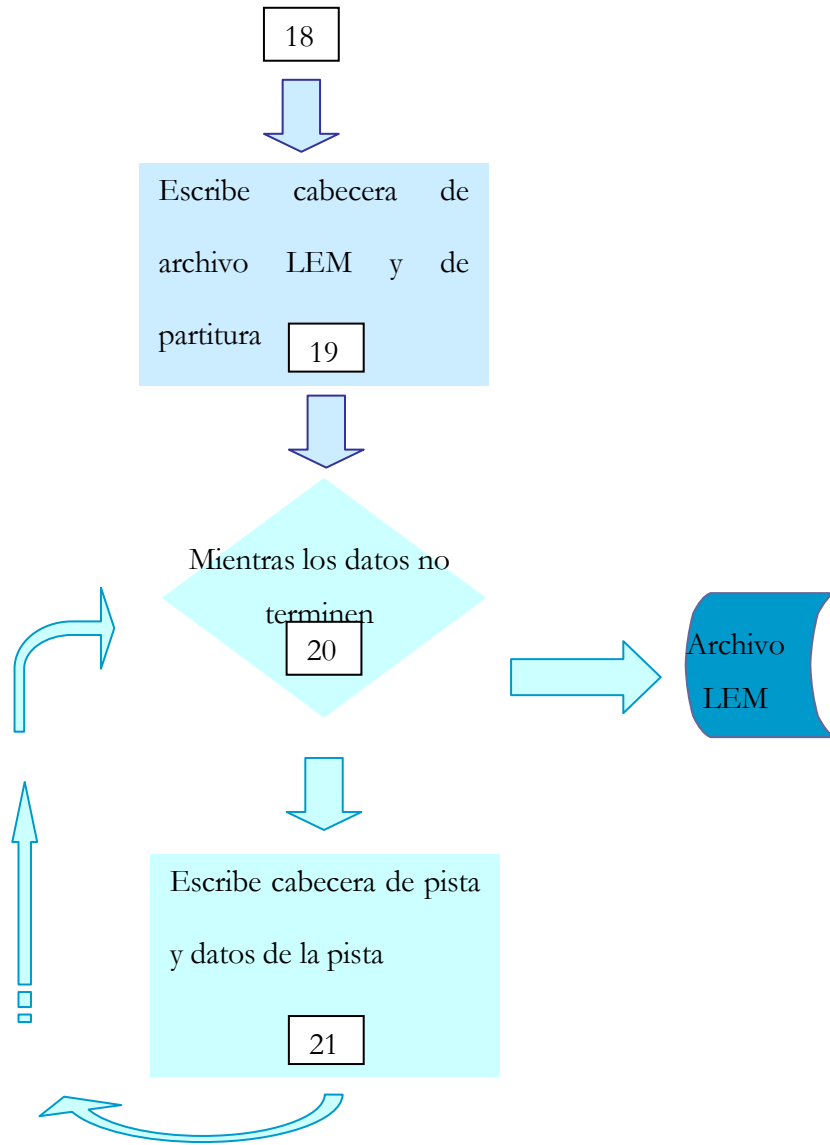
9.6 Diagrama de flujo











10 RESULTADOS OBTENIDOS

Para presentar los resultados del reconocimiento de las partituras, fue necesario establecer una forma cuantitativa de los mismos.

De manera que elegimos las siguientes tasas de reconocimientos parciales y las agrupamos según su importancia en una tasa de reconocimiento global.

Las tazas parciales son:

- Tasa de reconocimiento de compases 90%
 - Tasa de reconocimiento de sistemas 85%
 - Tasa de reconocimiento de pentagramas 95%
- 90%
- Tasa de reconocimiento de figuras 85%
 - Tasa de reconocimiento de silencios 80%
 - Tasa de reconocimiento de notas 75%
 - Tasa de reconocimiento de alteraciones accidentales 85%
 - Tasa de reconocimiento de puntillos 95%
- 83%
- Tasa de reconocimiento de Armadura: 90%
 - incluye clave
 - Tonalidad

La tasa global se calculará como: 40% ● +50% ● +10% ●

El resultado obtenido para 5 partituras que realizaron todos los procesos fue del 87%, es un buen resultado. Las partituras Las fallas pueden ser debido a una mala localización de los pentagramas, que provoca problemas con la localización de sistemas, y la posición de las notas.

