

KinectV2 を用いたモーションファイル生成

Motion file generation with KinectV2

○志田陽平¹, 吉川浩², 山口健²*Yohei Shida¹, Hiroshi Yoshikawa², Takeshi Yamaguchi²

Abstract: Using dedicated software for generation of the motion file, it is required technical and time. Therefore, using KinectV2, and generate a motion file from the action of the body. In this study, we have developed a motion capture system for the purpose of reducing the technical and time costs in motion file production.

1. まえがき

モーションの生成には専用のソフトを用い、キーフレームを打ち込む等の技術と時間が必要となる。そこで、KinectV2(以下 Kinect)を用い、体の動きからモーションを生成する。本研究ではモーション製作における技術及び時間のコストを軽減することを目的にモーションキャプチャシステムを開発する。

2. 原理・方法

2.1. システム概要

Figure 1 にモーションキャプチャシステムの流れを示す。本研究で使用する Kinect は、カラーカメラや深度センサー、マイクロフォンを搭載したジェスチャーや音声で入力できるデバイスである。また、Microsoft から提供されている Kinect SDK を用いることで、全身 23 個のジョイント(関節)情報を得ることができる。本研究では、連続したアクターのポーズを Kinect から得られるジョイント情報として取得し、データを保持する。キャプチャ終了時に、保持されたジョイント情報データ群に対し変換処理を行うことでモーションファイルとして出力する。モーションファイルのフォーマットとして BVH フォーマットを用いる。BVH フォーマットは Biovision 社が提唱したモーション用ファイルフォーマットで様々な商用アプリケーションでサポートされている。

2.2. アプリケーション概要

Kinect の認識しているポーズをリアルタイムに確認が行えない場合、正しく認識できているか判断ができない。そのため、モーションキャプチャと同時に Kinect からのジョイント情報をリアルタイム 3 次元ビューで表示することで、現在のポーズの認識状態を確認できるよアプリケーションを作成する。

2.3. モーションについて

モーションとは 3D モデルデータに定義されている頂点の位置を動的に変更することであたかも 3D モデルの一部が動いているかのように見せることである。モーション可能なモデルデータは、ジョイント情報を

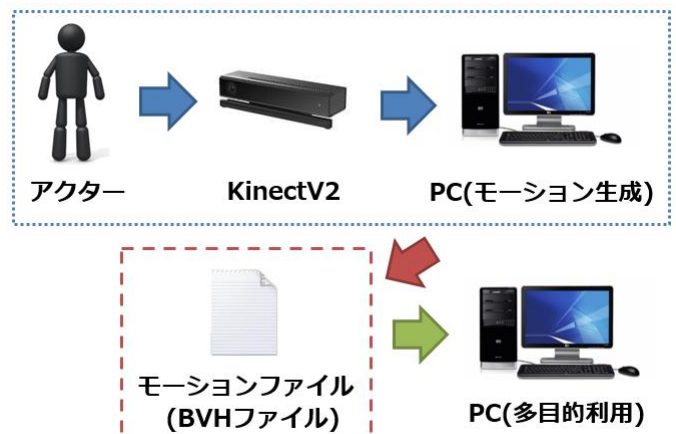


Figure 1. System overview

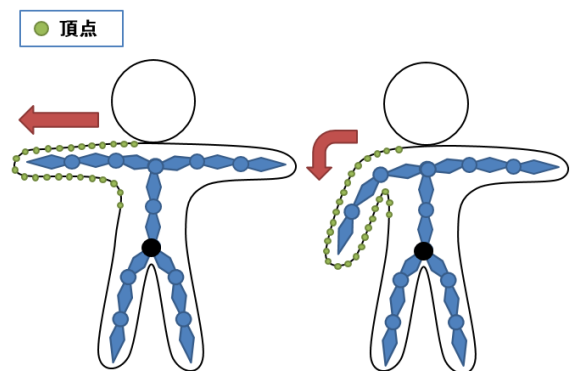


Figure 2. Flow of Motion

持っている。この情報は 3D モデルデータに存在する階層構造を持った情報群である。この情報を用いて、関節の様な動きを行う。親ジョイントの位置や回転情報に子ジョイントは常に追従する。人間で例えた場合には腕を動かした際に肘や手首が釣られて動くような役目を果たす。Figure 2 にモーションの流れを示す。ジョイントの回転によってジョイント同士の位置関係が変化する。その後、モデルデータの頂点位置をそれぞれ対応したジョイントに追従させることで頂点位置が変化し、3D モデルが動いているかのように見せることができる。

2.4. モーションファイル生成について

モーションファイルとは、どのフレームにどのジョイントがどの程度回転するのか、という情報が記述さ

れたファイルである。その為、モーションファイル生成にはフレーム番号、ジョイント名、回転量といった情報が必要となる。

2.5. Kinect から取得する回転情報の変換について

Figure 3 に回転情報の変換について示す。モーションの際の回転は親に追従するため、ここで必要とされる回転量は親ジョイントを基準とした相対的な回転量となる。Kinect から受け取る回転量は Kinect のカメラ座標系を基準としたものでため、親を基準とした相対的な回転量に変換する。任意ジョイントのカメラ座標系における回転行列を M_C 、その親ジョイントのカメラ座標系における回転行列を M_P 、親ジョイント基準の回転行列を M_{local} と表すと式(1)で表すことが出来る。

$$M_{local} = M_C * M_P^{-1} \quad (1)$$

3. 結果・考察

3.1. ソフトウェア構築

Figure 4 にモーションキャプチャシステムのアプリケーション画面を示す。画面左側でアプリケーションの制御を行う。画面右側には取得中ジョイント情報のリアルタイム 3 次元ビューが表示される。

