

3次元地図を用いた映像データのリアルタイム索引法

Real-time Indexing of Video Data using a Three-Dimensional Map

佐藤 有紀子[▼] 増永 良文[▲]
 Yukiko SATO Yoshifumi MASUNAGA

撮影者は街角で GPS とジャイロを装着されたビデオカメラで家並みを撮影する。時刻データと位置データ、ビデオカメラの姿勢データ、カメラの画角データに加えて、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自動抽出し映像に建物名を索引としてリアルタイムで付与することができるアルゴリズムを開発・検証したので、それを報告する。

This paper presents a real-time automatic indexing method for digital video contents. The method computes automatically the building objects in a city captured by a digital video camera with a GPS for time and position data and a Gyro for posture data of the camera. In addition, a 3-dimensional city map is used to identify the start and the end frame for each building object to be captured in a video unit.

1. はじめに

デジタル映像処理の研究には、(a) ショット検出 (shot detection) の研究、(b) シーン検出 (scene or story detection) およびシーンの類似検索の研究、(c) 映像あるいは映像の部分列にメタデータ、つまり索引 (index) あるいは注釈 (annotation) を付与する研究、(d) 映像検索技術の研究などがあり、これまで多くの研究がなされてきた[1, 2]。

本研究は上記(c)項に類別されるが、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従来の技術を用いて特徴量を抽出して索引付けを行おうとするのではなく、GPS、ジャイロ、ビデオカメラのレンズの画角、撮影地点の3次元地図を使って、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)に被写体オブジェクトを索引として付与しようとする研究である[3, 4, 5. 関連して 6]。本論文はこの研究をさらに進めて、撮られた映像に“リアルタイム”で索引付けをするアルゴリズムを開発して、その有効性を検証した結果を報告するものである。

▼ 学生会員 お茶の水女子大学大学院人間文化研究科博士前期課程 yukiko@db.is.ocha.ac.jp
 ▲ 正会員 お茶の水女子大学理学部情報科学科 masunaga@is.ocha.ac.jp

2. 被写体建物オブジェクトの自動抽出・索引付けシステム

図1は我々が開発しているシステムの概念図である。ビデオカメラ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している。カメラの画角横44°、縦35°で、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を計算により同定して、そのIDとその建物が写っているユニットを対応付けるインデックステーブル XBuildings が作成される。図中、リアルタイム性補償モジュールを除いたモジュールとテーブルの詳細は、先行研究[4, 5]を参照していただきたい。なお、ユニットに関しては、本研究に固有の概念なので、その概略を次に示す。

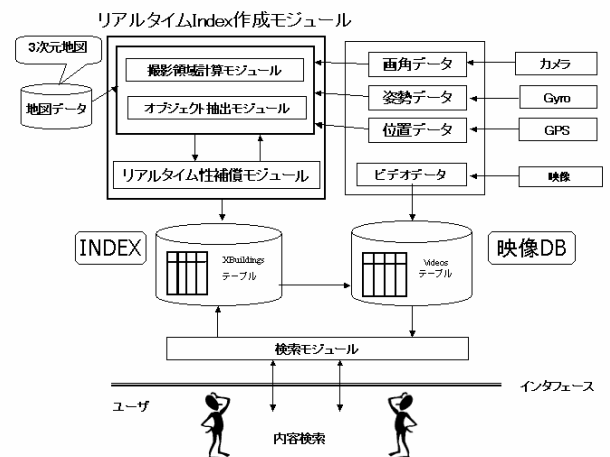


図1 3次元地図を用いた被写体建物オブジェクトの自動抽出・索引付けシステムの概念

Fig1 Concept of Automatic Extraction and Indexing System of Building Objects using a Three-Dimensional Map

撮影者は市街地を歩行しながらビデオカメラで家並みを撮影する。 $u_{v,o,i}$ は建物オブジェクト(oとする)が、ビデオ(vとする)のあるフレーム(bとする)から始まり、あるフレーム(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 とそれが連続して写っているユニットの対が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。

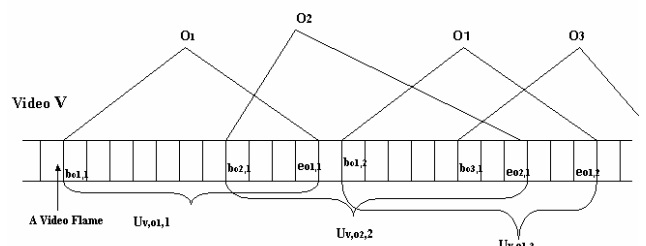


図2 被写体建物オブジェクトとユニットの関係

Fig2 Relationship between Building Objects and Units

3. リアルタイム性補償モジュール

本章では図1のリアルタイム Index 作成モジュール中に示した "リアルタイム性補償モジュール" の説明を行う。

3.1 基本的考え方

通常、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考ることなくリアルタイム性が保証される。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。

