RASTREAMENTO E IDENTIFICAÇÃO DE MOVIMENTOS DE CABEÇA PARA SISTEMA DE COMUNICAÇÃO ALTERNATIVA

CARLOS WELLINGTON PASSOS GONÇALVES

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RASTREAMENTO E IDENTIFICAÇÃO DE MOVIMENTOS DE CABEÇA PARA SISTEMA DE COMUNICAÇÃO ALTERNATIVA

CARLOS WELLINGTON PASSOS GONÇALVES

DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.

APROVADA POR:

_____

Prof. Antônio Padilha Lanari Bo, ENE/UnB
(Orientador)

_____

Prof. Adriano de Oliveira Andrade, FEELT/UFU
Examinador Externo

_____

Prof. Geovany Araújo Borges, ENE/UnB
Examinador Interno

BRASÍLIA, 11 DE JULHO DE 2014.

FICHA CATALOGRÁFICA

GONÇALVES, CARLOS WELLINGTON PASSOS
Rastreamento e identificação de movimentos de cabeça para sistema de comunicação alternativa [Distrito Federal] 2014.
xi, 146p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2014).
Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. Rastreamento de imagens	2. Visão computacional
3. Comunicação Aumentativa e Alternativa	4. Cadeias de Markov
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

GONÇALVES, C. W. P. (2014). Rastreamento e identificação de movimentos de cabeça para sistema de comunicação alternativa, Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGEA.DM-565/14, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 146p.

CESSÃO DE DIREITOS

AUTOR: Carlos Wellington Passos Gonçalves
TÍTULO: Rastreamento e identificação de movimentos de cabeça para sistema de comunicação alternativa.
GRAU: Mestre ANO: 2014

Carlos Wellington Passos Gonçalves
Departamento de Eng. Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

Para Manuela.

AGRADECIMENTOS

Gostaria de agradecer imensamente todas as pessoas de contribuíram para que este trabalho se concretizasse. Sem a ajuda e a confiança do prof. Antônio Padilha e do prof. Rogério Richa este trabalho nunca teria saído do papel.

Agradeço demais Valéria Baldassin, Ricardo Mendes, Leandro Soares, Sylvio Moura, Derimar Gonçalves e toda a equipe da Bioengenharia do Hospital SARAH, unidade Lago Norte. Desde sempre me ajudaram e apoiaram no dia-a-dia antes e depois do ingresso neste mestrado. Muito obrigado também aos comitês de ciência e ética da Rede SARAH, bem como sua diretoria na aprovação da proposta de projeto e na ajuda durante elaboração dos documentos necessários à pesquisa.

Não posso deixar de agradecer todos os colegas de laboratório que em conjunto formam este ótimo ambiente no LARA.

Muito obrigado a minha ***super*** *esposa Manuela Ribeiro por embarcar nesta aventura comigo, em que juntos conseguimos enfrentar e curtir por dois anos. Muito obrigado minha famíla, minha querida mãe Suzana Passos e minha irmã Liudimila Passos.*

RESUMO

RASTREAMENTO E IDENTIFICAÇÃO DE MOVIMENTOS DE CABEÇA PARA SISTEMA DE COMUNICAÇÃO ALTERNATIVA

Autor: Carlos Wellington Passos Gonçalves

Orientador: Prof. Antônio Padilha Lanari Bo, ENE/UnB

Programa de Pós-graduação em Engenharia Elétrica

Brasília, 11 de julho de 2014

As interfaces de controle de computadores são concebidas desde suas origens para, aproveitando o controle motor altamente preciso e rápido de mãos, braços e dedos, permitir uma taxa de comandos muito rápida, e aumentar a eficiência da interação homem-máquina. Contudo, estima-se que no Brasil 7% da população tem algum tipo de deficiência motora, sendo que diversas destas podem comprometer ou inviabilizar o uso do computador por uma pessoa. O uso de tecnologias assistivas permite mitigar ou até anular as dificuldades inerentes às condições do usuário, empoderando-o a realizar atividades em que sua condição clínica é desfavorável. Sistemas de acesso ao computador e equipamentos para comunicação aumentativa e alternativa são empregados nesta tarefa ajudando pessoas em todo o mundo. Entretanto, pessoas com movimentação involuntária possuem um número reduzido de soluções à sua disposição para uso do computador, principalmente devido a poucos movimentos funcionais. Até o momento não existe um sistema de visão computacional que permita a interpretação de movimentos funcionais de cabeça de uma pessoa com movimentação involuntária. Este trabalho propõe uma aplicação que consegue reconhecer o rosto do usuário, rastrear seu movimento, modelar o movimento funcional desejado, e reconhecê-lo com o uso de imagens coletadas por uma *webcam* convencional. Nosso classificador é flexível e capaz de se ajustar aos diferentes movimentos testados, obtendo uma boa precisão e número pequeno de falso positivos. O sistema desenvolvido ainda pode ser integrado com vários softwares que implementam pranchas de comunicação dinâmicas, permitindo o uso do computador e comunicação através da escrita ou voz sintetizada. Resultados para candidatos hígidos comprovam que o os classificadores HMM produzem resultados superiores ou equivalentes a um classificador que utiliza apenas um limiar de posição. O melhor destes apresenta uma área abaixo da curva ROC de 0,997. Contudo, os módulos de segmentação e interpretação do movimento foram sensíveis a movimentação involuntária apresentada nos dois candidatos com deficiência motora da pesquisa, diagnosticados respectivamente com movimentação coreodistônica e coreoatetótica.

ABSTRACT

TRACKING AND IDENTIFICATION OF HEAD MOVEMENT FOR AUGMENTATIVE AND ALTERNATIVE COMMUNICATION

Author: Carlos Wellington Passos Gonçalves

Supervisor: Prof. Antônio Padilha Lanari Bo, ENE/UnB

Programa de Pós-graduação em Engenharia Elétrica

The control interfaces for computers are designed since its origins to take advantage of the fast and fine motor control of hands, arms and fingers, and allow a great rate of commands, increasing the efficiency of the human-machine interaction. However, it is estimated that 7% of the population of Brazil has some motor impairment, which can prevent the computer use by a subject. The use of assistive technologies can mitigate or reduce the difficulties related to the user's physical condition, empowering him to accomplish activities that are unfavorable to his condition. Computer access systems and augmentative and alternative communication equipment are employed in this task, helping people around the world. Even though, persons with involuntary movements have fewer solutions at hand to use a computer, especially because of few functional movements. So far there is no computer vision system that allows the interpretation of functional head movements of a person with involuntary movements. This work proposes an application that can recognize the user face, track its motion, model the desired function movement and recognize it with images collect via a conventional webcam. Our classifier has a flexible structure and is capable of dealing with the different movements that were tested, accomplishing good precision and a reduced false positive rate. The developed system can be integrated with softwares that implement dynamic communication boards, allow keyboard and mouse emulation, and communication by writing or synthesized voice. Results for healthy candidates show that the HMM classifiers have equal or better performance than an classifier that uses only a position threshold. The best of them presents an area below the ROC curve of 0,997. However, the motion segmentation and motion interpretation modules were sensitive to involuntary movements presented on the two candidates with motion impairment, diagnosticated respectively with choreodystonic and choreoathetotic movements.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE SÍMBOLOS

Símbolos Latinos

x	valor da posição horizontal do *pixel* numa imagem.
y	valor da posição vertical do *pixel* numa imagem.
$\boldsymbol{I}$	imagem formada por uma matriz de pixels.
$I(x, y)$	valor do *pixel* na posição (x, y).
$\boldsymbol{II}$	imagem integral.
$\boldsymbol{I}_C$	imagem classificada.
h	classificador de nó de árvore de decisão.
f	função característica.
p	polaridade.
ta	taxa de acerto.
tf	taxa de falso positivos.
Ta	taxa global de acertos.
Tf	taxa global de falso positivos.
C_{TV}	conjunto de exemplos de treino válido.
C_{TI}	conjunto de exemplos de treino inválido.
$\boldsymbol{T}$	*template* da imagem procurada.
R_1	*normalized cross correlation*.
R_2	*normalized correlation coefficient*.
$\boldsymbol{q}$	histograma das características da imagem alvo.
$\boldsymbol{p}$	histograma das características da imagem candidata.
k	função de *kernel*.
k_b	*kernel bandwidth*.
d	medida de similaridade.
$\boldsymbol{s}$	histograma das características da região de procura.
$\boldsymbol{\pi}$	vetor de probabilidades iniciais de uma HMM.
$\boldsymbol{A}$	matriz de transição de estados de uma HMM.
$\boldsymbol{B}$	matriz de observação de uma HMM.
S	estado da cadeia de Markov.
$\boldsymbol{O}$	sequência de observações.
$\boldsymbol{Q}$	sequência de estados.
$\boldsymbol{H}$	matriz de saída do sistema em espaço de estados.
L	verossimilhança.
LL	valor logarítmico da verossimilhança.

Símbolos Gregos

η	limiar de classificação em árvore de decisão.
ω	peso.
ϵ	função de custo para classificador em árvore de decisão Adaboost.
β	parâmetro de atualização de peso de classificador Adaboost.
δ	função delta de Kronecker.
ρ	função de similaridade de Bhattcharryya.
ϕ	função de similaridade PPM.
λ	cadeia de Markov.
α	probabilidade acumulada no sentido *forward* de uma cadeia de Markov.
β	probabilidade acumulada no sentido *backward* de uma cadeia de Markov.
$\boldsymbol{\theta}$	vetor de variáveis de espaço de estados.
$\boldsymbol{\Phi}$	matriz de estados do sistema de espaço de estados.

Subscritos

0	valor inicial.
$target$	valor a ser atingido.

Sobrescritos

$\bar{}$	valor médio.
$\hat{}$	valor estimado.
^{T}	transposto.
$'$	derivada

Siglas

SNPDP	Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência
SDHPR	Secretaria de Direitos Humanos da Presidência da República
CAT	Comitê de Ajudas Técnicas
TA	Tecnologia Assistiva
CTI	Centro de Tecnologia da Informação Renato Archer
CNRTA	Centro Nacional de Referência em Tecnologia Assistiva
SECIS	Secretaria de Ciência e Tecnologia para a Inclusão Social
AVD	Atividade de Vida Diária
AVP	Atividade de Vida Prática
CAA	Comunicação Aumentativa e Alternativa
EMG	Eletromiografia
CV	*Computer Vision*
HCI	*Human Computer Interfaces*
IR	Infravermelho
ELA	Esclerose Lateral Amiotrófica
LARA	Laboratório de Robótica e Automação
CONEP	Conselho Nacional de Ética em Pesquisa
CEP	Comitê de Ética em Pesquisa
TCLE	Termo de Consentimento Livre e Esclarecido
ROI	*Region of Interest*
fps	*frames per second*
pdf	*probability desity function*
PPM	*Posterior Probability Measure*
MOCAP	*Motion Capture*
EM	*Expectation Maximization*
DP	*Dynamic Programming*
DSTW	*Dynamic Space Time Warping*
GPLVM	*Gaussian Process Latent Variable Model*
HMM	*Hidden Markov Models*
ZVC	*Zero Velocity Crosses*
SO	Sistema Operacional
ROC	*Receiver Operating Characteristics*

Neste trabalho vetores são representados por letras minúsculas em negrito. Matrizes são representadas por letras maiúsculas em negrito.

Por exemplo, tem-se o vetor linha $\mathbf{v} = [v_x,\ v_y]$ e a matriz

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} . \tag{1}$$

Todas as representações de imagens aqui utilizadas serão assinaladas com negrito e letra maiúscula, justamente por serem representadas matematicamente por matrizes.

1 INTRODUÇÃO

For people without disabilities,
technology makes things easier.
For people with disabilities,
technology makes things possible.
Mary Pat Radabaugh

Este trabalho é destinado ao desenvolvimento de um *software* destinado a possibilitar o uso do computador por uma pessoa com deficiência. Neste capítulo será exposto como o uso de sistemas auxiliares podem mitigar as dificuldades inerentes à condição clínica de uma pessoa, e até eliminá-las. Também serão descritos os objetivos específicos deste trabalho e como é feita uma apresentação do manuscrito.

1.1 CONTEXTUALIZAÇÃO

Estima-se que 23,9% da população brasileira apresenta pelo menos algum tipo de de deficiência visual, motora, auditiva ou mental/intelectual [1]. Aproximadamente 7% de toda a população (cerca de 14 milhões de pessoas) apresenta alguma deficiência motora, sendo esta indicada pelos pesquisados como (i) tem alguma dificuldade, (ii) tem grande dificuldade e (iii) não consegue de modo algum movimentar-se.

No Brasil, a Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência, subordinada à Secretaria de Direitos Humanos da Presidência da República, tem como objetivo articular e coordenar as políticas públicas voltadas para a pessoa com deficiência [2]. Vinculada a esta secretaria trabalha o Comitê de Ajudas Técnicas, CAT, criado em 16 de novembro de 2006, pela Portaria nº 142, estabelecido pelo Decreto nº 5.296/2004.

O CAT é formado por integrantes de órgãos governamentais e especialistas nas áreas de promoção de serviços e produtos de acessibilidade (também denominados ajudas técnicas). Uma de suas primeiras finalidades foi a definição de uma terminologia que caracterizasse as práticas voltadas para a pessoa com deficiência. O termo Tecnologia Assistiva (TA) surgiu de uma profunda pesquisa que estudou como é normatizado o auxílio à pessoa com deficiência em vários países do mundo [3] [4].

Outro órgão público voltado para o estudo da aplicação de TA no Brasil é o Centro Nacional de Referência em Tecnologia (CNRTA)[5]. Criado em fevereiro de 2012, com o objetivo de criar um grupo de coordenação dos projetos nacionais na área de TA, principalmente no âmbito da ciência e tecnologia do programa Plano Viver sem Limite, este centro, incubado no Centro de Tecnologia da Informação Renato Archer (CTI), é uma ação da Secretaria de

Ciência e Tecnologia para a Inclusão Social (SECIS).

1.1.1 Tecnologia Assistiva

Em 2009 temos a primeira definição do que é englobado pelo termo Tecnologia Assistiva no Brasil.

> Tecnologia Assistiva é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social [3].

A Tecnologia Assistiva portanto é um termo muito abrangente, incluindo desde palmilhas ortopédicas, até a construção de rampas de acesso para cadeirantes. Existem algumas subdivisões dos equipamentos de TA, dentre as utilizadas na análise realizada pelo CAT citamos a ISO 9999, a classificação Horizontal European Activities in Rehabilitation Technology (HEART), e a classificação Nacional de Tecnologia Assistiva, do Instituto Nacional de Pesquisas em Deficiências e Reabilitação, dos Programas da Secretaria de Educação Especial, Departamento de Educação dos Estados Unidos [3].

Atualmente uma classificação de 1998 feita por José Tonolli e Rita Bersch está sendo utilizada pelo Ministério da Fazenda; Ciência, Tecnologia e Inovação e pela Secretaria Nacional de Direitos Humanos da Presidência da República na publicação da Portaria Interministerial Nº 362, de 24 de Outubro de 2012 que trata sobre a linha de crédito subsidiado para aquisição de bens e serviços de Tecnologia Assistiva destinados às pessoas com deficiência e sobre o rol dos bens e serviços. As categorias existentes dentro da TA seriam [4]:

- auxílios para atividades de vida diária (AVD) e vida prática (AVP);
- comunicação aumentativa e alternativa (CAA);
- recursos de acessibilidade ao computador;
- sistemas de controle de ambiente;
- projetos arquitetônicos para acessibilidade;
- órteses e próteses;
- itens para adequação postural;
- auxílios de mobilidade;

(a) (b)

Figura 1.1: Exemplos recursos de acessibilidade ao computador: (a) botão mecânico fixado em haste [8], (b) sistema de rastreamento ocular [9].

- auxílios para qualificação da habilidade visual e recursos que ampliam a informação a pessoas com baixa visão ou cegas;
- auxílio para pessoas com surdez ou déficit auditivo;
- recursos para mobilidade em veículos;
- recursos para esporte e lazer.

Neste trabalho iremos focar em duas categorias, CAA e recursos para acessibilidade ao computador.

1.1.2 Recursos para Acessibilidade ao Computador

Para utilização do computador com *mouse* e teclado convencionais, o usuário precisa ter destreza no movimento dos dedos e deve ter grande amplitude no movimentos da mão e do braço. Diversas condições clínicas implicam na perda do movimento ou na falta de controle dos membros superiores [6] [7]. Recursos de acessibilidade ao computador têm a função de contornar estas situações utilizando-se dos movimentos funcionais existentes do usuário. As figuras 1.1a e 1.1b apresentam dois exemplos de interfaces para acesso ao computador. A primeira muito utilizada é um botão mecânico, enquanto que a segunda é um complexo aparelho de rastreamento ocular.

1.1.3 Comunicação Aumentativa e Alternativa

Comunicação Aumentativa e Alternativa significa qualquer método de comunicação que suplemente os métodos ordinários de fala e escrita, onde estas funções estão debilitadas.

(a) (b)

Figura 1.2: Exemplos de equipamentos de CAA: (a) prancha de comunicação pictográfica [11], (b) prancha dinâmica embarcada em computador portátil [12].

A ideia de comunicação aumentativa é utilizar toda a capacidade de comunicação que uma pessoa deficiente pode ter [10]. Pranchas de comunicação pictográficas ou até pranchas dinâmicas com síntese de voz são exemplos de equipamentos de CAA. A figura 1.2 apresenta exemplos destes recursos.

A categoria se CAA confunde-se com a categoria de recursos para acesso ao computador quando as soluções para uso do computador (hardware ou software) são utilizadas para promover a comunicação. Um exemplo seria uso de um botão mecânico para acionar uma varredura de opções em um teclado virtual ou uma prancha dinâmica com o intuito de escrever um e-mail ou sintetizar a leitura de uma sentença. Por esta razão, neste trabalho optamos por utilizar o termo CAA para também referenciar os equipamentos voltados a permitir o uso do computador por pessoas com deficiência.

A comunicação alternativa e aumentativa é tratada como um disciplina clínica e científica desde as décadas de 1980 e 1990. O tipo e quantidade de alternativas oferecidas são variados e levam em conta aspectos físicos, cognitivos, linguísticos e as habilidades do usuário e acompanhantes de interagir e comunicar-se [13].

As formas de comunicação podem ser feitas sem auxílios, como por exemplo o uso de gestos ou sinais, ou podem ser utilizados sistemas que ajudam a indicar a intenção. Dentro destes sistemas a seleção ainda pode ser direta ou indireta. Em sistemas de seleção direta, o usuário aponta os caracteres ou pictogramas que deseja utilizar durante a comunicação utilizando seu movimento funcional, seja apontando com os dedos em uma prancha de comunicação impressa, seja indicando com o olhar a posição de itens numa prancha de comunicação transparente (*eye gaze board*), ou utilizando o cursor do mouse para selecionar letras em um software que reproduz uma prancha alfabética. Em sistemas de seleção indireta, é utilizada uma forma de varredura de opções para que o movimento funcional limitado do usuário selecione entre múltiplas opções [14].

(a) (b)

Figura 1.3: (a) Sistema de seleção direta que utiliza prancha de comunicação e ponteira *laser* [15]. (b) Sistema de seleção indireta com uso de acionamento por sinais EMG [16].

Na figura 1.3a temos um exemplo de uso direto, em que os itens da prancha de comunicação são selecionados via uma ponteira laser posicionada na cabeça do usuário, e na figura 1.3b temos um sistema que utiliza sinais EMG para acionar a varredura de uma prancha de comunicação implementada em um software.

É possível notar que soluções simples como um cartela alfabética, até complexas como um sistema de rastreamento ocular podem ser utilizadas de maneira eficiente para comunicação. Tal diferença de recursos implicou no surgimento de uma outra subdivisão entre os equipamentos de CAA. Equipamentos eletrônicos ou computacionais foram denominados de sistemas de alta tecnologia, enquanto que pranchas de comunicação impressas, cartelas, dentre outros, foram denominados de baixa tecnologia.

Levando-se em conta apenas aspectos físicos, em [17] foi proposto um fluxograma de seleção de recursos de alta tecnologia que está representado na figura 1.4. Neste podemos observar que três componentes principais devem ser avaliados, se existe algum movimento dos membros, se há movimentos com controle fino acima do pescoço, e se há algum movimento funcional em outras regiões do corpo. Para esta última, o nível de sofisticação dos aparelhos pode ser mais reduzido, já que a intenção do usuário é mais facilmente detectada pela amplitude dos movimentos. À medida que a amplitude do movimento funcional é reduzida, soluções mais sofisticadas precisam ser utilizadas.

O sistema de acesso ao computador mais difundido é a chave binária. Tal sistema pode ser constituído apenas de uma chave mecânica ou por sistemas de saída binária que utilizam inclinação, movimento, pressão do ar (*sip and puff*), som ou força. Esses equipamentos além de permitir o uso do computador, ainda podem ser utilizados para acionar cadeiras de rodas motorizadas e sistemas de controle de ambiente [17] [18].

As chaves mecânicas são amplamente utilizadas pela sua facilidade de acesso, robustez,

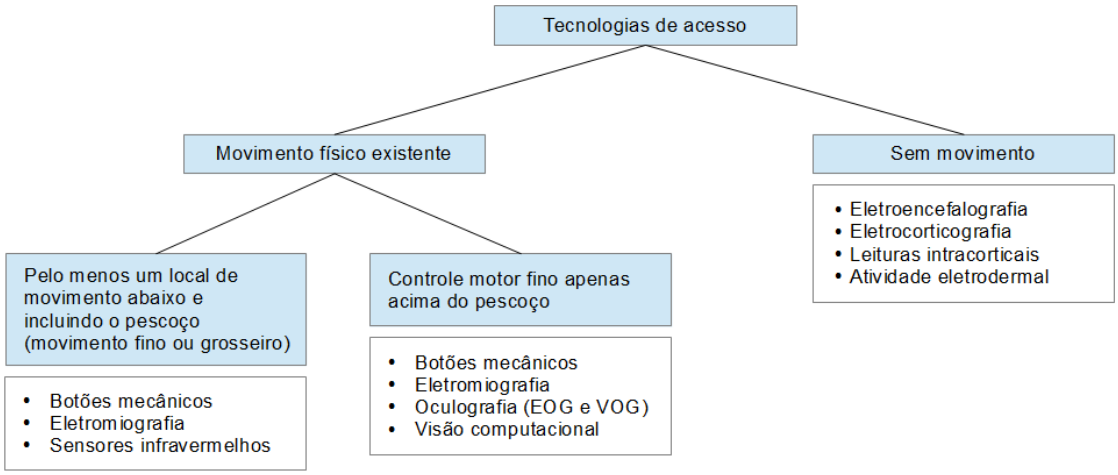


Figura 1.4: Fluxograma de escolha de sistemas CAA analisando apenas aspectos físicos do usuário, adaptado de [17].

bem como a simplicidade de uso. Este recurso sempre será utilizado em conjunto com outro equipamento responsável por efetivamente controlar o computador, escrever ou comunicar-se via sintetizador de voz. Pontos negativos do uso de botões mecânicos estão associados a mudanças nos padrões de movimentos do usuário que exigem reposicionamento das chaves, bem como a fixação destes dispositivos em cadeiras de rodas ou camas pode exigir montagens de hastes elaboradas [17].

Sistemas que utilizam visão computacional (*computer vision*, CV) estão cada vez mais acessíveis para permitir o controle do computador e comunicação. A disponibilidade de câmeras acopladas a *laptops*, *tablets*, celulares ou câmeras USB, bem como o aumento da memória RAM e velocidade dos processadores dos computadores pessoais criaram um cenário favorável à disseminação destes sistemas [17].

Acreditamos que é possível desenvolver programas que possam interpretar os movimentos de uma pessoa assim como é feito com os botões mecânicos, trazendo benefícios onde estes apresentam limitações. A seção 1.2 descreve como este trabalho pode contribuir no desenvolvimento de um novo sistema para CAA.

1.2 DEFINIÇÃO DO PROBLEMA

O uso de visão computacional na criação de sistemas para o uso do computador é muito amplo na área de interfaces homem-computador (*human computer interfaces*, HCI). Esta área fornece embasamento teórico na criação de aplicações mais abrangentes, que levam em conta limitações motoras para permitir o controle do computador, e a comunicação entre indivíduos [18].

Em equipamentos de acesso ao computador por pessoas deficientes, três métodos de

captura de movimento funcional se destacam: rastreamento ocular, rastreamento de face com uso de marcadores, rastreamento de face sem o uso de marcadores. Sistemas de rastreamento ocular utilizam câmeras infravermelho (IR) aliadas a emissores IR. A função dos emissores é iluminar a pupila de modo que esta seja de fácil identificação numa imagem IR. Os efeitos de pupila iluminada (*bright pupil*) e pupila escura (*dark pupil*) são aproveitados em várias aplicações [19].

Estes sistemas são implementados comumente por câmeras em miniatura fixadas em anteparos na própria cabeça do usuário, ou utilizando câmeras de maior resolução montadas próximas ao monitor do computador, como ocorre nos sistemas comerciais *Tobii PCEye, Dynavox Eyemax, EyeTech TM4* [18] [12] [20] [21].

Apesar de ser uma solução muito eficiente para pessoas que sofrem de doenças degenerativas como a esclerose lateral amiotrófica (ELA), em que o movimento ocular é preservado mesmo no estágio avançado da doença, estes sistemas são bastantes susceptíveis a movimentos de cabeça do usuário (principalmente nos casos de distonia, espasticidade ou tremores). Já quando implementados em montagens na cabeça do usuário são considerados invasivos.

Em sua maioria, soluções baseadas em IR são caras, necessitam de longas rotinas de calibração, e ainda há receio que a exposição prolongada à luz infravermelha possa ser danosa à visão [13] [18].

O rastreamento de face com o uso de marcadores também é usualmente feito com sistemas IR, em que um pequeno marcador IR pode ser colocado numa armação de óculos para ser utilizado por uma câmera IR de baixa resolução. Os movimentos de cabeça são então decodificados em movimentos proporcionais do cursor do *mouse* [22] [23] [24]. Estes sistemas podem também ser considerados invasivos por alguns usuários, além de também serem susceptíveis movimentos involuntários da cabeça [18].

Sistemas que não utilizam marcadores e baseiam-se em imagens coloridas adquiridas por *webcams* tem uma grande vantagem em relação aos citados acima pela grande disponibilidade de câmeras que atendem às especificações mínimas dos sistemas. Além de ser um sistema não intrusivo, no qual nenhum objeto precisa ser fixado no usuário ou à sua volta, e é apenas necessário que o usuário esteja com o rosto voltado à câmera. Podemos citar três projetos gratuitos de ampla utilização pela comunidade de CAA que apresentam estas características: *Headmouse* [25], *CameraMouse* [26], *Enable Viacam* [27]. Contudo, todos os sistemas existentes trabalham com a hipótese que o usuário apresenta controle fino dos movimentos da cabeça, o que permite um controle total das funções dos *mouse*.

Uma grande população que possui movimentos de cabeça funcionais ainda não é contemplada por um sistema compatível de visão computacional. A presença de movimento involuntário ou dificuldade no controle no tônus muscular implica na utilização de sistemas com poucas opções de acionamento, notadamente botões localizados próximos à cabeça. Desta maneira, além da tarefa de rastreamento facial, a interpretação correta dos movimen-

tos detectados deve ser feita, dificultando ainda mais a tarefa.

1.3 OBJETIVOS

Nosso projeto visa o desenvolvimento de um sistema de CAA baseado em *webcam* que possa se adaptar aos diversos tipos de movimentos funcionais de cabeça de uma pessoa com deficiência motora. Deste modo, teremos um equipamento totalmente customizável e com capacidade de escalabilidade, com possibilidade de facilmente utilizar mais de um movimento funcional. Diversos tipos de controle muscular bem como de patologias se beneficiam destas características. Esta abordagem é inédita na literatura e possui potencialmente grandes implicações futuras no desenvolvimento de sistemas de CAA.

Nosso objetivo principal é que o tipo de movimento normalmente captado por botões mecânicos, seja interpretado apenas com o uso de uma *webcam*. Esta nova alternativa pode solucionar algumas limitações encontradas pelos usuários de interfaces mecânicas, principalmente levando-se em conta a instalação em hastes de fixação em cadeiras de rodas e camas, e o acionamento indesejável na presença de movimentos involuntários.

Em segundo lugar, é nossa meta que o sistema proposto utilize ferramentas de código-livre e possa ser executado paralelamente a um teclado virtual destinado a produção de mensagens por varredura. Ao fim deste trabalho seu código fonte também estará disponível ao domínio público no servidor do Laboratório de Robótica e Automação - LARA da Universidade de Brasília.

Testes com pacientes da Rede SARAH de Hospitais de Reabilitação foram efetuados para agregar valor à pesquisa e comprovar sua aplicabilidade. A oportunidade de execução destes testes é de suma importância já que é carente a literatura de estudos de sistemas de visão computacional com pessoas deficientes [17].

1.4 TESTES COM PESSOAS

Este trabalho teve seu registro no CONEP, Comitê Nacional de Ética em Pesquisa, com o código CAAE de número 15055513.6.0000.0022. Foi aprovado no Comitê de Ética em Pesquisa (CEP) da Associação das Pioneiras Sociais-DF/Rede SARAH no parecer de número 508.817, bem como no parecer número 613.708 do CONEP.

As atividades propostas durante os testes com pessoas com deficiência foram detalhadas no decorrer deste trabalho e não trazem nenhum risco aos participantes. Todos assinaram e preencheram o termo de consentimento livre e esclarecido (TCLE) antes da participação neste trabalho. As identidades dos participantes foram preservadas através manipulação das imagens utilizadas, apenas imagens do autor estão sem tratamento.

1.5 APRESENTAÇÃO DO MANUSCRITO

Este trabalho segue dividido nos seguintes capítulos:

- Capítulo 2, Fundamentação Teórica. Neste capítulo são descritos os métodos que permitem a implementação de todas as partes deste trabalho. Uma revisão das teorias possivelmente aplicadas a cada problema é feita com um detalhamento das soluções utilizadas.
- Capítulo 3, Metodologia. Este capítulo detalha aspectos operacionais referentes a implementação dos módulos que compõem o sistema, com detalhes da implementação utilizando C++, MATLAB e a descrição do protocolo de teste com usuários.
- Capítulo 4, Resultados e Discussão. Utilizando-se de gráficos e tabelas, são apresentados os resultados obtidos através dos ensaios com usuários. Uma discussão dos pontos positivos e negativos do sistema é feita, já antevendo possíveis soluções.
- Capítulo 5, Conclusões. Este capítulo final busca sintetizar temas abordados neste trabalho, assim como concluir a interpretação dos resultados obtidos.

2 FUNDAMENTAÇÃO TEÓRICA

Trust The Computer.
The Computer is your friend.
"Computer's Credo", Paranoia

Neste capítulo estão descritos os trabalhos e teorias aplicadas relacionadas ao desenvolvimento de um sistema CAA que utiliza visão computacional e imagens provenientes de uma *webcam*. O objetivo principal é contextualizar os métodos existentes e implementados em outras pesquisas, assim como descrever aqueles utilizados neste trabalho.

Este capítulo está dividido em três seções que focam na descrição de técnicas de reconhecimento de faces, rastreamento de movimento da face e modelamento e classificação do movimento. Em todas estas seções, salvo algumas exceções, as técnicas apresentadas utilizam imagens digitais como entrada.

Imagens digitais são representadas como uma matriz $\boldsymbol{I}(x, y)$ de valores chamados *pixels* que compõe uma imagem. Vídeos digitais são estruturas que armazenam uma sequência de imagens consecutivas. Independente dos diferentes tipos de codificações utilizados, estas sempre são tratadas da mesma maneira, onde os eixos x e y tem sua origem no canto superior esquerdo e o valor $\boldsymbol{I}(x, y)$ representa o valor do *pixel* com dimensão igual ao número de canais que codifica a imagem [28]. A figura 2.1 exemplifica estas definições.

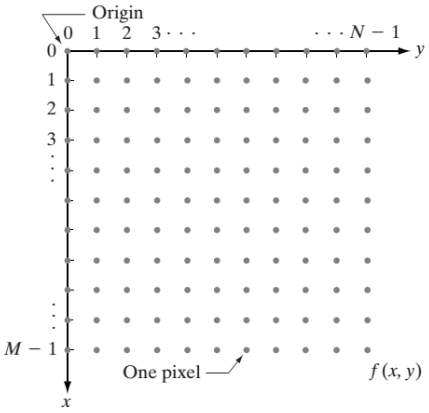


Figura 2.1: Definições de eixos x e y de uma imagem digital de tamanho $M \times N$, bem como do valor dos *pixels* [28].

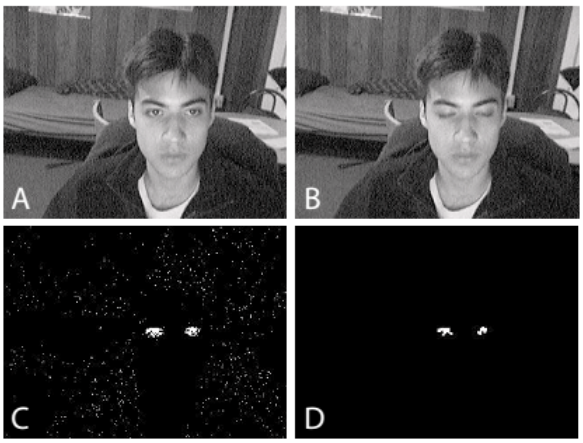


Figura 2.2: Método de detecção dos olhos numa imagem a partir da diferença entre *frames* consecutivos [30]. (a) Imagem antes do piscar. (b) Imagem durante o piscar. (c) Imagem resultante da subtração. (d) Imagem resultante depois da rotina morfológica de abertura.

2.1 RECONHECIMENTO DE FACES

Na criação de sistemas HCI que utilizam as informações sobre o estado da face, ou de componentes da face (olhos, sobrancelhas e boca), do usuário, existirá sempre a necessidade inicial de detectar a região de interesse (*region of interest*, ROI) de trabalho, determinada pela face do usuário.

Muitos sistemas interessados apenas no uso de determinadas regiões da face podem encontrá-las diretamente a partir da avaliação de alterações da imagem global. Tal abordagem pode ser vista em vários trabalhos voltados para uso da informação do estado dos olhos e posição destes para controle do computador [29] [30] [31].

A partir da imagem de diferença entre *frames* consecutivos, efetua-se uma operação morfológica de abertura para remover impurezas na imagem devido a variações de iluminação e condições de funcionamento da câmera. As regiões que contém *pixels* conexos, chamadas de *blobs*, que não foram removidas, são então segmentadas e rotuladas. Restrições biométricas são consideradas para filtrar as regiões correspondentes aos olhos. A figura 2.2 exemplifica este método. Restrições sobre a posição vertical da face e a existência de apenas uma face são comuns no uso destes métodos, bem como a necessidade que o usuário controle seus movimentos de cabeça muito bem.

A identificação completa da face ainda pode ser encontrada com um sistema que utiliza diferença entre *frames* mas tem acesso a informação sobre o *background* esperado durante o trabalho do usuário. Uma subtração consecutiva entre a imagem atual e a imagem de *background* retorna uma região de trabalho, *foreground*, contudo ainda pode conter muitos pontos situados fora da face do usuário. Ao subtrair novamente a imagem *foreground* em t da próxima imagem em $t + 1$, uma estimativa melhor da região de contorno do rosto é encontrada [32].

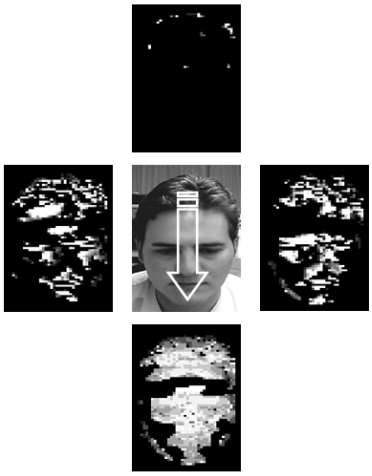


Figura 2.3: Resultado para a imagem contendo o fluxo óptico vertical e horizontal para o movimento de rotação da cabeça para baixo [33].

Mesmo sem ter conhecimento da imagem de *background*, a informação contida na diferença de duas imagens consecutivas devido ao movimento da face ainda pode ser tratada com base no uso de outra medida chamada de fluxo óptico. Em [33] uma implementação simplificada do fluxo óptico permite a detecção de movimentos verticais e horizontais da face.

Deste modo, afim de ser inicializada a posição inicial da face, o usuário deve rotacionar a cabeça verticalmente e horizontalmente. A região da imagem obtida pelo fluxo óptico definirá a ROI a ser utilizada em *frames* subsequentes. A figura 2.3 exemplifica o resultado obtido a partir do cálculo do fluxo óptico. Esta implementação também exige que não existam movimentos no plano de fundo da imagem durante a inicialização.

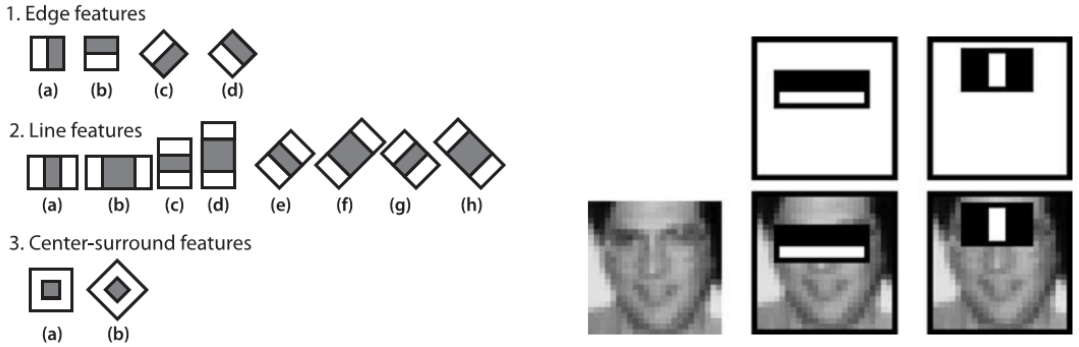
Muitos trabalhos ainda se utilizam de informação obtida *a priori* para detecção de face. Em [34] um banco de dados de treinamento dispondo do histograma de faces humanas é utilizado para encontrar quais *pixels* de uma imagem podem ser considerados pertencentes à face de um usuário. Uma binarização da região considerada como candidata a conter a face é base para uso de outras máscaras e filtros.

Em [35] uma binarização também é obtida, mas com a transformação da imagem para o padrão de cores YCbCr (*luma, chroma blue-difference, chroma red-difference*), e utilizando valores de limiares obtidos empiricamente. Após a binarização, o maior elemento conexo restante é tratado como a face.

2.1.1 Algoritmo Viola-Jones

O uso de imagens de treinamento também é utilizado numa das ferramentas mais utilizadas em detecção de face, conhecida como classificador Viola-Jones [36][1], sendo utilizado

[1]Trabalho já citado em 2444 trabalhos publicados, fonte Thomson Reuters - Web of Science



(a) (b)

Figura 2.4: Em (a) são apresentados exemplos das diferentes possibilidades de extração de características a partir da orientação de regiões onde há a soma de *pixels* e regiões onde há a subtração [40]. Em (b) vemos como algumas *features* podem caracterizar melhor determinadas regiões do rosto [36].

em [37] [38] [39]. Esse classificador é baseado em uma técnica de *boosting*, que usa uma cadeia de classificadores "fracos", que combinados se tornam um classificador discriminante de faces. A técnica de *boosting* adotada em [36] é o Adaboost.

Cada classificador "fraco"utiliza *Haar features*, que representam a diferença entre a soma de intensidades de duas regiões retangulares adjacentes. As figuras 2.4a e 2.4b exemplificam de maneira gráfica os diferentes tipos de *features* que podem ser utilizados. Pode-se notar que algumas *Haar features* podem caracterizar regiões onde há uma transição de *pixels* escuros para claros verticalmente (transição da região dos olhos para o nariz), ou existe uma região mais clara entre duas escuras (região das sobrancelhas).

A fase de treinamento exige que a base de dados seja rotulada com imagens de dimensões padrões (em [36] 24×24 pixels). Cada uma das imagens deve estar transformada nas dimensões predefinidas e codificada para níveis de cinza. Cada imagem candidata terá sua imagem integral definida para acelerar o cálculo do valor das características. A imagem integral é definida pela equação (2.1). Cada novo *pixel* da imagem integral terá o valor da soma dos *pixels* localizados no retângulo formado entre as posições $(0, 0)$ e (x, y).

$$\boldsymbol{II}(x, y) = \sum_{x_i \leq x, y_j \leq y} \boldsymbol{I}(x_i, y_j) \qquad , \tag{2.1}$$

onde $\boldsymbol{II}(x, y)$ corresponde ao valor da imagem integral e $\boldsymbol{I}(x, y)$ à imagem em tons de cinza.

