

映像データベースを用いた仮想空間システムにおける 問い合わせ機能とユーザインタフェース

久保 広樹[†] 何 書勉[†] 横田 裕介[†] 上林 弥彦^{†*}

[†] 京都大学情報学研究科 〒606-8501 京都市左京区吉田本町

E-mail: [†] {hkubo, shumian, yy, yahiko}@db.soc.i.kyoto-u.ac.jp

あらまし Retrax システムは複数のビデオストリームとハイパーメディアオブジェクトを用いて仮想空間を構築し、以前の内容を訪問できる特色を持っている。本システムでは SQL に基づくデータ問い合わせの汎用インタフェースとして RSQL (Retrax SQL) を用いている。ユーザの仮想空間内でのウォークスルー操作は内部的に RSQL に変換され、履歴として保持される。これによって単なる検索だけでなく編集による再利用が可能になる。ハイパーリンク、アノテーション等のハイパーメディア機能、およびユーザ間のインタラクション機能も RSQL エンジンを用いて実現されている。本研究では、RSQL による基本機能および各機能を RSQL で統合することによって実現される応用について述べる。

キーワード マルチメディア DB, 問合せ処理, メタデータ管理, DB 言語, ユーザインタフェース

Query Functions and User Interface for a Virtual Space System Utilizing Omni-directional Video Data

Hiroki KUBO[†] Shumian HE[†] Yusuke YOKOTA[†] Yahiko Kambayashi^{†*}

[†] Graduated School of Informatics, Kyoto University,
Yoshida Honmachi, Sakyo, Kyoto 606-8501, Japan

E-mail: [†] {hkubo, shumian, yy, yahiko}@db.soc.i.kyoto-u.ac.jp

Abstract Retrax system realizes video-based virtual space with multiple video streams and hypermedia objects. In this system, RSQL(Retrax SQL) is used as a general-purpose interface of the data inquiry based on SQL. Users' actions such as walkthrough and conversation in the virtual space are translated into RSQL, and preserved as histories. Hypermedia functions such as hyperlinks, annotations and interaction functions between users are realized by the RSQL engine. This paper describes the purpose and architecture of RSQL, and introduces a query mechanism dealing with hypermedia data and video data, then shows basic functions realized by RSQL and sample applications of RSQL.

Keyword Multimedia DB, Query Operation, Metadata Management, DB Language, User Interface

1. はじめに

技術進歩によってディスクなどの記憶容量は以前に比べて格段に大きくなり、またデータの圧縮技術の発展に伴い、まずすべてのデータを蓄えておいてからあとで必要な部分だけを検出して利用するという方法が可能になりつつある。そこで我々は、データベースに出来るだけ多くの情報を保存し、あとでそこから必要となる情報のみを選び出す“store-everything select-later”原則に基づき、Retrax システムの開発を行ってきた。Retrax プロジェクトは全方位センサによる映像を用いて仮想空間を構築し、過去のミーティングに参加できるような分散ビデオデータベースシステム

を構築することを目的としている。全方位センサを取り付けた DV カメラを複数個使うことで、ミーティングなどの状況を三次元的に保存する(図 1)。保存された三次元空間内を移動しながら再生することが可能になる。再生時にユーザは空間のオブジェクトにハイパーリンクを貼り、空間的・時間的移動、コメントの追加、音声・ビデオの追記などを行い、後でハイパーリンクの参照が可能である。各リンクは権限の属性を持ち、特定のユーザに対する表示・実行・更新(削除)などの操作権を指定する。アプリケーションとして、電子会議システムにおけるメモの追記や、仮想空間を利用したグループウェアなどが考えられる。

* 2004 年 2 月 6 日逝去

本稿では、ビデオデータから仮想空間を構築するための問い合わせ言語 RSQL(ReTrax SQL)とそれが提供する様々なユーザインタフェースや応用の可能性について述べる。

ユーザのマウス操作を内部的に RSQL に変換し、処理後の RSQL をログとして保持する。RSQL を中間インタフェース言語として用いることでビデオデータ・ハイパーメディアデータを問い合わせるための汎用インタフェースとなり、様々なユーザインタフェースを统一的に扱うことが可能になる。保持された RSQL からユーザの行動パターンや仮想空間内の重要なポイントを分析するなど多様な応用例が考えられる。



図 1 撮影状況イメージ

またシステム開発者にとって、RSQL エンジンが提供する標準 API を使って、仮想空間の構築や映像の再生など、様々なユーザインタフェースを開発できることは大きな利点である。RSQL に関する知識を持たない一般の利用者も、RSQL エンジンを利用したユーザインタフェースを使うことで、必要な情報の取得や仮想空間内のウォークスルーが可能になる。

