

格フレームを用いた動作データ検索と自然言語からのアニメーション生成

國分 剛† 尾下 真樹‡

九州工業大学大学院 情報工学研究科 情報科学専攻†
九州工業大学 情報工学部 システム創成情報工学科‡

〒820-8502 福岡県飯塚市川津 680-4

E-mail: kokusan@cg.ces.kyutech.ac.jp, oshita@ces.kyutech.ac.jp

我々は、自然言語からのアニメーション生成システムを開発している。本システムは、入力文章に対応する動作データを動作データベースから検索し、それらを接続・合成してアニメーションを生成する。我々は、入力文章が主にキャラクタの動きを表す文章に限定されることに注目し、比較的簡単なルールによる意味解析を実現した。格フレームの深層格の概念を用いた付随情報を予め動作データに与えておき、入力文章の解析結果と比較することで、適切な動作データを検索する。また、入力文章中の動詞間の時間関係を表す要素に注目することで、各動詞の間の時間制約条件を抽出し、各動作データの再生タイミングを決定する。実験の結果、基本的な入力文章に応じて、適切なアニメーションが生成されることを確認した。

Generating Animation from Natural Language Text using Case Frame Based Motion Search and Motion Scheduling

Tsuyoshi KOKUBU Masaki OSHITA

Department of Systems Innovation and Informatics Kyushu Institute of Technology
680-4 Kawazu, Iizuka-Shi, Fukuoka, 820-8502, Japan
E-mail: kokusan@cg.ces.kyutech.ac.jp, oshita@ces.kyutech.ac.jp

In this paper, we propose an animation system that converts a natural language input text to an output animation. Since our system only handles action related expressions in natural language, a simplified semantic analysis works well in our system. The system searches for an appropriate motion clip for each verb in an input text from motion database using case frames that are assigned to motion clips in advance. Our system also retrieves the temporal relationship between verbs in an input text in order to generate a time table which represents the execution schedule of corresponding motion clips. A synthesized animation is then generated based on the time table.

1. はじめに

近年、映画やTV等において、3DCGアニメーションが広く利用されている。3DCGアニメーションを制作するには、シーンに登場するキャラク

タの動作データが必要となる。現在、これらの動作データは、モーションキャプチャなどの技術を用いて毎回新しく作成されており、その作成作業には多くの手間や労力がかかる。そこで、毎回動

作データを作成するのではなく、一度使用した動作データをうまく再利用することで、作業を効率化したいという要求がある。しかし、現在は、主に2つの問題により、このようなシステムの実現は困難である。一つ目の問題として、動作データには、さまざまなキャラクタのさまざまな動作があり、大量の動作データを、従来の階層型ファイルシステム、単純なテーブル方式のリレーショナルデータベース等で管理することは困難である。また、二つ目の問題点として、実際にはデータベース内の動作データ同士をうまく接続したり合成したり組み合わせることで適切なアニメーションを作成する必要があるが、既存のアニメーション制作用ソフトウェアは、プロのアニメーター以外には使うのが困難である。

このような背景を踏まえ、我々は、脚本や小説などの文章を入力すると、動作データベースから適切な動作データを検索し、それらを組み合わせ、アニメーションを生成するシステムの開発を行っている[1][2]。本手法では、オブジェクト指向データベースを利用し、オブジェクト指向の継承の考え方をを用いることで、さまざまなキャラクタのさまざまな動作データを効率的に管理する。また、自然言語による文章を入力とすることで、誰でも容易にアニメーションの作成を行うことができる。例えば、監督や演出家が、映画の脚本から簡単なアニメーションを作成することで、演出の参考に使うことができる。近年の映画製作では、従来の絵コンテの代わり、簡単な3DCGを用いて演出を決定するプレビズやアニメティックスと呼ばれる手法が用いられており、本システムを利用することで、これらを容易に作成できる。また、一般の人が映像作品を製作したり、子供向けにお話を動画化したりといった、一般向けの娯楽用ツールとしての利用も考えられる。

