

## 個人的な小さな幸せを実現するブラウザ上での動画編集・共有手法

中村 聡史\* 石川 直樹† 渡邊 恵太‡

**概要.** ニコニコ動画では、動画を一緒に楽しむだけではなく、もともとある動画をもとに、それを改良し、さらなる動画とする N 次創作がそのコミュニティの一体感と、動画の再発掘に有効に働いている。しかし、こうした N 次創作はある程度の創作が前提となっているものであり、例えば音量の変更や、音楽の変更や、部分的な音声の差し替え、音声の追加といった事だけでは創作とはなりえないものである。そうしたちよつとした編集だけを行った動画をアップロードすると、パクリとなじられるであろう。本研究では、そうした微細な編集によって個人的な満足感を達成可能とするだけでなく、その編集を共有することで新たな動画視聴体験を可能とする仕組みを実現する。ここでは、「ブラウザ中で視聴中の動画に対する音量増減や音声の付与などの編集手法」と「付与された動画の共有手法」を実現とする。これにより、部分的に音量を調整したり、音楽を差し替えたり、音声を足したり、部分的にスロー再生やスキップしたいなど個人的な嗜好に基づく動画編集および視聴を可能とする。

### 1 はじめに

YouTube<sup>1</sup>やニコニコ動画<sup>2</sup>に代表される動画共有ウェブサイトが爆発的に成長している。YouTube では 2012 年 1 月時点で 1 秒あたりに動画長にして 1 時間分の動画が投稿され<sup>3</sup>ている。また、ニコニコ動画でも、2012 年 11 月 20 日の中村らの調査 [1] によると、1 日あたりに約 5500 本の動画が、1 秒あたりに動画長にして約 47 秒分の動画が投稿されている。YouTube は 2012 年 1 月の時点で 1 日に 40 億視聴、ニコニコ動画では 2012 年第 2 四半期に 1 日あたり 1 億視聴、平均視聴者数は 786 万人で、ユーザの平均視聴時間は 102.5 分であるという<sup>4</sup>。

ニコニコ動画では、先人が創作した動画を活かし、新たな動画を創作するという N 次創作 [2, 3] が多く見られる。N 次創作では、元々 VOCALOID などで作成および投稿された音楽に対して、それにマッチする映像を付けてさらにアップロードしたり、ピアノなどで曲を弾き直してみたり、曲はそのままに歌を自分が歌ったもので差し替えてみたり、踊りを考えて音楽に合わせて踊ってみたりなど、多様な発展が見られる。こうした N 次創作は、ニコニコ動画を発展させ、コンテンツをより良い物へと高めるとともに、愛されるコンテンツを生み出している理由の一つでもある。

ここで、ニコニコ動画で N 次創作に参加するハードルについて考える。オリジナルのコンテンツに対して「歌ってみた」「演奏してみた」「踊ってみた」などタグが付けられているような動画は N 次創作の 1 つである。ちなみに、2013 年 8 月 29 日時点で、ニコニコ動画上で「歌ってみた」で検索される動画は 691,641 件、「演奏してみた」動画は 166,800 件、「踊ってみた」動画は 100,078 件ある。YouTube で「歌ってみた」で検索される動画は約 904,000 件、「演奏してみた」動画は約 586,000 件、「踊ってみた」動画は約 540,000 件ある。なお、海外での「歌ってみた」に相当する（吹き替えも含まれる）「Fandub」で検索すると、その件数は約 1,960,000 件となる。

N 次創作は、オリジナルとなるコンテンツ（動画）を改良するものであるため、参入障壁は、コンテンツを最初から作成するよりは低いように感じられる。しかし、この「歌う」「演奏する」「踊る」という事であっても技量が要求され、ハードルが高いといえる。さらに、アレンジする場合は、その技術力も必要となる。映像をつけるのも同様のハードルの高さが存在するであろう。結果として、N 次創作は行われているものの、そこに参入しているユーザの総数は限られる。また、動画編集ソフトウェアを用いた動画編集や、動画のエンコード、動画のアップロードなどにも障壁は存在する。

ここで、ユーザが動画を視聴している際に、その動画に対して微妙に編集したいと思うことがある。例えば、ユーザが気に入っている楽曲動画なのだけれど、音量が終盤でフェードアウトしておらず、ぶつくと切れてしまうためフェードアウトさせたいと感じることや、作業用 BGM として音楽のメドレーを聴いている際に、ある曲だけは苦手なのでスキップしたいと感じることなどもある。それ以外にも、折角

Copyright is held by the author(s).

