

# Kinect for Windows を利用した顔認識技術の応用

金原 一歩 森安 昭太  
山成 大樹

## 1. 研究概要

研究目的は Kinect for Windows を用いて、顔認識技術及び機械学習を確立し、ユーザのコンピュータの使用をアシストするアプリケーションの開発をすることである。C#.net を主な開発言語とし、Kinect for Windows SDK と呼ばれる Kinect のソフトウェア開発キットを使用して、顔認識を行うモジュールと、顔認識を用いてコンピュータを操作するアプリケーションを開発した。

## 2. 研究の具体的内容

### (1) Kinect とは

Kinect とは、2010 年に Microsoft 社から発売された ゲームデバイスで、コントローラを使用することなく身体を使って直観的な操作ができる体感型のゲームシステムである。また、Kinect に搭載されているマルチアレイマイクロフォンにより音声による操作も可能である。Kinect には他にも RGB カメラや深度センサが搭載されている。発売当初は Xbox 360 専用であったが、後に Kinect for Windows と呼ばれる Windows 向けの Kinect が発売された。図 1 は Kinect 本体である。



図 1 Kinect 本体

### (2) 開発環境

#### a. 統合開発環境

- Microsoft Visual Studio 2010

#### b. SDK

- Kinect for Windows Ver1.7

#### c. 使用言語

- C#.net
- C/C++

### (3) 顔認識を行うアプリケーション

#### 【インタフェース】

インタフェース「Karen」は、各モジュールを統括し外部プログラムとデータのやり取りを行う。常時動作コマンドを受信しており、要求が発生した場合に適切な処理を各モジュールに行わせ、各モジュールから返ってきたデータを外部プログラムに渡す。適切な処理の例として、モジュール「Bianca」に対して認識している顔の口の形状 ID を要求するものである。図 2 に動作の手順を示す。また、モジュールとの関連図を図 3 に示す。

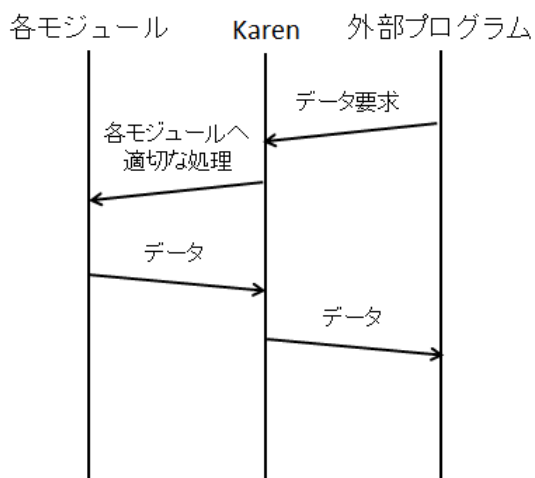


図 2 「Karen」の動作の手順

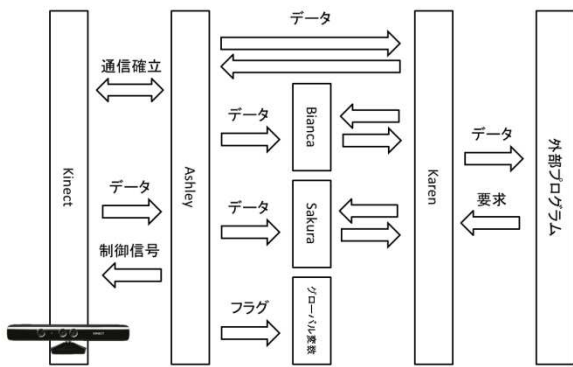


図3 関連図

【モジュール】

a. 「Ashley」

本モジュール「Ashley」は、Kinectの制御を行うものである。

Kinect for Windows SDK を利用し、Kinect との接続、通信、制御、RGB 画像データの取得、赤外線データの取得、骨格データの取得、顔認識などを行うものである。特に、本モジュールは顔認識を専門に処理を行っている。

Kinect から受信したデータを上位モジュール「Bianca」「Sakura」及びインタフェース「Karen」に渡すものである。また、全モジュール及びインタフェースがアクセスできる共有フラグレジスタを確保し、データを渡す際の同期をとっている。

