TREINAMENTO DE MODELOS DE UNIDADES FONÉTICAS COM VARIABILIDADE ACÚSTICA EM RECONHECEDORES DE VOZ CONTÍNUA BASEADOS EM CDHMM

Sidney Cerqueira Bispo dos Santos

Dep Eng Elétrica - IME - DE/3
Pça Gen Tibúrcio, 80 - Praia Vermelha
22290 - 000 Rio de Janeiro - RJ
Tel/Fax: (021) 542 09 99
E-mail: sidney@aquarius.ime.eb.br

Abraham Alcaim

CETUC – PUC/Rio
Rua Marquês de São Vicente, 225 – Gávea
22453-900 Rio de Janeiro – RJ
Tel (021) 529 92 54, 529 93 84
E-mail: alcaim@cetuc.puc-rio.br

Resumo - Este artigo descreve e analisa diversos algoritmos de treinamento dos modelos de unidades fonéticas com variabilidade acústica em reconhecimento de voz contínua. É proposto um algoritmo que soluciona o problema de descasamento entre os modelos de unidades fonéticas, representadas por CDHMM, e seus respectivos modelos acústicos. Esse descasamento ocorre ao se treinar os modelos utilizando-se métodos tradicionalmente empregados para treinamento de unidades como fones, difones e trifones.

Abstract - This paper presents several algorithms for training CDHMM–based continuous speech recognizers that use phonetic units with acoustical variability. An algorithm is proposed to solve the problem of mismatching between the phonetic unit models, represented by CDHMM, and the acoustical models. This mismatch arises when the models are trained with the traditional methods employed for subword units that do not show any significant acoustical variability, such as phones, diphones and triphones.

Palavras-chave: Reconhecimento de voz contínua, treinamento de modelos, modelos escondidos de Markov.

1. INTRODUÇÃO

As unidades mais utilizadas em reconhecimento de voz contínua, porém de difícil treinabilidade [1]-[3], são os *trifones*. A carga computacional envolvida é tão grande que só recentemente, com o avanço na tecnologia de armazenamento de dados e com o aumento na velocidade dos processadores foi que os reconhecedores modernos passaram a utilizar HMMs contínuos com trifones. No treinamento desses modelos são necessárias as estimativas das médias, covariâncias e coeficientes de cada componente da mistura de Gaussianas de cada estado. Para um sistema na língua inglesa que utilize um grande vocabulário, usualmente é necessário o treinamento de aproximadamente 60.000 modelos de trifones. Na língua portuguesa, onde existem aproximadamente 50 fones, existiriam 50^3 = 125.000 possibilidades (embora nem todos os trifones ocorram devido a restrições fonéticas da linguagem).

Como, na prática, a utilização de 5 componentes nas misturas produz um bom desempenho [4] e, assumindo que as matrizes covariâncias sejam diagonais, um reconhecedor com vetores acústicos compostos de 26 atributos necessitará estimar aproximadamente 265 parâmetros por estado, ou seja, 26x5 médias mais 26x5 variâncias mais 5 coeficientes da mistura. Utilizando-se HMMs com 3 estados para modelar cada um dos trifones teremos aproximadamente 48 milhões de parâmetros para serem estimados e armazenados. É óbvio que a utilização de um número maior de atributos ou de componentes por mistura na FPS (Função de Probabilidade de Saída) ocasionará um aumento significante nesse número de parâmetros.

O número excessivo de parâmetros e a quantidade necessária de dados para o treinamento, que nunca será suficiente o bastante para permitir uma boa estimativa de todos os contextos, são cruciais no projeto de um reconhecedor de voz. A escolha correta da unidade fonética (UF) influenciará não só o número de parâmetros a serem treinados como também a precisão da modelagem acústica.

Depois da palavra, uma das primeiras unidades fonéticas propostas para reconhecedores foi a sílaba [5]. Entretanto, experimentos com sílabas e semi-sílabas [6]-[8] mostraram que o seu desempenho era inferior ao de palavras. Em [9], um estudo dessa unidade fonética mostra que, ao contrário do resultado encontrado para a língua inglesa, a sílaba é uma unidade fonética que pode permitir um desempenho muito bom na língua portuguesa, para dicionários pequenos e médios. Entretanto, para dicionários grandes, o número de sílabas também aumenta, ao ponto de tornar o treinamento dos modelos tão problemático quanto o é para os trifones.

Por outro lado, em [10] foram propostos dois inventários de unidades fonéticas com ampla variabilidade acústica, deduzidos a partir de análises de características da língua portuguesa. Esses inventários, além de serem consistentes, são treináveis.

O treinamento de Modelos de Markov Escondidos Contínuos (CDHMM) que representam as unidades fonéticas de um sistema reconhecedor de voz contínua é uma tarefa trabalhosa e bastante sensível ao contexto em que vai ser empregado. Após a definição da estrutura do HMM, ou seja, do número de estados e do número M de componentes da mistura, esse treinamento se traduz na estimação da matriz de transição A e dos parâmetros cjm, μjm e Ujm da Função de Probabilidade de Saída (FPS), para cada unidade, dada por

$$b_j(O_t) = \sum_{m=1}^{M} c_{jm} N(O_t; \mu_{jm}, U_{jm}) \quad (1)$$

onde j é o j-ésimo estado, m é a m-ésima Gaussiana da mistura.

Existem algoritmos consagrados para o treinamento de CDHMM [1]-[3], [11]-[15]. Entretanto, esses procedimentos são bastante eficazes para unidades que possuam uma constância acústica tais como fones, difones, trifones etc. As unidades propostas nos inventários 1 e 2 [10] possuem eventos acústicos bastante variáveis pois englobam sílabas (transições CV) e fones (V, codas etc). Se esses métodos de treinamento forem aplicados diretamente nas unidades dos inventários 1 e 2, o resultado será bastante diferente do que foi proposto, como mostra a Fig. 1.

Este trabalho analisa um método de treinamento eficaz quando as unidades fonéticas possuem grande variabilidade acústica entre elas.

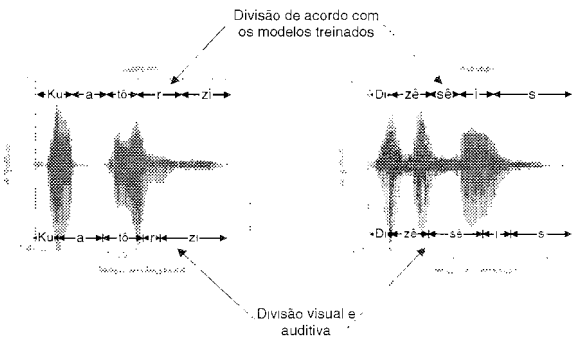

Figura 1. Segmentação das palavras quatorze e dezesseis de acordo com os modelos obtidos pelo treinamento tradicional.

