



仮想人間の動作制御技術

～ゲーム開発に応用できそうな研究の紹介～

尾下 真樹

九州工業大学 情報工学部 准教授

自己紹介



- 尾下 真樹（おした まさき）

- 九州工業大学 情報工学部
システム創成情報工学科 准教授

- 熊本県出身、1975年生まれ

- 研究内容

- コンピュータアニメーション技術の研究

- 略歴

- 2003年3月 九州大学 博士課程 修了 博士(情報科学)

- 2003年4月 九州工業大学 情報工学部 講師

- 2004年6月 九州工業大学 情報工学部 助教授

- 2007年4月 九州工業大学 情報工学部 准教授





介

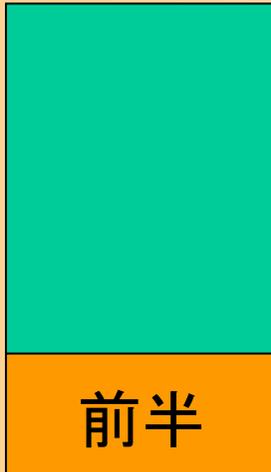
国立大学法人 九州工業大学 情報工学部

九州大学



構成

14:50~
16:10



- 前半(20分)

- イン트로ダクション
- 研究紹介

- 衣服・髪の実時間シミュレーション
- 仮想人間の操作インターフェース

16:40~
18:00



- 後半(40分)

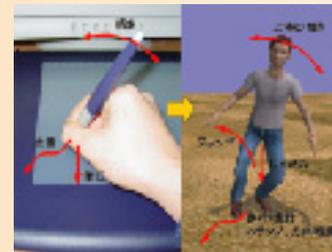
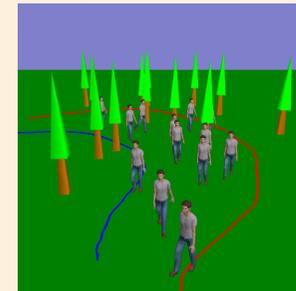
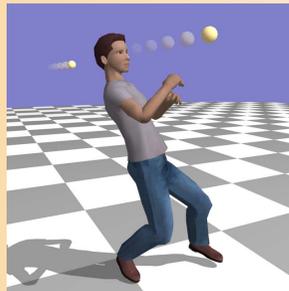
- 研究紹介・技術解説

- 仮想人間の動作制御技術
- 大学とゲーム業界の連携について



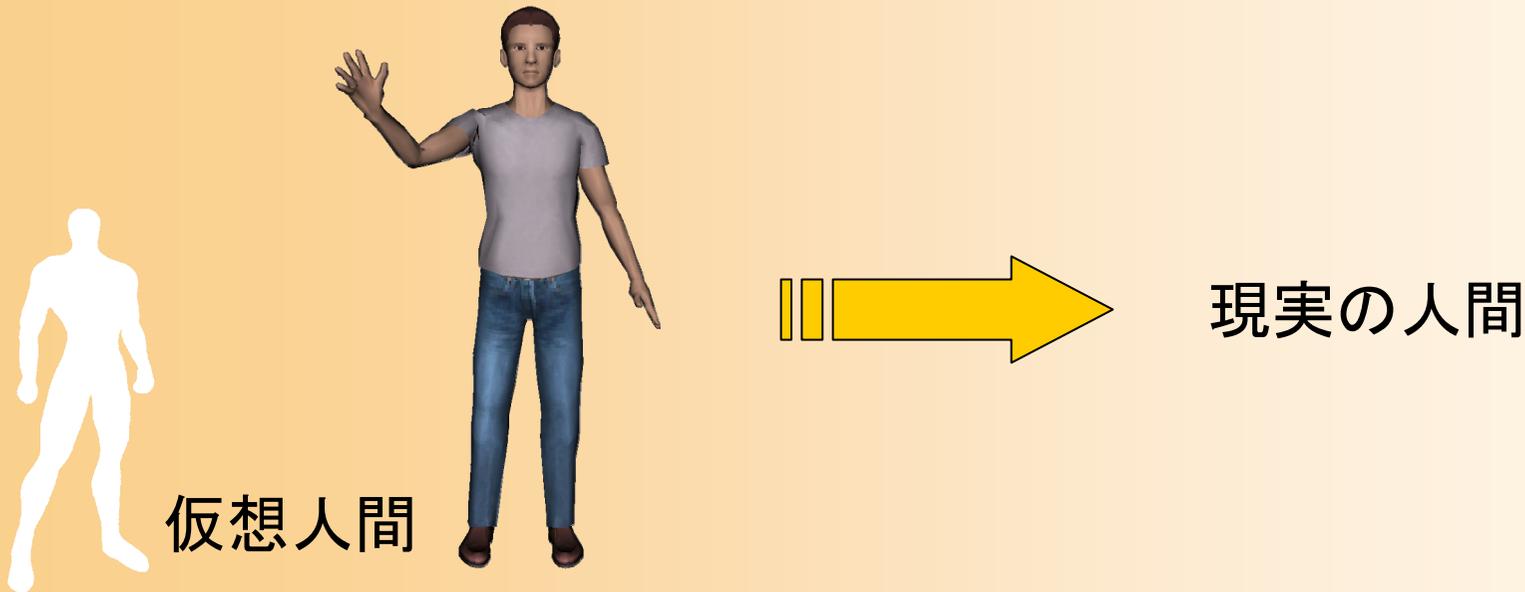
研究分野

- コンピュータアニメーション技術を主に研究
 - 特にリアルタイムにアニメーションを生成するための技術に注目
 - コンピュータゲームなどへの応用を目標



本分野の研究の最終目標

- 現実の人間と同様に動く **仮想人間** を実現
 - 人間と同様に、自律的に動作できる
 - 人間と同様に、動きを指示できる
 - 人間と同様の、自然な見た目（衣服・髪・皮膚）



本分野の主な課題

動作制御
(自律的な動作)

自然な外観の表現
(衣服・髪・皮膚)

操作インターフェース
(動作生成・変形・編集)

本分野の主な課題

- 仮想人間の自律動作制御

- 状況に応じた動作を動的に生成する必要がある
- 現在は、一定の動作の繰り返しのみ



- 操作インターフェース

- 利用者がさまざまな動作を直感的に操作(生成)したい
- 現在主流の入力装置では、限られた操作しかできない



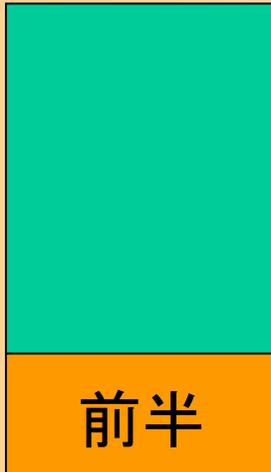
- 仮想人間の自然な外観の表現

- 衣服・髪・皮膚などのアニメーション
- 速度的な問題により、物理シミュレーションは困難



構成

14:50~
16:10



- 前半(20分)

- イン트로ダクション

- 研究紹介

- 衣服・髪の実時間シミュレーション
- 仮想人間の操作インターフェース

16:40~
18:00



- 後半(40分)

- 研究紹介・技術解説

- 仮想人間の動作制御技術

- 大学とゲーム業界の連携について





衣服・髪の実時間シミュレーション

物理シミュレーションの利用

- 近年、ゲームでの物理エンジンの利用も一般的になりつつある(ODE, PhysX 等)
- 物理シミュレーションの利用方法
 - ゲーム物理(Game Physics)
 - 物理シミュレーションがゲームの進行に影響を与える
 - 物理パズル、衝撃に応じた動作の変化、など
 - 効果物理(Effect Physics)
 - ゲームの進行には影響せず、単に自然な映像を実現
 - 物体の破壊、炎・水面、衣服・髪、倒れる人間、など
 - こちらのほうが、比較的容易に導入できる



仮想人間の外観のシミュレーション

- 自然な見た目の表現には「外観の動き」も重要
 - 衣服
 - 髪
 - 皮膚の変形
- 単純なワンスキンモデルでは、これらの表現はできない
 - 衣服や髪の形状は固定
 - 簡易的な物理シミュレーションでは、まだ不自然



衣服・髪の実時間シミュレーション

- 衣服の実時間シミュレーション
- 髪の実時間シミュレーション
- 皮膚の実時間シミュレーション





衣服の実時間シミュレーションと 仮想試着システム

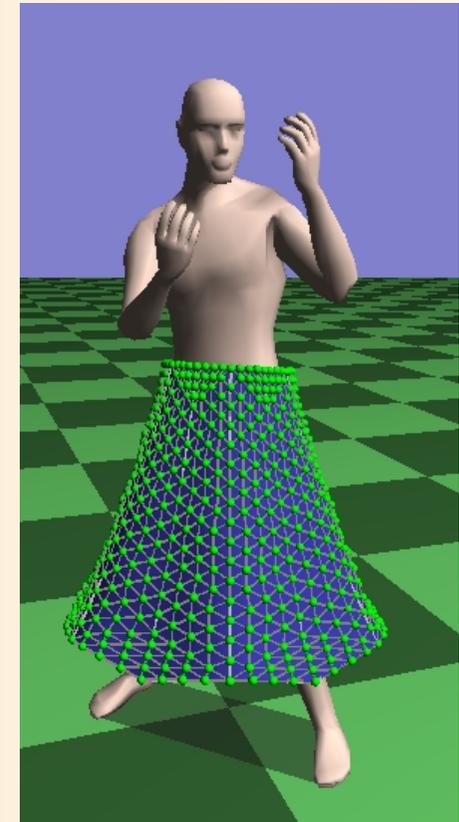
発表論文

Computer Animation 2001,
Visual Computing Symposium 2003

特願 2003-173742

実時間衣服シミュレーション

- 衣服の動力学シミュレーション
 - 映画などで実用的に使われている
 - 粒子モデルによる計算手法
 - 衣服を粒子モデルにより表現
 - 物理法則に従って粒子の運動を計算
- 実時間処理の問題点
 - 多数の粒子を必要とするため計算時間が増大
 - 少数の粒子では、しわなどの細部の特徴を表現できない



開発手法

1. 粗いメッシュ(少数の粒子)による 動力学シミュレーション

- 衣服の大まかな振る舞いを実現
 - 人体動作に応じた全体的な動きを計算

2. 幾何的手法により 衣服の表面形状を生成

- 衣服の細かい振る舞いを実現
 - 粗いメッシュからなめらかな表面を生成
 - 体に合わせて表面の形状を変形
 - 表面に働く力を分析してしわを変形

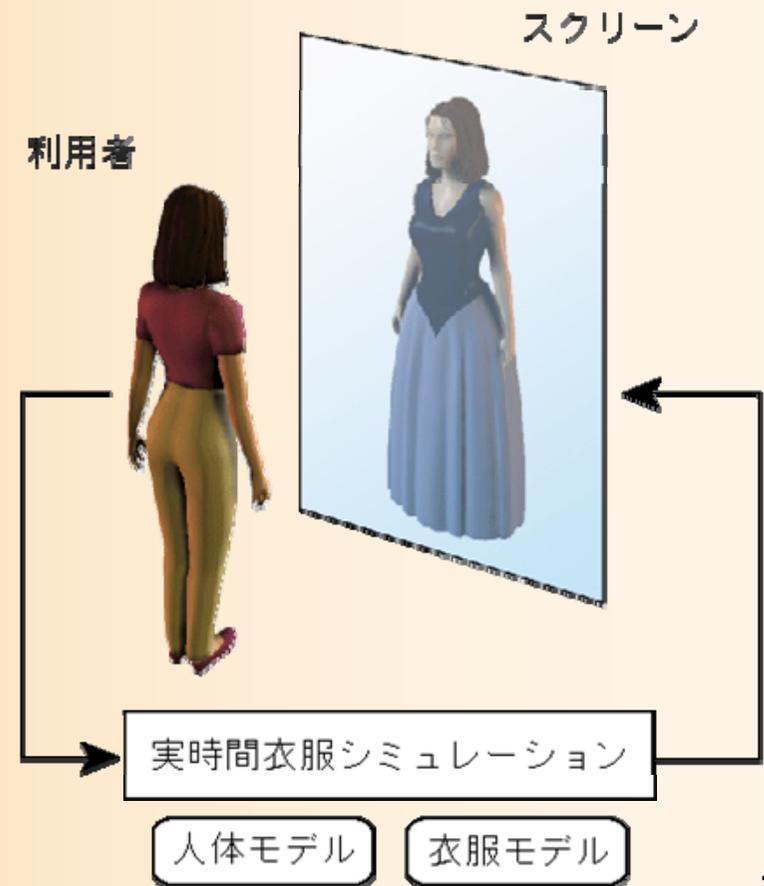


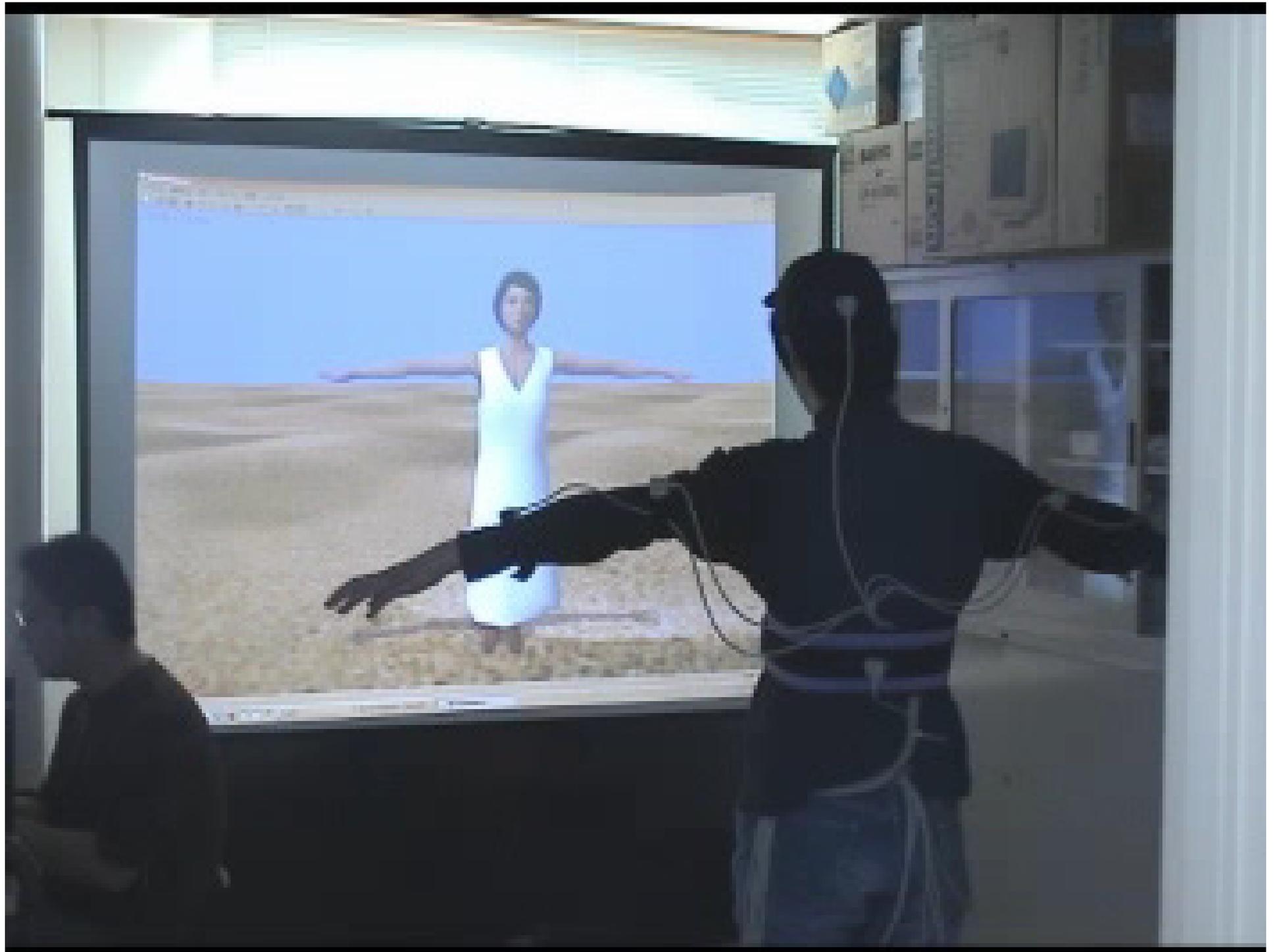
プログラムのデモ



仮想試着システム

- 利用者が仮想空間内で任意の衣服を試着できるようなシステム
 - モーションキャプチャ機器との組み合わせ
 - 衣服シミュレーションの究極の応用例
 - 将来はあらゆる衣服を自由にダウンロードして試着できるようになる？





