

# 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(5) — サウンド制御機構の設計と実装 —

## A Distributed Framework for Mobile Mixed Reality System (5) - Design and Implementation of Sound Control Mechanism -

前田貴裕, 柴田史久, 木村朝子, 田村秀行

Takahiro Maeda, 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:** We are aiming at the realization of a distributed framework for mobile mixed reality system. The purpose of this paper is design and implementation of sound control mechanism for this framework. In the past years, we have mainly focused our efforts on visual representation of our framework, so that we have proposed the script-based content control mechanism to control motion of virtual objects such as CG models, images, and so on. However, audio representation is of the same importance as visual representation when developers try to make rich applications. Therefore, we incorporate sound control mechanism into our framework. In order to realize it, we enhance the ability of the original script language we proposed. Consequently, application developers who use our framework can control sound components with equal way to other virtual objects easily.

**Key Words:** Mixed Reality, Mobile System, Framework, Content Control Mechanism

### 1. はじめに

我々はこれまで、モバイル複合現実感 (Mixed Reality; MR) システムのための機能分散型フレームワークの設計・開発を行ってきた[1]。本フレームワークは、サーバ・クライアント型のアーキテクチャを採用し、コンテンツをサーバが一元管理することによって、クライアント間で同一の MR 空間を共有している。MR 空間中に配置するコンテンツの動作は、我々の研究グループが独自開発したスクリプト言語[2]により制御する。スクリプト言語は JAVA に類似した言語であり、オブジェクトの位置や姿勢などのパラメータを、プログラムの的に操作することが可能となっている。このため、複雑な動作の制御が可能だが、これまで 3DCG や文字などの視覚的なデータを制御する機構の実現に傾注していた。

そこで本研究では、フレームワークの表現力向上を目指し、サウンド制御機構の設計・開発を行い、サウンドを扱

うためのスクリプト言語の設計を拡張した。

### 2. コンテンツ制御機構

#### 2.1 制御手法

本機構は以下の処理系によって構成される(図 1 参照)。

##### 【サーバ】

コンテンツを管理し、外部ファイルとして記述したスクリプトを解析実行することで、コンテンツ情報を更新する。スクリプトには、各オブジェクトの位置姿勢などのパラメータを変更するプログラムを記述しておき、解析結果を MR 情報としてサーバ自身に格納する。スクリプト解析を行うスクリプトエンジンはサーバのみが持ち、MR 情報を一元管理することで、全クライアントに同一の MR 空間を提供する。

##### 【クライアント】

通信開始時に、サーバからオブジェクトデータ (モデル、画像等) をダウンロードする。その後、サーバから取得した MR 情報とクライアント自身の位置・姿勢を基に、画像をレンダリングすることで、ユーザに対して MR 提示を行う。これにより、通信に必要な情報を削減している。

#### 2.2 コンテンツの構成

本フレームワークで扱うコンテンツは、クライアントとオブジェクトから構成される。オブジェクトは仮想オブジェクトと実オブジェクトの 2 つに大別される。仮想オブジェクトの構成要素であるコンポーネントは、3D モデル・画像・テキストを扱うことができる。これらのコンポーネントが図 2 に示すような階層構造をとることで、オブジェ

