

ビデオデータ検索のための区間グルー操作 と解のフィルタリング

プラダン スジット† 田島 敬史 †† 田中 克己 †††

ビデオ・データの特徴として、一時間のビデオ・データでさえ、意味のあるユニットが多数含まれているという点がある。手作業で全てのそのようなユニットを識別することは、ほとんど不可能である。一方、画像処理、音声や文字認識、自然言語処理のような技術の統合によるビデオ・データの自動解析やインデックス付けは可能であるが、そのような技術を適用しても、インデックスは断片的なビデオ・ユニットにしか付けることができない。この結果、ユーザが欲する全ての区間を予めデータベースに格納することは不可能である。従って、ビデオ・データベースでは、インデックスが付けられた区間の粒度と検索される区間の粒度間の相違が問題になっている。この問題を解決するために、本論文では、インデックスを付けられたビデオ・ユニットから答えの区間を動的に合成する問い合わせ機構を提案する。この問い合わせ機構では、幾つかのグルー (glue) 演算を定義し、これらの演算を用いて、ある問い合わせに対する最も適した区間の集合を返すことができる。しかし、それらの答えの集合の中には、その問い合わせに対して不適切な区間が多く含まれる可能性がある。そのような不適切な区間を減らすために、我々は、区間フィルタの概念を提案する。区間フィルタとは、インプットされた区間の集合からある条件を満たす区間の部分集合だけを返す関数である。

Interval Glue Operations and Answer Filtering for Video Data Retrieval

SUJEET PRADHAN ,† KEISHI TAJIMA †† and KATSUMI TANAKA †††

The nature of video is such that even an hour long video data may contain a large number of meaningful units. Manual identification of all such units can rarely be realized. There has been some success in automatically parsing and indexing video data through the integration of technologies such as image processing, speech/character recognition, and natural language understanding¹¹⁾. However, even by applying such techniques, it is only fragmentary video units that can be indexed. As a result, using the current state-of-art techniques, whether automatic or manual, complete identification of all the intervals required for answering all possible queries cannot be achieved. This is the reason why the problem of discrepancies between the granularity of the intervals that have been indexed and the granularity of the intervals that are expected to be retrieved has remained prevalent in video databases. In order to tackle this problem, a query mechanism which is capable of computing answer intervals dynamically from the indexed video units is proposed in this paper. We describe an algebra consisting of a set of *glue operations* used for synthesizing answer intervals to query. We also propose a set of *interval filters* to exclude intervals that are irrelevant to the query from the answer set.

1. はじめに

ビデオの効率的な検索のためには、通常、生のビデオ・データはより小さいユニットに分割する必要がある

。現在、ビデオ・データは物理的に分割するのではなく、論理的に分割することが普通になっている。論理的な分割により、データを再利用可能な意味的なまとまりに分割することができる¹⁰⁾。次に、そのような分割されたビデオデータには、自動キーワード抽出、あるいは手作業によるオーサリングのような様々な方法でインデックス付けを行う。しかし、自動的なものであれ、手作業によるものであれ、現在の技術では、インデックスは断片的にしか付けることができない。その結果、これらの断片的なインデックスでは、エン

† 神戸大学自然科学研究科
Div. of Intelligence Science, Kobe Univ.

†† 神戸大学工学部
Dept. of Computer & Systems Engineering, Kobe Univ.

††† 神戸大学自然科学研究科
Div. of Media & Information Sciences, Kobe Univ.

ド・ユーザが欲する全ての区間を必ずしも検索できるとは限らない。たとえば、ショットとシーンに自動分解されたビデオ・ストリームを考へてみる³⁾¹⁴⁾。ある問い合わせに対して、答へとして期待されている区間は、必ずしも一つのショット、また一つのシーンではないかもしれない。実は、答への区間は、いくつかの連続的なショットにわたる可能性がある。したがって、ビデオ検索システムでは、インデックスを付けられたビデオ・ユニットの粒度だけでは全ての問い合わせの解を用意できない。

従来のビデオ検索システムでは、答へとなり得る区間を予め識別し、データベースに格納することになっている⁹⁾⁴⁾¹²⁾。しかし、一時間のビデオでも、答へになり得る区間は多数含まれていることが考えられる。さらに、ビデオ・データは、多様に解釈できるため、インデックスを付ける人にとって意味のある区間と、問い合わせをする人にとって意味のある区間は必ずしも一致するとは限らない。実際問題として、手作業によるビデオの内容記述は手間がかかり、しかも不完全である。これらの理由のため、ビデオ・データベースでは、インデックスを付けられた区間の粒度と、検索される区間の粒度間の相違があると考えられる。従って、ただ断片的なデータ・ユニットにインデックスを付けるだけでは、ビデオ・データベースにおける解の粒度の問題は解決されたことにはならないと考えられる。

ビデオ・データベースのこの基本的な問題を解決するために、我々はインデックスを付けられたビデオ区間群から答への区間を動的に合成するビデオ問い合わせ機構を提案する。我々はデータベースの中にはキーワード付けされた断片的なデータは既にあることを仮定する。それらのキーワードは、図1で示すように、ビデオ・データそのものに付いているシナリオや字幕や音声の自動認識によって抽出したもの、あるいは手作業によるオーサリングのような方法で加えられたものである。ユーザは単にキーワードの集合によって問い合わせを表現することができる。また、ここでは問い合わせの解としてユーザーが求めているのはビデオ・データの中のある連続した部分区間であると仮定する。

本論文では、グルー (glue) 演算と呼ばれる区間演算群を定義する。これらの演算子を用いると、インデックス付けされたビデオ区間群を時系列的に連結することによって、あるビデオ問い合わせに対して答へになり得る全ての区間を生成できる。しかしながら、このように合成された答への集合の中には、その問い合わせに対して不適切な区間も多く含まれる可能性がある。

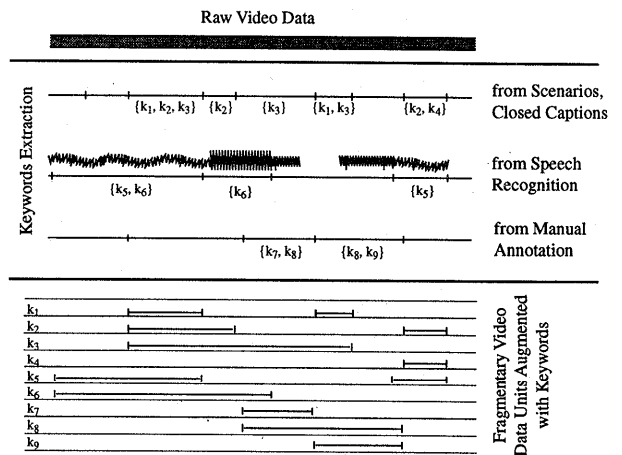


図1 キーワード抽出技術
Fig. 1 Keywords Extraction from Various Techniques

情報検索や類似検索システムなどでは、このような答へのオーバーローディングを扱うためにランキングの技術を適用するのが一般的である⁸⁾が、それぞれの答への適切さを評価する明確な方法がなければ、システムは答を正確に提示することができない。そこで、我々は答への区間集合から不適切な区間を取り除くために、区間フィルタの概念を提案する。区間フィルタとは、インプットされた区間集合からある条件を満たす区間だけを返す関数である。

