

## 3次元仮想空間における時間の可視化について

山崎 晃 嗣<sup>†</sup> 木村 洋<sup>††</sup>  
松岡 裕 人<sup>†</sup> 原田 育 生<sup>†</sup>

アニメーション技術の発展によって、3次元CG映像の需要が拡大している。映像編集におけるモーション機能が高度化する一方、その編集における操作性が複雑化しており、従来はオーサリングツールに熟練する必要があった。本報告では、より簡便に映像生成することを目的とし、新たにタイムラインオブジェクトという様々な運動パラメータの時間変化を可視化するオブジェクトを導入し、3次元空間上だけでモーション編集が可能となるインタフェースを提案する。現在、独自の3次元モーションオーサリングソフトを開発しており、本ツール上での実現を目指して検討中の手法について報告する。

### About Visualization of the Timeline Object in the 3D Virtual Space

KOJI YAMAZAKI,<sup>†</sup> HIROSHI KIMURA,<sup>††</sup> HIROTO MATSUOKA<sup>†</sup>  
and IKUO HARADA<sup>†</sup>

Recently, animation technology expands the demand of 3-D CG movies. The more motion control technologies are getting sophisticated, their operability is getting more complicated. Thus, we should have been required to have highly expertized knowledge and skill on authoring tools. In this paper we propose an effective interface for authoring animations by introducing an object called *TimelineObject* which visualizes various motion parameters. The timeline object is easy to edit directly in the 3-D virtual space to control 3-D objects along with time sequence. We describe the new method being implemented in our original 3-D motion authoring software under active consideration.

#### 1. はじめに

近年、科学技術分野の利用にとどまらずテレビや映画など実写映像での応用まで、コンピュータグラフィックス（以下CG）の普及は著しい。その大半が何らかの動き伴うCGアニメーションによる表現である。しかし、魅力的な映像を作るためには、専用編集ソフトウェアへの相応の習熟が求められる。

そこで本報告では、3次元空間において被写体やカメラの多様な運動パラメータの時間変化を視覚化する機能を持つオブジェクトをタイムラインオブジェクトと名付け、3次元オーサリングツールのモーション編集 (motion authoring) における、より直感的な方法として提案する。<sup>1)</sup>

従来、3DCGのアニメーションを構成する対話的なインタフェースとしては、タイムラインやファンク

ションカーブが用いられてきたが、編集時のモーションのイメージが把握しにくいという問題がある。本手法では、3次元シーンを構成する他の要素と等価に3D空間上に可視化される仮想オブジェクトを用いることで、被写体やカメラの運動イメージを見ながら簡単に編集することができる。

我々はこれまでイメージベース (image-based) 手法を用いたCGの3次元仮想空間レイアウトシステム WebReality<sup>2)</sup>を開発しており、本システムへの実装を検討している。

#### 2. 従来システム

3次元仮想空間上でアニメーションを構成する伝統的な方法として、キーフレーム法による中割り<sup>3),4)</sup>が定着している。キーとなる特徴的なモーションの状態を指定することで、中間フレームはスプライン関数で自動的に補間される。補間された値をグラフの曲線で示したものはファンクションカーブと呼ばれる。このとき、キーに対応するフレームを指定するためにタイムラインが用いられる。

<sup>†</sup> NTT マイクロシステムインテグレーション研究所  
NTT Microsystem Integration Laboratories

<sup>††</sup> 日本情報通信

Nippon Information and Communication Corporation

また、特にカメラのモーション (カメラワーク) を構成する場合、カメラの軌道を曲線等で描くことで、向きや姿勢を変化させながらアニメーションを構成する手法もよく使われる。この場合も、モーションパス上のカメラや被写体の開始キー・終了キーはタイムラインに並んだフレームに対応付けられる。

