

コンテンツに基づいたビデオ検索言語「ユニット代数」の提案

榎 美紀[†] 増永 良文[‡]

[†]お茶の水女子大学大学院 人間文化研究科博士前期課程 数理・情報科学専攻

[‡]お茶の水女子大学 理学部情報科学科

〒112-8610 東京都文京区大塚 2-1-1

E-mail: [†] miki@db.is.ocha.ac.jp, [‡] masunaga@is.ocha.ac.jp

あらまし 我々は、GPS とジャイロが装着されたビデオカメラを用いて街並みを撮影して、そこから得られた撮影者の位置や姿勢に関するデータ、および3次元地図データを使うことにより、ビデオのどのフレームからどのフレームまでどのような建物が写っていたかを自動抽出し、ビデオに建物名をリアルタイムで自動索引付けするシステムを開発してきた。ここでは、ビデオに写し込まれているコンテンツとしての被写体建物オブジェクトの連続したフレーム列をユニット(unit)と名づけているが、本稿では、ユニットに基づいたビデオ検索言語の理論的体系として「ユニット代数 (unit algebra)」を提案する。これはユニットが時区間であるという特徴に着目する一方、その体系化をリレーショナル代数とのアナロジーで明らかにしようとするものである。ユニット代数が確立すると、それに基づいたさまざまなビデオ検索言語の設計が可能となる。

キーワード 時区間論理, ビデオデータベース, ユニット代数, 検索言語

Proposal of "Unit Algebra": A Video Query Language based on its Contents

Miki ENOKI[†] and Yoshifumi MASUNAGA[‡]

[†] Graduate Division of Mathematics and Computer Science (Master's Program), Ochanomizu University

[‡] Department of Information Science, Faculty of Science, Ochanomizu University

2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 Japan

E-mail: [†] miki@db.is.ocha.ac.jp, [‡] masunaga@is.ocha.ac.jp

Abstract We have been conducting a project for building a video database system where video contents are indexed automatically and in real-time. That is, a video shooter walks on a street being equipped with a GPS and a Gyro sensor so that the sequence of video frames is collected along with the shooter's position and the camera's posture data. These data are processed using a "3-dimensional" city map so that the name of the buildings captured in each frame of a video clip is created as its index in real-time. In this paper, we propose "Unit algebra" as a theoretical framework of a video query language, where units represent frame sequences in which certain set of building objects are captured consecutively. This tries to clarify the systematization by the analogy with relational algebra while pay attention to the feature that the unit is a time interval. Based on the unit algebra, it becomes possible to design a variety of video query languages.

Keyword Time Interval Logic, Video Database, Unit Algebra, Query Language

1. はじめに

近年のビデオ関連技術の著しい進歩により、誰もがいつでもどこでもビデオ撮影ができる状況が出来上がっている。ビデオカメラの撮影記録媒体自体にもハードディスクが直接使用される時代となっており、ビデオのデータベースを構築することは容易になっている。

しかし、蓄積された膨大なビデオデータの中から、

自分の望むビデオ（あるいはシーン）を見つけ出すことは一般に容易ではない。その理由は、コンテンツ（＝内容）に基づいた索引（index）がビデオに付与されていないからである。

これまでに、Weiss ら[1]は、ニュースビデオに付随する全てのクローズドキャプションのテキストを抽出して、それらをビデオセグメントへのテキスト属性として付与し、セグメント間の時間的關係に着目してビ

ビデオ検索を行うビデオ代数を提案した。しかしここでは索引付けの対象となったビデオには字幕がついていなければならない。また、映像の中身を直接解析してはいない。

何ら[2]は、360度の全方位映像から、被写体を追跡して映像のメタデータを自動生成し、それを用いて特定の被写体を検索する手法を提案しているが、被写体を追跡するためには、あらかじめ被写体が写っていない元の背景映像を撮影しておき、その映像と比較した差分情報を基に被写体を追跡する必要がある。

そこで我々は、GPSとジャイロをビデオカメラに装着し、3次元地図を使うことにより、撮影されたビデオのどのフレームからどのフレームまでどのような建物がコンテンツとして写っていたかを自動抽出し、ビデオに建物名をリアルタイムで自動索引付けするシステムを開発してきた[3,4,5]。撮影されたビデオは、検索・再生に最適なサイズに分割され、ビデオデータベースに格納される[6]。

しかし、そのように自動索引付けされたビデオデータをどのように検索するかの間合せ体系については、今後の課題として残されてきた。

そこで本研究では、自動抽出された、ビデオに写しこまれている被写体オブジェクトの連続したフレーム列をユニット(unit)と定義し、これは時区間であるという特徴に着目し、ビデオ検索言語の理論的体系となる「ユニット代数 (unit algebra)」を提案する。これにより、ユニットを用いて任意の数の被写体建物オブジェクトを自由に検索し、その所望の部分のみを再生することが可能となる。

2. ビデオデータベースシステムの概要

本研究の基盤となっているビデオデータベースシステムの概要を述べる。ビデオカメラの撮影者はビデオ撮影者の位置と時刻を取得するためにGPSを身につけ、ビデオカメラに撮影者の姿勢を知るためにジャイロセンサを取り付けて撮影する。また、ウェアラブルコンピュータを用いてこれらのデータとそこに格納されている3次元地図データを総合的に処理しビデオデータに被写体建物オブジェクトの自動索引付けをリアルタイムで行い、それを利用したビデオ検索を実現する。図1は我々が開発しているシステムの全体像を表している。

