

タッチインタフェースによるインタラクティブな3Dアニメーションシステムの提案

古川真行^{†1} 福元伸也^{†1} 赤木康宏^{†1} 川崎洋^{†1} 河合由起子^{†2}

タッチインタフェースを搭載したタブレット型のコンピュータが一般的なものとなってきた。これらの端末において3Dグラフィックスは主にゲームなどのコンテンツに用いられており、ユーザ側が用意した3Dデータを利用するコンテンツはまだ少ない。本システムでは、単に3Dデータを読み込んで表示させるだけでなく、複数の連続した3Dデータの集合から、互いの関係性が定義された状態遷移モデルを持つアニメーションを作成し、タッチインタフェースの直感的な操作性をジェスチャ入力という形で活用した新しい3Dアニメーションシステムを提案する。また、専門的な知識や技術のないユーザが上記システムを簡単に作成するために、既存のデバイスを用いたコンテンツ作成システムを合わせて提案する。

Interactive 3D Animation System based on Gesture Input of Touch Interface

MASAYUKI FURUKAWA^{†1} SHINYA FUKUMOTO^{†1} YASUHIRO AKAGI^{†1}
HIROSHI KAWASAKI^{†1} YUKIKO KAWAI^{†2}

Nowadays tablet devices are considered most important category of information-processing devices, because they attract not only PC users, but also all generation of people including small kids to elderly people. In addition, tablet devices rapidly increase their ability and 3D graphics sometimes exceeds desktop PC. However, 3D graphics is mostly used by games and only few well-known applications exist. In this paper, we propose a new 3D animation rendering system, which dynamically changes the story dependent on the users' gesture. We introduce a state transition model to efficiently achieve smooth transition of multiple sequence of 3D animations. We also propose a content creation tools which help user to easily make such 3D animation.

1. はじめに

Apple社のiPhoneやiPad, Google社のAndroid OSを搭載したタブレット端末が普及し、タッチインタフェースを搭載したコンピュータの利用が日常的となっている。さらに、パソコンおよびタブレット端末向けのOSであるMicrosoft社のWindows 8も発売され、タッチインタフェースの需要はさらに高まっている。

一方で、タブレット端末やPC上で利用されるコンテンツとして、映像や3次元データなどリッチなものが求められるようになってきている[1]。特に3次元データは、一般的なタブレット端末に3Dコンピュータグラフィックス用のAPIが搭載されるようになるなど、利用が進んでいる。しかし、現在の利用は主にゲームなどであり、それ以外での利用方法としては、例えば3Dビューワ等があるが、広く使われるには至っていない。これは、静止した3次元モデルを単に表示するだけのコンテンツでは魅力に欠けているためと思われる。

ダイナミックな3次元データを、直感的なインタフェースで操作できれば、これまでにないコンテンツとして、広く利用される可能性がある[2]。そこで我々は、3Dビューワの機能にタッチインタフェースの直感的な操作性を活用した3Dアニメーション機能を追加したシステムを提案する。これは単に3Dオブジェクトを時間軸に沿って動かすものではなく、複数の短いアニメーションの互いの関係性が定義された状態遷移モデルによってインタラクティブ

な操作を可能とする新しい3Dアニメーションである[3]。

このようなシステムが広く普及する鍵として、専門的な知識や技術のないユーザが、簡単にコンテンツを構築できることの重要性が認識されている[4]。そこで、我々は既存のデバイスを用いて、簡単に上記コンテンツを作成可能なシステムを合わせて提案する。アニメーションに用いられる複数の連続した3Dデータを入力として、それらアニメーション同士をスムーズに接続する状態遷移モデルの作成や、そのためのジェスチャの定義を、ユーザの操作に従って自動で構築するシステムとなっている。

実験では上記システムを利用してコンテンツを作成し、実際にタブレット端末に実装したアプリケーションにより、動作確認を行った。

2. 提案システム

2.1 システムの概要

提案システムは、コンテンツ作成システムと、コンテンツ表示システムの2つからなる。また、短いアニメーションを互いの関係性を保ったままつなぐためのデータ構造を定義する必要がある。さらに、ジェスチャにより、直感的な操作を可能にするための機構も必要である。以下では、そのようなデータ構造やジェスチャとの関係について説明し、それらを効率良く構築するためのコンテンツ作成システムと、表示するコンテンツ表示システムについて説明する。

