1P－6

# VirStA System: 仮想ステージと仮想アクターによる分散CGアニメーションシステム II. 分散環境でのリアルタイム実装

阿部哲也　後藤真孝　松本英明　村岡洋一

早稲田大学　理工学部

## 1. はじめに

本稿では、VirStA System (Virtual Stage and Actors System) の分散環境でのリアルタイム性を確保するための実装方法について述べる。VirStA System では、CG キャラクター (仮想アクター) の動作決定処理 (VirStA Actor: 以下 Actor と略す) と画像生成処理 (VirStA Renderer: 以下 Renderer と略す) を分離して異なる計算機に負荷分散することで、リアルタイム性を損なうことなく多様な拡張を可能にしている [1]。VirStA System で扱う CG キャラクターは、制御パラメータつきの多関節骨格構造とポリゴンの表面モデル (CG アクターモデル) で表現される。Actor はこの CG アクターモデルを Renderer に事前に登録し、動作中は変化する最小限の制御パラメータのみを伝送する。この制御パラメータは関節の 3 自由度の角度や関節における平行移動の指定をする。

しかし一般に計算機の処理能力にくらべネットワーク (LAN) の伝送能力は低い。このため Actor の数が増えたり動作が複雑になったりすると、Renderer へ伝送される制御パラメータの量が増大し、ネットワークがボトルネックになってリアルタイム性を確保するのが難しくなる。

本研究ではこれを解決するために、短期間のパラメータの時間変化をまとめて指定できる時限スクリプトという概念を導入する。Renderer が Actor から受信した時限スクリプトを実行中は、Actor はそのパラメータの変化を伝送する必要がなくなるので、ネットワークの負荷を抑えられる。そこで、ネットワークの負荷と Renderer の負荷のトレードオフを考えて、時限スクリプトを Actor と Renderer のどちらで実行するかを変更することで、動作決定処理と画像生成処理の動的負荷分散を実現する。

## 2. 時限スクリプト

時限スクリプトとは、骨格構造の制御パラメータの時間変化関数やパラメータ間の束縛条件を期間限定つきで指定するものである。制御パラメータの時間変化は短期的には単純な関数になることが多く、関節ごとに常に独立な値を取るわけではない[1]。そこで、スクリプトによって、パラメータの時間変化関数やパラメータ間の束縛条件を指定することで、複数のパラメータを効率良く制御することができる。

Renderer は受信したパラメータ変化や時限スクリプトの実行結果に基づいて、CG アクターモデルをリアルタイムに更新する。VirStA Manager は、画像を生成しない以外は Renderer とまったく同様に CG アクターモデルを更新する。また、VirStA Manager は仮想ステージの状態 (登録された全 CG アクターモデルと動作中の全パラメータ値) を保存・再現する機能を持つ。

### 2.1 時限スクリプトの指定

Actor の多様な動作を記述できるためには、時限スクリプトは自由度が高く柔軟である必要がある。そこで、時限スクリプトの機能を以下の 3 つに分けて考え、各機能における自由度を高めることで全体の柔軟性を高める。これらは C 言語風のテキストで記述するが、伝送時には冗長度を減らした中間表現に変換する。

**時間指定の機能**　開始時刻と有効期間を指定するだけでなく、持続する変化の開始時刻だけを指定したり、周期変化の開始時刻と周期を指定したりできる。

**変化関数指定の機能**　二点間の様々な補間関数、漸化式、フーリエ級数展開の係数、周期関数の 1 周期分の数列データ、ユーザ定義関数などを指定できる。

**変化量伝搬指定の機能**　上記の時間指定と変化関数指定では抽象的な変化量だけを計算するので、それをどの関節の何のパラメータに影響させるかを指定する必要がある。この際、複数の関節に影響の重みを変えて同時に伝搬できるようにする。関節間の束縛条件は、ある関節の変化量をさらに別の関節の変化量へ伝搬させることで実現する。

### 2.2 複数スクリプトの同時実行制御

複数の時限スクリプトを同時に実行する場合には、実行順序によってパラメータの値が異なる可能性がある。そこで各時限スクリプトに実行順のプライオリティを設定し、その高いものから順に実行する。さらに複数が同一のプライオリティを持つ場合には、その開始時刻が早いものから順に実行する。そのためのプライオリティや開始時刻は、Actor の設計者が適切に定める必要がある。

---

VirStA System: A Distributed CG Animation System based on Virtual Stage and Virtual Actors
II. Real-time Implementation in Distributed Environment
Tetsuya Abe, Masataka Goto, Hideaki Matsumoto, and Yoichi Muraoka, School of Science and Engineering, Waseda University.

