SBrT 2000 - XVIII Simpsio Brasileiro de Telecomunicaes, 3 a 6 de Setembro, 2000, Gramado-RS

Reconhecimento de Voz em Tempo Real Baseado na Tecnologia dos Processadores Digitais de Sinais

Marcos da Rocha Vassali[1] José Manoel de Seixas[1] Carlos Espain[2]

[1]Laboratório de Processamento de Sinais, COPPE/EE,
Universidade Federal do Rio de Janeiro,
Caixa Postal 68504, Rio de Janeiro, RJ, Brasil
[2]Faculdade de Engenharia, Universidade do Porto, Portugal
e-mails: vassali@lps.ufrj.br, seixas@lps.ufrj.br, espain@fe.up.pt

Abstract

Um sistema de reconhecimento de voz em língua portuguesa é implementado em um processador digital de sinais (DSP) de alta velocidade, possibilitando um reconhecimento em tempo real. Este reconhecimento baseia-se em modelos escondidos de Markov, utilizando coeficientes mel-cepstrais. O algoritmo é completamente codificado em linguagem C, sendo utilizados dez dígitos para validar a operação de reconhecimento. A eficiência de detecção é de 92% para um sistema dependente do locutor, e de 82% para um sistema independente do locutor.

1 Introdução

A forma mais natural de comunicação humana, sem dúvida alguma, é a comunicação oral. Uma vez que a interação do homem com a máquina é cada vez mais comum, surge uma demanda natural por sistemas capazes de reconhecer nossa fala. Mais especificamente, com a ascensão da *Internet* como um meio privilegiado de comunicação multimídia, a existência de sistemas de processamento de voz para um determinado idioma torna-se uma atividade fundamental para sua inserção, e até mesmo para sua sobrevivência, num mundo de economia competitiva e globalizada.

Dessa forma, encontra-se em desenvolvimento um sistema de reconhecimento e síntese de voz para as versões européia e americana da língua portuguesa [1], de acordo com a colaboração internacional entre UFRJ e Universidade do Porto, Portugal. O presente trabalho dedica-se ao reconhecimento de palavras isoladas, ou mais especificamente, ao reconhecimento de dígitos (entre zero e nove) isolados. Desta forma, o reconhecedor aqui discutido pode ser utilizado em um sistema de informação controlado por menus, usando números como forma de seleção de opções.

O sistema que será apresentado tem por objetivo o reconhecimento de voz de forma completa, desde a captação da palavra pronunciada, até o resultado final de classificação. O processo de reconhecimento é feito a partir de modelos escondidos de Markov (HMM), obtidos para cada um dos dígitos pronunciados, através de um treinamento *offline*. A informação utilizada para o reconhecimento é baseada nos coeficientes mel-cepstrais[2], extraídos para cada segmento do sinal de voz. Foi utilizada uma taxa de amostragem de 11025Hz para o sinal de entrada, uma vez que os sinais de voz raramente possuem energia significante acima de 5512,5Hz (metade da freqüencia de amostragem).

A implementação em um processador digital de sinais (DSP) possui diversas vantagens: velocidade de processamento, baixo custo, programabilidade, compatibilidade, etc. O ambiente escolhido para o trabalho foi o EZkit Lite [3]. Este *kit* comercial de desenvolvimento inclui componentes de *hardware* e ferramentas de *software* necessárias à implementação do sistema no processador ADSP21061.

Com relação às ferramentas de *software*, este *kit* contém um pacote de aplicativos que possibilitam a programação em linguagem C, ou no *Assembly* específico desta família de processadores (ADSP21xxx). Neste projeto, optou-se pela implementação em linguagem C, pela sua maior independência em relação à plataforma de trabalho utilizada, e por ser de depuração mais simples e direta.

Para a linguagem C, além de um compilador, o ambiente de desenvolvimento oferece um conjunto de bibliotecas específicas, que possibilitam a utilização de instruções dedicadas deste processador, bem como a otimização, para esta arquitetura, de determinadas

instruções freqüentes em processamento de sinais[1].

