筑波大学大学院博士課程

システム情報工学研究科修士論文

マーカーによる撮影中のタグ付けを利用した 映像オーサリングシステム

中井川 峻

(コンピュータサイエンス専攻)

指導教員 高橋 伸

2012年3月

概要

動画共有サービスの流行と映像制作環境の普及により,映像作品を見るだけではなく制作 して世界中の人と共有することが一般的になりつつある.実写映像を用いた作品も人気となっ ており,エンドユーザ向けの映像制作環境へのニーズが増加している.

本稿では,映像制作における撮影時に2次元マーカーによるタグ付けを行うことにより,タ グ情報を基にした半自動的な編集を可能とするエンドユーザ向け実写映像作品制作支援シス テムについて述べる.提案システムは,一般的な映像オーサリングのプロセスとして用いら れる,企画,撮影,編集という3つのフェーズを総合的に支援する.

企画フェーズにおいて,作品に必要なシーン等の情報を入力することによりタグ情報とマー カーのセットをシーンの順序に沿って生成する.撮影フェーズでは,企画フェーズで生成さ れた順番に従ってマーカーをカメラに写すことにより,シーンの順序に沿ってタグが付加さ れた映像を生成することが出来る.そして編集フェーズでは,付加されたタグ情報を利用す ることで企画内容に沿った自動的な編集を行うと共に,その後の手動による編集も可能とす る.このように,タグとマーカーを組み合わせることにより,タグを介した各フェーズ間の 情報共有を可能とする.この共有された情報を用いて,今までにない有効な自動編集を可能 とし,エンドユーザにも容易に映像を制作することができる環境を実現する.

目 次

第1章	はじめに	1
1.1	研究の背景・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	1
1.2	本研究の目的と貢献・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	2
1.3	本論文の構成	3
第2章	関連研究	4
2.1	映像編集の自動化および半自動化手法..........................	4
2.2	メタデータを利用した編集支援手法...........................	5
2.3	映像への自動タグ付け手法................................	5
2.4	拡張現実感技術を用いた映像オーサリングに関する研究・・・・・・・・・	6
第3章	マーカーを利用した映像オーサリング支援手法	7
3.1	実写映像作品の制作プロセス..............................	7
3.2	映像オーサリングにおけるマーカーとタグを介した情報共有	9
	3.2.1 本研究におけるタグ	9
	シーンタグ	10
	オブジェクトタグ..............................	11
3.3	提案システムのインタフェースと操作........................	12
	3.3.1 企画モジュールのインタフェースと操作	12
	シーンタグを作成する..............................	12
	オブジェクトタグを作成する............................	14
	絵コンテを描く	15
	プロジェクトをエクスポートする	16
	3.3.2 カメラモジュールとマーカー表示モジュールのインタフェースと操作	16
	マーカーの準備	17
	カメラモジュールを起動する............................	19
	撮影を開始する.................................	19
	シーン内容の撮影を行う..........................	20
	シーン撮影を終える	20
	シーンの撮り直し................................	22
3.4	編集モジュールのインタフェースと操作	22
	3.4.1 プロジェクトファイルの読み込み	22

		文字列クエリによる映像内検索	22
		手動編集と映像ファイル出力	24
3.5	ビデオ	↑制作の流れ....................................	24
	3.5.1	企画フェーズ	24
	3.5.2	撮影フェーズ	25
	3.5.3	編集フェーズ	26
第4章	マーカ	コーを利用した映像オーサリングシステムの開発	28
4.1	企画モ	================================	28
	4.1.1	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	28
	4.1.2	企画モジュールの実装内容	28
			29
		絵コンテ画像の生成	29
4.2	カメラ		30
	421	カメラモジュールの実装環境	30
	422	カメラモジュールの実装成容	30
	1.2.2	プロジェクトファイルの読み込み	31
		カメラからの画像・音声取得とその録画	31
		マーカー認識と認識情報の記録	32
4.3	マーカ		33
	4.3.1	マーカー表示モジュールの実装環境	33
	4.3.2	マーカー表示モジュールの実装内容	33
		実装した Android 端末	33
		プロジェクトに関する情報の共有	34
4.4	編集モ	ミジュール	34
	4.4.1	編集モジュールの実装環境・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	34
	4.4.2	編集モジュールの実装内容・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	35
		プロジェクトの読み込み	35
		シーンの自動編集・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	36
		オブジェクトの自動編集	36
		キーワード検索	38
笛 5 音	証価		30
ポジ ギ 51	計画の)内容	39
5.2	山 山 市 の 加 市 の	* · · · · · · · · · · · · · · · · · · ·	30
53	就田理	ョ ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	30
5.5 5.4	前田始	*元 · · · · · · · · · · · · · · · · · · ·	70 70
5.4	n=v/=⊐i#⊏ 5_/_1	- ^ · · · · · · · · · · · · · · · · · ·	40
	517	エロッ ^{mu} 不 · · · · · · · · · · · · · · · · · · ·	40
	5.4.2	」ヰメ ホノマノ ハ山 ハヘ	42

5.5 5.6	自動編集後のタグ付き映像................................ アンケートへの回答結果と考察................................	42 44
第6章	まとめと今後の展望	46
	謝辞	47
	参考文献	48

図目次

1.1	モザイク編集を施した楽器演奏動画 ¹	2
3.1	絵コンテの一例 ¹	8
3.2	カチンコの一例 ¹	8
3.3	システムの全体像...................................	10
3.4	シーンタグ	11
3.5	オブジェクトタグ................................	11
3.6	シーンタグを作成する操作..............................	13
3.7	オブジェクトタグを追加する操作	14
3.8	絵コンテを描く操作・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	15
3.9	プロジェクトをエクスポートする操作.........................	16
3.10	撮影モジュールの概観	17
3.11	プロジェクトリスト画面	18
3.12	マーカーリスト画面	18
3.13	マーカー表示画面.................................	18
3.14	撮影前のプレビュー	19
3.15	撮影開始....................................	20
3.16	撮影中	21
3.17	シーン以外の区間....................................	21
3.18	プロジェクトファイル読み込み後の編集モジュール	23
3.19	研究紹介映像のイメージ図.................................	24
3.20	編集モジュールでの自動編集結果	26
4.1	各フィルタグラフの構成	32
4.2	Android アプリを起動した様子	34
4.3	絵コンテ画像レイヤー (オブジェクト) に設定したキーフレームの一例	37
51	企画を行う被騒者	40
5.2	山口になった。	40
53	シーン・「Wii リモコンの動き」の絵コンテ	41
5.5 5.4	オブジェクト・「投影画面」の絵コンテ	<u>4</u> 1
J. T		-11

5.5	シーン : 「 Wii リモコンの動き 」 における , オブジェクト : 「 バーチャル Wii リ	
	モコン」の時刻による変化..............................	43

第1章 はじめに

1.1 研究の背景

YouTube¹ やニコニコ動画² などの動画共有サービスの流行により,インターネットを用 いた映像閲覧が盛んに行われるようになった.また,安価なコンピュータや安価なカメラ, Microsoft のムービーメーカー[1] や Apple の iMovie[2] などの映像編集ソフトウェアといった 映像オーサリング環境の普及により,映像作品を見るだけではなく,制作して世界中の人と 共有することが一般的になりつつある.

動画共有サービスに投稿される作品には,楽器演奏,工作などの成果発表や技術講座をは じめとした実写映像を用いたものがあり,人気のカテゴリとして挙げられる.そういった実 写映像作品内で行われるパフォーマンスの内容や投稿者の知名度,投稿のタイミングなどの 条件が同程度であった場合,映像編集を効果的に利用した,映像としての質が高い作品であ るほど高く評価される傾向にあるが,その一方で無編集の作品も数多く見受けられる.

この原因は,映像オーサリングに慣れていないエンドユーザにとって,その作業に時間と 労力がかかるためであると考えられる.より良い作品を制作するためには映像の撮り直しが 不可欠であり,それに伴って撮り直しのための時間,労力は増加していく.ユーザが映像オー サリングに慣れていないほど撮り直しが増加することは明らかであり,それと共に映像素材 ファイルの数も増加していく.検索性の低い映像というメディアから必要なシーンを見つけ 出すことはただでさえ困難であるが,特に,撮り直しにより増加した映像素材ファイルの中 からトリミングの始点や終点などの編集タイミングを見つけ出すことは非常に困難であると 言える.

また,映像中の動物体に対するモザイク処理といった,座標を利用した編集が困難である ことも問題である.動画共有サービスに映像作品を投稿する上で,高度な編集として用いら れてきた座標を利用した編集へのニーズが増加している.これは,投稿した映像作品は全世 界の不特定多数の人に見られる可能性があり,映像中における人物の顔部分を隠す合成編集 が重要であるためである(参考画像:図1.1).また,映像に写っている人物の手や足など,任 意の動く物体への CG 画像合成や視覚効果の付加などの編集も,エンターテイメント性を向 上させる手法として効果的である.

しかし , 映像中で動く物体 (以降 , オブジェクト) の座標を手動で追跡しながら編集を行う

¹Youtube

²ニコニコ動画

⁽URL: http://www.youtube.com/)

⁽URL:http://www.nicovideo.jp/)



図 1.1: モザイク編集を施した楽器演奏動画¹

のは大変手間のかかる作業である.具体的には,手作業で行った場合,対象の動きに合わせ て映像フレーム単位での位置調整を行う必要があり,対象が写っている時間が5分程度であっ ても,100以上のフレームのパラメータ操作を必要とする場合もある.こういった座標を利用 した編集は作業に慣れたユーザであっても時間と労力を要するものであり,エンドユーザに とって映像オーサリングへの敷居を高くする要因となっていると考えられる.撮影時に,ユー ザの手や足,頭などの部位を認識し,それらの動きを保持することで,認識位置への合成編 集を容易にする手法として,モーションキャプチャが挙げられる.モーションキャプチャは 高い精度で特定部位のトラッキングを行うことが可能であるが,使用する機材が高額である ことや利用するために高い技術が必要となることが問題である.Adobe の AfterEffects [3] や Premiere [4], Apple の Motion5[5] などに実装されたモーショントラッキング機能は,編集時 にユーザが指定した映像中のオブジェクトをトラッキングし,それに追従するような編集を 行うことが可能である.しかし,3次元的な物体追跡やその向きの認識を行うことができな い,事前の設定が複雑であるといった問題がある.

1.2 本研究の目的と貢献

上記の背景から,我々はエンドユーザ向けの映像オーサリング支援が必要であると考え,よ り直感的で容易な操作による映像オーサリング環境の構築を目指す.目的達成のため,本研 究では(1)企画内容のタグ化と,(2)撮影中における2次元マーカーを用いた実世界上でのタ グ付け,(3)付加されたタグを用いた自動編集,という3段階で映像オーサリング支援を行う システムの実装を行った.提案システムを利用することにより,マーカーを順番に写しなが ら撮影を行うという単純な操作による,企画に沿った簡易的な映像編集が可能となる.また, マーカーの座標・向きパラメータや,タグ付けされた映像区間,タグに付加された文字列な

¹flutewave「EUB で Chaining Intention 弾いてみた」より (URL: http://www.nicovideo.jp/watch/sm12807397)

どの情報を自動編集後に利用することで,特定物体へのモザイクや画像の重畳表示といった 座標・向きを用いた編集や,文字列クエリによるタイミング検索など,手動編集の支援も行う.さらに,提案システムについての評価を行い,その有用性を確認した.

