

平成 2 2 年度 卒業論文

XML を利用したインターネット上の動画編集システム
に関する研究

指導教員
舟橋 健司 准教授
山本 大介 助教

名古屋工業大学 工学部 情報工学科
平成 16 年度入学 16115076 番

名前 佐野 直樹

目次

第 1 章	はじめに	1
第 2 章	従来の動画共有サービス	3
2.1	YouTube	3
2.2	ニコニコ動画	5
2.3	Ustream	7
2.4	YouTubeREMIXER	9
2.5	利点と欠点のまとめ	10
第 3 章	問題点	11
第 4 章	XML による動画編集システム	12
4.1	XML とは	12
4.2	XML による動画記述手法	14
4.3	XML による編集手法	16
4.3.1	タイトルの付加	17
4.3.2	素材動画の連結	18
4.3.3	素材動画の切り取り	18
4.3.4	連結部分のフェード	19
第 5 章	動画編集システムの実現法	23
5.1	システム構成	23
5.2	OpenCV での動画編集	25
5.2.1	XML の読み込み	26
5.2.2	素材動画の検索、読み込み	26
5.2.3	タイトルの作成	26
5.2.4	素材動画を編集し繋げ、新しい動画を作成	27
5.3	WPF で作成した UI で XML を作成	28
5.3.1	編集用 XML の下地の作成	30
5.3.2	タイトル情報の入力	31
5.3.3	素材動画を繋げる	32
5.3.4	素材動画の参照とプレビュー	33
5.3.5	フェード機能を付加	34
5.3.6	作成した XML をサーバーに送信	35

第 6 章 実験と考察	36
6.1 実験	36
6.1.1 実験環境	36
6.1.2 編集内容	38
6.1.3 ユーザーインターフェースを使用し編集用 XML を作成	39
6.1.4 編集用 XML を送信し、動画編集プログラムで動画を作成	41
6.2 考察	41
第 7 章 おわりに	42
謝辞	43
参考文献	44

第1章 はじめに

近年、インターネットの普及により映像や音楽のコンテンツがインターネット上で閲覧可能になった。中でも世界的に有名な YouTube、日本で人気のあるニコニコ動画をはじめとする、動画投稿サイトの発展と普及は目を見張るいきおいである。携帯電話や携帯ゲーム機や携帯音楽プレーヤーだけでなく、薄型テレビや HDD レコーダーといった情報家電も最近では動画投稿サイトにアクセスできる機能を備えたものが増えている。動画投稿サイトとライバル関係にある既存メディアのテレビ放送でも動画投稿サイトと連動した番組が散見され、書店では動画投稿サイトの使い方を解説したガイド本があふれている。動画投稿サイトでは、様々な動画が投稿されていて無料で観賞することができる。マスメディアによって制作されたコンテンツも閲覧が可能だが、現在では一般人の手によって制作されたコンテンツが爆発的な勢いで増加している。携帯電話等にもカメラ機能がつき、動画編集ソフトもパソコンに初めからインストールされていることが多い。手軽に撮影し編集を加えて動画を投稿することが、映像に詳しくない一般ユーザーにも出来るようになった。

しかし、そういった手法で制作された動画が投稿サイトにはあふれかえっているため、一定以上の人に見てもらえるような動画を制作するのは難しい。さらに、動画を制作する際に使用する素材となる動画集めもハードルが高い。各自が用意できるカメラで撮影した素材を集め、それらを編集したとしても、個人レベル以上の作品を作り出すことは非常に困難である。そのうえ、最近ではインターネット上における著作権の侵害に対する規制も見直されるようになってきた。素材となる動画を動画投稿サイトの中で探したとしても、ダウンロードに対する規制が多くあり、自分のパソコンに保存することは困難である。投稿動画の中にはマスメディアの制作した作品を編集しなおして、新しい動画として投稿される M A D と呼ばれる動画も多い。こういった動画の多くは無断で作品を使用して投稿されているため、権利者の申し立てにより削除されることもある。しかし、動画の一部分に削除対象の作品が使用されていたとしても、動画をすべて再生しない限り作品が使用されているのか判断が出来ない。動画投稿サイトの中には 4 0 0 0 万を超える動画数が投稿されているサイトもある現状で、すべての投稿作品をチェックすることは不可能である。

問題点が多く残る一方で、動画投稿サイトの中には独自の技術を導入しているサイトも増えてきている。従来のコンピュータ利用は、ユーザーがコンピュータのハードウェア、ソフトウェア、データなどを自分自身で保有、管理していた。しかし、クラウドコンピューティングという考え方では、ネットワークを介してユーザーが命令を与え、サーバー側のマシン上で処理を行う。ユーザーが用意すべきものは最低限の接続環境だけであり、蓄積するデータの管理などが軽減される。現在の動画投稿サイトにおけるクラウドコンピューティングといえば、動画をサーバーにアップロードしてブラウザ上で動画が

再生できるという今では当たり前の機能のことである。ただしそれだけではなく、インターネット上で動画が編集できる機能を付加した動画投稿サイトも少ないながら存在している。それらのサイトでは自らが集めてきた素材となる動画をアップロードし、ブラウザ上からの操作によってサーバー上で動画の編集を行う技術が使われている。動画の編集にはある程度以上のマシンの性能が要求されるため、処理の遅いコンピュータでは多大な時間を浪費してしまうことがあるけれども、サーバー上での編集ならば自身のマシンの性能に依存することはない。そして、動画を編集する際に使用する素材となる動画は、ローカルのマシンに残す必要がない。以上の点から優秀な機能だと言える。しかし、欠点も少なからずあり、動画を編集するシステムは処理量が多くなるため、大勢のユーザーが利用した時にサーバーにかなりの負荷がかかる。そのため、編集機能自体を軽く簡単で複雑な処理をしないものにしなければならない。そして、素材となる動画も、まずはローカルのマシンにユーザー自身が集めてこなければならず、大量に使用する場合はユーザー側の作業も、サーバー側の負担も大きくなる。これらの点は、ユーザーが多いサイトでは致命的な欠点になり、今現在の状況でこの機能が注目されていない理由でもある。

本研究では現在のインターネット上の動画編集機能における欠点を、システム面からの根本的な解決を目標とする。以下、2章では動画投稿サイトの概要について説明する。また3章では、新たに考案したインターネット上の動画編集システムの手法について説明する。4章では実験および結果、5章では本研究のまとめを述べる。

第2章 従来の動画共有サービス

本章では、インターネット上に現在開設されている動画共有サービスの概要についての説明する。まず、動画共有サービスについて、「日経パソコン用語辞典 2011」の説明を紹介する。

動画共有サービス

動画を投稿したり、ほかのユーザーが投稿した動画を閲覧したりするなどのサービスを提供する Web サイト。ビデオカメラで撮影した動画などを、インターネット上で複数の人に公開できる。動画投稿サイトとも呼ぶ。YouTube やニコニコ動画、Ustream などが代表的。動画共有サイトでは、ユーザーが私的に録画したテレビ番組映像などが投稿されることが多く、著作権侵害の問題が生じやすい。運営会社各社は、アップロードされた動画が違法動画かどうかを自動識別するツールを使ったり、権利者向けに投稿動画のチェックと削除依頼ができるツールを配布したりすることで対処している。

この説明にある、「YouTube」「ニコニコ動画」「Ustream」、そして動画編集がインターネット上で出来る「YouTubeREMIKXER」について概要を以下で説明する。

2.1 YouTube

まずは、YouTube について「日経パソコン用語辞典 2011」の説明を紹介する。

YouTube

米ユーチューブが 2005 年 12 月に開始した動画共有サービス。ユニークな動画が多く投稿されていることから利用者が急増し、類似の後発サイトも続々と登場した。2006 年 10 月には米グーグルがユーチューブを買収、傘下に収めた。ユーザーはサイトに登録さえすれば、デジカメや携帯電話、DV カメラなどで撮影した 100MB、10 分以内の動画 (AVI/MOV/MPG 形式など) を何本でも投稿できる。動画は Flash Video 形式に変換されて、ブラウザ上で再生できるようになる。動画には、タイトルのほか、検索用のキーワード (タグ) も付けられる。動画を見た人がコメントを付けたり、その動画をブログに貼り付けたりもできる。なお、著作権を侵害する動画の投稿が問題視されている。

ウェブサイトは多言語で構成されており、サービスはすべて無料で利用できる。初期の段階で、動画ファイルを無制限に無料でアップロードできるという仕組みだったため注目を集めるが、同時に大量のアダルト動画がアップロードされることになり、問題視されるようになった。人的リソースの問題から厳密な処理ができず削除対応がゆるや

かであり、それが視聴者を増やす要因となった。世界的に有名なサイトになり、手軽に動画が楽しめることからコンテンツ業界にも注目されている。YouTube でいろいろと実験的なサービスをリリースしている「TestTube」上にユーザー自身がアップロードした動画を編集することが出来る「YouTubeREMIXER」という機能が追加されたが、この機能についての詳しい説明は後述する。