2.2 状態遷移モデルによるデータ構造

本システムにおける 3D アニメーションとは、形状が少しずつ異なる 3D データの連続であり、複数の 3D アニメーションの関係を表したものを状態遷移モデルとする。状態遷移モデルは、開始状態を除いた各状態が短いアニメーションで構成される。ここで、2 つの状態間でアニメーションが物理的に滑らかに繋がっている場合、その 2 つの状態は双方向に遷移をもつと仮定する。この 2 つの状態間の繋がりを確認するプロセスを新しい状態を追加した際に逐次行うことで、破綻のないスムーズなアニメーションの繋がりをもった状態遷移モデルの作成が可能である[5]。

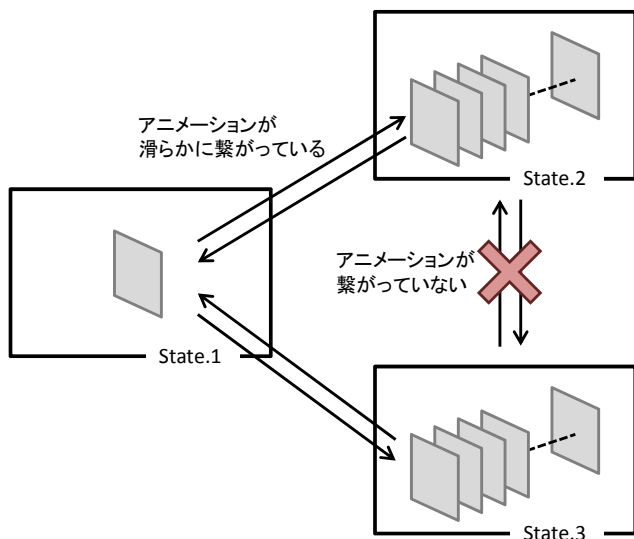


図 1 状態遷移モデル
Figure 1 State transition model.

2.3 ジェスチャと状態遷移モデルの直感的な対応付け

状態遷移モデルの状態から状態への遷移は、ジェスチャをトリガーとして実行される。直感的な操作性を実現するため、ジェスチャの向きはユーザがアニメーションの動作の様子から判断して決定する。

本システムでは、ジェスチャをトリガーとする状態遷移は、隣接している状態のみではなく、状態遷移モデル上のすべての状態を遷移先として選択することができる。状態遷移モデル上で隣接している状態への遷移は 1 種類のアニメーションを経由して実行すればよいが、隣接しておらず、直接的なアニメーションの繋がりが無い状態への遷移は、状態遷移モデルのパスを探索し、複数の遷移を経路として選択することで滑らかなアニメーションを伴った遷移を行う(図 2)。

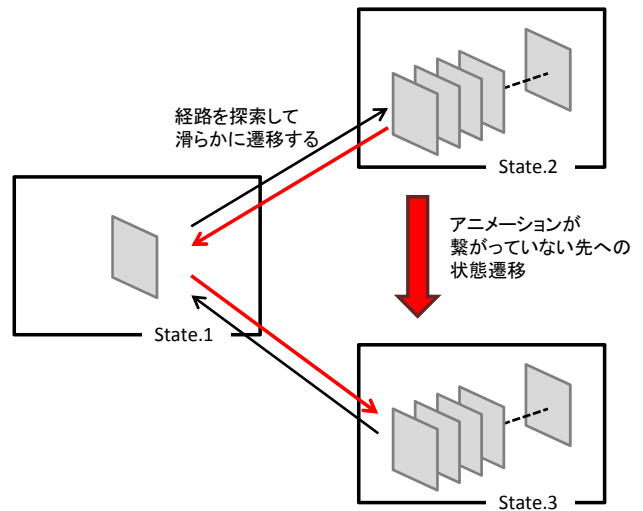


図 2 状態間の経路探索
Figure 2 Path search among states.