\* 明治大学 総合数理学部, JST CREST

† 東京農工大学 工学部

‡ 明治大学 総合数理学部, JST CREST

<sup>1</sup> <http://www.youtube.com/>

<sup>2</sup> <http://www.nicovideo.jp/>

<sup>3</sup> <http://jp.techcrunch.com/archives/20120123youtube-reaches-4-billion-views-per-day/>

<sup>4</sup> <http://dic.nicovideo.jp/a/ニコニコ動画>

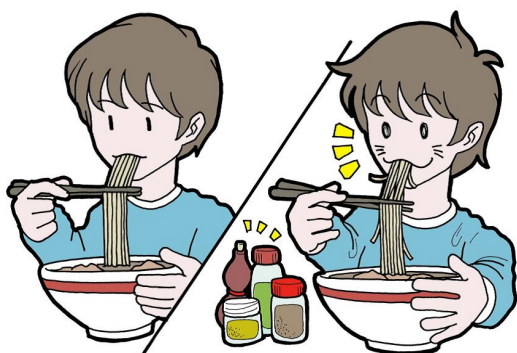


図 1. (左) 調味料をたさずにそのまま食べる (右) 調味料を足して自分好みに調整し食べる

の素晴らしい演奏動画なのだけれど普通に終わってしまうため拍手や歓声などを入れたいなどと感じることや、映像自体は好きなものだけれど BGM があまり好きなタイプではないため、BGM を差し替えたなど様々である。しかし、仮に高価な動画編集ソフトウェアをもっており、編集能力が高いユーザであったとしても、他者が投稿した動画をダウンロードし、音量を微調整だけしたり、部分的に音を追加したり、部分的にスキップしたりといった、微細な変更だけでアップロードするということは世の中から N 次創作として許容されるとは考えにくい。そうした動画をアップロードすると、他の視聴者から「パクリ」と糾弾されてしまうこともあるであろう。

そこで我々は、上記のような編集を、オリジナルの動画に手を加えるのではなく、システムによって再生している動画自体をコントロールすることによってコンテンツを個人化する手法を提案および実装する。また、他者と共有可能にすることによって、編集も 1 つのコンテンツとして楽しむことが可能な仕組みを実現する。これは、ラーメン屋などにラーメンを食べに行った時に、自分の好みに合うように調味料を足すことに似ている (図 1)。

本提案手法によって、N 次創作とは違う、よりカジュアルに動画と係ることが可能な N 次装飾とも言える一つの動画の楽しみ方を提供する。

## 2 ブラウザ上でのカジュアルな編集と個人化

ニコニコ動画に見られる N 次創作は、その創作のもととなるオリジナルのコンテンツに敬意を払いながら、利用するという点でとても良いコンテンツ生態系を作り出していると言える。しかし、動画の作成者 (N 次創作の作成者も含む) は、動画の視聴者数に比べたら圧倒的に少ない。これは、ニコニコ動画自体にアップロードされている動画の平均的な質が高いために、気後れしてしまっているという点もあるが、そもそも動画作成および N 次創作のハードルの高さも原因の 1 つでもある。

ここで、オリジナリティとクオリティが高い楽曲や映像の動画があり、多くの人から評価されているが、ユーザ個人としては「微妙に気持ち悪いから修正して欲しい」とか、「もう少しこうであつたら良いのに」などと感じることがある。実際、そうしたコメントはニコニコ動画中のコメントで現れていることも多く、不満コメントに対し「あなたが作りなおしてアップロードしたら良いじゃないか」という旨のコメントが付くことも珍しくない。不満を述べるだけというのは実際生産的ではないが、元々の動画の投稿者以外のユーザが、その高い品質の動画をダウンロードし、ユーザの個人的な好みに合わせて微調整だけした動画をアップロードするというのは、N 次創作とは言いがたいものであり、他者からも受け入れられないであろう。

ここで、ユーザ自身がつ、個人のための動画編集のニーズを調べるため、ニコニコ動画や YouTube などに投稿されている不満や要求に対するコメントを分類および整理し、ピックアップすると下記の通りとなる。