(ア) Kinect との通信確立

本モジュールは、インタフェース「Karen」の起動後に Kinect との通信を確立する。Kinect との通信確立後に、インタフェース「Karen」へ通信確立を報告する。通信確立までの手順を図4に示す。

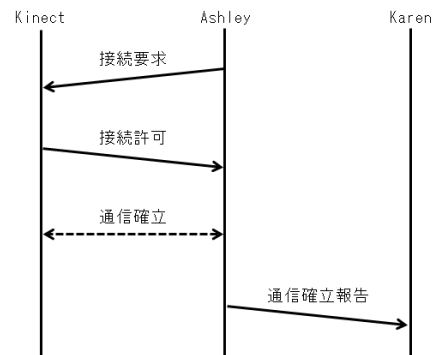


図4 Kinect との通信確立までの手順

(イ) 動作

Kinect との通信確立後、顔認識を常時行う。顔が認識され、Kinect から必要データ（顔の骨格データ、目の座標、口の座標、AU 値など）の受信が完了したら、上位モジュール「Bianca」「Sakura」及びインタフェース「Karen」へ受信完了を報告するため、起動時に確保したフラグレジスタにフラグ（True）を立てる。

上位モジュールまたはインタフェースからデータの要求が来ると、本モジュールはデータを受け渡す。受け渡すデータを表1に示す。

受け渡し先	データ
Bianca	Animation Unit 値
	口の位置座標
	口の開き距離
Sakura	目の位置座標
	RGB 画像データ
Karen	直接要求があったデータ

表1 「Ashley」の渡すデータ

(ウ) 動作画面

本モジュールは、RGB 画像データ及び認識に成功した顔をワイヤフレームで表示する機能を有する。また、表示・非表示は設定にて切り替えることができる。動作画面を図5に示し、認識された顔を図6に示す。



図5 「Ashley」の動作画面



図6 認識された顔

b. 「Bianca」

本モジュール「Bianca」は、下位モジュール「Ashley」により口の位置座標データを受け取り収集したデータにより機械学習を行い、口の状態を判断するものである。

(ア) 機械学習とは

機械学習とはコンピュータに人のような学習をさせるための技術の総称であり、今回はニューラルネットワークのフィードフォワードニューラルネットワーク多層パーセプトロン型を採用した。

(イ) ニューラルネットワークとは

脳神経系をモデルとした情報処理システムである。学習能力を持ち、教師信号の有無により「教師あり学習」と「教師なし学習」に分類される。ニューラルネットワークとは脳神経回路のように、神経細胞に相当する人工ニューロンが学習によってシナプスの結合強度を変化させ問題解決能力を持つようなモデル全般を指すものである。

フィードフォワードニューラルネットワークとは単純な構造の人工ニューラルネットワークモデルである。ネットワークにループする結合を持たないものである。

入力層、中間層、出力層の3種の層があり、入力層と出力層の2つの層で構成されているものを単純パーセプトロンといい入力層といくつかの中間層と出力層で構成されているものを多層パーセプトロンという。多層パーセプトロンは単純パーセプトロンと異なり、線形非分離の問題を解くことができる。多層パーセプトロンモデルを図7に示す。黒丸は人工ニューロンを示し、線は結合を示す。