アップロードできる動画に関して説明する。一般ユーザーは容量 100MB、長さ 10 分以内の動画を無制限にアップロード、投稿することが出来る。アップロードできるのは動画ファイルのみで、音声ファイルなどはそのままではアップロードできない。会員登録が実装されていて、会員になると容量 2GB、長さ 15 分 29 秒までの動画をアップロード、投稿できる。会員になれば、特定のメンバー同士で動画を共有することもできる。動画やメンバーにコメントを付けることも会員ならば可能。プレイヤーの右下のメニューによって画質が選択できるが、アップロードされた動画の画質によって選択できる画質は限られる。また、キューに追加することで動画をまとめたり、自動再生（リピート）や、シャッフル再生をすることができる。動画やメンバーにコメントも付けることができる。UTF-8 を採用しているため日本語も含めた多国語を利用できる。

アニメやドラマ、映画等の著作権の対策として、10 分をこえるファイルのアップロードを原則制限しているが、動画を分割してアップロードされることも多くある。しかし、YouTube では簡単に削除されるため、ユーザーは他の動画共有サービスに違法アップロードすることが増えている。音楽に関しては、既存の音楽を自身で歌ったり、演奏したりしたものをアップロードするユーザーが増えたため、YouTube が JASRAC に使用料を払い、JASRAC の管理楽曲を二次利用した動画のアップロード行為を合法とした。



図 2.1: YouTube

2.2 ニコニコ動画

次に、ニコニコ動画について「日経パソコン用語辞典 2011」の説明を紹介する。

ニコニコ動画

ニワンゴが運営する動画共有サービスの関連サービス。動画の再生中にユーザーがコメントを投稿し、ほかのユーザーの閲覧時にそのコメントを表示させるという機能がある。そもそもは、ほかの動画共有サービスにアップロードされた動画をニコニコ動画のサイトで表示し、それに対してコメントを投稿、表示するサービスとしてスタートした。現在は、専用の動画共有サービス SMILEVIDEO が用意されており、このサイトの機能を含めて、動画共有サービスの一つと見なされる場合が多い。

YouTube とは違い、会員登録制のサイトである。一般会員登録者数が約 1890 万人、有料会員は約 101 万人、モバイル会員は 567 万人 (2010 年 10 月 31 日現在)。ただし、アカウントはメールアドレスごとに発行されるため、1 人で複数の一般アカウントを持つことは容易である。サービス開始から 1 年に満たない 2007 年 11 月の時点で、すでに日本全体のトラフィックの約 12 分の 1 を占めていたといわれており、その急成長ぶりが窺える。2008 年 7 月までに台湾 (中国語)・スペイン語・ドイツ語版が作成された一方で、トラフィックの急増が懸念されるために英語版は未だに作成されていない。ニコニコ動画の特徴は、動画配信サイトで配信されている動画の特定の再生時間上にユーザーがコメントを投稿し表示できるコメント機能であり、その他にもユーザーやアップロード者同士が交流できる機能を数多く備えている。趣向を凝らして制作された動画が高い人気を得るなど、独特のコメントシステムもあいまって、独自の文化を築いている。

動画に関して説明する。投稿された動画を独自のコメント表示・投稿機能を通して閲覧させるというものである。閲覧者は、再生中の動画に対して再生画面上にコメントを書き込むことができる。コメントは、現在再生しているタイミング (時間軸) に対して投稿することができ、それ以降に動画が再生された際は、そのタイミングから画面の右から左に 3 秒間横切る形で表示される。その結果、閲覧者はチャットや掲示板のような時系列とは異なる「実時間を超越した擬似的な時間共有」(運営側はこれを「非同期ライブ」と名づけている) を体感することができる。コメントに時間の概念を導入したことで、ニコニコ動画は従来の動画投稿には無い「利用者同士の一体感」を獲得することに成功している。動画上のコメントは、サービス開始以来に書き込まれた全てのコメントが保存されており、動画の削除や時間の経過、件数によって消えることはない。ただし、一度に表示される件数は動画の長さによって制限されており、それを超えると、古いコメントから順に表示されなくなる。各動画には説明文のほか、タグと呼ばれる動画の内容を指し示す検索用キーワードを 10 個まで登録することができる。タグの導入により似たような動画を容易に探せるような仕組みになっている。タグは「ニコニコ大百科」と連動しており、そのタグに関する解説や経緯、そのタグをつけられた代表的な動画などを知ることができる。ニコニコ動画では動画投稿者だけではなく閲覧者も自由にタグを登録することができるのが特色である。本来は検索機能として用いられるタグだが、動画の内容に絡めたタグ付けやニコニコ特有のタグ付け (「才能の無駄遣い」や「孔明の畏」など) も多く見られ、独自の異彩を放っている。

ニコニコ動画にアップロードされているアニメ、ミュージカルライブ、テレビCM、テレビ番組などは、公式チャンネルをのぞくと、ほとんどの動画が著作権者に無断でアップロードされたものである。利用規約では著作権侵害になるファイルのアップロードを禁止しているが、著作権法違反の投稿は後を絶たないのが現状である。権利者（個人、法人、法定代理人含む）から権利を侵害されているとの申し立てがあれば、審査し削除するとしている。その他肖像権などの権利侵害もこれに準ずる姿勢を取っている。性的描写を含むアダルトコンテンツは閲覧者の判断で報告できるが、こと著作権に関しては権利者からの申し立てのみ受け付けているため、アダルトコンテンツほど迅速な対応はされていない。さらに、ときにはコメントやタグを使って、動画の内容に対する罵倒、他者に対する誹謗中傷や無意味な「荒らし」行為、および特定の国家・民族・宗教を攻撃する人種差別的な行為なども見られる。また、同一内容のコメントを同一時間に大量に投稿することで映像部分が見えなくする行為もある。このような荒らし行為の発生は、動画の上にコメントを無制限に書き込めるシステムであるが故につきまとう問題である。現在のニコニコ動画はアカウント制をとっているが、動画へのコメントは匿名で投稿することができる。ただし、匿名性があるとはいえ、投稿者のIDは投稿されたコメントと同時に保存されているため、完全な匿名ではない（当初はID番号がそのまま保存されていたが、2007年8月23日より毎週変化する暗号化された文字の羅列に変更されている）。現在では、後述の「NG設定」にこのIDを指定することで、そのIDで書き込まれたコメントをすべて非表示にすることが可能となっている。



図 2.2: ニコニコ動画

2.3 Ustream

次に、Ustream について「日経パソコン用語辞典 2011」の説明を紹介する。

Ustream

米ユーストリームが運営する動画配信サイト。2007年3月にサービスを開始した。映像や音声をその場で撮影し、リアルタイムでストリーミング配信できるのが特徴。インターネットに接続できる環境とデジタルカメラがあれば、誰でもコンテンツの送り手になれる。iPhoneからの投稿が多いという。ミニブログサイト Twitter と連携した使い方で人気を高めた。YouTube 同様、サイト上に保存してある動画の視聴も可能。ソフトバンクは2010年1月、ユーストリームに2000万ドル(約18億円)を出資した。出資比率は13.7%。2011年7月まで株式を追加取得できるオプションも有しており、すべて行使した場合、出資比率30%強の筆頭株主になる。

YouTubeのような単なる動画配信ではなく、生中継の用途に強みを持つ動画共有サービス。USB接続のカメラとパソコン、インターネットに接続できる環境があれば、誰もがどこからでも生中継が可能。パソコンでなくても、専用アプリケーションをインストールした iPhone や Android 端末といったスマートフォンからでも生中継ができる点が Ustream の特徴である。多くの人が Ustream を知るきっかけとなったのは、2009年1月に米国において、オバマ大統領の就任式典の生中継で利用されたことである。それから爆発的に利用が増え、2010年4月時点での1カ月間のサイト来訪者数は1億1000万人を超えた。Ustream は機能強化を続けていて、なかでも Twitter との連携機能は注目度向上につながっている。ユーザーは生中継の画面の横に用意された Twitter の投稿欄を使うことで、同じ生中継を見ているユーザーと情報を共有できる。

動画に関して説明する。なにより特徴的なのは撮影と配信が同時にできるライブ方式を採用した点である。ニコニコ動画にもニコニコ生放送というライブ放送ができる機能があるが、違いは時間制限がないことと、世界的に広く同時に使われていることである。そして、ライブ配信したその映像の録画も可能である。録画データは Ustream のサーバー上に保管されるため、利用者のパソコンに保存する必要がなくなる。利用方法も簡単で、Webカメラとマイクを接続したインターネット利用可能なパソコン、もしくは専用アプリケーションをインストールしたスマートフォンを用意するだけでいい。リアルタイムを特徴としているため、コメントも Twitter を利用した形式になっており、Twitter 利用者がそのまま Ustream を利用するケースが多い。

生中継であるがゆえの問題点は多くある。一方公共の場でライブ配信をする際に、通行者、特定企業や団体ロゴの映り込みなどは肖像権の問題から極力避けるべきだ。しかし肖像権はプライバシーの侵害やその人を毀損するような行為、または映像を営利目的に使用しなければ大きな問題にはなりにくい。しかし、DJプレイと呼ばれる音楽配信では原曲をそのまま流したり、アニメやドラマ、野球中継等を実況付きのライブで放送したりもしている。これらのような著作権の侵害に触れるかどうかの境目の配信が多くある。そのうえ、アダルトな内容を含んだ生放送が可能という危険性がある。手軽に撮影した映像を生放送で世界に発信できるサイトであるにも関わらず、そういった危険性に対する対応が少ない。