A Seção 2 deste artigo apresenta um inventário reduzido de unidades fonéticas com ampla variabilidade acústica, recentemente proposto para reconhecimento de voz contínua em português [10]. Esse será o inventário usado nos experimentos descritos no artigo. A Seção 3 descreve as características básicas do sistema de reconhecimento implementado nesse trabalho. Os principais aspectos do algoritmo level-building são resumidos na Seção 4 e a incorporação da gramática é tratada na Seção 5. Na Seção 6 é apresentada uma proposta de inicialização de parâmetros mais rápida e igualmente eficaz aos métodos utilizados normalmente.

Na Seção 7 são analisados os algoritmos de inicialização dos modelos das unidades. A Seção 8 compara os desempenhos quando se utiliza o algoritmo de Viterbi e o level-building. Na Seção 9 é apresentado o algoritmo de treinamento completo e, na Seção 10 as principais conclusões.

2. INVENTÁRIO REDUZIDO DE UNIDADES FONÉTICAS

Com o objetivo de aproveitar ao máximo as vantagens que o reconhecimento silábico pode oferecer e, ao mesmo tempo, procurando reduzir o número de UFs, foram analisadas diversas características da língua portuguesa que anteriormente haviam sido consideradas no contexto da síntese de voz [16],[17]. Dentre elas, pode-se citar:

1) O português é uma língua com forte estrutura silábica do tipo CV, onde o C representa um elemento do conjunto das consoantes (ou grupamentos de consoantes - 'Brasil', por exemplo) e o V as vogais (ou grupamentos de vogais - 'mau', por exemplo) e do tipo CVC, na qual a consoante que termina a sílaba é chamada de *coda silábica*. No dialeto falado no Rio de Janeiro, tem-se 4 tipos de coda: /r/ (car-ga), /rr/ (car-ta), /s/ (cas-ta), /z/ (mes-mo);

2) nas sílabas compostas por C_1C_2V, tais como, por exemplo, 'bra' e 'pla', existe um fenômeno denominado de *epêntese*, segundo o qual, em geral, quando se intenciona pronunciar uma sílaba como 'bra', por exemplo, na verdade, pronuncia-se 'bara' (com um pequeno /a/, entre o 'b' e o 'r'). A vogal inserida entre as consoantes, conhecida como *vogal epentética*, é breve e, espectralmente, possui a qualidade do núcleo vocálico da sílaba. A Fig. 1 mostra a visualização acústica e o espectrograma deste exemplo;

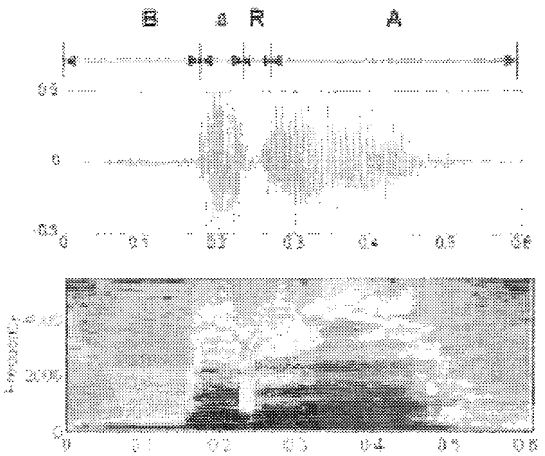

Figura 2. Sinal no tempo e espectograma da sílaba BRA.

3) numa transição *consoante-vogal nasal* a dinâmica espectral inicial é semelhante à dinâmica *consoante-vogal oral* correspondente. Isto quer dizer que a transição da sílaba 'sã', por exemplo, poderia ser aproximada pela transição 'sa' (onde a duração do a é curta), seguida do segmento 'ã', ou seja, a sílaba 'sã' pode ser representada pelo conjunto CV + $\tilde{V}$, onde $\tilde{V}$ é a vogal nasal correspondente à vogal V. A Fig. 3 mostra como exemplo a comparação entre os espectros das sílabas sã e sa, bem como suas formas de onda;

Seja C o conjunto das consoantes,

C = {b, k, d, f, g, j, l, lh, m, n, nh, p, r, rr, s, t, ch ,v, z, }..

V o conjunto das vogais orais,

V = {a, é, ê, i, ó, ô, u},

$\tilde{V}$ o conjunto das vogais nasais

$\tilde{V}$ = {ã, $\tilde{e}$, $\tilde{i}$, õ, $\tilde{u}$},

K o conjunto das codas silábicas

$$K = \{r, rr, s, z\},$$

onde 'r' simboliza a velar sonora e 'rr', a surda; ' z ' a pós-alveolar sonora e 's', a surda.

Utilizando-se as regras sugeridas anteriormente, o dicionário de unidades foi composto por unidades dos conjuntos V, $\tilde{V}$, K e das combinações CV produzindo um total de 7 + 5 + 4 + (19x7) = 149 unidades. A Tabela I apresenta o inventário completo [10].

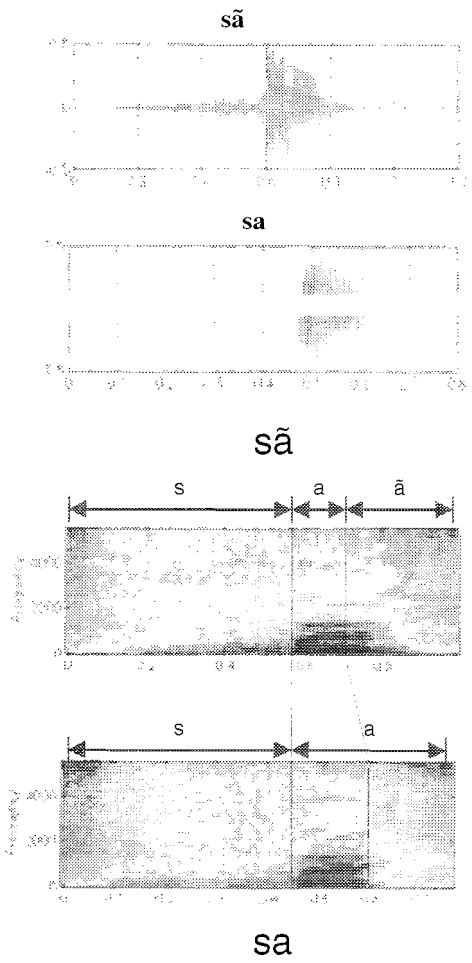

Figura 3. Comparação entre as formas de onda e os espectros das sílabas /sa/ e /sã/ .

