

マルチタッチデバイスを用いることによる 3Dモデルからのアニメーション生成手法

高原健輔¹ Natapon Pantuwong² 吉川毅¹ 野中秀俊¹ 杉本雅則¹

概要: 本稿では、マルチタッチデバイスを用いて、どこでも誰でも簡単にアニメーションを生成できる手法を提案する。提案手法では、物理的な人形等をキャプチャした3Dモデルから自動的にスケルトンを抽出する。得られたスケルトンの複数箇所を同時に操作することで、効率よく所望の動作をデザインできることを目指す。提案手法を実装したアプリケーションを作成し、パイロットスタディを通してその改善点を調査した。

A Character Animation Technique from 3D Model Using Multi-Touch Device

KENSUKE TAKAHARA¹ NATAPON PANTUWONG² TAKESHI YOSHIKAWA¹ HIDETOSHI NONAKA¹
MASANORI SUGIMOTO¹

Abstract: In this paper, we propose a technique that allows novice users to create character animation with a mobile multi-touch device. The proposed technique first elicits skeletons from a 3D model of a physical puppet. Users then choose multiple parts of the skeletons and simultaneously manipulate them to create a pose of a character in an efficient manner. We have conducted a pilot study using a prototype system to clarify problems to be solved and functions to be improved.

1. 序論

近年のコンピュータの性能向上と技術の発展によって、高品質な3次元コンピュータグラフィックス（以下3DCG）アニメーションが身近な存在になっている。また商用あるいはフリーのコンピュータアニメーションソフトの入手も容易になりつつある（[1]など）。

3DCGアニメーションとは、3次元の仮想空間内の立体物を2次元である平面上の情報に変換することで奥行き感を演出する映像のことである。3DCGアニメーション生成の手法は数多くあるが、生物のように複雑な動きをして3Dモデル自体が変形するキャラクターアニメーション手法としてスケルトン法が挙げられる。スケルトン法は骨格を階層的に連結したモデルであるスケルトンを特定の姿勢へアニメーションさせ、それに従って3Dモデルの表面構成点

を描画する手法である。

現在発売されている3DCGアニメーション制作ソフトで、スケルトン法によりキャラクターのアニメーションを制作するためには膨大な作業量が必要である。多くのソフトは、広く一般に普及しているマウスやタッチパッドなどによる操作にしか対応していない。マウスやタッチパッドは単一のポインタしか持たないため、キャラクターのモーション編集には一つ一つの部位の動作を逐次的に決定しなければならない。よって既存のソフトでは膨大な編集作業が必要となり、プロフェッショナルなコンピュータアニメーターにとっても簡単にアニメーション制作ができるわけではない。

本研究では、マルチタッチデバイスを用いて3Dモデルからのアニメーション生成手法を提案する。マルチタッチデバイスの利点は、2点以上の入力を同時に行うことができ、大がかりな装置を用いることなく直感的にタッチパネルで操作できることにある。マルチタッチデバイス上に表

¹ 北海道大学

² King Mongkunt's Institute of Technology Ladkrabang

示されるスケルトンに対して、「タップ」「ドラッグ」「ピンチイン・アウト」操作により、スケルトンのモーションを簡潔に編集し 3DCG アニメーションを生成する。提案するスケルトン操作インターフェースでは、マルチタッチにより同時に複数箇所の操作を行うことが可能となる。

2. 関連研究

モーション編集やモーション取得のためのインターフェースは、これまでに多数提案されている。Li ら [2] はスケッチによって key pose を作成し、モーションデータの key frame 中のそれとの比較によりモーション取得を行う手法について述べている。また、ユーザ自身が行った動作をカメラ [3] や圧力センサ [4] で取得することにより、モーションデータベースからのモーション取得を行う方法も提案されている。これらは、ユーザに描画のスキルやパフォーマンスを要求する点が問題となる場合がある。一方、物理的な人形に対する直接的な入力によりモーション編集や取得を行うために、視覚的なマーカー [5][6]、磁気トラッカー [7]、センサ [8] 等を取り付ける手法が挙げられる。これらはユーザに対して直感的な操作手法を提供できる反面、マーカーやセンサ等を新たに用意しなければいけない点が欠点である。