Definindo a função de classificação de um classificador fraco, $h(\boldsymbol{I}_C, f, p, \eta)$, onde $\boldsymbol{I}_C$ corresponde à imagem avaliada, f uma função baseada no valor de *Haar features*, p a polaridade utilizada e η o limiar de comparação, temos:

$$h(\boldsymbol{I}_C, f, p, \eta) = \begin{cases} 1 & p \cdot f(\boldsymbol{I}_C) < p \cdot \eta, \\ 0 & \text{caso contrário.} \end{cases} \tag{2.2}$$

O algoritmo de Adaboost para treino de cada classificador pode então ser determinado a partir do pseudocódigo representado pelo algoritmo 1. Ao criar um peso $w_{t,i}$ para cada imagem de teste i para cada nível da árvore de decisão t, as imagens ainda não classificadas corretamente são efetivamente utilizadas nos níveis mais profundos da árvore de decisão. Atingida a taxa de acerto ta e a taxa de falsos positivos tf especificada inicialmente, o treino é encerrado, garantindo que poucos nós da árvore serão criados, ou seja, poucas funções de características serão computadas, o que acelera a detecção.

A partir de n amostras de dados $(\boldsymbol{I}_{Ci}, r_i), \ldots, (\boldsymbol{I}_{Cn}, r_n)$ nos quais $r_i = 0, 1$ para dados classificados como falsos e verdadeiros respectivamente.
Inicialize os n pesos correspondentes a cadas amostra $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ para $r_i = 0, 1$ respectivamente, onde m corresponde ao total de valores de treino falsos e l ao total de valores de treino verdadeiros.
for $t = 1, \ldots, T$ **do**
    normalização dos pesos: $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
    seleção do melhor classificador com base na função de custo:
    $\epsilon_t = min_{f,p,\eta} \sum_i^n w_{t,i} \left| h(\boldsymbol{I}_{Ci}, f, p, \eta) - r_i \right|$
    Definir a função de classificação do nó h_t a partir do resultado acima.
    Atualizar os pesos para o próximo nó:
    $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$
    onde $e_i = 0$ se $\boldsymbol{I}_{Ci}$ foi classificado corretamente, e 1 senão, e $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
end

Algoritmo 1: Treinamento de classificador com Adaboost.

Como cada estágio será responsável por eliminar imagens que não possuem faces e passar a diante imagens que possuem faces, para o estágio posterior, as entradas classificadas como válidas permanecem no grupo de treino válido, e as imagens classificadas como falsos positivos são colocadas no grupo de treino inválido, permitindo a especialização dos estágios mais profundos do classificador em imagens de difícil classificação. A figura 2.5 exemplifica o conceito desta árvore de decisão degenerativa (*degenerate decision tree*).

Para treino do classificador como um todo, é necessário que parâmetros de convergência globais sejam determinados, principalmente o número máximo de nós da árvore de decisão de cada estágio do classificador, taxa de acerto mínima e taxa de falsos positivos máxima. Para treinamento desta cadeia como completo vale lembrar que os critérios de parada para cada classificador são fáceis de serem atingidos.

Se for necessário construir um classificador com uma cascata com 10 estágios, com taxa de acerto de 0,9 e taxa de falsos positivos de 10^{-6}, basta que em cada estágio seja estabelecida uma taxa de acerto de 0,99 ($0,9 \approx 0,99^{10}$) e em contrapartida podem assumir até uma taxa

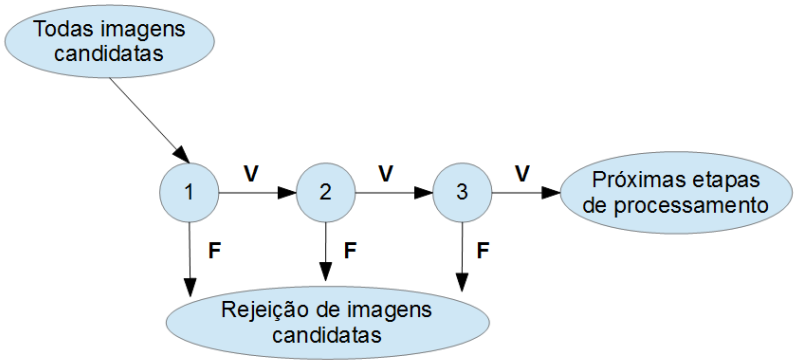


Figura 2.5: Esboço da cadeia de classificadores formando a árvore de decisão para detecção de face [36].

da falso positivos de 0,3 ($6 \times 10^{-6} \approx 0,3^{10}$). O algoritmo 2 demonstra como o classificador pode ser treinado.

Inicializar ta e tf para cada estágio.
Definir taxa global máxima de falso positivo Tf_{target}
C_{TV} = conjunto de exemplos de treino válido.
C_{TI} = o conjunto de exemplos de treino inválidos.
$Tf_0 = 1; Ta_0 = 1; i = 0;$
while $Tf_i > Tf_{target}$ **do**
    $i = i + 1; n_i = 0; Tf_i = Tf_{i-1};$
    while $Tf_i > tf \cdot Tf_{i-1}$ **do**
        $n_i = n_i + 1;$
        Utilizar C_{TV} e C_{TI} para treinar o classificador i com n_i características.
        Determinar Tf_i e Ta_i a partir do conjunto de validação.
        Diminuir η_i do classificador até que a taxa de detecção seja maior que $ta \cdot Ta_{i-1}$
    end
    Definir o conjunto de dados N como vazio.
    if $Tf_i > Tf_{target}$ **then**
        avaliar o classificador atual com o conjunto de dados inválidos e adicionar ao conjunto N apenas os falso positivos.
    end
end

Algoritmo 2: Treinamento da cadeia de classificadores para detecção de face.

A implementação final deste classificador ainda exige que múltiplas regiões candidatas

Figura 2.6: Varredura da imagem referência por toda uma nova imagem para localizar a nova posição [40].

sejam encontradas numa imagem independente da sua posição ou escala. Para implementar esta funcionalidade, múltiplas amostras são obtidas variando-se primeiramente a posição e depois a escala da imagem candidata, a partir do tamanho das imagens de treino (24×24). No caso de múltiplos candidatos com posições que se sobrepõem, apenas um será mantido, fruto da média dos sobrepostos.

2.2 RASTREAMENTO DO MOVIMENTO DA FACE

Com o objetivo de utilizar o movimento do rosto do usuário, ou de componentes da face como entrada de comando para um sistema, é preciso acompanhar a evolução destes durante o tempo. Dessa maneira, precisamos rastreá-los a cada novo *frame* fornecido pela câmera utilizada. Alguns dos procedimentos citados na seção 2.1 podem ser utilizados consecutivamente para outros *frames*, acompanhando o objeto rastreado. Contudo a taxa de atualização do vetor de estado pode ser baixa, impossibilitando o uso quando aplicado em conjunto com outros algoritmos.

Os autores de [36] reportam que conseguiram uma taxa de detecção de face de até 15 fps, contudo o uso contínuo desta ferramenta não é comum. Limitações quanto a taxa de atualização e características da base de imagens treinada podem levar a soluções não viáveis.

Uma das maneiras mais utilizadas para encontrar o elemento rastreado numa nova amostra de imagem é a comparação de padrões (*template matching*). Obtendo-se uma amostra do elemento a ser rastreado, este é comparado com todas as imagens em diferentes posições dentro da área de procura. Esta pode ser todo o novo *frame* ou uma região estabelecida da posição anterior do elemento rastreado. A figura 2.6 mostra como isso pode ser feito.

Esta forma de comparação leva em conta os valores dos *pixels* da imagem candidata e da

imagem alvo, assim como a posição destes em ambas imagens. As métricas utilizadas geram novas imagens respostas, onde o valor mais alto identifica a região de maior semelhança. Usualmente, são utilizadas as medidas de *normalized cross correlation*, R_1, e a *normalized correlation coefficient*, R_2, expressas nas equações (2.3) e (2.4) [41] [28]. Sendo $\boldsymbol{I}$ a imagem onde será realizada a busca pelo padrão $\boldsymbol{T}$ de tamanho $M \times N$ e definindo o operador $\sum_{x,y}$ como $\sum_{j=0}^{N-1} \sum_{i=0}^{M-1}$, temos:

$$R_1(x,y) = \frac{\sum_{x,y} \boldsymbol{I}(x+i,y+j).\boldsymbol{T}(i,j)}{\sqrt{\sum_{x,y} \boldsymbol{I}(x+i,y+j)^2 . \sum_{x,y} \boldsymbol{T}(i,j)^2}} \tag{2.3}$$

$$R_2(x,y) = \frac{\sum_{x,y} \left[\boldsymbol{I}(x+i,y+j) - \bar{\boldsymbol{I}}(x,y)\right] . \left[\boldsymbol{T}(i,j) - \bar{\boldsymbol{T}}\right]}{\sqrt{\sum_{x,y} \left[\boldsymbol{I}(x+i,y+j) - \bar{\boldsymbol{I}}(x,y)\right]^2 . \sum_{x,y} \left[\boldsymbol{T}(i,j) - \bar{\boldsymbol{T}}\right]^2}}, \text{ em que} \tag{2.4}$$

$$\overline{\boldsymbol{I}}(x,y) = \frac{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} \boldsymbol{T}(x+i,y+j)}{MN} \tag{2.5}$$

$$\overline{\boldsymbol{T}} = \frac{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} \boldsymbol{T}(i,j)}{MN} \tag{2.6}$$

A medida de *normalized cross correlation* pode ser observada numa série de trabalhos [35], [42] e [29], enquanto que a *normalized correlation coefficient* foi identificada nos trabalhos [43] ,[44], [28], [30], [34] e [37]. Como as equações mostram, a diferença principal entre as medidas seria a ponderação da semelhança em volta dos valores médios na equação (2.4).

Dentre as métricas de semelhança, existem ainda outras que não utilizam da informação local dos *pixels* numa imagem alvo, mas se beneficiam do comportamento de função de probabilidade do histograma do alvo. Este modelo permite que modificações no objeto rastreado, como oclusão e rotação não alterem de forma significativa o valor da função de probabilidade.

Em [45], alvos e candidatos a alvo foram representados pela função de densidade probabilística de seu vetor de característica, neste caso as cores. Foi utilizada uma função de *kernel* para aumentar a sensibilidade do histograma das intensidades de cores em relação a posição dos *pixels* que compõe na imagem alvo. Desta maneira, alterações pequenas na posição do alvo, que resultariam em mudanças muito pequenas no histograma, podem ser detectadas, melhorando a precisão.

Tomando uma estimativa de *pdf* do vetor característica da imagem alvo, $\hat{\boldsymbol{q}}$, e da imagem candidata de mesma dimensão e centrada em $\boldsymbol{y}$, $\hat{\boldsymbol{p}}(\boldsymbol{y})$, obtidas com histogramas com m subdivisões, temos:

$$\hat{\boldsymbol{q}} = \{\hat{q}_u\}_{u=1\ldots m} \quad \text{, em que } \sum_{u=1}^{m} \hat{q}_u = 1 \tag{2.7}$$

$$\hat{\boldsymbol{p}}(\boldsymbol{y}) = \{\hat{p}_u(\boldsymbol{y})\}_{u=1\ldots m} \quad \text{, em que } \sum_{u=1}^{m} \hat{p}_u(\boldsymbol{y}) = 1 \tag{2.8}$$

Aplicando-se o perfil de *kernel* de Epanechnikov $k(\cdot)$ [46], e utilizando a posição dos seus *pixels* normalizada em função do comprimento h_x e largura h_y, obtendo-se $\boldsymbol{x_i^*}$, podemos encontrar os valores de $\hat{q}_u$ e $\hat{p}_u(\boldsymbol{y})$ com as seguintes equações:

$$\hat{q}_u = C\sum_{i=1}^{n} k\left(\left\|\boldsymbol{x_i^*}\right\|^2\right)\delta\left[\tau\left(\boldsymbol{x_i^*}\right)-u\right] \quad \text{, em que} \tag{2.9}$$

$$C = \frac{1}{\sum_{i=1}^{n} k\left(\left\|\boldsymbol{x_i^*}\right\|^2\right)}, \tag{2.10}$$

$$\hat{p}_u(\boldsymbol{y}) = C_h\sum_{i=1}^{n_h} k\left(\left\|\frac{\boldsymbol{y}-\boldsymbol{x_i}}{k_b}\right\|^2\right)\delta\left[\tau\left(\boldsymbol{x_i}\right)-u\right] \quad \text{, em que} \tag{2.11}$$

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{\boldsymbol{y}-\boldsymbol{x_i}}{k_b}\right\|^2\right)}. \tag{2.12}$$

A função $\tau(\boldsymbol{x})$ indica o índice correspondente ao histograma da imagem, variando de $1\ldots m$, e é utilizada dentro da função δ de Kronecker. Para cada valor de $\hat{q}_u$ ou $\hat{p}_u(\boldsymbol{y})$, apenas os *pixels* mapeados com índice u terão seu valor utilizado. Já para o modelo do candidato, a diferença $\boldsymbol{y}-\boldsymbol{x_i}$ é utilizada e normalizada pelo raio da função de *kernel* (*bandwidth*) k_b [46].

A partir das estimativas das *pdfs* é possível utilizar uma medida de similaridade entre as imagens. Um das métricas se baseia no coeficiente de Bhattacharyya [45] [47]. Definindo uma medida de distância $d(\boldsymbol{y})$ temos:

$$d(\boldsymbol{y}) = \sqrt{1-\rho[\hat{\boldsymbol{p}}(\boldsymbol{y}),\hat{\boldsymbol{q}}]}, \tag{2.13}$$

$$\hat{\rho}(\boldsymbol{y}) \equiv \rho[\hat{\boldsymbol{p}}(\boldsymbol{y}),\hat{\boldsymbol{q}}] = \sum_{u=1}^{m}\sqrt{\hat{p}_u(\boldsymbol{y})\hat{q}_u}. \tag{2.14}$$

Podemos ver que o melhor candidato a alvo é aquele que minimiza a função (2.13), ou seja, que maximiza a função (2.14). Admitindo-se que possuímos uma estimativa inicial da posição do alvo, $\hat{\boldsymbol{y}}_{\mathbf{0}}$, a aproximação da função (2.14) por uma série de Taylor com seus dois primeiros termos é:

$$\rho[\hat{\boldsymbol{p}}(\boldsymbol{y}),\hat{\boldsymbol{q}}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{\boldsymbol{y}}_{\mathbf{0}})\hat{q}_u} + \frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(\boldsymbol{y})\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\boldsymbol{y}}_{\mathbf{0}})}}. \tag{2.15}$$

Apenas o segundo termo da expressão (2.15) pode ser maximizado em função de uma nova posição $\boldsymbol{y}$. Substituindo apenas este termo com o resultado da (2.11), temos que

$$\frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(\boldsymbol{y})\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\boldsymbol{y}}_{\mathbf{0}})}} = \frac{C_h}{2}\sum_{i=1}^{n_h}\omega_i k\left(\left\|\frac{\boldsymbol{y}-\boldsymbol{x_i}}{h}\right\|^2\right), \text{ em que} \tag{2.16}$$

$$\omega_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\boldsymbol{y}}_{\mathbf{0}})}}\delta[\tau(\boldsymbol{x_i})-u]. \tag{2.17}$$

O ponto $\hat{\boldsymbol{y}}_1$ que maximiza (2.16) pode ser encontrado iterativamente com o uso do algoritmo de *Mean-Shift*. Tendo em mãos a derivada do perfil do *kernel*, $\frac{\partial k}{\partial x \partial y}(\boldsymbol{x})$, temos

$$\hat{\boldsymbol{y}}_1 = \frac{\sum_{i=1}^{n_h} \boldsymbol{x_i} \omega_i g\left(\left\|\frac{\hat{\boldsymbol{y}}_0 - \boldsymbol{x}_i}{k_b}\right\|\right)}{\sum_{i=1}^{n_h} \omega_i g\left(\left\|\frac{\hat{\boldsymbol{y}}_0 - \boldsymbol{x}_i}{k_b}\right\|\right)} \text{, em que} \tag{2.18}$$

$$g(\boldsymbol{x}) = -\frac{\partial k}{\partial x \partial y}(\boldsymbol{x}) \tag{2.19}$$

A estimativa a cada passo do *mean-shift* leva a solução no sentido negativo da derivada do *kernel*. Esta subida dos valores de similaridade convergirá num valor próximo aquele que melhor representa o alvo numa região próxima à estimativa anterior.

Esta abordagem do uso de funções que mapeiam o comportamento estatístico é muito utilizada em diversos outros trabalhos, podemos citar em especial [48] [38] [49] [50]. Alguns dos principais benefícios desta abordagem é que ela é computacionalmente viável para que taxas de *fps* altas sejam obtidas. O cálculo do histograma para as imagens candidatas é simples e o cálculo de $\hat{\boldsymbol{q}}$, se não haver a necessidade de atualizar o modelo do alvo, precisa ser feito apenas uma única vez.

Em [47] os autores propuseram uma alteração na função de similaridade e obtiveram bons resultados, sua principal motivação é que o coeficiente de Bhattcharyya pondera da mesma maneira valores de *pixels* do alvo que são mais comuns no plano de fundo (*background*) da imagem.

Os autores denominaram a sua nova função de similaridade como *Posterir Probability Measure* (PPM) e esta é calculada pelos histogramas não normalizados do alvo, $\boldsymbol{q}$, do candidato, $\boldsymbol{p}$, e de uma área de procura que contém a imagem candidata, $\boldsymbol{s}$. A partir destes temos:

$$\phi(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{n} \sum_{u=1}^{m} \frac{p_u q_u}{s_u} \text{, em que} \tag{2.20}$$

$$\boldsymbol{q} = \sum_{u=1}^{m} q_u \quad , \; p_u = \sum_{i=1}^{n} \delta([\tau(\boldsymbol{x_i}) - u]) \,, \tag{2.21}$$

$$\boldsymbol{p} = \sum_{u=1}^{m} p_u \quad , \; q_u = \sum_{i=1}^{n} \delta([\tau(\boldsymbol{x_i}) - u]) \,, \tag{2.22}$$

$$\boldsymbol{s} = \sum_{u=1}^{m} s_u \quad , \; s_u = \sum_{i=1}^{n_s} \delta([\tau(\boldsymbol{x_i}) - u]) \,, \tag{2.23}$$

$$\tag{2.24}$$

Esta medida de similaridade ainda possui uma característica computacionalmente muito apreciável. A partir da definição de (2.22) é possível encontrar um valor de $\phi(\boldsymbol{p}, \boldsymbol{q})$ onde não

é necessário o cálculo de $\boldsymbol{p}$. Temos que

$$\phi(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{n} \sum_{i=1}^{n} \frac{q_\kappa}{s_\kappa}, \text{ em que} \tag{2.25}$$

$$\kappa = \tau(\boldsymbol{x_i}) . \tag{2.26}$$

Na figura 2.7 podemos ver uma comparação do resultado da função de similaridade PPM e o coeficiente de Bhattcharyya. Percebe-se que ao redor do candidato a alvo, a curva de similaridade de PPM é mais íngreme, melhorando a precisão, e não se deteriora tanto a medida que aumentamos a área do modelo do alvo. Quanto maior o número de elementos do plano de fundo neste modelo, piores serão os resultados utilizando o coeficiente de Bhattcharyya.

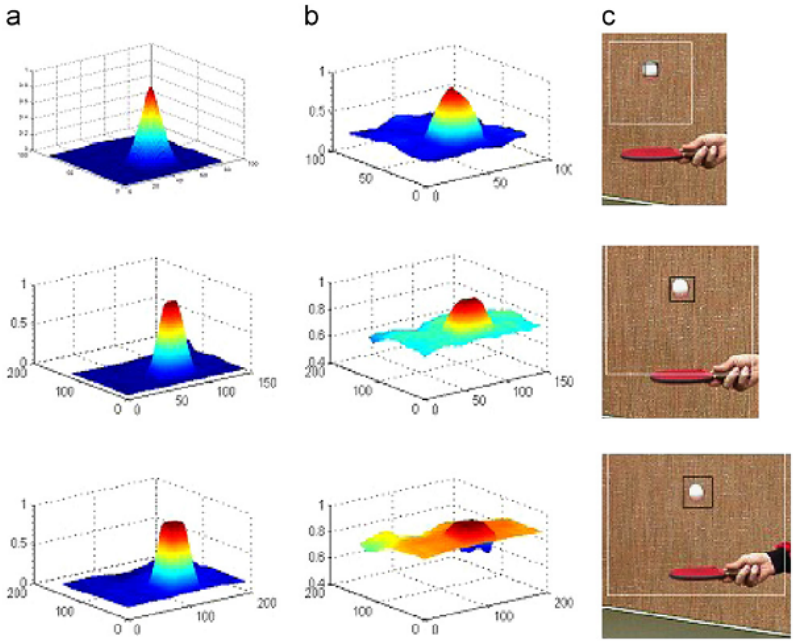


Figura 2.7: Comparação entre as funções de similaridade utilizando o (a) PPM e o (b) coeficiente de Bhattcharyya, a medida que a área do modelo do alvo aumenta (c) [47].

Para realizar a atualização da posição do melhor candidato a alvo, é utilizada uma adaptação do algoritmo de *mean-shift*, temos:

$$\hat{\boldsymbol{y}} = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i \omega_\kappa}{\sum_{i=1}^{n} \omega_\kappa}, \text{ em que} \tag{2.27}$$

$$\omega_\kappa = \frac{q_\kappa}{s_\kappa}, \text{ em que} \tag{2.28}$$

$$\kappa = \tau(\boldsymbol{x_i}) . \tag{2.29}$$

Na figura 2.8 podemos novamente comparar a performance entre o PPM e o coeficiente de Bhattcharyya, agora utilizando o algoritmo de *mean-shift* para rastreamento. Percebe-se que a estimativa utilizando PPM é mais precisa.

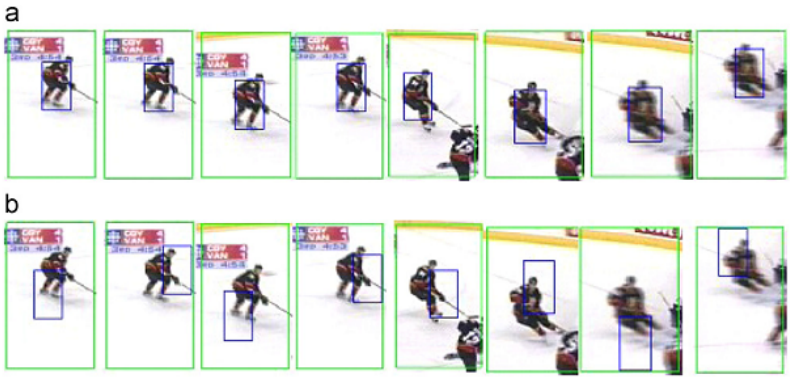


Figura 2.8: Comparação do rastreamento do jogador de *hockey* como (a) PPM e o (b) coeficiente de Bhattcharyya [47].

2.3 MODELAMENTO E CLASSIFICAÇÃO DO MOVIMENTO

A partir da medição da trajetória dos segmentos corporais, bem como estruturas menores como elementos da face e dedos das mãos, é possível utilizar diferentes técnicas para detectar e classificar os movimentos realizados. A partir de modelos matemáticos que descrevem o movimento, ou classificadores treinados para interpretá-los, é possível obter métricas de comparação, facilitando a aplicação em tarefas de alto nível, como a interpretação de dados de caminhada, análise de movimento de atletas, reconhecimento de comandos por sinais.

No âmbito esportivo e da reabilitação, é comum o uso de sistemas de de captura de movimento (*mocap*) para registrar do movimento dos membros envolvidos em uma atividade, e dos ângulos das articulações que produzem estes movimentos. Em [51], através de dados coletados com vinte marcadores reflexivos em bailarinas, foi possível extrair nove *movemas*, subestruturas que compõe um movimento completo de um passo de ballet, assim como fonemas que formam uma palavra.

Desta maneira, a partir de movimentos mais simples, é possível compreender e identificar movimentos mais complexos, de uma atividade com mais de 800 movimentos diferentes. Para cada um dos *movemas* foi estipulado um conjunto de variáveis que melhor os determinava. Este conjunto foi definido como um subespaço 2D do espaço de fase (*phase space*) obtido com os dados de posição e velocidade de cada segmento, e ângulo de articulação capturado. Um polinômio de terceiro grau é ajustado para representar os pontos deste espaço 2D utilizados para treinamento. Com esta função de similaridade, é possível estabelecer uma limiar de distância que define um ponto como pertencente a este *movema*.

Com o uso de um traje com sensores inerciais, em [52] é estudada a possibilidade de encontrar uma impressão digital dinâmica a partir do caminhar que possa diferenciar duas pessoas. Para isso uma árvore de decisão é utilizada com um vetor de características que englobam os ângulos de orientação dos pés, ângulos dos pés, joelhos, coxas, ombros e braços.

Em [53], diversas soluções são apresentadas pelo grupo de computação de imagem e

vídeo da universidade de Boston na interpretação de gestos em diferentes escalas. Para reconhecimento de movimentos da mão de grande amplitude, uma comparação entre modelo e candidato utilizando *Dynamic Space Time Warping*, DSTW é utilizada, enquanto que para reconhecimento de gestos estáticos da linguagem de sinais americana, é utilizado uma sistema de busca baseado em *BoostMap*. Para reconhecimento de movimentos amplos atribuídos a controladores de aeronaves no solo, uma redução dos espaço de características é proposta utilizando GPLVM, *Gaussian Process Latent Variable Model*.

Aplicações de redes neurais artificiais também são encontradas. Em [54], redes neurais com estrutura variável são empregadas no desenvolvimento do software SymbolDesign, permitindo que o usuário possa construir um interpretador de movimentos com número variável de comandos possíveis.

2.3.1 Cadeias de Markov

Dentre as ferramentas de modelamento estatístico do movimento, podemos destacar o uso de cadeias de Markov escondidas, *Hidden Markov Models*, HMM [55]. Com uma vasta aplicação na área de reconhecimento de fala, as cadeias de Markov abstraem numa única estrutura a chance de uma amostra estar representada em um dos estados da cadeia num determinado instante de tempo, a sequência de estados mais provável a partir de uma sequência de amostras, e a probabilidade de uma sequência de amostras pertencer a um modelo. Alia-se às estas características favoráveis o fato de apresentar um treinamento supervisionado de seus parâmetros, e permitir a utilização de vetores de características com distribuição discreta ou contínua.

Sendo v_i uma variável que pode assumir os valores pertencentes a um conjunto $\boldsymbol{V} = \{v_1, v_2, \ldots, v_M\}$ de tamanho M, uma cadeia de Markov de N estados $\boldsymbol{S} = \{S_i, \ldots, S_N\}$, que modela um sistema observável por meio dos valores v_i, apresenta três componentes que a definem: um vetor de probabilidade inicial $\boldsymbol{\pi}$ para cada um dos estados, uma matriz de transição de estados $\boldsymbol{A}$ que representa a probabilidade de transição entre os estados, e uma matriz de observação $\boldsymbol{B}$ que define a probabilidade de observação de v_i para cada um dos estados de $\boldsymbol{S}$. Representando o estado da cadeia no instante t como q_t, temos que a cadeia $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ apresenta as seguintes características:

$$\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_i, \ldots, \pi_N\}, \tag{2.30}$$

$$\pi_i = P[q_i = S_i], \tag{2.31}$$

$$\sum_{i=1}^{N} \pi_i = 1, \tag{2.32}$$

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \ldots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \ldots & a_{NN} \end{bmatrix}, \tag{2.33}$$

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \tag{2.34}$$

$$\sum_{i=1}^{N} a_{ij} = 1\,, \tag{2.35}$$

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & \dots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{N1} & \dots & b_{NM} \end{bmatrix}, \tag{2.36}$$

$$b_{ij} = b_i(j) = P[v_j \text{ em } t | q_t = S_i], \tag{2.37}$$

$$\sum_{i=1}^{N} b_{ij} = 1\,, \tag{2.38}$$

$$\sum_{j=1}^{M} b_{ij} = 1\,. \tag{2.39}$$

Na figura (2.9) podemos observar diferentes tipos de arquitetura de uma cadeia de Markov que podem ser utilizadas para modelar a evolução de um sistema. Uma cadeia em que existe a possibilidade de transição entre qualquer um dos estados é chamada de ergótica, enquanto que cadeias que modelam características sequenciais dos estados são construídas com modelos com arquitetura *left-right* ou paralela.

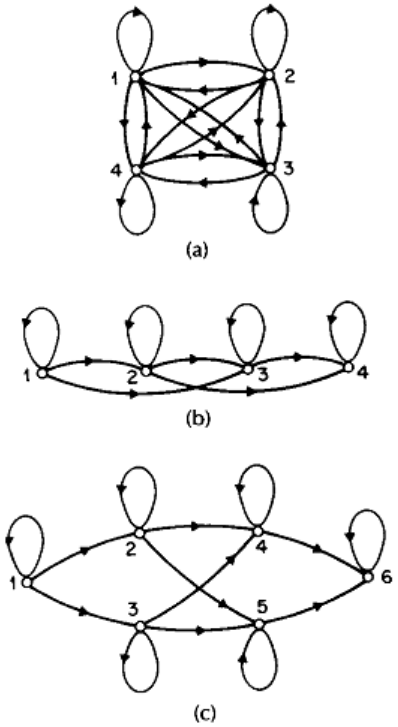


Figura 2.9: Tipos diferentes de arquitetura de uma HMM, (a) cadeia ergótica, (b) cadeia sequencial, (c) cadeia paralela [55].

Definida uma cadeia de Markov, esta passa a representar o comportamento de uma variável observável, de forma a indicar como a evolução dos seus valores pode estar relacionada

a transição entre os estados da cadeia. Este comportamento é extremamente desejável ao se modelar o movimento humano, seja da marcha [56], de mãos [57] ou de cabeça [58] [59].

Além disso, a formulação das cadeias de Markov ainda permite que seja calculada a probabilidade de uma sequência de eventos observados $\boldsymbol{O} = \{O_1, \ldots, O_T\}$ seja representada pela cadeia λ, $P(\boldsymbol{O}|\lambda)$, chamado de problema 1; permite que dada uma sequência de eventos observados $\boldsymbol{O}$, seja possível determinar a melhor sequência de estados $\boldsymbol{Q} = \{q_1, \ldots, q_T\}$ que poderia gerar esta sequência, chamado de problema 2; e finalmente é possível adequar os valores dos parâmetros da cadeia, $\boldsymbol{A}$, $\boldsymbol{\pi}$ e $\boldsymbol{B}$ para otimizarem a detecção de uma sequência $\boldsymbol{O}$, problema 3.

2.3.1.1 Solução do problema 1

A solução do problema 1 envolve o cálculo de todas as probabilidades das possíveis combinações de sequências de estados $\{\boldsymbol{Q_1}, \ldots, \boldsymbol{Q_{N^T}}\}$. Temos que a probabilidade de uma sequência $\boldsymbol{O}$ ser observada numa sequência de estados $\boldsymbol{Q}$:

$$P(\boldsymbol{O}|\boldsymbol{Q}, \lambda) = \prod_{t=1}^{T} P(O_t|q_t, \lambda)\,, \text{ a partir de (2.38)} \tag{2.40}$$

$$P(O_t|q_t, \lambda) = \prod_{t=1}^{T} b_{q_t}(O_t)\,. \tag{2.41}$$

A probabilidade que uma determinada sequência $\boldsymbol{Q}$ ocorra é:

$$P(\boldsymbol{Q}|\lambda) = \pi_{q_1} \prod_{t=2}^{T} a_{q_{t-1}q_t}\,. \tag{2.42}$$

Deste modo a probabilidade conjunta de $P(\boldsymbol{O}, \boldsymbol{Q}|\lambda)$ é:

$$P(\boldsymbol{O}, \boldsymbol{Q}|\lambda) = P(\boldsymbol{O}|\boldsymbol{Q}, \lambda)P(\boldsymbol{Q}|\lambda)\,. \tag{2.43}$$

Para obter a probabilidade de observação da sequência $\boldsymbol{O}$ para todas as possíveis combinações de sequências $\boldsymbol{Q}$, basta somar os valores de (2.43):

$$P(\boldsymbol{O}|\lambda) = \sum_{todas\boldsymbol{Q}} P(\boldsymbol{O}|\boldsymbol{Q}, \lambda)P(\boldsymbol{Q}|\lambda)\,, \tag{2.44}$$

$$P(\boldsymbol{O}|\lambda) = \sum_{todas\boldsymbol{Q}} \pi_{q_1} b_{q1}(O_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{qt}(O_t)\,. \tag{2.45}$$

Este cálculo é computacionalmente muito custoso, já que todas as transições possíveis da cadeia λ devem ser exploradas. Contudo com o uso do procedimento *forward-backward*, o custo computacional é reduzido. O passo *forward* implica em acumular numa matriz $\boldsymbol{\alpha}$

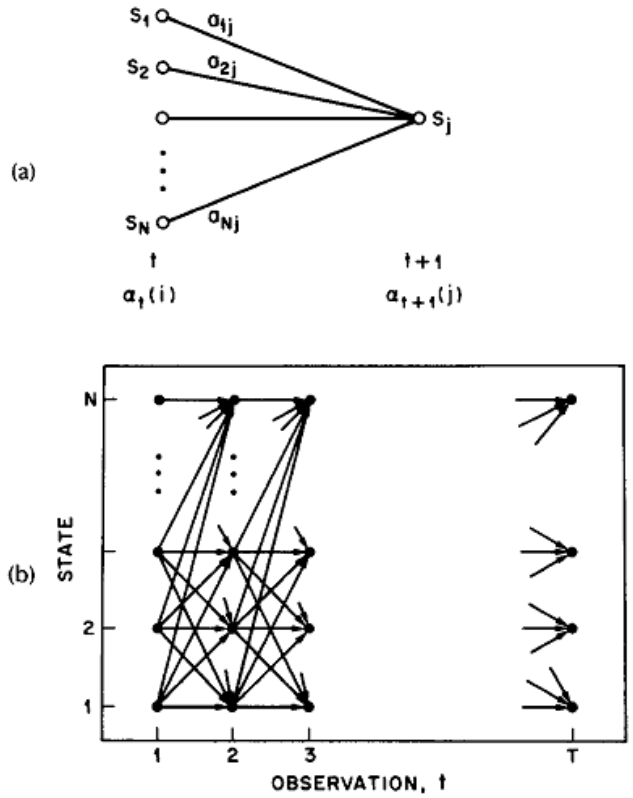


Figura 2.10: Esboço da criação da matriz $\boldsymbol{\alpha}$ pelo método *forward-backward* [55].

os valores de todas as possíveis combinações de estados e valores observados até então, de forma que:

$$\alpha_{tj} = \alpha_t(j) = P(O_1, \ldots, O_t, q_t = S_j|\lambda)\,. \tag{2.46}$$

Este método é caracterizado por formar uma "treliça" que acumula em cada um dos seus nós, os valores de todas as possíveis combinações até então. A figura 2.10 representa esta abstração.

Os valores de $\alpha_t(j)$ podem ser obtidos iterativamente, onde:

$$\alpha_1(j) = \pi_j b_j(O_1)\,, \tag{2.47}$$

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(j)a_{ij}\right] b_j(O_t)\,. \tag{2.48}$$

Deste modo, a partir de $\alpha_t j$ temos:

$$P(\boldsymbol{O}|\lambda) = \sum_{j=1}^{N} \alpha_T(j)\,. \tag{2.49}$$

2.3.1.2 Solução do problema 2

Para solução do problema 2, faz-se uso do algoritmo de Viterbi [55], este calcula de forma iterativa a sequência de estados $\boldsymbol{Q}$ que maximiza $P(\boldsymbol{Q}|\boldsymbol{O}, \lambda)$, que é equivalente ao

maximizar $P(\boldsymbol{Q}, \boldsymbol{O}|\lambda)$. Para isto é calculada outra matriz acumuladora $\boldsymbol{\delta}$, onde:

$$\delta_{jt} = \delta_t(j) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1\, q_2\, \ldots\, q_t = j, O_1\, O_2\, \ldots\, O_t|\lambda]\ , \tag{2.50}$$

$$\delta_{t+1}(j) = \max_i [\delta_t(i) a_{ij}] b_j(O_{t+1})\,. \tag{2.51}$$

Assim como no procedimento *forward-backward*, o cálculo dos valores de $\delta_t(j)$ são obtidos iterativamente. Para manter registrada a sequência cujos índices maximizam (2.51), outra matriz $\boldsymbol{\psi}$ é utilizada. Inicializando os valores $\delta_1(j)$ como:

$$\delta_1(j) = \pi_i b_i(O_1)\,, \text{ e} \tag{2.52}$$

$$\psi_{jt} = \psi_t(j) = 0\,, \text{ temos} \tag{2.53}$$

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t)\,, \tag{2.54}$$

$$\psi_t(j) = \arg\max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]\,. \tag{2.55}$$

Ao final do procedimento, precisamos encontrar a sequência de estados $\boldsymbol{Q}^*$ que melhor representa a sequência $\boldsymbol{O}$ percorrendo a matriz $\boldsymbol{\psi}$ de trás para frente. Desta forma:

$$q_T^* = \arg\max_{1 \leq j \leq N} [\delta_T(j)]\,, \tag{2.56}$$

$$q_{t-1}^* = \psi_{t+1}(q_{t+1}^*)\,. \tag{2.57}$$

2.3.1.3 Solução do problema 3

O problema 3 pode ser resolvido a partir de técnicas de otimização local. Uma das mais utilizadas é o algoritmo de Baum-Welch que obtém a estimação dos parâmetros de λ de forma iterativa. Para realizar esta tarefa, o algoritmo se vale dos valores já calculados para $\boldsymbol{\alpha}$ e do valor passo *backward* conhecido como $\boldsymbol{\beta}$. Podemos definir os elementos deste vetor como:

$$\beta_{it} = \beta_t(i)\,, \tag{2.58}$$

$$\beta_T(i) = 1\,, \tag{2.59}$$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(O_{t+1})\,. \tag{2.60}$$

Com estes valores em mãos, é possível encontrar o valor $\xi_t(i,j)$ que representa a probabilidade de encontrar o estado S_i no instante t e o estado S_j no instante $t+1$. Defini-se $\xi_t(i,j)$ como:

$$\xi_t(i,j) = P[q_t = S_i, q_{t+1} = S_j|O_t, \lambda]\,, \tag{2.61}$$

$$\xi_t(i,j) = \frac{P[q_t = S_i, q_{t+1} = S_j, \boldsymbol{O}|\lambda]}{P(\boldsymbol{O}|\lambda)}\,, \tag{2.62}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(\boldsymbol{O}|\lambda)}\,, \tag{2.63}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}\,. \tag{2.64}$$

Com os valores de $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$ é possível encontrar os valores de $\boldsymbol{\gamma}$, que representam a probabilidade de encontrarmos o estado S_j a partir de uma sequência de observação $\boldsymbol{O}$ e uma cadeia λ:

$$\gamma_{jt} = \gamma_t(j)\,, \tag{2.65}$$

$$\gamma_t(j) = P[q_t = S_j|\boldsymbol{O},\lambda]\,, \tag{2.66}$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(\boldsymbol{O}|\lambda)}\,, \tag{2.67}$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}\,. \tag{2.68}$$

As equações (2.64) e (2.68) ainda podem ser combinadas e obtemos:

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j)\,. \tag{2.69}$$

Nota-se que $\sum_{t=1}^{T-1}\gamma_t(i)$ corresponde o número esperado de transições a partir de S_i, enquanto que $\sum_{t=1}^{T-1}\xi_t(i,j)$ é o número esperado de transições entre S_i e S_j durante a sequência observada. A partir destes valores, é possível encontrar parâmetros de um novo modelo $\bar{\lambda}(\bar{\boldsymbol{A}},\bar{\boldsymbol{B}},\bar{\boldsymbol{\pi}})$. Temos finalmente que:

$$\bar{\pi}_i = \gamma_1(i)\,, \tag{2.70}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}\,, \tag{2.71}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^{T}\gamma_t(j)c(k,t)}{\sum_{t=1}^{T}\gamma_t(j)}\,\text{, onde} \tag{2.72}$$

$$c(k,t) = \begin{cases} 1 & O_t = v_k\,, \\ 0 & \text{caso contrário.} \end{cases} \tag{2.73}$$

2.3.1.4 Uso de HMM para reconhecimento de movimento

Com esta arquitetura bastante flexível, diversas pesquisas utilizam por completo os modelos obtidos a partir de HMM, ou utilizam parte de suas funcionalidades para modelar movimentos. Para reconhecimento de gestos feitos com a mão, formando tanto dígitos como palavras da linguagem de sinais americana, em [60] primeiramente são encontrados agrupamentos de pontos dentro do espaço das variáveis do vetor característica.