現在の自然言語処理技術では、あらゆる文章の意味を正確に解析することは困難である。しかし、本システムでは、入力される文章は、脚本のト書きのような、キャラクタの動きに関する文章に限定される。この点に注目することで、比較的単純なルールにより、入力文章の意味解析を実現した。また、文章中に登場するキャラクタやオブジェクトは、そのシーンにあるものに限られるため、あらかじめシーン情報を与えておくことで、名詞の辞書を簡略化することができる。また、本研究では、各動作データに、格フレームの深層格の概念・□を用いた付随情報をあらかじめ与えておくことで、文章中のそれぞれの動詞に対応する動作データを検索する。さらに、文章からアニメーションを生成するためには、それぞれの動作の再生タイミングを決定する必要がある。この際、時

間的に重なりを持つ複合動作の存在も考えられるが、本研究では、入力文章中の、動作の時間関係を表す要素に注目することで、各動詞の間の時間制約条件を抽出し、各動詞に対応する動作の再生タイミングを表す動作時間テーブルを生成する。この動作時間テーブルに基づいて、前後の動作データをなめらかに接続したり、複数の動作データを合成したりすることで、アニメーション生成を実現する。

第2章では関連研究について述べ、第3章では本システムの概要について述べる。第4章で格フレームを用いた動作検索について、第5章で複合動作スケジューリングについて述べる。第6章で実験と考察を行い、第7章でまとめを行う。

2. 関連研究

近年、情報検索や機械翻訳システム、会話システム等の広い分野において、自然言語処理の技術が用いられている。自然言語処理は、大きく形態素解析、構文解析、意味解析、文脈解析の4つの処理に分けられる。このうち、形態素解析・構文解析は、文章の構造に関する解析であり、現在の技術でほぼ正しく解析することができる。一方、意味解析・文脈解析は、構文解析された結果から文章の意味を解析する処理であり、さまざまな文章の意味を理解するためには、多くの辞書や知識ベース、解析ルールを必要とするため、意味解析を正しく行うことは困難である。情報検索などの応用であれば、構文解析までできれば十分であるため、すでに実用的に使われている。一方、機械翻訳システムや会話システムなどの意味解析が必要となるようなシステムは、まだ実用的に使うことは難しい。対して、本研究では、アニメーション生成に必要な動作に関する文章のみに限定することにより、精度の高い意味解析を実現できる。また、アニメーションに登場するキャラクタやオブジェクトなどのシーン情報はあらかじめ与えられるため、これらを辞書として用いることで、大量の辞書を必要とすることなく解析を行うことができる。

自然言語による文章からCG画像やアニメーション生成するための手法は、これまでも研究が行われている。Coyneらの開発したWordsEye[4]は、入力された文章から静止画を生成するシステムである。この手法では、自然言語解析の結果に基づき、描画すべきキャラクタや物、それらの位置関係、キャラクタのポーズを決定し、それらを3Dオブジェクトによって描画する。この研究では、オブジェクトの場所関係を表現するために、オブジェクトと場所に関する情報を付与している。本研究はこの考え方を参考にして、キ

キャラクタやオブジェクトの場面上的の位置の決定を行っている。また、出力が静止画のため、動作スケジューリングは行っていない。馬場らの研究[5]では、自然言語の入力文章から、入力文章の内容に沿った、キャラクタの移動とオブジェクトの配置を実現するアニメーションを生成する手法を提案している。この手法では、キャラクタの移動による時間の制約条件と、オブジェクトの配置に関する空間の制約条件を用いて評価を行っている。この際、時間制約条件は移動の速度・時間からオブジェクト間の距離を制約するが、本研究では、時間制約条件を用いる部分は共通であるが、動作に対してスケジューリングを行っている部分に違いがある。Wangら[6]は、音声によるコマンドと、マウスによる位置指定に応じて、キャラクタの動作をコントロールできるようなシステムを提案している。本システムでは、単純に指定された動作名に応じて動作を選択しており、本研究の提案手法のように文章や状況に応じた動作データの選択は行われていない。また、入力されたコマンドはすぐに実行されるため、本研究のような動作スケジューリングは行われていない。

