Vol.2017-MUS-116 No.21
2017/8/26

TextTimeline: 文字表示を保持した発話テキストの音響特徴可視化

中野 倫靖[1,a)]　加藤 淳[1,b)]　後藤 真孝[1,c)]

概要：発話を伴う文字テキストにおいて、各文字の発声タイミングやその音響特徴量を把握できるように可視化する TextTimeline を提案し、複数の実現方法を比較する。文章や歌詞等のテキストにおいて、その各文字は固定的な幅を持つフォントで表現されるにもかかわらず、それらに対応する発話の時間長は多くの場合可変である。したがって音響特徴を可視化するために、従来、ピアノロール等のように音声の時間軸を固定（優先）して文字を分割表示する方法や、逆に、カラオケ歌詞や詩吟の吟詠譜等のように文字位置を固定して各文字の音響特徴を表現する方法があった。しかし前者は、文字位置が音声に制約を受け、文字間の重なりや空白によってテキストの大局的な関係を把握しにくく、後者は、音響特徴を音声の時間軸通りに可視化できない問題があった。そこで TextTimeline では、テキスト表示を優先しながら音響特徴を文字の周辺に埋め込むが、その際には音声の時間軸を保ち、詳細な音響特徴の可視化も可能にする。

1. はじめに

字幕のように、人間の発話内容を記述した文字テキストは多様な場面で存在するが、その中でも、楽曲中の歌詞や外国語学習におけるテキストなどは、「何を」発声しているかという発話内容に加え、「どう」発声しているかという発話方法が重要となる。例えば、歌唱においては、音高・音量等の可視化（フィードバック）が歌唱能力向上に有効であることが知られている（ [1,2] 等）。同様に、発話訓練を対象としたいくつかの CALL（Computer-Assisted Language Learning）システムにおいても音響特徴の可視化が採用されており [*1]、音高・音量・発話長さを文字テキストと融合させて可視化するインタラクションにより、発話の表現力が改善することが報告されている [3,4]。

また、英会話における発声リズムの誤りは、ミスコミュニケーションの原因となることが知られており [5,6][*2]、アメリカ合衆国における子どもの発話教育においては、速度と正確さに加え、適切な表情付け（proper expression）[7] や自然性（ease or naturalness of reading）[8] も「流暢さ」の定義に加えられている。したがって、「どう」発声しているかを適切に可視化することは、音声を介したコミュニケーションや表現、教育の可能性を拡げると考えられる。

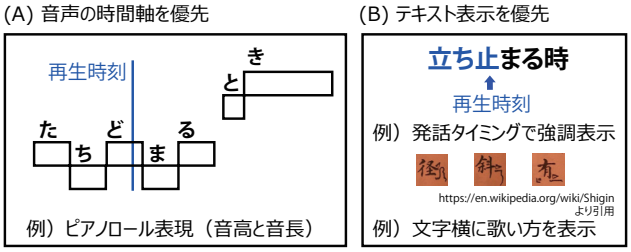

図 1　発話テキストの音響特徴可視化における例。(A) 音声の時間軸を優先させた可視化、(B) テキスト表示を優先させた可視化。

従来、朗読音声や歌声に関して、発話内容の文字テキストと発話の音響特徴を同時に表示する可視化が数多く提案されてきた。ここで、テキストの各文字は固定的な幅（あるいは各文字の表示に適した幅）を持つフォントで表現されるが、各文字に対応する発話音声の時間長は可変であるため、表示の際にどちらを優先させるかで、二通りの可視化方法がある（図 1）。

(1) 音声の時間軸を優先　発話テキストの表示が音声の制約を受ける。話声・歌声の時間軸に合わせて、テキストの分割表示が必要。

(2) 文章・歌詞の表示を優先　発話テキストの表示が音声の制約を受けない。各文字に音響特徴を付与するか、

[1] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)
[a)] t.nakano [at] aist.go.jp
[b)] jun.kato [at] aist.go.jp
[c)] m.goto [at] aist.go.jp

[*1] `https://en.wikipedia.org/wiki/Rosetta_Stone_(software)` 等。

[*2] “McDonald’s” という英単語は 3 音節であり、2 音節目に語彙強勢（lexical stress）を担う。しかし日本語では “makudonarudo” の 6 音節での発音となり、後ろから 3 番目にアクセントが置かれる [6]。また、日本人英語に触れた経験が少ないアメリカ人被験者に対し、日本人英語の理解度が低かったという報告がある [5]。