Modelando o movimento como uma cadeia de Markov sequencial, com distribuição uniforme na matriz de transições, o algoritmo de Baum-Welch [55] irá encontrar os valores do

vetor média $\boldsymbol{\mu_k}$ e da matriz de covariância $\boldsymbol{\Sigma_k}$ que definem a população de pontos de cada um dos k *movemas*. A partir do quadrado da distância de Mahalanobis [61],

$$d(\boldsymbol{x}, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) = (\boldsymbol{x} - \boldsymbol{\mu_k})^t \boldsymbol{\Sigma_k}^{-1} (\boldsymbol{x} - \boldsymbol{\mu_k}) \,, \tag{2.74}$$

é estabelecido um limiar para determinar a probabilidade do vetor de características atual $\boldsymbol{x}$ pertencer a determinado *movema*. A classificação final do movimento é obtida a partir de um método de busca de melhor solução por *Dynamic Programming* (DP).

Para reconhecimento dos movimentos de "sim"e "não"executados com a cabeça, duas cadeias de Markov são treinadas em [59], onde cada uma delas é treinada a partir das posições da estimativa da posição da cabeça x e y, respectivamente. Outra abordagem seria a utilização destas variáveis num mesmo vetor de características, como em [58] onde cinco *clusters* de de um vetor característica de dimensão contendo informações sobre posição (x, y) e angulação da cabeça globais e relativas entre amostras, são obtidos e utilizados para treinar duas cadeias distintas.

Com o uso de descritores de Fourier e estimativas da velocidade de rastreamento da mão por *optical flow* como vetor característica do movimento, [57] constrói uma cadeia de Markov a partir de valores quantizados deste vetor, num total de 64 possíveis valores observáveis, para interpretação do movimento das mãos.

Já em [56] a análise da marcha humana é decomposta na tarefa de segmentação de *blobs*, regiões de interesse com formato não definido, rastreamento destas regiões, modelamento estocástico da dinâmica destas regiões e o uso desta informação como a matriz de observação $\boldsymbol{B}$ para os estados de uma cadeia de Markov linear. Esta abstração permite que os *movemas* estejam representados pelos modelos dinâmicos, e a transição entre eles seja permitida pelos coeficientes da matriz $\boldsymbol{A}$ da HMM.

No estudo de movimentos indiretos do corpo humano, quando na análise das posições do volante e acelerador de um carro simulado, [62] utiliza este vetor de características para prever o comportamento do motorista. A partir da representação de modelos dinâmicos para diferentes estágios de um movimentos completo como a ultrapassagem, é utilizada uma medida de máximo verossimilhança do vetor atual com os modelos treinados com os valores observados da HMM. Com o treino de múltiplas cadeias, e utilizando o algoritmo de Viterbi, é possível prever o comportamento do motorista, e logo do veículo, antes que este seja completado.

Nos trabalhos de Dana Kulic [63] [64], é construído um classificador em cascata do movimento de perna utilizado numa sessão de reabilitação. Este movimento é primeiramente segmentado por um classificador inicial que utiliza a informação referente aos pontos de *zero velocity crosses*, ZVC [65], de um par de variáveis que representam o movimento da perna direita de uma pessoa. Obtendo então um segmento válido com o primeiro classificador, este é então classificado por uma HMM que representa os diferentes estágios que compõe o movimento treinado.

Para treinamento da cadeia, os valores dos vetores característica obtidos durante os movimentos de treino são agrupados em seis *clusters* pelo processo de *k-means* [66] [67]. Uma estimativa das funções *pdf* de cada um dos *clusters* é obtida com a suas populações de pontos. Estas funções são estimativas iniciais para observação do vetor característica em cada estado, e são ajustadas com o uso de uma variação do algoritmo de Baum-Welch, bem como todos os parâmetros da HMM. O cálculo final do valor de $P(\boldsymbol{O}|\lambda)$ e uso de um limiar também estipulado pelo treinamento, classifica os movimentos como válidos ou não.

3 METODOLOGIA

Programming:
when the ideas turn into the real things.
Maciej Kaczmarek

Para construir o sistema que possa reconhecer a face de um usuário posicionado a frente do computador, rastrear e interpretar o seu movimento de cabeça, foram utilizados alguns dos métodos mencionados no capítulo 2. Detalhes sobre a implementação destes métodos, bem como estes compõe a solução completa, serão encontrados neste capítulo.

Após as etapas de treinamento do classificador de movimento, o sistema final apresenta quatro estágios bastante distintos: reconhecimento de face, rastreamento de face, segmentação de movimentos candidatos e classificação de movimentos. A figura 3.1 indica como estes estágios estão associados com o objetivo de, ao ser obtida uma classificação válida para um movimento, seja enviado um comando para o sistema operacional (SO), de modo que uma prancha dinâmica possa ser acionada em paralelo.

O conteúdo deste capítulo está organizado de acordo com a sequência dos módulos presentes na figura 3.1. Na seção 3.1 é descrito o método de reconhecimento de face, em 3.2 explicamos o método de rastreamento de movimento utilizado, em 3.3 temos o método de segmentação de movimento e em 3.4 os métodos de classificação do movimento.

3.1 RECONHECIMENTO DE FACE E DEFINIÇÃO DE REGIÃO DE INTERESSE

O reconhecimento de face é o primeiro estágio para inicialização do sistema. Sua implementação é obtida com o uso de um classificador de Viola-Jones [36] já treinado para reconhecimento de rostos verticais. Esta implementação é disponível na biblioteca OpenCV 2.4, e o arquivo de configuração utilizado, *haarcascade_frontalface_alt.xml*, já apresenta todos os parâmetros resultantes do treino com Adaboost. Mais informações sobre a implementação deste classificador podem ser encontradas no Apêndice A e em [40].

Dos candidatos ao rosto do usuário, apenas o maior retângulo encontrado é analisado. Partimos do princípio que o usuário está posicionado a frente da *webcam* e este é a pessoa mais próxima da câmera, logo o tamanho do seu rosto na imagem é maior que os demais. Esta abordagem permite que múltiplas pessoas estejam presentes na imagem, sem comprometer o funcionamento.

Se após o comando de início do sistema, nenhum rosto for encontrado, ou o resultado do classificador precise ser desconsiderado, o sistema deverá ser reiniciado. Uma possível

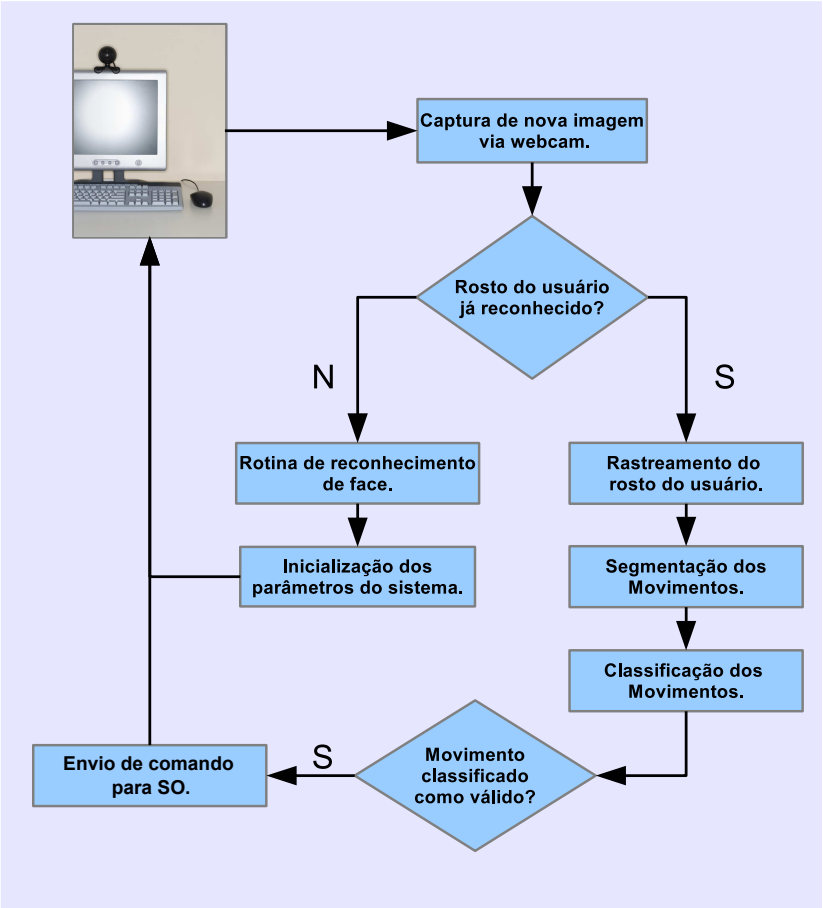


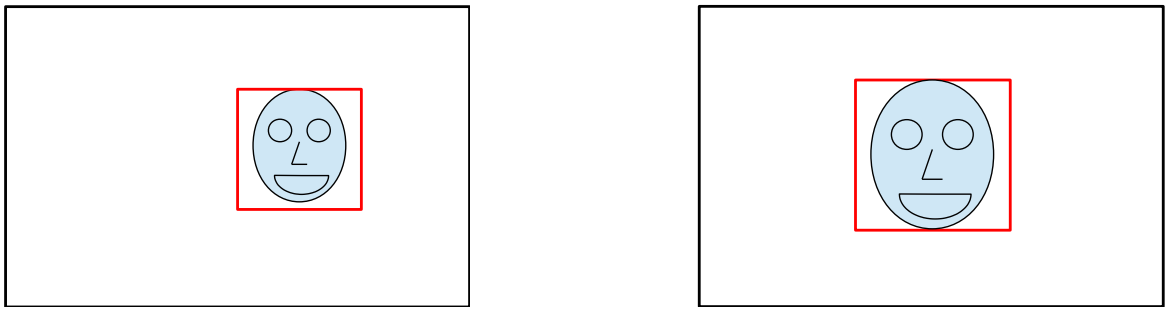
Figura 3.1: Fluxograma representativo da sequência de estágios que compõe o sistema final já treinado.

causa para a falha da etapa de reconhecimento é a inclinação da face desejada, visto que o classificador utilizado foi treinado apenas com exemplares de rostos na posição vertical com pouca inclinação. Caso esta última seja a razão da falha, o rosto do usuário pode ainda ser determinado manualmente com o uso do *mouse*.

Ao fim desta etapa, serão obtidos os seguintes parâmetros: tamanho do quadrado que delimita a face do usuário, $s_{h0} = (w_{h0}, h_{h0})$; e posição do retângulo que delimita a face do usuário, $p_{h0} = (x_{h0}, y_{h0})$. Esta informação inicialmente definirá o tratamento da nova imagem de trabalho do sistema.

Para cada posição diferente do usuário na imagem, seja no plano (x, y) ou em relação a distância em relação a câmera, tamanhos diferentes de s_{h0} bem como posições diferentes p_{h0} serão obtidos. Contudo essa variabilidade é indesejável para padronização dos módulos de rastreamento e segmentação, bem como dificulta a comparação entre diferentes amostras de movimento.

Mantendo o tamanho total da imagem de trabalho, $w_o \times h_o$, mas possibilitando padronizar o tamanho e posição inicial das faces rastreadas, duas operações são feitas: escala da imagem por completo, definição da nova região de interesse, ROI.



(a) (b)

Figura 3.2: Esboço da operação de reconhecimento de face (a), e escala e determinação da ROI (b).

O fator de escala da imagem, k_e, é determinado de modo que o quadrado que indica a posição da face ocupe um terço da largura da região de interesse final de $w_o \times h_o$. Deste modo:

$$k_e = \frac{w_o}{3 \times w_{h0}} \tag{3.1}$$

Uma operação de escala com interpolação linear é feita em toda a imagem, obtendo assim uma nova posição e um novo tamanho para o quadrado que contém a estimativa da posição da face, $\dot{p_{h0}} = (x_{h0} \times k_e, y_{h0} \times k_e)$ e $\dot{s_{h0}} = (w_{h0} \times k_e, h_{h0} \times k_e)$. Como não é interessante apenas aumentar a escala da imagem, a região de interesse deve ser delimitada de forma a centralizar a estimativa da posição de face. Fixando o tamanho da área de interesse, $s_{ROI} = (w_o, h_o)$, temos:

$$p_{ROI} = \dot{p_{h0}} - \left(\frac{w_o}{3}, \frac{h_o}{3}\right) \therefore \tag{3.2}$$

$$p_{h0} = \left(\frac{w_o}{3}, \frac{h_o}{3}\right) \tag{3.3}$$

A operação de escala e posicionamento da ROI será feita para todos os *frames* subsequentes, independentes das novas posições de cabeça do usuário. Todas as novas estimativas de posição da face serão obtidas com referência a p_{ROI}. A figura 3.2 esboça a operação de escala e determinação da nova região de interesse.

3.2 RASTREAMENTO DE FACE

O rastreamento da face é realizado utilizando-se a técnica de *Mean-shift* e a medida de similaridade PPM [47]. Desta maneira, após o reconhecimento de face, é necessário registrar o histograma do alvo para rastreamento, bem como definir a região de procura a ser utilizada no próximo *frame* adquirido. A posição p_{h0} e o tamanho s_{h0} são atualizados de forma a excluir parte dos *pixels* que compõem o cabelo do usuário, de forma que o centro do

quadrado é mantido, contudo seu tamanho é reduzido em 20%. Só então é gravada a imagem alvo a ser rastreada.

Este procedimento facilita a tarefa de rastreamento porque cria uma região de delimitação entre o rosto e o *background* bem definida, já que espera-se que a cor do cabelo seja consideravelmente diferente da pele do usuário.

A partir daí, a imagem alvo obtida, originalmente codificada com o padrão de cores *red green blue* (RGB) é convertida para uso do padrão de cores *hue saturation value* (HSV). Esta conversão visa salientar os *pixels* que compõe a pele, rosto e os olhos, já que estes possuem componentes predominantes no canal H, ajudando a diferenciar de possíveis *pixels* presentes no fundo da imagem [48]. Contudo, entendemos que para aumentar a robustez do sistema, todos dos canais HSV devem ser utilizados, ajudando a diferenciar os usuários de quaisquer objetos que estejam presentes na imagem.

Para isso, foi implementado um histograma $\boldsymbol{q}$ dos valores HSV com combinações três-a-três de suas camadas. Cada camada foi dividida em 16 grupos, de forma que cada valor de pixel $v_p = (h, s, v)$ foi quantificado em um dos 4046 grupos possíveis. Este é o número de intervalo de valores contidos no histograma.

Para definição da área de procura da imagem, é estipulado que esta tem um tamanho $s_s = 1,5 \times s_{h0}$ e está centrada na região estimada da face. O cálculo do histograma $\boldsymbol{s}$ é feito da mesma maneira que o feito para o alvo.

A partir dos próximos *frames* obtidos, o algoritmo de *Mean-Shift*, conforme indicado na equação (2.27), irá gerar uma estimativa *a priori* da próxima posição p_{hk} do rosto do usuário. Esta medida apresenta ruídos inerentes à mudanças na iluminação e limitações da *webcam* que inviabilizam o uso pra reconhecimento do movimento. Desta maneira um filtro de Kalman é utilizado para filtrar a evolução da estimativa da posição do rosto [68].

O modelo de velocidade constante representa bem o movimento desejado, já que as acelerações esperadas do movimento de cabeça são pequenas e não é desejável fixar um modelo de movimento devido a esperada variabilidade entre os possíveis usuários. Definindo o vetor de estado do rosto no instante k como $\boldsymbol{\theta}_k = [h_{vxk}, h_{vyk}, h_{xk}, h_{yk}]^T$, composto pela velocidade e posição da estimativa do rosto, e a medição do sistema no instante k como $\boldsymbol{z}_k = (h_x, h_y)$, temos:

$$\boldsymbol{\theta}_k = \boldsymbol{\Phi}\boldsymbol{\theta}_{k-1} + \boldsymbol{\omega}_{k-1}\,, \tag{3.4}$$

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{\theta}_k + \boldsymbol{v}_k\,, \tag{3.5}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \tag{3.6}$$

$$\boldsymbol{H} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.7}$$

O erro relativo ao modelo empregado, $\boldsymbol{\omega}_k$, e o erro relativo a medição, $\boldsymbol{v}_k$, são ruídos brancos com distribuição normal com média zero e matrizes de covariância $\boldsymbol{Q}$ e $\boldsymbol{R}$ respectivamente:

$$\boldsymbol{\omega}_k = N(0, \boldsymbol{Q}), \tag{3.8}$$

$$\boldsymbol{Q} = \begin{bmatrix} 0,1 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0 \\ 0 & 0 & 0,1 & 0 \\ 0 & 0 & 0 & 0,1 \end{bmatrix}, \tag{3.9}$$

$$\boldsymbol{v}_k = N(0, \boldsymbol{R}), \tag{3.10}$$

$$\boldsymbol{R} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}. \tag{3.11}$$

Os valores das matrizes de covariância foram atribuídos empiricamente e tem a intenção de reduzir a incerteza sobre o modelo escolhido e atribuir uma maior dúvida sobre os resultados obtidos com o algoritmo de *Mean-Shift* empregado. O resultado obtido foi uma estimativa da posição da face com trajetória suave, facilitando a tarefa de segmentação do movimento.

3.3 SEGMENTAÇÃO DE MOVIMENTOS

Com o objetivo de diminuir o número de testes em trechos de movimento pelo classificador, utilizamos um método de segmentação adaptado do implementado em [64] e [63]. Com isso é esperado desconsiderar trechos em que não há movimento, e avaliar apenas uma vez um movimento completo, delimitando-o precisamente.

A partir das estimativas de velocidade do rosto obtidas com o filtro de Kalman e presentes no vetor $\boldsymbol{\theta}_k$, segmentamos o movimento. Cada segmento é delimitado a partir dos valores estimados de velocidade, de forma que medidas sucessivas de $\boldsymbol{\theta}_k$ que apresentam velocidade reduzida são descartadas.

A motivação deste tipo de segmentação foi extraída do conceito de ZVC [65], de forma que os momentos em que a variável $\varphi_k = h_{vxk}^2 + h_{vyk}^2$ ultrapassa um limiar mínimo, apresenta um pico e um vale, delimitam o segmento como ilustrado na figura 3.3.

Segmentos são então definidos como conjuntos de dados $\boldsymbol{\theta}_k$ que apresentam um aumento de velocidade em qualquer um dos eixos x e y, e depois apresentam um declínio dessa velocidade até o repouso ou até uma nova aceleração. Esta abordagem difere um pouco da utilizada em [63] e [64], já que nestes trabalhos são analisados movimentos repetitivos de

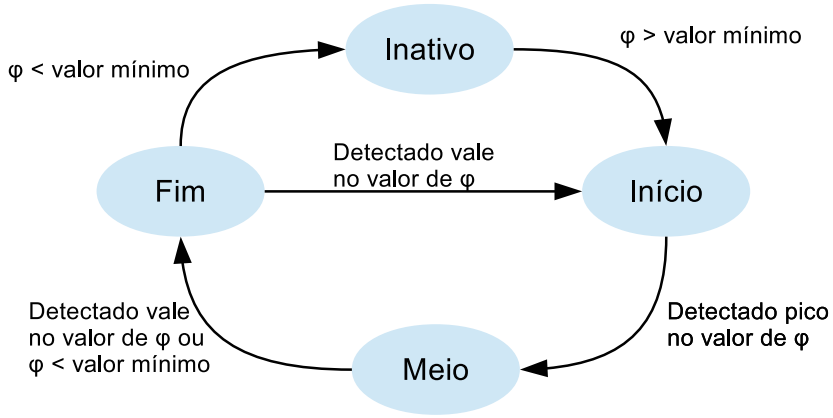


Figura 3.3: Máquina de estados utilizada na rotina de segmentação com medida ZVC.

intervalo longo e com características de retorno a posição inicial. Nossa abordagem é mais abrangente, possibilitando segmentar grupos menores de dados que representariam qualquer mudança de posição do rosto do usuário.

A figura 3.4 exemplifica como um trecho de dados representando um movimento contínuo de cabeça pode ser segmentado. Esta rotina de segmentação desprezará conjuntos de dados em que não foi observada uma modificação significativa da posição do rosto, diminuindo a demanda computacional.

3.4 CLASSIFICAÇÃO DO MOVIMENTO

Dois métodos diferentes foram utilizados para avaliar um movimento como válido ou inválido. Um limiar de posição calculado para o eixo x, e cadeias de Markov. O limiar de posição é uma maneira de representar a maneira como um botão interpreta o movimento do usuário, e as cadeias de Markov permitem um modelamento mais complexo e preciso do movimento funcional, esperando-se diferenciá-lo melhor. A seguir descrevemos os métodos de classificação.

3.4.1 Uso do limiar de posição

Uma maneira de interpretar o movimento funcional de cabeça de um indivíduo seria a utilização de limiares de posição. O uso de um limiar de posição no eixo x pode simular o uso de um botão posicionado ao lado do rosto, enquanto que um limiar no eixo y poderia simular o uso de um botão abaixo do queixo.

Desta maneira foi implementado um classificador "ingênuo" utilizando o limiar de posição L_x no eixo x. Neste método é desnecessário a etapa de segmentação do movimento, já que para qualquer momento que a posição do rosto ultrapasse L_x a resposta do classificador é verdadeira.

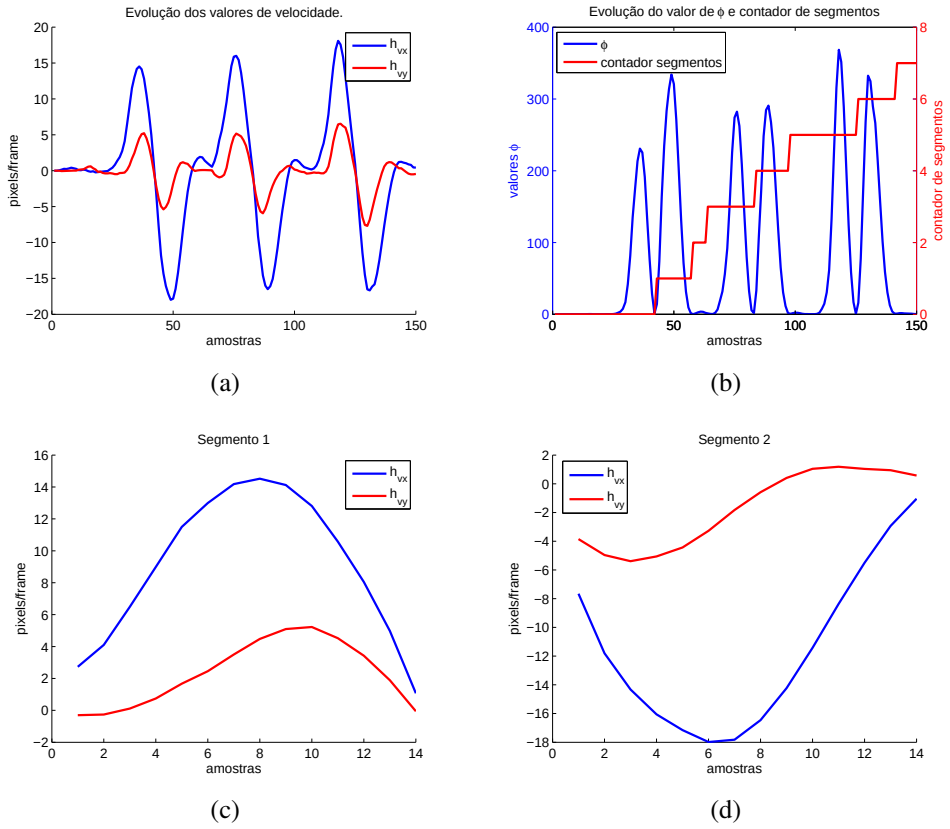


Figura 3.4: Exemplo do algoritmo de segmentação de um trecho de movimento (a) por meio da medida ϕ (b), obtendo o primeiro (c) e o segundo segmento (d).

O valor de L_x pode ser positivo ou negativo, e é calculado a partir da posição inicial obtida com o reconhecimento de face (h_{x0}, h_{y0}). O cálculo do valor máximo de $\delta_k = |(h_{xk}, h_{yk}) - (h_{x0}, h_{y0})|$ irá determinar o valor de L_x. Isto será feito até que um comando de fim de calibração seja enviado ao sistema.

O ponto (l_x, l_y) que obtém o valor máximo de δ_r será utilizado para estabelecer $L_x = l_x$. Para facilitar a identificação do movimento, de forma que o usuário não precise ultrapassar a amplitude máxima já atingida, definimos um limiar auxiliar L'_x que está 20 *pixels* mais próximo à referência r_m. Deste modo o movimento será válido se:

$$(h_{xk}, h_{yk}) \notin]h_{x0},\ h_{x0} + L'_x[\text{, quando } L'_x > 0 \tag{3.12}$$

$$(h_{xk}, h_{yk}) \notin]h_{x0} + L'_x,\ h_{x0}[\text{, quando } L'_x < 0 \tag{3.13}$$

3.4.2 Uso da cadeia de Markov

Partindo da hipótese que o movimento funcional de cabeça, este já detectável por um botão mecânico localizado ao lado do rosto, possa ser subdividido em etapas, a aplicação de cadeias de Markov para o modelamento do movimento funcional é plausível. A partir de uma cadeia treinada com movimentos considerados válidos, é possível extrair estados que

dividem o movimento, e modelar as possíveis transições entre eles.

Para nossa implementação de cadeias de Markov utilizaram-se três diferentes vetores de características.

$$\boldsymbol{\phi}_k^i = \begin{bmatrix} h_{vxk} \\ h_{vyk} \\ h_{xk} \\ h_{yk} \end{bmatrix}, \tag{3.14}$$

$$\boldsymbol{\phi}_k^{ii} = \begin{bmatrix} h_{xk} \\ h_{yk} \end{bmatrix}, \tag{3.15}$$

$$\boldsymbol{\phi}_k^{iii} = \begin{bmatrix} h_{vxk} \\ h_{vyk} \end{bmatrix}. \tag{3.16}$$

Isto é feito com objetivo de avaliar se há diferença na performance das cadeias com variáveis diferentes, e como estas conseguem classificar movimentos não treinados.

Serão destacados aqui os métodos utilizados para treinar as HMM e como utilizá-las para classificação.

3.4.2.1 Treinamento da HMM

Para efetuar o treinamento da cadeia, é necessário que uma rotulação dos segmentos válidos obtidos numa sessão de treino do sistema. Nesta o usuário deve executar algumas vezes o movimento escolhido como funcional.

Para qualquer um dos vetores de características escolhidos, existem múltiplas combinações possíveis de valores, principalmente se utilizados os valores de velocidade que são contínuos. A fim de criar uma estimativa mais precisa da probabilidade de observação de um valor $\boldsymbol{\phi}_k$ para um estado S_i, e diminuir a dimensão da matriz $\boldsymbol{B}$, nenhuma rotina de quantização do vetor característica foi realizada. A matriz $\boldsymbol{B}$ é então redefinida para comportar o valor calculado de *pdfs* que representam a distribuição dos valores de $\boldsymbol{\phi}_k$ para cada estado da HMM.

Deste modo, para cada estado S_i está definida uma função normal $N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ que define o valor de $b_i(\boldsymbol{O})$. $\boldsymbol{B}$ pode então ser definida como:

$$\boldsymbol{B} = \begin{bmatrix} b_1(\boldsymbol{O}) \\ \vdots \\ b_N(\boldsymbol{O}) \end{bmatrix}, \text{ onde } b_i(\boldsymbol{O}) = N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{3.17}$$

Com esta modificação da matriz $\boldsymbol{B}$, novos valores devem ser adicionados a lista de parâmetros. O algoritmo de Baum-Welch tem sua equação (2.72) substituída pelas equações

abaixo:

$$\boldsymbol{\mu}_i = \frac{\sum_k^T \gamma_k(i)\boldsymbol{O}_k}{\sum_k^T \gamma_k(i)} \tag{3.18}$$

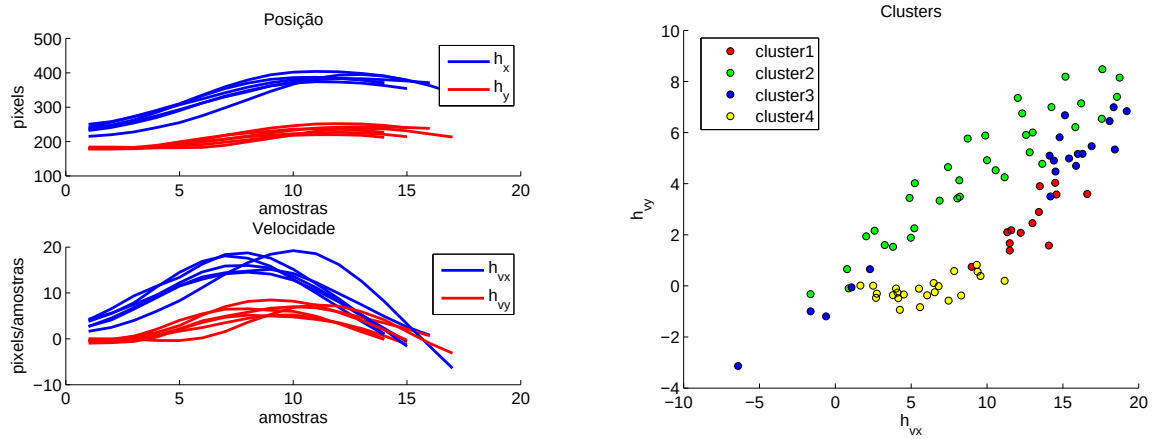
$$\boldsymbol{\Sigma}_i = \frac{\sum_k^T \gamma_k(i)(\boldsymbol{O}_k - \boldsymbol{\mu}_i)(\boldsymbol{O}_k - \boldsymbol{\mu}_i)'}{\sum_k^T \gamma_k(i)} \tag{3.19}$$

Contudo, assim como feito em [63] e [64], são esperados melhores resultados do modelamento da cadeia HMM se as estimativas iniciais de $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N$ e $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_N$ já modelarem algum comportamento observado dos dados válidos, comparado a simples inicialização randômica.

Uma boa estimativa inicial seria utilizar o algoritmo de *k-means* para encontrar um número de *clusters* N que separem os dados de treinamento, e a partir destes encontrar as estimativas iniciais de todos $\boldsymbol{\mu}_i$ e $\boldsymbol{\Sigma}_i$.

A partir de k estimativas iniciais para os *clusters*, são estipuladas aleatoriamente k estimativas $\boldsymbol{\mu}_i$. Para cada ponto classificado, é atribuído o *cluster* i cuja média está mais próxima. Em seguida é recalculada a média $\boldsymbol{\mu}_i$ a partir dos elementos já classificados no *cluster* i. Este processo iterativo é feito até atingir critérios de convergência, como o número de iterações ou diferenças nos incrementos dos valores $\boldsymbol{\mu}_i$ [66] [67].

Na figura 3.5 temos um exemplo da utilização de seis segmentos para criar quatro *clusters* com o vetor de características $\boldsymbol{\phi}_k^i$. Os *clusters* são identificados por cores diferentes.



(a) (b)

Figura 3.5: Superposição de seis segmentos (a), para construção de quatro *clusters* (b) utilizando o método de *k-means*.

Com esta estimativa inicial para os valores de $\boldsymbol{\mu}_i$ e $\boldsymbol{\Sigma}_i$, para todos os i estados da cadeia de Markov, inicializamos com valores randômicos o vetor $\boldsymbol{\pi}$ e a matriz $\boldsymbol{A}$ respeitando as propriedades descritas em (2.32) e (2.35). O algoritmo de Baum-Welch é utilizado para ajustar estes valores de $\boldsymbol{\pi}$ e $\boldsymbol{A}$ aos dados utilizados na criação dos *clusters*, construindo a cadeia que modela o movimento desejado.

3.4.2.2 Avaliação com a HMM

Para avaliar se o modelo treinado representa corretamente um movimento segmentado, utilizamos do cálculo da verossimilhança $L = P(\boldsymbol{O}|\lambda)$ e o número de amostras contidas no segmento. Como estes valores de L são muito baixos, é comum utilizar a medida $LL = log(P(\boldsymbol{O}|\lambda))$

O número de amostras n_a é importante para diferenciar segmentos que obtém um alto valor de LL apesar de possuírem um comprimento muito curto. Como os valores de a_{ij} são menores que um, a medida que mais amostras são utilizadas no cálculo de $P(\boldsymbol{O}|\lambda)$, menor será a influência dos valores de $b_i(\boldsymbol{O}_k)$.

Criamos então uma variável $\boldsymbol{\psi}$ que compõe as duas medidas:

$$\boldsymbol{\psi} = \begin{bmatrix} LL \\ n_a \end{bmatrix} \tag{3.20}$$

Os segmentos utilizados no treino são avaliados pela cadeia e os valores de $\boldsymbol{\psi}$ são utilizados no cálculo de uma *pdf*, $T = N(\boldsymbol{\mu}_\psi, \boldsymbol{\sigma}_\psi)$, que representa a variação esperada dos valores de $\boldsymbol{\psi}$ dos segmentos considerados válidos.

O quadrado da distância de Mahalanobis calculada entre o valor $\boldsymbol{\psi}$ de um movimento candidato e a distribuição T será o indicador da diferença entre a observação e o modelo HMM. Esta medida de distância pode ser definida como [61] [67]:

$$\Delta^2 \quad = \quad (\boldsymbol{\psi} - \boldsymbol{\mu}_\psi)' \boldsymbol{\sigma}_\psi^{-1} (\boldsymbol{\psi} - \boldsymbol{\mu}_\psi) \tag{3.21}$$

Um teste de hipótese pode ser feito a partir da distância de Mahalanobis. Partindo da hipótese nula de que um movimento que possui valor $\boldsymbol{\psi}$ é igual ao valor esperado $\boldsymbol{\mu}_\psi$ da distribuição $T = N(\boldsymbol{\mu}_\psi, \boldsymbol{\sigma}_\psi)$, podemos encontrar o valor C no qual rejeitaremos a hipótese nula quando $\Delta^2 > C$ com um nível de confiança α, ou seja, podemos calcular $P(\Delta^2 > C|\boldsymbol{\psi} = \boldsymbol{\mu}_\psi) = \alpha$ [69].

Sendo o quadrado da distância de Mahalanobis uma variável aleatória com distribuição χ^2_2, podemos fazer este teste de hipótese a partir dos valores tabelados da tabela $\chi^2_2(1-\alpha)$, criando a regra:

$$\Delta^2 > \chi^2_2(1-\alpha) \quad \Rightarrow \quad \text{movimento inválido} \tag{3.22}$$

$$\Delta^2 \leq \chi^2_2(1-\alpha) \quad \Rightarrow \quad \text{movimento válido} \tag{3.23}$$

3.5 TESTES COM USUÁRIOS

Os testes com usuários foram feitos com pessoas que não apresentam movimentação involuntária, bem como pessoas que apresentam algum comprometimento motor. A tarefa

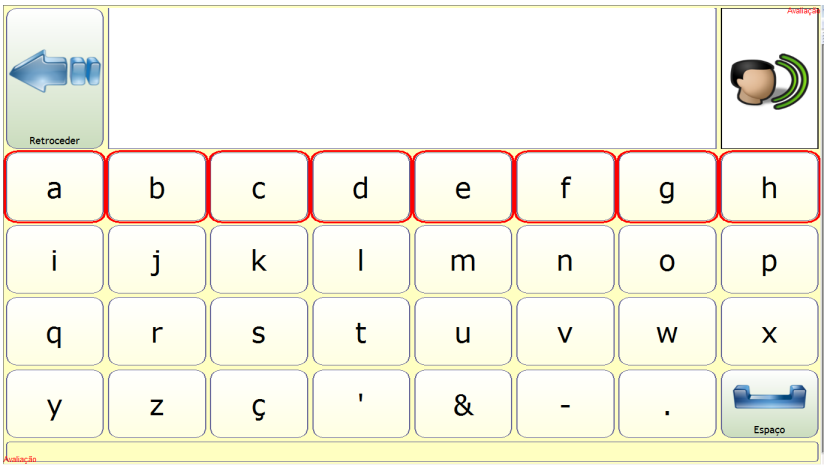


Figura 3.6: Prancha de comunicação utilizada nos testes com usuários desenvolvida no ambiente Tobii Communicator [70].

realizada por todos foi concebida para simular o uso de um botão mecânico para acionar a varredura de uma prancha de comunicação dinâmica.

Esta prancha foi configurada para impedir o uso de preditor de palavras, exigindo o acionamento de todas as teclas necessárias na escrita da sentença de teste. Para exigir intervalos variados entre as escolhas de linhas e colunas da prancha, as frases "boa tarde" e "boa tarde tchau" foram escolhidas. A primeira mais curta será utilizada por pessoas com deficiência, enquanto que a segunda por pessoas hígidas. Isto foi decidido de forma a diminuir o tempo de coleta, não exigindo demais dos participantes.

A primeira frase contém nove caracteres contando-se a tecla "espaço", necessitando portanto de no mínimo 18 acionamentos e a segunda necessita de 30. São quantidades suficientes para que sejam extraídos os movimentos responsáveis por compor o treino, bem como os utilizados na validação. A escolha destas sentenças exige que todas as quatro linhas da prancha precisam ser acionadas, bem como seis colunas. A figura 3.6 representa esta prancha.

Antes dos testes, todos usuários concordaram em participar da pesquisa lendo e assinando o termo de consentimento livre e esclarecido. Para usuários que não apresentam movimentação, foi estipulado uma posição próxima a face para posicionar o botão para acionamento da prancha. Foi pedido que além dos movimentos necessários para escrita, outros movimentos diferentes fossem efetuados. No caso dos usuários acometidos, estes escolheram a melhor posição para uso da interface e apenas focaram na escrita da sentença de teste.

A prancha foi configurada com um tempo de acionamento inicial de dois segundos, podendo ser alterado pelo usuário, e um intervalo após a seleção de um segundo. Todos os elementos da linha são varridos apenas uma vez a cada seleção de linha.

3.6 DETALHES DA IMPLEMENTAÇÃO

Nosso sistema foi desenvolvido com o uso da biblioteca de visão computacional OpenCV compilada para uso da linguagem de programação C++. Os SOs para os quais esta aplicação se destina são os sistemas Windows, principalmente os modelos Windows 7 ou posteriores. Esta escolha visa principalmente facilitar a disseminação deste sistema, e ainda permitir a integração com os principais programas de CAA utilizados em PC.

Contudo, nem todo o sistema foi implementado apenas em C++. Com o objetivo de acelerar o desenvolvimento, bem como auxiliar na análise de seus dados, a ferramenta MATLAB foi utilizada na fase de treino.

O sistema desenvolvido até aqui não é completamente acessível ao usuário com deficiência. Para isso, uma interface gráfica mais amigável teria que ser desenvolvida para que as configurações necessárias fossem modificadas sem o uso de teclado. Atualmente, para uso completo do software, é necessária a presença de um acompanhante.

Durante a sua inicialização devem ser passados ao sistema três parâmetros via linha de comando: modo de operação (execução ou calibração), identificador do usuário, e indicação da forma de entrada de dados de imagem (*webcam* ou arquivo de vídeo).

A escolha do modo de operação indica principalmente se será necessário gravar em vídeo as informações recebidas da *webcam* (modo calibração), e se serão gerados comandos para o SO (modo execução). A gravação de vídeo é feita para permitir a rotulação dos movimentos executados e a comparação entre diferentes métodos de rastreamento, segmentação e classificação do movimento.

3.6.1 Descrição do Hardware

Para simular a utilização do sistema por um usuário final, optamos por desenvolver todos seus componentes num *laptop* comum com câmera embutida. Esta configuração é muito comum e permite que a *webcam* esteja posicionada no nível dos olhos do usuário e sempre na mesma posição em relação a tela do computador.

As configurações da máquina são:

- Processador Intel Core i5;
- 4GB de memória RAM;
- Câmera USB 2.0 com resolução de 640×480 *pixels*;
- Sistema operacional Windows 7.

3.6.2 Aplicação em C++

Com o objetivo de produzir uma aplicação final que pudesse ser facilmente distribuída e utilizada por qualquer pessoa com interesse neste trabalho, foi desenvolvido um aplicativo para o sistema operacional Windows utilizando a linguagem C++. Este escolha foi definida pelo grande número de usuários deste sistema operacional, e a possibilidade de utilizar a biblioteca de visão computacional OpenCV [71] já compilada para C++.

