

# Reconocimiento automático de la tonalidad en partituras digitales

David Rizo José M. Ñesta Pedro J. Ponce de León Antonio Pertusa

Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Alicante, E-03080 Alicante, Spain  
drizo, inesta, pierre, pertusa@dlsi.ua.es

## Resumen

La música occidental está basada, en su mayoría, en el concepto de tonalidad<sup>1</sup>. La mayor parte de las partituras digitales que se pueden encontrar en internet de forma gratuita está disponible en formato MIDI<sup>2</sup>, y gran parte de ellas no contiene información acerca de la tonalidad inicial de la obra ni de los cambios de tonalidad que se producen a lo largo de ésta. En este trabajo se propone un algoritmo para inferir la tonalidad de un fragmento musical. Este algoritmo está basado en un esquema de representación simbólica de música polifónica mediante árboles, que mejora los resultados y el rendimiento de otros modelos analizados.

## 1. Introducción

La tonalidad puede definirse como el conjunto de sonidos que constituyen un sistema cuyo eje principal es la nota *tónica*, que se toma como referencia para todas las demás notas que aparecen en la obra o fragmento a considerar. En la música occidental este conjunto de sonidos se denomina *escala diatónica* y está formado por una serie ascendente de siete notas musicales, la primera de las cuales es la tónica. Existen doce notas musicales diferentes<sup>3</sup> en la música occidental (figura 1), cada una de las cuales puede actuar como tónica de una tonalidad. Es habitual construir dos es-

calas diatónicas diferentes sobre cada tónica, dando lugar a dos tonalidades diferentes que derivan de la misma tónica, a las que se denomina, respectivamente, modo mayor y modo menor (figuras 2 y 3). Consideraremos, por tanto, que un fragmento de música occidental está escrito en una de 24 tonalidades posibles. El problema de detectar esa tonalidad se puede enfocar entonces como el problema de clasificar un fragmento musical en una de 24 clases (tonalidades) posibles. El problema no es trivial, ya que el conjunto de notas que forman un fragmento musical es, en general, un conjunto de sonidos que no se corresponde con una única escala diatónica. Este conjunto puede contener todas o algunas de las notas de diferentes escalas diatónicas, en mayor o menor número. Además, para inferir la tonalidad no sólo es importante la distribución de notas en el fragmento, sino también cuáles de ellas suenan simultáneamente.

La mayoría de trabajos que detectan la tonalidad de un fragmento musical usan secuencias lineales de notas ([6],[8]), aunque existen otras alternativas como el método basado en una estructura de datos llamada *spiral array*, presentado por Chew [1]. En [4] los autores usaron árboles para representar música monofónica<sup>4</sup> con el objeto de medir la similitud entre fragmentos musicales, obteniendo mejores resultados mediante representaciones jerárquicas en forma de árboles que usando representaciones lineales (secuencias de cadenas). En este artículo extendemos el modelo de árboles propuesto en el trabajo anterior para representar

<sup>1</sup>Véase <http://www.teoria.com> como referencia sobre conceptos musicales

<sup>2</sup>[www.midi.org](http://www.midi.org)

<sup>3</sup>Son, en realidad, doce *clases* que agrupan sonidos cuya frecuencia es múltiplo de una frecuencia base determinada. Para simplificar denominamos a estas clases notas musicales

<sup>4</sup>En una melodía monofónica sólo puede haber una nota sonando simultáneamente. En una polifónica puede haber más de una.

melodías polifónicas y lo usaremos para inferir la tonalidad de fragmentos musicales.



Figura 1: Las doce notas musicales



Figura 2: Escala diatónica mayor



Figura 3: Escala diatónica menor extendida

## 2. Representación de secuencias musicales mediante árboles

Las características principales de una nota musical son su duración y su altura. En representaciones lineales, las duraciones de las notas se codifican usando símbolos explícitos, pero los árboles son capaces de representar implícitamente estas características. Para ello hacen uso de la naturaleza logarítmica de los tiempos musicales, en el sentido de que las duraciones de las notas son múltiplos de unidades de tiempo básicas, principalmente de estructura binaria (a veces, ternaria). De este modo, los árboles son menos sensibles que las cadenas al tipo de codificación usada para representar melodías.

