

## 両手デバイスを用いた 2物体の動きの同期手法

寺本興英<sup>†</sup> 五十嵐健夫<sup>††</sup>

本論文で、我々は2つの入力デバイスを用いることで2つの物体の動きの同期を取る手法について提案する。互いに関連性のある物体の動きを実時間での操作によって入力する際に、既存の手法では複雑な軌跡の生成や物体同士のタイミングの決定を行うことが困難であった。本手法では既に動きが決まっている物体の軌跡をタイムラインとし、ユーザは片方のマウスで軌跡をなぞりながらもう片方のマウスで新しい物体の軌跡を決定する。本手法の有用性を確かめるために、我々はユーザーテストを行った。

## Two-Handed Interaction for the Synchronization of Paired Object Motion

OKIHIDE TERAMOTO<sup>†</sup> TAKEO IGARASHI<sup>††</sup>

In this paper, we present the method of synchronizing the motion of two objects by using two-handed input device. We assume that their motion paths are relevant to each other. When user inputs the motion of objects in real time, it is very hard using the existing system to generate complex motion paths or decide the timing of objects. Our method regards the specified motion path of the object as time line. User traces a trajectory by one mouse and assigns the new motion path by another mouse. To verify the usability of our method, we also conduct a user test.

### 1. はじめに\*

初心者を対象とするアニメーション制作のインタフェースを提供する研究は過去に多数のものがなされてきた。しかし既存研究で用いられるアニメーション内のキャラクターの動きを決定するための手法には一長一短があり、ユーザは状況に応じて各手法の使い分けを行ってきた。本研究では動画内のキャラクター同士のインタラクションに着目し、ユーザが自由にキャラクターの動きを生成しつつ他のキャラクターとのインタラクション—衝突、並走、動きの同期など—にも対応可能な軌道の決定手法について提案する。

具体的な方法として、本手法では2つの入力デバイスを用いて操作を行う。ユーザは既に動きが決まっているキャラクターの中から1つを選び、その軌道を片方の手でなぞる。そしてもう片方の手で新しく動きを記録したいキャラクターを操作する。システムは既に決まっているキャラクターの軌跡と現在ユーザがなぞっている地点から時間を逆算し、新しく動きを決定するキャラクターに対してその時間と位置とを適用する。本論文は提案手法の利点、そして提案手法を実現する上での問題点とその解決法について紹介した後にユーザーテストとその結果を述べる。

### 2. 関連研究

初心者にも簡単なアニメーション作製を可能にする手法として、過去に数多くの提案がなされてきた。代表的なものとして、コンピュータ上で作成あるいは外部から取り込んだ静止画やオブジェクトに対して移動・変形などを行い、それを記録する手法[1][2]、ユーザがカメラなどの機器の前で何らかのパフォーマンスを行ってそれを利用する方法[3][4]、録画したビデオを編集あるいはフレームをつなぎなおして新しいアニメーションを生成する方法[5][6]などが挙げられる。またその他にも制約のもと物理シミュレーションを行う方法[7]もあるが、一般にこの手法はユーザからのコントロールが困難であり、また物理的制約にとらわれない自由なアニメーションを作るのには不向きである。本研究はプログラムなどの知識を必要としないアニメーション制作手法を目的としている。

アニメーション内でのキャラクターの動きを決定する手法についても過去に多くの提案がなされており、以下にその代表的なものとその手法の限界について考察を述べる。本来アニメーションは大量のセル画を作る必要があり、これには多くの時間と労力が必要であった。そこで動画内に幾つかのキーフレームを作成し、その間を補完することでアニメーションを作成する手法[8][9]が提案された。補完によってユーザ

\* <sup>†,††</sup> 東京大学 大学院情報理工学系研究科 Information Science and Technology, The University of Tokyo  
E-Mail: <sup>†</sup>tera1984@ui.is.s.u.tyoko.ac.jp, <sup>††</sup>takeo@acm.org

の労力は大幅に削減され、簡単にアニメーションを作ることができるようになった。しかし一般に補完によって生成される軌道は単純なものに限られ、ユーザが自然で複雑な動きを生成しようとするときだけ多くのキーフレームを作成する必要がある。

