Felippe Mendonça de Queiroz

Sistema de Localização de Gestos Utilizando um Sistema Multicâmeras

Brasil

2019

Felippe Mendonça de Queiroz

Sistema de Localização de Gestos Utilizando um Sistema Multicâmeras

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Universidade Federal do Espírito Santo

Programa de Pós-Graduação em Engenharia Elétrica

Orientadora Raquel Frizera Vassallo

Brasil

2019

Felippe Mendonça de Queiroz

Sistema de Localização de Gestos Utilizando um Sistema Multicâmeras

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Trabalho aprovado. Brasil, 19 de Dezembro de 2019:

Profa. Dra. Raquel Frizera Vassallo
Orientadora

Prof. Dr. Eduardo Oliveira Freire
Universidade Federal de Sergipe
Examinador Externo

Prof. Dr. Jugurta Rosa Montalvão Filho
Universidade Federal de Sergipe
Examinador Externo

Brasil
2019

Agradecimentos

Sem dúvidas, este trabalho não é só meu. Eu achei que tinha sido difícil escrever o referencial teórico, as conclusões, mas difícil mesmo é eu lembrar de todos (mentira, eu tive é que parar várias vezes para enxugar as lágrimas) que colaboraram de alguma forma para que eu concluísse mais esta etapa da minha vida. Essa etapa não inclui apenas a conclusão do mestrado, mas sim de toda minha trajetória na Ufes, desde o início da graduação.

Primeiramente, agradeço a minha família! Vocês foram fundamentais para eu conseguir concluir esta jornada. Sempre entendendo meus momentos de ausência, quando eu estava enfiado no laboratório até tarde, e no outro dia acordava e voltava pra ele. Nunca faltou apoio! Mãe, Pai, Nana, vocês são responsáveis por muito do que eu sou hoje. Não tenho palavras pra agradecer. Amo vocês!

Toda essa jornada teve início lá no começo da graduação, quando eu comecei a trabalhar em um laboratório (o melhor de todos) que virou minha segunda casa (ou primeira?). Raquel e Cabelo, muito obrigado por me proporcionarem todos esses anos, não só aprendizado técnico. Vocês fazem parte da família que eu construí esses anos no laboratório. Todas as festas, passeios, programas de índio, eternas discussões com o Hair (sinto falta disso!), a ida pra Bristol, tudo isso contribuiu para quem eu sou hoje, tanto como pessoa, quanto como profissional. Obrigado por tudo!

Agradeço aos amigos/irmãos da velha guarda, quando eu era o "menino novo, tem muito que aprender", alvo de todas as brincadeiras! Flávio, Frodo e Patrão, vocês são uns porcarias, mas eu gosto muito de vocês! Cresci muito convivendo com vocês! Não posso esquecer da irmã mais velha, a Mari. Você esteve presente em boa parte dessa trajetória. Aprendi muito com você, e tenho uma enorme admiração por ti, não só profissionalmente. Muito obrigado por toda colaboração que você teve neste trabalho (e não só neste), e também por escutar todos meus desabafos.

E nesses anos de trabalho, uma empresa surgiu. Três amigos se juntaram e formaram a 3M, para solucionar todos os problemas com as melhores gambiarras! Messi, você é o cara! Ainda não conheci alguém tão empolgado e alto astral que nem você. Te admiro muito, seu pinguinzão! Obrigado por toda parceria! *Germano, miraxx!!!* Esse é parceiro de todas as horas, seja pra serrar tronco pra festa Junina 10 da noite, ou pra ir num barzinho no Triângulo (sério cara?) nesse mesmo horário. Até entrar em lata de lixo, já entramos, hahahaha. Parceria é isso, não tem hora, nem lugar. Obrigado por tudo, irmão!

Ah, e o melhor reclamador e treteiro da NASA? Como esquecer do Flavinho? Apesar dos diversos atritos, sempre resolvemos de forma amistosa, e depois ficava tudo

certo, íamos pro bar, e vida que segue. *Abigo*, gosto de você um montão, obrigado por todo esse tempo trabalhando juntos! Sinto falta das discussões com você, que sempre acrescentavam de alguma forma. Você é fod..!

E o meu vermezão preferido? Rods! Essa trajetória é antiga, desde 2007! Às vezes, eu olho para onde estou hoje, fazendo o que eu faço, e vejo que essa parceria nos últimos 4 anos (6 meses com você roncando do meu lado, hahahaha) me fez crescer muito, e descobrir o que eu gosto de fazer! Valeu pela parceria, você é d+! Espero que possamos voltar a trabalhar juntos um dia!

Não posso esquecer daquele que informalmente orientou parte deste trabalho e é a pessoa mais empolgada com *deep learning* e afins que eu conheço! Jorge, obrigado por todas discussões mirabolantes e materiais. Você foi fundamental neste trabalho!

Tenho que também agradecer a duas pessoas que além de grandes amigos, contribuíram nessa jornada com revisões, discussões, desabafos. Cabrunco e T. Pedruzzi, minhas autarquias da ciência! Vocês são minha referência para muitas coisas na vida, obrigado por fazerem parte da minha trajetória e da minha vida!

Também agradeço a todos outros que passaram por minha vida nessa jornada, que se tornaram grandes amigos e amigas. A lista é extensa, e seria injusto esquecer de alguém! Vocês de alguma forma acrescentaram na construção de quem eu sou hoje. Agradeço também a aqueles da *idwall* que estão mais próximos de mim no dia-a-dia: Luiz, Djow, Serjão e Gabis, muito obrigado pela amizade e por tornarem o trabalho na *idwall* prazeroso. Esses últimos 5 meses teriam sido bem mais desgastantes! Sem todos vocês, eu não teria concluído esta etapa!

Agradeço ao CNPq e ao PPGEE que me concederam a bolsa de mestrado.

Muito Obrigado!

Felippe Mendonça de Queiroz

Resumo

Sistemas inteligentes capazes de compreender o movimento humano podem ser usados para diversas finalidades. Por isso, existe atualmente uma tendência e um esforço para que esses sistemas se tornem cada vez mais presentes no nosso dia-a-dia. Entre suas possíveis aplicações estão por exemplo, prover uma interface de interação com seres humanos, analisar o movimento de indivíduos durante processos de reabilitação, identificar atividades suspeitas a partir de imagens de videomonitoramento, dentre outras. Nesse contexto, esta Dissertação de Mestrado apresenta componentes de um sistema de localização de gestos, abordando aspectos práticos para a implementação deste. O problema foi abordado em duas etapas: a primeira consiste na obtenção da localização tridimensional de seres humanos, utilizando apenas um sistema multicâmeras calibrado. A etapa seguinte, é responsável por localizar a execução de gestos utilizando a pose humana. Esta localização consiste em determinar os instantes de início e fim deste gesto. Este sistema visa possibilitar o desenvolvimento de outros trabalhos, que possam utilizar tanto a pose humana quanto a localização dos gestos. Para ambas etapas, o método foi validado em bases de dados, avaliando além da precisão, a viabilidade de se utilizar em aplicações em tempo real. Ao final, uma discussão é feita envolvendo uma visão geral do sistema, e como ambas partes desenvolvidas podem ser utilizadas em conjunto.

Abstract

Intelligent systems capable of understanding human motion can be used for several purposes. That is why there is a great effort to develop and include such systems in our daily lives. They can be used to provide, for instance, interfaces for human-machine interaction or even help human-human interaction, movement analisys for people rehabilitation, detection of suspicious activities and many other applications. In this context, this Master's Thesis presents components of a gesture localization system, addressing practical aspects for its implementation. Gesture detection or localization means to determine the instants when a gesture starts and ends. The problem was tackled in two steps. The first is to obtain the three-dimensional location of humans, using just a calibrated multicamera system. The following step is responsible for detecting the execution of gestures or actions using human pose. For both steps, the method was validated using known databases, and calculating the accuracy and feasibility of using it in real time applications. The developed system may be used by other works that apply either human pose or gesture detection for their purposes. Finally, a discussion is made involving an overview of the system that shows how both developed methods can be used together.

Lista de ilustrações

Lista de tabelas

Lista de quadros

Sumário

1 Introdução

Os constantes avanços tecnológicos em eletrônica e computação fazem com que a cada dia se tenha dispositivos cada vez mais inteligentes e com funcionalidades que visam facilitar a vida do usuário. Estes avanços não se limitam a dispositivos portáteis como *smartphones*, mas também a ambientes equipados com uma variedade de sensores e atuadores, que estão, à todo momento, coletando e analisando dados para tomar decisões tendo em vista o benefício dos usuários e do meio ambiente. Tais ambientes vêm sendo tema de pesquisa desde meados da década de 1990, quando o conceito de Espaços Inteligentes foi apresentado em (LEE; APPENZELLER; HASHIMOTO, 1998).

Desde essas primeiras iniciativas, diversas pesquisas na área de Espaços Inteligentes vêm buscando maneiras de facilitar ou auxiliar o dia a dia dos usuários. Alguns trabalhos se limitam à tomada de decisões baseadas em dados de sensores que são coletados do ambiente como temperatura e uso de energia elétrica, sem se preocupar com a interação com o usuário (HELAL et al., 2005; BYUN et al., 2012; COOK et al., 2013; THOMAS et al., 2013). Em contrapartida, outros trabalhos propõem a utilização de telas para que o usuário possa interagir (LEE, 2007; LEE et al., 2012), enquanto outros possibilitam uma interação via voz (MAZO et al., 1995; GRANATA et al., 2010).

Entretanto, se o objetivo principal for auxiliar diretamente o usuário na realização de uma tarefa ou executar algo que envolva movimentação no ambiente de trabalho, esses ambientes necessitam de entidades físicas para realizarem tais atividades, como por exemplo, robôs móveis. Desta forma, juntamente com o surgimento do conceito de ambientes inteligentes, surgiam as pesquisas na área de robótica de serviço como em (SAAD et al., 1994; FIORINI; ALI; SERAJI,), com o objetivo de desenvolver sistemas robóticos para auxiliar pessoas em suas casas e/ou hospitais a realizarem tarefas do cotidiano. Tal ideia complementa a proposta de ambientes inteligentes, justificando o uso de robôs nestes, ao mesmo tempo que amplia a área de percepção e atuação dos robôs de serviço.

Nos casos em que há a necessidade da interação direta entre o usuário e o Espaço Inteligente ou dispositivos incluídos nele, é interessante que, para que a experiência seja a melhor possível, o espaço forneça ao usuário maneiras simples de interagir com ele (HASHIMOTO, 2003). Tal interação pode ser realizada por meio de diferentes interfaces, desde programação, controles manuais, controles mentais, gestos e voz. Dependendo do dispositivo com o qual o usuário está interagindo e a tarefa que se deseja realizar, pode haver uma interface mais apropriada para que tal interação ocorra de forma efetiva.

As interfaces menos intuitivas são normalmente as que requerem alguma formação técnica como conhecimento de linguagens de programação ou habilidades em manusear

equipamentos específicos. Mesmo assim, muitas vezes elas se mostram como a melhor opção. Por exemplo, o uso de controles manuais para a interação com robôs de resgate (MURPHY, 2004) e robôs cirurgiões (GREER; NEWHOOK; SUTHERLAND, 2008) mostraram-se apropriadas para tais tarefas. No primeiro caso, apresentam bons resultados mesmo não havendo nenhum conhecimento à priori (MURPHY, 2004), e no segundo são importantes exatamente por executarem apenas o que o cirurgião deseja, não correndo o risco de alguma incisão inapropriada (GREER; NEWHOOK; SUTHERLAND, 2008).

Por sua vez, esses tipos de interação podem ser pouco efetivos na execução de algumas tarefas, principalmente quando o Espaço Inteligente ou dispositivo deve ser utilizado por pessoas com pouco ou nenhum conhecimento técnico em computação e robótica, e sem nenhuma etapa de treinamento aprofundado para utilização do sistema. Desta forma, nos casos em que é possível, considerando-se a ubiquidade que o ambiente deve proporcionar, interfaces baseadas em gestos e/ou fala parecem atender melhor a tal propósito.

O uso da fala para interagir é considerado mais natural e intuitivo (BREAZEAL; ARYANANDA, 2002). Além do mais, a informação contida na voz vai além do conteúdo linguístico, podendo ser possível inferir características importantes para a interação, como a emoção do indivíduo, por exemplo (COWIE et al., 2001). Outro aspecto que a torna uma interface interessante, é que ela oferece a possibilidade de que mesmo pessoas com determinados tipos de deficiência possam interagir (MAZO et al., 1995).

Assim como a fala, os gestos compartilham da mesma fonte semântica (MCNEILL, 2000). Entretanto, gestos e fala não são redundantes, sendo na verdade complementares. Dessa maneira, uma interface baseada em voz e gestos pode fornecer mais informações para a interação e ser ainda mais intuitiva para o usuário.

Porém, a maneira como uma pessoa realiza uma interação por meio de uma interface pode variar de acordo com suas experiências anteriores, sua cultura, país de origem e língua. Por exemplo, se for pedido a pessoas de nacionalidades diferentes para que elas descrevam uma ação com palavras, é provável que a construção das frases se dê de forma diferente, devido às características da língua. Contudo, se for pedido para fazer uma descrição não-verbal dessa mesma ação, por meio de gestos por exemplo, haverá um padrão de execução entre as pessoas (GOLDIN-MEADOW et al., 2008). Portanto, a utilização de gestos para a interação em um Espaço Inteligente pode representar uma interface que seja mais universal, mantendo ainda a intuitividade e abrangendo um maior grupo de pessoas, com menor dependência da língua falada pelo usuário.

Há algum tempo que os gestos já vêm sendo pesquisados. Inicialmente, os trabalhos se concentravam no reconhecimento de gestos estáticos por serem mais simples de serem detectados. Mas, à medida que novos estudos e metodologias foram surgindo, a detecção e reconhecimento de gestos dinâmicos foram ganhando espaço. Atualmente,

muitos trabalhos utilizam diferentes técnicas e sensores (QAMAR et al., 2015; ZHU et al., 2017b; ESCALERA; ATHITSOS; GUYON, 2017; XU et al., 2015; KIM; CHUNG; KANG, 2016; KONEČN; HAGARA; SEMINÁR, 2014), mas poucos exploram a utilização desta interface de interação em Espaços Inteligentes (CASILLAS-PEREZ et al., 2016; KÜHNEL et al., 2011; Qian Wan et al., 2014).

O problema de reconhecimento de gestos e ações se insere em diversas áreas da visão computacional e processamento de sinais, como a detecção de pessoas em imagens, estimativa da pose humana, rastreamento de pessoas em vídeos, análise de séries temporais, entre outras (ZHANG et al., 2019). Além disso, o problema é abordado utilizando diferentes técnicas, uma vez que esta tarefa pode ser realizada a partir de diferentes fontes de dados: imagens apenas de uma câmera colorida, informação de profundidade, posições de juntas de esqueletos, ou ainda, a combinação desses dados, podendo até ter sensores redundantes com diferentes pontos de vista daquele em que se deseja reconhecer um gesto ou ação.

O uso de somente imagens RGB é bem desafiador, pois, para que se obtenha sucesso, é importante que a extração de características da imagem seja bem realizada. Este não é um problema apenas no reconhecimento de ações, mas sim em diversos problemas de visão computacional e aprendizagem de máquinas. Para esta abordagem, as características a serem extraídas estão nas imagens, bem como características temporais que representam a evolução do gesto ou ação. Assim, tentar resolver este problema utilizando apenas imagens RGB, significa lidar com características espaço-temporais. Dessa forma, a grande variabilidade de cenários e a possibilidade de alterações das características ao longo do tempo torna tal tarefa ainda mais complicada.

Em contrapartida, o surgimento e evolução de sensores de profundidade como o *Kinect*, possibilitou o desenvolvimento de diversas técnicas nesta área com ótimos resultados, uma vez que a partir da informação de profundidade, é possível segmentar o contorno do corpo humano, e utilizando assim apenas a silhueta deste para a execução da tarefa de reconhecimento de gestos e ações. Isto representa uma informação menos ruidosa, uma vez que não possui variações de iluminação, cores, ambientes, etc, fazendo com que métodos que utilizam esta informação para tal propósito apresentem melhores resultados.

Além disso, esses sensores de profundidade possibilitam a fácil extração de dados de esqueleto (SHOTTON et al., 2011), estimando com precisão as coordenadas tridimensionais de pontos de interesse do corpo humano. Estes por sua vez, podem ser utilizados no reconhecimento de ações, e, apresentam uma grande vantagem em relação à informação de profundidade ou às imagens RGB: sua informação é menos sujeita a ruídos provenientes do ambiente, além de ser muito mais representativa para gestos e ações (PATRONA et al., 2018). Dessa maneira, a utilização dos dados do esqueleto ao longo do tempo para o reconhecimento de ações e gestos apresenta excelente resultados. Contudo, esses sensores utilizam luz infravermelha para obter a informação de profundidade, o que os fazem ter um

alcance limitado além de não funcionarem bem em ambientes com incidência de luz solar. Além disso, a infraestrutura requerida para a instalação destes sensores é mais complexa do que a de câmeras comuns. Estas por sua vez, podem já estar presentes no ambiente, desempenhando a função de videomonitoramento.

Dessa maneira, sistemas desenvolvidos para realizar o reconhecimento de gestos e ações que utilize os dados de esqueleto, por oferecerem menos ruído em sua informação gerada, em geral atingem melhores resultados. Contudo, dada às limitações de sensores de profundidade, a utilização de apenas câmeras coloridas representa um melhor cenário pensando em aplicações mais versáteis. Nesse contexto, surgiu uma área que tem por objetivo estimar a pose humana a partir de apenas imagens. Essa pose pode ser bidimensional, ou seja, obtém-se a localização nas imagens de pontos de interesse do corpo humano, ou tridimensional, em que se obtém a localização no espaço desses pontos. O segundo é bem mais desafiador, uma vez que a obtenção de informação tridimensional utilizando uma câmera apenas, por exemplo, é um problema que pode possuir diversas soluções devido às ambiguidades existentes. Já o primeiro, é bem mais simples, e apresenta trabalhos muito bem sucedidos (PATRONA et al., 2018; CAO et al., 2016), o que fez retomar a atenção para métodos de reconhecimento de gestos e ações utilizando a pose humana.

Com isso, surgiram também trabalhos que tem como objetivo obter a pose humana tridimensional, que utilizam como entrada a pose 2D, substituindo assim sistemas mais complexos que são utilizados para obter a localização no espaço de juntas do esqueleto humano. Estes podem utilizar uma câmera ou múltiplas. A utilização de apenas uma câmera, por mais que seja interessante, pela natureza do problema vai sempre apresentar problemas de ambiguidade na obtenção das coordenadas tridimensionais, além de, em casos oclusão, não ser possível obter uma estimativa. Dessa maneira, a utilização de múltiplas câmeras para esta finalidade é bem promissora, além de ser viável economicamente dado o fácil acesso câmeras comuns que seriam suficientes para tal problema.

Contudo, para se realizar o reconhecimento de gestos ou ações em tempo real, o sistema deve ser capaz de responder com um pequeno atraso pois, normalmente, esses sistemas são utilizados em um ambiente de interação, e atrasos podem gerar más experiências aos usuários. Neste aspecto, podemos classificar os métodos em duas categorias: aqueles que possuem um baixo tempo computacional de resposta, e aqueles com baixo tempo de observação. Para o primeiro, uma resposta é gerada antes que uma nova amostra seja gerada, enquanto que para o segundo, o tempo de resposta não deve extrapolar o requisito de tempo real do sistema como um todo (LI et al., 2018b).

Além disso, o reconhecimento em tempo real envolve outros desafios como por exemplo determinar o tamanho da janela de execução, fazendo com que vários trabalhos tentem resolver esse problema utilizando janelas deslizantes, ou técnicas com múltiplas escalas (RAHMANI et al., 2014; NEVEROVA et al., 2014; ZHAO et al.,). Em contrapartida,

outros trabalhos buscam primeiramente segmentar uma execução para, em seguida, realizar a classificação, algumas vezes aproveitando-se de métodos *off-line* já bem sucedidos. Nesse contexto, alguns trabalhos buscam desenvolver meios de localizar um gesto ou ação em uma dada sequência, o que corresponde a determinar o instante de início e fim de uma execução.

1.1 Objetivos

O trabalho apresentado nesta dissertação tem como objetivo mostrar partes da construção de um sistema que realize a localização de gestos ou ações em tempo real, utilizando um sistemas multicâmeras a cores.

Contextualizando, este trabalho foi desenvolvido no Laboratório VIROS (*Vision and Robotics Systems*) do PPGEE (Programa de Pós-Graduação em Engenharia Elétrica), no qual realizam-se pesquisas nas áreas de Ambientes Inteligentes e Visão Computacional. Tais temas estão alinhados com estudos na área de reconhecimento e compreensão de atividades humanas e processamento de eventos. Dessa maneira, este trabalho visou desenvolver partes de um sistema que fosse capaz de extrair informações de apenas imagens coloridas, e fornecer os instantes de início e fim da execução de um gesto ou ação. Portanto, foram desenvolvidos os seguintes tópicos neste trabalho:

- Estimativa da Localização Tridimensional de Juntas de Esqueletos
- Localização de Gestos e Ações a Partir da Pose Humana

Ambos desenvolvimentos foram pautados em explorar aspectos de implementação que possibilitassem a execução de tais métodos em tempo real, trazendo métricas que mostrassem evidências desta característica, além de também avaliar a eficácia de tais métodos. Não foi um objetivo deste trabalho alcançar as melhores métricas ou os melhores modelos para resolver os problemas aqui propostos, mas sim, trazer abordagens que se mostrassem capazes de operar em um sistema, dando uma visão geral de seu funcionamento com cada módulo desenvolvido. Ao final, será apresentada uma visão geral de como as etapas desenvolvidas podem se conectar em um sistema único e, como outros trabalhos podem ser desenvolvidos a partir de tal sistema. Além disso, serão apresentadas possíveis melhorias ao sistema, para suprir limitações e compor uma série de trabalhos relacionados que podem vir a ser desenvolvidos.

1.2 Estrutura do dissertação

Esta dissertação está dividida em 5 capítulos. O Capítulo 1, corresponde à introdução, contendo uma breve contextualização do tema abordado, com as justificativas para o trabalho desenvolvido, além dos objetivos desta pesquisa. Em seguida, o Capítulo 2 apresenta os trabalhos relacionados e conceitos teóricos que abrangem os temas relacionados a este trabalho.

Os Capítulos 3 e 4 compreendem juntos o desenvolvimento deste trabalho, sendo respectivamente referentes à estimativa da localização tridimensional de juntas de esqueleto, e a localização de gestos a partir da pose humana. Para ambos, há uma seção de experimentos e resultados exclusiva para o tema abordado naquele capítulo. Por fim, o Capítulo 5 faz um apanhado geral das conclusões alcançadas com o desenvolvimento deste trabalho. Além disso, traz uma visão geral do sistema, e de como os dois temas abordados nesta dissertação estão relacionados. São também discutidas melhorias para o sistema proposto, enumerando possíveis trabalhos futuros.

2 Referencial Teórico e Trabalhos Relacionados

Este capítulo apresenta aspectos teóricos dos temas abordados neste trabalho, assim como trabalhos que estão relacionados. Como se propôs abordar dois temas normalmente abordados separadamente na comunidade científica, mas aqui foram relacionados, será destinada uma seção para a estimativa da localização tridimensional das juntas de esqueletos, e outra para a localização de gestos.

