

2009年1月30日

卒業研究論文

メタコミュニケーション機能を持つ エージェントの開発

大阪工業大学情報科学部

ヒューマンインタフェース研究室

C05-043 坂尾 和也

C05-122 山崎 敏文

目次

1. はじめに.....	1
2. ヒューマンコミュニケーション.....	2
2. 1 バーバル・ノンバーバルコミュニケーション.....	2
2. 2 対面コミュニケーション.....	2
3. 1 会話エージェントの定義.....	3
3. 2 会話エージェントの実験手順.....	3
3. 3 関連研究.....	5
4. アプリケーションの概要.....	5
4. 1 会話行動の収集・分析.....	5
4. 2 システム構成.....	6
4. 4 CamSpace.....	7
4. 5 トラッキングフェイズ.....	8
4. 5. 1 トラッキングモジュールの設計.....	9
4. 5. 2 注視判定.....	11
4. 6 対話フェイズ.....	13
4. 6. 1 音声入力モジュールの設計.....	13
4. 6. 2 うなずき動作.....	15
5. インタラクション評価.....	17
5. 1 インタラクション評価方法.....	17
5. 2 インタラクション評価結果.....	18
6. 考察.....	18
6. 1 ユーザヒアリング結果.....	18
6. 2 精度の向上.....	18
6. 3 うなずき以外のメタコミュニケーション動作.....	19
7. おわりに.....	19
謝辞.....	19
参考文献.....	20
付録.....	21

1. はじめに

近年，ソフトウェアエージェントは，一般ユーザに普及し始めてきた[1]．普及に伴い人とエージェント間のインタラクションデザインの研究や開発の分野であるヒューマンエージェントインタラクション HAI (Human-Agent Interaction) が広がりを見せている[1]．HAI が扱うエージェントには，「自律エージェント」，「マルチエージェント」，「インタフェースエージェント」，「会話エージェント」など様々な種類がある[1]．本研究では，対面での人とインタラクションを行う「会話エージェント」を扱う．

会話エージェントとは，人同士の対面での会話を人とコンピュータのインタラクションに取り入れたシステムである．人同士の対面会話を模倣するために，会話エージェントは「バーバルコミュニケーション」と「ノンバーバルコミュニケーション」を行う必要がある．バーバルコミュニケーションは，音声を用いたコミュニケーションであり，ノンバーバルコミュニケーションは音声によらないジェスチャなど身体を用いたコミュニケーションである．またノンバーバルコミュニケーションには，人同士の対面コミュニケーションにおいてコミュニケーションを円滑に行うためにコミュニケーションの内容の規定や進行を調整するといった重要な機能である「メタコミュニケーション」がある．しかし現在のエージェントに関する研究は，エージェントの音声に合わせたジェスチャの自動生成といったものが中心課題となっており，双方向的なインタラクションの研究はあまり行われていない[3]．

そこで本研究では，双方向インタラクションのためにメタコミュニケーションに重点を置き，会話の間に「うなずき」を行う会話エージェントの開発を行った．本エージェントは，ユーザ位置を追跡する「トラッキングフェイズ」，エージェントと対話を行う「対話フェイズ」に分けられる．トラッキングフェイズの目的は，ユーザとエージェントが会話の開始を行うためである．対話フェイズの目的は，ユーザの発話終了にうなずくことによりコミュニケーション維持を行うためである．また本エージェントに実装したモデルの有効性，システムの有用性を確認するために，うなずきの有無やタイミングの異なる動作を行うエージェントを用意しインタラクション評価を行う．

第 2 章では，ヒューマンコミュニケーションであるバーバルコミュニケーション，ノンバーバルコミュニケーションについて述べる．次にノンバーバルコミュニケーションで重要な機能であるメタコミュニケーションについて述べる．第 3 章では，会話エージェントの定義と実験手順について述べる．第 4 章は，本研究で開発したメタコミュニケーションに重点をおいた会話エージェントの設計や開発について述べる．第 5 章は，開発したエージェントを実際にユーザとインタラクションしてもらう方法や結果について述べる．第 6 章にインタラクション評価で得られた結果から本エージェントのインタラクションについての考察を述べた後に第 7 章で本研究のまとめを行う．

2. ヒューマンコミュニケーション

本章では、人が行うコミュニケーションについてどのようなものを述べる。

2. 1 バーバル・ノンバーバルコミュニケーション

人が行うコミュニケーションとしてバーバルコミュニケーションとノンバーバルコミュニケーションがある。バーバルコミュニケーションは、音声言語を用いて行うコミュニケーションである。対してノンバーバルコミュニケーションとは、音声言語によらない視線、表情や身振り、手振りといった身体を用いて行うコミュニケーションである。バーバルコミュニケーションは、1チャンネル、1方向性などといった性質をもつ[2]。ノンバーバルコミュニケーションは、多チャンネルかつ同時双方向性などといった性質をもつ[2]。

2. 2 対面コミュニケーション

対面コミュニケーションとは、人と人が相互に向かい合って行われるコミュニケーションである。対面コミュニケーションで行われるノンバーバルコミュニケーションの機能を次の表 1 に示す[3]。

表 1 対話におけるノンバーバル機能

会話における機能	ノンバーバル行動
発話強調	ジェスチャ, 眉を上げる表情
発話権移譲	次発話者へ視線を向ける
話し手へのフィードバック	相互注視, あいづち, 聞き手のうなずき
話題転換	姿勢変化

また、対面コミュニケーションにおいては、「発話権」と呼ばれる発話する権利の受け渡しを円滑に行うことがコミュニケーションを維持する上で重要である。この発話権の受け渡しにおいてノンバーバルコミュニケーションの重要な機能として「メタコミュニケーション」機能がある。

2. 3 メタコミュニケーション

メタコミュニケーションは、コミュニケーション内容の規定や進行を調整する働きを持つ。つまり、現在行われているコミュニケーションに関してのコミュニケーションを指す。メタコミュニケーションは、主に身振りや表情、視線が用いられており、会話の開始、発話の交替、会話の終了、発話の枠組みがある。それぞれの説明を以下に示す[2]。

(a) 会話の開始

「互いに相手を見る」や「挨拶をする」などの会話の開始行動が含まれる。具体的には、相手を認めたとき眉が上がるといった「眉のひらめき」と呼ばれる挨拶動作などの無意識行動[2]、相手と視線一致させ手を上げる・うなづくなどの動作である[10]。

(b) 発話の交替

発話権の受け渡しは、コミュニケーションを維持するために必要である。主に発話権の移譲信号、要求抑制信号、維持信号、維持指示信号などに区別される。具体的に移譲信号、要求抑制信号は手振り、身振りが用いられる。維持信号、維持支持信号は話し手が相手の顔を見たり、また受け手がうなずいたりすることで行われている。

(c) 会話の終了

会話の終了をする合図として、立ち話では片側の足への体重移動を行う。座位では、視線の交差を終える、膝や足をドアなどの出口に向けるといった合図がある。立ち話、座位の両方で前傾姿勢やうなずきといった会話終了の動作が行われている。

(d) 発話の枠組み

発話の意図、話し手と聞き手の関係、発話内容の真偽などのノンバーバルコミュニケーションによって伝達される。表情に比べてノンバーバル動作は、真偽の区別が容易である。

これら人の行うコミュニケーションは、会話エージェントが人とインタラクションを行うために考慮しなければならない。本研究では、会話の開始と発話の交替における維持信号に注目し会話エージェントに実装する。会話エージェントについては第3章で説明する。

3. 会話エージェント

本章では、会話エージェントの説明を行い、会話エージェントの研究方法を述べ、現状の問題点を述べる。

3. 1 会話エージェントの定義

会話エージェントとは、人同士の対面コミュニケーションを人とコンピュータに置き換えたものである。人同士の対面コミュニケーションを模倣させるためには、人が行うバーバルコミュニケーションやノンバーバルコミュニケーションの能力が必要である[3]。

3. 2 会話エージェントの実験手順

会話エージェントの実験手順は、人のコミュニケーション行動を模倣するために図 1 に

示すように4つの段階がある[3].

