

大規模映像ストリームのための映像情報統合基盤システムの提案

A Proposal of Video Stream Integration System for Massive Video Streams

渡辺 陽介[▼]秋山 亮[◆]大喜 恒甫[◆]北川 博之[▲]

Yousuke WATANABE

Ryo AKIYAMA

Kousuke OHKI

Hiroyuki KITAGAWA

近年、ネットワーク経由で利用可能なライブカメラが増加しており、これらから得られる映像データとセンサーやGPS等の様々な情報源のデータとの統合利用の需要が増加している。本研究では異種情報統合を目的とした映像情報統合基盤システムを提案する。このシステムはライブカメラを含む各種情報源に対するSQLライクな問合せインターフェースを提供する。映像ストリーム統合のために、映像データを扱う抽象データ型と、映像データからメタデータを抽出する関数を用いている。さらに本研究では、大規模なネットワークカメラ環境を対象とした動的情報源選択方式を提案する。本方式は、利用者の興味の移り変わりに応じて接続するカメラを逐次切り替えるといった処理を、問合せ言語レベルで実現するためのもので、移動オブジェクトに関する映像追跡のようなアプリケーションの開発を容易にする。

Since lots of live cameras are available in the Internet, integrating video data from these cameras and data from other information sources becomes more important. In this paper, we propose a video stream integration system for heterogeneous information sources. The system provides a SQL-like query interface for various sources including live cameras, sensors, GPS and databases. To integrate video streams, we employ an abstract data type to hold a sub-sequence of video frames and functions to extract metadata from video data. Beyond that, we also propose a dynamic source selection scheme for massive video streams. The scheme is used when information sources to be accessed may change according to changes in user interest. One of typical applications is tracking video data about some moving objects.

1. はじめに

近年のネットワーク技術の普及や、デバイス技術の発達に伴い、ライブカメラによるリアルタイム映像の配信および取得が容易になり、多方面で利用されるようになった。実際、

インターネット上でも、様々な地域の映像が配信されており、誰でも簡単に遠隔地の様子を見ることができるようになっている[11]。大量の映像データを同時に扱うシステムを、開発者が1から構築することは非常に開発コストが高い。そのため、リアルタイム映像処理専用の基盤ソフトウェア[10]が数多く提供されてきた。しかし、実世界から得られる情報は映像データだけではない。特に現在、センサーネットワークからのデータや、人や物などの位置情報データが容易に利用可能となってきた。時間とともに変化するデータを提供する、これらの情報源はストリーム[1], [2], [3], [4], [5]と呼ばれている。ストリーム型の情報源の多様化に伴い、これらを組み合わせたアプリケーション構築の需要が高まっている。

本研究では、実世界からの多数の映像データ、センサーデータおよび従来からのデータベースを統一的に扱う映像情報統合基盤システムを提案する。提案システムは、我々の研究グループで研究開発しているStreamSpinner[6], [7], [8]というストリーム処理エンジンをベースに開発されている。映像ストリーム統合の実現にあたって、提案システムでは次の3つの機能を新たに導入した。

まず1つ目として、映像ストリームと他の情報源とのデータモデルの違いを吸収するために、本研究では映像データをタプルに格納するための抽象データ型を新たに導入した。絶え間なく流れ続ける映像ストリームを一定時間ごとに区切り、このデータ型のオブジェクトとしてタプルに格納する。タプルへの変換により、映像ストリームと他の情報源との親和性が向上する。

2つ目は、映像からのメタデータ抽出機能である。一般に、映像などのマルチメディアデータに対して直接問合せの選択条件や結合条件を記述することは非常に困難であるので、実際はそれらに関するメタデータを問合せ記述の対象とすることになる。提案システムでは、多様な要求に対応するために、映像データ型に対するメタデータ抽出処理をユーザ定義関数として利用者が任意に追加できるようになっている。

3つ目は、大規模な映像ストリーム環境に対応するために導入した、情報源の動的選択機能である。多数の映像ストリームに対して同時に解析処理を行うには、メモリやCPU時間などの計算機資源が大量に必要となる。だが、特定人物の追跡監視を行うような問合せ処理の場合は、その人物の位置情報などの他の情報と組み合わせて参照することで、解析処理の対象となる情報源を大幅に絞り込む事ができる。従来、このような情報源の絞り込み処理は、アプリケーション開発者がプログラミング言語のレベルで個別に実装することで達成されてきたが、提案システムでは問合せ記述レベルでより簡潔に実現できるようになっている。

