

コンピュータアニメーションにおける仮想カメラの制御

1D-2

林美奈子

水野純子*

根本啓次

NEC C&C情報研究所

NEC 技術情報システム開発(株)*

1 はじめに

近年では、非常に多くの分野でコンピュータアニメーション(以下アニメーション)が利用されている。これに伴い簡単にアニメーションを作成できる技術の需要が高まっている。

アニメーションでは、仮想的な舞台を設定し、仮想舞台上の物体の動きを仮想的なカメラで撮影する。この仮想カメラを制御するには、複雑な操作が必要なので多大な時間を必要とした。そこで、本稿ではこの仮想カメラの複雑なカメラワークを簡単に実現できる仮想カメラの制御方法を提案する。

2 仮想カメラ制御の問題点

アニメーション作成における仮想カメラ制御の研究は数多く行われてきた[1]。

アニメーションの作成では、まず仮想的な舞台を設定し、登場物体の舞台上での位置と動作と動作を行なう時刻とを設定する。この設定の後、仮想舞台上にカメラを設定し、カメラにより仮想舞台をどのように表示するかを設定(=カメラワーク設定)する。

従来の多くのシステムのカメラワーク設定では、カメラ位置、視線方向、視野角などのカメラデータをキーフレーム毎に設定していた。

このような方法には以下の問題点があった。

- カメラデータと作成されるアニメーションとの対応を直観的に理解しにくい。
- 設定する情報量が非常に多大なため設定に時間を要するだけでなく、設定操作が複雑なため、情報の未設定、重複が発生しやすい。

3 仮想カメラの制御

提案する仮想カメラの制御方式について説明する。

3.1 カメラ制御関数

一般に、映画、テレビ等のカメラワークでは、ズーム、パン等のいくつかの標準的なカメラの動きの型があり、こうした動きの型を1つまたはいくつか組み合わせる複雑なカメラワークを実現している。本方式では、こうした標準的なカメラの動きの型を簡単に実現するために、カメラの動きを制御する制御関数(=カメラ制御関数)をそれぞれの型に対応させて設けている。各カメラ制御関数は、カメラ位置、視線方向、視野角といったカ

メラデータを、動きの型に合わせて順次算出する。ただし、カメラの上方向は常に舞台上方に固定している。

- fix: 全てのカメラデータを固定。
- follow: 対象の動きに沿ってカメラ位置を移動する。
- pan: カメラ位置を固定して、対象の動きを追う。
- zoom: 視野角を変化させて、対象を拡大・縮小。
- move: 視線を対象に拘束したまま、ユーザの指示通りにカメラ位置を自由に移動させる。
- swing: 視線を対象に拘束したまま、対象を中心にカメラ位置を回転させる。

3.2 カメラ制御関数の使用方法

カメラ制御関数の使用方法を説明する。

カメラ制御関数はアニメーション中の登場物体の動作に対応して指定する。

そして、カメラデータを直接入力するのではなく、各カメラ制御関数は設定した登場物体の動作に合わせてカメラデータを自動的に算出する。このカメラデータは、時刻に対応して順次算出される。

登場物体の動作のみでは算出できないカメラデータは、次のように定める。

1. カメラ制御関数で変化させないカメラデータ(例: panのカメラ位置)は、前時刻までのカメラデータを指定する。前時刻のカメラデータがない場合は、舞台全景を中央から撮影するカメラデータを使用する。
2. カメラ制御関数が必要とする特定のデータ(例: moveのカメラ位置、zoomの視野角度変化量)はユーザが入力する。

カメラ制御関数の指定およびカメラデータ算出の例を次に示す。図1の例では、登場物体TAROの動作walkに対応してカメラ制御関数followを指定している。この場合、カメラ制御関数followは登場物体TAROの動作walkに合わせてカメラデータを算出している。

3.3 カメラ制御関数の組合せ

本方式では、カメラ制御関数を登場物体の動作に対応して指定しカメラワークを実現している。更に、複雑なカメラワークを実現するために複数のカメラ制御関数を組み合わせるカメラワークを行なう方法を説明する。

まず、登場物体の動作に対応して複数のカメラ制御関数を指定する。しかし、各々のカメラ制御関数はカメラ

Virtual Camera Control in Computer Animation, Minako HAYASHI Junko MIZUNO Keiji NEMOTO, NEC Corp. NEC Scientific Information System Development, Ltd.*

