

複数 Kinect の姿勢データの時空間統合による モーションキャプチャシステム

森本 隆志^{1,a)} 満上 育久^{1,b)}

概要：本稿では、非同期な複数台の Kinect を用いて姿勢推定を行い、得られた姿勢データを時空間的に統合することによって、正確なモーションキャプチャを実現する手法を提案する。提案手法では、非同期で動作する各 Kinect から得られる姿勢データに対してリサンプリングを行い、疑似的に時刻同期されたデータ系列を生成する。また、各 Kinect から得られる姿勢データ中には、人との相対位置関係によって位置を大きく誤って推定された関節点が含まれたり、人の前後を誤ったものを含んでおり、これらから適切な情報のみを選択し統合することで、安定で高精度な全身姿勢を推定する。提案手法の有効性を確認するため、実験室に 4 台の Kinect を設置した環境内で動く人の全身姿勢を推定する実験を行い、提案手法の有効性を確認した。

1. はじめに

人の全身運動を三次元的に計測するモーションキャプチャシステムは、人の運動の計測や解析、映画やゲームの製作で CG キャラクターへモーションを付与するなどの目的で幅広く利用されているが、多くのシステムでは、計測対象人物が特殊なスーツを着用しなければならなかったり、全身にセンサを装着しなければならないなどの制約がある。また、多くの高価な機材を必要とするため、準備の大変さや見栄えの悪さ（に伴う利用シーンの制限）、着用物による違和感などの問題がある。

一方、特殊なスーツやセンサを着用する必要のないモーションキャプチャ手法も存在し、Microsoft Kinect（以下、Kinect）は、2 万円ほどと安価ながら、リアルタイムに人の全身姿勢を計測できるデバイスであり、人の動作分析に関する多くの研究で使用されている [1],[2]。

ただし、Kinect によるモーションキャプチャには自己遮蔽による姿勢推定性能の悪化という問題が存在する。Kinect は内蔵するデプスセンサにより取得される人物のデプス画像に対して姿勢を推定するが、図 1 に示すように、Kinect と人との相対位置関係によって人の自己遮蔽によりデプス画像には観測されない部位が生じ、その部位の関節位置は正しく推定することができない。

そこで本研究では、1 台の Kinect のみでは回避できない

自己遮蔽の問題を解決するために、複数の Kinect を人の周囲に設置してさまざまな方向から人を観測することで、人の位置・姿勢に関わらず自己遮蔽の影響を受けずに全身姿勢を推定できるモーションキャプチャ環境の構築を目指す。単純には、各 Kinect の相対位置・姿勢を事前のキャリブレーションにより獲得しておき、各 Kinect で得られた姿勢データを共通の世界座標系に変換して統合することで実現できるが、実際にはいくつかの技術的な難しさが存在する。

1 つは、複数台の Kinect 間での撮影タイミングの非同期性である。複数台の Kinect は非同期で撮影を行うため、動きのある人物を撮影する場合、撮影されるタイミングが各 Kinect で異なるため、わずかに異なる姿勢を計測してしまう。そこで提案手法では、得られた姿勢データをそのまま使用するのではなく、撮影時刻情報も用いてリサンプリング処理により仮想的に同期された姿勢データを生成することでこの非同期性を解決する。また、2 つ目の問題として、Kinect が人の前後を正しく判定できないということがある。Kinect はもともと Xbox の周辺機器としてゲーム用に発売されたセンサーであるため、人が Kinect に対して向き合って立っている状況しか想定されていない。そのため、人が Kinect に対して背を向けている場合、Kinect は人の背面を正面だと誤って認識してしまう傾向があり、その結果、右腕や右脚を左腕や左脚だと誤認して反転したデータを出力してしまう。そこで提案手法では、複数 Kinect 間で観測された姿勢データを共通の座標系にマッピングした際に、同時にそれらを左右反転させた姿勢データも生成・

¹ 広島市立大学
Hiroshima City University

a) morimoto@sys.info.hiroshima-cu.ac.jp

b) mitsugami@hiroshima-cu.ac.jp

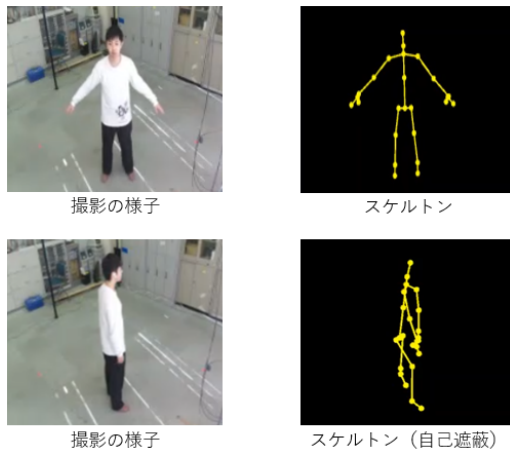


図 1 Kinect を用いた骨格推定の様子（上段左：撮影の様子 [人を正面から撮影]，上段右：人のスケルトン，下段左：撮影の様子 [人を側面から撮影]，下段右：自己遮蔽発生時の人のスケルトン）

