

MusicXML への電子透かし埋め込み方式の提案

渡 邊 集[†], 加 藤 貴 司[†]
 ベッド バハドゥール ビスタ[†] 高 田 豊 雄[†]

近年, Web における XML を用いた楽譜情報配信に対する次世代基盤技術である MusicXML が注目されている. 本論文では, まず, MusicXML フォーマットの詳細な分析に基づく MusicXML への電子透かし方式を提案する. 次に, 電子透かし埋め込みシステムの開発について述べ, 最後に, 埋め込み可能容量, 改竄耐性, コンテンツ品質, 対クリッピング耐性の観点から, 提案した MusicXML への電子透かし埋め込み手法の詳細な評価について述べる.

A Proposal on a Watermarking Scheme for MusicXML

ATSUMU WATANABE,[†] TAKASHI KATOH,[†] BHED BAHADUR BISTA[†]
 and TOYOO TAKATA[†]

MusicXML, which is considered a next generation fundamental technology for distribution of musical scores on the Web using XML, is receiving attention. In this paper, we propose a watermarking scheme for MusicXML (Musical Plan version) based on its detailed analysis. We also developed a web-based watermarking embedding/extracting system and evaluate the proposed embedding scheme in light of embedding capacity, tamper proofness, contents quality and anticlipping proofness.

1. はじめに

近年, インターネットの爆発的な普及により, HTML をはじめとした様々なコンテンツの配信が行われている. 今日では, Web において配信されるコンテンツは HTML や XML 等のテキスト情報だけでなく, PNG や JPEG 等の静止画像, MP3 や Ogg 等の音楽情報, MPEG や Windows Media Video 等の動画等, 多岐にわたる. その中で, Web における楽譜に基づく音楽情報の配信に対する次世代基盤技術として研究開発された, MusicXML¹⁾ というフォーマットによる楽曲の配信が行われている. MusicXML には, 現在のところ, Recordare L.L.C. (Recordare Limited Liability Company) が開発した Recordare MusicXML²⁾ と, 有限会社ミュージカル・プランが開発した Musical Plan MusicXML¹⁾ が存在し, それぞれに互換性はない. 本研究を始めるにあたり, それら 2 つの MusicXML について, 楽譜に基づく音楽情報の Web 配信という観

点から比較を行った.

一般に楽譜の持つ役割として, 1. 演奏情報の伝達, という側面と, 2. 審美的な側面が存在する. ここで審美的な側面について説明する. 一般に楽譜において記号の配置等に演奏上の意味の変わらない範囲で自由度がある場合がかなり存在する. たとえば, 1 つの五線上に 2 つの旋律が存在するとき片方の旋律の休符をどの高さに配置するか, 連鉤の傾きを前後の音符の配列からどのような傾きとするか等であり, 楽譜の記述者が審美的な観点からそれらを決定する. 2 つの MusicXML の仕様の差異について述べることは紙数の制限上困難であるためここでは省略するが, 楽譜の審美的な側面を考慮した場合, 記号のレイアウト情報をより詳細に記述することのできる Musical Plan MusicXMLの方が優れていると判断した.

また, 楽譜を単なる画像データではなく, XML 形式で公開/配布する理由は, 受け取った側での容易な音楽再生を可能とするためである. Musical Plan MusicXML は, 無料で配布されているプラグイン³⁾ を導入することによってウェブブラウザを介しての楽

[†] 岩手県立大学大学院ソフトウェア情報学研究所
 Graduate School of Software and Information Sciences,
 Iwate Prefectural University
 現在, 株式会社インターネットイニシアティブ
 Presently with Internet Initiative Japan Inc.

ここで述べているのはあくまでも MusicXML の記述能力の比較に関する著者の判断であって, コンポーザの楽譜記述能力の比較ではない点に注意されたい.

譜の閲覧、音楽再生を行うことが可能であるのに対し、Recordare MusicXML では少なくとも研究開始時には開発中のプロジェクト⁴⁾は存在したものの表示/再生用無料配布ソフトウェアは存在しなかった。したがって、基となる楽譜作成/編集ソフトウェア(コンポーザ)の普及具合の観点から見ると Recordare 社のコンポーザが世界的にも普及しているが、楽譜情報の Web による流通基盤という観点からは Musical Plan MusicXML の方が優れている。

以上の 2 つの理由により本研究では、Web における楽譜に基づく音楽情報配信フォーマットとして Musical Plan MusicXML (以下、MusicXML) を研究対象として選んだ。この Musical Plan MusicXML は特に日本国内の学校教育の現場で広く用いられている⁵⁾。

こういったマルチメディアコンテンツの配信が進む一方で、不正な複製品の流出等、著作権保護上の問題が深刻化している。これはデジタルデータが、まったく劣化のない複製を容易に行うことが可能であるという特徴に起因した問題であり、MusicXML で記述された音楽コンテンツにおいても同様の問題をかかえている。さらに、最近では一般の利用者がデジタルデータの編集や加工を行うことのできる環境が整いつつあり、著作権を示す情報をコンテンツ内に単純に含ませるようにしただけでは、編集加工処理によって著作権情報を容易に削除、改竄することが可能である。そのため、改竄によって著作者の特定が困難となり、不正利用を発見したとしても証拠を提示することができない。

近年、これら著作権保護上の問題の解決策の 1 つとして電子透かしと呼ばれる技術が注目されている。これはデジタルコンテンツ(主情報)内に、人間が知覚できないように別の情報(副情報)を付加・多重化する技術である⁶⁾。さらに、電子透かし技術においては主情報と副情報が一体化しており、主情報を劣化させることなく副情報を取り除くことは非常に困難である。そのため、不正利用を発見した際に副情報を確認することで、不正利用の証拠を提示することが可能となる。

2. 研究目的

MusicXML は従来のデジタルコンテンツと同様、不正な複製行為に対する耐性をまったく持っていない。そのため、MusicXML を用いた音楽情報を配信するにあたり、著作権の保護は重要な課題であると考えられる。

そこで本研究では、MusicXML に対する電子透かし

埋め込み手法を提案し、Web インタフェースを介して電子透かしの埋め込み/抽出を行うシステムを開発する。このシステムを用いて各埋め込み手法に対する詳細な評価を行い、これらの手法の有効性を示し、また適用範囲を示す。

3. MusicXML とその特徴

3.1 MusicXML の構成

MusicXML は、Web における楽譜に基づく音楽情報の表現および楽譜配信に対する次世代基盤技術として研究開発されたフォーマットである¹⁾。MusicXML は XML によって記述されており、マークアップによって構成されている。また、XML では、マークアップにおける開始タグ要素には属性と属性値を持つことが可能であり⁷⁾、MusicXML においても、いくつかの規定された属性と属性値を持つ。なお、MusicXML ではすべてのデータをマークアップで記述する。

MusicXML では楽譜上のすべての記号は XML におけるマークアップによって表現され、すべてのマークアップには楽譜上の位置情報を示す座標値が属性によって記述される。楽譜はページ、ブロック、五線と、音符や休符等の音楽記号によって構成されている。これらの要素をそれぞれ Page, Block, Line, Note, Rest と呼ぶ。また、MusicXML を XML Document Object Model (DOM) ツリーで表した際に葉となる、Note や Rest 等の記号は、Line 要素の子要素として、空要素で記述される。なお、空要素とは、文字データと子要素を持たない要素である。

また、MusicXML には異名同義要素は存在せず、要素間の包含関係についても厳密な決まりがある。たとえば、Block 要素が Page 要素の子要素として現れることはあっても、その逆のパターンが現れることはない。

図 1 に MusicXML の一部を記述例として示す。なお、MusicXML において生成されるマークアップはかなりの分量となるため、図においては一部を省略する。図中のマークアップにおける ... は省略であることを示す。