本論文の主な内容は以下のものである。

- (1) キーワード集合によるビデオ問い合わせの意味について論じる。
- (2) 答への区間を合成するために幾つかのグルー演算を定義し、その代数的な性質について述べる。
- (3) ビデオ問い合わせに適した様々な区間フィルタを定義し、その代数的な性質について述べる。
- (4) そのようなフィルタは我々のグルー演算と組み合わせられた場合、正しい結果が得られるかどうかを考察する。

以下、2章で本研究の動機について述べる。3章ではビデオ・データベースの基本的な定義を示し、キーワード集合によるビデオ問い合わせのセマンティックスを定義する。そして、4章では様々なグルー演算を定義し、それらの性質について述べる。5章では、ビデオ問い合わせに適した幾つかの区間フィルタを定義し、それらのフィルタをグルー演算と組み合わせられた場合、検索結果に与えられる影響について述べる。6章では関連研究について述べる。7章で、区間フィルタに関する問題を論議する。最後に、8章では本論文の主な貢献を述べてまとめとする。

2. 動 機

従来から、ビデオ区間集合から新しい区間を生成するために、intersection や union などの様々な区間演算が定義されている⁹⁾⁵⁾⁴⁾¹²⁾。しかし、これらの演算だけでは、必ずしもユーザが求める区間を生成することができない。例えば、「犬を訓練している人のビデオ区間を検索する」場合の問い合わせを考えてみよう。従来のアプローチでは、単に「人」、「訓練」、「犬」というキーワードが付いているそれぞれの区間の intersection が問い合わせの結果として返されるが、「犬を訓練している人」の実際の映像を作成するときは、たくさんのカメラの動き、ズームやパンのような動作を含むのが通常である。従って、そのシーンの中のあらゆるフレームに必ず「犬」と「人」が同時に現れるのはむしろまれである。それらのいくつかのフレームに「人」か「犬」かどちらかが現れたり、「人」と「犬」のどちらも現れず曇り空のみが現れたりするのが普通である。したがって、もともとユーザは答えの区間のあらゆるフレームの中で、必ず「犬」や「人」が現れることを期待していない。

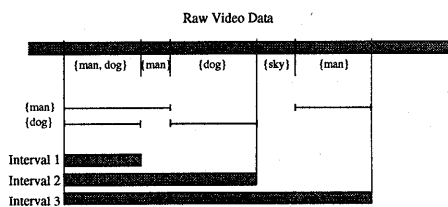


図2 ビデオ問い合わせのセマンティックス
Fig. 2 Video Query Semantics

このような単純な問い合わせにさえ、その問い合わせに対する適切な区間を生成するのに新しい区間演算が必要となる。我々の目的はビデオ・データベースの上でそのような問い合わせに対する答えを生成できるような演算を開発することである。例えば、図2はキーワード「犬」と「人」を含む問い合わせの例である。ただし、ビデオ・データ列の下の犬(dog), 人(man), 空(sky)はそれぞれの区間に断片的に示されたキーワードを示している。以下のような幾つかの区間がその問い合わせの答えになり得る。Interval 1 はあらゆるフレームに「犬」と「人」が同時に現れている区間である。Interval 2 はあらゆるフレームに「犬」か「人」かどちらか現れている区間である。Interval 3 の中の幾つかのフレームには、「犬」と「人」もどちらも現れてないが、意味的なまとまりの区間である。我々は

Interval 3 のような答えを生成するために区間グループと呼ぶ演算を定義する。ここで注意すべきことは、普通の区間 intersection 演算は Interval 1 だけを生成することができ、普通の区間 union 演算は Interval 2 を生成することができるが、Interval 3 を生成することができない。

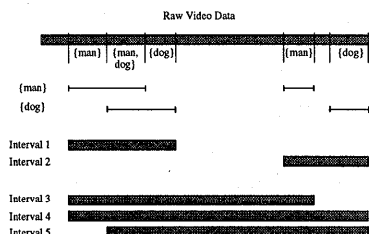


図3 不適切な区間が含まれている答えの集合
Fig. 3 Answer Set Containing Irrelevant Intervals

そのような区間演算を定義する際の1つの問題は、「ノイズ」、すなわち問い合わせ中のどのキーワードともマッチしない部分区間の存在である。前述の問い合わせの例では、我々がその問い合わせのセマンティックスを厳しく解釈するならば、そのように「ノイズ」が含まれている区間は、その答えにはならない。しかし、ビデオ・データベースの実際的なアプリケーションでは、問い合わせのそのような厳しい解釈は、上で述べたように意味をなさない。一方、我々がその問い合わせのセマンティックスをゆるめるならば、本来不適切な区間までが解に含まれてしまう。この結果、答えの集合から妥当でないと考えられるな区間を取り除かなければならない。たとえば、図3では、「犬」と「人」に対して、我々のグループ演算では、5つの答えが合成される。直観的に、Interval 1 と Interval 2 だけはその問い合わせに適切である。その他の区間は「ノイズ」をかなり含んでいるため、それらは意味がある区間とは想定しがたい。最悪の場合、もし問い合わせ中の2つのキーワードがビデオの最初のフレームと最終のフレームに出てくるならば、そのビデオ全体も答えになる可能性がある。我々は、このような長い不適切な区間を解集合から排除させることができるような区間フィルタを提案する。

3. 基 本

ビデオ・データベースは、一般には、生のビデオ・ストリームの集合である。ここでは単純にそのような生のビデオ・ストリームが1つ与えられているものとする。一つのビデオ・ストリーム F は、空でない有

限のフレーム集合であり、その集合は全順序集合である。 $f_s, f_e \in F$ かつ $f_s < f_e$ ならば、 F 上のビデオ区間 $i[f_s, f_e]$ を $\{f_k \in F \mid f_s \leq f_k \leq f_e\}$ と定義する。我々は任意の区間 i の最初のフレームと最後のフレームそれぞれを $f_s(i)$ と $f_e(i)$ で現わす。このようにビデオ区間 i は、連続するフレームの系列であり、 $f_s(i)$ と $f_e(i)$ によってユニークに識別できる。区間は、 $[f_s, f_e]$ を省略しても構わない場合には、単に i と書くことにする。 $I(F)$ は F の中の全ての区間の集合を示す。また、 $i_1 \subset i_2$ であれば、 i_1 は i_2 の『部分区間』と呼ぶ。