一方でユーザが軌道をあらかじめ入力しておき、それに沿ってキャラクターを動かすという手法[5][10]も存在する。この手法は複雑な軌道でも簡単に作成することができるが、ユーザがアニメーションの結果を実行するまで知ることができないという欠点がある。また軌道の決定は時間軸と独立しているため、アニメーション内にあるキャラクターの動きの速度に関する制約が存在する。

この時間に対する制約を解決する方法として、キャラクターに対する操作をリアルタイムに記録する手法[1][2]が挙げられる。この方法は非常にシンプルかつ直感的である反面、キャラクター間のインタラクションをとるのが困難であり、タイミングのシビアなアニメーションを作るには不向きであった。

キャラクターの軌道の生成を個別に行うのではなく、複数のキャラクターを同時に操作するという手法[3]も提案された。この手法はキャラクター間のインタラクションが簡単に設定できるが、ユーザの意識が分散されるためにリアルタイムに正確な動きを入力することは困難な作業になる。

先にキャラクターの軌道を確定させておき、ドラッグ操作によって時間軸の調節を行いキャラクター同士の同期をとる手法[11]も提案されている。この手法は非常に直感的で、タイミングの調節を容易に行うことができるが軌道そのものを生成あるいは変更することはできず、その部分は他の手法に頼らざるを得ない。

またインタラクションに着目した研究も数多くされており、3D モデル同士の接触、衝突についてトポロジーを用いて処理するもの[12]、人ごみの中での衝突回避のインタラクション[13]、スペクトルを用いて隊列の制御をおこなうもの[14]などの研究が既に発表されている。

一方でマルチデバイスを利用した研究も多数行われてきたが、それらの多くはデバイスを仮想空間の操作に利用するもの[15][16]や医療への応用[17]であり、平面空間内のオブジェクト同士のインタラクションを扱う研究はこれまでされてこなかった。

動画を見る際にシークバーを操作する代わりに動画内のキャラクターを直接ドラッグして操作する手法[18]が存在する。この論文では、本論文で提案する手法を実現する上で解決しなければならない問題の多くについて議論を行っている。詳細については3. 提案手法にて述べる。

### 3. 提案手法

本節ではまず提案手法の詳細と利点について述べ、次にアルゴリズムを紹介したの

ちにそれを実現する上での問題点およびその解決法を一つずつ紹介する。

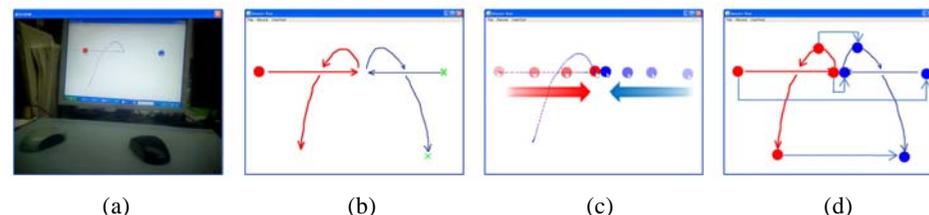


図 1 システムの概要

(a) 使用するデバイス (b) 作成する動きの例 (赤：既に動きが決定されている物体とその軌跡, 緑：スタート/ゴール地点, 青線：新たに作成する軌跡) (c) 実際の操作手順 (d) 時間の推定と各フレームにおけるキャラクター座標の決定

#### 3.1 提案手法の詳細と利点

本手法は2つの入力デバイスを使い (図 1a), 片方の手で既に決定されたキャラクターの軌道をなぞり (図 1c), 軌道上の空間的な位置から時間を推定してその時間における新しく動きを生成するキャラクターの位置を決定する (図 1d). 本手法の最大の特徴は時間的な制約を空間的な問題に置き換えているという点であり、これによってユーザにとっては以下のような利点がある。

まずユーザは時間にとらわれることがないため、入力を焦る必要がないという点である。一般に多入力デバイスの操作はユーザの意識を分散させるという点から複雑な軌道をリアルタイムになぞるのは困難であるが、本手法ではその負担をある程度軽減できることが期待される。

