

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Sistema de Reconhecimento de Ações - Uso para Monitoramento de Pacientes Hêmiparéticos Vítimas de Acidente Vascular Cerebral

Autor: Itallo Rossi Barcelos Lucas
Orientadora: Dra. Carla Silva Rocha Aguiar

Brasília, DF
2013



Itallo Rossi Barcelos Lucas

**Sistema de Reconhecimento de Ações - Uso para
Monitoramento de Pacientes Hemiplégicos Vítimas de
Acidente Vascular Cerebral**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dra. Carla Silva Rocha Aguiar

Brasília, DF

2013

Itallo Rossi Barcelos Lucas

Sistema de Reconhecimento de Ações - Uso para Monitoramento de Pacientes Hemiplégicos Vítimas de Acidente Vascular Cerebral/ Itallo Rossi Barcelos Lucas.
– Brasília, DF, 2013-

69 p. : il. (algumas color.) ; 30 cm.

Orientador: Dra. Carla Silva Rocha Aguiar

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2013.

1. Reconhecimento de Ações. 2. Hemiplegia. I. Dra. Carla Silva Rocha Aguiar. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Sistema de Reconhecimento de Ações - Uso para Monitoramento de Pacientes Hemiplégicos Vítimas de Acidente Vascular Cerebral

CDU 02:141:005.6

Itallo Rossi Barcelos Lucas

Sistema de Reconhecimento de Ações - Uso para Monitoramento de Pacientes Hemiplégicos Vítimas de Acidente Vascular Cerebral

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 12 de dezembro de 2013:

Dra. Carla Silva Rocha Aguiar
Orientadora

Dr. Edson Alves da Costa Júnior
Convidado

Dr. Tiago Franklin Rodrigues Lucena
Convidado

Brasília, DF
2013

*Este trabalho é dedicado ao meu pai, um milagre,
e à minha mãe, a mulher mais forte que conheço.*

Agradecimentos

Agradeço primeiramente a Deus, por tudo que tenho e sou.

Agradeço aos meus pais, pelo carinho, pelo apoio durante a faculdade, pois sem eles nada disso seria possível.

Agradeço a minha namorada, Larissa Lorena L. e V. dos Santos, pela paciência, pela força, pelo incentivo e por estar ao meu lado sempre durante o desenvolvimento desse trabalho, mesmo que eu não pudesse lhe dar atenção.

Agradeço a minha orientadora, Professora Carla Silva Rocha Aguiar, Dra., pelas sugestões, por toda ajuda e por ser compreensiva com as minhas dificuldades.

*“Troubles will come and they will pass,
Go find a woman and you’ll find love,
And don’t forget son,
There is someone up above.
(Lynyrd Skynyrd)*

Resumo

O Sistema de Reconhecimento de Ações desenvolvido nesse trabalho passou por três fases importantes antes da sua construção. A primeira fase foi uma investigação bibliográfica sobre os sistemas de reconhecimento de ações já existentes procurando saber quais problemas eles solucionam, quais suas limitações e quais técnicas o mesmos utilizam. A segunda fase constitui-se da definição de como o sistema seria construído, com base nas técnicas estudadas anteriormente, assim como a escolha dos usuários alvo do sistema, que no caso são os hemiplégicos. A terceira fase foi a definição de qual é o problema a ser resolvido pelo sistema, que no caso foram as dificuldades na marcha apresentadas pelos usuários hemiplégicos. O desenvolvimento aplicou técnicas de Engenharia de Software, através do levantamento de requisitos, definição de uma arquitetura e execução de testes, para validação do algoritmo e do sistema. A metodologia de desenvolvimento, a técnica usada para a codificação, assim como as ferramentas usadas no desenvolvimentos estão descritas neste trabalho. Por fim, são listadas as limitações do sistemas, as oportunidades de melhoria, os resultados obtidos com o desenvolvimento e sugestões de trabalhos futuros.

Palavras-chaves: Reconhecimento de Ações. Hemiplegia. Monitoramento.

Abstract

The Action Recognition System developed in this work passed through three steps before being built. The first step was the bibliographic investigation about existing action recognition systems looking for the problems that they solve, their limitations and which techniques they used. The second step was the definition of how the system was going to be built, based on techniques studied before, as well as the choice of whom would be the users of the system, in this case hemiplegics. The third step was the definition of the problem to be solved by the system, that in this case were the difficulties presented by hemiplegic users. The development applied Software Engineer techniques, such as software requirements, software architecture and tests, the tests were used to validate the algorithm and the system itself. The development methodology used, the code technique, as well as the tools used during the whole development are described in this work. Finally, were listed the limitations of the system, the improvement opportunities, the results obtained and suggestions of future workers.

Key-words: Action Recognition. Hemiplegia. Monitoring.

Lista de ilustrações

Figura 1 – Debilidade nos braços, exemplo de sequela do AVC.	24
Figura 2 – Movimentos do Pé.	27
Figura 3 – Detecção de pessoa usando posicionamento elíptico.	29
Figura 4 – Esquema de um protótipo de Rede de Sensores.	30
Figura 5 – Exemplo de um esquema de Rede de Câmeras.	31
Figura 6 – Estrutura do algoritmo usado para detecção de objetos.	33
Figura 7 – Arquitetura usada para definir a postura do usuário.	35
Figura 8 – Remoção do fundo, definição da silhueta e segmentação do corpo. . . .	36
Figura 9 – Processo de funcionamento da técnica <i>mean shift</i>	37
Figura 10 –Gráfico de Precisão por Relevância.	39
Figura 11 –Exemplo de imagem da base KTH, mostrando a diferença na escala. . .	39
Figura 12 –Precisão por número de <i>templates</i>	40
Figura 13 – <i>Framework</i> que combina RGB e profundidade.	42
Figura 14 –(a) Vídeo RGB (b) MHI para frente (c) MHI para trás.	42
Figura 15 –Juntas do esqueleto reconhecidas pelo <i>Kinect</i>	45
Figura 16 –Aquisição de <i>Templates</i>	46
Figura 17 –Fórmula para cálculo da distância das regiões.	47
Figura 18 –Resultado da execução do algoritmo.	47
Figura 19 –Arquitetura do Módulo de Criação de <i>Templates</i>	51
Figura 20 –Arquitetura do Módulo de Monitoramento e Reconhecimento de Ações	51
Figura 21 – <i>Templates</i> sendo comparados a cena sem ação nenhuma.	56
Figura 22 – <i>Templates</i> sendo comparados a cena com movimento do braço direito. .	56
Figura 23 –Marcha Hemiplégica.	60
Figura 24 –Marcha Normal.	60
Figura 25 –Mover Braço Direito.	60
Figura 26 –Mover Braço Esquerdo.	61
Figura 27 –Abaixar.	61
Figura 28 –Marcha Hemiplégica e Mover Braço Esquerdo.	61

Lista de tabelas

Tabela 1 – Comparativo de precisão com outros estudos.	40
Tabela 2 – <i>Templates</i> e juntas de interesse.	57
Tabela 3 – <i>Templates</i> e erro máximo.	58

Lista de abreviaturas e siglas

AVC	Acidente Vascular Cerebral
AVE	Acidente Vascular Encefálico
RGB	<i>Red Green Blue</i> - Vermelho Verde Azul
RGBD	<i>Red Green Blue Depth</i> - Vermelho Verde Azul Profundidade
HOG	<i>Histogram of Gradient</i> - Histograma de Gradiente
HOF	<i>Histograma of Flow</i> - Histograma de Fluxo
STIPs	<i>Spatio-Temporal Interest Points</i> - Pontos de Interesse no Espaço-Tempo
LDP	<i>Local Depth Pattern</i> - Padrão de Profundidade Local
MHIs	<i>Motion History Images</i> - Histórico de Movimento das Imagens
HCS	<i>Home Care Systems</i> - Sistemas de Cuidado Domiciliar
SCD	Sistemas de Cuidado Domiciliar

Sumário

1	Introdução	23
2	Fundamentação Teórica	25
2.1	Marcha Normal e Marcha de Paciente Hemiparéticos	25
2.2	Ambientes Inteligentes	27
2.2.1	Ambientes Inteligentes e <i>Assisted Living</i>	28
2.2.2	Ambiente Inteligente usando Redes de Sensores	28
2.2.3	Ambiente Inteligente para Detecção de Objetos	31
2.3	Monitoramento de Acidentes para <i>Assisted Living</i>	32
2.4	Técnicas para Reconhecimento de Ações	35
2.4.1	<i>Mean shift</i> temporal para Reconhecimento de Ações	35
2.4.2	Reconhecimento de Ações via Processamento de Imagens	38
2.4.3	Reconhecimento de Ações usando Sensores de Profundidade	41
3	Problema	43
3.1	Proposta da Solução	44
3.1.1	Módulo de Criação de <i>Templates</i>	44
3.1.2	Módulo de Monitoramento e Reconhecimento de Ações	46
4	Desenvolvimento	49
4.1	Requisitos	49
4.1.1	Visão Geral	49
4.1.2	Requisitos Funcionais	49
4.1.2.1	Módulo de Criação de <i>Templates</i>	49
4.1.2.2	Módulo de Monitoramento e Reconhecimento de Ações	50
4.1.3	Requisitos Não Funcionais	50
4.1.3.1	Módulo de Criação de <i>Templates</i>	50
4.1.3.2	Módulo de Monitoramento e Reconhecimento de Ações	50
4.2	Arquitetura	51
4.3	Solução Desenvolvida	52
4.3.1	Processo de Desenvolvimento	53
4.3.2	Caso de uso: Pacientes Hemiplégicos Vítimas de AVC	56
4.3.3	Limitações da Solução	58
4.3.4	Requisitos de Sistema	59
5	Conclusão	63

Referências 65

1 Introdução

Ambientes Inteligentes são definidos por (BURROWS; ABADI; NEEDHAM, 1990) como sendo um tipo de sistema que proporciona serviço de forma sensível e responsiva e estão integrados aos ambientes de forma não obstrutiva.

Ambientes Inteligentes vêm sendo bastante mencionados como uma forma de resolver o problema do *Assisted Living*. Sistemas de assistência à vida focam no suporte às pessoas com necessidades especiais em suas próprias casas (KLEINBERGER et al., 2007). Essas casas são chamadas de *Home Care Systems* (HCS), ou, em português, Sistemas de Cuidado Domiciliar (SCD).

O aumento acentuado da longevidade trouxe uma mudança no perfil demográfico não só nos países europeus, mas também no Brasil (PERLINI, 2000). Desde a década de 60 pode ser observado no Brasil um aumento de doenças crônico-degenerativas, principalmente as cardiovasculares, superando enfermidades infecciosas e parasitárias como causa de morte (NETTO, 1999).

É ressaltado por (PERLINI, 2000) que, dentre as doenças crônico-degenerativas, a doença cerebrovascular é a terceira no número de mortes no mundo, vindo depois somente das cardiopatias em geral e o câncer. Das doenças cerebrovasculares, o Acidente Vascular Cerebral (AVC) é a doença com maior incidência e possui maior número de óbitos, gerando incapacitação nos pacientes afetados (NITRINI; BACHESCHI, 2003).

Após seis meses da ocorrência do AVC, entre 40 e 50% dos pacientes vêm a falecer (PERLINI, 2000). O AVC é a doença que mais causa incapacitação funcional no mundo ocidental, devido as incapacidades neurológicas residuais (ANDRÉ, 2006).

De acordo com (PERLINI, 2000), apesar dos programas de reabilitação ajudarem na diminuição considerável dos danos causados pela doença, ainda há a permanência de sequelas, um exemplo pode ser visto na Figura 1. O AVC compromete a capacidade da vítima em administrar sua vida pessoal, pois pode trazer como sequelas: limitações motoras, sensitivas, sensoriais, de compreensão e expressão dos pensamentos, limitando suas interação com o ambiente e as pessoas a sua volta.

Segundo (SILVA, 1995), devido à dificuldade do paciente de administrar sua vida pessoal, os papéis familiares, muitas vezes, necessitam ser redefinidos para que o paciente possa ter os cuidados necessários. Algumas vezes, a redefinição do papel se torna tão brusca que é preciso que o paciente e um membro da família passem a viver como uma pessoa só, pois é imprescindível que haja um monitoramento constante do paciente (PERLINI,

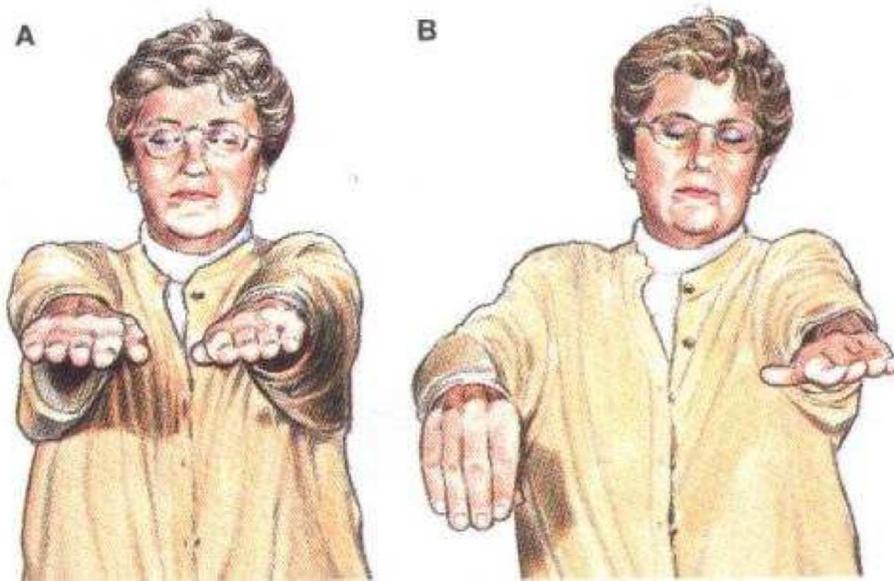


Figura 1 – Debilidade nos braços, exemplo de seqüela do AVC. Fonte:(GRE, 2002)

2000).

É lembrado por (PERLINI, 2000) que a necessidade constante do monitoramento de vítimas do AVC se dá pelo fato que, mesmo após algum tempo de reabilitação, tarefas como andar, escovar os dentes e até mesmo se vestir se tornam muito difíceis por consequência das debilidades motoras causadas pela doença.

Tendo em vista as dificuldades apresentadas pelos pacientes vítimas de AVC, o objetivo principal desse trabalho é a melhoria da qualidade de vida do paciente e de seus familiares, a partir da criação de um sistema de monitoramento e reconhecimento de ações, que será aplicado a um paciente hemiplégico vítima de AVC.

A primeira etapa do trabalho foi constituída de um estudo da bibliografia existente sobre *Assisted Living*, do reconhecimento de ações e de sistemas de monitoramento e detecção de acidentes. A segunda etapa do trabalho foi a definição do problema a ser resolvido e a proposta de uma solução. A terceira etapa foi o desenvolvimento da solução proposta e a execução dos testes.

2 Fundamentação Teórica

O capítulo a seguir tem com objetivo apresentar os principais conceitos teóricos abordados no presente trabalho sobre as sequelas do AVC, ambientes inteligentes e técnicas para reconhecimento de ações usando processamento de imagens com sensores de imagem RGB e sensores de profundidade.

2.1 Marcha Normal e Marcha de Pacientes Hemiparéticos por Acidente Vascular Encefálico

A marcha normal segundo (OTTOBONI; FONTES; FUKUJIMA, 2002) é a performance repetida dos membros inferiores que avançam o corpo ao longo de uma linha. Ela deve manter uma postura estável que sustente o peso corporal.

Para ser realmente funcional uma marcha deve (DAVIES, 1996):

- Ser segura, deixando o paciente livre de receios;
- Relativamente sem esforços, não pode ser uma tarefa que possua um alto nível de consumo energético;
- Esteticamente agradável onde o indivíduo possa caminhar sem ser notado;
- Sem o uso de aparatos de suporte, deixando os membros superiores livres para outras atividades, e
- Executado de forma automática, possibilitando que o indivíduo possa se concentrar em outras atividades simultaneamente.

De acordo com (OTTOBONI; FONTES; FUKUJIMA, 2002) a marcha equivale ao comprimento da passada que, por sua vez, equivale a um passo direito e um passo esquerdo. O comprimento da passada é calculado entre a distância entre o toque do calcanhar do pé de referência até o segundo toque do mesmo calcanhar.

