Vol.2019-HCI-181 No.10
2019/1/21

アバタ媒介型見守りシステムにおけるモーションキャプチャ情報を用いた機械学習の利用

小田 俊平[1,a)]　高野 保真[1,b)]　尾花 謙伍[3]　長谷川 大[2,c)]　佐久田 博司[1,d)]

概要：近年，長寿・高齢化により要介護者の人数は増加している．一方，介護従事者は少子化などの問題により要介護者の増加の割合に対して人数が充足していない．そのため介護従事者の一人当たりの負担が大きくなっており，負担軽減のため介護支援システムに注目が集まっている．代表的な方法であるカメラ映像のみによる見守りは，リアルタイムで見守ることが可能で，介護者の負担を減らせることが期待できる反面，プライバシー侵害の懸念が大きく，見守られる側である高齢者が監視されていることによる精神的負担を高めてしまう．この問題に対し，長谷川らは，センサーカメラの一種である Microsoft® Kinect™ を用いてアバタ媒介によってこの課題を解決し, 安全に高齢者を見守るシステムとして研究してきている．このシステムは部屋の様子などを除いた人の動きのみをアバタとして提示することで，プライバシーを保護したうえで，意思疎通をはかることが可能である．カメラ映像のみの場合と比べて，高齢者のカメラに対する受容性が高まり，大まかな動作で同程度の情報伝達を維持できることがこれまで確認出来ている．本研究では，既存のアバタ媒介による見守りシステムの実用性を高めるため，次の 2 点において改良を加えた．まず，1 点目は，事前のモーションキャプチャ情報を用いた機械学習により，アバタの動きの改善を目指した．次に 2 点目は Kinect 以外のセンサーカメラとして，Intel®RealSense™ を用いた場合の動作の追従性に対して比較した．

キーワード：高齢者見守り，アバタコミュニケーション，機械学習

Correcting Avatar Motion on Distant-Care System using Machine Learning

Shumpei Oda[1,a)]　Yasunao Takano[1,b)]　Kengo Obana[3]　Dai Hasegawa[2,c)]　Hiroshi Sakuta[1,d)]

1. はじめに

近年，日本では長寿・高齢化の進行により，介護を必要とする高齢者が増加している．また，ライフスタイルや家族形態の変化にともなって，独居高齢者世帯もしくは介護老人保健施設に住み，家族と離れて暮らすことも珍しくないこととなってきている．この傾向は今後強まっていくことは明らかであり，離れた家族や介護従事者とのコミュニケーションは，高齢者の孤立感を軽減する意味でも早急に取り組むべき社会課題である．また，介護老人保険施においても，介護従事者は少子化などの問題により要介護者の増加の割合に対して人数が間に合っていない．そのため，介護従事者の一人当たりの負担が大きくなっており，負担軽減のためには，情報テクノロジーを利用した高齢者見守りシステムの開発が望まれている．

以上のような社会的背景から，長谷川らは Microsoft® Kinect™ を用いたアバタ媒介見守りシステム（Avatar

1 青山学院大学理工学部情報テクノロジー学科
Department of Integrated Information Technology, College of Science and Engineering, Aoyama Gakuin University, Sagamihara, Kanagawa 252-5258
2 北海学園大学工学部生命工学科
3 ヤフー株式会社
a) a5814020@aoyama.jp
b) takano@it.aoyama.ac.jp
c) dhasegawa@lst.hokkai-s-u.ac.jp
d) sakuta@it.aoyama.ac.jp

ⓒ 2019 Information Processing Society of Japan

Mediated Distant-Care System, 以降 AMD と呼ぶ）の研究・開発を進めてきた [3][4]．このシステムでは，従来のカメラ映像による見守りシステム [2] の問題点であった被介護者のプライバシーの問題点を解決するために，3DCG で人間を模して作成したキャラクタ（アバタ）を介してコミュニケーションを行うことができる．先行研究では，高齢者に対する AMD を実際に使った調査実験から，カメラ画像による見守りシステムと比較して受容性が高いことが確認されている．その一方で，姿勢推定結果に含まれている不自然な関節位置角度が，アバタ描画時に与える影響により，細かな動作において不自然に描画されることにより，コミュニケーションの種類が限定されてしまうことが示唆された [3]．また，既存の AMD は，Kinect を利用することを前提としていたため，ハードウェアの選定において自由度がなかったという問題点もあった．