3. コンテンツ作成システム

3.1 短いアニメーションの集合の生成

動いている対象物を連続して撮影[6]することで、1 組の 3D アニメーションを生成する。撮影には Microsoft 社の Kinect[7]を使用する。Kinect から深度情報と RGB 画像を取得し、オープンソースのライブラリである Point Cloud Library[8]を用いて 2D データを 3D データへと変換する。ユーザはこの処理を必要な回数繰り返すだけで容易に 3D アニメーションの集合を作成することができる(図 3)。

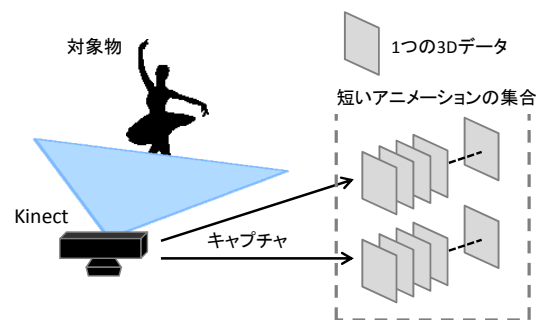


図 3 3D アニメーション生成システム
Figure 3 3D animation production system.

3.2 アニメーション間を考慮した状態遷移モデルの構築

3.1 節で生成されたアニメーションの集合に対して、アニメーションの物理的な繋がりと互いの関係性についての情報を、ユーザの手によって付与を行う。付与されたアニメーションの情報は、図 1 に示すような状態遷移モデルとして定義される。

また、図 4 に示すような、アニメーション情報の付与な

どのコンテンツの生成を補助する GUI を提供する .

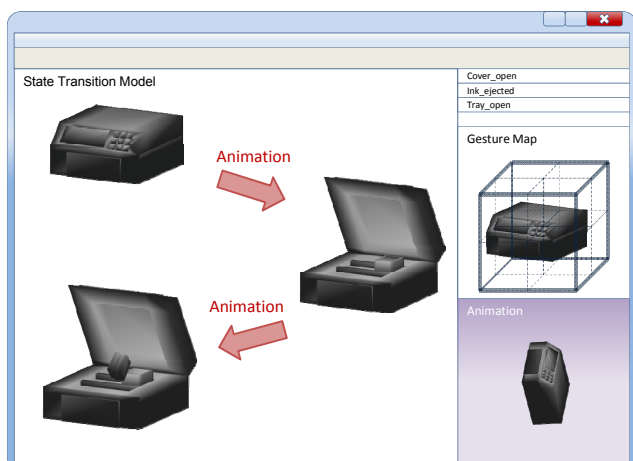


図 4 コンテンツの作成を補助する GUI
Figure 4 GUI for assist the creation of content.

3.3 ジェスチャ操作による動作の定義

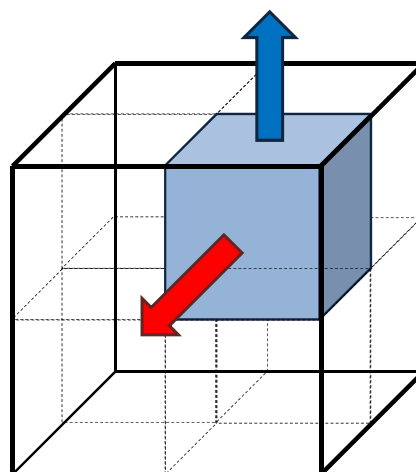
作成した状態遷移モデルを元に、ジェスチャ操作による状態遷移を定義する .

3.3.1 ジェスチャ開始座標の設定

状態遷移モデルが複雑化した場合、ジェスチャの方向のみを利用して状態遷移を行おうとすると、僅かなジェスチャの方向の違いによって異なる遷移が発生することが予想される . この場合、結果としてジェスチャの入力に高い精度が必要になり、直感的な操作が困難となる . そこで、ジェスチャ開始座標を設定するために、オブジェクトの座標系で 3 次元空間を分割し、ジェスチャが開始された 3 次元空間上の座標によって場合分けを行い、それぞれの空間ごとにジェスチャの向きと状態遷移先を設定することで、複雑な状態遷移モデルに対応する .

