

# コンピューターグラフィックスS

## 第14回 アニメーション

システム創成情報工学科 尾下真樹

## 今回の内容

- アニメーション
  - 動きのデータの生成・利用

The diagram illustrates the animation pipeline with the following components:

- オブジェクトの作成方法** (Object creation method)
- オブジェクトの形状表現** (Object shape representation)
- 表面の素材の表現** (Surface material representation)
- 動きのデータの生成** (Motion data generation) - highlighted with a red box
- 光源** (Light source)
- 光の効果の表現** (Light effect representation)
- カメラから見える画像を計算** (Calculate image seen from camera)
- 画像処理** (Image processing)

## 今回の内容

- 前回の復習
- アニメーション
  - アニメーションの原理
  - 人間のアニメーション
  - 人体以外の物体のアニメーション
  - リアルタイム・アニメーション
- 講義のまとめ
- 授業アンケート

## 教科書(参考書)

- 「コンピューターグラフィックス」  
CG-ARTS協会 編集・出版(3,200円)  
- 5章
- 「ビジュアル情報処理 -CG・画像処理入門-」  
CG-ARTS協会 編集・出版(2,500円)  
- 5章
- 「3DCGアニメーション」  
栗原恒弥 安生健一 著、技術評論社  
出版(2,980円)

## 前回の復習

## 光のモデル

The diagram shows a surface with normal vector  $N$  and view vector  $V$ . Incident light  $L$  is shown as a vector from the light source. The reflected light  $R$  is shown as a vector from the surface. The diagram also indicates environmental light (環境光) and transmitted light (透過光).

$$I = I_a k_a + \sum_{i=1}^{n_l} I_i \left[ k_d (N \cdot L) + k_s (R \cdot V)^n \right] + k_r I_r + k_t I_t$$

環境光 (周囲から来る光)      透過光

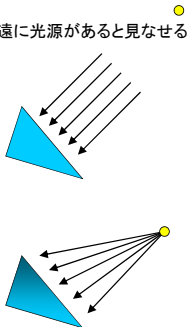
拡散・鏡面反射光 (光源から来る光)      鏡面反射光 (映り込み)

環境光      拡散反射光      鏡面反射光 (局所照明)      鏡面反射光 透過光 (大域照明)

それぞれの光源からの光(局所照明)      大域照明

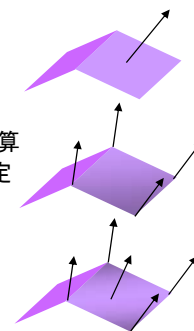
### 光源の種類と設定方法

- 平行光源
  - $(x,y,z)$  の方向から平行に光が来る
  - 光源位置の  $w$  座標を  $0.0$  に設定
- 点光源
  - $(x,y,z)$  の位置に光源がある
  - 光源位置の  $w$  座標を  $1.0$  に設定



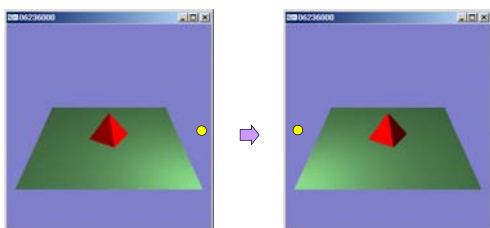
### シェーディング

- フラットシェーディング
  - 面の法線から面の色を計算
- グローシェーディング
  - 頂点の法線から頂点の色を計算
  - 頂点の色から、各点の色を決定
- フォンシェーディング
  - 頂点の法線から、面内の各点 (ピクセル) の法線を計算
  - 各点の法線から、色を計算



