

**DESARROLLO DE UN MÉTODO DIDÁCTICO PARA LEER PARTITURAS  
MUSICALES UTILIZANDO EL PROCESAMIENTO DIGITAL DE IMÁGENES  
APLICADO AL APRENDIZAJE DE LA MÚSICA**

**CARLOS HERNANDO PORTILLA ZAMBRANO**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE CIENCIAS EXACTAS NATURALES Y DE LA EDUCACIÓN  
INGENIERÍA FÍSICA  
POPAYÁN  
2011**

**DESARROLLO DE UN MÉTODO DIDÁCTICO PARA LEER PARTITURAS  
MUSICALES UTILIZANDO EL PROCESAMIENTO DIGITAL DE IMÁGENES  
APLICADO AL APRENDIZAJE DE LA MÚSICA**

**CARLOS HERNANDO PORTILLA ZAMBRANO**

**Trabajo de grado para optar el título de Ingeniero Físico**

**Director  
EDGAR LEONARIO PENCUE FIERRO  
Ingeniero Físico**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE CIENCIAS EXACTAS NATURALES Y DE LA EDUCACIÓN  
INGENIERÍA FÍSICA  
POPAYÁN  
2011**

**Nota de aceptación**

---

---

---

---

**Director** \_\_\_\_\_

**Ing. Edgar Leonairo Pencue Fierro**

**Jurado** \_\_\_\_\_

**Profesor Luis Fernando Echeverry**

**Jurado** \_\_\_\_\_

**Profesor René Ordoñez**

**Popayán, 4 de Febrero de 2011**

## CONTENIDO

	Pág.
<b>RESUMEN</b>	<b>17</b>
<b>INTRODUCCIÓN</b>	<b>18</b>
• <b>Introducción</b>	<b>18</b>
• <b>Estructura del informe de investigación</b>	<b>18</b>
<b>CAPÍTULO 1. ACERCAMIENTO A LA MÚSICA Y SU RELACIÓN CON LA TECNOLOGÍA</b>	<b>20</b>
<b>1.1. La música</b>	<b>20</b>
<b>1.2. La importancia de la música</b>	<b>20</b>
<b>1.3. La evolución de la representación musical</b>	<b>21</b>
<b>1.4. La tecnología musical</b>	<b>22</b>
<b>CAPÍTULO 2. FUNDAMENTOS DE TEORÍA MUSICAL</b>	<b>24</b>
<b>2.1. Sonido</b>	<b>24</b>
<b>2.2. Dimensiones del sonido</b>	<b>24</b>
<b>2.2.1. Frecuencia</b>	<b>24</b>
<b>2.2.2. Duración</b>	<b>25</b>
<b>2.2.3. Intensidad</b>	<b>25</b>
<b>2.2.4. Timbre</b>	<b>25</b>

<b>2.3. Ritmo musical</b>	<b>25</b>
<b>2.3.1. Pulso</b>	<b>26</b>
<b>2.3.2. Acento</b>	<b>26</b>
<b>2.3.3. Tempo</b>	<b>26</b>
<b>2.3.4. Representación del ritmo en el lenguaje musical</b>	
<b>Occidental</b>	<b>27</b>
<b>2.4. Las figuras</b>	<b>27</b>
<b>2.4.1. Proporción entre las figuras</b>	<b>28</b>
<b>2.5. Los Silencios</b>	<b>29</b>
<b>2.6. El Compás</b>	<b>30</b>
<b>2.6.1. El indicador de compás</b>	<b>30</b>
<b>2.7. Signos que modifican la duración</b>	<b>31</b>
<b>2.7.1. La Ligadura</b>	<b>31</b>
<b>2.7.2. El Puntillo</b>	<b>31</b>
<b>2.8. Melodía</b>	<b>32</b>
<b>2.9. Elementos básicos de la notación musical</b>	<b>32</b>
<b>2.9.1. Las siete notas de la escala natural</b>	<b>32</b>
<b>2.10. El Pentagrama</b>	<b>34</b>
<b>2.11. Las Claves</b>	<b>35</b>
<b>2.12. Las Alteraciones</b>	<b>35</b>
<b>2.12.1. El Sostenido</b>	<b>35</b>
<b>2.12.2. El Bemol</b>	<b>36</b>
<b>2.12.3. El Becuadro</b>	<b>36</b>
<b>2.13. La Armadura</b>	<b>36</b>

<b>CAPITULO 3. COMPONENTES DE UN SISTEMA DE VISION POR COMPUTADOR</b>	<b>38</b>
<b>3.1. Adquisición de imágenes</b>	<b>39</b>
3.1.1. Escáner óptico	39
3.1.2. Imagen Digital	40
3.1.2.1. Imagen Binaria	41
3.1.2.2. Imagen de Intensidad	41
<b>3.2. Preprocesamiento de Imágenes</b>	<b>42</b>
3.2.1. Convolución	42
3.2.2. Suavizado	43
3.2.3. Filtrado espacial de paso bajo	43
3.2.4. Filtros Morfológicos	44
3.2.4.1. Dilatación	44
3.2.4.2. Erosión	44
3.2.4.3. La apertura	45
3.2.4.4. El cierre	45
3.2.5. Suavizado binario	45
3.2.6. Binarización	45
3.2.7. Rotación	46
<b>3.3. Segmentación de Imágenes</b>	<b>47</b>
3.3.1. Codificación de segmentos	47
3.3.2. Componentes conectadas	47
3.3.3. Histograma	48
3.3.3.1. Histograma de proyección vertical	48
3.3.3.2. Histograma de proyección horizontal	48

3.3.4. Etiquetado	50
3.3.4.1. Objeto	50
3.4. Características de los objetos	51
3.4.1. Altura	51
3.4.2. Ancho	52
3.4.3. Tamaño	52
3.5. Reconocimiento de objetos	53
3.5.1. Métodos estructurales	53
<b>CAPÍTULO 4. SISTEMAS OPTICAL MUSIC RECOGNITION (OMR)</b>	<b>54</b>
4.1. Estructura general de un sistema OMR	55
4.1.1. Preprocesamiento	55
4.1.2. Reconocimiento de símbolos musicales	56
4.1.2.1. Detección y eliminación de las líneas del pentagrama	56
4.1.2.2. Segmentación de los símbolos	56
4.1.2.3. Clasificación de los símbolos	57
4.1.3. Gramática	58
4.1.4. Representación final	58
4.2. Investigaciones previas sobre los sistemas OMR	59
4.2.1. Preprocesamiento	59
4.2.2. Líneas del pentagrama	60
4.2.3. Clasificación y Reconocimiento	61
4.3. Distintos enfoques de los sistemas OMR	63

<b>4.4. Trabajos relacionados con los sistemas OMR</b>	<b>67</b>
4.4.1. McGee y Merkely (1991)	67
4.4.2. Rebelo (2008)	67
4.4.3. Tardón et al. (2010)	68
<b>4.5. Software disponible</b>	<b>69</b>
4.5.1. Software Comercial	70
4.5.1.1. SmartScore Pro 5	71
4.5.1.2. PDFtoMusic	71
4.5.1.3. Vivaldi-Scan	71
4.5.2. Software Libre	72
<b>4.6. Aplicaciones</b>	<b>72</b>
<b>CAPÍTULO 5. SISTEMA OMR IMPLEMENTADO</b>	<b>74</b>
5.1. Adquisición de las imágenes	74
5.1.1. Imágenes adquiridas	74
5.1.2. Características de las partituras y de la notación musical reconocidas	75
5.2. Preprocesamiento	77
5.2.1. Filtro paso bajo	77
5.2.2. Binarización	80
5.2.3. Filtro morfológico	81
5.2.4. Suavizado binario	81
5.3. Detección y eliminación de pentagramas	82
5.3.1. Detección de las líneas	82
5.3.2. Eliminación de las líneas	84
5.4. Segmentación y Reconocimiento	84



5.4.1. Rotación de la imagen	85
5.4.2. Selección del área de análisis	86
5.4.3. Etiquetado de los objetos	86
5.4.4. Detección y eliminación de las ligaduras	87
5.4.5. Reconocimiento de la clave	87
5.4.6. Reconocimiento de la armadura	88
5.4.7. Reconocimiento del indicador de compás	88
5.4.8. Eliminación de la clave, la armadura y el indicador de compás	88
5.4.9. Reconocimiento de las barras divisorias de compás	89
5.4.10. Segmentación y reconocimiento de notas, silencios, alteraciones y puntillos	90
5.5 Interpretación de la partitura	91
<b>CAPÍTULO 6. RESULTADOS Y ANÁLISIS DE RESULTADOS</b>	<b>93</b>
6.1. Introducción	93
6.2. Evaluación de los resultados	93
6.2.1. Preprocesamiento	93
6.2.2. Pentagramas	95
6.2.3. Área de análisis	98
6.2.4. Ligaduras	101
6.2.5. Clave	104
6.2.6. Armadura	104
6.2.7. Indicador de compás	105
6.2.8. Barras de compás	106
6.2.9. Notas, silencios, alteraciones y puntillos	107

6.2.10. Evaluación general	110
6.3. Eficiencia del sistema	110
<b>CAPÍTULO 7. CONCLUSIONES Y TRABAJOS FUTUROS</b>	<b>114</b>
7.1. Conclusiones	114
7.2. Trabajos Futuros	115
<b>REFERENCIAS</b>	<b>116</b>
<b>ANEXO</b>	
A Manual de Usuario	121

## LISTA DE FIGURAS

	<b>pág.</b>
<b>Figura 2.1</b> Frecuencia: Sonido grave y agudo	25
<b>Figura 2.2</b> Intensidad: Sonido fuerte y suave	26
<b>Figura 2.3</b> Figuras musicales	27
<b>Figura 2.4</b> Partes de una nota	28
<b>Figura 2.5</b> Representación de los grupos de corcheas y semicorcheas	28
<b>Figura 2.6</b> Proporción entre las figuras	29
<b>Figura 2.7</b> Silencios	29
<b>Figura 2.8</b> Situación del silencio de blanca en el pentagrama	29
<b>Figura 2.9</b> Acentos	30
<b>Figura 2.10</b> La barra de conclusión	30
<b>Figura 2.11</b> Indicadores de compás más usuales	31
<b>Figura 2.12</b> Indicadores del número del denominador con las figuras	31
<b>Figura 2.13</b> La Ligadura	32
<b>Figura 2.14</b> El Puntillo	32
<b>Figura 2.15</b> Las siete notas musicales	33
<b>Figura 2.16</b> Octava en el piano	33
<b>Figura 2.17</b> Escala de relaciones simples	33
<b>Figura 2.18</b> El Pentagrama	34
<b>Figura 2.19</b> Las líneas adicionales	34
<b>Figura 2.20</b> Tipos de claves más usadas	35

<b>Figura 2.21</b> Las Alteraciones	36
<b>Figura 2.22</b> Tonalidad de Mib	37
<b>Figura 3.1</b> Elementos de un sistema de visión por computador	38
<b>Figura 3.2</b> Escáner óptico	39
<b>Figura 3.3</b> Imagen digital y Píxel	40
<b>Figura 3.4</b> Imagen binaria	41
<b>Figura 3.5</b> Imagen de intensidad (256 niveles de gris)	41
<b>Figura 3.6</b> Ejemplo de suavizado de una imagen	43
<b>Figura 3.7</b> Filtros especiales de paso bajo	44
<b>Figura 3.8</b> Rotación de la imagen	46
<b>Figura 3.9</b> Concepto de vecindad	48
<b>Figura 3.10</b> Histograma de una partitura	49
<b>Figura 3.11</b> Histograma de proyección vertical	49
<b>Figura 3.12</b> Histograma de proyección horizontal	50
<b>Figura 3.13</b> Etiquetado	50
<b>Figura 3.14</b> Objeto y Fondo	51
<b>Figura 3.15</b> Altura de un objeto	51
<b>Figura 3.16</b> Ancho de un objeto	52
<b>Figura 4.1</b> Estructura general de un sistema OMR	55
<b>Figura 4.2</b> Laplaciano utilizado por Kim, Chung y Bien (1987)	60
<b>Figura 4.3</b> Cuerda de orientación $\theta$ en P (Martin y Bellisant 1991)	60
<b>Figura 4.4</b> El Wabot-2 (Matsushima et al., 1985)	62
<b>Figura 4.5</b> Diagrama de flujo del proceso (Adaptada de Rossant 2002)	65
<b>Figura 4.6</b> Estructura del sistema (Adaptada de McPherson 2003)	66

<b>Figura 4.7</b> Notación antigua (McGee y Merkely 1991)	67
<b>Figura 4.8</b> Arquitectura OMR (Adaptada de Rebelo 2008)	68
<b>Figura 4.9</b> Fragmentos de partituras en notación mensural blanca (Tardón et al., 2010)	69
<b>Figura 5.1</b> Escáner utilizado para digitalizar las partituras	76
<b>Figura 5.2</b> Ejemplos de imágenes adquiridas con diferentes calidades y tamaños	77
<b>Figura 5.3</b> Diagrama de flujo del sistema de preprocesamiento propuesto	78
<b>Figura 5.4</b> Resultado de aplicar el primer filtro	79
<b>Figura 5.5</b> Resultado de aplicar el segundo filtro	79
<b>Figura 5.6</b> Los dos tipos de filtro paso bajo usados	79
<b>Figura 5.7</b> Resultado de aplicar la binarización	80
<b>Figura 5.8</b> Histograma de la imagen resaltado el pico que determina el umbral de binarización	80
<b>Figura 5.9</b> Aplicación del filtro morfológico	81
<b>Figura 5.10</b> Aplicación del suavizado binario	81
<b>Figura 5.11</b> Extracción del pentagrama	83
<b>Figura 5.12</b> Eliminación del pentagrama	84
<b>Figura 5.13</b> Selección del área de análisis	86
<b>Figura 5.14</b> Detección y eliminación de las ligaduras	87
<b>Figura 5.15</b> Eliminación de la clave, la armadura y el indicador de compás	89
<b>Figura 5.16</b> Detección de las barras divisorias de compás	89
<b>Figura 5.17</b> Símbolos parcialmente destruidos en el proceso de eliminación de los pentagramas	91

<b>Figura 6.1</b> Partitura No 1. (a) Antes y (b) Después del preprocesamiento	93
<b>Figura 6.2</b> Partitura No 12. (a) Antes y (b) Después del preprocesamiento	94
<b>Figura 6.3</b> Partitura No 40. (a) Antes y (b) Después del preprocesamiento	94
<b>Figura 6.4</b> Determinación de los picos de los sostenidos	96
<b>Figura 6.5</b> Preprocesamiento: (a) Cabeza de nota rellena. (b) Aumento del grosor de líneas delgadas	96
<b>Figura 6.6</b> Mal procesamiento en la cabeza de la nota negra	96
<b>Figura 6.7</b> Detección de los pentagramas en la partitura No. 40	97
<b>Figura 6.8</b> Detección de los pentagramas en la partitura No 27	97
<b>Figura 6.9</b> Selección del área de análisis. Caso 1	99
<b>Figura 6.10</b> Selección del área de análisis. Caso 2	99
<b>Figura 6.11</b> Selección del área de análisis. Caso 3	100
<b>Figura 6.12</b> Detección de las ligaduras. Caso 1	102
<b>Figura 6.13</b> Detección de las ligaduras. Caso 2	102
<b>Figura 6.14</b> Detección de las ligaduras. Caso 3	103
<b>Figura 6.15</b> Error en la detección de la armadura	105
<b>Figura 6.16</b> Posición de las barras de compás dentro del pentagrama	106
<b>Figura 6.17</b> Error en la detección de las barras de compás	107
<b>Figura 6.18</b> Error de reconocimiento de la nota blanca	108
<b>Figura 6.19</b> Error en el reconocimiento de una nota afectada por un signo de expresión	109
<b>Figura A.1</b> Ícono del programa	121
<b>Figura A.2</b> Interfaz gráfica de usuario	121
<b>Figura A.3</b> Ventana de selección	122

<b>Figura A.4</b> Partitura cargada	122
<b>Figura A.5</b> Mensaje de proceso terminado	123
<b>Figura A.6</b> Resultados del reconocimiento	124
<b>Figura A.7</b> Mensaje de interpretación terminada	124
<b>Figura A.8</b> Selector de voz	125
<b>Figura A.9</b> Selector de velocidad	125

## LISTA DE TABLAS

	<b>pág.</b>
<b>Tabla 6.1</b> Resultados de la detección de pentagramas sobre 110 partituras	98
<b>Tabla 6.2</b> Resultados de la selección del área de análisis para 78 partituras	101
<b>Tabla 6.3</b> Resultados de la detección de ligaduras	103
<b>Tabla 6.4</b> Resultados de la detección de la clave	104
<b>Tabla 6.5</b> Resultados de la detección de la armadura	104
<b>Tabla 6.6</b> Resultados de la detección del indicador de compás	106
<b>Tabla 6.7</b> Resultados de la detección de las barras de compás	107
<b>Tabla 6.8</b> Resultados del reconocimiento de Notas, silencios, alteraciones y puntillos	109
<b>Tabla 6.9</b> Porcentaje de efectividad del sistema	111



## RESUMEN

En este trabajo se desarrolló un sistema OMR, de sus siglas en inglés *Optical Music Recognition*, para interpretar partituras musicales no manuscritas. Se digitalizaron varias partituras, las cuales se trataron con el algoritmo de procesamiento implementado y se reconocieron los símbolos más importantes de la partitura, posteriormente se reprodujeron haciendo uso del formato MIDI.

Para la implementación del algoritmo se utilizaron técnicas sencillas del procesamiento digital de imágenes como son los histogramas de proyección y algunas reglas básicas de la música para facilitar el proceso de segmentación y reconocimiento de símbolos.

El sistema representa una base sólida en el campo del reconocimiento de símbolos y posee la cualidad de adaptarse para tratar partituras más complejas como son las armónicas, permitiendo una posible comparación con sistemas mejor elaborados como son los programas OMR comerciales especialistas en partituras no manuscritas.

## ABSTRACT

In this paper an OMR system for interpreting no handwritten music scores is developed. Several scores were digitized and treated with the processing algorithm implemented and then the most important symbols recognized. Finally the results were reproduced using the MIDI format.

For the implementation of the algorithm simple techniques of digital image processing are used such as projection histograms, and some basic rules of music to facilitate the process of segmentation and recognition of symbols.

The system represents a solid foundation in the symbols recognition field and has the quality to adapt to deal with complex scores such as harmonics, allowing for a possible comparison with more elaborated system such as commercial OMR programs no handwritten sheet music specialist.

- **Introducción**

Existe un problema muy común en las personas que están aprendiendo música, y tiene que ver con la lectura de partituras. La cantidad de símbolos y reglas o teoría musical, que se deben conocer es bastante extensa, cuestión por la cual el aprendiz muchas veces termina aburrido y en algunos casos frustrado por el largo tiempo que le acarrea llegar a leer correctamente una partitura, es decir, interpretar la melodía escrita en ella.

Para solucionar este problema se han creado métodos que facilitan parcialmente la tarea, entre ellos se pueden citar el sistema de notación llamado tablatura o editores de partituras como Finale [1] o Sibelius [2], y programas de escaneo e interpretación de partituras como SmartScore [3] o PDFtoMusic [4].

La importancia de la partitura, tanto a nivel cultural como a nivel social, se hace evidente en su capacidad de conservar en un papel una gran obra a través del tiempo. Por lo cual, aprender a leerla es muy necesario para poder reproducirla posteriormente. Además, hay que tener en cuenta que hay personas que quieren interpretar las canciones, obras o ejercicios escritos en las partituras como un pasatiempo.

El propósito de este trabajo es desarrollar un sistema de visión por computador para la lectura de partituras y utilizarlo como método didáctico en el aprendizaje de la música. Para llevarlo a cabo se usaron técnicas de procesamiento de imágenes especializadas para este tipo de problemas, como el reconocimiento de patrones y formas, y se utilizaron las reglas musicales para obtener un eficiente reconocimiento de los símbolos de la partitura. Además, se empleó un archivo MIDI para la representación final de la partitura.

- **Estructura del informe de investigación**

El documento se encuentra estructurado de la siguiente manera:

En el capítulo 1, se da la definición de música, la importancia que ha tenido en nuestra vida diaria, las distintas formas de su representación y finalmente su estrecha relación con la tecnología.

En el capítulo 2, se da una breve descripción de los conceptos básicos de teoría musical necesarios para el entendimiento de las partituras. Se define el sonido y sus dimensiones, el concepto de ritmo, compás y melodía, y las figuras musicales para representar las dimensiones del sonido: tono y duración.

En el capítulo 3, se describen los componentes de un sistema de visión por computador. La adquisición de imágenes, el preprocesamiento, la segmentación de imágenes, las características y el reconocimiento de los objetos.

En el capítulo 4, se trata un sistema de interpretación de partituras basado en visión por computador o comúnmente llamado sistema OMR. Se presenta y se describe la estructura general del sistema, se describen las investigaciones previas sobre el tema, se presentan los distintos enfoques de los sistemas OMR, se habla del software OMR existente y por último se tratan los campos de aplicación.

En el capítulo 5, se muestran los módulos del sistema OMR implementado y las distintas técnicas usadas.

En el capítulo 6, se muestran los resultados obtenidos con las partituras usadas y un análisis de ellos.

En el capítulo 7, se presentan las conclusiones finales del trabajo y los trabajos futuros.

Por último, se presenta el manual de usuario, necesario para manejar correctamente la aplicación creada.

## **CAPÍTULO 1. ACERCAMIENTO A LA MÚSICA Y SU RELACIÓN CON LA TECNOLOGÍA**

### **1.1. La música**

La música (del griego: μουσική [τέχνη] - *mousikē [téchnē]*, "el arte de las musas") es, según la definición tradicional del término, el arte de organizar sensible y lógicamente una combinación coherente de sonidos y silencios utilizando los principios fundamentales de la melodía, la armonía y el ritmo, mediante la intervención de complejos procesos psico-anímicos. La música, como toda manifestación artística, es un hecho cultural. El fin de este arte es suscitar una experiencia estética en el oyente, y expresar sentimientos, circunstancias, pensamientos o ideas [5].

### **1.2. La importancia de la música**

La música ha sido utilizada a través del tiempo para diferentes actividades de la vida, como eventos religiosos, sociales, militares, de diversión, de trabajo, entre otros, determinando el desarrollo y el estado de los pueblos. Además, la música ha jugado un papel muy importante en el desarrollo de las personas y en el aprendizaje de otras disciplinas. En la Grecia clásica la música era materia fundamental, junto con las matemáticas, en la educación de los jóvenes. Actualmente, ha aparecido una disciplina de salud, llamada musicoterapia, en donde un individuo o grupo de individuos se someten a un proceso, que utiliza los elementos de la música, sonido, ritmo, melodía y armonía, para satisfacer las necesidades físicas, emocionales, mentales, sociales y cognitivas. La música está siendo introducida en la educación de los niños en edades preescolares, debido a la importancia que representa en su desarrollo intelectual, auditivo, sensorial, del habla, y motriz.

En fin, la música está presente en casi todos los aspectos de nuestra vida. Por lo tanto un estudio en este campo es muy importante teniendo en cuenta todos los beneficios que ésta posee. Además, el aprender a cantar o ejecutar un instrumento es algo que llevamos dentro desde que nacemos, ya sea con las canciones infantiles que nos cantaban nuestros padres o con las que nos enseñaban en la escuela, los diferentes grupos musicales que intentábamos imitar. En cualquier momento podemos decidir aprender música, a leer partituras y mejor aún a ejecutar un instrumento.

### **1.3. La evolución de la representación musical**

Debido a la importancia que la música tuvo y tiene tanto a nivel social como cultural, fue necesario buscar una manera de transmitirla tanto geográfica como temporalmente, para permitir que las personas de los distintos pueblos compartan sus composiciones, y que además se puedan conservar a través del tiempo.

De la misma manera que el hombre inventó la escritura para poder expresar a otras culturas sus conocimientos y sus sentimientos, desarrolló también una forma de escritura para poder conservar la música. Así, nace la idea de escritura o notación musical. La notación musical es el nombre genérico dado a cualquier sistema de escritura utilizado para representar gráficamente una obra musical, permitiendo a un intérprete que la ejecute de la manera deseada por el compositor. Hay tantas notaciones para expresar la música como culturas existen, pero una de las más difundidas y bien organizadas, es la notación musical occidental.

Los sistemas de notación musical existen desde hace miles de años. Se han encontrado pruebas arqueológicas de escritura musical practicada en Egipto y Mesopotamia hacia el III milenio a. C. El sistema moderno de notación musical tiene sus orígenes en las neumas (del latín: curvado), símbolos que representaban las notas musicales en piezas vocales del canto gregoriano, hacia el siglo VIII.

Actualmente se pueden conseguir por red un sinnúmero de composiciones de autores clásicos y contemporáneos en formato de partitura que utilizan el sistema de notación occidental. El deseo de saber “cómo suena” una partitura nos lleva a la necesidad de conocer el significado de esos raros símbolos que expresan las características del sonido: tono, duración, intensidad, etc. Realmente es difícil conocer a simple vista cómo va a sonar dicha composición y es necesario ensayar la partitura por un periodo de tiempo considerable hasta dominarla completamente.

Con la evolución de la tecnología, y por supuesto con la aparición del computador, era de esperar que se desarrollaran nuevas formas de expresar la música, entonces ahora podemos grabar la música, crear una fiel copia de nuestras interpretaciones en una pequeña herramienta llamada CD y compartirlas y/o reproducirlas cuantas veces lo necesitemos. Además, podemos representar la música digitalmente y de ahí la variedad de opciones que devienen, podemos guardar la música, compartirla por red, manipularla con programas adecuados, como por ejemplo Cubase [6], una serie de aplicaciones para editar audio digital. También podemos representar las composiciones con una notación más o menos estándar utilizando los editores de partituras, tales como Finale [1] y Sibelius [2].

#### 1.4. La tecnología musical

Como se vio en las secciones anteriores de este capítulo, la importancia de la música y su más o menos compleja notación, requieren de la participación de la actual tecnología para suplir las dificultades propias de este campo y el posible avance de las formas de representación existentes y el desarrollo de otras.

La tecnología ha surgido con el fin de hacer más fáciles ciertas tareas del hombre, suplir sus necesidades y utilizar al máximo los recursos. Como era de esperar la tecnología también se ha aplicado al campo de la música, así por ejemplo, aspectos como el desarrollo, la reproducción, la comercialización, la enseñanza y aprendizaje, y la divulgación de la música se han visto favorecidos, gracias a la aparición del sonido digital y del logro de poder grabar el sonido y manipularlo por medios externos, ya sean físicos (electrónicos) o virtuales (software).

Realmente, la relación entre la música y la tecnología comienza cuando el hombre en su afán por crear más sonidos y nuevas formas de expresarse, diseña y construye instrumentos externos a él, pasando de utilizar solo la voz como medio de expresión a desarrollar destrezas en el uso de instrumentos. Como era de esperarse con el advenimiento de la electrónica y de su aplicación al campo de la música surge otra revolución, aparecen nuevos instrumentos que enriquecen la gama de sonidos existentes como por ejemplo el sintetizador. El estado actual de la tecnología permite grabar y digitalizar la música, más aún, con los recursos hardware y software existentes, especializados en la manipulación, el transporte, el almacenamiento y la reproducción de la información digital, se pueden obtener una gran cantidad de beneficios, como por ejemplo la posibilidad de compartir la música, y de generar e inventar nuevos sonidos y nuevos instrumentos.

No debemos olvidar que el aprendizaje y la enseñanza de la música se han visto enormemente afectados con el desarrollo de la tecnología, ya que ahora es más fácil aprender a leer partituras e incluso a componer música. Al respecto, existen aplicaciones, como Fruity Loops [7], en los que el usuario puede crear sus propias composiciones de una manera muy fácil, incluso si no posee conocimiento musical alguno. Por otro lado, están los sistemas de reconocimiento de partituras, los cuales a través de partituras digitales en formato de imagen obtienen una fiel representación en sonidos MIDI de la composición. Con respecto a este último se puede decir que es una herramienta muy valiosa tanto para aprender como para enseñar música ya que con unos conocimientos básicos de notación musical el aprendiz es capaz de seguir la partitura a medida que la va escuchando, logrando de esta manera reforzar aspectos tales como la afinación y el solfeo, término empleado para referirse a la acción de ejecutar una partitura haciendo uso de la voz.

Como conclusión se puede decir que, siendo conscientes de la estrecha relación entre música y tecnología, no se debe ser indiferente a las ventajas que de estas herramientas tecnológicas podemos conseguir. Por un lado están las personas aficionadas a la música, las cuales pueden hacer más fácil su aprendizaje en la lectura de partituras y por otro lado están los músicos profesionales contemporáneos, los cuales podrían cambiar su perspectiva y aplicar dentro de su trabajo las posibilidades que la tecnología musical les ofrece.