[1]例えば、人間の指の関節などは、ある関節の角度との関連から従属的に他の関節角度が決まることがある。

## 3. 時限スクリプトによる動的負荷分散

ネットワークがボトルネックになりリアルタイム性が確保できない場合には、パラメータの制御に時限スクリプトを多用してネットワークの負荷を減らせばよい。しかしRendererで多くの時限スクリプトを実行し過ぎると、今度はRendererの方がボトルネックになる。

これを回避するには、Rendererの負荷とネットワークの負荷のトレードオフを考えて、各時限スクリプトをActorとRendererのどちらで実行するか動的に変更すればよい。負荷は動的に変動し設計時にはわからないため、ユーザが個々のスクリプトの実行場所を適切に指定することは難しい。そこで、実行時に負荷を計測しながら実行場所を決定し、動的負荷分散をおこなう。

### 3.1 動的負荷分散の実現上の課題と解決法

- 負荷の計測

  動的負荷分散では実行時にActorがネットワークの負荷とRendererの負荷を知る必要がある。これらの負荷を推定するために、Rendererは毎フレームごとに1) 受信したパケットのサイズの総和、2) 1フレームの時間に対するレンダリング等の処理時間の比率を計測し、一定時間おきに各Actorにブロードキャストする。ここで、1) はネットワークの負荷に相当し、2) はRendererの負荷に相当すると考えられる。そこで、現在は簡略化した実装として、時限スクリプトは基本的にActor側で実行し、もし1) の値が閾値より大きく2) の値が閾値より小さいときにはRendererで実行するようにする。

  また、様々な視点から見るために複数のRendererを実行する場合には、基準となる負荷の計測をおこなうMaster Rendererを一つだけ定め、これが負荷情報をブロードキャストする。他のRendererはSlave Rendererとなり、動的負荷分散には関与しない。

- スクリプトの実行制御

  Actorが制御パラメータの値に依存して次の状態を決定したいときに、Rendererが時限スクリプトを実行していると、その制御パラメータの現在の値をActorがわからなくなる問題がある。この場合Actorは、ネットワークの負荷を下げるために時限スクリプトは送信しても、自分で同じ時限スクリプトを二重に実行する必要がある。

  そこで時限スクリプトを、両者で二重に実行しなければならないスクリプトと、一方でしか実行しなくてもよいスクリプトの2種類に分類する。ユーザはActorを設計するときに、動作決定方法に応じてこれらを個別に選択する。

## 4. 実験結果と考察

本システムをEthernetで接続された3台のSGI Indigo2上に実装して実験した。現在の時限スクリプトの実装完全でなく、時間指定はすべて対応しているが、変化関数指定は初期値と最終値の二点間の補間関数(任意のべき乗関数と指数関数)のみに対応している。また変化量伝搬指定は、変化関数の値を重みづけした後、パラメータの絶対値として伝搬する方法と相対値として加算していく方法に対応している。関節間の変化量の伝搬は現時点では実装されていない。

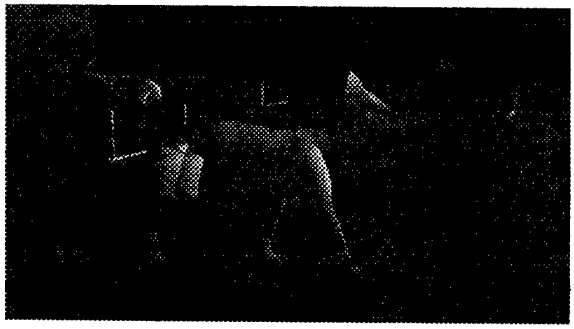

図1: システムの画像生成例

| 処理の種類 | Actor | Renderer |
|---|---|---|
| パケット送受信 | 6[ms] | 4[ms] ( 6) [2] |
| その他の処理 | 16[ms] | 12[ms] (20) |
| プロセス全体 | 22[ms] | 16[ms] (26) |

[2]() 中の数字はActorを2つ実行した場合

表1: 1フレームあたりの各プロセスの実行時間

実験ではCGアクターとして仮想犬[3]を用いた。仮想犬のActorを2つ実行した場合の画像生成例を図1に示す。2つのActorとRendererをすべて異なる計算機で実行したときのプロセスの実行時間を測定した結果を表1に示す。処理を分散したことで通信に数msのオーバヘッドを要したが、すべてのプロセスが33ms以内に1フレームの処理を実行できており、30fpsのリアルタイム生成が可能であった。一方、VirStA Systemを使わず2匹の仮想犬を単一の計算機で生成すると20fpsであった。以上からVirStA Systemの負荷分散がリアルタイム性の確保に有効であることが確認できた。

## 5. おわりに

本稿では、動作決定処理と画像生成処理を負荷分散可能なリアルタイムCGシステムVirStA Systemの実装方法について、時限スクリプトと動的負荷分散を中心に述べた。本システムを実装して実験した結果、本システムが実際にリアルタイム性の確保に有効であることを確認した。今後は、対応していない時限スクリプトのすべての機能を実装すると共に、より汎用的で複雑な動作も記述できるように拡張していく予定である。

### 参考文献

[1] 後藤 真孝、阿部 哲也、松本 英明、村岡 洋一: VirStA System I. システムの全体構想, 第53回情処全大 1P-05, 1996.

[2] 松本 英明、後藤 真孝、阿部 哲也、村岡 洋一: VirStA System III. ジャズセッションプレーヤーの実現, 第53回情処全大 1P-07, 1996.

[3] 阿部 哲也、水野 裕識、村岡 洋一: CGによる犬の柔軟なアニメーション生成、情処研報 グラフィックスとCAD 95-CG-73, Vol.95, No.18, 1995.