3.3.2 ジェスチャによる状態遷移先の設定

ジェスチャによる状態遷移先の設定は、分割された 3 次元空間に、ジェスチャの方向と状態遷移先を定義することで行う . ジェスチャの方向と、そのジェスチャによる状態遷移先は、ユーザが直感的な判断を行うことによって決定する . 例えば、上昇を伴うアニメーションによって遷移する状態が遷移先であれば上方向のジェスチャを用いて遷移を行うとし、下降を伴うアニメーションによって遷移する状態が遷移先であれば下方向のジェスチャを用いて遷移を行うとする .



2×2×2分割した3次元空間

図 5 ジェスチャ開始位置による空間分割
Figure 5 Space division for starting position of gesture.

4. コンテンツ表示システム

前章で作成したコンテンツをタブレット端末上のアプリケーションとして実装する .

4.1 3次元アニメーション表示システム

本システムでは、3D データの描画に OpenGL をはじめとした 3D 描画 API を用いる . アプリケーションは起動時に使用する 3D データをすべて読み込み、あらかじめレンダリングを行う .

3D アニメーションは、既にレンダリングされているオブジェクトを連続で切り替えて表示することで実行される . この方法により、オブジェクトの切り替えが発生する際の処理を減らし、アニメーション実行時の動作の遅延を少なくすることができる .

4.2 アプリケーションでの表示とジェスチャによる操作

この表示システムにおいて、オブジェクトを表示するタブレット端末の画面は 2 次元であるため、ユーザが入力するジェスチャも 2 次元的な情報となる . そのため、オブジェクトに対する 3 次元的なジェスチャ方向が正確に分からなくなってしまう . 本システムでは、これを改善するため、現在の視点におけるオブジェクトの物体座標系の XYZ 軸を 2 次元に射影したものとジェスチャの方向を比較し、最も近傍な軸をジェスチャの向きとして採用することで 3 次元的なジェスチャの入力を取得している . (図 6)

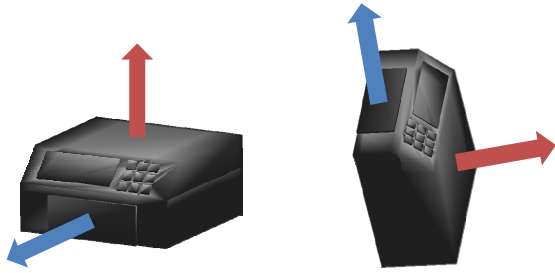


図 6 視点によるジェスチャ方向の変化

Figure 6 Directional change of gesture depending on observing point.

5. 実行例

本システムでは、人や物に問わず、様々なものの動的な状態を 3D アニメーションとして、コンテンツを作成することが可能である。ここでは 3D アニメーションの応用例として、一般的なオフィスでの複合機（プリンタ+FAX）の操作を想定し、電子的な操作マニュアルをインタラクティブ 3D アニメーションシステムのコンテンツとして作成する過程を述べる。

5.1 対象物の撮影

3.1 節で述べたように、3D アニメーションは Kinect を用いて撮影する。本項では、

1. 未操作の状態

を初期状態として、初期状態から発生する

2. スキャン部分を持ち上げて開く。
3. フロントカバーを開く。
4. 用紙トレイを開く。

の 3 通りの動作と、上記の状態から発生した動きである

5. 2 の状態から内部のトナーを取り出す。
6. 3 の状態から中の用紙を取り出す。

という 2 通りの動作の合計 5 通りのアニメーションを生成した。(図 7) それぞれのアニメーションは 20 個程度の連続する 3D データから構成されている。

5.2 状態遷移モデルの構築

生成した 5 通りのアニメーションから、それぞれのアニメーションの関係性を考慮して図 8 に示すような 6 つの状態を持つ状態遷移モデルを構築した。各状態は 5.1 節のリスト番号に対応しており、状態 1 以外の状態はそれぞれ図示されている形状までのアニメーションを持っており、状態遷移の方向に応じてアニメーションの通常の再生か逆方向の再生を行う。



図 7 撮影の様子

Figure 7 Scene of depth image capture.