## **CAPÍTULO 2. FUNDAMENTOS DE TEORÍA MUSICAL**

La teoría musical es un campo de estudio que tiene por objeto la investigación de los diversos elementos de la música, entre ellos el desarrollo y la metodología para analizar, escuchar y comprender música. En este apartado se dan los conceptos básicos de la teoría musical necesarios para entender los aspectos del sonido y para aprender a leer una partitura melódica, es decir la que contiene solamente una voz y no acordes o armonía, aunque no se descarte el hecho de que con estos conocimientos se puede ampliar a la comprensión de dichas partituras.

### **2.1. Sonido**

Es la sensación producida en el órgano del oído por el movimiento vibratorio de los cuerpos, transmitido por un medio elástico como el aire (DRAE). Por el contrario la ausencia de sonido o de ruido se denomina silencio.

### **2.2. Dimensiones del sonido**

Son aquellas propiedades que se pueden medir en el sonido, por ejemplo, la frecuencia, la duración, la intensidad y el timbre.

#### **2.2.1. Frecuencia**

La frecuencia (Figura 2.1) es la rapidez con la que vibra un objeto sonoro. La mayor o menor frecuencia es percibida por nuestro cerebro como altura, diferenciando el término altura con el término volumen alto de la música.

Un objeto que vibra rápido produce un sonido agudo, mientras que un objeto que vibra lento produce un sonido grave.



### 2.2.2. Duración

La duración es el tiempo que permanece oyéndose un sonido. Si el sonido dura mucho, decimos que es largo, y si dura poco, corto. En la música se suelen manejar sonidos con duraciones bastantes cortas.

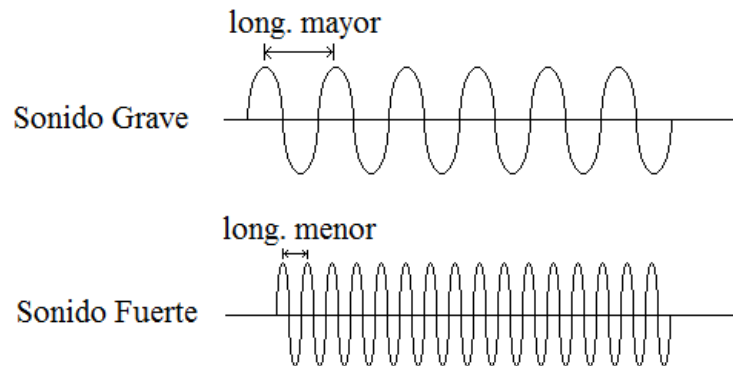


Figura 2.1 Frecuencia: Sonido grave y agudo

### 2.2.3. Intensidad

La intensidad (Figura 2.2) es la mayor fuerza con la que se produce un sonido. Se le suele llamar volumen. Un sonido de gran intensidad o de amplitud mayor es fuerte, fuerte (*f*) en la terminología musical y uno de poca intensidad o de amplitud menor es suave, piano (*p*) en la terminología musical.

### 2.2.4. Timbre

El timbre es aquella característica de los sonidos que nos permite distinguir su procedencia de diferentes instrumentos, incluso cuando tienen la misma frecuencia, la misma duración y la misma intensidad. El timbre depende de la cantidad de armónicos que tenga un sonido y de la intensidad de cada uno de ellos.

## 2.3. Ritmo musical

El ritmo musical es la ordenación de los sonidos o ruidos y los silencios en el tiempo, siguiendo la periodicidad de un patrón.

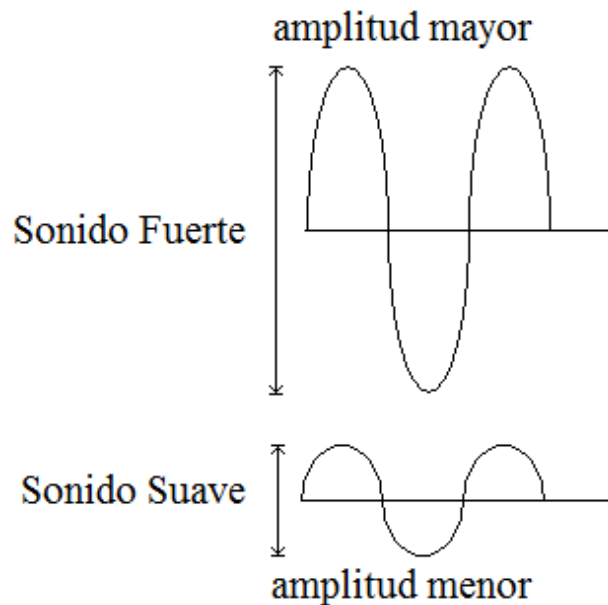


Figura 2.2 Intensidad: Sonido fuerte y suave

### 2.3.1. Pulso

El pulso es una medida constante de tiempo, son latidos, movimientos, empujes o cualquier otro estímulo, que se repite a lo largo del tiempo de forma regular, es decir, separado cada uno del siguiente por el mismo periodo de tiempo. El pulso es, por tanto, la forma más elemental de ritmo.

### 2.3.2. Acento

Se denomina acento a los pulsos fuertes, es decir a los pulsos que sentimos con mayor intensidad. Si el acento ocurre cada dos pulsos, denominamos a esa acentuación binaria; si se produce cada tres pulsos, la acentuación será ternaria, y si el acento ocurre cada cuatro pulsos, cuaternaria.

### 2.3.3. Tempo

El tempo de una obra musical es la velocidad a la que se interpreta dicha música. Para conocer la velocidad de los pulsos que sostienen el ritmo de la música se utilizan los indicadores de movimiento o los indicadores metronómicos.

Los indicadores de movimiento son términos italianos que indican más o menos la velocidad de los pulsos. Los indicadores metronómicos hacen uso de un aparato llamado metrónomo, el cual va marcando el tempo con mucha precisión.

### 2.3.4. Representación del ritmo en el lenguaje musical occidental

El lenguaje occidental es el que se desarrolló en Europa, y cuyo origen está en la Grecia clásica. *“La importancia del lenguaje musical, especialmente el occidental, que es el más difundido y que más ventajas tiene respecto a otros sistemas de escritura, radica en la cantidad de datos concretos que ofrece”* [8].

### 2.4. Las figuras

A los signos que representan los sonidos se les denomina figuras musicales, las cuales permiten representar dos de las cuatro dimensiones del sonido, la altura y su duración. Una nota es básicamente una pequeña circunferencia u óvalo. La representación de su altura depende de donde se la sitúe en el pentagrama que sirve para escribir la música. En cambio, para la representación de su duración dependerá de lo que se le haga o añada a esa pequeña circunferencia u óvalo.

La duración de una nota se representa por su figura (Figura 2.3), es decir, por lo que se le aplica a su forma elemental de pequeña circunferencia u óvalo.





Nombre	Redonda	Blanca	Negra	Corchea	Semicorchea
Figura					

Figura 2.3 Figuras musicales

La circunferencia u óvalo se llama cabeza; la línea que parte de ella, plica; y la prolongación, corchete (Figura 2.4).

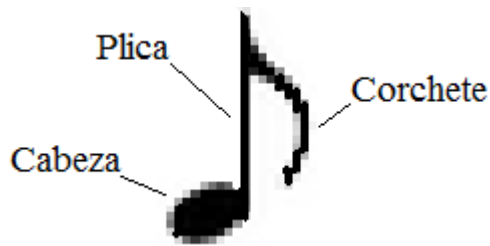


Figura 2.4 Partes de una nota

Quando se trata de escribir varias corcheas seguidas o varias semicorcheas seguidas (Figura 2.5) se puede hacer de la siguiente manera, unirlas todas mediante una o dos líneas transversales, respectivamente.



Figura 2.5 Representación de los grupos de corcheas y semicorcheas

Las figuras musicales pueden escribirse también en posición invertida. La elección de escribirlas con la plica hacia arriba o con la plica hacia abajo es una cuestión de caligrafía, y depende del lugar que ocupen las notas en el pentagrama.

### 2.4.1. Proporción entre las figuras

En nuestro sistema musical no le asignamos a cada sonido una duración física concreta. Las duraciones de los sonidos en la notación musical son proporcionales. La duración representada mediante las figuras es relativa (depende del tempo o velocidad de los pulsos) y proporcional (las figuras guardan una proporción entre sí).

Cada figura dura el doble de la siguiente y la mitad de la anterior (Figura 2.6).

Con las cinco figuras musicales (redonda, blanca, negra, corchea y semicorchea) se puede interpretar casi toda la música.

## 2.5. Los Silencios

Los silencios (Figura 2.7) son necesarios para recrear la melodía y permiten además, que el intérprete, pueda respirar cómodamente mientras interpreta la obra. Las duraciones de los silencios van a ser exactamente las mismas que las duraciones de las figuras, y van a tomar prestado el nombre de la figura de duración. El silencio de blanca se sitúa sobre la tercera línea del pentagrama (Figura 2.8).

Físicamente, los silencios no tienen altura, no poseen dos dimensiones como las notas, sino sólo una, duración. Sin embargo, en partituras que incluyen armonía como son las escritas para piano y guitarra, hay que tener en cuenta la posición de los silencios con el fin de determinar el sonido afectado.

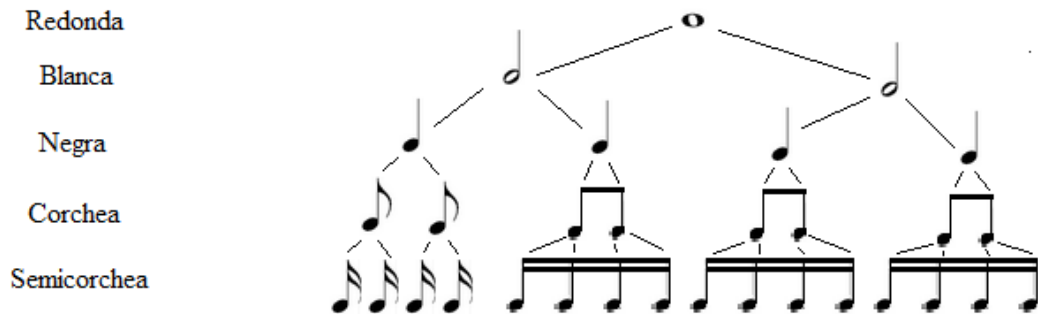


Figura 2.6 Proporción entre las figuras

Nombre	Blanca	Negra	Corchea
Silencio			

Figura 2.7 Silencios

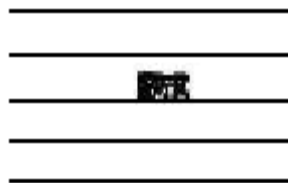


Figura 2.8 Situación del silencio de blanca en el pentagrama

## 2.6. El Compás

Los compases (Figura 2.9) son el medio que emplea el lenguaje musical para representar la acentuación propia de cualquier ritmo. Cada compás es una unidad, formada por dos o más pulsos, el primero de los cuales es siempre un acento.

Los compases van separados por una línea denominada barra divisoria de compás.

Acento binario	1 - 2   1 - 2   1 - 2   1 - 2   etc
Acento ternario	1 - 2 - 3   1 - 2 - 3   1 - 2 - 3   etc
Acento Cuaternario	1 - 2 - 3 - 4   1 - 2 - 3 - 4   1 - 2 - 3 - 4   etc

Figura 2.9 Acentos

La barra divisoria de compás final, es decir la que se coloca al final del último pentagrama de la partitura, se diferencia de las otras por tener dos barras, en donde la segunda de ellas más gruesa que la primera. A ésta se le llama barra de conclusión (Figura 2.10).



Figura 2.10 La barra de conclusión

### 2.6.1. El indicador de compás

El indicador de compás (Figura 2.11) consiste en una fracción cuyo numerador es el número de pulsos de cada compás, y cuyo denominador (Figura 2.12) es otro número que representa la figura que ocupa cada uno de esos pulsos.

Por lo general el compás de 4/4 se escribe como una C mayúscula.

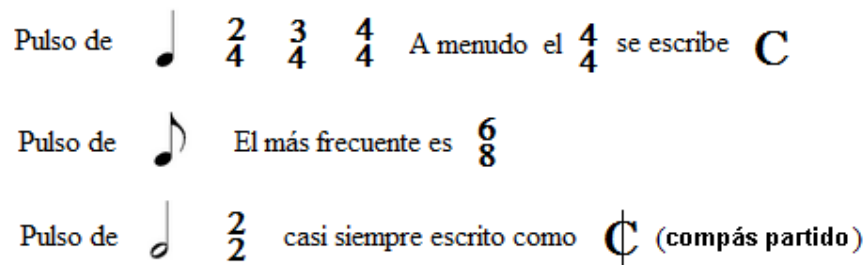


Figura 2.11 Indicadores de compás más usuales

Figura	Blanca	Negra	Corchea
Número	2	4	8

Figura 2.12 Equivalencias del número del denominador con las figuras

## 2.7. Signos que modifican la duración

Son signos auxiliares que, aplicados a las notas o silencios, permiten representar las fórmulas que se precisan para representar el ritmo en cualquier tipo de música. Entre ellos están las ligaduras y los puntillos.

### 2.7.1. La Ligadura

La ligadura (Figura 2.13) es un signo musical en forma de arco que se coloca entre dos notas. Existen dos tipos de ligaduras, las ligaduras de duración y las ligaduras de fraseo. La ligadura de duración es necesaria cuando se pretende que un sonido se prolongue más allá de una línea divisoria de compás, mientras que la ligadura de fraseo es necesaria para la interpretación.

### 2.7.2. El Puntillo

El puntillo (Figura 2.14) es un signo que debe colocarse a la derecha de una nota. Su efecto consiste en aumentar la duración de dicha nota la mitad más de lo que marca su figura.

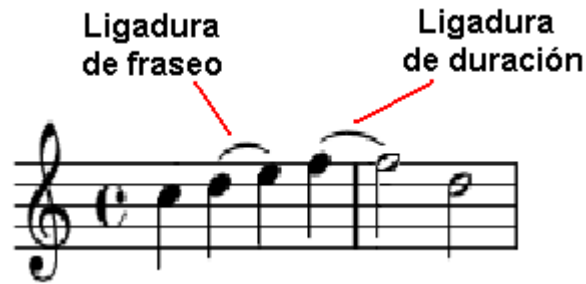


Figura 2.13 La Ligadura

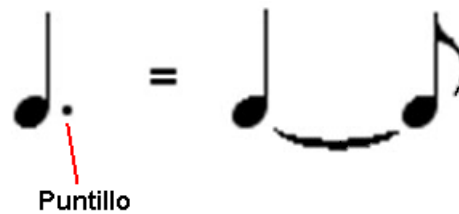


Figura 2.14 El Puntillo

## 2.8. Melodía

La melodía es una serie de silencios y sonidos de diversas alturas y, generalmente de duraciones diferentes, que se suceden en el tiempo con sentido musical.

La melodía consiste en una sucesión de sonidos, uno tras otro, y no en la simultaneidad de los mismos. La melodía lleva incluido el ritmo.

## 2.9. Elementos básicos de la notación musical

### 2.9.1. Las siete notas de la escala natural

Las 7 notas musicales (Figura 2.15) representan perfectamente una dimensión del sonido, la frecuencia. Cualquier melodía no tiene porque restringirse a las 7 alturas de sonido representadas por estas 7 notas. No todas las personas tienen una voz capaz de cantar correctamente más de 7 alturas distintas, además, casi todos los instrumentos musicales están preparados para abarcar una extensión de al menos 15 alturas distintas.



El orden de las 7 notas responde a una altura (frecuencia) creciente. A la sucesión de las 7 notas se le denomina escala natural.

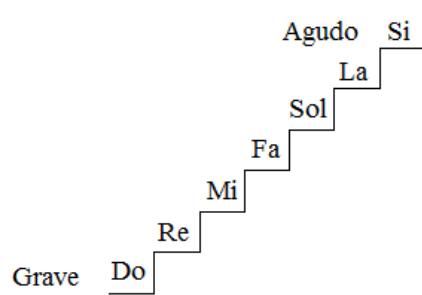


Figura 2.15 Las siete notas musicales

Cuando una melodía necesite más de 7 alturas distintas, tanto si son sonidos más graves como más agudos, se vuelve a emplear los mismos nombres, pero cambiando la altura a la octava inferior o superior, respectivamente (Figura 2.16).

Por convenio internacional, se elige 440 Hz como frecuencia del primer “La” que se encuentra a la derecha del Do<sub>3</sub>.

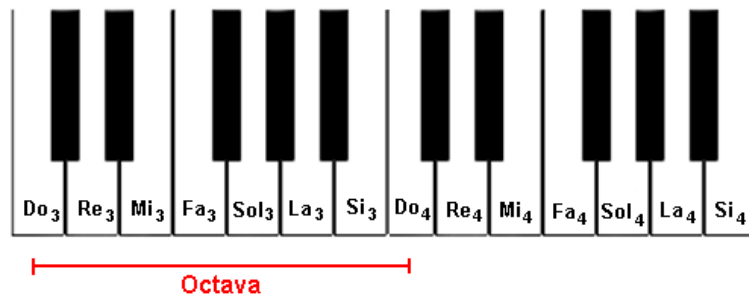


Figura 2.16 Octava en el piano

Las frecuencias de los sonidos musicales guardan una relación entre sí (Figura 2.17).

do	reb	re	mib	mi	fa	sol	lab	la	sib	si	do
1	16/15	9/8	6/5	5/4	4/3	3/2	8/5	5/3	9/5	15/8	2

Figura 2.17 Escala de relaciones simples

## 2.10. El Pentagrama

Un pentagrama (Figura 2.18) es un conjunto de 5 líneas paralelas y horizontales, con sus 4 espacios interlineales, en el que se colocan, junto con otros signos, las notas musicales para representar su altura.

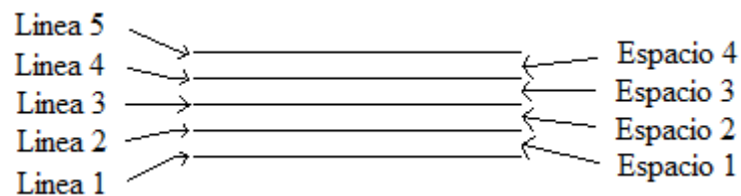


Figura 2.18 El Pentagrama

Las 5 líneas y los 4 espacios ofrecen un total de 9 lugares donde colocar las cabezas de las notas. Aunque esos 9 lugares dan mucho juego a menudo no son suficientes para colocar las cabezas de las notas. En este caso se recurre a las líneas adicionales (Figura 2.19) para situar aquellas cabezas de nota que no caben en el pentagrama.

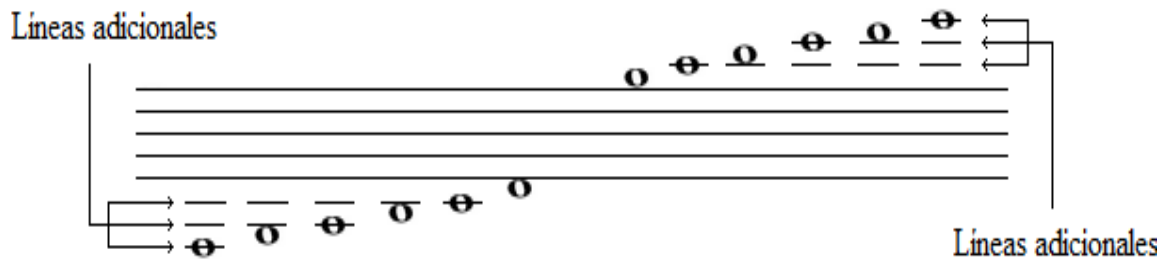


Figura 2.19 Las líneas adicionales

Cualquier nota cuya cabeza esté en la mitad superior del pentagrama, desde el tercer espacio hacia arriba, debe llevar la plica hacia abajo; pero si tiene la cabeza en la mitad inferior del pentagrama, desde el segundo espacio hacia abajo, debe llevarla hacia arriba. Si la nota tiene la cabeza justo en la tercera línea da igual si la plica va hacia arriba o hacia abajo.

## 2.11. Las Claves

Las claves (Figura 2.20) son signos musicales que nos dan el nombre que tienen las líneas y los espacios del pentagrama. La información que nos proporciona la clave es suficiente para saber, por deducción, el nombre de todas las notas posibles en el pentagrama. La clave se coloca al principio de cada pentagrama.

Existen varias claves, pero el uso ha quedado restringido a las siguientes:

- Para el registro agudo: Sol en segunda línea
- Para el registro medio: Do en tercera línea
- Para el registro grave: Fa en cuarta línea



**Clave de Sol  
en segunda línea**



**Clave de Do  
en tercera línea**



**Clave de Fa  
en cuarta línea**

Figura 2.20 Tipos de claves más usadas

## 2.12. Las Alteraciones

Son signos que alteran el sonido de la nota de referencia en un semitono ascendente o descendente (Figura 2.21). Entre ellos están: los sostenidos, los bemoles y los becuadros.

### 2.12.1. El Sostenido

Es una alteración que al aplicarlo a una nota eleva su sonido en un semitono.

### 2.12.2. El Bemol

Es una alteración que al aplicarlo a una nota disminuye su sonido en un semitono.

Las alteraciones afectan a la nota a la que preceden, y a todas las notas del mismo nombre que aparezcan en el mismo compás.

### 2.12.3. El Becuadro

Es un signo cuya función es anular el efecto de una alteración previa sobre una nota determinada.

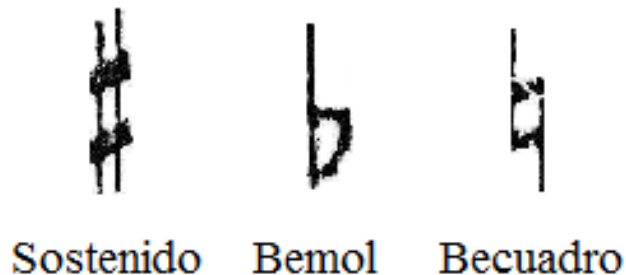


Figura 2.21 Las Alteraciones

### 2.13. La Armadura

La armadura (Figura 2.22) son símbolos que se colocan al principio de una partitura, justo a continuación de la clave, y nos dan la tonalidad. Consiste en escribir sostenidos o bemoles para que afecten a todas las notas de la partitura desde el principio hasta el fin.

Ninguna tonalidad lleva sostenidos y bemoles mezclados en su armadura y además, existe un orden de aparición de sostenido y otro, justamente el inverso, de bemoles. El orden de aparición de los sostenidos es: Fa, Do, Sol, Re, La, Mi, Si. El orden de aparición de los bemoles es: Si, Mi, La, Re, Sol, Do, Fa.

Estas características de las armaduras, o sea, la de aparecer después de la clave y la de tener un orden de aparición de las alteraciones simplifica considerablemente el proceso de reconocimiento.

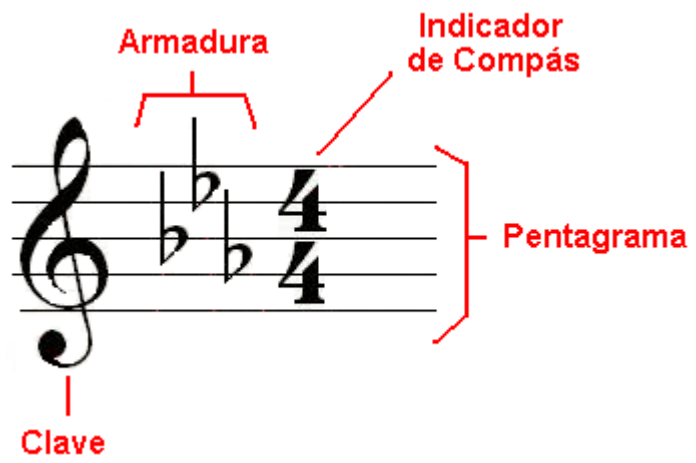


Figura 2.22 Tonalidad de Mib

### **CAPÍTULO 3. COMPONENTES DE UN SISTEMA DE VISIÓN POR COMPUTADOR**

Los esfuerzos del hombre por entender el funcionamiento del ojo y de cómo el cerebro interpreta lo que ve han llevado a la humanidad a desarrollar lo que se ha denominado un Sistema de Visión por Computador, también llamado, Visión máquina o Visión artificial. Un sistema de visión por computador al igual que un sistema de visión humano necesita para su funcionamiento de los siguientes elementos: una fuente de luz, un sensor óptico (cámara digital), un procesador de imagen, un analizador y un interpretador (computador). En la figura 3.1 se aprecian los elementos de un sistema de visión por computador, el escáner, que contiene el sensor óptico y la fuente de luz y el computador junto con la aplicación que representa el procesador, el analizador y el interpretador.



Figura 3.1 Elementos de un sistema de visión por computador

### 3.1. Adquisición de imágenes

La adquisición es el primer paso dentro de un sistema de visión por computador. La adquisición de la imagen se lleva a cabo mediante la utilización de un sensor óptico que puede ser: una cámara digital, un escáner y para ciertas aplicaciones algunos dispositivos más complejos como sensores de infrarrojo, de temperatura o de radar. El objetivo principal de la adquisición de imágenes es la obtención de una imagen digital, comprendida por el computador, a partir de una imagen abstracta del mundo exterior.

#### 3.1.1. Escáner óptico

Es un dispositivo de entrada para computador que utiliza un haz luminoso para detectar los patrones de luz y oscuridad (o los colores) de la superficie del papel, convirtiendo la imagen en señales digitales que se pueden manipular por medio de un software de tratamiento de imágenes o con reconocimiento óptico de caracteres. Un tipo de escáner (Figura 3.2) utilizado con frecuencia es el flatbed, que significa que el dispositivo de barrido se desplaza a lo largo de un documento fijo. En este tipo de escáneres, como las fotocopiadoras de oficina, los objetos se colocan sobre una superficie lisa de cristal y son barridos por un mecanismo que pasa por debajo de ellos. El escáner utilizado en este trabajo es de tipo flatbed.



Figura 3.2 Escáner óptico

### 3.1.2. Imagen Digital

Una imagen digital (Figura 3.3) es una forma de representar una imagen. La imagen digital puede ser comparada con una malla, en donde cada cuadrado representa un píxel, de sus siglas en inglés, *picture element*. Por lo tanto el píxel es la unidad mínima de una imagen digital, y una imagen puede contener millones de estos píxeles.

El tamaño de una imagen digital se mide en píxeles y da una idea de sus dimensiones físicas. A su vez el tamaño del píxel está determinado por la resolución del dispositivo. La resolución me informa sobre la cantidad de píxeles que hay en cierta unidad de longitud. La unidad de resolución más utilizada es ppp (píxeles por pulgada).

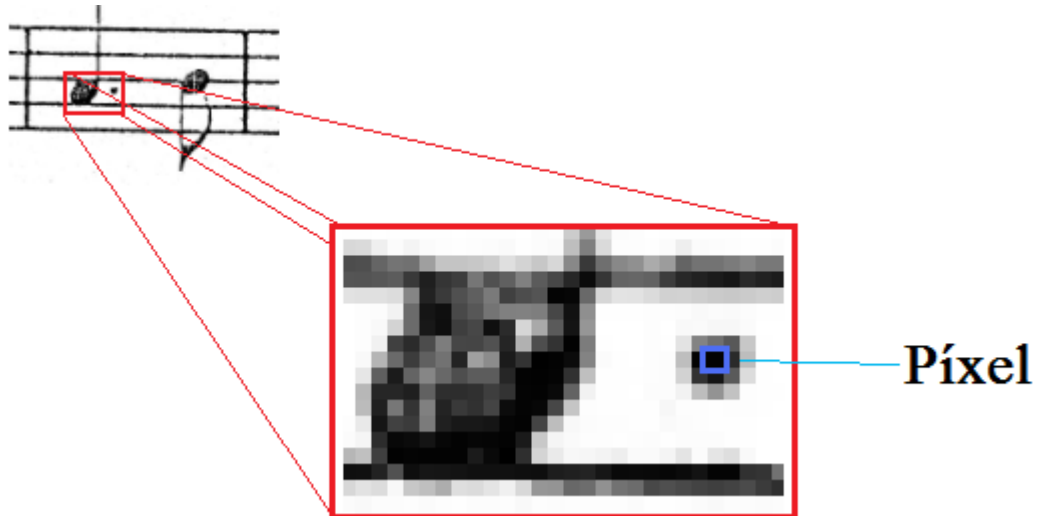


Figura 3.3 Imagen digital y Píxel

Matemáticamente, una imagen es una función bidimensional de intensidad de luz  $f(x,y)$ , donde 'x' e 'y' representan las coordenadas espaciales y el valor de  $f$  en cualquier punto cualquiera  $(x,y)$  es proporcional a la intensidad (nivel de gris) de la imagen en ese punto.

Una imagen digital es una función  $f(x,y)$  discreta tanto en las coordenadas espaciales como en la intensidad o nivel de gris. Una imagen digital es una matriz cuyos índices filas y columnas representan un punto de la imagen, y el valor del elemento correspondiente indica el nivel de gris en ese punto. Los elementos de la matriz son los píxeles.



