UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

USO DE GESTOS DE MÃO COMO UMA INTERFACE DE INTERAÇÃO ENTRE USUÁRIOS E A TV DIGITAL INTERATIVA

WALTER CHARLES SOUSA SEIFFERT SIMÕES

MANAUS-AM
2014

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

WALTER CHARLES SOUSA SEIFFERT SIMÕES

USO DE GESTOS DE MÃO COMO UMA INTERFACE DE INTERAÇÃO ENTRE USUÁRIOS E A TV DIGITAL INTERATIVA

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Informática, área de concentração: Visão Computacional e Robótica.

Orientador:

Prof. Dr. –Ing. Vicente Ferreira de Lucena Júnior

MANAUS
2014

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Charles Sousa Seiffert Simoes, Walter
C475u Uso de Gestos de Mão Como uma Interface de Interação Entre Usuários e a TV Digital Interativa / Walter Charles Sousa Seiffert Simoes. 2014
141 f.: il. color; 29,7 cm.

Orientador: Prof. Dr. -Ing. Vicente Ferreira de Lucena Júnior
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. TVDi. 2. JavaTV. 3. XletView. 4. Visão Computacional. 5. Reconhecimento de Padrões. I. Lucena Júnior, Prof. Dr. -Ing. Vicente Ferreira de II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

"Uso de Gestos de Mão Como uma Interface de Interação Entre Usuários e a TV Digital Interativa"

WALTER CHARLES SOUSA SEIFFERT SIMÕES

Dissertação defendida e aprovada pela banca examinadora constituída pelos Professores:

PROF. VICENTE FERREIRA DE LUCENA JÚNIOR – PRESIDENTE

PROF. WALDIR SABINO DA SILVA JÚNIOR – MEMBRO

PROF. JOSÉ PINHEIRO DE QUEIROZ NETO – MEMBRO

Manaus, 19 de março de 2014

Dedico este trabalho à minha família, mas principalmente a minha avó Clarice Seiffert Simões por ter sido minha segunda mãe, me dando total apoio, incentivo e investindo na minha qualificação educacional e pessoal e ao meu orientador, Prof. Vicente Ferreira de Lucena Junior, que me mostrou que para ser bom em algo só existe uma regra: trabalhar duro e sempre.

Agradecimentos

Antes de tudo, agradeço a Deus por me permitir concluir este curso.

Agradeço ao meu orientador, Professor Dr. Vicente Lucena Junior, por todo apoio, incentivo, paciência e confiança.

Agradeço também aos meus colegas e amigos do CETELI (Centro de Pesquisa e Desenvolvimento em Tecnologia Eletrônica e da Informação), Vandermi Silva e Wanderlan Carvalho pela ajuda e direcionamentos da pesquisa e da estratégia.

Agradeço ao Professor Ricardo Barbosa da Silva pelas participações nas produções cientificas de artigos que ajudaram a realizar a prova de conceito e a descrição formal da arquitetura deste trabalho.

Agradeço a minha esposa Tatiana pelo apoio e pela paciência durante a pesquisa e o processo de construção do protótipo e da escrita deste trabalho.

À toda minha família: minha filha Mariana, minha Mãe Maria, meu Pai Luiz e meus irmãos Moisés e David.

Este trabalho é de grande importância para minha carreira, pois representa o fim de um ciclo de aprendizado, chegando a determinar a próxima direção da minha vida. Portanto quero agradecer de coração a todos aqueles que compartilharam de alguma forma comigo momentos de trabalho e de descontração.

Há uma força motriz mais poderosa que a eletricidade,
vapor e energia nuclear: a vontade.

Albert Einstein (Prêmio Nobel em Física em 1921).

Resumo

Usuários da nova TV Digital Interativa (TVDi), particularmente da TV Inteligente (*Smart* TV), têm experimentado novas formas de se relacionar com a TV através controles remotos que incorporaram mais funcionalidades, controles de detecção de movimento e os controles através de reconhecimento de gestos. Os *layouts* operados por estes controles tiveram que se adequar as estas novas funcionalidades e apresentam as opções de funcionalidades ora por acesso direto (um único botão), ora pelo acesso indireto (combinação de botões), e, neste segundo caso, tornam a atividade de interação com a TVDi bastante difícil pois exigem um nível de vivência com os dispositivos eletrônicos de interação muito elevado. O controle remoto é o dispositivo padrão utilizado para permitir a interação do usuário com a TV, mas em alguns momentos a sua utilização se torna maçante e difícil. Neste cenário, o uso dos gestos surge como um modo mais natural e menos invasivo para auxiliar ou substituir o controle remoto como possibilidade de interação. Este trabalho propôs a construção de um protótipo para TV Digital Interativa que pudesse ser controlado em suas operações de ajuste de volume do som e de troca de canal de programação através do controle remoto e através de um conjunto de gestos, definidos a partir de um conjunto de regras da Engenharia da Usabilidade. O processo de definição do *layout* e do conjunto de gestos foi realizado com a participação de usuários, e, a partir dessas definições, as funcionalidades da TV construídas de modo a se ter na tela as informações sobre o volume do som e o canal de programação, além do processo de reconhecimento de gestos. O protótipo construído neste trabalho se diferencia de produtos comerciais, pois levou-se em consideração a relação entre o custo e o desempenho dos dispositivos utilizados, buscando oferecer uma opção acessível e mais flexível quanto ao seu uso em equipamentos diversos. A abordagem descrita neste trabalho trata dos desafios que foram enfrentados nas áreas de Engenharia da Usabilidade, TVDi e Visão Computacional tendo seu protótipo final comparado a outros métodos e produtos, mostrando um desempenho de aproximadamente 95% no acerto do gesto exibido e uma taxa de velocidade de 26 *frames* por segundo.

Palavras-chave: TVDi, JavaTV, MHP, DVB, HAVi, *XletView,* JavaCV, *Set-top box*, PACT, Visão Computacional, *Haar*, *AdaBoost*, *Canny*, *Sobel*, Reconhecimento de Padrões.

Abstract

New Interactive Digital TV (iDTV) users, particularly the Smart TV (Smart TV), have experienced new ways of relating to the TV via remote controls that incorporate more functionality, motion sensing controls and controls through gesture recognition. The layouts operated by these controls had to adapt these new features and functionalities have options either by direct access (a single button), sometimes by indirect access (button combination), and in this second case, make the activity interaction with iDTV quite difficult because they require a level of familiarity with the very high electronic interaction devices. The remote control is the standard device used to allow user interaction with the TV, but sometimes its use becomes dull and difficult. In this scenario, the use of gestures emerges as a more natural and less invasive to assist or replace the remote control as a possible interaction mode. This paper proposed the construction of a prototype for Interactive Digital TV that could be controlled in its operations adjusting sound volume and exchanging programming channel through remote and through a set of gestures, defined from a set rules of Usability Engineering. The defining the layout and set of gestures process was conducted with the participation of users, and from these definitions, the features of the TV so constructed as to have the on-screen information about the volume and channel programming in addition to the gesture recognition process. The prototype built in this paper differs from commercial products because it took into account the relationship between the cost and performance of the devices used, seeking to offer an affordable and flexible option as to their use in different equipment. The approach described in this paper addresses the challenges that were faced in the areas of Engineering, Usability, iDTV and Computer Vision with final prototype compared to other methods and products, showing a performance of approximately 95% in the accuracy of the displayed gesture and a rate of speed 26 frames per second.

Keywords: TVDi, JavaTV, MHP, *DVB, HAVi, XletView,* JavaCV, *Set-top box*, PACT, Visão Computacional, *Haar*, *AdaBoost*, *Canny*, *Sobel*, *Pattern-recognition.*

Índice de Figuras

Índice de Tabelas

Lista de Siglas

ABNT	*Associação Brasileira de Normas Técnicas*
ACAP	*Advanced Common Application Platform*
ADABOOST	*Adaptive Boosting*
API	*Application Programming Interface*
ATSC	*Advanced Television System Comitee*
AWT	*Abstract Windowing Toolkit*
CAMSHIFT	*Continuously Adaptive MeanShift*
CES	*Consumer Electronics Show*
CRT	*cathode ray tube*
DAVIC	*Digital Audio Visual Council*
DVB-T	*Digital Vídeo Broadcasting-Terrestrial*
DVD	*Digital Versatile Disc*
ELG	*European Lauching Group*
GEM	*Globally Executable MHP*
H.264	*MPEG-4 parte 10/AVC para codificação de vídeo avançada*
HAVi	*Home Audio Video Interoperability*
HD	*High-definition*
HDTV	*High-Definition TV*
HMM	*Hidden Markov Model*
HSV	*The hue-saturation-value (HSV) color model*
HTML	*HyperText Markup Language*
IHC	*Human-Computer Interaction (Interação Homem-Computador)*
IPTV	*Internet Protocol television*
IRD	*Integrated Receiver Decoder*
ISDB-T	*Integrated Services Digital Broadcasting – Terrestrial*
ISDB-TB	*Integrated Services Digital Broadcasting – Terrestrial Brazilian*
JAVACV	*Java Computer Vision*
JVM	*Java Virtual Machine*
LCD	*Liquid Crystal Display*
LED	*Light-emitting Diode*

LWUIT	*Lightweight User Interface Toolkit*
MEANSHIFT	*Mean Shift*
MHEG	*Multimedia and Hypermedia Information Coding Expert Group*
MHP	*Multimedia Home Platform*
MPEG	*Moving Picture Expert Group*
NBR	*Norma da Associação Brasileira de Normas Técnicas (ABNT)*
NCL	*Nested Context Language*
OPENCV	*Open Source Computer Vision Library*
PACT	*Pessoa, Atividade, Contexto, Tecnologia*
PC	*Personal Computer*
PUC	*Pontifícia Universidade Católica*
RGB	*Red*, *Green*, *Blue*
SBTVD	*Sistema Brasileiro de TV Digital*
STB	*Set-Top Box*
SURF	*Speeded up robust features*
TV	*Television*
TVDi	*Televisão Digital Interativa*
UCD	*Usage Centered Design*
UFAM	*Universidade Federal do Amazonas*
UFPB	*Universidade Federal da Paraíba*
UHF	*Ultra High Frequency*
USB	*Universal Serial Bus*
VC	*Visão Computacional*
VHF	*Very High Frequency*
WI-FI	*Wireless Fidelity*
XML	*Extensible Markup Language*

Sumário

Capítulo 1- Introdução

Algumas mudanças nos dispositivos computacionais de interação têm surgido e apresentado novas formas de comandar os equipamentos relacionados a eles. Entre estes novos modelos de interação estão os que realizam o reconhecimento de gestos, enviando comandos após a captura e interpretação de gestos mapeados. Porém, esta forma de interação necessita de recursos computacionais (processador, memória, etc.) maiores que os dispositivos tradicionais, como os teclados, *mouses* e controles remotos.

Esta forma de controle pode servir de ferramenta auxiliar para facilitar a relação entre os usuários e seus dispositivos eletrônicos, pois, os dispositivos de interação tradicionais podem apresentar diferenças significativas entre si, como os controles remotos de TVs em modelos e marcas diferentes, deixando o usuário se sentindo perdido em meio às configurações e opções de acesso dos aparelhos e serviços, diminuindo a sua usabilidade (ZUFFO, 2013).

A interação homem-computador (IHC) também deve direcionar esforços para a TVDi sob um conjunto de aspectos no tocante às questões de interação como acessibilidade, usabilidade, aceitabilidade, além de observar as questões culturais, sociais e educacionais. Estes aspectos ficam mais evidentes quando observados em populações de países em desenvolvimento, onde há um grande contingente de pessoas com baixo nível de alfabetização digital ao mesmo tempo em que deve observar o peso dos aplicativos finais e a relação destes com os dispositivos que irão executá-los (PESQUISA NACIONAL POR AMOSTRA DE DOMICÍLIOS DO IBGE, 2013).

Várias abordagens têm sido realizadas no sentido de se obter um bom desempenho e eficácia aos reconhecedores de gestos observando as características do alto consumo de recurso de processador e da memória para o processamento das imagens, o que força os usuários a adquirirem equipamentos especiais para esta tarefa, como, por exemplo, o console de videogame *Xbox*® com o *Kinect*® (YI, 2012).

1.1 Problema

Um requisito importante para aplicações interativas diz respeito ao sincronismo espaço temporal entre o envio do comando e a execução do mesmo. É imprescindível que a captura do comando e a resposta do sistema sejam dadas no momento correto (propriedade temporal) e que sejam exibidas nas regiões adequadas da tela da TV (propriedade espacial). Assegurar que estes aspectos sejam respeitados é um fator crítico para que uma aplicação interativa seja aceita pelos usuários (BENYON, 2011).

Várias abordagens já foram dadas para solucionar o problema, porém muitas resultaram em produtos caros e/ou outros lentos, como o *Kinect*® (YI, 2012), abrindo caminho para novas abordagens. Entre diversas abordagens citam-se os trabalhos de Miranda *et al*. (2009), Barros, (2006) e Vatavu, (2012).

(Miranda *et al.,* 2009) propôs a construção de um sistema de operação das funções da TV através reconhecimento de gestos definidos através do uso de marcadores coloridos, colocados sobre os dedos, usados em locais pintados de branco. Barros, (2006) propôs construir interfaces de aplicações televisivas consistentes através do uso de regras de Engenharia da Usabilidade e a participação dos usuários. Vatavu, (2012) definiu um conjunto de gestos das mãos para controlar o aparelho de TV observando o comportamento do usuário.

Este trabalho descreve os esforços planejados dentro da proposta de trabalho de pós-graduação do autor para desenvolver um modo de reconhecimento de gestos, rápido e de baixo custo, aplicado aos sistemas europeu e brasileiro de TVs Digitais.

Este campo de pesquisa é influenciado por várias abordagens dadas, como as de Sellen *et al.* (2009) e Benyon, (2011) em Engenharia da Usabilidade, Barros (2006) em TVDi, Bradski *et al.* (2008) em Visão Computacional, além da integração das técnicas em um único protótipo, como em Simões *et al.* (2013).

1.2 Descrição dos cenários baseados no problema

São desenvolvidos protótipos de TVDi e Visão Computacional que servem para oferecer o ambiente gráfico de manipulação na TV e um conjunto de gestos que, quando capturados e reconhecidos, auxiliam ou substituem as funções de ajuste do volume do som e de troca de canais de programação.

Os protótipos referentes à TVDi são construídos de forma participativa com o usuário observando modelos já existentes e consolidados no mercado de TVDi, aplicando as regras da Engenharia da Usabilidade. O acesso a este protótipo é construído de duas formas: em um emulador e em um *set-top box* real.

O protótipo de Visão Computacional é construído para reconhecer gestos exibidos pelos usuários, utilizando câmeras comuns em lugares onde não há controle de iluminação. Estes gestos são definidos através de experimentos com a participação dos usuários.

Com o objetivo de ilustrar o problema, é desenvolvido um cenário que simula a situação em que um usuário interage com uma TVDi. O cenário baseia-se em uma sala residencial, contendo objetos grandes como o sofá, poltronas, cadeiras, mesas, armários, etc., e objetos pequenos, como enfeites sobre as prateleiras dos armários, objetos sobre a mesa ou almofadas sobre o sofá. Neste cenário, o usuário pode interagir com a TV através do controle remoto ou através de gestos. Para que a interação gestual possa ser realizada, o usuário deve se posicionar em frente à TV, dentro do campo de visão da câmera.

Os objetos descritos no cenário têm de ser considerados neste trabalho, uma vez que o equipamento utilizado para a captura das imagens é uma câmera e um dos modos de segmentar imagem é realizando a detecção de tons de pele e alguns desses objetos se encontram no mesmo grupo de cores e tons causando separação errônea dos gestos encontrados na imagem.

De acordo com o cenário apresentado, são observadas as funcionalidades gerais para nortear o desenvolvimento da arquitetura e posteriormente a coleta de requisitos para o desenvolvimento dos protótipos.

1.3 Motivação

Os atuais usuários das TVs digitais podem ser classificados em dois perfis. São eles os usuários que possuem uma grande experiência no uso de dispositivos modernos de interação e aqueles usuários tradicionais da TV analógica que estão mais habituados ao uso do controle remoto. A grande diferença de experiência entre estes dois perfis de usuários tem forçado os fabricantes a lançarem novas propostas de interação, mais naturais que as encontradas nos controles remotos, para serem utilizadas como auxiliares ou substitutas destes controles

(GAWLINSKI, 2003). Uma forma de se realizar esta interação de forma mais natural se dá através de reconhecimento de gestos.

Entretanto, apesar desta forma de interação ser mais natural ao usuário, ainda não se tornou um padrão de mercado, sendo encontrado apenas em alguns poucos produtos disponíveis, tanto diretamente pelas atuais *Smart* TVs como indiretamente através dos *set-top boxes*, a preços que reduzem o público interessado em utilizá-lo.

Sendo assim, a motivação deste trabalho é investigar a integração da TVDi com a Visão Computacional, sempre observando os fatores de usabilidade e de baixo custo de aquisição do *hardware* e *software* necessários para utilizá-lo. A usabilidade é o fator essencial na busca da diminuição da curva de aprendizado dos usuários, sejam eles com pouca experiência em tecnologias de interação ou os mais familiarizados com os dispositivos modernos de interação e o baixo custo é um fator que atrai o interesse de um público interessado no uso de tecnologia, mas não em aplicar grandes quantias para adquiri-la.

1.4 Justificativa

O reconhecimento de gestos é um caminho natural de interação humano-computador e humano-TVDi, pois para muitas pessoas os gestos são o meio principal de comunicação. Várias tecnologias têm sido propostas para trazer benefícios às pessoas com limitações corporais, de comunicação ou baixa vivência em uso de recursos interativos. Estas tecnologias têm como principal propósito melhorar a qualidade de vida do ser humano no desenvolvimento de processos e ações de seu cotidiano. Trata-se dos ambientes inteligentes.

Este trabalho justifica-se por dois pontos de vista: um prático e um teórico. Do ponto de vista prático, a relevância sobre os achados e conclusões para o aumento da interação do usuário e sua TVDi através de gestos. Do ponto de vista teórico, a pesquisa apresenta uma proposta de solução para a interação através de reconhecimento de gestos da mão na TVDi.

1.5 Objetivo Geral

Investigar e propor a concepção de uma *interface* para controlar a TV Digital Interativa dos padrões de *middlewares* europeu e brasileiro, através de um conjunto de gestos

que, capturados por uma câmera de baixo custo e convertidos em comandos básicos, servem como ferramenta auxiliar ou substituta ao controle remoto na ação de interação com a TVDi.

1.6 Objetivos Específicos

Os objetivos específicos que direcionam este trabalho são:

- Identificar no estado da arte um conjunto mínimo de técnicas e ferramentas de Engenharia da Usabilidade, TVDi e Visão Computacional para construção de *layouts* e mapeamento de gestos para serem utilizados como comandos na TV; para a construção de aplicativos de TV;
- Adaptar as técnicas de Engenharia da Usabilidade, TVDi e Visão Computacional para a construção de *layout* e mapeamento de gestos com a participação do usuário, para a construção do protótipo de TV e para o reconhecimento de gestos;
- Experimentar em um estudo de caso, através de um protótipo, as operações básicas de ajuste do volume do som e a troca de canais de programação na TVDi com uso de recursos de Visão Computacional, realizando o reconhecimento de gestos de forma rápida e eficiente;
- Coletar e analisar os dados empíricos, resultantes da aplicação dos métodos adotados para a construção do protótipo de TVDi com uso de Visão Computacional.

1.7 Método de Pesquisa

O método de pesquisa utilizado para desenvolver este trabalho consiste em cumprir as etapas descritas pelos objetivos específicos da seção 1.6 através de uma extensa pesquisa bibliográfica. Nesta pesquisa são destacados os trabalhos relacionados envolvendo Engenharia da Usabilidade, a TVDi e a Visão Computacional, além da integração destes, fazendo uma revisão crítica da literatura em artigos nacionais e internacionais sobre o tema.

Na segunda etapa são realizados experimentos com *softwares*, além de experimentos com *set-top boxes* disponíveis nos laboratórios do Centro de Tecnologia Eletrônica e Telecomunicações (CETELI). Nesta etapa são escolhidas técnicas apresentadas em artigos consolidados para que sejam repetidos comparando os resultados obtidos com as conclusões observadas pelos autores. Em relação aos *set-top boxes*, estes são testados quanto as suas

características físicas e lógicas. A partir desses experimentos, são introduzidas modificações e contribuições necessárias às técnicas experimentadas, além da escolha das características dos equipamentos a serem utilizados.

Na terceira etapa, são modelados e implementados os protótipos de TVDi e Visão Computacional, integrados através de troca de mensagens entre os objetos pertencentes as aplicações.

Na etapa final os resultados são apresentados e comparados com os obtidos nos trabalhos relacionados. Estes resultados obtidos são extraídos dos experimentos realizados e descritos em artigos que foram submetidos a congressos nacionais e internacionais relacionados a cada área de pesquisa do tema.

1.8 Organização do Trabalho

O primeiro Capítulo apresentou a ambientação do trabalho, os conceitos iniciais sobre Engenharia da Usabilidade, TVDi e Visão Computacional, o cenário para auxiliar o entendimento do problema, os objetivos e o método de trabalho utilizados.

O Capítulo 2 apresenta o referencial teórico e os conceitos sobre a Engenharia da Usabilidade, da Televisão Digital Interativa, com ênfase aos modelos europeu e brasileiro e a Visão Computacional em relação ao mapeamento de gestos e o processo de reconhecimento.

O Capítulo 3 apresenta os trabalhos relacionados nesta pesquisa, trazendo aqueles que foram considerados os mais relevantes no processo de definição das técnicas e da construção dos protótipos, dando sustentação ao tema que está sendo abordado.

O Capítulo 4 descreve a arquitetura de sistema da solução proposta, utilizada para a condução do trabalho, apresentando em ordem sequencial como o mesmo é desenvolvido, de modo que a adoção dos métodos tomados faça sentido a outros pesquisadores das áreas abordadas.

O Capítulo 5 apresenta os *softwares*, ferramentas e equipamentos necessários para o desenvolvimento dos protótipos, baseados na arquitetura e estratégias definidas para este trabalho. Os protótipos são construídos para o cenário de utilização da TV em uma residência, observando as características descritas da arquitetura do sistema e das ferramentas utilizadas.

O Capítulo 6 apresenta os testes e resultados obtidos sobre os protótipos construídos e documentados no Capítulo 5, que tratou dos assuntos de Engenharia da Usabilidade, TVDi e

Visão Computacional. As análises e interpretações dos resultados têm como parâmetros os trabalhos apresentados no Capítulo de Trabalhos Relacionados e tem como foco o problema e os objetivos gerais e específicos apresentados neste trabalho.

O Capítulo 7 apresenta as considerações finais deste trabalho, onde são listadas as dificuldades encontradas dentro de cada área abordada, as sugestões para trabalhos futuros e contribuições da pesquisa tanto para a ciência como para a sociedade. Ao final, são apresentadas as conclusões deste trabalho relacionando desde o objetivo até o seu desfecho com a interpretação dos resultados.

Capítulo 2- Referencial Teórico

A fim de compreender os conceitos e soluções discutidas neste trabalho, é necessário compreender como funcionam algumas técnicas e tecnologias, como estas se ajustam e se relacionam para serem utilizadas em conjunto para construção de protótipos de interação gestual para a TV.

O capítulo inicia com uma explanação sobre a Engenharia da Usabilidade e as técnicas utilizadas para se modelar e desenvolver um produto e/ou serviço, com o foco no desenvolvimento centrado no usuário, buscando chegar a uma melhor combinação dos componentes que são as pessoas, atividades, contextos e tecnologias em um domínio em particular. Em seguida, são apresentados os ambientes de aplicações procedurais e declarativas de TVDi, as diferenças entre eles em relação as tecnologias de *hardware* e *software* que influenciam o processo de desenvolvimento de aplicativos e o modo de interação. Após definida a TVDi passa-se à apresentação da Visão Computacional que descreve a construção da base de conhecimento do sistema e os processamentos de imagens, que buscam eliminar os ruídos e os dados não necessários para a comparação com o classificador. As etapas de definição do classificador e de processamento de imagens também levam em conta as limitações de memória e os processadores presentes nas TVDi e *set-top boxes* atualmente disponíveis.

2.1 Engenharia da Usabilidade

A baixa usabilidade encontrada em muitos produtos de eletrônica de consumo como as TVs, videocassetes e tocadores de DVD é descrita por Brackmann, (2010), que afirma que o problema mais importante nestes equipamentos é a *interface* com o usuário. Esta *interface* sofre diversas modificações no tocante as suas funcionalidades quando verificados modelos e marcas diferentes, o que torna excessivamente complexo o entendimento e o uso destes produtos. Quando buscada a percepção da facilidade de uso da TVDi junto aos usuários, estes indicaram que a TVDi foi considerada mais difícil de usar do que um computador ou um

carro, enquanto que a TV tradicional foi considerada tão fácil de usar quanto um secador de cabelos (BRACKMANN, 2010).

Considerando que a TV é voltada principalmente para as atividades de entretenimento, o simples fato do usuário supor que será difícil interagir com o equipamento já é um fator impeditivo para o seu uso (BRACKMANN, 2010). A usabilidade é a qualidade que caracteriza o uso dos programas e aplicações e sua essência é o acordo entre *interface*, usuário, tarefa e ambiente (CYBIS *et al*., 2007).

Entre as técnicas de modelagem de *interfaces* para melhorar a usabilidade, destaca-se a *usage-centered-design* (desenvolvimento centrado no usuário). Ser centrado no ser humano, em termos de *design*, é caro e implica em observar pessoas, conversar e experimentar ideias com elas, o que demanda um tempo para ser realizado. Porém, este tempo gasto no processo de modelagem das *interfaces* é justificado com uma diminuição de recursos e estruturas necessárias para o atendimento de dúvidas ou reclamações por parte do usuário, redução de custos com a confecção de materiais de treinamento, etc., quando a *interface* estiver concluída. Além disso, o percentual de aceitação é maior quando comparado a produtos e serviços construídos sem o uso dessa técnica (CYBIS *et al*., 2007).

Para se desenvolver uma modelagem centrada no usuário, algumas grandezas precisam ser consideradas. São elas: a pessoa, a atividade, o contexto e a tecnologia. Para se atingir uma melhor combinação entre estes elementos pode-se utilizar ferramentas e técnicas como o *brainstorm* por meio de observações, entrevistas e *workshops* com pessoas do grupo de interesse e outras técnicas de antecipação e trabalho como o *Card Sorting* (arranjo de cartas), o diagrama de afinidade, o *Storyboard* (narração gráfica), maquetes (protótipo de papel) e a prototipagem rápida (ROSSON *et al*., 2002).

A técnica *Card Sorting* visa descobrir o modelo mental do usuário em relação a itens de informação para uma aplicação. Nesta técnica, o analista descreve os itens em fichas de papel e as espalha sobre a mesa. Um usuário é convidado para organizar estas fichas de acordo com o seu entendimento sobre o problema. Este processo é realizado no máximo seis vezes e os resultados são combinados pelo analista.

A técnica de diagrama de afinidade é utilizada para organizar uma grande quantidade de itens em grupos lógicos. Os projetistas e usuários alvo trabalham juntos para obter um consenso sobre a organização de itens, identificando e agrupando as funções de um produto em desenvolvimento.

Storyboard é uma técnica extraída do cinema – usando uma estrutura simples no estilo de desenho animado, em que momentos-chave da experiência interativa são representados. A vantagem do *storyboard* é que ele permite uma percepção do fluxo da experiência.

As maquetes, também chamadas de protótipos em papel, são usadas para esclarecer e desenvolver requisitos específicos para a interface de um programa. As maquetes permitem simular e testar a interação com o usuário, proporcionando a identificação precoce de problemas de usabilidade.

A prototipagem rápida diz respeito a técnicas e ferramentas para testar soluções a baixo custo de forma eficiente. Esta técnica permite realizar testes com o usuário antes do desenvolvimento da aplicação, obtendo resultados mais satisfatórios (SNYDER, 2013).

Uma das ferramentas que une estas diversas técnicas e auxilia na definição das estratégias do que realmente importa em um processo construtivo é o *framework* PACT (Pessoas, Atividades, Contextos, Tecnologias) (BENYON, 2011).

Para o item pessoas do PACT, deve-se pensar nas diferenças físicas, psicológicas e sociais e como essas diferenças mudam as circunstâncias com o passar do tempo. O entendimento e o conhecimento que se tem de alguma coisa é chamado modelo mental (CYBIS *et al*., 2007).

Os modelos mentais servem para dar uma noção sobre o que será construído, porém são incompletos, porque as pessoas entendem algumas partes de um sistema melhor que outras; as pessoas estão dispostas a realizar operações físicas adicionais para minimizar o esforço mental; por exemplo, desligam e reiniciam um dispositivo ou aplicativo em vez de tentar resolver um erro (MELO *et al.* , 2008). Devido estas características, diversos estudos são realizados para se descrever as relações entre as pessoas e seu ambiente, pois é essencial que seus *designs* sejam compatíveis com o fator humano e o cenário de uso (DIX *et al*., 1998).

O item atividade do PACT é utilizado tanto para as tarefas simples quanto para as altamente complexas e longas, por isso, é necessário ter cuidado quando se estiver descrevendo as atividades, pois não se pode perder o foco no objetivo da atividade principal. Suas principais características são os aspectos temporais, a cooperação, a complexidade, a segurança e a natureza do conteúdo. Diferentes técnicas podem ser utilizadas para a modelagem de atividades como a Forma Descritiva proposta por Antikainen *et al*. (2003) ou a Técnica de Roteiros de Cavazza, (2002) por serem consideradas baratas, fáceis e, de certa forma, intuitivas e lúdicas.

O item contexto do PACT é utilizado para situar as atividades. Atividades sempre acontecem em um contexto, de modo que é preciso analisar os dois em conjunto. Pode-se identificar quatro tipos úteis de contexto:

- O contexto organizacional, que indica que mudanças na tecnologia frequentemente alteram a comunicação e podem afetar funções já conhecidas, desqualificando-as e forçando os usuários a terem que reaprender tudo novamente;
- O contexto social, que indica o quão favorável o ambiente será para realizar a atividade;
- A circunstância física na qual acontece e delimita a atividade, como por exemplo, se um ambiente é barulhento, frio, úmido ou sujo;
- A análise de contexto de uso, que é feita sobre as fontes de informação. A principal fonte de informação e que deve ser primeiramente consultada, quando possível, é o próprio usuário.

O item tecnologia do PACT é utilizado para indicar que componentes de *hardware* e *software* permitem a interação do usuário e seu dispositivo eletrônico e o modo de transformação destes dados de entrada em instruções. Como a usabilidade é centrada no ser humano, os dispositivos de entrada se apoiam nas habilidades perceptivas de visão, audição e tato. É importante obter informações sobre estas percepções para a caracterização do contexto de uso de um produto ou serviço. As técnicas de obtenção de informações sobre estas percepções são classificadas em qualitativas e quantitativas.

As técnicas qualitativas são mais apropriadas quando se utiliza um conjunto pequeno de usuários (de 10 a 20). Uma desvantagem da pesquisa qualitativa é que ela não prova nada em definitivo uma vez que os dados não são analisados por meio de técnicas estatísticas, porém é uma excelente forma de gerar hipóteses sobre o problema e apontar direções para a pesquisa quantitativa (CYBIS *et al*., 2007).

Quando se deseja obter informações numéricas, tais como, grau de preferência entre versões de um mesmo produto, as técnicas quantitativas são necessárias. A vantagem da técnica quantitativa é oferecer uma compreensão melhor e mais realista da amostra de usuários que está sendo estudada e a desvantagem é que são necessários muitos dados para se elaborar a estatística do problema (CYBIS *et al*., 2007).

Ambas as técnicas, a qualitativa e a quantitativa, podem utilizar diversas ferramentas para a obtenção dos dados necessários à construção de um produto ou serviço. As ferramentas mais utilizadas são:

- Observação direta: podem ser realizadas no próprio ambiente de trabalho dos usuários ou em laboratório. A observação pode ser passiva (simplesmente ouvindo e observando) ou ativa (questionando);
- Entrevistas: têm o objetivo de colher informações sobre os usuários e suas atividades através de conversas com pessoas envolvidas, podendo ser feitas individualmente ou em grupo. Nas entrevistas individuais, busca-se descobrir as características que diferenciam os indivíduos e nas entrevistas em grupo, buscar aproximar as características dos integrantes do grupo em um modelo único;
- Levantamento por questionário: é indicada quando se deseja obter informações tais como a experiência, atitudes e preferências dos usuários que podem afetar o modo como eles realizam suas atividades. Geralmente são utilizadas quando se deseja investigar informações estatísticas sobre uma grande população de usuários.

2.2 TVDi

Para desenvolver uma aplicação para a TVDi, devem ser observadas características físicas e lógicas utilizadas, pois ainda não há um padrão unificado, fazendo com que este desenvolvimento seja considerado um negócio especializado (JUCÁ, 2006).

Para se realizar interatividade com a TVDi faz-se necessário utilizar um conjunto de funcionalidades, que podem ser acessadas diretamente no *hardware* da TV ou em um *set-top box*, quando disponibilizados pelos fabricantes. Os equipamentos que possuem capacidade de receber *softwares* oferecem duas formas para tal possibilidade: através da construção de *software* embarcado ou através de uso de servidores de aplicação que fornecem o aplicativo ou os resultados dos seus processamentos para a TVDi.

Para se construir um *software* embarcado, deve-se pensar nas dificuldades inerentes a sua execução em um equipamento específico, onde uma série de adequações é necessária para relacionar a programação com a mecânica ou a característica eletrônica do *hardware.* Esta situação é comum a todos os aplicativos que são embarcados (SIMOES *et al*., 2010), (LUCENA *et al*., 2007). Por esta razão, foram construídos emuladores de TVDi para que os aplicativos sejam testados no PC. Entre estes emuladores citam-se o *OpenMHP* e o *XletView* (SILVA, 2009).

O *OpenMHP* possui boa qualidade de documentação e ótimo acesso para execução de códigos *Xlet*, proporcionando amplo *debug* do código testado. Apesar destas características, o emulador possui algumas deficiências devido ao baixo número de classes MHP implementadas no emulador, principalmente as responsáveis pela *interface* gráfica.

O *XletView* foi desenvolvido em Java para ter sua execução independente do sistema operacional e simula uma TVDi em um *desktop* baseado na *middleware* MHP. Este emulador traz como vantagem o suporte a um número de classes MHP maior que o *OpenMHP* e por isso é o mais utilizado para auxiliar na construção de aplicativos para a TV.

Os *softwares* direcionados à construção de aplicativos para a TVDi podem ser dos tipos procedural e declarativo. Os *softwares* procedurais utilizam linguagens de programação com capacidade de manipular os recursos de *hardware* disponíveis na TV ou no *set-top box*, e possuem a linguagem Java como maior representante (ETHERIDGE, 2009). Os *softwares* declarativos apenas utilizam funções pré-concebidas e não tem a capacidade de manipular o dispositivo físico, tendo como seu maior representante a linguagem NCL-LUA (ITU-T *Recommendation* H.761, 2009), (PICANÇO *et al*., 2013).

2.2.1 Ambiente de Aplicação Procedural de TV

Os *middlewares* que permitem o desenvolvimento procedural de aplicativos para a TV mais utilizados atualmente são o MHP e o Ginga-J. Ambos os sistemas utilizam a mesma linguagem de programação e um conjunto de especificações, que compõem o GEM (*Globally Executable* MHP) (AMERINI *et al*., 2011). O GEM é um acordo de harmonização que captura as interfaces e a semântica do MHP e as tornam compatíveis com os outros padrões. O GEM utiliza a biblioteca AWT do Java e adiciona classes HAVi.

