学術・技術論文

Partly Locality Sensitive Hashing を用いた時系列データからの高頻度パターン抽出

小川原　光　一* 田　邉　康　史* 倉　爪　　亮* 長谷川　　勉*

Detecting Frequent Patterns in Time Series Data using Partly Locality Sensitive Hashing

Koichi Ogawara*, Yasufumi Tanabe*, Ryo Kurazume* *and* Tsutomu Hasegawa*

Frequent patterns in time series data are useful clues to learn previously unknown events in an unsupervised way. In this paper, we propose a method for detecting frequent patterns in long time series data efficiently.

The major contribution of the paper is two-fold: (1) Partly Locality Sensitive Hashing (PLSH) is proposed to find frequent patterns efficiently and (2) the problem of finding consecutive time frames that have a large number of frequent patterns is formulated as a combinatorial optimization problem which is solved via Dynamic Programming (DP) in polynomial time $O(N^{1+1/\alpha})$ thanks to PLSH where N is the total amount of data. The proposed method was evaluated by detecting frequent whole body motions in a video sequence as well as by detecting frequent everyday manipulation tasks in motion capture data.

Key Words: Frequent Pattern Mining, Approximate Nearest Neighbor Search, Unsupervised Learning, Video Analysis

1. は　じ　め　に

ロボット技術を生活空間に積極的に導入して生活支援などに活用していくための取り組みが注目を浴びているが，そのためには周囲の人間の活動を認識するための技術が欠かせない．人間の活動の中でも高次の行動を認識する方法として，想定するタスクにおいて必要十分な行動認識器を人間が事前に設計して用いる方法 [1]〜[3] が従来提案されてきたが，生活空間における人間の活動は多様であり，タスクを限定しない場合にはこれらを網羅する行動認識器を事前に用意することは困難である．

そのため，生活空間を対象とするシステムは，新規の行動に対する認識器を逐次的に自動獲得する仕組みを持つことが望ましい．このとき，観測データ中に何度も現れる運動パターンはタスクにとって重要な意味を持つ可能性が高く，行動認識器のための学習データとして，もしくは行動文脈の学習や行動予測のための手がかりとして有用であると考えられる．そこで，本稿では上記仕組みのための基礎技術として，タスクに関する事前知識なしに観測データから頻出する未知の運動パターンを効率よく抽出する方法を提案する．

提案手法の主な特長は，(1) 局所性を保持するハッシュ関数と局所性を保持しないハッシュ関数を組み合わせた Partly Locality Sensitive Hashing (PLSH) を提案し，近似最近傍探索の枠組みによって高頻度パターンを効率よく探索する点と，(2) 非線形伸縮しかつ種類の異なる高頻度パターンの抽出問題を組み合わせ最適化問題として定式化し，動的計画法を用いて全体として $O(N^{1+1/\alpha})$ の計算量で解く点の 2 点である．

以降では，まず 2 章で関連研究について述べ，3 章で提案手法の概要を述べる．次に，4 章で PLSH について説明し，5 章で時系列データから高頻度パターンを抽出する方法について説明する．最後に，6 章で提案手法の評価実験を行い，7 章で本論文をまとめる．

2. 関　連　研　究

データ列の中から既知パターンの出現個所を効率よく求める方法については多くの先行研究があるが [4] [5]，本研究では Fig.1 に示すように未知の高頻度パターンの抽出を目的とする．未知の高頻度パターンの抽出については，これまで生物情報学 [6],

Fig. 1　Frequent patterns in time series data

原稿受付
*九州大学
*Kyushu University

データマイニング [7]〜[9]，動画解析 [10] [11]，運動解析 [12]〜[14] などの分野で精力的に取り組まれてきた．

生物情報学の分野では，塩基配列を 4 種類の塩基から構成される離散値データ列とみなし，例えば Staden らによって長さが既知の塩基配列すべての組み合わせに対して投票を行うことにより計算量 $O(N)$ で未知の高頻度パターンを抽出する手法が提案された [6]．

一方，連続値データ列に対する解析はデータマイニングや運動解析の分野で盛んに取り組まれてきた．パターン長が既知の場合は全探索の計算量が $O(N^2)$ となるため，平均計算量を下げるアルゴリズムが数多く提案されてきた．Lin らは，連続値データを離散化し，ハッシュ関数を用いた投票によって未知の高頻度パターンを効率よく抽出する手法を提案した [7]．Mueen らは，パターン同士の類似度に応じて入力データ列を並び替えることにより，連続値データのまま最も類似したパターン組を正確かつ非常に効率よく求める手法を提案した [9]．

パターン長が未知の場合には，動的計画法（ＤＰマッチング）に基づく多くの手法が提案されてきた．内田らは，論理判定型ＤＰマッチングを用いて，1 つのデータ列中に複数回出現するパターンを計算量 $O(N^2)$ で抽出する手法を提案した [13]．平均計算量を $O(N^2)$ 未満に下げるアルゴリズムとして，Yankov らはパターン長を離散的に一様伸縮し，Lin らの手法 [7] を拡張して伸縮する高頻度パターンを抽出する手法を提案した [8]．Meng らは，Locality Sensitive Hashing (LSH) [15] を用いて時刻ごとに類似データを探索しこれらを接続することによって，モーションキャプチャデータから非線形伸縮する高頻度パターンを計算量 $O(N^{1+1/\alpha})$ で抽出する手法を提案した [14]．しかし，データ数 N の増加に伴いハッシュバケット内のデータ数も増えるため，バケットの大きさが固定値の場合には実際の計算量は $O(N^2)$ に近くなる．

本稿では，主にこの文献 [14] の問題点を解決する手法を提案する．文献 [14] の手法では時系列データの各データ点ごとに LSH を用いた近似最近傍探索を行っている．しかし，対象が連続した時系列データである場合には，時間方向に近いデータ点における探索結果を用いて現データ点の探索結果を補うことが可能である．そこで提案手法では，各データ点においては限定された近似最近傍探索を行い，近いデータ点同士でこの限定された近似最近傍探索の探索範囲が互いに独立になるように探索空間を構築することによって，計算時間の観点から効率のよい高頻度パターン探索を実現する．

そのために，本稿では近似最近傍探索法の一種である Partly Locality Sensitive Hashing (PLSH) を提案し，LSH を利用した近似最近傍探索法と比較してより少ない計算時間で高頻度パターンが抽出できることを示す．また，提案手法では高頻度パターン抽出を組み合わせ最適化問題として定式化し動的計画法を用いて解くことによって，全体の計算量を文献 [14] と同様に $O(N^{1+1/\alpha})$ に抑えた．ただし，$\alpha(>1)$ は PLSH のパラメータによって決定される定数である．

高頻度パターンの抽出は，高頻度パターンに低ビットのコードを割り付けるデータ圧縮問題として考えることもできる．Zhao らは，最小記述長 (Minimum Description Length) 基準に基づき符合化データ列と辞書の大きさの和を最小化することによって，舞踏の運動計測データの分節化を行った [12]．しかし，低頻度パターンが支配的である通常のデータ列に対しては，このような方法は適当ではない．

3. 提案手法の概要

Fig.1 に示すように，本研究では長時間の時系列データ (d 次元) が与えられたときに，そこから高頻度で出現する類似した未知パターンを抽出することを目的とする．パターンとは時系列データの部分データ列を指し，パターン同士でパターン長や対応するデータ値の差が小さいものを類似パターンとする．