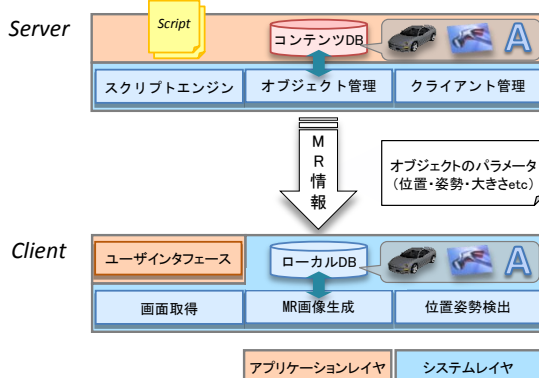


図 1 本フレームワークの機能配置図

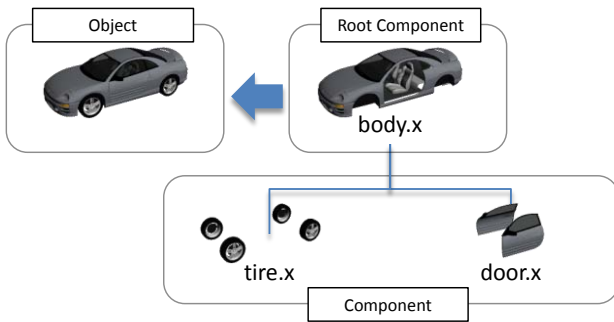


図2 仮想オブジェクトの構成要素

クトを構成する。これにより、1つのオブジェクトを操作するスクリプトを記述するだけで、同オブジェクトのコンポーネントをまとめて操作することが可能となる。

### 3. サウンド制御機構

#### 3.1 設計方針

フレームワークで扱うサウンドには様々な種類が考えられる。3次元空間中に存在するオブジェクトだけでなく、BGM やクライアントの操作音など、クライアント自身から発生するようなサウンドも想定する。サウンドの操作は新たな手段を導入するのではなく、従来のコンテンツと同様にスクリプトによって制御する。こうすることで3DCGと同様の操作が可能となり、容易な制御が実現できる。

#### 3.2 サウンドの導入方法

サウンドは図3に示すように、コンポーネントとして既存のオブジェクトに持たせることで、位置・姿勢をオブジェクトと同期させる。これにより、オブジェクトに対しサウンドに関するスクリプトを追加するだけで、オブジェクトと共に動くようなサウンドの表現が可能となる。サウンドデータをサーバからストリーミングダウンロードする形式も考えられたが、2.1節で述べた通信量の関係上、従来のコンポーネントと同様に扱う構造を採用した。

#### 3.3 サウンド表現方法

サウンドに関して、どのようなパラメータを扱うか考慮し、スクリプトの設計を行った。まず、サウンドを扱う上で最低限必要な操作は「再生」「停止」「一時停止」「繰り返し」「音量の調整」が挙げられる。また、単に音量を変化させるだけでなく、サウンドの位置・姿勢によって出力音を変化させるような表現が可能である。そこで、可聴範囲を設けることでクライアントとサウンド間の距離によって音の可聴・不可聴を切り替える表現が可能となる。しかし、可聴範囲の境界上付近でクライアントが停止した場合、再生と停止が連続して繰り返される可能性がある。そ

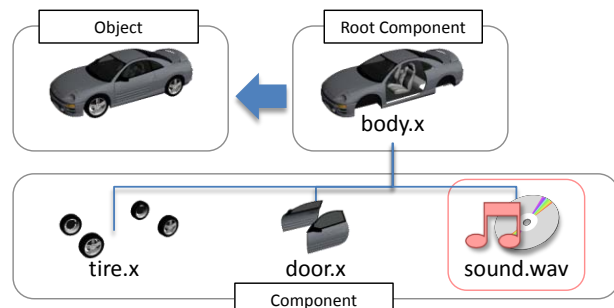


図3 仮想オブジェクトの構成図

こで、可聴範囲の境界線を2つに分け、内側に「再生開始範囲」、外側に「停止範囲」を用意する。これにより、再生開始範囲内に入った後、停止範囲外に出るまでに猶予ができ、再生後すぐに停止、または停止後すぐに再生されることを回避する。さらに、クライアントとサウンド間の距離から「距離による音量の減衰」を算出することで、サウンドから離れると聞こえる音量が小さくなるといった表現が可能となる。これにより、可聴範囲内でクライアントが移動する際、クライアントとサウンド間の距離に応じて音量を変化させるような表現が可能となる。また、サウンドの位置・姿勢を利用し、「指向性」を持たせることで、サウンドの向きに応じて音量の強弱をつけるような表現が可能となる。指向性の表現に際して、MSDN (Microsoft Developer Network) に記載されているサウンドコンモデル[3]を参考に、4つのパラメータを用意する(図4参照)。さらに、図5のように、サウンドが遮蔽物に隠れている場合を考慮し、遮蔽物となるオブジェクトに対し、音の通り抜け難さを表す「遮音度」を持たせる。これにより、遮蔽物に隠れているサウンドの音量が、遮音度によって変化するような表現が可能となる。