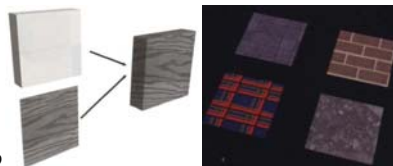
### プログラミング演習

- 光源の種類(点光源・平行光源) や位置の設定を変更し、描画結果を確認



### テクスチャマッピング

- マッピング
  - 面を描画する時に、面の表面に画像を貼り付ける技術
  - 複雑なモデリングをすることなく、細かい模様などを表現できる

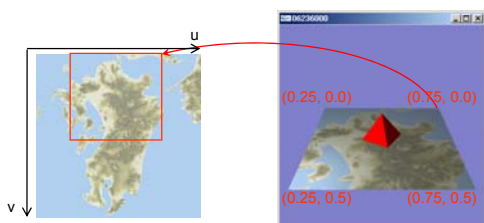


基礎知識 図3-19

基礎と応用 図5.2

### プログラミング演習

- 地面にテクスチャマッピングを適用して描画
  - 各頂点に、テクスチャ座標  $(u,v)$  を指定



### アニメーションの原理

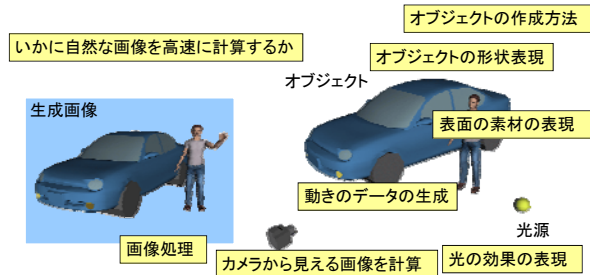
### 3次元グラフィックス(復習)

- CG画像を生成するためのしくみ
  - 仮想空間にオブジェクトを配置
  - 仮想的なカメラから見える映像を計算で生成
  - オブジェクトやカメラを動かすことでアニメーション



### 3次元グラフィックスの技術(復習)

- コンピュータグラフィックスの主な技術



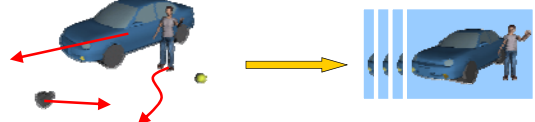
### コンピュータグラフィックスの分類(復習)

- 2次元グラフィックス
  - 画像データ(2次元)を扱う
    - ・ 画像処理、符号化などの技術
- 3次元グラフィックス
  - シーンデータ(3次元) → 画像データ(2次元)
    - ・ 出力データは、あくまで2次元になることに注意



### アニメーション(復習)

- 2次元アニメーション
  - 少しずつ変化する画像を連続的に生成することで、アニメーションになる
- 3次元アニメーション
  - 動きのデータを与えて、連続画像を生成



### アニメーションの原理

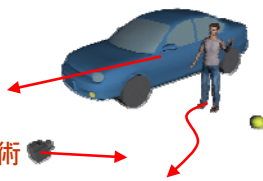
- 少しずつ変化する画像を連続して表示することでアニメーションとして見える
  - 1秒間に10枚~30枚毎程度 (fps: frame per sec)
    - ・ テレビ 30fps (60fps)、映画 24fps、アニメ 12fps、TVゲーム 30 or 60 fps
  - 3次元アニメーションは、少しずつ物体の位置・向きを変えながら、連続して描画することで実現

### 2D・3Dアニメーションの比較

- 2次元のアニメーション(セルアニメ)
  - モデリングに比べると絵を描くのは簡単
    - ・ カメラから見える絵だけを描けばいい
  - 1枚1枚描く必要があるのでアニメーションは大変
  - 最近はこちらもほとんどコンピュータ上で処理
    - ・ コンピュータ上で、原画の自動補間や彩色作業などを行う
- 3次元のアニメーション(CGアニメ)
  - モデリングに非常に多くの時間がかかる
    - ・ きちんとした3次元のデータを作りこむ必要がある
  - 一通りモデリングが完成すれば、動きを与えるだけで、比較的簡単にアニメーションが作れる

### アニメーションの生成

- アニメーション = 動きの生成 + 画像生成
  - 3次元空間中のオブジェクトに動きのデータを与える
  - 動きのデータをどう生成するか?
- 動きのデータを扱う技術
  - 単純な物体の動き
  - 人間(多関節体)の動き
  - 髪や衣服などのやわらかい物体、炎・流体など

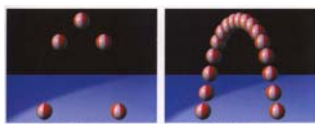


### 物体の動きのデータ

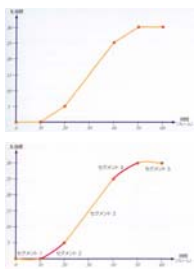
- 画像を描画するためには、各フレームにおける物体の位置・向きの情報が必要
  - 全てのフレームの情報をいちいち指定すると非常に時間がかかる
- キーフレームアニメーションが主に使われる
  - いくつかの代表的なフレームにおける、オブジェクトの位置・向きを指定
  - 指定されていないフレームは、前後のフレームの情報から位置・向きを決定

### キーフレームアニメーション

- 動きのキーとなる(時刻,位置)の組を指定し、動きのデータを生成
  - キーフレームの補間の方法には多くの種類がある
    - 単純補間(線形補間)
    - ベジェ曲線、スプライン曲線