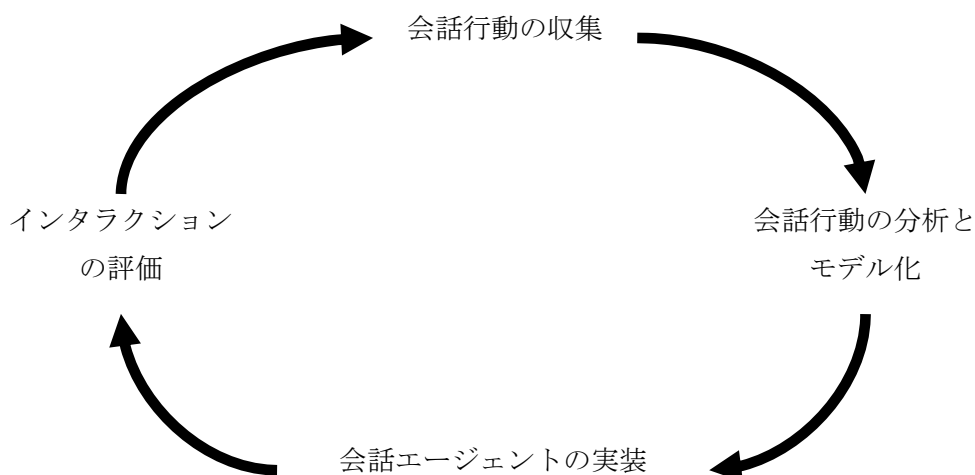


図 1 会話エージェントの実験手順

図 1 に示す各段階の説明は,

- (1) 会話行動の収集
会話エージェントを設計する基礎データとなる人同士のコミュニケーションを収集する.
- (2) 会話行動の分析・モデル化
収集した基礎データを, 発話音声の書き起こしやビデオ撮影をした身振りなどのアノテーションを行う. このデータを分析することによりエージェントに実装するバーバル・ノンバーバルコミュニケーションをモデル化する.
- (3) 会話エージェントの実装
データ分析から得られたモデルをエージェントに実装する.
- (4) インタラクション評価
実際にユーザとエージェント間でインタラクションを行う. その結果からエージェントのコミュニケーションへの円滑性や自然性が向上しているかなどの評価を行い, エージェントに実装したモデルの有用性とシステムの有用性を評価する.

以上の会話エージェントの実験手順に従い, 本研究のエージェント開発を行う.

3. 3 関連研究

対面コミュニケーションを行うエージェントの関連研究として、石井ら[4]が行ったエージェントを用いた音声チャットシステムがある。この研究では、エージェントの視線行動によりコミュニケーションの促進が可能であるという結果が得られている。また中野[5]の行ったユーザの視線情報を用いた会話エージェント研究では、ユーザの視線情報というノンバーバルコミュニケーションを用いたことによりユーザの負荷を軽減し、ユーザとエージェント間で自然なインタラクションが実現することが示唆されている。同様の結果が知野ら[6]の研究でも支持されている。また中野、知野らの研究ではエージェントにメタコミュニケーションを実装する重要性が説かれている。

このため、本エージェントもメタコミュニケーションを実装する。実装内容は、次の第4章に示す

4. アプリケーションの概要

本アプリケーションは、認識範囲内のユーザ位置を認識しユーザとの相互注視を行い、話を開始する。開発には Microsoft Visual C++[7]および DirectX SDK[8]を、エージェントのモデリングには Metasequoia[9]を使用した。

4. 1 会話行動の収集・分析

本研究では、会話行動の収集や会話行動の分析に関しては従来研究を参考に会話エージェントに実装する。

会話行動の収集・分析において Kendon による挨拶行動の分析を参考にする。Kendon らは、会話の開始に関して相手に視線を向けお互いを認識する。相手を認識したことを示す行動は、うなずきや手を振るノンバーバル動作を行う。一連のステップが会話の開始に存在する[10]。この相手の認識を行い会話の開始までの一連のステップを本研究ではトラッキングフェイズとする。

次に会話が始まると会話を維持する過程のために注視行動を図 2 に示す。注視行動は会話への参加意思を示し、相手へのフィードバックとなる。この会話を維持するステップを本研究では対話フェイズとする。

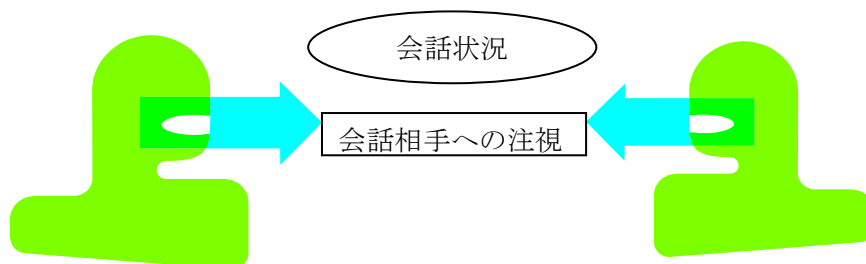


図 2 会話における注視行動

4. 2 システム構成

本アプリケーションは、マウス座標に追従し視線を移動させるトラッキングフェイズ、ヘッドセットマイクから入力される音声に従い動作する対話フェイズの二種類から構成される。さらに実際のトラッキング動作には Web カメラとトラッキングソフト「CamSpace」[11]を組み合わせる。にその概念図を図 3 示す。

