

時間的・空間的関連を表現できる ビデオデータベース操作言語

本坂 錦一[†] 大本 英徹^{††} 田中 克己^{†††}

[†]神戸大学大学院自然科学研究科情報知能工学専攻

^{††}京都産業大学工学部情報通信工学科

^{†††}神戸大学工学部情報知能工学科

最近、オーディオ/ビデオデータベースがたいへん注目されている。特に、オーディオ/ビデオデータモデルに関する研究とマルチメディア情報の蓄積配送技術に関する研究は、たいへん盛んに行なわれている。しかし、オーディオ/ビデオデータベースの操作言語 (DML) に関する研究は、Maier らの「DML は、いつ (when) どこに (where) どのような品質で (QOS) 結果を配送する必要がある」との指摘がある程度である。そこで本稿では、オーディオ/ビデオ情報を取り扱うことのできる DML である AVQL (Audio Video Query Language) について述べる。AVQL は、検索されたオーディオ/ビデオ情報に対して時間的な合成や空間的な配置を記述できる。AVQL では、ビデオデータモデルとして我々の研究室で開発した OVID データモデルを用いている。

A Video Database Manipulation Language with Temporal/Spatial Relationships Manipulation Capability

Kinichi Motosaka[†] *Eitetsu Oomoto*^{††} *Katsumi Tanaka*^{†††}

[†]Graduate School of Science and Technology, Kobe University

^{††}Faculty of Engineering, Kyoto Sangyo University

^{†††}Faculty of Engineering, Kobe-University

Recently, much attention has been focussed on audio/video (AV) database systems. Especially, much efforts have been devoted to the reaserach and development of audio/video data models and multimedia storage/delivery mechanisms. On the other hand, a few research is done concerned with Database Manipulation Languages (DMLs) for audio/video databases except the introduction of concepts of delivery and QOS into DMLs by D.Maier. In this paper, we will describe our DML for handling AV information, called AVQL (Audio/Video Query Language). AVQL focusses on manipulating temporal and spatial relationships for retrieved AV information. Our previous video data model, called OVID, is used as a basis for the AVQL.

1 はじめに

オーディオ/ビデオデータベースシステムに関する研究においてビデオデータモデルについては、OVID モデル [1] や Gibbs のモデル [2] や OMEGA データモデル [3] などのように数多く報告されるようになったが、このようなデータベースの操作言語 (DML: Database Manipulation Language) については、Maier ら [4] の「DML はいつ (when) どこに (where) どのような品質で (QOS) 結果を配送するか」という記述が行なえる必要がある」、との指摘・提案がなされている程度で、まだあまり研究開発が活発には行なわれていないのが現状である。

本研究では、AV データベースの操作言語は、どのような機能が必要であるかの提案を行ない、そしてその機能を満たすよう操作言語を実際に設計し、プロトタイプの開発を行なった。

大量に蓄積されたオーディオ/ビデオ情報のデータベースを利用しようとする時には、ビデオ映像に何の編集操作も加えずに利用することはあまりなく、ほとんどの場合はテロップを入れたり、複数の映像をつなぎあわせたり、音楽を入れたりなどの編集操作を施してから、実際に利用する。

例えば、ホームビデオでとった子どもの映像から「自分の子どもの成長の様子」と言った映像を作る場合でも、まず、0 才でとった映像の中から印象的なものを抜きだし、そして 1 才でとった映像からも同様に抜きだして 0 才の映像と組合せ、それから「0 才」とか「1 才」とか言ったテロップをいれ、そして音楽を入れてから表示するのが普通であろう。

このように、ビデオは検索・編集の過程を経て利用するのが一般的である。AV データベースに格納されているビデオ映像を利用する場合にも、検索してきた映像に対して編集操作を行なったから、ビデオ映像を利用するのが一般的であると考えられる。したがって、AV データベースを利用するアプリケーションを作る場合、ビデオ映像に対する編集機能をアプリケーション中に埋め込まなければならず、アプリケーションの開発に大変な手間がかかる。そこで、AV データベースシステムの側で編集機能を持たせれば良いのではないかと考えられる。

そこで、本稿ではオーディオ/ビデオに対して基本的な編集操作を提供する DML の提案を行なう。これにより、アプリケーションは編集機能を持たなくて

も良いので簡素化がはかれる。

2 AV データベース操作言語に求められる要件

この節では、AV データベースを利用する応用プログラムの例を 2 つ考えて、その上で実際の DML のビデオ映像の編集機能としてどのようなものが必要であるかを考える。