実時間・全自動索引付けモジュールでは、市街地でビデオ撮影している撮影者のGPSデータ、ジャイロデータを取得し、3次元地図と(建物名称を取得するための2次元地図)を用いてリアルタイムに被写体建物オブジェクト抽出と索引付けを行う。それにより抽出された被写体建物オブジェクトの情報は、ビデオデータベース内へ格納される。

ビデオカメラで撮影されたビデオデータは、最適サイズに分割された後、ビデオデータベースに格納され検索に供される。

間合せ処理系では、ユニット代数あるいはそれを基に開発されるSQL風のビデオ検索言語を用いることによりユーザの自由なビデオ間合せを可能にする。

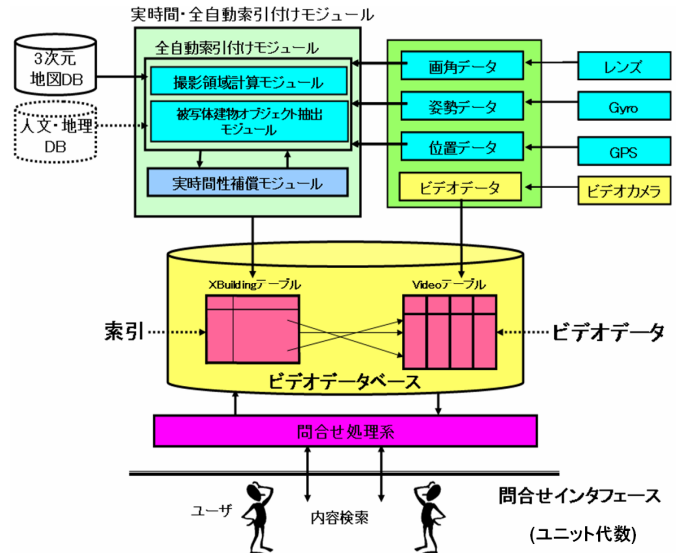


図1 3次元地図を用いた被写体建物オブジェクトの
実時間・自動索引付けビデオデータベースシステムの概念

3. ビデオ・セグメント・ユニット

ビデオデータベースの検索では、ビデオのコンテンツである被写体建物オブジェクトを対象として、例えば「銀座三越と松屋銀座が共に写っているビデオが欲しい」といった検索が行えるようにしたい。

そこで、ビデオ間合せ言語の理論的体系となる、ユニット代数を提案するが、その基となる「ユニット」を導入する。

3.1. ユニットの導入

【定義1】ビデオクリップ v

あるビデオクリップ v が与えられたとし、ビデオはフレーム列の集合であるから、 v の全フレーム数が m であるとする、 $v = [f_1, f_2, \dots, f_m]$ と表現される。 m を v の長さといい、 $|v|$ で表す。また、各フレーム f_i ($0 \leq i \leq m$) に写っている被写体建物オブジェクトの集合を、 $Obj(f_i) = \{o_{i1}, o_{i2}, \dots, o_{in_i}\}$ (n_i は f_i に写っている建物オブジェクトの総数) とすると、ビデオクリップ v 全体に写っている被写体建物オブジェクトの集合は、

$$Obj(v) = \bigcup_{i=1}^m Obj(f_i) = \{o_1, o_2, \dots, o_n\}$$

と表現される。以降、ビデオクリップを単にビデオと呼ぶこととする。

〔定義 2〕 ビデオセグメント s

ビデオ v のビデオセグメント s は、一般に $v = [f_1, f_2, \dots, f_m]$ の連続した部分フレーム列であり、 $s = [f_j, f_{j+1}, \dots, f_{j+l}]$ ($1 \leq j \leq n, l \leq n-j$) と表現される。 l を s の長さといひ、 $|s|$ で表す。以降、ビデオセグメントを単にセグメントと呼ぶこととする。

〔定義 3〕 ビデオユニット u

ビデオ $v = [f_1, f_2, \dots, f_m]$ のある被写体建物オブジェクトの集合 ($O \subseteq Obj(v), O = \{o_1, o_2, \dots, o_k\}$) に関するビデオユニット u とは、次の性質を満たすビデオ v のセグメント u をいう：

$$u = [f_j, f_{j+1}, \dots, f_{j+l}] \text{ such that } (\forall f_i)(j \leq i \leq j+l) ((O \subseteq Obj(f_i)) \wedge \neg(O \subseteq Obj(f_{j-1}))) \wedge \neg(O \subseteq Obj(f_{j+l}))$$

($j-1=0$ あるいは ($j+1=m+1$) のときは、 $\neg(O \subseteq Obj(f_{j-1}))$ と $\neg(O \subseteq Obj(f_{j+l}))$ は真とする。 \neg は

論理否定を表す。一般に、ビデオ v の O に関するユニットは複数存在しうる。また互いに素である。したがって、 v の先頭から数えて (開始フレーム順に) 第 i 番目のユニットを、 u_i^o と表現する ($i \geq 1$)。以降ビデオユニットを単にユニットと呼ぶことにする。