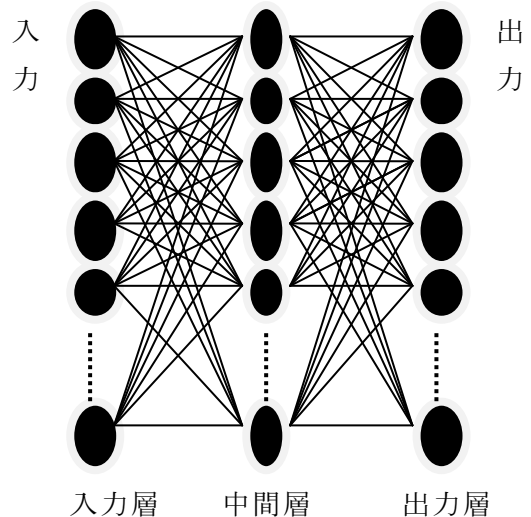


図7 多層パーセプトロンモデル

(ウ) 人工ニューロンとは

他のニューロンから入力された信号の合計値がしきい値  $\theta$  に達したときのみ他のニューロンに対して信号を出力する。

ニューロンの信号入力にはシナプス結合係数がありそれぞれの信号を取り込む割合が異なっている。例えば  $W1$  が 0.5、 $W2$  が 0.3、 $W3$  が 0.2 の場合は  $X1$  の値の重要度が 50%、 $X1$  の値の重要度が 30%、 $X1$  値の重要度が 20% として取り込まれる。

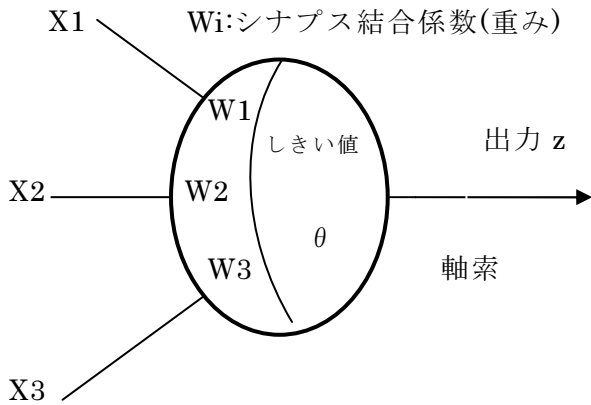


図8 ニューロンのモデル

Kinect の座標データ取得からの口の状態判断を行う前に動作確認のため、AND, OR, NOR, NAND の学習を行った。

例として AND: {000, 010, 100, 111} 4組のデータを持ち、一桁目 (X1) が教師データ (結果の答え) つまり二桁目 (X2)、三桁目 (X3) の2値のANDをとった値となっている。学習方法としてはランダムで結合係数を決め、出力された値が教師データとの誤差を計算し、誤差値が正なら結合係数を現在の値より小さい値にランダムに設定し、負ならば現在の値より大きい値にランダムに設定する。この動作を教師データと誤差が設定値以下に収束するまで繰り返す。この事からわかる通り誤差値0になるように学習するのはとても困難であるが、近似値が出るようなものを作ることはできる。

データ AND: { 000, 010, 100, 111 } , OR: {000, 011, 101, 111} , NOR: {001, 010, 100, 110} , NAND: {001, 011, 101, 110} (表 X 参照) をニューラルネットワークに入力した学習の様子 (誤差が減少し、収束していく様子) を図9に示す。

