

## 2009 年度下期未踏 IT 人材発掘・育成事業 採択案件評価書

### 1. 担当PM

首藤 一幸 PM(東京工業大学大学院情報理工学研究科 数理・計算科学専攻准教授)

### 2. 採択者氏名

チーフクリエイター: 塩川 浩昭(筑波大学大学院 システム情報工学研究科)  
コクリエイター: なし

### 3. プロジェクト管理組織

株式会社創夢

### 4. 委託金支払額

3,000,000 円

### 5. テーマ名

監視用映像ストリーム処理基盤 Eagle Eye の開発

### 6. 関連Webサイト

なし

### 7. テーマ概要

近年、映像ストリームは研究室を出て実社会で用いられる例が顕在化している。特に従来は商店や銀行などの限られた場所でのみ設置されていた監視カメラは、防犯意識の高まりから市街や道路などで多く見られるようになった。このような監視カメラ

を用いた監視システムは、プライバシー侵害という難しい問題も孕むが、システムとして実社会に存在し、さらに拡大していることから、技術として社会に求められていることは疑いの余地がない。

ところが、監視システムは多数の監視カメラから時々刻々と生じる映像ストリームに対して、人間が映像を目視することで異常検出を行っている。そのためこのような監視システムは、「人的なコストが高い」、「見落としにより正確な異常検知ができない」などの問題を有する。これらの問題に対する解として、画像処理やパターン認識処理により異常検出を行う監視システムが用いられることがあるが、このようなシステムの実装には、複雑なコーディングが不可欠であるため、その開発コストは大きなものになると考えられる。さらには、監視システムは導入環境により監視カメラ台数やネットワーク構成などが異なるため、特定環境に特化した専用システムを作りこむ必要がある。ゆえにシステムの汎用性は低く、監視カメラ台数の増加といった特定環境から少しでも異なる環境においては、監視システムが停止してしまう可能性がある。このような場合、システムを再開発することになり、開発コストがさらに膨れ上がるという問題が生じる。

本提案では、監視システム開発のための汎用映像ストリーム処理基盤 Eagle Eye を開発する。Eagle Eye ではストリーム処理エンジンの技術を応用し、SQL ライクの問合せを記述による画像処理・パターン認識を実現する。

## 8. 採択理由

多数の映像ストリームを処理(例:〇〇さんが現れた)する基盤ソフトウェアを開発して、使えることを実証するという提案である。映像ストリームに対する処理(問い合わせ)を宣言的に書けることや、分散処理で処理性能を稼げる点が特徴である。

基盤ソフトウェアの開発自体は大学の研究室のバックアップで強力に推進されている。未踏ユースでの取り組みは、本当に使えること、様々な応用できること、様々な映像処理を簡単に試せることなどを示すこととなるだろうか。

売り込み先を意識してか、応用例として人の監視を挙げているが、監視にこだわらず、明るい気持ちになれる応用も模索して欲しい。天体観測、動物観測、交通流調査、スポーツの試合解析、また、人の動きを撮影するにしても、身長推定、男女推定...など、前向きな応用が考えられるだろう。また、ネット越しにある程度の映像ストリームを集められる現状を踏まえると、カメラを自分たちで設置することにもこだわらなくてよいのではないか。日本ではまだマイナーな(?)ストリーム処理の価値を世に知らしめて欲しい。

## 9. 開発目標

本プロジェクトでは、監視システム開発のための映像ストリーム処理基盤 Eagle Eye の開発を目標とした。Eagle Eye の利用により、大規模なカメラプログラム環境におけるシステム実装のコスト軽減を目指す。

Eagle Eye ではストリーム処理エンジンの技術を応用し、宣言的な問合せによる処理の記述や、画像処理組み込み関数、パタン窓及び分散高信頼化機能の開発を行った。また、OpenCV を用いたプログラミングとの比較や、実運用実験を通して、開発したシステムの有用性が確認された。

