

# 全方位センサによる分散映像データベース のアーキテクチャとプロトコル設計

## Architecture and Protocol of Distributed Video Databases with Omni-directional Sensors

何 書 勉<sup>†</sup> 横 田 裕 介<sup>††</sup>  
石 黒 浩<sup>†††</sup> 上 林 弥 彦<sup>††</sup>

SHUMIAN HE,<sup>†</sup> YUSUKE YOKOTA,<sup>††</sup> HIROSHI ISHIGURO<sup>†††</sup>  
and YAHIKO KAMBAYASHI<sup>††</sup>

全方位センサを取り付けた DV カメラを複数個使うことによって、ミーティングなどの状況を 3 次元的に保存することができる。これによって保存された 3 次元空間内を移動しながら再生することが可能になる。我々はこれらの映像や音声データを効果的に利用するために、コンテンツデータベースサーバ・コントロールサーバ・クライアントマシンによるアーキテクチャとネットワークプロトコルを設計した。全方位センサの切り替えやユーザの操作による再生時の遅延を解消するために、コントロールサーバ内において、ユーザの動作を予測した上のバッファリング処理方式を開発した。これを使って、クライアントマシンとコントロールサーバとの通信の簡単化・標準化も図っている。また、SQL-like なクエリー文を使って、映像データベース内から映像データを検索する方法を考案した。

### 1. はじめに

我々が現在進めている Retrax プロジェクトでは、全方位センサによる新しい仮想空間の構築を課題としている。Retrax とは “Re-tracks” をもとにした造語で、過去を再び辿ることを意味する。データベースにできるだけ多くの情報を保存して、その中から必要なデータだけを選び出す、すなわち “store-everything select-later” 原則に基づいて、過去のミーティングに参加できるようにシステムが構築されている。本プロジェクトの目標は、過去の記録にアクセスすることで、過去と現在とのコミュニケーションを実現することである。

本研究では、映像を記録する手段として全方位セン

サ<sup>1)</sup> (Omni-directional Sensor) を利用している。全方位センサ (図 1) は、単一のカメラで 360 度視野角のある画像または映像を撮るものである。全方位センサにより得られたドーナツ型のデータをパノラマ形式に変換することで (図 2)、360 度視野角のある画像や映像が得られる<sup>2)</sup>。現在本システムで使っている全方位センサは、著者の 1 人である石黒浩により開発された。



図 1 全方位センサ

Retrax システムには、ユーザが空間内で前進・後退・自由な視点移動などの動きや、早送り・逆再生・

<sup>†</sup> 京都大学工学部

Faculty of Engineering, Kyoto University  
shumian@db.soc.i.kyoto-u.ac.jp

<sup>††</sup> 京都大学情報学研究所社会情報学専攻

Department of Social Informatics, Graduate School of Informatics, Kyoto University  
{yyokota,yahiko}@i.kyoto-u.ac.jp

<sup>†††</sup> 和歌山大学システム工学部情報通信システム学科

Department of Computer & Communication Sciences,  
Wakayama University  
ishiguro@sys.wakayama-u.ac.jp

再生・バック・停止などの再生支援を実現するためのユーザインターフェイスが標準装備されている(図3)。また、仮想空間中のユーザと各全方位センサ間の距離に応じて、音量を調節してミックスした音声データを生成し、臨場感を生み出す。ユーザはこのようなインターフェイスを通じて、仮想空間内でのウォークスルーが可能である。本論文は、ユーザの操作を解析して、コンテンツデータベースサーバから映像と音声を取り出すためのアーキテクチャ構成と内部処理の仕組みを説明する。

Retrax システムの具体的な利用例の一つとしては、ヴァーチャルパーティが考えられる。複数の全方位センサを実空間のパーティ会場に配置してシステムを接続し、その空間を自由にウォークスルーできるシステムを考える。リアルタイムシステムであれば、実際の会場にいる参加者とコンタクトを取ることにも可能である。時間的、地理的要因によりパーティに参加できない人をシステムを通して参加させるという試みは、CGを用いた仮想空間ではなく、実映像を用いたシステム上でのみはじめて現実的となる。パーティに実際に参加した人であっても、後にこのシステムを利用して、自分がそのときいなかった場所で交わされていた会話を聞くことも可能である。

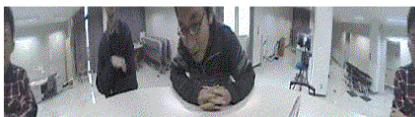
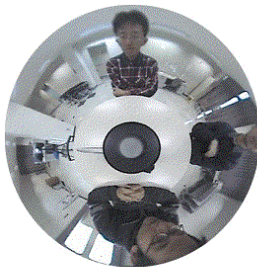


図2 ドーナツ型画像とそれを展開して得たパノラマ画像

我々は、記録された空間内の膨大な音声映像データを保存・検索・再生するために、独自のシステムを設計した。このようなシステムを実現するため、通信プロトコルや分散データベースの設計が必要となる。ユーザの操作に応じて映像や音声を取り出し、展開して表示する動作は非常に時間がかかるため、遅延が発生する可能性がある。これに対処するために、バッファリング処理を導入した。

