社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE

3 次元地図を用いた映像データのリアルタイムな自動索引法

佐藤　有紀子† 　増永　良文‡

†お茶の水女子大学人間文化研究科博士前期課程　〒112-8610 東京都文京区大塚 2-1-1
‡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: †yukiko@dblab.is.ocha.ac.jp, ‡masunaga@is.ocha.ac.jp

あらまし ディジタルビデオカメラによって撮影された映像に対して，写しこまれているであろう建物オブジェクトを計算により自動抽出して，映像の索引としてリアルタイム処理で付与するという新しいビデオコンテンツの自動索引法を提案する．撮影者はウェアラブルコンピュータを身に付け GPS とジャイロを装着して街角でビデオ撮影を行う．時刻データと位置データ，ビデオカメラの姿勢データに加えて，データベースに格納されている 3 次元地図を使うことにより，映像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計算により自動抽出することができ，それを索引としてリアルタイムで付与するアルゴリズムを開発して，リアルタイム性の実現による効率の良い映像データベースの構築を行った．

キーワード GIS， 映像データベース，ウェアラブル，インデクシング

A Real Time Automatic Indexing Method of Video Data using a 3-Dimentional City Map

Yukiko SATO† and Yoshifumi MASUNAGA‡

† Division of Mathematics and Computer Science, Graduate School of Humanities and Sciences,
Ochanomizu University 2-1-1 Otuka, Bunkyo-ku, Tokyo, 112-8610 Japan
‡ Department of Information Science, Ochanomizu University 2-1-1 Otuka, Bunkyo-ku, Tokyo, 112-8610 Japan
E-mail: †yukiko@dblab.is.ocha.ac.jp, ‡masunaga@is.ocha.ac.jp

Abstract This paper presents an real time and automatic indexing method for digital video contents. The method computes automatically the building objects in a city, which should be captured by a digital video camera. This is done by extracting the objects among the candidate objects using the following data: Location and time data of a video shooter taken by a GPS, posture data of a video camera taken by a Gyro attached on the camera, and a 3-dimensional city map stored in a database. The automatic calculation identifies the start and the end video frame for each building object to be captured in a video unit. An index is created to refer to the set of all video units where a specified building is really captured in real-time. The proposed algorithm, automatic indexing video frames in real-time, is implemented, and the system makes a real-time indexed database, so that users can access to the database as they take a data.

Keyword GIS, Video Database, Wearable, Indexing

1. はじめに

近年，ビデオカメラの小型化やバッテリの長寿命化が進み，ビデオカメラを自在に持ち歩いて長時間移動しながら多数の映像を撮影することが多くなっている．それに伴い，当然のこととして，大量の映像が取得されることとなった．また，一方では，MPEG に代表される映像の圧縮技術と標準化が進み，映像をデータベース化することが，今日の大容量ストレージ技術の進歩と相まって可能となっている．

しかし，大量な映像は格納されるだけでは価値が無く，的確に検索できうることが肝要である．そのため

には，映像に的確な索引付けがなされていなければならない．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．後者の問題は映像のデータベース化が叫ばれた当初から多大の関心を呼び，これまでに多くの研究がなされてきているが，問題は単純ではない．

映像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．ディジタルビデオ処理の研究・開発は次のように分類できる[1,2]：

(a) ショットを検出する（shot detection）研究．

(b) シーンを検出する（scene or story detection）研究．関連して，シーンの類似検索に関する研究などを含む．

(c) 映像あるいは映像の部分列にメタデータを付与する研究．メタデータは索引（index）あるいは注釈（annotation）と読み替えてもよい．従来，映像からオーディオビジュアルな特徴量を抽出してその内容記述を行おうとする研究が多数行われてきた．画像理解技術，被写体オブジェクトの抽出と追従技術，音声認識技術，話者認識技術，文字認識技術，あるいはカメラワーク情報などの手法が用いられている．

(d) 映像検索技術の研究．問合せを発行して所望の映像クリップやシーンを映像データベースからインタラクティブにあるいは自動で効率よく得る研究．

本研究は上記(c)項に類別されるが，特徴量を抽出して索引付けを行おうとする従来型の方法ではなく，映像に何が写っているかを直接計算して映像の部分列（これをユニットということにする）に 索引を付与しようとする研究である[3,4,5]．このような発想を現実のものとするために，我々が想定している映像収録環境は次のとおりである：

