



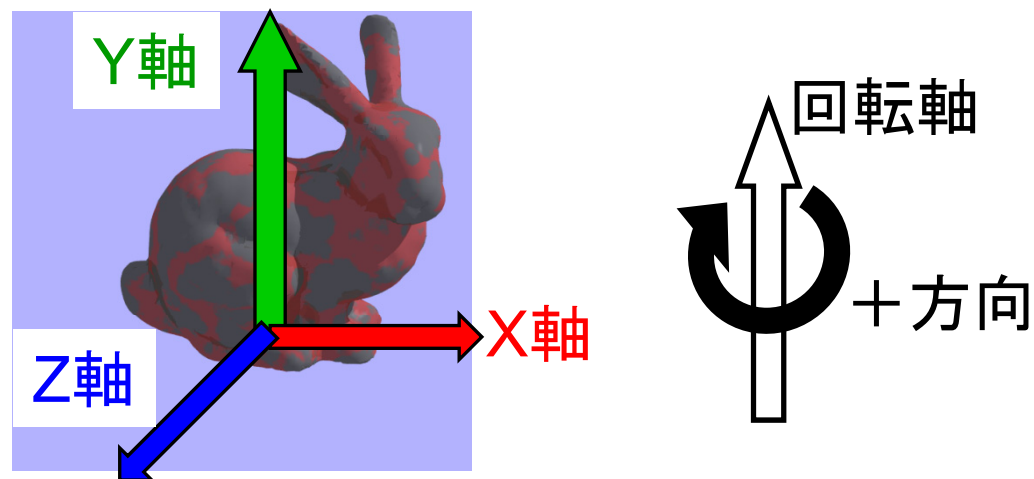
3次元回転を極める ～ 1歩進んだキャラクターアニメーションプログラミング ～

向井 智彦

株式会社スクウェア・エニックス 技術開発部

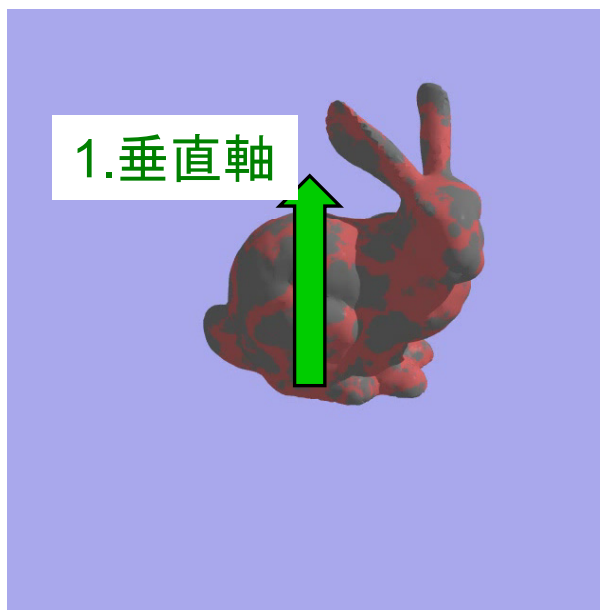
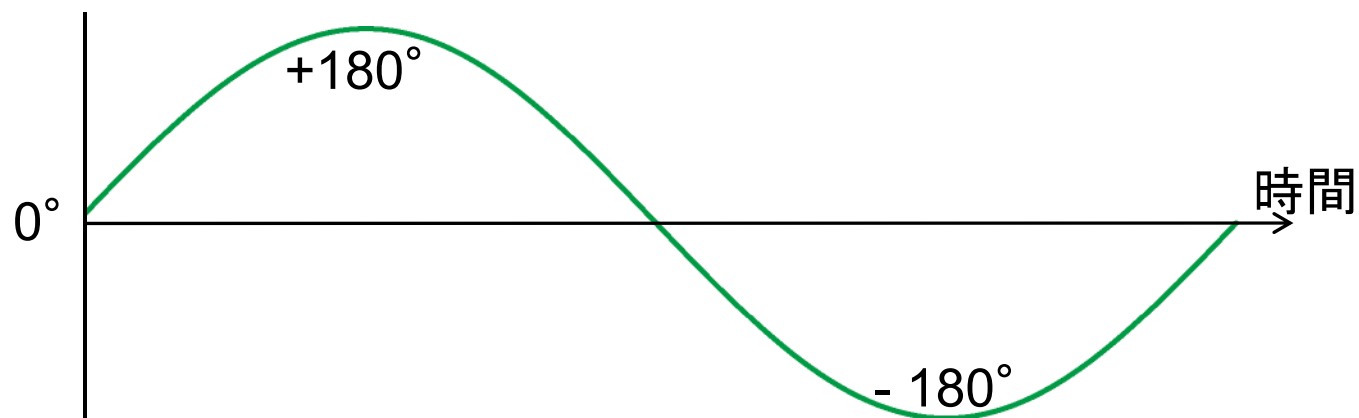
Introduction

はじめに

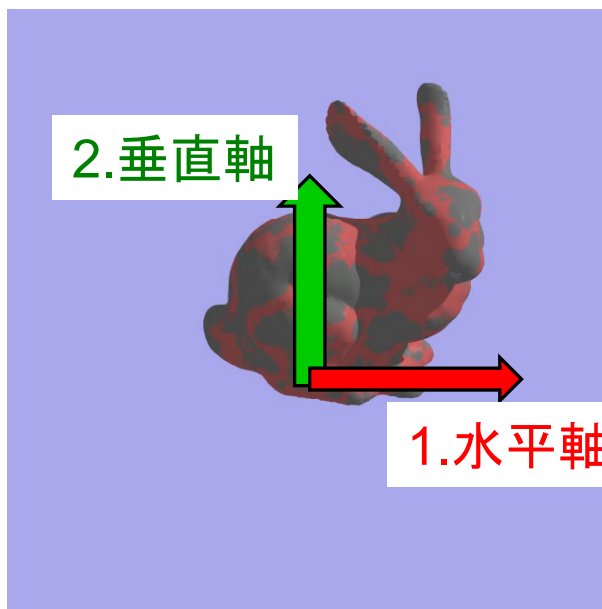
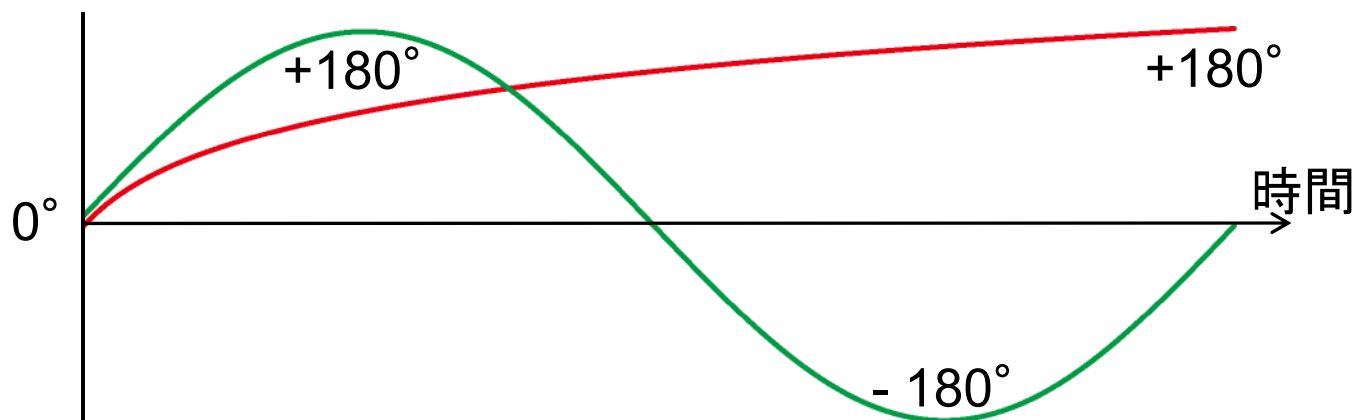