参考書 図3.3



参考書 図3.9

### 物体の動力学シミュレーション

- 物理法則に従ったアニメーション
  - 実際の物体は物理法則に従う
    - 重力、衝突、摩擦力、力を加えると運動する、落下、等
  - キーフレームアニメーションでは、これらの物理法則は考慮されないため不自然に見える可能性がある
- 動力学シミュレーション
  - 初期状態を与えると、各フレームごとに運動方程式を数値的に解いていくことで、アニメーションが生成される
  - 望むような結果を得るような初期条件の設定が難しいという問題がある
  - キーフレームアニメーションとの使い分け

### 人間のアニメーション

- 人体モデルの表現
- 人体モデルの動作データの作成
- 人体モデルの動作データの変形
- 動力学を考慮した動作データの生成・変形

### 人体モデルの表現

### 人間の各要素の表現

• 人体の表現

- 全身の骨格・形状の表現
- 顔の表現
  - 細かい表情変化を表現するためには体とは別のモデルが必要

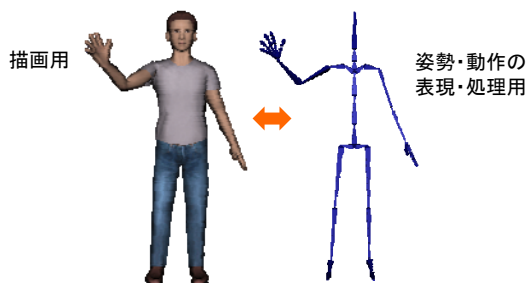


• 付属物の表現

- 髪の毛や衣服など
- シミュレーションによる動きの計算

### キャラクターの表現

形状モデル (ポリゴンモデル)      骨格モデル (多関節体)



### 人間の骨格の表現

• 人間を多関節体として扱う

- 人間の骨格をモデル化するのに、大体40~200くらいの自由度が必要になる
  - 基本的な関節だけで40弱
  - 手の指や足の指なども入れると200自由度くらい必要
- 全関節の角度により人間の姿勢を表せる
- 人体の形状モデルは、骨格の動きに応じて変形

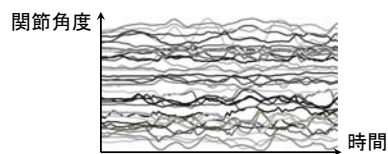


### 姿勢・動作データの表現

• 多関節体の姿勢の表現

- 各関節の回転角度 (40~自由度)
- 腰の位置・向き (6自由度)

• 関節角度(姿勢)の時間変化により多関節体の動作を表現

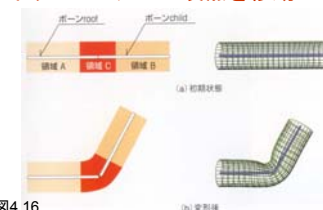


### 動作データの例



### ワンスキンモデル

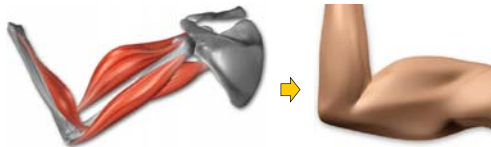
- 人間の形状を全身で1つのポリゴンモデルとして作成
- 骨格モデルの変形に応じてポリゴンモデルの頂点を移動



参考書 図4.16

### より高度な変形モデル(1)

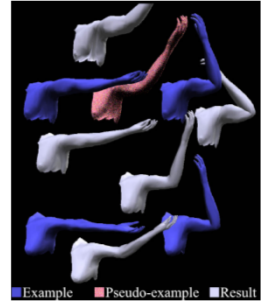
- 人間の筋肉をモデリング
  - 骨格の動きに応じて筋肉を伸縮
  - 筋肉の動きに応じて皮膚を変形
    - ※ 単純なワンスキンモデルよりもリアルな変形を実現できる



[Softimage XSI]

### より高度な変形モデル(2)

- 形状合成
  - あらかじめ入力された複数のサンプル形状を適切な重みでブレンドすることで、変形を実現
  - 姿勢に応じて、適切なブレンドの重みを計算する必要がある



[Sloan 01]

### 顔のモデリングと表情生成

- 顔のモデリングと表情生成
  - 顔の表情は人間の感情を伝える上で非常に重要
  - 顔のモデリングや変形の手法が研究されている
    - テクスチャによる表情の表現
    - 筋肉ベースの顔の形状の変形
    - サンプル形状のブレンドによる変形



参考書 図4.18 [Pighin 98]

### リップシンク

- セリフに合わせて唇の動きを生成する技術
  - あらかじめ、各音素ごとの口の形のデータを用意しておく
  - セリフを音素に分解し、各音素に応じて口の形を変形してゆく