・GPS(Global Positioning System)によりビデオ撮影者の位置と時刻がわかる．

・ビデオカメラに取り付けたジャイロセンサによりビデオカメラの姿勢を知る　　ことができる．

・撮影地点を含む３次元地図があり，建物オブジェクトの３次元データが取得可能である．

次章で示すように，これらのデータを用いることにより，ビデオカメラの視野に入り，実際にビデオに写っている建物オブジェクトを計算して，自動的にユニットにその建物 ID を索引として付与することができる．図 1 は銀座エリアにおける 3 次元描画ソフト AutoCad（AutoDesk 社）により描かれた 3 次元地図のイメージである．地図上の 3 角錐は撮影者によってビデオカメラに撮られた建物の例である．

関連研究として上田らの研究[6]がある．そこでは，ビデオ撮影者が自分の過去の行動を振り返る際に必要な機能としてビデオ検索を位置付け，撮影場所に注目した索引機構を提案している．これは，ビデオ撮影時に GPS を用いて撮影場所の位置情報を緯度・経度の形で取得し，それを地名やランドマーク名といった地理情報に変換し，それらをキーワードとしてビデオ検索を実現するというものである．しかしながら，この研究では，用いた地図は２次元であり，また被写体建物オブジェクトの抽出計算も行わなかったので，実際にビデオに写っているオブジェクトと索引が一般には一致せず，たとえば，我々の方法では答えられる「銀座三越デパートが 10 秒以上写っているビデオが欲しい」などという検索要求に答えられない．

我々は GPS とジャイロセンサを用いて得られるビデオカメラの位置・姿勢データに加えて，“3 次元地図データ”を用いることにより，映像に写しこまれているであろう建物オブジェクトを計算して自動的に索引付けをする研究（被写体建物オブジェクトの自動抽出アルゴリズム）を行ってきた[2，3，4]．しかしこれらは，完全なオフライン作業による映像の索引付けであり，映像を取得した後， オフラインで映像取得時の GPS 等のデータ，地理データを処理して映像の索引付けを行っていた．しかしこの方法ではその場での索引付き映像データベース生成が不可能であることから，リアルタイムでの検索が不可能になってしまう．

本研究では， 映像データに被写体建物オブジェクトを“リアルタイム”で索引として付与し，映像を撮影すると同時に映像データベースを構築してゆくアルゴリズムを開発することで，リアルタイム検索ができる映像データベースシステムの構築を行った．

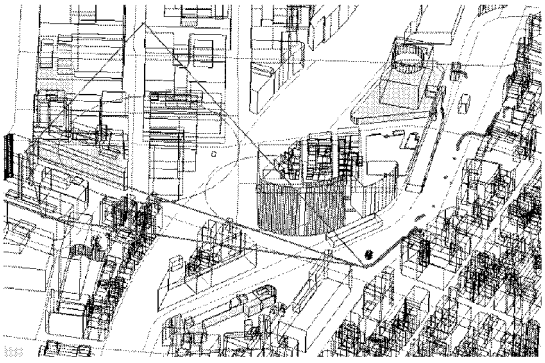

図 1. 3次元地図を用いた被写体建物オブジェクトの棚略
Fig.1 3-Dimensional Map for the use of Automatic Extraction of Building Objects

2. 被写体建物オブジェクトの自動抽出と索引付け

2.1. 自動抽出・索引付けシステムの概念

図 1，図 2 は我々が開発しているシステムの概念図である．ビデオ撮影者の位置と時刻を取得するために GPS，ビデオカメラの姿勢を知るためにジャイロを使用する．

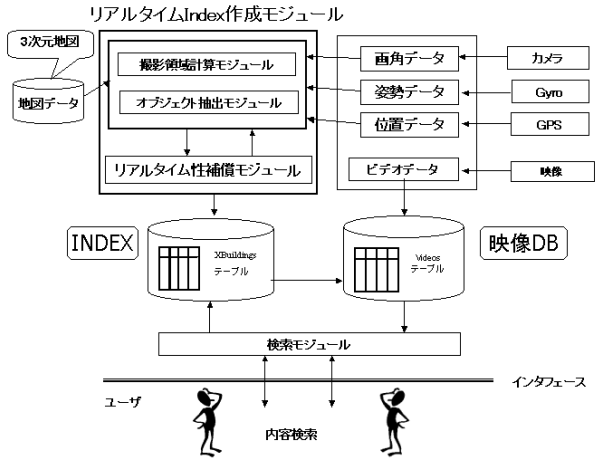

図 2. 3 次元地図を用いた被写体建物オブジェクトの自動抽出・索引付けシステムの概念

Fig.2 Concept of an Automatic Extraction and Indexing System of Building Objects using a Three-Dimensional Map

撮影者が身につけているウェアラブルコンピュータで，GPS データ，ジャイロデータ，カメラの画角データ（レンズの画角：今回使用するビデオカメラ（Sony 社製 DCR-PC1）の画角は横 44°，縦 35°），3 次元地図データ（三菱商事製 DiaMap）を，本論文で提案するアルゴリズムで総合的に処理して，被写体建物オブジェクト ID を取得し，その ID とその建物が写っているユニット（次節）を対応付けるインデックステーブル XBuildings が作成される．ビデオはショット単位で処理する．