O primeiro *middleware* citado, o DVB-MHP, é composto basicamente por uma máquina virtual Java, capaz de executar *bytecodes* dessa linguagem, e por um conjunto de bibliotecas encarregadas de fornecer diversas funções às aplicações específicas de uma televisão. As APIs mais importantes do MHP são a JavaTV, a HAVi e a Davic.

A API JavaTV introduziu o conceito de *Xlet* que se trata de uma aplicação Java para TV Digital Interativa. Essas aplicações possuem um controlador, o *Xlet Manager*, que gerencia a aplicação do terminal de recepção é o responsável por todo o ciclo de vida das *Xlets*.

A *interface* HAVi (*Home Audio Video Interoperability*) permite que aplicações escritas em Java determinem os recursos presentes em cada *set-top box* e com isso desenhem sua *interface* gráfica na tela, manipulem dados inseridos pelo usuário, executem pequenos arquivos de som, etc (TEIRIKANGAS, 2001).

A API DAVIC (*Digital Audio Visual Council*), baseado no padrão MHEG-6, adiciona um conjunto de novas APIs Java ao padrão MHP, permitindo controlar a apresentação de áudio e vídeo e gerenciar os recursos do receptor. O MHEG-6 é frequentemente associado à interatividade avançada MHEG. O MHEG (*Multimedia and Hypermedia Experts Group*) foi formado por um subcomitê da ISO (*International Standards Organization*) para resolver os problemas de construção de apresentação multimídia interativa (RANGEL *et al*., 2011).

Cada *Xlet* deve implementar os seguintes métodos:

- *initXlet*: inicializa o *Xlet* e muda o estado deste para Pausado. Esse método é chamado apenas uma vez;
- *startXlet*: o estado do ciclo de vida da *Xlet* é modificado para Ativo;
- *pauseXlet*: o estado do ciclo de vida da *Xlet* é modificado para Pausado;
- *destroyXlet*: o estado do ciclo de vida da *Xlet* é modificado para Destruído.

A Figura 2.1 mostra o ciclo de vida dos *Xlets.*

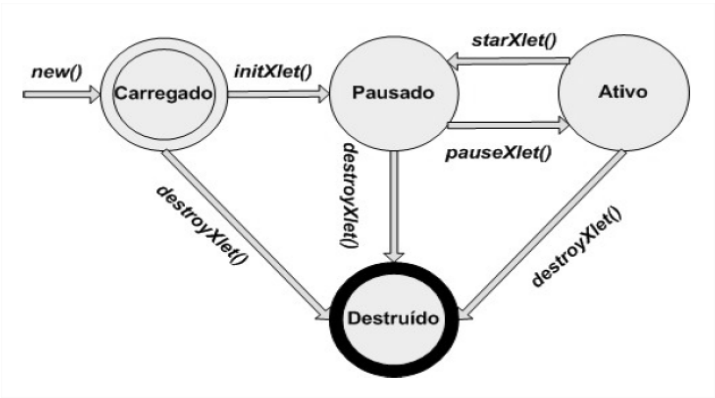


Figura 2.1 Ciclo de vida de uma Xlet.
Fonte: (SILVA, 2009).

O segundo *middleware* citado, o Ginga-J, foi inicialmente desenvolvido para prover uma infraestrutura de execução de aplicações baseadas na linguagem Java, incluindo facilidades especificamente voltadas para a TVDi, fornecidas pela especificação aberta JavaDTV (FILHO, 2007).

A API JavaTV é o núcleo do Ginga-J e é responsável pelo ciclo de vida das aplicações Xlet e pelas funcionalidades relacionadas a interface gráfica com o usuário, tendo como base a LWUIT. O principal objetivo da LWUIT é substituir a API HAVi do GEM (KULESZA, 2010).

Um emulador foi construído a partir do código do *XletView* para fornecer uma camada pura em Java para a execução de aplicações Ginga-J. Um dos principais objetivos da construção deste emulador é oferecer um ambiente para o desenvolvimento de aplicações Ginga-J livre das bibliotecas HAVi, DAVIC e DVB e com acesso as bibliotecas JavaTV, JavaDTV e LWUIT.

2.2.2 Ambiente de Aplicação Declarativo de TV

O Ginga-NCL é o padrão do *middleware* declarativo do sistema brasileiro de TVDi e foi desenvolvido com o objetivo de prover uma infraestrutura de apresentação de aplicações declarativas escritas na linguagem NCL (*Nested Context Language*), que é uma aplicação XML (*Extensible Markup Language*) com facilidades para a especificação declarativa de aspectos de interatividade, sincronismo espaço-temporal entre objetos de mídia, adaptabilidade, suporte a múltiplos dispositivos e suporte a produção ao vivo de programas interativos.

O padrão Ginga-NCL é baseado no NCM (*Nexted Context Model*), que é o modelo conceitual para especificação de documentos *hipermídia* com sincronização temporal e espacial entre seus objetos de mídia.

Para as tarefas que requerem uma programação algorítmica, NCL tem LUA como sua linguagem de *Script*. LUA combina uma sintaxe para programação imperativa com descrição de dados baseadas em tabelas associativas. Hoje o *Script* LUA tornou-se um padrão internacional na área de desenvolvimento de aplicativos para as áreas de TV e também de jogos (IERUSALIMSCHY *et al*., 2007).

2.3 Visão Computacional

A Visão Computacional (VC) faz parte do grupo de ciências que descrevem um ambiente inteligente. Os ambientes inteligentes são ambientes interativos que trazem a

computação, através de sistemas embarcados e tecnologias da informação e comunicação, como modo de melhorar as atividades comuns, fazendo com que esta computação não apenas seja amigável ao usuário, mas invisível. Assim, a Visão Computacional busca propor formas de interação do usuário com os seus dispositivos eletrônicos nestes ambientes de forma não invasiva e amigável (STEVENTON, 2006).

O reconhecimento de gestos é uma aplicação da área de Visão Computacional que está em evidência pelos diversos dispositivos que estão surgindo e substituindo os elementos de controle até então definidos como padrão (paradigma). Um conjunto de técnicas de processamento de imagens e análise de séries temporais é utilizado para permitir que um sistema interprete o gesto apresentado e o relacione a algum comando pré-determinado (WANGENHEIM *et al*., 2011). O melhor exemplo é o jogo *Xbox*® *Kinect*® (YI, 2012) que adicionou como opção de controle um subsistema que captura os gestos dos usuários. É denominado reconhecimento o processo de atribuição de um rótulo a um objeto baseado em suas características, traduzidas por seus descritores.

Os sistemas de interação baseados em visão não utilizam dispositivos de rastreamento explícitos, apenas câmeras para a captura das imagens, não existindo restrições para as características das câmeras, podendo ser dos tipos infravermelhos, sensíveis à temperatura, baseadas em distância, etc. Desde que a câmera seja a única fonte de captura de informação, diz-se que o mecanismo de interação é baseado em Visão Computacional.

Para se resolver o reconhecimento de gestos, dois processos são essencialmente importantes: a construção do classificador, que serve como base de conhecimento ao sistema e o processamento de imagens, que submete uma imagem ou um grupo de imagens a um processo de limpeza de ruídos e eliminação de dados não necessários à comparação com o classificador (BRADSKI *et al*., 2008). Estes dois processos são divididos em 6 partes, sendo que as 3 primeiras são inerentes ao processamento digital de imagens e as 3 últimas ligadas ao uso dos classificadores no processo de reconhecimento. São elas:

- Aquisição;
- Pré-processamento;
- Segmentação;
- Extração de características;
- Base de Conhecimento;
- Reconhecimento e interpretação.

2.3.1 Criação dos Classificadores de Gestos

Os classificadores são responsáveis pela representação dos gestos ou objetos que se pretende reconhecer, ou seja, são utilizados como base de conhecimento de sistemas de Visão Computacional. O modo de se construir um classificador pode ser definido a partir das características de cada gesto ou objeto que se pretende mapear.

Os gestos ou objetos sempre estão inseridos em um contexto e por isso é necessário realizar um isolamento destes para que não se mapeie também o contexto, aumentando a necessidade de se processar mais dados que o necessário e com um nível de eficiência menor quando realizado de forma isolada. O processo de isolamento dos gestos ou objetos, também chamado de separação do espaço de entrada, pode ser realizado de dois métodos: o supervisionado e o não supervisionado.

No método supervisionado, o classificador, em sua fase de aprendizado, recebe informações de como as classes devem ser identificadas. Pode-se dizer que o sistema de aprendizado supervisionado age sob a supervisão de outro sistema de reconhecimento que identificou anteriormente o gesto e permitirá a construção correta de seu espaço de medida e de sua função discriminante. Durante este processo deve-se modificar os parâmetros que compõem o espaço de medida e permitir um melhor ajuste da função discriminante, objetivando sempre que o sistema possa realizar com mais eficiência o processo de classificação. Ao final, é possível determinar a função discriminante responsável pela separação das diversas classes. Este processo pode ser lento e de elevado custo computacional.

No método não supervisionado, o classificador recebe os gestos desconhecidos e, a partir da medida de diferentes parâmetros (atributos dos gestos presentes na imagem), ele tenta alocá-los em diferentes classes. Entre os métodos que utilizam o treinamento não supervisionado estão a Árvore de Decisão (JONES *et al*., 2001) e o *Boosting* (FREUND *et al*., 1996).

Os dados responsáveis por representar os gestos ou objetos são observados em relação a alguma técnica de busca de informações. Entre estas técnicas estão as que observam propriedades estatísticas dos gestos, como o classificador de *Bayes* (AL-AIDAROOS *et al*., 2012), as que observam as distâncias entre os gestos ou objetos na imagem e suas formas padrões, como as redes neurais artificiais (AHMADI *et al*., 2009), as baseados em informações de texturas dos gestos e a descrição a partir da forma do gesto através de um

dicionário ou uma linguagem básica. Neste último, é definida uma sequência de elementos básicos que representem as formas dos objetos e em seguida é formada uma gramática e construída uma linguagem (NETO, 2000).

Para se buscar um padrão em processamento e análise de sinais e na compressão de dados, Haar (1910) propôs uma função, chamada transformada de Haar. Esta é um caso particular da transformada matemática discreta de *wavelet* (BANG-HUA *et al*., 2007). A *wavelet* é uma função capaz de decompor e descrever outra função, de forma a permitir que esta possa ser analisada em diferentes escalas e sua representação gráfica é dada em forma de um pulso quadrado. Na Figura 2.2 é ilustrada a *wavelet* de Haar.

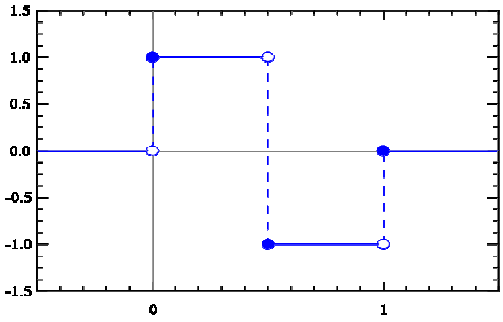


Figura 2.2 *Wavelet* de Haar.
Fonte: (HAAR, 1910).

Jones *et al*. (2001) propuseram um modo de codificação das características relacionadas a forma de um objeto e a existência de contrastes oriundos entre regiões em uma imagem, baseado na transformada de Haar, denominado *Haar-Like Feature*. Um conjunto destas características pode ser usado para codificar contrastes de um rosto humano, uma mão, um carro, etc. Uma vez que as características dos gestos são mapeadas, estas compõem uma classe de gestos. Esta classe de gestos é utilizada como modelo para buscar representantes desta classe em uma imagem.

As características da *Haar-Like* consistem em regiões retangulares, como apresentadas na Figura 2.3, aplicadas ao longo da janela de detecção. Para cada região, calcula-se a soma das intensidades de cada um dos subgrupos (branco e preto), efetuando-se a diferença entre as partes opostas. A diferença entre as partes é usada para categorizar subseções da imagem.

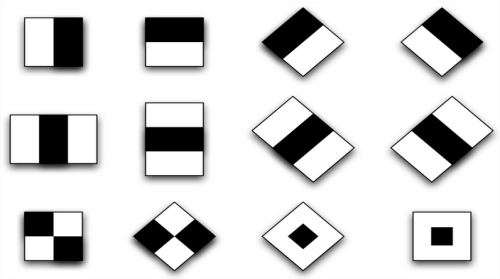


Figura 2.3 *Haar-Like Features*.
Fonte: (JONES *et al*., 2001).

O conhecimento sobre as características dos gestos nas imagens consiste basicamente em descobrir quais dados são úteis para serem armazenados em um modelo baseado nos padrões obtidos e na interpretação dos resultados. Entre as técnicas mais conhecidas e utilizadas encontram-se as árvores de decisão, as regras de classificação, as regras de associação e as redes neurais.

As árvores de decisão e as regras de classificação são consideradas métodos simbólicos que representam, através de expressões, o que é aprendido sobre os atributos dados. As redes neurais são métodos conexionistas, onde o aprendizado consiste em ajustar pesos em uma rede. Devido ao alto grau de legibilidade dos métodos simbólicos, as árvores de decisão e as regras de classificação são muito utilizadas para a geração de classificadores, que por sua vez, são utilizados para atribuir um rótulo a um novo objeto, cuja classe é desconhecida (LIENHART *et al*., 2002).

Uma árvore de classificação busca as características dos gestos nas imagens, onde cada nó é associado a uma medida que representa a relação entre a quantidade de cada classe de objeto no nó da árvore. Neste caso, a árvore representa a classe de gestos e cada nó um objeto mapeado ou uma variação deste. Entretanto, esse modo de busca de características tem como desvantagem ser sensível ao *overfitting* dos dados, que ocorre quando o modelo aprende detalhadamente ao invés de generalizar.

Um modo de aumentar a eficiência de uma árvore de decisão é transformá-la em um *Boosting*. Um *Boosting* é uma árvore de decisão modificada pela inserção de quebras (*splits*), que são efetuadas no conjunto de dados a partir de restrições nos valores de algumas das características. Deste modo, cada nó é convertido numa pequena árvore de classificação com o objetivo de diminuir a combinação entre diferentes classes num mesmo nó.

Um *Boosting* baseia-se na construção de um classificador forte a partir de uma série de classificadores fracos. O termo classificador fraco trata de uma medida de desempenho ao

redor de 50% +1. Ou seja, cada classificador fraco tem apenas a obrigação de ser melhor que o acaso, porém, ao se combinar os resultados de cada classificador fraco, o resultado final é um classificador forte. Em relação ao *Boosting*, pode-se citar em particular o *AdaBoost - Adaptive Boosting*, que estendeu o *Boosting* original para torná-lo adaptativo (LIENHART *et al*., 2002).

Na prática, apenas uma característica não é suficiente para realizar uma detecção com eficiência. Por isso o *AdaBoost* extrai características e as combina em pequenas árvores de decisão, representando em cada nível um classificador fraco. À medida que o algoritmo avança, os classificadores fracos se concentram nos pontos que os anteriores tiveram os piores resultados incrementalmente melhorando a qualidade da resposta final.

O processo de extração de características *Haar-Like* é calculado a partir de um conceito denominado imagem integral. Uma imagem integral é um meio rápido e eficaz de se calcular a soma dos valores dos *pixels* de uma imagem ou de um subconjunto de uma grande retangular nesta imagem, em tempo constante, acelerando o processamento da imagem (JONES *et al*., 2001).

O processo de mapeamento de objetos e gestos tem o uso intensivo de processamento, sendo objeto de estudo da *Intel Research*® para melhorar os seus processadores. Assim, a *Intel*® criou o projeto OpenCV para avançar na pesquisa de Visão Computacional e para otimizar o uso dos processadores em aplicações de uso intensivo de processamento e memória. O OpenCV possui módulos de processamento de imagens, de vídeos, estrutura de dados, álgebra linear, GUI (Interface Gráfica do Usuário), controle de mouse e teclado, além de uma infinidade de algoritmos de visão computacional como filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural, etc. O OpenCV fornece um conjunto de bibliotecas que realizam a construção de um classificador do tipo *AdaBoost*. Os algoritmos são: *Objectmarker*, *CreateSamples* e *Traincascade* (BRADSKI *et al*., 2008).

Objectmarker é responsável por marcar nas imagens positivas os gestos de interesse, criando um arquivo contendo o nome da imagem e as coordenadas da área de marcação. Este arquivo de texto é convertido em um vetor através da ferramenta *CreateSamples*, que ao mesmo tempo padroniza brilho, iluminação, e dimensiona um tamanho de janela para as imagens recortadas do grupo de imagens positivas.

Para se construir um classificador do tipo *AdaBoost* é necessário escolher dois grupos de imagens que são confrontadas para o mapeamento de características do que se deseja

reconhecer: as positivas, que contém o objeto que se deseja mapear e as negativas, que devem conter outros gestos ou objetos menos o que se deseja mapear.

Jones *et al*., (2001) define que cada estágio da cascata deve ser independente dos demais para permitir que se possa fazer a criação de uma árvore e à medida que se desejar aumentar a precisão da mesma, basta ir adicionando mais imagens e mais estágios à árvore.

Traincascade é uma evolução do *HaarTrainning* que gera apenas classificadores do tipo *Haar-like* e confronta as imagens positivas referenciadas pelo vetor criado pelo *CreateSamples* com as imagens negativas, utilizadas como plano de fundo, e tenta definir bordas e outras características. (LIENHART *et al*., 2002) indica que são necessários no mínimo, 14 etapas para iniciar o processo de reconhecimento de algum gesto. O resultado do *Traincascade* pode ser definido como um classificador do tipo LBP ou do tipo *Haar-Like* (WANG *et al*., 2009).

As funções *Objectmarker*, *CreateSamples* e *Traincascade* do OpenCV se preocupam somente com a definição dos grupos de imagens positivas e negativas e o uso destas para a construção do classificador, sem preocupar com a qualidade das imagens utilizadas. Para aumentar o nível de precisão na extração de características, outras técnicas se tornam necessárias para melhorar e padronizar as definições das imagens. Estas técnicas são organizadas em etapas e explicadas a seguir.

2.3.2 Processamento Digital de Imagens e Uso dos Classificadores

Entende-se por Processamento Digital de Imagens (PDI) toda a manipulação de uma imagem digital para facilitar a identificação e a extração das informações contidas nas imagens para posterior interpretação (YOUNG *et al*., 1995).

O PDI é dividido em etapas que lidam especificamente com problemas relacionados ao tratamento e a modificação dos valores contidos nas imagens. Estas etapas são: pré-processamento, segmentação de imagens e extração de características. Estas etapas são melhores descritas a seguir (YOUNG *et al*., 1995).

2.3.2.1 Pré-processamento

Na etapa de pré-processamento ocorrem às tarefas de calibração radiométrica do sistema, correção de distorções geométricas e remoção de ruídos. A calibração radiométrica

das imagens está relacionada às correções das degradações, buscando registrar valores fieis ao brilho e contraste da cena real. Para os seres humanos, a percepção visual do mundo aparenta ser relativamente simples, mas é uma tarefa extremamente complexa. A visão humana realiza uma série de combinações de fatores como cores, texturas, sombras, luminosidades, entre outros elementos para distinguir um objeto. Para um computador, pequenas diferenças de qualquer um desses componentes sobre um gesto representam para o sistema outro gesto, ou seja, um mesmo gesto visto sob uma luminosidade diferente pode ser interpretada de maneira completamente diferente.

As imagens contidas na Figura 2.4 são referentes à mesma mão, porém, sob a influência de brilhos diferentes. Apesar de parecerem diferentes, os seres humanos as interpretam como sendo o mesmo gesto, porém, para um sistema computacional, não se pode tomar a mesma decisão sem que ele passe por uma fase de aprendizado.

Um modo de se padronizar as imagens adquiridas em ambientes com iluminação diferente é aplicar uma equalização de histograma na imagem. Equalizar um histograma significa obter a máxima variância do histograma de uma imagem, obtendo assim uma imagem com o melhor contraste. Contraste é uma medida qualitativa e que está relacionada com a distribuição dos tons de cinza em uma imagem (CONCI *et al*., 2008).

Figura 2.4 Porta sem reflexo de luz e porta com reflexo de luz.
Fonte: PRÓPRIA.

Outra preocupação nesta etapa é a retirada de ruídos causados por inúmeros motivos, como resolução do equipamento utilizado, iluminação, distância do objeto ou gesto em relação a câmera, etc.

Os ruídos podem se apresentar de duas formas: os brancos e os pretos. O ruído branco, conhecido como ruído de sal, tem seu valor de intensidade (quantização) muito alto em relação aos seus vizinhos. O preto, conhecido como ruído de pimenta, tem seu valor muito baixo em relação aos seus vizinhos. Os valores que podem representar um canal de cor podem variar de 0 (preto ou tom mais escuro para aquela cor) até o 255 (branco ou mais claro para

aquela cor). A redução dos ruídos nas imagens pode ser feita principalmente de duas formas: aplicando-se a Transformação Morfológica e/ou a Suavização (*Smooth*). Para cada *pixel* se utiliza até 24 bits de informação, com partição de 8 bits para cada canal de tonalidade. Desta forma, este intervalo de 256 valores corresponde aos possíveis valores, ou intensidades, que devem ser tratados em cada canal de tonalidade (YOUNG *et al*., 1995).

As Transformações Morfológicas têm uma grande variedade de uso, seja para remoção de ruídos, isolamento ou união de elementos, encontrar intensidade de conexão ou buracos entre gestos e nos gestos. Ela é realizada através da aplicação das técnicas de corrosão e dilatação da imagem. Estes dois métodos são muito utilizados quando o nível de ruídos é muito elevado e podem ser utilizados de forma isolada ou combinados.

A outra forma de se retirar os ruídos de sal e pimenta é através da aplicação de Métodos *Gaussianos* de suavização ou alisamento. A técnica mais utilizada é o filtro *Gaussiano* 3x3, que realiza um cálculo de mediana sobre a matriz original e suaviza as diferenças entre os *pixels* vizinhos.

2.3.2.2 Segmentação de Imagens

A segmentação de imagens consiste na extração ou identificação de gestos ou objetos de interesse contidos na imagem, onde o gesto é toda região com conteúdo semântico relevante para a aplicação desejada. Como consequência, quaisquer erros ou distorções presentes nesta etapa se refletem nas demais etapas, de forma a produzir ao final do processo resultados que podem contribuir de forma negativa para a eficiência de todo o processamento.

A segmentação é um processo empírico, adaptativo e procura sempre se adequar às características particulares de cada tipo de imagem e aos objetivos que se pretende alcançar. De um modo geral, as técnicas de segmentação utilizam duas abordagens principais: a similaridade entre os *pixels* e a descontinuidade entre eles.

A técnica baseada em similaridade mais utilizada é a binarização. A binarização de imagens ou *image thresholding* é uma técnica eficiente e simples do ponto de vista computacional, sendo, portanto largamente utilizada em sistemas de Visão Computacional de tempo real, onde é necessário velocidade no processo de extração de características de imagens para serem entregues ao algoritmo de reconhecimento. Este tipo de segmentação é utilizado quando as amplitudes dos níveis de cinza são suficientes para caracterizar os "gestos" presentes na imagem.

Os sistemas de cores mais utilizados são os chamados sistemas 3-cores, que possuem como maiores representantes o RGB, YCbCr e o HSV (TSAMOURA *et al*., 2006). Estes modos de representação implicam em um maior esforço computacional e de armazenamento em relação a outros modos mais simples, como as imagens em tom de cinza ou as binárias, pois cada camada de cor precisa ser processada separadamente. Esta característica encoraja o uso de sistemas de visão binária. Uma imagem binária contém apenas uma camada de cor, na escala de cinza, e seus valores são apenas o 0 – zero e o 1 – um.

As técnicas baseadas em descontinuidade procuram determinar variações abruptas do nível de luminância entre *pixels* vizinhos. Estas variações, em geral, permitem detectar o grupo de *pixels* que delimitam os contornos ou bordas dos gestos na imagem. As descontinuidades podem ser de ponto, de linhas, bordas, onde se aplica uma máscara para destacar o tipo de descontinuidade existente. A técnica de segmentação baseada em descontinuidade mais utilizada é a de detecção de bordas.

Os filtros de detecção de bordas mais aplicados para destaque de gestos ou objetos contidos em imagens digitais são o *Sobel* (VAIRALKAR *et al*., 2013) e o *Canny* (HAN *et al*., 2012). O filtro de *Sobel* consiste num operador que calcula as diferenças finitas, dando uma aproximação do gradiente da intensidade dos *pixels* da imagem. Este filtro é muito útil quando se deseja verificar continuidades internas dos gestos. O filtro de *Canny* é bastante utilizado para o processo de suavização de ruído e localização de bordas externas.

2.3.2.3 Extração de características

Extração de características pode ser definida como sendo a captura das informações que são mais relevantes ao processo de classificação de um dado fornecido. Esta etapa procura extrair características das imagens resultantes da segmentação por similaridade ou descontinuidade através de descritores que permitam caracterizar cada dígito. Estes descritores devem ser representados por uma estrutura de dados adequada ao algoritmo de reconhecimento.

É importante observar que nesta etapa a entrada pode ser uma imagem ou um conjunto de dados, mas a saída é um conjunto de dados descritores correspondentes àquela imagem. Esse conjunto de dados é aplicado não só para o reconhecimento de gestos, mas também para agrupar características semelhantes para o processo de segmentação da imagem, ou seja, a extração de características é uma forma de redução dimensional.

O próximo passo é dar um rótulo para cada um desses grupos de *pixels*. Esta identificação permite que posteriormente se parametrizem os gestos segmentados, calculando para cada região de *pixels* contíguos um parâmetro específico, como área ou perímetro por exemplo.

Após o passo de rotulação, parte-se para a descrição dos atributos. Existem basicamente duas classes de medidas: a) atributos da imagem como um todo (*field Features*), por exemplo número de gestos ou área total de gestos e b) atributos de região (*region Features*) que se referem aos gestos de forma independente, como por exemplo, a área ocupada somente por um objeto, seu perímetro, pela sua forma, etc. Os atributos de região podem ser muito sofisticados, permitindo uma nova separação dos gestos em classes de similaridades, em função dos parâmetros medidos.

Transformar os dados de entrada em um conjunto de características é chamado de extração de características. Se as características extraídas forem escolhidas com cuidado, se espera que esse conjunto traga informações relevantes para se executar uma tarefa. Dois representantes significativos de extração de características e redução dos *pixels* não significativos são o *Motion Detection* e o *Skin Detection*.

O *Motion Detection* é a técnica que consiste em realizar uma separação dos componentes de uma imagem em dois grupos: os que se movimentaram e os que ficaram estáticos. Aos *pixels* estáticos de uma imagem dá-se o nome de *background* e aos que se movimentaram o nome *foreground*. Por esta razão, muitos autores como Migliore *et al*. (2006), Fujita *et al*. (2012) e Murthy *et al*., (2011) chamam esta técnica de *background subtraction*.

Existem duas formas de tratar os *pixels* resultantes do *foreground*: mantem-se somente as diferenças externas da movimentação, que em termos práticos seria equivalente a desenhar as bordas dos gestos que se movimentaram ou manter os *pixels* que pertencem a parte interna dos gestos que se movimentaram. Esta segunda abordagem é conhecida como Mistura *Gaussiana* (JAGADESH *et al*., 2012). O modo de funcionamento da técnica de detecção de movimento é descrito em 3 passos:

- Capturar dois *frames*;
- Comparar as cores dos *pixels* em cada quadro;
- Se a cor for a mesma, substituir pela cor preta, senão manter o novo valor do *pixel*.

O *Skin Detection* é a técnica que consiste em realizar uma separação dos componentes de uma imagem em dois tipos: os que se enquadram em um padrão de cor previamente estabelecido e os que não se enquadram neste padrão. Este padrão de cor é um modelo que descreve um limite de decisão no espaço das cores básicas, contemplando-as até mesmo dentro de aspectos que as modificam como as variações de iluminação, angulação em relação a câmera, etc. (JAGADESH *et al*., 2012).

Dos diversos formatos de imagens existentes, o mais utilizado para realizar a detecção de tons de pele humana é o HSV, formado pelos canais matiz (*Hue*), saturação (*Saturation*) e valor (*Value*) (TSAMOURA *et al*., 2006).

O formato HSV simplifica a limitação da detecção de padrões dos canais de saturação e matiz, para que os tons de pele sejam percebidos independentemente de variações étnicas, ou seja, do valor associado a cor pele. A cor não necessita de métodos específicos para extração, apenas de técnicas para detecção de certos padrões de cores.

O canal matiz representa uma gama de matizes e é representada por um círculo variando de 0 a 360º e tem suas cores variando no vermelho, amarelo e roxo. O canal de saturação descreve o quão puro é uma tonalidade do canal matiz, tomando como referência a cor branca. Por exemplo, uma cor que é totalmente vermelha é totalmente saturada, mas quando se adiciona algum branco a este vermelho, o mesmo começa a se tornar mais pastel e seu tom de vermelho menos saturado. A saturação de uma cor é o seu valor de pureza variando entre 0% e 100%. O último canal define o valor de brilho que uma cor possui, variando entre 0% e 100%. Se uma cor reflete muita luz, diz-se que ela é brilhante. Neste caso seu valor é alto e a medida que se diminui seu valor de brilho, a mesma se torna menos brilhante.

2.3.2.4 Reconhecimento e Interpretação

O processo de reconhecimento é responsável por atribuir um nome a um gesto ou objeto, baseado na informação fornecida pelo seu descritor. A identificação envolve a atribuição de significado a um conjunto de gestos ou objetos reconhecidos. Para sistemas computacionais esta atribuição de nomes aos gestos ou objetos pode ser realizada de duas formas: a primeira envolve um grau relativamente simples de complexidade e age apenas na busca de um padrão de *pixels* que foi previamente apresentado de forma estática, ou em movimento, e o busca em seguida em outras imagens. A outra forma, mais complexa, envolve uma etapa de criação da inteligência do sistema, que pode ser desde uma árvore de

classificação ou até mesmo uma rede neural. Entre os métodos mais simples citam-se o *MeanShift* (COMANICIU *et al*., 1999), (COMANICIU *et al*., 2002) e o *CamShift* (NADGERI *et al*., 2010), (ARAKI *et al*., 2012) e entre os métodos mais complexos estão o *AdaBoost Haar-Like* e o ANN *perceptron* (*Artificial Neural Network*) (BHOWMIK, 2009).

O *MeanShift* (média por deslocamento) é um método não supervisionado e não paramétrico para estimar o gradiente de uma função de probabilidade, tendo dados amostrados, conforme Comaniciu *et al*. (2002), Comaniciu *et al*.(1999). As primeira aplicações descritas do *MeanShift* foram nos campos de reconhecimento de padrões e filtragem de dados, porém, o algoritmo se tornou popular com trabalhos que focaram em agrupamento de dados e localização de modas de uma função de probabilidade. Comaniciu *et al*. (1999) realizou um estudo aprofundado do *MeanShift* e de suavização de imagens e os aplicou na área de Visão Computacional. A principal função do método *MeanShift* é a localização dos máximos locais evidenciando as regiões do espaço de características onde a densidade de pontos é mais alta.

O *CamShift* (*Continuously Adaptive MeanShift*) é um método de rastreamento que encapsula o *MeanShift* em loop variando o tamanho da janela até que ocorra uma convergência. A média de deslocamento em si é uma técnica não-paramétrica robusta utilizada para encontrar o pico em uma distribuição de probabilidade (COMANICIU *et al*., 1999), (COMANICIU *et al*., 2002). O algoritmo de *MeanShift* é modificado para que ele possa lidar com a mudança da dinâmica de distribuição de probabilidade da cor que é tomada a partir das imagens submetidas ao processo.

O *CamShift* começa com a seleção de uma região-alvo definida manualmente por um usuário, sem muito critério ou certeza de que esta área é a melhor a ser utilizada para o rastreio de um gesto ou objeto. Esta incerteza no momento de definir o retângulo para o objeto ou gesto pode ocasionar em erros ou na diminuição da robustez do método, pois se na seleção da região forem incluídas muitas informações do plano de fundo, estas também irão compor o conjunto de informações a serem rastreadas pelo processo de busca.

O algoritmo descritivo do *CamShift* é descrito através das seguintes etapas:

1. Definição da região de interesse inicial, que contém o objeto que se deseja acompanhar;

2. Criação de um histograma de cores da região contendo o objeto;

3. Fazer uma distribuição de probabilidade do quadro usando o histograma de cores;

4. Com base na imagem de distribuição de probabilidade, é buscado o centro de massa da janela usando o *MeanShift*;

5. Centro da janela de busca para o ponto de tomada a partir do passo 4 e a realização de loops do passo 4 até a convergência;

6. Processar o próximo quadro com a posição da janela de pesquisa do passo 5.

Este algoritmo descritivo é representado pelo fluxograma da Figura 2.5, onde a parte cinza corresponde à distribuição do algoritmo *MeanShift*.

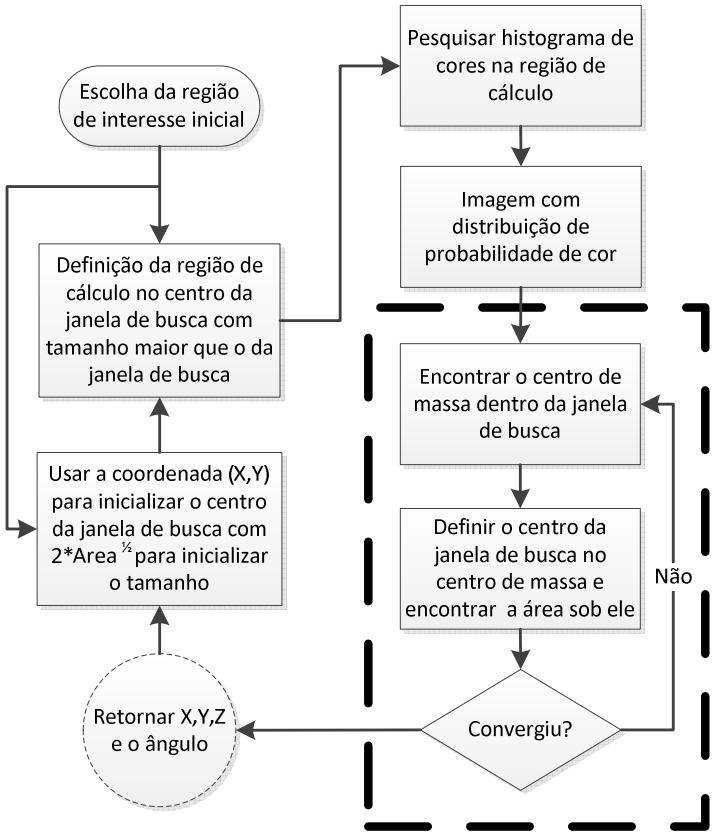


Figura 2.5 Algoritmo *CamShift*.
Fonte: (COMANICIU *et al*., 2002).

Como o *CamShift* necessita acompanhar o gesto a partir de uma imagem colorida, o sistema de cor adotado foi o HSV, retirando o componente de matiz para construir o seu histograma.

A detecção dos gestos utilizando o *AdaBoost Haar-Like* é feita utilizando uma janela de busca através da imagem, verificando se uma região da imagem, em certo local, pode ser classificada como o referido gesto. Em ambientes não controlados, os gestos ou objetos podem ser apresentados a distâncias diferentes das que foram utilizadas para a construção do

classificador. Por esta razão, o *AdaBoost Haar-Like* utiliza um método de escala para modificar o tamanho do detector ao invés de dimensionar a imagem.

