Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

上方視点距離画像を用いた人物姿勢推定手法の検討

渡邊昭信[†1]　味松康行[†1]　上村俊夫[†1]　中村克行[†1]

概要：製造現場における作業員の動作解析や，店舗における顧客行動の分析などにおいて，TOF（Time of Flight）センサから得られる距離画像データの活用が期待されている．距離画像データを用いた人物姿勢推定として多くの手法が提案されているが，上方から見下ろす視点でのユースケースは十分に検討が進んでいない．本研究では，上方視点距離画像からの人物推定手法として，逐次探索型のルールベース手法と，ランダムフォレストによる機械学習手法を開発し，人物腕関節座標の推定精度と処理速度を比較した．逐次探索手法では TOF センサの SoC にて正解率 75%でリアルタイムに姿勢推定可能なことと，機械学習手法では一般的な CPU にて正解率 60%でリアルタイムに姿勢推定可能なことが確認できた．

Vertical View Human Skeleton Recognition Method from Range Images

AKINOBU WATANABE[†1]　YASUYUKI MIMATSU[†1]
TOSHIO KAMIMURA[†1]　KATSUYUKI NAKAMURA[†1]

Abstract: In the analyses of the operation of workers in the manufacturing premise and the customer action in the retail store, the inflection of provided range image data by TOF (Time of Flight) sensor is expected. Many techniques are suggested as a person posture estimate using range image data, but, as for the use case in the point of view looking down, examination does not advance enough. In this study, we developed the rule base technique of the sequential search model and a machine learning technique by Random Forest as a person estimate technique from upward view range image and compared estimate precision and the transaction speed of the person arm joint coordinate. We confirmed the correct prediction ratio of the machine learning technique in posture estimate is 60% on general CPU and the one of the sequential search technique is 75% of correct prediction rates on SoC of the TOF sensor.

1. はじめに

センサ機器や IT システムの発達・低価格化により，それらを利用して機械・乗り物・建物等から情報を収集し，分析・制御に活用する IoT (Internet of Things)市場が拡大しつつあり, IoT の市場規模は 2015 年の約 7000 億ドルから 2019 年には約 1.3 兆ドルに成長すると予測されている[1].

IoT の進展に伴い，現場の Physical な機械や施設のデータを収集し，IT システム内の Cyber な世界で「デジタル・ツイン」として再現することにより，情報処理技術を使って現状分析・将来予測を行い，結果を現場にフィードバックしようとする動きがある．Siemens や GE もデジタル・ツインに関連する研究・開発を進め，既に一般媒体で情報を発信し始めている[2][3].

さらに，センシング対象を「モノ」から「ヒト」に拡大しつつある．ヒトも含めた現場をデジタル・ツインとして再現することにより，人の動きも含めた改善・効率化や作業ミスの検出・低減が可能となるが，そのためには，人の存在や動きを検出・認識し，デジタル情報として加工・定型化する必要がある．コンシューマ市場では Microsoft がゲーム向けに Kinect を発表した[a][4]. Kinect はプレーヤーの動きを動画としてキャプチャするだけでなく，被写体の関節位置を 3 次元空間内の座標データとして取得できる．また，PC 上で利用できる SDK が提供されているため産業界向けに応用する動きがある[5]．例えば，生産現場作業員の逸脱行動を検出する産業分野向けシステムの研究に Kinect が利用されている[6].

Kinect の利用に際しては，センサ 1 台に対して，GPU を搭載した PC が 1 台必要であり，さらに，側方からの視点を前提としている事から，生産作業現場では，遮蔽物により視野が確保できないなど，センサ設置に対して制約が多い事が問題となっている．

よって，上記問題を解決するべく，TOF センサをはじめとする，汎用 3D センサで利用可能な，産業用途に適した姿勢検出技術の確立を目的とした．

そのために，上方・近距離から撮影した距離画像(各画素

†1 (株)日立製作所
Hitachi Ltd.

a) Microsoft, Kinect は，米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

がセンサから被写体までの距離データを持つ画像)を対象に人の姿勢を検出する必要があり，さらに，より適用環境を広げるためには上方からの視点に限らず，被写体を撮影するセンサの角度や被写体との距離を変えた場合にも姿勢検出できることが望ましい．

そこで本研究では，人物を上方から撮影した距離画像に加え，側方から撮影した距離画像の両方に対して姿勢検出できる推定方法を検討・実装し，その精度と処理時間を評価した．

本研究では，人体を撮影した距離画像から，人体の関節位置(スケルトン)を抽出する処理を対象とする．特に，被写体の真上から見下ろした視点の撮影角度で，高速軽量にスケルトンを抽出できる処理の提案と，さまざまな撮影角度・撮影距離に対応できるスケルトン抽出方法の提案と，それらの精度および処理時間の評価を対象とする．

2. 上方視点スケルトン抽出手法の提案と評価

2.1 既存のスケルトン抽出手法の概要

人体の関節位置を抽出する手法としては，Kinect のスケルトン抽出処理が知られている[7]．人体を正面から撮影して取得した距離画像データから，画素ごとに周囲の画素との距離差を特徴量として算出し，画素が所属する人体部位をラベルとして，Random Forest 手法で機械学習を行う．推定された人体部位の画素群からミーンシフト手法で重心を求めて，関節座標として算出する．

Kinect のスケルトン抽出は，側方視点では検出するが，上方から見下ろす視点では検出できないという課題がある．

一方，機械学習手法を用いず，距離画像からスケルトンを抽出するルールベースの手法も知られている[8]．距離画像の点群に最もよく一致する姿勢の人体モデルからスケルトンを求める手法[9]や，人体のある点を起点とし部位の大きさから姿勢を推定する手法である[10]．このように，これまでは，主に側方視点の距離画像を対象とした研究が行われていた．

2.2 要件

産業分野向け逸脱動作検知システムの要件としては，上方からの見下ろし視点であることが必要である．これは，生産現場の作業員の周囲には大型の生産設備が連続して設置されており，作業員を側方から撮影できる位置にセンシングデバイスを設置するのは困難であり，遮蔽物の少ない上方からの撮影が必要なためである．

また，一般的な人の動作をリアルタイムに認識する場合のフレームレートとして 30fps を想定しており，さらに，複数のセンサから送られる距離画像を 1 台のサーバで処理するためには，あるいは，サーバと比べてコンピューティングリソースが少ないセンサ内蔵ボード上で処理するためには，処理負荷が軽く，処理時間は短いことが望ましい．さらに，未検出率よりも誤検出率が重視される．これは，座標精度が低い検出結果である誤検出よりも，未検出のほうが，逸脱動作の検知精度が向上するためである．

そこで，上記の課題と要件を解決するために，2 つのアプローチを検討した．

一つは，ルールベース手法であり，上方視点で撮影した距離画像から得られる点群をボクセルリスト化し，人体の部位の長さで探索することで，高速に検出する手法をベースに，推定処理やトラッキング処理の追加による精度向上，上方以外の視点への拡大により設置の自由度を拡大するアプローチである．

もう一つは，機械学習手法であり，Kinect のスケルトン抽出処理と同様な手法[7]をベースに，上方視点を追加し，さらに特徴量を工夫することで処理高速化を目指すアプローチである．