参考書 図4.20

### 人体モデルの動作データの作成

### 姿勢・動作データの表現

- 多関節体の姿勢の表現
  - 各関節の回転角度(40~自由度)
  - 腰の位置・向き(6自由度)
- 関節角度(姿勢)の時間変化により多関節体の動作を表現

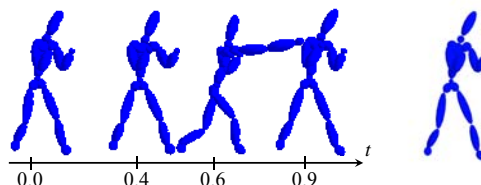


### 動作データの作成方法

- 動作データの主な作成方法
  - キーフレームアニメーション
  - モーションキャプチャ
  - (動力学シミュレーションによる方法)
    - 3番目の方法は、やや特殊な方法なので、後述

### キーフレームアニメーション

- 動作のキーとなる姿勢を手作業で作成
  - キー姿勢の時刻・姿勢を作成
    - 各関節の回転を操作、各部位の位置を操作
- キー姿勢の間を自動的に補間して動作生成



### キーフレームの編集

- キーフレームにおけるキー姿勢の編集
  - 基本的にはアニメータが手作業で作成
    - かなりの時間・労力がかかる
- 基本的な姿勢の作成方法
  - 関節の回転を操作
  - 各部位の位置を操作
    - 部位の位置に合わせて関節の回転を自動計算 (インバース・キネマティクス)



### モーションキャプチャ

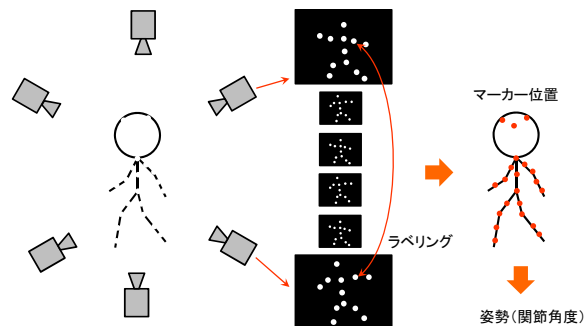
- 人間の身体にセンサをつけて、人間の動作を計測・取得する方法
- モーションキャプチャ機器の方式
  - 光学式、磁気式、機械式、RGB-Dカメラ式、等
  - 計測に使用するセンサーの種類に違いで、さまざまな方式がある

### モーションキャプチャ機器の種類(1)

- 光学式
  - 現在、主に使われている方式
  - 体の各部位にマーカを付けてカメラで撮影
  - 複数のカメラから姿勢を計算
  - 姿勢の計算には時間がかかる
- 磁気式
  - 磁界を発生させて、体の各部位につけたセンサの位置・向きを計算
  - 比較的高速



### 光学式モーションキャプチャ





### モーションキャプチャ機器の種類(2)

- **機械式**
  - 体の各関節に回転角度を計測するための計器を身につける
  - 身動きがしづらいのが難点
- **自発赤外線式**
  - 光学式の発展版
  - 光学式に近いが、各マーカが順番に自発的に発光する
  - ラベリング処理が容易になる



© spice inc.



© Phoenix Tech. Co.

### モーションキャプチャ機器の種類(3)

- **RGB-Dカメラ方式**
  - Microsoft Kinect など
  - 各ピクセルの奥行きを計測
  - 人体部位・姿勢の推定
  - 他の方式より精度は低い
- **マーカレスカメラ方式**
  - 部位の長さが既知であれば、通常のカメラのみでも姿勢推定は可能
  - RGB-Dカメラを使うより、さらに精度は落ちる



© Microsoft

### 動きのデータの作成方法の比較

- **モーションキャプチャ**
  - 人間の動きがそのまま取り込めるので、一見手軽そう
  - 実際には、ノイズや、俳優とキャラクターの骨格の違いなどのため、かなり編集が必要になる
  - そのままではキーフレームがないので編集がしづらい
- **キーフレーム編集**
  - ゼロから作らないといけないので大変
  - キーフレームが設定されているので、修正はしやすい
  - 人間らしい細部の動きや自然さを実現するのは難しい
- **両者の使い分けや組み合わせが必要**

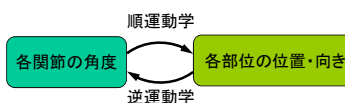
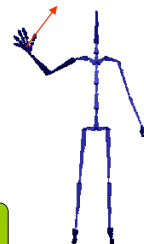
### 人体モデルの動作データの変形

