AN AUDIO FINGERPRINTING SYSTEM FOR LIVE VERSION IDENTIFICATION USING IMAGE PROCESSING TECHNIQUES

Zafar Rafii

Northwestern University
Evanston, IL, USA
zafarrafii@u.northwestern.edu

Bob Coover, Jinyu Han

Gracenote, Inc.
Emeryville, CA, USA
{bcoover,jhan}@gracenote.com

ABSTRACT

Suppose that you are at a music festival checking on an artist, and you would like to quickly know about the song that is being played (e.g., title, lyrics, album, etc.). If you have a smartphone, you could record a sample of the live performance and compare it against a database of existing recordings from the artist. Services such as Shazam or SoundHound will not work here, as this is not the typical framework for audio fingerprinting or query-by-humming systems, as a live performance is neither identical to its studio version (e.g., variations in instrumentation, key, tempo, etc.) nor it is a hummed or sung melody. 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.

Index Terms— Adaptive thresholding, audio fingerprinting, Constant Q Transform, cover identification

1. INTRODUCTION

Audio fingerprinting systems typically aim at identifying an audio recording given a sample of it (e.g., the title of a song), by comparing the sample against a database for a match. Such systems generally first transform the audio signal into a compact representation (e.g., a binary image) so that the comparison can be performed efficiently (e.g., via hash functions) [1].

In [2], the sign of energy differences along time and frequency is computed in log-spaced bands selected from the spectrogram. In [3], a two-level principal component analysis is computed from the spectrogram. In [4], pairs of time-frequency peaks are chosen from the spectrogram. In [5], the sign of wavelets computed from the spectrogram is used.

Audio fingerprinting systems are designed to be robust to audio degradations (e.g., encoding, equalization, noise, etc.) [1]. Some systems are also designed to handle pitch or tempo deviations [6, 7, 8]. Yet, all those systems aim at identifying the same rendition of a song, and will consider cover versions (e.g., a live performance) to be different songs. For a review on audio fingerprinting, the reader is referred to [1].

Cover identification systems precisely aim at identifying a song given an alternate rendition of it (e.g., live, remaster, remix, etc.). A cover version essentially retains the same melody, but differs from the original song in other musical aspects (e.g., instrumentation, key, tempo, etc.) [9].

In [10], beat tracking and chroma features are used to deal with variations in tempo and instrumentation, and cross-correlation is used between all key transpositions. In [11], chord sequences are extracted using chroma vectors, and a sequence alignment algorithm based on Dynamic Programming (DP) is used. In [12], chroma vectors are concatenated into high-dimensional vectors and nearest neighbor search is used. In [13], an enhanced chroma feature is computed, and a sequence alignment algorithm based on DP is used.

Cover identification systems are designed to capture the melodic similarity while being robust to the other musical aspects [9]. Some systems also propose to use short queries [14, 15, 16] or hash functions [12, 17, 18], in a formalism similar to audio fingerprinting. Yet, all those systems aim at identifying a cover song given a full and/or clean recording, and will not apply in case of short and noisy excerpts, such as those that can be recorded from a smartphone in a concert. For a review on cover identification, and more generally on audio matching, the reader is referred to [9] and [19].

We propose an audio fingerprinting system that can deal with live version identification by using image processing techniques. 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 (e.g., title, lyrics, album, etc.). As computer vision is shown to be practical for music identification [20, 5], image processing techniques are used to derive novel fingerprints that are robust to both audio degradations and audio variations, while still compact for an efficient matching.

In Section 2, we describe our system. In Section 3, we evaluate our system using live queries against a database of studio references. In Section 4, we conclude this article.

This work was done while the first author was intern at Gracenote, Inc.

2. SYSTEM

2.1. 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.

2.1.1. Constant Q Transform

First, we transform the audio signal into a time-frequency representation. We propose to use a log-frequency spectrogram based on the Constant Q Transform (CQT) [21]. The CQT is a transform with a logarithmic frequency resolution, mirroring the human auditory system and matching the notes of the Western music scale, so well adapted to music analysis. The CQT can handle key variations relatively easily, as pitch deviations correspond to frequency translations in the transform.

We compute the CQT by using a fast algorithm based on the Fast Fourier Transform (FFT) in conjunction with the use of a kernel [22]. We derive a CQT-based spectrogram by using a time resolution of around 0.13 second per time frame and a frequency resolution of one quarter tone per frequency channel, with a frequency range spanning from C3 (130.81 Hz) to C8 (4186.01 Hz), leading to 120 frequency channels.

2.1.2. Adaptive Thresholding

Then, we transform the CQT-based spectrogram into a binary image. We propose to use an adaptive thresholding method based on two-dimensional median filtering. Thresholding is a method of image segmentation that uses a threshold value to turn a grayscale image into a binary image. Adaptive thresholding methods adapt the threshold value on each pixel of the image by using some local statistics of the neighborhood [23].

