Automatic music monitoring and boundary detection for broadcast using audio watermarking

Taiga Nakamura, Ryuki Tachibana, Seiji Kobayashi

Tokyo Research Laboratory, IBM Japan, Ltd.

ABSTRACT

An application of watermarking for automatic music monitoring of radio broadcasts is discussed. By embedding information into the music as a watermark before broadcasting it, it is possible to keep track of what music has been on the air at what time, and for how long. However, to effectively implement this application, the handling of content transitions is important, because the detection reliability deteriorates at the content boundaries. In this paper, a method of detecting content boundaries using overlapping detection windows is described. The most probable pattern of content transition is selected under the condition that detection results from multiple windows are available. The derived rules are represented using a finite state model, which is useful for detection in real time. Experimental results on FM radio broadcasts are also presented.

Keywords: Digital watermarking, audio watermarking, broadcast monitoring

1. INTRODUCTION

Watermarking is a method of inserting information into digital content by adding a signal to the content data so that the difference between the original content and the watermarked version is imperceptible to human senses. The information watermarked into the content data is extractable even after the data has been changed by various processes, unless the damage to the watermark is too great. This resistance to obliteration is called the “robustness” of the watermark.

The relationship between the robustness of a watermark and the level of content modification has been studied, both theoretically, by modeling the process of content modification as a degradation of the S/N ratio, and experimentally, by actually measuring the successful detection rate for a given watermarking method and content. Both approaches show that watermarks are useful as a means to add information to the content data and that the watermarks can be designed to remain extractable after post-processing.

Automatic content detection from broadcasts (also known as broadcast monitoring) is an application that makes use of these robust characteristics of watermarks[2]. In this application, the information, which is typically the content identification, is inserted into content before the content is used, and is detected from the received broadcast to determine what content has been on the air at what time, and for how long. When a sufficiently robust watermark is used, the information is extractable even if the content has been altered during the broadcasting process. This benefits content providers (video creators, musicians, or video and record companies), market analysts who want to know how often the content has been broadcast, and watchers or listeners who want to get information about the content they saw or listened to. Once the content identification has been obtained, it is much easier to link to external databases with other information about the content, such as its name or its creator.

In this paper, we focus on audio content and radio broadcasts. The procedure of the broadcast monitoring is:

1. Original audio content (usually a music file) and ID information are prepared.
2. The ID information is embedded into the content, producing a watermarked music file.
3. The file is compressed. This process is normally needed to reduce the size of the data , especially when a large number of music files are to be collected and stored in a database.

Taiga Nakamura: taiga@jp.ibm.com; IBM Tokyo Research Laboratory, 1623-14, Shimotsuruma, Yamato-Shi, Kanagawa-ken, 242-8502 Japan