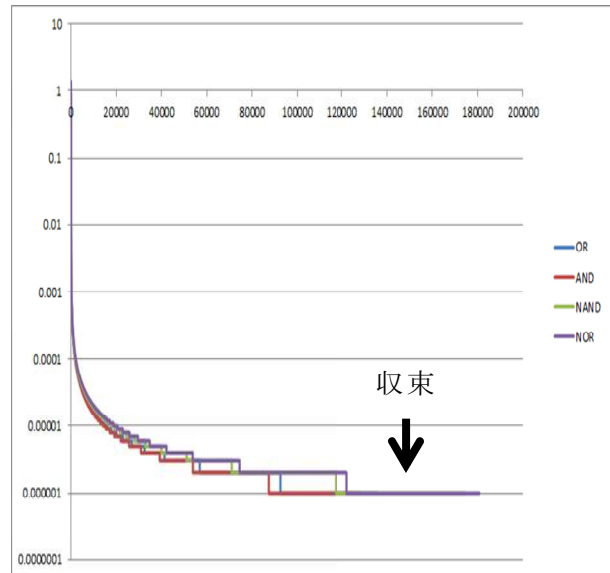


図9 学習の様子

(エ) 出力関数

出力関数に使う関数としてシグモイド関数を使用した。シグモイド関数を用いた理由としてはシナプスが発火 (電気信号の出力) するかの判断を数式で表す必要があったからである。脳機能のシナプスも一定以上の電気信号を受けたら電気信号を発信する。シグモイド関数では  $(-\infty, \infty) \rightarrow (0, 1)$  の単調増加関数で、一つの変曲点を持つ。Y=0, Y=1 を漸近線に持つ。また X=0 では変曲点は (0, 0.5) である。人工ニューロンも脳機能と同じく電気信号が出力されているか、されていないかの2つの状態がある。受け取った値 (電気信号) が一定以上なら出力 (電気信号出力) される。その一定以上の値を決めるための関数を出力関数という。図10に数式、図11にグラフを示す。このグラフ場合でいうと X軸が 0.5 以上なら Y軸はほぼ1になるつまり  $X > 0.5$  なら Y=約1となり脳機能でいうシナプスの電気信号の出力となる。

$$s_1(x) = \frac{1}{1 + e^{-x}}$$

図10 シグモイド関数

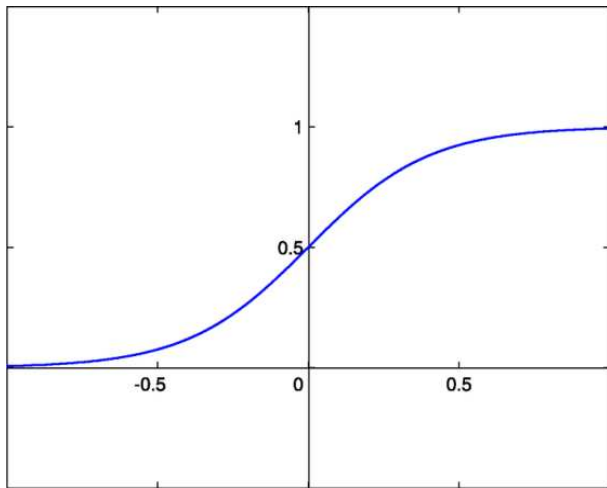


図 11 シグモイド関数グラフ

(オ) 分類器の説明

分類器とはあるものを決められたカテゴリーに分類するものである。例としてボールの分類を図 12 に示す。図 12 の分類器ではボールを分類するものであり、ボールの大きさ、形で野球ボール、バスケットボール、サッカーボールかどうか判断し、その 3 つかその他か分類する。



図 12 分類器

(カ) 口判断モジュールの説明

学習モードと判断モードが有り、下位モジュール「Ashley」からデータを受け取り、処理を行い判断モードではインタフェース「Karen」にデータを渡す。

(キ) 学習モード

学習モードでは下位モジュール「Ashley」から口の位置座標データを受け取り、人が口の状態のデータ（教師データ）を入力する。口の状態は「あ、い、う、お」の四種類とした。「え」は口の状態が「あ、い」と似ているため、音声がないと判断ができないので除いた。口の座標、状態データベースを作成しデータから機械学習を行い、分類器を作成する。上記にも記述した通り音声認識でないと困難な部分があるので現在、声と同時に判断し分類する機構を開発中である。表 2 は、口の位置座標データによって分類された写真である。

口	写真	データ
あ		AU0 : 0.9764161 AU1 : 0.5239825 AU2 : 0.05262316 AU3 : 0.4365219 AU4 : -0.01462844 AU5 : -0.03859904
い		AU0 : 0.7583919 AU1 : 0.07828864 AU2 : 0.4560076 AU3 : 0.2878533 AU4 : -0.3452756 AU5 : -0.124165
う		AU0 : 0.6596825 AU1 : 0.06069297 AU2 : -0.5156239 AU3 : 0.2339084 AU4 : -0.1874401 AU5 : -0.08095394


