ヒューマンインタフェース 93−6
（２００１．５．１１）

# 字幕表示システム VCML Player の新機能について

鈴木 隆広, 渡邉 括行, 杉山 雅英
会津大学 コンピュータ理工学研究科

あらまし　本報告では、字幕表示システムである VCML Player の新機能について述べる。”Video Caption Markup Language(VCML)”とは、映像データに付加する字幕情報を制御するための言語として開発された。VCML で記述されたドキュメントは “VCML Player” によって処理される。このプレイヤーは映像と字幕をリアルタイムで合成し、再生する。操作が CUI のみだった従来のバージョン (0.2) のプレイヤーを改良し、本報告では GUI を実装し、また表示される字幕言語の切り替え機能を実装させ、機能面での改良を行った。一方表示タイミングの自動決定は字幕表示の上で非常に重要であり、VCML では字幕表示タイミング自動生成モジュールを使用している。本報告ではそのタイミング自動決定の手法として、Voice-Pause 法を提案した。これにより、長時間の音声を処理する際の時間とメモリを低減することができるので、長時間の音声を処理可能となる。実験によりタイミングのずれは平均 0.072 秒以下である事が判った。

# New Functions of VCML Player

**T. Suzuki, K. Watanabe, M. Sugiyama**

**Graduate School of Computer Science and Engineering**
**The University of Aizu**

**Abstract**　This paper describes the enhancement of “VCML Player” and its time alignment module. “Video Caption Markup Language (VCML)” is a language to control caption information on video data. Based on the VCML document “VCML Player” displays video with captions time by time. The caption display timing information is one of the most important parts of the VCML document and is generated by the time alignment module. Voice-Pause method is proposed for the alignment between text and speech. This method can be applied long duration voice data and requires less processing time and memory size.

## 1　まえがき

現在、字幕の作成には多大な時間とコストが必要となっており、一部の限られたテレビ番組が字幕付けを行っている。そのため、ニュース番組等の字幕放送については、NHK や TAO などが研究を進めている [1]。またコンピュータ上での字幕付き映像は、Real Player で動作可能な SMIL[2] があげられる。これはそれぞれに独立したメディアを同期させ表示することを可能にした。字幕の場合、表示する字幕をストリームテキストとして用意する。SMIL ではそれぞれのメディアを独立させたオブジェクトとして考え配置していく。そのため表示位置などの異なる字幕を表示させるためには別のストリームテキストを用意しなければならないなど、字幕主体で考えると不備な点がある。

Video Caption Markup Language(VCML)[3] では 1 つの字幕を 1 つのオブジェクトとして表示時間を与えて表示させることで、より字幕表示に適した記述を可能にした。VCML によって記述されたドキュメントは、VCML Player というシステムによって再生される。前バージョン “VCML Player 0.2” は再生のための基本機能を備えているのみで、ユーザーインターフェースなどは、未完成であった。そこで改良を加え、“VCML Player 1.0” とした [4]。

また字幕を付与するにあたって、音声のタイミングに合わせて字幕文章を表示するということは非常に重要であり、VCML では字幕表示タイミング自動生成モジュールを作成してきた。これは入力音声とテキストを音素単位で対応付け、字幕表示タイミン

グを決定する。

しかし、時間軸整合において約 1 時間のビデオデータに音素単位の対応付けを適用した場合、膨大なメモリ容量と処理時間が必要となる。そこで、パワーや Block Cepstrum Flux の音声特徴量によって音声区間を事前に切り出した後，ワードスポッティングによって字幕表示タイミングを決定する方法が提案され、良好な結果が報告されているが処理に実時間の 7〜8 倍を必要とする [1]。

そこで本報告では字幕の表示制御を行う VCML を再生するためのプレイヤー “VCML Player 1.0” の新機能と、事前に音声・非音声区間の切り出しを行わず、無音情報及び朗読単位の継続時間情報を用いて長時間音声の朗読単位への区分化アルゴリズムを提案し、その評価について述べる。