3. ルールベースのスケルトン抽出手法

3.1 見下ろし視点のスケルトン抽出手法の概要

見下ろし視点では，遮蔽物が少ないため，頭部を安定して検出できることが期待される．よって，頭部を検出した後に，頭座標を起点として，肩，肘，手首，手の順番で探索して検出する逐次探索型のスケルトン抽出手法を開発した．

図 1 に，TOF センサと制御サーバからなるスケルトン抽出システムの構成図（左側）と，TOF センサのシステム構成図（右側）を示す．

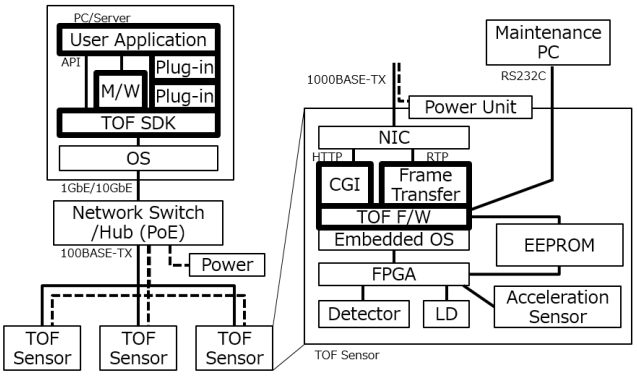


図 1　TOF スケルトン検出システムの構成

Figure 1　TOF Skeleton Detection System Structure.

図 1 は，日立 LG データストレージの TOF SDK マニュアル[11]より抜粋したシステム構成図に，一部追記した図である．スケルトン処理部は，PC/Server 内の M/W（Middleware）に含まれる．太線はソフトウェアで構成され，その他はハードウェアであることを示す．点線は TOF センサへの電源供給ラインを示す．

図 2 に，逐次探索型スケルトン抽出処理のフロー図を示す．

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

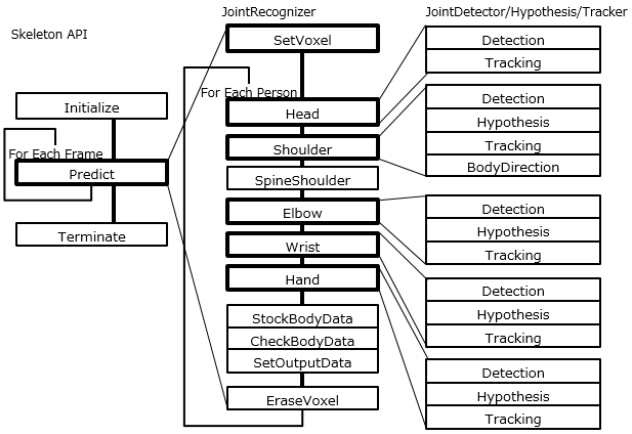


図 2　逐次探索型スケルトン抽出処理フロー
Figure 2　Skeleton Detection Sequence.

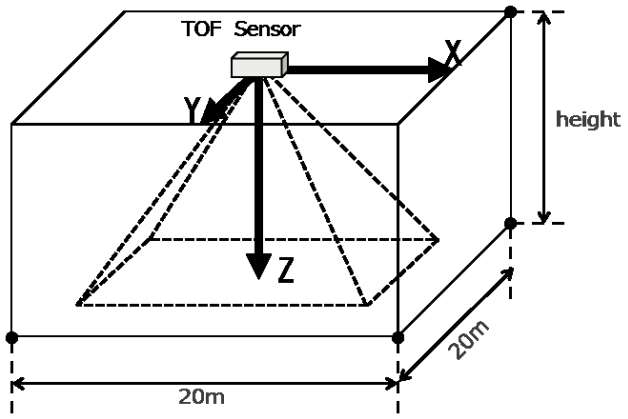


図 3　探索対象範囲
Figure 3　Skeleton Processing Range.

スケルトン抽出処理は，探索処理 Predict と，その準備処理 Initialize，および終了処理 Terminate，の 3 種類に大別できる．

探索処理は，TOF センサから取得される 1 フレームの距離データに対して，Predict が 1 回呼び出される．

Predict の中では，最初に 1 回だけ SetVoxel が呼び出されボクセルリストを生成する．そして，JointRecognizer で被写体 1 名分の点群データをボクセルリストから抽出し，各関節の探索処理を順次実行する．検出されたスケルトンデータは，Tracker のトラッキング用リストに蓄積される．

1 名分の探索が終了すると，EraseVoxel で探索に使用した 1 名分のボクセルデータをリストから消去し，残りのボクセルリストに対して，次の被写体 1 名分の探索を JointRecognizer で行う．これを，指定された最大被写体数まで繰り返し，検出できたスケルトン情報をすべて返して Predict は終了する．

TOF センサから次のフレームが取得されると，再度 Predict が呼び出され，上記と同様にスケルトン抽出処理が実行される．トラッキング用リストに十分なスケルトン情報が蓄積されていれば，過去のスケルトン情報をもとにした推定処理も各関節の探索時に実行される．

3.2 探索対象範囲

図 3 に， Predict の SetVoxel でボクセルリスト生成処理を行う探索対象範囲を示す．

対象範囲は，TOF センサを原点とした右手系の 3 次元座標空間において，X 軸と Y 軸の方向の最大範囲が，－10m から＋10m までの 20m 四方の広さであり，TOF センサの向きである Z 軸正方向に，TOF センサの設置高さと同じ高さ，を対象範囲とする．これは，上方からの見下ろし視点で，TOF センサのセンシング最大範囲を包含できる範囲である．

見下ろし視点から視点拡大して，水平視点や見上げる視点とする場合は，TOF センサよりも高い位置の点群も対象となるため，Z 軸負方向も対象とする必要がある．

3.3 距離画像のボクセル化

距離画像のボクセル化について説明する．

これは，後述する機械学習手法においても，特徴量を計算するための前提となるデータとして利用する．

探索処理および特徴量計算の前段階として，距離画像の各画素の 3 次元座標からボクセルデータを生成する．距離画像の各画素は，その点が存在する視野空間内の 3 次元座標を持つ．視野空間を小さな立方体(ボクセル，voxel)に区切り，その立方体が画素の座標を 1 つ以上含む場合にそのボクセルが存在するとみなす．図 4 に距離画像と，対応するボクセル群を可視化したイメージを示す．ボクセル化により，XYZ 座標で指定した空間内の点の有無を素早く判断できる．また，撮影距離が近い場合は，空間内で近接する複数の点が 1 つのボクセルにまとめられるため，処理対象のデータ量を削減できる．

立方体の 1 辺の長さ(ボクセルサイズ)は，25mm, 30mm, 35mm の 3 種類を事前実験した結果，関節座標推定結果の精度がよかった 25mm としている．

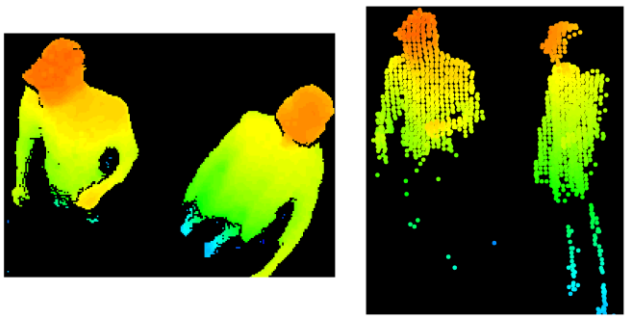


図 4　距離画像とボクセル
Figure 4　[Left] Depth image (range image), [Right] Voxels in 3D space.

