



Universidade Católica Dom Bosco
Curso de Bacharelado em Engenharia de Computação

**Reconhecimento de Gestos Utilizando Modelos Ocultos de
Markov**

Jéssica Barbosa Dias
Kleber Padovani de Souza

Orientador: Prof. Dr. Hemerson Pistori

Projeto apresentado à Coordenação do Curso de Bacharelado em Engenharia de Computação da Universidade Católica Dom Bosco como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

UCDB - Campo Grande - MS - Março/2006

Capítulo 1

Dados do Projeto

1.1 Reconhecimento de Gestos Utilizando Modelos Ocultos de Markov

1.2 Local de Realização

Universidade Católica Dom Bosco
Av. Tamandaré 6000 - Jd.Seminário - CEP: 79.117-900
Campo Grande - MS Caixa Postal: 100

1.3 Responsável pelo Projeto

Jéssica Barbosa Dias
Kleber Padovani de Souza

1.4 Professor Orientador

Prof. Dr. Hemerson Pistori

Capítulo 2

Introdução

Continuamente a vida humana está tendo um maior contato com os computadores e dispositivos eletrônicos, tanto nas atividades pessoais quanto nas profissionais. A facilidade na utilização destes equipamentos é uma característica essencial em seus projetos, pela simples razão de que dispositivos de difícil interação são menos utilizados e, geralmente, geram um desperdício de tempo ao usuário devido à necessidade de treinamento.

De forma geral, a interação entre pessoas e computadores (interface homem-máquina) atualmente ocorre através dos dispositivos de entrada de dados habituais, como o mouse e o teclado. Através da utilização de dispositivos óptico-eletrônicos, como as webcams, é possível enriquecer este paradigma de comunicação. A análise e reconhecimento dos gestos, posturas e expressões humanas podem auxiliar muito a interface homem-máquina.

Segundo o Censo Demográfico 2000 do Instituto Brasileiro de Geografia e Estatística, cerca de 937 mil brasileiros são tetraplégicos, paraplégicos ou hemiplégicos permanente, e mais de 5 milhões são incapazes, ou possuem alguma ou grande dificuldade permanente de ouvir. Considerando o fato de existir uma parcela considerável de pessoas atualmente integral, ou parcialmente impedidas de utilizar os computadores da forma convencional, um outro benefício muito importante do reconhecimento de gestos pelos computadores é a sua contribuição na inclusão digital desta categoria de pessoas que possuem necessidades especiais. Um sistema capaz de identificar comandos do usuário por meio da movimentação do globo ocular poderia ser utilizado por tetraplégicos como um meio de comunicação entre eles e os dispositivos, o que os incluiriam no mundo digital considerando suas limitações.

Por fim, existe uma grande quantidade de aplicações que podem ser desenvolvidas utilizando comunicação homem-máquina guiadas por sinais visuais. Porém, um dos empecilhos para a propagação do desenvolvimento deste tipo de aplicativos é a complexidade de implementação

destas interfaces. Para suprir esta necessidade foi criado um projeto, denominado SIGUS, que tem como objetivo o desenvolvimento de um ambiente computacional, de códigos-fonte abertos, que auxilie o desenvolvimento de aplicações que sejam guiadas por interfaces não-convencionais, mais especificamente, guiadas por sinais visuais, como movimentos da mão e da face.

Atualmente, esta ferramenta possui várias implementações de algoritmos da área de visão computacional e reconhecimento de padrões, dentre elas algoritmos de segmentação, extração de características e classificação de imagens. Através dela foram implementados alguns programas para análise da eficácia da aplicação destas teorias, e felizmente foram obtidos resultados satisfatórios. Um exemplo de implementação é um editor de textos para a Linguagem Brasileira de Sinais (LIBRAS), em que o usuário gestualiza o símbolo referente a uma letra do alfabeto LIBRAS em frente à câmera que captura as imagens; em seguida o computador processa as imagens e traduz o gesto para a linguagem natural exibindo, por fim a letra informada pelo usuário.

Contudo, na linguagem LIBRAS, assim como em outras linguagem corporais, a comunicação não se limita apenas em posturas, que são imagens sem movimento, mas também em gestos completos, o que torna o processo de reconhecimento computacional mais complexo. Entretanto, existem estudos na área de visão computacional que podem auxiliar este processo e conseqüentemente obter os resultados desejados. Para este problema específico uma das alternativas que auxiliaria sua solução seria a utilização de Modelos Ocultos de Markov, conforme será mostrado neste trabalho.