É dado um tamanho inicial do detector e depois de cada deslizamento sobre o *frame* inteiro que contem a imagem, a escala do detector é aumentada em α. A escolha do fator α afeta tanto a velocidade do processo de detecção quanto a precisão, fazendo com que seu valor seja cuidadosamente escolhido, a fim de se obter uma melhor relação entre velocidade e eficiência, pois estas são grandezas inversamente proporcionais, ou seja, quanto mais forte for o classificador mais tempo de processamento será exigido para que o mesmo percorra a imagem buscando o gesto mapeado.

Apesar de parecer que o *Haar-Like* executa uma busca exaustiva, a arquitetura interna possibilita uma rejeição precoce com o mínimo de avaliação possível, diminuindo drasticamente o custo computacional. Isso é baseado no fato que a maioria das janelas de detecção tem resposta negativa e apenas poucas conseguem passar por todas as etapas. Portanto, o poder computacional é focado nas janelas que possuem a maior probabilidade de serem positivas, uma vez que já passaram pelos estágios iniciais das árvores de decisão (JONES *et al*., 2001), (FREUND *et al*., 1996) e (LIENHART *et al*., 2002).

Outra característica interessante relacionada ao funcionamento do *AdaBoost Haar-Like* é a independência entre os classificadores e o programa que os utiliza. Desta forma, pode-se utilizar diferentes classificadores simultaneamente em uma mesma imagem. A Figura 2.6 mostra um esquema funcional de rastreamento de um classificador *Haar-Like*.

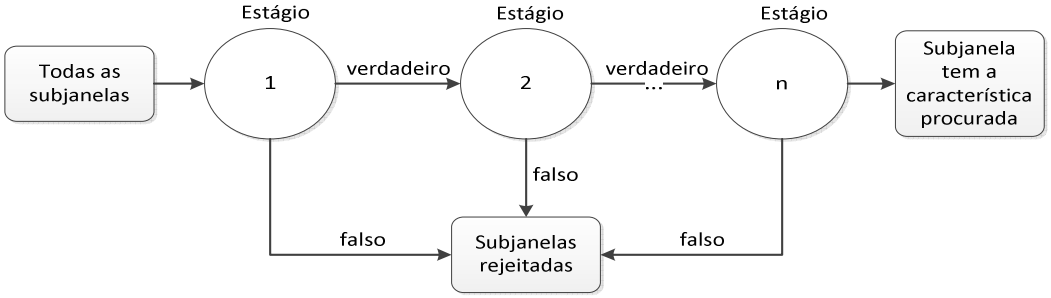


Figura 2.6 Cascata de classificadores.
Fonte: PRÓPRIA.

2.4 Integração da TVDi com a Visão Computacional

Alguns modelos de *Smart* TV (SALUJA *et al*., 2011) estão começando a oferecer a interação gestual com o usuário. Estes equipamentos apresentados no mercado utilizam

soluções proprietárias, fechadas e com custo bastante elevado, necessitando de *softwares* e *hardwares* próprios para poderem ser executados. Em geral, o modo de captura da imagem e de informações sobre os usuários é feita com o uso de câmera infravermelha, que consegue trabalhar em situações adversas de iluminação, associada a sensores de profundidade que passam a percepção de profundidade e distância dos gestos ou objetos percebidos pela câmera.

A grande dificuldade está relacionada ao embarque de aplicativos na TVDi, pois, cada fabricante oferece sua própria solução de *hardware* e *software*, o que influencia e limita o modo de desenvolver aplicativos para estes ambientes. Por exemplo, a fabricante *SAMSUNG*® inseriu no *hardware* de sua *Smart* TV um conjunto de câmeras para receber imagens e as repassar a um aplicativo que as segmenta para buscar gestos que representem algum comando em seu sistema. Já a fabricante LG® permite conectar sua *Smart* TV a um equipamento que se assemelha ao *Xbox*® com o *Kinect*® para realizar a interação gestual (LIANG, 2013).

2.5 Conclusão

Este capítulo serviu para estabelecer as bases sobre Engenharia da Usabilidade, TVDi e Visão Computacional, remetendo as pesquisas anteriores relacionadas ao problema e que ajudaram a compreender a evolução do pensamento sobre a construção de um protótipo de TV com controle gestual.

O capítulo iniciou com uma abordagem sobre Engenharia da Usabilidade, destacando as técnicas adotadas para se modelar e desenvolver produtos centrados no usuário, combinando os principais fatores que participam no processo que são as pessoas, atividades, contextos e as tecnologias, associadas através do *framework* PACT.

Em seguida foram abordados os conceitos sobre o modo de se desenvolver aplicativos interativos para a TVDi em seus ambientes computacionais procedurais e declarativos, destacando as características positivas e negativas de cada um e apresentando os principais *middlewares* utilizados para permitir a execução de *softwares* aplicativos. Os *middlewares* do tipo procedural citados foram o MHP e o GINGA. Estes permitem que aplicativos escritos em Java possam se relacionar diretamente com o *hardware* da TV ou do *set-top box*. O *middleware* declarativo apresentado foi o GINGA-NCL. Este permite que aplicativos escritos

em NCL-LUA possam apenas utilizar as funções existentes sem que se possa manipular o *hardware* de forma direta.

Após a abordagem dada sobre TVDi foram apresentados os conceitos sobre Visão Computacional, separados em duas etapas: a de construção de classificadores que funcionam como a base de conhecimento de um sistema de Visão Computacional e o processamento de imagens que apresentou técnicas para se eliminar os dados desnecessários ao processo de reconhecimento de gestos.

A compreensão desses conceitos, assim como as técnicas e tecnologias abordadas em cada área foi essencial para o desenvolvimento deste trabalho. Assim, um conjunto de trabalhos relacionados ao objetivo principal será apresentado no Capítulo 3 para que suas abordagens sejam comparadas, facilitando e reforçando as escolhas a serem feitas para a concepção da arquitetura de sistema.

Capítulo 3- Trabalhos Relacionados

A seleção dos trabalhos relacionados tem como principal característica os modelos e arquiteturas que propõem soluções para a interação gestual com a TV. As áreas de Engenharia da Usabilidade, TVDi e Visão Computacional foram abordadas separadamente para facilitar o entendimento e as escolhas adotadas para a construção da arquitetura do sistema proposto.

Os trabalhos relacionados à Engenharia da Usabilidade buscaram informações detalhadas sobre as dificuldades de interação dos usuários e seus dispositivos eletrônicos, devido o uso de comandos gestuais e de consistência de interfaces. Os trabalhos relacionados à TVDi realizaram a construção de *softwares* aplicativos que permitiram a interação com as funções internas da TV, disponíveis nos *set-top box* e nos emuladores. Os trabalhos relacionados à Visão Computacional realizaram a construção de classificadores de gestos e tratamento de imagens, necessários para a redução dos *pixels* que não participaram do reconhecimento de gestos. Em seguida são apresentados os trabalhos relacionados à integração da TV e Visão Computacional que buscaram acessar recursos de *hardware* de forma direta em um mesmo equipamento ou indireta através do uso de um computador.

Em cada assunto tratado é apresentada uma tabela com os itens comuns e divergentes, relacionando-os para facilitar a identificação de cada abordagem dada e que são de interesse para a definição deste trabalho.

3.1 Trabalhos relacionados na área de Engenharia da Usabilidade

No trabalho de Brackmann, (2010) foi tratada a interação e as dificuldades dos usuários com o novo conceito de interatividade, principalmente aqueles com mais idade e que precisavam do auxílio de alguém mais esclarecido em relação a dispositivos eletrônicos. Os aplicativos do estudo foram submetidos a uma avaliação qualitativa, observando os aspectos de usabilidade e de satisfação do uso por um grupo de usuários que ao final respondia a uma pergunta: “Você usaria esta proposta de interatividade?”. As respostas apontaram para um resultado que quando não são observados aspectos do modo de uso do usuário final, as

propostas tendem a ser classificadas como mal projetadas e complexas, trazendo assim uma má experiência e que na maioria das vezes resulta em um abandono no uso destes novos recursos.

Em Freeman *et al*. (2001) foi utilizado o método qualitativo de avaliação para identificação de problemas relacionados ao ato de assistir TV identificando as atitudes, emoções e comportamentos associados ao uso de produtos tecnológicos, através de entrevistas. No estudo, os usuários foram selecionados dentro de algumas características como faixa de idade (de 18 a 58 anos), classes sociais e gênero. Os resultados deste estudo trouxeram as percepções dos entrevistados em relação ao uso de novos recursos presentes nas TVDi, revelando um certo grau de confusão sobre os conceitos, nomenclaturas e principalmente sobre o modo de se utilizar o recurso.

Em Barros, (2006) foram tratadas questões relacionadas à consistência de interfaces com usuários em TVDi, considerando os aspectos teóricos e técnicos para a sua construção. Dentre as características abordadas citam-se a consistência de *layout*, utilizando modelos já adotados por diversos fabricantes e a proposição de modelos com a participação dos usuários quando as demandas não forem atendidas pelos modelos existentes.

Em Yi, (2012), uma aplicação utilizando *Kinect*® foi desenvolvida para facilitar a comunicação de pessoas que possuíam deficiência na fala e que necessitavam aprender a linguagem de sinais para realizarem a comunicação. O sistema construído foi capaz de detectar a presença de 9 gestos pré-programados em tempo real. O *Kinect*® foi escolhido por sua capacidade de lidar com condições pouco favoráveis de iluminação devido ao uso de sua câmera infravermelha. A precisão obtida neste sistema foi de 84% a 99% de acerto no reconhecimento dos gestos.

A Tabela 3.1 ilustra a comparação dos trabalhos de referência.

Tabela 3.1.Comparação entre os Trabalhos Relacionados em Engenharia da Usabilidade.
Fonte: PRÓPRIA.

Autor	a)	b)	c)	d)
(YI, 2012)	Sim	Sim	Sim	Sim
(BRACKMANN, 2010)	Sim	Sim	Não	Não
(BARROS, 2006)	Sim	Sim	Não	Não
(FREEMAN *et al*., 2001)	Sim	Não	Sim	Não
Este trabalho	**Sim**	**Sim**	**Sim**	**Não**

Legenda para a Tabela 3.1:

a) Uso de métodos qualitativos;

b) Uso de métodos quantitativos;

c) Tratamentos de usabilidade dos usuários com novos dispositivos de usabilidade;

d) Uso de equipamentos de Visão Estéreo.

Na Tabela 3.1 é mostrada a comparação entre as características dos trabalhos relacionados e o presente trabalho. Os resultados da comparação foram baseados nos métodos qualitativos, quantitativos, de tratamento para as dificuldades com os dispositivos de usabilidade e o uso de equipamentos de visão estéreo. Considerou-se a associação dos dois métodos, o qualitativo e o quantitativo, para a coleta de dados e a observação dos mesmos, pois, enquanto o qualitativo não se preocupa com a representatividade numérica, como o quantitativo, este busca um aprofundamento da compreensão do problema. Considerou-se também a necessidade de se utilizar de regras de usabilidade para se tratar os aspectos que causam dificuldade dos usuários com os novos dispositivos de usabilidade, como a falta de prática por parte dos usuários. Por último, foi observado o modo em que os gestos dos usuários foram capturados pelos sistemas de Visão Computacional, utilizando somente de uma câmera comum.

3.2 Trabalhos relacionados na área de TVDi

Em Lima *et al*. (2013) foi realizada uma abordagem para reduzir a complexidade da construção de uma aplicação NCL. A abordagem aplicada ao trabalho consistiu basicamente na retirada das redundâncias criadas pelo próprio NCL e substituição por códigos próprios. Foi construído um *player* para executar arquivos multimídia de forma mais livre das limitações do *middleware* Ginga-NCL, podendo ser aplicado a outros contextos como os dispositivos móveis e a Web.

Em Brito (2011) foi realizado a concepção de um modelo de referência para o desenvolvimento de artefatos de apoio ao acesso dos surdos aos conteúdos audiovisuais executados na TV. O artefato construído foi um *avatar* que viabilizou a tradução automática do áudio para a linguagem de Libras. O *middleware* NCL foi adotado para indicar o que tocar, onde, como e quando tocar. A definição dos locais de exibição das mídias foi feito por meio

da propriedade de regiões. O aplicativo foi executado no OpenGinga (CAROCA *et al*., 2009) que oferece uma implementação do Ginga-NCL para ser executado em computadores PC.

Em Wu *et al*. (2014) foi realizada a construção de um protótipo de TV centrado na interação familiar e o aparelho de TV. Esta interação foi realizada com o uso da Visão Computacional para permitir que o usuário pudesse realizar o acesso as lojas virtuais, jogos, navegação na *Internet*. O modo adotado para se capturar o gesto e o traduzir para os comandos da *Smart TV* foi o de gravá-lo e aplicar filtros relacionados a Mistura *Gaussiana* e submetê-lo a um classificador que relaciona AdaBoost e Florestas Aleatórias. Um gesto manual foi mapeado, definido pelo autor e, obtido de um grupo de seis pessoas. Cada pessoa apresentou o referido gesto em contextos diferentes como distâncias, iluminação, roupas usadas, etc., gerando vinte mapeamentos para cada usuário.

A Tabela 3.2 mostra a comparação entre os trabalhos de referência.

Tabela 3.2.Comparação entre os Trabalhos Relacionados em TVDi.
Fonte: PRÓPRIA.

Autor	a)	b)	c)	d)
(LIMA *et al*., 2013)	Sim	Sim	Não	Não
(BRITO, 2012)	Não	Sim	Sim	Sim
(WU *et al*. 2014)	Não	Sim	Não	Sim
Este trabalho	**Sim**	**Sim**	**Sim**	**Sim**

Legenda para a Tabela 3.2:

a) Retirada de redundâncias do código;
b) Mapeamento do controle remoto;
c) Acesso de recurso de *hardwares*;
d) Uso da aplicação em tempo real.

Na Tabela 3.2 é realizada a comparação entre as características dos trabalhos relacionados e o presente trabalho. Os resultados da comparação foram baseados na retirada de redundâncias do código, mapeamento do controle remoto, acesso de recurso de hardwares através de redes de computadores e o uso da aplicação em tempo real. Considerou-se a retirada das redundâncias de código geradas pelo Ginga-NCL, substituindo parte dos comandos nativos por programação mais livre do *middleware*, fazendo com que o protótipo possa ser utilizado, em tempo real, com o mínimo de adaptações em outros *hardwares* para

avaliar o desempenho ou quando houver disponibilidade de um dispositivo que aceite a conexão direta com uma câmera. O mapeamento do controle remoto deve ser realizado para que seja possível relacionar os comandos do *middleware* a qualquer dispositivo de entrada de dados, como um sistema de reconhecimento de gestos. A independência do código em relação ao *hardware* também é necessária para que o processamento de imagens, que consome muito processamento e memória, possa ser realizado em um computador acessado através de redes de computadores.

3.3 Trabalhos relacionados na área de Visão Computacional

Os trabalhos relacionados à Visão Computacional foram divididos em duas etapas distintas: a de construção de classificadores, que tratou das formas de se fornecer o conhecimento sobre o que se deve buscar nas imagens e a de processamento de imagens, que tratou das aplicações dos filtros responsáveis em eliminar os ruídos.

3.3.1 Construção de Classificadores

O trabalho de Wilson *et al*. (2009) definiu a construção de seu classificador *Haar-Like* utilizando como base os valores dos estudos de Viola-Jones (JONES *et al*., 2001), onde são indicadas as quantidades de 10 mil imagens contendo o gesto ou objeto que se deseja mapear e 10 mil imagens contendo imagens diversas e que não tenham o objeto que se pretende mapear. O tamanho das imagens utilizadas foi de 320x240 *pixels* e o tamanho da janela de busca das características *Haar-Like* foi definida em 24x24.

O trabalho de Rautaray *et al*. (2012) descreveu a construção de um detector de gestos das mãos, utilizando câmeras de baixo custo, para substituir o *mouse* na interação com os objetos virtuais criados a partir do *framework Open GL Library*, baseado no classificador *Haar-Like* associado ao algoritmo de *CamShift*. Para este trabalho foram utilizadas 6 câmeras que trabalhavam de forma independente entre si e buscavam identificar os gestos apresentados junto aos classificadores *Haar-Like*. Para cada gesto mapeado foram utilizadas 12.000 imagens. O tamanho das imagens utilizadas foi de 320x240 *pixels* e o tamanho da janela de busca das características *Haar-Like* foi definida em 20x20. A estratégia adotada foi a de utilizar 6 câmeras distantes 10 cm uma da outra apontadas para um ponto central, onde o

usuário deveria se posicionar, permitindo que o software utilize o gesto reconhecido com o maior grau de acerto.

O trabalho de Dardas *et al*. (2007) tratou do reconhecimento de gestos da mão usando características *Haar-Like* e uma gramática livre de contexto. O trabalho utilizou conceitos de autômatos para definir uma máquina de reconhecimento dos *strings* dos binários das imagens submetidas ao algoritmo baseado em padrões definidos via programação. Para este trabalho foram utilizadas 420 imagens para mapear cada gesto e 500 imagens para servirem e plano de fundo na construção do classificador. O tamanho das imagens utilizadas neste trabalho foi de 320x240 *pixels* e o tamanho da janela de busca das características *Haar-Like* foi definida em 20x20.

A Tabela 3.3 mostra a comparação entre os trabalhos de referência.

Tabela 3.3. Comparação dos trabalhos relacionados para construção dos classificadores.
Fonte: PRÓPRIA.

Autor	a)	b)	c)
(WILSON *et al*., 2009)	320x240	24x24	*Haar-Like*
(RAUTARAY *et al*., 2012)	320x240	20x20	*Haar-Like*
(DARDAS *et al*., 2007)	320x240	20x20	*Haar-Like*
Este trabalho	**800x600**	**24x24**	***Haar-Like***

Legenda para a Tabela 3.3:

a) Tamanho da imagem utilizada;
b) Tamanho da janela de rastreamento;
c) Modelo de classificador.

Na Tabela 3.3, é realizada a comparação entre as características dos trabalhos relacionados e o presente trabalho. Os resultados da comparação foram baseados na quantidade de imagens utilizadas, no tamanho da imagem e da janela de rastreamento e no modelo de classificador adotado. O tamanho da imagem considerada para o protótipo busca facilitar o trabalho do classificador *Haar-Like* que procura por características quadradas contidas nas imagens, pois quanto maior for a quantidade de *pixels*, mais os limites deste são destacados. A janela de busca considerada possui dimensão de 24x24 que processa a cada instante cerca de 2300 *pixels*, o que para os processadores atuais não causa grande impacto. Abaixo deste tamanho, a janela de busca poderá perder tempo de processamento analisando

informações irrelevantes como ruídos ou *pixels* que tem seus valores próximos aos procurados e acima deste tamanho, informações contidas em *pixels* menores podem passar despercebidas.

3.3.2 Pré-processamento

O trabalho de Jawas *et al*. (2013) apresentou uma técnica de preenchimento uniforme de regiões entre bordas de objetos inseridos em imagens através de operações morfológicas (YOUNG *et al*., 1995) e suavização *Gaussiana* de tal maneira que, restem na imagem final apenas os valores das bordas com os *pixels* internos destacados e os demais valores eliminados. As propriedades de textura e segmentação precisa da estrutura dos objetos, além da coloração daqueles destacados foram os objetos de estudo. O resultado obtido reduziu as áreas desconhecidas ou com pouco destaque através da aplicação dos filtros de erosão e suavização, além de realce das áreas percebidas com o uso do filtro de dilatação.

O trabalho de Napoleon *et al*. (2013) apresentou uma técnica de tratamento de imagens digitais através de transformação morfológica, assistida por um avaliador humano. O objetivo da técnica era melhorar a qualidade das imagens, obtidas de sensoriamento remoto e que apresentavam ruídos de sal (pigmentação branca dispersa) e ruídos de pimenta (pigmentação preta dispersa) (YOUNG *et al*., 1995). Os resultados obtidos pela aplicação dos filtros foram imagens sem ruídos, que puderam ser analisadas de forma mais precisa pelos observadores humanos.

O trabalho de Silva *et al*. (2013) apresentou a técnica de tratamento de imagens digitais através da aplicação de filtros de erosão, dilatação e suavização (*smooth*) para melhorar o nível de acerto em um protótipo que identifica pessoas pilotando motocicletas sem capacete. A imagem resultante do tratamento de retirada de ruídos foi entregue a um classificador construído com o padrão LBP (*Local Binary Pattern*) para que o processo de reconhecimento fosse realizado. O classificador foi construído a partir de um conjunto de 5 mil imagens de pessoas pilotando motos sem capacete a uma distância controlada (cerca de 20 metros da câmera) e com condições variáveis de iluminação.

A Tabela 3.4 mostra a comparação dos trabalhos de referência para a aplicação dos filtros de erosão, dilatação e suavização.

Tabela 3.4. Comparação entre os trabalhos relacionados para extração de características.
Fonte: PRÓPRIA.

Autor	a)	b)	c)
(JAWAS *et al.*, 2013)	Sim	Sim	Sim
(NAPOLEON *et al.*, 2013)	Sim	Sim	Não
(SILVA *et al.*, 2013)	Sim	Sim	Não
Este trabalho	**Sim**	**Sim**	**Sim**

Legenda para a Tabela 3.4:

a) Filtros de erosão;

b) Filtro de dilatação;

c) Filtro de suavização.

Na Tabela 3.4 é realizada a comparação entre as características dos trabalhos relacionados e o presente trabalho. Os resultados da comparação foram baseados no uso dos filtros de erosão, dilatação e suavização. Considerou-se o uso combinado dos filtros de erosão e dilatação para destacar os objetos inseridos nas imagens, sem eliminar *pixels* necessários para a identificação destes objetos. Considerou-se também a aplicação do filtro de suavização para aproximar os valores das intensidades dos *pixels* restantes nas imagens, descartando os demais como ruídos de sal e pimenta.

3.3.3 Segmentação

O trabalho de Vairalkar *et al*. (2013) tratou da aplicação do filtro *Sobel* como estratégia para filtrar informações inúteis preservando as estruturas importantes presentes nas imagens. O trabalho aplicou o filtro *Sobel* com janela de 3x3 por ter um número menor de bordas falsas resultantes.

O trabalho de Han *et al*. (2012) tratou de um estudo sobre esteganografia, que é a arte de esconder informação digital de tal forma que não se possa suspeitar de sua existência. A estratégia abordada no trabalho foi utilizar um método baseado na detecção de borda utilizando o filtro *Canny*. Para se aplicar o método foram realizadas as tarefas de eliminação de ruídos através de suavização das imagens, busca de valores de destaque através de

magnitudes em histogramas, binarização das imagens e o descarte de todos os *pixels* que não se conectassem.

O trabalho de Davoodianidaliki *et al*. (2013) tratou em sua pesquisa da relação do filtro de suavização *Gaussiana* com os filtros de *Canny* e *Sobel* para transformar o método de detecção de bordas em um modelo adaptativo. No processo, o autor buscou os valores dos *pixels* formadores das bordas através do filtro de *Canny* e para preencher os buracos nas bordas utilizou o filtro de *Sobel*. Esta forma adotada apresentou um processamento de baixo nível de consumo de processamento e memória e destacou os objetos em relação aos *pixels* vizinhos as bordas, facilitando a aplicação da técnica de suavização, responsável por aproximar os valores dos *pixels* e diminuir as diferenças do fator de luminosidade. Assim, a medida que o autor reaplicava a detecção das bordas na imagem resultante, suavizava ainda mais os demais pixels até que obtivesse uma imagem com os objetos bem destacados e o plano de fundo sem definição.

A Tabela 3.5 mostra a comparação entre os trabalhos de referência em relação à aplicação dos filtros de borda para realizar a segmentação nas imagens.

Tabela 3.5. Comparação entre os trabalhos relacionados a Segmentação.
Fonte: PRÓPRIA.

Autor	a)	b)	c)	d)
(HAN *et al.*, 2012)	Não	Sim	Não	Sim
(VAIRALKAR *et al.*, 2013)	sim	Não	Sim	Sim
(DAVOODIANIDALIKI *et al.*, 2013)	Sim	Sim	Não	Sim
Este trabalho	**Sim**	**Sim**	**Sim**	**Sim**

Legenda para a Tabela 3.5:

a) Filtros de *Sobel*;
b) Filtro de *Canny*;
c) Fator aplicado ao filtro 3x3;
d) Preservar os gestos ou objetos.

Na Tabela 3.5 é realizada a comparação entre as características dos trabalhos relacionados e o presente trabalho. Os resultados da comparação foram baseados no uso dos filtros de *Sobel* e *Canny*. Considerou-se o uso combinado dos filtros de detecção de bordas para destacar os *pixels* pertencentes aos objetos inseridos nas imagens, ligados a seus

vizinhos. Enquanto o filtro de *Sobel* busca a relação de vizinhança entre *pixels* mais discretos e com poucas ligações o *Canny* destaca os *pixels* mais fortes, principalmente aqueles que indicam as bordas externas dos objetos.

3.3.4 Motion Detection

O trabalho de Migliore *et al*. (2006) fez uma avaliação do processo de diferença entre *frames* para se realizar uma detecção de movimento em aplicações de vigilância de vídeo. A diferença de quadros permitiu obter uma imagem final com os *pixels* resultantes da diferença entre dois *frames* consecutivos. Para obter um maior grau de precisão, o protótipo foi construído utilizando imagens coloridas de formato RGB de 640x480 *pixels* através de uma câmera de velocidade de 15 *frames* por segundo. Neste trabalho foi utilizada Mistura Gaussiana como técnica de detecção de movimento principal e as bordas externas dos objetos que se movimentaram foram detectadas através da técnica de subtração de imagens obtidas de forma sequencial.

O trabalho de Fujita *et al*. (2012) apresentou um método de detecção de movimento para um robô identificar os gestos que eram realizados em uma área pré-determinada e auxiliar as pessoas na execução da tarefa de coleta de objetos que foram lançados nestas regiões. O processo adotado utilizou imagens em formato RGB de 480x320 *pixels* através de uma câmera de velocidade de 30 *frames* por segundo e a detecção dos movimentos através da diferença entre *frames*. A velocidade do sistema foi um fator decisivo para que determinasse a resolução da câmera e a dimensão da imagem.

O trabalho de Jagadesh *et al*. (2012) utilizou como estratégia para se realizar a detecção dos movimentos a Mistura *Gaussiana*. Neste método, os *pixels* resultantes eram os que compunham as partes internas dos objetos que se movimentaram. Esta técnica serviu para se realizar uma etapa de detecção de tons de pele com os *pixels* resultantes, o que não é possível com o uso da técnica de subtração simples entre imagens, pois o resultado é apenas o dos *pixels* de borda. O autor optou em realizar uma combinação de imagens de formatos RGB e HSV de 480x320 *pixels*. Esta combinação melhorou o índice de localização de pessoas, porém, aumentou o tempo necessário para o processamento das imagens.

A Tabela 3.6 mostra a comparação entre os trabalhos de referência quanto à aplicação dos filtros para detecção de movimentos.

Tabela 3.6. Comparação entre os trabalhos relacionados à Detecção de Movimentos.
Fonte: PRÓPRIA.

Autor	a)	b)	c)	d)
(MIGLIORE *et al.*, 2006)	Sim	Sim	640x480	HSV
(FUJITA *et al.*, 2012)	Sim	Não	480x320	RGB
(JAGADESH *et al.*, 2012)	Não	Sim	480x320	RGB, HSV
Este trabalho	**Sim**	**Sim**	**800X600**	**HSV**

Legenda para a Tabela 3.6:

a) *Motion Detection Diff*;

b) *Motion Detection* através de Mistura *Gaussiana*;

c) Tamanho da imagem;

d) Formato da imagem.

Na Tabela 3.6 é realizada a comparação entre as características dos trabalhos relacionados e o presente trabalho. Os resultados da comparação foram baseados no uso dos detectores de movimento baseados na diferença entre frames e na detecção de movimentos através da técnica da Mistura *Gaussiana*, observando o tamanho e o formato das imagens adotadas em cada trabalho. Considerou-se o uso combinado dos filtros de detecção de movimento, pois enquanto o detector de movimento por diferença de *frames* consegue perceber os *pixels* da borda externa dos objetos que se movimentam, o filtro de Mistura *Gaussiana* percebe os *pixels* internos, aumentando a possibilidade de se ter o objeto por inteiro representado ao final da detecção. Como busca-se detectar objetos que se movimentaram a distâncias maiores que as executadas nos trabalhos relacionados ou com maior riqueza de detalhes, considera-se o uso de imagens maiores e em formato HSV, que tem como característica destacar as características de brilho dos objetos.

3.3.5 Skin Detection

O trabalho de Ahmadi *et al*. (2009) apresentou uma abordagem de classificação de *pixels* baseados na cor de pele usando uma rede neural simétrica. Os classificadores neurais do tipo simétrico usam um único neurônio de saída ou a combinação de redes neurais separadas para cada uma das classes de pele e não pele. Os classificadores utilizam como base

inicial um valor de segmentação com valor mínimo e máximo de tons de pele nos componentes de uma imagem em formato HSV. A rede neural realizou um exaustivo treino nas imagens que foram submetidas para que fosse capaz de definir limiares ótimos para cada imagem, o que reduz drasticamente a quantidade de falsos positivos.

O trabalho de Jagadesh *et al*. (2012) utilizou como estratégia para se realizar a detecção dos tons de pele a aplicação de filtros de mínimo de máximo dentro do grupo de *pixels* resultantes da detecção de movimentos da técnica de Mistura *Gaussiana*. Este trabalho preocupou-se com a estimativa de uma função de probabilidade da cor da pele humana utilizando um modelo de Mistura *Gaussiana* finito cujos parâmetros foram definidos através de testes empíricos.

O trabalho de Bang-Hua *et al*. (2007) foi desenvolvido para se fazer a indexação dos conteúdos de vídeos baseados em detecção e o reconhecimento de rostos humanos. O protótipo foi construído levando em conta diferenças nos fatores de iluminação e tonalidades do que deveria ser considerado pele através dos padrões de imagens HSV e YCbCr para se obter regiões da cor de pele quantizados.

A Tabela 3.7 mostra a comparação entre os trabalhos de referência quanto à aplicação dos filtros para detecção de tons de pele.

Tabela 3.7. Comparação entre os trabalhos relacionados a detecção de tons de pele.
Fonte: PRÓPRIA.

Autor	a)	b)	c)
(AHMADI *et al*., 2009)	Sim	Rede neural	HSV
(JAGADESH *et al*., 2012)	Sim	Filtro de mínimo e máximo de cores	RGB
(BANG-HUA *et al*., 2007)	Sim	Filtro de mínimo e máximo de cores	HSV e YCbCr
Este trabalho	**Sim**	**Filtro de mínimo e máximo de cores**	**HSV**

Legenda para a Tabela 3.7:

a) Tratamento para variação de luminosidade;
b) Técnica de segmentação de tons de pele;
c) Formato da imagem;

Na Tabela 3.7 é realizada a comparação entre as características dos trabalhos relacionados e o presente trabalho. Os resultados da comparação foram baseados no uso de técnicas de tratamento para variação de luminosidade, técnica de segmentação de tons de pele

adotada e o formato adotado para exibir a imagem. Considerou-se o uso tratamento do fator de luminosidade para uniformizar a percepção visual dos tons de pele, pois a mesma pode ser classificada de formas diferentes se for submetida a fatores de iluminação diferentes. Outro ponto considerado é a segmentação das imagens por filtros de mínimo e máximo em cada componente de cor, eliminando os componentes isoladamente ou combinados que não representam um tom de pele. O formato da imagem considerado para este trabalho é o HSV para aproveitar a imagem construída na fase de detecção de movimentos e também por este formato ter como característica destacar o brilho dos objetos.

3.3.6 CamShift

O trabalho de Nadgeri *et al*. (2010) utilizou o filtro de *CamShift* sobre imagens de formato *HSV* para localizar movimentos das mãos em ambientes onde não se tinha o controle de iluminação e da distância em que os gestos eram apresentados em relação a câmera. O trabalho buscou eliminar os planos de fundo através da associação dos componentes *Hue* das imagens *HSV*, construindo um componente *Hue* para um grupo de imagens, padronizando a localização e o acompanhamento das mãos.

O trabalho de Araki *et al*. (2012) associou o *CamShift* a um método de detecção de movimentos para reduzir o conjunto de dados que são submetidos ao processo de busca e ter um aumento de desempenho. O protótipo utilizou uma câmera comum do tipo RGB (*Red, Green, Blue*) e obteve velocidades de 21 *frames* por segundo utilizando janelas de busca de 30x40.

A Tabela 3.8 mostra a comparação entre os trabalhos de referência quanto a aplicação da técnica de *CamShift* para detecção de gestos ou objetos.

Tabela 3.8. Comparação entre os trabalhos relacionados ao uso do *CamShift*.
Fonte: PRÓPRIA.

Autor	a)	b)	c)
(NADGERI *et al*., 2010)	Não	Não	HSV
(ARAKI *et al*., 2012)	Sim	Sim	RGB
Este trabalho	**Sim**	**Sim**	**HSV**

Legenda para a Tabela 3.8:

a) Tratamento para variação de luminosidade;

b) Tratamento para variação da distância onde o gesto foi apresentado;

c) Formato da imagem.

Na Tabela 3.8 é realizada a comparação entre as características dos trabalhos relacionados e o presente trabalho. Os resultados da comparação foram baseados no uso de técnicas de tratamento para variação de luminosidade, tratamento para a variação da distância onde o gesto foi apresentado e o formato de imagem adotado. Considerou-se o uso tratamento do fator de luminosidade para uniformizar a percepção visual dos tons percebidos pelo histograma criado pelo *CamShift* para aumentar a precisão do uso da técnica. Outro ponto considerado é o tratamento referente à variação da distância onde o gesto é apresentado através do redimensionamento da janela de busca referente ao histograma. Este tratamento busca suprir a carência do sistema em fornecer informações tridimensionais dos objetos, pois utiliza apenas uma câmera. O formato da imagem considerado para este trabalho é o HSV para aproveitar a imagem construída na fase de detecção de movimentos e modificada na fase de detecção de tons de pele, além de também ser o formato que tem como característica destacar o brilho dos objetos.

3.4 Trabalhos relacionados a Integração da Engenharia da Usabilidade, a TVDi e a Visão Computacional

O trabalho de Devasena *et al.* (2013) foi desenvolvido para controlar computadores e TVs. Para o caso específico das TVs, o objetivo foi o de substituir o controle remoto na execução de arquivos de mídia nas tarefas de mudança de canal e de volume do som. O modelo proposto foi construído em duas etapas: a de Visão Computacional e a de controle da TV. Na primeira etapa, a de Visão Computacional, foi adotado o uso de marcadores coloridos sobre as pontas dos dedos para facilitar a segmentação das imagens e construção dos classificadores. O classificador construído foi a seleção de uma imagem para cada gesto pretendido e o reconhecimento a comparação entre esta imagem modelo e as imagens apresentadas para a câmera. Para eliminar os elementos da imagem submetida que não pertenciam ao gesto mapeado foi aplicado um processo de detecção de movimento através da

Mistura *Gaussiana*. Como modo de integração com a TV foi projetada a construção de um dispositivo que receberia a interpretação do gesto, gerado através de um *Smart phone* para a TV. O protótipo foi testado em computadores para abrir e fechar aplicativos dispostos em suas áreas de trabalho apresentando uma perda muito grande (em torno de 40%) em ambientes controlados e perdas maiores em locais onde não se tinha o controle sobre a iluminação.