F に対するビデオ記述データベースを次のように定義する。

$$VD = \{(k_1, i_1), (k_2, i_2), \dots, (k_n, i_n)\}$$

ここで各 (k_i, i_i) は F に対する一個のビデオ記述である。 (k_i, i_i) の中の k は一個のキーワードであり、 i は $I(F)$ の要素である。ここで (k, i) は、そのキーワード k がビデオ区間 i のあらゆるフレームに該当することを意味する。

3.1 キーワード集合によるビデオ問い合わせのセマンティックス

ビデオ・データベースの問い合わせは、従来のデータベースの問い合わせと異なる。従来のデータベースの問い合わせのセマンティックスは次のようなものであった。まずデータベースはある離散的なデータ・ユニットの集合であり、問い合わせ条件は、プリミティブな属性値と論理演算 (AND, OR, NOT) で表現される。そして、その問い合わせの答えは、その条件を満たすデータ・ユニットからなる、データベースの部分集合である。しかし、ビデオ・データベースの場合は、問い合わせの答えはそのデータベースにあるデータ・ユニットの部分集合ではなく、そのビデオ・データの任意の部分区間が解集合の要素となりうる。

我々のグルー演算を定義する前に、ここで我々が想定しているビデオ問い合わせのセマンティックスを明らかにする必要がある。この研究では、ユーザがなにかの意味を持っている連続したビデオ区間を検索することと仮定する。ユーザからの問い合わせは次のような形式で表現する：

$$\text{人} \wedge \text{訓練} \wedge \text{犬}$$

しかし、従来のビデオ・データベースと違って、問い合わせの結果は単にそのキーワードが付いている区間の intersection を取ってはならない。なぜなら、実際のビデオ・データベースの中では、3つのキーワードの全てが同時に現れる区間が存在する確率は非常に

低く、たとえそのような区間が存在するとしても、それは短い断片的なショットであり、そのような断片的なショットだけ抜き出したのではもとのビデオ・データの中で、そのショットが持っていた意味がわからなくなってしまうからである。したがって、上記の問い合わせをより実際のなアプローチで解釈するなら、『人』と『訓練』と『犬』の各キーワードそれぞれが少なくとも一つのフレームに現われる区間を求めているということになる。この結果、我々はそのような問い合わせのセマンティックスを以下のようにゆるめて表現する：

$$(\text{人})^{\exists} \wedge (\text{訓練})^{\exists} \wedge (\text{犬})^{\exists}$$

ここで限量子 \exists は各キーワードが答え区間の中で少なくとも1つのフレームに該当していればよく、すべてのフレームに該当する必要はないことを強調している。

4. Glue 演算

この章では、キーワード集合によるビデオ問い合わせに対して、インデックス付けされた区間から解の候補となる全ての区間を合成できるようなグルー演算を形式的に定義する。これらの演算の性質は、前章で述べた問い合わせのセマンティックスを反映するものになっている。

4.1 区間グルー

任意の区間 x と y に対して、区間グルー演算 \oplus は以下のように一つの新しい区間 i を作り出す。

$$x \oplus y = i[f_s, f_e] \text{ 但し}$$

$$f_s(i) = \min(f_s(x), f_s(y)) \text{ かつ}$$

$$f_e(i) = \max(f_e(x), f_e(y))$$

例えば、 $x[10, 20]$ と $y[30, 40]$ であれば、 $x \oplus y = i[10, 40]$ となる。

区間グルーの重要な代数的性質は以下の通りである。

- 可換則： $x \oplus y = y \oplus x$ (区間グルーの定義により明らか)
- 結合則： $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ 。これ以降、 $(x \oplus y) \oplus z = x \oplus y \oplus z$ と書く。
- べき等則： $x \oplus x = x$ (区間グルーの定義により明らか)

4.2 ペアワイズ・グルー

任意の区間の集合 X と Y に対して、ペアワイズ・グルー演算 \oplus を次のように定義する。

$$X \oplus Y = \{i \mid \exists x \in X. \exists y \in Y. i = x \oplus y\}$$

結果は入力集合から要素をペアワイズに取りながら、それらの要素に区間グルー演算を適用し、作り出した区間の集合である。

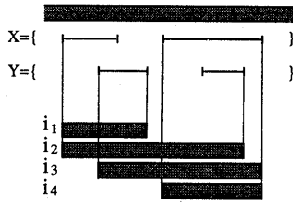


図4 ペアワイズ・グルー演算
Fig. 4 Pairwise Glue

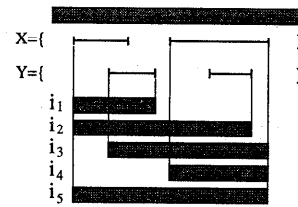


図5 XとYのパワーセット・グルー演算
Fig. 5 Powerset Glue Operation between X and Y

このペアワイズ・グルーも以下のような性質を持つ。

- 可換則： $X \oplus Y = Y \oplus X$ (ペアワイズ・グルーの定義により明らか)
- 結合則： $(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z)$

しかし、べき等則は成り立たない、すなわち $X \neq (X \oplus X)$ であることに注意を要する。例えば $X = \{i_1, i_2\}$ の場合、 $(X \oplus X) = \{(i_1 \oplus i_1), (i_2 \oplus i_2), (i_1 \oplus i_2)\} = \{i_1, i_2, (i_1 \oplus i_2)\}$ となり、 X とは異なる。

4.3 パワーセット・グルー

X と Y をビデオ区間の任意の集合とする。 X と Y のパワーセット・グルー演算を次のように定義する。

$$X \otimes Y = \{i \mid \exists X' \subseteq X. \exists Y' \subseteq Y. X' \neq \emptyset. Y' \neq \emptyset. i = \oplus(X' \cup Y')\}$$

但し $\oplus(\{i_1, \dots, i_n\}) = i_1 \oplus \dots \oplus i_n$.

