

研究論文

感情音声合成を用いた高齢者向けコミュニケーションツール

山中 裕介* 實廣 貴敏*

(2021年9月30日受理)

Communication Tool for Elderly Persons Using Emotional Speech Synthesis

Yusuke YAMANAKA* Takatoshi JITSUHIRO*

(Received September 30, 2021)

Abstract

In this paper, we propose a communication tool for watching over elderly people using speech information processing technology. It is one of messaging tools but it uses speech recognition and speech synthesis for elderly persons to make inputting messages easier. Especially, it can send one of four emotions for one sentence and synthesize emotional speech. This function can give more information than those of usual messaging applications, and it is friendlier to the elderly. In the limited experiments, the results show that the proposed communication tool is effective to watch over elderly persons.

キーワード: 高齢者, コミュニケーションツール, 感情音声合成, 音声認識, IoT

Keywords : Elderly person, Communication tool, Emotional speech synthesis, Speech recognition, IoT

1. はじめに

現在日本では少子高齢化により介護業界の人手不足が問題になっている。さらに団塊の世代が後期高齢者(75歳)の年齢に達し、医療や介護などの社会保障費の急増が懸念される2025年問題も迫ってきている。このような問題を解決する手段としてICTやセンシングシステムを使い遠隔地から高齢者を見守るサービスが増えている。しかしこの見守りサービスの多くは高齢者の安否や健康状態を見守る側に伝えるもので、高齢者の意思を伝えるものではない。

現在では様々な高齢者を見守るサービスが提案されており、例えば、象印マホービン株式会社が2001年3月より提供している「みまもりほっとライン」[1]がある。無線通信を内蔵した電気ポットを使用するだけで離れて暮らす家族の生活を見守ることが出来る。電気ポットの使用状況を、見守る家族の携帯電話やパソコンにEメールで知ることが出来るほかに、ホームページの契約者専用ページで1週間のポット使用状況をグラフで見ることが出来る。株式会社オートバックスセブンが2019年10月より提供している対話型AIサービスを搭載したロボット「ZUKKU」[2]は対話機能を搭載し、AIによる自然な会話を行うことができる。カメラ機能は搭載せずプライバシーにも配慮しており、人感センサ

を搭載することで、人の動きを感知するため離れて暮らす高齢者の見守りを行う。伝言機能では、離れて暮らす家族がテキストでメッセージを送ると、ZUKKUがそれをしゃべって知らせる。受けた側からメッセージを返信することもできる。

本研究の目的は、高齢者を見守ることに寄り添いつつもお互いの利便性を向上させることができる、見守る側はテキストで見守られる側は声を使うコミュニケーションツールを提案することである。遠隔地の人物とコミュニケーションをとる方法は、メール、手紙、チャットツールなどのテキストでのコミュニケーションと、電話などの声でのコミュニケーションの2つがある。テキストでのコミュニケーションのメリットは、時間的空間的拘束が少ないことと、および記録に残ること、デメリットは、声や表情からの情報がなく、無機質なやり取りになってしまうことで、誤解が生まれやすいことである。次に、声でのコミュニケーションのメリットは、感情的な情報を伝えられることで、デメリットは時間的空間的拘束があることと記録が残らないことである。高齢者を離れた場所から見守りたい家族を見守る側、高齢者を見守られる側とする。見守る側は、仕事など見守ること以外に時間が取られる場合が多い、よって時間的空間的拘束が少なく、および記録に残すことができるテキストのコミュニケーションが向いている。無機質なやり取りに

* 愛知工科大学工学部情報メディア学科, 〒443-0047 愛知県蒲郡市西迫町馬乗 50-2

Department of Media Informatics, Aichi University of Technology, 50-2 Manori, Gamagori 443-0047, Japan