- うさぎが回転するアニメーションの
回転角度カーブ (F-curve) をご覧いただきます
- どんな回転か、できるだけ早く想像して下さい

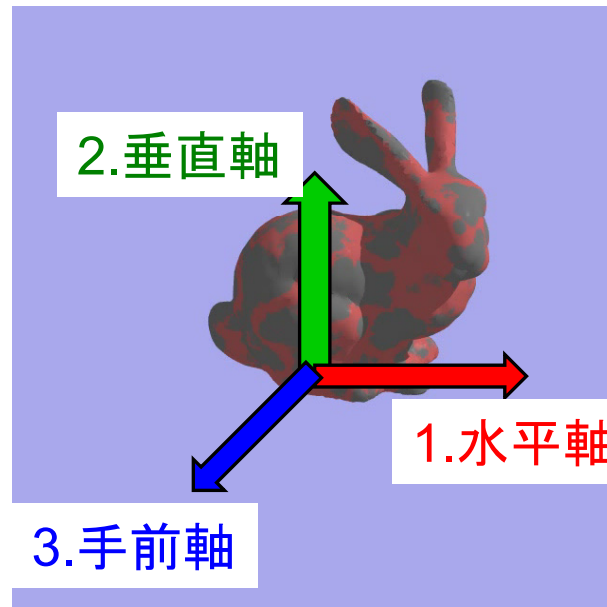
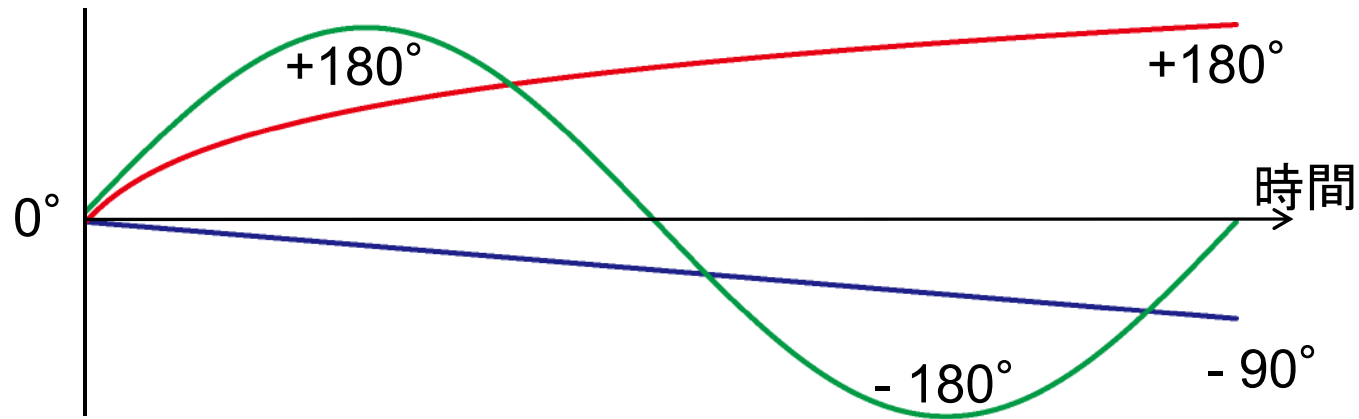
イントロ 1 - どんなアニメーション? その1



イントロ 1 - どんなアニメーション? その2



イントロ1 - どんなアニメーション? その3



- オイラー角は**本当に**直観的？
 - それぞれの値は単純
 - でも複数の回転軸の組み合わせは？
- 3Dプレビューがあったら？
 - 併用すれば, オイラー角はとても使いやすい
- ただし, 内部形式までも統一する理由はない
 - 計算処理しやすい表現を使うべき

- 行列, クォータニオン(四元数), オイラー角, Axis-angle(任意軸周り回転), Expmap など
- どうやら, 万能な表現は無さそう
 - 3次元回転 $SO(3)$ は奥が深い
 - ベクトルのように扱うのは無謀
- 場面に応じた とりあえず の定石をご紹介

- D3DXQuaternionLn
- D3DXQuaternionExp
 - 2つとも XNA framework には非実装
- ゲーム3D数学(O'Reilly) 9.4.11節
 - “...直接使われることは滅多にないものの...”

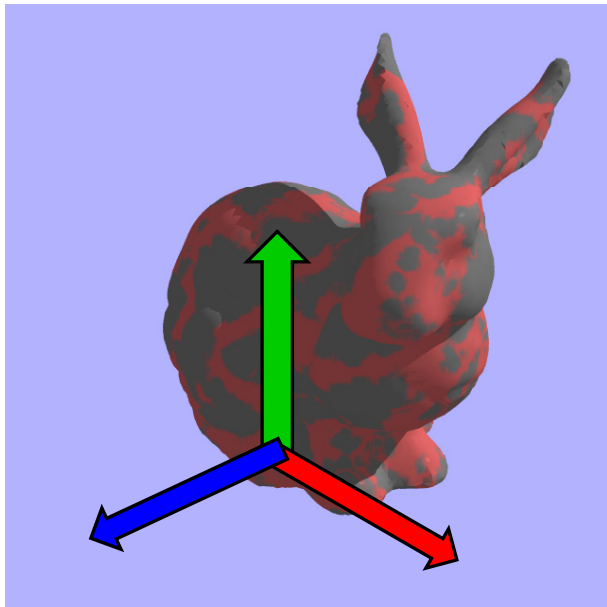
これを使わないなんて とんでもない！

- 3次元回転のデータ形式
 - 「方向」と「回転」の区別
 - 一般的なデータ形式のおさらい（既知とします）
 - Exponential map - クォータニオンの対数と指数
 - 開発の側面から見た特性と比較
- キャラクタアニメーションプログラミング
 - FK, IK, Mocapデータ処理
- まとめ

Orientation and rotation

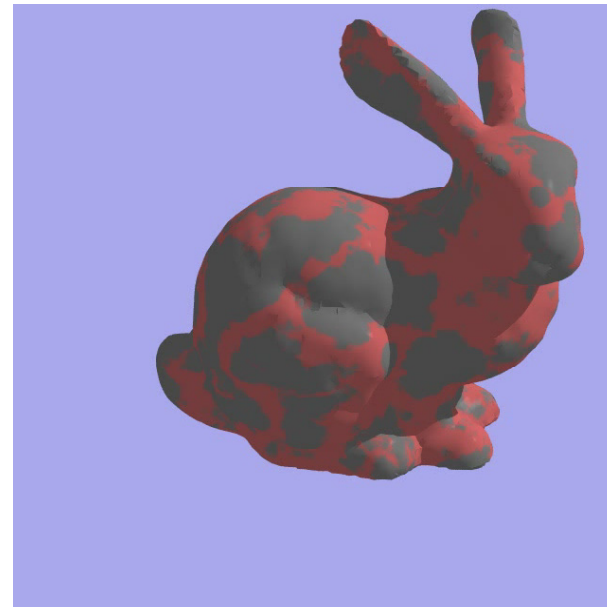
方向と回転

方向 (Orientation)



- 物体の姿勢
 - 値の範囲を定義
 - $0^{\circ} \sim 360^{\circ}$, $\pm 180^{\circ}$

回転 (Rotation)



- 物体の回転運動
 - 無限の値
- 方向の差分

方向 : 回転

≡

位置 : 移動ベクトル

≡

姿勢 : 運動

- 演算の定義

- 方向 + 回転 = 方向
- 回転 + 回転 = 回転
- 方向 + 方向 = 未定義

- 数値の範囲

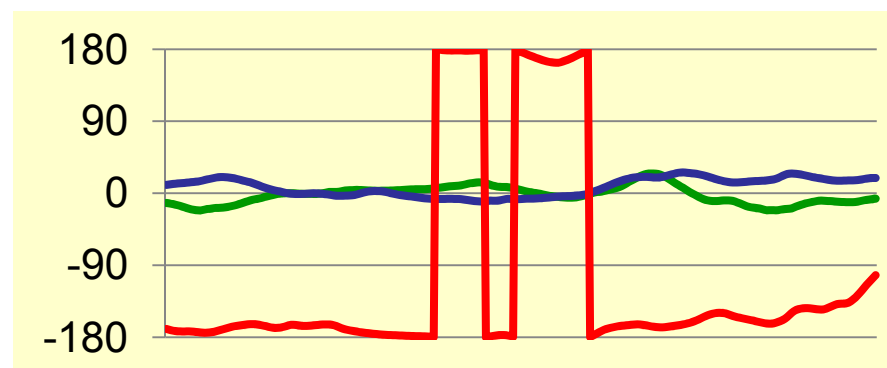
- 方向 : 一定の範囲に制限
- 回転 : 無限

Representation of 3D orientation and rotation

3次元回転のデータ形式

- 方向の特異状態

- 同一の方向を表す無数の値
- 例) $+180^\circ$ と -180°

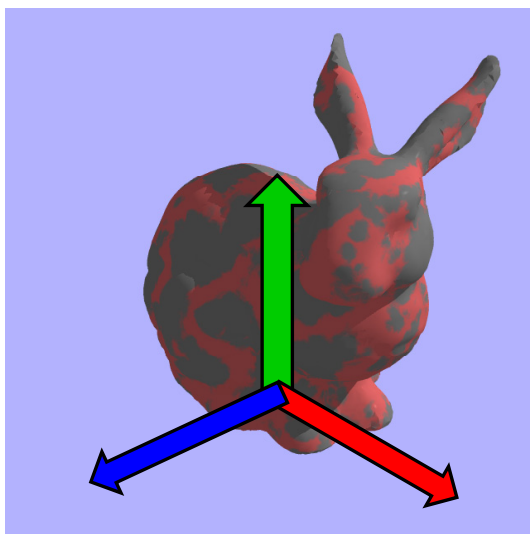


- 回転の特異状態

- 異なる回転量を表す単一の値
- 例) 180° , 540° ...