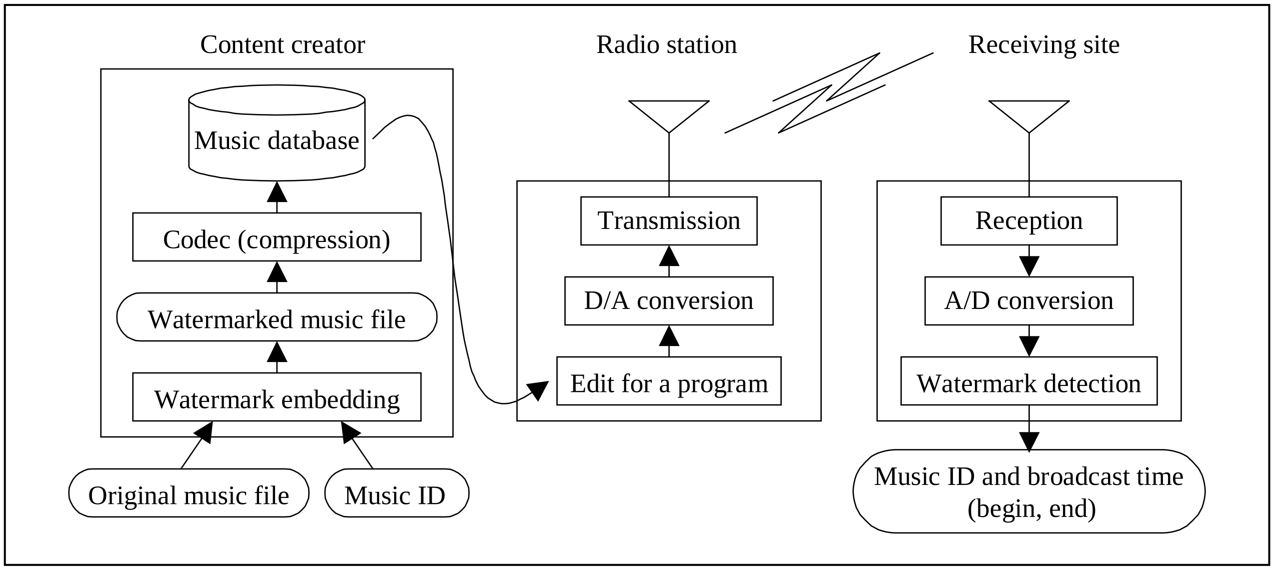

Figure 1: Diagram of the typical procedure of automatic music monitoring

4. A radio station selects the music files they want to use from the database and possibly edits them.
5. The music files are played on a radio program, D/A converted and transmitted over the radio.
6. After the radio signal is received, the program is A/D converted and sent to the watermark detector.
7. The detector extracts the information and records what content has been on the air, at what time, and for how long.

Figure 1 shows the process above. As already mentioned, such a monitoring application is not a new idea. However, there remain unsolved problems with the reliability and the time resolution.

When the watermark is detected directly from a music file, it can conveniently be assumed that the information is identical throughout the file. In music monitoring from broadcasts, the situation is very different. Usually the music is broadcast either in sequence or between other programs (news, sports, weather forecasts, etc.). Figure 2 illustrates this situation. If the watermark is detected at the content boundary, an error is more likely to occur because the watermark signal is interrupted or interfered with at the boundary.

Nevertheless, it is still required that the information from all music be correctly detected. Moreover, the position of the content boundaries determines the broadcast time of the content. For this application, it is important to determine when the broadcast of each song started and ended as accurately as possible. Therefore, the watermark detector must deal with the content transition at the boundaries to handle the change of the detection results, while still avoiding the output of incorrect watermark information.

In this paper, we first briefly describe the method of inserting a watermark into and extracting it from audio content, and the basics of detection using overlapping windows are defined. Then a method of determining the most probable content boundary is described. Experimental detection results from real FM broadcasts are also presented.

2. WATERMARKING METHOD

2.1. Basic algorithm and requirements for robustness

As we described in the previous section, audio content typically goes through various types of content modification during the broadcasting process. According to Figure 1, they include compression, trimming, A/D and D/A conversion,

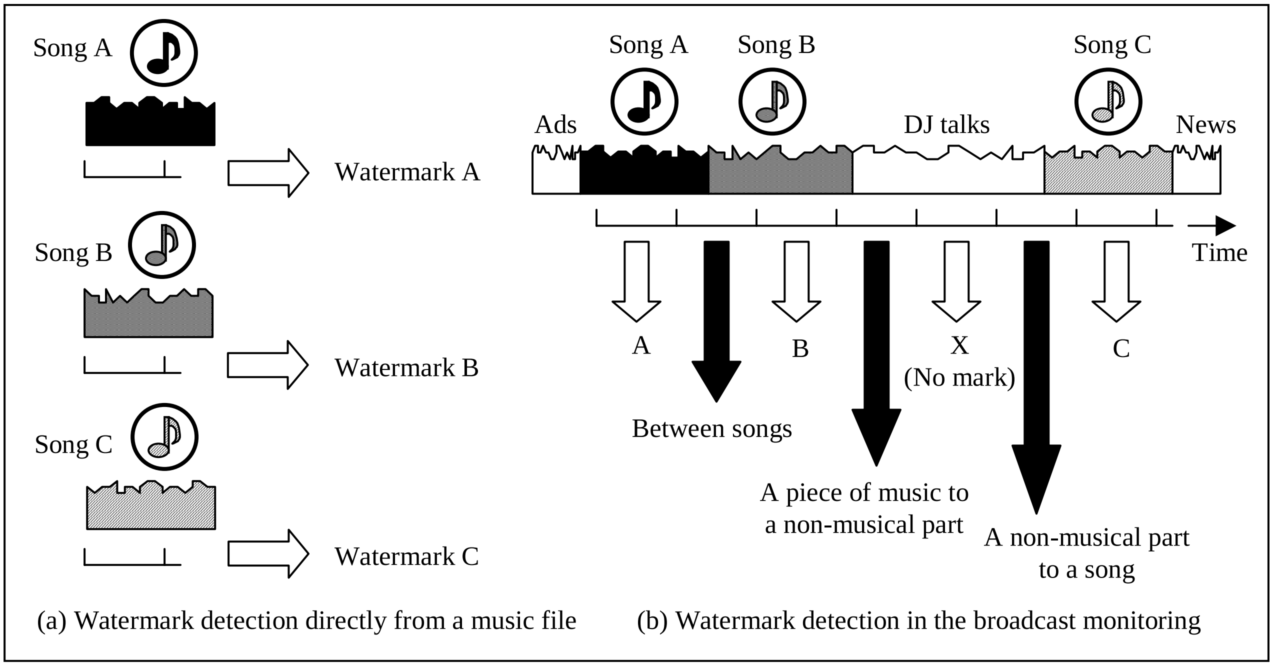

Figure 2: Characteristics of watermark detection in the monitoring application

and radio transmission and reception. In order to successfully extract the information at the radio receiver, we need to use a watermarking method that is at least sufficiently robust against these processes.

