

モバイル MR システム構築のための機能分散型フレームワーク(3) —コンテンツアニメーションの記録・再現手法の提案—

A Distributed Framework for Mobile Mixed Reality System (3)
- A Recording and Reproducing Technique of Content Animation -

岩黒達也, 柴田史久, 木村朝子, 田村秀行

Tatsuya Iwakuro, Fumihisa Shibata, Asako Kimura and Hideyuki Tamura

立命館大学大学院 理工学研究科 (〒525-8577 滋賀県草津市野路東 1-1-1)

Abstract: In this paper, we propose a method for recording and reproducing animation of the contents in order to realize “free-hand” motion. We have developed a distributed framework for mobile mixed reality system which controls the motion of the contents based on the original script language. However, it is difficult to express “free-hand” motion using such a language. Therefore, our method records position and orientation provided by a sensor system and removes noises caused by error of sensors. Then, to reproduce animation of the contents, our method interpolates position and orientation of the contents based on Bézier curve.

Key Words: Mixed Reality, Mobile System, Animation

1. はじめに

我々はこれまで、複数の端末が同一の複合現実 (Mixed Reality; MR) 空間を共有可能なモバイル MR システム構築のためのフレームワークについて検討してきた[1]. 本フレームワークは、サーバ・クライアント型のシステムを採用し、コンテンツをサーバが一元管理することで複数のクライアントで同一の MR 空間の共有を可能にする。また、MR 提示するコンテンツを独自に設計したスクリプト言語により制御する機構を開発し、各コンテンツの動き (以降、アニメーションと呼ぶ) を独自スクリプトで記述することで複雑な動きの表現やインタラクションに対する制御などが可能となった[2]. しかし、自由曲線上を移動するようなアニメーション、すなわち任意の点を結ぶ自由な軌跡は数式化が難しく、これをプログラムによって表現するのは、記述が煩雑であるといった問題がある。

そこで本研究では、任意のアニメーションを容易に入力可能で、かつ独自スクリプトによる制御機構内で扱うことができるアニメーションの記録・再現手法を検討する。

2. コンテンツ制御機構

2.1 コンテンツの制御方法

図 1 に示すように、本フレームワークにおけるコンテンツの制御機構は、サーバに集約されており、スクリプトを常時実行することで、コンテンツの位置姿勢を制御している。クライアントで MR を実現するには、理想的には、フレーム更新に同期してコンテンツの位置姿勢の情報をサーバから取得する必要があるが、通信するデータ量と速度から実現は困難である。そこで本フレームワークでは、サーバで生成したコンテンツ情報をクライアントに一定間隔で送信し、クライアントは次の通信までの間、前回サー

バから取得した情報と新たにサーバから取得した情報を補間することで、コンテンツを提示する。

補間処理には、3 次ベジェ曲線のアルゴリズムを応用し、クライアントの取得した 2 つの端点の位置情報に対し、速度ベクトル情報から制御点を生成することで滑らかなアニメーションを実現する[3].

2.2 独自スクリプトにおけるアニメーション制御の問題

本言語は、MR 空間中に存在する CG などの仮想オブジェクトに対して、位置姿勢情報を制御するプログラムを外部に記述することでその動作を制御する。プログラムでの動作記述により、無限に続くアニメーションや条件によって変化するアニメーションなど自由なコンテンツの制御が可能である。

しかしその反面、任意の点を結ぶ自由な軌跡は数式化が難しく、プログラムで表現するにはその軌跡を辿るすべての点をデータとして準備した上で各点を経過時間に沿って時系列に更新する処理の記述が必要であり、プログラムの作成が煩雑であるという問題がある。そのため本研究では、位置姿勢を取得可能なセンサデバイスを用いた容易な入力によって任意のアニメーションを記録し、その軌跡の表現に必要な記述を独自スクリプトで出力する手法を検討する。