Oshita[9] はインバースキネマティクスを用いたマルチタッチ入力による自由で多様なアニメーション生成を提案している。インバースキネマティクスはポーズの操作には都合がよいが、正確で連続的なアクションの操作に向かない。この問題を解決するためにパンチ・キック・ジャンプのようなそれぞれのアクションに 1 種類ずつの学習済みモデルを用意して、ユーザの入力とそれぞれの学習済みモデルから得たキャラクターの動いた軌跡の曲線とを比較することでアクションを選択し、自然で連続的なアニメーション生成を行っている。

3. 提案手法

本研究では、マルチタッチ入力を用いて人型 3D モデルのスケルトンを操作することによるアニメーションの生成手法を提案する。提案手法は次の 3 段階の行程に分かれる。

- (1) 3D モデルからスケルトンを生成する
- (2) マルチタッチデバイスによりスケルトンを操作しモーション編集を行う
- (3) モーション編集から得られた情報よりアニメーションを生成する

3.1 3D モデルからのスケルトン生成

Natapon ら [10][11] は、与えられた 3D モデルに対して疑似法線ベクトル場を用いてスケルトンを生成し、そのスケルトンに対してスケルトンデータベース中のテンプレートと照らし合わせて適切な関節を与えている。疑似法線ベ



図 1 使用する 3D モデル

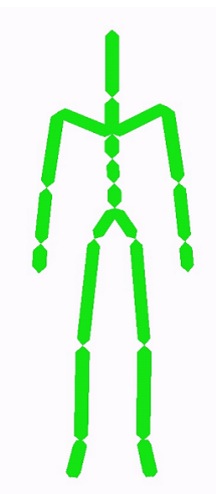


図 2 3D モデルから生成されたスケルトン

クトル場を用いることによって従来のスケルトン生成よりも十分に早く、かつノイズにロバストなスケルトンが生成できる。また生物であれば任意の 3D モデルに対してもスケルトンが生成できるため、アニメーション生成の幅がより広がる。本稿ではモバイルマルチタッチデバイスを用いるため、できるだけ計算量が小さい手法が望ましい。Natapon らの手法では従来のポテンシャル技法 [12] よりも小さい計算量でスケルトン生成が可能であり、よって本稿ではこの手法を利用する。

本研究に用いる 3D モデルは成人女性の 3D モデル (図 1) である。この 3D モデルに上記のスケルトン生成アルゴリズムを用いて、操作対象のスケルトン (図 2) を生成した。

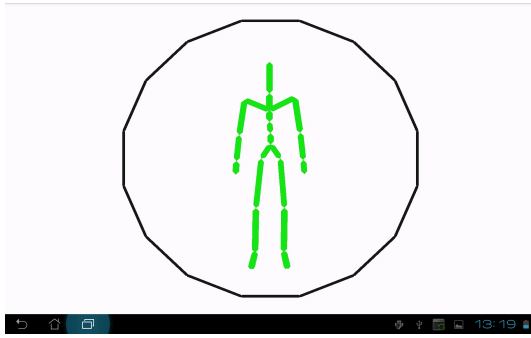


図 3 アプリケーションの画面

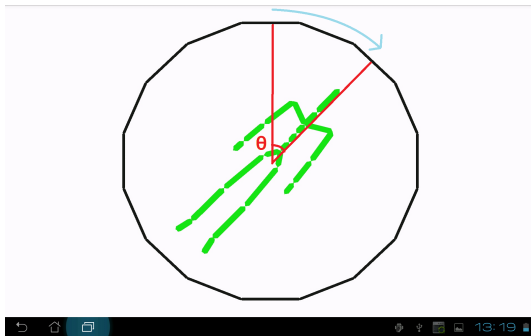


図 4 スケルトン全体の z 軸周りの回転

3.2 マルチタッチデバイスによるスケルトン操作インタフェース

マルチタッチデバイスを用いてスケルトン进行操作し、モーション編集を行うためのインタフェースを提案する。提案インタフェースを実行するためのアプリケーションは、図 3 のようにスケルトン操作に幅を持たせるための黒い円と操作対象のスケルトンを描画している。