Fig.2 に，時系列データを d 次元から 2 次元に投影して表示した例を示す．もし時刻 t のデータ点 $o(t)$ がある高頻度パターン群に属している場合，このデータ点の近傍に他の類似パターンも存在することになる．つまり，あるデータ点の近傍に他の多くのパターンが存在するということは，そのデータ点が高頻度パターン群に属しているための必要条件となる．そのため，近傍パターン数の多いデータ点が連続する区間は高頻度パターンのよい候補になると考えられる．

そこで,「近傍」を d 次元空間における半径 R の超球内と定義し,「セグメント」を時系列データのうち超球に含まれる各部分データ列と定義して,「データ密度」を全セグメントの長さの合計と定義する．すると，データ点 $\boldsymbol{o}(t)$ を中心としたデータ密度は

$$D(t) = \sum_{i \in S(t)} \|\boldsymbol{o}(i) - \boldsymbol{o}(i+1)\| \quad (1)$$
$$\text{where } S(t) = \{i; \|o(i) - o(t)\| \leq R\}$$

と計算され，これを各時刻ごとに計算することによって高頻度パターンの有無を評価することを考える．このとき，もし各時刻ごとに厳密に計算を行うと，データ構造を木構造にするなどの工夫をしたとしても計算時間は N^2 に依存した値となり時間がかかる．

そこで，本稿では近似最近傍探索の枠組みを用い，以下に示す方法でデータ密度を効率よく計算する．

Algorithm to find frequent patterns

1. 時刻 $t = 1$:
   超球内のセグメントの情報を保持するリンクリストを初期化（Fig.2(a)）
2. 時刻 $t = 2$ から N:
   リンクリストを更新（Fig.2(b),(c)）
   PLSH を用いて新規セグメントを検出しリンクリストへ追加（Fig.2(d)）
   PLSH を用いて分断セグメントを検出しリンクリストを更新（Fig.2(e)）
3. 時刻 $t = 1$ から N:
   大域最適化に基づき全高頻度パターンを抽出

時刻 $t = 1$ では，N 個の全データ点を調べ，超球内のセグメントの境界（時系列データと超球面との交点），つまり各セグ

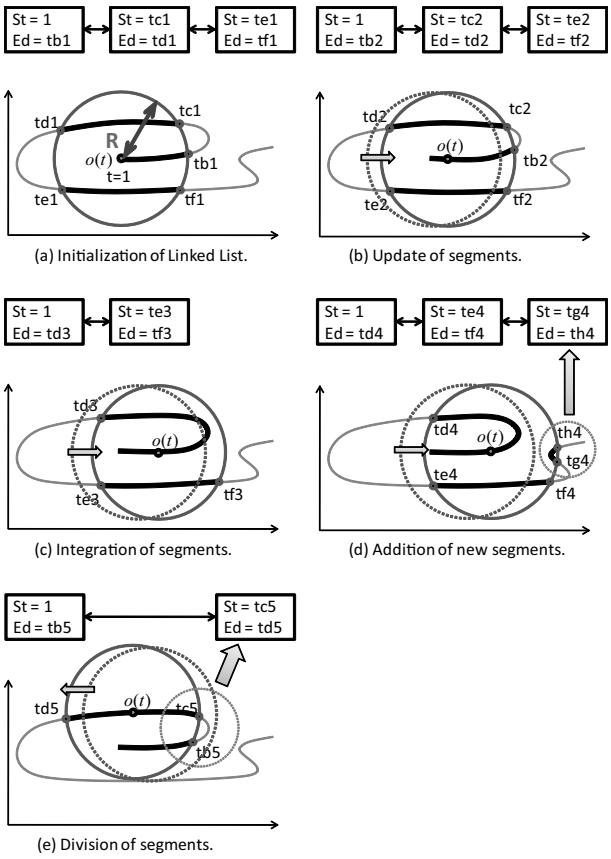


Fig. 2 Calculation of data density

メントの開始時刻 (st) と終了時刻 (ed) を Fig.2(a) に示すようにリンクリストとして保持する．

次に，時刻 t を 1 ずつ増加させながらリンクリストを更新する．まず，Fig.2(b) に示すように，前時刻で検出されリンクリストに保持されている各セグメントの境界を，現時刻での超球の境界を指すように更新する．通常，連続する時刻間での境界の移動幅は非常に小さく，多くの場合は 0 か 1 であることが期待される．そのため，リンクリストに保持するセグメント数に上限を設ける場合には，この更新にかかる計算量は定数であるとみなすことができる．このとき，セグメントが完全に超球の外に出た場合には，そのセグメントをリンクリストから除外する．また，もし Fig.2(c) に示すように二つのセグメントがつながる場合には，リンクリスト上でも統合する．

一方，超球内に新しく出現する新規セグメントについては，これを検出してリンクリストに追加する必要がある．このとき，もし N 個の全データ点を調べると全計算量が $O(N^2)$ となり時間がかかる．そこで，近傍のデータ点をサンプリングし，もしこのデータ点が超球内に含まれ，かつリンクリスト内の既知のセグメントに含まれない場合には，Fig.2(d) に示すように新規セグメントとみなして境界を求めてリンクリストに追加する．この場合，新規セグメントが超球内に出現した時点で発見されるとは限らないが，そのセグメントが高頻度パターンの一部である場合には，その後のいずれかの時刻において発見される確率が高い．そのため，発見された時点で過去に遡ってデータ密度を修正することによって，出現時点で発見されない問題を回避することができる．

このとき，サンプリング法として超球内のデータ点をランダムかつ効率よく選択することが重要になるが，それを実現する方法として Partly Locality Sensitive Hashing (PLSH) を 4 章で提案する．

また，Fig.2(e) に示すように超球内のセグメントが二つに分断されることがありうる．リンクリストにはセグメントの端点しか保持されていないため，分断を直ちに検出することはできない．これを検出するために，現時刻から一定時間 (T_{delay}) 前のデータ点 $\boldsymbol{o}(t - T_{\mathrm{delay}})$ において PLSH を使用して同様に近傍のデータ点をサンプリングし，もしその点を含むセグメントがリンクリストに含まれておりかつその点が $\boldsymbol{o}(t)$ における超球の外側にある場合には，そのセグメントを分割してリンクリストを更新する．この場合についても，分断が生じた時点で発見されるとは限らないが，発見された時点で過去に遡りデータ密度を修正することによって，分断が生じた時点で発見されない問題を回避することができる．

最後に，5 章で説明するように動的計画法を用いた大域最適化法によって全高頻度パターンを求める．

4. Partly Locality Sensitive Hashing を用いたデータ密度の計算

Partly Locality Sensitive Hashing (PLSH) は近似最近傍探索法の一種であり，局所性を保持するハッシュ関数を用いる Locality Sensitive Hashing (LSH) [15] に局所性を保持しないハッシュ関数を追加して拡張したものである．

4.1 Locality Sensitive Hashing

LSH の枠組みでは，ハッシュ関数 $g_l(\boldsymbol{p})$ $(1 \leq l \leq L)$ は以下のように定義される．

$$g_l(\boldsymbol{p}) =< h_{l1}(\boldsymbol{p}), h_{l2}(\boldsymbol{p}), \ldots, h_{lK}(\boldsymbol{p}) > \qquad (2)$$

ここで，$\boldsymbol{p}$ は d 次元の入力値であり，$g_l(\boldsymbol{p})$ は $\boldsymbol{p}$ から K 次元のハッシュ値を計算するハッシュ関数である．