〔定義 4〕 基本ユニット

ビデオ v の被写体オブジェクト集合 O に関するユニットの中で、 O が単集合あるいは空集合のものを基本ユニットと言うこととする。

注意すべきことは、このユニットを構成している各フレームには少なくとも O の全ての元が写っていることが要求されているのであって、 O 以外の元が写っていてもかまわない。丁度 O だけが写っているユニットという概念も定義できるが、一般にフレームには注目した建物オブジェクト以外のオブジェクトが写し込まれていても良いとする立場から、上述の定義とした。

図 2 に被写体建物オブジェクトとユニットの関係を示す。例では、建物 o_1 が写っている連続したフレーム列の区間が二つ存在するので、ビデオ v の $\{o_1\}$ に関するユニットは、少なくとも $u_1^{o_1}, u_2^{o_1}$ という二つのユニットがある。

また、図 3 に見られるように、建物 o_1 と o_2 が同時に写っている区間は建物 o_1 と o_2 が同時にあるフレームから始まり、あるフレームまで連続して写し込まれている、第 i 番目の部分であると捉えることができるので、

これもまた、 u_i^O ($O = \{o_1, o_2\}$) と表現されるユニットであると考えられる。

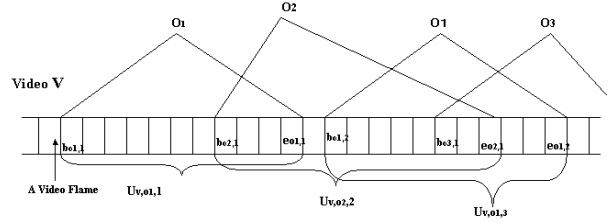


図 2 被写体建物オブジェクトとユニットの関係

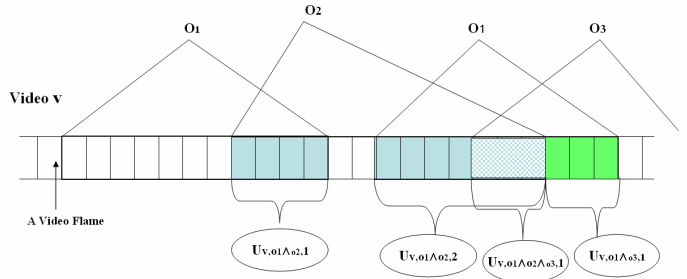


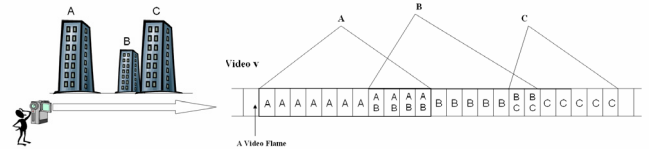
図 3 複数の被写体建物オブジェクトをもつユニット

3.2. コンテンツに基づいたビデオの索引表

ここで、ビデオ v のすべてのユニットを列挙することを試みる。 $Obj(v) = \{o_1, o_2, \dots, o_n\}$ とするとき、ビデオに写し込まれているかもしれない被写体建物オブジェクトの組み合わせは、表 1 の索引部に示したように、 $\{\phi, \{o_1\}, \{o_2\}, \dots, \{o_n\}, \{o_1, o_2\}, \dots, \{o_1, o_2, \dots, o_n\}\}$ の 2^n 通り存在することとなる (ϕ は空集合)。換言すれば、この組み合わせは、 v に写っているすべての被写体建物オブジェクト $Obj(v) = \{o_1, o_2, \dots, o_n\}$ の各々に対して、‘写っている’なら 1、‘何も指定しない’なら 0 とした 2 値で n 次元ベクトル表現をした、その組合せ総数 2^n と同意である。それにより、 ϕ は $(0, 0, \dots, 0)$ のベクトルに対応することになるが、これは、被写体建物オブジェクトすべてに対して、‘何も指定しない’という意味であり、このとき、ビデオ v の ϕ に関するユニットはビデオ v そのものとなる。一般的に、ビデオ v の $O = \{o_1, o_2, \dots, o_k\}$ に関するユニット集合 U^O の各ユニットは、少なくとも O の全ての元を被写体オブジェクトとして写し込んでいる。したがって、例えば、 $U^{\{o_1, o_2\}}$ は少なくとも o_1 と o_2 が共に移し込まれているユニットで、 $U^{\{o_1\}}$ と $U^{\{o_2\}}$ の積集合演算の結果を表すこととなる (詳細は 4.3 節)。

さて、 $Obj(v)$ の各部分集合に関してユニットを計算していくと、ビデオ v のコンテンツに基づいた索引表が作成されることになる。それを示したのが、表 1 である。なお、表 1 ではそれぞれのユニットに属する建物

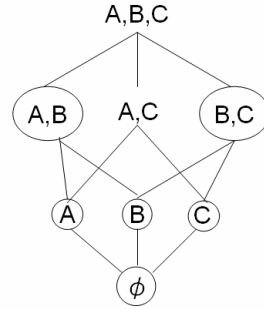
オブジェクトの集合が単一元からなるとき、すなわち単集合のとき、たとえば $\{o_1\}$ は単に o_1 と表現しており、以降の表記も同様にする。 $u_1^{o_1}, u_2^{o_1}, \dots, u_{n_{o_1}}^{o_1}$ は、開始時刻順に番号が振られているとする。