2.2. ユニット：索引付けの単位

ビデオ撮影者は市街地を歩行しながら，かつビデオカメラを左右上下に振りながらビデオ撮影をする．$u_{v,o,i}$ は建物オブジェクト（o とする）が，ビデオ（v とする）のあるビデオフレーム（b とする）から始まり，あるビデオフレーム（e とする）まで連続して写しこまれている，第 i 番目の部分とする（i≧1）．このビデオフレームの連続を $u_{v,o,i}$ =(v, o, i, b, e)で表わし，ユニット（unit）と呼ぶ．図 3 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 とそれが連続して写っているユニットの対が図 2 の INDEX データベースの XBuildings テーブルに記録されるので，少なくとも(O_1, $u_{v,o,1}$) と(O_1, $u_{v,o,2}$) の 2 つのタップルが存在する．その結果，建物 ID で問い合わせると，その建物が写っているユニット全てを知ることができる．映像は 1 秒間に 30 フレームづつ撮影されるので，ショットの撮影開始時刻を GPS データより取得すれば，各フレームの撮影時刻も割り出せる．

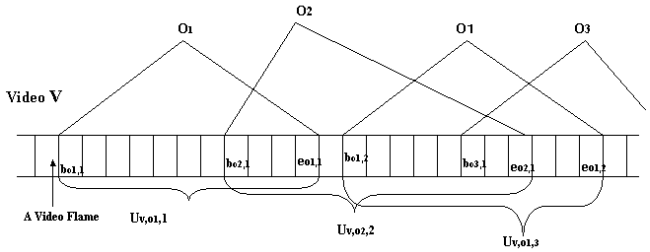

図 3. 被写体建物オブジェクトとユニットの関係

Fig.3 Relation between Building Objects and Units

3. 被写体建物オブジェクトの自動抽出アルゴリズム

3.1. 基本的考え方

ビデオカメラで撮影されているべき 3 次元建物オブジェクトの抽出アルゴリズムとその実装を示す．基本的な考え方は，建物は宙に浮いているわけではないので，図 2 のリアルタイム Index 作成モジュールに示したように，まず，3 次元のビデオ撮影領域を 2 次元の地図上に写像して撮影領域を特定し，続いて 3 次元地図から建物の高さ情報を取得して，その領域に存在して実際に写っているべき建物オブジェクトを計算する[3,4,5].

(1) 2 次元地図上での撮影領域の特定

ビデオ撮影時刻 τ で，2 次元平面地図で写っているべき空間オブジェクトの抽出を行う．図 4 の四角錐(P)の部分がビデオカメラで撮影される 3 次元領域であり，斜線の 3 角形の部分（T）はその 3 次元領域を 2 次元平面（X-Y 平面）に投影することにより求まる．今回は使用していないが将来ビデオカメラのズーム機能を使用することを考慮して，ジャイロセンサのヨー角（α）は横の画角の中心，ロール角（β）は縦の画角の中心からの角度と設定する．L は視野の距離を現すが，ビル街等建物の密集している地域では L が小さく，野原など見晴らしの良い地域では L を大とする必要がある．本研究では L=80m でビル街（銀座）での実験を行っている．使用するビデオカメラの横の画角は 44°で，2 次元平面上への投影視野 l は，$l = L \times \cos(\beta)$で求まることにより，T は 3 つの頂点：T'=(0, 0), T_0=($l \times \cos(\alpha-22)$, $l \times \sin(\alpha-22)$), T_{44}=($l \times \cos(\alpha+22)$, $l \times \sin(\alpha+22)$)で囲まれる三角形と定められる．被写体建物オブジェクトはその底面が T の中もしくは T と交わる建物オブジェクトでなければならないことがわかる．

(2) 被写体建物オブジェクトの抽出

次に，3 次元地図 DiaMap から建物オブジェクトの高さデータを加え，実際に写っているべき建物オブジェクトを抽出する．ビデオカメラの縦の画角(35°)，撮影者の身長（変数 h で表す）により，5 つの頂点：P_1=(0,

0， h)， P_2=(l × cos(α-22)， l × sin(α-22)， L × sin(β+17.5)+h)， P_3=(l×cos(α+22)， l×sin(α+22)， L ×sin(β+17.5)+h)， P_4=(l×cos(α+22)， l×sin(α+22)， L ×sin(β-17.5)+h)， P_5=(l×cos(α-22)， l×sin(α-22)， L ×sin(β-17.5)+h) からなる四角錐 P の範囲内，もしくは P と交わるオブジェクトが，(1)で得られた候補のうちで，実際に写っている建物オブジェクトの候補となるが，自分より前，つまり自分より撮影者に近い位置に自分より高い建物がある場合には実際には映像に写っていない．