お		AU0 : 0.2333735
		AU1 : 0.3262616
		AU2 : -0.5369856
		AU3 : 0.3746871
		AU4 : 0.3536172
		AU5 : 0.1322058

表 2 分類された画像

(ク) 判断モード

判断モードでは下位モジュール「Ashley」から口座標データのみを受け取り、学習モードで作成した分類器により口の状態を判断する。

現在データ数が少ないため精度が低い。今後データの充実が課題である。

c. 「Sakura」

本モジュール「Sakura」は、まばたきの検出を行うものである。下位モジュール「Ashley」からのデータを受け取り、まばたきの検出を行う。

(ア) 受け取るデータ

下位モジュール「Ashley」から送られてきた2つのデータの内容を説明する。この内容に含まれている画像は、実際の認識に使われたものである。

目データには、各目の中に右端 (R) ・左端 (L) ・上端 (T) ・下端 (B) の位置座標が含まれている。図 13 は、座標の位置を可視化したものである。

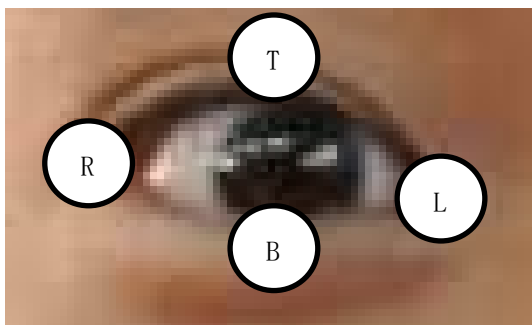


図 13 目データ

RGB 画像データは、縦 480 ピクセル、横 640 ピクセルのビットマップである。図 14 は、受信したデータである。



図 14 RGB 画像データ

(イ) 動作

本モジュールは、下位モジュール「Ashley」から RGB 画像データ及び目データを受け取り、動作を開始する。まばたきの検出は、下記の手順で行う。

1. RGB 画像データから目周辺の皮膚の色の平均値を計算する。図 15 参照。



図 15 抽出を行う部分

2. 目データより目の右端 (R) から左端 (L) まで直線を引く。図 16 参照。

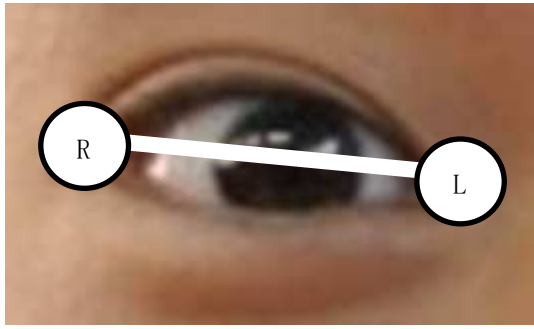


図 16 直線を引いた様子

3. 直線上のどの色の値も先ほど計算した皮膚の色でなければ「目は開いている(True)」、でなければ「目は閉じている(False)」という結果を得る。図 17 に判断された目を示す。



図 17 判断された目

4. 1～3 の動作を両目に行う。
5. インタフェース「Karen」にまばたきのデータを渡す。

#### (4) 顔認識を用いたアプリケーション

開発した顔認識を行うモジュールを用いて、以下の顔認識を利用したアプリケーションの開発を行った。

#### 【アプリケーション】

##### a. 「MMDFaceTracking」

本アプリケーションは、3DCG モデルの表情を、取得したデータをもとに変更するものである。いわゆるアバターシステムというものである。

動作に利用したデータは、「Ashley」より骨格データを取得し顔の傾きを得ている。「Bianca」より口 ID を取得し 3DCG モデルに

その口を適用させている。「Sakura」よりまばたき検出結果を取得し、3DCG モデルにまばたきを適用させている。

まばたきに関して、顔の移動と同時に行うと全く認識できなかった。これを改善したい。

動作の例として、首を傾けると同じ方向に 3DCG モデルが動き、口を開けると同じように口が開くなどである。動作の様子を図 18 に示す。

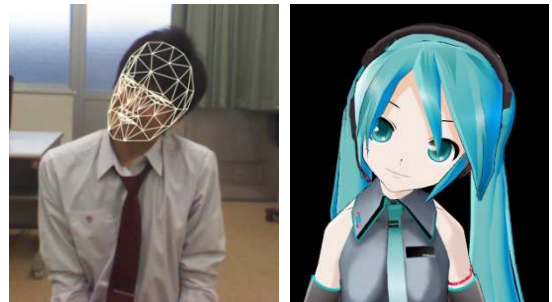


図 18 動作の様子

##### b. 「助けて小鳥さん」

本プログラムは、Kinect に困った顔をすればアクティブなソフトウェアのヘルプ画面及びミニ検索バーが表示されるものである。

動作に利用したデータは、「Bianca」の機械学習の内容を変更し、口ではなく表情の認識を行わせた。機械学習にはたくさんの困った顔の写真が必要であるが、あまり集められなかったため精度が低い。