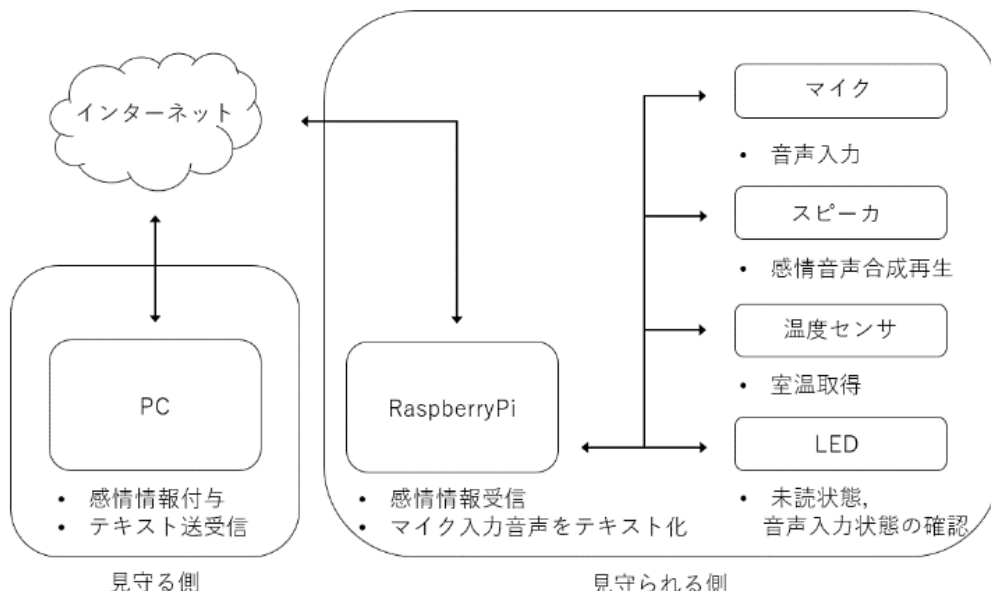


Fig. 1 Proposed communication tool.

なるというデメリットは感情音声合成を用いることで解決する. 見守られる側は, テキストより感情的な情報を受け取れる会話のほうが見守られている安心感があるため声のコミュニケーションが向いている. 感情合成音声聞き, 話したことを音声認識でテキスト化することで会話を再現する. また, 第 1 著者は本学 IoT モノづくりコース第 1 期生ということもあり, そのコースで学んだことも生かし, より良いツールの実装を目指す.

以降の本論文の構成について述べる. 2 節では提案するツールの概要, 実装方法を論じ, 3 節では見守る側のツール, ユーザインタフェースを論じ, 4 節では見守られる側のツール, インターフェイス, 音声認識エンジン, 感情音声合成エンジンを論じ, 5 節では評価実験について述べ, まとめる.

2. 提案するコミュニケーションツール

2.1. 概要

Fig. 1 に提案する見守りコミュニケーションツールのイメージを示す. 見守る側と見守られる側が, より良いコミュニケーションを行うためには, 一方のコミュニケーションツールに合わせるのではなく, それぞれが便利だと思うツールを使用する必要がある. それらの要件を満たすために提案するツールでは, 見守る側はテキストを送信し, 見守られる側は受信したテキストを合成音声で再生し, 音声認識で返信内容を入力する. そのため, 見守る側はキーボードを備えた PC のみだが, 見守られる側はマイクロホンとスピーカを装備する. また, 見守る側において, 送信するテキストに感情を付与できるようにし, 見守られる側では, 付与された感

情に対応する感情合成でテキストを読み上げる仕組みを導入する. これにより, 見守る側は伝えたい内容をより印象深く伝えることができ, 見守られる側は会話に近いコミュニケーションができ, 見守られている安心感が得られる.

LED は高齢者がツールの画面のみではわかりづらいことを考慮し装備した. 未読テキストがある場合に点灯するものと, 音声認識中のみ点灯するものがある. これにより, テキストの受信に気づかず未読の状態が続くことの防止と, 音声認識の言葉を話すタイミングが分かりやすくなる.

温度センサは室内での熱中症など室温に関する事故防止のために装備した. 30°Cを超えると警告が出て見守る側へテキストを送信する. これにより, 室内での熱中症対策になるほかに一時間ごとに記録される室温を見守る側が確認することで, 見守られる側の健康を気遣うことができる.

2.2. 実装方法

見守りコミュニケーションツールでは, Android, iOS, Linux, Windows で動作させることができる Python のマルチタッチアプリケーション開発のためのオープンソースライブラリである Kivy[3]を用いて実装した. Kivy は Python のコードとは別に, Kv Language と呼ばれる独自の言語を用いてボタンやラベルなどのウィジェットをデザインすることができる. Table 1 に見守る側の実装環境, バージョン番号を Table 2 に見守られる側の実装環境, バージョン番号を示す.

Table 1 Watching-side devices.

名目	使用物	バージョン
フレームワーク	Kivy	2.0.0
実装言語	Python	3.7.3
クライアント OS	Windows 10	1909

Table 2 Watched-side devices.