$h_{lk}(\boldsymbol{p})$ は入力値の局所性を保持して 1 次元のハッシュ値を返す任意のハッシュ関数 $h : R^d \rightarrow U$ である．すなわち，入力値同士が近いほどそのハッシュ値が衝突する確率が高くなる．例えば線型写像に基づく以下のハッシュ関数を用いることができる．

$$h(\boldsymbol{p}) = \lfloor (\boldsymbol{a} \cdot \boldsymbol{p} + b)/w \rfloor \qquad (3)$$

ここで，$\boldsymbol{a}, b$ はハッシュ関数 $h(\boldsymbol{p})$ ごとに $\boldsymbol{a} \in R^d, ||\boldsymbol{a}|| = 1, 0 \leq b < w$ を満足するようにランダムに決定する．また，w はハッシュバケットの大きさを表す．

近傍探索の手順は，まず探索対象となるすべてのデータ点に L 個のハッシュ関数 $g_l(\boldsymbol{p})$ を適用して L 個のハッシュ値を計算し，対応する L 個のバケットにそれぞれデータ点を格納する．入力データ点 $\boldsymbol{p}$ が与えられると，同様に L 個のハッシュ関数 $g_l(\boldsymbol{p})$ から L 個のハッシュ値を計算し，各ハッシュ値に対応する L 個のバケットに格納されているデータ点群を調べ，入力データ点と十分近いデータ点を近傍点とみなす．最近傍点が入った

バケットが必ずしも探索されるとは限らないため LSH は近似最近傍探索法であるが，計算量を $O(N^{1/\alpha})$ に抑えることが可能である．ただし，$\alpha(>1)$ は LSH のパラメータによって決定される定数である．

4.2 Partly Locality Sensitive Hashing

PLSH の枠組みでは，ハッシュ関数 $g_l(\boldsymbol{p})$ $(1 \leq l \leq L)$ は以下のように定義される．

$$g_l(\boldsymbol{p}) =< hs_{l,1}(\boldsymbol{p}), \ldots, hs_{l,K_s}(\boldsymbol{p}), hi_{l,1}(\boldsymbol{p}), \ldots, hi_{l,K_i}(\boldsymbol{p}) >$$

ここで，$hs(\boldsymbol{p})$ は入力値の局所性を保持する任意のハッシュ関数 (Locality Sensitive Hash Function) $hs : R^d \rightarrow U$ を表し，$hi(\boldsymbol{p})$ は入力値の局所性を保持しない任意のハッシュ関数 (Locality Insensitive Hash Function) $hi : R^d \rightarrow U$ を表す．例えば線型写像に基づくハッシュ関数として，以下で定義するハッシュ関数を用いることができる (Fig.3)．

$$\begin{aligned} hs(\boldsymbol{p}) &= \lfloor (\boldsymbol{a}_s \cdot \boldsymbol{p} + b_s)/w_s \rfloor \\ hi(\boldsymbol{p}) &= \lfloor (\boldsymbol{a}_i \cdot \boldsymbol{p} + b_i) \rfloor \bmod w_i \end{aligned} \tag{4}$$

ここで，$hs(\boldsymbol{p})$ は式 (3) の局所性を保持するハッシュ関数と同一であるが，$hi(\boldsymbol{p})$ は離散化された剰余を計算する．$\boldsymbol{a}, b$ はハッシュ関数 $hs(\boldsymbol{p})$，$hi(\boldsymbol{p})$ ごとに $\boldsymbol{a} \in R^d, ||\boldsymbol{a}|| = 1, 0 \leq b < w$ を満足するようにランダムに決定する．

近傍探索の手順は，LSH の場合と同様に，まず探索対象となるすべてのデータ点に L 個のハッシュ関数 $g_l(\boldsymbol{p})$ を適用して L 個のハッシュ値を計算し，対応する L 個のバケットにそれぞれデータ点を格納する．入力データ点 $\boldsymbol{p}$ が与えられると，L 個のハッシュ関数 $g_l(\boldsymbol{p})$ から L 個のハッシュ値を計算し，対応する L 個のバケット内のデータ点群を調べる．

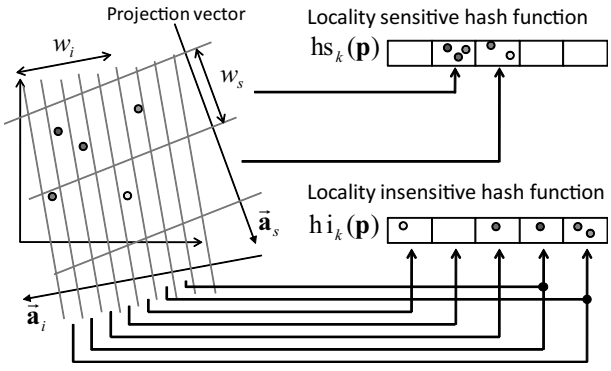


Fig. 3 Partly Locality Sensitive Hashing

4.3 PLSH を用いた疎なサンプリング法

3 章で述べたとおり，各時刻ごとに $\boldsymbol{o}(t)$ を中心とした超球内のデータ点をランダムかつ効率よくサンプリングしたい．もし Meng らの手法 [14] のようにデータ点の探索に LSH を使う場合は，バケットの大きさを表す w は超球内のデータ点を探索できるように半径 R に設定するため固定値となり，そのため Fig.4(a) に示すようにデータ数が増えるとバケット内のデータ数も増加し実際の計算量は $O(N^2)$ に近くなる．

このようにバケットの大きさが固定されている場合に PLSH は有用である．PLSH を使う場合，本研究では入力データ値を $d+1$ 次元ベクトル $(p_1, \ldots, p_d, t)^T$ で表現する．このうち最初の d 次元はデータを表し，最後の 1 次元はそのデータが観測された時刻を表す．

ハッシュ関数 $g_l(\boldsymbol{p})$ は，K_s 個の局所性を保持するハッシュ関数と 1 個の局所性を保持しないハッシュ関数により構成する．K_s 個の局所性を保持するハッシュ関数については，射影ベクトルである $\boldsymbol{a}_s$ の最後の要素は常に 0 とする．局所性を保持しないハッシュ関数については，射影ベクトルである $\boldsymbol{a}_i$ の最初の d 個の要素は常に 0 とし，最後の要素は 1 とする．これは，w_i (LIH width) で決定される数のハッシュ空間にデータ点を分散することに相当する．

これにより，Fig.4(b) に示すように，時間軸上で近傍に存在するデータ点同士では，互いに重複するデータを持たない独立したハッシュ空間を探索することになる．そのため，新規データ点（もしくは分断データ点）が発見された場合には過去に遡ってデータ密度を修正することにより，新規セグメント（もしくは分断セグメント）上のデータ点が発見される確率を LSH と同じに保ったまま，一つのバケット内に格納されるデータ数を大きく削減することが可能になり，探索時間を短縮することができる．バケットあたりのデータ数は $\frac{1}{w_i}$ となり，w_i は検出したいパターンの時間長より大きくならない範囲で決めればよい．

LSH の場合も，Fig.4(c) に示すように，時系列データをあらかじめある間隔 w_r (Reduction width) で間引いておくことによって，同等のデータ削減効果を得ることは可能である．しかし，PLSH の場合は，Fig.4(d) に示すように，時系列データを間隔 w_r で間引いた上でさらに時間軸方向にデータを分散することによって，$\frac{1}{w_i \cdot w_r}$ のデータ削減効果を得ることができる．