- 動画自体は質の高い映像コンテンツではあるものの、音楽が好みのものではないという、ユーザとしては映像と音楽が合っていないように感じている。そこで、映像はそのままに、音楽だけ好みのものに変更したい。
- ユーザが楽しんでいたストーリー性のある動画において爆発音が入っていたが、その爆発音の部分が好みに合わなかったため、他の爆発音と差し替えたい。
- N 次創作として「歌ってみた」や「演奏してみた」「踊ってみた」という動画は、オリジナルの同一楽曲を元に、各投稿者が歌ったり、演奏したり、踊ってみたいものである。このそれぞれの動画について、踊ってみた動画の映像と歌ってみた動画の音声を組み合わせたり、歌ってみた動画の音声とオリジナル動画の映像を組み合わせたり、ギターで演奏している動画とピアノで演奏している動画を組み合わせたりしたい。
- 作業用 BGM とタグ付けられている動画は、複数の楽曲をセットとして 1 動画とされているものであり、ユーザはこの動画を再生するだけで複数の楽曲を連続的に楽しむことができる。その楽曲の一部が気に入らないのでスキップしたり、ある曲だけ音量が低すぎてバランスが悪いため、その楽曲部分のみ音量を調整したい。
- 研究に関するデモビデオが無音であり面白みにかけていた。そこで、そのビデオに追加音声や、音楽を付けて面白くしたい。

- ある素晴らしい出来の動画があるが、音声がついていない。みんなで音声を持ちよって、この無声動画に音を付与したい。

上記のニーズは一部に過ぎないが、その多くはオリジナリティがあるとは言いがたく、独自に動画を投稿した時に許容されるかどうかは怪しい。つまり、上記のニーズを満たすには、動画として再編集およびアップロードするのではなく、個々の編集情報だけを記録し、その編集記録を元に動画を視聴可能な仕組みの実現が必要となる。我々は、こうした動画の個人化（自身の好みに変更すること）をブラウザ上で実現可能とし、その超個人化された編集情報を他者と共有することを可能とするものである。

### 3 プロトタイプシステム

#### 3.1 必要要件

ユーザ（視聴者）の嗜好に基づき、微妙に動画コンテンツを編集し、個人向けにしたいという欲求を満たすため、我々はウェブブラウザ上で、実際に視聴しながら編集することが可能な編集手法を実現する。システムとして求められる要件を整理すると、下記の通りとなる。

- 視聴中の動画の任意の再生時間に対して、ボリュームを上げる、下げる、消すなどの音量に対するコントロールを可能とする。
- 視聴中の動画の任意の再生時間に対して、一部スキップや、スロー再生などの再生に対するコントロールを可能とする。
- 視聴中の動画の任意の再生時間に対して、任意の音声、音楽追加を可能とする。
- 上記機能を Web ブラウザ上で実現可能とする。
- 動画と編集情報による、新たな動画編集 ID (URL) で動画を管理する。
- 動画編集 ID をベースとして、他者と共有することを可能とする。
- 共有された編集情報を元に、さらなる編集を可能とする。

#### 3.2 実装方法

本提案システムを、Google Chrome の拡張としてプロトタイプシステムを実装した。本システムは、YouTube やニコニコ動画などのページで動画がロードされると、動画の ID を取得 (YouTube やニコニコ動画上で動画を一意に識別する文字列) し、その ID に合致する編集情報の有無を確認する。編集情報が存在しない場合は、動画編集のためのコントロールを提示し、編集情報が存在する場合は、その情報に基づきコントロール上に編集情報を提示する。こ



図 2. システム導入前 (左) と導入後 (右)

こで、コントロールの表示は、ページ中にコントロール用のコードが含まれる DOM を挿入することで実現している。

音量の変更や、動画のスキップ、再生速度変更といった再生に関するコントロールは、ニコニコ動画や YouTube の動画を再生するコントローラの API を利用して操作している。また、付与される音声については、その音声がある URL 上の音楽/音声ファイルの場合は、そのファイルをロードし、時刻に合わせて同期的に再生させる。さらに、YouTube などの動画を追加音声として利用する場合は、別のタブとして動画をロードし、タイミングを合わせて再生するよう実装している。つまり、システムとしては同時に複数のコンテンツを再生しているだけである。