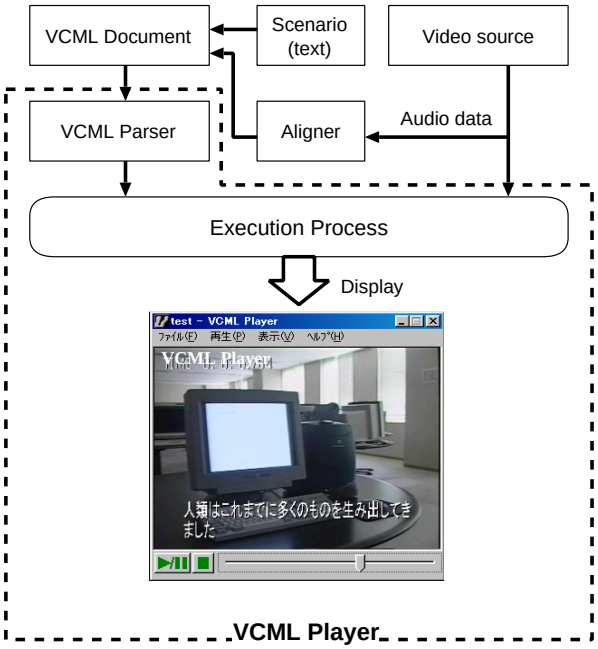


図 1: VC システムの処理の流れ

# 2 VC システム

## 2.1 VC システムの処理の流れ

VCML Player で字幕映像を再生するには、図 1 で示すようにいくつかのプロセスが必要となる。ビデオソースのうち、オーディオデータをタイミング自動生成モジュール (Aligner) を使って処理し、字幕表示タイミングを決定する。この決定したタイミング情報をシナリオデータ (テキスト) に加えて、VCML ドキュメントとする。作成されたドキュメントは VCML パーサーを通して解析し、プレイヤーの実行部分が再生処理を行って表示されることになる。

## 2.2 VCML の言語仕様

VCML の設計には、今後の拡張や発展性を考え XML[5] を利用した。図 2 に示すような VCML ドキュメントを用いて、製作者やユーザーは映像に直接文字等を書き込むことなく、また特別な編集ソフトを使用することなく、容易に字幕付き映像を作成することができる。

VCML では、表 1 にしめすようなタグを利用して様々な機能を実現させている。たとえば、文字のサイズや色、文字につける影の色などを指定できる。通常は表示位置を top,bottom などから選ぶが、座標を指定することでその位置に字幕を表示させることができる。また字幕の横及び縦表示、表示している文字列のスクロール機能などがあり、ユーザーはエフェクトを付けたり、思い通りの位置に字幕を表示させることができる。現在表示しているものとは別の映像を画面に重ねることもでき、そうして表示させた映像に別の字幕を付けることも可能である。

表 1: VCML の主要なタグとその機能

| Tag | Function |
|---|---|
| `<head>` | |
| `<title>` | define document title |
| `<author>` | define document author |
| `<copyright>` | define document copyright |
| `<date>` | define described date |
| `<body>` | |
| `<layout>` | specify the display layout |
| `<time>` | specify the display timing |
| `<font>` | specify the character style |
| `<caption>` | display the caption (region name) |
| `<sub>` | display the caption (x-y) |
| `<ruby>` | display rubies |
| `<img>` | display the image |
| `<video>` | play the specified video source |

## 2.3 プレイヤー

VCML Player の開発は、Windows98 上で行った。開発言語は”Visual C++ 6.0” を、XML Parser には

```
<?xml version="1.0"?>
<vcml>
  <head>
    <title>UNTITLED</title>
    <author>T.Suzuki</author>
    <date>May.11, 2001</date>
    <copyright>T.Suzuki</copyright>
    <rootlayout width="640" height="480"/>
  </head>
  <body>
   <time begin="0.0" end="10.0">
     <video src="aizu.avi"/>
   </time>
   <time begin="0" end="10">
     <caption language="Japanese">
        (日本語) 人類はこれまで多くのものを
                 生み出してきました
     </caption>
     <caption language="English">
        (English)Human Being produced
                  various items
     </caption>
   </time>
  </body>
</vcml>
```