次にユーザが既に動きが決まっているキャラクターの軌道をなぞる際に、必ずしもその軌道を正確になぞる必要がないという点である。ユーザは対象とする軌道の近くを大まかになぞるだけでシステムが自動的に軌道上の位置を推定し、その地点を用いて時間を割り出せばよい。これもユーザの負担を軽減し新しく決定するキャラクターの軌道に集中することができるようになることと期待される。

また本手法を用いる際に、キャラクターの軌道を利用する代わりにキャラクターとは独立したスライダーバーのようなものを用意し、それをドラッグするという方法も考えられたが、互いにインタラクションを行うもの同士を直接操作する方がユーザにとってより直感的なインタフェースであると考えた結果、我々は現在提案する手法を採用することにした。

### 3.2 アルゴリズム

アニメーション内において、時間 $t$ におけるキャラクター $c_1$ の位置 $p_1$ は以下のようにして定義される。

$$p_1 = P_{c_1}(t)$$

基本的な考え方としてユーザがなぞっている地点 $p$ から適切な時間を推定するためには、決定された軌跡上にある点のうち、 $p$ に最も近い地点にキャラクター $c_1$ が到達する時間を用いれば良い。すなわち、

$$t = \operatorname{argmin}_t \left( \operatorname{Distance}(p, P_{c_1}(t)) \right)$$

となる。 $\operatorname{Distance}$ は一般のアニメーションにおいてはユークリッド距離を用いれば十分であるため、以下のように定義される。

$$\operatorname{Distance}(p, P_{c_1}(t)) = \sqrt{(p_x - p_{1x})^2 + (p_y - p_{1y})^2}$$

よって、新しく動きを決定するキャラクター $c_{new}$ の軌跡は、先ほどの $p$ 、 $p_1$ および $c_{new}$ の位置 $p_{new}$ を用いて以下のように決定される。

$$P_{c_{new}} \left( \operatorname{argmin}_t \left( \operatorname{Distance}(p, P_{c_1}(t)) \right) \right) = p_{new}$$

### 3.3 本手法の問題点および解決法

前節で我々は、軌道決定の際の時間推定をユークリッド距離の最小化によって行っていた。しかし、単純にこの方法を適用すると、以下に紹介する特殊なケースにおいてユーザの期待に沿わない推定を行ってしまう。

Dragicevic ら[18]は自身の論文で一部これに類似した問題について議論を行っている。ビデオのブラウジングと動画の作成とは前提とするものが違うため、詳細についてはそちらの論文も参照されたい。

まず1つ目の問題は空間的方向性の保持が必要な点である。図2aに示されるような軌道をユーザがなぞる際に、一般にユーザは一度通ったパスに戻るよりも、まだ通っていないパスに進んでいくことを望んでいると考えられる。しかし単純なユークリッド距離の最小化ではこれが保証されるとは限らない。そこで、我々のシステムでは一度通ったパスを記録しておき、距離計算時に重みづけをすることでこの問題を解決している。

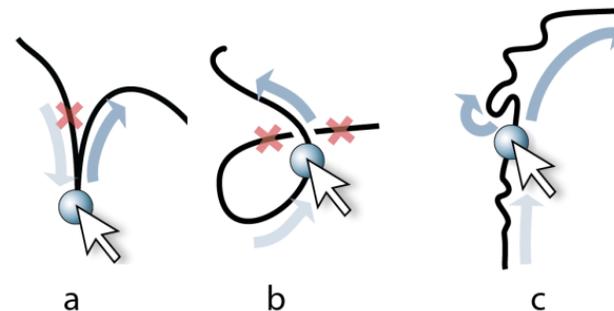


図 2 問題となる例

(Video Browsing by Direct Manipulation より引用)

すなわち、軌跡上の各点 $p_i$ からユーザのなぞっている地点 $p$ までの距離の計算式は

$$\operatorname{Distance}(p, p_i) = \sqrt{(p_x - p_{ix})^2 + (p_y - p_{iy})^2} + \operatorname{Passed}(p_i)$$