3. システムの概要

本節では、本研究で開発しているシステムの構成と処理の流れ、本システムでの自然言語処理、動作データベースについて、順に説明する。

3.1 システムの構成と処理の流れ

図1にシステム全体の構成と処理の流れを示す。入力として自然言語により記述された文章を

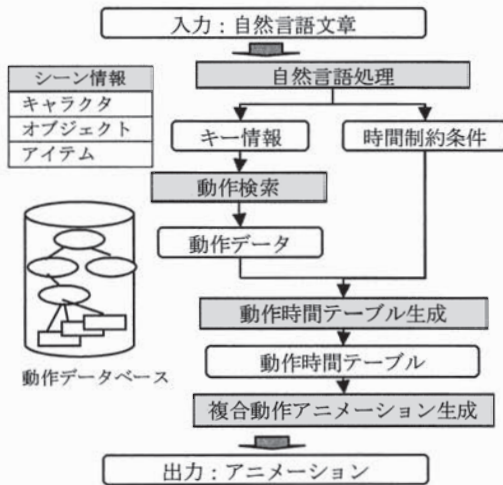


図1 システムの構成図

与えらると、自然言語解析を行い、動作検索に必要なキー情報と、動作スケジューリングに必要な時間制約条件を抽出する。ここで、キー情報とは、格フレームの深層格に基づいて入力文章中の各動詞に付随する情報を表したものである。また、時間制約条件とは、2つの動作間の時間関係を表すものである。これらを抽出した後、キー情報と、動作データベース内の各動作データに付随した動作フレームを比較・評価することにより、入力文章中の各動詞に対応する動作データを検索する。次に、時間制約条件を用いて、検索された各動作データの開始・終了時刻を決定する、動作スケジューリングを行う。動作スケジューリングの結果は、動作時間テーブルとして出力される（出力例は6.1節の図4を参照）。動作時間テーブルの各要素は、動作データと動作主、動作の開始・終了時間の情報を持つ。ここで、時間的に重なりをもつ複合動作を実現するために、複合動作アニメーション生成処理[2]により、動作時間テーブルに従って、動作データ同士の接続や合成を行うことで、全体のアニメーションを再生する。

3.2 アニメーションに特化した自然言語解析

通常、自然言語処理では、形態素解析、構文解析、意味解析、文脈解析の4段階の処理が必要となる[7]。本システムでは、形態素解析、構文解析には、奈良先端科学技術大学の松本研究室で開発された「南瓜」[8]を使用する。意味解析、文脈解析については、本システムに特化した独自の手法を開発した。

2節で述べたとおり、現在の自然言語処理技術では、あらゆる文章の意味解析を正しく行うことは困難である。また、一般に、文中の単語の意味情報を得るために、膨大な量の辞書が必要となる。このような問題に対して、本研究では、入力される文章は脚本などのト書き風の文章であることに注目し、主にアニメーションに関わる動詞のみを考慮する。これより、扱う動詞は、表1のように限定される。

表1 本システムで対応する動詞の種類

動詞の種類		例	対応
動作動詞	外的動作動詞	「走る」「食べる」	○
	内的動作動詞	「思う」「考える」	×
状態動詞		「ある」「知っている」「含む」	×
変化動詞		「着く」「止まる」	△

また、本システムで扱う入力文中の自立語は、ほとんどが、シーンに登場するキャラクターやアイテム等を指す。本システムでは、シーン情報はあらかじめ与えられているため、シーン情報を使用することで、大規模な名詞の辞書を用いることなく、文節内の自立語を判定することが可能である。

3.3 オブジェクト指向を用いた動作データの管理手法

1節で述べたように、多数のキャラクターのさまざまな動作を従来の階層型ファイルシステムや、単純なテーブル方式のリレーショナルデータベース等で管理することは困難である。そこで、本研究では、オブジェクト指向データベースを利用し、オブジェクト指向の継承の考え方をを用いることで、キャラクターとその動作データを効率的に管理する手法を提案する。まず、各キャラクターをひとつのオブジェクトとして扱い、より広いクラスのキャラクターを継承して、より具体的なキャラクターを定義する。例えば、人間の派生キャラクターとして男性と女性、男性の派生キャラクターとして学生や軍人、というように、継承によって階層的に管理を行う。このとき、それぞれのキャラクターの動作データは、各キャラクターのメソッドとみなす。派生キャラクターに新しい動作データを追加すると、基底キャラクターの同じ動作データをオーバーライドすることができる。一方、派生キャラクターに動作データが追加されていないならば、自動的に基底キャラクターの動作データが使用される。そのため、新しいキャラクターを追加するときに、そのキャラクターの全ての動作データを作成する必要はなく、ひとまず追加したい動作データのみを設定すれば良いため、キャラクターの追加を容易に行うことができる。