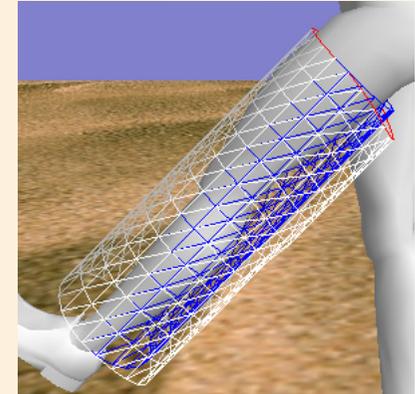
今後の課題

- より自然な形状の生成
 - 衣服同士の衝突・めり込みの処理
 - しわの変形アルゴリズムの改良
 - 各種素材によるしわの違いの表現
- 高速化
- 実用化



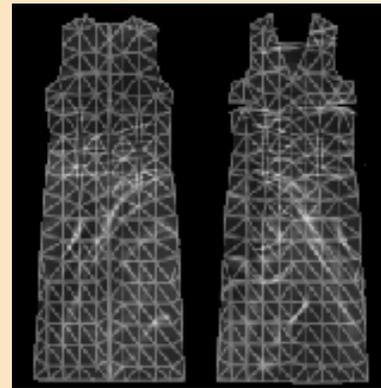
開発中の手法

1. 人体の動きに応じて、粗い衣服モデルの大まかな動きを計算
 - 正確なシミュレーションは用いない

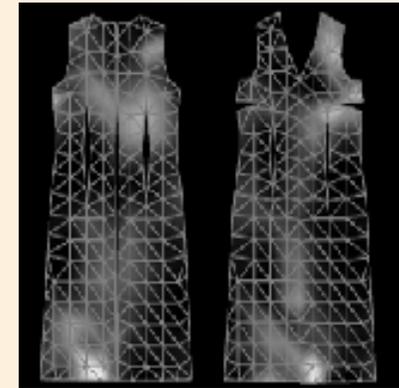


2. 衣服の表面上の圧縮力に応じて、テクスチャ空間で、詳細なしわの変化を擬似的に生成

(a) しわを表すハイトフィールド



(b) 圧縮力・人体からの接触力



- 全体の計算をGPU上で高速に実行



髪の実時間シミュレーション

発表論文

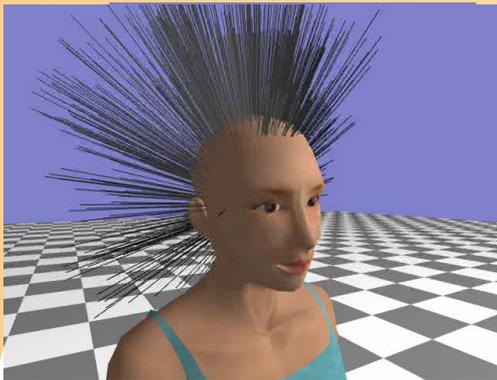
Computer Animation and Virtual Worlds 2007
(Computer Animation and Social Agents 2007)

特願 2007-153550

アイデア

- Setp 1. 粗い髪モデルの物理シミュレーション
- Step 2. 詳細な髪の毛を動的に生成
 - 髪の毛の周囲に、パラメタに応じて、髪の毛の束を生成
 - 速度に応じて髪の束や個々の髪の形状を変形
- グラフィックハードウェア上で高速に処理を実行

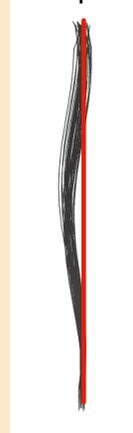
(a) Coarse hair model



(b) Fine hair model



(a) Generated wisp



(b) Static wisp in motion



(c) Dynamic wisp in motion



髪一束の動的変形モデル

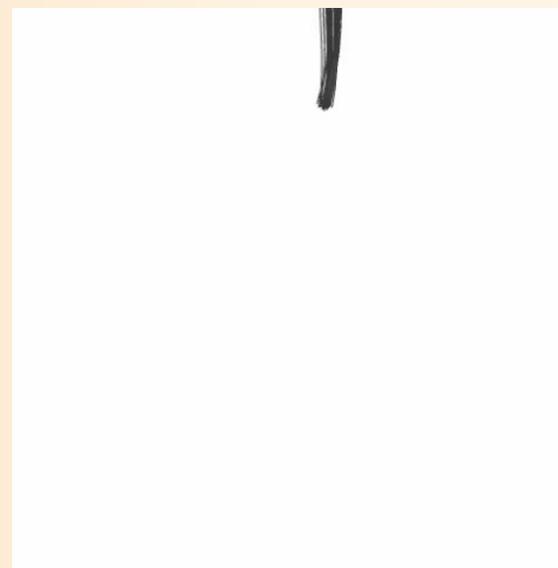
- 粗い髪の毛の運動速度に応じて、生成する髪の
 - 束の形状を変化（後方に広がるよう変形）
 - 髪の毛の形状を変化（水平方向に圧縮されるよう変形）



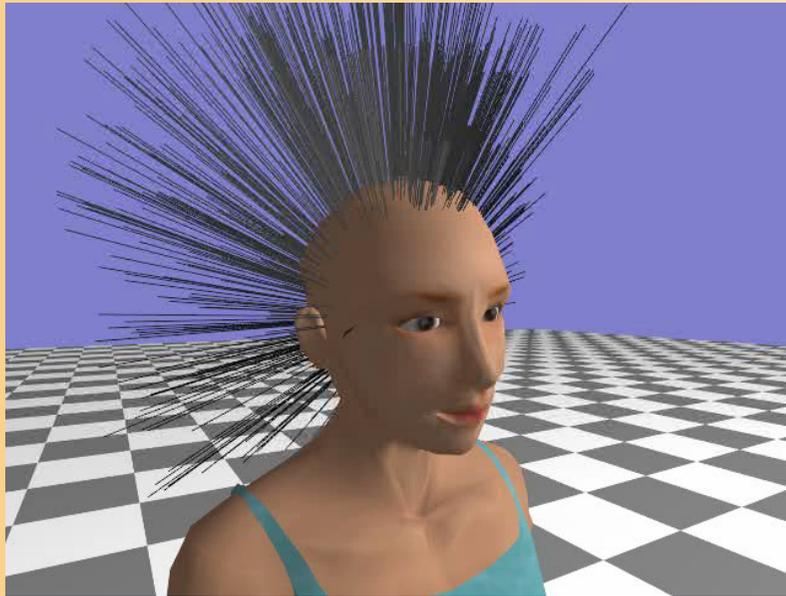
粗い髪の毛



静的な束の生成



動的な束の生成



粗い髪のシミュレーション(1000本)



動的な髪の束の生成(20,000本)



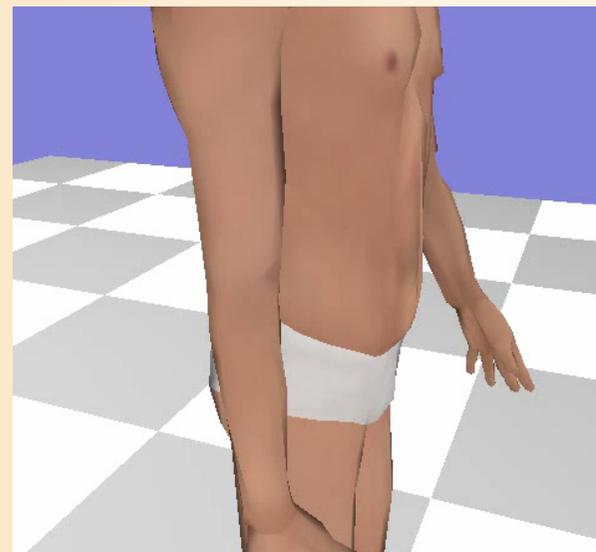
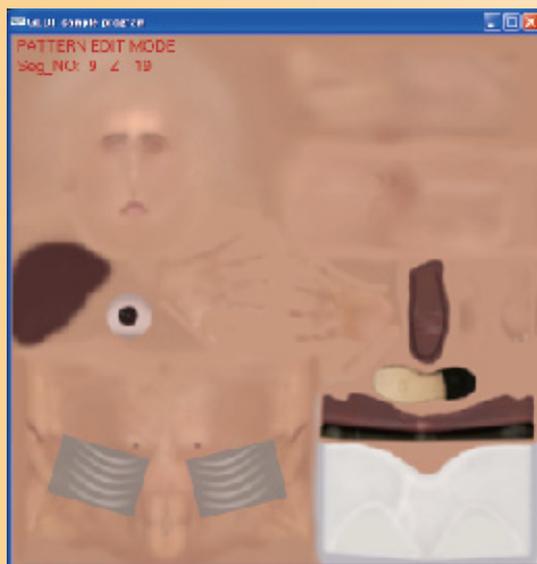
詳細な髪のシミュレーション(20,000本)₂₄



皮膚の変形モデル

皮膚の変形モデル

- 動的ハイトマップによる皮膚変形の表現
 - テクスチャ空間で、変形のパターン・位置を設定
 - 動作に応じて、ハイトパターンを変化
 - バンプマップを適用して描画



まとめ

- 仮想人間の外観のシミュレーション
 - 衣服
 - 髪
 - 皮膚の変形





仮想人間の操作インターフェース

キャラクターの操作の問題

- キャラクターの動作をきめ細かく制御できるようなインターフェースが欲しい
- 従来の技術の問題点
 - ゲームパッドなどを使った操作
 - ごく単純な動きしか操作できない
 - マウス + キーボード
 - 操作が難しい
 - 加速度・位置センサ
 - モーションキャプチャ
 - 設備にお金やスペースが必要



仮想人間の操作インターフェース

- 動作認識のための自動学習
- ペンを使った仮想人間の操作
- ストロークを使った仮想人間の操作
- データグローブによる仮想人間の操作





動作認識のための自動学習

動作認識

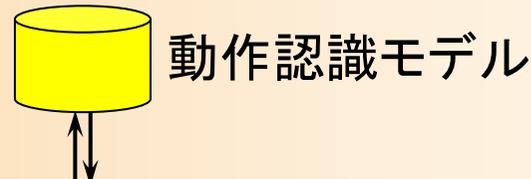
- 動作認識

- 何らかのセンサから入力した利用者の動きをもとに、何の動作を行っているかを判定する技術
 - 動作の速度や大きさの判定もできることが望ましい



入力デバイス

特徴量



動作認識

認識動作

例：Wii Remote、カメラ画像、
モーションキャプチャ

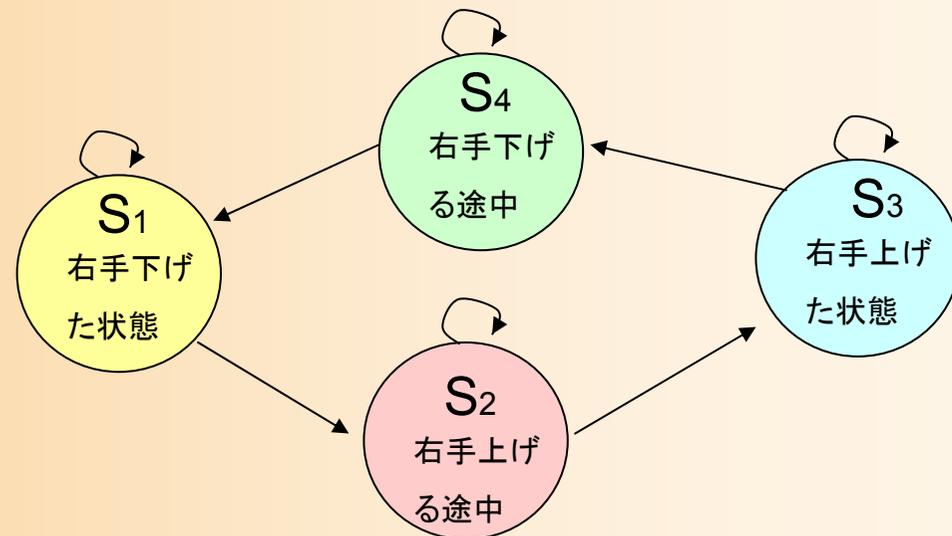
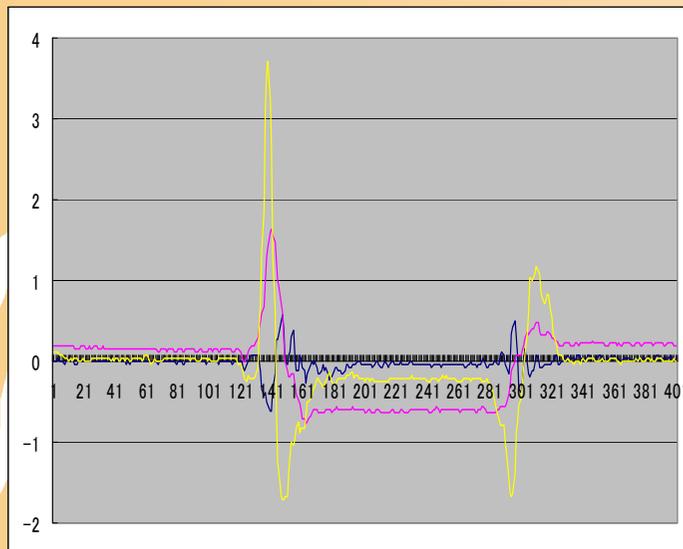
例：手を上げる、振る、
殴る、突く、投げる

- Wii Remote を入力として使用 (6自由度の加速度)

一般的な動作認識手法

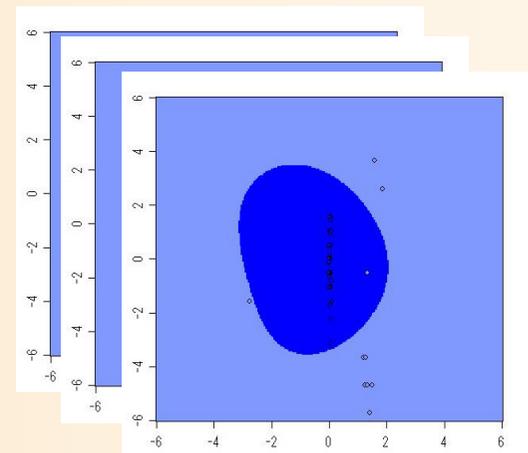
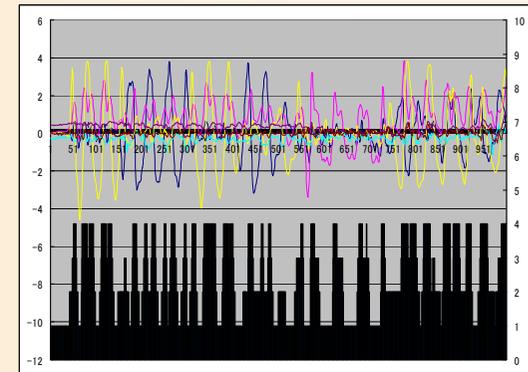
- 状態遷移を使った動作認識
 - 各動作をいくつかの状態に分ける
 - 状態間の遷移の特徴量の条件を適切に設定
 - 認識モデルの作成には非常に手間がかかる