タイムラインは時間軸に沿って対象とするシーンを統一的に管理できるため、3DCG のアニメーション編集<sup>5),6)</sup> だけでなく、2 次元映像のノンリニア編集においても標準的なインターフェースとなっている。しかし、このように 3 次元空間とは独立した 1 次元の時間要素の編集が必要なインターフェースは、各映像カットの関係を把握するには便利であるが、3 次元空間上でモーションを編集する際には、オブジェクトの時刻毎の運動の変化を視覚的にとらえ難いため、望みの映像を作成できるようになるまでに一定水準の熟練を要する。

例えばキーフレーム法であれば、物体の時間位置における変化を数値指定したり、同様に、3 次元シーンとは分離したファンクションカーブの画面を開いて一つ一つ編集を行わなければならない。モーションパスを描くならば、軌跡によってオブジェクトが移動する様子は容易に理解できるものの、回転量や速度等といった運動の変化を直感的に把握することは難しい。

これらの編集作業を容易にするためには、必要に応じて回転量や速さなどのパラメータを適切に視覚化するとともに、同一の 3 次元インターフェース上で直接編集できるようにする必要がある。

従来の 3 次元空間における運動の視覚化手法としては補間されたパラメータを 3 次元形状の変化としてマッピングするゴースティング法<sup>7)</sup> が挙げられるが、運動の変化を都度形状の変化に置き換えて描画しなければならないため、形状が複雑になればなるほど高い計算コストが要求される。

そこで本報告では、タイムラインオブジェクトによって運動パラメータを簡便に可視化する方法を提案し、オブジェクトの様々な運動パターンに従って予め用意された編集環境を提供する。

### 3. 時間の可視化手法

#### 3.1 タイムラインオブジェクトの定義

図 1 に本システムにおけるモーションのデータ構造をシーングラフで示す。3 次元空間における時間  $t$  と同時に、移動・回転・速度など様々な運動の単位時間あたりの変化量であるモーションパラメータ  $mp$  を定義可能な仮想オブジェクトを、タイムラインオブジェクト  $TimelineObj(t, mp)$  とする。

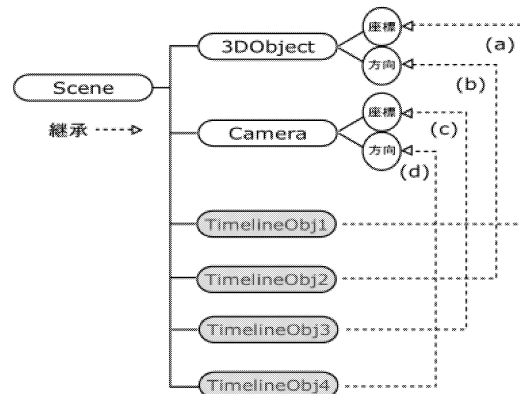


図 1 継承-タイムラインオブジェクトと静止物体の関連付け  
Fig. 1 Inheritance - association with *TimelineObject* and static object

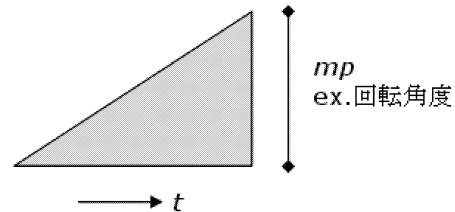


図 2 タイムラインオブジェクトの模式図  
Fig. 2 A view showing a frame format of *TimelineObject*

また本稿では、タイムラインオブジェクトによって時間が 3D オブジェクトやカメラ等静止物体の物理的な属性 (座標・方向) に関連付けられることを継承と定義するが、いわゆるオブジェクト指向で言う継承とは異なる。図 1 のように、3D オブジェクトやカメラにタイムラインオブジェクトを継承することで、3D オブジェクト (3DObject)、カメラ (Camera) に動きを与えることが可能となる。

従来は、時間と運動の変化を表すオブジェクトは可視化されず、タイムラインが静止物体と時間に関連付け、ファンクションカーブが静止物体と運動の時間変化を対応付けていた。いずれも 3 次元空間とは分離し 1 次元に時間展開をしたインターフェースであったため編集集中のモーションのイメージが把握しづらいのに対し、本提案手法では運動の時間変化を 3 次元空間上で直接参照することが可能である。タイムラインオブジェクトの模式図を図 2 に示す。