(a)ビデオを撮影する (b)撮影したビデオのフレーム表現

索引部	ユニット
ϕ	v
o_1	$\{u_1^{o_1}, u_2^{o_1}, \dots, u_{n_{o_1}}^{o_1}\} = U^{o_1}$
o_2	$\{u_1^{o_2}, u_2^{o_2}, \dots, u_{n_{o_2}}^{o_2}\} = U^{o_2}$
\vdots	
o_n	$\{u_1^{o_n}, u_2^{o_n}, \dots, u_{n_{o_n}}^{o_n}\} = U^{o_n}$
o_1, o_2	$\{u_1^{o_1, o_2}, u_2^{o_1, o_2}, \dots, u_{n_{o_1, o_2}}^{o_1, o_2}\} = U^{o_1, o_2}$
\vdots	
o_1, o_2, \dots, o_n	$\{u_1^{o_1, o_2, \dots, o_n}, u_2^{o_1, o_2, \dots, o_n}, \dots, u_{n_{o_1, o_2, \dots, o_n}}^{o_1, o_2, \dots, o_n}\} = U^{o_1, o_2, \dots, o_n}$

基本ユニット



(c) A,B,C の束. \circ はユニットが存在する頂点

表 1 コンテンツに基づいたビデオの索引表

さらに、注意しないといけないことは、ビデオ v が与えられたとき、一般に $Obj(v)$ の全ての部分集合に対してユニットが存在するわけではない。全ての部分集合は集合の包含関係で「束(lattice)」をなすが、ユニットの存在する頂点とそうでない頂点がある。

束表現されることにより、頂点 ϕ からはじまり、ある頂点 P でユニットが存在しないと、それ以降 P から構成されるユニットも全て存在しない。したがって、 P を含むユニットへの問合せは計算しなくても存在しないことが分かり、これにより、問合せ計算時間の短縮に役立てることができる。さらに、ユーザが問合せを行う際、頂点存在情報から同時に写しこまれている建物の候補をあらかじめ知り、問合せの見当をつけられることも考えられる。

ビデオ v のユニットの全集合を $Units(v)$ と表現する。

【例題 1】

被写体建物オブジェクト A,B,C を撮影した時の束。建物オブジェクト A,B と B,C は互いに重なる部分があるので、 $\{A,B\}, \{B,C\}$ のユニットは実際に存在するが、 $\{A,C\}$ のユニットは実際には存在しないことになる。

4. ユニット代数

4.1. ユニットの和、差、積演算子

ビデオ検索では、「 o_1 と o_2 が写っているビデオがほしい」といった「積」の問合せのみならず、「 o_1 または o_3 が写っているビデオがほしい」といった問合せや、「 o_1 は写っているが o_3 は写っていないビデオがほしい」といった、和や否定の要素を含んだ問合せ要求も存在すると考えられる。これらは、ユニットを定義した被写体オブジェクトの集合を被写体オブジェクトの「和」と「否定」を導入することにより定式化できる。まず、たとえば o_1 または o_3 が写っていることを表すための、和集合 $\{o_1+o_3\}$ を考える。

このときの被写体建物オブジェクト組合せ総数は、 $\{\{o_1+o_2\}, \{o_1+o_3\}, \dots, \{o_1+o_2+\dots+o_n\}\}$ となり、前述の 2^n 通りから基本ユニットの数を差し引いた、 $2^n - (n+1)$ 通りの組合せ数となる。

次に、 o_1 は写っているが、 o_3 は写っていないことを表す集合 $\{o_1, \neg o_3\}$ を考える。これは積の問合せ 2^n 通りに、さらに否定要素も含めた、 $\{\neg o_1\}, \{\neg o_2\}, \dots, \{\neg o_1, o_2\}, \dots$ といったユニットが追加されることになり、全体では、

$$\sum_{k=0}^n {}_n C_k 2^k = 3^n$$

となり、 3^n 通りの被写体オブジェクトの組合せが存在することになる。

この組合せ数は、先述の 2 値で表現した n 次元ベクトルからさらに、「写っていない」なら -1 、を追加した 3 値で n 次元ベクトルを表現した組合せ総数 3^n と同意である。

一方、和の問合せに否定要素を含めた組合せ総数は、 3^n 通りの被写体オブジェクトの組合せからユニット

の元が単集合であるユニット(基本ユニットと $\{\neg o_1\}, \{\neg o_2\}, \dots, \{\neg o_n\}$ のユニット)を差し引いて, $3^n - 2n + 1$ となる.

したがって, 和と否定を加えてユニットとなり得る被写体建物オブジェクトの全組合せ総数を計算すると, \langle 否定を含めた積の問合せ 3^n 通り $\rangle + \langle$ 否定を含めた和の問合せ $3^n - 2n + 1$ $\rangle = 2 \times 3^n - 2n + 1$ となる.

しかし, すべての被写体建物オブジェクトの組合せについてあらかじめ計算しユニットを予め計算しておこうとすると, 指数オーダの計算時間を必要とすることになり, とても実現的ではない.

そこで, 撮影と同時にリアルタイムでの計算, 索引付けが可能な基本ユニットを用いた演算により, 任意の建物オブジェクト集合に対するユニットを求めることを可能にするユニット代数を定義する.