名目	使用物	バージョン
フレームワーク	Kivy	2.0.0
実装言語	Python	3.7.3
クライアント OS	Raspbian	10

2.3. Raspberry Pi

見守られる側のデバイスは、Raspberry Pi 2 Model B[4]を使用した。Fig. 2は実際に使用したRaspberry Pi 2 Model Bの外観である。Raspberry Piとはイギリスのラズベリーパイ財団によって開発された、ARM プロセッサを搭載したシングルボードコンピュータである。Raspberry PiはGPIO (GeneralPurposeInputOutput) ピンを備えておりモータやセンサを接続し制御することができる。また、HDMI, USB, LAN ポートなどのインターフェイスも標準搭載されており、拡張性が高いためIoT デバイスや組み込み機器を作成することができる。作成したツールではUSBポート, AV端子, GPIOピンを使用しマイクスピーカ, LED, 温度センサの制御を行った。

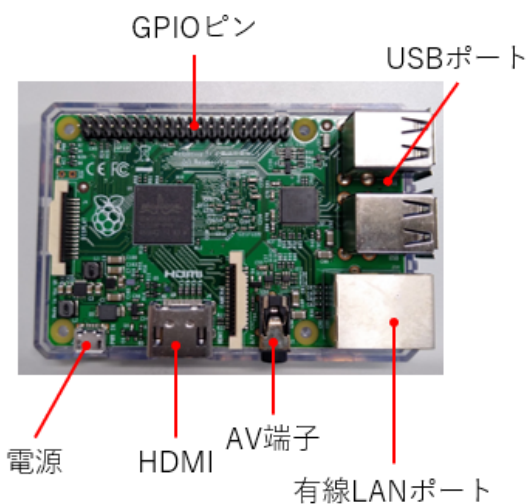


Fig. 2 RaspberryPi 2 Model B

3. 見守る側のツール

この節では、見守る側ツールの説明を行う。3.1 節ではテキストを送受信するシステムを論じ、3.2 節では

見守る側のユーザインタフェースを論じ、3.3 節では見守る側の送信内容を論じ、3.4 節では Google スプレッドシートに記録する機能を論じる。

3.1. テキストを送受信するシステム

テキストの送受信は WebSocket[5]を使用する。WebSocket はコンピュータネットワーク用の通信規格の 1 つである。従来の HTTP を使うクライアントとサーバ間の双方向通信はポーリングや comet という手法を用いることで実現していた。しかし、この手法は、新しくコネクションを行う必要があり、また、HTTP コネクションを長時間占領するため、その間にサーバに接続する他のアプリケーションの動作に影響を与える可能性がある、などの問題があった。

Fig. 3 が WebSocket を使用した通信のイメージである。WebSocket では、サーバとクライアントが最初に HTTP 通信によってコネクションを行った後、必要な通信全てを WebSocket プロトコルにより行う。この手法を使うことで新しくコネクションを行うことがなくなることや、データの送受信が一つのコネクションで行われるので、同じサーバに接続する他のアプリケーションへの影響が少なくなること、さらに HTTP コネクションより軽量なプロトコルを使うことで通信ロスが減る、といったメリットがある。

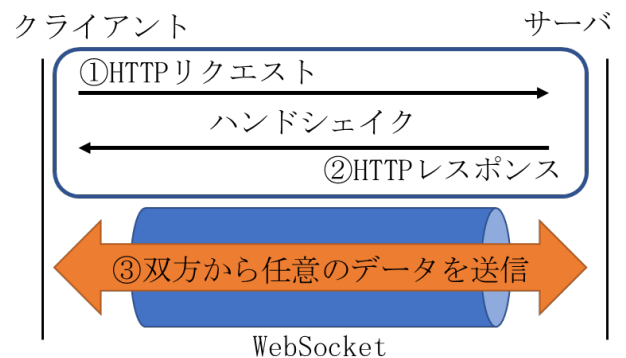


Fig. 3 Communication by WebSocket.

3.2. ユーザインタフェース

Fig. 4 が実際の見守る側ユーザインタフェースのキャプチャ画像である。画像の上半分が見守る側と見守られる側のやり取りになっており、左半分に見守る側、自分が送信したテキストが表示され、右半分に相手とサーバから送信されたテキストが表示される。中央より下にテキストを入力

する箇所があり，右の送信／既読ボタンでテキストを送信する．テキストに何も入力しない状態で送信／既読ボタンを押すと相手のメッセージを既読状態にすることができる．自分が既読状態にした場合は，Fig. 5のように，相手から受信した緑色のテキストが白色に変化する．また，相手が既読状態にした場合は，Fig. 6のように，自分が送信した青色のテキストが白色に変化する．既読の機能を実装することで相手の安否を確認することができる．また，視覚的に理解しやすくなるよう，色の変化で既読の確認ができるようにした．画像下部にテキストに付与する感情を選択する機能があり，左右のボタンを押すことで normal, happy, sad, angry の 4 種類を選ぶことができる．



Fig. 4 User interface of the watching side.