1.3 本論文の構成

本論文の構成は以下の通りである.本章では,エンドユーザが映像オーサリングを行う上で の現状と問題点を挙げ,それを踏まえて我々の研究の目的と貢献を述べた.第2章では,映像 オーサリング支援や映像へのアノテーションなどの本研究に関連する研究について述べ,第 3章では,本研究における映像オーサリング支援手法について述べる.第4章では,手法を実 装した映像オーサリング支援システムについて述べ,第5章では,実装したシステムの評価 について述べると共に,その結果に対して議論を行う.最後に,第6章で結論を述べる.

第2章 関連研究

2.1 映像編集の自動化および半自動化手法

尾関らは,料理などの机上作業映像を自動編集する手法として,注意喚起行動に基づいた 編集手法を提案している[6,7].注意喚起行動とは,特定のものを指示する動作や,「こちら を見てください」といった発話など,料理番組などで視聴者を注目させるような行動である. 机上作業映像編集の基本パターンを,作業者と作業対象全体を撮影するシーンと,作業対象 のみを拡大して撮影するシーンの切り替えであると定義し,作業者の注意喚起行動を認識す ることにより,2つのシーン切り替えをタイミングを決定している.

Girgensohn らは,ホームビデオの半自動編集を行う Hitchcock システムを提案している[8]. Hitchcock では,カメラの動きと映像の明度から,重要なムービークリップを抽出するための スコア計算を導入し,動画の自動的なトリミングを行っている.そして,抽出したムービーク リップをストーリーボードにドラッグすることにより,ストーリーボードに配置されたムー ビークリップとの長さのバランスを自動で調整しながら,映像作品を手動で作ることができる.

西崎らは会話シーンを対象とした,映像の自動撮影・編集手法を提案している[9].この手法では,撮影現場の演技者の位置を把握するためのカメラモジュールと,コンテンツを撮影しつつ演技者の顔向きを取得する複数のカメラモジュールを協調動作させる.また,それと同時に,独自に定義したイベント生起情報を音声認識により取得し,これらの情報を合わせて,より良い構図を維持した自動撮影を可能としている.

岩城らは,映像や音声の解析を行うことで重要なシーンを検出し,自動および手動による シーン順序設定を行うことで,個人映像を対象とした編集を支援するシステムを提案した[10]. このシステムは,人物中心の映像において重視されるポイントであるとして,カメラワーク の切り替えや人物出現,突発的な音や会話音の発生イベントを検出し,それらのイベント発 生タイミングを基に,重要シーンを抽出する.手動編集では,重要シーンを並べ替えること で映像作品をつくることができる.

Smith らの Video-Skims [11] では,まず,映像内の音声から作成したトランスクリプトから TF-IDF 法を用いて重要なキーワードを抽出する.そして,顔検出やカメラワークの検出など を手がかりにしてキーワードを発話した映像区間を認識し,認識した区間情報を利用して要 約映像を制作することができる.これに対して本研究の提案システムでは,自身で定義した 企画内容を基にしたタグを任意の区間に自由に付加できるため,意図した構成の映像を制作 することができる点で異なっている.

Silvia らの MoCA プロジェクト (Movie Content Analysis Project)は, テレビ放送上の映像 や映画を対象とした映像要約を目的とする [12].この研究では,映像をショット(映像の最小

単位)・シーンに分割し,主要人物のクローズアップ,爆発音や発砲音などを手がかりにして 特定のイベントを検出し、これらをつなぎ合わせることで予告編のような要約映像を作成し ている.

これらの手法は,それぞれの「適用分野における重要なシーン」を自動検出し,自動編集 に応用することができる.しかし,それぞれの手法について重要なシーンの定義が異なるこ とからもわかるように,適用分野が限定されるため,汎用性に欠ける.そのため,多くの映 像制作には適用できない可能性が高いと考えられる.一方で,提案システムは演技者が写る 実写映像作品全般に応用できる可能性があり,適用範囲が広いと言える.また,マーカーを 用いたロバストな認識によるリアルタイムのタグ付けを行うため,高い精度でシーン認識を 行うことが可能である.

2.2 メタデータを利用した編集支援手法

Casares らは,メタデータを用いた映像編集支援を行うシステムである Silver を提案した [13].Silver は, Informedia Digital Video Library[14] という映像ライブラリシステムから得た 映像素材を用いる.Informedia Digital Video Library は映像を解析し,シーンに分割したうえ で,各シーンを表すサムネイル画像を生成する.さらに,映像内の音声をテキストに書き起 こしたデータであるトランスクリプトを生成する.これらのデータをメタデータとして映像 に付加することにより,映像や音声の検索性を向上させ,手動編集を支援する.

あらかじめメタデータが付加された映像ソースに対する編集を行うという点で,提案シス テムと類似しているが,企画情報のような映像の設計に関する情報を持っているわけではな いため,自動編集を行うことができない,という点で提案システムとは異なる.

2.3 映像への自動タグ付け手法

映像編集や映像検索の支援を目的とした自動的な索引付けを行う研究が数多く行われている[15,16,17,18,19,20,21].これら全ての研究について,被撮影者による任意の時刻と位置へのアノテーションを行うことは出来ない,という点で提案システムとは異なっている.これらの内,いくつかの研究について以下に紹介する.

Kumanoらは[15],映像文法という独自に定義した映像のルールに基づき,素材として利用 できる区間などの映像の状態遷移を自動認識し,検索を行うための索引を付加する.これに 対して本研究の提案システムでは,撮影前の準備がキーワード入力のみであり,また,編集 モジュールに映像を読み込んだ時点で索引が付加された上で編集された状態となる.これら の点から,エンドユーザにとってより容易に映像オーサリングを行える環境であると言える.

長坂らは,特定の物体画像を検索クエリとして与えることで,その物体が映っているフレームを認識し,自動的にアノテーションを行う手法を提案している[16].提案システムではユーザが定義したキーワードがクエリとなっており,クエリ画像を用意する手間が必要ない.一方で,画像による検索も有用な場合があると考えられる.

Tzi-ckerらは、シーン自動検出と、映像内オブジェクトへのアノテーション支援を可能とす る映像編集システムZodiacを提案した[22].Zodiacは編集操作の履歴を基に、ビデオクリッ プの使用頻度と使用されている映像区間の情報から、画像認識を用いずにシーン検出を行う. また、ユーザがアノテーションしたいオブジェクトに対し、初期値として輪郭を設定するこ とで、そのオブジェクトのトラッキングを行うことにより、アノテーションを実現している.

2.4 拡張現実感技術を用いた映像オーサリングに関する研究

大島らは,映画製作における 3DCG 合成を支援するため,拡張現実感技術を利用した 3DCG モデルのリアルタイムなプレビューを行い,監督やカメラマン,そして演技者が合成結果を 確認しながら撮影する手法を提案している[23].ビデオシースルー型 HMD を装着させ,その トラッキングを行うことにより,演技者の視界に 3DCG を合成する.また,撮影カメラのト ラッキングを行うことで CG の合成位置を取得し,距離情報を利用することで自然な 3DCG 合成を行う.この手法では,任意の位置や時刻へのオブジェクト合成などは考慮していない. また,3DCG を活用した大規模な映画製作を対象としており,高価な機材が必要である.

モシモカメラ [24] は,被撮影者の動きや顔を認識し,それに合わせて撮影中に CG の合成 やエフェクト付加を行う Android アプリである.また,撮影中にタッチ操作等を行うことによ り,操作位置に対応したエフェクト等の編集を行うことができる.エンドユーザを対象とす る点や,任意の座標を利用したリアルタイムな編集が行える点で,提案システムと類似して いるが,ユーザが事前に企画した情報を用いた,シーン構成の自動編集などの機能は有して いないため,素材のトリミングやシーンの並び替えといった編集はユーザが行う必要がある.

第3章 マーカーを利用した映像オーサリング支 援手法

3.1 実写映像作品の制作プロセス

現在の実写映像オーサリングにおける問題を解決するために,本研究では映像オーサリン グの過程に注目した.映像オーサリングの過程は一般的に企画,撮影,編集という3つの大 きな作業フェーズに分けられる[25].とりわけ,映画やテレビドラマなどの映像作品を制作 する場合における各フェーズの大まかな内容は,以下のようなものである.

- 1. 企画フェーズ:構成を決定する
 - シナリオを台本に書き起こす
 - シーンや小道具,構図等を割り出し,絵コンテを書く
- 2. 撮影フェーズ:カメラを用いて映像素材を撮影する
 - 必要な映像をシーン単位で撮影する
 - シーンの切れ目ではカチンコを撮影することによる編集点の記述を行う
- 3. 編集フェーズ:素材を配置し,効果を加えて映像作品として完成させる
 - 編集ソフトウェアを用いてシーンを時間・位置的に配置する
 - 編集ソフトウェアを用いて字幕や図,音声を挿入する

このように分割された制作プロセスであるが,多くの映像オーサリング支援手法はフェー ズ単体を支援するものである.特に最後の編集フェーズのみを対象とする手法が多く,連続 した3つのフェーズを総合的に支援する手法がほとんど存在しない.単体のフェーズを支援 する手法の場合,他のフェーズと連携できる情報が少ないため,ユーザの負担増加の原因に なると考えられる.例えば,企画フェーズではシーン等の映像構成を決定するが,カメラな どの撮影機材に搭載されたソフトウェアや,映像編集ソフトウェアはその企画情報を直接利 用することが出来ない.そのため,撮影フェーズでは企画内容を思い出しながら撮影を行い, 編集フェーズではユーザ自身が撮影した映像素材の内容を確認しながら,企画内容と照らし合 わせて作業を行う必要がある.編集フェーズにおいて自動的にタグなどのメタデータを付加 することで映像素材の検索性を向上させる手法 [15] も提案されているが,企画内容との対応





図 3.1: 絵コンテの一例¹

図 3.2: カチンコの一例¹

付けはユーザが行う必要がある.さらに,その多くの手法ではメタデータ付加の事前準備を 行うために専門的な知識や技術が必要であるため,エンドユーザには適さないと考えられる.

既存の映像オーサリング支援手法における問題点として挙げられる各フェーズ間の情報共 有であるが、それを解決する点で我々が着目したのは、上記の映像オーサリングプロセスの 中で挙げた絵コンテとカチンコである.企画時に描く絵コンテ図 3.1 は、コマ割りされた簡易 的な図と文字列によって、(1) 企画時において制作者が作ろうとしている映像のアイディアを まとめる、(2) 撮影時においてカメラマンやその他のスタッフ、そして被撮影者(役者など)に 制作者の意図を伝える、(3) 編集時において編集者に制作者の意図を伝える、という目的で用 いられるものである.また、映像撮影時に用いられるカチンコ図 3.2 は、「カチン」という音 とカチンコ自体に記載したシーン名などの文字列によって撮影中に映像内へのアノテーショ ンを行う道具である.これにより編集フェーズにおいて映像と音声の同期をとることが容易 になる.さらに、映像を見るだけでそれがどのシーンであるか、といった情報が分かり、編 集が行いやすくなるため、映像と音声の同期が必要でない撮影方法を使用する現在でもカチ ンコは多くの映像撮影現場で用いられている[26].

これらの道具に共通する目的は,各フェーズで発生する映像作品の情報を他のフェーズに関わる作業者と共有することであると言える.これらの道具を用いた情報共有方法は映像オーサリング作業が分担制であるために生まれたものであると考えられる.我々は,映像オーサリングにおいて,これらの道具のような各フェーズ間での情報共有をシステムによって支援し,それぞれのフェーズにおける情報をより多く,素早く記録し,活用することにより,ユーザの労力を削減できると考えた.