2. 映像情報統合基盤システム

2.1 システムアーキテクチャ

提案システムは、図1のように、メディアータと複数のラッパーから構成されている。

ラッパーはストリームデータを受け取るモジュールで、ストリームからデータが到着すると、メディアータに到着のイベントを通知すると共に、受け取ったデータをリレーションのタプルの形式に変換して、メディアータに送る役割を持つ。DBラッパーは、RDBMSに保管されたデータを取得するためのラッパーである。映像データを扱うためのカメララッパーとして、ローカルカメララッパーとネットワークカメララッパーがある。ローカルカメララッパーはUSBやIEEE1394で接続

[▼] 正会員 科学技術振興機構戦略的創造研究推進事業

watanabe@kde.cs.tsukuba.ac.jp

[◆] 学生会員 筑波大学システム情報工学研究科博士前期課程

akiyama_ohki@kde.cs.tsukuba.ac.jp

[▲] 正会員 筑波大学システム情報工学研究科、筑波大学計算科学研究センター

kitagawa@cs.tsukuba.ac.jp

されたカメラから映像を直接取得し、ネットワークカメララッパーはRTP (Real Time Transfer Protocol)のストリームとして提供されるライブカメラの映像を取得する。映像データもリレーションのタプル形式に変換される。

メディアータは、利用者から登録された問合せを評価するモジュールである。問合せ記述については次節で述べる。メディアータはラッパーから受け取ったデータに対して、問合せの内容に基づいて演算評価を行う。本研究ではリレーショナル代数における選択、射影、結合、直積、和の各演算と、ユーザ定義関数への呼び出し処理を考慮する。本研究で新たに導入した演算については3節で述べる。メディアータで生成された処理結果は、問合せを登録した相手に逐次配信される。

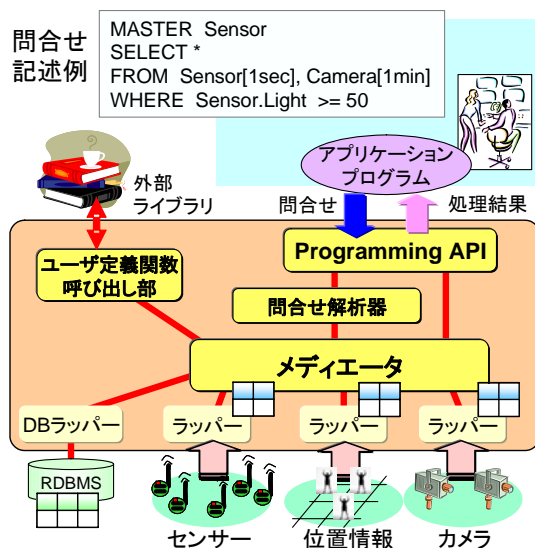


図1 映像情報統合基盤のシステム構成

Fig.1 System Architecture

2.2 問合せ記述

本システムにおける問合せ記述は、図1のようにSQLベースの文法に基づいている。MASTER節で指定したストリームからのデータ到着をきっかけとして、SELECT節以下に書かれた内容の処理を実行する。図中の記述例は、光センサーとカメラ映像を統合するもので、「光センサーの値が届いたときに、その値が50以上だった場合には、カメラからの映像を付加してその情報を知らせたい」という要求である。FROM節に書かれた[1sec]および[1min]は、演算評価時に処理対象とするデータを決める時間幅である。例えば[1min]は、演算評価時刻を基準として最新1分間のデータを処理対象にすることを指定したことになる。

2.3 映像データの取り扱い

本研究では、映像ストリームと他のセンサーやデータベースとの統合を目的としており、映像ストリームもカメララッパーにより共通データモデルであるリレーションに変換して扱われる。変換によって、映像データとしての利便性は若干低下するが、他の情報源との親和性は格段に向上する。

映像データをタプルに格納するために、本研究では映像専用のデータ型Videoを用いる。1つのVideo型には一定枚数のフレーム(静止画)が保持できる。カメララッパーは、カメ

ラから送られるフレームのシーケンスを一定枚数ごとに区切ってVideo型にまとめる処理を行う。最終的に映像ストリームは、Video型の属性とTimestampの属性の2つをもった無限のタプルを持つリレーションとして、利用者から見えるようになっている。