O trabalho de Vatavu, (2012) foi construído para confrontar o uso do controle remoto com novos modelos de comando da TV através de gestos. O protótipo foi construído utilizando a câmera comum e imagens de dimensão 640x480. As imagens foram processadas em um computador através de um *software* escrito em C++, pois não foi possível embarcar diretamente no equipamento de TV. O resultado do protótipo conseguiu realizar as operações básicas de ajuste do volume e de troca de arquivos de mídia previamente cadastrados em um *media player*, porém com uma perda de gestos em torno de 40% devido a grande quantidade de falsos positivos causados por objetos dispostos na cena e que confundiam o algoritmo de VC. Os gestos utilizados foram definidos por um grupo de pessoas através de uso de técnicas de Engenharia da Usabilidade.

O trabalho de Miranda *et al.* (2009) prospectou um modelo de interação com a TV baseada em gestos. Os gestos eram capturados através de marcadores utilizados junto ao usuário como dedais, luvas ou anéis coloridos, onde cada cor representava uma ação específica para a TV. A escolha dos gestos, bem como a relação destes com a TV foram definidos utilizando práticas participativas de *brainstorm* em reuniões com um grupo de usuários. Outra razão para a utilização de marcadores coloridos foi devido a diminuição do esforço necessário para a segmentação e detecção em uma imagem com uso de classificadores simples.

A Tabela 3.9 mostra a comparação entre os trabalhos de referência quanto a integração da Engenharia da Usabilidade, TVDi e Visão Computacional.

Tabela 3.9.Comparação entre os trabalhos relacionados a integração da Engenharia da Usabilidade, TVDi e VC. **Fonte**: PRÓPRIA.

Autor	a)	b)	c)	d)	e)	f)	g)
(DEVASENA *et al.*, (2013)	Não	Não	Não	Sim	Sim	Sim	Sim
(VATAVU, 2012)	Sim	Sim	Sim	Sim	Sim	Não	Não
(MIRANDA *et al.*, 2009)	Sim	Não	Não	Não	Sim	Sim	Não
Este trabalho	**Sim**	**Sim**	**Sim**	**Não**	**Sim**	**Não**	**Sim**

Legenda para a Tabela 3.9:

a) Aplicação de regras de usabilidade;
b) Execução do Volume do som;
c) Execução do Canal de programação;
d) Execução de outras funções na TVDi;
e) Uso de Visão Computacional;
f) Uso de marcadores junto ao usuário;
g) Uso de dispositivos de rede.

3.5 Conclusão

Neste capítulo, foram apresentados trabalhos relacionados para nortear a construção da hipótese. Sobre estes trabalhos relacionados, foi realizada uma análise das diferentes propriedades de cada uma das abordagens dadas pelos autores, apresentando-as em tabelas comparativas para que se incorporem várias características importantes na proposta deste trabalho. A função destas tabelas foi o de facilitar na escolha das características que farão parte da proposta deste trabalho para que não se crie algo simplesmente diferente daquilo que já existe, mas algo que incorpore várias características importantes em uma mesma proposta.

Na primeira abordagem, foi realizada uma análise sobre a Engenharia da Usabilidade, principalmente no que se refere aos métodos avaliativos qualitativos e quantitativos. Na segunda abordagem, foi realizada uma análise sobre TVDi referente a construção do *layout* e do modo de realizar a interação com o menu de opções da TV. Na terceira abordagem, foi realizada uma análise sobre a VC nos assuntos referentes a construção de classificadores e ao processamento digital das imagens que seriam submetidas ao reconhecimento. Finalmente, na quarta abordagem, foi realizada uma análise em relação à integração da Engenharia da Usabilidade com a TVDi e a VC.

Os trabalhos expostos mostram que a maioria dos desenvolvimentos de aplicativos voltados à interação utiliza a técnica qualitativa para iniciar o processo de análise, o método estatístico para mapear gestos ou objetos, a detecção de movimentos através da diferença entre *frames* e modelos pré-definidos de tons de pele para a detecção de pele e a integração dos *softwares* de TV e de VC é demonstrada em emuladores.

Em relação à forma de se aumentar a possibilidade de aceitação do aplicativo por parte dos usuários, será realizada a associação das técnicas qualitativa e quantitativa que dará maior confiança nas informações utilizadas para a construção da arquitetura do sistema. Em relação à diminuição do custo computacional de processamento das imagens serão realizadas associações das técnicas de eliminação de ruídos com o uso da transformação morfológica e suavização, detecção de movimento por diferença de *frames* e Mistura *Gaussiana* e detecção adaptativa de tons de pele em lugares que sofram mudanças de luminosidade.

A compreensão dessas análises realizadas em cada área é essencial para o desenvolvimento de um modelo teórico, relacionando-as de forma lógica e científica, detalhando os procedimentos metodológicos, de modo a esclarecer a articulação entre os conceitos apontados como relevantes para a construção do protótipo.

Assim, o relacionamento entre as decisões tomadas sobre as áreas de TVDi para a construção de um protótipo que permita a interação com o ajuste do volume do som e a troca de canais de programação, da Visão Computacional para construir classificadores de gestos que possam ser aplicados aos cenário onde se utilizam as TVs e Engenharia de Usabilidade, que trouxe o usuário para o processo de construção e de *layouts* e demais elementos necessários a interação da TV com a Visão Computacional. Este relacionamento é consolidado em uma arquitetura de sistema e apresentada no Capítulo 4.

Capítulo 4- Concepção da Arquitetura

Este capítulo apresenta a arquitetura proposta para a interação gestual com a TVDi através de uma análise em alto nível para garantir que os requisitos tenham consistência, sejam completos, corretos e operacionalmente definidos.

A forma adotada foi a de subsistemas para facilitar suas definições, minimizando a comunicação entre os módulos, diminuindo a redundância de códigos e recursos e eliminar as tomadas de decisões na fase de construção dos protótipos, descrita no Capítulo 5. Constituem a arquitetura de sistema a Engenharia da Usabilidade, a TVDi e a Visão Computacional.

O capítulo termina com uma visão geral dos modelos definidos em cada subsistema para facilitar o processo da busca de tecnologias que atendam os requisitos e permitam a construção dos protótipos que realizam a prova de conceito.

4.1 Concepção da Proposta

O modo de interação do usuário com a TVDi, proposto neste trabalho, foi a própria mão do usuário para a execução das funções da TVDi através de reconhecimento de gestos capturados por uma câmera.

A arquitetura do sistema, apresentada na Figura 4.1, mostra que as modelagens de TVDi e VC são independentes entre si, permitindo que o protótipo de TVDi e VC funcionem isoladamente em outros contextos e vinculadas através da passagem dos valores correspondentes a ação que deve ser realizada e a coordenada onde o referido gesto foi localizado na imagem original gerados pelo protótipo de VC. Nesta arquitetura, cada módulo é responsável por funcionalidades específicas e bem definidas que se comunicam por um módulo independente denominado Relacionamento.

O fluxo dos dados, que vai da câmera até o dispositivo de saída de TV, passa primeiramente por uma etapa de processamento de imagens, que aplica uma série de filtros para eliminação de ruídos e para destacar os *pixels* que compõem os objetos. Os objetos encontrados nas imagens são comparados com os classificadores no sub módulo de reconhecimento que, em seguida, gera um documento contendo os valores correspondentes à

ação que deve ser realizada e a coordenada onde o referido gesto foi localizado na imagem original.

Usando um dispositivo de rede que permite a conexão entre o módulo de Visão Computacional e a TVDi, os valores gerados pela fase de reconhecimento são acessados através de um sub módulo de ação, pertencente ao módulo de TVDi. Este sub módulo de ação verifica os valores recebidos pela Visão Computacional e os relacionam aos comandos dos *middlewares* de TV.

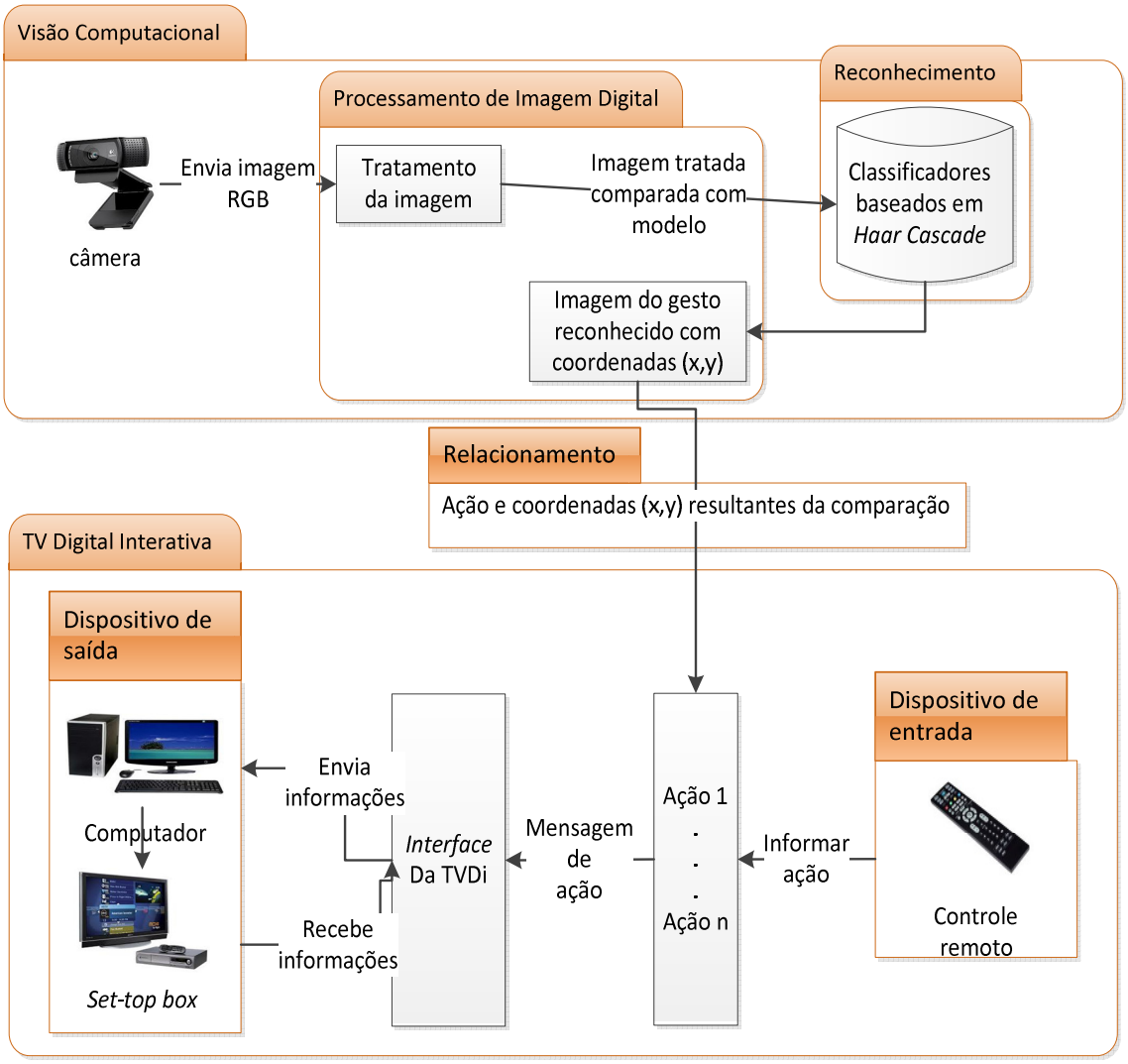


Figura 4.1 Arquitetura do modelo do sistema.
Fonte: PRÓPRIA.

Nos tópicos seguintes, a arquitetura dos módulos de Visão Computacional e de TVDi serão apresentados, usando um cenário de uso, que descreve como as técnicas foram desenvolvidas e combinadas para realizar o reconhecimento dos gestos na TVDi.

4.2 Cenários de uso Aplicados à Arquitetura

Para se desenvolver a arquitetura do sistema foi necessário pensar na convivência e na interatividade do usuário com o seu televisor e observar as diversas características das TVs, seja do ponto de vista de *hardware*, de *software*, além das atitudes do usuário de TV para criar uma interface útil e agradável.

A interação do usuário com uma TV ocorria dentro de dois cenários: um físico, que tratava das características do local onde a TV está localizada e um lógico, que tratava do sistema em si e que permitia que as escolhas realizadas pelo usuário através do elemento de interação, seja este um controle remoto ou um gesto, fossem de fato executados na TV.

Assim, deveriam ser observados os requisitos referentes a estes cenários para definir o *layout* do aplicativo e o conjunto de gestos para interagir com a TV, além das estratégias de manipulação de imagem para se realizar um reconhecimento de gestos mais preciso.

O *framework* PACT auxiliou no processo de elicitação, fazendo uso das informações de domínio do problema, buscando identificar as particularidades do que se desejava desenvolver e delimitando o problema.

4.3 Engenharia da Usabilidade – aplicação do PACT

Como o objetivo principal era investigar e propor a concepção de uma interface para controlar a TVDi através de um conjunto de gestos, foi necessário se conhecer que tarefas eram controladas neste dispositivo.

As normas técnicas da TV digital brasileira (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 15607:2008, 2008) mostrou que existiam 8 tarefas que englobavam as funcionalidades mais simples e as mais avançadas do aparelho de TV. As atividades elencadas foram:

- Ligar e desligar o equipamento (simples);
- Selecionar conteúdo (simples);

- Trocar de canal de programação (simples);
- Ajustar o volume (simples);
- Usar tele texto digital (avançado);
- Controlar legendas (avançado);
- Canais favoritos (avançado);
- Ajustar as configurações (avançado).

Com base nesta lista foram escolhidas as atividades de ajuste do volume do som e troca de canais de programação. Estas são as tarefas mais utilizadas pelos usuários e servem para demonstrar o objetivo do trabalho uma vez que todas as funcionalidades disponíveis nas TVDi já estão mapeadas nos *middlewares* existentes.

O modo como os ajustes do volume do som e a troca de canal deveriam ser acionados foi planejado observando os exemplos (VATAVU, 2012), (JUCÁ, 2006), (GAWLINSKI, 2003), (BARROS, 2006), bem como a norma (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 15607:2008, 2008). Estes exemplos indicavam que as informações referentes às opções do painel deveriam ser bem visíveis, em formato de botões e que a movimentação do cursor deveria ser informada ao usuário para que o mesmo fosse informado sobre a posição do cursor, fornecendo assim uma noção de navegabilidade.

Uma vez já definido o que seria comandado na TV através de gestos, foi realizada uma análise contextual sobre os componentes do PACT.

O componente Pessoa do PACT identificou as pessoas que potencialmente usariam o recurso de VC como meio de interação com a TVDi. Estudantes, professores, analfabetos, idosos, crianças foram dispostos em dois grupos de usuários: os que possuíam experiência em uso de sistemas de interação gestual como os *videogames*, TVDi, etc. e os que não possuíam conhecimento técnico ou de experiência em uso de tecnologias de interação gestual.

A técnica adotada para se identificar o perfil de usuários e conceber modelos a partir deste perfil foi a qualitativa. Esta técnica indica que é mais apropriado se utilizar de um conjunto pequeno de usuários (entre 10 e 20), pois oferece uma compreensão melhor e mais realista da amostra de usuários que se relacionam com a TV. Para se definir o nível de experiência dos usuários em uso de dispositivos de interação gestual foi utilizado um grupo de 20 pessoas. Estas foram apresentadas aos dispositivos que executavam ações a partir de comandos gestuais e submetidas a questionários para serem agrupadas e poderem participar

dos *brainstorm* que definiram a construção do *layout* do aplicativo de TV e o conjunto de gestos para execução de comandos na TV.

A componente atividade do PACT referiu-se à interação do usuário com TV através da passagem de informações de coordenada x e y da posição do gesto detectado e um valor que indica qual deveria ser a ação a ser executada. Por isso, a atividade foi abordada em relação à natureza do conteúdo e também em relação a aspectos temporais.

A abordagem da atividade em relação à natureza do conteúdo está voltada a observações dos aspectos do modo que a atividade deve ser realizada. Neste contexto de uso é vital observar os requisitos de dados da atividade e considerar o que acontece quando as pessoas cometem erros ou enganos.

A abordagem da atividade em relação a aspectos temporais estava voltada principalmente aos tempos de resposta que os protótipos iriam realizar suas tarefas. Se uma determinada opção leva muito tempo para dar uma resposta, pode causar frustração. Como regra, as pessoas esperam um tempo de resposta em torno de 100 milissegundos para atividades de coordenação mão-olho e de um segundo para uma relação de causa e efeito, como entre clicar um botão e acontecer alguma coisa. Qualquer resposta que leve mais que cinco segundos deixará uma pessoa frustrada e confusa, conforme (LEMOS, 1997) e Dix *et al*. (1998).

A técnica utilizada para modelar a atividade em relação aos aspectos temporais foi a técnica de Roteiros (CAVAZZA, 2002). Foi feito um roteiro simples que incluiu perguntas a serem feitas aos participantes relacionando as atividades e o sentimento deles ao realizá-las. As perguntas foram elaboradas de tal forma que o entrevistador poderia entrar na questão mais profundamente ou não, conforme julgasse apropriado.

O componente contexto do PACT definiu o contexto físico e o contexto lógico onde foram utilizadas as informações e opiniões dos usuários da TVDi além de modelos já existentes e as regras de usabilidade, facilitando assim as propostas dos novos aspectos desejados.

Esta análise propôs uma série de passos que se iniciaram com a identificação dos perfis dos usuários e das tarefas mais utilizadas no contexto lógico. O passo seguinte foi à observação ou entrevista dos usuários no contexto de uso físico. Desta forma conseguiu-se compreender a maneira como as tarefas eram realizadas e quais as motivações do usuário para a escolha de uma determinada forma de realizá-las. O contexto lógico estava mais vinculado à

atividade realizada pelo usuário. O contexto físico estava mais relacionado ao local onde a atividade ocorria e se utilizava uma TV.

As tonalidades do ambiente físico, assim como a dos objetos que o compõem foram observados e considerados na modelagem, pois se estes possuíssem tonalidades dentro dos limites do tom de pele, representavam um fator a mais de dificuldade na identificação das pessoas pela cor. Esta dificuldade ainda foi agravada pela oscilação da iluminação, que influenciou diretamente nas tonalidades do ambiente e dos objetos lá dispostos.

A última parte do *framework* PACT tratou das tecnologias, os meios com os quais se desenvolveu um produto e se realizou a própria interação. Os sistemas interativos de TVDi podem realizar várias funções e normalmente contêm uma boa quantidade de dados ou conteúdo de informação. As pessoas que usam esses sistemas envolvem-se em interações que, em termos físicos, têm vários graus de estilo ou estética e mudam muito rapidamente. Entre os atuais elementos de interação citam-se detectores de ocupação, movimento e orientação, distância e posição do objeto, toque, voz, olhar e gesto, biometria e estado emocional, etc. (WILSON, 2007).

A técnica inicialmente utilizada para relacionar os componentes do PACT foi a qualitativa, pois produzia um conjunto de informações, de modo bem amplo, levando em conta a preferência de uso da TV observando o dispositivo utilizado para tal tarefa.

A condução da pesquisa qualitativa se deu aplicando os seguintes passos:

- Entrevista individual, utilizada para traçar um perfil de usuário;
- Entrevista em grupo, baseado em um determinado perfil;
- Análise do ambiente real em que ocorria a observação das atividades e comportamentos dos usuários para confirmação do cenário de uso.

Para a fase de entrevista individual foram feitas perguntas claras e diretas sobre o modo que o usuário se comporta diante da TVDi para executar as operações de ajuste do volume do som e troca de canais de programação via controle remoto e via gestos. O indicativo de que se poderia parar de entrevistar era quando o entrevistador já conseguia prever como o usuário iria responder, significando que os padrões estavam começando a surgir. Assim, quando terminadas as entrevistas, todas as variáveis comportamentais foram listadas.

Para a fase de entrevista em grupo segmentou-se antes os usuários. Esta segmentação foi feita em função das atitudes e comportamentos obtidos na fase da entrevista individual, onde as diferenças entre os indivíduos foram observadas e anotadas.

Foram criados neste processo de segmentação dos usuários dois grupos: os experientes, que possuíam bons conhecimentos sobre uso de dispositivos interativos modernos e os básicos, que possuíam conhecimento limitado ou simples sobre o uso de dispositivos interativos avançados.

Os atributos observados na segmentação dos usuários foram:

- Comportamento: como eles faziam o ajuste do volume do som e a troca de canais de programação;
- Atitudes: como eles respondiam ao ter a ação executada ou não.

No que se referiu à definição de *layout*, uma grande parte dos problemas estava relacionada à navegação, ou seja, os usuários tinham dificuldade para encontrar a informação desejada no sistema ou não sabiam como retornar a uma opção visitada anteriormente.

Também foi passada a situação atual dos menus das *Smart* TVs que traziam cada vez mais opções e que o acesso a estas opções se dava através de combinações de teclas ou uma navegação complexa, visitando sub-menus e que seria interessante e mais natural se realizar a interação por meio de gestos.

Na etapa de definição visual foram apresentados *layouts* já conhecidos, os sucessos e os problemas inerentes a eles através dos resultados das entrevistas aplicadas na fase de modelagem, na construção do método de trabalho. A análise comparativa buscou exemplos de *interfaces* para serem utilizadas como referências, procurando observar os diversos enfoques adotados na resolução de questões de design de *interfaces*.

Na etapa de definição de uma linguagem gestual, foram apresentadas soluções já conhecidas, os sucessos e os problemas inerentes a elas através dos resultados das entrevistas aplicadas na fase de modelagem, na construção do método de trabalho. A análise comparativa buscou exemplos de dispositivos de interação gestual para serem utilizados como referências, procurando observar os diversos enfoques adotados na resolução de questões de interação do usuário com o dispositivo e os aplicativos comandado por ele.

4.4 TVDi

A elaboração do *layout* para a TVDi foi realizada seguindo as atividades determinadas na etapa de Engenharia da Usabilidade, onde as atividades que seriam mapeadas e tratadas

eram as relativas ao ajuste do volume do som e à troca de canais de programação da TVDi, dispostas em um painel de controle que facilitasse a navegação dos usuários.

As características do cenário estão definidas na Tabela 4.1.

Tabela 4.1. Cenário para construção do protótipo de TVDi.
Fonte: PRÓPRIA baseada em (BARROS, 2006) e (CAVAZZA, 2002).

Característica do cenário	Ação
Abertura	Executar tela de abertura, deixando um ícone no canto superior direito representando que a interação através do controle remoto está ativada.
Ativar painel de controle	Pelo controle remoto: Clicar no botão verde do controle remoto.
Ajustar volume	Pelo controle remoto: Clicar nos botões referentes ao aumento/ diminuição do volume.
Trocar canal	Pelo controle remoto: Clicar nos botões referentes ao aumento/ diminuição de canal.
Pré-condições	Sistema ativado.
Pós-condições	Sistema de TVDi recebe instruções vindas do controle. Sistema de TVDi executa a ação requisitada.

4.5 Visão Computacional

Segundo o modelo proposto por Bradski *et al*. (2008), o processo de VC se dividiu em duas etapas que eram referentes à de criação de um modelo e ao processamento de imagens. O modelo criado serviu de base ou memória e o algoritmo de processamento de imagens responsável pelas aplicações de filtros nas imagens que eram submetidas à tarefa de busca dos objetos.

A estratégia para a construção do classificador, bem como a eliminação de ruídos e informações desnecessárias ao processo de classificação, como o *Motion Detection*, *Skin Detection*, estão detalhados a seguir.

4.5.1 Projeto de Construção dos Classificadores dos Gestos

Neste trabalho foi aplicado como método de construção dos classificadores dos gestos uma modelagem baseada na aprendizagem de máquina, utilizando a técnica de Viola-Jones (JONES *et al*., 2001) e Lienhart (LIENHART *et al*., 2002). Estes classificadores, contendo a linguagem (variações de um gesto), são submetidos a um algoritmo de reconhecimento de padrões previamente definidos.

Para a construção do classificador passou-se a utilizar os algoritmos fornecidos no pacote OpenCV na seguinte ordem: *Objectmarker*, *CreateSamples* e *Traincascade* para manipular os dois grupos de imagens, as positivas, contendo os gestos que se desejava mapear e as negativas que deveriam conter qualquer tipo de objeto, menos o que se desejava mapear.

O algoritmo apresentado na Figura 4.2 ilustra o mapeamento das ferramentas fornecidas pelo pacote OpenCV para a criação do classificador *Haar-Like*.

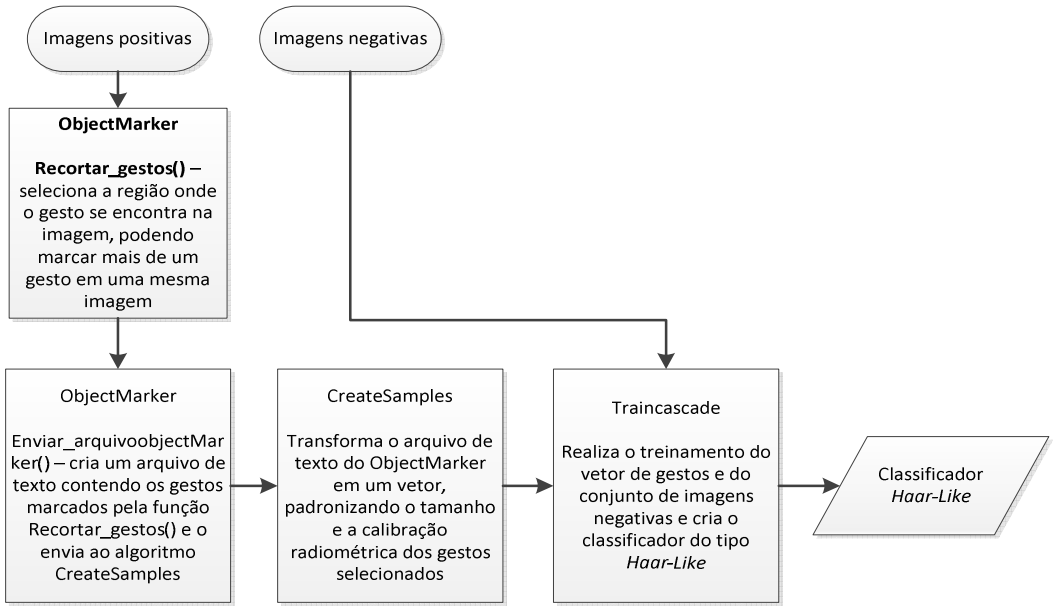


Figura 4.2 Algoritmo da criação do classificador *Haar-Like*.
Fonte: PRÓPRIA. Baseado em (JONES *et al*., 2001).

4.5.2 Processamento Digital de Imagens e Reconhecimento dos Gestos

Para que as imagens pudessem ser entregues ao processo de comparação foram necessárias etapas de: pré-processamento, segmentação, extração de características e o uso dos classificadores.

4.5.2.1 Pré-processamento

Na etapa de pré-processamento foram aplicadas as operações de calibração radiométrica das imagens em relação a brilho e intensidade, as transformações morfológicas responsáveis em diminuir a quantidade de ruídos e a aplicação do filtro de suavização entre os *pixels*.

Para a operação de calibração radiométrica foi aplicada a técnica de equalização de histograma, pois esta técnica padroniza as imagens adquiridas em ambientes onde há variações de iluminação. A operação morfológica combinou os algoritmos de erosão e dilatação, que seguiam um requisito mínimo de vizinhança. Para a aplicação da suavização, foi utilizado o filtro *gaussiano*, que em processamento de imagens, representa uma aproximação dos valores dos *pixels*. O efeito visual obtido com esta técnica foi o da retirada da nitidez dos *pixels*.

4.5.2.2 Segmentação

Para auxiliar na eliminação dos valores dos *pixels* que não participam de nenhum processo de definição e de reconhecimento dos gestos, pôde-se combinar o uso de dois filtros de continuidade: o *Sobel* (VAIRALKAR *et al*., 2013) e o *Canny* (HAN *et al*., 2012).

O filtro *Sobel* realizou a detecção dos *pixels* para se definir bordas mais finas e internas dos gestos. A função *Sobel* utilizou 5 parâmetros que foram respectivamente a imagem de origem em escala de cinza, a imagem destino, uma ordem de derivação para o eixo x, uma ordem de derivação para o eixo y e um fator de tamanho de abertura, que pode ser 1, 3, 5 ou 7. Em todos os casos, exceto para a abertura de valor 1, é aplicado uma abertura em escala 3x1 ou 1x3.

O filtro de *Canny* suavizou ruídos e conseguiu perceber as bordas externas por mais que estas fossem muito finas. O filtro utilizou 5 parâmetros que foram a imagem de origem, a imagem destino, que armazenou as bordas detectadas, um primeiro limiar de corte de valores sobre a imagem de origem, um segundo limiar de corte de valores sobre a imagem destino,

que informou o mínimo de vizinhos que um ponto devia possuir e o fator de tamanho de abertura.

A combinação dos dois filtros se mostrou interessante, pois reforçou as características dos gestos enquanto que os demais objetos ficavam com pouca ou nenhuma nitidez. A imagem resultante do processo de segmentação foi entregue a etapa de extração de características, que tinha como principais tarefas eliminar os valores dos *pixels* estáticos e os que não se encaixavam nos limites definidos de tonalidade de pele.

4.5.2.3 Extração de Características

Na etapa de extração de características, a principal preocupação foi o desconsiderar os *pixels* não significativos, ou seja, aqueles que não representavam nenhum objeto de interesse na imagem. Os métodos aplicados foram o *Motion Detection* e o *Skin Detection*.

O *Motion Detection* combinou duas técnicas: a que observava os *pixels* de borda e os que pertenciam à parte interna dos gestos. A imagem construída por esta combinação devia conter a maior quantidade de informação possível referente aos gestos que se movimentaram, aumentando o nível de informação que seria utilizada no processo de reconhecimento. O algoritmo do *Motion Detection* é apresentado na Figura 4.3.

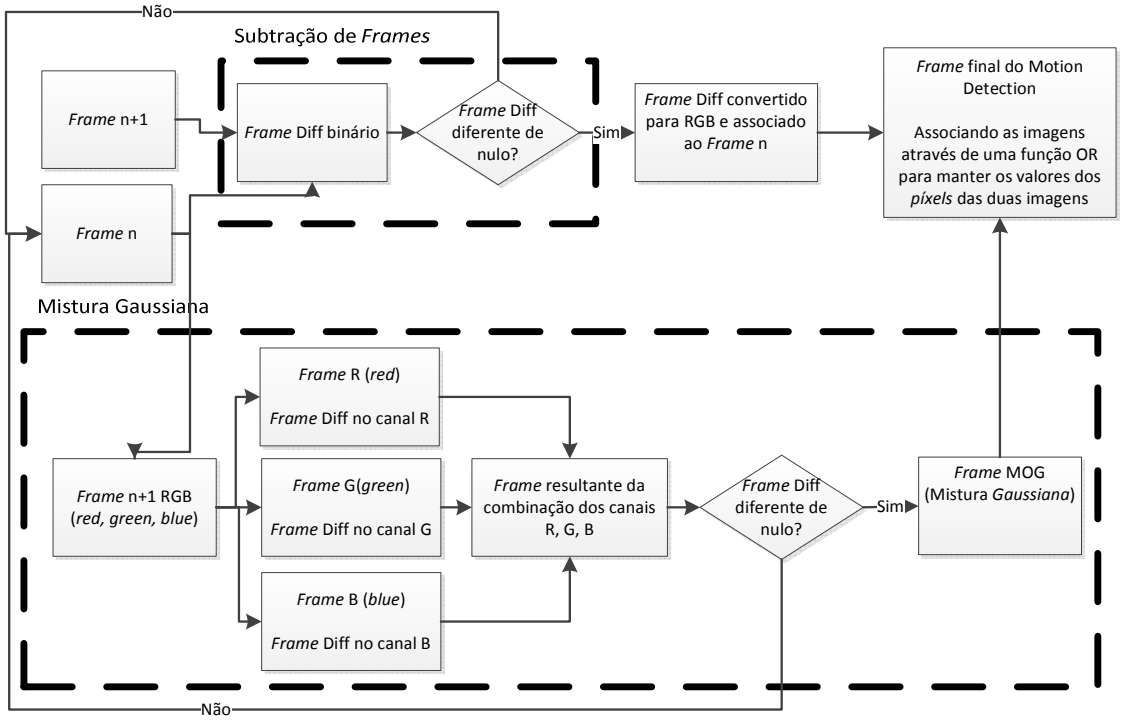


Figura 4.3 Algoritmo do *Motion Detection*.
Fonte: PRÓPRIA.

O *Skin Detection* realizou a separação dos *pixels* da imagem em dois grupos: os que possuíam valores dentro do que foi considerado tom de pele e os que não pertenciam a este grupo. O algoritmo do *Skin Detection* é apresentado na Figura 4.4 que representa o modo de funcionamento da técnica de filtragem com os limites inferior e superior em cada canal de cor. Estes valores influenciam diretamente nos eixos H (*hue*), S (*saturation*) e V (*value*) da imagem HSV e foram definidos a partir dos testes realizados na busca de um valor mais abrangente com o grupo de 20 pessoas integrantes da Engenharia da Usabilidade.

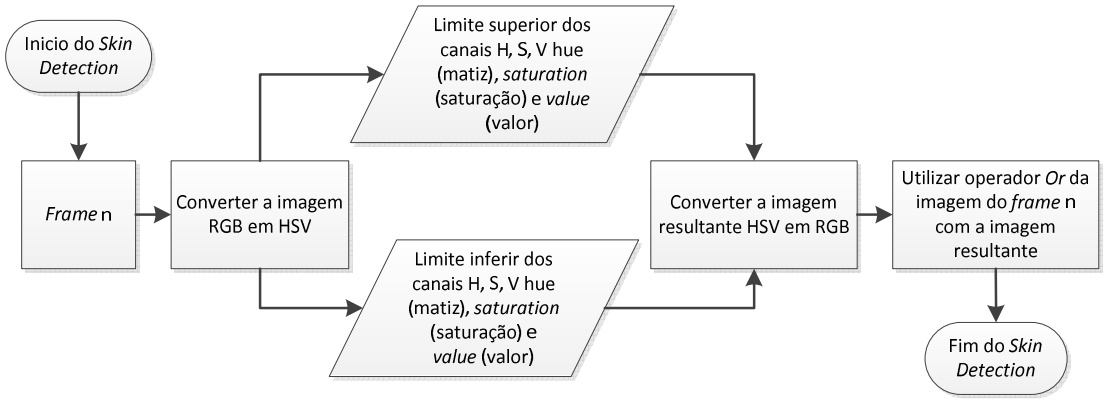


Figura 4.4 Algoritmo do *Skin Detection*.
Fonte: PRÓPRIA.

O modo de funcionamento desta técnica é descrita em 3 passos:

- Converter a imagem do padrão RGB para o HSV. Aplicar um filtro de média (*smoothing*) para suavizar e tornar o mais homogêneo possível à tonalidade de pele presente na imagem;
- Aplicar um limiar de tons de pele, contendo um valor mínimo e um máximo para cada camada de cor. Este limiar deve prever todos os tipos possíveis de tons de pele, mesmo que ainda restem alguns objetos na imagem que não são pele;
- Reescrever o resultado destes algoritmos na imagem original, eliminando os objetos que não se encaixam neste padrão.

