ヒューマンインタフェース 68－1
（1996．9．12）

唇の動き情報を利用した単語認識

奥村 晃弘　岡野 健治　宮崎 敏彦　藤井 明宏
沖電気工業(株)　研究開発本部　関西総合研究所

我々は、唇の動き情報を利用した発話理解の研究として、(1) 音声認識と唇情報の統合方式の研究, (2) 唇の動き情報(視覚情報)のみによる発話理解の研究を進めている。本報告では、我々が試作した唇情報のみによる単語認識システムにおける、マッチングの手法と実験結果について報告する。

A Word Recognition Method using Lip Movements Information

Akihiro OKUMURA　Kenji OKANO　Toshihiko MIYAZAKI　Akihiro FUJII
Oki Electric Industry Co.,Ltd.
Crystal Tower 2-27 Shiromi 1-chome,Chuo-ku,Osaka 540,JAPAN

We have been studying a speech recognition system which robust to background noises. Our major interests are (1) method to fuse auditory information and lip movements (2) lipreading system (use lip movements only). This paper describes matching algorithm in our lipreading system and result of recognizing performance test.

1　はじめに

最近では、ＯＳもグラフィカルなユーザインタフェースで利用できるようになり、パソコン利用の大衆化に拍車が掛かるようになった。また、銀行のＡＴＭや鉄道の切符自動販売機などにもタッチパネルが導入されるようになり、誰もが簡単に複雑な入力ができるようになった。ならば、グラフィカルなユーザインタフェースが万能であるかと考えると以外とそうでもなさそうである。例えば多くの選択枝の中から自分の欲している項目を選ぶのは、思いのほか骨が折れる。そもそも、グラフィカルなインタフェースが目で確認してそれをダイレクトに指し示すという性質上、画面の表示能力の限界が存在するし、一般に、迅速に必要な項目を選ぶことができる選択枝の数は５個程度であると言われているのだから、これは当然と言えよう。このような、状況ではやはり音声認識などの認識機能を持つインタフェースが有望である。認識機能を持つインタフェースとしては、音声認識の他に文字認識があるが、文字認識が携帯端末などで実用化され普及しているのに比べて、音声認識装置が本格的に実用化されている例は数少ない。我々はこの原因が利用環境での雑音にあると考えている。つまり、比較的静かな環境では実用に耐えうる認識率が得られるシステムであっても、実際に一般のオフィスなどで使用するとなると、周囲の雑音の影響により十分な性能を発揮できないのが現状である。これらの理由により、我々は唇の動き情報を使うことによって、周囲の雑音に影響されにくい音声認識システムを構成する研究を行なっている[1]。そのためには、音声処理と画像処理を融合させる必要があるが、それに先駆けて、唇の動き情報の有効性について検証するために、唇の動きをそのままマッチングさせることによって、特定話者の単語認識システムを構成した。

本稿では，唇の動き情報のみによる単語認識システムにおけるマッチング手法と、認識実験の結果について述べる。

2 唇の動き情報の抽出

2.1 抽出方法

唇の動きを捕らえるに当たって、図1に示すように、口の縦方向の開き具合(*height*)と、横方向の開き具合(*width*)を特徴量として利用することにした。実際には、唇は前に突き出すなど立体的に動き、また、唇そのものも口の開き具合などによって変形すると考えられるが、今回は単純化して扱うことにした。画像から特徴量を得る方法としては、口の輪郭を抽出する方法[2]などが提案されているが、縦方向の開き具合(*height*)と、横方向の開き具合(*width*)しか利用しないので輪郭の位置を正確に求める必要はない。むしろ、動きを正確に捕らえるべきであるから、唇上のある点の動きをテンプレートマッチングを使って追跡することにした。追跡する点(特徴点)としては、追跡の容易さも考慮して、図2に示す4点($P_0 \sim P_3$)を利用し、特徴点の第1フレーム画像での位置を手作業で指定した。この特徴点の座標から *height* と *width* を算出すれば良いのだが、式1で求めた H_o,W_o に関して、W_o は *width* と等しくなるが、H_o は *height* と等しくならない(図3A)。