2.1 Estimativa da Pose Humana

Detectar as juntas e esqueletos de indivíduos em imagens e vídeos tem sido o objetivo de diversas pesquisas na área de visão computacional, realidade virtual e realidade aumentada. A posição e movimentação de indivíduos na cena são muitas vezes informações essenciais para alguns trabalhos. Podemos encontrar aplicações em diversas áreas utilizando esta informação, como na análise comportamental (ZHU; CHEN; GUO, ; LUN; ZHAO, 2015), em sistemas de videomonitoramento (BALAZIA; SOJKA, 2018; LI et al., 2018a), interações com ambientes de realidade aumentada (ELHAYEK et al., 2018; SRIDHAR et al., 2016), ou ainda, no contexto médico para a análise de marcha e detecção de patologias relacionadas (CLOETE; SCHEFFER, 2010). Em qualquer dessas situações, é comum utilizar a informação ao longo do tempo para extrair alguma informação. Esta informação é conhecida como MoCap (do inglês, *Motion Capture*), e consiste em capturar a informação da pose humana por meio de sensores e armazená-la ao longo do tempo para algum propósito (BREGLER, 2007). Este termo surgiu na década de 70, e na época ficou famosa principalmente em aplicações voltadas para o entretenimento.

Desde então, diversos sistemas foram e vêm sendo desenvolvidos para realizar a captura da informação do movimento do corpo humano. Alguns deles é necessário adicionar algum dispositivo ao corpo do indivíduo. Naqueles sistemas que utilizam sensores inerciais, às vezes se faz necessário até incorporar um módulo extra com um elemento concentrador de informações, além dos dispositivos nos pontos estratégicos do corpo. Tais sistemas que acoplam um sensor de movimento ao corpo são classificados como diretos (COLYER et al., 2018). Para os sistemas que utilizam apenas sensores óticos, conhecidos como indiretos, se faz necessário adicionar marcadores ao corpo, podendo ainda, ter que utilizar uma vestimenta de cor única para que o sistema funcione adequadamente. Esses marcadores podem ser desde esferas de uma cor pré-determinada até diodos emissores de luz infravermelha. Ou o sistema pode dispensar a adição de dispositivos ao corpo da

pessoa. Esses, em geral, utilizam um conjunto de câmeras e podem também necessitar de alguma iluminação estruturada para compor o sistema.

A empresa pioneira no desenvolvimento desses sistemas foi a Vicon [1], que possui desde sensores inerciais, até sistemas com câmeras infra-vermelho voltadas para aplicações nas mais diversas áreas. Ou ainda a Qualisys [2], que também oferece esse tipo de soluções. Contudo, ambas possuem preços elevados e necessitam de uma preparação no ambiente para suas instalações. Entretanto, deve-se levar em consideração que tais sistemas possuem alta precisão, e são indicados para aplicações que possuem tal requisito.

Por outro lado, existem sistemas que dispensam o uso de marcadores e, consequentemente, apresentam um custo mais barato. No Quadro 1 estão apresentados os principais sistemas comerciais que dispensam o uso de marcadores. Contudo, estes não deixam claro seu nível de precisão ao comparar com sistemas que utilizam marcadores (COLYER et al., 2018), além de possuírem restrições quanto aos ambientes nos quais podem ser utilizados, à quantidade de câmeras, ou se operam ou não em tempo real. Ainda, esses sistemas demandam alto poder computacional e instalação de *hardware* específicos.

Quadro 1 – Comparativo entre os principais sistemas comerciais atuais de captura de movimento por imagem que não necessitam de utilização de marcadores.

Empresa	**Câmeras**	**Ambientes de captura**	**Tempo real**
Captury Studio Ultimate The Captury <www.thecaptury.com>	Ilimitado	Não apresenta restrições	Sim
BioStage Organic Motion <www.organicmotion.com>	8-18 (120 fps)	Ambientes controlados (laboratórios)	Sim
Shape 3D Simi <www.simi.com>	Até 8 e coloridas.	Pode funcionar em ambientes externos, mas necessita de *background* com pouca variação e bastante contraste	Não

Fonte: Adaptado de (COLYER et al., 2018).

Pensando em aplicações com custos mais acessíveis, e que dispensem uma grande intervenção no ambiente, além de não depender de marcadores, o desenvolvimento de sensores de profundidade possibilitou o aparecimento de um grande número de aplicações principalmente na área de entretenimento. Estes sensores de profundidade podem ser de dois tipos: aqueles que utilizam um par de câmeras bem próximas, formando um sistema

[1] https://www.vicon.com/
[2] https://www.qualisys.com

estéreo e que, a partir das imagens, obtém a profundidade correspondente a cada *pixel*; ou ainda, podem possuem apenas uma câmera, mas se utilizam de artifícios luminosos para obter a profundidade. Deste tipo, podem haver os que utilizam iluminação estruturada, usualmente infravermelha, ou os conhecidos como ToF (do inglês, *Time-of-flight*), que medem o tempo que um pulso de luz demora para retornar à câmera para determinar a distância.

O sensor que ficou mais famoso e revolucionou a indústria dos jogos interativos foi o Kinect (SHOTTON et al., 2011). Este consiste em um sensor RGB-D (*Red,Green,Blue - Depth*), isto é, ele produz um mapa de profundidade, juntamente com uma imagem colorida e, a partir deste mapa de profundidade, localiza a posição tridimensional de juntas estratégicas de esqueletos. Não só a indústria de jogos se beneficiou deste sensor, mas também seu surgimento impulsionou o desenvolvimento de diversas trabalhos na área de reconhecimento de gestos (LUN; ZHAO, 2015; LARABA et al., 2017) e compreensão do movimento utilizando as informações de profundidade e da pose humana.

A utilização da pose humana (esta por sua vez composta pelas juntas de esqueleto) e de informação de profundidade em trabalhos que buscam reconhecer e classificar gestos e ações apresenta uma série de vantagens ao se comparar com a utilização de apenas imagens coloridas. Primeiramente, estas informações compreendem uma massa de dados menor (ASADI-AGHBOLAGHI et al., 2017a) do que a de uma imagem comum, o que torna processos de treinamento de modelos menos custosos (WANG et al., 2018) computacionalmente. Além disso, são mais robustas a variações que o ambiente pode causar em imagens, como variações bruscas de iluminação.

Ainda assim, mesmo que mapas de profundidade correspondam a uma informação com menos ruído, pois conseguem extrair a informação de silhueta do indivíduo, a pose humana oferece uma vantagem: a partir de sua informação ao longo do tempo, é possível de maneira mais direta descrever o movimento do corpo humano com poucas variáveis. Isto é, a pose humana é muito mais representativa para descrever o movimento humano (PATRONA et al., 2018) ao ser comparada com imagens ou mapas de profundidade. Para recuperar e utilizar a informação de pose humana, é necessário definir uma maneira de representá-la.

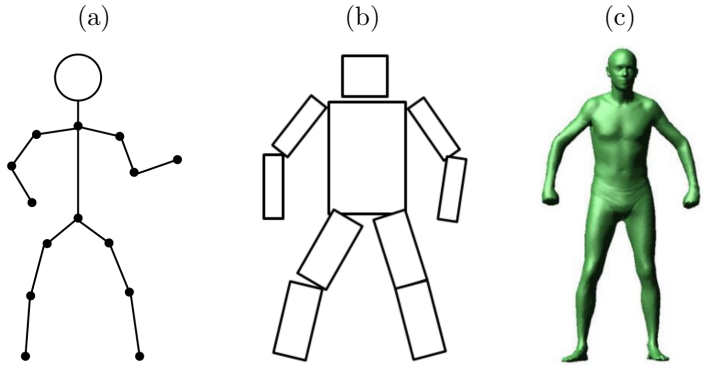
2.1.1 Modelos de Pose Humana

O problema de definir um modelo para a pose humana está relacionado com sua obtenção, seja no domínio bidimensional, seja tridimensional. Para ambas as situações, existem duas abordagens: uma é chamada de generativa e a outra de discriminativa. Técnicas generativas tentam ajustar os dados a um modelo previamente conhecido. Para o caso tridimensional, define-se um modelo volumétrico do corpo humano o qual é projetado na imagem a fim de ajustá-lo ao indivíduo. Já as técnicas discriminativas utilizam informações de aparência obtidas da imagem e então estimam uma localização. Para isso, constrói-se

um modelo que aprende a partir de dados rotulados como determinar a localização das juntas de um ser humano. Por esse motivo, técnicas discriminativas são mais rápidas que as generativas, uma vez que estas utilizam um processo de otimização que costuma envolver um grande número de parâmetros (GONG et al., 2016). Para as abordagens discriminativas, é necessário definir um modelo para o corpo humano.

Esse modelo pode ser cinemático, planar ou volumétrico. Para o cinemático, normalmente são escolhidos pontos de articulações do corpo, como joelhos e cotovelos, ou pontos com característica de aparência específica como por exemplo o nariz. Na Figura 1a é possível ver uma ilustração desse modelo. Além disso, a escolha desses pontos é de tal forma que existam estruturas rígidas entre eles, como por exemplo o antebraço, pois isso pode ser usado como uma condição de contorno, ou ainda, definir sistemas de referenciais para se medir ângulos entre segmentos definidos por pontos do modelo. Os modelos planares, como o exemplo da Figura 1b, definem regiões retangulares do corpo humano, e, normalmente, utilizam informações de cor dessas regiões para fazer o reconhecimento de tais áreas. Já os modelos volumétricos, são mais complexos e, portanto, mais realistas. Eles podem utilizar formas geométricas como cilindros e cones para definir membros do corpo. Ou ainda, podem modelar o corpo por meio de pequenos triângulos conectados formando uma superfície. Cada um desses triângulos forma uma textura, que combinados formam a superfície do corpo como mostrado na Figura 1c.

Figura 1 – Tipos de modelos para o corpo humano. Em (a) um exemplo de modelo cinemático, (b) planar e (c) volumétrico.

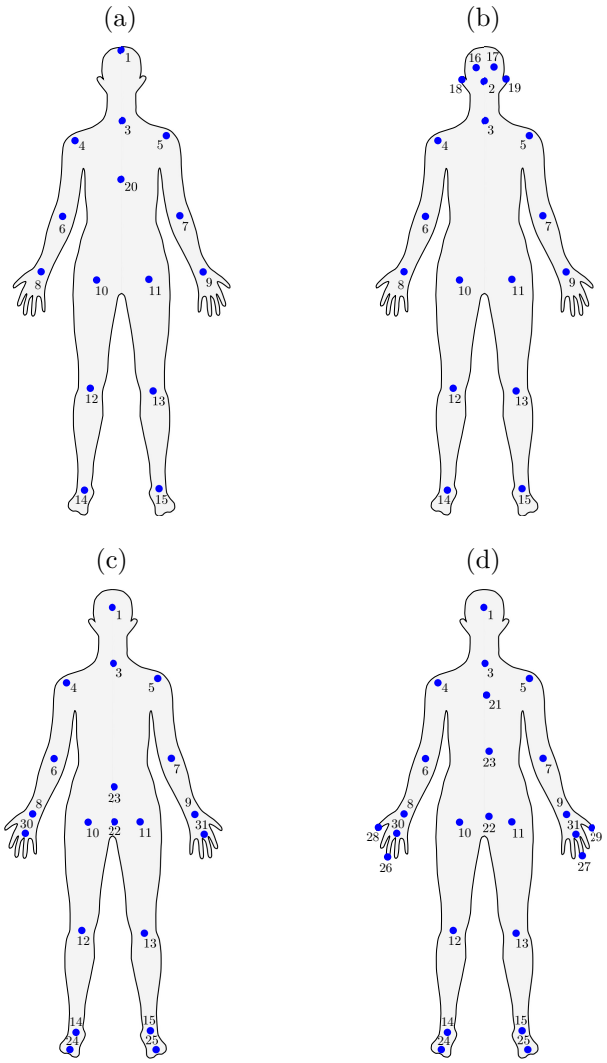


Fonte: Adaptado de (GONG et al., 2016).

Diversos modelos cinemáticos já foram definidos pela comunidade. Tais modelos possuem um número diferente de pontos do corpo humano, mas com alguns pontos em comum. Alguns desses foram aplicados em bases de dados amplamente conhecidas, já outros foram criados para um propósito específico, como para o uso do sensor *Kinect*, possibilitando o surgimento de novas bases de dados que foram capturadas com tais sensores. Na Figura 2 encontram-se 4 modelos muito comuns. Dois deles são utilizados em bases de dados, são eles o MPI (Figura 2a) e o COCO (Figura 2b), já os outros dois são

os encontrados nos sensores *Kinect* v1 (Figura 2c) e v2 (Figura 2d). Em cada uma das imagens dos modelos, foram identificados os pontos com números, e estes números são referentes às partes do corpo listadas no Quadro 2. Neste, é possível verificar a existência de partes em comum entre os modelos.

Figura 2 – Localização das juntas dos principais modelos apresentados no Quadro 2. (a) MPI, (b) COCO, (c) Kinect v1 e (d) Kinect v2.



Fonte: Adaptado de (ROCHA et al., 2015) e
<http://github.com/CMU-Perceptual-Computing-Lab/openpose>

2.1.2 Técnicas de Estimação de Pose Humana Através de Câmeras Comuns

Apesar de sensores RBG-D possuírem um custo acessível e conseguirem entregar em tempo real a localização das juntas de esqueleto de múltiplos indivíduos, tais sensores não funcionam bem em ambientes externos devido à incidência de luz solar, e também possuem limitação de distância para seu funcionamento. Além disso, o(s) indivíduo(s) deve(m) estar em posição frontal para o sensor. No aspecto físico, tais sensores normalmente possuem

Quadro 2 – Partes do corpo presentes nos principais modelos existentes e o total de cada modelo. As partes que apresentam dois nomes possíveis, aquele que está entre parênteses corresponde ao nome usado no modelo Kinect v2. Cada parte possui um identificador único colocado na coluna ao lado de seu nome. Aquelas que apresentam um * ao lado do nome, possuem o mesmo nome, mas localizações diferentes dependendo do modelo, enquanto que as que possuem § significa que foram utilizadas no método proposto para a localização de gestos e ações apresentado no Capítulo 4.

Parte do Corpo	#	**Modelo**			
		MPI	COCO	Kinect v1	Kinect v2
Head *§	1	✓		✓	✓
Nose	2		✓		
CenterShoulder (Neck) §	3	✓	✓	✓	✓
RightShoulder §	4	✓	✓	✓	✓
RightElbow §	5	✓	✓	✓	✓
RightWrist §	6	✓	✓	✓	✓
LeftShoulder §	7	✓	✓	✓	✓
LeftElbow §	8	✓	✓	✓	✓
LeftWrist §	9	✓	✓	✓	✓
RightHip §	10	✓	✓	✓	✓
RightKnee §	11	✓	✓	✓	✓
RightAnkle	12	✓	✓	✓	✓
LeftHip	13	✓	✓	✓	✓
LeftKnee	14	✓	✓	✓	✓
LeftAnkle	15	✓	✓	✓	✓
RightEye	16		✓		
LeftEye	17		✓		
RightEar	18		✓		
LeftEar	19		✓		
Chest	20	✓			
Spine	21				✓
HipCenter (BaseSpine) *§	22			✓	✓
Spine (MiddleSpine) *	23			✓	✓
LeftFoot	24			✓	✓
RightFoot	25			✓	✓
LetfTipHand	26				✓
RightTipHand	27				✓
LetfThumb	28				✓
RightThumb	29				✓
LetfHand	30			✓	✓
RightHand	31			✓	✓
Total de Partes		15	18	20	25

Fonte: Produção do próprio autor.

interface de comunicação com computador ou console no qual é utilizado, restringindo sua instalação sem a presença destes.

Por outro lado, a utilização de apenas câmeras para a estimação da pose humana

pode se mostrar vantajosa, uma vez que estas possuem uma variedade de interfaces de comunicação e, em vários casos, ambientes já possuem câmeras instaladas para fins de videomonitoramento. Ou ainda, a instalação destas câmeras para tal finalidade não se restringe apenas à estimativa de pose: outras aplicações podem ser desenvolvidas utilizando tais imagens (LORA et al., 2015).

Neste contexto, a utilização de câmeras comuns para a obtenção da pose humana vem despertando grande interesse nos últimos anos, apresentando resultados bastante promissores. A obtenção da pose humana utilizando imagens pode ser realizada no plano bidimensional, obtendo-se a localização das juntas do esqueleto na imagem, ou no espaço tridimensional quando se usa mais de uma imagem e se obtém a correspondência entre as juntas de interesse, possibilitando assim a recuperação da localização 3D dos pontos. Para a obtenção da pose 2D, diversos trabalhos foram desenvolvidos nos últimos anos utilizando *deep learning*, os quais obtiveram resultados muito acima da média comparado com as técnicas clássicas (DANG et al., 2019), além de conseguirem ser executados em tempo real com um *hardware* acessível.

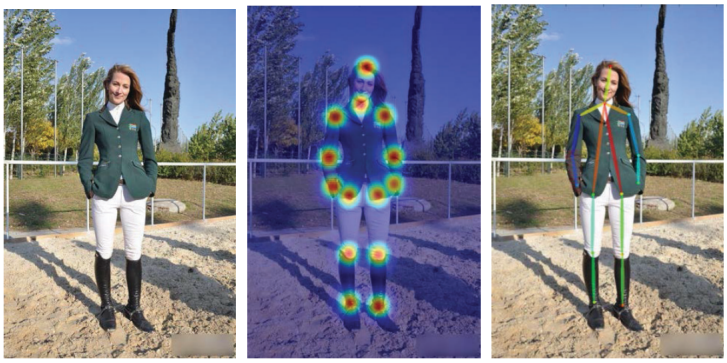
Os métodos de obtenção da pose humana podem ser subdivididos em dois subgrupos: os que funcionam apenas para um indivíduo, e aqueles que são capazes de recuperar esta informação para múltiplos indivíduos em uma mesma cena. Mesmo que este trabalho de dissertação se proponha a apresentar, em uma de suas partes, uma metodologia para obter a pose tridimensional, é importante também entender as técnicas de obtenção da pose 2D, pois alguns conceitos são estendidos para abordagens 3D.

2.1.2.1 Técnicas de Obtenção de Pose 2D

As técnicas que obtém a pose 2D a partir de imagens para apenas um indivíduo, consistem em basicamente detectar o indivíduo na imagem a partir da região na qual este foi encontrado. Duas abordagens podem ser tomadas: a primeira corresponde a fazer uma regressão para os pontos de interesse a partir de *features*, já a segunda, consiste em gerar mapas de calor para cada ponto de interesse, e então, localizá-los e conectá-los baseando-se em um modelo de juntas pré-estabelecido. Na Figura 3, encontra-se um exemplo de obtenção de pose humana para apenas um indivíduo, utilizando a técnica de mapa de calor.

Não existe um consenso de qual técnica é a melhor, mas alguns argumentos devem ser levados em consideração. Métodos que fazem regressão para obter a pose humana não podem ser aplicados a regiões na imagem com múltiplas pessoas, mas possuem a vantagem de poderem ser estendidos para a estimativa 3D de pose. Em contrapartida, as técnicas que utilizam mapas de calor conseguem ser aplicadas para múltiplos indivíduos, mas dependem fortemente da resolução desses mapas de calor. Além disso, necessitam de uma grande quantidade de memória para o treinamento, e não podem ser estendidas para

Figura 3 – Exemplo de um método de obtenção de pose humana 2D baseado em mapa de calor.



Fonte: Adaptado de (DANG et al., 2019).

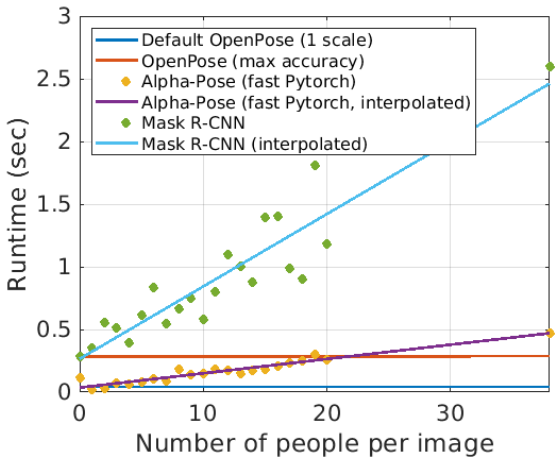
o espaço tridimensional (DANG et al., 2019).

Para a estimativa 2D com múltiplos indivíduos, existem basicamente duas abordagens: a *top-down*, que consiste em primeiramente detectar todos os indivíduos na imagem e, em seguida, localizar as juntas para cada região que contém um indivíduo; ou a abordagem *bottom-up*, que primeiro constrói mapas de calor, para em seguida conectar os pontos de interesse e gerar esqueletos para cada um dos indivíduos. Para as técnicas com abordagem *top-down*, o custo computacional está diretamente relacionado com a quantidade de indivíduos na imagem. Para as técnicas baseadas em *bottom-up*, o tempo para obtenção dos mapas de calor é fixo, mas a etapa que agrupa os pontos de interesse depende da quantidade de indivíduos. Entretanto, já existem trabalhos com implementações que atingiram tempos abaixo de 1 ms para 9 pessoas presentes na imagem (CAO et al., 2016), o que representa um parcela pequena se comparado com o tempo para gerar os mapas de calor. Dessa maneira, no contexto de aplicações com requisito de tempo real, as abordagens *bottom-up* são as mais indicadas, pois, além de apresentarem menor tempo de inferência, a razão tempo total por número de indivíduos também é menor.

Dentro dessa categoria de trabalho, o *OpenPose* (WEI et al., 2016; CAO et al., 2016) é um dos mais conhecidos. Ele utiliza a abordagem *bottom-up* e seu tempo de execução atinge valores totalmente plausíveis para se utilizar em aplicações de tempo real. Além de apresentar pouca variação de tempo com diferentes números de indivíduos na imagem. Na Figura 4, temos um gráfico comparativo entre o *OpenPose* e seus principais concorrentes, mostrando o tempo de execução para um dado número de indivíduos. Ele foi o que se mostrou constante quando aumentado o número de indivíduos.

De maneira geral, as técnicas de obtenção de pose 2D mostram-se bem maduras e passíveis de serem utilizadas em ambientes com requisitos de tempo real, além, é claro, de requisitarem *hardware* acessível. Em contrapartida, a estimativa tridimensional da pose humana ainda é um problema que está sendo explorado, principalmente quando se trata

Figura 4 – Comparação entre as principais bibliotecas de detecção de esqueletos 2D em imagens, apresentada no repositório oficial do *OpenPose*.



Fonte: Adaptado de <https://github.com/CMU-Perceptual-Computing-Lab/openpose>.

de múltiplos indivíduos.

2.1.2.2 Técnicas de Obtenção de Pose 3D

Atualmente, existe um grande interesse pela obtenção da pose humana 3D, pois existe uma grande quantidade de aplicações desenvolvidas utilizando a localização tridimensional de seres humanos. Os métodos desenvolvidos que tentam resolver este problema, possuem diferentes abordagens e restrições de operações, sendo difícil definir uma classificação para estes. Portanto, diversas são as classificações adotadas quando se discute esse tema.

Uma primeira classificação tem relação com a utilização ou não de um modelo de corpo. Esta divisão também se aplica às poses 2D, em que métodos generativos tentam ajustar uma informação a um modelo pré-determinado, e as técnicas discriminativas, que não se baseiam em um modelo. Outro ponto a se levar em conta na classificação destes métodos, é a quantidade de vistas utilizadas: existem métodos que utilizam apenas uma vista, enquanto outros utilizam múltiplas. Além disso, alguns métodos possuem restrição quanto ao número de indivíduos que se consegue reconstruir. O uso de técnicas de rastreamento também são abordadas para melhorar a localização, contudo por ser um problema de rastreamento, existe o problema de inicialização (HOLTE et al., 2012). Levando em conta também aspectos de implementação, nem toda abordagem leva em consideração a restrição de execução em tempo real.

Com as possíveis classificações apresentadas, além dos diferentes dados de entrada e saída utilizados em trabalhos que obtém a pose humana, apenas aqueles considerados mais relevantes no contexto desta dissertação serão abordados. Em (TAYLOR, 2000) por

exemplo, recupera-se informação tridimensional de pontos específicos do corpo humano, utilizando-se apenas uma câmera, a partir de pontos devidamente identificados na imagem. No referido trabalho, cada ponto corresponde a uma determinada junta do corpo. Conhecendo o modelo de juntas utilizado, sabe-se também os comprimentos relativos entre cada uma das juntas. Dadas essas restrições, é possível obter, a menos de um fator de escala, uma possível solução para a estrutura tridimensional do modelo de juntas.