続いて、映像データに対する問合せ記述について述べる。映像データはバイナリデータであり、映像の内容に基づいたフィルタリング条件などを直接記述することは困難である。そこで、映像の内容に関する、数値やテキストによる情報(「誰が映っているか」「いつ、どこで撮影されたものか」等)を、メタデータとして抽出することが求められる。メタデータを抽出する映像解析手法はすでに様々提案され、ライブラリとして提供されてものが多いため[10]、本システムではユーザ定義関数呼び出しの機能を実装して、外部ライブラリとの連携ができるようにした。これにより、利用者が用意した映像中の動きの検出アルゴリズムや、顔認識アルゴリズムなどを処理要求記述中で用いることができる。

例えば図2は「カメラ映像に対して動き検出処理を実行し、その値が一定値以上であればデータベースへその映像を格納して欲しい」という要求である。ただし、detectMotion()は映像内の動きに応じた実数値を返すユーザ定義関数で、VideoArchiveはデータベース中のテーブルであるとする。

```
MASTER Camera
INSERT INTO VideoArchive
SELECT Camera.Timestamp, Camera.Video
FROM Camera[1min]
WHERE detectMotion(Camera.Video) >= 1.0
```

図2 映像データに対する問合せ例

Fig.2 A Query for Video Data

このように、映像ストリームをVideo オブジェクトに変換するカメララッパーと、映像解析処理と連携するためのユーザ定義関数の仕組みを用意したことで、簡単な問合せ記述によって、映像データとセンサーおよびデータベースを扱うアプリケーションが容易に構築できる。

3. 情報源の動的選択機構

ネットワーク上では、複数のライブカメラやセンサーなど、多数のストリーム型の情報源が利用可能である。しかし、多くのストリームを同時に扱うような処理は、相応の計算機資源を必要とする。例えば「市内のライブカメラのどれかに特定の人物が映ったら、その映像を配信して欲しい」という要求では、利用可能なすべての映像データに対する人物検出処理を行わなければならない。このような要求の実現には、複数台の計算機での分散処理が必須であり、非常に多くのコストがかかる。それを避ける1つの手段として、位置情報のような他の情報源からのデータを用いることが考えられる。例えばもとの要求を「特定人物の近くに設置されているライブカメラにその人物が映っていたら、その映像データを配信して欲しい」と置き換えることができれば、システムが接続しなければならないカメラは追跡対象の人物の周りだけであり、計算機資源の浪費を抑えられるはずである。ただし、追跡対象人物が移動した場合には、その移動に伴って接続するカメラを逐次切り替えていかなければならない。

本稿では、上記の例のように利用者の興味の時間的変化に

合わせて情報源を選択的に利用することを情報源の動的選択と呼ぶ。情報源の動的選択は従来からも行われてきたが、それはアプリケーション開発者がプログラミング言語のレベルで個別に実装することで達成されていた。本研究は、問合せ記述でこれを実現することで、アプリケーション開発者の負担を削減することを目的とする。提案システムの情報源動的選択機構は以下の2つの機能から成り立っている。

- ASSIGN 演算：入力データに応じて、使用する情報源を実行時に変えることができる演算（3.1 節）
- ラッパー管理機能：情報源への接続を行うラッパーを必要に応じて生成・削除する機能（3.2 節）

3.1 情報源動的選択のための演算

既存のリレーショナル代数演算だけでは、利用者の興味に応じてシステムが利用する情報源を動的に変えるような処理を実現することはできない。また、SQL ベースの記述方式では、FROM 節で使用する情報源を明示的に指定しなければならないため、この用途に使えない。そこで本研究では、情報源動的選択のために新たに ASSIGN 演算を提案し、問合せ記述においては ASSIGN 節を導入する。

ASSIGN 演算は、入力タプルの属性値に基づいて、取得するリレーシヨンの名前および属性を変化させる演算である。ASSIGN 演算の定義を以下に示す。

$$ASSIGN_{R.A_i, \{R.A_1, \dots, R.A_j\}}(R) \\ = \{r \times t \mid r \in R \wedge t \in \pi_{\{R.A_1, \dots, R.A_j\}}(relation(r.A_i))\}$$

ASSIGN 演算は、各入力タプル r に対して、属性 $r.A_i$ の値に一致する名前のリレーシヨン ($relation(r.A_i)$) を取得し、属性 $r.A_1, \dots, r.A_j$ の値に一致する属性名だけを残すように射影した後に、入力タプル r との直積を出力する。一致するリレーシヨン名や属性名が存在しなかった場合は、空集合を返すものとする。