3.2 MusicXML における座標

MusicXML には座標の概念が存在し、座標により記号が画面上のどの位置に描画されるかを指示する。ルート要素の原点は画面左上隅であり、 x 座標値の増加に従って右方へ移動し、 y 座標値の増加に従って下方へ移動する。空要素以外のルートより下の要素は、親要素の起点を原点とした座標値で位置を指定される。図 2 に構成図を示す。なお、空要素すなわち各音楽記号の位置は、前記号からの相対 x 座標と音高を表す

```

<?xml version="1.0" encoding="UTF-8"?>
<MusicXML generator="MUSIC PRO XML Adaptor" version="1.0.0" ...>
<Score title="hohoghe march" data="hage" p_x0="0" p_y0="0" ...>
<Part partindex="1" namelongs="Violin" soundfile="MU10" ...>
<Part partindex="2" namelongs="Harp" brace_g="here" ...>
<Page blockcount="3" blockno="hide">
<Titleblock p_x0="1" p_y0="0" p_x1="520" p_y1="39"/>
<Block p_x0="1" p_y0="39" p_x1="520" p_y1="276" ...>
<Line p_x0="51" p_y0="0" p_x1="519" p_y1="96" ...>
<Clef s_x="20" s_y="2"/>
<Key s_x="6" s_y="4" key="sharp" keycount="2"/>
<Tempo s_x="8" s_y="12" value1="50"/>
<Beat s_x="8" s_y="4" upr_1="4" lwr="0"/>
<Rest s_x="80" s_y="5" length="whole"/>
<Notelength s_x="2" s_y="9" type="tenuto"/>
<Bar_line s_x="174"/>
<Rest s_x="52" s_y="5" length="whole"/>
<Dynamics s_x="48" s_y="4" type="mf"/>
<Bar_line s_x="108"/>
<Note s_x="8" s_y="8" dirc="lwr" length="half" ...>
<Slur s_x="0" s_y="9" id="39" width="50" height="0" ...>
<Note s_x="50" s_y="8" dirc="lwr" length="eighth" ...>
<Note s_x="12" s_y="6" dirc="lwr" length="eighth" ...>
<Note s_x="12" s_y="3" dirc="lwr" length="eighth" ...>
<Note s_x="14" s_y="6" dirc="lwr" length="eighth" ...>
<Note s_x="12" s_y="8" dirc="lwr" length="eighth" ...>
<Bar_line s_x="118"/>
</Line>
<Line p_x0="51" p_y0="96" p_x1="519" p_y1="237" ...>
<Clef s_x="20" s_y="6" gstaff="lwr" att="f"/>
<Clef s_x="0" s_y="2"/>
<Key s_x="6" s_y="4" key="sharp" keycount="2"/>
<Beat s_x="16" s_y="4" gstaff="lwr" upr_1="4" lwr="0"/>
<Beat s_x="0" s_y="4" upr_1="4" lwr="0"/>
<Dynamics s_x="13" s_y="12" gstaff="lwr" type="mf"/>
<Note s_x="19" s_y="-3" gstaff="lwr" dirc="lwr" ...>
<Rest s_x="0" s_y="5" length="eighth"/>
<Note s_x="14" s_y="1" gstaff="lwr" dirc="lwr" ...>
<Note s_x="0" s_y="6" gstaff="lwr" length="eighth" ...>
<Note s_x="12" s_y="4" gstaff="lwr" dirc="lwr" ...>
<Note s_x="0" s_y="8" gstaff="lwr" length="eighth" ...>
<Note s_x="12" s_y="6" gstaff="lwr" dirc="lwr" ...>
<Note s_x="0" s_y="-1" length="eighth" beam_id="4" ...>
<Note s_x="12" s_y="8" gstaff="lwr" dirc="lwr" ...>

```

図 1 MusicXML 記述例の一部 (... は省略を表す)

Fig. 1 A Part of an example of MusicXML document.

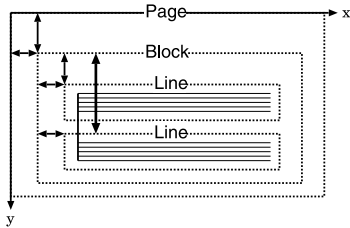


図 2 MusicXML の構成

Fig. 2 Structure of MusicXML.

y 座標によって指定される。

また、各音楽記号では基本的にマークアップの出現順序が各記号に対応する楽音の出現順序となるが、x 座標に負値が指定された場合にはその限りではない。しかし、ブラウザに組み込まれるプラグインや 3.4 節に述べるコンポーザでは演奏時においてマークアップの出現順序に従って音を発生させる。これは楽譜と旋律に矛盾を生じるため、これらのコンポーザが x 座標に負値を指定することは（装飾記号等も含めて）ない。



図 3 連鉤 Fig. 3 Beam.



図 5 アルペジオ Fig. 5 Arpeggio.



図 7 タイ Fig. 7 Tie.



図 4 和音 Fig. 4 Harmony.



図 6 連符 Fig. 6 Tuplet.

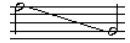


図 8 グリッサンド Fig. 8 Glissando.



```

<Note tie_id="1"/>
<Slur id="1"/>
<Note tie_id="1" beam_id="2" tail_id="3"/>
<Note beam_id="2" tail_id="3"/>
<Note beam_id="2" tail_id="3"/>

```

図 9 タイ、連鉤、連符のグループ化の例

Fig. 9 Examples of groupings for Tie, Beam and Tuplet.

3.3 グループ ID

MusicXML で用いられている各音楽記号は、以下の場合においてグループを形成する。

- 連鉤 (図 3)
- 和音 (図 4)
- アルペジオ (図 5)
- 連符 (図 6)
- スラー、タイ (図 7)
- グリッサンド (図 8)

グループの開始、終了に関して *cond_groupname* という属性があり、たとえば連鉤開始の場合 *cond_beam="bgn"*、連鉤の途中であれば *cond_beam="mid"*、連鉤の終了であれば *cond_beam="end"* という属性が付加される。しかしながら、これだけでは、たとえば 1 つの和音に複数のスラーが同時につくような場合に区別がつかなくなる。そのため、各グループには ID が割り振られる。グループはこの ID とグループの種類によって分類される。ID のとりうる値は 0 ~ 255 の整数値であり、グループの種類が異なれば ID 値が同じでも別のグループを表す。また、*cond_groupname="end"* 属性によってグループが終了した後は同じグループ ID 値が割り振られた場合でも、それは別のグループと見なされる。

タイ、連鉤、連符のグループ化の例を図 9 に示す。図中において、*<Note>* 要素は音符、*<Slur>* 要素はタイを表す。また、*<Note>* 要素の *beam_id* 属性は連鉤に使用される ID、*tie_id* 属性はタイに使用される ID、*tail_id* 属性は連符に使用される ID であり、*<Slur>* 要素の *id* 属性はタイに使用される ID であ



図 10 MUSIC PRO for Windows V4 PLUS

Fig. 10 Screenshot of MUSIC PRO for Windows V4 PLUS.

る。また図では繁雑さを避けるため開始，終了属性は省略している。

3.4 MusicXML を生成可能なコンポーザ

コンポーザとは，あるコンテンツを制作する際に利用するソフトウェアの総称であるが，ここでは楽曲を制作する際に利用するソフトウェアを指すものとする。

現在著者の知る限り MusicXML を生成することのできるコンポーザとしては，ミュージカル・プラン社によって開発，販売が行われている MUSIC PRO for Windows V4 シリーズ⁸⁾ のみである。この中の 1 つである MUSIC PRO for Windows V4 を起動すると図 10 のような画面が表示される。図中右部の音楽記号パレットから記号を選択し，譜面上の任意の点をクリックすることで記号の配置を行うことができる。これにより，楽譜の記述を行うことが可能である。

このコンポーザの特徴として，以下にあげるものがある。

- (1) マークアップの出現順序，属性値の出現順序，インデントはつねに一定である。
- (2) 和音の構成音はつねに低音から高音へ順に配置される。
- (3) 特徴 (1)，(2) の順序はファイルの保存時に整形される。
- (4) x 座標を機械的に整形する清書機能を有する。

4. 関連研究

これまでに，MIDI 符号⁹⁾ を用いた楽曲データに対する電子透かし埋め込み方式として，松本らによる演奏音を変化することなく Standard MIDI File (SMF) へ情報を埋め込む方式¹⁰⁾，岩切らによる SMF の半雑音化手法¹¹⁾ が報告されている。松本らによる方式では，和音の順序を入れ替えることによって情報を埋め込む。しかしながら，この方式を MusicXML へ応用

すると，3.4 節に述べたコンポーザの特徴 (2)，(3) により，コンポーザを用いてファイルを開き，保存し直すだけで埋め込んだ情報が損失するということが分かっている。また，岩切らの方式は SMF の音源制御パラメータ（ベロシティやデルタタイム等）に微細な変更を施すものであるため，元来，音源に関する概念のない MusicXML に対する応用は不可能であると考えられる。

