アバタ媒介型見守りシステムの提案と通信性能評価

長谷川　大[*1]　小林 裕[*1]　白川 真一[*2]　佐久田 博司[*1]　安彦 智史[*3]
中山 栄純[*4]

Proposal of Avatar Mediated Distant-Care System and Evaluation of Communication Performance

Dai Hasegawa[*1], Yu Kobayashi[*1], Shinichi Shirakawa[*2], Hiroshi Sakuta[*1], Satoshi Abiko[*3], and Eijun Nakayama[*4]

Abstract - We propose a privacy protective avatar mediated distant-care system. The system avateers an elderly person and animates the avatar representation based on their articular angles acquired by a motion capture system. The design of our distant-care system allows us to achieve privacy protected information delivery of the elderly person's bodily movements. We implemented the system in a form of web application with an inexpensive motion-capture device, and conducted a study in which we evaluated the system's communication performance. As the result, we confirmed that the system could draw fluid animation without delay under the condition of 700Kbps network bandwidth.

Keywords: Avatar, Distant Care-giving, Privacy Protection, Web Application

1.　はじめに

日本の高齢化率は 2060 年頃まで上昇すると推計されており，今後も孤立化した高齢者は増加することが予想される．また，近年では，少子高齢化の進行やライフスタイル・家族形態の変化にともなって，独居高齢者世帯や高齢者夫婦のみの世帯が増加しており，高齢者の社会的孤立が注目されている．

高齢者の社会的孤立は，心身状態の変化に対する観取が遅れ，問題を悪化してしまうケースが多く，自立した生活を困難にする間接的な要因となっている．また，従来，家族・地域の相互扶助によって支えられていた生活支援や心的ケアを，自治体や民間企業が提供する人的な介護・福祉サービスで代替するため，社会保障費の増大につながっており，財政問題の一因としても認識されている．

このような背景から，高齢者の見守りを行う ICT を利用した様々な研究がなされている．冷蔵庫ドア開閉やトイレの使用検知センサによって生活習慣を遠隔地から把握するシステム[1]，ドアセンサや人感センサ，室温センサなどを使用した健康状態を把握するシステム[2]，プレゼンスのみを双方向に伝達するバックグラウンドコミュニケーションに着目したシステム[3]などが開発されており，高齢者の日常生活を把握することによる安心感の獲得や高齢者への関心の保持が実現されている．しかし，身体動作をセンシングし，プライバシを阻害せずに提示する見守りシステムはこれまでに提案されていない．

そこで本稿では身体動作の観取によって日常生活を把握する見守りシステムを提案する．本システムはモーションキャプチャによって得られた関節位置角度情報をもとに 3D アバタとして被見守り者を再現するアバタ媒介型の見守り方式とし，Web アプリケーションとしてのシステム構成例，および実運用に向けた基礎的な調査結果を示す．

2.　システム概要

図 1 に提案するアバタ媒介型見守りシステムの構成を示す．

本システムは，被見守り側クライアントシステム，関節位置角度情報の中継を行うサーバ，見守り側クライアントとなる Web ブラウザの 3 点からなるクライアントサーバモデルで構成される．被見守り側クライアントシステムは関節位置角度情報を取得するためのセンサ機器，センサ情報から関節位置角度情報を取得・処理を行うプログラム，関節位置角度情報をサーバへ送信するクライアントプログラムから構成される．サーバは被見守り側クライアントシステムから受信したデータを，複数の見守り側の Web ブラウザで動作するクライアントプログラムへ送信する．

*1: 青山学院大学　理工学部
*2: 横浜国立大学大学院　環境情報研究院
*3: 仁愛大学　人間学部
*4: 北里大学　看護学部
*1: College of Science and Engineering, Aoyama Gakuin University
*2: Graduate School of Environment and Information Sciences, Yokohama National University
*3: Faculty of Human Studies, Jin-ai University
*4: School of Nursing, Kitasato University