Neste programa, o usuário realiza a coleta dos dados de treinamento e executa a classificação de movimento a partir das técnicas de limiar e HMM. O sistema também foi concebido para aceitar tanto o $stream$ de dados enviados por uma *webcam* conectada ao PC, quanto um arquivo de vídeo no formato AVI.

A execução do programa é feita por linha de comando e possui os seguintes parâmetros:

TrackerV03.exe [EXEC | CAL] *Usuário* <*caminho do arquivo de vídeo*>

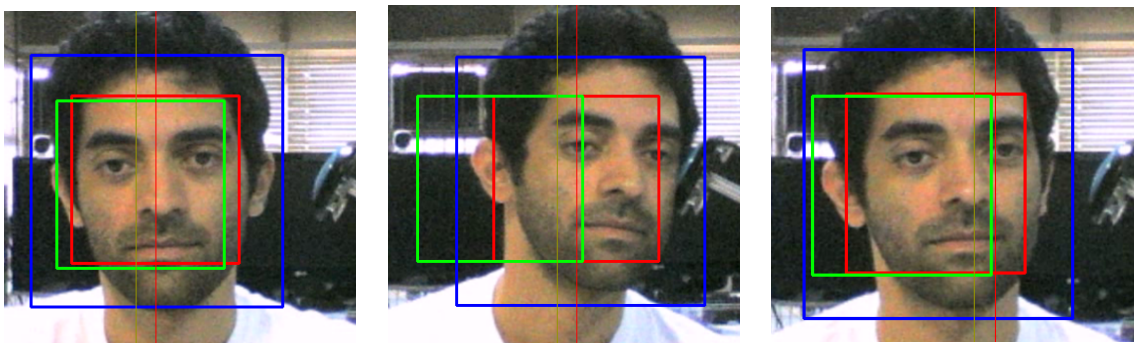
O primeiro campo é o nome do executável para o programa, o segundo refere-se ao modo de operação, execução (EXEC) ou calibração (CAL). Em seguida deve ser informado o usuário com um nome que não contenha espaços ou acentos. Ao final é opcional o uso do sistema com um arquivo de vídeo em AVI ou a imagem da *webcam* (campo vazio). A sequência lógica de uso do sistema será inicialmente coletar dados para treino com configuração CAL, e com o perfil de movimento do usuário treinado, é utilizada a configuração EXEC.

Ao final do programa quatro arquivos de texto são gerados:

- *Usuario*_fileDecode.txt
- *Usuario*_fileSegments.txt
- *Usuario*_fileZVC.txt
- *UsuarioSemana_Mês_Dia_Hora*#*Minuto*#*Segundo_Ano*.txt

Os campos em itálico do último arquivo indicam o horário da execução do programa, sendo os campos *Semana* e *Mês* os únicos escritos em inglês e por extenso. O arquivo *Usuario*_fileDecode.txt contém os valores LL obtidos para cada segmento classificado, o arquivo *Usuario*_fileSegments.txt apresenta todos os valores $\boldsymbol{\theta}_k$ de todos os movimentos segmentados durante a execução, o arquivo *Usuario*_fileZVC.txt registra os valores de ϕ obtidos com a rotina de segmentação descrita em 3.3. O último arquivo registra todos os dados referentes a trajetória do rosto do usuário, posição e velocidade, e a posição do limiar de acionamento vertical.

Para iniciar o rastreamento do rosto, e consequentemente a segmentação e classificação dos movimentos, é necessário encontrar o rosto como descrito na seção 3.1. No modo EXEC



(a) (b) (c)

Figura 3.7: Exemplo de determinação do limiar L_x para um movimento de cabeça. Em (a) temos o usuário em repouso com posição inicial determinada pelo quadrado verde e a posição atual (h_{xk}, h_{yk}) determinada pelo quadrado vermelho. Em (b) temos o movimento funcional desejado, alterando o valor L_x, indicado pela linha vermelha, e o valor L'_x indicado pela linha amarela. Em (c) já temos o valor do limiar definido ao enviar comando 'c' via teclado.

ou CAL é necessário enviar o comando de teclado 's' para iniciar o reconhecimento de rosto quando utilizando a *webcam* como entrada de dados.

Se for desejado utilizar um arquivo como entrada, isto é feito automaticamente para o primeiro *frame*. A partir daí é possível fixar o valor do limiar de posição a partir do comando de teclado 'c'. A figura 3.7 ilustra como pode ser definido os limiares L_x e L'_x.

Apenas no modo EXEC é feita a classificação do movimento a partir de uma cadeia HMM e necessita que esta cadeia já tenha sido treinada e seus parâmetros registrados num arquivo de texto de nome *Usuario*.txt. Para o cálculo de $LL = log(P(\boldsymbol{O}|\lambda))$ foi utilizada a biblioteca CvHMM [72].

Contudo esta biblioteca foi desenvolvida para trabalhar apenas com cadeias com observações de eventos discretos. Foi necessário implementar os componentes que faltam para substituir o uso de probabilidades discretas de observação para probabilidades contínuas, *pdfs*.

Ao executar o aplicativo no modo EXEC, todos os arquivos .txt gerados terão o sufixo "_EXEC"adicionado aos seus nomes. Desta forma não há perda dos arquivos utilizados para treino no advento do usuário utilizar o sistema no modo EXEC.

Para finalizar a execução do programa, basta a qualquer momento enviar o comando de teclado 'q', e no caso do uso de arquivos de vídeo, assim que todos os *frames* forem analisados o programa encerrará. Caso a execução seja feita com leitura da *webcam*, um arquivo de vídeo da sequencia de movimentos é feito. Este arquivo é utilizado posteriormente na rotulação dos movimentos registrados no arquivo *Usuario*_fileSegments.txt.

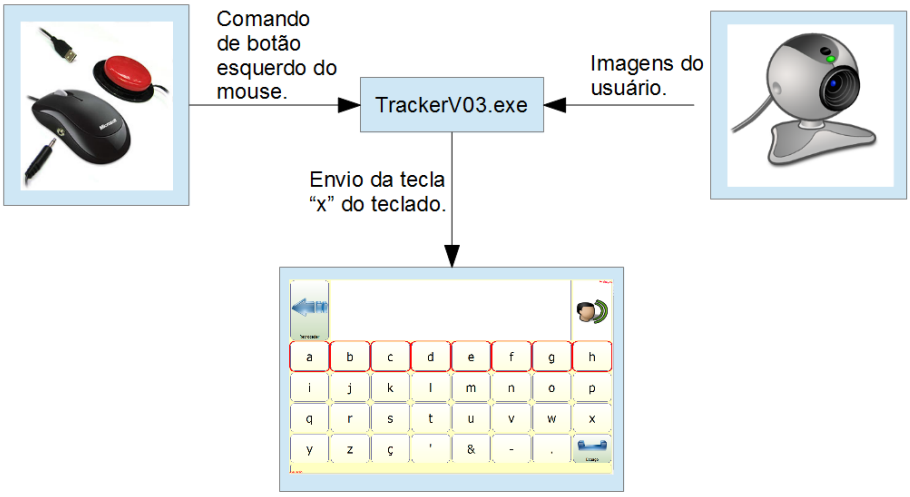


Figura 3.8: Diagrama da interação entre aplicativos e acionamento do botão próximo à cabeça.

3.6.3 *Scripts* em MATLAB

Para facilitar o treinamento da cadeia HMM, bem como analisar os dados de forma mais rápida, foram desenvolvidos *scripts* em MATLAB com o uso da biblioteca para cadeias de Markov chamada *HMM Toolbox* desenvolvida por Kevin Murphy [73].

O *script* de treinamento utiliza-se do arquivo *Usuario*_fileSegments.txt e de uma lista de rótulos para os segmentos contidos nesta amostra que são considerados válidos e devem ser utilizados no treinamento. Todos os passos referentes à seção 3.4.2.1 são feitos e a cadeia resultante do treino é salva no arquivo *Usuario*.txt. Os valores referentes a distribuição T também são registrados para uso posterior na aplicação C++.

Todos os gráficos gerados nesta dissertação foram obtidos através de *scripts* em MATLAB que utilizam os dados gerados pela aplicação C++.

3.6.4 Integração com aplicativo CAA

A etapa de rotução do movimento segmentado como válido e inválido é trabalhosa quando muitos segmentos são registrados, especialmente nos casos de movimentos involuntários. Para facilitar esta tarefa, a informação de acionamento do botão esquerdo do mouse, gerado a partir do toque no botão para uso da prancha, é fornecida primeiramente ao sistema desenvolvido. Este por sua vez irá registrar cada acionamento ao segmento correspondente. Só então é enviado o comando de tecla de teclado "x"para a prancha feita no ambiente Tobii Communicator. Esta integração pode ser vista no diagrama da figura 3.8.

4 RESULTADOS E DISCUSSÃO

Software and cathedrals are much the same,
first we build them, then we pray.
Sam Redwine

Neste capítulo estão detalhados os resultados obtidos com o sistema descrito no capítulo 3. A coleta de dados feita com voluntários simulou a utilização de um botão para escrita de uma frase de teste a partir uma prancha de comunicação. Dois voluntários portadores de paralisia cerebral também realizaram a atividade de maneira adaptada para suas limitações.

Os resultados estarão divididos de acordo com os módulos implementados: reconhecimento de faces, rastreamento de movimentos, segmentação de movimentos, calibração do limiar de posição, treinamento da cadeia de Markov, classificação da HMM, classificação com uso da distância de Mahalanobis.

Os resultados numéricos e gráficos aqui apresentados correspondem a uma coleta feita com um dos autores, os demais resultados serão comentados neste capítulo, mas seus gráficos e matrizes estarão registrados no Apêndice B. Nesta coleta foi escrita a frase "Boa tarde tchau" com uso de um botão do lado esquerdo do rosto. Entre a escrita das palavras "tarde" e "tchau", movimentos considerados inválidos são realizados, de modo a aumentar a dificuldade da tarefa de classificação. Estes movimentos compreendem:

- Movimento para direita, duas vezes;
- Movimento para baixo, duas vezes;
- Movimento para cima, duas vezes;
- Movimento para esquerda, desviando do botão pela frente, duas vezes;
- Movimento para esquerda, desviando do botão por trás, duas vezes;

Deste modo, adicionamos ao sistema quatro movimentos de mesma amplitude do movimento funcional, dificultando a classificação e extraindo mais informações sobre as limitações dos classificadores implementados. Ao total são segmentados 97 movimentos ao total, dos quais 34 são válidos.

A tabela 4.1 enumera como foi feita a coleta com os participantes, aqui mantido em sigilo seus nomes.

O usuário 5 apresenta movimentação involuntário do tipo coreodistônica, caracterizada por movimentos involuntários rápidos e bruscos (coreia) e a presença de co-contrações involuntárias de músculos, enrijecendo o movimento de partes do corpo. O usuário 6 apresenta

Usuário	Deficiência	Posição do Botão	Frase de teste
Usuário 1	—	Lado esquerdo	"Boa tarde tchau"
Usuário 2	—	Lado direito	"Boa tarde tchau"
Usuário 3	—	Lado direito	"Boa tarde tchau"
Usuário 4	—	Lado esquerdo	"Boa tarde tchau"
Usuário 5	Paralisia cerebral; coreodistonia	Lado esquerdo	"Boa tarde"
Usuário 6	Paralisia cerebral; coreoatetose	Lado esquerdo	"Boa tarde"

Tabela 4.1: Tabela de participantes na coleta de dados.

outra condição, chamada de coreoatetótica, na qual, além da coreia, apresentam-se movimentos lentos e repetitivos de membros (atetose) [74].

4.1 RECONHECIMENTO DE FACE

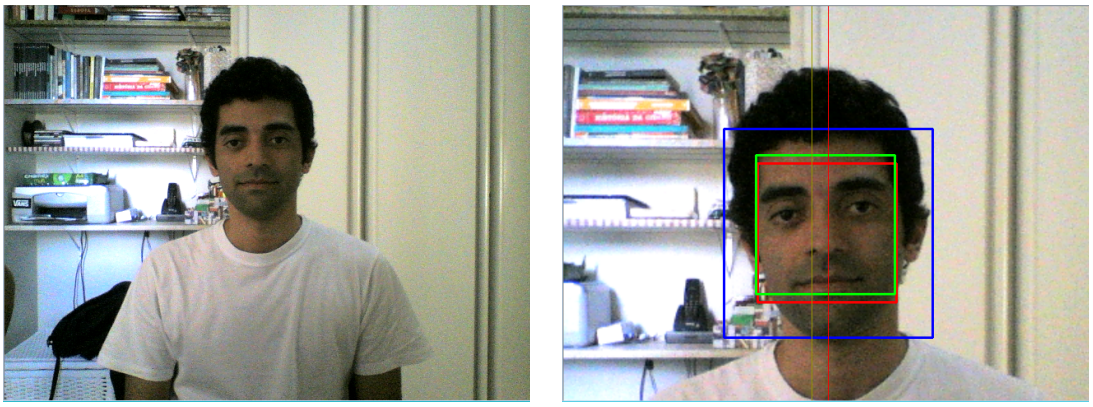
O sistema de reconhecimento de face fornece uma posição inicial para inicializar o rastreamento. Além disso, uma rotina de escala é feita, aumentando o rosto do usuário nos *frames* utilizados posteriormente, bem como uma demarcação de uma região de interesse do mesmo tamanho que a capturada pela câmera.

Na figura 4.1 está representado o resultado do reconhecimento de face, conforme descrito na seção 3.1. A partir dessa ampliação do rosto do usuário e posicionamento do mesmo no centro da imagem resultante, existe uma padronização nas amplitudes esperadas para tamanho de face, amplitude do movimento, amplitude de velocidade esperadas. Isto facilita a execução do sistema após treinado, bem como a determinação dos parâmetros de segmentação do movimento.

4.1.1 Discussão

A utilização do método de Viola-Jones, implementado na biblioteca OpenCV, não apresentou falhas. Em nenhuma das coletas com os usuários já citados foi necessário o registro manual da posição da cabeça. Em apenas um deles, usuário 3, o sistema teve problemas de encontrar o rosto.

Neste caso específico, havia uma fonte de luz muito forte posicionada ao lado direito do rosto. Esta luminosidade provavelmente prejudicou a codificação em níveis de cinza da imagem por completo, levando a não identificação do rosto. Como este comportamento esperado já era conhecido, apenas um descolamento do usuário se afastando deste emissor já foi suficiente para a convergência do algoritmo.



(a) (b)

Figura 4.1: Resultado da rotina de reconhecimento de face. Em (a) temos o frame inicial, e em (b) temos a estimativa da posição do rosto já feitas as operações de escala e definição da nova região de interesse.

A padronização do tamanho da imagem para ocupar no mínimo um terço da largura de toda região de trabalho 640×480, ocasionou que todas as imagens iniciais dos rostos fossem do tamanho 170×170. Esta padronização na comparação entre amostras de movimento coletadas de uma mesma pessoa e entre usuários diferentes.

A rotina de ampliação acabou por gerar imagens com pior qualidade de cores, ficando mais visíveis ruídos provenientes da câmera. Contudo este comportamento não prejudicou o rastreamento do alvo.

4.2 RASTREAMENTO DA FACE

A estimativa inicial obtida com o reconhecimento de face é utilizada para iniciar o rastreamento da face. A área de procura é criada com tamanho $1,5$ maior que a estimativa do alvo, e pode ser visualizada nas figuras 3.7 e 4.1.

A atualização da posição da face é obtida com o algoritmo de Meanshift utilizando a medida de similaridade PPM. O filtro de Kalman suaviza o deslocamento da estimativa do rosto, criando uma trajetória suave. Na figura 4.2 podemos ver como o movimento necessário para realizar o acionamento do botão, este posicionado do lado esquerdo do rosto.

4.2.1 Discussão

A utilização do rastreamento utilizando o método de Meanshift e a medida de similaridade PPM permitiu um rastreamento de face sem consumo excessivo de memória e processamento. Ainda com a presença das rotinas de gravação de vídeo, arquivos *.txt*, e demais

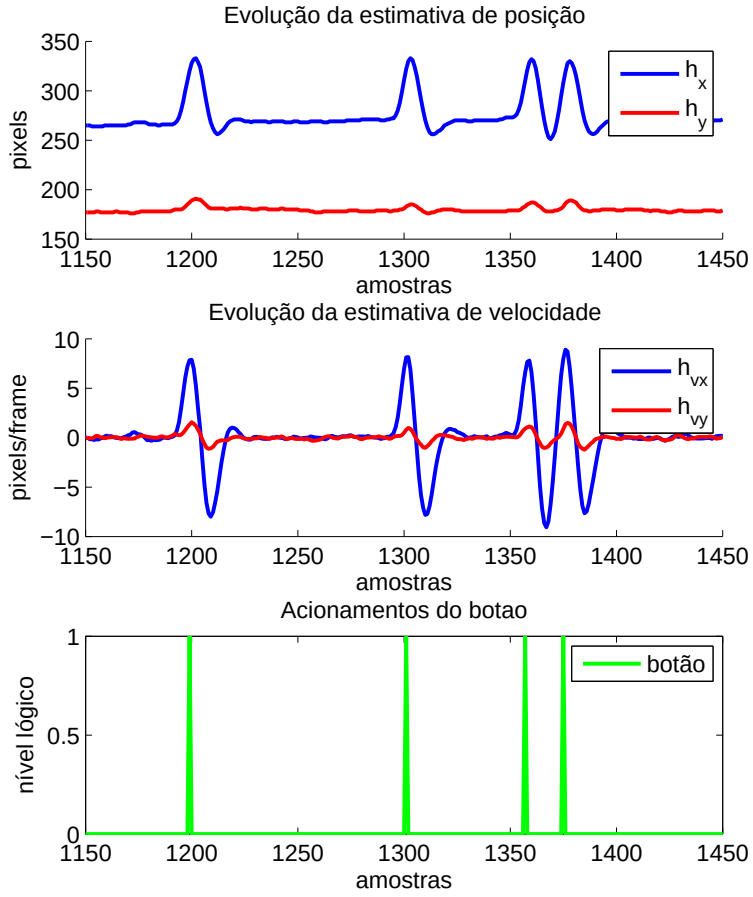


Figura 4.2: Dados coletados durante execução de quatro acionamentos do botão localizado do lado esquerdo do rosto. Estimativas geradas pelo algoritmo de Meanshift e filtradas pelo filtro de Kalman.

módulos, foi possível um rastreamento com uma atualização de 15 *fps*.

Aliado ao filtro de Kalman, o rastreamento se mostrou robusto à movimentação na imagem de fundo, como ocorrido com os usuários 2 e 4. A escolha por utilizar a codificação em HIV se mostrou também mais robusta que a utilizando o padrão RGB.

A definição a área de trabalho mais reduzida e em volta do rosto, proporcionou ainda mais robustez do sistema quando da oclusão da face, fato ocorrido durante coletas de dados prévias onde o movimento de cabeça de grande amplitude ocasionou que o cabelo de uma das candidatas encobrisse parte da face. Como a área possível de movimentação foi reduzida, assim que a oclusão terminou, o sistema voltou convergir para face desejada.

Este rastreamento contudo apenas fornece quatro componentes dinâmicas, não consegue detectar mudanças de escala do rosto (aproximação ou afastamento), e não percebe rotações do rosto quando executadas em baixas amplitudes. Em alguns momentos, principalmente nas condições de movimentos involuntários apresentados pelos usuários 5 e 6, o modelo

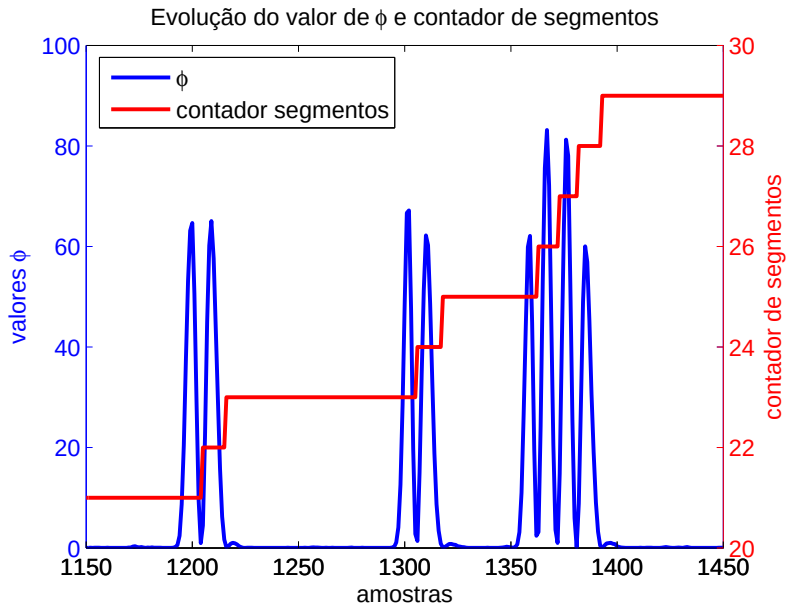


Figura 4.3: Registro de segmentos durante movimento que gerou 4 acionamentos do botão.

dinâmico escolhido para ser utilizado no filtro de Kalman é muito diferente do observado. Dessa maneira, o amortecimento introduzido na estimativa do vetor de estado acaba por criar efeitos adversos, e em alguns casos onde a amplitude horizontal é muito pequena, apenas deslocamento vertical é observado. Para os demais usuários o rastreamento foi eficiente, não gerando nenhum erro que é propagado para os próximos módulos.

Técnicas de rastreamento que permitem a medição da rotação de cabeça com câmeras monoculares devem ser implementadas em trabalhos futuros [75]. Contudo o sistema de rastreamento com Meanshift pode ser bastante útil para determinar a área de interesse onde a face está posicinada na imagem.

4.3 SEGMENTAÇÃO DE MOVIMENTOS

Baseado na medida de ZVC, foi possível segmentar os movimentos realizados a medida que estes foram observados. A metodologia utilizada e descrita na seção 3.3 permitiu a segmentação dos movimentos funcionais, e de todos os demais realizados pelo usuário. São descartados momentos de pouco movimento do usuário, diminuindo a tarefa de classificação realizada posteriormente.

Na figura 4.3 temos o resultado da medida φ_k para os mesmo movimento registrado na figura 4.2.

Podemos notar que o movimento é corretamente segmentado mesmo quando este é feito rapidamente e sem pausa.

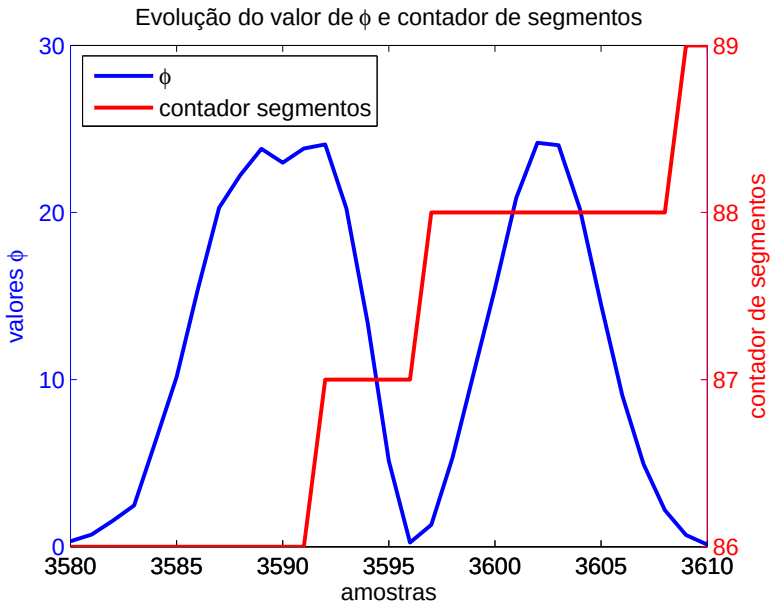


Figura 4.4: Segmentos múltiplos para um único movimento funcional.

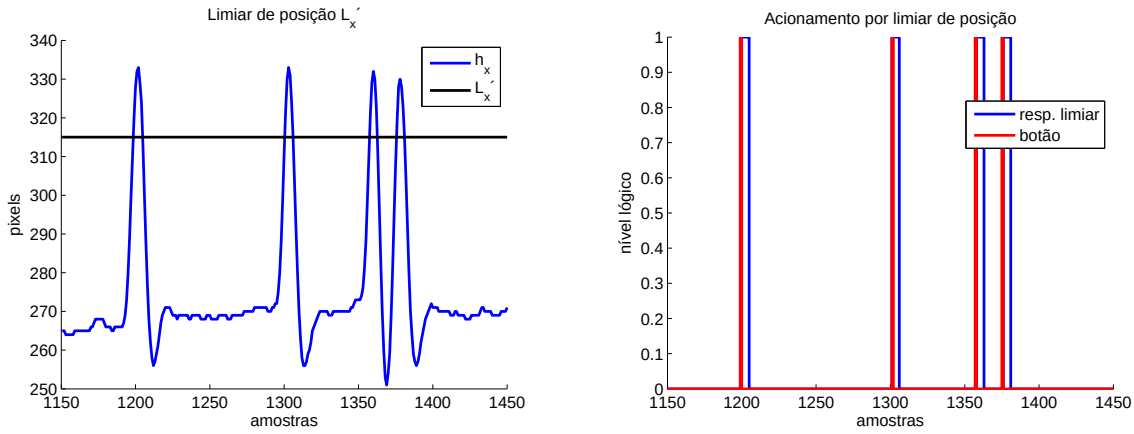
4.3.1 Discussão

A rotina de segmentação apresenta alguns problemas quando a medida de φ_k apresenta um vale próximo a um pico. Isto ocorre quando existe uma pequena desaceleração do movimento antes da sua finalização numa velocidade maior. Isto é representado na figura 4.4, onde um movimento funcional foi divido em dois.

A principal implicação deste comportamento é que aumentará a variância da população de número de amostras presentes nos segmentos válidos. Como a medida de n_a é utilizada no cálculo dos valores de $\boldsymbol{\mu}_\psi$ e $\boldsymbol{\Sigma}_\psi$, se muitos segmentos válidos sofrerem esta divisão, o efeito positivo de utilizar o número de amostras será neutralizado.

Em poucos momentos das coletas com usuários hígidos este problema foi observado, contudo, devido a movimentação involuntária apresentada pelos usuários 5 e 6, durante suas coletas este tipo de erro aconteceu frequentemente. Este fato prejudicou o desempenho do classificador HMM como descrito a seguir.

Para solucionar este problema de segmentação, outras máquinas de estado para determinação dos segmentos devem ser analisadas, bem como maneiras que possibilitem um processamento de uma sequência de segmentos a fim de verificar o ganho de informação em juntá-los, aumentando o número de amostras contidas em um movimento válido.



(a) (b)

Figura 4.5: Resultado da classificação de movimentos via limiar de posição.

4.4 CLASSIFICADOR POR LIMIAR DE POSIÇÃO

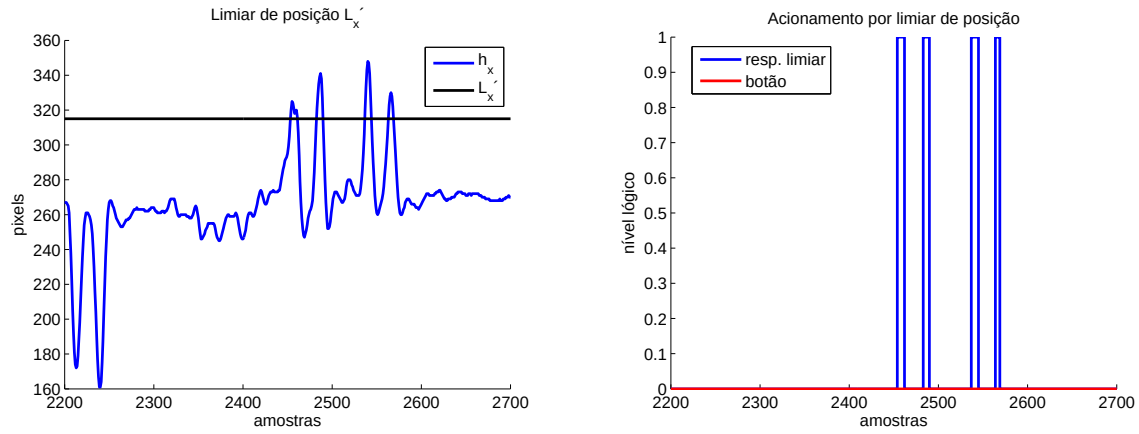
O limiar de posição é calculado a partir da amplitude máxima observada até o envio do comando de teclado 'c'. Contudo, para comparação deste método com o classificador HMM, foi necessário montar uma rotina de calibração e validação que permitisse utilizar os mesmos segmentos em ambos os métodos.

Desta maneira, uma lista de segmentos é utilizada para calcular o limiar L_x e L'_x. O valor de $|h_{xk}|$ máximo, atingido com estes segmentos selecionados será utilizado para cálculo dos limiares. Assim é possível verificar o comportamento do classificador por limiar nos demais movimentos.

A figura 4.5 apresenta a classificação dos movimentos da amostra representada nas figuras anteriores. Foram utilizados os dez primeiros movimentos válidos para calibração.

4.4.1 Discussão

O cálculo e utilização do limiar de posição é bastante simples, onde uma comparação de valores, sem tratamento das variáveis, indicará a classificação correta ou não de um movimento. Contudo, por nosso público alvo apresentar movimentação involuntária, é esperado que movimentos involuntários com amplitude similares a movimentos válidos devam coexistir. Na figura 4.6 temos o resultado do classificador por limiar treinado com os dez primeiros segmentos válidos. São avaliados os movimentos de validação já citados de amplitude superior ao movimento funcional. Vemos que não há como este classificador diferenciar segmentos nestas condições.



(a) (b)

Figura 4.6: Falso positivos detectados com uso do classificador por limiar.

4.5 CLASSIFICADOR HMM

4.5.1 Treinamento da Cadeia de Markov

A cadeia de Markov inicialmente é definida como ergódica, ou seja, a partir de qualquer um dos estados, é possível uma transição para todos os demais. Isto mostrou-se muito importante devido a variação dos possíveis vetores de características utilizados. O único fator que foi mantido entre as diversas iterações feitas para treino foi manter o número de estados em quatro.

Desta maneira,são construídos quatro *clusters* com o algoritmo de *k-means*. Os valores de média e covariância destes *clusters* são utilizados para inicializar os valores da matriz $\boldsymbol{B}$. Estes são os únicos parâmetros inicializados não randomicamente de todos que compõem a cadeia $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$.

Para os mesmos dez segmentos válidos iniciais podemos treinar a cadeia para os vetores de características $\boldsymbol{\phi}_k^i$, $\boldsymbol{\phi}_k^{ii}$, $\boldsymbol{\phi}_k^{iii}$. Os resultado de cada uma das etapas é apresentado comparando as três diferentes implementações.

4.5.1.1 Definição dos *Clusters*

Os diferentes vetores de características utilizados irão gerar distribuições diferentes de *clusters*. Os valores de média e covariância destes serão utilizados como estimativas iniciais para as *pdfs* que compõe a matriz de observação $\boldsymbol{B}$.

Sobrepondo os pontos obtidos em todos os dez segmentos de treino, obtemos a figura 4.7. Nesta podemos observar que a variabilidade de dados no eixo x do movimento é muito maior que no eixo y. Os valores de h_y e h_{vy} tem uma alteração muito pequena comparada com os dos valores de h_x e h_{vx}, fato que se explica pelo movimento de acionamento do botão

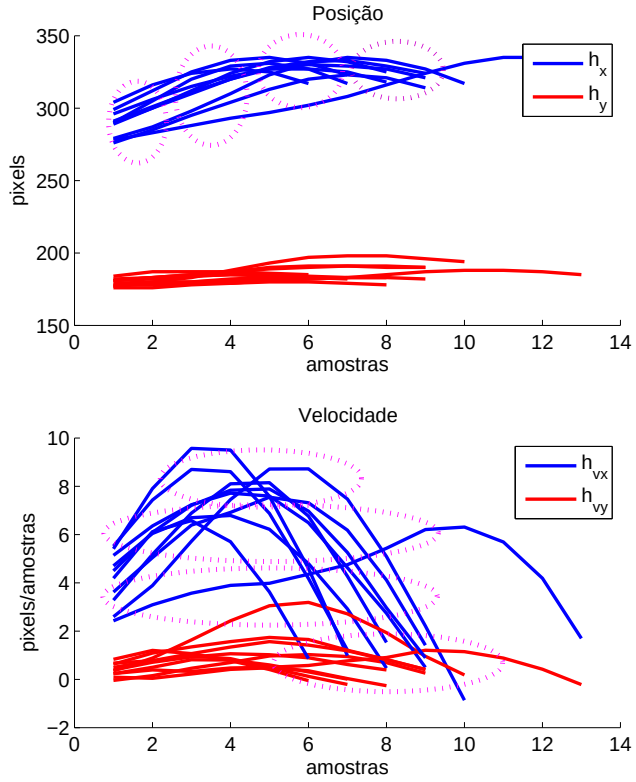


Figura 4.7: Sobreposição dos dez segmentos de movimento utilizados para treino das cadeias HMM.

do lado esquerdo do usuário ser praticamente horizontal.

Como os dados de h_x e h_{vx} possuem maior variação, espera-se que estes norteiem a definição dos *clusters*. Desta maneira, para o caso de utilizar a posição, temos que a distribuição de valores no tempo cria uma sequência de grupos que aparecem em sequência no decorrer do movimento. Enquanto que a distribuição de valores de velocidade apresenta grupos que apresentam-se intercalados.

Na figura 4.8 podemos comparar os resultados dos grupos para os três diferentes vetores de característica. Percebe-se que na figura 4.8a a informação contida nas variáveis de posição (h_x, h_y) influenciam muito mais na separação dos grupos que a velocidade (h_{vx}, h_{vy}). O *cluster* 3 por exemplo ocupa quase todos os valores observados de velocidade. Temos outros grupos onde isto ocorre de maneira menos evidente, como os *clusters* 1 e 4.

A distribuição obtidas nas três iterações obtiveram valores esperados e que vão ajudar na convergência do treinamento da cadeia. Os valores de média e covariância dos *clusters* é apresentado abaixo.

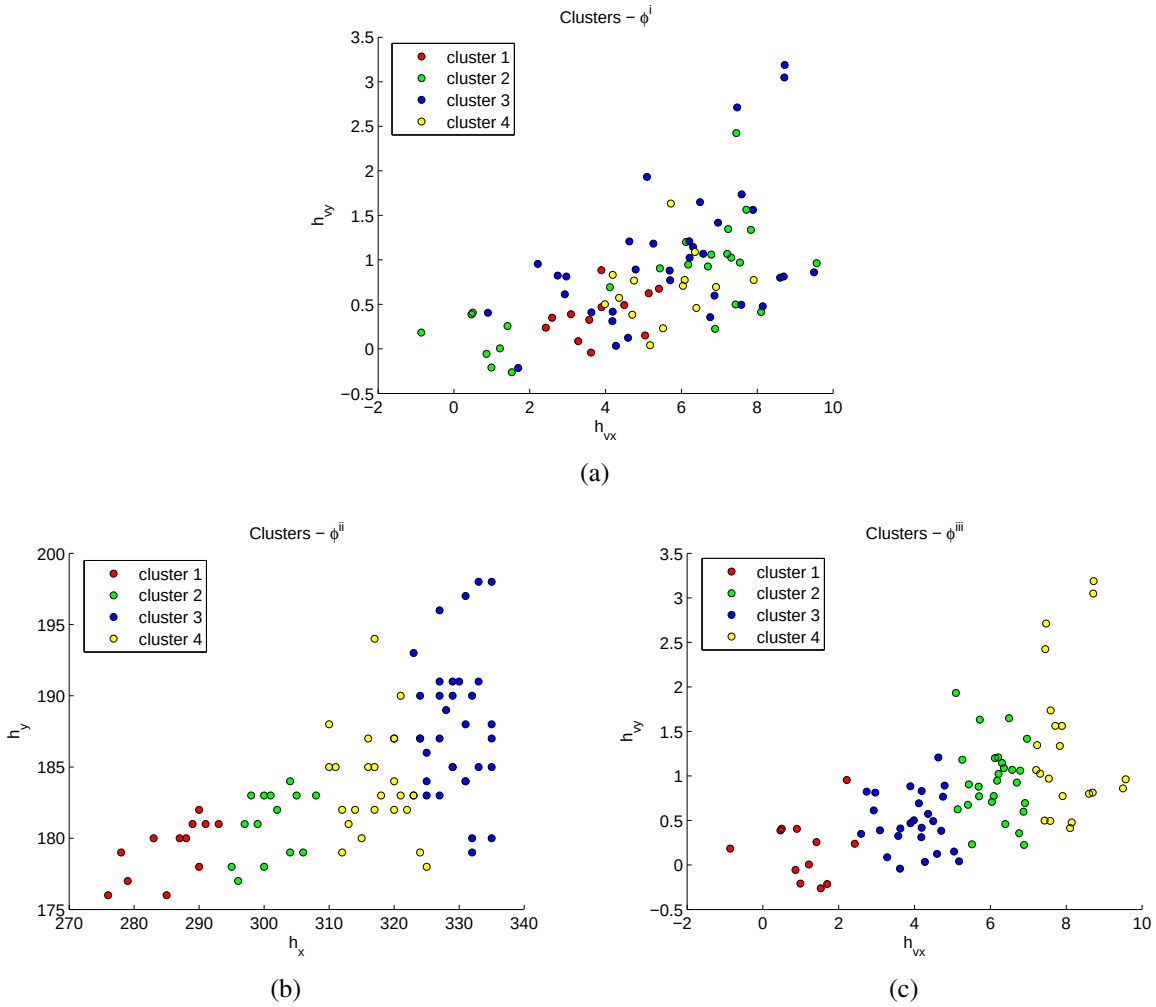


Figura 4.8: *Clusters* obtidos com mesma sequência de dados de treinamento, mas obtidos para diferentes vetores de características.

Para vetor de características ϕ^i temos:

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3,8686 \\ 0,3870 \\ 285,7500 \\ 179,2500 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0,9613 & 0,1008 & 3,2543 & 0,7062 \\ 0,1008 & 0,0692 & 0,7606 & 0,4109 \\ 3,2543 & 0,7606 & 31,1137 & 7,7955 \\ 0,7062 & 0,4109 & 7,7955 & 4,2046 \end{bmatrix} \tag{4.1}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 5,0283 \\ 0,7309 \\ 317,4800 \\ 183,8400 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 10,0190 & 1,3763 & -3,0785 & -2,6413 \\ 1,3763 & 0,3846 & -0,9271 & 0,7052 \\ -3,0785 & -0,9271 & 20,3434 & -2,3367 \\ -2,6413 & 0,7052 & -2,3367 & 13,2234 \end{bmatrix} \tag{4.2}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 5,7357 \\ 1,0204 \\ 329,3429 \\ 187,6571 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 4,9245 & 0,8641 & 1,5847 & -0,1724 \\ 0,8641 & 0,6043 & -0,0688 & 2,8821 \\ 1,5847 & -0,0688 & 13,4085 & 0,3269 \\ -0,1724 & 2,8821 & 0,3269 & 24,1144 \end{bmatrix} \tag{4.3}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 5,5776 \\ 0,6758 \\ 301,0714 \\ 181,0000 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1,2735 & 0,0960 & 0,7614 & -0,0673 \\ 0,0960 & 0,1470 & 0,2798 & 0,5613 \\ 0,7614 & 0,2798 & 15,4561 & 3,7692 \\ -0,0673 & 0,5613 & 3,7692 & 5,5386 \end{bmatrix} \tag{4.4}$$

Para vetor de características $\boldsymbol{\phi}^{ii}$ temos:

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 285,75 \\ 179,25 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 31,1137 & 7,7955 \\ 7,7955 & 4,2046 \end{bmatrix} \tag{4.5}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 301,0714 \\ 181 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 15,4561 & 3,7692 \\ 3,7692 & 5,5386 \end{bmatrix} \tag{4.6}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 329,3429 \\ 187,6571 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 13,4085 & 0,3269 \\ 0,3269 & 24,1144 \end{bmatrix} \tag{4.7}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 317,48 \\ 183,84 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 20,3434 & -2,3367 \\ -2,3367 & 13,2234 \end{bmatrix} \tag{4.8}$$

E para vetor de características $\boldsymbol{\phi}^{iii}$ temos:

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 1,1099 \\ 0,1749 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0,7604 & 0,0272 \\ 0,0272 & 0,1223 \end{bmatrix} \tag{4.9}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 6,1413 \\ 0,9400 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0,3407 & -0,0400 \\ -0,0400 & 0,1746 \end{bmatrix} \tag{4.10}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 3,9856 \\ 0,4824 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0,5248 & -0,0118 \\ -0,0118 & 0,1007 \end{bmatrix} \tag{4.11}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 8,0099 \\ 1,3367 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0,4951 & 0,0119 \\ 0,0119 & 0,7118 \end{bmatrix} \tag{4.12}$$

4.5.1.2 Arquitetura da cadeia

O treinamento das cadeias de Markov seguiu os passos determinados na seção 3.4.2.1. A seguir temos os parâmetros obtidos do treino das três HMM.