本稿は以下のように構成される。2 章で関連研究を紹介し、3 章でシステムアーキテクチャを説明する。4 章では読み込みの機能とインタフェース、5 章では書き出し系の機能、6 章では実装と評価を行い、7 章で RSQL の応用例を記述する。8 章でまとめと今後の課題について述べる。

2. 関連研究

2.1. 仮想空間とウォークスルー

全方位センサを利用したアプリケーションやインタフェースに関する研究はこれまで数多く行われてきた。この種のカメラで撮影された画像や映像は様々なアプリケーションを魅力的なものにしている。Yongら[1]は全方位カメラを利用してミーティングなどの状況を見ることを可能にするシステムを設計し、いくつかのインタフェースを開発した。一人一人を捉えた画面を表示するインタフェースや全体を表示するインタフェースなどがある。一方、Retrax システムでは複

数の全方位センサで空間全体の様子を撮影しビデオデータとしてデータベースに保存する。保存したデータから仮想空間を構築し、ミーティング等の状況を視聴すると同時に、その空間内のウォークスルーを実現するものである。

画像処理の幾何学的な反射モデルや明示的な三次元モデルにより映像を合成せず、実写画像から三次元仮想空間を直接生成する手法は将来有望な技術である。Aspen Movie Map[11]はイメージベースドレンダリング(IBR)を用いたシステムの先駆者である。市街地の道路に沿って画像をレーザディスクに保存し、利用者はパソコンを使って画像で再現した道路上を歩くことが可能である。しかしこのシステムにおいては、カメラ自由度が低い。一方、Retrax システムは全方位センサを採用することによって、センサ周囲の方向の動画像を使って仮想空間を構築し、利用者はいろいろな方向への移動が可能になった。

2.2. 時空間データベース

時空間データベースあるいは時間データベースの問い合わせに関する研究は数多く行われてきた。

Santini らは、ビデオシーケンスの中から興味深いイベントを問い合わせるための時空間データモデルを提案した[2]。そのモデルの特徴は、ビデオデータを未要約データと要約データの 2 層に分割することである。独自の問合せ言語を使わずに、コンピュータビジョン技術を用いて未要約データに対して要約をするメカニズムが提案されている。Retrax システムの場合、時空間データモデルを設計する目的は生のビデオデータからイベントやオブジェクトを要約することではなく、時空間情報を記述するための柔軟で簡単な方法と、ユーザが GUI を通してシステムにインスタンスを追加する方法を構築することである。

SQLST[3]では時空間データモデルと SQL3 標準の拡張としてのクエリ言語が定義された。これは時間的に変化するオブジェクトを詳細に記述できるだけでなく、詳細なデータモデリング手続きも必要とされている。一方、Retrax システムの場合、複雑なデータモデルが必要でないため、独自のハイパーメディアデータモデルに基づいた簡単な時空間データモデルを採用した。

Nelken と Francez[4]は、時間データベースを問い合わせる自然言語インタフェースを提案した。そのインタフェースを用いて自然言語のクエリを SQL/Temporal に変換する。SQL を書くことは一般のユーザにとって難しい作業であるため、データベースを問い合わせるもっと簡単な方法を提供することは重要である。そのアプローチと対照的に、我々はユーザが

GUIでの操作をSQLベースの言語に翻訳する方法を開発し、バーチャルリアリティシステムのような時空間データベースに基づくリアルタイムのアプリケーションの開発を簡単にした。さらに、翻訳結果となるクエリ文はユーザの操作記録として保存され、個人的なビデオデータを構成するときに再利用することができる。

