発話時の顔画像を用いた発声単語認識システムの構築

中村, 亮太 / NAKAMURA, Ryota

(発行年 / Year)
2011-03-24
(学位授与年月日 / Date of Granted)
2011-03-24
(学位名 / Degree Name)
修士(工学)
(学位授与機関 / Degree Grantor)
法政大学 (Hosei University)

2010 年度　修士論文

発話時の顔画像を用いた
発声単語認識システムの構築

指導教授　　赤松　茂

法政大学大学院
工学研究科システム工学専攻
09R6116
中村　亮太

アブストラクト

本論文では,発話時の顔の動画像から抽出される口周辺の動きの情報を用いることによって,音情報によらずに発声単語の自動識別を行うシステムについて述べる.時系列画像の各画素における動きを Optical Flow の算出によって求めた.それぞれの発話単語の発話の開始と終わりの位置を決定し,発話区間を抽出し,発話ピークフレームをその区間の中から取得する.また,部分空間法を用いて,辞書の作成を行い,類似度を計算することで識別を行った.発話単語認識に関する基本的な性能評価として,「ありがとう」「こんにちは」を含む発話 7 単語の識別実験を行った.

キーワード　　発声単語,読唇,オプティカルフロー,部分空間

Abstract

This paper describes a vision-based spoken word recognition system that utilizes, instead of audio signal, visual motion signal which is obtained from motion pictures during speech. Motion information on each pixel in the input time-series imagery was obtained by computation of optical flow. The utterance interval was extracted by defining both starting and ending points of time for each spoken word, and some peek frames in the utterance interval were obtained. Subspace method (CLAFIC: CLAss-Featuring Information Compression method) was applied to register in the classification dictionary and to classify each spoken words. As a preliminary performance evaluation of the proposed feature in spoken word recognition, discrimination test of seven spoken words including A-RI-GA-TO-U and KO-N-NI-CHI-WA was conducted.

Key Words: spoken words, lip-reading, optical flow, subspace method

目次

図・表目次

第1章　はじめに

1.1　福祉のためのインタフェース

今日，日本では高齢化が世界に類を見ないスピードで進行している[1]．一般的に高齢化に伴い身体の不自由な方が増加していく．その反面，技術の進歩により，システムの仕様がより複雑になってきているため，誰でも使いやすくより負担の少ないインタフェースの導入が必要になってきている．福祉に対する工学的なアプローチ方法も数多く研究されており[2][3]，インタフェースの実用化もされつつある．そこで，人間は視覚によって顔から様々な意味を抽出することができること[4]を利用し，日常動作である「発話」に注目してみる．発話によるインタフェースは環境にロバストであり，難しい動作を含まないため負担が少ない．発話の認識を行うのに一番有効な手段は音声認識であるが，音声認識では雑音に弱いため，唇の動き情報が必要になる．また，人は音声のみで相手の言葉を理解するのではなく，音声と唇の動きの影響も受けるということも言われている．これをマガーク効果という[5]．図 1 に唇と音声認識の効果について示す.

視覚処理を用いた発話認識の先行研究には単語の各母音に対応する口の輪郭に注目したもの，口の周辺の動きや見え方に注目したものの 2 つのアプローチ方法が挙げられる．前者の先行研究では, Avtive Appearance Model を用いて口の形の特徴を取得している研究[6][7]も挙げられている．しかしながら，これらのアプローチでは個人の各母音における口の形の違いによる影響を受けてしまうこともある．また，日本語を発話する際は口の開閉動作が大きくなくても発話することができてしまう.

後者の先行研究では，口の位置を色抽出法と Eigentemplate 法を用いて検出し，口の形の時間的な変化を固有空間法で記述し，認識を行っているアピアランスベースのもの[8]や，取得した顔画像を制約相互部分空間上で展開し，ジェスチャー認識を行っているものもある[9]．また，発話された区間の口の開き具合と口の動く速さを表す 2 つの差分系列を取得後，これらを 2 次元グラフ化し，検出線とグラフの交差回数を特徴として認識を行っているもの[10]や口の上下左右の領域の動きを Optical Flow より算出し主成分分析を適用し，口の開閉・伸縮の特徴を求め，それらを用いてマッチングを行い良好な結果が得られた研究も挙げられている[11].

口の動きではなく，顔の表情変化の認識でも，Optical Flow を用いて抽出された特徴を用いて良好な結果が得られている[12]． 表情の先行研究[13]では頬や口の部分に表情が表出されやすいと言われている．中村らは，発話に伴う動きにも有効であるかを確認するために，口周辺の部分，頬部分，両者を結合した部分それぞれに注目して識別性能の比較を行い，発話認識には．口周辺の領域に注目した方が良いという結果を示した[14].

本論文では，表情認識の先行研究[12]に動機づけられ，後者のアプローチ方法を採用し，発話中の口の動きの変化に注目した音声情報を用いない発声単語認識を目指す．また，複

数の被験者からサンプルを取得し，不特定話者に対応した辞書の作成を行い，識別する．

顔の動き特徴を用いる先行研究の多くは，顔の位置が変化しないと仮定しているが，実際の状況では頭の動きは避けられない．そこで，本研究では，株式会社 ATR-Promotions が提供している顔のリアルタイム検出・追跡ソフト accFace^{TN} SDK 版[15](以後，accFace と略す)を用いて，発声画像より両目位置を取得し，顔の位置・傾き・大きさの自動正規化を行った．

また， 発話した動きによる速さを用いて，発話のピークを求め，ピーク時の画像の輝度値を用いて各クラスの部分空間を作成したものを用いて識別を行い，口の周辺の動きや見え方の特徴と手法の違いによる識別性能の比較を行った．

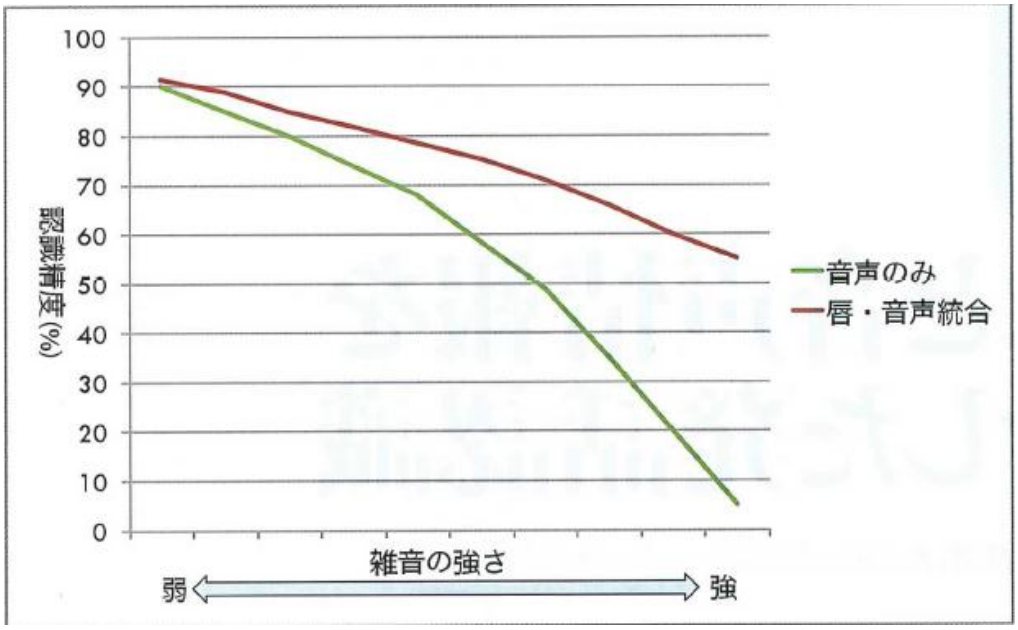


図 1　唇・音声統合認識の効果[5]

1.2　研究目的

本論文では，発声画像から読み取れる情報の中で，口周辺の動きに注目し，音によらない発声単語の判別を行う．

今回識別する単語は「ありがとう」「いいえ」「おめでとう」「こんにちは」「さようなら」「すみません」「はい」といった，誰にでも馴染みがあり，難しい動作の少ない挨拶の中から選択した．画像からでは母音と子音の違いは判断できないため，ある音からある音への一連の動作の流れを特徴として取得することを目指した．

そこで，Optical Flow 特徴を用いて辞書を作成したものと発話ピーク時の顔画像をベクトルにしたもので辞書の作成を行ったものを作成し，比較を行う．

第2章　理論

2.1　発声単語認識までの流れ

図 2 にコンピュータによる認識の流れを示す.
これはパターン認識の研究では一般的な流れである．パターン認識の研究をする上で一番考慮されるのが特徴抽出の部分である.
以下，本章では，今回研究に用いた理論部分の説明を行う.

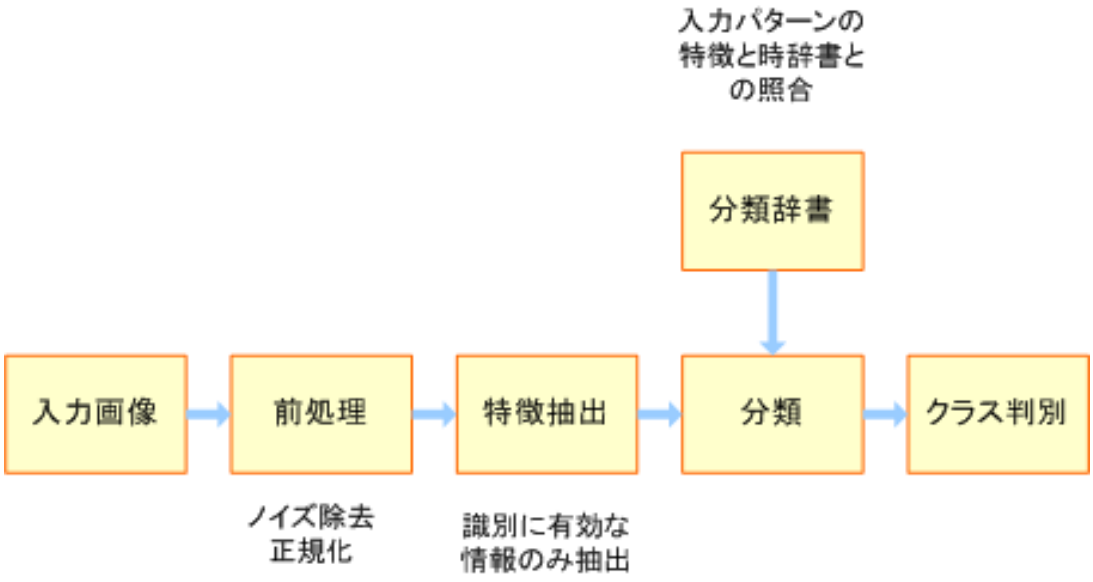


図 2　コンピュータによる単語認識の流れ

2.2　入力画像について

今回は入力画像に動画像を用いた．動画像は静止画の集まりであるため，その FPS さえ設定すれば, 簡単に静止画を抽出することができる. FPS とは, Frame Per Second のことで, 1 秒間に何枚の画像を表示するかを示すものである．FPS が高ければ，より滑らかな動画を撮影することできる．より滑らかな動きを取得するような研究用途，例えばジェスチャー認識には 60FPS，そこまで滑らか動きを必要としない追跡や顔認識には 30FPS を使用することが多いように感じられる. これは, 後に出てくる特徴ベクトルとの兼ね合いもある.

どのような特徴をどんな手法で識別するのかをしっかり考えてから，入力画像の詳細を決める必要がある.

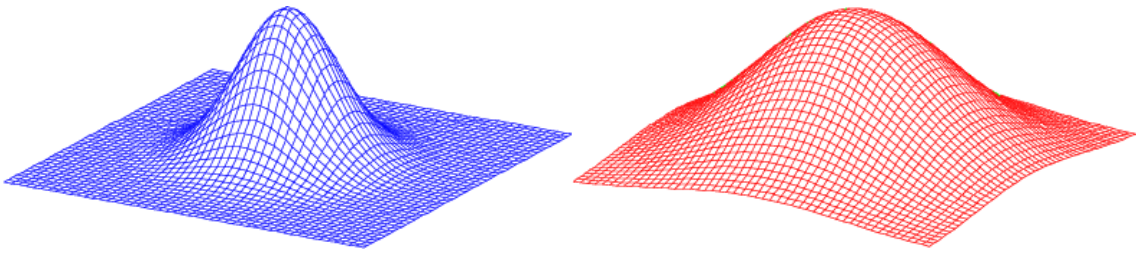
2.3　前処理

取得した画像には，Web カメラのノイズ，顔全体の傾きや，照明，顔の位置による差が生まれてしまう．
前半では，Web カメラのノイズや照明のノイズ除去について記述する．後半では，顔全体の傾きや位置，大きさの差を整える作業について記述する．

2.3.1 フィルタ処理

今回は Web カメラのノイズや照明のノイズ除去にモノクロ化とガウシアンフィルタを施すことにした．
モノクロ化は画像の各画素の R，G，B を足して，3 で割った値を計算した．

ガウシアンフィルタとは画像のノイズ除去を目的とした平滑化フィルタである．ガウシアンフィルタは画素の空間的配置を考慮して，対象画素に近い画素に大きな重みを，対象画素から遠い画素には小さい重みを付けた加重平均を取る．この重み付けにガウス関数を採用している[16]．図 3 に 2 次元のガウス関数を示す．標準偏差$\sqrt{\sigma}$ の値によって平滑化の度合を変化させることができる．



(a) $\sigma = 1$　　(b) $\sigma = 2$

図 3　2 次元のガウス関数　$f(x,y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}}$

2.3.2 顔画像の正規化

顔画像の正規化では，顔の位置，大きさ，平面上の回転について，同じ条件にする処理をかける．
本研究では顔が横を向いていたり上を向いていたりという傾きはないと仮定を立てて，サンプルの取得を行っている．もし，横顔から正面画像へ変換したいという場合は，複数の画像や極座標変換などを用いて 3 次元座標へ拡張する方法が研究されているので，そちらを参考にしていただきたい．

顔の位置，大きさ，平面上の回転の正規化には最低でも 2 点あれば可能である．そこで，今回は顔のリアルタイム検出･追跡ソフト accFace 用いて，顔画像より両目位置を取得した．accFace で使われている理論は参考文献[17]に詳しく書いてあるので，割愛する．

取得された両目の座標を基に，平面上の回転角度を算出し，その角度の分だけ反対方向に回転させれば，両目が水平になる．
次に，回転行列をかけて求めた両目座標の距離を算出し，基準となる値との倍率を計算する．倍率がでたら，画像の大きさをその数値で拡大縮小処理を行う．
更に，両目座標の中心座標を眉間座標として，そこから指定した顔の領域を切り出すと，正規化された時系列の画像が抽出できる．

尚，この手法だと顔の長さに少し誤差がでてしまう．
唇の特徴点を Active Appearance Model を用いて抽出する方法[6][7]もあるが，事前に学習が必要なため，今回は導入しなかった．

2.4　特徴抽出

顔画像からの特徴抽出法は，主に２つに分類され，静止画像を空間的特徴で表す方法と，時系列画像に見られる顔面の時間的変化を特徴として表す方法がある．
赤松研究室では，静止画像を空間的特徴で表す方法として多々あるが，中でも Gabor Jet 特徴ベクトルを用いたものなどがある，これは Gabor フィルタが局所的領域の構造的な特徴を抽出できるためである．

本研究では，空間的特徴ではなく，顔面の時間的変化に着目する．
人が発話した時には口腔，鼻腔，口唇などで構成されている声道を，筋肉を動かして変形させる．口唇と顎を動かす筋肉によって顔の表情にも変化をきたし，その変化をもとに話している言葉を理解する方法が読唇術である．もし筋肉の時間的な動きをとらえることができれば，そのとき発声された単語を識別できるはずである．また，生理学的にみれば話者が変わっても同じように筋肉を動かすと考えられるため，個人の特徴に影響を受けることがない．
また，顔の見え方(アピアランス)も顔認識やジェスチャー認識に有効だとされているため，本論文では動き特徴ベースとアピアランスベースのどちらも特徴を構成し，識別を行い，比較している．
そこで，顔の動きを取得するため，唇の動き特徴を Optical Flow を用いて抽出，唇の見え方(アピアランス)を画像の輝度値から抽出する．

以下に，OpticalFLow の原理と輝度値特徴の説明を行う．
また，それぞれの特徴を用いた場合の学習方法と識別方法を説明する．

2.4.1 Optical Flow を用いた手法

2.4.1.1 Optical Flow とは

Optical Flow とは画像内の明るさのパターンの動きの見かけの速さの分布のことである．画素の値は物体の動きによって変化するため，速度に関する情報を得ることができる．Optical Flow を計算する方法は勾配法とブロックマッチング法のアルゴリズムがあるが，今回は一般的に使われている勾配法を用いて，Optical Flow を計算することとした．

以下では勾配法について示す．

画像上の (x,y) の時刻 t での濃淡値を $f(x,y,t)$ とし，その点が時刻 $t+dt$ に $(x+dt,y+dt)$ に移り，明るさが変化しないとすると，次式が成り立つ．

$$f(x,y,t)=f(x+dx,y+dy,t+dt) \qquad (1)$$

上式をテーラー展開し，dx,dy,dt の二次以上の項は微小であるとして切り捨てる．両辺を dt で割り，

$$f_x(x,y,t)u+f_y(x,y,t)v+f_t(x,y,t)=0 \qquad (2)$$

ここで u は x 方向の速度成分，v は y 方向の速度成分，添字は偏微分を表す．これが，勾配法における拘束式である．上式は，2 つの未知数 u,v を含むので，このままでは，解くことができない．そこで，次のような方法を用いて u,v を求める．

Lucas-Kanade 法[18]

Lucas-Kanade 法とは同一物体の局所領域内ではオプティカルフローは一手になると仮定する空間的局所最適化法の一つである．また，Lucas-Kanade 法は高速で精密な Optical Flow を推定することができる．

同一物体の濃淡パターン上の局所領域において，得られる Optical Flow の拘束方程式((2)式)は同一の解とする．いま局所領域 w における Optical Flow (u,v) は

$$u=\frac{\sum_w \frac{\partial f}{\partial x}\cdot[J(p)-f(p)]}{\sum_w\left(\frac{\partial f}{\partial x}\right)^2} \qquad (3)$$

$$v = \frac{\sum_w \frac{\partial f}{\partial y} \cdot [J(p) - f(p)]}{\sum_w \left(\frac{\partial f}{\partial y} \right)^2} \qquad (4)$$

さらに,

$$f(p) = f(x, y, t) \qquad (5)$$

$$J(p) = f(x, y, t, dt) \qquad (6)$$

とあらわされる.

Optical Flow の例を図 4 に示す.
また, Optical Flow には Lucas-Knade 法以外にブロックマッチング法で求める方法もあるが, 計算コストが高いというデメリットがある.

この原理で求めた Optical Flow は，X 方向，Y 方向それぞれについての速度ベクトルをフレーム数分算出される.

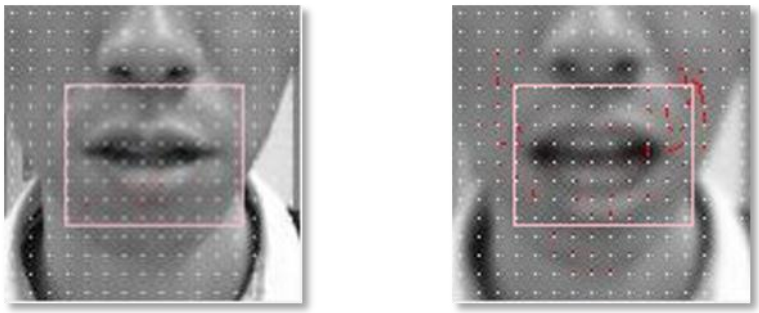


図 4　Lucas-Kanade 法を用いた Optical Flow

2.4.1.2　発話区間

取得した動画は発話していない時間も含まれるため，発話とは関係ない時間も混在している．そこで，全時系列画像から Optical Flow で得られたフロー成分を使い，発話している時間のみのフロー成分を取得する．
発話区間を設定するためには発話開始フレームと発話終了フレームを求めればよいはずである．以下のような定義を行う．

まず，各フレームについて特徴点における X,Y 方向の速度から速さ(スカラー)を算出し，T フレーム数分の空間平均速さを求める．更にそれを時間で平均し，時間平均速さを求め，これを口の動いた指標とする．　式(7)に空間平均速さ，式(8)に時間平均速さを示す．

$$\bar{v}_t = \frac{1}{X \times Y} \sum_{i=1}^{X} \sum_{j=1}^{Y} \sqrt{v_{x,t}(i,j)^2 + v_{y,t}(i,j)^2} \qquad (7)$$

$$\bar{v} = \frac{1}{T} \sum_{t=1}^{T} \bar{v}_t \qquad (8)$$

これを口の動いた指標とし，発話開始フレームと発話終了フレームを求める．

発話区間の定義は一つというわけでなく，他にも，口の開閉の速さと真顔からの差分値などの特徴を使用して，決定することもできる．

どの特徴を使った場合でも，全てのサンプルで同じ領域の特徴をとることが必要であるため，顔の正規化をしなければならない．

2.4.1.3　特徴ベクトルの抽出

ある時間幅について局所領域のフローの成分の平均速度，速度の大きさ，角度(radian)を計算し特徴ベクトルを抽出する．いま，あるサンプルの速度データは T”フレーム，n×m の局所領域の X，Y 方向成分の速度をもっている．時間 t，局所領域の位置(i，j)における速度の成分を ，　とするとき，特徴量を以下のように計算をする．

$$\bar{v}_{X,i,j} = \frac{1}{T''}\sum_{t=o}^{T''} v_{X,t}(i,j) \qquad (9)$$

$$\bar{v}_{Y,i,j} = \frac{1}{T''}\sum_{t=o}^{T''} v_{Y,t}(i,j) \qquad (10)$$

$$\bar{v}_{i,j} = \frac{1}{T''}\sum_{t=o}^{T''} \sqrt{v_{X,t}(i,j)^2 + v_{Y,t}(i,j)^2} \qquad (11)$$

$$\bar{\theta}_{i,j} = \frac{1}{T''}\sum_{t=o}^{T''} \tan^{-1}\frac{v_{Y,t}(i,j)}{v_{X,t}(i,j)}, \qquad (12)$$

この 4 つのパラメータ X，Y 方向成分の平均速度
$\bar{v}_{X,i,j}$ $\bar{v}_{Y,i,j}$，平均速さ$\bar{v}_{i,j}$，平均角度$\bar{\theta}_{i,j}$を使い，特徴ベクトル **F** を作成する．特徴ベクトル F は，発話区間 T’フレームを 5 つの区間に分割し，発話区間の抽出を行い，得られた発話区間 ***T’***を 5 分割し，各区間の時間幅を $\boldsymbol{T}''_{\mathbf{1}}$, …，$\boldsymbol{T}''_{\mathbf{n}}$, …, $\boldsymbol{T}''_{\boldsymbol{5}}$ とする．ここで, $\boldsymbol{T}''_{\mathbf{1}}$=…= $\boldsymbol{T}''_{\boldsymbol{5}}$=1/5 ***T’***である．分割された各区間で式(9)-(12)を計算する．特徴ベクトル **F** は **n×m**×4×5=20**nm** 次元のベクトルとなる．

$$\begin{aligned}\mathbf{F} &= \{\mathbf{F}_{T_1''}, \mathbf{F}_{T_2''}, \mathbf{F}_{T_3''}, \mathbf{F}_{T_4''}, , \mathbf{F}_{T_5''}\} \\ &= \{f_1, f_2, \cdots, f_k, \cdots, f_{20nm}\}\end{aligned} \qquad (13)$$

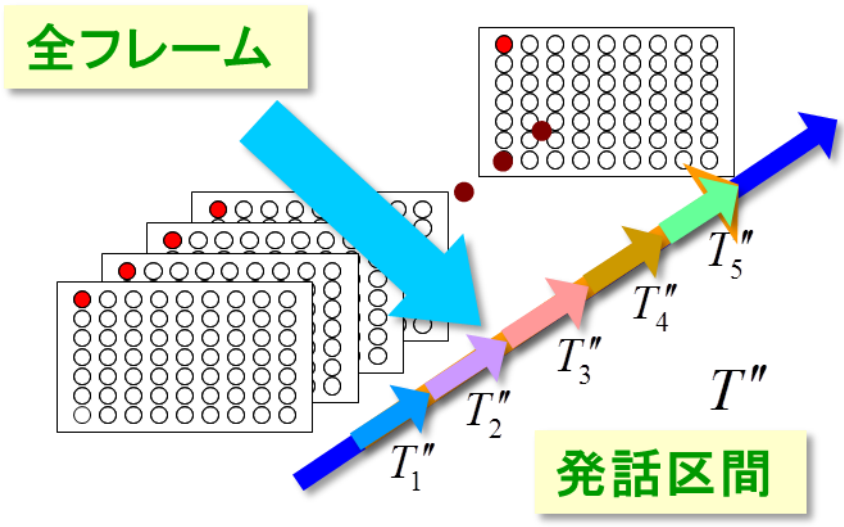


図 5　特徴ベクトルの作成概要

2.4.1.4　特徴の次元圧縮

多クラスへ拡張した Fisher の線形判別分析(Fisher Linear Discriminant Analysis)を Fisher の判別分析[19][20]と呼ぶ．多クラスに対する線形判別では，k 次元の特徴空間を $\tilde{k}$ 次元へ次元削減を行う．今回は Fisher の判別基準を用いて，クラス間の距離ができるだけ大きく，かつクラス内の距離ができるだけ小さな特徴空間を求め，次元圧縮を行う．

原特徴ベクトル $\mathbf{F}$ の第 k 成分のクラス内分散を $\mathrm{var}_w(k)$，クラス間分散を $\mathrm{var}_B(k)$ とする．各成分 k について以下の評価関数を計算し, 評価量の大きくなるものを特徴パラメータとする．

$$J(k) = \frac{\mathrm{var}_B(k)}{\mathrm{var}_w(k)} \qquad (14)$$

またクラス内分散，クラス間分散は以下のように計算を行う．

$$\mathrm{var}_w(k) = \sum_{i=1}^{c}\left(\frac{1}{n_i}\sum_{j\in\theta_i}\left(f_k^{(j)} - \bar{f}_k^{\,i}\right)^2\right) \qquad (15)$$

$$\mathrm{var}_B(k) = \sum_{i=1}^{c}\left(\bar{f}_k^{\,i} - \bar{f}_k\right)^2 \qquad (16)$$

ここで，$f_k^{(j)}$ はサンプル j の第 k 成分，c は識別するクラス数である．θ_i は i 番目のクラスのベクトルのサンプル集合を示し，n_i は i 番目のクラスのサンプル数を示す. また，$\bar{f}_k^{\,i}$ は i 番目のクラスのサンプル j に対する弟 k 成分 $f_k^{(j)}$ の平均値である．$\bar{f}_k$ はサンプル全体の第 k 成分の平均値である．

これを計算し，$J(k)$ を大きい順に並べ，上位 100 個の成分を選び，これらを成分とする低次元特徴ベクトル $\hat{\mathbf{F}}$ を決定する．

$$\hat{\mathbf{F}} = \{f_1, \cdots, f_k, \cdots f_{100}\} \qquad (17)$$

以上の手法を全サンプルより得られたベクトルに適用し，辞書を作成する．

2.4.1.5　識別判定方法

発声単語クラスの識別を行う K-Nearest-Neighbor 法[19]の概念を図 6 に示す.