11 LÍNEAS FUTURAS

Dada la complejidad que se presenta frente a la variedad de tipografías, normas y símbolos en las partituras impresas, consideramos que esta aplicación puede ser ampliada en sus capacidades en los siguientes aspectos a detallar:

- Sumar la posibilidad de reconocimiento de los símbolos tales como letras y números con O.C.R.:
- Reconocimiento de:
 - Acordes: su implementación no es demasiado complicada de adjuntar.
 - Ligaduras
 - Símbolos de tamaño reducido
 - Grupos de valor especial: dosillos, tresillos, etc.
 - Manejo de partituras de varias páginas
 - Manejo de varias partituras.
- Implementación de un conversor LEM-Braille.

12 CONCLUSIONES

La aplicación “*Partituras*” presentada es el resultado de la investigación y las pruebas realizadas en un área poco estudiada desde la perspectiva del procesamiento y reconocimiento de imágenes.

La programación se desarrolló en un entorno visual de fácil interpretación. Durante la ejecución se van describiendo las distintas etapas del reconocimiento facilitando la comprensión de la problemática musical. Esta visualización de las etapas se implementó pensando en la presentación del sistema como trabajo de investigación.

El tema tratado es muy amplio por lo que podría ser continuado en el sentido de perfeccionar y/o expandir algunos de los conceptos en él introducidos o también en otras áreas del tratamiento musical.

13 BIBLIOGRAFÍA

Xulio Fernández Hermida. Francisco Martínez Rico. *Reconocimiento Óptico de Música Impresa (O.M.R.). Desarrollo Visual y ampliación de Capacidades.* Universidad de Vigo, 2000

J.R.Parker. *Algorithms for Image Processing and computer Vision* 1996

N.P. Carter . *Automatic recognition of printed Music in the context of Electronic Publishing.* Phd Thesis, Universidad de Surrey, 1989

M ferrand. J. Leite. Amilcar Cardoso. *RIEM A system for Reconogtion and Interpretation of Musical writing.* Universidad de Coimbra y Universidad Nueva de Lisboa. Portugal. 1998

Venus, Alberto. MIDI Media Adaptation Layer for IEEE-13942001

C: MANUAL DE REFERENCIA -*Schildt,* Herbert Osborne/McGraw Hill

M ferrand. J. Leite. Amilcar Cardoso. *Improving Optical Musical Reconogtion by means abductive Abductive Constraint Logic Programing.* Universidad de Coimbra y Universidad Nueva de Lisboa. Portugal. 1999

Ichiro Fujinaga, Stephan Moore, David S. Sullivan, Jr. *Implementation of exemplar-based learning model for music cognition.* Peabody Institute, Johns Hopkins University, 1999

Libro de Teoría de la Música – *Alberto Williams-*

Apuntes de Teoría Musical – *G.Troinai y H.Forino*

14 INDICE

1	INTRODUCCIÓN	1.1—3
1.1	Aplicaciones	1.1—4
1.2	Comparación con Reconocimiento óptico de caracteres	1.2—5
2	Introducción a la música	1.2—7
2.1	Elementos del arte musical	2.1—7
2.2	Sonido	2.2—7
2.2.1	Cualidades del sonido	2.2—8
2.2.1.1	Altura o tono	2.2—8
2.2.1.2	Timbre	2.2—8
2.3	Intervalos musicales	2.3—8
2.4	Pentagrama	2.4—9
2.5	Notas musicales	2.5—9
2.5.1	Partes de la Nota	2.5—11
2.6	Grupos de notas	2.6—12
2.7	Acordes y arpeggios	2.7—12
2.8	Puntillos y puntos de aumento	2.8—13
2.9	Accidentes musicales, tonalidades y armaduras	2.9—13
2.9.1	Sostenido:	2.9—13

