

映像内容に基づいたビデオ検索言語「ユニット論理」の体系化

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。本報告では、このシステムでビデオ検索を可能とするために、ある被写体オブジェクトが連続して写しこまれているフレーム列をユニット(unit)と定義し、これが建物情報を持った時区間であるという特徴に着目して、ビデオ検索言語の理論的体系となる「ユニット論理 (unit calculus)」を提案しそのフォーマルな体系化を行う。ユニット論理の確立によって、よりユーザフレンドリなビデオ検索言語の設計が可能になる。

キーワード ビデオ検索言語, マルチメディアデータベース, ユニット論理, 時区間論理

Systematization of "Unit Calculus": 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 Calculus" 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 calculus while pay attention to the feature that the unit is a time interval. Based on the unit calculus, it becomes possible to design a variety of video query languages.

Keyword Video Query Language, Multimedia Database, Unit Calculus, Time Interval Logic,

1. はじめに

近年、映像記録媒体のDVD・ハードディスク化、インターネットを介した大容量のビデオ共有・配信可能なマルチメディアサーバ利用の進歩により、誰もがいつでもどこでもビデオを大量に撮影し配信することが

可能となると共に、多くの人がビデオ検索を必要とするようになった。

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、ビデオの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。

索引付けは大別すると、人手によるものと、自動索引付けの2種類に分けられる。映像が少量である場合には、人手による索引付けも可能であるが、大量である場合には、人手が嵩むだけでなく、多くの時間を必要とし、かつエラーの起こりやすいものと指摘されており、自動索引付けが望まれることは言うまでもない。

ビデオは物理的な単位であるショット、いくつかの連続したショットからなる論理的な単位であるシーン、そして一般には複数のシーンからなる映像クリップとして成り立っている。

ビデオの自動索引付け手法としては、次のようなアプローチがとられてきた。

- (a) 画像認識技術によるアプローチ
- (b) 音声やテキスト認識技術によるアプローチ

(a)については、例えば Shipman[1]らが、画像を直接解析してショットの分割を行い、そのショットから特徴量を抽出する研究を行っているが、画像解析は膨大な計算量を必要とし、リアルタイムでの索引付けは難しい。

(b)については、例えば Weissら[2]が、ニュースビデオに付随する全ての字幕情報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、ニュース内容の階層関係を代数的に表現してビデオの再生操作を行うビデオ代数を提案した。しかしここでも索引付けの対象となるビデオには字幕がついていなければならない、ビデオの中身自体を直接解析してはいない。また、字幕情報だけでは足りない部分は人手を介して索引をつける必要がある。

そこで我々は、字幕情報がついていない、街並みなどの風景を写したビデオでも人手を介さない自動的な索引をリアルタイムで付け、自由なビデオ検索・再生を可能とするため、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]。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[4]。

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いたため、本研究では、自由なビデオ検索・再生を実現するために、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「ユニット論理 (unit calculus)」の体系化を行う。

これにより、ユニットの持つ属性情報(建物名情報・時間情報)を用いて表現し得るすべての問合せ文を網羅し、ユニット論理を基に作られたSQL風の間合せ言語を用いて検索された所望のシーンのみを再生することが可能となる。