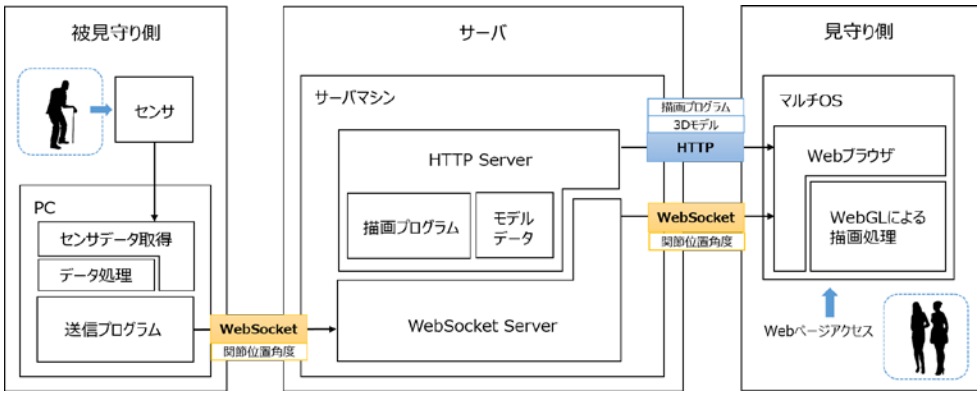

図 1 システム概要

Fig. 1 System Overview

ここで，見守り側クライアントでは 3D アバタで可視化された人物像の動作を通じて高齢者の見守りを行うこととなるが，被見守りクライアント 1 つに対して，複数の見守り者が様々なデバイスで利用できることが望ましい．また，サービスの運用コストを考えると，サーバへの負荷を軽減するために，グラフィックス計算は見守り側クライアント側で実行されることが期待される．

2.1 描画処理

3D グラフィックスを扱うソフトウェア開発では，OpenGL や DirectX やモバイル端末向けの OpenGLES などの API を利用するのが一般的であるが，利用するデバイスの OS 毎にソフトウェア開発を行う必要があることや，利用時にソフトウェアのインストール作業が必要があるなどのコンピュータ利用に不慣れなユーザへの障壁があった．しかし，近年，一般的なデスクトップ型 PC やモバイル端末で使用されるざまざまな OS に対応するモダン Web ブラウザ上で実行され，GPU による高速なグラフィックの描画処理が可能な JavaScript API である WebGL の整備が進められている．

本システムの見守り側クライアントではグラフィックス描画に，ブラウザを実行環境とする WebGL を採用する．実行環境を Web ブラウザとすることでマルチプラットフォームへの対応が可能であり，また利用時に必要な作業もブラウザのインストールのみとなり利用者への負担減につながる．また，描画計算はクライアント側で実行されるため，サーバに必要とされる処理は，データサイズの小さい関節位置角度情報の送受信のみとなり，動画像を利用する場合と比較して負担が少なく，サービス運用上のメリットとなる．現在では WebGL に対応しているブラウザは，PC 向けの Google Chrome，Mozilla Firefox，Internet Explorer，Opera，Safari，またモバイル端末においても主要なブラウザの多くでサポートされている．

2.2 通信方式

ここで被見守り側から受信したデータはサーバを経由して見守り側クライアントへ通知される．通常，サーバから Web ブラウザへデータ送信するためには，HTTP プロトコル上で Push 通知を実現する方法である Asynchronous JavaScript + XML (Ajax) を用いて定期的にリクエストを送ることでサーバの通知を確認するポーリングや Web ブラウザからのリクエストをサーバ内で保持しておき通知があった時にレスポンスを返す Comet と呼ばれる方法が使用される．しかし，これらの方式には冗長性やリアルタイム性に課題があり，滑らかなアニメーションとして知覚されるのに必要な 15-30fps 程度の頻度でデータを通信することは困難であった．そこで，本システムの通信には，近年，多くの Web ブラウザに実装されている応答性の高い双方向通信を実現する WebSocket プロトコルを採用する．WebSocket も WebGL と同様に PC 向け，モバイル端末向けの主要な Web ブラウザで対応されている．