のようになる。ここで、 $\operatorname{Passed}$ は一度通ったパスに対して重み付けをする関数であるが、新しく軌道を作成する場合、既に決定された軌道についてユーザは変更を望んでいないと考えられるため、我々は $\operatorname{Passed}$ を以下のように定義した。

$$\operatorname{Passed}(p_i) = \begin{cases} +\infty & \text{(if } i \text{ th point is passed)} \\ 0 & \text{(otherwise)} \end{cases}$$

次に2つ目の問題として、時間的な連続性が保たれていなければならないという点が考えられる。図2bに示されるような軌道をユーザがなぞる際に、一般にユーザは直前になぞっていた地点の付近の点が参照されることを望んでおり、時間的に離れたところにある点が参照されることは望ましくないと考えられる。単純なユークリッド距離のみの最小化ではこの要望は無視されてしまうため、我々のシステムではそれ以外にも、直前に参照していた点からのパスに沿った道のりを計算し、その値に応じて計算の対象に含めるかを判断した。すなわち、直前まで参照していた点を $p_{prev}$ としたとき、

軌跡上の各点 $p_i$ からユーザのなぞっている点 $p$ までの距離の計算式は

$$\operatorname{Distance}(p, p_i) = \sqrt{(p_x - p_{ix})^2 + (p_y - p_{iy})^2} + \alpha$$

$$\alpha = \begin{cases} +\infty & (\text{Path}(p_{prev}, p_i) > \beta) \\ 0 & (\text{otherwise}) \end{cases}$$

となる。ここで  $\text{Path}$  は受け取った 2 点間のパスに沿った道のりの長さを計算する関数であり、以下のように定義される。

$$\text{Path}(p_i, p_j) = \sum_{k=i}^{j-1} \sqrt{(p_{kx} - p_{(k+1)x})^2 + (p_{ky} - p_{(k+1)y})^2} \quad (i < j)$$

また  $\beta$  は計算の対象に含めるかどうかの閾値であり、この詳細については次の問題も交えて述べる。

3 つ目の問題は、システムはユーザが軌道をなぞる際の局所的な動きと大局的な動きとを区別しなければならないという点である。図 2c に示されるような軌道をユーザがなぞる場合、ユーザが大局的な動きを望んでいるにもかかわらず、局所的に引っかかりが生じることでユーザの意図に沿わない結果が得られてしまう。このような局所的な引っかかりは先ほどの時間的な連続性を考慮した結果、副次的に生じるものである。我々はこの問題を解消するには、先ほどの式で閾値となる  $\beta$  の値を動的に変化させる必要があると考えた。ユーザが大局的な動きを望んでいるのか、あるいは局所的な動きを望んでいるのかを判断する方法として、我々は直前までユーザが参照していた地点と、実際にユーザがなぞっていた地点との距離を利用することにした。基準とする軌道からユーザが遠くをなぞったり、あるいは引っかかりを無視して遠くの方にドラッグを続けたりした場合は大局的な動きを、そして基準となるパスの近くをなぞっている場合は局所的な動きをユーザは期待していると仮定した。すなわち、閾値  $\beta$  は以下のような動的な値となる。

$$\beta = k \times \sqrt{(p_x - p_{prevx})^2 + (p_y - p_{prevy})^2}$$

ここで  $k$  は適当な定数である。

4 つ目の問題は、基準となるキャラクターが空間で停止した状態でも新しく生成する動きの軌跡を記録できなければならないという点である。本システムでは時間的な制約を空間的な問題に置き換えている。そのためキャラクターが空間内で静止している間の正確な時間を割り出すことができず、新しく生成する動きの軌跡を決定できないという問題が生じる。我々はこの問題の解決策として、既に確定されたキャラクターを時間の基準として選択した際に前処理を行い、軌跡の中でキャラクターが静止する位置をピックアップしておく。そして新しく軌道を作成してその箇所に差し掛かった時だけ実時間で入力に切り替えることでこの問題を解決している (図 3)。