In this paper, we use the watermarking algorithm described in [1]. It is designed to be robust against time- and frequency-fluctuation, and has been experimentally confirmed to survive MPEG compression, additive noise, low-pass filtering, echo, etc.

We define the "detection window" as the content segment that is required for a single detection process. The time length of the detection window represents a minimum (or nominal) length required for correct watermark detection. In the watermarking system of [1], the detection window can start from an arbitrary position in a music file. The detection window is 30 seconds long for 64-bits of information.

Many watermarking methods other than [1] can be summarized in the same way as above, by the level of the robustness, the time length of the detection window, and the number of payload bits. The method of boundary detection described in this paper is applicable to other watermarking algorithms if they meet the robustness requirements.

2.2. Detection using overlapping detection windows

When a watermark is used in other applications, the detection process is usually done on detection windows that are aligned with no space between them, producing an output from each of them. In this approach, however, the time resolution of the broadcast monitoring is limited to the length of the detection window in the base case.

To improve the resolution, we introduce multiple detection sets that have their origins slightly offset from each other. Figure 3 illustrates this method. The overall effect is as though there are multiple detection windows that are overlapping with each other. If the overlap between neighboring windows is more than 50%, it is assured that all content boundaries will be included in multiple detection windows.[*]

[*] To be more accurate, if the overlapping ratio between neighboring windows is λ, the boundary will be included in at least $\lfloor 1/(1-\lambda) \rfloor$ windows.

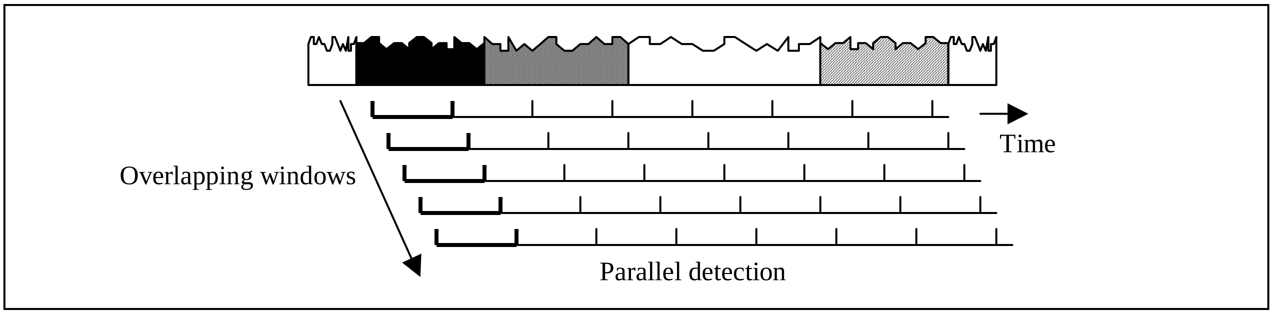

Figure 3: Illustration of detection using overlapping windows

If the detection window does not contain a content boundary, it is no different from ordinary watermark detection and the result should be correct, based on the reliability of the basic watermarking method. Otherwise, the result may be a wrong value that is produced by the mixture of fragmented watermark signals and content. To prevent incorrect results from being output, we integrate the results from multiple detection windows.

Although naive implementation using overlapping detection windows would require linearly increased computational cost compared to the non-overlapping detection, this can often be avoided by optimization. In our implementation, the increase of total computation is restrained to less than 7% above the base computations in [1], with an overlapping ratio around 90%, based on sharing the results of the Fourier transforms.

3. METHOD OF BOUNDARY DETECTION

3.1. Model of basic transition pattern

The effect of content transition on the watermark detection is regarded as degradation of the average SNR within any window that includes the boundary. Although the absolute value of the detection result is difficult to predict, since it depends on the initial signal strength, original content data, and the type of post processing, general characteristics can be reasonably assumed.

In general, there exist three basic transition patterns.

1. One song to another song (a watermark to another watermark)
2. A piece of music to a non-musical part (a watermark to unwatermarked)
3. A non-musical part to a song (unwatermarked to a watermark)

In this paper, a watermarked region is denoted by ‘A’ and an unwatermarked region is denoted by ‘X.’ When different watermark values need to be explicitly described, we use the characters A, B, C... in turn. Using these symbols, the three patterns are represented as (A,B), (A,X), and (X,A), respectively.