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

3.4 関節探索手法

3.4.1 関節探索手法の概要

上方視点のスケルトン検出対象の関節は，頭，左肩，右肩，肩中点，左肘，右肘，左手首，右手首，左手，右手，の 10 関節とした．これは，逸脱動作検知のためには，頭と両手の座標を取得が必要なためである．

探索処理は，図 2 で示したように，関節ごとに，隣接ボクセル探索，人体モデル仮説に基づく推定処理，さらに，過去のフレームで取得されたスケルトン情報に基づく推定処理，という順で実行される．

以下，各関節の探索処理の概要を説明する．

3.4.2 頭探索

図 5 に，頭部探索の模式図を示す．

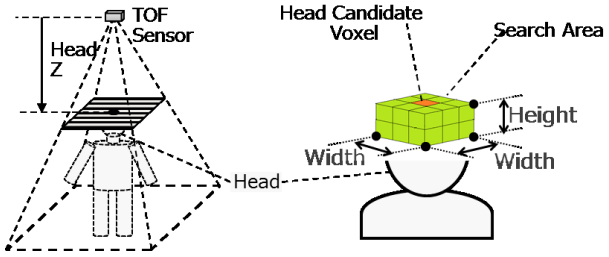


図 5　頭探索

Figure 5　Head Search.

探索対象のボクセルリストを，Z 座標でソートし，Z 最小値をもつボクセルを頭部ボクセルとする．

3.4.3 肩探索

次に，図 6 に，肩探索の模式図を示す．

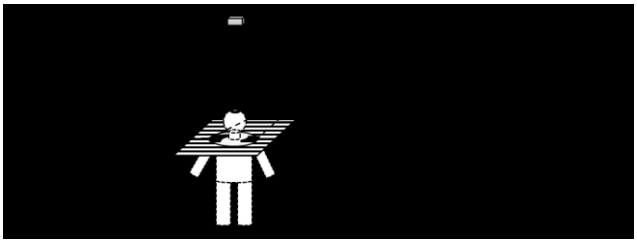


図 6　肩探索

Figure 6　Shoulder Search.

探索範囲は，Z 軸方向は，頭部の Z 座標から頭部の高さだけ下げた Z 座標を基準に一定の幅を持たせた範囲，X 軸および Y 軸方向は，頭部の XY 座標を中心とし，内径が頭幅の半分，外径が全肩幅の半分，という範囲の芯を抜いた円柱の領域である．

3.4.4 肘探索

図 7 に，肘探索の模式図を示す．

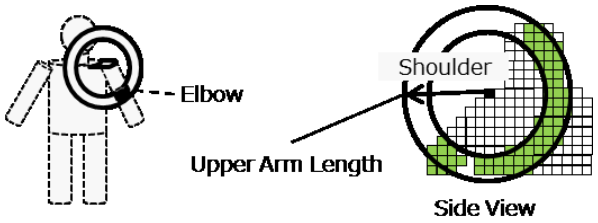


図 7　肘探索

Figure 7　Elbow Search.

探索範囲は，肩を中心とする，内径と外径の平均が上腕長である球殻の領域とする．

手首と手の探索は，肘探索と同様の処理となる．

3.5 結果

3.5.1 定量評価手法

上記のスケルトン検出処理を実装し，処理速度と関節座標精度を評価した．

距離画像は TOF センサを用いて上方視点で撮影した．センサから床までの距離は 2.75m，センサは視線が真下から 10 度持ち上げた方向を向くように設置している．被写体のモデルは直立した状態で，生産現場での模擬作業を行う．また，被写体は床上を横方向にセンサ位置を中心に最大±500mm, 奥行き方向にはセンサからの水平距離で 500mm の範囲で動き，体の向きは奥行き方向から左方向の間 90 度である．

関節座標精度の評価手法としては，すべての検出対象関節の座標データ（正解データ）とスケルトン検出処理により検出した関節の座標データを比較し，関節ごとの距離差を算出することで，距離差の大小を推定精度として評価した．

正解データは, 評価者が目視で距離画像上の関節位置（画素）を決定し，関節画素のもつ距離情報を 3 次元変換することで, 正解の 3 次元座標を得るという手順で, 生成した．正解データの全フレーム数は，157 フレームである．

3.5.2 関節座標推定結果の精度

表 1 に，スケルトン検出率を示す．

左腕（左肘，左手首，左手）の検出率が低いことが分かる．まずは，検出率の向上が必要である．

表 1　スケルトン検出率[%]

Table 1　Skeleton Detection Ratio [%].

Head	Shoulder		Elbow		Wrist		Hand	
	Right	Left	Right	Left	Right	Left	Right	Left
99	100	100	100	25	99	25	92	25

表 2 に，検出できたフレームに対するスケルトン検出精度を示す．

検出できたフレーム数を母数とした正解率は，誤差許容

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

範囲が 200mm の場合で，左腕が 70%台に留まっており，さらなる高精度化が必要である．

表 2　スケルトン正解率[%]
Table 2　Skeleton Prediction Correct Ratio for Detected Frames [%].

Head	Shoulder		Elbow		Wrist		Hand	
	Right	Left	Right	Left	Right	Left	Right	Left
100	73	99	90	98	79	92	77	93

3.5.3 処理時間の評価

上方から撮影したテスト画像に対して推定処理を実行した時の，1 画像あたりの処理時間を評価し，従来の処理時間と比較した．時間測定には比較的安価に入手できる一般的な PC と，TOF センサ内蔵の SoC 基板 を用いた．主な仕様を表 3 に示す．

表 3　評価に使用した PC と SoC の仕様
Table 3　Specification of PC and SoC used in this study.

Environment	CPU	RAM
PC	Intel Core-i7 6700 3.4GHz[b]	64GB
SoC	ARM Cortex™-A9 800MHz[c]	1GB

SoC 評価環境は，図 1 で示したシステム構成に対して，PC/Server 内の M/W として実装されていたスケルトン処理部を，TOF センサ内の TOF F/W として実装した環境となる．

表 4 に，スケルトン抽出処理時間を示す．

3.2ms/frame の処理速度となり，SoC 環境でリアルタイム処理実現の見込みが得られた．

表 4　スケルトン抽出処理時間
Table 4　Skeleton Prediction Speed.