4. 格フレームを用いた動作検索手法

本節では、入力文章に合った適切な動作データを検索する手法について述べる。例えば、「拾う」という動作でも、単純に動作名だけでは適切な動作は決まらず、軽いものを拾うか重いものを拾うか、立った姿勢から拾うか座った姿勢から拾うかなど、入力文章や状況によって適切な動作は異なる。本手法では、格フレームの考え方をを用い、あらかじめ動作データに付随情報を設定しておくことにより、この問題を解決する。

4.1 動作フレーム

本システムでは、格フレームの深層格の概念に基づき、各動作データの意味的な情報を表す動作

表 2 格フレームの深層格

動作主格	与えられた動作を引き起こすもの
道具格	道具としての役割
対象格	移動や変化する対象物
源泉格	対象物の移動における起点
目標格	対象物の移動における終点
経験者格	ある心理事象を体験するもの
場所格	ある出来事が起こる場所・位置
時間格	ある出来事が起こる時間

フレームを定義する。動作フレームは各動作データに、あらかじめ手作業で与えられる。格フレームとは、フィルモア・□によって提案された、文を意味的な構造として表現するための手段である。格フレームは、各動詞が取り得る深層格と、それに関する制約を記述する。深層格は、動詞に対する各文節の役割を、意味的な関係として捉えた格である。

フィルモアの定義した深層格は8つある(表2)。本研究では、動作データを表現するために、8つの深層格から、経験者格、場所格、時間格の3つを除いた5つの深層格を選び、動作データの情報を表す動作フレームの要素として定義した(表3)。経験者格は、動作としてみたときには動作主格と同じであるため、動作主格として扱う。また、本システムでは、シーン情報や時間はあらかじめ与えられるため、場所格・時間格は考慮しない。

動作フレームの要素(表3)は、大きくオブジェクト条件と姿勢条件に分けられる。ここで、オブジェクト条件は、その動作データの道具格・対象格となるキャラクターやオブジェクトについての条件である。動作フレームに、その動作データが道具格・対象格として取りうるキャラクター・オブジェクトの種類、オブジェクトのサイズ・重さなどの条件をあらかじめ指定しておくことで、入力文章とシーン情報から決定された深層格と一致する場合にのみ、その動作データが使用される。また、姿勢条件は、源泉格・目標格となる動作主の姿勢や手先の位置についての条件である。例えば、「動かす」動作であれば、動かす前のエフェクタの位置(源泉格)や動かす先のエフェクタの位置(目標格)の条件を、適切な動作を検索するために、動作による位置・向きの変化量を使用する。なお、エフェクタとは、動作主格が対象物と接触する場所(主に手の部分)のことを指す。また、前の動作データとなめらかにつながるように、動作開始時の姿勢も評価項目とする。

4.2 動作検索処理

表 3 動作フレームの評価項目

	評価項目	評価項目の説明
	動作名	実行する動作の名前
	キャラクタ	動作を実行するキャラクタ
オブジェクト条件	道具格の直接指定	データベース内のオブジェクト(道具)
	道具格のサイズ	オブジェクト(道具)の幅、高さ、奥行き
	道具格の重さ	オブジェクト(道具)の重さ
	対象格の直接指定	データベース内のオブジェクト、キャラクタ
	対象格のサイズ	オブジェクトの幅、高さ、奥行
	対象格の重さ	オブジェクトの重さ
姿勢条件	初期姿勢	キャラクタの動作開始時姿勢
	位置、向きの変化量	動作によるキャラクタの位置、向きの変化量
	エフェクタの源泉位置	キャラクタのエフェクタの源泉位置
	エフェクタの目標位置	キャラクタのエフェクタの目標位置

本節では、入力文から動作検索を行う部分の処理の流れを説明する。まず、システムに入力された文章は、「南瓜」によって形態素・構文解析される。南瓜の出力結果の例を図 2 に示す。この例では、入力文は 4 つの文節に分解され、各文節の品詞、その読みと原型、活用の種類が出力されている。なお、* で始まる行は各文節の始まりを表し、○D (○の中には数字) の部分は係り先の文節番号を示している。