アプリケーション内のワールド座標軸の取り方は水平方向を x 軸、鉛直方向を y 軸、奥行き方向を z 軸とする。

このアプリケーションは以下の機能を持つ。

- スケルトン位置の変更
- スケルトンのモーション編集

3.2.1 スケルトン位置の変更

スケルトン位置の変更では次の操作を実装している。

- スケルトン全体の z 軸（奥行き軸）周りの回転
- スケルトン全体の xy 平面上（操作者から見た平面）の任意軸周りの回転
- 拡大と縮小

3.2.1.1 スケルトン全体の z 軸周りの回転

スケルトン全体の z 軸周りの回転は、黒い円からはみ出さず 1本の指でドラッグすることによって行う。図 4 は水色の矢印がドラッグの軌跡、赤色で書かれた θ が回転量を表している。円の中心点からドラッグ開始点へのベクトルと、円の中心点からドラッグ終了点へのベクトルがなす角度分を回転させる。ドラッグ開始点を (x_1, y_1) 、ドラッグ終了点を (x_2, y_2) とする。このとき、 z 軸周りの回転量 θ_z は

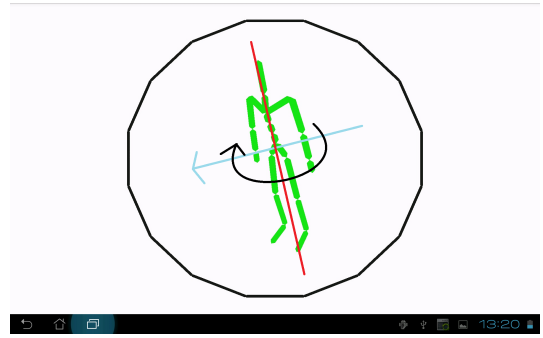


図 5 スケルトン全体の xy 平面上の任意軸周りの回転

$$\theta_z = \arctan \frac{y_1}{x_1} - \arctan \frac{y_2}{x_2} \quad (1)$$

となる。回転軸を z 軸として、原点を中心に、回転量 θ_z だけ回転する。

3.2.1.2 スケルトン全体の xy 平面上の任意軸周りの回転

スケルトン全体の xy 平面上の任意軸周りの回転は、黒い円の内部を 1本の指でドラッグすることによって行う。図 5 は水色の矢印がドラッグの軌跡、赤色の線が回転軸、黒色の矢印が回転方向を表している。ドラッグ開始点が始点でドラッグ終了点が終点のベクトルに対して、垂直な xy 平面上のベクトルを回転軸として回転する。ドラッグ開始点を (x_1, y_1) 、ドラッグ終了点を (x_2, y_2) とする。このとき、入力されたドラッグの軌跡と xy 平面上の x 軸の正の部分となす角度 ψ_{xy} は

$$\psi_{xy} = \arctan \frac{y_2 - y_1}{x_2 - x_1} \quad (2)$$

となる。よって xy 平面上のベクトル $(x, y) = (-\sin \psi_{xy}, \cos \psi_{xy})$ を中心に回転する。 θ_{xy} はドラッグの距離に比例する回転量である。 θ_{xy} は係数 k を用いて

$$\theta_{xy} = k\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

で与えられる。係数 k は予備実験で得た値を用いる。回転軸を $(-\sin \psi_{xy}, \cos \psi_{xy})$ とし、原点を中心に、回転量 θ_{xy} だけ回転する。

3.2.1.3 拡大と縮小

スケルトンの拡大は 3本の指によるピンチアウト（図 6）、縮小は 3本の指によるピンチイン（図 7）によって行う。

3.2.2 スケルトンのモーション操作

スケルトンのモーション操作は、シングルタップによる操作部位の選択とドラッグによる部位の操作の 2段階にわけて行う。

3.2.2.1 操作部位の選択

モーション操作で動かす部分は、シングルタップによって同時に 2カ所まで選択することができる。選択された部位の始点の関節には小さな青い円が表示され、選択されたことを示す。2本の指で同時に部位のモーション編集を行った場合、それぞれの部位の関節（小さい青い円）と各ドラッグの始点について xy 平面上の 2次元ユークリッド