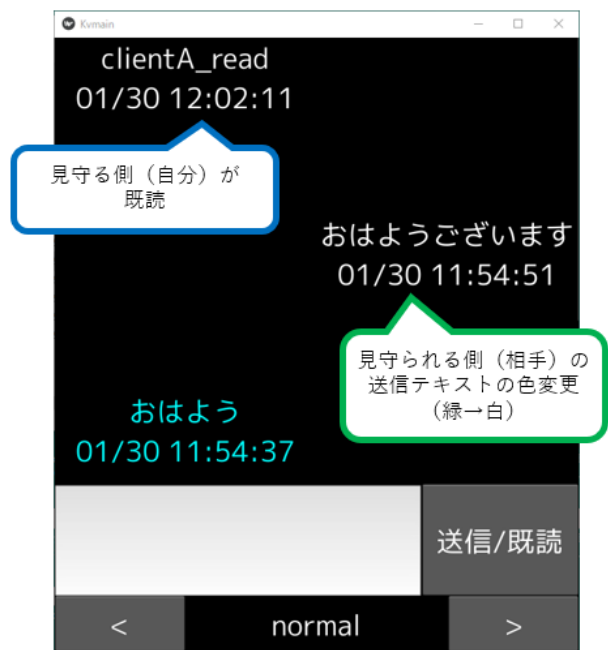


Fig. 5 Screen display after the watching person read the message.



Fig. 6 Screen display after the watched person read the message.

3.3. 送信内容

見守る側からの送信内容は 3 つで，

- テキスト
- IP アドレス
- テキストを再生する時の感情

である．Fig. 7 は実際にサーバに送信された内容


```

pi@raspberrypi:~ $ python3 ws_server2.py
INFO:websocket_server.websocket_server:Listening on port 9001 for clients..
new client connected and was given id 1
new client connected and was given id 2
client(1) said: 送信内容のテキスト@192.168.0.13@normal@
client(1) said: うれしいです@192.168.0.13@happy@
client(1) said: 悲しいです@192.168.0.13@sad@
client(1) said: 怒っていません@192.168.0.13@angry@
client(1) said: clientA_read@192.168.0.13@normal@
client(2) said: 音声で入力してください@127.0.1.1@emotion@
client(2) said: 20.74℃@127.0.1.1@emotion@
client(2) said: clientB_read@127.0.1.1@emotion@
INFO:websocket_server.websocket_server:Client closed connection.
client(1) disconnected
WARNING:websocket_server.websocket_server:Client must always be masked.
client(2) disconnected

```

Fig. 7 Messages received by the server.

で、3つの送信内容は@で区切り一度に送信している。Fig. 8に既読機能のイメージを示す。テキストに何も入力しない状態で「送信/既読」ボタンを押したとき「clientA_read」というテキスト内容が送信される。この内容をサーバから受信すると既読機能が使用でき、見守る側では、見守られる側が送信した緑色のテキストが白色に変更され、見守られる側では、見守る側に送信した青色のテキストが白色に変更される。

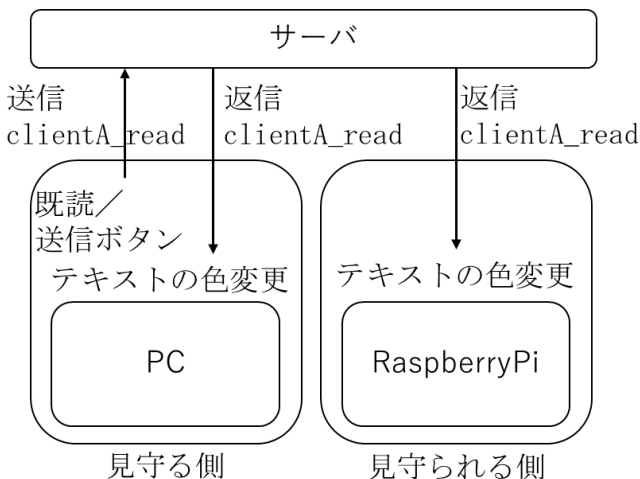


Fig. 8 Flow when the watcher reads a message.