この解析結果より、入力文章の深層格を決定し、検索のための動作フレーム(キー情報)を決定する。まず、各文節内の自立語の決定を行い、次に各文節の表層格の決定、そして動詞にかかる文節から深層格を決定し、動作フレームの各項目の決定を行う。

4.2.1 文節の自立語の役割決定

文節の表層格・深層格を決定するための前処理として、自立語の役割決定を行う。動詞以外の自立語を持つ文節に対し、シーン情報に問い合わせを行うことで、各自立語の役割決定を行う。自立語は、基本的に、シーンに存在するキャラクタ、または、オブジェクトとなる。しかし、自立語が方向や体の部位の場合、シーン情報とは別に用意した、動作に関連する方向・部位の辞書に問い合

太郎は走って、ボールを蹴る。
* 0 1D 0/1 0.13199898 太郎タロウ 太郎 名詞-固有名詞-人名-名 B-PERSON は ハ は 助詞-係助詞
* 1 3D 0/1 4.69804954 走っ ハシッ 走る 動詞-自立 五段・ラ行 連用タ接続 て テ て 助詞-接続助詞 、 、 、 記号-読点
* 2 3D 0/1 0.00000000 ボール ボール ボール 名詞-一般 を ラ を 助詞-格助詞-一般
* 3 1O 0/0 0.00000000 蹴る ケル 蹴る 動詞-自立 五段・ラ行 基本形 。 。 。 記号-句点

図 2 形態素・構文解析機「南瓜」の出力結果

わせを行い、対応する位置や向き、部位に関して座標や角度、部位の情報を与える。また、助詞「の」についての処理として、例えば「机の上に」という入力があった場合、オブジェクト「机」の「上」という位置情報が付随された形になる。この様にキャラクタやオブジェクトに位置や方向などの情報が付随する場合、これらをセットにして位置や部位の情報として扱う。

4.2.2 表層格の決定

次に、各文節の表層格を決定する。日本語の場合、表層格(格助詞)は「ガ格」「ヲ格」「ニ格」「カラ格」「デ格」「ヘ格」「マデ格」「ヨリ格」「ト格」の 9 個である。ただし、格助詞ではないが主語の助詞「は」の場合も、「ガ格」として扱う。

4.2.3 深層格の決定

表層格に基づき、各文節の深層格を決定する。一般に、それぞれの深層格になりうる表層格の種類はあるが、表層格から深層格を一意に決めることはできない。しかし、本システムでは、入力文がト書き風の簡単な文に限定される点、本システムでは深層格を 4.1 節で述べたものに限定している点により、いくつかの単純なルールにより表層格から深層格を決定出来ると考えた。以下、そのルールを説明する。

- 動詞を動作名とする
ただし、受身の助動詞が有る場合は、能動態の文章になるように表層格を書き換える
(例：太郎が次郎に殴られる→次郎が太郎を殴る)
- ガ格を動作主格とする
ただし、ガ格の文節がない場合は、助詞がない文節の自立語でキャラクタと判断された自立語を動作主格とする(例：太郎座る)
- デ格を道具格とする

ただし、デ格的自立語（オブジェクト）がある任意の大きさを超えるものは目標格とする（例：太郎がドアの前で止まる）

- ・ヲ格を対象格とする
ただし、ヲ格で方向と判断された自立語は、目標格とする（例：太郎が右を向く）
- ・カラ・ヨリ格を源泉格とする
- ・ニ・ヘ・マデ格を目標格とする

4.2.4 キー情報の各項目の決定

深層格の情報から、動作データの動作フレームと比較を行うための、キー情報を作成する。キー情報は、表 3 に示した動作フレームと同様の情報を持つ。オブジェクト条件に関しては、道具格、対象格となったオブジェクトやキャラクタの情報を設定する。道具格、対象格がオブジェクトの場合は、さらに、シーン情報から得られたオブジェクトのサイズ（幅、高さ、奥行き）、重さも設定する。姿勢条件に関して、源泉格に関しては現在のキャラクタの初期姿勢、目標格に関しては動作による位置・向きの変化量と、手足などのエフェクタの位置の情報を評価項目として与える。このとき、4.2.1 節の処理により、源泉格や目標格となるオブジェクトやキャラクタに詳しい位置や部位の情報が付随していれば、その情報に応じて適切な座標を設定する。