A associação das duas técnicas buscou destacar os objetos descritos na imagem resultante do *Motion Detection* que se enquadravam nos limites estabelecidos pelos limites inferior e superior da imagem resultante do *Skin Detection*. A imagem resultante do processo de extração de características foi entregue a etapa de reconhecimento e interpretação.

4.5.2.4 Reconhecimento e Interpretação

A imagem resultante do processamento de imagens digitais foi convertida para escala de cinza e submetida ao processo de comparação com o classificador *Haar-Like*. Ao ser localizado um gesto mapeado por um classificador na imagem submetida, o mesmo é destacado através da inserção de uma moldura quadrada ao redor do referido gesto, enquanto que a sua identificação e coordenada são salvas para serem repassados ao protótipo de TVDi.

A função *Haar-Like* é um autômato que busca um *string* de dados binarizados em uma árvore do tipo *AdaBoost* (é uma árvore onde cada nó é uma sub-árvore) e por isso, tem um custo computacional alto. A Figura 4.5 ilustra a aplicação das técnicas de tratamento das imagens para geração da *string* submetida ao processo de classificação do *Haar-Like*.

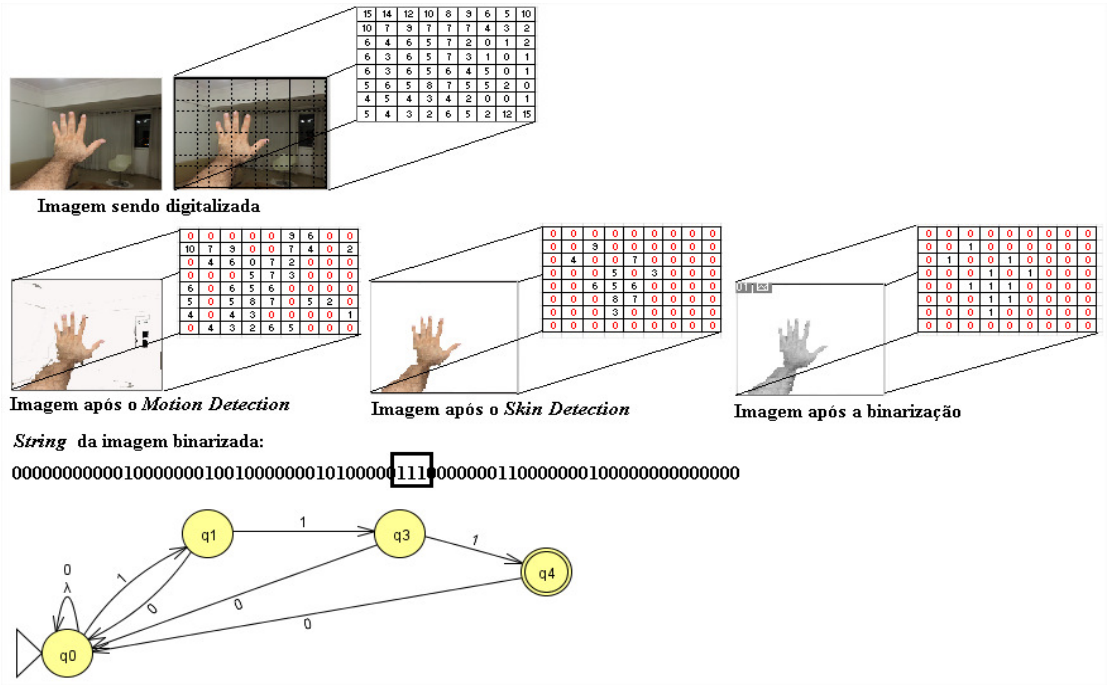


Figura 4.5 Trecho do código do reconhecimento e *tracking* do *Haar-Like* do JavaCV.
Fonte: PRÓPRIA.

Para aumentar a velocidade do protótipo de VC foi aplicado o *CamShift* após o *Haar-Like* detectar o gesto definido no classificador. O *CamShift* é um algoritmo que rastreia uma seleção de uma região-alvo de uma imagem, definida manualmente pelo usuário, em custo constante apenas observando os valores padrão de informações em um histograma. O *CamShift* possui certa robustez para lidar com mudanças de iluminação e é mais rápido que o *Haar-Like* pois não leva em conta o crescimento de regiões, considerações sobre contorno, suavização e predição, que são operações importantes ao funcionamento do *Haar-Like*.

O diagrama de sequência apresentado na Figura 4.6 mostra a Visão Computacional, desde a técnica de processamento de imagens até a etapa de reconhecimento através do classificador *Haar-Like* e o uso do *CamShift* para o rastreamento.

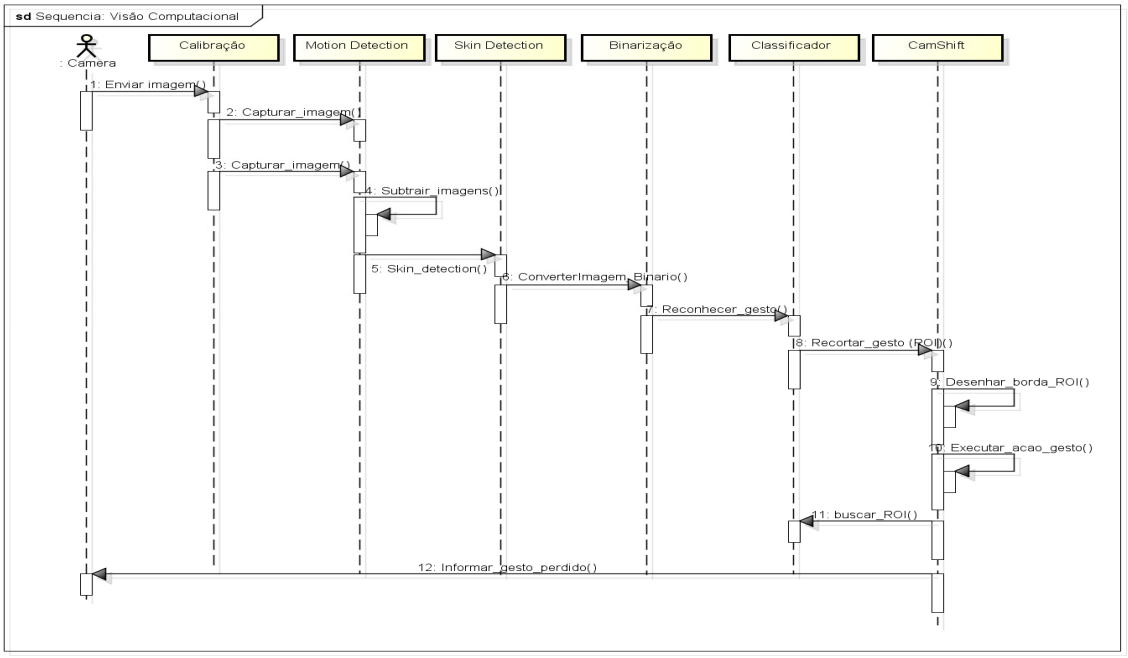


Figura 4.6 Diagrama de sequência do protótipo de Visão Computacional.
Fonte: PRÓPRIA.

Todos os elementos e estratégias adotadas pela VC estão representados pela Figura 4.7. A Figura foi construída de modo a deixar claro o que foi adotado de outros trabalhos e o que foi definido neste trabalho.

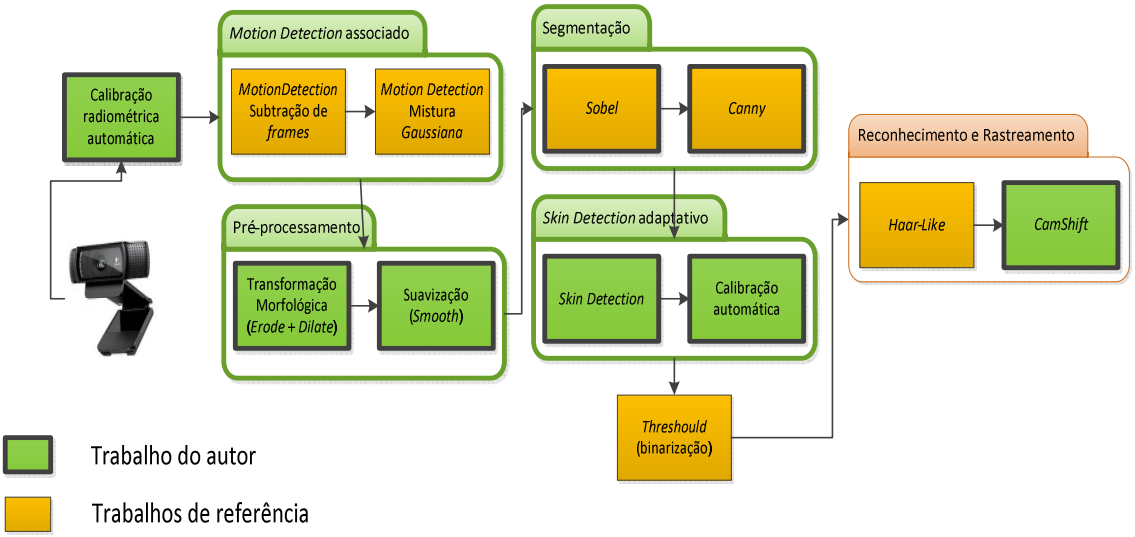


Figura 4.7 Esquema de funcionamento do trabalho de Visão Computacional.
Fonte: PRÓPRIA.

4.6 Módulo de Integração da TVDi com a Visão Computacional

A Tabela 4.2 apresenta informações importantes do contexto físico de uso da televisão através do controle remoto ou através de reconhecimento de gestos. Por meio deste sistema de interação o usuário poderá mudar o canal e o volume através de gestos na TV. Estas informações foram de fundamental importância realizar a união dos protótipos de VC.

Tabela 4.2. Informações sobre o contexto de uso da televisão.
Fonte: PRÓPRIA. Adaptado de (BARROS, 2006) e Rosson *et al*. (2002).

Tópico	**Informações**
Dispositivo de entrada	Tradicionalmente se faz uso do controle remoto, sem fio.
Ambiente de uso	Sala, quarto, cozinha, varanda, quintal (ambientes de descontração e relaxamento).
Mão do usuário	Nem sempre o usuário possui as duas mãos livres para serem usadas na interação com o sistema de televisão.
Distância da tela da televisão	Normalmente, não se está muito perto (mais de um metro).
Postura do usuário	Relaxada, reclinado, em pé, deitado.
Número de usuários	Social: muitas pessoas podem ver a tela (muitas vezes, várias pessoas estarão na sala quando a TV estiver ligada).

O cenário lógico planejado para a integração da TVDi com a Visão Computacional manteve a independência de cada módulo e relacionava os módulos através da passagem dos valores correspondentes a ação que deve ser realizada e a coordenada onde o referido gesto foi localizado na imagem original. Neste cenário, o usuário recebe informações visuais sobre a disponibilidade dos comandos gestuais através da exibição de um ícone. Os botões de troca de canais e de volume estão representados em um painel de controle de operações, que também pode ser operado via controle remoto. As características do cenário estão definidas na Tabela 4.3.

Tabela 4.3. Cenário para construção do protótipo de TVDi com VC.
Fonte: PRÓPRIA.

Característica	Ação
Abertura	Executar tela de abertura, deixando um ícone no canto superior direito representando que a interação através do controle remoto está ativada.
Ativar painel de controle de operações	Pelo controle remoto: Clicar no botão verde do controle remoto. Através de reconhecimento de gestos: Movimentar a mão direita aberta até o ícone e deixar nessa posição por 2 segundos.
Ajustar volume	Pelo controle remoto: Clicar nos botões referentes ao aumento/ diminuição do volume. Através de reconhecimento de gestos: Ativar painel de controle de operações e movimentar a mão aberta até as opções de aumento/diminuição do volume e repousar a mão aberta nesta posição por 2 segundos ou apresentando a mão direita com o dedo indicador apontado para cima ou para baixo para o aumento/diminuição do volume respectivamente.
Trocar canal	Pelo controle remoto: Clicar nos botões referentes ao aumento/ diminuição de canal Através de reconhecimento de gestos: Ativar painel de controle de operações e movimentar a mão aberta até as opções de aumento/diminuição de canal e repousar a mão aberta nessa posição por 2 segundos.
Pré-condições	Câmera conectada. Sistema de VC ativado e operacional. Sistema de VC envia dados ao servidor de aplicação que serão recebidos pela TVDi
Pós-condições	Sistema de TVDi recebe instruções vindas do controle remoto ou do sistema de VC. Sistema de TVDi executa a ação requisitada.

Neste cenário, o usuário comanda sua TV através do controle remoto ou através de gestos, utilizando a Visão Computacional. Em ambos os casos, são enviados valores para o *middleware* da TVDi correspondentes a ação que deve ser realizada e para o caso de uso da Visão Computacional, também os valores correspondentes a coordenada onde o referido gesto foi localizado na imagem original. O diagrama de sequência apresentado na Figura 4.8 mostra os passos necessários para o funcionamento do sistema.

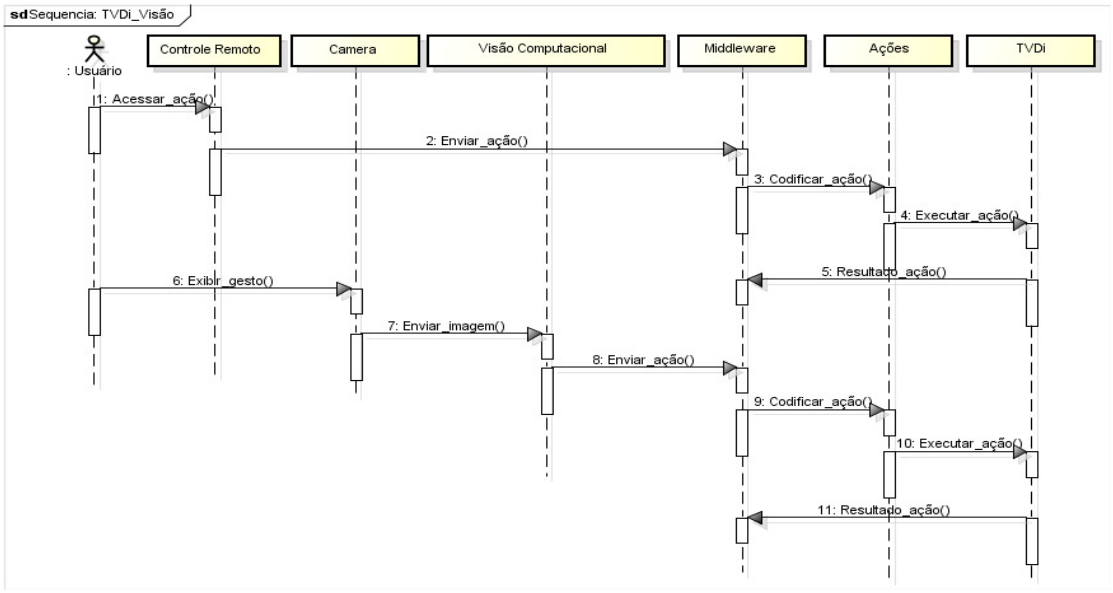


Figura 4.8 Diagrama de sequência para a integração da TVDi e da VC.
Fonte: PRÓPRIA.

4.7 Conclusão

Neste capítulo foram apresentados os conceitos da arquitetura de sistema, onde melhorias foram propostas através da apresentação de estratégias, que demonstram o funcionamento e a forma de comunicação entre os sistemas de TVDi e VC. Os diagramas construídos neste capítulo auxiliam na identificação dos requisitos necessários para a implementação dos protótipos que serão apresentados no Capítulo 5.

Como decisão de projeto, conclui-se que a forma de interligação dos protótipos de TVDi e VC é através do uso de um arquivo XML que contém as coordenadas e a ação correspondente ao gesto identificado. Esta decisão foi tomada visto que as TVDi e os *set-top boxes* ainda não permitem vínculo direto com câmeras e os seus poderes de processamento de dados e armazenamento ainda são muito baixos quando comparados aos computadores. A

câmera adotada deve possuir resolução que permita obter o máximo de informações possíveis para serem entregues ao processamento de imagens e o reconhecimento de gestos.

O próximo passo é definir a forma como o conceito de arquitetura pode ser construído através da análise das diferentes soluções que podem satisfazer as especificações realizadas neste capítulo. Selecionar o que melhor cumpre a especificação não é uma tarefa trivial, e, portanto, as decisões das quais foram utilizadas ferramentas de *hardware* e modelos de *software* são detalhados no Capítulo 5.

Capítulo 5- Implementação do Modelo Proposto

Este capítulo apresenta a implementação da arquitetura do sistema proposta para a interação gestual com a TVDi através da aplicação de ferramentas, equipamentos, *softwares* que garantam que os requisitos sejam alcançados, seguindo os modelos definidos no Capítulo 4. A implementação do modelo proposto tem como principal característica apresentar em um estudo de caso a possibilidade real do uso dos modelos propostos e gerar resultados que serão utilizados na etapa de avaliação.

A forma adotada para a construção do protótipo foi a de desenvolver os subsistemas planejados na arquitetura do sistema, explicando as adequações destes para as características dos *hardwares* e *softwares* escolhidos para este trabalho. Os subsistemas que constituem a arquitetura de sistema são a Engenharia da Usabilidade, a TVDi e a Visão Computacional.

As ferramentas, equipamentos e softwares escolhidas para cumprirem com as definições da arquitetura do sistema também precisavam atender ao requisito de custo, para oferecer um protótipo barato e simples a todos os usuários de TVDi. No entanto, algumas características ausentes nas TVDi e nos *set-top boxes* não permitem o vínculo direto de câmeras, fazendo com que duas propostas de implementação, uma no emulador e outra *no set-top box*, sejam realizadas.

Ao final, para ambas as propostas de implementação, será mostrada a relação com o protótipo de visão Computacional e o sub módulo de relacionamento, responsável pela passagem de informações da Visão Computacional e a TVDi.

5.1 Tecnologias utilizadas na Implementação

Duas abordagens foram necessárias para se desenvolver o protótipo de TVDi, uma baseada na arquitetura do *middleware*, que conseguia controlar o *hardware* e acessar a câmera de forma nativa, ou seja, um aplicativo que utilizou um *middleware* e uma linguagem procedural, e outra, que para acessar a câmera vinculada a outro dispositivo de *hardware*, utilizou uma linguagem declarativa.

Um ponto verificado por este trabalho foi o de que não se tinha disponível ainda no mercado um equipamento de TV capaz de receber a conexão de uma câmera em sua porta USB e que permitisse que aplicativos o controlassem a partir desta câmera. Por esta razão foram necessárias duas abordagens para se realizar a construção do protótipo: uma direcionada a TVDi que aceitou a integração direta com a câmera através do emulador *XletView*, o *middleware* MHP e a linguagem Java (ambiente de aplicação procedural), e outro que acessou informações fornecidas por um servidor de aplicação em rede, como o Ginga-NCL (ambiente de aplicação declarativo).

5.1.1 Tecnologias da TVDi

Para a TVDi, seja no emulador *XletView* ou no *set-top box* real, foram utilizadas as ferramentas descritas na Tabela 5.1.

Tabela 5.1. Ferramentas utilizadas no ambiente de aplicação declarativa.
Fonte: PRÓPRIA.

Nome do recurso	Função
Emulador *XletView*	Contexto lógico semelhante ao encontrado nas TVDi, com as mesmas restrições de recursos físicos e funcionais das TVDi.
Set-top box	Contexto físico e lógico com as mesmas restrições de recursos físicos e funcionais das TVDi.
Middleware MHP	Tecnologia que permite aos aplicativos procedurais serem executados.
Middleware Ginga-NCL	Tecnologia que permite aos aplicativos declarativos serem executados.
TV	Fornece a tela para exibição do painel de opções de controle da TV e do reconhecimento de gestos.
Computador	Computador com o sistema operacional de *Windows* 8 64 bits, processador mínimo 1,5 GHZ, memória RAM mínima de 4 GB.
Câmeras digitais *de* 3 e 12 *Mpixels*	Equipamentos de captura das imagens que serão submetidas ao algoritmo de VC.
Apache TomCat 2.2	Servidor de aplicação que fornece o ambiente lido pelo *set-top box* para atualização dos valores referentes as ações a serem executadas na TV.
Roteador *wireless*	Permite a interligação do computador ao *set-top box* para a troca de mensagens entre os protótipos de TVDi e a VC.

O *XletView* foi o emulador escolhido para a execução do protótipo MHP devido ao volume de bibliotecas disponíveis neste emulador e à possibilidade de se incluir outras necessárias a este trabalho, principalmente as que realizam as tarefas da VC. O *set-top box* escolhido possuía o *middleware* Ginga-NCL e permitiu a inserção do *software* escrito em NCL-LUA através da sua porta USB e da sua porta de rede *Ethernet*.

5.1.2 Tecnologias de Visão Computacional

Observando a arquitetura construída para este trabalho, foi possível indicar as características mais importantes de *hardware* e *software* e algumas ferramentas e equipamentos necessários à construção dos protótipos. A Tabela 5.2 traz os itens necessários para se desenvolver os protótipos de VC.

Tabela 5.2. Ferramentas utilizadas na VC.
Fonte: PRÓPRIA.

Nome do recurso	Função
Câmera	Câmeras digitais *de* 3 e 12 *Mpixels.*
Computador	Computador com o sistema operacional de *Windows* 8 64 bits, processador mínimo 1,5 GHZ, memória RAM mínima de 4 GB.
*C*onstrução de classificadores	OpenCV e JavaCV com os algoritmos *ObjectMaker*, *CreateSamples* e *TrainCascade*
Software de PDI e VC	*Software* de tratamento das imagens e reconhecimento de gestos construído em Java e JavaCV

O protótipo de VC adotou uma câmeras de 3 *Mpixels* e *frame rate* de 30 fps a 12 *Mpixels* e *frame rate* de 60 fps com resolução de 640x480 *pixels* para gerar imagens com a resolução definida na arquitetura do sistema. Na análise realizada nos trabalhos relacionados sobre Visão Computacional, foi sugerido que quanto maior fosse a resolução da câmera mais elementos seriam capturados, aumentando as chances de se reconhecer os gestos com maior precisão. Esta câmera trabalha com o sistema 3-cores RGB (*Red*, *Green*, *Blue*) além do fator determinante de luminância, representado por valores em cinza.

Para implementar os algoritmos de VC optou-se pelo uso do pacote JavaCV, um *wrapper* do OpenCV (*Open Source Computer Vision Library*) que fornece um conjunto de algoritmos já testados e que diminuem o tempo de construção de *softwares* de tratamento de

imagens e de reconhecimento de gestos ou objetos. Um *wrapper* é um termo que se refere a um empacotador de funções que convocam outras funções. Neste caso, o JavaCV necessita das funções pertencentes ao OpenCV (BRADSKI *et al*., 2008).

5.1.3 Tecnologias Comuns a TVDi e a Visão Computacional.

Para este trabalho foi escolhida a linguagem de programação Java devido sua característica de independência em relação aos sistemas operacionais e à arquitetura de inúmeros *hardwares*, como as TVDi e os *set-top boxes*. A linguagem Java também é utilizada para o desenvolvimento dos *softwares* de VC através da *wrapper* JavaCV.

As atuais TVs e *set-top boxes* possuem porta de comunicação de redes para acessar serviços fornecidos pelas operadoras e também para se interconectarem. Assim, para conexão entre o *set-top box*, a TVDi e o computador foi necessário utilizar um roteador de acesso a rede de dados.

5.2 Implementação do Cenário

No cenário deste trabalho, a TV estava localizada em uma sala residencial juntamente com todos os objetos que a compõe, como sofá, poltronas, cadeiras, mesas, armários, etc., e tem as operações de ajuste do volume do som e troca de canais de programação comandados via controle remoto e via gesto. Para se chegar ao protótipo final, alguns passos foram necessários.

O primeiro passo tratou da definição do *layout* do protótipo e do conjunto de gestos responsáveis por executar as ações na TVDi. Aqui foram adotadas as estratégias definidas na arquitetura de sistema referentes à Engenharia da Usabilidade com a participação do usuário neste processo.

O segundo passo tratou da construção do protótipo de TVDi, modelado a partir de duas possibilidades: a procedural, onde foi possível acessar de forma direta recursos de *hardware* da câmera e fazer a integração da TV com a VC, e a declarativa, onde o protótipo de TV foi construído com uma linguagem que apenas acessou as funções já existentes em sua sintaxe e necessitou se relacionar com outro aplicativo que era capaz de realizar o reconhecimento dos gestos.

O último passo referiu-se a construção do protótipo de VC. A implementação da VC dependeu de dois trabalhos separados que foram: a construção do classificador e o *software* de processamento de imagens e reconhecimento de gestos.

A forma adotada para o relacionamento entre os protótipos se deu através de compartilhamento de um arquivo XML contendo as informações obtidas pelo protótipo de VC referentes à localização do gesto e à ação que o mesmo representa na TVDi, acessado através de rede de computadores.

5.2.1 Engenharia da Usabilidade

A primeira etapa para a construção baseada no usuário foi escolher um conjunto composto por 20 pessoas e classifica-las de acordo com o seu perfil. O fator de observância considerado para definição do perfil do usuário foi o de experiência no uso de equipamentos com interação gestual.

Estas pessoas foram submetidas a entrevistas individuais baseadas na técnica de roteiros (CAVAZZA, 2002), registrando o modo que estas se comportavam diante de uma TVDi para executar as operações de ajuste do volume do som e troca de canais de programação via controle remoto e via gestos. As perguntas foram realizadas enquanto não era possível ao entrevistador prever as respostas dos usuários.

A entrevista buscou deixar o usuário à vontade, pois muitas vezes as pessoas em uma situação de questionamentos e testes se sentiam nervosas. Para isso, foi explicado que quem estava sendo testado era o protótipo e não o usuário, e lembrando que ele poderia interromper o teste a qualquer momento. Foram necessárias 8 aplicações do questionário para que se definisse o padrão de respostas do grupo de estudo. O Apêndice D contém o modelo desta entrevista.

Ao final das entrevistas individuais, o conjunto de pessoas foi dividido em dois grupos: o grupo que continha os integrantes experientes e o grupo de integrantes com pouca ou nenhuma experiência em uso de tecnologias de interação gestual.

O trabalho da Engenharia da Usabilidade foi realizado seguindo o roteiro de descrição de cenário da Tabela 5.3 com vários aspectos que foram observados.

Tabela 5.3. Roteiro de descrição de cenário.
Fonte: PRÓPRIA. Adaptado de (BENYON, 2011).

Roteiro	Significado dos aspectos usados no roteiro.
Cenário	Sala residencial com sofá, mesa de centro, cadeiras, enfeites sobre os móveis.
Atores	Usuário experiente e usuário pouco experiente em uso de recursos de interação gestual.
Objetivos da tarefa	Ferramenta auxiliar ou substituta ao controle remoto na execução das tarefas de volume e troca de canal da TV.
Avaliação	Operabilidade para medir o grau de dificuldade de execução dos comandos; Satisfação através de relatos da experiência do uso do protótipo com uso de questionário pós-teste.
Ações	Observação dos comportamentos e atitudes dos usuários.
Eventos	Observação de eventos externos que possam influenciar a atividade como barulhos, outros usuários no ambiente, mudança repentina de iluminação.

5.2.1.1 Definição de *Layout* do protótipo de TV

Para a etapa de definição de *layout*, foi aplicada a observação e coleta de informações dos usuários através da técnica *brainstorm* e entrevistas. Em relação a outras fontes de informações, foram buscados modelos já existentes de *layout* de TVDi para confrontar o modelo construído com os padrões adotados pelo mercado.

As perguntas feitas aos usuários estão descritas no Apêndice E, porém as principais foram:

- Questão 1: Plano de fundo: qual a melhor cor para ser aplicada a tela de plano de fundo para exibição dos componentes da TVDi?
- Questão 2: Qual é o aspecto visual do componente responsável pelo volume?
- Questão 3: Qual é o aspecto visual do componente responsável pela troca de canal?

Para facilitar a construção do *layout* da TVDi foram apresentados os modelos de Brackmann (2010) e Jucá (2006) visando melhorar o entendimento sobre o funcionamento da TVDi. O *layout* final, definido pelos usuários, contendo os aspectos visuais do painel de controle e dos componentes de volume, canal e de desligamento do aplicativo, é apresentado na Figura 5.1.

Figura 5.1 *Layout* da aplicação de controle através de gestos para a TVDi.
Fonte: PRÓPRIA.

Neste *layout*, o usuário movimenta o cursor que inicialmente está localizado ao centro do painel de controle até uma das opções de canal, volume ou para desligar o reconhecimento de gestos do protótipo. Este cursor deve ficar sobre esta opção por 2 segundos para que a ação seja realizada. Este tempo foi dado para impedir que gestos involuntários sejam erroneamente traduzidos em ações no protótipo. O tamanho dos botões também foi definido a partir da observação da distância em que o usuário se posicionava em relação a TV, com cerca de 4,0 metros.

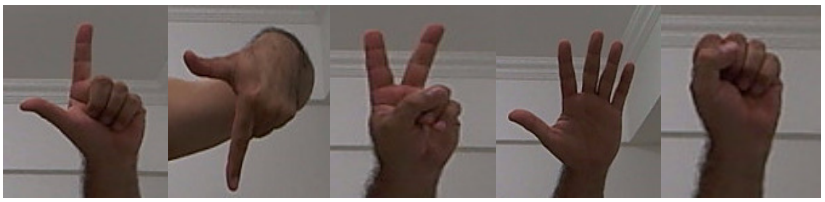
5.2.1.2 Definição de uma Linguagem Gestual

Para a etapa de definição de uma linguagem gestual, os usuários foram informados que o trabalho buscava encontrar um gesto que pudesse executar as ações utilizadas via controle remoto na interação com a TVDi. Um questionário foi aplicado ao grupo contendo perguntas simples e que levavam os integrantes a refletir sobre o modo de realização de tarefas junto à TV. Todas as questões utilizaram como pré-requisito a norma brasileira ABNT NBR 15604 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 15604, 2008), que descreve quais funções e o modo que estas devem ser viabilizadas pelo controle remoto. Logo, deixou-se claro que qualquer solução proposta pelo grupo que viesse a auxiliar ou substituir o controle remoto deveria prover essas funcionalidades básicas sem perder de vista a simplicidade da solução.

As questões descritas no Apêndice F foram:

- Questão 1: Qual gesto representa a ação: Aumentar/diminuir volume?
- Questão 2: Qual gesto representa a ação: Trocar canal?
- Questão 3: Qual gesto representa a ação: Movimentar cursor livremente pela tela?
- Questão 4: Qual gesto representa a ação: Executar uma opção?

A aplicação do questionário junto aos na etapa de definição do grupo de gestos resultou no conjunto de gestos exibido na Figura 5.2.



(a) (b) (c) (d) (e)

Figura 5.2. Gestos sugeridos para serem utilizados no protótipo de TVDi onde (a) aumenta volume, (b) diminui volume, (c) troca canal, (d) movimenta cursor e (e) seleciona opções no protótipo de TVDi.
Fonte: PRÓPRIA.

Após definidos os gestos apresentados na Figura 5.2 foi realizado um encontro com os usuários. Neste encontro foi apresentado o *layout* do painel de controle da TVDi representado pela Figura 5.1 e solicitado que os participantes fizessem uso dos gestos sugeridos de forma isolada ou combinados para as tarefas de ajuste do volume do som e troca de canais de programação sobre o *layout*. O resultado das sugestões foi:

- Alterar volume: Podia ser executada de duas formas: o gesto (a) para aumentar e o (b) para diminuir pode ser executado isoladamente ou exibido o gesto (c) para mostrar o painel de botões de volume e canal para então executar o gesto (d) para movimentar o cursor e (e) para executar a operação.
- Trocar canal: Podia ser executada da seguinte forma: o gesto (c) deve ser exibido para apresentar o painel de botões de canal e volume. Deste ponto em diante, o modo de alteração dos canais deveria ser executado da mesma forma que a Operação 1.

5.2.2 TVDi

As primeiras implementações do protótipo de TVDi associado a Visão Computacional foram desenvolvidas em linguagem Java, onde o *software* de TV recebia as instruções vindas do reconhecimento de gestos. Esta aplicação realizava a troca de mensagens com o servidor de aplicação *Apache TomCat*, atualizando o arquivo XML que continha os dados da coordenada e do código referente a ação do gesto reconhecido em seu diretório.

A aplicação de VC utilizou o componente *IplImage*, baseado na biblioteca de processamento de Imagens (IPL) da *Intel*®, que era responsável em receber e exibir as imagens obtidas da webcam. Este componente era incompatível com os componentes do

AWT, *Swing*, HAVi, JavaTV que tratavam de imagem na TVDi. Por isso, a técnica utilizada para exibir as imagens adquiridas no MHP por este componente foi dividida em três etapas:

(1) Receber o *Buffer* do *IplImage* e enviá-lo a um componente *BufferedImage*,

(2) Receber em um componente *Image* o conteúdo do *BufferedImage* e

(3) Enviar o *Image* a um JLabel, componente pertencente ao pacote *Swing* responsável em exibir texto e imagem na linguagem Java.

Quando se executava o protótipo de VC na TV através do *startXlet* no emulador *XletView*, a aplicação MHP travava enquanto a classe de VC estivesse funcionando. Este comportamento deveu-se pelo fato do *Xlet* ter sido projetado para executar somente uma tarefa por vez. O *Xlet* foi modificado, incorporando ao método *startXlet* um *thread* responsável pela execução em paralelo da classe de VC, eliminando assim a concorrência ao recurso de *hardware*.

A implementação do protótipo procedural foi realizada de modo a permitir que as operações do *middleware* MHP recebesse suas instruções a partir do arquivo XML fornecido pelo *Apache TomCat*. Foi utilizada a linguagem Java (JDK versão 7.5) e a API JavaTV (versão 1.1). o código MHP foi executado no emulador XletView.

O principal elemento na construção da *interface* gráfica é representado pela classe HScene. Esta classe se assemelha ao *Frame* do pacote AWT ou *Swing* de Java e é utilizado como *container* principal. Esta classe apresentou algumas restrições como, por exemplo, a exibição de uma única instância durante qualquer momento da existência da aplicação. Para realizar as operações pertinentes ao protótipo de TV foram utilizados os componentes HAVi (TEIRIKANGAS, 2001). A utilização deste pacote foi essencial, pois existiam algumas operações do protótipo de TV que não eram realizáveis com os componentes nativos. A Tabela 5.4 mostra as operações previstas no *middleware* MHP.

Tabela 5.4. Mapeamento dos botões do controle remoto no *middleware* MHP.
Fonte: (AMERINI *et al*., 2010).

Ação	Comando
Diminuir canal	HRcEvent.VK_CHANNEL_DOWN
Aumentar canal	HRcEvent.VK_CHANNEL_UP
Diminuir volume	HRcEvent.VK_VOLUME_DOWN
Aumentar volume	HRcEvent.VK_VOLUME_UP

A implementação do protótipo em GINGA-NCL, escrita para ser executada em um *set-top box*, utilizou um código escrito em LUA para realizar a leitura do arquivo XML fornecido pelo servidor de aplicação *Apache* TomCat.

A Tabela 5.5 mostra as operações previstas no *middleware* NCL/LUA.

Tabela 5.5. Mapeamento dos botões do controle remoto no *middleware* NCL/LUA.
Fonte: (IERUSALIMSCHY *et al*., 2007).