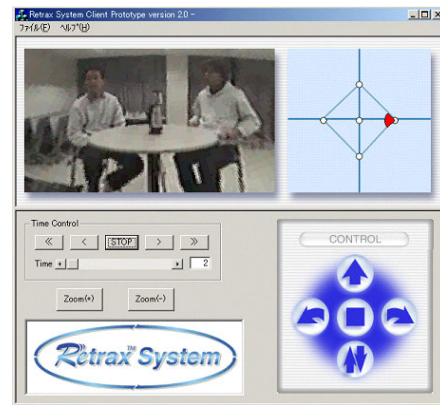


図3 Retrax システムの GUI

本アーキテクチャにおいて、Retrax システムはあくまでも分散映像データベースの応用の一つと位置付けている。分散映像データベースの汎用的な利用を実現するため、SQL 構文を参考にした RSQL (Retrax SQL) という質問言語を定義し、そのクエリーに基づいてデータを検索するためのプロトコルを設計した。これによって、映像データベース内から映像や音声を検索することが可能になる。将来的には映像データベースを使って仮想空間を構築するだけでなく、テレビ中継や音声画像情報の管理およびラジオ放送での活用が可能になると考えている。

なお、本論文は以下のように構成される。2章ではシステムのアーキテクチャについて説明する。3章では高速化するためのバッファリング処理について述べ、4章では質問言語及びその解析を説明する。5章では先読みとバッファリング処理の有効性を検証するためのシミュレーション実験を説明し、6章では関連研究について言及し、7章と8章ではそれぞれ残された課題及び今後の展望と結論について述べる。

## 2. システムアーキテクチャ

Retrax システムはクライアントマシン、コントロールサーバ、コンテンツデータベースサーバ群により構成されている(図4)。クライアントマシンから映像や音声を表示するためのクエリーを出して、コントロールサーバがそれに応じてコンテンツデータベースサーバから映像や音声を取り出し、クライアントマシンへ送る。

全方位センサは三脚に固定された DV カメラに取り付けられており、空間の中心に1台、その周りを囲むように複数台等間隔に配置する。それぞれの全方位センサの高さは合わせておく(図5)。1つの全方位センサに対して、複数の指向性マイクをつけると同じ位置

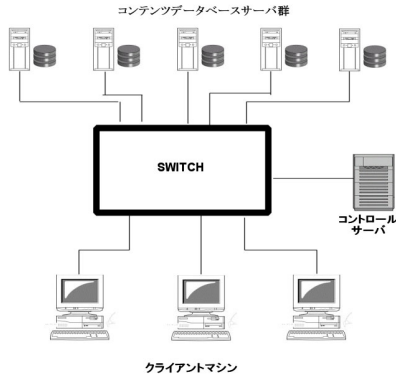


図 4 システムアーキテクチャ

でも向きによって聞こえる音が違う状況を生産できるが、現状のシステムでは1つのセンサに対して1つの無指向性マイクを対応させている。

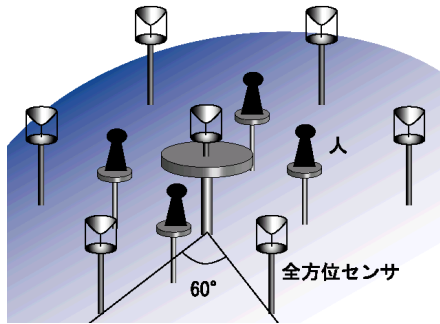


図 5 全方位センサの配置例

次に、各種のサーバ及びクライアントマシンについて説明する。

**コンテンツデータベースサーバ群** 複数のサーバにより構成され、全方位センサにより撮影された映像と音声格納される。全方位センサ一台につきサーバ台を用意する。サーバ内には、複数の会議や空間のコンテンツを格納することができる。ただし、同じ位置にある全方位センサで記録した音声と映像は一台のサーバに格納されるものとする。複数のコンテンツには、コンテンツ名でインデックスを付けて管理する。コントロールサーバから来たクエリーを受け付け、指定された全方位センサによる映像の開始時刻及び終了時刻に応じて、映像や音声をコントロールサーバへ返す。

**コントロールサーバ** クライアントマシンから送られてきたクエリーを解析し、取り出すべきコンテンツ

の種類(音声か画像か)、コンテンツの格納場所(データベースサーバの IP アドレス)、映像の開始時刻及び終了時刻を判断して、相応するコンテンツデータベースサーバへクエリーを出す。また、コンテンツデータベースサーバから受信した映像と音声を、クエリーを出したクライアントマシンへ送り返す。

**クライアントマシン** クライアントマシンは空間内におけるユーザの位置と移動方向のベクトルを管理し、コントロールサーバに対し必要なコンテンツを要求するクエリーを出す。また、そのクエリーに基づいてコントロールサーバから映像と音声のコンテンツを受信する。