これは、 X と Y から、各々0個でない任意の数個要素に区間グルー演算を適用して生成した区間の集合である。図5では、区間 i_1, \dots, i_4 は各々 X と Y の中の1つずつの要素にペアワイズ・グルーを適用して生成された区間である。区間 i_5 は X の中の2つの要素と Y の中の1つ、あるいは2つの要素にペアワイズ・グルーを適用して生成された区間である。幾つかの区間は Y の中の2つの要素と X の中の1つの要素にペアワイズ・グルーを適用して生成することも考えられるが、結果は i_2, i_3 と同じである。

上の定義から、 X と Y のパワーセット・グルー演算は次の式で計算できる：

$$\begin{aligned} X \otimes Y &= (X \oplus Y) \cup \\ & (X \oplus X \oplus Y) \cup \\ & (X \oplus Y \oplus Y) \cup \\ & (X \oplus X \oplus X \oplus Y) \cup \\ & (X \oplus X \oplus Y \oplus Y) \cup \\ & (X \oplus Y \oplus Y \oplus Y) \cup \\ & \vdots \end{aligned}$$

パワーセット・グルーのこの定義は複雑に見えるが、

これは以下のように3つのペアワイズ・グルー演算による式に変形でき、効率的に計算できる。

定理1 X と Y を区間の任意の集合とすると、次の式が成り立つ。

$$X \otimes Y = (X \oplus X) \oplus (Y \oplus Y)$$

証明： $U = X \otimes Y, V = (X \oplus X) \oplus (Y \oplus Y)$ とすると、 $U = V$ を証明するのに $U \supseteq V$ かつ $U \subseteq V$ を証明すればよい。まず $U \supseteq V$ をするために、任意の要素 $z \in V$ を考える。そうすると、 $x \oplus y = z$ であるような $x \in (X \oplus X)$ と $y \in (Y \oplus Y)$ が存在するはずである。区間グルーの定義により、 z の最初のフレームと最終のフレームについては以下の4つの場合が考えられる。

- I. $z \equiv z[f_s(x), f_e(y)]$ if $f_s(x) \leq f_s(y)$ and $f_e(y) \geq f_e(x)$
- II. $z \equiv z[f_s(y), f_e(x)]$ if $f_s(y) \leq f_s(x)$ and $f_e(x) \geq f_e(y)$
- III. $z \equiv z[f_s(x), f_e(x)]$ if $f_s(x) \leq f_s(y)$ and $f_e(x) \geq f_e(y)$
- IV. $z \equiv z[f_s(y), f_e(y)]$ if $f_s(y) \leq f_s(x)$ and $f_e(y) \geq f_e(x)$

I の場合、 $x \in (X \oplus X)$ なので $f_s(x') = f_s(x)$ かつ $f_e(x') \leq f_e(x)$ であるよう $x' \in X$ が存在するはずである。同様に、 $y \in (Y \oplus Y)$ なので $f_e(y') = f_e(y)$ かつ $f_s(y') \geq f_s(y)$ であるよう $y' \in Y$ が存在するはずである。さらに、 U は X と Y のパワーセット・グルー演算であるため、 $z' = x' \oplus y'$ であるよう $z' \in U$ が存在するはずである。

$f_s(x') = f_s(x)$ かつ $f_s(x) \leq f_s(y)$ かつ $f_s(y') \geq f_s(y)$ なので、 $f_s(x') \leq f_s(y')$ である。同様に、 $f_e(y') = f_e(y)$ かつ $f_e(y) \geq f_e(x)$ かつ $f_e(x') \leq f_e(x)$ なので、 $f_e(y') \geq f_e(x')$ である。その結果、 $z' \equiv z'[f_s(x'), f_e(y')]$ がいえる。 $f_s(x') = f_s(x)$ かつ $f_e(y') = f_e(y)$ なので $z' = z$ 。従って、 $z \in U$ が言える。図6に、これらの区間の位置関係を図にしたものを示す。

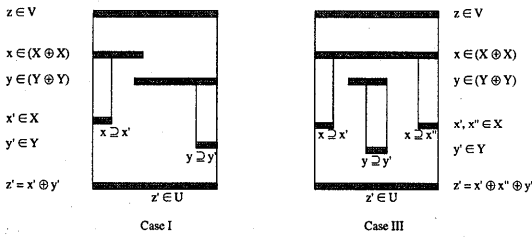


図 6 定理 1 の証明のための図
Fig. 6 Illustration for the Proof of Theorem 1

II の場合も同様に証明できる。III の場合は、 X の中の 2 つの区間と Y の中の 1 つの区間を考慮し、IV の場合は、 X の中の 1 つの区間と Y の中の 2 つの区間を考慮し、ほぼ同様に証明できる (図 6 右を参照)。従って、 $U \supseteq V$ 。

次に $U \subseteq V$ を示す。この場合もまず任意の要素 $z \in U$ を考える。 U は X と Y のパワーセット・グルー演算であるため、 $z = x \oplus y$ と書ける。但し、 x と y はそれぞれ X と Y の中の 1 つ以上の要素にペアワイズ・グルーを適用し作り出した区間である。ここでも区間グルーの定義により z の最初のフレームと最終のフレームについては以下の 4 つの場合が考えられる。

- I. $z \equiv z[f_s(x), f_e(y)]$ if $f_s(x) \leq f_s(y)$ and $f_e(y) \geq f_e(x)$
- II. $z \equiv z[f_s(y), f_e(x)]$ if $f_s(y) \leq f_s(x)$ and $f_e(x) \geq f_e(y)$
- III. $z \equiv z[f_s(x), f_e(x)]$ if $f_s(x) \leq f_s(y)$ and $f_e(x) \geq f_e(y)$
- IV. $z \equiv z[f_s(y), f_e(y)]$ if $f_s(y) \leq f_s(x)$ and $f_e(y) \geq f_e(x)$

I の場合、 x は X の中の 1 つ以上の要素にペアワイズ・グルーを適用し生成した区間であるため、 $f_s(x') = f_s(x)$ かつ $f_e(x') \leq f_e(x)$ であるような $x' \in X$ が存在するはずである。同様に、 y は Y の中の 1 つ以上の要素にペアワイズ・グルーを適用し生成した区間であるため、 $f_e(y') = f_e(y)$ かつ $f_e(y') \geq f_e(y)$ であるよう $y' \in Y$ が存在するはずである。

また、 $X \oplus X$ は X と自分自身とのペアワイズ・グルー演算であるため X の中の任意の要素が $X \oplus X$ の中にも存在する。従って、 $x' \in (X \oplus X)$ 。同様に、 $y' \in (Y \oplus Y)$ 。この結果、 $z' = x' \oplus y'$ であるよう $z' \in V$ が存在するはずである。

$f_s(x') = f_s(x)$ かつ $f_s(x) \leq f_s(y)$ かつ $f_s(y') \geq f_s(y)$ であるため $f_s(x') \leq f_s(y')$ である。同様に、 $f_e(y') = f_e(y)$ かつ $f_e(y) \geq f_e(x)$ かつ $f_e(x') \leq f_e(x)$ であるため $f_e(y') \geq f_e(x')$ である。従って、

$z' \equiv z'[f_s(x'), f_e(y')]$ 。 $f_s(x') = f_s(x)$ かつ $f_e(y') = f_e(y)$ なので $z' = z$ といえる。この結果、 $z \in V$ が言える。

II の場合も同様に証明できる。III の場合は、 X の中の 2 つの区間と Y の中の 1 つの区間を考慮し、IV の場合は、 X の中の 1 つの区間と Y の中の 2 つの区間を考慮し、ほぼ同様に証明できる。従って、 $U \subseteq V$ 。以上から $U = V$ が証明された。□