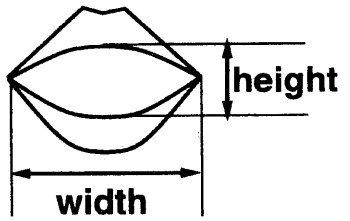

図 1: マッチングに利用する特徴量

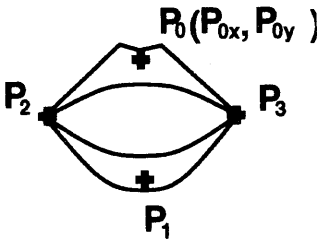

図 2: 唇上の4つの特徴点の位置

$$H_o = |P_{0y} - P_{1y}|, \quad W_o = |P_{2x} - P_{3x}| \qquad (1)$$

これは、特徴点追跡の都合により特徴点 P_0,P_1 を唇の外輪近くに設定したためである。そこで、唇の厚さを一定とみなし口を閉じた状態(図3B)の特徴点間の距離を H_c として以下のように表すことにした。

$$height = H_o - H_c, \quad width = W_o \qquad (2)$$

口を閉じた状態は、発話の前後はなるべく口を閉じるように指導しておき、後述する音節ラベルから発話開始時間を得て、それ以前で H_o が比較的安定した状態の時を利用した。

以上の方法で唇の動き情報を抽出し、この動き情報と同期した音声データを使って、各音節区間のラベリングを人手にて行なった。

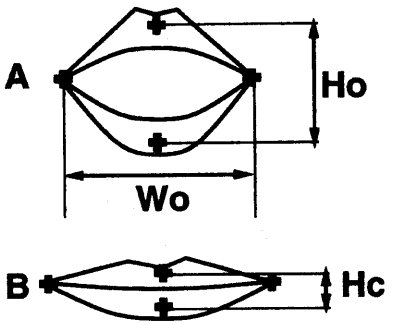

図 3: 式1で算出される量

2.2 データ収集

唇の動きの最大の特徴として「両唇音」がある。'b','m','p' で始まる音がそれで、発音する際に上唇と下唇が触れ合う必要がある音をさす。この両唇

音を含むものを中心とする３０種類の単語を選び、話者がこの単語を発話するところを正面から撮影した。撮影は２日に分けて行ない、１日に約６００データ(１セット＝３０単語×５回を４セット)、総計１２２１データを撮影した。

１日目のデータは辞書作成に利用し、２日目のデータを使って認識性能の評価を行なった。

3 パターンマッチング

3.1 マッチングの前処理

特徴量の *height* と *width* は目の間の距離を使って正規化し、撮影倍率の違いを吸収する。目の間の距離は特徴点の初期値と同様に、手作業で指定した。

また、線形補完により特徴量を増加させた上で、ローパスフィルターを通すことでスムージングを行なう。これは、１秒間 30 フレームのサンプリング数の少なさを補うためである。後述する認識実験においては、補完数を０～４と様々に変化させて実験を行なった。

発話時間長の変動や特徴点の時間的なずれが存在するパターンを比較する必要があるので、マッチングにはＤＰマッチングを利用することにした。また、発話前後の口の動きを避けるために、発話部分だけを切り出して始終端固定で行なった。この発話部分の切り出しには、音節ラベルを利用する。しかし、音声によって設定した音節ラベルをそのまま使うと、音声での発話終了時では、まだ口が閉じていない場合が存在し、マッチングがうまくいかない。そこで、音節ラベルによる発話終了後で口がある程度閉じたときをもって発話終了とした。

