

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 -

A Distributed Framework for Mobile Mixed Reality System
- System Architecture and Content Control Mechanism -

山下智紀, 荒川祥太郎, 柴田史久, 木村朝子, 田村秀行

Tomonori Yamashita, Shotaro Arakawa, 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, yamasita@rm.is.ritsume.ac.jp)

Abstract: This paper describes design and implementation of a distributed framework for building mobile mixed reality (MR) system. We are aiming at developing the framework that can support variety of mobile devices and share the same MR space via network. We adopt not peer-to-peer architecture but client-server architecture for our framework to get through diversity of mobile devices. Our framework consists of four parts; a MR server, a mediator, clients, and thin clients. All functions necessary for MR are distributed to the four parts according to the role of the functions. In addition, we designed a script language, which borrows object-oriented technology from the Java language, to control the motion of contents such as virtual 3D models and annotations. This framework makes it possible to develop applications in which variety of mobile devices share the same MR space.

Key Words: *Mixed reality, Mobile device, System architecture, Content control*

1. はじめに

近年の急激な性能向上を背景に, モバイル端末を利用した複合現実感 (Mixed Reality; MR) システムへの期待が高まりつつある. しかし, システムの普及を念頭に置くと, 個々のアプリケーションを個別に一から開発するのは無駄が多いため, アプリケーション開発を容易にするフレームワークが今後の MR の発展において重要な鍵となる.

モバイル複合現実感のためのフレームワークについては, 既にいくつかの試みがある. Bauer らは, ウェアラブルコンピュータのためのフレームワークとして DWARF (Distributed Wearable Augmented Reality Framework) [1] を提案している. AR に必要な機能をモジュール化することでシステムの再利用を可能にしているが, 携帯電話などの処理能力の低い機器は考慮していない. Schmalstieg らは, ハンドヘルドデバイスのためのフレームワークである Studierstube ES を提案している [2]. 当該フレームワークは, PDA やスマートフォンなど, 小型のモバイル端末で MR 機能を実現することに重点を置いているため, 多様な端末が同時に MR 空間を共有することは想定していない. 我々も先行研究 [3] として, 多種類のモバイル機器に対応したフレームワークを提案していたが, インタラクティブ性やコンテンツ共有のリアルタイム性の点で問題を抱えていた.

また, フレームワーク上でアプリケーションを開発する場合, 提示する CG などのコンテンツの制御を行う機構を開発するアプリケーションの表現の幅を決定づける重要な要素となる. コンテンツを制御する既存の手法として,

VRML や X3D といったモデル記述言語の使用が考えられる. しかしこれらは時系列に沿ったアニメーションの記述を行うため, 無限に続く動きや不連続な動きなどの表現ができず, 柔軟性が乏しいといえる.

そこで本研究では, 携帯電話, PDA, UMPC など, 多様なモバイル機器に対応可能であり, 複雑なアニメーション制御を容易に実現可能な MR システム構築のためのフレームワークの開発を目指す.

2. 機能分散型フレームワーク

2.1 設計方針

我々は, モバイル MR システム構築のためのフレームワークには以下の 3 要件が必要と考えている.

- アプリケーションに非依存
- モバイル機器の種類や性能の差を吸収可能
- 複数端末が同一の MR 空間を共有可能

これらの要件を満たすフレームワークを設計した. 図 1 に提案フレームワークのコンセプト図を示す.

本フレームワークは, サーバ・クライアント型のシステムとして, MR サーバ, メディエータ, クライアント, シンククライアントの 4 つの要素で構成する. 処理能力の低い機器 (シンククライアント) については, サーバとの間にメディエータを導入し, 負荷の大きい処理を委託させる. これにより機器間の性能の差異を吸収し, 様々な機器に対応することができる. また, 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. さらに, システム構成をシステムレイヤとア



図1 提案フレームワークのコンセプト

アプリケーションレイヤに分割した上で、システムレイヤを我々が提供することにより、アプリケーションの開発が容易になる。

2.2 機能設計

提案するフレームワークでは、MRシステムに必要な機能の分散配置を行う。一般にMRを実現するためには表1の機能が必要となる。このうち、クライアント管理とデータベース管理は複数のユーザが同時に参加可能なサーバ・クライアント型を採用するために必要となる。