Capítulo 3

Objetivos

Estudar e desenvolver aplicações sobre os Modelos Ocultos de Markov aplicados ao reconhecimento de gestos humanos através da visão computacional, utilizando suas teorias e analisando sua eficácia através da aplicação em bancos de imagens.

- Conhecer os conceitos dos Modelos Ocultos de Markov e reconhecimento de gestos;
- Criar e analisar um banco de imagens de gestos da linguagem LIBRAS;
- Desenvolver e implementar um Modelo Oculto de Markov para o reconhecimento de gestos da linguagem LIBRAS;
- Desenvolver uma aplicação para a detecção de movimentos humanos através dos Modelos Ocultos de Markov;
- Adaptar e adicionar as implementações de reconhecimento de gestos através de HMM à plataforma SIGUS.

Capítulo 4

Revisão de Literatura

4.1 Visão Computacional e Reconhecimento de Padrões

A visão computacional, ou visão de máquina, é uma área de conhecimento que se dedica a desenvolver teorias e métodos voltados à extração automática de informações “úteis” contidas em imagens, sendo que a utilidade de uma informação é altamente dependente da aplicação. Por exemplo, em uma indústria de manufatura, que fabrica determinado produto, uma informação útil pode estar relacionada com alguma diferença de cor ou forma padrão, esperada para um produto sem defeitos.

Basicamente, sistemas de visão computacional capturam imagens através de dispositivos óptico-eletrônicos, como câmeras e filmadoras digitais, e buscam produzir descrições úteis das informações contidas nas imagens. Essas descrições podem ser utilizadas, por exemplo, na classificação de objetos ou no controle automático de algum dispositivo atuador, como um braço robótico ou uma rede de esteira rolantes em uma linha de montagem. Estas tarefas relativamente simples de serem realizadas por seres humanos, como diferenciar chaves de fenda de chaves alemãs em uma linha de montagem, apresentam-se como grandes desafios para sistemas automáticos de visão computacional.

Embora o reconhecimento de padrões possa ser aplicado a problemas sem qualquer relação com imagens e visão, existe uma rica intersecção entre essa área e a área da visão computacional, uma vez que o reconhecimento de padrões é uma importante etapa em boa parte dos problemas de visão computacional. Reconhecimento de padrões é a disciplina científica cujo objetivo é a criação de teorias e técnicas que permitam a classificação de objetos, ou padrões, dentre um conjunto de categorias ou classes [TK99]. Dependendo da aplicação, esses objetos podem ser imagens, seqüências de caracteres, sons ou qualquer outro tipo de sinal, geralmente digitalizado,

capturado através dos sensores de um sistema computacional. Um exemplo que se encaixaria nesta intersecção entre as duas áreas seria um sistema de tradução de linguagem corporal para natural, em que o usuário informaria comandos ao computador através de suas imagens capturadas em uma webcam. Existem trabalhos relacionados que têm como alvo a construção de sistemas de reconhecimento de língua de sinais para deficientes auditivos[Pis03].

Nestes sistemas geralmente existem duas etapas importantes: o reconhecimento de posturas e reconhecimento de gestos. Este reconhecimento se baseia na busca pelo modelo que possui características que melhor se assemelham com os parâmetros extraídos da imagem corrente. Posturas são sinais que não envolvem movimentação, ou seja, são estáticas, e por este motivo tornam sua modelagem mais simples comparada aos gestos. A modelagem dos gestos inclui informações temporais e análise de seqüências de imagens. A grande maioria dos trabalhos de reconhecimento de gestos utiliza técnicas adaptadas de reconhecimento da fala, como as baseadas em cadeias de Markov[Pis03].

4.2 Modelos Ocultos de Markov

Durante os últimos 15 anos, os Modelos Ocultos de Markov têm sido amplamente utilizados em diversas áreas, incluindo no reconhecimento de padrões[MO98]. Dentre elas podem ser destacadas as aplicações de reconhecimento de voz, em que predominam a utilização de modelos estatísticos, em destaque os baseados em Modelos Ocultos de Markov.[Yno99].