マッピングし，それらの姿勢データ群に対してコンセンサスのとれた組み合わせを選択した上で統合することで，この問題についても解決する。

提案手法の有効性を確認するため，実験室に 4 台の Kinect を設置した環境を構築し，その環境内で動く人物の姿勢の推定を行う実験を実施した．各 Kinect は非同期で動作し，得られる姿勢データには人との相対位置・姿勢関係に応じて大きな誤差を持つ関節点が存在するが，提案手法によって統合された姿勢データはそれらの誤った関節点は排除され，適切に全身姿勢が推定されることを確認した。

2. 関連研究

Zennaro らが，Kinect の特徴について詳しく述べている [3]．Kinect には，初期型の KinectV1 と，後期型として発売された KinectV2 がある．KinectV1 は，1 台の PC に 4 台まで接続可能だが，赤外線干渉の影響を受けやすいといった問題点を Butler らが報告している [4]．KinectV2 は，単一 PC で複数台使用することができないものの，Time-of-Flight (ToF) という赤外線の反射時間から奥行きを推定する原理を用いていることから干渉の影響を受けにくい．本研究では KinectV2 を使用する．なお，KinectV1 と KinectV2 の詳しい性能の違い（温度やシーンの色等が Kinect に与える影響）については，Wasenmüller らの報告を参照されたい [5]．

Kinect は距離データや RGB データを取得できるため，センサとしての利用はもちろん，そもそもゲームのコントローラとして開発されているため，ユーザインタフェースとしても使用できる [6]．ユーザインタフェースとしては主に人の骨格推定に使用されており [7],[8]，従来のモーションキャプチャに代わり，Kinect を使用したさまざまな手法が複数提案されている。

複数台の Kinect を用いた研究としては，Nakazawa ら

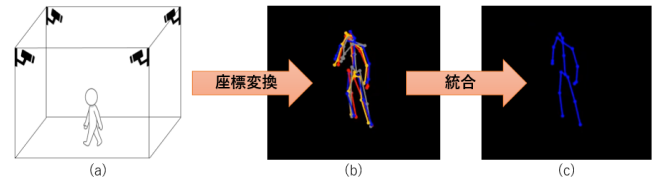


図 2 提案手法のアウトライン：(a) 複数台の Kinect で撮影 (b) 疑似的に同期された姿勢データ系列を生成，各姿勢データを共通の座標系にマッピング (c) スケルトンの統合

は，複数 Kinect のデプス画像とカラー画像を用いて環境全体のテクスチャ付き三次元形状復元を行う手法を提案している [9]．また，彼らの研究グループでは，本研究で取り扱う複数 Kinect の非同期性の問題についても既に取り扱っている [10]．ただし，彼らはデプス画像のモーフィングにより仮想的に時刻同期のとれたデプス画像を生成するアプローチを採用しており，パラメトリックな姿勢データを取り扱っていないという点で，本研究と異なる。

Kinect を用いず複数カメラを用いてモーションキャプチャを行うものとして Joo らの Panoptic Studio [11] がある．これは，同グループにより提案された単眼カメラ画像に対する姿勢推定手法 OpenPose [12] を用いて，複数のカメラから得られる姿勢データを統合するものだが，同期撮影可能なカメラを多数設置しなければならない，その環境構築は一般に容易ではない．本研究の提案システムは，この環境構築の容易性という点で優位性である。

3. 提案手法

複数台の Kinect を用いたモーションキャプチャシステムの構築を行う提案手法の処理の流れを図 2 に示す．図 2(a) のように複数台の Kinect を人の周囲に設置し，さまざまな方向から観測を行うことで，人の位置・姿勢に関わらず各関節点が常にいずれかの Kinect で観測できるようにし，それによって安定に全身姿勢が推定されるようなモーションキャプチャ環境の構築を行う。

空間的なキャリブレーションとして，事前に複数の Kinect で静止している人の撮影を行い，得られた姿勢データから剛体変換パラメータを取得する．これにより，各姿勢データは図 2(b) のように共通の座標系にマッピングされる。

時間的なキャリブレーションとして，非同期で動作している複数の Kinect に対して，リサンプリング処理により疑似的に時刻同期の取れた姿勢データを算出する。

また，Kinect は人の正面・背面を区別できないため，各姿勢データを共通の座標系にマッピングした際に，それらを左右反転させた姿勢データも同時にマッピングし，コンセンサスのとれた組み合わせを選択した上で統合することで，図 2(c) のような統合を行うことができる。