なお、システムはニコニコ動画および YouTube では、動画プレイヤーをコントロールする API を利用しているだけである。そのため、ラッパーを作成するだけで、他の動画共有サイトなどにも適用可能となる。

#### 3.3 操作方法

図 2 の左は、本プロトタイプシステムを導入する前の YouTube 上のオリジナルのインタフェースで、右が本プロトタイプシステムを導入した後の YouTube 上のインタフェースである。

編集可能インタフェースは、音量コントロールグラフとシークバー、スキップ、再生速度、共有という 3 つのボタン、そして「Drop a media file here」という点線の四角形で囲まれた領域からなる。YouTube では動画下部に編集インタフェースが追加され、ニコニコ動画ではシステムの都合上動画上部に編集インタフェースが追加される。

音量コントロールグラフでは、ユーザはグラフ上の点をマウスのドラッグ・アンド・ドロップ操作によって上下させることにより、その再生時間における音量を変更することが可能となっている。また、シークバーは、通常の動画のシークバーのインタフェースと同じく、動画の再生に応じて自動的に動いていくうえ、ユーザの操作によって任意の再生時間に移動することが可能である (図 3)。



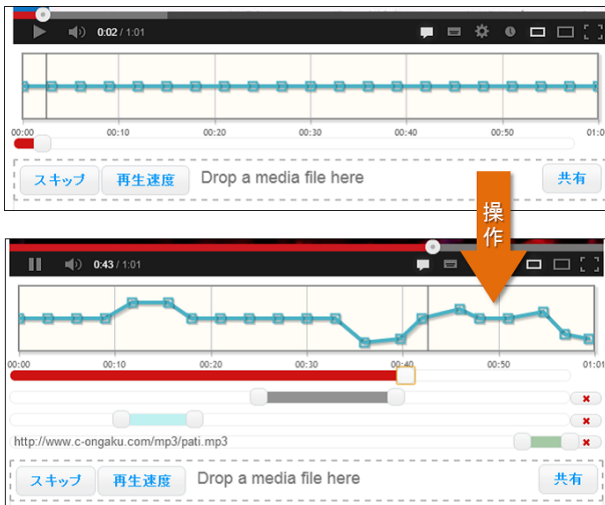


図 3. 編集インターフェースのみを拡大した状態

ユーザがスキップボタンを押すと、編集インターフェースに1つの時間軸が追加され、レンジスライダー（灰色背景の部分）が現在の動画の再生場所を左端に、そこから10秒後を右端にした状態で追加される。このレンジスライダーが指定されている部分が自動的にスキップされる区間となる。ここで、スキップ開始場所はレンジスライダー左端のドラッグ・アンド・ドロップ、スキップ終了場所はレンジスライダー右端のドラッグ・アンド・ドロップによって変更することが可能である。

ユーザが再生速度ボタンを押すと、編集インターフェースに1つの時間軸が追加され、スキップボタンの時と色が違うレンジスライダー（水色背景の部分）が追加される。現在は、この再生速度ボタンによって指定された部分は、0.5倍速で再生される（プロトタイプシステムでは固定）。

「Drop a media file here」という点線で囲まれたスペースに、ユーザがローカルのメディアファイルや、ウェブ上のメディアファイルのURLをドラッグ・アンド・ドロップすると、編集インターフェースに1つの時間軸が追加され、レンジスライダーが追加される。レンジスライダーは、現在の動画の再生場所を左端に、そこからドロップしたファイルの長さ分後を右端にした状態で追加される。ここでは、ユーザがレンジスライダーの左端、右端どちらをドラッグ・アンド・ドロップしても、メディアファイルの開始位置が変更される。また、ドロップされたファイルがどこから取り込まれたのかという情報は、時間軸インターフェース上に合わせて提示される。これにより、リファレンス情報が明確になる。

ボタンの押し下げや、ファイルのドロップなどによる編集内容は、追加された時間軸の右端に登場する×ボタンを押すことにより削除することが可能となっている。また、対象となる動画をユーザがあらかじめ編集している場合は、その編集情報が自動的に