Para o vetor de características $\boldsymbol{\phi}^{i}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,9912 \\ 0 \\ 0 \\ 0,0088 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,4813 & 0 & 0 & 0,5187 \\ 0 & 0,7934 & 0,2066 & 0 \\ 0 & 0,0561 & 0,9439 & 0 \\ 0 & 0,6956 & 0 & 0,3044 \end{bmatrix} \tag{4.13}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 4,2954 \\ 0,4218 \\ 290,7644 \\ 179,7147 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1,1394 & 0,0726 & 5,2106 & 0,5395 \\ 0,0726 & 0,0743 & 0,8609 & 0,4180 \\ 5,2106 & 0,8609 & 63,6198 & 10,4181 \\ 0,5395 & 0,4180 & 10,4181 & 4,9102 \end{bmatrix} \tag{4.14}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 5,4604 \\ 0,6829 \\ 323,8601 \\ 183,4770 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 7,0926 & 0,9302 & 4,9083 & 1,7258 \\ 0,9302 & 0,3166 & -1,0100 & 0,9119 \\ 4,9083 & -1,0100 & 36,0856 & -4,2691 \\ 1,7258 & 0,9119 & -4,2691 & 6,6630 \end{bmatrix} \tag{4.15}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 4,8133 \\ 1,2061 \\ 328,8421 \\ 190,7581 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 7,3929 & 1,8961 & 6,8861 & 1,2726 \\ 1,8961 & 0,7919 & 0,9628 & 2,2175 \\ 6,8861 & 0,9628 & 22,4397 & -0,8545 \\ 1,2726 & 2,2175 & -0,8545 & 14,9589 \end{bmatrix} \tag{4.16}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 6,7020 \\ 0,8815 \\ 308,4891 \\ 182,2180 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1,0894 & -0,0165 & 3,1743 & 0,1341 \\ -0,0165 & 0,1407 & -0,2176 & 0,5826 \\ 3,1743 & -0,2176 & 39,8543 & 3,6786 \\ 0,1341 & 0,5826 & 3,6786 & 6,0852 \end{bmatrix} \tag{4.17}$$

Para o vetor de características $\boldsymbol{\phi}^{ii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,7333 \\ 0,2667 \\ 0 \\ 0 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,4513 & 0,5487 & 0 & 0 \\ 0 & 0,1332 & 0 & 0,8668 \\ 0 & 0 & 0,9146 & 0,0854 \\ 0 & 0 & 0,4546 & 0,5454 \end{bmatrix} \tag{4.18}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 286,8038 \\ 179,2078 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 35,4593 & 6,1656 \\ 6,1656 & 3,7768 \end{bmatrix} \tag{4.19}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 301,0782 \\ 181,1721 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 10,6372 & 2,1372 \\ 2,1372 & 4,9836 \end{bmatrix} \tag{4.20}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 328,3651 \\ 187,5473 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 21,3954 & 0,3202 \\ 0,3202 & 26,4669 \end{bmatrix} \tag{4.21}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 317,1677 \\ 183,4710 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 30,0329 & 1,0451 \\ 1,0451 & 5,6996 \end{bmatrix} \tag{4.22}$$

Para o vetor de características $\boldsymbol{\phi}^{iii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,3972 \\ 0,0372 \\ 0,5656 \\ 0 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,0002 & 0 & 0,9998 & 0 \\ 0 & 0,5111 & 0,3040 & 0,1849 \\ 0,5526 & 0,4438 & 0,0035 & 0,0001 \\ 0 & 0,3985 & 0 & 0,6015 \end{bmatrix} \tag{4.23}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 1,7600 \\ 0,1484 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1,7611 & 0,0365 \\ 0,0365 & 0,0690 \end{bmatrix} \tag{4.24}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 6,3668 \\ 0,9242 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0,8306 & -0,1316 \\ -0,1316 & 0,1447 \end{bmatrix} \tag{4.25}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 4,1028 \\ 0,4955 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0,7774 & -0,1332 \\ -0,1332 & 0,0840 \end{bmatrix} \tag{4.26}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 8,0843 \\ 1,5994 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0,6501 & -0,0894 \\ -0,0894 & 0,6920 \end{bmatrix} \tag{4.27}$$

4.5.2 Discussão

O treinamento da cadeia não sofre alterações conforme é alterado o número de estados da cadeia, bem como a alteração do vetor de características. Estas são características muito usuais, permitindo que com grande facilidade uma grande quantidade de possibilidades seja testada, mantendo-se a mesma abstração do modelo.

Percebemos que os valores das *pdfs* $N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ são alterados desde sua estimativa inicial criada pelos *clusters*, até o final do treinamento utilizando o algoritmo de Baum-Welch. Contudo, estes ajustes não são de grande magnitude, sendo nossa relação entre os grupos iniciais e os estados da cadeia ainda mantida.

Como já comentado na seção 4.5.1.1, as cadeias onde predominaram os *clusters* formados a partir de valores de posição, ou seja, as que utilizaram os vetores de características $\boldsymbol{\phi}^i$ e $\boldsymbol{\phi}^{ii}$, percebe-se que a arquitetura obtida do treino é de uma HMM sequencial. Pode-se ver que as matrizes $\boldsymbol{A}$ possuem linhas com apenas dois termos não nulos, sendo que um deles sempre é o da diagonal principal.

Já para a HMM que foi treinada a partir do vetor $\boldsymbol{\phi}^{iii}$, percebe-se que o número de combinações possíveis de transições de estados e possíveis realizações da cadeia é muito maior.

Os vetores de probabilidade inicial $\boldsymbol{\pi}_i$ também acompanham o mesmo raciocínio feito para a matriz de transição. Para cadeias onde é mais definido o modelo de sequência, o vetor apresenta um elemento com valor muito próximo a 1.

Contudo, no advento de poucas amostras de movimentos válidos, seja por problemas na segmentação, como descrito na seção 4.3.1, ou pelo posicionamento do botão muito próximo ao rosto dos usuários, muitas vezes são formados *clusters* com número reduzido de elementos.

Este fato inviabiliza o uso de um número fixo de *clusters* para montar a cadeia de Markov. Principalmente, ocorrem situações onde a matriz de transição apresenta até duas linhas onde a probabilidade de permanecer ou alterar o estado é 1. Isto pode ser visualizado em algumas das matrizes de transição de estado $\boldsymbol{A}$ dos usuários 4, 5 e 6.

A interpretação que temos é que ao invés de mapear uma sequência de estados que compõe um movimento, a cadeia mapeia os possíveis estados da variável de estados ao final do movimento. A performance deste tipo de cadeia acaba sendo pior de quando temos mais amostras.

Uma das maneiras que podem contornar este problema seria o aumento da taxa de cap-

tura. Isto pode ser adquirido apenas excluindo alguns módulos necessários para análise do sistema, mas que nada servem ao usuário final. Um deles em principal é o módulo de registro em vídeo da utilização.

Com as coletas feitas é difícil escolher se há uma regra definida para escolha dos vetores de estado $\boldsymbol{\phi}^{i}$, $\boldsymbol{\phi}^{ii}$ e $\boldsymbol{\phi}^{iii}$. Percebemos que deve ser realizado uma análise quantitativa da performance da cada uma delas para escolha na implementação em C++. A característica dos movimentos dos usuários só nesta pequena amostra de 7 indivíduos já mostrou diversas maneiras de acesso ao botão, tanto em amplitude, velocidade e perfil de aceleração do movimento. Permitir que o sistema escolha entre as melhores opções parece ser melhor do que economizar em número de iterações de treinamento das cadeias candidatas.

4.6 CLASSIFICAÇÃO DA HMM POR VEROSSIMILHANÇA

A partir da cadeia treinada para cada um dos vetores de característica, é possível avaliar o valor de $LL = log(P(\boldsymbol{O}|\lambda))$ e visualizar como os movimentos segmentados são identificados pelas cadeias. Este teste é feito com os 87 segmentos restantes de validação, dos quais 24 são válidos.

Na figura 4.9 podemos verificar como as características dos vetores de dados e das arquiteturas das cadeias alteram como são identificados os segmentos. Para facilitar a visualização, apenas dados de LL num intervalo $(0, -300)$. Isto ocorre porque vários segmentos muito diferentes geram valores muito negativos para LL, prejudicando a visualização dos valores maiores, os corretamente classificados pela cadeia.

4.6.1 Discussão

Ao analisarmos a disposição dos pontos considerados válidos dos inválidos podemos perceber três características. Muitos segmentos inválidos não são apresentados nos gráficos 4.9a, 4.9b e 4.9c, enquanto que todos os 34 segmentos válidos estão representados dentro do intervalo $(0, -300)$. Percebemos que quando utilizado, o vetor de características $\boldsymbol{\phi}^{i}$, os últimos movimentos válidos executados tiveram uma pontuação muito diferente dos demais, desta maneira seriam facilmente considerados como falso negativos na implementação do classificador.

Para as cadeias treinadas com o vetor $\boldsymbol{\phi}^{ii}$ e $\boldsymbol{\phi}^{iii}$ houve uma generalização maior da classificação, onde todos os valores válidos conseguem ser separados mais facilmente do restante do grupo.

Por último percebemos que existem alguns segmentos inválidos com valores de LL muito próximos à média dos valores válidos encontrados. Isto é devido a segmentos muito curtos, muitas das vezes gerados por ruídos no rastreamento do movimento, ou por movimentos

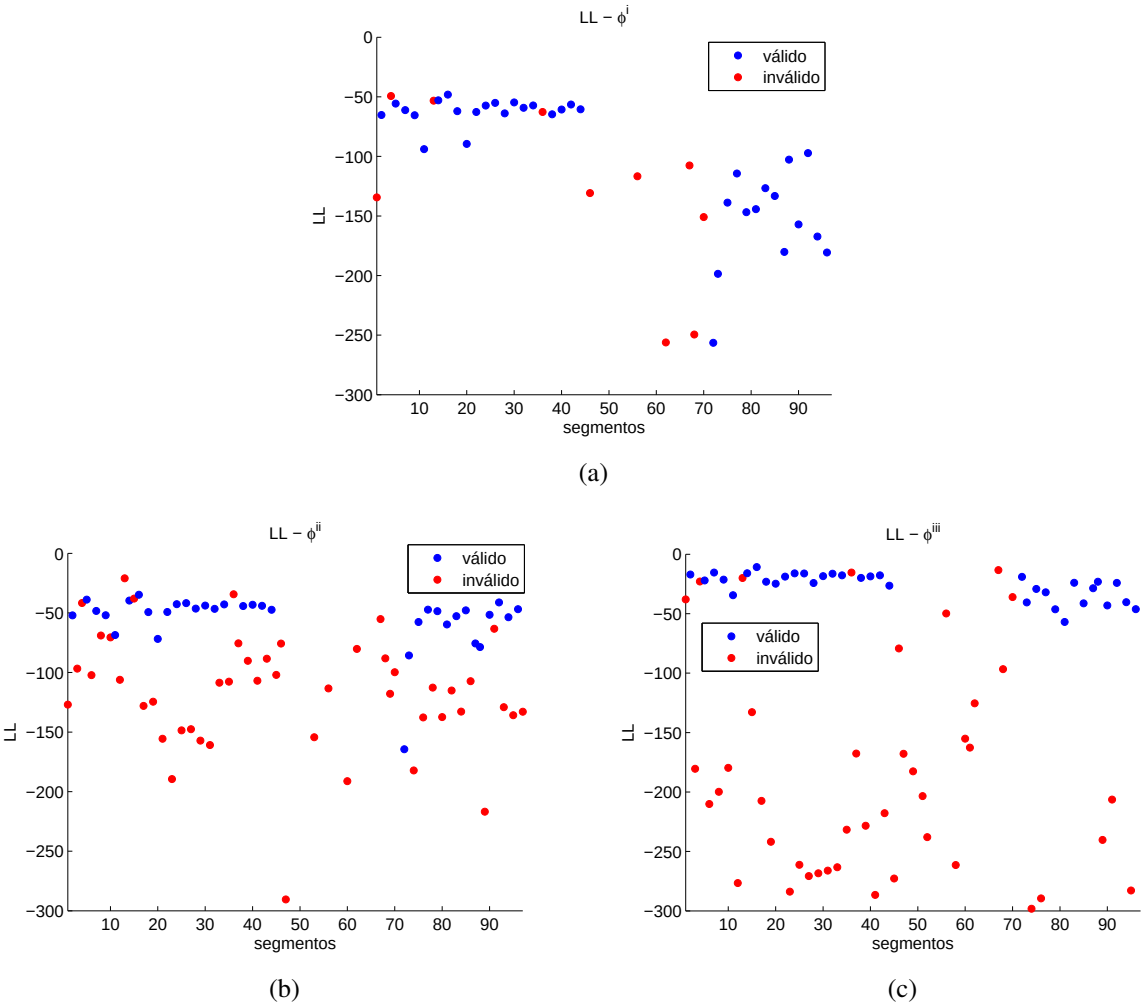


Figura 4.9: Valores de LL obtidos com os segmentos de validação.

involuntários. Esta é a principal razão por utilizarmos a informação sobre o tamanho dos segmentos para construir a medida ψ e a partir dela decidir sobre a classificação do segmento.

Para os usuários hígidos a necessidade de uso da variável ψ é menos evidente que nos candidatos que apresentam dificuldade de controle do movimento de cabeça. Nos gráficos de LL presentes no Apêndice B podemos ver que o número de elementos inválidos com valores semelhantes aos válidos é enorme. Um classificador com apenas esta informação de escolha gera uma taxa de falso positivos que inviabiliza sua aplicação.

4.7 CLASSIFICAÇÃO COM USO DA DISTÂNCIA DE MAHALANOBIS

Com o resultado dos valores de LL para os segmentos utilizados no treino de cada uma das cadeias, e do número de amostras n_a de cada um destes segmentos. Construímos o vetor $\boldsymbol{\psi} = [LL \quad n_a]^T$ para cada um dos conjuntos de treinamento, e calculamos a média $\boldsymbol{\mu}_\psi$ e covariância $\boldsymbol{\Sigma}_\psi$ desta população.

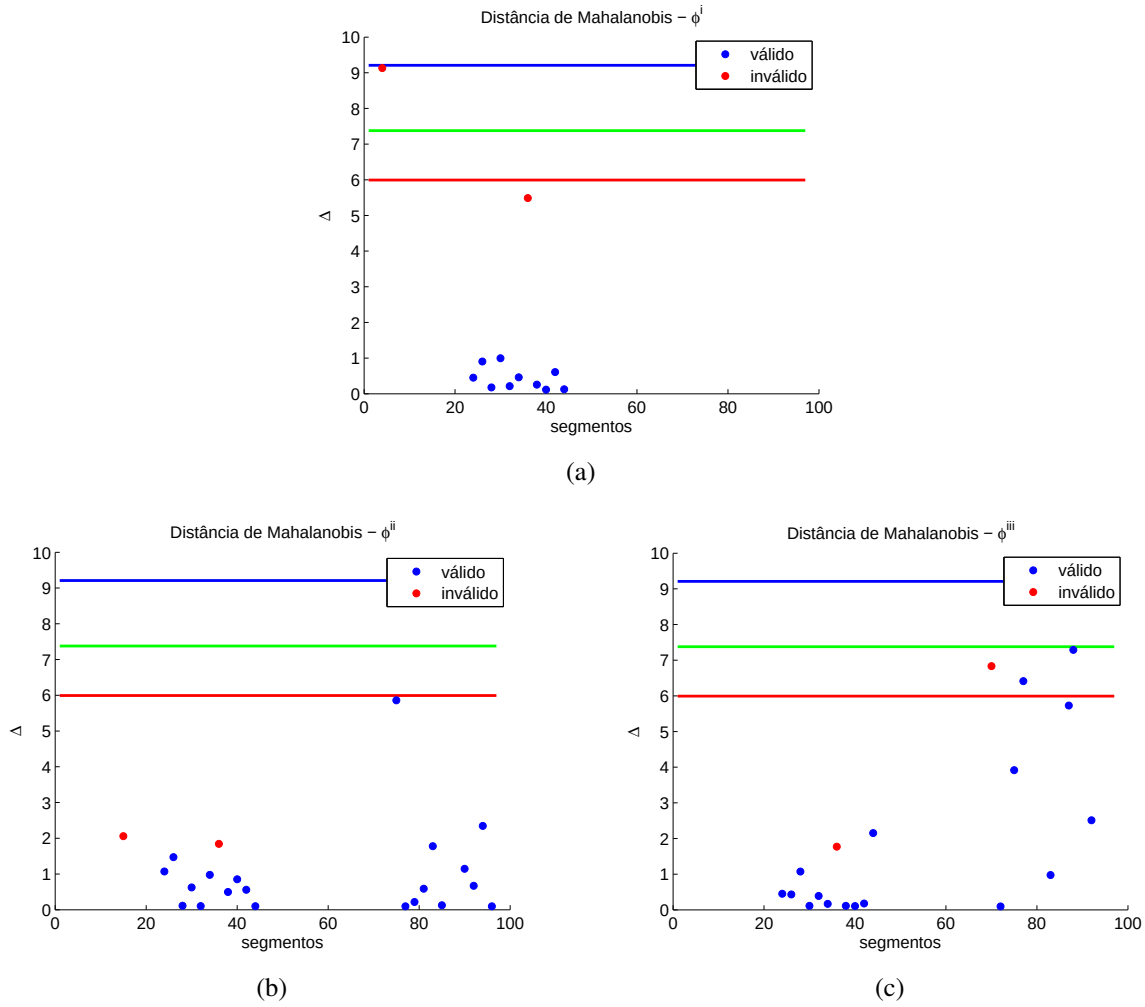


Figura 4.10: Valores de Δ^2 obtidos com os segmentos de validação.

Utilizando a equação (3.21), encontramos para cada segmento o valor da distância de Mahalanobis Δ^2 e a partir destes obtemos a figura 4.10

As linhas horizontais correspondem a três limiares C diferentes calculados a partir de três níveis de confiança α, ou seja, a probabilidade de que o segmento analisado não pertence à população $N(\boldsymbol{\mu}_\psi\,, \boldsymbol{\Sigma}_\psi)$. Desta maneira, valores $\Delta^2 > C = \chi^2_2(1-\alpha)$, com α igual a $0,05$, $0,025$ e $0,01$ são utilizados para classificar segmentos como inválidos.

Tomando como padrão utilizar o valor $\alpha = 0,01$, temos na tabela 4.2 os resultados para estes classificadores.

4.7.1 Discussão

Ao analisarmos os valores da tabela 4.2 percebemos que os três classificadores obtiveram uma performance semelhante no tratamento de falso positivos, contudo a cadeia HMM que utiliza o vetor de características $\boldsymbol{\phi}^i$ obteve menos de 50% de acerto de verdadeiros positivos.

Cadeia HMM	Nº de Verdadeiros Positivos - *VP*	Taxa de Verdadeiros Positivos - *vp*	Nº de Falso Positivos - *FP*	Taxa de Falso Positivos - *fp*
ϕ^{i}	10	0,42	2	0,03
ϕ^{ii}	20	0,83	2	0,03
ϕ^{iii}	17	0,71	2	0,03

Tabela 4.2: Tabela de participantes na coleta de dados.

Como já discutido previamente, a cadeia que utiliza ϕ^{i} apresenta um poder menor de generalização, pois apresenta uma arquitetura que permite uma menor variabilidade de transições possíveis. Ao utilizar menos informação, as cadeias que utilizam ϕ^{ii} e ϕ^{iii} tiveram um poder mair de generalização, sem comprometer a taxa de falso positivos.

Ao analisarmos os gráficos de Δ^2 das figuras do Apêndice B, percebemos uma melhora significativa na separação dos dados válidos e inválidos dos usuários 2 e 4 em relação aos gráficos que utilizam a medida LL. Esta é a principal vantagem de utilizar a informação do número de amostras n_a para a decisão do classificador.

Nos usuários 5 e 6 este comportamento não foi observado. Os segmentos válidos em sua maioria apresentam o problema reportado na seção 4.3.1, pois em diversos momentos a medida φ apresenta vales antes de chegar a um valor máximo e reduzir a um valor inferior ao limiar de início de movimento.

Desta maneira, alguns dos segmentos válidos encontrados são de tamanho muito semelhante aos segmentos provenientes de movimentação involuntária. Como já explicado na seção 4.6.1, muitos deste movimentos apresentam valores de LL também muito semelhantes aos válidos. Nestas condições é de se esperar que uma grande taxa de falso positivos na aplicação destes classificadores por HMM.

4.8 ANÁLISE DO DESEMPENHO DOS CLASSIFICADORES

Analisar os classificadores somente com treino utilizando os dez primeiros segmentos válidos gera pouca informação sobre o classificador, já que não estamos validando o sistema proposto com todas as amostras válidas. Nestas condições, uma análise por *cross-validation* é muito utilizada. Neste tipo de avaliação, são criados *k-folds*, conjuntos, e a cada iteração um *fold* é escolhido para validação e os demais para treino. Assim, são utilizados para teste, os resultados médios obtidos sobre verdadeiros positivos e falso positivos de todos os testes, estimando melhor o comportamento do classificador com outras amostras [66].

Escolhemos utilizar o método *3-fold cross-validation* por apresentar uma relação de equilíbrio entre o número de dados que são validados e o número de dados usados para treino. Com um terço dos valores válidos sendo utilizados para validação, temos uma boa resolução

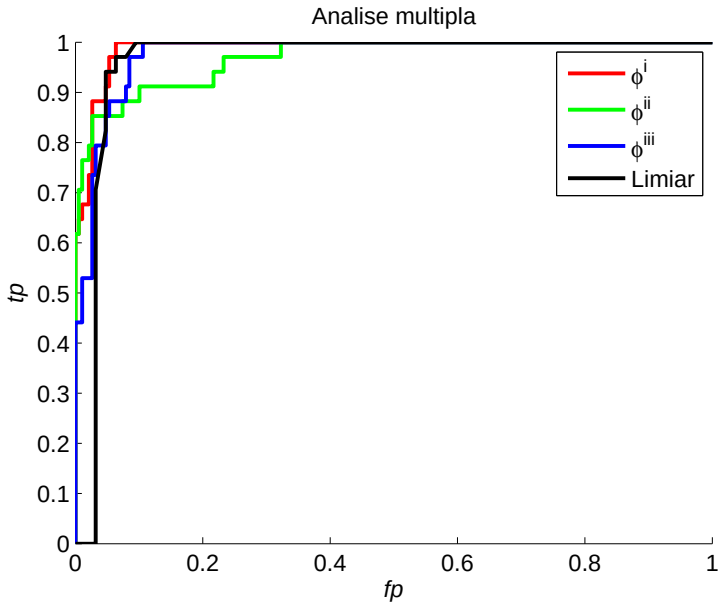


Figura 4.11: Gráfico ROC comparando classificadores diferentes com a aplicação do *3-fold crossvalidation*.

dos resultados de tp para diferentes valores de limiar para a medida Δ^2.

Ao testar vários valores para o limiar $C = [-\infty\,,\,+\infty]$, é possível criar um gráfico que sintetiza a variação de valores de *tp* e *fp* a partir dos valores de C. Este gráfico chamado de *Receiver operating characteristics*, ROC, é extremamente útil para analisar diferentes métodos de classificação. Partindo de valores de limiares muito baixos para cada um dos classificadores, no instante inicial não é classificado nenhum item válido, e no instante final, com limiares muito altos, todos os itens são classificados como válidos.

A figura 4.11 apresenta o gráfico ROC obtido com o *3-fold cross validation*. Podemos observar que todos os classificadores aqui utilizados apresentam altos valores de tp em relação aos valores de fp. Contudo o classificador por limiar de posição, por não ser robusto a diferenciar movimentos inválidos de mesma amplitude de movimentos válidos, apresenta uma taxa de falso positivos basal mais alta que os demais.

Para comparar as performances dos classificadores empregados de maneira quantitativa, pode-se utilizar a área abaixo da curva ROC. Com esta medida, é possível entender como se comportam os classificadores sem visualizar suas curvas ROC, quanto melhor o desempenho, o valor da área deve se aproximar de 1. Os valores da tabela 4.3 foram obtidos com a rotina de *3-fold cross-validation* para todos os candidatos da pesquisa, e estão representados pelo valor médio e seu intervalo 3-sigma da área abaixo da curva ROC.

Usuário	ϕ^i	ϕ^{ii}	ϕ^{iii}	***Limiar de posição***
Usuário 1	$0,997 \pm 0,008$	$0,997 \pm 0,007$	$0,997 \pm 0,008$	$0,964 \pm 0,030$
Usuário 2	$0,916 \pm 0,061$	$0,846 \pm 0,132$	$0,976 \pm 0,03$	$0,94 \pm 0,146$
Usuário 3	$0,981 \pm 0,055$	$0,829 \pm 0,281$	$0,982 \pm 0,017$	$0,995 \pm 0,021$
Usuário 4	$0,866 \pm 0,198$	$0,817 \pm 0,236$	$0,853 \pm 0,18$	$0,926 \pm 0,146$
Usuário 5	$0,657 \pm 0,096$	$0,642 \pm 0,315$	$0,62 \pm 0,348$	$0,892 \pm 0,163$
Usuário 6	$0,772 \pm 0,389$	$0,78 \pm 0,425$	$0,779 \pm 0,599$	$0,947 \pm 0,016$

Tabela 4.3: Resultados dos classificadores utilizando área abaixo da curva ROC.

4.8.1 Discussão

Para o conjunto de dados utilizados para descrever nossos resultados nessa seção, concluímos que o classificador HMM, utilizando principalmente o vetor ϕ^i e ϕ^{iii} pode gerar um classificador com performance de acerto semelhante ao classificador por posição, mas com uma qualidade muito desejável para acesso ao computador, e principalmente para uso de pranchas dinâmicas: a baixa taxa de falso positivos.

Isto fica claro quando um movimento mal interpretado gera um acionamento que deve ser corrigido pelo usuário. No caso da escrita de uma sentença por exemplo, um acionamento mal interpretado no momento de escolha da coluna, gerará um caractere indesejado e desta maneira necessitará de dois movimentos corretamente interpretados para corrigir o erro.

Resultados semelhantes foram encontrados para os usuários 1, 2 e 3. Seus movimentos apresentaram características que beneficiam a aplicação do classificador HMM: movimentos segmentados sem divisão e número de amostras uniforme entre os segmentos válidos identificados. Para o usuário 1, todas as cadeias HMM obtiveram um gráfico ROC similar e superior ao gráfico do limiar de posição, para o usuário 2 a cadeia HMM que utiliza ϕ^{iii} foi a que obteve melhor desempenho, e para o usuário 3, ambos vetores ϕ^i e ϕ^{iii} obtiveram melhores resultados.

Isto concorda com nossa interpretação que utilizando apenas a informação de posição, são criadas cadeias que apresentam poucas possíveis transações entre estados, cadeias sequenciais. O uso de componentes de velocidade acaba por gerar cadeias com maior capacidade de generalização. Contudo, o nível de diversidade de movimentos presentes na sequência utilizada de treinamento influencia diretamente na capacidade de generalização em qualquer um dos casos.

Para os usuários 4, 5 e 6 foram observados problemas na geração de segmentos de treino com um número de amostras substancial. Para 4 o problema estava na proximidade do botão do rosto do usuário, enquanto que para 5 e 6, a presença de movimentos involuntários acabou por criar segmentos muito curtos.

Vale salientar que a performance do classificador por limiar de posição também se dete-

riorou conforme a amplitude de movimentos válidos também não foi homogênea. Isto pode ser explicado por problemas relativos ao rastreamento, mas no caso dos usuários 5 e 6, muito se deve ao movimento involuntário presente. O acionamento do botão muitas vezes foi sutil e muito vezes bastante agressivo, deslocando o botão muitas vezes de lugar. Para o usuário 6 que apresenta um movimento mais controlado, foi obtido um desempenho melhor para todos classificadores, ficando claro que a influência da movimentação involuntária é muito importante e de difícil modelamento.

5 CONCLUSÕES

Essentially, all models are wrong,
but some are useful.
George E. P. Box

Muitas pessoas com deficiência motora grave são extremamente dependentes de seus cuidadores, tanto para atividades de higiene, locomoção, alimentação e comunicação. Quadros de paralisia cerebral, lesões cerebrais implicam muitas vezes na redução dos movimentos e na presença de movimentos involuntários, enquanto que doenças degenerativas como esclerose lateral amiotrófica na implicam na perda progressiva dos movimentos.

Para muitos, a capacidade de expressar suas intenções, seus desejos ou até mesmo pedir ajuda é comprometida, sendo atenuada com a ajuda de cuidadores. Contudo, apenas a interpretação de gestos e sons contém pouca informação, limitando a conversas onde são feitas perguntas com respostas "sim"ou "não"somente.

Diversas ferramentas de comunicação alternativa e aumentativa são utilizadas para atenuar essa dependência e prover as pessoas com deficiências com maior potencial de comunicação. O uso de pranchas pictográficas ou alfabéticas mostra como com apenas um movimento funcional, um indivíduo pode escolher entre letras e símbolos que irão expressar com melhor precisão seus desejos.

Contudo, hoje isto apenas não basta. Nosso cotidiano é cada vez mais dependente de ferramentas computacionais. Infelizmente muitas delas não apresentam um acesso universal, que permita que pessoas que não apresentam movimento pleno de braços, mãos e dedos usufrua de ferramentas de comunicação e produção de conteúdo.

Esta limitação é muito séria, porque a partir de um computador o usuário pode realizar tarefas que não refletem sua dependência com cuidadores, ou sua dificuldade de comunicação e locomoção. Todo este potencial deve ser sim aproveitado e utilizado, potencializando indivíduos a se inserirem na sociedade, e de se comunicar livremente.

Ferramentas de comunicação alternativas aliadas a ferramentas de acesso ao computador são de extrema necessidade para este grupo de pessoas, e muitas soluções existem para diversos quadros de comprometimento motor. Contudo, para a população de indivíduos que não apresenta fala articulada, e ainda apresenta movimentos involuntários, não são oferecidas opções de acesso variadas, basicamente restringindo-se ao uso de botões mecânicos.

O grupo de pessoas que acionam botões mecânicos para acesso ao computador com movimentos de cabeça é objetivo deste estudo. Seus movimentos funcionais são observáveis com o uso de *webcans* já presentes na maioria dos dispositivos, e a partir da imagem do usuário, nos propusemos a criar um software que permitisse utilizar dos mesmos movimen-

tos funcionais de cabeça para uso do computador. Além disso, acreditamos que é possível modelar o movimento funcional de forma a evitar acionamentos involuntários do sistema.

Esta preocupação em reduzir o número de falso positivos baseia-se no fato que ao acionar de forma não planejada uma prancha dinâmica ou uma função de um software, múltiplos outros acionamentos corretos devem ser feitos. Acionamentos voluntários exigem da concentração do usuário, bem como exigem fisicamente deste. Estratégias de redução do número de acionamentos incorretos de sistemas de acesso ao computador são extremamente necessárias.

Propusemos dois sistemas diferentes, o primeiro utiliza um simples limiar de posição e o segundo utiliza uma cadeia de Markov. Ambos sistemas são treinados a partir de movimentos rotulados como válidos, identificados durante a escrita de uma sentença com uso de botão mecânico e uma prancha dinâmica com varredura. De antemão, o sistema de limiar de posição apresenta uma inspiração direta na aplicação de botões mecânicos. O modelamento do movimento funcional é feito apenas com base na amplitude esperada de movimento, contudo, movimentos involuntários de mesma amplitude são tratados da mesma maneira.

Sistemas de rastreamento por visão computacional podem fornecer outras informações além da posição no eixo x de um objeto. A partir de um algoritmo de Meanshift utilizando a medida de similaridade de PPM, obtivemos um rastreamento de rosto do usuário, estimando velocidade e posição no plano (x, y) da imagem. Muitas outras técnicas permitem que outras variáveis como rotação do rosto sejam estimadas, contudo neste trabalho optamos por utilizar um rastreamento simples mas efetivo, que gerasse um vetor de estados com até quatro variáveis, mas que esperava-se já ser capaz de representar os movimentos esperados da população de usuários em questão.

A aplicação da cadeia de Markov mostrou-se extremamente flexível e apresenta poucos parâmetros de inicialização. Os principais parâmetros definidos foram o número de estados da cadeia, quatro, e a utilização de dados observáveis de distribuição contínua. O treinamento das cadeias com seus diferentes vetores características apenas utiliza-se de dados considerados válidos, e a avaliação da cadeia a um determinado movimento é feita com uma variável contínua que representa a chance desse movimento ser representável pela cadeia.

Para usuários testados que não apresentam movimentação involuntária, a performance dos classificadores HMM chegou a superar o classificador a partir do limiar de posição. Isso se deve principalmente pelo modelamento mais preciso do movimento válido, desprezando movimentos involuntários de amplitudes similares.

Para usuários hígidos mas que apresentaram movimentos funcionais muito pequenos, gerando segmentos válidos com poucas amostras, as cadeias resultantes apresentam mais de um estado que não permitem transição para outros. Dessa maneira, a dinâmica do movimento funcional fica resumida apenas ao seu estado final. Apesar deste inconveniente, a generalização chegou a obter taxas de verdadeiros positivos de $0,7$ para taxas de falso positivos de

$0,2$.

Já para usuários que apresentam movimentação involuntária, o treinamento da cadeia foi prejudicado por limitações no algoritmo de segmentação utilizado. Movimentos funcionais foram divididos devido a mudança brusca na velocidade, obtendo novamente segmentos de movimento válidos com poucas amostras. A maior presença de segmentos inválidos gerados por movimentos involuntários acaba por aumentar o número de falso positivos encontrados para estas amostras de dados. Os resultados obtidos inviabilizam o uso destes classificadores.

A presença de movimentação involuntária não só comprometeu a classificação com cadeias de Markov, mas também o uso do limiar de posição, demonstrando como seu efeito impede o correto modelamento do movimento desejado. Com usuários com deficiência mostrou-se claro que além disso, seu movimento funcional de cabeça ainda apresenta variações dinâmicas pouco perceptíveis ao olho nu, talvez demandando múltiplas cadeias para a correta interpretação.

Soluções que permitam a criação de segmentos com um número maior de amostras podem potencializar a capacidade de classificação das cadeias HMM, diminuindo sua taxa de falso positivos e talvez permitindo uma performance superior a encontrada com o sistema de limiar. Contudo, o tratamento do movimento involuntário por variações dos métodos aqui propostos, bem como o uso de outros sistemas de identificação, são necessários para que a substituição do botão mecânicos seja feita sem perdas pelo uso apenas de um software e uma *webcam*.

Trabalhos voltados a desenvolver as potencialidades de indivíduos com deficiência , levando em conta suas limitações físicas, são de grande importância para a população com deficiências motoras graves. Muito ainda pode ser feito além da simples aplicação de um botão mecânico, e soluções a partir de visão computacional tornam-se excelentes candidatas pela sua facilidade de acesso, e permitirem interpretar cada vez mais movimentos funcionais do usuário. Acreditamos que este projeto é um passo inicial nesta direção, principalmente voltando-se para um grupo de pessoas com sérias demandas ainda não atendidas.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] L. M. Borges, *Cartilha do Censo 2010 - Pessoas com Deficiência.* Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência (SDH/PR), 2012.

[2] "Secretaria nacional de promoção dos direitos da pessoa com deficiência." [Online]. Available: http://www.pessoacomdeficiencia.gov.br/app/sobre-a-secretaria

[3] *Tecnologia Assistiva*. Secretaria Especial dos Direitos Humanos, 2009.

[4] R. Bersch. (2013) Introdução à tecnologia assistiva. [Online]. Available: http://www.assistiva.com.br/Introducao_Tecnologia_Assistiva.pdf

[5] "Centro Nacional de Referência em Tecnologia Assistiva." [Online]. Available: http://www.cti.gov.br/cnrta

[6] R. R. Carlos Gonçalves, Antônio Padilha Lanari Bo, "Tracking head movement for augmentative and alternative communication," in *Simpósio Brasileiro de Automação Inteligente*. XI Simpósio Brasileiro de Automação Inteligente (SBAI 2013), 2013.

[7] J. Light and K. Drager, "AAC technologies for young children with complex communication needs: state of the science and future research directions," *Augmentative and alternative communication*, vol. 23, no. 3, pp. 204–16, 2007.

[8] I. Technology. [Online]. Available: http://www.inclusive.co.uk/

[9] Dynavox. [Online]. Available: http://www.dynavoxtech.com

[10] S. Millar and J. Scott, *What is Augmentative and Alternative Communication? An Introduction*. CALL Centre & Scottish Executive Education Dep, 1998, ch. 1, pp. 3–13.

[11] S. I. L. Technologies. [Online]. Available: http://www.spectronicsinoz.com

[12] "Tobii pceye." [Online]. Available: http://www.tobii.com/en/assistive-technology/global/products/hardware/pceye/

[13] D. J. Higginbotham, H. Shane, S. Russell, and K. Caves, "Acces to aac: Present, past and future," *Augmentative and Alternative Communication*, vol. 23, pp. 243 – 257, 2007.

[14] D. C. D. Sharon Glennen, *The Handbook of Augmentative and Alternative Communication*. Cengage Learning, 1997.

[15] L. T. Solutions. [Online]. Available: http://lowtechsolutions.org/

[16] C. Bionics. [Online]. Available: http://www.controlbionics.com/

[17] K. Tai, S. Blain, and T. Chau, "A review of emerging access technologies for individuals with severe motor impairments." *Assistive Technology*, vol. 20, no. 4, pp. 204 – 221, 2008.

[18] M. Betke, "Ambient intelligence and smart environments: A state of the art," in *Handbook of Ambient Intelligence and Smart Environments*, H. Nakashima, H. Aghajan, and J. Augusto, Eds. Springer US, 2010, pp. 3–31. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-93808-0_1

[19] Tobii, "An introduction to eye tracking and tobii eye trackers." [Online]. Available: http://www.tobii.com/en/eye-tracking-research/global/library/white-papers/tobii-eye-tracking-white-paper/

[20] "Dynavox eyemax." [Online]. Available: http://www.dynavoxtech.com/products/eyemax/

[21] "Eyetech tm4." [Online]. Available: http://www.eyetechaac.com/products/tm4

[22] "Headmouse extreme." [Online]. Available: http://www.orin.com/access/headmouse/

[23] "Tracker pro." [Online]. Available: http://www.madentec.com/products/tracker-pro.php

[24] "Smartnav 4." [Online]. Available: http://www.naturalpoint.com/smartnav/products/4-at/

[25] "Headmouse." [Online]. Available: http://robotica.udl.cat/

[26] "Cameramouse." [Online]. Available: http://www.cameramouse.org/index.html

[27] "Enable viacam." [Online]. Available: http://eviacam.sourceforge.net/

[28] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.

[29] K. Grauman, M. Betke, J. Lombardi, J. Gips, and G. Bradski, "Communication via eye blinks and eyebrow raises: video-based human-computer interfaces," *Universal Access in the Information Society*, vol. 2, no. 4, pp. 359–373, 2003. [Online]. Available: http://dx.doi.org/10.1007/s10209-003-0062-x

[30] M. Chau and M. Betke, "Real time eye tracking and blink detection with usb cameras," Department of Computer Science Technical Report, Boston University, Tech. Rep., 2005.

[31] D. O. Gorodnichy and G. Roth, "Nouse 'use your nose as a mouset' perceptual vision technology for hands-free games and interfaces," *Image and Vision Computing*, vol. 22, pp. 931–942, 2004.

[32] B. N. Waber, J. J. Magee, and M. Betke, "Fast head tilt detection for human-computer interaction." in *Proceedings of the ICCV Workshop on Human Computer Interaction*, 2005.

[33] T. Pallejà, E. Rubión, M. Tresanchez, and A. Fernández, "Using the optical flow to implement a relative virtual mouse controlled by head movements," *Universal Computer Science*, vol. 14, no. 19, pp. 3127–3141, 2008.

[34] J. J. Magee, M. Betke, J. Gips, M. R. Scott, and B. N. Waber, "A human-computer interface using symmetry between eyes to detect gaze direction," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, pp. 1248–1261, 2008.