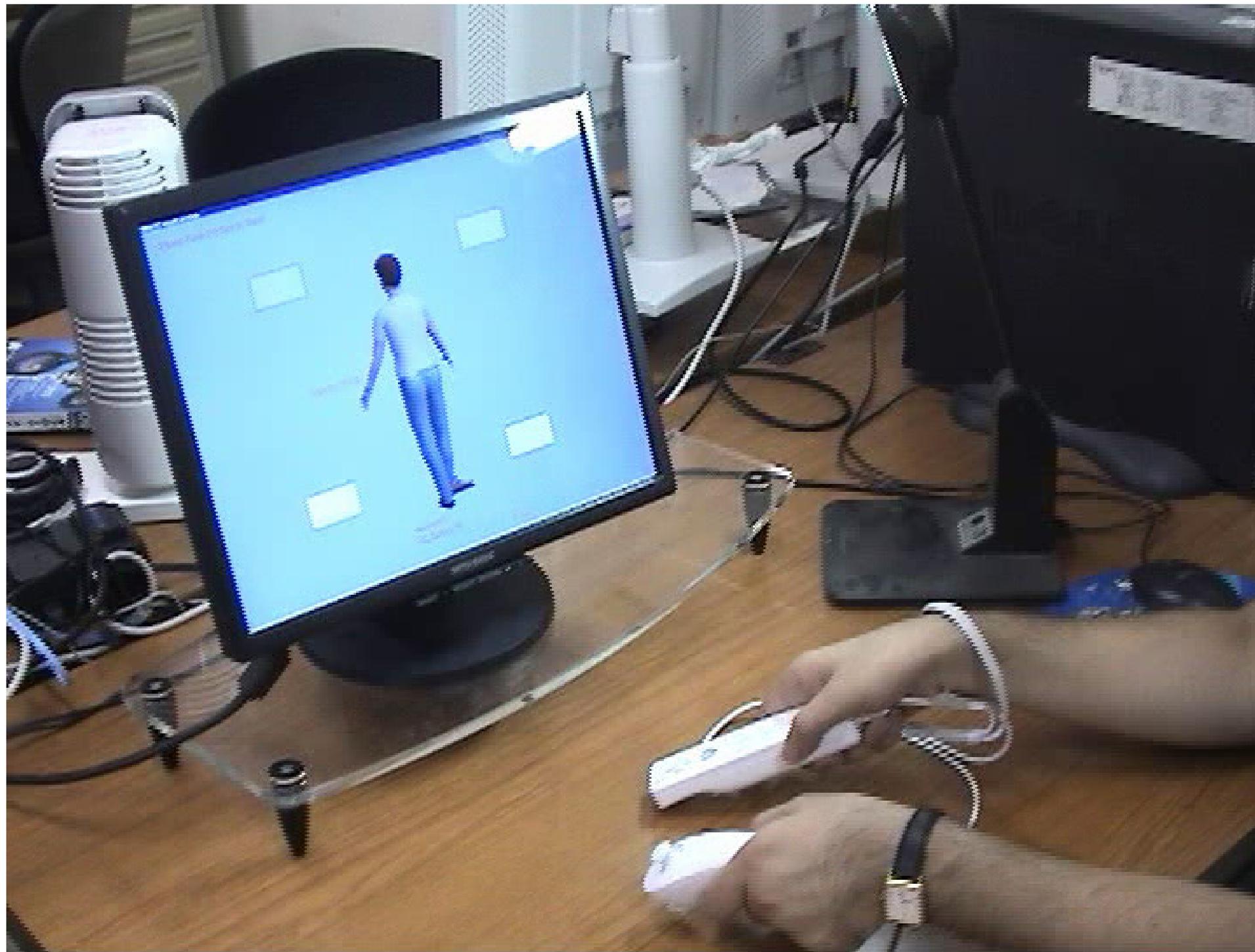
例：右手を上げ下げする動作



開発中の手法

- 入力された一定数のサンプル動作をもとに、認識モデルを自動的に構築
 - SOM (Self Organization Map) による状態の分類
 - 多数のベクトルを、値が似通ったもの同士でクラスタリングする技術
 - SVM (Support Vector Machine) による各ノードの遷移条件推定
 - 多数のサンプルにもとづいて、特徴量空間を分割する技術







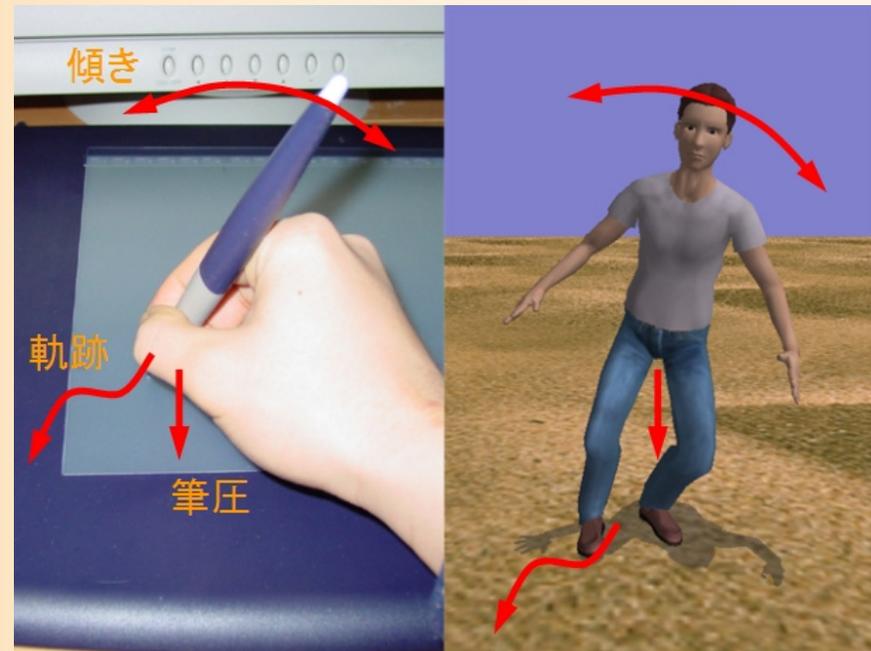
ペンによる 仮想人間の操作インターフェース

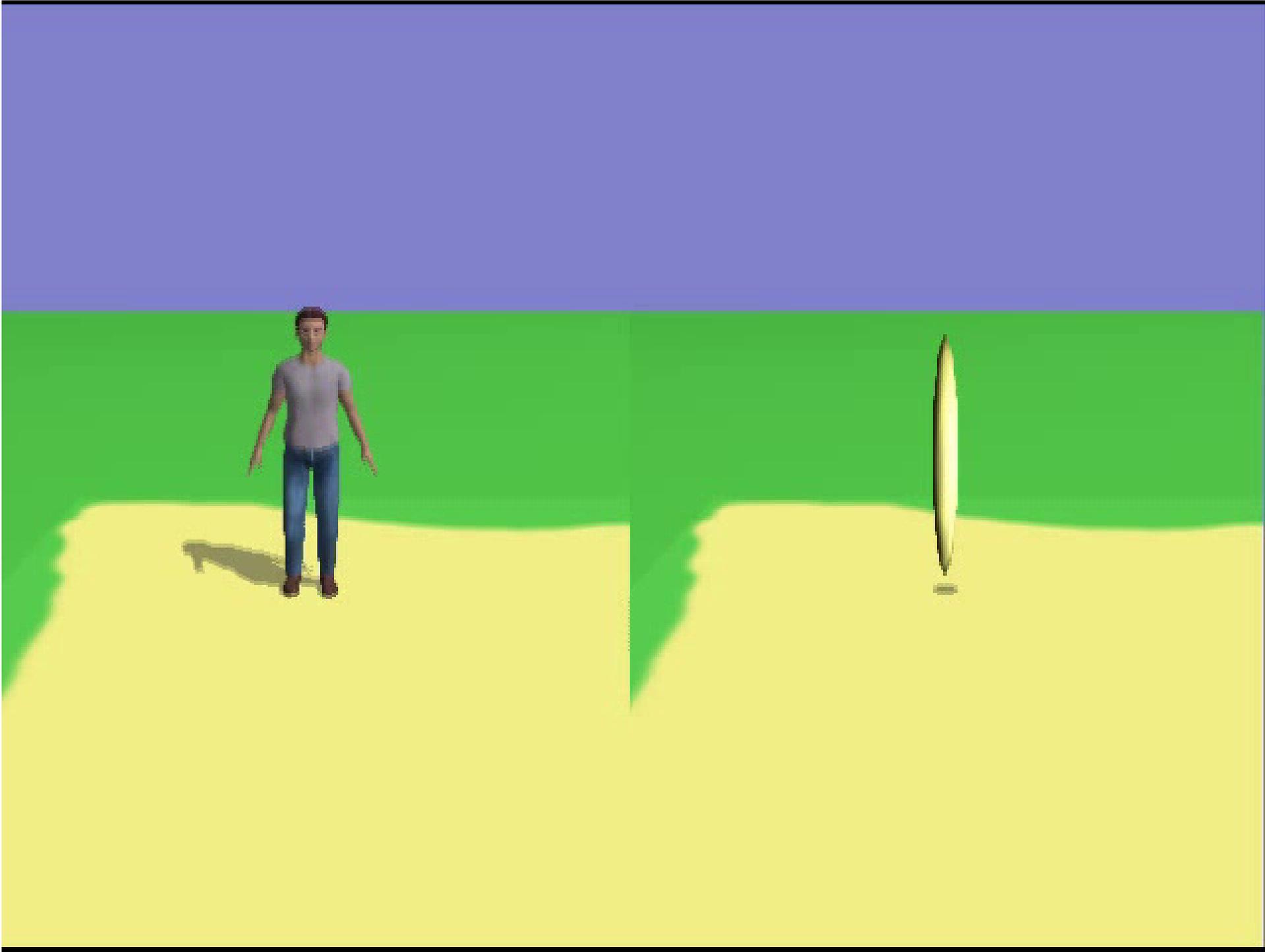
発表論文

Computers & Graphics 2006,
Eurographics Workshop on Sketch
Based Interfaces & Modeling 2004

ペンによるキャラクターの操作

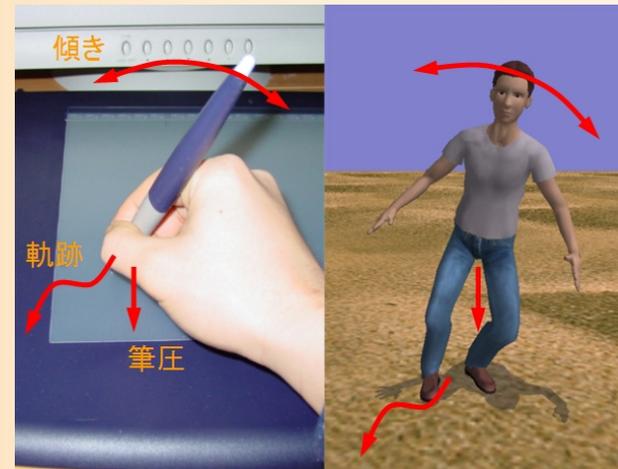
- ペンを使った直接操作インターフェース
 - ペンの軌跡・速度に応じてキャラクターが歩行・走行
 - ペンの傾きや筆圧などの情報も使用
 - 最近のペンデバイスは高機能になっており、多くの入力情報が利用できる





応用例

- コンピュータゲーム、仮想空間内の散歩、仮想空間でのコミュニケーション、等
 - さまざまな動きを表現可能
 - ゲームパッドに不慣れな人でも利用可能





ストロークによる 仮想人間の操作インターフェース

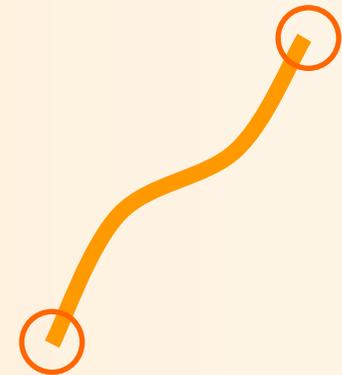
発表論文

Computer Animation and Virtual Worlds 2005
(Computer Animation and Social Agents 2005)

特願2006-162666

ストロークによるキャラクター操作

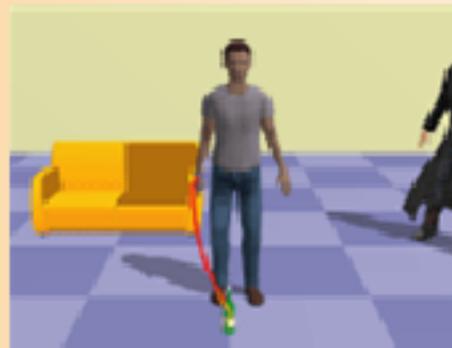
- ストロークによる動作指定
 - ストロークは**開始点**と**終了点**を持つ
 - 1回のストロークにより、動作の**起点**(道具格・源泉格)と**終点**(対象格)を同時に指定可能
- ストロークによる動作の変化も可能



(a) punch



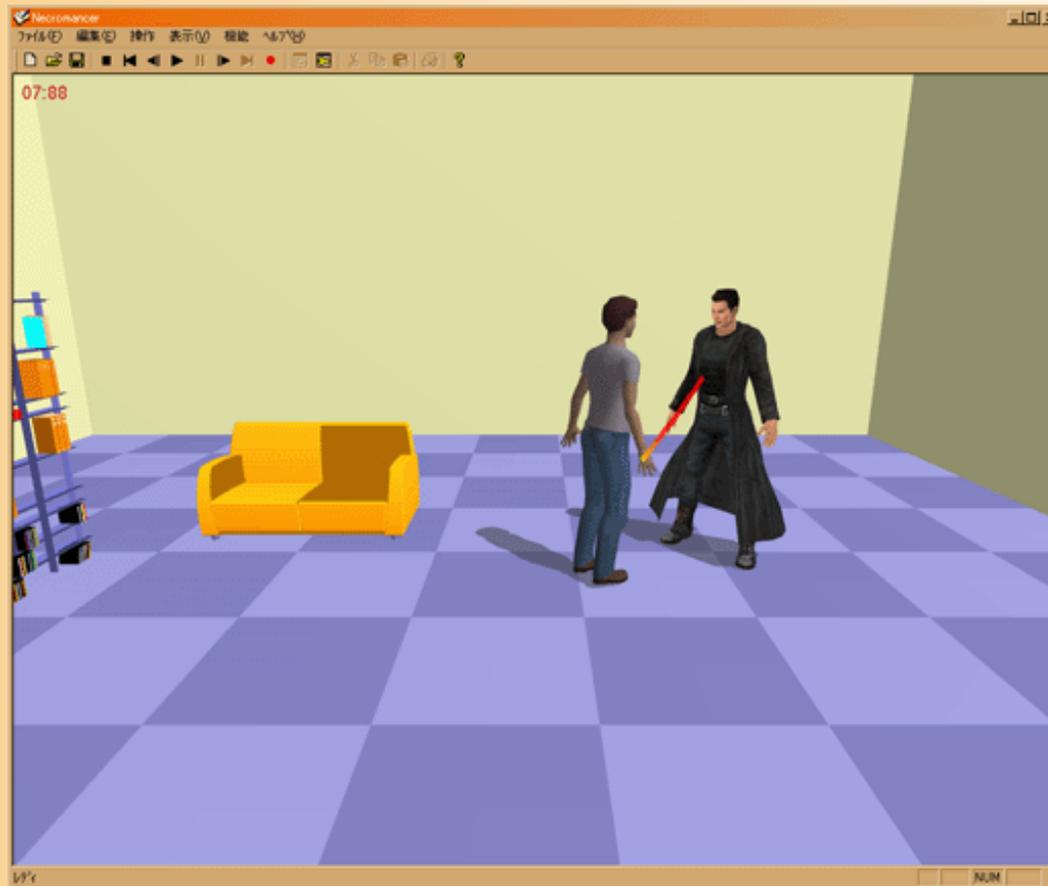
(b) take



(c) drink



プログラムのデモ



応用例

- コンピュータゲーム

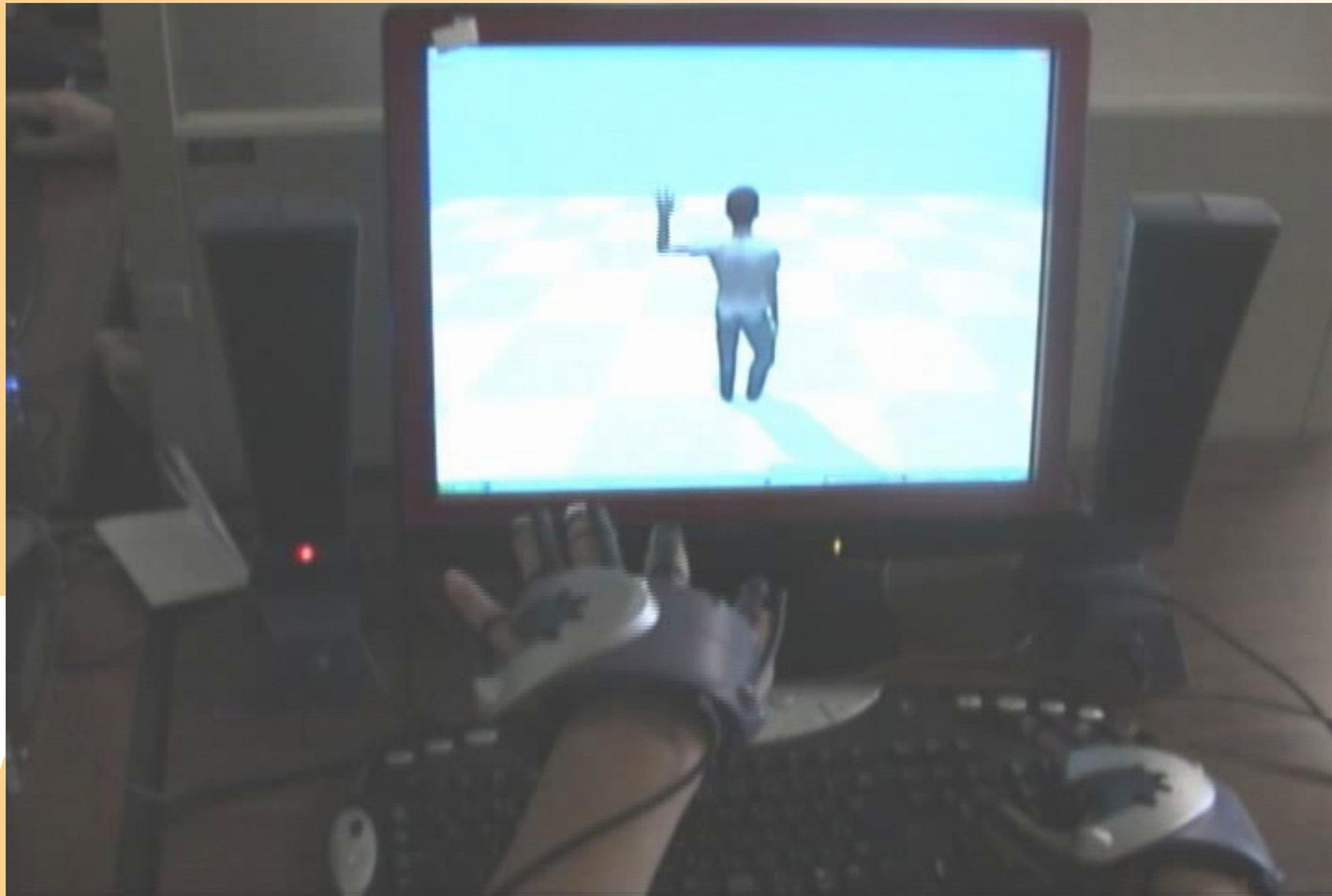
- Nitendo DS や Smart Phone など、ペン入力機能を持った携帯用ゲーム機
- 仮想空間内で、さまざまな動きを行ったり、さまざまな物を扱うようなゲーム
 - 例：アドベンチャーゲーム、アクションゲーム、等