Antes de descrevermos os Modelos Ocultos de Markov (*HMM - Hidden Markov Models*) é necessário que saibamos sobre sua origem, os processos de Markov. Em meados de 1907, Markov definiu e investigou algumas propriedades que hoje são conhecidas como processos de Markov[CV02]. A principal característica dos processos de Markov é a definição que toda a história passada está resumida no valor atual do processo. Em alguns padrões geralmente existe uma estrutura que influencia a probabilidade do próximo evento ocorrer. Por exemplo, em alguns idiomas, como no Português e no Inglês, a probabilidade de se encontrar a letra u após ter detectado a letra q é muito alta, considerando que praticamente sempre após a letra q é encontrada a letra u . Um processo estocástico é chamado de processo de Markov de ordem j se a probabilidade do evento corrente ocorrer, dados todos os eventos anteriores e o presente, depende somente dos j eventos mais recentes.

Um Modelo Oculto de Markov é uma Cadeia de Markov onde os estados do modelo não são conhecidos, mas apenas o sinal emitido em cada unidade de tempo t . Um Modelo Oculto de Markov é um processo duplamente estocástico. A primeira camada estocástica é um processo de Markov de primeira ordem, e não é diretamente observável. Cada estado é uma possível

observação do processo de Markov, e uma probabilidade de transição de um estado A para outro B é a probabilidade de transitar para o estado B no tempo $t+1$ estando em A no tempo t . A segunda camada estocástica é um conjunto de probabilidades para cada estado do modelo que indicam a probabilidade de uma observação ocorrer estando em determinado estado. A seqüência de estados percorrida em um modelo, dada a seqüência de observações, é oculta ao observador. Ou seja, dada uma seqüência de saída, não se sabe a seqüência de estados percorrida pelo modelo, mas somente uma função probabilística deste caminho, e por isso o modelo é chamado de *Modelo Oculto de Markov* ou *Hidden Markov Models (HMM)*.

4.2.1 Elementos do HMM

Um Modelo Oculto de Markov geralmente é definido por uma tripla $\lambda = (A, B, \pi)$, e possui os seguintes elementos:

1. Número de estados do modelo (N), sendo que cada estado é denotado por $S = \{s_1, s_2, \dots, s_N\}$.
2. O número total de símbolos distintos reconhecidos (M), em que $V = \{v_1, v_2, \dots, v_M\}$ representa o alfabeto de sinais do modelo.
3. A quantidade de símbolos existentes na observação (T).
4. Um conjunto de estados $Q = \{q_1, q_2, \dots, q_T\}$, onde q_t é o estado no momento t .
5. Um conjunto de símbolos $O = \{o_1, o_2, \dots, o_T\}$, onde o_t é o símbolo observado no instante t .
6. Um conjunto de valores (π) que define as probabilidades de cada estado ser um estado inicial. $\pi = \{\pi_1, \pi_2, \dots, \pi_N\}$.
7. Uma matriz bidimensional $A_{N \times N}$ que possui os valores das probabilidades de transições entre os estados do modelo.
8. Um matriz bidimensional $B_{N \times M}$ que representa as respectivas probabilidades de ocorrências dos símbolos em cada estado do modelo.

Supondo um HMM onde existam 3 estados: S1, S2 e S3. A probabilidade de uma cadeia iniciar no estado S1 é de 25%, no S2 35%, e no S3 40%. Existem três símbolos possíveis que podem ser gerados em cada estado do modelo: $V = \{V1, V2 \text{ e } V3\}$. A probabilidade de, estando no estado S1, ocorrer um V1 é de 7%; de ocorrer um V2 é de 63%; e um V3 é 30%. Estando em S2 as probabilidades serão de¹ 30% para ocorrer um V1 e 70% de ocorrer um V3. Por fim,

¹Note que no estado S2 a probabilidade de ocorrer um V2 é nula.

estando em S3 as probabilidades serão de 83% de ocorrer um V1, 3% de ocorrer um V2 e 14% de ocorrer um V3. Estando no estado S1, existe 28% de chance que ele permaneça em S1 no próximo instante de tempo ($t + 1$), 53% de chance de ir para o estado S2, e 19% de chance de ir para S3; estando em S2 é impossível retornar a ele, e pode tanto ir para S1 quanto para S3 com a mesma probabilidade; finalmente, estando em S3 existe 14% de possibilidade que ele vá para o estado S1, 85% para S2 e 1% que fique onde está.