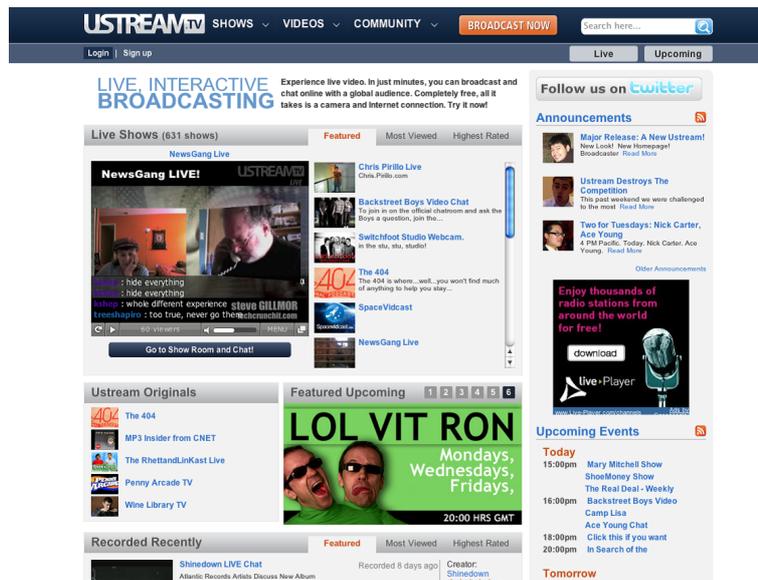


図 2.3: Ustream

2.4 YouTubeREMIXER

次に、オンラインビデオ編集ツールの YouTubeREMIXER について説明する。

YouTube でいろいろと実験的なサービスをリリースしている「TestTube」上にて、Photoshop など有名な Adobe が制作したオンラインビデオ編集ツール。機能として備わっているのは、主に以下の4つである

- 動画の一部の抜き出し
- 動画の連結（フェードイン、フェードアウト付き）
- テロップの挿入
- フレームなどの装飾品を付ける

ユーザー自身が動画の編集に使用する素材となる動画をアップロードする。編集はブラウザ上の操作のみですべて可能となっている。編集には、自身がアップロードした動画しか使用することは出来ない。操作はドラッグ&ドロップが基本で複雑な操作がないため、簡単な編集しか出来ないが、動画の編集になれていないユーザーでも使いやすい機能になっている。携帯電話から動画を撮影してアップロード後、Flash に対応したフルブラウザから携帯電話経由で編集すれば、すべてモバイルで完結するという使い方を念頭に置いているように思われる。さらに、YouTube に投稿することを前提とした編集のため、動画のファイル形式やビットレート、画質等の制限を考慮せずに編集しても問題がない。



図 2.4: YouTubeREMIXER

2.5 利点と欠点のまとめ

今までに例にあげた、動画共有サービスの「YouTube」「ニコニコ動画」「Ustream」、動画編集がインターネット上で出来る「YouTubeREMIXER」についての利点と欠点をまとめ、そして動画共有サービス全般に言える欠点を以下で述べる。

- YouTube

利点 会員になれば、特定のメンバー同士で動画を共有できる

利点 多国籍の利用が可能

欠点 検索機能が弱い

- ニコニコ動画

利点 再生中の動画に対して再生画面上にコメントを書き込むことができる

利点 タグの導入により似たような動画を容易に探せる

欠点 権利者からの申し立てのみで削除が決まる

欠点 コメントやタグを使った動画の内容に対する罵倒等の「荒らし」が発生しやすい

- Ustream

利点 映像や音声をその場で撮影し、リアルタイムでストリーミング配信できる

利点 ライブ配信したその映像の録画も可能である

欠点 ライブ配信での映り込み等の肖像権の問題

欠点 放送禁止となる内容を含んだ生放送が可能で、対策が不十分

- YouTubeREMIXER

利点 編集がブラウザ上の操作のみですべて可能

利点 動画の編集になれていないユーザーでも使いやすい機能

欠点 自身がアップロードした動画しか編集できない

- 全般

欠点 著作権の侵害について対策が不十分

欠点 シーンごとの検索が出来ない

これらの利点をなるべく残し、欠点がある程度克服したシステムを次の3章で説明する。

第3章 問題点

本章では、既存の動画共有サービスの問題点を説明する。

既存の動画共有サービスのシステムの多くでは映像の編集がサポートされていない。その理由は、主に映像の編集にかかる処理の多さに関係している。少数ではあるが、Web上で映像の編集を行うことが出来る動画共有サービスも存在する。しかし、編集に利用できる素材となる動画はユーザー自身がアップロードしたものに限られ、そのアップロードする動画にも容量の制限がかかっている。本格的な動画編集をWeb上で編集のすべてを行うことはこれらの点から難しくなっている。そのため、ローカルなPCにインストールされた編集ソフトを用いて編集作業を行い、編集された後の完成品として投稿されているのが現状である。投稿された動画はすべて完成品だったため、動画を作成する際に使用した素材の動画についての情報を得ることが難しい。タイトルやコメント、検索用タグ等が、わずかな情報として与えられているだけである。今までのシステムでは、動画の内容に関する情報は、ユーザーや投稿者により主導で与えられていた。そのため、動画の内容や、シーンに対する検索が困難であった。

そこで本研究ではXMLを用いて動画のメタデータを記述する方式を提案する。動画のメタデータとは、既存の動画共有サービスでメタデータとして扱われている、タイトルや動画の尺等の動画そのものの情報と、コメントや検索用タグ等の動画に付加された情報、それに加え、Web上で映像の編集をおこなうその編集履歴をメタデータとしてXMLに記述する。XMLについて後の4章で詳しく説明するが、XMLはWebサービスにおいて標準的なデータ記述方式の一つであるため、Webとの親和性が高い。そのため、動画をWebで扱うことが容易になる。既存のシステムでは動画に使用された素材の動画を判別することが難しい。Web上でXMLにより映像の編集が可能となるシステムならば、動画編集に使用した素材動画の種類や数、使用した長さまでXMLに記録できる。完成後の動画の編集の過程が記録されているため、素材動画の情報をを用いた検索が可能となる。素材動画の情報をを用いた検索とは、ある動画に使用されている動画の元を探したり、ある動画が他の動画の素材として利用されているかを調べたり、動画の中である動画を使用して作成された部分だけを検索すること等が可能になる。

次章で、XMLによる動画編集システムの詳しい設計について説明する。

第4章 XMLによる動画編集システム

本章では、XMLによる動画編集システムについて説明する。

本システムのかなめとなる機能は、動画の編集の内容を記述したデータを残すことである。編集の過程を残すことにより、シーンごとの検索や、動画の再編集が容易になる。編集のデータを分かりやすく、かつ、サーバーにかかる負担が少ない形で保持する必要があった。そこで、本システムでは、近年ゲーム開発等によく用いられているXMLと呼ばれるデータ形式を利用した。

4.1 XMLとは

まずはじめに、「IT用語辞典 e-Words」より、XMLの説明を紹介する。

XML

文書やデータの意味や構造を記述するためのマークアップ言語の一つ。マークアップ言語とは、「タグ」と呼ばれる特定の文字列で地の文に情報の意味や構造、装飾などを埋め込んでいく言語のことで、XMLはユーザが独自のタグを指定できることから、マークアップ言語を作成するためのメタ言語とも言われる。XMLにより統一的な記法を用いながら独自の意味や構造を持ったマークアップ言語を作成することができるため、ソフトウェア間の通信・情報交換に用いるデータ形式や、様々な種類のデータを保存するためのファイルフォーマットなどの定義に使われている。XMLを様々な場面で利用しやすいよう、関連技術の規格も数多く存在する。文書を表示する際の書式や装飾などを指定するXSL、ハイパーリンク機能を実現するXLink/XPointer、XMLベースの言語の仕様を記述するためのスキーマ言語であるXML SchemaやRELAX、XMLをプログラムで利用するためのAPIであるDOMやSAXなどである。

マークアップ言語には、XMLの他に、有名なHTMLなどがある。XMLとHTMLの大きな違いは、タグと呼ばれる<と>で囲まれた情報を、自分で設定することができる点にある。HTMLがブラウザでの表示に特化している構造をもつのに対し、XMLは統一的な記法を用いながら独自の意味や構造を持ったマークアップ言語を作成することができる。そのため、XMLには高い拡張性と自由度が備わっている。さらに、フォーマットが世界共通で利用できるため、インターネット上でのデータの格納に適している。使用例を下に示す。

```
<車>  
  <名前 分類="乗用車">カローラ</名前>  
  <色>青</色>  
  <国>日本</国>  
</車>
```

この例では<車>というカテゴリの中に、<名前><色><国>という3つの情報が格納されている。このとき、<車>のタグを親、<名前><色><国>を子という。さらに<名前>のタグの中に"分類"の情報が付加されている。この"分類"にあたる部分を、アトリビュート(属性)という。例を見てもわかるように、非常にシンプルな階層構造をしていて、開発者が見たときにも、理解しやすくなっているのがXMLの特徴である。

