Vol.2013-HPC-139 No.15
2013/5/30

# 多数の動画像を対象とするリアルタイム異常値検出の検討

小川 宏高[1]　中田 秀基[1]　工藤 知宏[1]

**概要**：我々は，オンライン機械学習向け分散処理フレームワーク Jubatus を基盤として用いた，大量センサデータに対するリアルタイムかつ複雑な解析を実現する処理エンジンの構築を目指している．特にさまざまな応用分野へのシステムの適用を想定した場合，多種多様なリアルタイムデータを処理対象として取り扱えることが重要である．そのなかでも映像や音声に代表されるメディアデータは，汎用性が高く，実世界へのセンサー装置の浸透が顕著に進んでおり，内包している情報量の多さから高い利用価値が期待される．本稿では，Jubatus を基盤として実際に多数の動画像を対象としたリアルタイム異常値検出を行うシステムを構築し，その構成概要を示した．また，性能特性の調査を行い，その結果を示した．その結果，学習データに基づいた異常値検知はリアルタイムに実現できたが，学習フェーズは学習データの増加とともにリアルタイムに処理することが困難になった．レスポンス時間についてより詳細な調査を行い，LSH から取得した擬似近傍点の個数が多い場合に著しい性能劣化が見られることが判明した．

# Preliminary Study on Real-time Anomaly Detection from Multiple Video Streams

Hirotaka Ogawa[1]　Hidemoto Nakada[1]　Tomohiro Kudoh[1]

***Abstract:*** We aim to build a real-time and complex data analisys engine for large-scale sensor data, based on a distributed online machine-learing framework, Jubatus. In order to adapt this engine to various application areas, it is crucial that we can handle a wide variety of real-time data. Especially, multimedia data, including video and audio, are general-purpose and feature rich, and their sensors have already been penerated into the real world deeply and widely. Hence, we expect that they are quite valuable for various applications. In this paper, we realize an actual system that provides real-time anomaly detection for multiple video streams and describe the overview of our system. And, we also investigate the performance characteristics of the system. As a result, anomaly dectection based on learnt video frames can be performed at real-time, but learning phase can hardly be processed at real-time according to the growth of learnt video frames. We conduct more detailed investigation into the response time of the system, and clarify that peformance degradation is observed mostly when the amount of pseudo-neighbors extracted from LSH is relatively large.

## 1. はじめに

近年，モバイルネットワーク技術の普及と各種センサ技術の発展に伴い，多種多様なモノがネットワークに接続され，現実世界のさまざまな事象を情報技術の世界から捉えることが可能になってきた．こうした多種多様大量な情報（ビッグデータ）を対象とした技術としては，従来より，CEP（Complex Event Processing）に代表されるリアルタイムに取得されたデータに対するストリーム処理を実現する技術，MapReduce に代表される蓄積された大容量データに対するバッチ処理を実現する技術がそれぞれ開発されてきており，マーケットでも急速に受け入れられつつある．

その一方で，これらの特性を同時に満たす，すなわちリアルタイムに取得されたデータに対する複雑な解析処理を実現する技術は未踏の領域であると言ってよい．MapReduce に代表されるバッチ処理システムではリアルタイム性を満足することは困難である．また，CEP に代表されるリアルタイム処理システムは，リアルタイム性と高速性を両立するために，センサデータに対して行う処理を単純なルール

[1] 独立行政法人 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)

ベースのマッチングなどに限ったり，処理が動作するサーバ上のメインメモリに載るように，参照する履歴データを限定するなどしている．そのため，低遅延な処理が可能である一方で，複雑な解析や広範な過去データにアクセスすることができないという問題がある．

こうした問題を解決するため，我々は，複雑な解析処理を高速に実行可能なリアルタイム解析エンジンと，解析エンジンが必要する大量の過去データを迅速に供給可能なストリーム支援高速分散データストアによって，この問題を解決することを目指している [1]．