4.2. 時区間としてのユニット

1本のビデオ v における全ユニット集合 $Units(v)$ について考え, 被写体建物オブジェクト集合 A のユニット $u^A = [f_p, f_{p+1}, \dots, f_{p+\alpha}]$ と, 被写体建物オブジェクト集合 B のユニット $u^B = [f_q, f_{q+1}, \dots, f_{q+\beta}]$ があるとする($A \neq B$).

まず, 検索の対象となるユニットは連続したフレーム列なので, それは Allen の時区間[7]であると捉えることができることに注意する. 2つの時区間の間に存在する時間的関連は, Allen の時区間論理が示しているように, 図6に示す13種である. よって, 2つのユニット同士の関係は, この13種のいずれかに該当することになる.

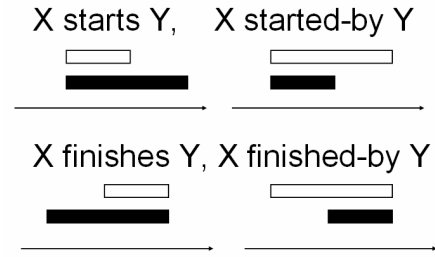
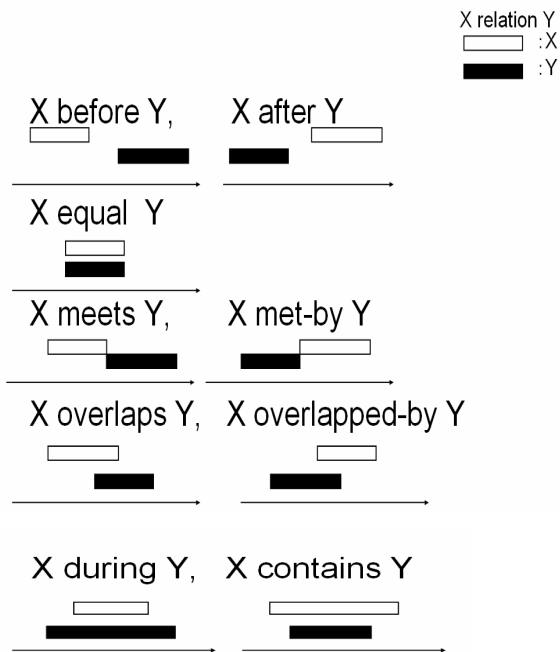


図6 Allen の時区間の13種の時間的関連

4.3. ユニット演算子の導入

そこで, ユニット間の演算として以下の2種を導入する. この演算の導入に関して留意している点は2つである:

- (1) A と B は被写体建物オブジェクトの集合なので, A と B の「和」, 「差」, 「共通」集合演算が考えられるので, それらに対応したユニット間の演算子を定義・導入する.
- (2) ユニット間の演算の結果得られるセグメントはまたユニットであるような演算定義とする. この制約はユニット演算子を再帰的に適用可能とするために重要である.

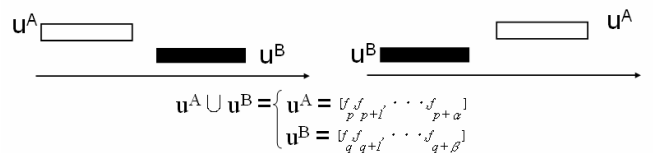
4.3.1. 和集合演算子

【定義5】 u^A と u^B をあるビデオ v の各々被写体オブジェクト集合 A と B に関するユニットとすると, その和, これを $u^A \cup u^B$ と書く, は次のように定義される.

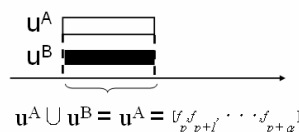
$$u^A \cup u^B = \{u \mid Obj(u) \supseteq A \vee Obj(u) \supseteq B\}$$

つまり, $u^A \cup u^B$ は建物オブジェクト集合 A または B のどちらかが写っている連続したフレーム列を抽出したユニットを表す. 前述の通り, 一般に u^A と u^B の間には時区間として13種の関連が存在するので, それを基に積集合演算子を具体的に計算すると, 以下のようになる.

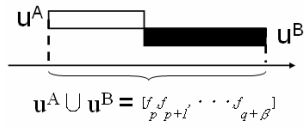
- ① u^A before u^B あるいは, ② u^A after u^B のとき