### 3.1.2.1. Imagen binaria

Una imagen binaria (Figura 3.4) es una imagen digital que está compuesta por solo dos niveles de gris, es decir, cada elemento de la matriz o píxel puede tomar solamente uno de dos valores, generalmente los valores son '1' o '0', donde '1' representa al color blanco y '0' al color negro. Algunos programas informáticos para el tratamiento de imágenes representan el blanco de una imagen binaria con el valor 255. El resultado es una imagen en blanco y negro. Las imágenes binarias son de mucha importancia en el tratamiento de imágenes, especialmente textos o partituras, en donde, los objetos (letras o notas) en negro quedan claramente separados del fondo que usualmente es blanco.

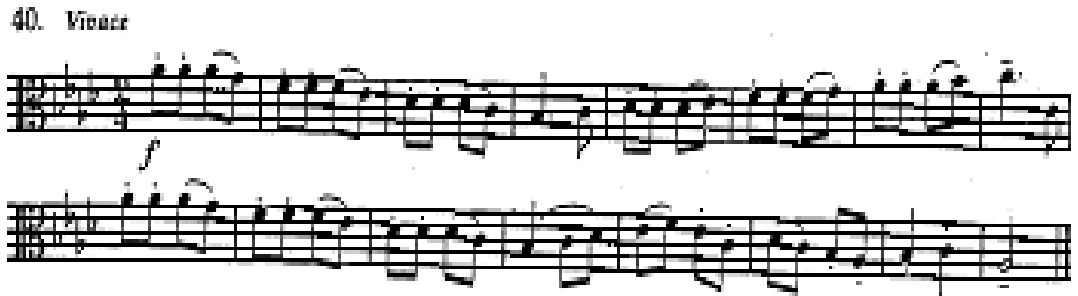


Figura 3.4 Imagen binaria

### 3.1.2.2. Imagen de intensidad

Una imagen de intensidad (Figura 3.5) es una imagen compuesta por n niveles de gris, es decir, cada elemento de la matriz o píxel puede tomar uno de n valores de nivel de gris. Generalmente n equivale a 256, en donde el cero representa al color negro y el 255 al color blanco, los valores intermedios representan tonalidades de gris que van del más oscuro (cero) hasta el más claro (blanco).



Figura 3.5 Imagen de intensidad (256 niveles de gris)

## 3.2. Preprocesamiento de Imágenes

El preprocesamiento de la imagen es el tratamiento que se realiza sobre la imagen con el fin de eliminar ciertos aspectos y resaltar otros. El preprocesamiento es necesario ya que sirve para recuperar la calidad de la imagen que generalmente es afectada por factores tanto internos como externos al sistema.

*“El preprocesamiento comprende aquellos algoritmos cuya finalidad es conseguir una mejora en la apariencia de la imagen original. Esta mejora en la apariencia consiste en resaltar determinadas características de la imagen o en eliminar aquello que las oculta. No se trata de mejorar la imagen de manera que quede lo más ideal posible sino de que pueda ser analizada de la forma más simple. Por último, es una etapa orientada hacia el problema específico que se quiera solucionar, por lo que los algoritmos que funcionen bien en unos casos no lo harán en otros” [9].*

### 3.2.1. Convolución

Una convolución es una operación matemática que se aplica entre dos funciones. En el tratamiento de imágenes, la convolución consiste en operar la función de la imagen con una función especial con el fin de obtener una nueva función con ciertas características.

Para el caso unidimensional, se define como convolución de la función  $f(x)$  respecto a la función  $h(x)$  una nueva función  $g(x)$  tal que:

$$g(x) = h(x) * f(x) = \int_{i=-\infty}^{i=\infty} f(i)h(x-i)di$$

Si se tratara del caso discreto con dos secuencias  $f(x)$  y  $h(x)$  se obtendría una secuencia de salida  $g(x)$ :

$$g(x) = h(x) * f(x) = \sum_{i=-\infty}^{i=\infty} f(i)h(x-i)$$

Para el caso de imágenes digitales se usan convoluciones bidimensionales discretas, cuya fórmula es:

$$g(x,y) = h(x,y) * f(x,y) = \sum_{i=-\infty}^{i=\infty} \sum_{j=-\infty}^{j=\infty} f(i,j)h(x-i,y-j)$$

Lo más normal es usar convoluciones de 3x3 elementos. Entonces la expresión anterior puede concretarse en:

$$g(x, y) = h(x, y) * f(x, y) = \sum_{i=0}^{i=2} \sum_{j=0}^{j=2} f(i, j)h(x - i, y - j)$$

Para un mejor tratamiento sobre el tema de convolución consultar [9], [10], [11].

### 3.2.2. Suavizado

El suavizado es un proceso que se aplica a una imagen con el fin de homogenizar las variaciones de niveles de gris localmente (Figura 3.6). Una zona de la imagen con grandes variaciones en los niveles de gris (bordes) se dice que tiene alta frecuencia, por el contrario, una zona de la imagen con pequeñas variaciones en los niveles de gris, se dice que tiene baja frecuencia. Por lo tanto, si el suavizado consiste en homogenizar las variaciones de niveles de gris, esto se consigue, aplicando un filtro pasa bajas a la imagen. *“Esta operación es útil para reducir el ruido y otros efectos no deseados que pueden estar presentes en una imagen digital como resultado del muestreo, cuantización y transmisión, o bien por perturbaciones en el sistema tales como partículas de polvo en el sistema óptico”* [10].

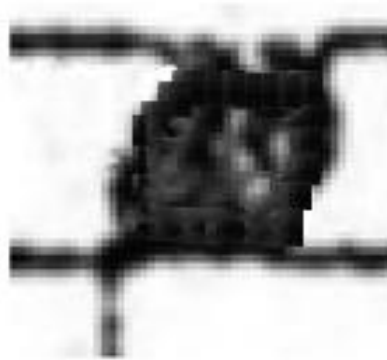


Figura 3.6 Ejemplo de suavizado de una imagen

### 3.2.3. Filtrado espacial de paso bajo

La forma de la respuesta de un impulso necesaria para implementar un filtro espacial de paso bajo (Figura 3.7) indica que el filtro ha de tener todos sus coeficientes positivos. Para un filtro espacial de 3x3, la construcción más simple

consistiría en una máscara en la que todos sus coeficientes fuesen iguales a 1. Si al resultado se multiplica por 1/9 se tiene la media de todos los píxeles, por lo que el ruido disminuye. Sin embargo, este filtro presupone que la influencia de todos los píxeles es igual. Otra consideración es que cuanto más alejado esté el píxel del central, su valor será menor.

1	2	1
2	4	2
1	2	1

1	1	1
1	1	1
1	1	1

Figura 3.7 Filtros espaciales de paso bajo

### 3.2.4. Filtros Morfológicos

Son operaciones que se realizan sobre la imagen con el fin de modificar la forma de los objetos presentes en ella. Los filtros morfológicos son necesarios cuando se quiere recuperar ciertas zonas de la imagen que se han perdido o también para eliminar las zonas de la imagen que no son de interés.

**3.2.4.1. Dilatación:** es el crecimiento progresivo de los campos (0 o 1). Un elemento del campo a crecer será convertido si posee algún vecino perteneciente al campo que se expande. En caso contrario permanecerá igual. Los elementos pertenecientes al campo a expandir evidentemente no se modifican [11].

Matemáticamente, la dilatación se expresa:

$$A \oplus B = \{x \mid [(\hat{B})_x \cap A] \subseteq A\}$$

Donde A es el conjunto de entrada y B el elemento estructural.

**3.2.4.2. Erosión:** es la degradación progresiva de uno de los campos (0 o 1). Un elemento del campo a degradar seguirá perteneciendo al mismo si está rodeado de elementos iguales a él. En caso contrario, pasará al otro campo [11].

Matemáticamente, la erosión se expresa:

$$A \ominus B = \{x | [(B)_x \cap A] \subseteq A\}$$

Donde A es el conjunto de entrada y B es el elemento estructural.

En sí, la dilatación expande la imagen y la erosión la contrae.

**3.2.4.3. La apertura:** generalmente suaviza el contorno de una imagen, rompe istmos estrechos y elimina protuberancias delgadas [9].

Matemáticamente, la apertura se expresa:

$$A \cdot B = (A \oplus B) \ominus B$$

**3.2.4.4. El cierre:** también tiende a suavizar secciones de contornos pero, en oposición a la apertura, generalmente fusiona separaciones estrechas y entrantes delgados y profundos, elimina pequeños huecos y rellena agujeros del contorno [9].

Matemáticamente, el cierre se expresa:

$$A \circ B = (A \ominus B) \oplus B$$

### 3.2.5. Suavizado binario

La idea básica del suavizado binario consiste en buscar una configuración espacial determinada y asignarle un "0" o un "1".

*"Las imágenes binarias, como es sabido están formadas por dos únicos valores lógicos "0" y "1". El ruido en este caso produce efectos tales como contornos irregulares, pequeños huecos, esquinas perdidas y puntos aislados" [10].*

### 3.2.6. Binarización

La binarización es el proceso mediante el cual una imagen con varios niveles de gris se transforma en una imagen con solo dos, generalmente '1' y '0' (blanco y negro).

Matemáticamente la binarización se consigue aplicando a la imagen la siguiente función.

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) \geq U \\ 0 & \text{si } f(x, y) < U \end{cases}$$

En donde  $g(x, y)$  representa la imagen binaria,  $f(x, y)$ , la imagen original y  $T$  el umbral.

La elección del umbral de binarización es muy importante ya que determina si la imagen binaria obtenida conserva todos los aspectos importantes de la partitura original y elimina aquellos que son considerados ruido.

### 3.2.7. Rotación

La rotación (Figura 3.8) es una transformación geométrica que se aplica a la imagen con el fin de adecuarla y facilitar ciertas operaciones propias de la segmentación, por lo tanto consiste en rotar los ejes coordenados de la imagen cierto ángulo con respecto a un punto determinado.

Matemáticamente la rotación se define de la siguiente manera:

$$\begin{bmatrix} x_f \\ y_f \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

En donde,  $x_f$  y  $y_f$  representan las coordenadas de un elemento en la imagen resultante,  $x_i$  y  $y_i$  representan las coordenadas de un elemento en la imagen original,  $x_0$  y  $y_0$  representan el punto de rotación. Y  $\theta$  es el ángulo de rotación.



Figura 3.8 Rotación de la imagen

### 3.3. Segmentación de Imágenes

*“El objetivo de la segmentación es separar los objetos presentes en la imagen del medio en el que se encuentran y distinguirlos entre sí. Para ello se utilizan algunas de las propiedades o características de la imagen. La segmentación se basa en los aspectos similares que presentan los objetos: forma, tamaño, simetría. Un elemento de mucha ayuda para la segmentación es el histograma” [10].*

#### 3.3.1. Codificación de segmentos

La codificación de segmentos o algoritmo RLE, de sus siglas en inglés, Run Length Encoding, es un algoritmo muy sencillo que se aplica generalmente para cadenas de datos. El algoritmo se encarga de contar cadenas sucesivas del mismo valor, en el caso de la imagen binarias unos (blancos) o ceros (negros), y almacena el valor y la cantidad por pares.

El siguiente ejemplo muestra cómo funciona el algoritmo:

Teniendo la siguiente cadena de valores binarios:

1,1,0,0,0,0,1,1,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,0,0,0

Se puede codificar mediante el algoritmo RLE, representado de la siguiente manera:

(1,2)(0,4)(1,2)(0,2)(1,4)(0,4)(1,6)(0,3)

Como se puede observar es un algoritmo de codificación sin pérdida ya que permite reconstruir la imagen original de forma exacta, sin variación alguna.

Este algoritmo se suele utilizar para calcular el grosor de las líneas y la longitud en una partitura.

#### 3.3.2. Componentes conectadas

Una componente conectada es una región que contiene características similares en todos sus píxeles. Para imágenes binarias, una componente conectada es una región en donde todo los píxeles tiene un mismo valor, uno (blanco) o cero (negro), y además cumplen con un criterio de conectividad o vecindad. Es decir cada uno de los píxeles debe ser vecino de al menos uno de los otros píxeles de la componente. En la Figura 3.9, el píxel 4 (centro) es vecino o está conectado con los píxeles 1, 2, 3, 4, 6, 7, 8 y 9.

1	2	3
4	5	6
7	8	9

Figura 3.9 concepto de vecindad

El píxel 4 (Figura 3.9) presenta un total de cuatro vecinos en el plano horizontal y vertical, números 2, 6 y 8. Este conjunto de píxeles se denomina vecindad de tipo 4 del píxel 4, y se representa  $N_4(4)$ .

Dos píxeles  $p$  y  $q$  presentan una conectividad-4 si  $q$  pertenece al  $N_4(p)$ .

### 3.3.3. Histograma

El histograma (Figura 3.10) de una imagen es una función que relaciona el número de píxeles con el número de niveles de gris, generalmente, 256. Gráficamente el histograma muestra cuántos píxeles hay de cada nivel de gris en la imagen.

#### 3.3.3.1. Histograma de proyección vertical

El histograma de proyección vertical (Figura 3.11) usualmente se aplica a imágenes binarias y consiste en sumar los píxeles de determinado valor de gris (cero o uno) a lo largo de la dirección vertical. El resultado es un vector fila de tamaño igual al ancho de la imagen, en donde cada elemento del vector contiene el número de píxeles con determinado valor de gris para cada columna.

#### 3.3.3.2. Histograma de proyección horizontal

El histograma de proyección horizontal (Figura 3.12) usualmente se aplica a imágenes binarias y consiste en sumar los píxeles de determinado valor de gris (cero o uno) a lo largo de la dirección horizontal. El resultado es un vector columna de tamaño igual al alto de la imagen, en donde cada elemento del vector contiene el número de píxeles con determinado valor de gris para cada fila.

Los histogramas, en todas sus posibilidades, son herramientas poderosas utilizadas para distintos tipos de objetivos dentro de los sistemas de visión por



computador. Por un lado se utilizan para analizar la dinámica de los niveles de gris de las imágenes, para calcular umbrales de binarización, para segmentar símbolos, para calcular distancias, etc.

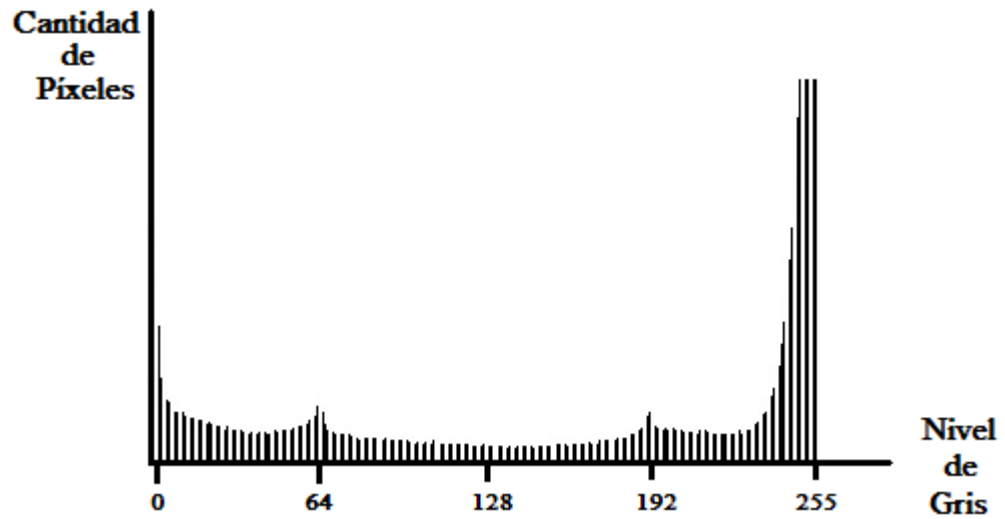


Figura 3.10 Histograma de una partitura

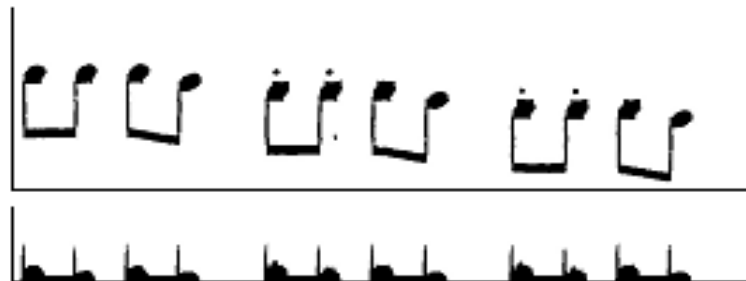


Figura 3.11 Histograma de proyección vertical



Figura 3.12 Histograma de proyección horizontal

### 3.3.4. Etiquetado

El etiquetado (Figura 3.13) es el proceso mediante el cual se asignan etiquetas a los objetos de la imagen con el fin de diferenciarlos. Generalmente el etiquetado se aplica sobre imágenes binarias.

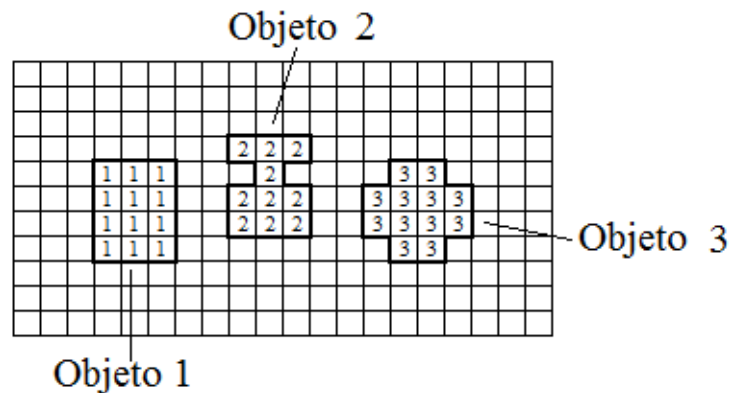


Figura 3.13 Etiquetado

#### 3.3.4.1. Objeto

Un objeto (Figura 3.14) es una zona de la imagen que presenta características similares. Por ejemplo zonas con valores de nivel de gris iguales. En una imagen binaria se diferencian los objetos (nivel de gris cero o negro) del fondo (nivel de gris uno o blanco).

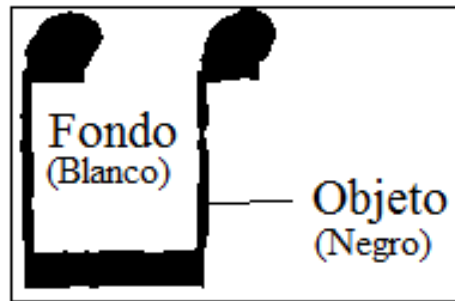


Figura 3.14 Objeto y Fondo

### 3.4. Características de los objetos

Las características de los objetos son valores numéricos que describen a los objetos. El proceso de caracterización es muy importante ya que de su buen desempeño depende el proceso de reconocimiento de los objetos.

*“Una vez que se han segmentado los objetos de la imagen, el siguiente paso es la descripción. Consiste en extraer propiedades o atributos para su uso en las aplicaciones. Básicamente, se trata de reconocer las diferentes estructuras de los objetos para su identificación de forma inequívoca” [10].* Algunas de las características comúnmente más usadas son:

#### 3.4.1. Altura

La altura (Figura 3.15) de un objeto se refiere a la longitud vertical y se mide en píxeles.

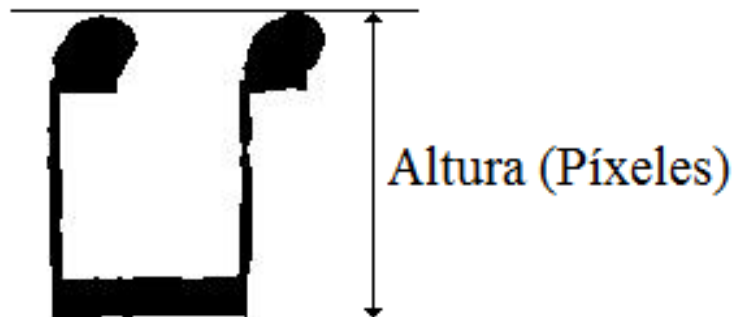


Figura 3.15 Altura de un objeto

### 3.4.2. Ancho

El ancho (Figura 3.16) de un objeto se refiere a la longitud horizontal y se mide en píxeles.

### 3.4.3. Tamaño

El tamaño es el número de píxeles que contiene un objeto.

Matemáticamente, el tamaño del objeto se expresa así:

$$A = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i, j)$$

Donde  $g(x,y)$  es la imagen binaria.

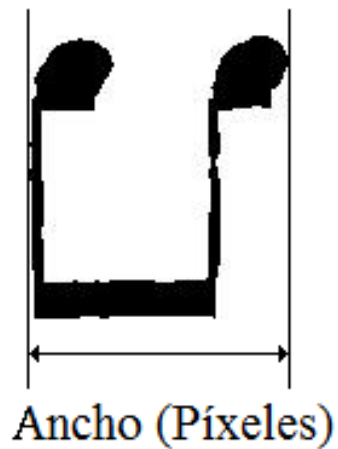


Figura 3.16 Ancho de un objeto

### **3.5. Reconocimiento de objetos**

El reconocimiento de los objetos es una de las etapas finales dentro del sistema de visión por computador y una de las más complejas.

*“Su objetivo es la clasificación de objetos en un cierto número de categorías o clases. Históricamente los dos enfoques en el reconocimiento de patrones han sido el estadístico (o teoría de la decisión) y el sintáctico (o estructural). Recientemente, el desarrollo de las redes neuronales ha proporcionado un nuevo enfoque. El aumento de la capacidad de almacenamiento ofrecida por los modernos sistemas ha proporcionado el reconocimiento basado en la apariencia” [10].*

#### **3.5.1. Métodos estructurales**

*“Los métodos estructurales realizan la discriminación de objetos utilizando las relaciones geométricas de los objetos. El reconocimiento de un patrón desconocido se lleva a cabo comparando su representación simbólica con un número de modelos de objeto predefinidos” [10].*

Esta es una de las formas más simples de realizar el proceso de reconocimiento, y es el que se usa en el sistema desarrollado. Hay que tener en cuenta que el término simple no significa ineficaz y también que la implementación de un sistema de reconocimiento de este tipo reduce notablemente el tiempo de ejecución, especialmente cuando se tratan con imágenes con gran cantidad de objetos como es el caso de las partituras musicales.

## CAPÍTULO 4. SISTEMAS OMR

Un sistema OMR es una aplicación hecha para interpretar partituras. El reconocimiento óptico de música (OMR) es una forma de análisis de documentos en el cual es particularmente importante un conocimiento a priori [12]. OMR “es el proceso de extraer automáticamente la información con sentido musical de una partitura musical impresa” [13]. Estos sistemas utilizan los conocimientos y técnicas del tratamiento de imágenes digitales para reconocer los símbolos de una partitura y posteriormente realizar alguna interpretación de ellos.

La tarea básica de un sistema OMR es tomar la partitura en un formato legible para la máquina y analizarla para identificar los símbolos musicales [14].

Un sistema de reconocimiento óptico de música (OMR) práctico y relativamente barato permitiría la simplificación de la investigación asistida por computadora en musicología y, al mismo tiempo, simplificaría muchas tareas en la interpretación musical. En un sistema OMR ideal, el usuario no necesitaría una formación especial, tampoco necesitaría elaborar complejos esquemas de codificación y además, el sistema sería capaz de adquirir toda la información necesaria para reproducir fielmente la partitura [14].

Los sistemas OMR suelen ser llamados OCR musical, de sus siglas en inglés *Optical Character Recognition*, debido a que ambos tipos de aplicaciones tienen por objetivo el reconocimiento de símbolos. Sin embargo, el reconocimiento de partituras es más complejo que otro tipo de reconocimiento de símbolos como el reconocimiento de texto, debido principalmente a la estructura bidimensional de las partituras además de otras características.

Mientras que los sistemas OCR tienen que tratar con símbolos ordenados en una sola dirección (Horizontalmente) y tienen características morfológicas parecidas, por ejemplo las letras del alfabeto latino tienen aproximadamente la misma altura y están dispuestas en orden a lo largo de la imagen, los sistemas OMR tienen que tratar con símbolos que poseen longitudes bastante diferentes y que además sus disposiciones dentro de la partitura son en dos dimensiones, una horizontal para el tiempo y otra vertical para el tono. Además, en este último los símbolos están superpuestos a los pentagramas lo que hace más difícil su segmentación.

## 4.1. Estructura general de un sistema OMR

La estructura general de un sistema OMR (Figura 4.1), comúnmente utilizada por la mayoría de los investigadores es la siguiente:

- Preprocesamiento
- Reconocimiento de símbolos musicales
- Reglas sintácticas y gráficas
- Representación final

El módulo de Reconocimiento de símbolos musicales está a su vez dividido en tres etapas: (1) detección y eliminación de las líneas del pentagrama, (2) segmentación de símbolos y (3) clasificación de símbolos.

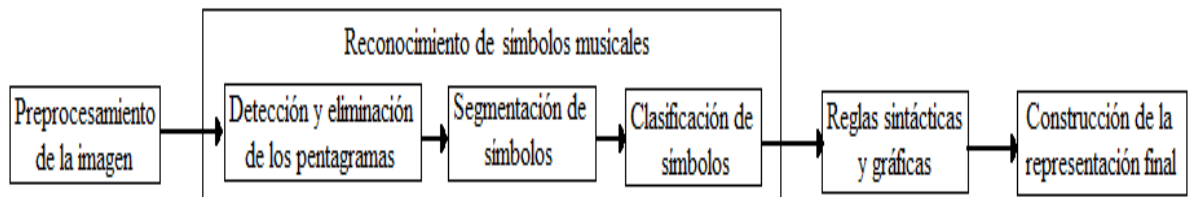


Figura 4.1 Estructura general de un sistema OMR

### 4.1.1. Preprocesamiento

El Preprocesamiento es el módulo en el cual se adecua la imagen con el fin de facilitar análisis posteriores, especialmente en la segmentación y la clasificación de los símbolos musicales. Dentro del preprocesamiento se destacan: la eliminación de ruido, la rotación de la imagen, la binarización, la eliminación de información innecesaria (por ejemplo, el título de la pieza musical), entre otros.

Algunos autores consideran la eliminación de las líneas del pentagrama como parte del preprocesamiento, ya que consideran que las líneas interfieren drásticamente en la segmentación y posterior clasificación de los símbolos musicales, es decir, ven a los pentagramas como cierto tipo de ruido. Esto no es del todo válido ya que, las líneas son muy importantes y pueden ser consideradas un símbolo más de la notación musical, su importancia radica en que son los que permiten determinar el tono de las notas.

En el preprocesamiento de partituras generalmente se emplean filtros especializados para este tipo de imágenes, como por ejemplo, el suavizado, ideal para separar y/o unir ciertas zonas de la partitura.

La binarización es un paso bien importante en el preprocesamiento porque ayuda a identificar fácilmente los objetos; sin embargo la binarización puede borrar información significativa de la partitura o romper objetos. Escoger un buen método de binarización garantiza un buen desempeño de los posteriores módulos del procesamiento. Para este propósito existen estudios que tratan varios algoritmos de binarización y presentan los resultados obtenidos en varias partituras [15].

Rotar la imagen es un paso opcional ya que hay sistemas que pueden trabajar perfectamente con inclinaciones, pero en cambio para otros trabajos, el resultado final depende de una buena rotación, por ejemplo para los que usan las proyecciones para detectar los pentagramas.

La eliminación de los símbolos sin importancia ayuda en el proceso de segmentación e identificación de símbolos, ya que como hay menos símbolos, se tiene una mayor certeza de que los objetos restantes sean símbolos musicales. En el sistema desarrollado en este trabajo no se tienen en cuenta ni los símbolos de expresión y de movimiento ni el título y número de la partitura, por lo tanto estos símbolos se consideran sin importancia.

#### **4.1.2. Reconocimiento de símbolos musicales**

El módulo de reconocimiento consta principalmente de tres etapas que son, primero, la detección y eliminación de las líneas del pentagrama, segundo, la segmentación y por último la clasificación de símbolos. A continuación se tratan cada una de estas.

##### **4.1.2.1. Detección y eliminación de las líneas del pentagrama**

La detección de las líneas del pentagrama es uno de los primeros pasos realizados en el proceso de reconocimiento. El objetivo de la eliminación de los pentagramas es quitar las líneas tanto como sea posible dejando los símbolos, superpuestos a las líneas, intactos [16]. Desde un punto de vista más general, la eliminación de pentagramas es un problema de segmentación: los objetos del fondo (las líneas del pentagrama) deben ser separados de los objetos del primer plano (los símbolos musicales) [16].

Las líneas de los pentagramas juegan un muy papel importante en la notación musical ya que definen el sistema de coordenadas vertical para los tonos y suministran una dirección horizontal para el sistema de coordenadas del tiempo. Además, la distancia entre las líneas de un pentagrama proporciona un tamaño normalizado que es útil tanto para el reconocimiento como para la interpretación de los símbolos musicales [17].