A velocidade da marcha pode ser calculada a partir de aproximadamente três passos, onde é atingido um estado rítmico (OTTOBONI; FONTES; FUKUJIMA, 2002). A maior parte dos adultos possui um ritmo de caminhada de 90 a 120 passos/minuto.

A marcha normal é representada por uma sucessão de transferências de peso e é subdividida em basicamente duas fases (GK, 1987):

1. Suporte, também conhecida como fase de apoio (*stance*) que ocupa por volta de 60% (sessenta por cento) do ciclo normal, e
2. A progressão, denominada balanço (*balance*), que ocupa por volta dos 40% (quarenta por cento), restantes do ciclo normal.

É definido por (GK, 1987) que a fase de apoio fornece estabilidade e propulsão, e por sua vez a fase de balanço fornece avanço e afastamento do membro.

Em (OTTOBONI; FONTES; FUKUJIMA, 2002) é lembrado que a disfunção motora é uma das sequelas mais persistentes em pacientes vítimas de AVC, que é caracterizada pela paralisia total do lado (hemiplegia ou hemiparesia) oposto ao hemisfério cerebral atingido pelo AVC.

Os desvios na marcha de pessoas com hemiplegia vem sendo definidas pelas suas anormalidades motoras, causadas pela perda dos mecanismos de controle neuromotor (MOORE, 1994).

A marcha após o AVC possui várias alterações quanto a velocidade, cadência, simetria, tempo, comprimento dos passos, desajustes na postura, equilíbrio, reação de proteção, dificuldade na iniciação do passo e diferenciação na duração; também, a falta de balanceio nos membros superiores ou até mesmo a indisponibilidade de um membro durante a marcha (OTTOBONI; FONTES; FUKUJIMA, 2002).

De acordo com (OTTOBONI; FONTES; FUKUJIMA, 2002) existe uma grande diferença na marcha dos diferentes paciente hemiparéticos, apesar da grande similaridade na aparência. No entanto foram listadas algumas características gerais da marcha do hemiparético, que são as seguintes:

- Aumento da flexão do quadril durante toda a fase de apoio;
- Diminuição do deslocamento lateral pélvico;
- Hiperextensão do joelho durante toda a fase de apoio;
- O contato inicial que antes era feito pelo calcanhar agora é feito pelo antepé, causando uma diminuição da plantiflexão e conseqüente redução da força propulsiva, e
- Diminuição na dorsiflexão.

A Figura 2 mostra como é realizado o movimento da plantiflexão e da dorsiflexão.

A partir dessa informações, (OTTOBONI; FONTES; FUKUJIMA, 2002) concluíram que uma pessoa hemiplégica apresenta anormalidades biomecânicas significantes que

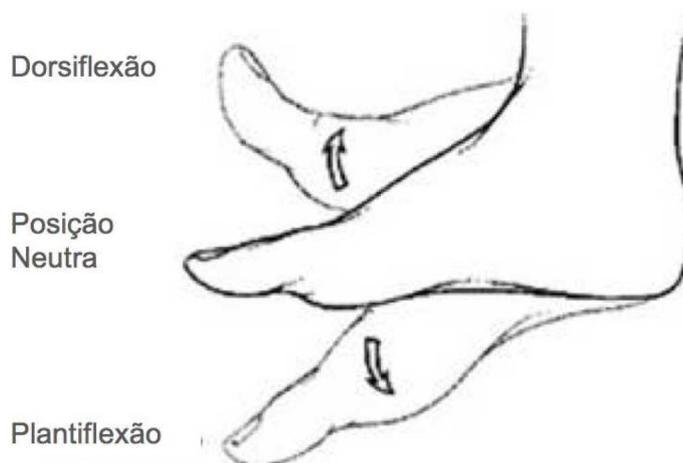


Figura 2 – Movimentos do Pé. Fonte:(UFSC, 2004)

alteram sua marcha em relação a marcha normal. Essas alterações podem ser percebidas visualmente tornando possível diferenciar um paciente hemiplégico de uma pessoa normal.

A partir das anormalidades biomecânicas que o paciente vítima do AVC apresenta, sistemas de monitoramento e ambientes inteligentes podem ser usados para melhorar a qualidade de vida desse paciente e de seus cuidadores, pois tais ambientes podem ser usados para detecção de acidentes e acompanhamento remoto da execução de atividades.

2.2 Ambientes Inteligentes

Ambientes Inteligentes são definidos como ambientes eletrônicos inteligentes que são sensíveis e responsivos à presença de uma pessoa. Ambientes Inteligentes fazem o uso de dispositivos computacionais embarcados para integrar de forma inteligente, utilizando modelos de tomadas de decisão automatizadas (e.g, a partir da leitura de informações de sensores decidir se uma pessoa está caindo), o ambiente onde pessoa se encontra, tornando a tecnologia somente um plano de fundo e deixando o usuário em evidência (AARTS; WICHERT, 2009) .

A visão de Ambientes Inteligentes tem sido cada vez mais amadurecida, e tornando-se influente no desenvolvimento de novos conceitos para processamento de informação, combinando diferentes disciplinas, como a ciência da computação, desenho de interfaces e ciências cognitivas (AARTS; WICHERT, 2009).

O paradigma de ambientes inteligentes é usado como base para novos modelos de inovação tecnológica, cuja integração em larga escala de componentes eletrônicos no ambiente possibilita o usuário de interagir com o espaço a sua volta de forma natural (AARTS; WICHERT, 2009).

O trabalho realizado por (REMAGNINO; FORESTI; ELLIS, 2005) também traz

ideias e conceitos sobre ambientes inteligentes, mostrando uma perspectiva de pesquisadores da área de Inteligência Artificial e Visão Computacional sobre o assunto.

2.2.1 Ambientes Inteligentes e *Assisted Living*

Ambientes Inteligentes vêm sendo bastante mencionados como uma forma de resolver o problema do *Assisted Living*. Sistemas de assistência à vida focam no suporte às pessoas com necessidades especiais em suas próprias casas (KLEINBERGER et al., 2007). Essas casas são chamadas de *Home Care Systems* (HCS), ou, em português, Sistemas de Cuidado Domiciliar (SCD).

O objetivo do HCS é tornar possível a pessoa assistida viver de forma monitorada o maior tempo possível em seu ambiente usual, que comumente é sua casa. HCS é um tipo de Ambiente Inteligente que pode ser estruturado de acordo com as necessidades da pessoa assistida, mas, de acordo com (KLEINBERGER et al., 2007), essas necessidades normalmente se encontram nos seguintes domínios:

- Assistência para emergências;
- Melhoria da autonomia, e
- Conforto.

Cada domínio possui objetivos específicos, por exemplo, para conforto, os objetivos são: segurança, automação do ambiente e encontrar objetos (KLEINBERGER et al., 2007).

Assim, (KLEINBERGER et al., 2007) e também (COELHO, 2012) mostram que os Ambientes Inteligentes têm muito a contribuir para o *Assisted Living*, pois os HCS são um tipo de sistemas que andam lado a lado com Ambientes Inteligentes. Dois bons exemplo do uso da tecnologia para o *Assisted Living* é o Sistema Especialista para diagnóstico de crises epiléticas proposto por (BRASIL; AZEVEDO; BARRETO, 2001) e a Palmilha Sensorizada para Pés Diabéticos proposto por (RODRIGUES; ROCHA, 2011).

A partir da definição dos conceitos sobre Ambientes Inteligentes, muitos trabalhos foram feitos na área, desde Ambientes Inteligentes com foco na detecção de quedas até Ambientes Inteligentes que têm por objetivo detectar objetos no ambiente. Mais sobre o funcionamento desses ambientes será descrito nas seções 1.2.2 e 1.2.3.

2.2.2 Ambiente Inteligente usando Redes de Sensores para Detecção de Quedas

O rápido avanço nas tecnologias de sensores de imagem e processadores embarcados possibilitam a inclusão de módulos de visão em aplicações de ambientes inteligentes que usam Redes de Sensores (BRAMBERGER et al., 2006). Esses sensores geram uma quantidade rica de informações tanto para o ser humano, que observa esses dados, quanto para o computador, que realizará a interpretação dos dados. Essa quantidade rica de informações que a rede de sensores gera, deixa de ser somente para monitoramento e, segundo (KESHAVARZ, 2006), possibilita o processamento das seguintes informações:

- Capturar o estado do usuário;
- Analisar eventos, e
- Gerar resposta para os eventos.

A partir da evolução dos sensores de imagens, técnicas que usam câmeras para a detecção de acidentes vem emergindo. É proposto por (HSU; ABDEL-MOTTALEB; JAIN, 2002) um sistema baseado em detecção de quedas e transmissão ao vivo de vídeo via celular para conhecimento do estado do usuário. Uma outra aplicação importante foi feita por (LO; WANG; YANG, 2005), a qual é realizada a determinação da postura usando o posicionamento elíptico e histograma, como pode ser visto na Figura 3. A partir da determinação da postura via monitoramento constante, é possível detectar quedas e acidentes somente identificando pequenas variações na postura.

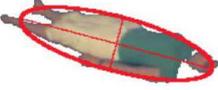
Original Image	Human Mask	Detected Posture
		<p style="text-align: center;">Lying down</p>

Figura 3 – Detecção de pessoa usando posicionamento elíptico. Fonte:(BRAMBERGER et al., 2006)

O sistema proposto por (KESHAVARZ, 2006) é implementado usando uma Rede de Sensores e funciona da seguinte forma:

- Uma pulseira equipada com um acelerômetro e um indicador de força de sinal é colocada no usuário;

- A pulseira se comunica com 3 nós de rede espalhados pelo ambiente, indicando um local aproximado do usuário;
- O indicador de posição é usado para ativar os sensores de imagem apontando para a posição mais precisa possível, quando ocorre uma queda;
- Um alerta de queda é enviado pela pulseira quando ocorre uma mudança significativa na medição realizada pelo acelerômetro,
- Processamento de imagem é utilizado para analisar a situação e indicar a postura do usuário quando ocorre um alerta. Esse mecanismo ajuda a reduzir falsos positivos.

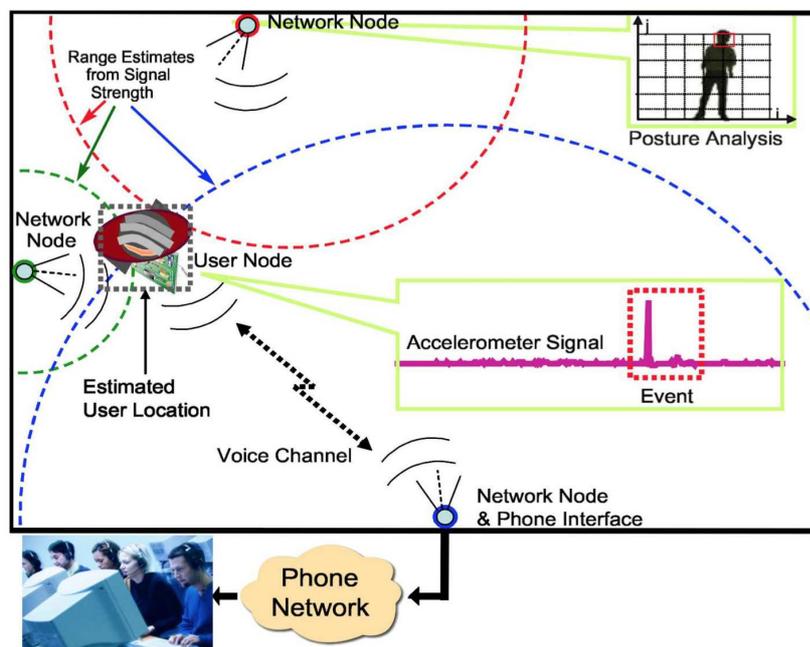


Figura 4 – Esquema de um protótipo de Rede de Sensores. Fonte:(KESHAVARZ, 2006)

O protótipo do sistema feito por (KESHAVARZ, 2006), como pode ser visto na Figura 4, consiste basicamente de 4 elementos:

1. Três nós de rede estáticos associados a câmeras de vídeo, instalados no teto do ambiente;
2. A pulseira utilizada pelo usuário, que se comunica com os nós estáticos através de um link de rádio;
3. Um transmissor de voz localizado na pulseira, que pode ser usado para comunicação, e
4. Uma interface telefônica, anexada a um dos nós da rede, que usa uma linha de telefone para fazer ligações aos interessados.

Assim, o sistema definido por (KESHAVARZ, 2006) faz uma análise distribuída da cena, a qual cada uma das câmeras processa independentemente a imagem para tentar identificar um ser humano no local. Considerando que não pode-se esperar que uma única câmera capture todas as informações de uma cena, a identificação do estado do usuário é feito a partir da análise colaborativa das imagens obtidas, e pré-processadas, por cada uma das câmeras presentes no ambiente.

Como resultado, pode-se esperar um sistema mais confiável, que gerará relatórios mais precisos e que diminuirá consideravelmente o número de falsos positivos.

2.2.3 Ambiente Inteligente para Detecção de Objetos centrado no Usuário usando Redes de Câmeras

A abordagem de detecção de objetos fundamentada em imagens tem como base modelos de local de aparição, cujo sistema aprende a partir do padrões gerados pelas imagens coletadas pelas Redes de Câmeras, como no esquema da Figura 5 (PINZ, 2005). Uma técnica de detecção de objetos é proposta por (PEURSUM; WEST; VENKATESH, 2005), onde rotulam os segmentos da imagem que tem objetos baseados na postura do usuário. Uma outra técnica é usada para detecção de objetos manipuláveis, (GUPTA; DAVIS, 2007) propõem que a movimentação da manipulação, alcance do movimento, reação do objeto e evidência do objeto na cena geram dados suficientes para a detecção do mesmo.

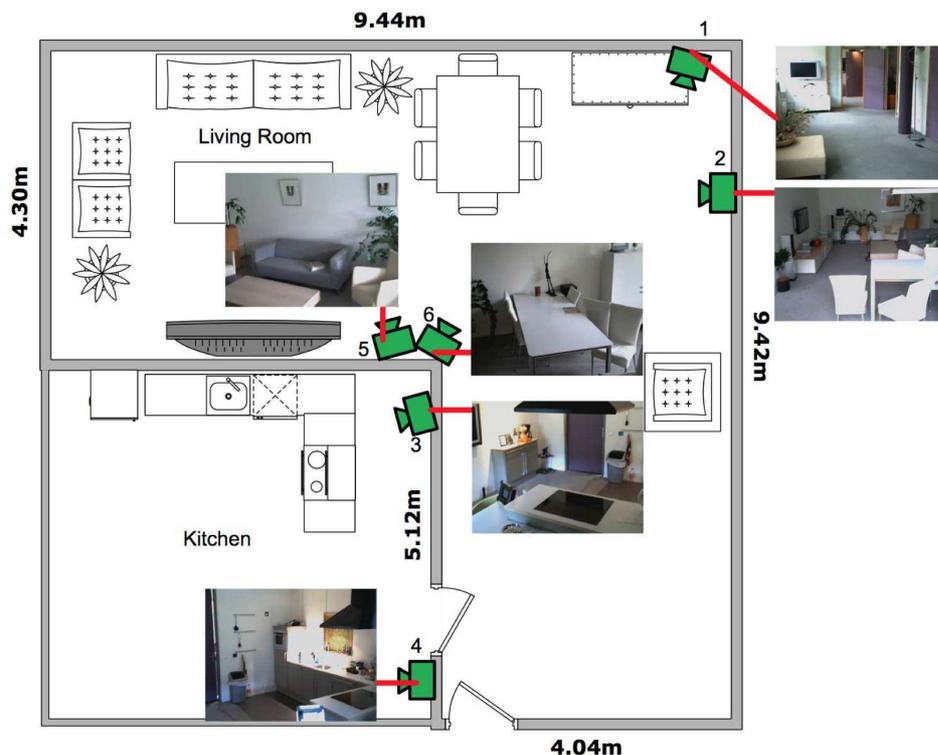


Figura 5 – Exemplo de um esquema de Rede de Câmeras. Fonte:(WU; AGHAJAN, 2010)

A proposta de (WU; AGHAJAN, 2010) é que as atividades humanas oferecem informações importantes para o reconhecimento de objetos em imagens de ambientes conhecidos e é baseada em duas observações:

1. Ambientes normalmente possuem muitos objetos com tamanhos e formas diferentes. As câmeras da rede normalmente capturam diversas coisas do ambiente, de modo que as imagens obtidas possuem um grande número de objetos e alguns objetos pequenos, que, muitas vezes, só podem ser descobertos pela interação do usuário com eles;
2. Alguns espaços onde o usuário se encontra são melhores definidos pela interação feita no local do que por objetos contidos no ambiente. Nem todos os locais de um ambiente possuem definições exatas do que estará presente no mesmo, como uma sala de reuniões com sua mesa e cadeira, ou uma cozinha com sua geladeira, fogão e microondas.