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

方向 (Orientation)

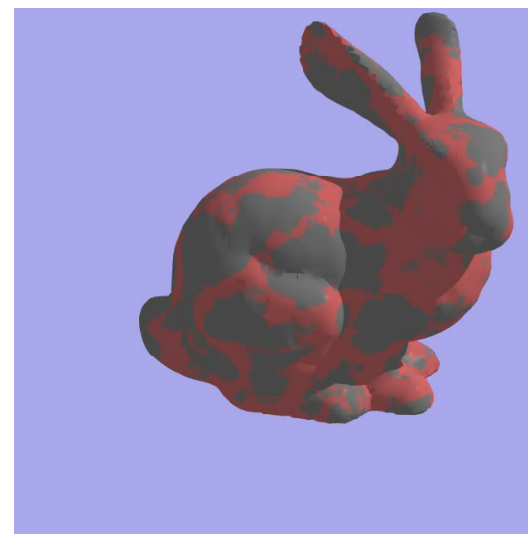


1. 回転行列
2. クォータニオン

方向

回転

回転 (Rotation)



3. オイラー角
4. Axis-angle
5. Expmap

回転

方向

- 正規直交行列(3行3列)

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

- 長所: 2回転合成, ベクトル回転, HW処理
- 短所: 補間, 要正規化

- 3つの虚数 + 1つの実数, 長さ = 1.0

$$\mathbf{q} = [q_X \quad q_Y \quad q_Z \quad q_W]$$

– 得体の知れない名前 & 数値...

- 長所
 - 補間 (SLERP), 2回転合成, ベクトル回転
- 短所
 - 要正規化

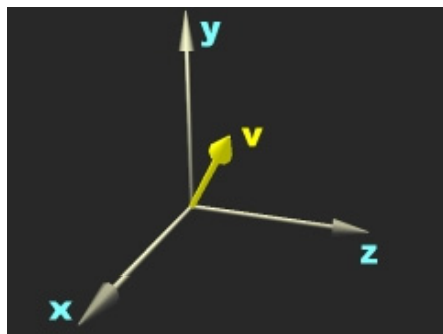
- 座標軸周りの回転角 → 3つ組のスカラー値
- 例: X軸周り → Y軸周り → Z軸周り

$$\mathbf{M}_X(r_X) \cdot \mathbf{M}_Y(r_Y) \cdot \mathbf{M}_Z(r_Z)$$

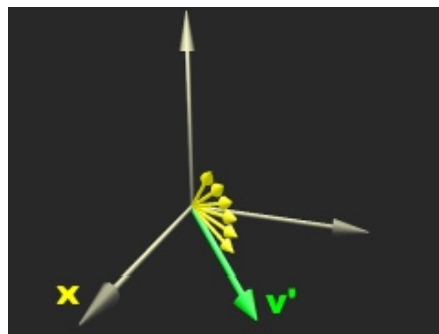
X軸の回転行列

Y軸の回転行列

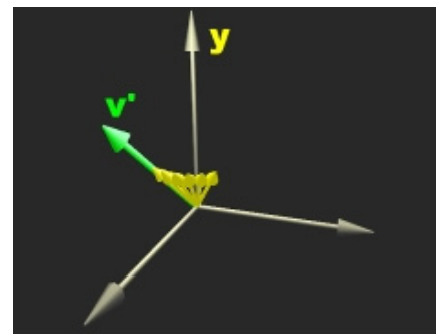
Z軸の回転行列



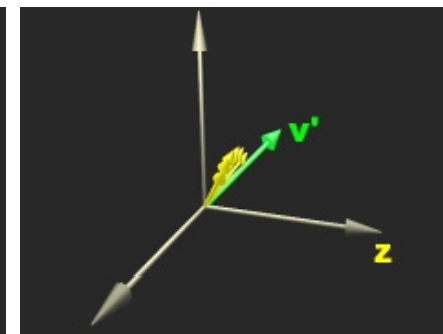
元ベクトル



X軸回転



Y軸回転



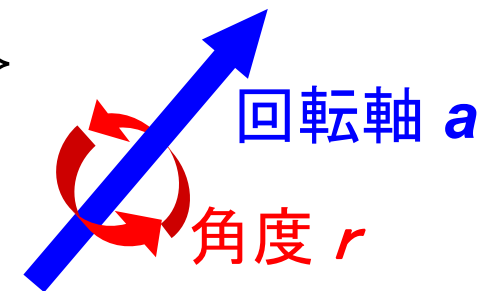
Z軸回転



- 長所
 - 直観的な数値, 最少データ
- 短所
 - 複数の合成パターン(XYZ, YXZ, ZXY, ..)
 - 数値の単位 (rad or deg)
 - ジンバルロック ...致命的！
 - 2回転の合成, 補間
 - ベクトル回転

- 回転軸を表す単位ベクトル + 回転角度

$$\{ [a_x \quad a_y \quad a_z] \quad r \}$$



- 長所
 - 直観的, クォータニオンとの相互変換
- 短所
 - 数値の単位 (rad or deg), ベクトル回転
 - 2回転合成, 補間

$$\log(\mathbf{q}) = [e_X \quad e_Y \quad e_Z]$$

↑
クォータニオン

(仮名: 対数クォータニオン)

クォータニオンへの変換式

$$\mathbf{q} = \exp [e_X \quad e_Y \quad e_Z]$$

$$= \begin{bmatrix} \frac{e_X}{\theta} \sin \theta & \frac{e_Y}{\theta} \sin \theta & \frac{e_Z}{\theta} \sin \theta & \cos \theta \end{bmatrix}$$

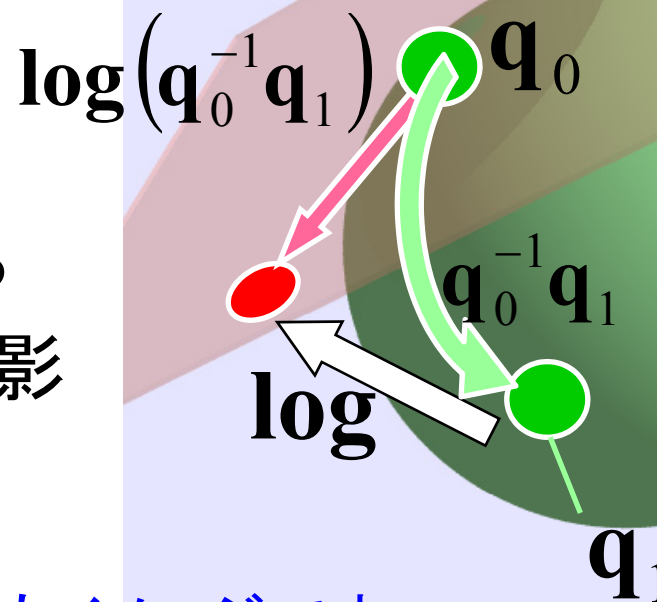
$$\theta = \sqrt{e_X^2 + e_Y^2 + e_Z^2}$$

$\log(\text{回転})$

球面上の回転移動の
2次元平面への投影

3次元回転の場合...

クォータニオン(4D)から
3次元線形空間への投影



※あくまでもイメージです

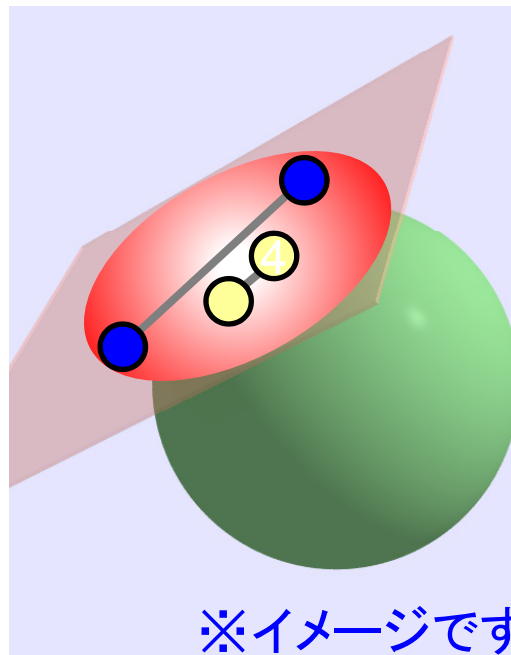
- 加算による回転合成の近似

$$\mathbf{q}_0 \mathbf{q}_1 \approx \exp \{ \log(\mathbf{q}_0) + \log(\mathbf{q}_1) \}$$

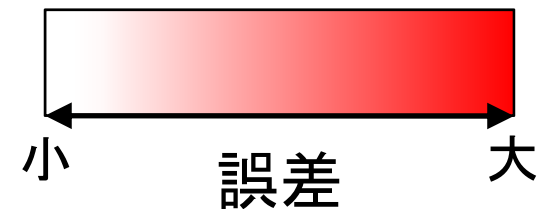
クォータニオンの乗算

Expmapの加算

- 合成の誤差



※イメージです





- 長所
 - 最少データ, 加算による合成と補間 (※近似値)
- 短所
 - ベクトル回転
- オイラー角に対する優位性
 - ジンバルロック無し, 一意な表現
 - 2回転の合成と補間
 - クォータニオンとの相互変換