グラフとして考えるならば、一見従来のファンクションカーブと同様の物に思われるが、本手法では、静止物体に時間  $t$  と運動  $mp$  の両方を関連付けできる仮想オブジェクトとして 3D 空間上に直接描画される。

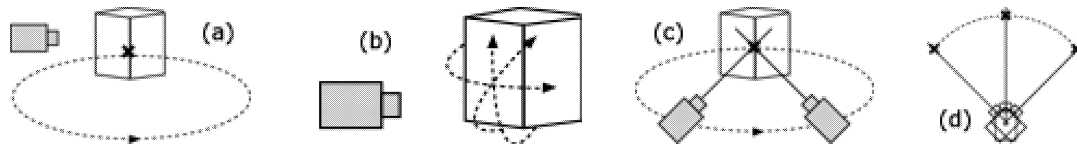


図3 継承によるオブジェクトの振り舞い  
Fig.3 Object motion by inheritance mechanism

### 3.2 オブジェクトの振り舞いの定義

タイムラインオブジェクトは、それが継承されたオブジェクトの属性に応じて定義された意味づけに従ってオブジェクトの動きを制御する。タイムラインオブジェクトの継承を受けるオブジェクト属性によって時間と運動が付与される。この様子を図3(a)~(d)に示す。また、図3中の(a)~(d)は以下の継承例(a)~(d)に対応する継承を表す。

- (a) 3DObjectの座標：3Dオブジェクトの移動
- (b) 3DObjectの方向：3Dオブジェクトの回転
- (c) Cameraの座標：カメラの移動
- (d) Cameraの方向：カメラアングルの変化

まとめると、タイムラインオブジェクトは、

- 図形形状として時間  $t$  とモーションパラメータ  $mp$  を定義する
- 継承機構によって静止オブジェクトの運動を規定する仮想オブジェクトである
- 他の静止オブジェクトと等価に3次元空間上に可視化され、操作できる

という特徴をもつ。以上により、ビデオ画像に再構成するための多次元の情報を、同一の3次元空間上で編集することが可能となる。

### 3.3 タイムラインオブジェクト間の継承

図1で示した継承の考え方は、もともと時間概念を持たない静止物体への継承であったが、時間概念を持つ物同士の継承である本機構により、回転しながら移動といった複合的なモーションの可視化が可能となる。図4に、シーングラフを示す。

タイムラインオブジェクト間の継承は、タイムラインオブジェクトの視覚的な次元の拡張として定義する。

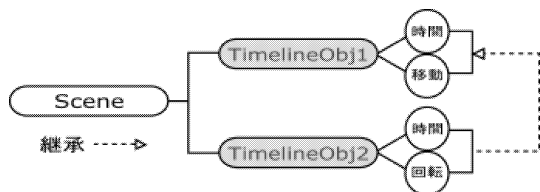


図4 タイムラインオブジェクト同士の継承  
Fig.4 Interrelation of *TimelineObjects*

例えば、3Dオブジェクトとして地球が太陽の周囲を公転しながら自転する運動を構成する場合、移動パラメータだけでなく回転パラメータも継承する必要がある。図4では、*TimelineObj2*を*TimelineObj1*に継承することにより、*TimelineObj1*の視覚的な状態が遷移する。すなわち、新たに加えられた動作によって*TimelineObj1*の形状や色が変化する。

この定義をする理由としては、静止オブジェクトに対して継承する動作が1:1の場合と1:nの場合で、可視化方法を変える必要があるためである。具体的に視覚化する方法は次章のモーションのテンプレートで整理し、5章で具体例を示す。

## 4. モーションのテンプレート

オブジェクトの典型的な運動パターンを抽出し、直ちにモーション編集に利用できる環境を本稿ではモーションのテンプレートと呼ぶ。時間の関数としての運動パラメータは移動や回転といった6次元の基本モーション以外に、サイズの変化や速度・カメラのレンズアクション等様々なタイプを考えることができる(表1)。また、これらを複合的に組み合わせることで多彩な運動パターンを構成することができるが、極めて編集の自由度が高い3次元空間において効果的なモーション編集を行うには、利用頻繁の高いパターンを予め整理し再利用性を高めることでユーザの負担を軽減できる。そこで、これらをモーションプリミティブと定義し、モーション編集の支援を行う。