#### 3.4 サウンド再生処理

##### 3.4.1 使用するパラメータ

前節で述べたサウンド表現を行うために用意した変数を表1にまとめる。また、サウンドを扱うスクリプトの記述例を図6に示す。

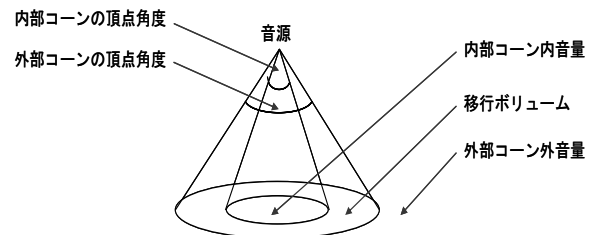


図4 サウンドコーン

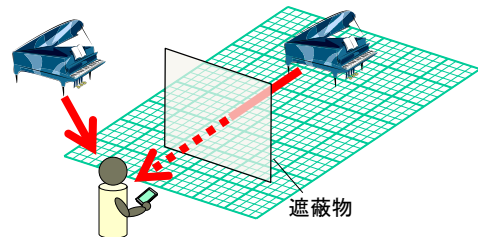


図5 遮蔽物とサウンド

表1 スクリプトにおける変数とパラメータの関係

型	変数名	パラメータ	用途
int	volume	$V$	音量
int	state	$S$	再生・停止・一時停止
int	loop	$L$	繰り返し再生数
int	playRadius	$R_{play}$	再生開始範囲
int	stopRadius	$R_{stop}$	停止範囲
int	attenuation	$a$	音量の減衰率
int	inVolume	$V_{in}$	内部コーン内の音量
int	outVolume	$V_{out}$	外部コーン外の音量
int	inAngle	$A_{in}$	内部コーンの頂点角度
int	outAngle	$A_{out}$	外部コーンの頂点角度
double	soundInsulationRate	$I$	遮音度

### 3.4.2 処理手順

ユーザに提示する最終的な出力音は、状態 (S) と、出力音量の2つによって決まる。その他のパラメータは出力音量を変化させる際の条件となる。クライアントはサーバから受け取ったパラメータと、サウンドの位置姿勢、クライアント自身の位置姿勢から、最終的な出力音量を導き出す。また、遮音度を持つオブジェクトがある場合は、そのオブジェクトの位置を保持しておく。

サウンド処理の流れを図7に示す。

- (a) サウンド・クライアント間の距離を算出  
 サウンド  $(x_2, y_2, z_2)$  とクライアント  $(x_1, y_1, z_1)$  の3次元位置座標から式 (1) を用いて距離  $(d)$  を算出する。

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

- (b) 再生開始範囲内かどうか判断  
 (a) で求めた  $d$  が  $d \leq R_{play}$  であれば (c) の処理を行い、 $d > R_{play}$  であれば、(a) の処理に戻る。
- (c) 距離を考慮した音量 ( $V'$ ) を算出  
 式 (2) に示すように、サウンドからの距離に対し  $a$  (%単位) を掛け、 $R_{stop}$  に対する割合を求め、それを  $V$  と掛けることで、距離を考慮した  $V'$  を求める。

$$V' = \frac{V}{100} \times \frac{R_{stop} - d \times \frac{a}{100}}{R_{stop}} \quad (2)$$

- (d) サウンドコーンを考慮した音量 ( $V''$ ) を算出  
 サウンドの正面方向を0度、背面方向を180度としクライアントの位置がどの方向にあるか算出する。図8に示すようにサウンドの正面方向のベクトルを  $u$ 、クライアントからサウンドへのベクトルを  $v$  とし、式 (3) を用い、サウンドに対し、どの角度にクライアントが位置し

ているか算出する。

$$\theta = 180 - \frac{(\arccos(u \cdot v) \times 180)}{\pi} \quad (3)$$

- i. 内部コーン内の場合  
 (d) で求めた  $\theta$  はサウンドの正面方向を  $0^\circ$  とし、背面方向を  $180^\circ$  とする値である。  $A_{in}$  は正面方向を  $0^\circ$  とし、背面方向へ左右に広がる  $360^\circ$  を最大値とする値であるため、 $\theta \leq A_{in}/2$  であれば、式 (2) で求めた  $V'$  に対して、 $V_{in}/100$  を掛けた音量を  $V''$  とする。
- ii. 外部コーン外の場合  
 内部コーン内の場合と同様に  $A_{out}$  は  $0 \sim 360^\circ$  の値であるため、 $\theta \geq A_{out}/2$  の場合、式 (2) で求めた  $V'$  に  $V_{out}/100$  を掛けた音量を  $V''$  とする。
- iii. 上記どちらにも当てはまらない場合  
 移行ボリューム ( $V''$ ) を求めるため、式 (4) を用いる。  $\alpha$  を  $A_{out}$  からの角度、 $\beta$  を  $A_{in}$  からの角度とし、 $\alpha$  と  $\beta$  の割合によって  $V_{in}$  と  $V_{out}$  を掛ける割合を算出する。

$$V'' = V' \times \left( \frac{\alpha}{\alpha + \beta} \times \frac{V_{in}}{100} + \frac{\beta}{\alpha + \beta} \times \frac{V_{out}}{100} \right) \quad (4)$$