4.2.5 動作フレームとの比較・評価

作成したキー情報と、動作データベース中の各動作データの動作フレームとの比較、評価を行い、最適な動作データを検索する。このとき、無駄な計算を避けるために、検索対象となる動作データの範囲は、データベース中の全動作データのうち、動作主の持つ動作データと、動作主の継承元の持つ動作データに限定する。評価方法として、まず、道具格などの直接指定は、オブジェクトが一致していれば 0、していなければ 1 の評価値を与える。姿勢情報に関しては、ルート（腰）の高さ、腰から胸の相対座標、腰から右膝、左膝の相対座標のそれぞれの差分に重みをつけた総和を評価値とする。その他の項目についてはそれぞれ差分を取る。最終的に、以下のような、各評価項目の重みつき 2 乗総和を求める。この評価値 E が最も高い（値が最も小さい）動作フレームに対応した動作データを最適な動作として出力する。また、どの評価項目を重視するかという、評価の重み値は、ユーザーが主観によって決定する。

- $E_0, E_1, E_2, \dots, E_{10}$ 各評価項目の評価値
 $w_0, w_1, w_2, \dots, w_n$ 各評価項目の比重値
 E 総合評価値

n 評価項目の数

$$E = \sum_{k=0}^n w_k E_k^2 = w_0 E_0^2 + w_1 E_1^2 + w_2 E_2^2 + \dots + w_n E_n^2$$

5. 複合動作スケジューリング

本節では、複合動作実現のための動作スケジューリング手法について説明する。4 節の処理により、文章中の各動詞に対応する動作データが、データベースから検索される。アニメーションを生成するために、このそれぞれの動作データをどのような順序・組み合わせで再生するかを決定する必要がある。そこで、入力文章中の、動作同士の再生タイミングに関する要素に注目し、複数の時間制約条件を抽出する。そして、抽出した時間制約条件に基づき、各動作データの再生時間を決定する。

5.1 時間制約条件

本手法での時間制約条件は、2 つの動詞の間の時間関係を表すもので、関連する前後の動作と、それらの時間的な関係を表す情報を持つ。動作の時間的な関係を、大きく以下の 3 つに分類する。また、遷移・接続の場合、前後の動作間の時間間隔の情報を持つ。

- ・ 遷移
前の動作から後の動作へなめらかに遷移する
- ・ 接続
前の動作の後、しばらくしてから、後の動作を行う
- ・ 同時
2 つの動作を同時に行う

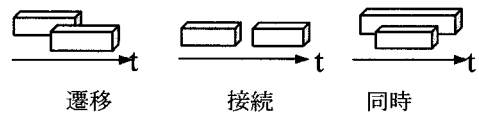


図 3 動作の時間的な関係

5.2 時間制約条件抽出ルール

本手法では、4.2 節で説明した入力文章の形態素・構文解析の結果から、接続助詞・句読点や、動作のタイミングを表す副詞句に注目することで、時間制約条件を抽出する。この際の条件として、大きく三通りに分けられる。

例えば、「走る。ボールを蹴る。」という文章であれば、それぞれ独立した別の動作に思えるが、「走ってボールを蹴る。」という文章ならば、一

「走ってボールを蹴る。」という文章ならば、一連のなめらかな動作になると考えられる。このような、同じ動作でも文章の要素により結果が変わるという考え方により、以下のルールを設定した。ここで、ルール中の矢印は、文節が左側から右側へ係り関係を持つことを表す。

- 動詞の文節どうして係り関係を持つ場合
 - ・ 動詞A + 接続助詞「て」の文節
 - 動詞B + 任意の助詞の文節
 - A から B へ遷移の条件
 - ・ 動詞A + 任意の助詞 + 読点「、」の文節
 - 動詞B + 任意の助詞の文節
 - A から B へ遷移の条件
 - ・ 動詞A + 接続助詞「ながら・つつ」の文節
 - 動詞B + 任意の助詞の文節
 - A と B の同時の条件
- 係り関係を持つ動詞の文節の間に、後の動詞の文節に係る副詞句がある場合
 - ・ (動詞Aの文節、「前に」の文節)
 - 動詞Bの文節
 - B から A へ遷移の条件
 - ・ (動詞A + 任意の助詞の文節、「すぐに」「すかさず」「急に」...の文節)
 - 動詞Bの文節
 - A から B へ遷移の条件
(遷移時間は短い)
 - ・ (動詞A + 任意の助詞の文節、「〇秒後に」の文節)
 - 動詞Bの文節
 - A から B へ接続 + 秒数指定
- 係り関係を持たない場合
 - ・ 動詞A + 任意の助詞 + 句点「。」の文節
(文末)
 - 文章が続く場合は次の動詞と接続の条件
(以降に文章がない場合は条件を出力しない)