Em (ZIEGLER; NICKEL; STIEFELHAGEN, 2006), um sistema multicâmeras é responsável por gerar uma nuvem de pontos e, a partir desta, realizar o rastreamento da parte superior do corpo, utilizando um filtro de *Kalman*. A abordagem utiliza múltiplas câmeras, gerando a nuvem de pontos a partir de pares destas. Um modelo de juntas é obtido por meio apenas dos pontos da parte superior do corpo. O método se restringe a apenas um indivíduo e pode ser executado em aproximadamente 1 *fps*.

Utilizando um sistema com três câmeras de calibrações conhecidas, (HOLTE et al., 2012) gera poses candidatas a partir das imagens por meio de um modelo que se baseia na forma, e as projeta no plano da imagem de cada câmera para realizar uma validação multi-vista, utilizando a informação de detecções 2D. Utiliza também um sistema de rastreamento com reinicialização automática, além de informação de textura para auxiliar no processo. Quando proposto, levava entre 30 e 40 segundos para ser executado, e apresentava erros menores que 10 *cm*.

Até 2014, não haviam trabalhos utilizando CNNs (do inglês, *Convolutional Neural Networks*) ao abordar problemas de localização tridimensional de seres humanos. (LI; CHAN, 2014) é o primeiro trabalho a fazer isso, e aborda o problema com uma arquitetura mista, que realiza tanto a detecção das juntas quanto a regressão da localização destas. A entrada dessa rede neural é a região da imagem na qual o humano se encontra. Ou seja, é uma solução que necessita de uma etapa inicial para determinar esta região na imagem, além de resolver o problema para apenas um único indivíduo. Sua proposta utiliza uma única câmera com um modelo de corpo humano descrito por juntas, com um erro médio entre 80 e 200 milímetros, a depender do conjunto de atividades do *dataset* utilizado para a avaliação do método. Nenhuma evidência foi dada a respeito do tempo de execução, mas, por se tratar de um trabalho que utiliza CNN, é provável que não fosse possível de se executar em tempo real.

Mesmo com o aparecimento de técnicas que utilizam CNNs, as demais continuaram sendo exploradas. Também em 2014, (BELAGIANNIS et al., 2014a) definiu um modelo pictórico formado por segmentos que representam partes do corpo humano, como antebraço e perna, além de definir restrições de rotação, translação e colisão entre as partes. Para o treinamento do modelo, ele utilizou um *dataset* multicâmera contendo as calibrações. Assim, após o treinamento, a inferência é realizada em configurações monoculares. O trabalho teve como objetivo recuperar a localização de uma ou mais pessoas na mesma

cena. No entanto, para que o segundo caso seja possível, é necessário conhecer a quantidade de pessoas da cena, ou tentar inferir pela quantidade de juntas iguais presentes na imagem. Isto, por sua vez, não é confiável já que utiliza detecções 2D que podem não retornar todas as partes do corpo. Sua execução durava cerca de um segundo para cada instante, já considerando a disponibilidade da localização 2D dos indivíduos nas imagens. Em uma das bases de dados na qual o trabalho foi avaliado, apresentou erro médio de localização abaixo de 70 mm e, além disso, um percentual de juntas corretamente recuperadas entre 62,7% e 76,0% dentre os *datasets* utilizados.

Ainda assim, o uso de CNNs se tornou bastante comum, possibilitando que, por exemplo, técnicas monoculares que podem ser executadas em tempo real fossem desenvolvidas. No trabalho apresentado em (MEHTA et al., 2017), a estimativa das coordenadas tridimensionais do esqueleto é realizada, mesmo usando-se apenas uma câmera. Entretanto, este método tem a limitação de funcionar apenas quando o indivíduo está de frente para a câmera, além de possuir uma etapa de inicialização que consiste na detecção da região que o indivíduo se encontra. Por ser uma técnica monocular, além de enfrentar problemas de oclusão, bem como ambiguidades na localização de juntas, as coordenadas são obtidas a menos de um fator de escala. Como alternativa, o autor propõe informar a altura do indivíduo para que o esqueleto detectado possa ser escalado em função desta medida.

Embora diversos trabalhos apresentem bons resultados ao tentar recuperar a pose de seres humanos a partir de imagens, sejam com uma ou múltiplas câmeras, esta é uma tarefa bem complexa, por envolver uma grande variedade de aparências dos indivíduos, cenários, pontos de vista, oclusões, dentre outros problemas encontrados em ambientes não controlados. Muitos dos desafios enfrentados na detecção 2D são também encontrados na tridimensional (LIN et al., 2017). Dessa forma, é comum encontrar trabalhos que, de alguma maneira, utilizam detecções 2D, sejam como uma entrada de etapa de algum método, sejam como inspiração para desenvolver técnicas que atendam ao problema de localização tridimensional. Em (LIN et al., 2017), por exemplo, a localização 2D é utilizada para ajudar a refinar a estimativa da localização 3D, via um processo recorrente que utiliza uma sequência de imagens. Apesar de conseguir ser executado em apenas 50 ms por imagem, o método funciona apenas para um único indivíduo presente na cena.

Na mesma linha de se utilizar informação 2D para recuperar a pose humana, (AKHTER; BLACK, 2015) estima as posições 3D das juntas a partir de detecções 2D em uma imagem. Para isso, inicialmente foi capturado um *dataset* com indivíduos em diversas poses, capaz de cobrir boa parte dos casos de ângulos entre as partes dos esqueletos anatomicamente possíveis. Com esta base de dados foi gerado então um modelo, a partir das poses que um indivíduo pode assumir. A seguir, foi proposto um método que estima a localização tridimensional das juntas usando a informação bidimensional obtida na imagem. Assim como em (MEHTA et al., 2017), esse método só funciona com um indivíduo por

vez.

Apesar dos sensores RGB-Ds entregarem a localização tridimensional de juntas, (CARRARO et al., 2018) utiliza uma rede dessas câmeras RGB-Ds para estimar a pose tridimensional das juntas de pessoas presentes no ambiente. O trabalho usa as imagens coloridas para localizar esqueletos 2D, utilizado o *OpenPose*, e com a informação de profundidade, realiza a correspondência entre as detecções na imagem, para que, em uma etapa posterior, realize a reconstrução tridimensional das juntas de cada indivíduo. Também é necessária a calibração das câmeras para realizar o processo de reconstrução. Embora apresente erros de reconstrução na faixa de 20 a 60 *mm*, a utilização de sensores deste tipo se torna inviável em certos cenários, devido a suas limitações de distância, incidência de iluminação, e instalação no ambiente.

Com uma abordagem um pouco diferente de (CARRARO et al., 2018), em (LORA et al., 2015), a estimativa da localização tridimensional dos esqueletos é feita através de uma rede de câmeras convencionais. Um detector de esqueletos 2D é utilizado, e é levado em consideração um conjunto de restrições algébricas no processo de triangulação e otimização por mínimos quadrados. Apesar dos autores alegarem que o método pode ser aplicado em tempo de execução, o artigo não traz nenhuma medição de desempenho que comprove tal afirmação. O método também necessita da calibração das câmeras do sistema.

Numa abordagem híbrida, o trabalho em (KIM, 2017) usa uma rede de câmeras convencionais e câmeras RGB-Ds para gravar os movimentos de dançarinos. A fusão das informações busca selecionar a melhor correspondência entre as detecções obtidas, empregando-se uma abordagem probabilística com filtro de partículas. Todavia, este método só funciona bem com um indivíduo, e requer um custo maior de instalação, uma vez que, além das câmeras comuns, utiliza sensores de profundidade.

Outro método multivistas foi o proposto por (EREN et al., 2017). Nesse trabalho, os autores propõem um método para obter um modelo de regressão para a posição tridimensional das juntas, contudo, utilizam a calibração das câmeras. Em cada sistema multicâmeras que o método é aplicado, uma pose candidata pode ser obtida para cada par de câmeras. Uma média dessas poses é então calculada, seguida de uma triangulação não-linear para obter a localização 3D. O método só funciona para um único indivíduo na cena.

(KADKHODAMOHAMMADI; PADOY, 2018) também resolveu o problema de estimar a pose tridimensional em duas etapas, tendo como a inicial a detecção de esqueletos nas imagens, utilizando textitOpenPose para esta. Após realizar as detecções na imagem, ele utiliza geometria epipolar para encontrar correspondências entre pares de câmeras, eliminando correspondências com distância média maior que 20 *pixels*. Além disso, (KADKHODAMOHAMMADI; PADOY, 2018) assume que pode haver falsas corres-

pondências, além de juntas com menos de duas detecções, o que inviabiliza a utilização de uma triangularização para determinar a posição tridimensional das juntas. Assim, o autor utiliza uma função de regressão para determinar a posição das juntas, cujas entradas são as detecções em todas as vistas. Vistas sem detecções são preenchidas com zeros. Dessa forma, seu modelo de regressão é treinado utilizando, além das anotações tridimensionais, as calibrações do sistema multicâmeras. Entretanto, para aplicar o método a uma outra configuração de câmeras, um novo treinamento necessita ser realizado.

Também realizando uma etapa inicial com detecção de esqueletos nas imagens, (DONG et al., 2019) combina informações geométricas entre as poses 2D com informações de aparência das imagens associadas, com o objetivo de reduzir ambiguidades. Além disso, utiliza um processo de otimização para resolver simultaneamente todas as correspondências. A utilização de aparência combinada com informação geométrica é questionável dependendo do contexto. Indivíduos com aparências muito semelhantes podem gerar falsas correspondências. Ademais, para utilizar informação de aparência, é necessário ter uma etapa para extrair tal informação, a qual, dada a região de interesse em que o indivíduo se encontra, deve computar os descritores correspondentes. Tal procedimento pode ser custoso computacionalmente o que pode inviabilizar sua utilização por uma aplicação de tempo real. Em (DONG et al., 2019) são mostradas situações em que a aparência melhorou os resultados, mas deve-se sempre considerar se a melhora é substancial quando comparada com o custo computacional e de tempo envolvidos no processo.

Outro aspecto sobre (DONG et al., 2019), é a comparação entre utilizar triangulação para realizar as correspondências entre as detecções e, em seguida, realizar a reconstrução, ou utilizar a técnica 3DPS (do inglês, *3D Pictoral Structure*) (BELAGIANNIS et al., 2014a) para recuperar a informação tridimensional direto das detecções. Ambos os métodos podem apresentar bons resultados, mas, se combinados, uma busca prévia de correspondências utilizando informação geométrica ou de aparência, pode reduzir a quantidade de estados do 3DPS, tornando sua execução mais rápida.

Uma abordagem com multicâmeras calibradas é apresentada em (ERSHADI-NASAB et al., 2018). Ela utiliza evidências extraídas de múltiplas cenas a fim de criar espaços de busca 3D. Esses espaços de busca para cada junta são então clusterizados para criar um espaço de junta referente a cada indivíduo. Informações de um detector 2D são utilizados para compor uma função potencial utilizada para estimar a pose de cada indivíduo. O algoritmo proposto funciona para múltiplas pessoas, e nenhuma medida sobre o tempo de execução é apresentada.

Portanto, como pode-se verificar, diversas são as abordagens utilizadas para a recuperação da pose humana, uma vez que esta é uma informação preciosa para diversas aplicações. Verifica-se que a utilização de múltiplas câmeras para resolver este problema vem sendo bastante explorada, pois, com um sistema multivistas, casos de ambiguidades

são evitados. Além disso, percebe-se que diversos trabalhos utilizam em seus *pipelines* a detecção 2D de esqueletos para, de alguma maneira, recuperar a informação tridimensional a partir de detecções em múltiplas vistas. Apenas alguns trabalhos se preocupam em avaliar o tempo de execução em relação ao erro de localização, bem como a viabilidade de sua utilização em aplicações de tempo real, que são a grande maioria das que se propõem em utilizar a pose humana. Vale ressaltar ainda que o emprego desta informação para a localização de gestos e ações é uma das aplicações que vem sendo exploradas, contexto no qual este trabalho de mestrado se encaixa.

2.2 Localização e Classificação de Gestos e Ações

Este é um tema que é aplicado a diversas áreas: sistemas de videomonitoramento inteligentes, interfaces de interação humano-computador, tecnologias assistivas, sistemas de direção inteligentes, entretenimento, dentre outras áreas. Nos últimos anos, esforços vem sendo concentrados em pesquisas que buscam maneiras de compreender o comportamento humano. Antes de explorar as técnicas desenvolvidas para resolver o problema de localizar e classificar gestos e/ou ações, é necessário compreender o que significam.

De acordo com (WANG et al., 2018), um gesto consiste em um movimento básico, ou um simples posicionamento de um membro do corpo como mãos, braços ou cabeça, utilizado para expressar uma ideia ou emoção. Enquanto que uma ação, consiste em um movimento realizado por uma pessoa durante um curto período de tempo, que envolve várias partes do corpo. Aumentando o nível de complexidade, uma interação é um tipo de movimento executado por dois atores, que podem ser seres humanos, ou uma combinação de ser humano e objeto. E, por fim, o nível mais complexo consiste em atividades em grupo, executadas por múltiplas pessoas e com a presença ou não de objetos. Independente da complexidade, os processos que buscam identificar gestos, ações ou atividades consistem em uma área de pesquisa que busca compreender o movimento humano. Embora possuam definições diferentes, gestos e ações são em geral abordados de maneira semelhantes quando se deseja detectá-los e classificá-los (ASADI-AGHBOLAGHI et al., 2017b).

Um dos maiores desafios da tarefa de localizar e classificar um gesto ou ação, consiste em modelar os dados de entrada para extrair informações relevantes deles. Isto é, manipular os dados para descrever o movimento humano ao longo do tempo (ZHANG et al., 2019). Diferentes tipos de representação, utilizando dados de entrada de natureza diferentes vem sendo abordados ao longo dos anos. São trabalhos que utilizam imagens, mapas de profundidade, pose humana, ou ainda a combinação de dois ou mais destes dados. Sem contar com as diferentes técnicas utilizadas para extrair as características de tais dados, que vão desde técnicas clássicas, comumente referidas como *hand-crafted features*, até algoritmos que aprendem a descrever as características, automaticamente

comuns com o avanço do *Deep Learning*.

Contudo, muitos estudos focam apenas na área de reconhecimento de gestos e ações, realizando a extração de características de uma sequência de imagens para, a seguir, classificar o gesto ou ação presente na sequência. Isto é, a partir de um vídeo já segmentado, em que o primeiro *frame* corresponde ao início da execução de um gesto, e o último o seu fim, um algoritmo é aplicado para classificar esta sequência dentro de um conjunto de classes. Pensando em uma aplicação real, estas técnicas não são efetivas, uma vez que para classificar é preciso primeiro segmentar o início e fim da ação ou gesto. Dessa maneira, uma nova área de pesquisa surgiu, que consiste na detecção e/ou localização de gestos e ações. Esta tem como objetivo determinar o espaço de tempo em que uma execução de gesto ou ação ocorreu. Algumas técnicas tentam fazer isto em conjunto com o processo de classificação, já outras entregam apenas os instantes de início e fim da ocorrência do gesto.

2.2.1 Detectando gestos e ações a partir de imagens e mapas de profundidade

O uso de apenas imagens se tornou muito frequente devido ao avanço e sucesso das técnicas utilizando redes neurais convolucionais, mas ainda é desafiador devido às variações de iluminação e cenários. Alguns trabalhos combinam o uso de imagens com alguma outra informação, como mapa de profundidade, ou ainda, fluxo ótico calculado a partir das imagens. Há ainda os que utilizam também informação de pose humana junto da informação visual, mas estes são menos comuns. Os principais trabalhos que utilizam apenas informação visual utilizam redes neurais, e possuem em geral 3 diferentes abordagens (HERATH; HARANDI; PORIKLI, 2017): Redes espaço-temporais, Redes com múltiplos caminhos e Redes neurais recorrentes.

As **Redes espaço-temporais** (do inglês, *Spatiotemporal networks*) são inspiradas nas redes neurais convolucionais tradicionais, que foram inicialmente concebidas para trabalharem com imagens, isto é, o dado de entrada pertence a um domínio bidimensional. Nas redes com abordagem espaço-temporal, se utiliza o que é conhecido como convolução 3D, em que uma nova dimensão é incorporada, e portanto, filtros tridimensionais são utilizados. Desta maneira, a informação temporal passa a compor a terceira dimensão dos dados inseridos na rede. Assim, a rede é capaz de perceber variações espaciais ao longo do tempo. Entretanto, estas arquitetura são em geral pouco flexíveis em relação ao tamanho da janela de tempo utilizada.

Mesmo que hajam estratégias para contornar o problema de escala de tempo nas redes espaço-temporal, como a utilização de *pooling*, é difícil afirmar que esta abordagem é robusta o suficiente para resolver o problema de reconhecer gestos em tempo real. Isto por que a execução destes pode ser realizada em velocidades diferentes, e em intervalos de tempos distintos. Entretanto, diversos trabalhos utilizam esta abordagem, mas poucos levam em consideração a extensão do método para uma execução em tempo real.

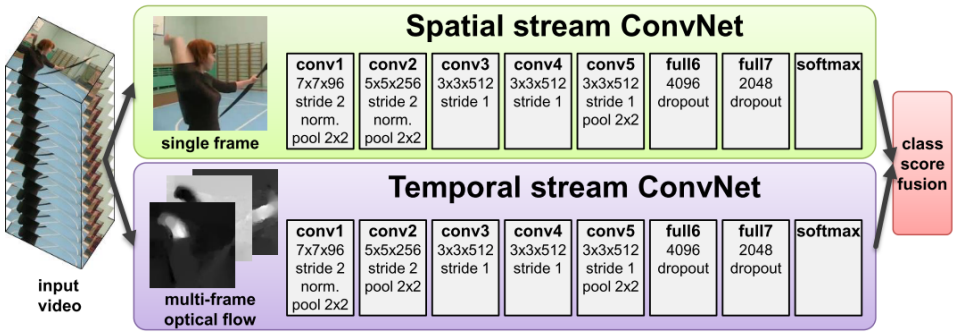
O trabalho proposto em (JI et al., 2010) por exemplo, utiliza uma rede neural convolucional 3D, cuja entrada é composta por um conjunto de sete *frames*, centrados em torno do *frame* atual. Operações são realizadas sobre este conjunto, com o objetivo de extrair características espaciais de cada um deles. São calculados gradientes e fluxo óptico, além de utilizar a própria imagem em escala de cinza. A rede neural proposta possui uma camada totalmente conectada no final, responsável por classificar qual o gesto corresponde à sequência inserida na entrada da rede. O modelo é avaliado nos *datasets* KTH (LAPTEV; LINDEBERG,) e TRECVID 2008. Ambos conjuntos de dados possuem vídeos segmentados para cada uma das classes.

Também utilizando convolução 3D, (LIU; ZHANG; TIAN, 2016) optou por mapas de profundidade, por serem menos susceptíveis a variações de iluminação, e serem mais descritivos quanto a informações geométricas. Também foi utilizado o esqueleto associado ao mapa de profundidade. A CNN 3D é utilizada para extrair características espaço-temporais de uma sequência, que é normalizada para um tamanho fixo de 38 amostras, a fim de suportar sequência com diferentes durações. A partir da sequência de esqueletos referente a cada mapa de profundidade, são calculadas distâncias e ângulos entre as juntas, construindo um vetor de características ao longo do tempo. Ambos vetores de características provenientes dos mapas de profundidade e dos esqueletos são agregados em uma etapa final, seguida de uma etapa de classificação. O trabalho é avaliado em bases de dados com vídeos segmentados, sendo portanto, um trabalho de classificação.

De outra maneira, as **Redes com múltiplos caminhos** (do inglês, *Multiple stream networks*) incorporam informação temporal e espacial em diferentes caminhos de uma rede neural, agregando essas informações em camadas finais. (SIMONYAN; ZISSERMAN, 2014) foi um dos primeiros trabalhos a introduzir esta abordagem, ao propor uma rede neural com dois caminhos. Na Figura 5 está apresentada uma ilustração com a arquitetura proposta, na qual se pode observar os dois caminhos que a compõem: um deles é responsável por extrair características espaciais e tem como entrada uma única imagem, já o segundo caminho da rede recebe uma combinação de vários fluxos ópticos calculados a partir de uma sequência de imagens. Ambas informações dos caminhos são fundidas, utilizando dois métodos: calculando uma média, ou usando um SVM linear multi-classe.

Há outros trabalhos derivados do proposto em (SIMONYAN; ZISSERMAN, 2014), como (FEICHTENHOFER; PINZ; ZISSERMAN, 2016), em que uma arquitetura semelhante é proposta, mas com outras estratégias de fusão. Ou ainda, (CHENARLOGH; RAZZAZI, 2019) que testa três arquiteturas diferentes utilizando redes neurais convolucionais 3D. Duas das arquiteturas possuem múltiplos caminhos, nos quais os dados de entrada são compostos por fluxos ópticos nas direções x e y, incorporando informação temporal, além do gradiente também em ambas direções, trazendo informação espacial para a rede. Embora tais estratégias tragam vantagens ao incorporar informação temporal,

Figura 5 – Exemplo de arquitetura de uma rede neural convolucional com dois caminhos, para realizar o reconhecimento de ações.



Fonte: Produção do próprio autor.

Adaptado de (SIMONYAN; ZISSERMAN, 2014).

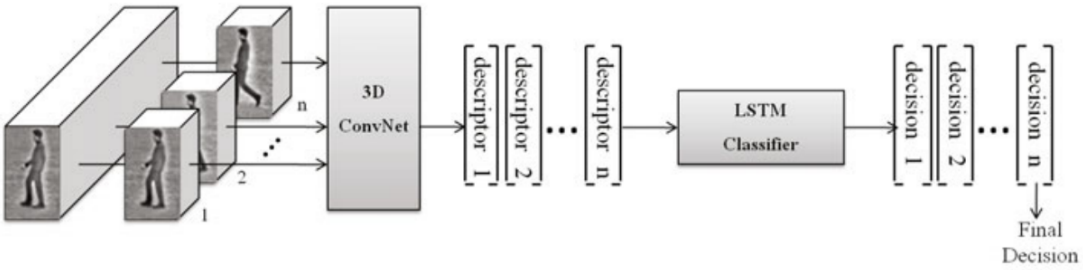
ainda possuem a limitação de serem pouco robustas à variação de escala, isto é, ao tempo e velocidade de execução de um gesto. Para aplicações em tempo real, esta característica faz diferença no desempenho de um detector de gestos.

As **Redes neurais recorrentes** (do inglês, *Recurrent neural networks*) são bastante exploradas em trabalhos na área de detecção de gestos e ações, não só com informações visuais mas também com dados de pose humana. Estas redes neurais possuem a capacidade de processar dados sequenciais, apresentando mecanismos de memória ao longo do tempo, gerando uma saída baseada não apenas no dado inserido em um determinado instante de tempo, mas sim a partir de uma combinação gerada pelos estados anteriores ao momento atual.

(BACCOUCHE et al., 2011) por exemplo, foi um dos primeiros trabalhos a usar redes recorrentes, utilizando uma sequência de imagens como entrada para classificar gestos. Neste trabalho foi proposta uma rede neural com uma etapa inicial formada por uma rede convolucional 3D. Esta é responsável por extrair características espaço-temporais de uma sequência de tamanho pré-definido, com 9 imagens cada. Tais características são inseridas em um classificador LSTM (do inglês, *Long short-term memory*), que é um tipo de rede neural recorrente. Na Figura 6 está apresentado o diagrama referente a esta arquitetura.