5. 高頻度パターンの抽出

同じ期間に計測された M 個の時系列データ $\boldsymbol{O} = \{\boldsymbol{O}_i = (\boldsymbol{o}_i(1), \cdots, \boldsymbol{o}_i(N)); 1 \leq i \leq M\}$ が与えられたときに，未知の高頻度パターンを抽出する問題を以下の 2 段階処理によって解く．

(1) データ密度に基づく高頻度パターンの抽出
(2) 高頻度パターンのクラスタリング

5.1 データ密度に基づく高頻度パターンの抽出

まず，パターンの種類とは無関係に，高頻度パターンである可能性の高い区間を $\boldsymbol{O}$ から抽出する．この問題を，$\boldsymbol{X} = (x_1, \cdots x_N), x_t \in \{1, \ldots, M, \mathrm{Non}\}$ のように各時刻にラベルを割り当てる組合わせ最適化問題として定式化する．ここで，$1, \ldots, M$ は高頻度パターンが属する時系列データを表し，Non はどの時系列データにおいても高頻度パターンではない，つまり低頻度パターンであることを表す．

そして，以下のエネルギー関数 $E(\boldsymbol{O}, \boldsymbol{X})$ を最小化することによって $\boldsymbol{X}$ を求める．

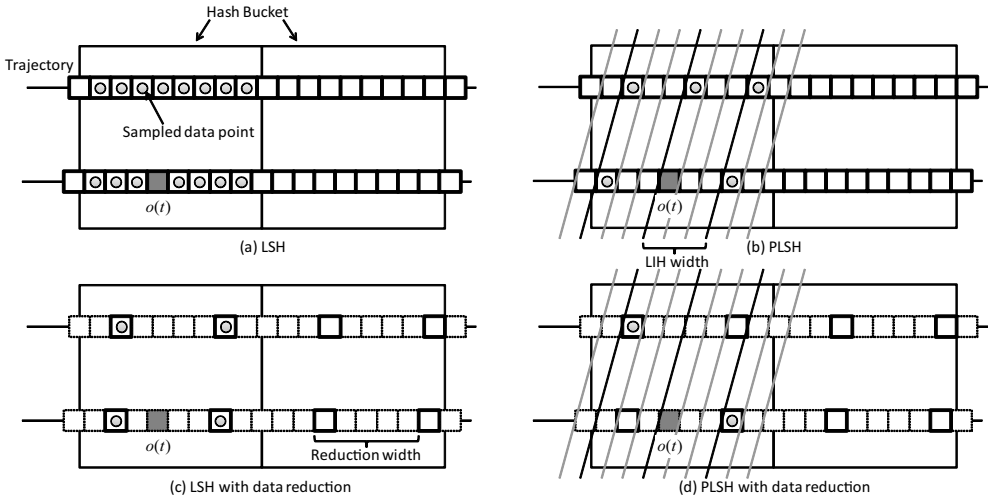


Fig. 4 Difference between LSH and PLSH

$$E(\boldsymbol{O},\boldsymbol{X}) = E_v(\boldsymbol{O},\boldsymbol{X}) + E_d(\boldsymbol{O},\boldsymbol{X}) + E_s(\boldsymbol{X}) \qquad (5)$$

エネルギー関数は，速度項，データ密度項，平滑化項により構成され，以下で各項について説明する．

5.1.1 速度項

この項の有無は入力データの特性に依存する．6.1 節で扱う対象のように計測値の時間変化に基づいて各時刻のデータ値が定義される場合には，計測値の時間変化が小さな区間ではデータ値 $\boldsymbol{o}_i(t)$ の大きさは 0 に近づく．そのため，原点付近のデータ密度は一般に非常に大きくなり，この区間からの検出を抑制する必要がある．

速度項 $E_v(\boldsymbol{O},\boldsymbol{X})$ はデータ値の変化量が小さい場合にペナルティを与える項であり，以下のように定義する．

$$E_v(\boldsymbol{O},\boldsymbol{X}) = \sum_t -\log(1-\exp(-\frac{|\dot{\boldsymbol{o}}_{x_t}(t)|}{<|\dot{\boldsymbol{o}}_{x_t}(t)|>}))$$

ただし $<|\dot{\boldsymbol{o}}_{x_t}(t)|>$ は $|\dot{\boldsymbol{o}}_{x_t}(t)|$ の平均である．

5.1.2 データ密度項

データ密度項 $E_d(\boldsymbol{O},\boldsymbol{X})$ はデータ密度 (式 (1)) が小さい場合にペナルティを課す項であり，時系列ごとに定義されたデータ密度 $D_m(t)$ を用いて以下のように定義する．

$$E_d(\boldsymbol{O},\boldsymbol{X}) = \sum_t -\log(1-\exp(-\frac{D_{x_t}(t)}{<D_{x_t}(t)>}))$$

ただし $<D_{x_t}(t)>$ は $D_{x_t}(t)$ の平均である．

5.1.3 平滑化項

平滑化項 $E_s(\boldsymbol{X})$ はラベル割り当て $\boldsymbol{X}$ の事前分布を規定する項であり，本研究では連続するラベルが異なる値を取るときに以下のペナルティを与えることによって，短いパターンの出現を抑制する．

$$E_s(\boldsymbol{X}) = \sum_t T(x_t \neq x_{t+1}) \cdot C_{\mathrm{smooth}}$$

ただし，C_{smooth} は定数であり，$T(s)$ は $T(\mathrm{true}) = 1, T(\mathrm{false}) = 0$ と定義する．

式 (5) のすべての項は一次のマルコフ性を満足しており，動的計画法によって解析的に最小化することができる．

5.2 高頻度パターンのクラスタリング

前節で抽出されたパターンには異なる種類の動作が混在しているため，凝集的クラスタリングによって同じ種類の動作を統合する．本研究では，抽出されたすべてのパターンの組ごとに，パターン長に対して対応点同士の距離が超球の半径 R 以下である部分の割合を求め，その割合が閾値 C_{mutual} 以上の場合は同じ動作に属するとする．

ただし高速化のため，時刻 t のリンクリストに保持された j 番目のセグメント (Fig.2 の St〜Ed) の中点 t_j を，$o(t)$ に対する距離が R 以下の j 番目の対応点として用いる．そして，式 (6) が満足された場合に，時刻 t を含むパターンと時刻 t_j を含むパターンを統合する．なお，対応するパターンがリンクリストにない場合は，その時刻における式 (6) の分子は 0 とする．

$$\frac{\sum_{t=t_s}^{t_e} 1 - |\frac{t-t_s}{t_e-t_s+1} - \frac{t_j-t_{sj}}{t_{ej}-t_{sj}+1}|}{t_e-t_s+1} \geq C_{\mathrm{mutual}} \qquad (6)$$

ここで t_s, t_e および t_{sj}, t_{ej} は各パターンの始点と終点の時刻を表し，各パターン内で正規化された t と t_j の差をペナルティとして与えることによって非線形伸縮マッチングを実現している．

5.3 計算量

計算量は，PLSH のハッシュテーブルの作成およびデータ密度の計算に $O(N^{1+1/\alpha})$，動的計画法の計算に $O(N)$ を要する．抽出されたパターン数が N に対して十分に小さいと見なすと，全体の計算量は $O(N^{1+1/\alpha})$ となる．

6. 実　　験

提案手法を評価するため，画像列からの高頻度な全身運動の抽出実験とモーションキャプチャデータからの高頻度な物体操作の抽出実験を行った．実験では，新規セグメントおよび分断セグメントのサンプリング法として，LSH に基づく方法と提案する PLSH に基づく方法との比較を行った．なお，提案手法