Figure 4 illustrates the qualitative characteristics of the detection probabilities for these basic patterns. When the window is occupied by one song, the probability of detection is as the same as for normal watermark detection. We assume that the changes of the probabilities that the watermark is correctly detected are monotonic with regard to the location of the boundary in a detection window. This assumption is reasonable in most cases, because the average SNR of the watermark signal within the window changes in proportion to the location of the boundary.

Note that in Figure 4, the bit error rate (the probability of getting the result ‘C’ in the pattern (A,B) and getting the result ‘B’ in the pattern (A,X) or (X,A)) is drawn rather high. Actually the effect of interruption and interference caused by the content transition depends on how the basic watermarking algorithm is implemented. For example, if the signal for synchronization is inserted independently from the information as in Tilki’s method[3], information is falsely determined

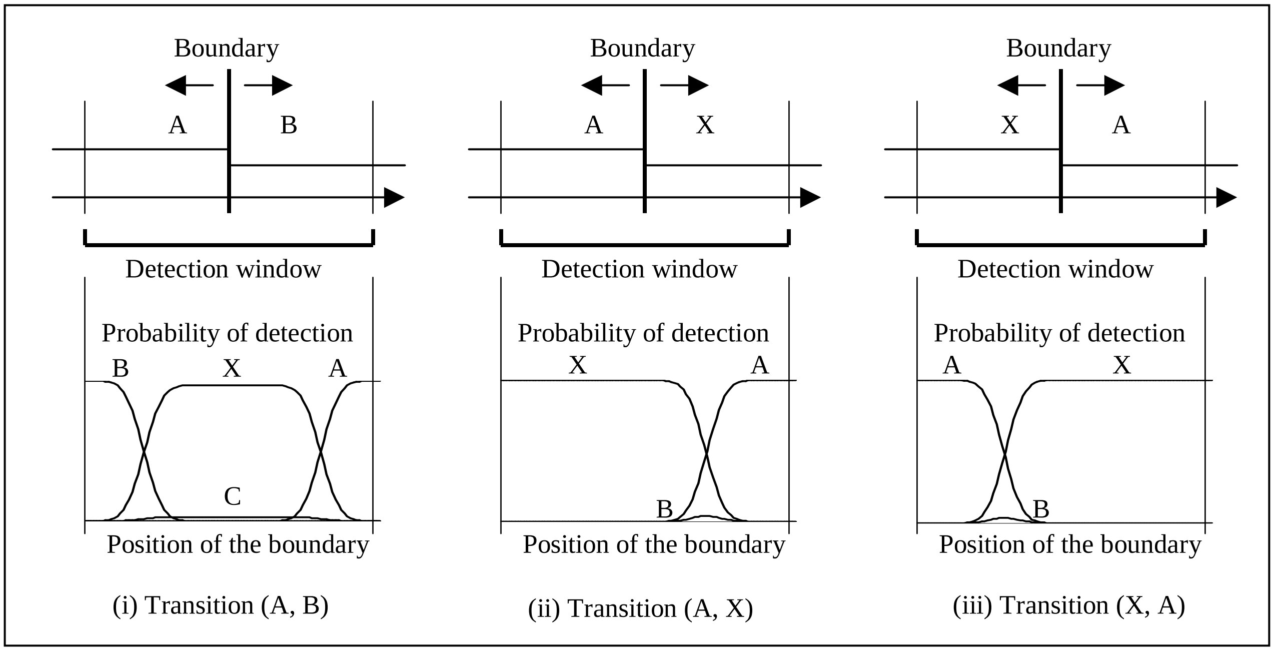

Figure 4: Basic transition pattern and the qualitative characteristics of the detection probabilities

to be extractable even if the signal for the information is destroyed at the boundary, which will result in erroneous output. In our implementation based on [1], the interference occurs only if two watermarked songs transit just at the border of the pattern blocks, and wrong output is rejected in the decoding process using the error correction code. As a result, no bit error was observed throughout our experiment.