$$\alpha = \frac{A_{out} - \theta}{2}$$

$$\beta = \theta - \frac{A_{in}}{2}$$

- (e) 遮蔽物を考慮した音量を算出  
 あらかじめ保持しておいた遮音度を持つオブジェクトの位置を基に、クライアントとサウンドを繋ぐベクトル上に、遮音度を持つオブジェクトが存在するか判定する。オブジェクトがある場合は、ベクトル上に存在する全てのオブジェクトの遮音度を  $V''$  に掛ける。
- (f) 状態に応じた処理を行う  
 $S$  が1なら再生、2であれば一時停止、3であれば停止処理を行う。再生時は (e) で求めた音量で、 $L$  の回数分再生を行う。
- (a) ~ (f) の処理を繰り返すことでクライアントはユーザに対し音を提示する。

## 4. 動作確認

### 4.1 確認手順

サウンドコーンと遮蔽物による音量変化を確認するため2種類の実験を行う。確認対象としてクライアントに実験中のパラメータを出力させ、そのログの分析を行う。

#### (1) サウンドコーンを設定したオブジェクトを回転

マーカ上に一定音量の音源を持たせた車のオブジェクトを示させ、鉛直上向き方向を回転軸とし、1秒毎に  $90^\circ$  ずつ回転させる。設定するサウンドコーンのパラメータを表2に示す。この場合、図9で示すようにサウンドコーンが広がる。クライアントの位置からオブジェクトへ伸ばしたベクトルと、オブジェクトの正面方向へ伸ばしたベクトルとのなす角は図10に示す通りそれぞれ約  $45^\circ$ 、 $90^\circ$ 、 $135^\circ$ 、 $90^\circ$  である。初期位置を (a) に設定し、回転させることで、(b) ~ (d) に示すようにオブジェクトの正

```

/* コンストラクタ */
public Object1(Position3D pos, Orientation ori) {
    /* 初期設定 */
}
/* 再生するメソッド */
public void start () {
    this.root.sound1.state = 1;
}
/* 無限ループさせるメソッド */
public void loop () {
    this.root.sound1.loop = 0;
}
    
```