A partir das observações feitas, (WU; AGHAJAN, 2010) definem 3 (três) tipos de comportamentos dos objetos baseado nas atividades do usuário ou na proximidade com outros objetos.

- Ações diretas: entrar e sair por uma porta, sentar ou levantar de um sofá;
- Ações sequenciais: referem-se a uma relação temporal de atividades subsequentes. Por exemplo, pegar coisas na geladeira, colocar no microondas, ir à mesa de jantar para comer, e
- Ações concorrentes: o usuário pode interagir com objetos também à distância, gerando um cenário como sentar no sofá e assistir televisão.

Baseado nos estudos preliminares, (WU; AGHAJAN, 2010) definem que o *framework* gerado terá o processamento de imagens separado do processamento semântico dos tipos de objeto. Dessa forma, o algoritmo definido será dividido em camadas, como podemos ver na Figura 6.

A primeira camada, ou a camada inferior, realiza a análise da imagem para que possa ser extraída a atividade que está sendo realizada. A segunda camada, ou camada do meio, realiza o reconhecimento da atividade, que serve de entrada para a terceira camada. A terceira camada, ou camada superior, baseada na atividade do usuário instancia uma base de conhecimento por meio da Cadeia de Markov (NORRIS, 1998). A base de conhecimento pode ser usada para obter a distribuição de probabilidade espacial de um objeto que possibilitará inferir um certo objeto.

O sistema proposto por (WU; AGHAJAN, 2010) é diferente do outros, pois não se preocupa em detectar o objeto conforme ele aparece na imagem, inferindo a localização do objeto de acordo com a interação do usuário com o mesmo.

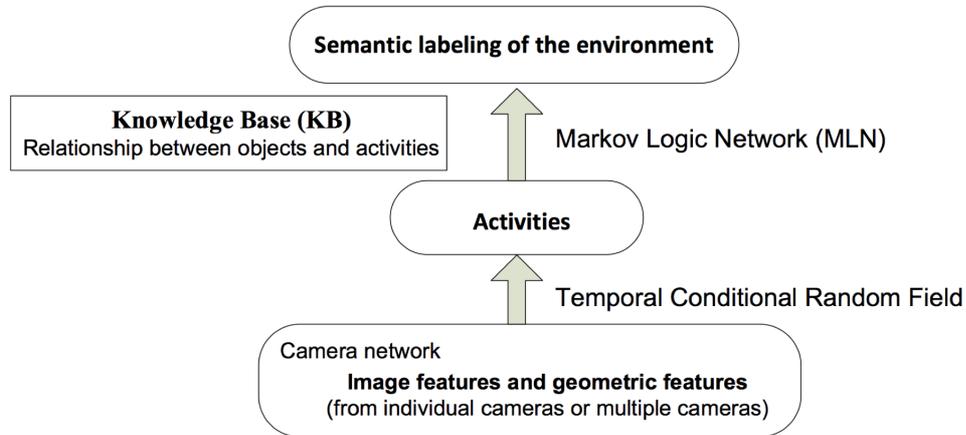


Figura 6 – Estrutura do algoritmo usado para detecção de objetos. Fonte:(WU; AGHAJAN, 2010)

2.3 Monitoramento de Acidentes para Assisted Living

Segundo (AGHAJAN et al., 2007), o uso de redes de sensores para aumento da qualidade de serviços de cuidado monitorado e *assisted living* vêm crescendo. De acordo com os autores, as redes de sensores podem ser utilizadas para realizar deduções a partir dos dados adquiridos, provendo observações e monitoramento do ambiente, mas ressaltam que essas redes possuem muito potencial para aumentar de forma drástica o desenvolvimento de sistemas centrado no usuário, monitorando e prevendo eventos nocivos ao usuário. É lembrado por (AGHAJAN et al., 2007) que apesar de sistemas tradicionais conseguirem prover soluções relacionadas a esses tipos de eventos, eles acabam por limitar a independência do usuário.

Mesmo que existam pesquisas sendo conduzidas para o aumento de qualidade de vida de idosos com demência, os usuários devem obrigatoriamente interagir ou fazer uso de vários dispositivos físicos diferentes. É proposto por (AGHAJAN et al., 2007) um *framework* que visa sobrepor essas duas suposições, mas mantendo as premissas básicas, que são: a detecção de eventos perigosos e notificações aos cuidadores. O foco do sistema deste sistema está na propensão dos idosos que sofrerem quedas:

- Segundo (SHAW et al., 2003), elas são as maiores causas de incapacitação e perda de independência, causando fraturas graves, depressão e isolamento social;
- Também de acordo com (SHAW et al., 2003), a queda é a maior causa de morte acidental, pois aproximadamente 35% (trinta e cinco por cento) das pessoas com

mais de 65 (sessenta e cinco) anos enfrentam uma queda fatal todo ano, e

- Apesar de menos de 10% (dez por cento) das quedas com idosos resultarem em fratura, um quinto das quedas com idosos necessitam de atendimento médico, o que segundo (GILLESPIE et al., 2001) representa um fardo significativo para o serviço de saúde.

O sistema implementado por (AGHAJAN et al., 2007) torna o uso de dispositivos físicos opcional, e também diminui de forma considerável o tamanho dos mesmos, pois faz uso de um pequeno “distintivo” que possibilita a captura de posicionamento, envio do sinal de um acelerômetro e comunicação por voz. A associação do distintivo usado pelo usuário a uma rede de sensores de imagem, monta a base do sistema para a realização de interpretações sobre o estado da pessoa sob cuidados.

As imagens capturadas pelos sensores de imagens são utilizadas para avaliar se a pessoa está se movendo apropriadamente, realizando atividades da vida diária, se teve uma queda ou até mesmo se alguma inatividade inesperada ocorreu (AGHAJAN et al., 2007). O sistema modelado segue as seguintes etapas:

1. Os sensores de imagem são usados para avaliar a postura do usuário antes, durante e depois de um alerta, que poderá ou não ser gerado pelo distintivo;
2. Cada sensor de imagem faz a análise independente da situação, a qual é sucedida por uma técnica de raciocínio colaborativo é aplicada ao resultados das câmeras gerando um relatório,
3. O sistema faz o encaminhamento do relatório para um *call center*, ou cuidador.

O circuito de voz permite também que o sistema crie um *link* de voz com o cuidador automaticamente ou a partir de uma requisição do usuário.

A utilização de uma solução com vários sensores de imagem, permitiu que o sistema feito por (AGHAJAN et al., 2007) alcançasse um dos objetivos desejados, o qual era tornar o uso do distintivo opcional. No caso do usuário não estar usando o distintivo, o mesmo não poderá ser usado para detectar uma queda. Sendo assim as câmeras serão ativadas frequentemente, ou a partir de um *timeout*, ou se não existir resposta do distintivo. Assim, a postura do usuário é determinada de acordo com a arquitetura em camadas definida na Figura 7.

A arquitetura proposta por (WU, 2007) funciona da seguinte maneira:

- As imagens são capturadas pelos diferentes sensores e o fundo é removido;
- As partes do corpo são segmentadas por meio da distribuição de cores;

- Vetores de movimentação são associados as partes do corpo por meio da análise da movimentação, movimentação global e das cores dos segmentos (AGHAJAN et al., 2007) e (WU, 2007);
- A partir dessas imagens, é realizada uma análise colaborativa, que tem como resultado o reconhecimento de um gesto.

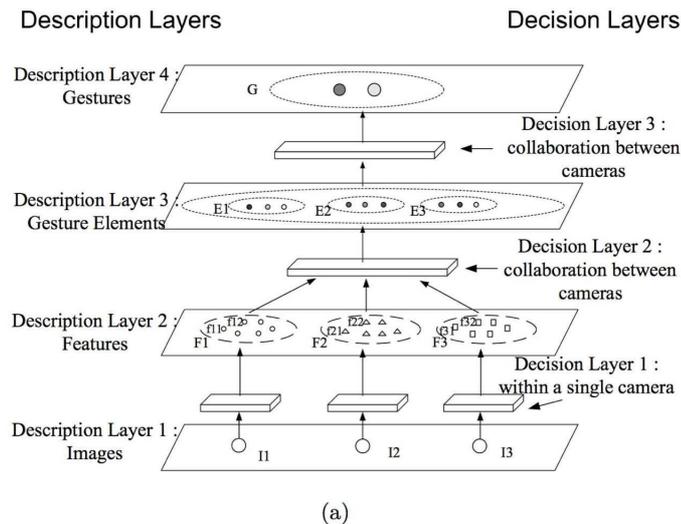


Figura 7 – Arquitetura usada para definir a postura do usuário. Fonte:(AGHAJAN et al., 2007)

A Figura 8 mostra o funcionamento do algoritmo proposto por (AGHAJAN et al., 2007).

É conhecido que captura de imagens acaba trazendo a tona problemas de privacidade e ética, influenciando de forma decisiva na aceitação dos usuários. Entretanto, (HEANEY, 2006) afirma que 90% (noventa por cento) da população idosa deseja ficar em suas casas, portanto, a solução proposta por (AGHAJAN et al., 2007) luta para atingir um meio termo entre otimização do cuidado e aceitação do usuário.

2.4 Técnicas para Reconhecimento de Ações

Deteção de eventos em vídeo vem se tornando um das mais importantes aplicações das áreas de visão computacional, particularmente no contexto de reconhecimento de ações (KE; SUKTHANKAR; HEBERT, 2005b). As abordagens mais comuns dadas a esses problemas são, normalmente, categorizadas em:

- Baseadas em modelo, ou
- Baseadas em aparência.

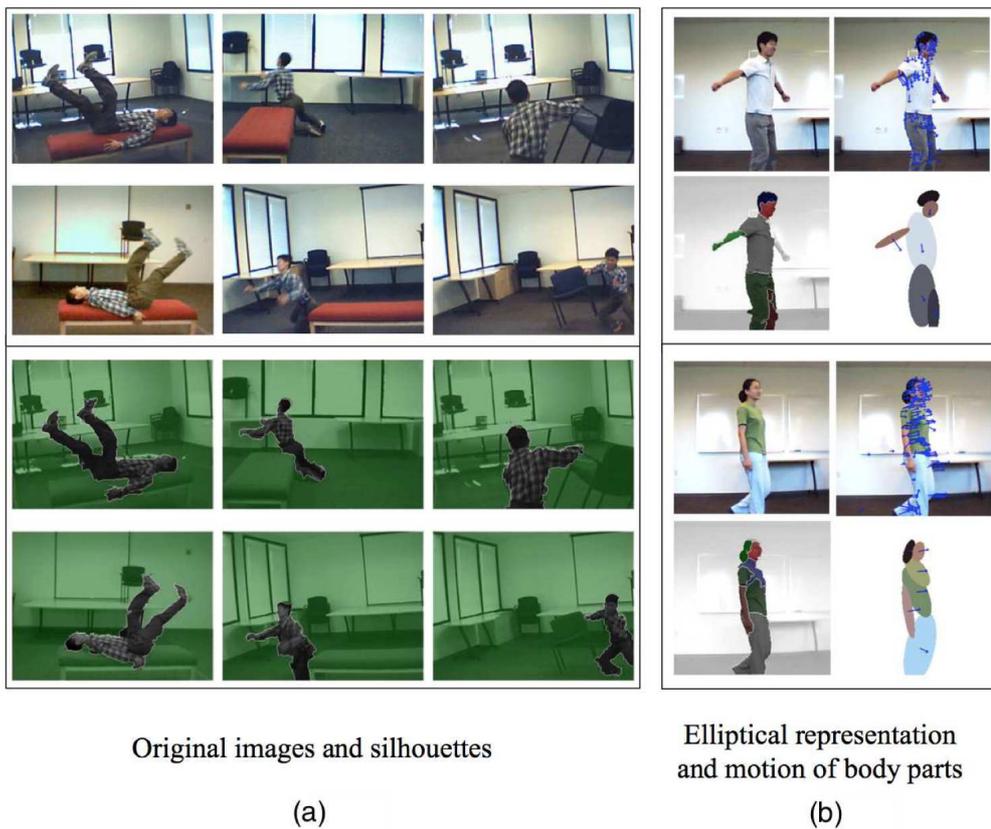


Figura 8 – (a) Remoção do fundo e definição da silhueta. (b) Segmentação das partes do corpo e associação dos vetores de movimentação. Fonte: (AGHAJAN et al., 2007)

Geralmente, as técnicas baseadas em modelo são preferíveis quando se existe um bom modelo prévio do que se deseja detectar, enquanto as técnicas baseadas em aparência se estendem mais facilmente à diferentes atividades (KE; SUKTHANKAR; HEBERT, 2005b).

Recentemente surgiram novas abordagens, como a exploração de pontos de interesse em vídeos (GAUTAMA; HULLE, 2002) e pontos de interesse no espaço tempo (*Spatio-Temporal Interest Points* - STIPs).

2.4.1 *Mean shift* temporal para Reconhecimento de Ações

É afirmado em (WANG et al., 2004) que a técnica do *mean shift* (da tradução literal, algo como “mudança de média”), é um estimador de densidade, também conhecido como *clustering* (agrupamento) não paramétrico, que vem sendo usado para segmentação de vídeos e imagens. O *mean shift* tradicional usa como núcleo uma circunferência simétrica para estimar o local com maior densidade de pontos.

A Figura 9 ilustra o funcionamento do *mean shift*, onde P_1 é o ponto inicial e a circunferência verde define a primeira área de *cluster*, o algoritmo do *mean shift* é executado mais uma vez e o centro da circunferência é movido para o local com maior densidade

dentro da circunferência. O novo centro é definido em P_2 e uma nova circunferência é desenhada, assim é feito até que o valor da densidade em P_n seja igual a densidade em P_{n+1} (DEMENTHON, 2002).

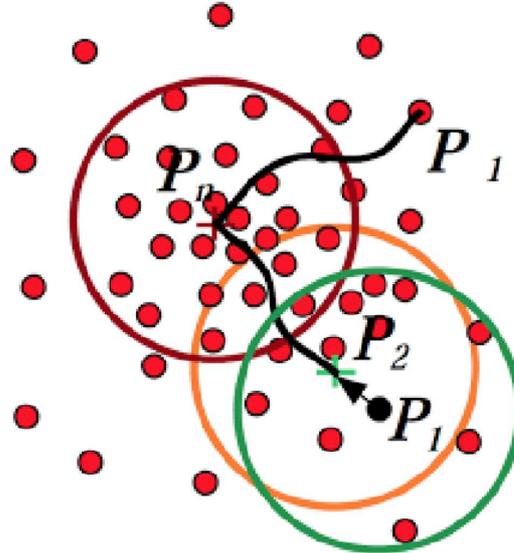


Figura 9 – Processo de funcionamento da técnica *mean shift*. Fonte:(AGHAJAN et al., 2007)

É proposto por (KE; SUKTHANKAR; HEBERT, 2005a) um algoritmo de *mean shift* temporal para agrupamento de regiões no espaço-tempo em vídeos. A extração de regiões no espaço-tempo é frequentemente utilizada como o primeiro passo do pré-processamento de sistemas de reconhecimento de ações, como pode ser visto em seu trabalho.

Adicionalmente, ressaltam a importância do algoritmo proposto para a redução da quantidade de dados a serem processados pelos algoritmos de alto nível, que fazem as inferências. É essencial ter um algoritmo eficiente de segmentação de imagem, que processa rapidamente uma grande quantidade de dados recolhidos dos vídeos.

Um algoritmo eficiente de *clustering*, que pode automaticamente extrair regiões no espaço-tempo, permite o reconhecimento de uma variedade maior de ações. Foi proposto por (KE; SUKTHANKAR; HEBERT, 2005a) uma nova forma de usar o *mean shift*, porque, apesar de todos os esforços na tentativa de melhorar a técnica padrão do *mean shift*, ele ainda é muito lento para uso em vídeo.