Comparative evaluation

開発の側面から見た特性と比較

回転行列	方向 回転
クォータニオン	方向 回転
オイラー角	回転, 角速度, 角加速度
Axis-angle	回転, 角速度, 角加速度
Expmap	回転, 角速度, 角加速度

- クォータニオンと回転行列は等価
 - 方向を記述
 - 半周以内の回転
- オイラー角とExpmapは類似
 - スケーリング可能
 - 無限の回転

回転行列	9
クォータニオン	4
オイラー角	3
Axis-angle	4
Expmap	3

- オイラー角とExpmapが最小
 - アニメーションデータのファイル形式に適切
- 行列はサイズ大
 - 他形式の2倍以上

回転行列	乗算 (27回) ※並列化は容易
クォータニオン	乗算 (16回)
オイラー角	×
Axis-angle	×
Expmap	加算 (3回) ※

- オイラー角やAxis-Angleの合成は、他形式を經由
- 回転行列やクォータニオンは乗算で合成
 - 乗算順序で結果が変化
- Expmapのみ加算で合成
 - 合成順序に依存しない
 - ただし、近似値

回転行列	乗算 (9回) ※並列化は容易
クォータニオン	乗算 (24回)
オイラー角	×
Axis-angle	×
Expmap	×

- 回転行列が効率的
- クォータニオンによる直接回転も可能

2つの方向の線形補間



回転行列	×
クォータニオン	SLERP (球面線形補間)
オイラー角	×
Axis-angle	×
Expmap	加算(3回)

- SLERPのみ正しい結果
- Expmapは近似計算

ブレンドイング $\Sigma(\text{ウェイト}) \times (\text{方向})$



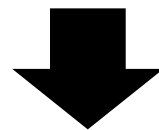
回転行列	△
クォータニオン	△
オイラー角	×
Axis-angle	×
Expmap	△

組み合わせることで
高精度化

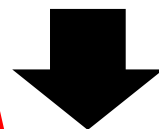
- (理論的に)正しい計算はほぼ不可能
- 粗い近似値は加重平均 + 正規化で計算可
- クォータニオンと Expmapの組み合わせで精度向上

$$w_1 \mathbf{q}_1 \oplus w_2 \mathbf{q}_2 \oplus w_3 \mathbf{q}_3$$
$$w_1 + w_2 + w_3 = 1.0$$

⊕ 方向の合成



$$\mathbf{q}_1 \oplus (1 - w_1)(\mathbf{q}_2 \ominus \mathbf{q}_1) \oplus (1 - w_1 - w_2)(\mathbf{q}_3 \ominus \mathbf{q}_2)$$



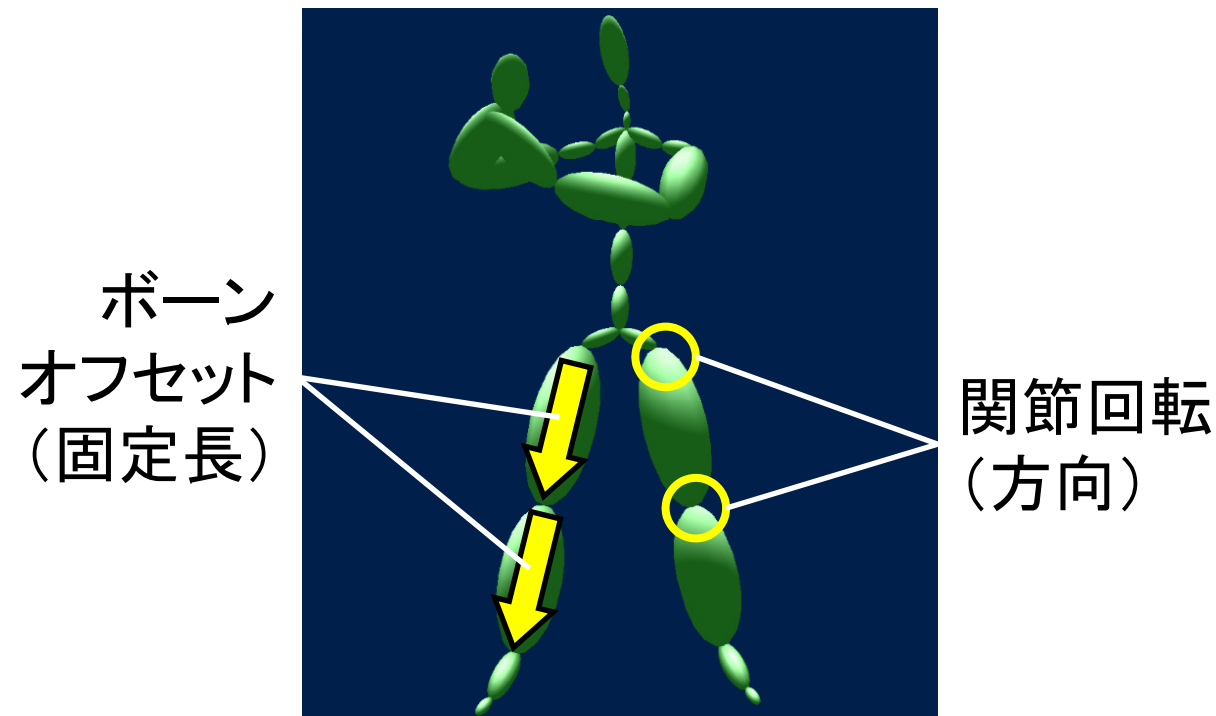
$$\mathbf{q}_1 \exp \left\{ a_1 \log(\mathbf{q}_1^{-1} \mathbf{q}_2) + a_2 \log(\mathbf{q}_3^{-1} \mathbf{q}_2) \right\}$$

- 万能な回転表現は存在しない
 - 方向と回転の区別
 - クォータニオンとExpmapが有力
- 使い分けの一提案
 - 数値UIにはオイラー角
 - 外部メモリ入出力にはExpmap
 - 少メモリ, クォータニオンとの一意な変換
 - 各種計算アルゴリズムに応じた選択
 - 場面に応じてオイラー角, Axis-angleも活用

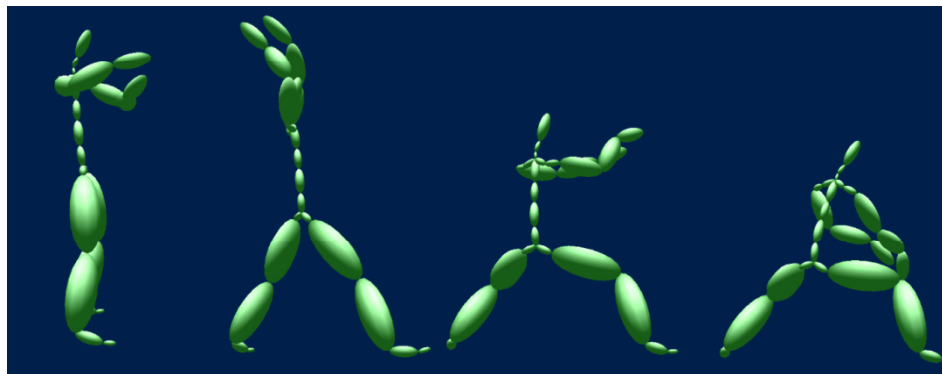
(Slightly) Advanced character animation programming