© Nintendo, 2004



データグローブによる操作



まとめ

- 仮想人間の操作インターフェース
 - 動作認識のための自動学習
 - ペンを使った仮想人間の操作
 - ストロークを使った仮想人間の操作
 - データグローブによる仮想人間の操作





仮想人間の動作制御技術

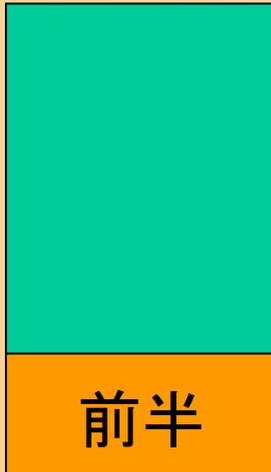
～ゲーム開発に応用できそうな研究の紹介～

尾下 真樹

九州工業大学 情報工学部 准教授

構成

14:50~
16:10



- 前半(20分)

- イン트로ダクション
- 研究紹介

- 仮想人間の操作インターフェース
- 衣服・髪の実時間シミュレーション

16:40~
18:00



- 後半(40分)

- 研究紹介・技術解説

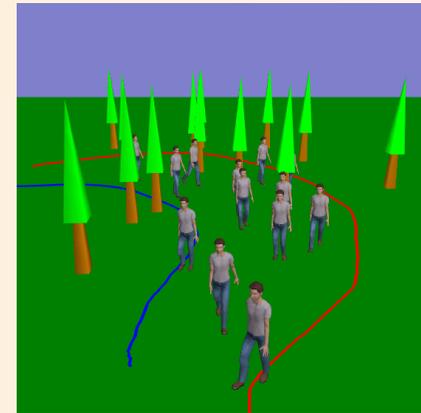
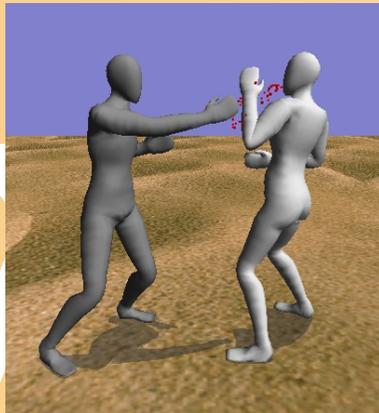
- 仮想人間の動作制御技術

- 大学とゲーム業界の連携について



後半の内容

- 仮想人間の動作制御技術
 - 動力学を考慮した動作制御
 - モーショングラフを利用した自律動作制御
 - 学習にもとづく群集の制御
 - ペン入力による群集の制御





仮想人間の動作制御技術

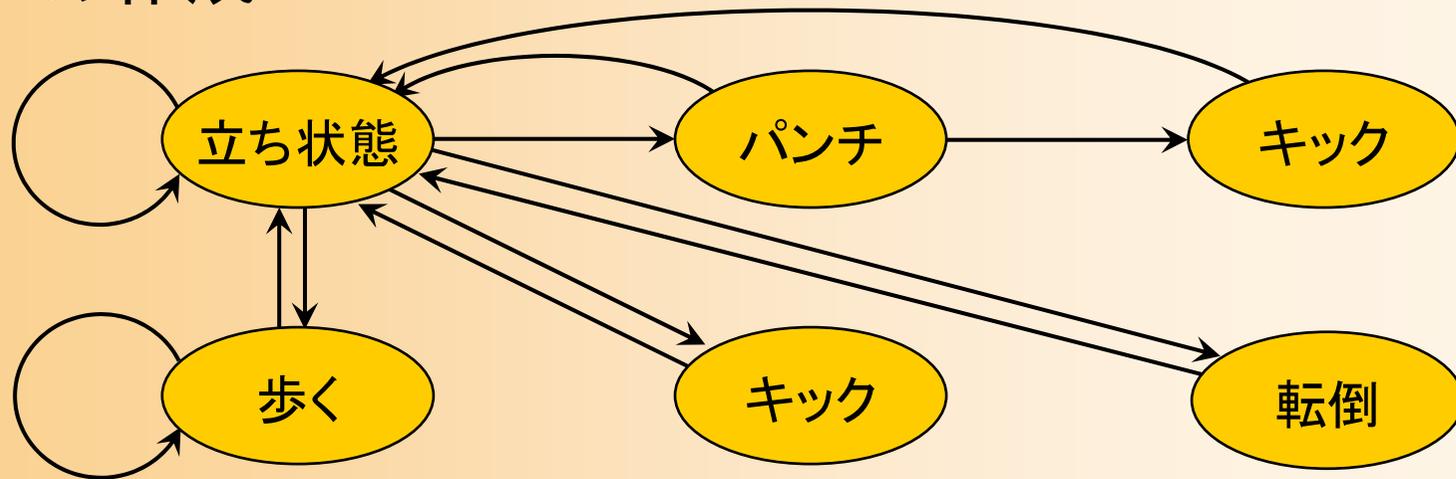
発表論文

Eurographics 2001, 他

特願 2002-189163

現在の一般的な動作制御手法

- 動作ツリーを使った動作生成
 - 短い単位動作と、動作間のつながりを、あらかじめ作成



- ユーザーの入力やルールに従って、次に再生する動作を決定
- 必要に応じて、動作ブレンディングや IK を適用



現在の技術の問題点

- 動作ツリーの作成には多くの手間がかかる
- 一定動作の繰り返ししかできない
 - ワンパターンな動きになる
 - 特に衝突や外力などの力学的な影響に応じた自然な動作が困難
- 自律動作の実現は困難
 - 敵キャラクターや群集などを自律的に動かすためには、適切な動作ルールが必要になる



仮想人間の動作制御技術

- 動力学を考慮した動作制御
- モーショングラフを利用した自律動作制御
- 学習にもとづく群集の制御
- ペン入力による群集の制御



動力学を考慮した動作制御

- 特に衝突や外力などの力学的な影響に応じた動作
 - さまざまな状態・衝撃に応じた動作を全て用意しておくことは非現実的
 - 何らかの方法で、動的に動作を生成することが必要になる



一般的な手法の問題点

- 動作データの再生による動作制御
 - 衝突に応じて、再生動作を切り替え
 - 同じ動きの繰り返ししかできない
 - 多数の動作データや、動作選択のルールが必要
- 物理シミュレーションによる動作制御
(ラグドール・シミュレーション)
 - 衝突時に、物理シミュレーションに切り替え
 - 受動的な動作のみで、能動的な動作は実現できない
 - 倒れずに回復するような動作の実現は困難

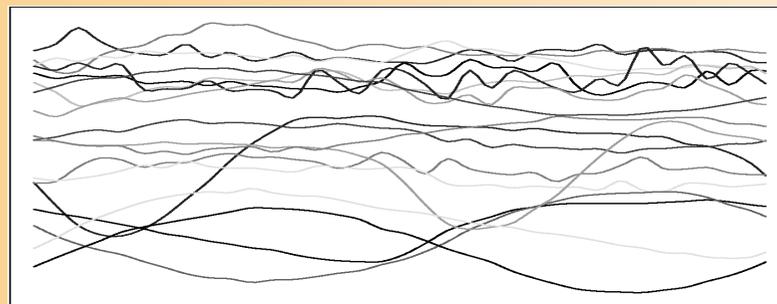
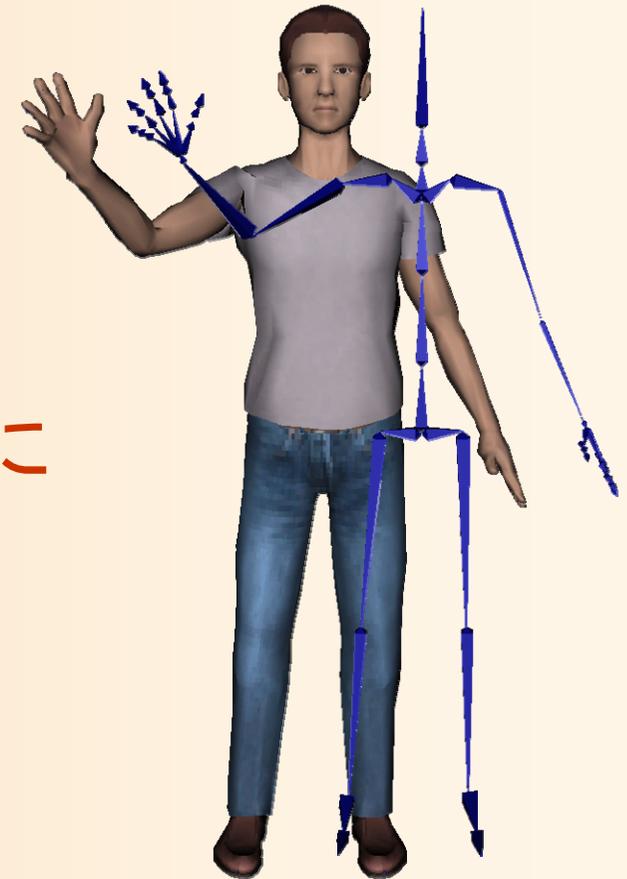


仮想人間の姿勢・動作の表現

- 多関節体の姿勢の表現
 - 腰の位置・向き(6自由度)
 - 各関節の回転角度(n自由度)

$$P_{root} \quad q_{root} \quad \theta_0 \sim \theta_{n-1}$$

- 関節角度(姿勢)の時間変化により多関節体の動きを表現



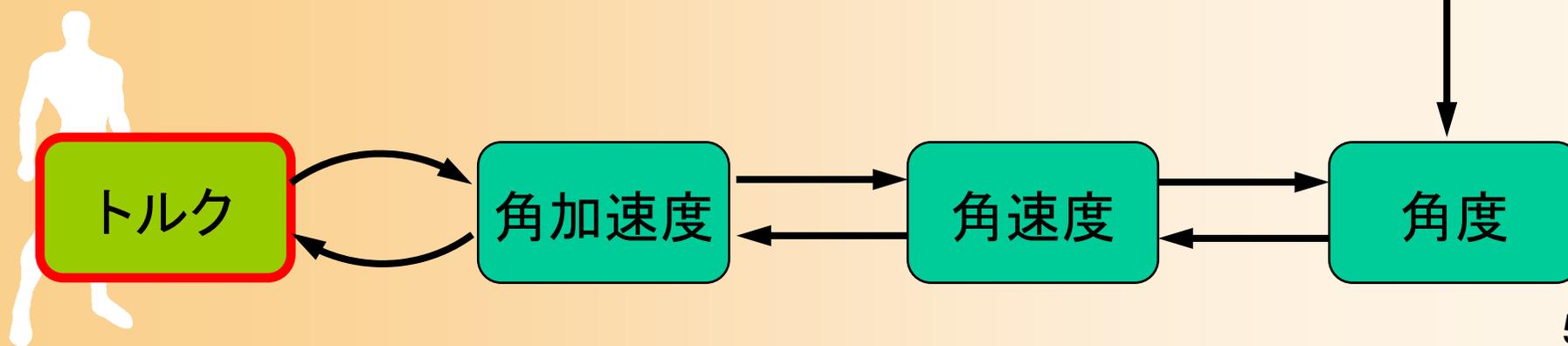
Popovic et al. © 1995

人体モデルの動力学

- 関節に回転力(トルク)が発生することで、運動が実現される
 - 多関節体の運動方程式

$$\tau = H(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) + K(\theta)F$$

関節トルク 関節角加速度



一般的なコントローラ

- PD制御 (Proportional Derivative Control)
 - 各関節のトルクを独立に制御
 - 単純に目標角度との差に比例したトルクを適用