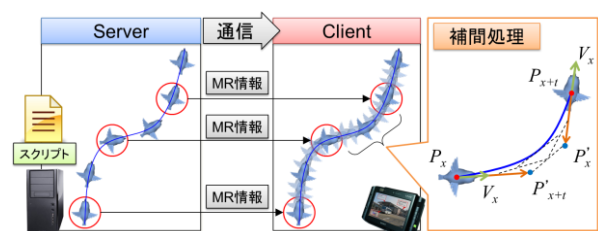


図 1 コンテンツ制御機構の概略図

3. アニメーションの記録・再現手法

3.1 設計方針

本研究では、以下の3つを目標とするアニメーションの記録・再現手法について検討する。

- 任意のアニメーションを表現可能
- センサデバイスを用いた容易なアニメーション入力
- 独自スクリプトによる制御機構との高い互換性

任意のアニメーションは、マーカやセンサなどの位置姿勢検出機構をデバイスとして利用し、デバイスを実空間中で自由に動かすことでその動作の軌跡を入力する。基本的には、入力された軌跡はデバイスのサンプリング間隔で取得される点列として独自スクリプトに記述し、記録した点情報を経過時間に沿って時系列に更新することで任意のアニメーションを再現することが可能となる。

しかし、位置姿勢を計測するデバイスの出力は一般に誤差を含んでおり、単純に取得した点を記録するだけでは意図した軌跡を再現することは難しい。また、取得する点の数はアニメーションの長さに比例して膨大となり、結果として独自スクリプトのコード量が増加する。そこで、できる限り少ない点情報からアニメーションを再現することを考える。

また、生成したアニメーションを、独自スクリプトのメソッドとして出力することを考える。これにより、既存の制御機構で表現可能なアニメーションに加え、任意のアニメーションを既存の制御機構内で扱うことができ、より柔軟性の高いコンテンツの表現が可能となる。

3.2 概要

本手法によるアニメーション生成の流れを図2に示す。本手法では、入力したアニメーションの再現には2.1節で解説した速度ベクトル情報を利用した3次ベジエ曲線のアルゴリズムを用いる。このアルゴリズムでは、端点となる点情報のみで曲線を生成できるため、少ない点情報でアニメーションを再現することができる。そのため本手法では、取得した全ての点情報からアニメーションの再現に必要な点をキーフレームとして抽出する。

しかし、デバイスから取得したアニメーションの点情報には、デバイスの検出ミスや手ぶれにより、必然的に誤差が含まれるため、アニメーションの再現に不要な点が誤差によりキーフレームとして抽出される可能性がある。そこで、取得点の最適化を行い、誤差によるキーフレーム抽出

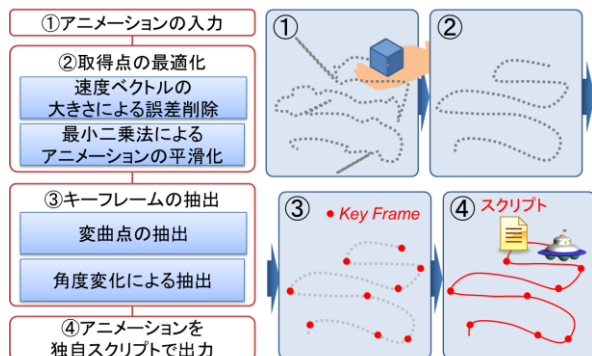


図2 アニメーション生成の流れ

表1 取得点のパラメータ

パラメータ	内容
位置情報	世界座標系における3次元座標[mm]
姿勢情報	モデルの各軸に対する回転[rad]
速度ベクトル情報	3次元ベクトルの各ベクトルの成分[mm/sec]
時間情報	アニメーション入力開始時からの経過時間[sec]

を少なくする。

また、生成したアニメーションは独自スクリプトでメソッドとして記述するが、メソッドはキーフレームから3次ベジエ曲線を生成する補間処理部とキーフレームを更新するキーフレーム情報部で構成され、経過時間に沿ってキーフレームを更新することで、キーフレーム間を補間する3次ベジエ曲線を生成し、アニメーションを再現する。

次節にこれらの処理の詳細を説明する。

3.3 アニメーションの生成

3.3.1 アニメーションの入力

本手法では、位置姿勢検出機構をデバイスとして用い、実空間中で自由に動きを描くことで、任意のアニメーションを入力する。実空間中で行うことで、配置が入り組んだ実物体の間を仮想物体が通り抜けるなど、実物体と仮想物体の位置関係を考慮したアニメーションを入力することが可能となる。

