音声認識における音響モデル

音素表記は、できるだけ実際の発音に忠実に。
「京都」は正書法では「きょーと (ky o- t o)」のように記述。
格助詞「は」「へ」も要注意。
「日本（にっぽん／にほん）」のように複数の読みを持つ場合は、
各々を記述。

このようにして、単語列W={w1,w2,…wk}が音素列{m1,m2,…ml}
に展開されるので、p(X|W)は以下のように計算できる。

$$p(X \mid W) = \prod_i p(x \mid m_i) \quad \ldots(3)$$

ここでp(x|mi)は、音素単位の音響的特徴を表現したHMM
入力音声（の一部）x とマッチングすることにより計算。

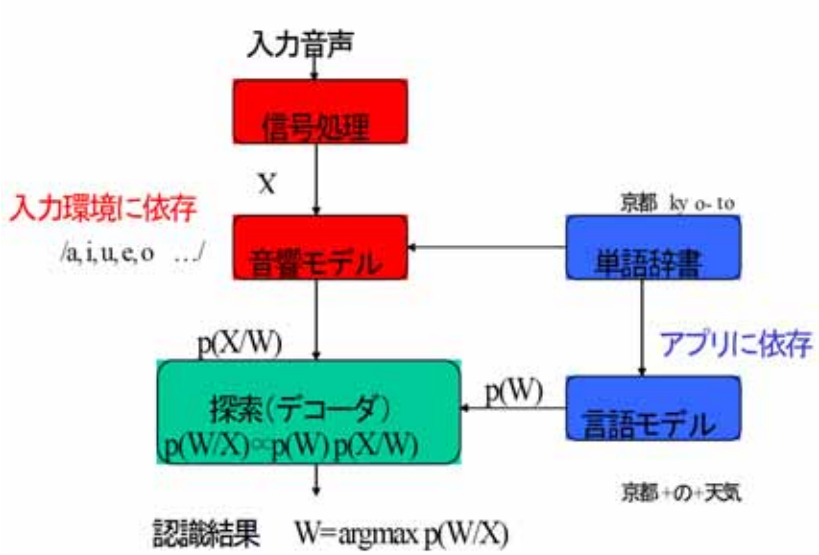
音声認識の基本

音声認識は、入力音声Xに対する事後確率p(W|X)が最大となる
単語列Wを見つける問題として定式化できる。
p(W|X)を直接計算することは困難であるので、ベイズ則
により以下のように書き換える。

$$p(W \mid X) = \frac{p(W) * p(X \mid W)}{p(X)} \quad \ldots(1)$$

この分母は、Wの決定に影響しないので、無視ができる。
単純なパターン認識（数字の認識など）では、W の事前確率p(W)を
等しいと仮定することができ、その場合はp(X|W)で決定される。
連続音声認識においてはp(W)も大きく関与する。

音声認識全体の基本



音声認識における音響モデル

音素表記は、できるだけ実際の発音に忠実に。
「京都」は正書法では「きょーと (ky o- t o)」のように記述。
格助詞「は」「へ」も要注意。
「日本 (にっぽん／にほん)」のように複数の読みを持つ場合は、各々を記述。

このようにして、単語列W={w1,w2,…wk}が音素列{m1,m2,…ml}に展開されるので、p(X|W)は以下のように計算できる。

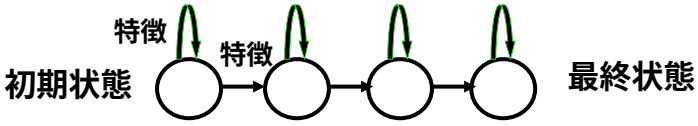
$$p(X \mid W) = \prod_i p(x \mid m_i) \quad \ldots(3)$$

ここで$p(x|m_i)$は、音素単位の音響的特徴を表現したHMM
入力音声 (の一部) x とマッチングすることにより計算。

音声認識における音響モデル

p(X|W)は単語列W から音声のパターンX が生起する確率。
音響的なモデルによるマッチングに基づく。
時系列を柔軟に扱えるHMM(Hidden Markov Model)が主流。

