



# 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(9)

## —動的負荷分散機構の設計と実装—

### A Distributed Framework for Mobile Mixed Reality System (9) - Design and Implementation of Dynamic Load Distribution Mechanism-

荻野 翼, 木村 朝子, 柴田 史久  
Tsubasa Ogino, Asako Kimura, and Fumihisa Shibata

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

**概要:** 我々は、コンテンツの動きが同期した MR 空間を複数の端末で共有可能なモバイル MR システム構築のためのクライアント・サーバ型フレームワークを提案している。本フレームワークでは、同時に利用できる幾何位置合わせ手法は 1 種類であった。そのため、広範囲での利用を想定した実用的なアプリケーションの開発が容易ではなかった。複数の幾何位置合わせ手法を同時に実行する方法が考えられるが、処理性能に限界があるモバイル端末において、現実的な解決策ではない。そこで本稿では、複数の幾何位置合わせ手法を切り替える機構を導入するためのシステム設計、および CGI 生成と幾何位置合わせの処理負荷を動的に分散する機構を設計・実装した結果を述べる。

**キーワード:** 複合現実感, モバイル, フレームワーク, 負荷分散

#### 1. はじめに

近年、スマートフォン等の急速な普及を背景に、モバイル型複合現実感 (Mixed Reality; MR) システムへの期待が高まりつつある。我々は、この種のシステムの普及にはそれを容易に開発するための枠組みが必要との立場から、コンテンツの動きが同期した MR 空間を複数のモバイル端末で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を提案してきた[1]。これまでのフレームワークのシステム設計では、複数のモバイル端末による MR 空間の共有に研究の焦点を当てていたため、同時に利用できる幾何位置合わせ手法は 1 つに限られていた。しかし、広範囲での利用を想定した実用的なアプリケーションを開発するためには、モバイル端末という計算資源が限られたデバイス上で、状況に応じて複数の幾何位置合わせ手法を組み合わせる必要がある。

そこで本稿では、1) 本フレームワークに海津らが提案した位置合わせ手法の動的切替機構 (切替機構) [2]を導入するためのシステム設計と、2) 切替機構の導入によってクライアントの処理負荷が増すため、クライアントの CGI 生成と幾何位置合わせの処理負荷を動的に分散する機構 (ストリーミングサーバ) を設計・実装した結果について述べる (図 1 参照)。

#### 2. システム・アーキテクチャの概要

##### 2.1 MR サーバ

本フレームワークの MR サーバでは、単一の MR 空間を管理・制御する。MR 空間内の CG などのコンテンツの

振る舞いは独自のスクリプト言語で記述され、スクリプトエンジンで解析・実行される。また、各クライアントから受信した情報を基に、MR 空間内の CG の位置姿勢情報などを実時間で更新する。

##### 2.2 クライアント

本フレームワークにおいてクライアントは、実背景取得、カメラ位置姿勢推定、MR 画像の生成・提示などの処理を担う。システムにおけるユーザ・インタフェースの役割も持ち、取得したインタラクション情報などを MR サーバに送信し、MR サーバから CG の位置姿勢情報や時刻情報などを受信する。受信した情報を基に MR 画像を生成・提示することでモバイル端末での MR を実現している。

##### 2.3 クライアントの問題

クライアントには、位置姿勢推定や MR 画像生成など、

##### MRサーバ

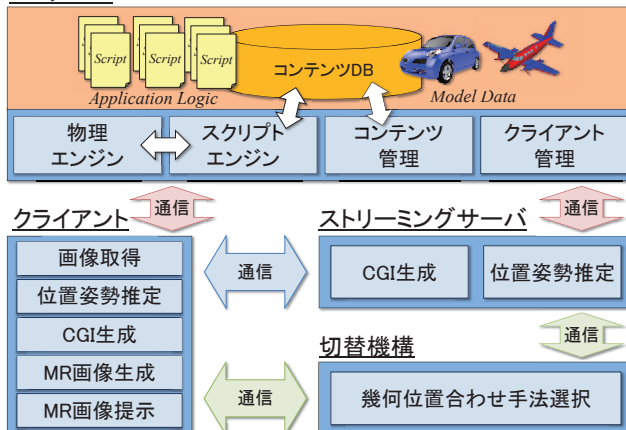


図1 動的負荷分散機構を導入したシステム・アーキテクチャ

コンテンツ制御以外の処理が集中している。そのため、この状態でさらに複数の位置合わせ手法を同時に実行するとパフォーマンスが著しく低下することが予想される。また、コンテンツによっては、CGI生成の負荷が大きくなることも予想される。