Pensando em uma execução em tempo real, o método proposto por (BACCOUCHE et al., 2011) apresenta alguns problemas. Embora possua uma arquitetura que a cada 9 imagens da sequência gere uma saída que corresponde a uma das classes presentes no *dataset*, uma decisão sobre qual ação foi executada só é tomada na última saída produzida por aquela sequência. Outro detalhe, é que o *dataset* utilizado, KTH (LAPTEV; LINDEBERG,), é construído por sequências compostas de execuções contendo apenas uma única classe de ação. Dessa maneira, o problema resolvido por (BACCOUCHE et al., 2011) consiste em classificar qual ação foi executada para uma dada sequência. Contudo, foi uma

Figura 6 – Arquitetura proposta em (BACCOUCHE et al., 2011) para reconhecer ações, utilizando uma rede neural composta de uma parte convolucional 3D, seguida de uma recorrente do tipo LSTM.



Fonte: Adaptado de (BACCOUCHE et al., 2011).

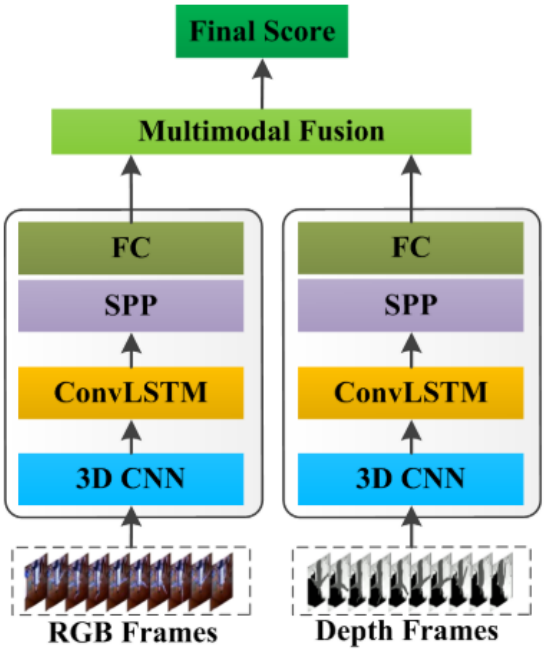
contribuição notável ao mostrar a utilização de redes recorrentes para o reconhecimento de sequências de movimentos humanos.

Na mesma linha que (BACCOUCHE et al., 2011), (ZHU et al., 2017a) utiliza uma rede neural convolucional 3D para extrair características, e então entregá-las a uma rede recorrente LSTM. São utilizadas imagens coloridas e mapas de profundidade como informações para classificar sequências de gestos, o que a faz uma técnica multimodal. Na Figura 7 encontra-se a arquitetura proposta, na qual se pode verificar a utilização de uma rede com dois caminhos iguais: um responsável pela extração de características de uma sequência de imagens coloridas, e a outra para os mapas de profundidade.

Uma vez que a primeira etapa da arquitetura proposta em (ZHU et al., 2017a) corresponde a uma CNN 3D, o dado de entrada deve corresponder a um tensor com 3 dimensões. Assim, uma sequência de imagens ou mapas de profundidade compõem os dados de entrada. Entretanto, como já mencionado aqui neste trabalho, gestos e ações correspondem a uma sequência de movimentos e, possuem diferentes velocidades de execução e tempos de duração, e, portanto, número de imagens diferentes. Para adequar esta característica a este tipo de arquitetura que recebe em sua entrada dados de dimensão fixa, algum tipo de normalização costuma ser feito para tornar o método robusto a variações de escala.

A alternativa mais simples seria analisar a taxa de amostragem dos dados de entrada, tomar como referência um tempo médio de execução, e escolher um quantidade fixa de amostras. Contudo, (ZHU et al., 2017a) realiza uma subamostragem em todas as sequências de gestos para um tamanho L fixo, além de utilizar uma estratégia para aumentar a base de dados, aplicando uma amostragem aleatória com pequenos deslocamentos na janela de tempo. Embora tenha apresentado uma abordagem voltada para gestos já segmentados presentes na base de dados IsoGD, o método proposto em (ZHU et al., 2017a) propôs um método capaz de receber como entrada execuções com diferentes tempos de duração.

Figura 7 – Arquitetura proposta em (ZHU et al., 2017a) para reconhecer gestos utilizando imagens e mapas de profundidade, com uma rede neural com dois caminhos, no qual cada um é composto de uma parte convolucional 3D seguida de uma recorrente do tipo LSTM.



Fonte: Adaptado de (ZHU et al., 2017a).

Os trabalhos que utilizam imagens coloridas e mapas de profundidade são muito comuns e estão apresentando bons resultados no reconhecimento de gestos e ações. Entretanto, devido ao fato de nos últimos anos diversos estudos, que tem como objetivo recuperar a pose humana terem avançado, houve também um crescimento nos trabalhos que utilizam a localização tridimensional de esqueletos para a compreensão do movimento humano.

2.2.2 Detectando gestos e ações a partir da pose humana

O uso de pontos estratégicos do corpo humano para a compreensão de seu movimento, teve seu primeiro registro no trabalho publicado em (JOHANSSON, 1973), em meado dos anos 1970. O experimento realizado ficou famoso ao utilizar fontes luminosas em pontos específicos do corpo humano, com o objetivo de entender seus movimentos no espaço a partir de padrões 2D. Este trabalho demonstrou que não apenas a quantidade de pontos utilizados, mas também a localização deles, influencia na compreensão do movimento humano. Mostrou-se também que esta é uma informação preciosa, que pode ser utilizada para o reconhecimento de padrões de movimento, aplicadas por exemplo à análise de atividades humanas. Uma observação interessante em (JOHANSSON, 1973), é que do ponto de vista mecânico, as juntas do corpo humano são pontos que conectam membros

de comprimento constante (PRESTI; CASCIA, 2016a).

Como mencionado na Seção 2.1, há um avanço mútuo no desenvolvimento de equipamentos e métodos utilizados para a aquisição de pose humana, e nos trabalhos que utilizam tal informação para algum propósito, como por exemplo, a detecção de gestos e atividades (XIA; CHEN; AGGARWAL, 2012; KEROLA; INOUE; SHINODA, 2015; KE et al., 2017). O avanço dos métodos de obtenção da pose humana trouxe não só melhoria em sua precisão, mas também na utilização de apenas imagens de câmeras comuns.

Assim como o uso de mapas de profundidade, o uso da pose humana para o reconhecimento de gestos e ações são se popularizou por diversos motivos. Além de corresponderem a uma quantidade menor de dados quando comparados com imagens coloridas, a informação contida é mais representativa. Isto é, movimentos e formas são mais expressivos se comparados com apenas imagens comuns. Entretanto, de acordo com (YAO et al., 2011), utilizar a pose humana para o reconhecimento de ações pode não ser tão simples assim. Isso porque, características extraídas que são semanticamente similares, podem não possuir semelhança numérica. Nesse sentido, um dos maiores desafios, em se utilizar a pose humana para o reconhecimento de gestos e ações, é em como representar o movimento humano a partir da localização das juntas ao longo do tempo.

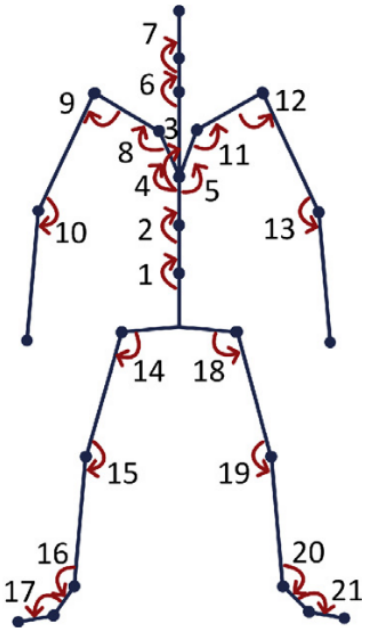
Desta maneira, as técnicas para reconhecer gestos utilizando a pose humana focam em representar o movimento humano, de forma que este seja invariante a diversos aspectos. Por exemplo, indivíduos possuem corpos com formatos diferentes, e a maneira na qual se executa o gesto pode variar entre indivíduos (PRESTI; CASCIA, 2016b), ou até, para o mesmo indivíduo. Estes fatores se aplicam tanto para a localização quanto para a classificação de gestos e ações.

Serão apresentadas algumas abordagens adotadas por alguns trabalhos, para pré-processar uma sequência de pose humana, visando torná-la normalizada em relação a diferentes indivíduos e maneiras de execução. Diversas métricas e estratégias são utilizadas: cálculo de distâncias e ângulos entre as mais variadas combinações de juntas, velocidades e acelerações, sem contar com as diversas normalizações adotadas.

Em (OFLI et al., 2014) por exemplo, o ângulo entre cada par de membros conectados são calculados. Estes ângulos compõem séries temporais, utilizadas para determinar quais são as juntas mais significativas do modelo de corpo humano adotado para a tarefa de classificar ações. Na Figura 8 estão ilustrados os ângulos utilizados.

Normalizações também são comuns, como por exemplo, em (WEI et al., 2013), as poses são normalizadas alinhando-se torsos e ombros, enquanto que em (VEMULAPALLI; ARRATE; CHELLAPPA, 2014), uma mudança de referencial é feita para que todas as poses possam ser comparadas entre si. Já em (HUSSEIN et al., 2013), as distâncias entre as juntas são normalizadas baseando-se na distância média obtida a partir da base de dados

Figura 8 – Ângulos calculados a partir das juntas de esqueleto, utilizados na representação adotada por (OFLI et al., 2014), para reconhecer ações humanas.



Fonte: Adaptado de (OFLI et al., 2014).

utilizada. Ou ainda, (ZANFIR MARIUS LEORDEANU, 2013) suaviza a pose com um filtro gaussiano de tamanho 5, ao longo do tempo, após realizar sua normalização (PRESTI; CASCIA, 2016b).

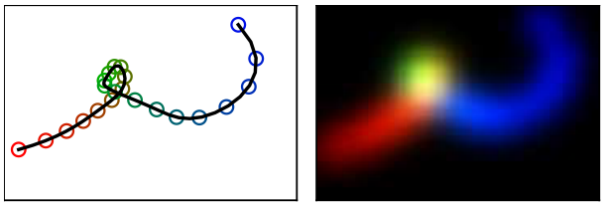
A manipulação que é feita na pose humana, vai de acordo com o método utilizado para a detecção do gesto ou ação. Em (KE et al., 2017), o esqueleto é dividido em 5 partes, correspondendo aos membros inferiores e superiores, e ao tronco. São então calculadas duas métricas em cada uma das partes, para então, formar imagens que serão em seguida inseridas em uma CNN para a extração de características. Nestas imagens, na direção horizontal estão os valores calculados da métrica para um instante de tempo, e na direção vertical a métrica é calculada para cada instante da sequência. A primeira métrica, CD (do inglês, *Cosine Distance*), corresponde ao cosseno entre dois vetores formados por pares de juntas, e NM (do inglês, *Normalized Magnitude*), que representa uma distância entre duas juntas, normalizada em relação à norma de um vetor de referência.

Para que o método seja invariante à duração de execução, as imagens geradas são redimensionadas para um tamanho fixo. Características de maior nível são extraídas utilizando a CNN, para que então a ação seja classificada. Esta é uma abordagem muito comum, e muitas vezes, CNNs já treinadas em outras bases de dados são utilizadas como uma maneira de inicializar os pesos, tornando o processo de treinamento mais rápido. Essa técnica é conhecida como *transfer learning*.

Como apresentado em (KE et al., 2017), a informação temporal é incorporada

é representada em uma das dimensões da imagem gerada. Já no trabalho proposto por (CHOUTAS et al., 2018), a trajetória de cada uma das juntas é usada parar criar uma imagem, em que o rastro da posição da junta na trajetória vai mudando de cor com o passar do tempo. Um exemplo pode ser observado na Figura 9. Uma CNN é utilizada com o conjunto de imagens geradas como entrada, para classificar a ação executada no clipe.

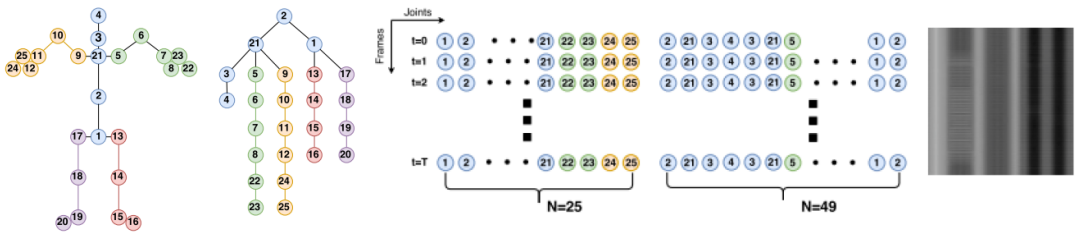
Figura 9 – Representação utilizada em (CHOUTAS et al., 2018) para descrever o movimento de uma junta ao longo do tempo.



Fonte: Adaptado de (CHOUTAS et al., 2018).

Também utilizando CNNs com imagens geradas a partir da sequência de pose humana, (YANG et al., 2018) crítica que as representações utilizadas na maioria dos trabalhos não levam em consideração a ordem em que as juntas estão conectadas, desprezando assim informação semântica. Além disso, ele discute que nem todas as juntas são relevantes, e introduz a utilização de mecanismos conhecidos como *attention*, que tornam a rede neural capaz de aprender as informações mais relevantes. (YANG et al., 2018) considera então o esqueleto como uma árvore, e percorre as juntas utilizando o algoritmo de busca por profundidade DFS (do inglês, *Depth-first Search*) para determinar a ordem na qual percorrerá as juntas. Esta ordem é utilizada para gerar a representação em forma de uma imagem da sequência do movimento. Na Figura 10 pode-se verificar, da esquerda para a direita, a representação do esqueleto, a árvore referente à este com a ordem de visitação das juntas e, em seguida, a maneira na qual a imagem é montada, considerando-se na direção horizontal as juntas, e na vertical as amostras ao longo do tempo. Por fim, pode-se ver na imagem em escala de cinza que foi gerada e que será entregue a uma CNN para classificação.

Figura 10 – Metodologia utilizada por (YANG et al., 2018) para representar uma sequência de pose humana, levando em consideração a estrutura semântica do esqueleto.



Fonte: Adaptado de (YANG et al., 2018).

Redes neurais recorrentes também são utilizadas juntamente com a pose humana para reconhecer gestos. Embora (YANG et al., 2018) afirme que RNNs são pouco eficientes em aprender relações espaciais entre as juntas, (LIU et al., 2018) propõe utilizar este tipo de rede neural para reconhecer ações. (LIU et al., 2018) cita inclusive, que nesta área de pesquisa, redes neurais recorrentes são comumente adotadas para tentar modelar relações temporais do movimento humano. Contudo, ele propõe um *framework* que, baseado em uma estrutura de árvore relacionada com o modelo de esqueleto, explora as relações cinemáticas entre as juntas, modelando as relações espaciais da pose humana. Assim, ele utiliza módulos LSTM para extrair características espaço-temporais de uma sequência de esqueletos.

Os trabalhos mencionados nesta seção utilizam dados de pose humana pra classificar gestos ou ações. Mesmo que a pose humana seja considerada um dado mais simples de se trabalhar, por ser mais representativa para o movimento humano, foi possível verificar a importância da representação desta ao longo do tempo. Contudo, assim como apresentado na Seção 2.2.1, estes trabalhos consistem em classificar uma sequência, isto é, não realizam a localização do gesto ou ação, o que é fundamental para uma aplicação em tempo real.

2.2.3 Localizando e detectando gestos e ações a partir da pose humana

Grande parte dos estudos na área de reconhecimento de gestos se concentram somente na extração de características, e na classificação dos gestos a partir de sequências já segmentadas (ZHANG et al., 2019; SHARAF et al.,). A localização, ou como também mencionada como detecção do gesto, é abordada tanto em conjunto com trabalhos de classificação de gestos ou ações, quanto de maneira separada.

As bases de dados existentes nem sempre possibilitam o emprego de técnicas para a localização de gestos e ações, uma vez que, para que isso seja possível, é necessário que esta seja rotulada com os instantes de início e fim do gesto. (ESCALERA; ATHITSOS; GUYON, 2016) apresenta uma coletânea com diversas base de dados na área de localização e classificação de gestos, evidenciando características comuns entres os *datasets*, sendo uma delas se este é ou não segmentado.

Assim como nos trabalhos que realizam a classificação de gestos utilizando esqueletos, aqueles que realizam a localização buscam uma maneira de representar a pose humana. Em (ZANFIR MARIUS LEORDEANU, 2013) por exemplo, uma de suas principais colaborações foi propor um descritor de pose, baseado em características espaciais e temporais, utilizando uma pequena janela de tempo em torno do instante atual. Este descritor é testado em cenários em que as sequências já estão segmentadas, realizando portanto apenas a classificação.

Entretanto, esse mesmo descritor foi testado em vídeos não segmentados. Nesse caso,

o método proposto por (ZANFIR MARIUS LEORDEANU, 2013) é modificado, e passa a utilizar uma abordagem com janela deslizante. O tamanho W da janela é obtido através de um processo de aprendizagem. Dentro desta janela, ocorre uma votação utilizando a probabilidade obtida em cada instante de tempo, para definir a qual classe aquele instante de tempo pertence. Os segmentos do vídeo (onde se consideram que há gestos) são obtidos ao final desse processo a partir dos instantes que possuíram a mesma classificação. Os intervalos de execução com até 5 *frames* são descartados. Os testes foram realizados em sequências já segmentadas de um *dataset*, que foram concatenadas para gerar um vídeo contínuo com múltiplas sequências, e em uma base de dados criada por eles para tal finalidade. Neste cenário, o método é avaliado da mesma maneira feita com sequências segmentadas: acurácia de classificação. Portanto, (ZANFIR MARIUS LEORDEANU, 2013) não estava preocupado em determinar os instantes de início e fim, mas sim em segmentar e gerar uma classificação para aquele intervalo.

(SHARAF et al., 2015) também realiza localização e classificação de ações, utilizando um classificador do tipo SVM, cuja entrada recebe um descritor composto por ângulos e velocidades angulares, calculados a partir da localização 3D das juntas. O descritor é calculado em múltiplas escalas, a fim de adicionar informação temporal e ser robusto à variação de tempos de execução distintos. A probabilidade obtida, para diferentes escalas, é feita interpolando seus picos de detecção. O resultado disso é um instante no qual ocorreu uma determinada ação. Semelhante a (ZANFIR MARIUS LEORDEANU, 2013), a localização é feita, mas apenas com o intuito de se classificar, sem determinar quais são os instantes de início e fim.

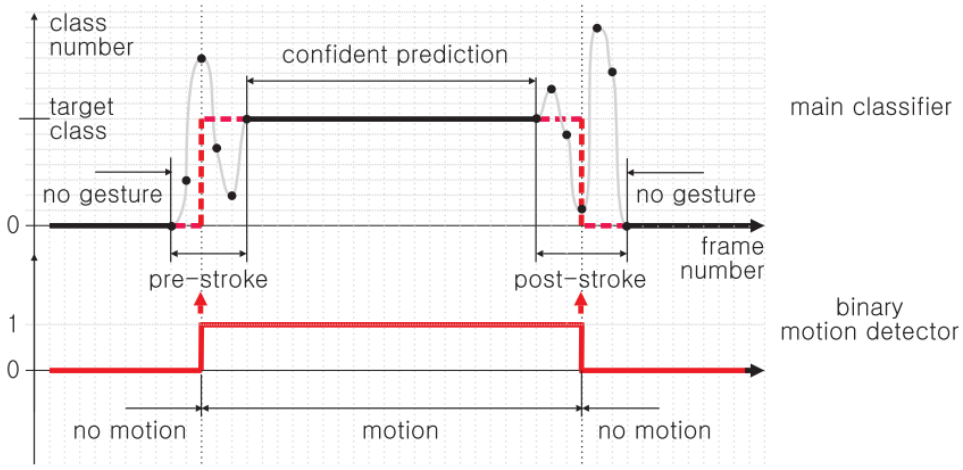
(PATRONA et al., 2018) propõe um *framework* para a detecção e reconhecimento em tempo-real de ações. Este utiliza apenas informações de movimento das juntas de esqueleto além de ângulos, dos quais extrai informações que caracterizam o movimento. Cada característica extraída é chamada de *gesturelets* e o conjunto de todos estes descritores compõe uma *Bag of gesturelets*. É computado então o histograma dessa representação, e esta informação é utilizada para se treinar um classificador linear binário para cada uma das ações que compõe o *dataset* utilizado. Os vetores obtido a partir de cada classificador de uma ação são então agregados com uma soma ponderada. Realiza-se então a detecção do início da ação no momento em que esta soma ponderada ultrapassa um determinado limiar obtido durante o treinamento. O fim do gesto é detectado no momento em que valores negativos ocorrem após o evento de início de uma ação.

(NEVEROVA et al., 2014; NEVEROVA et al., 2016) apresentam um método capaz de localizar e classificar gestos capturados utilizando um sensor do tipo *Kinect*. O *dataset* utilizado faz parte do desafio *ChaLearn Looking at People Challenge 2014* (ESCALERA et al., 2015). Este é formado por um conjunto de vídeos que contém execuções de gestos emblemáticos da língua italiana. Em um mesmo vídeo há várias execuções de diferentes

gestos e, portanto, não são segmentados. As rotulações desta base de dados consistem em um conjunto de instantes que indicam o início e o fim da execução de um gesto, além da classe do gesto. O desempenho deste desafio é avaliado utilizando-se o índice de *Jaccard*. Este índice corresponde à razão entre a interseção e a união da janela na qual o gesto foi detectado, e aquela na qual este realmente foi executado, ou seja, o *ground-truth* do *dataset*. Para um índice igual a 1, significa que o instante de início e fim do gesto foi detectado com sucesso, e sua classificação foi feita corretamente.

Em seu trabalho, (NEVEROVA et al., 2014; NEVEROVA et al., 2016) propõe uma abordagem multi-modal, utilizando imagens coloridas, mapas de profundidade, esqueletos e áudio disponíveis na base de dados. A localização dos gestos é feita separadamente, aplicando-se uma rede neural do tipo MLP, utilizando apenas a pose humana para tal finalidade. Métricas são calculadas a partir da pose para então inseri-las no classificador, responsável por determinar se um determinado instante corresponde a repouso ou movimento. Na Figura 11, há uma ilustração hipotética da saída do classificador binário utilizado para localizar os gestos. A saída desejada do localizador de gestos também é apresentada, e corresponde a uma janela de tempo no qual ocorreu o movimento da execução do gesto.

Figura 11 – Ilustração da saída do classificar binário utilizado em (NEVEROVA et al., 2014) para localizar gestos, evidenciando as fases de transição deste.



Fonte: Adaptado de (NEVEROVA et al., 2014).

(NEVEROVA et al., 2014) também comenta que, a utilização de janelas deslizantes para resolver o problema de localização de gestos pode causar problemas, como detecções ruidosas das fases de execução do gesto, ou até mesmo, a sobreposição de diversas instâncias de gestos. Com a abordagem escolhida, (NEVEROVA et al., 2014) atinge uma precisão de 98% em classificar os instantes de repouso ou movimento, e um índice de *Jaccard* igual a 0,8500, ao executar todo o processo incluindo localização e classificação dos gestos.

De maneira semelhante ao apresentado por (NEVEROVA et al., 2014; NEVEROVA et al., 2016), (ZHU et al., 2019) propõe um método para continuamente segmentar e classificar gestos. Um módulo é responsável por realizar a segmentação, utilizando imagens coloridas e mapas de profundidade. Este é construído com convolução 3D e blocos recorrentes do tipo LSTM, para a extração de características espaço-temporais. A partir da saída deste módulo, são determinados o início e o fim de um gesto. A sequência então segmentada é entregue a um segundo módulo responsável pela classificação do gesto, composto também por convolução 3D e LSTM. Este segundo módulo tem como entrada imagens coloridas, mapa de profundidade e fluxo óptico.

