

圧縮映像データベースシステムにおける 映像演算と実現手法

小川政行 石川佳治 植村俊亮

奈良先端科学技術大学院大学 情報科学研究科

本論文では、圧縮映像のデータベースシステム実現を目的として、映像操作のための論理構造及び演算、並びに圧縮映像の格納構造を提案し、実装について論じる。

映像情報の論理構造及び演算に関しては、生映像、場面映像、合成映像に構造化して、この構造化に基づいた映像の生成／検索のための演算を提案する。また、映像のデータ量の大きさを考慮して、圧縮映像データを対象とし、圧縮映像特有の構造と前述の映像の論理構造との同期をとるための圧縮映像の格納構造を定義し、圧縮映像データベースシステムの提案を行なう。

Operation Mechanism and Implementation of a Video Database System with Compressed Data

Masayuki OGAWA, Yoshiharu ISHIKAWA and Shunsuke UEMURA

Graduate School of Information Science,
Nara Institute of Science and Technology (NAIST)

In this paper, we propose a video database system. We describe the logical structure and operations of the video data model, and propose physical storage structure of compressed video to implement our video database system.

The proposed video operations for logical video structure fall into three categories: for raw videos, for scene videos, and for composite videos. We discuss the synchronization problem of logical and physical video data.

1. はじめに

ハードウェアの高速／高性能化，2次記憶媒体の大容量化，動画像圧縮や高速ネットワーク技術の進歩により，マルチメディアを対象としたデータベースの実現が可能な環境になりつつあり，それらに対する要求も高まってきている。しかし，各コンテンツの持つ様々な特徴により，多くの技術課題が存在し，その実現は困難を極めている。そこで，本研究では，対象とするコンテンツとして映像情報に焦点をあて，映像データベースシステムの実現を試みる。

映像データベースの応用例として Video-on-Demand(VOD)が考えられるが，これは，映画などの題名検索，登場人物検索などにより，見たい映画を検索し，先頭から順次視聴するような，オンラインレンタルビデオのような形式が一般的である。しかし，これでは，利用者の多様な要求に応えることはできない。映像情報をデータベース管理システムで管理できれば，任意の映像の任意の場面を取り出したり，複数の場面をつなぎ合わせて利用者独自の映像を編集／視聴したりというようなことが可能になる。

しかし，その情報量は膨大なので，映像情報をそのままデータベース管理システムに格納することは，現実的ではない。そこで，圧縮映像を情報源とする。この場合，その特有の構造のため，非圧縮映像の場合に意識する必要のなかった論理構造と物理構造の同期を考えなければならない。

本論文では，映像の論理構造の構造化を行ない，それらに対する演算を提案する。また，圧縮映像を利用した場合の論理構造と物理構造の同期問題を論じ，圧縮映像をデータベース管理システムに格納する場合の構造を定義する。そして，以上の機能を有する圧縮映像データベースシステムの提案を行なう。

以下，2章では，関連研究について述べ，3章で，映像の論理構造の構造化とそれらに対する演算の定義について述べる。4章では，圧縮映像を利用した場合の問題点をあげ，圧縮映像特有の格納構造を定義し，圧縮映像データベースの提案について述べ，最後に，5章で，今後の課題について触れる。

2. 関連研究

大本ら [OT93] は，動画を一つのオブジェクトとして扱い，時間区間の包含関係をオブジェクト指向モデルの継承としてとらえ，属性値に動画を用いることが可能な OVID データモデルを提案した。また，その問合せ言語として，VideoSQL を提案している。

増永 [Mas94] は，時間依存属性を導入し，オブジェクト間に Allen の 13 種の時間関係 [All83] を導入したモデル OMEGA を提案し，時間依存拡張オブジェクト参照法と呼ばれるマルチメディアデータベースの構築法を提案した。

Little ら [LG93] は，Allen の時間関係を n 項時間関係に拡張し，木構造を用いて，各ノード毎にマルチメディアデータ間の時空間的な関係を持たせることで，同期などを表現している。

Weiss ら [WDG95] は，映像を構造化するために，video algebra という代数を提案し，これを使って入れ子状に映像を定義していくことで，表示までも考慮した様々な合成映像を容易に作成でき，検索を可能にするモデルを提案している。