[35] E. Perez, N. López, E. Orosco, C. Soria, V. Mut, and T. Freire-Bastos, "Robust human machine interface based on head movements applied to assistive robotics," *The Scientific World Journal*, vol. 2013, p. 11, 2013.

[36] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, pp. 137–154, 2004.

[37] L. R. Sapaico, H. Laga, and M. Nakajima, "The use of tongue protrusion gestures for video-based communication," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 2009, pp. 2017–2024.

[38] J. Varona, C. Manresa-Yee, and F. J. Perales, "Hands-free vision-based interface for computer accessibility," *J. Netw. Comput. Appl.*, vol. 31, pp. 357–374, 2008.

[39] T. Pallejà, A. Guillamet, M. Tresanchez, M. Teixidó, a. F. Viso, C. Rebate, and J. Palacín, "Implementation of a robust absolute virtual head mouse combining face detection, template matching and optical flow algorithms," *Telecommunication Systems*, 2011.

[40] G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly, 2008.

[41] L. Di Stefano and S. Mattoccia, "Fast template matching using bounded partial correlation," *Machine Vision and Applications*, vol. 13, no. 4, pp. 213–221, 2003.

[42] E. Perini, S. Soria, A. Prati, and R. Cucchiara, "Facemouse: A human-computer interface for tetraplegic people," in *Computer Vision in Human-Computer Interaction*. Springer, 2006, pp. 99–108.

[43] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Machine vision and applications*, vol. 12, no. 2, pp. 69–83, 2000.

[44] C. Fagiani, M. Betke, and J. Gips, "Evaluation of tracking methods for human-computer interaction," in *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, 2002.

[45] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003.

[46] ——, "Real-time tracking of non-rigid objects using mean shift," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2. IEEE, 2000, pp. 142–149.

[47] Z. Feng, N. Lu, and P. Jiang, "Posterior probability measure for image matching," *Pattern Recogn.*, vol. 41, pp. 2422–2433, 2008.

[48] G. R. Bradski and S. Clara, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal Q2*, 1998.

[49] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object tracking using camshift algorithm and multiple quantized feature spaces," in *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing*, ser. VIP '05. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 3–7. [Online]. Available: http://dl.acm.org/citation.cfm?id=1082121.1082122

[50] A. Salhi and A. Y. Jammoussi, "Object tracking system using camshift , meanshift and kalman filter," *World Academy of Science, Engineering & Technology*, vol. 64, p. 674, 2012.

[51] L. Campbell and A. Bobick, "Recognition of human body motion using phase space constraints," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, 1995, pp. 624–630.

[52] A. Hesami, F. Naghdy, D. Stirling, and H. Hill, "Perception of human gestures through observing body movements," in *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on*, 2008, pp. 97–102.

[53] S. Sclaroff, M. Betke, G. Kollios, J. Alon, V. Athitsos, R. Li, J. Magee, and T.-P. Tian, "Tracking, analysis, and recognition of human gestures in video," in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, Aug 2005, pp. 806–810 Vol. 2.

[54] M. Betke, O. Gusyatin, and M. Urinson, "Symbol design: a user-centered method to design pen-based interfaces and extend the functionality of pointer input devices," *Universal Access in the Information Society*, vol. 4, no. 3, pp. 223–236, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10209-005-0013-9

[55] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[56] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 568–574.

[57] F.-S. Chen, C.-M. Fu, and C.-L. Huang, "Hand gesture recognition using a real-time tracking method and hidden markov models," *Image and Vision Computing*, vol. 21, no. 8, pp. 745 – 758, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885603000702

[58] W. Harwin and R. Jackson, "Analysis of intentional head gestures to assist computer access by physically disabled people," *Journal of Biomedical Engineering*, vol. 12, no. 3, pp. 193 – 198, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/014154259090040T

[59] H.-I. Choi and P.-K. Rhee, "Head gesture recognition using {HMMs}," *Expert Systems with Applications*, vol. 17, no. 3, pp. 213 – 221, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417499000354

[60] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 9, pp. 1685–1699, Sept 2009.

[61] D. G. Stork, R. O. Duda, and P. E. Hart, *Pattern Classification*. Wiley-Interscience, 2000.

[62] A. Pentland and A. Lin, "Modeling and prediction of human behavior," *Neural Computation*, vol. 11, pp. 229–242, 1995.

[63] J. F.-S. Lin and D. Kulic, "Automatic human motion segmentation and identification using feature guided hmm for physical rehabilitation exercises," in *Robotics for Neurology and Rehabilitation, Workshop at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[64] J. Feng-Shun Lin and D. Kulic, "Segmenting human motion for automated rehabilitation exercise analysis." *Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2012, pp. 2881–4, 2012.

[65] A. Fod, M. J. Mataric, and O. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous robots*, vol. 12, pp. 39–54, 2002.

[66] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[67] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[68] G. Welch and G. Bishop, "An introduction to the kalman filter," SIGGRAPH 2001, 2001.

[69] P. Meyer, *Introductory probability and statistical applications*, ser. Addison-Wesley series in statistics. Addison-Wesley, 1965. [Online]. Available: http://books.google.com.br/books?id=-T7qkiDdpJMC

[70] "Tobii communicator." [Online]. Available: http://www.tobii.com/en/assistive-technology/global/products/software/tobii-communicator/

[71] "Opencv." [Online]. Available: http://opencv.org/

[72] "Cvhmm." [Online]. Available: http://sourceforge.net/projects/cvhmm/

[73] K. P. Murphy, "Hmm toolbox." [Online]. Available: http://people.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

[74] R. Adams, M. Victor, and A. Ropper, *Principles of Neurology*, 6th ed. New York: McGraw-Hill, 1997.

[75] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607–626, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2008.106

APÊNDICES

A. IMPLEMENTAÇÃO EM C++

A implementação dos sistema em C++ segue a estrutura descrita na figura A.1. Nesta estão identificadas as funções principais que são utilizadas no sistema.

O reconhecimento de face e o rastreamento são incorporados nas classes *Tracker* e *MeanShift*. Na primeira é implementado o classificador Viola-Jones como descrito na seção 3.1. Resultados positivos da função *FindFace* implicarão na criação do objeto *MeanShift*, responsável por rastrear a face.

Esta divisão foi feita com o objetivo de permitir a adição de novos algoritmos de rastreamento, simplesmente pela criação de novos objetos na classe *Tracker*.

Na classe *MeanShift* são implementadas as funções *GetHist*, responsável pelo cálculo dos histogramas, a função *UpdateTarget*, responsável pela atualização da posição da face, e a função *UpdateReference*, que atualiza a posição da referência.

A posição do alvo é então utilizada para primeiro segmentar o movimento, e depois, caso o sistema esteja no modo EXEC, é feita a classificação do movimento. Estas funcionalidades são implementadas pela classe *Classifier*.

Para segmentação com o uso da medida de ZVC, quatro funções que reconhecem eventos específicos da variável φ_k são utilizadas. Os eventos de detecção de vale, pico, passagem por limiar e de não movimento são encontrados pelas funções *ValleyDetect*, *PeakDetect*, *ThresholdCross* e *NoMotion* respectivamente.

Estas funções são utilizadas na função *GetSegments_3* que é responsável pelo armazenamento de todos os segmentos obtidos durante a execução do programa. Nesta função também são classificados os movimentos utilizando a função *decodePDF* da biblioteca *CvHMM* e a função *DistMahalanobis*. Para criar os arquivos *.txt* utilizados nos *scripts* em MATLAB, são utilizadas as funções *PlotSegments* e *PlotData*.

Criada a função *gaussMultv*, foi possível calcular o valor de *pdfs* multivariadas gaussianas, o que permitiu a criação da função *decodePDF*, ampliando as funcionalidades da biblioteca *CvHMM*.

A função *main* reúne todas as funcionalidades desenvolvidas, gerenciando a aquisição dos *frames* e realizando a integração do programa com outros. Isto é feito com a leitura de comandos de botão esquerdo do mouse, utilizando a função *mouseSniffer*, e repassando estes comandos para o SO criando eventos de teclado, função *eventoPrancha*.

Isto permite que na utilização de botões como interfaces com o computador, sejam registrados os momentos de acionamento do usuário sem interromper o uso da prancha de comunicação.

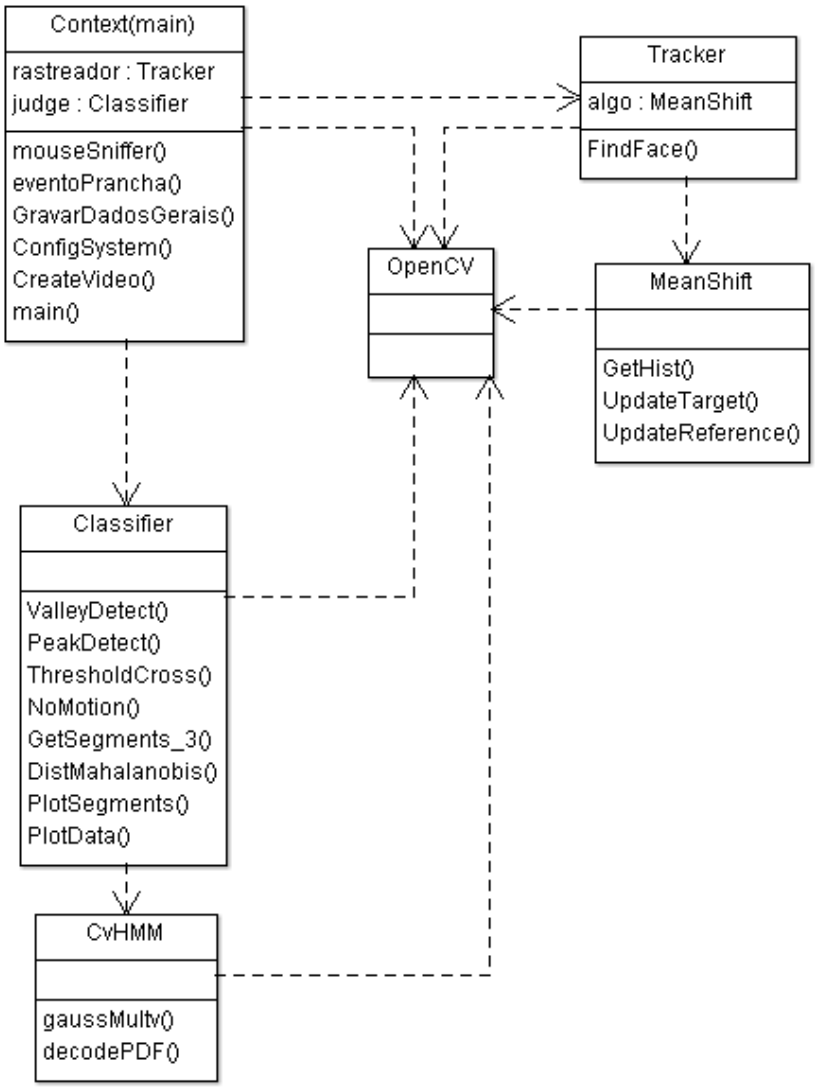


Figura A.1: Diagrama das classes que constroem a aplicação em C++.

A função *ConfigSystem* é responsável por definir os parâmetros de execução do programa a partir os argumentos passador por linha de comando. Caso seja necessário, a função *CreateVideo* é utilizada para registro em vídeo do sistema.

B.1 RESULTADOS PARA USUÁRIO 1

B.1.1 Identificação

Idade	33
Nível escolar	superior completo
Deficiência	não se aplica
Posição do botão	lado esquerdo
Frase escrita	"Boa tarde tchau"
Número de segmentos da amostra	86
Número de segmentos válidos	32

Tabela B.1: Dados do Usuário 1

B.1.2 Resultados para classificador HMM

Abaixo seguem os parâmetros das cadeias de Markov para os vetores de características ϕ^i, ϕ^{ii} e ϕ^{iii}.

Para o vetor de características ϕ^i temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0 \\ 0,0909 \\ 0 \\ 0,9091 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,0732 & 0,0417 & 0,8850 & 0 \\ 1,0000 & 0 & 0 & 0 \\ 0,2486 & 0 & 0,7514 & 0 \\ 0 & 0,3826 & 0 & 0,6174 \end{bmatrix} \tag{B.1}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3,9053 \\ 0,9943 \\ 281,9979 \\ 215,4342 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 17,1392 & 1,7330 & 9,0055 & -8,7789 \\ 1,7330 & 0,2992 & 0,5690 & -0,4209 \\ 9,0055 & 0,5690 & 16,6201 & -11,2486 \\ -8,7789 & -0,4209 & -11,2486 & 18,5134 \end{bmatrix} \tag{B.2}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 6,5652 \\ 0,8244 \\ 272,6980 \\ 211,0599 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 5,4597 & 0,5183 & 4,3743 & -6,3676 \\ 0,5183 & 0,1451 & 0,1169 & -0,4185 \\ 4,3743 & 0,1169 & 13,6868 & -9,1569 \\ -6,3676 & -0,4185 & -9,1569 & 20,7555 \end{bmatrix} \tag{B.3}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 5,4990 \\ 1,5609 \\ 291,7831 \\ 217,9659 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 4,8981 & 0,5789 & 4,1512 & -2,3892 \\ 0,5789 & 0,2867 & 0,0974 & 0,6901 \\ 4,1512 & 0,0974 & 14,2099 & -9,2069 \\ -2,3892 & 0,6901 & -9,2069 & 17,3036 \end{bmatrix} \tag{B.4}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 4,6733 \\ 0,1451 \\ 255,4257 \\ 207,2365 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1,4699 & 0,2393 & 4,9311 & 0,4845 \\ 0,2393 & 0,1222 & 1,0091 & 0,2775 \\ 4,9311 & 1,0091 & 62,5479 & 1,9328 \\ 0,4845 & 0,2775 & 1,9328 & 10,4940 \end{bmatrix} \tag{B.5}$$

Para o vetor de características $\boldsymbol{\phi}^{ii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,8985 \\ 0 \\ 0,1015 \\ 0 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,4157 & 0 & 0,5843 & 0 \\ 0 & 0,7361 & 0 & 0,2639 \\ 0 & 0 & 0,4137 & 0,5863 \\ 0 & 0,7687 & 0 & 0,2313 \end{bmatrix} \tag{B.6}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 251,5771 \\ 206,9196 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 42,1401 & -0,0502 \\ -0,0502 & 11,7842 \end{bmatrix} \tag{B.7}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 292,0069 \\ 217,9264 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 12,6410 & -8,6367 \\ -8,6367 & 16,9911 \end{bmatrix} \tag{B.8}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 268,3453 \\ 209,1864 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 35,5507 & 3,1104 \\ 3,1104 & 12,2970 \end{bmatrix} \tag{B.9}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 281,7021 \\ 215,7487 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 25,0202 & -13,2715 \\ -13,2715 & 20,5820 \end{bmatrix} \tag{B.10}$$

Para o vetor de características $\boldsymbol{\phi}^{iii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,9536 \\ 0 \\ 0,0464 \\ 0 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,1979 & 0,3987 & 0,4034 & 0 \\ 0 & 1,0000 & 0 & 0 \\ 0,4881 & 0,0010 & 0,0682 & 0,4428 \\ 0 & 0 & 0,2815 & 0,7185 \end{bmatrix} \tag{B.11}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3,5051 \\ 0,4724 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0,8317 & -0,1898 \\ -0,1898 & 0,4844 \end{bmatrix} \tag{B.12}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} -0,1998 \\ 0,6209 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1,3084 & 0,2163 \\ 0,2163 & 0,1888 \end{bmatrix} \tag{B.13}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 5,5984 \\ 0,9896 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0,3446 & -0,1046 \\ -0,1046 & 0,4762 \end{bmatrix} \tag{B.14}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 7,6814 \\ 1,4870 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0,5044 & -0,0047 \\ -0,0047 & 0,2838 \end{bmatrix} \tag{B.15}$$

B.1.3 Gráficos

Abaixo seguem os gráficos que representam a divisão de *clusters*, valores de LL e distância de Mahalanobis com treinamento com os dez primeiros segmentos válidos. Também está a análise por *3-fold cross-validation* com gráfico ROC.

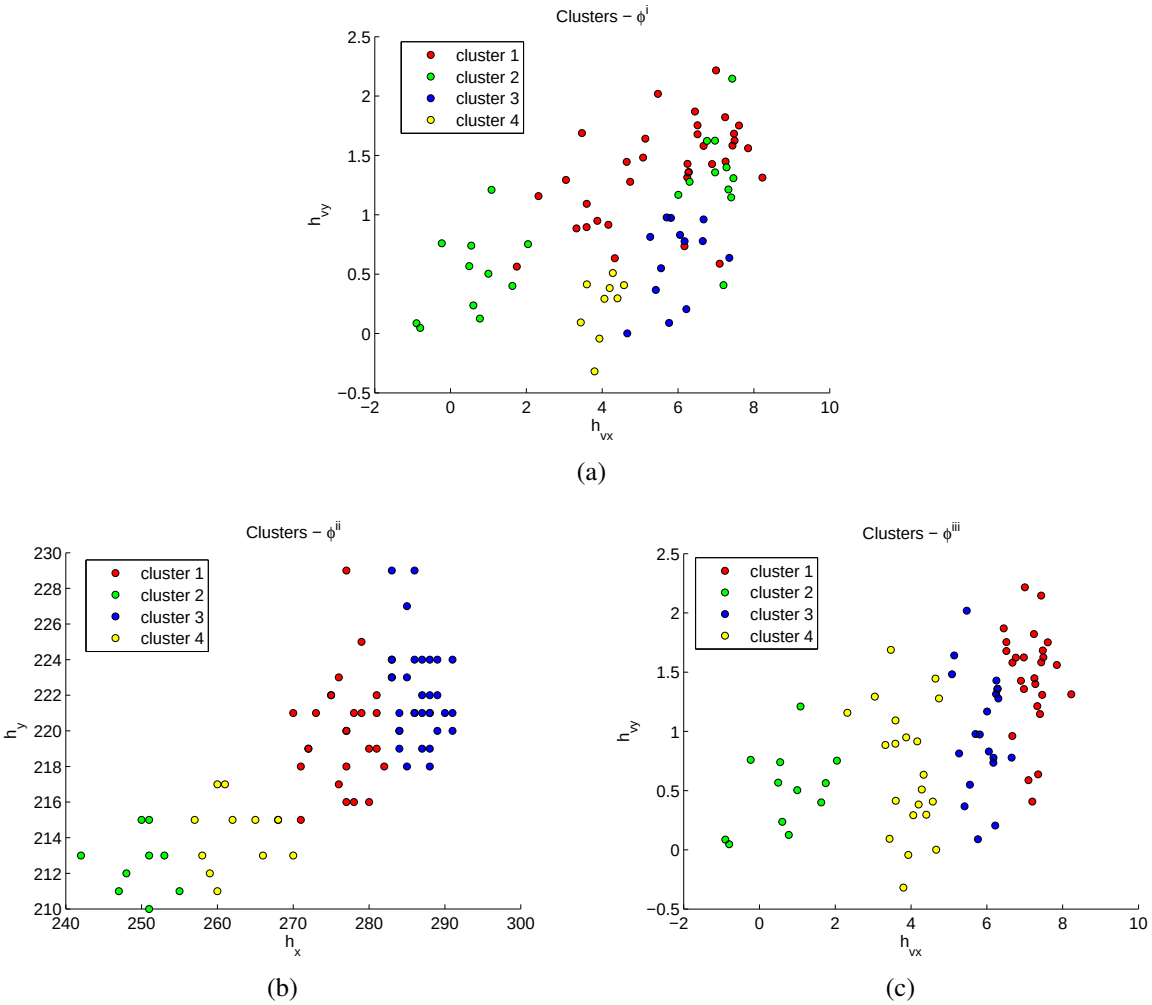


Figura B.1: *Clusters* obtidos com os dez primeiros segmentos válidos do usuário 1.

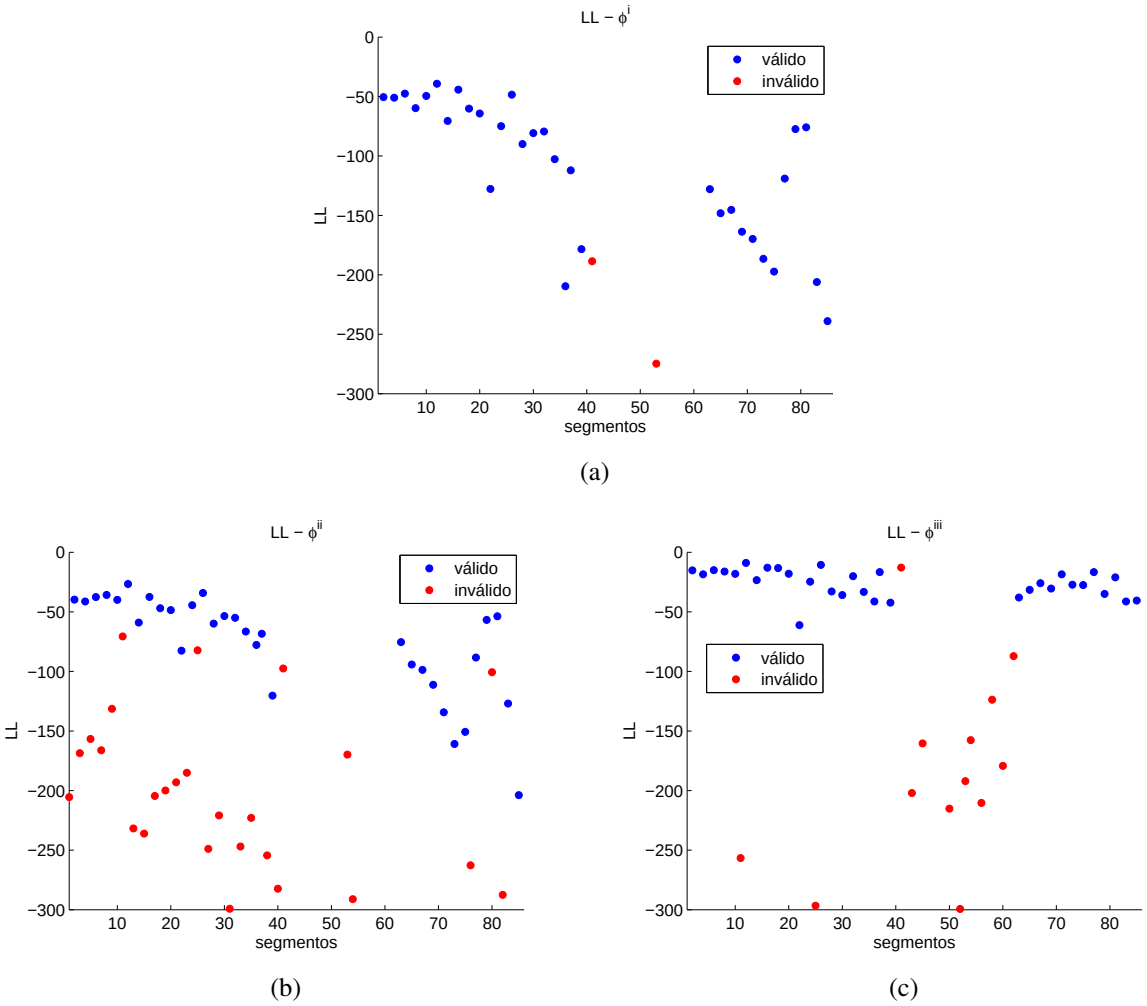


Figura B.2: Valores de LL obtidos com os segmentos de validação do usuário 1.

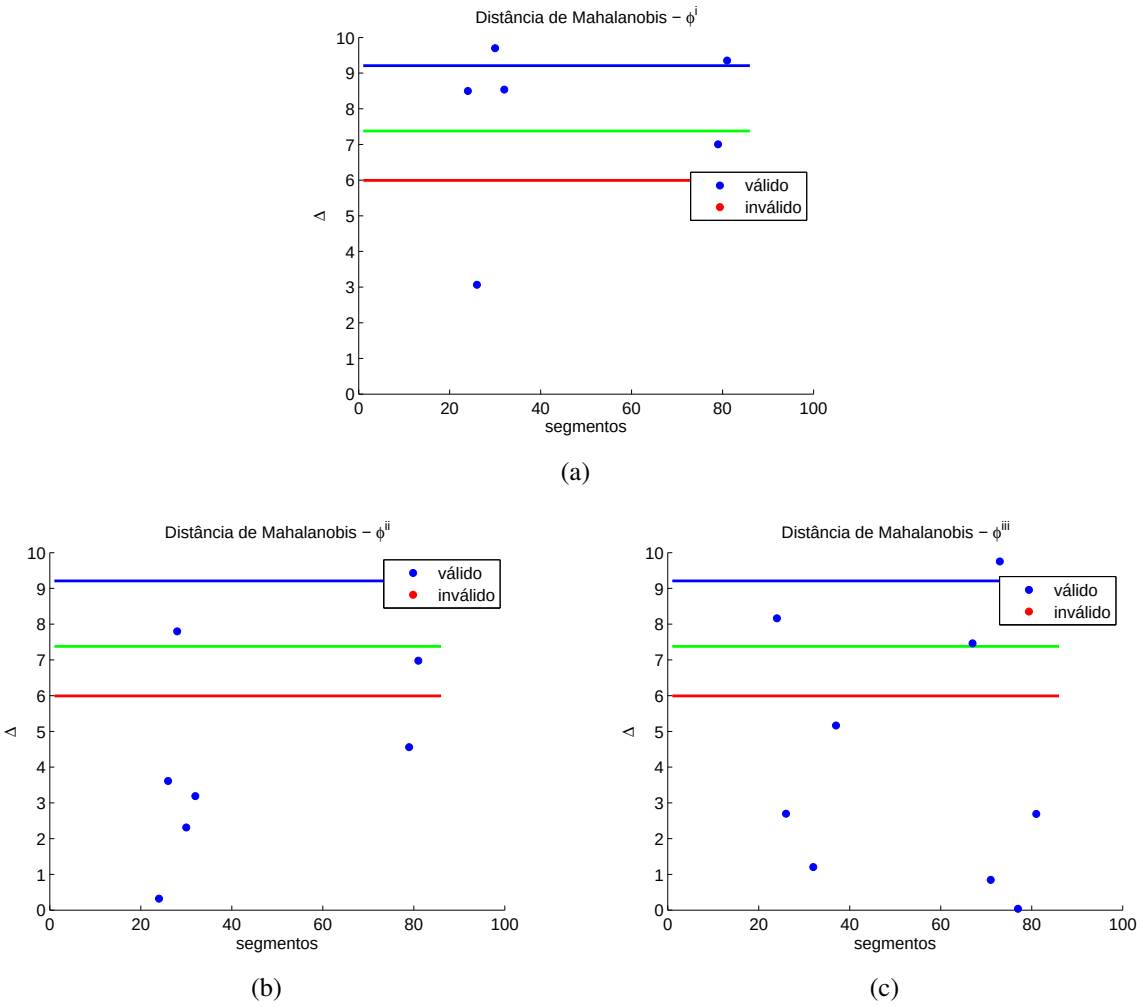


Figura B.3: Valores de Δ^2 obtidos com os segmentos de validação do usuário 1.

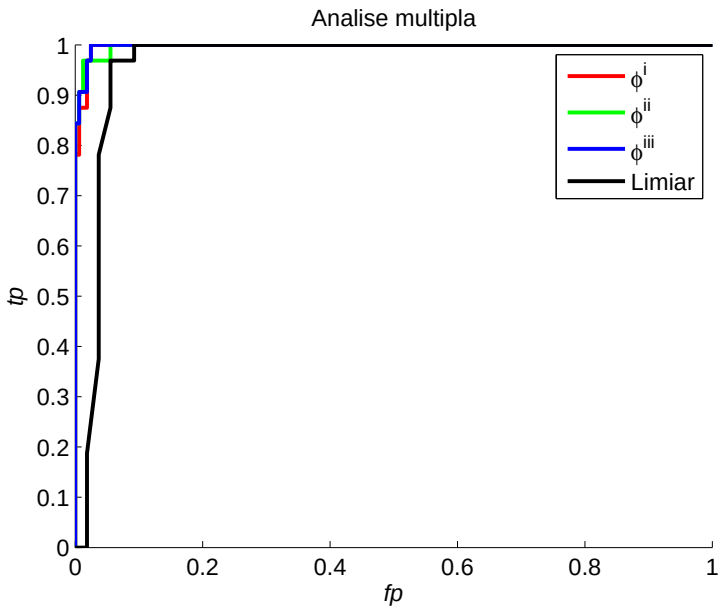


Figura B.4: Gráfico ROC comparando classificadores diferentes com a aplicação do *3-fold crossvalidation* para usuário 1.

B.2 RESULTADOS PARA USUÁRIO 2

B.2.1 Identificação

Idade	32
Nível escolar	superior completo
Deficiência	não se aplica
Posição do botão	lado direito
Frase escrita	"Boa tarde tchau"
Número de segmentos da amostra	108
Número de segmentos válidos	34

Tabela B.2: Dados do Usuário 2

B.2.2 Resultados para classificador HMM

Abaixo seguem os parâmetros das cadeias de Markov para os vetores de características $\boldsymbol{\phi}^{i}$, $\boldsymbol{\phi}^{ii}$ e $\boldsymbol{\phi}^{iii}$.

Para o vetor de características $\boldsymbol{\phi}^{i}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,0909 \\ 0,0455 \\ 0,8636 \\ 0 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 1,0000 & 0 & 0 & 0 \\ 0 & 1,0000 & 0 & 0 \\ 0 & 0 & 0,6073 & 0,3927 \\ 0 & 0,2051 & 0 & 0,7949 \end{bmatrix} \tag{B.16}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -3,8062 \\ 2,1891 \\ 239,1111 \\ 226,5000 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 2,937 & -1,088 & 6,055 & -2,730 \\ -1,088 & 0,725 & 5,336 & -1,940 \\ 6,055 & 5,336 & 247,998 & -101,333 \\ -2,730 & -1,940 & -101,333 & 42,038 \end{bmatrix} \tag{B.17}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} -4,0433 \\ 1,5972 \\ 176,5954 \\ 218,7235 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 5,5011 & -1,5999 & 5,6494 & -0,9084 \\ -1,5999 & 0,5323 & -1,9018 & 0,7604 \\ 5,6494 & -1,9018 & 29,7957 & -5,2253 \\ -0,9084 & 0,7604 & -5,2253 & 6,3114 \end{bmatrix} \tag{B.18}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} -4,1190 \\ 1,1806 \\ 204,5072 \\ 208,8429 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0,9168 & -0,3069 & 4,2281 & -1,4976 \\ -0,3069 & 0,1754 & -1,7956 & 0,7971 \\ 4,2281 & -1,7956 & 47,0749 & -14,5610 \\ -1,4976 & 0,7971 & -14,5610 & 6,2187 \end{bmatrix} \tag{B.19}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} -3,9734 \\ 1,3487 \\ 189,8328 \\ 214,4428 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 3,8451 & -1,0444 & 3,8086 & 0,2404 \\ -1,0444 & 0,3872 & -1,4221 & 0,4591 \\ 3,8086 & -1,4221 & 18,2587 & -4,6736 \\ 0,2404 & 0,4591 & -4,6736 & 4,7008 \end{bmatrix} \tag{B.20}$$

Para o vetor de características $\boldsymbol{\phi}^{ii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,8636 \\ 0 \\ 0,0909 \\ 0,0455 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,5032 & 0,4968 & 0 & 0 \\ 0 & 0,8095 & 00,1905 & \\ 0 & 0 & 1,0000 & 0 \\ 0 & 0,0008 & 0 & 0,9992 \end{bmatrix} \tag{B.21}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 206,6746 \\ 207,9872 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 34,3003 & -9,0557 \\ -9,0557 & 3,8771 \end{bmatrix} \tag{B.22}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 191,4027 \\ 213,9331 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 18,6888 & -5,3680 \\ -5,3680 & 4,7094 \end{bmatrix} \tag{B.23}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 239,1111 \\ 226,5000 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 247,9977 & -101,3333 \\ -101,3333 & 42,0378 \end{bmatrix} \tag{B.24}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 177,1721 \\ 218,4923 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 31,0766 & -6,1306 \\ -6,1306 & 6,6684 \end{bmatrix} \tag{B.25}$$

Para o vetor de características $\boldsymbol{\phi}^{iii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,0630 \\ 0 \\ 0,9239 \\ 0,0132 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,5496 & 0,0008 & 0,3062 & 0,1435 \\ 0 & 1,0000 & 0 & 0 \\ 0,3755 & 0,4249 & 0,1601 & 0,0394 \\ 0,1973 & 0,0232 & 0,0581 & 0,7215 \end{bmatrix} \tag{B.26}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -4,7849 \\ 1,5980 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0,4001 & -0,1094 \\ -0,1094 & 0,1242 \end{bmatrix} \tag{B.27}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} -1,0448 \\ 0,6107 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0,6165 & -0,1418 \\ -0,1418 & 0,1082 \end{bmatrix} \tag{B.28}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} -3,1342 \\ 1,0548 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0,3421 & 0,0657 \\ 0,0657 & 0,1483 \end{bmatrix} \tag{B.29}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} -6,0528 \\ 2,3669 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1,7791 & -0,0321 \\ -0,0321 & 0,1891 \end{bmatrix} \tag{B.30}$$

B.2.3 Gráficos

Abaixo seguem os gráficos que representam a divisão de *clusters*, valores de LL e distância de Mahalanobis com treinamento com os dez primeiros segmentos válidos. Também está a análise por *3-fold cross-validation* com gráfico ROC.

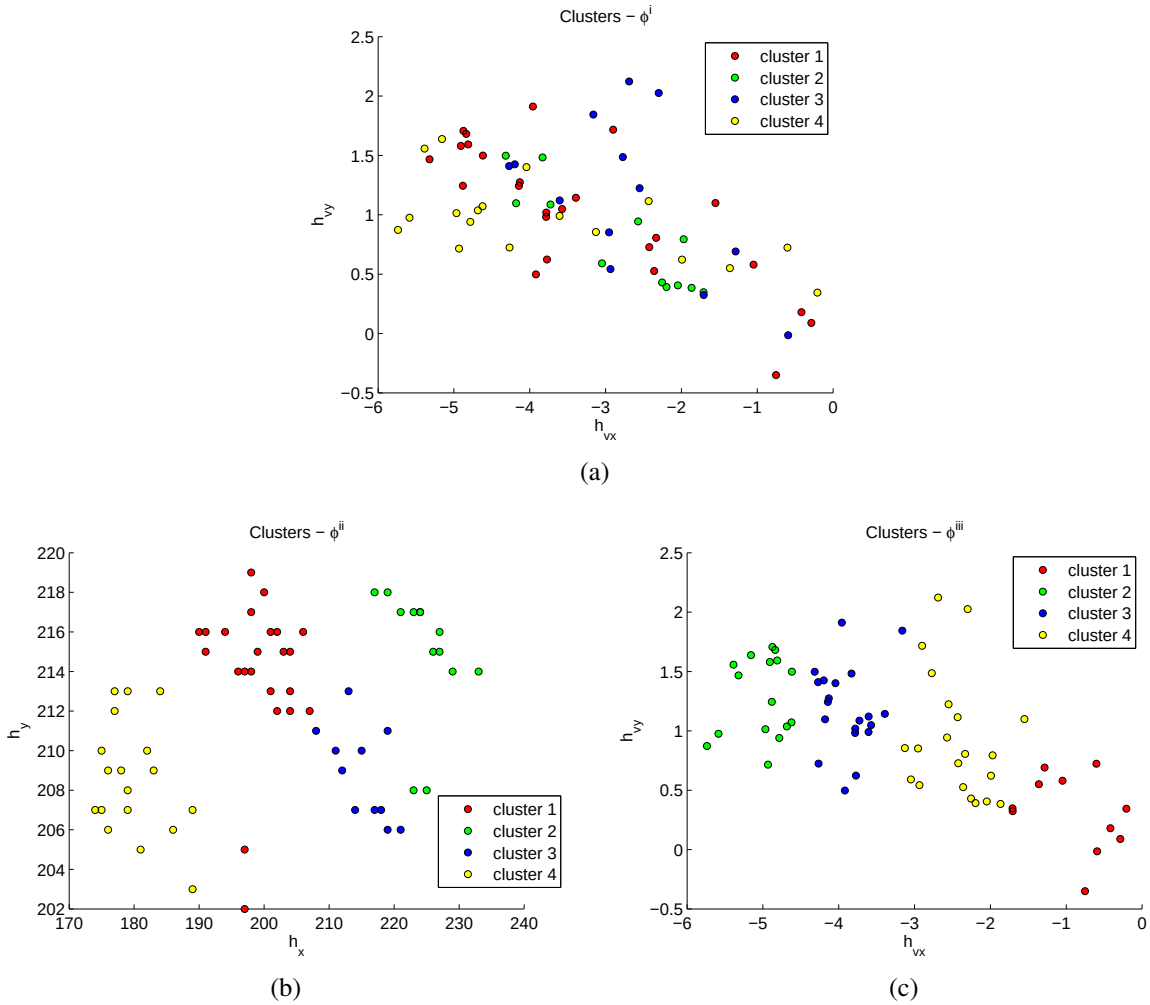


Figura B.5: *Clusters* obtidos com os dez primeiros segmentos válidos do usuário 2.

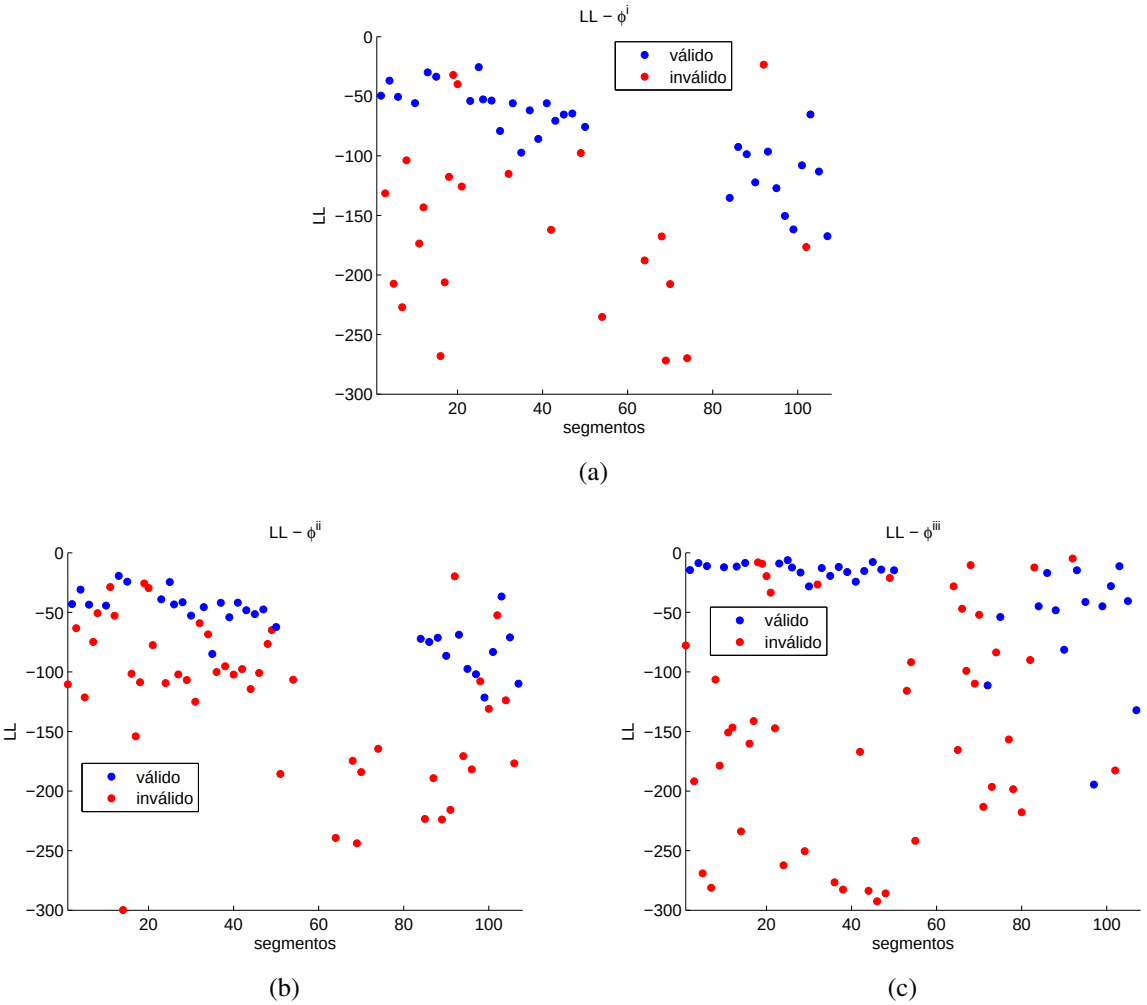


Figura B.6: Valores de LL obtidos com os segmentos de validação do usuário 2.