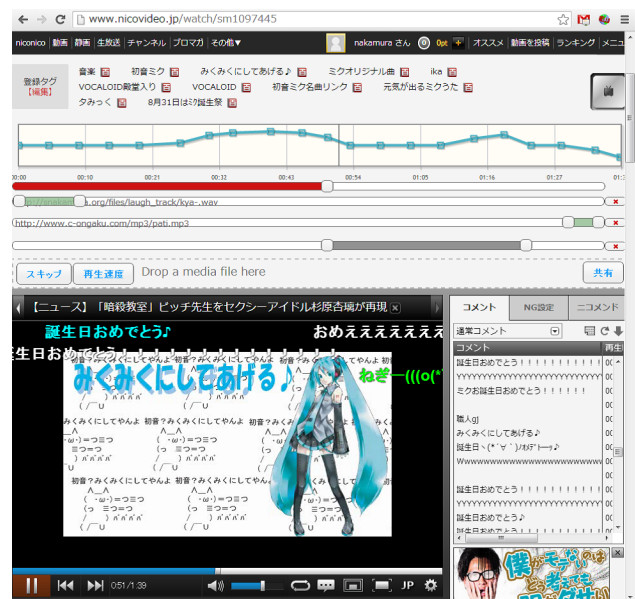


図 4. ニコニコ動画上のインターフェース

にロードされる。

なお、編集情報はローカルに保存されるが、共有ボタンを押すことによって、編集情報のみを他者と共有することが可能となっている。

### 3.4 利用例

図4は、本システムをニコニコ動画上で利用している様子である。ここでは、動画の最初の部分に大歓声を入れ、通常再生しサビ部分に向けて音量を徐々に上げている。また、1つ目のサビ終了後、2つ目のサビは飛ばしてすぐに音楽の終盤に飛び、音楽の最後では音量を下げるように指定されている。さらに、動画の最終盤に拍手と歓声が再生されるようになっている。

図5は、本システムをYouTube上で利用している様子である。動画自体は、ある研究者が20個のテーマを45秒ずつ（トータル15分）で発表している様子である。ユーザがこの発表の動画を他者に紹介しようと思った際に、10分しか紹介に使える時間がなかったため、紹介相手が興味をもってもらえなさそうなテーマについてはスキップするように指定する一方、あつという間に進んでしまっただけに、再生速度を変更しスロー再生することによって時間を長めにとっている。また、発表者の音量が大きかったり小さかったりとぶれがあったため、なるべく一定になるように音量コントロールインターフェースで変更している。さらに、全体的に音声盛り上がり欠けていたため、笑って欲しい所にラフトラックを入れたり、最後に盛大な拍手を入れることによって、盛り上がり演出している。

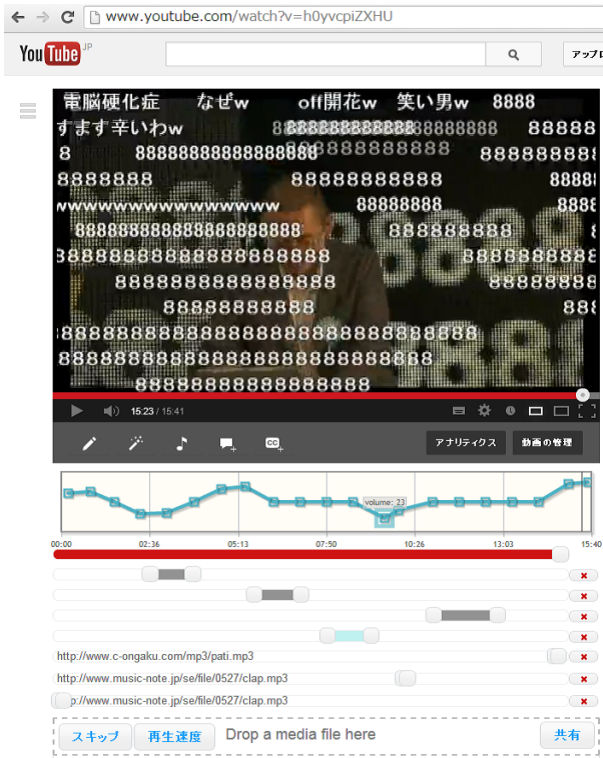


図 5. YouTube 上でのインターフェース

#### 4 考察

本提案手法により、演奏してみた動画の最後に拍手を入れたり、踊ってみた動画の最後に歓声を入れたり、歌ってみた動画の途中途中で合いの手の声を入れたり、科学系の動画の途中途中で解説の説明を音声として入れたり、わかりにくい部分はスロー再生または停止して説明したりといった事が可能となった。この既にある動画に対する編集（装飾）は、その動画コンテンツを変質させる。例えば、ストーリー性のある動画について、途中途中でラフトラック（笑い声）を挿入すると、より面白く感じるという事が実際のコンテンツ制作より観察された。今後は、その有効性をユーザ実験などによって明らかにしていく予定である。