2.3 実装例

アバタ媒介型見守りシステムの実装例を示す．本実装では，被見守り側の関節位置角度情報を取得するためのセンサとして Microsoft 社の KINECT を用い，KINECT から得た距離画像をもとに KINECT SDK for Windows v1.8 を使用して，全 20 関節の位置・角度推定を行った．また，得られた関節位置角度情報をサーバに送信する WebSocket クライアントプログラムとして C¥#による実装であるライブラリの Fleck を使用した．データ送受信用の WebSocket サーバの実装には JavaScript ライブラリ ws を使用し，また 3D モデルデータや HTML ファイル，JavaScript プログラムを配信する HTTP サーバは JavaScript ライブラリ express を用いて構築し，共に実行環境には Node.js を用いた．見守り側でデータ受信を行う WebSocket クライアントは JavaScript ライブラリ ws を使用し，描画には WebGL のラッパーライブラリである

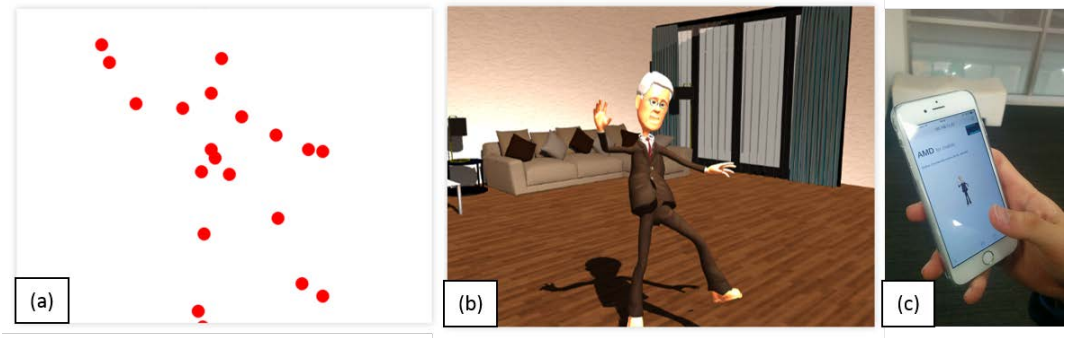

図 2 アバタ媒介型見守りシステムの動作例

Fig 2. Examples of Avatar Mediated Distant-Care System

Three.js を使用した.

また，コンピュータグラフィックスにおいて人物を表現する際には，アバタに階層化された骨格構造を利用し，各関節の回転角度をコントロールすることでアニメーションが生成されるが，モーションキャプチャソフトウェアや描画ライブラリによって骨格構造の定義や回転角度表現が異なる場合が多く，関節位置角度情報の変換処理が必要となると考えられる．本例でも KINECT SDK for Windows で取得されるスケルトンの関節回転角の表現が描画で使用する Three.js における定義と異なっているため，変換処理を見守りクライアント側の描画プログラム内に実装した．具体的には，KINECT SDK for windows v1.8 で取得される相対関節角度は，各ボーンの親となるボーンの向きをベクトル(x,y,z)=(0,1,0)とした時の回転角度を表したものである．親を持たない Root に対しては，モデル全体の向きを，絶対座標系で表したものが設定される．

一方, Three.js のアニメーション定義のフレームデータにおける回転角度は，ボーンの初期状態からの回転を示すことと，四元数表現（回転軸と回転角度）の軸表現が絶対座標系におけるものである点で KINECT と異なっている．そのため，描画プログラム内で，各関節ごとに補正用の四元数を用意し，前処理として各フレーム毎に四元数の乗算処理で変換をおこなった. アバタには, Blender によりモデリングされた頂点数 20,677 の 3D モデルを使用し，KINECT が出力する骨格構造に合わせて 20 個の関節を組み込んだ.