図 2: VCML ドキュメントの記述例

IBM の”XML for C++”を使用した。

VCML Player 1.0 では、GUI から各種操作することができるようにした。図 3 のプレイヤーの例が示すように、ユーザーは映像の表示部分下にあるボタンを用いて再生、一時停止、停止の操作が可能で、その隣のスライダーを使用して必要なシーンへ移動させることができる。またその他にも、メニューから VCML ファイルのヘッダ情報をダイアログに表示させることや、字幕の表示/非表示を切り替えることが可能である。

## 2.4 字幕の表示言語の切替え

映像に対して字幕を表示する目的のひとつとして、視聴者が理解できない言語での対話や、音声に対し、母国語の字幕を表示することでその理解を助けるというものがある。海外の映画やニュース番組などに字幕を表示するのがこの一例である。

また言語学習の目的で外国語の映像を見る場合、そこに表示される字幕に同じ国語の文が記されていたほうが都合が良い場合がある。こういった場合、ユーザーが表示される字幕の言語を必要に応じて手動で

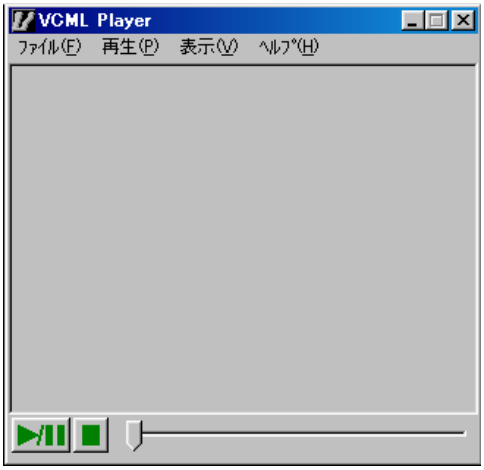


図 3: VCML Player の GUI

切り替えることができれば、字幕表示システムとしてその利便性が増すことになる。

VCML Player では、新たに言語切り替え機能を実装した。これは、VCML 文書中で指定された表示言語を切りかえる機能である。製作者があらかじめ字幕情報に対して言語情報を与えておけば、ユーザーはその範囲で自由に表示する言語を切り替えることができる。言語は、<caption>、<sub>の *language* というアトリビュートで指定している。

図 4 では、言語切り替えを行うための VCML ドキュメント記述例を示す。この例では前半 5 秒、後半 5 秒の合計 10 秒間の字幕が流れる。前半で流れる字幕は Japanese, English の二種類、後半では French, English の二種類になる。ユーザーがプレイヤーでの字幕言語として English を選択していた場合、両方とも字幕を表示させる。Japanese を選択していた場合は前半のみ指定された字幕を流し、後半では指定が無いため空白となる。French の場合はその逆となる。

```
<time begin="0" end="5">
    <caption language="Japanese">..</caption>
    <caption language="English">...</caption>
</time>
<time begin="5" end="10">
    <caption language="French">...</caption>
    <caption language="English">..</caption>
</time>
```

図 4: VCML における言語切り替えの記述例

## 3　表示タイミングの決定法

これまでの手法は音声とテキストを音素単位で対応付けを行い、字幕表示タイミングを決定し、テキスト (文章) を表示単位として表示してきた。ビデオデータ中の音声データはフレーム毎に特徴抽出して特徴ベクトル系列へ変換し、一方でテキストの漢字/かなも音素特徴ベクトル系列に変換する。これら二つの特徴ベクトル系列は DP マッチングによって時間軸整合が行われる。詳細な処理の流れは図 5 のようになる [6, 7]。