動画の再生時間が長い場合に、スキップや挿入した音声などがどこを指しているのかがわかりにくく操作しづらいという問題があった。この問題については、動画編集ツールにありがちな時間軸を拡大縮小するようなインターフェースを導入することである程度解決可能であると考えられる。

今回提案している手法は、様々な所から集めてきたリソースを1つの動画としてコンテンツ内に取り込むのではなく、部品として外部に存在させつつも1つの動画であるかのように振る舞わせることができるということである。つまり、どの音声、どの音楽がどこから来たのかというリファレンス情報が明確に存在しているということであり、そのリファ

ンス情報を元にオリジナルコンテンツへのアクセスも可能にするものである。ウェブ上の動画などで、リソースの取得元を明記していないケースも多いというえ、仮に明記されていたとしてもアクセスは容易ではない。そういった点で、オリジナルの作成者も尊重しつつも、リファレンスしながら新たなコンテンツを生み出すことができる事が可能となっている。

これまで、ニコニコ動画など動画共有サイトでは、オリジナル動画の作成者と、N次創作をする動画の作成者、動画の視聴者という3つのタイプの利用者が存在していた。ここで、オリジナル動画の作成者やN次創作をする動画の作成者に比べ、動画の視聴者の数の方が圧倒的に多い。我々の手法は、この圧倒的に数が多い動画の視聴者自体に、編集という形で動画に積極的に関与することを可能にするものであり、動画コンテンツを中心とした生態系がより成長するのではないかと期待される。

一方、本提案手法はオンライン上での編集および視聴を可能とするものであるため、公式サイトから配信されているような動画コンテンツも、本手法を使うことによって擬似的に編集し、視聴することが可能となる。例えば、公式のチャンネルとして、アニメなどが配信されているものについて、オープニングやエンディングを飛ばしたり、ダイジェストシーンのみにするよう編集しておき、ダイジェストのみ視聴といったことも可能となる。また、そうした公式の動画に対して、音声を付与したり、別の音楽に差し替えたりといった事も可能となる。また、本手法ではある公式動画の映像と、ある公式動画の音声の組み合わせを動画を違法にダウンロードして編集するのではなく、同期的に再生することで組み合わせとして楽しむことが可能となる。このような動画の使用は、コンテンツの世界を広げるものと期待される。

また、本提案手法を利用することによって、ユーザが好きなアーティストの楽曲をCDから取り込んでmp3化し、ウェブ上のお気に入りの動画の映像に合わせて再生する（その動画の音声自体は消音にする）ということも可能となる。ここでは、ユーザはそうしたmp3ファイルをウェブ上にアップロードして他者と共有するわけではなく、個人的に楽しむだけであるため、著作権の侵害にも当たらず、楽しむことが可能となる。

現在は、YouTubeとニコニコ動画のみに対応しているが、システムの処理としてはYouTubeやニコニコ動画の動画コントロールのためのAPIを利用しているだけである。そのため、そうしたAPIさえ存在し、ラッパーとなるプログラムを書けば、他の動画サービスでも本手法を適用可能となる。例えば、Ustreamなどに保存されている、WISSの登壇発表などに対して、説明の音声を付与したり、発表と無関係なシーンを飛ばしたりといったことが動画

を編集すること無く実現可能となる。

## 5 関連研究

Seek Rope[4] は、YouTube などの動画共有サイトにおいて、シークバーをロープのように取り出し、ロープを丸めることで繰り返し再生したり、ロープを繋げることで複数の動画をつなげたり、ロープを格子状に配置することでランダム再生を行ったりすることを可能とするシステムである。ウェブ上のコンテンツを活用するという点で類似性があるが、本研究は再生のコントロールのみならず、音声や音楽、他動画などによって装飾可能であるという点で違いがある。