INVENTÁRIO 1

a	é	ê	i	ó	ô	u	an	en	in
õ	un	rr	r	s	z	ba	ka	da	fa
ga	ja	la	lha	ma	na	nha	pa	ra	rra
sa	ta	va	cha	za	bé	ké	dé	fé	gué
jé	lé	lhé	mé	né	nhé	pé	ré	rré	sé
té	vé	ché	zé	bê	kê	dê	fê	guê	jê
lê	lhê	mê	nê	nhê	pê	rê	rrê	sê	tê
vê	chê	zê	bi	ki	di	fi	gui	ji	li
lhi	mi	ni	nhi	pi	ri	rri	si	ti	vi
chi	zi	bó	kó	dó	fó	guó	jó	ló	lhó
mó	nó	nhó	pó	ró	rró	só	tó	vó	chó
zó	bô	kô	dô	fô	guô	jô	lô	lhô	mô
nô	nhô	pô	rô	rrô	sô	tô	vô	chô	zô
bu	ku	du	fu	gu	ju	lu	lhu	mu	nu
nhu	pu	ru	rru	su	tu	vu	chu	zu	

Tabela 1. Inventário de unidades fonéticas.

3. SISTEMA DE RECONHECIMENTO

No sistema de reconhecimento de voz contínua implementado neste trabalho, optou-se por utilizar seqüências de 3 dígitos, por ser uma base de dados de fácil obtenção e de largo emprego, além do que, o reconhecimento de dígitos é uma das tarefas mais exigentes para um reconhecedor pois não há como utilizar gramática no nível de palavras [4].

A Fig. 4 mostra o sistema de reconhecimento utilizado e a Fig. 5 os modelos utilizados no sistema. As seqüências, os dígitos e as unidades fonéticas foram representados por modelos de Markov Escondidos Contínuos (CDHMM) [11],[12]. Cada unidade foi modelada utilizando-se três estados e modelos Bakis [11]. Os modelos dos dígitos foram representados como uma combinação dos modelos de unidades e as seqüências como uma combinação dos modelos dos dígitos.

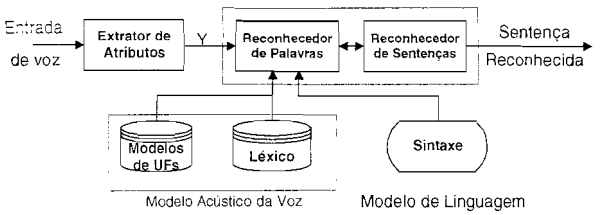

Figura 4. Sistema utilizado para o reconhecimento das seqüências de dígitos.

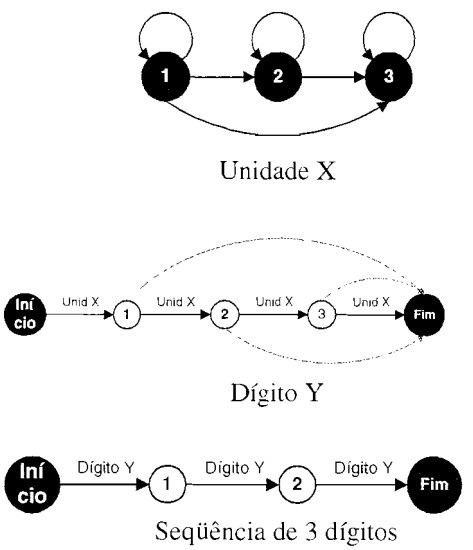

Figura 5. Estrutura dos modelos.

4. O ALGORITMO LEVEL-BUILDING (LB)

Devido à importância do algoritmo level-building (LB), e como ele servirá de base para propostas e conclusões em Seções posteriores deste artigo, nesta Seção, serão analisadas algumas de suas características.

Embora o LB tenha sido concebido para o reconhecimento de palavras conectadas, utilizando o DTW,

ele pode ser utilizado para o reconhecimento de voz contínua usando quaisquer unidades, desde que as consideremos conectadas como se fossem palavras. Entretanto, para a sua utilização com HMM são necessárias algumas adaptações, já que no DTW, os padrões de referência são representações de unidades, que possuem quadro inicial e quadro final, o que determina exatamente o início e fim de cada nível, enquanto que no HMM, os padrões de referência são modelos, compostos de probabilidades, e o início e fim de cada nível é indeterminado.

O LB representa uma tentativa de contornar o excessivo número de cálculos necessários à implementação de uma busca exaustiva em todas as possíveis concatenações das unidades do dicionário de referência na tarefa de reconhecimento.

Normalmente se impõe o paralelogramo de Itakura [11],[12] como restrição do caminho global e assume-se que existe um número máximo L de unidades na seqüência que se quer decodificar.

A Fig. 6 apresenta a grade de busca utilizada no algoritmo LB, onde o eixo t é o eixo das observações da seqüência que se quer reconhecer, $0<t\leq T$ e T é o comprimento da seqüência. O eixo J é o eixo dos níveis e estados. O número de níveis varia de 1 até o número máximo de unidades nas sentenças. O número de estados j em cada nível variará com as unidades apresentadas e poderá ser $J_1, J_2, \ldots J_k$, onde J_k é o número de estados da k-ésima unidade apresentada como hipótese.

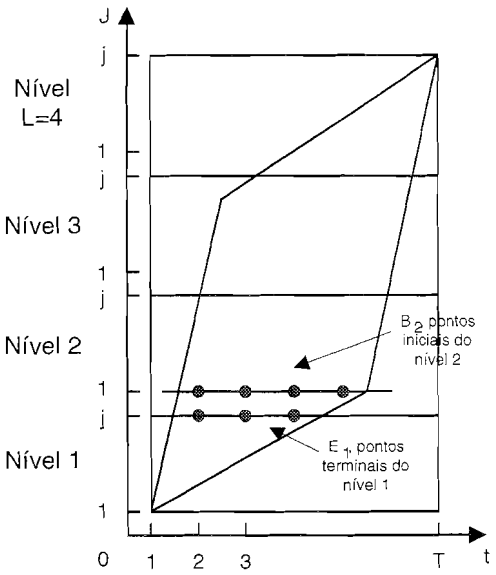

Figura 6. Grade de procura utilizada no LB.

Inicialmente divide-se o eixo de referência em L níveis e coloca-se unidades isoladas ao longo desse eixo. Inicia-se no nível 1, introduzindo-se uma unidade de referência candidata naquele nível. O algoritmo calcula o melhor caminho (seqüência de pontos com menor verossimilhança), para cada ponto terminal em E_1. O algoritmo sempre avança um passo no eixo t e calcula todas distâncias no eixo J. Ao chegar ao fim do nível 1 (último estado da última unidade apresentada) e t=T, em vez de passar para o segundo nível, insere-se a segunda unidade de referência no primeiro nível e repete-se o mesmo procedimento até que tenha se examinado todas as unidades de referência - guardando-se sempre o índice da unidade que apresentou a menor distância bem como o seu valor.

Depois que todas as unidades foram apresentadas no primeiro nível passa-se para o segundo nível e assim sucessivamente até o nível máximo L. No final, quando a última unidade for apresentada, ou seja, após chegar no último estado da última unidade e t=T, volta-se ao início procurando-se o caminho que apresentou a menor distância.