### 動作データの変形処理

- **動作データを編集したり、既存の動作データをもとに新しい動作を生成したりする技術**
  - インバース・キネマティクス
  - モーション・インターポレーション
  - モーション・ブレンディング
  - モーション・リターゲティング
- **これらの手法を利用することで、毎回新しい動作データを作成する手間を省ける**

### キネマティクス(運動学)

- **キネマティクス(運動学)**
  - 多関節体の動きの表現の基礎になる考え方
  - 人間の姿勢は、全関節の関節角度により表現できる
  - 各関節の角度と、各部位の位置・向きとの関係性を計算するため手法





## キネマティクス(運動学)

- フォワード・キネマティクス(順運動学)
  - 多関節体の関節角度から、各部位の位置・向きを計算
  - 回転・移動の変換行列の積により計算
- インバース・キネマティクス(逆運動学)
  - 指定部位の位置・向きから、多関節体の関節角度を計算
    - 手先などの移動・回転量を与えられた時、それを実現するための関節角度の変化を計算
  - 動きを指定する時、関節角度よりも、手先の位置・向きなどを使った方がやりやすい
  - ロボットアームの軌道計画等にも用いられる



## IK計算の例

- インバース・キネマティクス(IK)計算のデモ
  - 後に説明する解析的解法による計算

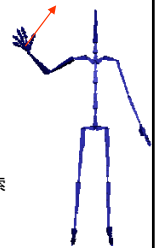


## IK計算における制約条件

- 人体姿勢のIK計算で生じる問題
  - アンダー・コンストレインツ(少な過ぎる制約)
    - ある部位の位置・向きを指定しただけでは姿勢が一意に決まらない
      - 例: 手の位置がきまっても、その手の位置を実現するような全身の関節角度の組は無数に存在する
    - 何らかの評価基準を入れて適当な解を決定
      - 例: 前ステップの姿勢との差がなるべく小さくなる解を選択
  - オーバー・コンストレインツ(多過ぎる制約)
    - 逆に制約が多すぎて、与えられた制約を満足する姿勢がない

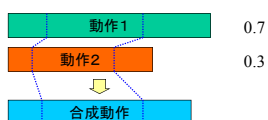
## IKの計算方法

- いくつかの計算方法がある
  - 数値的解法
    - 擬似逆行列を使った計算方法
      - 各関節の微小回転に応じた指定部位の移動・回転量の関係をもとに、条件を満たすように関節を少しずつ回転
    - 非線形計算による計算方法
      - 最適化問題と考えて、指定部位の位置・向きの条件を満たすような関節角度を探索
  - 解析的解法
    - 人間の骨格に注目した効率的な計算方法
    - ※ 詳しい説明は省略(「ロボット工学」等の授業で習うかも)



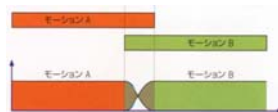
## モーション・インターポレーション

- 動作補間(モーション・インターポレーション)
  - 2つの動作を混ぜ合わせて新しい動作を生成
    - 例: ゆっくり走る動作と早く走る動作から、中くらいの速度で走る動作を生成
    - 任意の比率で混ぜ合わせることができる
  - あらかじめ2つの動作を同期しておく必要がある
    - 同じタイミングにキーフレームが設定されている必要



## モーション・ブレンディング

- 動作合成・遷移(モーション・ブレンディング)
  - あらかじめキャラクタの基本的な動作を作成
  - 基本的な動作をつなげて長い動作を生成
  - 動作と動作の間をなめらかにつなぐ手法
    - あらかじめ、前の動作の終了姿勢と、次の動作の開始姿勢が、ほぼ同じになるよう作成しておく

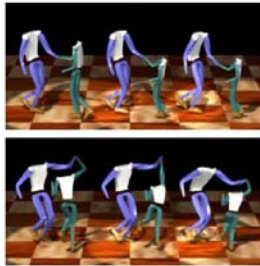


参考書 図4.12

### モーション・リターゲティング

• 骨格の異なるキャラクタ間で動作データを変換

- 例: モーションキャプチャデータを、元の俳優の骨格から、実際のキャラクタの骨格用に修正
- 単純に関節角度をコピーするだけでは、足の位置がずれたりするため、修正する



[Gleicher 98]

### 動力学を考慮した動作の生成・変形

### 動力学を考慮したアニメーション

• 運動学(キネマティクス)

- これまでに紹介してきた手法
- 多関節体の機構を扱うための手法(関節角度と各部位の位置・向きを扱う)
- 現在は、こちらが主に使われている
- 力学的に正しい動きが生成されることは保障されない

• 動力学(ダイナミクス)