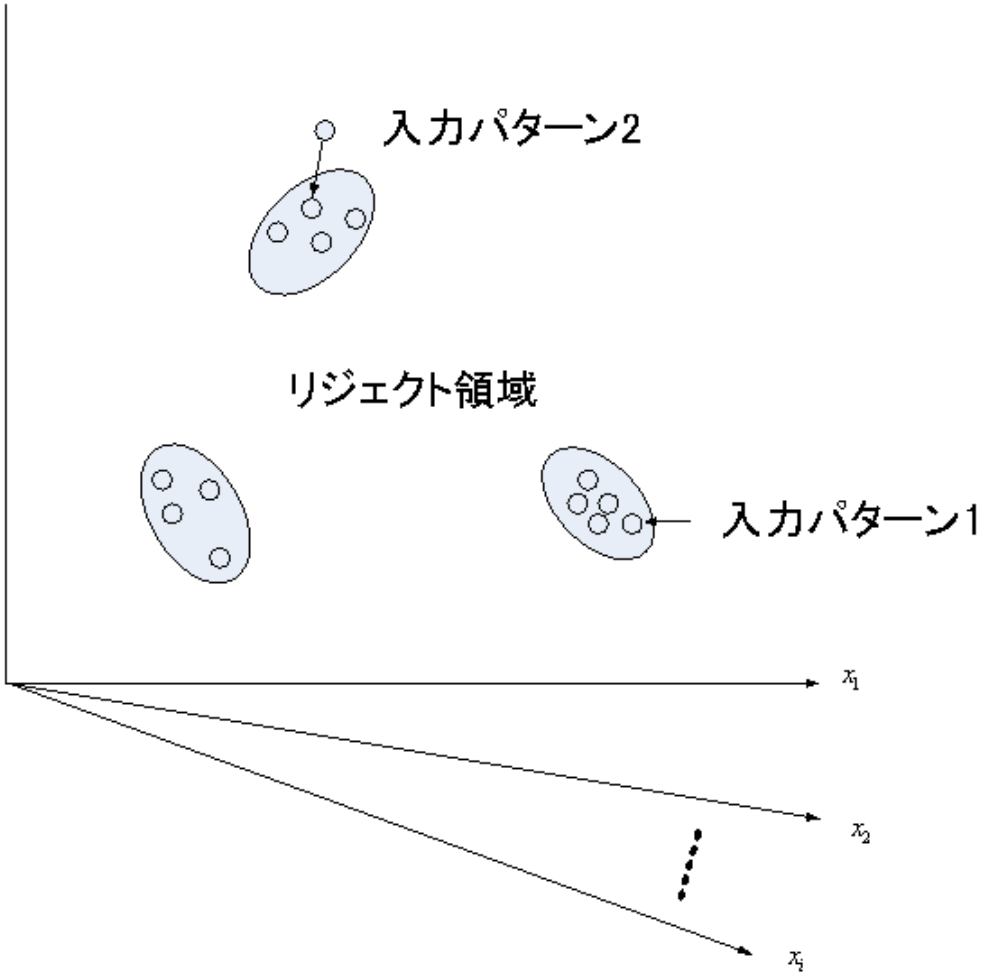


図 6　K-Nearest-Neighbor 法の概念

K-Nearest-Neighbor 法は入力パターンにもっとも近い k 個の学習サンプルをとり，その中で最も多数を占めたクラスを識別結果として出力する識別法の 1 つである．発声単語の特徴ベクトルを用い，特徴空間上での未知サンプルと辞書の学習サンプルの距離を求め，発声単語を判別する.

2.4.2 輝度値特徴を用いた手法

2.4.2.1　輝度値特徴とは

輝度の求めかたは 2 通りある．
一つは，各画素の R と G と B のそれぞれの値を加算して，3 で割る方法である．
もう一つが，各画素に対して，0．299×R + 0．587×G + 0．114×B で求める方法である．

前者と後者の見え方はさほど変わらないことが分かっている．また，輝度は２５６階調で表されるのが一般的である．

ここで，画像を扱うにあたり，必要となる基本的なことを説明する．

パソコン内の画像は，全てデジタル化されているため，画素という最小単位のもので構成されています．画素のことをピクセルともいいます．
フルカラーである場合，各々の画素は R，G，B のカラーデータを持っています．
もし，画像の解像度が M×N であるならば，その画像を **X** とすると，

X=[RGBRGBRGB…RGB]という長さ M×N×3 というベクトルで表すことが可能である．
M と N の大きさが大きくなる程，ベクトルの長さが大きくなることが分かる．

ベクトルで表すことができるということは，この画像に対して，四則演算が可能になったこともわかる．

例えば，「2 枚の画像を平均する」といった場合，それぞれの画像をベクトルに直し，

$$\overline{\mathbf{X}} = (\mathbf{X} + \mathbf{Y})/2 \qquad (18)$$

で計算することができる．これは複数の画像に対しても有効である．
ただし, 全ての画像ベクトルの大きさを揃える必要があるので, RGB をそのまま使うのか, 輝度値に直してから使うのかは，各自で考えていただきたい．

今回は 1 つのサンプル **X** に対して，時系列の画像があるので，1 つの画像 **x** ，フレーム数をすると，$\mathbf{X}=(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$で表すことができる．

2.4.2.2　発話ピークの抽出

2．4．1．2 と同様の方法で発話区間を抽出し，発話区間内の輝度特徴のみの取得を行い，更に，発話ピークフレームを求める．発話ピークフレームとは，各音節を発話した瞬間のフレームのことを指す.
今回は 2 通りの発話ピークフレーム抽出を以下のように定義する.

(a)　各フレームの空間平均速さを式(7)より求め，前後 1 フレームの速度差の絶対値の和をフレーム数分取得する．その後，その値の大きい順から上位 10 フレームを選択.
以上の条件を満たしたフレームを発話ピークフレームとして，それを取得する.

(b)　各フレームの空間平均速さを式(7)より求め，更にそれを平均し，時間平均速さを式(8)より取得する．各フレームの空間平均速さが時間平均速さを超えたフレームだけを発話ピークフレームとして取得する.

(a)で得られた 10 フレームの発話ピーク画像をベクトルに表すと，$\mathbf{x}_1$，$\mathbf{x}_2$，…，$\mathbf{x}_{10}$ と表すとき，$\mathbf{X}=(x_1, x_2, \ldots, x_{10})$を 1 サンプルの特徴ベクトルとし，全サンプル分のベクトル $\mathbf{X}_1$，$\mathbf{X}_2$，…，$\mathbf{X}_m$ を作成した．(m：サンプル数)

(b)で得られた発話ピークフレームは，各サンプルで異なるため，数を揃えなければならない．そこで，再サンプリング処理をかけて，数を揃える作業を行い，$\mathbf{X}=(x_1, x_2, \ldots, x_t)$(t: 基準となるフレーム)を 1 サンプルの特徴ベクトルとし，全サンプル分のベクトル $\mathbf{X}_1$，$\mathbf{X}_2$，…，$\mathbf{X}_m$ の作成をする．(m：サンプル数)

2.4.2.3　再サンプリングによる時間軸の正規化

発話ピークフレーム抽出を行ったことで，各サンプルのフレーム数は異なってしまっている．そこで，全サンプルのフレーム数を統一するために，以下のような時間軸の再サンプリングを行う．

フレーム数T'の再サンプリング前の輝度値を$\mathbf{v}$，フレーム数T''の再サンプリング後の輝度値を$\mathbf{V}$とする．1 フレームをサンプリング点 1 点と考え，サンプリング点T'個の$\mathbf{v}$を用い，サンプリング点T''個の$\mathbf{V}$上での輝度値を推定していく(図 7)

今，$\mathbf{v}$のサンプリング点の間隔が 1 であるとすると，$\mathbf{V}$のサンプリング点の間隔は$\frac{T'}{T''}$となることがわかる．このことから，$\mathbf{v}$の$\frac{T'}{T''}+1$番目の点が$\mathbf{V}$の 1 番目の点となる．また，$\mathbf{v}$の第$\frac{T'}{T''}n+1$番目の点が，$\mathbf{V}$上ではn番目の点と対応することになる．$(n \le T'')$

そこで，$\mathbf{V}$の第i番目の輝度値V_iは

$$V_i = v_{\lfloor N \rfloor} + \left(v_{\lceil N \rceil} - v_{\lfloor N \rfloor}\right) \times \left(N - \lfloor N \rfloor\right)$$
$$N = \frac{T'}{T''} i + 1 \qquad (19)$$

で求めることができる．式(19)と図 8 は，V_iを求めるためには$\mathbf{v}$の$\frac{T'}{T''}i+1$番目の点の前後の輝度値から算出できることを表している．

以上の方法を用いて，各フレームの輝度値に適用しそれぞれの成分で再サンプリングを行っていく．

また，今回は T”は 2．4．2．2(b)で求めた発話ピークフレームの最大値に設定した．

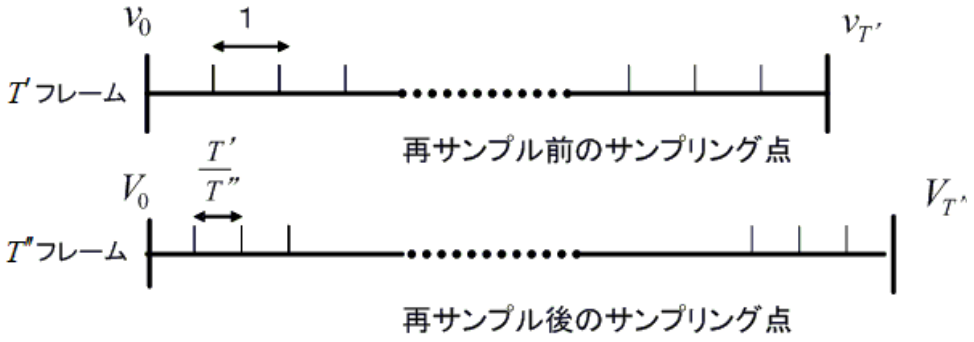


図 7　再サンプル前後のサンプリング点

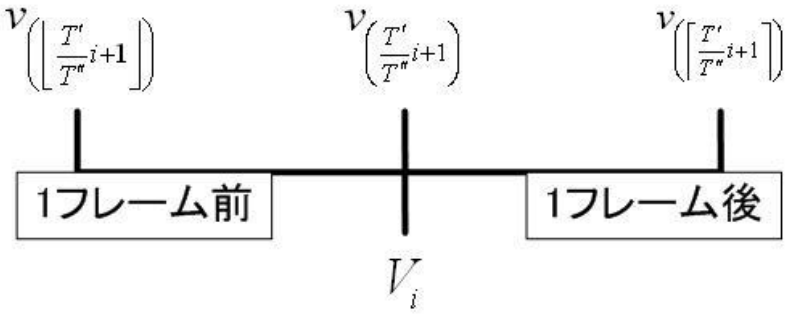


図 8　再サンプル後の値の算出

2.4.2.4　部分空間法による辞書の作成

部分空間法は，各クラスごとにそのクラスを表現する低次元の部分空間を用意し，未知サンプルがどの部分空間で最もよく近似表現できるかを比較することにより，未知パターンを識別する方法である．

今回は CLAFIC 法(CLAss-Featuring Information Compression)を用いることにした．CLAFIC 法では，部分空間の基底を決めるために，自己相関行列を用いて主成分分析を行い，認識時には，データを部分空間に射影した時の大きさを類似度として用いる手法である．本論文では，7 つの発話クラスそれぞれの部分空間を算出し，辞書として登録した．

部分空間法を以下に示す．

ここで，k 個のクラス c_1，c_2，…，c_k のそれぞれの部分空間を L_1，L_2，…，L_k，その次元を d_1，d_2，…，d_k とする．各クラス c_i について部分空間 L_i を張る d_i 個の d 次元正規直交ベクトルを $\mathbf{u}_{i1}$，$\mathbf{u}_{i2}$，…，$\mathbf{u}_{id}$ とする．

クラス c_i に着目し，d 次元特徴空間から d_i 次元部分空間への変換を表す行列を $\mathbf{A}_i$ とすると，$\mathbf{A}_i=(\mathbf{u}_{i1}, \ldots, \mathbf{u}_{idi})$と書ける．この式より，

$$A_i^t A_i = I \tag{20}$$

が成り立つ．I は単位行列である．部分空間に射影された di 次元特徴ベクトル$A_i^t x$を元の d 次元空間で見ると $A_i^t A_i$ となり，元の空間から部分空間 L_i への変換は直交射影行列

$$P_i = A_i A_i^t = \sum_{j=1}^{d_i} u_{ij} u_{ij}^t \tag{21}$$

によって表すことができる．

そして，$\mathbf{x}$ の部分空間 L_i への正射影は $P_i\mathbf{x}$ である．

次に，部分空間の基底を求めるための主成分分析について，説明する．

2.4.2.5　主成分分析

さまざまな見え方をもつ顔画像をベクトル表現した多次元ベクトルについて，主成分分析を適用することでそのベクトルの多様性を少数のパラメータで表現することができる．

以下にその計算手順を示す．なお，N 次元ベクトル $\mathbf{x}_m(m=1,2,\cdots,M)$ は M 個の顔画像の輝度値ベクトルを表すものとする．

(ⅰ)サンプル集合に含まれる M 個の顔画像を表す N 次元ベクトル $\mathbf{x}_m(m=1,2,\cdots,M)$ の平均ベクトル $\boldsymbol{\mu}$ を求める．

$$\boldsymbol{\mu}=\frac{1}{M}\sum_{m=1}^{M}\mathbf{x}_m \tag{22}$$

(ⅱ) M 個の N 次元ベクトル $\mathbf{x}_m(m=1,2,\cdots,M)$ をサンプル全体の平均ベクトル $\boldsymbol{\mu}$ との差分ベクトル $\boldsymbol{\varphi}_m(m=1,2,\cdots,M)$ に変換する．

$$\boldsymbol{\varphi}_m=\mathbf{x}_m-\boldsymbol{\mu}\ \ (m=1,2,\cdots,M) \tag{23}$$

差分ベクトル $\boldsymbol{\varphi}_m(m=1,2,\cdots,M)$ の第 i 成分を $\phi_{m,i}(i=1,2,\cdots,N)$ とすると，

$\phi_m{}^t=(\phi_{m,1},\phi_{m,2},\cdots,\phi_{m,N})$ と表される．

(ⅲ) M 個の差分ベクトル $\boldsymbol{\varphi}_m(m=1,2,\cdots,M)$ を並べた $N\times M$ の行列を $\mathbf{A}$ とする．

$$\mathbf{A}=(\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2,\cdots,\boldsymbol{\varphi}_M) \tag{24}$$

(ⅳ)行列 $\mathbf{A}$ とその転置行列 $\mathbf{A}^t$ との積により，$\mathbf{R}=\frac{1}{M}\mathbf{A}\cdot\mathbf{A}^t$ のような $N\times N$ の標本共分散行列 $\mathbf{R}$ を求め，$\mathbf{R}$ に対する固有値，固有ベクトルを求めるのだが，学習サンプル $\mathbf{x}_m(m=1,2,\cdots,M)$ の次元数 N は膨大な値の多次元ベクトルになるため固有値, 固有ベクトルを求めることは困難である．そこで特異値分解により計算を以下のように求めることにした．行列 $\mathbf{A}$ の転置行列 $\mathbf{A}^t$ を求め，$\mathbf{A}^t\cdot\mathbf{A}$ によって $M\times M$ の行列 $\mathbf{L}$ を導く．

$$\mathbf{L} = \mathbf{A}^t \cdot \mathbf{A} = \begin{bmatrix} L_{1,1} & \cdots & L_{1,M} \\ \ddots & & \\ \vdots & L_{m,n} & \vdots \\ & \ddots & \\ L_{M,1} & \cdots & L_{M,M} \end{bmatrix} \tag{25}$$

ただし， $L_{m,n}$ は $L_{m,n} = \frac{1}{M}\sum_{i=1}^{N} \phi_{m,i} \cdot \phi_{n,i}$ $(m = 1,2,\cdots,M, n = 1,2,\cdots,M)$ である．

(ｖ) $M \times M$ の行列 $\mathbf{L}$ について， M 個の固有ベクトル $\mathbf{V}_l (l = 1,2,\cdots,M)$ を求める．

$$\mathbf{L}\mathbf{V}_l = \lambda_l \mathbf{V}_l \quad (l = 1,2,\cdots,M) \tag{26}$$

ただし， $l = 1,2,\cdots,M$ の順序は，固有値 λ_l の大きい順に定めるものとする．

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_l \geq \cdots \geq \lambda_M \tag{27}$$

(ⅵ) M 次元ベクトル $\mathbf{V}_l$ から，以下の式(28)のように N 次元ベクトル $\boldsymbol{\psi}_l (l = 1,2,\cdots,M)$ を求める．

$$\begin{aligned} \boldsymbol{\psi}_l &= \mathbf{A} \cdot \mathbf{V}_l \\ &= (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_M) \cdot \mathbf{V}_l \\ &= \sum_{m=1}^{M} v_{i,m} \cdot \boldsymbol{\varphi}_m \quad (l = 1,2,\cdots,M) \end{aligned} \tag{28}$$

M 個の N 次元ベクトル $\boldsymbol{\psi}_l (l = 1,2,\cdots,M)$ は $N \times N$ 行列 $\mathbf{A}\mathbf{A}^t$ の固有ベクトルであり，式(29)を満足することが知られている．

$$\mathbf{A}\mathbf{A}^t \cdot \boldsymbol{\psi}_l = \lambda_l \cdot \boldsymbol{\psi}_l \quad (l = 1,2,\cdots,M) \tag{29}$$

(vii) N 次元ベクトル $\boldsymbol{\psi}_l (l=1,2,\cdots,M)$ を正規化し，単位ベクトル $\mathbf{U}_k (k=1,2,\cdots,K)$ を求める．

$$\mathbf{U}_k = \frac{1}{\sqrt{\sum_{i=1}^{N} {\psi_{k,i}}^2}} \boldsymbol{\psi}_k \tag{30}$$

ただし，${\boldsymbol{\psi}_k}^t = (\psi_{k,1}, \psi_{k,2}, \cdots, \psi_{k,N})$ とする．ところで式(6-3)により，$\mathbf{A} = (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_M)$ であったから $N \times N$ 行列 $\mathbf{A}\mathbf{A}^t$ は

$$\mathbf{A}\mathbf{A}^t = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\varphi}_m {\boldsymbol{\varphi}_m}^t \tag{31}$$

となり，$\mathbf{A}\mathbf{A}^t$ は M 個の N 次元ベクトル $\mathbf{x}_m (m=1,2,\cdots,M)$ の標本共分散行列に他ならない．M 個の N 次元ベクトル $\mathbf{U}_k (k=1,2,\cdots,K)$ は，この標本共分散行列に互いに直交する大きさ 1 の固有ベクトルとなる．したがって $\mathbf{U}_k (k=1,2,\cdots,K)$ は，サンプル集合 M 個の N 次元ベクトル $\mathbf{x}_m (m=1,2,\cdots,M)$ を主成分分析することによって求める正規直交基底となる．

(viii) N 次元ベクトル $\mathbf{x}_m (m=1,2,\cdots,M)$ で表された車体の形状ベクトルまたは，テクスチャはパラメータ空間上で k 次元ベクトル $\mathbf{f}_m (m=1,2,\cdots,M)$ として次元圧縮される．

$f_{m,k} (m=1,2,\cdots,M, k=1,2,\cdots,K \le M)$ と表される $\mathbf{f}_m (m=1,2,\cdots,M)$ の第 k 成分は式(32)のように k 番目の正規直交基底 $\mathbf{U}_k (k=1,2,\cdots,K)$ に $\mathbf{x}_m - \boldsymbol{\mu}$ $(m=1,2,\cdots,M)$ に射影することによって得られる．

$$f_{m,k} = {\mathbf{U}_k}^t \cdot (\mathbf{x}_m - \boldsymbol{\mu}) \ (m=1,2,\cdots,M, k=1,2,\cdots,K \le M) \tag{32}$$

今回は次元圧縮を行っていないため，(viii)は省略している．
このようにして求めた，各部分空間を張る正規直交基底を用いて射影行列 P を作成する．

2.4.2.6　識別判定の方法

部分空間法で学習された辞書と，未知サンプルの類似度 S を計算し，一番類似度の高いクラスに属すると識別する．部分空間法の類似度は部分空間への射影の長さで定義され，

$$S_i(\mathbf{x}) = \left\| \mathbf{P}_i \mathbf{x} \right\|^2 = \mathbf{x}^t \mathbf{P}_i \mathbf{x} \tag{33}$$

で計算することができる．さらに効率の良い計算方法を求めると，

$$S_i(\mathbf{x}) = \sum_{j=1}^{d_i} \left(u_{ij}^t \mathbf{x} \right)^2 \tag{34}$$

となる．この値が大きい程，類似度が高いと判定される．

今回の発話単語は 7 個なので，7 クラスそれぞれの空間を作成し，類似度を計算し，一番類似度の高いクラスを正解単語とした．

2.4.3 識別性能の評価法

本研究では識別性能の評価法として leave-one-out 法[19]を用いる．
leave-one-out 法とは，n サンプルの中から 1 サンプルを選んで未知サンプルとし，残りの n-1 サンプルを学習サンプルとして辞書を差作成し，1 サンプルの識別を行い，これを n 個の未知サンプルについて繰り返しテストする．

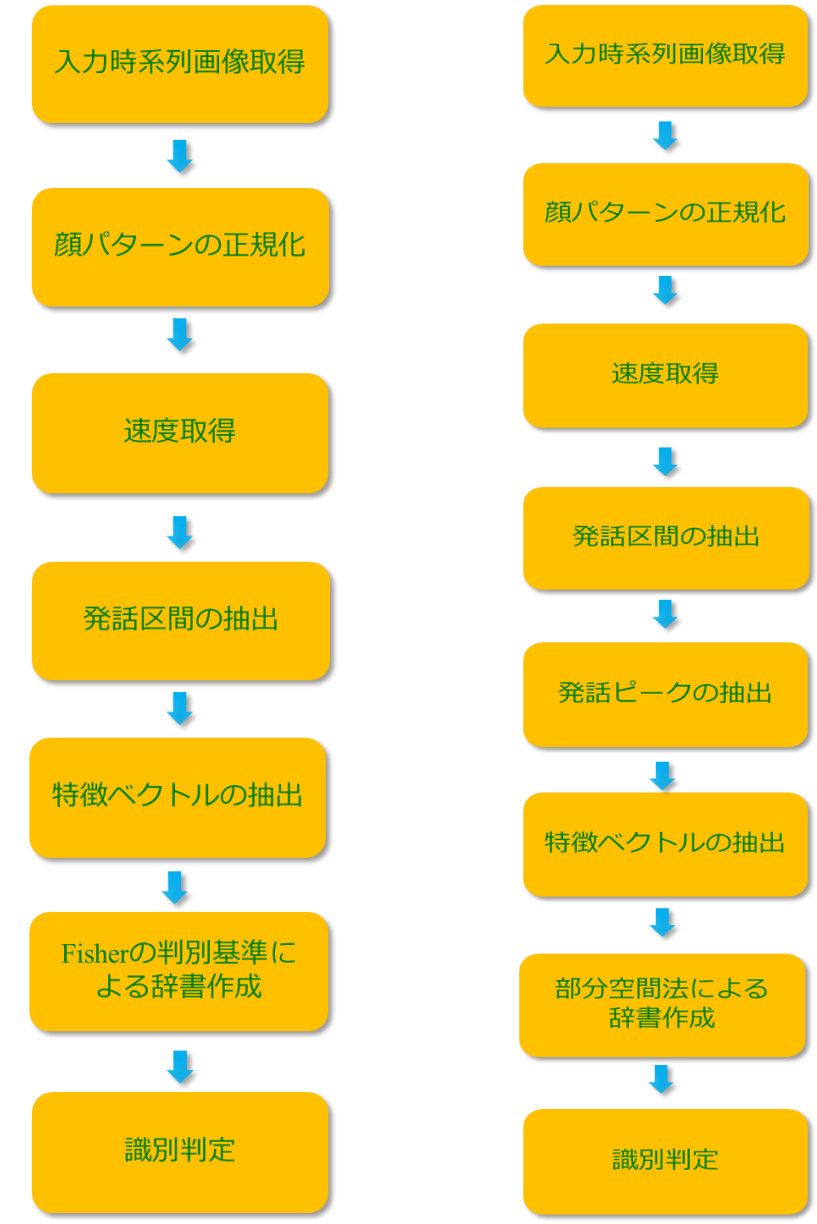
第3章　発声単語認識システムについて

3.1システムの概要

3.1.1 全体の流れ

本研究の全体の流れを図 9 に示す.

一連のシステムを C#言語と OpenCV，DirectShow を用いて構築する.



(a)Optical Flow を用いた流れ　　(b)部分空間法を用いた流れ

図 9　本研究の流れ

3.1.2 OpenCV と DirectShow について

OpenCV[21]とは Intel 社による画像処理向けのライブラリのことで, Open Source Computer Vision Library の略である.

OpenCV は画像処理プログラムを行う上で有用なライブラリであり, 画像処理を記述する労力を大きく削減できる.

DirectShow[22]は Microsoft 社の Windows 用マルチメディア拡張 API 群である DirectX に含まれる API の 1 つである．DirectShow を使うことで，動画や音声などの大容量のマルチメディアデータをストリーミング再生(データを読み込みながら同時に再生すること)することができるようになる．DirectShow を利用すれば，ハードウェアの違いを気にすることなく，動画や音声などの再生制御をアプリケーションソフトから手軽に行えるようになる.

また，OpenCV，DirectShow は C#言語は非対応である．そこで，今回は参考文献[23]にある Visual Basic 用の DLL(ダイナミックライブラリ)を使用し，間接的に OpenCV，DirectShow の関数を呼び出してくることにした.

3．1．3　研究環境

今回研究に使用した環境を以下に示す.

＜研究環境＞

動画撮影 PC & 実験処理 PC

OS	：	Microsoft Windows 7
言語	：	C＃言語，OpenCV，DirectShow
CPU	：	Inetel Core2 Quad 3．0Ghz
メモリ	：	DDR2 4．00GB
開発ツール	：	Microsoft Visual C# 2008 Express Edition
Web カメラ	：	Microsoft LifeCamVX-7000

本実験システムにおいて行列とベクトルを扱うプログラムを提供していただいた D1 稲葉善典氏には感謝する. また, 画像処理を施す部分は OpenCV に寄与するところが多々ある. 画像処理については後述する.

以下で処理の流れ(理論)にそった詳しい詳細と，作成したアプリケーションの解説をしていく.

3.2動画撮影

3.2.1 動画撮影条件

本研究では，動画の撮影を 2 回行っている．

第一期に取得した条件は以下のようになっている．
顔の正面画像を Web カメラから取得する．このとき以下のような条件をつけて撮影する．

① 正面顔を維持してもらうように指定
② 各音節の発話のタイミングは指定
③ 発話後はそのままの口の形を維持

今回はサイズ 320×240 で 30FPS の動画を撮影した．

また，6 人×5 単語×4 回の計 120 サンプルを用意した．
発話してもらう単語は「ありがとう」「おめでとう」「こんにちは」「すみません」「さようなら」の 5 単語である．

第二期に取得した条件は以下のようになっている．
顔の正面画像を Web カメラから取得する．このとき以下のような条件をつけて撮影する．

① 正面顔を維持してもらうように指定
② 発話語は真顔に戻してもらう

今回はサイズ 320×240 で 30FPS の動画を撮影した．

また，6 人×7 単語×4 回の計 168 サンプルを用意した．
発話してもらう単語は「ありがとう」「いいえ」「おめでとう」「こんにちは」「すみません」「さようなら」「はい」の 7 単語である．

1 人 25 回という回数を行ってもらうため，所要時間がかかる，インタフェース部に難しい作業がある，といったことは被験者の方に大きなストレスを与えかねない．そこで，時間に無駄のないよう，かつインタフェースが簡単であることが，いいサンプルをとれることにつながると思われる．よって，そのような心得のもとアプリケーションの開発を行った．

また，動画取得中に照明による輝度の変化が見られたときはもう一度協力をしてもらう．これは今後の処理で時間的な動きを着目しているため，輝度の変化も動きに取られてしまうためである．

3.2.2 動画撮影用システム

第一期と第二期で使用した動画撮影用システムの解説をする.
第一期で使用したシステムは「MovieRecorder」という自作のソフトを用いている.

MovieRecorder は時間を指定して，動画(サイズは任意)を撮影することができるようにした.
起動後，名前，単語名を入力するフォーム(図 10)が表示される.
PC の USB に Web カメラが差してあれば，自動的に認識してカメラを起動する.
OK ボタンを押したあと図 11 のようなフォームが表示される.
カメラ起動ボタンをクリックすると fmCameraOne に入力画像が 30FPS で描画されるループが開始する.
スタートを押すと，タイマーが切り替わり，WordViewer にカウントダウンが表示されたあと，各単語の 1 音だけが 5/4 秒間隔で表示されるので，タイミングを合わせて発話する.
全単語が発話し終わると，自動的にカメラはストップする.
最後に保存ボタンを押して，保存コーデックを選択し，取得した画像を無圧縮の avi に保存する.
今回はカメラからの入力画像を DirectShow の SampleGrabber クラスを用いて，Bitmap を取得し，List 型で指定時間分取得する.

また, 動画の保存時に全 Bitmap を OpenCV の cvCreateVideoWriter と cvWriteFrame いう関数を用いた.

MovieRecoder は現在 2 つのカメラまで使用することができるように拡張してある.

本プログラムは被験者のストレスをかけないように，ボタン 3 つで動画取得を行える．起動ボタン，スタートボタン，保存ボタンを押すだけなので，インタフェースの操作は容易である.
ただ，最初のフォームの位置とカメラの着ける位置が対応できないため，最初だけはフォームの位置合わせを手動でする必要がある.