## 10. 進捗概要

当初の提案通り、以下を達成した：

- 宣言的な問い合わせの記述による、画像ストリームに対する問い合わせ処理。
- 具体的な画像処理ルーチンいくつかの実装。
- 分散システムとしての構成。

これによって性能・耐故障性を向上させた。

- カメラ 20 台(実機 10 台、仮想カメラ 10 台)による実運用実験。

また、ごく簡単な問合せ記述によって顔認識や移動物体認識などが可能であることを示すデモンストレーションが可能となっている。今後のアピールへの活用が期待できる。

## 11. 成果

Eagle Eye は、大量に流れ来る映像ストリームを効率的に処理するために、ストリーム処理エンジンをコアシステムとする。ストリーム処理エンジンとは、時々刻々と到着するストリームデータに対して、宣言的な SQL ライクの問合せ言語を用いて関係演算処理を行う基盤システムである。本プロジェクトでは、コアシステムであるストリーム処理エンジンを開発し、さらに、画像処理・パタン認識処理機能、時系列パタン照合処理、分散機能を拡張する。これにより、Eagle Eye の利用者は SQL ライクな問合せを記述するだけで、監視システムに必要な画像処理やパタン認識を実現できる。

まず拡張の 1 つ目として、ストリーム処理エンジンに配信された映像ストリームデータから特徴量を抽出するために、映像ストリームデータに対して画像処理を行う画像処理組込関数を導入した。本プロジェクトでは、画像処理ライブラリの標準である OpenCV が有する機能に則り、基本的な画像処理関数を選択し、画像処理組込関

数として実装を行う。この関数を問合せで記述することにより、ストリーム処理エンジン内部で画像処理が自動実行され、特徴量抽出や画像変換を行う。

2 つ目の拡張は、パタン窓の導入である。一般に映像ストリームデータを継続的に監視する場合、映像ストリームデータの持つ瞬間的な特徴量だけではなく、特徴量が時系列的にどのような変化をしているかということに注目することが少なくない。本プロジェクトでは、映像ストリームデータの示す時系列パターンを検出可能にするための仕組みとして、パタン窓の導入を行う。パタン窓の導入により、ユーザーは検出したい時系列パターンをシステムに登録・検出することが可能になる。

3 つ目の拡張は、分散機能の導入である。扱うカメラの台数が増加した際や、離れた場所に設置されたカメラを利用する際、システムの負荷分散を実現したい際には、システム内部で実行される処理を別々のマシンに分散配置し処理性能を稼ぐ必要がある。本プロジェクトでは、ストリーム処理エンジンの従来研究に基づき、分散環境において Eagle Eye を動作可能にする分散化機能の導入を行う。分散化機能の導入により、システムのスケーラビリティの確保やシステムが動作するマシンの故障に備えた高信頼化処理等が実現可能になる。

Eagle Eye のシステムアーキテクチャを図 1 に示す。Eagle Eye では、デバイス毎の差異を吸収するアダプタにより各カメラを接続する。アダプタでは、カメラから映像ストリームデータを取得する毎に、受信した映像ストリームデータをタプル形式に変換し、Eagle Eye のコアシステム部へと受け渡す役割を持つ。

アプリケーションからは、Eagle Eye の提供する API を介して SQL ベースの問合せを登録する。登録された問合せは問合せ解析器により、ストリーム処理エンジンが有する基本演算子(関係演算子)、画像処理組込関数、パタン窓から構成される処理木へと変換され、コアシステム部へと登録される。コアシステム部に処理木が登録された後、Eagle Eye はアダプタを経由して各カメラから映像ストリームデータを受け取り、登録された処理木に合わせた処理を実行する。最終的に得られた処理結果は Eagle Eye の提供する API を介してアプリケーションへと出力される。