Na seção seguinte, serão apresentas algumas características importantes da plataforma utilizada para o projeto, além de alguns detalhes a respeito do processador (DSP). Em seguida, serão descritos todos os algoritmos implementados. Os resultados obtidos para alguns blocos individuais, e para o reconhecimento, como um todo, são descritos mais adiante na Seção 4. Finalizando, a Seção 5 apresenta algumas conclusões.

2 Ambiente de Trabalho

O *hardware* do EZkit Lite é formado por uma placa de desenvolvimento (de 11,4cm x 16,5cm) e fonte de alimentação (5V), e pretende-se que seja suficiente para acomodar o sistema de reconhecimento. Assim, toda a comunicação e controle do DSP pode ser feita com circuitos auxiliares presentes na placa, incluindo-se o *clock* do procesador, fixado em 40MHz. A entrada e saída de dados pode ser feita de duas formas: analógica, utilizando o conversor AD1847 [4] (conversor A/D e D/A) presente na placa; ou digital, através uma porta serial, que pode ser ligada a um computador.

A Figura 1 mostra a arquitetura do processador ADSP21061, conhecida como SHARC (*Super Harvard Architecture*). Trata-se de um processador de alta velocidade, em ponto flutuante, com 32 bits de precisão expansíveis até 40 bits.

De imediato, nota-se que existe uma ampla rede de barramentos, permitindo um intenso fluxo de dados. Outra característica interessante desta arquitetura é a divisão do processamento matemático em três circuitos independentes, que atuam em paralelo: ALU, multiplicador e *shifter*. Dessa forma, podem ser processadas de uma a quatro instruções em apenas um ciclo de *clock*. Vale lembrar que, em processamento de sinais, é muito comum a necessidade de se executar uma série de adições, multiplicações e deslocamentos (filtragem, correlação, etc.), e assim, esta arquitetura possibilita um ganho de velocidade considerável na execução destes algoritmos.

No canto superior esquerdo da Figura 1, localizam-se duas unidades geradoras de endereços (DAG's), que permitem, entre outras formas de endereçamento, a criação dos chamados *buffers* circulares. Estes *buffers* circulares constituem vetores de indexação circular, e são de grande importância para este projeto, como será visto na Seção 3. A indexação circular significa que todo elemento do vetor possui um antecessor e um sucessor, e portanto, não mais existe

[1] Um algoritmo de FFT, por exemplo, já encontra-se implementado de forma otimizada em uma destas bibliotecas.

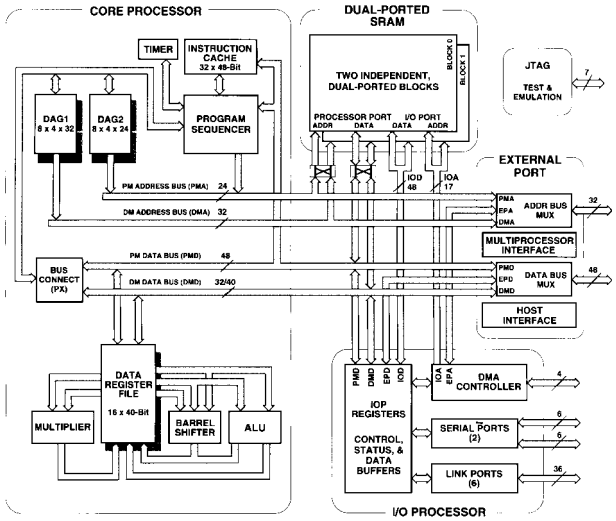

Figure 1: Arquitetura do processador ADSP21061.

a definição de primeiro ou último elemento, mas somente uma relação de ordem.

A comunicação do conversor AD1847 com o processador é feita através de uma das portas serias deste processador, através de um protocolo próprio.