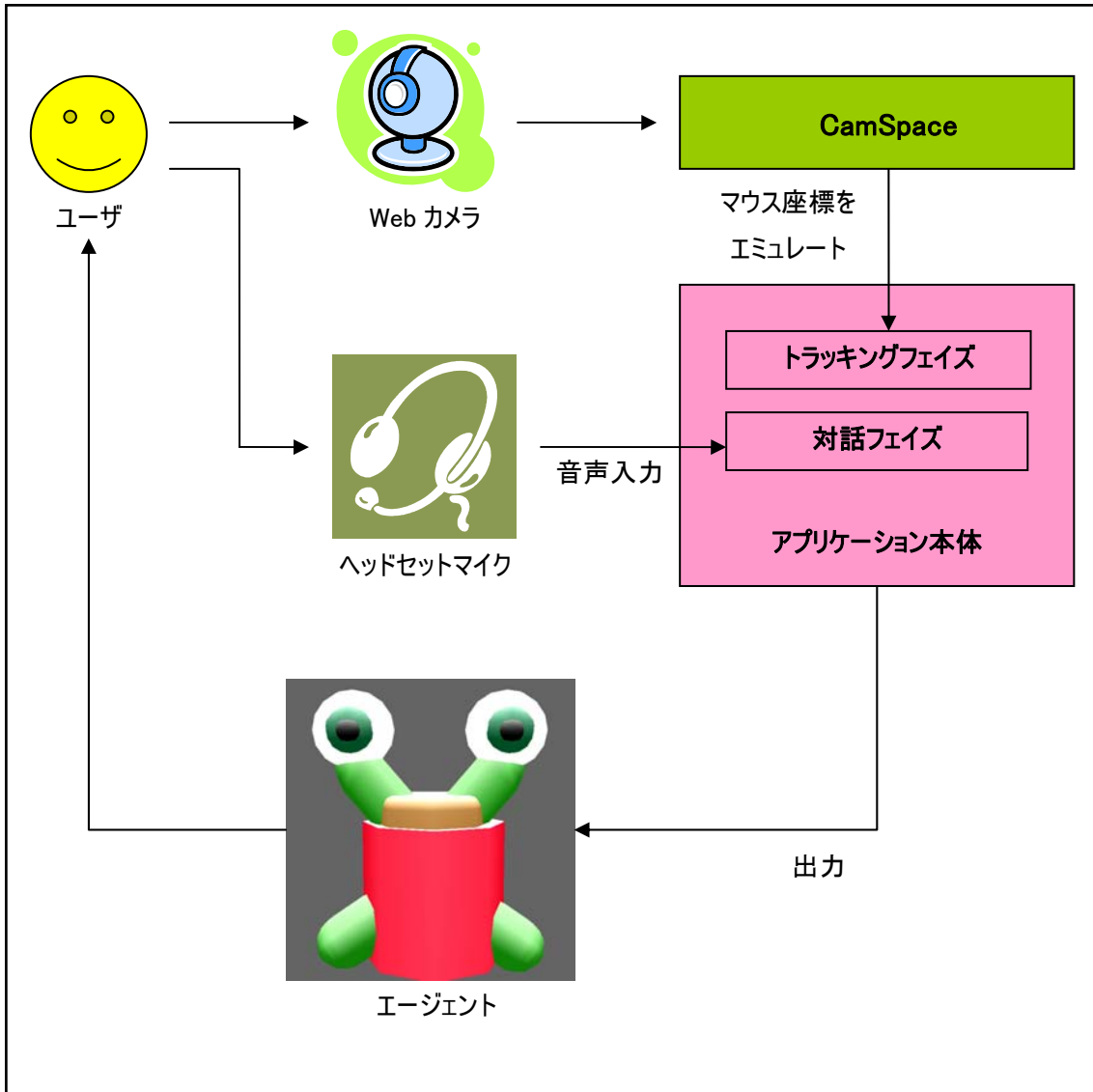


図 3 システム概念図

4. 3 使用するエージェントについて

本アプリケーションで使用するエージェントには、図 4 のような形状を用いた。

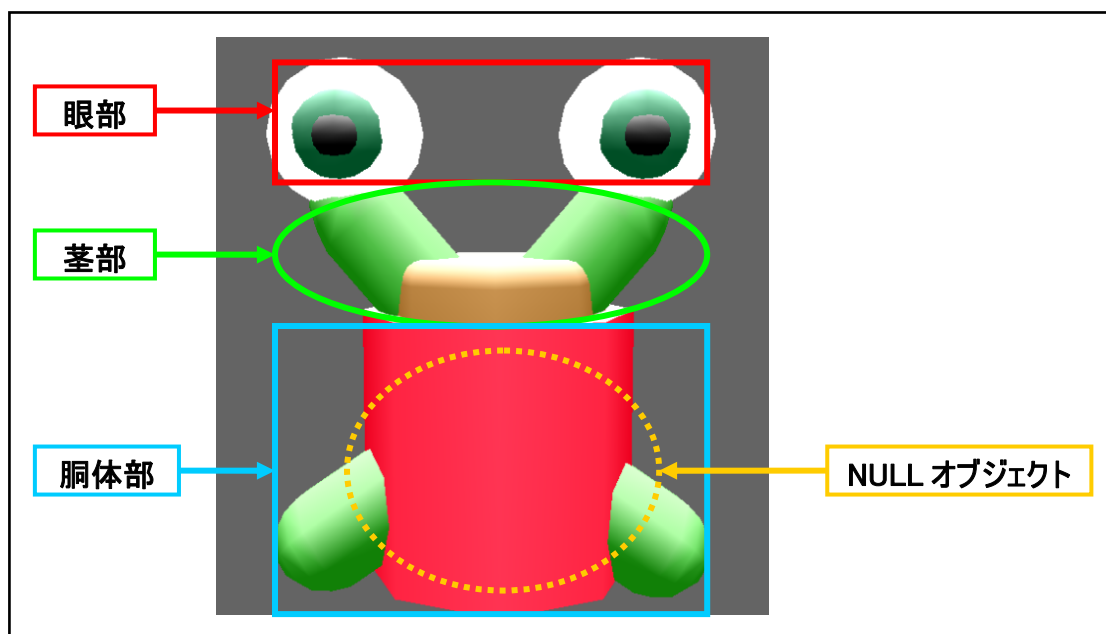


図 4 使用するエージェントの構成

本エージェントは、眼部・茎部・胴体部より構成される。眼部・茎部は X 軸・Y 軸を中心に回転する二自由度を有している。また、本アプリケーションでは主に視線に着目しているため目を強調した形状をとり、同時に眉・口・手などといった要素をできるだけ排除することで、人とエージェントとのインタラクション時に生じる適応ギャップを小さくする働きも持つ。

適応ギャップとは、エージェントの外見に基づいて期待される機能と、実際にエージェントに実装されている真の機能との差異のことである。人とエージェントの継続的なインタラクションを構築するためには、エージェントの外見から人によってモデル化される機能が実際の機能よりも低くなるように外見を設計する必要がある[12]。

(NULL オブジェクトについては 4. 5. 2 節 注視判定にて後述する。)

4. 4 CamSpace

CamSpace[11]は、CamTrax Technologies 社が開発したフリーのトラッキングソフトである。ほぼ全てのパソコン用 Web カメラに対応し、最大 4 つの物体をリアルタイムで追跡できる。ロッキングとトラッキング (X, Y, Z 軸方向および角度) はすべて自動で行われ、マウスやキーボード、ジョイスティックなどの入力デバイスのエミュレーションを行う。本アプリケーションでは、トラッキングによるマウスのエミュレーション機能を利用

し、次節にて説明するトラッキングフェイズと組み合わせることで、ユーザ位置のトラッキングを実現する。

4. 5 トラッキングフェイズ

トラッキングフェイズでは、エージェントの視線をマウスカーソルに追従させる。同時にオブジェクトの衝突を判定することにより、ユーザ視線とエージェント視線の注視判定を行う。この後プログラムを 1/60 秒停止させることにより、動作が遅くなることを防いでいる。注視判定時に必要な時間の判定も同時に行う。図 5 にトラッキングフェイズの流れを示したフローチャートを示す。