以後，これら各処理の具体的な方法について述べる。

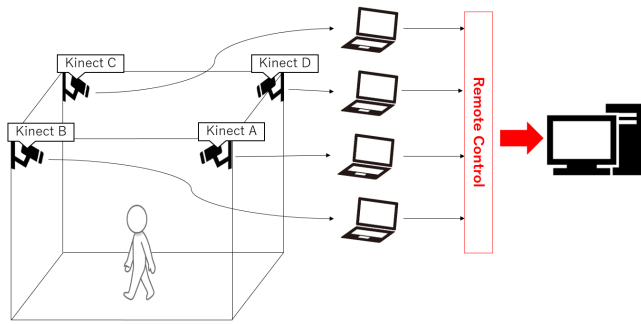


図 3 複数 Kinect の管理システム

3.1 複数台の Kinect を用いた撮影環境の構築

Kinect と人の相対位置関係によっては自己遮蔽が発生するため、人の姿勢によっては正しく姿勢推定を行うことができない。また、Kinect から離れすぎた場合も同様に正しく撮影されないため、これらの欠点を克服するために複数台の Kinect をそれぞれ異なった位置に配置し、さまざまな角度から撮影を行う。

しかし、KinectV2 を使用する場合、単一の PC では複数台を使用することはできないため、図 3 のようなシステムを構築する。使用する Kinect の台数分の PC を用意し、各 Kinect をそれぞれ専用の PC（以下、ノード PC）に接続する。また、各ノード PC から得られる姿勢データ群を集約し、各姿勢データを統合するための統合用 PC を用意する。統合用 PC では、姿勢データ群の共通座標系へのマッピングや、疑似的に時刻同期されたデータ系列の生成、各姿勢データの統合を行うことによって、複数台 Kinect によるモーションキャプチャシステムを実現する。

3.2 空間的キャリブレーション

各 Kinect で撮影された姿勢データは、それぞれの Kinect を原点とした座標系の値として取得されている。そこで、異なる位置・姿勢に設置された複数台の Kinect によって計測される姿勢データを共通の座標系にマッピングするために、外部キャリブレーションを行う必要がある。具体的には、ある姿勢で静止した人物を全ての Kinect を用いて撮影し、姿勢データを取得することができれば、対応する関節点を対応点とみなして剛体変換のための外部パラメータを取得できる。なお、外部パラメータとは、変換前の座標を v 、変換後の座標を V とした時の、式 (1) における、回転と平行移動のパラメータ R, t のことである。

$$V = Rv + t \quad (1)$$

外部パラメータは、図 4 のように最低 4 点以上の対応点があれば、線形に求めることができる。また、複数台の Kinect を用いる際には、この外部パラメータ R, t をそれぞれの Kinect に対して求める必要がある。

なお、Kinect と人との相対位置関係によっては自己遮蔽が発生するため、キャリブレーションを行う際には、正し

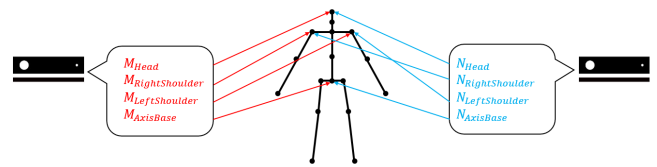


図 4 対応する人の関節点を用いた外部パラメータの導出

く推定されていない関節点が含まれている可能性を考慮する必要がある。正しく推定されていない関節点座標を使用すると、適切な座標変換パラメータを取得することができないため、RANSAC 法を用いてロバストに（外れ値が取り除かれた対応点のみを用いて）変換パラメータを算出する。

3.3 時間的キャリブレーション

各 Kinect は異なる PC（ノード PC）に接続され、姿勢の計測を行う。しかし、Kinect の内部クロックが異なるため、計測は非同期で行われる。そこで、各データが計測された時刻の基準を統一した上で、リサンプリングによって疑似的に時刻が同期された姿勢データ系列の生成を行う必要がある。

まず、計測データに付与する時刻を、共通して UNIX タイムで表すこととする。UNIX タイムは PC の内部時計をもとに計算されるため、各ノード PC の内部時計さえ一致していれば、全ての計測データは、共通した時間軸の中で取得されることになる。よって、各ノード PC の内部時計合わせが重要となる。リサンプリングを行う際も、各データがいつ取得されたのかミリ秒単位で知る必要があるため、ほんのわずかでも各ノード PC の内部時計に狂いが生じると、致命的となる。そこで、定期的に各ノード PC で共通した NTP サーバーにアクセスすることで、内部時計の修正を行う。

次に、時刻情報が付与された非同期姿勢データの時系列に対するリサンプリング処理を行う。図 5 に示すように、リサンプリングを行う各時刻 T について、前後一定時間幅内のデータを抽出し、それらに対する回帰直線を求め、その回帰直線における時刻 T での値を求め、それを時刻 T の値とする。