文字の横に音響特徴を可視化する。

例えば、前者にはピアノロールや楽譜等、後者にはカラオケ歌詞のハイライト表示や吟詠譜（詩吟の楽譜表現）における発声方法の表示等がある。

前者のように音声の時間軸を優先させる場合、文字に制約を受けることなく、音響特徴を自由に可視化できる。これは、「どう話すか」「どう歌われているか」の表示を優先させているといえる。その際、テキスト表示は発話時間に対応付けることになるが、表示位置が発話内容によって制限されるため、表示するスペースを確保できなかったり、文字（単語）間に大きな空白が生じてしまう場合がある。したがって、朗読したい文章や歌詞のストーリーを把握する上で、単語間の大局的な関係を把握しにくくなってしまう可能性がある。例えば、日本語に関しては文節間へのスペース挿入は冗長で読み速度の低下をもたらす結果が報告されており [9]、また文字テキストの読みやすさが、その提示方法（文字配置や改行位置）に影響することが知られているが [10–12]、そのような知見を活かしにくい。

一方、後者のように文字表示を優先させる場合、文字の表示位置やサイズに音声側からの制約がないため、テキストの大局的な関係を保存したり、表示の仕方を工夫した表現手段となりうる。したがって、「何を話す」か、もしくは「何が歌われている」かの表示が優先できる。しかし、テキストに合わせるために、従来、音声の時間軸を非線形伸縮させて可視化する必要があり、各音長や発音タイミングを事前に把握しにくい [*3] 。そのような音長や発音タイミングは、歌声及び話声においても重要である。例えば、英語では音長を stress 知覚の手がかりにしていることが知られており [13]、また日本語や韓国語における母音の無声化が自然な発音には重要であるが、それには発音開始タイミングや音長も関係する。さらに非線形伸縮は、実際の分析結果を圧縮させる必要性が生じる場合がある [*4] 。

そこで本稿では、テキスト表示を優先させて意味的な内容を把握することを可能としながら、その発声方法も詳細に把握できるような可視化方法を提案し、TextTimeline と呼ぶ。TextTimeline では、発話音声の時間軸をテキスト表示に合わせて、変形もしくは回転させるが非線形伸縮は行わない。これによって、音声の時間軸通りの情報表示を可能とし、音声の発声内容の分析結果をテキスト付近に表示させる応用に向けた、共通的な可視化基盤の構築を目指す。

[*3] 例えばカラオケ歌詞等、発音に合わせた文字のハイライト表示も、音声の時間軸を文字表示に合わせて非線形伸縮した結果である。またこのような動的なハイライト表示はさらに、特定の分析結果を確認するために、クリックしたり時間を指定するなど何らかの手段が必要となり、一覧表示には向かない。

[*4] 例えば、音節単位の相対音高表示は、平均操作によって情報が圧縮される。また、無音や文字間の音などを表示しにくい。

2. 関連研究

従来、テキストと対応付いた発声を可視化する手法は、数多く提案されてきた。それらは前述のように、音声優先表示とテキスト優先表示の二種類に加え、さらに時間軸の粒度の観点からも分類できる。具体的には、音声単位もしくはテキスト単位での音響特徴の表示であり、前者は音響特徴そのもの、後者は文字に合わせた圧縮表示となる。

また、TextTimeline においては音声の時間軸を変形するが、時系列メディアの時間軸を変形させる研究がある。

2.1 音声優先表示＋音声単位特徴

音声の時間軸を優先させて、音声単位で音響特徴を可視化する方法は、従来広く使われてきた。このような方針における、歌声の発声内容を可視化する一般的な方法として、楽譜とピアノロール表示がある。音高の違いを位置の上下で表現した上で、楽譜では各音価が記号化され、ピアノロールでは音長とともに矩形で表現される。通常の楽譜では音符に合わせるように歌詞が併記され、ピアノロールでは各歌詞の音節（日本語の場合、ひらがな）等や単語が対応付けられる。

ピアノロールは、カラオケの画面や、歌声合成や歌唱支援においても用いられる。Kenmochi *et al.* による歌声合成ソフトウェア VOCALOID の制作画面 [14]、中野 他による歌声合成パラメータ推定インタフェース VocaListener [15]、そして、香山 他による合唱学習支援システム [16] のそれぞれは、ピアノロール表示に合わせて歌詞の音節が表示されていた。また Nakano *et al.* による、歌声録音インタフェース VocaRefiner では、音高・音量・音色変化に、歌詞のテキストが形態素単位で対応付けられ、音素時刻も可視化していた [17] 。その他、ほとんどは歌詞の表示を伴っていないが、歌声の音響特徴をリアルタイムに視覚フィードバックすることで、歌唱訓練に活用する研究は Howard *et al.* の SINGAD [18] 以降、様々な研究がある [1, 2, 16, 19–21]。

話声に関しては、CALL において音高軌跡等を表示する場合があり、Petal *et al.* は ReadN'Karaoke（V2：統合のみ）で、テキストに音高・音量軌跡を重畳表示させた [3] 。また、Rahman *et al.* は音素単位 [*5] で、調音位置の可視化を行った [22]。その他、テキストの表示を伴っていないが、Schaefer *et al.* は音高と音量を上下位置の違いやサイズの違いで表現し [23] や、Pietrowicz *et al.* は音素の違いを色の違いで可視化した [24]。