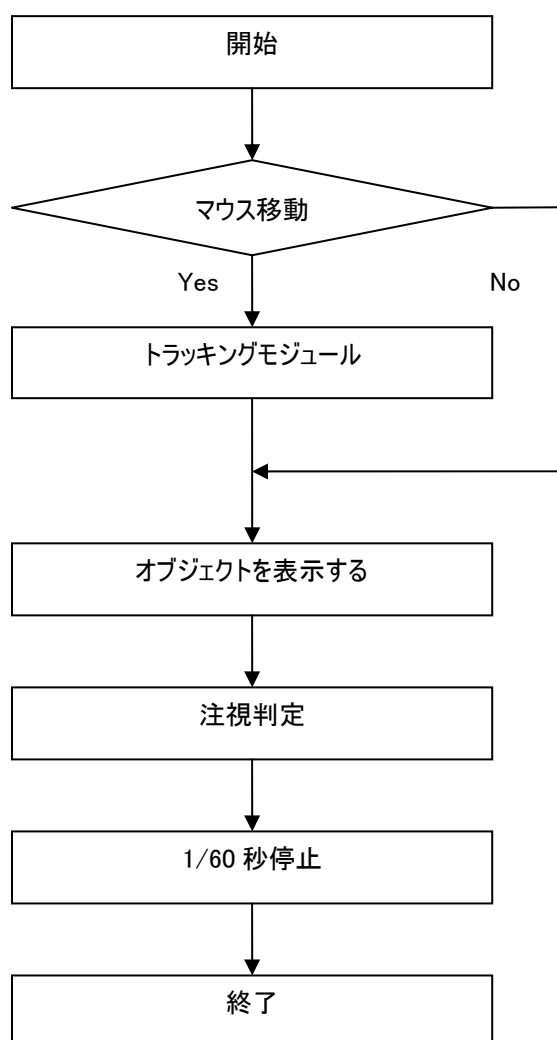


図 5 トラッキングフェイズの流れを示したフローチャート

4. 5. 1 トラッキングモジュールの設計

このモジュールでは、マウスの動作を元にエージェントの視線方向となるオブジェクトの座標を実際に決定する。まずマウスが動作することにより Win32API のメッセージである WM_MOUSEMOVE メッセージが送られてくる。このメッセージは LPARAM という引数を持っており、この引数にはモニタ上のマウス移動先の座標が格納されている。引数 LPARAM の下位 2 バイトが X 座標、上位 2 バイトが Y 座標となっているので、これらの情報を LOWORD マクロおよび HIWORD マクロを用いて取り出すことにより、マウス座標の実際の値を得ることができる。ただしこの値は「マウス移動先の座標」であり、「マウスを動作させたことにより移動した量」を知ることはできない。そこで、直前のマウス座標を格納する変数 prevX, prevY を用意し座標を格納する。現在のマウス X, Y 座標の値と、直前のマウス座標 prevX, prevY の値の差をとることにより、マウスの移動量を測定することが可能である。

次に実際のトラッキング時における処理を考える。ユーザが移動しなければ、CamSpace によってエミュレートされるマウスの移動量は 0 である。ユーザの位置は変わらないのでエージェント視線の位置は変更する必要がない。よって現在のマウス座標を変数 prevX, prevY に格納しトラッキングモジュールから抜ける。ユーザが移動すると CamSpace によりトラッキングが行われ、エミュレートされたマウスが移動する。ユーザが位置を移動したのでエージェント視線は追従する必要がある。このときマウス移動量を変数 dx, dy へと格納する。D3DXToRadian マクロを用いて変数 dx, dy をラジアンへ変換し、この値を元エージェント眼部の縦 (X 軸) 回転、横 (Y 軸) 回転および基部の横 (Y 軸) 回転を行う。これによりエージェント視線を画面上マウスカーソルに追従させ、ユーザ位置のトラッキングを実現する。最後に現在マウス座標を変数 prevX, prevY に格納しトラッキングモジュールから抜ける。以上がトラッキングモジュールの設計である。これらの一連の処理を示したフローチャートを図 6 に示す。