そこで本研究は，機械学習を用いて，不自然な関節位置角度を自然な関節位置角度を改善し，アバタを補正するように既存の AMD を拡張した．「マーカを付けて測定した高精度のモーションキャプチャで取得した動き」を関節位置角度の正解データとして事前に学習しておき，カメラで得られた関節位置を学習結果から作成したモデルによりリアルタイムで補正する．また，Kinect に加えて，Intel®RealSense™ を利用することができるように拡張を加えた．拡張済み AMD を用いて，機械学習による補正の有無と，ハードウェアの違いによるアバタの動作について，比較した．

2. AMD の概要

AMD は，1 節で述べた問題を解決することを目標として，以下の特徴を持ったシステムである．

(1) アバタによるコミュニケーション
(2) 1 台の安価なカメラによるマーカレスモーションキャプチャシステム
(3) Web 技術を用いた遠隔見守り

(1) の特徴により，部屋の状態や服装など，詳細かつ多量のプライバシー情報が含まれてしまうカメラ動画像から情報を隠蔽した上で，コミュニケーションを取ることが可能となった．また，3DCG で自由にアバタや部屋を設定できるという機能より，コンピュータによるコミュニケーションへの抵抗感を下げることもできる．なお，AMD では基本的に見守りを対象とするため，要介護者の映像を見守るという一方向のコミュニケーションを想定している．

(2) の特徴について，センサーカメラなどを複数用いることなく 1 台の安価なマーカレスモーションキャプチャシステムでアバタ生成することがシステムの普及を考えると重要である．一方で，マーカーを付けたモーションキャプチャのような高い精度で動きを再現することはできないため，導入コストと動きの精度にトレードオフの関係があるといえる．AMD では，日常のコミュニケーションでは，動きの高度な追跡は必要ないという設計判断から，マーカーレスで実用的な精度を実現している．

(3) の特徴について，広く用いられている Web 技術を使うことによって，見守り側は，通常の Web ブラウザでサーバにアクセスすることで，見守りが可能である．具体的に，通信には WebSocket，描画には WebGL を用いている．標準技術を用いることで，特別なクライアントソフトを必要とせず，導入が容易であり，PC のみでなくタブレット端末などでも利用である．また，WebGL によるアバタの描画では，コミュニケーションには十分実用的な 3D 描画性能を示している．

3. 提案手法

3.1 目的

本システムの目的は，以下の 2 つである．

- アバタの動きにおける自然さと再現性の向上
- RealSense への対応

一つ目の項目について，プライバシーの保護を保ちつつ高度なコミュニケーションを行うためには，アバタのリアルタイムで自然な動きを伝えることが重要である．二つ目の項目について，Kinect の代替となるセンサー機器として，RealSense は同等の性能を持っている上，サイズも小さく，AMD の実用化に向けて必要である．

3.2 概要

図 1 に提案システムの全体像を示す．本研究では, 先行研究 [3][4] で開発したアバタ媒介型見守りシステムを改良する．そのシステムの構成としてセンサ機器から関節位置角度情報を取得しサーバー側に送信を行う被見り守側システム，関節角度位置情報のデータ処理・中継を行うサーバ, サーバから取得した関節位置角度情報を元にアバタ化し表示する身守り側クライアントの 3 つで構成されている．ここでいうサーバーのデータ処理とは，送られてきた関節位置情報を機械学習結果を使って補正を行い，モデリングのための適切な形式に変換し，クライアントからアクセスされた際にアバタ表示が出来るようにしておくことを指す．本研究で追加する機構はサーバ内に機械学習を用いて補正行うプログラムである（図中の赤枠）．被見守り側から送信された関節情報のデータ処理を行う前に関節位置角度情報の内の関節回転角度データの数値補正を行い，補正後のデータをクライアントに送ることとする．

3.3 あらかじめ決めた動作の補正

マーカーを付けたモーションキャプチャーデータのデータを利用して機械学習すれば，アバタの動きを補正できることが確認されている [1]．そこでは,「手を挙げる」や「伸びをする」といった決められた動作について，マーカーを