3.4. Google スプレッドシートに記録する機能

さらに、やり取りを Google スプレッドシートに記録す

る機能を実現した。Python で Google スプレッドシートを操作することができる gspread[6]ライブラリを用いた。ツールを起動または、テキストを送受信したタイミングでスプレッドシートに現在の日付のシートがない場合は、現在の日付のシートを作成し、内容が記入される。やり取りを記入した日付と時間、見守る側が送信したテキストと既読した時に送信される clientA_read が記入される。また、見守る側が送信したテキストと既読した時に送信される clientB_read も記録される。見守る側のテキストの送信タイミングと見守られる側の既読タイミングを知ることができ、長時間既読が付かなかった場合に見守られる側の異変に気付くことができる。また、1時間ごとに送られてくる見守られる側の室温も記録され、一日の室温の変化を知ることが出来る。これによって見守る側は、見守られる側の熱中症など室温による事故と健康により気を付けることができる。

4. 見守られる側のツール

この節では、見守られる側ツールの説明を行う。4.1節では音声認識システムを論じ、4.2節では感情音声合成を論じ、4.3節では見守られる側のユーザインタフェースを論じ、4.4節では見守られる側の送信内容を論じる。

4.1. 音声認識 Google Search API

音声認識エンジンには、Google Speech API[7]を用いる。Google Speech APIは録音した音声ファイルをAPIに入力することで音声に対応する自

然文を出力することが出来る．この API を利用するには，GoogleDeveloperConsols[8]での登録と Chromium-dev グループ[9]への加入が必要になる．Table 3 に録音条件を示す．

Table 3 Recoding conditions.

種類	値
音声フォーマット	16 ビット
チャンネル数	1
サンプリングレート	44100 Hz
データ点数	512

4.2. 感情音声合成 OpenJTalk

音声合成エンジンには，日本語テキスト音声合成システム OpenJTalk[10]を用いる．OpenJTalkは形態素解析部に MeCab[11]，発話辞書に NAUSRJapaneseDictionary[12]を用いた，音声合成には hts-engineAPI[13]を組み込んでいるシステムで，HTS で学習した音響モデルを扱うことが出来る．HTS には感情音声合成技術が組み込まれており，見守る側でテキストを再生する際に，感情に応じた合成音声を生産することができる．

4.3. ユーザインタフェース

Fig. 9 が実際の見守られる側ユーザインタフェースのキャプチャ画像である．画像の上半分が見守る側と見守られる側のやり取りになっており，左半分に自分が送信したテキストが表示され，右半分に相手とサーバから送信されたテキストが表示される．画像の下部分には時間，室温，音声認識結果が表示されている．その下には再生／既読ボタン，録音ボタン，送信ボタンがある．

再生／既読ボタンを押すと相手とサーバから送信されたテキストの中で最新のテキストが設定された感情で再生され，相手のテキストが既読状態になる．自分が既読状態にした場合は，Fig. 10 のように，相手から受信した緑色のテキストが白色に変化する．また，相手が既読状態にした場合は，Fig. 12 のように，自分が送信した青色のテキストが白色に変化する．録音ボタンを押すと，録音を開始し最終的に音声認識結果が表示される．音声認識に失敗した場合はもう一度ボタンを押すことで音声認識を開始することが出来る．送信ボタンを押すと音声認識結果の欄に表示されているテキストが相手に送信される．



Fig. 9 User interface of the watched side.



Fig. 10 Screen display after the watched person read the message sent by the watching person.

```

pi@raspberrypi:~ $ python3 ws_server2.py
INFO:websocket_server.websocket_server:Listening on port 9001 for clients..
new client connected and was given id 1
new client connected and was given id 2
client(1) said: 送信内容のテキスト@192.168.0.13@normal@
client(1) said: うれしいです@192.168.0.13@happy@
client(1) said: 悲しいです@192.168.0.13@sad@
client(1) said: 怒っています@192.168.0.13@angry@
client(1) said: clientA_read@192.168.0.13@normal@
client(2) said: 音声で入力してください@127.0.1.1@emotion@
client(2) said: 20.74℃@127.0.1.1@emotion@
client(2) said: clientB_read@127.0.1.1@emotion@
INFO:websocket_server.websocket_server:Client closed connection.
client(1) disconnected
WARNING:websocket_server.websocket_server:Client must always be masked.
client(2) disconnected

```

Fig. 11 Messages sent by the watched person received by the server.