これらの研究は、音声単位特徴の可視化であり、固定された分析シフト幅（10ms 等）に基づく可視化、もしくは、音素単位に基づく可視化がなされてきた。その点では TextTimeline と関係があるが、テキスト位置が音声に制約

[*5] 音素は、文字で表現されるが音声単位として扱う。

を受ける点で異なる。

2.2 音声優先表示＋テキスト単位特徴

音声を優先表示しながらテキスト単位で音響特徴を可視化するためには、音響特徴を文字のレイアウトやデザインに埋め込む必要がある。

歌声においては、詩吟の楽譜表現において、歌詞（文字）を五線譜上に配置することがある [25]。Diaz-Marino *et al* は、カラオケのようなハイライト表示に加え、歌の音高に基づいて、音節単位で歌詞テキストを上下させて表示する LyricText を提案した [26]。そこでは、声の種類（Spoken, Sonorant, and Yell）を色で表現した他、長めの無音の次の発音開始をアニメーションで表現した。また Kato *et al.* による歌詞アニメーション制作支援環境 TextAlive [27] は、発音タイミングという特徴に基づいた Kinetic Typography のデザインを支援する。

話声においては、Petal *et al.* による ReadN'Karaoke（V1：音高・音長・音量・統合の全て）では、音高・音長に基づいて文字位置を移動して表示し、音量に基づいて濃さを変更した [3]。また Rude は、文字の配置やサイズ・太さを音長・音高・音量に応じて変形する提案を行った [4]。

以上の手法は、テキストの位置が音声に合わせて変更され、音響特徴もテキスト単位で可視化されるため、TextTimeline とは方針が異なる。

2.3 テキスト優先表示＋音声単位特徴

テキスト表示を優先させながら音声単位特徴を可視化する方法の一つに、カラオケにおける採点画面等、歌詞のテキストと音響特徴を併記してそれぞれで時間軸を表現する方法がある。つまり、発音タイミングに合わせた歌詞が強調表示されるとともに、音響特徴上の時刻も移動し、それらの絶対的な表示位置が異なる。その他、Petal *et al.* による ReadN'Karaoke（V2: 音高・音量）[3] があり、音高と音量の時間軸をテキストに合わせて非線形伸縮した。

このような可視化方針は、音高・音量軌跡等を確認可能で、TextTimeline と方針が最も近い。しかし、前者は音符と音節などの細かな対応が表示できず、後者は時間軸の非線形伸縮によって音長やタイミングを把握しにくい。

2.4 テキスト優先表示＋テキスト単位特徴

テキストの表示を優先させて、発話内容を可視化する一般的な方法として、カラオケにおける歌詞のハイライト表示がある。そのような表示に関する研究としては、Fujihara *et al.* による歌詞テキストを歌声に同期させる LyricSynchronizer [28] や、さらに類似歌詞も表示させる Nakano *et al.* の LyricListPlayer [29] などが提案されてきた。

その他、文字領域へ情報を埋め込むことで、テキストの表示位置を音声と独立にデザインできる方法として、文字横に歌唱スタイルを可視化する方法がある（例えば、吟詠譜 [25] [*6]）。また、通常のテキストに対する情報埋め込みには Tufte の sparklines があり [30]、それを音楽に応用した、歌詞の音節毎の音高、コード、リズムを埋め込む Oh の Musical sparklines がある [31]。そこではさらに、意味ある単語を太くしたり、音楽とテキストの大局的な繰り返しも可視化されており [31]、focus+context（fisheye）表現の一種として考えられる [32, 33]。その他、松浦 他による声質に応じた文字フォントの変更に関する研究がある [34]。

前節同様、これらは、テキストの大域的な表示も可能とするが、音響特徴はさらに圧縮した可視化を必要とするため、可視化可能な情報が制限される。

2.5 時間軸の変更

TextTimeline では、音声の時間軸を変形させるが、それと関連して、シークバーを拡張する研究がこれまで多く研究されてきた。例えば、シークバーを切り貼りしたり曲げたりして音楽制作を行う青木 他による SeekRopes [35] や、探索的データ分析を目的としてシークバーを変形させる高嶋 他による TbVP Browser [36] などである。

しかしこれらは、テキスト等、時系列メディア以外との対応付けをしていない点で、本研究とは目的が異なる。

3. TextTimeline のデザイン

TextTimeline は、テキスト優先表示を行い、かつ音響特徴量を音声の時間軸通りに表示する。ただし、音声時間軸の方向等によって様々な実現方法が考えられるため比較検討する。本章では、TextTimeline における可視化方法のデザインにおいて、満たすべき制約条件と、各デザインにおける特徴（以降、素性と呼ぶ）を述べる。その後、実現可能な様々なデザインの実例について、それぞれの特徴を議論する。

本稿では、日本語と英語に対応することを想定し、横書きのテキストを前提とする。また、テキストの単位としては、カラオケや Kinetic Typography のように一文字ずつ時間を表現するのではなく、リズムを表現し、意味のある発音の単位としての音節を用いる。ただし、表示上の文字数と音節の数には違いがあることが多いため、まず日本語は形態素、英語は空白で区切られた単語に分割する。その後、形態素もしくは英単語を音節の数で分割して扱う。