しかし，これらの研究者による提案は，映像の論理的な構造，意味的な構造だけに終始しており，物理的な構造については議論していない。非圧縮映像や静止画での圧縮映像の場合はそれでもよいが，これが国際標準である MPEG [安田 94] などを対象とすると，物理的な構造の議論が必要になる。本稿では，MPEG を対象とした圧縮映像データベースシステムについての議論を行なう。また，映像の特性による構造化を行ない合成映像というレベルを考慮することで，利用者が容易に独自の映像編集を行なうことを可能にする映像演算を提案する。

3. 映像の論理構造

まず，映像の論理的なレベルの議論を行なう。映像は，一般に，フレーム，カット，シーンという構造からなると考えられ，画像処理の技術によるカットの自動検出による自動索引づけなどの手法も提案されている [有澤 93]。しかし，ここでは，利用者から見て意味のある単位，シーン（場面）についての議論を行ない，意味のある映像情報の構造化をする。

本論文では，映像は，生映像，場面映像，合成映像から構成される [小川 95] (図 1)。それぞれの映像は次の特徴を持つ。

【生映像】 一つの線形時間軸を形成する映像素材で，VTR テープなどがこれに相当する。つまり，TV 番組を製作するさいの各映像素材などがこれに相当する。また完成した番組なども生映像とみなせる。

【場面映像】 生映像の任意の時区間に，利用者が意味づけを行なって定義する映像。一つの線形時間軸上での情報であり，時間的な問合せが可能である。

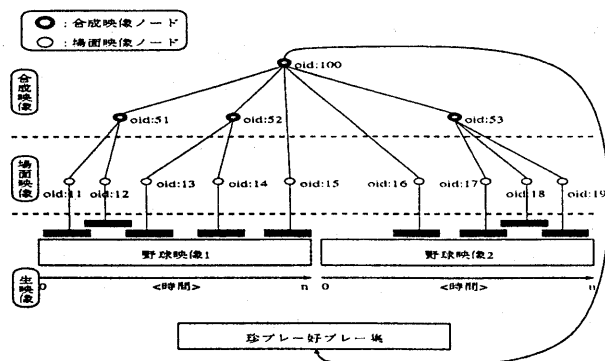


図 1 映像の構造

【合成映像】 利用者が場面映像や合成映像から映像を合成し、意味づけをして定義する映像。これにより、映像の編集が可能であり、映像間の入れ子構造を表現できるので、その参照関係を利用した映像の間合せが可能となる。また、任意の合成映像は、生映像とみなすことができ、再帰的に場面映像、合成映像を定義できるものとする。

3.1 映像演算の定義

3.1.1 場面映像

場面映像は、一つの映像素材上のそれぞれで定義する。場面を定義する場合に、従来のモデルでは、(開始時間、終了時間、意味記述)という三組の形式を取り扱うことが多い([OT93][WDG95]など)。ここでも、これを取り入れる。

この段階では、場面映像を定義するための演算を次のように定義する。

- (1) *define*: 場面映像属性集合生成

```
define(schemaname, {attribute})
```

schemaname は、生成した属性集合名。第2引数は属性集合で、*attribute* は属性である。

- (2) *create*: 場面映像生成

```
create(oid, schemaname, st, et,
      {attribute : {value|objref}})
```

oid は、生映像の識別子。 *schemaname* は、定義されている属性集合名を指定。 *st, et* は、開始/終了時間。 *attribute* は、映像の意味記述で、

指定した属性集合で定義してる属性のみ指定可能で、値 (*value* は文字列) か他のオブジェクトへの参照値 (*objref* は *oid*) を指定できる。また属性値は集合である。この指定により、場面オブジェクトが生成され、一意な識別子 (*oid*) が与えられる。

- (3) *delete*: 場面映像削除

```
delete(oid)
```

場面映像 (*oid*) の削除をする。

- (4) *update*: 場面映像更新

```
update(oid, {(st, et)},
      {attribute : {value|objref}})
```

各引数の意味は *create* と同じである。 $\{(st, et)\}$ で開始/終了時間の組の集合を指定する。

3.1.2 合成映像

合成映像は、複数の場面映像、または合成映像から構成され、複数の生映像にわたってもよい。

以下に、合成映像を定義するための演算を次のように定義する。

- (1) *compose*: 合成映像生成

```
compose({(oid, st, et)},
      {attribute : {value|objref}})
```

(*oid, st, et*) はオブジェクト識別子、開始/終了時間の組。 *attribute* は、前述と同様。ただし、第1引数には、後述する合成式を使用できる。