モデルの単位（状態）は、音素（ローマ字1文字にほぼ相当）を用いる。
単語と音素表記（もしくはかな表記）の対応づけは単語辞書で記述。



音素の音響的特徴は前後の音素によって変動する。
前後の音素に応じてテンプレートを用意するのがトライフォンモデル。
「会社（かいしゃ）」に対するトライフォンは、
「**k**+a k-**a**+i a-**i**+sh i-**sh**+ash-**a**」。

音声認識における言語モデル

p(W)はある単語列W が生起する確率。言語的な確からしさ。

ディクテーションでは、使用される単語の統計量に基づいて推定。
限定タスクの場合は、文法的･意味的に正しくないものの確率を0 とすることにより、認識の候補を絞り込む。

言語モデルは、統計的なモデルに基づくものと、
決定的な記述文法に基づくものに大別できる。
音声認識では、先頭の単語から逐次的に言語モデルを適用。
単語列W={w1,w2,…wk} (wiは各単語)に対して、

$$p(W) = \prod_i p(w_i \mid w_1 \ldots w_{i-1}) \ldots (2)$$

音声認識における言語モデル

統計モデルの場合、p(wi|w1…wi-1)を直近のN単語連鎖
p(wi|wi-N+1,..wi-1)でモデル化する。
これを単語N-gramモデルと呼ぶ。
N=2（2単語連鎖）の場合がバイグラム、
N=3（3単語連鎖）の場合がトライグラムである。

単語辞書と言語モデルはアプリケーションに依存。
ホテル検索のシステムでは、それに特化したものを用意。
汎用的ディクテーション用モデルは、
日本語の大規模なテキストデータベースから構築しているので、
固有名詞がカバーされない。

Juliusの動作原理

音声認識では、式(1)を様々な単語列Wについて計算し、
最も事後確率の高いものを選択する（＝最大事後確率原理）。

$$\begin{aligned}\hat{W} &= \arg\max_{i} p(W_i \mid X) \\ &= \arg\max_{i} p(W_i) * p(X \mid W_i) \qquad \ldots(4) \\ &= \arg\max_{i} \{\log p(W_i) + \log p(X \mid W_i)\}\end{aligned}$$

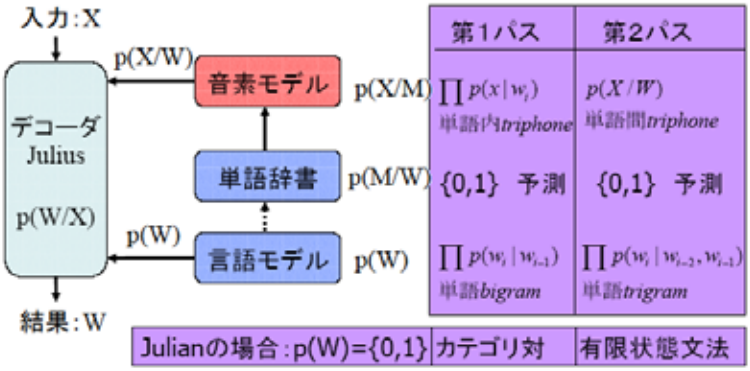
実際には、argmax の中を以下のように設定する。

$$\log p(X \mid W) + \alpha \log p(W) + \beta * N \ldots(5)$$

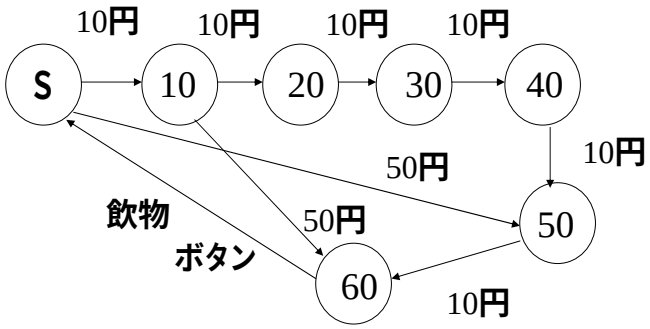
ここで、αは言語モデル重み、βは単語挿入ペナルティ、
N は仮説W に含まれる単語数（単語列の長さ）。