Ação	Comando
Diminuir canal	CHANNEL_DOWN
Aumentar canal	CHANNEL_UP
Diminuir volume	VOLUME_DOWN
Aumentar volume	VOLUME_UP

A Figura 5.4 apresenta um trecho de código fonte da classe app.ncl contendo a construção da região.

```
<regionBase>
    <region id="rgVideoFullscreen" zIndex="1"/>
    <region id="rgLua" width="100%" height="100%" right="0" top="0" zIndex="10" />
</regionBase>
```

Figura 5.3 Trecho do código fonte da classe app.ncl.
Fonte: PRÓPRIA.

Para apresentar o documento multimídia foram utilizados os elementos chamados de regiões no NCL. Estes elementos, definidos no cabeçalho do documento NCL, indicaram a posição e o tamanho da área onde cada elemento visual foi apresentado. Os atributos utilizados para se definir uma região foram:

- id: responsável pela identificação da região;
- zindex: definiu a sobreposição das camadas, informando, de através um valor numérico, se uma região seria apresentada sobre outras regiões com zindex menor.
- width: definiu a dimensão horizontal da região;
- heigth: definiu a dimensão vertical da região;
- right: definiu a coordenada horizontal à direita da região;
- top: definiu a coordenada superior da região.

5.2.3 Visão Computacional

O protótipo de VC foi construído em duas etapas. Na primeira, foram construídos os classificadores do tipo *Haar-Like* para servirem de modelo. Estes classificadores foram definidos de tal forma que abrangesse a maior quantidade possível de identificadores para cada gesto, obtidos a partir de variações em que o mesmo pode ser apresentado. Em uma segunda etapa, foi construído um *software* para realizar o processamento de imagens, que buscou diminuir a quantidade de *pixels* submetidos ao processo de reconhecimento de gestos. Os detalhamentos destas duas etapas são apresentadas na seção abaixo.

5.2.3.1 Criação do Classificador de Gestos

Um classificador não supervisionado do tipo *Haar-Like* foi criado e utilizando os seguintes passos:

1. Criar um grupo de imagens do gesto que se desejava mapear, chamado de imagens positivas, executado pelas 20 pessoas participantes do grupo da Engenharia da Usabilidade, variando o ângulo de apresentação dos gestos, a iluminação, as tonalidades do plano de fundo e as distâncias dos gestos em relação à câmera.
2. Criar um grupo de imagens contendo qualquer cena ou objeto, excluído o gesto escolhido para mapear, chamado de imagens negativas, que foi utilizado como plano de fundo no processo de treinamento.

Foi construído um classificador para mapear cada um dos 5 gestos definidos pelo grupo de estudo. Para cada gesto foram utilizadas 2.000 imagens positivas, obtidas por um *software* próprio e 2.000 imagens negativas, construído a partir de um conjunto de fotos pessoais e que poderiam ser utilizadas como plano de fundo para todos os gestos mapeados.

A quantidade de imagens foi definida inicialmente de modo empírico, uma vez que os trabalhos relacionados descreviam em seus experimentos o uso de quantitativos de imagens com baixa resolução.

Para aumentar a resolução das imagens pertencentes ao banco de imagens, foi adotado como padrão o uso de uma câmera de 12 *Mpixels*. Com este aumento da resolução das imagens foi observada uma melhora na construção da árvore de características que alcançava os mesmos quantitativos de características encontradas sobre o gesto mapeado quando comparado aos trabalhos relacionados, porém com quantidades menores de imagens.

Após definidos estes dois grupos de imagens, passou-se a utilizar os algoritmos *ObjectMarker*, *CreateSamples* e *Traincascade*, fornecidos pelo conjunto de bibliotecas do OpenCV.

O *ObjectMarker* criou o arquivo contendo o nome da imagem e as coordenadas da área de marcação. A Figura 5.4 mostra o algoritmo *ObjectMarker* sendo utilizado, marcando o objeto com uma borda e salvando em um arquivo de texto a referência da imagem com as coordenadas onde foi recortado o referido gesto.

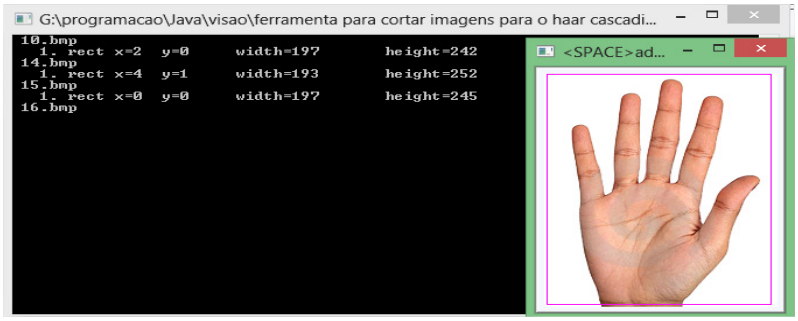


Figura 5.4 *ObjectMarker* mapeando as imagens positivas para gerar o arquivo de texto.
Fonte: PRÓPRIA.

Após criado o arquivo de texto contendo as referências das imagens, foi utilizado o *CreateSamples*, que converteu o arquivo de texto gerado no *ObjectMarker* em um vetor, ao mesmo tempo que padronizou o brilho, iluminação e dimensionamento de janela para as imagens recortadas do grupo de imagens positivas. O tamanho padrão escolhido para as imagens deste trabalho é 24x24 *pixels*.

De posse do vetor de imagens positivas e o diretório contendo as imagens negativas executou-se o *Traincascade* que realizou o treinamento e criação do classificador. Esta etapa levou muito tempo para ser executada, pois em uma janela de 24x24 *pixels* existem mais de 160 mil características. Por isso, para serem observadas, foi importante acompanhar as estimativas de tempo que eram exibidas na tela e perceber se o classificador iria ser eficiente ou não baseado nos sucessos e nas taxas de falso alarme de cada etapa. Como o processo de reconhecimento é estatístico, utilizou-se como métrica para analisar a eficiência de cada nó gerado na cascada o valor percentual máximo de falso alarme. O valor do falso alarme em cada estágio era gerado pelo processamento das características utilizando o parâmetro MaxFalseAlarmRate.

A taxa máxima de falso alarme é definida pela relação de alarmes falsos e o número de estágios definidos na construção da cascada. Para esta taxa de falso alarme considerou-se que

em 1000 amostras negativas (aquelas que não possuíam o gesto a ser reconhecido) o sistema detectou de forma equivocada cerca de 200 amostras. Para esta situação, o falso alarme será 200/1000 = 0,2. Se esse valor fosse abaixo de 0,2, a probabilidade de se ter um classificador mais eficiente seria maior, porém, com o risco deste valor tender ou alcançar zero, podendo-se chegar aos problemas de *overfitting* e *overtraining*, que são problemas relacionados a uma especialização das amostras de treino e não o de generalização do problema.

O resultado da execução do *Traincascade* contendo o mapeamento das informações de classificação do gesto em formato de cascata foi posto em um arquivo em formato XML. Este arquivo armazenou as informações referentes aos pesos (limiar de separação, quantidade de retângulos que eram utilizados na estrutura da árvore de decisão) e a função de localização, que para o *Haar-Like* é utilizado também como critério para redimensionar os gestos mapeados.

A função de localização também informou o quanto à imagem original deveria ser reduzida a ponto de que a menor mão existente nesta imagem se encaixe em uma janela de 24x24 *pixels*. Como se utilizou somente de uma câmera, a imagem gerada possui somente as informações bidimensionais, que não fornecem dados sobre a distância do gesto em relação à câmera e do gesto em relação ao plano de fundo. Para realizar a detecção dos gestos dentro um limite mínimo e um limite máximo de distância, um parâmetro referente à escala da janela de busca foi adotado para dimensioná-la em 10% a partir do tamanho original adotado. Este dimensionamento representava que, em uma segunda visita ao grupo de *pixels*, esta janela buscaria por valores que se encaixassem em suas dimensões e características. A aplicação deste fator de dimensionamento foi escolhida observando a relação entre a eficiência e o desempenho do sistema. Quanto maior fosse o parâmetro de escala mais lenta era a busca, mas em compensação era possível localizar mãos muito pequenas ou apresentadas no limite máximo de distância. Por outro lado, quanto menor fosse o parâmetro mais rápido era a busca, porém, mãos pequenas eram ignoradas.

5.2.3.2 Processamento Digital de Imagens e Uso dos Classificadores

Os primeiros experimentos referentes à etapa de pré-processamento deram uma atenção especial aos ruídos, que foram gerados por diversas fontes como a deficiência da câmera utilizada, a iluminação do ambiente, a posição relativa do objeto de interesse e a câmera, a cor da pele do usuário, a cor dos objetos do plano de fundo. As técnicas adotadas foram a de transformação morfológica e da suavização.

A calibração radiométrica aplicou a equalização de histograma através da normalização dos contrastes e brilhos na imagem original obtida pela câmera para que o sistema pudesse se adequar de forma automática as novas condições do ambiente. Para normalizar os contrates e brilhos na imagem original foi utilizada a função *GaussianBlur* que realiza a suavização *Gaussiana*, dando como resultado uma imagem com aspecto de borrado.

A Figura 5.5 ilustra o trecho do código responsável pela calibração radiométrica e adequação as condições de luz do ambiente.

```
public static IplImage tratar_imagem(IplImage EntradaImage,int width, int height  ){
    IplImage TempImageReduz       = IplImage.create(width/scale, height/scale, IPL_DEPTH_8U, 3);
    IplImage WorkImageReduz       = IplImage.create(width/scale, height/scale, IPL_DEPTH_8U, 3);
    IplImage grabbedImageReduz    = IplImage.create(width/scale, height/scale, IPL_DEPTH_8U, 3);
    IplImage WorkImage            = IplImage.create(width, height, IPL_DEPTH_8U, 3);

    cvResize(EntradaImage, grabbedImageReduz, CV_INTER_LINEAR);

    //normalizar os contrates e brilhos na imagem original
    GaussianBlur(grabbedImageReduz, TempImageReduz, cvSize(5,5), 0, 0, BORDER_DEFAULT);

    // increase contrast and adjust brightness and create blended image
    cvAddWeighted(grabbedImageReduz, gama1, TempImageReduz, -gama2, 0.2, WorkImageReduz);
    cvZero( TempImageReduz );

    cvResize(WorkImageReduz, WorkImage, CV_INTER_LINEAR);

    return WorkImage;
}
```

Figura 5.5 Trecho do código de calibração radiométrica do protótipo de VC.
Fonte: PRÓPRIA.

O efeito obtido pela aplicação da calibração radiométrica foi o da anulação dos valores de um grupo de *pixels* destoantes dos seus vizinhos que causavam descontinuidade em certas regiões da imagem. O nível de borramento dado na imagem foi definido pelo valor inserido em *cvSize*, que percorreu a imagem com uma janela do tamanho definido em seus parâmetros de dimensão. O parâmetro *BORDER_DEFAULT* definiu a propriedade de *borderMode* e indicou que os *pixels* na imagem destino que correspondessem aos valores extremos deveriam ser modificados juntamente com os demais. O *GaussianBlur* reduziu os componentes de alta frequência da imagem e foi utilizado como um filtro passa-baixa. A função *cvAddWeighted* foi utilizada para unir as imagens original e a normalizada, incrementando e os contraste e ajustes de brilho obtidos pelo processamento do *GaussianBlur*.

A transformação morfológica, que é a combinação dos filtros de erosão (*erode*) e dilatação (*dilate*), utilizou o fator de dilatação e erosão de 2 *pixels*, conforme o trecho de código ilustrado na Figura 5.6. O valor do fator foi deixado como parâmetro para que o

protótipo pudesse ser testado com outras câmeras e o valor direcionado as configurações destes outros equipamentos.

```
static IplImage  Transformacao_morfologica(IplImage EntradaImage, int fator){
    cvDilate(EntradaImage, EntradaImage, null, fator);
    cvErode(EntradaImage,  EntradaImage, null, fator);

    return EntradaImage;
}
```

Figura 5.6 Trecho do código da transformação morfológica.
Fonte: PRÓPRIA.

Após a aplicação da transformação morfológica foi aplicado um filtro de suavização que aproximou os valores dos *pixels* e eliminou outros tipos de ruídos que ainda restavam na imagem através da função *cvSmooth*, que possuía como componentes a imagem de entrada, a imagem de saída, o tipo de suavização e o fator dado ao tipo de suavização. O tipo de suavização utilizado foi o *CV_MEDIAN* com o fator igual a 5, pois este tinha um custo computacional constante. O trecho do código do pré-processamento responsável por esta etapa da retirada dos ruídos está descrita no trecho do código da Figura 5.7.

```
static IplImage Suavizacao(IplImage EntradaImage, int fator){
    cvSmooth(EntradaImage, EntradaImage, CV_MEDIAN, fator);
    return EntradaImage;
}
```

Figura 5.7 Trecho do código de suavização do pré-processamento.
Fonte: PRÓPRIA.

Os outros tipos de suavizações existentes no JavaCV são o *CV_BLUR_NO_SCALE*, que realiza a convolução na imagem inteira e com diferentes tamanhos de janelas; o *CV_BLUR*, que realiza a suavização na escala de 1/(*size1xsize2*), onde *size1* e *size2* são as dimensões da janela de suavização; o *CV_GAUSSIAN*, que realiza a suavização com o núcleo *Gaussiano* e o *CV_BILATERAL*, que realiza a mesma suavização do *CV_MEDIAN*, além de incluir outros dois parâmetros responsáveis pela suavização em cores especificas.

A etapa de segmentação consistiu em aplicar filtros de bordas *Sobel* e o *Canny* para reforçar a diferença de valor de *pixel* dos gestos e o plano de fundo através do processo de aproximação de valores por semelhança.

O filtro *Sobel r*ealizou uma aproximação do gradiente da intensidade dos pixels da imagem com um fator de tamanho de abertura de 3. O trecho do código da segmentação responsável pela aplicação do *Sobel* está descrita no trecho do código da Figura 5.8.

```
static IplImage Sobel(IplImage EntradaImage, int width, int height){
    //filtro de sobel - bordas menos significativas e internas
    IplImage result    = IplImage.create(width, height, IPL_DEPTH_16S, 1);
    IplImage grayImage = IplImage.create(width, height, IPL_DEPTH_8U, 1);

    //converte a imagem colorida de entrada para uma imagem em escala de cinza
    cvCvtColor(EntradaImage, grayImage, CV_RGB2GRAY);

    cvSobel(grayImage, result, 1, 0, 3);
    IplImage SobelImage = IplImage.create(width, height, IPL_DEPTH_8U, 1);
    cvConvertScale(result, SobelImage, 1, 0);

    //funcao de binarizacao da imagem final da aplicação do Sobel
    cvThreshold(SobelImage, SobelImage, 25, 255, CV_THRESH_BINARY);

    return SobelImage;
}
```

Figura 5.8 Trecho do código da segmentação com o filtro de *Sobel*.
Fonte: PRÓPRIA.

O filtro de *Canny* utilizou como limiar de corte de valores sobre a imagem de origem o fator 120, um segundo limiar de corte de valores sobre a imagem destino determinando o número mínimo de vizinhos que um ponto deve possuir igual a 3 e a fator de tamanho de abertura igual a 3. O trecho do código da segmentação responsável pela aplicação do *Canny* está descrita no trecho do código da Figura 5.9.

```
static IplImage Canny(IplImage EntradaImage, int width, int height){
    IplImage CannyImage  = IplImage.create(width, height, IPL_DEPTH_8U, 1);
    cvCvtColor(EntradaImage, CannyImage, CV_RGB2GRAY);

    cvCanny(CannyImage, CannyImage, 120, 3, 3);

    //aplicacao da transformação morfológica com fator=2 para eliminar ruidos
    CannyImage = pre_proc.Transformacao_morfologica(CannyImage, 2);
    cvThreshold(CannyImage, CannyImage, 1, 255, CV_THRESH_BINARY_INV);

    return CannyImage;
}
```

Figura 5.9 Trecho do código da segmentação com o filtro de *Canny*.
Fonte: PRÓPRIA.

Na etapa de extração de características a principal preocupação estava relacionada na redução dos *pixels* não significativos, ou seja, aqueles que não tinham nenhum interesse ao processo de classificação. As técnicas utilizadas foram o *Motion Detection* e o *Skin Detection*.

Para o uso do *Motion Detection*, foi observado o estudo do estado da arte, combinando duas técnicas: a que observava os *pixels* de borda (*Motion Diff*) e a que observava os *pixels*

internos (Mistura *Gaussiana*) em um movimento. Esta decisão foi tomada para preservar a maior quantidade possível dos valores dos *pixels*, pois o classificador possuía mapeado o gesto inteiro e não somente a borda ou os *pixels* internos.

O *Motion Detection* que realizou a subtração de imagens sequenciais utilizou a função cvAbsDiff do JavaCV, que possuía como parâmetros 3 campos para serem postas a segunda imagem recebida, a primeira imagem e o resultado da operação. Logo após a função de ação de subtração, foram realizadas as operações de transformação morfológica, suavização e a binarização da imagem resultante para diminuir o seu "peso", conforme ilustrado no trecho de código da Figura 5.10.

```
if (prevImage != null) {
    // perform ABS difference
    cvAbsDiff(grayImage, prevImage, diff);

    //aplica a transformacao morfologica e a suavizacao
    diff = pre_proc.Transformacao_morfologica(diff, 3);
    diff = pre_proc.Suavizacao(diff, 3);
    //binarizacao da imagem resultante
    cvThreshold(diff, diff, 20, 255, CV_THRESH_BINARY_INV);
}
```

Figura 5.10 Trecho do código da extração de características com a função cvAbsDiff.
Fonte: PRÓPRIA.

O *Motion Detection* que realizou a Mistura *Gaussiana* de imagens utilizou a função BackgroundSubtractorMOG2 do JavaCV, que possuía como parâmetros 2 valores de limiares inferior e superior para realizar o corte na imagem e 1 valor que indica se serão mantidos os valores dos *pixels* que se modificaram e que possuíam menor intensidade. Normalmente estes *pixels* que possuem menor intensidade representam sombras e neste trabalho foram descartadas. Logo após a função de ação de subtração foi realizada a transformação morfológica com fator = 3, a suavização com fator = 5 e a binarização da imagem resultante para diminuir o seu peso, conforme ilustrado no trecho de código da Figura 5.11.

Este algoritmo, apesar de reduzir o volume de *pixels* na imagem que foi apresentada ao processo de reconhecimento de gestos ainda podia apresentar alguns elementos que não diziam respeito ao gesto propriamente dito, como por exemplo, a roupa do usuário ou outro objeto que poderia estar se movimentando nas imagens capturadas e que apenas aumentariam a necessidade de processamento de *pixels* que não são aproveitados no final. Por esta razão, o *Motion Detection* foi associado ao *Skin Detection* para realizar a segmentação da imagem

resultante do *Motion Detection* eliminando os *pixels* da imagem que não possuíam valores dentro dos limites mínimo e máximo do que foi considerado tom de pele.

```
static IplImage Mistura_Gaussiana(IplImage EntradaImage, int width, int height){
    IplImage ForegroundImage         = IplImage.create(width, height, IPL_DEPTH_8U, 1);

    mog.apply(EntradaImage, ForegroundImage, -1); // aplicacao da mistura gaussiana
    mog.getBackgroundImage(ForegroundImage);      //remocao dos elementos estáticos da imagem final
    //binarizacao da imagem resultante
    cvThreshold(ForegroundImage, ForegroundImage, 165, 255, CV_THRESH_BINARY);

    //aplica a transformacao morfologica e a suavizacao
    ForegroundImage = pre_proc.Transformacao_morfologica(ForegroundImage, 7);
    ForegroundImage = pre_proc.Suavizacao(ForegroundImage, 3);

    //inversao da imagem
    cvNot(ForegroundImage, ForegroundImage);

    return ForegroundImage;
}
```

Figura 5.11 Trecho do código da Mistura *Gaussiana* com a função getBackgoundImage.
Fonte: PRÓPRIA.

O *Skin Detection* é um processo de extração de características através dos padrões das cores e sua aplicação foi experimentada em Simoes *et al*. (2013). Para o processo ser realizado foi necessário definir um limiar de corte mínimo e outro máximo e os valores de cores em uma imagem de formato HSV que não se encaixassem neste grupo eram automaticamente descartados, conforme ilustrado no trecho de código da Figura 5.12.

```
static IplImage Skin(IplImage EntradaImage, int width, int height){
    IplImage HSVImage         = IplImage.create(width, height, IPL_DEPTH_8U, 3);
    IplImage SkinImage        = IplImage.create(width, height, IPL_DEPTH_8U, 1);

    //converter a imagem RGB para HSV
    cvCvtColor(EntradaImage, HSVImage, CV_RGB2HSV);

    //limiares minimo e maximo manipulando os componentes de R (red), G (green) e B (blue)
    CvScalar rgba_min = cvScalar(r, g, b, 0);
    CvScalar rgba_max = cvScalar(rr, gg, bb, 0);
    //gerando a imagem em escala de cinza dos elementos que estavam
    //nos limites minimo e maximo
    cvInRangeS(HSVImage, rgba_min, rgba_max, SkinImage);
    //binarizacao da imagem resultante
    cvThreshold(SkinImage, SkinImage, 25, 255, CV_THRESH_BINARY_INV);

    return SkinImage;
}
```

Figura 5.12 Trecho do código do *Skin Detection* com a função cvInRangeS.
Fonte: PRÓPRIA.

O modo adotado foi realizado da seguinte forma:

- A imagem originalmente em formato RGB foi convertida para HSV;

- Aplicado um limiar de tons de pele com os valores r=25,g=55,b=5 para o limiar mínimo e rr=160, gg=255, bb=190 para o limiar máximo.

A imagem resultante, em escala de cinza, era a combinação da aplicação dos dois limiares através da função cvInRangeS. Logo após a construção da imagem resultante em cinza, ela era submetida ao processo de binarização.

As imagens finais do processo de *Motion Detection* e do *Skin Detection* foram combinadas através da função *cvAnd* que manteve na imagem final os *pixels* que se movimentaram e tinham tom de pele.

O reconhecimento, interpretação e acompanhamento dos gestos utilizando o classificador *Haar-Like* foram feitos deslizando uma janela de busca através da imagem, verificando se uma região da imagem em certo local podia ser classificada como o gesto do classificador ou não. A Figura 5.13 ilustra o trecho do código responsável em fazer a busca do gesto e desenhar a janela de marcação ao redor do mesmo se for encontrado.

```
double fator_redimensionamento = 1.1;
int num_quadros = 4;
CvSeq gesto = cvHaarDetectObjects(inImage, classifier, storage, fator_redimensionamento, num_quadros, CV_HAAR_DO_CANNY_PRUNING);

cvClearMemStorage(storage);

//desenha um quadro para o tracking do gesto reconhecido
if (!gesto.isNull()){
    int totalGestos = gesto.total();
    for (int i = 0; i < totalGestos; i++) {
        CvRect r = new CvRect(cvGetSeqElem(gesto, i));
        //dimensao do quadro
        int x = r.x(), y = r.y(), w = r.width(), h = r.height();
        cvRectangle(outImage, cvPoint( x, y ), cvPoint(x + w, y + h ),colour, 3, CV_AA, 0);
    }
}
```

Figura 5.13 Trecho do código do reconhecimento e *tracking* do *Haar-Like* do JavaCV.
Fonte: PRÓPRIA.

Após o reconhecimento do gesto ser realizado pelo Haar-Like, é repassada a região-alvo para o método de *CamShift*. O *CamShift* se utilizou de um modelo de remoção de plano de fundo através de cálculo de padrões nas imagens que foram submetidas e armazenadas em um vetor para se observar o que foi mantido em uma sequência. A função *cvCalcBackProject* realizou este processo e atualizou o histograma que foi utilizado no processo de rastreio da função *cvCamShift* do JavaCV, que ficava responsável em utilizar o histograma como modelo a ser buscado nas imagens. O trecho de código da Figura 5.14 ilustra o uso das funções no processo do *CamShift*.

```
cvCalcBackProject( imageArray, BackImage, hist );
cvAnd( BackImage, MaskImage, BackImage, null );
cvCamShift(BackImage, track_window,
        cvTermCriteria( CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, 10, 1 ),track_comp,track_box);
```

Figura 5.14 Trecho do código do reconhecimento e *tracking* do *Haar-Like* do JavaCV.
Fonte: PRÓPRIA.

A Figura 5.15 mostra os itens definidos neste trabalho e os que foram adotados dos trabalhos relacionados, acompanhados das respectivas aplicações sobre uma imagem digital de entrada. As 20 pessoas do grupo da Engenharia da Usabilidade participaram de cada etapa para evitar que os classificadores ficassem direcionados somente a uma pessoa.

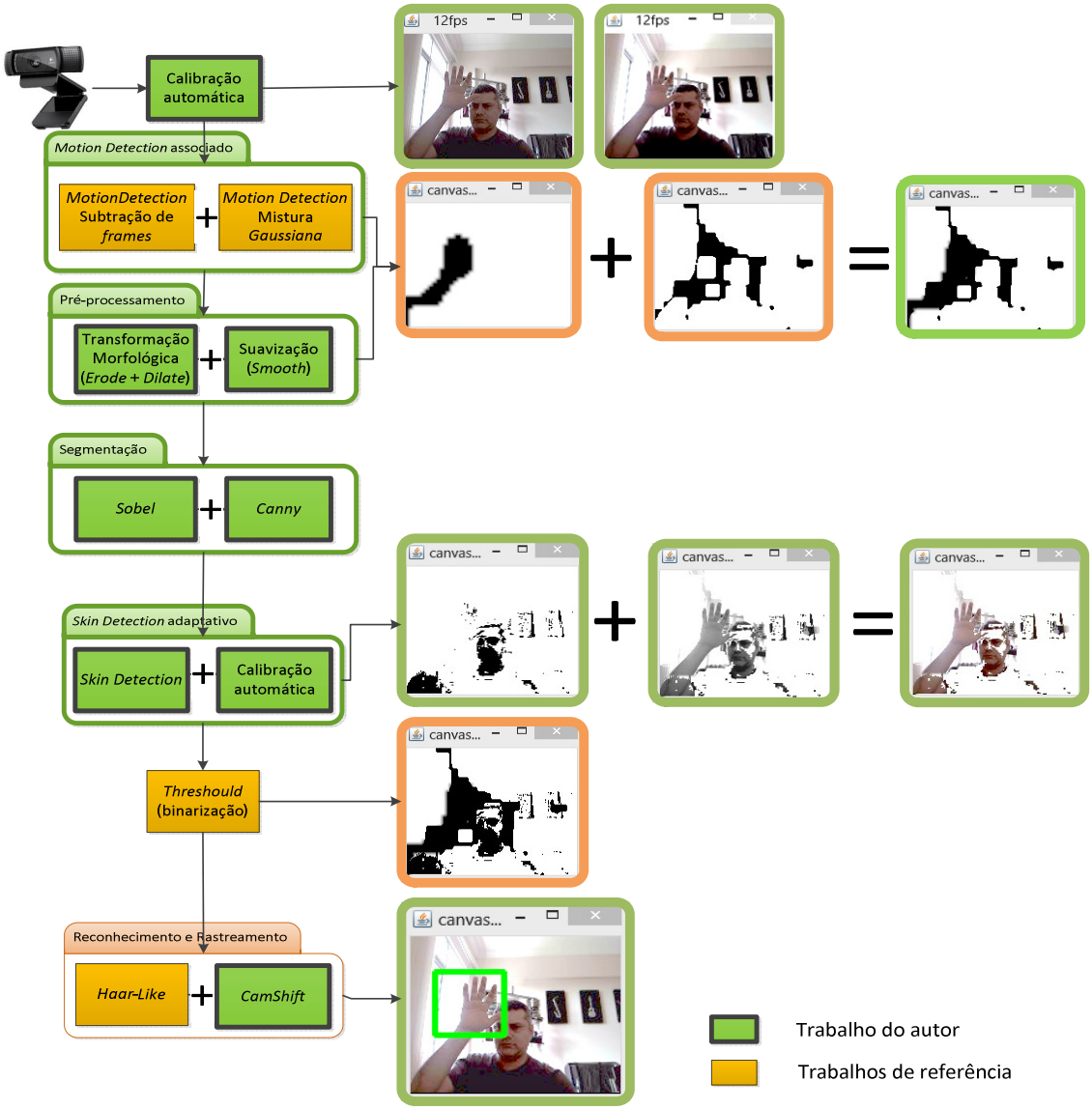


Figura 5.15 Divisão de itens construídos e os adotados dos trabalhos relacionados.
Fonte: PRÓPRIA.

A associação das diversas técnicas atacaram os dois principais problemas da VC que eram os processamentos das imagens para eliminar os *pixels* que não participam do processo de reconhecimento e o custo computacional para reconhecer automaticamente as imagens submetidas.

Devido à velocidade do método em relação ao uso de qualquer classificador, a técnica de *Motion Detection* foi também utilizada para ativar o sistema de reconhecimento de gestos. O modo de funcionamento ficou assim descrito: ao se movimentar a mão durante 4 segundos, o sistema ativava o algoritmo que realizava o reconhecimento dos gestos.

5.2.4 Integração da TVDi com a Visão Computacional

A etapa final do trabalho consistiu em integrar os protótipos de TVDi com a VC, criando um único protótipo que pudesse realizar a prova de conceito, respeitando as delimitações dos objetivos específicos deste trabalho e as limitações e restrições de cada etapa abordada.

A interação do usuário ocorria dentro de dois cenários: um físico e um lógico. O cenário físico, estão ilustrados na Figura 5.16, tratou das características do local onde a TV está localizada, observando o distanciamento mínimo e máximo alcançados pelo sistema de Visão Computacional. O cenário lógico tratou dos *softwares* e *hardwares* envolvidos que permitiram realizar a interação gestual com a TV.

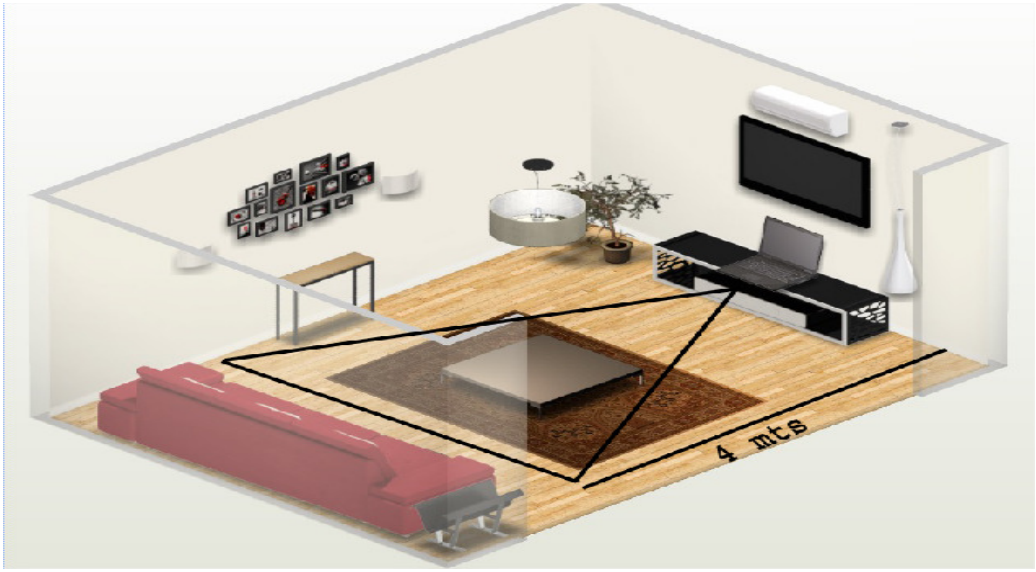


Figura 5.16 Cenário de uso da integração dos protótipos de TV e VC.
Fonte: PRÓPRIA.

Em Simoes *et al*. (2013), foi definido um modo para se realizar o relacionamento entre a TVDi e a Visão Computacional utilizando um arquivo XML. Este arquivo XML é apresentado na Figura 5.17 que traz as *tags coordenates* e *action*. As *tags coordenates* com as *sub-tags* x e y indicam as coordenadas de localização do gesto na imagem capturada pelo sistema de VC e a *tag action* indica através de um código numérico o que devia ocorrer no protótipo da TVDi.

```
<?xml version="1.0" encoding="UTF-8"?>
<gesture>
  <coordenates>
    <x>10</x>
    <y>10</y>
  </coordenates>
  <action>
    <action>1</action>
  </action>
</gesture>
```

Figura 5.17 Arquivo XML preenchido pelo *software* de VC.
Fonte: PRÓPRIA.

Para obter um melhor resultado era fundamental que a câmera estivesse posicionada no centro do eixo horizontal da tela da TVDi. A movimentação lateral do usuário gerava uma variação no eixo x e a movimentação na vertical gerava uma variação no eixo y. A aproximação ou distanciamento do usuário em relação a câmera modificava o tamanho do gesto, porém o fator de redimensionamento do classificador *Haar-Like* tratava desta característica aceitando variações de até 10% do tamanho padrão.

O usuário deveria estar posicionado também dentro da região interna de captura da câmera para que seu gesto fosse capturado e consequentemente reconhecido. As distâncias mínima e máxima que o usuário deveria ficar de sua TV era 1,50 e 4,00 metros. Esta distância está compatível com outros sistemas de reconhecimento de gestos como o *Kinect*® (YI, 2012) que trabalha dentro das distâncias mínima e máxima de 1 e 3.

5.3 Conclusão

Neste capitulo foram apresentadas as implementações dos protótipos que permitiram experimentar a arquitetura de sistema definida no Capítulo 4. Foram utilizados diagramas de classes e diagramas de estados para demonstrar quais eram os elementos que participavam em cada ambiente e como estes se comportavam ao longo da execução.

A interação gestual entre a TVDi e a VC se deu através de um *layout* da TVDi e de um conjunto de gestos que foram definidos pelos usuários participantes do processo de Engenharia da Usabilidade. A integração levou em consideração características de *hardware* e *software* (*middleware*), principalmente no que se referiu à conexão de câmeras de forma direta aos equipamentos de TV ou *set-top boxes* e as necessidades de processamento e armazenamento de imagens em alta resolução em tempo real.

Foi observado que as atuais TVDi e *set-top boxes* disponíveis não possuíam conexão de câmeras diretamente as suas portas USB e também não possuíam uma configuração capaz de lidar com o processamento de imagens com grande resolução, necessárias para realização do reconhecimento de gestos apresentados a certa distância da câmera. Assim, decidiu-se construir um *software* que fosse executado em um computador para processar as imagens e reconhecer os gestos ali inseridos, fornecendo os resultados desse reconhecimento em um arquivo XML para os *middlewares* de TV em MHP e em GINGA-NCL.

O protótipo de VC, construído com o uso do *wrapper* JavaCV forneceu classificadores do tipo *Haar-Like* que reconheciam o conjunto de gestos mesmo que estes fossem apresentados em ambientes com ruídos e outros elementos que causassem alguma dificuldade no processo. As preocupações do processamento de imagens eram a quantidade de *pixels* entregues ao processo de reconhecimento e ao peso que as imagens representam para o sistema.

O próximo passo é realizar uma avaliação da arquitetura de sistema e os protótipos construídos. Esta avaliação é detalhada no Capítulo 6, onde são realizados testes e apresentados os resultados obtidos. Estes resultados servem para realizar as análises comparativas com os trabalhos relacionados apresentados no Capítulo 3.