### 3. 幾何位置合わせ切替機構の導入

#### 3.1 導入方針

前節で述べたクライアント側に処理が集中する問題を踏まえ、本研究では海津らの手法を導入するにあたり、単一の手法で幾何位置合わせする場合と同程度の処理時間で、複数の幾何位置合わせ手法が切替可能となるようなシステム設計を実現することを目標とした。これまでの個々の研究成果からシステム全体の通信量が増える点が問題になると考え、通信の負荷を可能な限り抑える形で、切替機構を導入する方針を定めた。

#### 3.2 通信間隔の動的変更

海津らの手法では、クライアントはサーバに対して定期的に幾何位置合わせ手法の切替を確認する要求を送信していたが、実際に手法の切替が発生するのは、クライアントの位置が大きく変化した場合である。そこで、クライアントの移動量にあわせてとサーバとの通信間隔  $I$  を動的に変更し、通信量の削減を図る。以降では、マーカベース方式の幾何位置合わせ手法と特徴点ベース方式の手法に分けて、通信間隔  $I$  の決定方法を説明する。

##### 3.2.1 マーカベース方式での決定方法

マーカベース方式では、1フレーム後のマーカの各頂点の2次元座標と移動量、および再投影誤差を利用する。1フレーム後のマーカの各頂点の2次元座標と移動量を推定する方法として、モーションモデルを利用する(図2)。マーカの頂点  $k$  の次の位置を計算するモーションモデルを式(1)、(2)に示す。

$$\mathbf{m}_{t+1,k} = \mathbf{p}_{t,k} + \mathbf{d}_{t+1,k} \quad (1)$$

$$\mathbf{d}_{t+1,k} = 0.5 \cdot \{ (\mathbf{p}_{t,k} - \mathbf{p}_{t-1,k}) + \mathbf{d}_{t,k} \} \quad (2)$$

ただし、 $\mathbf{m}_{t+1,k}$  は次のフレームにおける頂点  $k$  の推定した位置ベクトル、 $\mathbf{p}_{t,k}$  は現在のフレームにおける頂点  $k$  の位置ベクトル、 $\mathbf{d}_{t+1,k}$  は式(2)で算出した移動量を示すベクトルである。実背景の中心から各頂点までのユークリッ

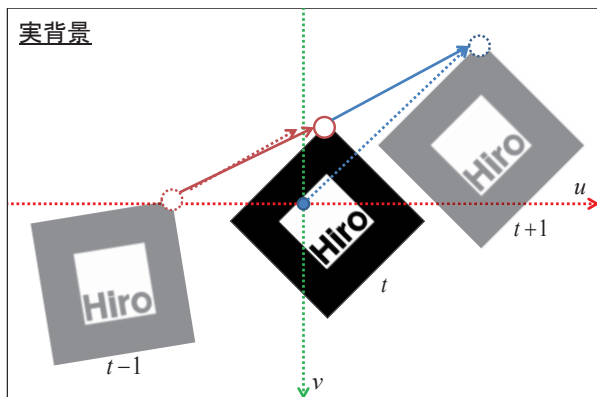


図2 モーションモデル

ド距離  $|\mathbf{m}_{t+1,k}|$  を算出し、最大の数値を利用する。距離が最大となる頂点が実背景の端に最も近く、次のフレーム以降で認識されなくなる可能性が高いためである。

そして、再投影誤差を  $e$  とし、これら  $|\mathbf{m}_{t+1,k}|$ ,  $e$ ,  $|\mathbf{d}_{t+1,k}|$  を用いて通信間隔  $I$  を式(3)、(4)で算出する。

$$I = \frac{C}{|\mathbf{m}_{t+1,k}| \cdot e \cdot |\mathbf{d}_{t+1,k}|} \quad (3)$$

$$k = \arg \max_{i=1,2,\dots,N} |\mathbf{m}_{t+1,i}| \quad (4)$$

ただし、 $C$  は  $I$  を調整するための定数である。式(3)、(4)を用いた場合、常に移動している状況などでは従来[2]よりも通信量が多くなる。

##### 3.2.2 特徴点ベース方式での決定方法

特徴点ベース方式では、追跡しているインライア数と移動量、再投影誤差を利用する。インライアとは、マッチング結果が良好であると判断された特徴点である。特徴点ベース方式の場合、インライア数が一定数以上維持できていれば位置合わせできると考えられる。そこで、次フレームでのインライア数を式(5)で推測する。ただし、 $N_t$  は  $t$  フレーム目におけるインライア数である。