クライアントマシンにはバッファ領域が用意され、ユーザの操作に応じて、映像音声コンテンツの先読みに用いられる。これについては、次章において説明する。

受信された映像データはドーナツ型の映像となっているため、クライアント上でパノラマ形式の映像に展開し、表示する。また、クライアントは音声をミックスする機能も持つ。音声は全てのコンテンツデータベースサーバから受信され、ユーザが仮想空間にいる位置と各全方位センサ間の距離に応じて、各全方位センサで記録された音の大きさを調節してミックスし、ユーザの位置で聞こえる音を合成する。例えば2つのマイクの間にいると両方の音が混じり合う。

### 3. バッファリング処理

ユーザの操作に応じて映像や音声を取り出し、展開して表示する操作は非常に時間がかかるため、遅延が発生する可能性がある。これに対処するために、バッファリング処理を導入した。これによって以下のような効果が得られる<sup>3)</sup>。

- 不安定・低速な転送回線にも対応する
- 受信したデータをそのまますぐに再生せず、キューイングを行う
- データ落ちなどにも対応する

#### 3.1 アルゴリズム

##### 3.1.1 パラメータの定義

以下のパラメータを定義する。

定数

$N_s$  コンテンツデータベースサーバ群に蓄積されている利用可能なコンテンツの数

$N_o$  全方位センサの個数

$X_j$  全方位センサ  $j$  の  $x-y$  座標系の仮想空間内における  $x$  座標 ( $1 \leq j \leq N_o$ )

$Y_j$  全方位センサ  $j$  の  $x-y$  座標系の仮想空間

- 内における  $y$  座標 ( $1 \leq j \leq N_o$ )
- $v$  ユーザが空間内で移動する速度
- $\tau$  時間的先読みの閾値
- $\sigma$  空間的先読みの閾値
- 変数
- $C_k$  コンテンツサーバに蓄積している各コンテンツに付けるユニークな番号  
( $1 \leq C_k \leq N_s$ )
- $x$  クライアントマシンを操作するユーザの  $x-y$  座標系仮想空間内における  $x$  座標
- $y$  クライアントマシンを操作するユーザの  $x-y$  座標系仮想空間内における  $y$  座標
- $\phi$  クライアントマシンを操作するユーザの仮想空間内の移動角度
- $ts$  クライアントマシンを操作するユーザが指定した映像の取得開始時刻
- $te$  クライアントマシンを操作するユーザが指定した映像の取得終了時刻
- $c$  現在使われている全方位センサの番号

バッファの中では時間と空間に基づいて音声と映像のストリームを管理する．全方位センサ  $j$  と  $ts$  で必要なストリームが一意的に決まり，それを領域内に格納する．

バッファ内容は，メモリ内に管理される．バッファ領域の中のストリームは，メモリの中で双方向リストで管理される．より先に格納されたバッファ内容は，リストのより先頭に現れ，直前に利用されたバッファ内容はリストの後部に現れる．バッファの領域がいっぱいになった場合，常にリストの先頭の節点が保持しているストリームを破棄する．

### 3.1.2 時間的先読み

クライアントマシンがユーザの操作に応じて，映像や音声を取得するためのクエリーをコントロールサーバへ出す．クエリーの中には， $x, y, \phi, c, ts, te$  の情報が含まれる．コントロールサーバはこれらの情報で所在位置や進行角度などユーザの属性と，取得する映像や音声の開始時刻や終了時刻など再生の属性が分かる．

現在のシステムでは，初期状態はユーザが空間の真ん中に立ち， $ts = 0$  から再生を始めることとなっている．ユーザがシステムにログインして，コンテンツ（空間）を選択したら，クライアントマシンではすぐにバッファ領域の確保などのバッファリング処理を開始する．真ん中の全方位センサで記録された映像と音声のストリームを時間的先読みの閾値  $\tau$  だけコンテン

ツデータベースサーバから取得し，バッファ領域に格納する．メモリ内のリストの先頭にそのストリームの場所を記録し，最初の節点を生成する．

クライアント側で再生を始めると， $c$  と  $ts$  を使ってバッファ領域から再生すべきストリームを検索し，再生する．同時に，全方位センサ  $c$  で記録した時間  $ts + \tau$  から  $\tau$  までの間の映像や音声をコントロールサーバ経由でコンテンツデータベースサーバから取得し，バッファ領域に格納し，メモリ内のリストの先頭にそのストリームの場所を記録し，次の節点を生成する．

クライアントマシンがユーザの操作を受け付けた後，最初に行う操作はバッファ領域から蓄積されているストリームを検索することである．コンテンツ一個あたりのバッファ領域サイズ  $s$  の大きさが十分であるなら，多数の時間帯の映像と音声はバッファ内に保持され，再生すべき映像や音声はバッファ内から取得できるため，コンテンツデータベースサーバとの通信データ量が減る．