Capítulo 6- Testes e Resultados

Este capítulo apresenta os testes e avaliações realizados sobre os protótipos de TVDi e Visão Computacional, propostos no Capítulo 5. A aplicação dos testes tem como principal característica gerar os resultados para serem utilizados na avaliação deste trabalho com os resultados apresentados pelos trabalhos de referência.

A forma adotada para a realização dos testes foi a de aplicar testes em cada subsistema construído, utilizando uma série de métricas junto aos usuários de teste. O sistemas de TVDi foi testado em relação ao seu *layout* e quanto a operabilidade e satisfação atingidas junto ao grupo de testes. O sistema de Visão Computacional foi testado em relação ao classificador e em relação ao conjunto de abordagens dadas sobre as imagens para eliminação dos pixels que não participavam do processo de reconhecimento de gestos.

A avaliação dos resultados obtidos dos testes sobre os protótipos foram comparados com os resultados experimentados e/ou divulgados pelos trabalhos de referência, apresentados no Capítulo 3.

6.1 Cenário Ajustar Volume e Trocar Canal da TVDi

O cenário no qual o usuário ajusta o volume do som e troca os canais de programação da TVDi, foi apresentado no Capítulo 4 e construído no Capítulo 5. Neste cenário o usuário apresentava os gestos para a câmera que entregava a imagem ao computador que continha o protótipo de VC. O protótipo de VC realizava os processamentos de tratamento de imagem e de reconhecimento de gestos. Caso o gesto executado fosse reconhecido, um conjunto de informações contendo o código e a coordenada (x,y) deste gesto era salvo em um arquivo XML. Este arquivo XML era disposto em um diretório controlado por um servidor de aplicação e era acessado pelo protótipo de TVDi através do endereço IP da máquina servidora. Tão logo o usuário executasse o gesto era dada uma informação visual ao usuário através do uso de um cursor animado em formato de mão.

6.1.1 Considerações sobre o protótipo

Para uniformizar as estratégias e os resultados obtidos, os testes seguiram um mesmo padrão sobre o protótipo de TVDi emulado e o protótipo executado em um *set-top box* real.

Para dar a noção visual do *status* do protótipo de TVDi ao usuário, quando este era executado exibia-se um pequeno ícone no canto superior esquerdo da tela. Após a interação do usuário era exibida a janela de serviços, localizada no canto inferior esquerdo, que consistia de um painel contendo os botões de volume do som e canal. Após a escolha de uma ação, a janela de serviços ficava novamente oculta voltando a ter somente o ícone do canto superior esquerdo como indicativo de que a aplicação continuava ativa.

O protótipo de visão Computacional deveria estar operacional no computador e com a câmera posicionada no centro da tela da TV para que o usuário fosse detectado.

6.1.2 Testes e Resultados sobre os Protótipos de TVDi e VC

Os testes realizados sobre os protótipos de TVDi buscaram aplicar as métricas de operabilidade e satisfação direcionadas ao contexto de seu uso. A métrica de operabilidade considerou o tempo gasto para a conclusão de cada fase e a métrica satisfação foi medida através da avaliação do questionário respondido pelo usuário ao final do teste (ISO9241, 1998).

O *layout* foi desenvolvido para atender as duas principais tarefas que eram o ajuste do volume do som e a troca dos canais de programação, que serviu de ambiente para ser controlado pelo algoritmo de reconhecimento de gestos.

Os usuários da fase de testes foram os mesmos que participaram da fase de elaboração da linguagem gestual. Estes usuários foram mantidos nos mesmos grupos da fase de definição do *layout* e dos gestos, os que possuíam experiência em uso de dispositivos com interação gestual e os que não possuíam experiência no uso de tais recursos. Os testes tiveram caráter formal, onde cada usuário recebeu um formulário contendo as funcionalidades de ajuste de volume e de troca de canal e de *layout*, que foi preenchido ao final de sua experiência de uso do protótipo. Ou seja, a TV juntamente com os equipamentos necessários a interação através de gestos foram entregues aos usuários para que os mesmos os utilizassem em um ambiente que simulasse as condições reais de uso da TV.

Um modo de se medir a operabilidade e satisfação foi verificando o grau de dificuldade em utilizar o sistema de VC para executar o ajuste de volume do som e a troca de canais. Os usuários de início não se posicionavam corretamente em frente à câmera, o que causava uma falha na captura do gesto por este estar sendo exibido em uma região fora do alcance da câmera. Para medir o nível de dificuldade de execução de cada comando, os tempos gastos por cada usuário foram medidos e mostrados na Tabela 6.1.

Tabela 6.1. Resultados dos tempos de execução de cada comando.
Fonte: PRÓPRIA.

Usuários	Aumentar Volume (s)	Diminuir Volume (s)	Trocar canal (s)
Usuário 1	1,5	2,0	1,0
Usuário 2	1,8	2,0	1,5
Usuário 3	1,6	2,4	1,6
Usuário 4	2,6	4,0	2,6
Usuário 5	1,6	3,0	2,2
Usuário 6	1,6	1,8	3,0
Usuário 7	2,7	2,7	2,5
Usuário 8	3,1	2,5	2,8
Usuário 9	3,2	2,7	3,7
Usuário 10	3,2	2,5	2,0
Usuário 11	2,7	3,0	2,2
Usuário 12	3,2	4,0	2,0
Usuário 13	3,3	2,5	1,5
Usuário 14	3,3	2,0	4,2
Usuário 15	2,2	2,5	3,0
Usuário 16	1,6	2,5	1,4
Usuário 17	1,2	2,0	2,0
Usuário 18	1,5	2,6	1,1
Usuário 19	1,2	1,5	1,1
Usuário 20	1,3	2,0	1,5

A média dos tempos encontrados mostrou que a maioria das pessoas não teve dificuldades em utilizar o protótipo. A média dos tempos para se executar as operações está descrita na Tabela 6.2.

Tabela 6.2. Média dos tempos de execução de cada comando.
Fonte: PRÓPRIA.

Aumentar Volume (s)	Diminuir Volume (s)	Trocar canal (s)
2,2	2,5	2,1

Para medir o grau de satisfação do grupo de usuários, estes foram submetidos a um questionário pós-teste sobre o relacionamento com a aplicação de TV e o acionamento dos comandos através de gestos. O questionário pós-teste pode ser encontrado no Apêndice C. como resultado, o protótipo gerou um bom grau de satisfação diante o grupo de avaliadores.

Nos testes realizados pelos usuários, os protótipos executados a partir do emulador *XletView* e do *set-top box* levavam menos de 1 segundo para acessar o servidor *Apache* que fornecia o arquivo XML contendo a referência do gesto reconhecido para o *middleware* de TV e a coordenada onde o mesmo foi localizado.

O acesso realizado por ambos os protótipos de TV ao servidor *Apache* ocorreu sem maiores problemas e registraram um atraso inferior a 1 segundo. Este atraso foi causado pelo tempo necessário para o *set-top box* acessar via rede de dados o servidor *Apache* e ler o arquivo XML. Este modo de funcionamento foi a solução encontrada para se comandar o protótipo de TV através de gestos, pois os *set-top boxes* e as TVs ainda não permitem a conexão de câmeras em suas portas USB, necessárias em aplicações de VC.

O conjunto de gestos foi experimentado inicialmente em um protótipo que utilizava o *CamShift* para testar se um gesto gerava confusões com os demais. O *CamShift* foi o modelo escolhido para testar os gestos definidos pelo grupo de estudo por ser uma abordagem mais simples quando comparada a outras técnicas que necessitam de fase de treinamento. Cada um dos cinco gestos foi testado e verificado o nível de acerto dos mesmos para que eles pudessem ser efetivados no trabalho.

O protótipo de VC, construído em JavaCV, foi testado em relação ao seu classificador e em relação ao modo adotado para o processamento de imagens e reconhecimento de gestos. A versão final apresentou uma taxa de atualização média de 26 *frames* por segundo, o que, juntamente com a aplicação da técnica do *CamShift* trouxe uma sensação de fidelidade na movimentação do cursor durante os testes.

Para medir a eficiência do classificador construído neste trabalho foram realizados testes de reconhecimento sobre um banco de imagens que continha 1000 arquivos de cada gesto. Estes arquivos foram gerados juntamente com os 2000 arquivos que foram utilizados para a construção do classificador e abrangem pessoas de diferentes raças sob planos de fundo com cores e iluminações variadas. Estes arquivos não foram utilizados no classificador, sendo guardados para o teste de reconhecimento e definição da precisão do classificador.

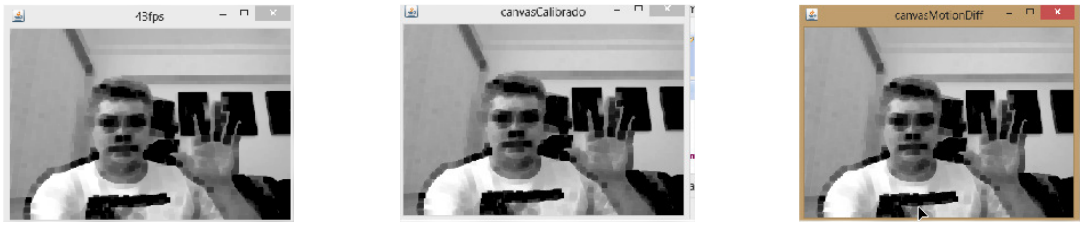
A Tabela 6.3 mostra os resultados comparativos dos classificadores de cada gesto construído para este trabalho com os trabalhos de referência.

Tabela 6.3. Resultados obtidos nos testes de eficiência dos classificadores construídos.
Fonte: PRÓPRIA.

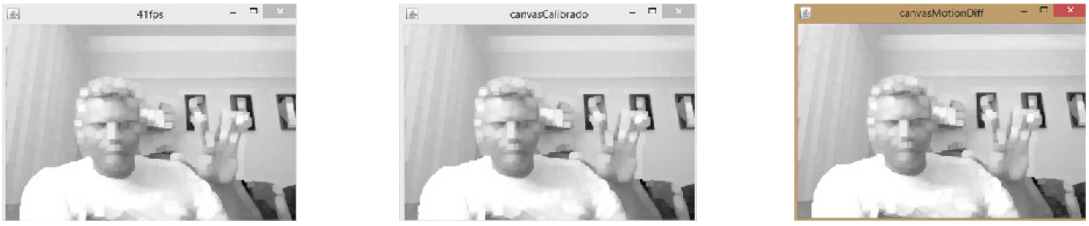
Resultados		
Gestos	***True positive***	***False positive***
Gesto a	92,2%	7,8%
Gesto b	92,8%	7,2%
Gesto c	97,0%	3,0%
Gesto d	93,3%	6,7%
Gesto e	99,0%	1,0%

Para medir a eficiência do processamento digital das imagens adotadas neste trabalho foram realizados testes referentes às etapas de pré-processamento, segmentação, extração de características e uso dos classificadores.

Na etapa de pré-processamento foram testados os valores dos componentes da transformação morfológica e observado se os gestos continuavam sendo percebidos pelo algoritmo de reconhecimento. A melhor configuração observada nos testes foi a que aplicou o fator de erosão igual a 3 e a dilatação igual a 3, pois as demais eliminavam partes importantes dos gestos. Os testes foram realizados com todos os gestos mapeados e os resultados ficaram muito próximos. A Figura 6.1 mostra as variações dos valores do componente de erosão e dilatação.



Erode = 1 *Erode* = 2 *Erode* = 3



Dilate = 1 *Dilate* = 2 *Dilate* = 3

Figura 6.1 Teste de aplicação dos filtros de Erosão e Dilatação.
Fonte: PRÓPRIA.

Ainda na etapa de pré-processamento foram testados os parâmetros do filtro de suavização. A Figura 6.2 mostra a imagem resultante da aplicação de suavização.

Smooth = 1 *Smooth* = 2 *Smooth* = 3

Figura 6.2 Teste de aplicação dos filtros de suavização.
Fonte: PRÓPRIA.

A melhor configuração do filtro de suavização observada nos teste foi a que utilizou o fator igual a 2 pois não eliminou os valores dos *pixels* dos gestos mapeados e uniformizou os restantes dos *pixels*. Os demais gestos também foram submetidos ao teste e seus resultados foram semelhantes ao gesto ilustrado na Figura 6.2.

Para a etapa de segmentação foram testados os parâmetros dos filtros de bordas *Sobel* e *Canny*. A Figura 6.3 mostra a ação desses filtros modificando as imagens e fornecendo uma nova imagem resultante com as bordas melhor definidas.

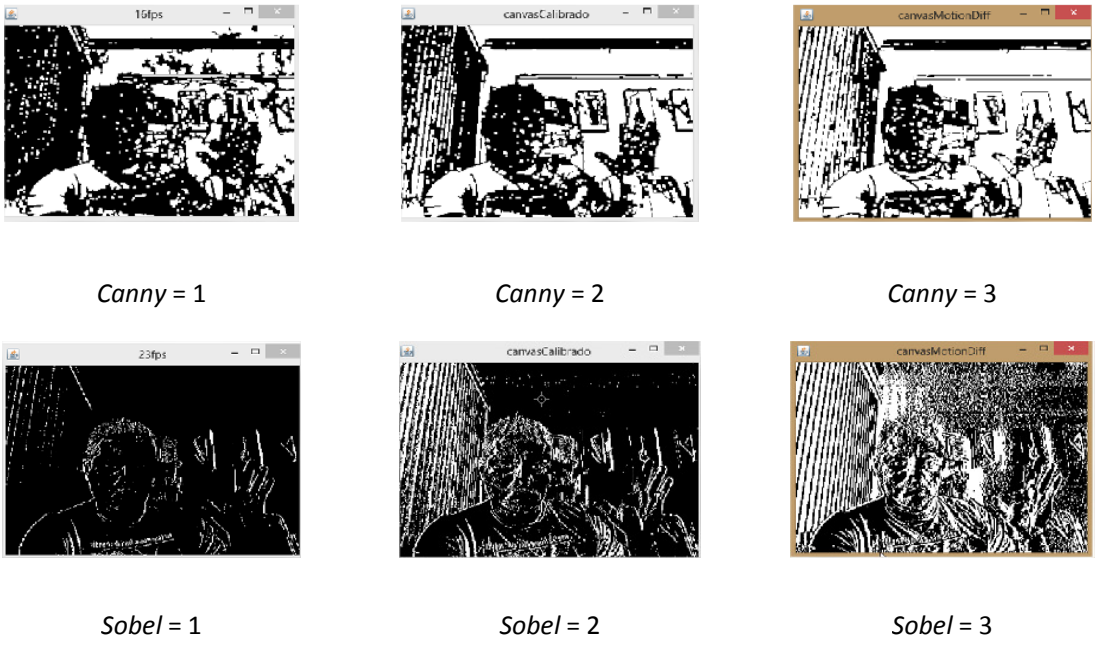


Figura 6.3 Teste de aplicação dos filtros de *Canny* e *Sobel*.
Fonte: PRÓPRIA.

A melhor configuração foi a que possuía o filtro *Sobel* com o fator igual a 3 e o *Canny* igual a 120, também com o número mínimo de vínculos entre os pontos igual a 3. Este arranjo forneceu um reforço nas bordas dos gestos e eliminou os elementos que não se encaixavam neste padrão. Os demais gestos também foram submetidos ao teste e seus resultados foram semelhantes ao gesto ilustrado na Figura 6.3.

A extração de características utilizou duas técnicas: o *Motion Detection* e o *Skin Detection*. A associação destas técnicas buscou eliminar os *pixels* estáticos e que não de encaixavam nos limiares mínimo e máximo do arranjo de cores definido como tom de pele.

O *Motion Detection* aplicou duas técnicas, a de detecção das bordas e a de detecção dos *pixels* internos, conhecida como *Mistura Gaussiana*. A melhor configuração para o *Motion Detection* foi a que utilizou a diferença entre os frames com o fator igual a 3, a Mistura *Gaussiana* com o fator de transformação morfológica igual a 5 e a suavização igual a 3, pois foram os modelos que mantiveram a quantidade de *pixels* que se movimentaram (mudaram de valor quando observados frames em uma sequencia). Os testes realizados sobre os parâmetros destas duas técnicas estão representados na Figura 6.4.

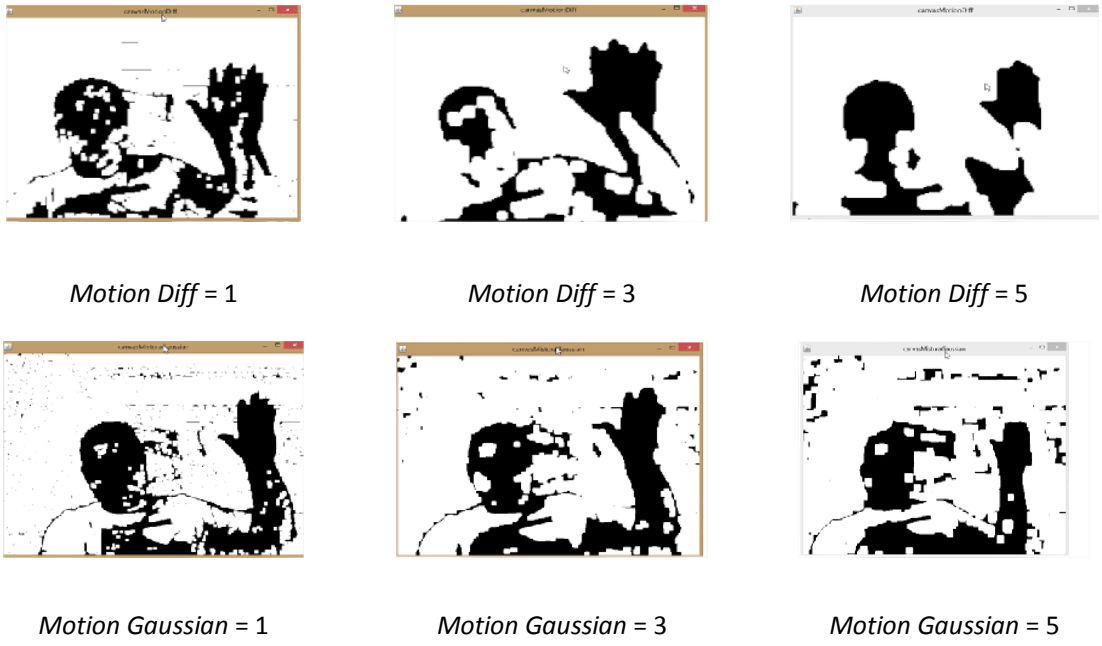


Figura 6.4 Teste de aplicação para a função de detecção de movimentos.
Fonte: PRÓPRIA.

O *Skin Detection* utilizou o limiar inferior em seus componentes de r=25, g=55 e b=5 e o limiar superior em seus componentes de rr=160, gg=255, bb=190. Estes valores foram adotados após a aplicação dos testes apresentados na Figura 6.5.

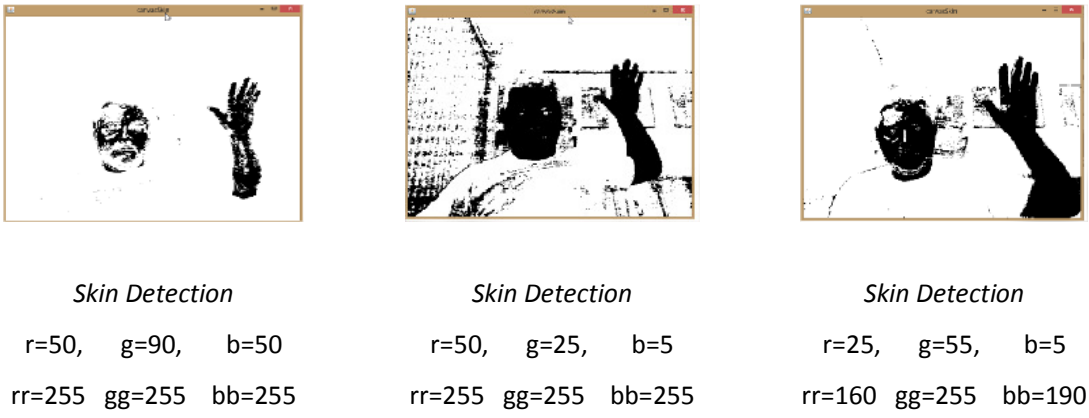


Figura 6.5 Teste de detecção de movimentos pela diferença entre *frames*.
Fonte: PRÓPRIA.

Após os testes que utilizaram somente os classificadores *Haar-Like*, foram adicionados os demais recursos testados da extração de características com a técnica de *Motion Detection, Skin Detection* e o *CamShift*. Os resultados destes testes estão demonstrados na Figura 6.6.

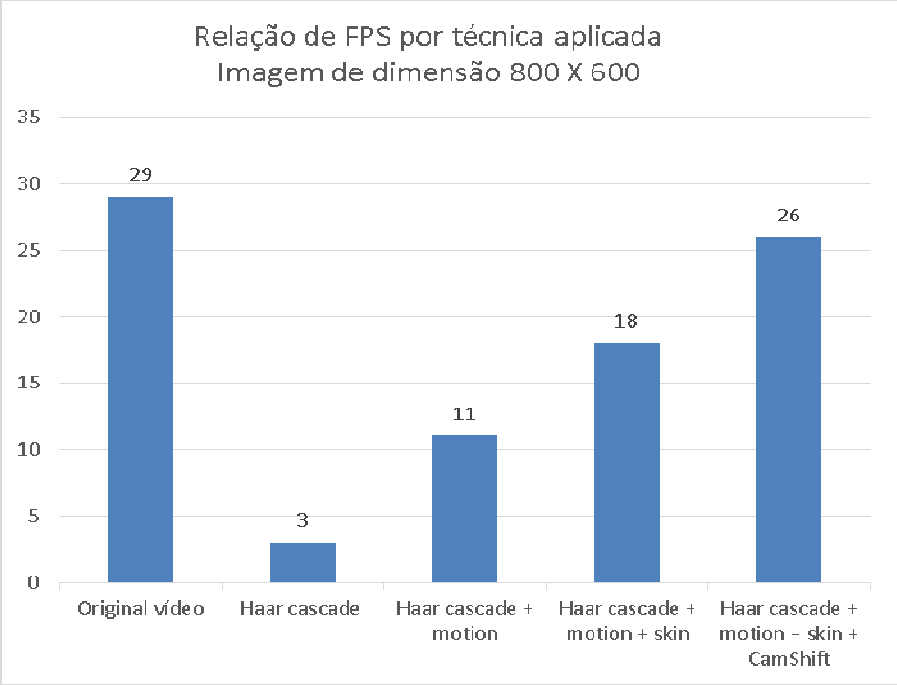


Figura 6.6 Diagrama comparativo dos desempenhos dos métodos adotados no trabalho.
Fonte: PRÓPRIA.

Notou-se que o *Haar-Like* teve um consumo bastante elevado de processamento quando realizou a tarefa de rastreio do gesto mapeado no classificador. Esta percepção foi explicitada pela contagem de *frames* por segundo que puderam ser processados por cada método. Este esforço foi diminuído com a associação das técnicas de *Motion* e *Skin Detection* que reduziram a quantidade de *bits* que foi submetida ao processo de comparação com o classificador, mas que ainda ficou bastante distante do desempenho oferecida pela câmera. Além do fator de desempenho, o classificador também foi prejudicado com as mudanças de iluminação, de mudança de ângulos e rotações dos gestos apresentados. Como o valor de desempenho buscado em processamento de imagens de tempo real é o apresentado pelo dispositivo de captura, foi aplicada a técnica de *CamShift*, que é uma técnica de rastreio de gestos ou objetos de custo constante após o classificador *Haar-Like* realizar o rastreio dos gestos mapeados nos classificadores.

A adição do *CamShift*, retirando a tarefa de rastreamento do *Haar-Like* apresentou um desempenho de 26 frames por segundo, ou seja, um ganho de desempenho de 44,44% sobre o classificador *Haar-Like*.

Num primeiro momento, o funcionamento do protótipo foi definido de modo a executar de imediato a ação na TV assim que um gesto fosse reconhecido. Esta forma não se

mostrou ser a mais indicada, uma vez que, em alguns momentos o usuário pode apresentar gestos de forma involuntária, que não indicam a vontade de executar alguma ação na TV. Este modo de funcionamento foi modificado, dando 3 segundos para o gesto ser executado sobre a opção no *layout* do protótipo de VC.

6.1.3 Avaliação

Para avaliar o nível de dificuldades na execução de cada comando gestual no protótipo de TV e o grau satisfação geral no grupo de usuários, mesmo quando comparado a produtos já consolidados no mercado, foram utilizados os trabalhos (YI, 2012), (BRACKMANN, 2010), (BARROS, 2006) e (FREEMAN *et al*., 2001) como referência. A Tabela 6.4 ilustra a comparação dos trabalhos adotados como referência e os resultados obtidos e testados por este trabalho.

Tabela 6.4.Comparação entre os trabalhos relacionados em Engenharia da Usabilidade.
Fonte: PRÓPRIA.

Fonte	Tempo de execução (s)
(YI, 2012)	3,0 s
(BRACKMANN, 2010)	4,5 s
(BARROS, 2006)	5,0 s
(FREEMAN *et al*., 2001)	4,0 s
Este trabalho	**3,7 s**

O trabalho de referência (YI, 2012) utilizou um equipamento que possui *hardware* e *softwares* responsáveis pelo tratamento de luminosidade e de remoção do plano de fundo. Os trabalhos de (BRACKMANN, 2010), (BARROS, 2006) e Freeman *et al.* (2001) foram construídos para elaborar um melhor método de IHC (interação homem-computador) entre o usuário e a TV Digital Interativa. Eles prospectaram a construção de protótipos de controle remoto com um *layout* que tivesse à disposição, de forma mais direta, as ações que estão se tornando comuns em TV como, por exemplo, os teclados virtuais, movimentação de objetos virtuais na tela, etc., e que não estão disponíveis no menu principal.

Para avaliar os protótipos construídos sobre a as ferramentas MHP e NCL-Lua foram utilizados os trabalhos de Lima *et al.* (2013), Brito (2012) e Wu et. Al (2014) . Os testes foram direcionados ao acesso de recursos físicos no *set-top box* real e no emulador.

O trabalho relacionado que mais se aproximou deste trabalho foi o descrito em Wu *et al.* (2014), porém este está limitado ao ambiente de aplicação procedural desenvolvido em DVB-MHP, acessado através do emulador *XletView*.

Para avaliar o classificador foram utilizados os trabalhos de Wilson *et al.* (2009), Rautaray *et al.* (2012) e Dardas *et al.* (2007). Os algoritmos foram implementados para que os classificadores pudessem ser testados. Foram reproduzidos e parametrizados os *softwares* dos trabalhos relacionados de acordo com as descrições feitas pelos autores.

A Tabela 6.5 mostra os resultados obtidos dos classificadores de cada gesto construídos para este trabalho com os trabalhos de referência.

Tabela 6.5.Comparação dos resultados dos classificadores.
Fonte: PRÓPRIA.

Resultados								
	Este Trabalho		(WILSON *et al.*, 2009)		(RAUTARAY *et al.*, 2012)		(DARDAS *et al.*, 2007)	
Gestos	***True positive***	***False positive***	***True positive***	***False positive***	***True positive***	***False positive***	***True positive***	***False positive***
Gesto a	**92,2%**	**7,8%**	85,0%	7,0%	91,5%	8,5%	93,4%	6,6%
Gesto b	**92,8%**	**7,2%**	91,2%	8,8%	90,4%	7,6%	91,1%	9,9%
Gesto c	**97%**	**3,0%**	89,0%	5,0%	92,5%	7,5%	92,3%	7,7%
Gesto d	**93,3%**	**6,7%**	89,4%	10,6%	91,2%	8,8%	92,8%	7,2%
Gesto e	**99,3%**	**0,7%**	95,3%	4,7%	95,1%	1,9%	94,8%	0,9%

Para avaliar os resultados do processamento digital foram utilizados os trabalhos de Jawas *et al.* (2013), Napoleon *et al.* (2013) e Silva *et al.* (2013) que executaram a aplicação de filtros de erosão, dilatação e suavização para eliminação de ruídos.

O trabalho de Jawas *et al.* (2013) aplicou o filtro de erosão com valor igual a 2 e a dilatação igual a 3 que funcionou bem com objetos apresentados a curta distância (cerca de 1 metro de distância da câmera) mas que não serviu para distâncias acima de 2 metros, pois os objetos já não apresentavam em destaque os seus limites com o plano de fundo. O trabalho de Napoleon *et al.* (2013) não aplicou um filtro de erosão ou dilatação de forma automática e sim oferecia os recursos para que um operador os acionasse de acordo com sua percepção visual. Percebeu-se que a aplicação isolada de apenas um dos filtros (erosão ou dilatação) causava a eliminação de valores de *pixels* importantes dos objetos e que somente com a combinação de

ambos foi possível destacar os objetos representados nas imagens. O trabalho de Silva *et al.* (2013) foi construído para realizar a limpeza de ruídos em imagens capturadas a longas distâncias (cerca de 20 metros). A esta distância, gestos pequenos eram facilmente eliminados da imagem final, como uma mão exibindo um gesto qualquer. Desta forma, a combinação de valores dos filtros de erosão e dilatação utilizados para eliminar os ruídos e manter os gestos na imagem final foram 2 e 15 respectivamente.

Para avaliar os resultados obtidos na etapa de segmentação, que tratou da aplicação de filtros de bordas para destacar gestos contidos nas imagens, foram utilizados os trabalhos de Han *et al*. (2012), Vairalkar *et al.* (2013) e Davoodianidaliki *et al.* (2013) que executaram a aplicação de filtros de *Sobel* e *Canny*. A Tabela 6.6 mostra os resultados comparativos das aplicações dos filtros neste trabalho com os trabalhos de referência.

Tabela 6.6.Comparação dos resultados de Segmentação.
Fonte: PRÓPRIA.

Fonte	Tempo de execução (s)
(HAN *et al.,* 2012)	0,05 s
(VAIRALKAR *et al.,* 2013)	0,07 s
(DAVOODIANIDALIKI *et al*., (2013)	0,06 s
Este trabalho	**0,05 s**

Para avaliar os resultados da etapa de extração de características através de movimentos foram utilizados os trabalhos de Migliore *et al*. (2006), Fujita *et al.* (2012) e Jagadesh *et al.* (2012). A Tabela 6.7 mostra os resultados comparativos das aplicações dos filtros neste trabalho com os trabalhos de referência.

Tabela 6.7.Comparação dos resultados da Detecção de Movimentos.
Fonte: PRÓPRIA.

Fonte	Velocidade da aplicação do filtro (fps)
(MIGLIORE *et al*., 2006)	30 fps
(FUJITA *et al*., 2012)	30 fps
(JAGADESH *et al*., 2012)	21 fps
Este trabalho	**26 fps**

Para avaliar os resultados obtidos da etapa de extração de características através de detecção de tons de pele foram utilizados os trabalhos de Ahmadi *et al.* (2009), Jagadesh *et al.*

(2012) e Bang-Hua *et al.* (2007). A Tabela 6.8 mostra os resultados comparativos das aplicações dos filtros neste trabalho com os trabalhos de referência.

Tabela 6.8.Comparação dos resultados da Detecção de Tons de Pele.
Fonte: PRÓPRIA.

Fonte	Identificação do tom de pele	Tempo de execução da ação (s)
(AHMADI *et al*., 2009)	96%	0,8 s
(JAGADESH *et al*., 2012)	88%	0,9 s
(BANG-HUA *et al*., 2007)	98%	0,7 s
Este trabalho	**86 %**	**0,6 s**

Para avaliar os resultados obtidos da etapa de extração de características através de detecção de gestos do *CamShift* foram utilizados os trabalhos de Nadgeri *et al.* (2010) e Araki *et al*. (2012). A Tabela 6.9 mostra os resultados comparativos das aplicações dos filtros neste trabalho com os trabalhos de referência.

Tabela 6.9.Comparação dos resultados do uso do *CamShift*.
Fonte: PRÓPRIA.

Resultados						
	Este Trabalho		(NADGERI *et al.,* 2010)		(ARAKI *et al*., 2012)	
Gestos	Localização do gesto	Tempo de execução do gesto	Localização do gesto	Tempo de execução do gesto	Localização do gesto	Tempo de execução do gesto
Gesto a	**96,7%**	**0,8 s**	96,0%	2,7 s	93,1%	1,5 s
Gesto b	**97,0%**	**0,9 s**	98,8%	1,5 s	97,2%	1,1 s
Gesto c	**97,6%**	**0,6 s**	97,4%	0,8 s	95,1%	1,0 s
Gesto d	**95,2%**	**0,7 s**	93,8%	1,7 s	94,3%	1,2 s
Gesto e	**99,1%**	**0,6 s**	97,5%	1,1 s	98,4%	1,6 s

Para avaliar a integração da TVDi com a VC foi realizada uma comparação com outros trabalhos. Os trabalhos utilizados foram: (DEVASENA *et al.,* (2013), (VATAVU, 2012) e Miranda *et al.* (2009). A Tabela 6.10 mostra os resultados obtidos das aplicações descritas nos trabalhos relacionados e este trabalho.

Tabela 6.10.Comparação dos resultados da Integração da TVDi e VC.
Fonte: PRÓPRIA.

Fonte	Localização dos gestos	Tempo de execução da ação (s)
(DEVASENA *et al.,* (2013)	68%	1 s
(VATAVU, 2012)	89%	3 s
(MIRANDA *et al*., 2009)	-	-
Este trabalho	**95%**	**1 s**

As tabelas utilizadas na avaliação mostraram a comparação das características dos trabalhos relacionados e o presente trabalho. O trabalho que mais se aproximou foi o (VATAVU, 2012) em relação à descrição da construção de protótipos de TV que utilizou recursos de VC e não foi possível embarcar diretamente o protótipo de VC devido a limitações dos equipamentos, seja pela desempenho ou pela falta de possibilidade de se vincular diretamente a câmera.

O trabalho (VATAVU, 2012) possui muitas limitações de reconhecimento dos gestos em ambientes com ruídos, sem controle de iluminação e estava limitado ao ambiente de aplicação procedural desenvolvido a linguagem utilizada para sua construção, a linguagem C, além de ser executado em equipamento especial, construído para comandar um modelo específico de TV.

6.2 Conclusão

Os algoritmos de TV MHP e GINGA-NCL foram construídos para serem o mais independente possível do processo de reconhecimento de gestos devido às características físicas encontradas nos *hardwares* das atuais TVs e *set-top boxes*. Assim, um *software* que ficou hospedado em um computador ficou com a obrigação de realizar o processamento das imagens e escrever em um arquivo XML os valores da coordenada e da referência do gesto encontrado nas imagens, que eram repassados para os protótipos de TV.

No caso da aplicação executada no *set-top box* real, que utilizou um roteador *wireless* para se vincular ao computador que processava as imagens, não foi notado nenhum atraso significativo em relação a aplicação executada no emulador, hospedado na mesma máquina onde se tinha a câmera conectada.

O protótipo de VC foi dividido em duas etapas: a de criação do classificador do tipo *Haar-Like* e o processamento de imagens. Na primeira etapa, que se preocupou em construir os classificadores, os mesmos foram testados com os gestos definidos pelo grupo de estudos formado utilizando as regras da Engenharia da Usabilidade, obtendo uma média de acertos em torno de 95%. Quando analisados isoladamente, o gesto (e), apresentado na Figura 5.3 obteve um nível de acerto em torno de 99,3%, uma taxa encontrada somente em sistemas que utilizam um volume de imagens muito maior para se construir o classificador. Na segunda etapa, que se preocupou com o processamento de imagens, foi realizada uma série de intervenções sobre as imagens submetidas ao processo de reconhecimento para diminuir seu peso de consumo de memória e de processamento, ocasionando imediatamente num aumento da velocidade do sistema.