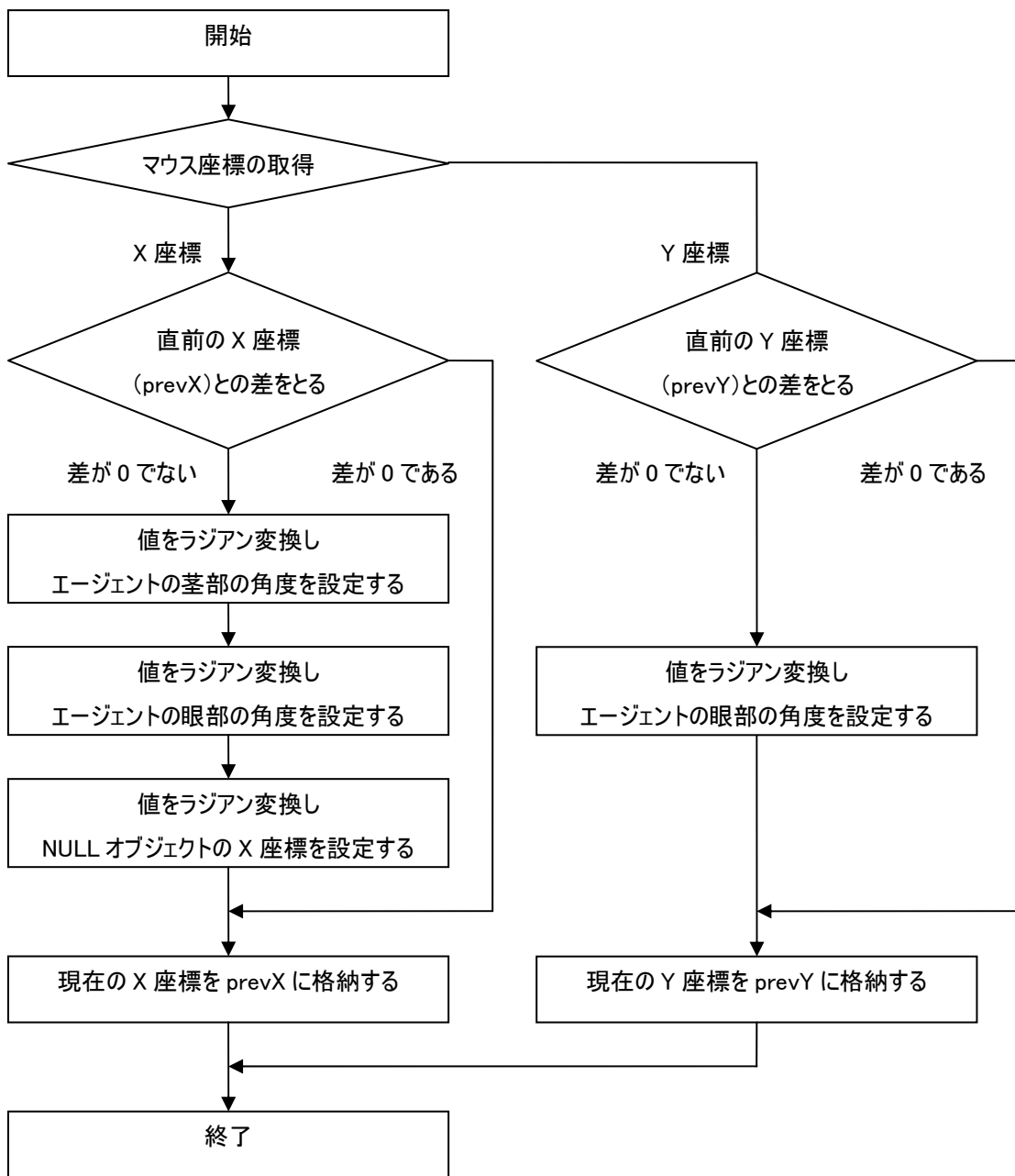


図 6 トラッキング処理のフローチャート

4. 5. 2 注視判定

前節のトラッキングにおいて、3次元的に移動可能なエージェント眼部・茎部以外に、X軸方向にのみ移動可能な「見えないオブジェクト」(以下 NULL オブジェクト)も同時に移動させている。この NULL オブジェクトにはあらかじめ半径を設定し、同じく半径を設定しているエージェント胴体部との衝突を判定することを可能としている。具体的には、2つのオブジェクト間の距離を測定し、その距離が2つのオブジェクトの半径の合計より短ければ衝突していると判断する、というものである。

まずエージェント胴体部と NULL オブジェクトとの座標の差を取得し変数 `vecDistance` へと格納する。`vecDistance` へ格納される値はあくまで「X, Y, Z 座標それぞれの値の差」なので、オブジェクト間の直線距離を計算する必要がある。3次元座標空間における距離とはすなわちベクトルなので、`D3DXVec3Length` 関数を用いて `vecDistance` を3次元ベクトルへ変換し、変数 `fDistance` へと格納する。ここで前述の通り、「2つのオブジェクト間の距離 < 2つのオブジェクトの半径の合計」であれば2つのオブジェクト同士は衝突している。これを利用してユーザとエージェントの注視判定を行う。

人同士のコミュニケーションにおいて、互いの視線が一致したと判断するためには一定の時間が必要である。Kendon の研究[10]によると、14人のアイコンタクト持続時間の平均値が約1秒、10人の凝視持続時間の平均値が約2秒であったと報告されている。また深山らの研究[13]によると、この凝視持続時間はエージェントにおいても成立することが報告されている。よって本アプリケーションでも、互いの視線一致にかかる時間を1秒として設定した。

次に実際に「1秒間の注視」を判定する方法である。まずエージェント胴体部と NULL オブジェクトが衝突している間、フラグ変数 `ph_flag 1` の値を1ずつ加算する。本アプリケーションの動作は1/60秒ごとに停止するようにしているので、`ph_flag 1` の値が60になると1秒経過したと判断することができる。もし `ph_flag 1` の値が60になる前にエージェント胴体部と NULL オブジェクトが離れた場合、`ph_flag 1` の値を0へ戻し注視時間を初期化する。これによりユーザとエージェントの視線一致を判断する。これらの一連の処理を示したフローチャートを図7に示す。

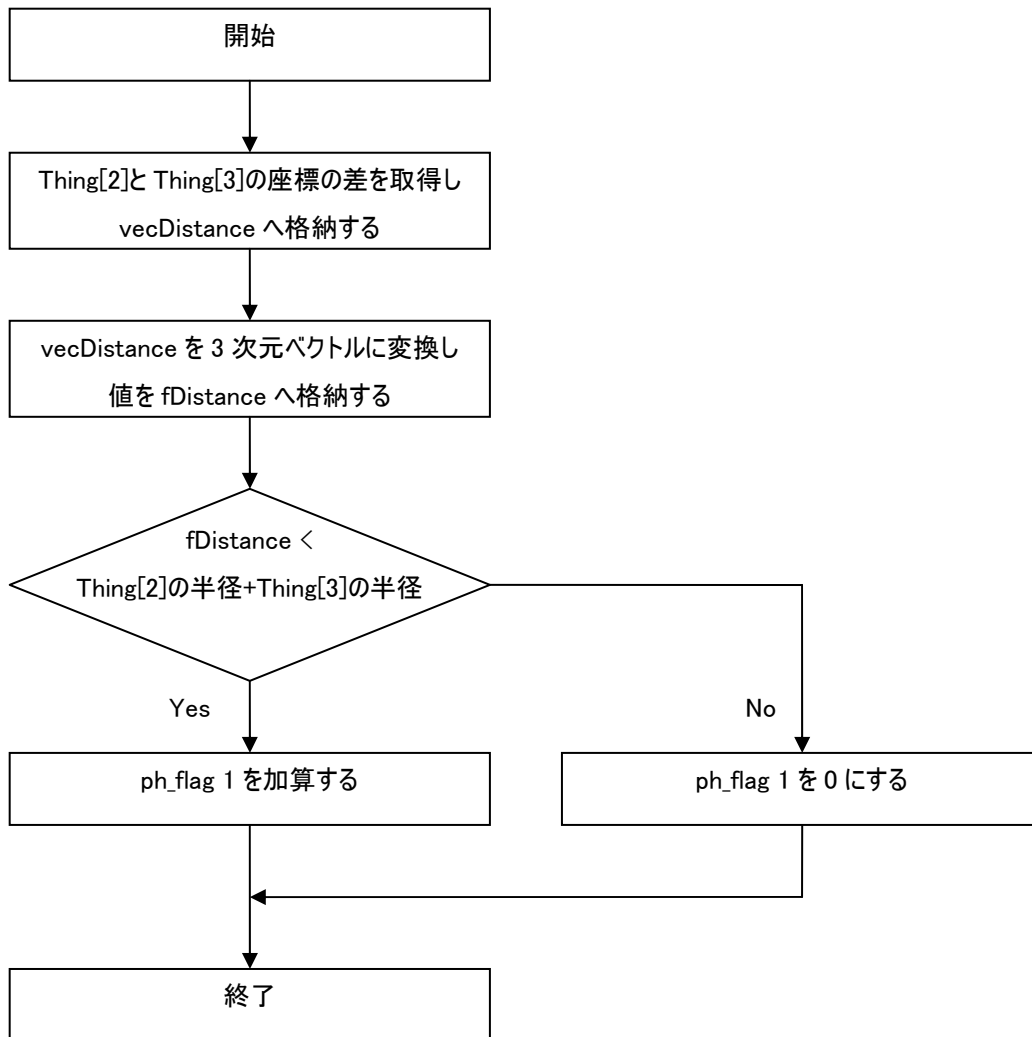


図 7 注視判定のフローチャート

4. 6 対話フェイズ

対話フェイズでは，ユーザに装着したヘッドセットマイクからの音声入力を元に，エージェントにうなずき動作をさせる．図 8 に対話フェイズの流れを示したフローチャートを示す．