### 3.1.3 空間的先読み

空間的先読みは，ユーザが全方位センサを切り替える直前に発生する． $x, y, \phi, c$  などの情報から，ユーザの位置と現在の全方位センサの番号が分かる．また，ユーザの全方位センサ  $j$  との距離  $a_j$  は

$$a_j = \sqrt{(X_j - x)^2 + (Y_j - y)^2}$$

ただし，( $1 \leq j \leq N_o$ )．

$a_j$  から最小となる  $a$  を求める．

$$a = \min \{a_1, a_2, \dots, a_{N_o}\}$$

この  $a$  が，ユーザに一番近い全方位センサ  $c$  との距離である．

空間的先読みが発生する条件は， $a \leq \sigma$  である．

ユーザが空間内で移動する速度  $v$  が一定だとすると，空間的先読みは，全方位センサ  $c$  に対し，時刻  $ts + a/v$  から  $ts + a/v + \tau$  の間の映像と音声をコンテンツデータベースサーバから取得する．

## 4. 通信プロトコル

分散映像データベースを実現するには，効率的なネットワークの利用が必須である．ネットワーク機能を最大限に発揮するために，クライアントマシン，コントロールサーバ，コンテンツデータベースサーバ間の通信プロトコルを設計した．このプロトコルは，以下の部分より構成されている．

### 4.1 RSQL

RSQL (Retrax SQL) は，我々が定義した分散映像データベースより映像と音声を取得するための質問

言語である .RSQL を使って、分散映像データベースとアプリケーションを分けて構築することが可能となる .RSQL は、SQL の SELECT 構文に映像の再生に必要な動きを記述する文法を追加し、拡張する形で設計されている .

ユーザがクライアントマシンを使って空間内でウォークスルーをする時、クライアントマシンはユーザの仮想空間内の位置座標、進行方向、取得する映像や音声の開始および終了時刻などの情報をもとに、RSQL を生成してコントロールサーバに送信する .

RSQL は、コンテンツデータベースサーバと通信するための標準記述であり、Retrax のクライアント以外のアプリケーションを開発する際に、RSQL を使って簡単に問い合わせできるのが最大の利点である .

#### 4.1.1 定義

RSQL は以下のように定義される .

```
SELECT *|AUDIO|VIDEO
FROM space_references
WHERE CAMNO, TIME[,X,Y,ANGLE]
[ORDER BY TIME [ASC |DESC]]
[SPEED_definition]
```

RSQL 特有の識別子は以下のとおりである .

AUDIO	音声だけを取得する
VIDEO	映像だけを取得する
ANGLE	現在のユーザが向いている方向
X	現在のユーザ位置の X 座標
Y	現在のユーザ位置の Y 座標
TIME	抽出する映像及び音声の時刻
CAMNO	抽出する映像を撮影する全方位センサの番号
SPEED	再生時の倍速

SELECT の対象は、AUDIO、VIDEO、\* である . \* は、AUDIO と VIDEO を同時に取得するという意味である .

システムが起動するとき、コンテンツデータベースサーバに格納されている利用可能なコンテンツ(会議)の一覧が表示され、ユーザがその中から一つを指定する . FROM の対象は、この選択されたコンテンツである .

WHERE 成分の要素は、X、Y、ANGLE、TIME、CAMNO の 5 種類が用意されている . TIME、CAMNO は必須要素で、他のものは Retrax システムにおいてバッファリング処理するために必要なパラメータである .

ORDER BY の対象は、TIME だけである . デフォルト

は ASC(昇順)で、時間軸に沿って再生する . DESC(降順)の場合は逆再生となる .

SPEED は再生時の倍速で、デフォルトは SPEED=1 である . SPEED>1 の場合は早送りや逆再生で、0<SPEED<1 の場合はスロー再生である .

#### 4.1.2 利用例

ユーザが参加している会議は WISE2001、会場内の位置は (100,200)、向かっている方向は 45 度、表示すべき映像は 2 番全方位センサで撮られたとする . この場合、X = 100、Y = 200、ANGLE=45、CAMNO=2 であると分かる .

i) コマンド：時刻 15.5 秒から映像と音声を再生する

```
RSQL: SELECT * FROM WISE2001
WHERE X = 100
AND Y = 200
AND ANGLE = 45
AND CAMNO = 2
AND TIME >= 15.5
```

ii) コマンド：時刻 15.5 秒と 20 秒の間の映像だけを再生する

```
RSQL: SELECT VIDEO FROM WISE2001
WHERE X = 100
AND Y = 200
AND ANGLE = 45
AND CAMNO = 2
AND TIME BETWEEN 15.5
AND 20
```

iii) コマンド：時刻 20 秒から映像だけを逆再生する

```
RSQL: SELECT AUDIO FROM WISE2001
WHERE X = 100
AND Y = 200
AND ANGLE = 45
AND CAMNO = 2
AND TIME <= 20
ORDER BY TIME DESC
```

iv) コマンド：時刻 20 秒から音声だけを 2 倍速で早送りする

```
RSQL: SELECT AUDIO FROM WISE2001
WHERE X = 100
AND Y = 200
AND ANGLE = 45
AND CAMNO = 2
```

```

        AND TIME >= 20
        SPEED = 2