Hay dos enfoques para resolver el problema de la detección y eliminación de los pentagramas. El primero es tratar de remover las líneas del pentagrama sin eliminar las partes de los símbolos musicales que están superpuestos. El otro método es dejar las líneas del pentagrama sin tocar e idear un método para segmentar los símbolos (Carter 1989, Fujinaga 1988) [18]. En este trabajo se sigue el primer enfoque.

Para conocer otras técnicas para la detección y eliminación de las líneas del pentagrama puede consultar en [16], un buen artículo que recopila algunos de los algoritmos que han dado buenos resultados. En la sección 4.2.2 se tratan varias técnicas de diferentes autores.

#### **4.1.2.2. Segmentación de los símbolos**

La segmentación es el proceso donde los símbolos son separados y diferenciados espacialmente unos de otros. Una vez que las líneas de los pentagramas han sido detectadas y eliminadas en la partitura quedan objetos aislados (las regiones 4-conectadas) que necesitan ser reconocidos. Por esto, un paso fundamental es la segmentación, es decir la identificación de las zonas donde se encuentran los objetos aislados. En este trabajo se usan el etiquetado de símbolos y los histogramas de proyecciones como métodos de segmentación.

En la sección 4.2 se tratan varias técnicas de segmentación de símbolos de diversos autores, considerados los más destacados en el campo del OMR.

#### **4.1.2.3. Clasificación de los símbolos**

La clasificación de los símbolos musicales es la etapa en donde se le asignan a cada objeto segmentado de la etapa anterior, características que permitan su posible identificación. Las características son aspectos cuantificables de un símbolo dado y son conjuntos de las propiedades medibles del símbolo [18].

La etapa de extracción de características calcula las propiedades de los símbolos, produciendo un conjunto de medidas llamado vector de características para cada componente conectada [18]. Y posteriormente se utiliza dicho vector para clasificar los objetos. En este campo existen bastantes técnicas, por ejemplo, redes neuronales, clasificadores estructurales, k-NN, entre otros, los cuales se tratan con más detalle en [19].

Las características más usadas para la clasificación de los símbolos musicales son [13]:

- La caja de delimitación (el rectángulo más pequeño que encierra el símbolo).
- La razón entre píxeles negros y píxeles blancos dentro de la caja de delimitación.
- Las razones máximas y/o mínimas entre ancho y alto.
- Proyecciones horizontal y vertical.
- Transformadas de Hough
- Técnicas de corte (buscando una línea de píxeles en cualquier ángulo particular en el símbolo).
- Plantilla de igualación y plantilla de igualación normalizada (comparar cada píxel con el píxel correspondiente en un mapa de bits idealizado).
- Análisis de conectividad (¿existe un camino de píxeles negros entre dos o más puntos dados?).
- Análisis de la forma (el número de agujeros encerrados en la forma).
- Algoritmos de vectorización y adelgazamiento.

#### **4.1.3. Gramática**

Ciertos autores consideran que las reglas sintácticas y de la música ayudan enormemente al proceso de reconocimiento de símbolos musicales, ya que ayudan a la clasificación, a detectar errores potenciales y en algunos casos a corregirlos. Para este propósito se han hecho varias investigaciones con el fin de diseñar un sistema que incluya este conocimiento en sus módulos de proceso, con el fin de proporcionar un sistema de reconocimiento más eficiente y confiable.

Las investigaciones más interesantes en este campo son las realizadas por [13], [20], [21], [22], [23], en las cuales se tratan temas como modelado fuzzy, niveles de conocimiento, gramática, entre otros. Además, en la sección 4.3 se tratan algunos de los enfoques que en este campo se hacen para obtener un sistema OMR más eficiente.

#### **4.1.4. Representación final**

La representación de la información de la partitura es la etapa final de un sistema OMR. La representación final varía según lo que se quiere obtener. Por ejemplo, si lo que se requiere es la reproducción de la partitura, entonces su representación final puede ser la creación de un archivo de audio (.WAV, .mp3, etc.) o incluso la creación de un archivo de formato MIDI, que se puede reproducir en el

computador o un sintetizador especializado. Por otro lado, si lo que se quiere es representar la partitura en un formato especial para ciegos, en ese caso su representación debe ser una partitura con notación Braille.

Si se trata del reconocimiento de partituras manuscritas se puede requerir la representación en una partitura en formato tipográfico. En general, la representación final de la información depende fuertemente de la aplicación.

## **4.2. Investigaciones previas sobre los sistemas OMR**

Las primeras investigaciones en el reconocimiento óptico de música se dieron a finales de 1960 en varias instituciones, principalmente en el MIT (Instituto tecnológico de Massachusetts). Y eran desarrolladas en nivel de postgrado. (Pruslin, Dennis Howard (1966)). *“La investigación en el campo OMR empezó con Pruslin y Prerau. Sin embargo no fue sino hasta la década de 1980, cuando el equipo de digitalización fue accesible, que este trabajo fue expandido”* [24].

A continuación se hace una recopilación de los sistemas más conocidos sobre el tema. El propósito es dar a conocer los trabajos existentes y dar una idea de lo que los diferentes investigadores proponen en este campo.

### **4.2.1. Preprocesamiento**

Cómo se definió en la sección 4.1.1 el preprocesamiento es la etapa de un sistema OMR en la que principalmente se realizan las tareas de reducción de ruido, binarización y rotación de la partitura.

Para la eliminación de ruido, por ejemplo, manchas o grupos de píxeles aislados sin significado, inclusive para recuperar las zonas de símbolos borradas, se utilizan filtros laplacianos de 3x3 (Figura 4.2) y filtros lineales de suavizado [25].

La binarización se suele realizar manualmente y mediante umbrales adaptativos. En [15] se realiza un estudio de los métodos más conocidos en el campo de la binarización de documentos y su aplicación a partituras musicales.

La rotación puede ser tratada manualmente o mediante el empleo del concepto de cuerda (Figura 4.3) [26]. El texto se puede tratar utilizando un sistema OCR.

Otros trabajos sobre las tareas del preprocesamiento son los realizados por [27], [28], [29], en los cuales se tratan varias técnicas con buenos resultados.

$$H = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 12 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Figura 4.2 Laplaciano usado por Kim, Chung y Bien (1987)

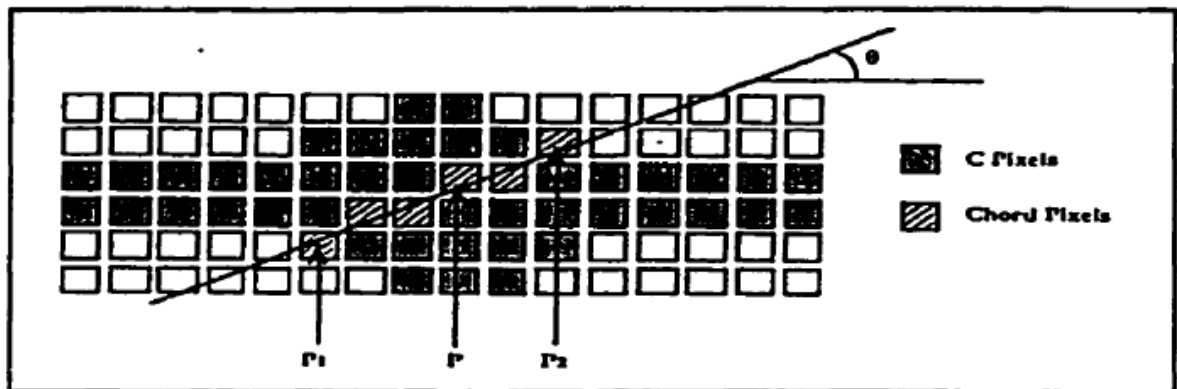


Figura 4.3 Cuerda de orientación  $\theta$  en P (Martin y Bellisant 1991)

#### 4.2.2. Líneas del pentagrama

Cómo se mencionó en la sección 4.1.2.1 el tratamiento de las líneas del pentagrama es un paso muy importante ya que facilita la segmentación de los símbolos, por lo tanto puede ser considerado un campo de investigación por sí solo.

Uno de los primeros enfoques para el tratamiento de las líneas del pentagrama fue la utilización del histograma de proyección horizontal, en el cual se ubican los pentagramas buscando los picos que superan cierto umbral. Este método tiene la ventaja de ser rápido, pero falla cuando la partitura está rotada [14], [27], [30], [31]. Otro enfoque es el de usar plantillas, para esto se construyen un conjunto de modelos de líneas horizontales y se comparan con las líneas existentes en la partitura, después, se hace un análisis de las características de las líneas encontradas y se determina cuales líneas corresponde al pentagrama y cuáles no [32]. También se suelen utilizar redes neuronales, las cuales tienen el inconveniente de ser un tanto complejas y lentas con el respectivo consumo de memoria [26]. Otro método usado para la detección de las líneas del pentagrama es la transformada de Hough [33], que además de consumir bastante memoria

tiene dificultades cuando las líneas están rotas o curvadas. En [34] se propone un método que consiste en ubicar los segmentos horizontales usando el concepto de camino más corto entre dos puntos.

Un método bastante interesante es el propuesto por Miyao y Okamoto [35] que consta de cuatro pasos. Primero, se ubican los puntos candidatos a ser pentagramas, segundo, mediante el modelo de programación dinámica (*DP matching*) se enlazan los puntos candidatos, tercero, se etiquetan los grupos resultantes y se descartan los que no corresponden a pentagramas, y cuarto, se corrigen los bordes de los pentagramas. El modelo de programación dinámica (*DP matching*) es un enfoque general para la solución de problemas en los que es necesario tomar decisiones en etapas sucesivas. Las decisiones tomadas en una etapa condicionan la evolución futura del sistema, afectando a las situaciones en las que el sistema se encontrará en el futuro, denominadas estados, y a las decisiones que se plantearán en el futuro.

En el sistema Wabot-2 [35] se utiliza un filtro hardware para la detección de las líneas del pentagrama, el cual usa plantillas de líneas horizontales para buscar los pentagramas en la partitura.

Algunos de los métodos presentados requieren del conocimiento del grosor de las líneas y de los espacios del pentagrama, para este fin se realiza un análisis del histograma de segmentos verticales de la imagen. En [16] se realiza un estudio de varios algoritmos para la eliminación de pentagramas. En [36], [37], [38], [39], [40], [41], [42], se tratan con detalle otras técnicas para el tratamiento de pentagramas.

### **4.2.3. Clasificación y Reconocimiento**

La etapa de clasificación y reconocimiento de símbolos musicales es considerada una de las etapas más complejas en un sistema OMR y para su realización existen muchos enfoques que van desde análisis de proyecciones hasta complejas redes neuronales.

El sistema Wabot-2, un robot construido a principios de los 80's (Figura 4.4) [40], [41], [42], utiliza para el reconocimiento, un sistema implementado en hardware, el cual se basa en comparar los símbolos encontrados en la partitura con plantillas almacenadas en una base de datos.

El análisis de componentes conectadas es un enfoque muy común en varios autores, los cuales realizan medidas de altura y anchura de la caja que encierra el símbolo y por medio de conocimiento de sintaxis y semántica musical determinan la naturaleza de los símbolos. El conocimiento del grosor de las líneas y de los espacios del pentagrama es necesario en este método, ya que permite normalizar la longitud, es decir, todos los tamaños de los símbolos (altura, anchura, etc.) se

miden con respecto a estas dos medidas (grosor de línea y grosor de espacio del pentagrama).



Figura 4.4 El Wabot-2 (Matsushima et al., 1985)

Otro enfoque en el reconocimiento es la comparación de los símbolos con un conjunto de modelos o *template matching*. En este método se construyen conjuntos de símbolos de diferentes editores y se almacenan en una base de datos. Las redes neuronales son bien utilizadas en el reconocimiento, las cuales realizan buenos trabajos pese a su gran consumo de memoria y tiempo de ejecución.

Otra serie de enfoques utilizan métodos tales como proyecciones verticales para la segmentación, adelgazamiento y detección de bordes, mezclas de proyecciones, transformada de Hough y *slicing* [43], [44] y algoritmos morfológicos [45].

En [23] se propone un método basado en un modelo fuzzy para la segmentación y reconocimiento, el cual hace uso de conocimiento de las reglas musicales. En [46] se utiliza el método de *contour tracing* y en [47], el método de análisis de adyacencia de grafos LAG. También son utilizados clasificadores tales como k-NN y el modelo de cadenas escondidas de Markov, entre otros.

En [14], [19], [48], [49], [50] se tratan con detalle estos métodos y se pueden observar los resultados realizados a varias partituras.

En el sistema desarrollado en este trabajo se tratan conceptos sencillos para la segmentación y el reconocimiento, tales como los histogramas de proyección, etiquetado y medidas de altura, ancho y tamaño.

A manera de conclusión se puede decir que uno de los aspectos que se debe tener en cuenta es que algunas de las técnicas empleadas por los sistemas más eficientes son de tipo hardware que requieren de presupuestos grandes que son difíciles de ser financiados en nuestro medio.

### **4.3. Distintos enfoques de los sistemas OMR**

Varios autores han propuesto diferentes enfoques para cambiar la estructura de los sistemas OMR con el fin de mejorar su rendimiento. La mayoría de estos enfoques están orientados a la manera de llevar a cabo las etapas de segmentación y reconocimiento.

En [51], [52] y [53] se propone una nueva solución para el tratamiento de partituras, en el cual se usa una gramática para guiar la segmentación de los objetos y para su reconocimiento. La gramática esencialmente es una descripción de las relaciones (posición y tamaño relativo, adyacencia, etc.) entre los objetos. Inspirados por las técnicas *Definite Clause Grammar*, la gramática puede ser directamente implementada en  $\lambda$ Prolog, un dialecto de alto nivel de Prolog. El enfoque utilizado se justifica con los alentadores resultados obtenidos con un prototipo para el reconocimiento de partituras.

En [21] se propone un enfoque original para el reconocimiento de partituras. Con el fin de superar las limitaciones de los sistemas existentes, sus autores proponen una arquitectura que permite una interacción continua y bidireccional entre conocimiento de alto nivel y datos de bajo nivel, la cual es capaz de mejorar por sí misma el tiempo de procesamiento por medio de aprendizaje. Esta arquitectura está hecha de tres capas que cooperan entre sí, una hecha de detectores de características parametrizadas, otra trabajando como un depósito de conocimiento orientado a objetos y la otra como un metaprocador de supervisión bayesiano.

En [54] se describe un sistema que ha sido diseñado para ser extensible sin necesidad de cambiar el código fuente del sistema. La extensibilidad se consigue proporcionando herramientas para el reconocimiento de música que se utilizan para que el sistema se adapte al tipo de música que va a reconocer. Las herramientas incluyen una selección de métodos para la identificación de los pentagramas y el aislamiento de los objetos, métodos para la identificación de

formas musicales primitivas y una gramática para especificar las relaciones entre las formas reconocidas. El sistema es bastante flexible para trabajar con conjuntos de símbolos de diferentes editores, e incluso con diferentes tipos de notación musical, tales como la notación cuadrada utilizada en la música antigua.

En [55] se describen los desafíos presentes en el reconocimiento óptico de música (OMR). Primero, se describe el problema; luego se presenta un entorno software generalizado enfatizando en las etapas clave que deben ser resueltas: identificación de las líneas del pentagrama, localización de los objetos musicales, clasificación de características musicales y semántica musical. Además, en este artículo se revisan otros trabajos de investigación sobre el área. El trabajo concluye con la discusión de la mayoría de preguntas abiertas en el campo, por ejemplo, cómo comparar la eficacia y éxito de sistemas rivales, resaltando ciertos pasos que facilitan la tarea.

En [22] se presenta un sistema OMR que puede reconocer automáticamente los principales símbolos musicales de una partitura escaneada (Figura 4.5).

En el trabajo mencionado previamente se distinguen dos etapas principales: la primera usa preprocesamiento de bajo nivel para detectar los objetos aislados y da como resultado algunas hipótesis sobre ellos; la segunda tiene que tomar la decisión correcta final, haciendo uso de procesamiento de alto nivel, de información contextual y de las reglas de la escritura musical. Este artículo expone las dos etapas del método, después explica detalladamente el primero, el proceso de análisis de símbolos, que muestra a través de primeros experimentos que su salida se puede usar eficientemente como entrada a un proceso de decisión de alto nivel.

En [56] se describe un trabajo que incluye el uso de retroalimentación entre las diferentes fases del proceso OMR para corregir errores automáticamente, tales como, errores en la clasificación de símbolos o errores en la detección de pentagramas (Figura 4.6).

En [57] se estudia la posibilidad de lograr mejorar sustancialmente la precisión de los programas OMR usando la salida de varios programas para “triangular” y obtener mejores resultados que cualquiera de los programas individuales. Este enfoque de múltiple-reconocimiento ha tenido éxito en otras áreas pero nunca ha sido probado en música. Uno de los mayores obstáculos es que la complejidad de la notación musical es tal, que la evaluación de la calidad de los sistemas OMR es difícil, incluso para las partituras más simples.

Fujinaga [18], en su tesis doctoral desarrolla un sistema OMR adaptativo (AOMR), es decir, que tiene la capacidad de aprender. El punto fuerte de este sistema es su capacidad para aprender nuevos símbolos musicales y notación manuscrita. La



tasa de reconocimiento se mejora al ajustar continuamente los parámetros internos.

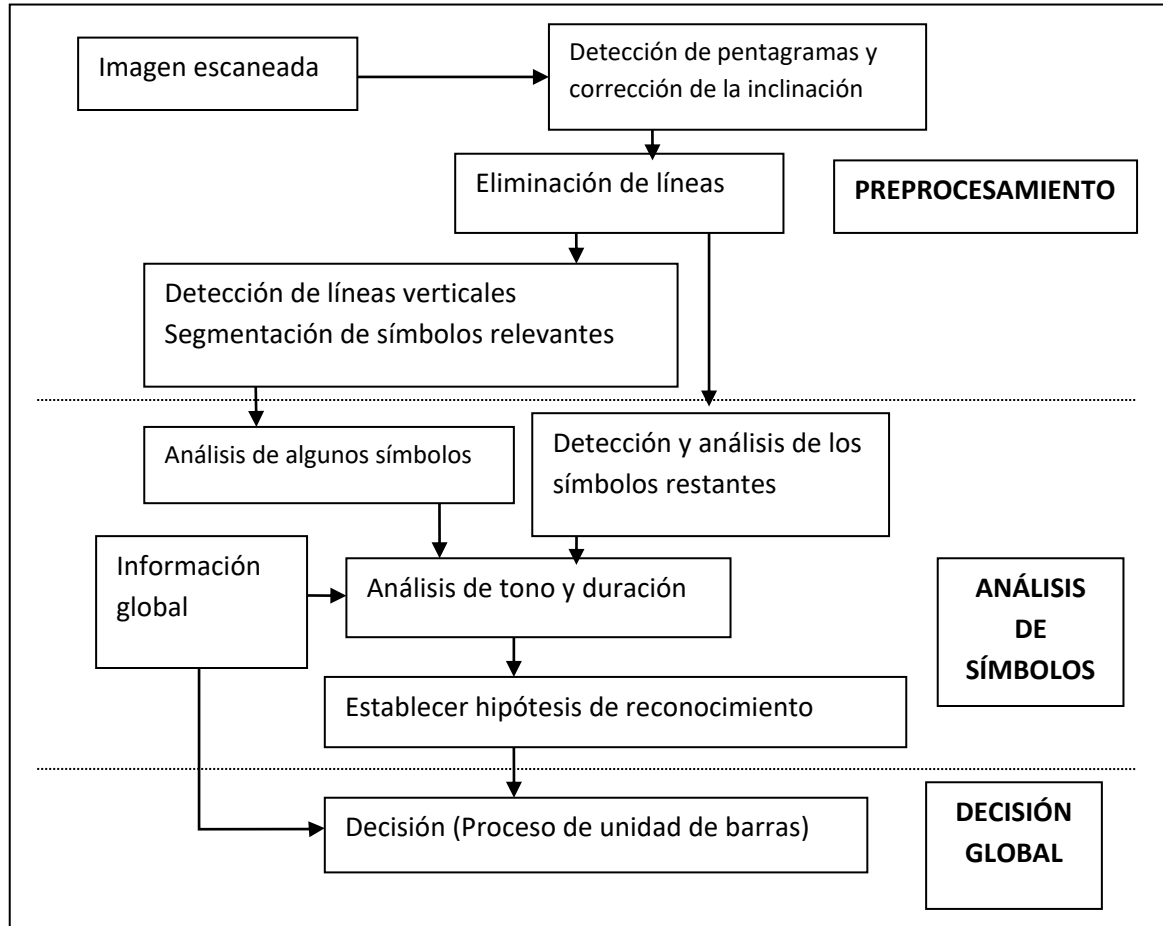


Figura 4.5 Diagrama de flujo del proceso (Adaptada de Rossant 2002)

La implementación del sistema adaptativo de Fujinaga está basada en el aprendizaje incremental mediante ejemplos o aprendizaje por ejemplos, que identifica símbolos desconocidos por sus similitudes con ejemplos conocidos almacenados. El proceso está basado en dos simples pero poderosos algoritmos, un clasificador del vecino más cercano  $k$  y un algoritmo genético. Haciendo uso de estos algoritmos, el sistema está diseñado para mejorar su precisión a medida que procese más datos.

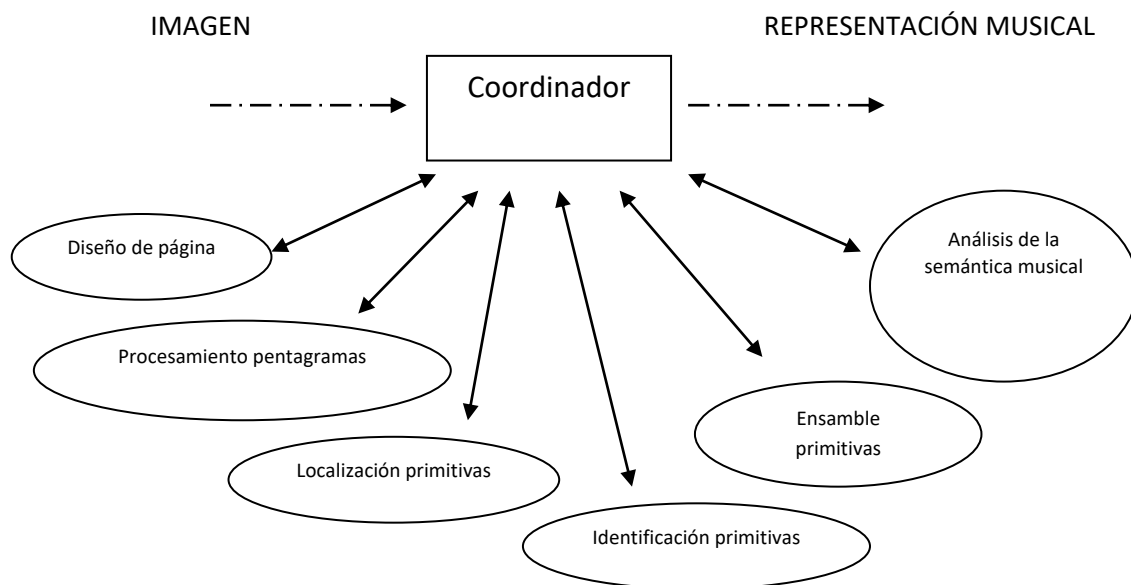


Figura 4.6 Estructura del sistema (Adaptada de McPherson 2003)

El sistema está compuesto básicamente por una base de datos y tres procesos independientes: un procesamiento (localiza, separa y clasifica los símbolos musicales), edición (muestra y reproduce el resultado del procesador, hace uso del editor de dominio público Nutation de Glen Diener) y aprendizaje (mejora la rapidez y eficacia de futuras sesiones de reconocimiento).

Blostein y Baird [17] estudian y examinan críticamente la literatura de investigación concerniente al análisis automático de partituras musicales manuscritas y no manuscritas, para el periodo de 1966 a 1990.

El trabajo de investigación de Blostein y Baird [17] además, trata sobre el objetivo y las aplicaciones de los sistemas OMR, métodos no ópticos de entrada y sobre la terminología empleada. Blostein y Baird hace una recopilación cronológica de las investigaciones más importantes en el campo de los sistemas OMR, haciendo énfasis en el tratamiento de las líneas de pentagrama y la clasificación de símbolos musicales. Trata, también los métodos sintácticos, la notación de baile, que es una notación con muchas similitudes a la notación musical y hace algunas críticas a los problemas más comunes en el área (defectos de la imagen y objetos rotos).

#### 4.4. Trabajos relacionados con los sistemas OMR

##### 4.4.1. McGee y Merkely (1991)

McGee y Merkely [58] diseñaron un sistema para el reconocimiento de neumas procedentes de la notación antigua (Figura 4.7). Se buscan las líneas delgadas suficientemente largas para detectar las cuatro líneas de su sistema de escritura y posteriormente se eliminan y se enderezan. Para la clasificación de los símbolos usaron las medidas de las cajas que los encierran.

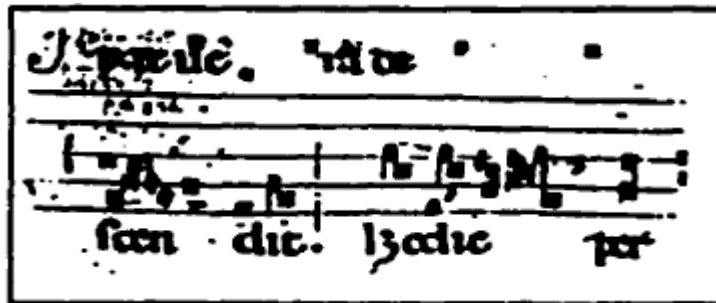


Figura 4.7 Notación antigua (McGee y Merkely 1991)

##### 4.4.2. Rebelo (2008)

Rebelo [24] investiga un método de propósito general y conocimiento libre para la detección automática de las líneas del pentagrama basado en el método del camino estable. Las líneas afectadas por curvatura, discontinuidades e inclinación son detectadas robustamente. También ha desarrollado un algoritmo de eliminación de líneas adaptando un proceso ya existente para usar el algoritmo de camino estable en la fase de detección. Los resultados experimentales obtenidos muestran que la técnica propuesta supera consistentemente los algoritmos bien establecidos.

Rebelo propone un método de segmentación basado en la descomposición jerárquica de la partitura. Los símbolos son divididos en cuatro tipos para facilitar su extracción. Ha presentado una nueva metodología donde se usa un modelo elástico en conjunción con varios clasificadores, por ejemplo redes neuronales y cadenas de Markov escondidas. Ha presentado un profundo estudio comparativo sobre estos clasificadores con o sin modelación elástica aplicados a las partituras manuscritas (Figura 4.8).

#### 4.4.3. Tardón et al. (2010)

Tardón et al. [59] presentan un sistema OMR especializado adaptado para el tratamiento de partituras de los siglos XVII y XVIII escritas en notación mensural blanca (Figura 4.9). El sistema ejecuta una secuencia completa de etapas de análisis: la entrada es una imagen RGB de la partitura a ser analizada y, después de un preprocesamiento que binariza la imagen con la rotación correcta, los pentagramas son procesados para obtener una imagen sin líneas de pentagrama; luego una etapa de procesamiento aísla los símbolos musicales contenidos en la partitura y, finalmente, el proceso de clasificación obtiene la transcripción en un formato electrónico apropiado para que pueda ser almacenado o reproducido. Este trabajo ayudará a preservar la herencia cultural manteniendo la información musical de las partituras en un formato digital que también da la posibilidad de ejecutar y distribuir la música original contenida en esas partituras.