アニメーションの入力は点列として、各点においてデバイスから取得した位置姿勢情報、速度ベクトル情報、および入力開始時からの経過時間を記録することで実現する。各点 P_i における速度ベクトル V_i は、次点 P_{i+1} を用いて以下の式1から算出する(図3参照)。

$$V_i = (P_{i+1} - P_i) \quad (式1)$$

各点において記録されるパラメータを表1に示す。

3.3.2 取得点の最適化とキーフレームの抽出

入力したアニメーションは、2.1節で解説した手法を用いて再現するため、取得した全ての点情報からアニメーションの再現に必要な点をキーフレームとして抽出する。

しかし、デバイスにより取得した点情報には必然的に誤差が含まれるため、取得した点群から誤差を削除し、平滑化関数を生成することで取得点を最適化する。

以下にそれぞれのアルゴリズムと特徴を述べる。

(1) 取得点の最適化

(A) 速度ベクトルの大きさによる誤差の削除

取得されるアニメーションの点情報取得間隔はデバイスのサンプリング間隔によって決まるため一定であり、速

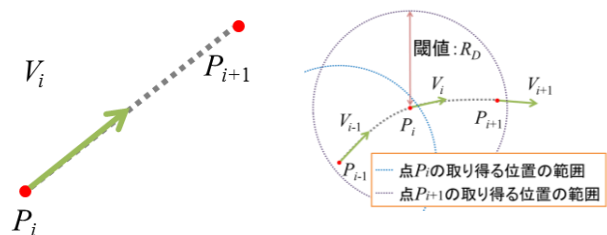


図3 速度ベクトル情報の計算

図4 速度ベクトルの大きさによる制限

度ベクトルの大きさにおいてもその変化量の限界は一定であると仮定できる。そこで、図 4において点 P_i に対する次点 P_{i+1} の取り得る位置の範囲を限定する。そのためこの速度ベクトルの大きさの変化量に閾値 R_D を設け、閾値を超える点を異常点として点群から削除する (式 2)。

$$|P_{i+1} - P_i| > R_D \quad (式 2)$$

(B) 最小二乗法によるアニメーションの平滑化

誤差を含むアニメーションに対し 3.3.1 項で解説した速度ベクトル情報の算出を行うと、各点における速度ベクトルの向きは誤差を含む 2 点間から算出されるため、アニメーションの進行方向と必ずしも一致しない。そのため本手法では、取得したアニメーションの点情報から、平滑化関数を生成し、速度ベクトル情報の再計算を行うことで、より正確な速度ベクトル情報を算出する。

平滑化関数は、任意の点群から指定した次数の近似関数を生成する最小二乗法により生成する。また、アニメーションにおいて平滑化を行う区間と、そこで生成される平滑化関数の次数を自由に設定し、アニメーションの平滑化具合を任意に変更することができる。

(2) キーフレーム抽出処理

上記で最適化したアニメーションの点情報からアニメーションの再現に必要な点をキーフレームとして抽出する。本手法では、3 次ベジェ曲線のアルゴリズムを応用することでアニメーションを表現するが、3 次ベジェ曲線で表現できる曲線は、曲線における変曲点が 1 つ以下のものである。すなわち、変曲点を 2 つ以上含む曲線は表現できないため、アニメーションの軌跡における変曲点をキーフレームとして抽出する。

変曲点は曲線の凹凸が変化する点であるため、曲線においてベクトルの単調増加と単調減少が入れ替わる点である (図 5)。よって、式 3 を用いて変曲点を検出し、キーフレームとして抽出する。

$$(V_{i-1} \cdot V_i)(V_i \cdot V_{i+1}) < 0 \quad (式 3)$$

しかし、変曲点のみをキーフレームとして利用した場合、単純な円運動など同一方向にベクトルの向きが変化し続ける動きは、変曲点が存在しないためアニメーションを再現できない。そこで変曲点に加え、同一方向への角度変化量が閾値 R_K を超えた場合、その点をキーフレームとして利用する。図 6 において前回取得したキーフレーム K_j の速度ベクトルを V_{K_j} とし、以下の式 4 を満たす点 P_i をキーフレームとして抽出する。これは、 $K_j \leq i \leq K_{j+1}$ を満たす。