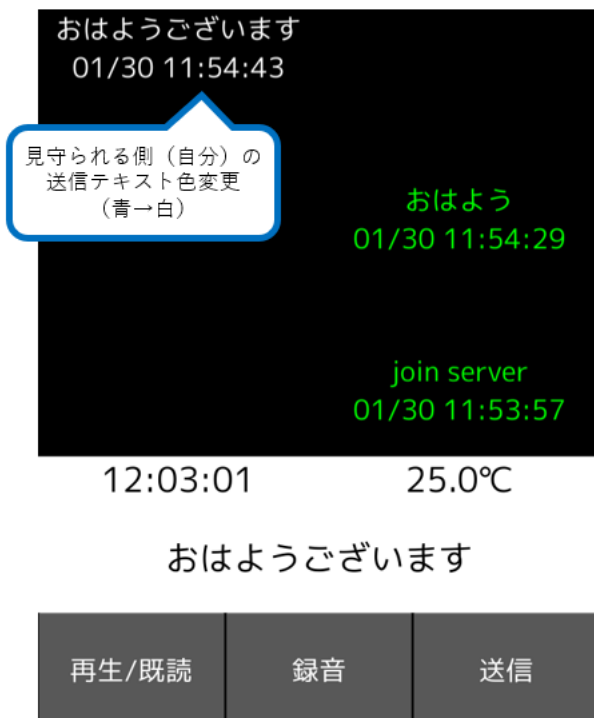


Fig. 12 Screen display after the watching person read the message sent by the watched person.

4.4. 送信内容

見守られる側からの送信内容は2つで

- テキストまたは室温
- IPアドレス

である。Fig. 11は実際にサーバに送信された内容で、2つの送信内容は@で区切り一度に送信している。Fig. 13に既読機能のイメージを示す。「再生／既読」ボタンを押したとき「clientB_read」というテキスト内容が送信される。この内容をサーバから受信すると既読機能が使用でき、見守られる側では、見守る側が送信した緑色のテキストが白色に変更され、見守る側では、見守られる側に送信した青色のテキストが白色に変更される。見守られる側から感情を送信する機能はないので、3つ目の内容はemotionで固定する。見守られる側から感情を送信する機能を実装するにあたり、自動で感情を推定し、付与できるものが望ましいと考えられる。例えば、音声感情解析AI「Empath」[14]というものがある。Empathは音声の物理的特徴量を解析し言語に依存せず、「喜び／平常／怒り／悲しみ」の4つの感情と元気を解析するサービスである。このようなものが組み込み可能であれば、実現可能である。今回は使用する要件に合わないため、具体的には検討していない。

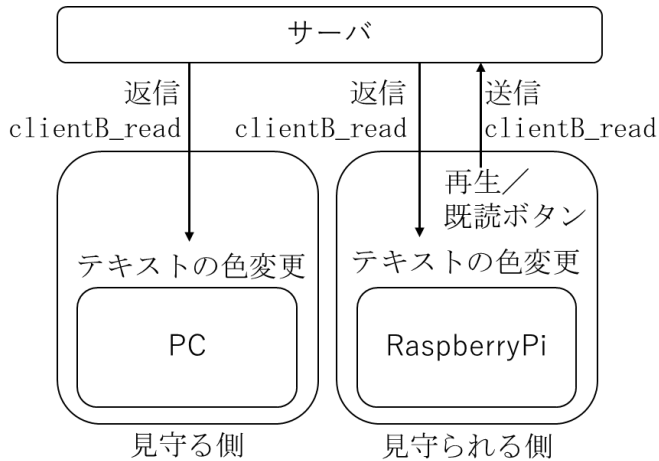


Fig. 13 Flow when the watched person reads a message.

5. 実験

この節では、コミュニケーションツールを評価するために行った実験の説明を行う。評価してもらうことでツールの有用性の確認やプログラムのバグの発見や修正を行うことができる。実験は、見守る側と見守られる側に分かれ、4種類の感情を順に使用したやり取りを行った後、見守る側と見守られる側を交代し、同じように4種類の感情を使用したやり取りを行う。5.1節では実験概要を述べ、5.2節では実験結果を論じる。

5.1. 実験概要

Table 4の(1)～(8)の項目について評価することを目的として、被験者に実際にツールを利用してもらう。Table 5はシステムを実際に使う実験のやり方を簡単にまとめたものである。実験は二人一組で行い、はじめに見守る側、見守られる側、両方のシステムの使い方について説明する。次に、見守る側、見守られる側に分かれてもらい、見守る側は感情 normal を使用してやり取りを行う。その後、感情 happy, sad, angry を順に使用してやり取りを行ってもらい見守る側、見守られる側を交替して同じように感情を使用してやり取りを行う。やり取りを終了した後に、Table 4の(1)～(8)の項目について5段階評価を行ってもらい。今回は10人5組に実験を行ってもらった。

Table 4 Questions for five grade evaluations.