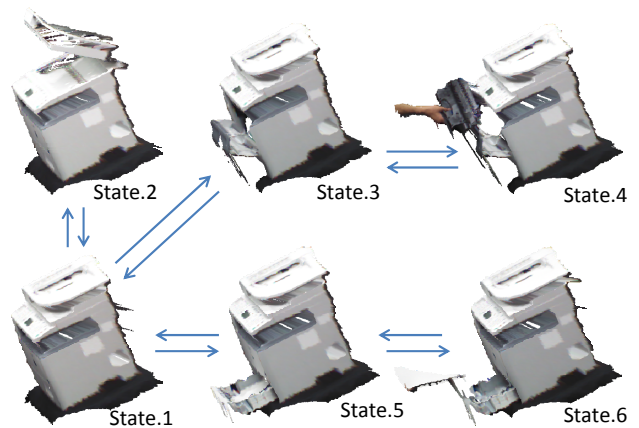


図 8 複合機の状態遷移モデル

Figure 8 State transition model of All-in-one printer.

5.3 ジェスチャの設定

タブレット端末上でアニメーションを直感的に操作できるよう考慮しながら、空間の分割の仕方やジェスチャの開始位置や方向、状態遷移先を決定した。今回使用した複合機の場合は、スキャン部分とフロントカバー、用紙トレイにアニメーションが存在するので、該当部分がジェスチャの開始位置になった場合に遷移が行われるような 3 次元空間の分割を行った。また、スキャン部分は該当部を上を持ち上げるというアニメーションなので、上方向のジェスチャをトリガーとしてアニメーションが発生するように設定する。他のアニメーションについても同様に設定を行い、図 9 に示すようなジェスチャマップを作成した。

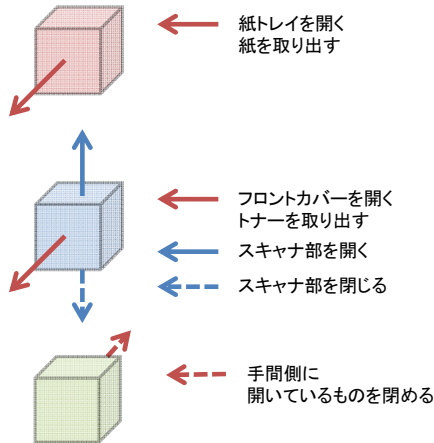
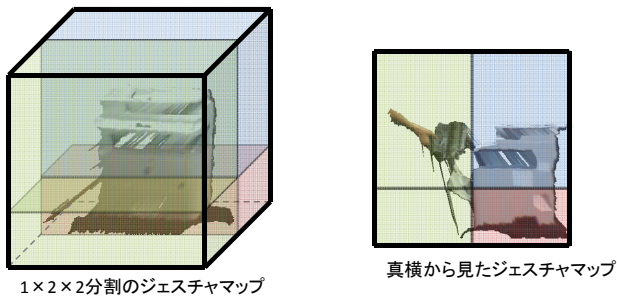


図 9 具体的なジェスチャマップ
Figure 9 Concrete gesture map.

5.4 タブレット端末上での利用

実際にタブレット端末のアプリケーションで本システムを動作させたものを図 10 に示す。今回使用した端末は ASUS 社の Eee Pad TF201 で、Android バージョンは 4.0.3 である。3D モデルの描画には OpenGL ES[9]を用いている。

本アプリケーションは起動時に利用されている全てのオブジェクトを読み込みプリレンダリングする。そのため、アニメーションに利用される 3D データが多い場合や、それぞれの 3D データが大規模であると起動時の処理に多くの時間が必要となる。今回の実行例である複合機においても、オブジェクトの形状がやや複雑であり、起動時の処理に時間を要した。

オブジェクトが表示されてからのアプリケーションの動作は、視点の移動とジェスチャによるアニメーションの双方が軽快に実行されていることを確認した。また、ジェスチャの開始位置による場合分けについても、5.3 節の設定で設定したものの同様の挙動を確認した。なお、視点の位置によってジェスチャの向きも変化しており、基本的な機能が正常に行われていることを確認した。