O ADSP21061 possui, ainda, uma memória interna de 1 Mbit, dividida em dois blocos, que podem ser acessados concomitantemente pelas outras unidades do processador. No primeiro bloco, tem-se a memória de programa (PM), com 512 kbits, que são subdivididos entre as instruções do programa e dados. No segundo bloco, define-se a memória de dados (DM), também com 512 kbits, que são acessados como dados de 32 bits (16k palavras de 32 bits).

Apesar do ADSP21061 possuir 32 bits para endereçamento de memória (24 para PM e 32 para DM), o que permite a formação de até 4 Giga endereços, a placa EZ Lite não possui memória externa, e por isso toda a memória disponível fica restrita à memória interna do processador. Assim, o número máximo de palavras de 32 bits para armazenamento de dados é de apenas 20k (16k da memória de dados e 4k da memória de programa).

Como a freqüência de amostragem utilizada foi de 11025Hz, são enviadas 11025 amostras de sinal por segundo ao DSP. Com uma memória com apenas 20k palavras disponíveis para todos os tipos de dados, fica evidente que a gravação do sinal na memória é completamente inviável, pois toda a memória estaria ocupada com menos de 2s de gravação. Entre-

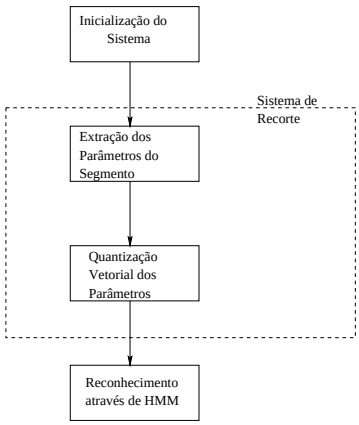

Figure 2: Diagrama de blocos simplificado da execução do programa.

tanto, numa operação em tempo real, em que os dados provenientes do sinal de voz são processados à medida em que são captados, esta restrição de memória não chega a comprometer o desempenho do sistema, como será visto adiante.

3 Algoritmos Implementados

Antes dos algoritmos utilizados em cada um dos blocos funcionais do programa serem descritos, é interessante identificar quais são estes blocos. Um diagrama bastante simplificado da execução do programa pode ser visto na Figura 2.

No bloco de inicialização do sistema, além de inicializar os valores de algumas variáveis, o sistema irá captar e extrair, do ruído de fundo, três grandezas que serão utilizadas pelo sistema de recorte da palavra: a média do módulo, da energia, e da taxa de cruzamento por zero. Além disso, para evitar problemas no recorte, devido à componente contínua introduzida pela tensão de *offset* do conversor A/D, utilizam-se os primeiros 400ms de ruído para a estimação desta componente contínua, que será retirada do sinal captado posteriormente.

A seguir, podem ser vistos dois blocos dentro de um grande bloco denominado sistema de recorte. Esta figura deve ser entendida da seguinte forma: terminado o processo de inicialização do sistema, este entrará em um modo de execução do qual só sairá quando for detectada e recortada uma determinada palavra. No entanto, não é possível saber *a priori* se um segmento de sinal pertence a uma palavra.

Dessa forma, é necessário processar este segmento e obter toda a informação necessária ao reconhecimento (quantização vetorial dos parâmetros formados pelos coeficientes mel-cepstrais). Caberá ao sistema de recorte identificar se esta informação será utilizada no reconhecimento ou não, o que vai depender de como este segmento foi classificado pelo sistema de recorte: como ruído ou como parte de uma palavra. Fica claro, portanto, que a extração de coeficientes e a quantização vetorial são blocos que realizam suas funções enquanto o sistema encontra-se em um modo de execução específico, que é o sistema de recorte.

