SRM: *Framework* para o Reconhecimento de Som em Dispositivos Móveis

Marcelo Ruaro[1], Denilson Rodriges da Silva[2]

Departamento de Engenharias e Ciência da Computação – Universidade Regional Integrada do Alto Uruguai e das Missões (URI)
Caixa Postal 98.802-407 – Santo Ângelo – RS – Brasil

mceloruaro@gmail.com, deniro@urisan.tche.br

Resumo. *Este trabalho propõe o Sound Recognizer ME (SRM), um framework desenvolvido na plataforma Java Micro Edition, concebido para oferecer funcionalidades de reconhecimento de sons abstratos e de palavras isoladas dependentes de locutor. Na realização dessa proposta foi utilizada a API MMAPI para a captura e execução do áudio na codificação PCM. Para a fase de extração de informações do sinal foi utilizada a extração dos coeficientes Mel-Cepstrais (MFCC) derivados da Transformada Rápida de Fourier (FFT), e para o reconhecimento foi empregada a comparação através da Dinamic Time Warping (DTW). Como produto final obteve-se um motor de reconhecimento de fácil integração a aplicações móveis Java ME e que se mostrou eficaz no reconhecimento dos gêneros de som envolvidos.*

1. Introdução

Ao longo dos anos, o avanço tecnológico dos microprocessadores proporcionou a evolução dos dispositivos portáteis, que consequentemente, permitiu a incorporação de recursos que normalmente se localizavam em ambientes computacionais de grande porte. Seguindo esta linha, a área de reconhecimento de voz voltada aos dispositivos móveis, pode se beneficiar dessa capacidade e oferecer um processamento mais robusto, através da integração de novas metodologias [Huerta 2000]. Entre estas metodologias, destaca-se a extração dos coeficientes *Mel-Cepstrais* (MFCCs) que fornecem uma boa base de parâmetros para o uso de técnicas de reconhecimento de padrões, principalmente, no reconhecimento independente de locutor [Rabiner e Juang 1978] [Yoma 1993].

Contudo, as técnicas de reconhecimento de voz são relativamente complexas, principalmente devido a sua natureza interdisciplinar requisitada em seu desenvolvimento, podendo-se citar entre as áreas envolvidas o processamento de sinais digitais, linguagens de programação, reconhecimento de padrões, inteligência artificial, linguística, teoria das comunicações, entre outras, variando de acordo com a complexidade do reconhecimento [Petry 2002][Sphinx-4 2008]. Em consequência disto, desenvolver um sistema desse gênero demanda – além de um amplo conhecimento – um grande esforço de tempo, o que faz com que esta capacidade deixe de ser integrada pelos desenvolvedores de *software* de uma maneira mais ocasional principalmente em aplicações móveis.

Com base neste cenário, este trabalho descreve a metodologia implementada, e estrutura desenvolvida para o reconhecimento de som no *framework* SRM [Ruaro 2010]. O SRM é inteiramente escrito na linguagem Java ME, com a capacidade de

oferecer, de uma forma abstraída e de fácil integração, o reconhecimento de voz através do reconhecimento, envolvendo um vocabulário pequeno, de palavras isoladas dependente de locutor e o reconhecimento de sons abstratos, este último, definido como qualquer som que não caracterize uma palavra.

Para elaboração desta proposta foi utilizada na parte de Extração da Informação a extração dos coeficientes MFCC derivados da Transformada Rápida de Fourier (FFT), e da análise por meio de um banco de filtros na escala *Mel* seguindo a metodologia proposta por Rabiner *et al.* (1978). Já para o reconhecimento, foi utilizada a *Dinamic Time Warping* (DTW), que além da vantagem do baixo custo computacional possibilita o reconhecimento de elocuções que apresentam variações de velocidade em sua composição [Yoma 1993].

A sessão 2 abaixo, explica quais as técnicas e parâmetros utilizados na captura e digitalização do sinal, a sessão 3 descreve a metodologia da fase de Extração de Informação, na sessão 4, é exposta a forma de reconhecimento empregada, a sessão 5 descreve a estrutura do *framework,* a sessão 6 exibe os testes e resultados, e a sessão 7 encerra, apresentando as conclusões.

2. Captura e Digitalização do Sinal

Inicialmente, o sinal foi capturado utilizando a API MMAPI [Goyal 2006], na modulação PCM (*Pulse Code Modulation*), com o formato *Wave*, a uma taxa de 64Kbps, portanto, com uma frequência de amostragem de 8KHz e 8 bits por amostra. Estes parâmetros foram pré-definidos visando diminuir a carga computacional envolvida nos processos de Extração de Informação e Reconhecimento, e também, respeitando os parâmetros mínimos de qualidade de áudio considerados na área de reconhecimento de voz [Rabiner e Juang 1978].