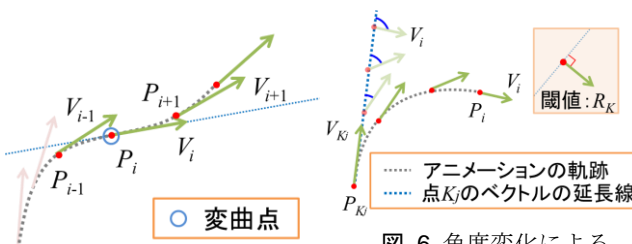


図 5 変曲点の検出

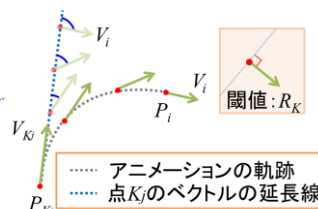


図 6 角度変化によるキーフレーム抽出

$$a \cos \left(\frac{V_{K_j} \cdot V_i}{|V_{K_j}| |V_i|} \right) > R_K \quad (式 4)$$

これらにより、あらゆる曲線の生成に必要な点をキーフレームとして抽出することができる。

本手法で誤差として検出する情報は、それが任意によるものか、デバイスによる誤差のいずれによるものであるか判断できないため、本項で設定した閾値を変更することにより、最適化処理、キーフレーム抽出処理の度合いを変更することができる。これにより揺れの少ない滑らかなアニメーションだけでなく、小刻みに揺れるアニメーションなども再現することができる。

3.3.3 独自スクリプトコードの出力

前項で抽出したキーフレームを独自スクリプトで出力し、キーフレーム間を補間することでアニメーションを再現する。生成するスクリプトコードは、キーフレーム情報を記録し、時系列にキーフレームの更新を行うキーフレーム情報部と、キーフレーム間を 3 次ベジェ曲線によって補間する補間処理部から構成される。これらを独自スクリプトのメソッドとして出力することで、任意のアニメーションを既存の制御機構内で扱うことが可能となる。

また、本手法におけるアニメーション生成は、実物体と仮想物体の位置関係を考慮したアニメーションの生成、プログラムで記述し難いアニメーションの生成の 2 つの目的が考えられるため、本手法ではアニメーションの記録開始点を保持し、開始点を原点とした座標系を出力する。

4. 検証実験

4.1 実験目的

本手法の各処理が正しく動作し、本手法によりアニメーションの記録・再現が可能か確認するため、以下の 3 つの実験を行う。なお、位置姿勢検出機構には ARToolKit[4] を全面に貼り付けた立方体をデバイスとして使用する。

【実験 1】理論値に対するキーフレーム抽出

XY 平面上でサインカーブするアニメーションにおいて、手入力の代わりにプログラムで生成した理論値に対し、本手法のキーフレーム抽出を適用することで、キーフレーム抽出処理が正しく動作するか確認する。

【実験 2】機材により入力したアニメーションの最適化

実空間中で半径 11.5cm の円運動を毎分 7 回転行う実験機材をモーターとギアボックスを組み合わせて作成し、機材の先端にマーカを固定して実験を行った。取得した点情報に対し本手法の最適化を適用することで、最適化処理が正しく動作するか確認する。

【実験 3】数式化困難なアニメーションの記録・再現

三日月型のアニメーションを、デバイスを用いて手入力し、取得した点情報に対し本手法の全処理を適用することで、入力したアニメーションを正しく再現可能か確認する。また、各処理に設けた閾値と平滑化を行う区間の変更により、アニメーションの再現度を変更可能か確認する。

各実験で設定した閾値を表 2 に示し、扱う点情報の数を

表 2 実験で設定した閾値

	実験 1	実験 2	実験 3 (閾値 A)	実験 3 (閾値 B)
ベクトルの大きさによる制限 [mm/sec]		50.0	50.0	50.0
平滑化関数の次数[次式]		3	15	10
角度変化によるキーフレーム [°]	90.0	90.0	90.0	90.0

表 3 実験で扱う点情報の数

	実験 1	実験 2	実験 3 (閾値 A)	実験 3 (閾値 B)
理論値, 入力値	249	323	86	86
キーフレーム	11	16	12	16

表 3に示す。また、実験 1の結果を図 7、実験 2の結果を図 8、実験 3の結果を図 9、図 10に示し、各図中の灰色点は理論値・入力値、橙色枠は平滑化開始点、緑色枠は平滑化終了点、ピンク色点は最適化処理結果、赤色点はキーフレーム、水色点はアニメーション再現結果を表す。