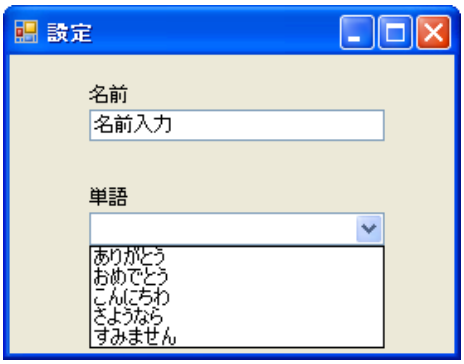


図 10　入力フォーム

図 11　起動後のフォーム

第二期で使用したシステムは Microsoft 社製の Web カメラ「LifeCam」付属のソフトである．LifeCam で撮影された動画は解像度 320×240 で，30FPS で撮影を行っている．こちらの方が容易に音声付動画を撮影することができる．

また，撮影時の操作は，被験者が行うのではなく，こちらの合図に従って，発話の開始を促した．図 12　LifeCam の起動画面に起動画面を示す．

ビデオのアイコンボタンを押すと Rec が始まる．

図 12　LifeCam の起動画面

3.3時系列画像の切り出し

撮影した動画像を 1 フレームごとの画像に分割して時系列画像の作成を行う．また，同時に，音声データの保存も行う．

使用したソフトは AviUtl[24]というソフトである．このソフトは拡張機能もあるため，自由度の高いソフトウェアである．
時系列画像の切り出しには参考文献[24]の HP から「aviutl 本体」と「連番 BMP 出力」という Plugin をダウンロードして使用する．
また，LifeCam のソフトウェアで取得した画像のコーデックが wmv なので，読み込めないため，参考文献[25]の HP から「DirectShow File Reader プラグイン for AviUtl」をダウンロードし，インストールを行う．図 13 に起動画面を示す．

ファイルを読み込ませたら，プラグイン出力で連番 BMP を選択し，ファイル名を付けて保存する．また，この時，音声ファイルも同時に保存をする．

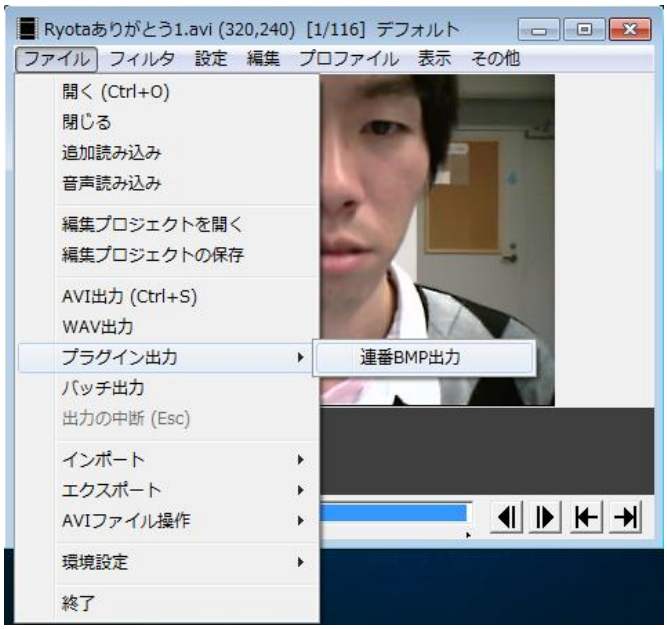


図 13　AviUtl の起動画面

3.4顔画像の正規化

得られた時系列の画像の顔は各被験者によって位置や大きさが異なっている．そこで，両目座標の取得を行い，それを用いて正規化を行うことにする．
両目座標を得るために，accFace^{TN} SDK を用いて「accFaceWithppm」というソフトの作成をした．
accFace^{TN} SDK は C++で作成されたライブラリなので，C++で作成した．

本ソフトは ppm ファイルのみ読み込み可能なので，事前に画像ファイルを「bmp2ppm」などのソフトで ppm ファイルに変換する必要がある．また，accFace は 320×240 の画像のみにしか対応していないため，注意すること．

実行手順を以下に示す．
. . ¥accFaceWithppm¥Sample¥Windows¥Debug¥Sample．exe を実行し，図 14 の起動画面を出す．

読み込みボタンを押して，連番 ppm をオーダー通りに複数選択する(最初のファイルにカーソルを当てて，Shift を押しながら，最後のファイルを選択)

実行ボタンを押す．

パラメータの変更は基本的にいらないので，気にしない．

両目座標の保存先は, . . ¥accFaceWithppm¥Sample¥Windows¥Eyepoint．txt である．

Eyepoint．txt の中身は，向かって左目の x 座標，左目の y 座標，右目の x 座標，右目の y 座標の順で保存されている．

また，1 サンプル保存することに Eyepoint．txt をリネームまたは移動を行う必要がある．

図 15 に accFace での両目座標の取得例を示す．

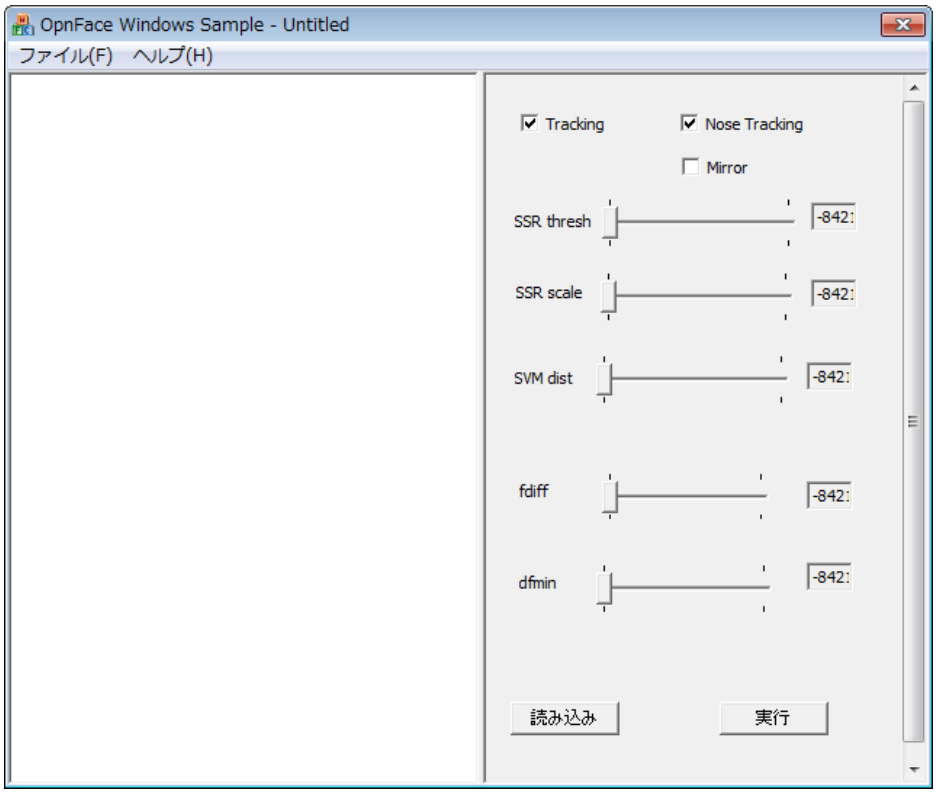


図 14　accFaceWithppm の起動画面

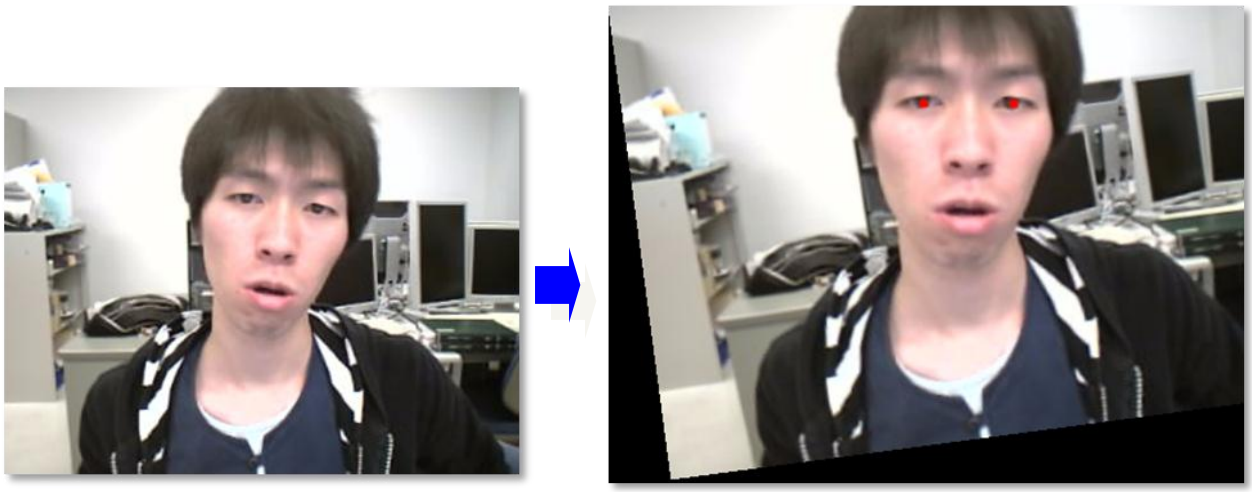


図 15　accFace による両目座標の取得

3.5Optical Flow の取得

3.5.1 Optical Flow 取得にあたって

本研究は，時系列画像から Optical Flow を算出し，それを特徴量や指標としている．時間的な動きを推定していくのだから，顔の動き，カメラのノイズ，輝度の変化等の時間的ノイズを緩和しなくてはならないということになる．今回は，輝度はそのサンプル中は一様であると仮定し，カメラのノイズ除去処理を行うことにした．

3.5.2 前処理

Optical Flow を取得する前に時系列画像ノイズ除去のためにモノクロ化とガウシアンフィルタを施す．
モノクロ化は C#のみでは時間がかかるため(高速化可能であり実装済みである), OpenCV の cvCvtColor 関数を使用し，ポインタを使用し計算速度を高速にする．cvCvtColor は第 3 引数にコンバートする処理を設定することができ，RGB から HSV 表色系や YCrCb 表色系への変換も容易に行うことができる．

その後，ガウシアンフィルタを OpenCV の cvSmooth 関数を使用し，平滑化を施す．
ガウシアンフィルタは正規分布で画像の平滑化を行う荷重平均フィルタである．移動平均フィルタより，注目点の画素情報が残り，周りの 8 近傍の画素との差を滑らかにするフィルタである．
cvSmooth は第 3 引数に平滑化を施すカーネルの設定ができ，ガウス関数の他に，単純平滑化やメディアンフィルタなどを設定することができる．また，処理を施すブロックのサイズも設定可能なので，ノイズ処理を容易に行える．
今回はパラメータとして，7×7 で平滑化を行う．また，ガウス関数 σ の値は標準偏差で計算を行う．
この他にゴマしおノイズというノイズを除去するために，メディアンフィルタというものもある．
以下に移動平均フィルタ，ガウシアンフィルタ，メディアンフィルタの例を図 16 に示す．

図 16　平滑化フィルタ(左から移動平均，ガウシアン，メディアンフィルタ)

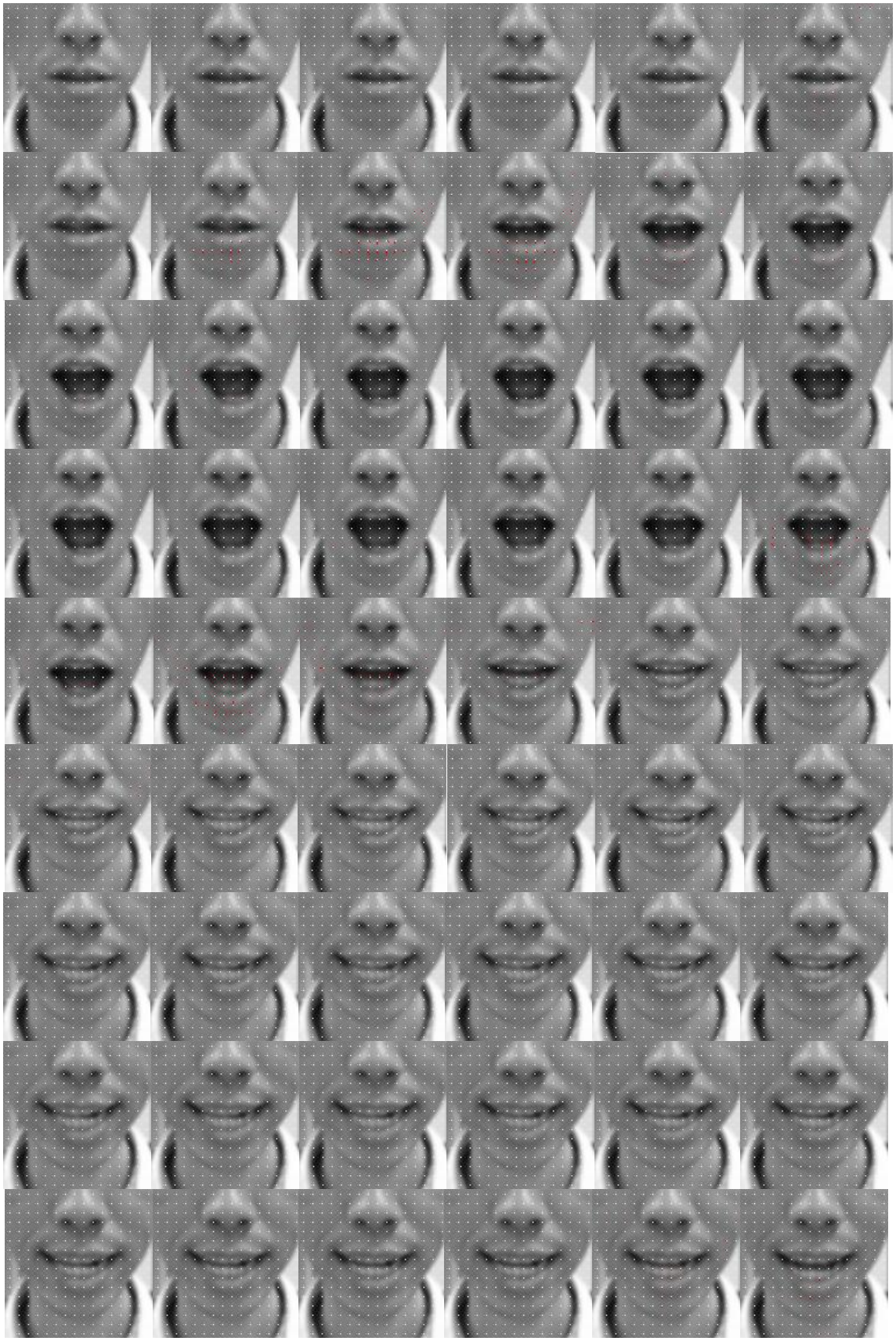
前処理後の時系列画像に対し，画像の切り出し処理を行う．切り出す範囲は切り出す中心を指定して，128×128 のサイズにする．

本研究では口周辺の動きを求めたいので，accFace で取得した両目座標を基に，正規化を行い，口周辺を切り出す．図 17 に画像の切り出し例を示す．

切り出した画像に対して，Optical Flow を(全画像数-1)フレーム分を取得していく．
Optical Flow は Lucas-Kanade 法を用いるため，OpneCV 内の cvCalcOpticalFlowLK 関数を用いた．

また，フローが乱れてしまっている影響を排除するため，8×8 の小領域にわけ，局所的にフローを平均する．また，その中心を特徴点と定め，局所平均速度を特徴点の速度とし，それを T フレーム分取得する．今回，128×128 のサイズの画像を用いたため，特徴点の数は 16×16 と計算できる．さらにこの特徴点から，口周辺の 9×7 の特徴点の X,Y 方向の速度のみを速度データとして取得した．
図 18 に Optical Flow を取得した画像例をいくつか示す．図 19 には口周辺取得例を示す．
また，取得した X,Y 方向の速度はバイナリ形式で保存をし，「mtxlist」という拡張子で保存を行う．この時，同時に画像の輝度値も保存を行っている．

図 17　画像の切り出し



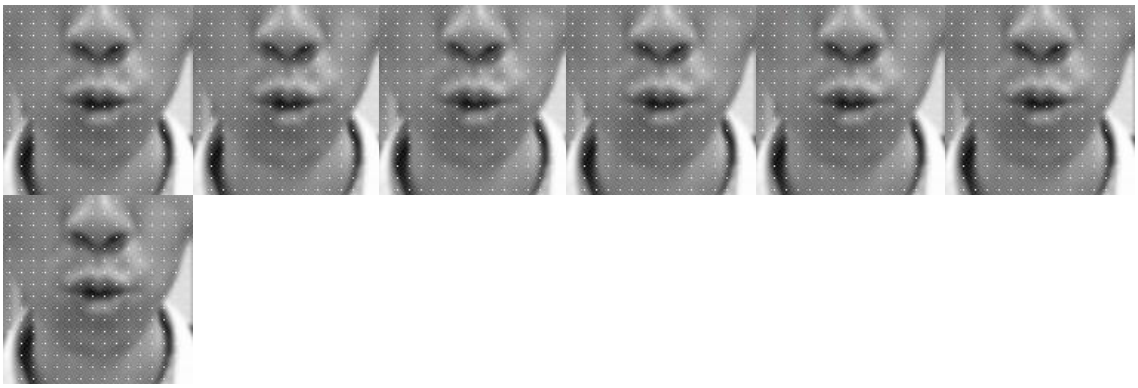


図 18　Optical Flow を取得した画像

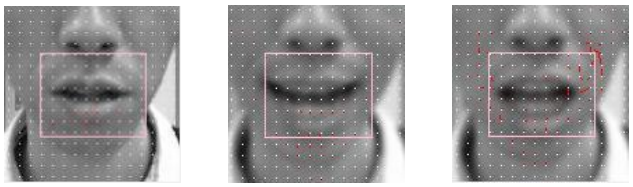


図 19　口周辺速度取得例

3.5.3 Optical Flow 取得用プログラム

Optical Flow 取得用のプログラムと局所領域速度取得のソフトの使い方を以下に示す.

プログラム名　LipReadingWithOpticalFlow

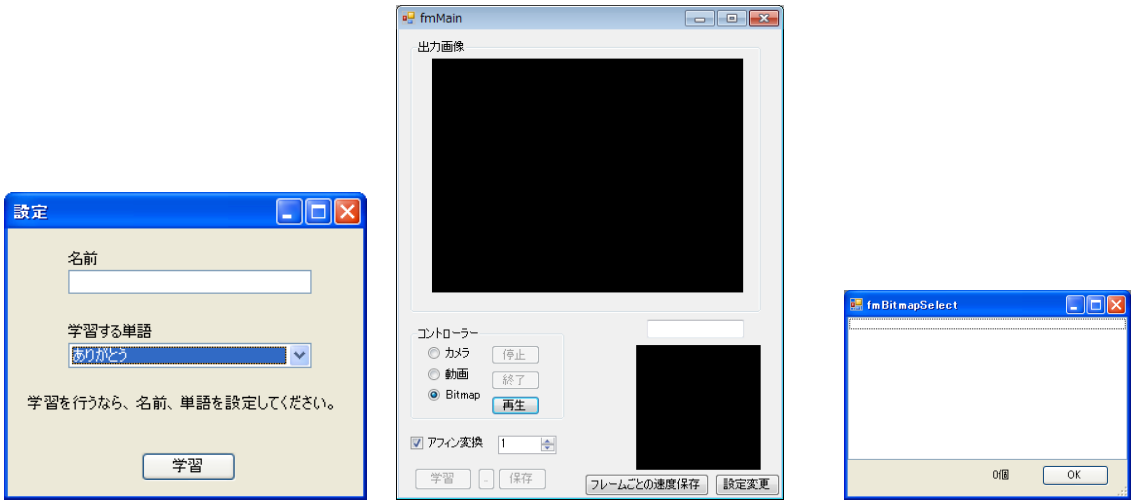


図 20　システムの外観(右から(a)(b)(c))

まず，上図(a)のような設定フォームで名前，学習する単語を指定する.
学習ボタンを押すと(b)のようなフォームが表示される.
入力はカメラ，動画，時系列画像から選択可能である．今回は時系列の Bitmap を用いたので，Bitmap にチェックして再生ボタンを押す.
次に，(c)のようなダイアログが開かれるので，画像の入ったフォルダごとドラッグ＆ドロップを行い，OK を押す．その後，上に記述したような前処理を行う.
前処理が終わったら，学習ボタンを押し，(b)の上下の PictureBox に時系列画像が表示されると同時に Optical Flow を取得する.

また，口を中心に切り取りを行うのは，上の PictureBox をクリックして 2 点を選択すると，切り出す場所を指定できる.
アフィン変換にチェックを入れると，指定した 2 点から，基準の大きさとの倍率と回転角度を計算し，入力画像にアフィン変換を行う.
今回の場合，accFace で得られたテキストを読み込み，その座標からアフィン変換を行うようになっている．また，そのテキストを読み込む際に，画像を読み込んだ順に選択を行うこと.

「フレームごとの速度保存ボタン」を押すと，”mtxlist”という拡張子のファイルに速度データと画像も輝度値が保存される．
以降，この mtxlist を用いて，処理を行っていく．

3.6　局所領域速度取得用プログラム

取得した速度データは，全特徴点における速度である．ここから，口周辺の 9×7 の特徴点のみを抽出していく．
GetFlowArea というソフトでそれを行う．

ソフトを実行すると図 21 のようなフォームが表示される．
全特徴点の速度データの保存してある mtxlist ファイルをドラッグ&ドロップをし，実行ボタンを押す．

図 21 中の右に表示された画像を元に, 抽出する左上の X, Y 座標と横幅, 高さを指定する．中途半端な場所を選んだとしても，その中の特徴点の数のみを計算しているので，見たままの特徴点の速度データのみが取得できる．

最後に保存ダイアログが表示されるので，フォルダを選択し，ドロップされたファイルすべての X,Y 方向の口周辺速度データと輝度値を mtxlist ファイルとして取得する．

尚，このソフトは口周辺だけでなく，指定したエリアならどこでも取得可能である．

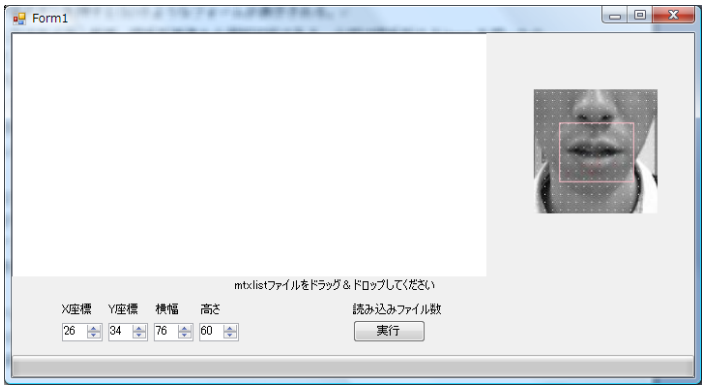


図 21　GetFlowArea の外観

3.7　発話区間の設定

全時間に対しての Optical Flow には発話をしている時間としていない時間のどちらも含まれている．そこで，発話している時間だけの速度を使い関係ないデータを含めないようにするために，発話区間の設定を行う．

3.7.1 発話区間の条件

2.4.1.2 でも書いたが，以下に発話開始フレームの検出条件と発話終了フレームの検出条件とその詳細を示す．
取得したフレーム数をTフレームとし，発話区間後の設定後のフレーム数はT'とする．

まず，各フレームについて 9×7 の特徴点におけるX,Y方向の速度から速さ(スカラー)を算出し，Tフレーム数分の空間平均速さ$\bar{v}_t$を求める．更にそれを時間で平均し，時間平均速さ$\bar{v}$を求め，これを口の動いた指標とする．

発話開始フレーム：各フレームの空間平均速さが時間平均速さを超えたフレームだけを抽出し，最初フレームを発話開始フレームとする．

発話終了フレーム：各フレームの空間平均速さが時間平均速さを超えたフレームだけを抽出し，最後のフレームを発話開始フレームとする．
以上の条件を満たした開始フレームと終了フレームの間の T’フレームを発話区間として抽出し，そのフレーム数分の Optical Flow のみを取得をした．

以上の処理を適用し，全サンプルの区間T'フレームのみのX,Y方向の速度データと輝度値を抽出する．図 22　発話区間設定例 に発話区間を設定した例を示す．

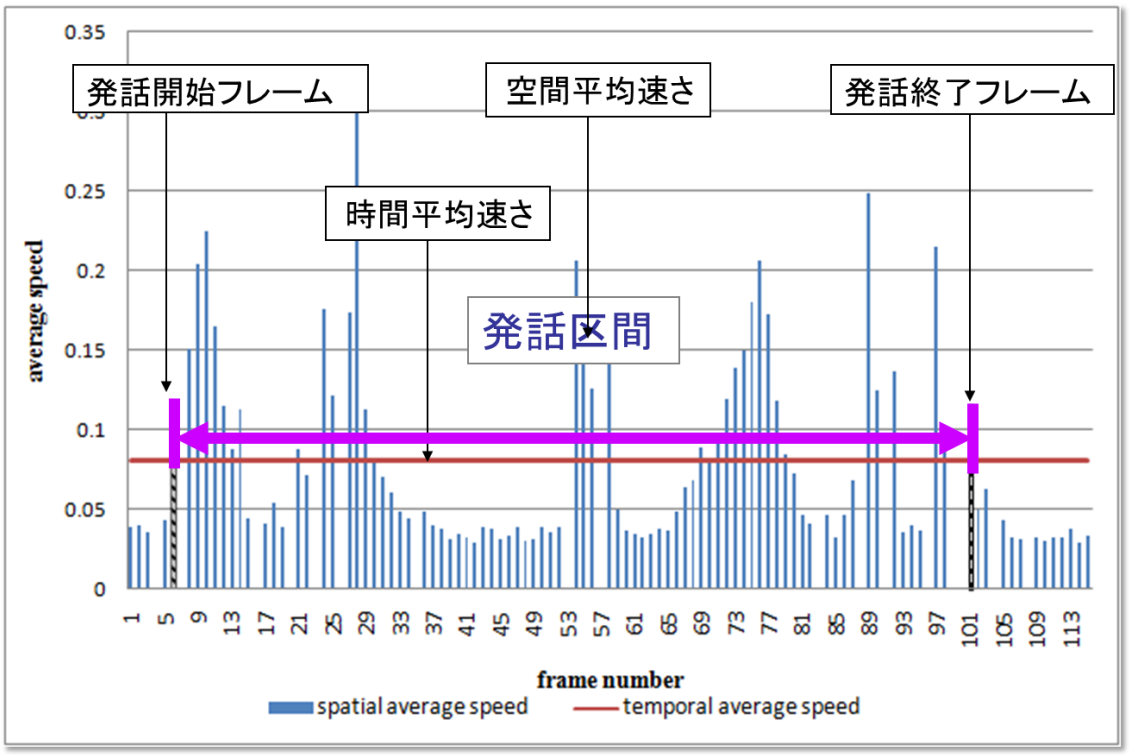

発話開始フレーム
空間平均速さ
発話終了フレーム
時間平均速さ
発話区間
average speed
0.35
0.25
0.2
0.15
0.1
0.05
0
1
5
9
13
17
21
25
29
33
37
41
45
49
53
57
61
65
69
73
77
81
85
89
93
97
101
105
109
113
frame number
spatial average speed
temporal average speed

図 22　発話区間設定例

3.7.2 発話区間抽出プログラム

発話区間の抽出のソフトとは，ReSample というものである．
ReSample は発話区間を抽出し，時間軸の再サンプリングを行う機能を持っているが，今回時間軸の再サンプリングを行わないため，機能していない．図 23 に起動画面を示す．

読み込めるファイル拡張子は mtxlist のみとなっている．
ファイルを読み込んだ後，実行ボタンを押すと，処理が始まり，保存ダイアログが表示されたら，保存を行う．

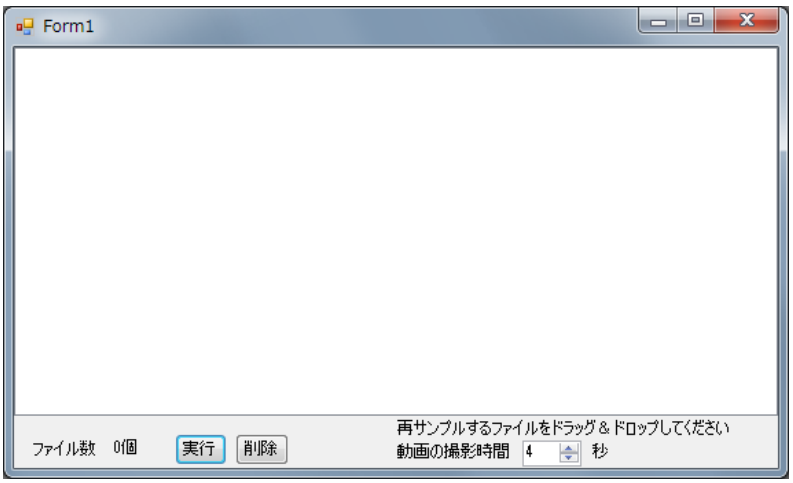


図 23　発話区間抽出用ソフトの起動画面

3.8　発話ピークフレームの抽出

3.8.1 発話ピークフレームの条件

2.4.2.2 でも書いたが，発話ピークフレームとは，各音節を発話した瞬間のフレームのことを指す．

以下に定義を再提示する．

(a) 各フレームの空間平均速さを式(7)より求め，前後 1 フレームの速度差の絶対値の和をフレーム数分取得する．その後，その値の大きい順から上位 10 フレームを選択．
以上の条件を満たしたフレームを発話ピークフレームとして，それを取得する．