2.1 想定される応用

● パーチャルデパート

ネットワークが発展してくると家に居ながらにして買い物をするというようなことができるようになると思われる。そのようになってくると、無数の商品から自分の欲しい商品を見つけ出す手助けをするものが必要となる。このような一例として考えたのがパーチャルデパートである。これは、デパートの中を歩いているような気分にするような画面 (Walk Through のような感じ) を提供して、商品を選ぶとその詳しい説明がビデオで見ることができるといものである。デパートでは置いてある商品はいつも同じとは限らなもので、商品説明であるビデオも頻繁に変わることが多い。このようなことに対応するには、その時点で該当する商品の AV 情報を動的に検索・編集して表示することが求められ、データベース操作言語の利用が適当であると考えられる。商品の説明のビデオを AV データベースに格納しておき、説明が欲しい時は操作言語を用いて呼び出せば良い。

● ニュース作成支援システム

ニュースはいろいろな関連する場面からなっていることが多い。また、ニュースは編集した後には報道しなければ意味のなくなってしまうことが多い。そのため、ニュースを作るのには多くの映像を短時間で探し出し (作りだし) 編集をしなければならない。このようなことは、データベースの得意とするところである。したがって、ニュースを作るのに AV データベースを利用すれば効率が上がるのではないかと考えられる。このような点から考えられたのがニュース

作成支援システムである。ニュースの元となるような映像を AV データベースに格納しておき、ニュース作成時にはいろいろな映像を AV データベースから引き出してそれらを組み合わせる。

2.2 DML の要件

上の 2 つの例を見ると、バーチャルデパートではビデオ映像に対して音楽やテキストを入れる、2 つのビデオ映像を同時に表示させる、ニュース作成支援システムでは、複数のビデオ映像を時間的に組み合わせる、BGM やテキストのスーパーインポーズをつける、といった編集作業を行なっている。

このように、ビデオ映像を時間的に組み合わせるとか 2 つのビデオ映像を同時に表示させるとかいった編集操作は、時系列データでありまた 2 次元的な空間の広がりを持つビデオならではのものと考えられ、これらの編集操作がビデオ映像に対する編集操作の中で基本的なものと考えられる。

以上のことより、AV データベースの操作言語は検索以外に、次のような操作機能を持っている必要があると考える。

- 問い合わせの結果として得られたビデオやオーディオの時間的な合成が行なえる。
- 問い合わせ結果の表示環境、すなわち空間的な配置・合成（表示画面の位置や大きさ）が指定できる。
- 編集上の特殊な効果のサポート。

このような指定を DML で行なえることにより次のような利点がある。

- 応用プログラムの簡素化。
- データベースへのアクセスの最適化にシステム側が責任をもてる。
- マルチメディア素材の更新・追加が頻繁な環境では検索操作言語を用いる方が有効。

3 AVQL の設計

前節で述べた要件をみたまデータベース操作言語 AVQL(Audio Video Query Language) を実際に設

計してみた。SQL 風の問い合わせ文を基本にして Select 句の中に新たに window 節、time 節、space 節、option 節を導入して、時間的な組合せや空間的な配置などを指定できるようにしている。ここで、データモデルとしては、OVID モデルを利用した。OVID モデルの特徴は、(1) 任意のビデオフレーム系列や音声をオブジェクトとして定義・属性づけを行なえ、(2) オブジェクト間の時間的な包含関係による属性情報の動的継承ができ、(3) 属性値としてオブジェクトを代入できる、といった点である。

実際の構文は以下ようになる。

```
select    window1{ time{—}
           space{—}
           option{—} }
          window2{ time{—}
           space{—}
           option{—} }
from      Video v1,v2
           Audio a1,a2
where     <検索条件>
```

ディスプレイ上で 1 連の映像を表示する枠の単位を window として、それぞれの window に対して、time、space、option を指定するようにしている。time 節では、「A と B の同時スタート」や「A がスタートしてから t 秒後に B スタート」といった時間的な合成の記述を start(A,B),startstart(A,B,t) といった形で表現する。space 節では画面の位置や大きさなどといった空間的な配置の記述を行ない、option 節では編集上の特殊な効果に関する記述を行なう。また、From 句において OVID モデルにおけるビデオオブジェクト、オーディオオブジェクトを表すオブジェクト変数の定義を行ない、Where 句でその変数を用いて検索条件の記述を行なう。

