An Audio Fingerprinting System for Live Version Identification using Image Processing Techniques

Zafar Rafii[1] Bob Coover[2] Jinyu Han[2]

[1]Northwestern University, Evanston, IL, USA. zafarrafii@u.northwestern.edu
[2]Gracenote, Inc., Emeryville, CA, USA. {bcoover,jhan}@gracenote.com

INTRODUCTION

We propose an audio fingerprinting system that can deal with live version identification by using image processing techniques. Compact fingerprints are derived using a log-frequency spectrogram and an adaptive thresholding method, and template matching is performed using the Hamming similarity and the Hough Transform. The system is specially intended for applications where a smartphone user is attending a live performance from a known artist and would like to quickly know about the song that is being played.

SYSTEM - Fingerprinting

In the first stage, compact fingerprints are derived from the audio signal, by first using a log-frequency spectrogram to capture the melodic similarity and handle key variations, and then an adaptive thresholding method to reduce the feature size and handle noise degradations and local variations.

❶Constant Q Transform

- time/frequency resolution of 0.13 second/1 quarter tone
- frequency range from C3 (130.81 Hz) to C8 (4186.01 Hz)
- log-frequency resolution to handle pitch deviations

❷Adaptive Thresholding

- if t-f bin > median of neighbors, replace by 1; 0, otherwise
- window size of 15 time frames by 35 frequency channels
- segmentation into regions of locally high and low energy to handle noise degradations and local variations

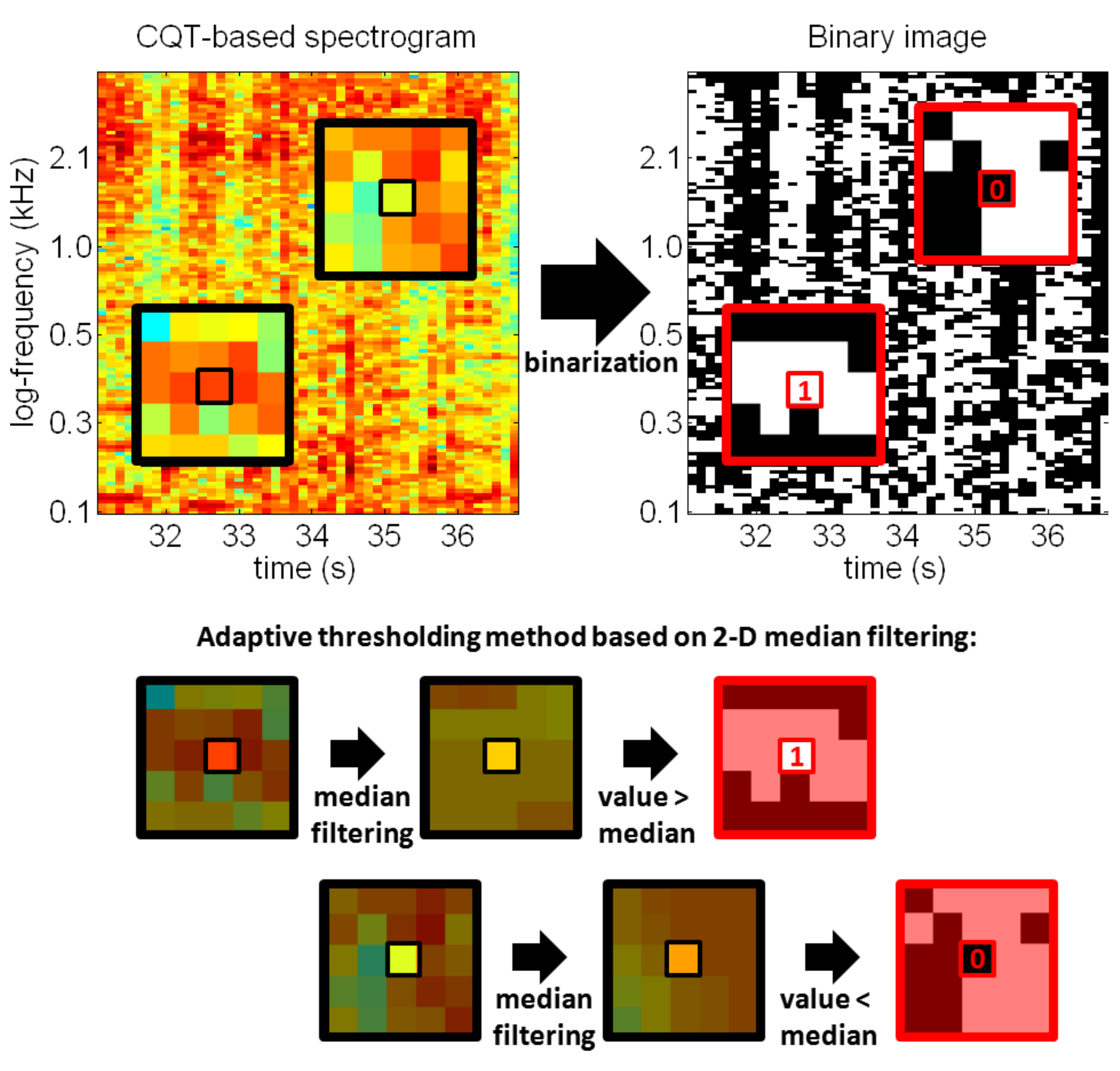

Figure 1: Overview of the fingerprinting stage. The audio signal is first transformed into a log-frequency spectrogram by using the CQT. The CQT-based spectrogram is then transformed into a binary image by using an adaptive thresholding method.