3.4 複数台 Kinect による姿勢データの統合

3.3 節で生成された疑似同期姿勢データ系列を、3.2 節で得られた変換パラメータで座標変換することにより、時空間的に一致した姿勢データ群としてマッピングすることができる。しかし、これらの姿勢データには一般に、人の前後を誤って推定したものが含まれることがあり、また前後を正しく推定できていても、一部の関節点位置の推定を大きく誤ったデータを含むことがある。これらの問題が起こる原因と、それを踏まえた統合方法について詳細を以下で述べる。

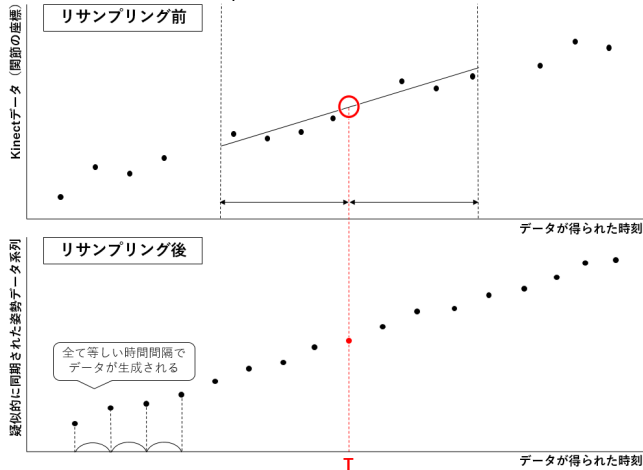


図5 リサンプリング手法

まず, Kinect が人の前後を区別できないがために, 本来の人の向きに対して左右反転したデータを取得してしまう問題について述べる. Kinect はもともと Xbox の周辺機器としてゲーム用に発売されたセンサーであるため, 人と Kinect が向かい合っている状況しか想定されていない. よって, 人が Kinect に対して背を向けた場合, 背中を人の正面だと誤って認識する傾向がある. つまり, 人の右腕や右脚を左腕や左脚だと認識し, 左右反転したデータを取得してしまうのである. そこで, 各 Kinect が取得したデータに加えて, それらを反転させた候補 (以下, ミラーデータ) も挙げて, 対応する関節点間の距離が小さくなる組み合わせを選択することで, この問題を解決する. つまり, 各 Kinect による姿勢データについて, 元のデータとミラーデータのどちらを採用するほうが他の Kinect の姿勢データと対応する関節点の位置が近くなるかを調べ, 近くなる方を採用するという手続きを全ての Kinect の姿勢データについて行うことで, 全ての姿勢データの裏表関係を揃える.

次に, 個々の関節点位置に誤りが含まれる問題について述べる. 図6に, 4台の Kinect (A,B,C,Dとする) で撮影したスケルトンを共通座標系にマッピングした一例を示す. 各 Kinect は, 人との相対的な位置関係により正しく推定できない関節点があり, この例では, Bは右腕を, Dは両脚の推定に失敗している. 適切な統合を行うためには, 関節位置の推定を誤ったデータは使用せず, 右腕の統合には A,C,Dのデータを, 両脚の統合には A,B,Cのデータを統合する必要がある. まず, 左右の認識が揃った各 Kinect による姿勢データを共通の座標系にマッピングした上で, 各関節点について, それらの姿勢データにおける該当関節点の集合に着目し, その中で互いの距離が近いクラスターを検出 (すなわち, 関節点の位置についてコンセンサスを取り), その平均位置を採用するというアプローチで, これらの問題を解決する.