3.1 制約

テキストと音響特徴量を融合的に可視化するために、デザインにおいて以下の制約を設ける。

発音時刻を同一のカーソルで表すこと 発音されている音声に対応する文字と、その音響特徴量の時刻を表す

[*6] https://en.wikipedia.org/wiki/Shigin

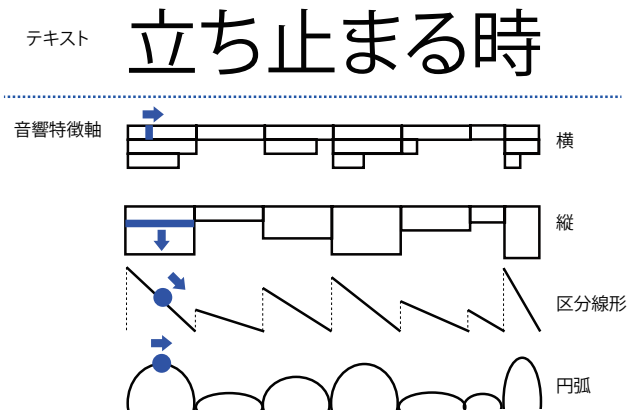

図 2 4 種類の可視化デザイン案。全て「発音時刻を同一のカーソルで表す」「カーソルが等速移動する」の二つの制約を満たす。

カーソル（の機能を持った表現）が一つにまとまっていること。それぞれが独立に表現されている場合、再生前に文字と音響特徴の対応関係を把握することができず、また再生中の対応付けも困難である。

カーソルが等速移動すること　カーソルが等速で動くためには、音声の時間軸を伸縮なく保存する必要がある。それによって、発音タイミングも適切に表現できる。

2.3 節における、テキストと音響特徴の併記は前者の制約を満たしておらず、Petal *et al.* による ReadN'Karaoke（V2: 音高・音量）[3] は後者の制約を満たしていない。

以上の制約を満たすように TextTimeline をデザインする必要があるが、横方向のテキストに対して音響特徴の時間軸としては、「横」「縦」「それ以外」の三種類の方向が選択可能である（詳細は 3.3 節で後述）。

3.2 素性

デザインの特徴を議論するために、以下の素性を考える。これらは、同時に複数を満たすことが難しい場合があるが、使用可能な状況を議論することで、使い分けが可能となる。

時間軸が常に繋がっているか　視線の移動量を最小にするためには、時間軸がつながっている必要がある。もし時間軸が途中で途切れている場合（以降、跳躍と呼ぶ）、ユーザが次の位置を探す必要があり、場合によってはカーソルを見失う可能性がある。

時間軸が同じ方向に動くか　カーソルが常に左から右へ移動するなど、同一の方向に動くことで、それと直交する方向に同じ意味を持たせやすくなり、結果として軸上の音響特徴を比較しやすくなる。

可視化に必要な領域サイズ　テキスト付近に情報を埋め込む場合、領域が必要となる。そのサイズが大きいと、複数行の場合にテキスト優先表示を保ちにくい。

この他、「可視化位置」も素性として考えられるが、本稿では全て文字の下（行間）に配置する前提での議論を行う。従来、内容に関連した情報のテキスト中への配置において

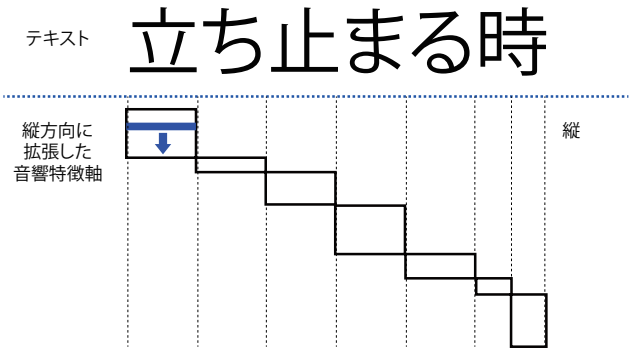

図 3 「縦」を拡張して跳躍をなくしたデザイン案。

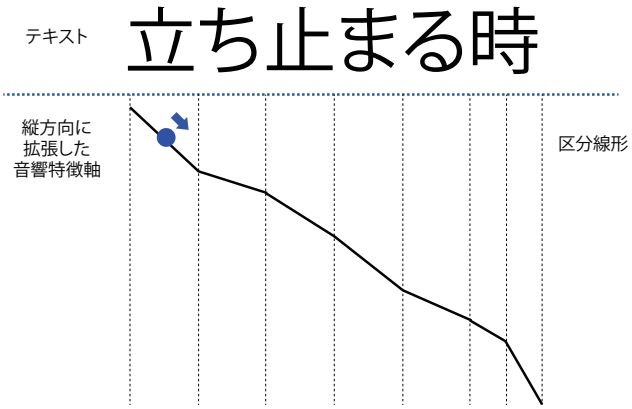

図 4 「区分線形」を拡張して跳躍をなくしたデザイン案。

Goffin *et al.* は 7 種類の配置方法を提案したが [37]、「行間」への配置が、情報同士の重なりがなく、テキスト表示に制約が少なく、音響特徴の確認にマウスオーバーなどのインタラクションを要さない点で適していると判断した。