SYSTEM - Matching

In the second stage, template matching is performed between query and reference fingerprints, by first using the Hamming similarity to compare all pairs of time frames at different pitch shifts and handle key variations, and then the Hough Transform to find the best alignment and handle tempo variations.

❶Hamming Similarity

- find % of bins that match between all pairs of time frames
- pitch shifts of ±10 for max key variation of ±5 semitones
- similarity for both regions of locally high and low energy

❷Hough Transform

- find best diagonal line in the binarized similarity matrix
- angles of $-45° \pm 5°$ for max tempo variation of ±20%
- similarity+alignment to handle tempo variations

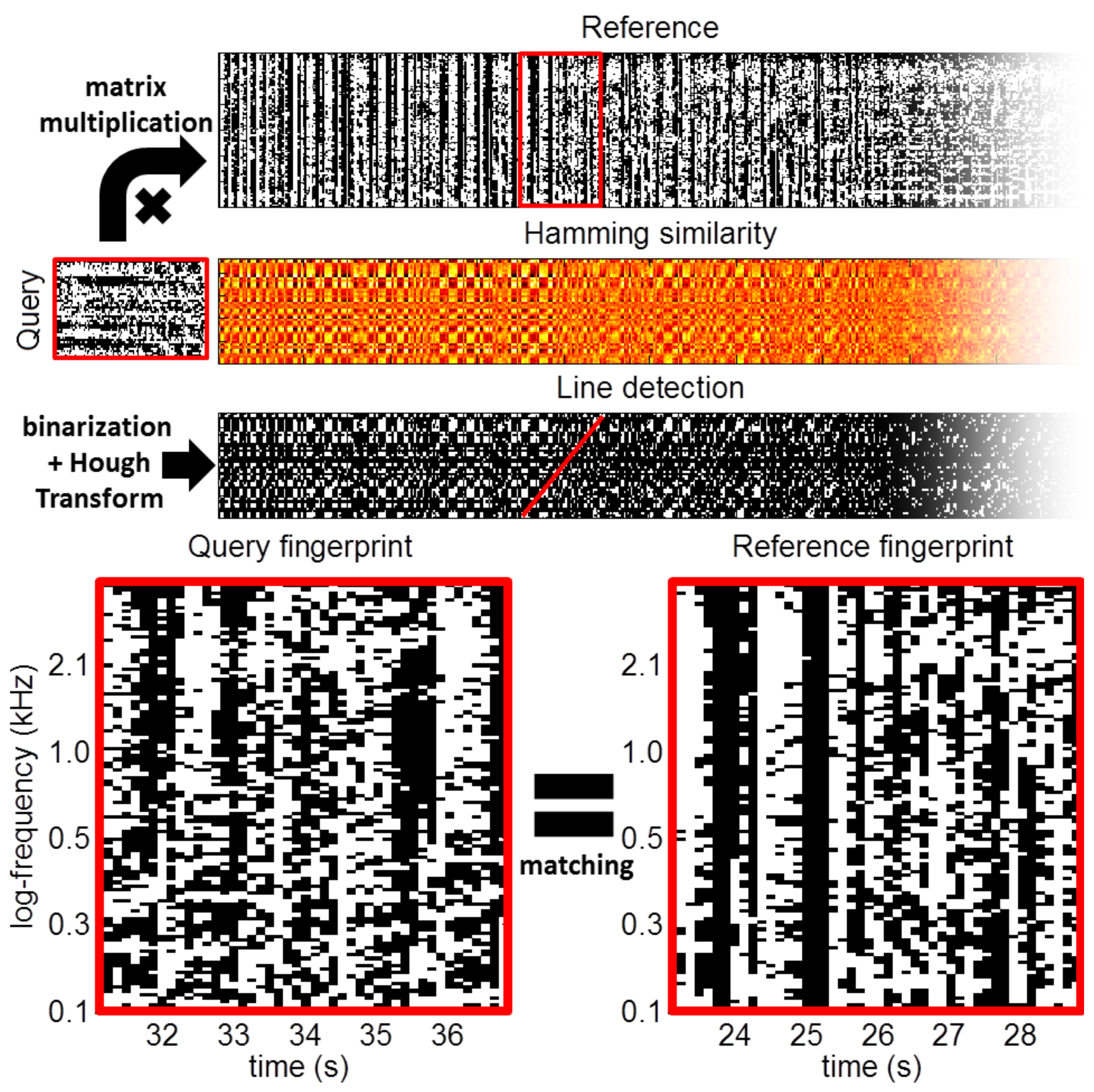

Figure 2: Overview of the matching stage. The query and the reference fingerprints are first compared by using the Hamming similarity. The similarity matrix is then processed to find the best alignment by using the Hough Transform.