(1) 感情の付与
(2) 音声入力
(3) 音声出力
(4) 感情合成音声
(5) 認識精度
(6) 再生された音声の聞きやすさ
(7) システム全体 (総合的) にどうか
(8) 高齢者向けとしてどうか

Table 5 Experimental procedure.

(1) システムの使い方を説明し、見守る側と見守られる側に分かれる
(2) 見守る側は感情 normal を使用してやり取りを行う
(3) 見守る側は感情 happy を使用してやり取りを行う
(4) 見守る側は感情 sad を使用してやり取りを行う
(5) 見守る側は感情 angry を使用してやり取りを行う
(6) 見守る側見守られる側を交替して(2)～(5)を行う

5.2. 実験結果

Fig. 14は見守る側ツールを使用している実験の様子である。ノートパソコンのキーボードでテキストを入力する。Fig. 15は見守られる側ツールを使用している実験の様子である。スピーカから感情合成音声を聞き、マイクに音声を入力する。Fig. 16はTable 4の(1)～(8)の項目をそれぞれ5段階評価してもらった各項目の平均値を表している。

見つかったバグや問題点として、音声認識中に再度、録音ボタンを押すとプログラムがエラーを起こしてツールが停止してしまうというものがあった。要因としてマイクを使用している最中に、もう一度マイクの使用を要求するプログラムになっているためだった。対応として音声認識が開始から終了までの間録音ボタンを押してもマイクの要求をしないようにプログラムを修正した。次に、音声認識にかかる時間が遅いという問題があった。これは音声の録音を10秒に設定していたが、被験者が話し終わるまでにかかる平均的な時間が約4.5秒であったため被験者は認識時間が長いと感じたのだと考えられる。この問題は音声の録音時間を10秒から7秒に短縮することで対

応した。見守る側のツールで、入力中の日本語のテキストが見ることができず使いづらい問題があった。これは、Kivyの文字入力を行う textinput ウィジェットでIMEが表示されていないためである。

実験後の意見として、音声認識はキーボードより入力が楽で簡単であるため高齢者に向いているという意見があった。音声認識は高齢者向けのツールとして向いていることが分かった。チャットアプリのようなテキストのみのやり取りの方が早くコミュニケーションできて良いという意見もあった。評価実験の被験者は全員 20 代で、キーボード操作に慣れているためこの意見が出たと考えられる。このことから、見守る側の世代はテキストのやり取りの方が向いていることが確認できた。見守る側が感情を付けることができるのは良いが実際に再生される感情合成音声の感情の差が小さいという意見があった。今後の課題として感情により差をつける必要がある。

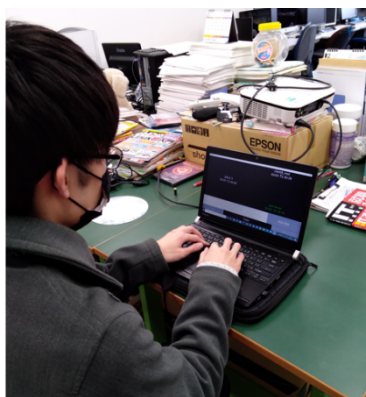
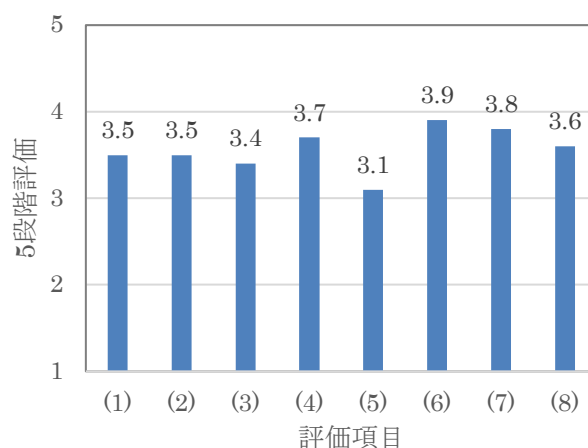


Fig. 14 A scene of an experiment using the proposed watching tool.



Fig. 15 A scene of an experiment using the proposed watched tool.



(1)	感情の付与
(2)	音声入力
(3)	音声出力
(4)	感情合成音声
(5)	認識精度
(6)	再生された音声の聞きやすさ
(7)	システム全体(総合的)にどうか
(8)	高齢者向けとしてどうか

Fig. 16 Experimental results for the proposed communication tool.