以下は, 4台の Kinect で撮影した場合の本節の提案手

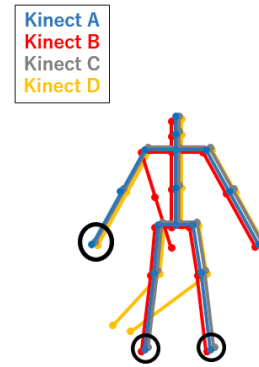


図6 4体のスケルトンの統合

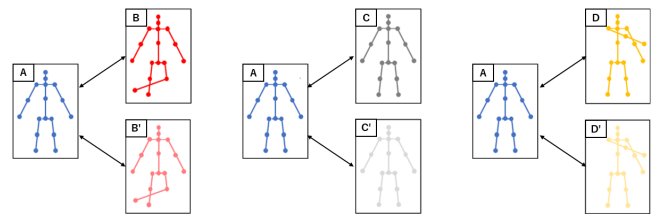


図7 4台の Kinect で撮影した際の統合に使用するデータの選択方法

法の具体的な処理手順である. 各 Kinect で推定された姿勢データを A,B,C,D とし, それらをそれぞれ左右反転したミラーデータを A',B',C',D' とする. まず, Kinect が人の前後を誤る (左右反転したデータを取得する) 問題を解決する. 初めに基準となるデータを1つ決める. 仮に A を基準にした場合, B,C,D,B',C',D'の中から A と向きが揃った (左右の認識が同じ) データを選択すればよい. A の各関節点に対して, B,C,D,B',C',D' の該当関節点とのユークリッド距離をそれぞれ求める. そして, A-B,A-B',A-C,A-C',A-D,A-D' ごとに, 全該当関節点のユークリッド距離群の中から中央値をそれぞれ求め, 図7のように A-B と A-B', A-C と A-C', A-D と A-D' から, 中央値が小さい方をそれぞれ選択することで, 左右の認識が揃ったデータを取得することができる. 次に, Kinect が個々の関節点の推定を誤る問題を解決する. 図7の例で, ユークリッド距離の中央値が小さかったのが, それぞれ A-B,A-C',A-D' だった場合, A,B,C',D' が左右の認識が揃ったデータの組み合わせだということになる. このとき, まず, 図8のように A,B,C',D' を同時にマッピングする. そして, 閾値を設定し, 各関節点ごとに閾値を満たす点セットを検出し, それらの平均位置を算出する. 閾値以内の点セットを統合することで, B や D' に含まれている, 推定を誤った関節点の使用されことなく, 適切な統合を行うことができる.

4. 実験

本実験では, 4台の Kinect (A,B,C,Dとする) を使用して図9(a)のような環境を構築した. Kinect での骨格

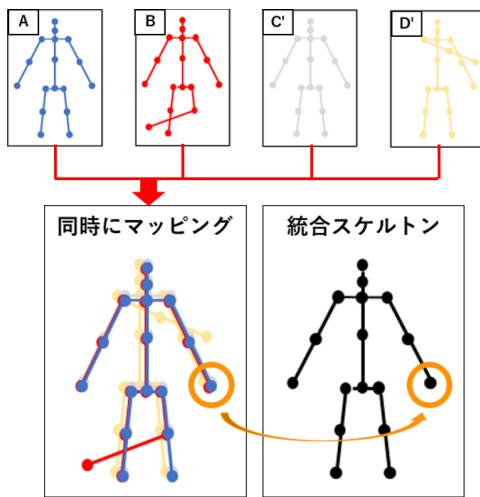


図 8 4体のスケルトンの統合手法

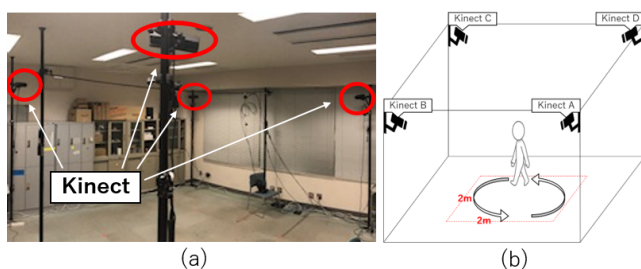


図 9 実験: (a)4台の Kinect を用いた実験環境 (b)撮影時の実験参加者の動き

推定は, Microsoft 社が提供している Kinect for Windows SDK 2.0 [13] と, Processing の Kinect 用ライブラリである Kinect v2 Processing library for Windows [14] を使用して行う. また, 人が動ける範囲は, 撮影環境中心の2メートル四方に設定した. そして, その2メートル四方の範囲を, 実験参加者が図 9(b)の矢印のように円を描くように何周も歩いている様子を撮影し, 4台の Kinect で計測された姿勢データが提案手法通りに統合されたのか確認した.

4.1 座標変換による姿勢データの座標系の統一

3.2 節に示した方法を用いて, 全ての Kinect で計測された姿勢データを共通の座標系にマッピングした. 各データの変換先の座標系 (共通の座標系) を A の座標系に設定し, 他3台の B,C,D のデータを変換した. まず, 外部キャリブレーションによって, B,C,D のデータを A の座標系にマッピングするための3種類の外部パラメータ R, t を求める作業を行った. そして, 得られた外部パラメータを用いて, 全ての Kinect で計測した姿勢データを, 共通の座標系 (A の座標系) にマッピングした結果を図 10 に示す.

図 10(b),(c) が, それぞれ座標変換前後の各スケルトンの様子である. 図 10(a) のように, 異なる位置・姿勢の Kinect を用いて撮影を行っているため, 計測された各姿勢データは, それぞれ撮影を行った Kinect を中心とした座標系の値であり, 図 10(b)の座標変換前のスケルトンを見ると,

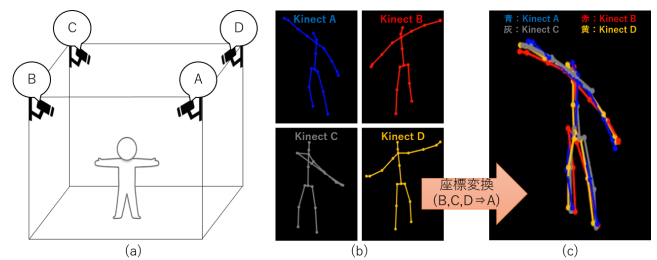


図 10 外部パラメータの取得による, 全データの座標系の統一: (a) 4台の Kinect を用いて, 人を撮影している様子 (b) Kinect A,B,C,D で撮影したスケルトン (変換前) (c) 全ての Kinect データを Kinect A の座標系にマッピングした際のスケルトン

それぞれが異なった方向を向いている様子が確認できる. そして, 図 10(c)の座標変換後のスケルトンを見ると, A のスケルトンに対して他の3体のスケルトンが重なるように変換されている様子が確認できるため, キャリブレーションによって適切な外部パラメータが計算され, 全ての姿勢データが共通の座標系にマッピングされたことが分かる.