2. システム構成

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。

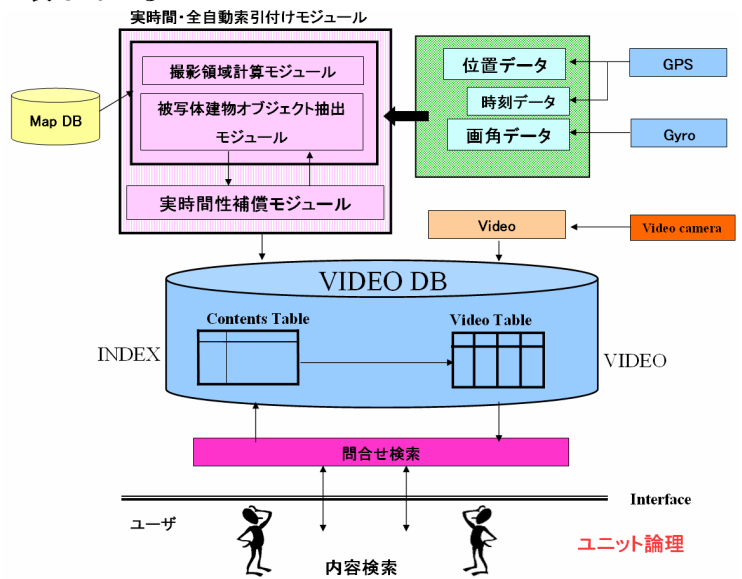


図1 3次元地図を用いた被写体建物オブジェクトの実時間・自動索引付けビデオデータベースシステムの概念

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。

問合せ処理系では、ユニット論理を基に開発される

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問合せを可能にする。

近年では、GPS機能とジャイロセンサが搭載された携帯電話端末も発売されており、ユーザが携帯電話を用いて撮ったビデオにも自動索引付けが可能になるなど、有用性のあるシステムであると考えられる。

3. 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」といった検索や、「SONYビルが写っていた後10分以内に三越が写っているビデオが欲しい」といった検索を行えるようにしたい。

そこで、ビデオ問合せ言語の理論的体系となるユニット論理を提案するが、まずはその基となる「ユニット」を導入する。

3.1. ユニットの導入

【定義1】ビデオ v

ビデオ v とは、(通常 1/30 秒間隔でとられた)連続したフレーム列であり、その全フレーム数が 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。

【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、 $s = [f_j, f_{j+1}, \dots, f_{j+l}]$ ($1 \leq j \leq m, l \leq m-j$) と表現される。 l を s の長さといい、 $|s|$ 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^O とは、次の性質を満たすビデオ v のセグメント $u = [f_j, f_{j+1}, \dots, f_{j+l}]$ をいう：

$$(\forall i)(j \leq i \leq j+l)(O \subseteq Obj(f_i)) \wedge \neg(O \subseteq Obj(f_{j-1})) \wedge \neg(O \subseteq Obj(f_{j+l+1}))$$

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

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。

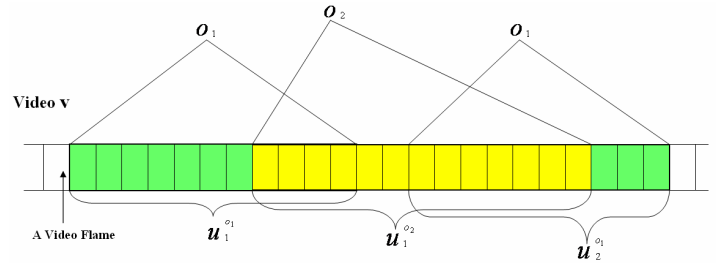


図2 被写体建物オブジェクトとユニットの関係

3.2. ユニット合成演算 —和・積・差—

建物 o_1 または o_2 が途切れることなく連続して写っている区間は建物 o_1 または o_2 が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こまれている第 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、ユニットである。そこで一般に被写体建物オブジェクトの集合を O_1 と O_2 とし、 O_1 と O_2 に関するユニットを $u_i^{O_1}, u_j^{O_2}$ としたとき、 O_1 の全ての元または O_2 の全ての元が途切れることなく連続して写っているユニット、これを $u_i^{O_1}$ と $u_j^{O_2}$ のユニットの和といい、これを $\text{union}(u_i^{O_1}, u_j^{O_2})$ と書く、が定義できる。

同様に、「 O_1 と O_2 が共に写っている」といったユニット「積」 $\text{intersection}(u_i^{O_1}, u_j^{O_2})$ や、「 O_1 は写っているが O_2 は写っていない」といったユニット「差」 $\text{difference}(u_i^{O_1}, u_j^{O_2})$ を定義できる[5]。

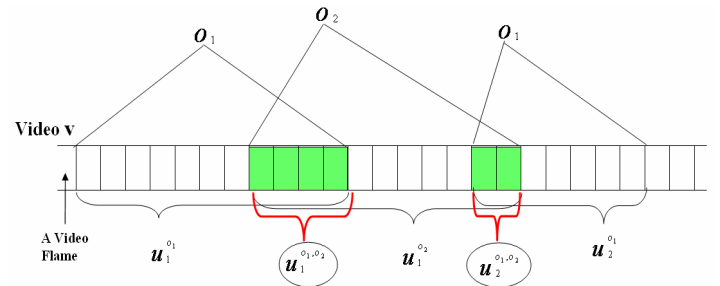


図3 積の被写体建物オブジェクトをもつユニット

図3は、積の被写体建物オブジェクトをもつユニ

ットを表している。

4. ユニット論理

4.1. Allen の時区間論理とその拡張

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[6]であると捉えることができる。よって、ビデオ検索にこの時区間論理を導入することで、ユニット同士の時間的関連を用いた検索が行えると考える。