6. おわりに

本研究では、音声認識と感情合成音声を用いた高齢者用の見守りコミュニケーションツールを提案した。従来の見守りサービスは、高齢者側でセンシングした情報を見守る側に伝えるものの、高齢者の意思を伝えることはできないものや、テキストのやり取りは可能だが、感情情報については考慮していないものであった。

そこで、本研究では、高齢者を見守りつつ、お互いの利便性が向上させることができ、見守る側はテキストで、見守られる側は声を使うコミュニケーションツールを提案した。見守る側は、仕事など見守ること以外に時間が取られる場合が多い、よって時間的空間的拘束が少なく、および記録に残すことができるテキストのコミュニケーションが向いている。見守られる側は、テキストより感情的な情報を受け取れる会話のほうが見守られている安心感があるため声のコミュニケーションが向いている。無機質な合成音声による再生ではなく、感情合成音声により、見守り側のメッセージを見守られる側（高齢者側）で再生する。高齢者側からは、話したことを音声認識技術

により、テキスト化して見守り側へメッセージを送る。

見守る側と見守られる側のデータの送受信は WebSocket を使用した。WebSocket を使用することで、HTTP 通信によるハンドシェイク後はサーバとクライアントの双方向通信を低コストで行うことができる。見守られる側の音声認識エンジンは Google Speech API を使用した。この API は録音した音声ファイルを API に入力することで音声に対応する自然文を出力することが出来る。感情音声合成は OpenJTalk を使用した。HTS の感情音声合成技術を利用して作成された、平常(normal)、嬉しい(happy)、悲しい(sad)、怒り(angry)の4種類の感情が見守る側の設定に合わせて再生される。

このコミュニケーションツールの評価実験を10人5組の被験者に対して行った。5段階評価の結果、平均値はすべて3.0以上4.0以下となった。

「再生された音声の聞きやすさ」や「感情合成音声」を導入した点に対する評価が特によく、「システム全体(総合的)にどうか」が高い点数となった。「認識精度」が思いのほか低く、Googleのサービスを使ったのだが、現段階ではよい音声認識精度が得られなかったようである。実験後の意見から、音声認識によるテキスト入力は高齢者に向けており、キーボードによるテキスト入力は見守る側に向いていることが確認できた。

今後の課題としては、より音声認識の精度の向上をあげる必要がある。また、感情音声合成について、今回はオープンソースで公開されている OpenJTalk の機能をそのまま用いたが、見守られる側で再生される音声の感情に差異をもっとつけることができれば、より人間らしい声でのコミュニケーションにすることができる。見守る側のツールは、今回、Windows で動作させたが、Kivy は Android, iOS でも動作させることができるので、利便性を向上できるように、スマートフォン、タブレットで使えるようにしていきたい。

参考文献

- [1] みまもりホットライン,
<https://www.mimamori.net/>(アクセス日:2021年1月30日)
- [2] ZUKKU,
<https://hatapro.co.jp/group/robotics/> (アクセス日:2021年2月4日)
- [3] Kivy, <https://kivy.org/#home> (アクセス日:

2021年1月30日)

- [4] Raspberry Pi 2 Model B,
<https://www.raspberrypi.org/> (アクセス日:2021年1月30日)
- [5] WebSocket,
<https://html.spec.whatwg.org/multipage/web-sockets.html#network> (アクセス日:2021年1月30日)
- [6] gspread,
<https://gspread.readthedocs.io/en/latest/>
(アクセス日:2021年1月30日)
- [7] GoogleSpeechAPI,
<https://cloud.google.com/speech-to-text/docs/quickstart-protocol> (アクセス日:2021年1月30日)
- [8] GoogleDeveloperConsoles,
<https://console.developers.google.com/> (アクセス日:2021年1月30日)
- [9] Chromium-dev グループ,
<https://groups.google.com/a/chromium.org/forum/?fromgroups#!forum/chromium-dev> (アクセス日:2021年1月30日)
- [10] OpenJTalk,
<http://open-jtalk.sp.nitech.ac.jp/> (アクセス日:2021年1月30日)
- [11] MeCab,
<http://taku910.github.io/mecab/> (アクセス日:2021年1月30日)
- [12] NAUSRJapaneseDictionary,
<http://naist-jdic.osdn.jp/> (アクセス日:2021年1月30日)
- [13] hts-engineAPI,
<http://hts-engine.sourceforge.net/> (アクセス日:2021年1月30日)
- [14] Empath,
<https://webempath.com/jpn/> (アクセス日:2021年2月5日)