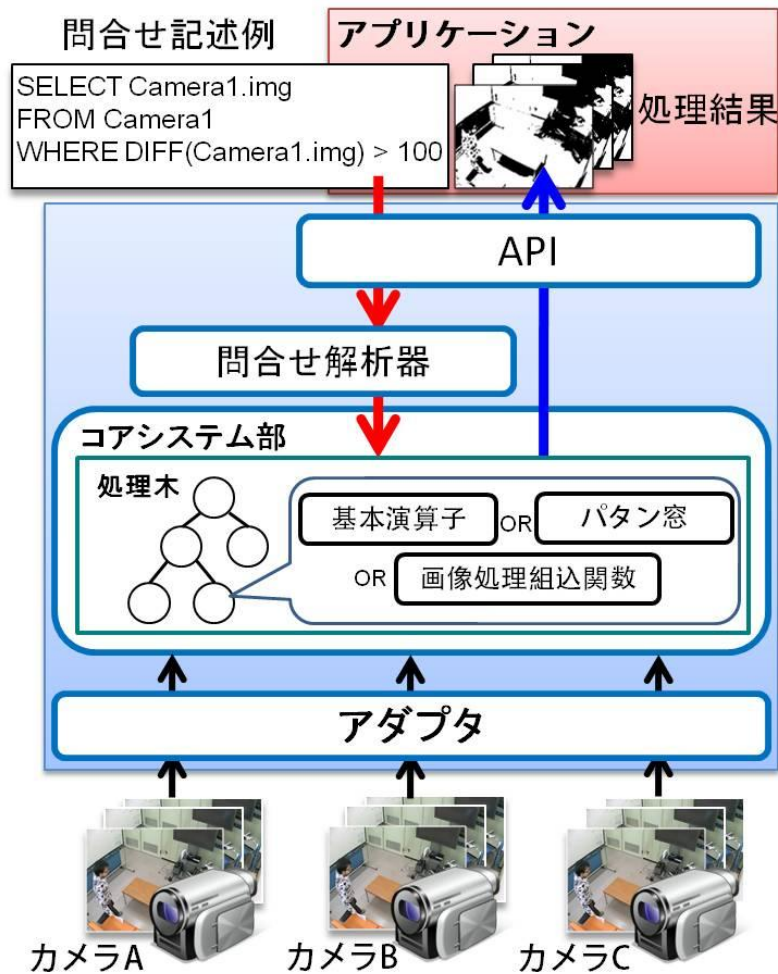


図 1: Eagle Eye のシステムアーキテクチャ

## ■コアシステム

Eagle Eye のコアシステムとなるストリーム処理エンジンとして、スタンフォード大学が研究開発する STREAM に準ずるストリーム処理エンジンを実装する。本プロジェクトにおけるコアシステムとは、図 1 におけるアダプタ、問合せ解析器、コアシステム部に当たる。それぞれの開発詳細について以下に述べる。なお、コアシステムの実装には Java 言語を用いた。

### ◆アダプタ

アダプタは、Eagle Eye に接続されるカメラデバイスのデバイス毎の差異を吸収するために用意された機構である。具体的な処理として、アダプタはまず、映像取得メソッドによりカメラデバイスから映像を取得する。そして、取得した映像ストリームデータを数フレームずつ切り出し、タプルと呼ばれる Eagle Eye の内

部データ型へと データの変換を行う(図 2)。このデータ型は Camera、id、timestamp、Image という属性を持つ。それぞれの属性は表 1 の意味を持つ。ここで生成されたタプルは順次コアシステム部へと送信される。