```

- v) コマンド：時刻 15 秒から映像と音声を 3 倍速で逆再生する

```

RSQL :  SELECT * FROM WISE2001
        WHERE X = 100
        AND Y = 200
        AND ANGLE = 45
        AND CAMNO = 2
        AND TIME <= 20
        ORDER BY TIME DESC
        SPEED = 3

```

以上は Retrax システムのクライアントマシンを使った仮想空間内のウォークスルー時の例である。

以下は、分散映像データベースサーバを使ったテレビ中継の応用例である。

- vi) コマンド：DEWS2002 会場内の 3 号全方位センサで撮った映像の中から 200 秒からの映像と音声を再生する

```

RSQL :  SELECT * FROM DEWS2002
        WHERE CAMNO = 3 AND
        TIME >= 200
        SPEED = 3

```

- vii) コマンド：DEWS2002 会場内の 2 号全方位センサで撮った映像の中から 200 秒からの映像だけを 0.5 倍速でスローで再生する

```

RSQL :  SELECT VIDEO FROM DEWS2002
        WHERE CAMNO = 2 AND
        TIME >= 200
        SPEED = 0.5

```

## 4.2 クエリー解析及び通信状態

### 4.2.1 クライアントとコントロールサーバ間の通信

クライアントマシンからコントロールサーバへ送られた RSQL クエリーには、クライアントの現在の状態と映像を取り出すための情報が入っている。コントロールサーバからクライアントマシンへ送るものは、クエリーを受理した応答と映像および音声コンテンツである。

コントロールサーバはクライアントから受信した RSQL の中の CAMNO と TIME のパラメータと

VIDEO か AUDIO のメディア種類に応じて、該当する全方位センサで記録した映像や音声のストリームをクライアントへ返す。各クライアントマシンとコントロールサーバ間の音声専用ポートをそれぞれ 1 つ用意する。クライアントは受信した各全方位センサで記録した音声を、仮想空間中のユーザと各全方位センサ間の距離に応じて、音量を調節してミックスした音声データを生成する。

### 4.2.2 コントロールサーバとコンテンツデータベースサーバ間の通信

コントロールサーバがコンテンツデータベースサーバへ、断続的に get クエリーを送り、映像と音声コンテンツを取得する。コンテンツデータベースサーバは get クエリーに応じて、断続的な受理応答と連続的なストリームをコンテンツサーバへ返す。

RSQL はクライアントとコントロールサーバ間のクエリー形式であるのに対し、コントロールサーバとコンテンツデータベースサーバ間の通信のクエリーはよりシンプルなもので、get と send メソッドにより構成される。

get と send クエリーの構文は以下のように定義される。

```
get|send c v|a ts te
```

各識別子の意味は以下のとおりである。

get	コンテンツを取得するためのクエリーを示す
send	コンテンツを送信開始するためのクエリーを示す
c	コンテンツを記録した全方位センサの番号
v	映像 ( Video ) を取得する
a	音声 ( Audio ) を取得する
ts	取得するコンテンツの開始時刻
te	取得するコンテンツの終了時刻

4.1.2 節に示した i) の RSQL を解析する動作を以下に示す。RSQL の中で、CAMNO と TIME は映像を取り出すための必要パラメータで、X, Y, ANGLE はコントロールサーバにおける先読み処理に必要なパラメータである。

ここで、3.1.1 節で定義したパラメータを使って、 $x = 100$ ,  $y = 100$ ,  $\phi = 45$ ,  $te = 15.5$  が分かる。

時間的先読み閾値  $\tau = 0.5$  ( 0.5 秒の先読み ) とすると、2 番の全方位センサで記録した初期時点から 16

秒 から 16.5 秒 の間だけ映像と音声を、コントロールサーバから取得するために、get メソッドを使って 2 番の全方位センサが記録したコンテンツが格納されているコンテンツサーバへ以下のようなクエリーを送信する。

```
get v 16 16.5 (映像を取得)
get a 16 16.5 (音声を取得)
```

該当するコンテンツサーバが上記のクエリーを受信して、映像と音声を取り出し、ストリーム形式でコントロールサーバへ返す。ストリームのヘッダに、コンテンツの種類、コンテンツを記録した全方位センサの番号と開始および終了時刻を含めた文字列を付ける。

```
send 2 v 16 16.5 (映像を送信)
send 2 a 16 16.5 (音声を送信)
```

コントロールサーバは受信したストリームのヘッダにあるこの文字列内の全方位センサ番号、コンテンツ種類、開始時刻に応じて、バッファリング処理を行う。

### 5. 実験

提案したバッファリング処理の有効性を証明するため、シミュレーション実験を行った。バッファリング処理の有無や時間的・空間的先読みがシステムパフォーマンスに与える影響を、遅延時間、CPU 効率など、いくつかの尺度を用いて測定した。