一般には基準となるキャラクターが動画内で静止している時間と、新しいキャラクターの動きの決定にかかる時間は等しくない。そのため、我々のシステムでは必要に

応じてフレーム数を補完によって増やす、あるいは減らすことによってこの問題に対応している。

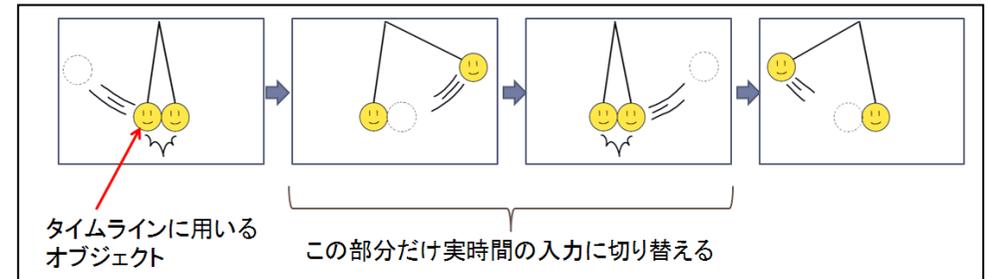


図 3 基準となるキャラクターが静止した状態での動きの記録

#### 4. ユーザーテスト

我々は予備実験として本システムをユーザに利用してもらい、他の動き決定手法との比較を行った。またユーザからのフィードバックをもとに、今後行うべき拡張についての議論を行った。ユーザーテストにあたって、我々は 2 名の被験者を用意した。いずれの被験者もアニメーション制作の経験のない初心者である。

まず 1 つ目の実験として、我々は図 4 に示されるような 2 つのタスクを用意し、ユーザにキャラクター同士のインタラクションのある動きをそれぞれの手法で作成してもらった。比較対象はキーフレーム (KeyFrame)、1 つのマウスで個別に入力 (Single Mouse)、1 つのマウスで入力を行った後にタイミングを調節する (Single Mouse + Timing)、2 つのマウスを使って同時に入力 (Double Mouse)、そして我々の提案手法 (Our Method) である。基本的に動きの入力は既に動きが決まっている物体の動きを見ながらユーザが新しい物体の入力を行うものであるが、Double Mouse の手法に限ってはもともと動きが決まっていた物体に対しても改めて動きの入力を行うことにした。そしてユーザが納得のいく結果を作成するために要した時間と結果におけるインタラクションの正確さを調べた。最後に被験者に対してアンケートを行い、ユーザにとっての使いやすさを調査した。

インタラクションの正確さの数値化はいくつかの方法が考えられるが、今回我々はそれぞれの動画において 2 つのキャラクター (ボール) が局所的に最も近づくとときのキャラクター同士の距離を求め、その値と 2 つのボールの半径の合計との差を評価することにした。

次に2つ目の実験として、我々の手法を使ってユーザにインタラクションが存在するアニメーションを実際に制作してもらった。そして提案手法についていくつかのコメントを得た。