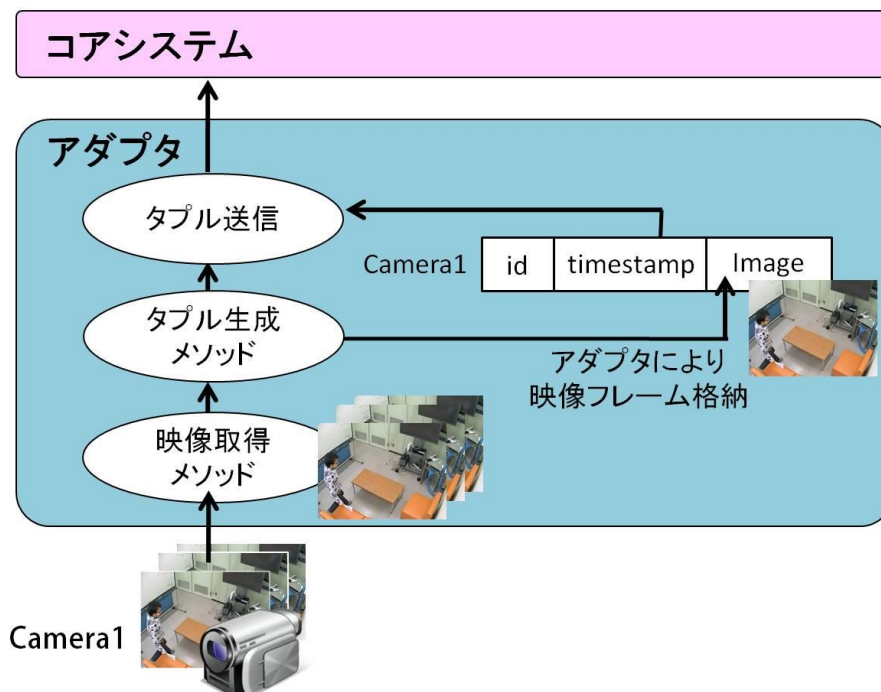


図2: 映像ストリームデータの変換

表1: タプルの属性一覧

属性名	意味
Camera	カメラデバイスの識別番号
id	カメラ毎に生成されたタプルの通し番号
timestamp	タプルが生成された時間
Image	取得した映像フレーム

ユーザーは Eagle Eye の提供するアダプタクラスを継承し、オリジナルの映像取得メソッドをコーディングすることにより、任意のカメラデバイスを Eagle Eye に接続することが可能である。

アダプタの起動のために、ユーザーには事前にアダプタに関する設定ファイルを用意してもらう必要がある。設定ファイルには、実行するアダプタ名、情報源のアドレス、および、そのアダプタから映像ストリームデータが入力された際に、他の映像ストリームデータと識別するための識別名を記述する。ここで記述した識別名は、表 1 の Camera に反映される。

#### ◆問合せ解析器

問合せ解析器では、ユーザーにより登録された SQL ライクの問合せ記述をシステム内部での処理実行順序を表す処理木への変換を行う。処理木への変換は問合せが登録される度に自動で実行され、変換された処理木はコアシステム部へと登録され、映像ストリームデータが到着する毎に処理木に合わせた処理がコアシステム部にて実行される。この問合せ解析器にて生成される処理木は、従来のストリーム処理エンジンと同様にコアシステム部が提供する基本演算子(関係演算子)及び、画像処理組込関数、パタン窓から構成される。Eagle Eye が提供する基本演算子(関係演算子)の説明は事項にて説明する。

#### ◆コアシステム部

コアシステム部では、問合せ解析器によって生成された処理木に基づいて、アダプタから配信される映像ストリームデータの処理を実行する。コアシステム部では、従来のストリーム処理エンジンと同様に基本演算子として関係演算子を有している。Eagle Eye の有する関係演算子とその処理動作例を以下に示す。

#### ◇関係演算子 1: 選択

選択演算子は、処理木に入力された映像ストリームデータのフィルタリングを実行する演算子である。選択演算子を実行する際の問合せ記述及び実行した際の動作例を図 3 に示す。

#### 問合せ記述例

```
SELECT camera1          # 出力属性の指定
FROM camera1            # 処理対象情報源の指定
WHERE camera1.id > 10  # 選択演算実行条件の指定
```