Além da parte técnica e de programação, os protótipos também foram avaliados em relação a operabilidade e a satisfação dos usuários por meio de testes e avaliações que mediram o grau de dificuldade em se utilizar o sistema de TV com reconhecimento de gestos. Os testes foram realizados com o mesmo grupo de Engenharia da Usabilidade que ao final respondeu a um questionário sobre a sensação de uso.

Os tempos de operação das ações da TV através de gestos, descritos na Tabela 6.2, quando comparados aos tempos apresentados pelos trabalhos de referência e pelas normas de usabilidade, demonstrou que foram atendidos os requisitos estabelecidos pela Engenharia da Usabilidade.

Os resultados encontrados na fase de teste e avaliação demonstraram que foi possível realizar as operações da TV através de reconhecimento de gestos, atendendo as características de operabilidade e satisfação dos usuários e respondendo as questões apresentadas nos objetivos específicos.

Para finalizar este trabalho, no Capítulo 7 serão feitas as considerações finais, os pontos de melhoria e sugestões para os trabalhos futuros.

Capítulo 7- Considerações Finais

Este capítulo apresenta as conclusões, relacionando os objetivos com os resultados obtidos após a execução das avaliações sobre os protótipos e apresentar as dificuldades encontradas, as sugestões para trabalhos futuros e as contribuições para a sociedade e a ciência. O capítulo inicia com uma explicação conclusiva sobre os resultados, evidenciando a relevância de cada abordagem dada e os limites do desenvolvimento do trabalho.

Em seguida são apresentadas as dificuldades encontradas em cada etapa da construção da integração da TVDi com a visão Computacional e as decisões tomadas para que o produto desenvolvido não divergisse da arquitetura do sistema. Em seguida são apresentadas as sugestões para trabalhos futuros nas áreas de Engenharia da Usabilidade, TVDi e Visão computacional, além da integração das áreas, decorrentes das oportunidades que surgiram a partir dos resultados obtidos pelos testes e avaliações e que demandam um esforço de pesquisa adicional, onde novas proposições precisam ser testadas e validadas.

Por fim são apresentadas as contribuições deste trabalho em relação às áreas de Engenharia da Usabilidade, TVDI e Visão Computacional. Na área de Engenharia da Usabilidade as contribuições se fazem através de um conjunto mínimo de passos para a construção de *softwares* interativos. Na área de TVDi através de um conjunto de estratégias para aproximar os códigos dos middlewares MHP e Ginga. Na área de Visão Computacional com as definições de quantidades e resoluções de imagens para se construir um classificador Haar-Like e da aplicação de um conjunto de filtros para se obter com mais precisão os objetos ou gestos que se movimentaram e possuíam tons de pele em um conjunto de imagens.

7.1 Conclusões

Durante o levantamento bibliográfico, foram vislumbrados vários caminhos que poderiam ser trilhados para se atingir o objetivo principal do trabalho: investigar e propor a concepção de uma *interface* para controlar a TV Digital Interativa nos padrões de transmissão e *middleware* europeu e brasileiro, através de um conjunto de gestos capturados por uma câmera de baixo custo e convertidos em comandos básicos, servindo como ferramenta

auxiliar ou substituta ao controle remoto na ação de interação com a TVDi. Esses vários caminhos foram catalogados e tratados por área.

Para atender ao primeiro objetivo específico, buscou-se identificar na catalogação dos trabalhos relacionados, também conhecida como estado da arte, um conjunto mínimo de técnicas e ferramentas de Engenharia da Usabilidade, TVDi e Visão Computacional. Este objetivo foi alcançado com a construção dos Capítulos 2 e 3, que trataram do referencial teórico e dos trabalhos relacionados, necessários para se destacar os trabalhos que tinham uma relação mais forte com o que se pretendia realizar.

Pode-se verificar que muitas obras citadas, os autores propuseram a realização de reuniões de *brainstorm* com os usuários para a definição dos *layouts* e interatividade. Nestas reuniões os usuários indicavam o modo de funcionamento, a construção de aplicativos de TV direcionados a um *middleware* específico, principalmente ao MHP e ao GINGA-NCL e a utilização de métodos estatísticos, como o *AdaBoost Haar-Like* para se classificar um gesto, além de uma série de processamentos sobre as imagens para diminuir o peso que estas exercem sobre o consumo de processador e memória.

Para atender ao segundo objetivo específico buscou-se adaptar as técnicas de Engenharia da Usabilidade, TVDi e Visão Computacional para a construção de *layout* e mapeamento de gestos com a participação do usuário, a construção do protótipo de TV e o reconhecimento de gestos. Este objetivo foi alcançado com a construção de uma da arquitetura do sistema, descrita no Capítulo 4.

A arquitetura de sistema foi construída observando as comparações realizadas sob os aspectos funcionais e estratégicos destacados nos trabalhos relacionados. Nestas comparações, levaram-se em consideração os problemas relacionados à interação do usuário e sua TV, as formas de se construir aplicativos de TV, a construção de classificadores de gestos e o processamento de imagens. Ao se considerar os problemas relacionados à interação do usuário e a TVDi, foram observadas as questões relacionadas à consistência de interfaces, através de técnicas que tiveram a participação e o foco no usuário, implementadas através do *framework* PACT.

Ao se considerar os problemas relacionados às formas de se construir aplicativos de TV, foram observadas as incompatibilidades entre os *middlewares* existentes e a falta de padrão nos *layouts* disponíveis. Para a construção de classificadores de gestos foram observadas as técnicas de construção de modelos que guardam as características dos gestos mapeados e os formatos e resoluções das imagens utilizadas para se extrair estas

características. Para a etapa de processamento de imagens foram adotados os processos de calibração radiométrica para a padronização de brilho e luminosidade das imagens e as técnicas de transformação morfológica e suavização para eliminação dos ruídos além das técnicas de *Motion Detection*, *Skin Detection* para eliminação dos *pixels* desnecessários ao reconhecimento de gestos.

O modo de acompanhamento do gesto reconhecido também foi analisado e escolhidas duas formas de serem utilizadas: através do *Haar-Like* para localizar inicialmente o gesto e o *CamShift* para receber o padrão do *Haar-Like* e realizar o restante do *tracking*. Para manter a independência dos protótipos de TV e de Visão Computacional foi adotada uma forma para a troca de informações: através de um arquivo XML. Este arquivo era escrito pelo protótipo de Visão Computacional e tinha os valores das coordenadas e o código de identificação do gesto encontrado escritos pelo protótipo de Visão Computacional.

Para atender ao terceiro objetivo específico experimentou-se em um estudo de caso, através de um protótipo, as operações básicas de ajuste o volume do som e troca de canais de programação na TVDi com uso de recursos de Visão Computacional, realizando o reconhecimento de gestos de forma rápida e eficiente.

Foram seguidas as definições descritas pela arquitetura de sistema, observando cada área abordada e determinando as técnicas e ferramentas que foram adotadas para a construção dos protótipos. O PACT foi trabalhado através da metodologia qualitativa e a técnica de roteiros, que utilizou entrevistas individuais e em grupo para realizar o levantamento dos requisitos que deveriam ser observados para a construção do *layout*, do conjunto de gestos e também de fornecer informações importantes para a construção do protótipo de TV e de Visão Computacional. Para a construção dos protótipos de TVDi foram selecionados dois *middlewares*, o MHP e o GINGA-NCL. Esta necessidade de se produzir dois protótipos de TV em *middlewares* diferentes deveu-se as características e limitações dos *set-top boxes* disponíveis, que não permitiram a conexão uma câmera diretamente a suas portas USB.

A construção dos 5 classificadores de gestos, do tipo *Haar-Like*, definidos pelos usuários foi feita utilizando um *software* próprio de captura de imagens, que ao mesmo tempo padronizava o brilho e o fator de luminosidade para se obter um mesmo padrão nestas imagens. As imagens que eram submetidas ao processo de reconhecimento e busca dos padrões mapeados nos classificadores passavam por um processamento de eliminação de ruídos, de detecção de bordas para reforçar os traçados dos objetos inseridos nas imagens e de

Motion Detection, *Skin Detection* para eliminação dos valores dos *pixels* desnecessários ao reconhecimento de gestos.

O relacionamento entre os protótipos de TV e de Visão Computacional foi realizada com o uso de um arquivo XML que continha os valores das coordenadas e o código de identificação do gesto encontrado, hospedado e disponibilizado pelo servidor de aplicação *Apache TomCat*.

Para atender ao quarto objetivo específico foram realizadas coletas, testes e análises sobre os dados obtidos por este trabalho e os valores apresentados pelos trabalhos relacionados. Os resultados das analises apontaram para um bom nível de aceitação por parte dos usuários e o nível de dificuldade considerado baixo, principalmente quando se observa os tempos obtidos na execução dos comandos pelos usuários do grupo de estudo. O tempo de execução das tarefas também reflete a facilidade de se encontrar as opções no *layout* construído e disponibilizado no protótipo de TV e também do rápido processamento das imagens exibidas e entregues ao processo de reconhecimento de gestos.

O resultado da análise de eficiência dos classificadores construídos apontou que foi obtida uma média de 95% de acertos dos gestos apresentados e uma taxa de atualização média de 26 *frames* por segundo. Esta velocidade trouxe uma sensação de fidelidade de movimentação, sem travamentos ou esperas de mudança de posição do cursor do protótipo. Esta sensação de fidelidade não foi afetada com a integração dos protótipos de TV com a Visão Computacional, mesmo quando se utilizou o *set-top box* que fazia o acesso do arquivo XML através de um roteador *wireless* pois o tempo gasto na leitura do arquivo era inferior a 1 segundo.

O objetivo geral deste trabalho foi investigar e propor a concepção de uma interface para controlar a TV Digital Interativa dos padrões europeu e brasileiro através de um conjunto de gestos que são capturados por uma câmera de baixo custo e convertidos em comandos básicos, servindo como ferramenta auxiliar ou substituta ao controle remoto na ação de interação com a TVDi. Este objetivo foi alcançado ao se verificar as conclusões atingidas em cada objetivo específico.

Uma vez que o objetivo de toda pesquisa científica é fornecer um modelo para a ciência e resultados práticos que possam ser utilizados para resolver problemas reais, este trabalho traz contribuições nas áreas de Engenharia da Usabilidade, TV e Visão Computacional.

A principal contribuição foi a definição de uma forma de se realizar o acesso e controle de uma TVDi em dois *middlewares* diferentes e através de reconhecimento de gestos. Este acesso e controle ocorreu em ambientes com diferenças de iluminação e a distância em que o gesto era apresentado. Outro fator importante a ser destacado foi o uso de uma câmera comum no processo de reconhecimento de gestos, que buscou atender a configuração normalmente encontrada nos computadores, desobrigando o usuário a ter que adquirir uma câmera especifica e cara para esta tarefa.

Na etapa de extração de características foi definida uma combinação de duas técnicas de destaque dos *pixels* que deveriam ser mantidos na imagem. Estas técnicas eliminaram os valores dos *pixels* estáticos através da técnica de *Motion Detection* e os que não se encaixavam nos limiares mínimo e máximo de tonalidade definidas para compor o tom de pele através da técnica de *Skin Detection.* Estas duas técnicas conseguiram abranger de forma bastante versátil uma grande variedade de tons de pele, mesmo sob condições variáveis de iluminação.

7.2 Dificuldades Encontradas

As principais dificuldades encontradas para se construir os protótipos de TVDi foram relacionadas a baixa capacidade do *hardware* dos *set-top boxes* para se permitir o embarque de aplicativos que necessitavam de processamento e memória em volumes encontrados somente nos computadores. Também foi observada a falta de padronização entre os *middlewares* disponíveis de TV, obrigando a escrita de dois protótipos de TV, cada um direcionado a um *middleware* específico. Esta falta de padronização ficou bastante evidente, pois os *middlewares* utilizados não possuíam compatibilidade em seus componentes de operação de *hardware* e também entre seus componentes responsáveis pelo fornecimento das bibliotecas de construção de *layout*. A solução adotada foi construir um módulo no *software* de Visão Computacional que após o processamento das imagens e a identificação das coordenadas e o código de referência do gesto, entregava o resultado aos protótipos de TV, independente de qual *middleware* estivesse sendo executado.

As principais dificuldades encontradas no protótipo de Visão Computacional estão relacionadas ao processo de se reconhecer gestos em ambientes onde a iluminação oscilava muito ou onde os objetos tinham tonalidades dentro dos limiares de definição de tom de pele.

Esta oscilação causou um maior impacto ao sistema de Visão Computacional devido ao modelo de câmera adotado para se realizar o processo de captura das imagens: uma câmera comum, do tipo RGB, de baixo custo e sem nenhum recurso extra. Esta falta de padronização da iluminação teve dois impactos: um no processo de construção do classificador e outro no processamento de imagens. O impacto do processo de construção do classificador foi o de buscar uma configuração de câmera que fosse capaz de fornecer imagens com resoluções que permitissem a percepção dos objetos inseridos pelo algoritmo de *Traincascade* para a geração da árvore de *Haar-Like*.

O impacto no processamento de imagens foi voltado para a busca de diminuição do peso das imagens e dos processos para a eliminação dos valores dos *pixels* desnecessários ao processo de reconhecimento de gestos. As técnicas adotadas eliminaram os valores dos *pixels* estáticos através da técnica de *Motion Detection* e os que não se encaixavam nos limites inferior e superior dos valores determinados para representar o tom de pele através da técnica de *Skin Detection*.

7.3 Sugestões para Trabalhos Futuros

As propostas para se estender este trabalho, decorrentes das oportunidades que surgiram pelos pontos que não puderam ser elucidados são:

- Na área de Engenharia de Usabilidade, deve-se trabalhar na construção participativa com os usuários de um modo de se guardar o perfil e as preferências dos usuários de TV, que seria utilizado para realizar ajustes automáticos de *layout* do painel de botões e também da calibração radiométrica do sistema de Visão Computacional. Ainda referente a usabilidade, o *layout* pode receber um painel contendo números para facilitarem a mudança de canal e uma barra de progressão para se alterar o volume do som.
- Na área de VC, deve-se adicionar outra câmera, com recurso de infravermelho, que trabalhando em conjunto com a câmera RGB formaria um sistema de visão estéreo, permitindo que a segmentação das imagens pudesse ser realizada com maior precisão, uma vez que será possível perceber outras grandezas como profundidade e a distância dos objetos e a câmera e entre os objetos. Ainda na área de VC, para melhorar o classificador de gestos, deve-se adicionar outras técnicas como as redes neurais que irão classificar as imagens a serem utilizadas na construção do classificador *Haar-Like*, eliminando aquelas

imagens que para o programador aparentam serem importantes, mas quando observadas em relação as suas características por um sistema, estas não se enquadram nos padrões esperados para serem utilizados no referido classificador (AHMADI *et al*., 2009). Para a etapa de reconhecimento de gestos, deve-se utilizar técnicas como o SURF (*Speed-up Robust Features*) (H, 2006), que prometem ser mais rápidos que o *Haar-Like*.

Capítulo 8- Referências

ABNT. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 15607: 2008 - Televisão Digital Terrestre – Canal de Interatividade, 2008. 16 f.

AHMADI, M.; ERFANIAN, A..; An On-Line BCI System for Hand Movement Control Using Real-Time Recurrent Probabilistic Neural Network. Proceedings of the 4th International IEEE EMBS Conference on Neural Engineering, Antalya, Turkey, 2009.

AL-AIDAROOS, K. M.; BAKAR, A. A.; OTHMAN, Z. Medical Data Classification with Naive Bayes Approach. Information Technology Journal, 11: 1166-1174. 2012.

AMERINI, I.; BALLOCCA, G.; BECARELLI, R.; BORRI, R.; CALDELLI, R.; FILIPPINI, F. A DVB-MHP web browser to pursue convergence between digital terrestrial television and internet. Multimedia Tools and Applications, 2010.

ANTIKAINEN, A.; KALVIAINEN, M.; MILLER, H. User information for designers: a visual research package, in 'DPPI' ’03 Conference, Pittsburgh, PA, 23-26 ju. Nova York: ACM Press, p. 1-5, 2003.

ARAKI, R.; GOHSHI, S.; IKENAGA, T. Real-time both hands tracking using CamShift with motion mask and probability reduction by motion prediction. In: Signal & Information Processing Association Annual Summit and Conference, (APSIPA ASC) 2012.

BANG-HUA, Y.; GUO-ZHENG, Y.; RONG-GUO, Y.; TING, W. Adaptive subject-based feature extraction in brain–computer interfaces using wavelet packet best basis decomposition. Medical Engineering & Physics - Journal - Elsevier, Vol.29, pp. 48-53, 2007.

BARROS, G. G. A consistência da interface com o usuário para a TV interativa. Dissertação (Mestrado em Engenharia Elétrica) – Departamento de Engenharia de Sistemas Eletrônicos, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, 2006.

BENYON, D. Interação Humano-Computador. 2ª. Edição, São Paulo: Pearson Books, 2011.

BHOWMIK, M. K.; BHATTACHARJEE, D.; NASIPURI, M.; BASU, D. K.; KUNDU, M. A Parallel Framework for Multilayer Perceptron for Human Face. International Journal of Computer Science and Security (IJCSS), 3(6): 491-507, 2009.

BRACKMANN, C. P. Usabilidade em TV Digital. Pelotas, RS.: Universidade Católica de Pelotas (UCPel). Programa de Pós-graduação da Universidade Católica de Pelotas, Mestrado em Ciência da Computação, 2010.

BRADSKI, G; KAEHLER, A. Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, 2008.

BRITO, R. F. de. Modelo de Referência para Desenvolvimento de Artefatos de Apoio ao Acesso dos Surdos ao Audiovisual. Tese (Doutorado). Universidade Federal de Santa Catarina- Departamento de Engenharia e Gestão do Conhecimento. Florianópolis, 2012.

CAROCA, C.; TAVARES, T. A. Um Modelo de Processo de Testes para os Componentes do Núcleo Comum do Middleware OpenGinga. In: IX Workshop de Teses e Dissertações do WEBMEDIA, 2009.

CAVAZZA, M.; Charles, F.; MEAD, S. "Under the Influence: using Natural Language in Interactive Storytelling". International Workshop on Entertainment Computing. Makuhari, Japan, 2002.

COMANICIU, D; MEER, P. MeanShift analysis and applications. The Seventh IEEE International Conference on Computer Vision. Vol. 2. pp. 1197–1203 vol.2, 1999.

COMANICIU, D; MEER, P. MeanShift: A Robust Approach Toward Feature Space Analysis. IEEE Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619. Princeton, New Jersey, USA, May 2002.

CONCI, A.; AZEVEDO, E.; LETA, F. R. Computação gráfica: teoria e prática. São Paulo: Campus/Elsevier, 2008.

CYBIS, W.; BETIOL, A. H.; FAUST, R. Ergonomia e Usabilidade: Conhecimentos, Métodos e Aplicações – Novatec Editora, São Paulo, 2007.

DARDAS, N.; CHEN, Q.; GEORGANAS, N.; PETRIU, E. M. Real-time Vision based Hand Gesture Recognition Using Haar-like features. IEEE International Instrumentation and Measurement Technology Conference, 2007.

DAVOODIANIDALIKI, M.; ABEDINI, A.; SHANKAYI, M. Adaptative Edge Detection Using Adjusted Ant Colony Optimization. Sensors and Models in Photogrammetry and Remote Sensing Conference (SMPR 2013), Volume XL-1/W3, October 2013, Tehran, Iran.

DEVASENA, D.; LAKSHANA, P.; POOVIZHIARASI, A.; VELVIZHI, D. Controlling of Eletronic Equipament Using Gesture Recognition. International Journal of Engineering and Advanced Technology (IJEAT). ISSN: 2249-8958, Volume-3, Issue-2, 2013.

DIX, A. J.; FINLAY, J. E.; ABOWD, G. D.; BEALE, R. Human Computer Interaction – Second Edition – PRENTICE HALL EUROPE 1998, London, New York, Toronto, Sydney, Tokyo, Singapore, Madrid, Mexico City, Munich, 1998, Paris.

ETHERIDGE, D. JAVA: Graphical User Interfaces – An Introduction to Java Programming. Ventus Publishing ApS. ISBN 978-87-496-0, 2009.

FILHO, G. L. de S; LEITE, L. E. C.; BATISTA, C. E. C. F. Ginga-J: The Procedural Middleware for the Brazilian Digital TV System. Journal of the Brazilian Computer Society, no 4, Vol. 12, (ISSN 0104-6500) pp. 47-56, 2007. Porto Alegre, RS, Brasil.

FREEMAN, J.; LESSITER, J. Using Attitude Based Segmentation to Better Understand Viewers Usability Issues with Digital and Interactive TV. European Conference on Interactive Television EuroITV 2003, 2003, pp. 91–97.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. In MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-, pages 148–156. Citeseer, 1996.

FUJITA, T.; PRIVITERA, C. M. Robot Hand Moviment Detection based on Top-Down and Buttom-Up Scanpath Prediction. International Symposium on Robotics and Intelligent Sensors. Elsevier, 2012.

GAWLINSKI, M. Interactive Television Production, 1ª. Edition, Focal Press, p.2, 2003.

H. B; T. T.; L. V. G. SURF: Speeded up robust features. In ECCV, 2006.

HAAR, A. Zur Theorie der orthogonalen Funktionensysteme. (German) Mathematische Annalen 69, no. 3, 331—371, 1910.

HAN, X..; SAMEI, Y.; DOMER, R. System-level modeling and refinement of a canny edge detector. Technical Report CECS-TR-12-13, Center for Embedded Computer Systems, University of California, Irvine, Nov. 2012.

IERUSALIMSCHY, R.; FIGUEIREDO, L. H. de; CELES, W. The evolution of Lua. History of Programming Languages HOPL III ACM Conference, San Diego, CA, USA, 2-1-2-26, 2007.

ITU-T Recommendation H.761. Nested Context Language (NCL) and Ginga-NCL for IPTV Services. Geneva, 2009.

JAGADESH, B.N.; SATYANARAYANA, K. S. R. A Robust Skin Colour Segmentation Using Bivariate Pearson Type IIαα (Bivariate Beta) Mixture Model. International Journal of Image, Graphics and Signal Processing (IJIGSP) ISSN: 2074-9074(Print), ISSN: 2074-9082 (Online). IJIGSP Vol.4, No.11, 2012.

JAWAS, N.; SUCIATI, N. Image Inpainting using Erosion and Dilation Operation. Department or Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nobember (ITS) Surabaya, Indonesia. International Journal of Advanced Science and Technology, Vol. 51, Fabruary, 2013.

JONES, M; VIOLA, P. "Rapid Object Detection using a Boosted Cascade of Simple Features", IEEE Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA, 2001.

JUCÁ, P.; LUCENA, U de. Experiências no desenvolvimento de Aplicações para Televisão Digital Interativa. In: III Fórum de oportunidades em TV Digital Interativa, Poços de Caldas, 2005.

KULESZA, R. Ginga-J: Implementação de Referência do Ambiente Imperativo do Middleware Ginga. Belo Horizonte: Webmedia 2010 – Simpósio Brasileiro de sistemas Multimidia Web, 2010.

LEMOS, A. L. M. Anjos Interativos e retribalização do mundo: sobre interatividade e interfaces digitais, 1997.

LIANG, S-F M. Control with Hand Gestures in Home Environment: A Review. In: Institute of Industrial Engineers Asian Conference, pp. 837-843, 2013.

LIENHART, R., KURANOV, A., PISAREVSKY, V., "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection," MRL Technical Report, 2002.

LIMA, G. F.; SOARES, L. F. G.; MORENO, M. F.; AZEVEDO, R. G. A. Reducing the Complexity of NCL Player Implementations. Anais do XIX Simpósio Brasileiro de Sistemas

Multimídia e Hipermídia. Salvador, Bahia. November, 2013; pp. 297-304. DOI:10.1145/2526188.2526217.

LUCENA, V. F.; QUEIROZ NETO, J.P.; BENCHIMOL, I.B.; MENDONÇA, A. P.; SILVA, V. R.; FILHO, M. F. "Teaching Software Engineering for Embedded Systems: an Experience Report from the Manaus Research and Development Pole". In: 37th ASEE/IEEE Frontiers in Education Conference, Milwaukee, USA, 2007.

MELO, A. M.; PICCOLO, L. S. G.; ÁVILA, Ismael; TAMBASCIA, C. de A. Usabilidade, Acessibilidade e Inteligibilidade aplicadas em interfaces para analfabetos, idosos e pessoas com deficiência – Resultados do Workshop. IHC 2008 - VIII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais, CPqD – Campinas, Brasil, 2008.

MIGLIORE, D.; MATTEUCCI, M.; NACCARI, M. A revaluation of frame difference in fast and robust motion detection. In: 4th ACM International Workshop on Video Surveillance and Sensor Networks (VSSN), pp. 215–218. Santa Barbara, California, 2006.

MIRANDA, L. da C.; HORNUNG, H. H.; BARANAUSKAS, M. C. C. Prospecting a Gesture Based Interaction Model for iDTV. In: IADIS International Conference on Interfaces and Human Computer Interaction (IHCI) / IADIS Multi Conference on Computer Science and Information Systems (MCCSIS), 2009, Algarve, Portugal. Proceedings of the IADIS International Conference on Interfaces and Human Computer Interaction. Lisbon, Portugal: IADIS Press, 2009. p. 19-26.

MURTHY, N. K. N.; KUMARASWAMY, Y. S. A Novel Method for Efficient Text Extraction from Real Time Images with Diversified Background using Haar Discrete Wavelet Transform and K-Means Clustering. International Journal of Computer Science Issues (IJCSI), Vol. 8, Issue 5, No 3, September 2011. ISSN (Online): 1694-0814.

NADGERI, S. M; SAWARKAR, S. D; GAWANDE A. D. "Hand gesture recognition using CAMSHIFT algorithm," International Conference on Emerging Trends in Engineering and Technology (ICETET), pp. 37-41, 2010.

NAPOLEON, D.; MAGESHWARI, V.; REVATHI, P. A Resourceful Filtering Technique for Texture Segmentation and Enhancement in Remote Sensing Images Using Morphological Operations. International Journal or Research in Advent Technology, Vol. 1, Issue 5, December 2013. E-ISSN 2321-9637.

NETO, J. J. Solving complex problems efficiently with adaptative automata. In: Conference on the Implementation and Application of Automata - CIAA 2000, CIAA, London, Ontario, Canada, 2000.

RANGEL, V.; EDWARDS, R. Performance Analysis and Optimization of the Digital Video Broadcasting/ Digital Audio Visual Concil Cable Modem Protocol for the Delivery of Isochronous Streams. IEEE Global Communications Conference, Exhibition & Industry Forum (GLOBECOM), Houston, Texas, USA, 2011.

RAUTARAY, S. S.; AGRAWAL, A. Real time hand gesture recognition system for dynamic applications. International Journal of UbiComp (IJU), Vol. 3, No. 1, January, 2012.

ROSSON, M. B., CARROLL, J. M. Usability Engineering: Scenario-Based Development of Human-Computer Interaction. England: Morgan Kaufmann Publishers. 2002.

SALUJA, A.; MOKAYA, F.; PHIELIPP, M.; KVETON, B. Automatic Identity Inference for Smart TVs. Lifelong Learning: papers from the 2011 AAAI Workshop (WS-11-15), 2011.

SELLEN, A.; ROGERS, Y.; HARPER, R.; RODDEN, T. Reflecting human values in the digital age. Communications of the ACM - Volume 52, Issue 3. Being Human in the Digital Age - Pages 58-66, 2009.

SILVA, F. P. R. XTATION: Um Ambiente para Execução e Teste de Aplicações Interativas para o Middleware Ginga. Dissertação (Mestrado). Universidade Federal da Paraíba, 2009.

SILVA, R.; AURES, K.; SANTOS, T.; ABDALLA, K.; VERAS, R. Segmentação, Classificação e Detecção de Motocicletas sem Capacete. Universidade Federal do Piauí. Simpósio Brasileiro de Automação Inteligente – SBAI, 2013.

SNYDER, C. “Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces”, Snyder Consulting. Disponível em: <http://www.Snyderconsulting.net/article_paperprototyping.htm>, Acesso em: 09 nov. 2013.

STEVENTON, A.; WRIGHT, S. Intelligent Spaces: The Application of Pervasive ICT, Springer-Verlag, 2006.

TEIRIKANGAS, J. HAVi: Home Audio Video Interoperability. Technical report, Helsinki University of Technology, 2001.

TSAMOURA, E.; Mezaris, V.; Kompatsiaris, I. Gradual transition detection using color coherence and other criteria in a video shot metasegmentation framework. 15th IEEE International Conference on, Image Processing (ICIP), pages 45 48, 2008.

VAIRALKAR, M. K; NIMBHORKAR, S. U. Edge Detection of Images Using Sobel Operator. International Journal of Emerging Technology and Advanced Engineering – IJETAE. ISSN: 2250-2459, Volume 2, Issue 1, January 2012. Disponível em: www.ijetae.com. Acesso em 30 set. 2013.

VATAVU, R. D. User-defined gestures for free-hand TV control. In: Paper presented at the 10th European conference on interactive TV and video (EuroITV'12). Berlin, Germany, 4–6 July 2012, pp 45–48, 2012.

WANG, X.; HAN, T. X.; SHUICHENG, Y. Um Detector Human HOG-LBP com Handling oclusão parcial"., ICCV 2009.

WANGENHEIM, A. V.; COMUNELLO, E. Visão Computacional: Seminário Introdução à Visão Computacional. The Cyclops Project. PPGCC-INE-UFSC. 2005. Disponível em http://www.inf.ufsc.br/~visao/. Acesso em 02 de mar. 2011.

WILSON, A. Sensor and recognition-based input for interaction. In: SEARS, A., JACKO, J. A. (Org.). The Human-Computer Interaction Handbook. 2. Ed. Mahwah, NJ: Lawrence Erlbaum Associates, 2007.

WILSON, P. I.; FERNANDEZ, J. Facial feature detection using Haar classifiers. Texas A&M University – Corpus. 2009.

WU, H.; DENG, D.; CHEN, X.; LI, G.; WANG, D. Localization and Recognition of Digit-Writing Hand Gestures for Smart TV Systems. Journal of Information & Computational Science (JOICS), Vol. 11, No. 3, February, 2014.

YI, L. Hand Gesture Recognition Using Kinect. Compt. Eng & Eng. Sci. University of Lousville, Lousville, KY, USA. Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference.

YOUNG, I. T.; GERBRANDS, J. J.; VLIET, L. J. V. Fundamentals of Image Processing. Paperback, ISBN 90-75691-01-7, 1995.

ZUFFO, M. K. TV Digital Aberta no Brasil – Políticas Estruturais para um Modelo Nacional - Departamento de Engenharia de Sistemas Eletrônicos. Escola Politécnica - Universidade de São Paulo, 2006.

Apêndice A- Publicações Diretamente Relacionadas à Proposta

SIMOES, W.C.S.S.; BARBOZA, R. da S.; LUCENA JR., Vicente; LINS, R.D. A Fast and Accurate Algorithm for Detecting and Tracking Moving Hand Gestures. VipImage. IV ECCOMAS Thematic Conference on Computacional Vision and Medical Image Processing. Madeira Island Funchal, Portugal. 2013.

SIMOES, W.C.S.S.; BARBOZA, R. da S.; LUCENA JR., Vicente; LINS, R.D. Use of hand gestures as interface for interaction between multi-users and the IDTV. XI EuroITV – European Interactive TV Conference. Como - Italy, 2013.

SIMOES, W.C.S.S.; LUCENA JR., Vicente; LEITE, Jandecy C.; SILVA, C.A de S. Visíon por computador para manos a base de reconocimiento de gestos para la interacción com los sistemas operativos de escritório Windows y Linux. XXXIII UPADI - Convención Panamericana de Ingenierías. La Habana - Cuba, 2012.

SIMOES, W.S.S.S.; LUCENA JR., Vicente. Remoção do Fundo da Cena para Detecção da Silhueta da Mão Humana e Detecção de Movimentos. I SIGES - Simpósio de Informática e Geotecnologia de Santarém-PA, Brasil, 2011.

Apêndice B- Outras Publicações

PICANÇO, W. de S; SIMOES, W.C.S.S.; ALBUQUERQUE, W.C. de; BARBOZA, R. da S.; ESTEVES, A.F. Framework of animation limited: structure to handle still images via idtv. XI EuroITV – European Interactive TV Conference. Como - Italy, 2013.

SIMOES, W.C.S.S.; LUCENA JR., Vicente; COLLINS, E.; ALBUQUERQUE, W.; PADILLA, R.; VALENTE, R. Avaliação de ambientes de desenvolvimento para automação do problema do cubo mágico para o robô lego mindstorms nxt. V CONNEPI – Congresso Norte-Nordeste de Pesquisa e Inovação. Maceió – Alagoas – Brasil. ISBN: 978-85-64320-00-0, 2010.

Apêndice C- Questionário sobre o grau de Satisfação dos Usuários

Questionário – Avaliação do grau de satisfação dos usuários					
Preencha os campos abaixo de acordo com a sua opinião: Para cada linha da tabela abaixo, escolha apenas uma coluna, marcando-a com um X. 5 – Muito bom; 4 – Bom; 3 – Indiferente; 2 – Ruim; 1 – Muito Ruim.					
	5	4	3	2	1
Como você avalia a proposta de utilizar os recursos de TV através de gestos?					
Como você avalia a movimentação do cursor na tela da TV?					
Como você avalia a facilidade em posicionar a mao sobre os botões do protótipo de TV?					
Como você avalia a dificuldade de se utilizar o sistema?					
Como você avalia a sua sensação após utilizar um protótipo de reconhecimento de gestos?					

Apêndice D- Questionário de Entrevista Individual para Definição de Perfil de Usuário

Questionário – Identificação do Usuário	
Preencha os campos abaixo com os seus dados:	
Idade:	
Sexo:	
Possui experiência com computadores?	
Possui experiência com jogos em primeira pessoa?	
Possui experiência em jogos em controles gestuais?	

Apêndice E- Questionário de Entrevista para Definição de *Layout* de TV

Questionário – Definição de *Layout* de TV	
Preencha os campos abaixo com os seus dados com um texto ou desenhos se preferir	
Como você desenharia uma tela para dispor os botões de controle de volume e canais de programação de uma TV para ser localizada rapidamente?	
Como você desenharia os botões para ajustar o volume de uma TV para serem localizados rapidamente?	
Como você desenharia os botões para trocar canais de programação de uma TV para serem localizados rapidamente?	
Qual cor você escolheria para ser utilizada como plano de fundo da tela de botões para ser localizada rapidamente?	
Qual cor você escolheria para ser utilizada nos botões de volume para serem localizados rapidamente?	
Qual cor você escolheria para ser utilizada nos botões de canais de programação para serem localizados rapidamente?	

Apêndice F- Questionário de Entrevista para Definição do Conjunto de Gestos para comandar a TV

Questionário – Definição do conjunto de gestos para comandar a TV	
Preencha os campos abaixo com os seus dados com um texto ou desenhos se preferir	
Qual gesto você usaria para acionar a tela de botões de volume do som e canal da TV?	
Qual gesto você usaria para fechar a tela de botões de volume do som e canais de programação da TV?	
Qual gesto você usaria para aumentar o volume do som?	
Qual gesto você usaria para diminuir o volume do som da TV?	
Qual gesto você usaria para avançar os canais de programação da TV?	
Qual gesto você usaria para voltar os canais de programação da TV?	