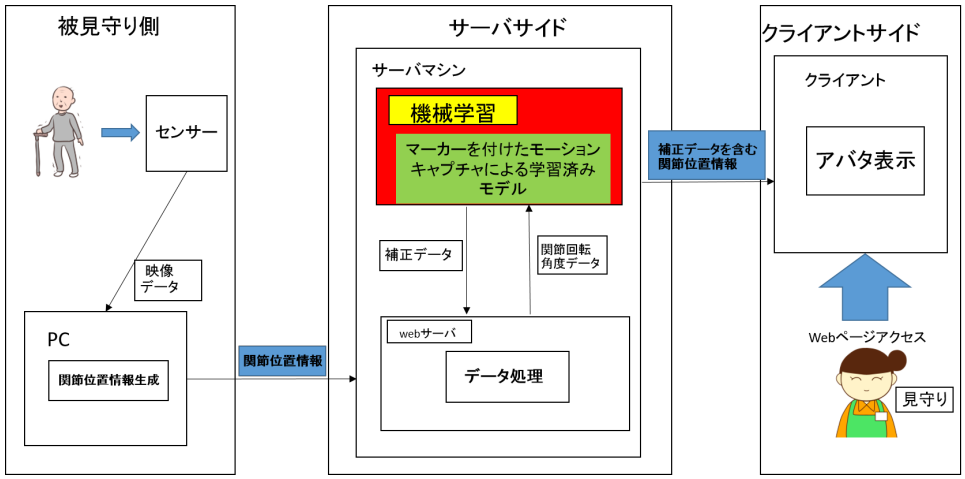


図 1　機械学習を追加した AMD の概要図
Fig. 1　Overview of AMD with machine learning

付けたモーションキャプチャーデータを蓄積し，同じ動作をセンサーカメラの前で行ったときの補正度合を比較している．条件として，1 つの関節位置角度には (回転軸; 回転角度) = (x,y,z;w) の 4 変数が含まれており，Kinect から 25 関節分のデータを取得している．また，関節位置角度を補正をする際，前後 10 フレームずつの関節位置情報データを入力とし，そこから動きを予測し補正した．なお，センサーカメラには Kinect を用いている．

学習モデルへの適用結果を「マーカ付きの高精度のモーションセンサで取得した動き」からの関節位置誤差の総和の平均で約 10.4 % 改善できるという結果が得られた．つまり，学習した動作と，Kinect 前で行う補正対象の動作がほぼ等しいという理想的な状況を想定すれば，十分な補正効果が得られることが分かっている．

本研究では，尾花らの研究結果を元に，同様の機構を AMD に実装しリアルタイムにセンサーカメラからのデータを得て，学習済みモデルに適用してアバタの動きを補正することとする．

3.4　機械学習を用いた動作の補正

マーカーを付けたスーツを着用し，典型的な動きをした場合のモーションキャプチャーデータを事前に大量に取得しておく．そのデータを機械学習し，人間の関節の動きの正解モデルを作成する．

そのモデルを用いて，センサーカメラから得られた各関節の動きを補正し，アバタの不安定な動きを減らすことを目指す．なお，見守りにおけるコミュニケーションでは，下半身の動きは重要でないと考えて，モーションキャプチャーによる学習対象としなかった．そのため，下半身の関節点については，モデルを適用せずにセンサーカメラから得られた値をそのまま用いる．

3.3 節の結果より，あらかじめ決められた動作であれば，

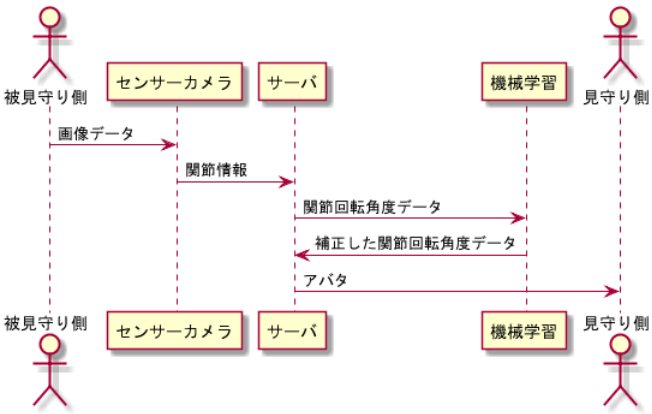


図 2　機械学習を適用する場合のシーケンス図
Fig. 2　Sequence diagram with machine learning

有意に動作を補正できることが分かっており，リアルタイムに学習結果を使うように AMD を拡張する．上のモデルで補正をかけるため，センサーカメラからのデータを 20 フレーム分を必要とする．そこで，20 フレームのデータが貯めてから機械学習でデータを補正することとした．予備実験では前後 10 セット分のデータであったが，今回はリアルタイムで動かすため実際に動き始めてからの 20 セット分のデータを使用する．それにより開始直後のみは 20 セット分のデータセットを貯めるまで わずかな遅延が発生するが，20 セット貯めてからは常に 20 セット分貯まるため遅延は起こらないようになっている．