#### 処理木

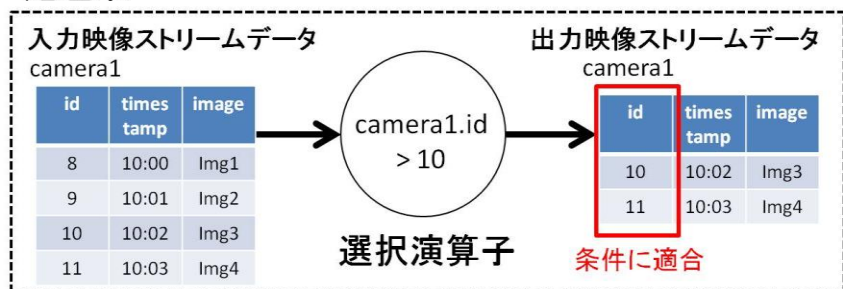


図 3: 選択演算子の実行



選択演算子では、図 3 の上部に示されたような問合せ記述例において、WHERE 節(赤字で示された部分)の記述を追加することにより、問合せ解析器により処理され、図 3 下部のような選択演算子を持つ処理木を実行することができる。WHERE 節には、実際に選択演算によってフィルタリングを行う際に、フィルタリング条件として使用したい状況を指定する。図 3 の例では、camera1 から配信される映像ストリームデータの持つ id が 10 より大きい場合のみ選択してくるという条件を指定している。この例では、条件選択のために二項演算子である「>」を利用しているが、これ以外にも、同じく大小関係を比較する「>=」、「<=」、「<」や等値な関係を求める「=」などが記述・実行可能になっている。

◇関係演算子 2:射影

射影演算子は、処理木に入力された映像ストリームデータのタプルから指定した属性のみを抽出する演算子である。選択演算子を実行する際の問合せ記述及び実行した際の動作例を図 4 に示す。

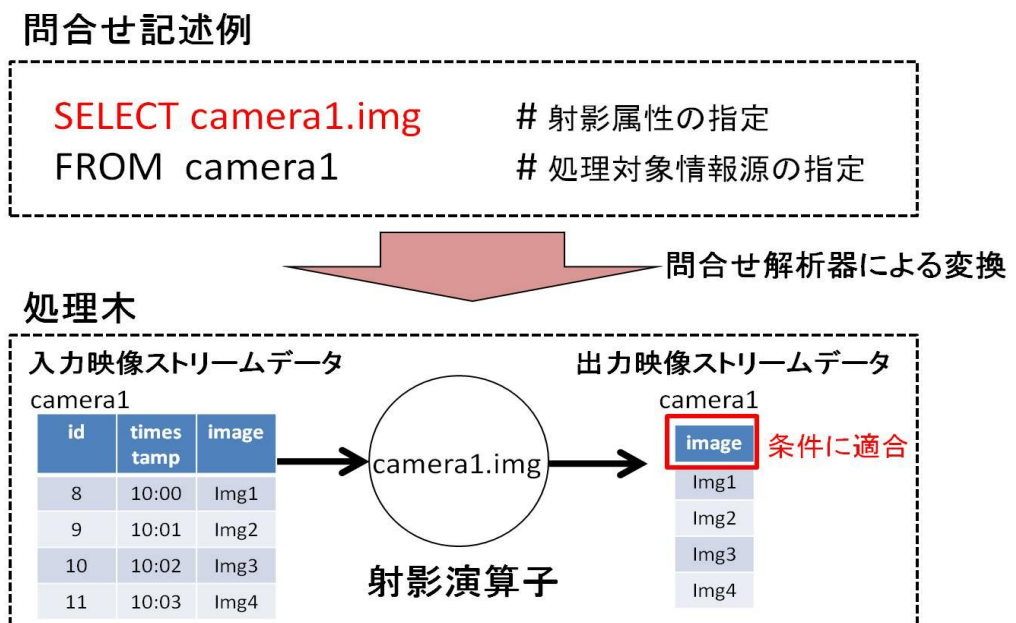


図 4:射影演算子の実行

射影演算子では、図 4 の上部に示されたような問合せ記述例において、SELECT 節(赤字で示された部分)の記述をすることにより、問合せ解析器にて処理され、図 4 下部のような射影演算子を持つ処理木を実行することができる。SELECT 節には、実際に射影演算によって属性抽出をおこない