次節ではこれらを考慮した被写体建物オブジェクト抽出のアルゴリズムを説明する．

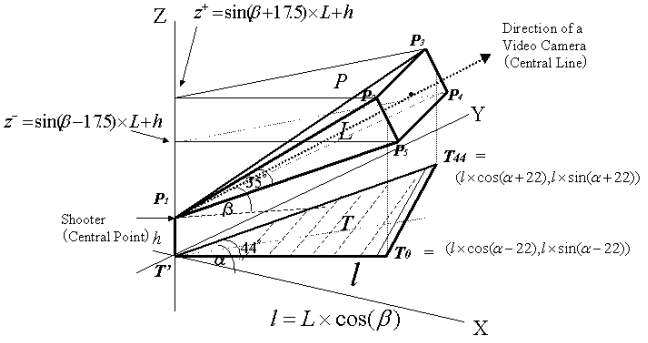

図 4. ビデオ撮影領域の 3 次元空間から 2 次元空間への投影

Fig.4 Mapping of a Video Captured Area from Three-Dimensional to Two-Dimensional Space

3.2. 被写体建物オブジェクトの抽出アルゴリズム

図 5 に，前節の考え方に基づき実装した被写体建物オブジェクトの自動抽出アルゴリズムの概略を示す．

3.2.1. 建物オブジェクトに関するパラメタ

抽出アルゴリズムは，建物オブジェクト（o）に関して次のパラメタを有する．

- 底面の重心座標: (o.x， o.y)
- 高さ: o.height
- 撮影者から重心への距離: o.r
- 建物オブジェクトが撮影地点から見えている/見えていないのフラグ: o.visible

3.2.2. 2 次元地図上での建物オブジェクト抽出

2 次元地図上（DiaMap の 2 次元地図のみを使用）に投影された撮影領域中の建物オブジェクトを抽出する．3.1 で述べたように，T の中もしくは T と交わる建物オブジェクトがビデオカメラに写るべき候補である．図 4 に示したように，T 内で直線 T'T_i (T_i=(l ×cos(α-22+i)， l ×sin(α-22+i)) を 1°づつ(0≦i≦44)動かしていき，直線 T'T_0 から直線 T'T_{44} まで交差する建物オブジェクトを地理情報システム ArcView3.2（ESRI 社製）に備わっているフィーチャ選択機能を用いて抽出し，各 i に対して配列 a[ ][ ]={a[i][ ] | i=0， …，44}に r の昇順にその建物 ID を格納していく．

3.2.3. 高さ情報を用いた被写体建物オブジェクトの抽出

次に，3 次元空間的視点により，DiaMap から建物オブジェクトの高さデータを加え，実際に写っているべき建物オブジェクトを抽出する．

ある i において直線 T'T_i 上にある建物オブジェクト（o）を全て配列{a[i][0]， a[i][1]，…，a[i][n_i]}に o.r の昇順に格納する．ここで n_i とは，a[i][n_i+1]=null であるが，ある n(n ≦ n_i)においては，どのような a[i][n] も null ではない数である．すなわち直線 T'T_i と交わった建物オブジェクトは n_i+1 個ある．次に配列の 1 番目に格納された建物オブジェクト（a[i][0]）の高さとその位置での P の底辺の高さ (a[i][0].r×tan(β -17.5) +h)を比べ，高ければ a[i][0].visible の値を“on”とし，そうでなければ“off”として，変数 s の初期値を 0 とする．

次に図 5 の内側のループにおいて，m=s+1 とおく．配列のはじめの建物オブジェクトの高さ（a[i][s].height）と配列の 2 番目以降のオブジェクトの高さ（a[i][m].height）を比べ，後者が高く且つ P の底辺の高さ (a[i][m].r×tan(β -17.5) +h)以上ならば，後者のフラグの値（a[i][m].visible）を“on”にし s=m とする．そうでなければ“off“とし，m=m+1 とする．これを a[i][m]が null でない限りループをまわして配列（a[i][]）に格納されている全ての建物オブジェクトの visible のフラグを設定する．すべての i の値で調べ終わり，最終的に visible のフラグが立っている全てのオブジェクトが，時刻 τ において映像に写っているべき建物オブジェクト群である．