(ZHU et al., 2019) avalia seu método no *dataset* do desafio *Chalearn LAP ConGD*. Esta base de dados, diferente das citadas anteriormente nesta seção, é composta de vídeos em que os indivíduos executam diversos gestos em sequência, isto é, os vídeos não estão segmentados e possuem diferentes classes de gestos. O desempenho do algoritmo também é avaliado utilizando-se o índice de *Jaccard*.

Diversos trabalhos além dos apresentados nesta subseção também utilizam a pose humana para, de alguma maneira, localizar gestos ou ações (MONNIER; GERMAN; OST, 2015; JOSHI et al., 2017; LIU et al., 2019). É possível verificar que, alguns destes trabalhos realizam localização e classificação em uma única etapa e, portanto, não se determinam os instantes de início e fim da execução. De outra maneira, há trabalhos que possuem uma etapa responsável exclusivamente pela localização, para que então seja enviada a sequência já segmentada para a classificação. Independente da abordagem adotada, em geral, o que esses trabalhos possuem em comum é que eles realizam algum pré-processamento na pose humana, calculando ângulos, velocidades, acelerações, dentre outras métricas. Ambas abordagens apresentam bons resultados, e são passíveis de serem executadas em tempo real. O atual cenário, em que esses métodos se encontram, está diretamente relacionado com o recente avanço das técnicas de obtenção da pose humana.

3 Estimativa da Localização Tridimensional de Juntas de Esqueletos

Este capítulo apresenta a metodologia proposta neste trabalho para a obtenção da localização tridimensional de juntas de esqueletos. Esta consiste na etapa inicial de um sistema que foi desenvolvido para localizar e classificar gestos em tempo real. Primeiramente, será descrita a parte teórica do método, além de uma discussão de aspectos de sua implementação, uma vez que um dos objetivos do desenvolvimento deste é que funcione em um sistema real, com suas devidas restrições de tempo e *hardware*. Os resultados obtidos com o objetivo de validar o método, bem como as devidas discussões, estão apresentadas no fim deste capítulo.

3.1 Descrição do método

O método aqui proposto reconstrói as coordenadas tridimensionais de juntas dos esqueletos de indivíduos, utilizando um sistema multicâmeras calibrado. Todo o processo pode ser subdividido em quatro etapas: (1) detecção de esqueletos em todas as imagens do sistema multicâmera; (2) busca de correspondências, resultando em um conjunto de pares de detecções que provavelmente corresponderão ao mesmo indivíduo; (3) agrupamento de correspondências, que consiste em juntar os pares de correspondências que possuem detecções em comum; e, finalmente (4) a reconstrução tridimensional das juntas para cada grupo de correspondências obtido.

3.1.1 Detecção de esqueletos nas imagens

Esta primeira etapa consiste em detectar esqueletos nas imagens do sistema multicâmera. Neste trabalho foi utilizado um método conhecido como *OpenPose* (WEI et al., 2016). Este método é capaz de detectar juntas do corpo, mãos, e pontos na face de mais de um indivíduo na mesma imagem, podendo chegar a um total de 135 pontos de interesse. Contudo, aqui, só foi utilizada a detecção de juntas do corpo. A implementação [1] oficial utilizada oferece três modelos diferentes para o corpo, podendo ser composto por 15, 18 ou 25 juntas. Para garantir portabilidade com o modelo utilizado por outro banco de dados na segunda parte deste trabalho, optou-se pelo modelo contendo 18 juntas, apresentado na Figura 2b.

[1] https://github.com/CMU-Perceptual-Computing-Lab/openpose/tree/v1.4.0

Essa implementação possibilita executar as detecções em GPU (*Graphics Processing Unit*, do inglês Unidade de Processamento Gráfico), podendo atingir taxas maiores que 10 *fps* dependendo do ajuste de parâmetros feito. Dois parâmetros que influenciam o tempo de processamento são a resolução da imagem na entrada da rede, e o fator de escala aplicado à saída de rede, uma vez que esta é pós-processada a fim de se obter as coordenadas das juntas de cada esqueleto. Mais detalhes do ajuste desses parâmetros serão mostrados nos resultados juntamente de seus respectivos tempos de execução.

3.1.2 Busca de correspondências

Com as detecções em todas imagens, a próxima etapa consiste em buscar detecções que são correspondentes, ou seja, correspondem ao mesmo indivíduo, porém, capturado por câmeras diferentes. Cada detecção de esqueleto é constituída por um conjunto de até 18 juntas devidamente identificadas e com coordenadas $(u,\ v)$ na imagem da câmera na qual foi detectada. Vale frisar que mesmo o detector sendo capaz, em algumas situações, de estimar a localização de juntas ocultas, nem sempre todas elas são detectadas. Além disso, cada uma das juntas detectadas possui uma confiança associada, que pode ser utilizada para filtrar juntas, permanecendo apenas com aquelas que possuam um determinado nível de confiança.

Em seguida, para cada detecção de esqueleto, busca-se um possível correspondente nas detecções obtidas pelas outras câmeras. Para uma dada imagem, o seu conjunto de detecções pode ou não resultar em um correspondente, uma vez que podem haver oclusões e um determinado indivíduo pode não ser visto por uma ou mais câmeras. Ou ainda, a correspondência pode não atender aos critérios estabelecidos nesta etapa do método proposto.

A busca por correspondências é realizada utilizando geometria epipolar. Para melhor descrição, nas Figuras 13, 14 e 15, estão representadas as imagens de um par de câmeras, C_0 e C_1, e dois indivíduos, P_1 e P_2. Os esqueletos nas figuras são versões simplificadas com apenas sete juntas ao invés das 18 utilizadas no trabalho. O indivíduo P_2 não aparece na imagem da câmera C_0, pois está ocluso pelo indivíduo P_1. Já na câmera C_1, os dois indivíduos estão presentes na imagem. Portanto, há três esqueletos identificados nas duas imagens e seus indivíduos correspondentes: $\{(A, P_1),\ (B, P_1),\ (C, P_2)\}$. A Figura 12 traz uma legenda para melhor entendimento do exemplo.

A Figura 13 mostra o processo de avaliação do esqueleto detectado A, visto pela câmera C_0, ou seja, é realizada uma busca de correspondentes nas detecções das outras câmeras. Nesse exemplo, na imagem da câmera C_1, há dois esqueletos, B e C. As juntas marcadas com quadrados verdes correspondem as que foram identificadas nas três detecções. Utiliza-se então a projeção dos pontos das juntas identificadas $\mathbf{m}^A_{1(0)}$, $\mathbf{m}^A_{2(0)}$ e $\mathbf{m}^A_{3(0)}$ do esqueleto A, para obter as linhas epipolares na imagem da câmera C_1. Observe que na

Figura 12 – Legenda para as Figuras 13, 14 e 15.

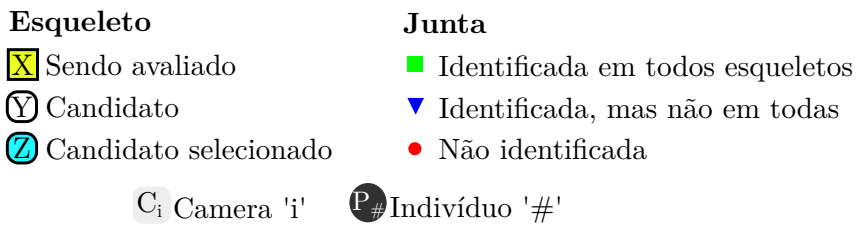
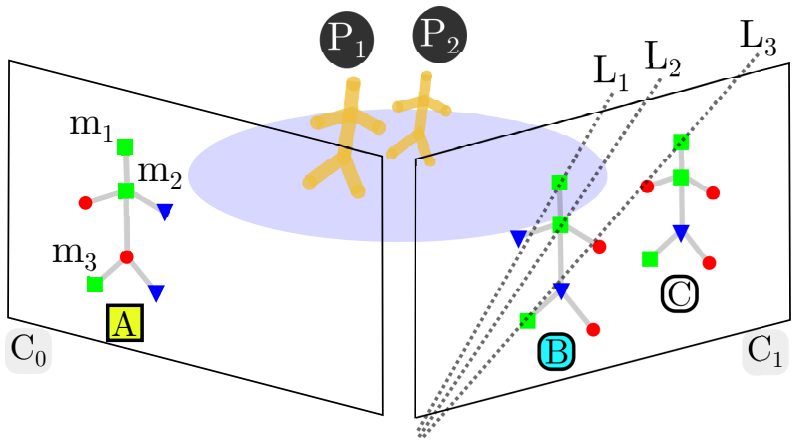


Figura 13 a notação das variáveis foi simplificada para não poluir a imagem. O mesmo foi feito nas Figuras 14 e 15.

Figura 13 – Exemplo do método proposto avaliando o esqueleto A.



Portanto, de modo geral, para cada ponto $\mathbf{m}^j_{k(o)}$ pertencente à câmera de origem C_o, obtém-se uma linha epipolar $L^j_{k(d)}$ na câmera de destino C_d calculada por

$$L^j_{k(d)} = \mathbf{F}^d_o \, \tilde{\mathbf{m}}^j_{k(o)} = \mathbf{F}^d_o \, \left[u^j_{k(o)},\ v^j_{k(o)},\ 1\right]^T, \tag{3.1}$$

em que $u^j_{k(o)}$ e $v^j_{k(o)}$ são as coordenadas do k-ésimo[2] ponto $\mathbf{m}^j_{k(o)}$ do j-ésimo esqueleto na imagem de câmera C_o; $L^j_{k(d)}$ é a linha epipolar na imagem da câmera de destino C_d, correspondente ao ponto $\mathbf{m}^j_{k(o)}$, sendo $\tilde{\mathbf{m}}^j_{k(o)}$ sua representação em coordenadas homogêneas; e $\mathbf{F}^d_o$ é a matriz fundamental que associa pontos na imagem da câmera C_o com suas linhas epipolares na imagem de C_d, calculada com os parâmetros de calibração intrínsecos e extrínsecos das câmeras como

$$\mathbf{F}^d_o = \mathbf{K}_d^{-T} \widehat{\mathbf{t}^d_o \mathbf{R}^d_o} \mathbf{K}_o{}^{-1}, \tag{3.2}$$

onde $\mathbf{K}_o$ e $\mathbf{K}_d$ são as matrizes de parâmetros intrínsecos das câmeras C_o e C_d, respectivamente; e $\widehat{\mathbf{t}^d_o \mathbf{R}^d_o}$ corresponde ao produto vetorial entre o vetor de translação $\mathbf{t}^d_o = [t_1,\ t_2,\ t_3]^T$

[2] O índice k é utilizado para identificar cada junta do modelo de juntas utilizado. Podem haver juntas que se repetem em modelos diferentes. Estas, por sua vez, terão o mesmo índice independente do modelo. Este é, além de um detalhe de notação, mas também de implementação, uma vez que o sistema proposto é independente do modelo de juntas utilizado.

e a matriz de rotação $\mathbf{R}_o^d$. O par $\mathbf{t}_o^d$ e $\mathbf{R}_o^d$ levam um ponto no referencial da câmera C_o para o referencial de C_d. Esse produto vetorial pode ser escrito como um produto entre a matriz antissimétrica do vetor de translação e a matriz de rotação. Com as linhas epipolares correspondentes aos pontos $\mathbf{m}_{1(o)}^j$, $\mathbf{m}_{2(o)}^j$ e $\mathbf{m}_{3(o)}^j$ da imagem de C_o projetadas na imagem de C_d, calcula-se, para cada esqueleto candidato na imagem, a distância de cada junta à sua linha correspondente. Depois, para cada esqueleto, calcula-se a média das distâncias das juntas às linhas epipolares da seguinte forma:

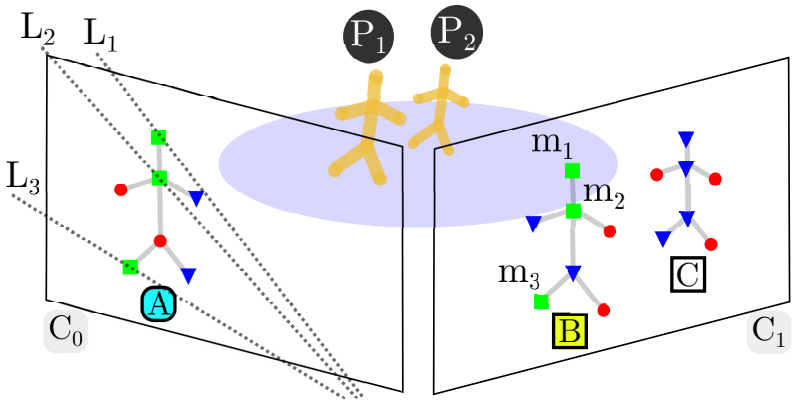
$$\bar{d}_{j,l} = \frac{\sum_k \left\langle \mathbf{m}_{k(o)}^l,\ L_{k(d)}^j \right\rangle}{N}, \tag{3.3}$$

em que o índice j representa o esqueleto avaliado na imagem da câmera de origem C_o; l representa o esqueleto na imagem da câmera C_d, candidato a correspondente de j; $k = 1, \cdots, N$; e N corresponde à quantidade de pontos do esqueleto j também identificados em l. Após todos os esqueletos candidatos da imagem de C_d serem avaliados, toma-se a menor distância euclidiana média encontrada. Se esta for menor que um limiar d_{max}, assume-se que o esqueleto associado a essa distância é o correspondente ao que está sendo avaliado na imagem de C_o.

Tomando como exemplo a Figura 13, se a distância média do esqueleto B for menor que a do esqueleto C e menor que o limiar d_{max}, o par de esqueletos (A, B) é adicionado ao conjunto de correspondências, i.e., $\mathcal{Q} = \{(A, B)\}$.

Ainda seguindo o exemplo apresentado, uma vez que a imagem da câmera C_0 possui apenas um esqueleto, o algoritmo passa a analisar os esqueletos da imagem de C_1, como ilustrado na Figura 14. O próximo esqueleto avaliado é, então, o esqueleto B. Nessa situação, só existe um candidato presente na imagem da câmera C_0, que é o esqueleto A. Caso seja atendida a condição de $\bar{d}_{B,A} < d_{max}$, adiciona-se (B, A) ao conjunto de correspondências: $\mathcal{Q} = \{(A, B),\ (B, A)\}$.

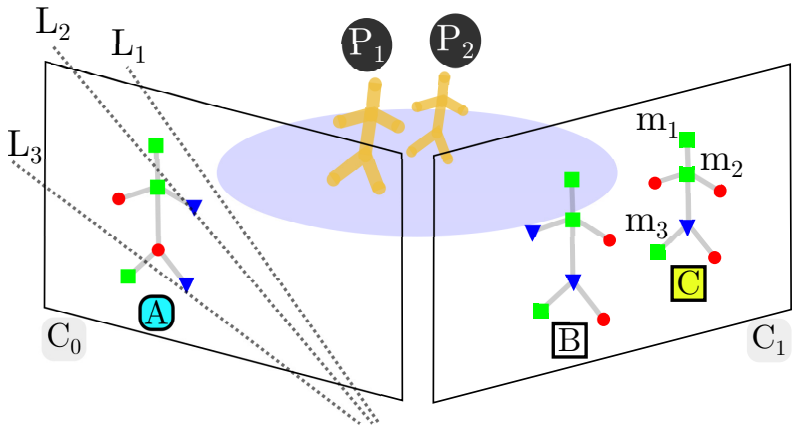
Figura 14 – Exemplo do método proposto avaliando o esqueleto B.



Por fim, avalia-se o esqueleto C que possui apenas um candidato na imagem de

C_0, como observado na Figura 15. Supondo que a condição para a distância média seja atendida, mesmo que o par (C, A) seja uma falsa correspondência, este será adicionado ao conjunto: $\mathcal{Q} = \{(A, B),\ (B, A),\ (C, A)\}$.

Figura 15 – Exemplo do método proposto avaliando o esqueleto C.



O próximo passo, é eliminar correspondências que são potencialmente falsas. Para isso, é analisado se existem pares que possuam correspondentes repetidos para diferentes esqueletos na imagem de uma mesma câmera. Por exemplo, para os esqueletos da câmera C_1, há dois pares cujo o correspondente é o esqueleto A, que são (B, A) e (C, A). Como isso não é possível de ocorrer com apenas um par de câmeras, mantém-se apenas o par que possua o menor erro médio $\bar{d}$.

Assim, prováveis falsas correspondências são eliminadas, como o caso (C, A), que representava uma correspondência do indivíduo P_2, visto por C_1, com o indivíduo P_1, visto por C_0. Isso ocorre, pois no campo de visão de C_0, o indivíduo P_2 está ocluso por P_1. Portanto, apenas o indivíduo P_1 será identificado nessa configuração de câmeras.

Após terem sido buscados possíveis correspondentes para todas as detecções, realiza-se o agrupamento destas, o que será detalhado na próxima seção.

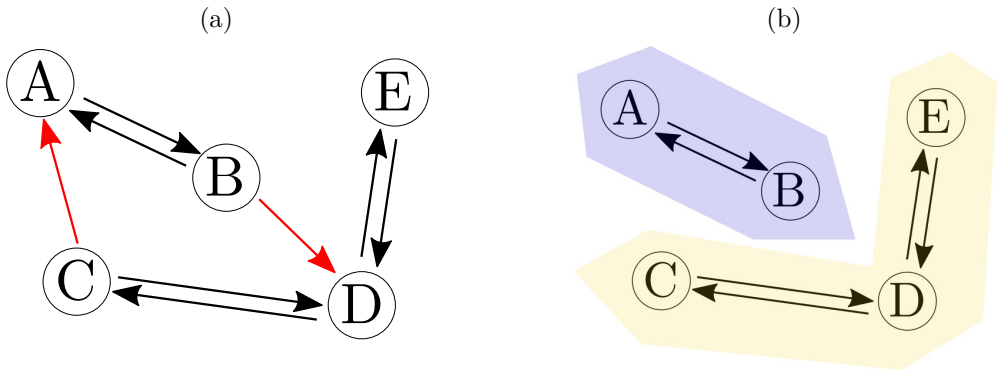
3.1.3 Agrupamento de correspondências

A etapa anterior resulta em um conjunto de correspondências. Cada correspondência é composta por um par de detecções em câmeras diferentes. Nesse conjunto podem haver pares que possuem detecções em comum. Portanto, esses pares podem ser reunidos em um único conjunto, uma vez que correspondem a detecções do mesmo indivíduo. Para realizar o processo de agrupamento desses pares, o conjunto de correspondências pode ser representado como um grafo no qual os vértices correspondem a cada detecção de esqueletos nas imagens das câmeras, e as arestas são as correspondências encontradas.

Para ilustrar isto, na Figura 16a, está um grafo que representa as correspondências encontradas de um grupo de cinco detecções. Não foi utilizado o exemplo da seção anterior,

com apenas 2 indivíduos e 3 detecções, pois este era muito simples, um vez que não possuía pares de correspondências para serem agrupados. No exemplo então citado, as arestas são representadas por setas, que apontam do vértice que representa a detecção de referência para o vértice que foi considerado como uma correspondência. Como o processo de busca de correspondências é feito para todos os vértices, podem haver setas nas duas direções. No caso das arestas (C, A) e (B, D) (destacadas em vermelho), a correspondência ocorre apenas em uma direção. Estas são removidas antes do processo de agrupamento de correspondências, como explicado na sessão anterior. Com isso, temos um grafo não-dirigido, ou seja todos as arestas são antiparalelas, e, portanto, o agrupamento de correspondências pode ser realizado buscando-se componentes conectados deste grafo.

Figura 16 – Representação em forma de grafo das correspondências obtidas no processo de busca de correspondências. Em (a), detecções e correspondências são respectivamente vértices e arestas do grafo. Enquanto em (b), estão destacados os grupos obtidos a partir de busca de componentes conectados aplicado ao grafo.



Para este caso, foi utilizado o algoritmo DFS (*Depth-first search*, do inglês Busca em profundidade-primeiro) para obter os componentes conectados. Escolhe-se então um vértice qualquer para se iniciar o algoritmo, e identificam-se os vértices já visitados. Quando todos vértices forem visitados, um grupo de componentes conectados é obtido. Se ainda houver vértices não visitados, outro é escolhido e se reexecuta o algoritmo até que todos sejam visitados. Dessa maneira, obtém-se os grupos de vértices que possuem conexões, ou seja, as correspondências que possuem detecções em comum. Para cada grupo de componentes conectados, será realizada a próxima etapa que corresponde à reconstrução tridimensional das juntas. Na Figura 16b, podem ser verificados os dois grupos obtidos após a execução do algoritmo.

3.1.4 Reconstrução tridimensional das juntas

A partir do conjunto das correspondências agrupadas, realiza-se o processo de reconstrução tridimensional de cada junta. Como não há informação tridimensional *à priori* das juntas, com o método utilizado, só é possível efetuar a reconstrução se houver

juntas detectadas em pelo menos duas câmeras. Portanto, para cada junta, verifica-se em quais câmeras ela foi detectada, e então utilizam-se suas coordenadas em cada imagem para o processo de reconstrução.