これらの7つの機能をサーバとクライアントに振り分けることについて検討した。まず、MRサーバにクライアント管理とデータベース管理を配置することでクライアントの一元管理を行う。加えてコンテンツの位置姿勢などのMR情報生成機能をサーバに配置することで、唯一のMR空間を構築することが可能となる。一方、画像取得機能とMR画像提示機能は、クライアント側に配置する。それ以外の機能については、クライアントの処理能力に合わせ、クライアントまたはクライアントとサーバを取り持つメディエータへ配置することで、処理能力差の吸収を目指す。

2.3 システムアーキテクチャ

図2は、前述の7つの機能を分散配置した図である。以降では、それぞれの役割について概要を述べる。

[MRサーバ]

MRサーバはシステム全体を管理し、MR空間の管理・制御を行う。各クライアントの情報とコンテンツの情報を統合し、MR画像をレンダリングするために必要なMR情報を生成する。コンテンツの動きやユーザのインタラクションの反映は後述するスクリプト言語によって記述し、スクリプトエンジンによって処理される。

表1 機能の概要

機能	概要
画像取得	現実背景の画像を取得する
位置姿勢検出	ユーザ(端末)の位置や姿勢を検出する
MR情報生成	重畳するMR情報を生成する
MR画像生成	MR情報を現実の光景に重畳描画する
MR画像提示	ユーザにMR画像を提示する
クライアント管理	クライアントの情報を管理する
データベース管理	コンテンツなどのデータを管理する

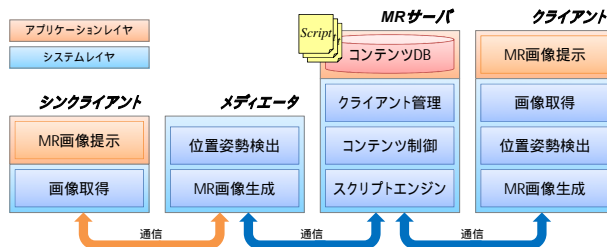


図2 機能の分散配置

[クライアント]

クライアントとしては、UMPCのような比較的处理能力の高い機器を想定する。画像取得、位置姿勢検出、MR画像生成、MR画像提示の機能を持ち、MR画像は動画像で提供する。また、MR空間に存在するコンテンツに対し、インタラクションを行うことができる。

[シンクライアント]

シンクライアントは、携帯電話のような処理能力の低い機器を想定する。画像取得、MR画像提示の機能のみを有し、位置姿勢検出やMR画像生成などの負荷の重い機能はメディエータに委ねる。MR画像の提示は、シンクライアントの処理能力を考慮して、静止画で提示する。

[メディエータ]

メディエータは、シンクライアントとMRサーバの仲介を行い、シンクライアントの代わりに位置姿勢検出、MR画像生成の処理を実行する。メディエータが仲介を行うことで、MRサーバからはシンクライアントの存在を隠蔽し、通信内容を統一することが可能となる。

2.4 処理の流れ

クライアントにおける画像提示の流れを図3に、シンクライアントにおける画像提示の流れを図4に示す。MRサーバは、スクリプトエンジンによってMR空間を構築するための情報を更新し、クライアントとメディエータに送信する。クライアントはMRサーバから得た情報から自身でMR映像を生成・提示し、メディエータはシンクライアント

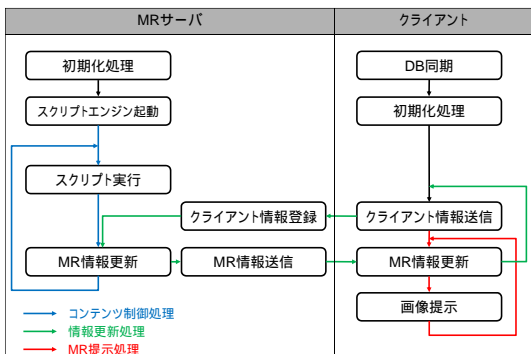


図3 処理の流れ(クライアント)