4.2 実験結果

【実験 1】理論値に対するキーフレーム抽出

変曲点によるキーフレームが抽出され、その補助の役割として前キーフレームとのベクトルの角度変化によりキーフレームが抽出された。以上の結果から、キーフレーム抽出処理が正しく動作していることを確認した。

【実験 2】機材により入力したアニメーションの最適化

入力値に誤差を含むアニメーションに対し、最適化処理により滑らかなアニメーションを生成できた。ベクトルの大きさにより検出された誤差は削除され、平滑化関数により滑らかな曲線を生成することができた。以上の結果から、最適化処理が正しく動作していることを確認した。

【実験 3】数式化困難なアニメーションの記録・再現

数式化困難な軌跡に対し、正しくアニメーションが再現される結果が得られた。また、閾値と平滑化する区間の変更により、アニメーションの再現度が変化することを確認できた。以上の結果から本手法が数式化困難なアニメーションに容易に対応でき、閾値によりその再現度を変更することができることを確認した。

4.3 考察

前節の実験結果から、実装したアニメーションの記録・再現手法が正しく動作していることを確認した。しかし、アニメーションを滑らかにすることでバウンドなどの速度ベクトルが急激に変化する点がキーフレームとして抽出されなかったため、これらの点を多く含むアニメーションを滑らかに再現することは難しいと考えられる。

5. 今後の課題

本手法では、アニメーションを点の連続として扱い、各点は3.3.1項で示した多くのパラメータを持つため、点情報のみからアニメーションの全容を把握・編集することは困難である。そこで、本手法を用いたアニメーションの生成・編集を支援するオーサリングツールの開発を検討する。

本オーサリングツールは、アニメーションの点情報からその軌跡を VR 画面上にプロットすることで可視化する。

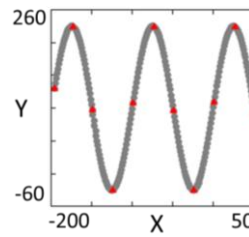


図 7 実験 1 の結果

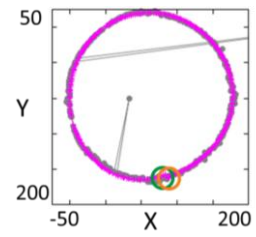


図 8 実験 2 の結果

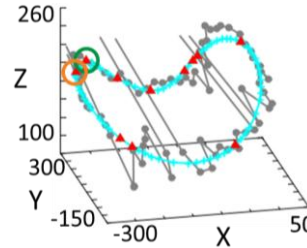


図 9 実験 3 (閾値 A) の結果

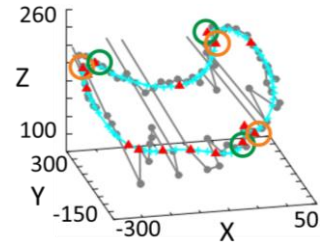


図 10 実験 3 (閾値 B) の結果

また、本手法における閾値の変更によるアニメーションの変化を変化前と同時に表示することで、その設定を容易にする。さらに、以下の機能を備えることで、アニメーションの編集を容易にする。

- シークバーによる閾値の調整
- アニメーションの拡大・縮小機能
- アニメーションのプレビュー機能

6. むすび

本稿では、モバイル MR システムのためのアニメーション記録・再現手法について述べた。本手法により、任意のアニメーションを容易に入力可能で、且つ独自スクリプトによる制御機構と互換性のあるアニメーションの記録・再現が可能となる。今後は、オーサリングツールの開発を行い、本手法を用いたアニメーションの生成・編集を支援する。

参考文献

- [1] 柴田他：“多様な可搬型機器に対応可能な複合現実感システムの共通フレームワークの設計と実装”，日本バーチャルリアリティ学会論文誌，Vol. 10, No. 3, pp. 323 - 332, 9. 2005.
- [2] 山下他：“モバイル MR システム構築のための機能分散型フレームワーク - システムアーキテクチャとコンテンツ制御機構 -”，第 14 回日本バーチャルリアリティ学会大会論文集，3A2-3, 9.2009
- [3] 縄谷他：“モバイル MR システム構築のための機能分散型フレームワーク (2) -コンテンツ制御機構の拡張-”，本大会，2010.
- [4] 加藤他：“マーカ追跡に基づく拡張現実感システムとそのキャラブレーション”，日本バーチャルリアリティ学会論文誌，Vol. 4, No. 4, pp. 607 - 616, 1999.