O algoritmo proposto por (KE; SUKTHANKAR; HEBERT, 2005a) segue a seguinte sequência: cada pixel i recebe uma posição (x_i, y_i) e o tempo (t) recebe uma cor (r_i, g_i, b_i) , com essas informações é realizado o cálculo do local de aparição e a direção da movimentação. Cada *pixel* é transformado em um vetor, que contém sua aparência e informações sobre a direção da sua movimentação. A partir desse tratamento feito no *pixel*, o *mean shift* é aplicado, mas recebendo as seguintes alterações para que o tempo

de processamento possa ser reduzido:

- Acelerar a procura pelo vizinho mais próximo. Isso é possível pois, descartando o fundo cada *frame*, possui pequenas diferenças em relação ao anterior, portanto, o centro de agrupamento do *frame* seguinte estará próximo do atual, diminuindo a procura pelo próximo centro, e
- Reduzir o número de iterações para convergir para o centro de agrupamento ideal. A redução do número de iterações será alcançado, alterando a forma de inicializar o modo. O modo é representado por y e deve ser inicializada em (x_i, y_i, t_i) como y_t^i ao invés da forma da equação tradicional $y_{(i, I)}^{(t, I)}$

Os resultados dessas alterações são evidentes, pois a distância dos *pixels* da versão original para o algoritmo proposto (KE; SUKTHANKAR; HEBERT, 2005a) é de 4.9 *pixels*, o que acelera a busca pelo próximo vizinho. O número de iterações foi reduzido praticamente pela metade, o que aumentou a velocidade de convergência em quase 100% (cem por cento).

2.4.2 Reconhecimento de Ações via Processamento de Imagens

O objetivo do reconhecimento de ações é localizar um evento de interesse em vídeo, tanto em espaço quanto em tempo. A tendência nessa área é o uso de técnicas baseadas em análise volumétrica de vídeo, cuja uma sequência de imagens é tratada e um volume no espaço-tempo pode ser extraído (KE; SUKTHANKAR; HEBERT, 2007b).

A análise volumétrica não pode ser facilmente generalizada, devido as modificações na aparência, como luz e posição da câmera. Essas observações motivaram o uso das técnicas que são robustas para aparência, as baseados na forma (*shape-based*) e as baseadas em fluxo (*flow-based*) (KE; SUKTHANKAR; HEBERT, 2007b).

Técnicas baseadas em silhueta tentam reconhecer ações por meio da caracterização da silhueta no espaço-tempo (GORELICK et al., 2005). Entretanto existem duas grandes limitações, pois essas técnicas assumem que a silhueta é precisamente definida e que a pessoa é representada como uma só região. (KE; SUKTHANKAR; HEBERT, 2007b) lembram que um outro problema, mas não menos relevante, é que essas técnicas ignoram a movimentação interna do objeto.

Segundo (KE; SUKTHANKAR; HEBERT, 2005b), as técnicas baseadas em fluxo estimam o campo de visão, o fluxo ótico entre *frames* adjacentes é usado como base para o reconhecimento da ação. A vantagem das técnicas de fluxo é que as mesmas não necessitam de remoção do fundo por serem invariantes em relação a mudanças na aparência, entretanto, por ser um técnica muito grosseira, ela acaba apresentando fluxos muito si-

milares para *frames* com pequena variação no tempo (KE; SUKTHANKAR; HEBERT, 2007b).

Tendo definido claramente o que é uma técnica baseada em fluxo e uma técnica baseada em silhueta fica fácil perceber a natureza complementar das duas. É utilizado por (KE; SUKTHANKAR; HEBERT, 2007b), como exemplo, uma bola texturizada rolando, a qual possui uma forma definida, mas a extração da silhueta não possibilita distinguir se a mesma está rolando, entretanto o cenário pode ser facilmente reconhecido usando uma técnica de fluxo.

Algumas ações podem ser difíceis de serem distinguidas usando somente fluxos, ou seja, é importante a associação das duas técnicas. Por isso (KE; SUKTHANKAR; HEBERT, 2007b) defendem que técnicas baseadas em fluxo e forma devem ser usadas juntas para o reconhecimento de ações.

Foi feito por (KE; SUKTHANKAR; HEBERT, 2007b) a junção das duas técnicas em um único algoritmo, e o aplica em um vídeo de 30 (trinta) minutos de uma partida de tênis. Para realizar uma análise comparativa, (KE; SUKTHANKAR; HEBERT, 2007b) executam nesse mesmo vídeo três outras técnicas:

- Técnica do Fluxo;
- Técnica da Silhueta/Forma, e
- Técnica da Silhueta/Forma Normalizada.

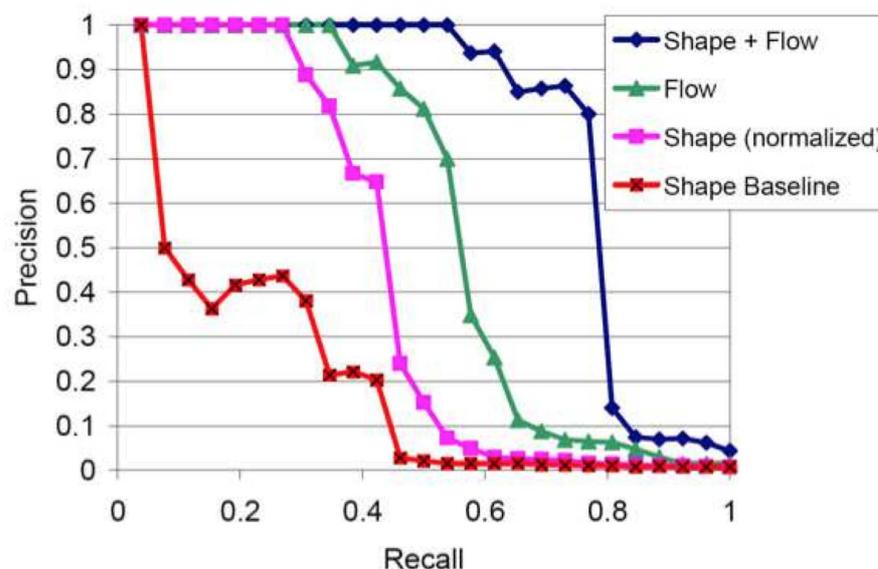


Figura 10 – Gráfico de Precisão por Relevância, comparando quatro algoritmos. Fonte:(KE; SUKTHANKAR; HEBERT, 2007b)

Os resultados podem ser vistos na Figura 10.

A proposta feita por (KE; SUKTHANKAR; HEBERT, 2007b), que correlaciona a técnica de fluxo com a de silhueta, se mostrou eficiente em relação as outras, sendo aplicadas separadamente e alcançando 80% (oitenta por cento) de relevância por 80% (oitenta por cento) de precisão, enquanto a técnica que mais se aproximou foi a de fluxo chegando a aproximadamente 60% (sessenta por cento) de relevância e aproximadamente 70% (setenta por cento) de precisão.

Apesar do algoritmo proposto por (KE; SUKTHANKAR; HEBERT, 2007b) ter como objetivo a detecção e localização de eventos, eles submeteram o algoritmo à base de dados de ações KTH (SCHULDT; LAPTEV; CAPUTO, 2004). A base de dados de ações KTH possui 25 (vinte e cinco) pessoas efetuando 6 (seis) ações diferentes em 4 (quatro) ambientes diferentes. Essa base de dados é considerada difícil, pois possui mudanças drásticas na iluminação, nas roupas e mudanças de escala, como pode ser visto na Figura 11.



Figura 11 – Exemplo de imagem da base KTH, mostrando a diferença na escala. Fonte:(KE; SUKTHANKAR; HEBERT, 2007b)

O objetivo dos experimentos com a base KTH é realizar a classificação do vídeos nas seguintes categorias: andando, trotando, correndo, lutando, batendo palma e acenando. Os resultados obtidos por (KE; SUKTHANKAR; HEBERT, 2007b), usando o método combinado, tiveram uma precisão de quase 81% (oitenta e um por cento), necessitando de um número menor de *templates* para alcançar tal precisão, em comparação ao fluxo e a silhueta, como podemos ver na Figura 12.

Mesmo não sendo feito especificamente para classificação de vídeo inteiros, o algoritmo de (KE; SUKTHANKAR; HEBERT, 2007b) se mostrou competitivo em relação a estudos feitos recentemente.

Em (KE; SUKTHANKAR; HEBERT, 2007b) é apresentado, com as informações da Tabela 1, que a combinação entre fluxo e forma resultam em um algoritmo competitivo com o estado-da-arte.

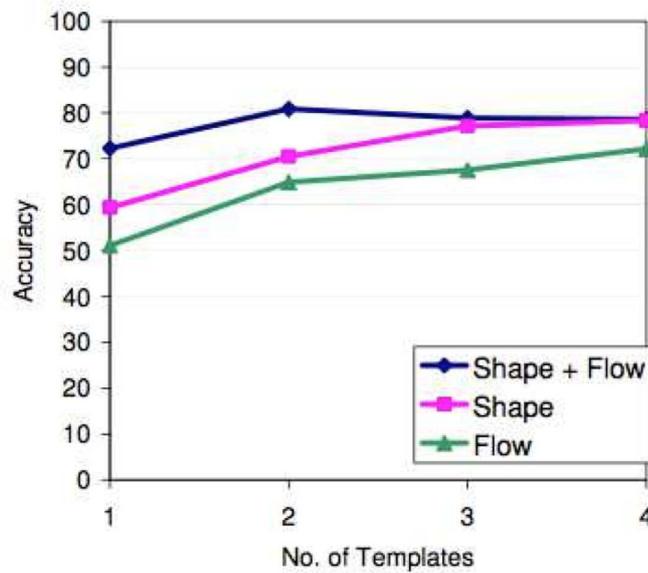


Figura 12 – Precisão por número de *templates*. Fonte:(KE; SUKTHANKAR; HEBERT, 2007b)

Trabalho Relacionado (%)	Precisão [μm]
(KE; SUKTHANKAR; HEBERT, 2007b)	80.9%
(KE; SUKTHANKAR; HEBERT, 2005b)	63.0%
(SCHULDT; LAPTEV; CAPUTO, 2004)	71.7%
(DOLLAR et al., 2005)	81.2%
(NIEBLES; WANG; FEI-FEI, 2008)	81.5%
(JIANG; DREW; LI, 2006)	84.4%

Tabela 1 – Comparativo de precisão com outros estudos. Fonte:(KE; SUKTHANKAR; HEBERT, 2007b)

2.4.3 Reconhecimento de Ações via Processamento de Imagens usando Sensores de Profundidade

Atualmente, a maioria das técnicas de reconhecimento de ação só consideram o movimento do corpo humano em $x-y-t$, pois a captura é realizada usando câmeras que capturam somente cor, dessa forma, as informações relacionadas a profundidade são perdidas. Todavia, tanto o corpo humano quanto os movimentos realizados por ele, são feitos em 4 (quatro) dimensões, $x-y-z-t$ (ZHAO et al., 2012).

O recente progresso no desenvolvimento de sensores de profundidade, por exemplo o *Microsoft Kinect*, tem aumentado o interesse no desenvolvimento do reconhecimento de ações com sensores *Red Green Blue Depth* (RGBD) (SHOTTON et al., 2011). Sensores como o *Microsoft Kinect* tem facilitado as técnicas de captura do movimento, pois define a localização das juntas do esqueleto humano (ZHAO et al., 2012). Entretanto, como aponta (ZHAO et al., 2012), câmeras de monitoramento normalmente estão em uma posição alta, o que acaba gerando oclusão parcial e dificultando a captura do esqueleto. Por isso, o reconhecimento de ações deve depender principalmente da cor das imagens e da

profundidade.

Ciente dos fatores abordados, (ZHAO et al., 2012) propõem uma *framework* que combina RGB (*Red Green Blue*) e profundidade para realizar o reconhecimento de ações. Para facilitar a representação, (ZHAO et al., 2012) dividem a ação em dois passos:

- Representação de regiões de informação (*features*), e
- Pontos de interesse;

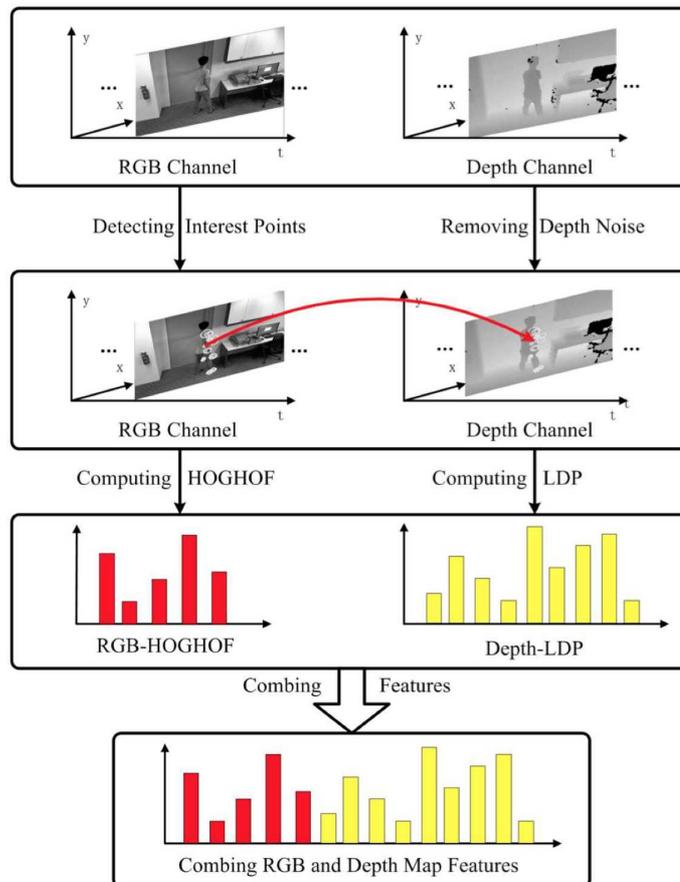


Figura 13 – *Framework* que combina RGB e profundidade. Fonte:(ZHAO et al., 2012)

A representação de regiões de informação é feito usando HOG (Histograma de Gradiente) e HOF (Histograma de Fluxo), que são técnicas populares. A técnica usada para os pontos de interesse será a de Pontos de Interesse no Espaço-Tempo, em inglês é *Spatio-Temporal Interest Points* (STIPs), proposta por (LAPTEV, 2005). Para completar a *framework* proposta, (ZHAO et al., 2012) usam LDP (*Local Depth Pattern* ou Padrão de Profundidade do Local, em português) para representar regiões de informação que possuem profundidade.

É ressaltado por (ZHAO et al., 2012), que existem dois canais para obter pontos de interesse, RGB e profundidade, isso resulta na produção de várias representa-

ções da ação através da combinação de diferentes formas de geração de pontos de interesse. A abordagem da *framework* proposta pode ser vista de forma bem clara na Figura 13. Ainda com o uso de sensores de profundidade, (KOSMOPOULOS et al., 2013) propõem um abordagem de combinação de RGB e profundidade para reconhecimento de ações. (KOSMOPOULOS et al., 2013) também fazem uso de STIPs, mas, diferente de (ZHAO et al., 2012), a combinação é feita com HMIs (Histórico de Movimento das Imagens).

A metodologia proposta por (KOSMOPOULOS et al., 2013) começa com um passo de extração, combinando duas diferentes fontes: vídeos RGB e vídeos com profundidade. Em seguida, o vídeo com profundidade é separado em dois vetores, um com movimentos para frente e outro com movimentos para trás. Por último todas as sequências são combinadas gerando a entrada para um classificador que decidirá qual ação está sendo realizada. A Figura 14 representa os passos tomados antes da combinação das sequências.



(a) Color image



(b) Backward MHI (from depth)



(c) Forward MHI (from depth)

Figura 14 – (a) Vídeo RGB (b) MHI para frente (c) MHI para trás.. Fonte:(KOSMOPOULOS et al., 2013)

3 Problema

Tendo em vista as limitações que o AVC traz, listadas por (PERLINI, 2000), foi feito um estudo com 35 (trinta e cinco) cuidadores para identificar as atividades mais realizadas para o auxílio a pessoa vitimada. De acordo com (PERLINI, 2000), essas atividades dependerão do grau de incapacidade da pessoa vítima do AVC.