- (a) 1 フレーム分の被写体オブジェクト群の抽出計算が 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フレームごとに取得し処理して、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に時間遅れがでる。

そこで、我々は、処理の対象となるフレームを動的に "間引いて" リアルタイム性を補償することを考えた。

3.2 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に p 秒を要したとする。このとき、次に処理するフレームは、 p に依存して、 $i + \lceil p \times 30 \rceil$ ($= p \times 30$ 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 4.5 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。一方、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である 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た。

3.3 リアルタイム性補償モジュールの概要

図 3 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モジュール構成の概要である。"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索引付けの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。

- **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、次のフレーム処理をするための時間間隔を決定する。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 1 ずらす。これがフレームの間引きのタイミングにつながる。
- **ObjectQueue:** キューの働きをする。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共に 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。その処理が終わったら、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るが、あるフレームで抽出されなくなる時に決まるユニットのことである。ユニットを構成し、SQL 文を発行するかしないか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。

新規オブジェクトの場合は新規ユニットを構成する。

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データを更新する。

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プル" とみなされて SQL 文を発行し、このタプルを削除する。(詳しくは次節のアルゴリズム)

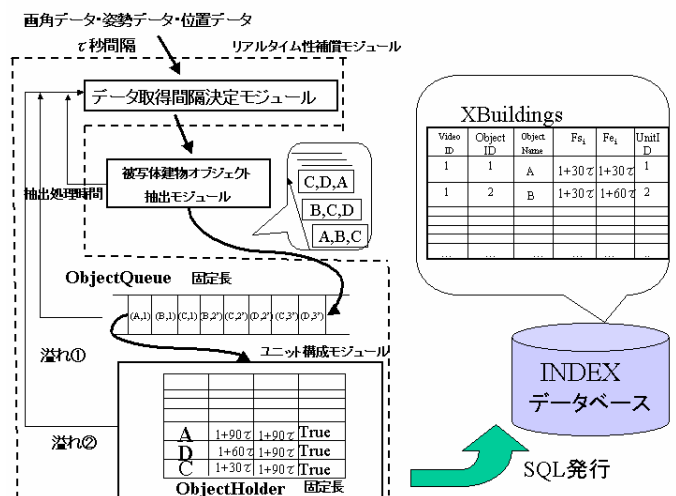


図3 リアルタイム性補償モジュール
Fig3 Real-Time Compensation Module

3.4 リアルタイム索引作成アルゴリズムの実装

図 4 に前節の考え方にに基づき実装した XBuildings テーブルのリアルタイム作成のアルゴリズムを示す。"リアルタイム性補償モジュール" がどのように稼働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。

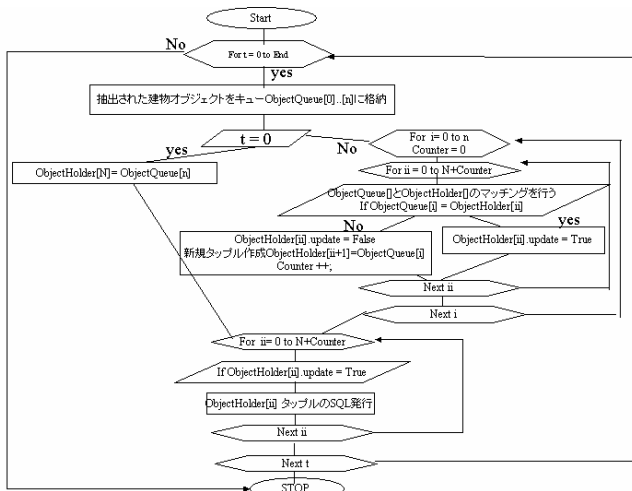


図4 XBuildings テーブルのリアルタイム作成アルゴリズム
Fig4 Realtime Algorithm of Creating XBuildings Table

3.4.1 XBuildings テーブルの属性

図 5 に示すように、XBuildings テーブルの属性は映像番号 (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)、ユニット番号(UnitID)である。

3.4.2 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5 XBuildings テーブル
Fig5 XBuildings Table

3.4.3 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 () 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群はストリームとしてフレーム番号と共に次々にキューObjectQueue に挿入される。

たとえば、第 1 フレーム (時刻 t=0) で映っているであろうオブジェクト群 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 にフレーム番号と共に格納する。