$$N_{t+1} = N_t \cdot \frac{N_t}{N_{t-1}} \quad (5)$$

移動量には、式(6)で示す各特徴点の移動量の平均  $d_t$  を用いる。ただし、 $\mathbf{d}_{t,k}$  は  $t$  フレーム目での特徴点  $k$  の移動ベクトル、 $N$  はインライア数である。

$$d_t = \frac{1}{N} \sum_{k=0}^N |\mathbf{d}_{t,k}| \quad (6)$$

再投影誤差  $e$  を用いて、通信間隔  $I$  を式(7)によって算出する。この場合も、最大通信量は[2]より多くなる。

$$I = \frac{N_{t+1} \cdot C}{d_t \cdot e} \quad (7)$$

## 4. ストリーミングサーバの設計

### 4.1 設計方針

現状のクライアントの設計では CGI の描画はすべてクライアント上で実行するため、2.3 節で述べたように、コンテンツによっては CGI 生成の負荷が増大し、パフォーマンスに影響を与える場合が考えられる。そこで本研究では、クライアント側の処理負荷が増大した際に、一部の処理を代行するストリーミングサーバを導入することを考える。本フレームワークでは、クライアントの一部機能を代わりに実行するメディアータという機構を設計・実装済みであるが[1]、本稿で提案するストリーミングサーバは、代行する機能を動的に変更する点が異なっている。

ストリーミングサーバの導入にあたっては、以下の2つの方針を定めた上でシステムの設計を行った。

- (1) クライアントの処理負荷に応じて、代行する処理を動的に選択する
- (2) 処理時間と遅延時間を可能な限り短く抑える

## 4.2 ストリーミングサーバでの処理

ストリーミングサーバでは、クライアントの処理で特に負荷の大きい CGI 生成と幾何位置合わせを代行する。

### 4.2.1 幾何位置合わせ

ストリーミングサーバで幾何位置合わせする場合、図 3 のように、クライアントは実背景などをストリーミングサーバへ送信する。実背景取得、MR 画像提示はクライアントで実行し、CGI 生成はクライアントの負荷に応じて、クライアント、もしくはストリーミングサーバで実行される。

本フレームワークのメディアータと比べ、幾何位置合わせだけをストリーミングサーバで実行する場合、通信内容が幾何位置合わせ結果のみになるため、通信負荷が小さくなる。また、切替機構とストリーミングサーバで通信することで、クライアントにかかる通信の負荷を軽減する。

### 4.2.2 CGI 生成

ストリーミングサーバで CGI を生成する場合、クライアントは幾何位置合わせに必要な実背景、もしくはクライアントで幾何位置合わせした結果をストリーミングサーバへ送信する(図 4)。そして、クライアントはストリーミングサーバから CGI を受信し、その CGI を実背景に重畳描画することで MR 画像を生成する。

本フレームワークのメディアータとの違いは、ストリーミングサーバからクライアントへ CGI を送信する点である。通信内容を CGI にすることで、クライアントからストリーミングサーバへ送信する実背景の解像度を下げることができる。この場合、ストリーミングサーバによる幾何位置合わせの精度は下がるが、MR 画像提示には影響しない。クライアントで取得した解像度の実背景にストリー