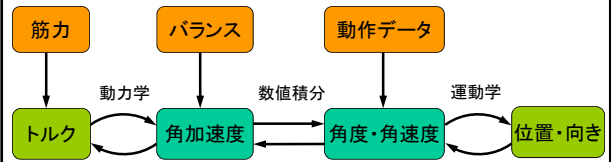
- 運動学に加えて、力学的な影響を考慮する手法

### 動力学を考慮したアニメーション

• 主に2種類の方法がある

- 動力学シミュレーションによるアプローチ
- 動作最適化によるアプローチ

• 多関節体の動力学



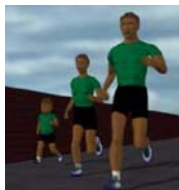
### 動力学シミュレーション

• コントローラ + シミュレーション

- 本物のロボットと同様に、制御理論に基づいて関節のトルクを制御するコントローラを構築
- 動力学シミュレーションを行って動きを生成

• 問題点

- まだ人間らしい動きを生成するのは困難
  - 各動作ごとのコントローラのチューニングが必要
- 結果をうまく制御することが難しい



[Hodgins 97]

### 動作最適化

• 動作最適化

- 動作データの各時刻における関節トルクやバランスの不自然さを評価する関数  $f(\text{動作})$  を設計
- $f(\text{動作})$  が小さくなるように、動作データを少しずつ修正していく
  - 計算に非常に時間がかかる
  - 人間のような多関節体には、自由度が高すぎてそのまま適用するのは困難(自由度を下げる工夫が必要)



[Popovic 98]

## 人体以外の物体のアニメーション

## 髪・衣服のアニメーション

- 髪や衣服などのキャラクタの付属物
  - 自然なキャラクタアニメーションには非常に重要
  - これらの柔らかい物体の動きを手で設定するのは困難
  - 動力学シミュレーションを使って計算する技術が開発されてきた
  - コンピュータゲームなどでは、全身の形状と同様に扱い、骨格に応じて単純に変形することが多い



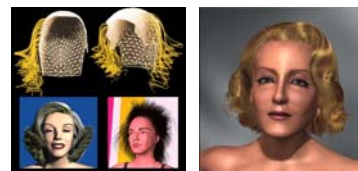
## 衣服のシミュレーション

- 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が主に使われている
  - 衣服を粒子モデルにより表現
  - 粒子に働く弾性力、屈折力、重力、摩擦力、空気抵抗などをモデル化
  - 物理法則に従って粒子の運動を計算 → 衣服の変形を計算



## 髪の表現・シミュレーション

- ポリゴンによる大まかな表現方法
- 一本一本を表現する方法
  - 代表的な髪を与えて、髪形を作成する方法
  - 流れ場を与えて、髪形を生成する方法



[Koh and Huang, CAS 2001]

[MIRA Lab.]

## その他の柔らかい物体の表現

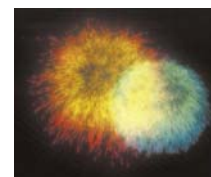
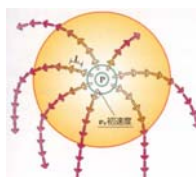
- 有限要素法
  - 物体を細かいメッシュに区切る
  - それぞれのメッシュのエネルギーを数式で定義
  - エネルギーが安定するように変形を計算



参考書 図6.9  
[Terzopoulos 87]

## パーティクル

- 細かい粒子の運動によって、炎などの自然物表現する技術
  - パーティクルの運動、生成・消滅条件を定義
  - 自動的にアニメーションが生成される



参考書 図7.2

## リアルタイム・アニメーション

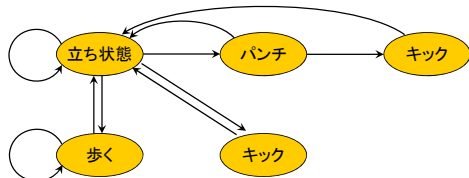
## リアルタイム・アニメーション

- コンピュータゲームが代表例
  - 映画などの動画製作とは異なり、動きのデータを動的に生成する必要がある
- 動的な動きの生成方法
  - 単純な物体の制御(車・飛行機など)
    - キー入力や自動制御にもとづき制御
    - 物理法則も考慮
  - キャラクター(多関節体)の制御
    - 動作グラフによるアニメーション
      - 必要な動作データをあらかじめ作成して用意
      - 入力などに応じて適切な動作データを順番に再生
    - 現在は物理法則はほとんど考慮されていない(不自然)



## 動作グラフ(動作ツリー)

- あらかじめ各動作データ間がつながるように考慮して動作データとそのつながりを作成
  - 動作の終了姿勢と次動作の開始姿勢を同じに
  - ユーザの入力や状況に応じた動作を実行できる