Formalmente, o algoritmo é o seguinte:

Considere os modelos de unidades individuais U_k, $1\leq k\leq K$, do dicionário e uma dada seqüência O de observações de teste O_t, t=1,2,....T. A tarefa do reconhecedor é decodificar O em uma seqüência de unidades $\{U_1, U_2, \ldots, U_P\}$, onde P é o número de palavras na sentença decodificada, que melhor descreva a seqüência de observações no sentido de que a probabilidade conjunta da seqüência de observações e da seqüência de estados seja maximizada.

No nível l=1 (nível inicial), apresenta-se o modelo $q(U_q)$, da unidade U_q, à seqüência de observações O, começando pelo quadro 1. O cálculo das distâncias é realizado utilizando-se o algoritmo de Viterbi, da seguinte forma:

1) Inicialização

$$\delta_1(1) = b_1^q(O_1). \qquad \textbf{(2a)}$$

onde $\delta_t(j)$ está relacionado à probabilidade conjunta da seqüência parcial de estados e de observações até o instante t e estado j, dadas as probabilidades de transições A e as probabilidades de saída B, ou seja, $P(O_1, O_2, \ldots, O_t, x(1), x(2), \ldots, x(t-1), x(t) = j|A, B)$, onde x(i) é o estado no instante t=i e

$$\delta_1(j) = 0, \quad j = 2, 3, \ldots, N. \qquad \textbf{(2b)}$$

2) Recursão

Para $2 \leq t \leq T$ e $1 \leq j \leq N$

$$\delta_t(j) = \max_{1\leq i\leq N}[\delta_{t-1}(i) * a_{ij}^q] * b_j^q(O_t) \qquad \textbf{(2c)}$$

3) Término

$$P(l,t,q) = \delta_t(N), \qquad 1 \leq t \leq T \qquad \textbf{(3a)}$$

$$G(l,t,q) = 0. \qquad \textbf{(3b)}$$

Quando todas as palavras tiverem sido apresentadas no nível 1, deverão ser salvos os seguintes valores:

$$P_{max}(l,t) = \max_{q} [P(l, t, q)] \qquad \textbf{(4a)}$$

$$G_{max}(l,t) = G[l, t, \arg\max_{q} P(l, t, q)] \quad (4b)$$

$$W_{max}(l,t) = \arg\max_{q} [P(l, t, q)] \quad (4c)$$

onde P_{max} é a melhor probabilidade de saída do nível, G_{max} é o ponteiro de retorno do nível (guarda o instante de início da unidade) e W_{max} é o indicador da palavra de saída, ou seja, guarda a unidade que é mais provável de ter produzido a seqüência de vetores de observações.

O cálculo para o segundo nível e os subsequentes difere somente quanto ao cálculo das distâncias no primeiro estado, pois é preciso considerar as distâncias entre-palavras, ou seja, a distância acumulada do nível anterior e as distâncias intra-palavra, ou seja, a distância local. Esse cálculo permite determinar o instante em que a palavra do nível anterior termina e a do nível atual começa.

A inicialização desses níveis é efetuada da seguinte maneira:

$$\delta_1(1) = 0 \quad (5a)$$

$$\delta_t(1) = \max [P_{max}(l-1,t-1),\ a_{11}^{q}\,\delta_{t-1}(1)].[\,b_1^{q}(O_t)], \quad 2 \le t \le T \quad (5b)$$

$$\alpha_t(1) = \begin{cases} t-1 & \text{Se } P_{max}(l-1,t-1) > \delta_{t-1}(1) * a_1^{q} \\ \alpha_{t-1}(1) & \text{De outra maneira} \end{cases} \quad (6a)$$

A Eq. (5b) escolhe a distância apropriada do nível anterior e a Eq. (6a) cria o vetor de retorno inicial que guarda o quadro em que a palavra do nível anterior termina ou que a atual começa. Durante a recursão (PASSO 2) esse vetor é atualizado da seguinte maneira:

$$\alpha_t(j) = \alpha_{t-1}\,[\arg\max_{1 \le i \le N} (\delta_{t-1}(i) * a_{ij}^{q})\,] \quad (6b)$$

e no final do nível, a distância acumulada da palavra q e o vetor de retorno se tornam:

$$P(l, t, q) = \delta_t(N) \quad 1 \le t \le T \quad (7a)$$

$$G(l, t, q) = \alpha_t(N) \quad 1 \le t \le T \quad (7b)$$

Quando todas as palavras do nível tiverem sido apresentadas, as distâncias máximas são atualizadas de acordo com as Eq. 4 e prossegue-se para o nível seguinte.

O procedimento termina quando se atinge o número máximo de níveis L e t=T. A melhor seqüência de unidades, de comprimento L e probabilidade $P_{max}(L,T)$ é obtida utilizando-se o vetor de retorno $G_{max}(l,t)$, que guarda os instantes em que as palavras com maiores probabilidades, em cada nível, começam. A melhor seqüência de unidades será aquela que possuir a maior probabilidade $P_{max}(L,T)$ sobre todos os níveis.

5. INCORPORAÇÃO DA GRAMÁTICA

Na prática, o reconhecimento de dígitos não permite a utilização de gramática, no nível de palavras, pois qualquer dígito pode vir após qualquer outro. Formalmente, diz-se que a gramática utilizada com dígitos é a gramática irrestrita ou Tipo 0, ou seja, todos os dígitos possuem a mesma probabilidade de ocorrência após qualquer seqüência de dígitos.

Por outro lado, no nível das unidades, podemos utilizar alguma gramática já que suas concatenações, dentro dos dígitos, possuem probabilidades de ocorrência diferentes.

Neste trabalho, utilizou-se gramática bigrama, calculada de acordo com a seguinte expressão:

$$p(U_2|U_1) \approx f(U_2|U_1) = \frac{c_{12}}{c_1} \quad (8)$$

onde c_{12} é o número de vezes em que a seqüência de unidades $\{U_1, U_2\}$ é observada e c_1 o número de vezes em que a unidade U_1 é observada.

As probabilidades referentes aos bigramas foram incorporadas na Eq. (2a), para o nível 1, e na Eq. (5b) e Eq. (6a) para os demais níveis. Essas equações se tornaram, respectivamente:

$$\delta_1(1) = b_1^{q(U_k)}(O_1) \cdot P(U_k|sil) \quad (9)$$

e

$$\delta_t(1) = \max [P_{max}(l-1,t-1) \cdot P(U_k|W_{max}(l-1,t-1)),\ a_{11}^{q(U_k)}\,\delta_{t-1}(1)].[\,b_1^{q(U_k)}(O_t)] \quad 2 \le t \le T \quad (10)$$

$$\alpha_t(1) = \begin{cases} t-1 & \text{Se } P_{max}(l-1,t-1) \cdot P(U_k / W_{max}(l-1,t-1)) > \delta_{t-1}(1) * a \\ \alpha_{t-1}(1) & \text{De outra maneira} \end{cases} \quad (11)$$

onde *sil* corresponde ao silêncio.