3.2. Determining the optimum transition pattern

In principle, we select the most probable pattern of content transition for given detection results from multiple windows. The value of $D \in \mathcal{D}$ denotes the detection results from detection windows that are overlapping with each other. It is easily understood that the overall region where overlapping windows are spread is less than the 2 times the length of a detection window. For example, if there are three windows within this region, the set of possible detection results $\mathcal{D}$ is as below:

$$\begin{aligned}\mathcal{D} &= \{(X,X,X),(A,A,A),(X,X,A),(A,A,X),(A,A,B),(X,A,X),(X,A,A),(X,A,B),\\ &\quad (A,X,X),(A,X,A),(A,X,B),(A,B,X),(A,B,A),(A,B,B),(A,B,C)\} \qquad (1)\end{aligned}$$

The value of $H \in \mathcal{H}$ denotes the possible transition patterns within the given content region. Each transition pattern, H, corresponds to the content boundaries that exist in this region. In this paper, we assume that all watermark content should be on the air at least for the length of the detection window. If the broadcast length is shorter than that, it is treated no differently from unwatermarked content. With this assumption, we can limit the possible content transition patterns, $\mathcal{H}$, within the length of 2 times the window as below:[†]

$$\begin{aligned}\mathcal{H} &= \{(X),(A),(B),(C),(D),(X,A),(X,B),(X,C),(X,D),(A,X),(A,B),(A,C),(A,D),\\ &\quad (B,X),(B,A),(B,C),(B,D),(C,X),(C,A),(C,B),(C,D),(D,X),(D,A),(D,B),(D,C),(D,E),\\ &\quad (X,A,X),(X,A,B),(X,A,C),(X,A,D),\ldots,(D,C,B),(D,C,D),(D,C,E),(D,E,F),\\ &\quad (X,A,X,A),(X,A,X,B),(X,A,X,C),\ldots,(D,X,C,X),(D,X,D,X),(D,X,E,X)\} \qquad (2)\end{aligned}$$

[†] The number of patterns becomes larger if we take the position of the boundaries in relation to the windows into account.

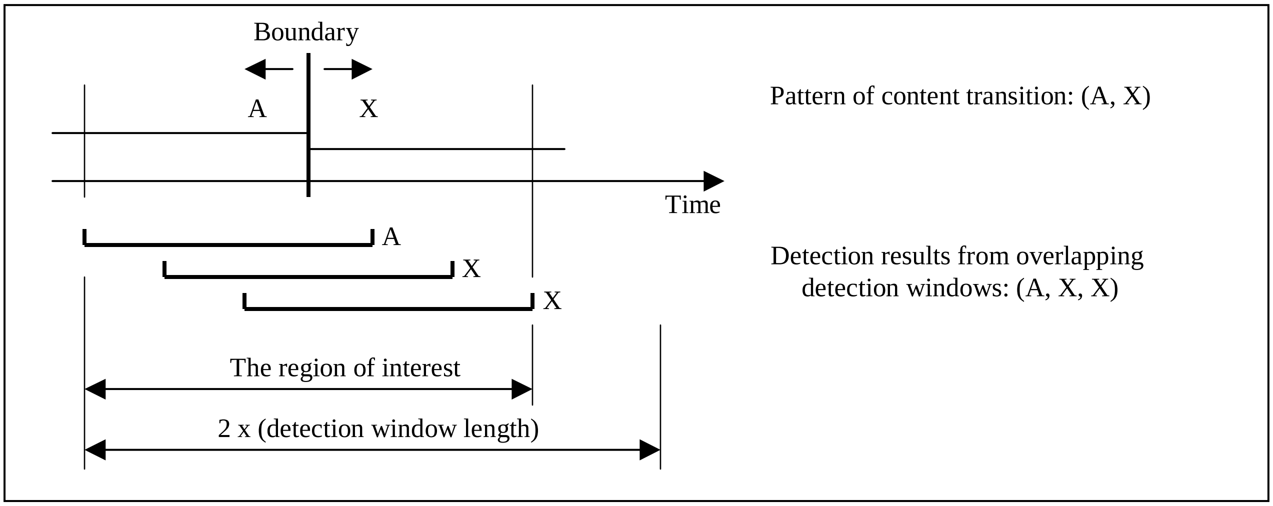
Boundary
A
X
Time
A
X
X
The region of interest
2 x (detection window length)
Pattern of content transition: (A, X)
Detection results from overlapping detection windows: (A, X, X)

Figure 5: Relationship between the detection results and the pattern of the content transition

Figure 5 illustrates the relationship between the detection results and the pattern of content transition. If the probability $Prob(D|H)$, the probability of getting the detection result D given the condition that the content transition pattern is H, is calculated for all possible values of $D \in \mathcal{D}$ and $H \in \mathcal{H}$, then we can determine the most probable content transition point using the equation:

$$Prob(H|D) = \frac{Prob(D|H)Prob(H)}{Prob(D)} \tag{3}$$

and select from all possible transition patterns $H \in \mathcal{H}$ the one that gives the greatest probability. As mentioned above, the absolute value is difficult to predict, but relative comparisons are possible based on the relationships illustrated in Figure 4.