本実装では，被見守り側 WebSocket クライアント，WebSocket サーバ，HTTP サーバ，Web ブラウザ間で送受される関節位置角度データ並びに 3D モデルデータは全て JSON 形式とした．全 20 関節の 3 次元座標および回転角度と回転軸を示す四元数（クォータニオン）のデータサイズは 6KByte であった．また，3D モデルデータは，瞬きや簡易なリップシンク用のモーフアニメーションデータおよびテクスチャ用画像も含め，データサイズは 1.2MByte であった．しかしアバタや背景の 3D モデルデータには，接続開始時に HTTP サーバから WebSocket クライアントプログラムや描画プログラムとともにダウンロードされるだけであるため，その後の通信に影響はない．%但し，携帯端末等の Web ブラウザで利用する場合にはグラフィック処理およびブラウザのキャッシュメモリへの負荷を考慮し，頂点数やデータサイズへの配慮をおこなった.

本実装における動作例を図 2 に示す．図 2(a)に示す KINECT から取得された関節位置情報が取得され，Web ブラウザ上で図 2(b)にあるアバタ表現が生成される．また背景に使用するモデルデータなどを削除し，ブラウザが保持しなければならないデータ量を抑えることで Web ブラウザに割り当て可能なメモリ量が比較的小さなスマートフォンでも見守りを行うことができる（図 3(c)）.

3. 実験および考察

提案システムは，WebSocket 上に毎秒 15 フレームから 30 フレームの頻度で関節位置角度データを送信することになるが，ネットワークの帯域幅が送信データ量を下回る場合，ネットワーク上にデータが滞留し，実現されるアニメーションに遅延が発生する．そこで，提案システムが滑らかなアニメーションを遅延なく動作するために必要なネットワーク要件の調査を行う．ここでは，視覚的に不自然でないアニメーションが生成可能なフレームレートの下限値を 15fps とした．ここで，アニメーションのフレームレートは，送信するデータ量・頻度とネットワークの帯域幅のほかに，見守り側クライアント PC の描画処理速度およびデータを中継するサーバマシンの通信処理速度などのハードウェア性能にも依存する．本実験では，これらのハードウェアとして後述する一般に入手可能なものを使用したが，性能要件を十分に満たしているためフレームレートへの影響はないと考えられる.

検証の際に使用したハードウェアは，被見守り側クライアント PC にタブレット型 PC (OS: Windows8.1, Intel Core i3-3217U Processor (3M Cache, 1.80 GHz)，主メモリ: 4GB，Intel HD Graphics 4000 (350 MHz，1.05 GHz)),