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 4 に示す 13 種である。しかし、例えば”X before Y”であることと”Y after X”であることは同値なので、{before, after}, {meets, met-by}, {overlaps, overlapped-by}, {during, contains}, {starts, started-by}, {finishes, finished-by}の対については、それぞれ{before, meets, overlaps, during, starts, finishes}を本研究では検索に使用する。

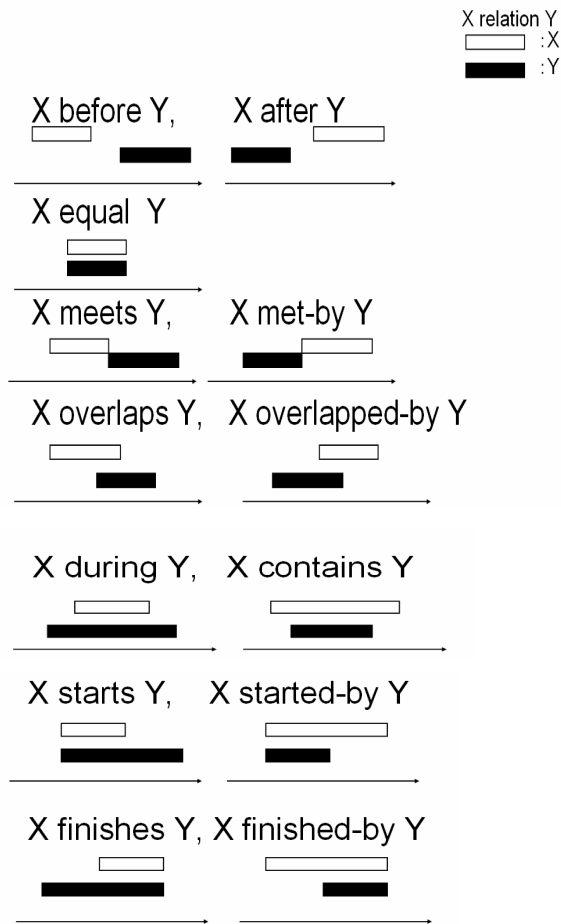


図 4 Allen の時区間の 13 種の時間的関連

Allen の時区間論理では、例えば時区間 X と Y が X before Y の関係にあるとしても、X が Y の何秒前に終了していたのか、ということは表現できない。しかしながら、ビデオ映像の表現においては、このような時間関連を直接表現できる関連性を定義してくべきであ

る。

そこで、我々は Allen の時区間論理を図 5 に示されるように拡張する。例えば、before(X, Y, =, d)は X が終了して、丁度 d 秒後に Y が生起する関連を表す。これは Stp(Start time point), Etp(End time point)を用いると次のように表現できる。

$$\text{before}(X, Y, =, d) \Leftrightarrow X.\text{etp} < Y.\text{stp} \text{ and } Y.\text{stp} - X.\text{etp} = d$$

さらに、beforeの関係の中で、「Xが開始してからn秒後にYが開始する」といった指定を行いたい場合は、before(X, Y, =, n - (X.etp - X.stp))と表現できる。before, overlaps, starts, finishesについては図 5 に示したそれぞれ一箇所をdとしておくことで、XのStp, Etp, YのStp, Etpを用いて表現されるすべての時区間指定が可能である。duringについては、図に見られる2箇所をd1, d2としておく必要がある。