En esta sección se muestra el método de construcción de árboles introducido en [4] para representar un segmento monofónico de música y se definen los términos necesarios para construir el modelo.

### 2.1. Construcción de un árbol para cada compás

La representación en árbol se basa en el hecho de que la duración de las notas musicales está especificada en una escala logarítmica. Una nota *redonda* dura cuatro tiempos musicales, que es el doble de lo que dura una *blanca*, que a su vez dura el doble que una *negra*, etc. (ver Fig. 4).

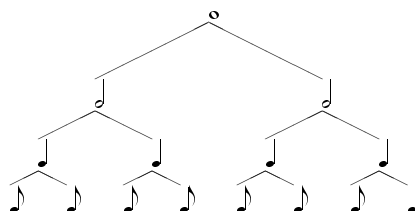


Figura 4: Jerarquía de duración para algunas notas musicales. De arriba a abajo: redonda (4 tiempos musicales), blanca (2 tiempos musicales), negra (1 tiempo musical) y corchea (1/2 tiempo musical).

Cada compás<sup>5</sup> en una melodía se representa mediante un árbol  $\tau$ , y cada nota o silencio (ausencia de notas) es un nodo hoja. El recorrido de las hojas de izquierda a derecha mantiene el orden temporal de las notas en el compás. El nivel de cada hoja en el árbol determina la duración de la nota que representa, como muestra la figura 4; el nodo raíz (nivel 1) representa la duración de todo el compás (una nota *redonda* en una métrica 4/4), y cada uno de los nodos del nivel 2 representa la duración de una *blanca*. Generalizando, los nodos del nivel  $i$  representan la duración de  $1/2^{i-1}$  de compás.

Durante el proceso de construcción de un árbol, los nodos internos se crean cuando es necesario alcanzar el nivel de hoja adecuado. Inicialmente, sólo contendrán un valor de etiqueta los nodos hoja, pero estas etiquetas se propagan de abajo a arriba para etiquetar todos los nodos del árbol. Las reglas de propagación se describen en la sección 2.4.

<sup>5</sup>Un compás es la unidad básica de estructura musical

Las etiquetas de los nodos son conjuntos de símbolos que indican la altura de las notas sin información de octava. El símbolo que denota una altura es el número de nota MIDI módulo 12. El símbolo que denota un silencio es  $s$ .

La figura 5 muestra un ejemplo de este esquema. En el árbol, el hijo izquierdo del nodo raíz se divide en dos subárboles para alcanzar el nivel 3, que es el que corresponde a la primera nota (una negra, de duración  $1/2^2$  del compás), y cuya altura es SI (altura 11). Para representar las duraciones del silencio y de la nota SOL(7), que son de  $1/8$  del compás, se necesita un nuevo subárbol para el hijo derecho del nivel 3. Por lo tanto se generan dos nuevos nodos hoja para representar el silencio ( $s$ ) y la nota SOL (7). La nota blanca DO (0) comienza en el tercer tiempo musical del compás, y dura dos tiempos, por lo que se representa como un nodo hoja etiquetado en el nivel 2.

Como puede verse en la figura 5, el orden temporal de las notas en la partitura se preserva cuando se recorre el árbol de izquierda a derecha. Los tiempos de inicio y las duraciones de las notas están implícitamente representadas en el árbol. Esta representación es invariante ante cambios en el *tempo*<sup>6</sup> o ante diferentes representaciones de la métrica de la misma melodía (ej. 2/2, 4/4, ó 8/8).

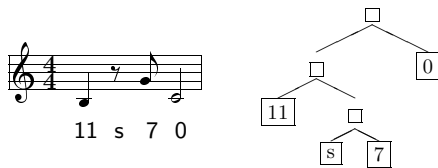


Figura 5: Ejemplo simple de la creación de un árbol

## 2.2. Representación de la melodía completa

El método descrito anteriormente es capaz de representar un compás mediante un árbol

<sup>6</sup>El *tempo* es una indicación de la rapidez con que se debe ejecutar una obra. Se suele cuantificar en tiempos o pulsos por minuto.