一歩進んだ キャラクタアニメーション プログラミング

- キャラクタのボーンは剛体モデルと仮定
 - ボーンオフセット＋回転のみ，スケーリング無し

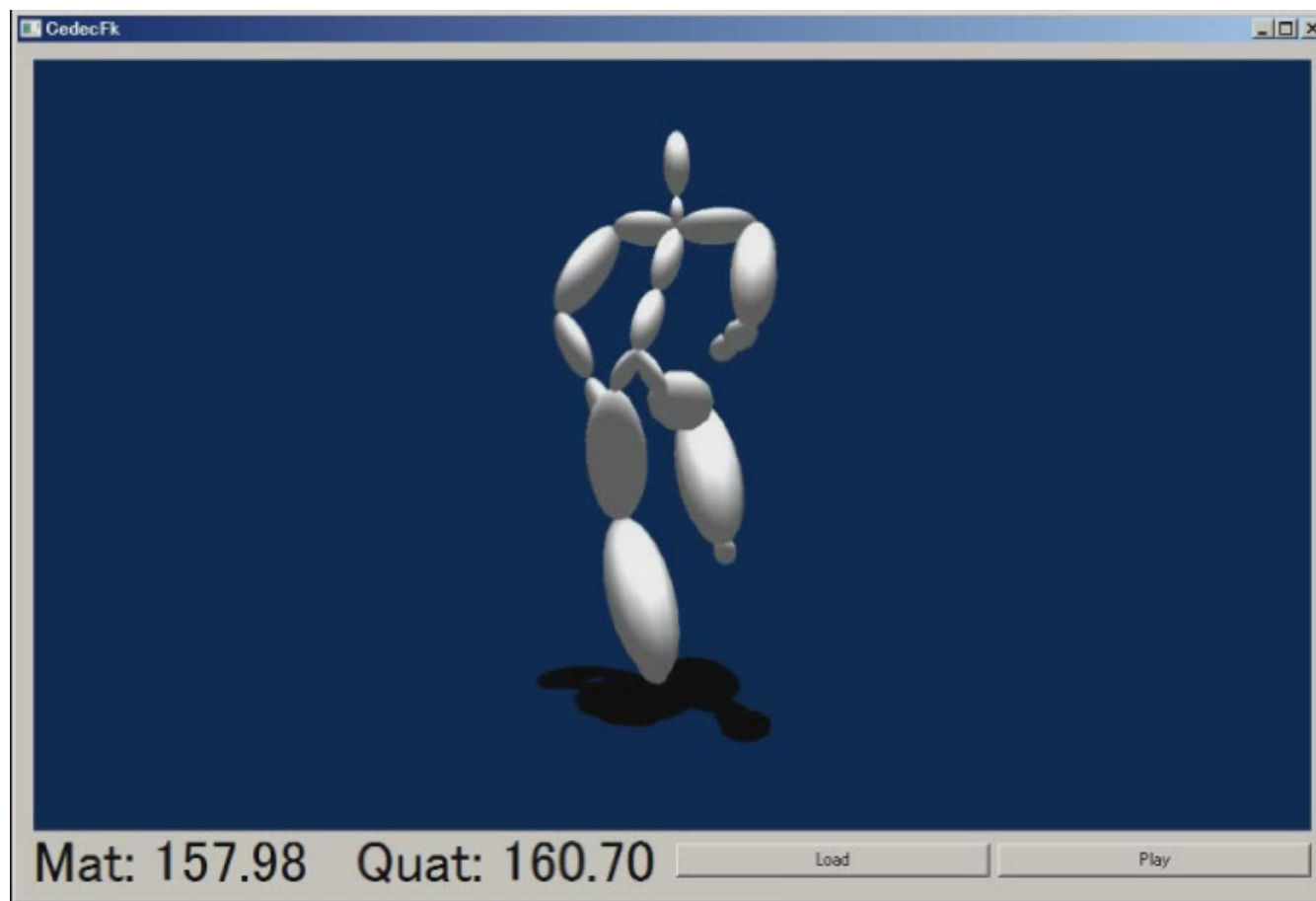


- 関節回転→ボーン的位置, 方向
– モーションキャプチャデータの再生



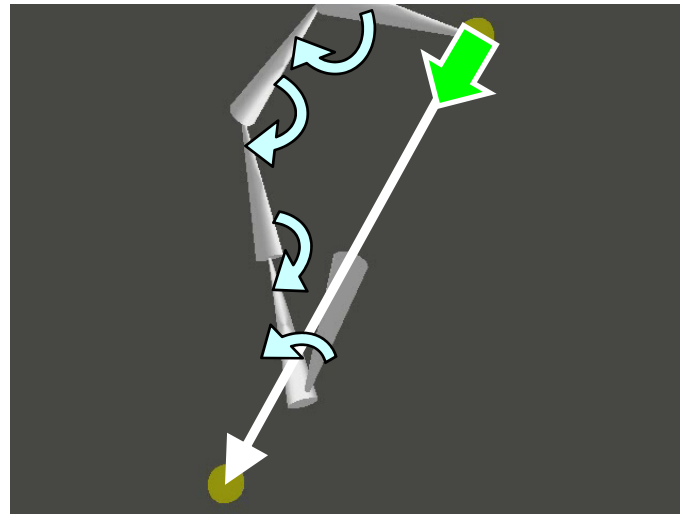
- 関節の方向, ボーン姿勢の表現形式
 1. D3DXMATRIX (16要素)
 2. D3DXVECTOR3 + D3DXQUATERNION (7要素)

FK 速度比較 デモ

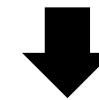


[単位 μsec / 20 frames]

- オイラー角とExpmapの計算精度比較



先端移動速度

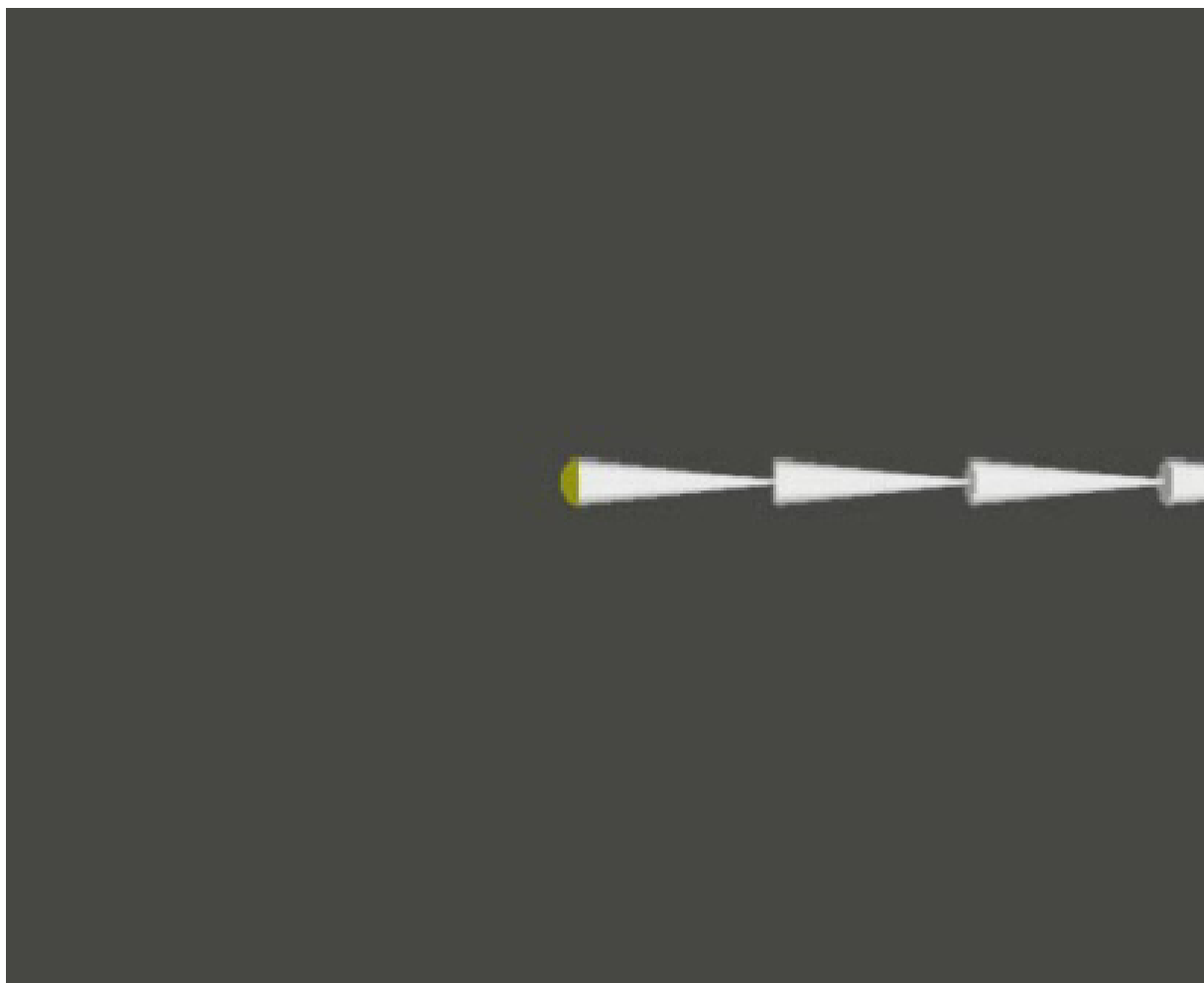


関節角速度

$$\begin{array}{|c|} \hline \text{姿勢の} \\ \text{微小変化} \\ \hline \end{array} = \begin{array}{|c|} \hline \frac{\partial \text{ エフェクタの移動}}{\partial \text{ 姿勢の変化}} \\ \hline \end{array}^{-1} \times \begin{array}{|c|} \hline \text{エフェクタの} \\ \text{微小移動} \\ \hline \end{array}$$