また，XML 文書への情報埋め込み手法として，井上らによる 9 つの手法が報告されている¹²⁾。これらの手法は XML が構造化文書であるという点に着目しており，その表現の冗長性を利用する。また，これらの手法は以下の 2 つのグループに大別される。

- (1) XML 文書の文法に着目する手法
- (2) XML 文書の論理構造に着目した手法

このうち，(1) に関する手法では，3.4 節に述べた特徴のうち，特徴 (3) によって，いずれもファイルを保存し直すだけで情報が損失してしまうことが分かっている。また，MusicXML では論理構造として 3.1 節に述べた，1) 異名同義要素が存在しない，2) 要素間の包含関係に厳密な決まりがある，という特徴と，3.2 節に述べた，マークアップの出現順序が各記号の出現順序となるという特徴を持っているため，それらに情報を埋め込む (2) にあげられた手法を応用することができない。

また，楽譜に対する電子透かしに関する研究として，Schmucker らによる研究¹³⁾ が存在する。その研究では楽譜を画像データと見なし，楽譜の意味を変えない範囲での透かし情報の埋め込み手法を提案している。しかしながら，XML のようなテキストデータとは異なり，画像データは本質的に冗長性の大きな，すなわち埋め込みによる歪みが目立ちにくいコンテンツであり，その手法をただちに MusicXML に適用することは困難である。





5. MusicXML への電子透かし埋め込み方式

本章では MusicXML への電子透かし埋め込み方式について説明する。

電子透かしにおいては，悪意のある第三者が容易に透かしを除去できないようにしなければならない。また，コンテンツの価値を損なうことがあってはならないため，楽譜，旋律ともに歪みを生じさせない必要がある。

これらをふまえ，本論文では，MusicXML への情報埋め込み手法として，以下の 4 手法を提案する。これらの手法は，手法 2 以外は楽譜，旋律ともに歪みを生じさせない手法である。

表 1 記号の重複を利用した情報埋め込み例
Table 1 Examples of information embedding by employing symbol duplications.

Data	Element	Display
00	<Note s_x="44" s_y="4"/>	
01	<Note s_x="44" s_y="4"/> <Note s_x="0" s_y="4"/>	
10	<Note s_x="44" s_y="4"/> <Note s_x="0" s_y="4"/> <Note s_x="0" s_y="4"/>	
11	<Note s_x="44" s_y="4"/> <Note s_x="0" s_y="4"/> <Note s_x="0" s_y="4"/> <Note s_x="0" s_y="4"/>	

- (1) 記号の重複を利用した手法
- (2) 座標値の微細な操作を利用した手法
- (3) グループ ID 値の操作を利用した手法
- (4) 連符の y 座標を利用した手法

以下、これら 4 つの手法について詳説する。

5.1 記号の重複を利用した手法 (手法 1)

3.2 節に述べた特徴のうち、各音楽記号を示す空要素の x 座標が前記号からの相対 x 座標であり、 y 座標が音高を示すという点に着目した手法である。前記号と同一の要素の x 座標を示す属性の値を 0 としたうえで y 座標を示す属性の値を前記号と同一にし、前記号と連続して配置すると、これら 2 つの記号は完全に重なる。また、楽譜としての歪みが起こらないだけでなく、少なくとも現在流通しているブラウザに組み込まれるプラグイン (MusicX) やコンポーザ (MUSIC PRO シリーズ) によって演奏を行った際には音の生起するタイミングが同一となるため、旋律についても歪みは起こらない。

この手法では、この前記号と同一の要素の x 座標を示す属性の値を 0 としたうえで y 座標を示す属性の値を前記号と同一にし、前記号と連続して配置すると、これら 2 つの記号は完全に重なるという特徴を MusicXML の冗長性としてとらえる。この特徴を利用し、重複していない状態を 0、重複している状態を 1 とする等して情報を埋め込む。なお、3 つ以上の重複があった場合でもこの特徴が失われることはないため、たとえば表 1 のように、重複していない状態を 00、二重の状態を 01、三重の状態を 10、四重の状態を 11 とする等して情報を埋め込むことも可能である。ここで、表中の <Note> 要素は音符を表し、 s_x 属性

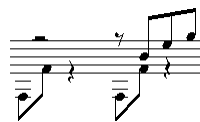


図 11 休符の y 座標の変化
Fig. 11 Shifting of y -Coordinate of Rest.

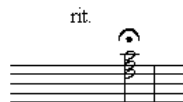


図 12 フェルマータとリタルダンド
Fig. 12 Fermata and Ritardando.

```
<Note s_x="284" s_y="6" dirc="lwr" length="half"
  type_add_beat="fermata" s_adbeat_x="-2"
  s_adbeat_y="-7"/>
<Note s_x="0" s_y="8" dirc="lwr" length="half"
  type_add_beat="fermata"
  s_adbeat_x="-2" s_adbeat_y="-5"/>
<Note s_x="0" s_y="10" dirc="lwr" length="half"
  type_add_beat="fermata"
  s_adbeat_x="-2" s_adbeat_y="-3"/>
```

図 13 フェルマータ記号の MusicXML 表現
Fig. 13 Expression of a Fermata symbol in MusicXML.

は x 座標、 s_y 属性は y 座標を表す。

5.2 座標値の微細な操作を利用した手法 (手法 2)

3.2 節で述べた MusicXML における座標について、操作を行っても楽譜の意味と旋律に歪みが生じない記号が存在する。この手法では、そのような記号に対して、以下にあげる記号の座標を操作し情報を埋め込む。休符、付加的な記号の y 座標 通常休符の y 座標は定まっているが、ピアノ譜等のように 1 つの五線上に 2 つの旋律が存在する場合等はその限りではない (図 11)。また、フェルマータやリタルダンド (図 12) 等のような付加的な記号についても、他記号と重なる等といった場合があるため、その y 座標にある程度の自由度が存在する。

なお、ここで付加的な記号と定義するものは、MusicXML の DTD (Document Type Definition) において要素として定義されておらず、対象となる要素の属性とその属性値によって自動的に付加される記号、および速度標語や速度変化記号、強弱記号、奏法記号、装飾記号、発想記号、楽器用記号等である。たとえば図 12 における右側の音符を MusicXML で表すと図 13 のようになっており、フェルマータ記号 (fermata) は音符記号への属性値として記述されている。

単独グリッサンドの x 座標 付加的な記号のうち、単独で現れるグリッサンド記号 (グリッサンドの



図 14 連鉤による符尾の長さの変化

Fig. 14 Variation of stem length due to connected by Beam.

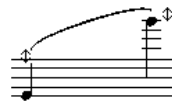


図 16 スラーにおける座標の変化

Fig. 16 Variation of Coordinate of a Slur symbol.

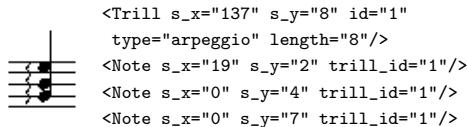


図 15 アルペッジョ記号の MusicXML 表現

Fig. 15 Expression of an Arpeggio symbol in MusicXML.

終了位置に音符がないもの)については例外的に扱わなければならない。単独で現れるグリッサンドは、*x* 座標を微細に操作しても楽譜としての意味や旋律に大きな歪みは生じないが、*y* 座標について操作を行うと終点の音が変わり、旋律に歪みが生じる。したがって、*y* 座標を操作して情報を埋め込むことはできないが、代わりに *x* 座標を操作し情報を埋め込むことができる。

連鉤の符尾の長さ 全音符(付点が付いたものを含む)以外のすべての音符には符尾(棒)と呼ばれる直線が描画される(文献 14)「符尾」の項)。符尾は 1 オクターブ(3.5 間)であることが一般的だが、連鉤の場合でかつ 1 オクターブにこだわらない方が美しいと判断される場合においてはその限りではない(図 14)。したがって、連鉤における符尾の長さを微細に操作することで情報を埋め込むことが可能となる。

アルペッジョ記号の長さ と *y* 座標 アルペッジョ記号はその上端から下端までの範囲内に対象となる音符が重なっていれば楽譜として意味をなす。その条件を満たす範囲内で、長さや *y* 座標を微細に操作することで情報を埋め込むことが可能である。なお、MusicXML におけるアルペッジョ記号は `type` 属性の値が `arpeggio` である `<Trill>` 要素の `id` 属性の値と、対象となる記号の `trill.id` 属性の値によってグループ化が行われて初めて旋律において意味をなすが、アルペッジョ記号自体は `<Trill>` 要素によってのみ表現されるため、ここでは付加的な記号とは見なさないこととした。図 15 にアルペッジョ記号の MusicXML 表現を例示する。

スラー、タイの *y* 座標 対象となる記号からスラーやタイ記号までの距離の大きさは、始点と対象記号との間の距離、終点と対象記号との間の距離が

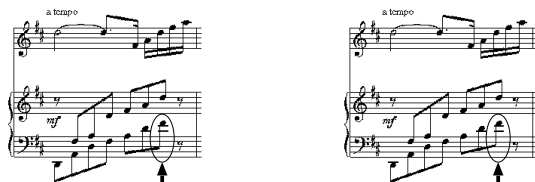


図 17 連鉤の符尾の長さの変化とそれともなう MusicXML 表現の変化の例

Fig. 17 Variation of Stem length due to connected by Beam and Corresponding Variation of MusicXML description.