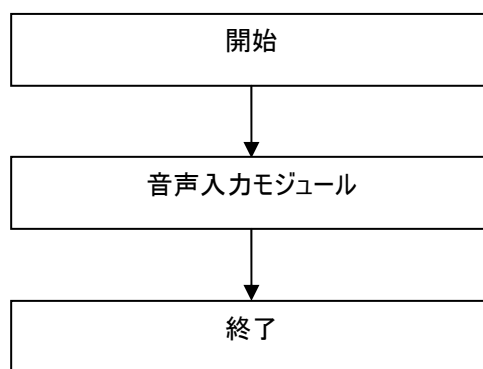


図 8 対話フェイズの流れを示したフローチャート

4. 6. 1 音声入力モジュールの設計

このモジュールでは，ユーザに装着したヘッドセットマイクからの音声入力を元に，エージェントの動作を決定する．うなずき動作はユーザの発話が終了し，次の発話が開始されるまでの間に生じる無音区間で行われる．この無音区間を検出するために，入力される音声を常に測定し，しきい値を用いて判別する必要がある．音声の入力は 1/22, 050 秒といったような非常に細かい周期（サンプリングレート）で行われる．そのためこの値全てを判別すると，突発的な大音量や無音にも反応してしまい，正確な動作を行うことが不可能になる．そこで，1 秒間の平均値を算出しその値を利用することで，突発的な音量の変化にも対応することを可能とした．同時にフラグ変数 **flag 3** を用いて発話終了のタイミングを捉えることにした．

まず入力された音声の値を格納する変数 **p** を用意する．次に 1 秒分の値を加算し平均を算出する必要がある．ここで「1 秒分の値」とはすなわち「サンプリングレート回分の値」である．このとき，より平均的な値を算出するために「その瞬間より前後 1/2 秒分ずつの値」を利用する．加算された値 **p** をサンプリングレートで割ることにより平均値を算出することができるが，音声の入力値は負の値を取ることもある．そこで **abs** 関数を用いて平均値の絶対値を算出し，これを判定に利用した．変数 **p** の値がしきい値未満のとき，変数 **flag 3** を 0 にする．この状態で変数 **p** の値がしきい値を超えらうなずき動作を行う．うなずき動作終了後，すぐに変数 **flag 3** を 1 にすることで，連続してうなずくなどの誤動作を防ぐ．これにより音声の入力，および入力に起因するエージェントのうなずき動作のタイミング検出を実装した．これらの一連の処理を示したフローチャートを図 9 に示す．

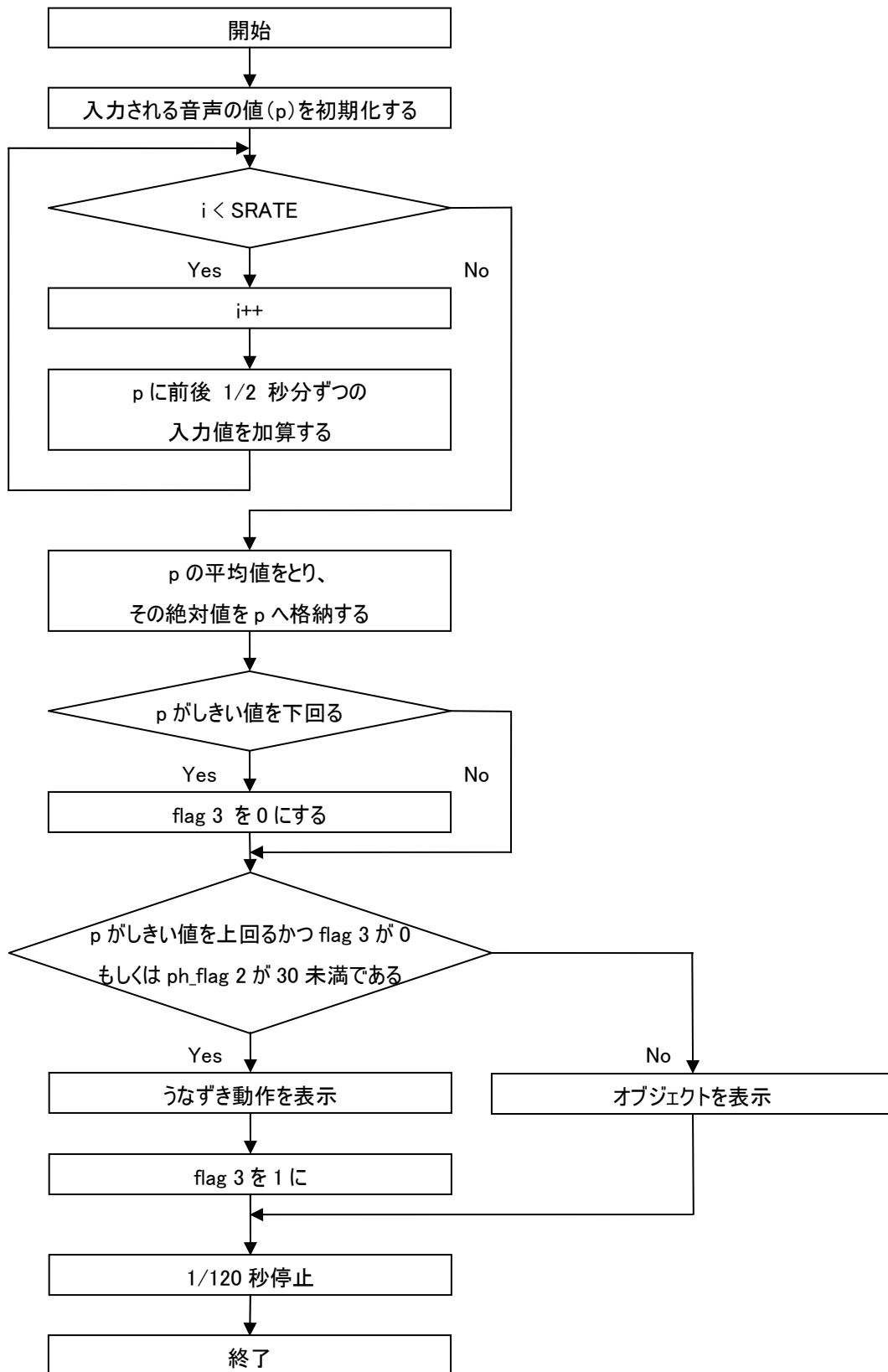


図 9 音声入力モジュールのフローチャート

4. 6. 2 うなずき動作

本アプリケーションのエージェントが行ううなずき動作は、オブジェクトの座標を細かく変化させることによって実現している。まずエージェントの眼部および茎部の X 軸に対する角度を変数 argX0 , argX1 に代入する。また眼部については、茎部が前傾することで Z 軸方向の位置が移動するため変数 posZ1 を用意し、Z 座標を格納する。次にフラグ変数 ph_flag 2 を用いて時間の測定を行う。対話フェイズでは 1/120 秒ごとにプログラムを停止させているので、 ph_flag 2 を 30 カウント、つまり 0. 25 秒でうなずき一回とした。 ph_flag 2 の値が 13 未満のとき、 argX0 , argX1 , posZ1 をそれぞれ 0. 015f ずつ減算する。 ph_flag 2 の値が 17 以上の場合は、 argX0 , argX1 , posZ1 をそれぞれ 0. 015f ずつ加算する。これらの値をエージェントの眼部および茎部の角度、座標へと代入しエージェントを表示する。この動作を 1/120 秒ごとに行うことで、うなずき動作を再現した。前述の通り、30 カウント (0. 25 秒) でうなずき 1 回となるので ph_flag 2 は 30 を超えると初期化される。これらの一連の処理を示したフローチャートを図 10 に示す。