(b) 各フレームの空間平均速さを式(7)より求め，更にそれを平均し，時間平均速さを式(8)より取得する．各フレームの空間平均速さが時間平均速さを超えたフレームだけを発話ピークフレームとして取得する．

3.8.2 発話ピークフレーム取得用プログラム

発話ピークフレーム取得用のソフトは PeekDetection というものである．図 24 に起動画面を示す．

PeekDetection は(a)(b)それぞれの条件で発話ピークフレームのみを保存できる機能を持っているが，それを切り替えるためのインタフェースの実装は行っていない．

なので，ソースコード上で使用するメソッドを変更しなければならない．
DetectPeek(int PeekCount)が上から PeekCount 分のデータだけ取り出すメソッドで，
DetectPeekUsingAverageSpeed()が各フレームの空間平均速さが時間平均速さを超えたフレームだけを取り出すメソッドである．

読み込みファイルは mtxlist のみである．
発話ピークフレームを保存すると，発話ピークフレームの画像が保存される．

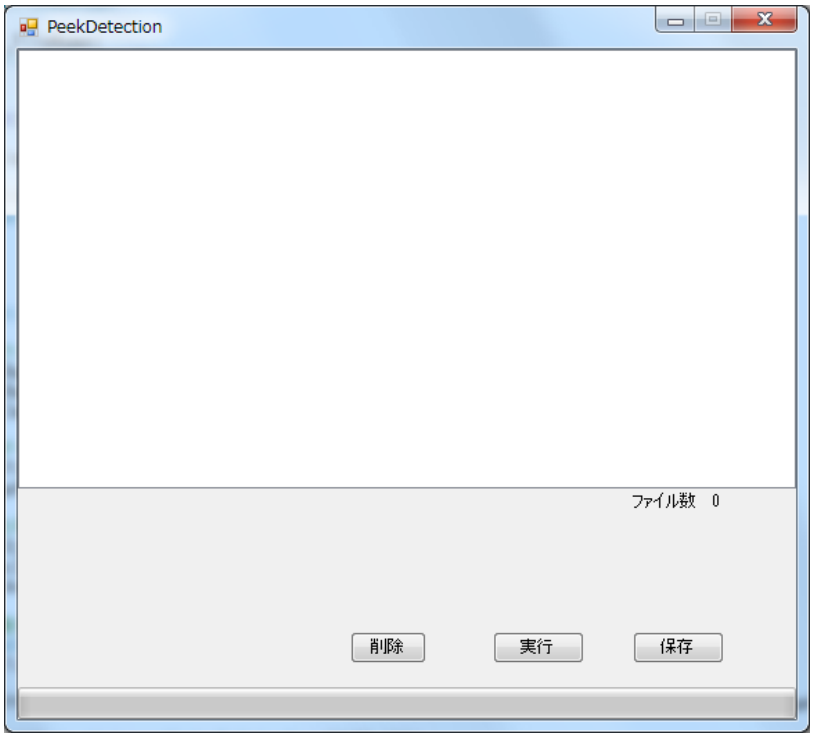


図 24　PeekDetection の起動画面

3.9　時間軸の再サンプリングによる正規化

3.9.1 再サンプリングの定義

発話ピークフレームの抽出(b)を行ったことで，各サンプルのフレーム数は異なってしまっている．そこで，全サンプルのフレーム数を統一するために時間軸の再サンプリングを行う．理論は 2.4.2.3 で述べた通りで，1 フレーム前と後のフレームからその中間の速さを算出するということである．今回は再サンプリング後のフレーム数をT''とするとT''は 2. 4. 2. 2(b)で求めた発話ピークフレームの最大値にした．(10 フレームだった)

この再サンプリング方法を考える上でアイディアを下さった同学年猪俣拓利氏，D1 稲葉善典氏には感謝する．

3.9.2 再サンプリング用プログラム

再サンプリングを行うソフトは，ResampleImages である．図 25 に起動画面を示す．
読み込みは複数の画像を 1 つのフォルダにしたものが対応している．
全サンプルのフォルダを一度に読み込むことができる．

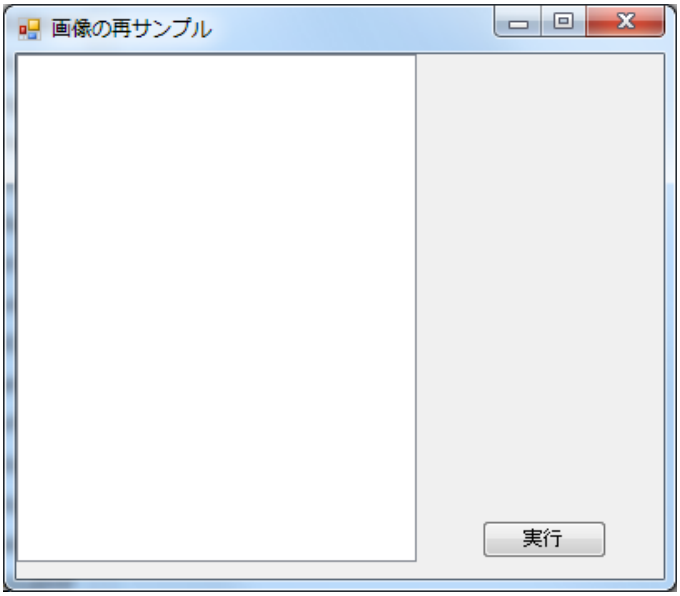


図 25　ResampleImages の起動画面

再サンプリングを行った例を図 26 に示す.
これ見ると再サンプリングされていることがわかる.

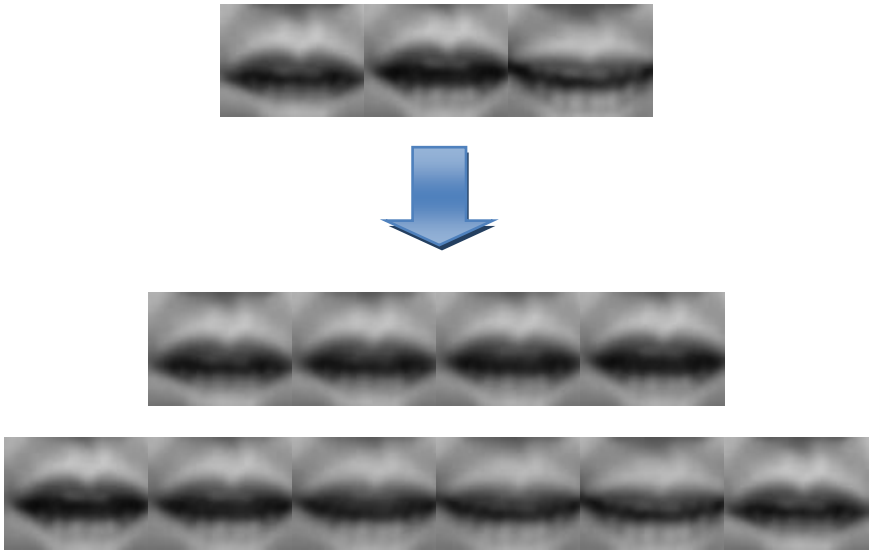


図 26　再サンプリングの例

3.10　特徴ベクトルの抽出

今回，2 種類の特徴量を比較するために，2 種類の特徴ベクトルを作成する．

3.10.1　Optical Flow を用いた特徴ベクトルの作成方法

得られた速度データから特徴ベクトルを作成する．詳細は 2.4.1.3 に記述した通りである．

3.10.2　輝度値特徴を用いた特徴ベクトルの作成方法

得られた再サンプリング画像より，輝度値特徴ベクトルを作成する．今回は各画素の (R+G+B)/3 を計算した輝度値 $G_{n,\ m}$ を並べて，フレーム順にベクトルを作成し，結合を行った．

$$G= \{G1,\ 1,\ 1,\ G1,\ 2,\ 1,\ \ldots,\ Gn,\ m,\ t\} \tag{35}$$

(n，m:画像の横幅と高さ，t:フレーム番号)

3.10.3　Optcal Flow 特徴ベクトル作成用プログラム

以下に特徴ベクトルを作成するために作成したソフトを説明する．CreateFeatureVector というソフトを使用する．起動画面を図 27 に示す．
mtxlist ファイルをドラッグ&ドロップする．
平均するフレーム数に時間を何分割するか選択する．
今回は 5 分割したので，5 といれた．

また，特徴ベクトルの作成が終わると，csv 形式でサンプル数分保存される．この csv ファイルを手動で各単語クラスのフォルダを作成しそれぞれに割り振っておく．

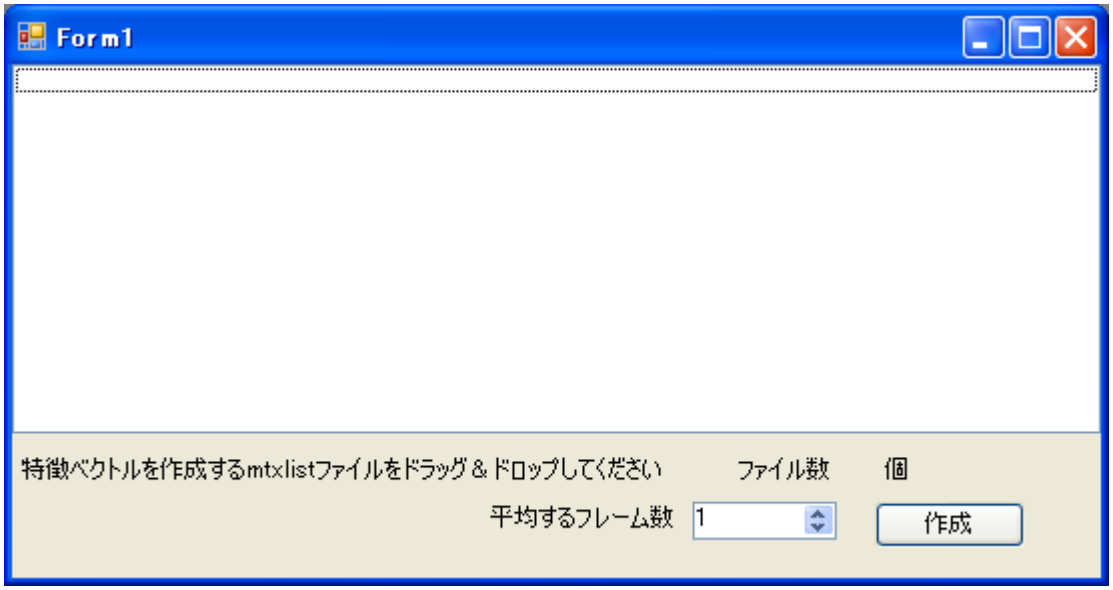


図 27　CreateFeatureVector の起動画面

3.10.4　輝度値特徴ベクトル作成用プログラム

ソフトの名前は，BitmapToCsv である．これは複数の画像が入ったフォルダを読み込み，Csv ファイルにベクトルを書き出す機能を持っている．図 28 に起動画面を示す．

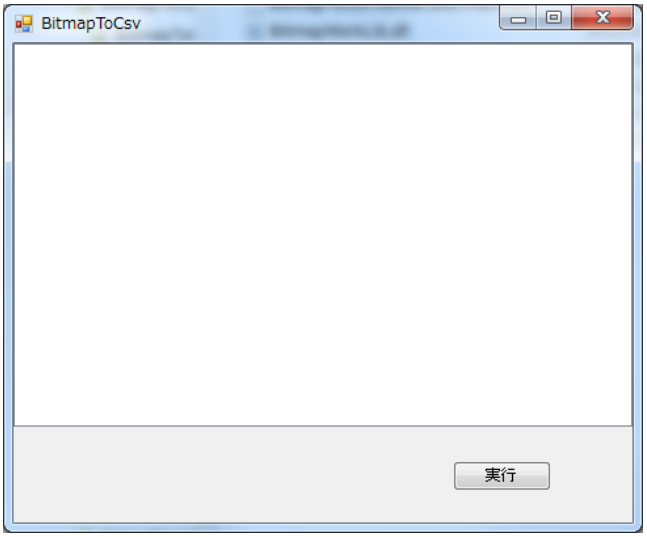


図 28　BitmapToCasv の起動画面

3.11　辞書の作成

3.11.1　　Fisher の判別基準

Optical Flow 特徴で得た特徴ベクトルは高次元になっているため, 次元の呪いの問題がある. そこで，Fisher の判別分析で使われる Fisher の判別基準を用いて，次元圧縮を行うこととする.
Fisher の判別基準については 2.4.1.4 を参照されたい.
式(14)(15)(16)を用いて特徴ベクトルの次元を低次元に圧縮し，判別に有効な成分だけを取り出す.

取り出したものを辞書に登録を行う.

3.11.2　　部分空間法

発話ピークフレームの画像各発話単語で見え方が異なると思われる．そこで，各クラスの特徴を表す空間を作成し，類似度を計算するために部分空間の形成を行う．部分空間を形成するためには主成分分析を行う必要がある.

3.11.3　Fisher の判別基準を用いた次元圧縮用

以下に次元圧縮を行うためのプログラムを示す. ソフト名は FisherDiscriminantAnalysis Ver. 1 である.

クラス分けした特徴ベクトルのフォルダをドラッグ&ドロップする.

元々の特徴ベクトルの次元数は未知であるため，次元数の最大値は 100 としてある.

今回，100 次元まで次元を圧縮するため，次元数は 100 のまま使用する.

処理が終わると，指定したフォルダの中に「FDA」というフォルダが作成されている．これが次元圧縮を行った特徴ベクトルである．これが辞書になる.

中身の csv ファイルは 1 列目に特徴次元，2 列目にその値が順に並んでいる.

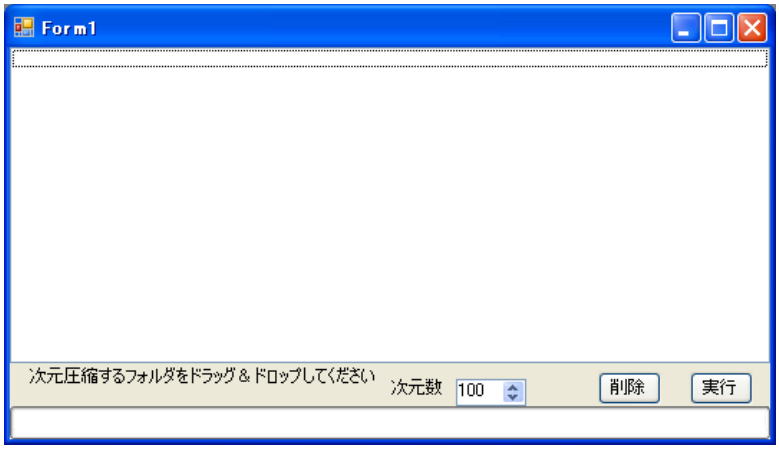


図 29　FisherDiscriminantAnalysis Ver．1 の起動画面

3.11.4　主成分分析プログラム

主成分分析を行うソフト名は PrincipalComponentAnalysisβ である．これは D1 稲葉氏の作成したプログラムを自己相関行列で主成分分析できるように直し，かつ，読み込みの仕方を 2 通りできるに改造したものである．

1 通り目は，クラスごとに分類された特徴ベクトルの入ったフォルダを読み込ませて，クラスごとの部分空間を作成する

2 通り目は，leave-one-out 法を用いて，複数のファイルから 1 つ抜き出して，ファイル数分主成分分析を行い，部分空間を作成するもの

図 30 に起動画面を示す．

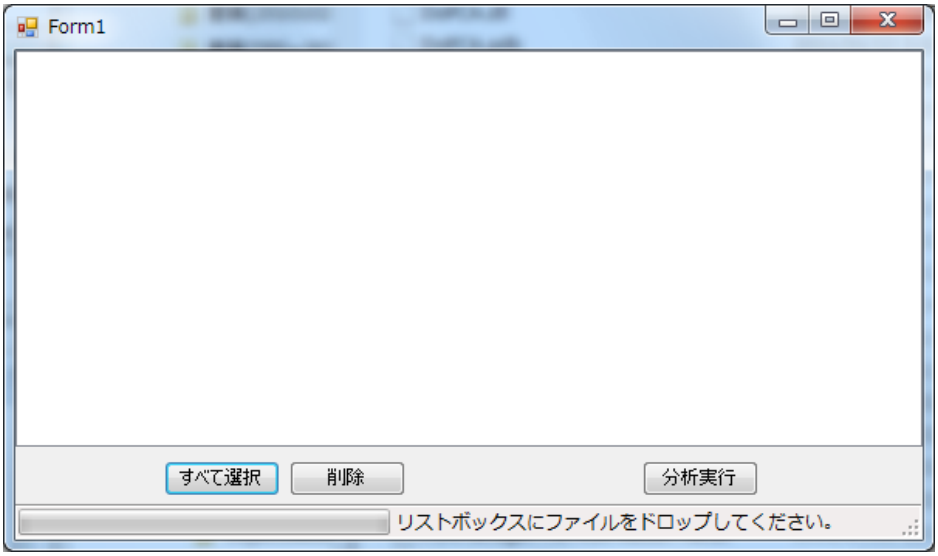


図 30　PrincipalComponentAnalysisβ の起動画面

3.12　識別判定

3.12.1　k-NN 法を用いた識別

今回，Optcal Flow 特徴の識別判定の方法として，k-Nearest-Neighbor 法[19]を用いた．ここでは, k=3 として, 各未知サンプルに対して, 距離が近い学習サンプルの上位 3 個を選び, その所属するクラスの多数決によって識別を行う．もし同時に 3 つの異なるクラスが選択されたときはリジェクトとし，識別を行わないこととした．

3.12.2　部分空間法の類似度を用いた識別

部分空間法の類似度は，部分空間への射影の長さで定義されている．その類似度の一番高いと判定されたクラスに投票を行うことにする．

3.12.3　k-NN 法のプログラム

k-NN 法を行うソフトは，K-NearestNeibor である．
プログラムの起動画面を図 31 に示す．
テストサンプルの特徴ベクトルファイル(csv 形式)を図 31(a)にドラッグ&ドロップする．
実行ボタンを押すと図 31(b)が現れるので，作成した FDA を含んだフォルダ(辞書)を選択し OK を押す．

最後に保存のダイアログが表示されるので，保存を行う．

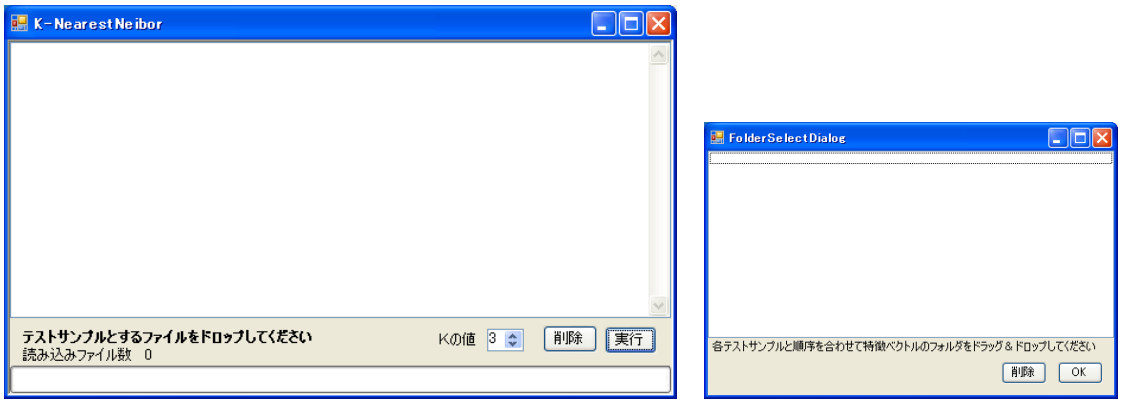


図 31　K-NearestNeibor の起動画面(右(a)左(b))

3.12.4 部分空間法による類似度の類似度計算プログラム

類似度求めるためのソフトは SubSpaceClassificationMethod である．起動画面を図 32 に示す．

判別する学習部分空間には，leave-one-out で 1 サンプル抜かれて作成した，1 クラスの部分空間を読み込ませる．Pca データの入った mtx ファイルを読み込む．

学習部分空間には，判別する学習部分空間で読み込ませたクラス以外の部分空間，これは全サンプルで学習した空間，を読み込ませる．

テストサンプルには，leave-one-out で抜き出した未知サンプルを読み込せる．

実行ボタンを押すと，類似度を計算し，テキストファイルに保存される．

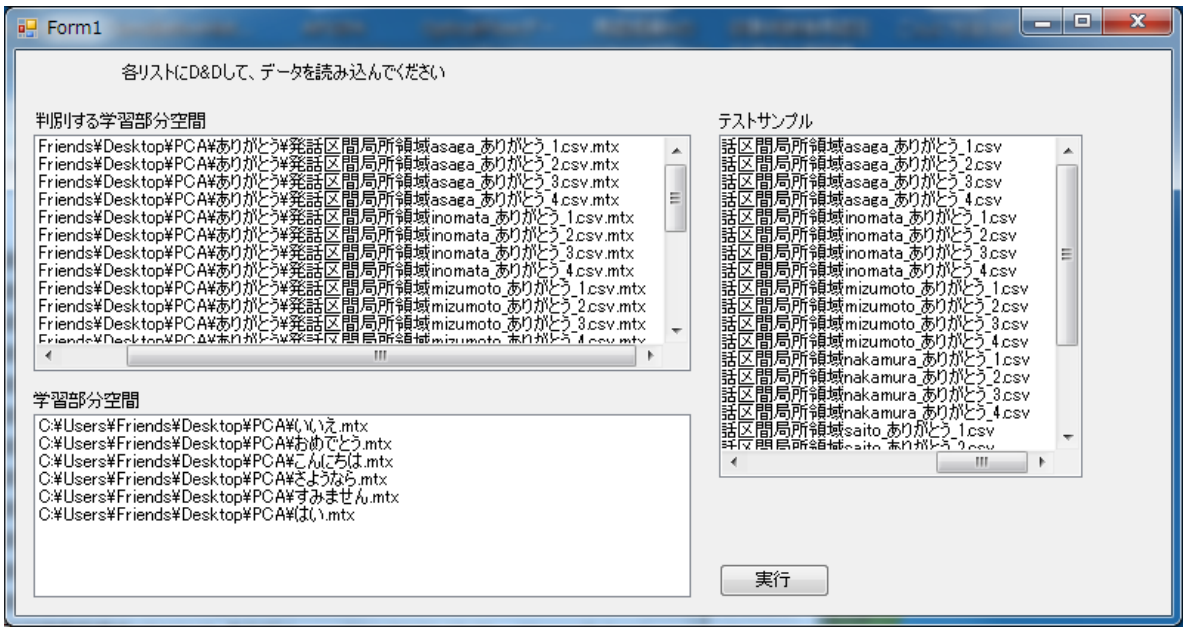


図 32　SubSpaceClassificationMethod の起動画面

第4章　実験・考察

4.1　サンプルデータの収集

3.2.1 の条件のもと，サンプルを収集する．
第一期では，6 人に「ありがとう」「おめでとう」「こんにちは」「さようなら」「すみません」の 5 単語を各 4 回ずつ発話してもらった．つまり 6 人×5 単語×4 回の計 120 サンプルを用意した．

第二期では，6 人に「ありがとう」「いいえ」「おめでとう」「こんにちは」「さようなら」「すみません」「はい」の 7 単語を各 4 回ずつ発話してもらった．つまり 6 人×7 単語×4 回の計 168 サンプルを用意した．

4.2　識別性能の評価方法

前述したように，識別性能の評価法として leave-one-out 法を用いて，繰り返し識別を行い評価を行う．

4.3　抽出された発話区間の検証

発話区間が正確に抽出されていることを確認するために，Optical Flow より得られた発話区間と音声信号より得られた発話区間の結果の比較を行った.

手順を以下に示す.

(1) 表 1 に示す，4 種類の条件で被験者 4 人から各 4 回，計 64 サンプルを取得する．この時，被験者には「HO-U-SE-I DA-I-GA-KU」と発話してもらう.

(2) 音声信号の発話開始ポイントと発話終了ポイントを目視で取得し, 発話区間を抽出する.

(3) 2．5．で示したように Optical Flow より得られた速度を用いて，時系列画像から発話開始ポイントと発話終了ポイントを取得し，発話区間を抽出する.

(4) (2)と(3)の区間差を計算する．このとき，(2)-(3)の順番で計算を行う.

図 33 に比較の結果を示す．この図を見ると，Optical Flow より求められた発話区間は音声信号から得られた発話区間よりほとんど全てにおいて長いことが分かる．次に，(4)で求めた開始フレームと終了フレームの差を図 34(a)，(b)に示す．Optical Flow より得られた発話開始フレームはいつも音声信号のフレームより前にあることが分かる．また，終了フレームはその逆であると見てとれる. 言い換えれば, 音声信号より得られた発話区間は Optical Flow より得られた区間の中に含まれていると考えられる．これらの結果から， 画像情報を用いて抽出された発話区間には以下の 3 つの区間が存在することが確認された.

(1) 発話開始準備区間
(2) 音声発話区間
(3) 発話終了準備区間

図 35 に音声信号と Optical Flow による発話区間の開始・終了フレームの差の平均と標準偏差を 4 種の取得条件ごとに示す．これらの実験の結果から，Optical Flow を用いて抽出された発話区間の精度は多少の顔の動きには影響しないことが分かった.

表 1 動画取得時の 4 条件

a	発話中は動かない
b	発話中に前後に動く
c	発話中に左右に動く
d	発話中に回転

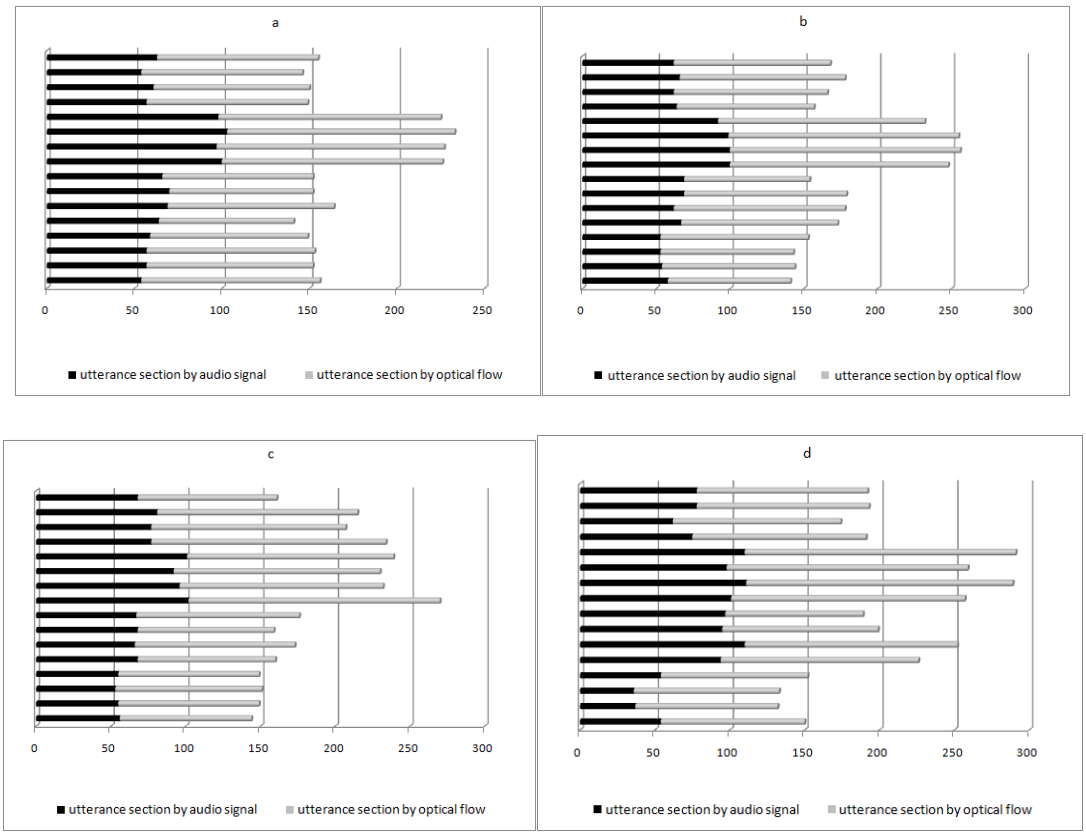
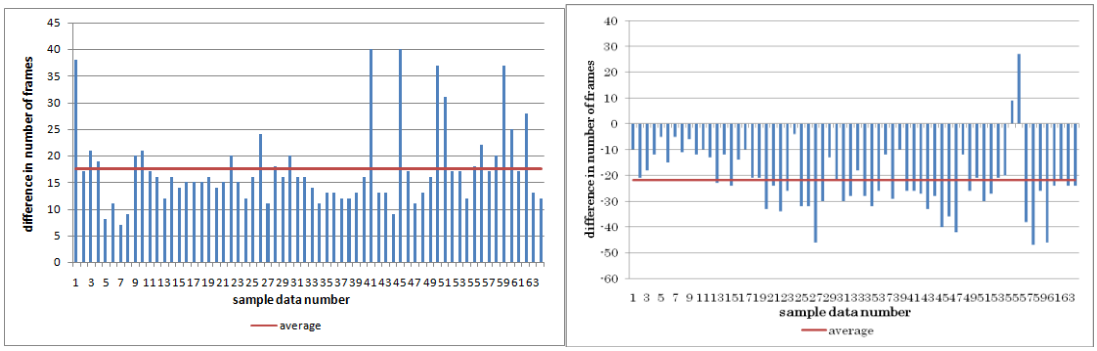
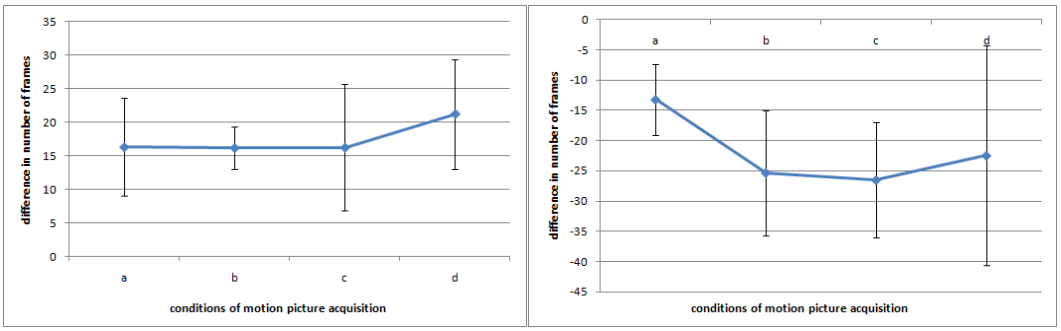


図 33 検出された発話区間の長さ



(a)検出された発話開始フレームの差　　(b) 検出された発話終了フレームの差

図 34　音声信号と Optical Flow によって検出された発話区間の開始・終了フレームの差



(a)開始フレーム　　(b)終了フレーム

図 35 各発話区間の開始・終了フレームの差の平均と標準偏差

4.4　各注目領域から得られた動きの評価

第一期の動画を使用

特徴ベクトルを作成する際に，注目する局所領域を以下のようにして，特徴ベクトルを作成した.