これらの前処理を行なう過程を図4に、また、前処理の結果の例を図5に示す。

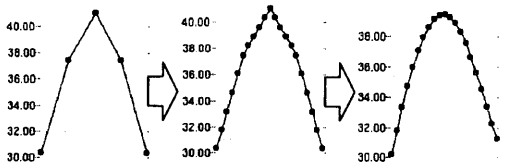

図 4: 前処理の過程 (補完数＝４)

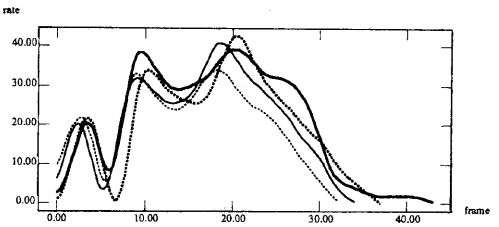

図 5: 前処理の結果 (にまいがい)

3.2 角度によるマッチング

図5を見ても分るように、同一単語を発話した場合特徴量のグラフは、山や谷の大まかな位置や形など概形の類似性が高い。従って、波形の特徴となっている極大値や極小値を取る時が対応していると言える。従って、ＤＰマッチングによって時間的なずれをなくすためには、極大値や極小値を取る時間を対応させる必要がある。そこで、特徴ベクトル間の距離計算に１つ前の特徴量との変化量から算出した角度を利用する。この角度は極大値および極小値付近で０に近付くので、極値どうしの距離を小さくすることができる。

今、辞書パターン T と認識パターン R は特徴量の時系列であり、これらをＤＰマッチングで比較するものとする。

$T = a_1, a_2, \ldots, a_i, \ldots, a_I$

$R = b_1, b_2, \ldots, b_j, \ldots, b_J$

時間変換関数を $F = c(1), c(2) \ldots, c(k), \ldots, c(K)$ とすると、パターン T とパターン R の距離 $D(T,R)$ は F を様々に変えたときの特徴量間距離の荷重平均の最小値であるから、以下の式で表すことができる。

$$D(T,R) = \min_F \frac{\sum_{k=1}^{K} d(c(k)) \cdot w(k)}{\sum_{k=1}^{K} w(k)} \qquad (3)$$

但し $c(k) = (i(k), j(k))$ であり、$i(1), \ldots, i(K)$ はパターン T を、また、$j(1), \ldots, j(K)$ はパターン R を伸縮させた結果の添字の系列を示す。また、$d(c(k))$ は $a_{i(k)}$ と $b_{j(k)}$ の特徴量間距離であり、$w(k)$ は荷重係数である。

ここで、特徴量間距離 $d(c(k))$ を、１つ前の特徴量との変化量から算出した角度の差とするので、式4のように定義できる。

$$\left.\begin{aligned} d(c(k)) &= d(i,j) = |r_a - r_b| \\ r_a &= tan^{-1}\left(\frac{a_i - a_{i-1}}{v}\right) \\ r_b &= tan^{-1}\left(\frac{b_j - b_{j-1}}{v}\right) \end{aligned}\right\} \quad (4)$$

但し、r_aはパターンTでの角度、r_bはパターンRでの角度であり、vは感度を示す係数である。今回はvに辞書用の全体データにおける平均変化量を用いた。

3.3 変化量に対応したマッチング手法

式4のように特徴量間の距離を定義する場合に、iとjの関数として表すと、伸縮とは無関係になっている。そのため、式4を使って求めたパターンどうしの距離は、ＤＰマッチングの際の伸縮により時間を正規化したパターンどうしの距離と厳密には異なってしまう。

そこで、特徴量間の距離の定義を、伸縮結果を参照する様に改良することを考える。$i(1),\ldots,i(K)$と$j(1),\ldots,j(K)$は伸縮後の添字の系列であるからこれを使うと、式5となる。

$$\left.\begin{aligned} d'(k) &= |r'_a - r'_b| \\ r'_a &= tan^{-1}\left(\frac{a_{i(k)} - a_{i(k-1)}}{v}\right) \\ r'_b &= tan^{-1}\left(\frac{b_{j(k)} - b_{j(k-1)}}{v}\right) \end{aligned}\right\} \quad (5)$$