ウェブ上のコンテンツを組み合わせる新たなコンテンツを作り出す試みとして Massh![5] がある。Massh! では、楽曲同士をインタフェース上でつなぐことによってマッシュアップ音楽を作成可能としている。ウェブ上のコンテンツを組み合わせ、利用するという点で類似はあるが、我々の手法は個人的な欲求を満たすというものであり、その表現方法や目的は異なる。

MixBit[7] は、短い動画クリップ (16 秒) を組み合わせることで、新たな動画を作成可能とするシステムである。MixBit では、ユーザ自身が動画を撮影せずとも他者が撮影した動画を活用して動画制作が可能となっており、動画制作のハードルを下げている。ただ、MixBit は閉じたコミュニティの中の動画を再生時間に沿って順に繋ぎ合わせるものである。我々の手法は、コミュニティに閉じることを無くすすべてのオンラインコンテンツを対象としており、また映像と音声などの組み合わせも可能としている。

ラジへえ[8] は、ラジオを聞きながらその音声に対して音声コメントをボタン1つで入れる事を可能にするものであり、コンテンツの他者との楽しみを強調するものである。本研究で提案する手法の一部は、ラジへえの音声付与に似ている。本提案手法は、編集を目的としており、ラジへえのようなリアルタイムでのコミュニケーションは対象としていない。

SMIL[6] はウェブ上でのマルチメディアコンテンツの振る舞いを記述するマークアップ言語である。SMIL では、同期的にマルチメディアコンテンツを再生するといった記述も可能であり、異なるフォーマットの動画ファイルを並べて再生することなども可能である。我々の手法は、SMIL の仕様とは異なり、ブラウザ自体がウェブ上のリソースおよびプレイヤーを直接操作する事が可能となっているため、多様な表現が可能である。ただ、SMIL の記述スタイルは参考になる部分が多いため、今後編集情報の共有などにおいて参考にする予定である。

## 6 まとめ

本研究では、ウェブ上の動画に対して、ブラウザ上で音量コントロールや、スキップ操作、音声や音楽の付与といった編集を可能とし、その編集情報に基づき視聴可能な仕組みを実現した。また、編集情報を他者と共有可能な仕組みを実現し、他者の編集を参考に視聴したり、他者の編集を利用して再編集したりといった事を可能とした。

今後は、本プロトタイプシステムをウェブ上で公開し、多くのユーザに利用してもらうことでフィードバックを集める予定である。また、現在の操作インタフェースおよび機能は、まだ十分に練られていないため、ユーザからのフィードバックなどを元に改良を行っていく予定である。

ユーザによる編集情報と、ニコニコ動画のコメントにはある程度の相関がある可能性がある。そこで、ユーザの編集情報と、ニコニコ動画のコメントを相互分析することによって、自動編集を可能とする仕組みも実現する予定である。

## 参考文献

- [1] 中村聡史, 山本岳洋, 後藤真孝, 濱崎 雅弘: 視聴者反応と音楽的特徴量に基づくサムネイル動画の生成手法, 情報処理学会論文誌 (トランザクション) データベース (TOD58), Vol.6, No.3, pp.148-158 (2013-06-28).
- [2] 濱崎雅弘, 武田英明, 西村拓一: 動画共有サイトにおける大規模な協調的創造活動の創発のネットワーク分析-ニコニコ動画における初音ミク動画コミュニティを対象として-, 人工知能学会論文誌, Vol. 25, No. 1, pp. 157-167 (2010).
- [3] 濱野智史: アーキテクチャの生態系 - 情報環境はいかに設計されてきたか, NTT 出版 (2008) .
- [4] 佐藤剛, 宮下芳明: SeekRope: 曲げて切って結べるシークバー, インタラクシオン 2010 論文集, pp. 197-200 (2010).
- [5] 徳井直生: Massh!, <http://www.sonosphere.com/mash/> (2008).
- [6] W3C: Synchronized Multimedia, <http://www.w3.org/AudioVideo/>.
- [7] mixbit: <https://mixbit.com>.
- [8] 加藤由訓, 苗村健: ラジへえ: ラジオ聴取時における感想共有システム, インタラクシオン 2013, pp.32-39 (2013).