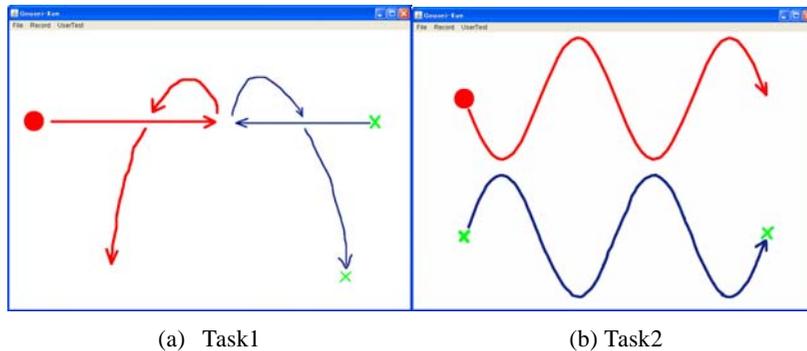


図4 ユーザーテストのタスク

赤：既に動きが決定されている物体とその軌跡 緑：スタート/ゴール地点  
 青線：被験者に作成してもらった軌跡（目安として）

## 5. 結果と考察

ユーザーテストの結果は図5のようになった。また、テスト後に被験者に対して行ったアンケートの結果は図6のようになった。これらを総合して見ると、我々のシステムは数値の上では良い結果を出しているものの、ユーザからの評価はあまり良いとは言えない。これに関してユーザからのコメントを交えて考察した結果、原因はハードとソフトの両面にあると考えられる。ハード面の問題としては、そもそも逆手側でのマウス操作にユーザが不慣れであるという点が第一に挙げられる。これは両手を使って一度に入力を行う方法の評価が低い原因でもありと考えられる。また、新しく動きを決定するための操作を利き手で行うか、あるいは逆手で行うかによっても操作の利便性は変わってくる。これに対する解決策として、我々はマルチタッチデバイスの利用を考えている。通常、逆手側でのマウス操作はユーザにとって慣れないものであるが、指をさす程度ならば逆手でも十分に可能な操作であり、またより直感的な操作であるため、利便性は増すものと考えられる。近年、安価なマルチタッチデバイス[19]が提供されるようになったこともあり、今後のハードウェア面での改善は期待できる。

ソフト面の問題としては、基準となるキャラクターの軌跡からの時間推定や未決定の部分に対する補間が満足のいくものではないという意見が見られた。これらの問題について、時間推定についてはより多くの例について実験を行い、適切なパラメータ設定をすることで改善がみられると考えられる。補間の方法についても今後より良い方法について模索をしていきたいと考えている。

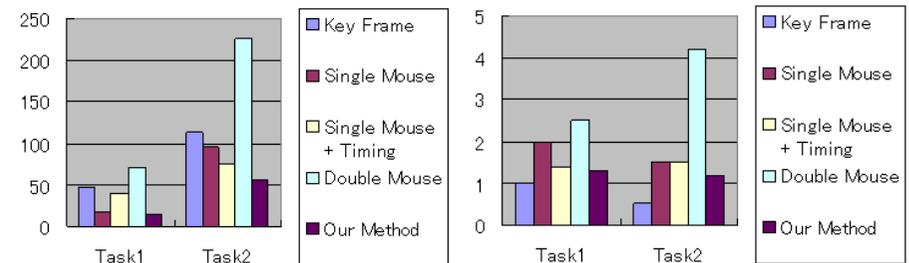


図5 ユーザーテストの結果 左：時間（単位は秒），右：誤差（単位は pixel）

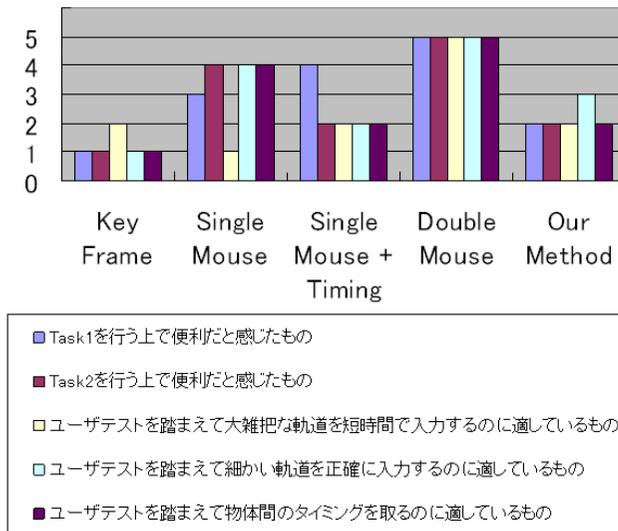


図6 被験者による投票（数字は同順ありの順位）

また本システムを用いて実際にユーザにアニメーションを作ってもらった(図7)。ユーザからはキャラクター同士のインタラクションを取るのには非常に便利だとのコメントを得た。しかし、両手を同時に動かすのは複雑な動きをさせるのには向かないという意見も見られた。



図7 被験者が作成したアニメーション

## 6. まとめ

本論文で我々は、2入力のデバイスを用いた新しい動画の合成手法についての提案を行った。本手法によりユーザは動画内でのキャラクターの動きを他のキャラクターとのインタラクションも含めて決定することが可能になる。またユーザーテストによって動きを決定するための各手法の向き不向きを実際に調査し、我々の手法がどのような動きの生成に適しているかを確かめた。