2.3. ビデオデータ検索

ビデオデータを処理する実際的なアプリケーション開発には、ビデオデータを検索し取得するハイレベルな機能が必要である。それはビデオデータのメタデータを生成するために適切な設計および技術が必要とされる。「マルチメディア内容記述インタフェース」と呼ばれる MPEG-7[9]は様々なマルチメディアデータのメタデータの標準記述である。Libsie と Kosch[10]は、MPEG-7 のフレームワークを用いて、MPEG-4 のビデオデータに適用するためのメタデータ記述手法を提案した。我々は、Retrax システムのメタデータのフォーマットとして独自のハイパーメディアデータモデルを選び、システムの内容の時空間的な性質に注目した。MPEG-7 のメタデータは一本のビデオストリームの特徴・イベントやアノテーションを記述する。一方、Retrax システムでは、仮想空間は複数のビデオストリームによって構成される。空間内の同じオブジェクトは複数の全方位センサで記録した映像に映る可能性はあるが、それぞれのビデオストリームに関するメタデータに同じオブジェクトを記述するより、システム全体を見下ろすようなメタデータのほうが本システムに適すと考えられる。すなわち、ある空間内のオブジェクトはシステムのメタデータ情報に唯一のオブジェクトに対応することとなる。ビデオデータあるいは他のデータから生成された時空間情報は、ハイパーメディアデータへ併合される。ハイパーメディアデータモデルは、さらに様々なタイプのリンク、アノテーションなど、標準のハイパーメディアの機能を持っている。

3. システムアーキテクチャ

Retrax システムは映像データベース・コントロールサーバ・クライアントで構成される。映像・音声ファイルは映像データベース内に蓄積され、ユーザのクライアント上での操作がコントロールサーバへ送信され、コントロールサーバで必要なデータを判断する。最後に映像データベースからクライアントへそれらのデータを送信し、クライアントで映像や音声を再生する。

3.1. データ構造

Retrax システムに使われるデータは三種類である。**マルチメディアデータ** デジタル形式の音声・ビデオデータファイル。仮想空間を構成する「素材」であ

る。

ハイパーメディアデータ ビデオアノテーションやユーザがハイパーリンクを追加する際に、データベースへどんどん保存していくデータ。また、仮想空間内の人間や特定のオブジェクトの座標もハイパーメディアデータとして扱う。

メタデータ 多くのビデオデータベースでは、メタデータはビデオデータについての情報として扱われる[6]。本システムでは、これをビデオデータとハイパーメディアデータについての情報として定義する。ビデオやアノテーションなどのデータを検索する際に必要である。メタデータの概要は表 1 に示す。MPEG-7 のメタデータ記述は標準化されているが、複数のビデオストリームで仮想空間を構築するこのシステムでは、ファイルごとにメタデータを用意せず、仮想空間全体の特徴を一括管理するデータモデルを提案した。

表 1 メタデータ

項目	概要
vdata	ビデオデータの特徴
products	仮想空間の情報
sensorinfo	全方位センサの情報
events	空間内のイベントの情報
anchors	アンカーに関する情報
objects	空間内のオブジェクトの情報
action_log	利用者の操作履歴の特徴情報

3.2. RSQL による問い合わせの抽象化

本システムにおいて仮想空間内のウォークスルーを実現する際、一つ一つの動作、例えば映像の再生や停止・ユーザの前進や旋回などに対して問い合わせを行っていた。しかしこの場合、新たなアプリケーションインタフェースを作成する場合に最初から問い合わせを検討していかなければならない。これは新たなインタフェースの作成を検討する場合非常に面倒である。ここでこの内部的な問い合わせを一括して処理する仕組みがあればアプリケーションの開発の際には非常に便利であると考えられる(図 2)。さらにこの仕組みでは、これまでの内部的な問い合わせに比べ抽象度の高い問い合わせを実現するため、問い合わせそのものがログとして利用しやすくなると考えている。これによって各データの再利用性が高くなり、履歴を用いた重要シーン抽出などの応用アプリケーションの開発も可能になるのではないかと考えている。

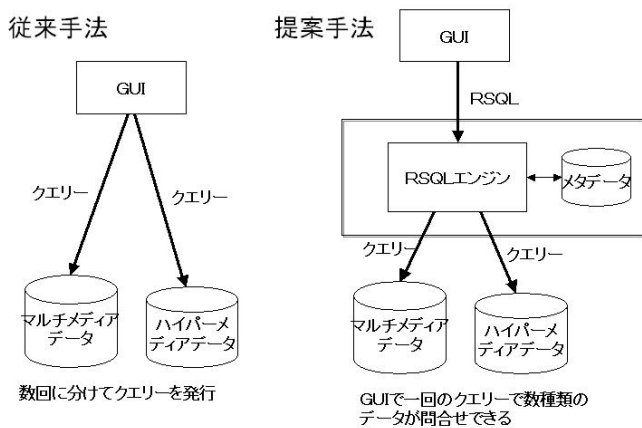


図 2 問い合わせ処理方式の比較

3.2.1. RSQL

本システムでは、ユーザのクライアントでの操作を RSQL 文に翻訳してコントロールサーバへ送信する。RSQL (Retrax SQL) [5]は、我々が定義した映像や音声などのマルチメディアデータあるいはハイパーメディアデータを取得し蓄積するための質問言語であり、またそれらの仕組みのことである。RSQL とはメタデータから検索の対象の特徴情報を取得し、独自の RSQL エンジンによって解析され、抽象度の高いクエリーを生成するものである。本システムでは複数のファイルを同時に扱う必要があり、またビデオデータや音声データだけでなくアノテーションなどのハイパーメディアデータも扱うため一括して問い合わせることができる RSQL を提案する。