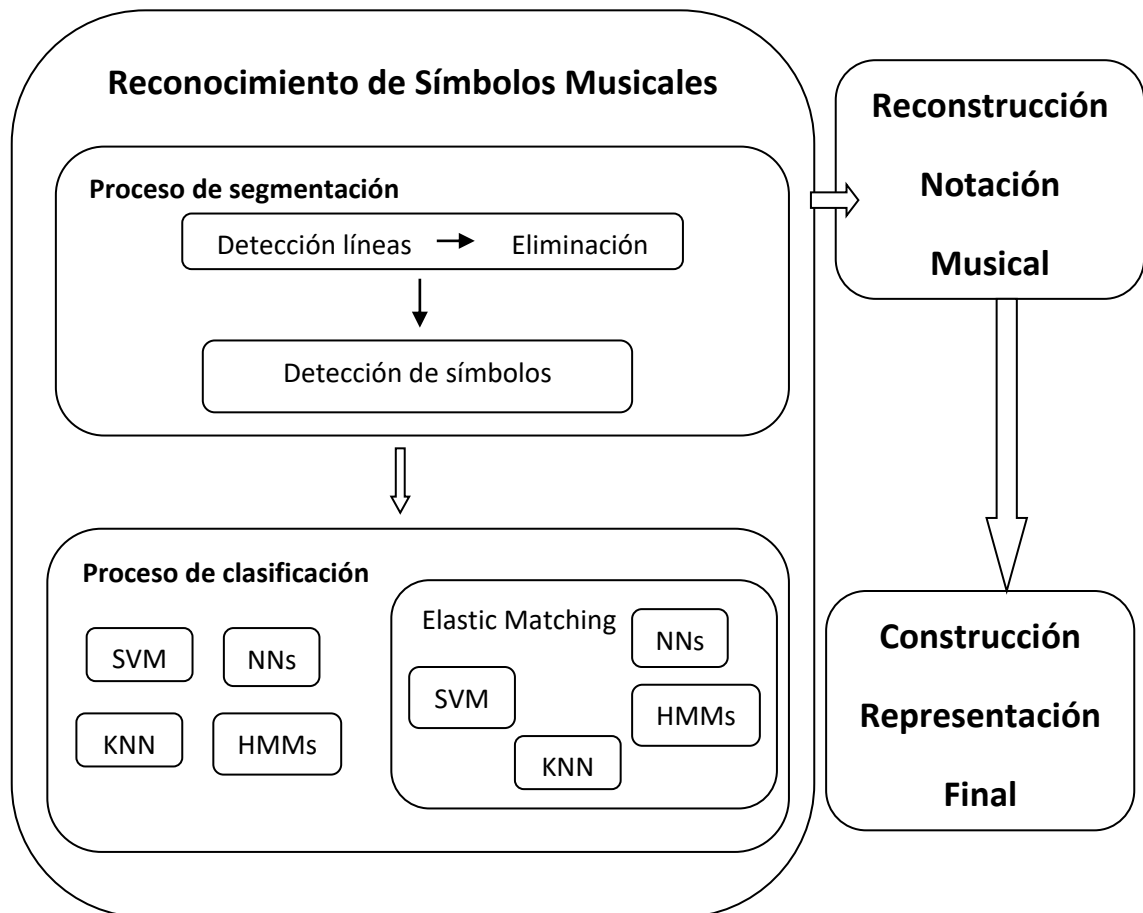


Figura 4.8 Arquitectura OMR (Adaptada de Rebelo 2008)

#### 4.5. Software disponible

El primer programa comercial que permitió la digitalización de partituras musicales mediante técnicas OMR fue lanzado en 1993 a cargo de la empresa Musitek Corporation. Esta empresa fue fundada en mayo de 1991 por Christopher Newell llevó a cabo una serie de investigaciones para lanzar en agosto de 1993 el programa MIDISCAN para Microsoft Windows. Las primeras versiones tenían una tasa de acierto del 25% de la notación de la partitura, por lo que el usuario tenía que corregir manualmente el 75% de error. Después, con la versión dos del programa, la tasa aumentó al 80% para partituras no manuscritas y del 50% para partituras manuscritas.

Años más tarde, Musitek desarrolló un programa que integraba la tecnología desarrollada para escanear música junto a un editor de partituras por computador, utilizando un sintetizador MIDI para grabar y sintetizar la música, naciendo así SmartScore para Microsoft Windows y Masintosh, en 1996.



(a) Fragment of a score written in the style of *Stephano di Britto*



(b) Fragment of a score written in the style of *Francisco Sanz*

Figura 4.9 Fragmentos de partituras en notación mensural blanca (Tardón et al., 2010)

Actualmente se han desarrollado una multitud de programas de reconocimiento basados en técnicas OMR. La eficacia de estos programas puede llegar al 99% cuando la partitura escaneada está limpia (sin manchas, ni deformaciones, ni cortes y discontinuidades en la imagen) y la notación que se utiliza no contiene símbolos inusuales.

#### 4.5.1. Software Comercial

A lo largo de los años, han ido apareciendo varios software OMR comerciales, pero ninguno tiene un desempeño perfecto en términos de precisión y robustez. La complejidad de la tarea OMR, causada por la estructura bidimensional de la notación musical y también por la existencia de varios símbolos combinados organizados en torno a la cabeza de las notas ha ido condicionando el progreso en esta área [24].

La mayoría de los software OMR comerciales escanean las partituras y luego las presentan en un formato particular. El procesamiento de las partituras incluye la detección de las líneas de los pentagramas, el aislamiento de los objetos (símbolos musicales) y el análisis y reconocimiento de estos símbolos. El resultado del procesamiento es una nueva partitura que puede ser impresa o ser escuchada. El formato más usado para presentar el resultado es el MIDI y el MusicXML, por lo que una vez realizado el reconocimiento, el archivo de audio generado en un programa puede ser abierto y tratado en programas editores de música como Finale y Sibelius. Los símbolos que el programa reconoce son: las notas, los silencios, las alteraciones, signos de duración. Inclusive hay algunos programas que usan motores OCR para reconocer los títulos de las partituras y los signos de expresión de la música, que generalmente consisten en palabras. Además, los programas permiten realizar cambios en los resultados cuando el reconocimiento no es correcto. Los costos de estos programas oscilan entre los 200 y los 800 dólares.

Hasta ahora, aún los sistemas de reconocimiento más desarrollados (PIANOSCAN, NOTESCAN en Nightingale, MIDISCAN en Finale, PHOTOSCORE en Sibelius, SMARTSCORE, SHARPEYE, etc.) no pueden identificar todas las notaciones musicales [24].

Algunos de los programas comerciales más conocidos son: Capella-scan, PhotoScore, ScoreMaker, SharpEye, SmartScore, SmartScore Lite, PDFtoMusic, VivaldiScan, OmeR.

A manera de conclusión, se puede decir que estos programas comerciales son muy sensibles a la calidad de impresión de la partitura, y que las reglas musicales no están suficientemente integradas en el proceso de decisión (errores en la métrica, alteraciones en una posición incorrecta o incoherente, entre otros). Esto es importante porque los trabajos realizados hasta el momento se limitan generalmente a la estructura y a los gráficos, y dejan de lado los aspectos sintácticos, tales como las normas relativas al tono y las alteraciones [23].

#### **4.5.1.1. SmartScore Pro 5**

Es un programa OMR desarrollado por Musitek Corporation. SmartScore corre en plataformas Windows XP, Windows Vista y Macintosh OSX. SmartScore le permite al usuario escanear sus partituras impresas, mostrarla en la pantalla en forma editable, reproducirla, manipularla, transponerla, reformatearla y reimprimirla [60]. Tiene restricciones en cuanto al formato de las partituras de entrada, permitiendo únicamente formatos MusicXML, PDF, Finale y TIFF. Además, SmartScore no trabaja con partituras manuscritas. El precio de este paquete oscila entre los 400 dólares [61].

SmartScore es un software para Windows muy reconocido. Este programa no es capaz de encontrar una solución correcta de acuerdo a las reglas elementales de escritura de la música, como la longitud de un compás [22].

#### **4.5.1.2. PDFtoMusic**

PDFtoMusic es una aplicación que le permite al usuario reproducir partituras en formato PDF, de esta manera se escuchará la melodía escrita en el archivo de una manera eficaz cuando la calidad de la partitura es 100% perfecta. Cuando el escaneo no es limpio o cuando hay algún defecto en la imagen, el reconocimiento puede llegar a ser tan pobre que la canción que se reproduce no va de acuerdo con la real, por lo general faltan notas, la canción se adelanta o se atrasa en cuanto a ritmo. El tiempo de reconocimiento de la partitura es muy grande.

Este software cuenta con diversas e interesantes funciones que no solamente le permiten al usuario reproducir la partitura escrita en el archivo PDF, sino que además, permite seleccionar el tempo de reproducción, el volumen y hasta exportar dicha partitura a diversos formatos para que puedan ser editadas por medio de otras aplicaciones como Melody o Harmony Assistant [62].

#### **4.5.1.3. Vivaldi-Scan**

Vivaldi-Scan permite convertir casi cualquier partitura en un archivo MIDI, con solo pasarla por el escáner. Además del formato MIDI, el archivo puede ser exportado en el formato del editor de partituras Vivaldi Gold, lo que permite una personalización absoluta del resultado. Vivaldi-Scan posee un sistema de detección de posibles errores: el programa avisa, remarcando esas notas en color gris, los pasajes donde el software ha tenido dudas de reconocimiento [63]. El reconocimiento de partituras tiene sus limitaciones, sobre todo si se pretende escanear partituras manuscritas, o con grafía algo compleja.

Para que Vivaldi-Scan funcione perfectamente, las partituras deben estar limpias y sin ninguna falla en los símbolos.

#### **4.5.2. Software Libre**

Los programas libres en el campo OMR no necesitan una licencia para su uso, surgen como trabajos de investigación en determinados departamentos de instituciones o universidades, muchas veces como proyectos para resolver determinados problemas que no son capaces de solventar los programas comerciales. El principal problema de estos programas es que no suelen tener un grado de madurez equiparable al de un sistema comercial, por lo que es común que dichos proyectos sean mejoras de sistemas más desarrollados. Algunos de los programas no comerciales más conocidos son: Audiveris (Java), Gamera, OpenOMR.

#### **4.6. Aplicaciones**

El reconocimiento automático de partituras ha sido realizado por diferentes razones y por lo tanto los objetivos técnicos también varían. Por ejemplo, para el análisis de estilos musicales, solamente es suficiente extraer el tono, la duración y la simultaneidad de todas las notas. Para producir particellas dada una partitura (o viceversa), deben ser reconocidos y asociados con voces todos los símbolos musicales, no solo las notas [17].

La siguiente lista de aplicaciones del reconocimiento de música impresa ha sido recopilada de varios autores, incluidos [14], [64] y [65]. Una parte importante de las aplicaciones es la que concierne a la edición de partituras para reimpresión, revisión y preparación de materiales de ejecución:

- Adaptar trabajos existentes a otros instrumentos, por ejemplo reducir partituras completas a partituras de piano.
- Tomar varios trabajos en ediciones viejas y producir una edición nueva.
- Hacer ediciones críticas de composiciones musicales dadas diferentes versiones impresas de la misma composición.
- Transponer una partitura musical a otras claves.
- Producir partes desde una partitura dada o una partitura desde las partes.
- Convertir las partituras existentes a la notación Braille para ayudar a los músicos ciegos.
- Imprimir automáticamente la música escrita (si el programa de reconocimiento fuera extendido al reconocimiento de notación musical manuscrita)



- Producir versiones de audio de una composición dada; el computador puede ser usado como una combinación músico-instrumento.

Otra clase de aplicaciones son las que conciernen a la creación de bases de datos:

- Crear índices de temas y otras características musicales.
- Análisis de estructura y estilo musical.
- Probar teorías musicales.
- Evaluar algoritmos para el análisis automático o composición de música.

La primera parte de las aplicaciones, especialmente la concerniente a representar la partitura mediante un archivo de audio puede ser encaminada al aprendizaje de la música ya que el aprendiz puede utilizar el sistema OMR como un medio para ensayar solfeo y afinación. Además, el intérprete de un instrumento puede utilizar el computador como acompañante y por consiguiente mejorar su capacidad para tocar en conjunto.

## **CAPÍTULO 5. SISTEMA OMR IMPLEMENTADO**

El sistema OMR implementado en este trabajo consta de cinco etapas fundamentales que son:

1. Adquisición de las imágenes
2. Preprocesamiento
3. Detección y eliminación de pentagramas
4. Segmentación y reconocimiento de símbolos musicales
5. Interpretación de la partitura

Cada una de estas etapas se trata con detalle a continuación.

### **5.1. Adquisición de las imágenes**

Para la adquisición de las imágenes fue necesario el uso de un conjunto de partituras y de un escáner para la digitalización (Figura 5.1). Las partituras, obtenidas de un libro de música, son un conjunto de 110 ejercicios de dificultad creciente, las cuales se usan como ejercicios de solfeo y de repaso con el instrumento en los cursos básicos de música. Las partituras son monofónicas, es decir contienen una sola melodía y están compuestas por sistemas de un solo pentagrama.

En este trabajo solo se tienen en cuenta las partituras que contienen melodía y no las que contienen armonía, es decir la simultaneidad de los sonidos. En estas últimas la notación musical empleada difiere un poco ya que se deben utilizar otros símbolos para representar los sonidos.

#### **5.1.1. Imágenes adquiridas**

Las imágenes que se obtuvieron fueron escaneadas con una resolución de 400 dpi y sus tamaños varían desde los 1600 hasta los 2800 píxeles para el ancho y desde los 400 hasta los 1000 píxeles para el alto (Figura 5.2). Son imágenes en escalas de grises y presentan bastantes zonas de baja calidad, como borrosidad y objetos rotos, etc. La razón de la elección de estas partituras se debe a los aspectos favorables que poseen, por ejemplo, no contienen demasiados elementos de adorno y la notación musical presente es básica y suficiente para empezar en el aprendizaje de la música.

### 5.1.2. Características de las partituras y de la notación musical reconocidas

Los elementos que pueden ser reconocidos por el sistema son los siguientes:

- **Pentagramas.** Sistemas de un pentagrama.
- **Clave.** De sol en segunda línea, de fa en cuarta línea o de do en tercera línea.
- **Armadura.** Conjunto de hasta cuatro alteraciones (sostenidos o bemoles).
- **Indicador de compás.** Mayor, compasillo, 4/4, 3/4, 2/4 o 6/8.
- **Barras de compás.**
- **Ligaduras.**
- **Notas individuales.** Redonda, blanca, negra y corchea.
- **Grupos de notas.** Grupos de hasta cuatro notas que pueden ser combinaciones de corcheas y semicorcheas.
- **Silencios.** Blanca, negra y corchea.
- **Puntillos.**
- **Alteraciones.** Sostenidos, bemoles y becuadros.

El sistema presentado en este trabajo no es capaz de detectar cambios de clave, tonalidad o de indicador de compás.

La detección de las barras de compás garantiza el uso de una gramática para el reconocimiento de las notas como se verá en la sección 5.4.10.

Las ligaduras son detectadas pero no se tienen en cuenta en la interpretación final de la partitura. Por otro lado los puntillos si son reconocidos y tenidos en cuenta en la interpretación.

El sistema presentado en este trabajo es capaz de reconocer tanto las notas que están dentro del pentagrama como las que se encuentran por fuera. Para estas últimas solo se tienen en cuenta las notas que se encuentran hasta en la tercera línea adicional por encima y por debajo del pentagrama.



Figura 5.1 Escáner utilizado para digitalizar las partituras

En este trabajo se reconocen las notas más comunes, es decir las que aparecen con más frecuencia en la mayoría de la música popular y en buena parte de la música clásica. Dentro de las notas individuales se reconocen la redonda, blanca, negra y corchea y dentro de los grupos de notas se reconocen combinaciones de hasta cuatro corcheas y semicorcheas.

El sistema es capaz de reconocer los silencios de blanca, negra y corchea solamente por ser los únicos tipos que aparecen en el conjunto de partituras escogidas para las pruebas.

En este trabajo no se reconocen los indicadores de movimiento, pero en la aplicación se pueden seleccionar varias velocidades de reproducción, a la conveniencia del usuario (ver Anexo A). A modo de información hay programas comerciales que hacen uso de un motor OCR para el reconocimiento de este tipo de símbolos.

El presente trabajo tiene en cuenta tres de las cuatro dimensiones presentadas, estas son: la frecuencia, la duración y el timbre. La frecuencia y la duración son determinadas por el algoritmo cuando se reconoce la posición y la figura de las notas. El timbre para interpretar la partitura puede ser seleccionado por el usuario al momento de configurar la aplicación (ver Anexo A). El programa permite utilizar hasta 17 voces diferentes.



Figura 5.2 Ejemplos de imágenes adquiridas con diferentes calidades y tamaños

## 5.2. Preprocesamiento

La etapa de preprocesamiento hace uso de básicamente cuatro algoritmos que son: (1) el filtro paso bajo, (2) la binarización, (3) el filtro morfológico y (4) el suavizado binario, los cuales se aplican convenientemente (Figura 5.3).

En el preprocesamiento, a partir de una imagen borrosa en escala de grises, se obtiene una imagen binaria de mejor calidad, en la que se resaltan mejor los objetos y se recuperan ciertas zonas, para que pueda ser tratada con mayor facilidad en los procesos siguientes.

### 5.2.1. Filtro paso bajo

El filtro de paso bajo (ver sección 3.2.3) se aplica para suavizar los aspectos de la imagen, reducir algo de ruido y especialmente porque modifica considerablemente la configuración del histograma, facilitando el cálculo de un umbral para la etapa de binarización (Figuras 5.4 y 5.5). Los filtros usados son dos: el primero (Figura 5.6a) es un filtro de suavizado haciendo énfasis en el píxel central de la ventana de cálculo y el segundo (Figura 5.6b) es un filtro de suavizado homogéneo para todos los píxeles de la imagen.

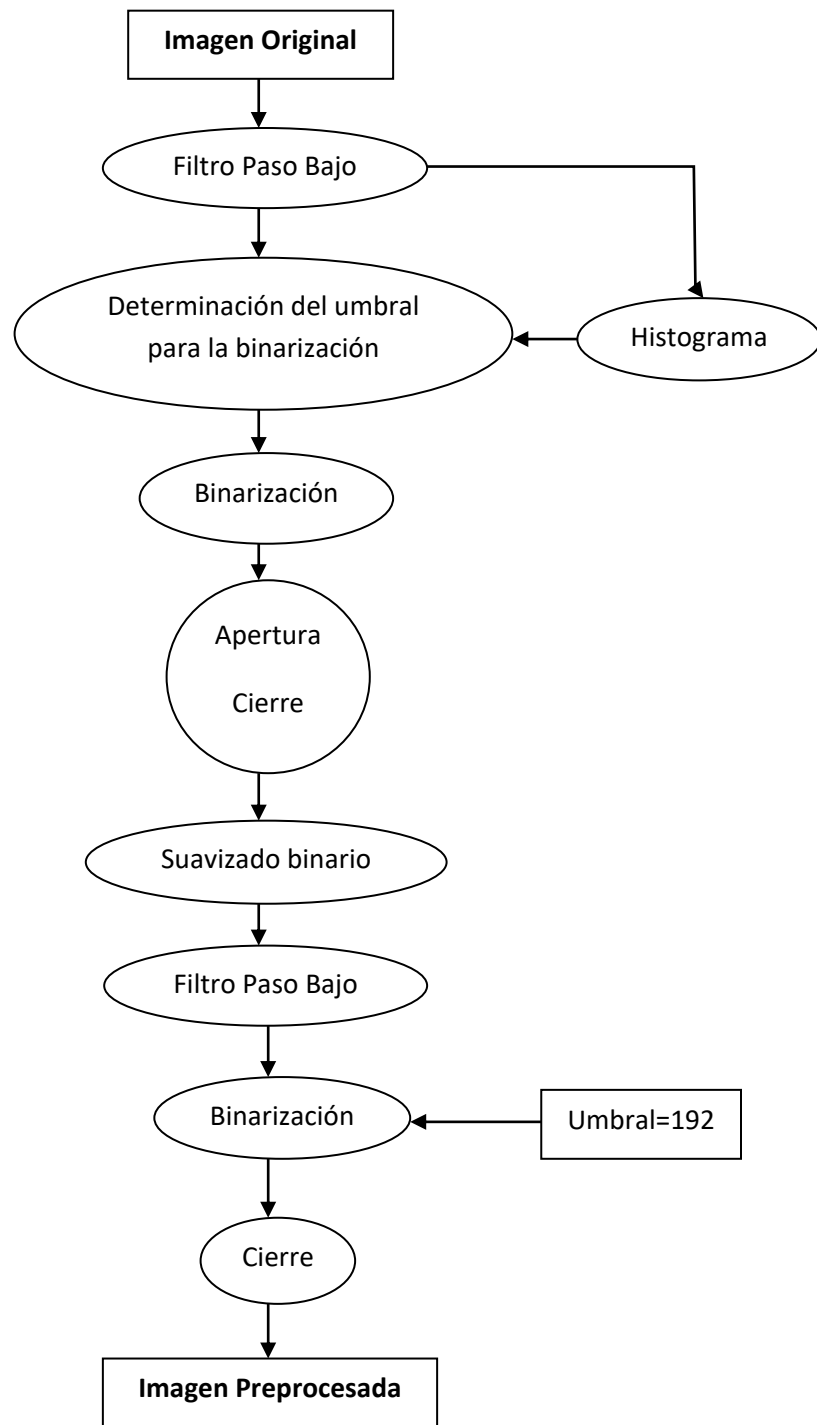


Figura 5.3 Diagrama de flujo del sistema de preprocesamiento propuesto

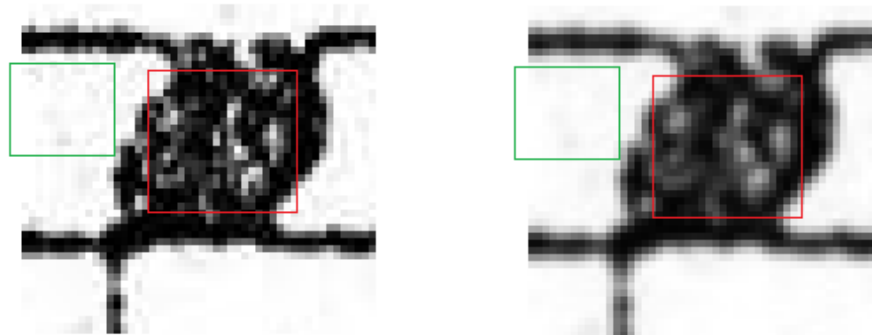


Figura 5.4 Resultado de aplicar el primer filtro

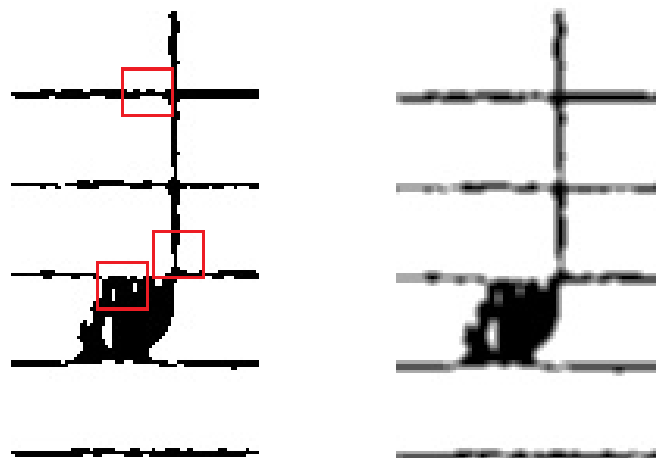


Figura 5.5 Resultado de aplicar el segundo filtro

a)

1	2	1
2	4	2
1	2	1

b)

1	1	1
1	1	1
1	1	1

Figura 5.6 Los dos tipos de filtro paso bajo usados

### 5.2.2. Binarización

Las imágenes adquiridas están en escala de grises por lo que es mucho más difícil su tratamiento, ya que los bordes de los objetos no están muy bien definidos, por lo tanto es necesario convertir dichas imágenes a un formato binario, en donde se resalten mejor los objetos y se diferencien claramente del fondo (Figura 5.7).

El proceso de binarización se realiza como se describe en la sección 3.2.6, utilizando como umbral, el valor extraído directamente del pico por encima del nivel 128 en el histograma de la imagen (Figura 5.8). Se determinó empíricamente que este pico corresponde a las zonas de importancia (objetos), por lo que es de vital importancia ubicarlo.

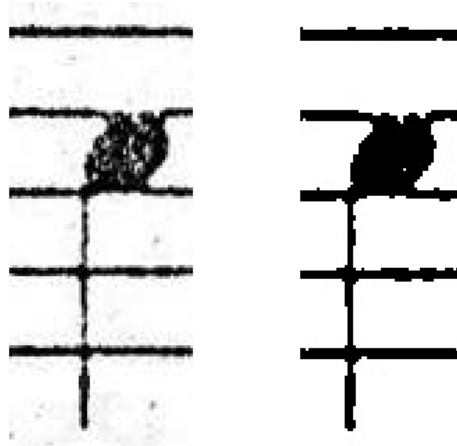


Figura 5.7 Resultado de aplicar la binarización

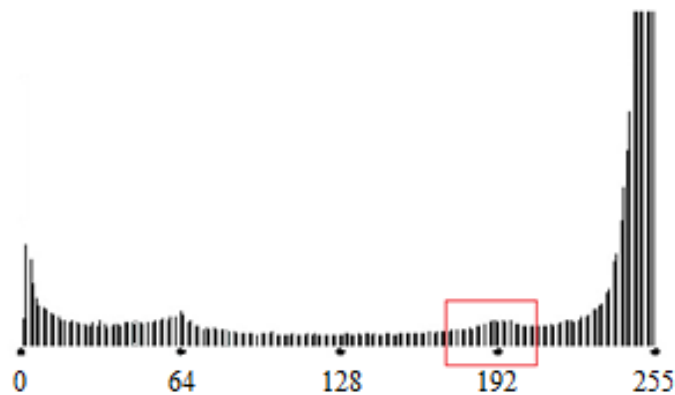


Figura 5.8 Histograma de la imagen resaltando el pico que determina el umbral de binarización



### 5.2.3. Filtro morfológico

La aplicación del filtro morfológico (sección 3.2.4) se realiza para recuperar ciertas zonas de las notas perdidas ya sea por el uso de las hojas antes de ser escaneadas o en los procesos de fotocopiado y escaneo. Además el filtro se usa para eliminar elementos de la imagen que no contienen información musical, píxeles aislados y conjuntos de píxeles demasiado pequeños (Figura 5.9).

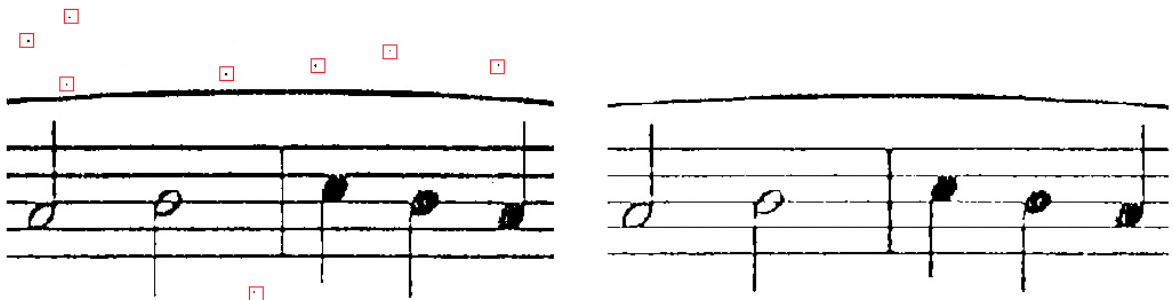


Figura 5.9 Aplicación del filtro morfológico

### 5.2.4. Suavizado binario

El suavizado binario (sección 3.2.5) elimina ciertas configuraciones de píxeles que se consideran que no contienen información. Para ello se diseñaron once máscaras (Figura 5.10).

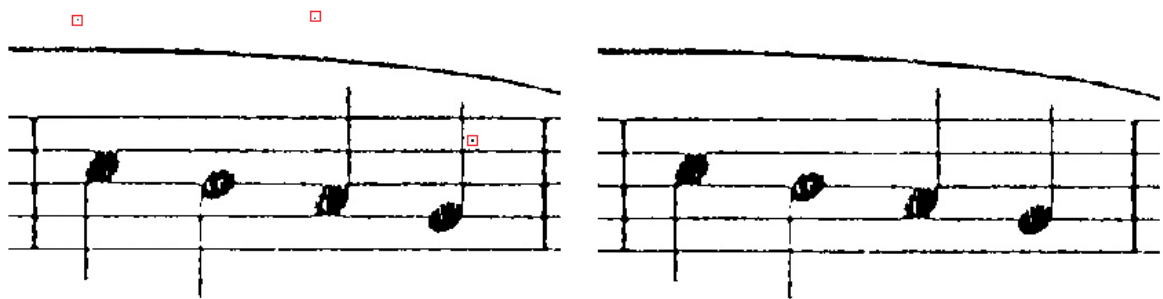


Figura 5.10 Aplicación del suavizado binario

### 5.3. Detección y eliminación de pentagramas

Con la imagen binarizada y mejorada considerablemente, el siguiente paso es la detección y la eliminación de las líneas del pentagrama. El resultado de este paso es una imagen con objetos aislados que pueden o no ser símbolos musicales. El proceso de detección de las líneas se desarrolló haciendo uso de un modelo propuesto por Miyao y Okamoto en [35], el cual consiste fundamentalmente de cuatro pasos (sección 5.3.1).

Una vez detectadas las líneas se procede a eliminarlas, para lo cual se eliminan todos aquellos segmentos verticales que están sobre la posición de las líneas que cumplan una condición de longitud. Algunos símbolos quedarán rotos después de aplicar este paso, especialmente las zonas de los objetos que son tangentes a las líneas.

Para la implementación de este modelo es necesario conocer el grosor de las líneas y la longitud de los espacios del pentagrama. Para ello se realizó un análisis del histograma de segmentos codificados (sección 3.3.1). Se seleccionaron 10 líneas verticales igualmente espaciadas a lo largo de toda la imagen y se midieron los segmentos negros y blancos en cada una de esas líneas. A continuación se construyeron dos histogramas de segmentos, uno para los segmentos negros y otro para los segmentos blancos. El grosor de las líneas corresponde al valor con mayor frecuencia en el histograma de segmentos negros y la longitud de los espacios corresponde al valor con mayor frecuencia en el histograma de segmentos blancos.