¹「映画工房カルフのように」より(URL: http://www.karufu.org/)

3.2 映像オーサリングにおけるマーカーとタグを介した情報共有

上で述べた実写映像作品の制作環境における問題点を考慮し, 絵コンテとカチンコを参考 に, 各フェーズ間の作業情報記録とその共有を行うことで映像オーサリングの全てのフェーズ を通してユーザの支援を行う手法を提案する.映像作品に関する作業情報は映像に付加する ことが出来るタグとして保持し, 一つの作品に関する情報は, タグをまとめて一つのプロジェ クトとして扱う.各フェーズでタグ情報を含むプロジェクトを共有することにより, フェー ズ間の情報共有を可能とする.

映像に対するタグ付けを行う手法として,2次元マーカーの利用を提案する.2次元マー カーを認識した際に得られる ID とタグ情報をあらかじめ対応づけておくことにより,2次元 マーカーを写した時刻に対してタグ付けを行うことができる.また,2次元マーカーを認識 する際に,カメラからの3次元的な相対位置と向きを得ることができる.これをタグに付加 することで,時間的なタグ付けのみでなく,時刻毎の3次元的な座標・向きを利用したタグ 付けが可能となり,それを応用した画像の重畳などの編集を支援することができる.さらに, その認識に特殊なトラッキング装置が必要でなく,カメラと印刷した紙さえあればタグ付け を行うことができることや,他の画像認識や音声認識と比較して高い精度でタグの認識が可 能であることが利点として挙げられる.

マーカーを利用したタグ付け手法はカチンコをメタファとしており,カチンコのように撮影 フェーズでカメラにマーカーを写すという単純かつ素早く行うことが可能な操作により,映 像への視覚的なアノテーションとタグの持つ編集情報のアノテーションを同時に行うことが できる.ここで,マーカーによる視覚的なアノテーションを行う際に,本研究ではタグを表 現する要素として絵コンテ画像を利用する.絵コンテを企画フェーズで描くことで,ユーザ が作品のアイディアをまとめる事を支援すると共に,それをアノテーション時の視覚フィー ドバックとして表示することにより,付加しているタグの把握を支援することができると考 えられる.

本手法の全体的な流れを 図 3.3 に示す.企画フェーズにおいて,(1) ユーザが考えた企画内 容を入力することで,(2) システムはタグとそれを操作する 2 次元マーカーの組を出力する. 次に撮影フェーズにおいて,(3) ユーザがカメラにマーカーを写しながら映像作品の撮影を行 うことで,(4) システムはタグ付けした映像を録画し出力する.最後に編集フェーズにおいて, (5) ユーザがタグ付けされた映像を入力することで,(6) 映像に付加されたタグに含まれる企 画内容に沿った自動編集を行う.

3.2.1 本研究におけるタグ

本研究における「タグ」とは,撮影した映像素材ファイルの特定の映像区間に対して紐付けることができるメタデータである.また,タグはキーワードとなる文字列を含む.これはカチンコに記載した文字列による映像へのアノテーションの効果を狙ったものであるが,カチンコのように画像としての情報だけでなく,文字列としてのアノテーションを行うため,タ



図 3.3: システムの全体像

グを紐付けた映像内のタイミングを,編集時に文字列により検索することが可能となり,映 像の検索性の低さを緩和することができる.

さらに本研究では,タグを「シーンタグ」と「オブジェクトタグ」という2種類に分類して扱う.それぞれの詳細について以下に述べる.

シーンタグ

シーンタグは映像内の一つの連続した場面であるシーンに付加するタグである.ここで, シーンに関するキーワードとマーカーを,以降では「シーンキーワード」,シーンマーカー」 と呼ぶこととする.映像作品は複数のシーンを順序立てて並び替えることによって作成され るものであり,提案システムでは映像中の任意の区間を「シーン」としてタグ付けすること で,シーンへの意味付けや順序付けといったアノテーションを可能とする.このシーンタグ の情報を映像オーサリングの各作業フェーズで更新,共有していくことにより,不必要な部 分のカット編集(トリミング)や,定義した通りの順序にシーンを並び替える編集を自動的に 行うことができる.

シーンタグを付加する際は,シーン開始とシーン終了に対応する2つのシーンマーカーを用いる.図3.4のように,ユーザはシーンXの撮影を始める際にシーンX開始用のシーンマーカーをカメラに写すことで,シーンタグを付加する.そして,シーンXの撮影が終わる際にシーンX終了用のシーンマーカーをカメラに写すことにより,シーンタグの付加を終える.このように,シーンの最初と最後にタグ付けを行うことにより,2つのシーンマーカーが写った



図 3.4: シーンタグ

間の映像区間をシーンXとして認識することができる.この手法を用いることにより,マー カーをカメラに写し続けることなくシーンタグを付加し,通常のシーン撮影やオブジェクト タグの操作などを容易に行えるようにする.

オブジェクトタグ



図 3.5: オブジェクトタグ

本研究におけるオブジェクトとは,3次元座標・向き情報を用いた編集の対象となる要素で ある.オブジェクトタグはオブジェクトを扱うタグであり,提案システムでは撮影中にユー ザの任意の位置にオブジェクトタグを付加することで,そのオブジェクトタグと全く同じ動 きをする画像や部分的なエフェクトなどの編集を容易にする.ここで,オブジェクトに関す るキーワード,マーカーを,以降では「オブジェクトキーワード」,「オブジェクトマーカー」 と呼ぶこととする.

オブジェクトタグを付加する際は、付加したいオブジェクトタグに対応する1つのオブジェ クトマーカーを用いる.オブジェクトXのタグを付加している最中の映像と,付加されるオ ブジェクトタグは図 3.5 のように対応する.図 3.5 左のようにオブジェクトマーカーを写すこ とにより,その3次元位置と向きをシステムが認識する.そして,認識した位置と向きをタ イミングと共に記録することにより、図3.5右のように、オブジェクトマーカーと同じ3次元 位置と向きのオブジェクトタグを映像に付加する.

3.3 提案システムのインタフェースと操作

本研究では、映像オーサリングのプロセスを3つのフェーズに分けてユーザを支援する上 で,それぞれのフェーズに対応するモジュールを用いる.

フェーズ	モジュール
企画	企画モジュール
撮影	カメラモジュール
	マーカー表示モジュール
編集	編集モジュール

表 3.1: 各フェーズとモジュールの対応

提案システムを構成するモジュールは表 3.1 のような対応関係となる. それぞれのモジュー ルのインタフェースとその操作について述べる.

3.3.1 企画モジュールのインタフェースと操作

シーンタグを作成する

企画モジュールは,ユーザの入力するシーンキーワードを基にシーンタグを作成する.シー ンタグを作成する操作を行った直後の企画モジュールのスクリーンショットを図 3.6 に示す. ユーザはシーンタグを作成する際にあらかじめシーンを表すキーワードを考えておき、シー ンキーワード入力フォームに入力する.このキーワードは,タグを識別する他に,編集時の 映像検索に利用する、企画モジュールでは、入力文字列をカンマで区切ることで、複数キー ワードを入力可能とし,柔軟なタグ付けや検索性の向上を図っている.また,キーワードが 一つの場合は入力文字列全体を,キーワードが複数の場合は1つ目のキーワードをシーン名 として扱う.



図 3.6: シーンタグを作成する操作

シーンキーワード入力フォームの右側にあるシーンタグ追加ボタンが押下されると、シー ンキーワード入力フォームに入力されたキーワードとして取得し、そのキーワードから上記 の法則にしたがってシーン名を取得する、そして、取得したシーン名とキーワードを基に、ツ リー型のタグ一覧へシーンを表すノード(以下、シーンノード)を追加し、選択状態にする、 追加位置はツリーのルートの位置とし、追加した順に上から下へ並べる、シーンノードのラ ベルは「シーン:(抽出したシーン名)」とし、ユーザがタグの種類を識別しやすくしている、 また、ノードにマウスオーバーした際にキーワードを表示することで、全てのキーワードを 閲覧できるようにしている、

また,シーンノードの追加と同時に,シーンを表す絵コンテ画像データを作成する.この 画像は,追加したシーンノードをタグ一覧のツリーにおいて選択状態にすることで,図3.6の ように表示される.



オブジェクトタグを作成する

図 3.7: オブジェクトタグを追加する操作

企画モジュールはシーンタグと同様に,ユーザの入力するオブジェクトキーワードを基に オブジェクトタグを生成する.オブジェクトタグ追加した直後の企画モジュールのスクリー ンショットを図 3.7 に示す.

オブジェクトタグを追加する際は、タグ一覧からシーンノードをマウスクリックにより選

択する.すると,図3.7のように,選択中のシーンが表示され,オブジェクトキーワード入力 フォームに文字列を入力することが可能となる.

キーワードを入力し,オブジェクトキーワード入力フォームの右側にあるオブジェクトタ グ追加ボタンが押下されると,ユーザが入力した文字列からシーンタグと同様の手順でシー ン名とキーワードを取得し,それを基にタグ一覧のツリーへオブジェクトを表すノード(以 降,オブジェクトノード)を追加する.オブジェクトノードは選択中のシーンノードの子ノー ドの一番下に追加される.

絵コンテを描く



図 3.8: 絵コンテを描く操作

簡単な絵コンテを使用した映像オーサリング支援を行うため,企画モジュールにおいてタ グに対応した絵コンテ画像を作成できるようにする.タグを作成する操作の説明でも述べた が,タグと同時に絵コンテ画像のひな型が作成され,図3.8のようにタグ一覧において現在選 択中のタグに対応する絵コンテを表示する.ここで,絵コンテ画像にはあらかじめタグの種 類(図3.8 では,[シーン])と,シーン名(図3.8 では,最初のシーン)を描画することで,識別 しやすい絵コンテを素早く描けるようにする.描画はマウスの左ボタンドラッグ操作による, 自由な線描画によって行う.線の色は「線の色」というラベルの右側のパネルを押下して表 示される色選択ウィンドウから選択し,選択中の色はそのパネルに表示される.描画した部 分を消す際は,マウスの右ボタンドラッグ操作により,絵コンテのキャンバスにおける背景 色(図 3.8 では白色)を用いた線描画によって行う.この時の線幅は,左ボタンドラッグによる線と同様である.描画する線の幅はコンボボックスから変更することができる.また,絵コンテを描き直したい,という場合を想定し,絵コンテのリセットボタンを配置する.これらの機能により,単純ながらも自由な描画による絵コンテ画像を作製することができる.

VideoTagPlanningTool Menu	and a second sec	
yeru ①シーンタグ作成 キーワード: ②オブジェクトタグ作成 選択中のシーン:オーブニング キーワード: オブジェクトタグ追加	件成したタグ 一覧 □ <u>レ</u> オブジェクト:顔 ⊕ シーン:解脱 □ ⊥オブジェクト:顔 1	③給コンテ 選択中のタグ:オープニング 左クリック・・・ペン 左クリック・・・消しゴム 線の幅:6 線の色: ■ ^{このモコンテを} リセット
	マーカー数:6/17 (0)	プロジェクト名を入力してプロジェクトファイルを保存 解説動画 保存
	プロジェク を入力	ト名 レスポートボタン を押下

プロジェクトをエクスポートする

図 3.9: プロジェクトをエクスポートする操作

全てのタグを作成し,絵コンテを完成させた後,企画内容を撮影フェーズに伝達するための操作としてユーザはプロジェクトのエクスポートを行う必要がある.プロジェクトをエクスポートする際は図 3.9 に示す様に,プロジェクト名を入力し,エクスポートボタンを押下する.プロジェクトファイルは規定のディレクトリに出力され,以降のフェーズにおいて容易に利用することができる.

3.3.2 カメラモジュールとマーカー表示モジュールのインタフェースと操作

撮影フェーズではカメラモジュールとマーカー表示モジュールという2つのモジュールに より撮影を行う.撮影フェーズにおけるモジュール構成を,図3.10に示す.