Como em todo sistema de processamento de voz, o sinal é segmentado de modo que o trecho processado possa ser considerado uma amostra de um processo estocástico estacionário[5, 2]. Para tal, neste trabalho, cada segmento foi definido como um trecho do sinal de entrada com duração de 23,22ms, o que corresponde a exatamente 256 amostras, devido à freqüência de amostragem de 11025Hz. Para o janelamento, foi utilizada a janela de Hamming, com uma superposição de 50%, ou seja, das 256 amostras de um segmento, as 128 primeiras também se encontram no segmento anterior, e as 128 últimas no segmento posterior. Este procedimento evita que o trecho do sinal que seria atenuado pelas extremidades da janela do segmento não contribua com informação útil para o algoritmo de reconhecimento.

O algoritmo utilizado para o recorte pode ser subdivido em duas etapas: detecção robusta e detecção refinada. Na etapa de detecção robusta, utiliza-se apenas a informação de módulo médio do ruído. Este é então comparado com os valores de módulo calculados para cada segmento do sinal, até que algum deles seja maior do que 30 vezes o módulo médio do ruído. Isto indica para o sistema que o segmento que está sendo processado pertence a uma palavra. Assim, permanecerão armazenadas, somente, as informações do segmento atual, dos 79 segmentos seguintes, e dos 20 segmentos anteriores a este. Obtém-se, dessa forma, informação a respeito de 100 segmentos, dentro dos quais encontra-se uma palavra. Os 100 segmentos, com superposição de 50%, equivalem a 1,161s de sinal, tempo suficiente para a pronúncia das palavras em questão (dígitos).

Os valores de energia e taxa de cruzamento por zero, calculados para cada segmento de sinal, juntamente com os índices do *codebook* relativos à quantização vetorial dos parâmetros utilizados pelo reconhecimento, são armazenados em *buffers* circulares, de modo que os valores dos segmentos mais recentes sobreponham-se aos valores dos segmentos antigos, que não mais interessam, uma vez que não pertencem à palavra alguma.

A partir daí, o sistema interrompe a captação de sinal, uma vez que uma palavra já foi detectada, e dá início à detecção refinada. A detecção refinada irá

utilizar as informações de energia e taxa de cruzamento por zero para implementar uma detecção mais rigorosa, na tentativa de identificar, dentre os 100 segmentos selecionados pela detecção robusta, quais os que realmente correspondem a uma palavra [6].

Portanto, a cada novo segmento formado haverá uma extração de coeficientes mel-cepstrais e uma quantização vetorial. Somente com o início da detecção refinada estes blocos não serão mais executados, uma vez que esta assume que toda a palavra já foi pronunciada, interrompendo, assim, o sistema de aquisição.

A formação do vetor de parâmetros para o reconhecimento utiliza as seguintes variáveis: a energia do segmento janelado, os doze coeficientes mel-cepstrais extraídos, a variação entre a energia atual e a do segmento anterior (delta-energia), e a variação dos coeficientes mel-cepstrais (delta-coeficientes). Portanto, o vetor de parâmetros é formado por 26 elementos.

A extração dos coeficientes mel-cepstrais utilizada baseia-se, primeiramente, no cálculo da energia em diversas sub-bandas, que são obtidas através dos filtros de banda crítica [2]. Neste projeto, foram utilizados 24 destes filtros. Extraindo-se o logaritmo destes 24 elementos[2], basta calcular sua DCT para que sejam obtidos os coeficientes no domínio mel-cepstral. No entanto, a separação das informações de excitação e do trato vocal só será realizada com o janelamento deste sinal no domínio mel-cepstral. Utiliza-se, para isso, a janela de Tohkura[5] que irá selecionar somente as componentes de baixa qüefrência (trato vocal), ponderadas de acordo com o valor correspondente da janela, cuja equação segue abaixo:

$$\begin{aligned} l(n) &= 1 + \frac{N_{coef}}{2}.sen(\frac{\pi.n}{N_{coef}}), n = 1...N_{coef} \\ l(n) &= 0, n > N_{coef}, \end{aligned}$$

onde N_{coef} é o número de coeficientes mel-cepstrais que se deseja extrair.

A quantização vetorial destes parâmetros utiliza a distância euclidiana para avaliar qual o vetor do *codebook* (de 128 palavras) que mais se aproxima do vetor de parâmetros extraído.