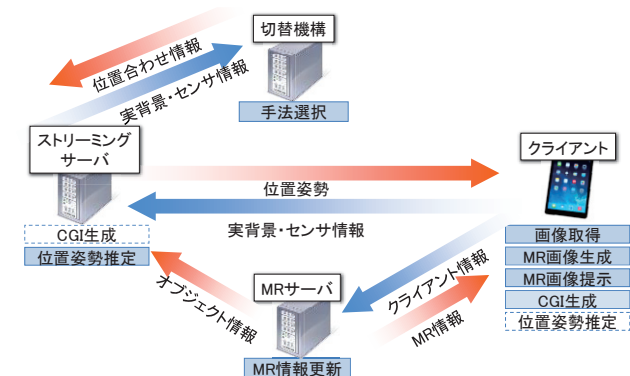


図3 ストリーミングサーバで幾何位置合わせ

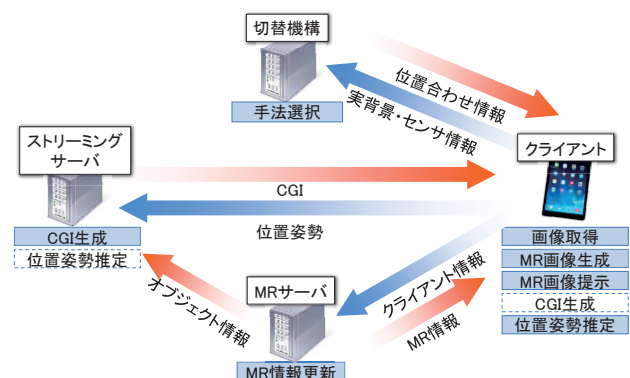


図4 ストリーミングサーバで CGI 生成

ミングサーバから受信した CGI を重ねるからである。

## 4.3 CGI の圧縮

遅延時間を短くするために、ストリーミングサーバがクライアントに送信する CGI を圧縮する。CGI の仮想オブジェクトがない部分は透明である。そこで、仮想オブジェクト付近だけを圧縮して送信する。図 5 は CGI を 16 個の領域に分割した例で、全て透明な領域内を黒色で塗りつぶしている。この領域を送信しないことで通信量を削減する。

## 5. 動作確認

### 5.1 動作機器

ここまで述べた導入・設計方針に沿って、切替機構とクライアントの通信間隔の決定式と、ストリーミングサーバを実装し、動作確認した。動作確認に使用した機器の主な仕様を表 1 に示す。

### 5.2 動作確認と考察

#### 【実験 1】ストリーミングサーバの動作確認

CGI 生成をクライアントで処理した場合(状態 1)と、ストリーミングサーバで処理した場合(状態 2)のそれぞれの 1 フレームの処理時間と遅延時間を計測した。MR 空間に配置するオブジェクト数は、2 個刻みで 2~20 個まで変化させて計測した。配置したオブジェクトはポリゴン数 47187 のアニメーション付 CG モデルである。計測は 100 回行い、平均値をグラフ化した。また、幾何位置合わせはクライアントで ARToolKit を実行した。

#### 【実験 1 の結果と考察】

状態 1 と状態 2 での 1 フレームの処理時間を比較したグラフを図 6 に、遅延時間を比較したグラフを図 7 に示す。状態 1 の場合、オブジェクト数が増加すると 1 フレームの処理時間と遅延時間の両方が増加している。一方、状態 2 の場合、1 フレームの処理時間は常に 30 [ms] 程度である。

表1 動作確認に使用した機器

	クライアント	MR サーバ
CPU	Apple A7	Intel Corei7-2600
Memory	1 GB	8 GB
Network	IEEE802.11n (5GHz 帯) 300Mbps	有線 LAN (1.0 Gbps)
	切替機構 or ストリーミングサーバ	
CPU	Intel Corei7-3960X	
Memory	16 GB	
Video Card	NVIDIA GeForce GTX 560	
Network	有線 LAN (1.0 Gbps)	