このうち，本稿が対象とする前者のリアルタイム解析エンジンでは，オンライン機械学習向け分散処理フレームワーク Jubatus[2] を基盤として用いて，大量センサデータに対するリアルタイムかつ複雑な解析を実現する処理エンジンの構築を目指している．特にさまざまな応用分野へのシステムの適用を想定した場合，多種多様なリアルタイムデータを処理対象として取り扱えることが重要である．そのなかでも映像や音声に代表されるメディアデータは，汎用性が高く，実世界へのセンサー装置の浸透が顕著に進んでおり，内包している情報量の多さから高い利用価値が期待される．その一方で，データ量または扱う特徴データ量自体が多いため，解析エンジンの処理全体のリアルタイム性が達成できるかどうか疑問が残る．

そこで，我々は Jubatus を基盤として実際に多数の動画像を対象としたリアルタイム異常値検出を行うシステムを構築し，性能特性の調査を行った．本稿ではシステムの構成概要と評価結果を示す．

## 2. 多数の動画像を対象としたリアルタイム異常値検出システム

本節では，多数の動画像を対象としたリアルタイム異常値検出を実現するシステムの概要について述べる．

### 2.1 Jubatus

Jubatus[2] は，オンライン機械学習器（多値分類，線形回帰，統計分析，推薦，グラフマイニング，外れ値検知），特徴ベクトルコンバータ（データの前処理と特徴抽出）を提供する，耐故障性を持つ分散機械学習のためのフレームワークである．Preferred Infrastructure と NTT SIC によって共同開発され，オープンソースソフトウェアとして公開されている．

さまざまな応用分野への Jubatus の適用を想定した場合，多種多様なリアルタイムデータを処理対象として取り扱えること，応用分野に依存した特徴抽出を簡便に実現できること，が重要である．これらを実現するため，我々は，従来入力値として数値ないし文字列しか扱えなかった Jubatus の特徴ベクトルコンバータを拡張し，画像，動画を含むマルチメディアデータを扱えるようにするとともに，OpenCV ライブラリの提供する特徴抽出器ならびに CHLAC 特徴抽出器を利用する特徴抽出モジュールを設計・実装している．また，外部サーバに対して RPC 経由で特徴抽出処理を要求するための RPC サーバ，RPC クライアント（特徴抽出）モジュールも設計・実装している．後者は多言語による特徴抽出処理の記述を可能にし，サードパーティの開発者が簡便に特徴抽出を実現できるようにすることを目的としている[*1]．

動画像を対象としたリアルタイム異常値検出を実現するためには, Jubatus の提供する外れ値検知器 jubaanomaly を利用する．jubaanomaly では，Local Outlier Factor（LOF），N 次元空間で近傍の点がどの程度あるかを検査することで外れ値を検出する手法，を利用した外れ値検知をサポートしている．jubaanomaly は，クライアントから特徴ベクタを `add` すると特徴点の追加を行うとともにその点の異常度を算出して返し，`calc_score` することで特徴点の異常度を算出して返すことができる．

### 2.2 CHLAC 特徴

HLAC（Higher-order Local Auto-Correlation）特徴 [3] は，画像平面の 2 次元の局所領域における相関パターン（局所マスク）の出現を全体で数え上げることで特徴の算出を行うものである．HLAC 特徴の性質としては，画像全体からの特徴を記述することが可能であり，画像全体から相関を取るという性質から位置の影響を受けない．2 次元の局所領域を用いる HLAC 特徴に対して，動画像の 3 次元の局所領域を用いる特徴量が CHLAC（Cubic Higher-order Local Auto-Correlation）特徴 [4] である．3 次元を画像平面の 2 次元と時間軸の 1 次元として用いることで時間方向の相関パターンの情報も得られる．CHLAC では独立な局所マスクのパターンは 251 通りとなることから，二値動画像の各フレームごとに 251 次元の特徴ベクタを算出することになる．

CHLAC 特徴の応用例としては，動作認識や異常検出に関する研究がすでに複数行われている．前者の具体例としては，[5]，[6]，[7] など，後者の例としては [8]，[9] などがある．

本稿の主眼は，多数の動画像ソースから同時並行的に算出される CHLAC 特徴を対象として，Local Outlier Factor（LOF）を利用した外れ値検出を行うことで，外れ値処理自体のスループットの最大化を目指すというものであり，上記の既存研究とは目的や実現方法を異にする．