ここで RSQL と SQL の違いについて説明する。RSQL は SQL のサブセットであり、構文は SQL と同じものである。RSQL は RSQL エンジンによって処理され、その結果としてデータセットではなく低レベルの SQL 文が生成される。RSQL エンジンは RSQL 文の条件とメタデータの情報に基づき、ハイパーメディアデータやマルチメディアデータを蓄積しているデータベースと映像サーバを決め、従来の SQL 文を発行する。従来は GUI から低レベルの複数の SQL 文を発行しなければならず、映像情報やハイパーメディアデータの特徴情報もあらかじめクライアントに指定する必要があった。RSQL の利用によって、アプリケーションの開発者は複雑なデータ構造を意識する必要が無く、一つの文で複数の問合せ先を指定することができる。

RSQL を利用することによって、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。一例としてユーザがクライアントマシン上で空間内をウォークスルーする場合を説明する。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・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 はコントロールサーバ内の RSQL エンジンによって分析され、再生に必要なデータを抽出するためのファイル情報や位置情報を取得する。

3.2.2. RSQL エンジン

クライアント上でのユーザの操作は、仮想空間内の現在の位置・方向・ウォークスルー開始からの経過時間などの情報を含んだ RSQL に直接変換される。これらはコントロールサーバへ送られる。そして RSQL 文を解析し、クライアント上で仮想空間を構築するために送信すべきマルチメディアデータやハイパーメディアデータを決定する。我々はこのモジュールを RSQL エンジンと呼んでいる。図 3 に RSQL エンジンとユーザインタフェースおよびデータベースの関係を示す。RSQL エンジンはユーザインタフェースとデータベースの仲介役である。RSQL エンジンはマルチメディアデータやハイパーメディアオブジェクトを問い合わせるための共通インタフェースとして設計されているため、データ検索に関して開発者の助けとなる共通 API を提供する。RSQL エンジンを利用することで、仮想空間内でのデータの更新や検索のための様々なユーザインタフェースが利用可能になる。

Retrax システムのマルチメディアデータとハイパーメディアデータはそれぞれ違うデータベース内に蓄積される。データを問い合わせる際に、必要とされるデータの種別とそれを格納するデータベースを、RSQL 文の解析を通して知る必要がある。

RSQL 文の SELECT の対象は、音声・映像・ハイパーメディアデータである。WHERE の条件として指定できるものには、利用者の空間内にある座標、向いている方向、経過時間、速度指定などがある。RSQL エンジンはこれらの条件を使ってメタデータから、全方位センサや画面上に表示されるオブジェクトのアンカー位置などの情報を取得する。例えば、利用者の空間内の座標と向いている方向のベクトルで、再生すべき映像を記録した全方位センサのおおまかな位置が得られる。このおおまかな位置情報をメタデータの sensorinfo において検索すると、全方位センサの番号が分かる。さらにこの番号の全方位センサで記録した映像の特徴をメタデータの vdata から検索する。最後に検索の結果に基づき、ビデオデータベースから適切な映像を取得することができる。上記の複雑な問合せはメタデータを格納する RDB において従来の SQL 文を用いて実現される。すなわち、RSQL 文の直接処理結果は、画面上に表示されるマルチメディアデータとハイパーメディアオブジェクトではなく、これらのデータを問い合わせるためのクエリーである。ユーザの

操作はクライアントにおいて RSQL に変換され、RSQL はエンジンにおいて低レベルの SQL クエリーに変換される。最後に、低レベルの SQL クエリーを使ってデータベースから情報を取得することとなる。RSQL を使うことによって、操作を RSQL への変換、および RSQL を SQL への変換のステップが必要となるため、クライアントから直接 SQL を発行する方法に比べ若干処理速度が遅くなる可能性があるが、RSQL エンジンの処理能力を把握した上で改善することができると考えられる。RSQL エンジンの処理能力の評価は第 6 章で紹介する。