Juliusの動作原理

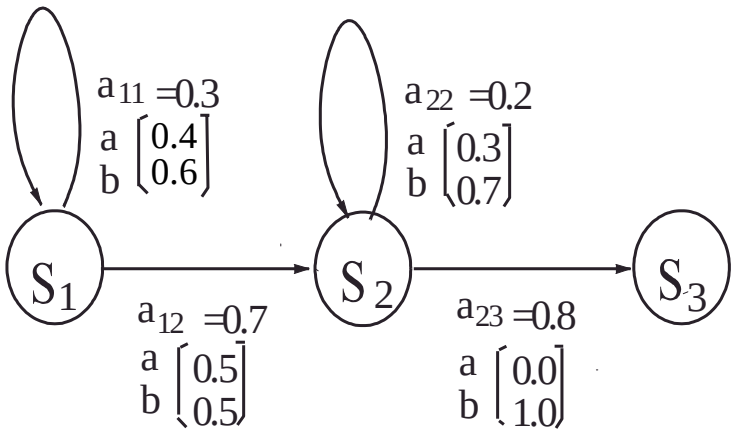
Julius は2パスで探索。
第1パスで単語バイグラムモデルを用いて荒い照合を行い、
その結果に対して第2パスで単語トライグラムモデルを適用。
音響モデルについても、第1パスでは単語間のトライフォンを厳密に適用せず、第2パスにおいて正確な尤度を計算する。