Do modelo acima descrito podem ser extraídos os seguintes valores para seus elementos:

$$N = 3 \text{ (referente a S1, S2 e S3);}$$

$$M = 3 \text{ (referente aos símbolos V1, V2 e V3)}$$

$$A = \begin{pmatrix} 0.28 & 0.53 & 0.19 \\ 0.50 & 0.00 & 0.50 \\ 0.14 & 0.85 & 0.01 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.07 & 0.63 & 0.30 \\ 0.30 & 0.00 & 0.70 \\ 0.83 & 0.03 & 0.14 \end{pmatrix}$$

$$\pi = [0.25 \quad 0.35 \quad 0.40]$$

4.2.2 Problemas Básicos

Existem três problemas principais implícitos no HMM que precisam de solução para que ele possa ser utilizado de maneira eficaz nas aplicações do mundo real[MO98]. Antes de descrevermos o problema é interessante sabermos que sempre que no decorrer do texto forem encontrados os símbolos λ ele se refere a um HMM; a letra O a uma seqüência de símbolos observados; e $P[O|\lambda]$ é a probabilidade da observação O ter sido gerada pelo modelo λ .

1. Problema de avaliação

Dada uma seqüência de observações O e um modelo λ , como é possível obter a probabilidade desta observação ter sido gerada pelo modelo? Este tipo de situação pode ser muito freqüente nas aplicações de HMM. Como exemplo, no reconhecimento de voz, ao se produzir um fonema qualquer, esta entrada será classificada como pertencente ao modelo que apresentar a maior probabilidade ($P[O|\lambda]$).

A maneira mais simples de calcular a probabilidade de uma observação ter sido gerada por um modelo é através da verificação de todas as seqüências de estados de tamanho

T (número total de observações) possíveis, e posteriormente calcular suas probabilidades. Considerando uma destas seqüências como $Q = \{q_1, q_2, \dots, q_T\}$, em que q_1 é o estado inicial. A probabilidade de uma seqüência de símbolos ter sido gerada por esta seqüência de estados Q é a seguinte:

$$P(O|Q, \lambda) = \prod_{i=1}^T b_{q_i}(O_i) \quad (4.1)$$

em que $b_{q_i}(O_i)$ é a probabilidade de se gerar um símbolo O_i estando no estado q_i . A probabilidade da seqüência de estados Q ocorrer pode ser representada pela seguinte equação:

$$P(Q|\lambda) = \pi_{q_1} \cdot a_{q_1, q_2} \cdot a_{q_2, q_3} \dots a_{q_{T-1}, q_T} \quad (4.2)$$

em que π_{q_i} é a probabilidade do estado inicial do modelo ser q_i e a_{q_i, q_j} a probabilidade de transição do estado q_i para q_j .

Por fim, a probabilidade de O ter sido gerada pelo modelo é a soma das duas probabilidades acima para todas as seqüências de estados de Q , da seguinte forma:

$$P(O|\lambda) = \sum_{all Q} P(O|Q, \lambda) \cdot P(Q|\lambda) \quad (4.3)$$

A probabilidade $P(O|\lambda)$ através das equações acima é obtida com cálculos de ordem computacional $2T \cdot N^T$, pois existem N possíveis estados que podem ser alcançados para cada $t = 1, 2, \dots, N$, ou seja, existem N^T possíveis seqüências de estados, e para cada seqüência desta $2 \cdot T$ cálculos. Estes cálculos são computacionalmente impraticáveis, até mesmo quando os valores de N e T são baixos. Obviamente é necessário um procedimento mais eficiente para resolver este problema. Este procedimento existe, e é chamado de procedimento *forward (backward)* [Rab89].

Este procedimento envolve duas variáveis: *forward* α e *backward* β . A variável *forward* α é descrita por $\alpha_t(i)$, sendo referente à probabilidade do estado no tempo t ser q_i através da seqüência de observações $O = \{o_1, o_2, \dots, o_t\}$ (até o tempo t), em um dado λ . É possível calculá-la recursivamente através dos seguintes passos:

(a) Inicialização

$$\alpha_1(i) = \pi_i \cdot b_i(o_1), 1 \leq i \leq N \quad (4.4)$$

(b) Indução

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right] \cdot b_j(o_{t+1}), 1 \leq t \leq T-1 \text{ e } 1 \leq j \leq N \quad (4.5)$$