しかし、このままでは同じ特徴量を繰り返し使用してパターンを伸ばす場合に極大値および極小値以外の場合も変化量が0になってしまう。従ってその様な場合は前後の角度から補完して求めることにする(図6)。

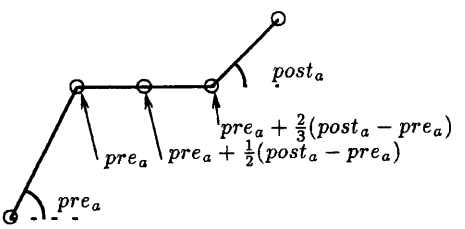

図 6: 同一特徴量を連続使用した場合の角度算出法

補完の係数は現時点(k)までの連続使用回数がqのときは、$\frac{q-1}{q}$を利用する。

これを使うと、式6となる。

$$\left.\begin{aligned} d''(k,l_a,l_b) &= |r''_a - r''_b| \\ r''_a &= pre_a + \frac{q_a - 1}{q_a} \cdot (post_a - pre_a) \\ pre_a &= tan^{-1}\left(\frac{a_{i(k)} - a_{i(l_a)}}{v}\right) \\ post_a &= tan^{-1}\left(\frac{a_{i(k)+1} - a_{i(k)}}{v}\right) \\ q_a &= k - l_a \\ r''_b &= pre_b + \frac{q_b - 1}{q_b} \cdot (post_b - pre_b) \\ pre_b &= tan^{-1}\left(\frac{b_{j(k)} - b_{j(l_b)}}{v}\right) \\ post_b &= tan^{-1}\left(\frac{b_{j(k)+1} - b_{j(k)}}{v}\right) \\ q_b &= k - l_b \end{aligned}\right\} \quad (6)$$

但し、l_aは、$i(l_a) \neq i(k), l_a < k$を満たす最大の整数であり、同様に、$l_b$は、$j(l_b) \neq j(k), l_b < k$を満たす最大の整数である。また、$q_a$および$q_b$は現在の特徴量までに同じ特徴量を連続使用した回数を示す。

特徴量間距離がkとl_a, l_bの関数として定義されるので、伸縮結果を前方にサーチしてl_aおよびl_bの値を求める必要がある。この点において、一般のＤＰマッチングのアルゴリズムとは若干異なっている。そこで、このマッチング方法を変化量対応型マッチングと呼ぶことにする。これにより、伸縮の結果生じたパターンを変化量を使って比較することが可能になる。本手法は伸縮の結果生じた変化量を用いるという点で、変化量を特徴量とするパターンのＤＰマッチングとは明らかに異なっている。

3.4 統計による重み付け

パターン間距離は特徴量間距離の荷重平均であるが、単語の特徴に合わせて適切な重み付けができれば、ＤＰマッチングの性能を向上させることができる。この重み付けは、発話内容が同一の複数の発話パターンにおいて、発話の違いによる変化が大きい部分には軽く、変化が小さい部分には重く設定すれば良い。そこで、発話内容が同一の複数の辞書用パターンに対してＤＰマッチングによる時間正規化を行なった後に、時間毎に統計処理を行ない平均値と分散値を辞書に記録しておく。そして、認識実行時に特徴量間の距離を偏差値を求めるのと類似した方法で換算することで、重み付けを行なった。

以下では、辞書作成の方法について述べる。

まず、複数の辞書用パターンから１つの代表パターンを選出し、残りを比較パターンとする。代表パターンは、全パターンの中で標準的なパターンを

選ぶことが望ましい。今回は、全てのパターンを相互にＤＰマッチングを行ない、他の全てのパターンとの距離の分散と平均が小さいパターンを代表とした。

次に、代表パターンと比較パターンを代表パターンを固定して非対称形[1]でＤＰマッチングを行なうことで、全てのパターンの時間軸を代表パターンの時間軸を使って正規化する。その上で時間毎に特徴量間の距離を集計し、平均値と分散値を求める。