### 2.3 システムの概要

システムは，**図 1** に示す処理フローで異常値検出を行う．

正常画像の学習フェーズの手順は以下の通りである．

[*1] これらの詳細については別の機会に公表する予定である．

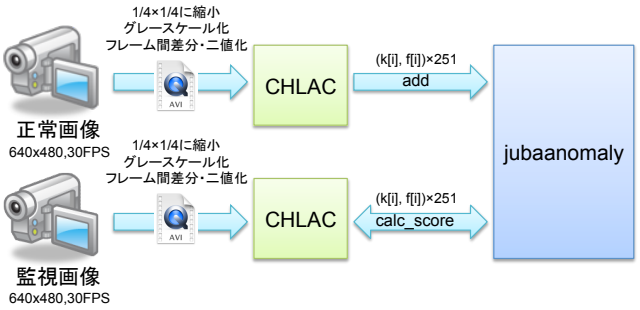


**図 1** 異常値検知処理フロー

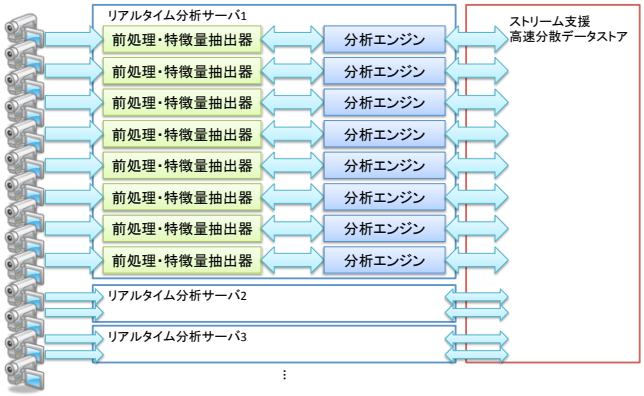


**図 2** 異常値検知処理の全体

(1) ビデオカメラから 640×480，30FPS の動画像を取得
(2) 1/4 × 1/4 に縮小，グレースケール化
(3) フレーム間差分と大津の自動しきい値選定法による二値化
(4) 二値差分画像から CHLAC 特徴を算出
(5) 特徴ベクタを引数に jubaanomaly に `add` して，特徴ベクタの追加を行うとともにその異常度を算出

監視画像の検知フェーズの手順もほぼ同様で以下の通りである．

(1) ビデオカメラから 640×480，30FPS の動画像を取得
(2) 1/4 × 1/4 に縮小，グレースケール化
(3) フレーム間差分と大津の自動しきい値選定法による二値化
(4) 二値差分画像から CHLAC 特徴を算出
(5) 特徴ベクタを引数に jubaanomaly に `calc_score` して，特徴ベクタの異常度を算出

システム全体では，多数の動画像ストリームを同時並列に検査するため，クラスタの各ノードで図 1 の構成を複数並列に動作させている（**図 2**）．この場合，多数のビデオカメラをエミュレートするために，ローカルストレージに格納されたキャプチャ済み動画像を用いている．

また，jubatus の特徴として，複数の外れ値検知器の学習結果を定期的に同期する mix という機能があるが，本稿の時点では複数クライアントから利用する際にデッドロックを起こす問題が存在する（パッチは送付済み）ため，再現性のある評価に十分なコンフィギュレーションとは言えない．したがって，本稿の範囲では，各動画像ストリームについて正常画像を独立に学習し，監視画像も独立に検知するものとする．Jubatus の分散機能を用いた評価結果は別の機会に公表する予定である．

