

3Dソフトウェア構築システムIntelligentBoxによる 手間をかけないCGアニメーション作成

岡田義広 田中譲

北海道大学工学部

060 札幌市北区北13条西8丁目

電子メール:okada,tanaka@huee.hokudai.ac.jp

著者らは、固有の機能をもつ3次元可視オブジェクト(ボックス)を画面上での直接操作により組合せ機能合成することにより3次元ソフトウェアの開発が行える3次元ソフトウェア構築システム(IntelligentBox)の研究開発を行っている。IntelligentBoxの応用の1つとしてCGアニメーション作成が挙げられる。動き定義、形状変形制御、シーン管理、カメラやライトの機能をもつボックスを実現することで、カメラ、ライトの効果やオブジェクトの動きを画面上で直接操作し確かめながらCGアニメーションを作成することが可能となる。本稿では、CGアニメーション作成のためにIntelligentBoxが提供する機能とBoxについて説明し、CGアニメーション作成におけるIntelligentBoxの有用性を述べる。

Easy Creation of CG Animation Using IntelligentBox

Yoshihiro Okada and Yuzuru Tanaka

Faculty of Engineering, Hokkaido University

N.13, W.8, Kita-ku, Sapporo, 060 Japan

e-mail:okada,tanaka@huee.hokudai.ac.jp

The authors have been studying a constructive visual 3D software development system called IntelligentBox. The IntelligentBox system enables users to construct 3D graphic applications by combining 3D functional objects called Boxes through direct manipulations on a computer screen. One of the application fields of the IntelligentBox is computer animation creation. By providing Boxes whose functions are for motion definition, deformation, scene management, camera and light model, it is possible to create CG animation rapidly with visually checking those effects on a computer screen. This paper describes the functions of the IntelligentBox system and characteristic Boxes, and clarifies their availability for CG animation creation.

1. はじめに

我々は、固有の機能をもつ3次元表示されたオブジェクトを組合せ機能合成することにより、3次元表示機能をもつ応用システムを構築可能な3次元ソフトウェア構築システムの研究・開発を行っている。すでに、IntelligentBox[1,2]と呼ぶプロトタイプ・システムを開発している。IntelligentBoxの機能合成の機構は、IntelligentPad[3]の機構を採用している。IntelligentPadは、現在、北海道大学で研究・開発を行っているシンセティックメディア・システムである。IntelligentPadにおける紙のイメージをもつ機能オブジェクトであるパッドを3次元形状をもつ機能オブジェクトへと拡張したシステムがIntelligentBoxであるといえる。すべての3次元オブジェクトがボックスの形状をもつものではないが、IntelligentBoxでは多面体の形状をもつ3次元オブジェクトを総称してボックスと呼ぶ。各ボックスは3次元形状と固有の機能をもつ。このボックスを画面上で互いに親子関係を与え組み合わせることにより、合成された形状と合成された機能をもつボックスを構築できる。IntelligentBoxシステムでは、この構築過程が3次元応用システムの開発過程となる。3次元応用システムを開発する上で有用となる種々の基本部品（特定の機能をもったボックス）を現在までに開発している[4,5]。

現在、IntelligentBoxシステムは、シリコングラフィックス社のグラフィックスワークステーション上で稼動しており実時間描画が可能である。フルカラー描画された機能部品であるボックスを画面上での直接操作により組み合わせることによって形状設計と機能合成を同時に行うことができ、手間をかけずにCGアニメーション作成を行うことができる。

すでに、商用のCGアニメーション作成用ソフトウェアが多数開発されている。ライト効果や種々のレンダリング手法を採用して高精細な画像を作り出すことのできるものもある。ただし、これらのソフトウェアには、新たな機能を部品レベルで加えるとといった機能はない。IntelligentBoxでは、必要とする機能を可視部品として定義登録することで、容易に機能拡張が行える。IntelligentBoxシ

ステムでは、オブジェクトの形状モデリング機能と描画機能はIntelligentBoxシステム自身の機能として提供する。モーション定義および形状変形制御はボックスの機能として提供する。また、カメラやライトの機能もボックスとして実現している。本稿では、モーション定義機能をもついくつかのボックスとソフトオブジェクトを扱うための形状変形機能をもつボックスの使用例を挙げる。また、カメラやライトの機能をもつボックスやシーンの切り替えを自動で行うためのボックスについて述べ、これらボックスの有用性を示す。