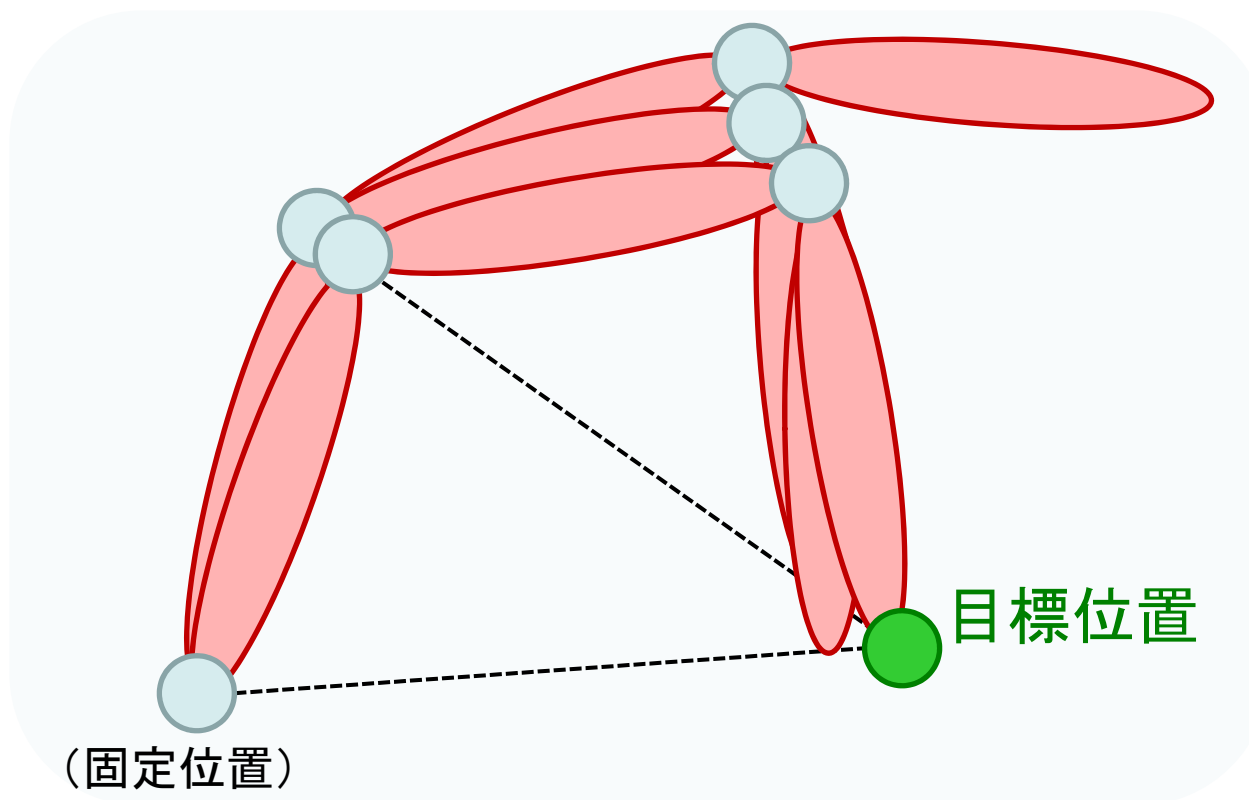
赤字の部分にオイラー角 もしくは Expmap を利用

ヤコビ IK 精度比較 デモ

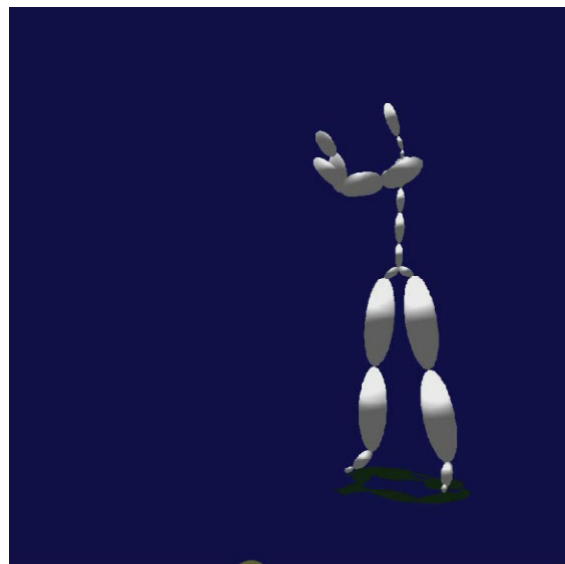


白 : Expmap
赤 : オイラー角

- Axis-angleで最適化, クォータニオンへ変換