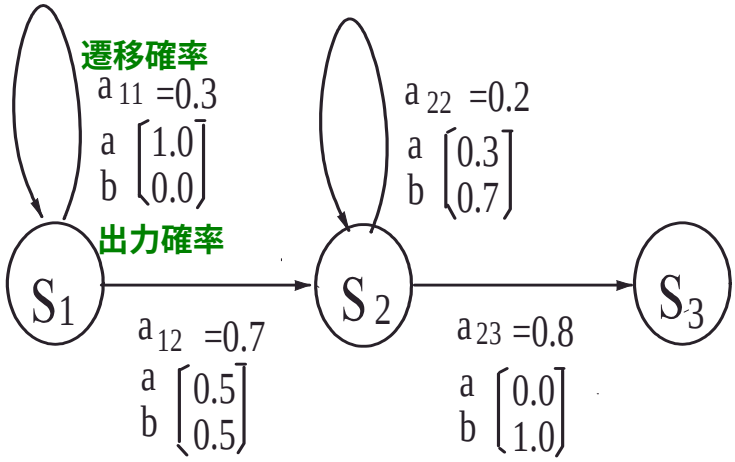
有限オートマトン



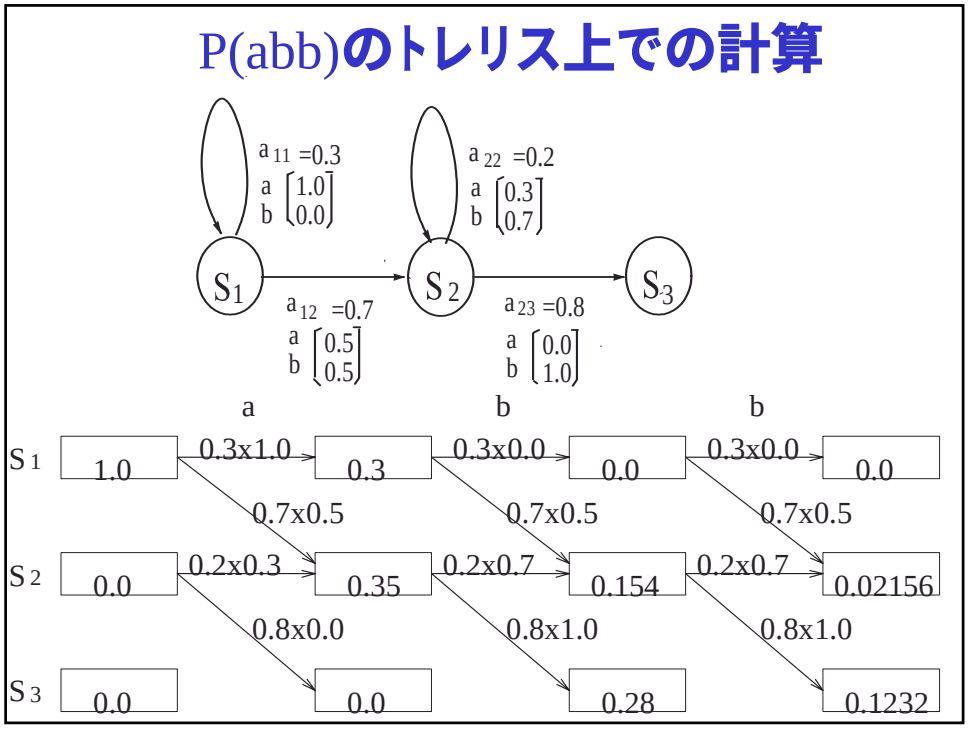
確率的オートマトン



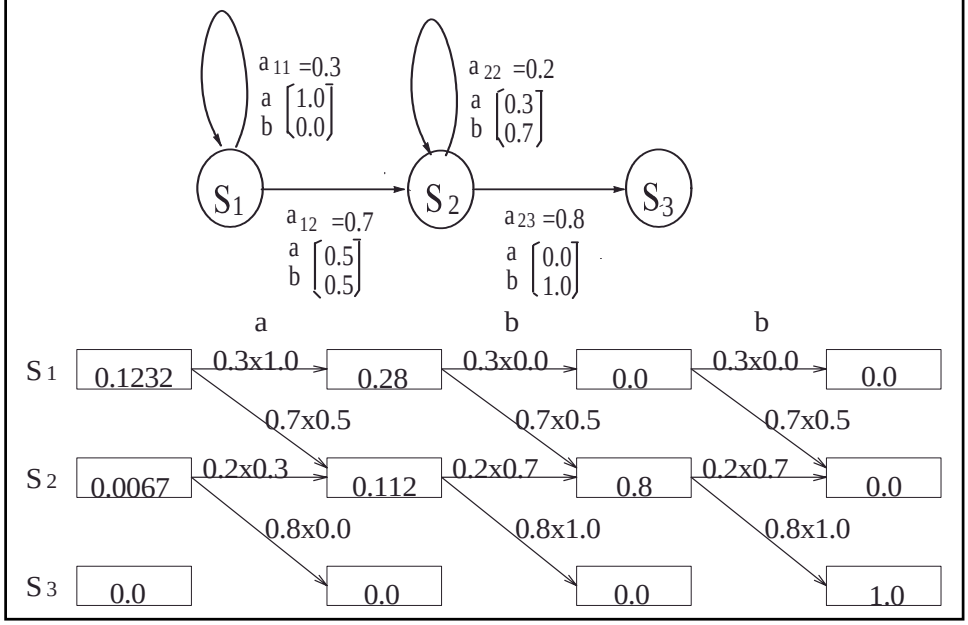
HMMの例



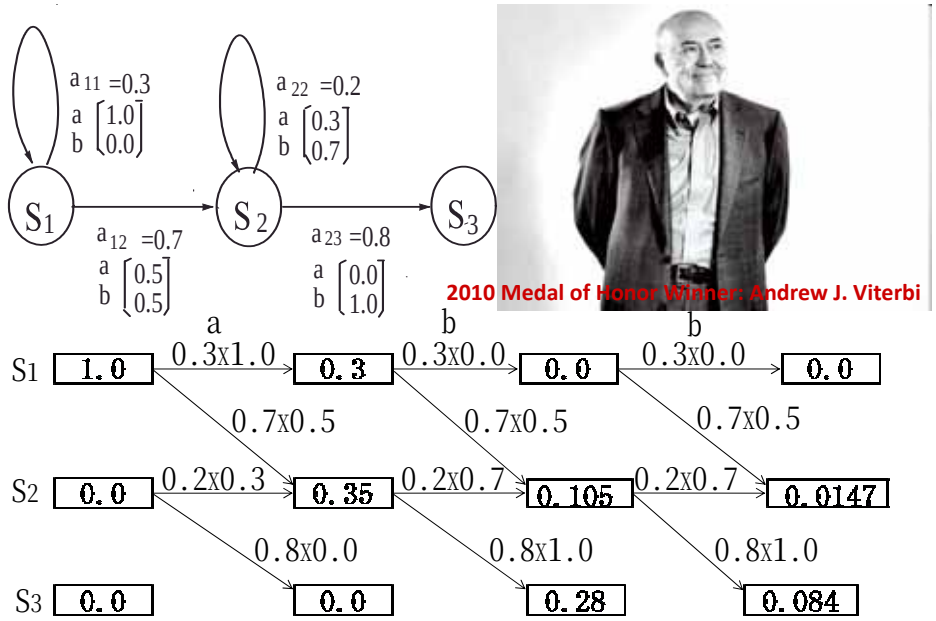
P(abb)のトレリス上での計算



P(abb)の後ろ向きアルゴリズムの計算

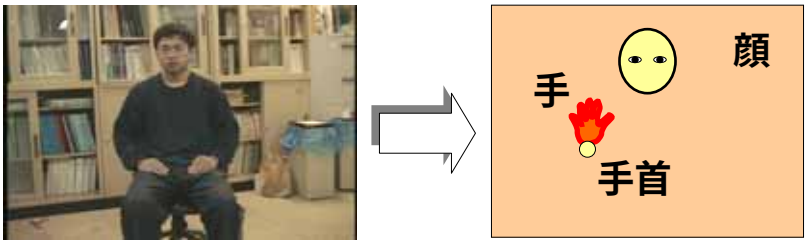


Viterbiアルゴリズムのトレリス上での計算



HMMを用いた手話単語の認識

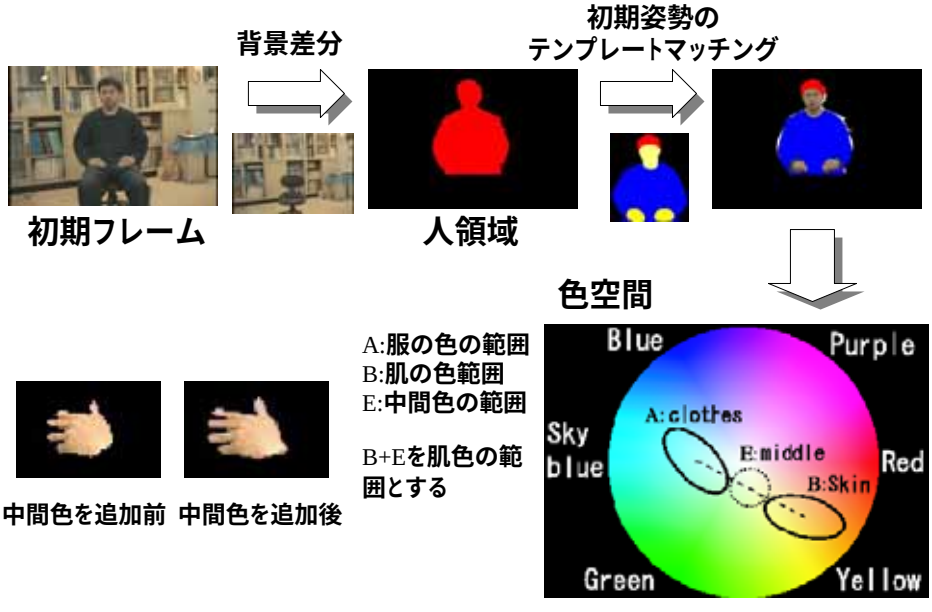
特徴量の抽出



□複雑背景下：背景に肌色に似た領域が存在する
□複数被験者：被験者によって肌色の範囲が異なる