Assim, de posse do sinal, este foi submetido a técnicas de Processamento de Sinais Digitais (PSD), iniciado pela Extração de Informação [Petry 2002].

3. Extração de Informação

A Extração de Informação visou gerar o menor volume de parâmetros suficientes para caracterizar cada amostra de som, e compreendeu as fases de Pré-processamento, Análise Espectral e Extração dos Parâmetros. Seguindo a metodologia de Rabiner *et al*(1978) e analisada comparativamente de acordo com o proposto por Lima (2000), para obtenção dos coeficientes *Mel-Cepstrais* – ver figura 1.

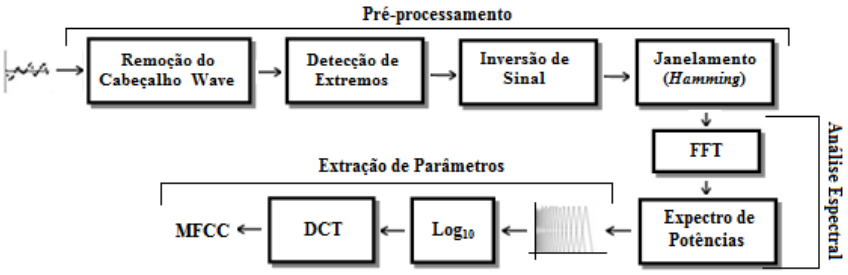

Figura 1. Processamento de Sinal realizado

A extração dos coeficientes MFCC teve sua implementação baseada no *framework Sphinx-4* (2008), onde neste trabalho, foi implementado um banco de 15 filtros triangulares passa-faixa, com frequência mínima de 100Hz e máxima de 3700Hz

seguindo a escala *Mel* [Davis e Mermelstein 1980]. Em seguida foi aplicada a Transformada Discreta dos Cossenos que equivale a Transformada Inversa de *Fourier* [Cuadros 2007], obtendo desta maneira 11 coeficientes MFCC. Normalmente como processo adicional e observado em [Sphinx-4 2008] [Cuadros 2007] é descartado o primeiro coeficiente MFCC obtido, pois este geralmente pode carregar muita informação do meio de transmissão, desnecessária para o reconhecimento. Resultado em 10 coeficientes MFCC extraídos para cada janela e variáveis conforme as características do sinal.

4. Reconhecimento

No reconhecimento buscou-se identificar através de comparação pela DTW, o som de teste, que é o som imposto ao reconhecimento, com algum padrão armazenado no banco de padrões (persistido utilizando um *Record Store*).

Na DTW, cada comparação entre coeficientes MFCC do som de teste com um padrão resulta em uma distância [Furtună 2008], onde se adotou o uso de coeficientes de seletividade Sel_1 e Sel_2 propostos na página 4.8 no trabalho de Yoma (1993).

Estes coeficientes foram utilizados devido à qualidade dos resultados obtidos naquele trabalho. De maneira em que, quanto maior seus valores, melhor será a qualidade ou seletividade do reconhecimento. Onde, neste trabalho, baseando-se no melhor desempenho obtido na fase de testes, foram definidos os valores mínimos de 0,25 para Sel_1 e 0,8 para Sel_2 (se Sel_1 for maior que 0,25). Sendo que, se estes índices não forem alcançados, o *framework* retorna um estado de não reconhecimento.

5. Estrutura do *Framework*

Um dos objetivos do SRM, além do reconhecimento de som, é oferecer este recurso de uma maneira simples e abstraída para que sua integração em aplicações seja de forma rápida, não demandando muitos esforços de aprendizado nas áreas abrangidas pelo reconhecimento de voz. Desta maneira, o SRM é divido em pacotes, cujos objetivos são organizar as principais funcionalidades oferecidas. Os pacotes foram denominados *sound, back-end* e *front-end*, e a composição destes está representada na figura 2 abaixo.