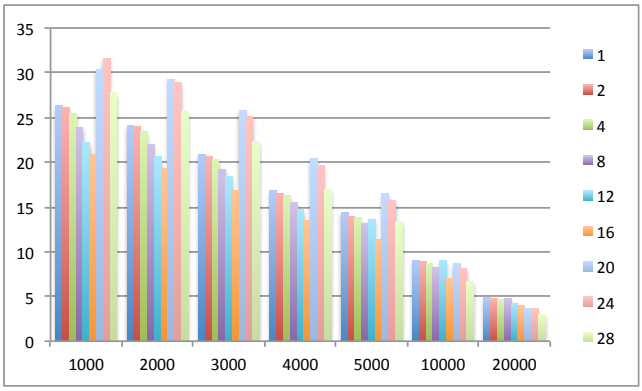


**図 3** Throughput: jubaanomaly learning phase

## 3. 評価

評価環境としては，Intel(R) Xeon(R) CPU E5-2620 2.00GHz × 2CPU，メインメモリ 32GB のサーバ 8 台からなる小規模クラスタを用いた．各 CPU のコア数は 6，スレッド数は 12 である．サーバのオペレーティングシステムとしては CentOS 6.4 x86_64，Jubatus は version 0.4.3 を用いた．Jubatus の外れ値検知器 jubaanomaly では，バックエンドの近傍探索アルゴリズムとして `euclid_lsh` を用いている．

### 3.1 スループット

ノードあたり 1–28 個の 640×480，30FPS の動画像ストリームの処理性能を調査した．

**図 3** は，正常画像の学習フェーズで，学習データのフレーム数を 1,000，2,000，...，20,000 個と増加させていったときの，一秒あたりの平均処理フレーム数をグラフ化したものである．図 3 は，一秒あたりのノードの総処理フレーム数を示したものである．

ほぼすべてのケースで元データのフレームレート（30FPS）を下回る結果となった．仮に正常画像データを半分のフレームレートに間引いて学習するものとしても，学習フレーム数が 6,000 を超えると間に合わなくなり始める．また，ノードあたりのスレッド数（24）を超える並列度ではスループットが低下し始めることが分かる．

**図 5** は，それぞれ 1,000，2,000，...，20,000 個の正常画像を学習させた状態で，監視画像の検知フェーズの一秒あたりの平均処理フレーム数をグラフ化したものである．**図 6** は，一秒あたりのノードのそう処理フレーム数を示したものである．