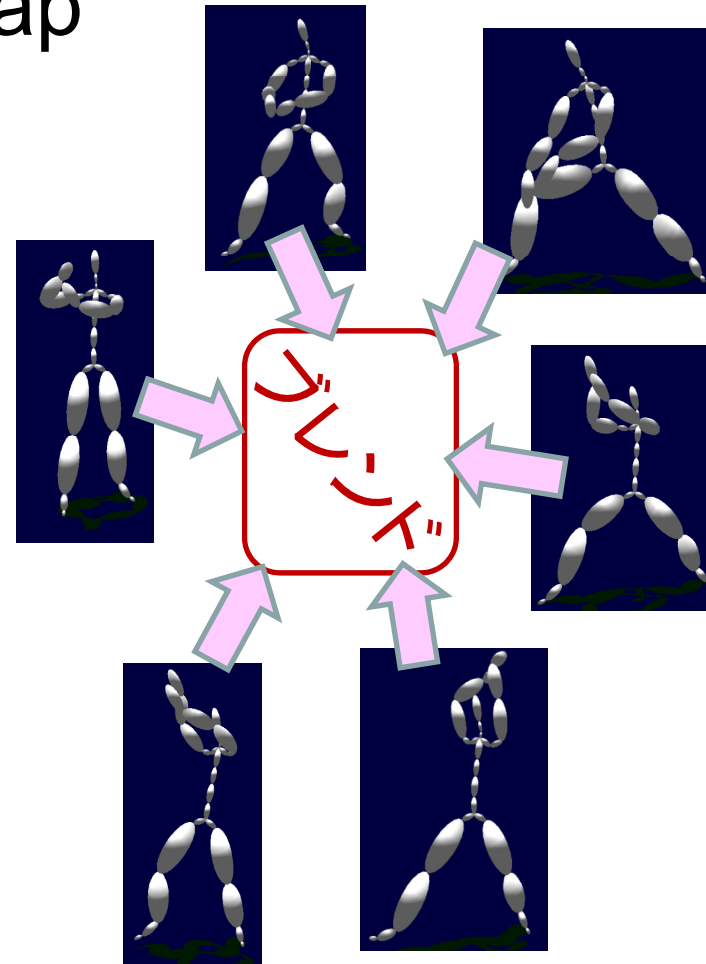


- クォータニオン + Expmap

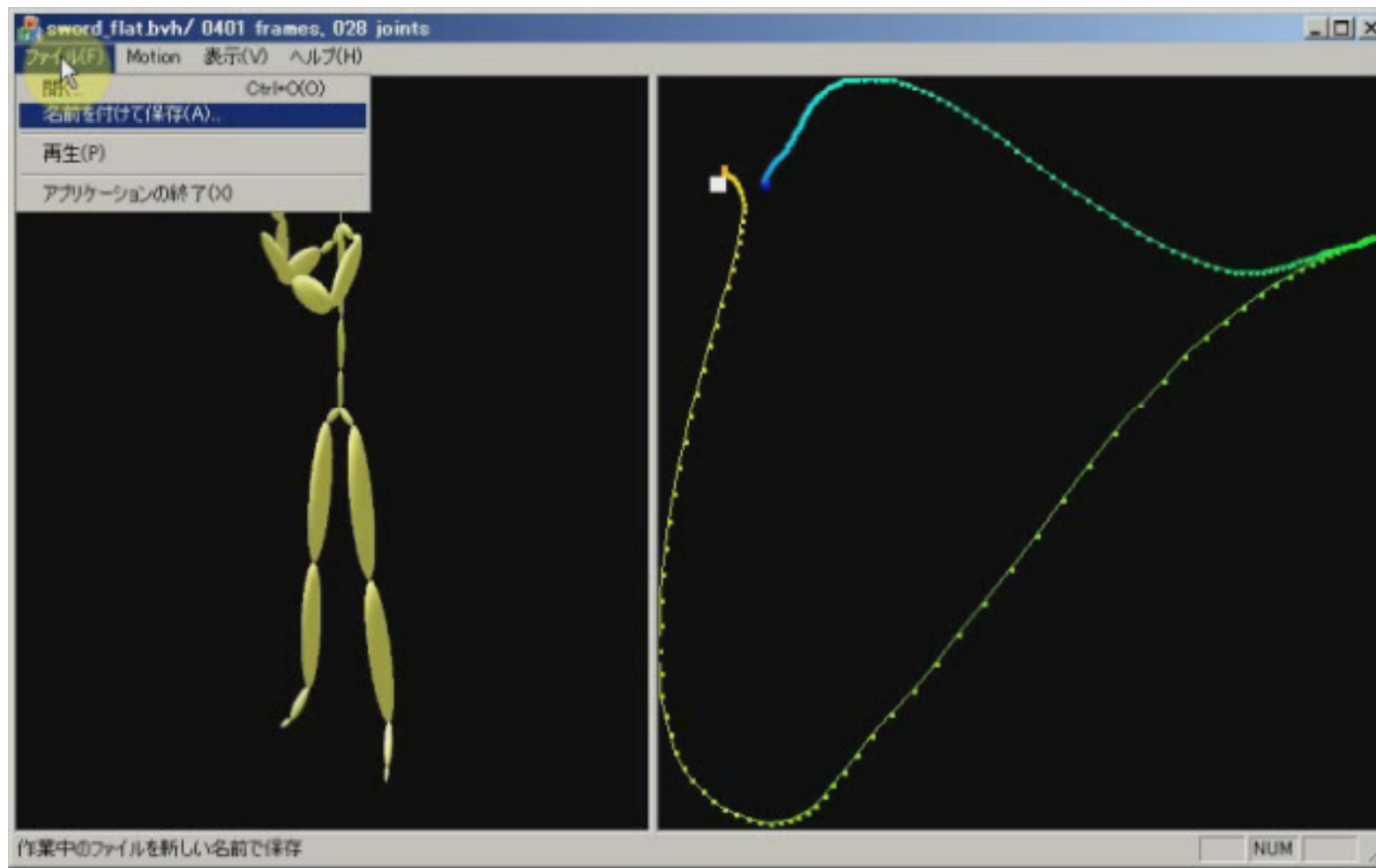


アニメーションシーケンス

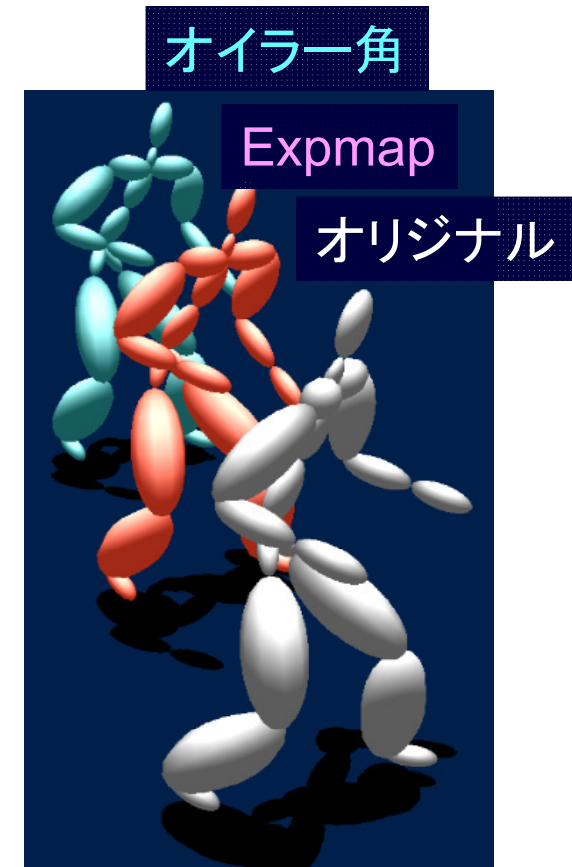
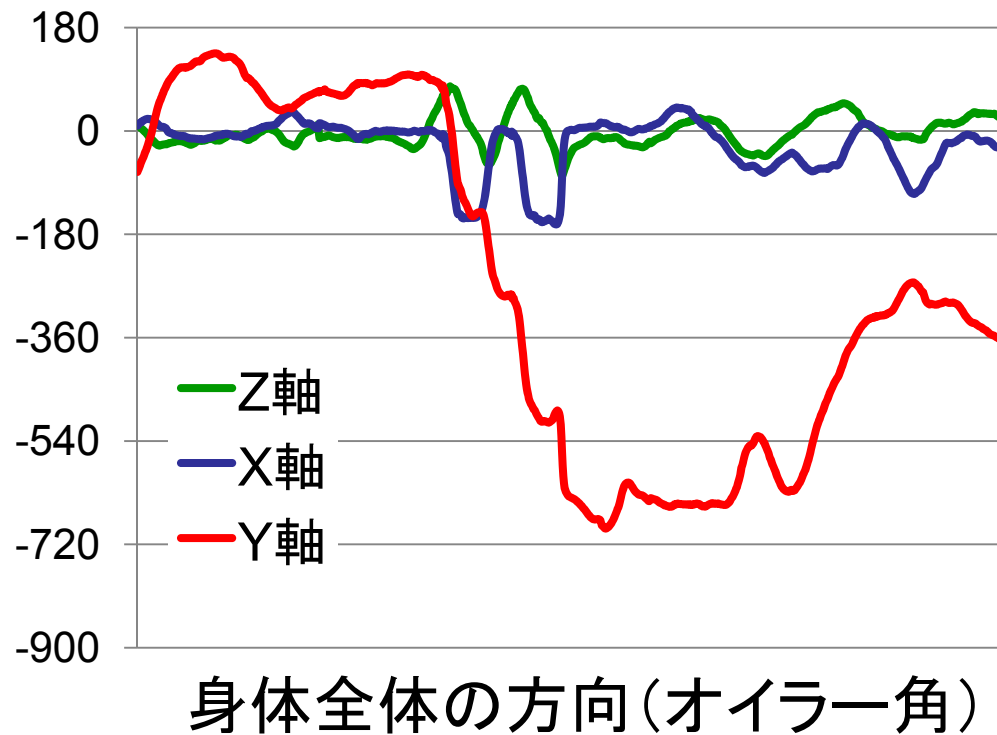
2次元配置