表1に挙げるパターンで、基本的な動きはほぼカバーできる。例えば、カメラワークにはドリー・トラック・パン・ティルト・ズームの5つの基本動作がある<sup>?)</sup>が、カメラ位置を水平垂直・前後左右に移動した

表1 モーションプリミティブのテンプレート化  
Table 1 Motion patterns of each transition, rotation, scaling, velocity and optics

モーションタイプ	プリミティブ	可視化方法
移動 (transition)	直線/楕円/弧/曲線/螺旋	軌跡
回転 (rotation)	回転/旋回	円柱/帯のねじれ
規模 (scaling)	拡大/縮小 (サイズ)	濃淡の変化
速度 (velocity)	加速/減速	帯幅/濃淡
光学 (optics)	拡大/縮小 (視野角)	色の変化

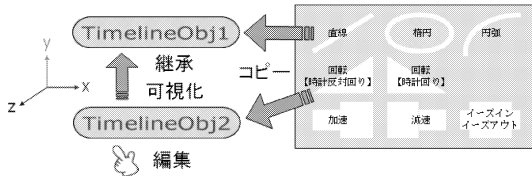


図 5 モーションプリミティブのパレット  
Fig. 5 Tool palette of the motion primitives

り、アングルを水平・垂直に振ったり、というようなアクションで容易に実現できる。必要に応じてカメラの光軸基準に回転するモーションやレンズの画角  $fov$  を変化させることもできる。

単純な移動のみであれば従来の 1 次元のモーションの軌跡で表現可能であるし、位置固定の回転運動であれば、円柱の運動で回転を可視化することができる。運動の複合的な状態を可視化する場合、それらとは違うパターンを用意し、回転しながら移動する運動は帯のねじれで表現するという具合に描画パターンが対応づけられる。

図 5 に、テンプレートの利用イメージを示す。プリミティブを表す図中のアイコンの中から新規に編集したいモーションパターンを選び、3D 空間中にコピーすることで、動作を追加することができる。

## 5. モーション編集の定義例

### 5.1 移動+回転動作の定義

3次元空間上で実際にタイムラインオブジェクトを用いてモーション編集を行う様子を具体例を用いて説明する。例として、人形がモーションパスに沿って、反時計周りに回転しながら移動するパスアニメーションを挙げる (図 6)。

#### (1) 移動経路を描く—図 6(a)

まず、図 5 のテンプレートから移動 (円弧) のモーションパラメータ  $mp_t$  をもつタイムラインオブジェクトを 3 次元空間上にコピーする。これによって移動経路上の 3D オブジェクトの時間位置を指定する。これは、図 1(a) のタイプの継承に対応する。この時点でオブジェクトの 3 次元空間上での位置が決まるが、向きは未だ規定できていない。

#### (2) 支軸の追加—図 6(b)

次に、ユーザはオブジェクトの向きを表すために、タイムラインの  $y$  軸方向に幅を与える。モーションパス上の各点を通る鉛直線をオブジェク

トの  $y$  軸に継承することで、オブジェクトの方向を指定でき、人形の姿勢が定まる。結果、タイムラインオブジェクトは図に示すように带状の図形になり、帯の長さ方向は軌道の時間長を表し、帯の中心線はオブジェクトの中心に継承できる。この帯を  $TimelineObj1$  とする。

#### (3) 回転動作の追加—図 6(c)

さらに、 $y$  軸を中心として反時計周りに回転する動作を加える。図 5 のテンプレートから回転角度のモーションパラメータ  $mp_r$  をもつタイムラインオブジェクトを 3 次元空間上にコピーし、これを  $TimelineObj2$  とする。図 3(b) で示したように、3D オブジェクトの方向ベクトルに対してタイムラインオブジェクトを継承することにより、3D オブジェクトが回転する動作が記述できる。