しかし一般的に音声が長くなる程、テキスト中の音素総数は増加し膨大なメモリ容量が必要となる。必要なメモリ量は入力テキスト中の音素数とフレーム数に依存し、音素数 × フレーム数で表される。

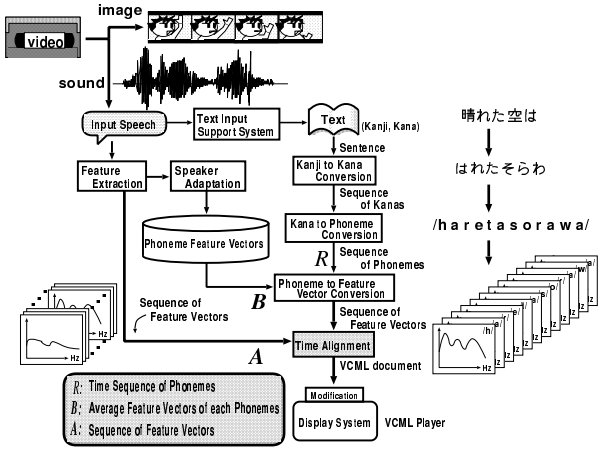


図 5: Aligner の処理の流れ

そこで、話者は音声発話とポーズの 2 つの状態を交互に遷移すると仮定し、DP マッチングによって入力音声と状態系列の対応を取り長時間音声を朗読単位に区分化し、字幕表示タイミングとする。これを、Voice-Pause 法と呼ぶことにする。図 6 はその概念図になる。

従来の方法が音素を整合の単位としていたのに対し、Voice-Pause 法では朗読単位を整合単位としている。必要なメモリ量はテキスト中の朗読単位数とフレーム数によって表され、朗読単位数 × フレーム数となる。

これは、例として 2000 年会津大学入学案内ビデオ中の学長講話約 70 秒の音声 [6] を考えると、従来法では約 14MBytes のメモリ量が必要だったのに対し、Voice-Pause 法では約 0.08MBytes のメモリ量で済む計算になる。

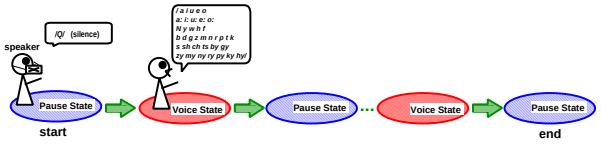


図 6: Voice-Pause 法の概念図

### 3.1　Voice-Pause 法

Voice-Pause 法では音声を朗読単位で対応付け、テキストを表示単位として表示する。朗読単位を検出の単位にするのは、音声データから事前にポーズを検出しそれを用いて区分化すると、ポーズの長短や有無により、正しく処理できないことが考えられるためである。ポーズが入る箇所は息継ぎや意味の切れ目であると考えられるので、テキストから 3.3 のルールに従って推定する。

また、朗読状態の特徴付けは以下のように行う。音声に対応する特徴ベクトルはポーズ状態で無音の特徴ベクトル、音声状態では音声を表す音素特徴ベクトルに近いと考えられる。従って横軸に入力音声のフレーム番号 $i$、縦軸に状態番号 $j$ を取ると、格子点 $(i, j)$ における音響的な距離 $d_{i,j}$ (local distance) は、入力フレームの特徴ベクトル $\boldsymbol{a}_i$ と状態 $j$ において選択された音素特徴ベクトルとの距離であり、以下のように定義される。

$$d_{i,j} = \begin{cases} \min\limits_{0 \le p \le N-2} d^2_{\mathrm{CEP}}(\boldsymbol{a}_i, \boldsymbol{b}_p) & (j\text{: Voice}) \\ d^2_{\mathrm{CEP}}(\boldsymbol{a}_i, \boldsymbol{b}_{N-1}) & (j\text{: Pause}) \end{cases}$$