$$\tau = k_g (\theta_{target} - \theta) - k_d \dot{\theta}$$

- 各関節のトルクから、物理シミュレーションにより、運動(姿勢変化)を計算

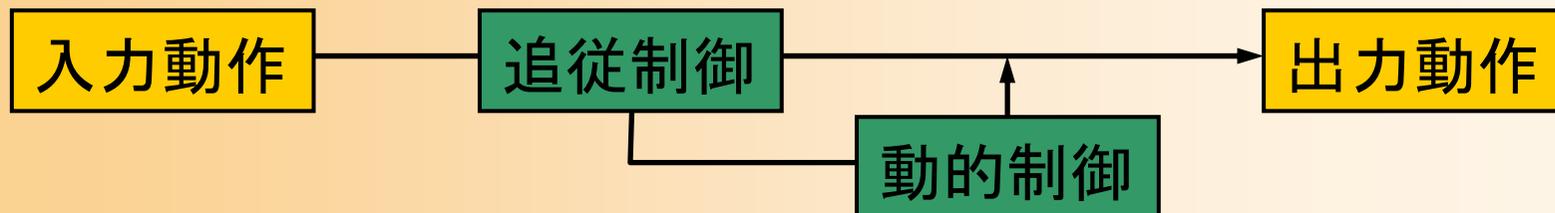
問題点

- パラメタの調整が困難(動作ごとに値が異なる)
- 必ずしも人間らしい動作は生成できない

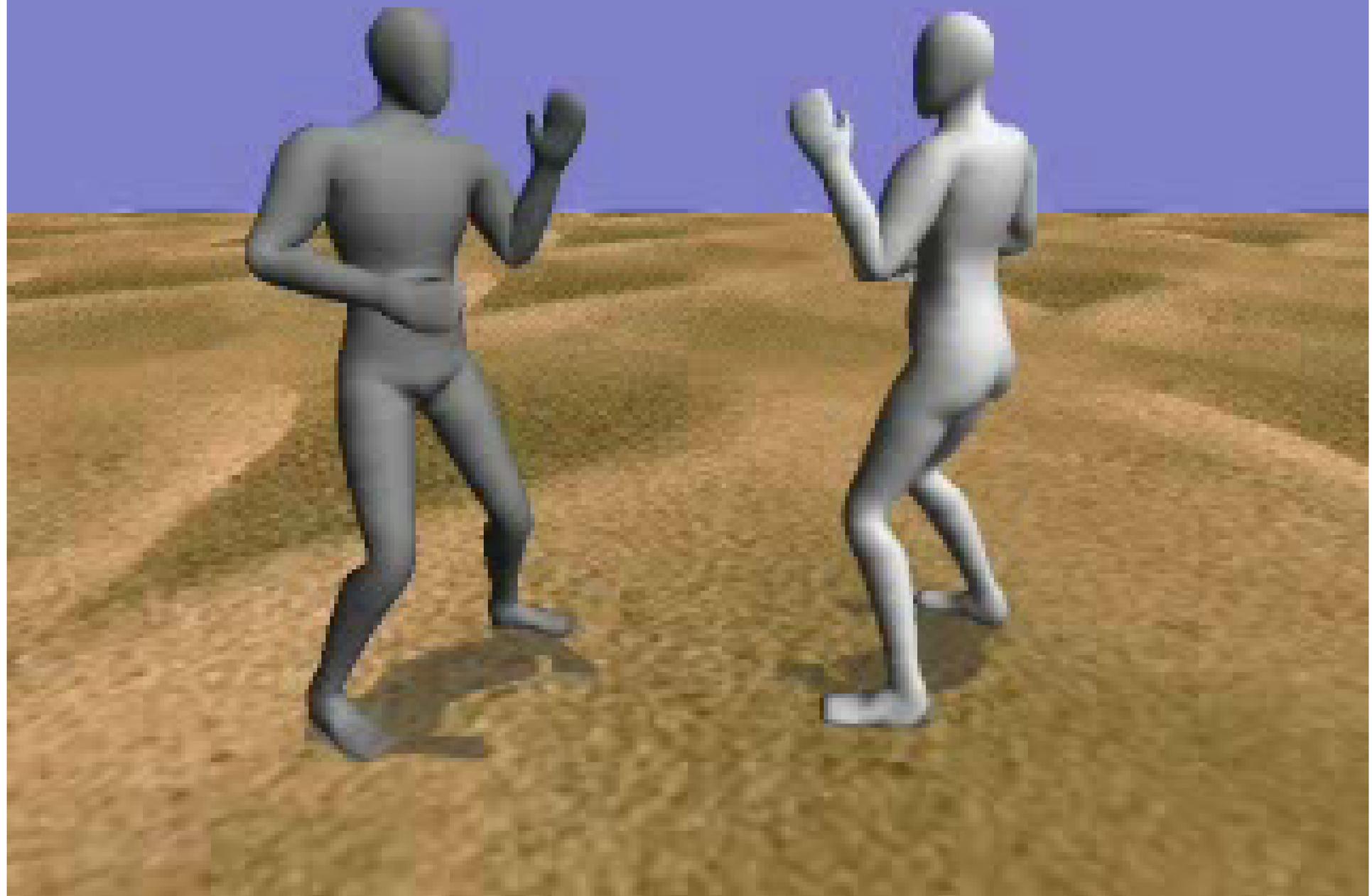


開発手法

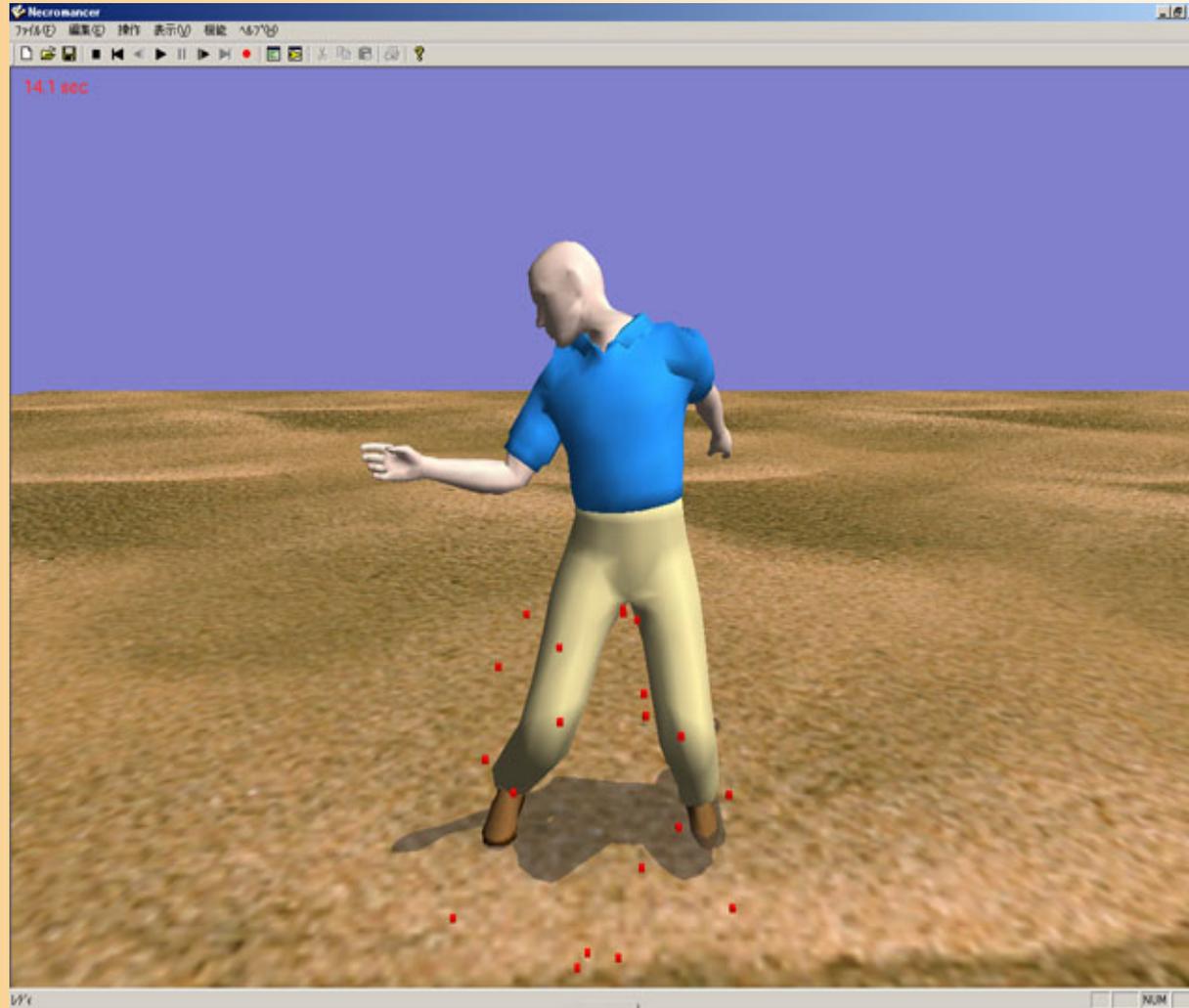
- 動作データをもとにしたダイナミック制御
 - 基本的には入力動作を実行するよう追従制御
 - 力学的な影響に応じて動的制御を行う
 - 全身のバランスを保つような制御
 - 関節の負荷を軽減するような制御
 - 角加速度空間で関節の動きを制御することで、このような制御アルゴリズムを実現



Example #1

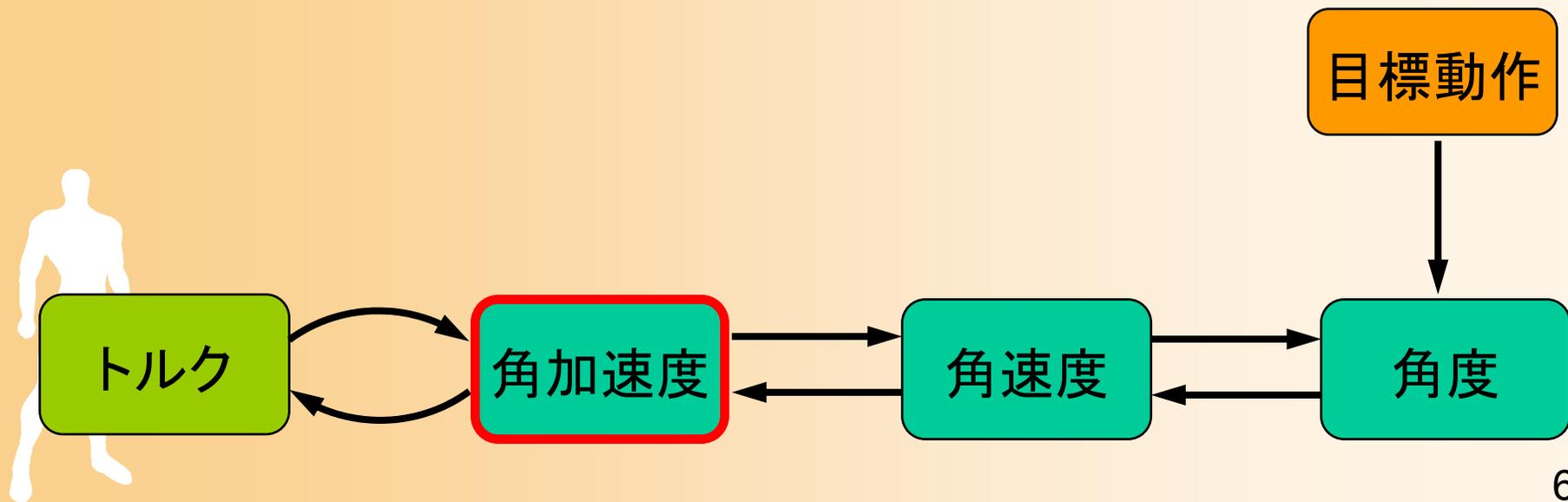


プログラムのデモ



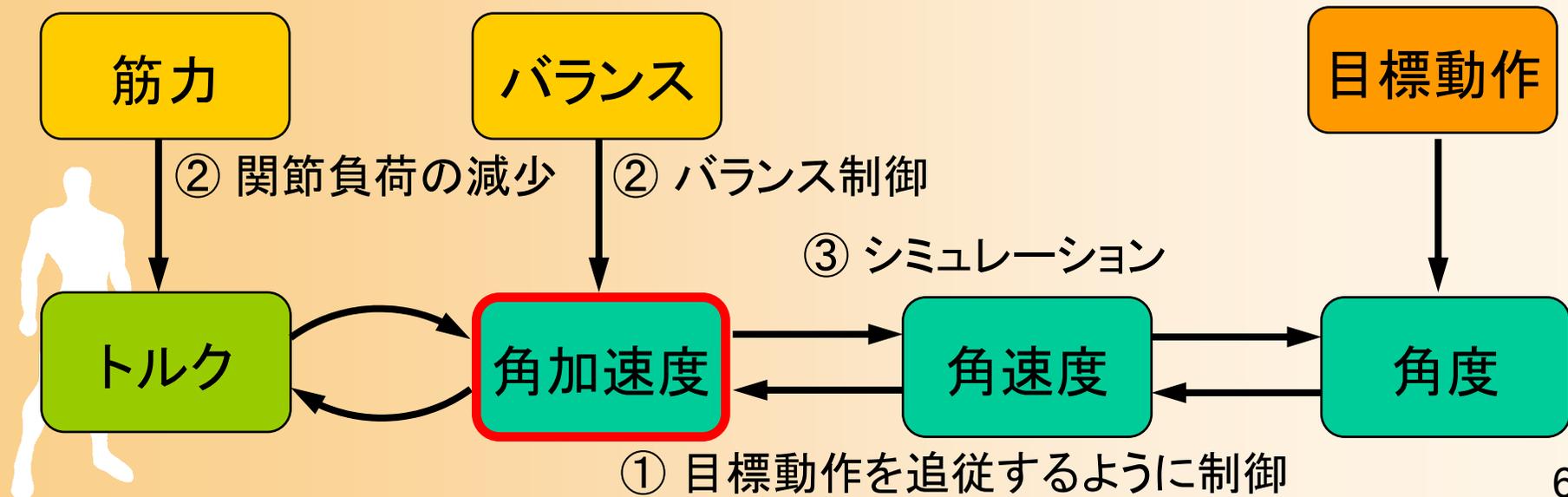
開発手法

- 角加速度空間で各関節を制御
 - トルク制御と比べて安定した制御が可能
 - 物理シミュレーションは不要



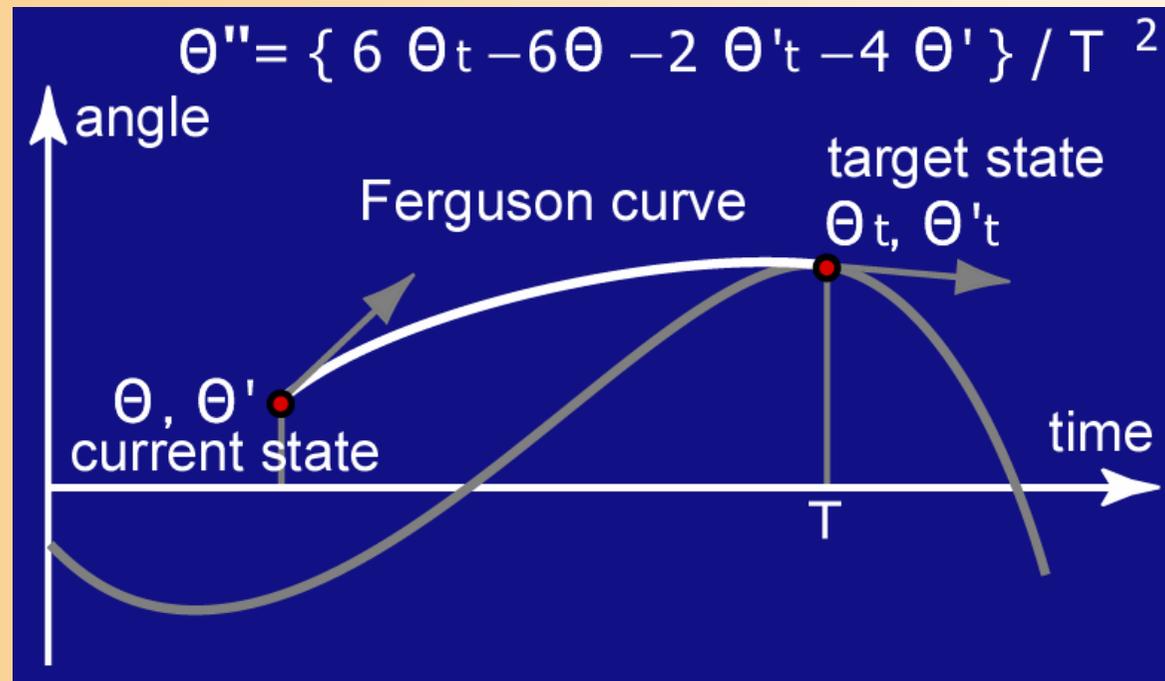
開発手法(続き)

- 基本的には目標動作を追従する
 - 目標動作を追従するような加速度を計算
- 関節負荷・バランスを考慮して加速度を修正
 - 関節相互の影響を考慮して加速度を制御



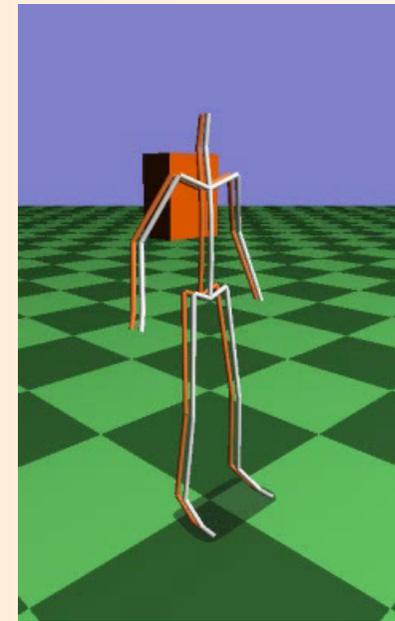
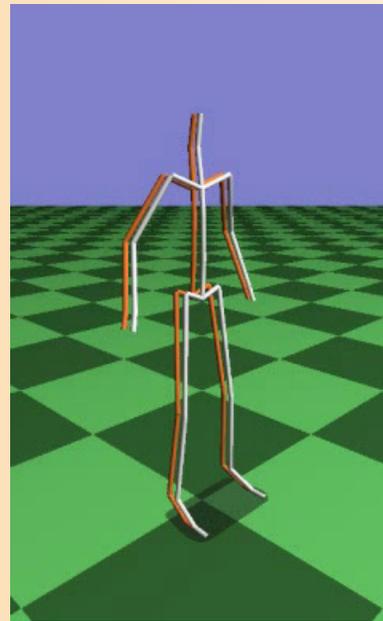
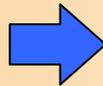
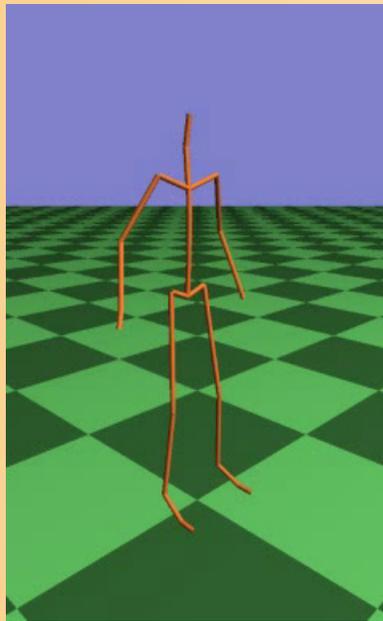
目標動作への追従

- 各関節の角加速度を計算
 - PD制御では、適切な加速度の決定は困難
 - 目標状態と Hermite 曲線から加速度を決定



関節負荷に応じた制御(1)

- ある値以上のトルクが関節に発生したら、その負荷を減らすように、他の関節を運動
 - 関節相互の影響を考慮して、角加速度を修正



関節負荷に応じた制御(1)