(c) Terminação

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (4.6)$$

A variável *backward* $\beta(\beta_t(i))$ é a probabilidade da seqüência de observação de $t + 1$ até o fim, dado o estado S_i no tempo t e o modelo λ . Indutivamente *backward* β pode ser calculada da seguinte maneira:

(a) Inicialização

$$\beta_T(i) = 1, 1 \leq i \leq N. \quad (4.7)$$

(b) Indução

$$\beta_t(i) = \sum_{j=1}^N a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j), t = T - 1, T - 2, \dots, 1, 1 \leq i \leq N. \quad (4.8)$$

(c) Terminação

$$P(O|\lambda) = \sum_{i=1}^N \pi_i \cdot b_i(o_1) \cdot \beta_1(i) \quad (4.9)$$

Vale lembrar que apenas uma das variáveis α ou β são necessárias para solucionar o problema da avaliação.

2. Problema da busca da melhor seqüência de estados

Diferentemente do problema anterior em que uma solução exata pode ser dada, este problema possui várias formas de ser resolvido. Geralmente ele é resolvido utilizando o algoritmo de Viterbi[MO98], que localiza a melhor seqüência de estados $Q = \{q_1, q_2, \dots, q_T\}$ para a seqüência de observações $O = \{o_1, o_2, \dots, o_T\}$ dada.

(a) Inicialização

$$\alpha_1(i) = \pi_i \cdot b_i(o_1), 1 \leq i \leq N \quad (4.10)$$

$$\Psi_1(i) = 0 \quad (4.11)$$

(b) Recursão

$$\delta_{t+1}(j) = \left[\max_{1 \leq i \leq N} \delta_{t-1}(i) \cdot a_{ij} \right] \cdot b_j(o_t), 2 \leq t \leq T \text{ e } 1 \leq j \leq N \quad (4.12)$$

$$\Psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) \cdot a_{ij}], 2 \leq t \leq T \text{ e } 1 \leq j \leq N \quad (4.13)$$

(c) Terminação

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (4.14)$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \quad (4.15)$$

(d) Caminho (seqüência de estados) - *backtracking*

$$q_t^* = \Psi_{t+1}(q_{t+1}^*), t = T - 1, T - 2, \dots, 1 \quad (4.16)$$

3. Problema de treinamento

O terceiro e último problema é também o mais difícil de solucionar, que é encontrar um método para configurar os parâmetros $\lambda(A, B, \pi)$ do modelo de forma que atenda a um critério de otimização. A seqüência O utilizada para este ajuste é chamada de seqüência de treinamento, pois é utilizada para treinar o HMM. Não existe uma maneira conhecida de realizar este ajuste para resolver analiticamente o modelo que maximize a probabilidade da seqüência de observações. Porém, podemos escolher o modelo que sua probabilidade seja localmente maximizada usando um procedimento iterativo como o método de Baum-Welch. Para descrever o procedimento, primeiramente é definido $\xi_t(i, j)$, que é a probabilidade de estar no estado i no tempo t e estar em j no tempo $t + 1$, dado o modelo e a seqüência de observações. O valor de ξ é obtido pela equação abaixo:

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j | O, \lambda) \quad (4.17)$$

Através das definições das variáveis de *forward* e *backward*, pode-se encontrar $\xi_t^u(i, j)$ na forma:

$$\xi_t^u(i, j) = \frac{\alpha_t^u(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}^u(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t^u(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_t^u(j)} \quad (4.18)$$

em que u se refere a imagem. A probabilidade de estar em um estado em determinado tempo pode ser encontrada pela variável $\gamma_t(i)$, em que t e i são o tempo e o estado desejados. Sua definição é:

$$\gamma_t(i) = P(q_t = i | O, \lambda) \quad (4.19)$$

Esta probabilidade também pode ser obtida através das variáveis *forward* e *backward* da seguinte forma:

$$\gamma_t^u(i) = \frac{\beta_t^u(i) \alpha_t^u(i)}{P^u(O | \lambda)} \quad (4.20)$$