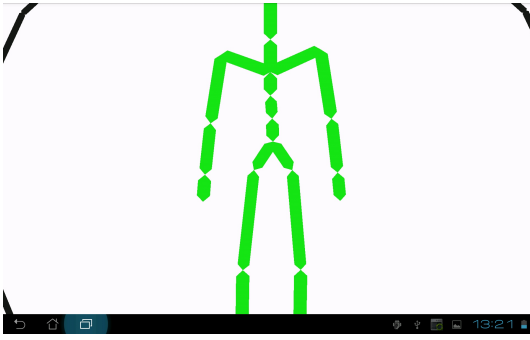


図 6 スケルトンの拡大図

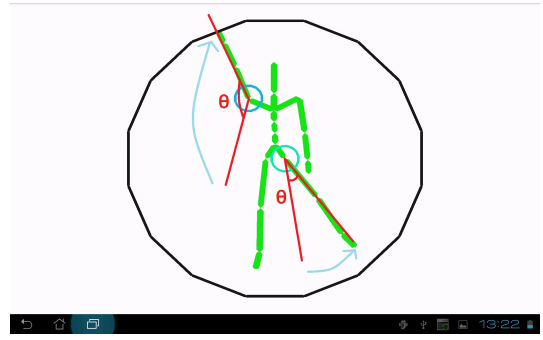


図 8 選択された部位の z 軸周りの回転

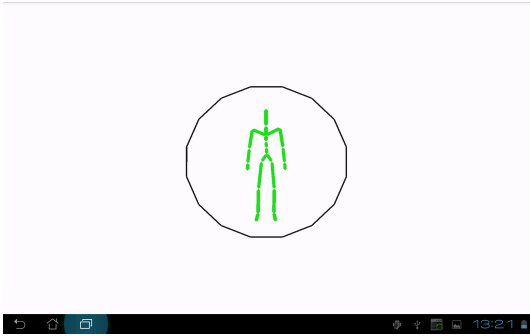


図 7 スケルトンの縮小図

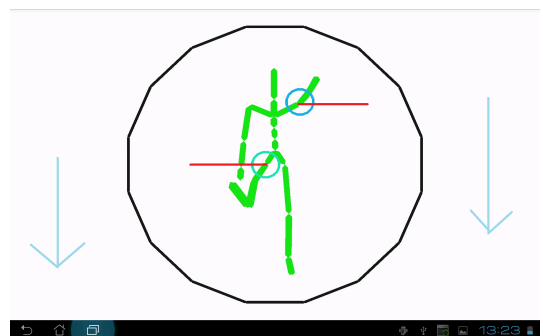


図 9 選択された部位の xy 平面上の任意軸周りの回転

距離の近い方が選択され、回転量に反映される。編集する部位を 2 カ所選んで、モーション編集は指一本で行う場合は同様に移動量が反映され 1 カ所だけのモーション編集となる。

3.2.2.2 フォワードキネマティクスによるモーション操作

選択された部位におけるモーション操作はドラッグにより同時に 2 カ所まで操作できる。部位の動作はフォワードキネマティクスにより行われるため親の部位の動きに従って子の部位も動く。互いに独立した四肢（右上腕と左太腿など）ならば同時に操作できるが、親子関係にあたる部位（左前腕と左手など）は今回の提案手法では同時操作できない。

選択された部位におけるモーション操作は z 軸周りの回転と xy 平面上の任意軸周りの回転の 2 種類である。

3.2.2.3 選択された部位における z 軸周りの回転

選択された部位における z 軸周りの回転は、部位を選択した後に黒い円の内部をドラッグすることによって行われる。図 8 は水色の矢印がドラッグの軌跡、 θ が回転量を表している。

回転軸は z 軸に平行で、選択した部位の親側の関節（小さい青い円）を通る線分である。その関節の xy 平面上の位置を (x_o, y_o) 、ドラッグ開始点を (x_1, y_1) 、ドラッグ終了点を (x_2, y_2) とすると回転量 θ_z は

$$\theta_z = \arctan \frac{y_1 - y_o}{x_1 - x_o} - \arctan \frac{y_2 - y_o}{x_2 - x_o} \quad (4)$$

と表される。z 軸を回転軸として、選択部位の親側の関節 (x_o, y_o) を中心に、回転量 θ_z 分だけ回転する。

3.2.2.4 選択された部位における xy 平面上の任意軸周りの回転

選択された部位における xy 平面上の任意軸周りの回転は、部位を選択した後に黒い円の外部をドラッグすることによって行われる。図 9 は水色の矢印がドラッグの軌跡、赤色で書かれた線が部位の回転軸を表している。ドラッグ開始点を (x_1, y_1) 、ドラッグ終了点を (x_2, y_2) とすると、入力されたドラッグの軌跡と xy 平面上の x 軸の正の部分がなす角度 ψ_{xy} は

$$\psi_{xy} = \arctan \frac{y_2 - y_1}{x_2 - x_1} \quad (5)$$

となる。 $(x, y) = (\cos \psi_{xy}, \sin \psi_{xy})$ を図 10, 11 に示すように

- 範囲 1 に収まれば $\psi_{xy} = 0$
- 範囲 2 に収まれば $\psi_{xy} = \frac{\pi}{4}$
- 範囲 3 に収まれば $\psi_{xy} = \frac{\pi}{2}$
- 範囲 4 に収まれば $\psi_{xy} = \frac{3\pi}{4}$
- 範囲 5 に収まれば $\psi_{xy} = \pi$
- 範囲 6 に収まれば $\psi_{xy} = \frac{-3\pi}{4}$
- 範囲 7 に収まれば $\psi_{xy} = \frac{-\pi}{2}$
- 範囲 8 に収まれば $\psi_{xy} = \frac{-\pi}{4}$