Environment	Time[ms/frame]	Fps[frame/sec]
PC	6.5	153
SoC	47	21

4. 機械学習スケルトン抽出方法

4.1 従来のスケルトン抽出方法の概要と課題

研究着手当初に開発したスケルトン抽出処理の流れを図 8(a)に示す．全体の処理は，身体部位の推定と関節座標の推定の 2 つの処理に分けられる．さらに細分化すると，機械学習で生成した識別器に入力するための各画素の特徴量の生成，識別器による各画素の身体部位の推定，関節座標候補点の生成，関節座標の選択，という 4 段階で行われる．

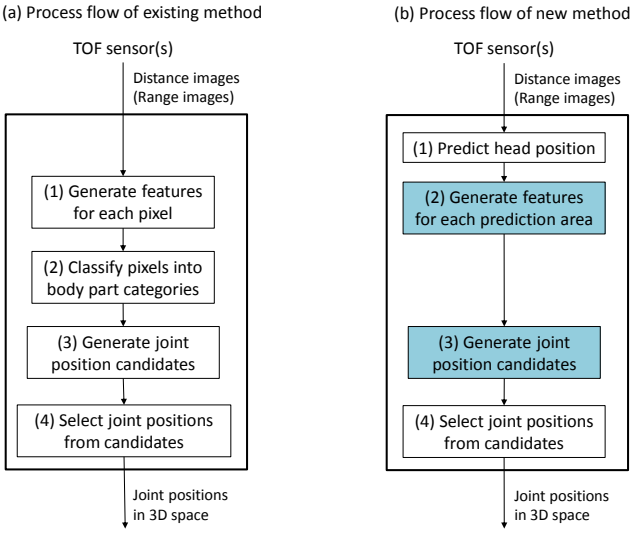


図 8　スケルトン抽出処理の流れ: (a)初期手法, (b)新手法
Figure 8　Process flow of skeleton recognition: (a) existing method, (b) new method.

最初の処理段階で生成される各画素の特徴量は，Kinect と同様，周辺画素間の距離差分である．差分を計算する周辺画素の組み合わせ，すなわち，特徴量の次元数は，1 画素につき 900 である．第 2 段階では，生成した特徴量を入力として，Random Forest による機械学習で作成した識別器で各画素の身体部位を推定する[7]．第 3 段階では，推定された各部位ごとに，ミーンシフト(Mean shift)[12]により画素の位置をクラスタリングし，3 次元空間内で画素が局所的に集まっている位置を 1 つ以上求める．最後に，ミーンシフトで集約された画素が最も多かった座標を関節座標として選択する．

従来の方式には下記のような問題点がある．

(1) 異なる撮影角度・距離への対応が困難

距離差分を計算する画素の組み合わせは，各画素の 2 次元画像としての並び方(画像内での XY 位置やピクセル単位の画素間隔)に基づいて選択される．しかし，センサの撮影角度や撮影距離が変化すると，同じ被写体の同じ身体部位でも画像平面上での見え方(形状や画素数)は大きく変化し，画素の並びは全く異なる．したがって，ある角度・距離に対して学習・調整した識別器を，異なる角度・距離で撮影された画像に適用することは難しい．また，側方からの撮影では，上方からの撮影と比べて腕と胴の各部位がさまざまな距離間隔で重なって映ることが多くなり，周辺画素との距離差分は複雑になる．そのため，部位ごとの特徴量の共通性が低下し，機械学習した識別器で解く部位推定問題が複雑化する．

b)　Intel, Intel Core, Xeon は，米国およびその他の国における Intel Corporation の商標です

c) Cortex™-A9 は，ARM Limited の商標または登録商標です

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

(2) **特徴量生成時間が長い**

すべての画素について特徴量を生成し，また特徴量の次元数が 900 と大きいため計算量が多い．推定処理のフレームレートが 30fps を下回る原因の 1 つであり，30fps を達成できる場合でも全処理時間の 2/3 を占めるほど大きな処理負荷となっている．

本研究では，上記の問題点を解決するために，2 次元画像上の見え方に依存しない推定方法を実現し，特徴量生成をはじめとする処理負荷を削減することが課題となる．以下，検討した方法について説明する．

4.2 3 次元位置情報を利用したスケルトン抽出処理の概要

本研究で提案するスケルトン抽出処理は，図 8(b)に示すように 4 つの段階からなる．

(1) **頭位置の推定**

最初に，頭の位置を機械学習に依存しない方法で推定する．具体的には，ルールベーススケルトン抽出手法と同様に, センサが検出した点群を床から高い方から順番に調べ，最も高い点の位置を頭の座標として認識する．推定した頭の位置は，第 2 段階で生成する一部の特徴量の処理で利用する．

(2) **機械学習で生成した識別器に入力する特徴量の生成**

特徴量は画素単位ではなく，いくつかの画素をまとめた推定領域単位で生成する．これより特徴量生成時間を短縮する．特徴量の値は，各画素の 2 次元画像内での位置関係ではなく，撮影の角度・距離が変化しても(センサに映っている限りは)変化しない 3 次元空間内での位置関係に基づいて計算する．これにより，側方からの撮影および上方からの撮影の両方に対応する．また，各推定領域の頭座標との位置関係や，センサとの位置関係に関する情報も特徴量に取り込む．

(3) **関節座標候補集合の生成**

生成した特徴量を入力として，機械学習を利用して生成した識別器により各推定領域と関節との位置関係(近さ)を推定する．関節に近い推定領域群の座標に基づき，関節座標の候補点群を生成する．従来の方法の第 2 段階で行っていた各画素の身体部位の推定を不要化することにより，推定処理時間を短縮する．

(4) **関節座標の選択**

各関節の候補点を組合せ，関節の位置関係が身体構造として有効であることを確認し，無効な組合せは除外する．

次節以降では，第 2 段階の特徴量生成と第 3 段階の関節座標候補集合の生成について詳しく説明する．

4.3 3 次元位置情報に基づく特徴量とその生成方法

本節では，機械学習で生成した識別器が扱う特徴量とラベルについて説明する．

4.3.1 推定領域とラベル

ボクセル化した空間を区切る 1 辺 N ボクセルの立方体空間を「推定領域」と呼ぶ．推定領域は機械学習を利用した識別器の学習・推定の単位となる．学習の際には，各推定領域の特徴を表す特徴量とラベルの対応付けを学習させる．推定の際には，推定領域の特徴量を識別器に入力し，出力としてその推定領域に対応するラベルを得る．

推定領域のラベルは，その推定領域と関節の「近さ」を表す. 関節種別毎に, 推定領域の中心座標がその関節に「該当」，その関節に「近い」，関節から「遠い」，の 3 種類のラベルのいずれかを付ける．両手，両肘など左右 2 関節が 1 つの推定領域で重なることがある関節種別については，2 関節に該当，1 関節に該当し別の 1 関節に近い，2 関節に近い，1 関節だけに該当，1 関節だけに近い，いずれの関節からも遠い，の 6 種類のラベルを付ける．

従来の方式では関節(身体部位)の左右を区別してラベルを付けていた．例えば，右肩と左肩には異なるラベルを付けていた．しかし，本方式では左右を区別しないことにより推定問題を簡単化し，左右の識別は第 4 段階の候補点の選択処理内で行うこととした．

4.3.2 推定領域の特徴量

推定領域の特徴を表す特徴量は，推定領域内のボクセル配置の形状に基づくものと，ボクセル形状以外の情報に基づくものの 2 種類を含む．

(1) **ボクセル配置の形状に基づく特徴量**

関節を構成するボクセルの配置は，関節毎に異なる場合が多いと考えられる．例えば，肩関節は比較的大きな丸みを帯びた形状で，手の手首に近い部分は細い円柱状，指先に近い部分は先端が途切れている，などの特徴がある．推定領域内のボクセル配置の形状を区別するために，形状を数値化して推定領域の特徴量に含める．

新しい特徴量では，推定領域内のより小さな立方体領域を「セグメント」と定義し，セグメント内のボクセル数を数える．セグメントの位置をボクセル単位で XYZ 方向にずらしながら，推定領域内で取り得るすべてのセグメント位置についてセグメント内に含まれるボクセル数を数え，カウントされた各ボクセル数とセグメント数のヒストグラムを作成する．