Para isso, considera-se o modelo de câmera *pinhole* conforme Equação 3.4, na qual λ_i corresponde a um fator de escala; $\tilde{\mathbf{m}}_i = [u_i,\ v_i,\ 1]^T$ é um ponto na imagem da câmera i; $\mathbf{K}_i$ é a matriz de parâmetros intrínsecos; Π a matriz de projeção; $[\mathbf{R}_i, \mathbf{T}_i]$ a matriz de parâmetros extrínsecos, composta, respectivamente, por uma rotação e uma translação; e, por fim, $\tilde{\mathbf{M}} = [x,\ y,\ z,\ 1]^T$ corresponde ao ponto tridimensional, no referencial global no qual as câmeras foram calibradas, que gera as projeções $\tilde{\mathbf{m}}_i$ em cada imagem. O subíndice i tem objetivo de diferenciar as câmeras, e as variáveis indicadas com $\tilde{}$ estão representadas em coordenadas homogêneas.

$$\lambda_i \tilde{\mathbf{m}}_i = \mathbf{K}_i\ \Pi\ [\mathbf{R}_i, \mathbf{T}_i]\ \tilde{\mathbf{M}} \tag{3.4}$$

Pode-se então, para cada junta, com seus respectivos pontos $\tilde{\mathbf{m}}_i$ detectados em no mínimo duas câmeras, montar um sistema de equações para determinar o ponto $\tilde{\mathbf{M}}$, que representa a posição tridimensional da junta em questão. A Equação 3.4 é reescrita na forma da Equação 3.5, na qual as incógnitas são λ_i e $\mathbf{M} = [x,\ y,\ z]^T$.

$$\lambda_i \tilde{\mathbf{m}}_i = \mathbf{K}_i(\mathbf{R}_i\mathbf{M} + \mathbf{T}_i) \tag{3.5}$$

Manipulando-se a Equação 3.5 para que fique na forma de um sistema de equações, tem-se

$$\lambda_i(\mathbf{K}_i\mathbf{R}_i)^{-1}\tilde{\mathbf{m}}_i - \mathbf{M} = \mathbf{R}_i^{-1}\mathbf{T}_i, \tag{3.6}$$

que pode ser escrita matricialmente como mostrado na Equação 3.7, em que $\mathbf{I}$ é uma matriz identidade 3×3

$$\begin{bmatrix} -\mathbf{I} & (\mathbf{K}_i\mathbf{R}_i)^{-1}\tilde{\mathbf{m}}_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \lambda_i \end{bmatrix} = \mathbf{R}_i^{-1}\mathbf{T}_i. \tag{3.7}$$

Generalizando para duas ou mais câmeras, adicionam-se fatores de escala referentes a cada uma delas como variáveis do sistema de equações, uma vez que o vetor $\mathbf{M}$ é o mesmo para todas as câmeras. Observe que cada câmera que detecta a junta fornece três equações para resolver as coordenadas do ponto tridimensional $\mathbf{M}$ e o fator de escala λ_i. Logo, tem-se a Equação 3.8, que representa o sistema de equações na forma matricial, onde

$(\mathbf{K}_i\mathbf{R}_i)^{-1}\tilde{\mathbf{m}}_i$ é representado por $\mathbf{W}_i$, e $\mathbf{0}_{m\times n}$ é uma matriz de zeros de dimensão $m \times n$.

$$\begin{bmatrix} -\mathbf{I} & \mathbf{W}_1 & \mathbf{0}_{3\times n-1} & \\ \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I} & \mathbf{0}_{3\times i-1} & \mathbf{W}_i & \mathbf{0}_{3\times n-i} \\ \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I} & & \mathbf{0}_{3\times n-1} & \mathbf{W}_n \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \lambda_1 \\ \vdots \\ \lambda_i \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^{-1}\mathbf{T}_1 \\ \vdots \\ \mathbf{R}_i^{-1}\mathbf{T}_i \\ \vdots \\ \mathbf{R}_n^{-1}\mathbf{T}_n \end{bmatrix} \tag{3.8}$$

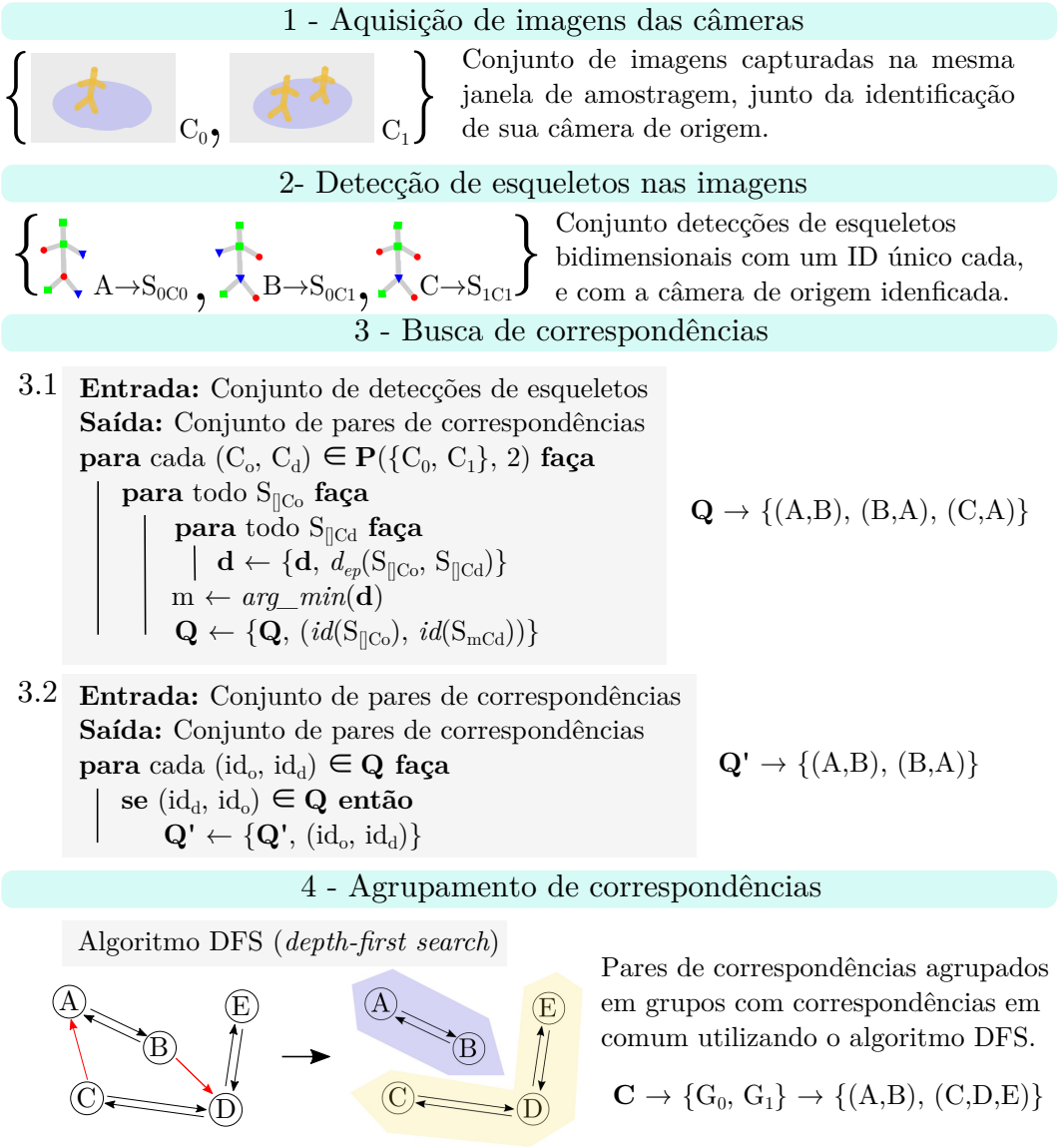
Resolvendo esse sistema de equações para cada conjunto de pontos de cada junta, obtém-se as coordenadas 3D das juntas detectadas para cada esqueleto. Vale ressaltar que o esqueleto reconstruído pode ou não conter todas as 18 juntas que compõem o modelo utilizado neste trabalho.

3.1.5 Visão geral e aspectos de implementação

A Figura 17 mostra de uma forma resumida todo o processo de obtenção das juntas tridimensionais do esqueletos em um sistema multicâmeras. A primeira das cinco etapas consiste na aquisição das imagens. Esse é um processo que idealmente deveria ser sincronizado. Contudo, tal característica nem sempre pode ser atendida, principalmente em sistemas com câmeras de videomonitoramento que não possuem tal funcionalidade. Entretanto, quando se trabalha com taxas de amostragem suficientemente grandes, o sincronismo deixa de ser uma necessidade, por exemplo, porque os movimentos capturados não são rápidos o suficiente para que *frames* consecutivos tenham grande diferença entre si. Desse modo, pouco influenciaria a precisão da reconstrução utilizar *frames* de janelas de amostragem diferentes. Em contra-partida, há a necessidade que o sistema multicâmeras opere com todas as câmeras na mesma taxa de amostragem, afinal, deseja-se que exista informação visual de todas as câmeras a cada período de amostragem do sistema. Modelos e tipos de lentes, sensores, resolução, entre outros parâmetros não são impeditivos para o funcionamento do sistema descrito, contudo, pode ser necessário realizar o ajuste de parâmetros de captura, como por exemplo, o tempo de abertura do obturador das câmeras. Esse tempo é crucial principalmente ao se capturar cenas com movimentos, podendo gerar imagens com borrões, o que atrapalha no desempenho da etapa seguinte do processo.

A segunda etapa, que consiste na detecção dos esqueletos nas imagens, é executada, neste trabalho, por um detector conhecido na literatura como *OpenPose*. Este possibilita o ajuste de diversos parâmetros que influenciam sua acurácia e tempo de processamento. Além disso, o *hardware* no qual ele é executado também é um fator que compromete o seu tempo de computação. É importante ter em mente que, por se tratar de um sistema que

Figura 17 – Visão geral das etapas do processo de obtenção das coordenadas tridimensionais de juntas de esqueletos.



Fonte: Produção do próprio autor.

deve operar em tempo real, dado uma requisitada taxa de amostragem, se faz necessário que a detecção de todas as imagens das câmeras ocorra, no máximo, no tempo do período de amostragem. Caso esse requisito não seja atendido, antes que todas as imagens sejam processadas, uma nova imagem de alguma das câmeras será capturada. Se o processamento dessas imagens for inserido em uma fila, esta começará a crescer caso não haja uma política de descarte para manter a informação mais nova presente. Contudo, a existência de uma política de descarte causa um efeito de sub-amostragem no sistema, o que pode não ser desejável.

Portanto, o processo de detecção de todas as imagens em um dado período de amostragem deve obedecer essa restrição de tempo, caso contrário, inviabiliza a operação em tempo real. Este requisito pode ser atendido trabalhando-se com várias instâncias do mesmo detector. Ao se utilizar essa abordagem, é bom deixar claro que não faz sentido que haja mais instâncias do que número de câmeras. O caso extremo seria ter uma instância por câmera, mas mesmo assim, cada instância deve ser capaz de processar em um tempo menor do que o tempo de amostragem. Mesmo que o processamento ocorra no seu limite de tempo, o sistema ainda é capaz de operar em tempo real, o que vai ocorrer, é apenas um atraso maior no tempo de resposta deste, mas a taxa na qual as respostas são geradas continuará sendo a mesma que as imagens são produzidas. Esta etapa tem como saída para o sistema uma lista com todas as detecções, como mostrado na Figura 17. Cada um dos esqueletos possui uma anotação no formato $\mathrm{S}_{p\mathrm{C}i}$, em que o índice p está relacionado a cada indivíduo, e o índice i corresponde à câmera em que a sua detecção foi realizada.

De posse das detecções, a terceira etapa consiste na busca de correspondências. Na Figura 17, esta etapa foi divida em duas subetapas. A primeira delas, 3.1, consiste em percorrer cada par de câmeras possíveis do sistema, e para cada detecção em uma câmera, buscar no seu par a melhor correspondência. Esse é um processo que depende do número de câmeras que compõem o sistema, além do número de detecções obtidas em cada uma das imagens. Para cada comparação feita entre o esqueleto de uma câmera e o possível correspondente da outra, computa-se a distância apresentada na Equação 3.3. Esse processo pode claramente ser paralelizado, uma vez que os cálculos são independentes, exigindo apenas uma votação final para decidir qual a melhor correspondência para uma determinada detecção em uma câmera. Além disso, outros aspectos podem ser levados em consideração para reduzir o tempo de execução desta etapa. Pode-se definir um número máximo de detecções por câmeras a serem processadas, eliminando, por exemplo, as que apresentarem o menor *score* médio, uma vez que cada junta detectada possui um *score* associado, possibilitando assim de determinar o valor médio referente a um esqueleto. Dessa maneira, conhecendo o número de câmeras do sistema, é possível chegar em um valor aproximado do tempo máximo que esta etapa pode levar.

Para concluir a terceira etapa, são eliminadas as correspondências potencialmente

falsas, como explicado no fim da Seção 3.1.2. Esse processo consiste em verificar se uma dada correspondência ocorreu nos dois sentidos, ou seja, se existe a correspondência (A, B) e (B, A). Essa verificação pode ser realizada já durante o processo de busca por correspondências, armazenando-se aquelas com correspondências complementares e eliminando as que não atenderem a essa condição. Dessa maneira, esse processo torna-se menos custoso, influenciando pouquíssimo no tempo total da terceira etapa do processo aqui descrito. Na seção de resultados deste capítulo, serão apresentados os tempos de execução medidos correspondentes a essa etapa, ficando assim mais claro que a dependência majoritária do tempo total é da primeira parte.

Em seguida, a quarta etapa é executada e consiste em, tomando os pares de correspondências obtidas como o grafo, encontrar os componentes conectados utilizando o algoritmo DFS (*depth-first search*). Este por sua vez, possui complexidade computacional linear, a depender da quantidade de vértices do grafo. Assim, uma vez que cada detecção em imagem é posta como um vértice do grafo, o tempo de execução dependerá de no máximo a quantidade de detecções em todo o sistema em um dado instante de tempo. Essa etapa também contribui pouco para o tempo total de execução.

Por fim, após formados os grupos de correspondências, as juntas que possuem duas ou mais detecções em cada grupo são então reconstruídas, obtendo-se assim suas posições tridimensionais. O processo descrito é apresentado na Seção 3.1.4, o qual é executado para cada junta a ser construída. Assim como o processo anterior, este também não tem muita influência no tempo total.

3.2 Experimentos e Resultados

Nesta seção serão apresentados os resultados obtidos com o método proposto, bem como a descrição da metodologia utilizada nos experimentos realizados. Primeiramente, será descrito o *dataset* escolhido: *CMU Panoptic* [3]. A escolha deste foi baseada nos seguintes requisitos: 1) possuir múltiplas vistas (câmeras) com posições semelhantes às de câmeras de videomonitoramento; 2) as sequências de imagens devem possuir mais de um indivíduo simultaneamente; 3) o sistema multicâmera deve ser calibrado e os parâmetros devem ser fornecidos; e 4) o *ground truth* da localização tridimensional das juntas deve ser fornecido e terem sido obtidos com boa precisão.

3.2.1 *Dataset CMU Panoptic*

Este *dataset* foi desenvolvido com o objetivo de capturar a estrutura tridimensional e o movimento de grupos de pessoas durante interações sociais (JOO et al., 2015). Um dos principais desafios para tal tarefa era lidar com casos de oclusão. Para enfrentar esse e

[3] http://domedb.perception.cs.cmu.edu/

outros problemas, o grupo da *Carnegie Mellon University* construiu um estúdio em forma de um domo com 5,49 m de diâmetro e 4,15 m de altura. Esta estrutura foi construída utilizando painéis pentagonais e hexagonais, dos quais 20 dos hexagonais possuem um total de 24 câmeras com resolução VGA (640×480 *pixels*) cada. Além disso, há 31 câmeras com resolução HD (1920×1080 *pixels*), 10 sensores *Kinect II*, além de 5 projetores de imagens utilizados nas rotinas de calibração. Toda essa estrutura foi utilizada para realizar uma captura sincronizada de sequências envolvendo diversos tipos de interações sociais, e, a partir das imagens obtidas, determinar a posição tridimensional das juntas de esqueleto de cada indivíduo presente na sequência. O diferencial deste *dataset* é que, além de oferecer uma precisão de localização na ordem de unidades de centímetros, ele é capaz de identificar o maior número possível de juntas mesmo em casos de oclusão extrema, em que duas pessoas estão muito próximas. Como foi mostrado em (JOO et al., 2015), em algumas das sequências capturadas, isso não seria possível apenas com sensores do tipo *Kinect*.

Apesar da enorme quantidade de câmeras presentes nesse *dataset*, foram utilizadas apenas imagens de 10 câmeras HD neste trabalho, pois julgou-se como o suficiente para a validação do método, uma vez que sua proposta tem como característica utilizar poucas vistas para estimar a pose de esqueletos, diferentemente de métodos que fazem uso de nuvem de pontos.

O *dataset* contém 65 sequências agrupadas em 8 categorias, totalizando 5,5 horas de vídeos e 1,5 milhão de anotações de esqueletos 3D. Existem dois tipos de anotações: uma com 15 e outra com 18 juntas, sendo esta última a usada neste trabalho. Dentre as categorias, escolheu-se a *Haggling*, por possuir, na maior parte do tempo, três pessoas interagindo. Além das anotações, são também disponibilizadas as calibrações das câmeras e os vídeos capturados pelas mesmas. Até o momento em que a parte experimental deste trabalho foi realizada, a categoria de sequências escolhida no *dataset* possuía um total de quatro sequências. As informações sobre cada sequência estão listadas no Quadro 3.

Quadro 3 – Informações das sequências utilizadas. O número de amostras correspondem ao total de esqueletos tridimensionais presentes em todos instantes da sequência. A quantidade de indivíduos corresponde ao número de pessoas diferentes que participaram da gravação da sequência, sendo que aparecem no máximo 3 em cada instante de tempo.

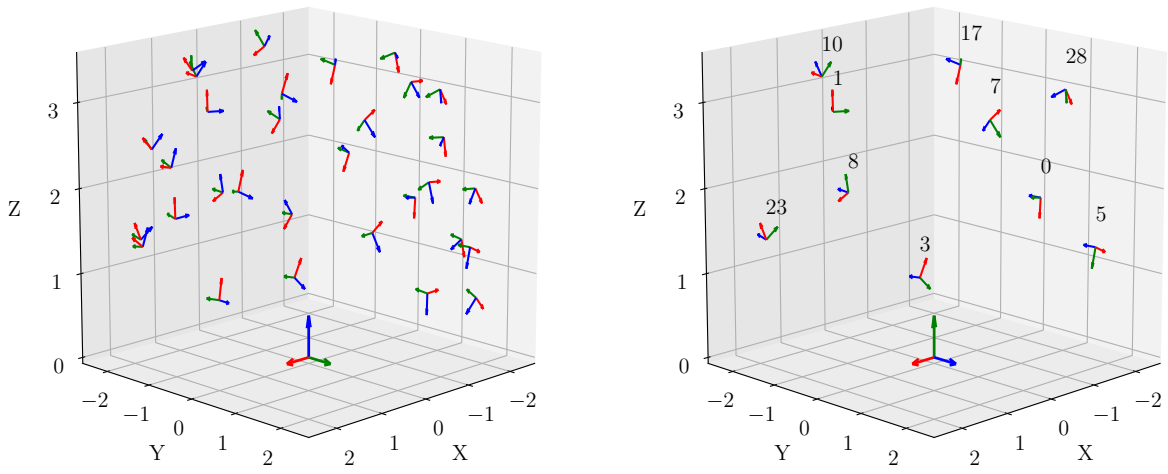
Sequência	**Nome**	**Tempo (min)**	**Amostras**	**Indivíduos**
S1	*160224_haggling1*	5:00	8715	12
S2	*160226_haggling1*	8:00	11464	18
S3	*160422_haggling1*	8:00	13579	18
S4	*161202_haggling1*	8:00	10849	12
Total	-	29:00	44607	-

Fonte: Adaptado de <http://domedb.perception.cs.cmu.edu/>.

A escolha de se utilizar apenas 10 câmeras foi baseada também na distribuição destas no espaço. Isso pode ser verificado na Figura 18a, na qual estão desenhadas todos os referenciais das 31 câmeras HD. Enquanto que na Figura 18b, encontram-se apenas as utilizadas neste trabalho. A escolha foi feita da seguinte maneira: as 31 câmeras foram dispostas em seis grupos de cinco câmeras cada, cada grupo em uma altura diferente, além de uma última câmera que ficou mais baixa que todas as outras. Escolheu-se, partindo do mais alto, o segundo e o quarto grupos de câmeras. As imagens de uma mesma cena capturadas por essas 10 câmeras podem ser verificadas na Figura 19.

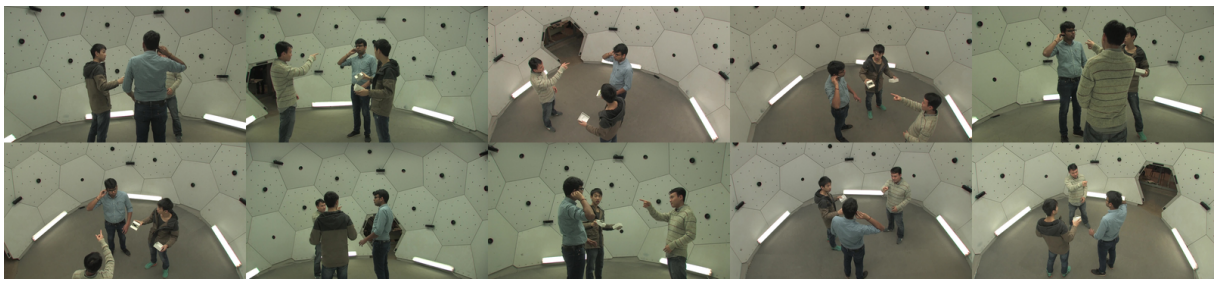
Figura 18 – Visualização das câmeras do *dataset*. Em (a), estão representados todos os referencias das 31 câmeras HD, enquanto que em (b), estão representadas apenas as 10 câmeras utilizadas nos experimentos com seus respectivos identificadores.

(a) (b)



Fonte: Produção do próprio autor.

Figura 19 – Imagens capturadas pelas 10 câmeras do *dataset CMU Panoptic* selecionadas para os experimentos, em uma das sequências escolhidas.



Fonte: Adaptado do *dataset CMU Panoptic*.

3.2.2 Metodologia Experimental e Métricas Utilizadas

Para avaliar o método proposto, foram realizados experimentos com as quatro sequências apresentadas no Quadro 3 e com as 10 câmeras mostradas na Figura 18b. O conjunto de câmeras foi separado em três grupos, sendo dois com cinco câmeras, e o terceiro com as 10. Para os de cinco câmeras, um possuía duas câmeras posicionadas a uma altura mais baixa e três mais alta, e o outro o contrário. Na Figura 19, cada uma das linhas corresponde às imagens dos grupos de cinco câmeras, e, no Quadro 4, encontram-se os identificadores das câmera que compõem cada grupo.

Quadro 4 – Grupos de câmeras do *dataset CMU Panoptic* utilizados nos experimentos.

Grupo	**Câmeras**
G1	$0, 3, 7, 10, 23$
G2	$1, 5, 8, 17, 28$
G3	$0, 1, 3, 5, 7, 8, 10, 17, 23, 28$

Fonte: Produção do próprio autor.

Primeiramente, buscou-se avaliar se o método proposto é capaz de, em um sistema multicâmeras, agrupar corretamente as detecções de cada indivíduo presente na cena, para então, reconstruir tridimensionalmente a posição das juntas. Para fazer isto, utilizaram-se as localizações tridimensionais da juntas e as calibrações das câmeras fornecidas pelo *dataset*. Essas localizações são fornecidas juntamente um identificador único para cada indivíduo, o que possibilitou verificar a eficácia do processo de agrupamento de detecções.

Para isso, foram projetadas as coordenadas tridimensionais das juntas no referencial das câmeras, a fim de se obter as coordenadas das juntas em cada uma das imagens das sequências. Projeções que excederam os limites da resolução da câmera foram desconsideradas. Dessa maneira, obteve-se a posição das juntas nas imagens com a maior precisão possível, ou seja, desconsiderando o erro inserido por um processo de detecção de esqueletos. Além disso, também foram obtidas as coordenadas das juntas mesmo em casos de oclusão.

Uma vez que o desenvolvimento desta parte do trabalho tem por objetivo ser utilizando como entrada de um sistema que seja capaz de, dado os requisitos, realizar a localização e classificação de gestos em tempo real, também foi avaliado o tempo de processamento gasto para cada amostra. Para cada execução, foi levado em consideração o número de detecções por câmera, uma vez que este é um fator que influencia no tempo de processamento. Assim sendo, para cada sequência e grupo de câmeras, foram avaliadas as seguintes métricas:

- Percentual de indivíduos G_{ind} que tiveram suas detecções agrupadas com sucesso, mesmo que nem todas as suas juntas tenham sido reconstruídas. Isto é, o grupo

obtido após o processo de agrupamento de correspondências deve conter apenas detecções com os mesmos identificadores de indivíduo do *dataset*.
- Para cada junta do modelo utilizado, o erro médio de localização em milímetros.
- A razão entre o tempo de execução e o número de vezes em que foi computada a distância média $\bar{d}_{j,l}$ (Equação 3.3), para se obter os pares de correspondências em um dado instante.

Em seguida, uma vez que se propõe um sistema completo, que inclua a etapa de obtenção das localizações das juntas nas imagens, para todos os vídeos de todas as câmeras aqui utilizadas e para todas sequências, foi realizada a detecção de esqueletos. O processo de detecção foi realizado com o detector em diferentes configurações, e em diferentes modelos de GPU que haviam disponíveis no laboratório onde o trabalho foi desenvolvido, a fim de mostrar a influência do detector no processo como um todo.

De posse dessas detecções, o processo de obtenção da localização tridimensional das juntas foi executado. Nessa etapa, uma vez que não há identificação de cada indivíduo nas imagens, não é possível verificar se o agrupamento está sendo feito de maneira correta. Seria necessário uma identificação após o processo de detecção, já que se pode ter, por exemplo, falsas detecções que não correspondem a pessoas nas imagens.

O tempo gasto no processo de detecção também foi aferido. Como o objetivo é ter todo o processo sendo executado em tempo real, era preciso verificar sua viabilidade. Além disso, esses tempos foram utilizados como base de comparação para outras métricas. Portanto, para cada configuração do detector, foram avaliadas as seguintes métricas:

- Erro médio de localização de cada junta em milímetros.
- Para cada modelo de GPU diferente utilizada, o tempo gasto em milissegundos para executar a detecção de esqueletos.