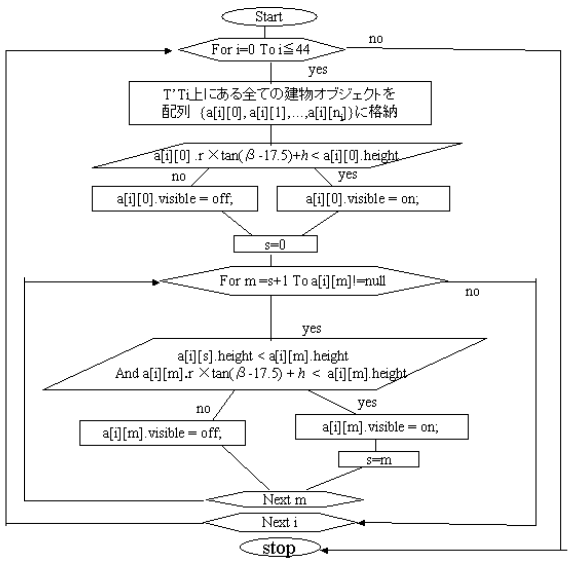

図 5. 自動抽出アルゴリズムの概略

Fig.5 Outline of Automatic Extraction Algorithm of Building Objects

4. XBuildings テーブルのリアルタイム作成法

本研究では街中で映像を撮影中，もしくは撮影直後に索引付き映像データベース（XBuildings）にアクセスして所望の映像を取得できるようにする．そのため索引付き映像データベース構築のリアルタイム性が重視されるが，リアルタイムでシステムを実装するためには処理時間等，様々な不具合が生じる． 本章では図２のリアルタイム Index 作成モジュールの中に示した，映像フレームに対して動的に間引くことを考えリアルタイム性を補償する XBuildings テーブルを生成するモジュール,”リアルタイム性補償モジュール”の説明を行う．

4.1. 基本的考え方

映像は 1/30 秒毎に所得されるので，各フレームに写っているべき建物オブジェクト群の抽出とそれに基づいたビデオの索引付けが常に 1/30 秒以内に行われるなら特段の方策を労することなくリアルタイム性が保証されるといえる．しかし，このリアルタイム性は大別すると次の要因で阻害されることが考えられる．

(a) 1 フレーム分の被写体オブジェクト群の抽出が 1/30 秒以内に終了しない．

(b) (よしんば(a)項のリアルタイム性が保障されたとしても，)次々と抽出されるオブジェクト群を取得してユニットを構成し，かつユニットを XBuildings テーブルに格納する処理で，時間遅れが出たり，不都合が出たりする．

そこで，我々は，処理の対象となる映像フレームを動的に“間引いて”リアルタイム性を補償することを考えた．

4.2. 映像フレームの動的間引き

映像の第 i(i≧1)フレームの被写体オブジェクト群の抽出処理をする．この処理に p 秒を要したとする．次に処理するフレームは，p に依存して，$i+\lceil p\times 30\rceil$ (＝p×30 を下回らない最小の整数)番目のフレームとする．(換言すると，この間のフレームが処理の対象から間引かれたことになる．)

さらに，ユニット構成・格納処理で時間遅れが出たときは(この状況は 3.5 節で詳述)，被写体オブジェクト抽出の対象となるフレーム間隔に 1 を加える．

あるフレーム間隔でリアルタイム性が一定時間保障されていれば，間引くフレーム数を 1 減じる．

なお，フレーム取得の最小時間単位 1/30 秒毎に映像データに索引付けがなされることは理想的かもしれないが，人が認識できる映像はその長さが 3 秒以上であるという報告[7]もあり，また，我々の予備実験では，時間間隔を 1 秒にとるとリアルタイム索引付けが問題なく行われているようなので，上記の考え方に無理はないと考えられる．

4.3. リアルタイム作成アルゴリズムの概要

図６は上記の考え方で映像データをリアルタイムで索引付けするアルゴリズムの概要である.”リアルタイム性補償モジュール"は"データ取得決定モジュール"と被写体建物オブジェクトで抽出された建物オブジェクトのストリームを一時的に格納するキュー"ObjectQueue"と"ユニット構成モジュール"から成り，画角データ･姿勢データ・位置データの取得間隔をコントロールすることでリアルタイム性を補償する．各モジュールの働きを下に示す．