以下では、2節で、まずIntelligentBoxの概要としてIntelligentBoxの機能合成機構を述べる。3節ではCGアニメーション作成のためにIntelligentBoxが提供する機能と機能部品であるボックスについて個々に説明する。それぞれについて簡単な例を示しIntelligentBoxシステムのCGアニメーション作成における有用性を示す。最後に4節でまとめと今後の課題を述べる。

2. IntelligentBoxの概要

2.1 MDモデリング

旧版のIntelligentBoxは、IntelligentPadと同様にMVCモデリング[1,2,3,4]を採用していた。1つのボックスは、Model、View、Controllerと呼ばれる3つのオブジェクトで構成された。Modelはボックスの内部状態を保持するオブジェクトである。状態情報はスロットと呼ばれるものに保持される。Viewはボックスの画面上での視覚的な形状を表すオブジェクトである。Controllerはボックスに対する使用者からのマウスイベントを受け取り、Viewへ送るためのオブジェクトである。一方、現在のIntelligentBox[5]ではMDモデリングを採用している。ボックスに対するユーザ操作はすべてのボックスにおいて共通であるため、複数種のController

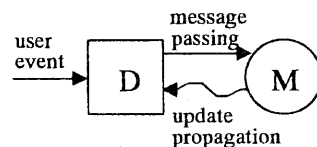


Fig.1 ボックスのMD構造

を用意する必要がない。新しいボックスを作成する際のコーディングの冗長性をなくすため、Controllerの機能をViewの機能に含めた。このControllerの機能を合わせもつViewをディスプレイオブジェクトと呼ぶ。図1に示す方式でメッセージ送信と状態の更新報告を行う。

2.2 スロット結合によるデータ授受

IntelligentBoxは、IntelligentPadと同様のプロトコルにより、親子関係があるボックス間でデータの授受を行う。親子関係は各ボックスのディスプレイオブジェクト間の関係として決められており、以下のメッセージもディスプレイオブジェクト間でやりとりされる。

図2に示すように、`set <slotname> <value>`、`update`、`gimme <slotname>`メッセージがある。`set <slotname> <value>`は、各ボックスが自分の親ボックスがもつ<slotname>で指定されるスロットに対して値<value>を送るためのメッセージである。`gimme <slotname>`は、自分の親ボックスがもつ<slotname>で指定されたスロットから値を読み出すためのメッセージである。`update`は、各ボックスが自分のすべての子ボックスに対して、自分の内部状態が変化したことを知らせて状態更新を要求するためのメッセージである。

上記のメッセージの送信許可は、ディスプレイオブジェクトがもつ`set`、`gimme`、`update`フラグによって決められる。`set`フラグが立っているボックスはその親ボックスに対する入力器として機能する。逆に、`gimme`フラグが立っているボックスは

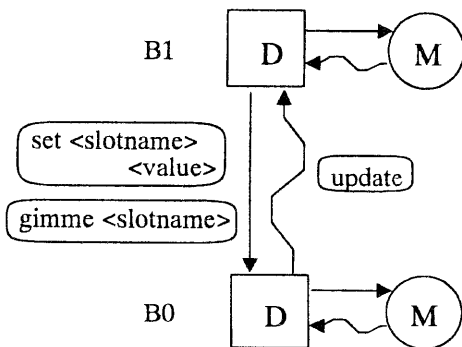


Fig.2 ボックス間の標準メッセージ

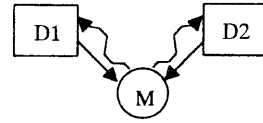


Fig.3 2つのボックス間のModel共有

その親ボックスのもつスロット値を取り出す出力器として機能する。

2.3 Model共有によるデータ共有

2.1節で述べたように、それぞれのボックスはModel、DisplayObjectの2者で構成される。図3に示すように、ひとつのModel部は複数のDisplayObjectをもつことができる。この際、複数のDisplayObjectはModelを共有しているという。これにより、Model部を共通にもつ異なる2つのボックス間でデータの共有が可能となる。