ここで、音響特徴を行間に配置するためには領域が必要となり、複数行の文字表示に関しては一種の制約となる。しかし、その領域のサイズや意味が、音声に依存して変わらなければ、テキストのデザインに影響が少ないと考える。場合によっては、普段は行間を折りたたんでおいたり [38]、下の行に重畳表示させる可視化を行えば [39]、通常のテキストに適用可能である。

3.3 デザイン案

前節までで議論した制約を満たす音声軸について、4 種類のデザインとして「横」「縦」「区分線形」「円弧」を**図 2** に、それぞれの素性を**表 1** の上 4 行に示した。「横」は、テキストと同じ横方向の音声軸を、音節単位で「改行」したデザイン案である。これによって、非線形伸縮することなく、音節に合わせた時間軸を実現できるが、改行による跳躍を含んでしまう。そのような改行をなくすために、時間軸を「縦」にすると、改行は不要となるが音節間の跳躍が残る。さらに、この跳躍を減らすために「区分線形」を、跳躍をなくすために「円弧」の軸が実現可能である。

まずはこれらを比較すると、「横」は音節の長さによって改行の数が変わってしまうため、音節間の長さの比較が難

表 1　各デザインの素性：軸に跳躍しない場合、カーソルが同一方向に移動する場合、領域サイズが相対的に小さい場合に○、それ以外を×とした。ただし、領域サイズに関してのみ、 fisheye view を採用することで、改善できたことを△で示した。

軸のデザイン	跳躍なしか	同一方向か	領域サイズ
横	×	○	○
縦	×	○	○
区分線形	×	×	○
円弧	○	×	○
縦（拡張）	○	○	×
区分線形（拡張）	○	×	×
縦（fisheye）	○	○	△
区分線形（fisheye）	○	×	△

しいが、それ以外は、どの音節が長く発声されているかを視覚的に把握しやすい。しかし、「横」「縦」「区分線形」にはそれぞれ跳躍が発生してしまう。最も跳躍が少ない「区分線形」を実装して確認したところ、直線上においてはカーソルは等速に移動するが、跳躍が含まれることで等速移動を感じにくかった。したがって跳躍を伴うと、音声再生中に音響特徴を確認しにくい可能性がある。一方、「円弧」には跳躍が含まれないが、軸が直線でなくなるため、ここに音響特徴を描画した場合に比較しにくい可能性がある。

ここで、「縦」「区分線形」を縦方向に拡張すると跳躍を完全になくすことができるため、それら 2 種類のデザインを**図 3** 及び**図 4** に、素性を表 1 の 5,6 行目に示す。しかしそのままでは、文字数が増えるごとに領域サイズが単調増加するため、文字数に依存せずに領域サイズを固定するために、fisheye view (focus+context) 表現を採用して [32, 33] 実装した。素性を表 1 の 7,8 行目に示す。このようにして拡張した「縦」のデザインは、全ての素性において、望ましい性質を持っている。

以上から、本研究の目的においては fisheye view を導入した「縦」の拡張デザインが、相対的に適していると考えられる。

4. 実装及び検証

前章で述べたように、fisheye view を導入した「縦」の拡張デザインの実装画面を**図 5** に示し、これ以降はこのデザインのインタフェースを TextTimeline と呼ぶ。本章では、TextTimeline の実装方法と、その有効性を検証する。

4.1 インタフェース

本実装では、音量を TextTimeline 上に描画した（図 5）。各音節における音量は、左右対称な幅で表現され、音節の中央にセンタリングされる。音声軸を縦にしたことで、音量を示す横幅が増えても重複しないため、全ての音節の音量は、比較可能なように同じ幅で表現した。