図 9 に例を示す. 説明を簡単化するため 2 次元で説明する．図は推定領域サイズ(1 辺のボクセル個数)が 3，セグメントサイズ(1 辺のボクセル個数)が 2 の例である．図(A)の推定領域(大きな黒い正方形)の中でセグメント(小さな赤い正方形)を 1 ボクセル単位で動かし, 小さな正方形の各位置で含まれるボクセル数をカウントする．この場合，セグメントの位置は a-d の 4 種類がある．推定領域内でボクセルが図(B)のように配置されているとき，a，c のセグメント位置ではセグメント内に含まれるボクセル数は 0，d の位置では 1，b の位置では 2 となる．ボクセル数 0 のセグメント位置が 2 つ，ボクセル数 1 および 2 のセグメント位置が 1 つずつであるから，(B)に示す推定領域の特徴量は(C)のヒストグラムとなる．

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

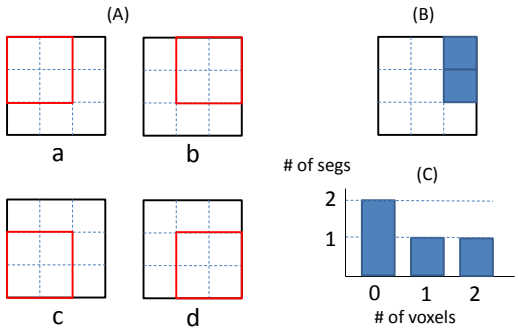


図 9　特徴量(2D)　(A)推定領域とセグメント，(B)推定領域中のボクセル，(C)特徴量

Figure 9　Example of new features (2D).　(A) Prediction area and segment,　(B) Example of voxels in prediction area,　(C) Feature of prediction area shown in (B).

このヒストグラムにより，推定領域内のボクセル個数が同一であっても配置形状の違いを認識することができる．

本報告では，セグメントサイズとして 2，3，4 の 3 種類を用い，それぞれ 5，10，13 次元の合計 28 次元の特徴量でボクセル配置の形状を表している．

(2)　**ボクセル配置の形状以外の情報に基づく特徴量**

ボクセル配置の形状を表す 28 次元に加え，それ以外の情報を表す下記 3 つの特徴量を利用する．これらは，身体構造の特徴や，撮影条件に関する情報を織り込むことを狙った特徴量である．

1. **推定領域と床の距離**
2. **推定領域と頭の水平距離**
3. **推定領域とセンサの距離**

上記の 3 次元を加え，全体で 31 次元の情報を新しい特徴量として用いる．

従来の 900 次元の特徴量と比べると，次元数が約 30 分の 1 に低減されている．また，従来は画像の各画素について特徴量を計算していたが，複数の画素(ボクセル)をまとめた推定領域ごとの計算となるため，処理量を減らすことができる．新旧の方法で利用する特徴量とラベルを表 5 にまとめる．

表 5　特徴量とラベル

Table 5　Summary of feature and label of existing and new method.

	Former method	New method in this study
Unit of feature	Pixel	Prediction area (a 6-voxels cube space)
Dimension of feature	900	31
Feature based on	Difference of pixel depth	Voxel position in prediction area, distance from head, and distance from camera

次節では，新しい特徴量を用いて関節候補点の集合を生成する具体的な方法を説明する．

4.4　関節座標候補の生成

識別器による推定領域のラベル推定処理と推定結果に基づく候補点集約処理を関節種別毎に行うことにより候補点集合を生成する．

まず，前節で説明したように，与えられた距離画像をボクセル化し，空間を推定領域に区切る．

切り出した各推定領域の特徴量を計算し，関節種別毎の識別器に入力してラベルを得る．ただし，含まれるボクセルが少ない推定領域は，特徴を反映できる十分な情報量がないため学習・推定の対象から除外し処理量を削減する．

特徴量を各関節の識別器に入力して各推定領域のラベルを取得し，その結果に基づいて関節種別ごとに候補点の集合を生成する．

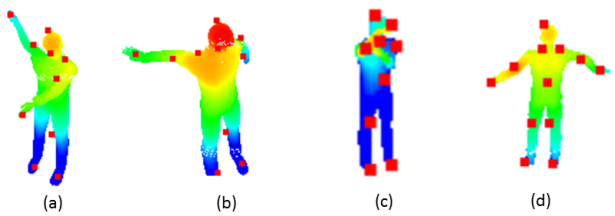
4.5　結果

4.5.1 推定処理の評価

提案した方法を実装し，推定精度を評価した．距離画像はすべて CG ツールで生成し，センサから床までの距離を 2.5m，センサは視線が水平から 20 度下に向くように傾けて設置している．被写体のモデルは直立した状態で腕を様々な位置に動かしている．また，被写体は床上を横方向にセンサ位置を中心に最大±50cm，奥行き方向にはセンサからの水平距離が 1m-4m の範囲で動き，体の向きは 360 度自由に変える．

各関節の識別器の学習に使用した距離画像は 2500 枚，学習したサンプル数(推定領域数)は約 1.1 億サンプル，総ボクセル数は約 14 億である．特に断りがない限り，推定対象のテスト画像は，学習対象には含まない上記条件の距離画像 200 枚を使用した．

推定結果の例を図 10 に示す．赤い点が各関節として推定された位置である．(a)(b)はセンサからの距離が 2-3m，(c)(d)は 4-5m の例である．



Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

図 10　関節位置の推定結果（例）
Figure 10　Examples of predicted joint positions.

4.5.2 識別器の推定精度

図 11 に関節種別ごとのラベル推定精度を示す. 青グラフの TRUE (Strict)はラベルが正解と一致した割合，赤グラフの TRUE (Correct joint)はラベルの「該当」と「近い」の違いをなくした場合の一致率である．後者は，関節に対する近さは区別できないが，そのいずれかであることがわかる割合を意味する．図示していないラベルは，テスト画像では検出されなかった．

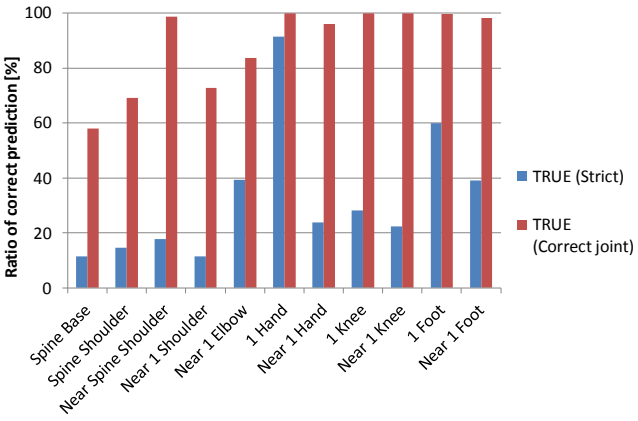


図 11　関節毎のラベル推定精度
Figure 11　Accuracy of label prediction for each label type.