肌色の範囲を初期フレームで決定

肌色の範囲の決定



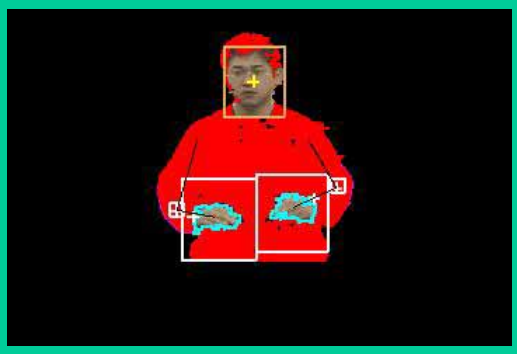
顔、手のトラッキング

初期フレーム

顔，手の初期位置の決定

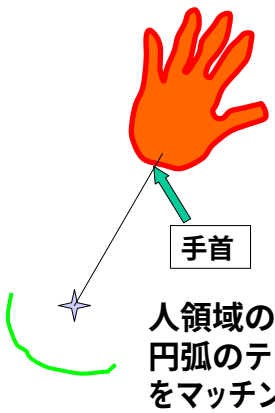
肌色の範囲の決定

その後のフレーム



顔の探索範囲：前フレームの顔の近傍
手の探索範囲：速度，加速度より予測

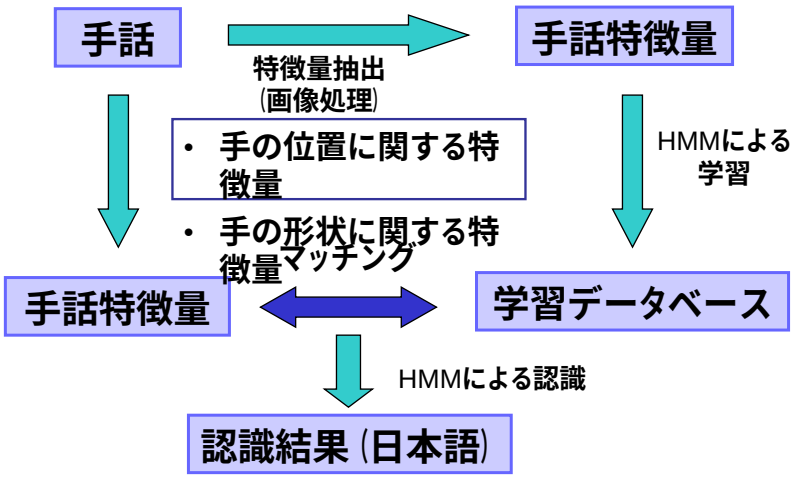
手首の位置の決定



人領域の輪郭に
円弧のテンプレート
をマッチング
して肘を見つける

処理結果

HMMによる認識の概要

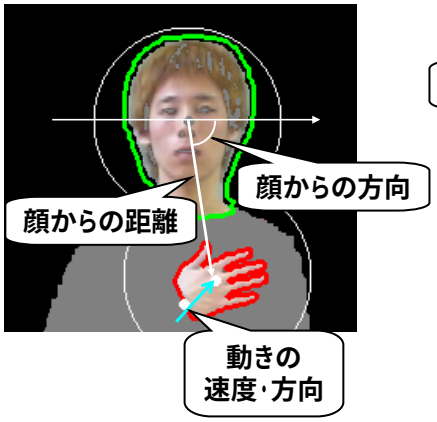


特徴量抽出の概要

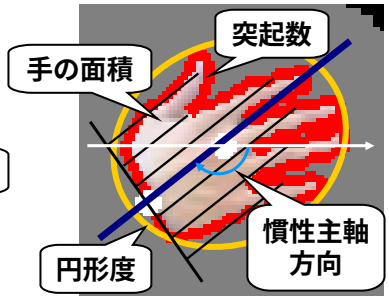
- 手話特徴量抽出のための画像処理