この指定により、合成映像オブジェクトが生成され、一意なオブジェクト識別子 (oid) が与えられる。

- (2) *delete*: 合成映像削除
場面映像と同様。
- (3) *update*: 合成映像更新

```
update(oid, {(oid, st, et)},  
       {attribute : {value|objref}})
```

引数の意味は、*compose* と同様。

合成用の演算式は、

- (1) *intersection*: 二つの映像の論理積

```
intersection(oid, oid)
```

任意の二つの場面の時間的共通場面の (oid, st, et) の組の集合を返す。この演算で合成映像を生成した場合は、次のように属性を継承する。値の場合は、両方の場面の属性を継承。オブジェクト参照の場合は、この演算結果の時間区間と参照オブジェクトの時間区間の重複があれば継承する。

- (2) *union*: 二つの映像の論理和

```
union(oid, oid)
```

任意の二つの場面の時間的結合場面の (oid, st, et) の組の集合を返す。ただし、共通部分の繰り返しは行なわない。この演算で合成映像を生成した場合は、次のように属性を継承する。値、オブジェクト参照のどちらの場合も、両方の場面の属性を継承する。

- (3) *concatenation*: 二つの映像の連結

```
concatenation(oid, oid)
```

任意の二つの場面の時間的結合場面の (oid, st, et) の組集合を返す。ただし、共通部分の繰り返しを行なう。この演算で合成映像を生成した場合は、次のように属性を継承する。値、オブジェクト参照のどちらの場合も、両方の場面の属性を継承する。

- (4) *difference*: 二つの映像の論理差

```
difference(oid, oid)
```

任意の二つの場面の時間的差分場面の (oid, st, et) の組の集合を返す。第1引数の属性を継承す

る。ただし、オブジェクト参照の場合は、この演算結果の時間区間と参照オブジェクトの時間区間の重複がなければ継承しない。

3.1.3 問合せ

場面映像には、時間的問合せ、内容に基づく問合せが行なえる。また、合成映像には、時間情報に基づく問合せは不向きなので、内容に基づく問合せと入れ子構造の参照関係を用いた問合せが可能である (図2)。

- (1) *query*: 時間的関連を用いた問合せ

```
query(結果, 範囲条件, 時間関連条件)
```

範囲条件は、以下の *content* 演算を使って探索場面の範囲を指定する。時間関連条件は、[All83]の時間関連を用いて、映像間の時間関連を指定する。

- (2) *content*: 内容に基づく問合せ

```
content(属性 | schemaname = 値)
```

場面／合成映像を探索し oid の集合を返す。

- (3) *referTo*: 参照元映像探索

```
referTo(oid)
```

映像 oid を参照している映像の oid の集合を返す。

- (4) *referredBy*: 参照要素映像探索

```
referredBy(oid)
```

映像 oid が参照している映像の oid の集合を返す。

また、問合せ結果に対する演算として、任意の二つの場面集合間の論理積 (*intersection*)、論理和 (*union*)、論理差 (*difference*) を行なう演算も導入する。

3.1.4 利用例

例えば、プロ野球中継ライブラリに対して場面映像、合成映像を生成する (図2)。

これらに対して、「松井のホームランの後に落合がホームランを打った場面検索」という検索を行なう (図3)。

```

define("ホームラン", {batter, pitcher, point})
defnc("ファインプレー", {fieldplayer, batter, pitcher})
....
a1 = create(v1, "ホームラン", 12, 13, {batter= "松井", pitcher= "今中", point= 3})
a2 = create(v1, "ホームラン", 15, 16, {batter= "落合", pitcher= "今中", point= 1})
....
b1 = compose(union(a1, a2), title= "アベックホームラン")
....

```

図 2 映像生成

```

query(x1, x1 = content(schemaname = "ホームラン") and
      x2 = content(schemaname = "ホームラン") and
      y = content(player = "落合") and
      z = content(player = "松井"),
      mccts(z,y) and during( x1, y ) and during( x2, z ))

```

図 3 問合せ

4. 圧縮映像の物理構造

4.1 圧縮映像導入の動機

非圧縮の映像を対象にする場合は、映像は静止画の順序集合とみなすことができ、物理的にも論理的にも各静止画は 1 対 1 に対応するので、前節の議論までで十分である。しかし、映像データのサイズは膨大（例えば、2 時間の TV 映像の非圧縮時の容量は 90GB に達する）であるので、映像ライブラリを想定すると、そのままでは 10 巻ほどの映画でたちまちテラバイトの規模になってしまう。そこで、本論文では圧縮映像（国際標準化された MPEG1[安田 94]）を対象とすることとし、これを用いた場合における問題を検討する。