実験結果から，ラベルの「該当」と「近い」の取り違えが多く発生し，関節からの距離の区別は難しいものの，腰以外の関節については 70%以上の精度で関節に近いことが推定できるという結果となった．特に高い精度を示した手関節に関しては，新しい特徴量を使った推定が非常に有効であることがわかる．

4.5.3 候補座標の推定精度

推定したラベルと各推定領域の座標に基づいて生成した候補点の精度を図 12 に示す．ここでは，産業分野向け逸脱動作検知システムへの適用を想定し，候補点が関節から 200mm 以内の距離にある場合に正解としている．実験結果から，生成した候補点のうち，腰以外では 8 割以上の精度で正しく候補点を生成できていることがわかる．

肩の候補点精度(82.9%)はラベル推定精度(72.8%)よりも高くなっている．これは，ラベル推定とミーンシフトの組み合わせによる効果と考えられる．すなわち，関節からやや遠い不正解座標がミーンシフトの入力に含まれていても，正しく推定された座標が入力中に多数ある場合にはミーンシフトによって少数の誤りが多数の正解座標に集約されることが期待できる．

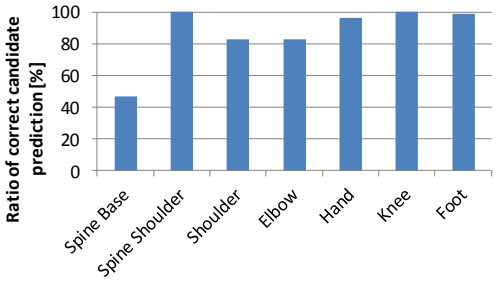


図 12　候補座標の推定精度
Figure 12　Accuracy of candidate position for each joint type. (Margin = 200mm).

4.5.4 関節座標推定結果の精度

候補座標から最終的に選択した関節座標の精度を図 13 に示す.

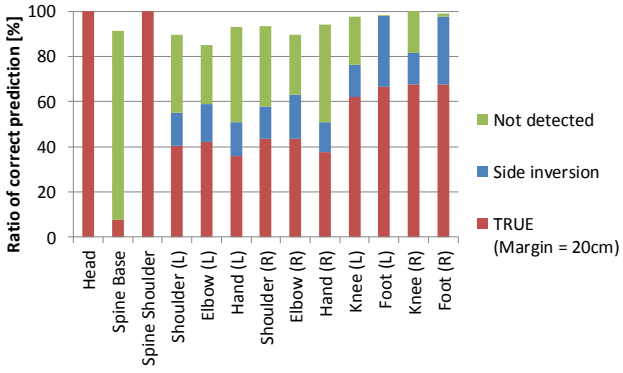


図 13　関節座標の推定精度
Figure 13　Accuracy of predicted joint position.

誤差 200mm 未満で正しく推定できた関節位置の割合は，多くの関節で 40-60%程度となっていて高くない. それ以外の場合は，関節位置が未検出，あるいは誤った位置を検出している．原因として以下の要因が考えられる．

1. **画像に関節が写っていない**
   提案した方式では，画像に写っているボクセルに基づいて関節位置を推定している．したがって，もともと画像に写っていない関節については推定できない. その場合，関節は未検出となる．
2. **左右の関節を取り違えている**
   グラフの青部分(Side inversion)は，誤った推定位置のうち，左右の関節を入れ替えると正解になるものの割合を示している. 今回の実験では左右判定のロジックが未熟なため，左右の取り違えが最大で 30%以上発生している．
3. **正しい候補位置を除外している**
   候補は正しく生成できているが，候補点の選択ロジックが未熟なため，選択段階で除外されてしまう場合が

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

ある．図 14 の例では，左側に示すように左手の候補位置(赤点部)は正しく生成されているが，右側の最終的な推定結果には含まれていない．これは未検出率の高さの一因と考えられ，左右判定と同様，候補選択のロジックを改善する必要がある．

4. 候補点の精度が悪い

図 12 で示したように，腰の候補点は精度が悪く，最終的な推定結果では多くが無効点として除外され，未検出が多くなっている．

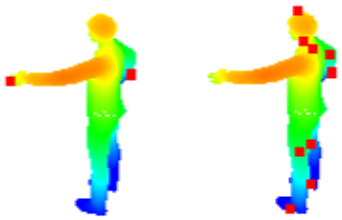


図 14　候補位置（例）: [左] 2 つの手候補，[右]選択された関節位置

Figure 14　Example of candidate selection. [LEFT] Two candidates of hand position,　[RIGHT] selected joint positions.

今回の評価結果では最終的な関節位置の推定精度は高いとは言えないが，未検出でも左右誤認でもない，まったくの誤推定はすべての関節で 2 割未満と比較的少ないため，候補集合からの選択処理の改善と，今回の評価では使用していない時系列情報の利用(過去の関節位置情報の利用)により，精度向上の可能性は十分にあると考える．

4.5.5 関節座標推定結果の精度

逸脱動作検知システムでは，左右誤認の影響を除外でき，また，未検出による検出率低下よりも，検出できたスケルトンの精度（正解率）のほうが重要である．

表 6 に，上半身関節のスケルトン検出率を示す．
いずれの関節も検出率 50-60%であり，改善の余地がある．

表 6　スケルトン検出率[%]

Table 6　Skeleton Detection Ratio [%].

Head	Shoulder		Elbow		Hand	
	Right	Left	Right	Left	Right	Left
100	58	55	62	58	50	50

表 7 に，検出できたフレームに対する上半身関節のスケルトン検出精度を示す．

検出できたフレーム数を母数とした正解率は，左肘が 70%台である以外は 80%台であり，正解率 90%以上に向けてはさらなる高精度化が必要である．

表 7　スケルトン正解率[%]

Table 7　Skeleton Prediction Correct Ratio for Detected Frames [%].

Head	Shoulder		Elbow		Hand	
	Right	Left	Right	Left	Right	Left
100	87	84	86	77	89	86

4.5.6 センサ角度および撮影距離の変化への対応

ここまでの評価と同じ識別器を使い，異なるセンサ設置条件で撮影した画像を対象に推定精度を評価した．評価対象画像は，センサから床までの距離は 3m，センサ角度は鉛直下方で，被写体はセンサ位置を中心に床上を縦横方向に最大±1m 動くという条件で撮影した．図 15 に示した画像例でわかるように，これまでとは人物の見え方が全く異なる．この撮影条件の画像は識別器の学習対象には全く含まれていない．

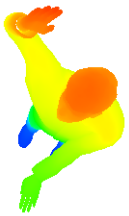


図 15　上方視点のテスト画像（例）

Figure 15　Example of test image (vertical view).

図 16 のラベルの推定結果を見ると，手のように認識率が 60%未満に低下した関節もあるが，肩や両肩の中点(spine shoulder)のように精度向上している関節もあり，全体としては前項の結果と同様，関節に近いことは高い割合で認識できている．このことから，新しい特徴量はセンサの設置角度や撮影距離によらず汎用的に適用できることがわかる．

また，図 17 に示す通り，ラベル推定とミーンシフトの組み合わせによる候補集合の生成は上方視点でも有効であることがわかる．

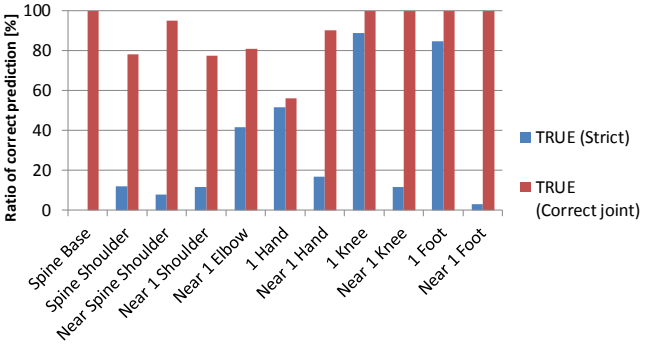


図 16　上方視点のラベル推定結果

Figure 16　Accuracy of label prediction for each label type (vertical view).