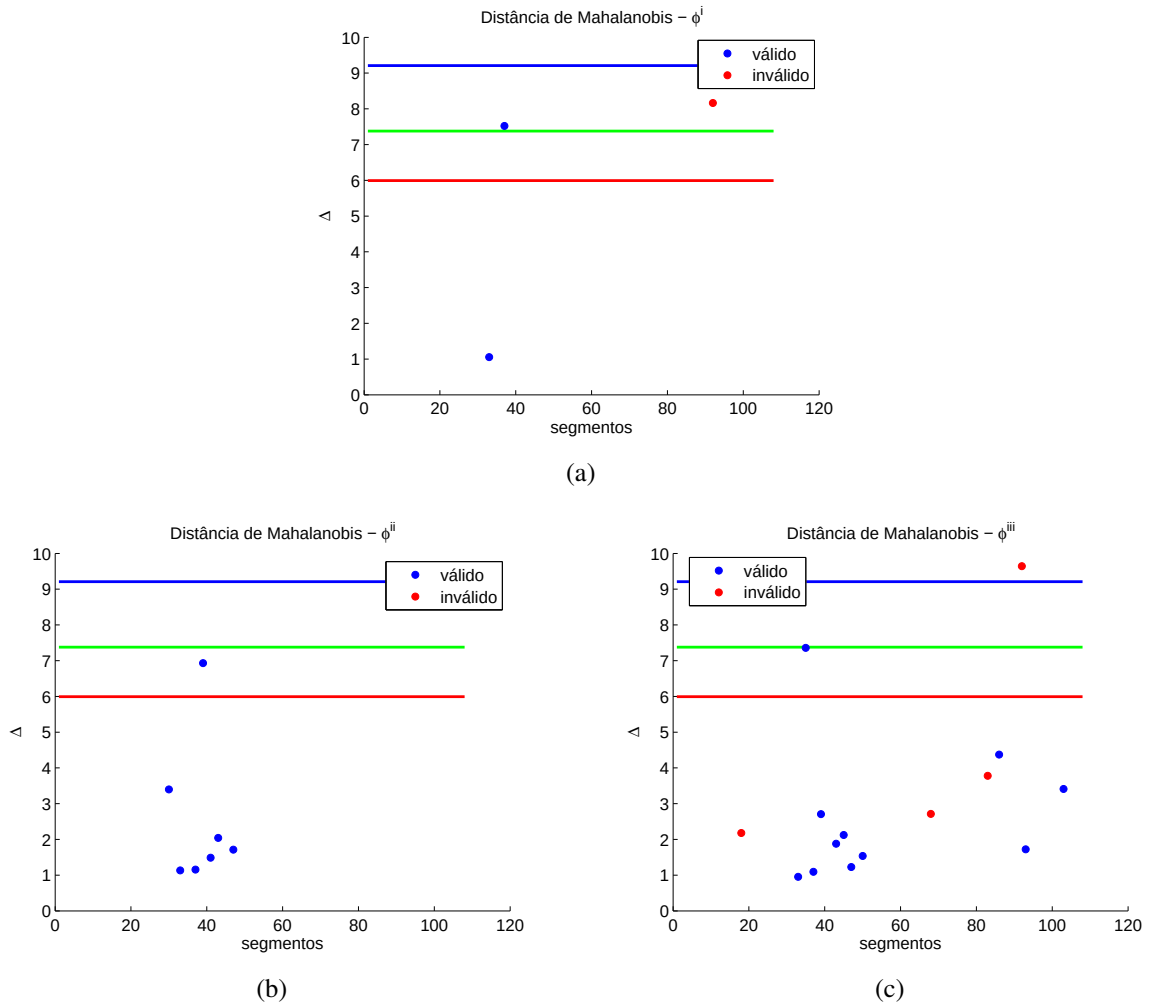


Figura B.7: Valores de Δ^2 obtidos com os segmentos de validação usuário 2.

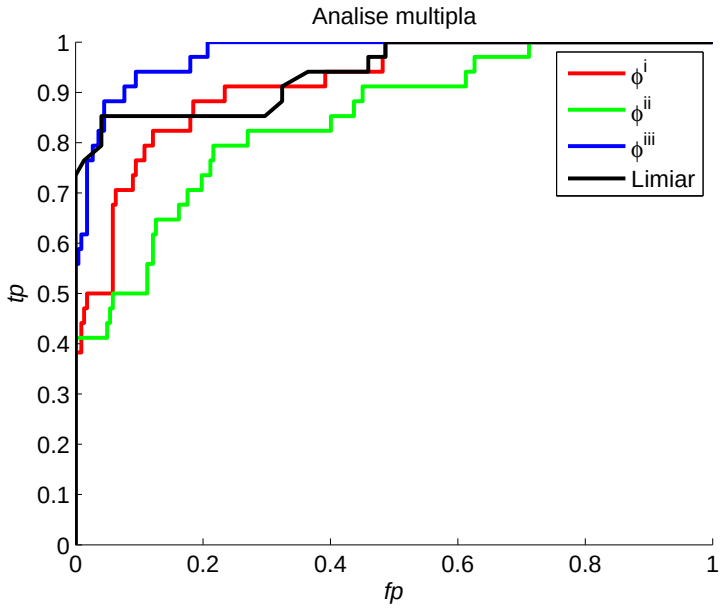


Figura B.8: Gráfico ROC comparando classificadores diferentes com a aplicação do *3-fold crossvalidation* para usuário 2.

B.3 RESULTADOS PARA USUÁRIO 3

B.3.1 Identificação

Idade	29
Nível escolar	superior completo
Deficiência	não se aplica
Posição do botão	lado direito
Frase escrita	"Boa tarde tchau"
Número de segmentos da amostra	117
Número de segmentos válidos	35

Tabela B.3: Dados do Usuário 3

B.3.2 Resultados para classificador HMM

Abaixo seguem os parâmetros das cadeias de Markov para os vetores de características ϕ^i, ϕ^{ii} e ϕ^{iii}.

Para o vetor de características ϕ^i temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,2000 \\ 0,3965 \\ 0,4035 \\ 0 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 1,0000 & 0 & 0 & 0 \\ 0 & 0,5099 & 0,4901 & 0 \\ 0 & 0 & 0,4941 & 0,5059 \\ 0 & 0 & 0 & 1,0000 \end{bmatrix} \tag{B.31}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -2,4804 \\ 0,8901 \\ 178,9000 \\ 205,2000 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1,7349 & -0,3257 & 1,1189 & 0,9228 \\ -0,3257 & 0,0889 & -0,1514 & -0,0412 \\ 1,1189 & -0,1514 & 8,1000 & 0,0200 \\ 0,9228 & -0,0412 & 0,0200 & 2,3700 \end{bmatrix} \tag{B.32}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} -3,2095 \\ 1,3545 \\ 194,1889 \\ 208,4315 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0,2137 & -0,1411 & 1,7356 & -1,0153 \\ -0,1411 & 0,2055 & -1,5036 & 0,9564 \\ 1,7356 & -1,5036 & 18,2411 & -11,0695 \\ -1,0153 & 0,9564 & -11,0695 & 6,8667 \end{bmatrix} \tag{B.33}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} -3,3872 \\ 1,5085 \\ 186,7900 \\ 212,7822 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0,2812 & -0,0580 & 1,0327 & 0,0761 \\ -0,0580 & 0,1646 & -0,7077 & 0,2497 \\ 1,0327 & -0,7077 & 9,3774 & -3,1180 \\ 0,0761 & 0,2497 & -3,1180 & 5,2902 \end{bmatrix} \tag{B.34}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} -2,2562 \\ 1,2005 \\ 181,7892 \\ 215,6381 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1,0523 & -0,3875 & 1,1088 & 0,2773 \\ -0,3875 & 0,2091 & -0,4995 & 0,0535 \\ 1,1088 & -0,4995 & 3,2799 & 0,5828 \\ 0,2773 & 0,0535 & 0,5828 & 2,4350 \end{bmatrix} \tag{B.35}$$

Para o vetor de características $\boldsymbol{\phi}^{ii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0 \\ 0,2000 \\ 0,4025 \\ 0,3975 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 1,0000 & 0 & 0 & 0 \\ 0 & 1,0000 & 0 & 0 \\ 0,6540 & 0 & 0,3460 & 0 \\ 0 & 0 & 0,6010 & 0,3990 \end{bmatrix} \tag{B.36}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 182,0979 \\ 215,5275 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 3,7729 & 0,3399 \\ 0,3399 & 2,4492 \end{bmatrix} \tag{B.37}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 178,9000 \\ 205,2000 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 8,1000 & 0,0200 \\ 0,0200 & 2,3700 \end{bmatrix} \tag{B.38}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 188,1532 \\ 211,8513 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 6,4728 & -0,7968 \\ -0,7968 & 3,5481 \end{bmatrix} \tag{B.39}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 195,5017 \\ 207,5864 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 12,0502 & -6,9660 \\ -6,9660 & 4,1605 \end{bmatrix} \tag{B.40}$$

Para o vetor de características $\boldsymbol{\phi}^{iii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0 \\ 0,8271 \\ 0,1729 \\ 0 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 1,0000 \\ 0,4610 & 0,2732 & 0,2658 & 0 \\ 0 & 0,3479 & 0,6521 & 0 \\ 0 & 0 & 0 & 1,0000 \end{bmatrix} \tag{B.41}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -1,9502 \\ 1,0506 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0,0900 & -0,0171 \\ -0,0171 & 0,0616 \end{bmatrix} \tag{B.42}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} -2,8540 \\ 1,3071 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0,0819 & 0,0048 \\ 0,0048 & 0,1231 \end{bmatrix} \tag{B.43}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} -3,7561 \\ 1,5881 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0,1169 & 0,0139 \\ 0,0139 & 0,1389 \end{bmatrix} \tag{B.44}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} -0,9183 \\ 0,6283 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0,1544 & -0,0567 \\ -0,0567 & 0,0553 \end{bmatrix} \tag{B.45}$$

B.3.3 Gráficos

Abaixo seguem os gráficos que representam a divisão de *clusters*, valores de LL e distância de Mahalanobis com treinamento com os dez primeiros segmentos válidos. Também está a análise por *3-fold cross-validation* com gráfico ROC.

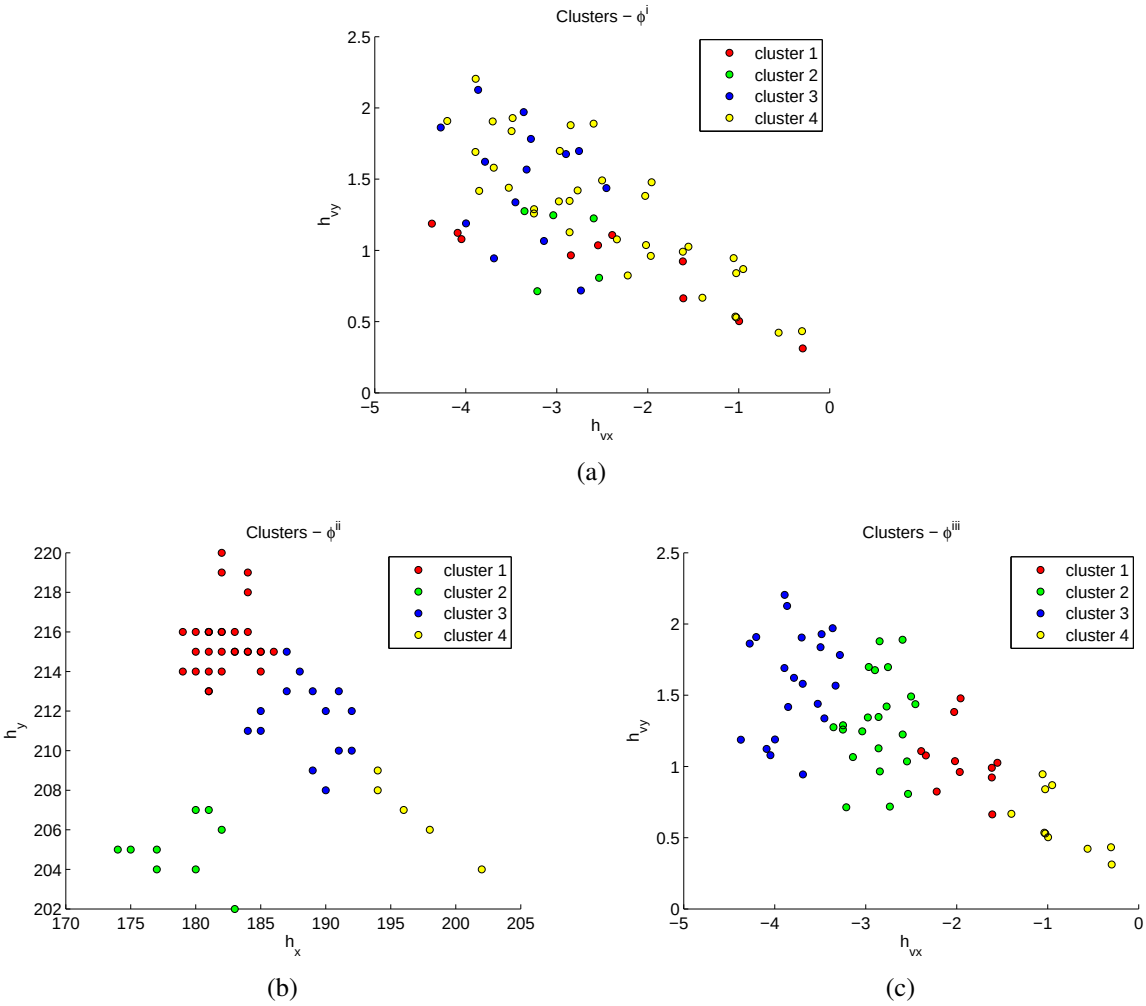


Figura B.9: *Clusters* obtidos com os dez primeiros segmentos válidos do usuário 3.

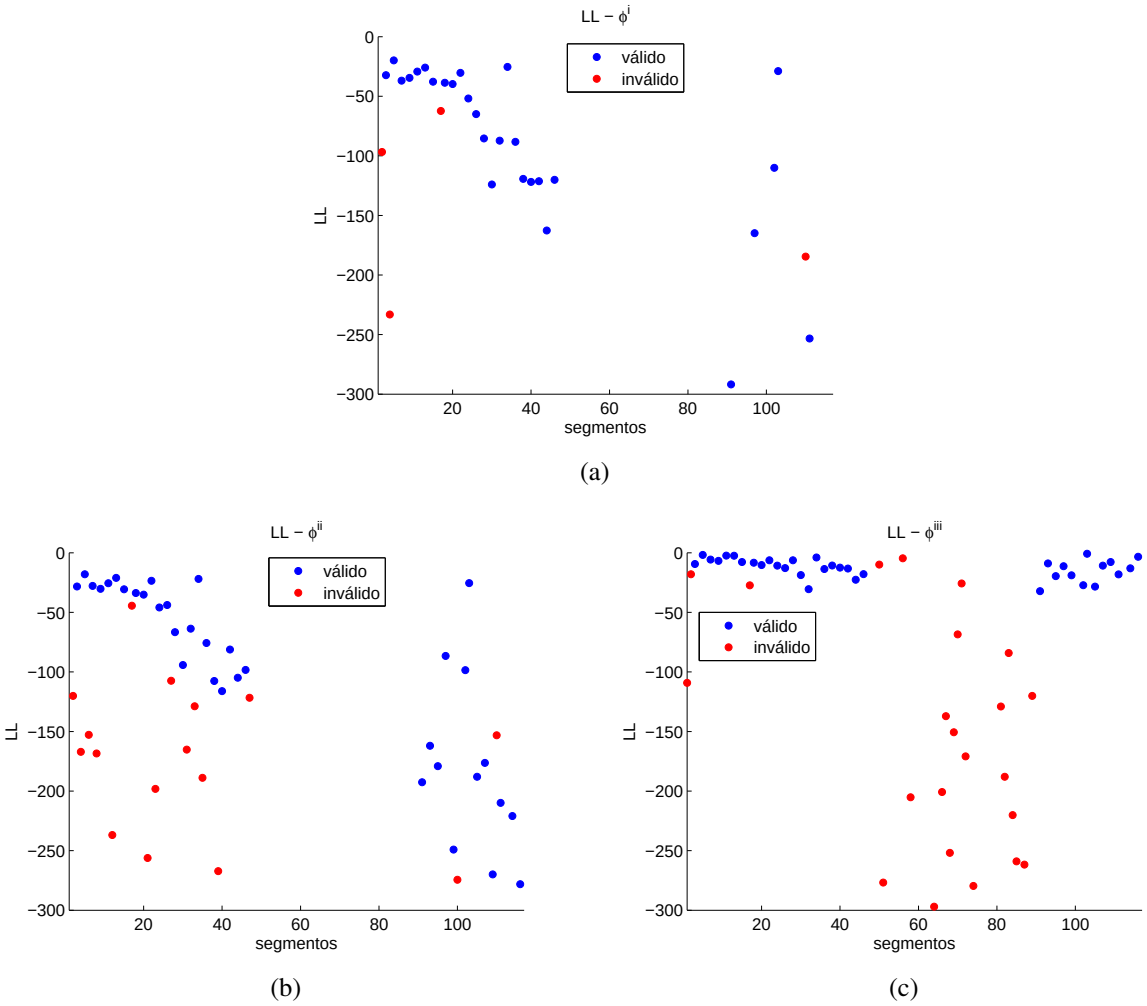


Figura B.10: Valores de LL obtidos com os segmentos de validação do usuário 3.

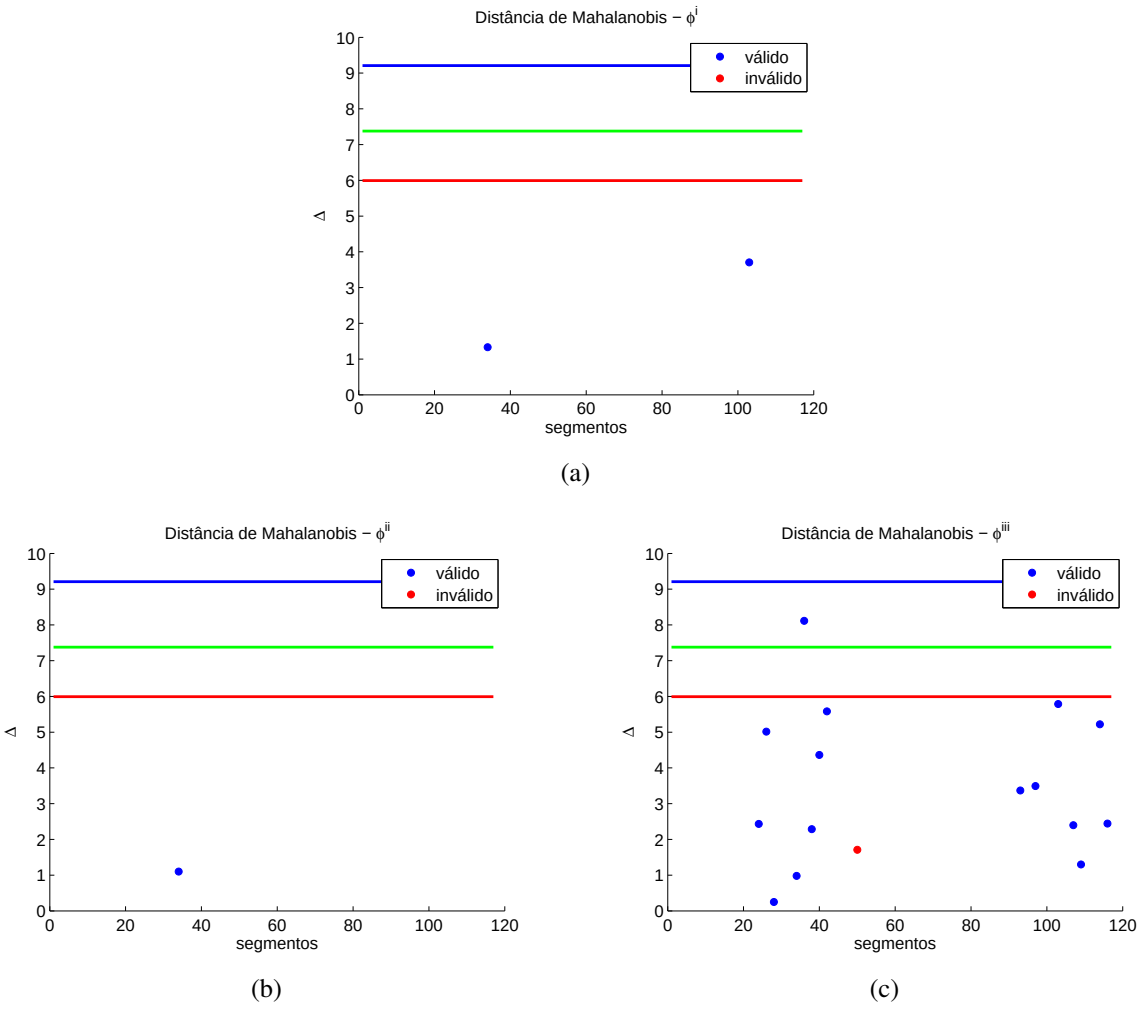


Figura B.11: Valores de Δ^2 obtidos com os segmentos de validação usuário 3.

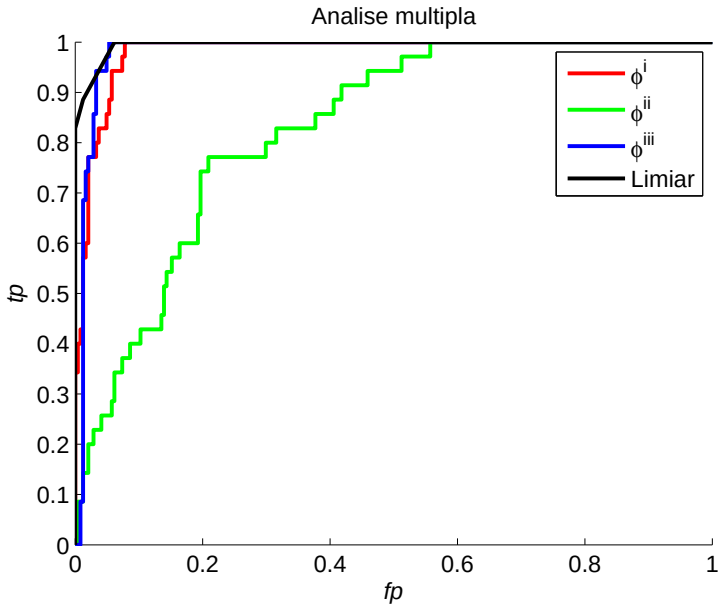


Figura B.12: Gráfico ROC comparando classificadores diferentes com a aplicação do *3-fold crossvalidation* para usuário 3.

B.4 RESULTADOS PARA USUÁRIO 4

B.4.1 Identificação

Idade	23
Nível escolar	superior incompleto
Deficiência	não se aplica
Posição do botão	lado esquerdo
Frase escrita	"Boa tarde tchau"
Número de segmentos da amostra	272
Número de segmentos válidos	40

Tabela B.4: Dados do Usuário 4

B.4.2 Resultados para classificador HMM

Abaixo seguem os parâmetros das cadeias de Markov para os vetores de características $\boldsymbol{\phi}^{i}$, $\boldsymbol{\phi}^{ii}$ e $\boldsymbol{\phi}^{iii}$.

Para o vetor de características $\boldsymbol{\phi}^{i}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,1000 \\ 0,4000 \\ 0 \\ 0,5000 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 1,0000 & 0 & 0 & 0 \\ 0 & 1,0000 & 0 & 0 \\ 0 & 0 & 1,0000 & 0 \\ 0 & 0 & 0,3987 & 0,6013 \end{bmatrix} \tag{B.46}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3,8072 \\ 1,5028 \\ 365,0000 \\ 207,3333 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 2,5095 & 0,8375 & 2,3283 & 0,2047 \\ 0,8375 & 0,3317 & 2,0137 & 0,6822 \\ 2,3283 & 2,0137 & 39,3433 & 18,6667 \\ 0,2047 & 0,6822 & 18,6667 & 9,2322 \end{bmatrix} \tag{B.47}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 3,3830 \\ 1,1530 \\ 336,8261 \\ 230,0435 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 2,4141 & 0,7848 & 3,3731 & 1,1838 \\ 0,7848 & 0,5515 & 0,0020 & 1,1853 \\ 3,3731 & 0,0020 & 48,1537 & 12,1815 \\ 1,1838 & 1,1853 & 12,1815 & 14,7472 \end{bmatrix} \tag{B.48}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 2,9120 \\ 1,0835 \\ 322,3204 \\ 210,8573 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 2,0828 & 0,5128 & 1,8904 & 0,2473 \\ 0,5128 & 0,1612 & 0,5498 & 0,1274 \\ 1,8904 & 0,5498 & 9,2426 & 4,9003 \\ 0,2473 & 0,1274 & 4,9003 & 3,7979 \end{bmatrix} \tag{B.49}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 2,7563 \\ 0,5295 \\ 313,3163 \\ 204,3950 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0,6806 & 0,3312 & -0,2232 & 1,6137 \\ 0,3312 & 0,2488 & 0,0127 & 0,9939 \\ -0,2232 & 0,0127 & 16,2734 & 4,3156 \\ 1,6137 & 0,9939 & 4,3156 & 8,0535 \end{bmatrix} \tag{B.50}$$

Para o vetor de características $\boldsymbol{\phi}^{ii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0 \\ 0,4000 \\ 0,5000 \\ 0,1000 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 1,0000 & 0 & 0 & 0 \\ 0 & 1,0000 & 0 & 0 \\ 0,3095 & 0 & 0,6905 & 0 \\ 0 & 0 & 0 & 1,0000 \end{bmatrix} \tag{B.51}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 322,6473 \\ 211,0978 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 8,1675 & 3,9590 \\ 3,9590 & 3,1089 \end{bmatrix} \tag{B.52}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 336,8261 \\ 230,0435 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 48,1537 & 12,1815 \\ 12,1815 & 14,7472 \end{bmatrix} \tag{B.53}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 313,6492 \\ 204,6270 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 16,2906 & 4,9118 \\ 4,9118 & 8,0809 \end{bmatrix} \tag{B.54}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 365,0000 \\ 207,3333 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 39,3433 & 18,6667 \\ 18,6667 & 9,2322 \end{bmatrix} \tag{B.55}$$

Para o vetor de características $\boldsymbol{\phi}^{iii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,8885 \\ 0 \\ 0,1115 \\ 0 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,0822 & 0,4442 & 0,4735 & 0 \\ 0,3467 & 0,5052 & 0,0234 & 0,1247 \\ 0 & 0 & 1,0000 & 0 \\ 0,4017 & 0 & 0 & 0,5983 \end{bmatrix} \tag{B.56}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3,0274 \\ 1,0532 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0,1841 & -0,0189 \\ -0,0189 & 0,1860 \end{bmatrix} \tag{B.57}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 4,3386 \\ 1,3770 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0,1829 & -0,0196 \\ -0,0196 & 0,1611 \end{bmatrix} \tag{B.58}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 1,4199 \\ 0,3420 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0,5495 & -0,0216 \\ -0,0216 & 0,1159 \end{bmatrix} \tag{B.59}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 5,5326 \\ 2,2022 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0,3187 & 0,0799 \\ 0,0799 & 0,0377 \end{bmatrix} \tag{B.60}$$

B.4.3 Gráficos

Abaixo seguem os gráficos que representam a divisão de *clusters*, valores de LL e distância de Mahalanobis com treinamento com os dez primeiros segmentos válidos. Também está a análise por *3-fold cross-validation* com gráfico ROC.

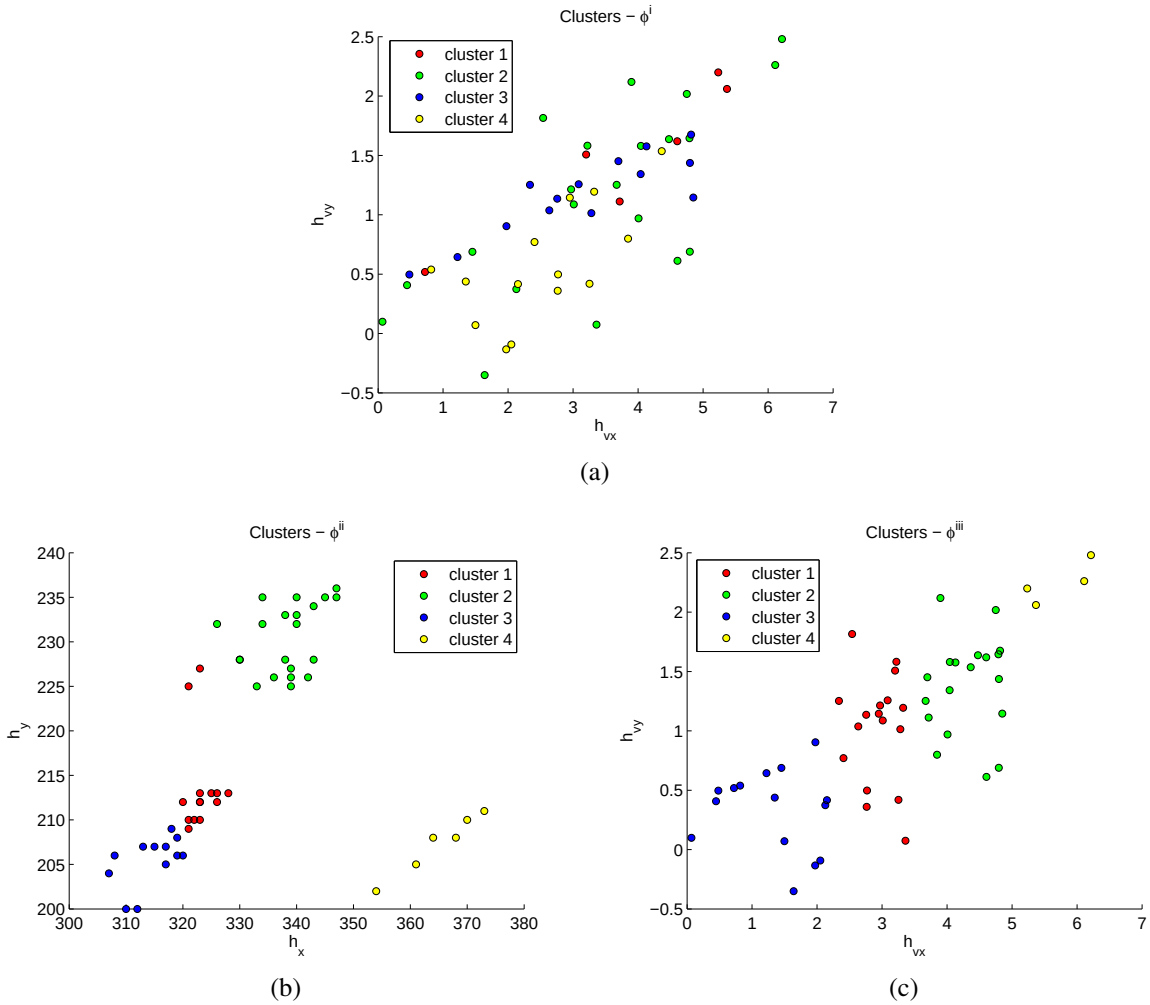


Figura B.13: *Clusters* obtidos com os dez primeiros segmentos válidos do usuário 4.

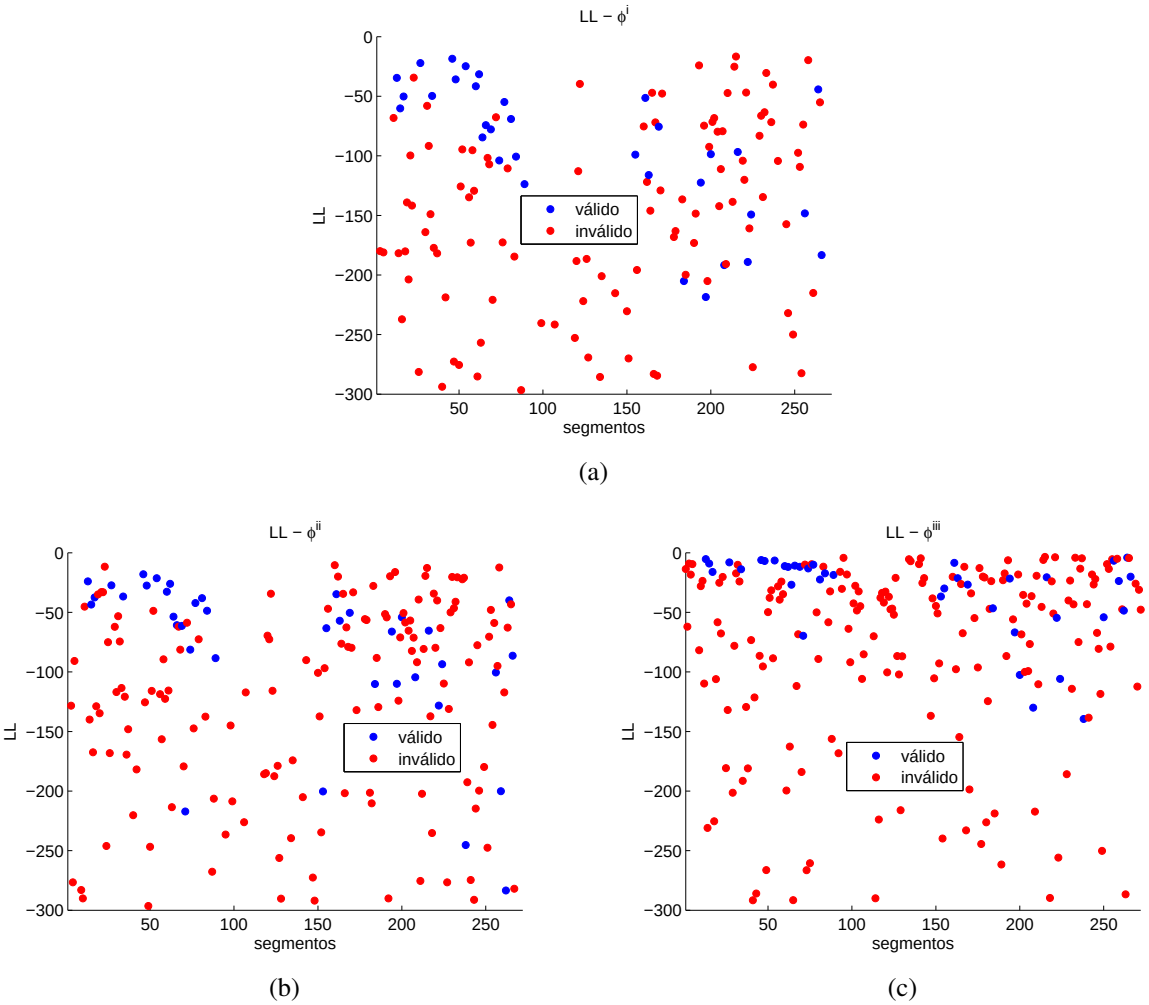


Figura B.14: Valores de LL obtidos com os segmentos de validação do usuário 4.

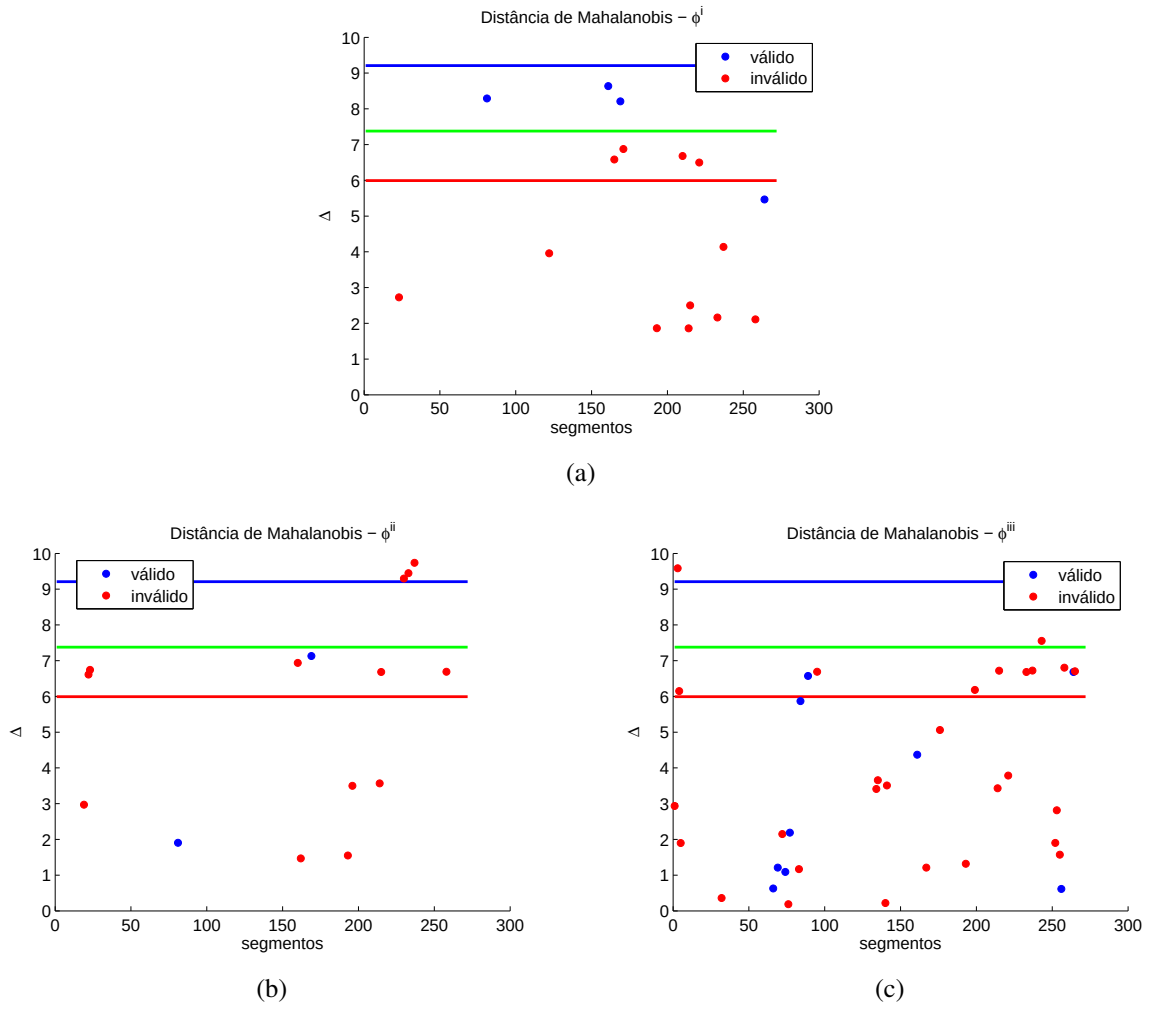


Figura B.15: Valores de Δ^2 obtidos com os segmentos de validação do usuário 4.

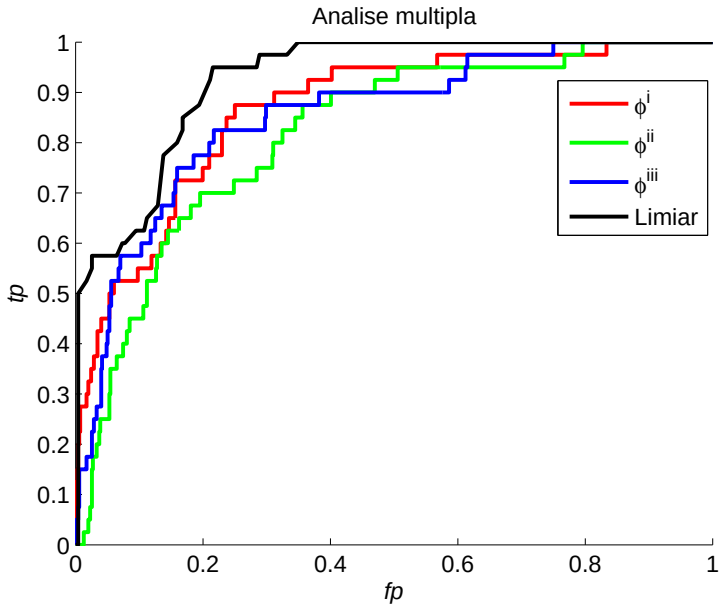


Figura B.16: Gráfico ROC comparando classificadores diferentes com a aplicação do *3-fold crossvalidation* para usuário 4.