$\tau$ . Un compás es la unidad básica de estructura musical, pero una melodía se compone de una secuencia de varios compases. En [4] se propone construir un árbol con un nodo raíz para toda la melodía, siendo cada compás un subárbol de esa raíz. Esta representación puede considerarse como un bosque de subárboles unidos a un nodo raíz común que representa la melodía completa. La figura 6 muestra un ejemplo de una melodía simple compuesta por tres compases y cómo se representa en un árbol compuesto por tres subárboles, uno por compás, enraizados al mismo nodo padre. A este nodo común se le asigna el nivel 0.

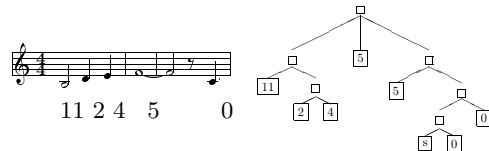


Figura 6: Un ejemplo de la representación de un árbol de una melodía completa. La raíz conecta todos los subárboles de compases.

## 2.3. Polifonía

El método para representar música polifónica es sencillo. Todas las notas se colocan en un mismo árbol. Si un nodo ya existía cuando una nueva nota se inserta en el árbol, la altura de esta nota se añade a la etiqueta del nodo actual. Si la etiqueta del nodo era un silencio, se sustituye por la altura de la nota. La figura 7 muestra un ejemplo de una melodía con un acorde. Antes de la propagación (ver sección 2.4) sólo se etiquetan las hojas.

## 2.4. Propagación de las etiquetas de abajo a arriba

Una vez se ha construido el árbol, se realiza la propagación. Las reglas de propagación son diferentes a las propuestas en [4], donde se eliminaban notas que no se consideraban importantes ya que el objetivo era la búsqueda de similitudes entre melodías. El proceso de propagación se realiza recursivamente en un

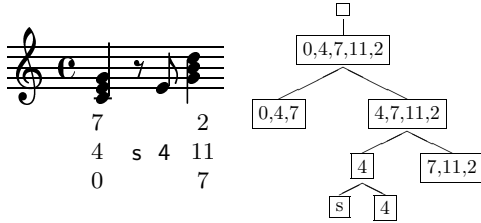


Figura 7: Ejemplo de una melodía polifónica. Los nodos internos se etiquetan durante la propagación.

recorrido postorden del árbol. Dado un subárbol  $\tau$  con hijos  $c_i$ , las reglas de propagación y poda son las siguientes:

1. Si  $\tau$  tiene una etiqueta vacía y cada  $c_i$  de  $\tau$  contiene la misma etiqueta, ésta se copia en  $\tau$ . Si todos los hijos de  $\tau$  son hojas, se eliminan.
2. Si  $\tau$  tiene una etiqueta vacía y todas las etiquetas hijo son  $s$  o vacías excepto una, ésta se copia en  $\tau$ .
3. Si no se aplica ninguna de las reglas anteriores, las etiquetas de todos los  $c_i$  se mezclan en  $\tau$ , añadiéndose a las ya existentes.

En la figura 7 puede verse cómo asciende la corchea ([4]) que compartía un padre con el silencio. Uniendo esta nota con el acorde siguiente ([7,11,2]) se obtiene una etiqueta padre ([4,7,11,2]).

### 3. Algoritmo de detección de la tonalidad

En la representación de música polifónica en forma de árboles, presentada en la sección 2, cada nodo hoja contiene una o varias alturas que pueden formar un acorde<sup>7</sup>. Ya sea una nota simple o un acorde, estas alturas encajan mejor en algunas tonalidades que en otras. En esta sección se define un algoritmo que puntúa cada nodo del árbol para cada una de las 24 tonalidades (12 mayores y 12 menores). Una vez

<sup>7</sup>conjunto de notas que suenan simultáneamente.

calculadas las puntuaciones, las tonalidades se ordenan por puntuación.

La búsqueda de la tonalidad principal de la melodía se basa en un recorrido postorden del árbol. En el proceso recursivo, una vez los hijos tienen las puntuaciones para cada una de las tonalidades calculadas, sus resultados se combinan para obtener un ranking para cada tonalidad en el nivel del padre (ver sección 3.2). Finalmente, el nodo raíz del árbol  $\tau$  contiene un lista de tonalidades ordenadas por puntuación. La tonalidad de la melodía o del fragmento es la que tiene la mejor puntuación.

El algoritmo 1 busca la tonalidad en un árbol polifónico.