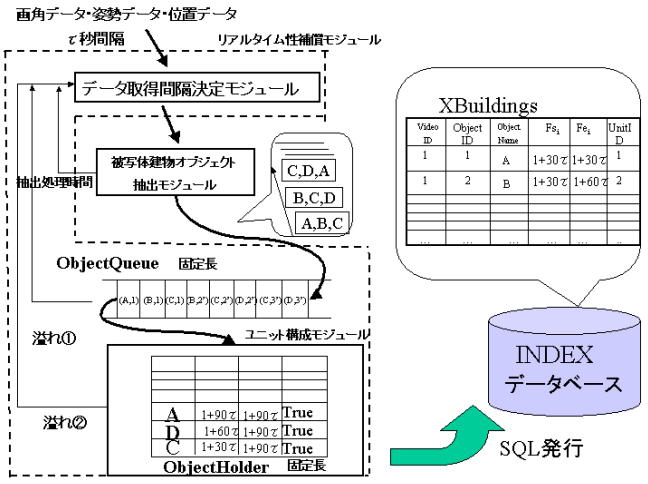

図 6: リアルタイム性補償モジュール

Fig6. Real-Time Compensation Module

- **データ取得間隔決定モジュール:** 被写体抽出オブジェクトモジュールから抽出処理時間を取得し，次の処理するための間隔を決定する．また，キューの ObjectQueue，配列 ObjectHolder は固定長となるため，溢れが発生した場合には次の処理間隔をずらす．これが間引きのタイミングにつながる．
- **ObjectQueue :** キューの働き．被写体オブジェクト抽出モジュールによって抽出されたオブジェクトがストリームとなってフレーム番号と供に ObjectQueue で一時的に保持される．ObjectQueue の渡したオブジェクトがユニット抽出モジュールで処理されるとすぐに，次のオブジェクトを渡す．
- **ユニット構成モジュール：**ObjectQueue から渡された 1 フレームに相当するオブジェクト群を固定長の配列 ObjectHolder に取り込み，“ユニットとして完全なタップル”ができれば，それを XBuildings テーブルに挿入するための SQL 文を発行する．その処理が終ったら，ObjectQueue から次のオブジェクト群を取り込み，処理を続ける．ここで“ユニットとして完全なタップル”とはある ObjectID が毎回フレームで被写体オブジェクトとして連続して抽出されていて，あるフレームで抽出されなくなる時のユニットのことである．ユニットを構成し，SQL 文を発行する・しないの見分けをす

るためには，新たに ObjectQueue から渡されたオブジェクトと既に ObjectHolder にユニットを構成するオブジェクトとのマッチングをする必要がある．マッチングは以下の 3 通りに分けられる．

①新規オブジェクトの場合新規ユニットを構成する，

②オブジェクトが ObjectHolder にユニットを構成するオブジェクトとして存在する場合，そのユニットのフレーム情報を更新する．

③存在するユニットの中で更新されなかったユニットは”ユニットとして完全なタップル”とみなされ SQL 文を発行し，このタップルを削除する．（詳しくは次節のアルゴズム）

4.4. リアルタイム作成アルゴリズム実装

図 7 に前節の考え方に基づき実装した XBuildings テーブルのリアルタイム作成のアルゴリズムである． "リアルタイム性補償モジュール”がどのように稼動してリアルタイムで映像データの索引付けが出来るのかを具体的に説明する.特に，ユニット構成モジュールの場合分けに注目する．

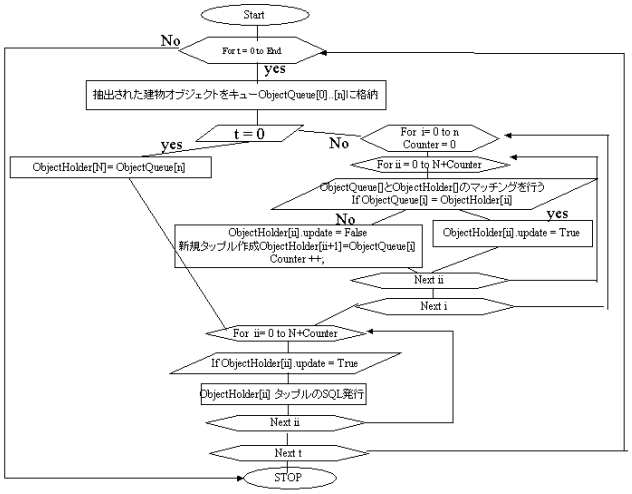

図 7: XBuildings テーブルのリアルタイム作成アルゴリズム

Fig7. Algorithm of Creating XBuildings Table in Real Time

4.4.1. XBuildings テーブルの属性

図 8 で示すように，XBuildings テーブルは映像番号（VideoID)，建物番号(ObjectID)，建物名(ObjectName)，開始フレーム番号（Fs)，終了フレーム番号(Fe)，一意となるユニット番号(UnitID)の属性値を持つ．

4.4.2. パラメタの説明

配列 ObjectHolder[N]は構造体として以下のパラメタを有する．

- 建物名：ObjectName
- フレーム開始番号：Fs
- フレーム終了番号：Fe
- ユニット番号（一意）：UnitID
- アップデートされている/いないのフラグ：Update

XBuildings

VideoID	ObjectID	Object Name	Fs_i	Fe_i	UnitID
1	1	銀座松屋	1	2100	1
1	2	SONY	1	7800	2
1	1	銀座松屋	5800	24000	3
…	…	…	…	…	..