4.2 XMLによる動画記述手法

本システムでは動画の編集にXMLを用いる。3章で述べたように、XMLに動画の情報を記述することで、動画をWeb上で扱うことが容易になり、編集にXMLを使用することで、XMLに記述された編集履歴から、動画の部分検索等に利用することが可能になるためである。したがって、動画編集に入る前に、まずはそれぞれの動画の情報をXMLに格納しなければならない。格納する情報は、2章で関連研究にあげた3つの動画投稿サイトを参考にし、動画のメタデータと呼ばれる情報のなかで、投稿者やユーザーが自由に設定できる情報だけでなく、システムティックに決まる動画の情報を多く採用した。このメタデータの記述されたXMLをメタデータXMLと呼ぶ。

動画のメタデータXMLの構造を説明する。動画の情報として格納するのは以下の8つである。

- id
- title
- address
- size
- length
- framerate
- category
- face

この8つのうち「face」を除く7つのデータは、参考にした動画投稿サイトでは基本のメタデータとして保存され、検索等にも使用される一般的なデータである。「id」はタイトルとは関係なく動画に割り振られた番号、「title」は動画の題名、「address」は動画の保管先へのアクセス、「size」は動画の容量、「length」は動画の総フレーム数、「framerate」は動画のフレームレートである。これらは動画のメタデータの中で投稿者やユーザーではなくシステムティックに決められる情報である。「category」は動画の内容の分類で、システムティックには決まらないものだが、検索機能の充実のために情報を格納してある。「face」は今後付け加える予定の、顔にモザイクやフィルターをかける機能の情報を格納する。

動画のメタデータ XML には投稿されたすべての動画の情報が格納されている。すべての動画の情報は < allmovie > タグに格納されていて、それぞれの動画の情報は < allmovie > の子である < movie > タグに格納される。 < movie > タグには属性に "id" の番号が格納されていて、下図 3.1 の例では、はじめに「00000001」の id を持つ動画の情報、次に「00000034」の id を持つ動画の情報が格納されている。下図 3.1 の例では "id" が「00000001」の動画の子の情報に、 < address > が「C:\Users\sano\Desktop\sano\OpenCV\LaughingMan\LaughingMan\sample.avi」、 < title > が「サンプル動画」、 < size > が「20MB」、 < length > が「2000」、 < framerate > が「30」、 < category > が「動物」、 < face > が「on」と格納されている。他の動画に関しても、同じようにそれぞれの情報が格納されている。

```
<allmovie>
  <movie id="00000001">
    <address>C:\Users\sano\Desktop\sano
      \OpenCV\LaughingMan
      \LaughingMan\sample.avi
    </address>
    <title>サンプル動画</title>
    <size>20MB</size>
    <length>2000</length>
    <framerate>30</framerate>
    <category>動物</category>
    <face>on</face>
  </movie>
  <movie id="00000034">
    <address>C:\Users\sano\Desktop
      \sano\OpenCV\LaughingMan
      \LaughingMan\sozai034.avi
    </address>
    <title>指34</title>
    <size>12.5MB</size>
    <length>270</length>
    <framerate>30</framerate>
    <category>人</category>
    <face>on</face>
  </movie>
  :
</allmovie>
```

図 4.1: すべての動画のメタデータ XML

この階層的な格納には意味がある。動画の編集に用いる XML では、素材となる動画を、address や title ではなく、id で指定する。プログラムで XML を読み込み、その一部を検索する場合には、XML の情報を親から順にたどる。id の情報を < movie > タグの属性に格納しておくことで、素早い検索が可能となる。

4.3 XMLによる編集手法

本研究では Web 上での動画編集を目的としている。動画を編集する際に使用する素材となる動画が、著作権の侵害等の理由からダウンロードに規制がかかったため、ダウンロードによる収集が難しくなった。Web 上で動画の編集を行うことが可能になれば、ダウンロードの必要がなくなる。ダウンロードが著作権等の問題に大きくかかわっているため、ダウンロードせずに Web 上に投稿された動画を使用して動画編集することで、自由な動画制作が可能となる。そして、Web 上にある素材で作成した動画であれば、その編集履歴を残すことにより、分類分けや、使用した素材の検索等の機能の多くに利用することが可能となる。動画の編集はローカルな PC に編集ソフトをインストールして行うことが大半である。しかし、動画の編集の処理は膨大であり、使用する環境によっては非常に使い勝手が悪く、ローカルの PC にハイスペックが要求される。しかし、Web 上で利用できる動画編集システムであれば、ネット環境さえ整っていれば、スペック等に影響することはない。そして、ローカルな環境で利用する動画編集ソフトでは、編集機能等がアップグレードされたとき、新しい商品をインストールしたり、ユーザーが自発的に更新を行わなくてはならない。しかし、動画編集システムを Web 上に構築した場合、更新等でユーザーにかかる負担はない。ただし、Web 上で多くのユーザーが利用することを考慮に入れると、システムの動作の軽量化は必須になってくる。そこで、XML による動画編集システムを提案した。XML が Web 上で動画を扱うのに適していることは以前に説明した。Web 上で XML により映像の編集を行うシステムならば、動画編集に使用した素材動画の id やフレーム数等を、編集履歴として XML に格納することが可能である。完成後の動画の編集履歴が記述されているため、素材動画の情報をを用いた検索が可能となる。本章では、XML による編集手法の説明をする。

動画編集に用いる編集用 XML を提案する。編集用 XML は、ローカルな PC でインストールされている WindowsMovieMaker 等の編集ソフトの、一般的によく利用される編集機能を参考にした。まず、動画の作成には、素材となる動画が必要になる。編集ソフトでは素材となる動画をタイムライン上に並べることで、その順番通りに動画が繋げられる。そこで、編集用 XML でも素材となる動画の id が、動画の時間軸に合わせて順に < movie > タグの属性に格納されている。そして、素材動画にカット等の編集を加えて他の動画と繋げる際には、< movie > タグの子に格納する。この格納方法では、新しく編集機能を付け加えるときに子の要素を増やすことで拡張が可能となる。これらの要件をふまえ、XML を設計した。動画の編集として XML に格納されている機能を以下に示す。

- タイトルの付加
- 素材動画の連結
- 素材動画の切り取り
- 連結部分のフェード

XML に格納されるこれらの編集内容を順を追って説明する。

4.3.1 タイトルの付加

はじめに「タイトル」について説明する。ここで使われるタイトルは、作成する動画のはじめの部分に、黒い画面の上に白い文字でタイトルネームが表示される機能である。多くの場合、完成した動画の冒頭にタイトルが付けられているため、この機能を付けた。ファイル名や動画の ID と同じでなければいけないといった制約はない。格納される情報は、表示される「タイトルネーム」と表示する「フレーム数」の2つである。下図の XML のように < titleplus > のタグに「sample」というタイトルネームが格納されている。そして属性の”length”に「200」というフレーム数が格納されている。この XML の記述により編集機能として下図のようなタイトルが動画の冒頭に付けられる。

```
<titleplus length="200">sample</titleplus>
```

図 4.2: タイトルを付ける XML の例

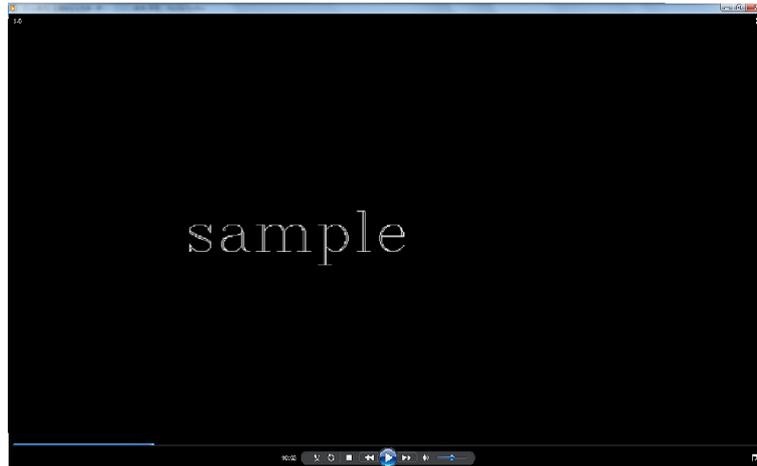


図 4.3: タイトルのサンプル

4.3.2 素材動画の連結

次に「素材動画の連結」の説明をする。素材動画とは動画を作成する際に素材として用いる動画のことで、サーバー上の ID がついている動画ならば編集に使用可能である。素材動画を XML で並べて格納することで、動画が上から順に連結して一つの動画になる。素材動画を指定する際には、ファイル名でなく割り振られた ID を用いる。タイムラインと同じように時系列順に並んでいるため、下図では id が「00000001」と「00000002」の動画が上から順に連結される。

```
<movie id="00000001"></movie>
<movie id="00000002"></movie>
```

図 4.4: 素材動画の連結の XML

4.3.3 素材動画の切り取り

次に「素材動画の切り取り」について説明する。素材動画はそのまま使用することもできるが、一部分を切り取って使用する機能も付けた。切り取りはカットを始めるフレームとカットを終えるフレームの指定により行う。< movie >タグの子に、< cut >タグを入れ、さらに< cut >に< start >と< end >の子を付ける。id が「00000003」の素材動画の0秒から4秒の部分の切り取りたい場合には、下図のように< start >に0、< end >に120を要素として格納することで使用できる。

```
<movie id="00000003">
  <cut>
    <start>0</start>
    <end>120</end>
  </cut>
</movie>
```