- 関節負荷の計算

$$\tau = H(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) + K(\theta)F$$

関節トルク

関節角加速度

- 関節負荷と加速度の関係は、慣性モーメント行列のみにより決まる
→ どの関節を制御すれば良いかが分かる



負荷のかかっている関節

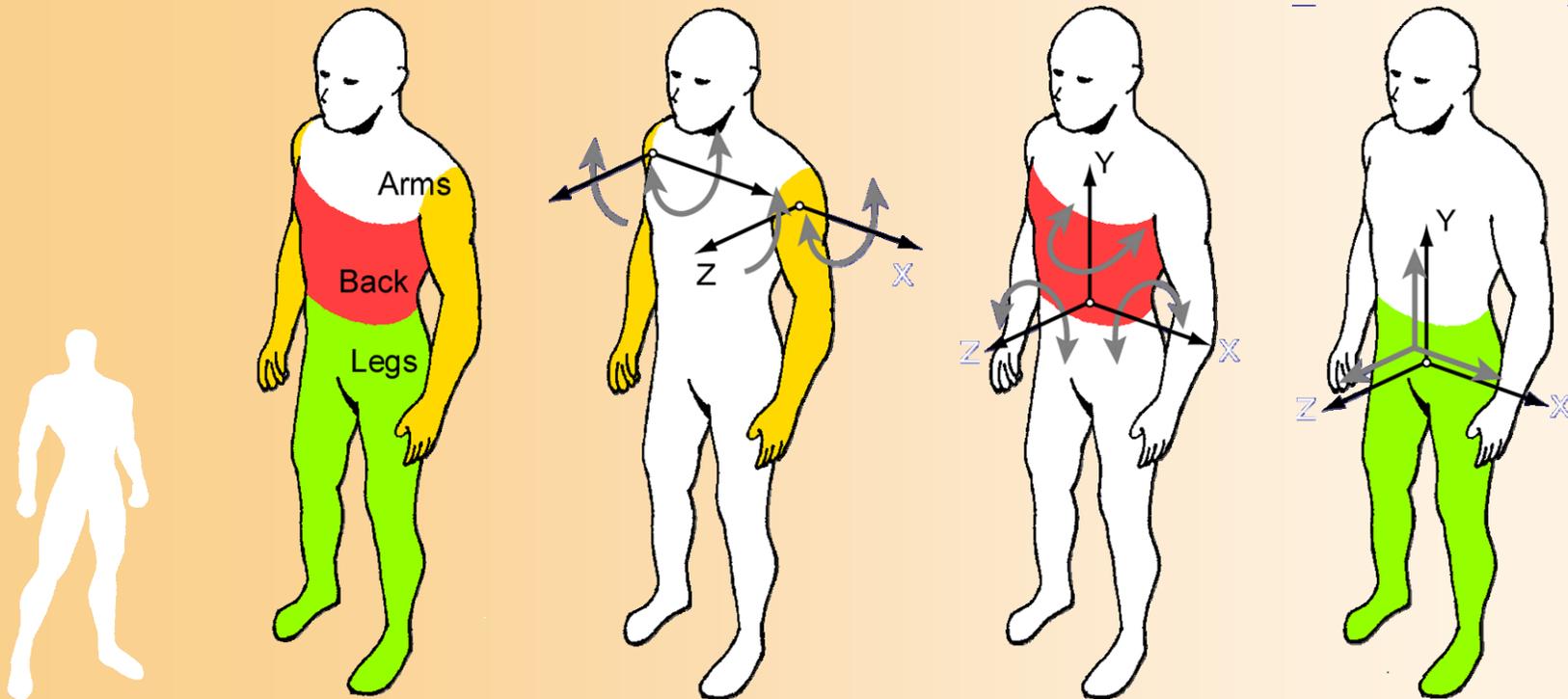
$$\begin{pmatrix} \text{○} \\ \Delta \tau \end{pmatrix} = \begin{pmatrix} \text{○} & \text{○} \\ H(\theta) \end{pmatrix} \begin{pmatrix} \Delta \ddot{\theta} \\ \text{○} \end{pmatrix}$$

制御する関節

関節負荷に応じた制御(2)

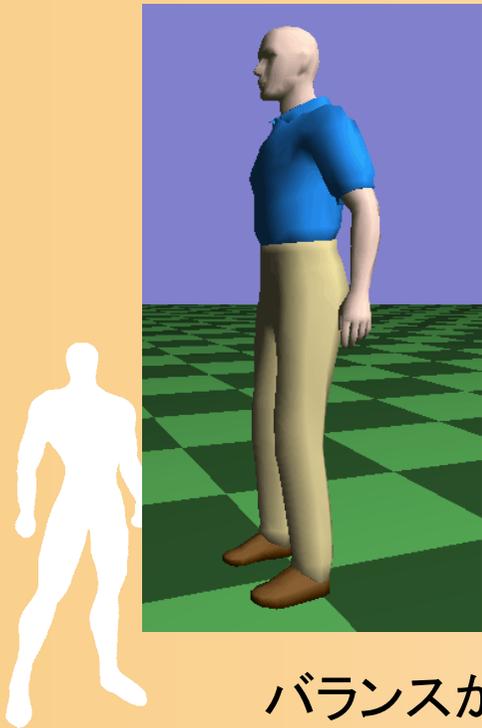
- 主要な関節のみに注目して制御

$$\Delta \tau_{\text{負荷を減らしたい関節}} = H(\theta)' \Delta \ddot{\theta}_{\text{制御する関節}}$$

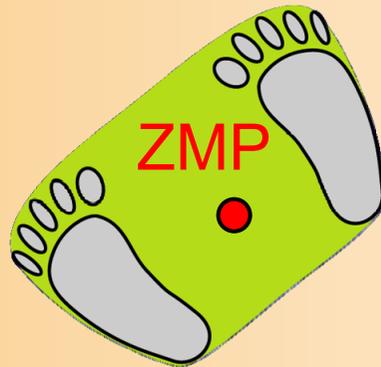


バランスに応じた制御(1)

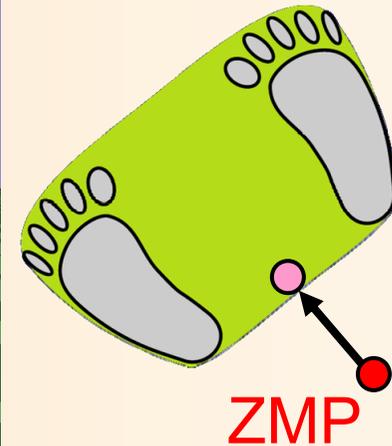
- バランスが崩れたら、バランスを保つように、複数の関節を制御
 - ZMP (Zero Moment Point) によりバランスを判定



バランスが取れた状態



バランスが崩れた状態

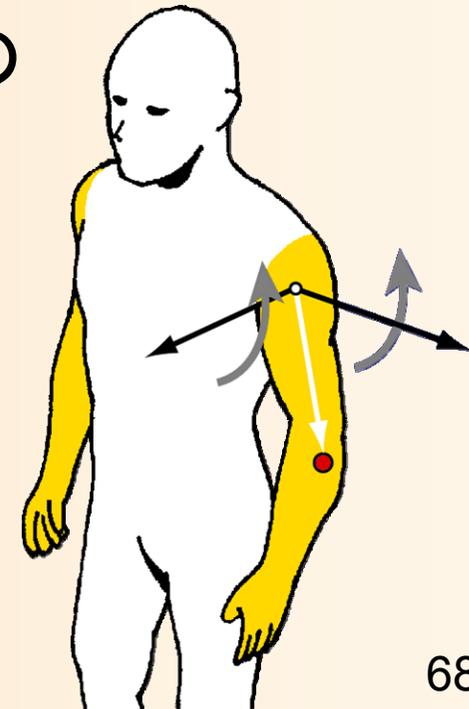


バランスに応じた制御(2)

- ZMPは、全関節の角加速度から計算できる
- 各関節の角加速度が変化したときに、ZMPの位置がどれだけ変化するかも計算可能
 - ZMPを適切な位置に動かすための主要関節の角加速度変化を計算

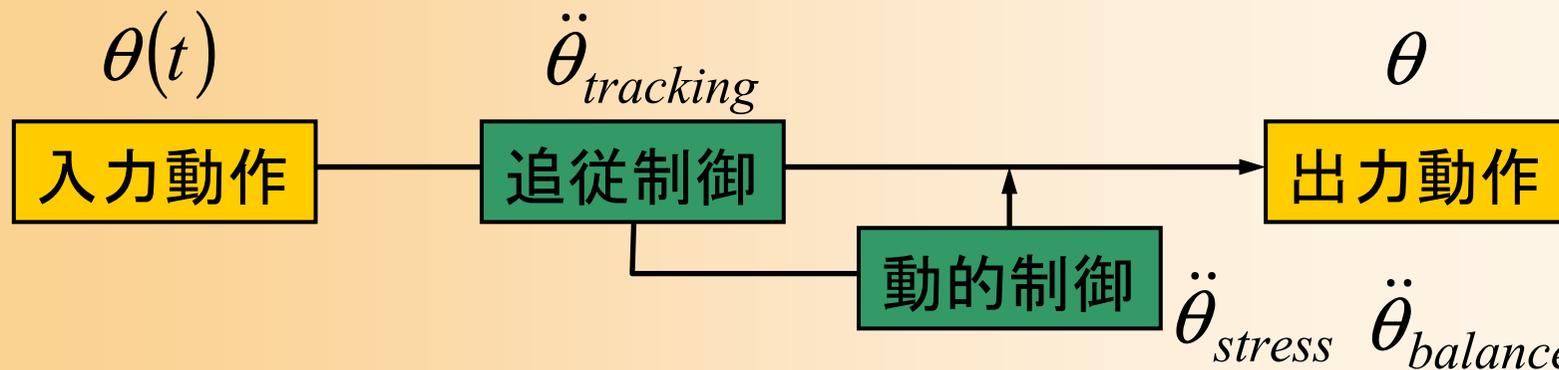


$$\Delta ZMP = \frac{\partial ZMP}{\partial \ddot{\theta}_{arms}} \Delta \ddot{\theta}_{arms}$$

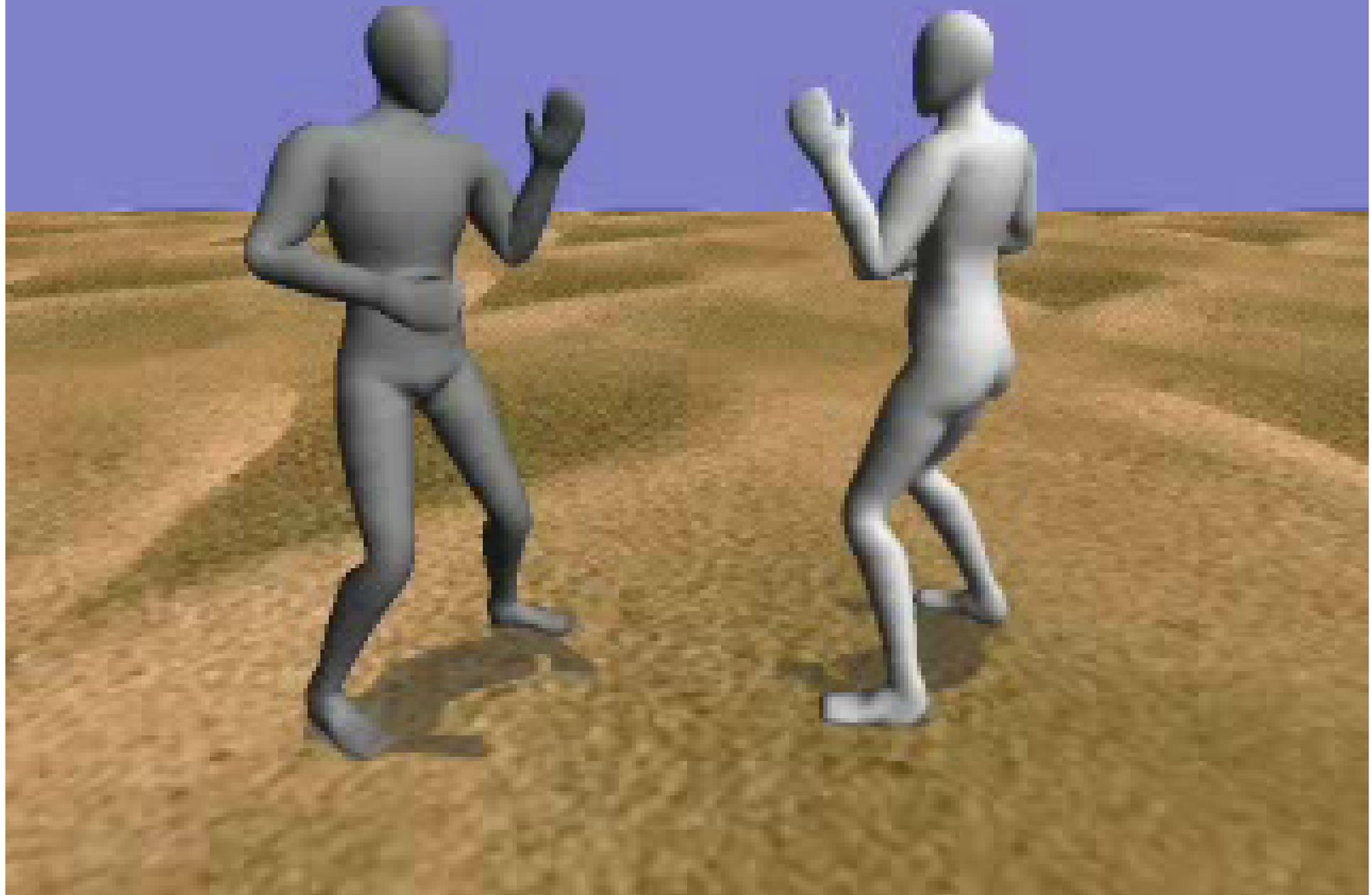


動作制御のまとめ

- 加速度の計算
 - 基本的には入力動作を追従
 - 力学的な影響に応じて動的制御を行う
 - 関節の負荷を軽減する制御、バランスを保つ制御
- 加速度にもとづき運動を計算
 - ※ 実際は、下半身の姿勢は、腰と足の位置より計算



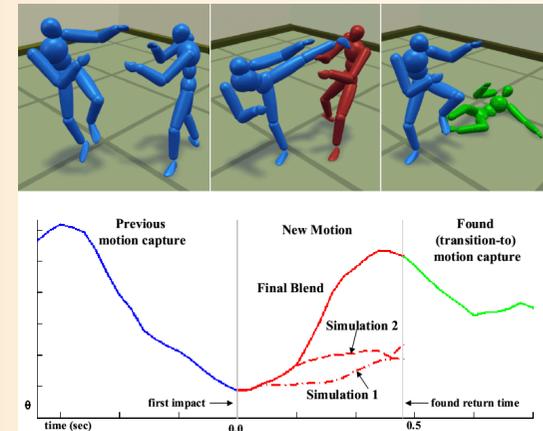
Example #1



関連研究

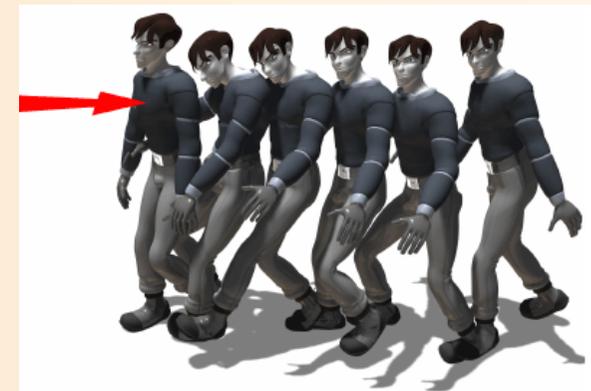
- シミュレーションと動作データの利用 [Zordan 05]

- 衝撃後の動作を物理シミュレーションで計算 (ODE)
- 後にうまくつながるような転倒動作をデータベースから検索



- 動作データの選択と変形 [Arikan 05]

- 衝撃・姿勢に応じたリアクション動作データを検索
- 各部位に一定の速度を加えてIKにより姿勢を変形



最近の研究の動向

- モデルベース手法 vs データベース手法

- モデルベース手法

- 何らかのモデル(プログラム)に従って動きを生成
- 個人ごとの動きの変化の実現などは困難

- データベース手法

- 大量のデータを用意しておくことで、問題を解決
- 準備が必要、適切なデータの検索・変形は困難

- ハイブリッド手法

- 両方をうまく組み合わせる手法
- どのように組み合わせるかは、難しい



Natural Motion 社の技術との関連

- Endorphin, Euphoria

- 基本的には、本研究と同様、人間の動作制御モデルが実装されている



Natural Motion Ltd

- さまざまなビヘイビアの制御モデルが用意されており、ビヘイビアの種類やタイミングを指定することで、自動的に動作が生成される (endorphin)
- 詳細な内部技術は公開されていない
- モデルベースなので、個性を出すことは難しい？