4.2 MPEG データの構造とその問題

まず、最初に本論文で対象とする圧縮映像 MPEG1 の構造とその問題について述べる。以降、特に断わらないかぎり MPEG という。

4.2.1 MPEG 映像ストリーム構造

MPEG 映像は、それぞれ符合化方式の異なる、以下の三種類のフレームから構成される（図 4）。

- (1) **I-Picture**: 1 枚のフレーム自身のみから符合化を行なうフレーム内符合化フレーム

- (2) **P-Picture**: 直前の I/P-Picture から予測符合化を行なう順方向予測符合化フレーム
- (3) **B-Picture**: 直前、直後の I/P-Picture から予測符合化を行なう双方向予測符合化フレーム

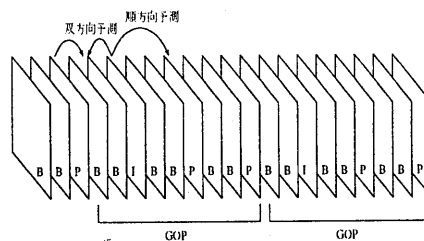


図 4 MPEG による映像の構造

これらそれぞれのフレームを VAU (Video Access Unit) といい、これらのフレームは複数集められて、GOP (Group of Picture) を構成する。I-Picture は、フレーム内符合化を行なっているの、単独復号可能であるが、P/B-Picture は、フレーム間予測符合化を行なっているの、単独復号が不可能である。

したがって、圧縮映像の処理では、次の問題を解決しなければならない。

- (1) フレームの再生順序と格納順序の相違
任意の場面の検索／再生を考えると、非圧縮映像の場合は単純に検索により得られた開始フレームから終了フレームまでの再生を行えば良いが、圧縮映像の場合は、フレームの再生順と格納順が異なるため、単純に開始／終了フレーム間を復号／再生することは困難である。

- (2) 任意フレームの単独再生不能

I-Picture 以外のフレームは、その予測符号化を行っているため、復号には他の前後のフレームを必要とする。また、前述した GOP という単位を導入することでランダムアクセスを可能にしているが、これに関しても GOP の先頭フレームが、前の GOP のフレームに依存している場合、GOP での単独再生も困難になる。ただし、クローズド GOP フラグ、ブローケンリンクフラグを用いて、これらの問題を解消するすべはある。

4.2.2 MPEG 音声ストリーム構造

MPEG 音声も MPEG 映像と同様に、フレームという概念が存在し、このフレームが、AAU(Audio Access Unit)といい、それぞれ単独復号可能な単位である。これは、映像との同期という意味で問題があるが、これに関しては次に述べる。

4.2.3 MPEG システムストリーム構造

MPEG 映像ストリームや MPEG 音声ストリーム、および付加情報を応用で利用する場合は、同期や物理形式を考慮して多重化する必要がある。MPEG システムは、多層物理構造を持ち、MPEG 映像と MPEG 音声をバケット見出しを伴いバケットを構成する。この時、各バケット見出しは、映像／音声の識別情報や復号／再生のための時刻管理情報が記述される。そして、これらのバケットは、複数集められて、バック見出しを伴いバックを構成する。この時、各バック見出しには、映像と音声の同期再生に用いられるタイムスタンプや復号器の基準時刻参照値 SCR(System Clock Reference) 等が記述される。そして、これらのバックが複数集まって、システム見出しを伴い、MPEG システムストリームを構成する。システム見出しにはシステム全体の管理方法などが記述される。

よって、その問題点として、バック、バケットのデータ長が可変／固定のどちらでもよいという仕様により、

- (1) 音声／映像フレームの先頭が、バック、バケットがの途中に現れるので、任意バックからの再生ができない。

- (2) 映像と音声の同期を考えると、最低バック単位でデータを扱う必要があるが、任意のバック内に存在する映像バケットとそれと同期をとるべき音声バケットが同じバック内に存在しない可能性がある。