Os estudos feitos por (PERLINI, 2000) mostram que as vítimas do AVC possuem alta dependência motora e sua maioria precisa de ajuda para as atividades simples do dia-a-dia, como trocar de roupa, se locomover, tomar banho, escovar os dentes e até mesmo pentear o cabelo.

Como citado na Seção 2.1, (PERLINI, 2000) resalta que a hemiplegia (disfunção motora que incapacita o paciente de movimentar os membros do lado oposto ao dano cerebral (SMELTZER; BARE, 2005)) é um dos déficits motores apresentados pelo paciente e que altera a sua forma de andar em relação a uma pessoa comum.

Um estudo realizado por (OTTOBONI; FONTES; FUKUJIMA, 2002) comparou a marcha normal e a marcha de um paciente hemiplégico, concluiu que a pessoa hemiplégica apresenta anormalidades biomecânicas significantes que alteram sua marcha em relação a normal. Os autores identificaram 5 (cinco) pontos principais que caracterizam a marcha do paciente hemiparético:

- Aumento da flexão do quadril;
- Diminuição do deslocamento lateral pélvico;
- Hiperextensão do joelho;
- O contato inicial do pé antes feito pelo calcanhar agora é feito pelo antepé, diminuindo a força propulsiva,
- Diminuição da dorsiflexão (movimento de aproximação do dorso do pé a parte anterior da perna (UFSC, 2004)).

Das limitações elencadas por (PERLINI, 2000), vale também destacar a dificuldade de compreensão e expressão dos pensamentos, normalmente causada pela dificuldade de comunicação oral e escrita (SKILECK, 1996). Tais limitações acabam reduzindo a interação do paciente com o ambiente a sua volta (OTTOBONI; FONTES; FUKUJIMA, 2002).

Com todos os problemas apresentados pelos pacientes hemiplégicos o principal problema a ser tratado por esse trabalho será a marcha do hemiplégico. A partir das diferenças da marcha hemiplégica com a marcha normal será realizada a construção de um Sistema de Reconhecimento de Ações com o objetivo de diferenciar esses dois tipos de usuários. A diferenciação entre as duas marchas será o primeiro passo na construção de um sistema maior que auxiliará os cuidadores, diminuindo a necessidade de monitoramento, e o paciente, aumento a independência.

3.1 Proposta da Solução

Realizando estudos da bibliografia sobre Ambientes Inteligentes ((KESHAVARZ, 2006) e (KLEINBERGER et al., 2007)), Monitoramento de Acidentes (AGHAJAN et al., 2007) e também técnicas de Reconhecimento de Ações via Processamento de Imagens ((KE; SUKTHANKAR; HEBERT, 2007b) e (KE; SUKTHANKAR; HEBERT, 2005b)), não foi possível encontrar uma solução pronta que atendesse o objetivo principal desse trabalho, mas foram encontradas técnicas que poderão ser usadas na solução do problema.

O uso de sensores de profundidade foi escolhido como técnica principal para desenvolvimento do sistema, pois os pacientes hemiplégicos apresentam a debilidade motora em todo o lado oposto ao do dano cerebral, sendo assim o uso de sensores que necessitassem ser colocados no paciente para o correto funcionamento do sistema poderiam comprometer a qualidade das informações geradas. Tendo como objetivo o aumento da qualidade de vida desse paciente, tornar o mesmo dependente do uso de um dispositivo que pode ser difícil de ser colocado não cumpriria com a ideia inicial.

Para atingir os objetivos deste trabalho, serão utilizadas como base os trabalhos de (KE; SUKTHANKAR; HEBERT, 2007b), (KE; SUKTHANKAR; HEBERT, 2005b), (KOSMOPOULOS et al., 2013) e (ZHAO et al., 2012). Além de abordagens usadas nos sistemas de propostos por (KESHAVARZ, 2006) e (WU; AGHAJAN, 2010).

O sistema será constituído de dois módulos principais, o Módulo de Criação de *Templates* e o Módulo de Monitoramento e Reconhecimento de Ações.

O sistema também será preparado para receber um módulo de alerta, que não será desenvolvido nesse trabalho. Esse módulo de alerta poderá ser acoplado ao Módulo de Reconhecimento de Ações, fazendo uso das informações produzidas por este módulo para gerar os alertas.

3.1.1 Módulo de Criação de *Templates*

No trabalho feito por (KE; SUKTHANKAR; HEBERT, 2007b) a extração em 3D do volume da pessoa no espaço-tempo é feito por meio da segmentação por cores, com

o auxílio do algoritmo *mean shift*, usando sensores que só capturam a imagem RGB. A proposta de (KE; SUKTHANKAR; HEBERT, 2007b) apresenta uma dificuldade que é a definição do tamanho da circunferência inicial para a execução do *mean shift*, dessa forma, o uso de um sensor de profundidade, que realiza também detecção do esqueleto (e.g., *Microsoft Kinect*) auxiliará na criação dos *templates*.

Dessa forma, o Módulo de Criação faz a extração das informações de todas as juntas do esqueleto no espaço-tempo criando um volume 3D. Esse volume 3D é então processado de acordo com os passos abaixo para a geração dos *templates*:

- O usuário seleciona quais juntas são interessantes para a comparação do movimento feito. Por exemplo, quando o usuário move o braço esquerdo os pontos interessantes são aqueles que compõe o mesmo;
- O usuário entra com o erro máximo que aquele *template* pode atingir para ser considerado um movimento daquele tipo, e
- O usuário seleciona qual será o arquivo de entrada para a geração do arquivo de *template*.

Com as informações acima fornecidas pelo usuário, o arquivo de *template* é então gerado. O arquivo tem o erro máximo que pode ser alcançado para ser considerado um ação daquele tipo, o arquivo contém também todas as juntas de interesse inseridas pelo usuário e por último os pontos são normalizados em relação a junta do pescoço (*SHOULDER CENTER*). A normalização das juntas é realizada para que os movimentos sejam independentes da posição do usuário em relação a câmera.

Ao fim de todo o processamento o *template* deve se parecer com o mostrado na Figura 16, mas diferente da imagem o mesmo só conterá as informações das juntas do esqueleto.

3.1.2 Módulo de Monitoramento e Reconhecimento de Ações

O algoritmo proposto por (KE; SUKTHANKAR; HEBERT, 2007b) faz uso de imagens segmentadas em várias regiões e é baseado no cálculo da distância binária entre as duas formas (A, B) , essa diferença pode ser calculada através da união e interseção das duas regiões $|A \cup B - A \cap B|$. Devido ao fato de usar imagens segmentadas, (KE; SUKTHANKAR; HEBERT, 2007b) adaptam o cálculo da distância entre duas formas para que o mesmo possa funcionar para seu algoritmo.

O primeiro passo é fazer a procura geral na imagem, dado um *template* T de volume $|T|$, o mesmo é movido nas dimensões x, y e t do vídeo V , considerando que temos

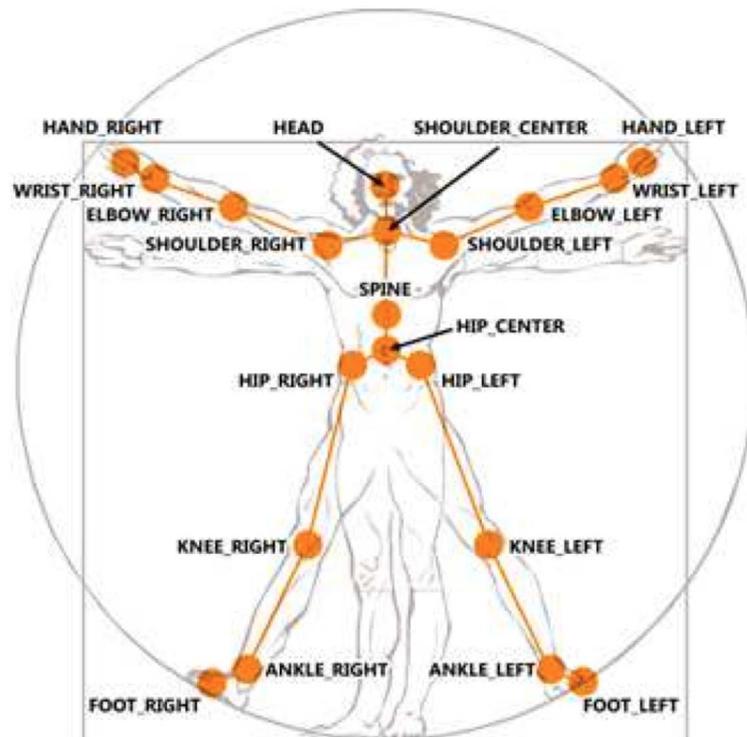


Figura 15 – Juntas do esqueleto reconhecidas pelo *Kinect*. Fonte: (MSDN, 2012)

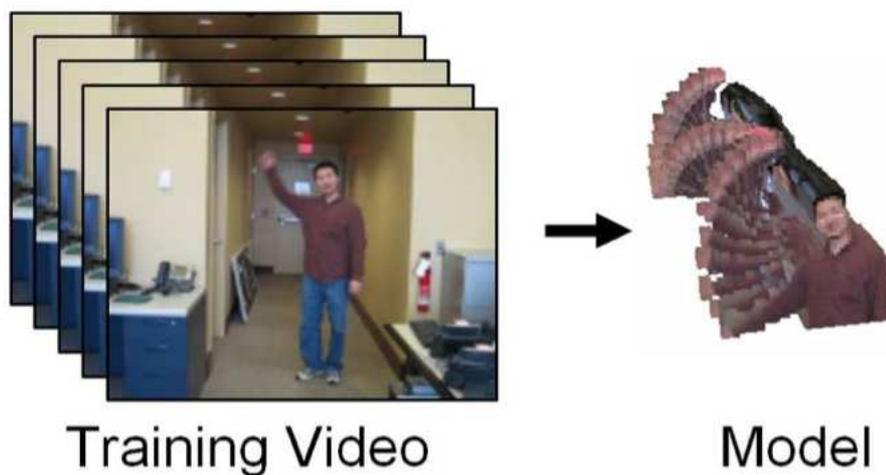


Figura 16 – Aquisição de *Templates*. Fonte: (KE; SUKTHANKAR; HEBERT, 2007b)

um vídeo de volume V . Devido a segmentação do vídeo o volume V pode conter k regiões, sendo V igual a soma das k regiões (KE; SUKTHANKAR; HEBERT, 2007b).

Sendo assim para descobrirmos quais regiões V_i do vídeo segmentado pertencem ao *template*, quatro considerações devem ser feitas: se uma região tem interseção total com o *template*, ela está contida no conjunto mínimo de regiões; se uma região não tem nenhum grau de interseção com o *template*, essa nunca faz parte do conjunto mínimo; os outros dois casos são decididos através de quão sobrepostos os mesmos estão,

se $|V_i \cap T| > |V_i|/2$ então a região é incluída no conjunto mínimo caso contrário não (KE; SUKTHANKAR; HEBERT, 2007a). Dessa forma a distância entre cada uma das regiões e a distância entre o *template* T e o volume V , são calculados de acordo com as fórmulas abaixo.

$$d(T, V_i) = \begin{cases} |T \cap V_i| & \text{if } |T \cap V_i| < |V_i|/2 \\ |V_i - T \cap V_i| & \text{otherwise.} \end{cases}$$

$$d(T, V) = \sum_{i=1}^k d(T, V_i).$$

Figura 17 – Fórmula para cálculo da distância das regiões. Fonte: (KE; SUKTHANKAR; HEBERT, 2007b)

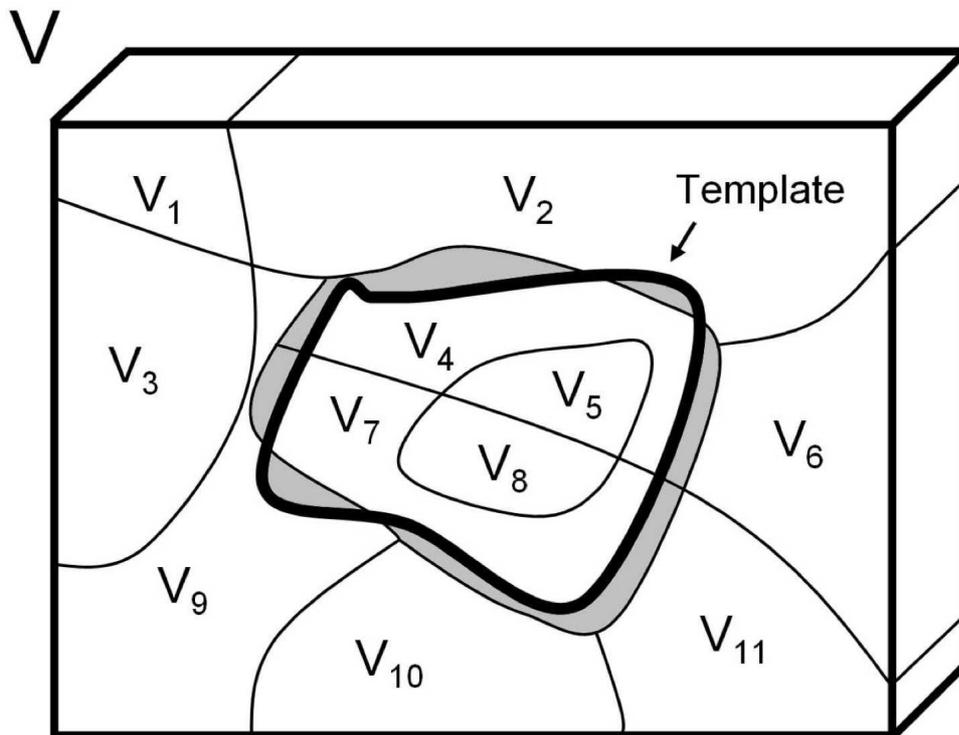


Figura 18 – Resultado da execução do algoritmo. Fonte: (KE; SUKTHANKAR; HEBERT, 2007b)

A partir da abordagem proposta por (KE; SUKTHANKAR; HEBERT, 2007b), o que é realizado no Módulo de Reconhecimento é também a comparação entre a distância do *template* e um volume V , sendo que nesse caso o volume é gerado pelo *Kinect*. Entretanto, a diferença chave da solução a ser desenvolvida do algoritmo proposto acima é que as regiões não são encontradas a partir da segmentação da imagem por cores, mas sim a partir das juntas de interesse anteriormente definidas para o *template*. Ou seja, no

algoritmo de (KE; SUKTHANKAR; HEBERT, 2007b) as k regiões eram segmentos de cor, na solução proposta as k regiões são as juntas do esqueletos.

Para evitar processamento desnecessário de informações, é requisitado no momento da criação dos modelos que sejam definidas as juntas de interesse. Essa abordagem além de diminuir a carga de dados a ser processada, diminui também erros na comparação (falso positivo), pois não estão sendo comparadas juntas irrelevantes.

O Módulo de Reconhecimento de Ações executa sempre que for iniciado o carregamento de todos os *templates* existentes, dessa forma todas as ações com modelos disponíveis são reconhecíveis pelo sistema.

Para validar o funcionamento do sistema proposto, o mesmo será testado em um ambiente real, com um paciente hemiplégico e uma pessoa comum.

4 Desenvolvimento

A partir do problema definido e da proposta de uma solução, iniciou-se o desenvolvimento. Os primeiros passos dados foram a definição dos requisitos e da arquitetura, com a finalização dessa primeira parte foi possível ter a estrutura necessária para dar início à codificação.

Para a etapa de codificação, foi criado um repositório público para o armazenamento e versionamento de todo o código produzido durante o desenvolvimento do sistema.

4.1 Requisitos

4.1.1 Visão Geral

O Sistema de Reconhecimento de Ações desenvolvido nesse trabalho contará com dois módulos, um módulo de criação de *templates* e outro módulo para o monitoramento e reconhecimento das ações.

O módulo de criação de *templates* tem por objetivo capturar, com o uso do *Kinect*, ações executadas pelos usuários, gravar essas ações e então processá-las e transformá-las em modelos que são utilizadas pelo módulo de reconhecimento.

O módulo de reconhecimento de ações tem como objetivo realizar o constante monitoramento das ações que são realizadas em frente à câmera, que está conectada, e possibilitar o reconhecimento das ações sendo executadas. No caso, a câmera é um sensor *Kinect* que colhe as informações do esqueleto e faz a comparação com os *templates* previamente gerados pelo módulo de criação.

