Key Estimation from Polyphonic Audio

Emilia Gómez
Music Technology Group, IUA, Universitat Pompeu Fabra
Ocata, 1
08003 Barcelona
emilia.gomez@iua.upf.es

ABSTRACT

This document briefly describes an approach for audio key estimation based on the extraction of a vector of features, the Harmonic Pitch Class Profile, and the adaptation of a tonal model to these features. The algorithm is evaluated in the context of the 2005 ISMIR Contest.

1 Algorithm description

A diagram of the algorithm is shown in Figure 1. We first extract a vector of low-level features from the audio signal, which are related to its pitch class distribution. These features are averaged over a given segment and compared to a tonal model in order to find the key of the piece. We refer to Gómez (2004a,b); Gómez and Herrera (2004) for a detailed description of the algorithm.

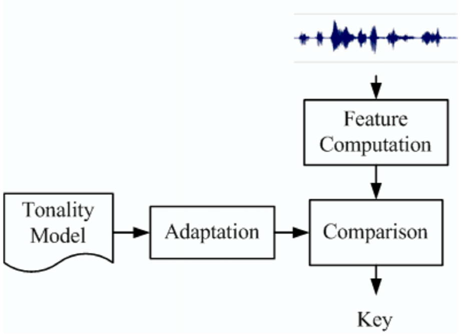

Figure 1: Block diagram for Key Estimation

1.1 Feature Extraction

The feature used in this algorithm is the Harmonic Pitch Class Profile (HPCP), measuring the relative intensity of each pitch class for an analysis frame. The extraction procedure is shown in Figure 2.

The signal is first pre-processed using DFT, frequency filtering ($100-5000$ Hz) and spectral peak location. Then, the tuning frequency is estimated by analyzing the frequency deviation of the located spectral peaks. The mapping between frequency and pitch class values is made using a logarithmic function with respect to the computed reference frequency. HPCP is then defined as:

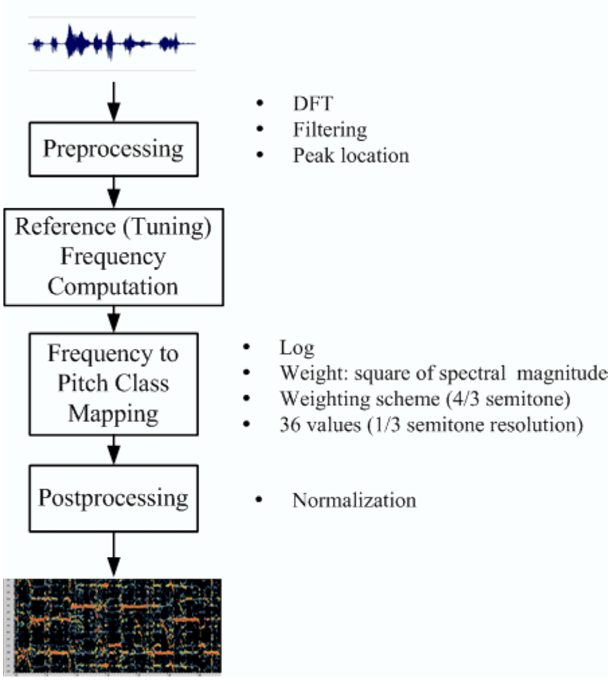
Preprocessing
• DFT
• Filtering
• Peak location
Reference (Tuning) Frequency Computation
Frequency to Pitch Class Mapping
• Log
• Weight: square of spectral magnitude
• Weighting scheme (4/3 semitone)
• 36 values (1/3 semitone resolution)
Postprocessing
• Normalization

Figure 2: Block diagram for HPCP computation

$$HPCP(n) = \sum_{i}^{nPeaks} w(n, f_i) \cdot a_i^2 \quad n = 1...size \quad i = 1...nPeaks \tag{1}$$

where a_i and f_i are the spectral magnitude and frequency of the i-th peak, $size = 36$ and $w(n, f_i)$ represents a window function having a $\frac{4}{3}$ semitones width and centered in f_i. In the post-processing step, the HPCP vector is normalized in order to make it independent on dynamics.

A global HPCP vector is computed by averaging the instantaneous values. Two different versions of the algorithm were provided, considering the whole signal (*global*) or only the first 15 seconds of it (*start*).

1.2 Tonality model

This approach is based on using the tone profiles proposed by Temperley (1999) T_M and Tm. We adapt these profiles to deal with polyphonic audio. First, we consider the simultaneous presence of several notes of a chord. Second, we consider the simultaneous presence of several harmonics of each note.

$$T_{Mp}(i) = \sum_{j=1}^{12} \alpha_M(i,j) \cdot T_M(j) \;\; i = 1...12 \tag{2}$$

$$T_{mp}(i) = \sum_{j=1}^{12} \alpha_m(i,j) \cdot T_m(j) \;\; i = 1...12 \tag{3}$$

The coefficients $\alpha_M(i,j)$ and $\alpha_m(i,j)$ are chosen following two rules: first, the same weight is given to the notes of the three main triads of a key (tonic, dominant and subdominat); second, the weight for the harmonics decrease following an exponential function.

These profiles are finally interpolated in order to obtain 36 values and correlation is the distance measure employed to compare templates to HPCP.

2 Evaluation results

Dataset: 1.252 audio files synthesized from MIDI Note, two databases: Winamp synthesized audio (w) and Timidity with Fusion soundfonts synthesized audio (t). The composite score is calculated by averaging the Winamp and Timidity scores. Results are presented in Table 1.

Rank		3	4
Method		start	global
Composite % Score		86.05%	85.90%
Total Score	w	1081.9	1076.1
	t	1072.9	1073.8
% Score	w	86.4%	86.0%
	t	85.7%	85.8%
Correct Keys	w	1048	1019
	t	1034	1015
Perfect 5th Errors	w	35	69
	t	44	73
Relative Major/Minor Errors	w	38	62
	t	43	59
Parallel Major/Minor Errors	w	25	20
	t	20	23
Other Errors	w	106	82
	t	111	82
Runtime (s)	w	1560	2041
	t	1531	1971
Machine		B0	B0

Table 1: Evaluation results of the method for the ISMIR 2005 Contest

3 System requirements

- **Format**: statically linked binary. Format: *./UPFAudioKeyContest-static input.wav output.txt*
- **Input**: *in.wav*. Audio file: wav format, signed 16 bit Little Endian, Rate 44100 Hz, Mono.
- **Output**: *out.txt*. Text file (estimated key). Default value: *in.wav-UPFKey.txt*
- **Computation time**:
  - For the training set (96 files, 30 seconds each).
  - Time of processing: real: 9m7.768s, user: 6m49.170s, sys: 0m56.140s.
- **Computer features**: LINUX (debian sarge), Intel(R) Pentium(R) 4 CPU 2.00GHz, cache size: 512 KB.

References

Emilia Gómez. Tonal description of polyphonic audio for music content processing. *Journal on Computing, Special Cluster on Computation in Music*, accepted, 2004a.

Emilia Gómez. Tonal description of polyphonic audio for music content processing. In *Audio Engineering Society conference on metadata*, March 2004b.

Emilia Gómez and Perfecto Herrera. Estimating the tonality of polyphonic audio files: cognitive versus machine learning modelling strategies. In *International Conference on Music Information Retrieval*, October 2004.

David Temperley. What's key for key? the krumhansl-schmuckler key finding algorithm reconsidered. *Music Perception*, 17(1):65–100, 1999.