For each time-frequency bin in the CQT-based spectrogram, we first compute the median of the neighborhood given a window size. We then compare the value of the bin with the value of its median, and assign a 1 if the former is higher than the latter, and 0 otherwise, as shown in Equation 1. We use a window size of 35 frequency channels by 15 time frames.

$$\begin{aligned}\forall(i,j),\quad M(i,j) &= \underset{\substack{i-\Delta_i \le I \le i+\Delta_i\\ j-\Delta_j \le J \le j+\Delta_j}}{\text{median}} X(I,J)\\ \forall(i,j),\quad B(i,j) &= \begin{cases}1 & \text{if } X(i,j) > M(i,j)\\ 0 & \text{otherwise}\end{cases}\end{aligned} \tag{1}$$

The idea here is to cluster the CQT-based spectrogram into foreground (1), where the energy is locally high, and background (0), where the energy is locally low, as shown in Figure 1. This method leads to a compact fingerprint, that can handle noise degradations, while allowing local variations. It can be thought as a relaxation of the peak finder used in [4].

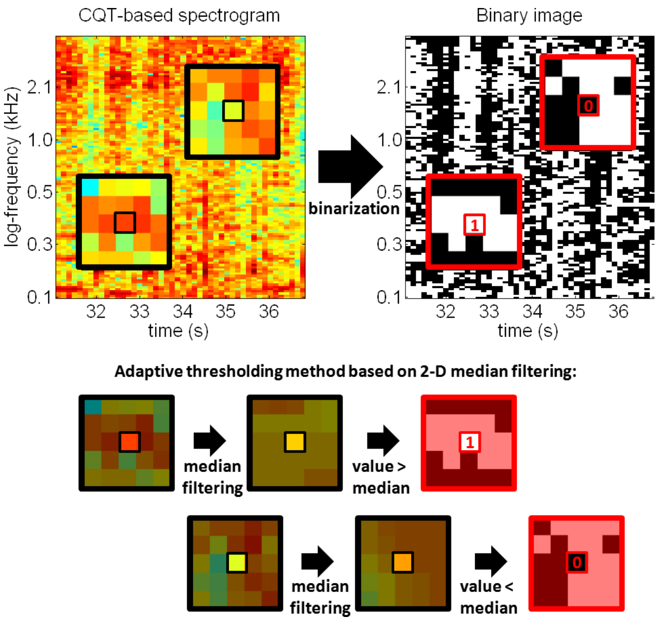

Fig. 1. Overview of the fingerprinting stage. The audio signal is first transformed into a log-frequency spectrogram by using the CQT. The CQT-based spectrogram is then transformed into a binary image by using an adaptive thresholding method.

2.2. 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.

2.2.1. Hamming Similarity

First, we compute a similarity matrix between the query and all the references. We propose to use the Hamming similarity between all pairs of time frames in the query and reference fingerprints. The Hamming similarity is the percentage of bins that matches between two arrays (1's and 0's) [24].

We first compute the matrix product of the query and reference fingerprints, after converting the fingerprints via the function $f(x) = 2x - 1$. We then convert the matrix product via the function $f^{-1}(x) = (x+1)/2$, and normalize each value by the number of frequency channels in one fingerprint. Each bin in the resulting matrix then measures the Hamming similarity between any two pairs of time frames in the query and reference fingerprints. We compute the similarity matrix for different pitch shifts in the query. We used a number of ± 10 pitch shifts, assuming a maximum key variation of ± 5 semitones between a live performance and its studio version.

The idea here is to measure the similarity for both the foreground and the background between fingerprints, as we believe that both components matter when identifying audio.

2.2.2. *Hough Transform*

Then, we identify the best alignment between the query and the references, which would correspond to a line around an angle of 45° in the similarity matrix, that intersects the bins with the largest cumulated Hamming similarity. We propose to use the Hough Transform, based on the parametric representation of a line as $\rho = x \cos\theta + y \sin\theta$. The Hough Transform is a technique used to detect lines (or other shapes) in an image by building a parameter space matrix and identifying the parameter candidates that give the largest values [25].

We first binarize the similarity matrix by using a threshold. We then compute the Hough Transform, and identify the (ρ, θ) candidate that gives the largest normalized value in the space parameter matrix, i.e., the highest overall Hamming similarity. We used a threshold of 0.6, a ρ range equal to the number of time frames in the reference fingerprints, and a θ range of around $-45° \pm 5°$, which corresponds to a number of ± 10 time shifts, assuming a maximum tempo variation of ± 20 % between a live performance and its studio version.

The goal here is to identify a short and noisy excerpt, typically recorded from a smartphone at a live performance, by comparing it to a database of studio recordings from a known artist. As far as we know, this is a problem that has not been really addressed before. Note that since we are dealing with relatively short queries ($<$10 seconds) and small databases ($\approx$ 50-100 songs per artists), we chose not to use hash functions as we want the identification to be as accurate as possible.