---

#### Algoritmo 1 Búsqueda de tonalidad en el árbol $\tau$

---

```

if arity( $\tau$ ) = 0 then
  Calcular puntuación de tonalidades para el nodo raíz del árbol  $\tau$  (ver sección 3.2)
else
  for all  $child(\tau) \in children(\tau)$  do
    Algoritmo 1 para el árbol  $child(\tau)$ 
  end for
  Combinar los resultados para el nodo raíz de  $\tau$  y todos los hijos del nodo raíz de  $\tau$ 
end if

```

---

### 3.1. Escalas, grados y acordes

#### 3.1.1. Escalas

Las escalas que se utilizan como modelo de la tonalidad son:

**Escala mayor**

$$S_M = [1, 0, 2, 0, 3, 4, 0, 5, 0, 6, 0, 7]$$

**Escala menor**

$$S_m = [1, 0, 2, 3, 0, 4, 0, 5, 6, 6, 7, 7]$$

La posición en la lista, comenzando desde cero, es el intervalo desde la nota tónica de la tonalidad, medido en semitonos. Los valores distintos de cero son los grados (ver sección 3.1.2) en la escala, y los valores iguales a cero se corresponden con las notas que no pertenecen a la escala. La escala menor se ha

representado como la unión de la escala menor natural y la escala menor melódica, denominándose escala menor extendida.

### 3.1.2. Grados

Una tonalidad  $k$  se compone de una nota tónica y de su modo (mayor o menor), representados por su correspondiente escala  $S$ . Dada una altura  $p$ , una tonalidad  $k$  y su escala  $S$  asociada, el *grado* de la altura  $p$  se define como:

$$\text{grado}(p, k, S) = S[((p + 12) - k) \bmod 12] \quad (1)$$

En la expresión anterior se suma 12 a  $p$  para evitar un módulo negativo.

Los grados 0, 5 y 7<sup>8</sup> son los llamados *grados tonales*. El grado 4 es característico del modo mayor y el grado 3 lo es del modo menor<sup>9</sup>, y se denominan *grados modales*.

### 3.1.3. Acordes

Sólo consideraremos las triadas (acordes de tres notas) que se construyen sobre cada nota de la escala diatónica. El conjunto de alturas contenidas en la etiqueta de un nodo pueden constituir tanto una triada completa como una parcial. Dado el conjunto de alturas  $P = \{p_1, p_2, \dots, p_{|P|}\}$  en un nodo, se define  $\varphi$  como el número de elementos en  $P$  que pertenecen a un acorde  $c$  de la tonalidad  $k$ .

En la figura 7, si  $c$  fuera la triada de la tónica de *DO mayor* (el primer acorde de la figura) y  $P$  fuera la etiqueta del nodo más a la izquierda del árbol, el número de notas comunes entre ambos sería  $\varphi = 3$ . En cambio, si  $c$  fuera el acorde *LA menor* (formado por las notas LA, DO y MI), el resultado sería  $\varphi = 2$ .

### 3.2. Ordenación de las tonalidades en un nodo

El algoritmo 3.2 calcula la puntuación para cada tonalidad dependiendo del conjunto de alturas que hay un nodo. Devuelve números

<sup>8</sup>Estos valores se corresponden con los grados musicales *I, IV* y *V*

<sup>9</sup>Ambos valores corresponden al grado musical *III*

	grado base del acorde $c$						
$\varphi$	0	2	3,4	5	7	9	11
3	16	15	15	15	16	15	15
2	9	8	8	8	9	8	8

Cuadro 1: Puntuaciones con  $|P| = 3$

	grado base del acorde $c$						
$\varphi$	0	2	3,4	5	7	9	11
2	10	9	9	9	10	9	9

Cuadro 2: Puntuaciones con  $|P| = 2$

negativos debido a que las ordenaciones que se realizan posteriormente son de menor a mayor.