動作の様子を図 19 に示す。



図 19 困った顔を認識している様子

### 3. 研究のまとめ

今回の課題研究では、顔認識を行うモジュールをメインに開発してきた。開発はスケジュール通りに行かなかったが、顔認識を行うモジュール「Karen」並びに顔認識を用いたアプリケーションの開発に成功した。顔認識を行うモジュール「Karen」は、「Ashley」「Bianca」「Sakura」の3つのモジュールからなり、3人で1つずつモジュールを開発した。

「Bianca」の口判断において「あ、い、え」の同時の認識は難しいという発見があった。「あ、い」の分類はできるが、ここに「え」が入ると境界がなくなり全く分類できないという状態に陥った。このため「え」の認識を諦め、「あ、い」の分類だけとなった。「え」を分類するために、音声認識を用いて音声と同時に判断し分類する機構を開発中である。

顔認識を用いたアプリケーションとして、Kinect に困った顔をすればヘルプ画面が表示されるものを開発した。顔認識を行うモジュール「Karen」を用いたため、アプリケーションの開発が容易であった。モジュール「Bianca」の口判断のデータベースを困った顔のデータベースに差し替えることによって認識を行わせた。

グループの開発は非常に難航し、モジュールの結合テストで、与える側と受け取る側の変数の型が違いエラーに悩まされていた。グループでの開発には、きちんとした仕様、計画そしてお互いのコミュニケーションが非常に重要であった。

### 4. 今後の発展

「Bianca」の今後の発展としては、現在起動時に必ず初めにデータベースからデータを読み込み、学習モードを行う必要があるこれでは起動毎に時間がかかってしまい不便である。学習結果も外部データベースに保存し起動時に読み込む様に変更したいと思う。また判断モード中もバックグラウンドで学習モードを動かしデータベースが更新されたらリアルタイムで判断モードの分類器を更新する処理を行いたい。今後この分野の発展として現在は口の状態の判断だけだが、顔全体の状態や人の行動を観測して実際にそこにいるようなバーチャルヒューマンの開発など、人と実際に対話しそれに応じた行動をとる、現在でもそのような技術はあるがまだまだ人とプログラムやロボットの壁は大きい、もっと人間らしい、機械学習によって自分で学ぶ人工知能などが作りたいと思う。

「Sakura」の今後の発展としては、まばたき検出に加えて黒目の検出を搭載したいと思う。この報告書には書いていないが、「Sakura」は黒目の検出が実験段階ながら可能である。検出率が高くなく誤検出も多かったので記述しなかった。黒目の検出はまばたきの検出の応用であり、RGB 画像データから白黒白と並んでいる部分を検出している。しかし、RGB 画像データだけでは限界があり、撮影場所が暗いなどの影響により検出できなかつたりする。これを改善するためには赤外線カメラデータを組み合わせて検出率を上げ、黒目の検出機能を追加したい。黒目の検出を用いることで、医療分野において身体の麻痺した人でも目を動かし、右なら「はい」、左なら「いいえ」など単純な応答が可能となると思われる。



## 5. 感想

金原 一步

今回自分の興味のある分野、機械学習、人工知能を Kinect に実装することができた。(今後、顔認識の技術が進み、ネットワークでは感情が伝えにくく、トラブルも起きているのが現代社会である。感情を伝えるというのはコミュニケーションの上とても大切なことである。情報技術がめまぐるしい速度で発展している、ネットワーク上でコミュニケーションをとる場合も増えてくる。その際感情も伝えたいと思った。表情というのは、感情に左右され変化するものである所以このアプリケーションを開発しようと思った。) 参考書籍などによりプログラムを実装することはできたが、理論的なことはすべては理解できなかった。今後も情報技術に関わるものとして理解し使えるようになっていきたい。また、グループでの制作の難しさも知れた、特にグループの制作ではコミュニケーションが大切だと思った。