同一である等一般的には一定であるが、この原則にこだわらない方がより美しいと判断される場合においてはその限りではない。そのため、*y* 座標を微細に操作して情報を埋め込むことが可能である。図 16 に、スラーにおける座標の変化を例示する。なお、図中の矢印記号は対象記号とスラーとの間の距離を表す。

譜面全体の *x*, *y* 座標 図 2 にあげた Page, Block, Line 等の要素について *x*, *y* 座標を微細に操作することで情報を埋め込んで楽譜の意味や旋律に歪みは生じない。ただし、同一 Page 内の Block、あるいは同一 Block 内の Line の座標を操作する際には、*x* 座標の移動量を同一とするものとする。なお、特に Block 要素と Line 要素については、極端な低音部、高音部に記号が配置される場合を考慮してマージンを有しているため、比較的自由度が高いといえる。

座標値の操作による情報埋め込みとそれともなう MusicXML 表現の変化の例としてここでは連鉤の符尾の長さや強弱記号の *y* 座標操作の例をそれぞれ図 17 と図 18 に示す。図中のマークアップにおける ... は一部属性が省略されていることを示す。

なお文献 13) では、画像データとして表された楽譜について、本節で述べたような演奏者に認識されにくい記号の位置変更の例(同文献 3.3 節)とそれを用いた透かしについて述べている。しかしながら、そこであげられている例は画像データであるからこそ位置変更が感知されにくいものもあり、ただちに MusicXML

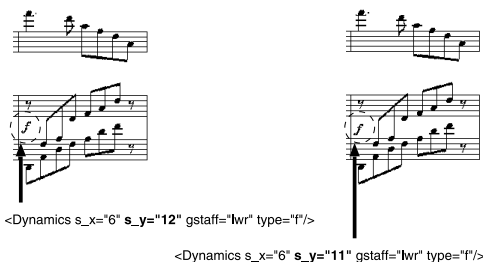


図 18 強弱記号の y 座標の変化とそれともなう MusicXML 表現の変化の例

Fig. 18 Variation of y -Coordinate of a Dynamic Mark and corresponding variation of MusicXML description.

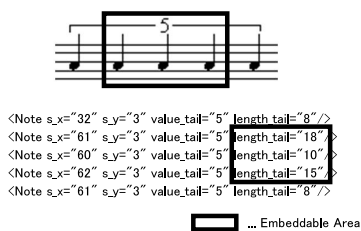


図 19 連符の y 座標における埋め込み可能位置の例

Fig. 19 An example of embeddable area for the scheme of coordinate manipulation in Tuplet.

に活用できるわけではない。

5.3 グループ ID 値の操作を利用した手法 (手法 3)

MusicXML には、3.3 節で述べたグループ ID という概念が存在する。このグループ ID 値は、MusicXML のパーザであるコンポーザやプラグインがグループを識別、分類するためにコンポーザが与える値であり、楽譜や旋律に影響を及ぼすものではない。すなわち、1) 隣り合う同じ種類のグループどうして同一の ID 値を使用しない、2) グループ内では同一の ID 値を使用する、という原則に従っていればこのグループ ID 値を変更しても楽譜、旋律ともに歪みを生じない。そこで、この特徴を冗長性としてとらえ、グループ ID 値をグループごとに変更することによって情報を埋め込む。

5.4 連符の y 座標を利用した手法 (手法 4)

連符記号の傾きは開始位置と終了位置の y 座標を決定する属性の値によってのみ決定され、間の記号に付加された同属性の値にはまったく依存しない。そこで、この特徴を利用し、連符記号の付加された記号のうち、連符の開始位置と終了位置にある記号を除くすべての記号における連符記号の y 座標を操作して情報を埋め込む。図 19 に埋め込み可能位置の例を示す。図中において、`value_tail` 属性値は何連符を表すかを決定し、`length_tail` 属性値は音符の始点 (符頭)

から連符記号までの距離を音階の数で表したものである。なお、図における `length_tail` 属性値は情報埋め込み前のものであり、操作を加えていない状態でもこのように不自然な値になりうることを示している。

6. Web を利用した MusicXML への電子透かしシステムの開発

5 章に述べた 4 つの埋め込み手法を実装し、電子透かしに関する知識や技術を持たない一般の利用者にも扱えるよう、Web を利用した電子透かしシステムを提案し、実際にプロトタイプシステムの開発を行った。

6.1 システムの仕様および利用環境に関する仮定

本論文で提案・開発したシステムでは以下の仮定をおく。

(1) 情報を埋め込む前のオリジナルコンテンツをコンテンツ制作者以外の者が入手することはないものとする。埋め込みシステムにおいても透かし埋め込み処理終了後、埋め込み前のオリジナルコンテンツを破棄するものとする。

(2) 不正利用を発見するためのコンテンツの追跡に関しては対象外とする。

(3) 同一曲と目されるコンテンツの埋め込み済みコンテンツや埋め込み前のオリジナルコンテンツなしで透かしデータを抽出することができる。

一般に電子透かしシステムでは、(3) を実現する目的で以下の方法がしばしば用いられる (文献 15) の 3.1 節の Type-II 埋め込み、図 20)。

(1) すべてのコンテンツ (原コンテンツ、透かし入りコンテンツ、改ざんを受けた透かし入りコンテンツ) はコンテンツがとりうる値の空間上の点と考える。

(2) 透かし入りコンテンツは、その点集合の部分集合 S' の元 (図 20 中では、'○' で表された点) しかとりえないものとする。

(3) S' の集合分割を考える¹⁶⁾。図 20 中では、'○' で表された点の集合は、それぞれ '00', '01', '10', '11' というラベルのついた 4 つの集合に分割される。

(4) 埋め込まれる透かし情報の集合から、集合分割への写像を考える。図 20 中では、たとえば透かし情報 00 に対応する '00' とラベルのついた 4 つの点に対応する。

(5) 実際の埋め込み後コンテンツは、埋め込まれる透かし情報で指定された集合の点から原コンテンツから見た歪みが最も小さくなる点を選ばれる。

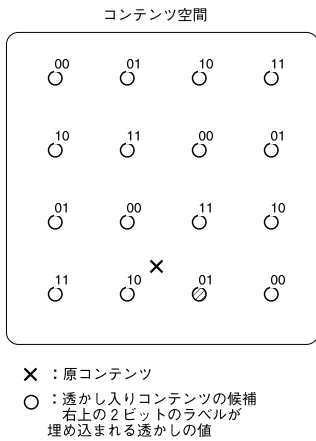


図 20 透かし情報と埋め込まれたコンテンツの対応モデル
 Fig. 20 A model of information embedding for contents.