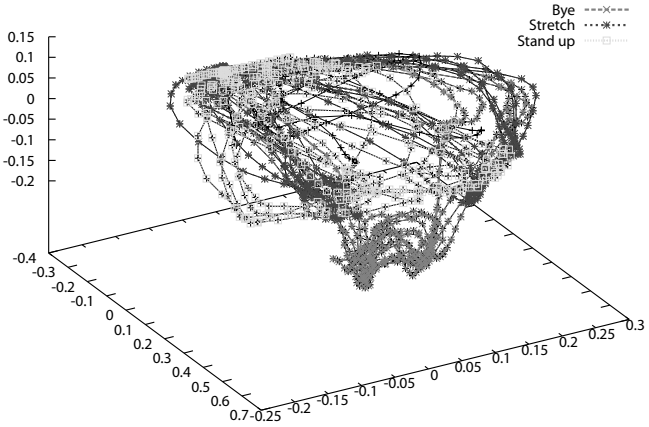


Fig. 6 Visualization of dataset 1

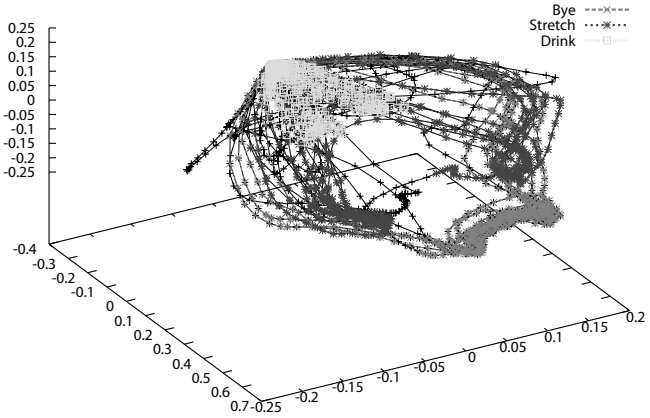


Fig. 7 Visualization of dataset 2

は超球境界付近の疎なサンプリングが目的であるため，A-NN 法 [16] のような超球の内側に探索範囲を絞る方法は比較対象として不適当である．また，LSH の拡張法である，例えばハッシュバケット内のデータ数を均等化する Principal Component Hashing [17] に対して PLSH を使用しても同様の効果を発揮することが予想されるが，本実験では簡単のため LSH と比較を行った．

なお以降の実験では，LSH および PLSH のいずれも L は 8 に K_s は 3 に設定した．また，すべての計算は Xeon 3.0GHz の計算機上で行った．

6.1 画像列からの高頻度な全身運動の抽出

6.1.1 実験条件

本実験では，静止したカメラで環境を観測したときに，画像中の任意の場所で発生する高頻度パターンを検出することを目的とする．画像中に局所的に現れる時系列パターンを位置に不変な特徴量で記述するため，高次局所自己相関特徴量 (CHLAC) [18] を画像のデータ表現として用いる．

CHLAC 特徴量は，連続する 3 枚のフレーム間差分画像 $f_t(\boldsymbol{p})$ から式 (7) で定義される高次局所自己相関特徴量 (本研究では N_{C}=2) を計算することによって求める．

$$c_d(t) = \int f_t(\boldsymbol{p}) f_t(\boldsymbol{p}+\boldsymbol{a}_1)\cdots f_t(\boldsymbol{p}+\boldsymbol{a}_{N_{\mathrm{C}}}) d\boldsymbol{p} \qquad (7)$$

3×3×3 の時空間領域では，冗長な組み合わせを除くと $(\boldsymbol{a}_1, \boldsymbol{a}_2)$ の取りうる組み合わせは 251 とおりとなるため，CHLAC 特徴量は $d = 251$ 次元のベクトル $\boldsymbol{c}(t) = (c_1(t), \ldots, c_d(t))^T$ となる．ただしこのままでは次元数が大きいため，事前に $\boldsymbol{c}(t)$ を PCA によって 6 次元に削減した $\boldsymbol{o}(t)$ を各時刻のデータ値とする．

本実験では時系列データは一つであるため，{1 = 高頻度, Non} の 2 種類のラベルの割り当て問題となる．また，5.1.1 節で述べたように原点付近からの検出を抑制するため，式 (5) のエネルギー関数として速度項を含むものを使用する．

実験では，Fig.5 に示す 4 種類の全身運動パターンをそれぞれ複数回含む二つの動画像データセットを用意し，これから高頻度パターンを抽出した．データセット 1 には Bye が 5 回，Stretch が 6 回，Stand-up が 5 回含まれている．データセット 2 には，Bye が 7 回，Stretch が 7 回，Drink が 8 回と，ノイズとして無関係な動作が 4 種類含まれている．

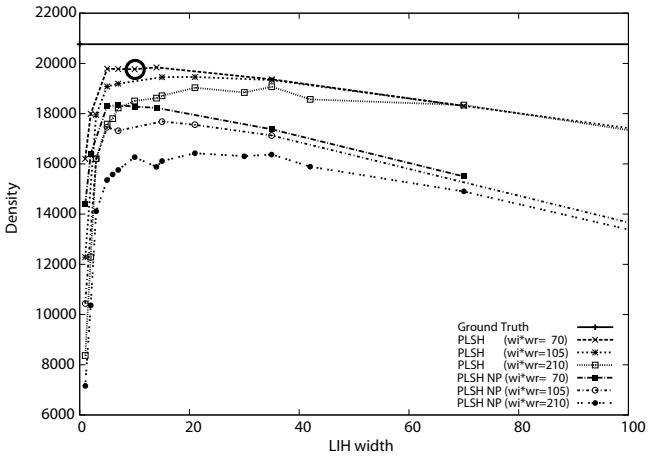


Fig. 8 Estimation of data density using dataset 1

Fig.6 と Fig.7 に，画像のデータ表現である $\boldsymbol{o}(t)$ の主要な 3 成分を可視化した結果を示す．図の異なる濃度の線はそれぞれ 3 種類の動作を表し，黒い線はその他の区間を表す．3 種類の動作がそれぞれ類似した判別可能な軌跡を描いており，本論文で提案するデータ密度に基づいた高頻度パターンの抽出法が妥当な方法であることが分かる．

6.1.2 データ密度推定におけるサンプリング方法の評価

4.3 節で述べたように，PLSH では w_r の間隔で時系列データを間引き，かつ w_i (LIH width) で決定される数のハッシュ空間にデータ点を分散することによって，ハッシュバケット内のデータ数を $\frac{1}{w_i \cdot w_r}$ に削減し，疎なサンプリングを実現する．

計算時間は $\frac{1}{w_i \cdot w_r}$ に比例するため，データ密度の推定精度の低下を抑えた最適な w_i, w_r を決定する必要がある．

異なる w_i, w_r の組に対して全データ点におけるデータ密度の総和である累積データ密度を計算した結果を Fig. 8, 9 に示す．グラフの縦軸は累積データ密度を表し，横軸は w_i を表す．グラフの折れ線は，$w_i \cdot w_r$ の値が等しい点を結んだものである．NP (Non Propagation) は，新規セグメントもしくは分断セグ

(a) Bye (b) Stretch

(c) Stand-up (d) Drink

Fig. 5 4 whole body motions to be detected

Table 1 Evaluation of dataset 1 [2700 frames]