図 8: XBuildings テーブル

Fig8. XBuildings Table

4.4.3. 配列 ObjectHolder への格納と SQL 文発行

データ取得間隔決定モジュールにより決められた間隔（τ）で被写体オブジェクトモジュールによって抽出されたオブジェクトはストリームとしてフレーム番号と供に次々にキューObjectQueue に挿入される．

第一フレーム（時刻 t=0)で映っているであろうオブジェクト A，B，C が抽出され ObjectQueue に格納されたとする．ユニット構成モジュールはこれらのオブジェクト群を ObjectHolder にフレーム番号と供に格納する．

次に，第二フレーム（時刻 t=τ）の処理がなされ，オブジェクト B，C，D が被写体オブジェクトとして抽出され ObjectQueue に格納されたとする．ユニット構成モジュールは第一フレームの ObjectQueue のオブジェクトの処理が終っていれば，第二フレームの ObjectQueue のオブジェクトを ObjectHolder に取得し既存のオブジェクトとマッチングをする．このとき B，C オブジェクトは既に ObjectHolder に存在するのでフレーム番号と供にそれらのタップルを更新する（フラグ Update=True)．D オブジェクトは新規なので新たにタップルを作成する（フラグ Update＝Ture)．一方，1 番目に抽出されたオブジェクト A は今回は抽出されたかったので（フラグ Update＝false）A のタップルは“ユニットとして完全なタップル”となるので，フレーム開始番号，終了番号と供に XBuildings テーブルに格納するべく SQL 文が発行され，ObjectHolder から削除される．

次に，第三フレーム（時刻 t ＝2τ）の処理がなされ，オブジェクト C，D，A が被写体オブジェクトとして抽出され ObjectQueue に格納されたとする．ユニット構成モジュールが以前のタップルの処理中でなければ第三フレームの ObjectQueue のオブジェクトを

ObjectHolder に取得しマッチングする．このとき C，D オブジェクトは既に ObjectHolder に存在する（フラグ Update＝Ture）のでフレーム番号と供にそれらのタップルを更新する．A オブジェクトは t=0 の ObjectHolder にはタップルとして存在していたが，t=τ で削除されているため新規タップルとして作成する（フラグ Update＝Ture）．一方，オブジェクト C，D は今回は抽出されなかったので（フラグ Update＝False）C，D のタップルは“ユニットとして完全なタップル”となる．その結果フレーム開始番号，終了番号と供に XBuildings テーブルに格納するべく SQL 文が発行され，ObjectHolder から削除される．以下，t が END になるまで，すなわちデータを取り終わるまで同様に繰り返す．この方法により刻々と抽出された被写体建物オブジェクト群が ObjectHolder に流れ込むと同時に，XBuildings を構築していくというリアルタイム性を実現することができる．

4.5. 溢れによるリアルタイム性阻害要因の検出

4.1 節で，リアルタイム性を阻害する 2 つの要因を挙げたが，(b)項の要因は，さらに次の 2 つに分かれる．

- ObjectQueue が溢れる(図 6 の①)．ユニット構成モジュールの処理に時間がかかり，被写体オブジェクト抽出モジュールが供給してくるオブジェクト群の処理に追いつけない．
- ObjectHolder が溢れる(図 6 の②)．ObjectHolder 中のタップルがなかなかユニットとして完全なタップルにならず，かつそのようなタップルが増加していく．(ObjectHolder のサイズをある程度大きくしておけば回避できる．)

なお，データベース管理システムの SQL 挿入文処理は，それを発行すると瞬時にコミットしており，リアルタイム性阻害の要因にはなっていない．

図 6 に示すように，オブジェクト抽出の対象となるべきフレーム(間隔)を決める「データ取得間隔決定モジュール」は 3 入力である．そして，「リアルタイム性補償モジュール」はそれらから成り立つ．

5. リアルタイムアルゴリズムの検証

XBuildings テーブルのリアルタイム作成アルゴリズムの一部実装を行った．提案したアルゴリズムの基本的な有効性を検証するために，ObjectHolder は配列 30 として，銀座エリアで収録した 10 分間の映像がリアルタイムで索引付け可能かどうか検証した．

実装環境は以下のようとする．

- OS：Windows XP
  (CPU:Mobil intel Pentium4, RAM:256Mbyte)
- 描画ソフト：
  ゼンリン社製　Zmap-Town Ⅱ，
  Autodesk 社製　AutoCad Map 3D
- 開発言語：　VB.NET
- データベースシステム：　Microsoft Access