図 20 は、透かし情報が 2 ビットのときの空間分割のモデルである。いま、図中の ‘×’ 印で示された原コンテンツに ‘01’ という 2 ビットの情報が埋め込まれるとき、斜線の入った ‘O’ 印のコンテンツが透かし入りコンテンツとして選択される。

本システムでもこの方式を採用し、さらにそのときの空間分割法や写像等のパラメータを秘匿することにより、透かしの頑健性の向上を図る。次節の埋め込み/抽出の方法で述べるとおり、電子透かしに関する知識や技術を持たない一般の利用者を想定して、透かしの埋め込み/抽出は中立な第三者が運営する特定のウェブサーバにおいて行われる方式を想定しているため、埋め込みと抽出で同じ写像が使われることが保証されており、そのため写像を公開する必要がない。

写像は基本的に秘匿とするが、しばしば見られる著作権侵害形態であるクリッピング（楽譜の不正な部分窃用）を考慮して、埋め込み時には入力された情報を繰り返し符号化し埋め込むものとし、抽出時には多数決復号を行うものとする。

このシステムを利用することで、コンテンツ制作者は MusicXML への電子透かし埋め込みと、電子透かしの埋め込まれた MusicXML からの情報抽出を自由に行うことが可能となる。その結果、コンテンツ制作者側で自由に著作権保護に関する対策を行うことができる。

6.1.1 システムを利用した電子透かしの埋め込み (図 21)

- (1) コンテンツ制作者はまず、MusicXML を生成することのできるコンポーザを用いて作曲を行い、MusicXML ファイルを生成する。
- (2) (1) で得られた MusicXML ファイルと自身の

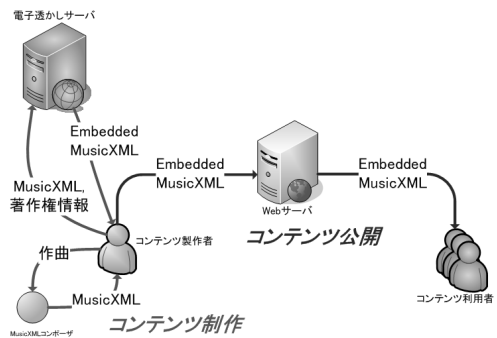


図 21 電子透かしシステム利用想定図
 Fig. 21 Model of the proposed watermarking system.

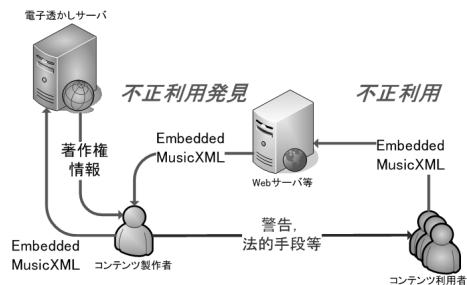


図 22 電子透かしシステム利用想定図 (不正利用発覚時)
 Fig. 22 Model of the proposed watermarking system when abuse is detected.

著作権情報を、本システムの稼働しているサーバに対して送信する。その結果、送信した著作権情報の埋め込まれた MusicXML ファイルを得る。

- (3) コンテンツ制作者は (2) で得られた著作権情報の埋め込まれた MusicXML ファイルを、任意の Web サーバにアップロードする等して公開する。
- (4) コンテンツ利用者は、(3) で公開された MusicXML ファイルをダウンロードし、利用することができる。

6.1.2 システムを利用した電子透かしの抽出 (図 22)

- (1) コンテンツ利用者が、著作権情報の埋め込まれた MusicXML ファイルを著作者の許可なしに第三者の閲覧可能な Web サーバにアップロードし、著作権法における公衆送信権の侵害を行った。
- (2) コンテンツ制作者が疑わしいファイルがアップロードされているのを発見した際には、電子透かしサーバに対して疑わしいファイルを送信し、そのファイルに著作権情報が埋め込まれていな

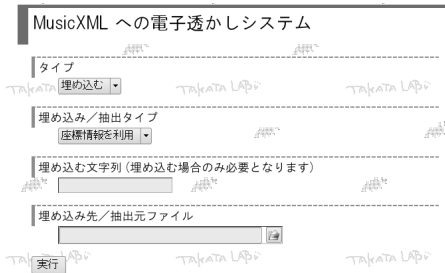


図 23 Web インタフェース

Fig. 23 Snapshot of Web interface of the watermarking system.

いかどうかを確認する。

- (3) (2)を行った結果、埋め込まれていた著作権情報としてコンテンツ制作者のものが返ってきた。
- (4) コンテンツ制作者は、この情報をもとにコンテンツ利用者に対し警告を発したり、場合によっては法的手段により訴えたりすることができる。

なお、本システムでは正規の一般利用者が流通しているコンテンツの著作権情報を確認する目的で透かし抽出機能を用いることを想定し、本機能を埋め込みを行った著作権者以外も利用可能としている。

6.2 システム構成要素

このシステムでは、MusicXML への情報埋め込みや、情報の埋め込まれた MusicXML ファイルからの情報の抽出を行えるような簡潔なインタフェースを利用者へ提供する。図 23 に、システムのインタフェースを示す。

現在、Web インタフェースの可用性を考慮し、埋め込むことのできる情報は文字列データに限定しているが、インタフェース部以外はバイナリデータを取り扱うことが可能である。また、今回の実装では圧縮処理は行わないため、埋め込み可能容量は埋め込む情報の性質に依存せず一定である。

なお、開発・動作確認に使用した環境は以下のとおりである。

OS Gentoo Linux 2004.2

Java Virtual Machine Sun J2SDK 1.4.2_05

サーバソフトウェア Apache Jakarta Tomcat 5.0.27

本システムは以下にあげる 5 つの構成要素からなり、それぞれの構成要素は図 24 のように構成される。

埋め込みモジュール群 図 24 における Embed Duplicate Tags Module, Embed Coordinates Module, Embed Group ID Module, Embed Tails Module の 4 つである。これは 5 章に述べた 4 つの埋め込み手法 (手法 1-4) をそれぞれ

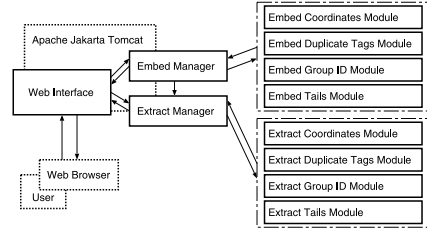


図 24 電子透かしシステムアーキテクチャ

Fig. 24 Architecture of the watermarking system.

モジュールとして実装したものである。本実装では、手法 1 では、記号の一重重複まで行い記号 1 つにつき、手法 2 では操作可能箇所 1 カ所につき、いずれも 1 ビット埋め込むものとする。手法 3, 4 では埋め込み可能箇所 1 カ所につき、それぞれ 8, 16 ビット埋め込むものとする。各モジュールは、埋め込みマネージャから埋め込み先ファイル名と出力先ファイル名を受け取り、実際に埋め込みを行うクラスのインスタンスを生成する。埋め込みたい情報をメッセージとし、電子透かしを埋め込むためのインスタンスメソッドを呼び出すことで、入力として与えられた MusicXML ファイルへ情報を埋め込み、出力する。

埋め込みマネージャ 図 24 における Embed Manager である。Web インタフェースから受け取った埋め込み先ファイル、埋め込む情報、埋め込みに利用する手法の ID と、出力先ファイルとして自身が生成した一時ファイルのファイル名を、適切な埋め込みモジュールへ送信する。埋め込みが成功した場合、この一時ファイルへの入力ストリームを生成し Web インタフェースへ返す。埋め込みが失敗した場合はその旨を利用者に通知するよう Web インタフェースへ指示し、処理を終了する。

抽出モジュール群 図 24 における Extract Coordinates Module, Extract Duplicate Tags Module, Extract Group ID Module, Extract Tails Module である。情報を埋め込まれた MusicXML ファイルから、情報の抽出を行う。5 章に述べた 4 つの手法それぞれについて抽出モジュールが存在している。各モジュールは入力先ファイル名を受け取り、埋め込まれていた情報を抽出して返す。無効なファイル名が指定された場合等、抽出が正しく行えなかった場合は例外を送出し、抽出マネージャにその旨を通知する。

なお、本モジュール群には、入力先ファイル名を受け取り、ファイルに情報が埋め込まれているかどうかを返す静的メソッドが実装されている。この

メソッドは、埋め込みが行われる際に抽出マネージャを介し埋め込みマネージャによって呼び出され、入力先ファイルにすでに情報が埋め込まれているかどうかの検定が行われる。これは、すでに情報の埋め込まれている MusicXML ファイルに対し多重に情報を埋め込むことを防ぐためである。抽出マネージャ 図 24 における Extract Manager である。Web インタフェースから送られるデータを適切な抽出モジュールへ送信するための、モジュール選定作業を担う。抽出が成功した場合、ファイルに埋め込まれていた情報を抽出モジュールから受け取り、Web インタフェースへと返す。無効なファイルが指定されていた場合等、抽出が正しく行えなかった場合はその旨を利用者に通知するよう Web インタフェースに指示する。