口周辺：口の中心の 9×7 の局所領域から得られた 1260 次元の特徴を 100 次元に圧縮した場合

頬部分：頬を含む上側の 16×5 の局所領域得られた 1600 次元の特徴を 100 次元に圧縮した場合

顔全体：顔全体の 16×16 の全局所領域得られた 5120 次元の特徴を 100 次元に圧縮した場合

図 36 に概要図を示す.

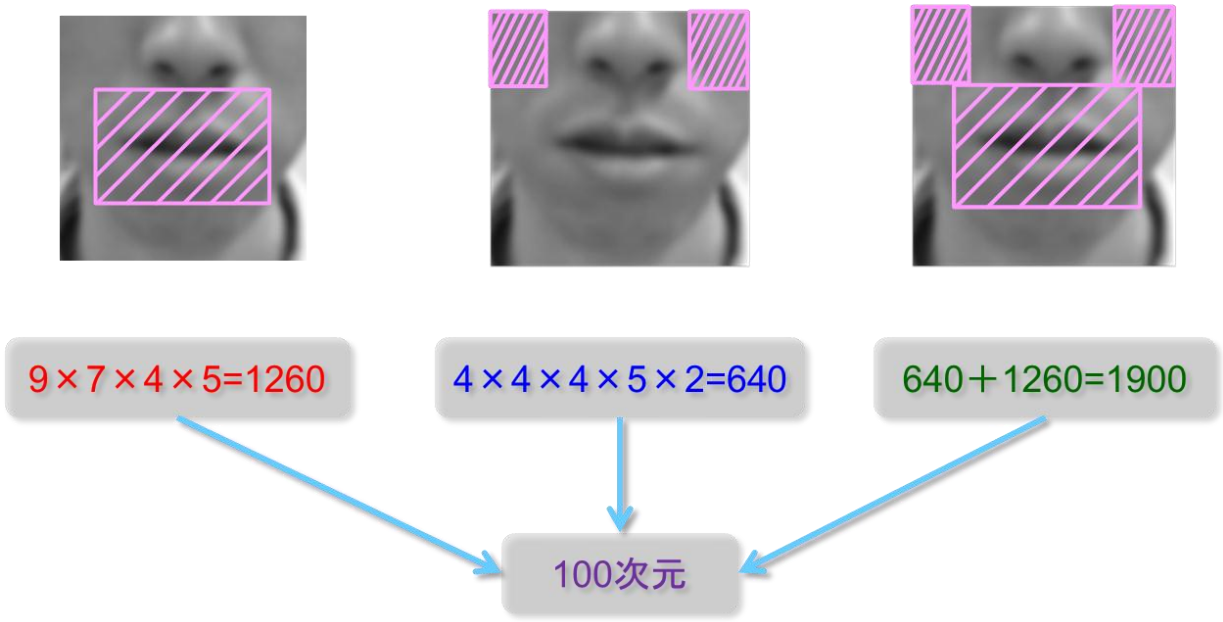


図 36　各注目領域の設定(左(a)：口周辺，真ん中(b)：頬領域，右：(c)口と頬)

実験結果を図 37 に示す．平均識別率は口周辺が 90．8%，頬領域が 29．2%，口と頬が 40.

8%であった.

口周辺(A)に注目した場合，顔の正規化により空間的変動を抑制し，且つ，発話区間，時間分割を行ったことで，識別に有効な時間的特徴を取得できたため，高い識別性能を得ることができたと考えられる．頬を含む領域(B)に注目した場合は良好な結果が得られなかったことがわかる．これは，頬部分だけでは各単語を分類する特徴を取得できなかったと考えられる．また，口周辺と頬領域両方に注目した領域(C)では，どちらの特徴も含まれているにも関わらず，識別性能の向上は見られなかった.

表 1 に口周辺と口と頬領域のそれぞれに注目した場合の混同行列を示す．行の単語名は正解単語を表し，列の単語は認識結果を，数値は識別率(%)を表している．口周辺から特徴を抽出した場合，識別失敗数はさほど多くないことが分かる．口と頬領域に注目した場合，頬領域に注目した場合に比べ，識別性能の向上は見られるが，不正解単語へ識別されてしまった単語もあった.

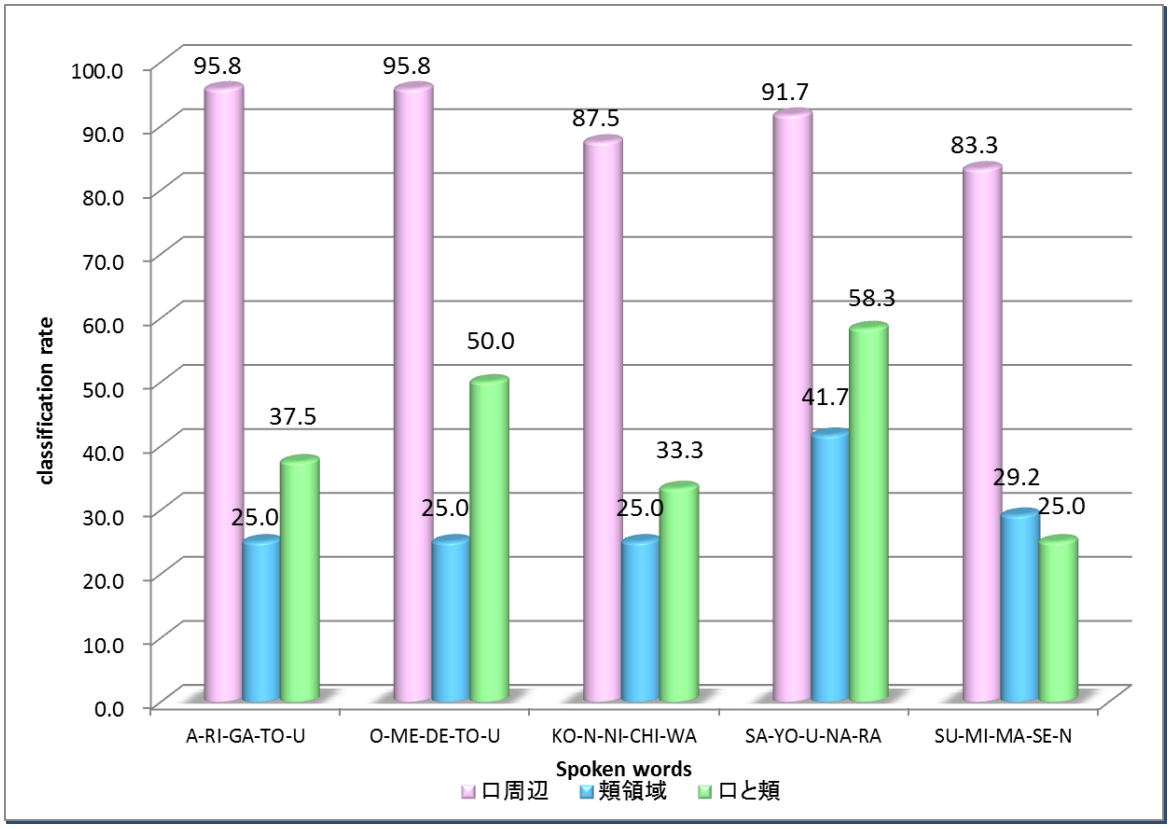


図 37　各注目領域の識別結果

4.5　部分空間を用いた識別性能評価

口周辺の領域画像を用いて部分空間を作成，Optical Flow 特徴を用いた手法と比較を行った．

この実験では第二期の動画を使用した．
そのため，図 37 の結果と比較できないため，もう一度 Optical Flow 特徴を取得し，識別結果を比較した．

比較する手法は以下の通りである．

(a)Optical Flow 特徴を Fisher の判別基準で学習したもの(OpticalFlow+Fisher)
(b)速度差の大きい上位 10 フレームに部分空間法を用いて学習したもの(SubSpace)
(c)再サンプリングを行った後，部分空間法を用いて学習したもの(ReSample+SubSpace)

図 38 に識別結果を示す．
平均識別率は(a)，(b)，(c)，それぞれ 43．5%，36．9%，23．8%となった．
識別結果を見ると，(a)の手法を用いた場合の識別率が高いことがわかる．また，部分空間を用いた結果(b)(c)では，部分空間によって作成された各単語の空間が類似してしまっていることが，識別率の低下につながってしまったと考えられるが，識別はすることが可能であると考えられる．再サンプリングを導入したもの(c)は，もともと存在しない画像を作成しているため，被験者ごとの分散が大きくなってしまったものと考えられる．これを解決するには，部分空間を作成する際，または識別をする際に，各単語のサンプルに共通するパターンとの差分を算出するか，各単語のサンプルに重みづけする必要があると考えらえる．
表 2 に混同行列を示す．行の単語名は正解単語を表し，列の単語は認識結果を，数値は識別率(%)を表している．(b)，(c)の表を見ると，「おめでとう」と「はい」への誤識別が多くなっていることが分かる．取得した発話ピークフレーム数が少なかったため，単語自体の特徴が表れにくかったことと考えられる．

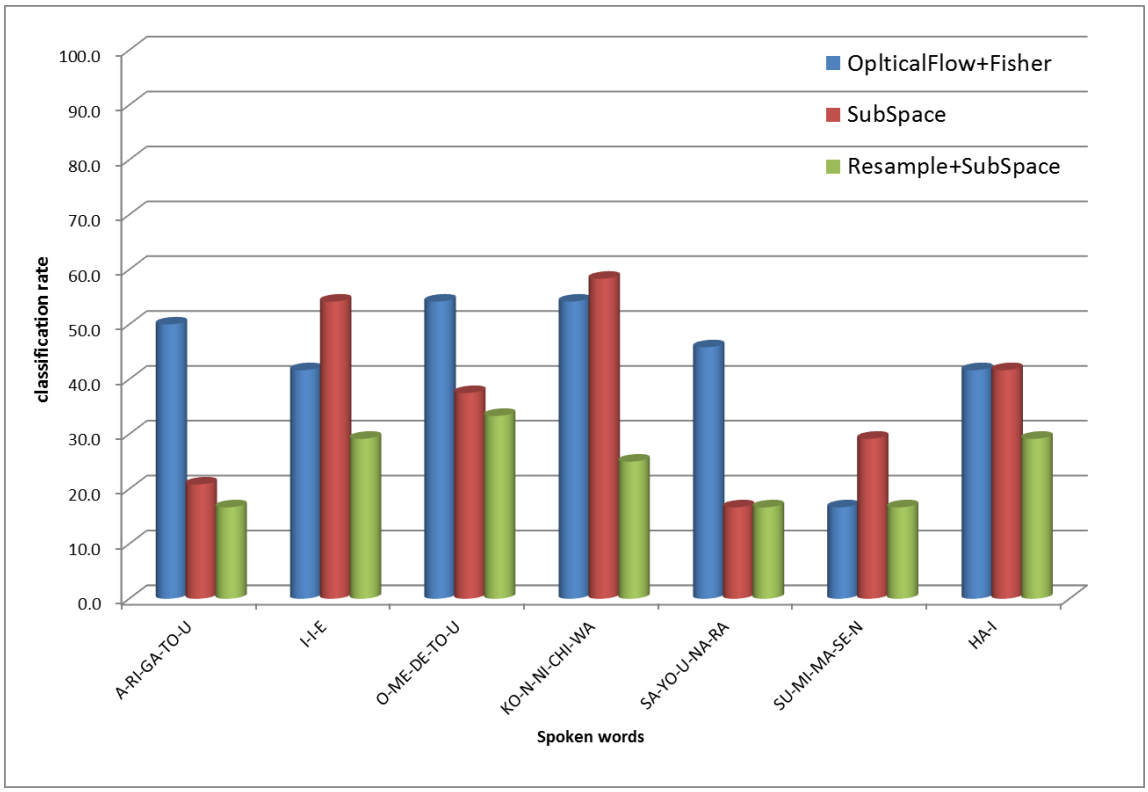


図 38　部分空間法を用いた識別結果

表 2　各手法での混同行列

(a) OpticalFlow+Fisher)

		recognized word						(%)
		A-RI-GA-TO-U	I-I-E	O-ME-DE-TO-U	KO-N-NI-CHI-WA	SA-YO-U-NA-RA	SU-MI-MA-SE-N	HA-I
testing word	A-RI-GA-TO-U	50.0	16.7	0.0	0.0	4.2	4.2	12.5
	I-I-E	4.2	41.7	8.3	4.2	4.2	8.3	4.2
	O-ME-DE-TO-U	4.2	0.0	54.2	0.0	0.0	0.0	4.2
	KO-N-NI-CHI-WA	0.0	4.2	4.2	54.2	4.2	4.2	0.0
	SA-YO-U-NA-RA	8.3	12.5	4.2	0.0	45.8	12.5	0.0
	SU-MI-MA-SE-N	4.2	8.3	16.7	4.2	12.5	16.7	4.2
	HA-I	12.5	0.0	8.3	0.0	0.0	4.2	41.7

(b) SubSpace

		recognized word						(%)
		A-RI-GA-TO-U	I-I-E	O-ME-DE-TO-U	KO-N-NI-CHI-WA	SA-YO-U-NA-RA	SU-MI-MA-SE-N	HA-I
testing word	A-RI-GA-TO-U	20.8	41.7	29.2	0.0	0.0	0.0	8.3
	I-I-E	0.0	54.2	20.8	8.3	4.2	8.3	4.2
	O-ME-DE-TO-U	12.5	20.8	37.5	0.0	0.0	0.0	29.2
	KO-N-NI-CHI-WA	8.3	4.2	16.7	58.3	0.0	0.0	12.5
	SA-YO-U-NA-RA	4.2	33.3	29.2	4.2	16.7	8.3	4.2
	SU-MI-MA-SE-N	0.0	4.2	20.8	0.0	12.5	29.2	33.3
	HA-I	0.0	20.8	33.3	0.0	0.0	4.2	41.7

(c) ReSample+SubSpace

		recognized word						(%)
		A-RI-GA-TO-U	I-I-E	O-ME-DE-TO-U	KO-N-NI-CHI-WA	SA-YO-U-NA-RA	SU-MI-MA-SE-N	HA-I
testing word	A-RI-GA-TO-U	16.7	16.7	33.3	16.7	0.0	0.0	16.7
	I-I-E	0.0	29.2	37.5	0.0	0.0	0.0	33.3
	O-ME-DE-TO-U	16.7	12.5	33.3	0.0	0.0	0.0	37.5
	KO-N-NI-CHI-WA	0.0	16.7	25.0	25.0	0.0	0.0	33.3
	SA-YO-U-NA-RA	0.0	25.0	29.2	4.2	16.7	4.2	20.8
	SU-MI-MA-SE-N	0.0	25.0	33.3	0.0	0.0	16.7	25.0
	HA-I	0.0	12.5	37.5	16.7	4.2	0.0	29.2

第5章　おわりに

5.1　まとめ

提案した発話区間の抽出では口周辺の局所領域を用いたことで，顔の動きと口の動きを分けることができたため，精度の高い発話区間の抽出ができた．　この処理を行わないと，人がいつ発話したかわからないため，識別することが容易でない．視覚情報を用いた識別手法と音声認識とを統合させるためにも，この発話区間の抽出が必要であり，今回の提案では高い抽出精度を得ることができたので，良い結果と言えるだろう．また，人の発話区間には音声発話区間だけでなく，その前後の準備区間があることが確認された．そのため，この音声発話区間のみの発話特徴を用いた識別手法の導入が可能になったことが分かる．

また，顔の各注目領域の識別性能は口周辺に注目したものが最も良いということが分かった．表情認識で有効とされている頬領域では，本研究において良好な結果は得られなかった．また，口周辺外の識別性能が口周辺の識別性能より低いことから，発声単語の特徴は口周辺の領域に多く含まれていることも確認できた．当たり前のことであるかもしれないが，こういった比較した論文は少ないため，実験的に立証することで，読唇の研究をする上で，より唇の動きに焦点をあてた方がよいことが示唆できる．

現状の状態では，OticalFlow を用いると識別性能が良いと分かるが，部分空間を用いるよりも特徴量が多いため，次元の呪い問題が起きやすくなる．また，部分空間法では主成分分析を行っているため，数少ないサンプルで発話単語クラスの形成することができるが，Fisher の判別基準で学習を行う場合，７クラスで分類を行うとすると，必要なサンプル数が膨大になってしまうデメリットもある．

部分空間法を用いた手法でも識別可能ということもわかった．新たに導入した再サンプリングと部分空間を用いた手法では，良好な結果を得ることができなかった．部分空間を用いた識別の先行研究では，時系列画像を１つの特徴ベクトルとせず，時系列画像を１つの空間として，空間と空間の類似度を測る手法があげられているが，これには多くのサンプルを用いてしまうため，個人に依存しない発話認識には有効ではない．各単語のサンプルに共通するパターンとの差分を算出し，より単語の違いを表す特徴ベクトルの作成方法を今後，検討しなければならないことが分かった．

第6章　謝辞

本研究を進めるにあたり，ご指導ならびに貴重なご意見を頂いた，法政大学赤松茂教授に心より感謝致します.

また，日頃から常々お世話になり，プログラムやアルゴリズムに関して相談にのって頂きました法政大学博士課程 1 年生稲葉善典氏はじめ，研究に協力していただいた赤松研究室の皆様に深く感謝致します.

参考文献

[1] 内閣府編，”高齢社会白書”，2008.

[2] 金山和功，白井良明，島田伸敬，”HMM を用いた手話単語の認識”，信学技，Vol．104，No．89，PRMU2004-16，pp． 21-28，May，2004.

[3] 藤野雄一，伊福部達，”マルチメディア時代における医療・福祉支援技術”，電子情報学会誌，Vol．80，No．8，pp822-829，Aug.，1997.

[4] 赤松茂，”コンピュータによる顔の認識-サーベイ”，信学論(D-2)，Vol．J80-D-2，no．8，pp．2031-2046，Aug.，1997.

[5] 有木康雄，駒井祐人，”画像と音声情報を統合した発話認識”，情報処理学会誌，Vol．52 No．1，pp．87-94，Jan．2011

[6] 齊藤剛史，森下和敏，小西亮介，” 複数口唇領域を利用した多言語に有効な単語読唇”，信学技，PRMU2008-76，pp．181-186，Sep.，2008

[7] 森下和敏，齊藤剛史，小西亮介，”発話シーンからのキーフレーム検出と単語読唇”，MIRU2009， IS1-58，pp．753-760，Jul.，2009

[8] 中田康之，安藤護俊，” 色抽出法と Eigentemplate 法を併用した口の位置検出と読唇処理への適用”，信学技，PRMU2001-95，pp．7-12，Sep,．2001

[9] 西山正志, 山口修, 福井和広, ”制約相互部分空間法を用いたジェスチャー認識, ” SSII04, Vol10，pp．439-444，2004

[10] 清田公保，内村圭一，”口唇周辺画像情報を用いた発話単語認識”，信学論，Vol．J76-D-Ⅱ，No．3，pp．813-814，Mar.，1993.

[11] 間瀬健二，アレックス・ペントランド，”オプティカルフローを用いた読唇”，信学論，J73-D-2，6，pp．796-803，Jun.，1990.

[12] 間瀬健二，前田英作，末永康仁，”表情動画像からの感情の認識の 1 手法”，信学技，PRU91-24，pp．95-101，1991.

[13] 中島 慶， 藤江 真也， 松坂 要佐， 小林 哲則，” 対話調整的役割を果たす顔表情の認識”，信学技，PRMU2005-105，pp．7-12，Oct.，2005

[14] R. Nakamura，S. Akamatsu，“Recognition of Spoken Words Using Motion Features Extracted from Video Sequences:Comparison of Recognition Performance Dependent on Attention Area of Face”， APSIPA， Dec.， 2010

[15] ATR-Promotions Ltd.， "accFaceTN SDK"， http://www．atr-p．com/af．html

[16] “TEO ライブラリによる画像処理プログラミングガイド”， http://teo．sourceforge．jp/doc/TeoProgrammingGuide/section5-2．html

[17] 川戸慎二郎, 鉄谷信二, ” SSR フィルターと SVM を用いた顔の実時間検出と追跡, ” 電子情報通信学会技術研究報告． HIP， ヒューマン情報処理 103(454)， 47-52， Nov. 2003

[18] Lucas．B and Kanade．T， “An Iterative Image Registration Technique with an Application to Stereo Vision”， Proc． of 7th International Joint Conference on ArtificialIntelligence (IJCAI)，pp． 674-679， 1981.

[19] 石井健一郎，前田英作，上田修巧，村瀬洋，”わかりやすいパターン認識”，オーム社，1998.

[20] 藤川祐介，諸岡健一，長橋宏，”高次元特徴の圧縮に基づく多クラスパターンの認識”，信学技，Vol．103，No．737，pp．97-102，PRMU2003-272，Mar.，2004

[21] ”OpenCV リファレンスマニュアル”，http://opencv．jp/opencv-1．0．0/document/

[22] 土井滋貴, ”はじめての動画処理プログラミング Win32API と DirectX で実装する動画処

理の基礎”，CQ 出版，2007.
[23] “M. Oshikiri HomePage Tips 集”， http://www. geocities. co. jp/SiliconValley/7406/tips/index. html
[24] “AviUtl のお部屋” http://spring-fragrance．mints．ne．jp/aviutl/
[25] “AviUtl 実験室”http://www．tenchi．ne．jp/~yoko/aviutl/

付録

顔の意図的学習と印象判断時における視線の動きの比較
－停留領域と停留時間の分析－

指導教授　　赤松　茂

2010 年度　法政大学大学院
工学研究科システム工学専攻
09R6116
中村　亮太

目次

図表目次

あらまし

我々は，人が顔を意図的に学習しようとしている場合，例えば「有能そうかどうか」といった印象を判断している場合，そして，それらの後に「見たことのある顔かどうか」の再認を行っている場合について，それぞれの眼球運動を測定した..眼球運動の計測には SR Research 社の急速眼球運動解析装置 EyeLink CL を用いた.また, 刺激の提示と行動データ(印象判断または再認判断)の取得を行うシステムは同社の Experiment Builder を用いて構築した.得られた視線の停留点の位置(x 座標または y 座標の顔の中心からの距離)とその停留時間を基に，提示された顔画像の各領域の累積停留時間ヒストグラムの作成を行い，顔画像観察条件における分析を行った.その結果，意図的学習時と印象判断時ではほぼ同じ領域を見ているが，再認時では異なる場所を見ていることが分かった.

キーワード　視線解析，意図的学習，印象判断，累積停留時間ヒストグラム

Abstract

This paper describes the investigation of eye-movement during two different tasks, intentional learning and impression judgment of faces. Rapid eye movement measurement system called EyeLink CL, a commercial product of SR Research Ltd., was used for the measurement of eye–movement and Experiment Builder produced by the same firm was used to show some visual stimuli and to obtain and impression judgments recognition judgments. We analyzed different strategies of visual information processing by creating the cumulative histograms of fixation time based on the position of fixation point and its fixation time. As a result, the almost same areas were gazed during two different tasks but the different areas were gazed during recognition.

Keyword Analysis of eye-movement, Intentional Learning, Impression Judgment, Cumulative Histogram of fixation time

1. はじめに

認知心理学分野において顔の学習，印象判断について論じた幾つかの先行研究がある．

Hsiao & Cottrell (2008) によれば，顔認知においては，2 つの停留点（視線が一定時間留まるポイント）があれば十分再認が可能であり，それ以上増えても再認成績は変わらないことが，眼球運動のデータより示されている． また，学習時と再認時では停留点位置が異なっていたことから，顔を学習する際と思い出す際とでは異なる方略が取られている可能性が示唆された[1]．しかし，印象判断時の眼球運動について論じている論文は少ない．

Bower & Karlin (1974)によれば，物理的特徴に関する判断を行うことにより顔を意図的に学習した場合に比べ，印象判断を行った場合のほうがより記憶成績が高くなる現象が確認されている[2]．しかし，意図的学習時と印象判断時での眼球運動の違いを比較した研究は少ない．

また，Willis & Todorov(2005)により，「魅力的」「信頼できる」などの項目についてそれぞれ印象判断を行う場合，0．1 秒あれば判断が可能であり，それ以上の時間をかけた場合には，判断結果への確信度は増加するが，判断結果自体にはほとんど変わらないことが示されている[3]．したがって，印象判断時においても，必要な停留点の数は多くないことが考えられる．

中村らは，顔の意図的学習時と印象判断時での眼球運動を計測し，停留点の位置(刺激画像の中心からのピクセル差)と停留点の順序(第 1 停留点から第 3 停留点)に分散分析を適用した結果，顔画像観察条件間の有意な差は見られないことを示した．だが，課題の種類によらず，停留点の位置は徐々に左側から右側に寄っていく傾向があること，意図的学習後と印象判断後の再認時の停留点の順序に有意な傾向が見られることを明かした[4].