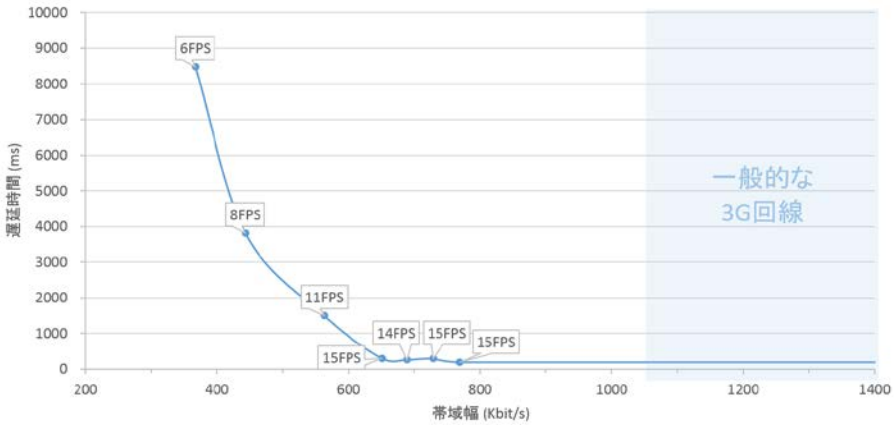

図 3 実験結果

Fig. 3 Result of Experiment

サーバマシンにはワークステーション (OS: Ubuntu14.04, CPU:Intel Core i7-920 Processor (8M Cache, 2.66 GHz), 主メモリ: 6GB)，見守り側クライアントの Web ブラウザは Firefox 35.0 であった．また，データ送信時と受信時の時間差を正確に計測するために Firefox は被見守り側クライアント PC 上で実行した. KINECT で取得された 1 フレーム分の関節位置角度データは，6KByte/frame であった．これらの PC およびサーバマシンを Wifi ルータ (Buffallo WZR-HP-AG300H，IEEE802.11n，5GHz 帯，最大帯域幅 300Mbps)で構築したプライベートネットワークに接続した．ルータとサーバマシンは有線で接続し，クライアント PC は無線による接続とした．ここで，被見守り側からサーバまでのネットワーク要件は十分に帯域が確保されていると仮定し，サーバから見守り側の経路となるサーバマシンのアウトバウンドの帯域幅に制限をかけて，検証を行った．帯域幅の制限には Linux コマンドの tc (trafic control) コマンドを使用した．また，遅延時間の測定には, データ送信開始から 10 秒後に被見守りクライアントから送信された関節位置角度データが，見守りクライアント Web ブラウザに受信されるまでの時間を 10 回計測し，平均値を計算した．

図 3 に, 計測によって得られた帯域幅と 10 秒後の遅延時間，および実現されるアニメーションのフレームレートの関係を示す．本計測では，700kbps 程度の帯域幅で遅延なく 15fps のアニメーションが実現されることが確認された．また，大域幅が 700kbps 以下になると，徐々に遅延が現れ，見守り側アニメーションのフレームレートが低下する現象が見られた．また，ここでは 10 秒後の遅延を計測したが，遅延は経過時間とともに大きくなるため，帯域幅が 700kbps 以下になる部分が存在するネットワーク環境では本システムは利用できない．また，見守り側のクライアントが複数存在する場合は，サーバマシンのアウトバウンド通信量が増加するが，帯域が 100Mbps の回線の場合であれば 140 クライアント程度の同時接続が見込める．

4. おわりに

本稿では，プライバシを保護しながら長期・継続的な見守りを行うことを目標として，モーションキャプチャセンサから得られた被見守り者の関節位置角度情報をアバタの動作として再現して見守り者に提示する，アバタ媒介型見守りを提案した．また，提案方式によるシステムが日常的に使用されることを想定した場合の利便性を考慮し，アバタの閲覧機能を Web ブラウザを実行環境とする Web アプリケーションとして実装し，通信・描画性能について基礎的な検討を行った．提案システムを安価なモーションキャプチャデバイスを利用して実装を行い，通信性能を検証した結果, 700kbps 程度の帯域幅で遅延なく滑らかなアニメーションが実現されることを確認した．

謝辞

本研究は，総務省戦略的情報通信研究開発推進制度 (SCOPE) 地域 ICT 振興型研究開発「介護支援人型エージェントによる地域医療コミュニティネットワークの研究開発」の一環として行われた．

参考文献

[1] 小林，沼田，目黒 ：平時から災害時まで利用可能な高齢者の生活習慣の遠隔見守り支援システムの研究 ；生産研究，vol.63，no.4，pp.465-470 (2011).

[2] 前川，中島，今西，樋口 ：居住空間のスマート化に向けた高齢者見守りシステム開発の取り組み ；ヒューマンケア研究学会誌，vol.5，no.2，pp.51-54 (2014).

[3] 宮島，伊藤，渡邊 ：バックグラウンドコミュニケーションをベースとした新しい見守りサービス ；電子情報通信学会論文誌 D, vol.88, no.12, pp.1785-1794 (2005).