RSQL エンジンは、RSQL PARSER と FIXED MODULE によって構成される。RSQL PARSER は RSQL を解析し、FIXED MODULE は Meta DB からメタデータを読み取り、カメラの位置情報などに基づいて計算することによって、再生に必要な情報の取得や、アノテーションなどの操作を行う。操作例は第 6 章で紹介する。処理された RSQL は操作履歴として Log DB に蓄積される。

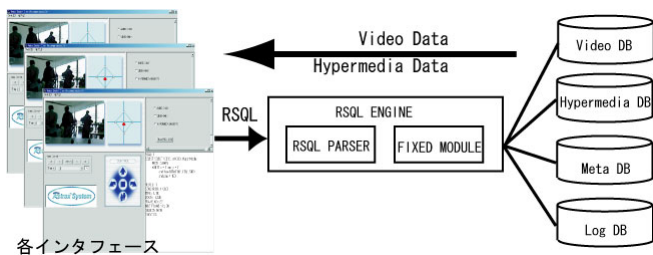


図 3 RSQL エンジン

4. 読み込み系の機能

RSQL エンジンを使うことで、数種類のユーザインタフェースが利用可能になる。ユーザのマウスによる操作を自動的に RSQL に変換し、再生に必要な情報を取得して映像・音声やハイパーメディアデータをクライアントで表示するウォークスルーインタフェース (図 4) とユーザが明示的に RSQL 文を入力し、高度で柔軟な問い合わせを行う直接入力インタフェース (図 5) がある。またこれら以外にも必要に応じてインタフェースを開発可能である。処理後の RSQL は履歴として Log DB に蓄積していく。

4.1. ウォークスルーインタフェース

ユーザは前進・後退・旋回などの特定のボタンをクリックすることで仮想空間内を自由に歩き回ることができる。ユーザの仮想空間内の位置や方向に応じて、映像データベースから適切な映像と音声を取得して表示している。再生支援として、一時停止・早送り・巻き戻しなどの操作がある。画面上にハイパーリンク等のハイパーメディアデータが表示され、さらにユーザ

がハイパーリンクを貼ることやアノテーションを行うことができる。ウォークスルーインタフェースでは、0.5 秒おきにユーザの操作と仮想空間内の状態を RSQL に変換してコントロールサーバ内の RSQL エンジンに送る。

例：ユーザが CAMP5 という仮想空間内を歩き回っている。現在の座標は(0,0)、時間は 1.000 秒、向いている方位角は 180° とする。0.5 秒間の映像と音声およびハイパーメディアデータを取得するために、以下の RSQL が発行される。

```
SELECT VIDEO, AUDIO, HyperMedia
FROM CAMP5
WHERE x = 0 and y = 0
and time BETWEEN(1.000,1.500)
and pan = 180
```

これは RSQL エンジンで解析され、Meta DB の中に蓄積されたカメラ位置などのメタデータと照合して、適切な映像を記録した全方位センサの番号・各音声ストリームの音量と映像再生の拡大率などを計算し、クライアントへ返す。クライアントはこれらの値を映像サーバへ送信して、必要なデータを取得する。

これによってユーザは単にイベント時の映像を視聴するだけではなく、その会場内をウォークスルーすることによりあたかも過去のイベントに参加しているかのような体験をすることが可能である。



図 4 ウォークスルーインタフェース

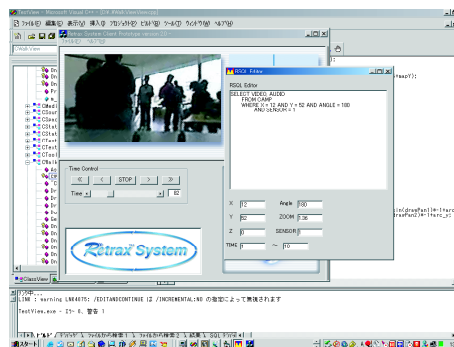


図 5 直接入力インタフェース

4.2. 直接入力インタフェース

RSQL をクライアントが自動的に発行するウォークスルーインタフェースに対し、直接入力インタフェースはユーザが意図的にデータを取得するために RSQL を作成する機能を提供する。例えばユーザが特定のカメラの撮影した映像を拡大し分析する時などに有用である。ユーザは RSQL の組み合わせで高度な問い合わせが可能である。4.1 節の例において、映像取得の持続時間 (1.000 秒から 1.500 秒) を自由に変えることも応用例の一つである。

また RSQL を自由に記述できるため、既知の情報に対しては直接検索することが可能になる。これまでのインタフェースだけでは見たいシーンがある場合、それがどの場所までウォークスルーインタフェース上でわざわざ歩いていき視聴する必要があった。しかし本インタフェースを用いることで、注目したい人物がいつどこにいたのかを把握していれば、直接そのシーンを検索し再生することができるようになる。