4.1.2 Requisitos Funcionais

4.1.2.1 Módulo de Criação de *Templates*

RF1 – O sistema deve coletar as informações do esqueleto do usuário e salvar em arquivo para a geração de *templates* de duas maneiras diferentes: usando um *timer* e por meio de um comando do teclado indicando o início e o fim da captura.

RF2 – O sistema deve carregar todos os arquivos salvos para disponibilizá-los para geração de *templates*.

RF3 – O sistema deve possibilitar que o usuário escolha qual dos arquivos salvos deseja processar e gerar um *template*.

RF4 – O sistema deve possibilitar que o usuário defina o erro máximo de um *template* e as juntas importantes para o reconhecimento.

RF5 – O sistema deve possuir uma janela com a visualização da imagem que o sensor está capturando para melhor definição dos *templates*.

Em um versão futura do sistema, o erro máximo, definido no RF4, deve ser calculado automaticamente a partir da criação de um módulo que seja capaz de receber as informações e comparar com cenas de vários movimentos diferentes, definindo assim de forma automática o erro máximo a ser atingido por aquele *template*.

4.1.2.2 Módulo de Monitoramento e Reconhecimento de Ações

RF1 – O sistema deve carregar todos os *templates* disponíveis para reconhecimento assim que for iniciado.

RF2 – O sistema deve realizar a captura constante das informações de um esqueleto assim que um for detectado, para o a comparação com os *templates*.

RF3 – O sistema deve comparar constantemente as informações capturadas com os modelos disponíveis para efetuar o reconhecimento de uma ação.

RF4 – O sistema deve mostrar na tela uma mensagem toda vez que uma ação for identificada.

RF5 – O sistema deve possibilitar o acesso à imagem que está sendo capturada pelo *Kinect* para confirmação visual da mensagem gerada pelo sistema.

4.1.3 Requisitos Não Funcionais

4.1.3.1 Módulo de Criação de *Templates*

Esse módulo não possui requisitos não funcionais.

4.1.3.2 Módulo de Monitoramento e Reconhecimento de Ações

RNF1 – O sistema deve realizar o reconhecimento das ações em tempo real, para maior segurança dos usuários.

RNF2 – O sistema deve gerar as mensagens em um padrão que facilite o acoplamento a sistemas de comunicação para envio de mensagens de celular, de e-mails e outros tipos de alerta.

4.2 Arquitetura

A característica modular do sistema simplifica a arquitetura do mesmo, caso o sistema de criação de *templates* fosse junto com o de reconhecimento de ações a arquitetura seria muito mais complexa.

A seguir, podemos ver os diagramas da arquitetura de cada um dos módulos. Os diagramas abaixo representam os módulos em funcionamento.

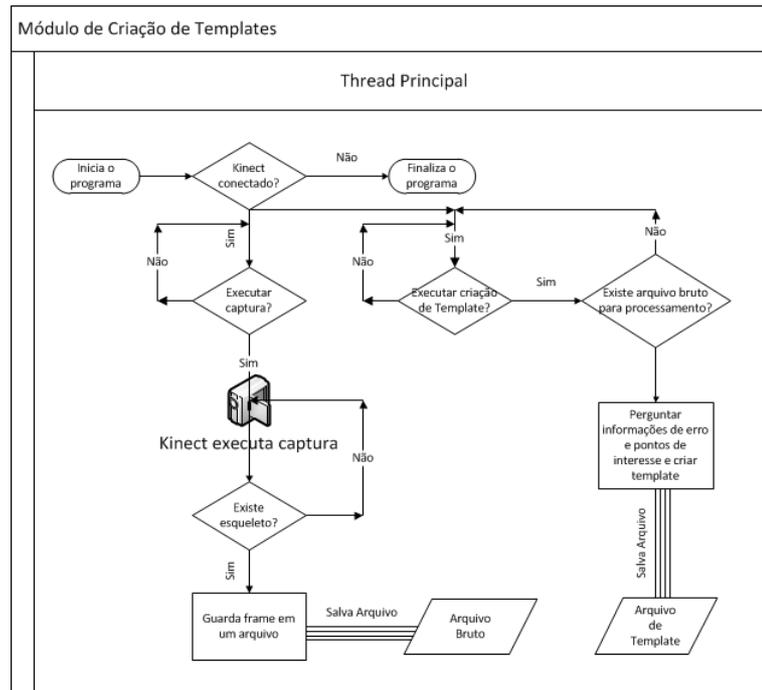


Figura 19 – Arquitetura do Módulo de Criação de *Templates*. Fonte: O Autor

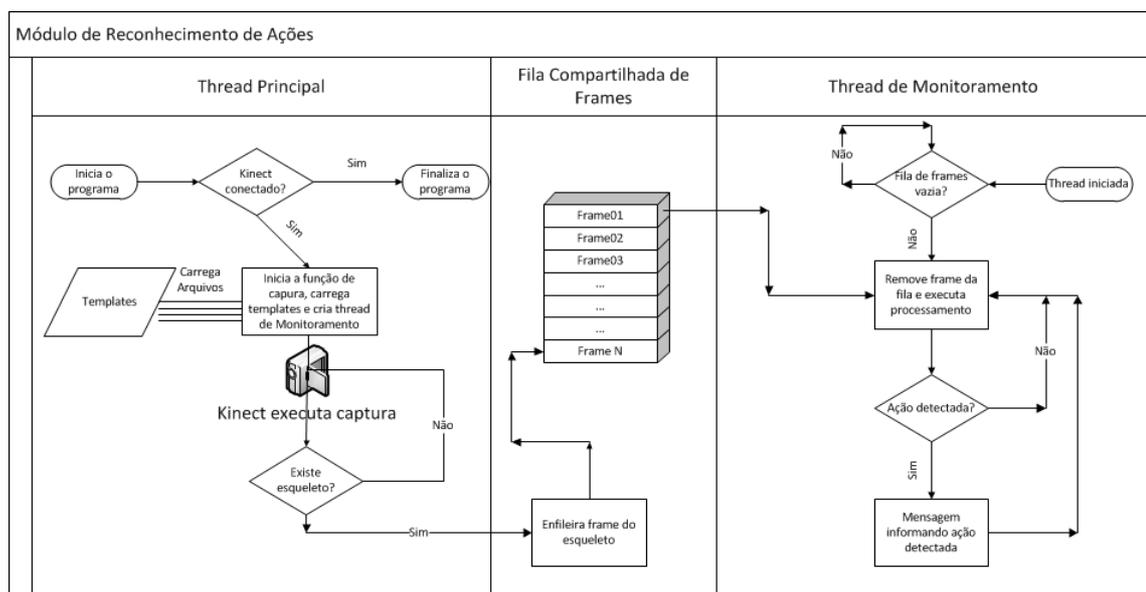


Figura 20 – Arquitetura do Módulo de Monitoramento e Reconhecimento de Ações. Fonte: O Autor

O módulo de criação precisou de apenas um caminho de execução, pelo qual capturou as informações do esqueleto, processou essas informações e gerou os *templates*. Já o módulo de reconhecimento necessitou de dois caminhos, pois era necessário que o processamento fosse feito ao mesmo que haviam captura de novos *frames* do esqueleto.

4.3 Solução Desenvolvida

Com o problema, os requisitos e a arquitetura definida, realizou-se uma análise, a qual deveria ser o sistema operacional usado para desenvolver o sistema e com base nisso seria a linguagem para codificação.

Como o *Kinect* é um *hardware* desenvolvido e mantido pela *Microsoft*, a priori a escolha óbvia do sistema operacional seria o *Windows*. Todavia existem bibliotecas *open source* do *Kinect* para Mac OS X e para outros sistemas Unix, então, foi realizado um estudo de algumas dessas bibliotecas, como a *libfreenect* e a *openkinect*. Com o estudo, foram encontradas bibliotecas que não recebem uma boa manutenção e com documentação pobre. Diferentemente das outras, a biblioteca oficial da *Microsoft* possui uma ótima documentação, além dos exemplos incluídos na instalação do *Kinect for Windows SDK (Software Developer Kit)*, que ajudam o usuário iniciante a se familiarizar com a biblioteca. Diante do cenário analisado, foi decidido que o sistema operacional usado seria o *Windows*, pois, apesar do problema óbvio de portabilidade, o mesmo apresenta melhor suporte para o desenvolvimento usando o *Kinect*.

Com a escolha do *Windows* como sistema operacional, para que pudesse ser feito o uso do *Kinect for Windows SDK* mantido pela própria *Microsoft*, as opções de linguagem ficaram restritas ao C# ou C++. Entre as linguagens disponíveis foi escolhido o C++, porque não existe diferenciação das funcionalidades disponíveis para cada uma das linguagens e a mesma possui um maior desempenho que o C# (referência). Como um dos requisitos não funcionais do módulo de reconhecimento de ações é o desempenho, é importante que o mesmo seja desenvolvido em uma linguagem que produza programas mais rápidos, que é o exemplo do C++.

O segundo passo foi a preparação do ambiente de desenvolvimento. Os programas necessários são:

- *Kinect for Windows SDK*: fornece o *driver* para o correto funcionamento do sensor e a biblioteca para desenvolvimento de aplicações;
- *Microsoft Visual Studio*: IDE (*Integrated Development Kit*) da *Microsoft* que possibilita uma configuração mais fácil para o desenvolvimento com *Kinect*;

- *GitHub*: ferramenta para controle de versões, instala também um *shell* que facilita na hora de submeter as modificações no repositório;
- *Sublime Text 2*: editor de texto usado para a codificação, apesar do código ter sido debugado e compilado no *Visual Studio*, toda a escrita do mesmo foi realizada utilizando este editor.

Com tudo preparado, a codificação foi realizada. O processo de desenvolvimento e codificação será descrito com mais detalhes na próxima seção.

4.3.1 Processo de Desenvolvimento

Tendo requisitos, arquitetura e linguagem definidas e ambiente preparado para a codificação foi dado início ao desenvolvimento.

O **primeiro passo** dado foi pensar em qual metodologia de desenvolvimento seria usada, mas essa ideia foi rapidamente resolvida, pois como a codificação seria realizada por somente um programador, o Autor deste trabalho, o uso de metodologias só iria acabar pesando o desenvolvimento com a produção de documentação e execução de etapas desnecessárias. Para resolver esse problema foi cogitado o uso de metodologias ágeis, mas mesmo sendo mais leves com menos documentação e mais programação a forma como o desenvolvimento seria realizada, com somente uma pessoa, não justificaria o uso dessa metodologia.

O **segundo passo** consistiu-se basicamente da familiarização do programador com a API fornecida pela *Microsoft* para o *Kinect*. Os SDK do sensor disponibiliza alguns programas de exemplo, como sistemas que fazem o rastreamento do esqueleto e o mostram na tela em 2D, sistemas de reconstrução 3D entre outros. O programador já possuía algum conhecimento da API do *Kinect* em C++ e por isso extrair informações e conhecimentos úteis desses exemplos foi mais rápido.

Com a conclusão do segundo passo o **terceiro** e mais importante foi dado, o início real da codificação, onde linhas de código e programas começaram a ser escritos. Dentro da etapa de codificação foram executadas as seguintes tarefas, começando pelo Módulo de Criação de *Templates*:

1. Criação da janela onde ficaria sendo mostrada a imagem, em RGB, do que o sensor (*Kinect*) estava capturando. As bibliotecas escolhidas para o desenvolvimento desta etapa foram a GLUT e a OpenGL, pela familiaridade do programador com a mesma;
2. Realização da captura das juntas do esqueleto, mostrando as mesmas na linha de comando;

3. Início da criação dos arquivos iniciais que continham as informações cruas das juntas, salvando os mesmos em arquivos .txt para o processamento;
4. Com os arquivos contendo todas as juntas, foi realizada a etapa de processamento, que foi realizado através da normalização dos valores de todas as juntas em relação a *Shoulder Center*, da indicação de quais seriam as juntas de interesse (juntas a serem comparadas pelo algoritmo) e definição do erro inicial para começar os testes, e
5. Os dados gerados pelo processamento foram salvos em arquivos, assim foram criados os *templates*.

Para a codificação do Módulo de Reconhecimento de Ações, foi aproveitado o código feito até ponto 2 (dois), pois a partir daí foi dado início ao desenvolvimento do algoritmo de reconhecimento:

1. Com o sistema mostrando a imagem e recolhendo as informações das juntas do esqueleto o primeiro passo foi carregar os *templates* gerados pelo módulo de criação. Foi criada uma estrutura de dados específica para manipulação desses arquivos;
2. Para que a comparação, do esqueleto atual com o modelo, pudesse ser feita ao mesmo tempo que o programa capturava as informações do *Kinect*, foi necessária a criação de uma *thread* que faria todo o processamento em paralelo;
3. Os dados recolhidos pelo sensor que seriam processados pela *thread* foram armazenados em uma fila, para que os primeiros dados a serem inseridos fossem os primeiros a serem lidos e tratados. Como a fila seria lida tanto pela *thread* principal do programa como a de reconhecimento, foi necessária a criação de uma região crítica em torno dessa variável, para que não houvesse dois processos fazendo uso da fila ao mesmo tempo. Isso foi implementado usando um *mutex*, e
4. O último passo e mais crítico para a codificação foi a implementação do algoritmo de reconhecimento, essa etapa foi tão importante que antes de ser inserido no sistema de reconhecimento o algoritmo foi testado de forma *offline*, como pode ser visto na Figura 21 e 22. A implementação do algoritmo foi feita da seguinte forma:
 - a) A captura é iniciada, assim o primeiro *frame* de um esqueleto é enfileirado;
 - b) O *frame* do esqueleto enfileirado é removido da fila;
 - c) O esqueleto atual é comparado com o primeiro *frame* de cada modelo, mas somente as juntas de interesse são comparadas para evitar comparações desnecessárias. O erro é calculado, para cada um dos modelos, e guardado;

- d) A comparação continua, somando os erros, até que tenham sido comparados um número de *frames* igual ao tamanho do modelo, e
- e) Quando um certo modelo alcança o seu número máximo de *frames* ele verifica se o erro está dentro do limite aceitável (menor ou igual ao erro máximo do modelo), caso sim um evento foi detectado, caso não o erro é zerado e a comparação volta para o início do modelo, reiniciando o ciclo.

O código foi desenvolvido totalmente usando o paradigma estruturado, pois houve um momento da codificação que o programa foi escrito novamente usando Orientação a Objetos, mas isso acabou causando um perda de performance e como isso é muito importante para o sistema o desenvolvimento voltou para o paradigma estruturado.

O **quarto passo** foi constituído pelos testes, validação e ajustes do erro máximo. A estratégia adota para a realização dos testes foi realizar os mesmos de acordo com o desenvolvimento, assim sempre que algo ficava pronto aquilo era testado, quaisquer erros eram corrigidos e caso novos erros fossem encontrados o mesmos eram corrigidos, assim era feito até que houvesse uma versão funcional da *feature* que desejava ser desenvolvida. Alguns exemplos de testes foram os realizados no algoritmo para verificar se o mesmo funcionava como esperado, nos arquivos dos *templates* para saber se os mesmos estavam sendo escritos e carregados de forma correta e por último na *thread* para saber se o processamento estava acontecendo de forma paralela sem nenhum bloqueio.

Com os módulos implementados já testados foi feita a validação primeiramente através da criação dos *templates* da marcha normal e hemiplégica, do movimento dos braços e do movimento de abaixar. A partir dos modelos criados os mesmos foram testados no Módulo de Reconhecimento de Ações e após alguns ajustes finos nos erros máximos de cada modelo o sistema conseguia fazer a diferenciação entre uma pessoa com marcha normal e uma pessoa com marcha hemiplégica, assim como identificar o movimento do braço direito/esquerdo ou até mesmo o movimento dos dois braços ao mesmo tempo.

A limitação da altura foi identificada na fase de validação quando uma pessoa de aproximadamente 1,58m (um vírgula cinquenta e oito metros) tentou fazer uso do sistema com modelos criados por uma pessoa com 1,77m (um vírgula setenta e sete metros) e não obteve nenhuma resposta coerente ou até mesmo não identificava nada.

A característica do sistema, de ser capaz de reconhecer duas ações ao mesmo tempo, também só foi percebida na fase de validação, pois quando o algoritmo foi pensado essa possibilidade não tinha sido prevista. Essa foi um grata surpresa, pois isso também ajuda na diminuição da carga de dados a serem processados e aumenta capacidade de inferência do sistema.