- 独自のエンジンを使うことによるメリット



ロボット工学との関連

- ロボット制御とはやや目的や手段が異なる
 - ロボット制御
 - 力学的に正確な制御が必要
 - 現在ではごく単純な動作しか実現できていない
 - 人間らしい動きの実現はあまり考えられていない
 - 人間とはモータの機構も異なる
 - アニメーション
 - 力学的には厳密でなくても人間らしい動きを実現
 - 動力学などの基礎的な理論はかなり共通
- 本研究の成果も、将来的にはヒューマノイドロボットの実現に貢献することが期待される



今後の研究課題

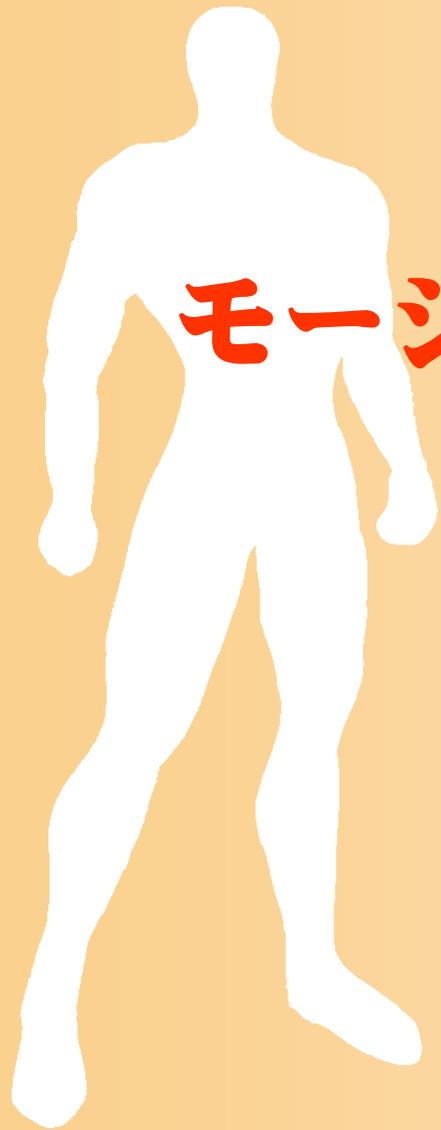
- 現在は、直感的な運動モデルに基づいている
- 実際の人間のモーションデータを反映した運動制御モデルが必要
 - 大量の動作データを解析・学習することによって、より洗練された運動モデルを構築
 - コントローラのパラメータを最適化
 - リアクションを記録・再現
 - 個人ごとの運動制御モデルの区別を実現可能
 - ある人間の動作データを学習させると、その人と同じようなリアクションを行うモデルが実現できる？



仮想人間の動作制御技術

- 動力学を考慮した動作制御
- モーショングラフを利用した自律動作制御
- 学習にもとづく群集の制御
- ペン入力による群集の制御

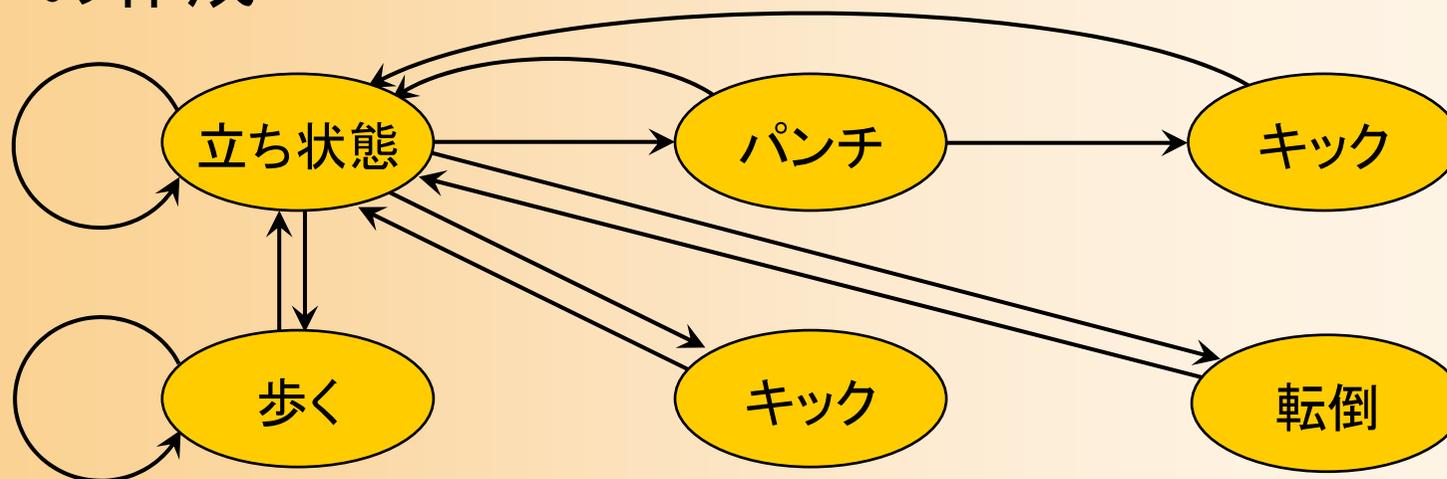




モーショングラフを利用した 自律動作制御

現在の一般的な動作制御手法(確認)

- 動作ツリーを使った動作生成
 - 短い単位動作と、動作間のつながりを、あらかじめ作成



- ユーザーの入力やルールに従って、次に再生する動作を決定
- 必要に応じて、動作ブレンディングや IK を適用



モーショングラフ

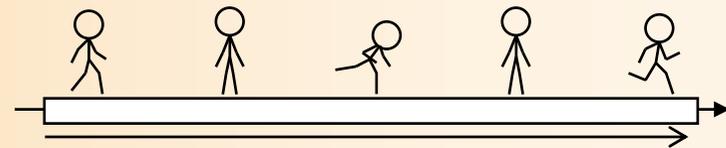
- モーショングラフ [Kovar 02]

- モーションキャプチャデータは、有向グラフ構造に変換できる

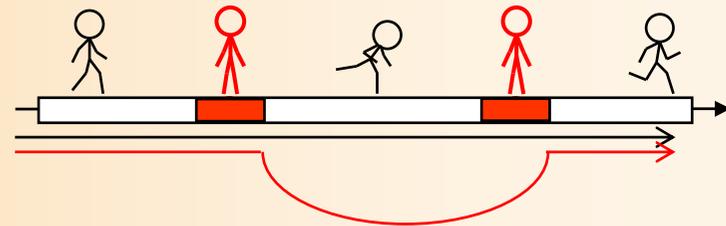
- ノードを順に遷移することで、連続的な動作を生成できる

- 遷移のルールが重要

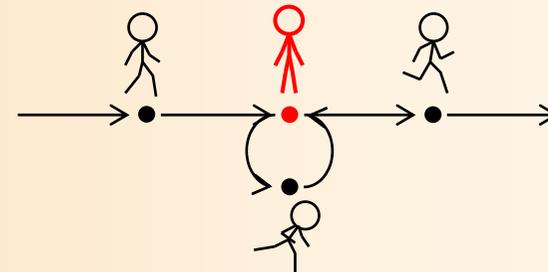
(a) 動作データは単に再生しかできない。



(b) 類似部分があれば、切り替え可能



(c) 類似部分に注目しグラフ構造に変換

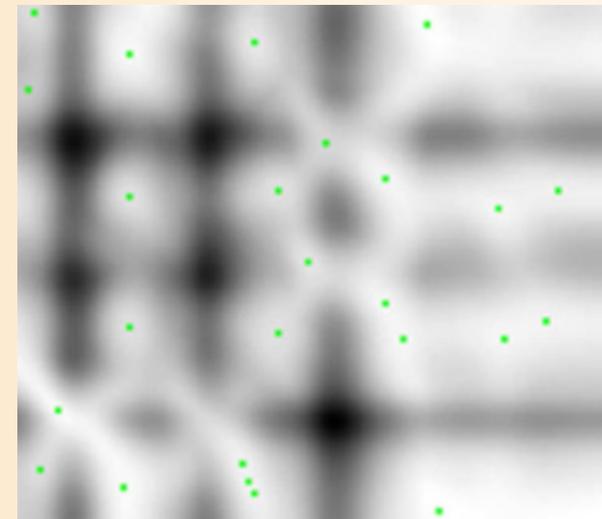


モーショングラフの作成方法

- モーション中の各フレーム同士の姿勢間の距離を計算(マッチウェブ)

- 姿勢同士の距離の評価方法も重要

- 例: 全主要部位同士の距離の和など



Kovar 02

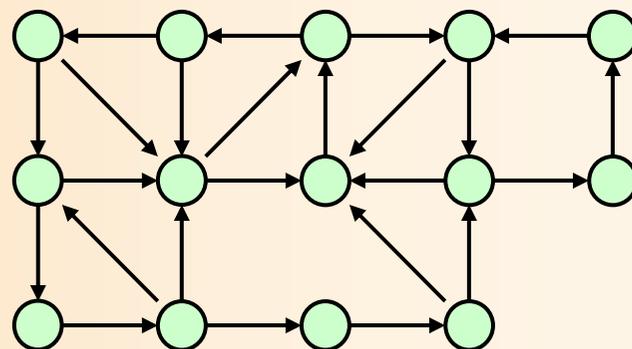


- マッチウェブ中の極小点をモーショングラフのノードとし、ノード間の動作をエッジとする

モーショングラフによる動作生成

- データ構造

- ノード・・・姿勢
- エッジ・・・動作
 - 一般的な動作ツリーとは逆



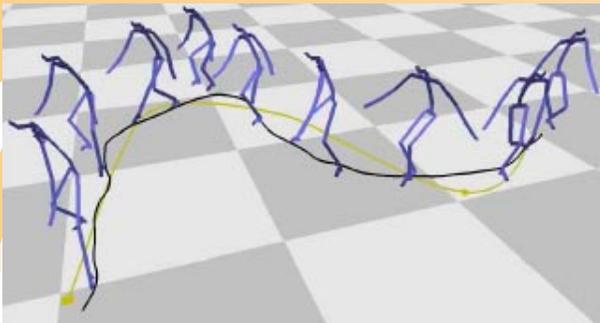
- 動作生成

- ノードを辿りながらエッジの動作を再生していくことで、動作を生成
 - 実際には、ノードの前後で、動作ブレンディングや、足を地面に固定するための IK が必要
- ノード遷移のためのルールが重要

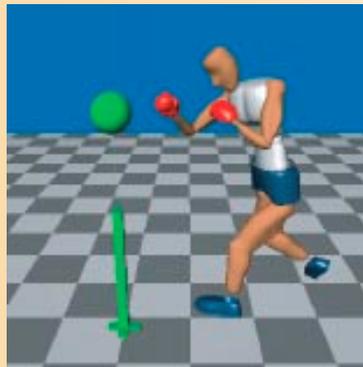


関連研究

- モーショングラフにおけるノード選択条件
 - 歩行パスによる条件[Kovar 02]
 - 手先位置の条件[Lee 05]
 - 動作の種類条件[Arikan 04]
 - 音楽に合わせた動作



[Kovar 02]



[Lee 04]



[Arikan 04]

モーショングラフによる 回避動作の実現

- 回避動作

- さまざまな攻撃をぎりぎりで回避するような華麗な回避動作を実現したい

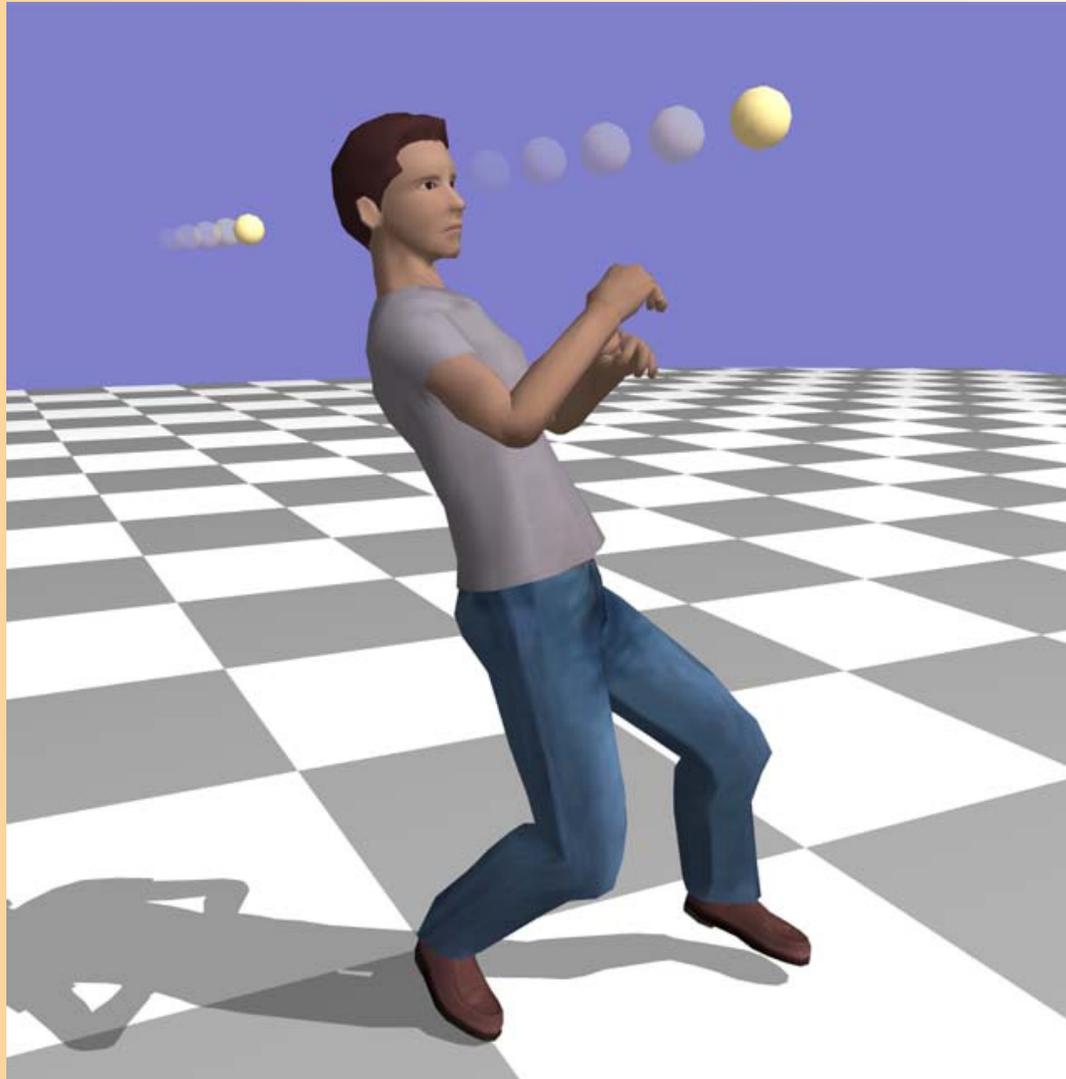


- モーショングラフによる回避動作の実現

- 通常は、ランダムにノードを遷移
- 攻撃が来たときには、攻撃をうまく回避できるように、ノードを遷移
 - 全てのエッジについて衝突判定を行うと、多くの計算時間がかかるため、前計算により高速化(詳細は省略)
 - 必要に応じて動作の再生速度も変更



デモ



モーショングラフの利点

- モーションデータから自動的に構築できる
- 連続的な動作が無限に生成できる
 - 毎回異なる動作が生成される
 - 基本的には同じ動作の繰り返しだが、ノード数が十分にあれば、十分に自然に見える
- デメリット
 - 毎回異なる動作が生成される
 - ゲームによっては必ずしも良いとは限らない？
 - 適切な遷移ルールが必要(計算速度の考慮)



仮想人間の動作制御技術

- 動力学を考慮した動作制御
- モーショングラフを利用した自律動作制御
- 学習にもとづく群集の制御
- ペン入力による群集の制御



学習にもとづく群集の制御

- 自律動作する仮想人間の必要性
 - 特に映画やコンピュータゲームでの群衆シーン
 - 個々のキャラクタの動きを制作するのは不可能
 - 群衆モデル(一定のルールに従ってキャラクタを制御)
 - 現在は、プログラマが手作業でルールを設定
 - 効果的なルールを設定するのは困難、手間がかかる