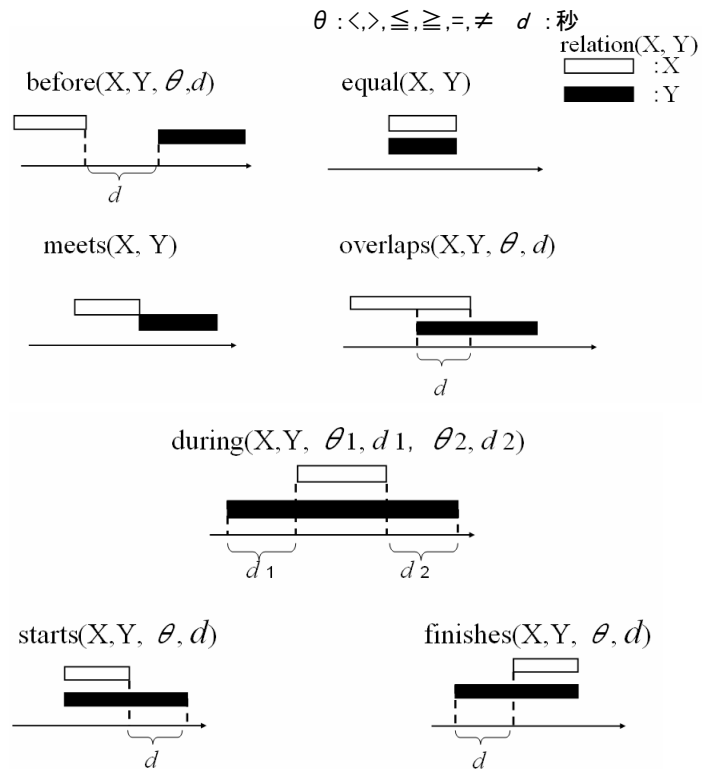


図 5 Allen の時区間論理の拡張

4.2. ユニット論理表現

4.2.1. ユニット論理表現の導入

本研究では、ユニットに対する和・積・差の演算と拡張した時区間論理を用いた検索を可能とする、ビデオをユニット単位で検索するための問合せ表現の体系として、ユニット論理(unit calculus)を提案する。

まずはユニットの持つ属性情報を用いて表現できる問合せを体系化するため、コンテンツリレーション

テーブルを作成する。

ユニットには *Uid* がその唯一識別子として振られ、*Bid* はそのユニットに写っている被写体建物オブジェクト、*Stp* はユニットが写り始める開始日時・時刻・フレーム番号(1/30 秒で 1 フレーム)を表現した数値、*Etp* にも同様の終了時刻の数値がそれぞれ格納される。したがって、ユニットが持つ属性情報とは、ユニット識別子に加えて 1. 建物情報と 2. 開始・終了時刻の時間情報である。

AllContents

Uid	Bid	Stp	Etp
$u_{1}^{b_1}$	b_1	2005011018563225	2005011018571405
$u_{2}^{b_1}$	b_1	2005011018581918	2005011018585826
⋮	⋮	⋮	⋮
$u_{46}^{b_1 \vee b_2 \wedge b_3}$	$b_1 \vee b_2 \wedge b_3$	2005011019050511	2005011019050916
⋮	⋮	⋮	⋮
$u_{60}^{b_1 \vee b_2 \wedge b_3}$	$b_1 \vee b_2 \wedge b_3$	2005011019551520	2005011019551822
⋮	⋮	⋮	⋮

図 6 全ユニットを表す AllContents テーブル

図 6 に示す通り、3.2 節で示した、ユニット和・積・差演算を用いて定義される全ユニットをテーブルに格納することができれば、このテーブルを用いて、全ユニットを対象とした時間的関連の問合せを行うことが出来る。

しかし、和・積・差を用いた被写体建物オブジェクトの組み合わせ総数は、 $2 \times 3^n - 2n + 1$ であり、このテーブルを作成するために、全ユニットの計算を行おうとすると指数オーダーの計算時間を必要とし、とても実現的ではない。

そこで、AllContents テーブルを持つのではなく、基本ユニットのみを格納したコンテンツテーブル Contents のみを持ち、そこに格納されている基本ユニット群にユニットの和・積・差演算を再帰的に適用して問合せを表現する体系を考える。

Contents

Uid	Bid	Stp	Etp
$u_{1}^{b_1}$	b_1	2005011018563225	2005011018571405
$u_{2}^{b_1}$	b_1	2005011018581918	2005011018585826
⋮	⋮		
$u_{m}^{b_n}$	b_n		

図 7 基本ユニットを表す Contents テーブル

図 7 に示すテーブルが多項式オーダーで計算可能であれば、このテーブルを基にユニット論理表現の定義を

行うことができる。以降、コンテンツテーブルと言う場合は、基本ユニットを表すコンテンツテーブル Contents のことを指す。

4.2.2. コンテンツテーブル計算量

コンテンツテーブルの計算量を算出するため、以下にテーブル作成アルゴリズムを示す。

開始フレームから順に被写体建物オブジェクトを抽出し、例えばある建物が開始フレームから連続してフレームに写っていて、その次のフレームに写っていないければ、前フレームにてその建物は写り終わったとみなし、1 ユニットが完成したとしてコンテンツテーブルに格納する。