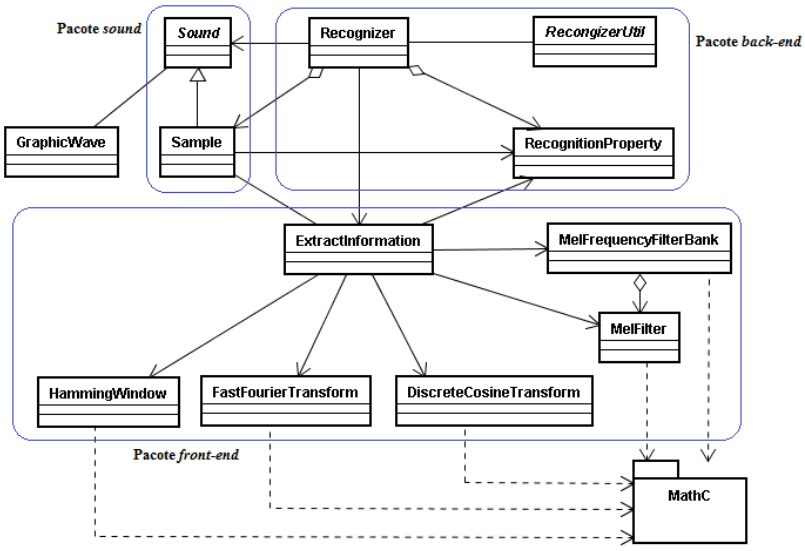

Figura 2. Diagrama de Classe do *framework* SRM

Resumidamente, o pacote *sound* é responsável pela aquisição, execução, e manipulação de amostras que podem ser definidas como padrão. Já o pacote *front-end* é inteiramente dedicado ao processamento do sinal digital, capaz de extrair as informações necessárias do sinal original para que o pacote *back-end* possa realizar a interface de reconhecimento com o usuário através de métodos que abstraem este processo. Já a classe *GraphicWave* tem como único objetivo a representação gráfica das amplitudes do sinal ao longo do tempo, que por fins de estudo, foi mantida. E a biblioteca *MathC* foi integrada para a realização de funções matemáticas, que não são implementadas pela biblioteca *Math* genuína do Java ME.

5.1. Reconhecimento utilizando o SRM

Após definir as amostras necessárias para o reconhecimento, já é possível reconhecer um som de teste com as amostras/padrões. Nesta etapa, é efetuado o processo de extração da informação do sinal capturado. E seus coeficientes MFCC são comparados com os coeficientes armazenados no banco de padrões pela DTW, retornando um *id* que é utilizado para recuperar a palavra associada a este som.

Todo o reconhecimento ocorre na classe *Recognizer*, que fornece a função *startRecognizer*, onde esta recebe a amostra a ser reconhecida e as propriedades do reconhecimento (*RecognizerProperty*), e elabora o reconhecimento de uma forma abstrata. Um exemplo dessa implementação, pode ser observado na figura 3 abaixo.

```
Recognizer recognizer = new Recognizer(new Sample(recordedSound), property);
int id = recognizer.startRecognizer();
if (id != -1){
       String palavraReconhecida = Sample.getSampleString(id, "word");
}
```

Figura 3. Exemplo de reconhecimento utilizando o *framework SRM*

Caso o reconhecimento tenha acorrido a variável *id* conterá a localização da palavra correspondente ao banco de palavras criado, caso contrário, se não houver reconhecimento, o valor de *id*, será igual a -1.

6. Testes e Resultados

Para realizar os testes e obter os resultados baseando-se em dispositivos móveis, foi desenvolvido um aplicativo chamado *CallDictation* integrando o *framework* SRM, cujo objetivo foi reconhecer dígitos e realizar a chamada dos dígitos reconhecidos por voz, totalizando em 12 padrões ao banco de padrões, que são os dígitos decimais, mais dois comandos de ações. Este aplicativo foi instalado na memória do dispositivo a fim de diminuir o tempo de acesso em relação a sua instalação em um cartão de memória.

Como padrão, adotou-se 3 dispositivos com diferentes tipos de processador, estes dispositivos são caracterizados como dispositivo 1, 2 e 3, com processadores operando a frequência de 434 MHz, 369 MHz e 332 MHz respectivamente.

O desempenho de reconhecimento neste teste foi exposto em duas tabelas abaixo. A tabela 1 corresponde ao tempo gasto de processamento, em segundos, para o reconhecimento entre todos os 12 padrões definidos no teste. E a tabela 2, corresponde a porcentagens de acertos envolvendo 120 reconhecimentos.

Tabela 1. Custo de processamento representado em segundos

PALAV.:	ZERO	UM	DOIS	TRÊS	QUATRO	CINCO	SEIS	SETE	OITO	NOVE	OK	SAIR	MÉDIA
DISP. 1	0,89	0,57	0,69	0,69	0,89	0,86	0,66	0,85	0,95	0,86	0,87	0,79	**0,855**
DISP. 2	1,32	1,09	1,08	1,19	1,3	1,22	1,14	1,28	1,5	1,51	1,45	1,34	**1,29**
DISP. 3	2,14	1,16	1,73	1,75	2,22	2,22	1,74	2,17	2,4	2,2	2,15	1,82	**2,145**

Tabela 2. Taxa de acertos representados em porcentagem

PALAV.:	ZERO	UM	DOIS	TRÊS	QUATRO	CINCO	SEIS	SETE	OITO	NOVE	OK	SAIR	MÉDIA
ACERTOS	90	100	100	90	90	100	100	100	100	100	100	90	**96,66**

Uma característica importante deste processo, é que toda vez que o reconhecimento é efetuado, também é realizada a extração de informação da amostra que esta sendo reconhecida, demandando um maior tempo geral ao processo de reconhecimento.