図 3.10: 撮影モジュールの概観

マーカーの準備

まず,撮影を行う前にマーカーの準備を行う.本研究では,撮影に必要なマーカーとタグ の情報を保持し,それを関連付けて表示するマーカー表示モジュールを提案する.これを携 帯端末に実装し,撮影に必要な順番でのマーカー表示を行うことにより,マーカーをカメラ に写す操作をより簡単にできると考えられる.また,利用するマーカーが増えた場合,それ ら全てを印刷することは大きな手間であると考えられる.マーカー表示モジュールを用いる ことで,多くのマーカーの印刷を避けることが可能となる.

マーカー表示モジュールを起動すると図 3.11 のようなプロジェクトリスト画面が表示され, 画面左下の更新ボタンを押下することにより,企画モジュールで作成したプロジェクトファイ ルを取得して表示を更新する.また,マーカー画像を作製した際には画面下中央のマーカー 更新ボタンを押下することにより,利用するマーカーを更新することができる.

次に,表示されるプロジェクト名のリストから,企画モジュールで作成したものをタッチで 選択することにより,図3.12に示すようなマーカーリスト画面を表示する.この画面にリス ト表示されている項目は,選択中のプロジェクトにおける撮影操作に用いるマーカーを選択 するためのものであり,一つ一つがマーカーに対応している.一つの項目に表示されている 画像と文字列は,各マーカーに対応するタグの絵コンテ画像とマーカー情報を表す文字列と している.マーカー情報を表す文字列としては,マーカー ID,マーカーの種類(シーン開始, シーン終了,オブジェクト),タグ名,キーワードを並べることで,マーカーとそれを写すこ とによる操作内容の対応をとり易くし,素早くマーカーを選択できるようにしている.通常 の撮影では,このリスト中の一番上の項目を選択することで,最初のシーンの開始操作を行 うマーカーを表示する.





図 3.11: プロジェクトリスト画面

図 3.12: マーカーリスト画面



図 3.13: マーカー表示画面

マーカー画面を表示した様子を図3.13に示す.このマーカー表示画面に表示されるマーカー 画像は、マーカーリスト画面で選択したマーカー画面であり、この表示領域を左にスワイプ することにより、次に表示するべきマーカー画像を表示する.逆に、マーカー画像表示領域 を右にスワイプすることにより、一つ手前のマーカー画像を表示する.マーカー表示領域の 下に表示している画像は、マーカーに対応する絵コンテ画像であり、選択中のマーカーがど のタグを操作するものであるかをユーザに分かりやすく表示する.また、マーカー画像表示 領域の上に表示される文字列は、マーカーの種類とキーワード、表示しているマーカーが全 てのマーカーの中で何番目であるかを表示している.



カメラモジュールを起動する

図 3.14: 撮影前のプレビュー

撮影モジュールは起動されると,ユーザにプロジェクト名の入力を求める.ここでユーザ が企画フェーズで作成したプロジェクト名を入力することにより,定義されたタグや絵コン テの情報を読み込み,カメラに写っている画像のプレビュー表示を開始する.このように,録 画を開始する前にプレビューを表示することで,カメラの位置や角度などを調整することが できる.プレビュー状態では図3.14のように,最初に撮影するシーン名をユーザに提示する.

撮影を開始する

プレビュー中に,1番最初のシーンの開始マーカーをカメラに写すことにより,撮影開始までのカウントダウンが始まる(図 3.15). ここでカウントする秒数は,映画やテレビなどにお



図 3.15: 撮影開始

いて,一般的に用いられる3秒を用いた.撮影開始までに数秒の間をとることにより,ユー ザがマーカーを隠し,撮影の準備を行う時間を確保している.

また,撮影開始前,撮影中に関わらず,シーンマーカー,オブジェクトマーカーを写した位 置には,企画フェーズで作成した絵コンテ画像を映像上に重畳表示する.これにより,ユー ザ自身がどのタグを,どの位置に付加しているのかを把握しやすくする.また,これにより マーカー認識状態を即座に把握することができる.

シーン内容の撮影を行う

企画モジュールにおいて,撮影中のシーンの子要素として設定したオブジェクトは,そのオ ブジェクトマーカーをカメラに写すことにより,写したタイミングに写したマーカーと同様 の3次元座標と方向を持つオブジェクトタグを付加することができる.マーカーはマーカー 表示モジュールを用いることで,次々とスワイプするだけで撮影で必要な順番に表示するこ とが可能であり,それらを順番にカメラに写す操作のみで企画した通りの映像を作製するこ とができる.また,必要に応じて印刷したマーカーを用いることも可能である.撮影中のイ ンタフェースは図 3.16 に示す通りであり,画面には録画状態(録画中か否か)と撮影中のシー ン名を表示する.

シーン撮影を終える

シーンの開始マーカーが認識された後は,撮影中の映像にシーンタグが付加され続ける. ユーザはシーン終了マーカーをカメラに写すことにより,シーンに必要な映像を撮影し終え



図 3.16: 撮影中



図 3.17: シーン以外の区間

たことをカメラモジュールに伝え、シーンタグの範囲を設定することができる.

シーンタグの付加を終えたカメラモジュールは,次のシーンの撮影開始を待機する状態と なる.この時,カメラモジュールのインタフェースは図 3.17の状態となり,シーンの合間で あることをユーザに示す.この時,録画自体は続いており,この映像区間を手動編集にて利 用することも可能である.

シーンの撮り直し

シーン内容を撮影する際には,何らかの失敗をしてしまい,撮り直しが必要となる,ということが多々起こる.とりわけ,本研究の対象であるエンドユーザが撮影を行う場合,撮り 直しの回数はより多くなることが予想できる.そこで,カメラモジュールでは,シーンの撮り直しを機能を実装する.その手順を説明する.

カメラモジュールを用いたシーンの撮影中に何らかの失敗をしてしまった際は,一旦その シーンを終了させるため,シーン終了マーカーをカメラに写す.そして,撮り直しを行うた めに,もう一度同様のシーンを撮影する.これにより,最初に撮影した際のタグ情報は破棄 され,編集で用いるシーンが差し替えられる.

3.4 編集モジュールのインタフェースと操作

3.4.1 プロジェクトファイルの読み込み

編集フェーズでは,ユーザは映像編集ソフトウェアである編集モジュールに撮影モジュール から出力されるプロジェクトファイルを読み込ませる.すると,絵コンテ画像やタグ付けさ れた映像ファイルが読み込まれ,シーンタグが付けられた映像区間のみが切り出される(トリ ミング)と共に,企画モジュールでタグを作成した順番にシーンを並べた状態の編集プロジェ クトがコンポジションという単位で生成される.編集モジュールでは,コンポジション単位 でのリアルタイムな編集プレビューが可能であり,プレビューと同様の状態での映像ファイ ル書き出しが可能である.また,オブジェクトタグが付加された映像区間には,付加された 位置に対応する絵コンテ画像を表示し,撮影されたオブジェクトマーカーの動きを再現する. この絵コンテ画像は非表示に設定することが可能であり,座標のみを利用した編集を支援す ることができる.さらに,タグ付けしたシーンやオブジェクトには,そのタグのキーワード 文字列をエフェクトとして追加する.編集プロジェクトを生成した後の編集モジュールは図 3.18 のようになる.

文字列クエリによる映像内検索

エフェクトとして追加したキーワード文字列を利用し,図3.18のインタフェース左下に示 す検索フォームに入力したクエリ文字列による映像内検索を可能とする.ここでの検索結果 は,タグ付けした映像区間の先頭位置としている.



図 3.18: プロジェクトファイル読み込み後の編集モジュール

手動編集と映像ファイル出力

ユーザは,システムによって大まかに自動編集されたプロジェクトを,編集モジュールの インタフェースを操作して修正することが可能である.例えば,ユーザは「顔」という文字 列のクエリで検索することで,タグ付けされた映像区間の始点を探してタグの座標,つまり 映像中の顔オブジェクトマーカーの座標にCG画像を合成したり,「オープニング」という文 字列のクエリでオープニングシーンの先頭時刻を検索し,その開始時間を前後に調整した後 に映像ファイルとして出力する,といったことが可能である.

3.5 ビデオ制作の流れ

3.5.1 企画フェーズ



図 3.19: 研究紹介映像のイメージ図

ここでは例として,本研究を紹介するデモビデオを作成することを想定し,具体的な制作の流れを説明する.ビデオの登場人物はN君であり,図3.19に示すイメージ図の様に,N君が本システムにおける企画,撮影,編集について,関係する画像を指示棒で指し示しながら説明を行う,とする.シーン構成は以下の3シーンとする.

- 1.「オープニングシーン」最初に始めの挨拶をする
- 2. 「解説シーン」本システムについて解説を行う
- 3.「エンディングシーン」最後に終わりの挨拶をする

また,撮影後に以下のような合成処理を行いたいとする.

- N君は顔を出すのが恥ずかしいので、全てのシーンで顔を「マスク」のオブジェクトで 隠すようにする。
- 解説に用いる画像は、撮影後に合成してビデオに埋め込む.また、埋め込み画像をうまく指示する演技は難しそうなので、解説シーンでは「指示棒」オブジェクトを用いて画像中の解説したい位置を指示するように編集を行うことにする.

これでビデオの企画が固まったので,キーワードを入力する.具体的には,3つのシーンの キーワード「オープニング」「解説」,エンディング」と,解説シーンで用いる指示棒のオブ ジェクト用の「指示棒」,全シーンで顔を隠すマスク用の「マスク」という2つのオブジェク トのキーワードをそれぞれ入力する.すると,シーン/オブジェクトのタグがそれぞれ生成さ れ,またそれぞれに対しマーカーが対応づけられる.例えば,オープニングシーンについて は「オープニングシーン開始」マーカー,オープニングシーン終了」マーカーという2つの マーカーが,指示棒オブジェクトには「指示棒オブジェクト」のマーカーが対応づけられる. そして,それぞれのタグについて絵コンテ画像を描画する.

3.5.2 撮影フェーズ

マスクオブジェクトタグを頭部付近に付加したいため,その付近にマーカーを付けることにする.N君は「マスク」オブジェクトのマーカーを紙に印刷して首の部分に装着する.指示 棒オブジェクトタグやシーンタグの操作には,マーカー表示モジュールを用いることにする.

次に , カメラモジュールを起動して , ディスプレイに表示されるプレビュー画面を見ながら 撮影範囲を調整する、マーカー表示モジュールではマーカーリストから,最初のマーカーで ある「オープニングシーン開始」を選択することでマーカー画像を表示する.これをカメラ に向けて写すと,オープニングシーン撮影開始までのカウントダウンが始まる.そして,カ ウントが0になった瞬間から撮影が始まり,それと同時にオープニングシーンタグが映像に 付加され始める.また,首に付けたマスクのマーカーも認識され始め,マーカー位置にマス クのタグが付加される、マーカーの位置には企画で作成した絵コンテ画像が表示されるため、 タグ付けしているシーンやオブジェクトの内容やタグ付けされる位置,マーカーの認識状態 が把握しやすくなる.オープニングシーンの撮影を行った後,シーンを終了する際には撮影操 作アプリケーション上に表示されたマーカー画像を左へスワイプ操作することにより、オー プニングシーン終了」マーカーを表示させ、それをカメラに写した瞬間にオープニングシー ンタグの付加が完了する、解説シーンではシーンタグの付加操作に加え「指示棒」オブジェ クトのタグを表示させたマーカー表示モジュールを手に持ち,説明している様な仕草をする ことにより,指示棒の持ち手の位置をタグ付けする.エンディングシーンに関してはオープニ ングとほぼ同様の流れでタグを付加しながらシナリオに沿った撮影を行う.このように,基 本的には企画モジュールで作成した順番にマーカーを写すだけで必要なタグ情報を付加する ことができる.