B.5 RESULTADOS PARA USUÁRIO 5

B.5.1 Identificação

Idade	33
Nível escolar	superior incompleto
Deficiência	paralisia cerebral
Posição do botão	lado esquerdo
Frase escrita	"Boa tarde"
Número de segmentos da amostra	226
Número de segmentos válidos	34

Tabela B.5: Dados do Usuário 5

B.5.2 Resultados para classificador HMM

Abaixo seguem os parâmetros das cadeias de Markov para os vetores de características $\boldsymbol{\phi}^i$, $\boldsymbol{\phi}^{ii}$ e $\boldsymbol{\phi}^{iii}$.

Para o vetor de características $\boldsymbol{\phi}^i$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,1999 \\ 0,4001 \\ 0,4000 \\ 0 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 1,0000 & 0 & 0 & 0 \\ 0 & 1,0000 & 0 & 0 \\ 0 & 0 & 0,7369 & 0,2631 \\ 0 & 0 & 0,1356 & 0,8644 \end{bmatrix} \tag{B.61}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3,6369 \\ 1,5831 \\ 290,0990 \\ 238,1011 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 5,9429 & 0,8322 & -5,0194 & 4,0485 \\ 0,8322 & 0,2708 & -0,0122 & 1,7855 \\ -5,0194 & -0,0122 & 75,3070 & 6,8974 \\ 4,0485 & 1,7855 & 6,8974 & 28,6981 \end{bmatrix} \tag{B.62}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 2,5489 \\ 0,3952 \\ 302,2100 \\ 219,2117 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1,243 & -0,403 & -0,318 & 3,410 \\ -0,403 & 0,678 & 2,076 & -0,838 \\ -0,318 & 2,076 & 50,386 & -18,839 \\ 3,410 & -0,838 & -18,839 & 44,819 \end{bmatrix} \tag{B.63}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 2,4364 \\ -2,4652 \\ 294,7984 \\ 198,6995 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 13,2197 & 1,0420 & 21,9536 & 26,8461 \\ 1,0420 & 1,5913 & -2,3978 & 5,0845 \\ 21,9536 & -2,3978 & 83,8487 & 8,9333 \\ 26,8461 & 5,0845 & 8,9333 & 86,8901 \end{bmatrix} \tag{B.64}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 3,7960 \\ -2,4870 \\ 313,2619 \\ 192,9324 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 5,2016 & -0,8454 & 8,3483 & 5,8672 \\ -0,8454 & 0,6652 & -1,6983 & 0,1186 \\ 8,3483 & -1,6983 & 26,8260 & 11,1836 \\ 5,8672 & 0,1186 & 11,1836 & 16,3197 \end{bmatrix} \tag{B.65}$$

Para o vetor de características $\boldsymbol{\phi}^{ii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,4006 \\ 0 \\ 0,4001 \\ 0,1994 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,9033 & 0 & 0 & 0,0967 \\ 0 & 0,8624 & 0,1376 & 0 \\ 0 & 0,3039 & 0,6961 & 0 \\ 0 & 0 & 0 & 1,0000 \end{bmatrix} \tag{B.66}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 291,3904 \\ 232,5792 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 60,9248 - 11,1546 \\ -11,154670,3183 \end{bmatrix} \tag{B.67}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 312,9456 \\ 193,4109 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 25,5881 & 8,4278 \\ 8,4278 & 17,6465 \end{bmatrix} \tag{B.68}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 293,1157 \\ 198,7737 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 64,6411 & 11,4357 \\ 11,4357 & 95,8294 \end{bmatrix} \tag{B.69}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 306,2108 \\ 217,2888 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 9,3246 & 1,2333 \\ 1,2333 & 51,8633 \end{bmatrix} \tag{B.70}$$

Para o vetor de características $\boldsymbol{\phi}^{iii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0 \\ 0,3606 \\ 0,2395 \\ 0,3999 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 1,0000 & 0 & 0 & 0 \\ 0 & 0,7468 & 0,1127 & 0,1405 \\ 0,1799 & 0 & 0,8201 & 0 \\ 0 & 0 & 0 & 1,0000 \end{bmatrix} \tag{B.71}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -1,0261 \\ -2,5995 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 8,6559 & 0,8115 \\ 0,8115 & 0,9702 \end{bmatrix} \tag{B.72}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 4,3997 \\ 0,3521 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1,7436 & 1,3204 \\ 1,3204 & 1,6012 \end{bmatrix} \tag{B.73}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 4,7524 \\ -2,6475 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 1,6893 & -0,1715 \\ -0,1715 & 0,8837 \end{bmatrix} \tag{B.74}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 1,7954 \\ 0,9157 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0,8596 & 0,0520 \\ 0,0520 & 0,3980 \end{bmatrix} \tag{B.75}$$

B.5.3 Gráficos

Abaixo seguem os gráficos que representam a divisão de *clusters*, valores de LL e distância de Mahalanobis com treinamento com os dez primeiros segmentos válidos. Também está a análise por *3-fold cross-validation* com gráfico ROC.

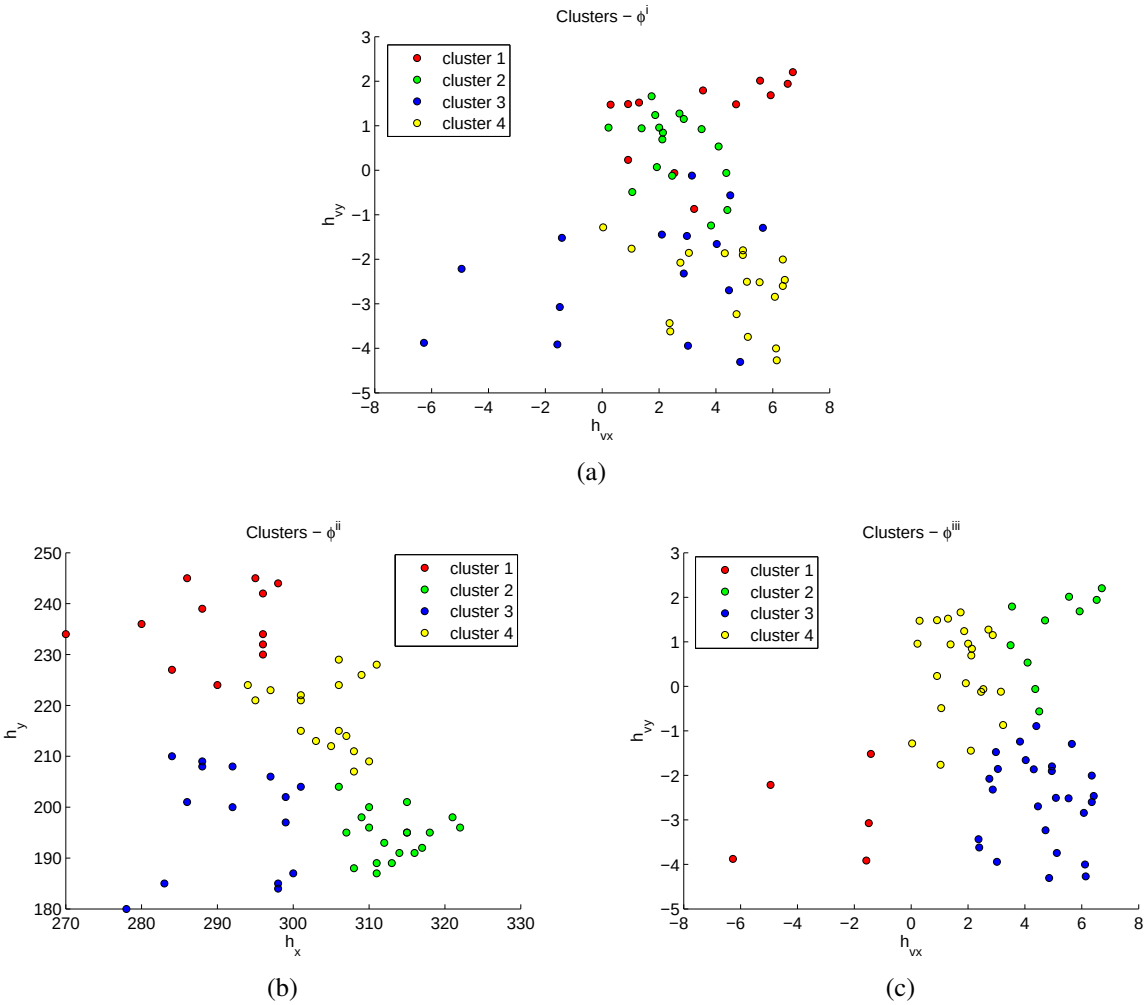


Figura B.17: *Clusters* obtidos com os dez primeiros segmentos válidos do usuário 5.

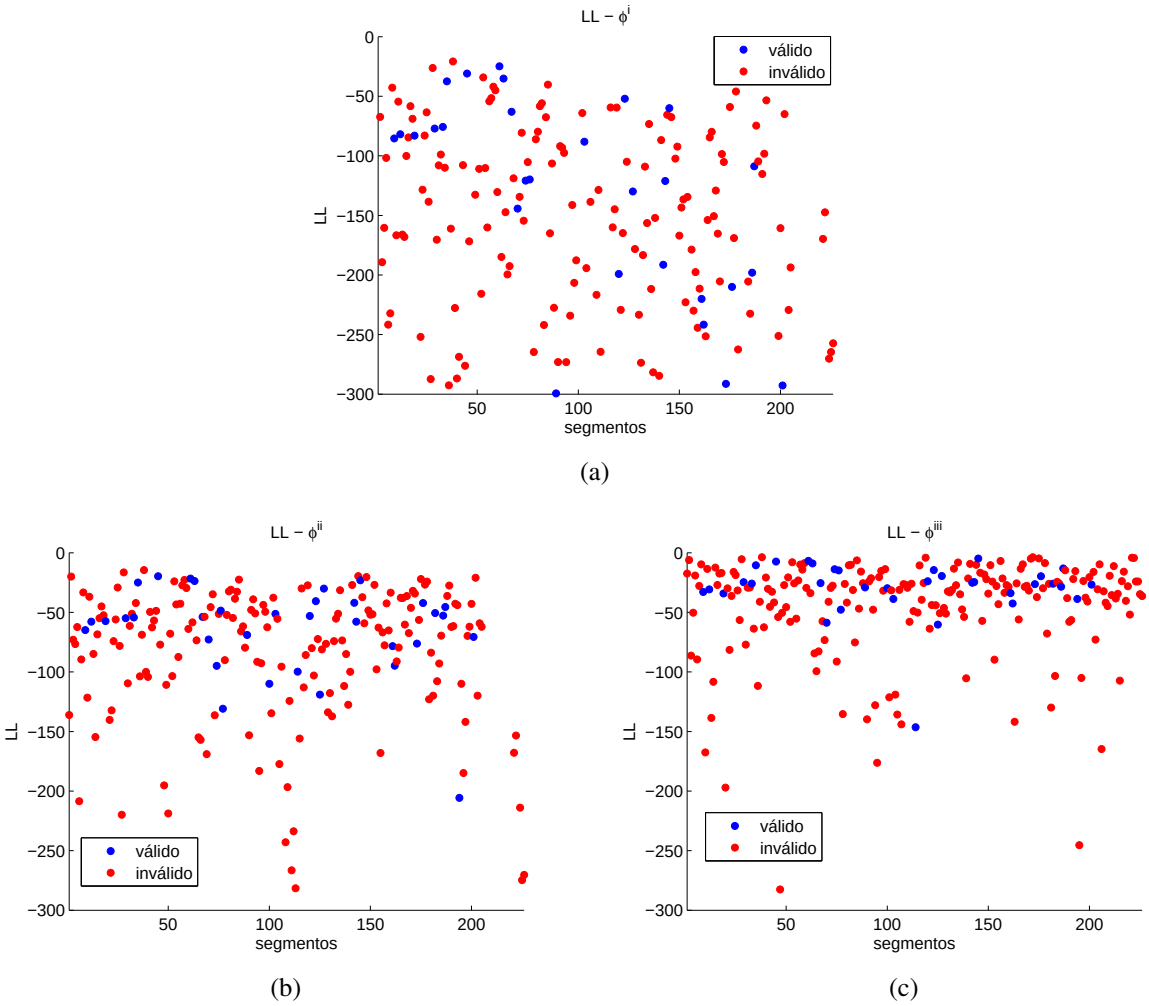
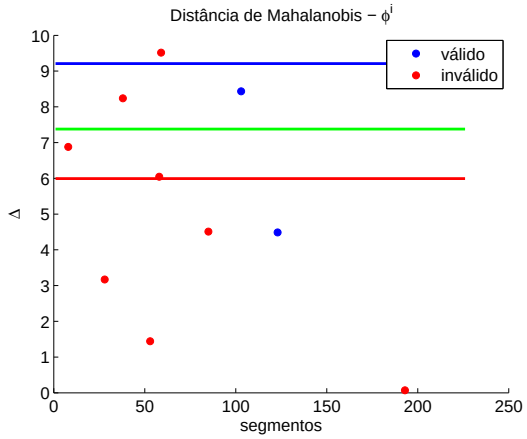
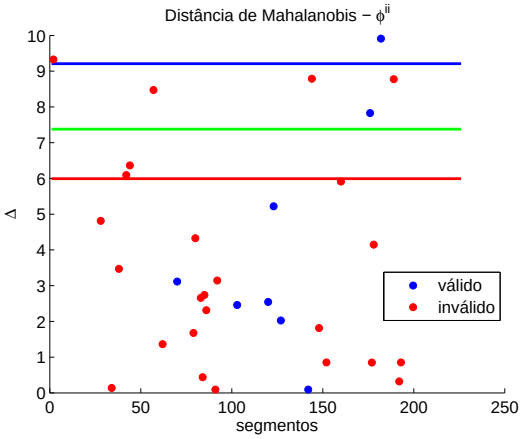


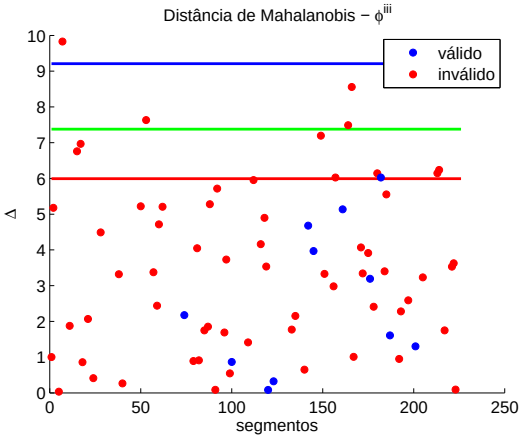
Figura B.18: Valores de LL obtidos com os segmentos de validação do usuário 5.



(a)



(b)



(c)

Figura B.19: Valores de Δ^2 obtidos com os segmentos de validação do usuário 5.

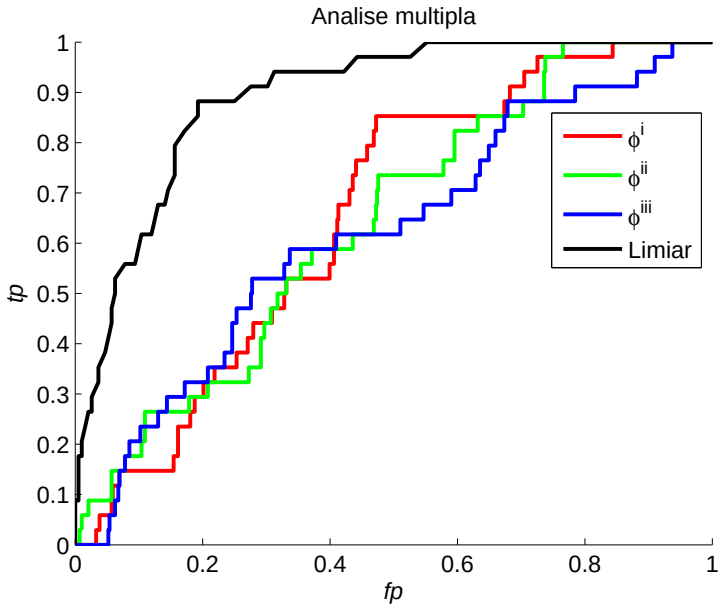


Figura B.20: Gráfico ROC comparando classificadores diferentes com a aplicação do *3-fold crossvalidation* para usuário 5.

B.6 RESULTADOS PARA USUÁRIO 6

B.6.1 Identificação

Idade	15
Nível escolar	médio incompleto
Deficiência	paralisia cerebral
Posição do botão	lado esquerdo
Frase escrita	"Boa tarde"
Número de segmentos da amostra	133
Número de segmentos válidos	17

Tabela B.6: Dados do Usuário 6

B.6.2 Resultados para classificador HMM

Abaixo seguem os parâmetros das cadeias de Markov para os vetores de características $\boldsymbol{\phi}^i$, $\boldsymbol{\phi}^{ii}$ e $\boldsymbol{\phi}^{iii}$.

Para o vetor de características $\boldsymbol{\phi}^i$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,1000 \\ 0 \\ 0,3326 \\ 0,5674 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,7500 & 0 & 0,2500 & 0 \\ 0 & 1,0000 & 0 & 0 \\ 0 & 0 & 0,8044 & 0,1956 \\ 0 & 0,2125 & 0 & 0,7875 \end{bmatrix} \tag{B.76}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 5,8660 \\ 0,6306 \\ 291,0000 \\ 179,0000 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 3,6766 & 0,4503 & 25,1444 & 3,0091 \\ 0,4503 & 0,0727 & 2,9844 & 0,3711 \\ 25,1444 & 2,9844 & 174,5100 & 20,7500 \\ 3,0091 & 0,3711 & 20,7500 & 2,5100 \end{bmatrix} \tag{B.77}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 2,2158 \\ -2,6645 \\ 368,4138 \\ 169,0695 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 9,5286 & 1,1814 & 23,0969 & 15,9518 \\ 1,1814 & 1,1644 & 3,7102 & 4,0585 \\ 23,0969 & 3,7102 & 133,0975 & 24,7287 \\ 15,9518 & 4,0585 & 24,7287 & 45,0688 \end{bmatrix} \tag{B.78}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 4,3024 \\ -0,3343 \\ 341,6213 \\ 189,9872 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 8,596 & 3,591 & -10,003 & -5,439 \\ 3,591 & 2,229 & -5,047 & -0,647 \\ -10,003 & -5,047 & 94,305 & 41,936 \\ -5,437 & -0,647 & 41,936 & 42,195 \end{bmatrix} \tag{B.79}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 3,3281 \\ -0,7613 \\ 359,6698 \\ 185,8199 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 2,047 & 0,107 & -2,493 & 0,313 \\ 0,107 & 1,287 & -2,250 & 1,153 \\ -2,493 & -2,250 & 16,121 & -5,665 \\ 0,313 & 1,153 & -5,665 & 10,655 \end{bmatrix} \tag{B.80}$$

Para o vetor de características $\boldsymbol{\phi}^{ii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,1000 \\ 0 \\ 0,5461 \\ 0,3539 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,7500 & 0 & 0 & 0,2500 \\ 0 & 1,0000 & 0 & 0 \\ 0 & 0,2043 & 0,7957 & 0 \\ 0 & 0 & 0,2009 & 0,7991 \end{bmatrix} \tag{B.81}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 290,9985 \\ 178,9998 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 174,4972 & 20,7487 \\ 20,7487 & 2,5099 \end{bmatrix} \tag{B.82}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 368,5646 \\ 168,4101 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 139,7410 & 27,7560 \\ 27,7560 & 39,2793 \end{bmatrix} \tag{B.83}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 360,1966 \\ 185,4718 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 15,9698 & -6,1003 \\ -6,1003 & 11,6647 \end{bmatrix} \tag{B.84}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 341,9082 \\ 189,9554 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 93,6811 & 40,2137 \\ 40,2137 & 40,9181 \end{bmatrix} \tag{B.85}$$

Para o vetor de características $\boldsymbol{\phi}^{iii}$ temos $\lambda(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$:

$$\boldsymbol{\pi} = \begin{bmatrix} 0,6581 \\ 0,3411 \\ 0 \\ 0,0008 \end{bmatrix} \quad , \quad \boldsymbol{A} = \begin{bmatrix} 0,7480 & 0,1427 & 0,1093 & 0 \\ 0,0556 & 0,6098 & 0 & 0,3346 \\ 0 & 0,1174 & 0,8826 & 0 \\ 0 & 0 & 0 & 1,0000 \end{bmatrix} \tag{B.86}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 2,9101 \\ 0,0119 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1,0931 & 0,1256 \\ 0,1256 & 0,3620 \end{bmatrix} \tag{B.87}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 3,4050 \\ -2,0758 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0,5747 & 0,3231 \\ 0,3231 & 0,6759 \end{bmatrix} \tag{B.88}$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 7,2097 \\ 0,0847 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 1,2561 & 0,1591 \\ 0,1591 & 1,0946 \end{bmatrix} \tag{B.89}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 0,2511 \\ -3,0454 \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 3,0898 & -0,8809 \\ -0,8809 & 0,9224 \end{bmatrix} \tag{B.90}$$

B.6.3 Gráficos

Abaixo seguem os gráficos que representam a divisão de *clusters*, valores de LL e distância de Mahalanobis com treinamento com os dez primeiros segmentos válidos. Também está a análise por *3-fold cross-validation* com gráfico ROC.

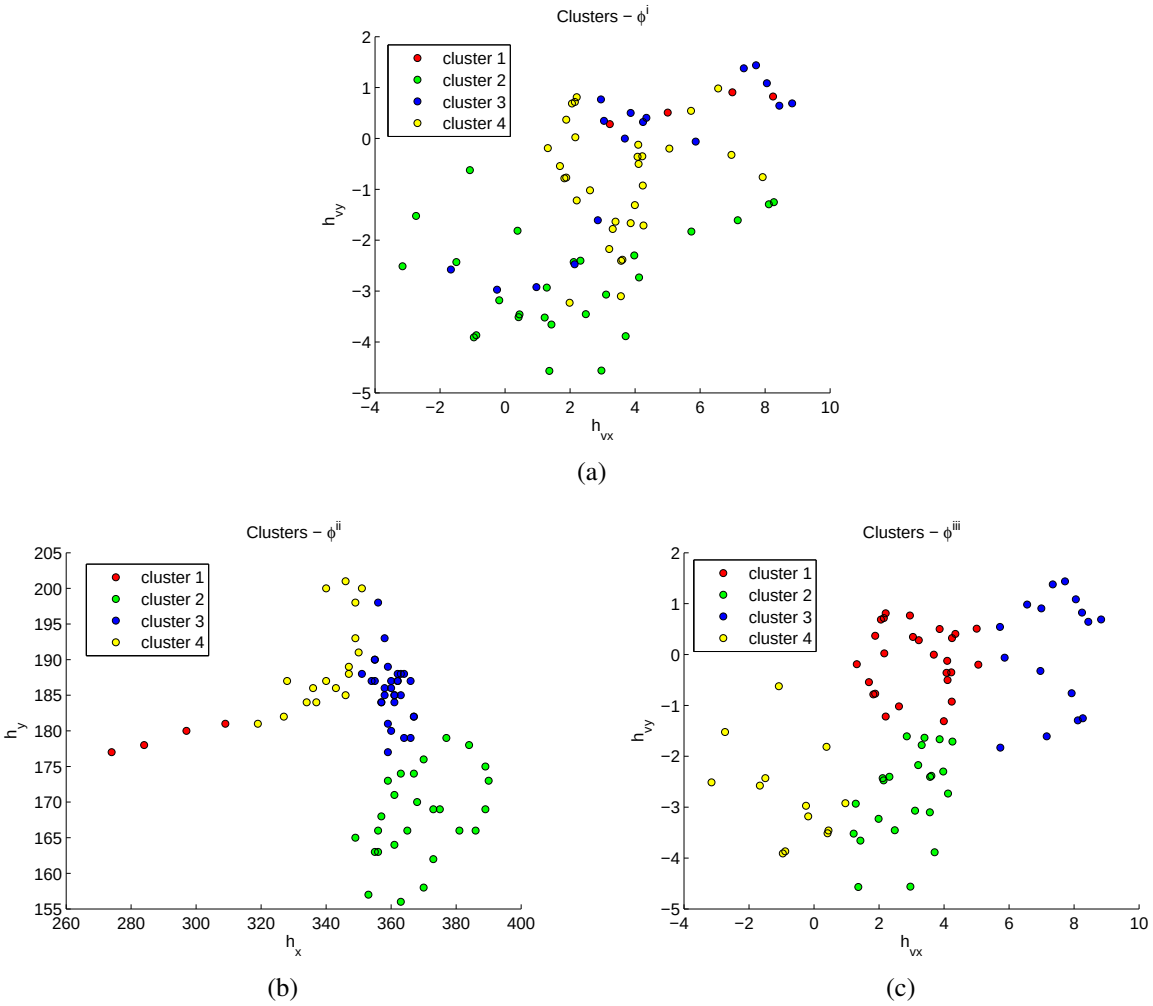


Figura B.21: *Clusters* obtidos com os dez primeiros segmentos válidos do usuário 6.

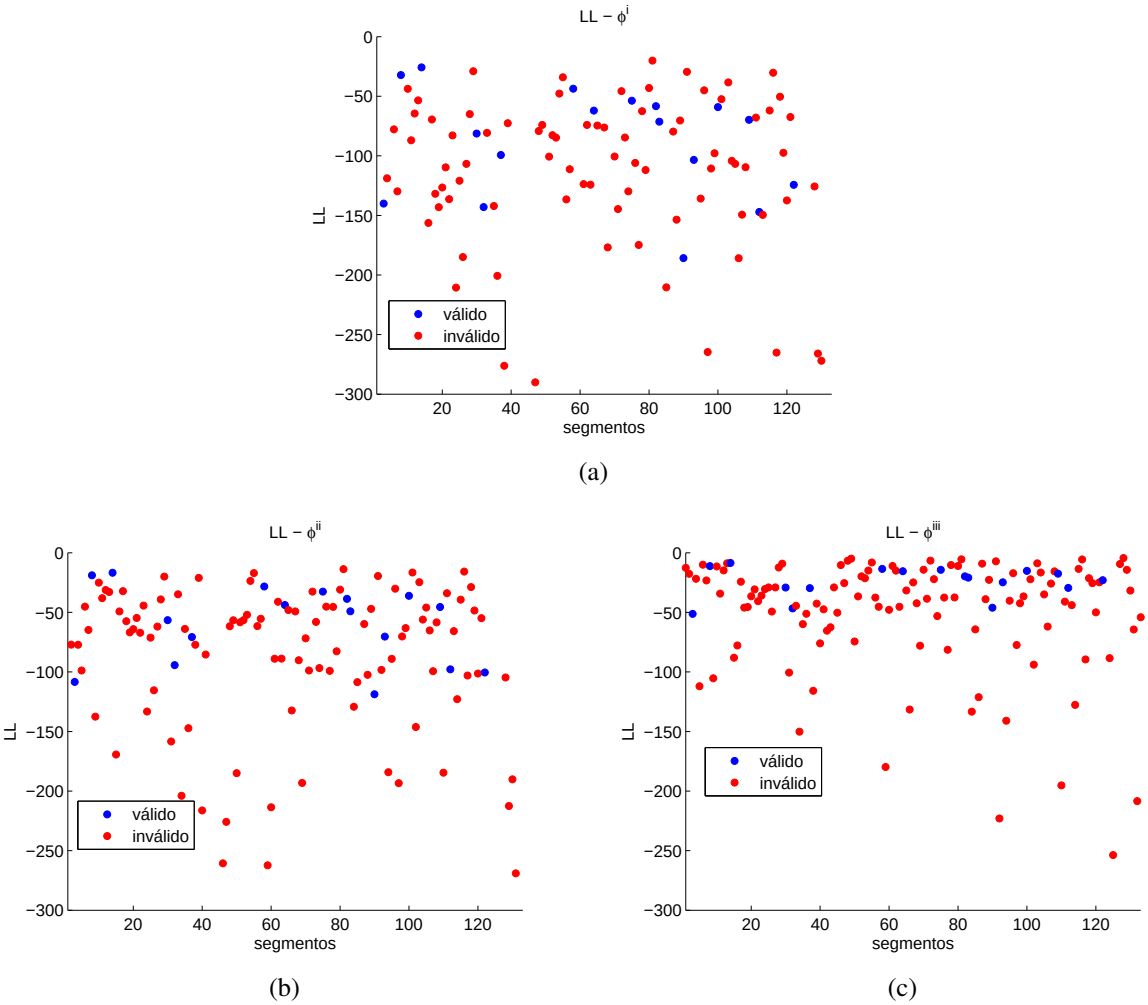


Figura B.22: Valores de LL obtidos com os segmentos de validação do usuário 6.

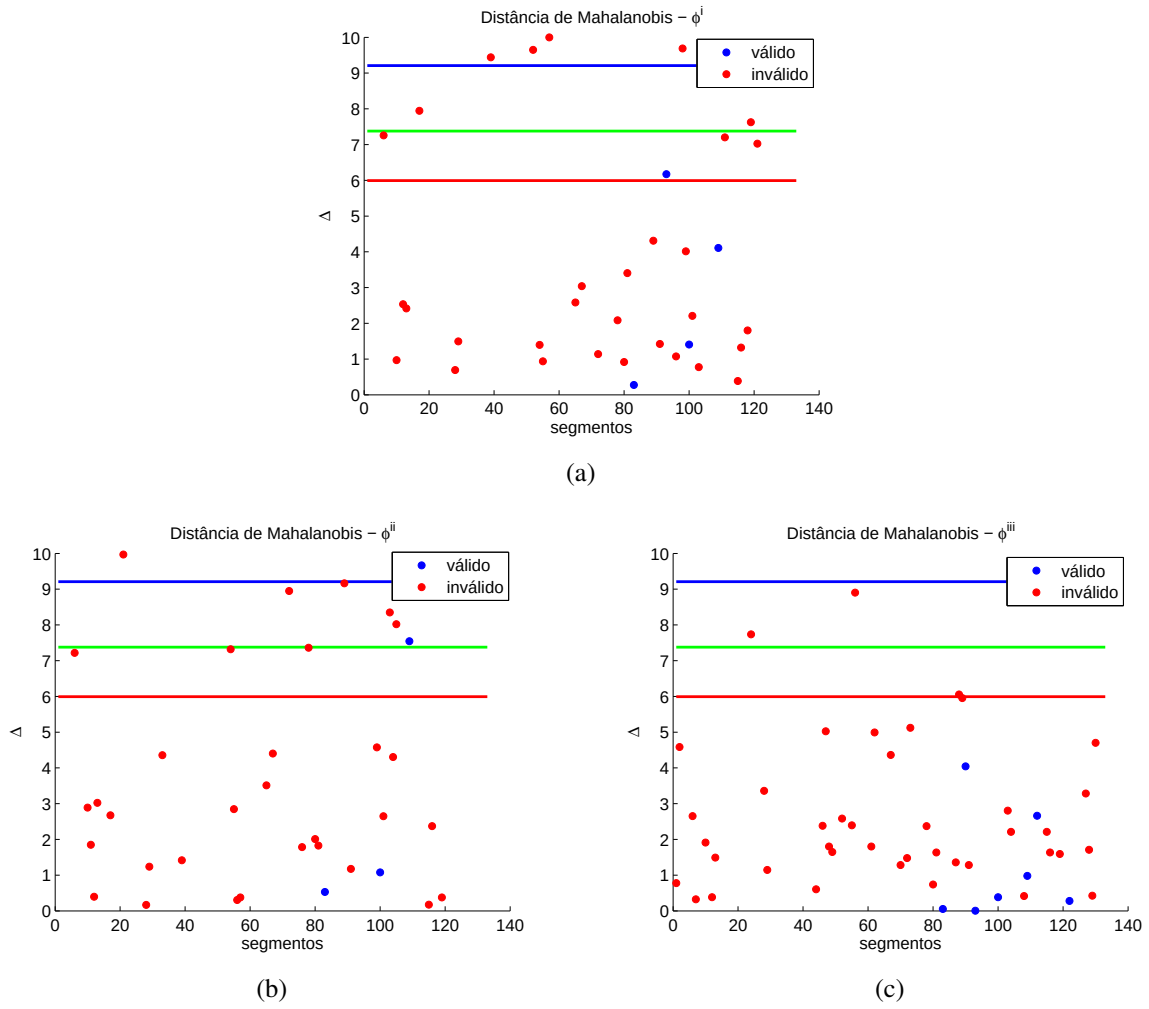


Figura B.23: Valores de Δ^2 obtidos com os segmentos de validação do usuário 6.

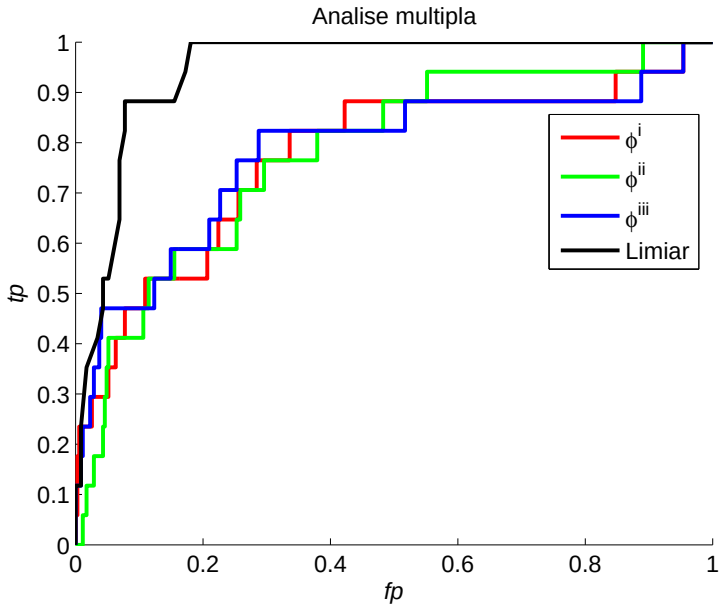


Figura B.24: Gráfico ROC comparando classificadores diferentes com a aplicação do *3-fold crossvalidation* para usuário 6.

C. TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA MAIORES DE 18 ANOS

Você está sendo convidado(a) para participar, como voluntário, da pesquisa "Avaliação de sistema de comunicação alternativa baseado no rastreamento e identificação de movimentos de cabeça". Após ser esclarecido(a) sobre as informações a seguir, no caso de aceitar fazer parte do estudo, assine ao final deste documento, que está em duas vias. Uma delas é sua e a outra é do pesquisador responsável. Desde logo fica garantido o sigilo das informações coletadas. Em caso de recusa você não será penalizado(a) de forma alguma.

1. Informações básicas sobre a pesquisa

Título do Projeto: Avaliação de sistema de comunicação alternativa baseado no rastreamento e identificação de movimentos de cabeça.
Pesquisador Responsável: Carlos Wellington Passos Gonçalves
Telefone para contato (inclusive ligações a cobrar): 61 8152-8699
Pesquisador participante: Antônio Padilha Lanari Bó
Telefone: 61 8169-8477

2. Descrição da participação no projeto

2.1. Objetivo

O objetivo deste trabalho é testar um aplicativo de software que reconhece os movimentos funcionais de uma pessoa e desta forma permite o uso do computador. Este novo sistema será comparado com o equipamento já utilizado por você. Uma tarefa específica de escrita de texto será usada para comparação e os resultados serão utilizados para futuras melhorias e indicação do software.

2.2. Atividades previstas

As atividades previstas durante este projeto serão as mesmas para as interfaces já utilizadas e o novo sistema avaliado. As atividades são divididas em duas etapas:

2.2.1. Apresentação e treino das interfaces A atividade de avaliação é apresentada e um treino é realizado com o objetivo de tirar qualquer dúvida sobre o uso das interfaces no computador utilizado na pesquisa, bem como o uso do seu teclado virtual com preditor de texto. A atividade consiste na reprodução do texto "minha terra tem palmeiras onde canta o sabia"da maneira como esta descrita aqui.

2.2.2. Testes finais com as interfaces O teste final que será utilizado para comparação de dados futura consiste da escrita da mesma frase de treino.

CONSENTIMENTO DA PARTICIPAÇÃO DA PESSOA COMO VOLUNTÁRIO

Eu,_____, RG _____, residente à Av./Rua_____ n. _____, complemento _____, Bairro _____, na cidade de_____, abaixo assinado, concordo em participar do estudo "Avaliação de sistema de comunicação alternativa baseado no rastreamento e identificação de movimentos de cabeça"como voluntário. Fui devidamente informado e esclarecido pelo pesquisador _____ sobre o presente estudo. Foram informados notadamente todos os procedimentos envolvidos, assim como os possíveis riscos e benefícios decorrentes de minha participação. Foi-me garantido o sigilo das informações e que posso retirar meu consentimento a qualquer momento, sem que isto leve a qualquer penalidade ou interrupção de meu acompanhamento. Estou ciente ainda de que as informações obtidas durante as avaliações laboratoriais serão mantidas em sigilo e não poderão ser consultadas por pessoas leigas, sem a minha devida autorização. As informações obtidas, no entanto, poderão ser usadas para fins de pesquisa científica, desde que a minha privacidade seja sempre resguardada.

Local, _____ de_____ de 2_____.
Assinatura: _____
CPF: _____
RG: _____

D. TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA MENORES DE 18 ANOS

Você está sendo convidado(a) para participar, como voluntário, da pesquisa "Avaliação de sistema de comunicação alternativa baseado no rastreamento e identificação de movimentos de cabeça". Após ser esclarecido(a) sobre as informações a seguir, no caso de aceitar fazer parte do estudo, assine ao final deste documento, que está em duas vias. Uma delas é sua e a outra é do pesquisador responsável. Desde logo fica garantido o sigilo das informações coletadas. Em caso de recusa você não será penalizado(a) de forma alguma.

1. Informações básicas sobre a pesquisa

Título do Projeto: Avaliação de sistema de comunicação alternativa baseado no rastreamento e identificação de movimentos de cabeça.
Pesquisador Responsável: Carlos Wellington Passos Gonçalves
Telefone para contato (inclusive ligações a cobrar): 61 8152-8699
Pesquisador participante: Antônio Padilha Lanari Bó
Telefone: 61 8169-8477

2. Descrição da participação no projeto

2.1. Objetivo

O objetivo deste trabalho é testar um aplicativo de software que reconhece os movimentos funcionais de uma pessoa e desta forma permite o uso do computador. Este novo sistema será comparado com o equipamento já utilizado por você. Uma tarefa específica de escrita de texto será usada para comparação e os resultados serão utilizados para futuras melhorias e indicação do software.

2.2. Atividades previstas

As atividades previstas durante este projeto serão as mesmas para as interfaces já utilizadas e o novo sistema avaliado. As atividades são divididas em duas etapas:

2.2.1. Apresentação e treino das interfaces A atividade de avaliação é apresentada e um treino é realizado com o objetivo de tirar qualquer dúvida sobre o uso das interfaces no computador utilizado na pesquisa, bem como o uso do seu teclado virtual com preditor de texto. A atividade consiste na reprodução do texto "minha terra tem palmeiras onde canta o sabia"da maneira como esta descrita aqui.

2.2.2. Testes finais com as interfaces O teste final que será utilizado para comparação de dados futura consiste da escrita da mesma frase de treino.

CONSENTIMENTO DA PARTICIPAÇÃO DA PESSOA COMO VOLUNTÁRIO

_____, RG _____
,neste ato representado por mim _____,
RG _____ residente à Av./Rua _____
n. _____, complemento _____, Bairro _____, na
cidade de _____, abaixo assinado, concorda em participar do estudo
"Avaliação de sistema de comunicação alternativa baseado no rastreamento e identificação
de movimentos de cabeça"como voluntário. Fui devidamente informado e esclarecido pelo
pesquisador _____ sobre o presente es-
tudo. Foram informados notadamente todos os procedimentos envolvidos, assim como os
possíveis riscos e benefícios decorrentes de minha participação. Foi-me garantido o sigilo
das informações e que posso retirar meu consentimento a qualquer momento, sem que isto
leve a qualquer penalidade ou interrupção do acompanhamento. Estou ciente ainda de que
as informações obtidas durante as avaliações laboratoriais serão mantidas em sigilo e não
poderão ser consultadas por pessoas leigas, sem a minha devida autorização. As informa-
ções obtidas, no entanto, poderão ser usadas para fins de pesquisa científica, desde que a
privacidade seja sempre resguardada.

Local, _____ de_____ de 2_____.
Assinatura: _____
CPF: _____
RG: _____