まず、ビデオ *v* のフレーム数を *m* とし、*v* に写っている被写体建物オブジェクトの総数を *n* とする。アルゴリズム中に使用するテーブル ObjQueue は被写体建物オブジェクト名を表す *Bid* (primary key)、開始時刻を表す *Stp*、終了時刻を表す *Etp*、状態フラグを表す *Flg* を属性としてもつ。

[コンテンツテーブル作成アルゴリズム]

start

①第一フレーム目に映っている全被写体建物オブジェクト情報を抽出順に ObjQueue に格納する

(Stp=0,Etp=0,Flg=0)

for i = 2 to m //☆

i フレームから m フレームまで、以下について繰り返す

for j = 1 to k ② //◆

(k : i フレームに写っている被写体建物オブジェクト数)

if : i フレームに写っている被写体建物オブジェクト *b_j* が ObjQueue に格納されている

then : ObjQueue 内の Bid = *b_j* の Etp の値を 1 フレーム足す。Flg = 1 にする。

else : ObjQueue に *b_j* の情報を新たに追加

end for //◆

③ObjQueue 内の Flg=0 の被写体建物オブジェクトは、i フレームには写っていなかったため、ユニット完成となる。よって、コンテンツテーブルに格納。ObjQueue から削除。

end for //☆

end

上記のアルゴリズムにおいて、①の計算量が最大になるのは、第一フレーム目に全ての被写体建物オブジェクトが写っているときで、*n* である。②においては、ObjQueue のタプル数が最大(=n)で、i フレームの被写体建物オブジェクト数が最大(=n)の場合であるため、

計算量は $n \times n$ となる．③においては，Flg の値を確認するため，ObjQueue のタプル数が最大(=n)の時，計算量も最大となり， n である．

②と③は全体で $m-1$ 回繰り返されるため，総計算量は最大で $n + (n^2 + n) \times (m-1) = n^2(m-1) + mn \leq 2mn^2$ と

なる．したがって，オーダは $O(mn^2)$ であり，多項式時間でコンテンツテーブルを作成できるため，コンテンツテーブルは実現的に作成可能であることがわかる．

このテーブルを用いたユニット論理表現により，建物に関しては基本ユニットを和・積・差を用いて表現される任意の建物集合が写ったユニットを返す問合せを可能にし，時間情報に関しては，単一のユニットに対する開始時刻，終了時刻，時間長に関する問合せ，複数のユニットに対する7種の拡張した時区間論理を用いた問合せを可能にする．

4.2.3. ユニット論理表現定義

ユニット論理表現は，以下の四段階の定義から成る．

- (1) アトム の 定義
- (2) 式 の 定義
- (3) ユニット論理表現の定義
- (4) 安全なユニット論理表現の定義

コンテンツリレーション Contents をドメイン(Uid, Bid, Stp, Etp)上のリレーションとするならば，直積 $dom(Uid) \times dom(Bid) \times dom(Stp) \times dom(Etp)$ をドメインとするタプル変数を t とする．

なお， $dom(Stp) = dom(Etp)$ とし，時間を表すドメインを $dom(Time)$ とする．

【アトム(atom)の定義】

- ① t をタプル変数とするととき， $Contents(t)$ はアトムである．
- ② t をタプル変数とし， $u \in dom(Uid)$ とした時， $t[Uid] \gamma u$ はアトムである．($\gamma := , \neq$)
- ③ t と t' をタプル変数とするととき， $t[Uid] \gamma t'[Uid]$ はアトムである．($\gamma := , \neq$)
- ④ t をタプル変数とし， $b \in dom(Bid)$ とした時， $t[Bid] \gamma b$ はアトムである．($\gamma := , \neq$)
- ⑤ t と t' をタプル変数とするととき， $t[Bid] \gamma t'[Bid]$ はアトムである．($\gamma := , \neq$)
- ⑥ t と t' をタプル変数とし， R をユニット合成演算(union(和)，intersection(積)，difference(差))のうちの一つとするととき， $R(t, t')$ はアトムである．
- ⑦ t と t' をタプル変数とし， R をユニット合成演算(union(和)，intersection(積)，difference(差))のうち

の一つ， w^* をタプル変数とするととき， $\equiv(w, R(t, t'))$ はアトムである．ここに， \equiv とは w と $R(t, t')$ の同一性を表す．