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

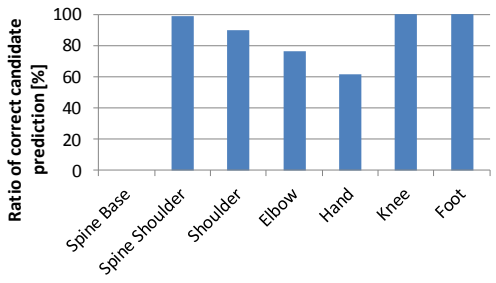


図 17　上方視点の各関節の候補位置
Figure 17　candidate position for each joint type (vertical view).

図 18 に示す関節座標推定結果の精度では，膝，足の未検出が増えているが，これは上方視点では下半身が見えにくいことが影響していると考えられる．また，左右誤認の割合が増え，未検出や左右誤認以外の誤りが手関節を中心に+10%程度増えているが，極端な増加ではなく本方式が上方視点にも適用可能であることがわかる．特に誤推定の多い手関節(右手)は，前段処理のラベル推定結果および候補点の精度が低いことから，識別器の学習対象に上方視点の画像を加え，これらの精度を向上させることで最終的な推定結果の精度も改善できると考えられる．

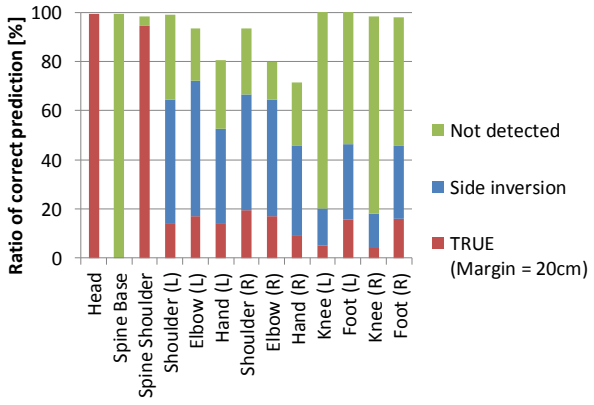


図 18　上方視点の関節座標の推定結果
Figure 18　Accuracy of predicted joint position (vertical view).

表 8 に，上半身関節のスケルトン検出率を示す．

表 8　スケルトン検出率[%]
Table 8　Skeleton Detection Ratio [%].

Head	Shoulder		Elbow		Hand	
	Right	Left	Right	Left	Right	Left
100	65	63	63	72	45	52

逸脱作業検知システムの対象となる，生産設備とセンサが固定された生産現場環境では，被写体の正面が設備に向かう方向に限定され，その前提を利用することで左右誤認による影響を除外できるため，検出できたフレームに対する正解率は表 9 のようになる．

表 9　スケルトン正解率[%]
Table 9　Skeleton Prediction Correct Ratio for Detected Frames [%].

Head	Shoulder		Elbow		Hand	
	Right	Left	Right	Left	Right	Left
99	89	98	75	90	60	86

4.5.7 処理時間の評価

上方から撮影したテスト画像に対して推定処理を実行した時の，1 フレームあたりの処理時間を評価し，従来の処理時間と比較した．従来方式と比較するために上方から撮影した画像を対象に評価したが，側方から撮影した画像に対しても同様の結果が得られている．

時間測定には比較的安価に入手できる一般的な PC を用いた．主な仕様を表 10 に示す．

表 10　時間測定用 PC 仕様
Table 10　Specification of PC used in this study.

Environment	CPU	RAM
PC	Intel Xeon CPU E3-1220v5 3GHz[b]	16GB

200 枚の画像に対する評価を 4 回繰り返しその平均値をとった結果を図 19 に示す．

本方式の推定処理時間は全体で 9ms 程度という結果となった．従来と比べると，位置推定する関節の数が 7 から 13 にほぼ倍増しているにもかかわらず，1/3 以下に処理時間を短縮することができた．計算上は 110fps までサポートできる処理時間であり，目標フレームレートの 30fps を達成することに加え，3 台のセンサから送られる画像を同時に 30fps のフレームレートで処理することが可能となる．

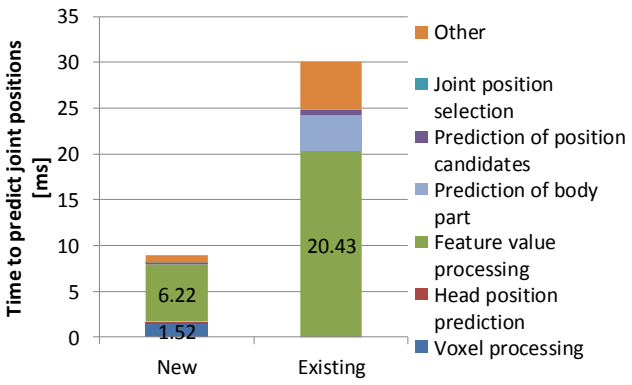


図 19　スケルトン抽出の各処理の平均時間
Figure 19　Average time of each process in skeleton

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

recognition.

課題であった特徴量の処理時間は 6ms 程度に高速化できており，新たに追加されたボクセル化処理の時間を含めても従来の特徴量処理時間(20.43 ms)の 1/3 程度に短縮することができた.

精度の評価で改善が必要であることがわかった候補選択の処理時間(Joint position selection)は 0.05ms 程度と短く，今後，精度向上のために処理量を増やす余地は十分にあると考える.

5. 既存技術との比較

5.1 比較結果の概要

表 11 に本研究手法と, 逸脱検出システムで利用している Kinect との比較を示す．Kinect では側方視点から撮影した距離画像により，スケルトン(身体姿勢)以外にも手の形状や顔の表情をある程度認識することが可能である.

一方，本研究では上方視点から撮影した距離画像によりスケルトン抽出が可能であり，差別化ポイントとなっている．加えて，機械学習手法では側方視点から撮影した場合でもスケルトン認識が可能となり，側方からの撮影を前提とした環境にも適用できる可能性が出てきた．また，処理時間短縮により 30fps 以上のフレームレートでの処理が可能となった.

ルールベースの逐次探索手法では TOF センサの SoC にて正解率 75%でリアルタイムに姿勢推定可能なことと，機械学習手法では一般的な CPU にて正解率 60%でリアルタイムに姿勢推定可能なことが確認できた.

表 11　Kinect 比較概要
Table 11　Comparison to Kinect Overview.