ここで $\boldsymbol{b}_n (0 \le n \le N-2)$ は予め作成された日本語音素の特徴ベクトル、$\boldsymbol{b}_{N-1}$ は音声毎に学習される無音 (ポーズ) の特徴ベクトルである。本稿では音素数を 38 としているので、$N = 39$ とする。$d_{\mathrm{CEP}}$ は入力フレーム $\boldsymbol{a}$ と $\boldsymbol{b}$ の LPC ケプストラム距離であり，$w_p$ はパワー項への重みである。

$$d^2_{\mathrm{CEP}}(\boldsymbol{a}, \boldsymbol{b}) = w_p \, (\log_{10} \frac{a_0}{b_0})^2 + \sum_{k=1}^{K} (a_k - b_k)^2$$

図 7 に示す DP マッチングに用いるパス制御は音響的な差と音声およびポーズの各々の平均継続時間を

考慮したものであり、格子点 $(0,0)$ から $(i,j)$ までの累積距離の最小値 $g_{i,j}$ は以下の漸化式で与えられる。

$$g_{i,j} = \begin{cases} \min\limits_{0\le k\le \lceil w_l l_j \rceil} (D_{i,j,k} + w_L\ L_{j,k} + g_{i-k-1,j-1}) \\ +\infty \qquad\qquad (i,j<0) \end{cases}$$

ここで，$D_{i,j,k}$ は 格子点 $(i-k,j)$ から $(i,j)$ までの local distance の累積，$L_{j,k}$ は 状態 $j$ が $k+1$ フレーム継続する場合の推定フレーム長との差、$l_j$ は 状態 $j$ の推定フレーム長、$w_L(>0)$ は 状態 $j$ の継続時間長制御用変数をそれぞれ表し、以下で計算される。

$$\begin{aligned} D_{i,j,k} &= \sum_{n=0}^{k} d_{i-n,j} \\ L_{j,k} &= (l_j - (k+1))^2 \end{aligned}$$

本稿では $w_L = 1.5$ とする。これは全ての状態が推定継続時間の 1.5 倍の範囲内であることを意味する。$I$ を入力音声のフレーム数、$J$ を状態数として $g_{i,j}$ を格子点 $(0,0)$ から $(I-1,J-1)$ まで求めた後、バックトレースすることにより全体としてコストが最小となるパスを求める。これにより入力音声と状態系列が対応付けられ、長時間音声の区分化を行うことが可能となる。

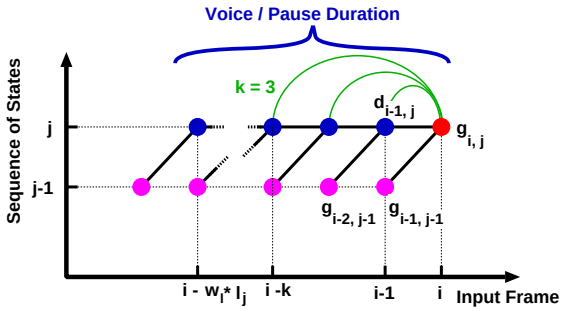


図 7: DP マッチングに用いるパス制御方法

## 3.2 無音特徴ベクトル作成方法

入力音声フレームの対数パワーを 256 レベルに量子化し、図 8 に示すパワー値のヒストグラムを作成する。判別分析法 [8] により入力フレームを無音と有音に分類する閾値を自動的に決定し、無音の平均値を求め，その前後 5 レベルの範囲のパワーを持つ特徴ベクトルを平均化し、無音特徴ベクトル $\boldsymbol{b}_{N-1}$ を作成する。4.1 に示すデータ Gotai-A を用いた場合、閾値は 105、無音の平均は 48 であった。

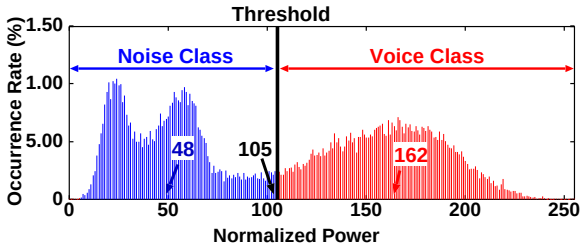


図 8: フレームのパワーとその出現頻度の関係

## 3.3 テキストの朗読単位への分割方法

朗読者は文の意味や聞き手の理解のしやすさを考慮して朗読単位を決定している。本稿では、漢字仮名まじり文で書かれた入力テキストに以下のルール 1，2，3 を順番に適用し朗読単位を決定する。