学習済みデータが 10,000 フレーム以下であれば，元データのフレームレート（30FPS）を上回る処理性能を示すこ

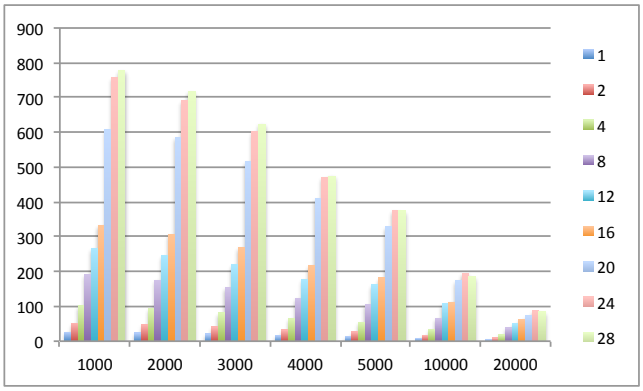


**図 4** Throughput: jubaanomaly learning phase (total)

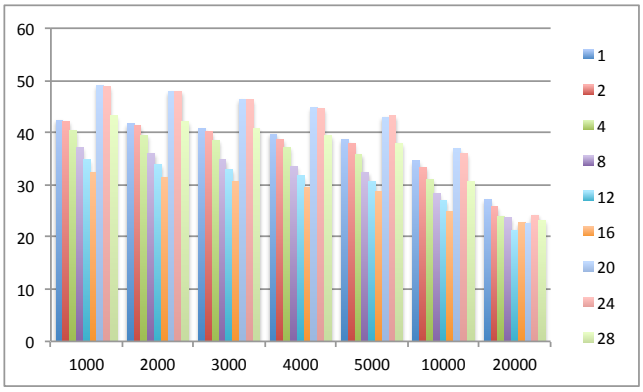


**図 5** Throughput: jubaanomaly detection phase

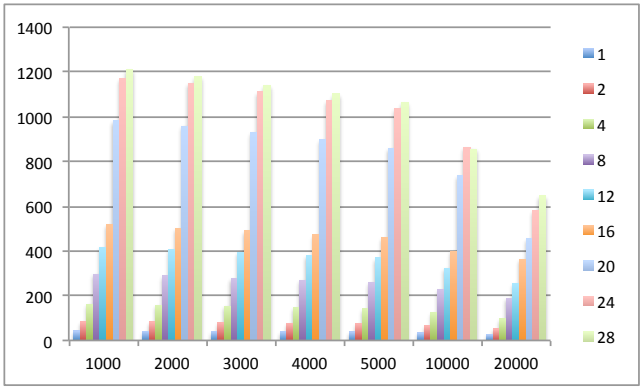


**図 6** Throughput: jubaanomaly detection phase (total)

とが分かる．また，特徴点数の増加につれスループットの低下が見られるが，学習フェーズでの低下と比較するとごく緩やかである．

学習フェーズでも検知フェーズでも 20 並列度を超えると，性能が向上している．これはターボ・ブースト機構が有効になったためではないかと推定される．

また，いずれの並列度でも（学習フレーム数が 1,000 の場合に）検知スループットが 30FPS を超えていることから，ローカルストレージの I/O，特徴抽出の処理がボトルネックとなることはないことが分かる．また，学習フレーム数 1,000，並列数 28 のとき，トータルで秒間 1,210 フレーム処理しているが，これはほぼ 120Mbps に相当するスループットであり，現代のコモディティハードウェアで十分賄える値である．

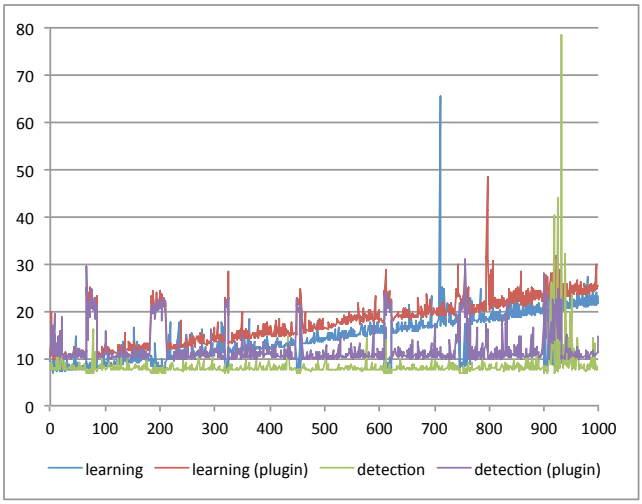


**図 7** Response Time

### 3.2 レスポンス時間

学習フェーズならびに検知フェーズのレスポンス時間を学習データが少ない領域で調査した（**図 7**）．

この調査では，学習データが空の状態から 1,000 フレームに達するまで，1 フレームずつ追加するのに要するレスポンス時間（learning，learning(plugin)），1,000 フレーム追加した状態で 1 フレームずつ検知するのに要するレスポンス時間（detection，detection(plugin)）をミリ秒で示している．このうち，learning, detection は，クライアント側で CHLAC 特徴抽出を行うもの，learning(plugin), detection(plugin) は，クライアント側はフレームデータをサーバに送信してサーバ側で特徴抽出を行ったものである．

図 7 からは，検知フェーズではほぼレスポンス時間が横ばいであるのに対して，学習フェーズではほぼ蓄積したデータ数に比例したレスポンス時間がかかることが分かる．3.1 の結果で学習フレーム数の増加に伴い，平均処理フレーム数が大幅に劣化するもの同じ理由である．

また，learning(plugin)，detection(plugin) がそれぞれ learning，detection に対して 1〜2 ミリ秒遅い結果が得られている．これは，フレームデータをクライアント・サーバ間で送受信するコストと推定されるが，実用上問題のないオーバーヘッドと言える．

### 3.3 レスポンス時間に関する考察

Jubatus の LOF の実装は，以下のような内部処理を行っている．

- `nn_engine->decode_row()`
  既に登録された列かどうかの判定
- `nn_engine->update_row()`
  追加する点を LSH（Locality Sensitive Hash）に格納