ここで、代表パターンをT、Tと同じ発話内容の比較パターンを$C_1, C_2, \ldots, C_n, \ldots, C_N$とすると、パターンは特徴量の時系列なので、

$$T = a_1, a_2, \ldots, a_i, \ldots, a_I$$
$$C_1 = b_{11}, b_{12}, \ldots, b_{1i}, \ldots, b_{1J_1}$$
$$\vdots$$
$$C_n = b_{n1}, b_{n2}, \ldots, b_{ni}, \ldots, b_{nJ_n}$$
$$\vdots$$
$$C_N = b_{N1}, b_{N2}, \ldots, b_{Ni}, \ldots, b_{NJ_N}$$

今、TとC_nが最もマッチしたときの時間変換関数F_nを、$F_n = c_n(1), c_n(2), \ldots c_n(i), \ldots c_n(I)$とする。同様に最もマッチしたときの$C_n$の伸縮結果を、$j_n(1), j_n(2), \ldots j_n(i), \ldots j_n(I)$とする。

このとき、辞書パターンTと比較パターンC_nの距離$D(T, C_n)$は、非対称形なので式7で表せる。

$$D(T, C_n) = \frac{\sum_{i=1}^{I} d(c_n(i))}{I} \qquad D(T, C_n) = \frac{\sum_{i=1}^{K} d(i, j_n(i))}{I} \tag{7}$$

式7を展開すると、

$$D(T, C_1) = \frac{d(c_1(1))}{I} + \ldots + \frac{d(c_1(i))}{I} + \ldots + \frac{d(c_1(I))}{I}$$
$$D(T, C_2) = \frac{d(c_2(1))}{I} + \ldots + \frac{d(c_2(i))}{I} + \ldots + \frac{d(c_2(I))}{I}$$
$$\vdots$$
$$D(T, C_n) = \frac{d(c_n(1))}{I} + \ldots + \frac{d(c_n(i))}{I} + \ldots + \frac{d(c_n(I))}{I}$$
$$\vdots$$
$$D(T, C_N) = \frac{d(c_N(1))}{I} + \ldots + \frac{d(c_N(i))}{I} + \ldots + \frac{d(c_N(I))}{I}$$

従って、時間iの特徴量間距離の平均をμ_i、分散をσ_iとすると、それぞれ式8、および、式9で表される。

$$\mu_i = \frac{\sum_{n=1}^{N} \frac{d(c_n(i))}{I}}{N} \tag{8}$$

$$\sigma_i = \frac{\sum_{n=1}^{N} \left(\mu_i - \frac{d(c_n(i))}{I}\right)^2}{N-1} \tag{9}$$

これらの$(\mu_1, \mu_2, \ldots, \mu_I)$と、$(\sigma_1, \sigma_2, \ldots, \sigma_I)$を辞書に記録しておく。

認識の際は、まず従来の特徴量間の距離を算出した後に、先のμ_iとσ_iを使って換算する。認識時の特徴量間の距離を$d(c(i))$とすると、実際に利用する距離dl_iは式10で表される。

$$dl' = 9.0 \times \frac{d(c(i)) - \mu_i}{3\sigma_i} + 1.0$$
$$dl_i = \begin{cases} 0.0 & if \quad dl' < 0.0 \\ 100.0 & if \quad dl' > 10.0 \\ dl'^2 & otherwise \end{cases} \tag{10}$$