が挙げられる。

4.3 これまでの研究

[友田 95] では、MPEG 特有の構造に対応するために、MPEG データベースのスキーマ設計を行なった。それは、論理スキーマ、物理スキーマ(1):映像／音声別のレベル、物理スキーマ(2):映像／音声の同期レベルという3層で構造をとらえ、格納／検索を考慮したスキーマを提案した。非圧縮映像の場合は、利用者により指定された任意の場面の時区間(開始時間、終了時間)を(開始フレーム、終了フレーム)に対応づければよかった。しかし、圧縮映像の場合はその特性から任意のフレームの抽出が困難なため、さらに、(開始 GOP、終了 GOP)から(開始バック、終了バック)へ変換し、場面を抽出することを可能にした。しかし、本研究では MPEG システムストリームを商用データベース管理システムが提供している BLOB(Binary Large Object) クラスを用いて格納することを前提としているため、現状の商用データベース管理システム上での実装を考えると、そのデータ量の膨大さから実現は困難と考えられる。ここでは、この問題を考慮して、映像のランダムアクセスの単位である GOP に着目し、これを BLOB クラスとして格納することを考える。

4.4 MPEG データベース

4.4.1 格納時の構造変更

そこで、本論文では、図 5 のように MPEG ストリームを階層化し、図の GAU(Group of Access Unit)でストリームを BLOB クラスとして格納する。GAU とは、ここで定義した単位で、映像のランダムアクセスの単位 GOP を規準とし、同期するべき音声も含んだ単位である。システムストリームの下位に GAU の順序集合、GAU の下位に AU(Access Unit)(映像または音声の 1 フレーム)の順序集合、AU の下位にバックの順序集合、バックの下位にバケットの順序集合、そして、バケットは映像バケットもしくは音声バケットである。

この構造でデータを格納するため、以下のような処理を施す。

- (1) 各 AU の見出しは、バケット見出しの直後に位

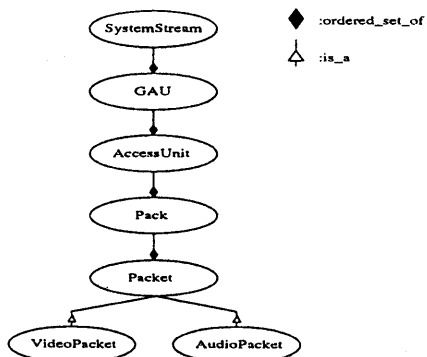


図 5 MPEG ストリーム階層構造

置させる (図 6)。

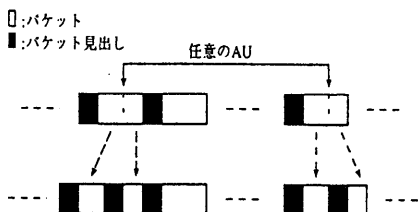


図 6 アクセスユニットの変更

- (2) I-Picture は、高速再生用に I-Picture のみでの再生を可能にするため、バックで完結する (図 7)。

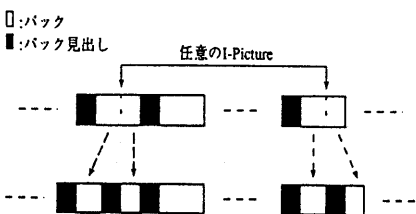


図 7 I-Picture の変更

- (3) 各映像ストリーム見出しは、バック見出しの直後に位置する。GOP でストリームを完結する (図 8)。映像ストリーム見出しがないものは先頭につける。

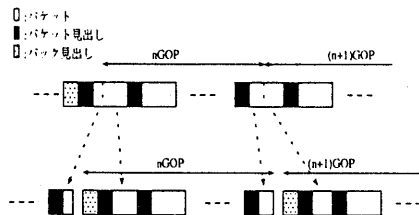


図 8 GOP の変更

- (4) GOP 見出しにある、前の GOP との参照関係を記述するフラグを操作する機能が必要 (クローズド GOP フラグの立っている場合は、なにもしない。そうでない場合は、再生シーケンスの先頭であれば、ブローケンリンクフラグを立てる) (図 9)。