Action	Bye	Stretch	Stand-up	False	False	Precision	Recall	Time
Presented #	5	6	5	Positive	Negative			[msec]
LSH (wr=1)	5.00	5.00	5.00	0.00	1.00	1.00	0.94	5807
LSH (wr=15)	5.00	5.00	5.00	0.00	1.00	1.00	0.94	512
LSH (wr=70)	5.00	4.90	0.00	0.00	6.10	1.00	0.62	198
PLSH	5.00	5.00	5.00	0.00	1.00	1.00	0.94	224

Table 2 Evaluation of dataset 2 [3600 frames]

Action	Bye	Stretch	Drink	False	False	Precision	Recall	Time
Presented #	7	7	8	Positive	Negative			[msec]
LSH (wr=1)	7.00	6.00	8.00	1.00	1.00	0.95	0.95	13499
LSH (wr=15)	7.00	4.00	8.00	1.00	3.00	0.95	0.86	1134
LSH (wr=70)	4.00	0.00	7.20	0.30	10.80	0.97	0.51	338
PLSH	7.00	3.00	8.00	1.00	4.00	0.95	0.82	482

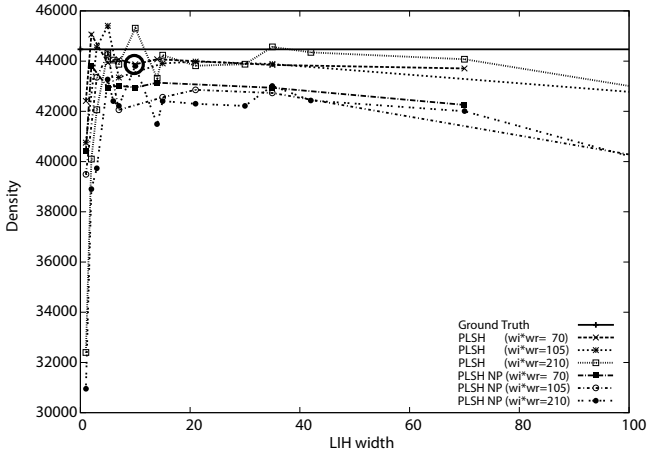


Fig. 9 Estimation of data density using dataset 2

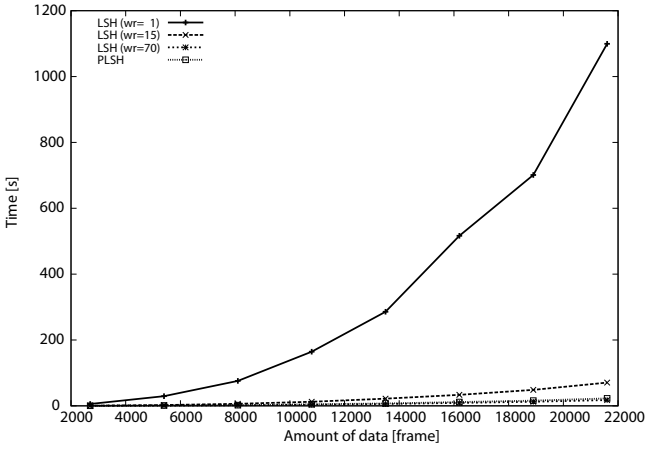


Fig. 10 Evaluation of computational time v.s. amount of data

メントがサンプリングによって発見された後に，過去に遡ってデータ密度を修正しない場合の結果を示す．

Fig.9 において，厳密解 (Ground Truth) よりも累積データ密度が大きく推定されている場合があるが，これはこのデータセットでは分断セグメントが十分に検出されなかったためである．

折れ線の一番左側の点が，w_i (LIH width) が 1 の場合，つまり LSH を使用した場合に対応する．グラフより，$w_i \cdot w_r$ を一定にした場合には，w_i を 1 から大きくするにつれて累積データ密度が厳密解に近くなり，さらに大きくすると厳密解から離れていくことが分かる．これは，w_i と w_r のいずれか片方が大きな値を取ると，サンプリングの間隔が広くなり十分なサンプリングがなされないことを意味する．

この結果をふまえ，以降の実験ではグラフに丸で示したように PLSH の w_i は 10 に w_r は 7 に設定した．

6.1.3 高頻度パターンの抽出結果

(1) LSH (間引きなし, $w_r = 1$)，(2) LSH (PLSH と同等の

累積データ密度, $w_r = 15$), (3) LSH (PLSH と同等の削減率, $w_r = 70$), (4) PLSH の四つの手法に基づき二つのデータセットから高頻度パターンを抽出した結果を Table 1,2 に示す．ここで (2) LSH は，PLSH で計算される累積データ密度 (Fig.8,9 の丸) 以上の累積データ密度が得られるように設計した LSH を表す．結果の評価として適合率 (Precision) と再現率 (Recall) を以下のように計算した．

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

ただし，TP は True Positive，FP は False Positive，FN は False Negative を表す．ハッシュ関数のパラメータは乱数で決まるため，それぞれ 10 回試行を行った平均を計算した．

LSH (間引きなし, $w_r = 1$) と LSH (PLSH と同等の累積データ密度, $w_r = 15$) では PLSH とほぼ同等の抽出結果が得られたが，データの削減率で劣るため計算時間は長くなる．一方，LSH (PLSH と同等の削減率, $w_r = 70$) は PLSH と比べてやや計算時間が短いが，抽出結果は特に再現率の点で大きく劣っている．なお，同等の削減率であるにもかかわらず LSH の計算時間が短くなる理由は，サンプリングが不十分なためリンクリストに保持されるセグメントの数が少なくなり，リンクリストの更新に要する時間が短縮されたからである．

この結果から，PLSH は LSH と比較して少ないサンプル数で高頻度パターンを抽出できることが分かる．

6.1.4 データ量の増加に対する計算時間の評価

異なる量のデータに対して前節の 4 手法を比較した結果を Fig.10 に示す．本実験では，Table 1 のデータセット (2700 frame) にノイズを加えて単純に連結する方法によって異なるデータ量のデータセットを 10 個作成した．なお，CHLAC 特徴量や PCA の計算にかかる時間は N に対して線型であり，また本実験の目的は 4 手法の比較であるため，表の計算時間には含めない．

グラフより，ハッシュバケット内のデータ数を PLSH に基づき削減することによって，計算時間が大きく削減されたことが分かる．

6.2 モーションキャプチャデータからの高頻度な物体操作の抽出

6.2.1 実験条件

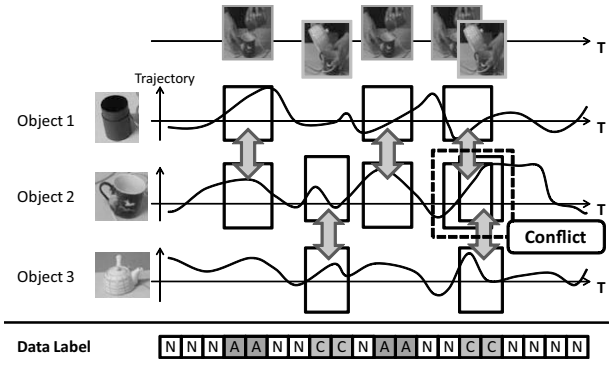
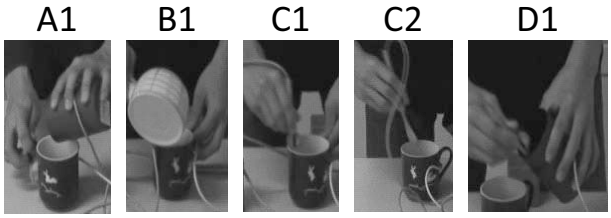