問合せ記述中では、ASSIGN 節を用いて ASSIGN 演算の使用を宣言する。図 3 は ASSIGN 節の利用例である。

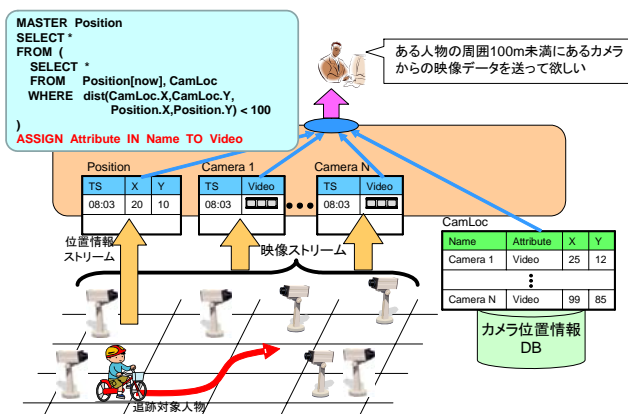


図 3 位置情報を用いた人物映像追跡

Fig. 3 Moving Object Tracking

図 3 の例では、N 台のカメラ、位置情報、カメラ位置 DB を統合して、「ある人物の周囲 100 メートル未満にあるカメラからの映像を送って欲しい」という要求を実現している。図 3 の問合せ記述は、まず内側の副問合せで、位置情報とカ

メラ位置 DB を統合し、追跡対象人物からの距離が 100 未満のカメラの名前を特定している。dist は 2 点間の距離を求めるユーザ定義関数である。外側の問合せでは、ASSIGN 節によって、Name 属性の値をリレーシヨン名、Attribute 属性の値を属性名とみなして、ASSIGN 演算を実行するように指定している。図 4 はこの要求を提案システムで処理した場合の処理例である。位置情報とカメラ位置 DB との結合処理によって、Camera1 が対象の近傍にあると判明した場合は、ASSIGN 演算が Camera1 リレーシヨンの Video 属性から映像データを取得する。

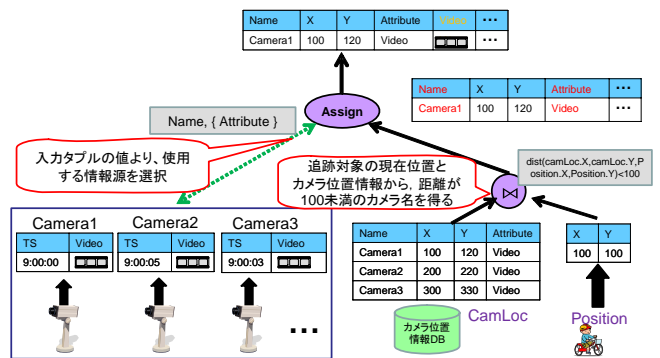


図 4 人物映像追跡における処理例

Fig. 4 Processing Example in Moving Object Tracking

3.2 ラッパー管理機能

提案システムは、ライブカメラ等の情報源への接続を行うラッパーを動的に起動・削除する機能を提供しており、利用者は問合せ記述中でユーザ定義関数として使用することができる。以下の2つの関数が利用可能である。

- create_wrapper(config): 指定されたラッパーを生成する関数である。引数にはラッパー生成のための設定情報（ラッパー名、接続先 URL）を渡す。すでに同じ名前のラッパーが存在する場合は何も行わない。
- delete_wrapper(name): 指定されたラッパーを削除する関数である。引数にラッパー名を渡す。対応する名前のラッパーが存在しない場合は何も行わない。

この機能を用いると、情報源にアクセスする必要がある間だけ、対応するラッパーを生成・利用し、不要になったらラッパーを削除する、といった処理をさせることができる。

例えば、図 3 の処理要求記述において、ASSIGN 演算が使用する可能性があるのは、追跡対象から 100 メートル未満の距離にあるカメラのみである。そこで、図 5 の処理要求のように記述して、追跡対象とカメラの距離が 100 メートル + α （図 5 では 200 メートル）を下回ったら、そのカメラに接続するカメララッパーを生成させるように指定できる。ただし、図 5 の ConfDB はラッパーを生成するために必要な設定情報が格納された DB とする。

```
MASTER Position
SELECT create_wrapper(ConfDB.Conf)
FROM Position[now], CamLoc, ConfDB
WHERE dist(CamLoc.X, CamLoc.Y,
           Position.X, Position.Y) < 200
AND CamLoc.Name = ConfDB.Name
```

図 5 ラッパー生成を行う問合せ例

Fig. 5 A Query Using create_wrapper()