#### 5.1 概要

Retrax システムのクライアントマシンの動作に基づき、Microsoft 社の Visual Basic 6.0 を使ってシミュレータを開発した(図 6)。シミュレータソフトは以下の機能をシミュレートできる。



図 6 シミュレータの実行画面

- 再生、停止、早送りなどの再生機能による動作の

$$15.5 + 0.5 = 16$$

$$15.5 + 0.5 + 0.5 = 16.5$$

切り替え

- 前進、後退などのユーザの進行方向の選択  
受信するデータの種類について、以下の項目が選択できる。

- 映像・音声両方
- 映像のみ
- 音声のみ

バッファリング処理について、以下の方式が選択できる。

- バッファリング処理なし
- 時間・空間的先読み
- 空間的先読みのみ
- 時間的先読みのみ

また、情報表示として、ユーザが仮想空間内にいる位置座標や送信中の全方位センサの ID、メモリと CPU の使用率、転送レート、反応(遅延)時間などが装備されている。今回行った実験はバッファ領域変化実験と空間的先読み実験である。それぞれのログを取るために、専用のボタンが設けられている。

シミュレーションは以下のような場面を想定している(図 7)。

- 対角線の長さが 5 メートルの正方形の 4 つの角と中心にそれぞれ全方位センサ一台を設置する。
- ユーザは隣り合う 2 台の全方位センサを結ぶ線分上を移動できる。
- 実空間上の 1cm の距離を、仮想空間上では単位 1 の座標距離に換算する。中心にあるセンサの座標を (0, 0)、一番左のセンサの座標を (-500, 0)、一番右のセンサの座標を (500, 0) とする。三者を結ぶ直線を  $x$  軸とする。ここで、実空間上の 1cm の移動距離を、仮想空間上では単位 1 の座標距離に換算する。

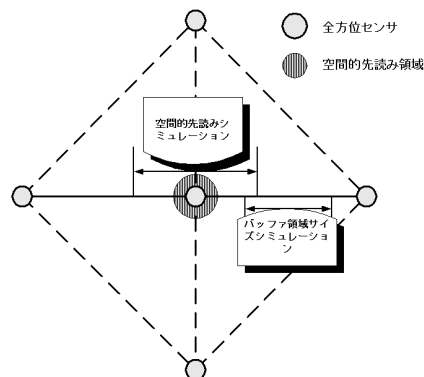


図 7 シミュレーションの仮想移動範囲

最終段階では、複数のユーザが同時にシステムをインターネット経由で利用することを目標としているが、今回行った実験では、シングルユーザが LAN 環境で使用するという特定の条件下での操作を想定し、ユーザが  $x$  軸上で移動することを前提とする。

現在の Retrax システムのクライアントソフトを解析することによって、以下の定数を実験で使うこととなった。

定数	値
映像データ転送率 (KBPS)	3705
音声データ転送率 (KBPS)	172
ネットワーク通信速度 (KBPS)	12800
ドーナツ型画像を展開する処理速度 (KBPS)	18500
ユーザの移動速度 (cm/s)	50

### 5.2 バッファ領域変化実験

バッファリング処理の目的は、遅延の発生を抑制することである。再生可能なデータをバッファ領域に格納することにより、再生するデータが確定してからコンテンツデータベースサーバから取得するのではなく、バッファ領域から取得する。バッファサイズが小さいなら、バッファ領域から映像データを取得する際に発生するミスヒット率が高くなり、遅延が発生するのにに対し、サイズが大きくなるにつれバッファ領域内のデータを取得及び破棄するには手間がかかり、CPU 使用率が高くなる。本実験を通じて、バッファ領域の大きさと遅延時間及びシステム全体の CPU 使用率との関係を示す。

図 7 で示した空間の中で、ユーザーは線分上で (100, 0) から (300, 0) へ移動するとする。時間的先読みだけを行い、バッファ領域のサイズを 0KB から 4MB まで 400KB ずつ増やし、それぞれの時点での遅延時間と CPU 使用率を算出し、グラフで示す (図 8, 図 9)。

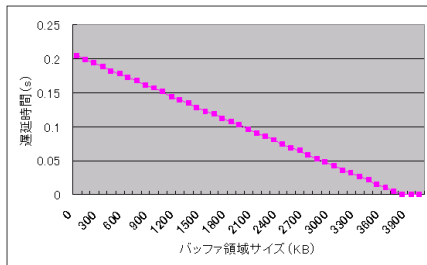


図 8 バッファ領域サイズと遅延

実験でシミュレーションするのはユーザが一定の速

度を保ちながら直線上で移動するという単純な動作であり、先読みは常に中心のセンサで記録したコンテンツの再生開始時刻から順次データを取得してバッファ領域に格納する。バッファ領域が 0 の場合、全てのデータをユーザが操作してからコンテンツサーバから取得するから遅延時間は最大となり、0.21 秒に達する。バッファ領域が大きくなるにつれ、ミスヒット率が下がり、遅延時間も減る。バッファ領域が 3900KB に達すると、遅延時間が 0 になり、映像と音声スムーズに再生される。