3.5.3 編集フェーズ



図 3.20: 編集モジュールでの自動編集結果

N君がプロジェクトファイルを編集モジュールに読み込ませることで,オープニングシーン のタグが付加された映像区間,解説シーンのタグが付加された映像区間,エンディングシー ンのタグが付加された映像区間,が編集モジュールのインタフェース上に順番に並べられる. シーンタグが付加されていない部分は自動的にトリミングされる.さらに,マスクのオブジェ クトタグの座標と向きに従い,各シーンにおいて絵コンテ画像オブジェクトが追加される.自 動編集を行った結果は図3.20のようになる.平面オブジェクトの座標と向き(回転量)は,そ の後に CG を重ね合わせる用途などに利用でき,イメージした様な映像を完成させることが 可能である.この手動編集の際には,N君は「マスク」という文字列のクエリで検索するこ とで,マスクのオブジェクトタグが付加された映像区間の始点を即座に見つけ,顔の座標に CG 画像を合成したり「オープニング」という文字列のクエリで挨拶のシーンを検索し,その 開始時間を前後に微調整した後に映像ファイルとして出力する,といったことが可能である.

このように,本手法を用いることにより,(a)企画内容をリアルタイムにタグ付け可能とな り,素材の内容確認や企画内容との対応付けの手間を削減できる,(b)被撮影者自身が操作で きるため,追加人員が必要ない,(c)実世界での容易な操作で自動編集した映像を生成できる ため,エンドユーザに優しいだけではなく,作業時間の削減につながる,(d)撮影されたマー カーの様子から3次元座標・向きを取得することができるため,座標・向き情報を利用した CG 合成などの高度な編集を支援することができる,といったメリットが得られると考えら れる.

第4章 マーカーを利用した映像オーサリングシ ステムの開発

3章に示した通り,提案システムは企画モジュール,カメラモジュール,マーカー表示モジュール,編集モジュールから構成され,それぞれのモジュールが連携することでタグ情報 を介した映像オーサリング環境を実現する.本章では,各モジュールの実装について述べる. 全てのモジュールに共通する開発マシンのスペックは以下の通りである.

- CPU : Intel Core i3 540 3.07 GHz
- RAM : 4.00 GB

それぞれのモジュール実装の詳細について以下に述べる.

4.1 企画モジュール

4.1.1 企画モジュールの実装環境

企画モジュールの実装環境は以下の通りである.

- ソフトウェア形態:デスクトップアプリケーション
- 開発環境: Visual Studio 2010
- 開発言語:C#
- 使用したライブラリ:.Net framework
- 使用したソフトウェア: Dropbox

4.1.2 企画モジュールの実装内容

企画モジュールは入力されたシーンやオブジェクトに関するキーワードを基に順序立てた タグ情報,タグに対応した絵コンテとマーカーの情報を含むプロジェクトファイルを生成し, ネットワークを介して他のモジュールと共有する. タグ情報の生成

入力されたキーワードとそこから抽出したシーン名を用い,タグ一覧のツリーにタグノー ドを追加していくことで,タグの情報を保持すると共に,タグの一覧を可能としている.エ クスポートボタンが押下されると同時に,タグ一覧のツリーに保持されたタグ全体を「(プロ ジェクトファイル名).tagproj」という名称の XML ファイルとして出力する.プロジェクト ファイルの出力先は,プロジェクト毎に作成するプロジェクトディレクトリとしている.ま た,プロジェクトディレクトリは Dropbox の共有ディレクトリ内に作成しており,マーカー 表示モジュールにおいて DropboxAPI を利用したプロジェクトデータの共有を可能としてい る.出力されるプロジェクトファイルの一部分をソース4.1 に示す.また,XML ファイルの フォーマットを表4.1 に示す.

ソースコード 4.1: 出力されるプロジェクトファイルの一部分

<projectdata></projectdata>
<formatver>1.0</formatver>
<projectname>解説動画</projectname>
<projectmoviepath>Movie_解説動画.avi</projectmoviepath>
<markerdata></markerdata>
<markerid>0</markerid>
<markername>オープニング</markername>
<markerrollid>1</markerrollid>
<tagid>0</tagid>
<sceneid>0</sceneid>
<keyword>オープニング</keyword>
<markerdatafilepath>tag_data_解説動画_0.td</markerdatafilepath>
<markerimagefilepath>pattImage_A.png</markerimagefilepath>
<markerpattfilepath>pattImage_A.patt</markerpattfilepath>
<contefilepath>conteImage_解説動画_0.png</contefilepath>
•
•

絵コンテ画像の生成

タグの作成と同時に一枚の画像オブジェクトを生成し,タグ一覧ツリー内のノードと対応 させて保持する.エクスポートボタンが押されると,タグ一覧ツリー内の全ノードに対応す る絵コンテ画像オブジェクトを PNG 画像に変換し,プロジェクトのディレクトリに出力する. この時出力される絵コンテ画像ファイル名は、「conteImage_(プロジェクト名)_(タグ ID).png」 とし,プロジェクトファイルの ConteFilePath に設定する.

表 4.1: プロジェクトファイルのフォーマット

XML 要素名	内容
ProjectData	企画モジュール
FormatVer	書式のバージョン
ProjectMoviePath	プロジェクト名
MarkerData	マーカー1つ毎の情報
MarkerID	マーカーを表す ID
MarkerName	対応するタグの名前 (シーン名,オブジェクト名)
MarkerRollID	マーカーの役割を表す ID (シーン開始 , シーン終了 , オブジェクト)
TagID	タグを表す ID
SceneID	シーンを表す ID
Keyword	対応するタグのキーワード
MarkerDataFilePath	フレーム毎にタグの認識情報を記録したファイル
MarkerImageFilePath	マーカー画像ファイルのファイルパス
MarkerPattFilePath	マーカーのパターンデータファイルのファイルパス
ConteFilePath	タグに対応する絵コンテ画像ファイルのファイルパス

4.2 カメラモジュール

4.2.1 カメラモジュールの実装環境

- ソフトウェア形態:デスクトップアプリケーション
- 開発環境: Visual Studio 2010
- 開発言語: C + + (ネイティブ)
- 使用したハードウェア: LogicoolR HD Pro Webcam C910(USB Web カメラ)
- 使用したライブラリ・API: ARToolKit [27] (マーカー認識), DirectShow(撮影・録画), Apache Xerces(XMLパース,記述), FTGL(OpenGL用日本語フォント表示), glpng(OpenGL 用 PNG 画像表示),
- 使用した映像圧縮コーデック: Ut Video Codec Suite

4.2.2 カメラモジュールの実装内容

カメラモジュールは画像の取得と録画,マーカー認識等の機能を同時に行い,それぞれの 処理が比較的処理時間を必要とするものである.そのため,開発言語を C++とした他,処理 速度の高速化を重視した実装を行った.ユーザの使いやすさを考慮し,リアルタイム性を出 すと共に,映像における一般的なフレームレートで録画を行うため,動作フレームレートを 30FPS とした.

プロジェクトファイルの読み込み

企画モジュールで作成したタグの情報や絵コンテ画像を共有するために,プロジェクトファ イルを読み込む機能を実装した.XMLのパースを行うために Apache Xerces を用い,撮影で 出力する映像ファイル名や,マーカー毎の情報(プロジェクトファイルの MarkerData)の読み 込みを行う.

カメラからの画像・音声取得とその録画

映像取得とその録画の処理には DirectShow を用いた. DirectShow は映像や音声の再生,カ メラからの画像取得や録画といった処理を,比較的高速に行うことができる API である¹. DirectShow では,マルチメディアデータの加工処理の1段階をフィルタとして扱う.フィル タは他のフィルタとの入出力を行う「ピン」を持っており,ピンを介して複数のフィルタを 接続したフィルタグラフを作成することでマルチメディアデータへの複雑な処理を行う事が できる.

カメラモジュールでは、プレビュー用のフィルタグラフと録画用のフィルタグラフを実装 し、それらを切り替えることでプレビューと録画の状態を切り替えている.ここで、2つの状 態について、カメラ画像は撮影前・撮影中における撮影状態のプレビューや、ARToolKitの マーカー認識、マーカー認識状態のプレビューに使用するため、フィルタグラフ内から静止 画のデータを格納した配列を取得する必要がある.そこで、この要件を実現するフィルタで ある SampleGrabber を利用した.フィルタグラフ内には静止画データが各フィルタの間を流 れており、その中で必要となる処理を終えた位置に SampleGrabber フィルタを挿入すること により、挿入位置以降のフィルタへの流れを妨げることなく、静止画データの取得が可能と なる.

各フィルタグラフは図4.1 に示す通りに構成した.プレビュー用フィルタグラフは,音声のプレビューを行わず,カメラ画像を取得し,SampleGrabberで静止画データ配列を出力するというフィルタ構成にした.録画用フィルタグラフは,カメラから取得した画像をSampleGrabber によって出力すると共に,SampleGrabberを通過した映像データに圧縮処理を行い,カメラに内蔵されたマイクから取得した音声データと合わせてAviフォーマットに変換し,Aviファイルとして出力する,という構成にした.出力するAviファイルはプロジェクトファイルにおけるProjectMoviePathから取得するファイル名とし,プロジェクトディレクトリに出力している.

¹Introduction to DirectShow Application Programming

⁽URL: http://msdn.microsoft.com/en-us/library/ms786509.aspx)



図 4.1: 各フィルタグラフの構成

ここで、録画時に行う映像の圧縮(エンコード)は、映像ファイルのファイルサイズや映像 ファイル処理時に必要なメモリ容量を小さくし、映像ファイルを扱いやすくするために、必須 であると考える.ただし、エンコードには比較的大きな処理時間を要する.また、エンコード 対象となる映像の解像度が高くなるほど処理時間が大きくなるため、30FPSを維持すると同 時に、マーカー認識率と画質の向上を目的としたより高い解像度の利用を実現するため、エ ンコードは無視できない処理である.そこで、速度面と画質面を考慮し、可逆圧縮形式のコー デックによるエンコードを行うことにした.可逆圧縮コーデックは、圧縮率が低く、出力さ れる映像ファイルのファイルサイズは非可逆圧縮コーデックと比較して大きくなるが、高速 なエンコードが可能、編集時に高速なプレビューが可能、元のデータに完全に復元可能、と いった利点があり、映像編集における中間形式として多く利用されている.我々は、圧縮速 度と出力サイズのバランスを考慮し、カメラモジュールにおいて利用する可逆圧縮コーデッ クとして UtVideoCodec を採用した.また、音声の圧縮は編集時のデコードで問題が起こる可 能性があるため、無圧縮の Wave 形式を採用した.

マーカー認識と認識情報の記録

マーカーによるタグ付けを行うため,リアルタイムにマーカー認識し,認識した情報の記録を行う.高速なマーカー認識を行うため,マーカー認識ライブラリとしてARToolKitを用いた.ARToolKitはマーカー画像とそのパラメータファイルを事前に用意しておき,パターンファイルを読み込むことで,入力される画像から対応するマーカー画像の有無・位置・向きを認識することができる.認識したマーカーの位置は,カメラからの座標変換行列として出力され,描画時に変換行列を適用することによりマーカー位置・向きに合わせて3次元オブジェクトを重ね合わせて表示する,Augmented Reality(AR)の実装を容易とするライブラリである.カメラモジュールにおける認識情報の記録は,このARToolKit から得られるマーカー

の有無と,座標変換行列を1フレーム毎にマーカーデータファイルへ保存することで実現する.保存するマーカーデータファイルは,プロジェクトファイルにおける MarkerDataFilePathから得たファイル名としており,「tag_data_(プロジェクト名_(マーカー ID).td」という形式で, プロジェクトディレクトリに保存する.ここで,マーカーデータファイルは,マーカー1つ に対して1ファイルずつ保存するものとしている.