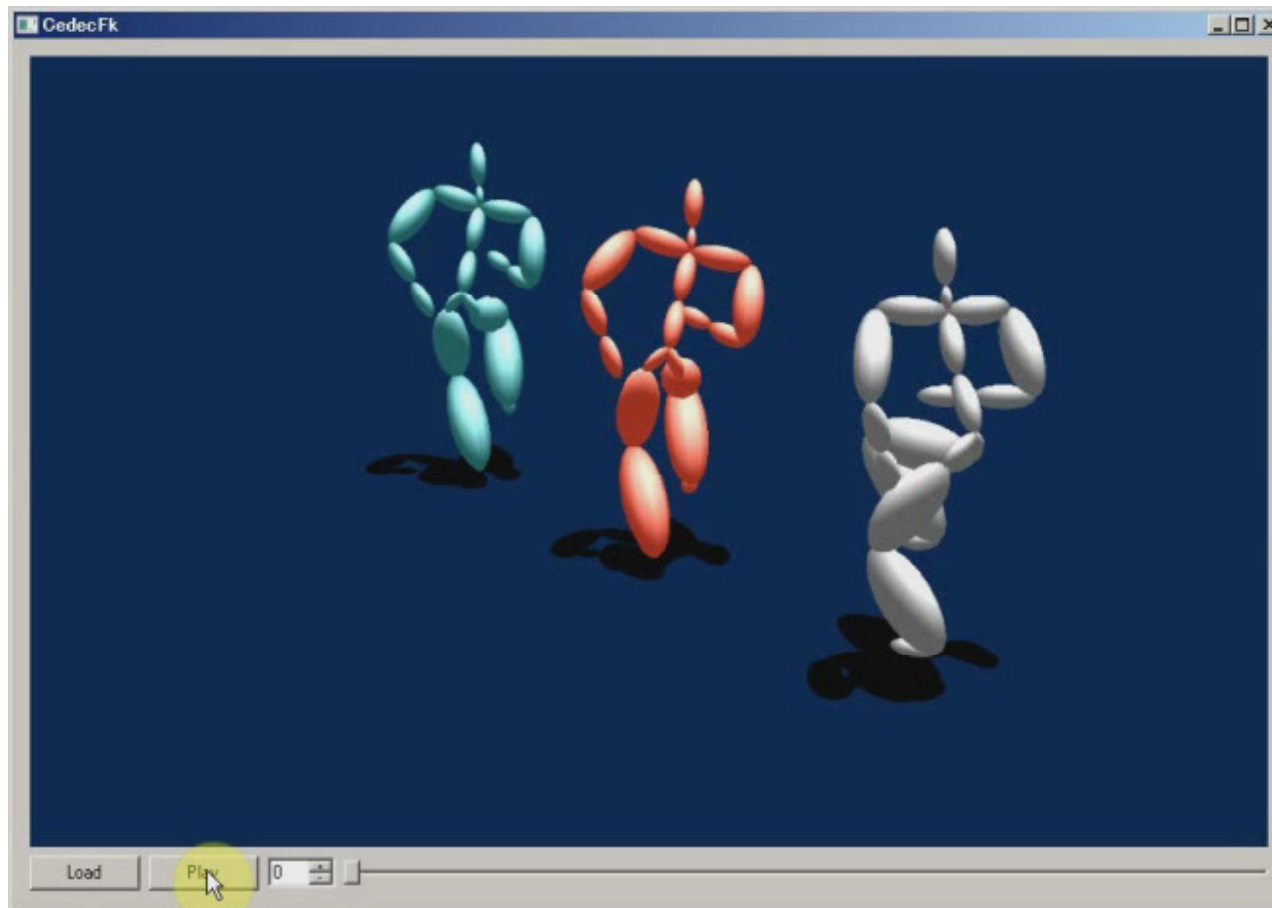
ブレンドイング デモ



- 平滑化フィルタ
– オイラー角とExpmapの比較



モーション平滑化 デモ

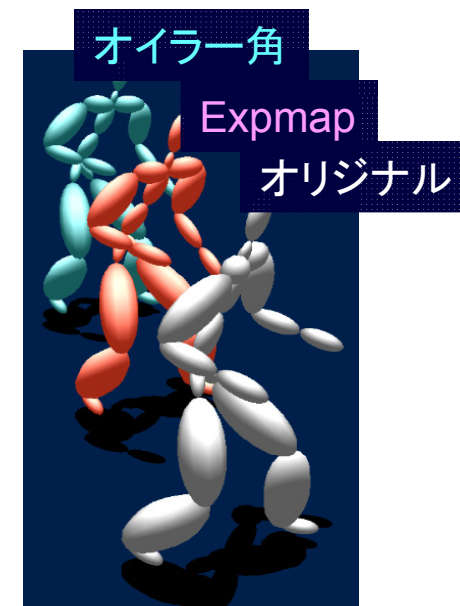
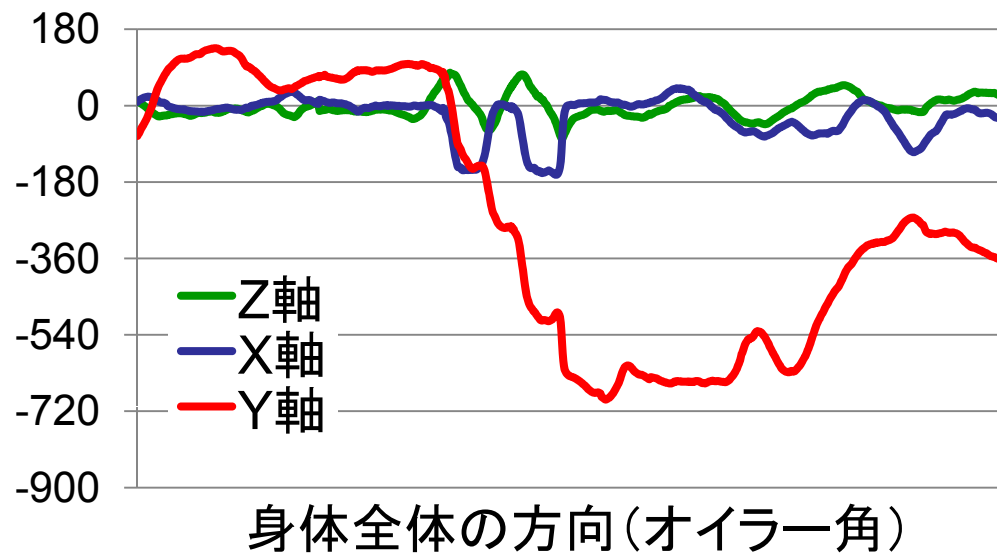
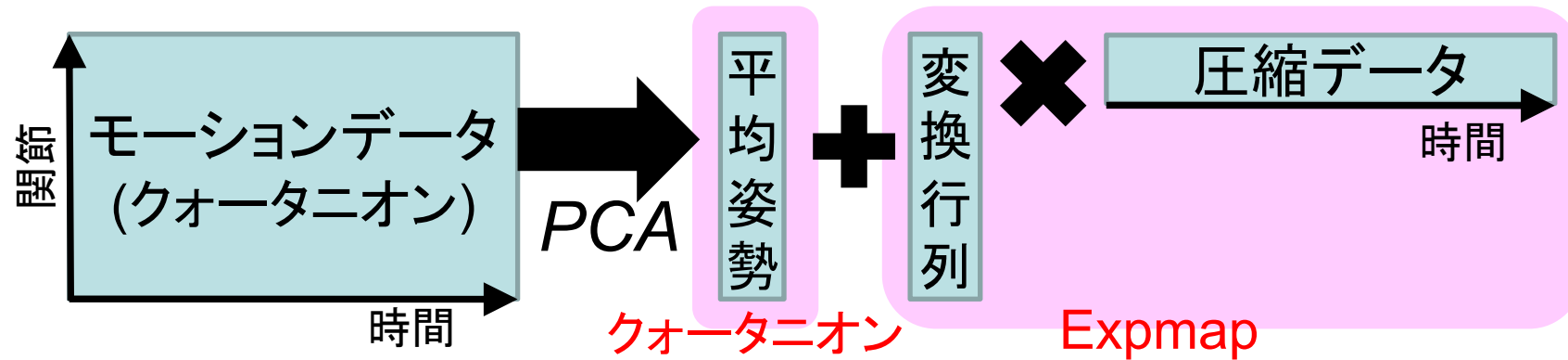


オイラー角

Expmap

オリジナル

統計処理 - 主成分分析によるモーション圧縮



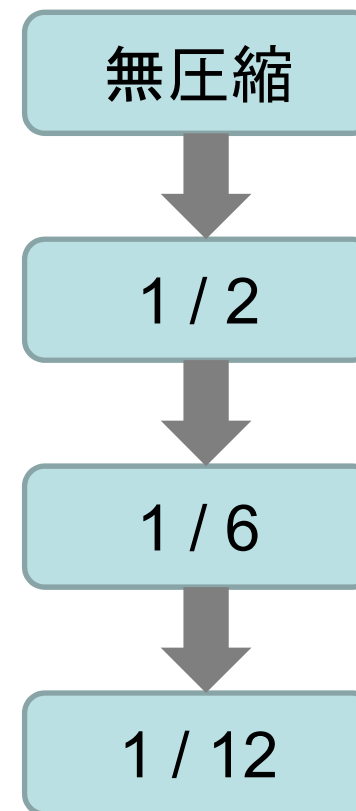
モーション圧縮 デモ



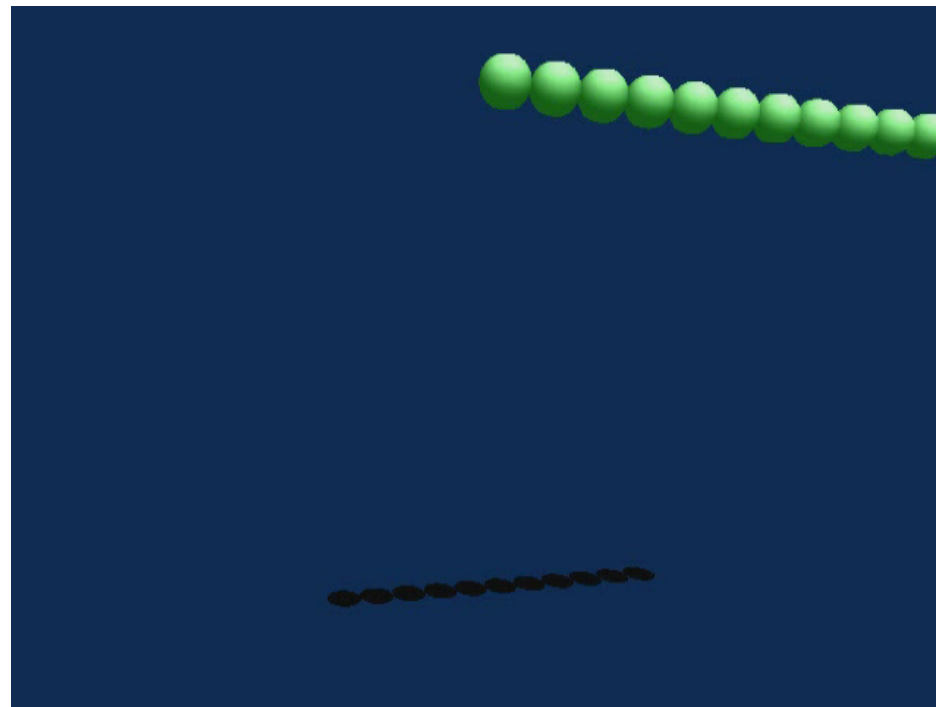
オイラー角

Expmap

オリジナル



- オイラー・ラグランジュ運動方程式
 - クォータニオン(姿勢)
 - Expmap(角速度, 角加速度)



Conclusions

まとめ

- 万能な回転表現は存在しない
 - 場面に応じた使い分けと妥協
- Expmap
 - 回転, 角速度, 角加速度に適した表現
- アニメーションプログラムへの応用
 - 行列, Axis-angle, オイラー角が優位な場面も

- Linear Combination of Transformations
[SIGGRAPH 2000]
- Errors and Omissions in Marc Alexa's
“Linear Combination of Transformations”

- 実例で学ぶゲーム3D数学 (O'Reilly)
- Representing Rotations and Orientations in Geometric Computing
 - Jehee Lee, IEEE CG&A, 2008
- CMU Mocap database
 - ダンスモーションデータ
- Stanford Graphics Lab.
 - ウサギモデルデータ