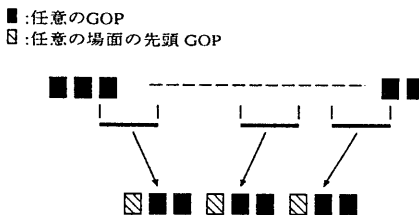


図 9 GOP 単位の編集

- (5) 任意の同期すべき映像/音声パッケージは、同じ GAU 内で完結する (図 10)。

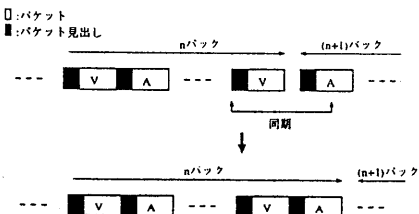


図 10 音声パッケージの変更

- (6) システムストリーム見出しは、独立したストリームとする。

4.4.2 検索処理

前述した形式で、MPEG データを格納すると、任意の場面を検索して取り出す場合の間合せ時には次

のような処理が行なわれる。

- (1) 利用者により要求映像の問合せが行なわれる。
- (2) 問合せ演算に変換されて、データベースへ問合せが行なわれる。
- (3) データベース側は問合せに基づき候補オブジェクトを提示する。
- (4) 利用者が、任意の映像オブジェクトを選択する。
- (5) その選択により（開始時間、終了時間）の集合が得られる。つまり、映像オブジェクト生成時には物理構造の単位で生成しているので、（開始GAU、終了GAU）の集合が得られる。
- (6) GAUの順序集合を取り出す。
- (7) 要求映像の各GAU集合（元映像において連続）の先頭GAUの編集フラグについての処理を行なう。つまり、クローズドGOPフラグが立っていないければ、ブローケンリンクフラグを立てる。
- (8) システムストリーム見出しを先頭バック見出しの直後に挿入する。
- (9) 要求映像を送出する。

つまり、格納時にデータ再構成を行ない、この際に映像と音声の同期用のタイムスタンプ等の変更を行なっているため、検索時のコストを軽減できる。

5. まとめ

映像の構造化を行ない、それらに対する演算を定義した。さらに、圧縮映像を用いた場合の論理構造と物理構造の非同期性に着目し、圧縮映像の格納構造を提案した。これにより、圧縮映像を用いた映像データベースシステムの実現手法を確立した。

最後に今後の課題を挙げる。

- (1) 実装を行ない、GOP単位でストリームを格納した場合の妥当性の検証を行なう。
- (2) 実用段階を考慮すると、MPEG2への拡張が必須である。MPEG2の場合は、蓄積メディア用のMPEG2-PS、通信／放送メディア用のMPEG2-TSの2種類があるが、我々はデータベースというデータの蓄積を目的としているので、MPEG2-PSの統合を考えていく。

謝辞： 日頃から有益な御指導・御討論を頂く、植村研究室の皆様へ深謝いたします。

本研究の実施にあたり、EAGL事業推進機構からの助成を受けました。ここに謝意を表します。

参考文献

- [All83] J. F. Allen: "Maintaining Knowledge about Temporal Intervals", *Comm. of the ACM*, Vol. 26, No. 11, pp. 832-843, November 1983.
- [GBT94] Simon Gibbs, Christian Breiteneder, and Dennis Tsichritzis: "Data Modeling of Time-Based Media", in *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 91-102, May 1994.
- [LG93] T. D. C. Little and. Ghafoor: "Interval-Based Conceptual Models for Time-Dependent Multimedia Data", *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 4, p. 551, August 1993.
- [Mas94] Y. Masunaga: "Temporal Multimedia Data Modeling in OMEGA", in *Proc. of the International Symposium on Advanced Database Technologies and Their Integration*, pp. 190-199, October 1994.
- [OT93] E. Oomoto and K. Tanaka: "OVID: Design and Implementation of a Video-Object Database System", *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 4, pp. 629-643, August 1993.
- [WDG95] Ron Weiss, Andrzej Duda, and David K. Gifford: "Composition and Search with a Video Algebra", *IEEE Multimedia*, Vol. 2, No. 1, pp. 12-25, 1995.
- [安田 94] 安田 浩: MPEG/マルチメディア符号化の国際標準, 丸善, 1994年9月。
- [小川 95] 小川 政行, 石川 佳治, 植村 俊亮: "映像データベースにおける代数的データモデルの提案", 情報処理学会第51回全国大会, 5D-6, 1995年9月。
- [堀内 95] 堀内 優希, 友田 政明, 石川 佳治, 植村 俊亮: "映像データベースのための論理データモデルとその実装", 電子情報通信学会第6回データ工学ワークショップ, 1995年3月。
- [友田 95] 友田 政明, 堀内 優希, 石川 佳治, 植村 俊亮: "圧縮映像データの格納と検索", 情報処理学会第101回データベースシステム研究会 研究報告, 1995年1月。
- [有澤 93] 有澤 博, 由井 仁, 富井 尚志: "映像データベースシステムの構成の方式", in *Proceedings of Advanced Database Symposium '93*, pp. 181-190, 1993年12月。