4.2 Kinect の時刻同期結果

まず, 各 Kinect に接続されているノード PC 上で NTP クライアントを動作させ, 定期的 (1 分ごと) に強制的に内部時計の同期を行うよう設定した.

各ノード PC では, Kinect から取得される姿勢データに時刻情報を付与したログを生成した. そのログに対して 3.3 節で示した手法でリサンプリングを行い, 疑似的に時刻同期のとられた姿勢データ系列を作成した.

図 11 にリサンプリング処理による疑似同期の効果を示す. 図 11 は, 撮影開始からある程度時間が経過した時のスケルトンの様子である. 図 11 (左) は, 4台の Kinect で計測された姿勢データを, A の座標系にマッピングしただけのスケルトン群である. それに対して図 11 (右) は, 全ての Kinect で計測した姿勢データを, A の座標系にマッピングした上で, 疑似的に時刻同期をとったデータのスケルトン群である. リサンプリングをしていないデータは非同期であるため, 図 11 (左) のようにスケルトンにばらつきが生じている. これに対し, 図 11 (右) を見ると, 4体のスケルトンが重なっている様子が確認できる. よって, 共通の開始時刻と共通の時間間隔で正しいリサンプリングが行われたと言える.

4.3 姿勢データの統合結果

3.4 節で示した通り, 左右の認識が揃ったデータの組み合わせを検出した上で, 各姿勢データにおける該当関節点の中で点間距離が近い (閾値以内の) 関節点の平均位置を算出し, 各 Kinect で計測した姿勢データを統合した. 本節では, 統合されたスケルトンと, 統合に Kinect A,B,C,D (以下, ミラーデータは A',B',C',D') の, どのデータが使

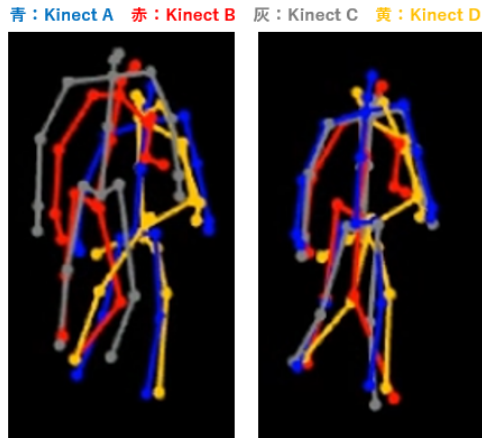


図 11 リサンプリング前後のスケルトン（左：リサンプリング前のスケルトン，右：リサンプリング後のスケルトン）

用されたのかを確認することで，提案手法を評価する．また，閾値は 10cm に設定した．

4.3.1 成功例

まず，全身の関節が正しく統合された例を図 12 に示す．図 12(a) は，全ての計測データを A の座標系にマッピングし，疑似的に時刻同期をとったスケルトンであり，図 12(b) は，その統合スケルトンである．各 Kinect に対しての人の位置は図 12(c) の通りで，人の前後を誤っている Kinect を赤文字（C,D）で示している．A と B が人の正面，C と D が人の背面から撮影している．図 12(a) を見ると，A,B,D では全身のスケルトンがほぼ正確に撮影されており，C では自己遮蔽が原因で左腕が正しく撮影されていない様子が確認できる．また，この時，3.4 節の処理により，左右の認識が揃ったデータとして検出されたデータは，A,B,C',D' だった．図 12(c) の通り，人の前後を正しく認識している A,B に対して，C,D が人の前後を誤っているため，これは正しい結果と言える．

ここで，実際にどの関節の統合にどのデータが使用されたのかを表 1 で確認する．頭や背骨の関節点は自己遮蔽の影響を受けにくく，提案手法の検証材料としてはふさわしくないため，特に自己遮蔽の影響を受けやすい肩や肘，膝，踵の統合にどのデータが使用されたのかを確認する．表 1 を見ると，右肩と右肘の統合に D' が，左肩と左肘の統合に C' が使用されていない．前述したとおり，左腕に関しては，自己遮蔽によって C' では正しく撮影されていないため，左肩と左肘の統合に C' が使用されなかったのは正しいと言える．また，図 12(a) で右腕について見てみると，D のスケルトンの右腕のみ，他のスケルトンの右腕から大きく離れており，閾値の 10cm を満たさなかったと考えられる．よって，右肩と右肘の統合に D' が使用されなかったことも正しい結果と言える．