可視化された音響特徴量は、音声の再生とともに上方向

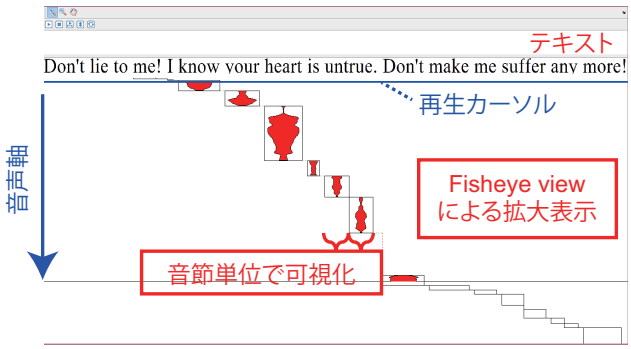

図 5　「縦」を拡張して跳躍をなくし、fisheye view を導入した TextTimeline の実装画面。

にスクロール表示される。テキスト下部の直線が再生カーソルであり、音響特徴量がカーソルにたどり着いた際に、該当する音声が再生される。

4.2 アラインメント

発話とテキスト文字の対応付けには、LyricSynchronizer [28] 及び LyricListPlayer [29] と同様の手法を用いた。具体的には、発話された音声信号から MFCC、ΔMFCC、Δ パワーを 10ms 毎に抽出し、日本語音素と英語音素の HMM に基づいた音響モデル（monophone HMM）を用いて認識した。日本語の読みは MeCab [40] で形態素及び読みを推定し、それを音素列に変換した。また、英語の読みは The CMU Pronouncing Dictionary [41] を用いて得た。

音節の数は、形態素・単語中の母音の数によって決定する。日本語については、「母音」もしくは「子音と母音」を一つの音節として扱った。英語については、各母音の前後の子音を付与した分割を行った。このようにして得た音節を文字と対応付ける必要があるが、英単語や漢字など、文字の境界を音節情報から明確に決められないことが多い。そこで、形態素・単語単位の文字長さを音節数で等分割して、文字表示上の区間とした（図 5）。

4.3 ユーザによるフィードバック

TextTimeline の特徴を検証するために、6 人のユーザ（男性 5 人: U1-U4, U6、女性 1 人: U5）に、以下の三種類のインタフェースを使用してもらい、感想を得た。

D1)　テキストと音声の並行表示（**図 6**）
D2)　表 1 における「縦」（図 6）
D3)　TextTimeline（図 5）

対象音声は、AIST ハミングデータベース [42] に含まれる、一人の女性が歌った二つの伴奏なし歌唱とその朗読音声である。ユーザは D1〜D3 を使いながら、それらの音声を 20 分程度聴取した。

得られたフィードバック（Pros/Cons）を、まとめながら以下に示す。まず発音タイミングに関して、従来デザインの D1 については、発音タイミングや音量の大小の比較

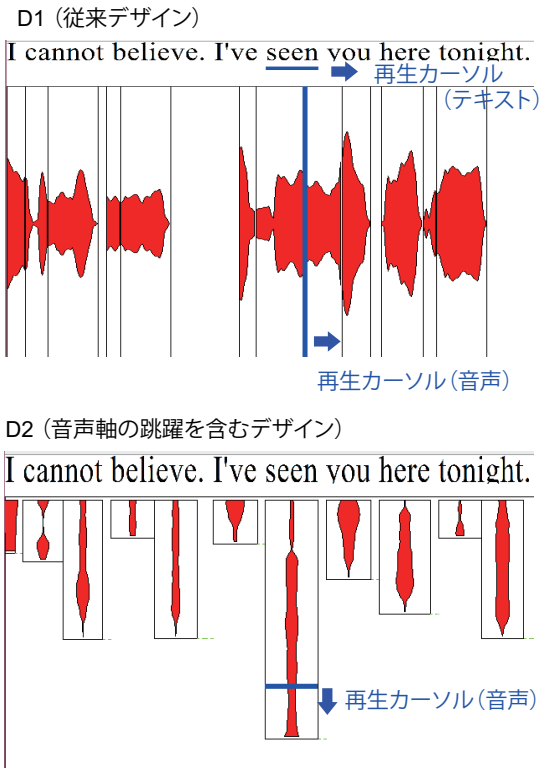

図 6 提案デザイン D3（TextTimeline）と比較するための、従来デザイン（D1）及び比較デザイン（D2）。D2 には時間軸の跳躍が含まれるため、その影響を調査する目的がある。

が分かりやすいというコメントとともに、「文字と音量の対応がわかりにくい (U1, U2, U3, U4 and U6)」というコメントを得た。それに対して、D3 について「各音節のタイミングが予測しやすい (U1, U2, U3 and U4)」というコメントが得られた。これは、これまでの考察と対応する。また D2 については、無音後の発音開始が分かりにくいというコメントがあったが、今回の実装において無音を表現していなかったことが原因である可能性があり、今後改善して再確認したい。

また、D3 の音量表示に関しては、文字と音量の対応が分かりやすく、文字に沿って流れるデザインの新しさや楽しさに関するコメントがあった。しかし一方で、フレーズが長い場合に音量が見えにくかったり、文字と音量に距離があって対応が分かりにくいというコメントもあり、改善点が一部明らかになった。逆に、D2 では文字と音量の対応が分かりやすく、各音節の長さが把握しやすいというコメントがあったことから、D2 と D3 を融合させたような表示によって、より改善できる可能性がある。