また,マーカー認識の高速化を行うためにNyARTransMat_C_version¹を用いた.NyARTrans-Mat_C_version は ARToolKit を参考にした AR アプリケーションのためのライブラリである 「NyARToolKit」において実装された,より高速な変換行列計算を可能とする関数群である. これを ARToolKit の同様の機能を持つ関数と置き換えることにより,マーカー認識後に発生 する行列変換が高速化される.

4.3 マーカー表示モジュール

- 4.3.1 マーカー表示モジュールの実装環境
 - ソフトウェア形態: Android アプリ
 - 開発環境: Eclipse
 - 開発言語: Java (+ Android 拡張)
 - 使用した Android 端末: Android Tablet Camangi FM600(H180.3mm × W111.5mm × D11.5mm,画面サイズ7インチ,重さ380g),GALAXY S II SC-02C(H126mm × W66mm × D8.9mm,画面サイズ4.3インチ,重さ120g)
 - 使用したライブラリ:DropboxSDK(ファイル転送)

4.3.2 マーカー表示モジュールの実装内容

実装した Android 端末

2 台の Android 端末に実装したマーカー表示モジュールが動作する様子を図 4.2 に示す.図 4.2 左は Android Tablet Camangi FM600 というタブレット型端末であり,画面左の淵に白色の テープを貼っている.これは,ARToolKit が画像からマーカーを認識する際に,黒色の矩形枠 を探索することにより,端末本体の黒色の枠をマーカーとして誤認識することを防ぐためで ある.比較的大きなサイズであり,マーカーを大きく表示することで,カメラモジュールに よるマーカー認識率を向上させる.一方で,そのサイズと重さは撮影時のユーザによる操作 を妨げる可能性がある.

¹NyARToolkit project (URL: http://nyatla.jp/nyartoolkit/wp/?page_id=219)



図 4.2: Android アプリを起動した様子

そこで,図4.2右のGALAXY S II SC-02C にも実装を行った.GALAXY S II SC-02C は比較的にサイズが小さく,重さが軽いスマートフォン型端末であり,ユーザによる操作性が高いと考えられる.しかし,表示できるマーカーのサイズが小さいため,カメラモジュールにおけるタグ付け操作の精度は低下する.これらの端末は,撮影するユーザの好みと撮影状況によって使い分けることを想定している.

プロジェクトに関する情報の共有

先に述べたように,企画モジュールで作成するプロジェクトディレクトリは,Dropboxの 共有ディレクトリに保存される.マーカー表示モジュールがプロジェクトディレクトリを共 有する際には,Android 端末をネットワークに接続したうえで,DropboxAPIによりDropbox ディレクトリにアクセスし,プロジェクトディレクトリ全体のダウンロードを行う.ダウン ロードしたディレクトリはAndroid 端末内のSDカードに保存し,ネットワーク接続を切断し てもプロジェクトファイルや絵コンテ画像,マーカー画像にアクセスできるようにしている.

4.4 編集モジュール

4.4.1 編集モジュールの実装環境

- ソフトウェア形態:デスクトップアプリケーション
- 開発環境: Eclipse
- 開発言語: Java

• OpenGL Java 用 API: jogl

実装は Javie [28] というオープンソースソフトウェアに機能を追加する形で行った. Javie は Adobe AffterEffects に似たインタフェースを持つ映像編集ソフトウェアである. Java と Java 向 けの OpenGL API である jogl を用いた実装を行っており,動作環境は Windows XP, Windows Vista, Windows 7, Mac OS X など,多くの環境をサポートしている.また,編集中の映像の プレビューに GPU を利用した描画を用いることで,高速なプレビューを実現している.映像 ファイルや静止画ファイル,字幕など多くの要素をレイヤーとして扱い,それぞれのレイヤー に複数のエフェクトをかけることができる.レイヤーの位置や再生時間,エフェクトの各パ ラメータはフレーム単位で値を制御することが可能であり,マウスを用いたパラメータ調整 により自由度の高い映像編集が可能である.さらに,レイヤーを 3D 化することも可能であ り,奥行き方向の移動や3軸の回転を利用した編集が可能である.

Javie における映像編集のワークスペースは Javie プロジェクトとして扱い,コンポジションや画像ファイル,映像ファイルを追加可能である.ここで,コンポジションとは,Javie における一つの映像の単位であり,レイヤーはコンポジションに追加することで,プレビューや映像ファイルへの出力が可能となる.

上記のように,編集ソフトウェアとして完成度が高い点,本研究で用いる映像のトリミン グや並び替えや画像の3次元移動・回転が可能である点,映像ファイルとしてエンコードを 伴う出力が可能である点,エフェクトの開発が可能である点,そして,オープンソースであ る点から,本研究では Javie を拡張する形での編集モジュールの実装を行った.

4.4.2 編集モジュールの実装内容

編集モジュールは企画フェーズと編集フェーズで作成したプロジェクトファイル,映像ファ イル,絵コンテ画像を読み込み,映像の自動編集を行うと共に,キーワードによるレイヤー 開始時間検索を実装し,Javieの基本機能と合わせて手動編集の支援を行う.

プロジェクトの読み込み

Javie のファイルメニューからプロジェクトファイルを選択することにより,そのプロジェ クトファイルに記載された XML を解析し,映像ファイル名やマーカー毎のマーカーデータ ファイル名,絵コンテ画像ファイル名やキーワード文字列などを取得することで,企画情報 や撮影情報の共有を行う.プロジェクトファイルはファイルダイアログから,Dropboxのプロ ジェクトディレクトリを参照することで選択可能である.

次に,全てのマーカーから,シーン開始マーカーとシーン終了マーカーのセット(シーン マーカー)を抽出する.これは,プロジェクトファイルにおける MarkerRollID と SceneID に よって識別することができる.抽出したシーンマーカーのリストはシーンの自動編集に用い る.そして,シーンマーカー以外のマーカーをオブジェクトマーカーとして抽出する.抽出 したオブジェクトマーカーのリストはオブジェクトの自動編集に用いる. また,プロジェクトファイル読み込みの最後に取得した映像ファイルをレイヤーとして読み込み,その映像レイヤーと同様の幅と高さに設定したコンポジションを Javie プロジェクトに追加する.このコンポジションに対し,シーンを表す映像レイヤーや,オブジェクトを表す画像レイヤーを追加することにより自動編集を行う.

シーンの自動編集

シーンマーカーの認識情報からシーンの映像区間を求め,その映像区間以外の部分をトリ ミングし,定義した順番に並び替えることにより,シーンの自動編集を行う機能を実装した. ここでは,その実装を説明する上での例として,プロジェクトファイルを読み込んで抽出し たシーンマーカーのリストから「シーンA」というシーンのシーンマーカーについて,映像 区間を求め,シーンAとしてタグ付けすることを考える.

まず,プロジェクトファイルから取得した,シーン A の開始マーカーのマーカーデータファ イルを読み込み,解析を行う.上で述べた通り,マーカーデータファイルには,マーカー認識 の成否と座標変換行列が記録されている.この内,マーカー認識の成否を全フレームについて 調べていき,その最後のフレームをシーン A の開始フレームとして認識する.次に,シーン A の終了マーカーのマーカーデータファイルを読み込み,解析を行う.マーカー認識の成否 を,先ほど取得したシーンAの開始フレームから順に1フレームずつ調べていき,終了マー カーが認識されたフレームの内,開始フレームに最も近いフレームをシーン A の終了フレー ムとする.上記の方法で開始フレーム,終了フレームの両方が認識できた時,シーン A のタ グ付けが行われたものとし,開始フレームと終了フレームの間の映像区間をシーン A として 認識する.そして,プロジェクトファイル読み込み時に Javie プロジェクトへ追加した映像レ イヤーに対し,シーンAの区間「以外」の部分を非表示に設定することで,シーンAの映像 区間のみを表示する映像レイヤーを作製する.この映像レイヤーに対し,シーンキーワード を付加するため,タグエフェクトを付加する.タグエフェクトとは,我々が実装したエフェ クトであり,画像や音声に対する効果は与えず,キーワード文字列の付加を可能とする.こ のタグエフェクトにはフレーム情報が付加されており,ユーザが入力したキーワードによる 文字列検索に利用することができる.

ここまでが「、シーンA」についてのみのシーンの抽出処理である.そして,これらの処理 を全てのシーンマーカーについて行い,その結果の映像レイヤー群をシーン ID について昇順 に並び替え,コンポジションに追加していくことによりシーンの自動編集を実現している.

オブジェクトの自動編集

オブジェクトの自動編集機能として,オブジェクトタグの位置に絵コンテ画像を設定する 機能を実装した.オブジェクトに対応する絵コンテ画像を画像レイヤーとしてコンポジショ ンに追加し,その絵コンテ画像レイヤーに対し,撮影時にオブジェクトタグが付加された時 刻と位置を設定する.この座標を基準とすることで,オブジェクトタグの3次元座標・向き 情報を利用可能とする.また,絵コンテによりオブジェクトの位置と内容を,視覚的に表現 する.

この処理を行う上で事前に,プロジェクトファイルを読み込んで抽出したオブジェクトマー カーのリストから,親となるシーンが認識された(つまり,親シーンが映像レイヤーとしてコ ンポジションに追加された)オブジェクトのマーカーのみを抽出する.これは,提案システム において,オブジェクトをシーンの子要素として扱うためである.ここからは,具体的な説 明を行うための例として「オブジェクトA」というオブジェクトタグについての自動編集を 行う流れを述べる.

検 索		0:00) 00:11:06) 00:12:00	00:12:2
ソース名		₽*/ケ₽₽₽₽		
▷ 醫力>	ら(いじらないで!消さ)	₽.		
⊿ 🖬 顏		₽ /fx 😡		
⊳	エフェクト			
۵	トランスフォーム			
	③ アンカーポイント	175.0, 175.0, 0.0		
	③ 位置	-332.3, 811.1, 1703.0		
	◎ スケール	📾 <u>35.0%, 35.0%, 35.0</u> %		
	⑤ 方向	0.0* .0.0*68.4*	400000000000000000000000000000000000000	
	() X 回転	0x +36.3*	400000000000000000000000000000000000000	
		0× +351.1*	400000000000000000000000000000000000000	
	⊙ Z ⊡≢∑	0× +90.0*		
	③ 不透明度	0.0%		

図 4.3: 絵コンテ画像レイヤー (オブジェクト) に設定したキーフレームの一例

まず,プロジェクトファイルから取得した,オブジェクトAのマーカーデータファイルを 読み込み,オブジェクトAの親となるシーンが認識された区間の解析を行う.マーカーデー タファイル中の上記の区間において,オブジェクトAのマーカーが1フレームでも認識され ていれば,オブジェクトAに対応した絵コンテ画像を画像レイヤーとして Javie プロジェク トに追加し,それをコンポジションに追加する.絵コンテ画像レイヤーには,オブジェクト Aのマーカーが認識されたフレームに対し,そのフレームで認識された座標変換行列から求 めた3次元座標と方向を設定する.また,オブジェクトAのマーカーが認識されなかったフ レームに対しては,絵コンテ画像レイヤーの不透明度を0%に設定する.