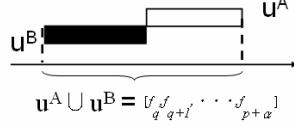
- ③ u^A equal u^B のとき



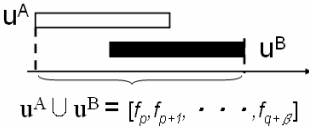
④ u^A meets u^B のとき



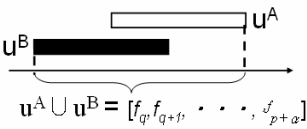
⑤ u^A met-by u^B のとき



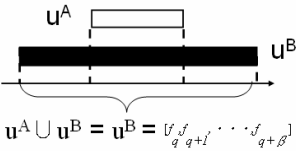
⑥ u^A overlaps u^B のとき



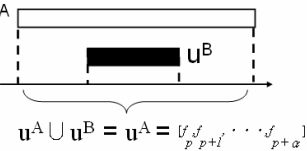
⑦ u^A overlapped-by u^B のとき



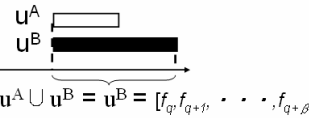
⑧ u^A during u^B のとき



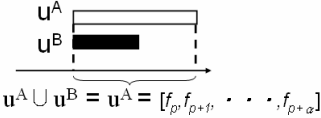
⑨ u^A contains u^B のとき



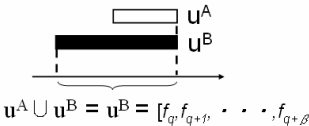
⑩ u^A starts u^B のとき



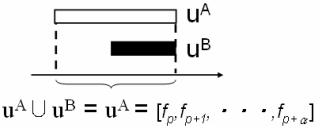
⑪ u^A started-by u^B のとき



⑫ u^A finishes u^B のとき



⑬ u^A finished-by u^B のとき



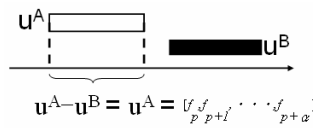
4.3.2. 差集合演算子

【定義 6】 u^A と u^B をあるビデオ v の各々被写体オブジェクト集合 A と B に関するユニットとすると、その積、これを $u^A - u^B$ と書く、は次のように定義される。

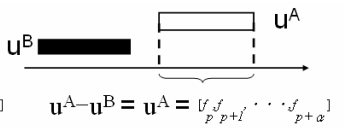
$$u^A - u^B = \{u \mid \text{Obj}(u) \supseteq A \wedge \neg(\text{Obj}(u) \supseteq B)\}$$

$u^A - u^B$ とは、建物オブジェクト集合 A の建物は写っているが、 B の建物は写っていない部分のユニットを表す。これは、各々の u^A に対し、 u^B と時区間が重なっている部分を除去することによって、得ることができる。したがって、否定要素の入ったユニット、 $\{u^A, \neg u^B\}$ 等は、この差集合演算を用いて、 $\{u^A, \neg u^B\} = u^A - u^B$ と捉えることができる。 u^A と u^B の間には前述のように一般に 13 種の関連が存在するので、差集合演算子は、具体的には以下のような。

① u^A before u^B のとき



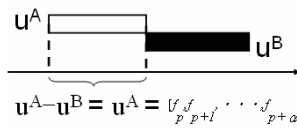
② u^A after u^B のとき



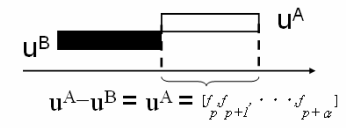
③ u^A equal u^B のとき

該当部分なし

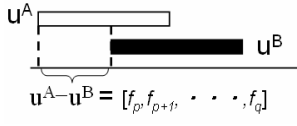
④ u^A meets u^B のとき



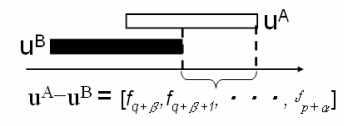
⑤ u^A met-by u^B のとき



⑥ u^A overlaps u^B のとき



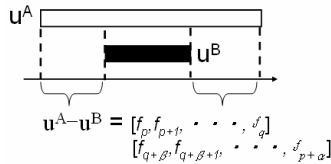
⑦ u^A overlapped-by u^B のとき



⑧ u^A during u^B のとき

該当部分なし

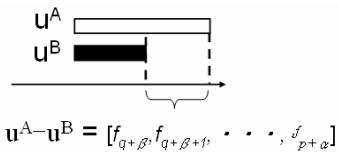
⑨ u^A contains u^B のとき



⑩ u^A starts u^B のとき

該当部分なし

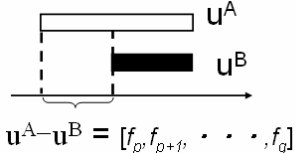
⑪ u^A started-by u^B のとき



⑫ u^A finishes u^B のとき

該当部分なし

⑬ u^A finished-by u^B のとき



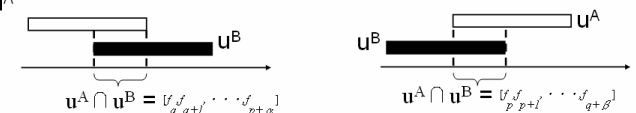
4.3.3. 積集合演算子

【定義 7】 u^A と u^B をあるビデオ v の各々被写体オブジェクト集合 A と B に関するユニットとすると、その積、これを $u^A \cap u^B$ と書く、は次のように定義される。

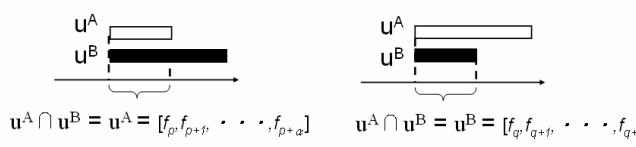
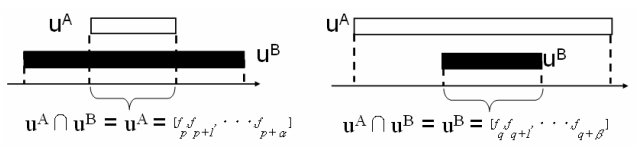
$$u^A \cap u^B = \{u \mid \text{Obj}(u) \supseteq A \wedge \text{Obj}(u) \supseteq B\}$$