O algoritmo LB exige que se conheça *a priori* o número de unidades em cada sentença a ser reconhecida ou pelos menos que todas as seqüências possuam o mesmo número de unidades, o que é virtualmente impossível, a menos que se restrinja o dicionário de forma a obedecer a essa restrição.

Para que se possa lidar com sentenças de comprimentos diferentes ou com restrições sintáticas impostas por um outro tipo de gramática são necessárias algumas alterações na maneira de se implementar o LB - as equações permanecem as mesmas.

As modificações são as seguintes:

1) Associe cada nível com as transições entre unidades em vez de associá-los com a posição das unidades na sentença. Dessa forma, as palavras do nível l+1 não precisam mais ser contíguas às do nível l.
2) Crie mais um vetor de retorno para guardar a seqüência de níveis (estados) que será necessária para recuperar a seqüência de unidades no final do procedimento.

3) Utilize uma lista de informações, baseadas nos dados, para guiar o processo de busca. A Tabela II mostra, como exemplo, a lista de informações para as seqüências de três dígitos utilizadas neste trabalho. As informações foram extraídas da concatenação dos modelos apresentados na Seção 3 e Fig. 4. As unidades utilizadas são as constantes do inventário apresentado na Tabela I.

Estado Corrente	Unidades Usadas	Estado Pre-decessor	Nível Corrente	Nível Pre-decessor
2	6, 12, 50, 54, 63, 70, 88, 102, 114, 132	1	1	0
3	1, 4, 10, 67, 89, 90, 143	2	2	1
4	1, 4, 15, 132, 146	3	3	2
5	15, 124, 143	4, 3, 2	4	3
6	6, 12, 50, 54, 63, 70, 88, 102, 114, 132	5	5	4, 3, 2, 1
7	1, 4, 10, 67, 89, 90, 143	6	6	5
8	1, 4, 15, 132, 146	7	7	6
9	15, 124, 143	8, 7, 6	8	7
13	6, 12, 50, 54, 63, 70, 88, 102, 114, 132	9	9	8, 7, 6, 5
13	1, 4, 10, 67, 89, 90, 143	10	10	9
13	1, 4, 15, 132, 146	11	11	10
13	15, 124, 143	12, 11, 10	12	11

Tabela 2. Lista de informações utilizada no processo de reconhecimento.

6. MÉTODO DE INICIALIZAÇÃO DOS PARÂMETROS

O primeiro problema a ser resolvido para o treinamento de modelos de unidades fonéticas é o da inicialização dos parâmetros das unidades. Como o treinamento é iterativo, para o seu início é necessário a existência das probabilidades de transições iniciais e dos parâmetros iniciais da FPS. A matriz de transições é pouco sensível à escolha inicial e pode-se utilizar valores aleatórios. Porém, a convergência do treinamento é muito sensível aos valores da FPS.

Com relação a este problema, foram analisados dois métodos:

Método 1: Divida cada elocução do conjunto de treinamento igualmente pelos estados do modelo (j=1:N, onde N é o número de estados). Após a divisão, aplique o algoritmo Médias-K Modificado [4] para separar todos os vetores associados ao estado j (j=1:N) em M agrupamentos. Dentro de cada grupamento calcule os parâmetros μ_{jm}, U_{jm} e c_{jm}.

Método 2: Selecione M elocuções das K a serem utilizadas para treinar o modelo, onde M é o número de componentes da mistura. Divida os vetores da primeira elocução igualmente pelos estados (j = 1:N). Calcule, para cada estado, o vetor média (m_{j1}) , a matriz covariância (U_{j1}) e o coeficiente de ponderação (c_{j1}). Associe estes parâmetros à primeira componente da mistura. Repita esse procedimento para a segunda elocução e associe seus parâmetros à segunda componente da mistura. Continue este procedimento até a elocução M. No final deste procedimento, teremos os parâmetros iniciais da mistura $b_j(O_t)$ para cada estado.

Para os dois métodos, os parâmetros μ_{jm} e U_{jm} são a matriz média e a matriz de covariância amostral dos vetores do agrupamento m do estado j e c_{jm} é o número de vetores classificados no agrupamento m do estado j dividido pelo número de vetores no estado j.

O método 1 é o método normalmente utilizado para treinamento de voz contínua e o método 2 é uma das propostas examinadas neste artigo.

A Fig. 7 apresenta dois resultados típicos que ocorreram no treinamento. Nela, são apresentadas as verossimilhanças após a primeira e segunda iteração, para os modelos das palavras *um* e *dois* inicializadas com os métodos 1 e 2. Verifica-se que os resultados finais são aproximadamente os mesmos.

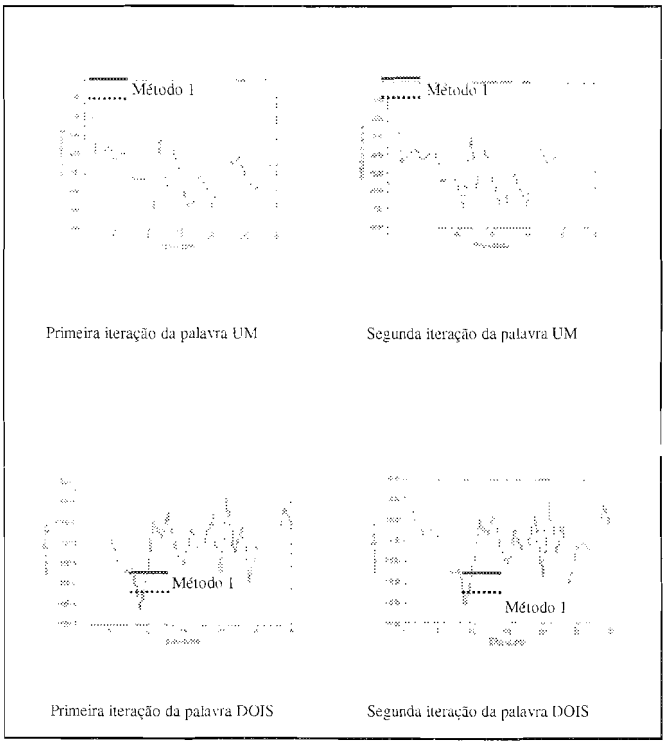

Figura 7. Comparação entre as verossimilhanças finais dos Esquemas 1 e 2, das elocuções UM e DOIS.

O método 1 começa com valores mais próximos dos valores de convergência porque utiliza o algoritmo médias-k modificado para estimar os parâmetros iniciais, entretanto, o método 2 é extremamente mais rápido.