これらの,時間によって変化する座標や方向,不透明度などのパラメータは,Javie においてキーフレームという形で設定することが可能である.キーフレームを設定した絵コンテ画像レイヤーの一例を図4.3 に示す.カメラモジュールで取得したマーカーの動きは1/30sec毎に記録していたが,Javieのキーフレームとして入力する際には1sec毎にしか設定することが出来ない.これをそのまま利用した場合,オブジェクトの動きはカクカクしたものになって

しまう.この問題を回避するため,Javieの機能であるキーフレーム間の補間を利用した.こ れにより,パラメータの値をキーフレーム間で線形補完することが可能であり,オブジェク トの動きをスムーズなものにすることができる.補間を適用したキーフレームは,図4.3にお ける位置や方向といったパラメータのタイムラインに表示されている様に,ひし形のアイコ ンとして示される.また,補間を利用する必要が無い不透明度パラメータについては,補間 なしのキーフレームを設定している.補間なしのキーフレームは,図4.3における不透明度パ ラメータのタイムラインに表示されている様に,正方形のアイコンとして示される.

これらのキーフレーム設定により,撮影時にタグ付けした位置に追従し,タグ付けした時間にのみ表示される絵コンテ画像レイヤーを生成する.そして,この絵コンテ画像レイヤー に対し,シーンと同様にタグエフェクトによるキーワード文字列の付加を行い,オブジェクトAのレイヤー作成処理を終える.

この一連の処理を,全オブジェクトについて行うことにより,オブジェクトについての自動編集を実現している.

キーワード検索

キーワード文字列をクエリとした映像内の時刻検索を実装した.Javie では,レイヤーに適用するエフェクトのパラメータとして,1秒毎に変化させることが可能な文字列を持たせることができる.これを利用し,キーワード文字列を「キーワード」という文字列パラメータとして保持するタグエフェクトを実装した.

また,Javie ではエフェクトインスタンスに対して,特定のパラメータの設定されたフレームを検索するメソッドが用意されている.例えば,レイヤーBのx軸座標情報を保持する「x」というパラメータについて,フレーム検索を行うとする.パラメータxには1フレーム目で0,2フレーム目で1,3フレーム目で2,というように時間推移に伴う値の変化が設定されている.この時,x=2というクエリを検索メソッドに渡すと,レイヤーBのx軸座標が2に設定される最初のフレームである,3フレーム目を結果として得ることができる.このメソッドを利用して,検索フォームに入力された文字列をクエリとし,シーンを表す映像レイヤーとオブジェクトを表す絵コンテ画像レイヤーに付加した,タグエフェクトの文字列パラメータである「キーワード」を対象としたフレーム検索を行うことで,キーワードが付加されたフレームの取得を行う.そして,取得したフレームに対応するコンポジション内時刻を求め,コンポジションのプレビュー映像をその時刻にシークすることで,キーワード検索を実現している.

第5章 評価

提案システムを利用した感想を聴き,今後の改良と発展につなげるための試用実験を行った.

5.1 試用の内容

試用内容は,提案システムによる実写映像作品のオーサリングである.作品のテーマは「自身の行っている研究の紹介」とし,テーマに沿って企画と撮影を行ってもらった.

試用を始める前に研究の概要と,試用内容を説明した.また,各フェーズの作業を始める 前に,そのフェーズで用いるモジュールに関して操作説明を行い,2分程度の練習時間を設け た.企画フェーズでは,ペンタブレットで自由に絵コンテを描画してもらいながら構成を考 えてもらい,実写での撮影が困難な道具を用いる場合は,オブジェクトタグを利用するよう にアドバイスした.撮影フェーズでは,企画内容にしたがい,マーカーを利用してタグ付け しながら映像撮影を行ってもらった.映像オーサリングは自動編集が終わった時点で終了と し,被験者に自動編集の結果を確認してもらいながら,アンケートに回答してもらうと共に, 口頭での意見聴取を行った.

5.2 被験者

試用実験の被験者は、コンピュータサイエンスを専攻する20代および30代の男性2名であった. 被験者Aは映像オーサリングを趣味としており、Adobe Premiere やWindowsMovieMaker など、いくつかの映像編集ソフトウェアに慣れていた.また、映像オーサリングにおける絵 コンテの利用経験もあった.しかし、制作する映像作品の多くは自分自身を撮影したもので はなかった. 被験者Bは、映像オーサリングの経験期間、頻度共に少ない初心者であった.

5.3 試用環境

使用したシステムは4章で示した環境に加え,絵コンテを描きやすいようにするため,ペ ンタブレットを用いた.また,使用する可能性のあるマーカーを事前に印刷し,場合に応じ てマーカー表示モジュールと使い分けるように指示した.



図 5.1: 企画を行う被験者

図 5.2: 撮影を行う被験者

5.4 試用結果

実験を行う様子を図 5.1 と図 5.2 に示す.

5.4.1 企画の結果

被験者 A はオーサリングする研究説明映像において, Wii リモコンで投影方向を操作する ことができる回転プロジェクターの研究を題材とした.回転プロジェクタは Wii リモコンを 向けた方向と同期して回転し,周囲の広い方向へ映像を投影できる.被験者 A は企画フェー ズにおいて,タグを以下のように構成した.

- 1. シーン:オープニング
- 2. シーン: Wii リモコンの動き
 - (a) オブジェクト: バーチャル Wii リモコン
- 3. シーン:回転プロジェクターの動き
- 4. シーン:投影画面の動き
 - (a) オブジェクト:投影画面

2 つ目のシーン:「Wii リモコンの動き」と,4 つ目のシーン:「投影画面の動き」では,撮影に必要なWii リモコンと回転プロジェクターがその場に無かったため,編集時にCGとして合成することを想定し,Wii リモコン」と回転プロジェクターから投影される「投影画面」をそれぞれオブジェクトとして作成した.

一方,被験者Bはオーサリングする研究説明映像において,イス型の入力インタフェース に関する研究を題材とした.イスに座り,左右からのぞきこむような姿勢をとることで,デ スクトップ画面内のウィンドウ同士が重ならないように移動させ,隠れたウィンドウの内容 をのぞき見る,といった操作が可能である.また,イスの上での重心移動をボブスレーの重 心移動に見立て,ボブスレーゲームの操作に利用することができる.被験者Bは企画フェー ズにおいて,タグを以下のように構成した.

- シーン:概要
- シーン:のぞきこみ操作

- オブジェクト:のぞきこむ物体

• シーン:ボブスレーゲーム

- オブジェクト:ボブスレー





図 5.3: シーン : 「Wii リモコンの動き」の絵コ ンテ

次に,被験者 A が作成した絵コンテ画像を幾つか示す.図 5.4 は,オブジェクト:「投影画 面」の絵コンテであり,編集時に置き換える予定である画面のイメージ図として風景画を描 いている.図 5.3 は,シーン:「Wii リモコンの動き」の絵コンテであり,シーンの撮影時に 被験者 A が手に持つWii リモコンの様子を描くと共に,行う動作を矢印で示している.この ように,被験者 A は絵コンテにシーン毎の内容を分かりやすく描いていた.

一方,被験者Bは絵コンテの利用経験が無く,戸惑っていたが,絵コンテを描きながら映像の構成を作っていく様子も見受けられた.被験者Bの絵コンテで興味深かった点は,オブジェクトの絵コンテとして,そのオブジェクトのイメージ図だけでなく,撮影中における操作方法を描いていた点である.これは,タグの内容を分かりやすく描くという点で有効だと考えられる.

5.4.2 撮影の結果

両方の被験者について、シーン開始マーカーを写してから、シーン終了マーカーを写すま での間にシーンタグが付く」という言葉による簡単な説明のみで、戸惑うことなくシーンタ グの付加操作を行うことができた.各シーン1,2回程度の撮り直しがみられたが、カメラモ ジュールの機能により、スムーズに撮り直しを行うことができた.シーンマーカーについて は、この撮り直しの容易さを含め、とても便利であるとの意見をもらった.

オブジェクトタグの付加操作に関しては,シーンタグよりもさらに分かりやすいという意 見をもらい,被験者は迷うことなく操作を行うことができた.しかし,オブジェクトタグは シーンタグと異なり,付加したい区間においてマーカーを認識させ続ける必要があるため,場 合によっては,被験者が意図した位置に付加させ続けることが難しいようであった.

マーカー認識精度に関する問題は,マーカーとカメラとの距離が離れている場合,カメラ に対してマーカーの面を傾け過ぎてしまう場合に発生していた.また,指でマーカーを隠す ことにより認識が不可能になる問題も発生した.これらの問題は,認識精度を向上させる手 法を用いることが最も有効な解決策であると考えられるが,ユーザへの事前説明やフィード バックを行うことで,多少なりとも改善することができると予測できる.そういった意味で, マーカー表示モジュール側のインタフェースにも改善の余地を感じた.

さらに,撮影した映像を見てみると,両方の被験者とも,撮影中の視線がカメラに向いて いない,という状態が見受けられた.この原因として,被験者がカメラで自分自身を撮影す ることに慣れていないこと,そして,ディスプレイへフィードバックされた映像を見てしま うことが挙げられる.後者はシステムとしての問題であり,特にマーカー認識状態を確認し てしまうことが問題になると考えられる.マーカーの認識状態を含む撮影映像のフィードバッ クの方法を改善していく必要があると感じた.

5.5 自動編集後のタグ付き映像

被験者の作例として,自動編集後の編集プロジェクトにおけるオブジェクトタグの時刻に よる変化の様子を図 5.5 に示す.これは,被験者 A が撮影した映像を編集モジュールに読み 込ませて自動編集したものであり,シーン:「Wii リモコンの動き」における,オブジェクト: 「バーチャル Wii リモコン」の様子を示している.

プレビュー中の映像内には,マーカー表示モジュールのマーカー位置に,オブジェクトタ グを表す絵コンテ画像が追加され,表示されている.時刻Aから時刻Bにプレビュー位置を シークすると,マーカー表示モジュールの向きが変更される.図5.5を見ると,このマーカー 表示モジュールの動きに追従し,絵コンテ画像のオブジェクトが移動している様子が見て取 れる.このように,被験者が意図した位置に対するオブジェクトタグの付加が行えたことが 分かる.



図 5.5: シーン : 「Wii リモコンの動き」における,オブジェクト : 「バーチャル Wii リモコン」 の時刻による変化

5.6 アンケートへの回答結果と考察

自動編集した映像作品を編集モジュール上で確認してもらった上で,アンケートに回答してもらった.それぞれのアンケート結果について考察する.

- 提案システムにおける絵コンテの有効性について
 - 被験者 A: 撮影するときに絵コンテがあると,何を撮るのかを意識出来て良い.
 - 被験者 B: あまり上手く利用できなかったが,長い映像をオーサリングする際には
 有効だと思う.

絵コンテについては,不慣れである被験者 B からも有効性を感じてもらえた.一方で,初 心者には絵コンテ描画が困難であった.これは描画インタフェースの実装によるところが大き く,Undo/Redoや,図形描画ツールなどを実装することで,初心者にも絵コンテを描きやす くできると考えられる.また,絵コンテを利用する際に何を描けばいいのかわからない,と いう状態になることも十分考えられるため,書き方の教示も支援すべき点である.

- オブジェクトタグの有用性について
 - 被験者 A: 実物体が無い状態で撮影し, 容易に合成編集できる, オブジェクトタグのコンセプトは良いと思う.