6.1. Resultados Gerais

Os resultados apresentados a seguir refletem o reconhecimento para cada tipo de som e palavra isolada, em um contexto geral, onde foram realizados testes com locutores de diferentes idades e sexo, e variados sons abstratos comuns ao cotidiano humano.

As duas colunas representam os tipos de reconhecimento já descritos, e este reconhecimento ainda é dividido em reconhecimento em tempo real e reconhecimento estacionário. No reconhecimento em tempo real captura-se o som em intervalos pré-definidos (para estes testes foi considerado o tempo de 1,5 segundos), e então se verifica se o som capturado corresponde a algum padrão armazenado. No reconhecimento estacionário, o recurso de reconhecimento pode ser acionado a partir de um botão disponível na interface da aplicação, e que quando utilizado, realiza a captura de som por 2 segundos.

É importante registrar que as condições de realização dos testes representados pela tabela 3, foram em ambientes silenciosos, e as palavras e sons emitidos tanto para a definição dos padrões quanto para o reconhecimento foram emanados de forma clara e inteligível.

Tabela 3. Precisão de reconhecimento do framework SRM

	Palavras Isoladas	Sons Abstratos
Estacionário	92%	94%
Tempo Real	88%	87%

A porcentagem de acertos exposta na tabela 3 abaixo caracteriza o desempenho final do SRM para palavras isoladas dependente de locutor e sons abstratos.

7. Conclusão

Conclui-se que o *framework* SRM, se caracteriza como um motor de reconhecimento abstraído, de fácil integração a aplicações móveis Java ME e que se mostrou eficaz no reconhecimento dos gêneros de som envolvidos, apresentando uma precisão, no melhor caso de 94%, e no pior caso de 87% de acertos nas amostras utilizadas nos testes em dispositivos. Outra característica apresentada pelo SRM foi um consumo de processamento aquém do esperado, se caracterizando como um sistema de baixo custo, levando-se em consideração a complexidade das técnicas abordadas sobre o contexto de processamento embarcado atual.

Também, pode-se concluir que *framework* SRM apresentou um bom desempenho levando-se em consideração a baixa frequência de amostragem de 8KHz e a qualidade do microfone do dispositivo. Outra consideração, é que o reconhecimento apresentou certas confusões ao reconhecer palavras muito semelhantes, como por exemplo: “casa”, “asa”, “testa”, “festa”, “e”, ”d”. Este problema é ocasionado principalmente devido a baixa frequência de amostragem utilizada. Para suprimir este problema recomenda-se aumentar a frequência de amostragem (caso suportada pelo dispositivo) e/ou definir um banco de padrões menor.

Referências

Cuadros, C. D. R. (2007) "Reconhecimento de Voz e de Locutor em Ambientes Ruidosos: Comparação das técnicas MFCC e ZCPA". Escola de Engenharia da Universidade Federal Fluminense, Niterói.

Davis, S. B. and Mermelstein, P. (1980) "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences". IEEE Transactions on Acoustics, Speech, and Signal Processing.

Furtunã, T. F. (2008) "Dynamic Programming Algorithms in Speech Recognition". Revista Informatica Economică nr. 2(46)/2008, Academy of Economic Studies, Bucharest.

Goyal, V. (2006) "Pro Java ME MMAPI Mobile Media API for Java Micro Edition". New York: apress.

Herath, I. and Rangel, R. G. (2007) "Implementation of an Electronic Tuner in J2ME using Fast Fourier Transform". Peradeniya University Research Sessions, Sri Lanka, Vol.12, Part II, Peradeniya.

Lima, A. A. D. (2000) "Análises Comparativas em Sistemas de Reconhecimento de Voz". UFRJ, Rio de Janeiro, p102.

Petry, A. (2002) "Reconhecimento Automático de Locutor Utilizando Medidas de Invariantes Dinâmicas Não-Lineares". URGS, Instituto de Informática, Porto Alegre.

Rabiner, L. and Juang, B.-H. (1978) “Fundamentals of Speech Recognition”. New Jersey: Englewood Cliffs: Prentice Hall.

Ruaro, M. (2010) "SRM: Framework para Reconhecimento de Som em Dispositivos Móveis". Universidade Regiona Integrada do Alto Uruguai e das Missões (URI), Santo Ângelo, p 91.

Sphinx-4. Framework Sphinx-4. (2008) "A speech recognizer written entirely in the JavaTM programming language",. Disponivel em: <http://cmusphinx.sourceforge.net/sphinx4/>. Acesso em: 1 out. 2010.

Yoma, N. B. (1993) "Reconhecimento Automático de Palavras Isoladas: Estudo e aplicação dos métodos Determinísticos e Estoásticos". Departamento de Comunicações da Faculdade de Engenharia Elétrica - UNICAMP, Campinas.