図 2 に機械学習による補正時のシーケンス図を示す．センサーカメラで介護者を撮影し，Kinect SDK が映像データから人が検知できた場合，関節情報を生成を行い，サーバに送信する（人が検知出来なかった場合は空の関節情報を送信する）．サーバーはまず送信されてきた関節情報内から関節回転角度データを抽出する．その後，関節回転角度を機械学習済のモデルにより補正を行い，関節回転確度データに上書きし，サーバに関節情報を送り返す．サーバ

表 1　Kinect と RealSense の性能比較
Table 1　Comparison between Kinect and RealSense

	Kinect v2	RealSense D435
RGB カメラ解像度	1920x1080	1920x720
RGB カメラ fps	30fps	30fps
深度センサー解像度	512 × 424	1280 × 720
深度センサー fps	30fps	90fps
深度取得範囲	0.5m-4.5m	0.2m-10m
RGB 視覚角度-水平	70 度	69.4 度
RGB 視覚角度-垂直	60 度	42.5 度
関節数の数	25	25

はその関節情報を元にデータ処理，アバタデータを送る．見守り側はサーバにアクセスすることでアバタを確認することが出来る．

3.5　RealSense への対応

表 1 に Kinect と RealSense の性能を比較した．比較の結果，Kinect v2 と RealSense D435 ではハードウェアの性能面に大きな差がなく，RealSense D435 は十分に Kinect v2 の代用が出来ると考えられる．しかし，深度取得範囲の違いなどから表示するアバタの大きさのパラメーターを調整する必要がある．

一方で，ソフトウェアにおいては，取れるデータ形式などの変換が必要となる．まず，既存の AMD では，Microsoft が提供する Kinect SDK を利用して，センサーカメラからのデータから関節情報を生成していた．本研究で新たに開発する RealSense をセンサーカメラとして用いる場合には，Intel の SDK に関節情報を生成する機能が含まれていない．そこで，本研究では，Nuitrack 社の SDK [*1] を利用する（以後 Nuitrack と呼ぶ）．Nuitrack で取得できる関節情報は，Kinect SDK の関節情報とデータ形式が異なっており，Nuitrack では取得する関節回転角度データは回転行列データのため 1 関節に対して 9 点のデータが取得できる．一方で，Kinect SDK で取得できる関節回転角度データはクォータニオンである．AMD のアバタ描画プログラムはクォータニオンデータを前提としているので，これらのデータ形式間で変換の必要があった．

表 2 に Kinect SDK を基準に Nuitrack で取得できる関節点を示す．表 1 から分かるように Kinect SDK と Nuitrack で取得できる関節数の数は同じだが Nuitrack SDK だと認識できない関節点がある。そのため，既存の AMD で実装済みの Kinect SDK を基準に，Nuitrack で取得した関節点を調整する必要がある．Nuitrack では両手の親指を認識出来ないが，アバタを描画する際に手先や足先がない場合の処理があるため並び替えの処理を行えば対処することが出来た．並び替えの処理は，回転行列データをクォータニオンデータに変換する処理を，被見守り側 PC で行う．

[*1] https://nuitrack.com/

表 2　Kinect SDK と Nuitrack の関節点比較
Table 2　Comparison between Kinect SDK and Nuitrack

	Kinect SDK	Nuitrack
0	spine base	Waist
1	spine mid	Torso
2	neck	Neck
3	head	Head
4	shulder right	Right Shoulder
5	elbowright	Right Elbow
6	wrist right	Right Wrist
7	hand right	Right Hand
8	shoulder left	Left Shoulder
9	elbowl eft	Left Elbow
10	wrist left	Left Wrist
11	hand left	Left Hand
12	hip right	Right Hip
13	knee right	Right Knee
14	ankle right	Right Ankle
15	foot right	Right Foot
16	hip left	Left Hip
17	knee left	Left Knee
18	ankle left	Left Ankle
19	foot left	Left Foot
20	spine sholder	Right Collar,Left Collar
21	hand tip right	Right Finger Tip
22	thumb right	×
23	hand tip left	Left FingeTip
24	thumb right	×