EVALUATION - Dataset

We first build, for different artists of varied genres, a set of studio references, by extracting full tracks from studio albums, and two sets of live queries, by extracting short excerpts from live albums and from smartphone videos, using the same subset of songs from the set of studio references.

❶Studio References

- 10 different artists of varied genres (see Table 1)
- 389 full tracks from several studio albums
- durations from 01’04” to 11’06”

❷Live Queries

- 87 full tracks from live albums (see Tables 2 & 3)
- 87 audio tracks from smartphone videos (see Tables 4 & 5)
- 10 queries per tracks, 6 and 9 second length

artist	genre	#studio	#live
AC/DC	hard rock	36	60
Arcade Fire	indie rock	33	100
Bonobo	electronic	42	100
Eagles	rock	32	90
Foreigner	rock	29	100
Jefferson Airplane	psychedelic rock	65	40
Led Zeppelin	rock	40	80
Phoenix	alternative rock	38	100
Portishead	electronic	33	100
Suprême NTM	French hip hop	41	100
all	-	389	870

Table 1: Overview of the dataset.

EVALUATION - Results

We then evaluate our system on the database of reference fingerprints, by processing the live queries from the live albums and the smartphone videos, for durations of 6 and 9 seconds, and showing the results for different top-k matches (a match is declared if the correct reference is in the top-k matches).

	k=1	k=2	k=3	k=4	k=5
AC/DC	0.82	0.88	0.92	0.92	0.93
Arcade Fire	0.70	0.83	0.86	0.89	0.90
Bonobo	0.75	0.85	0.87	0.90	0.95
Eagles	0.88	0.90	0.93	0.97	0.97
Foreigner	0.71	0.82	0.85	0.87	0.93
Jefferson Airplane	0.60	0.70	0.78	0.80	0.83
Led Zeppelin	0.61	0.73	0.76	0.83	0.83
Phoenix	0.84	0.86	0.89	0.92	0.93
Portishead	0.78	0.87	0.89	0.91	0.92
Suprême NTM	**0.89**	**0.97**	**0.98**	**0.98**	**0.98**
all	0.77	0.85	0.88	0.90	0.92

Table 2: Live queries from live albums (6 seconds).

	k=1	k=2	k=3	k=4	k=5
AC/DC	0.92	0.95	0.95	0.97	0.97
Arcade Fire	0.84	0.92	0.94	0.96	0.97
Bonobo	0.83	0.89	0.92	0.92	0.96
Eagles	**0.93**	**0.97**	**0.98**	**0.99**	**0.99**
Foreigner	0.88	0.93	0.93	0.95	0.97
Jefferson Airplane	0.60	0.68	0.78	0.78	0.80
Led Zeppelin	0.74	0.81	0.84	0.85	0.90
Phoenix	0.88	0.92	0.93	0.97	0.98
Portishead	0.92	0.93	0.93	0.93	0.93
Suprême NTM	0.87	0.95	0.96	0.97	0.97
all	0.86	0.91	0.92	0.94	0.95

Table 3: Live queries from live albums (9 seconds).

	k=1	k=2	k=3	k=4	k=5
AC/DC	0.65	0.67	0.68	0.80	0.87
Arcade Fire	**0.75**	**0.85**	**0.87**	**0.91**	**0.93**
Bonobo	0.49	0.60	0.70	0.75	0.79
Eagles	0.62	0.69	0.73	0.78	0.80
Foreigner	0.50	0.64	0.70	0.78	0.83
Jefferson Airplane	0.23	0.28	0.33	0.35	0.43
Led Zeppelin	0.24	0.36	0.43	0.51	0.55
Phoenix	0.57	0.66	0.71	0.77	0.78
Portishead	0.64	0.77	0.80	0.82	0.86
Suprême NTM	0.23	0.32	0.40	0.48	0.53
all	0.51	0.60	0.66	0.72	0.76

Table 4: Live queries from smartphone videos (6 seconds).

	k=1	k=2	k=3	k=4	k=5
AC/DC	0.70	0.83	0.85	0.87	**0.93**
Arcade Fire	0.79	**0.86**	**0.89**	**0.91**	**0.93**
Bonobo	0.60	0.75	0.83	0.89	**0.93**
Eagles	0.70	0.77	0.88	**0.91**	0.91
Foreigner	0.68	0.83	0.86	0.86	0.88
Jefferson Airplane	0.40	0.53	0.55	0.60	0.63
Led Zeppelin	0.28	0.39	0.48	0.53	0.54
Phoenix	0.67	0.76	0.82	0.86	0.87
Portishead	**0.80**	**0.86**	0.87	0.87	0.87
Suprême NTM	0.30	0.42	0.45	0.51	0.55
all	0.61	0.71	0.76	0.79	0.81

Table 5: Live queries from smartphone videos (9 seconds).