- ⑧ t をタプル変数とするととき， $\tau \in dom(Stp)$ とした時， $t[Stp] \theta \tau$ はアトムである．($\theta := <, >, \leq, \geq, =, \neq$)
- ⑨ t をタプル変数とし， $\tau \in dom(Etp)$ とした時， $t[Etp] \theta \tau$ はアトムである．($\theta := <, >, \leq, \geq, =, \neq$)
- ⑩ t をタプル変数とし， $d \in dom(Time)$ とした時， $R(t[Etp] - t[Stp], d)$ はアトムである．($R := <, >, \leq, \geq, =, \neq$)
- ⑪ t と t' をタプル変数とし， R を時間的関連(equal, meets)のうちの一つとするととき， $R(t, t')$ はアトムである．
- ⑫ t と t' をタプル変数とし， R を時間的関連(before, overlaps, starts, finishes)のうちの一つとし， θ を比較演算子($<, >, \leq, \geq, =, \neq$)のうちの一つ， $d \in dom(Time)$ とするととき $R(t, t', \theta, d)$ はアトムである．
- ⑬ t と t' をタプル変数とし， R を時間的関連(during)とし， θ_1, θ_2 を比較演算子($<, >, \leq, \geq, =, \neq$)のうちの一つ， $d_1 \in dom(Time), d_2 \in dom(Time)$ とするととき $R(t, t', \theta_1, d_1, \theta_2, d_2)$ はアトムである．
- ⑭ ①～⑬で定義されるもののみがアトムである．

【式(formula)の定義】

- ① 全てのアトムは式である．
- ② P を式とするととき $\neg P$ は式である．
- ③ P_1 と P_2 を式とするととき， $P_1 \wedge P_2, P_1 \vee P_2$ は式である．
- ④ $P(t)$ を自由なタプル変数 t を含んでいる式とする．このとき， $(\forall t)P(t)$ と $(\exists t)P(t)$ も式である．
- ⑤ ①～④で定義されるもののみが式である．

【ユニット論理表現の定義】

$P(t)$ を t のみを自由変数とする式としたとき， $\{t | P(t)\}$ をユニット論理表現(unit calculus expression)という．

ここで，次の事柄に注意する： t をタプル変数としたとき， $\{t | \neg Contents(t)\}$ はユニット論理表現の式である．しかし，これはコンテンツテーブル Contents に属さないユニットからなる集合を表示し，一般的に

* w は被写体建物オブジェクトの和・積・差演算の集合で表されるすべてのユニットを格納した仮想テーブル(=AllContentsテーブル)のタプル変数とする．

無限集合となってしまうので計算のしようがなく、無意味である。したがって、意味のある質問のみを表現できるように、安全なユニット論理表現の定義を行う。

まずは、式 P のドメイン $DOM(P)$ を定義する。 $DOM(P)$ とは、 P に現れた定値と P に現れたユニットの全属性値の集合として定義する。

【安全なユニット論理表現の定義】

ユニット論理表現 $\{t|P(t)\}$ が安全(safe)とは、以下の条件を満たすときをいう。

- ① t を P (t) を真とするタプルとすると、 t の各成分値は $DOM(P)$ に属する。
- ② P に $(\exists u)Q(u)$ なる形の部分式があったとする。このとき、ある u に対して $Q(u)$ が真となるなら、その各成分値が $DOM(Q)$ に属する u_0 が存在して $Q(u_0)$ が真となる。
- ③ P に $(\forall u)Q(u)$ なる形の部分式があったとする。このとき、ある u に対して $Q(u)$ が偽となるなら、その各成分値が $DOM(Q)$ に属する u_0 が存在して $Q(u_0)$ が偽となる。

4.2.4. ユニット論理表現の例

以下にコンテンツリレーションを用いたユニット論理表現例を示す。

(C :Contents, B :Bid, U :Uid, S :Stp, E :Etp とする)