先行研究の多くは，刺激顔画像の停留点の位置と停留時間について別々に分析を行っている．そこで本研究は，上記の研究知見に基づき，顔の意図的学習時と印象判断時，および，その後の「見たことのある顔かどうか」の再認時での眼球運動を計測し，視線の停留点の位置とその停留時間を基に，提示された顔画像の各領域の「累積停留時間ヒストグラム」を作成し，比較することにした．

2. 実験方法

2.1　実験環境

本実験における眼球運動の計測には, 米国 SR Research 社の急速眼球運動解析装置 EyeLink CL[5]を用いて行った．EyeLink CL は片眼計測であり，第一期実験ではすべて左目で計測した．第二期実験ではデータの精度を上げるために左目と右目の眼球運動を被験者ごとに，交互に測定した．実験プログラムの作成には，付属ソフトである視覚刺激作成支援ソフト Experiment Builder を用いて作成し，測定結果の解析には，付属ソフトである注視点解析ソフト Data Viewer を用いて行った．

先行研究に沿った形で実験を行っていくため，あご台を用意して被験者の顔の位置を固定して測定を行った．（図 1 図 2 に示す）また背景などの視覚的要因をなくため，暗幕を用いた．視距離は各期の実験において変更した．

図 1　実験環境内観

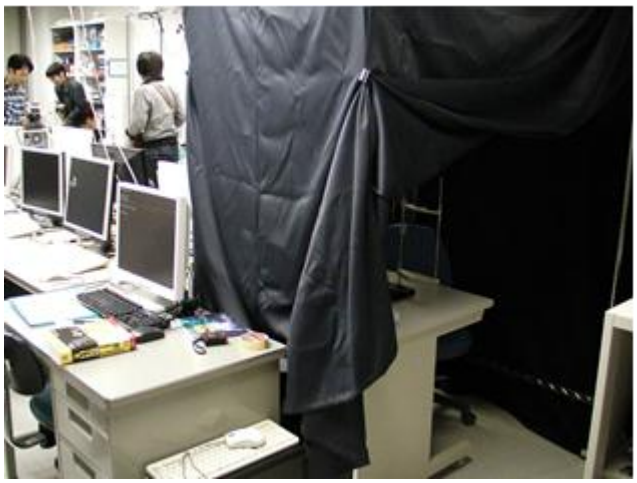


図 2　実験環境外観

2.2 Eyelink CL

Eyelink CL（図 3 に示す）は非接触にて計測できるビデオ式のアイトラッキングシステムである．非接触で計測できるため，被験者への負担は軽減される．また，シールを同時にモニタリングし，まばたき後のデータ復旧のスピードのほか，頭部運動によりアイカメラから眼球映像が外れた場合でもデータ損失を最小限に抑えて計測できる．

本来シールは額に貼り付けるものだが，第一期実験では，額付きのあご台を用いるため額をモニタリングすることができない．よってシールはあご台に貼り付けた．

第二期実験では顎台の額当てを取り払ったので，額にシールを貼ることができた．頭部の運動にも対応し，安定した追従が可能になった．

そして，キャリブレーションには EyeLink 独自の注視状況評価技術が組み込まれており，短時間でおこない精度の高いデータを取り込むことができる．

図 3　Eyelink CL

EyeLink CL のシステムは Display PC と Host PC の２つで構成されている．

Display PC は実験の作成，実行，解析を行う．実験の作成には視覚刺激作成支援ソフト Experiment Builder を用いることで，視覚刺激を作成する．Experiment Builder は GUI によりアイコン同士をワークスペース上でつなぐだけで視覚刺激は容易に完成することができる．また，データは EDF 形式で保存され，注視点解析ソフト DataViewer で解析することができる．DataViewer は記録したデータの動画再生や，マッピング機能によって視覚的に視線の情報の確認をすることができる．

Host PC はリアルタイムで目の追跡を実行し，注視やサッケードを検出する．

また，Display PC と Host PC は Ethernet を通してリアルタイムで情報交換を行うことができる．（図 4 に示す）

図 4　モニタリングの映像

EyeLink CL のシステム構成を図 5 に示す.

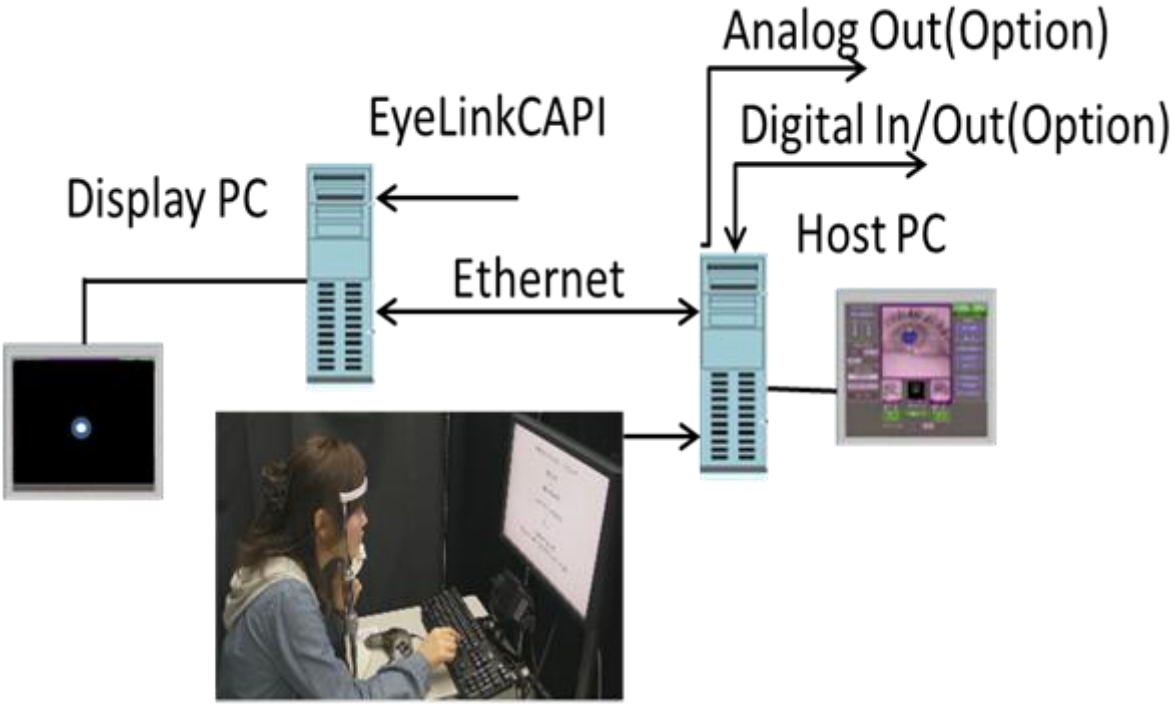


図 5　システム構成

2.3 Experiment Builder

視覚刺激作成支援ソフト Experiment Builder[6]は Eyelink CL のオプションソフトであり，視覚刺激の作成に用いる．Experiment Builder は図 6 に示すように GUI によりアイコン同士をワークスペース上でつなぎ視覚刺激を作成するため容易に作成することができる．また，EyeLink CAPI により HostPC とシンクロすることで，キャリブレーションから計測開始，終了まで自由に必要なアイコンをつなぐだけで完成させることができる．データは EDF 形式で保存され，付属ソフト DataViewer で解析していく．

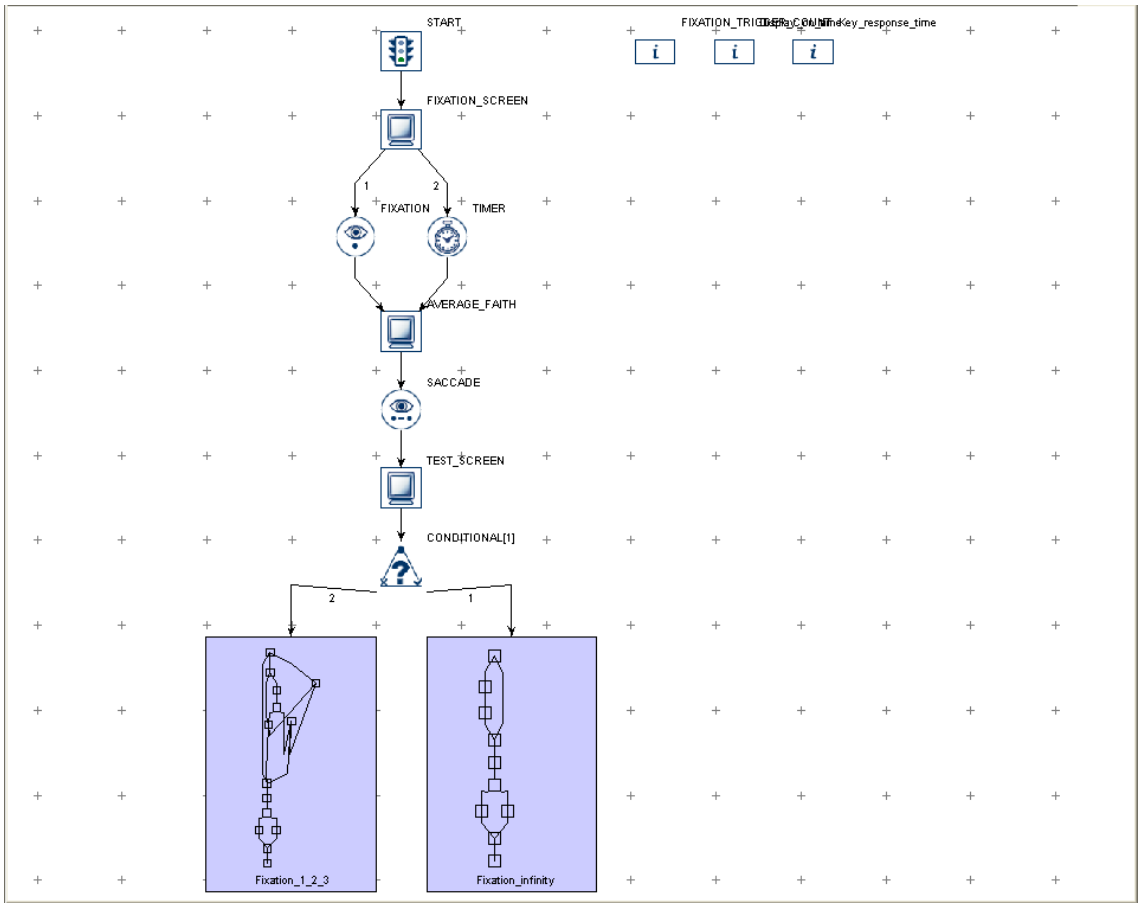


図 6　Experiment Builder で作成された視覚刺激の例

2.4 Data Viewer

注視点解析ソフト Data Viewer は記録した眼球運動を解析するソフトウェアである．図 7 に示すように視覚刺激画像に対する注視点やサッケードを表示し，記録したデータの動画再生や視線のマッピング機能により視覚的に視線の情報を確認することができる．

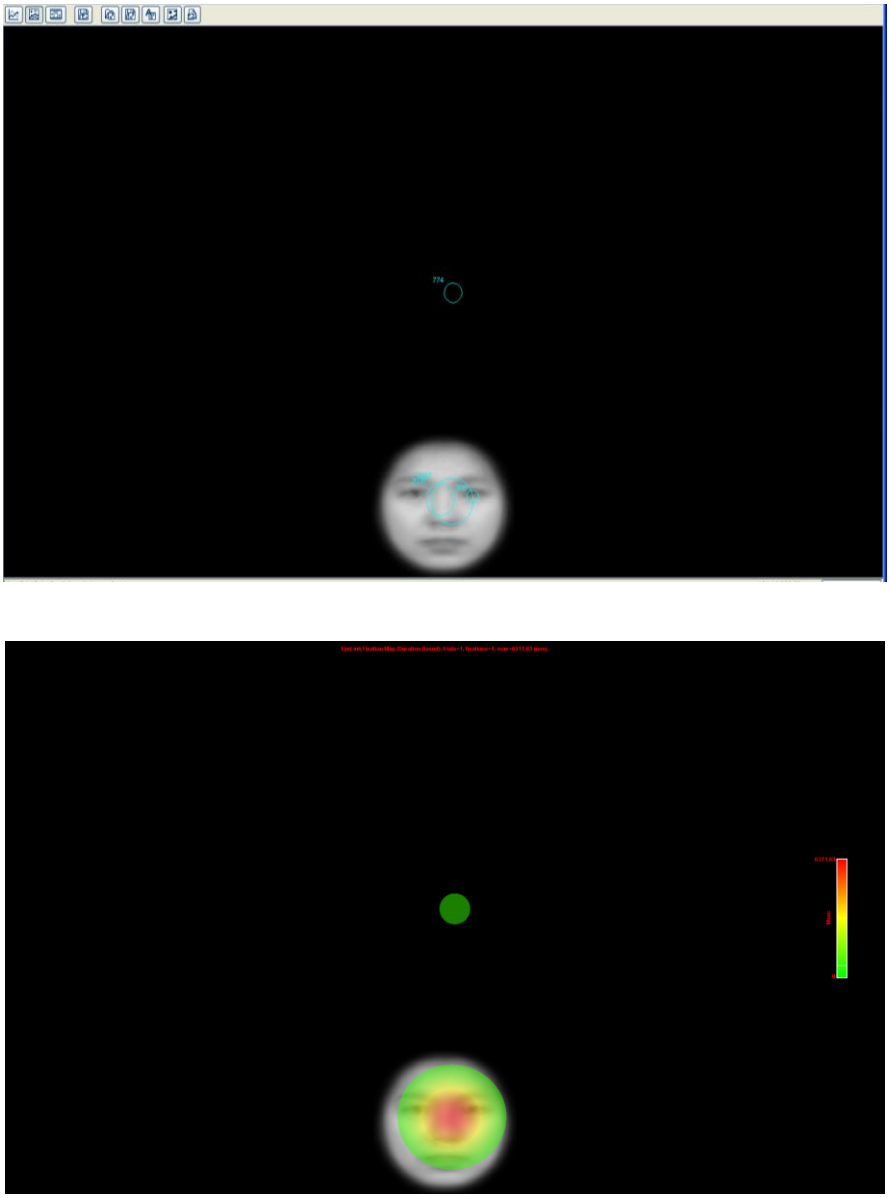


図 7　Data Viewer における結果表示の例 1・2

2.5　視覚刺激について

刺激は，図 8 の手順で正規化し，図 9 の手順で加工された，男性 32 枚，女性 32 枚の計 64 枚の顔画像を用いた. サイズは会話をしている際の対人距離(1m)を想定して, 8. 2cm×8. 2cm(312×312pixel)に設定した.

顔画像に対しての顔画像合成システム FUTON(株式会社 ATR-Promotions 社) [7]を用いて顔の位置，傾き，大きさを正規化するとともに，顔の輪郭ぎりぎりのところで楕円形に切り出し，楕円の縁はぼかしをいれた．顔の左右差の影響を排除するため，左右反転した画像も作成した． よって, 本実験では刺激画像は全てで 130 枚(＝(64 枚＋1 枚)×2 種類)用いた.

FUTON（Foolproof Utilities for Facial Image Manipulation system）とは,（株）ATR 人間情報通信研究所で開発された，顔に関する研究の心理実験で用いる顔刺激画像を容易に作成することを主な目的としたソフトウェア群のことである.

更に，全ての顔画像を用いて，平均顔を作成し，実験で使用するマスク画像とした.
視覚刺激の一部を図 10，平均顔を図 11，反転加工例を図 12 に示す.

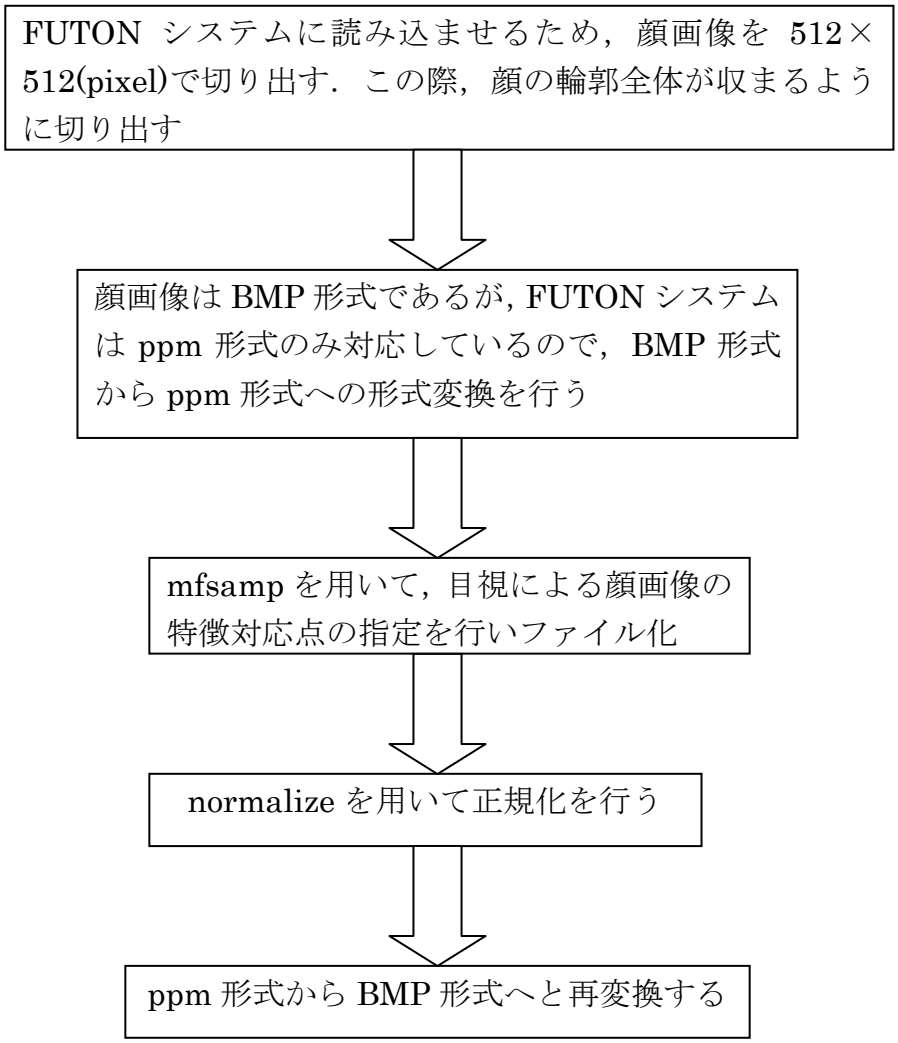

FUTON システムに読み込ませるため，顔画像を 512×512(pixel)で切り出す．この際，顔の輪郭全体が収まるように切り出す
顔画像は BMP 形式であるが, FUTON システムは ppm 形式のみ対応しているので，BMP 形式から ppm 形式への形式変換を行う
mfsamp を用いて，目視による顔画像の特徴対応点の指定を行いファイル化
normalize を用いて正規化を行う
ppm 形式から BMP 形式へと再変換する

図 8　正規化の手順

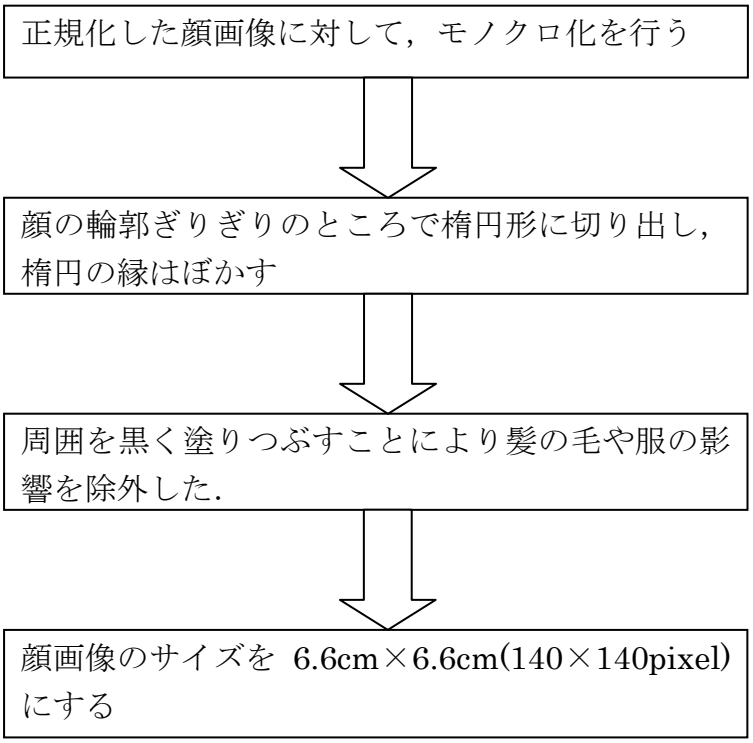


図 9　正規化後の顔画像の加工手順

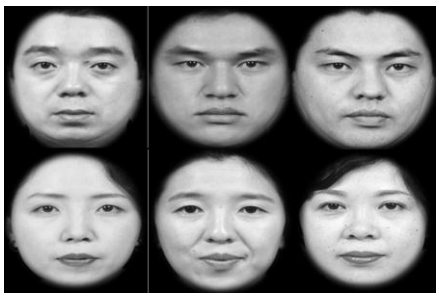


図 10　視覚刺激に用いた画像

図 11　マスクに用いた平均顔

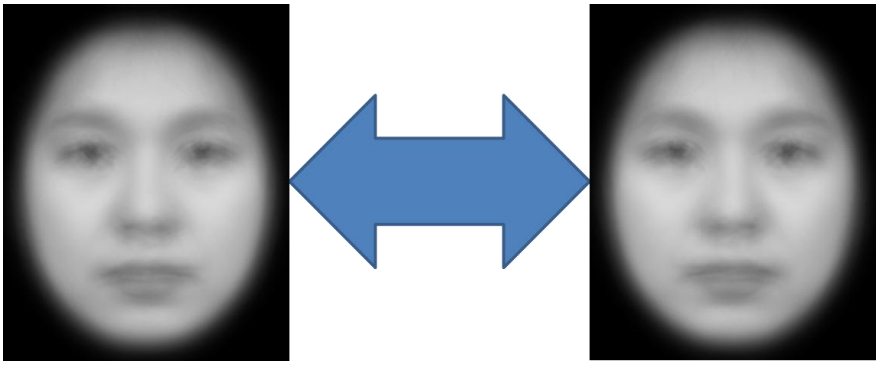


図 **12**　平均顔画像の反転加工

2.6　実験全体の流れ

実験全体の流れを図 13　実験全体の流れに示す. 本実験は,3 つの段階で構成されている. 3 つの段階それぞれについて以下に記述する.

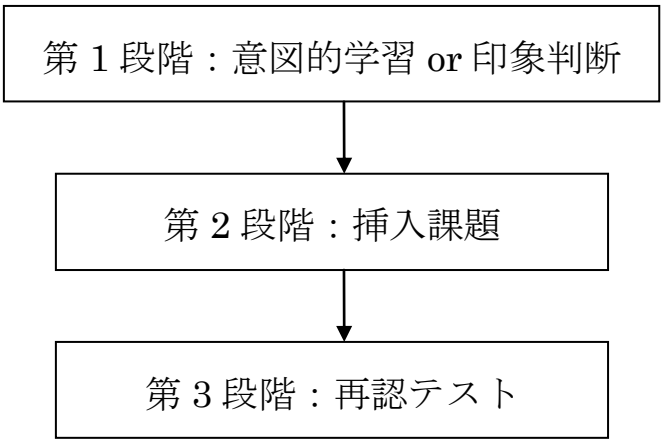


図 13　実験全体の流れ

第 1 段階：意図的学習 or 印象判断

32 個の顔画像（正画像と反転画像を含む 64 個の顔画像の中から,正と反転を同じ比率になるようにランダムに選んだ 32 個）をターゲット画像として提示し,半数の実験参加者は意図的学習,もう半数の参加者は印象判断を行った．被験者は意図的学習の場合,3 秒間顔画像を提示して記憶し「顔が覚えやすいかどうか」判断し,印象判断の場合,各顔画像に対して「有能さ」について 4 段階で評価した.

第 2 段階：挿入課題

再認成績の天井効果を防ぐため,挿入課題として 20 分間の単純視覚課題を行った．単純視覚課題は,「ウォーリーを探せ」[8]を行ってもらった.

第 3 段階：テスト

64 個の顔画像を提示し,第 1 段階において見たことがあるかを判断した．ここで提示する顔画像は,ターゲット画像 32 個とディストラクタ画像（意図的学習 or 印象判断で使われなかった画像）32 個を用いた.

意図的学習時,印象判断時,テスト時にそれぞれ眼球運動の計測を行った.

2.7　眼球運動の計測

３．１．３．１　眼球運動計測の流れ

眼球運動の計測は図 14　眼球運動計測の流れの流れで行った.

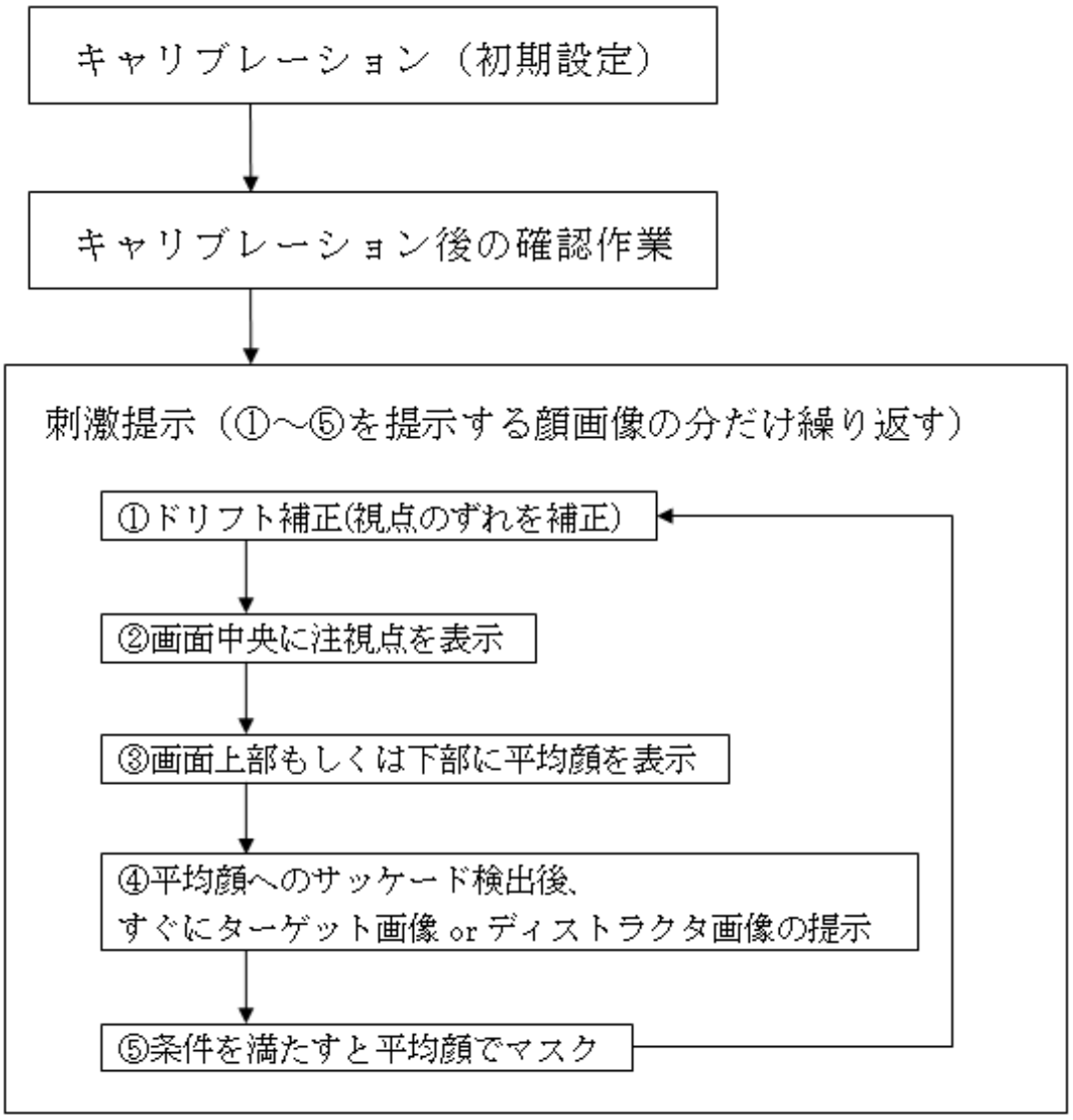


図 14　眼球運動計測の流れ

また,刺激提示の流れ(③から⑤まで)を図 5 に示す.

⑤の平均顔でマスクをする条件は,意図的学習,印象判断,再認テスト時で異なる条件を設定した.

意図的学習：ターゲット画像を表示して 3 秒経過した時

印象判断：印象判断後またはターゲット画像を表示して 3 秒経過した時

再認テスト：停留点の数が指定された値(1 個, 2 個, 3 個)を越えた時,または,ターゲット画像を表示して 3 秒経過した時

また,意図的学習時での「顔が覚えやすいかどうか」の判断と,印象判断時での「有能さ」の評価には,コントローラーのボタンを押してもらうことで検出を行った.