Como mencionado, por não haver a identificação de cada indivíduo, o erro de localização de cada junta foi avaliado considerando-se que o seu *ground-truth* é o esqueleto mais próximo do conjunto de esqueletos fornecidos pelo *dataset* para aquele instante.

3.2.3 Parâmetros e Equipamentos Utilizados

Para todos os experimentos descritos na seção anterior, a distância d_{max} referente à métrica da Equação 3.3 foi o único parâmetro a ser ajustado no método desenvolvido, o qual teve o valor de 50 *pixels*. Esse valor foi obtido experimentalmente, sendo possível verificar a influência da resolução da imagem no valor utilizado. Além disso, o processo foi executado em um computador com processador Intel Core i7-6850K @ 3.60GHz e 64 GB de memória RAM. Contudo, limitado ao uso de recursos para um núcleo do processador, uma vez que a implementação não utiliza técnicas de processamento paralelo, e apenas

128 MB de memória, uma vez que, a partir de testes verificou-se que não era necessário mais do que isso.

O detector utilizado possibilita configurar alguns parâmetros[4] que influenciam na acurácia e no tempo de execução. Dentre os parâmetros, o único alterado foi a resolução de entrada. Todos os outros foram deixados em seus valores padrão. Foram utilizadas as seguintes configurações apresentadas no Quadro 5.

Quadro 5 – Diferentes configurações utilizadas para o detector *OpenPose* nos experimentos realizados. Na resolução indicada para cada configuração, o valor -1 indica que o valor será calculado baseado na relação entre as dimensões (largura e altura) da imagem.

Configuração	**Resolução de Entrada**
C1	-1×256
C2	-1×192
C3	-1×144

Fonte: Produção do próprio autor.

Além disso, quatro modelos diferentes de GPUs foram utilizados e estão apresentados no Quadro 6. Todos modelos pertencem à arquitetura Pascal de placas da NVIDIA. Juntamente com o modelo da GPU, está apresentado no Quadro 6, os modelos de processador do computador no qual a GPU estava instalada. Nesse caso, memória RAM não foi um fator influenciador no desempenho do detector. Ambos os recursos computacionais foram limitados a um núcleo do processador e a 1024 MB de memória para cada instância do detector, de modo que não houvesse compartilhamento de recursos, o que poderia influenciar nas medições.

Quadro 6 – Diferentes modelos de GPU utilizados no processo de detecção de esqueletos, sua quantidade de memória, e o processador do computador na qual a placa gráfica está instalada.

GPU	**Modelo**	**Processador**
GPU1	NVIDIA GeForce GTX 1070 - 8GB	Intel Core i7-7700K @ 4.20GHz
GPU2	NVIDIA GeForce GTX 1080 - 8GB	Intel Core i7-7700K @ 4.20GHz
GPU3	NVIDIA GeForce GTX 1080 Ti - 11GB	Intel Core i7-960 @ 3.20GHz
GPU4	NVIDIA TITAN Xp - 12GB	Intel Core i7-7700 @ 3.60GHz

Fonte: Produção do próprio autor.

[4] Os parâmetros de configuração do detector *OpenPose*, assim como seus valores padrão, podem ser encontrados nas opções do executável de exemplo, disponível em <https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/v1.4.0/examples/openpose/openpose.cpp>.

3.2.4 Resultados

Como descrito anteriormente, o primeiro experimento teve como objetivo verificar se o método proposto era capaz de agrupar corretamente detecções de múltiplos indivíduos em um sistema multicâmeras. Para cada grupo de câmeras apresentado no Quadro 4, e para cada sequência do Quadro 3, foi avaliado o percentual de indivíduos agrupados corretamente. Os resultados estão apresentados na Tabela 1. Como pode ser observado, os percentuais obtidos ficaram entre 97,85% e 99,83%. Um dos fatores para se obter resultados dessa magnitude, é devido ao fato de neste processo haver apenas erros resultantes do processo reprojeção, que está relacionado com a precisão da calibração das câmeras, e não com erros relacionados ao processo de detecção de esqueletos.

Tabela 1 – Percentual de indivíduos agrupados corretamente, separados por sequências do *dataset* e grupos de câmeras.

Grupo de Câmeras	**Sequência**	G_{ind} (%)
G1	S1	99,34
	S2	98,98
	S3	99,51
	S4	99,36
G2	S1	99,79
	S2	99,76
	S3	**99,83**
	S4	99,65
G3	S1	99,41
	S2	**97,85**
	S3	99,07
	S4	98,09

Fonte: Produção do próprio autor.

Este resultado é importante para mostrar a capacidade do método de agrupar múltiplos indivíduos em uma cena, utilizando apenas informações geométrica obtidas a partir das informações de calibração do sistema de câmeras. Em (KADKHODAMOHAMMADI; PADOY, 2018), também foi utilizada geometria epipolar para realizar a correspondência entre as detecções em múltiplas câmeras, entretanto, a posição tridimensional das juntas foi obtida a partir de uma regressão cujo modelo foi treinado considerando a configuração de câmeras do sistema utilizado. Já em (BELAGIANNIS et al., 2014b), diferente do método aqui proposto, foi necessário o conhecimento à priori da quantidade de pessoas presentes na cena para se obter as coordenadas tridimensionais das juntas do esqueleto de cada indivíduo.

É claro que, mesmo que os resultados destes experimentos sejam expressivos, deve-se ressaltar que as coordenadas na imagem das juntas dos esqueletos são provenientes de um

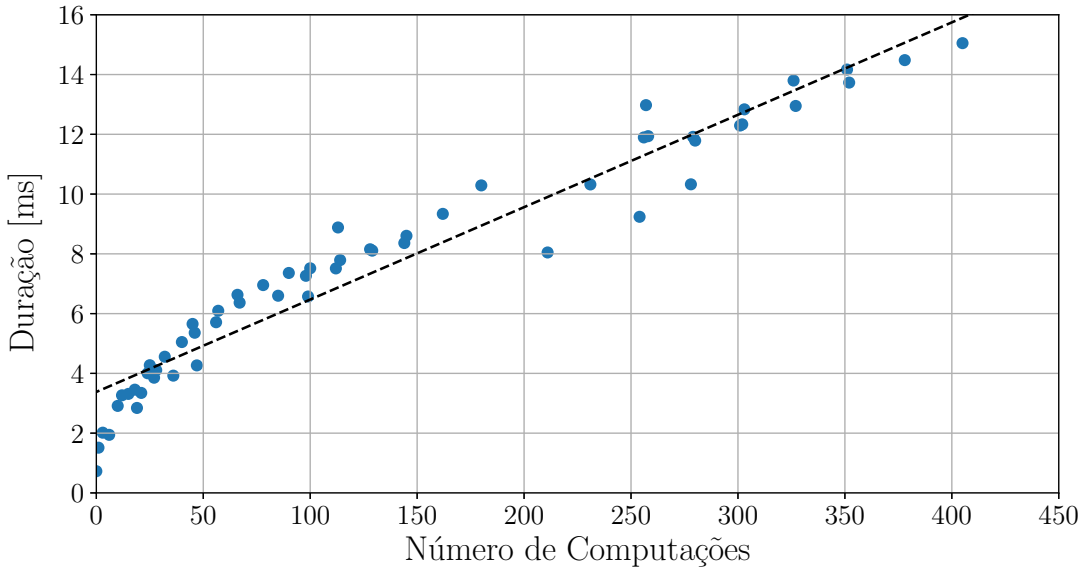
processo de detecção, que, a depender da configuração do detector, pode entregar detecções com erros significativos, ou pior, não conseguir detectar sequer uma junta de um indivíduo. Esses fatores influenciam no resultado do processo de busca de correspondências quando se utiliza apenas informação geométrica para tal. Entretanto, existem métodos como o proposto em (DONG et al., 2019), que utilizam informação geométrica e de aparência para obter melhores resultados no processo de correspondências. Contudo, isto significa incluir uma etapa de extração de características da região na qual o indivíduo se encontra, o que contribui para o aumento no tempo de processamento.

Nesse contexto, uma vez que se propôs neste trabalho um método que fosse capaz de ser executado em tempo real, dada as restrições de taxa de execução, foi medido o tempo gasto no processo de obtenção das coordenadas tridimensionais de juntas de esqueletos. Como explicado na Seção 3.1.5, esse tempo depende majoritariamente da quantidade de vezes que a métrica apresentada na Equação 3.3 é computada. Assim, a Figura 20 apresenta um gráfico que relaciona o número de computações da referida métrica e o tempo gasto. O tempo foi medido para todas as execuções dos experimentos que foram realizados com as projeções do *ground-truth*, já que neste caso, como não há problemas de oclusão e falha de detecção, têm-se em uma grande quantidade de esqueletos. Dessa forma, foi calculado o tempo médio para cada quantidade de computações da métrica. Cada um desses valores médios está representado por pontos no gráfico. Também foi feita uma aproximação linear com os pontos, a qual encontra-se representada pela linha tracejada no gráfico, o que ajuda a ilustrar e confirmar a dependência do tempo total de execução com a quantidade de vezes que a métrica foi computada.

Como já explicado, o tempo total depende da quantidade de câmeras e de detecções em cada câmera. Os tempos apresentados na Figura 20 foram coletados dos experimentos executados para obter os resultados apresentados na Tabela 1, que em seu pior caso, possuía um total de 10 câmeras e de até 3 detecções por câmera. Este cenário representa o ponto no gráfico com mais de 400 computações, que apresentou um tempo de aproximadamente 15 ms. Contudo, o então pior cenário destes experimentos realizados possuía um número grande de câmeras para um espaço com diâmetro de 5,49 m, ou seja, com menos câmeras e, portanto, com um número menor total de detecções, conseguiria-se, com certeza, resultados satisfatórios e um tempo de processamento menor que 15 ms para realizar todo o processo de busca de correspondências e reconstrução tridimensional das juntas dos esqueletos.

Para mostrar a viabilidade de utilizar o método proposto em um sistema que funcione em tempo real, também foram realizadas medições de tempo do processo de detecção de esqueletos nas imagens. Utilizando as quatro GPUs disponíveis, listadas no Quadro 6, e para cada uma das configurações do detector apresentadas no Quadro 5, estão apresentados na Tabela 2, os tempos médios para cada configuração. Sendo assim, foram coletados os tempos de 10000 execuções para cada configuração possível para se obter o

Figura 20 – Tempo de duração média em milissegundos, para execução do processo de obtenção das coordenadas tridimensionais das juntas de esqueletos, medido a partir da etapa de busca de correspondências. Cada ponto no gráfico corresponde ao tempo médio para as execuções que apresentaram a mesma quantidade de computações da métrica da Equação 3.3. Valores medidos nos experimentos realizados a partir das projeções do *ground-truth* do *dataset CMU Panoptic*.



Fonte: Produção do próprio autor.

valor médio apresentado.

Como pode ser observado na Tabela 2, para a configuração C1, que oferece melhor precisão, o maior tempo obtido foi de 68,68 *ms*, utilizando-se a GPU3, enquanto que para a configuração C3, de menor precisão, o menor tempo obtido foi de 30,16 *ms* utilizando a GPU2. A partir dos tempos obtidos, é possível verificar que dependendo da quantidade de câmeras do sistema, da taxa de aquisição, e da precisão exigida, é possível definir qual a configuração mais adequada, bem como o número de GPUs necessárias.

Por exemplo, no sistema disponível no laboratório em que este trabalho foi desenvolvido, existem quatro câmeras instaladas bem como as GPUs listadas no Quadro 6. Caso se deseje operar na configuração de precisão mais baixa, a C3, com quatro instâncias do detector disponíveis, é possível operar com um período de amostragem de 50 ms (20 fps). Vale ressaltar que o tempo de detecção também depende do processador instalado correspondente à GPU, pois a imagem precisa ser descomprimida e redimensionada. Tais operações não são realizadas em GPU, contudo, correspondem a uma pequena parcela do tempo total de execução do detector. No quesito tempo de processamento, este trabalho pode ser comparado com (DONG et al., 2019), o qual comenta que seu método proposto é capaz de funcionar em taxas maiores que 20 quadros por segundo. Entretanto, apenas

fornece os tempos gastos em seu método, não considerando o tempo gasto pelo detector, o que deve ser considerado em um sistema operando em tempo real.

Tabela 2 – Tempo médio em milissegundos para execução do processo de detecção dos esqueletos nas imagens do *dataset Panoptic CMU*, utilizando 10000 medições para cada combinação de modelo de GPU, apresentados no Quadro 6, e configuração do detector, apresentadas no Quadro 5.

GPU	**Configuração**	**Duração (ms)**
GPU1	C1	68,12
	C2	43,56
	C3	36,64
GPU2	C1	59,50
	C2	39,52
	C3	**30,16**
GPU3	C1	**68,68**
	C2	55,98
	C3	52,04
GPU4	C1	46,30
	C2	37,37
	C3	35,14

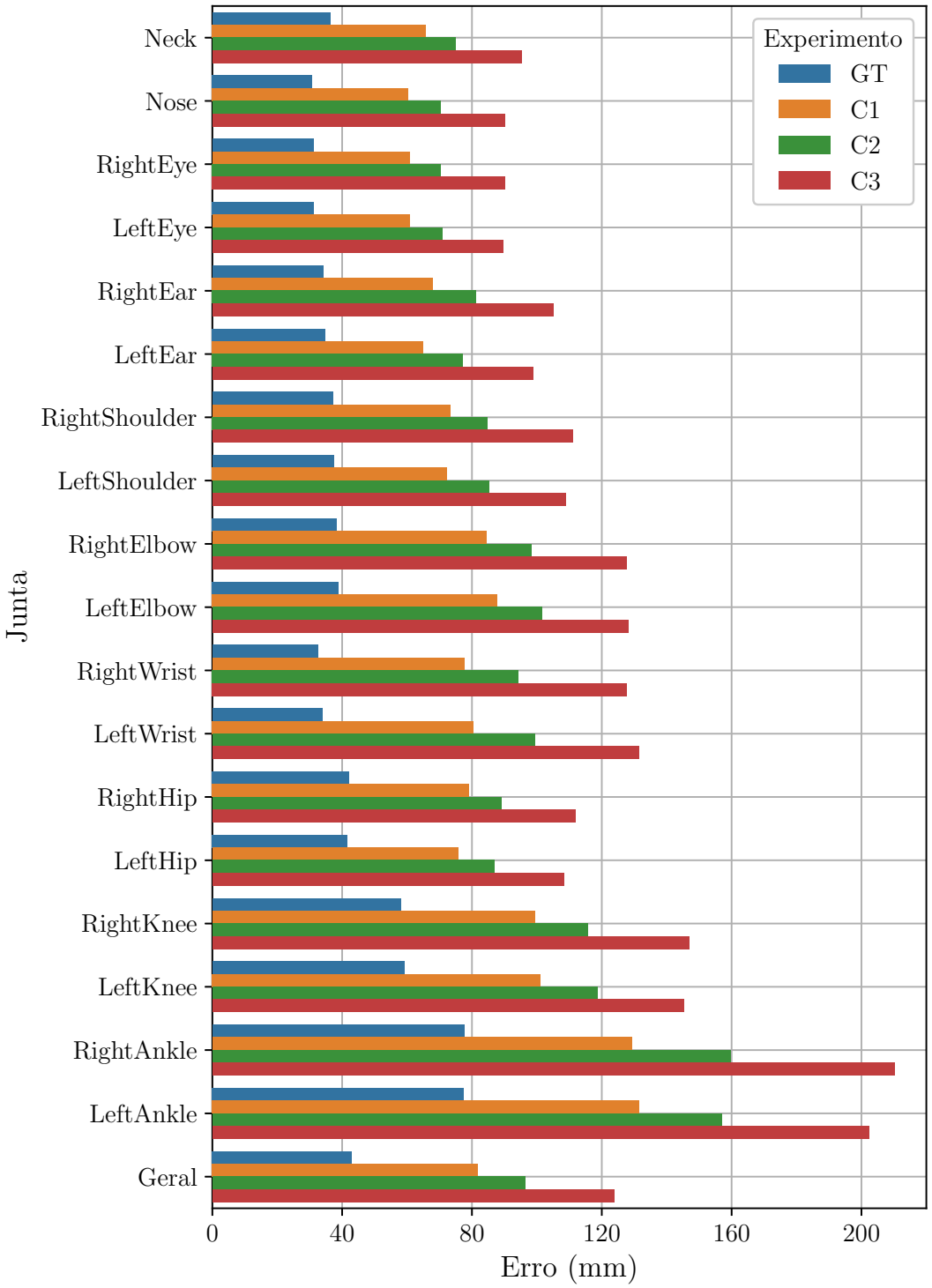
Fonte: Produção do próprio autor.

Outro aspecto avaliado do método aqui proposto, é a precisão na localização das coordenadas das juntas detectadas. Utilizando as mesmas configurações do experimento feito para se avaliar o tempo de processamento do detector, bem como as projeções do *ground-truth*, foram avaliados o erro de reconstrução de cada junta do modelo utilizado em milímetros. Na Figura 21, está apresentado o erro médio para cada junta, compreendendo todas as sequências do *dataset* utilizado. Além disso, também está apresentado o erro médio geral para cada um dos quatro experimentos.

Para o experimento que utiliza o *ground-truth* do *dataset*, identificado na legenda pela sigla GT, verifica-se que ele apresentou os menores erros para todas as juntas do modelo, e, consequentemente, o menor erro geral. Esse resultado era esperado, uma vez que a realização deste experimento não incluiu a etapa de detecção. As coordenadas das juntas foram obtidas projetando-se o *ground-truth* no plano da imagem. Isto quer dizer que o erro médio para cada junta obtido corresponde apenas ao erro acumulado pelo processo de projeção e em seguida o de reconstrução. A presença deste erro está associada à precisão obtida no processo de calibração das câmeras. Além disso, ao fato de se considerar ou não parâmetros de distorção radial.

Nos experimentos com as configurações apresentadas no Quadro 5, de maneira geral, pode-se verificar como esperado: para uma configuração com resolução de entrada

Figura 21 – Erro médio de localização de cada junta em milímetros, utilizando a localização das juntas nas imagens obtidas a partir da projeção do *ground-truth* (experimento GT), e a partir das detecções obtidas com as três configurações do detector apresentadas no Quadro 5 (experimentos C1, C2 e C3).



maior, C1, o erro médio de localização é menor que para configurações com resolução de entrada menores, como C2 e C3. Uma comparação entre cada uma das configurações é melhor observada no último grupo do gráfico: aquele que corresponde à média geral das

juntas. Neste, verifica-se, por exemplo, que ao se comparar o experimento feito com o *ground-truth* e o de configuração C1, o erro médio vai de aproximadamente 40 mm para 80 mm. Isso mostra a influência que o processo de detecção tem na precisão da localização das juntas.

Ainda observando o valor médio geral, mas agora apenas comparando os experimentos com as três configurações do detector, nota-se que esta configuração tem grande influência no erro obtido, indo de aproximadamente 80 mm para pouco mais de 120 mm da configuração C1 para a C3, o que corresponde a um aumento de aproximadamente 50%. É claro que, deve-se levar em consideração o aumento do tempo de processamento ao se exigir uma melhor precisão, o que pode ser verificado na Tabela 2, em que, por exemplo, para a GPU1, a configuração C1 apresentou tempo de 68,12 ms, enquanto que C3 36,64 ms, uma redução de aproximadamente 46%. Esses resultados mostram o compromisso que se deve ter entre precisão e taxa na qual se deseja operar o sistema.

Outro ponto a ser observado nos resultados da Figura 21, é que independente do experimento realizado, o erro para algumas juntas foi maior. Por exemplo, as juntas *RightAnkle* e *LeftAnkle* apresentaram os maiores erros, ultrapassando 200 mm para a configuração C3. Estas, de maneira geral, encontram-se normalmente nas bordas da imagens. Junto deste fato, tem-se que as lentes das câmeras utilizadas para capturar as imagens do *dataset CMU Panoptic* apresentam uma grande distorção radial. Esta, por sua vez, ao não ser considerada no processo de reconstrução, faz com que o erro de reconstrução seja maior que os das juntas que se localizam mais próximas ao centro da imagem. Esse é um problema específico desse *dataset*, devido ao posicionamento das câmeras e dos participantes das capturas, além é claro, das lentes utilizadas. Tais fatores podem e devem ser considerados em sistemas que necessitam de uma maior precisão, nos quais as capturas serão realizadas de forma semelhante às feitas neste *dataset*. O efeito da distorção é pouco notado nas imagens apresentadas na Figura 19, pois não há nestas a presença de elementos retilíneos que ajudem a evidenciar as curvaturas nas extremidades da imagem.

Contudo, é importante também comparar estes valores obtidos com os de outros trabalhos que obtém as coordenadas tridimensionais com métodos parecidos, além de comparar aspectos técnicos de cada um. Em (KADKHODAMOHAMMADI; PADOY, 2018) por exemplo, no qual também se utiliza geometria epipolar como informação para realizar a correspondência entre detecções, seu método de reconstrução tridimensional consiste em um processo de regressão, diferentemente do método aqui proposto, que obtém as coordenadas de maneira clássica, resolvendo um sistema de equações. Mesmo que a abordagem utilizada aqui seja menos confiável, uma vez que ao realizar a reconstrução com um conjunto de detecções, que podem ter erros ou detecções incorretas, pode resultar em grandes erros de localização, os valores de erro obtidos são bem próximos dos apresentados por (KADKHODAMOHAMMADI; PADOY, 2018). Para todas as configurações do

método proposto em (KADKHODAMOHAMMADI; PADOY, 2018), o menor erro médio encontrado foi de 49,1 *mm* e o maior 119,6 *mm*. Entretanto, esses valores foram obtidos a partir de experimentos realizados com o *dataset* Human3.6M. Além disso, nada sobre o tempo de computação gasto foi comentado nesse trabalho.

Por sua vez, (DONG et al., 2019) utiliza informação geométrica e de aparência para realizar o processo de busca de correspondência. Ele não avalia o erro de reconstrução, mas sim mostra mostra os resultados de seu processo de busca de correspondências. O *dataset CMU Panoptic* também foi utilizado em (DONG et al., 2019), contudo, apenas para resultados qualitativos. (DONG et al., 2019) afirma que o *ground-truth* desse *dataset* não está disponível, o que aparentemente é um equívoco, pois o mesmo encontra-se na página de seus criadores, como apontado na Seção 3.2.1.

Outro trabalho que utiliza múltiplas câmeras, porém estas também possuindo sensores de profundidade (RGB-D), é o apresentado em (CARRARO et al., 2018). Seu método inclui uma etapa inicial de detecção de esqueletos, contudo, o processo de busca de correspondências e reconstrução das juntas é feito utilizando os dados de profundidade de cada um dos sensores. Além disso, ainda existe um rastreamento aplicado às juntas, o que ajuda no caso de detecções ruins ou inexistentes. No melhor cenário (quatro câmeras), o método proposto por (CARRARO et al., 2018) consegue um erro médio de localização das juntas de 38,9 mm[5]. Estes valores estão na mesma ordem de grandeza que os obtidos no trabalho apresentado nessa dissertação, com a configuração C1. Entretanto, (CARRARO et al., 2018) utiliza mais informação (sensores de profundidade), além de uma abordagem temporal no rastreamento das juntas, o que oferece uma vantagem, apesar de adicionar uma limitação: a necessidade da utilização de sensores de RGB-D.

Diferente de (CARRARO et al., 2018) e (KIM, 2017), no qual são combinados sensores RGB-Ds e câmeras convencionais, (LORA et al., 2015) utiliza uma rede de câmeras comuns para obter as coordenadas tridimensionais de juntas de esqueletos. Esse trabalho avalia o erro de localização projetando os pontos tridimensionais no plano da imagem e os compara com o *ground-truth*, uma vez que o *dataset* possui apenas suas imagens anotadas manualmente. Assim, não é possível a comparação com tal trabalho, pois não há conhecimento de seu erro de reconstrução das coordenadas tridimensionais das juntas. Outra característica desse trabalho é que os autores afirmam que o sistema consegue funcionar em tempo real, mas não há nenhuma evidência ou medida apresentada no texto.