図 4.5: 素材動画の切り取りの XML

4.3.4 連結部分のフェード

次に「連結部分のフェード」について説明する。素材動画の連結の際に、映像に別の映像を徐々に掛け合わせることで、フェードを実現する。XMLには<fade>タグの要素にフレーム数を格納し、フェードの種類は属性の”type”に格納される。ここで使用するフェードの種類ごとに説明する。

黒い画面から徐々に素材動画が見えるようになる効果がフェードインである。下図では、idが「0000001」の動画に30フレームのフェードインをかける効果を付けたXMLであり、下図のように暗い画面から指が徐々に現れる。

```
<fade type="fadein">30</fade>  
<movie id="0000001"></movie>
```

図 4.6: フェードインのXML



図 4.7: フェードイン

素材動画が徐々に暗くなる効果がフェードアウトである。下図では、id が「0000001」の動画に 30 フレームのフェードアウトをかける効果を付けた XML であり、下図のように指が徐々に黒に隠れていく。

```
<movie id="0000001"></movie>  
<fade type="fadeout">30</fade>
```

図 4.8: フェードアウトの XML



図 4.9: フェードアウト

前の素材動画が徐々に暗くなり一度黒い画面が表示されてから次の素材動画が徐々に見えてくる効果が、フェードインとフェードアウトを組み合わせたフェードアウトインである。下図では、id が「00000001」の動画と「00000002」の動画に 30 フレームのフェードアウトインをかける効果を付けた XML であり、下図のように 1 の指が徐々に黒に隠れていき、2 の指が徐々に見えてくる。

```
<movie id="00000001"></movie>  
<fade type="fadeoutin">30</fade>  
<movie id="00000002"></movie>
```

図 4.10: フェードアウトインの XML

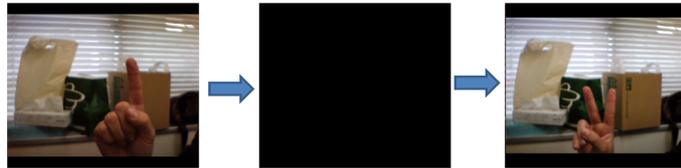


図 4.11: フェードアウトイン

前の素材動画が次の素材動画に徐々に切り替わっていく効果がクロスフェードである。下図では、id が「0000003」の動画と「0000001」の動画に 60 フレームのクロスフェードをかける効果を付けた XML であり、下図のように 3 の指が徐々に 1 の指に変わっていく。

```
<movie id="0000003"></movie>  
<fade type="crossfade">60</fade>  
<movie id="0000001"></movie>
```

図 4.12: クロスフェードの XML



図 4.13: クロスフェード

これらの編集機能を利用して、動画を編集する。動画を編集する指示を与える XML でありながら、そのまま編集履歴として利用できる構造をもった XML である。次章で、XML による動画編集システムの実装について説明する。

第5章 動画編集システムの実現法

本章では、XML を利用したインターネット上での動画編集システムの実現法について説明する。

5.1 システム構成

動画編集システムの実装を説明する前に、本システムの狙いを説明する。現在の動画投稿システムでは投稿された動画の編集内容、使用した動画をユーザーは知ることができない。そのため、動画の関連付け、動画の一部のシーン検索、動画のソース検索等が困難である。そこで、サーバー上に投稿された動画を使用し、インターネット上で XML を利用して動画を編集、完成した動画には編集時に使用した XML を関連付けさせることで、完成した動画の詳細を知ることが出来る。以上が本システムの狙いであり、前章までで説明した XML、OpenCV で作成した動画編集プログラム、WPF を用いて作成したユーザーインターフェースにより、そのシステムの実装は可能になる。

動画編集システムの実現について説明する。本システムはインターネット上での操作によって一連の作業すべてが完結することを目標としている。その実装に必要なものを順に説明する。まず、素材となる動画を用意する。それぞれに ID を振り分け、関連したメタデータ XML を作成する。素材となる動画とメタデータ XML はサーバーに保管する。そして、編集の際に XML を作成する必要があるが、XML を知らないユーザーにも作成出来るように、ユーザーインターフェース (UI) を使用することで編集用 XML が記述できるようにする。ユーザーインターフェースは、編集用 XML の作成のみに使用するのではなく、操作により作成された編集用 XML をサーバーに送る機能を付ける。OpenCV で作成した動画編集プログラムは、ユーザーインターフェースにより作成され送信された XML を読み込み、動画を作成する。完成した動画をサーバー内に保存し、動画の編集履歴を格納した XML がサーバ内に保管される。以上がシステムの構成と流れである。

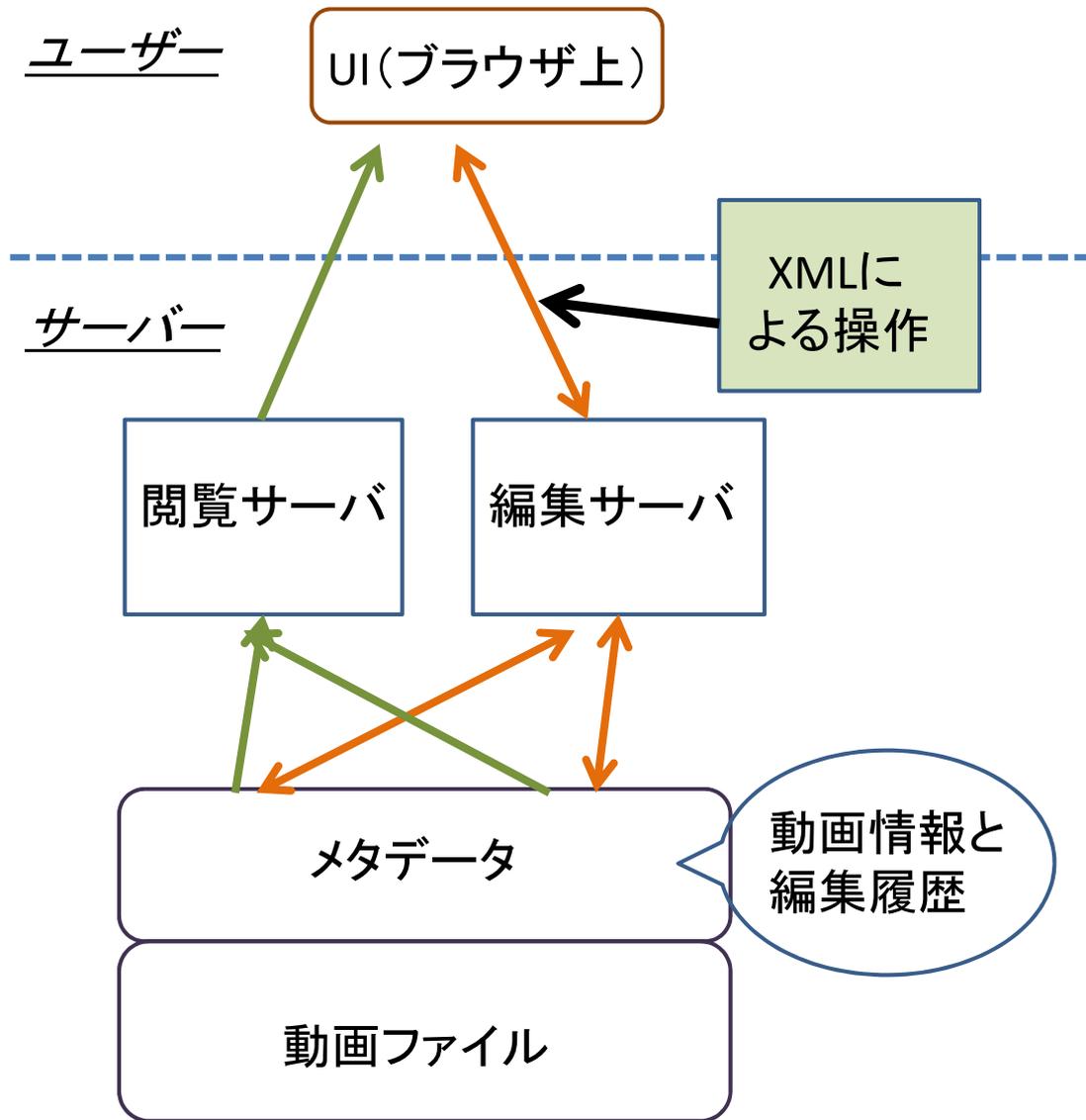


図 5.1: システムの構成

5.2 OpenCV での動画編集

XML に編集内容が格納された後、そのデータを読み込み、編集を行う。その編集を OpenCV を用いて行う。まず、以下で OpenCV を説明する。

OpenCV

OpenCV とは、Intel によって開発された、画像認識に関連する機能のライブラリのことである。OpenCV は、コンピュータビジョンと呼ばれる、画像認識・解析に関するコンピューティング技術が主な用途として想定されている。C 言語、C++ によって記述することが可能であり、Windows や Linux など複数のプラットフォームに対応している。BSD ライセンスに基づくオープンソースソフトウェア (OSS) として提供されているため、誰でも無償で利用することができる。OpenCV を用いることによって、例えば物体の認識、パターン認識、動作の認識といった、コンピュータビジョンに関する高度な画像処理機能を容易に利用できるようになる。OpenCV は米国の SourceForge.net などダウンロード可能である。日本では opencv.jp の Web サイト上でリファレンスマニュアルの製作・公開が行われている。