3.5 開き具合のスコアとの統合

ＤＰマッチングによって特徴量のグラフの形の類似性が定義できた。これらは、特徴量の変化量に基づくので、グラフ全体が上下に並行移動した場合は区別がつかない。しかし、両唇音や母音の発音に関して考えれば、口の開き具合そのものも非常に重要な要素であり無視できない。そこで、ＤＰマッチングによるパターン間距離とは別に、口の開き具合が適切であるかを得点化し、これを開き具合のスコアとする。方法としては、重み付けと同様にＤＰマッチングによる時間正規化と統計処理を使う。つまり、辞書作成用パターンから正規化した時間での特徴量の平均値と分散値を求めておき、認識時に認識パターンの特徴量から換算する。今回は、辞書パターンにおいて極大値と極小値をとる時間をあらかじめ調べておき、このときの換算値の平均を認識パターンの開き具合のスコアとした。

従って、時間iの特徴量の平均をm_i、分散をs_iとすると、それぞれ式11、および、式12で表される。

$$m_i = \frac{\sum_{n=1}^{N} b_{j_n(i)}}{N} \tag{11}$$

[1] 文献[3]を参照

$$s_i = \frac{\sum_{n=1}^{N}(m_i - b_{j_n(i)})^2}{N-1} \qquad (12)$$

これらの $(m_1, m_2, \ldots, m_I)$ と、$(s_1, s_2, \ldots, s_I)$ を辞書に記録しておく。

従って、認識時の時間 i における換算値 dv_i は、認識パターンの時間正規化後の対応特徴量を $b_{j(i)}$ とすると式13で表される。

$$dv' = 10.0 \times \frac{|b_{j(i)} - m_i|}{3s_i}$$

$$dv_i = \begin{cases} 100.0 & if \quad dv' > 10.0 \\ dv'^2 & otherwise \end{cases} \qquad (13)$$

今、辞書パターンに W 個の極大値、および、極小値が存在し、辞書パターンにおける極大値、および、極小値をとる時間の添字を $g(1), g(2), \ldots, g(W)$ とすると、口の開き具合のスコア S は式14で表される。

$$S = \frac{\sum_{n=1}^{W} dv_{g(n)}}{W} \qquad (14)$$

パターン間の距離 $D(T,R)$ とこの口の開き具合のスコア S を統合して、一つの尺度にする必要がある。今回は単純に式15の様にした。

$$score = D(T,R) + K \cdot S \qquad (15)$$

ここで、K は統合のための係数である。

4 認識性能の評価

認識実験には以下の4つのパラメータを変化させて認識率を算出した。

特徴量 2.1章で述べた特徴量に関して、*height* と *width* の両方を用いる場合と、*height* のみを用いる場合とを比較した。

補完数 3.1章で述べた補完数に関して、補完数を0～4に変化させた。

マッチング方法 以下の2種類の方法を試した。

- **ノーマル** 3.2章で述べた、式4を特徴量間距離として用いる、一般のＤＰマッチング
- **変化量対応** 3.3章で述べた、式6を特徴量間距離とする、変化量対応型マッチング

辞書および計算方法 以下の3種類の方法を試した。

- **単一パターン** 辞書として、1つの辞書用パターンだけを用いる方法
- **角度のみ** 3.4章で述べた、統計による重み付けを用いる方法
- **角度＋開き具合** 3.5章で述べた、口の開き具合の得点と統合したスコアを用いる方法 (統合係数 $k = 4.0$)

2.2章でも述べたが、1日目のデータ (30単語×約20回) から辞書を作成し、2日目のデータ (30単語×約20回) を認識させてみた。認識実験の結果を表1に示す。

これによると、補完数に関しては2にするのが最も良い結果になっている。但し、マッチングの際のパラメータなどによって変化する可能性はある。

次に、使用する特徴量についてだが、*height* と *width* の両方を使用する方が *height* のみ使用する場合より少し良い結果が出ているが、期待したほどには効果が上がっていない。