森安 昭太

課題研究では「Ashley」「Sakura」「MMDFaceTracking」を開発した。参考書籍やサンプルプログラムをもとにプログラムを作り上げることができた。「Ashley」の開発当初は、Kinect のサンプルプログラムが全く分からず難航していたが、サンプルプログラムを自分なりに改造し、参考書籍をくまなく読みつくし Kinect を自分の動かしたいように動かせるようになった。「Ashley」は、Kinect を容易に扱えるライブラリ的な機能を有している。完全なライブラリの形式ではないが、いつかライブラリを作り配布したいと考えている。まだまだ実装したい機能があるので今後も実装していきたい。今後もこの研究で学んだことを活かしていきたいと思う。

山成 大樹

今回の課題研究では「音声認識」について学んだ。本来は「黒目の動きを認識し、眼球を動かす」というプログラムを作成する予定だったが、参考になりそうなサイトを検索してみたところ、原理のみはわかったが数千枚の画像データが必要なことと、自分の知識では完成はかなわないだろうということ、そして参考ソースコードが1ミリも落ちていなかったことから断念せざるを得なかった。このように自分は正直0からのスタートで、kinect の事はおろか、C#さえもよくわかっていなかった。しかし音声認識について調べているうちにそこそこはプログラムがわかるようになっていた。

この研究をうまく生かせられれば「電話での顧客の要望に対して受付嬢を使うこと無く、音声で認識し、自動返答するプログラム」や「声を発することができない人でも、口の動きを認識して料理などを自動注文してくれるプログラム」等が作れそうで、とてもわくわくしている。ただ、案はあっても技術がないので、現段階では自分の力だけでは不可能である。今回の研究ではサンプルプログラムや参考サイトのおかげでなんとか完成までもちこめたが、これからは自分で1から組んでみたい。



## 6. 参考文献

・小高知宏(2011) 『はじめての機械学習』  
オーム社 208pp.

・中村薫、宮城英人、田中和希(2012)  
『KINECT for Windows SDK プログラミング C#  
編』 秀和システム 325pp.

・中村薫(2011) 『KINECT センサープログラ  
ミング』 秀和システム 351pp.

・谷尻豊寿(2011) 『KINECT センサー画像  
処理プログラミング』 カットシステム  
287pp.

・山本康彦(2013) 『C#によるマルチコアの  
ための非同期/並列処理プログラミング』 技  
術評論社 256pp.

・井上雅裕、上田洋平、小野弘貴、片山陽  
亮(2013) 「KINECT for Windows を使用した  
アプリケーションの制作」 『課題研究 課題  
研究のまとめ 第26号』 pp.5-8. 岡山県立岡  
山工業高等学校 情報技術科

・「Face Tracking」

<[http://msdn.microsoft.com/en-us/library  
/jj130970](http://msdn.microsoft.com/en-us/library/jj130970)> (2013/07/05 アクセス)

・「DX ライブラリ置き場」

<[http://homepage2.nifty.com/natupaji/DxL  
ib/](http://homepage2.nifty.com/natupaji/DxLib/)> (2013/07/05 アクセス)

・「Kinect for Windows で機器を作るヒン  
ト(4) — 日本語の音声認識をやってみる」  
<[http://www.kumikomi.net/archives/2012/0  
6/ep08knc4.php](http://www.kumikomi.net/archives/2012/06/ep08knc4.php)> (2013/08/01 アクセス)

・「MMD 基準モデルフォーマット、PMD/PMX  
について」

<[http://mikudan.blog120.fc2.com/blog-ent  
ry-280.html](http://mikudan.blog120.fc2.com/blog-entry-280.html)> (2013/11/18 アクセス)

## 7. ライセンス

「MMDFaceTracking」には、「初音ミク」を  
利用しております。

「初音ミク」はクリプトン・フューチャー・  
メディア株式会社の著作物です。

© Crypton Future Media, INC.

[www.piapro.net](http://www.piapro.net)

使用モデル「Lat 式ミク Ver2.3」