本システムでは動画について扱うが、編集を行い動画を作成する際にフレーム単位で処理が行われる。上の説明にあるように、OpenCV は画像の処理に特化したライブラリである。画像の編集に特化したライブラリを使用してプログラムを作成することで、動画の編集機能のフェードや切り取りの実装が容易であり、新たな機能を開発する際にも、プログラムの拡張の幅が広がる。そのため、OpenCV を使用した。OpenCV で開発した動画編集プログラムでは主に以下の 4 点が順に行われる。

- XML の読み込み
- 素材動画の検索、読み込み
- タイトルを作成
- 素材動画を編集し繋げ、新しい動画を作成

ユーザーインターフェースから送信されてきた編集用 XML を、サーバー内にある OpenCV で作成した動画編集プログラムが受け取り、各データを読み込む。編集用 XML に格納されたデータをもとに素材動画を読み込み、編集する。編集を終えた完成動画をサーバー内に保存し、編集用 XML もサーバー内に保存する。以下で主に行われる 4 点を詳しく説明する。

5.2.1 XML の読み込み

まずは、ユーザーインターフェースから送信されてきた編集用 XML を読み込み、データを一時的に保持する。ただし、編集用 XML には素材動画のデータとして id しか記述されていない。そのため、素材動画の保存先アドレスが編集用 XML だけでは判別できない。メタデータ XML も読み込む必要があるため、編集用 XML が送信されてきたタイミングで、同時にメタデータ XML も読み込んでおく。

5.2.2 素材動画の検索、読み込み

次に、編集用 XML から抜き出したデータの中から、素材動画の id を使用される順に抜き出す。その id と、メタデータ XML に格納されている id が一致する素材動画のデータを検索する。一致したメタデータ XML の素材動画の id の子には、動画の保存先のアドレスが格納されている。このアドレスのデータをもとに素材に使う動画を読み込む。動画の読み込みは使用する順に行い、同時に 3 つ以上の素材動画を利用する編集がないため、同時に最大 2 つまでしか読み込むことが出来ないようになっている。しかし、フェード機能では黒い画面との掛け合わせが必要になるため、素材動画の読み込みとは別に、黒い画面のみの動画をあらかじめ読み込み、保管してある。

5.2.3 タイトルの作成

編集用 XML に格納してあるタイトルのデータを利用し、タイトルを作成する。利用するタイトルのデータは、タイトルネームとフレーム数である。黒い画面だけが映った動画からフレームごとに画像を抜き出し、白いタイトルネームを張り付ける。その画像をフレーム数分繋げることで、動画のタイトル部分となる。

5.2.4 素材動画を編集し繋げ、新しい動画を作成

次に、動画の本編部分の作成を説明する。フレーム単位の画像を連続して書き出すことにより動画と作成する。編集用 XML の編集内容に記述されている順番通りに新しい動画に書き出していく。素材動画を使用する場合は、切り取りがあるかどうかをまず判別する。切り取りがない場合は、読み込んだ素材動画をワンフレームずつ読み込み、新しい動画に書き出していく。切り取りの編集がある場合には、編集用 XML から抜き出したスタートフレームとエンドフレームのデータを用いて、カットスタートフレームから書き出しをはじめ、カットエンドフレームで書き出しを止める。フェードの機能を付ける際には、フェードイン、フェードアウト、フェードアウトインとクロスフェードで区別する。フェードイン、フェードアウト、フェードアウトインでは、タイトルの作成時にも使用した黒い画面だけが映った動画と素材動画の掛け合わせで行う。この編集も、画像ごとに処理を掛け、順に新しい動画に書き出していく。クロスフェードでは、素材動画と素材動画をフレーム単位でそれぞれ画像を抜き出し、画像の掛け合わせにより行う。編集用 XML に記述された編集内容を順に書き出していき、すべての編集内容を書き出し終えた時点で、新しい動画が編集用 XML に書かれた編集内容通りに作成されている。こうして新しい動画が作成される。

5.3 WPF で作成した UI で XML を作成

XML を知らないユーザーでも動画の編集の内容を記述した XML を作成することが出来るように、ユーザーインターフェースを作成した。本システムを利用する際には、このユーザーインターフェースだけで作業を完結できるように、サーバーへの編集用 XML の送信も可能にした。作成には、WPF(Windows Presentation Foundation) を用いた。WPF はデスクトップ上でのアプリケーションだけでなく、ウェブブラウザ上で配置・実行することができる。Java よりも 2D および 3D オブジェクトの描画、ベクトルグラフィックス、ラスタグラフィックス、アニメーション、音声および動画の再生などといった表現手法を統一的に利用することができるため、動画編集システムのユーザーインターフェースに適している。さらに XAML と呼ばれる XML をベースとしたマークアップ言語で作成できるため、容易に XML の利用ができる点で、本システムには適している。

本研究で作成したユーザーインターフェースでは、以下の機能を持つ。

- 編集用 XML の下地の作成
- タイトル情報の入力
- 素材動画を繋げる
- 素材動画の参照とプレビュー
- フェード機能を付加
- 作成した XML をサーバーに送信

下図に示すインターフェースのボタンと入力部分によりそれぞれの操作が可能になる。「新規作成」ボタンにより編集用 XML の下地の作成、「タイトル設定」ボタンによりタイトル情報の入力、「combine」ボタンにより素材動画を繋げ、「参照」ボタンと「Play」ボタンにより素材動画の参照とプレビュー、「フェード付加」ボタンによりフェード機能を付加、「確定」ボタンにより作成した XML をサーバーに送信することができる。それぞれの機能について、順に説明する。

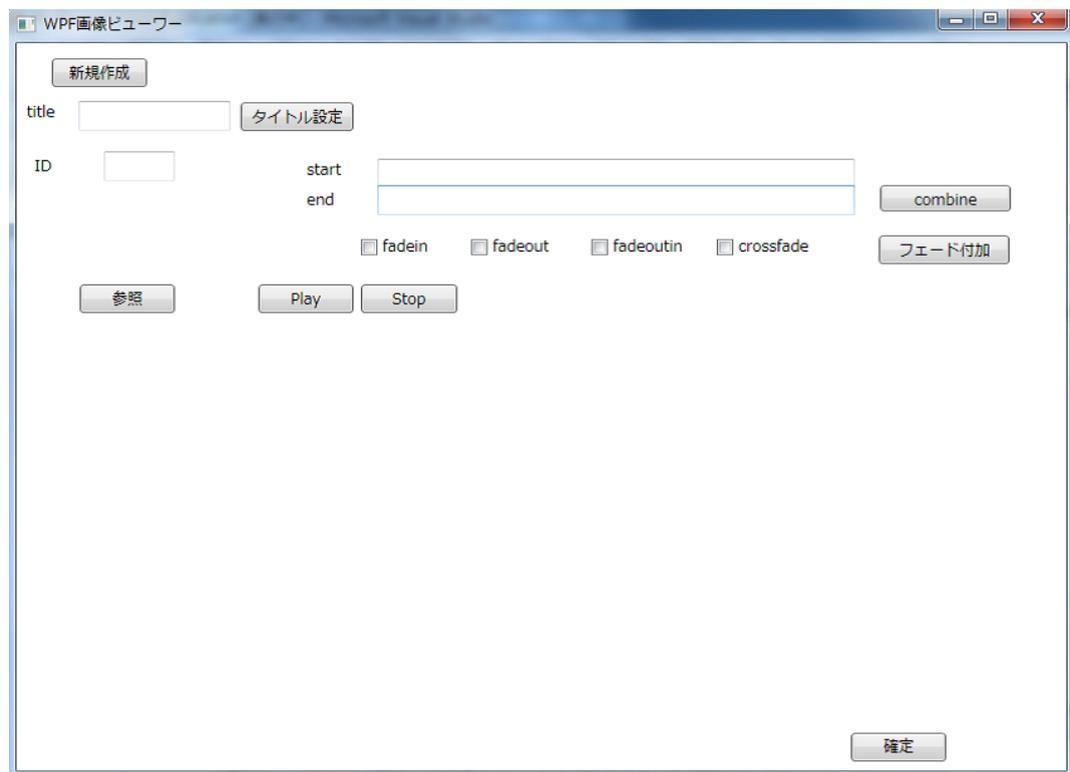


図 5.2: 作成したユーザーインターフェース

5.3.1 編集用 XML の下地の作成

「新規作成」ボタンを押すことにより、id、address の 2 つのメタ情報と、編集内容を格納する < edit > タグを記述した XML が下図のように作成される。ユーザーインターフェースによる操作は、参照とプレビューを除いて、「新規作成」ボタンにより作成された下地となる XML の < edit > タグの中へ記述される。下図では、作成される新しい動画の id が「1000000」、保存先は「1000000.avi」と記述してある。これは新しい動画のメタデータにあたる情報である。

```
<?xml version="1.0" encoding="UTF-8"?>
<themovie>
  <header>
    <id>1000000</id>
    <address>1000000.avi</address>
  </header>

  <!--edit内容-->
  <edit>
  </edit>
</themovie>
```