さらに履歴として残っている RSQL 文をコピーすることで、ユーザの以前の操作を再現することも可能である。

5. 書き込み系の機能

Retrax システムでは、ユーザがハイパーリンクを貼ったりビデオアノテーションを行ったりすることが可能である。このハイパーリンクを作成する操作やビデオアノテーションも RSQL に変換される。ハイパーリンクやアノテーションは内部構造的には同様のものであり、リンクとして表現される。リンクには権限情報も保持することができる。リンクはクライアントの映像再生画面上に表示し、ユーザは自由にリンクを作成可能である (図 6)。本インタフェースで作成可能なリンクは時空間ハイパーメディアリンク、ビデオアノテーションリンク、外部アプリケーションへのリンクなどである。それぞれの特徴を以下に示す。

- 時空間ハイパーリンク
仮想空間内の別の場所・時間を示す
 - 外部アプリケーションへのリンク
ウェブページの URL や特定のプログラム
 - ビデオアノテーション
ビデオデータへのリンク。将来的にはユーザが新たにビデオデータを追加することで過去のインタラクションを目標としている
 - テキストリンク
ユーザがある時刻のある場所にコメント等を付ける
- ハイパーリンクやアノテーション等のリンクはユーザ

インタフェース上で表示される。図 6 に新規のリンクを追加するためのユーザインタフェースを示す。

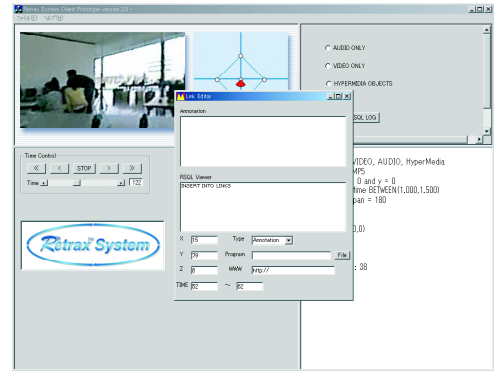


図 6 リンク作成インタフェース

リンクのデータ構造は以下のようなものである：

$$h_i = (\text{anchor}, \text{type}, \text{contents_destination}, \text{authority})$$

- i : ハイパーリンクの番号
- anchor : 元データに対する位置と時間の指定。
- type : リンクの種類。各種アノテーション、WWW リンクなど。
- $\text{contents_destination}$: リンクの内容/行き先。ビデオアノテーションの場合であればビデオデータ。他の時空間へのリンクであれば anchor に準じる形式のデータ。WWW へのリンクであれば URL など。
- authority : どのユーザかグループに対して有効であるかを示す。

例えば、ユーザ a は再生開始から 90 秒後に、仮想三次元空間内の座標 (453,24,90) にリンクを貼り、自分のコメントを記入すると、以下の RSQL が生成される：

```
INSERT INTO LINKS
(time, type, contents_destination, x, y, z, authority)
VALUES
(90.000, text, "アノテーションの内容",
453,24,90, "a")
```

6. 実装と評価

これまで述べてきたように、RSQL エンジンには RSQL 文の解析とマルチメディアデータを取得するためのパラメータの決定を目的として設計されている。本章では RSQL エンジンとそれに基づくいくつかのインタフェースを開発し、従来のインタフェースと比較することで、RSQL エンジンの性能を評価する。以下での実装は、Pentium III マシン、Windows2000 で動作し、Visual C++ で開発した。

6.1. RSQL エンジンの実装

RSQL エンジンにクエリを投げ、結果を受け取るインタフェースに用いられる共通 API を開発してきた。

処理された RSQL 文が保存される Log DB と RSQL エンジンから参照されるメタデータは、現在 Microsoft Access 2000 によって実装されている。

RSQL エンジンは三つの動作から構成されている。

- ユーザインタフェースと接続し、RSQL 文を発行する
- データベースから得たメタデータをもとに RSQL 文を解析する
- ユーザインタフェースに適切なパラメータを返す。同時に処理済の RSQL 文を Log DB に保存する

実験の結果、RSQL エンジンの RSQL 文の処理効率は、メタデータテーブルとログテーブルへのアクセス時間に非常に影響を受けていた。CPU 1GHz、メモリ 512MB の環境下で、一つの RSQL 文に対する処理時間の平均は 0.13 秒であった。これについての評価は以下の節で行う。