$TimelineObj1$  は横軸を時間  $t$ 、縦軸が移動 (transition)  $mp_t$  を表し、 $TimelineObj2$  は横軸が時間  $T$ 、縦軸が回転角度 (rotation)  $mp_r$  を表す。双方とも、形状をもつオブジェクトとして 3D 空間上に可視化されているのがわかる。

#### (4) 回転動作の編集—図 6(d)

必要に応じて、3D 空間上の  $TimelineObj2$  に編集を加える。図形の編集点をピックし、形状を変更する。このように帯の形状を変更することで、望む回転動作に変更することができる。

#### (5) 回転動作の可視化—図 6(e)

モーションラインにおける人形がどのように回転しているのかを確認するため、 $TimelineObj2$  による編集の結果が  $TimelineObj1$  の形状の変更として反映され、編集によって変更を加える度にモーションが変化するイメージが可視化される。回転動作の変化のポイントを視覚的に理解することができ、ユーザの編集プロセスを支援する。

以上のように、 $TimelineObj2$  を制御インターフェースとして、 $TimelineObj1$  を直感的な理解を助ける視覚的インターフェースとして 3D 空間上で統合的に扱うことにより、運動のイメージを確認しながら直感的にモーション編集を行うことが可能となる。

### 5.2 移動+加速動作の定義

同様に、3D オブジェクトである人形の移動速度を

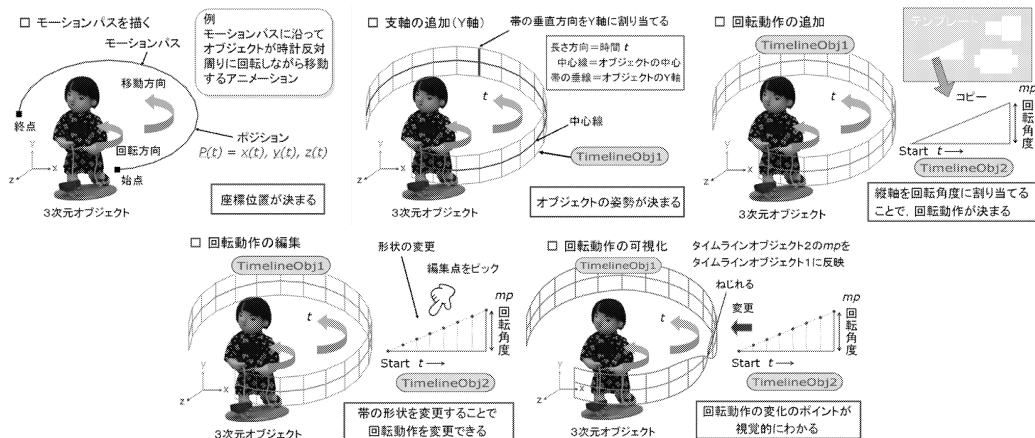


図 6 タイムラインオブジェクトによる移動・回転動作編集の例  
 Fig. 6 Example : Transition and rotation control by *TimelineObject*

制御する場合を考える。新規の *TimelineObj3* は横軸に時間  $t$ 、縦軸に速度  $mp_v$  をパラメータに持つ。図 5 のツールパレットから加速を表すテンプレートを 3 次元空間にコピーし、*TimelineObj3* を作成する。速さの表現は、座標や方向等と違い静止オブジェクトが持つ直接的な属性ではないため、そのままでは継承ができない。このため、以下の前処理を行う。

モーションパスを表す *TimelineObj1* とは独立に速度パラメータ  $mp_v$  を持つ *TimelineObj3* を作成する。この時間方向の積分が、モーションパス上の座標位置を表すが、この場合加速度をパラメータにするため、2 回積分を行う必要がある。このように、静止オブジェクトの持つ座標や方向という属性値と、*TimelineObj3* が持つパラメータとの関係を表現する演算を定義することで、継承と可視化が可能となる。