マッチング方法に関しては、わずかではあるが、変化量対応のマッチング方法の方が認識率が高いことがわかる。また、辞書および計算方法に関しても、それぞれ効果が上がっていることがわかる。この中で、最も良い認識率を得られる「角度＋開き具合」の場合をグラフ化した (図7)。これによると、変化量対応型マッチングを利用し補完を行なえば、*height* だけでも *height* と *width* の両方を特徴量として用いる場合にかなり近い認識率を得ることが可能であることがわかる。

5 デモシステムの構築

以上の結果を踏まえてリアルタイムのデモシステムを構築した。システムの構成図を図8に示す。図の様に2台のコンピュータを使い、1台で画像のキャプチャリングと特徴点の追跡を行ない、他のもう1台で辞書とのマッチングを行なう。

認識性能評価の際の手法には手作業が部分的に含まれていたので、以下の点に関して変更した。

特徴量の抽出 特徴量としては、*height*(縦方法の開き具合) のみを利用する。

変化量対応　　母数:611

	height & width						height only					
	単一パターン		角度のみ		角度+開き具合		単一パターン		角度のみ		角度+開き具合	
0	508	83.14%	470	76.92%	506	82.82%	344	56.30%	346	56.63%	405	66.28%
補 1	564	92.31%	579	94.76%	594	97.22%	531	86.91%	556	91.00%	594	97.22%
完 2	566	92.64%	589	96.40%	606	99.18%	549	89.85%	573	93.78%	597	97.71%
数 3	564	92.31%	587	96.07%	602	98.53%	564	92.31%	576	94.27%	594	97.22%
4	564	92.31%	585	95.74%	599	98.04%	570	93.29%	570	93.29%	590	96.56%

ノーマル　　母数:611

	height & width						height only					
	単一パターン		角度のみ		角度+開き具合		単一パターン		角度のみ		角度+開き具合	
0	493	80.69%	448	73.32%	476	77.91%	379	62.03%	292	47.79%	364	59.57%
補 1	554	90.67%	572	93.62%	589	96.40%	504	82.49%	522	85.43%	576	94.27%
完 2	566	92.64%	585	95.74%	602	98.53%	544	89.03%	544	89.03%	571	93.45%
数 3	566	92.64%	575	94.11%	600	98.20%	548	89.69%	548	85.76%	559	91.49%
4	563	92.14%	558	91.33%	591	96.73%	551	90.18%	526	86.09%	550	90.02%

表 1: 認識実験の結果 (正答数および認識率)

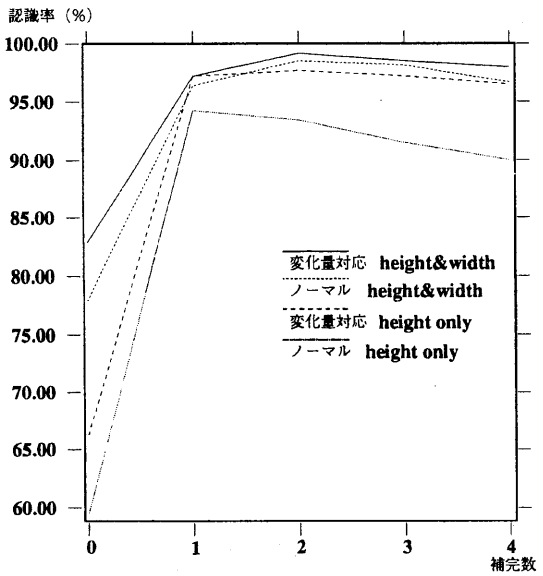

図 7: マッチング方法と特徴量による認識率の変化

特徴点は、顔の中心線上だけを動くものとして、最初に顔の中心線上に多数の追跡点を設定し、しばらく追跡を行ない距離変化が最も大きい2つの追跡点を選択することによって唇の動きを抽出する。(詳しいアルゴリズムは [4] を参照) これにより、手作業による特徴点の開始位置の設定をなくした。