## 動作グラフの自動作成

- 長い動作データを解析して、動作グラフを自動的に作成する技術もある
  - 複雑なグラフも自動的に生成できる半面、計画的に設計されているわけではないので、ゲームなどへの利用は難しい
  - キャラクタをランダムに動かすような用途に有効
    - 群集モデルなど



## リアルタイムアニメーションの問題点

- 現在の主なアプローチ
  - あらかじめ必要な動作データを全て作成しておき、ユーザの操作に応じて動作データを再生
- 問題点
  - 決められた動作の繰り返ししかできない
  - 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
- 状況に応じて動的に動作を作成する技術が必要



## 群集モデル

- 映画の戦闘シーンなど、大量の仮想人間の動きをシミュレートする技術
  - 個々のキャラクターの動きを制作するのは不可能
  - 一定のルールに従って、キャラクターを自律動作
  - キャラクター同士の相互作用によって、それらしい群集の動きが生成される



Lord of the Rings, 2002



Chronicles of NARNIA, 2005



NINETY-NINE NIGHTS, 2005

### 群衆モデルの実現方法

- 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

### 今日の内容のまとめ

- アニメーションの原理
- 人間のアニメーション
  - 人体モデルの表現
  - 人体の動作データの作成
  - 人体の動作データの変形
  - 動力学を考慮した動作データの生成・変形
- 人体以外の物体のアニメーション
- リアルタイム・アニメーション

### 今回の内容

- 前回の復習
- アニメーション
  - アニメーションの原理
  - 人間のアニメーション
  - 人体以外の物体のアニメーション
  - リアルタイム・アニメーション
- 講義のまとめ
- 授業アンケート

### 講義のまとめ

### 講義の達成目標(シラバスより)

- コンピュータグラフィックスの歴史や、どのような応用に利用されているかといった背景知識を理解させる。
- コンピュータグラフィックスを扱う上で必要となる、レンダリング技術や変換行列などの、基礎的な知識を理解させる。
- 3次元グラフィックスを用いた簡単なプログラムを開発できる技術を習得させる。

### 講義の位置づけ

- 情報の選択必修科目
- 学科の学習・教育目標(B)に対応
  - コンピュータ応用とシステム理論を学び、時代の要請に呼応した新たな情報システムを創造し、開発を行うための基礎能力を身に付ける

### 単位の認定方法(確認)

- 期末テスト(40点)
  - 教科書・ノート持込不可
    - 基本的な原理を理解しておけば解ける問題を出題
- 期末レポート(40点)
  - 与えられた課題を作成して提出
- 毎回の講義の演習問題・演習課題(20点)
  - 講義中の演習問題、プログラミング演習課題
- 出席(ICカード+演習問題提出)
  - 成績には考慮しない、一定回数の欠席で不合格
    - 自分の出席回数は、きちんと自分で把握すること

### 期末試験・レポート

- 期末試験
  - 8月4日(火) 3限目 1305講義室
    - 正確な時間・会場は、掲示を確認すること
  - 試験範囲は、講義で扱った内容全て(演習も)
- 期末レポート
  - 8月10日(月) 17:00(厳守)
    - 締め切り以降の提出は一切認めない
  - Moodleから提出
  - 端末室の利用可能期間に注意すること

### レポート課題

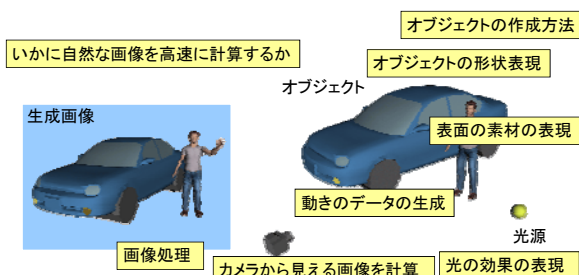
- レポート課題
  1. ポリゴンモデルの描画
  2. 視点操作インターフェースの拡張
  3. アニメーション
  4. テクスチャマッピング
  - 各自、自分に与えられた課題を実現するプログラムを作成すること

### 再試験・再レポート

- 再試験・再レポート提出は一切行わない
- 最初から試験・レポートに全力を尽くすこと
- 特に、試験に自信のない人は、レポートを頑張ること
- 万一、再試験・再レポート等を実施する場合は、Moodleで連絡する

### コンピュータグラフィックスの概要

- コンピュータグラフィックスの主な技術



### 演習内容

- OpenGL + GLUT による演習
  - 講義中に端末を使って演習を行う
  - 簡単な物体描画、マウスによる視点操作、アニメーションなど
  - 具体的な描画処理はOpenGLが行ってくれる
- 演習の目的
  - 実際にプログラムを作成することで3次元CGの仕組みをより深く理解する
  - 将来、3次元CGのプログラミングが必要になる時のために、とりあえず最低限使えるようになる