3. CGアニメーション作成

アニメーション作成過程は、以下に示すようにモデリング機能、レンダリング機能、モーシオン定義機能の3つに分けられる。

3.1 モデリングおよびレンダリング機能

従来のIntelligentBoxシステムは、Symbolics社のLISPマシン上のS-Productsというコンピュータアニメーション作成ソフトウェアにIntelligentPadがもつ基本機構を加えることにより実現されていた。各ボックスの形状モデリングはこのソフトウェアが提供するモデリング機能を用いることにより可能であった。グラフィックスワークステーション上に今回実装したIntelligentBoxシステムでは、このソフトウェアと同様のモデリング手法を採用しており、このソフトウェアが提供する機能の一部を実現している。3次元形状の各構成要素(頂点、稜線、面、立体)を単位として、種々の変形操作が画面上での直接操作により可能である。

現在のIntelligentBoxシステムは、C++言語によりOpenGLライブラリを用いて開発されている。その描画処理はOpenGLライブラリにすべて依存している。ワイヤーフレーム、フラットシェーディング、グローシェーディングおよびテクスチャマッピング描画が実時間で可能である。

3.2 モーション定義機能

すでに述べたように、オブジェクトの動きに制約を与えるための機能をもつボックスがいくつか実現されている。以下の図4,5,6に示すのはこれらの例である。

3-2-1 汎用稼動部品の合成による制約

伸縮部品や回転部品等の汎用稼動部品を合成することにより、オブジェクトの動きに制約を与えることができる。図4は、回転部品(RotationBox)と伸縮部品(ExpandBox)を用いて機能合成することにより、シャトルが回転しながら上昇する動きを実現した例である。

3-2-2 軌道部品

特定の軌道上を移動するという動作を実現するために、軌道自身を表すボックス(TrajectoryBox)とその軌道上のみ移動するという機能をもつボックス(TrajectoryMoverBox)を用意している。図5は、与えられた軌道に沿って進んで行くシャトルの動きを定義した例である。

3-2-3 物体表面を移動する制約部品

特定の物体の表面に沿って移動するという動作を実現するために、親ボックスの表面上を移動するという機能をもつボックス(CreepingBox)を用意している。図6は、机の上をアリが這っていく例である。

3.3 形状変形制御機能

アニメーション作成において、ボールが壁に当たってつぶれるようなそれ自身の形状が変形するオブジェクトを含む場合がある。このようなソフトオブジェクトを扱うためにFree Form Deformation(FFD)とよばれる立体形状の変形制御の

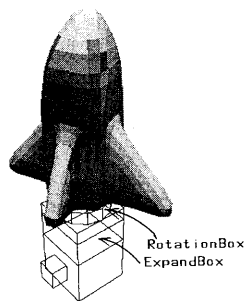


Fig. 4 稼動部品の合成による動きの定義例

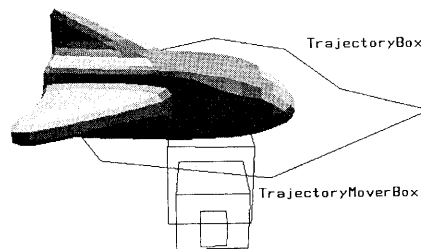


Fig. 5 軌道部品を用いた動きの定義例

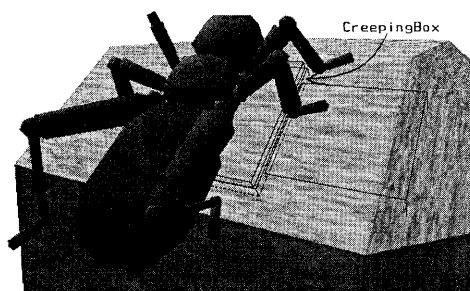


Fig. 6 机の上をアリが這う例

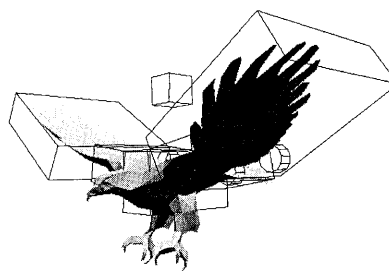


Fig. 7 FFDControlBoxを用いたワシの翼の変形制御例

方法[6]が提案されている。これをFFDControlBoxとよぶボックスの機能として実現している。FFDControlBox自身の形状が制御部品として働くのではない。FFDControlBoxは、別に用意される制御部品の形状情報にしたがって制御対象の形状を変化させる機能をもつ。制御部品は合成されたものでもよく、汎用稼動部品を任意の組み合わせで用いてもよい。これにより種々の動きを定義できる。図7に示すのは、FFDControlBoxを用いてワシ