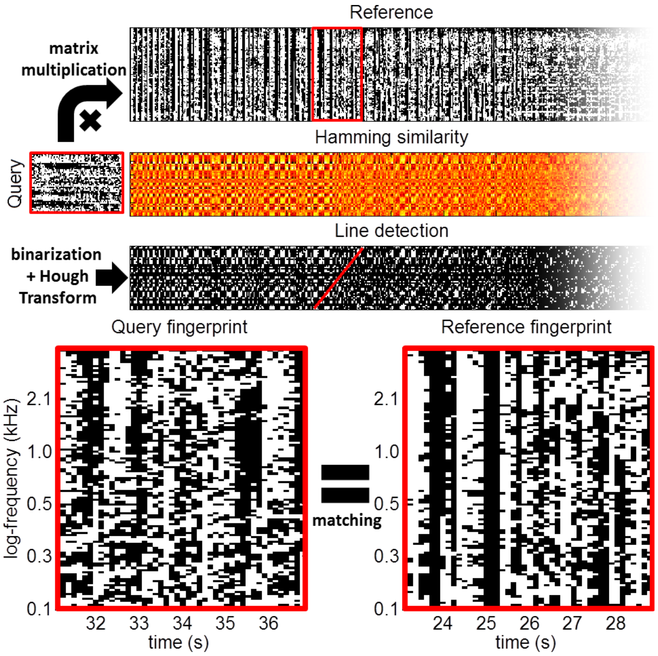

Fig. 2. Overview of the matching stage. The query and the reference fingerprints are first compared by using the Hamming similarity. The similarity matrix is then processed to find the best alignment by using the Hough Transform.

3. EVALUATION

3.1. 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.

3.1.1. *Studio References*

We first selected 10 different artists of varied genres. For each artist, we extracted a number of full tracks from several studio albums, for a total of 389 studio references. The durations of the audio files range from 01’04” to 11’06”. For each studio reference, we then derived a fingerprint using our system.

3.1.2. *Live Queries*

For each artist, we then extracted a number of full tracks from several live albums, using songs from the studio references, for a total of 87 full tracks. The durations of the audio files range from 02’56” to 09’37”. We also extracted the audio tracks from smartphone videos (posted on YouTube), using the same songs that were extracted from the live albums. The durations of the audio files range from 00’22” to 07’24”.

For each audio file, we selected 10 excerpts, for both the tracks from the live albums and the smartphone videos, for a total of 870 live queries. We used durations of 6 and 9 seconds. For each artist, we then computed the similarity between the fingerprints of a live query and all the studio references, and measured the accuracy using the top-k matches.

Compared with their studio versions, the live queries have noticeable audio variations (e.g., instrumentation, key, tempo, etc.). In addition, the live queries extracted from the live albums have occasional background noises (e.g., applause, screams, whistling, etc.), while the live queries extracted from the smartphone videos have also considerable audio degradations (e.g., compression, interference, saturation, etc.).

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.

3.2. 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).

3.2.1. Album Queries

As we can see in Tables 2 and 3, the system can achieve a high accuracy, showing that it is rather robust to audio degradations and audio variations. Furthermore, allowing more top-k matches and increasing the duration of the query generally improve the results. Incidentally, the system also often identifies the right location of the query in its correct reference.

Results for Bonobo and Foreigner are pretty good, even though many of their live performances have large tempo variations (e.g., up to $\pm 20\%$ for Bonobo) or key variations (e.g., up to ± 5 semitones for Foreigner) compared with their studio versions. Results for Eagles are rather high, as such artists are fairly consistent between their studio and live performances, while results for Jefferson Airplane are rather low, as such artists improvise a lot from one performance to another.

	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).

3.2.2. Smartphone Queries

As we can see in Tables 4 and 5, the system can still achieve a high accuracy in many cases, confirming that it is rather robust to audio degradations and audio variations.

Results for Bonobo and Suprême NTM are rather low, because of considerable audio degradations and audio variations (e.g., very noisy recordings or the performers speak to the audience). Results for Jefferson Airplane and Led Zeppelin are particularly low, because of larger audio variations, as such bands have long been separated, so recent recordings (i.e., from smartphones) come from reformations (e.g., Jefferson Starship) or reunions (e.g., Jimmy Page and Robert Plant).

	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).

4. CONCLUSION

We proposed an audio fingerprinting system that can deal with live version identification by using image processing techniques. The system can achieve high accuracy in many cases, showing that it is rather robust to audio degradations and audio variations, while still being relatively fast, as it takes about 10 seconds to process a query, when implemented in Matlab on a laptop with a 2.53 GHz processor and 8 GB of RAM.