Web インタフェース 図 24 における Web Interface である。MusicXML への情報埋め込みや、情報の埋め込まれた MusicXML ファイルからの情報の抽出を簡便かつ円滑に行えるような簡潔なインタフェースを利用者へ提供する。このインタフェースに Web ブラウザを用いてアクセスすると、図 23 のような画面が表示される。

この画面上の入力フォームから以下の情報を入力して送信することで、必要な処理を行わせることができる。

- 埋め込み・抽出のどちらを行うか
- 埋め込み・抽出を行う際に、5 章に述べた 4 手法のうちどの手法を利用するか
- 埋め込む場合は、どのような情報を埋め込むか
- 埋め込み先、あるいは抽出元のファイル

7. 電子透かし埋め込み手法の評価

本章では、5 章に述べた 4 つの埋め込み手法を、以下にあげる 4 つの観点から評価する。

- 埋め込み可能容量
- 対改竄耐性
- コンテンツへの影響
- 対クリッピング耐性

以下、それぞれについて詳説する。

7.1 埋め込み可能容量

一般的な楽譜を無作為に 14 選出し、どれだけの情報を埋め込むことが可能であるかを 5 章に述べた 4 つの埋め込み手法それぞれについて調査した。ただしここでは、6 章と同様、手法 (1)–(4) で埋め込み可能箇所 1 カ所につき、それぞれ 1, 1, 8, 16 ビット埋め

込むものとし、埋め込み可能容量は楽譜中の埋め込み可能箇所と 1 カ所あたりの埋め込み可能ビット数の積で求めている。

それぞれの調査結果を表 2 に示す。表中の楽譜の項の“クラシック 1”とはヴァイオリンとハーブからなる室内楽曲を表し、“クラシック 2”とは木管、金管からなる四重奏曲を表す。また、ファイルサイズの項は埋め込み前のオリジナルのファイルサイズを表す。

連符の y 座標を利用した手法 (手法 4) 以外の手法では、おおむね記号数やファイルサイズに比例した容量となっている。また、記号の重複を利用した手法 (手法 1) や座標値の微細な操作を利用した手法 (手法 2) と比べると、グループ ID の操作を利用した手法ではより多くの情報を埋め込むことが可能であることが分かる。これは、グループ ID を利用した手法では 1 グループについて埋め込むことのできる情報量が多いことや、グループを生成する連符等の記号は一般的な楽譜中で頻繁に使用されるものであるということに由来すると考えられる。また、連符の y 座標を利用した手法において、楽譜によって埋め込み可能容量に極端な差が生じることについては、連符記号の出現頻度によるものと考えられる。すなわち、連符記号の出現頻度は一般的には低いといえるが、楽譜によっては大量に使用される場合も存在し、そのような場合においては大量の情報の埋め込みが可能であるということを示している。なお、万国著作権条約¹⁷⁾ 第三条では著作者名、発行年、©記号を示せば著作権保護の条件を満たすものとされる。また、現在、コンテンツ ID フォーラムでは、コンテンツを一意に定める識別コードであるコンテンツ ID の仕様制定がなされており、その中では透かし埋め込みに用いられるコンテンツ ID は 64 ビット長¹⁸⁾ となっている。MPEG-2¹⁹⁾ の著作権保護表示のためのフィールドである `copyright_id extension` も 64 ビットである。連符記号の出現数が少ない楽譜においても、他の 3 つの手法を併用して情報を埋め込むことで、これらの情報を埋め込むための情報量を確保することが十分に可能であると考えられる。

また、埋め込みにともなうファイルサイズの増加であるが、手法 2–4 については、当該埋め込み対象の属性値 (座標値やグループ ID 等) の変化にともない、元のバイト数で表現できない値となったときのみ、増加が生じる (たとえば、図 17 において右図から左図に変化した場合がそれに相当する)。したがって、たかだか、(埋め込み可能箇所 × 2) バイトまでしか増加しない。手法 1 については、頑健性を考慮せず埋め込み可能位置にすべて一重重複で埋め込むとすれば、たかだか埋め込

表 2 埋め込み可能容量

Table 2 Capacity of the amount of embeddable information.

	楽譜	パート	ページ	ファイル サイズ	記号数	埋め込み可能容量 (bits)			
						手法 1	手法 2	手法 3	手法 4
1	ピアノ譜	2	10	136 KB	2,818	970	926	2,160	16
2	クラシック 1	2	8	180 KB	3,718	1,523	1,021	3,248	544
3	ピアノ譜	1	2	24 KB	900	169	99	392	0
4	ピアノ譜	1	2	28 KB	691	265	206	824	0
5	ピアノ譜	1	2	20 KB	490	189	105	416	0
6	ピアノ譜	1	4	56 KB	1,383	425	375	744	0
7	ピアノ譜	1	5	80 KB	1,288	444	377	2,008	2,000
8	ピアノ譜	1	3	28 KB	864	277	123	488	0
9	ピアノ譜	1	2	24 KB	744	193	118	528	0
10	ピアノ譜	1	2	16 KB	430	163	73	288	0
11	ピアノ譜	1	3	28 KB	735	280	201	808	0
12	ピアノ譜	1	4	60 KB	1,130	418	319	1,280	960
13	クラシック 2	4	10	184 KB	4,244	1,416	1,324	2,600	0
14	ピアノ譜	1	2	28 KB	783	222	127	472	0

手法 1) 記号の重複を利用した手法

手法 2) 座標値の微細な操作を利用した手法

手法 3) グループ ID の操作を利用した手法

手法 4) 連符の y 座標を利用した手法

み前ファイルの重複可能タグのバイト数だけ増加する。したがって、たとえば、表 2 の楽譜 2 の場合、手法 1 で埋め込み可能容量一杯まで埋め込んだ場合、ファイルサイズは最大で 151 KB 増加し、ほぼ倍のサイズとなる。

7.2 対改竄耐性

本節では各手法における対改竄耐性の評価について述べる。

7.2.1 攻撃想定

図 1 にあげたように、MusicXML では 1 つの楽譜につき大量の XML 要素を生成する。また、(1) 3.2 節で述べたとおり記号の x 座標値が相対値であるため 1 つの記号操作が少なくとも五線全体に波及すること、(2) 各音符はしばしばグループ ID によって（しかも 1 つの音符は複数のグループに属することがある）、XML 上では必ずしも隣接していない多くの他の音符と関連付けられていること、という理由から、コンテンツに歪みを生じさせることなく、テキストエディタ等で XML を直接改竄することはかなり困難であると考えられる。したがって、3.4 節に述べたコンポーザを用いて改竄を行うものとする。そこで、以下に定義するコンポーザを用いた変換操作による攻撃を想定する。

- (1) コンポーザで読み込み、保存
- (2) コンポーザで読み込み、人為的に記号を別の場所へ移動し、再び戻して保存
- (3) コンポーザで読み込み、人為的に記号を新しく

追加し、その記号を削除したのち保存

- (4) コンポーザで読み込み、デフォルトの保存形式である .mpd 形式で保存し、再びそれを読み込み保存
- (5) コンポーザで読み込み、人為的に ID 値を有する記号グループのグループを解除し、再びグループ化し保存

なお、変換操作 (2)、(3)、(5) についてはコンポーザに備わっている機能によって改竄を行うだけでなく、人手による操作を加えた改竄を行っている。特に変換操作 (5) は、記号によっては譜面に著しい歪みを生じる操作である。

また、ほぼすべての記号が情報の埋め込み対象となることを考慮すれば、変換操作 (2) を人手ですべての記号に対して行うことが考えられるが、結局そのような操作は人手による写譜にかなり近い手間がかかるため、そのような攻撃は考慮の対象外とする。

7.2.2 個別手法の対改竄耐性

前節に述べた変換操作を行った MusicXML ファイルから情報の抽出を行ったところ、情報損失の評価は表 3 のとおりとなった。

詳細は以下のとおりである。

記号の重複を利用した手法（手法 1）変換操作 (2) を行うことで、当該埋め込み対象記号に埋め込まれた情報が全部損失する。これは、XML レベルでの記号の重複についてはコンポーザが関知す

表 3 情報損失の評価
Table 3 Summary of information loss.