  - 人物領域抽出
  - 肌色領域抽出
  - 肘･手首抽出
  - 顔･手領域の追跡
  - 領域の隠蔽判定･処理

手話特徴量

位置に関する特徴量

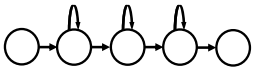


形状に関する特徴量



HMMによる学習・認識

- 認識に用いるHMMはLeft-to-Rightモデル
- 単語ごとに状態数を設定

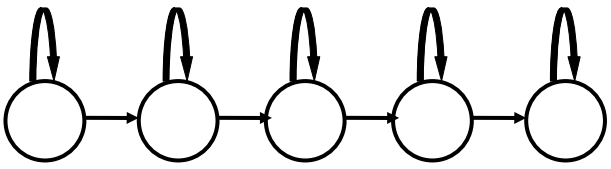


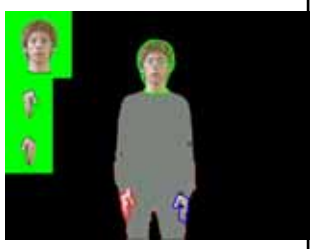
初期状態　　最終状態

手話認識のための学習（HMM）

- HMMはLeft-to-Right
- 単語ごとに状態数を設定する必要がある
- 手の移動中、静止中、手の形を変化時に対して各々状態を1つ割り付ける

状態数決定の例（状態数：5）





初期状態 移動中 静止中 移動中 最終状態