6.2. テストデータの作成

実験に用いたデータは、実際のワークショップの様子を撮影したものである。以下のように動きや会話がある状況が望ましいと考えている。

- 複数の人間がいて、いくつかのグループがある
- 動作は激しすぎず穏やかである
- 会話が行われている

今回は CAMP/大川センターにて、20 名程度のワークショップを撮影した。これは当ワークショップが上記の条件に適していると判断したためである。ワークショップ会場のそれぞれの角に 1 台ずつと中央に 1 台の合計 5 台の全方位センサを設置し撮影を行った(図 7)。

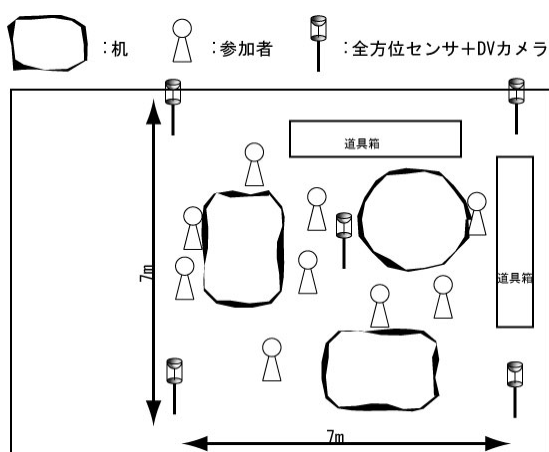


図 7 ワークショップ撮影状況

6.3. RSQL エンジンの性能評価

RSQL エンジンの性能はシステム全体の性能に大きな影響を与える。ウォークスルーインタフェースでは、一定の頻度で利用者が仮想空間で行う行動や位置・角

度などの状況を RSQL 文に変換し、RSQL エンジンはそれらを解析する。システムパフォーマンスから見れば、インタフェース側での RSQL 文の生成頻度が高くなるにつれ、単位時間内にエンジンはより多くの RSQL 文を処理する必要がある。生成頻度が一定の閾値より小さくなると、RSQL 文の生成速度はエンジンの処理速度を上回り、RSQL 文の処理が追いつかなくなる恐れがある。ユーザビリティから見れば、RSQL 文の生成頻度が低くなると、利用者の仮想空間内でのウォークスルー操作に対する反応時間が長くなり、臨場感が低くなる。パフォーマンスとユーザビリティを考慮した上での最適な RSQL 文の生成頻度を求める必要がある。

RSQL の性能をウォークスルーインタフェースにおける実行時間の比較によって行う。RSQL エンジンを用いない従来のインタフェースと RSQL エンジンを実装したインタフェースで比較する。

RSQL 文は等間隔で周期的に発行されるが、この間隔を 0.1 秒から 1.0 秒まで変化させる。RSQL 文が生成される間隔は RSQL エンジンの性能に影響すると考えられる。もしその間隔が閾値より小さければ、RSQL 文は適宜処理されない可能性があり、これらのキューが RSQL エンジン内に生成されていく。この閾値は実験を通して決定していく。

RSQL 文生成に適した間隔を調べるために、我々は RSQL エンジン内のキューにたまっている未処理の RSQL 文の数と、一つの RSQL 文に対する平均処理時間を計測した。表 2 に実験結果を示す。

表 2 RSQL エンジン性能評価実験結果

間隔(秒)	キュー内の未処理の RSQL 文数	一文あたりの平均処理時間(秒)
0.1	186	0.2000
0.2	64	0.1500
0.3	25	0.1600
0.4	12	0.1400
0.5	0	0.1340
0.6	0	0.1290
0.7	0	0.1280
0.8	0	0.1290
0.9	0	0.1254
1.0	0	0.1223

間隔が長くなるにつれて、処理時間と未処理文の数が増加する傾向を示している。間隔が 0.4 秒以上の時、未処理の RSQL 文は 0 となり、平均処理時間は約 0.13 秒であった。

この結果から閾値を 0.4 秒から 0.5 秒の間であると考え、ウォークスルーインタフェースにおいて、RSQL 文を発行する間隔は 0.5 秒とするものとする。

7. RSQL の利用

RSQL は RSQL エンジンによって分析されたあと、ログとして Log DB へ蓄積される。RSQL の構文は柔軟であるため、ログの再利用が可能である。本章ではログの利用方法と応用例について述べる。