O algoritmo de reconhecimento utilizado para o cálculo das probabilidades de cada modelo foi o conhecido método de Baum-Welch [2].

[2]Utiliza-se o logaritmo para que os espectros dos sinais de excitação e do filtro do trato vocal componham uma combinação linear, e não mais um produto [5].

4 Resultados

Para a avaliação do sistema de recorte, foi feita uma comparação do resultado obtido pelo DSP com uma implementação *offline*. Entre os recortes obtidos por estas duas implementações, foi observado um deslocamento médio de 32ms, com um desvio padrão de 38ms. Portanto, também neste aspecto quantitativo, a implementação no DSP pode ser considerada bem realizada, pois o deslocamento relativo é bastante reduzido. Mediu-se também a razão entre os comprimentos dos sinais recortados pelas duas implementações do algoritmo de recorte (DSP e *offline*), obtendo-se 1,2 para a média, e 0,4 para o desvio padrão. Estes resultados significam que, devido ao ruído que se soma ao sinal de voz, um sinal recortado pelo DSP, em média, estará deslocado de 32ms e terá 20% mais amostras do que um sinal recortado *offline*, o que pode ser considerado bastante satisfatório.

Quanto à qualidade da extração de parâmetros e da quantização vetorial, os testes também se basearam na comparação das implementações em DSP (usando 32 bits) e *offline* (usando 64 bits). A diferença média entre ambas as realizações foi de $5,2\text{x}10^{-7}$, com uma variância de $2,0\text{x}10^{-9}$, o que significa que os erros acumulados pela menor precisão do DSP não acarretam problemas para o sistema de reconhecimento.

Uma outra medida bastante relevante é o tempo de processamento. A detecção robusta possui um tempo de execução de apenas 1,825μs, o que não acarreta problema algum para o reconhecimento em tempo real que se deseja, uma vez que o período de amostragem é de aproximadamente 90μs (1/11025Hz). Já o tempo necessário à extração e quantização dos parâmetros merece um maior cuidado. Sabe-se que estes dois blocos devem ser completamente executados antes que seja formado um novo segmento, e que, devido à superposição dos segmentos, a diferença entre o final de dois segmentos consecutivos é de 128 amostras. Portanto, todo o processamento para um segmento deve ser feito em um tempo inferior a 128 vezes o período de amostragem (aproximadamente 11,6ms). Verificou-se que o tempo médio de processamento é de 8,0ms, com um desvio padrão de 0,2ms, satisfazendo, portanto, este requisito.

Uma vez finalizada a pronúncia de uma palavra, falta apenas refinar sua detecção e executar o algoritmo de reconhecimento, já que as informações necessárias foram extraídas em tempo real. Portanto, o único requisito para o tempo de processamento destes algoritmos é que seja curto o suficiente para garantir o reconhecimento em tempo real. Foi obtido, para a detecção refinada, um tempo médio de

54,9μs, com um desvio padrão de 10,6μs. Já para o algoritmo de reconhecimento, foi verificado que são necessários 8,7ms, em média, para sua realização (o desvio padrão foi de 0,9ms). Portanto, ao fim da pronúncia da palavra, o sistema é capaz de fornecer sua resposta em menos de 10ms, o que garante o processamento em tempo real, como desejado.

Para obter as eficiências de reconhecimento, foi utilizado, primeiramente, um conjunto de treino para um usuário, composto por dez arquivos de som *wave* para cada dígito. Estes sinais entram na placa EZ Lite através de uma conexão com a saída da placa de som de um computador que contenha estes arquivos de som. O treinamento dos modelos escondidos de Markov (HMM) é feito *offline*, e estes modelos são inseridos juntamente com o programa que é executado neste processador.

A partir destes mesmos arquivos de som, foi obtida uma eficiência de reconhecimento de 98%. No entanto, este não é um bom critério para avaliar o sistema de reconhecimento, uma vez que os sinais a serem reconhecidos são muito parecidos com os que treinaram os modelos, diferindo apenas devido à aleatoriedade do ruído que se adiciona ao sinal.