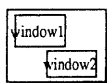
AVQL で記述した例を図 1 に示す。

4 AVQL の実装

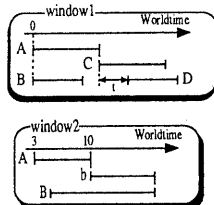
この節では、3 節で述べた AVQL が動作するシステムのプロトタイプについて説明を行なう。本システムは、Macintosh 上において HyperCard と QuickTime XCMD¹を用いてインプリメントされている。また、

¹HyperCard 上で QuickTime を使うことができるようにするための外部拡張コマンド

○画面上の配置



○タイムチャート



```

select window1{ time {start (A,B);
                    follow(A,C);
                    startstart(C,D(t));
                    Worldtime(A:0,*) }
                space { +50+50@200x100 }
                option { A:fast}}
window2{ time{ Worldtime(a:3,10);
                follow(A,B);
                end(B,F) }
          space{+300+300@150x60}
          option{B:slow :random} }

from Video A,C,E;
Audio B,D,F;
where A.car= skyline and C.car=corona and
      B.title=yesterday and D.title=freedom and
      E.creature=dog and F.title=thriller

```

図 1: AVQL の記述例

AV データベースシステムとしては OVID システムを用いている。図 2 に実行例を示す。

QuickTime は、ビデオ、音声、テキストを別々のトラックとして持っているの、動画のみ、音声のみ、テキストのみなどの形で保存できる。この機能を利用して音声のみの QuickTime Movie を作ることによって、ビデオのみを扱う OVID システムでもオーディオを扱えるようにして、オーディオデータベースを構築した。

OVID システムでは、ビデオオブジェクト情報を保持するデータベースは独立したスタックとして存在し、実際の映像である QuickTime Movie とは切り離されている。そのため、AVQL が処理をする時にはまずビデオデータベースであるスタックに対して検索を行ない、その検索結果に対応する QuickTime Movie を呼び出すという形を採っている。

このシステムは、検索部と表示部の 2 つのサブシステムからなっている。検索部では、ビデオ／オーディオデータベースに対して検索を行ない、その結果を表示部に渡す。表示部ではビデオ映像の時間的な合成などの編集作業を行なって、ユーザに対して実際に表示をおこなう。

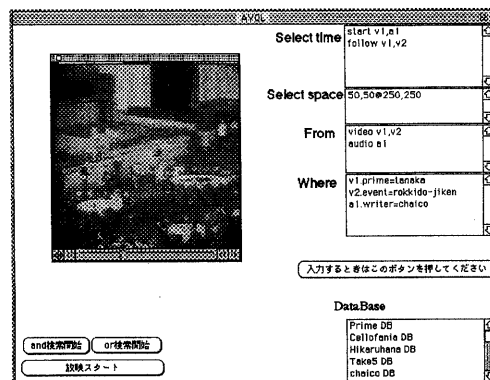


図 2: 実行例 ©NHK Service Center[5]

5 おわりに

本稿では、ビデオデータベースの操作言語はビデオ映像に対する基本的な編集機能を備えているべきであるとの提案を行なった。そして、その要件を満たす操作言語を設計し、プロトタイプを設計した。

今後の課題としては、select 句の拡張が中心であった AVQL の言語仕様の再検討、質問処理の最適化、実際規模のデータ上での実証などが考えられる。

参考文献

- [1] Oomoto, E. and Tanaka, K., *OVID: Design and Implementation of a Video-Object Database System*, IEEE Trans. on Knowledge and Data Engineering, Vol.5, No.4, pp.629-643, August 1993.
- [2] Gibbs, S., Breiteneder, C. and Tsichritzis, D., *Audio/Video Databases*, Proc. of the 4th Intl. Conf. on Data Engineering, pp.381-390, 1993.
- [3] 増永 良文, 「マルチメディアデータモデル OMEGA における音データと動画データデータのサポート」, 情報処理学会アドバンスト・データベース・シンポジウム'93 論文集, pp.163-179, 1993 年 12 月.
- [4] Maier, D., Walpole, J. and Stachli, R., *Storage System Architectures for Continuous Media Data*, Proc. of 4th Intl. Conf. on data Organization and Algorithms (FODO'93), Lecture Notes in Computer Science 730, Springer-Verlag, pp.1-18, October 1993.
- [5] NHK Service Center., *Showashi 9 Saisho Retsuden* (The Showa Era: the History of Prime Ministers), PIONEER LDC Corp., Japan, 1987.