5. おわりに

本稿では、発話を伴う文字テキストの可視化について、文字の表示を優先させながら、その発話音声の時間軸を可視化する TextTimeline を提案した。音声の時間軸通りに、テキストと音響特徴量を融合的に可視化するために、インタフェースデザインにおいて制約を設けて 8 種類のデザイン案を比較した。その中から、横方向のテキストに対して、それと直交する縦方向の時間軸を設定したデザインを採用し、時間軸の跳躍をなくし、fisheye view によって文字数に依存しにくい可視化領域の実装を提案した。

最終的なデザインでは、可視化に必要な領域サイズが大きくなったが、普段は行間を折りたたむ [38]、もしくは下の行に重畳表示させる可視化 [39] により、複数行テキストへの適用も可能だと考えている。またユーザからのフィードバックより、改善の可能性が示唆されたため、音量以外の音響特徴量の可視化を含めて、今後の研究を進めていく。特に、音高の可視化が課題である。

謝辞

本研究の一部は、JSPS 科研費 JP17K12721 および JST ACCEL (JPMJAC1602) の支援を受けました。また、AIST ハミングデータベースを使用しました。

参考文献

[1] Hoppe, D., Sadakata, M. and Desain, P.: Development of real-time visual feedback assistance in singing training: a review, *Journal of computer assisted learning*, Vol. 22, pp. 308–316 (2006).
[2] Moschos, F., Georgaki, A. and Kouroupetroglou, G.: FONASKEIN: An Interactive Application Software for the Practice of the Singing Voice, *Proc. SMC 2016*, pp. 326–331 (2016).
[3] Patel, R. and Furr, W.: ReadN'Karaoke: Visualizing Prosody in Children's Books for Expressive Oral Reading, *Proc. CHI 2011*, pp. 3203–3206 (2011).
[4] Rude, M.: Native-like Duration Ratio of Stressed vs. Unstressed Syllables through Visualizing Prosody, *Proc. Speech Prosody 2012* (2012).
[5] Minematsu, N., Okabe, K., Ogaki, K. and Hirose, K.: Measurement of Objective Intelligibility of Japanese Accented English Using ERJ (English Read by Japanese) Database, *Proc. Interspeech 2011*, pp. 1481–1484 (2011).
[6] ドナエリクソン：英語のリズムと第二言語教育への応用，日本音響学会誌， Vol. 69, No. 4, pp. 184–190 (2013).
[7] National Institute of Child Health and Human Development: *Report of the National Reading Panel. Teaching Children to Read: An Evidence-based Assessment of the Scientific Research Literature on Reading and its Implications for Reading Instruction*, US Government Printing Office (2000).
[8] of Educational Progress (NAEP), N. A.: Listening to Children Read Aloud: Oral Fluency, http://nces.ed.gov/pubs95/web/95762.asp.
[9] Sainio, M., Hyönä, J., Bingushi, K. and Bertram, R.: The role of interword spacing in reading Japanese: An eye movement study, *Vision Research*, Vol. 47, No. 20, pp. 2575–2584 (2007).
[10] 中野聡子，金澤貴之，牧原 功，黒木速人，上田一貴，井野秀一，伊福部達：音声認識技術を利用した字幕呈示システムの活用に関する研究―聴覚障害者のニーズに即した呈示方法―，メディア教育研究，Vol. 5, No. 2, pp. 63–72 (2008).
[11] 村田匡輝，大野誠寛，松原茂樹：読みやすい字幕生成のための講演テキストへの改行挿入，電子情報通信学会論文誌，Vol. J92-D, No. 9, pp. 1621–1631 (2009).