Desta forma, o modelo $\lambda(A, B, \pi)$ pode ser aproximado por $\bar{\lambda}(\bar{A}, \bar{B}, \bar{\pi})$, em que seus parâmetros são calculados através das seguintes fórmulas:

$$\bar{\pi}_i = \text{número de vezes no estado } S_i \text{ no tempo inicial (t=1)} = \gamma_1(i) \quad (4.21)$$

$$\begin{aligned} \bar{a}_{ij} &= \frac{\text{número esperado de transições do estado } S_i \text{ para o estado } S_j}{\text{número esperado de transições do estado } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \quad (4.22)$$

$$\begin{aligned} b_j(\bar{k}) &= \frac{\text{número de vezes no estado } S_j \text{ e observando o símbolo } v_k}{\text{número de vezes no estado } S_j} \\ &= \frac{s.t. O_t = v_k \sum_{t=1}^{T-1} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \quad (4.23)$$

Capítulo 5

Metodologia

1. Conhecer os conceitos dos Modelos Ocultos de Markov e reconhecimento de gestos;
 - (a) Leitura de artigos sobre os conceitos de HMM.
 - (b) Utilização de ferramentas prontas e implementações básicas sobre HMM.
 - (c) Leitura de artigos relacionados à área de reconhecimento de gestos através de visão computacional e aplicações de HMM.
2. Criar e analisar um banco de imagens de gestos da linguagem de sinais LIBRAS;
 - (a) Análise dos gestos que alimentarão as bases de imagens.
 - (b) Captura das imagens dos gestos através de uma câmera digital.
 - (c) Pré-processamento das imagens, com auxílio das implementações da plataforma SIGUS.
 - (d) Classificação manual dos gestos capturados e pré-processados.
3. Desenvolver e implementar um Modelo Oculto de Markov para o reconhecimento de gestos da linguagem LIBRAS e uma aplicação computacional para utilização do mesmo;
 - (a) Geração do Modelo Oculto de Markov através dos algoritmos existentes com base nos gestos armazenados no banco de imagens.
 - (b) Implementação de uma aplicação capaz de classificar os gestos de entrada.
 - (c) Testes das classificações obtidas através do modelo e da aplicação.
 - (d) Análise dos erros e acertos obtidos das classificações.
 - (e) Proposta de melhorias no modelo para minimização de erros no reconhecimento de gestos.

4. Criar e analisar um banco de imagens de um ambiente para a detecção de presença humana;
 - (a) Análise e captura das imagens dos gestos.
 - (b) Pré-processamento das imagens.
 - (c) Classificação manual dos gestos capturados e pré-processados.

5. Desenvolver uma aplicação para a detecção de movimentos humanos através dos Modelos Ocultos de Markov;
 - (a) Geração do Modelo Oculto de Markov através dos algoritmos existentes com base nos gestos armazenados no banco de imagens.
 - (b) Implementação de uma aplicação capaz de classificar os gestos de entrada.
 - (c) Testes das classificações obtidas através do modelo e da aplicação.
 - (d) Análise dos erros e acertos obtidos das classificações.
 - (e) Proposta de melhorias no modelo para minimização de erros no reconhecimento de gestos.

6. Adaptar e adicionar as implementações de reconhecimento de gestos através de HMM à plataforma SIGUS.

7. Realização das alterações necessárias para perfeita compatibilidade entre a plataforma e as implementações.

8. Testes de execução e criação de aplicação básica utilizando os recursos de reconhecimento de gestos através dos HMM.

Capítulo 7

Resultados e Impactos Esperados

Ao concluir o projeto utilizando as teorias sobre Modelos Ocultos de Markov estaremos buscando como resultado uma melhoria no reconhecimento de gestos, de modo que possa adaptar à plataforma SIGUS, podendo beneficiar diretamente o grupo de pessoas que possuem necessidades especiais, facilitar a interação homem-máquina e gerar um material que possa contribuir para pesquisas futuras.

Referências Bibliográficas

- [CV02] A. M. Cunha and L. Velho. Hidden markov models. *IEEE Computer Graphics and Applications*, 2002.
- [MO98] M. Morita and L. S. Oliveira. Introdução aos modelos escondidos de markov. November 1998.
- [Pis03] H. Pistori. *Tecnologia Adaptativa em Engenharia de Computação: Estado da Arte e Aplicações*. PhD thesis, Universidade de São Paulo, São Paulo, São Paulo, Brasil, 2003.
- [Rab89] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE Computer Graphics and Applications*, 77(2):42–53, 1989.
- [TK99] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern recognition*. Academic press, 1999.
- [Yno99] Carlos Alberto Ynoguti. *Reconhecimento de Fala Contínua Usando Modelos Ocultos de Markov*. PhD thesis, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, São Paulo, Brasil, 1999.