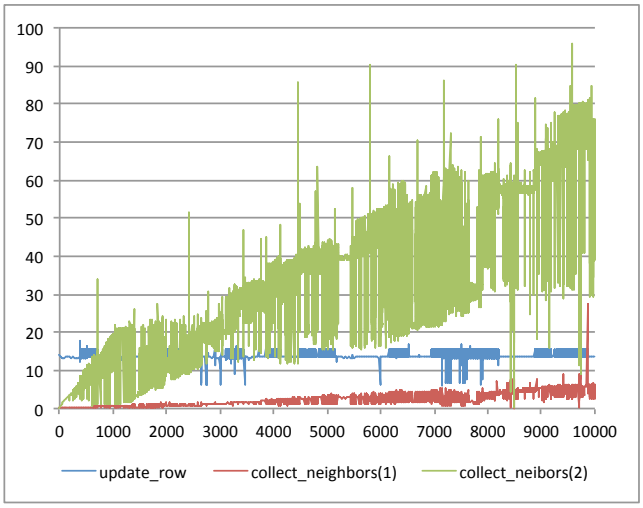


**図 8** Response Time (Breakdown)

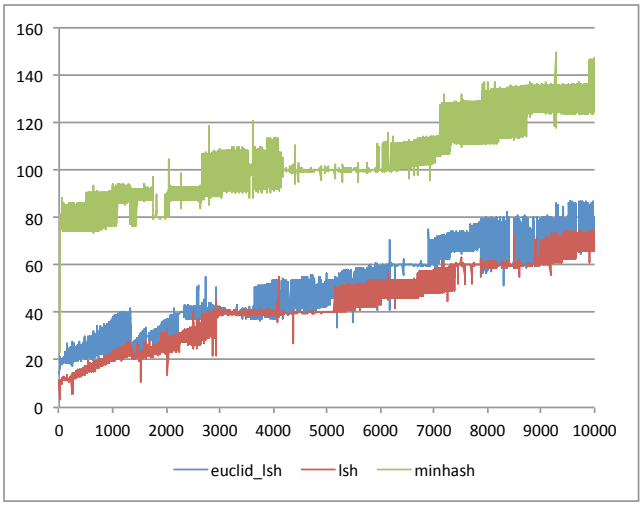


**図 9** Response Times for three backend algorithms

| bin_width | 総時間（秒） |
|---|---|
| 1 | 211 |
| 10 | 363 |
| 50 | 506 |
| 100（デフォルト） | 502 |

**表 1** euclid_lsh において bin_width を変更した場合の性能

- `collect_neighbors(1)`
  LSH から疑似近傍点の取得（対象点の近傍点）
- `collect_neighbors(2)`
  LSH から疑似近傍点の取得（対象点の近傍点の近傍点）
- `update_kdist_with_neighbors()`
  k-distance の計算
- `update_lrd_with_neighbors()`
  lrd の計算

これらのうち，`update_row()`，`collect_neighbors()` 以外は，ごく短時間（100usec 未満）で終わる処理なので，残りの処理について 10,000 フレーム学習する際の 1 フレームごとの所要時間のブレークダウンを**図 8** に示す．横軸の単位はミリ秒である．

`update_row()` はほぼ一定であるのに対し，`collect_neighbors()` は学習フレーム数の増加に伴って増加する．特に `collect_neighbors(2)` の占める割合は大きい．

擬似近傍点を $n$ 個取得する処理は，LSH の中で擬似近傍点の候補をハッシュ値から求め，それらを対象点との距離でソートし，近い順から $n$ 個を選択して返すものである．擬似近傍点の候補の数を確認したところ，実際に 10,000 フレーム分追加した時点で 7,041 フレームに達した．これは，動きの少ない動画像の場合，多くのフレームが同一のハッシュ値を取るためと考えられる．

この対策として，ハッシュの量子化幅（`bin_width`）を小さくとるというチューニングが可能になっている．**表 1** に示す通り，`bin_width` を 10 程度まで小さくとれば処理時間は 30%弱削減できる．ただし，量子化幅を小さくすれば，LSH において擬似近傍とみなされる点の個数が減少して計算の精度は低下するというトレードオフがある．また，処理時間の 30%弱の削減は，スループットを十分改善するには不十分でもある．