[12] 小林潤平，関口　隆，新堀英二，川嶋稔夫：文節単位を考慮した文字配置の工夫がもたらす日本語電子リーダーの可読性向上，人工知能学会論文誌 32(2A)， Vol. 32, No. 2A, pp. 1–24 (2017).
[13] Fry, D. B.: Intuitive visualization of pitch and loudness in speech, *JASA*, Vol. 27, pp. 765–768 (1955).
[14] Kenmochi, H. and Ohshita, H.: VOCALOID – Commercial Singing Synthesizer based on Sample Concatenation, *Proc. 8th Annual Conference of the International Speech Communication Association (INTERSPEECH 2007)* (2007).
[15] 中野倫靖，後藤真孝：VocaListener：ユーザ歌唱の音高および音量を真似る歌声合成システム，情報処理学会論文誌，Vol. 52, No. 12, pp. 3853–3867 (2011).
[16] 香山瑞恵，中西　将，岡部真実，浅沼和志，伊東一典，為末隆弘，橋本昌巳：指導者知識に基づく合唱学習支援システムの構築とその評価，情報処理学会論文誌，Vol. 51, No. 2, pp. 365–379 (2010).
[17] Nakano, T. and Goto, M.: VocaRefiner: An Interactive Singing Recording System with Integration of Multiple Singing Recordings, *Proc. SMAC-SMC2013* (2013).
[18] Howard, D. M. and Welch, G. F.: Microcomputer-based singing ability assessment and development, *Applied Acoustics*, Vol. 27, pp. 89–102 (1989).
[19] 平井重行，片寄晴弘，井口征士：歌の調子外れに対する治療支援システム，電子情報通信学会論文誌，Vol. J84-D-II, No. 9, pp. 1933–1941 (2001).
[20] Nakano, T., Goto, M. and Hiraga, Y.: MiruSinger: A Singing Skill Visualization Interface Using Real-Time Feedback and Music CD Recordings as Referential Data, *Proc. ISMW 2007*, pp. 75–76 (2008).
[21] Mayor, O., Bonada, J. and Loscos, A.: Performance Analysis and Scoring of the Singing Voice, *Proc. AES 35th International Conference* (2009).
[22] Abdul-Rahman, A., Lein, J., Coles, K., Maguire, E., Meyer, M., Wynne, M., Johnson, C. R., Trefethen, A. and Chen, M.: Rule-based Visual Mappings – with a Case Study on Poetry Visualization, *Proc. EuroVis 2013* (2013).
[23] Schaefer, R. S., Beijer, L. J., Seuskens, W., Rietveld, T. C. M. and Sadakata, M.: Intuitive visualization of pitch and loudness in speech, *Psychonomic Bulletin & Review*, Vol. 23, pp. 548–555 (2016).
[24] Pietrowicz, M. and Karahalios, K. G.: Visualizing Vocal Expression, *Proc. CHI 2014* (2014).
[25] Nakayama, M.: Fundamental research on a singing training support system for Shigin: Japanese traditional singing, *Proc. IEEE SoutheastCon 2012*, pp. 1–6 (2012).
[26] Rob Diaz-Marino, S. C. and Greenberg, S.: LyricText: An Animated Display of Song Lyrics, *Report iLab-2005-1, Grouplab, Dept. Compute Science, University of Calgary*, pp. 3203–3206 (2005).
[27] Kato, J., Nakano, T. and Goto, M.: TextAlive: Integrated Design Environment for Kinetic Typography, *Proc. CHI 2015*, pp. 3403–3412 (2015).
[28] Fujihara, H. et al.: LyricSynchronizer: Automatic Synchronization System Between Musical Audio Signals and Lyrics, *IEEE Journal of Selected Topics in Signal Processing*, Vol. 5, No. 6, pp. 1252–1261 (2011).
[29] Nakano, T. and Goto, M.: LyricListPlayer: A Consecutive-Query-by-Playback Interface for Retrieving Similar Word Sequences from Different Song Lyrics, *Proc. SMC 2016*, pp. 344–349 (2016).
[30] Tufte, E.: *Beautiful Evidence*, Graphics Press (2006).
[31] Oh, J.: Text Visualization of Song Lyrics, *Center for Computer Research in Music and Acoustics* (2010).
[32] Furnas, G. W.: Generalized fisheye views, *Proc. ACM CHI'86*, pp. 16–23 (1986).
[33] Rao, R. and Card, S. K.: The table lens: merging graphical and symbolic representations in an interactive focus+context visualization for tabular information, *Proc. ACM CHI'94*, pp. 318–322 (1994).
[34] 松浦泰仁，角谷亮祐，秋庭祐貴，菅沼佑太，菊川裕也，馬場哲晃，串山久美子：声質ヴィジュアライザの提案，情報処理学会インタラクション 2012 論文集，pp. 451–456 (2012).
[35] 青木惇季，宮下芳明：SeekRopes：複数スライダとシークロープによる音楽制作，情報処理学会インタラクション 2011 論文集，pp. 429–432 (2011).
[36] 高嶋章雄，山本恭裕，中小路久美代：探索的データ分析のための時間的な概観と詳細の表現およびインタラクションに関する研究，情報処理学会論文誌，Vol. 44, No. 11, pp. 2767–2777 (2003).
[37] Goffin, P., Willett, W., Fekete, J.-D. and Isenberg, P.: Exploring the Placement and Design of Word-Scale Visualizations, *IEEE Trans. on Visualization and Computer Graphics*, Vol. 20, No. 12, pp. 2291–2300 (2014).
[38] Yoon, D., Chen, N., Guimbretiere, F. and Sellen, A.: RichReview: Blending Ink, Speech, and Gesture to Support Collaborative Document Review, *Proc. UIST 2014* (2014).
[39] Chang, B.-W., Mackinlay, J. D., Zellweger, P. T. and Igarashi, T.: A Negotiation Architecture for Fluid Documents, *Proc. UIST'98*, pp. 123–132 (1998).
[40] Kudo, T.: MeCab: Yet Another Part-of-Speech and Morphological Analyzer, http://mecab.sourceforge.net/.
[41] at CMU, S.: The CMU Pronouncing Dictionary, http://www.speech.cs.cmu.edu/cgi-bin/cmudict.
[42] 後藤真孝，西村拓一：AIST ハミングデータベース：歌声研究用音楽データベース，情報処理学会研究報告音楽情報科学研究会，2005-MUS-61, pp. 7–12 (2005).