5. REFERENCES

[1] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma, "A review of audio fingerprinting," *Journal of VLSI Signal Processing Systems*, vol. 41, no. 3, pp. 271–284, November 2005.

[2] Jaap Haitsma and Ton Kalker, "A highly robust audio fingerprinting system," in *3rd International Conference on Music Information Retrieval*, Paris, France, October 13-17 2002, pp. 107–115.

[3] Christopher J. C. Burges, John C. Platt, and Soumya Jana, "Distortion discriminant analysis for audio fingerprinting," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 11, no. 3, pp. 165–174, May 2003.

[4] Avery Li-Chun Wang, "An industrial-strength audio search algorithm," in *4th International Conference on Music Information Retrieval*, Baltimore, MD, USA, October 26-30 2003, pp. 7–13.

[5] Shumeet Baluja and Michele Covell, "Audio fingerprinting: Combining computer vision & data stream processing," in *32nd International Conference on Acoustics, Speech and Signal Processing*, Honolulu, HI, USA, April 15-20 2007, pp. II–213 – II–216.

[6] Rolf Bardeli and Frank Kurth, "Robust identification of time-scaled audio," in *AES 25th International Conference: Metadata for Audio*, London, UK, June 17-19 2004, pp. 1–12.

[7] Bilei Zhu, Wei Li, Zhurong Wang, and Xiangyang Xue, "A novel audio fingerprinting method robust to time scale modification and pitch shifting," in *18th International Conference on Multimedia*, Firenze, Italy, October 25-29 2010, pp. 987–990.

[8] Sebastien Fénet, Gaël Richard, and Yves Grenier, "A scalable audio fingerprint method with robustness to pitch-shifting," in *12th International Society for Music Information Retrieval*, Miami, FL, USA, October 2428 2011, pp. 121–126.

[9] Joan Serrà, Emilia Gómez, and Perfecto Herrera, "Audio cover song identification and similarity: Background, approaches, evaluation, and beyond," in *Advances in Music Information Retrieval*, Zbigniew W. Ras and Alicja Wieczorkowska, Eds., vol. 274 of *Studies in Computational Intelligence*, chapter 14, pp. 307–332. Springer-Verlag Berlin / Heidelberg, 2010.

[10] Daniel P.W. Ellis and Graham E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," in *32nd International Conference on Acoustics, Speech and Signal Processing*, Honolulu, HI, USA, April 15-20 2007, vol. 4, pp. 1429–1432.

[11] Juan Pablo Bello, "Audio-based cover song retrieval using approximate chord sequences testing shifts, gaps, swaps and beat," in *8th International Conference on Music Information Retrieval*, Vienna, Austria, September 23-27 2007, pp. 239–244.

[12] Michael A. Casey, Christophe Rhodes, and Malcolm Slaney, "Analysis of minimum distances in high-dimensional musical spaces," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 1015–1028, July 2008.

[13] Joan Serrà, Emilia Gómez, Perfecto Herrera, and Xavier Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1138–1152, August 2008.

[14] Meinard Müller, Frank Kurth, and Michael Clausen, "Audio matching via chroma-based statistical features," in *6th International Conference on Music Information Retrieval*, London, UK, September 11-15 2005, pp. 288–295.

[15] Frank Kurth and Meinard Müller, "Efficient index-based audio matching," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 382–395, February 2008.

[16] Peter Grosche and Meinard Müller, "Toward musically-motivated audio fingerprints," in *37th International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, March 25-30 2012, pp. 93–96.

[17] Matija Marolt, "A mid-level representation for melody-based retrieval in audio collections," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 10, no. 8, pp. 1617–1625, December 2008.

[18] Thierry Bertin-Mahieux and Daniel P. W. Ellis, "Large-scale cover song recognition using hashed chroma landmarks," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, October 16-19 2011, pp. 177–180.

[19] Peter Grosche, Meinard Müller, and Joan Serrà, "Audio content-based music retrieval," in *Multimodal Music Processing*, Meinard Müller, Masataka Goto, and Markus Schedl, Eds., vol. 3 of *Dagstuhl Follow-Ups*, chapter 9, pp. 157–174. Dagstuhl Publishing, Wadern, Germany, April 2012.

[20] Yan Ke, Derek Hoiem, and Rahul Sukthankar, "Computer vision for music identification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 20-26 2005, pp. 597–604.

[21] Judith C. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, January 1991.

[22] Judith C. Brown and Miller S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, November 1992.

[23] Mehmet Sezgin and Bulent Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–165, January 2004.

[24] Moses S. Charikar, "Similarity estimation techniques from rounding algorithms," in *34th ACM Symposium on Theory of Computing*, Montréal, Québec, Canada, May 19-21 2002, pp. 380–388.

[25] Richard O. Duda and Peter E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, January 1972.