次に、第 2 フレーム (時刻 t=) の処理がなされ、オブジェクト群 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 1 フレームの ObjectQueue のオブジェクトの処理が終わってれば、第 2 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共にそれらのタプルを更新する (フラグ Update=True)。D オブジェクトは新規なので新たにタプルを作成する (フラグ Update=Ture)。一方、一番目に抽出されたオブジェクト A は今回は抽出されなかった (フラグ Update=False), A のタプルは “ユニットとして完全なタプル” となり、フレーム開始番号、終了番号と共に 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 3 フレーム (時刻 t=2) の処理がなされ、オブジェクト群 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プルの処理中でなければ第 3 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共にそれらのタプルを更新する。A オブジェクトは t=0 の ObjectHolder にはタプルとして存在していたが、t= で削除されているため新規タ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 (フラグ Update=False) C, D のタプルは “ユニットとして完全なタプル” となる。その結果フレーム開始番号、終了番号と共に 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。

3.5 溢れによるリアルタイム性阻害要因の検出

3.1 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 3 の)。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 3 の)。ObjectHolder 中のタプルがなかなかユニットとして完全なタプルにならず、かつそのようなタ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よう。)

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ことを確認して、要因とはしなかった。

したがって、図 3 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 入力となる。

4. 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である。

- OS : Windows XP
(CPU: Mobil Intel Pentium4, RAM:256Mbyte)
- 描画ソフト :
ゼンリン社製 Zmap-Town
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 6 に見られるように被写体建物オブジェクトが抽出された (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は、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。しかしながら、オブジェクトが抽出されると同時に 図 7 に示す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 テーブルのタプル数は 188 であった。

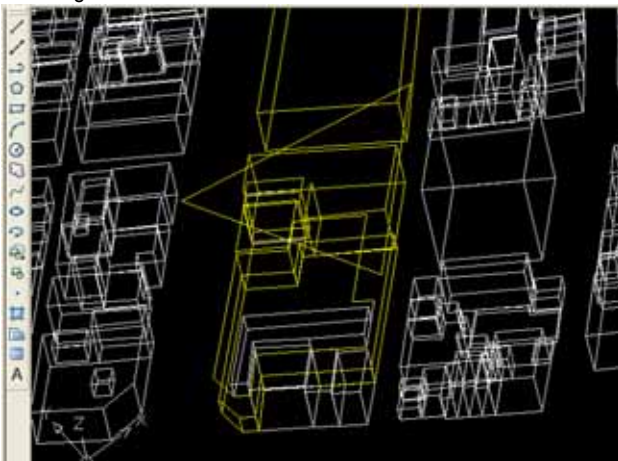


図6 3次元での被写体建物オブジェクト抽出
Fig6 Three Dimensional Building Object Extraction

| 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 |
| マンザビル | 66 | 145 | 80 | 9 |
| 玉屋A.Sビル | 67 | 167 | 101 | 10 |
| サエウサ本館 | 69 | 460 | 392 | 11 |
| 井上商会ビル | 71 | 76 | 6 | 12 |
| 燐五亭 | 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7 リアルタイム生成されたXBuildings テーブル
Fig7 XBuildings Table created in Real-Time

5. 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、レンズ画角データに加えて、撮影地点の 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にあたり、計算環境の変化に動的に対応して、映像にリアルタイムで建物オブジェクトを索引付けるアルゴリズムを提案し実装してその有効性を検証した。

今後の課題としては、実際に全てのモジュールを一台のモバイルコンピュータに組み込み、フィールドでの実験を行い、アルゴリズム、リアルタイム性の効率をより具体的に検証することがあげられる。他に、GPS やジャイロセンサなどの精度の検証と対策、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などが挙げられる。

末筆ながら、本研究を支援して下さった方々に深謝する。

【文献】

- [1] Vaughan, G., Smeaton, A., Gurrin, C., Lee, H., and 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, 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, and JianHua Wu: "Video Parsing, Retrieval and Browsing", Intelligent Multimedia Information Retrieval, ed. Mark T. Maybury, PP.139-158, MIT Press, Massachusetts, 1997.

佐藤 有紀子 Yukiko SATO

2005 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・情報科学専攻修了。映像データベースシステムの研究・開発に従事。日本データベース学会学生会員。

増永 良文 Yoshifumi MASUNAGA

お茶の水女子大学理学部情報科学科教授。1970 東北大学大学院工学研究科博士課程修了, 工学博士。データベースシステムの研究・開発に従事。情報処理学会および電子情報通信学会フェロー。日本データベース学会会長。著書に「リレーショナルデータベース入門[新訂版]」(サイエンス社)など。