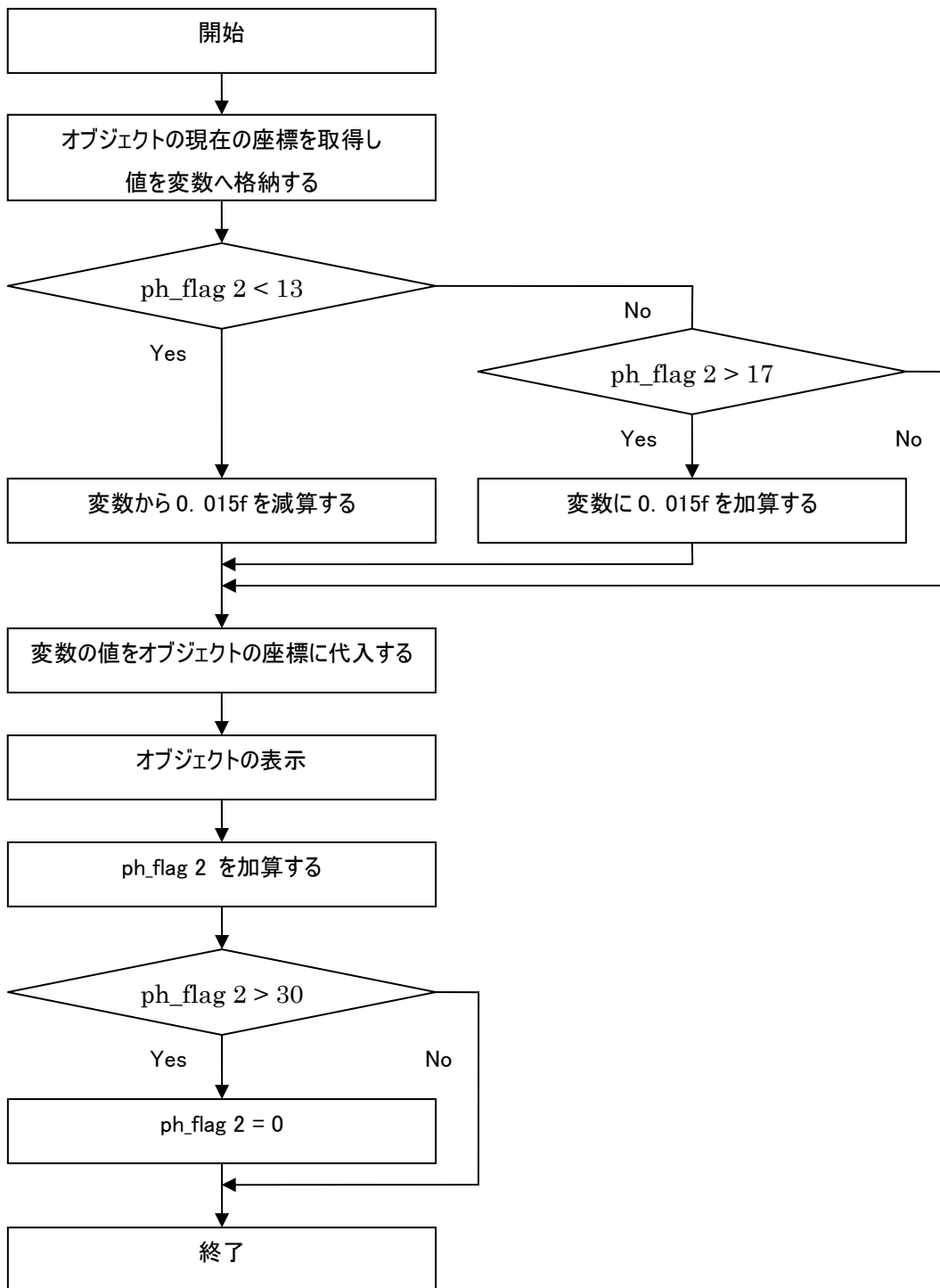


図 10 うなずき動作のフローチャート

5. インタラクション評価

本章では、インタラクション評価とその結果を示す。

5. 1 インタラクション評価方法

メタコミュニケーションを実装したエージェントの有用性を示すために、実際に本アプリケーションを用いてユーザとエージェント間でのインタラクションを行い、その評価を行った。このとき、今回実装した「ユーザ発話終了後の無音状態を検出しようなく」メタコミュニケーション動作が有効かを示すために、以下の異なる動作をする 3 種類のエージェントを用意した。

- ・ ユーザ発話終了後の無音状態を検出しようなく（正常なタイミング）
- ・ 正常なタイミング以外でうなずく（タイミングをずらす）
- ・ うなずきをしない

「正常なタイミング」では実装したアプリケーションそのままに、ユーザ発話終了後の無音状態を検出しようなく動作を行う。「タイミングをずらす」では、ユーザの発話に関係なく 5 秒おきにうなずき動作を行う。「うなずきをしない」では対話中にうなずき動作を一切行わない。上記 3 種類のエージェントを用いてユーザに順にインタラクションを行ってもらい、「人らしさ」及び「話を聞いているか」の 2 種類を 5 段階で評価した。いずれの場合も視線一致後の「会話の開始」の合図となるうなずきは行うものとし、その後の動作は各々の条件に従うものとした。被験者は大阪工業大学情報学部生 10 名を対象とし、実験終了後にはユーザヒアリングを行なった。実験風景は次の図 11 に示す。

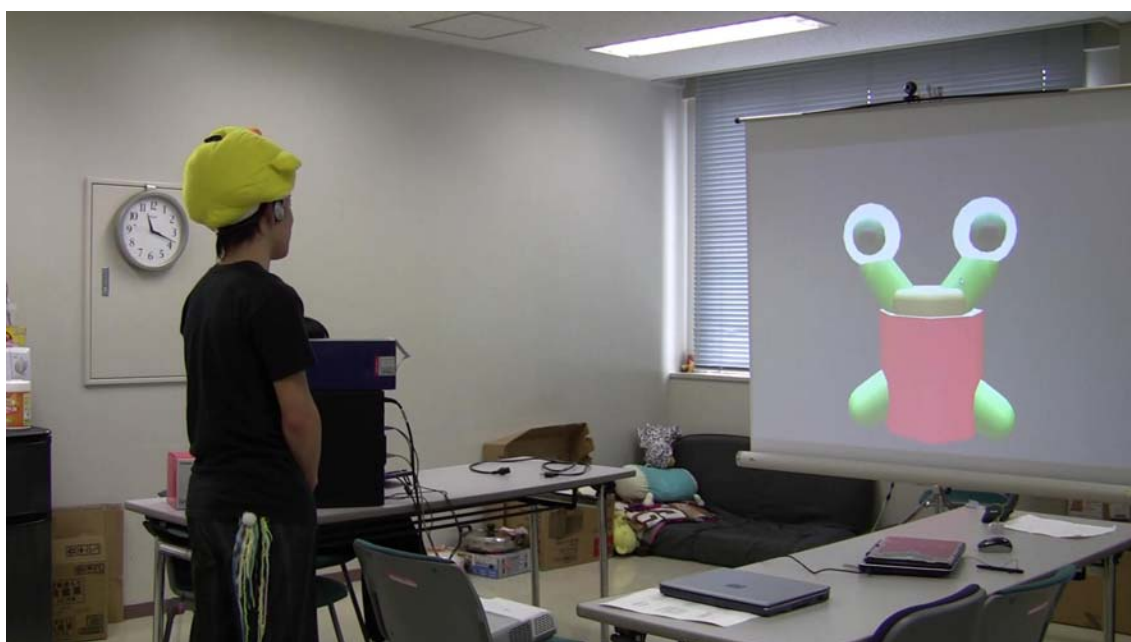


図 11 実験風景

5. 2 インタラクション評価結果

インタラクション評価の結果を表 2 に示す.