Portanto, definiu-se um conjunto de teste, composto por cem novos arquivos de som (dez arquivos para cada dígito). Para este novo conjunto, o percentual de acerto obtido foi de 92%. A queda deste resultado com relação ao primeiro já era esperada, uma vez que este novo conjunto de sinais de voz não participou do treinamento, constituindo, assim, um verdadeiro conjunto de validação.

Em seguida, foi utilizado um conjunto de treino com vozes de dez pessoas diferentes, de modo a treinar um sistema independente do locutor. Este conjunto é composto de 500 arquivos de som, sendo 50 para cada dígito. Para este conjunto (de treino), foi obtida uma eficiência de 93%. Foi então utilizado um conjunto de validação com vozes de outras quatro pessoas, totalizando 200 novos arquivos de som (vinte para cada dígito). Para este novo conjunto (de validação), a eficiência de reconhecimento foi de 82%.

A tabela 1 mostra, para as vinte amostras de cada dígito, como foi a discriminação feita pelo sistema. A primeira coluna indica qual o dígito inserido no sistema, enquanto as demais colunas indicam o número de vezes que cada dígito foi reconhecido. Este tipo de tabela é usualmente conhecido como tabela de confusão. A coluna X indica uma indecisão por parte do sistema (nenhum dígito reconhecido).

Table 1: Tabela de confusão.

	Dígito Reconhecido										
	0	1	2	3	4	5	6	7	8	9	X
0	13	1	0	0	0	0	0	6	0	0	0
1	0	16	0	0	0	4	0	0	0	0	0
2	0	0	19	0	0	0	0	0	1	0	0
3	0	0	0	17	0	1	2	0	0	0	0
4	0	0	0	0	16	0	0	0	0	3	1
5	0	1	0	0	0	18	1	0	0	0	0
6	0	1	0	4	0	2	12	0	0	0	1
7	0	0	0	0	0	1	0	16	0	0	3
8	0	1	1	0	0	0	0	0	18	0	0
9	0	1	0	0	0	0	0	0	0	18	1

5 Conclusão

Foi apresentado um sistema de reconhecimento de palavras em tempo real implementado em um processador digital de sinais. Os tempos de processamento encontrados satisfizeram os requisitos de operação em tempo real, validando a plataforma utilizada. Pode-se afirmar, também, que o sistema foi capaz de exibir uma boa generalização, devido às altas eficiências obtidas: 92% para dependente do locutor, e 82% para independente do locutor.

Este sistema de reconhecimento encontra-se em expansão para funcionamento em ambas as versões da língua portuguesa (americana e européia), de acordo a colaboração UFRJ/Universidade do Porto.

Agradecimentos

Nossos agradecimentos para CAPES, CNPq, FAPERJ e FUJB (Brasil), e ICCTI (Portugal) que apoiaram financeiramente o projeto, e aos professores M. N. Souza, F. G. V. Resende Jr. e S. L. Netto pelas enriquecedoras discussões a respeito de temas relacionados a este trabalho.

References

[1] Marcio N. Souza, Luiz P. Calôba, and José M. de Seixas, "Developing a voiced information retrieval system of the accented syllable in portuguese: a contribution to the naturalness of speech synthesis," Budapest, Hungria, 1999, Eurospeech.

[2] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[3] Analog Devices, *SHARC EZ-KIT 2106X User's Manual*, 1997.

[4] Analog Devices, *AD1847 Serial-Port 16-bit SoundPort Stereo Codec*, 1996.

[5] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-time Processing of Speech Signals*, MacMillan, New York, 1995.

[6] M. R. Vassali, C. L. Matos, J. M. Seixas, M. N. Souza, and L. P. Calôba, "Implementação de um algoritmo de recorte de palavras em tempo real na tecnologia dos processadores digitais de sinais," Submetido ao XIII Congresso Brasileiro de Automática, 2000.