## CGの社会での応用

- 出版(静止画)・アニメーション(動画)・Web など

- 基本的には、市販のソフトウェアを使用して制作
  - プログラミングは不要(まれに必要)
  - 原理は理解しておく必要がある
- デザイン能力などが必要



Toy Story 2, 1999, © Disney-Pixar

- コンピュータゲーム など

- プログラミングが不可欠
- 基本的には、本講義で学習する知識で制作可能
- 実際にゲームを制作するためには、多数のデータも必要になる



DEAD OR ALIVE 3 © 2002 TECMO, LTD.

## CGはどのような役に立つか

- CGは社会で幅広く使われている

- 基礎的な技術や考え方は、常識として押さえておくべき

- 就職との関連

- 本学科から、CG関係・ゲーム関係の分野に就職する人は、ほとんどいない?
- CG作品制作に携わる職業を目指す人は、デザインなどの能力も必要
- ゲームプログラマなどの職業を目指す人は、トップクラスのプログラミング能力や深い知識、積極性が必要

- ツールとしてCG技術を利用することは多い

- ロボットのシミュレーション結果の表示、各種解析結果の表示、Virtual Reality インターフェースなど

## この講義はどのような役に立つか

- この講義を受講したからと言って、何らかの企業に簡単に就職できるようなことはない
- 本講義では、ごく基礎的な知識・演習を扱う
  - 単にコンピュータグラフィックスの知識だけでなく、情報工学の理解を深める、という意味でも重要
  - 情報工学の常識として、最低限知っておくべき内容
- 自分でより勉強したい人のための出発点
  - 興味がある人は、自分で勉強することが必要

## 卒業研究との関連

- 尾下研究室で卒業研究や大学院研究を行う人は、本講義の内容は最低限の基礎となる
  - コンピュータアニメーション技術の研究
    - キャラクタの動き生成、操作インターフェース、物理シミュレーション、アニメーション制作システム
  - 他の研究室でも、一部の内容は関連する
- 研究では、より高度な内容を扱うことになる



## 講義のまとめ(1)

- 講義
  - 第1回 ガイダンス、グラフィックスのしくみ
  - 第2回 グラフィックスの応用、要素技術
  - 第3回 演習(1): OpenGL&GLUT入門
  - 第4回 モデリング
  - 第5回 レンダリング
  - 第6回 レンダリング・パイプライン
  - 第7回 演習(2): ポリゴンモデルの描画
  - 第8回 座標変換(1)
  - 第9回 演習:座標変換によるアニメーション

## 講義のまとめ(2)

- 講義(続き)
  - 第10回 演習(3):座標変換
  - 第11回 シェーディング
  - 第12回 シェーディング、マッピング
  - 第13回 演習(4):シェーディング、マッピング
  - 第14回 アニメーション
- 演習と講義を交互に行なう
- レポート課題
- 期末試験



### より詳しく勉強したい人へ

- **OpenGLプログラミング**
  - 多数の参考書がある
- **アプリケーション開発**
  - ゲームエンジンの利用 (Unity, Unrealなど)
    - モデルデータ、モーションデータなども入手可能
  - 携帯端末用ゲーム・アプリ開発 (iOS, Android)
- **資格**
  - CGエンジニア検定 (エキスパート・ベーシック)
    - 授業の参考書が公式テキスト



### ゲーム・アニメ制作の勉強方法

- **アニメーション制作の独習**
  - 市販のソフトウェアが利用可能
    - Maya, Softimage, 3ds max 等
    - 教育目的であれば無償利用可能
- **ゲーム開発の独習**
  - ゲーム開発環境が利用可能
    - Unity, Unreal 等
    - 自分の開発したゲームを販売することもできる
  - プログラミング言語の習得



3ds max © Autodesk



Unity © Unity

### 大学院授業

- **コンピュータグラフィックス特論 II**
  - コンピュータグラフィックス技術を用いるプログラム開発で必要になる要素技術を学ぶ
    - 視点操作、幾何形状データの読み込み
    - 影の描画 (高度な描画技術)
    - キーフレームアニメーション
    - 物理シミュレーション、衝突判定
    - キャラクターアニメーション



### 演習問題・授業アンケート

- **演習問題・授業アンケート (選択式)**
  - いつものマークシートに記入、回収
- **授業アンケート (選択式・記述式) (学部共通)**
  - 他の科目と一緒に、Webから入力