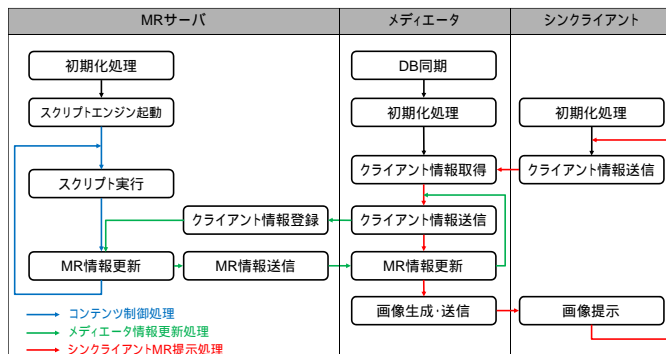


図4 処理の流れ(シンクライアント)

トからアクセスがあった場合、保持する情報から MR 画像を生成して送信する。

3. コンテンツ制御機構

3.1 設計方針

本フレームワークでは以下の 2 点に重点を置き、コンテンツ制御機構の設計を行った。

- 複雑なアニメーションを表現可能
- 記述が容易

提案する制御機構は、MR 提示に用いる 3D モデルなどのコンテンツに対し、その動きを制御するスクリプトを外部に記述することで実現する。本制御機構は、個々のコンテンツの情報とその動きを記述したスクリプトを併せて 1 つのオブジェクトとして捉える、というオブジェクト指向的な考えを導入して設計した。スクリプトには、時間経過やクライアントからのインタラクション等によって発生するコンテンツの変化を記述する。

提案するスクリプト言語は、プログラミングの基本的な知識がある者の使用を前提とする。そのためオブジェクト指向言語の一つである Java と類似した記述法を採用した。一般的なプログラミング言語をベースに文法を設計することによって、アプリケーション開発者は初めて本言語を使用する場合でも、本言語独自の部分を覚えるだけで容易に利用が可能となる。しかし、本言語は様々な機能を実現できる C 言語や Java のような汎用性を実現することは目的としない。本言語が持つ機能は、MR においてコンテンツを制御することに特化したものに限定し、それらの機能を容易に利用できる設計とする。

3.2 コンテンツの構造

提案フレームワーク上で扱うコンテンツは、オブジェクトとクライアントの 2 つに分けられる。オブジェクトは CG などの仮想オブジェクトと実物体を示す現実オブジェクトに大別される。仮想オブジェクトは、3D モデル、2D モデル、音源などをノードとした木構造で構成され、そのノードをコンポーネントと呼ぶ。提案するスクリプト言語は、仮想オブジェクトの動作を、オブジェクト全体あるいは、コンポーネント単位で操作する。また、仮想オブジェクトを操作する際には、MR 空間中に存在する現実オブジェクトやそのアプリケーションを実行しているクライアントが持つ情報を利用した制御が可能とする。

3.3 提案制御手法の概要

以上の設計方針を基に設計したコンテンツ制御機構の処理系について解説する。本制御機構はサーバに集約し、スクリプトの解析・実行を繰り返すことで MR 空間中に存在する全てのオブジェクトが保持する表 2 に示すパラメータの更新を行う。またクライアントは一定時間間隔でサーバから MR 空間の情報を取得し、その際に自身の位置姿勢情報やユーザインタラクションの情報をサーバに送信することで MR 空間を更新する。

この制御法を用いてクライアントで MR 空間を提示するには、端末のフレーム更新時間と同間隔で毎フレームの

表 2 仮想物体のパラメータ

パラメータ	内容
double x, double y, double z	位置
double x, double y, double z	姿勢 (各軸の回転)
boolean presentation	表示 (再生) / 非表示 (停止)
int state	状態