Fig. 11 Finding consistent frequent motion patterns



A1: Pour from a container to a cup from the left
B1: Pour from a teapot to a cup
C1: Mix inside a cup with a spoon
C2: Put into a cup with a spoon
D1: Put from a container with a spoon

Fig. 12 5 manipulation actions

本実験では，Fig.11 に示すように互いに関連する複数の時系列データを扱う．入力は物体（図の例では 3 個）の運動軌跡であり，入力軌跡中に高頻度で現れる運動パターンを求めることが目的である．もし運動軌跡が互いに独立であれば，各運動軌跡から独立に高頻度運動パターンを抽出すればよいが，もし非独立である場合には互いに矛盾しない運動パターンを抽出する必要がある．

非独立な例として，物体間の相互作用，すなわち物体間の相対運動を求める場合が挙げられる．この場合，Fig. 11 に示すように，ある瞬間には (A) 物体 1 と物体 2，(B) 物体 1 と物体 3，(C) 物体 2 と物体 3 の 3 とおりの相互作用のうちどれか一つしか発生しない．もし，2 種類の相互作用が同時に検出された場合は，矛盾した状態を意味する．

本実験では，ある瞬間には同時に 1 種類の相互作用しか発生しないと仮定する．物体が 4 個以上ある場合や，同時に 3 個以上の物体が相互作用を行う場合など，この仮定が成り立たない状況も考えられるが，多くの場合は成立すると考える．

この場合，ラベル $\{1, \ldots, M, \text{Non}\}$ の M は 2 物体の組み合わせの数と等しくなる．また，式 (5) のエネルギー関数として速度項を含まないものを使用する．

実験では，2 種類の異なる物体操作タスクを被験者が実行し，提案手法によって観測データから高頻度パターンを抽出してその評価を行った．物体は 4 種類存在し，各物体の運動軌跡の計測には磁気式位置計測装置 (Polhemus FASTRAK, 30[Hz]) を使用した．

物体操作は Fig.12 に示す 5 種類を定義し，被験者はあらかじめ決められた回数だけこれらの操作を任意の順序で実行した．また，長期計測の場合の環境変化を模擬するために，タスク実行中に適宜テーブル上の物体の位置を入れ替えるように指示された．

6.2.2 データ密度推定におけるサンプリング方法の評価

異なる w_i, w_r の組に対して全データ点におけるデータ密度の総和である累積データ密度を計算した結果を Fig. 13, 14 に示す．グラフの縦軸は累積データ密度を表し，横軸は w_i を表す．グラフの折れ線は $w_i \cdot w_r$ の値が等しい点を結んだものである．NP (Non Propagation) は，新規セグメントもしくは分断セグメントがサンプリングによって発見されたあとに，過去

Table 3 Evaluation of dataset 3 [3737 frames]

Action	A1	B1	C1	False	False	Precision	Recall	Time
Presented #	5	7	6	Positive	Negative			[msec]
LSH (wr=1)	5.00	7.00	6.00	0.00	0.00	1.00	1.00	15665
LSH (wr=15)	5.00	7.00	6.00	0.00	0.00	1.00	1.00	1386
LSH (wr=70)	4.00	7.00	6.00	0.00	1.00	1.00	0.94	438
PLSH	5.00	7.00	6.00	0.00	0.00	1.00	1.00	482

Table 4 Evaluation of dataset 4 [5177 frames]

Action	B1	C1	C2	D1	False	False	Precision	Recall	Time
Presented #	7	5	10	10	Positive	Negative			[msec]
LSH (wr=1)	7.00	3.00	6.00	0.00	0.00	16.00	1.00	0.50	18671
LSH (wr=15)	7.00	3.00	5.00	0.00	0.00	17.00	1.00	0.47	1537
LSH (wr=70)	4.00	1.50	0.00	0.00	0.00	26.50	1.00	0.17	552
PLSH	7.00	3.00	5.00	0.00	0.00	17.00	1.00	0.47	602

に遡ってデータ密度を修正しない場合の結果を示す.

折れ線の一番左側の点が，w_i (LIH width) が 1 の場合，つまり LSH を使用した場合に対応する．グラフより，6.1.2 項の実験と同様に，$w_i \cdot w_r$ を一定にした場合には，w_i を 1 から大きくするにつれて累積データ密度が厳密解に近くなり，さらに大きくすると厳密解から離れていくことが分かる.

この結果をふまえ，以降の実験ではグラフに丸で示したように PLSH の w_i は 10 に w_r は 7 に設定した.

6.2.3 高頻度パターンの抽出結果

(1) LSH (間引きなし, $w_r = 1$)，(2) LSH (PLSH と同等の累積データ密度, $w_r = 15$)，(3) LSH (PLSH と同等の削減率, $w_r = 70$)，(4) PLSH の四つの手法に基づき二つのデータセットから高頻度パターンを抽出した結果を Table 3,4 に示す. 6.1.3 項と同様に，適合率 (Precision) と再現率 (Recall) を結果の評価に用いた.

LSH (間引きなし, $w_r = 1$) と LSH (PLSH と同等の累積データ密度, $w_r = 15$) では PLSH とほぼ同等の抽出結果が得られたが，データの削減率で劣るため計算時間は長くなる．一方，LSH (PLSH と同等の削減率, $w_r = 70$) は PLSH と比べてやや計算時間が短いが，特に Table 4 の結果において再現率が大きく劣っている．この結果から，PLSH は LSH と比較して少ないサンプル数で高頻度パターンを抽出できることが分かる.

なお Table 4 において D1 が常に未検出である理由は，この実験では D1 の直後に常に C2 が続くため，この二つの動作が一つの動作として検出され C2 とラベル付けされたためである.

6.2.4 データ量の増加に対する計算時間の評価

異なる量のデータに対して前節の 4 手法を比較した結果を Fig.15 に示す．本実験では，Table 4 のデータセット (5177 frame) にノイズを加えて単純に連結する方法によって異なるデータ量のデータセットを 10 個作成した.

グラフより，6.1.4 項の結果と同様に，ハッシュバケット内のデータ数を PLSH に基づき削減することによって，計算時間が大きく削減されたことが分かる.

7. ま と め

本稿では，時系列データから非線形伸縮する比較的長い高頻度パターンを効率よく抽出する方法を提案した.

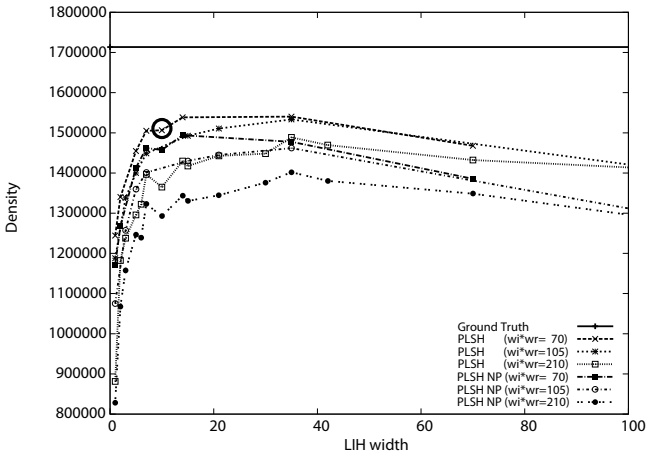


Fig. 13 Estimation of data density using dataset 3

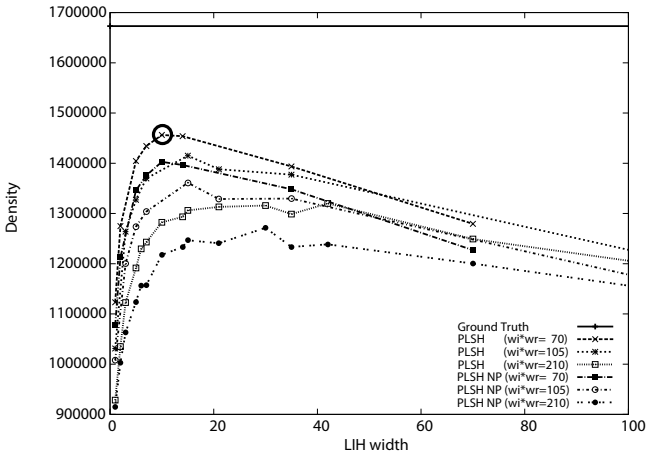


Fig. 14 Estimation of data density using dataset 4

Partly Locality Sensitive Hashing (PLSH) を提案し，近似最近傍探索の枠組みを利用した疎なサンプリング法に基づき近傍の類似パターンを効率よく探索することによって，全体の計算量を $O(N^{1+1/\alpha})$ に抑えた.