Además, el conocimiento de estos valores permite normalizar las longitudes de los símbolos. Por lo tanto, de aquí en adelante todas las medidas se harán con respecto a estos dos valores. En otras palabras el grosor de las líneas, representado con  $L$ , y la longitud de los espacios, representada con  $E$ , son las unidades de medida para las distancias.

#### 5.3.1. Detección de las líneas

Una vez obtenido el grosor de las líneas y la longitud de los espacios se procede con el modelo de detección propuesto en [35].

1. Se recorre la imagen en 34 líneas verticales igualmente espaciadas en busca de segmentos negros iguales al grosor de las líneas del pentagrama. Los puntos medios de estos segmentos son considerados como puntos candidatos.
2. Mediante el uso del modelo de programación dinámica, *DP matching*, se enlazan los puntos candidatos de cada línea de escaneo con los puntos candidatos de la línea vecina teniendo en cuenta dos condiciones. Primero,

la inclinación de la línea recta que une los dos puntos candidatos debe estar dentro de cierto umbral de inclinación ( $20^\circ$ ). Y segundo, el camino que forma dicha línea recta debe al menos contener un 90% de píxeles negros.

3. Se usa el etiquetado para separar los grupos resultantes del paso anterior, en grupos de pentagramas. Se cuentan las líneas horizontales para cada grupo y los grupos que contengan menos de 5 líneas horizontales son eliminados, los grupos que contienen más de 5 líneas horizontales son procesados hasta obtener un grupo de 5 líneas, para esto se empieza eliminando las líneas horizontales de menor longitud hasta obtener un total de 5 líneas. Los grupos con 5 líneas se dejan intactos. Los grupos resultantes corresponden a los pentagramas.
4. Por último, se ubican los bordes de los pentagramas y se corrigen sus posiciones.

Con este modelo, además de detectar los pentagramas, se calculan las coordenadas que determinan su posición, dato muy importante para la selección del área de análisis que se trata en la sección 5.4.2 y el grado de inclinación de la partitura, el cual es necesario para la rotación que se trata en la sección 5.4.1.

En la figura 5.11 se observa una partitura tratada con este modelo (imagen superior) y los pentagramas extraídos correctamente (imagen inferior). En esta última se observan las cuatro coordenadas que determinan la posición del pentagrama.

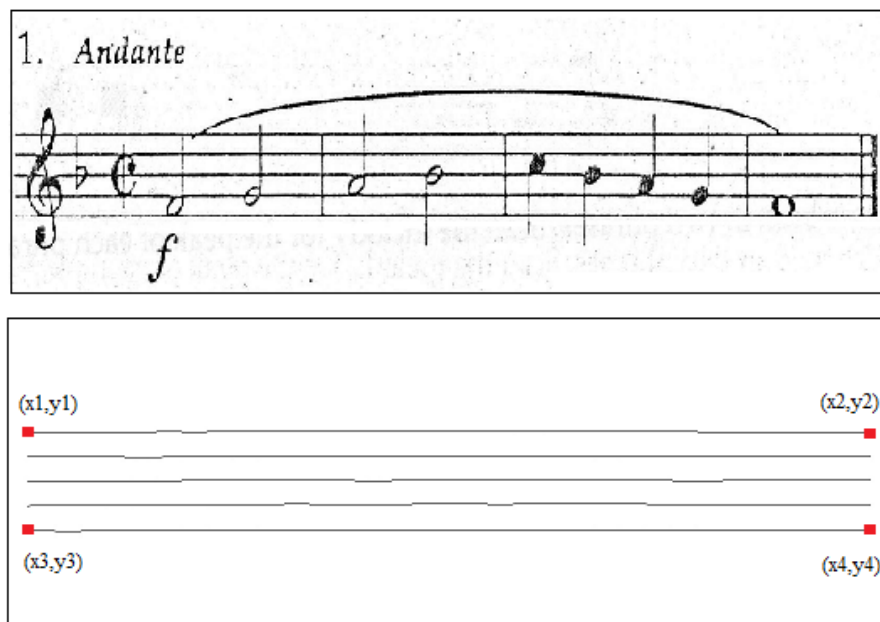


Figura 5.11 Extracción del pentagrama

### 5.3.2. Eliminación de las líneas

Una vez que los pentagramas han sido localizados, el siguiente paso es su eliminación. El proceso consiste en recorrer las líneas detectadas en el paso anterior y eliminar los segmentos verticales que tienen una longitud específica. El objetivo es eliminar las secciones de los pentagramas que no se superponen con los símbolos musicales.

Los símbolos que tienen zonas que son tangentes a los pentagramas, tales como blancas y redondas ubicadas en los espacios, las ligaduras, los bemoles y becuadros ubicados en los espacios, son parcialmente destruidos, por lo que es muy importante tener en cuenta este aspecto en la etapa de segmentación (sección 5.4). En la figura 5.12 se observa el resultado de la eliminación del pentagrama de una partitura. Observe cómo algunos símbolos (clave, indicador de compás, nota blanca y redonda) quedan parcialmente destruidos después de realizar este proceso.

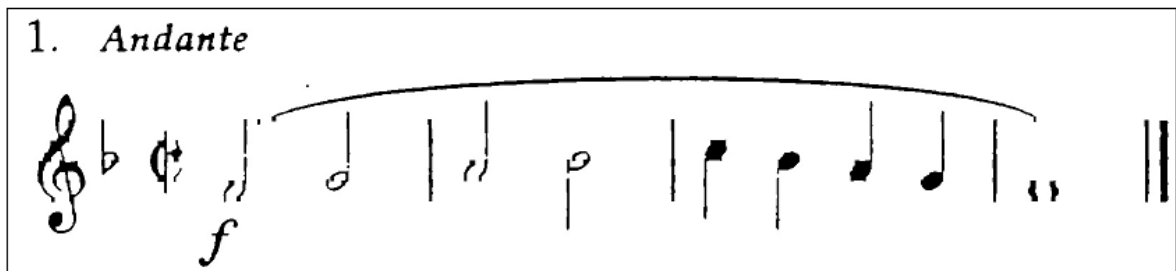


Figura 5.12 Eliminación del pentagrama

### 5.4. Segmentación y Reconocimiento

La entrada a este módulo es una imagen con símbolos aislados como la que se muestra en la figura 5.12. El objetivo de este módulo es identificar cada uno de los símbolos y hacer una interpretación de ellos.

En el sistema desarrollado en este trabajo el reconocimiento de objetos se lleva a cabo paralelamente con el proceso de segmentación, es decir, el algoritmo va segmentando y reconociendo símbolo por símbolo, lo que es posible gracias a las características de las imágenes que se van a tratar. Además, el orden de reconocimiento de los símbolos es muy importante para obtener una buena interpretación de la partitura.

De forma general, el proceso de segmentación y reconocimiento se realiza de la siguiente manera: primero se detectan y eliminan las ligaduras, la clave, la armadura y el indicador de compás.

Luego, se ubican las zonas de los compases, para lo cual es necesario ubicar las barras divisorias de compás, y se procede con el reconocimiento de los símbolos musicales, el cual está controlado totalmente por una gramática, es decir, el uso de las reglas de la música, con el fin de hacer una predicción del objeto que va a ser identificado.

Los conceptos básicos para clasificar los símbolos como son la altura, el ancho, el tamaño de los objetos y el área de zonas específicas de los objetos.

El módulo de segmentación y reconocimiento está dividido en las siguientes etapas: rotación de la imagen, selección del área de análisis, etiquetado de los objetos, detección y eliminación de las ligaduras, reconocimiento de la clave, reconocimiento de la armadura, reconocimiento del indicador de compás, reconocimiento de las barras de compás y reconocimiento de los símbolos musicales.

Toda la información del reconocimiento se almacena en una matriz, que será la entrada al módulo denominado interpretación de la partitura, descrito en la sección 5.5, el cual se encargará de reproducir la partitura.

#### **5.4.1. Rotación de la imagen**

El sistema desarrollado en este trabajo trata con partituras rotadas, es decir aquellas donde las líneas de los pentagramas tienen cierta inclinación con respecto a la horizontal, debido a que no se tiene cuidado al escanear las imágenes. En esta etapa se trata dicha inclinación, haciendo uso del algoritmo de rotación descrito en la sección 3.2.7, teniendo como ángulo de rotación el calculado en el módulo de detección de pentagramas (sección 5.3.1).

La utilización de técnicas de proyección vertical para la segmentación de símbolos requiere de una buena rotación de las partituras por lo que es necesario que esta etapa se realice eficazmente.



#### 5.4.4. Detección y eliminación de las ligaduras

Las ligaduras interfieren el proceso de segmentación y reconocimiento de símbolos musicales (sección 5.4.10) por lo tanto, son detectadas y eliminadas (Figura 5.14). Aunque las ligaduras son detectadas, en este trabajo no son tenidas en cuenta en el proceso de interpretación de la partitura.

Las ligaduras se caracterizan por ser objetos bastante largos horizontalmente y bastante delgados verticalmente, por lo que un análisis específico a cada objeto etiquetado permite su precisa identificación.

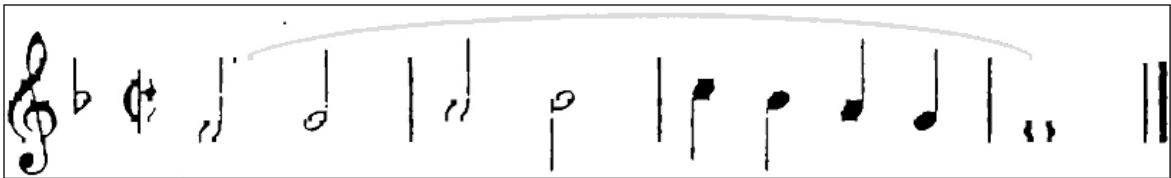


Figura 5.14 Detección y eliminación de las ligaduras

Hay que realizar un tratamiento especial con las ligaduras que aparecen rotas y las que se superponen a los símbolos, ya sea por la mala calidad de las partituras o por su alteración en una etapa anterior, este procedimiento se explica en el capítulo de resultados. El sistema implementado garantiza la detección de todas las ligaduras con longitudes horizontales mayores a 2E.

#### 5.4.5. Reconocimiento de la clave

La clave es el primer símbolo que se encuentra en un pentagrama, por lo tanto su localización se vuelve más fácil. La utilización de las reglas de sintaxis musical ayudan a la identificación de ciertos símbolos, por ejemplo, la clave se encuentra siempre al inicio de cada pentagrama, la armadura (sección 5.4.6) se ubica siempre después de la clave y el indicador de compás (sección 5.4.7) se ubica siempre después de la armadura. La información de la clave es muy importante a la hora de asignar el tono a las notas por lo que se guarda su información.

Para la segmentación de la clave se hace un análisis en el histograma de proyección vertical, considerando los símbolos que cumplen ciertos umbrales (verticales y horizontales) de longitud.

Para el reconocimiento de la clave se vale de las notables diferencias que se presentan en estos símbolos, por ejemplo, la clave de sol es la de mayor longitud vertical y la clave de fa contiene dos puntos al final, entre otras.

#### **5.4.6. Reconocimiento de la armadura**

La armadura al igual que la clave es fácil de localizar usando reglas sintácticas. Puede estar conformada por sostenidos o por bemoles y solo un tipo a la vez. Hay que tener en cuenta que no todas las partituras poseen sostenidos o bemoles en la armadura, en cuyo caso el siguiente símbolo a la clave es el indicador de compás. La información de la armadura ayuda a asignar el tono a las notas afectadas, por lo tanto también se guarda esta información.

La segmentación y el reconocimiento se realizan haciendo uso del histograma de proyección vertical, en el cual los bemoles aparecen con un pico y los sostenidos con dos. Además, sus longitudes horizontales son considerablemente diferentes, siendo el sostenido más ancho que el bemol.

#### **5.4.7. Reconocimiento del indicador de compás**

El indicador de compás es el siguiente símbolo en aparecer después de la armadura. La segmentación y el reconocimiento de este símbolo se basan en el análisis de los histogramas de proyección horizontal y vertical del objeto encontrado. Para este propósito se hacen ciertas mediciones de longitudes y áreas específicas, así como de picos presentes en el objeto.

Las semejanzas de los indicadores de compás, especialmente  $2/4$ ,  $3/4$  y  $4/4$ , y las rupturas de estos, ocurridas en la etapa de eliminación de pentagramas, hacen que el análisis se complique un poco.

#### **5.4.8. Eliminación de la clave, la armadura y el indicador de compás**

Una vez que la clave, la armadura y el indicador de compás se localizan y reconocen, el siguiente paso es su eliminación. Esto es importante ya que reduce la cantidad de información a ser procesada en las siguientes etapas. Esta eliminación debe realizarse en todos los pentagramas de la partitura (Figura 5.15). Un aspecto a tener en cuenta es que el indicador de compás sólo aparece en el primer pentagrama de la partitura, mientras que la clave y la armadura aparecen en todos los pentagramas.



El sistema implementado no garantiza el reconocimiento de la clave, la armadura y el indicador de compás, si estos aparecen en lugares que no sea el principio del pentagrama, como suele ocurrir en cierto de tipo de composiciones cuando hay cambio de clave o de armadura.

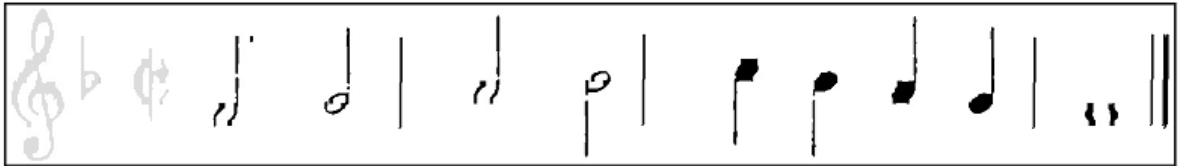


Figura 5.15 Eliminación de la clave, la armadura y el indicador de compás

#### 5.4.9. Reconocimiento de las barras divisorias de compás

El siguiente paso dentro del proceso de reconocimiento, es la detección de las barras divisorias de compás. Este paso se realiza mediante un análisis del histograma de proyección vertical (sección 3.3.3.1) del pentagrama, en el cual se buscan los picos que estén dentro del rango  $(4E+5L) \pm 2L$ . Las coordenadas de las barras divisorias detectadas se almacenan y se utilizan en la siguiente etapa.

Observe en la figura 5.16 las barras de compás detectadas y las coordenadas que determinan sus posiciones.

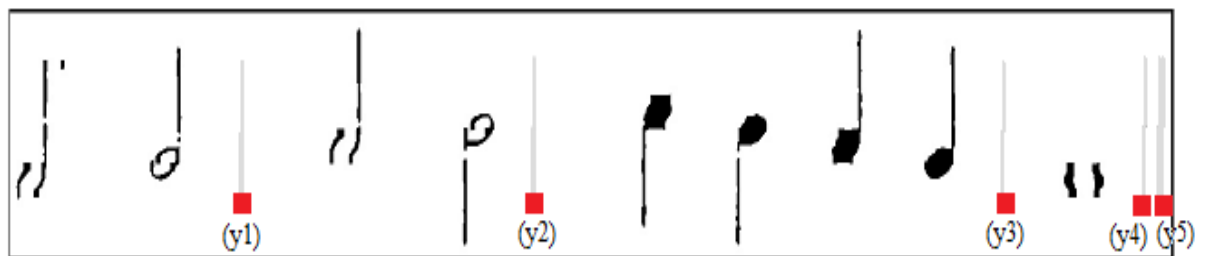


Figura 5.16 Detección de las barras divisorias de compás

#### 5.4.10. Segmentación y reconocimiento de notas, silencios, alteraciones y puntillos

Esta etapa es una de las más complejas ya que la variedad de los símbolos es muy amplia, además, el orden de aparición de éstos en el pentagrama es casi que aleatoria.

El conocimiento de la gramática musical y de las reglas musicales es de gran ayuda en esta etapa, ya que garantizan que la aparición de los símbolos está dentro de cierto rango de probabilidad, por ejemplo, la aparición de un puntillo solo tiene sentido si el símbolo anterior es una nota y la posición vertical de la cabeza de la nota y de aquel son similares.

El uso de la gramática musical y de las reglas de la música mejora la efectividad del sistema, ya que ayudan a detectar errores de reconocimiento y en algunos casos a corregirlos. Sin embargo, su uso requiere un algoritmo más complejo y mayor tiempo de ejecución.

Las reglas musicales y de gramática utilizadas en el sistema implementado en este trabajo son las siguientes:

1. Los símbolos se han clasificado en cuatro grande grupos:
  - (a) Notas (**N**), incluye la redonda, la blanca, la negra, la corchea, la semicorchea y grupos de éstas.
  - (b) Silencios (**S**), incluye el silencio de blanca, de negra y de corchea.
  - (c) Alteraciones (**A**), incluye el bemol, el sostenido y el becuadro.
  - (d) Puntillo (**P**).

La gramática me informa que la aparición de un símbolo musical está determinada por el símbolo anterior. Partiendo de este hecho, se han definido las siguientes reglas:

El primer símbolo en aparecer puede ser **N**, **S** o **A**. A partir de entonces la aparición de los símbolos está determinada por el primer símbolo.

Si el objeto es **N**, el siguiente puede ser **N**, **S**, **A** o **P**.

Si el objeto es **S**, el siguiente puede ser **N**, **S**, **A** o **P**

Si el objeto es **A**, el siguiente solo puede ser **N**.

Si el objeto es **P**, el siguiente puede ser **N**, **S** o **A**.

2. El puntillo solo puede aparecer a la derecha de la cabeza de la nota afectada.

La segmentación de los símbolos se realiza analizando el histograma de proyección vertical. El proceso básicamente es el siguiente: se detecta el primer objeto en el histograma y se analiza. Si cumple con ciertas condiciones de dimensión es considerado un posible símbolo musical y pasa a una etapa de análisis más profundo para su reconocimiento, si no las cumple, entonces se considera ruido y se sigue con el siguiente objeto. Este procedimiento funciona solamente si los símbolos están completos y todas las partes que lo conforman están unidas.

Como se mencionó en la sección 5.3.2, cuando se realiza la eliminación de los pentagramas algunos símbolos son parcialmente destruidos. Esto hay que tenerlo en cuenta al momento de la segmentación, ya que ciertos objetos que en un principio pueden parecer ruido suelen ser partes de un símbolo que ha sido roto, observe en la figura 5.17 algunos símbolos afectados.



Figura 5.17 Símbolos parcialmente destruidos en el proceso de eliminación de los pentagramas

Este proceso de segmentación y reconocimiento de los símbolos se realiza compás por compás hasta terminar todos los pentagramas y la información de cada uno se va guardando en la matriz de interpretación de la partitura. La información guardada consiste de dos datos para el caso de las notas, tono y duración, y de un dato para el caso de los silencios, duración. Las alteraciones son asignadas a las notas en la etapa siguiente y los puntillos se convierten en notas con tono y duración que dependen de la nota anterior ya reconocida.

## 5.5. Interpretación de la partitura

Una vez se tiene la información de los símbolos existentes en cada compás de cada pentagrama, el siguiente paso es la interpretación de la partitura, es decir, la representación mediante sonidos MIDI.

El objetivo de este trabajo está encaminado a la utilización del sistema para el aprendizaje de la música, concretamente en la lectura de partituras, por lo tanto es necesario que la representación final sea la ejecución de la partitura. Por este motivo se ha decidido representar la partitura haciendo uso del formato MIDI, que además de ser uno de los formatos más comunes para representar partituras, permite al usuario escoger el timbre de los sonidos y la velocidad de interpretación.

Antes de realizar este proceso es necesario asignar las alteraciones, tanto las que indica la armadura como las encontradas en cada compás. Aunque el sistema es capaz de reconocer los tres tipos de alteraciones, sostenidos, bemoles y becuadros, en la asignación solo se tienen en cuenta los sostenidos y los bemoles.

Igualmente, se realizan las asignaciones de los puntillos. Para esto se inserta en la matriz un nuevo símbolo con tono igual al del símbolo afectado y con duración igual a la mitad de la duración del mismo.

Una vez hecho lo anterior, se implementó una función software que reproduce los símbolos reconocidos por el sistema, haciendo uso del formato MIDI, la cual le permite escoger al usuario entre 17 voces (sonidos de instrumentos) y 6 velocidades de reproducción (ver Anexo A). Para la implementación se hizo uso de las funciones de multimedia de la API de win32 [66].

## CAPÍTULO 6. RESULTADOS Y ANÁLISIS DE RESULTADOS

### 6.1. Introducción

En este capítulo se darán a conocer los resultados obtenidos con el sistema implementado (Capítulo 5). Primero se darán los resultados y análisis para cada etapa del sistema, posteriormente, se hará una evaluación general del sistema y finalmente se comentará la eficiencia del sistema en términos del tiempo de ejecución.

### 6.2. Evaluación de los resultados

#### 6.2.1. Preprocesamiento

La etapa de preprocesamiento comentada en la sección 5.2 se realizó de manera empírica. Se tomaron partituras de distintas calidades y se le aplicaron varios algoritmos, entre ellos, filtros Laplacianos, filtros de Sobel, filtros de mediana, filtros de paso bajo y filtros de paso alto. Posteriormente se analizaron los resultados obtenidos y se determinó que el que mejor respondía a los objetivos propuestos eran los filtros de paso bajo de la sección 5.2.1.

En las figuras 6.1 a 6.3 se pueden observar algunos resultados obtenidos con la implementación de este módulo.



Figura 6.1 Partitura No 1. (a) Antes y (b) Después del preprocesamiento

a) 12. *Allegro deciso*

b) 12. *Allegro deciso*

Figura 6.2 Partitura No 12. (a) Antes y (b) Después del preprocesamiento

a) 40. *Vivace*

b) 40. *Vivace*

Figura 6.3 Partitura No 40. (a) Antes y (b) Después del preprocesamiento

Como se puede observar en estos tres ejemplos los resultados son muy buenos ya que se cumplió con el objetivo que era resaltar las zonas degradadas, eliminar las zonas con ruido y binarizar la imagen sin perder información de importancia.

La función de preprocesamiento se aplicó a 110 partituras y se encontró que los sostenidos de algunas partituras fueron deformados, como se puede observar en el caso 1 de la figura 6.4.

Los sostenidos están compuestos por dos barras verticales paralelas ligeramente separadas, por lo que su identificación se realiza detectando el número de picos en el histograma de proyección vertical superiores a cierto umbral.

Mientras los picos del sostenido del caso 2 de la figura 6.4 se diferencian claramente, los picos del sostenido del caso 1 aparecen como uno solo.

Por otro lado, las cabezas de las notas negra y corchea suelen estar parcialmente borrosas como se puede observar en las figuras 6.1 a 6.3. El módulo de preprocesamiento responde muy bien en estos casos ya que estas zonas grises y blancas son rellenadas, dejando una nota en perfecto estado, es decir, totalmente negra (Figura 6.5.a).

Cuando la borrosidad de las cabezas de las notas es mayor, el preprocesamiento no funciona muy bien y ocurre que las notas que originalmente son negras parecen blancas incluso para un observador humano (Figura 6.6).

El filtro morfológico aumenta el grosor de las líneas delgadas tales como las plicas de las notas y las barras de compás. En este caso el resultado es muy beneficioso ya que facilita la identificación de las líneas (Figura 6.5.b).

### **6.2.2. Pentagramas**

La identificación de los pentagramas es muy importante ya que de su correcta detección depende el resto del sistema. En la figura 6.7 se observa una buena detección de los pentagramas y en la figura 6.8 se observa una mala detección de los pentagramas.

En la tabla 6.1 se observan los resultados de aplicar el algoritmo de detección de pentagramas a 110 partituras. El porcentaje de detección fue del 90.84%.

Como se mencionó en la sección 5.3.1 el algoritmo de detección busca segmentos verticales que tengan una longitud más o menos igual al grosor de las líneas de pentagrama. Sin embargo, hay símbolos tales como las ligaduras que contienen zonas con una longitud más o menos igual al grosor de las líneas del pentagrama por lo que son considerados puntos candidatos a pentagrama.

Por otro lado, el algoritmo enlaza los puntos candidatos basándose en el hecho de que dos puntos candidatos horizontalmente contiguos deben estar unidos por una línea recta con al menos 90% de píxeles negros. Sin embargo, existen zonas que aunque no corresponden a pentagramas cumplen esta condición y por lo tanto son considerados como tal.

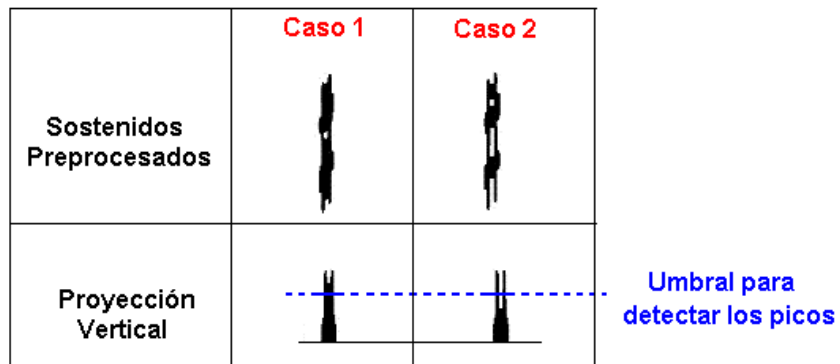


Figura 6.4 Determinación de los picos de los sostenidos

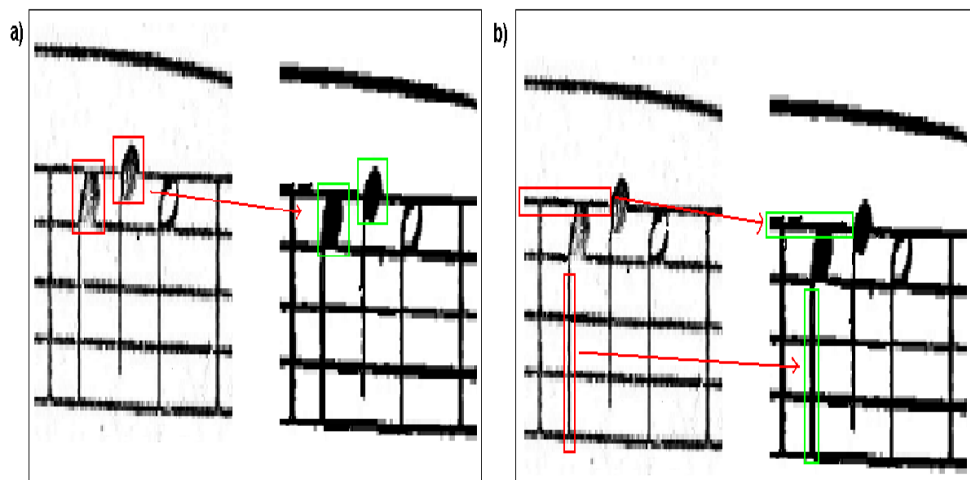


Figura 6.5 Preprocesamiento: a) Cabeza de nota rellenada. b) Aumento del grosor de líneas delgadas

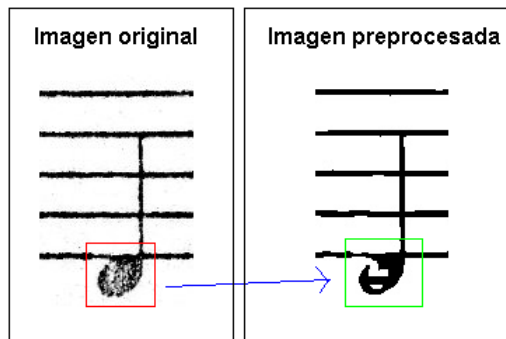


Figura 6.6 Mal preprocesamiento en la cabeza de la nota negra



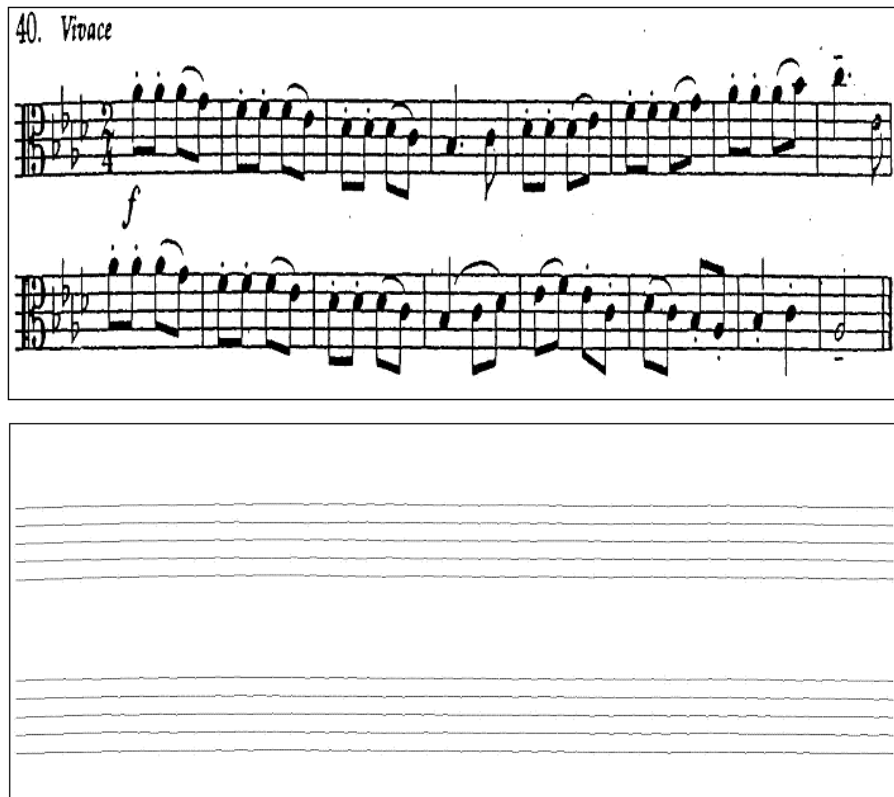


Figura 6.7 Detección de los pentagramas en la partitura No 40



Figura 6.8 Detección de los pentagramas en la partitura No 27

Número real de pentagramas	Número de pentagramas detectados	Porcentaje de detección de pentagramas (%)
153	139	90.84

Tabla 6.1 Resultados de la detección de pentagramas sobre 110 partituras

### 6.2.3. Área de análisis

La selección del área de análisis se realizó considerando de manera empírica cierta zona en la que existe mayor probabilidad de encontrar los símbolos.