---

**Algoritmo 2** Puntuación de la tonalidad  $k$ , para las alturas  $P$

---

```

if  $|P| = 3$  then
  return ((mejor puntuación para  $P$  y  $k$ 
  en el cuadro 1)*(-1))
else if  $|P| = 2$  then
  return ((mejor puntuación para  $P$  y  $k$ 
  en el cuadro 2)*(-1))
else
  if notas de la escala de  $P \in k > |P| - 2$ 
  then
    if grados tonales  $\in P > 0$  then
      return (-4)
    else if grados modales  $\in P > 0$  then
      return (-3)
    else
      return (-2)
    end if
  end if
end if

```

---

Las puntuaciones de las tablas 1 y 2 utilizadas en el algoritmo se han establecido empíricamente. El esquema da preferencia a las triadas respecto a otros conjuntos de alturas.

Tras calcular las puntuaciones, las tonalidades se ordenan de menor a mayor puntuación. El objetivo de esta ordenación es evitar la propagación de una tonalidad con una puntuación buena en un nodo que pueda condicionar el resultado final. Si dos tonalidades tienen la misma puntuación se les asigna el mismo número de orden. La función  $\text{rank}(\tau, k)$  devuelve la

posición de la tonalidad  $k$  en este ranking.

### 3.3. Combinación y propagación de tonalidades de subárboles

Una vez se ha calculado el ranking de tonalidades para los nodos hijos y el nodo padre, deben combinarse para reemplazar todas las puntuaciones en el nodo padre. Esta operación se lleva a cabo en dos pasos. Primero, se calculan las puntuaciones de las tonalidades y después se vuelve a calcular un nuevo ranking.

Dado un nodo padre  $\tau$  con hijos  $c_1, c_2, \dots, c_{arity(\tau)}$ , la nueva puntuación para cada tonalidad  $k$  se calcula como:

$$punt(\tau, k) = rank(\tau, k) + \sum_{i=1}^{arity(\tau)} rank(c_i, k) \quad (2)$$

Como se ha descrito anteriormente, con estas nuevas puntuaciones es necesario volver a calcular un nuevo ranking.

## 4. Experimentos y resultados

Para evaluar nuestro algoritmo, se han usado 251 ficheros MIDI de música clásica de Cimarosa, Albinoni, Bach, Beethoven, Chopin y Dvorak entre otros. Para evitar en lo posible cambios de tonalidad, se han utilizado únicamente los 8 primeros compases de cada fichero.

Hemos comparado nuestro método con dos sistemas existentes de detección de tonalidad. El primero es el programa *key* del sistema *Melisma*<sup>10</sup>. Este sistema implementa tres algoritmos diferentes, usándose uno u otro según seleccione el usuario: *CBMS* ([5]), un algoritmo bayesiano de detección de tonalidad [6] y el algoritmo *Krumhansl-Schmuckler* (K-S) ([2]). El otro sistema comparado es el programa *key* del sistema *Humdrum* (HUM)<sup>11</sup>, el cual implementa también el algoritmo de *Krumhansl-Schmuckler* con los parámetros indicados por los autores en [3].

En el caso del programa *key* de *Melisma* se obtiene una lista de tonalidades ordenadas en

Tonalidad detectada	Puntuación
Correcta	1
Quinta justa	0.5
Mayor/menor relativa	0.3
Mayor/menor paralela	0.2

Cuadro 3: Puntuaciones MIREX 2005

el tiempo. Se considera como la tonalidad de todo el fragmento aquella que más se repite.

Para comparar los resultados, hemos seguido el proceso de evaluación propuesto en el tema *Audio and Symbolic Key Finding* de la *2nd Annual Music Information Retrieval Evaluation eXchange (MIREX 2005)*<sup>12</sup>, como se detalla en la tabla 3.

La tasa de acierto del algoritmo se obtiene como los puntos obtenidos promediados para todos los ficheros del corpus.

El sistema *Melisma* está implementado en ANSI C, nuestro sistema usa la máquina virtual Java 1.4.2-38, que es incluso más lento que el código nativo generado desde el C. Todos los experimentos se han realizado sobre un ordenador Apple PowerBook, que usa un procesador PowerPC G4 (1.1) 1.33 Ghz con 512 Mb de memoria RAM.

La gráfica 8 muestra los resultados de las puntuaciones medias de nuestro algoritmo (*Trees*), cada uno de los tres métodos implementados en *Melisma* y el sistema *Humdrum*. Los mejores resultados son aquellos con valores más cercanos a 1. Asimismo, en la gráfica 9 se representan los tiempos de cálculo total en segundos de todos los algoritmos probados.

