

映像ストリームとタプルストリームの統合利用のためのモデルの提案

高橋 翼[†] 川島 英之^{†‡} 北川 博之^{†‡}筑波大学 大学院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[‡]

1. はじめに

近年、ネットワーク環境の発達やカメラの低価格化などにより映像ストリームの配信、取得、蓄積が容易となってきた。そのため、小規模なものではペットの留守番監視システムや家の防犯システム、大規模なものでは、ロンドンの監視カメラネットワークである Ring of Steel などが実際に利用されている。

映像ストリームの普及に伴い、その処理要求も複雑化している。例えば、ある特定の人物が映っている時のみ映像を配信してほしいといった要求がある。このように、監視システムにおいて利用者の求めている情報は膨大な映像データの中のほんの一部であることが多い。そのため、映像ストリームの配信、取得、蓄積は選択的に行われることが望まれる。映像ストリームを選択的に処理するためには映像のメタデータが必要となる。また、各種センサデータとの異種統合利用により更に高度な処理が可能となる。

メタデータや各種センサデータは一般的にタプル形式で表現される。そのため、上記の映像ストリームの選択的な配信、取得、蓄積を行うためには、映像ストリームとタプルストリームを統合的に扱える必要がある。

映像ストリームとタプルストリームの統合利用が可能なシステムに、Ramesh Jain らの MedSMan[1]や我々の研究室で研究開発を行ってきた StreamSpinner[3, 4]がある。両システムともに映像ストリームのフレームを一定枚数ごとにタプルに格納することによって、映像ストリームをタプルストリームに変換し統合を実現している。

しかし、映像ストリームとタプルストリームはデータの扱いが異なるため、映像配信サーバやストリーム処理エンジンなどの異なる処理系が存在する。従って、映像ストリームとタプルストリームの統合には、両処理系を統合的に扱える新しい処理系が必要となる。そこで本論文では両ストリームを統合的に扱える、関係演算に整合する演算体系を提案する。提案演算体系では映像ストリ

ームをタプルストリームに変換せず、両ストリームを同等に扱うことにより効率的に統合を行う。また、映像ストリームのメタデータにストリーミングタグ[2]を利用することにより効率的な問合せ処理を目指す。

2. 映像ストリームの効率的な統合手法

従来研究では映像ストリームをタプルストリームに変換して問合せ処理を行っていた。しかし、選択演算や結合演算などのリレーショナル演算はメタデータを対象として行われるため、問合せ処理において映像データはタプルに格納されている必要が無い。

そのため、本研究では論理的には映像ストリームをタプルストリームとして扱うが、物理的には映像ストリームとタプルストリームを別々に扱う。映像データを一時的に保持し、アプリケーションが必要としたタイミングで保持している映像データを配信することができれば、実装レベルでの映像ストリームからタプルストリームへの変換やタプルストリームから映像ストリームへの変換といった処理を省ける。

2.1. 映像ストリームの論理モデル

本研究では論理的に映像ストリームをタプルストリームとして扱う。映像タプルはストリーム情報源の識別子、タプルの識別子、一定枚数のフレーム、そしてタイムスタンプを属性値として持つ。

2.2. ストリーミングタグ

問合せ処理を行うためには、映像データに対するメタデータがタプルに付与されている必要がある。本研究ではメタデータにストリーミングタグを用いる。ストリーミングタグはタプルストリームに埋め込まれるタプル形式のメタデータであり、特定のタプルや属性値を識別することが可能である。コンテンツという属性を持ち、そこに「誰が映っているか」、「何を行っているか」等といった情報を格納することができる。

映像タプルに新たなメタデータ属性を追加することで問合せ処理を行うことも可能だが、タグを用いる利点には以下のものがある。複数のタプルや属性等にタグを付与できるため、メモリ使用量を削減できる、タグ選択演算やタグ結合演算など、タグ指向の演算を利用すること

A model for integrating media streams and tuple streams
Tsubasa Takahashi[†], Hideyuki Kawashima^{†‡}, Hiroyuki Kitagawa^{†‡},
[†] Graduate School of Systems and Information Engineering,
University of Tsukuba
[‡] Center of Computer Sciences, University of Tsukuba