と単純化する。

単純化された ψ_{xy} を用いて、選択部位の親側の関節 (x_o, y_o) を中心に、ベクトル $(x, y) = (-\sin \psi_{xy}, \cos \psi_{xy})$ を回転軸として回転する。 θ_{xy} はドラッグの距離に比例する回転量である。回転量 θ_{xy} は係数 k を用いて

$$\theta_{xy} = k\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (6)$$

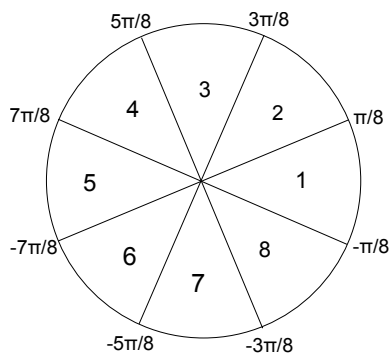


図 10 入力されたドラッグの分類

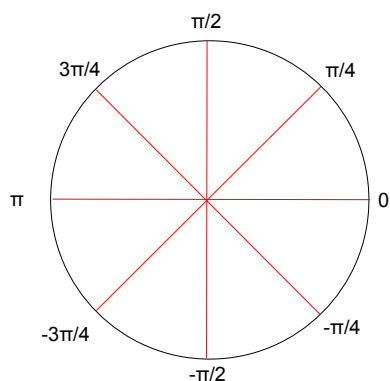


図 11 回転軸の決定

で与えられる．係数 k は予備実験で得た値を用いる．

上記のように入力に対して回転方向を 8 通りに制限した理由は，マルチタッチインタフェースによる入力が繊細すぎることによって，ユーザの思惑とは異なる動きをしてしまうことを防ぐためである．

3.3 アニメーション生成

スケルトンの操作から得られた情報によりアニメーションを生成する．現在のスケルトン操作インタフェースでは同時に 3 カ所以上の部位のモーション操作ができない．3 カ所以上部位を同時に動かすアニメーション生成のためには，複数回モーション編集を行った後に各モーションを統合する．図 12 のようにモーション編集を行うことにより，図 13 のように 3D モデルに反映される．

4. 実験

3 節で説明した提案手法のうちスケルトン操作インタフェースを実装したアプリケーションを用いて，マルチタッチによるスケルトンの操作しやすさを調査するパイロットスタディを行った．

4.1 実験詳細

実験の被験者の人数は 21 歳から 24 歳までの男性 6 名，女性 1 名の計 7 名である．全員が日常的にコンピュータを使用している．また 6 名がスマートフォンなどのマルチ

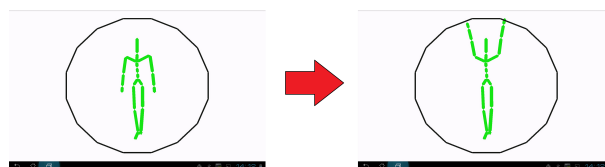


図 12 スケルトンのモーション

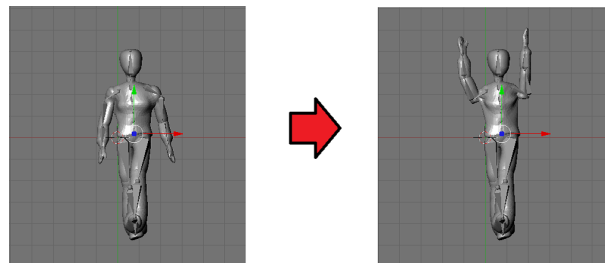


図 13 3D モデルのモーション

タッチデバイスを所持しており，1 名が所持していない．また左利きが 1 名，右利きが 6 名である．アニメーション制作の経験がある被験者はいなかった．

実験は，次の 3 段階で行う．

- (1) 被験者にスケルトンの操作方法を説明し，アプリケーションに触れてもらい操作に慣れてもらう (5 分)
- (2) 被験者はアプリケーションを用いて「歩く」モーションを両足の部分をマルチタッチにより同時に編集する (1 分)
- (3) 被験者は操作により得られたモーション情報より生成されたアニメーションを見て操作感と合わせた評価を述べる