図6 スクリプト記述例

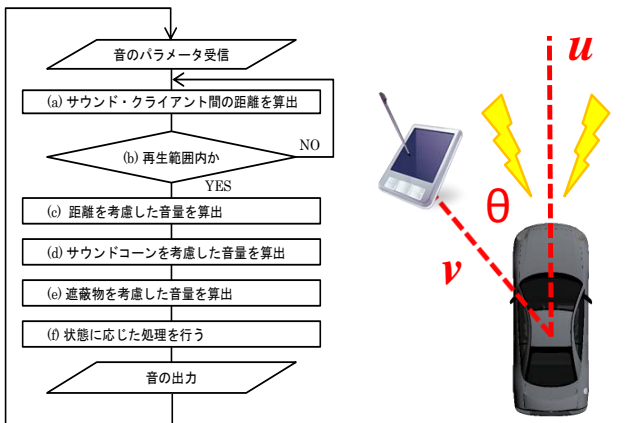


図7 サウンド処理の流れ



図8 内積による角度検出

表 2 サウンドーンの変数の設定値

inVolume	100
outVolume	0
inAngle	90
outAngle	270

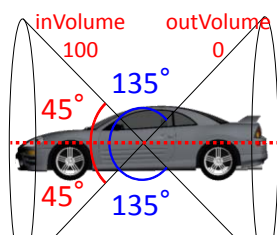


図 9 設定したサウンドコーン

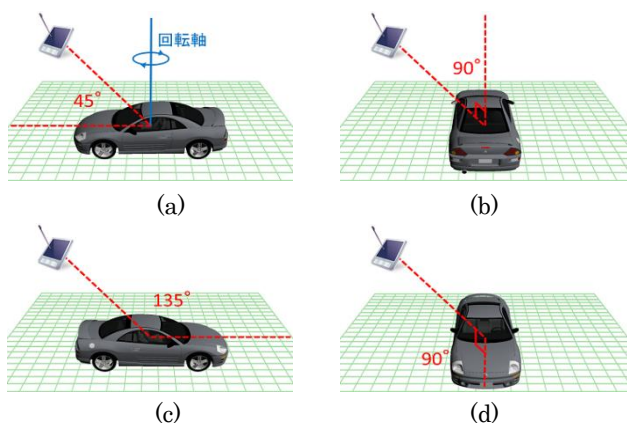


図 10 クライアントから見たオブジェクトの正面方向

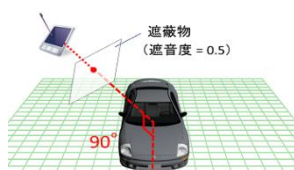


図 11 遮蔽物を考慮した動作確認

面方向を変化させる。

## (2) 動作確認 (1) に遮蔽物を配置

動作確認 (1) と同様の実験を行うが、(d) の方向時に、図 11 に示すように、クライアントとオブジェクトとの間に遮蔽物を配置する。この遮蔽物に遮音度 (0.5) を持たせ、クライアントに提示する音量を半減させる。その他の実験手順は (1) と同等である。

## 4.2 確認事項

### (1) サウンドコーンによる音量変化

inAngle が  $90^\circ$  であるため、車のオブジェクトの正面方向を頂点とし、頂点角度  $90^\circ$  の円錐形に inVolume が適用される内部コーンが広がる。よって (a) は、内部コーン内であるため volume の 100% の音量が提示され、同様に外部コーン外の (c) では音量は 0 となることを確認する。次に (b) と (d) は inAngle からの角度と outAngle からの角度の丁度中間に位置しており、移行ボリュームは volume の 50% の音量であることを確認する。また、移行ボリュームの計算が正しいことを確認するため、(a) ~ (d) 間の音量が連続的に変化しているか確認する。

### (2) (1) との音量の差分の確認

動作確認 (1) に比べ、(a) ~ (c) において提示される音量が同等であることを確認する。また (d) では、遮音度 (0.5) を持たせた遮蔽物を設置しているため、クライアントに提

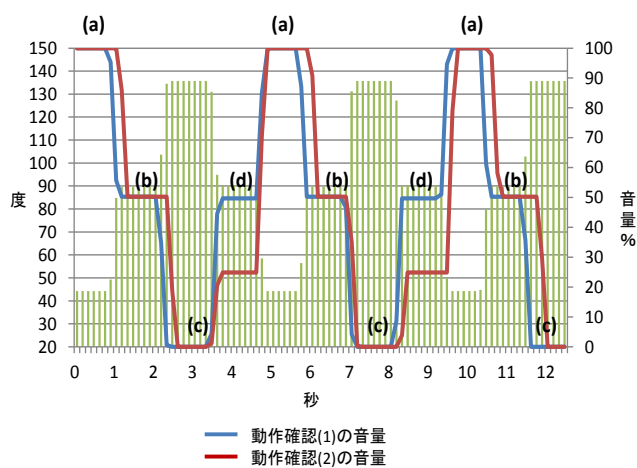


図 12 動作確認 (1) と (2) の結果

示する音量が半減する。よって、動作確認 (1) における (d) の移行ボリュームが volume の 50% である場合、さらに 50% を掛けた 25% となることを確認する。

## 4.3 確認結果

実験結果 (1) の音量を図 12 の青線部分に示す。緑の棒グラフは横軸の時間 [sec] におけるベクトルの角度差を示す。この図から、(a) の角度では volume の 100% 付近の音を確認した。(b) と (d) では volume の 50% 付近、(c) の角度では音量が 0 となったことを確認した。また、(a) ~ (d) 間の移行ボリュームは連続的に推移しているため、移行ボリュームの計算結果が正しいことを確認した。これらの結果により、サウンドコーンによる音量変化が正しく動作することを確認した。

実験結果 (2) の音量を図 12 の赤線部分に示す。この図から (a) ~ (c) では動作確認 (1) とほぼ同等の結果を示しており、(d) において、動作確認 (1) に比べ、音量が半減した 25% 付近の音量を確認した。この結果から遮蔽物体による音量変化が正しく動作することを確認した。

## 5. むすび

前節の実験結果から、実装したサウンド制御機構が正しく動作していることを確認した。この結果からフレームワークの表現力が向上し、アプリケーション開発の幅が広がったと言える。しかし遮蔽物を考慮した音量変化の際、隠れているか否かという判断により処理を行っているため、急な音量変化が発生する。今後は遮蔽面積を考慮した音量変化機構の導入を目指す。

## 参考文献

- [1] 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-”，第 14 回日本バーチャルリアリティ学会大会論文集，3A2-3，2009。
- [2] 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-”，第 15 回日本バーチャルリアリティ学会大会論文集，pp. 374 - 377，2010。
- [3] サウンドコーン：  
<http://msdn.microsoft.com/ja-jp/library/cc370760.aspx> (2011 年 7 月 21 日)