その結果，この状況では ObjectQueue と ObjectHolder に溢れが生じることなく時々刻々と図 9 に見られるように被写体建物オブジェクトが抽出され（平均取得間隔 0.5 秒），同じ銀座エリアでも建物が密集しているエリアの場合，被写体建物オブジェクト抽出モジュールの抽出処理時間が 1 秒を超えた．オブジェクトが抽出されると同時に 図 10 に見られるようなデータベースがリアルタイムで作成され，リアルタイムで XBuildings テーブルは 1 秒ごとに即時更新される状況が確認できた．また，インタフェースからの検索操作は同時実行可能であった．なお，10 分後の XBuildings テーブルのタップル数は 188 であった．

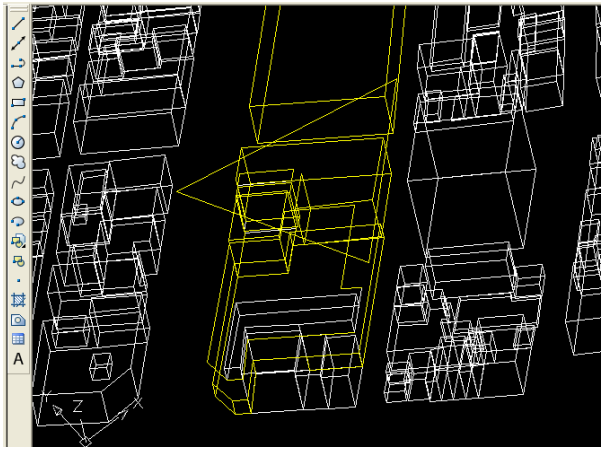

図 9: 3 次元での被写体建物オブジェクト抽出画面

XBuildingsRealtime：テーブル

ObjectName	Fs	Fe	Duration	Unit
銀座シルクビル	1	1	1	1
銀座三越	1	5	5	2
銀座アクトビル	1	13	13	3
銀座三和ビル	1	27	27	4
松屋銀座	1	98	98	5
日本香堂ビル	66	155	90	6
池田屋ビル	66	138	73	7
松島眼鏡店本社	66	135	70	8
マツザワビル	66	145	80	9
玉屋ASビル	67	167	101	10
サエグサ本館	69	460	392	11
井上商会ビル	71	76	6	12
煉瓦亭	80	85	6	13
島田ビル	81	94	14	14
石川ビル	82	149	68	15
第二島田ビル	83	147	65	16
銀座サニービル東側	97	123	27	17
喜代永ビル	102	166	65	18
山野ビル	106	171	66	19
菱進銀座ビル	109	137	29	20

図 9:　リアルタイム生成された XBuildings テーブル

Fig8.　XBuildings Table created in Real Time

6.　まとめと今後の課題

本論文では，GPS とジャイロセンサデータに加えて，3 次元地図を使うことで，映像に写し込まれている建物オブジェクトを自動計算する考え方と，その自動計算アルゴリズムを提案し実装をした．さらに，この計算に基づいて，映像の建物オブジェクトによるリアルタイムな自動索引法を考案，一部実装した．

今後の課題として実際に検証実験を行い，アルゴリズム，リアルタイム性の効率を検証したい．実験の際にはシステムの正確性・誤差がインデックスを付けるにあたり重要となってくる．そこで，GPS の精度はデファレンシャル，ジャイロセンサの精度の検証，街路樹などの障害物に対する対処法，建物屋上に設置された看板類の扱い，視野長 L の設定法，建物オブジェクトの重要度の考慮や富士山・東京タワーといったランドマークの扱い方，さらに高度な索引付けの研究・開発が挙げられる．

文　献

[1] Gaughan, G., Smeaton, A., Gurrin, C., Lee, H., McDonald, K.: “Design, Implementation and Testing of an Interactive Video Retrieval System, ” Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval, pp.23-30, November 2003.

[2] Wang, Y., Ostermann, J., and Zhang, Y-Q.: “Video Processing and Communications, ” (book) Prentice Hall, 2002.

[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），Vol2003，№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ィ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, Yoshifumi Masunaga. :”A Novel Indexing Method for Digital Video Contents using a 3-Dimentional City Map ”, Proceedings of the 4th International Workshop on Web and Wireless Geographical Information Systems (W2GIS) , pp.333-343, November 2004.

[6] 上田隆正，天笠俊之，吉川正俊，植村俊亮：“位置情報と時刻情報を用いた映像データの索引付け手法，”電子情報通信学会第 12 回データ工学ワークショップ（DEWS2001），2001 年 3 月．

[7] HonJian Zhang, Chien Yong Low, Stephen W.Smoilar, JianHua Wu :”Video Parsisng, Retrieval and Browsing”, Intelligent Multimedia Information Retrieval, ed. Mark T.Maybury, PP.139-158, MIT Press, Massachusetts, 1997.