また，Jubatus の LOF では，バックエンドの近傍探索アルゴリズムを以下の中から選ぶことができる：

- `euclid_lsh`（デフォルト）
- `lsh`
- `minhash`
- `inverted_index`

これらのうち，`inverted_index` はアイテム数が増加すると極めて時間がかかるため除外して，残りの 3 つについて同様にレスポンス時間を測定したものを**図 9** に示す．

`euclid_lsh` に比べて，`lsh` の方が速いが，`bin_width` のチューニングによって逆転し得る範囲である．

## 4. まとめ

我々は，オンライン機械学習向け分散処理フレームワーク Jubatus を基盤として用いた，大量センサデータに対するリアルタイムかつ複雑な解析を実現する処理エンジンの構築を目指している．特にさまざまな応用分野へのシステムの適用を想定した場合，多種多様なリアルタイムデータを処理対象として取り扱えることが重要である．そのなかでも映像や音声に代表されるメディアデータは，汎用性が高く，実世界へのセンサー装置の浸透が顕著に進んでおり，内包している情報量の多さから高い利用価値が期待される．

本稿では，Jubatus を基盤として実際に多数の動画像を対象としたリアルタイム異常値検出を行うシステムを構築し，その構成概要を示した．また，性能特性の調査を行い，その結果を示した．その結果，学習データに基づいた異常値検知はリアルタイムに実現できたが，学習フェーズは学

習データの増加とともにリアルタイムに処理することが困難になった．レスポンス時間についてより詳細な調査を行い，LSH から取得した擬似近傍点の個数が多い場合に著しい性能劣化が見られることが判明した．

今後の課題としては，まず性能向上のための諸手法の試行がある．LSH に格納されたデータを忘却させるアルゴリズムの設計・実装，LSH に格納されたデータを縮約する手法の考案が必要である．また，今回実装の問題から実施できなかった Jubatus の分散フレームワークを用いた評価，また動画像データのみならず音声データを用いたリアルタイム異常値検出への拡張も今後の課題である．

## 謝辞

本研究の一部は，独立行政法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務「IT 融合による新社会システムの開発・実証プロジェクト」の成果を活用している．

### 参考文献

[1] NEDO: IT 融合による新社会システムの開発・実証プロジェクト，http://www.nedo.go.jp/activities/ZZJP_100048.html.
[2] PFI and NTT: Jubatus : Distributed Online Machine Learning Framework, http://jubat.us/.
[3] Otsu, N. and Kurita, T.: A new scheme for practical, flexible and inteligent vision systems, *IAPR Workshop on Computer Vision*, pp. 431–435 (1998).
[4] Kobayashi, T. and Otsu, N.: Action and Simultaneous Multiple-Person Identification Using Cubic Higher Order Local Auto-Correlation, *Int. Conf. on Pattern Recognition*, Vol. 4, pp. 741–744 (2004).
[5] 白木孝義，石黒勝彦， 深野亮，鴨志田良和，白井達也，斉藤秀雄，田浦健次郎，大武美保，佐藤知正，大津展之：CHLAC 特徴と Grid コンピューティングを併用したリアルタイム動作認識，信学技報 PRMU，Vol. 105, No. 615, pp. 97–12 (2006).
[6] 森下雄介， 小林匠，森崎巧一，大津展之：時間重みと外的規準を用いた動作評価手法，*CVIM*, Vol. 2008, No. 27, pp. 47–52 (2008).
[7] 佐藤竜太，亀田能成，大田友一：CHLAC 特徴量と部分空間法による複数行動の分類，画像の認識・理解シンポジウム 2009 論文集 (MIRU2009)，pp. 1344–1349 (2009).
[8] 南里卓也，大津展之：複数人動画像からの異常動作検出，*CVIM*, Vol. 2004, No. 91, pp. 173–186 (2004).
[9] 村井泰裕，藤吉弘亘，数井誠人：時空間特徴に基づくエスカレータシーンにおける人の異常行動検知，信学技報 PRMU，Vol. 2008, No. 82, pp. 247–254 (2008).