- ルール 1: ”。” は朗読単位の終了地点である。ただし、”「」” 内、”（ ）” 内の”。” は除く。
- ルール 2: ”「」” の前後にある改行は、共に朗読単位の終了地点である。
- ルール 3: 1 回以上の繰り返す記号の後にある改行は朗読単位の終了地点である。ただし、記号には全角の”！”，”。” が含まれる。

## 3.4 状態継続時間長の推定

音声状態の継続時間長は、図 9 に示すように朗読単位に含まれる漢字仮名まじりの文字数と継続時間から 1 次式 $y = ax + b$ による近似を用いる。表 2 の Gotai-A を使用した場合 $a = 0.145, b = -0.377$ となった。相関係数は 0.975 であり強い相関を示した。ポーズ状態の継続時間長は、目視でラベルを付与することで得られた朗読単位間ポーズの平均継続時間を用いた。

# 4 評価実験

## 4.1 実験データと評価方法

テキストとして「五体不満足」（乙武洋匡著，講談社）まえがきから第 I 部前半を使用した。入力音声は視覚障害者用朗読サービス用にカセットテープに収録された朗読音声であり、朗読者は熟練した朗読ボランティアの女性 1 名である。音声の前半 Gotai-A でパラメータ $w_L$、$w_p$ を学習し、後半 Gotai-B を評価に用いた。音素モデル学習に女性話者 4 名の ATR 音素バランス単語データの重複しない 54 単語（計 216

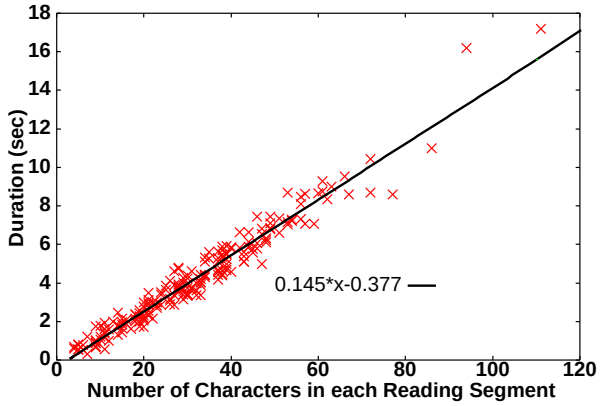


図 9: 朗読単位に含まれる日本語文字数と継続時間長の関係

単語）を用いた。この 4 名には朗読者は含まれていない。音声分析条件は次のとおりである。音声特徴量として LPC ケプスラム係数 (16 次) +パワー、標本化周波数 12 kHz、LPC 分析次数 (14 次)、分析窓長 21.3 msec（256 点)、フレーム周期 42.6 msec（512 点)、窓関数 ハミング窓、高域強調 $(1-0.97z^{-1})$ とした。朗読単位の開始・終了時刻のラベルを目視によって作成し、提案手法により得られた結果との差を評価した。

表 2: 五体不満足朗読データの仕様

| データ名 | Gotai-A | Gotai-B | 合計 |
|---|---|---|---|
| 朗読単位数 | 201 | 272 | 473 |
| データ長 | 18 分 15 秒 | 23 分 38 秒 | 41 分 53 秒 |
| 音声容量 | 26.2MB | 34.1MB | 60.3MB |
| 抽出後容量 | 1.7MB | 2.3MB | 4.0MB |