### 謝辞

本研究を行う上で多くの助言を頂いた研究室の多くのメンバー、そしてユーザーテストにご協力頂いた参加者の皆様にこの場を借りて謹んで感謝の意を表します。

## 参考文献

- 1) Moving Sketch Takeo Igarashi  
“www-ui.is.s.u-tokyo.ac.jp/~takeo/research/rigid/movingsketch/index.html”
- 2) Richard C. Davis, Brien Colwell, and James A. Landay. K-sketch: a 'kinetic' sketch pad for novice animators. In CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pp. 413-422, New York, NY, USA, 2008. ACM.
- 3) Connelly Barnes, David E. Jacobs, Jason Sanders, Dan B Goldman, Szymon Rusinkiewicz, Adam Finkelstein, and Maneesh Agrawala. Video Puppetry: A performative interface for cutout animation. ACM Transactions on Graphics (Proc. SIGGRAPH ASIA), Vol. 27, No. 5, December 2008.

- 4) Andrew J. Davison, Jonathan Deutscher, and Ian D. Reid. Markerless motion capture of complex full-body movement for character animation. In Proceedings of the Eurographic workshop on Computer animation and simulation, pp. 3-14, New York, NY, USA, 2001. Springer-Verlag New York, Inc.
- 5) Arno Sch'odl and Irfan A. Essa. Controlled animation of video sprites. In SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 121-127, New York, NY, USA, 2002. ACM.
- 6) Jue Wang, Steven M. Drucker, Maneesh Agrawala, and Michael F. Cohen. The cartoon animation filter. ACM Transactions on Graphics, Vol. 25, No. 3, pp. 1169-1173, 2006.
- 7) Christopher D. Twigg and Doug L. James. Many-worlds browsing for control of multibody dynamics. In SIGGRAPH '07: ACM SIGGRAPH 2007 papers, p. 14, New York, NY, USA, 2007. ACM.
- 8) Adobe Flash “adobe.com/jp/products/flash/”
- 9) Takeo Igarashi, Tomer Moscovich, and John Hughes. Spatial keyframing for performance-driven animation. In 2005 ACM SIGGRAPH / Eurographics Symposium on Computer Animation, pp. 107-116, July 2005.
- 10) M. Thorne, D. Burke, and M. Panne. Motion doodles: An interface for sketching character motion, 2004.
- 11) Manmyung Kim, Kyunglyul Hyun, Jongmin Kim, and Jehee Lee. Synchronized multicharacter motion editing. ACM Trans. Graph., Vol. 28, No. 3, pp. 1-9, 2009.
- 12) Edmond S. L. Ho and Taku Komura. Character motion synthesis by topology coordinates. In P. Dutr'e and M. Stamminger, editors, Computer Graphics Forum (Proc. Eurographics 2009), Vol. 28, Munich, Germany, March 2009.
- 13) Takeshi Sakuma, Tomohiko Mukai, and Shigeru Kuriyama. Psychological model for animating crowded pedestrians: Virtual humans and social agents. Comput. Animat. Virtual Worlds, Vol. 16, No. 3-4, pp. 343-351, 2005.
- 14) Shigeo Takahashi, Kenichi Yoshida, Taesoo Kwon, Kang Hoon Lee, Jehee Lee, and Sung Yong Shin. Spectral-based group formation control. Comput. Graph. Forum, Vol. 28, No. 2, pp. 639-648, 2009.
- 15) Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. Voodoo dolls: seamless interaction at multiple scales in virtual environments. In I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics, pp. 141-145, New York, NY, USA, 1999. ACM.
- 16) Robert C. Zeleznik, Andrew S. Forsberg, and Paul S. Strauss. Two pointer input for 3d interaction. In SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics, pp. 115-ff., New York, NY, USA, 1997. ACM.
- 17) Lawrence D. Cutler, Bernd Fr'ohlich, and Pat Hanrahan. Two-handed direct manipulation on the responsive workbench. In SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics, pp. 107-114, New York, NY, USA, 1997. ACM.
- 18) Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowicz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. Video browsing by direct manipulation. In CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pp. 237-246, New York, NY, USA, 2008. ACM.
- 19) iPod touch “www.apple.com/jp/ipodtouch/”