図 3　旧 AMD のアバタ表示

Kinect SDK の spine base に当たる部分は，Nuitrack では Right Collar と Left Collar 2 種類である．しかし，この 2 種類の値はを比較するとどちらも同じであるため今回は Right Collar を使用する．Kinect と RealSense の性能面でセンサーカメラの範囲が異なるため, Kinect と同じ値を適応するとアバタが正常に表示できなくなるため数値を適宜変更する必要がある．また Kinect SDK と Nuitrack では取得する位置の座標系が異なり，Y 座標が 180 度反対のためアバタが逆さ向きになる．そのため，Nuitrack で取得する Y 座標の値を 180 度反転させた．

ⓒ 2019 Information Processing Society of Japan



図 4　学習済みモデルにより補正されたアバタ表示



図 5　RealSense のアバタ表示

4. 評価

4.1 機械学習を用いた AMD と旧 AMD との比較

機械学習の有無による動作が与える印象を確認するため，主観的な評価を行った．機械学習を通したアバタは，機械学習を通していないアバタに比べ，不自然な関節位置角度を補正したことにより不安定な動作が減ったと感じた．しかし，全体に関節の可動範囲に制限が加わっている印象で，手を上に伸ばしてもアバタの手が伸びきっていない，または，顔が下に向きがちであった．比べてみると，予備実験ほどの効果が実感できず，リアルタイムで補正を行うには，さらなる分析が必要であると感じられた．予備実験のアバタは，前後 10 フレーム分のデータを貯めてから学習済みモデルで補正するため，アバタの動きの自然性や再現性がリアルタイムでは補正されにくかったと考えられる．

4.2 RealSense を用いた AMD

RelaSense に関しては遅延がひどく，学習済みモデルによる補正を行うときには，アバタが表示すらされなかった．RealSense を使用した AMD に関しては RealSense から送られてくるデータ量が Kinect SDK に比べて少なく，アバタが人の動きに追いついてきておらず，十分に人の動きを模すことが出来なかった．こちらに関しては，アバタを生成する際に必要とする関節情報を取得する SDK の性能を上げることが出来れば，リアルタイム性を維持できるには十分なデータ量を得られるのではないかと考えている．他にはセンサーカメラを接続した PC のスペックを上げることで，この問題を解決可能であるかもしれない．

5. おわりに

本研究では，既存のアバタ媒介による見守りシステムの実用性を高めるため，次の 2 点において改良を加えた．まず，1 点目は，事前のモーションキャプチャ情報を用いた機械学習により，アバタの動きの改善を目指した．次に 2 点目は Kinect 以外のセンサーカメラとして，RealSense を用いた場合の動作の追従性に対して主観的な比較を行った．比較結果から，機械学習を用いた見守り介護システムは機械学習を用いていない見守り介護システムに比べて大きな改善は見られなかった．しかし，機械学習済みモデルを用いたことで関節位置角度に補正がかかり，不安定なアバタの動作が起こりにくくなったたという点については一定の効果は得ることが出来たと考える．また，RealSense を AMD で利用するためには，適切な関節情報を得られる SDK が必要であることが分かった．

今後の課題は，機械学習で学習するデータ量をさらに増やすことで，リアルタイムであっても補正が行えるように改善する事である．予備実験では，補正の効果が確認されており，AMD のコミュニケーションで想定されるより細かい動作についても学習データが必要であると考えている．また，別の課題として，AMD の実証実験などにより，補正に対するより詳細な評価を行う予定である．

参考文献

[1] 尾花謙伍，長谷川大，白川真一，金子直史，佐久田博司，鷲見和彦：アバタ媒介型見守りシステムにおける Kinect の姿勢推定エラー補正，HAI シンポジウム 2017，p. P26 (2017).
[2] 杉原太郎，藤波　努，高塚亮三：グループホームにおける認知症高齢者の見守りを支援するカメラシステム開発および導入に伴う問題，社会技術研究論文集，Vol. 7, pp. 54–65 (2010).
[3] 長谷川大，盛川浩志，佐久田博司，中山栄純，安彦智史：アバタ媒介型見守りシステムの高齢者における受容性について，ヒューマンインタフェース学会論文誌，Vol. 20, No. 2, pp. 163–172 (2018).
[4] 長谷川大，小林　裕，白川真一，佐久田博司，安彦智史，安達栄治郎，中山栄純：アバタ媒介型見守りシステムの開発，知能と情報，Vol. 28, No. 6, pp. 974–985 (2016).

ⓒ 2019 Information Processing Society of Japan