7.1. ログの利用

RSQL ログは時間順に Log DB に蓄積されていく。ワークスルーインタフェースで生成された RSQL をユーザの操作と仮想空間内での行動履歴と見なす。同一仮想空間内の複数ユーザのログを分析することで、興味深いあるいは重要なポイントの発見が可能であると思われ、この情報をユーザに提示する事によってコンテンツの再利用性が高まると考えられる。またログからユーザの行動パターンの分析や、ユーザの意図を発見することが期待できる。

Log DB 内の RSQL を順次 RSQL エンジンへ送信することで、ユーザの操作を再現可能である。一人のユーザが仮想空間内で見えた物や、取った行動などの履歴情報を視覚化することも有用と思われる。

4.2 節の直接入力インタフェースでは RSQL の修正を行うことができる。ユーザが自分のログを修正することで、別のシナリオを自分で作り仮想空間を再体験することもできる。

7.2. 応用例

RSQL のログを利用して、過去に自分が注目した場面や他の多くの人に注目されている場面を発見する。これらを組み合わせることで個人毎のイベント体験記録やイベント中の興味深い部分をまとめた記録を作成・再利用することが可能になると考えている。

また、複数ユーザのログを合成・編集して、多視点ビデオも作成できると思われる。さらに、クエリーを自由に発行することができるので、データの再利用時には当初と異なる視点からイベントを再体験することが可能になる。

8. まとめと今後の予定

本稿では、ビデオデータから仮想空間を構築するための問い合わせ言語 RSQL とそれが提供した多様なインタフェースを紹介した。ユーザの仮想空間内でのワークスルー操作は内部的に RSQL に変換され、操作と行動履歴として保存される。履歴を編集することによって様々な方法で再利用可能となる。また、本システムのハイパーリンク、アノテーション等のハイパーメディア機能も RSQL エンジンを用いて実現される。

現在本研究チームが力を注ぐもう一つの課題は位置情報による映像の再構成である[8]。撮影時にセンサ等を用いて記録した参加者の位置情報及び映像に記録された音声データなどを基に検索用のインデックスを

生成することによって大量の映像データから必要な部分のみを選択し映像を再構成することが可能になる。この機能を用いることで、例えば撮影データから特定の人物を追う映像を生成することや、興味のある会話が行われているグループの映像のみを取り出すといった利用が可能になると考えられる。位置情報をいかにして RSQL に統合させることも今後の課題の一つである。

謝 辞

本研究は独立行政法人・科学技術振興機構(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。また映像コンテンツとしてワークショップの撮影をご承諾いただいた CAMP/大川センターに、深く感謝いたします。

文 献

- [1] Yong R., Gupta A., and Cadiz J., Viewing meetings captured by an omni-directional camera. In ACM Trans. Comput.-Hum. Interact., March 2001.
- [2] S. Santini, A. Gupta, and R. Jain, Querying multiple perspective video. In Proc. of SPIE Vol.3656, Storage and Retrieval for Image and Video Database VII, January 1999
- [3] C.X. Chen and C. Zaniolo, SQLST. A Spatio-Temporal Data Model and Query Language. In ER2000 Conferences, LNCS1920, p96-111, 2000.
- [4] Nelken, R. and Francez, N., Querying Temporal Databases Using Controlled Natural Language. In the 18th International Conference on Computational Linguistics (COLING 2000), Jul. 2000.
- [5] 何書勉, 横田裕介, 上林弥彦, 全方位センサによる分散ビデオデータベースのアーキテクチャとプロトコル設計, 第 13 回データ工学ワークショップ (DEWS2002), Mar, 2002
- [6] R. Jain and A. Hampapur. *Metadata in Video Databases*. In ACM SIGMOD RECORD, 23(4): p27-33, Dec. 1994
- [7] Gupta A., T. Weyworth, R. Jain, "Semantic Queries with Pictures: The VIMSYS model", in Proceedings of the 17th VLDB Conf., Barcelona, Spain, 1991.
- [8] 久保広樹, 何書勉, 横田裕介, 上林弥彦, "複数カメラを用いた映像データベースにおける参加者の個別映像生成機能," 情報科学技術フォーラム (FIT2003)
- [9] <http://ipsi.fraunhofer.de/delite/Projects/MPEG7/>.
- [10] Libsie, M. and Kosch, H., Content adaptation of multimedia delivery and indexing using MPEG-7, Proc. of the 10th ACM international conference on Multimedia, Dec. 2002, pp.644-646.
- [11] A. Lippman, Movie-maps: an application of the optical videodisc to computer graphics. In Proc. of ACM SIGGRAPH '80, pages 32-43, 1990.