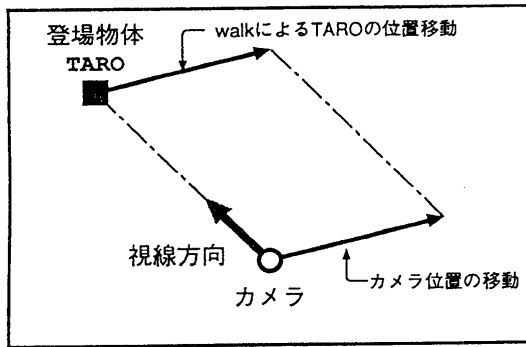


図1 カメラ制御関数指定の例

データの算出方法が異なるので、指定された複数のカメラ制御関数はそれぞれ別々のカメラデータを算出してしまふ。そこで、各々のカメラ制御関数間のカメラデータ毎の優先規則をあらかじめ決めておき、カメラデータ毎にどのカメラ制御関数の算出した値を使用するかを決定する。

例えば pan と zoom を組み合わせる場合には、図2に示すようにカメラ位置と視線方向とは pan が優先なので pan の値を用い、視野角は zoom が優先なので zoom の値を使用する。その結果、図2ではカメラ位置は固定のまま視線を対象の動きに追従させながら視野角を変化させることになる。

zoom & pan

- ・カメラの位置：pan が優先
- ・視線方向：pan が優先
- ・視野角：zoom が優先

図2 カメラデータ優先規則の例

3.4 スケジューリングの方法

アニメーションの全時間にカメラワークを指定する方法を説明する。

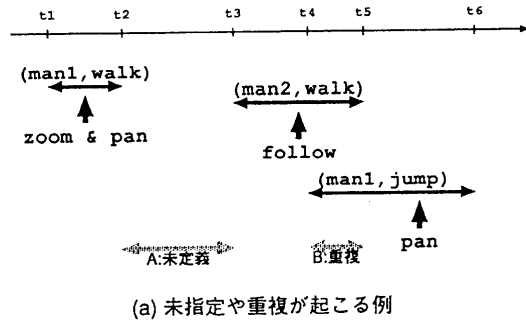
アニメーションのある時刻においてカメラ制御関数が指定されていないか、異なる登場物体の動作にそれぞれカメラ制御関数が指定されていて異なるカメラ制御関数が重複してしまうことがある(図3(a))。このような場合、カメラデータは決定できない。そこで以下の規則を定めている。

未指定の場合

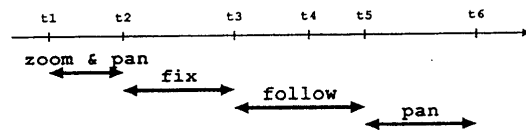
カメラ制御関数として fix を指定する。(図3(b))。

指定が重複する場合

各登場物体の動作に優先度を与えておく。カメラ制御関数が重複する場合には、優先度の高い動作に指定されたカメラ制御関数を使用する。優先度が設定されていない場合や優先度が同じ場合には、開始時刻の早いものを優先する(図3(b))。



(a) 未指定や重複が起こる例



(b) スケジューリング規則適用後

図2 スケジューリング規則適用の一例

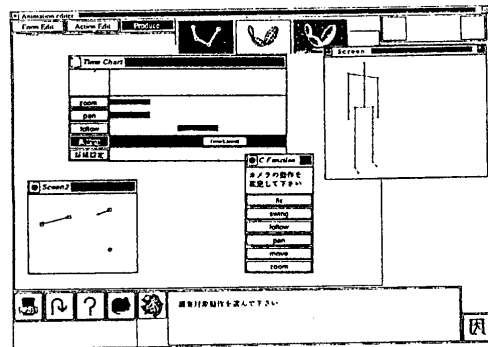


図3 画面の一例

4 おわりに

仮想カメラの制御方法を提案した。この提案方式を用いたアニメーション作成作業の画面の一例を図4に示す[2]。提案方式により、複雑で多時間を要した仮想カメラの制御を簡単な手順で短時間に行なうことが可能となった。また、カメラ制御関数を用いることにより作成されるアニメーションを直観的に推測することが容易であることが確認できた。更に、カメラワークの未指定や重複を気にせずに作業できるのでより容易にアニメーションを作成できることを確認した。

参考文献

[1] 乃万、岡田：コンピュータ・アニメーションにおける自動仮想カメラ、情報処理学会第44回全国大会
 [2] 林、水野、根本：アニメーション作成環境—アニメーション・エディター、情報処理学会第43回全国大会