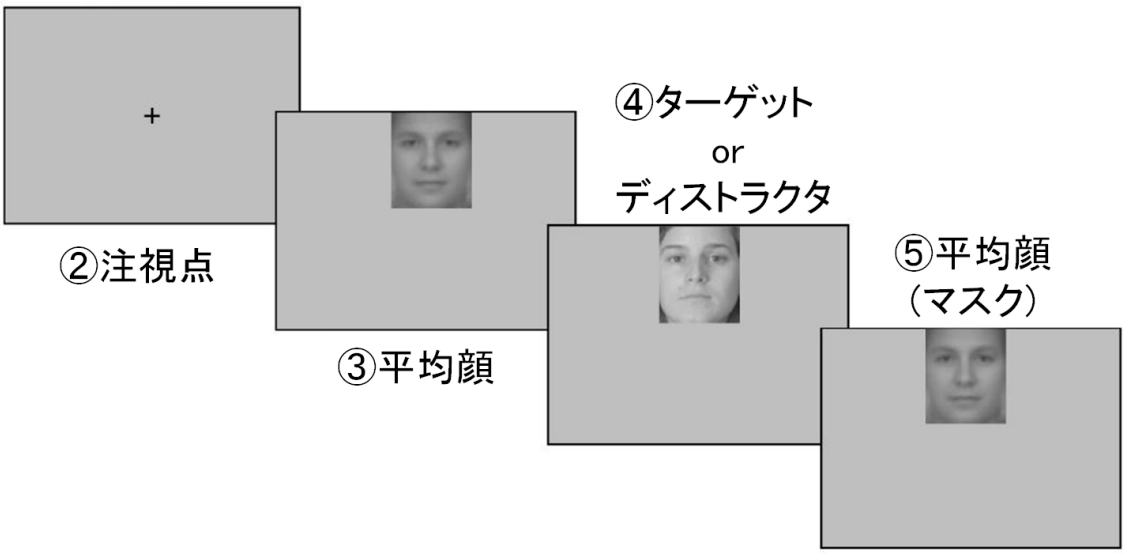


図 15　刺激提示の流れ

2.8　被験者について

2.5 で作成した視覚刺激を用いて,大学生および大学院生 40 名に対して行った(平均年齢 21．3 歳,標準偏差 0．96)．また,EyeLink CL は片眼計測であるので,半数の被験者は左目,もう半数は右目の眼球運動を計測した.

2.9　キャリブレーション確認作業

EyeLinkCL で計測したデータの信頼性を向上させるためにキャリブレーションの後に，確認作業を行った．

ディスプレイの中心位置と，刺激画像が表示される上下の位置に小さな円を表示させ，被験者にはその円を見てもらいながらボタンを押してもらった．

そのデータを元に，測定データの補正を行った．

その補正の方法は以下のとおりである．

今回は，刺激提示ディスプレイの左上を座標の原点とし，X 座標は右，Y 座標は下を正とした．図 16 にディスプレイの座標を示す．

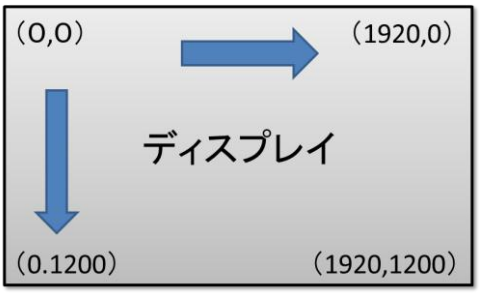


図 16　刺激提示ディスプレイの座標

今回は，ディスプレイに対して正しい位置を表 1　確認作業での正しい座標のように設定した．

表 1　確認作業での正しい座標

正しい位置	X 座標	Y 座標
中央	960	600
上	960	156
下	960	1044

そして，確認作業で得られたデータ X_m,Y_m から，正しい位置 X_c,Y_c を引いた値を誤差値 X_dY_d とした．式(1)に示す．

$$(X_d,Y_d)=(X_m,Y_m)-(X_c,Y_c) \qquad (1)$$

誤差値 X_d が正の場合は，正しい座標より右を見ているので，本実験の測定データから X_d を引いたものを，真の測定データとした．

誤差値 X_d が負の場合は，正しい座標より左を見ているので，本実験の測定データから X_d を引いたものを，真の測定データとした．

Y 軸は上を見ているときが負，下を見ているときが正なので，X 軸と同じように補正し，真の測定データを導いた．

2.10　分析方法

2.9.1　分散分析について

分散分析とは，観測データにおける変動を誤差変動と各要因およびそれらの交互作用による変動に分解することによって，要因および交互作用の効果を判定する，統計的仮説検定の一手法である[9].

今回は分散分析を行うために，web 上で分散分析を行うことのできる ANOVA４[10]を用いた.

ANOVA4 は，４要因までのデザインで，被験者間要因・被験者内要因のすべての組み合せについて分散分析を行うことができる．ｎ＝１のモデルにも対応.また分散分析のほか, 下位検定として, 主効果が有意な場合の多重比較(Ryan 法), 交互作用が有意な場合の単純効果の検定（３要因の交互作用まで)，さらに単純主効果・単純主効果が有意な場合の多重比較を行うことができる．[10]

2.9.2 停留領域と停留時間

得られた眼球運動のデータから各停留点の位置(x 座標または y 座標の刺激顔画像の中心からの距離)と,その停留時間を基に,累積停留時間ヒストグラムの作成を行った. 累積停留時間ヒストグラムとは,x,y 軸方向に停留点位置の座標,z 軸方向に停留時間の累積和を表した,3 次元空間分布ヒストグラムである. 今回は第 1 停留点,第 2 停留点,第 3 停留点,全ての停留点についてのヒストグラムの作成をした. 停留点位置の座標をそのままプロットするのではなく,刺激顔画像の縦横を 12 分割して 144 個の停留領域を作成し,その領域ごとに停留時間を累積し,長く注視している領域を比較した.

また,顔の左右の領域それぞれの総停留時間を算出し,その結果を比較した.

2.9.3 再認成績

意図的学習後の再認テスト時と印象判断後の再認テスト時の再認成績の比較を行う.
今回は再認成績の指標に Hit 率と FA 率,分別感度 A'の 3 種類を用いた. Hit 率とは第 1 段階で提示された顔画像を「見たことがある」と判断した確率,FA 率とは第 1 段階で提示されていない顔画像を「見たことがある」と判断した確率のことを言う. また,分別感度 A'は,(2)式によって算出した.

$$A' = \frac{1}{2} + \frac{(Hit - FA)(1 + Hit - FA)}{4 \times Hit \times (1 - FA)} \qquad (2)$$

算出した指標について画像観察条件(意図的学習後再認/印象判断後再認)×停留点条件(1/2/3/無制限)の 2 要因分散分析を行った.

3. 実験結果と考察

3.1　停留領域と停留時間について

意図的学習時の累積停留時間ヒストグラムを図 17　意図的学習の累積停留時間ヒストグラム,印象判断時の累積停留時間ヒストグラムを図 18　印象判断の累積停留時間ヒストグラム,意図的学習後再認テスト時の累積停留時間ヒストグラムを図 19　意図的学習後再認の累積停留時間ヒストグラム,印象判断後再認テスト時の累積停留時間ヒストグラムを図 20　印象判断後再認の累積停留時間ヒストグラムに示す．長く見ている程,赤くなるように設定をしている．プロットは図 11　マスクに用いた平均顔の平均顔に行った.

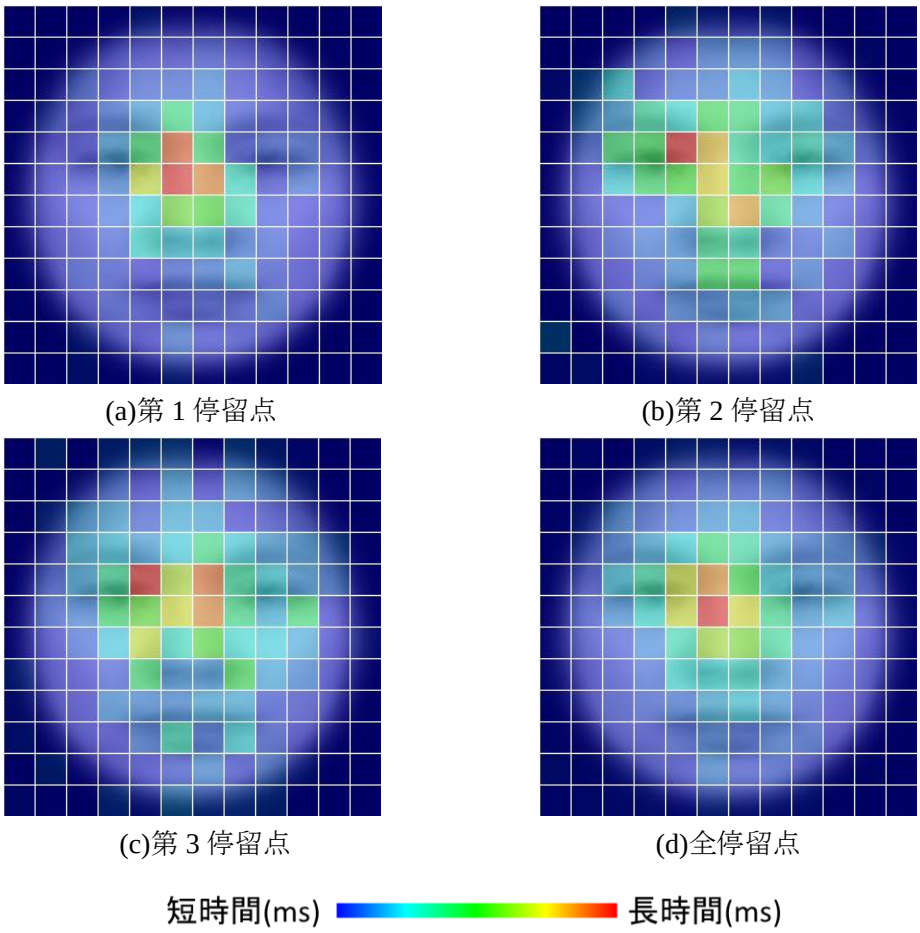
意図的学習(図 17)と印象判断(図 18)を比較すると,全停留点の累積停留時間ヒストグラムにおいて,顔画像観察条件における長く注視される領域に,違いは見られない．だが,図 19(d)と図 20(d)を比較すると,図 20(d)の方が,注視される領域が多いことから,印象判断時では,個人による主観や好みの差があるということが考えられる.

意図的学習(図 17)とその再認テスト(図 19)では,全停留点の累積停留時間ヒストグラムにおいて,長く注視される領域が異なることがわかる. これは先行研究[1]と同じ結果である. 一方,印象判断(図 18)とその再認テスト(図 20)では同じ様な部分を見ている.

意図的学習後の再認テスト(図 19)と印象判断後再認テスト(図 20)では全停留点の累積停留時間ヒストグラムにおいて,長く注視している領域に違いがみられた．意図的学習後の再認では顔画像の中心より右,印象判断後の再認では顔画像の中心を見ている傾向があった. この場合も,印象判断後の方が,分散が大きかった.

意図的学習,印象判断,意図的学習後再認,印象判断後再認の顔の左右それぞれの領域の総停留時間を集計し,一要因分散分析を行ったが,全て有意でないことが分かった．これは，各被験者の停留領域のはらつきが大きいためであると考えられる.

以上の結果から,意図的に顔を覚えることと印象を判断する際では,長く注視している部分は同じであり,注目している領域にも差異はないということが分かった.
また,意図的学習した後と印象判断した後の顔を思い出す作業では注目している領域の総停留時間について有意な差はないが,長く注視している部分が異なるため,再認時に異なる方略を取っていることも考えられる.



(a)第 1 停留点　(b)第 2 停留点

(c)第 3 停留点　(d)全停留点

図 17　意図的学習の累積停留時間ヒストグラム

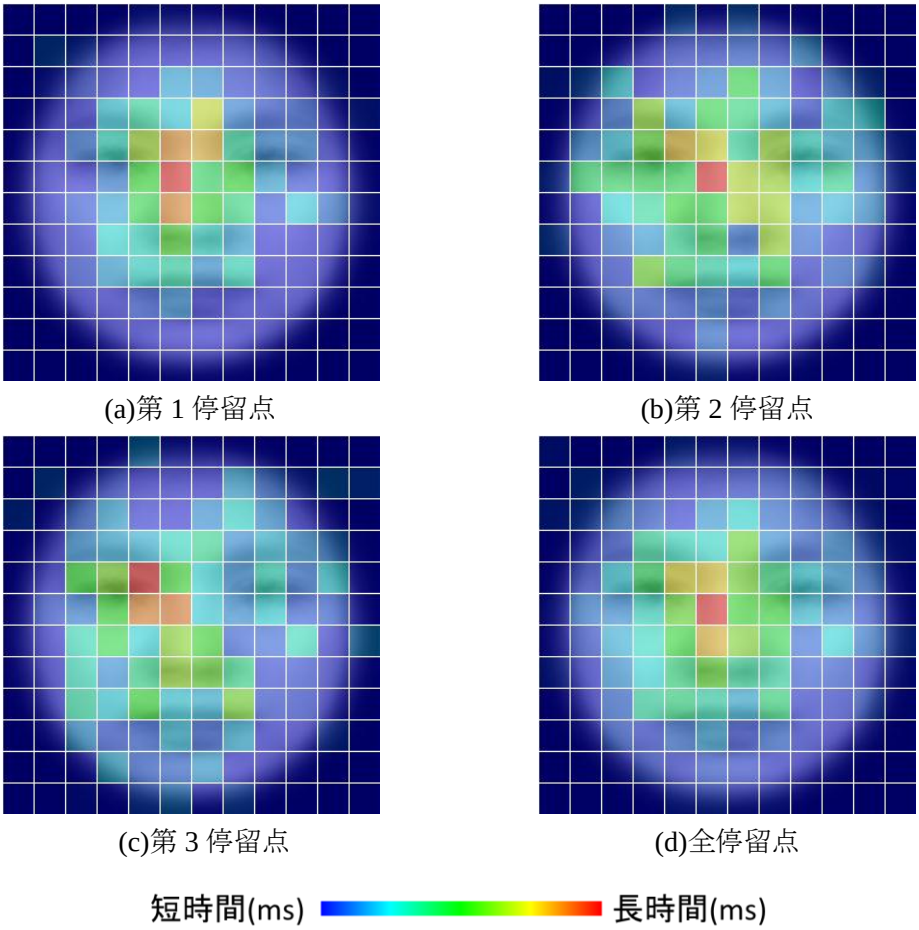


図 18　印象判断の累積停留時間ヒストグラム

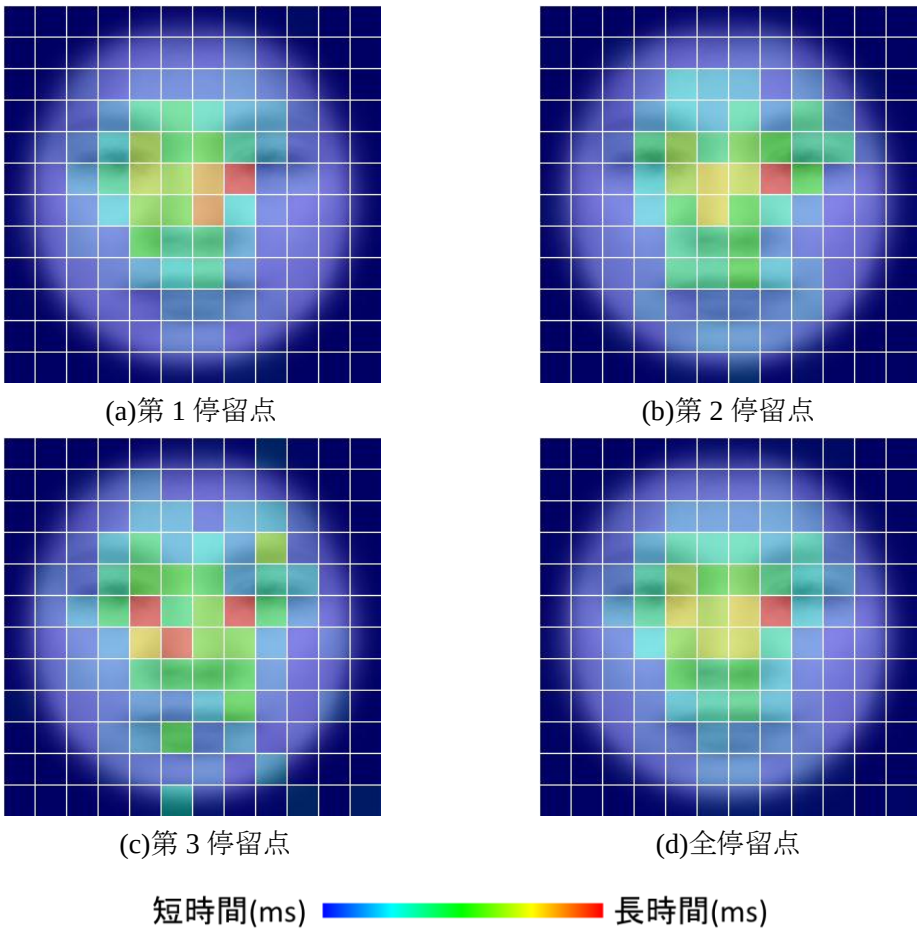
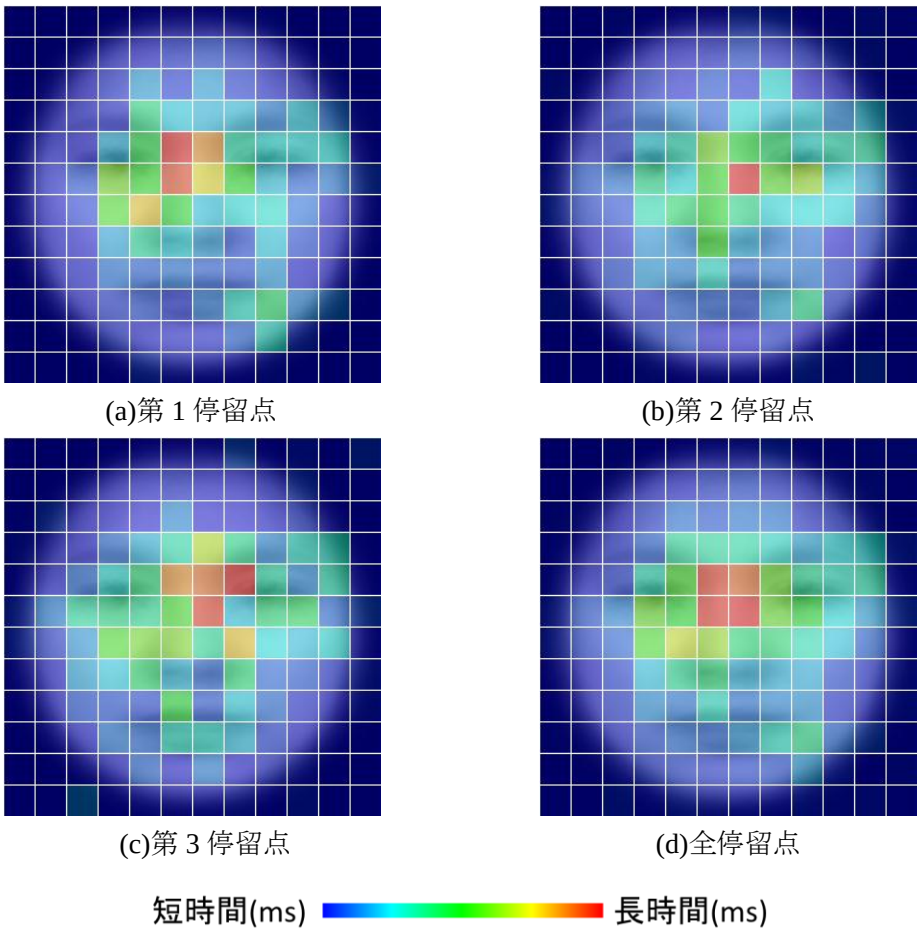


図 19　意図的学習後再認の累積停留時間ヒストグラム



(a)第 1 停留点　(b)第 2 停留点

(c)第 3 停留点　(d)全停留点

図 20　印象判断後再認の累積停留時間ヒストグラム

3.2　再認成績について

Hit 率について画像観察条件(意図的学習後再認/印象判断後再認)×停留点条件(1/2/3/無制限)の 2 要因分散分析を行った結果,有意ではなかった．FA 率について画像観察条件(意図的学習後再認/印象判断後再認)×停留点条件(1/2/3/無制限)の 2 要因分散分析を行った結果,停留点条件の主効果が有意であった(F(1,38)=2．532,p<．01)．Ryan's 法による多重比較の結果,停留点が 1 点と 3 点に制限された時の間に有意な差があった(p<．005).

分別感度 A’について画像観察条件(意図的学習後再認/印象判断後再認)×停留点条件(1/2/3/無制限)の 2 要因分散分析を行った結果,停留点の条件の主効果に有意な傾向が見られた(F(1,38)=2．449,p<．10)が,顔画像観察条件には有意な差は見られなかった図 21 に画像観察条件ごとの各停留点条件の分別感度 A’を示す図 21 より,顔画像観察条件によらず,停留点 1 点のみで 70%以上再認が可能であり,停留点が増えるにつれて,再認成績が良くなるというわけではないことが分かった．この結果は先行研究[1]と一致していた.

また,両者の分別感度の平均は,意図的学習後再認では 0．75,印象判断後再認では 0．70 となった．認成績がよいという,本研究の結果から,意図的に顔を覚えようという意思がなくても,顔の印象を判断しただけでも,意図的に覚えようとした場合と同等の再認成績が得られることが示された．ただし,意図的学習条件と印象判断条件の被験者は再認の際に注視する部位が異なっていたことから,それぞれ異なる方略を用いて顔の早期を行っている可能性が示唆された.

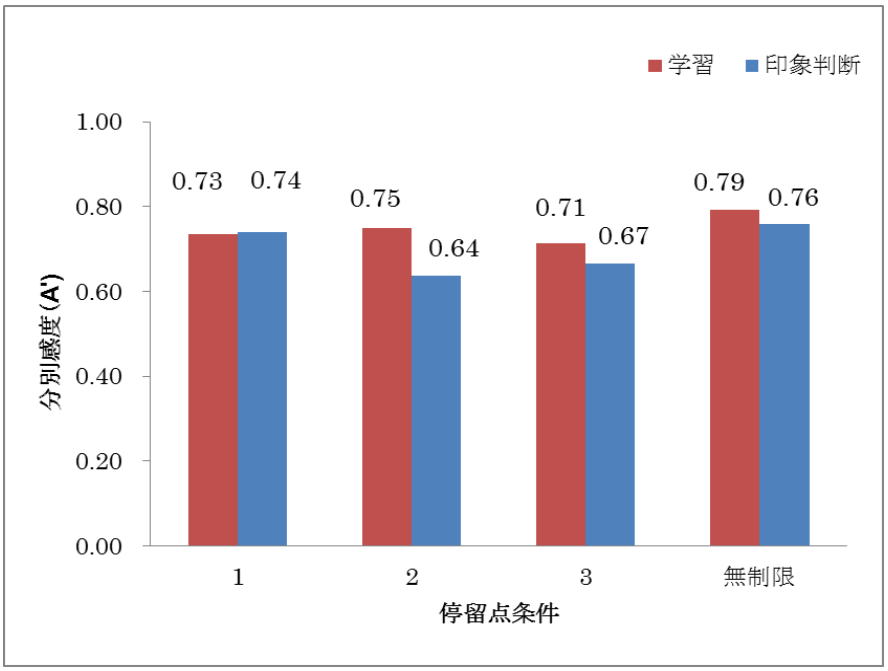


図 21　分別感度 A’の結果

4. まとめ

意図的学習時とその後の再認時で,長く注目する視線の停留点領域が異なっていることが見られ,Hsiao ら[1]の先行研究結果を確認することができた．また,意図的学習と印象判断という顔の観察条件の違いによって,長く注視される停留点領域には差異が認められなかったが,印象判断の方が停留点領域のばらつきが大きいことも分かった．再認時では,第 1 段階での顔画像観察条件に応じて,長く注視される停留点の領域が異なっていることが分かった．

また,顔画像観察条件の違いによる再認成績に差異が無かったが,停留点条件には有意な差が認められた．このことから,1 つの停留点があれば再認可能であり,先行研究[3]の結果も確認することができた． 意図的に顔を覚えた方が,印象判断を行った時よりも再認成績がよいという,先行研究[2]とは逆の結果が得られた,この理由についての考察は今後の検討課題としたい

5. 謝辞

本研究の一部は，科学研究費補助金（新学術領域研究 20119005, 基盤研究(B) 21300084）の助成を得た．記して謝意を表す.

本研究を遂行するにあたり，日頃からご指導，ならびに貴重なご意見を頂いた，法政大学工学部 赤松茂 教授に厚く御礼申し上げます.

また，共同研究者としてご意見，ご協力いただいた早稲田大学 作田由衣子准教授，法政大学工学部システム制御工学科 4 年の中村夏子氏をはじめ，研究に協力していただいた赤松研究室の皆様に深く感謝致します.

参考文献

[1] Hsiao,J.H. & Cottrell,G.: “Two fixations suffice in face recognition” , Psychol Sci. 2008 Oct;19(10):998-1006.

[2] Bower,G.H. & Karlin, M. B.: ”Depth of processing pictures of faces and recognition memory” , Journal of Experimental Psychology, 1974, 103(4), 751-757.

[3] Willis,J. & Todorov,A.,“First impressions: Making up your mind after a 100-ms exposure to a face,” Psychol. Sci., 17(7), pp.592-598, 2005.

[4] 中村亮太,中村夏子,遠藤聖也,作田由衣子,赤松　茂,” 顔画像の意図的学習時と印象判断時での観
察行動の眼球運動計測による比較,”映像情報メディア学会年次大会, 4-12, Oct.2010.

[5] “EyeLink CL” http://www.sr-research.com/camup_remote.html

[6] “Experiment Builder” http://www.monte.co.jp/images/products/srr/option/Experiment.pdf

[7] "顔画像合成システム FUTON", http://www.atr-p.com/futon.html

[8] Martin Handford, “ウォーリーをさがせ”, フレーベル館,1987

[9] “Wikipedia”
http://ja.wikipedia.org/wiki/%E5%88%86%E6%95%A3%E5%88%86%E6%9E%90

[10] “ANOVA4 on the Web” http://www.hju.ac.jp/~kiriki/anova4/

発話時の顔画像を用いた発声単語認識システムの構築

CONSTRUCTION OF RECOGNITION SYSTEM OF SPOKEN WORDS USING FACE IMAGES AT THE UTTERANCE

中村　亮太

Ryota NAKAMURA

指導教員　赤松　茂

法政大学大学院工学研究科システム工学専攻修士課程

This paper describes a vision-based spoken word recognition system that utilizes, instead of audio signal, visual motion signal which is obtained from motion pictures during speech. Motion information on each pixel in the input time-series imagery was obtained by computation of optical flow. The utterance interval was extracted by defining both starting and ending points of time for each spoken word, and some peek frames in the utterance interval were obtained. Subspace method (CLAFIC: CLAss-Featuring Information Compression method) was applied to register in the classification dictionary and to classify each spoken words.

As a preliminary performance evaluation of the proposed feature in spoken word recognition, discrimination test of seven spoken words including A-RI-GA-TO-U and KO-N-NI-CHI-WA was conducted.

Key Words: spoken words, lip-reading, optical flow, subspace method

1. はじめに

今日, 日本では高齢化が世界に類を見ないスピードで進行している[1]. 一般的に高齢化に伴い身体の不自由な方が増加していく. その反面, 技術の進歩により, システムの仕様がより複雑になってきているため, 誰でも使いやすくより負担の少ないインタフェースの導入が必要になってきている. 福祉に対する工学的なアプローチ方法も数多く研究されており[2][3], インタフェースの実用化もされつつある. そこで, 人間は視覚によって顔から様々な意味を抽出することができること[4]を利用し, 日常動作である「発話」に注目してみる. 発話によるインタフェースは環境にロバストであり, 難しい動作を含まないため負担が少ない.

視覚処理を用いた発話認識の先行研究には単語の各母音に対応する口の形に注目したもの, 口の周辺の動きに注目したものの 2 つのアプローチ方法が挙げられる. 前者の先行研究では, 口の位置を色抽出法と Eigentemplate 法を用いて検出し, 口の形の時間的な変化を固有空間法で記述し, 認識を行っているもの[5]がある. 最近では, Avtive Appearance Model を用いて口の形の特徴を取得している研究[6][7]も挙げられている. しかしながら, これらのアプローチでは個人の各母音における口の形の違いによる影響を受けてしまうこともある. また, 日本語を発話する際は口の開閉動作が大きくなくても発話することができてしまう.

後者の先行研究では, 発話された区間の口の開き具合と口の動く速さを表す 2 つの差分系列を取得後, これらを 2 次元グラフ化し, 検出線とグラフの交差回数を特徴として認識を行っているもの[8]や口の上下左右の領域の動きを Optical Flow より算出し主成分分析を適用し, 口の開閉・伸縮の特徴を求め, それらを用いてマッチングを行い良好な結果が得られた研究も挙げられている[9].

また, 口の動きではなく, 顔の表情変化の認識でも, Optical Flow を用いて抽出された特徴を用いて良好な結果が得られている[10]. 表情の先行研究[11]では頬や口の部分に表情が表出されやすいと言われている. 中村らは, 発話に伴う動きにも有効であるかを確認するために, 口周辺の部分, 頬部分, 両者を結合した部分それぞれに注目して識別性能の比較を行い, 発話認識には. 口周辺の領域に注目した方が良いという結果を示した[12].

本論文では, 表情認識の先行研究[10]に動機づけられ, 後者のアプローチ方法を採用し, 発話中の口の動きの変化に注目した音声情報を用いない発声単語認識を目指す. また, 複数の被験者からサンプルを取得し, 不特定話者に対応した辞書の作成を行い, 識別する.