For example, when we have the results from three overlapping windows, the decision rules are determined as in Table 1. The optimum rules depend on the probability of bit errors occuring at the boundaries. If the bit error rate is low, the optimum rule is that all detected watermarked should be considered to exist, while if the bit error rate is high, the detection patterns that include wrong watermark values become more probable. Similar rules can be obtained if the number of overlapping windows increases.

The method discussed above determines the existence of the content boundaries and their locations. To further improve the time resolution, an additional refinement process should be introduced. For example, the position of the boundary can be estimated by calculating the energy level of the watermark signal for a shorter period of time and identifying the inflection point. In this paper, we do not go into the details of these kinds of techniques, since they strongly depend on the implementation of each watermarking algorithm.

3.3. Representation using a finite state model

For practical use, the boundary detection often needs to be done in real time, where the results from the detection windows are obtained in sequence. In this situation, the final output should be evaluated successively and displayed as soon as a certain content is confirmed to be broadcast.

To achieve this, we describe the derived rules in a finite state model. In this model, we keep track of three variables during the detection process according to status of the detection results. They are used to evaluate the detected information and determine whether or not the information is derived from the actual broadcast signal.

current denotes the detection result of the most current detection window. It can be one of the values below:

- x : no watermark

Table 1: Rules applied for 3 detection windows

Observed detection pattern D	Selected H (Low bit error rate at the boundary)	Selected H (High bit error rate at the boundary)
$D = (X, X, X)$	$H = (X)$	$H = (X)$
$D = (A, A, A)$	$H = (A)$	$H = (A)$
$D = (X, X, A)$	$H = (X, A)$	$H = (X, A)$
$D = (A, A, X)$	$H = (A, X)$	$H = (A, X)$
$D = (A, A, B)$	$H = (A, B)$	$H = (A, B)$
$D = (X, A, X)$	$H = (X, A, X)$	$H = (X, A, X)$
$D = (X, A, A)$	$H = (X, A)$	$H = (X, A)$
$D = (X, A, B)$	$H = (X, B)$	$H = (X, A, B)$
$D = (A, X, X)$	$H = (A, X)$	$H = (A, X)$
$D = (A, X, A)$	$H = (A)$	$H = (A)$
$D = (A, X, B)$	$H = (A, B)$	$H = (A, B)$
$D = (A, B, X)$	$H = (A, X)$	$H = (A, B, X)$
$D = (A, B, A)$	$H = (A)$	$H = (A, B, A)$
$D = (A, B, B)$	$H = (A, B)$	$H = (A, B)$
$D = (A, B, C)$	$H = (A, C)$	$H = (A, B, C)$

- a : any watermark (also denotes the detected information.)

prev denotes the second most recent detection result from one or more detection windows that returned the same results. It can have one of the values below:

- S : start (Special value at the beginning of the detection process)
- x : no watermark (nonexistence is not confirmed)
- X : no watermark (nonexistence is confirmed)
- a : any watermark (existence is not confirmed)
- A : any watermark (existence is confirmed)

pprev denotes the third most recent detection result from one or more detection windows that returned the same results. It can be one of the values below:

- S : start (Special value at the beginning of the detection process)
- X : no watermark (nonexistence is confirmed)
- A : any watermark (existence is confirmed)

By definition, *pprev* must have a different value from *prev*.

For watermarks, capital characters represent the values that are confirmed to really exist and therefore can be output, while lower-case characters represent the unconfirmed values. When different information needs to be explicitly labeled, we use the characters A, B, C... or a, b, c... in turn.

Figure 6 shows the rules of boundary detection using finite state models, $(pprev, prev)$. We define four possible states for the watermarking algorithms that have low bit error rates at the boundary, and six states for the algorithms with high bit error rates. '*' means the value is arbitrary. For each state, the next operation is determined according to the value of *current*. The transition paths are also shown in Figure 6.

When the bit error rate at the boundaries is low, the decision rules are as follows:

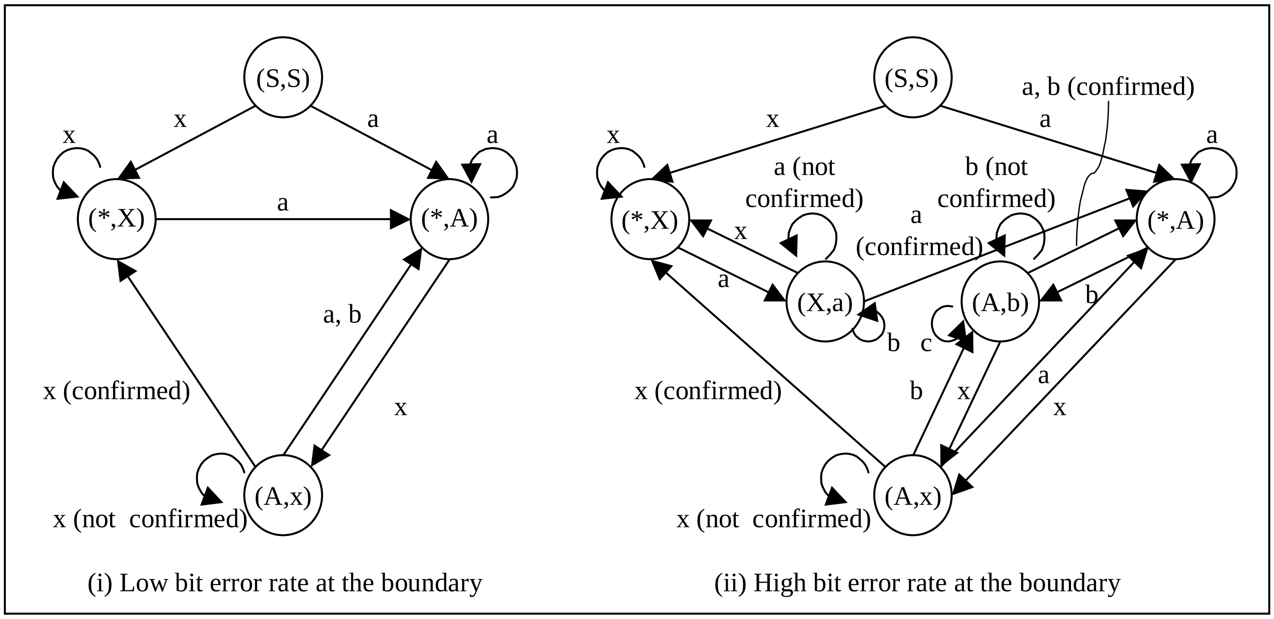

Figure 6: Finite state models to describe the decision rules for boundary detection

- (S, S)
  This is the special state at the beginning of the detection process.
  - If *current* is ‘x,’ the non-musical part ‘x’ is believed to exist, and the state becomes (*, X).
  - If *current* is ‘a,’ the music ‘A’ is believed to exist, and the state becomes (*, A).

  Since the result from the first detection window always exists, the states (S, x) and (S, a) are forbidden.
- (*, X)
  - If *current* is ‘x,’ a non-musical part is believed to be continuing, and the state remains (*, X).
  - If *current* is ‘a,’ the music ‘A’ is believed to exist, and the state becomes (*, A).
- (*, A)
  - If *current* is ‘x,’ the state becomes (A, x).
  - If *current* is ‘a,’ the music ‘A’ is believed to be continuing, and the state remains (*, A).
- (A, x)
  - If *current* is ‘x,’ a check is done to see whether a non-musical part can be confirmed to exist. It is confirmed if the current detection window is not overlapping the last ‘A’ window. In this case, the state becomes (*, X), and otherwise the state remains (A, x).
  - If *current* is ‘a,’ ‘x’ is determined not to exist and the state goes to (*, A).
  - If *current* is ‘b,’ ‘x’ is determined not to exist and the state goes to (*, A). (The symbol ‘A’ now represents the new watermark, previously denoted by ‘b.’)

When the bit error rate at the boundaries is high, the model becomes more complicated because it needs to deal with unconfirmed watermark values. The decision rules are as follows:

- (S, S)
  Same as when the probability of the bit error is low.

- (*, X)

  - If *current* is ‘x,’ a non-musical part is believed to be continuing, and the state remains (*, X).
  - If *current* is ‘a,’ the state becomes (X, a).

- (*, A)

  - If *current* is ‘x,’ the state becomes (A, x).
  - If *current* is ‘a,’ the music ‘A’ is believed to be continuing, and the state remains (*, A).
  - If *current* is ‘b,’ the state becomes (A, b).

- (A, x)

  - If *current* is ‘x,’ a check is done to see whether a non-musical part can be confirmed to exist. It is confirmed if the current detection window is not overlapping the last ‘A’ window. In this case, the state becomes (*, X), and otherwise the state remains (A, x).
  - If *current* is ‘a,’ ‘x’ is determined not to exist, and the state goes to (*, A).
  - If *current* is ‘b,’ ‘x’ is determined not to exist, and the state goes to (A, b).

- (X, a)

  - If *current* is ‘x,’ ‘a’ is determined not to exist, and the state goes to (*, X).
  - If *current* is ‘a,’ a check is done to see whether an ‘A’ part can be confirmed to exist. It is confirmed if the current detection window is not overlapping the last ‘X’ window. In this case the state becomes (*, A) and otherwise the state remains (X, a).
  - If *current* is ‘b,’ ‘a’ is determined not to exist, and the state remains (X, a). (The symbol ‘a’ now represents the new watermark, previously denoted by ‘b.’)