A parte mais demorada do treinamento é a divisão dos vetores em grupos dentro dos estados. Para essa tarefa foram testados diversos algoritmos, tais como: médias-k [18], médias-k modificado utilizando pseudocentro [4], médias-k modificado utilizando minimax [4], rede neural

competitive learning [19] e Isodata [18]. O que apresentou melhor desempenho em termos de precisão e tempo foi o médias-k modificado utilizando minimax. Entretanto, a maioria desses algoritmos, exigem o cálculo da matriz de distância de um vetor para todos os demais. Considerando-se um conjunto de treinamento pequeno, da ordem de 500 repetições por unidade, o número de vetores por estado normalmente excede 5000. Desse modo, a matriz de distância de cada estado conterá da ordem de 25 milhões de posições. Se cada posição for representada por 4 bytes, somente essa matriz necessitará de 200 Mbytes de memória. Isto faz com que esta unidade leve em torno de 48 horas para ser treinada rodando em um PC Pentium de 150 MHz com 32 M de memória RAM.

7. MÉTODO PARA INICIALIZAÇÃO DOS MODELOS DAS UNIDADES

Foram analisados três esquemas de inicialização das unidades (utilizou-se apenas as necessárias para o reconhecimento de dígitos):

Esquema 1: Segmenta-se palavras em fones utilizando-se o algoritmo level-building e treina-se os modelos como apresentado em [13]. Após o treinamento, monta-se os modelos das unidades dos inventários pela concatenação dos modelos dos fones que a constituem (neste estágio os modelos das unidades terão número de estados diferentes). Segmenta-se novamente as palavras utilizando-se os modelos concatenados e treina-se os modelos das unidades, agora todas com o mesmo número de estados.

Esquema 2: Utilizando-se gravações de palavras monossilábicas ou dissilábicas, tais como as apresentadas na Tabela III, realiza-se o treinamento como apresentado em [11], considerando-se as palavras como concatenação de unidades. Em seguida, separa-se os modelos e faz-se aNN = 1.

Esquema 3: Grava-se as unidades individualmente e faz-se o treinamento como apresentado em [11]. Para as codas, utiliza-se as palavras MAS, MÊS, MAR e CARGA considerando-se as palavras como concatenação de fones. Após o treinamento separa-se os modelos dos fones correspondentes às codas.

Rei	Uns	Ôco	Toró
Tia	Teve	Céu	Seu
Fico	Nós	Toco	Couve
Dolo	Unte	Zero	Cito
Indo	Dorê	Morou	Ua

Tabela 3. Palavras utilizadas para inicialização das unidades no Esquema 2.

Após a inicialização dos modelos das unidades, deve-se fazer o treinamento utilizando-se palavras e frases para que os HMMs possam incorporar a coarticulação existente na fala contínua.

As elocuções referentes aos dígitos ou seqüência de dígitos são segmentadas em unidades utilizando-se o algoritmo level-building [11]. Os arquivos segmentados, em conjunto com os utilizados na etapa anterior, são utilizados para novo treinamento e assim sucessivamente até a convergência do modelo. Assume-se que o modelo convergiu quando a soma das verossimilhanças do conjunto de elocuções, utilizadas para treinar o modelo, pára de aumentar.

A Tabela IV apresenta os resultados obtidos para os três esquemas. No treinamento utilizou-se 30 repetições de cada dígito e 40 seqüências de três dígitos, pronunciados por um mesmo locutor, e na fase de teste, utilizou-se outras 40 seqüências de três dígitos, pronunciadas pelo mesmo locutor.

	D	*I*	*S*	*Total*	*Taxa de Acerto*
Esquema 1	48	5	25	78	92,01%
Esquema 2	36	2	10	48	95,08%
Esquema 3	33	2	0	35	96,31%

D = Deleções I = Inserções S = Substituições

Tabela 4. Resultado do teste em seqüências de três dígitos.

Os resultados mostram que o Esquema 3 é o que apresenta melhor resultado na inicialização dos modelos das unidades.

Utilizando-se o procedimento que apresentou o melhor resultado, Esquema 3, foram realizados mais alguns testes com relação à segmentação das palavras e frases, em unidades.

8. ALGORITMO DE VITERBI X LEVEL BUILDING

Os testes anteriores foram efetuados utilizando-se o algoritmo Level-Building para a segmentação. Entretanto, o algoritmo de Viterbi pode ser usado para essa tarefa, utilizando-se apropriadamente a seqüência de estados ótima fornecida. O único problema é: como concatenar as matrizes de transição dos HMMs das unidades para que componham um HMM de palavra?. A Fig. 8 apresenta um exemplo da concatenação das matrizes de transição A das unidades que compõem a palavra do dígito 2 (/dô/ /i/ /s/).

Denominando a unidade /dô/ de unidade 1, a unidade /i/ de 2 e a unidade /s/ de 3, vemos que o problema aparece na transição do estado a133 da unidade 1 para o início da unidade 2 e na transição do estado a233 da unidade 2 para o início da unidade 3.

Sejam X e Y os valores que as probabilidades aNN e aNN+1, respectivamente, devem assumir no modelo concatenado. Através de exames visuais e auditivos,

inicialmente foram analisados e selecionados os valores de 0,77 a 0,93 para X e de 0,07 a 0,23 para Y, onde X+Y = 1. Esses valores foram obtidos partindo-se inicialmente de X=1 e Y=0 e sucessivamente diminuindo 0.1 de X e aumentando o mesmo valor em Y até que não se conseguiu mais constatar qualquer diferença na segmentação resultante, em relação à segmentação real. O mesmo procedimento foi adotado partindo-se de X=0.5 e Y=0.5 e aumentando-se X com passos de 0.1 e diminuindo Y do mesmo valor. Depois, através de treinamento e teste com o mesmo conjunto de dígitos e seqüências utilizados anteriormente, foram selecionados os valores que apresentaram melhor desempenho: X=0,8 e Y=0.2.

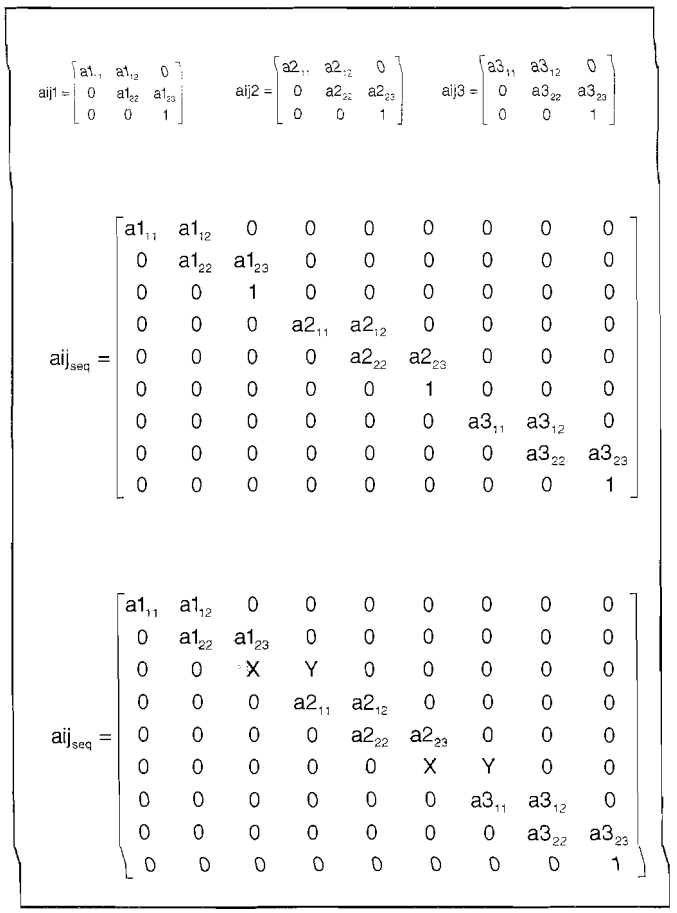

Figura 8 - Concatenação das matrizes de transição A das unidades para composição de palavras e seqüências.