図 5.3: 下地となる編集用 XML

5.3.2 タイトル情報の入力

次に、title の右の空欄に入りたいタイトルネームを入力し、「タイトル設定」ボタンを押すことにより、「新規作成」ボタンで作成された編集用 XML の < edit > タグの子に < titleplus > タグが作成され、タイトルネームが格納される。今回はタイトルの表示フレーム数を 200 に固定して使用している。下図は < edit > タグの下に、< titleplus > タグの要素を「sample」としての新たに格納した編集用 XML である。この「sample」はタイトルネームとして編集時に利用される。

```
<?xml version="1.0" encoding="UTF-8"?>
<themovie>
  <header>
    <id>10000000</id>
    <address>10000000.avi</address>
  </header>

  <!--edit内容-->
  <edit>
    <titleplus length="200">sample</titleplus>
  </edit>
</themovie>
```

図 5.4: タイトルの格納された XML

5.3.3 素材動画を繋げる

新しく作成する動画の本編部分を作成する。素材となる動画を使用する順番に繋げていく機能を説明する。ID の右の空欄に、使用する素材動画の id を入力する。素材動画の一部を使用したい場合には、start の右にある空欄に切り取りを始めるフレームを、end の右にある空欄に切り取りを終えるフレームを入力することで、カットの機能を付け加えることが出来る。入力を終え、「combine」ボタンを押すことにより、編集用 XML の < edit > タグの子に < movie > タグが記述され、その属性に id を格納し、子に < cut > タグを作成して start と end の情報を要素として格納する。下図では、id が「00000001」の動画を「0」フレームから「200」フレームまで使用する編集内容が格納されている。

```
<?xml version="1.0" encoding="UTF-8"?>
<themovie>
  <header>
    <id>10000000</id>
    <address>10000000.avi</address>
  </header>

  <!--edit内容-->
  <edit>
    <titleplus length="200">sample</titleplus>
    <movie id="00000001">
      <cut>
        <start>0</start>
        <end>200</end>
      </cut>
    </movie>
  </edit>
</themovie>
```

図 5.5: 使用する素材動画の格納された XML

5.3.4 素材動画の参照とプレビュー

素材動画の id が分からない場合には、「参照」ボタンを押すことで、下図のように投稿された動画の一覧を見ることが出来る。参照で選んだ素材動画の内容を確認したい場合には、「Play」ボタンを押すことで、下図のように動画の内容が再生できる。

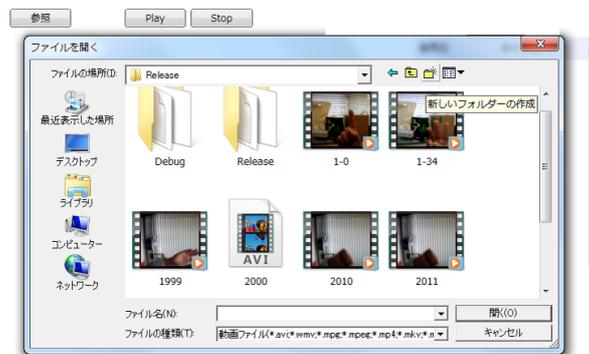


図 5.6: 参照のサンプル

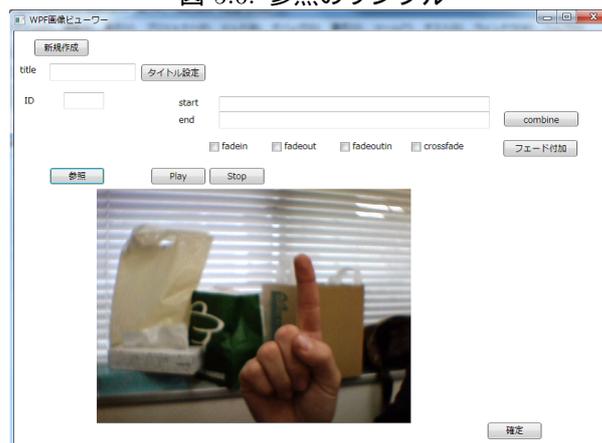


図 5.7: プレビューのサンプル

5.3.5 フェード機能を付加

フェード機能を付け加える際には、fadein、fadeout、fadeoutin、crossfade のいずれかにチェックを付け、「フェード付加」ボタンを押すことで、編集用 XML の < edit > タグの子に < fade > タグが作成され、その属性にフェードのタイプが格納され、要素にフレーム数が格納される。フェードイン、フェードアウト、フェードアウトインではフレームが 30、クロスフェードではフレームが 60 に固定してある。下図では id が「00000001」の動画をフェードアウトさせる「fadeout」が属性に、フレーム数の「30」が格納されている。

```
<?xml version="1.0" encoding="UTF-8"?>
<themovie>
  <header>
    <id>10000000</id>
    <address>10000000.avi</address>
  </header>

  <!--edit内容-->
  <edit>
    <titleplus length="200">sample</titleplus>
    <movie id="00000001">
      <cut>
        <start>0</start>
        <end>200</end>
      </cut>
    </movie>
    <fade type="fadeout">30</fadeout>
  </edit>
</themovie>
```

図 5.8: フェード機能が格納された XML

5.3.6 作成した XML をサーバーに送信

すべての編集内容を編集用 XML に記述した後、「確定」ボタンを押すことで、作成した編集用 XML をサーバーに送信することが出来る。送信すると同時に、OpenCV で作成した動画編集プログラムも起動され、送信した編集用 XML から編集内容を読み取り、新しい動画が作成される。

以上で説明した機能が、ユーザーインターフェースには搭載してある。これらの機能により、本研究で使用する動画の編集機能を実現することが出来た。次章で実際にシステムを構築し、動作を確認する。

第6章 実験と考察

本章では、動画編集システムを用いて動画を作成する実験を行った。今回、実装上の都合により本研究のシステムをオンライン上ではなくローカルでの利用とした。ローカルな環境で試行し、システムの動作を確認することで、その有用性を確認する。

6.1 実験

実際にシステムを使用し、動画を作成した。実験の流れを順に説明する。

6.1.1 実験環境

実験を行う前に、動画作成に用いる素材動画と、素材動画のメタデータを格納したメタデータ XML を用意しなければならない。素材動画を3つ用意した。一本の指が5秒間円を描いて動く動画、二本の指が5秒間円を描いて動く動画、三本の指が3秒間円を描いて動き、途中から指が四本になり3秒間円を描いて動く動画の3本である。それぞれのidは、指1が「00000001」、指2が「00000002」、指34が「00000034」である。素材動画のメタデータとメタデータ XML について一本の指の動画を例に挙げて説明する。一本の指の動画のメタデータは、idが「00000001」、動画のタイトルが「指1」、保存先のアドレスが「C:\Users\sano\Desktop\sano\OpenCV\LaughingMan\LaughingMan\sozai01.avi」、動画の容量が「6.3MB」、総フレーム数は「150」、フレームレートは「30」、動画の分類は「人」である。これらのメタデータは、すべての動画のメタデータを格納したメタデータ XML に格納されている。下図のように”id”に「00000001」、< title >に「指1」、< address >に「C:\Users\sano\Desktop\sano\OpenCV\LaughingMan\LaughingMan\sozai01.avi」、< size >に「6.3MB」、< length >に「150」、< framerate >に「30」、< category >に「人」が格納されている。タグの< face >は今回使用しないため、すべて「on」にしてある。他の二本の素材動画の情報も下図のようにメタデータ XML に同じ形式で格納されている。

```
<allmovie>
  <movie id="00000001">
    <address>C:\Users\sano\Desktop\sano\OpenCV\LaughingMan
    ¥LaughingMan¥sozai01.avi</address>
    <title>指1</title>
    <size>6.3MB</size>
    <length>150</length>
    <framerate>30</framerate>
    <category>人</category>
    <face>on</face>
  </movie>

  <movie id="00000002">
    <address>C:\Users\sano\Desktop\sano\OpenCV\LaughingMan
    ¥LaughingMan¥sozai02.avi</address>
    <title>指2</title>
    <size>6.3MB</size>
    <length>150</length>
    <framerate>30</framerate>
    <category>人</category>
    <face>on</face>
  </movie>

  <movie id="00000034">
    <address>C:\Users\sano\Desktop\sano\OpenCV\LaughingMan
    ¥LaughingMan¥sozai034.avi</address>
    <title>指34</title>
    <size>12.5MB</size>
    <length>270</length>
    <framerate>30</framerate>
    <category>人</category>
    <face>on</face>
  </movie>

</allmovie>
```

図 6.1: メタデータ XML

6.1.2 編集内容

次に、今回の実験で作成する動画の編集内容を説明する。下図にある素材動画3つを使用して、1つの動画を作成する。今回の編集内容は、下図の編集内容にあるように、まずタイトルを冒頭に付ける。タイトルネームは「fingers」とする。次に、指34の素材動画から三本の指が円を描いている部分だけを抜き出し使用する。次に指1の素材動画をフェードインの効果とフェードアウトの効果をつけて使用する。次に指2の素材動画をフェードインの効果とフェードアウトの効果をつけて使用する。次に指1と指34をクロスフェードをかけて繋げ、最後の指34にフェードアウトの効果をつける。今回の実験では以上の編集を行うこととする。