4.3.2 失敗例

次に統合が上手くいっていない例を図 13 に示す．表 2 を見ると，右肩（基準 A がミラーデータであるため，実際

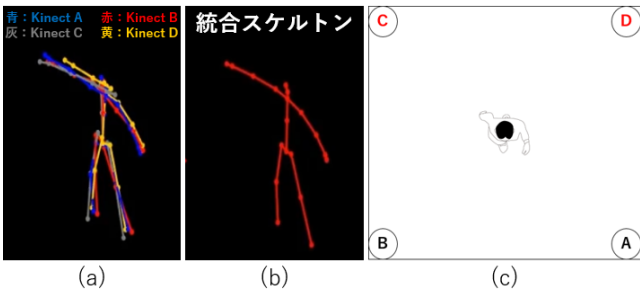


図 12 スケルトンの統合 (a) 統合前のスケルトン (b) 統合後のスケルトン (c) 各 Kinect に対する人の位置・姿勢（赤文字は人の前後を誤っている Kinect）

表 1 図 12 のスケルトンの統合に使用されたデータ

	A	B	C'	D'
右肩	○	○	○	
左肩	○	○		○
右肘	○	○	○	
左肘	○	○		○
右膝	○	○	○	○
左膝	○	○	○	○
右踵	○	○	○	○
左踵	○	○	○	○

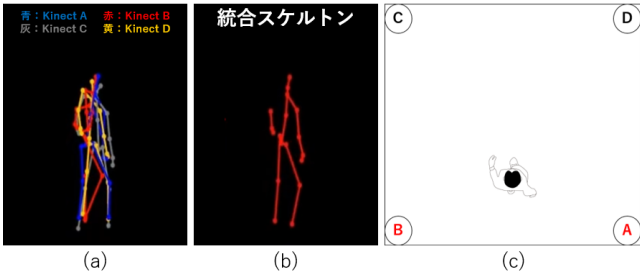


図 13 スケルトンの統合 (a) 統合前のスケルトン (b) 統合後のスケルトン (c) 各 Kinect に対する人の位置及び，視線方向

表 2 図 13 のスケルトンの統合に使用されたデータ

	A	B	C'	D'
右肩				
左肩	○		○	○
右肘		○	○	○
左肘	○		○	○
右膝	○	○		○
左膝	○		○	○
右踵	○	○		○
左踵	○		○	○

は人の左肩の関節点)の統合が行われていない。この問題は、左右の認識を揃えた4つの姿勢データの右肩の関節点について、どの2点間も閾値を満たさない場合に起きる。提案手法は、「閾値以内に2点以上存在する場合にそれらが信用できる点であるとみなす」という考え方に基づいているため、1つの姿勢データの関節点のみが正しく残りがすべて正しくない、という状況において正しいものを選択することは不可能である。

4.3.3 結果の総括

以上の結果より、統合自体は提案手法通りに行われたことが確認できた。そして、適切な該当関節点の組み合わせさえ見つければ提案手法によって正しく統合されることは証明された。しかし、自己遮蔽等によって、閾値を満たすように該当関節点がマッピングされなかった場合、統合が行われないという事例も確認された。本実験では、人の行動範囲を2メートル四方と設定し、4台のKinectで撮影を行っているが、このセッティングでは、すべての関節点が2台以上のKinectで撮影されるという要件が満たされていないことが原因であると考えられる。

5. おわりに

本稿では、複数台のKinectを用いて人を撮影し、各Kinectで得られた姿勢データを時空間的に統合することでモーションキャプチャを実現する手法を提案した。Kinectには、人との相対位置関係によっては自己遮蔽によりデプス画像が観測されないという問題があるため、この手法では、複数台のKinectを用いてさまざまな方向から人を観測し、人の位置・姿勢に関わらず全身姿勢を推定できる環境を構築する。各Kinectは非同期で撮影を行うため、NTPによりノードPCの時刻を同期し、得られた姿勢データと撮影時刻情報を用いてリサンプリング処理を行い、仮想的に同期された姿勢データ系列を生成する。また、キャリブレーションで得られた外部パラメータを用いて剛体変換を行い、全てのKinectで計測された姿勢データを、共通の座標系にマッピングする。そして、各姿勢データ及び、それらを反転したデータをマッピングし、そのデータ群に対してコンセンサスの取れた該当関節点の平均位置を算出することで、統合を行うという手法である。本実験では4台のKinectを用いて撮影を行ったが、使用するKinectの台数を増やすことによって、撮影範囲の拡張やデータ統合の精度を向上させることができると考えられる。

提案手法の問題点としては、統合処理の際に、各Kinectによる姿勢データ及びそれらを左右反転した姿勢データについて、対応する関節点群の点間距離が全て閾値以上だった場合、統合が行えないということである。本稿の提案手法では、点間距離が閾値以内である関節の組み合わせが最低でも1組存在しなければ統合を行うことができない。今後の課題としては、Kinectのカラー画像から得られる姿勢