5.3 動作時間テーブルの生成方法

抽出された時間制約条件より、各動作の順序・組合せを決定し、各動作の開始時刻を決定する。

まず、全ての時間制約条件から、一番最初に実行する動作を見つけて、これを最初の動作とする。次に、最初の動作に関連する時間制約条件を探索し、その時間制約条件より、次に実行する動作を決定する。この繰り返しにより、すべての動作を実行する順番に並べる。各動作の順番が決定したら、動作間の時間制約条件、各動作に対応する動作データの動作時間より、各動作の開始・終了時間、遷移・接続の場合はその時間を決定する。実際には、動作間の遷移・接続時間は、複合動作ア

ニメーション生成処理において、動作接続・合成手法に基づき、出力結果が自然なアニメーションになるように調整される[2]。この処理において、矛盾する時間制約条件が存在することがある。このとき、同じ2動作間に接続助詞・句読点と副詞句の両方の時間制約条件が見つかった場合、副詞句による条件を優先する。同タイプの条件が同じ動作間に複数出現する場合も考えられるが、現在のルールでは対応していない。

上記の処理により、文章から時間制約条件に基づき、動作スケジューリングを行うことができる。しかし、複数のキャラクタが同時に動作を行うという場合には、現在のルールでは対応することができない。例えば、「太郎と花子は一緒に踊る。」や「次郎は三郎に皿を渡す。」などといった場合である。原因は、複数の動作主格に対応できていないことと、本来の動作主格以外のキャラクタが動作を行う必要がある場合であることが考えられる。前者については動作主格を複数に対応させることで解決できると考えている。後者の場合は、対になる相手の動作を自動的に用意することで解決できると考えている。

6. 実験

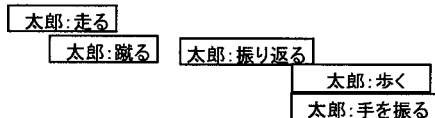
6.1 実験内容・結果

本節では、今回行った実装とテストの結果を示す。本システムの検証方法としては、映画の脚本などの既存の文章を入力として与えて、どの程度の割合の文章を正しく解析できるかを評価していきたいと考えている。しかし、現段階ではプログラムの実装やデータの準備が十分ではないため、今回は、これまでに実装したプログラムをもとに、本論文で提案した手法の動作テストを行った。動作データに関しても、本来は数人のキャラクタと100種類程度の動作データが必要と思われるが、今回は最低限のテストのために、1人のキャラクタと20種類の動作データを手動で登録したものを使用した。本システムは、C++を用いて実装している。現在は、データベースシステムは使用せず、動作フレーム情報は単純にメモリ上に格納している。また、形態素・構文解析のために、「南瓜」のC++ライブラリであるlibcabochoaを用いている[8]。

今回行った動作テストとして、いくつかの単純な文章を入力し、期待通りの動作時間テーブル出力されることを確認した。その例のひとつとして、以下のような例文を与えたときの、プログラム中の中間出力を表4、動作時間テーブルの出力結果と、それによって生成されるアニメーションのキャプチャ画像をそれぞれ図4、図5に示す。

入力文：

「太郎は走ってボールを蹴る。太郎は振り返り、歩きながら手を振る。」



入力文章中の5つの動詞について、それぞれ深層格を決定してキー情報を作成し、動作検索を行って、それぞれの動詞に対応する適切な動作が検出された。蹴る動作は、下段キックや中段キックなど、複数の動作がデータベースに登録されているが、ここでは対象格のボールに位置に応じて適切な動作が検索されている。時間制約条件についても、5.2節で示したルールに基づき、4つの条件を出力し、動作時間テーブルを出力した。図4は、時間軸上に配置された各動作データを表している。