4.3.2 Caso de uso: Pacientes Hemiplégicos Vítimas de Acidente Vascular Cerebral

A versão final do Sistema de Reconhecimento de Ações foi testada com um paciente hemiplégico. A principal característica que diferencia o hemiplégico de uma pessoa comum é a marcha, como dito anteriormente por (OTTOBONI; FONTES; FUKUJIMA, 2002), uma pessoa normal dá entre 90 (noventa) e 120 (cento e vinte) passos por minuto. Observou-se, durante os testes do sistema, que o paciente hemiplégico dá uma passada em aproximadamente 2,5 (dois vírgula cinco) segundos, o que resulta em uma média de 48 (quarenta e oito) passos por minuto.

A partir das observações realizadas durante os testes, foi feito o recolhimento de um *template* de 2,5 (dois vírgula cinco) segundos da marcha hemiplégica e um *template* de menor duração para a marcha normal. Além dos modelos para marcha, foi recolhido um para o movimento do braço esquerdo, braço direito e da ação de abaixar.

Depois de recolhidos os *templates*, foram recolhidas também algumas cenas. Uma cena continha movimentos dos *templates* e outras não continham nenhuma ação específica. Foi então feito um teste *offline* do algoritmo, para poder validar o mesmo, onde os modelos eram comparados as cenas usando o algoritmo proposto. Se o modelo obtivesse um erro muito grande em uma cena com o seu movimento contido o algoritmo iria falhar, caso contrário a hipótese seria confirmada.

```
NotAnAction1:
MoverBracoDireito = 150.368
MoverBracoEsquerdo = 123.761
MarchaHemiplegica = 219.118
MarchaNormal = 251.366

NotAnAction2:
MoverBracoDireito = 151.212
MoverBracoEsquerdo = 158.474
MarchaHemiplegica = 256.473
MarchaNormal = 279.446

NotAnAction3:
MoverBracoDireito = 105.172
MoverBracoEsquerdo = 143.874
MarchaHemiplegica = 246.822
MarchaNormal = 289.692
```

Figura 21 – *Templates* sendo comparados a cena sem ação nenhuma. Fonte: O Autor

```
MoverBracoDireito1:
MoverBracoDireito = 10.3377
MoverBracoEsquerdo = 98.4621
MarchaHemiplegica = 233.821
MarchaNormal = 272.926

MoverBracoDireito2:
MoverBracoDireito = 14.3536
MoverBracoEsquerdo = 98.6585
MarchaHemiplegica = 236.78
MarchaNormal = 275.903

MoverBracoDireito3:
MoverBracoDireito = 15.2507
MoverBracoEsquerdo = 97.8516
MarchaHemiplegica = 238.928
MarchaNormal = 278.487
```

Figura 22 – *Templates* sendo comparados a cena com movimentos do braço direito. Fonte: O Autor

As Figura 21 e 22 mostram que quando temos uma cena com a ação do *template* contida nela, obtemos um erro muito menor se comparado a outras cenas, o que confirma a hipótese, validando o algoritmo.

Com o algoritmo validado, os *templates* foram então processados e começaram a ser testados no Módulo de Monitoramento e Reconhecimento de Ações. Antes de fazer o uso do sistema, foi necessário a realização de alguns ajustes durante a fase de testes aos erros mínimos de cada *template*, para que o reconhecimento fosse feito de forma correta. Os resultados obtidos com a execução do sistema podem ser vistos da Figura 23 a 28, ao final do capítulo.

As Figuras mostram que o sistema funciona muito bem para o caso de uso aplicado. Após executar os testes foi percebido que o sistema é capaz de:

- Diferenciar a marcha normal da marcha hemiplégica;
- Reconhecer o movimento de abaixar, e
- Reconhecer o movimento do braço esquerdo e do braço direito, tanto quando as ações eram executadas separadamente quando eram executadas ao mesmo tempo.

A tabela 2 e 3 mostram informações mais detalhadas dos modelos que foram utilizados para o teste do sistema de reconhecimento de ações.

Nome do <i>Template</i>	Pontos de Interesse
Abaixar	<i>Hip Center, Head</i>
MarchaHemiplegicaDireitaEsquerda	<i>Hip (Center, Left, Right), Spine, Head</i>
MarchaHemiplegicaEsquerdaDireita	<i>Hip (Center, Left, Right), Spine, Head</i>
MarchaNormalDireitaEsquerda	<i>Hip (Center, Left, Right), Spine, Head</i>
MarchaNormalEsquerdaDireita	<i>Hip (Center, Left, Right), Spine, Head</i>
MoverBracoDireito	<i>Shoulder, Elbow, Wrist, Hand (All Right)</i>
MoverBracoEsquerdo	<i>Shoulder, Elbow, Wrist, Hand (All Left)</i>

Tabela 2 – *Templates* e juntas de interesse. Fonte: O Autor

Vale ressaltar que o uso do algoritmo proposto trouxe uma grande vantagem ao sistema, que foi a capacidade de reconhecer mais de uma ação ao mesmo tempo. Essa característica possibilita que o sistema consiga fazer mais inferências com um menor número de *templates* o que causa também uma diminuição na carga de processamento. Por exemplo, caso seja necessário reconhecer 3 ações, mover braço esquerdo, mover braço direito e mover os dois braços, não será necessário recolher 3 *templates* diferentes, pois quando houver a detecção da ação do braço direito e do braço esquerdo ao mesmo tempo, pode-se inferir que existe o movimento dos dois braços.

Nome do <i>Template</i>	Erro máximo
Abaixar	3,5
MarchaHemiplegicaDireitaEsquerda	8,0
MarchaHemiplegicaEsquerdaDireita	8,0
MarchaNormalDireitaEsquerda	5,0
MarchaNormalEsquerdaDireita	4,0
MoverBracoDireito	30,0
MoverBracoEsquerdo	30,0

Tabela 3 – *Templates* e erro máximo. Fonte: O Autor

O sistema realiza o reconhecimento de ações, mas pode ser transformado em um sistema de reconhecimento de acidentes seguindo alguns pequenos passos:

- Deve ser criada uma nova *thread* onde será feito o monitoramento dos eventos detectados;
- Os eventos detectados devem ser enfileirados para que possam ser processados pela função responsável;
- As regras de inferência para uma ação se considerada acidente devem ser criadas, e
- O sistema deve ser conectado a um módulo de alerta para que a ocorrência de um acidente seja transmitido para um responsável pelo monitoramento.

4.3.3 Limitações da Solução

Apesar do resultados obtidos, o sistema possui algumas limitações que valem ser ressaltadas:

- O sistema não pode ser facilmente portado para outros sistemas operacionais por ter sido desenvolvido usando uma biblioteca exclusiva para *Windows*;
- O sistema possui um pequeno atraso no reconhecimento, pois a comparação do *template* com a ação sendo executada em tempo real é realizada a cada x segundos, onde x é a duração do *template* a ser reconhecido. Dessa forma, a ação só é reconhecida ao fim da execução da mesma, ou alguns milésimos de segundos depois, como foi observado na execução do sistema;
- O sistema possui certa limitação no reconhecimento. Devido ao reconhecimento ser feito usando o esqueleto, no momento em que o *template* é gerado, os pontos das juntas estão distantes em relação ao tamanho da pessoa, por isso, se o sistema tiver somente o *template* de um pessoa de 1,70m (um metro e sessenta centímetros), ele só reconhecerá pessoas que tenham aproximadamente essa altura, esse problema pode

ser contornado com a obtenção de *templates* de todas as pessoas que farão uso do sistema, e

- Mesmo tendo sido desenvolvido em C++, o sistema possui uma carga alta de processamento, pois está constantemente comparando as ações sendo executadas com os *templates* disponíveis para reconhecimento.

Como explicado nos tópicos acima as limitações do sistema não impossibilitam seu uso, pois as mesmas podem ser contornadas facilmente.

4.3.4 Requisitos de Sistema

Como dito na seção sobre limitações, o sistema possui uma alta carga de processamento, o que gera alto consumo de memória RAM e processador. Por isso, para que o sistema funcione bem e da forma como foi planejado e testado, serão definidos como requisitos mínimos de sistema a configuração do computador onde foi desenvolvido.

Os requisitos mínimos de sistema serão:

Processador: 2.3 GHz Intel Core i7.

Memória RAM: 8GB 1600 Mhz DDR3.

Placa Gráfica: NVIDIA GeForce GT 650M 1024 MB.

Programas: *Kinect for Windows SDK*, inclui o *driver* para o funcionamento correto do *Kinect* no computador.

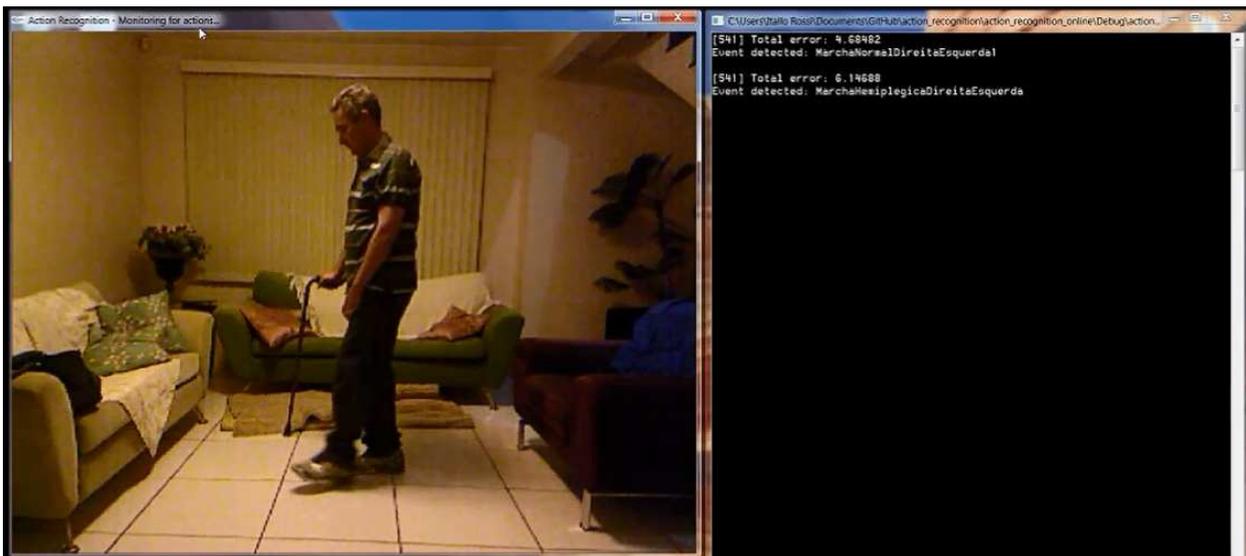


Figura 23 – Sistema realizando o reconhecimento da Marcha Hemiplégica. Fonte: O Autor

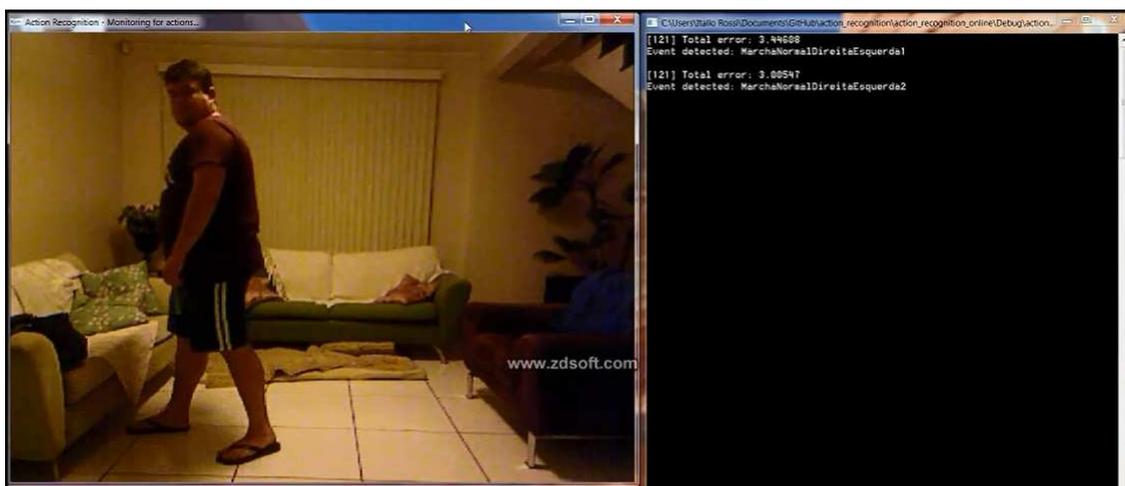


Figura 24 – Sistema realizando o reconhecimento da Marcha Normal. Fonte: O Autor

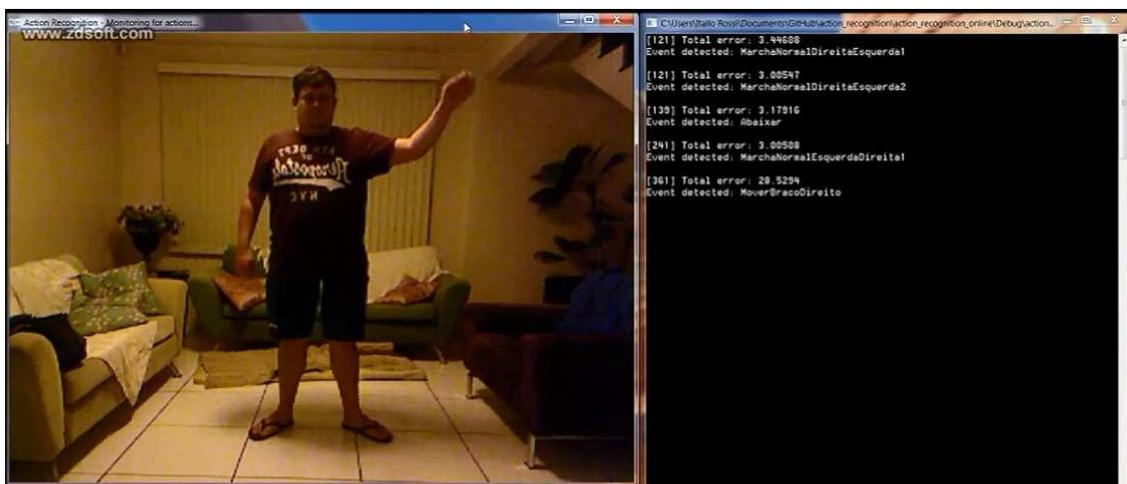


Figura 25 – Sistema reconhecendo a ação de Mover Braço Direito. Fonte: O Autor

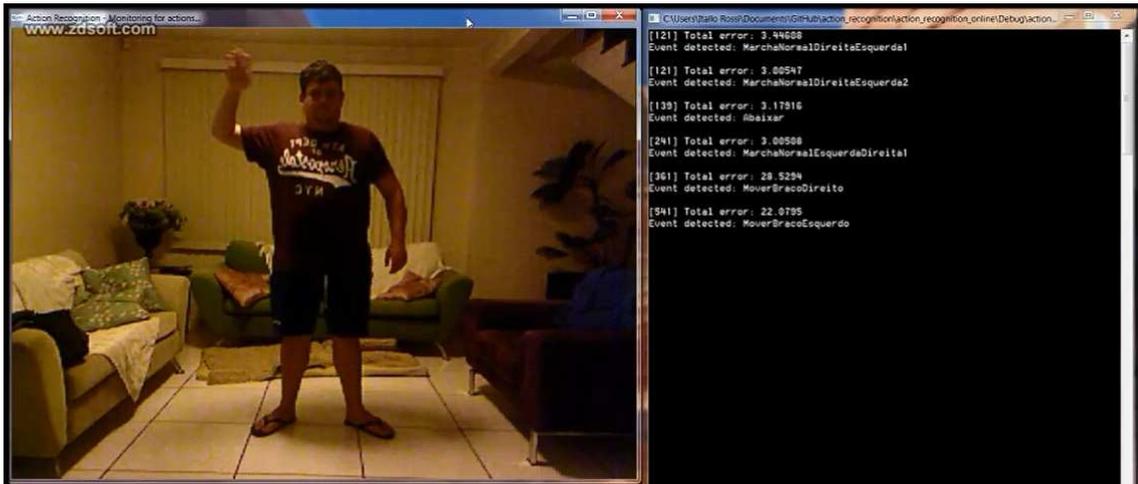


Figura 26 – Sistema reconhecendo a ação de Mover Braço Esquerdo. Fonte: O Autor

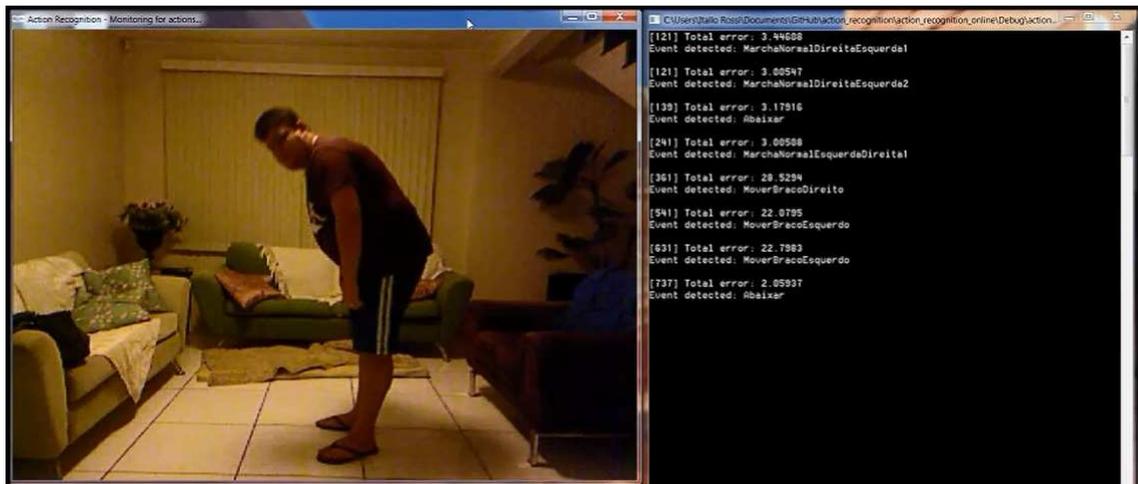


Figura 27 – Sistema reconhecendo a ação de Abaixar. Fonte: O Autor



Figura 28 – Marcha Hemiplégica e Mover Braço Esquerdo. Fonte: O Autor

5 Conclusão

O resultado obtido com o desenvolvimento deste trabalho foi um sistema de reconhecimento de ações que é capaz de reconhecer inúmeras ações. Uma vez que exista o *template* para certa ação, a mesma poderá ser reconhecida. A solução proposta conseguiu ser aplicada com sucesso no desenvolvimento do sistema gerando o resultado esperado.