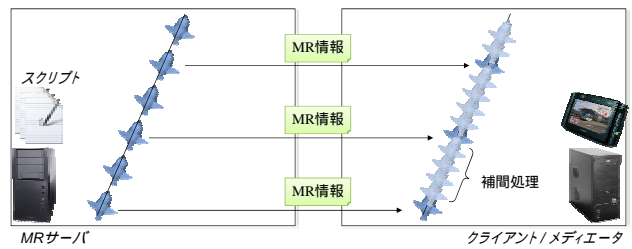


図 5 コンテンツ制御の概略

情報をサーバが送信する必要があるが、通信するデータ量と速度から実現は困難である。そこで本手法では、サーバは常時 MR 空間の更新を行い、クライアントはサーバから MR 空間の情報を一定間隔で取得する。クライアントは次の通信までの間、自身が保持する現時点の情報とサーバから取得した情報を補間することで現在時刻での MR 空間の情報を生成し、映像を提示する (図 5)。これにより、サーバとの通信頻度に依存しない MR 提示が可能となる。

4. 提案スクリプト言語

4.1 提案スクリプト言語の概要

本言語では、MR 提示する個々のコンテンツ毎に、雛形となるクラスを定義し、そこにメソッドとしてオブジェクトの動きを記述する。後述する MR 空間を表現したクラスの中で、これらのクラスのインスタンスを生成することで、MR 空間中に仮想オブジェクトを配置する。クライアントと現実オブジェクトの情報は、システムが自動的にそれぞれその情報を格納するクラスのインスタンスを生成する。

4.2 言語仕様

前節の方針を踏まえ、本言語の仕様を設計した。記述の容易性を重視するため Java 言語をベースに設計を行った。基本的には、Java の仕様のサブセットとして実現している。本言語特有の仕様を表 3 に示す。

本言語の字句構造、構文構造は Java と同一のものを採用する。変数はプログラムの構造を簡素化するため、プリミティブ型の変数 5 つに限定する。クラスではなく文字列型として文字列を扱うのは Java と異なる点である。変数には一次元配列を定義可能とし、固定長の配列と可変長の配列の宣言を可能とする。また配列長については、定数フィールドとして扱う Java とは異なり、配列長を取得する length メソッドを用意する。

本言語では、表 3 に示すクラスを用意し、プログラム作成者がその他の処理のために独自にクラスを定義することは不可とする。MRWorld クラスは MR 空間を表現するクラスであり、全てのコンテンツのインスタンスを管理する。アプリケーション 1 つに対し唯一のインスタンスを生成し、コンテンツ間の関係性を利用した記述や、仮想オブジェクトの生成、削除を記述することが可能である。Object クラスは、すべての仮想オブジェクトの親クラスに相当す

表3 言語仕様

変数 (プリミティブ型)	
整数型	int, long
実数型	double
論理型	boolean
文字列型	string
演算子	
関係演算子	<, <=, >, >=, ==, !=, instanceof
数値演算子	+, -, *, /, %
条件演算子	&&,
クラス	
MRWorld	コンテンツのインスタンスを管理
Object	オブジェクトの情報を格納
Client	クライアントの情報を格納
Position	コンテンツの位置情報を格納
Orientation	コンテンツの姿勢情報を格納
Scale	コンテンツの拡大率を格納
Math	円周率と算術メソッドを保持
Time	時刻情報[時間, 分, 秒]を格納
System	スクリプトエンジン特有の情報を保持
システム変数・メソッド (System クラス)	
diffTime	スクリプトの実行間隔 [msec]
elapsedTime	システム起動からの経過時間 [msec]
getObject	オブジェクトの情報を取得する
システムメソッド (MRWorld クラス)	
isCollision	コンテンツ間の衝突判定を行う
getAroundObjects	周辺のコンテンツを取得する
getDistance	コンテンツ間の距離を取得する

る抽象クラスである。このクラスを暗黙的に継承したクラスに仮想オブジェクトの雛形を定義し、インスタンスを生成することで、仮想オブジェクトを MR 空間に配置する。表 2 に示すパラメータを操作することで、オブジェクト自身のアニメーションの制御を記述することが可能である。これら 2 つのクラス内に、記述者が自由にメソッドを定義可能とするほか、一般的に使用頻度の高い三角関数やランダム関数などの算術関数を保持した Math クラス、スクリプトエンジン内の情報を保持する System クラスを呼び出し、利用することが可能である。また MRWorld クラスには、本フレームワーク上でコンテンツを制御する際に利用が想定される処理をメソッドとして用意する。このメソッドを利用することによって、本来全てを記述すると煩雑になる処理を少ない記述で実現することが可能になる。