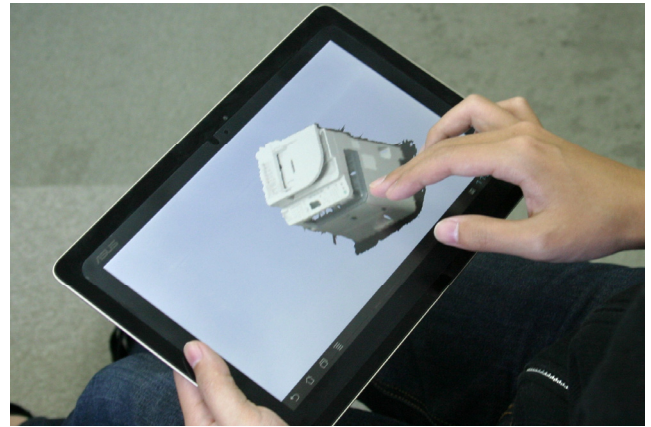


図 10 タブレット端末での操作の様子
Figure 10 Operation on Tablet PC.

6. 考察

本システムでは、3D アニメーションの作成に Kinect を用いて連続撮影を行っている。これは 3DCAD のような本格的なモデリングソフトを必要としない手軽さを得られる反面、撮影時に発生したノイズによって、1つ1つの 3D データの容量が大きくなってしまいうという問題がある。現在は 1つのアニメーションに使用する 3D データを少なくすることで対応しているが、ノイズを削除するなど、データ量を削減するための処理が必要であると考えられる。

また、対象物によってはアニメーションの構造やジェスチャ情報が複雑になる可能性がある。その解決策として、コンテンツの情報を視覚的に構築することを可能にする GUI を提供することで、アニメーションシステムの開発を効率的に行う環境を提供することを検討している。

タブレット端末側のシステムにおいては、現段階では、機能がジェスチャによる入力とオブジェクトの回転のみとなっている。今後、インタフェース上にジェスチャ操作のガイドやオブジェクトの情報を表示するなどして、より使いやすいインタフェースを提供する必要がある。

7. 結論

我々は、タッチインタフェースを搭載したタブレット型コンピュータ上での 3D コンテンツの価値を高めるために、効果的な 3D アニメーションと、3D アニメーションを容易に制作するためのシステムについて提案した。本システムでは、Kinect を用いて誰でも 3D アニメーションのためのデータを作成することが可能であり、オブジェクトの向きとジェスチャの向きを活用することで、タブレット端末上でのアニメーションの操作をより直感的に行うことが可能である。

今後は 3D アニメーションの構造をより容易かつ直感的に構築できるようなインタフェースの実装と、タッチインタフェース上での操作性の改善を行っていく。

謝辞

本研究の一部は、総務省戦略的情報通信研究開発制度 (SCOPE) ICT イノベーション創出型研究開発 (101710002)、内閣府・最先端・次世代研究開発支援プログラム(LR030) および JSPS KAKENHI 24500120 の助成を受けて実施されたものである。ここに記して謝意を表す。

参考文献

- 1) M. Ishikawa, H. Kawasaki, R. Furukawa, Y. Kawai: Shape Rank:Efficient Web3D search technique using 3D features, SIGMM IR, pp.393-396, 2010.
- 2) R. Furukawa, H. Kawasaki: Interactive Shape Acquisition Using Marker Attached Laser Projector, 3DIM 2003, pp.491-498, 2003.
- 3) M. Furukawa, S. Fukumoto, H. Kawasaki, Y. Kawai: Interactive 3D Animation System for Web3D, ICME 2012, p.666, 2012.
- 4) Y. Kawai, S. Tazawa, R. Furukawa, H. Kawasaki: Efficient Meta-information Annotation and View-dependent Representation System for 3D Objects on the Web, ICPR08, pp.1-4, 2008.
- 5) 四方正輝, 灘本明代, 角谷和俊, 田中克己: 携帯端末を用いた機器マニュアル情報の検索および自動呈示, 情報処理学会論文誌データベース, Vol.44, No.SIG 3, pp.65-77, 2003.
- 6) 池村翔, 藤吉弘亘: 距離情報に基づく局所特徴量によるリアルタイム人検出, 電子情報通信学会論文誌 D, Vol. J93-D, No.3, pp.355-364, 2010.
- 7) Microsoft Kinect:
<http://www.microsoft.com/en-us/kinectforwindows/>
- 8) Point Cloud Library: <http://pointclouds.org/>
- 9) OpenGL ES: <http://www.khronos.org/opengles/>