### 4.2 朗読単位の決定実験結果

実験結果を表 3 に示す。入力音声の大きさへの頑健さを調べる為に波形振幅の増幅率を 0.5 から 3.0 まで変化させた。パラメータの値として増幅率が 1.0、$w_L = 0.40$，$w_p = 50.0$ の場合、Gotai-B に対してラベルとの差は平均で 0.072 秒であり提案手法は良好に動作することが分かる。音声データに対する処理時間は PentiumIII-500MHz の CPU、384MB のメインメモリを搭載したマシンを用いた場合 207.93 秒となった。これは実時間の 0.146 倍であり高速な処理が可能であることを示している。また、一括して処理した場合（A+B）の処理時間は 623.92 秒であった。

表 3: 提案手法による朗読単位の決定結果とラベルの差

| | | | 平均誤差（秒） | | |
|---|---|---|---|---|---|
| 増幅率 | $w_L$ | $w_p$ | Gotai-A | Gotai-B | A+B |
| 0.5 | 4.00 | 110.0 | 0.123 | 0.093 | 0.108 |
| 1.0 | 0.40 | 50.0 | 0.094 | 0.072 | 0.080 |
| 2.0 | 0.03 | 2.0 | 0.092 | 0.066 | 0.076 |
| 3.0 | 0.04 | 4.0 | 0.082 | 0.068 | 0.072 |

## 5 むすび

VCML Player について、GUI の開発、字幕表示言語切り替えなどの機能を実装し、VCML Player 1.0 とした。一方字幕表示タイミング決定モジュールの改良を行った。Voice-Pause 法によって求められた字幕表示タイミングの誤差は平均で 0.072 秒、処理速度は実時間の 0.146 倍であり、正確で高速な処理が可能であることが示された。

今後は自動生成モジュールとの統合、ネットワーク上の映像の再生、またモジュールでは様々な発話スタイル、BGM や音声以外が重畳したデータに対する適用を検討する [10]。さらに笑い声などの声以外の音響情景音の検出の検討も行う [11]。

## 参考文献

[1] 白井, 他, 視聴覚障害者向け放送ソフト製作技術研究開発プロジェクトの研究状況, 視聴覚障害者のためのテレビ用字幕製作に関する国際ワークショップ論文集, pp.9-27, (1999-11).

[2] S. Bugai, D. Bulterman, Synchronized Multimedia Working Group, “Synchronized Multimedia Integration Language (SMIL) 1.0,” W3C (Jun. 15, 1998).

[3] 深田, 渡邉, 杉山, 字幕表示用言語 VCML の設計とその表示システムの開発, 情報処理学会ヒュー

マンインタフェース研究会, 2000-HI-87-7, pp.37-42, (2000-01).

[4] 鈴木, 杉山, 字幕表示用言語 VCML とプレイヤーの改良, 情報処理学会, 7Q-2, pp.73-74 (Mar. 2001).

[5] T. Bray, J. Paoli, C.M. Sperberg-McQueen, editors, “Extensible Markup Language (XML) 1.0 (Second Edition),” W3C (Oct. 6, 2000).

[6] 渡邉, 杉山, 字幕自動生成における字幕と音声の時間軸整合の検討, 信学技報, SP99-27, pp.7-14(June 1999).

[7] K. Watanabe, M. Sugiyama, “Automatic Caption Generation for Video Data, – Time Alignment between Caption and Acoustic Signal –”, Proc. of MMSP99, pp65-70 (Sep. 1999).

[8] 白井，谷内田，パターン情報処理，新コンピュータサイエンス講座, pp.35-37, pp.35-37, オーム社, 1998.

[9] 渡邉, 杉山, 長時間音声字幕化のための朗読単位への区分化, 音学講論, 3-P-2, pp.167-168 (Mar. 2001).

[10] 竹内, 山下, 内田, 杉山, 音楽と音声のセグメンテーションの最適化, 音学講論, 3-P-1, pp.164-166 (Mar. 2001).

[11] 金田, 杉山, 音響情景字幕表示のための笑い声の検出, 音学講論, 3-P-3, pp.169-170 (Mar. 2001).