素材動画



編集内容

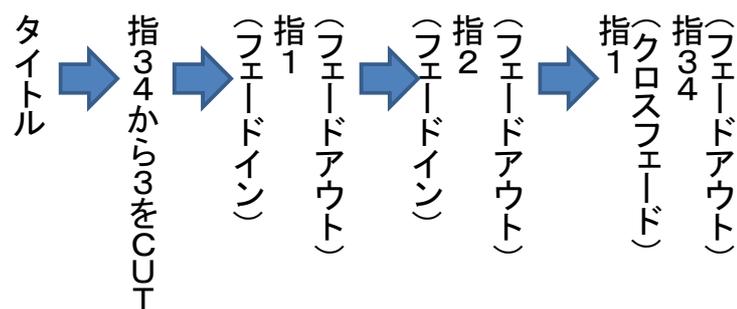


図 6.2: サンプル動画の編集内容

6.1.3 ユーザーインターフェースを使用し編集用 XML を作成

ユーザーインターフェースを使用して編集内容を編集用 XML に記述していく。まず「新規作成」ボタンを押す。それにより、編集用 XML のひな型が作成される。その後、title の欄に「fingers」と入力し、「タイトル設定」ボタンを押す。それにより、タイトルの編集内容が下図の < titleplus > タグのように編集用 XML に記述される。次に、指 3 4 を使用するため、ID の欄に「00000034」と入力する。しかし、その一部を抜き出して使用するため start と end の欄にも入力する。指 3 4 の素材動画の中で三本の指が表示されているのは、はじめから 3 秒間なので、start の欄に 0、end の欄に 90 と入力する。入力後に「combine」ボタンを押すことで、下図の属性 id="00000034" を持つ < movie > タグのように切り取りの < cut > タグを含めた編集内容が記述される。次にフェードインを指 1 の素材動画に付けるが、編集用 XML はタイムラインと同じように記述されるため、指 1 の素材動画を繋げる前に「fadein」にチェックを付けて「フェード付加」を押す。それにより、下図のような属性に"fadein" が格納された < fade > タグが作成される。その後、ID の欄に「00000001」と入力後、「combine」ボタンを押すことで下図のように < fade > タグの下に < movie > タグが作成され、「00000001」が格納される。このように、ボタンを押すことにより、編集内容が編集用 XML に格納されていく。次は、指 1 のフェードアウトだが、その次の編集がフェードインとなるため、ここでは「fadeoutin」にチェックを付けて「フェード付加」を押す。そのあとは指 2 の ID を入れ「combine」を押し、「fadeout」にチェックを付け「フェード付加」を押す。次も指 1 の ID を入れ「combine」を押し、「crossfade」にチェックを付け「フェード付加」を押し、指 3 4 の ID を入力して「combine」を押し、最後に「fadeout」にチェックを付けて「フェード付加」を押す。以上の操作により、最終的に下のような編集用 XML が完成する。

```
<?xml version="1.0" encoding="UTF-8"?>
<themovie>
  <header>
    <id>00000000</id>
    <address>fingers.avi</address>
    <category>指</category>
  </header>

  <!--edit内容-->
  <edit>

    <titleplus length="200">fingers</titleplus>

    <movie id="00000034">
      <cut>
        <start>0</start>
        <end>90</end>
      </cut>
    </movie>

    <fade type="fadein">30</fade>
    <movie id="00000001"> </movie>
    <fade type="fadeoutin">30</fade>
    <movie id="00000002"></movie>
    <fade type="fadeout">30</fade>

    <movie id="00000001"></movie>
    <fade type="crossfade">60</fade>
    <movie id="00000034"></movie>
    <fade type="fadeout">30</fade>

  </edit>
</themovie>
```

図 6.3: 編集内容を格納した編集用 XML

6.1.4 編集用 XML を送信し、動画編集プログラムで動画を作成

編集用 XML を完成させた後は、ユーザーインターフェースの「確定ボタン」を押す。それにより、OpenCV で作成した動画編集プログラムが起動し、作成された編集用 XML を読み込む。編集内容を一時的に保持する。今回では、「00000034」「00000001」「00000002」「00000001」「00000034」のように使用する順に id が保持してある。そして、動画を順に書き出していく。まず、タイトルが作成される。黒い画面のみが映った画面に、白い文字で「fingers」と表示されるフレームを 200 フレーム書き出す。その後、id の並び順にメタデータ XML に格納された保存先の情報を検索し、素材動画を読み込む。そして、フェードや切り取りの機能を加えながら、素材動画を加工して、タイトルの後ろに順に書き出していく。すべての編集を書き出した後には、編集用 XML に記述された編集内容通りに作成された「fingers.avi」の動画と、その編集履歴が格納された XML が保存される。ここでシステムの一連の流れが終了する。

6.2 考察

今回の実験では、システムの動作を確認することで、その有用性を確認することが目的であった。本システムの最重要となる XML に関しては、作成時にユーザーインターフェースの利用により、編集用 XML の作成が比較的容易にでき、動画作成後に残った編集用 XML も、今後の検索等に利用が可能な形で編集履歴を格納することが出来たため、XML の性質を本システムに有効活用することが出来た。しかし、OpenCV で作成した動画編集プログラムでは、音声の編集が出来ないため、動画の編集として足りない部分もあった。

第7章 おわりに

既存の動画共有サービスでは、Web 上の動画を観賞することに重きを置いて設計されている。動画を観賞するシステムの効率化や、観賞の満足度を高めるために、検索システムやコメント等の様々な機能が搭載されている。本研究では、Web 上で動画編集を行うことが可能なシステムを提案した。動画共有サービスに投稿された動画を観賞だけでなく、新しい動画の制作にも利用可能としたため、動画共有サービスの利用の幅が広がり、その結果、新しい動画の編集履歴を格納した XML を利用する検索機能の、大幅な拡張が見込めることとなった。

本研究では、XML を利用した動画編集システムの提案が狙いであったため、システムの基礎の部分となる、メタデータ XML、編集用 XML、OpenCV による動画編集プログラム、WPF によるユーザーインターフェースを構築した。今後は、動画を編集するだけでなく、その後の Web 上での観賞や、XML を利用した検索システムといった拡張を行っていききたい。今後の課題として、以下のような事柄をあげる。

- サーバーを構築し、Web 上でのシステム構築
ローカルではなく、Web 上のサーバーとブラウザでの操作によりシステムを構築する。
- 音声編集プログラムを付け、動画編集プログラムに組み込む
音声編集プログラムを作成する。別のプログラムで作成することにより音声だけの抽出や再利用が容易になる。
- 動画編集プログラムの拡張
Web 上の動画を素材として利用できるシステムのため、モザイク機能等のプライバシー保護の対策を編集機能に追加する。
- ユーザーインターフェースの改良
編集用 XML の作成に ID の入力に依存するのではなく、直感的にドラック & ペーストのマウス操作により使い勝手の良いシステムに改良する。
- 検索システムの構築と実装
メタデータ XML と編集履歴を格納した XML を利用した、動画の検索システムの構築と実装。

これらの課題を含め、XML を利用したインターネット上での動画編集システムに関する研究を進めることで、動画編集履歴を用いた様々な利用が可能なシステムを目指し、多くのユーザーに利用されうる利便性を高めた動画編集システムにしていきたいと考えている。

謝辞

本研究を進めるにあたって、日ごろから多大な御尽力をいただき、ご指導を賜った名古屋工業大学 舟橋健司 准教授、名古屋工業大学 山本大介 助教に心から感謝いたします。また、本研究に対してご検討、ご協力いただきました名古屋工業大学 伊藤宏隆 助教に心から感謝いたします。最後に、本研究を進めるにあたり多大な協力をいただいた舟橋研究室諸氏に心から感謝いたします。

参考文献

- [1] 北山 洋幸 : “OpenCV で始める動画プログラミング” , カットシステム , 2010 .
- [2] PROJECT KySS : “Silverlight 実践プログラミング” , 工学社 , 2010 .
- [3] 高橋 麻奈 : “やさしいXML” , ソフトバンククリエイティブ, 2005.
- [4] 山本 大介, 増田 智樹, 大平 茂輝, 長尾 確 : “タグクラウド共有に基づく協調的映像アノテーション” , 人工知能学会論文誌, Vol. 25, No. 2, pp. 243-251, 2010.1.
- [5] Daisuke YAMAMOTO, Tomoki MASUDA, Shigeki OHIRA, Katashi NAGAO : “Video Scene Annotation Based on Web Social Activities” , IEEE Multimedia, Vol.15, No.3, pp.22-32, 2008.9.
- [6] Tomoki Masuda, Daisuke Yamamoto, Shigeki Ohira, and Katashi Nagao : “Video Scene Retrieval Using Online Video Annotation” , New Frontiers in Artificial Intelligence - JSAI 2007 Conference and Workshops Revised Selected Papers -, LNAI 4914, Springer-Verlag, pp.54-62 (2008.2)
- [7] 山本 大介, 増田 智樹, 大平 茂輝, 長尾 確 : “映像を話題としたコミュニティ活動支援に基づくアノテーションシステム” , 情報処理学会論文誌, Vol.48, No.12, pp. 3624-3636, 2007.12.
- [8] 山本 大介, 長尾 確 : “閲覧者によるオンラインビデオコンテンツへのアノテーションとその応用” , 人工知能学会論文誌, Vol.20, No.1, pp.67-75, 2005.1
- [9] YouTube : “YouTube” , <http://www.youtube.com/?gl=JP&hl=ja>
- [10] ニコニコ動画 : “ニコニコ動画 (原宿)” , <http://www.nicovideo.jp/>
- [11] Ustream : “Ustream” , <http://www.ustream.tv/>