O paciente hemiplégico possui uma marcha diferente da pessoa normal. Essa característica conseguiu ser utilizada de maneira bem sucedida no caso de uso proposto, para fazer com que o sistema seja capaz de diferenciar uma pessoa comum de uma pessoa hemiplégica a partir da sua marcha. O caso de uso também fez o teste do movimentos do braços, pois como o paciente hemiplégico pode também apresentar dificuldades de expressão do pensamento e na fala, esses movimentos podem ser usados para criações de padrões de pedido de ajuda.

Para trabalhos futuros, tem-se como proposta aperfeiçoar o processamento dos *templates*, para tentar reduzir para praticamente zero o tempo de atraso no reconhecimento, mesmo que o atraso existente já seja baixo. Outro trabalho futuro, é a integração desse sistema com um sistema de comunicação que poderá informar o acontecimento de certa ação via SMS, e-mail e/ou ligação. Por último, outro aperfeiçoamento que poderia ser feito usando o sistema de reconhecimento de ações deste trabalho, é a criação de um sistema de detecção de acidentes que poderia, inclusive, usar as diretrizes propostas no capítulo anterior.

Referências

- AARTS, E.; WICHERT, R. Ambient intelligence. In: BULLINGER, H.-J. (Ed.). *Technology Guide*. Springer Berlin Heidelberg, 2009. p. 244–249. ISBN 978-3-540-88545-0. Disponível em: <http://dx.doi.org/10.1007/978-3-540-88546-7_47>. Citado na página 27.
- AGHAJAN, H. et al. Distributed vision-based accident management for assisted living. In: *In Int. Conf. on Smart homes and health Telematics (ICOST*. [S.l.: s.n.], 2007. Citado 7 vezes nas páginas 32, 33, 34, 35, 36, 37 e 44.
- ANDRÉ, C. *Manual de AVC*. Revinter, 2006. ISBN 9788573099706. Disponível em: <<http://books.google.com.br/books?id=5CKPPwAACAAJ>>. Citado na página 23.
- BRAMBERGER, M. et al. Distributed embedded smart cameras for surveillance applications. *Computer*, 2006. IEEE Computer Society Press, Los Alamitos, CA, USA, v. 39, n. 2, p. 68–75, fev. 2006. ISSN 0018-9162. Disponível em: <<http://dx.doi.org/10.1109/MC.2006.55>>. Citado 2 vezes nas páginas 28 e 29.
- BRASIL, L. M.; AZEVEDO, F. M. de; BARRETO, J. M. A hybrid expert system for the diagnosis of epileptic crisis. *Artificial intelligence in medicine*, 2001. Elsevier Science Publishing, v. 21, n. 1, p. 227–233, 01 2001. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0933365700000907?showall=true>>. Citado na página 28.
- BURROWS, M.; ABADI, M.; NEEDHAM, R. A logic of authentication. *ACM TRANSACTIONS ON COMPUTER SYSTEMS*, 1990. v. 8, p. 18–36, 1990. Citado na página 23.
- COELHO, V. Creating ambient intelligence: Applications and design guidelines for the home. 2012. 2012. Citado na página 28.
- DAVIES, P. *Passos a seguir: um manual para o tratamento da hemiplegia no adulto*. Manole, 1996. ISBN 9788520404096. Disponível em: <<http://books.google.com.br/books?id=LI0TGQAACAAJ>>. Citado na página 25.
- DEMENTHON, D. Spatio-temporal segmentation of video by hierarchical mean shift analysis. In: *Center for Automat. Res., U. of Md, College Park*. [S.l.: s.n.], 2002. Citado na página 36.
- GAUTAMA, T.; HULLE, M. M. V. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE TRANS. NEURAL NETWORKS*, 2002. v. 13, n. 5, p. 2002, 2002. Citado na página 35.
- GILLESPIE, L. D. et al. Interventions for preventing falls in elderly people. *Cochrane database of systematic reviews (Online)*, 2001. n. 3, 2001. ISSN 1469-493X. Disponível em: <<http://dx.doi.org/10.1002/14651858.cd000340>>. Citado na página 33.

- GK, R. Biomecânica da marcha. In: *Fisioterapia em ortopedia e reumatologia*. [S.l.: s.n.], 1987. p. 19–31. Citado 2 vezes nas páginas 25 e 26.
- GORELICK, L. et al. Actions as space-time shapes. In: *In ICCV*. [S.l.: s.n.], 2005. p. 1395–1402. Citado na página 38.
- GRE. *Acidente Vascular Cerebral*. 2002. Disponível em: <www.gre.com.br>. Citado na página 24.
- GUPTA, A.; DAVIS, L. S. Objects in action: An approach for combining action understanding and object perception. In: *CVPR*. [S.l.]: IEEE Computer Society, 2007. Citado na página 31.
- HEANEY, M. Developing a tele care strategy in northern ireland. In: *Proceedings of the Workshop on The role of e-Health and Assistive Technologies in Healthcare. UK*. [S.l.: s.n.], 2006. Citado na página 34.
- HSU, R.-L.; ABDEL-MOTTALEB, M.; JAIN, A. Face detection in color images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2002. v. 24, n. 5, p. 696–706, 2002. ISSN 0162-8828. Citado na página 29.
- KE, Y.; SUKTHANKAR, R.; HEBERT, M. Efficient temporal mean shift for activity recognition in video. In: *NIPS Workshop on Activity Recognition and Discovery*. [S.l.: s.n.], 2005. Citado 2 vezes nas páginas 36 e 37.
- KE, Y.; SUKTHANKAR, R.; HEBERT, M. Efficient visual event detection using volumetric features. In: *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*. Washington, DC, USA: IEEE Computer Society, 2005. (ICCV '05), p. 166–173. ISBN 0-7695-2334-X-01. Disponível em: <<http://dx.doi.org/10.1109/ICCV.2005.85>>. Citado 3 vezes nas páginas 35, 38 e 44.
- KE, Y.; SUKTHANKAR, R.; HEBERT, M. Event detection in crowded videos. In: *IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2007. Citado na página 46.
- KE, Y.; SUKTHANKAR, R.; HEBERT, M. Spatio-temporal shape and flow correlation for action recognition. In: *In 7th Int. Workshop on Visual Surveillance*. [S.l.: s.n.], 2007. Citado 6 vezes nas páginas 38, 39, 40, 44, 46 e 47.
- KESHAVARZ, A. Distributed vision-based reasoning for smart home care. In: *ACM SenSys Workshop on Distributed Smart Cameras (DSC'06)*. [S.l.: s.n.], 2006. Citado 4 vezes nas páginas 28, 29, 30 e 44.
- KLEINBERGER, T. et al. Ambient intelligence in assisted living: Enable elderly people to handle future interfaces. In: *Proceedings of the 4th International Conference on Universal Access in Human-computer Interaction: Ambient Interaction*. Berlin, Heidelberg: Springer-Verlag, 2007. (UAHCI'07), p. 103–112. ISBN 978-3-540-73280-8. Disponível em: <<http://dl.acm.org/citation.cfm?id=1763296.1763308>>. Citado 3 vezes nas páginas 23, 28 e 44.
- KOSMOPOULOS, D. I. et al. Fusion of color and depth video for human behavior recognition in an assistive environment. In: STREITZ, N.; STEPHANIDIS, C. (Ed.). *Distributed, Ambient, and Pervasive Interactions: Proc. of the First International*

- Conference (DAPI 2013), Las Vegas, NV, USA*. Berlin: Springer, 2013, (Lecture Notes in Computer Science, 8028). p. 42–51. ISBN 978-3-642-39783-7. Citado 3 vezes nas páginas 41, 42 e 44.
- LAPTEV, I. On space-time interest points. *International Journal of Computer Vision*, 2005. Kluwer Academic Publishers, v. 64, n. 2-3, p. 107–123, 2005. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1007/s11263-005-1838-7>>. Citado na página 41.
- LO, B. P. L.; WANG, J. L.; YANG, G. zhong. From imaging networks to behavior profiling: Ubiquitous sensing for managed homecare of the elderly. In: *Adjunct Proceedings of the 3rd International Conference on Pervasive Computing*. [S.l.: s.n.], 2005. Citado na página 29.
- MOORE, K. S. S. Observation and analysis of hemiplegic gait: Swing phase. *Australian Phys*, 39:258-267, 1994. 1994. Citado na página 26.
- MSDN, M. *Kinect Sensor*. 2012. Disponível em: <<http://msdn.microsoft.com/en-us/library/hh438998.aspx>>. Citado na página 45.
- NETTO, M. *Gerontologia: a velhice e o envelhecimento em visão globalizada*. Atheneu, 1999. ISBN 9788573791099. Disponível em: <<http://books.google.com.br/books?id=7vyoAAAACAAJ>>. Citado na página 23.
- NITRINI, R.; BACHESCHI, L. *A neurologia que todo que todo médico deve saber*. Atheneu, 2003. Disponível em: <<http://books.google.com.br/books?id=5vtDHQAACAAJ>>. Citado na página 23.
- NORRIS, J. *Markov Chains*. Cambridge University Press, 1998. (Cambridge Series in Statistical and Probabilistic Mathematics, no. 2008). ISBN 9780521633963. Disponível em: <<http://books.google.it/books?id=qM65VRmOJZAC>>. Citado na página 32.
- OTTOBONI, C.; FONTES, S. V.; FUKUJIMA, M. M. Estudo comparativo entre a marcha normal e a de pacientes hemiparéticos por acidente vascular encefálico: Aspectos biômecânicos. *Revista Neurociências* 10(1), 2002. p. 10–16, 2002. Citado 4 vezes nas páginas 25, 26, 43 e 56.
- PERLINI, N. *Cuidar de pessoa incapacitada por acidente vascular cerebral no domicílio: o fazer do cuidador familiar*. Universidade de Sao Paulo. Escola de Enfermagem, Sao Paulo, 2000. Disponível em: <<http://books.google.com.br/books?id=FGMBHQAACAAJ>>. Citado 3 vezes nas páginas 23, 24 e 43.
- PEURSUM, P.; WEST, G.; VENKATESH, S. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In: *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*. Washington, DC, USA: IEEE Computer Society, 2005. (ICCV '05), p. 82–89. ISBN 0-7695-2334-X-01. Disponível em: <<http://dx.doi.org/10.1109/ICCV.2005.57>>. Citado na página 31.
- PINZ, A. Object categorization. *Found. Trends. Comput. Graph. Vis.*, 2005. Now Publishers Inc., Hanover, MA, USA, v. 1, n. 4, p. 255–353, dez. 2005. ISSN 1572-2740. Disponível em: <<http://dx.doi.org/10.1561/06000000003>>. Citado na página 31.

- REMAGNINO, P.; FORESTI, G.; ELLIS, T. *Ambient Intelligence: A Novel Paradigm*. Springer, 2005. ISBN 9780387229904. Disponível em: <<http://books.google.com.br/books?id=ZkWlivh4O6QC>>. Citado na página 27.
- RODRIGUES, S. de S.; ROCHA, A. F. D. Palmilha sensorizada para pés diabéticos. 2011. 2011. Citado na página 28.
- SCHULDT, C.; LAPTEV, I.; CAPUTO, B. Recognizing human actions: A local svm approach. In: *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*. Washington, DC, USA: IEEE Computer Society, 2004. (ICPR '04), p. 32–36. ISBN 0-7695-2128-2. Disponível em: <<http://dx.doi.org/10.1109/ICPR.2004.747>>. Citado na página 39.
- SHAW, F. E. et al. Multifactorial intervention after a fall in older people with cognitive impairment and dementia presenting to the accident and emergency department: randomised controlled trial. *BMJ*, 2003. v. 326, n. 7380, p. 73, 1 2003. Citado na página 33.
- SHOTTON, J. et al. Real-time human pose recognition in parts from single depth images. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2011. (CVPR '11), p. 1297–1304. ISBN 978-1-4577-0394-2. Disponível em: <<http://dx.doi.org/10.1109/CVPR.2011.5995316>>. Citado na página 41.
- SILVA, I. da. *As relacoes de poder entre o adulto dependente e a mulher-cuidadora*. Pontificia Universidade Catolica de Sao Paulo, Sao Paulo, 1995. Disponível em: <<http://books.google.com.br/books?id=hME6twAACAAJ>>. Citado na página 23.
- SKILECK, C. Psychological aspects of stroke. In: *Handbook of Clinical Psychology of Elderly People*. [S.l.: s.n.], 1996. Citado na página 43.
- SMELTZER, S.; BARE, B. *Brunner & Suddarth tratado de enfermagem médico-cirúrgica*. Guanabara Koogan, 2005. ISBN 9788527710572. Disponível em: <<http://books.google.com.br/books?id=NCIrpwAACAAJ>>. Citado na página 43.
- UFSC. *Anatomia do Pé*. 2004. Disponível em: <<http://www.hu.ufsc.br/~grumad/anatomia.htm>>. Citado 2 vezes nas páginas 27 e 43.
- WANG, J. et al. Image and video segmentation by anisotropic kernel mean shift. In: PAJDLA, T.; MATAS, J. (Ed.). *Computer Vision - ECCV 2004*. Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3022). p. 238–249. ISBN 978-3-540-21983-5. Disponível em: <http://dx.doi.org/10.1007/978-3-540-24671-8_19>. Citado na página 35.
- WU, C. H.: Layered and collaborative gesture analysis in multi-camera networks. In: *In: ICASSP*. [S.l.: s.n.], 2007. Citado na página 34.
- WU, C.; AGHAJAN, H. User-Centric Environment Discovery With Camera Networks in Smart Homes. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 2010. 2010. ISSN 1083-4427. Disponível em:

<http://dx.doi.org/10.1109/tsmca.2010.2073701>>. Citado 4 vezes nas páginas 31, 32, 33 e 44.

ZHAO, Y. et al. Combing rgb and depth map features for human activity recognition. In: *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*. [S.l.: s.n.], 2012. p. 1–4. Citado 3 vezes nas páginas 41, 42 e 44.