5. 実装結果

5.1 実験内容

以上の設計に沿ってフレームワークを開発し、提案した機能が想定通りに動作するかを確認した。実験内容は以下の 2 項目である。実験に用いた機器を表 4 に示す。

● 実験 1：クライアント-クライアント間の同期

クライアント A, B の 2 台を用いて、クライアント間でリアルタイムな同期が取れているかについて実験を行った。クライアント A から仮想オブジェクトに対してインタラクションを行い、変化した MR 空間を両端末でリアルタイムに共有できるか、異なる角度から確認した。

表4 機器構成

機器	OS	CPU	SDK
サーバ	WinXP	Core2Duo 1.86GHz	JDK6 / Apache
メディアータ	WinXP	Corei7 2.66GHz	JDK6 / Apache + Tomcat
クライアント A	WinXP	Core2Duo 3.16GHz	VC++ / DirectX 9.0c
クライアント B	WinXP	CoreSolo 1.20GHz	VC++ / DirectX 9.0c
シンクライアント	Symbian	- (docomo SH-04A)	J2SE / Star1.0 Profile

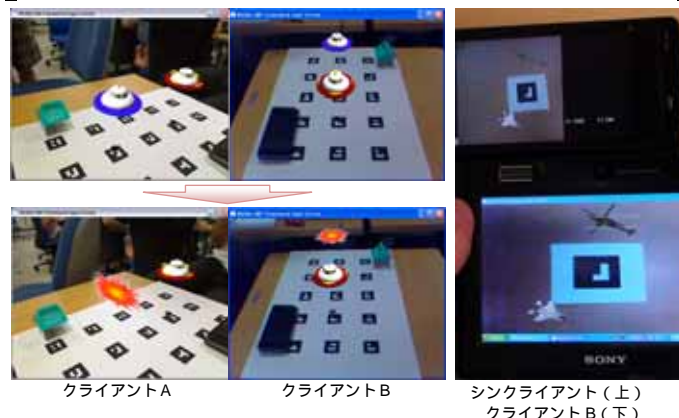


図6 実験結果

● 実験 2：クライアント-シンクライアント間の同期

クライアント B とシンクライアントで、同一の MR 空間を提示する実験を行った。シンクライアントが撮影した MR 空間のスナップショットが、その時刻でのクライアント A が提示した MR 映像と同期がとれているか確認した。

5.2 実験結果と考察

実験 1 の結果を図 6 (左)、実験 2 の結果を図 6 (右) に示す。結果、処理能力の異なる全ての端末間で、同一の MR 空間をリアルタイムに同期可能であることを確認した。しかし、特異点を含む動きの表現をスクリプトで記述した場合、クライアント上で補間した結果が意図した動きと異なるものになるという問題が残った。また、クライアント B のような処理能力が比較的低い端末では、通信にかかる時間が大きくなることで、補間処理の間隔が大きくなり、意図した動きと誤差が生じる問題も発生した。

6. まとめ

本稿では、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について報告した。提案フレームワークでは、MR に必要な機能を分散配置することで多様なモバイル機器に対応が可能である。また、サーバに MR 空間の制御を集約させることで、複数端末間での MR 空間共有を実現した。コンテンツ制御に独自スクリプト言語を設計し、それによる制御機構を実装した。本制御機構を利用することで、アプリケーション開発者は、複雑なコンテンツの動きを容易に表現することが可能となる。今後はアーキテクチャの改良を行うことで、前章で挙げた問題点の解決を図り、より汎用性のあるフレームワークの構築を目指す。

参考文献

- [1] M. Bauer, *et al.*: Design of a component-based augmented reality framework, Proc. of 2nd IEEE and ACM Int'l. Symp. on Augmented Reality (ISAR '01), pp.45 - 54, 2001.
- [2] D. Schmalstieg, *et al.*: Experiences with handheld augmented reality, Proc of 6th IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR 2007), pp.3 - 15, 2007.
- [3] 田中他：多様な携帯・可搬型機器に対応可能なモバイル複合現実感システム(6)-イベント駆動型コンテンツ制御機構の設計と実装-、日本 VR 学会第 12 回大会論文集 pp.39 - 42, 2007.