たいストリーム名と属性を指定する。図 4 の例では、camera1 から配信される映像ストリームデータの持つ属性の内、Image 属性のみを抽出する処理条件を指定している。

#### ◇関係演算子 3:和

和演算子とは、処理木に入力された複数本の映像ストリームデータを 1 本の映像ストリームデータへと束ねる処理をする演算子である。和演算子を実行する際の問合せ記述及び実行した際の動作例を図 5 に示す。

#### 問合せ記述例

```
SELECT camera1          # 射影属性の指定
FROM camera1, camera2 # 処理対象情報源の指定
```

問合せ解析器による変換

#### 処理木

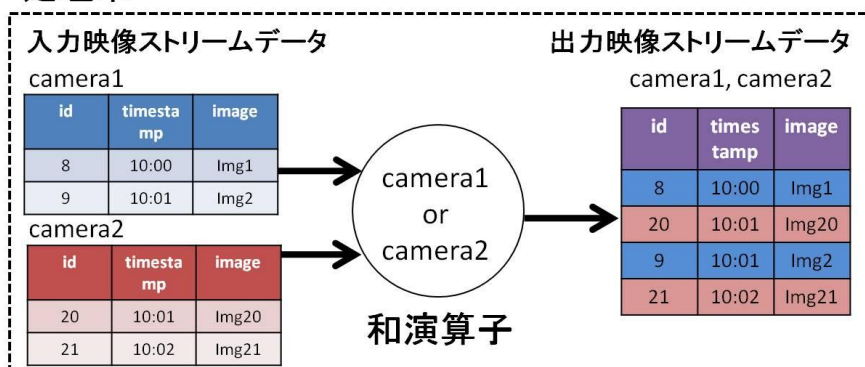


図 5: 和演算子の実行

和演算子では、図 5 の上部に示されたような問合せ記述例において、FROM 節(赤字で示された部分)のように情報源を並べて記述することにより、問合せ解析器により処理され、図 5 下部のような和演算子を持つ処理木を実行することができる。FROM 節には、実際に和演算によって束ねたい映像ストリームデータ情報源を指定する。図 5 の例では、camera1 と camera2 から配信される映像ストリームデータを 1 つの映像ストリームデータとして束ね上げる処理を示している。またコアシステム部では、Eagle Eye に接続されている全てのカメラを利用する際の all-cameras と言った略記法も有している。

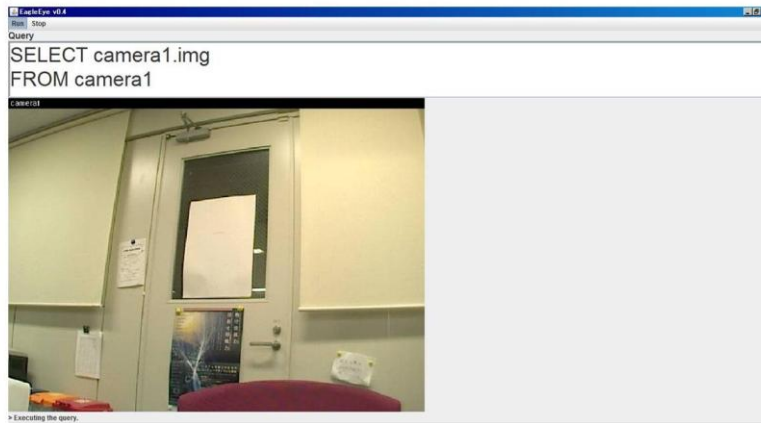
◆問合せの記述、実行例

コアシステムによる問合せの記述、実行例を図 6 に示す。

問合せ記述例

```
SELECT camera1.img      # 出力の指定  
FROM camera1           # 処理対象情報源の指定
```