顔の動き特徴を用いる先行研究の多くは, 顔の位置が変化しないと仮定しているが, 実際の状況では頭の動きは避けられない.

そこで，本研究では，株式会社 ATR-Promotions が提供している顔のリアルタイム検出・追跡ソフト accFace^{TN} SDK 版[13](以後，accFace と略す)を用いて，発声画像より両目位置を取得し，顔の位置・傾き・大きさの自動正規化を行った．

また， 発話した動きによる速さを用いて，発話のピークを求め，ピーク時の画像の輝度値を用いて各クラスの部分空間を作成したものを用いて識別を行い，識別性能の比較を行った．

2. 発声単語認識

2. 1. 発声単語クラス

今回識別する単語は「ありがとう」「おめでとう」「こんにちは」「さようなら」「すみません」「はい」「いいえ」といった，誰にでも馴染みがあり，難しい動作の少ない挨拶の中から選択した．

2. 2. 時系列顔画像の取得

Web カメラで発声時の顔を正面から撮影し，サイズ 320×240 の動画を取得する．取得の際に以下の 2 つの制約条件を適用する．

(1) 正面を向いて発話してもらう
(2) 発話語は真顔に戻してもらう

2. 3. 顔画像の正規化

顔の大きさ，位置，傾きによらない動き情報を取得するために，顔の大きさ，位置，回転の正規化を行う．両目の座標を”accFace”を用いて取得し，両目間の距離が一定値(90 ピクセルに設定)になるように拡大・縮小処理を行う．更に，左右の目座標を基に回転処理を行う．また，両目座標に用いて口を中心に 128×128 の領域を切り出し，それにモノクロ画像に変換する．最後に，ガウシアンフィルターでノイズ除去を行った．図 1 に取得例を示す．今回用いたガウシアンフィルターのカーネルは 7×7 のサイズで，ガウス関数の標準偏差を 1．55 とした．

2. 4. 動き特徴の抽出

得られた時系列画像から各画素での速度の X, Y 成分を表す特徴を Optical Flow(Lucas-Knade method)を計算することで取得する．この時，フローの乱れを取り除くために，8×8 画素の領域で局所的に平均する．平均された速度を各局所領域の表す速度と仮定し，T フレームの時系列画像上の各領域においてこれを取得する．更に，合計 16×16 個の局所領域から口周辺の 9×7 個の平均速度を使って次章の発話区間の抽出を行っていく．図 2 に速度の取得例を示す．

2. 5. 発話区間の抽出

取得した T フレームの速度データには非発話区間も含まれているため，余分なデータが混在していると考えられる．そこで，発話区間を抽出し，フレーム数を T⇒T’(T>T’)にすることを考える．2．4 で取得した速度データを基に，各発声単語の発話の開始・終了フレームを自動的に求める．

まず，各フレームについて 9×7 の領域における X，Y 方向の速度から速度の大きさを算出し，T フレーム数分の空間平均速さを求める．更にそれを時間で平均し，時間平均速さを求める．

これを口の動いた指標とする．口周辺の 9×7 の局所領域を使用する理由は，全部の局所領域の平均速度を使用してしまうと，口が動いていない時も動いていると判断されてしまうからである．発話の開始・終了フレームの検出を以下のように定義する．

発話開始フレーム：各フレームの空間平均速さが時間平均速さを超えたフレームだけを抽出し，最初フレームを発話開始フレームとする．

発話終了フレーム：各フレームの空間平均速さが時間平均速さを超えたフレームだけを抽出し，最後のフレームを発話開始フレームとする．

以上の条件を満たした開始フレームと終了フレームの間の T’フレームを発話区間として抽出した．図 3 に空間平均速さと時間平均速さに基づく発話区間の抽出の例を示す．

2. 6. 発話ピークフレームの抽出

2．5 で取得した発話区間 T’の中から，更に発話ピークフレームを求める．発話ピークフレームとは，各音節を発話した瞬間のフレームのことを指す．

Fig. 1 顔の自動正規化

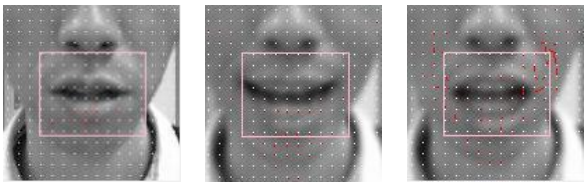


Fig. 2 Optical Flow の取得

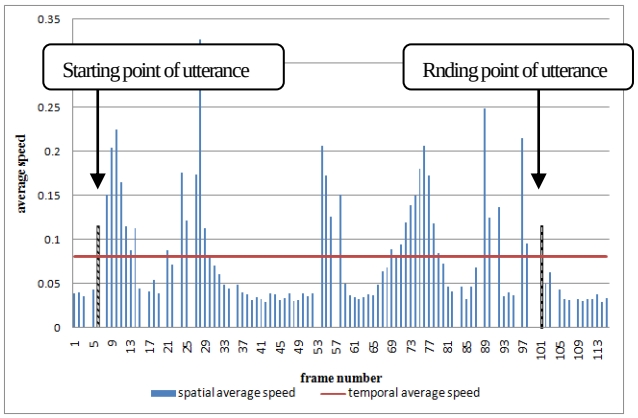


Fig. 3 発話区間の抽出

今回は 2 通りの発話ピークフレーム抽出を以下のように定義する.

(a) 各フレームの空間平均速さを求め，前後 1 フレームの速度差の絶対値の和をフレーム数分取得する．その後，その値の大きい順から上位 10 フレームを選択.
以上の条件を満たしたフレームを発話ピークフレームとして，それを取得する.

(b) 各フレームの空間平均速さを求め，更にそれを平均し，時間平均速さを取得する．各フレームの空間平均速さが時間平均速さを超えたフレームだけを発話ピークフレームとして取得する.

2. 7. 特徴ベクトルの作成

抽出された 10 フレームの発話ピーク画像の輝度値をベクトルにしたものを，$\mathbf{x}_1$，$\mathbf{x}_2$，…，$\mathbf{x}_{10}$と表すとき，$\mathbf{X}=(x_1, x_2, \ldots, x_{10})$を 1 サンプルの特徴ベクトルとし，全サンプル分のベクトル $\mathbf{X}_1$，$\mathbf{X}_2$，…，$\mathbf{X}_n$を作成した．(n：サンプル数)

2. 8. 再サンプリング

2．6 の(b)で抽出された発話ピークフレームは各サンプルで数が異なっているため，主成分分析にかけることができない．そこで，フレーム数を揃える再サンプリング処理を行った．揃えるフレーム数は全サンプルの中で最大のピークフレーム数とした.

2. 9. 部分空間法[14]による辞書の作成

部分空間法は，各クラスごとにそのクラスを表現する低次元の部分空間を用意し，未知サンプルがどの部分空間で最もよく近似表現できるかを比較することにより，未知パターンを識別する方法である．今回は CLAFIC 法(CLAss-Featuring Information Compression)を用いることにした．CLAFIC 法では，部分空間の基底を決めるために，自己相関行列を用いて主成分分析を行い，認識時には，データを部分空間に射影した時の大きさを類似度として用いる手法である．本論文では，7 つの発話クラスそれぞれの部分空間を算出し，辞書として登録した.

部分空間法を以下に示す.

ここで，k 個のクラス c_1，c_2，…，c_k のそれぞれの部分空間を L_1，L_2，…，L_k，その次元を d_1，d_2，…，d_k とする．各クラス c_i について部分空間 L_i を張る d_i 個の d 次元正規直交ベクトルを $\mathbf{u}_{i1}$，$\mathbf{u}_{i2}$，…，$\mathbf{u}_{id}$ とする.

クラス c_i に着目し，d 次元特徴空間から d_i 次元部分空間への変換を表す行列を $\mathbf{A}_i$ とすると，

$\mathbf{A}_i=(\mathbf{u}_{i1}, \ldots, \mathbf{u}_{di})$

と書ける．この式より，

$$A_i^t A_i = I \tag{1}$$

が成り立つ．I は単位行列である．部分空間に射影された di 次元特徴ベクトル$A_i^t x$を元の d 次元空間で見ると$A_i A_i^t$となり，元の空間から部分空間 L_i への変換は直交射影行列

$$P_i = A_i A_i^t = \sum_{j=1}^{d_i} u_{ij} u_{ij}^t \tag{2}$$

によって表すことができる.

そして，$\mathbf{x}$ の部分空間 L_i への正射影は $P_i\mathbf{x}$ である.

2. 10. 識別判定の方法

部分空間法で学習された辞書と，未知サンプルの類似度 S を計算し，一番類似度の高いクラスに属すると識別する．部分空間法の類似度は部分空間への射影の長さで定義され，

$$S_i(\mathrm{x}) = \|\mathrm{P}_i\,\mathbf{x}\|2 = \mathbf{x}^t P_i \mathbf{x} \tag{3}$$

で計算することができる．さらに効率の良い計算方法を求めると，

$$S_i(\mathrm{x}) = \sum_{j=1}^{d_i} \left(u_{ij}^T \mathbf{x}\right)^2 \tag{4}$$

となる．この値が大きい程，類似度が高いと判定される.

3. 実験

3. 1. サンプルの収集

動画は，6 人に 2．2 の条件を設定し，「ありがとう」「おめでとう」「こんにちは」「さようなら」「すみません」「はい」「いいえ」の 5 単語を各 4 回ずつ発話してもらい取得した．つまり，6 人×7 単語×4 回の 168 サンプルを用意した.

3. 2. 識別性能の評価法

本研究では識別性能の評価法として leave-one-out 法[12]を用いる． 120 サンプルの中から 1 サンプルを選んで未知サンプルとし，残りの 119 サンプルを用いて学習部分空間を作成し，類似度の高いクラスに分類し，これを 120 個のテストサンプルについて繰り返しテストする.

3. 3. 部分空間を用いた識別性能評価

今回の実験では，先行研究[12]を参考に，口周辺の領域画像を用いて部分空間を作成した.

再サンプリングは各サンプルの最大フレーム数を求め，それに合わせるように計算を行った．今回の実験では 10 フレームに揃えるように行った．また，先行研究[12]の手法と比較を行った．先行研究では，Optical Flow 特徴を用いて，各特徴領域の x，y 方向の速さとスカラーと角度を時間軸で平均したものを特徴ベクトルとして計算を行い，Fisher の判別基準を用いて，学習を行っている．(以下，Fisher と表す)

3. 4. 実験結果と考察

識別結果を図 4 に示す.

識別結果を見ると，先行研究の手法を用いた場合の識別率が高いことがわかる．また，部分空間を用いた結果では，部分空間によって作成された各単語の空間が類似してしまっていることが，識別率の低下につながってしまったと考えられるが，識別はする

ことが可能であると考えられる. 再サンプリングを導入したものは, もともと存在しない画像を作成しているため, 各被験者ごとの分散が大きくなってしまったものと考えられる. これを解決するには, 部分空間を作成する際, または識別をする際に, 各単語のサンプルに共通するパターンとの差分を算出するか, 各単語のサンプルに重みづけする必要があると考えらえる.

表１に混同行列を示す. 行の単語名は正解単語を表し, 列の単語は認識結果を, 数値は識別率(%)を表している.

(b), (c)の表を見ると, 「おめでとう」と「はい」への誤識別が多くなっていることが分かる. 取得した発話ピークフレーム数が少なかったため, 単語自体の特徴が表れにくかったことと考えられる.

4. まとめ

今回, ある時間軸での顔の空間的な特徴に注目した発声単語認識の手法を検討した.

発話区間の抽出では口周辺の局所領域を用いたことで, 顔の動きと口の動きを分けることができたため, 精度の高い発話区間の抽出ができた. 先行研究に準じて求めた結果と部分空間を用い識別結果は, 今回用いたサンプルについては識別が可能であると分かった.

新たに導入した再サンプリングと部分空間を用いた手法では, 良好な結果を得ることができなかった. 部分空間を用いた識別の先行研究では, 時系列画像を１つの特徴ベクトルとせず, 時系列画像を１つの空間として, 空間と空間の類似度を測る手法があげられているが, これには多くのサンプルを用いてしまうため, 個人に依存しない発話認識には有効ではない. 各単語のサンプルに共通するパターンとの差分を算出し, より単語の違いを表す特徴ベクトルの作成方法を検討しなければならないことが分かった.

参考文献

[1] 内閣府編,”高齢社会白書”,2008.

[2] 金山和功,白井良明,島田伸敬,”HMM を用いた手話単語の認識”, 信学技 ,Vol.104,No.89,PRMU2004-16,pp.21-28,May,2004.

[3] 藤野雄一,伊福部達,”マルチメディア時代における医療・福祉支援技術”, 電子情報学会誌,Vol.80,No.8,pp822-829,Aug.,1997.

[4] 赤松茂,”コンピュータによる顔の認識-サーベイ”,信学論(D-2),Vol.J80-D-2,no.8,pp.2031-2046,Aug.,1997.

[5] 中田康之,安藤護俊,” 色抽出法と Eigentemplate 法を併用した口の位置検出と読唇処理への適用”, 信学技,PRMU2001-95,pp.7-12,Sep,.2001

[6] 齊藤剛史,森下和敏,小西亮介,” 複数口唇領域を利用した多言語に有効な単語読唇”, 信学技,PRMU2008-76,pp.181-186,Sep.,2008

[7] 森下和敏,齊藤剛史,小西亮介,”発話シーンからのキーフレーム検出と単語読唇”,MIRU2009, IS1-58,pp.753-760,Jul.,2009

[8] 清田公保,内村圭一,”口唇周辺画像情報を用いた発話単語認識”,信学論,Vol.J76-D-Ⅱ,No.3,pp.813-814,Mar.,1993.

[9] 間瀬健二,アレックス・ペントランド,”オプティカルフローを用いた読唇”,信学論,J73-D-2,6,pp.796-803,Jun.,1990.

[10] 間瀬健二,前田英作,末永康仁,”表情動画像からの感情の認識の１手法”,信学技,PRU91-24,pp.95-101,1991.

[11] 中島 慶, 藤江 真也, 松坂 要佐, 小林 哲則,” 対話調整的役割を果たす顔表情の認識”, 信学技,PRMU2005-105,pp.7-12,Oct.,2005

[12] R.Nakamura, S.Akamatsu, “Recognition of Spoken Words Using Motion Features Extracted from Video Sequences:Comparison of Recognition Performance Dependent on Attention Area of Face”, APSIPA, Dec., 2010

[13] ATR-Promotions Ltd., "accFaceTN SDK", http://www.atr-p.com/af.html

[14] 石井健一郎,前田英作,上田修巧,村瀬洋,”わかりやすいパターン認識”,オーム社,1998.

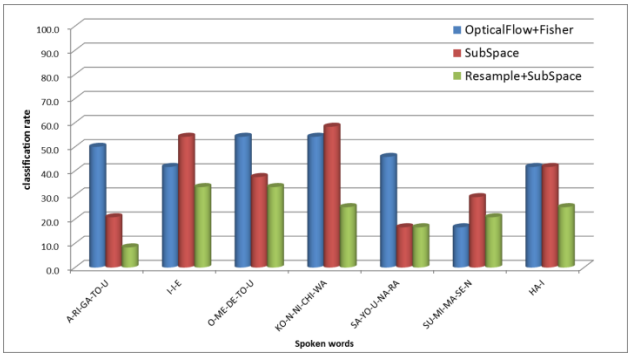


Fig. 4 識別結果

Table.1 各手法での混同行列

(a) Fisher

		recognized word						(%)
		A-RI-GA-TO-U	I-I-E	O-ME-DE-TO-U	KO-N-NI-CHI-WA	SA-YO-U-NA-RA	SU-MI-MA-SE-N	HA-I
testing word	A-RI-GA-TO-U	50.0	16.7	0.0	0.0	4.2	4.2	12.5
	I-I-E	4.2	41.7	8.3	4.2	4.2	8.3	4.2
	O-ME-DE-TO-U	4.2	0.0	54.2	0.0	0.0	0.0	4.2
	KO-N-NI-CHI-WA	0.0	4.2	4.2	54.2	4.2	4.2	0.0
	SA-YO-U-NA-RA	8.3	12.5	4.2	0.0	45.8	12.5	0.0
	SU-MI-MA-SE-N	4.2	8.3	16.7	4.2	12.5	16.7	4.2
	HA-I	12.5	0.0	8.3	0.0	0.0	4.2	41.7

(b) SubSpace

		recognized word						(%)
		A-RI-GA-TO-U	I-I-E	O-ME-DE-TO-U	KO-N-NI-CHI-WA	SA-YO-U-NA-RA	SU-MI-MA-SE-N	HA-I
testing word	A-RI-GA-TO-U	20.8	41.7	29.2	0.0	0.0	0.0	8.3
	I-I-E	0.0	54.2	20.8	8.3	4.2	8.3	4.2
	O-ME-DE-TO-U	12.5	20.8	37.5	0.0	0.0	0.0	29.2
	KO-N-NI-CHI-WA	8.3	4.2	16.7	58.3	0.0	0.0	12.5
	SA-YO-U-NA-RA	4.2	33.3	29.2	4.2	16.7	8.3	4.2
	SU-MI-MA-SE-N	0.0	4.2	20.8	0.0	12.5	29.2	33.3
	HA-I	0.0	20.8	33.3	0.0	0.0	4.2	41.7

(c) ReSample+SubSpace

		recognized word						(%)
		A-RI-GA-TO-U	I-I-E	O-ME-DE-TO-U	KO-N-NI-CHI-WA	SA-YO-U-NA-RA	SU-MI-MA-SE-N	HA-I
testing word	A-RI-GA-TO-U	8.3	20.8	45.8	4.2	0.0	0.0	20.8
	I-I-E	4.2	33.3	41.7	4.2	8.3	0.0	8.3
	O-ME-DE-TO-U	4.2	4.2	33.3	0.0	0.0	0.0	58.3
	KO-N-NI-CHI-WA	0.0	8.3	25.0	25.0	0.0	0.0	41.7
	SA-YO-U-NA-RA	4.2	8.3	29.2	8.3	16.7	8.3	25.0
	SU-MI-MA-SE-N	0.0	8.3	25.0	8.3	0.0	20.8	37.5
	HA-I	0.0	8.3	45.8	12.5	4.2	4.2	25.0

顔の動き特徴を用いた発声単語認識システムの構築

法政大学工学研究科
システム工学専攻修士2年
中村　亮太

はじめに

- 情報バリアフリー

年齢の増加に伴い、身体・認識能力の低下

↕

複雑な操作が操作を伴うシステム

誰でも使いやすくより負担の少ない
インタフェースの導入が必要

研究目的

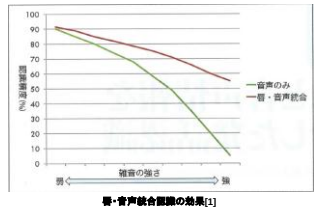
音情報を用いず、口周辺の動きに注目した発声単語の識別を行う

→ 読唇術

- 読唇について
  - 雑音下においても特徴をとらえることができる
  - 声を出さない状態でも、その動きから識別を行うことができる
  - 識別性能があがりにくい

発話認識の現状

- 唇の動きを用いた認識の位置づけ[1]
  - 音声認識との統合認識



唇・音声統合認識の効果[1]

雑音下に強い画像情報を用いた発話認識手法が必要

唇の動きを用いた単語認識の先行研究

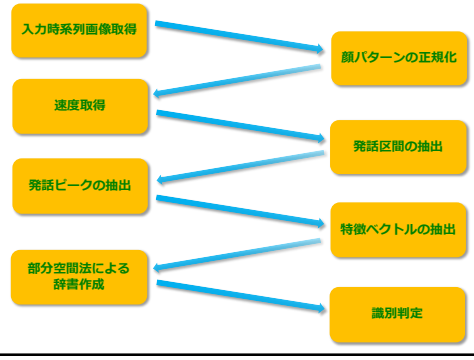
- 2つのアプローチ方法
  - 輪郭ベース特徴量
    - 唇の輪郭モデルのパラメータを特徴量[2]
    - Active-Appearance-Modelを用いて口の形（唇領域の横幅、高さなど）の特徴を取得[3]

各個人の口の形に依存

アプローチ

- アピアランスベースの特徴量
  - 口の上下左右の領域の動きをOptical Flowより算出し口の開閉・伸縮の特徴を取得[4]
  - 本人についてのみの唇領域の動画像を用いて、制約相互部分空間法によって特徴抽出[5]

識別までの処理フロー



入力時系列画像取得の流れ

- 動画の撮影
  - ① 正面顔を維持してもらうように指定
  - ② 発話語は真顔に戻してもらう

- 7単語を6人に4回ずつ発話してもらう
  - A-RI-GA-TO-U
  - I-I-E
  - O-ME-DE-TO-U
  - KO-N-NI-CHI-WA
  - SA-YO-U-NA-RA
  - SU-MI-MA-SE-N
  - HA-I

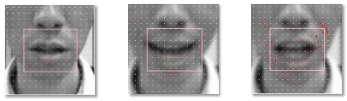
計168サンプルを取得

顔パターンの正規化

- ATR-Promotionsが提供しているaccFace^{TN}SDKを用いて正規化
  - accFaceで得られた両目座標を基に、拡大・縮小・回転
  - 両目座標を基に、口唇領域を切り出す

速度取得

- 口を中心に128×128で切り出された画像から、Optical Flow(Lucas-Kanade)を用いて速度を取得する
  - モノクロ化処理を行う
  - ガウシアンフィルタをかける
  - 8×8の局所領域でフローを平均する
    - 全部で16×16個の局所領域とその局所平均フローを取得する

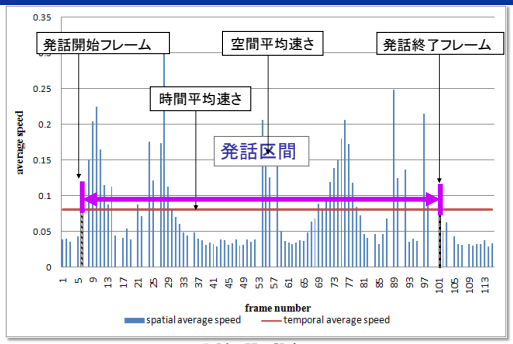


発話区間の抽出

- 発話区間の抽出とは
  - 発話開始フレームと発話終了フレームを決定
  - その間の区間を単語を発声している区間とし、フローを抽出
  - 口周辺の9×7個の局所領域のX,Y方向の速度のみを使用する

余分な時間を省くことで発話と関係のない画像を削除

発話区間の抽出



発話区間の設定

発話ピークフレームの抽出

- 発話ピークフレームとは，各音節を発話した瞬間のフレームのことを指す
- 2通りの発話ピークフレームを定義
  - 各フレームの空間平均速さを求め，前後1フレームの速度差の絶対値の和をフレーム数分取得する．その後，その値の大きい順から上位10フレームを選択．
  - 各フレームの空間平均速さを求め，更にそれを平均し，時間平均速さを取得する．各フレームの空間平均速さが時間平均速さを超えたフレームだけを発話ピークフレームとして取得する．

抽出されるフレーム数が異なるため、時間軸の再サンプリングを行う

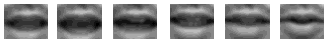
特徴ベクトルの抽出

- 抽出されたフレームの発話ピーク画像の輝度値をベクトルにしたものを

  $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$

  と表す
- 1サンプルを$\mathbf{X}_1=(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$で表し、全てのサンプルを$\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$として抽出した

  (n : サンプル数)



$\mathbf{X}_k$

12

部分空間法による辞書の作成と識別判定

- 部分空間法とは
  - 各クラスごとにそのクラスを表現する部分空間を用意し，未知サンプルがどの空間に近いか比較する方法
- CLAFIC法
  - 部分空間の基底を決めるために，自己相関行列を用いて主成分分析を行い，認識時には，データを部分空間に射影した時の大きさを類似度として用いる手法
- 7つの発話クラスそれぞれの部分空間を算出し，辞書として登録した
- 類似度の一番高いクラスに識別する

$$S_i(\mathbf{x}) = \sum_{j=1}^{d_i} (u_{ij}^T \mathbf{x})^2$$

13

実験概要

- 識別性能の評価法
  - leave-one-out法を用いる
    - 168サンプルの中から1サンプルを選んで未知サンプルとし、残りの167サンプルを学習サンプルとし識別を行う
    - 未知サンプルは168個あるので繰り返し行う

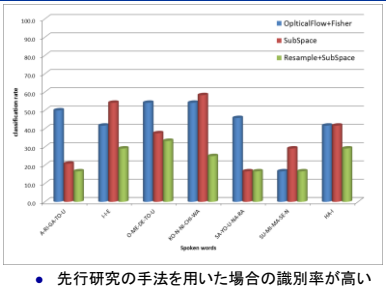
14

実験

- 各手法と識別性能評価
- 比較する手法
  - 先行研究[6]に準じたもの(OpticalFlow+Fisher)
  - 速度差の大きい上位10フレームに部分空間法を用いたもの(SubSpace)
  - 再サンプリングを行った後、部分空間法を用いたもの(ReSample+SubSpace)

15

実験結果と考察



平均識別率

Fisher 43.5 %

SubSpace 36.9 %

Resample+SubSpace 23.8%

- 先行研究の手法を用いた場合の識別率が高い
- 部分空間を用いた結果では，部分空間によって作成された各単語の空間が類似してしまっている
  - 識別率の低下につながってしまったと考えられるが識別はすることが可能であると考えられる

16

実験結果と考察

- **各手法の混同行列**

OpticalFlow+Fisher

(%)

testing word \ recognized word	A-RI-GA-TO-U	I-I-E	O-ME-DE-TO-U	KO-N-NI-CHI-WA	SA-YO-U-NA-RA	SU-MI-MA-SE-N	HA-I
A-RI-GA-TO-U	50.0	16.7	0.0	0.0	4.2	4.2	12.5
I-I-E	4.2	41.7	8.3	4.2	4.2	8.3	4.2
O-ME-DE-TO-U	4.2	0.0	54.2	0.0	0.0	0.0	4.2
KO-N-NI-CHI-WA	0.0	4.2	4.2	54.2	4.2	4.2	0.0
SA-YO-U-NA-RA	8.3	12.5	4.2	0.0	45.8	12.5	0.0
SU-MI-MA-SE-N	4.2	8.3	16.7	4.2	12.5	16.7	4.2
HA-I	12.5	0.0	8.3	0.0	0.0	4.2	41.7

SubSpace

(%)

testing word \ recognized word	A-RI-GA-TO-U	I-I-E	O-ME-DE-TO-U	KO-N-NI-CHI-WA	SA-YO-U-NA-RA	SU-MI-MA-SE-N	HA-I
A-RI-GA-TO-U	20.8	41.7	29.2	0.0	0.0	0.0	8.3
I-I-E	0.0	54.2	20.8	8.3	4.2	8.3	4.2
O-ME-DE-TO-U	12.5	20.8	37.5	0.0	0.0	0.0	29.2
KO-N-NI-CHI-WA	8.3	4.2	16.7	58.3	0.0	0.0	12.5
SA-YO-U-NA-RA	4.2	33.3	29.2	4.2	16.7	8.3	4.2
SU-MI-MA-SE-N	0.0	4.2	20.8	0.0	12.5	29.2	33.3
HA-I	0.0	20.8	33.3	0.0	0.0	4.2	41.7

17

まとめ

- 発声単語認識への1手法
  - 発話区間の抽出では口周辺の局所領域を用いたことで，顔の動きと口の動きを分けることができたため，発話ピーク時の特徴を捉えることができた
  - 音声認識の補助的な手法として，発話ピークフレームを用いた手法は有効である
  - 再サンプリングと部分空間を用いた手法は未だ検討の余地がある
  - 各単語のサンプルに共通するパターンとの差分を算出し，より単語の違いを表す特徴ベクトルの作成方法を導入する必要がある

18

参考文献

[1] 有木康雄,駒井祐人,"画像と音声情報を統合した発話認識",情報処理学会誌,Vol.52 No.1,pp.87-94,Jan.2011

[2] 齊藤剛史,小西亮介,"横顔画像の輪郭形状に基づく読唇,"信学技,PRMU2008-77,pp.187-192,Sep.2008

[3] 齊藤剛史,森下和敏,小西亮介," 複数口唇領域を利用した多言語に有効な単語読唇",信学技,PRMU2008-76,pp.181-186,Sep.2008

[4] 間瀬健二,アレックス・ペントランド,"オプティカルフローを用いた読唇",信学論,J73-D-2,6,pp.796-803,Jun.1990.

[5] 西山正志,山口修,福井和広,"制約相互部分空間法を用いたジェスチャー認識," SSII04,Vol10,pp.439-444,2004

[6] R.Nakamura, S.Akamatsu, "Recognition of Spoken Words Using Motion Features Extracted from Video Sequences:Comparison of Recognition Performance Dependent on Attention Area of Face", APSIPA, Dec., 2010

19

ご清聴ありがとうございました

20