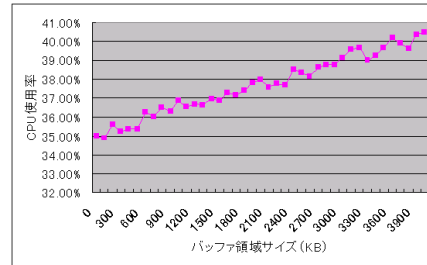


図 9 バッファ領域サイズと CPU 使用率

バッファ領域が大きくなると、ネットワーク領域内のバッファ内容を管理するために CPU 利用率が高くなる傾向を示している。遅延時間が 0 となる 3900KB を超えてさらにバッファ領域を拡大することは CPU 利用率が高くなるため、不必要であるといえる。

### 5.3 空間的先読み実験

空間的先読みは全方位センサの切り替え時に発生する。

5.2 節に示したデータにより、時間的先読みの閾値  $\tau$  を 0.21 秒、バッファ領域サイズを 3900KB とする。また、ユーザの移動速度  $v = 50 \text{ cm/s}$  より、空間的先読みの閾値  $\sigma$  を  $\tau v$  より大きい値の  $15 \text{ cm}$  とする。

図 7 で示した空間の中で、ユーザーは線分上で (-50, 0) から 50, 0 へ移動するとする。空間的先読みの発生範囲は、 $-15 \leq x \leq 0$  である。ユーザが (-15, 0) に進むと、現在再生されている映像を記録した全方位センサ (-500, 0) に対して時間的先読みを続けると同時に、進行方向に存在する、次に使われそうな全方位センサ (0, 0) に対し時間的先読みを始めるため、時間的先読みのプロセス数が倍増し、その分の CPU 使用率が高くなる (図 10)。センサの切り替え地点は (0, 0) で、それから空間的先読みが終了する。



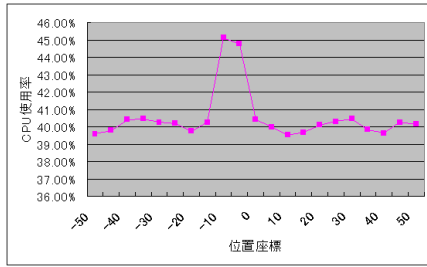


図 10 空間的先読み

## 6. 関連研究

### 6.1 動画像データベース

京都大学情報学研究科田中研究室では、動画像データベースの研究が進んでいる。動画像データの内容記述を行うために、動画像データの時区間の包含関係にもとづいて属性情報を共有できるデータモデルとデータベースシステム OVID<sup>4)</sup>を開発している。また、最近では、動画像データの内容記述を World Wide Web 上で公開された環境で行うシステムの研究開発にも着手している。

本研究との違いは、動画像データの検索などがメインで、仮想空間の構築を目的とはしていないことにある。

### 6.2 パーチャル番組制作環境システム

NHK が開発している番組制作を支援仮想スタジオである<sup>5)</sup>。実写映像と CG を効率よく組み合わせて番組を制作できるように支援する作業環境システムであり、仮想環境の中で、過去の番組で利用した素材を組み合わせてより自由に番組制作を行う。

このシステムでは、アーカイブ化された過去の映像に対して、自由にアクセスできるのが特徴である。

### 6.3 視点を 360 度全方位に移動できる映像ソフト

ソニーは視聴者が視点を 360 度全方位に移動できる映像ソフトを制作する事業 “FourthVIEW” を開始したと発表した<sup>6)</sup>。同社では、360 度全方位を同時に撮影できる特殊ビデオカメラ “FC1-V8” を開発した。このカメラを使って、再生時に視聴者が 360 度全方位から任意の視点をリアルタイムで選択できる映像ソフトを制作する。

NHK とソニーが開発したシステムが本研究との違いは、前者は画像編集に重んじていて、本研究は過去の映像データを使って仮想空間を構築することである。

## 7. 今後の課題

Retrax System の実装は大規模なものとなるため、

段階を追った開発を考えている。しかし、映像の圧縮や転送速度の問題など、技術的な課題は少なくない。ここでは、現状の技術と照らし合わせながら、それらの課題について述べる。

### 7.1 映像および音声の圧縮技術

映像および音声の圧縮では、MPEG(Moving Picture Experts Group) が有名である。Retrax システムの趣旨にも関係するが、リアルさを求めた場合、MPEG2 が適当であるといえる。しかし、MPEG2 により圧縮された映像では、すべてのフレームのデータがあるわけではない。キーフレームとなるフレームの前後いくつかは、その差分データしか保持していないため、全方位センサによるドーナツ型のフレームショットをパノラマ型に変換するために、MPEG2 圧縮された映像をデコードし、1 フレームずつ取り出す作業が必要になってくる。一方、Motion JPEG という圧縮方式では、各フレームを JPEG 圧縮し、それらを動画として保存している。この圧縮方式を採用すれば、MPEG2 による圧縮を利用した場合のようなデコード作業は必要ない。しかし、圧縮率は明らかに MPEG2 方式の方が優れている。