口を閉じた時の特徴量 口を閉じた時の特徴量 Hc を求めるには、追跡を開始した後に口の開閉を数回繰り返してもらい、その間の特徴量の最小の値を Hc とする。

目の間の距離 目の位置を自動で検出することにより、手作業による指定を省くようにした。

ワードスポッティング化 マッチングの際に、始終端をフリーにしワードスポッティングを行なうようにした。これにより、手作業による発話区間の設定をなくした。ワードスポッティングによる認識性能は現在のところ 90% 程度である。

デモシステムとしてのトータルの認識性能テストは実施していない。キャプチャリングの性能が 26 ～ 28 フレーム / 秒程度であるので、この原因による能力低下が予想されるが、使用した際の実感としては 9 割程度である。

両唇音数＝0		両唇音数＝1				両唇音数＝2	
あいうえお	AaaAa	第1音節		第4音節		もらいもの	cAaca
きたきつね	aAaaA	ばってりー	CQAa:	あせんぶる	AANca	つまみぐい	aCcaa
きーのーと	a:a:a	もらいなき	cAaAa	らいとぺん	AaaCN	かぶとむし	Acaca
とりのこし	aaaaa	第2音節		てんこもり	ANaca	くるまえび	aaCAc
		あめあられ	ACAAA	ゆーとぴあ	a:acA	ありのまま	AaaCC
		にまいがい	aCaAa	いじっぱり	aaQCa	うずらまめ	aaACC
		しめすへん	aCaAN	おひざもと	aaAca	ひとつまみ	aaaCc
		第3音節		第5音節		両唇音数＝3	
		はねむーん	AAc:N	あかとんぼ	AAaNc	うみつばめ	acaCC
		あんぱいや	ANCaA	わるだくみ	AaAac	まごむすめ	CacaC
				こてしらべ	aAaAC		
				ころしあむ	aaaAc		

コード表記

	母音 ('a','e')	母音 ('i','o','u')
両唇音 ('b','m','p')	C	c
両唇音以外の子音	A	a

その他	
長音 ー	:
促音 っ	Q
撥音 ん	N

表 2: 認識単語の両唇音による分類とコード表記

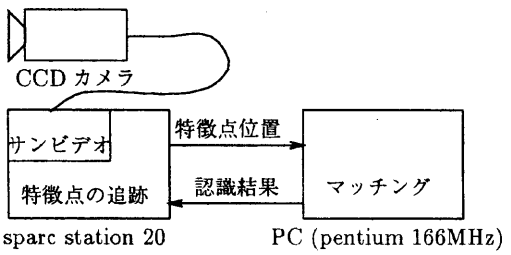

図 8: デモシステムの構成図

6 おわりに

唇の動き情報をそのままＤＰマッチングすることによって、特定話者の単語認識を行なった。今回認識に使用した単語の一覧を表2に掲載する。これを見るとわかるように、問題としてはかなり易しいものであったが、この実験により以下のことがわかったと言える。

- 我々の扱った唇の動き情報の抽出方法が有効であること
- 唇の動きをマッチングするのに、変化量対応型マッチングが有効であること

今後は、これらの知見に基づいて動き情報と音声情報との融合による、周囲雑音に影響されにくい音声認識システムの構成方法を探っていく予定である。

参考文献

[1] 宮崎敏彦, 奥村晃弘, 藤井明宏, 岡野健治,: “騒音環境下での音声理解のための唇認識と音声認識”, 情処研報,96-SLP-12,pp.97-102(1996)

[2] 松岡清利, 古谷忠義, 黒須顕二:” 画像処理による読唇の試み －母音口形の識別およびそれに基づく単語認識－”, 計測自動制御学会論文集 Vol.22,No.2,pp.67-74(1986)

[3] 中川聖一: “確率モデルによる音声認識”, 電子情報通信学会 (1988)

[4] 岡野健治, 宮崎敏彦, 奥村晃弘, 藤井明宏: “動き情報を用いた唇の抽出法”, 情処研報,96-CV-98, pp.13-18(1996)