A Tabela V apresenta o resultado do teste para o treinamento utilizando segmentação com o algoritmo *Level-Building* e com o algoritmo de Viterbi, utilizando-se os valores X=0,8 e Y=0,2.

Os resultados apresentados nessa tabela permitem verificar que o treinamento efetuado utilizando-se o algoritmo de Viterbi produz um resultado melhor no reconhecimento. Isto acontece, provavelmente, porque na junção das matrizes de transições A das unidades para formar as matrizes A das palavras ou seqüências, indiretamente se leva em consideração a coarticulação que existe entre as unidades ao se associar valores a X e Y, como mostrado na Fig. 8.

Estudou-se, também, um outro método de treinamento em que a segmentação seria desnecessária. Os parâmetros iniciais das unidades foram inicializados de acordo com o Esquema 3. Em seguida montava-se as matrizes das palavras ou seqüência de palavras de acordo com o método apresentado para treinamento com o algoritmo de Viterbi. Entretanto, no lugar de se fazer a segmentação e treinar as unidades como palavras isoladas, treinou-se palavras e seqüências de palavras. Após a convergência, os modelos das palavras ou seqüências eram desmembrados nos modelos das unidades, fazendo-se as componentes referentes a a_{NN} igual a 1.

	D	I	S	Total	Taxa de Acerto
Level-	33	2	0	35	96,31 %
1.1 Buildin g					
Viterbi	4	1	1	6	99,39 %

D = Deleções I = Inserções S = Substituições

Tabela 5. Resultado do teste com seqüências de três dígitos utilizando os algoritmos Level-Building e Viterbi.

Esse método se mostrou excelente para o reconhecimento das palavras e seqüências utilizadas no treinamento, chegando a 100% de acerto. Entretanto, para as seqüências de teste, o resultado foi bem pior que o obtido para o treinamento utilizando o *Level-Building*.

9. ALGORITMO PARA TREINAMENTO DAS UNIDADES

O algoritmo proposto para treinamento dos modelos de unidades que possuem grande variabilidade acústica pode ser resumido no seguinte:

INICIALIZAÇÃO

a) Segmente M gravações de cada unidade, exceto as codas, uniformemente pelo número de estados N.
b) Para cada estado j de cada unidade e de cada gravação calcule o vetor média amostral e a matriz covariância amostral. Associe os valores do vetor média e matriz covariância da gravação m de cada unidade à m-ésima componente da mistura do j-ésimo estado. Armazene o número de vetores por estado.
c) Calcule o m-ésimo coeficiente da mistura do j-ésimo estado, de cada unidade, dividindo o número de vetores existente em cada estado da m-ésima locução pelo número total de vetores existente no j-ésimo estado das M locuções.
d) Inicialize a matriz A, de cada unidade, com a distribuição uniforme.
e) Treine cada unidade como se fosse palavra isolada utilizando o algoritmo Modified Segmental K-means

até a convergência. Esses serão os modelos iniciais das unidades.

f) Para inicialização das codas, utilize repetições de palavras monossilábicas tais como mar, mas, mês etc. Considere essas palavras como se fossem unidades compostas de fones com três estados por fone e realize os passos b), c) e d), onde o número de estados será 3 vezes o número de fones.

g) Treine as unidades conectadas como se fossem palavras isoladas utilizando o algoritmo Modified Segmental K-means, até a convergência.

h) Separe os parâmetros referentes aos três últimos estados dos modelos conectados do passo f). Esses parâmetros serão as estimativas iniciais dos modelos das codas.

TREINAMENTO COM PALAVRAS[1]

i) Utilize os modelos estimados para segmentar K repetições de cada palavra em unidades, utilizando o algoritmo de Viterbi.

j) Utilize os segmentos das palavras e as repetições das unidades isoladas para, em conjunto, treinar as unidades como se fossem palavras isoladas. Repita os passos i) e j) até a convergência.

TREINAMENTO COM SEQÜÊNCIAS OU FRASES

k) Utilize os modelos estimados para segmentar K repetições de cada seqüência ou frase em unidades, utilizando o algoritmo de Viterbi.

l) Utilize os segmentos das seqüências em conjunto com os segmentos das palavras e as repetições de cada unidade isolada para treinar as unidades como se fossem palavras isoladas. Repita os passos k) e l) até a convergência.

A Fig. 9 apresenta o resultado da segmentação das palavras QUATORZE e DEZESSEIS utilizando as unidades treinadas com o algoritmo proposto. Comparando com a Fig. 1 constata-se de imediato a eficiência do método de treinamento proposto.

10. CONCLUSÕES

Este trabalho analisou vários métodos para inicialização dos parâmetros e dos modelos iniciais de unidades fonéticas representadas por CDHMM e utilizadas no reconhecimento de voz contínua. Também foram examinados algoritmos para treinamento de unidades fonéticas que posssuam grande variabilidade acústica. Finalmente, foi proposto um algoritmo completo para treinamento dessas unidades.

Foi mostrado que o treinamento de modelos de unidades fonéticas tradicionalmente utilizado produz resultados insatisfatórios, ou seja, um descasamento entre a unidade modelada e a unidade acústica, e que o algoritmo proposto consegue resolver esse problema. Os resultados também mostraram que houve uma redução de 92,37% no erro de reconhecimento com a utilização do algoritmo proposto, em relação ao odtido com a utilização do esquema 1 com o algoritmo *level-building*.

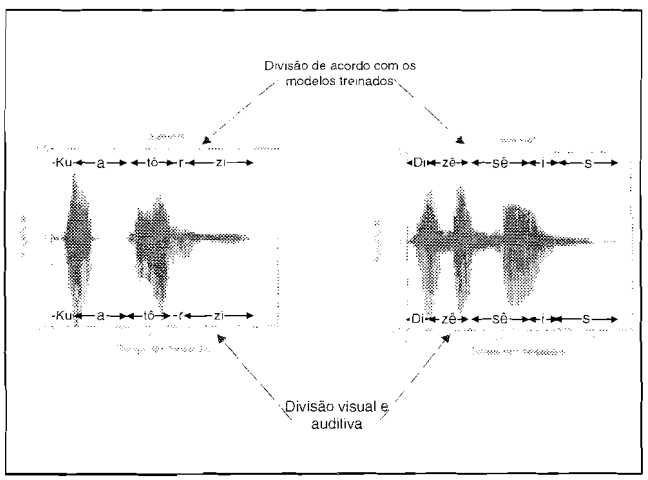

Figura 9. Segmentação das palavras *quatorze* e *dezesseis* de acordo com os modelos obtidos pelo treinamento proposto.