ラッパーの削除についても図5とほぼ同様な記述で実現でき、提案システムでは適切な情報源に必要な間だけ接続することで計算機資源の浪費を抑えることが可能である。

4. プロトタイプシステム

我々が現在実装を行っているプロトタイプシステムについて述べる。開発プラットフォームはJDK1.6.0を使用している。ネットワークカメララッパーはQuickTime for Java[9]を用いて実装されている。映像からメタデータを抽出する関数もいくつか用意されており、映像からの動き検出などが可能である。この他にも、各種センサーや位置情報、データベースに対応したラッパーも実装されている。現在、問合せ解析器が一部未実装のため、ASSIGN節を用いたSQLベースの記述はできないが、構文解析済みの処理木(XML形式)を直接与えることでASSIGN演算が使用可能である。図6はプロトタイプシステムの実行画面である。図6中央の表データは問合せの処理結果で、画面下はASSIGN演算によって取得されたカメラからの映像である。



図6 プロトタイプシステムのスクリーンショット

Fig.6 A Screenshot of the Prototype System

5. まとめと今後の課題

本稿では我々の研究グループで開発中の映像情報統合基盤システムについて述べた。現在は、提案システムの残りの実装作業と評価実験のためのシミュレーション環境の準備を進めている。今後は評価実験を行って、情報源動的選択の有効性についての検討などをする予定である。

【謝辞】

本研究の一部は科学研究費補助金特定領域研究(#19024006)、科学研究費補助金基盤研究(A)(#18200005)、科学技術振興機構CREST「自律連合型基盤システムの構築」による。

【文献】

- [1] D. J. Abadi et al. "Aurora: a new model and architecture for data stream management," VLDB Journal vol.12, no.2, pp.120-139, 2003.
- [2] D. J. Abadi et al. "The Design of the Borealis Stream

Processing Engine," Proc. CIDR, 2005.

- [3] A. Arasu et al. "The CQL Continuous Query Language: Semantic Foundations and Query Execution," VLDB Journal, vol. 15, no.2, pp.121-142, 2006.
- [4] S. Chandrasekaran et al. "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World," Proc. CIDR, 2003.
- [5] R. Motwani et al. "Query Processing, Resource Management, and Approximation in a Data Stream Management System," Proc. CIDR, 2003.
- [6] 山田真一, 渡辺陽介, 北川博之, 天笠俊之, "データストリーム管理システムHarmonicaの設計と実装", 情報処理学会論文誌: データベース, Vol.48, No.SIG14 (TOD35), pp.91-106, 2007年.
- [7] Y. Watanabe et al. "Integrating a Stream Processing Engine and Databases for Persistent Streaming Data Management," Proc. DEXA, pp. 414-423, 2007.
- [8] StreamSpinner. <http://www.streamspinner.org/>
- [9] QuickTime for Java. <http://developer.apple.com/quicktime/qtjava/>
- [10] OpenCV. <http://sourceforge.net/projects/opencvlibrary/>
- [11] Fisgonia. <http://www.fisgonia.com/>

渡辺 陽介 Yousuke WATANABE

2001 筑波大学第三学群情報学類卒。2006 同大学院博士課程システム情報工学研究科修了。博士(工学)。現在は科学技術振興機構戦略的創造研究推進事業における研究員として、自律連合システムにおけるデータ・インターオペラビリティに関する研究活動に従事。情報処理学会、日本データベース学会、ACM 各会員。

秋山 亮 Ryo AKIYAMA

2006 筑波大学第三学群情報学類卒。現在、同大学院システム情報工学研究科博士前期課程に在学中。StreamSpinnerプロジェクトにおける研究・開発に従事。日本データベース学会学生会員。

大喜 恒甫 Kousuke OHKI

2007 早稲田大学人間科学部卒。現在、筑波大学大学院システム情報工学研究科博士前期課程に在学中。StreamSpinnerプロジェクトにおける研究・開発に従事。日本データベース学会学生会員。

北川 博之 Hiroyuki KITAGAWA

1978 年東京大学理学部物理学科卒業。1980 年同大学理学系研究科修士課程修了。日本電気(株)勤務の後、1988 年筑波大学電子・情報工学系講師。同助教授を経て、現在、筑波大学大学院システム情報工学研究科教授、ならびに計算科学研究センター教授。理学博士(東京大学)。異種情報源統合、XML とデータベース、データマイニング、センサーデータベース、WWW データ管理等の研究に従事。著書「データベースシステム」(昭晃堂), 「The Unnormalized Relational Data Model」(共著, Springer-Verlag)等。日本データベース学会理事、情報処理学会フェロー、電子情報通信学会フェロー、ACM, IEEE-CS, 日本ソフトウェア科学会、各会員。