ができる、また様々な利用者がタグを付与できるため複数の利用者間で情報共有が行える。

3. 演算体系の提案

物理的な処理において映像ストリームとタプルストリームを別々に扱うためには、タプルデータに対する演算体系、映像データに対する演算体系、そして 2 つのデータを連結する演算体系を考える必要がある。タプルデータに対してはリレーショナル演算体系が存在するが、映像データと 2 つのデータを連結するための演算体系は新たに考案する必要がある。そのため、本研究ではこれらに対する演算体系を提案する。

3.1. 提案演算

映像データに対する演算体系に必要な基本的な機能として、映像データの保持、映像データの取得が考えられる。そのため、これらに必要な処理を行うための演算を 2 つ提案する。提案演算を以下に示す。

Keep 演算：

映像データを受け取り保持する。

Fetch 演算：

Keep 演算によって保持された映像データを取得する。

映像データとタプルデータを連結する演算体系に必要な機能としては、映像データからのメタデータの抽出、メタデータのタグへの変換が挙げられる。これらの処理を行うために 1 つの演算を提案する。提案演算を以下に示す。

Meta Data Extract 演算：

映像ストリームからメタデータを抽出しタグに変換する。

3.2. 問合せ処理の例

ここでは、映像データを自律的に取得するプル型アプリケーションに対する問合せ処理の例を示す。

プル型アプリケーションに対する処理例を図 1 に示す。映像ストリーム情報源から映像データを Meta Data Extract 演算子が受け取り、メタデータと映像データに切り分け、それぞれ処理木と Keep 演算子に渡す。Keep 演算子は Meta Data Extract 演算子から送られてきた映像データを一時的に保持する。アプリケーションが処理木の結果を受け取り、映像データを必要としたときに Fetch 演算子を利用し、処理木の結果から必要な映像データを取得することができる。

このようなプル型アプリケーションの例として、監視カメラによる不審者検出システムが挙げられる。映像データに写った人物の顔認識を行い、人物データベースと照合することによって、特定の人物がカメラに写ったと

きにアプリケーションに通知を出し、映像データを取得することができる。

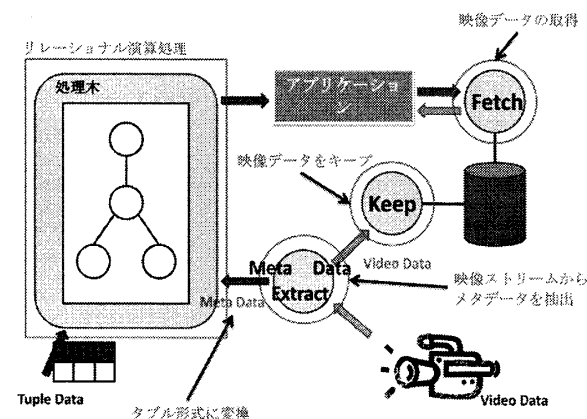


図 1 プル型アプリケーションに対する処理例

4. 終わりに

本論文では映像ストリームとタプルストリームの統合のための新たな演算体系を提案した。提案演算体系を用いることで、映像データをその利便性を保持したまま利用することが可能となる。そして、映像データの処理に必要な Keep 演算、Fetch 演算、映像データとタプルデータの連結処理に必要な Meta Data Extract 演算を提案した。

今後の課題としては、提案演算体系をベースとしたストリーム処理システムの構築が挙げられる。システム構築のためには、Keep 演算における映像データの保持方法、問合せ記述方式などについて考える必要がある。Keep 演算に関してはどの程度一時保持を行うか、来歴をどのように管理するかといったことを検討する予定である。

謝辞

本研究の一部は、科学研究費補助金基盤研究(A) (#21240005)、科学研究費補助金若手研究(B) (#20700078)による。

参考文献

- [1] Bin Liu, Amarnath Gupta, Ramesh Jain: MedSMAN: a live multimedia stream querying system. Multimedia Tools Appl. 38(2): 209-232, 2008.
- [2] Rimma V. Nehme, Elke A. Rundensteiner, Elisa Bertino: Tagging Stream Data for Rich Real-Time Services. PVLDB 2(1): 73-84, 2009.
- [3] 大喜恒甫, 渡辺陽介, 北川博之, 川島英之, 「対象情報源を動的に選択可能なストリーム処理機能の実装と評価」情報処理学会論文誌：データベース, Vol.2, No.3, pp.1-17, 2009年9月.
- [4] Yousuke Watanabe, Hiroyuki Kitagawa, "Query Result Caching for Multiple Event-driven Continuous Queries", Information Systems 2009.