3.2. モーションファイルの生成

Figure 5 の画像群の左が実際のポーズ、中央が生成モーションファイルの再生時の画像である。実際のポーズがモーションファイルで再現され、再生出来ていることが分かる。

3.3. モーションファイルの利用

Figure 5 の画像群の右が生成モーションの利用例である。今回は統合開発環境を備えた汎用ゲームエンジンである Unity を用い、3D モデルにモーションファイルを適用した。BVH ファイル単体で確認したものと同一のアニメーションを画面上で確認することができた。

4. むすび

本研究では、Kinect を用いたマーカーレスで安価なモーションキャプチャシステムを構築し、短時間でモーション製作技術を要せずにモーションファイルの生成を行った。また、生成したモーションファイルを用いて Unity で利用できることを確認できた。

5. 参考文献

[1] MSDN Kinect for Windows SDK

<https://msdn.microsoft.com/en-us/library/dn799271.aspx>

[2] Biovision BVH

<http://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html>

[3] Unity

<http://japan.unity3d.com/>

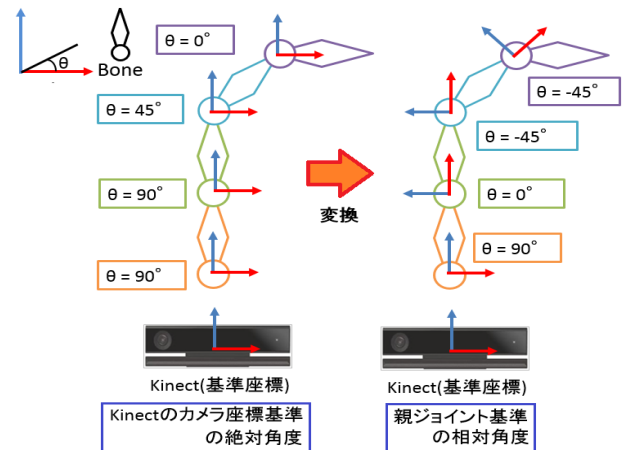


Figure 3. Conversion from the Kinect camera base angle to the parent joint base angle

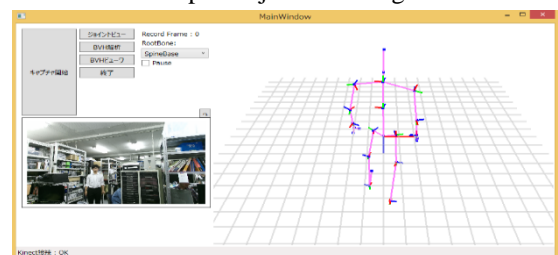


Figure 4. Application view of motion capture

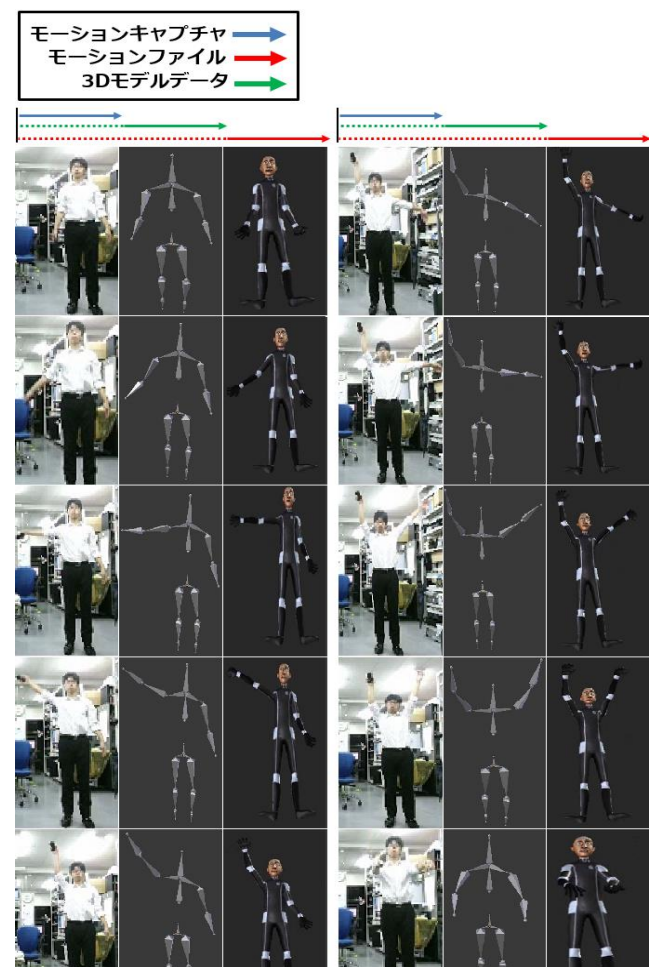


Figure 5. Comparison actor attitude and BVH file and playback motion in Unity