REFERÊNCIAS

[1] Kai-Fu Lee, 'Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition', IEEE Trans ASSP, Vol 38, No 4, pp. 599-609, April 1990.

[2] S. Young. "A Review of Large-vocabulary Continuous-speech Recognition". IEEE Signal Processing Magazine, pp. 45-57, september 1996.

[3] Chin-Hui, L. R. Rabiner and R. Pieracini, 'Speaker Independent Continuous Speech Recognition Using Continuous Density Hidden Markov Models', NATO ASI Series, Speech Recognition and Understanding, Edited by P. Laface and R. De Mori, Springer-Verlag, 1992.

[4] J. G. Wilpon and L. R. Rabiner, 'A Modified k-Means Clustering Algorithm for Use in Isolated Word Recognition', IEEE Trans. ASSP, Vol 33, pp. 587-594, june 1985.

[5] O. Fujimura, 'Syllable as a Unit of Speech Recognition', IEEE Trans. ASSP, Vol 23, No 1, pp. 82-87, February 1975.

[6] A. E. Rosenberg et al. 'Demisyllable-Based Isolated Word Recognition System', IEEE Trans. ASSP, Vol 31, No 3, pp. 713-725, June 1983.

[7] M. J. Hunt, M. Lenning, and P. Mermelstein, 'Experiments in Syllable-Based Recognition of Continuous Speech', ICASSP'80, pp 880-883, 1980.

[8] G. Ruske, 'Automatic Recognition of Syllabic Speech Segments Using Spectral and Temporal Features', ICASSP'82, pp. 550-553, 1982.

[1] O treinamento com palavras é utilizado como passo intermediário para permitir uma primeira incorporação dos efeitos da coarticulação.

[9] S.C.B. Santos, Reconhecimento de Voz Contínua para o Português Utilizando Modelos de Markov Escondidos, Tese de Doutorado, Dezembro 1997, DEE, PUC-Rio.

[10] S. C. B. dos Santos e A. Alcaim, 'Reduced Sets of Subword Units for Continuous Speech Recognition of Portuguese', Electronics Letters, Vol 36, No. 6, pp. 586-588, March 2000.

[11] L. R. Rabiner and B. H. Juang, 'Fundamentals of Speech Recognition', Ed Prentice Hall, 1993.

[12] J. R. Deller et al.,'Discrete-Time Processing of Speech Signals', Ed McMillan Publishing Company, 1993.

[13] Chin-Hui Lee et al., 'Automatic Speech and Speaker Recognition - Advanced Topics', Kluwer Academics Publishers, 1996.

[14] Y. J. Chung and C. K. Un, 'Use of Different Numbers of Mixtures in Continuous Density Hidden Markov Models', Electronics Letters, Vol. 29, No 9, april 1993.

[15] J. Picone, "Continuous Speech Recognition Using Hidden Markov Models', IEEE ASSP Mag. Vol 7, No. 3, pp. 26-41, july 1990.

[16] J. A. Solewicz, J. A. Moraes and A. Alcaim, 'Text-to-Speech System for Brasilian Portuguese Using a Reduced Set of Synthesis Units', Proc. International Symposium on Speech, Image Processing and Neural Networks, april 1994, Hong Kong.

[17] J. A. Solewicz, Síntese de Voz a Partir de Texto para o Português do Brasil, Tese de Mestrado, Agosto 1993, DEE, PUC-Rio.

[18] J. T. Tou and R. C. Gonzalez, 'Pattern Recognition Principles', Ed. Addison-Wesley Publishing Company, 4a edição, 1981.

[19] Simon Haykin, 'Neural Networks – A Comprehensive Foundation', Ed. Prentice-Hall, New Jersey, 1994.

Sidney Cerqueira Bispo dos Santos recebeu o diploma de Engenheiro de Telecomunicações e o título de Mestre em Ciências em Engenharia Elétrica pelo Instituto Militar de Engenharia (IME) em 1985 e 1989, respectivamente, e o título de Doutor em Ciências em Engenharia Elétrica pela Pontifícia Universidade Católica do Rio de Janeiro (PUC/Rio) em 1997. Desde 1988 ele é professor do Departamento de Engenharia Elétrica do IME, sendo atualmente Diretor do Departamento. Suas áreas de interesse em pesquisa são: criptofonia e tecnologias relacionadas à interface homem-máquina, principalmente nas áreas de reconhecimento automático da voz e reconhecimento automático de locutores.

Abraham Alcaim recebeu o diploma de Engenheiro Eletricista e o título de Mestre em Ciências em Engenharia Elétrica pela Pontifícia Universidade Católica do Rio de Janeiro (PUC/Rio) em 1975 e 1977, respectivamente, e os títulos de D.I.C. e Ph.D. em Engenharia Elétrica pelo Imperial College of Science and Technology, University of London, em 1981. Desde 1976 ele é professor do Centro de Estudos em Telecomunicações da Universidade Católica (CETUC), tendo atualmente o cargo de Professor Associado. O Dr. Alcaim trabalha há mais de 20 anos nas áreas de codificação digital e transmissão de sinais e processamento digital de voz e imagem. Ele é autor de diversos artigos publicados em congressos e revistas nacionais e internacionais. Em 1984 ele esteve por um período curto com o Centre National d'etudes des Télécommunications (CNET), em Lannion, França, onde trabalhou em medidas de qualidade objetivas e subjetivas para codificadores de voz. De Dezembro de 1991 a Setembro de 1993 ele foi Cientista Visitante no Centro Científico Rio da IBM Brasil, onde trabalhou no projeto de novos codificadores de imagens, com aplicação especial para imagens obtidas por satélites de sensoriamento remoto. O Dr. Alcaim foi o Technical Program Chairman dos simpósios internacionais SBT/IEEE International Telecommunications Symposiums de 1990 e 1994 - ITS'90 e ITS'94 - realizados no Rio de Janeiro em Setembro de 1990 e em Agosto de 1994, respectivamente. Ele foi o Executive Chairman da IEEE Global Telecommunications Conference (GLOBECOM'99), realizada no Rio de Janeiro em Dezembro de 1999. O Dr. Alcaim é correspondente regional do IEEE Global Communications Newsletter. Desde Março de 1996 ele é membro do Conselho Deliberativo da Sociedade Brasileira de Telecomunicações. O Dr. Alcaim é membro do Comitê de Assessoramento de engenharia elétrica, biomédica e microeletrônica (CA-EE) do CNPq durante o período de Julho de 1998 a Julho de 2001.