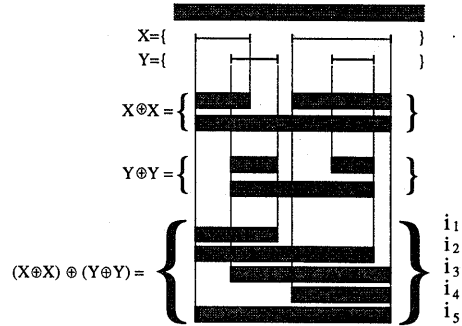


図 7 パワーセット・グルー演算の代数的な変換
Fig. 7 Powerset Glue's Algebraic Transformation

以下に例を示す。図 7 では、 $(X \oplus X)$ と $(Y \oplus Y)$ 演算がそれぞれ 3 つの区間を含む 2 つの集合を返す。これらの 2 つの集合にさらにペアワイズ・グルー演算を適用することにより、5 つの区間が返される。この結果は正しく $X \otimes Y$ の結果となっている (図 5 を参照)。

4.3.1 問い合わせの結果

我々のキーワード集合によるビデオ問い合わせの結果を計算するためには定理 1 を適用する。 $Q = (k_1)^3 \wedge (k_2)^3$ を与えられた問い合わせとすると、その Q に対する結果 $[Q]$ を次のように定義する。

$$[Q] = [(k_1)^3 \wedge (k_2)^3] = [[k_1] \otimes [k_2]],$$

$$\text{但し } [k] = \{i \mid \exists (k, i) \in \mathcal{VD}\}$$

さらに、定理 1 により

$$[Q] = ([k_1] \oplus [k_1]) \oplus ([k_2] \oplus [k_2])$$

となる。

5. 区間フィルタ

前章で紹介したペアワイズ・グルーやパワーセット・グルーのようなグルー演算は、ある問い合わせに対して多数の不適切な区間を生成する可能性が非常に高いということは、2 章に挙げた例から明らかである。以下の節では、そのような不適切な区間を答えの集合から取り除くための合理的なフィルタについて述べる。

また、これらのフィルタを上で提案したグルー演算と
うまく統合化できるかどうかを考察する。

5.1 タイム・ウインドウ・フィルタ

Video on Demand 以外の多くのビデオ検索アプリ
ケーションでは、限られた幅の区間が求められるのが
一般的である。それは、指定された幅より長い区間を
検索結果から取り除かなければならないということだ
る。我々はこのようなフィルタとしてタイム・ウイン
ドウ・フィルタを定義する。タイム・ウインドウ・
フィルタは検索結果から幅が長すぎる不必要な区間を
排除する。

$|i|$ を区間 i の幅とする、すなわち $|i| = f_c(i) - f_s(i)$ 。タイム・ウインドウ・フィルタ F_w とは、任意
の区間の集合 X から X の部分集合への関数である。
まず、区間 i に対して部分関数 F_w を次のように定義
する。

$$F_w(i) = \begin{cases} i, & \text{if } |i| \leq w; \\ \text{undefined,} & \text{otherwise} \end{cases}$$

ここで、 w は指定されたタイム・ウインドウである。
また区間の集合 X に対して、 $F_w(X)$ を以下のように
定義する。

$$F_w(X) = \{i \mid i \in X \text{ and } |i| \leq w\}.$$

5.1.1 Glue 演算とタイム・ウインドウ・フィルタ

以前に定理 1 で述べたグルー演算の性質が、もしそ
れにタイム・ウインドウ・フィルタをかけた場合でも、
成立するならば、我々の問い合わせ機構とタイム・ウ
インドウ・フィルタを結合することを可能になる。つま
り、区間の任意の集合 X と Y に対して、以下のよう
なもの成り立つことになる。

$$F_w(X \otimes Y) = F_w(F_w(X \oplus X) \oplus F_w(Y \oplus Y))$$

ここで、 F_w は指定されたタイム・ウインドウ・フィルタ
である。もし上記の式が成立すれば問い合わせ処理の
効率化が図れる可能性がある。上式の左辺は $X \otimes Y$
を求めてからその結果に対して F_w をしているが、右
辺では、 \otimes 演算を実行する前に、その引数に対して
タイム・ウインドウ・フィルタを適用することによっ
て、予め候補区間の数を減らすことが可能である。

以下の補題では、タイム・ウインドウ・フィルタと組
み合わせたグルー演算に関する基本的な性質を示す。
これらの性質は、定理 2 を証明する際に用いる。

補題 1 任意の区間 x と y の区間グルー演算は以下
の性質をもつ。

$$F_w(x \oplus y) = F_w(F_w(x) \oplus F_w(y)),$$

ここで F_w はタイム・ウインドウ・フィルタである。

証明: $|x| \leq w$ かつ $|y| \leq w$ の場合は、このレ
ンマは明らかである。 $|x| > w$ 、あるいは $|y| > w$
の場合は、 $F_w(F_w(x) \oplus F_w(y)) = \text{undefined}$ である。
 $|x \oplus y| \geq |x|$ 、 $|y|$ なので $|x| > w$ 、ある
いは $|y| > w$ の場合、 $|x \oplus y| > w$ が明らかで
ある。 $|x \oplus y| > w$ の場合は $F_w(x \oplus y) = \text{unde}$
 fined である。よって $F_w(x \oplus y) = F_w(F_w(x) \oplus F_w(y))$ 、
が成り立つ。□

補題 2 区間の任意の集合 X と Y のペアワイズ・
グルーは以下の性質をもつ。

$$F_w(X \oplus Y) = F_w(F_w(X) \oplus F_w(Y)),$$

但し F_w はタイム・ウインドウ・フィルタである。

証明: $U = F_w(X \oplus Y)$ 、 $V = F_w(F_w(X) \oplus F_w(Y))$
とする。 $U = V$ を証明するために $U \supseteq V$ かつ $U \subseteq V$
を証明する。

任意の $x \in F_w(X)$ を考える。当然、 $x \in X$ である。
このため $X \supseteq F_w(X)$ 。同様に、 $Y \supseteq F_w(Y)$ 。このた
め $X \oplus Y \supseteq F_w(X) \oplus F_w(Y)$ 。また、 $F_w(X \oplus Y) \supseteq$
 $F_w(F_w(X) \oplus F_w(Y))$ 。従って、 $U \supseteq V$ 。

次に、任意の要素 $z \in U$ を考える。 $|z| \leq w$
とすると、 $x \oplus y = z$ であるような $x \in X$ かつ
 $y \in Y$ が存在するはずである。 $|z| \leq w$ ならば
 $|x| \leq w$ なので、 $x \in F_w(X)$ と言える。同様に、
 $y \in F_w(Y)$ とも言える。また、 $|x \oplus y| \leq w$ な
ので $(x \oplus y) \in F_w(F_w(X) \oplus F_w(Y))$ である。この結
果、 $z \in F_w(F_w(X) \oplus F_w(Y))$ である。それゆえに、
 $U \subseteq V$ 。よって題意が証明された。□