データの利用が挙げられる。Kinectはデプス画像とは別でカラー画像も取得しているため、OpenPose [12] などを利用して、カラー画像に対しても姿勢推定を行い、統合や補正に利用することでより高精度で安定した姿勢推定を行う予定である。

提案システムは、特殊なスーツやセンサを装着する必要がなく、人の自然な動きを計測する上で都合が良いことから、今後の活用方法としては、アイトラッカーとVRゴーグルを併用した、人の注視・姿勢を同時取得可能なシステムの構築を考えている。人は興味のあるものに自然と眼を向けるため、注視情報は人の意図や興味を知る上で極めて有用であるが、世の中に多く普及している監視カメラ等は非常に低解像度であり、とても眼球から注視方向を推定することなどできない。この問題に対して、Mitsugamiらは、人の頭部と眼球の協調関係をモデル化しておくことで、頭部運動の観測だけから注視方向を推定することが可能であることを示した [15]。また沖中らは、この手法と顔画像に対する顔向き推定と組み合わせることで、人物俯瞰映像からの注視方向推定を行っている [16]。これらの先行研究の発展させ、頭部運動だけでなく、全身運動からの注視方向推定を行うことを考えている。人の注視方向と姿勢の関係性を詳しく分析するためには機械学習を行う必要があり、そのためには、人の注視方向と姿勢のデータセットを大量に用意する必要があり、本研究が非常に有用であると考ええる。

謝辞 本研究はJSPS科研費JP15K16026およびJP18H03312の助成を受けたものです。

参考文献

- [1] 紅林秀治, 小林健太, 兼宗進, “KINECT センサーを用いた簡易動作分析システムの開発. 情報処理学会研究報告,” コンピュータと教育研究会報告, Vol.2013-CE-118, No.20, pp.1-7, 2013 年.
- [2] 吉田聡, 泉正夫, 辻洋, “人物判定における Kinect の骨格線情報の利用,” 映像情報メディア学会技術報告, Vol.36, No.8, pp.123-126, 2012 年.
- [3] S. Zennaro, M. Munaro, S. Milani, P. Zanuttigh, A. Bernardi, S. Ghidoni, E. Menegatti, “Performance Evaluation of the 1st and 2nd Generation Kinect for Multimedia Applications,” 2015 IEEE International Conference on Multimedia and Expo, pp.1-6, 2015.
- [4] D. A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, D. Kim, “Shake’ n’ sense: reducing interference for overlapping structured light depth cameras,” CHI2012, pp. 1933-1936, 2012.
- [5] O. Wasenmüller, D. Stricker, “Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision,” Asian Conference on Computer Vision Workshops, pp.34-45, 2016.
- [6] 餅川穂, 山下晃弘, “指さし動作認識による移動ロボットの操作,” Human-Agent Interaction Symposium 2014, G-6, 2014 年.
- [7] K. Berger, K. Ruhl, C. Bruemmer, Y. Schroeder, A. Scholz, and M. Magnor, “Markerless Motion Capture us-

- ing multiple Color-Depth Sensors,” Vision, Modeling and Visualization, pp.317324, 2011.
- [8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-Time Human Pose Recognition in Parts from Single Depth Images,” IEEE Conference Computer Vision and Pattern Recognition, Vol.2, pp.12971304, 2011.
- [9] M. Nakazawa, I. Mitsugami, H. Habe, H. Yamazoe, Y. Yagi, “Calibration of Multiple Kinects with Little Overlap Regions,” IEEJ Transactions on Electrical and Electronic Engineering, Vol.10, No.S1, 2015.
- [10] M. Nakazawa, I. Mitsugami, Y. Makihara, H. Nakajima, H. Yamazoe, H. Habe, Y. Yagi, “Dynamic Scene Reconstruction using Asynchronous Multiple Kinects,” International Conference on Pattern Recognition, 2012.
- [11] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, Y. Sheikh, “Panoptic Studio: A Massively Multiview System for Social Motion Capture,” International Conference on Computer Vision, 2015.
- [12] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” IEEE Conference on Computer Vision and Pattern Recognition, 2017.
- [13] Kinect for Windows SDK 2.0,
<https://www.microsoft.com/en-us/download/details.aspx?id=44561>
- [14] Kinect v2 Processing library for Windows,
<http://codigogenerativo.com/code/kinectpv2-k4w2-processing-library/>
- [15] Ikuhisa Mitsugami, Yamato Okinaka, Yasushi Yagi, “Gaze Estimation Based on Eyeball-Head Dynamics,” The 1st International Workshop on Human Activity Analysis with Highly Diverse Cameras (HDC2017), 2017.
- [16] 沖中大和, 満上育久, 西野恒, 八木康史, 「低解像度俯瞰映像からの注視推定」, 第 21 回画像の認識・理解シンポジウム (MIRU2018), 8 月, 2018 年.