El algoritmo *Trees* obtiene las mejores tasas de acierto y requiere hasta ocho veces menos tiempo de cálculo que los otros.

## 5. Conclusiones y trabajos posteriores

En este trabajo se ha presentado una representación en árbol para música polifónica que se ha mostrado simple y adecuada para encontrar la tonalidad de una melodía. Los porcen-

<sup>10</sup><http://www.link.cs.cmu.edu/music-analysis/>

<sup>11</sup><http://dactyl.som.ohio-state.edu/Humdrum/>

<sup>12</sup>[http://www.music-ir.org/mirexwiki/index.php/MIREX\\_2005](http://www.music-ir.org/mirexwiki/index.php/MIREX_2005)

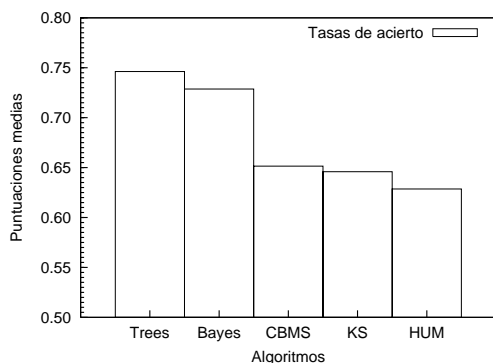


Figura 8: Puntuaciones medias

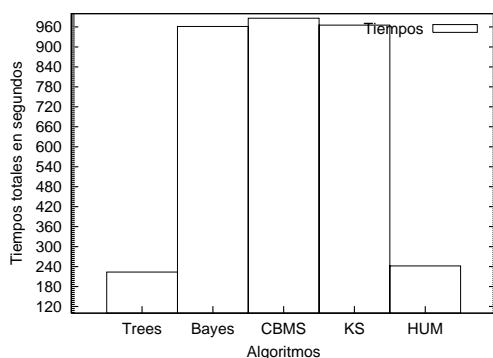


Figura 9: Tiempos totales

tajes de acierto son algo superiores a los obtenidos por los sistemas *Melisma* y *Humdrum* y los tiempos de cálculo mejoran significativamente.

Como se ha visto, con muy poca información armónica se puede obtener un buen detector de tonalidad. El sistema puede mejorar usando un modelo armónico más complejo con el que previsiblemente se obtendrán mejores resultados.

Actualmente también se está trabajando en la detección de cambios de tonalidad en una melodía obteniendo resultados prometedores que esperamos comparar con otros trabajos como [1] y [7]. También se está investigando

en la aplicación de algoritmos de aprendizaje supervisado para inferir las puntuaciones especificadas en los cuadros 1 y 2 a partir de fragmentos musicales para los cuales se conoce a priori su tonalidad.

## 6. Agradecimientos

Este trabajo ha sido subvencionado por el proyecto CICYT TIC2003-08496-C04, y parcialmente por los proyectos ERDF de la UE y GV043-541 de la Generalitat Valenciana.

## Referencias

- [1] E. Chew. A walk through tonal space: Charting the establishing of key of a melody. In *3rd Annual Symposium on Systems Research in the Arts.*, 2001.
- [2] C. Krumhansl. *Cognitive Foundations of Musical Pitch.* Oxford University Press, New York, NY, USA, 1990.
- [3] C. Krumhansl and E.J. Kessler. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review*, 89:334-368, 1982.
- [4] D. Rizo, F. Moreno-Seco, and J.M. Iñesta. Tree-structured representation of musical information. *Lecture Notes in Computer Science - Lecture Notes in Artificial Intelligence*, 2652:838-846, 2003.
- [5] D. Temperley. *The Cognition of Basic Musical Structure.* MIT Press, 2001.
- [6] D. Temperley. A bayesian approach to key-finding. *Lecture Notes in Computer Science*, 2445:195-206, 2002.
- [7] O. Yli-Harja, I. Shmulevich, and K. Lemström. Graph-based smoothing of class data with applications in musical key finding. In *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, 1999.
- [8] Y. Zhu and M. Kankanhalli. Key-based melody segmentation for popular song. *17th International Conference on Pattern Recognition (ICPR'04)*, 3:862-865, 2004.