これらの補題から次に、定理 1 で述べたパワーセッ
ト・グルー演算の変形は、タイム・ウインドウ・フィ
ルタと組み合わせる時でも成り立つことを証明する。

定理 2 X と Y を区間の任意の集合とすると以下
の式が成り立つ。

$$F_w(X \otimes Y) = F_w(F_w(X \oplus X) \oplus F_w(Y \oplus Y))$$

但し、 F_w はタイム・ウインドウ・フィルタである。

証明: 定理 1 によって、

$$X \otimes Y = (X \oplus X) \oplus (Y \oplus Y).$$

この両辺にタイム・ウインドウ・フィルタ F_w をかけ
ると $F_w(X \otimes Y) = F_w((X \oplus X) \oplus (Y \oplus Y))$ になる。
 $X \oplus X = X'$ 、 $Y \oplus Y = Y'$ とすると補題 2 によ
り、 $F_w(X' \oplus Y') = F_w(F_w(X') \oplus F_w(Y'))$ 。置き換え
により、

$$F_w((X \oplus X) \oplus (Y \oplus Y)) =$$

$$F_w(F_w(X \oplus X) \oplus F_w(X \oplus X)).$$

従って、

$$F_w(X \otimes Y) = F_w(F_w(X \oplus X) \oplus F_w(Y \oplus Y)). \square$$

定理 2 は解となる区間を合成するとともに検索結果から不適切な区間を取り除くことができる。補題 2 で示した $F_w(X \oplus X) = F_w(F_w(X) \oplus F_w(X))$ による性質は、 $F_w(X)$ と $F_w(Y)$ のようなフィルタを適用することによって候補区間の数を問い合わせ処理の初期でも減らせる大きい利点がある。従って、タイム・ウィンドウ・フィルタはビデオ問い合わせと統合することによって効率的な問い合わせメカニズムが可能となる。

5.2 最長ノイズ・フィルタ

ある問い合わせに対して『ノイズ』とは、その区間中に、その問い合わせ中のどのキーワードも全く該当しないような区間である。2章で議論したように、解として適切なある区間は短い期間のノイズが含まれることがあり得るが、長い期間のノイズが含まれている区間はある問い合わせに対して不適切と考えられることが多い。その結果、そのような不適切な区間を検索結果から取り除かなければならないと考えられる。

$Q = (k_1)^{\exists} \wedge (k_2)^{\exists} \wedge \dots \wedge (k_n)^{\exists}$ と区間 i に対して、以下が成り立つ時 \bar{i} を答え区間 i の中にある『ノイズ』と呼ぶ：

$\bar{i} \subset i$, かつ、問い合わせ中のどのキーワード k_i も区間 \bar{i} では該当しない。

答え区間の中にはそのような区間が多数含まれている可能性があるため、その中の最も長い区間 $noise_i$ だけを考慮する。最長ノイズ・フィルタとは、検索結果からある幅 N より長い『ノイズ』が含まれている全ての区間を排除するものである。

$|noise_i|$ はある区間 i の中の最長のノイズの幅を示す。最長ノイズ・フィルタ F_N とは、任意の区間の集合 X から X の部分集合への関数である。まず、区間 i に対して部分関数 F_N を次のように定義する。

$$F_N(i) = \begin{cases} i, & \text{if } |noise_i| \leq N; \\ \text{undefined,} & \text{otherwise} \end{cases}$$

ここで、 N は指定された最大限のノイズの幅である。また区間の集合 X に対して、 $F_N(X)$ を以下のように定義する。

$$F_N(X) = \{i \mid i \in X \text{ and } |noise_i| \leq N\}$$

5.2.1 Glue 演算と最長ノイズ・フィルタ

残念ながら、定理 1 に示したパワーセット・グルー

演算の変形は最長ノイズ・フィルタと組み合わせると、ある条件を満たさない限り、成り立たない。すなわち、 X と Y を区間の任意の集合とすると、 $F_N(X \otimes Y) \neq F_N(F_N(X \oplus X) \oplus F_N(Y \oplus Y))$ である。ここで、 F_N は最長ノイズ・フィルタである。

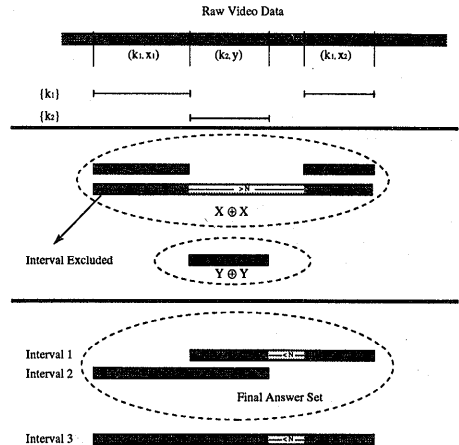


図 8 ノイズ除去の問題

Fig. 8 Relevant Interval being Excluded

例えば、図 8 に示したように、区間 (k_1, x_1) と (k_1, x_2) が N より離れているとすると、ある問い合わせ $Q = (k_1)^{\exists} \wedge (k_2)^{\exists}$ に最長ノイズ・フィルタ F_N を \oplus を計算する前に適用する場合には、Interval 3 が、指定された幅 N より長いノイズが含まれてない事実にもかかわらず、それが検索結果から取り除かれる。なぜなら、問い合わせ処理の中間の $F_N(X \oplus X)$ 演算は $(x_1 \oplus x_2)$ によって作り出した区間を排除し、検索の最終結果の中にはその区間を使用して作り出せる区間が含まれなくなってしまうからである。従って、その検索の最終結果としては Interval 1 と Interval 2 だけが返される。任意の 2 つの区間はそれぞれ最長ノイズ・フィルタをかける時に条件を満たさなかったとしても、それらの区間グルー演算が生成した区間はその条件を満たす可能性があることがこの問題の原因である。我々はこの問題を『ノイズ除去の問題』と呼んでいる。

ある問い合わせ $Q = (k_1)^{\exists} \wedge (k_2)^{\exists}$ に最長ノイズ F_N をかけると、

$F_N(F_N(X \oplus X) \oplus F_N(Y \oplus Y))$ (但し X と Y はそれぞれ k_1 と k_2 に該当する区間の集合である) は、以下の場合に正しい検索結果を返すことができない。

$$\exists x_1 \in X. \exists x_2 \in X. \exists y \in Y \text{ かつ } |x_1 \oplus y| \leq N \text{ かつ}$$

$$\begin{aligned} |y \oplus_{\text{noise}} x_2| &\leq N \text{ かつ} \\ |x_1 \oplus_{\text{noise}} x_2| &> N. \end{aligned}$$