つまり、 $u^A \cap u^B$ は u^A と u^B の時区間が重なる部分が存在するとき、その重なった部分の連続したフレーム列を抽出したユニットを表す。前述の通り、一般に u^A と u^B の間には時区間として 13 種の関連が存在するので、それを基に積集合演算子を具体的に計算すると、以下ようになる。

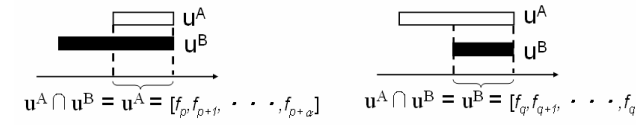
- ① u^A before u^B , あるいは② u^A after u^B のとき
該当部分なし
- ③ u^A equal u^B のとき
 $u^A \cap u^B = u^A = u^B = [f_p, f_{p+1}, \dots, f_{p+\alpha}]$
- ④ u^A meets u^B , あるいは⑤ u^A met-by u^B のとき
該当部分なし
- ⑥ u^A overlaps u^B のとき ⑦ u^A overlapped-by u^B のとき



- ⑧ u^A during u^B のとき ⑨ u^A contains u^B のとき
- ⑩ u^A starts u^B のとき ⑪ u^A started-by u^B のとき



- ⑫ u^A finishes u^B のとき ⑬ u^A finished-by u^B のとき



4.4. 問合せとしてのユニット演算子

問合せとしてユニット演算子を規定してみると次の 2 つの立場がある。ここに、 θ をユニット間の演算子とする：

- (1) ユニット u^A が与えられたとして、 u^A と θ の関係にある全てのユニット u^B を求める。
- (2) ビデオ v が与えられたとして、被写体オブジェクト集合 A と B を与えて、 $u^A \theta u^B$ の演算結果を全て求める。

前節で示した演算子の定義は、(1) の場合の計算法をそのまま表している。ここでは、(2) の結果をどのようにして求めるかを示す。

4.4.1. 和集合演算の場合

被写体建物オブジェクト集合 A と B を与えた場合、一般には、図 7 に示すように、複数のユニット $\{u^A\}$ と $\{u^B\}$ が存在する。

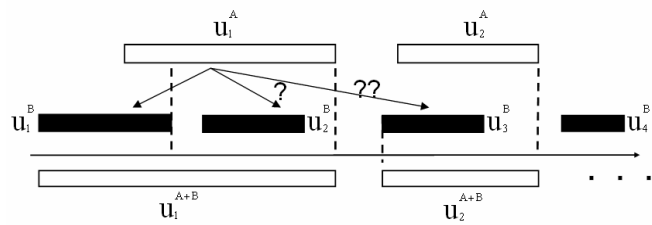


図 7 和集合演算の概念

そこで、まず $u_1^A \cup u_1^B$ を行くと、図 7 に示すように、 u_1^{A+B} のユニットを得ることができるが、次に、 u_1^A と u_2^B の演算を行うと、はじめに得られた u_1^{A+B} のユニットを無視して計算し、 u_1^{A+B} と時間が重複した演算結果を返してしまう。

この現象を回避するため、和集合演算ではまず $u_1^A \cup u_1^B$ から得た u^{A+B} のユニットを、 u_1^A として置き換え 次の u^B と演算を行うことを、 θ が after になるまで続け、最終的に残った u^A の集合を、 u^{A+B} の集合として置き換えることとする。

4.4.2. 差集合演算の場合

被写体建物オブジェクト集合 A と B を与えた場合、一般には、図 8 に示すように、複数のユニット $\{u^A\}$ と $\{u^B\}$ が存在する。

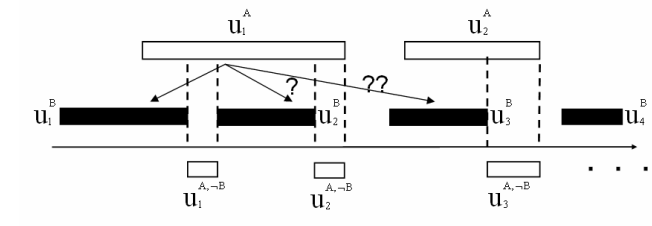


図 8 差集合演算の概念

まず $u_1^A - u_1^B$ を行くと、図 8 に示すように、 u_i^{A-B} のユニットを得ることができるが、次に u_1^A と u_2^B の演算を行うと、はじめに得られた u_i^{A-B} のユニットを無視して u_2^B は写っていないが u_1^B は写っている u_i^{A-B} のユニットを新たに作成することになってしまう。さらに $u_1^A - u_3^B$ にいたっては、before の関係になるために、 u_i^{A-B} の演算結果として u_1^A をそのまま残すことになる。この現象を回避するため、差集合演算ではまず $u_1^A - u_1^B$ から得た u_i^{A-B} のユニットを u_1^A として置き換え 次の u^B と演算を行い、最終的に残った u^A の集合を、 u_i^{A-B} の集合として置き換えることとする。この様子を図

9に示す.