Loads of the Rings, 2002



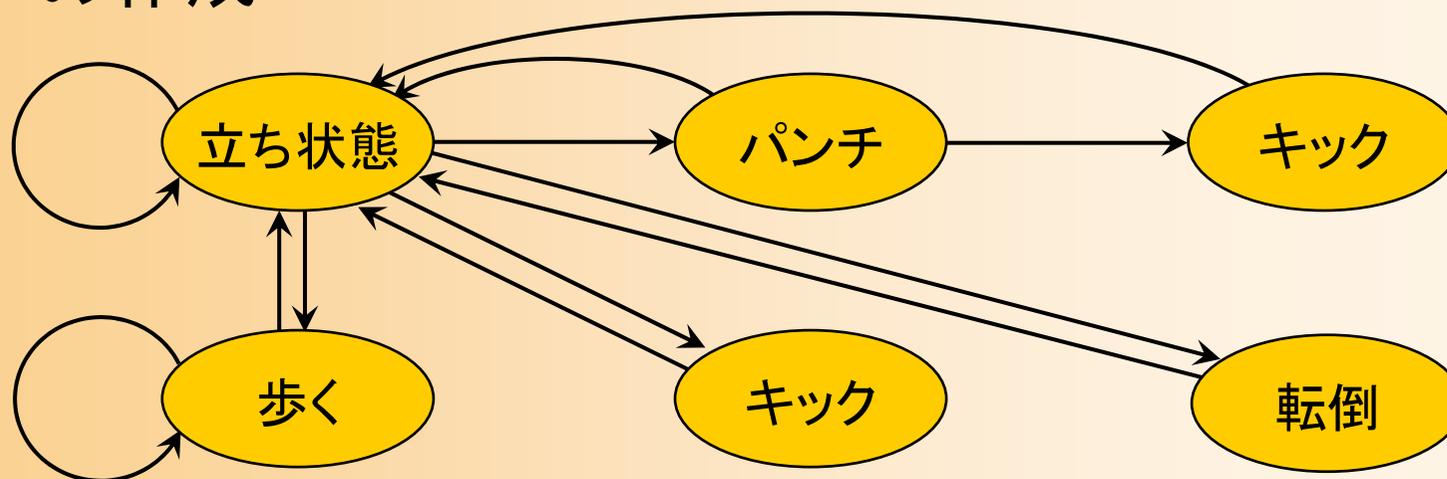
Chronicles of NARNIA, 2005



NINETY-NINE NIGHTS, 2005

現在の一般的な動作制御手法(確認)

- 動作ツリーを使った動作生成
 - 短い単位動作と、動作間のつながりを、あらかじめ作成



- ユーザーの入力やルールに従って、次に再生する動作を決定
- 必要に応じて、動作ブレンディングや IK を適用 88



背景技術

- 映画やゲームでの群衆モデルの実現方法

- プログラムによる動作ルールの記述

- if (条件) then (動作) の形式の動作ルールを多数、ゲームなどのプログラム内に直接記述

- GUI環境によるルールの記述

- 群衆アニメーションソフト

Massive が有名

- 多数の映画で使用されている

- ファジールールの条件をGUI画面上で組み合わせることができる



Massive

→ 高度な動作ルールの設定・修正は困難



操作ログからの動作ルールの自動生成

操作ログ記録アプリケーション

入力: ユーザーの操作



各時刻
で記録

操作ログ 仮想人間の動作+特徴量

ユーザー動作	前	左前	シュート	→t
ユーザー位置x	-1m	-1m	-2m	→t
ユーザー位置z	4m	3m	2m	→t
	...			

動作開始時の特徴量と
開始動作を抽出

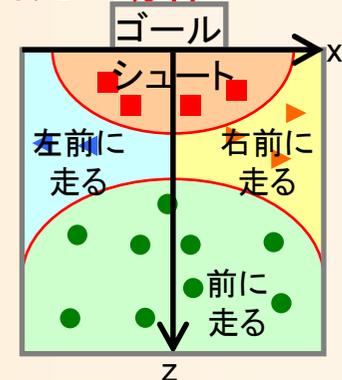
動作開始条件の教師データ

動作開始時の特徴量			開始動作
位置x	位置z	...	
-1m	4m	...	前に走る
-1m	3m	...	シュート
...

SVMを用いて
動作ルール抽出

動作開始条件の範囲を推定
任意の特徴量での開始動作がわかる

出力: 動作ルール



戦闘動作の動作ルールの実験例



学習に使った入力データ(人間が操作)



学習した動作ルールによる動作生成

サッカーの動作ルールの実験例



学習に使った入力データ(人間が操作)



学習した動作ルールによる動作生成

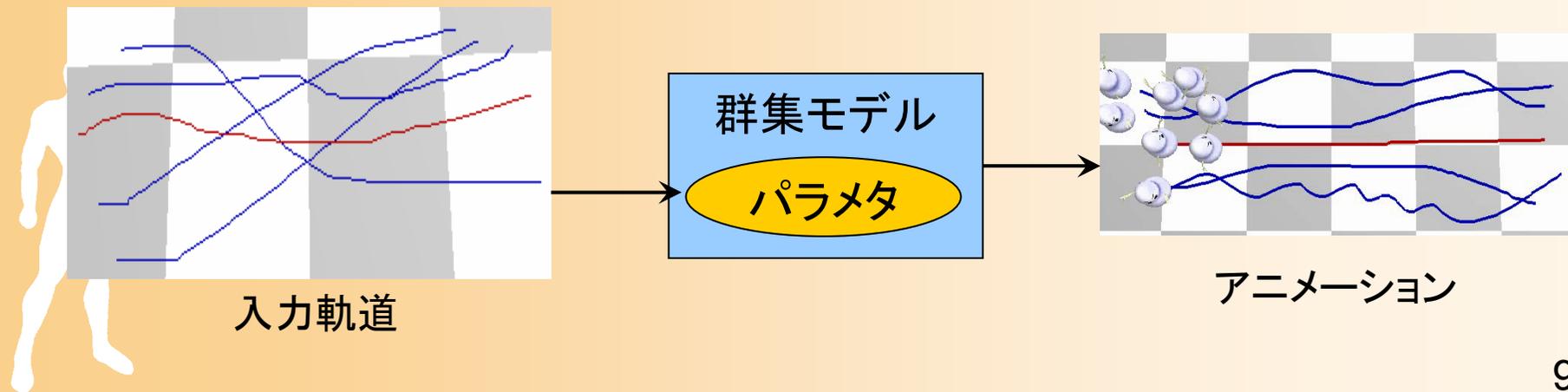
仮想人間の動作制御技術

- 動力学を考慮した動作制御
- モーショングラフを利用した自律動作制御
- 学習にもとづく群集の制御
- ペン入力による群集の制御

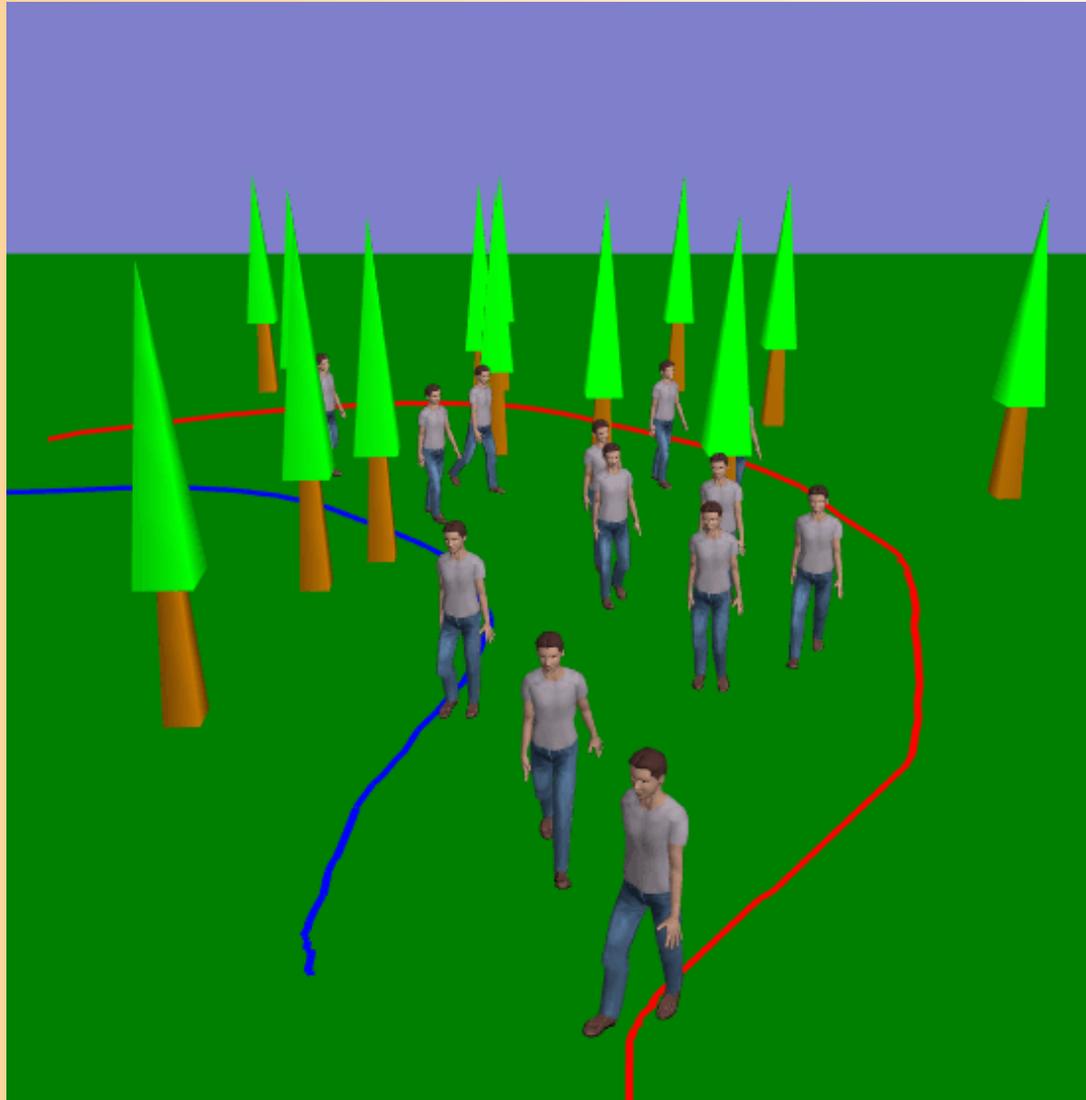


ペン入力による群集の制御

- 一般的な群集モデル
 - 一定のルールに従って各エージェントを動作
 - 期待通りに動かすには、パラメタの微調整が必要になる(初期位置、目標軌道、速度、距離等)
- 開発手法

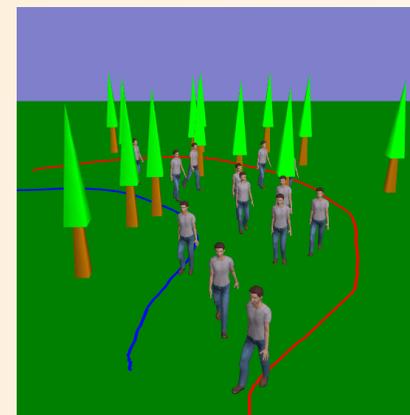
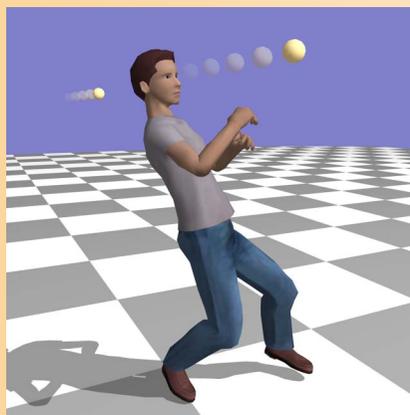
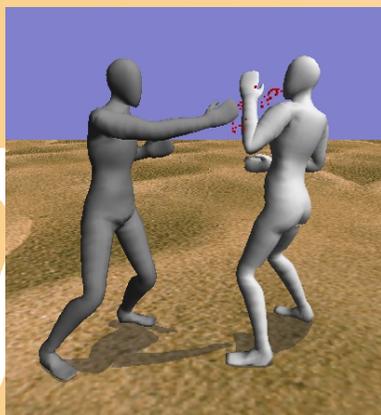


デモ



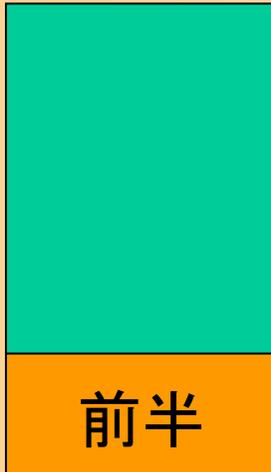
まとめ

- 仮想人間の動作制御技術
 - 動力学を考慮した動作制御
 - モーショングラフを利用した自律動作制御
 - 学習にもとづく群集の制御
 - ペン入力による群集の制御



構成

14:50~
16:10



- 前半(20分)

- イン트로ダクション
- 研究紹介

- 仮想人間の操作インターフェース
- 衣服・髪の実時間シミュレーション

16:40~
18:00



- 後半(40分)

- 研究紹介・技術解説

- 仮想人間の動作制御技術

- 大学とゲーム業界の連携について





大学とゲーム業界の連携について

研究を応用する際の問題(1)

- 研究で開発した手法やソフトウェアを、そのまま実用化するのには、難しいことが多い
 - 研究では、必ずしも、完成度は要求されない
 - 新規性(新しい技術のアイデア)が重要
 - 研究アイデアが実現可能かどうかを実証するための、最低限のソフトウェアを開発することが必要
 - ソフトウェア(デモ)の完成度を高めるのは、時間的に難しい
 - 広範囲に利用可能な技術の方が、評価が高い
 - 特定のゲームに特化したような技術は研究しにくい



研究を応用する際の問題(2)

- 研究・ソフトウェア開発をできる人材・時間の不足
 - 学生(学部生・修士学生)が研究を担当
 - 基本的なプログラミング技術から教える必要がある
- 新しい技術を使うだけで、ゲームが面白くなるか(売れるか)、とは限らない



ゲーム業界との連携(1)

- 研究技術のゲームへの応用
 - 研究成果をそのまま実用化することは困難？
 - ある程度、共同で開発をすることが望ましい
 - 研究の初期の段階から、応用を想定して研究を始めることが望ましい
- 共同研究・受託研究
 - 実際にゲーム業界で問題となっている課題を提案していただいて、取り組むことも可能



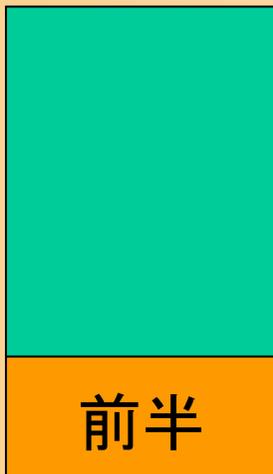
ゲーム業界との連携(2)

- 人材・技術交流
 - 大学→企業
 - セミナー・技術相談
 - 学生の就職先
 - インターンシップ
 - 企業→大学
 - 共同研究(出向)(短期 or 長期)
 - 博士課程での研究(フルタイム or パートタイム)



まとめ

14:50~
16:10



- 前半(20分)

- イン트로ダクション

- 研究紹介

- 仮想人間の操作インターフェース
- 衣服・髪の実時間シミュレーション

16:40~
18:00



- 後半(40分)

- 研究紹介・技術解説

- 仮想人間の動作制御技術

- 大学とゲーム業界の連携について

参考文献(1)

- 尾下の論文は、ほぼ全て、尾下研究室のサイトで公開(発表文献のページ)

www.oshita-lab.org



参考文献(2)

- [Arikan03] Okan Arikan, David A. Forsyth, James O'Brien. **Motion Synthesis from Annotations**. ACM Transactions on Graphics (Proc. of SIGGRAPH 2003), 33(3), pp 402-408, 2003.
- [Arikan05] Okan Arikan, David A. Forsyth, James F. O'Brien. **Pushing People Around**. Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 59-66, 2005.
- [Kovar 02] Kucas Kovar, Michael Gleicher, Fedderic Pighin, **Motion Graphs**, SIGGRAPH 2002.
- [Lee 04] Jehhee Lee, Kang Hoon Lee, **Precomputing Avatar Behavior From Human Motion Data**, Eurographcis/ACM SIGGRAPH Symposium on Computer Animation 2004.
- [Zordan05] Zordan, V. B., Chiu, B., Majkowska, A., Fast, M. **Dynamic Response for Motion Capture Animation**. ACM Transactions of Graphics (Proc. of SIGGRAPH 2005), 24(3), 697-701, 2005





尾下 真樹

oshita@ces.kyutech.ac.jp

九州工業大学 情報工学部

システム創成情報工学科

尾下研究室

www.oshita-lab.org