$F_N(F_N(X \oplus Y) \oplus F_N(X \oplus Y))$ とすれば、この問題が解決されるかと思われるが、以下の場合にはそれも不可能である。

$$\begin{aligned} \exists x_1 \in X, \exists x_2 \in X, \exists y \in Y \text{ かつ} \\ |y \oplus_{\text{noise}} x_1| &\leq N \text{ かつ} \\ |x_1 \oplus_{\text{noise}} x_2| &\leq N \text{ かつ} \\ |y \oplus_{\text{noise}} x_2| &> N. \end{aligned}$$

しかしながら、以下の条件が満たされれば、今述べた2つのパワーセット・グルー演算の和は正しい検索結果となる。すなわち、すべての $x \in X$ 、すべての $y \in Y$ に対し、 $x \not\subseteq y$ かつ $y \not\subseteq x$ ならば、

$$\begin{aligned} F_N(X \otimes Y) = \\ F_N(F_N(X \oplus X) \oplus F_N(Y \oplus Y)) \cup \\ F_N(F_N(X \oplus Y) \oplus F_N(X \oplus Y)) \end{aligned}$$

この条件さえ満たされれば、上の2つのパワーセット・グルー演算によって、その各演算で作れなかった適切な区間を埋め合わせることができる。従って、ノイズ除去の問題は、幾つかのパワーセット・グルー演算だけによってある程度解決できる。ここで注意すべきことは、この新しく変換された演算はもとの演算より複雑に見えるが、計算の初期段階で多くの解候補を取り除くことができる場合には、元々の定義、すなわち $F_N(X \otimes Y)$ よりは効率的なことである。その結果、最長ノイズ・フィルタを問い合わせ機構に結合することによってより良いパフォーマンスが予想できる。

最長ノイズ・フィルタは、タイム・ウインドウ・フィルタと比べて計算量が多いが、ビデオの検索結果から不適切な区間を減らす最も自然な方法である。実際的なアプリケーションでは、最大限のノイズの幅「 N 」には高い値を割り当てることができる。なぜなら、我々の目的はノイズがかなりの部分が含まれているの区間だけを取り除くからである。さらに、その最大限のノイズの幅に割り当てられた値が非常に高いならば、ノイズ除去の問題が起こることはめったにないと思われる。

6. 関連研究

現在、ビデオ・データの検索に多くの研究者が深い興味を持っている。アレンの時区間論理¹⁾に基づいたビデオ関係の研究が幾つかある²⁾⁷⁾⁶⁾。Littleらはビデオのような時刻に依存するマルチメディアのデー

タのための時区間論理に基づくモデルを提示した⁶⁾。我々の研究は時区間論理に基づいた研究とは独立である。我々は、むしろキーワードによるビデオ問い合わせのセマンティックスやそのセマンティックスと適合する答えの区間生成するための演算をどう定義すればいいのかに注目した。以前に挙げられた例のように意味のあるどんなビデオ区間の中でも、同じキーワードは何回も、しかも不特定の順で出現することが多い。従って、それらの時区間論理に基づいた問い合わせを本研究で提案された問い合わせ機構と統合することは興味深い今後の課題であるが、より深い調査が必要である。

問い合わせモデルよりむしろ内容記述モデルに注目した研究も幾つかある⁹⁾⁵⁾⁴⁾¹²⁾。前述の通り、どんな内容記述モデルでも、検索結果になりえる全てのビデオ区間を予め識別するのは現在の技術で不可能である。既存の区間から新しい区間を計算するためにそれらが幾つかの区間演算を定義しているが、それらは我々の目的のためには不十分である。

Informedia プロジェクト¹¹⁾では、キーワードの集合によるビデオ問い合わせに対して情報検索システムの世界でよく知られている技術 *tf/idf* (term frequency/inverse document frequency) 法が適用されている。しかし、答えの区間は、動的に合成するでなく、「ビデオ・パラグラフ」と呼ばれているインデックスを付けられたユニットになる。

文献¹³⁾では、視覚的に類似したショットでも、もしそれらのショットは時間的にある距離より離れている場合に、別々のストーリー・ユニットの構成要素になる可能性があることが示されている。彼らはこのアイデアをストーリー・ユニットに基づいたビデオ・ショットをクラスタリングするために適用している。我々の最長ノイズ・フィルタの概念もこのアイデアに基づいている。

7. 議論

答えのオーバーローディングはどんな IR システムでも重要な問題である。テキストやイメージ・ドキュメントと比べて、ビデオ・ドキュメントは目を通すのにかなりの時間がかかってしまうので、その問題は、ビデオ検索システムにおいてより目立ってくるのが考えられる。本論文では2つの区間フィルタしか紹介しなかったが、「合計ノイズ」や「ノイズ数」のような他にも様々なフィルタが考えられる。そのようなフィルタは、どんな条件を満たしたら、定理2のように変形でき、問い合わせの正しい結果を得られるのかを調

べる必要がある。それによってビデオ問い合わせに適するフィルタ関数の定義は一般化できるであろう。

7.1 区間包含の問題と区間取り切り

5章ではノイズ消去の問題について述べた。ここでは「区間包含の問題」と呼ぶ問題について述べる。ある問い合わせに対して、1つのマッチした区間はもう1つのマッチした区間に包含される場合は、それらの区間グルー演算で作出す区間には包含される区間の最初のフレームと最終のフレームの情報なくなってしまう。その結果、もしもその包含される区間とまた別の区間の演算で作出せるような答えの区間があったとしても、その区間を計算できない可能性がある。

ある問い合わせに対してマッチする2つの区間の間に、もしそのような包含関係があるならば、包含される区間の境界を包含する区間のどちらかの境界まで広げることができる。なぜなら、少なくとも1つのマッチしたキーワードがある区間部分の中に出現すれば、その区間部分はノイズが含まれるはずがないからである。区間包含の問題を解決するために「ノイズ埋め合わせのための演算」と呼ぶ以下の演算を定義する。

x と y を区間とする。 x と y の区間取り切りを次のように定義する (図9を参照)。

$$x \circledast y = \begin{cases} \{x_1, x_2\} & \text{if } x \subset y \\ x & \text{otherwise} \end{cases}$$

ここで、 $x_1 \equiv x_1(f_s(x), f_e(y))$, $x_2 \equiv x_2(f_s(y), f_e(x))$ である。

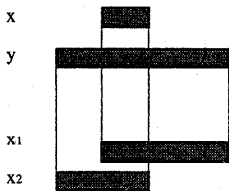


図9 X と Y の区間取り切り

Fig. 9 Splitting an Interval for Noise Compensation

X と Y を区間の任意の集合とする。 X と Y の区間取り切りを以下のように定義する。

$$X \circledast Y = \{i \mid \exists x \in X. \exists y \in Y. i = x \circledast y\}$$