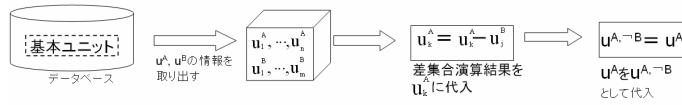


図9 差集合演算のながれ

4.4.3. 積集合演算の場合

被写体建物オブジェクト集合 A と B を与えた場合、一般には、図 10 に示すように、複数のユニット $\{u^A\}$ と $\{u^B\}$ が存在する。このとき積集合演算では、 u_1^A から、 $\{u^B\}$ の各ユニットに対し、 θ が after になるまで積集合演算を続け、 $\{u^{A,B}\}$ を得る。続いて u_2^A が、 $\{u^B\}$ の各ユニットに対し、同様の演算を行うことにより、 $\{u^{A,B}\}$ の全ユニットを過不足なく得ることが出来る。

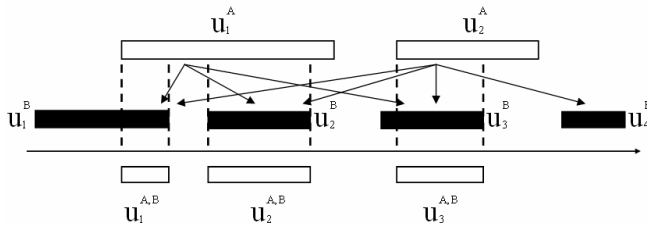


図10 積集合演算

4.5. ユニット代数表現の導入

【定義 8】 ユニット代数表現の定義

ビデオを v ，被写体建物オブジェクトの集合を O とする。

- (1) O を ϕ (空集合) とするとき、 $u^\phi (=v)$ は、ユニット代数表現である。
- (2) O を単集合 (singleton) $\{o_i\}$ とするとき、 u^{o_i} は、ユニット代数表現である。
- (3) u^O をユニット代数表現であるとし、 O と disjoint な単集合 $\{o'\}$ を考える。 ($O \cap \{o'\} = \phi$)
この時、 $u^O \cup u^{o'}$ 、 $u^O - u^{o'}$ 、 $u^O \cap u^{o'}$ もユニット代数表現である。
- (4) (1)~(3) で表現されるもののみがユニット代数表現である。

【例題 2】 ユニット代数表現の例

- (a) 銀座三越が写っている全てのユニットを求める
 $u^{\{\text{銀座三越}\}}$
- (b) 銀座三越か松屋銀座が写っている全てのユニットを求める
 $u^{\{\text{銀座三越} + \text{松屋銀座}\}}$
- (c) 銀座三越が写っていて、松屋銀座が写っていない全てのユニットを求める
 $u^{\{\text{銀座三越}, -\text{松屋銀座}\}}$

- (c) 銀座三越と松屋銀座が写っている全てのユニットを求める
 $u^{\{\text{銀座三越}, \text{松屋銀座}\}}$

5. まとめと今後の課題

本論文では、検索の対象としてユニットの概念を導入し、これは時区間であるという点に着目した映像問合せ表現の体系として和・差・積集合演算を持つユニット代数表現を定義した。さらに検索時にも時区間論理を用いることにより、「三越よりも前に撮られたユニット」といった時間的関連の問合せが可能になると考える。また、問合せで返ってくるものもユニットであるため、これらを複合オブジェクトとして返し再生を行えるような、値式の拡張を今後行っていく予定である。

これらを基にした、ビデオデータベース問合せ体系によってコマンドベースのみならず、使い易いグラフィカルなユーザインタフェースを設計・実装して、さまざまな問合せをサポートできる問合せ処理系を目指す。

文 献

- [1] R. Weiss, A. Duda, and D. Giord: "Content-based access to algebraic video." In IEEE Int'l Conf. on Multimedia Computing and Systems, Boston, USA, 1994.
- [2] 何 書勉, 田中 克己 "被写体追跡による全方位映像のメタデータ生成", 第 16 回データ工学ワークショップ(DEWS2005), 2005 年 3 月
- [3] 石黒玲, 佐藤有紀子, 増永良文 "3 次元地図を用いたビデオコンテンツの自動索引法—被写体建物オブジェクトの自動抽出—", 情報処理学会研究報告 (DBWS2003), Vol.2003, No.133-55, 2003 年 7 月.
- [4] 佐藤有紀子, 石黒玲, 増永良文 "3 次元地図を用いたデジタルビデオコンテンツの自動索引法の提案と検証", 日本データベース学会 (DBSJ Letters), Vol.3, No.1, pp.149-152, 2004 年 6 月.
- [5] Yukiko Sato and Yoshifumi Masunaga: "A Novel Indexing Method for Digital Video Contents using a 3-Dimensional City Map", Proceedings of the 4th International Workshop on Web and Wireless Geographical Information Systems (W2GIS), pp.333-343, Springer, November 2004.
- [6] 榎美紀, 増永良文 "3 次元地図を用いた自動索引付きビデオデータベースシステム—ビデオデータの格納と検索法—", 情報処理学会データベースシステム研究会報告 (DBWS2005), Vol.1. 2005,
- [7] J. Allen: "Maintaining Knowledge about Temporal Intervals", Communications of the ACM, Vol.26, No.11, pp.832-843, November 1983.