表 2 インタラクション評価の結果

	正常なタイミング	タイミングをずらす	うなずきをしない
人らしさ	3.80	3.20	1.30
話を聞いているか	4.30	2.90	1.30

表 2 より, 正常なタイミングでうなずきを行なった場合が最も「話を聞いている」と感じられることがわかった. しかし, 「人らしさ」に関しては「正常なタイミング」が「タイミングをずらす」場合よりもわずかに高いだけであった. この結果については次章にて考察する.

6. 考察

本章では, インタラクション評価で得られた結果から本エージェントに関する考察を行う.

6. 1 ユーザヒアリング結果

ユーザヒアリングにおいて, 被験者の一人より「適当にうなずいているほうが, 人間らしさがある」という意見が得られた. これは, 通常のタイミングでは対話を「一字一句逃さずに」聞いているように感じられるが, タイミングをずらしたものでは「話半分に聞いている」ように感じられ, そちらのほうが人間らしい, というものである. 人とエージェントとの対話研究においては, 発話促進のために威圧感を低減させながらも効果を失わない「あいまいさ」が有効であることが示されている[4]. 本研究においても, 「全て聞いているわけではないが, 全く聞いていないわけでもない」というこの「あいまいさ」がより「人間らしい」という印象を与えたのではないかと考えられる.

また, 別の被験者の意見として「うなずきよりも別の部分に視線がいく」というものがあつた. これは, 画面上に表示される本アバタにおいて最も面積を占めている胴体部が, 目立ちやすい赤色だったことに起因する. このためアバタの眼部よりも, まず胴体部に目が行く, というものであつた. この意見に関しては, アバタの色を変更することで回避することが可能だと考えられる.

6. 2 精度の向上

本アプリケーションにおける課題の一つとして, 精度の向上がある. 現状の CamSpace を用いたトラッキングシステムでは, ユーザに比較的大きめなマーカーを装着してもらう必要がある. また, 実際には「視線の追従」ではなく「マーカーの追従」のため, 本来の

意味での「視線の移動」を取得することが出来ない。そのため、「顔は前を向いているが、視線は横を見ている」といった認識をすることが出来ない。視線の追従に関しては、「アイトラッカー」[14]などの装置を用いれば非常に正確なトラッキングが可能となりマーカも不要となるので、ユーザの負担なくインタラクションを行なうことが可能となるだろう。

6. 3 うなずき以外のメタコミュニケーション動作

本研究では視線によるメタコミュニケーションに重点を置き、うなずきの動作を実装した。しかし、メタコミュニケーションにおける動作はうなずきだけではなく、身振りや手振りなど様々なものが存在する。これらの動作をエージェントに実装することで、より自然で円滑な、人間とのインタラクションが可能となるだろう。

7. おわりに

本研究では、メタコミュニケーションを考慮しユーザと対話をする目的でアプリケーションの開発を行った。開発したアプリケーションでは、ユーザの視線検出を行いユーザ位置にエージェントの視線を向ける「トラッキングフェイズ」と、ユーザの音声に合わせて「うなずき」を行う「対話フェイズ」の二つのフェイズを実装した。各フェイズにおけるエージェントのインタラクション動作に関してトラッキングフェイズでは、会話開始におけるユーザとエージェントの視線一致とうなずき、対話フェイズではコミュニケーション維持のための発話交替時のうなずき動作を行った。インタラクション評価により本エージェントはうなずきを入れることにより人らしいエージェントとなった。また、うなずきのタイミングをユーザの会話終了時に行うことで傾聴度の向上が見られた。今後は、ユーザヒアリングで得られたうなずきによる印象の違いやうなずき以外のメタコミュニケーション機能をエージェントに実装を検討することで、より自然で円滑な、人との双方向的なインタラクションが可能なエージェントとなるだろう。

謝辞

本研究を進めるにあたって適切な御指導をいただいた大阪工業大学情報科学部情報メディア学科神田智子准教授に心より感謝いたします。また、インタラクション評価や日頃から貴重なご意見を下さった大阪工業大学情報科学部情報メディア学科ヒューマンインタフェース研究室の関係者各位に心よりお礼を申し上げます。

参考文献

- [1] 山田誠二, 人とロボットの〈間〉をデザインする, 東京電機大学出版局, 2007, pp. 1-22
- [2] 黒川隆夫, ノンバーバルインタフェース, オーム社, 1994.
- [3] 中野有紀子, 第4章 エージェントによるしぐさと視線のコミュニケーション,
人とロボットの〈間〉をデザインする, 東京電機大学出版局, 2007, pp. 88-113.
- [4] 石井亮, 宮島俊光, 藤田欣也, アバタ音声チャットシステムにおける会話促進のための注視制御,
ヒューマンインタフェース学会誌, Vol. 10, No. 1, 2008, pp. 87-94.
- [5] 中野有紀子, ユーザの視線に気づく会話エージェント-アテンションの知覚と制御を利用した会話の
円滑化-, 第19回人工知能学会全国大会, 3B2-08, 2005.
- [6] 知野哲郎, 福井和広, 山口修, 鈴木薫, 田中克己,
“GAZE TO TALK”:メタコミュニケーション能力を持つ非言語メッセージ利用インタフェース,
情報処理学会シンポジウム「インタラクション98」予稿集, 1998, pp. 169-176.
- [7] Visual C++ デベロッパー センター, <http://msdn.microsoft.com/ja-jp/visualc/default.aspx>
- [8] DirectX デベロッパー センター, <http://msdn.microsoft.com/ja-jp/directx/default.aspx>
- [9] metaseq.net, <http://www.metaseq.net/>
- [10] Kendon A., “Some functions of gaze-direction in social interaction.”, *Acta Psychologica*26, 1967,
pp. 22-63.
- [11] CamSpace - Any Game, Any Webcam , <http://www.camspace.com/>
- [12] 山田誠二, 小野哲雄, 寺田和憲, 小松孝徳, 角所考, 人とロボットの〈間〉をデザインする,
東京電機大学出版局, 2007, pp. 23-68.
- [13] 深山篤, 大野健彦, 武川直樹, 澤木美奈子, 萩田紀博, 擬人化エージェントの視線による印象評価,
情報処理学会論文誌 Vol. 43, 2002, pp. 3596-3606.
- [14] Tobii, <http://www.tobii.co.jp/japan/home.aspx>

付録