	手法 1	手法 2	手法 3	手法 4
変換操作 (1)	○	○	○	○
変換操作 (2)	△	△	○	○
変換操作 (3)	○	○	○	○
変換操作 (4)	○	○	○	○
変換操作 (5)	○	△	△	△

○ ...情報の損失なし

△ ...特定条件下で微量の情報損失

ることはないが、コンポーザ上での操作により記号を完全に重ねると、それらの記号は 1 つの記号として扱われるというコンポーザの仕様に起因している。

座標値の微細な操作を利用した手法 (手法 2) 変換操作 (2), (5) を行うことで、当該埋め込み対象に埋め込まれた情報が全部損失する。座標値は DOM ツリーにおける葉となる記号の y 座標以外はピクセル単位で指定されており、人手によって記号の座標位置を完全に戻すことは難しい。これにより、変換操作 (2) を行うことで情報の損失が起こる。また、連鎖グループを解除すると、グループ内の音符の符尾の長さは既定値に戻る。そのため、連鎖の符尾の長さを操作して情報を埋め込んでいた場合においては、変換操作 (5) を行うことで情報の損失が起こる。

グループ ID 値の操作を利用した手法 (手法 3) 変換操作 (5) を行うことで、埋め込み対象グループに埋め込まれた情報が全部損失する。これは、コンポーザ上でグループを解除するとグループ ID が初期化され、再度グループ化しても別の ID 値が割り振られるというコンポーザの仕様に起因している。

連符の y 座標を利用した手法 (手法 4) 変換操作 (5) を行うことで、埋め込み対象の連符に埋め込まれた情報が全部損失する。連符グループを解除すると、対象となる記号から、連符記号の y 座標という概念、すなわち、図 19 における `length_tail` 属性自体が消失する。再度連符記号を設定しても、`length_tail` 属性値には規定値が設定されるため、情報の損失が起こる。

さて、個々の改竄が埋め込まれた情報にどの程度影響を与えるかについて議論する。

まず、埋め込む情報がコンテンツ ID フォーラム等と同様、64 ビットの場合を考える。改竄に対する情報の保護に標準的な誤り訂正符号である 2 元 BCH 符号を用いるとすると、文献 20) の表 9.2 より、埋め込

み可能容量が 127, 255 ビット以上あれば、それぞれ、10, 29 ビットまで誤り訂正可能であることが分かる。したがって、たとえば表 2 の楽譜 2 において手法 2 を用いて 64 ビットの情報を埋め込む場合を仮定すると、 $1021 > 127 \times 8$ であることから、楽譜の各ページごとに 64 ビットの情報を埋め込むことができ、それらは 10 ビットまでの誤り訂正が可能となる。埋め込み可能容量が 127 ビット未満の場合は短縮符号を用いる必要があるが、たとえば、表 2 の楽譜 5 において、手法 2 を用いて 64 ビットの情報を埋め込んだ場合、(127, 92) BCH 符号を 22 ビット短縮して用いることにより 5 ビットまでの誤り訂正が可能となる。

埋め込むべき情報が“(C) 2004 Atzm Watanabe”のようなバイト単位であり、改竄が埋め込んだ情報に与える影響も手法 3 のようにバイト単位である場合、有限体 $GF(2^8)$ 上の Reed-Solomon 符号 (たとえば文献 21), 7.3 節) の適用が考えられる。たとえば表 2 の楽譜 1 (ページ数は 10) の場合、手法 3 の埋め込み可能容量が 270 バイトであることから、仮に 2 ページごとに“(C) 2004 Atzm Watanabe”という 22 バイトの情報を埋め込むとすると、 $GF(2^8)$ 上の (255, 223) Reed-Solomon 符号を 201 バイト短縮した (54, 22) 符号を用いれば、15 バイトまでの誤り訂正が可能となる。

これらの誤り訂正可能な範囲内であれば埋め込まれた情報を改竄から保護することが可能となる。

なお、本研究では透かし攻撃ツールのようなものは考慮の対象外としているが、将来そのようなツールが出現しないと限らない。特に手法 1, 3, 4 では、人手では相対座標から絶対座標を求めつつ、同一グループに所属する記号群を把握したまま埋め込み情報の改竄を行うことは困難であっても、何らかの攻撃ツールが開発される可能性はある。しかしながら、その場合においても、現在他の様々なコンテンツで問題となっているカジュアルコピーに対する抑止にはなるものと考えられる。

手法 2 についても攻撃ツールが開発される可能性はある。しかしながら、オリジナルのコンテンツが分からない状態で、どこにどのように情報が埋め込まれているかを判別できないまま、手法 2 に対する機械的な改竄攻撃を行った場合、各記号の配置について、もとの透かしによる微細な歪みがかなりの確率で拡大されるため、楽譜全体として体裁に与える影響がかなり大きくなることが予想される。

7.3 コンテンツ品質への影響

情報の埋め込みを行うことで生じるコンテンツの歪みを表 4 に示す。

表 4 コンテンツの歪み
Table 4 Distortion to the contents.

	手法 1	手法 2	手法 3	手法 4
譜面における歪み	○	△	○	○
旋律における歪み	○	○	○	○

○ ...変化なし

△ ...埋め込み前と比較することで判別可能な程度

座標値の微細な操作を利用した手法以外では譜面、旋律ともに歪みは生じなかった。座標値の微細な操作を利用した手法でも旋律への歪みは生じず、また、譜面への歪みについても、実質的に問題とならないことが予想される。

このことを主観的評価実験により、確認した。

評価実験では 12 人の被験者に楽譜を視察してもらう方法で行った。各被験者は平素、楽器や楽譜に慣熟している学生であり、被験者の楽器経験の最小、平均、最大年数はそれぞれ、4, 8.8, 14 年である。7.1 節で調査対象とした楽譜の 2, 13 番目について、タイトルが表示されていたり、ページの半分ほどしか譜面のない最初と最後のページを取り除いた 15 ページについて、埋め込み前の楽譜ページと“(C) 2004 Atzm Watanabe”という文字列を可能な限り繰り返し埋め込んだ楽譜ページを用意する。それら 30 枚の楽譜ページを、(1) 埋め込み前後のページが両方入らないように、(2) 埋め込み前後のページ数になるべく均等に入るようにランダムに、2 つの群 (各群 15 ページ) に分ける。被験者もランダムに 6 人ずつの 2 つのグループに分け、1 つのグループに 1 つの楽譜ページの群を対応させる。被験者に 1 つの群の 15 枚の楽譜について、各楽譜 30 秒程度観察してもらい、5 段階評価 (1—楽譜として不自然さがめだつ、5—楽譜として自然である) を行ってもらった。

各楽譜ページの評価値の平均を図 25 に示す。

各楽譜ページの埋め込み前後の評価値平均の差について t 分布による両側検定を行った。その結果、 t の実現値は最大でも 1.43 (楽譜番号 12) であり、 $t(0.05, 10) = 2.23$ 以下であることから、埋め込み前後の楽譜の自然さに関する主観評価の平均値に有意な差がないことが確認された。

7.4 対クリッピング耐性

各手法について、クリッピングに対する耐性を有するかどうかについて評価を行った。

クリッピングにはコンポーザのコピーアンドペースト機能を用いる。また、楽譜において 8 小節以下のクリッピングについては慣用的に引用扱いとされるため、8 小節以下のクリッピングは正当な引用と仮定し、

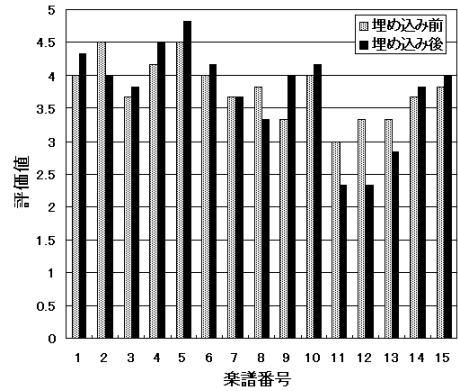


図 25 楽譜の歪みに関する評価実験結果

Fig. 25 The result of the experiments on subjective evaluation for the distortion to musical scores.