Comparison items		Objective of this research		Kinect
		Rule-based	Machine Learning	
Depth Data		Captured by TOF Sensor	Created by CG tool	Captured by Kinect
Output of skeleton data (joint position data)		OK	OK	OK
Frame rate		More than 150 fps	More than 100 fps	up to 30 fps
For vertical view	Recognize human body posture	OK	OK	NG
	Correct Ratio	75%	60%	N.A.
For side view images	Recognize human body posture	NG	OK	OK
	Correct Ratio	N.A.	75%	75%
	Recognize hand posture	NG	NG	OK
	Recognize face expression	NG	NG	OK

5.2 比較結果の詳細

5.2.1 検出精度の比較

ルールベース手法について, 詳細に比較した結果を示す.

表 12 に，検出できたフレームに対する，本研究のルールベース手法の検出精度と，Kinect のスケルトン検出精度との比較を示す.

200mm 以下の距離差を正解とみなす条件で，Kinect の正解率は左腕の関節が 20%未満であり，逆に右腕の関節は 70%以上である．これは，Kinect が被写体の右側側方から撮影したため，左側の関節が隠れているフレームが多かったためと推測される.

右腕の関節で比較すると，同等以上の正解率であるといえる．なお，両者とも頭の検出率は 100%であった.

表 12　正解率比較
Table 12　Comparison (Correct Ratio for Detected Frames[%], <200[mm]).

Method	Shoulder		Elbow		Wrist		Hand	
	R	L	R	L	R	L	R	L
Kinect	74	17	80	17	75	16	76	17
This Study (Rule-based)	73	99	90	98	79	92	77	93

5.2.2 処理負荷の比較

本研究のルールベース手法の検出処理負荷と，Kinect の処理負荷を比較した.

時間測定には比較的安価に入手できる一般的な PC を用いた．主な仕様を表 13 に示す.

表 13　時間測定用 PC 仕様
Table 13　Specification of PC used in this study.

Environment	CPU	RAM
PC	Intel Core-i7 6700 3.4GHz	64GB

表 14 に本研究の TOF スケルトン検出処理負荷と, Kinect

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

の処理負荷の比較を示す.

グラフィック描画処理による負荷分を除外するため，グラフィックを OFF にした状態で比較すると，平均 CPU 占有率で約 14 分の 1，メモリ消費量で約 6 分の 1 のリソースで済む.

表 14　処理時間比較
Table 14　Comparison (Performance).

Graphic	Method	CPU	RAM
On	Kinect	37%	280MB
	This Study (Rule-based)	17%	45MB
Off	Kinect	20%	260MB
	This Study (Rule-based)	1.4%	42MB

以上をまとめると，Kinect との比較においては，スケルトン検出正解率は同等以上，処理負荷はより低く，産業用途に適しているといえる.

5.2.3 他の姿勢推定技術

Kinect 以外の姿勢推定技術としては, 例えば OpenPose が挙げられる[13]. 距離画像を必要とせず，2D の画像から高精度に人の姿勢推定が可能なオープンソースソフトウェアである.

OpenPose は，深層学習による推定処理を行っており，CPU,GPU やメモリなど, Kinect 以上にリソースが必要という欠点が現状ではあるが，今後は，ハードウェアの高性能化やクラウドコンピューティングと組み合わせたソリューションへの展開に伴い，深層学習による高精度な姿勢推定技術が普及する可能性が高いと考えられる.

6. おわりに

6.1 結論

上方からの見下ろし視点で撮影された距離画像に適用可能なスケルトン抽出手法を提案し，実装・評価した.

(1) 従来の側方視点の機械学習方式ではなく, 上方からの見下ろし視点の距離画像の特徴に基づいた, ルールベースの逐次探索手法を提案した.
(2) 本手法で手検出率 75%以上を達成し, 逸脱動作検知システムに適用可能な見込みを得た.
(3) 機械学習方式に比べて，50 倍以上の処理速度を実現し，TOF センサ内の SoC にスケルトン検出処理を内蔵しリアルタイム処理を実現できる見通しを得た.

また，さまざまなセンサの角度・距離で撮影された距離画像に適用可能で，従来よりも処理時間を短縮した新しいスケルトン抽出方式を提案し，実装・評価した.

(4) 提案した推定方法により, 学習データと同じセンサ設置角度・距離で撮影した画像に対してはほとんどの関節について約 75%以上の精度で, 学習データとは異なる視点の画像に対しても約 60%の精度で関節位置を推定できることを示した.
(5) 推定処理時間は，当初手法と比べて 1/3 以下の 1 画像あたり平均約 9ms で，目標の 30fps を十分に達成できることを示した.
(6) 生成した候補点集合から関節座標を選択する段階で, 正しく推定した候補点を誤って除外することや, 左右の関節を取り違えることが多く, 改善が必要であることが分かった.

6.2 今後の課題

ルールベース手法の課題は以下のとおりである.

(1) 探索ロジックの改善によるスケルトン検出精度向上および検出率向上
(2) 視点拡大と被写体多様性への対応によるスケルトン検出の汎用化

機械学習手法の課題は以下のとおりである.

(3) 候補点の選択ロジックの改善による精度向上
(4) 関節座標トラッキング(過去の関節位置情報の利用)による精度向上および未検出関節の補完
(5) 実際のセンサ出力データに含まれるノイズに対する耐性の検証
(6) 正解率，処理負荷の定量評価

参考文献

[1] Worldwide Internet of Things 2016-2019 Forecast: Market Opportunity by Region and Narrowing the Lens on Use Cases, IDC, 2016
[2] Digitalization in machine building - The digital twin (http://www.siemens.com/customer-magazine/en/home/industry/digitalization-in-machine-building/the-digital-twin.html)
[3] 'Digital Twin' Technology Changed Formula 1 and Online Ads. Planes, Trains and Power Are Next (http://www.gereports.com/digital-twin-technology-changed-formula-1-online-ads-planes-trains-power-next/)
[4] Kinect ハードウェア (https://developer.microsoft.com/ja-jp/windows/kinect/hardware)
[5] Kinect for Windows SDK (https://msdn.microsoft.com/ja-jp/library/dn799271.aspx)
[6] 現場作業員の逸脱動作や設備不具合の予兆を検出する画像解析システムを開発 (http://www.hitachi.co.jp/New/cnews/month/2016/07/0713.html)
[7] J. Shotton, et al., 'Real-Time Human Pose Recognition in Parts from a Single Depth Image', IEEE CVPR, 2011
[8] 岩井儀雄, モデルベース手法による身体動作計測, 情報処理学会研究報告コンピュータビジョンとイメージメディア（CVIM）,2001(66(2001-CVIM-128)),73-80 (2001-07-05)
[9] Ganapathi, V., Plagemann, C., Koller, D. and Thrun, S., Real-time human pose tracking from range data, European conference on computer vision, Springer, pp. 738-751 (2012).
[10] Schwarz, L. A., Mkhitaryan, A., Mateus, D. and Navab, N., Human skeleton tracking from depth data using geodesic distances and optical flow, Image and Vision Computing, Vol. 30, No. 3, pp. 217-226 (2012).

Vol.2017-CG-168 No.29
Vol.2017-DCC-17 No.29
Vol.2017-CVIM-209 No.29
2017/11/9

[11] Hitachi-LG Data Storage, SDK(ソフトウェア開発キット) v2.1.0 (http://hlds.co.jp/product/tofsdk)
[12] Y. Cheng, ‘Mean shift, mode seeking, and clustering’, IEEE PAMI, 1995
[13] OpenPose (htptps://github.com/CMU-Perceptual-Computing-Lab/openpose)