- (1) 三越のユニット。
 $\{t | C(t) \wedge t[B] = \text{'三越'}\}$
- (2) 三越かつ松屋が写っているユニット。
 $\{w | (\exists t)(\exists u)(C(t) \wedge C(u) \wedge t[B] = \text{'三越'} \wedge u[B] = \text{'松屋'} \wedge \equiv(w, \text{intersection}(t, u)))\}$
- (3) 松屋のユニットと equal の関係をもつユニット。
 $\{t | C(t) \wedge (\exists u)(C(u) \wedge \text{equal}(t, u) \wedge u[B] = \text{'松屋'})\}$
- (4) SONY と共に写っているユニット。
 $\{w | (\exists t)(C(t) \wedge (\exists u)(C(u) \wedge u[B] = \text{'SONY'} \wedge \equiv(w, \text{intersection}(t, u))))\}$
- (5) SONY より 30 秒以上前に写っているユニット。
 $\{t | C(t) \wedge (\exists u)(C(u) \wedge u[B] = \text{'SONY'} \wedge \text{before}(t, u, \geq, 30))\}$
- (6) SONY より 30 秒以上前に写っているユニットの建物名。
 $\{v | (\exists t)(C(t) \wedge (\exists u)(C(u) \wedge u[B] = \text{'SONY'} \wedge \text{before}(t, u, \geq, 30) \wedge v = t[B]))\}$
- (7) SONY が写っていて、Apple ビルが写っていないユニット。
 $\{w | (\exists t)(C(t) \wedge t[B] = \text{'SONY'} \wedge (\exists u)(C(u) \wedge u[B] = \text{'Apple ビル'} \wedge \equiv(w, \text{difference}(t, u))))\}$
- (8) SONY だけが写っているユニット
 $\{w | (\exists t)(C(t) \wedge t[B] = \text{'SONY'} \wedge (\exists u)(C(u) \wedge \equiv(w, \text{difference}(t, \text{intersection}(t, u))))\}$

(9) SONY が 10 秒以上写っているユニット

$$\{t | C(t) \wedge t[B] = \text{'SONY'} \wedge \geq(t[Etp] - t[Stp], 10)\}$$

5. UnitSQL

5.1. UnitSQL の導入

前章のユニット論理を基に、SQL に上位互換な問合せ言語、UnitSQL を定義する。UnitSQL に求められる問合せ機能は SQL に加えて、次の三種類の演算である。

- (1) ユニットの和・積・差演算とその結果ユニットを返す演算：3 種
 $\equiv(w, \text{union}(t, u))$, $\equiv(w, \text{intersection}(t, u))$,
 $\equiv(w, \text{difference}(t, u))$
- (2) 1 ユニットの時間長演算：6 種
 $R(t[Etp] - t[Stp], d)$ ($R : <, >, \leq, \geq, =, \neq$)
- (3) ユニットの時間的関連演算：7 種
 $\{\text{before, equal, meets, overlaps, during, starts, finishes}\}$

そこで、UnitSQL で上述の三種の演算に対する構文を例示すれば、次の通りである。

(a) ユニット和演算

三越または松屋が写っているユニット

```
SELECT W.*
FROM Contents T, Contents Y
WHERE T.Bid = '三越'
AND Y.Bid = '松屋'
AND \equiv(W, union(T, Y))
```

(b) ユニット時間長演算

SONY が 10 秒以上写っているユニット

```
SELECT *
FROM Contents
WHERE Bid = 'SONY'
AND \geq(Etp - Stp, 10)
```

(c) ユニット時間的関連演算

SONY より 10 秒以上オーバーラップしているユニット

```
SELECT Y.*
FROM Contents T, Contents Y
WHERE T.Bid = 'SONY'
AND overlaps(T, Y, \geq, 10)
```

5.2. UnitSQL のユニット完備性と実装法

SQL はリレーショナル代数における五つの独立な演算、和、差、直積、射影、選択について、それぞれ対応する SELECT 文が存在することでリレーショナル完備と言われた。同様に、UnitSQL がユニット論理の演算すべてを網羅しているときユニット完備(unit complete)と言うことにする。しかしこれは、前節で導入した UnitSQL が SQL にユニット論理の演算を可能とするため三種類の演算を追加したことからほとんど自

明である。

次に UnitSQL の実装法について述べる。UnitSQL 文は SQL の単なるシンタックスシュガーに留まるのではなく、プリプロセッサを作成することにより、それに等価な SQL 文か、ユニット合成演算の場合は埋込み SQL 文に変換する必要がある。図 8 に、UnitSQL 文の実装法の概略を示す。