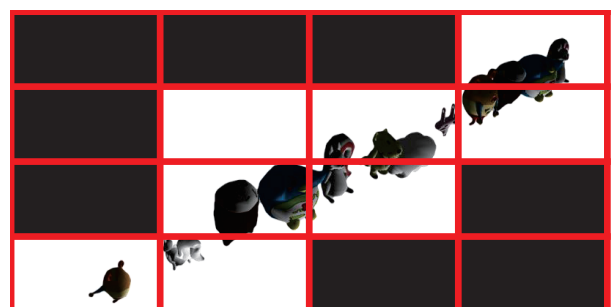


図5 CGI の分割例

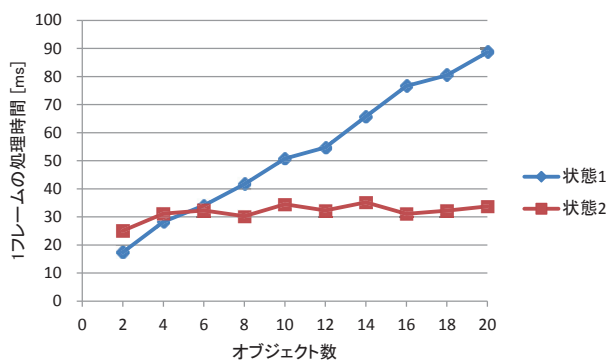


図6 1 フレームの処理時間

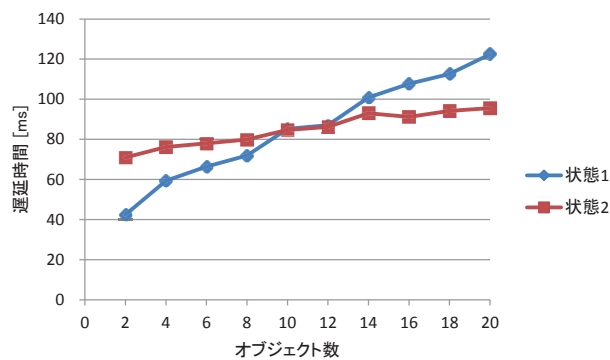


図7 遅延時間

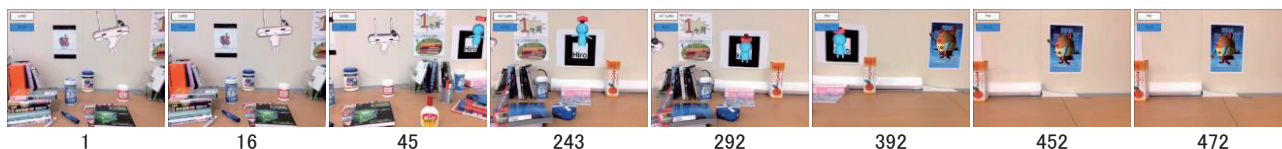


図8 動作確認のシーケンス

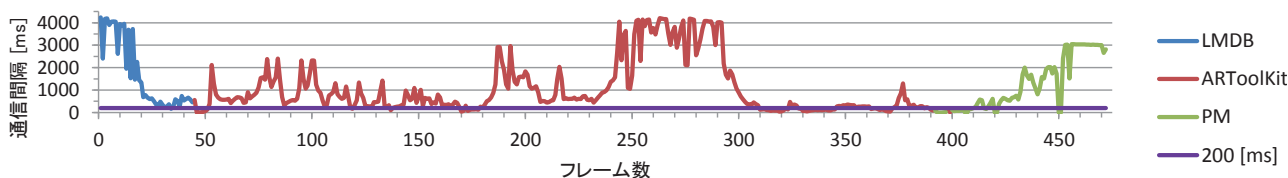


図9 通信間隔の変化

遅延時間は状態2でも増加しているが、状態1よりも増加量が少ない。ストリーミングサーバを利用する状態2では、描画するオブジェクト数が増加するにつれ、CGI画像生成に必要な処理性能も向上するからだと考えている。一方で、処理性能が低いモバイル端末でCGIを生成する状態1では、性能要求を満たせないと考えている。以上より、ストリーミングサーバを利用する有用性を確認した。

#### 【実験2】通信間隔変更の動作確認

実装したクライアントと通信間隔を決定する式をもとに、通信間隔が状況に応じて変更されていることを確認する。幾何位置合わせに用いた手法は[2]の動作確認と同様である。また、実験環境のうち特徴点が多数検出可能な一部領域に対して予めLMDBを構築した。この領域は、図8の1フレームから44フレーム目に該当する。

#### 【実験2の結果と考察】

実験2での画像シーケンスの一部とそのフレーム番号を図8に示す。また、そのシーケンスにおける通信間隔の変更結果のグラフを図9に示す。従来[2]は200[ms]で定期的に通信していたが、実装した決定式により、通信間隔が動的に変更していることを確認した。1フレームから16フレーム目、243フレームから292フレーム目、452フレームから472フレーム目の3箇所は、通信間隔が各幾何位置合わせ手法で特に長い。これらのフレーム間では、他のフレームと比べてクライアントが移動していない。そのため、幾何位置合わせ手法を切り替える必要がなく、通信間隔が長くなっている。一方、幾何位置合わせ手法が切り替わっている45フレーム目と392フレーム付近では、通信

間隔が短くなっている。

実験2において、クライアントと切替機構が通信した回数は72回である。図8のシーケンスは合計で29933[ms]であるため、従来通りの200[ms]間隔で定期的に通信した場合、150回程度通信すると推測できる。このことから、クライアントと切替機構の通信回数を半分以下にできることを確認した。

## 6. むすび

本稿では、我々が提案しているモバイルMRシステム構築のためのフレームワークにおける切替機構の導入、およびストリーミングサーバを設計・実装した結果について述べた。そして動作確認の結果から、ストリーミングサーバや切替機構を利用することの有用性を示した。今後の展望として、水面や影付きのCGオブジェクトといった高品質なCGIをストリーミングサーバで生成することで、計算資源が限られたモバイル端末で高品質なMR画像提示を実現することが考えられる。

**謝辞** 本研究の開発作業に携わった松田祐樹氏に感謝の意を表す。

#### 参考文献

- [1] 柴田他：“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の設計と実装”，日本VR学会論文誌，Vol. 19, No. 2, pp. 215 – 225, 2014.
- [2] 海津他：“クライアントサーバモデルに基づく携帯端末の位置姿勢推定機構(5)～位置合わせ手法の動的切替機構の開発～”，信学技報，Vol. 112, No. 385, PRMU2012-123, pp. 335 – 340, 2012