4.2 実験結果

被験者の 7 名中 5 名はマルチタッチによる 2 カ所のモーション編集はわかりやすいとの評価だった．これは本研究の目的に合致するものである．また，被験者の 7 名中 4 名がスケルトンのモーション編集自体が楽しいと評価した．

被験者の 7 名中 5 名がスケルトンの部位が手前にあるのか奥にあるのか判別しにくいと評価した．そのためスケルトンの奥行き方向での表示方法を改善する必要がある．具体的には影やフォグをつけることにより奥行き方向での位置を明らかにするなどの方法がある．また，被験者の 7 名中 6 名が xy 平面上での任意軸周りの回転は方向の指定はしにくいと評価した．入力方向の単純化を行ったがそれでも奥行き方向に関する動きは思い通りにいかなかったと思われる．事前に回転方向を指定するオプションを追加してほしいという意見もあった．煩雑な操作を追加することは避けつつインタフェースを改善する必要がある．

5. 結論

本研究では，マルチタッチデバイスによるスケルトンの直感的な操作方法とそれにより得られた簡潔なアニメー

ション生成手法を提案し実装した。被験者にマルチタッチを用いたスケルトンの複数部位の同時操作によるモーション編集を行ってもらい、操作感の評価を調査した。

今後の課題としてはインタフェースの改善と機能の拡張がある。パイロットスタディで得た評価を反映させたスケルトン操作インタフェースの改善を行い、より簡単なアニメーション生成手法の提案を目指す。またドラッグする指の本数で部位の動作方法が変わる新たな機能も提案したい。

参考文献

- [1] Blender, <http://blender.jp/>.
- [2] Li, Q. L., Geng, W. D., Yu, T., Shen, X. J., Lau, N. & Yu, G. (2006). MotionMaster: Authoring and Choreographing Kung-Fu Motions by Sketch Drawings. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 233-241.
- [3] Lee, J. Chai, J., Reitsma, P.S.A., Hodgins, J.K. & Pollard, N.S. (2002). Interactive Control of Avatars Animated with Human Motion Data. *ACM Trans. Graph*, 21(3), 491-500.
- [4] Yin, K. & Pai, D.K. (2003). Footsee: An Interactive Animation System. In *Proceedings of SCA'03*, 329-338.
- [5] Feng, T.C., Gunawardane, P., Davis, J. & Jiang B. (2008). Motion Capture Data Retrieval Using an Artist's Doll. In *Proceedings of ICPR'08*, 1-4.
- [6] Narukawa, H., Pantuwong, P. & Sugimoto, M. (2012). A Puppet Interface for the Development of an Intuitive Computer Animation System, In *Proceedings of ICPR 2012*.
- [7] Barakonyi, I. & Schmalstieg, D. (2005). Augmented Reality Agents in the Development Pipeline of Computer Entertainment, In *Proceedings of ICEC'05*, 345-356.
- [8] Numaguchi, N., Nakazawa, A., Shiratori, T. & Hodgins, J. K. (2011). A Puppet Interface for Retrieval of Motion Capture data. In *Proceedings of SCA'11*, 157-166.
- [9] Oshita, M. (2012). Multi-Touch Interface for Character Motion Control Using Example-Based Posture Synthesis, *20th International Conference on Computer Graphics, Visualization and Computer Vision 2012 (WSCG 2012)*.
- [10] Pantuwong, N. & Sugimoto, M. (2012). Skeleton Growing: an Algorithm to Extract a Curve Skeleton from a Pseudonormal Vector Field. *The Visual Computer*, 29(3), 203-216.
- [11] Pantuwong, N. & Sugimoto, M. (2012). A Novel Template-Based Automatic Rigging Algorithm for Articulated-Character Animation. *Computer Animation and Virtual Worlds*, 23(2), 125-141.
- [12] Cornea, N. D., Silver, D., Yuan, X. & Balasubramanian, R. (2005). Computing Hierarchical Curve-Skeletons of 3D Objects. *The Visual Computer*, 21(11), 945-955.