以下の式によって、定理1で述べたパワーセット・グルー演算の変形と最長ノイズ・フィルタを組み合わせた時に起る「ノイズ消去の問題」と「区間包含の問題」両方とも解決できる。

$$F_N(X \circledast Y) = F_N(F_N(X' \oplus X') \oplus F_N(Y' \oplus Y')) \cup$$

$$F_N(F_N(X' \oplus Y') \oplus F_N(X' \oplus Y'))$$

ここで $X' = X \circledast Y$, $Y' = Y \circledast X$ として、 X, Y は k_1, k_2 それぞれとマッチした区間の集合である。

8. 結 論

現在のどんなビデオ・データベース・システムでも、エンド・ユーザが本当に求めている区間を検索することが困難である。なぜなら、検索結果になり得るそのような全ての区間がそのデータベースの中で答えのユニットとして格納されていることがほとんどないからである。自動的なものであれ、手作業によるものであれ、ナレーションに関する音声認識による、あるいは手作業的な内容記述のインデックスは断片的なユニットにしか付けることができない。そのような場合に、答えの区間はそのデータベースにあるインデックス付けユニットから動的に合成する必要がある。

ビデオ区間の検索では union や intersection や concatenation のような代数演算が必ず答えの区間を計算できるとは限らない。なぜなら、これらの演算はビデオの意味のある区間の中に当然なことのように存在するノイズを考慮しないからである。本論文の主な貢献はキーワードの集合によるビデオ問い合わせのために提案したグルー演算である。これらの演算は、ある問い合わせに対して答えになりえる全ての区間を断片的なユニットから合成できる。

我々は、検索結果から不適切な答え区間を排除するために区間フィルタの概念も提案した。それから、全てのフィルタは提案されたパワーセット・グルー演算と直接に組み合わせることができるかどうかを考察した。その結果、タイム・ウインドウ・フィルタの場合にはうまく組み合わせることができるが、最長ノイズ・フィルタの場合はパワーセット・グルーの式を修正する必要があることがわかった。しかし、修正された式でも能率的に問い合わせの結果を得ることができるともわかった。

参 考 文 献

- 1) J.F. Allen. "Maintaining Knowledge about Temporal Intervals". *Communications of the ACM*, 26(11):832-843, 1983.
- 2) C. Djeraba and H. Briand. "Temporal and Interactive Relations in a Multimedia Database Systems". *Lecture Notes in Computer Science*, 1242:457-473, May 1997.
- 3) C. Faloutsos and K. Lin. "FastMap: A Fast Al-

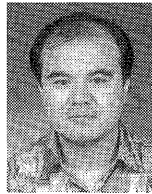
gorithm for Indexing and Data-Mining and Visualization of Traditional and Multimedia Datasets". *SIGMOD Record*, 24(2):163-174, 1995.

- 4) R. Hjelsvold, R. Midtstraum, and O. Sandst. "Searching and Browsing a Shared Video Database". *Multimedia Database Systems. Design and Implementation Strategies*. chapter 4. Kluwer Academic Publishers, 1996.
- 5) E.J. Hwang and V.S. Subrahmanian. "Querying Video Libraries". *Journal of Visual Communications and Image Representation*, 7(1):44-60, March 1996.
- 6) T.D.C. Little and A. Ghafoor. "Interval-Based Conceptual Models for Time-Dependent Multimedia Data". *IEEE Transactions on Knowledge and Data Engineering*, 5(4):551-563, August 1993.
- 7) N.A. Lorentzos and Y.G. Mitsopoulos. "SQL Extension for Intervals Data". *IEEE Transactions on Knowledge and Data Engineering*, 9(3):480-499, May/June 1997.
- 8) A.D. Narasimhalu, M.S. Kankanhalli, and J. Wu. "Benchmarking Multimedia Databases". *Multimedia Tools and Applications*, 4(3):333-356, May 1997.
- 9) E. Oomoto and K. Tanaka. "OVID: Design and Implementation of a Video-Object Database System". *IEEE Transactions on Knowledge and Data Engineering*, 5(4):629-643, August 1993.
- 10) T.G. Aguiere Smith and G. Davenport. "The Stratification System: A Design Environment for Random Access Video". In *Proc. 3rd Int'l Workshop on Network and Operating System Support for Digital Audio and Video*, pages 250-261, 1992.
- 11) H.D. Wactlar, T. Kanade, M.A. Smith, and S.M. Stevens. "Intelligent Access to Digital Video: Informedia Project". *IEEE Computer*, 29(5):46-52, May 1996.
- 12) R. Weiss, A. Duda, and D. Gifford. "Composition and Search with a Video Algebra". *IEEE MultiMedia*, 2(1):12-25, Spring 1995.
- 13) M. Yeung, B. Yeo, and B. Liu. "Extracting Story Units from Long Programs for Video Browsing and Navigation". In *International Conference on Multimedia Computing and Systems*, pages 296-305, June 1996.
- 14) H.J. Zhang, C.Y. Low, S.W. Smoliar, and J.H. Wu. "Video Parsing and Retrieval and Browsing: An Integrated and Content-based Solution". In *Multimedia 95 Proceedings*, pages 15-24, November 1995.

(平成 10 年 9 月 20 日受付)

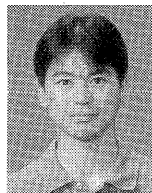
(平成 10 年 11 月 5 日採録)

(担当編集委員 片岡 良治)



ブラダン・スジット (正会員)

1988年インドのラザスタン大・工・機械工学卒, 1995年神戸大学大学院工学研究科修士修了。現在, 神戸大学大学院自然科学研究科博士後期課程在学中, 日本学術振興会特別研究員。ビデオ・データベースの研究開発に従事。プロトタイプベース言語, マルチメディア・オーサリング, 半構造データベースに関する研究に興味を持つ。情報処理学会員。



田島 敬史 (正会員)

1991年東京大学理学部情報科学科卒業。1993年東京大学大学院理学系研究科情報科学専攻修士課程修了。1994年より1996年まで京都大学数理解析研究所研究生。1996年東京大学大学院理学系研究科情報科学専攻博士課程修了。理学博士。同年より神戸大学工学部情報知能工学科助手。主にデータベースシステムの研究に従事。情報処理学会, ソフトウェア科学会, 各会員。



田中 克己 (正会員)

1974年京大・工・情報工学卒, 1976年同大学大学院修士修了。1979年神戸大学教養部助手, 1986年同大学工学部助教授。1994年同大学工学部教授(情報知能工学専攻)。1995年同大学大学院自然科学研究科(現在, 情報メディア科学専攻)専任教授, 現在に至る。工博。主にデータベースの研究に従事。現在, 情報処理学会データベースシステム研究会会主査。96年度より通信・放送機構「次世代デジタル映像通信の研究開発」の研究統括責任者, 文部省科研費重点領域研究「分散発展型データベースシステム技術の研究」の研究代表者, 神戸マルチメディアインターネット協議会会長, 情報処理学会, 人工知能学会, IEEE Computer Society, ACM等各会員。