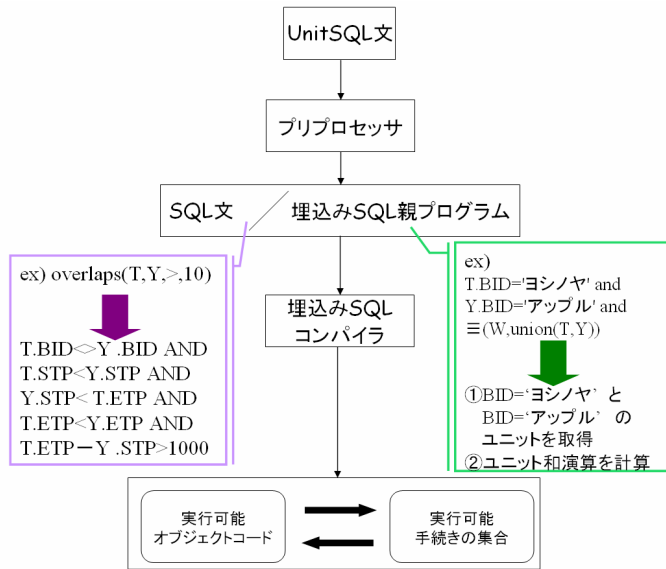


図 8 UnitSQL 実装法の概略

6. ビデオ検索手法の関連研究

これまでのビデオ検索手法としては、Pradhan ら[7]がキーワードを用いて、何が写っているか索引付けされたビデオデータが存在すると仮定し、そのビデオに対してキーワードと量化記号・和・積の代数演算を用いて検索を行う研究を行った。検索処理では、グループ演算等を用いることにより、例えば「A または B が写っているシーンがほしい」といった問合せに対して、A の映っている部分と B の映っている部分を単純に結合するのではなく、もし両者が時間的にも近ければ、同一のシーンである可能性が高いとみなし、A と B が写っていない部分をカットせずに、一つの連続したシーンを検索結果として返す。この研究は、検索結果の複数の時区間を別個として返すのではなく、意味のある形(シーン)に構成して再生することに焦点をあてており、検索部分では、「A のあとに B が写っているシーン」といった時区間論理を用いた検索や、「A が写っているが B は写っていないシーン」といった差の演算を用いた検索はできない。

Allen の時区間論理を用いたビデオ検索については、TVQL[8]がある。これは、スライダとボタンを用いることによって、時間的関連性のあるビデオをビジュアル的に指定して問い合わせることができる。このユーザインタフェースを用いて、時区間論理を直感的に表

現することができるが、スライダで指定できる時間幅が限られており、また、二つの対象に対する時間的関連についての問合せしか出来ないため、自由度が低いといえる。

7. まとめと今後の課題

本論文では、位置・姿勢情報と 3 次元地図を用いて索引付けられたビデオに対し、検索の対象となるユニットは時区間であるという点に着目したビデオ問合せ表現の体系としてユニットに関する問合せを完備したユニットの論理表現の体系化を行った。加えて、SQL の上位互換のユニット問合せ言語 UnitSQL を導入した。

現段階の映像問合せでは、ユニットの属性情報である建物名を用いた検索が可能であるが、例えば、三越や松屋であればデパート、Dior やルイヴィトンの建物であればブランド品店、などというように、その建物名によってカテゴリ分けを行って階層関係を作成し、そのカテゴリ名を建物名と同様な問合せを行えるようにすることで、よりいっそうユーザの自由な問合せが出来ると考えられる。

文 献

- [1] Frank Shipman, Andreas Girgensohn, and Lynn Wilcox Proc : "Generation of Interactive Multi-Level Video Summaries", ACM Multimedia 2003. pp. 392-401, November 2, 2003.
- [2] 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.
- [3] 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.
- [4] 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.2005.
- [5] 榎美紀, 増永良文 "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", 電子情報通信学会データ工学ワークショップ (DEWS2006), Vol.2006.
- [6] J. Allen: "Maintaining Knowledge about Temporal Intervals", Communications of the ACM, Vol.26, No.11, pp.832-843, November 1983.
- [7] Sujeet Pradhan, Keishi Tajima, Katsumi Tanaka: "A Query Model to Synthesize Answer Intervals from Indexed Video Units", IEEE Transactions on Knowledge and Data Engineering, Vol. 13, No. 5, pp. 824-838, IEEE Computer Society Press, 2001.
- [8] Hibino, S. and Rundensteiner, E. : "User Interface Evaluation of a Direct Manipulation Temporal Visual Query Language", ACM Multimedia'97 Conf. Proc., pp99-107, 1997.