- (A, b)

  - If *current* is ‘x,’ ‘b’ is determined not to exist, and the state goes to (A, x).
  - If *current* is ‘a,’ ‘b’ is determined not to exist, and the state goes to (*, A).
  - If *current* is ‘b,’ a check is done to see whether a ‘B’ part can be confirmed to exist. It is confirmed if the current detection window is not overlapping the last ‘A’ window. In this case the state becomes (*, A). (The symbol ‘A’ now represents the new watermark, previously denoted by ‘b.’) Otherwise the state remains (A, b).
  - If *current* is ‘c,’ ‘b’ is determined not to exist, and the state goes to (A, b). (The symbol ‘b’ now represents the new watermark, previously denoted by ‘c.’)

As described in each rule, a confirmation process is performed to determine whether or not the detected value really came from the broadcast content. When the value of *prev* is not yet determined to exist (*prev* $= x$ or *prev* $= a$), the following process is performed:

- *prev* is determined to exist if the end point of the *pprev* part is earlier than the beginning point of the *current* window.
- *prev* is determined not to exist if the value of *prev* is different from the value of *current* and the end of the *pprev* part is later than the beginning of the *current* window.

Table 2: Results of the experiment on the FM broadcast

Content transition pattern	Correct detection ratio	Average time resolution	Worst time resolution
XA	100%	-3.9 sec	-13.0 sec
AB	100%	$+1.5$ sec	$+3.0$ sec

4. EXPERIMENT

To validate our method, we conducted an experiment using multiple music files that are compressed and transmitted using FM after being watermarked. The conditions were:

- Embed ISRC (International Standard Recording Code) information (5 ASCII + 7 digits) and some additional digits as 64-bit watermarks
- MPEG-2 AAC 128 kbps compression and a real FM broadcast
- Detection window of 30 seconds
- Overlapping ratio of 90%
- 2 music samples, each embedded with one of 10 different ISRCs
- 12 music samples, each embedded with one ISRC

In the experiments, 20 ($= 2 \times 10$) music samples were on the air individually in various programs, and 12 samples were broadcast in a single program in sequence. The length of the samples were approximately from 3 to 5 minutes. All results were compared to the cue sheets, the official record of broadcast songs as recorded at the ratio station by hand.

Table 2 shows the detection results. It indicates that all songs were correctly detected, with better time resolution than the length of the detection window, 30 sec. The time resolution of (X,A) was worse than (A,B). This is because our implementation had a tendency for a non-musical part 'X' part to be evaluated more quickly than a music part.

5. CONCLUSION

We have discussed automatic music monitoring from broadcasts using audio watermarking. Music files are embedded with identification bits as watermarks, to be detected with radio receivers. Accurate identification and accurate time resolution are required. For radio programs, music is on the air either in sequence or between other non-musical segments. The reliability decreases if the watermark is extracted at the boundary of two music files or musical and non-musical segments. Conventional methods do not take this into account, and therefore tend to output the wrong times for the boundaries.

The method proposed in this paper is designed to deal with multiple music files and output the correct ID values with good time resolution. In essence, the watermark detection process is performed in parallel, so that the music boundaries are contained in multiple detection windows that are overlapping with each other. The detector can determine whether or not the detected value has really been broadcast using these results by selecting the most probable boundary. In some cases, the optimum rules depend on how often bit errors occur at the boundaries. If the basic watermarking algorithm tends to output erroneous results with high probability when the watermark signal is interrupted or interfered with, the detector should filter the incorrect results to prevent them from being displayed.

The derived decision rules can be represented as a finite state model that covers all transition patterns of detection results. This representation is useful for real time monitoring, where the detection results are obtained in sequence and the output should be displayed as soon as the content is confirmed to be actually broadcast.

The experimental results show that our method is practically applicable to automatic music monitoring. Further improvement is required to achieve better resolution.

REFERENCES

1. Ryuki Tachibana, Shuichi Shimizu, Seiji Kobayashi, and Taiga Nakamura, “An audio watermarking method robust against time- and frequency-fluctuation,” *SPIE Conference on Security and Watermarking of Multimedia Contents III*, 2001.
2. Jaap Haitsma, Michiel van der Veen, Ton Kalker and Fons Bruekers, “Audio Watermarking for Monitoring and Copy Protection,” *Proceeding of ACM MM 2000*, 2000.
3. J. F. Tilki and A. A. L. Beex, “Encoding a hidden digital signature onto an audio signal using psychoacoustic marking,” *7th International Conference on Signal Processing Application & Technology*, 1996.