El algoritmo se probó sobre las 78 partituras que tuvieron éxito en la detección de los pentagramas, esto da un total de 94 pentagramas.

Se presentaron tres casos particulares con la aplicación del algoritmo. En el primero (Figura 6.9) ocurre una excelente detección del área de análisis, en él se excluyen los símbolos sin información importante, es decir, el número de la partitura y los indicadores de movimiento y expresión y algunos puntos aislados.

En el segundo caso (Figura 6.10) se presenta una falla en la selección del área de análisis. El error ocurre al momento de la exclusión de los símbolos ya que el algoritmo deja por fuera del área de análisis ciertas notas. En las 78 partituras analizadas se encontró que las notas más propensas a ser excluidas son las redondas y los puntillos cuando se encuentran por fuera del pentagrama a una distancia mayor a  $1E+1L$  de la línea de borde superior del pentagrama o  $3/2E + 1L$  de la línea de borde inferior del pentagrama.

En el tercer caso (Figura 6.11) se presenta una falla en la selección del área de análisis. El error ocurre al momento de la inclusión de los símbolos dentro del área ya que el algoritmo incluye ciertos símbolos (indicadores de expresión) como símbolos de importancia. Estos símbolos generalmente se ubican fuera del pentagrama en la parte inferior. En algunos casos estos símbolos se encuentran demasiado cerca al pentagrama como para que el algoritmo los detecte como símbolos de importancia.

19. *Allegretto*

*p* *f*

símbolos sin información importante

Figura 6.9 Selección del área de análisis. Caso 1

33a. *Moderato*

símbolo sin importancia

nota excluida del área de análisis

Figura 6.10 Selección del área de análisis. Caso 2

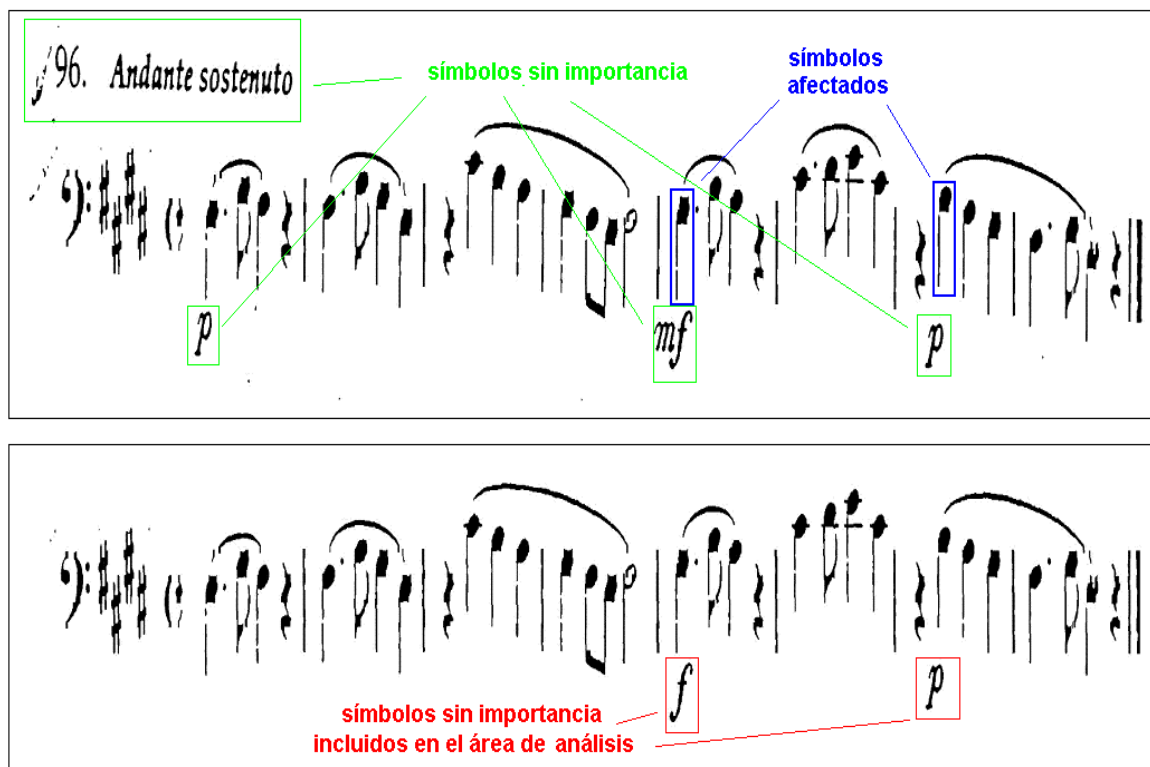


Figura 6.11 Selección del área de análisis. Caso 3

Los errores de los casos 2 y 3 se propagan a lo largo del sistema y afectarán drásticamente la etapa de segmentación y reconocimiento de notas y puntillos.

En el caso 2, cuando el algoritmo deja por fuera notas como la redonda o los puntillos, la etapa de segmentación simplemente pasará por alto estos símbolos ya que no los va a encontrar en la zona de análisis.

Por otro lado, en el caso 3, cuando el algoritmo incluye signos de expresión, la etapa de segmentación detectará mal los símbolos afectados que estén por encima de estos signos y por consiguiente los reconocerá mal. Por lo general, siempre ocurre que el sistema identifica estos símbolos como corcheas, independientemente de cuáles sean.

En la tabla 6.2 se observan los resultados de la aplicación del algoritmo a las 78 partituras.

Número de Partituras	Número total de Pentagramas	Caso 1		Caso 2		Caso 3	
		No. Pentagramas	% de aparición	No. Pentagramas	% de aparición	No. Pentagramas	% de aparición
78	94	80	85.10	9	9.57	7	7.44

Tabla 6.2 Resultados de la selección del área de análisis para 78 partituras

#### 6.2.4. Ligaduras

En este trabajo no se tienen en cuenta las ligaduras para la interpretación de la partitura (sección 5.5), sin embargo, es necesario identificarlas y eliminarlas ya que de no hacerlo interferirían drásticamente el proceso de segmentación de los objetos el cual está basado en el análisis del histograma de proyección vertical.

Algunas de las ligaduras son eliminadas en la selección del área de análisis por encontrarse alejados del borde del pentagrama como se observa en la figura 6.10.

Las ligaduras restantes se detectan y se eliminan con el algoritmo implementado. Para su detección se analiza la imagen etiquetada y se consideran los objetos que poseen una gran longitud horizontal y una pequeña longitud vertical.

En la figuras 6.12, 6.13 y 6.14 se pueden observar los tres casos presentados en la detección de ligaduras aplicados a las 78 partituras.

En el primer caso (Figura 6.12) se observa una buena detección y eliminación de las ligaduras.

En el segundo caso se observa una particularidad y es que la ligadura aparece rota, ruptura ocasionada en la etapa de eliminación del pentagrama. Cuando se presenta este caso el algoritmo detecta y elimina la parte de la ligadura con mayor longitud y deja la parte con menor longitud intacta (Figura 6.13). Este caso se debe tener en cuenta en el reconocimiento del símbolo afectado, es decir el que está por debajo o por encima del pequeño segmento de ligadura.

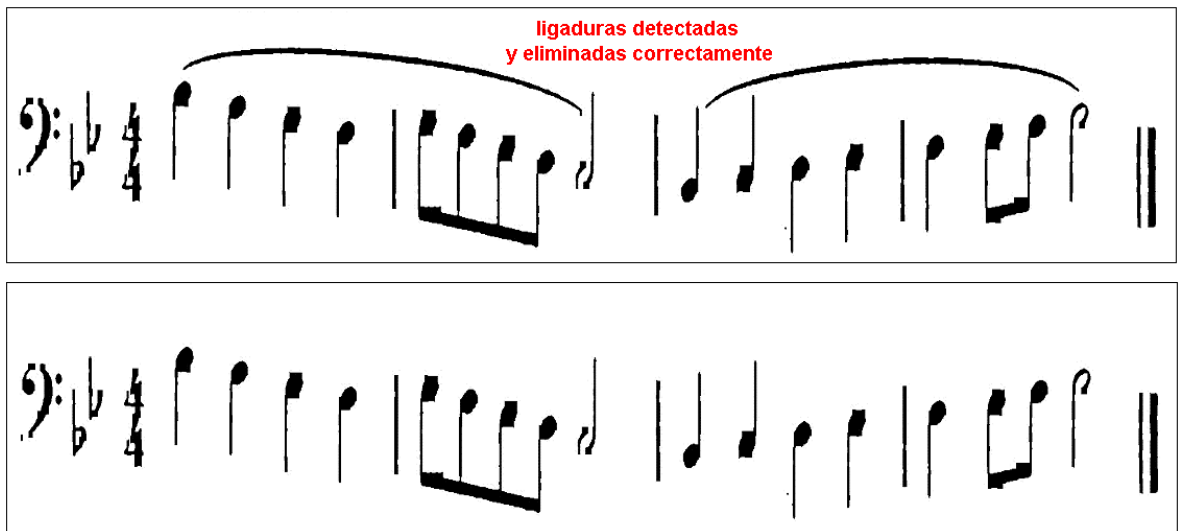


Figura 6.12 Detección de las ligaduras. Caso 1

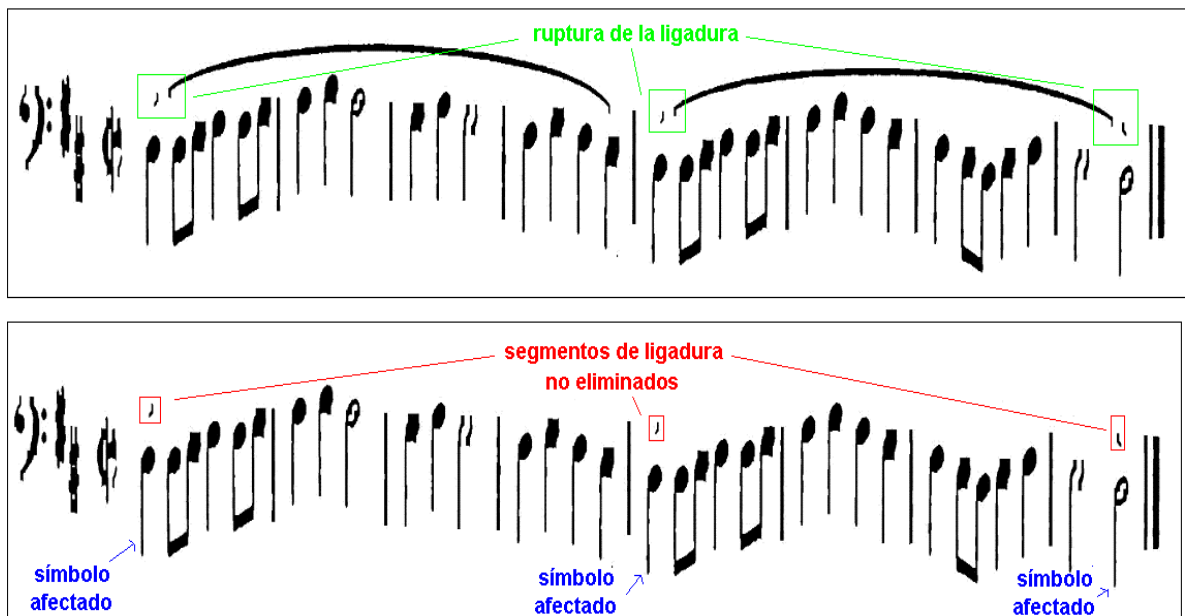


Figura 6.13 Detección de las ligaduras. Caso 2

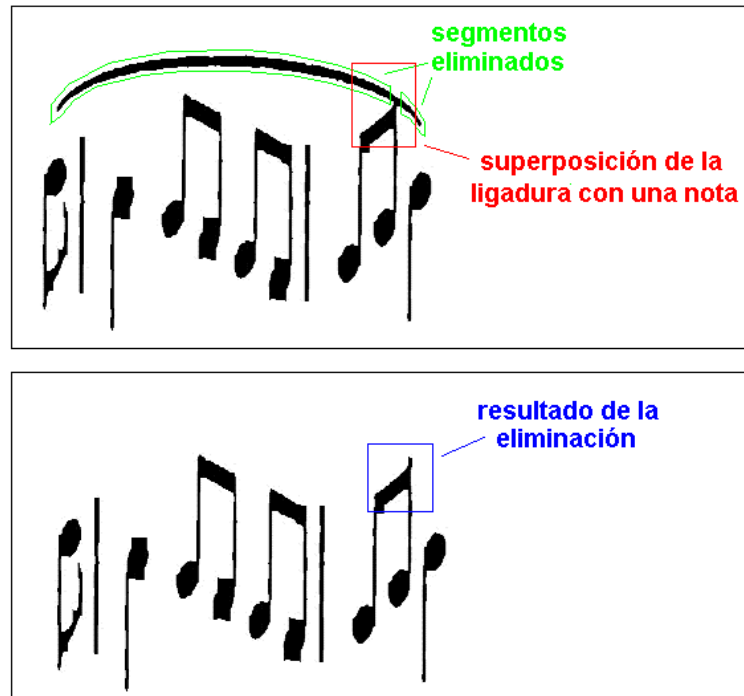


Figura 6.14 Detección de las ligaduras. Caso 3

En el tercer caso aparece una ligadura superpuesta a un símbolo. La borrosidad de las partituras empleadas hace que ciertos símbolos se junten en la binarización. Para corregir este error se eliminaron solamente aquellos segmentos verticales que tenían un grosor menor o igual a  $3L$  (Figura 6.14).

En la tabla 6.3 se observan los resultados de aplicar el algoritmo a las 78 partituras.

Número de Partituras	Número real de ligaduras	Número de ligaduras detectadas	Porcentaje de detección (%)
78	277	271	97.83

Tabla 6.3 Resultados de la detección de ligaduras

### 6.2.5. Clave

Las reglas de la música me indican que al comienzo de toda partitura se pone la clave. Este hecho facilita el proceso de detección de este símbolo. El sistema es capaz de detectar y reconocer tres tipos de claves: clave de sol en segunda línea, clave de fa en cuarta línea y clave de do en tercera línea. En la tabla 6.4 se observan los resultados del reconocimiento para las 78 partituras. Los resultados fueron excelentes ya que se obtuvo un 100% de detección.

Por lo general estos símbolos fueron poco afectados por las etapas anteriores y el algoritmo de detección y reconocimiento fue capaz de superar las alteraciones que en estos símbolos pudieron ocurrir.

<b>Número de partituras</b>	<b>Número real de claves</b>	<b>Número de claves detectadas</b>	<b>Porcentaje de detección (%)</b>
78	78	78	100

Tabla 6.4 Resultados de la detección de la clave

### 6.2.6. Armadura

La armadura es un conjunto de alteraciones (sostenido o bemol) que se ubica siempre después de la clave. Para su identificación se analiza el histograma de proyección vertical. Los resultados obtenidos con 78 partituras se muestran en la tabla 6.5. Se obtuvo un 97.43% de detección, por lo que el algoritmo funciona muy bien.

<b>Número de Partituras</b>	<b>Número real de armaduras</b>	<b>Número de armaduras detectadas correctamente</b>	<b>Porcentaje de detección (%)</b>
78	78	76	97.43

Tabla 6.5 Resultados de la detección de la armadura



Hay un caso muy particular que hay que tener en cuenta y es cuando el algoritmo no detecta un sostenido de la armadura. Observe la clave, el ruido y el sostenido de la Figura 6.15. Estos tres objetos son detectados como uno solo en el histograma de proyección vertical. El algoritmo reconoce este conjunto como clave de sol y por consiguiente ocurre la pérdida de un sostenido. El sistema no soluciona este tipo de errores.

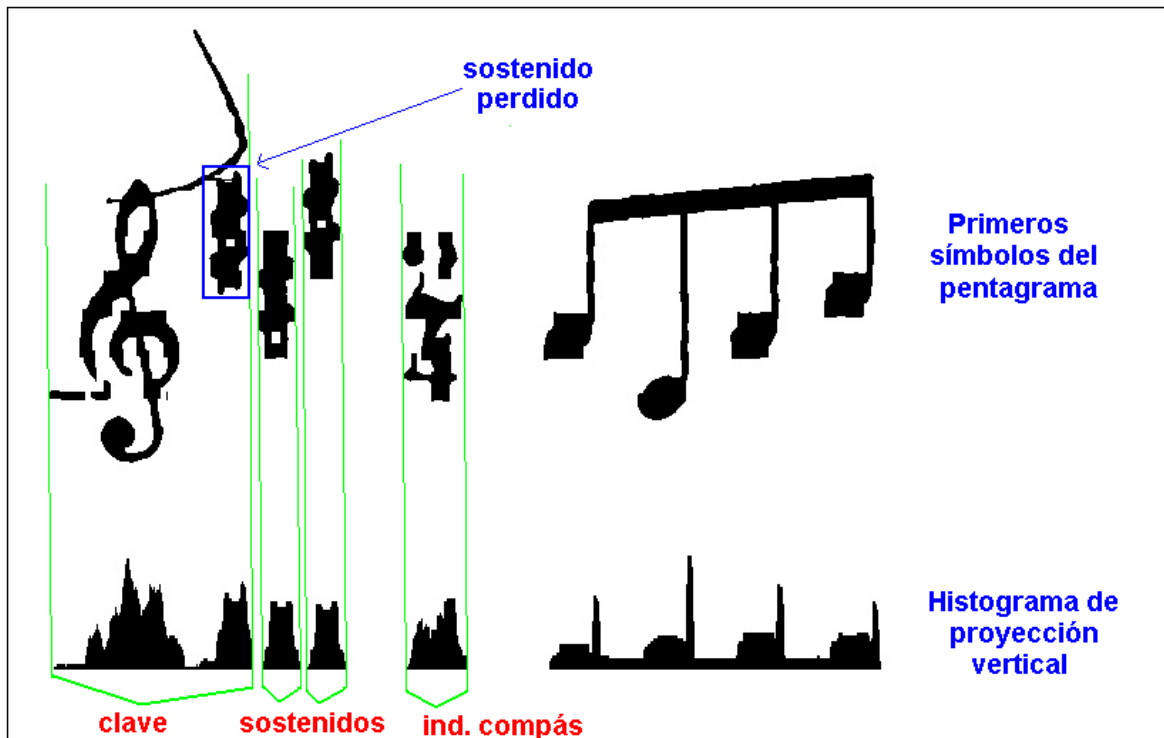


Figura 6.15 Error en la detección de la armadura.

Los resultados de la detección de la armadura son bastante buenos y el sistema puede detectar hasta cuatro alteraciones en la armadura y funciona muy bien cuando no existen alteraciones en ella.

### 6.2.7. Indicador de compás

El indicador de compás es el siguiente símbolo después de la armadura. Este símbolo fue un poco difícil de reconocer, especialmente para 2/4, 3/4 y 4/4, por su

gran parecido. No obstante los resultados fueron bastante buenos y se pueden apreciar en la tabla 6.6.

No de Partituras	Número real de Indicadores de Compás	Número de Indicadores de Compás detectados correctamente	Porcentaje de detección (%)
78	78	77	98.71

Tabla 6.6 Resultados de la detección del indicador de compás

### 6.2.8. Barras de compás

Las barras de compás son fácilmente confundibles con las plicas de las notas ya que ambas poseen las mismas características: son horizontalmente delgadas, aproximadamente 2L y verticalmente largas, aproximadamente 4E + 5L. El algoritmo para su identificación se basa en el hecho de que su posición debe estar lo más cercana posible al centro del pentagrama (Figura 6.16). Aunque se obtuvieron muy buenos resultados, hubo casos en los que el algoritmo confundió la plica de una nota con una barra de compás (Figura 6.17). Observe como esta plica ha sido separada de la cabeza de la nota blanca en el proceso de eliminación de las líneas del pentagrama y ha quedado parecida a una barra de compás.

El porcentaje de reconocimiento fue de 98.57% y se puede observar en la tabla 6.7.

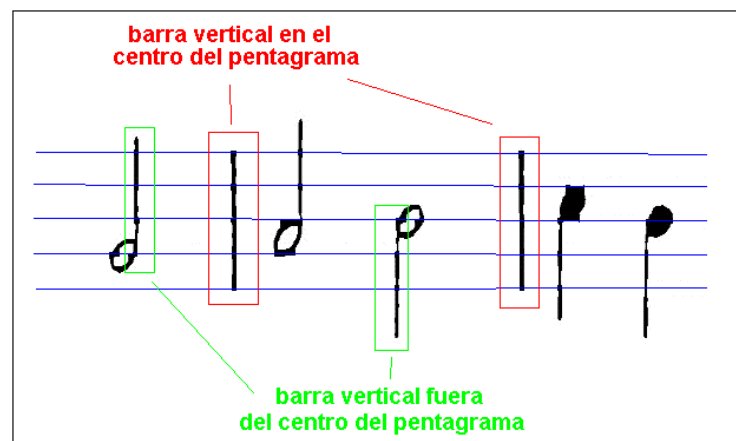


Figura 6.16 Posición de las barras de compás dentro del pentagrama

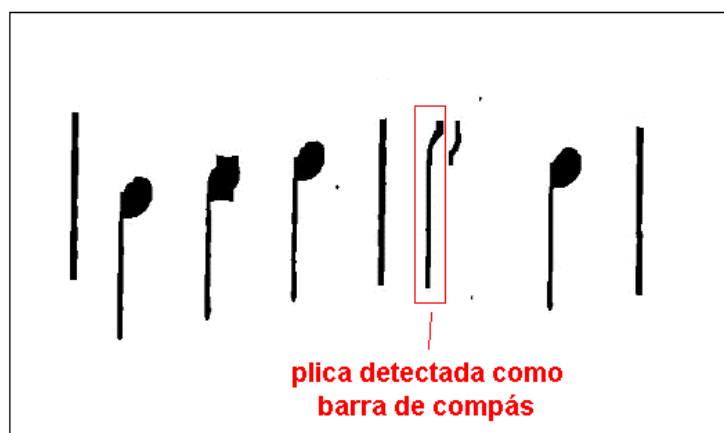


Figura 6.17 Error en la detección de las barras de compás

Número de Partituras	Número real de Barras de Compás	Número de Barras de Compás detectadas correctamente	Porcentaje de detección (%)
78	420	414	98.57

Tabla 6.7 Resultados de la detección de las barras de compás

### 6.2.9. Notas, silencios, alteraciones y puntillos

El reconocimiento de las notas, silencios, alteraciones y puntillos está controlado por una gramática (Sección 5.4.10).

Hay dos tipos de errores en el reconocimiento de estos símbolos. El primero ocurre cuando el sistema reconoce mal un símbolo, es decir, le asigna la etiqueta equivocada. Para las notas se pueden dar tres casos, que el sistema reconozca bien la figura de la nota pero mal el tono, que el sistema reconozca bien el tono pero mal la figura o que el sistema reconozca mal el tono y la figura.

En la figura 6.18 se observa un error de reconocimiento típico. El algoritmo le asigna la etiqueta “negra” a una nota blanca, ya que como se puede apreciar la cabeza de esta nota está más rellena que lo normal. El funcionamiento del algoritmo consiste en calcular el área dentro del cuadro rojo, el cual tiene dimensiones  $E \times E$ , y determinar mediante un umbral si es una cabeza negra o blanca. Para la selección del umbral se debe llegar a un balance entre lo que es

considerado una nota negra y una nota blanca. Esta decisión que en un principio parece fácil se complica cuando se tiene el conocimiento de que hay cabezas de negras que no están completamente rellenas, como se vio en la figura 6.6.



Figura 6.18 Error de reconocimiento de la nota blanca

Otro error típico ocurre cuando el sistema le asigna a una nota negra la etiqueta de "corchea". Como se mencionó anteriormente el algoritmo de selección del área de análisis suele incluir signos de expresión y ya que el algoritmo de segmentación busca en el histograma de proyección vertical zonas separadas por espacios vacíos ocurre que este signo junto con el símbolo afectado (Figura 6.19) es considerado por el sistema como uno solo. El ancho del símbolo aumenta notablemente llegándose a parecer al ancho de una corchea, por lo tanto el sistema reconoce mal este símbolo.

Otro error que cabe dentro de este tipo es el que ocurre cuando el sistema confunde un sostenido con un bemol. El algoritmo diferencia estas dos alteraciones según el número de picos que contengan en su histograma de proyección vertical. Si el símbolo tiene un pico se reconoce como bemol, si el símbolo tiene dos picos se reconoce como sostenido. Como se vio en la etapa de procesamiento suele ocurrir que los sostenidos aparecen con un pico en su histograma (Figura 6.4), hecho que explica el error de reconocimiento.

El segundo tipo de error ocurre cuando el sistema no detecta un símbolo. Este tipo de error se suele dar cuando el sistema en una etapa anterior por alguna razón elimina el símbolo por lo tanto, cuando se llega a este punto del proceso, simplemente el símbolo no aparecerá en el histograma de proyección vertical y por consiguiente el algoritmo de segmentación no lo detectará.

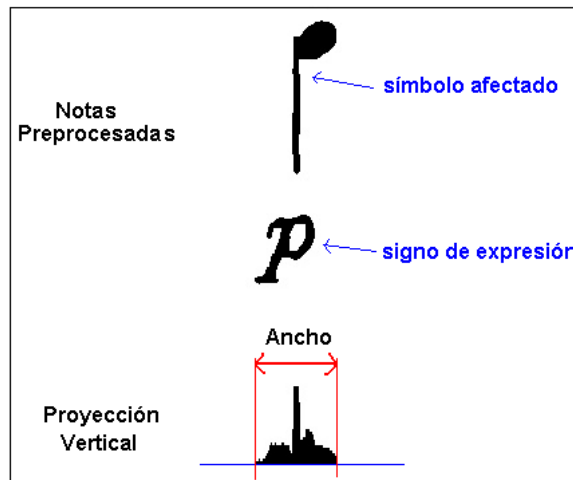


Figura 6.19 Error en el reconocimiento de una nota afectada por un signo de expresión

Los puntillos por fuera del pentagrama son los más propensos a ser eliminados en la etapa de selección del área de análisis por estar aislados es decir, no tener conexión con el pentagrama. Igualmente ocurre con las redondas cuando se encuentran alejadas del pentagrama, ya sea por encima o por debajo de éste.

En las tablas 6.8 se pueden apreciar los resultados del reconocimiento de notas, silencios, alteraciones y puntillos para las 78 partituras.

Símbolo	Número real de símbolos	Número de símbolos reconocidos correctamente	Porcentaje de detección (%)
<b>Notas</b>	2073	2002	96.57
<b>Silencios</b>	39	38	97.43
<b>Alteraciones</b>	29	25	86.20
<b>Puntillos</b>	113	99	87.61

Tabla 6.8 Resultados del reconocimiento de Notas, silencios, alteraciones y puntillos

### 6.2.10. Evaluación general

Como se vio anteriormente el resultado final del sistema depende de la efectividad de cada módulo, ya que el error cometido en una fase primera se propagará hasta el final, causando errores en fases posteriores.

Sin embargo, se debe dar una evaluación total del sistema ya que al usuario le interesa es que el sistema sea capaz de cumplir con su objetivo y no el resultado de cada módulo.

Hay que tener en cuenta que para hablar de un rendimiento general es necesario considerar el estado original de las partituras que fueron analizadas. Al respecto, las partituras no estaban en muy buen estado ya que presentaban cierto grado de deterioro debido a la manipulación.

Además, las partituras presentaban manchas de tinta, deformaciones en la imagen, huellas de lápiz, factores que en cierto modo disminuían la eficacia en algún punto del sistema, ya sea en el preprocesamiento, la detección de los pentagramas, la segmentación o el reconocimiento de los símbolos.