の翼の動きを実現した例である。回転部品と直方体の形状をもつ部品を合成することにより、翼の動きを定義している。この合成ボックスをワシの翼の立体形状を制御する部品としてFFDControlBoxと組み合わせて用いることにより、ワシが翼を羽ばたかせる様子を模倣する。

3.5 カメラ機能

アニメーション作成において視線は重要な要素である。IntelligentBoxでは視線の管理を行うための機能部品(CameraBox)[6]を用意している。CameraBoxは、固有のビューワーウィンドウを持ち、指定された面の中心を視点としその面の法線方向を視線として見える視野をこのウィンドウの内部に表示する機能をもつ。また、このCameraBox自身が他の部品によって制御され移動する際に、特定のオブジェクトに常に視線を向きたい場合がある。このような場合のために、特定の面が指定されたオブジェクトを常に向くという機能をもつ機能部品(AutoDirectionalBox)を用意している。これらの部品を他の機能部品と組み合わ

せて用いることにより、アニメーション作成の際の視線管理が容易に行える。

3.6 ライト機能

すでに述べたように、IntelligentBoxの描画処理はすべてOpenGLライブラリの関数を用いている。したがって、OpenGLライブラリが提供する関数を容易にボックスの機能として実現可能である。OpenGLが提供するライトモデルを独立なボックス(LightBox)として実現している。ライトモデルの定義に必要なパラメータはボックスのロットとして実装している。したがって、他のボックスによりそれらのロット値を制御することで、ライト色の変更やスポットライトの効果を実現できる。

図8は、軌道上を進む動作とワシが翼をはばたく動きを組み合わせた例である。この軌道上を飛んでいくワシに、スポットライトを当てて、それをカメラボックスで撮しているところである。スポットライトとカメラ自身も別々の軌道上を進んでいる。画面の前面右下にあるボックスの3つのボリュームスイッチによって、スポットライトの