どちらにする、現在の非圧縮映像データを何らかの形式で圧縮することで、Retrax システムにおける映像の通信コストを削減することができる。また、ネットワーク上におけるデータ転送量にも影響する問題であるため、圧縮率が高い方がよいが、その分クライアント側での処理コストが高くなるようにする必要があるのであるため、容易に圧縮方式を決定することはできない。

### 7.2 コンテンツ情報管理データベース

現在のシステムは単一のコンテンツにしか対応しない。実用化するためには複数のコンテンツをコントロールデータベースに蓄積し、ユーザはその中から利用するコンテンツを自由に選択できるようにしなければならない。コンテンツの情報をコントロールサーバ上で集中管理し、各会議・空間の中に設置された全方位センサの台数、それぞれの位置、コンテンツの開始・終了時刻などをデータベース化する。こうして、ユーザが Retrax システムにログインする際にコンテンツの一覧を取得し、その中の一つの会議に参加する。コントロールサーバは選択された会議のコンテンツ情報を取得し、クライアントマシンに各パラメータを生成させ、システム全体の柔軟性が高まる。

### 7.3 ロードバランス

Retrax システムは複数の全方位センサを用いて仮想空間を構築する。現在は全方位センサー一台の映像や音声をコンテンツデータベースサーバー一台に蓄積する

が、全方位センサが会場内にある位置や記録したコンテンツによって、利用頻度がそれぞれ違う。

例えば、重要な発言を記録した全方位センサのコンテンツを保存するコンテンツデータベースサーバへのアクセスが集中するが、発言者から遠く離れた全方位センサで記録したコンテンツは見る人がほとんどいないためアクセスも少ない。このようにアクセスが一台のコンテンツデータベースサーバに集中する場合、レスポンスが低下していき、システム全体のパフォーマンスに大きな影響を与えてしまう可能性がある。

各全方位センサの任意の時間帯のコンテンツの人気度を把握して、バランスよくコンテンツをコンテンツデータベースサーバ群に配置し蓄積することが大事である。人気度の高い全方位センサで記録した映像や音声を複数のコンテンツデータベースサーバに分散したり、人気度の低い複数の全方位センサで記録した映像や音声を一台のコンテンツデータベースサーバに蓄積することが考えられている。コンテンツの保存状態を前に述べたコンテンツ情報管理データベースに反映する。

ある全方位センサに記録されたコンテンツのある時点の人気度を測ったり、コンテンツの再配置したりする操作を自動化することが大事である。人気度を測る尺度やコンテンツを再配置するアルゴリズムを開発することはこれからの大きな課題である。

## 8. 結 論

本論文では、全方位センサによる分散映像データベース Retrax システム及びこのシステムを高速化するためのバッファリング処理と標準化するための質問言語 RSQL について述べた。

Retrax System は複数の全方位センサと DV カメラを用いて、パーティなどのように複数の独立した会話が同時に行われている状況すべてを映像と音声で同時に記録することができる。遅延はネットワーク通信時の大きな問題点で、それにうまく対処することは非常に重要である。クライアントマシンは、ユーザの操作に応じて RSQL を生成してコントロールサーバから映像や音声を取得する。RSQL を使えば、簡単にコントロールサーバから映像や音声を取得するアプリケーションを作ることが可能である。

現在の問題点としては画像データの解像度が低く、ズームを拡大すると、見えにくくなることが挙げられる。また、通信量削減のため、圧縮技術を導入したストリーミング処理を検討する必要がある。今後は、前章の課題を取り入れ、機能を拡張しつつ Retrax シス

テムの開発を進めていきたい。

## 謝 辞

本研究は科学技術振興事業団 (JST) 戦略的基礎研究推進事業 (CREST) における「デジタルシティのユニバーサルデザイン」の支援によって行われたものです。

## 参 考 文 献

- 1) H. Ishiguro: "Compact omnidirectional sensors and their applications", Kougyou-Chosakai, March, (1998).
- 2) T. Sogo, H. Ishiguro, M. M. Trivedi: "Real-time human tracking system with multiple omni-directional vision sensors Transactions of IEICE", The Institute of Electronics, Information and Communication Engineers, (2000).
- 3) 山本文治, 櫻井智明: "ストリーミング技術", Internet Week 2000, (2000).
- 4) 廣瀬 竜男, 中西 吉洋, 秦 淑彦, 田中 克己: "サンプル映像とカメラの空間情報に基づく多視点映像の検索と直列化", 第 11 回データ工学ワークショップ (DEWS2000) 論文集, (2000).
- 5) NHK 放送技術研究所,  
<http://www.strl.nhk.or.jp/aboutstrl/index-j.html>
- 6) <http://biztech.nikkeibp.co.jp/wcs/show/leaf?CID=onair/biztech/elec/153255>