Si consideramos todos los símbolos presentes en la partitura y calculamos el porcentaje de efectividad obtenemos los valores de la Tabla 6.9. Los símbolos tenidos en cuenta son los propuestos en la sección 5.1.1.

### 6.3. Eficiencia del sistema

De cara al usuario es importante hacer un análisis de la eficiencia del sistema y dar valores cuantitativos del tiempo de ejecución.

El escaneo de las partituras se realizó en un escáner Epson Stylus TX200 con el software EPSON Scan con los siguientes de datos de configuración:

- **Tipo de imagen:** Grises 8 bits.
- **Resolución:** 400 ppp (píxeles por pulgada)
- **Escala:** 100 %

El tiempo promedio de escaneo por partitura fue de 40 segundos.

<b>Símbolo</b>	<b>Número real de símbolos</b>	<b>Número de símbolos detectados correctamente</b>	<b>Porcentaje de reconocimiento (%)</b>
<b>Clave</b>	78	78	100
<b>Tonalidad</b>	78	76	97.43
<b>Indicador de Compás</b>	78	77	98.71
<b>Barras de Compás</b>	420	414	98.57
<b>Ligaduras</b>	277	271	97.83
<b>Notas</b>	2073	2002	96.57
<b>Silencios</b>	39	38	97.43
<b>Puntillos</b>	29	25	86.20
<b>Alteraciones</b>	113	99	87.61
<b>Total</b>	3185	3080	95.60

Tabla 6.9 Porcentaje de efectividad del sistema

El computador empleado tanto para la implementación del sistema como para las pruebas tiene las siguientes características:

- Sistema operativo Windows XP Profesional
- Procesador Intel Celeron 550 de 2 GHz
- RAM de 1 GB

La aplicación fue desarrollada en el IDE wxDev C++ con la librería para procesamiento de imágenes OpenCV de Intel y la librería winmm para la reproducción del archivo MIDI.

El tiempo promedio de procesamiento fue aproximadamente de 5 segundos por pentagrama. El mayor tiempo se gastó en la etapa de detección de las líneas (sección 5.3.1) y en la segmentación y reconocimiento de notas, silencios, alteraciones y puntillos (sección 5.4.10).

También es necesario hacer un análisis del sistema implementado en este trabajo teniendo en cuenta sistemas OMR ya desarrollados.

Lo más común es encontrar en la literatura publicaciones enfocadas a ciertos módulos de un sistema OMR, así por ejemplo, hay trabajos sobre detección y eliminación de pentagramas, binarización y reconocimiento de símbolos.

También existen trabajos de tesis de pregrado y de maestría encaminados a desarrollar un sistema OMR con ciertas características y por lo general especializados a cierto tipo de partituras y utilizando un conjunto de notación musical reducido.

Por último, están los programas comerciales, que además de que hay que pagar una buena suma de dinero para obtenerlos tienen limitaciones en cuanto al tipo de partituras que analizan.

Un proyecto importante que hay que tener en cuenta como caso especial es el realizado en la universidad de Waseda, en el Japón, el cual fue desarrollado a principios de 1985 y consiste en un robot capaz de interpretar partituras musicales para piano, utilizando procesamiento de imágenes con módulos tanto hardware como software.

La información sobre el desarrollo de estos trabajos es limitada, especialmente en los programas OMR comerciales, sin embargo siempre hay información cuantitativa respecto al porcentaje de efectividad del sistema y al tiempo de ejecución del procesamiento de la partitura.

El porcentaje de efectividad se mide con relación a la capacidad del sistema para reconocer correctamente los símbolos musicales en partituras comunes, es decir, las que contienen cierto grado de deterioro y es del 95% para los programas mejor implementados. Por otro lado, el tiempo varía notablemente, habiendo programas que tardan alrededor del minuto para reconocer un solo pentagrama.

El enfoque del sistema implementado en este trabajo difiere sustancialmente de los trabajos existentes y su importancia radica en que fue desarrollado con técnicas de procesamiento de imágenes sencillas que reducen enormemente el tiempo de análisis de una partitura, siendo este de alrededor de 5 segundos por



pentagrama. Por otro lado, se puede ver que los resultados obtenidos en este trabajo igualan considerablemente a los obtenidos con sistemas más complejos como los utilizados por programas comerciales. Lo anterior supone que no siempre lo más complejo conlleva a mejores resultados.

## **CAPÍTULO 7. CONCLUSIONES Y TRABAJOS FUTUROS**

### **7.1. Conclusiones**

Se desarrolló un sistema de interpretación de partituras haciendo uso del procesamiento de imágenes, el cual, a partir de una partitura previamente digitalizada, obtiene una reproducción de ésta con sonidos MIDI. El sistema fue probado en 78 partituras de diferentes calidades con un conjunto amplio de símbolos musicales.

Los resultados obtenidos con el sistema implementado confirman que pueden ser equiparables a los obtenidos por sistemas más elaborados como los programas comerciales, los cuales utilizan complejas técnicas de análisis. Sin embargo, aún queda mucho por trabajar para poder llevar el sistema a este nivel.

El sistema desarrollado puede ser utilizado como un complemento en el aprendizaje de la música especialmente en el área de la lectura de partituras melódicas, ya sea como una herramienta para practicar el solfeo, afinación o dictado musical, especialmente para partituras que no contienen figuras musicales muy complejas, armonía o símbolos de expresión y movimiento.

El módulo de preprocesamiento funciona correctamente incluso para las partituras que están deterioradas. Es capaz de mejorar detalles en los símbolos y borrar zonas ruidosas.

La detección de pentagramas se realizó satisfactoriamente incluso cuando las líneas de los pentagramas se encontraban inclinadas, flexionadas o rotas. El algoritmo además detectó los bordes de los pentagramas eficazmente.

La segmentación y el reconocimiento de los símbolos dependen fuertemente de la eficacia de las etapas anteriores, especialmente del preprocesamiento y de la selección del área de análisis.

El carácter modular del sistema permite que se puedan adherir otros objetivos, como el de extender el análisis a partituras que contienen sistemas de dos pentagramas, armaduras con 7 alteraciones o incluso que contienen armonía.

Se diseñó una interfaz gráfica que permite al usuario utilizar el sistema de una manera fácil y amigable. La aplicación permite al usuario cargar la partitura original, mostrar los resultados del procesamiento, seleccionar la voz y la velocidad de interpretación y ejecutar la partitura con sonidos MIDI.

Los objetivos propuestos se cumplieron satisfactoriamente reconociendo los símbolos propuestos con un porcentaje de error pequeño.

## **7.2. Trabajos Futuros**

Mejorar la etapa de preprocesamiento para que el sistema pueda ser aplicado a imágenes con distintas resoluciones y tamaños así como a partituras en mal estado. Para este fin se pueden consultar artículos sobre técnicas de binarización ya que es esta etapa la más importante para obtener mejores resultados.

Utilizar un análisis de las componentes conectadas para mejorar la segmentación, aunque es una técnica que requiere mayor tiempo de ejecución garantiza que los resultados obtenidos mejoren, especialmente cuando existen varios objetos en una zona pequeña, los cuales con otras técnicas puedan ser identificados como uno solo.

Ampliar el conjunto de símbolos musicales que van a ser reconocidos, ya sea para la utilización del sistema en partituras polifónicas como las de piano o para reconocer símbolos propios de la armonía en el cual los símbolos comunes se superponen para formar otras notaciones como son los acordes. Además, se puede implementar un sistema OCR para el reconocimiento de los signos de expresión y movimiento.

Utilizar la cantidad de reglas musicales mayor posible para mejorar la segmentación y el reconocimiento y por consiguiente obtener mejores resultados. Además, tratar de detectar errores de reconocimiento y corregirlos efectivamente.

Permitir que la aplicación sea capaz de generar un archivo con formato MIDI para que los resultados obtenidos puedan ser usados en otros programas o reproducidos en otros medios, como PCs, sintetizadores o reproductores MIDI y también adherir en la aplicación la opción de edición para poder corregir los errores de reconocimiento o para agregar y cambiar las notas de la partitura.

Un buen proyecto sería analizar los tipos de errores presentes en las partituras, así como también las diferentes variaciones que se suelen encontrar en la notación musical común. Teniendo presente toda esta información en la elaboración del sistema, hay mayor probabilidad de que el reconocimiento mejore.

## REFERENCIAS

[1] <http://www.finalemusic.com>

[2] <http://www.sibelius.com>

[3] [www.musitek.com/smartscore.html](http://www.musitek.com/smartscore.html)

[4] <http://www.myriad-online.com/en/products/pdfmusicpro.htm>

[5] <http://es.wikipedia.org/wiki/Música>

[6] <http://es.wikipedia.org/wiki/Cubase>

[7] [http://es.wikipedia.org/wiki/Fruity\\_Loops](http://es.wikipedia.org/wiki/Fruity_Loops)

[8] F. Abad, ¿Do re qué? Guía práctica de iniciación al lenguaje musical. Editorial Berenice, Septiembre, 2008.

[9] A. De la escalera, Visión por computador. Fundamentos y métodos. Pearson Educación. S.A., Madrid, 2001. ISBN 84-205-3098-0.

[10] G. Pajares, J. De la Cruz, Visión por computador: Imágenes digitales y aplicaciones. Alfaomega Grupo Editor, S.A. de C.V., México, 2008. ISBN 978-97-15-1356-9.

[11] R. González, R. Woods, Tratamiento Digital de Imágenes. Wilmington, USA: Addison-Wesley, 1996. ISBN 0-201-62576-8.

[12] B. Couasnon, J. Camillerapp. A way to separate knowledge from program in structured document analysis: Application to OMR. 1995.

[13] J. McPherson, D. Bainbridge. Coordinating Knowledge within an Optical Music Recognition System. 2001.

[14] I. Fujinaga. Optical Music Recognition using projections. 1988.

[15] J. Burgoyne, L. Pugin, G. Eustace, I. Fujinaga. A Comparative survey of image binarization algorithms for optical recognition on degraded musical sources. 2007.

[16] C. Dalitz, M. Droettboom, B. Pranzas, I. Fujinaga. A comparative study of staff removal algorithms. 2008.

- [17] D. Blostein, H. Baird. A Critical Survey of Music Image Analysis. 1992.
- [18] I. Fujinaga. Adaptive Optical Music Recognition. 1996.
- [19] A. M. Rebelo, G. Capella, J. Cardoso. Optical Recognition of Music Symbols. A Comparative Study. 2009.
- [20] B. Couasnon, B. Rétif. Using a grammar for reliable full score recognition system. Septiembre, 1995.
- [21] M. Stückelberg, C. Pellegrini, M. Hilario. An architecture for musical score recognition using high-level domain knowledge. 1997.
- [22] F. Rossant. A global method for music symbol recognition in typeset music sheets. 2002.
- [23] F. Rossant, I. Bloch. Robust and adaptive OMR system including fuzzy modeling, fusion of musical rules, and possible error detection. 2007.
- [24] A. M. Rebelo. New Methodologies towards an Automatic Optical Recognition of Handwritten Musical Scores. Octubre 2008.
- [25] W.J. Kim, M.J. Chung, Z. Bien. Recognition System for Printed Music Score. 1987
- [26] P. Martin y C. Bellisant. Neuronal networks at different levels of musical score image analysis system. Proceedings of 7th scandinavian conference on image analysis. 1102-9. 1991.
- [27] M. Roth. An approach to recognition of printed music. *Technical Report 210*. Department of Computer Science, Swiss Federal Institute of Technology. 1994.
- [28] D. S. Prerau. Computer pattern recognition of printed music. In Proceedings, Fall Joint Computer Conference, volume 39, pages 153-162, Montvale, NJ, November 1971. A.F.I.P.S. Press.
- [29] D. Bainbridge, *Preliminary experiments in musical score recognition*. B.Sc. Thesis. University of Edinburgh. 1991.
- [30] H. Aoyama y A. Tojo. Automatic recognition of music score (in japanese). Electronic image conference journal. 11(5): 427-35. 1982.
- [31] E. Sicard. An efficient method for the recognition of printed music. *Proceedings of 11th International Conference on Pattern Recognition (IAPR)*. 573-6. 1992.

- [32] J. V. Mahoney. Automatic analysis of musical score images. B. s. thesis, Dept of Computer Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02129 usa, May 1982. Advisor: B. K. P. Horn; 53 pages.
- [33] H. Miyao, T. Ejima, M. Miyahara and K. Kotani. Symbol recognition for printed piano scores based on the musical knowledge (in Japanese). *Transactions of the Institute of Electronics, Information and Communication Engineers D-III* J75D11(11): 1848-55. 1992.
- [34] A. Rebelo, A. Capela, J. Pinto, C. Guedes, E. Carrapatoso, J. Cardoso. A shortest path approach for staff line detection. 2007.
- [35] H. Miyao, M. Okamoto. Stave Extraction for Printed Music Score Using DP Matching. 2003.
- [36] A. Andronico and A. Ciampa. On automatic pattern recognition and acquisition of printed music. In *Proceedings, International Computer Music Conference* , pages 245-278, Venice, Italy, 1982. Computer Music Association Publications.
- [37] J. W. Roach and J. E. Tatem. Using domain knowledge in low-level visual processing to interpret handwritten music: an experiment. *Pattern Recognition* , 21(1):33-44, 1988.
- [38] H. Kato and S. Inokuchi. A recognition system for printed piano music using musical knowledge and constraints. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis*. Springer-verlag, Heidelberg, 1991.
- [39] T. Kobayakawa, Auto music score recognition system. *Proceedings SPIE: Character Recognition Technologies*. D. P. D'Amato, ed. 1906. 1993.
- [40] T. Matsushima. Automated high speed recognition of printed music (Wabot-2 vision system). In *Proceedings, 1985 International Conference on Advanced Robotics*, pages 477-482. Japan Industrial Robot Association (JIRA), 3-5-8, Shiba Koen Minato-ku, Tokyo, 1985.
- [41] Mu Research Group. Automated recognition system for musical score. *Bulletin* 112, Science and Engineering Laboratory, Waseda University, 61 Nishiwaseda 1-chome, Shinjuku-ku, Tokyo 160, 1985. 28 pages.
- [42] C. Roads. The Tsukuba musical robot. *Computer Music Journal*, 10(2):39-43, Summer 1986.
- [43] D. Bainbridge, Dealing with superimposed objects in optical music recognition. 1997.

- [44] D. Bainbridge, Optical music recognition: A generalized approach. Sept. 1997.
- [45] B. R. Modayur, V. Ramesh, R. M. Haralick, and L. G. Shapiro. MUSER-a prototype musical recognition system using mathematical morphology. *Intelligent Systems Laboratory, EE Dept, FT-JO*. University of Washington. 1992.
- [46] B. Modayur. Music Score Recognition – A Selective Attention Approach using Mathematical Morphology. Marzo 2, 1996.
- [47] D. H. Pruslin. 1966, Automatic recognition of sheet music. Sc.D. diss., MIT.
- [48] N. P. Carter. Automatic Recognition of Printed Music in the Context of Electronic Publishing. PhD thesis, Univ. of Surrey, Depts. of Physics and Music, February 1989. 174 pages.
- [49] O. Yadid, E. Brutman, L. Dvir, M. Gemer, and U. Shimony. RAMIT: Neural network for recognition of musical notes. *Proceedings of the International Computer Music Conference*. 128-31. 1992.
- [50] D. Carretero. Sistema de reconocimiento de partituras musicales. 2009.
- [51] B. Coüasnon, P. Brisset, I. Stéphan. Using logic programming languages for optical music recognition. Abril, 1995.
- [52] B. Coüasnon, J. Camillerap. A way to separate knowledge from program in structured document analysis: application to optical music recognition. Agosto, 1995.
- [53] B. Coüasnon, J. Camillerap. Using grammars to segment and recognize music scores. In international association for pattern recognition workshop on document analysis system. Pages 15-27. Kaiser slautern. Germany. October, 1994.
- [54] D. Bainbridge, T. Bell, An extensible optical recognition system. 1997.
- [55] D. Bainbridge, T. Bell. The challenge of optical music recognition. 2001.
- [56] J. McPherson. Introducing feedback into an optical music recognition system. 2003.
- [57] D. Byrd, M. Schindele. Prospects for improving OMR with multiple recognizers. 2006.
- [58] McGeey W. F. and P. Merkley. 1991. The optical scanning of medieval music. *Computers and the Humanities*. 25(1): 47-53.

- [59] L. Tardón, S. Sammartino, I. Barbancho, V. Gómez, A. Oliver. Optical music recognition for scores written in white mensural notation. 2010.
- [60] [www.musitek.com/smartscore.html](http://www.musitek.com/smartscore.html)
- [61] [http://www.freedownloadmanager.org/es/downloads/SmartScore\\_Pro\\_50512\\_p/](http://www.freedownloadmanager.org/es/downloads/SmartScore_Pro_50512_p/)
- [62] <http://www.ircfast.com/lv/group/view/kl79817/PDFtoMusic.htm>
- [63] [http://www.aulaactual.com/revista/bancodepruebas/?id\\_prod=461](http://www.aulaactual.com/revista/bancodepruebas/?id_prod=461)
- [64] M. Kassler. An essay toward specification of a music-reading machine. In B. S. Brook, editor, *Musicology and the Computer*, pages 151-175. City University of NY Press, New York, NY, 1970.
- [65] D. S. Prerau. Do-re-mi: A program that recognizes music notation. *Computers and the Humanities*, 9:25-29, 1975.
- [66] <http://msdn.microsoft.com/en-us/library/dd743586%28v=VS.85%29.aspx>



## ANEXO A Manual de Usuario

En esta sección se explica el manejo de la interfaz gráfica de usuario para que el programa pueda ser utilizado correctamente.

Una vez ejecutado el programa (Figura A.1) aparecerá una ventana como la que se muestra en la figura A.2.



Figura A.1 Ícono del programa

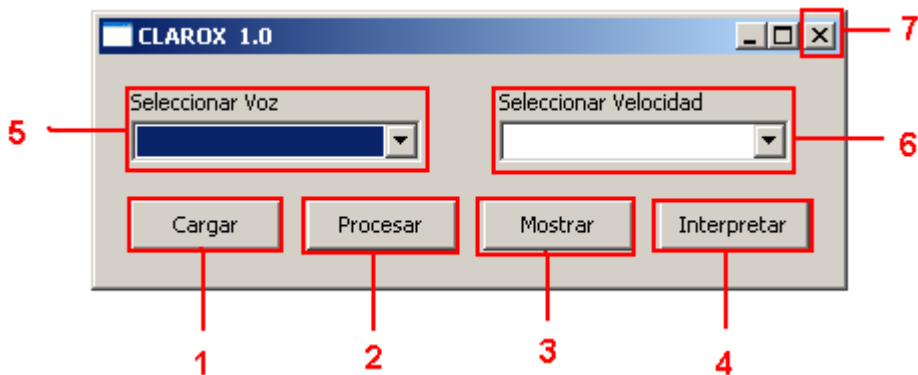


Figura A.2 Interfaz gráfica de usuario

### 1. Cargar

Este botón permite cargar la partitura que va a ser interpretada. Al dar click sobre este botón (número 1 en la figura A.2) aparecerá una ventana de selección como la que se muestra en la figura A.3.

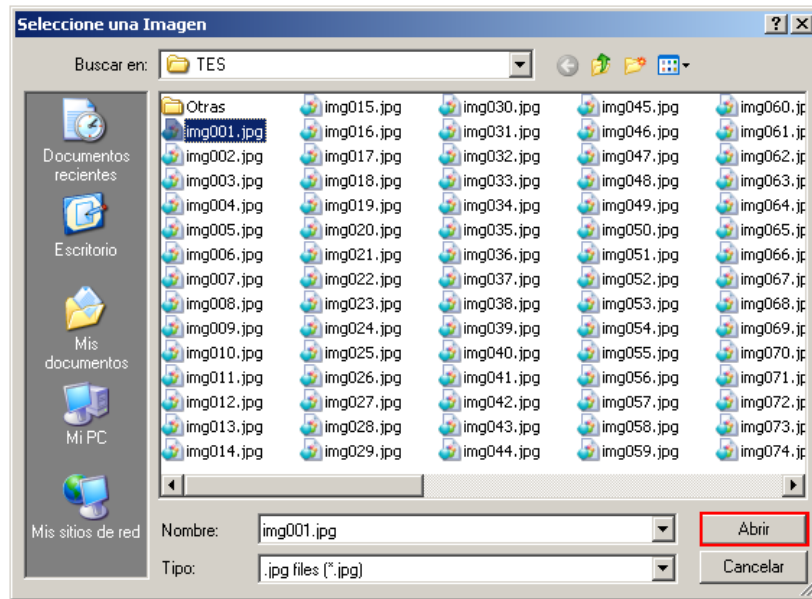


Figura A.3 Ventana de selección

A continuación seleccione la partitura que desee y de click en abrir.

El programa mostrará la partitura seleccionada como se muestra en la figura A.4.

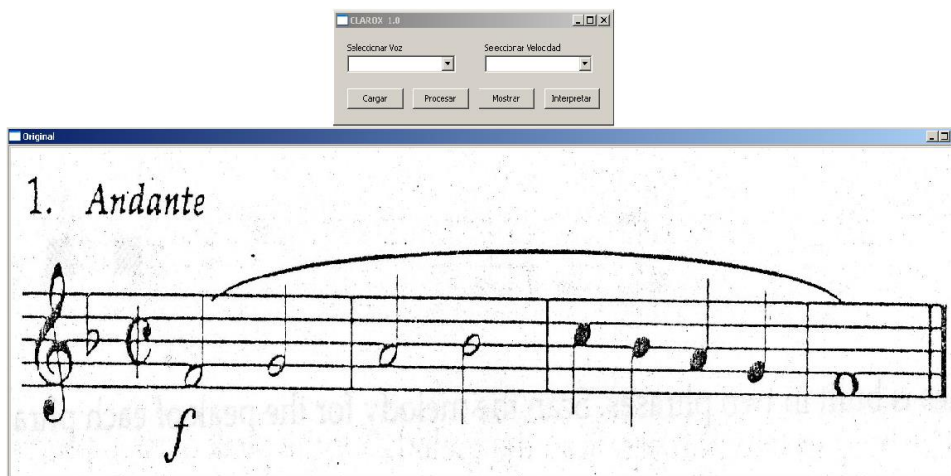


Figura A.4 Partitura cargada

## 2. Procesar

Una vez cargada la partitura el siguiente paso es procesarla. Para esto de click en el botón procesar (número 2 de la figura A.2).

El programa tardará un tiempo en procesar la imagen dependiendo del tamaño y el número de pentagramas de la partitura. Una vez la imagen ha sido procesada aparecerá un aviso como el que se muestra en la figura A.5, de click en aceptar y continúe con el siguiente paso.

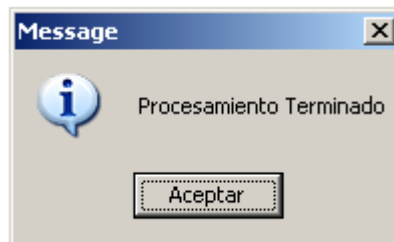


Figura A.5 Mensaje de proceso terminado

## 3. Mostrar

Esta opción permite ver la partitura procesada así como los símbolos reconocidos en la partitura. Al dar click en el botón mostrar (número 3 en la figura A.2) aparecen dos ventanas, una (Imagen Rotada) con la imagen de la partitura procesada y otra (Resultados) con los resultados del reconocimiento (Figura A.6).

## 4. Interpretar

El siguiente paso es la interpretación de la partitura. Para esto de click en el botón Interpretar (número 4 en la figura A.2). El programa ejecutará la partitura con sonidos MIDI según los símbolos reconocidos. Por defecto la ejecución se realizará con voz de piano y a una velocidad de 180 ppm (pulsos por minuto).

Una vez el programa haya ejecutado la partitura aparecerá un mensaje de terminado como el que se muestra en la figura A.7. De click y continúe.

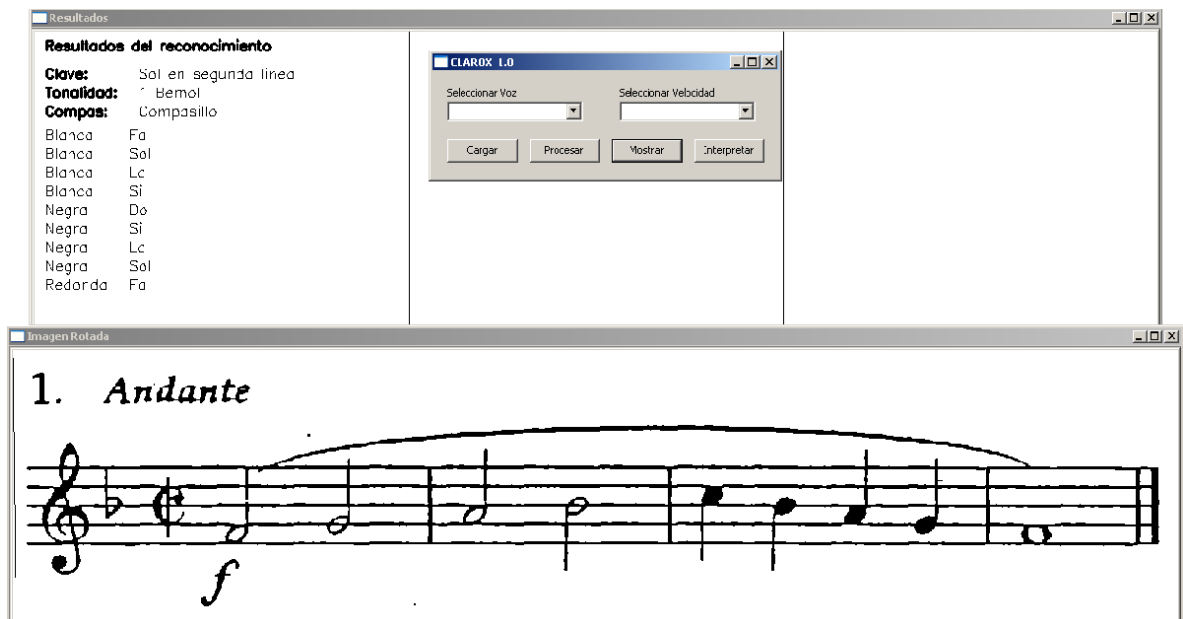


Figura A.6 Resultados del reconocimiento



Figura A.7 Mensaje de interpretación terminada

## 5. Seleccionar Voz

El programa permite seleccionar entre diecisiete voces distintas para la ejecución de la partitura. Para seleccionar una voz de click en el selector "Seleccionar Voz" (número 5 en la figura A.2). Aparecerá una lista como la que se muestra en la figura A.8.

Las voces disponibles son: piano, órgano, acordeón, guitarra acústica, guitarra eléctrica, bajo, violín, violonchelo, contrabajo, trompeta, trombón, tuba, trompa, saxo alto, saxo tenor, clarinete y flauta.

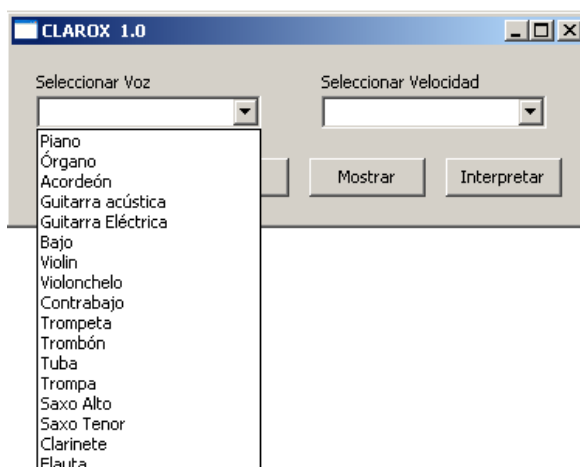


Figura A.8 Selector de voz

## 6. Seleccionar Velocidad

El programa permite seleccionar entre seis velocidades distintas para la ejecución de la partitura. Para seleccionar una velocidad de click en “Seleccionar Velocidad” (número 6 en la figura A.2). Aparecerá una lista como la que se muestra en la figura A.9.

Las velocidades disponibles son: lento (50 ppm), adagio (70 ppm), andante (90 ppm), moderato (110 ppm), allegro (140 ppm) y presto (180 ppm).

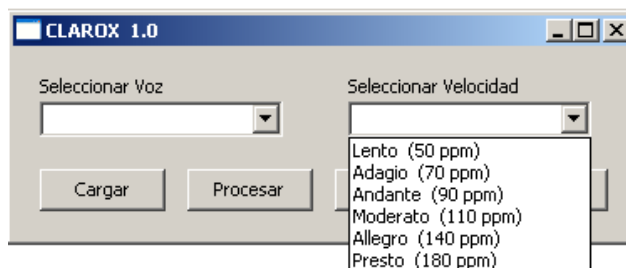


Figura A.9 Selector de velocidad

## 7. Cerrar

Para finalizar el programa de click en cerrar (número 7 en la figura A.2).