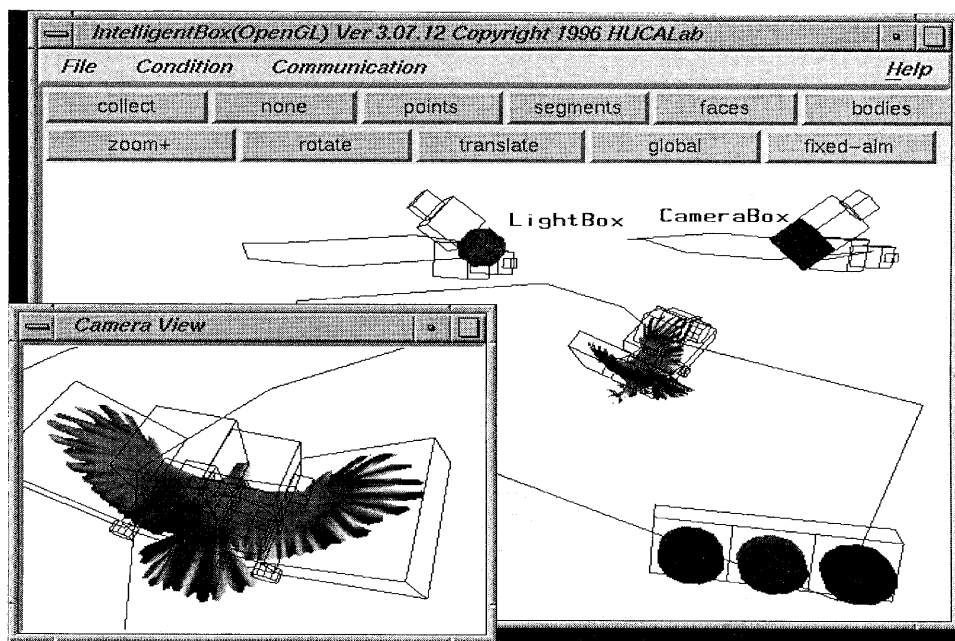


Fig. 8 軌道上を飛んでゆくワシにLightBoxを用いてライトを当て、CameraBoxを用いてそれを撮してカメラビューに表示している例

光源色の赤、緑、青のそれぞれを設定する。左下の小さなウィンドウがカメラのビューである。

3.7 シーン管理

IntelligentBoxでは実時間描画可能であるので描画された画面のイメージを1画面ずつ記録してひとつの連続したアニメーションにする必要はない。しかし、複数のシーンで構成されるアニメーションを見せる場合、シーンを切り替えるための機能が必要になる。そのための機能も独立なボックス(SceneSwapBox)の機能として実現している。

図9に示すように、このボックスはNextSceneという文字列型のスロットをもつ。このスロットにはファイル名が入り、ファイル形式で保存されたボックスを指し示す。ファイル名をスロットとしてもつプッシュボタンボックスを押すことにより、図の矢印の流れにしたがってNextSceneスロットの値が更新される。SceneSwapBoxは、更新された値によって対応するファイルを読み込みボックスを生成し自分の子どもとする。この読み込みの前に、すでにある子孫のボックスはすべて削除される。このようにして、シーンの切替が行われる。それぞれのシーンをファイル形式で保存しておくことで、シーンの切替を簡単に行うことができる。さらに、時間情報をもつボックスと数値の大小比較判定を行うボックスを組み合わせて使い判定結果を図9のプッシュボタンボックスに与えることにより、決められた時間にシーンを自動で切り替えることが可能となる。

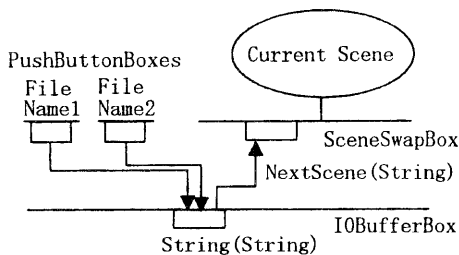


Fig. 9 SceneSwapBoxを用いたシーンの切替

4. おわりに

IntelligentBoxシステムで提供する機能部品は自由に組み合わせ可能である。アニメーションのための機能をもつ部品を組み合わせることで、ボトムアップにアニメーション作成が行える。本稿では、グラフィックスワークステーション上に実装したIntelligentBoxシステムのCGアニメーション作成のための機能と機能オブジェクトであるボックスについて説明した。いくつかのアニメーションの簡単な例を示すことで、アニメーション作成が容易に行えることを明らかにした。

現在、ユーザが対話的に操作した履歴情報を保存し、その情報を元にしてCGアニメーションを自動作成する機能について検討している。今後は、種々のアニメーションを実際に作成することにより、IntelligentBoxシステムの問題点を明らかにし、それらの問題点を検討し解決するつもりである。

参考文献

- [1] 岡田義広、田中譲：対話型3Dソフトウェア構築システム-IntelligentBox, コンピュータソフトウェア, Vol. 12, No. 4, pp. 84-94, 1995.
- [2] Okada, Y. and Tanaka, Y.: A Constructive Visual Software Development System for Interactive 3D Graphic Applications, Proc. of Computer Animation 95, pp. 114-125, Geneva(1995).
- [3] 長崎祥、田中譲：シンセティックメディア・システム-IntelligentPad-, ソフトウェア学会論文誌, Vol. 11, No. 1, pp. 36-48, 1993.
- [4] 岡田義広、田中譲：3次元オブジェクト機能合成システムにおける視野管理のための機能部品の導入, 情報処理学会第50回全国大会講演論文集(5), pp. 281-282, 1995.
- [5] 岡田義広、田中譲：IntelligentBoxにおけるコンピュータアニメーション作成環境について, 情報処理学会研究報告Vol.96, No. 29, pp. 17-24, 1996.
- [6] Sederburg, T. W.: Free-Form Deformation of Solid Geometric Models, ACM SIGGRAPH '86 Proceedings(1986), pp. 151-160.