また，画像列から高頻度な全身運動を抽出する実験とモーションキャプチャデータから高頻度な物体操作を抽出する実験にお

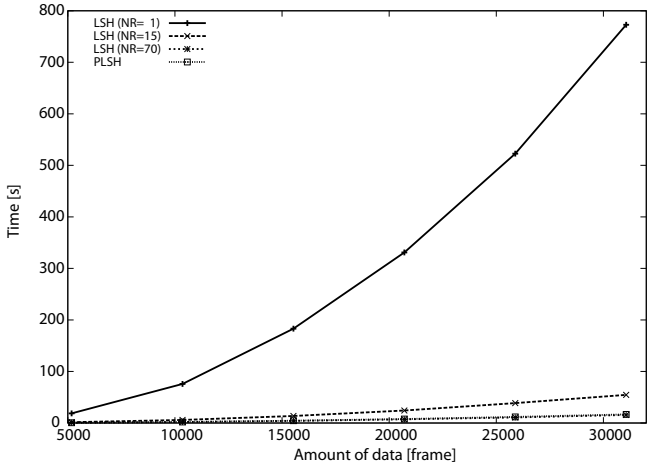


Fig. 15 Evaluation of computational time v.s. amount of data

いて，従来の LSH を用いた方法との比較の観点から提案手法の優位性を示した．

謝辞 本研究は，科学研究費補助金若手 (B)(21700224) および科学技術総合推進費補助金「若手研究者の自立的研究環境整備促進」の補助を受けている．

参 考 文 献

[1] K. Ikeuchi and T. Suehiro. Toward an assembly plan from observation part i: Task recognition with polyhedral objects. *IEEE Trans. Robotics and Automation*, Vol. 10, No. 3, pp. 368–384, 1994.

[2] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching. *IEEE Trans. Robotics and Automation*, Vol. 10, No. 6, pp. 799–822, 1994.

[3] Keni Bernardin, Koichi Ogawara, Katsushi Ikeuchi, and Ruediger Dillmann. A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models. *IEEE Transactions on Robotics*, Vol. 21, No. 1, pp. 47–57, 2005.

[4] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. In *Proc. of 4th International Conference on Foundations of Data Organization and Algorithms*, pp. 69–84, 1993.

[5] Chang-Shing Perng, Haixun Wang, Sylvia R. Zhang, and D. Stott Parker. Landmarks: A new model for similarity-based pattern querying in time series databases. In *16th International Conference on Data Engineering (ICDE'00)*, pp. 33–42, 2000.

[6] R. Staden. Methods for discovering novel motifs in nucleic acid sequences. *Computer Applications in the Biosciences*, Vol. 5, No. 5, pp. 293–298, 1989.

[7] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Pranav Patel. Finding motifs in time series. In *Proc. of the 2nd Workshop on Temporal Data Mining*, pp. 53–68, 2002.

[8] Dragomir Yankov, Eamonn Keogh, Jose Medina, Bill Chiu, and Victor Zordan. Detecting time series motifs under uniform scaling. In *Proc. of the 13th ACM KDD Intl. Conf. on Knowledge Discovery and Data Mining*, pp. 844–853, 2007.

[9] Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, and Brandon Westover. Exact discovery of time series motifs. In *Proc. of 2009 SIAM International Conference on Data Mining: SDM*, pp. 1–12, 2009.

[10] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *Proc. of BMVC*, 2006.

[11] 木谷 クリス真実, 岡部孝弘, 佐藤洋一, 杉本晃宏. 視覚的文脈を考慮した人物動作カテゴリの教師無し学習. In *Proc. of Meeting on Image Recognition and Understanding (MIRU)*, pp. 1–6, 2008.

[12] T. Zhao, T. Wang, and H. Shum. Learning a highly structured motion model for 3d human tracking. In *Proc. of Asian Conference of Computer Vision*, 2002.

[13] Seiichi Uchida, Akihiro Mori, Ryo Kurazume, Rinichiro Taniguchi, and Tsutomu Hasegawa. Logical dp matching for detecting similar subsequence. In *Proc. of Asian Conference of Computer Vision*, 2007.

[14] Jingjing Meng, Junsong Yuan, Mat Hans, and Ying Wu. Mining motifs from human motion. In *Proc. of EUROGRAPHICS'08*, 2008.

[15] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. of the twentieth annual Symposium on Computational Geometry*, pp. 253–262, 2004.

[16] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, Vol. 45, No. 6, pp. 891–923, 1998.

[17] Yusuke Matsushita and Toshikazu Wada. Principal component hashing: An accelerated approximate nearest neighbor search. In *Proc. of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology (PSIVT)*, pp. 374–385, 2009.

[18] T. Kobayashi and N. Otsu. Action and simultaneous multiple-person identification using cubic higher-order local auto-correlation. In *Proc. Int. Conference on Pattern Recognition: ICPR*, pp. 741–744, 2004.

小川原光一 (Koichi Ogawara)

2002 年東京大学大学院工学系研究科電子情報工学専攻博士課程修了，博士（工学）．科学技術振興機構博士研究員，東京大学生産技術研究所特任助手を経て，2006 年 12 月より九州大学特任准教授，現在に至る．コンピュータビジョン・知能ロボットの研究に従事．2007 年 IEEE/ICRA'07 Best Vision Paper Award を受賞．（日本ロボット学会正会員）

田邉康史 (Yasufumi Tanabe)

2010 年九州大学大学院システム情報科学府情報知能工学専攻修士課程修了．同年株式会社デンソー入社，現在に至る．知能ロボットの研究に従事．（日本ロボット学会学生会員）

倉爪 亮 (Ryo Kurazume)

1991 年東京工業大学機械物理工学専攻修士課程修了．同年（株）富士通研究所入社，1995 年東京工業大学機械宇宙学科助手，2000 年スタンフォード大客員研究員，同年東京大学生産技術研究所博士研究員，2002 年九州大学システム情報科学研究院助教授，2007 年より同教授，現在に至る．群ロボット，歩行機械，レーザ計測の研究に従事．博士（工学）．（日本ロボット学会正会員）

長谷川勉 (Tsutomu Hasegawa)

1973 年東京工業大学電子物理工学科卒業．同年電子技術総合研究所勤務．1992 年より九州大学工学部情報工学科教授．現在同大学大学院システム情報科学研究院教授．知能ロボットの研究に従事．工学博士．計測自動制御学会，電気学会，日本機械学会などの会員．（日本ロボット学会正会員）