最後に、静止オブジェクトに対して *TimelineObj3* が継承されることで加速動作が追加され、速度パラメータの編集に応じて *TimelineObj1* の帯の幅が変化する。*TimelineObj1* の幅の変化という視覚的な次元の拡張によって、幅が太くなるほど *TimelineObj1* 上の人形の移動速度が速くなり、細くなるほど速度は遅くなるように加速運動の様子が可視化できる (図 7)。

## 6. む す び

本論では、タイムラインオブジェクトというインタフェースの提案を行った。各種の運動パラメータを可視化しながら、モーション編集の操作系を 3 次元空間上に統合し、ダイレクトに“つまむ”ことで制御が行える。これにより、ユーザは直感的なモーション編集手段を得ることができ、簡単に映像作成を行うことが可能となるというのが、提案内容の骨子である。

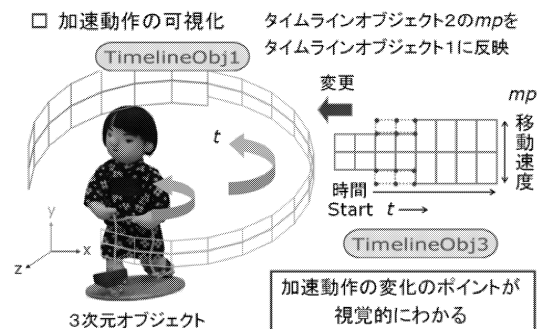


図 7 加速運動が可視化される様子  
 Fig. 7 Example : Visibility of acceleration

また、タイムラインオブジェクトをビジュアルなモーションの部品と考えることで、モーションプリミティブを構成し、さらに、モーションプリミティブを組み合わせることで複雑な動きが合成・可視化する案を示した。ユーザの編集履歴のうち頻度の高いパターンを抽出し、テンプレートにフィードバックさせるといった再利用性を向上させることで、さらにオーサリング環境を充実させることも可能であろう。このあたりは次回の課題としたい。

序段に述べたように、今日のアニメーション技術の発展はめざましく、モーションキャプチャ等デバイス入力による動作データの入手手段や、物理法則に従うシミュレーション的アプローチによるモーション表現等、質の高いモーションデータを簡便に作成するための技術は多岐に渡っているが、本報告は 3DCG ソフトウェアのオーサリングにおけるユーザビリティ向上を対象としたものであることを付記しておく。

今後は、本提案内容であるタイムラインオブジェク

トの GUI と可視化部分, そしてモーションプリミティブを開発中の WebReality に実装し, ユーザビリティの評価を行う.

**謝辞** 本研究の予備検討を行う上でデータを提供していただいた(株)NTT-ME 青森, ならびに貴重なアドバイスをいただいた NTT マイクロシステムインテグレーション研究所ユビキタスインターフェース研究部ホームコミュニケーション研究グループの青木孝主任研究員, 前田典彦研究主任はじめ, 関係者の皆様に感謝致します.

### 参 考 文 献

- 1) 山崎晃嗣, 木村洋, 松岡裕人, 原田育生 “3次元仮想空間における時間の一定義手法” 2005年電子情報通信学会総合大会論文集, CD-ROM
- 2) 高リアリティ3D キャプチャシステム:  
<http://www.contents4u.com/>
- 3) 藤代一成, 斉藤隆文, 乃万司他: コンピュータグラフィックス, CG-ARTS 協会 (2004).
- 4) Edwin E. Catmull. *The problems of computer-assisted animation*. Computer Graphics (SIGGRAPH '78 Proceedings),12(3):348.353, August 1978.
- 5) Alias MotionBuilder:  
<http://www.alias.co.jp/motionbuilder/>
- 6) SOFTIMAGE—XSI:  
<http://www.softimage.jp/softimage/XSIv4/>
- 7) アニメーションゴースティング  
<http://www.comtec.daikin.co.jp/si-xsi/product/vup/v40/>
- 8) B.Javidi, F.Okano. et al.: *Three-Dimensional Television, Video, and Display Technologies*, Springer-Verlag Berlin Heidelberg (2002).