2.9.2	Doble sostenido	2.9—13
2.9.3	Bemol	2.9—13
2.9.4	Doble bemol	2.9—13
2.9.5	Becadro	2.9—14
2.10	Ligaduras	2.10—14
2.11	Indicadores de compás	2.11—15
2.12		2.12—15
2.13	Barras divisorias	2.13—15
3	<i>Reconocimiento de partituras</i>	2.13—16
3.1	Problemas	3.1—17
3.1.1	Decoloración del documento	3.1—17
3.1.2	Fragmentación y pliegues en la imagen	3.1—17
3.1.3	Ruidos de alta frecuencia	3.1—17
3.1.4	Rotación o distorsión global de la imagen	3.1—18
3.1.5	Intersección de símbolos	3.1—18
3.1.6	Dificultades en el reconocimiento	3.1—18
3.1.7	Variación de tamaño de los símbolos	3.1—18
3.1.8	Dificultades de Interpretación	3.1—18
3.1.9	Métrica incompleta	3.1—18
4	<i>Técnicas usadas por distintos sistemas existentes</i>	3.1—19

4.1	Análisis y segmentación de documentos – Detección del pentagrama	4.1—20
4.1.1	Xulio Fernandez Hermida y Carlos Sánchez-Barbudo Vargas	4.1—20
4.1.1.1	Eliminación de las líneas del pentagrama	4.1—21
4.1.1.2	Localización de las zonas a procesar	4.1—21
4.1.2	RIEM	4.1—22
4.1.2.1	Detección de las líneas del pentagrama	4.1—22
4.1.2.2	Localización de fragmentos de las líneas	4.1—23
4.1.3	Proyecciones	4.1—24
4.1.4	Gráfico de líneas adyacentes (LAG)	4.1—25
4.1.4.1	Extracción del ruido	4.1—26
4.1.4.2	Reconocimiento de filamentos y pentagramas	4.1—26
4.1.4.3	Reconocimiento de strings de filamentos y pentagramas	4.1—27
4.1.4.4	Simplificación de la transformada de LAG	4.1—27
4.1.5	Lee Sau Dan	4.1—29
4.2	Clasificación y segmentación de símbolos	4.2—29
4.2.1	Xulio Fernández Hermida y Carlos Sánchez-Barbudo Vargas	4.2—29
4.2.2	RIEM	4.2—31
4.2.2.1	Construcción de contornos	4.2—31
4.2.2.2	Abertura de un contorno	4.2—31
4.2.2.3	Fichado de un contorno	4.2—32

4.2.2.4	Unión de dos contornos	4.2—32
4.2.2.5	División de contornos en primitivas gráficas	4.2—33
4.2.2.6	Localización de objetos	4.2—33
4.2.2.7	Pre-clasificación de objetos	4.2—34
4.2.2.8	Reconocimiento de objetos	4.2—35
4.2.2.9	Técnicas de reconocimiento utilizadas	4.2—35
4.2.3	Fujinaga	4.2—36
4.2.4	Carter	4.2—36
4.2.4.1	Formación del objeto	4.2—37
4.2.4.2	Ordenación de símbolos	4.2—37
4.2.4.3	Reconocimiento	4.2—37
4.2.5	Lee Sau Dan	4.2—39
4.3	Análisis sintáctico o semántico	4.3—40
4.3.1	Xulio Fernández Hermida y Carlos Sánchez-Barbudo Vargas	4.3—40
4.3.2	RIEM	4.3—40
4.3.2.1	Representación orientada a primitivas	4.3—40
4.3.2.2	Representación orientada a eventos	4.3—40
4.4	Salida de datos	4.4—41
4.4.1	Xulio Fernández Hermida y Carlos Sánchez-Barbudo Vargas	4.4—41
4.4.2	RIEM	4.4—41