オブジェクトタグを活用できていた被験者 A からは,良い評価を得られた.被験者 B は, 企画する段階で,シーンに適するかどうかを見誤ってしまったために,実際の撮影時に問題 が発生し,オブジェクトタグを活用できなかったものと考えられる.さらに,シーン適する かどうかを見誤ってしまった原因は,企画における絵コンテを活用したシーン構成が上手く いかなかったためだと考えられる.やはり,企画における初心者支援に関しては更なる検討 が必要である.

- シーンタグの有用性について
 - - 被験者 B: 既存システムでは離れた位置にカメラがあると,撮り直しが面倒だった.

 提案システムはマーカーによってシーン管理ができるので,便利だと感じた.
 - - 被験者 A: シーンタグによる自動編集によって,タイムライン上での編集がほとん どいらない点が便利だと思った。
 - 被験者 A: 1 度の録画開始操作で撮り直しを含めた一発撮りができるのが素晴ら しい.

撮影や映像素材の管理に慣れてい被験者 B からだけでなく,映像オーサリングやそのツー ルに慣れ,撮影や素材管理も行ってきた被験者 A からも高い評価を得られた.今回撮影した 作品のシーン数は4つ程度であり,シーンタグを十分活用できるかわからなかったが,撮り 直しが容易である点や自動編集に活用できる点が評価され,有効性を確認できた.

- その他の意見・アイディア

 - 被験者 B: シーンタグ終了時に、マーカーを操作する様子が写らないようにカット する手法を実現できればより良くなると思う。

2つのカメラで撮影した映像間の同期に利用する,というのはとても興味深い意見である. 今回撮影してもらった研究紹介などの映像では,複数カメラにより同時撮影した映像を画面 内に複数並べるような編集技法が利用されることは少なくない.複数カメラの操作は一人で 撮影する場合には煩わしいものであるが,マーカーを利用することで容易に操作できること が予想できる.また,タグ付けを行うことで,同期を図ることや,カメラ間の座標の関係を 求めることも可能となり,さらなる応用が期待できる.

シーン終了マーカーをカメラに写す際の余分な映像については,我々も問題であると感じ, これをカットする手法は必要であると考える.最も単純な案としては,カット対象とする固 定の秒数をあらかじめ決めておくことが考えられる.例えば,カット対象を3秒に設定して おき,撮影中にシーン終了マーカーを写すと,マーカーを写した時刻の3秒前から後ろの映 像をカット対象とする,という手法である.この手法は,カットの精度がユーザに大きく依 存し,エンドユーザを対象とした手法として好ましくないと考えられる.この問題に関して は,より有効な手法の検討が必要である.

第6章 まとめと今後の展望

エンドユーザ向けの映像オーサリングを支援する新しいシステムを提案し,プロトタイプ システムの実装を行った.提案システムは,一般的な映像オーサリングのプロセスとして用 いられる,企画,撮影,編集という3つのフェーズを総合的に支援する.

提案システムでは,顔へのモザイク編集など,特定位置への編集を支援するオブジェクト タグと,シーンとなる特定区間を指定するためのシーンタグという2つの種類のタグを提供 する.撮影フェーズにおいて用いる,マーカーを用いた新しいタグ付け手法と,各フェーズ 間のタグ情報の共有により,映像の半自動編集を実現した.

システムが支援する自動編集の範囲はシーン情報を基にした不要な映像のカット(トリミン グ)と、トリミングされた映像の順序立てた並び替えである.また、オブジェクトタグの位置 へ絵コンテ画像オブジェクトを自動的に挿入し、ユーザへの視覚的な座標提示を行うと共に、 その座標を利用した部分的な編集を可能とする.そして最後に、システムの評価を行い、エ ンドユーザでも素早く簡単に操作可能であり、映像オーサリングにおける新しく有効な手法 であることを確認した.

今後の展望として,マーカー表示モジュールによるテイク管理を可能とすることが考えられる.現在の手法ではシーンを撮り直した際に,最も新しいテイクを使用し,その他のテイクは削除している.これらのテイク全てを保持し,撮影中にテイク管理を行うことで,より容易で自由度の高い素材選択が可能となり,映像制作における労力の削減に貢献できると考えられる.

また,複数カメラへの対応することも発展として考えられる.マーカーによる映像へのタ グ付け手法は,複数カメラの撮影操作や,出力される複数の映像素材の管理を行う上で,有 効に応用できると考える.例えば,2つのカメラに同時にマーカーを写すことにより,2つの カメラの録画を同時に開始したり,2つの映像内に同期させたタグを付加したりすることが考 えられる.このように,複数カメラを利用した撮影,編集をより容易にできると思われる.

謝辞

本論文を執筆するに至るまで,数多くの人にご指導,ご協力をいただきました.

田中二郎先生には研究室の教授として,ご指導をいただきました.研究に関するご指導は もちろんのこと,研究室での計算機管理や就職活動,大学生活に至るまで,多大なご指導を いただくとともに,相談に乗っていただきました.心よりお礼申し上げます.

高橋伸先生には指導教員として,また,私が所属する Ubiquitous チームの担当教員として, 日頃から多くのご指導をいただきました.毎週のゼミや日々の研究活動,就職活動に至るま で,数え切れないほどのご助言をいただきました.論文執筆に関しては,さらに綿密なご指 導とご協力をいただきました.研究活動をここまで行って来ることができたのは,高橋先生 のおかげだと思っております.心より感謝いたします.

三末先生,志築先生には研究室の教員として,ゼミを通じて多くのご指導をいただきました.発表練習などにおいて,的確なご助言をいただき,研究に対する意欲がわきました.心よりお礼申し上げます.

研究室の皆様には研究活動から日常生活まで,非常に多くのご協力をいただきました.皆様には非常に多くの刺激をいただきました.特に,Ubiquitous チームの皆様には,本当に多くのご助言やご協力をいただきました.本論文の執筆に際しては,今まで以上のご協力をいただきました.お世話になりました.本当にありがとうございました.

日常生活を支えて頂いたサークルの友人や,出身校である茨城工業高等専門学校の友人,全 ての友人にもこの場を借りて心より感謝いたします.

画像利用をご快諾頂いた「映画工房カルフのように」主宰オリカワ様,ウッドベース演奏 動画の投稿者である flutewave 様, Javie の利用と開発に関する助言をいただいた, Javie 開発 者の Rakusan 様にもこの場を借りて感謝を申し上げます.

そして,最後にこれまで私を支えてくださった両親と姉,祖父母に,心よりのお礼申し上 げます.本当にありがとうございました.

参考文献

- [1] Windows ムービーメーカー. http://windowslive.jp.msn.com/moviemaker. htm.
- [2] iMovie. http://www.apple.com/jp/ilife/imovie/.
- [3] Adobe After Effects. http://www.adobe.com/products/aftereffects.html.
- [4] Adobe Premiere. http://www.adobe.com/products/premiere.html.
- [5] Motion5. http://www.apple.com/finalcutpro/motion/.
- [6] 尾関基行, 中村裕一, 大田友一. 注目喚起行動に基づいた机上作業映像の編集 (画像・映像 処理). 電子情報通信学会論文誌. D-II, 情報・システム, II-パターン処理, Vol. 88, No. 5, pp. 844–853, 2005.
- [7] 尾関基行, 中村裕一, 大田友一. 机上作業映像のためのイベント駆動型自動編集手法:注 目喚起行動による映像編集の有効性とその検証(映像・マルチメディアとパターン認識・ 理解). 電子情報通信学会技術研究報告. MVE, Vol. 103, No. 209, pp. 7–12, 2003.
- [8] Andreas Girgensohn, John Boreczky, Patrick Chiu, John Doherty, Jonathan Foote, Gene Golovchinsky, Shingo Uchihashi, and Lynn Wilcox. A semi-automatic approach to home video editing. In *Proceedings of the 13th annual ACM symposium on User interface software* and technology, pp. 81–89, 2000.
- [9] 西崎隆志, 尾形涼, 仲村裕一, 大田友一. 会話シーンを対象とした自動撮影・編集システム. 電子情報通信学会論文誌. D, 情報・システム, Vol. 89, No. 7, pp. 1557–1567, 2006.
- [10] 岩城澄, 中沢実, 服部進実. パーソナル映像の自動編集システム. 電子情報通信学会技術研 究報告. MVE, Vol. 102, No. 737, pp. 1–4, 2003.
- [11] M. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding. In *Proc. ICCV98*, pp. 61–70, 1998.
- [12] Silvia Pfeiffer, Rainer Lienhart, Gerald Kuhne, and Wolfgang Effelsberg. The MoCA Project
 Movie Content Analysis Research at the University of Mannheim. Informatik '98, pp. 329–338, 1998.

- [13] Juan Casares, A. Chris Long, Brad A. Myers, Rishi Bhatnagar, Scott M. Stevens, Laura Dabbish, Dan Yocum, and Albert Corbett. Simplifying video editing using metadata. In Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques, pp. 157–166, 2002.
- [14] He Informedia Project, Howard D, Michael G, Yihong Gong, and Alexander G. Lessons Learned from Building a Terabyte Digital Video Library, 1999.
- [15] M. Kumano and Y. Ariki. Automatic useful shot extraction for a video editing support system. In *Proc. MVA*, pp. 310–313, 2002.
- [16] 長坂晃朗,田中譲.カラービデオ映像における自動索引付け法と物体探索法.情報処理学 会論文誌, Vol. 33, No. 4, pp. 543–550, 1992.
- [17] Huitao Luo and Alexandros Eleftheriadis. Designing an interactive tool for video object segmentation and annotation. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pp. 265–269, 1999.
- [18] 宮森恒. 動作インデックスを用いた映像の自動注釈付けとその柔軟な内容検索への応用. 情報処理学会研究報告. データベース・システム研究会報告, Vol. 2002, No. 41, pp. 9–16, 2002.
- [19] 南憲一, 阿久津明人, 浜田洋, 外村佳伸. 音情報を用いた映像インデクシングとその応用. 電子情報通信学会論文誌. D-II, 情報・システム, II-情報処理, Vol. 81, No. 3, pp. 529–537, 1998.
- [20] 金田瑞規, 土橋健太郎, 高木真一, 小舘亮之, 富永英義. 映像編集支援システムのためのインデキシングパラメータ抽出手法の検討. 情報処理学会研究報告. [オーディオビジュアル複合情報処理], Vol. 2001, No. 121, pp. 13–18, 2001.
- [21] 高木真一,池上大介,浜崇之,石川孝明,石川裕也,近藤雅恵,松岡正悟,小舘亮之,富永英義. カメラワーク情報を主なパラメータとする映像オーサリングシステムに関する検討.電 子情報通信学会技術研究報告. CS,通信方式, Vol. 102, No. 516, pp. 69–74, 2002.
- [22] Tzi-cker Chiueh, Tulika Mitra, Anindya Neogi, and Chuan-Kai Yang. Zodiac: a history-based interactive video authoring system. In *Proceedings of the sixth ACM international conference* on Multimedia, pp. 435–444, 1998.
- [23] 大島登志一,黒木剛,小林俊広,山本裕之,田村秀行. 2001 年 mr 空間の旅: 複合現実感技術の映像制作分野への応用.電子情報通信学会技術研究報告. MVE, Vol. 101, No. 389, pp. 7–12, 2001.
- [24] モシモカメラ. http://www.aitia.co.jp/pressroom/20110209-1.html.

- [25] 岡村征夫. 映像編集入門. オーム社, 2010.
- [26] 小川睦. めざせ!映像クリエイター デジタルビデオカメラとパソコンで、キミもフィル ムメーカーになれる. 玄光社, 2005.
- [27] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proc. 2nd IEEE and ACM International Workshop on Augmented Reality*, pp. 85–94, 1999.
- [28] Javie (SourceForge.JP プロジェクトページ). http://sourceforge.jp/projects/ javie/.