図5 動作時間テーブル出力結果

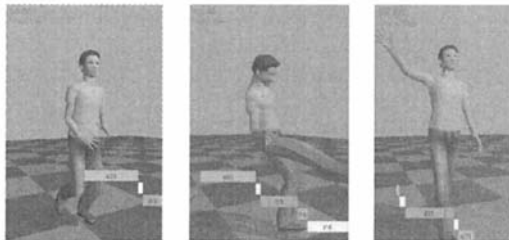


図4 アニメーション生成結果

6.2 考察

実験より、簡単な入力文章を与えたときに、提案手法によって、接続助詞および句読点によるスケジューリングが行われることを確認した。ただし、副詞句に関するルールについては、まだ実装を行っていないので、今回はそれ以外のルールが実装されたプログラムによる結果を示した。

また、問題点として、5.3節で挙げた問題点と、「名詞+する」の形の場合、この文節を動作と判断することが出来ないという点が挙げられる。例えば「ガッツポーズする」といった場合である。対応策として、「名詞+する」の場合、名詞を動作名とするルールを追加することで解決を考えている。

表4 プログラムの中間出力

<ul style="list-style-type: none"> ➤ キー情報1 ◇ 動作主格：太郎 ◇ 動作名：走る 	<ul style="list-style-type: none"> ➤ 時間制約条件1 ◇ 動作A：走る ◇ 動作B：蹴る ◇ 制約：遷移
<ul style="list-style-type: none"> ➤ キー情報2 ◇ 動作主格：太郎 ◇ 動作名：蹴る 	<ul style="list-style-type: none"> ➤ 時間制約条件2 ◇ 動作A：蹴る ◇ 動作B：振り返る ◇ 制約：接続
<ul style="list-style-type: none"> ➤ キー情報3 ◇ 動作主格：太郎 ◇ 動作名：振り返る 	<ul style="list-style-type: none"> ➤ 時間制約条件3 ◇ 動作A：振り返る ◇ 動作B：歩く ◇ 制約：遷移
<ul style="list-style-type: none"> ➤ キー情報4 ◇ 動作主格：太郎 ◇ 動作名：歩く 	<ul style="list-style-type: none"> ➤ 時間制約条件4 ◇ 動作A：歩く ◇ 動作B：振る ◇ 制約：合成
<ul style="list-style-type: none"> ➤ キー情報5 ◇ 動作主格：太郎 ◇ 動作名：振る ◇ 対象格：手 	

7. まとめ

本論文では、自然言語からのアニメーション生成システムとして、格フレームを用いた動作検索手法と、時間関係に注目した意味解析による動作時間テーブルを用いた動作スケジューリング手法を提案した。今後の課題として、副詞句によるルールの実装と、不足ルールの追加、実現が難しい文章の調査・検討が挙げられる。

参考文献

- [1] 寺崎卓也, 上川千絵, 尾下真樹, “オブジェクト指向による動作データの管理手法と自然言語からのアニメーション生成システム”, *Visual Computing* 2004, pp. 197-202 (2004).
- [2] 森脇卓也, 國分剛, 尾下真樹, “自然言語からの仮想人間の複合動作アニメーションの生成”, *情報処理学会研究報告*, 2006-CG-124, Vol.2006, No. 91, pp. 67-72 (2006).
- [3] チャールズ J. フィルモア-著, 田中春美, 船城道雄-訳, 「格文法の定理」, 三省堂, (1975)
- [4] Bod Coyne, Richard Sproat, “WordsEye”, *Proc.of ACM SIGGRAPH* 2001., pp. 487-496 (2001).
- [5] Hiromi BABA, Tsukasa NOMA, Naoyuki OKADA, “Visualization of Temporal and Spatial Information in Natural Language Descriptions”, *IEICE Transaction on Information and Systems*, Vol. E79-D, No. 5, pp. 591-599 (1996).
- [6] Zhijin Wang, Michiel van de Panne, “Walk to here”: A Voice-Driven Animation System, *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* 2006, pp. 243-250 (2006).
- [7] 乾健太郎, 白井清昭, “例文を使って文を解析しよう”, *情報処理学会論文誌*, Vol.41, No.7, pp. 763-768 (2000).
- [8] 工藤拓, 松本裕治, “チャンキングの段階適用による日本語係り受け解析”, *情報処理学会論文誌*, Vol43, No.6, pp. 1834-1842 (2002).