4.4.3	Lee Sau Dan	4.4—43
5	<i>El procesamiento en el reconocimiento de música impresa</i>	4.4—44
5.1	Búsqueda de características del pentagrama	5.1—44
5.2	Localización de los pentagramas	5.2—45
5.3	Detección de la inclinación de la imagen	5.3—46
5.4	Eliminación del pentagrama	5.4—47
5.5	Definición de sistemas	5.5—51
5.6	Delimitación de compases	5.6—52
5.7	Doble barras	5.7—54
5.8	Eliminaciones de barras divisorias	5.8—54
6	<i>segmentación y Reconocimiento de símbolos</i>	5.8—56
6.1	Símbolos reconocidos por la aplicación	6.1—56
6.2	Metodología de trabajo ‘ Trabajo en Capas	6.2—56
6.3	Casos de excepción	6.3—59
6.4	Reconocimiento de las claves	6.4—59
6.5	Reconocimiento de tonalidades	6.5—60
6.6	Localización y reconocimiento de notas con cabeza negra	6.6—62
6.6.1	Localización de notas con cabeza negra	6.6—62
6.6.2	Reconocimiento de notas de cabeza negra	6.6—63
6.7	Localización y Reconocimiento de plicas	6.7—64

6.7.1	Localización de corchetes y barras de grupo	6.7—65
6.8	Localización y Reconocimiento de alteraciones	6.8—66
6.9	Localización y Reconocimiento de puntillos	6.9—68
6.10	Localización y Reconocimiento de blancas	6.10—69
6.10.1	Problemas y resolución de los mismos	6.10—70
6.11	Localización y reconocimiento de silencios	6.11—71
6.12	Localización y reconocimiento de redondas	6.12—72
7	<i>Archivo de salida</i>	6.12—73
7.1	Especificación de los archivos LEM (Lista de Eventos Musicales)	7.1—73
7.1.1	Encabezado del archivo	7.1—73
7.1.2	Pistas	7.1—74
7.1.3	Fin de archivo	7.1—77
7.2	EJEMPLO	7.2—78
8	<i>Técnicas y algoritmos aplicados</i>	7.2—80
8.1	Algoritmo de Bresenham (Algorithm for Computer Control of a Plotter Digital – IBM Journal 1965)	8.1—80
8.2	Thining	8.2—81
8.3	Erosión	8.3—82
9	<i>desarrollo del programa “partituras”</i>	8.3—83
9.1	Descripción del trabajo	9.1—83
9.1.1	Requerimientos Mínimos	9.1—83

9.1.2	Consideraciones generales	9.1—83
9.2	Entorno visual	9.2—83
9.3	Conocimiento a priori	9.3—87
9.4	Descripción del programa	9.4—88
9.4.1	Ingreso de la imagen	9.4—88
9.4.1.1	Imagen tomada del escáner	9.4—88
9.4.1.2	Imagen tomada de un archivo	9.4—88
9.4.2	Procesamiento	9.4—89
9.4.2.1	Extracción de datos de la imagen y alojamiento de la imagen en memoria	9.4—89
9.4.2.2	Cálculo del espaciado interlineal y el grosor de las líneas	9.4—89
9.4.2.3	Búsqueda de la cantidad y posición de los pentagramas	9.4—89
9.4.3	Búsqueda del formato de la partitura	9.4—89
9.4.4	Verificación de la inclinación de los pentagramas	9.4—90
9.4.5	Eliminación de los pentagramas	9.4—90
9.4.6	División por compases	9.4—90
9.4.6.1	Compases simples	9.4—90
9.4.6.2	Compases de sistemas	9.4—91
9.4.7	Eliminación de barras divisorias	9.4—91
9.4.8	Reconocimiento de símbolos	9.4—92
9.4.8.1	Busca la clave	9.4—93

9.4.8.2	Reconoce la tonalidad	9.4—93
9.4.8.3	Reconoce las figuras musicales	9.4—94
1.1.1.1	Figuras negras	9.4—95
1.1.1.2	Figuras Blancas	9.4—96
1.1.1.3	Silencios	9.4—96
1.1.1.4	Redonda	9.4—97
9.4.9	Escritura del archivo de Salida	9.4—97
9.4.10	Manejo de errores	9.4—97
9.5	Cuadro comparativo	9.5—98
9.6	Diagrama de flujo	9.6—100
10	<i>Resultados obtenidos</i>	9.6—105
11	<i>Líneas futuras</i>	9.6—106
12	<i>conclusiones</i>	9.6—107
13	<i>bibliografía</i>	9.6—108