実行例



問合せ記述例

```
SELECT all-cameras.img  # 出力の指定  
FROM all-cameras       # 処理対象情報源の指定
```

実行例

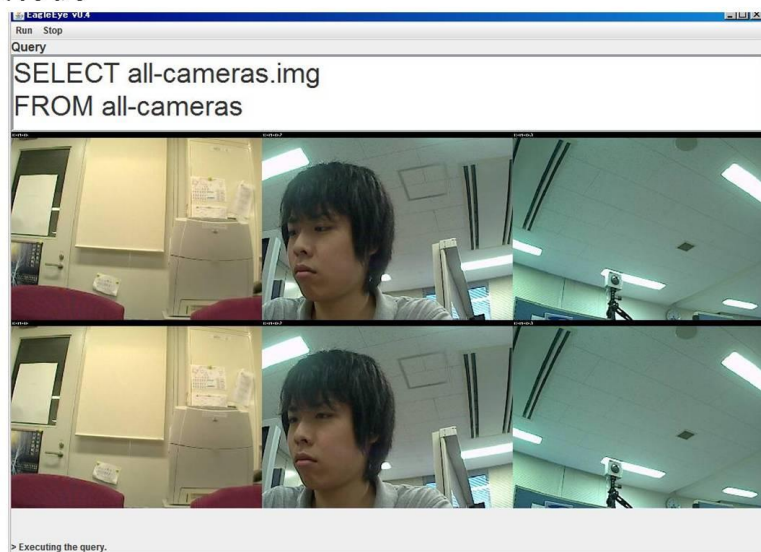


図 6: 問合せ記述例、実行例

図 6 に示したように、SQL ライクの問合せを記述することにより、複数カメラにまたがるような映像ストリームデータの取捨選択が容易に実行可能になっている。

Eagle Eye では、本節で説明した各機能を有するコアシステムを軸とし、映像ストリームデータに対する処理を実現している。

コアシステムに対する拡張である以下については、Eagle Eye についての他の発表資料を参照のこと。

- 画像処理組み込み関数
- パタン窓
- 分散高信頼化機能

## 12. プロジェクト評価

クリエイターは、構想自体が野心的だった、高度な分散システムを作り上げてくれた。開発や実験は、クリエイターの腕や意欲を鑑みるに、もっと進めてくれると期待していたが、期待を上回るころまでは至らなかった。大学院生としての研究発表実績が素晴らしく積み上がったが、そのために、プロジェクトの開発が割を食ったのではないだろうか。仕方がないことであるが、研究との両立について考えさせられる。

成果報告会では、大変コンパクトな宣言的記述で高度な映像処理が可能であることを示してくれた。見せ方や、映像処理の楽しい内容はデモの効果を大変高めたと感じている。クリエイター自身も、他のクリエイターへうまくアピールできたと感じており、成果報告会やその準備で当人が学んだことは大きかったのではないだろうか。

成果物には、次の社会的効果が期待できる：

- ・データストリーム処理自体の認知向上
- ・データストリーム処理の応用可能性、実用可能性の提示

これだけでも、未踏ユースの成果としては非常に高いコストパフォーマンスが得られると考えている。

一方で、確たる根拠があるわけではないが、ソフトウェア自体の開発継続、発展性はあまり望めないのではないかと踏んでいる。その理由は、クリエイター自身が、来年度は企業研究所に就職する予定で、そこで本テーマを継続できる見込みはそれほど高くないことである。クリエイターの所属研究室で関連研究テーマを継続するとしても、今回の成果物が引き継がれる可能性は高くないのではないだろうか。（これは PM の予想であり、そうなると限った話ではない。）

### 13. 今後の課題

クリエイター自身がEagle Eyeに取り組めるうち(~2011年3月?)に、最大限のパフォーマンスを発揮して欲しい。発表、展示、デモなどの機会を最大限にとらえて、アピールを続けて欲しい。