9 小節のクリッピングに耐えうるかどうかで対クリッピング耐性の評価を行う。

4 手法とも、一般的な楽譜において 9 小節に埋め込むことのできる情報は極端に少ない。たとえば 1 小節に全音符が 1 つだけ配置されているような簡単な部分では、残存する情報が 0~1 ビット程度であることもあり、クリッピングに耐えうるものではないと判断した。しかしながら、コンポーザのコピーアンドペースト機能を用いてクリッピングを行うと、楽譜に対し著しい歪みを生じることが分かっている。そのため、譜面や旋律ともに著しい変化を生じる。このことにより、クリッピングを行うこと自体が困難であるということが分かった。

8. おわりに

本論文では、まず、我々が独自に分析した MusicXML のフォーマットとその特徴について述べ、この特徴を利用した電子透かしの埋め込み手法について述べた。次にこれらの手法を実装した、Web を利用した電子透かしシステムの開発について述べた。最後に本システムを用いて各電子透かし埋め込み手法についての詳細な評価を行い、クリッピングへの耐性こそ持たないものの、コンポーザを用いた整形による攻撃に対しての耐性を有することを示した。これらの手法では、XML における一般的な冗長部分¹²⁾ について埋め込みを行っているわけではなく、MusicXML の特徴に基づく冗長部分に対して埋め込みを行っている。そのため、一般的に XML に存在するといわれる冗長部分への改竄に対しても耐性を有する。

また、このシステムを利用することで、一般のユーザが自由に電子透かしの埋め込みが可能となる。これより、不正利用への心理的抑止力となることが期

待でき、これは楽曲の複製品の流出等の著作権を侵害する不正行為に対する対抗策となることが期待できる。

一般にテキストのような、いわゆる representative contents は音声、画像データのような perceptual contents とは異なり、情報隠蔽は困難であるとされている。MusicXML も本質的に、この困難性を免れたわけではないため、悪意ある利用者が本手法を悪用し、電子透かしを除去するツールを開発する等して電子透かしの除去を行う場合について考察する。

提案手法 1, 3, 4 においてはコンテンツに歪みを生じることなく電子透かしを部分的に攪乱することが可能となりうるが、その場合でも、本提案手法は、いわゆるカジュアルコピー対策として十分有用であると考えられる。一方、提案手法 2 においては、人手で行う限りは写譜とかなり近い手間をかけない限り、歪みを生じることなく電子透かしを除去することはかなり困難であり、それゆえ単なるカジュアルコピー対策以上の効果を有するものと考えられる。そこで除去ツール対策として、ある程度の頑健さが見込まれる提案手法 2 を用いる場合を考えると、楽譜やページに埋め込むことができる情報が楽譜/ページごとに一定である以上、クリッピング耐性と改竄耐性はトレードオフの関係となる。本来このトレードオフの折り返しは各著作権者が判断するものであるが、今回の 6 章のシステム実装では、ユーザの可用性を考慮した場合パラメータ入力を極力減らす方が好ましいため、細かな入力を省く方針とし、結果として繰り返し符号化を実装している。しかしながら 7.1 節、7.2 節の結果より、1, 2 ページ単位のクリッピングならば、7.2.2 項で述べたような強力な誤り訂正符号を導入することにより、ある程度の改竄耐性を持たせることが十分可能であると予想される。一般のバイナリデータの埋め込みを含めた、このあたりのユーザインタフェース設計は今後の検討課題としたい。

また、今後評価実験を重ね、提案手法の評価をより確実なものとするのも今後の検討課題としたい。

謝辞 本研究を始めるにあたり、いろいろと励ましいただいた(有)ミュージカル・プラン社の江守幸一氏ならびに水町嘉秀氏に感謝いたします。本論文の査読者の方々には、論文を精読していただき、有益なご指摘をいただきました。皆様に深く感謝いたします。

参 考 文 献

- 1) 江守幸一, 黒田大輔, 中塚秀子: XML を用いた楽譜表示演奏システムの開発, *IPA Technology EXPO ITX 2001 — Part 1* (2001).
- 2) Recordare L.L.C.: MusicXML Definition Version 1.1 (2006).
<http://www.recordare.com/xml.html>
- 3) (有)ミュージカル・プラン: MusicX (2003).
<http://xml.musicalplan.com/>
- 4) The XEMO&trade Project Home Page.
<http://www.xemo.org/>
- 5) 深見友紀子: 日本の音楽教員のコンピュータ・リテラシーに関する調査結果とその考察, *KCME シンポジウム* (2004).
- 6) 松本隆明, 岡本龍明: 情報セキュリティ技術, p.193, (株) オーム社 (2000).
- 7) World Wide Web Consortium: Extensible Markup Language (XML) 1.0 (3rd Edition) (2004). <http://www.w3.org/TR/2004/REC-xml-20040204/>
- 8) (有)ミュージカル・プラン: MUSIC PRO for Windows V4 (2004).
http://www.musicalplan.com/pro_mpv4.html
- 9) MIDI Manufacturers Association: The Complete MIDI 1.0 Detailed Specification, document version 96.1 2nd edition (2001).
- 10) 松本 勉, 井上大介, 北林創太: 演奏データファイル SMF への情報ハイディング方式, 2000 年暗号と情報セキュリティシンポジウム, C03 (2000).
- 11) 岩切宗利, 関根健一郎, 山本紘太郎, 松井甲子雄: 楽音符号への電子透かしに関する一提案, 2001 年暗号と情報セキュリティシンポジウム, pp.915-920 (2001).
- 12) 井上信吾, 村瀬一郎, 滝澤 修, 松本 勉, 中川裕志: XML におけるステガノグラフィ手法の提案, 2002 年暗号と情報セキュリティシンポジウム, pp.301-306 (2002).
- 13) Schmucker, M., Busch, C. and Pant, A.: Digital Watermarking for the Protection of Music Scores, *Proc.SPIE: Security and Watermarking of Multimedia Contents*, Vol.4314 (2001).
- 14) 浅香 淳: 新音楽辞典, p.503, (株) 音楽之友社 (1977).
- 15) Wu, M. and Liu, B.: *Multimedia Data Hiding*, Springer, NY (2003).
- 16) Chen, B. and Wornell, G.W.: Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding, *IEEE Trans. Information Theory*, Vol.47, pp.1423-1443 (2001).
- 17) 日本ユネスコ国内委員会: 万国著作権条約 (1955). <http://www.mext.go.jp/unesco/horei/pdf/j003.pdf>
- 18) コンテンツ ID フォーラム: cIDf 仕様 2.0 版 (2003).
- 19) ISO/IEC 13818-1:2000, Information technology — Generic coding of moving pictures and associated audio information: Systems.

<http://www.iso.org/iso/en/ISOOnline.frontpage>

- 20) Peterson, W.W. and Weldon, Jr., E.J.: *Error-Correcting Codes, 2nd edition*, pp.274-275, The MIT Press, MA (1972).
- 21) Lin, S. and Costello, Jr., D.J.: *Error Control Coding, 2nd edition*, Prentice Hall, MA (2004).

(平成 18 年 5 月 8 日受付)

(平成 18 年 10 月 3 日採録)



渡邊 集

1981 年生。2004 年岩手県立大学ソフトウェア情報学部ソフトウェア情報学科卒業。2006 年同大学大学院ソフトウェア情報学研究科修了。同年株式会社インターネットイニシアティブ入社。情報隠蔽とモバイルエージェントに関する研究に従事。



加藤 貴司

1971 年生。2001 年東北大学大学院情報科学研究科博士後期課程修了。現在、岩手県立大学ソフトウェア情報学研究科講師。博士(情報科学)。マルチエージェントシステムにおけるエージェントの協調に関する研究に従事。人工知能学会、電子情報通信学会各会員。



ベッド バハドゥール ビスタ(正会員)

1967 年生。1991 年 York 大学電子工学科卒業。1997 年東北大学大学院情報科学研究科博士課程修了。1997~1998 年宮城大学勤務。現在、岩手県立大学ソフトウェア情報学研究科助教授。博士(情報科学)。プロトコルの仕様記述と合成、モバイル通信に関する研究に従事。



高田 豊雄(正会員)

1962 年生。1989 年大阪大学大学院基礎工学研究科博士後期課程修了。現在、岩手県立大学ソフトウェア情報学研究科教授。工学博士。セキュリティと誤り制御通信に関する研究に従事。電子情報通信学会、情報理論とその応用学会、IEEE、ACM 各会員。