Mesmo que a comparação quantitativa do erro com trabalhos que utilizam uma única câmera seja desfavorável, uma vez que possui limitações ao resolver ambiguidades associadas ao processo de reconstrução tridimensional apenas com informação monocular, a comparação se torna interessante para mostrar as possibilidades ao utilizar sistemas

[5] Valor calculado a partir dos valores da última linha da Tabela I de (CARRARO et al., 2018).

monoculares. Em (MEHTA et al., 2017), por exemplo, o erro de localização obtido está entre 124,7 mm e 142,8 mm. Valores próximos ao obtido com a configuração C3 (menor precisão), porém, com a vantagem de utilizar uma única câmera. Além disso, (MEHTA et al., 2017) propõe um método capaz de ser executado à 30 fps. Contudo, por ser uma solução com uma única câmera, não é robusta a oclusões, além de apresentar a limitação de estimar a localização das juntas de apenas uma pessoa.

3.2.5 Considerações Finais

Neste capítulo foi apresentada a metodologia para a obtenção da localização de juntas de esqueletos utilizando um sistema multicâmera, juntamente com seus resultados obtidos da parte experimental. Pode-se verificar a partir dos resultados que o sistema proposto é capaz de entregar a localização de vários indivíduos na mesma cena, com precisão comparável com outros trabalhos e ainda, que sua execução é possível para taxas de amostragens condizentes com tarefas que utilizam como entrada sequências de poses humanas.

Foi possível verificar também que, deve-se buscar o melhor compromisso entre os diversos parâmetros de operação do sistema, isto é, taxa de amostragem desejada, quantidade de câmeras e precisão. Ainda, com a saída entregue pelo sistema, algumas ações podem ser adotadas como melhoria na qualidade da localização, como a utilização de filtragens temporais, ou até mesmo a realização de técnicas de rastreamento para auxiliar o preenchimento de juntas não detectadas.

4 Localização de Gestos e Ações

No capítulo anterior, foi apresentada uma metodologia para se obter em tempo real a localização das juntas de esqueleto. Estas por suas vez podem ser utilizadas como entrada para diversas aplicações. Para a tarefa de reconhecer gestos ou ações em tempo real, como já foi discutido neste trabalho, é preciso determinar o instante em que o gestou ou ação se inicia e termina, independente do tipo de dado utilizado como entrada, para que então seja feita sua classificação. Desta maneira, este trabalho buscou desenvolver uma solução para que, dada uma sequência de poses de um indivíduo, representada pela localização de juntas do esqueleto ao longo do tempo, o sistema fosse capaz de determinar os instantes final e inicial de um gesto ou ação.

Assim, este capítulo irá descrever a metodologia adotada e os desafios enfrentados para desenvolver um modelo capaz de atingir o objetivo desta etapa do trabalho, desde a adequação dos dados de entrada, até o treinamento do modelo, comparação com outras arquiteturas, e validação de tempos de execução a fim de verificar a possibilidade de se utilizar em uma aplicação de tempo real. Será também descrita a metodologia utilizada para a validação do método, e em seguida apresentando os resultados obtidos com a base de dados escolhida.

4.1 Adequando os dados de entrada

Esta é sem dúvida uma etapa de extrema importância para todo o processo de localização de gestos e ações em tempo real. Independente da abordagem que será utilizada para determinar os instantes de início e término em uma sequência de poses, ter os dados de entrada bem adequados garante robustês ao sistema. Diversos são os fatores que justificam a necessidade dessa adequação. Por exemplo, ao se utilizar dados de esqueleto para esta tarefa, podem existir variações na distância entre as juntas, o que é perfeitamente normal, afinal cada pessoa possui uma estatura e biotipo diferentes, o que faz variar essas distâncias.

Ademais, dependendo das métricas extraídas da sequências das poses, o sistema de coordenadas no qual a captura da sequência foi realizada deve ser levado em consideração. Por exemplo, se for utilizado um *Kinect* para isso, por ser um sistema monocular, seus dados são retornados em relação à sua câmera. É claro que uma calibração pode ser feita pra um referêncial externo, mas não é o usual. De outra maneira, se for utilizado um sistema de captura de movimento multi-camera, como o proposto aqui neste trabalho, é comum adotar um sistema de coordenadas com referêncial externo às câmeras. Para ambos os casos, uma adequação deve ser feita para que o dado utilizado seja invariante à localização deste referêncial.

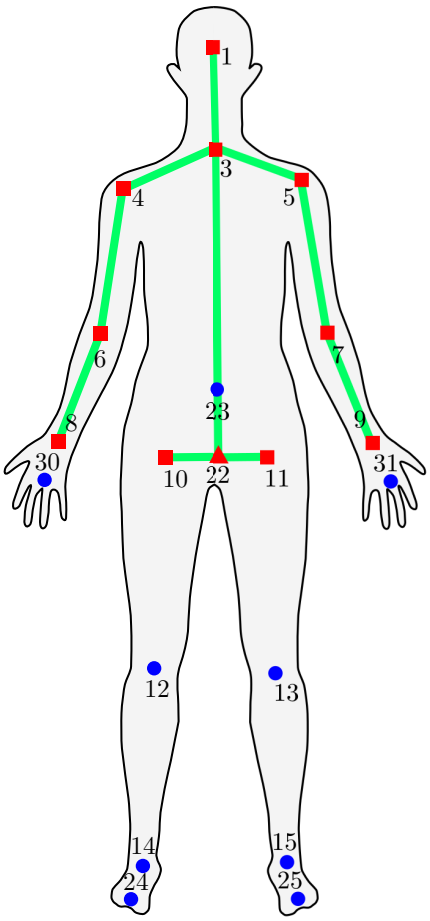
Outro ponto necessário a ser comentado é sobre o modelo de juntas utilizado. Uma vez que este trabalho se propôs a utilizar a pose humana para localizar gestos e ações, e como já foi mostrado na Figura 2 e no Quadro 2, existem vários modelos de pose, e que entre eles existem pontos em comum. É desejável que o sistema desenvolvido seja capaz de operar com diferentes modelos de pose, contudo isso nem sempre é possível. Entretanto, em alguns casos é possível interpolar pontos para obter as coordenadas de um ou mais pontos de outro modelo. Mesmo que a abordagem desenvolvida aqui tenha sido baseada em um modelo de juntas que é utilizado no *dataset* adotado, é importante ressaltar que o método não se limita a um único modelo de juntas, podendo ser usado com outros modelos também.

A metodologia aqui proposta para a localização de gestos e ações baseou-se na proposta no trabalho aprensentado em (NEVEROVA et al., 2014). Este por sua vez apresentou um método para localizar e classificar gestos do *dataset Montalbano Gesture Recognition* (ESCALERA; ATHITSOS; GUYON, 2016), que utilizou um *Kinect v1* para a captura das sequências. Mais detalhes sobre este *dataset* serão apresentados na seção de metodologia experimental deste capítulo. Quando se usam juntas de esqueletos para a localização de gestos, é comum se escolher apenas algumas juntas que são mais representativas nos movimentos para esta tarefa. Isso se deve ao fato de que, por exemplo, em gestos manuais, os membros inferiores permanecem praticamente imóveis durante a execução, e portanto, o movimento das juntas dos membros inferiores acrescentam pouca informação.

Semelhante ao trabalho utilizado como referência, também escolheu-se um grupo de juntas. Estas 11 juntas estão identificadas no Quadro 2 por um símbolo §. A única diferença em relação à (NEVEROVA et al., 2014), foi a escolha das juntas do pulso (*RightWrist* e *LeftWrist*) ao invés das mãos (*RightHand* e *LeftHand*). Esta escolha foi feita pensando em maior compatibilidade com outros modelos que não apresentam a junta das mãos, como pode ser observado no Quadro 2. Na Figura 22, estão apresentadas as localizações das juntas do modelo *Kinect v1*, destacando-se aquelas que foram utilizadas nesse trabalho, bem como conexões entre as juntas e a junta de referência, que será abordada posteriormente.

A partir do conjunto de juntas selecionadas, o primeiro passo é realizar uma mudança de referencial, uma vez que para o *dataset* utilizado o referencial está localizado na câmera do *Kinect*. Assim, escolheu-se a junta correspondente ao *HipCenter* seguida da representação das demais juntas em relação a este ponto. Além disso é preciso garantir que os dados de entrada sejam o mais robustos possíveis a diferentes tamanhos, formato e proporções de corpos. Para isso, é feita uma normalização na pose que consiste no escalamento das distâncias de cada uma das conexões representadas na Figura 22, baseado na distância média de cada conexão, obtida a partir dos dados do *dataset* de treinamento. Este procedimento deve garantir que os ângulos formados pelas conexões não sejam

Figura 22 – Modelo de juntas do *Kinect v1* utilizado no *dataset Montalbano Gesture Recognition*. As 11 juntas utilizadas para a localização de gestos estão destacadas por um quadrado na cor vermelha, com exceção da junta de referência (*HipCenter* - 22) que é um triângulo vermelho. As demais juntas estão representadas com círculos na cor azul. As conexões entre as juntas que foram utilizadas neste trabalho, um total de 10, estão destacadas por segmentos de linha na cor verde.



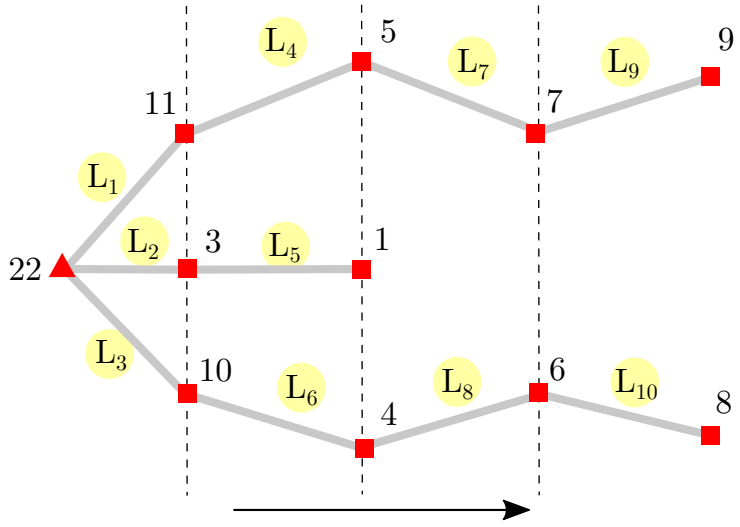
Fonte: Produção do próprio autor.

alterados, pois isso ocasionaria uma distorção da informação. Assim, nesta etapa, é necessária uma representação diferente das juntas do modelo, para garantir que o processo de escalamento não cause qualquer distorção de informação, além de ser necessário determinar o fator de escala para as conexões.

Este processo de normalização da pose foi utilizado em (NEVEROVA et al., 2014) mas foi proposto inicialmente por (ZANFIR MARIUS LEORDEANU, 2013). O primeiro passo consiste em calcular a distância média L_i de cada uma das conexões, construir um vetor $\mathbf{L} = [L_1, L_2, ..., L_n]$ com as $n = 10$ conexões, e então normalizá-lo com a norma L2, obtendo assim os coeficientes $\tilde{L}_i$ utilizados na próxima etapa deste processo. Tais coeficientes serão utilizados para todos dados de entrada. Representa-se então as juntas e conexões destacadas na Figura 22 com a estrutura de árvore, cujo nó raiz corresponde à

junta de referência mencionada anteriormente. Na Figura 23 está apresentada a árvore em questão, com as conexões entre as juntas, além da indicação de cada uma das distâncias L_i. Os níveis de profundidade da árvore estão também seccionados por linhas tracejadas na vertical.

Figura 23 – Representação em forma de árvore das juntas do modelo *Montalbano Gesture Recognition*. A junta *HipCenter*, assinalada por um triângulo vermelho, foi adotada como referência e representa o ponto raiz da árvore. Cada *i-ésimo* segmento que conecta um par de juntas está identificado e destacado por círculos amarelos. Além disso, cada nível de profundidade da árvore está separado por uma linha tracejada vertical, e na parte inferior a seta indica o sentido que a árvore fica mais profunda.



Fonte: Produção do próprio autor.

Para o modelo de árvore apresentado, cada uma das conexões tem um par de juntas associados. Este par é formado por uma junta inicial $\mathbf{p}_i^0$, que corresponde à junta mais próxima a de referência da árvore, e uma final $\mathbf{p}_i$. Então, iniciando do nível menos profundo da árvore, que é aquele com o nó de referência, para cada conexão L_i, determina-se a nova posição para a junta final $\tilde{\mathbf{p}}_i$ com a Equação 4.1. Esta equação consiste em determinar o novo ponto final de uma conexão. Para isto, calcula-se um vetor unitário na mesma direção que a conexão e aplica-se o fator de escala previamente calculado com o *dataset*. Este vetor é então utilizado para calcular a posição nova de $\mathbf{p}_i$ em relação à $\tilde{\mathbf{p}}_i^0$. Vale lembrar, que as coordenadas de ambas as juntas já foram transladadas para a junta de referência antes de iniciar este processo. Para a junta de referência, como está é igual a $[0,\ 0,\ 0]$, temos que $\mathbf{p}_i^0 = \tilde{\mathbf{p}}_i^0$.

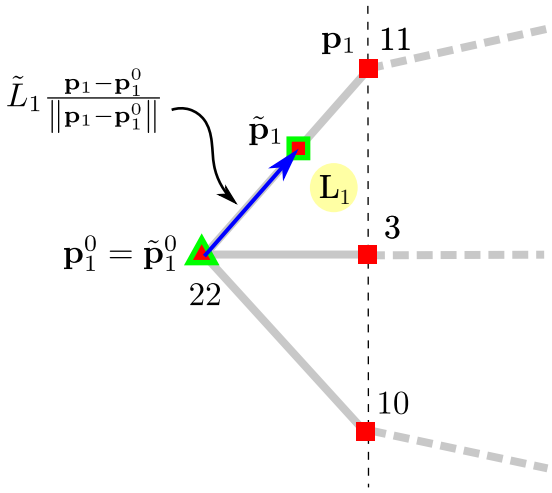
$$\tilde{\mathbf{p}}_i = \tilde{L}_i \frac{\mathbf{p}_i - \mathbf{p}_i^0}{\|\mathbf{p}_i - \mathbf{p}_i^0\|} + \tilde{\mathbf{p}}_i^0 \tag{4.1}$$

O processo é realizado no sentido do nó de referência para os níveis mais profundos,

como indica a seta da Figura 23. Realizar este processo nesta ordem equivale à ordem de visita de nós ao se executar o algoritmo de busca utilizado em grafos e árvores conhecido como BSF (do inglês, *breadth-search first*). O algoritmo que consiste em, a partir do nó raiz, visitar todos os nós adjacentes na profundidade atual, para então mover para os nós da próxima profundidade. Portando, a representação da árvore mostrada na Figura 23, consiste em realizar a normalização apresentada pela Equação 4.1, para as conexões em cada nível de profundidade. Cada conexão foi enumerada em uma possível ordem em que este procedimento pode ser executado.

Como já mencionado, a ordem de execução do processo de normalização se faz necessária para que não haja distorção de informação. Isso se dá devido ao fato de que, como pode ser observado na Equação 4.1, a posição da junta é atualizada. Na Figura 24 podemos verificar a normalização da conexão L_1, em que a nova posição da junta 11, representada na figura por $\mathbf{p}_1$ é determinada. Nesta ilustração pode-se também verificar a representação geométrica da Equação 4.1, em que a nova posição do ponto final da conexão $\tilde{\mathbf{p}}_1$ corresponde à soma da nova posição do ponto inicial, $\tilde{\mathbf{p}}_0$ e o vetor unitário escalado pelo fator de normalização da conexão $\tilde{L}_1$.

Figura 24 – Representação gráfica da normalização da conexão L_1. As juntas que já tiveram posições atualizadas pelo processo de normalização estão destacadas por uma borda verde em torno de seu marcador, e o vetor calculado para determinar a nova posição de $\mathbf{p}_1$ encontra-se na cor azul sobre a conexão L_1.

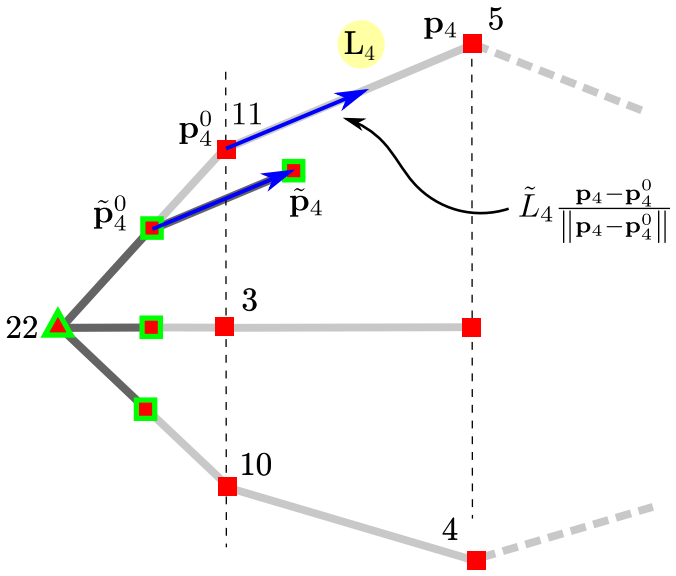


Fonte: Produção do próprio autor.

Após realizar o redimensionamento de todas as conexões do primeiro nível de profundidade da árvore, isto é, as conexões L_1, L_2 e L_3, parte-se para a conexão L_4, cuja representação gráfica está na Figura 25. Neste exemplo, fica mais claro o porquê deste procedimento ser executado desta maneira. Ou seja, assim como o algoritmo BFS é executado, pois, para se determinar a nova posição de $\mathbf{p}_4$, é preciso que já se tenha a nova posição de $\mathbf{p}_4^0$, isto é, $\tilde{\mathbf{p}}_4^0$, que foi determinado anteriormente ao se realizar o

redimensionamento de L_1, em que na ocasião se determinava a nova posição de junta de número 11. Determina-se então a nova posição da junta de número 5, $\tilde{\mathbf{p}}_4$, que será utilizada na próxima camada para determinar a nova posição da junta de número 7, aquela que está conectada à $\mathbf{p}_4$ pela junta L_7.

Figura 25 – Representação gráfica da normalização da conexão L_4. As juntas que já tiveram posições atualizadas pelo processo de normalização estão destacadas por uma borda verde em torno de seu marcador, e o vetor calculado para determinar a nova posição de $\mathbf{p}_4$ encontra-se na cor azul sobre a conexão L_4.

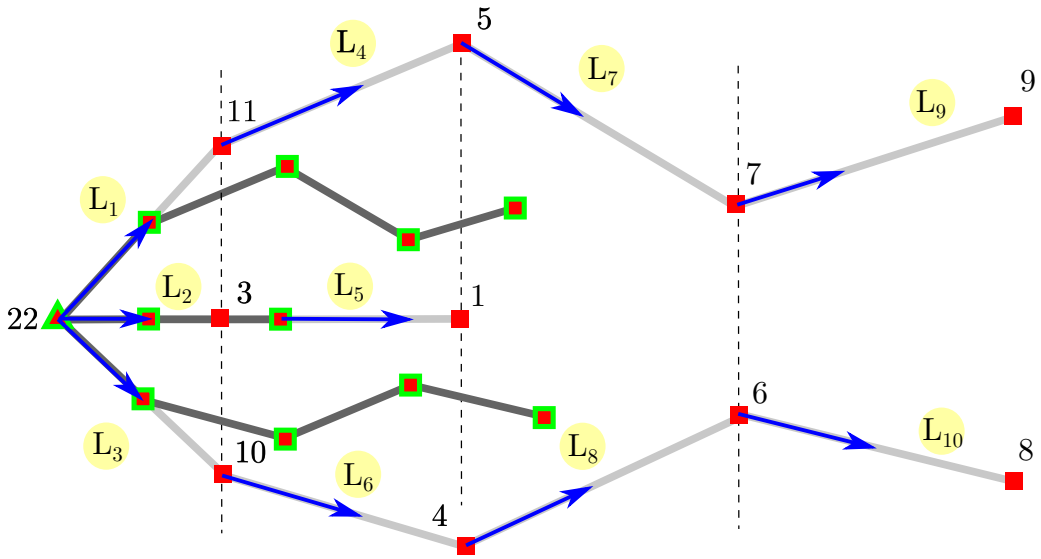


Fonte: Produção do próprio autor.

Ao fim do processo, obtemos um esqueleto cuja referência está na junta de número 22 (*HipCenter*), e todas as dimensões das conexões entre as juntas estão normalizadas, fazendo com que a entrada do sistema seja o mais robusta possível a diferenças de tamanho, forma e proporções. Na Figura 26 podemos observar o esqueleto obtido após a normalização, com suas conexões desenhadas em um tom mais escuro que o esqueleto original, e os marcadores das juntas com uma borda verde, como feito nas figuras anteriores. Todos os vetores calculados durante o processo também estão desenhados. Assim fica mais claro entender como o processo funciona. Outra observação importante, é a preservação da forma do esqueleto normalizado ao se comparar com o esqueleto original. É claro que ocorrem pequenas variações nos ângulos entre as conexões adjacentes, uma vez que o fator de escala aplicado em cada conexão não é proporcional ao tamanho original desta. Isso porquê, como já mencionado, existem corpos com proporções diferentes, e estas proporções entre as conexões podem destoar do esqueleto médio obtido a partir do *dataset*.

Agora que temos uma pose normalizada, a próxima etapa consiste em determinar um descritor de características espaço-temporal para que sejam utilizadas na classificação do instante de execução como gesto ou não-gesto.

Figura 26 – Posições das justas iniciais e obtidas após o processo de normalização, com seus marcadores destacados por uma borda verde. Todos os vetores calculados durante o processo estão posicionados com sua base na junta de origem na qual foi calculado. É notável a manutenção da geometria ao se realizar o procedimento como fora descrito.



Fonte: Produção do próprio autor.

4.2 Descritor de pose

Esta etapa está fortemente associada a uma etapa subsequente, que consiste em classificar se um determinado instante é o início ou final de um gesto ou ação. Como já mencionado, diversas são as abordagens para tal tarefa, e dependendo da abordagem, diferentes maneiras de se pré-processar a sequência de poses pode ser utilizada. Dado o escopo deste trabalho, no qual deseja-se localizar um gesto em tempo real, optou-se por utilizar a pose de cada instante, e para cada instante gerar uma saída que seja interpretada posteriormente como o início ou fim de uma execução. Outras abordagens podem ser adotadas, em que se utilizam janelas deslizantes, mas essa possui algumas desvantagens para uma abordagem em tempo real. Mais detalhes sobre esta escolha serão abordados quando for descrita a etapa de classificação.

Assim sendo, a etapa de descrever a pose consiste em, para cada instante de tempo, gerar um vetor de características que sirvam de entrada para um modelo classificar se aquele instante corresponde, neste caso, a um instante em que o indivíduo está executando um gesto ou ação, ou se está em repouso. Essas características podem ser utilizadas para, além de localizar um gesto, isto é, determinar seus instantes de início e fim, para classificar o gesto ou ação executada. Em (NEVEROVA et al., 2014), por exemplo, um vetor de características é utilizado junto com outras informações provenientes de outros sensores além de uma imagem RGB, para realizar a tarefa de classificação de gestos.

A definição dessas características leva em consideração características espaciais e temporais da sequência de poses. Características espaciais estão relacionadas a posições, distâncias e ângulos entre partes do corpo. Tais distâncias podem ser expressas de forma absoluta ou normalizada, enquanto que informações de ângulo na forma de alguma função trigonométrica, o que ajuda a trazer robustez e invariância. Em contrapartida, características temporais estão relacionadas a como uma característica espacial evolui ao longo do tempo. Por exemplo, a velocidade que uma determinada junta se move ao longo do tempo, ou até mesmo a aceleração da mesma.

Dessa forma, esse trabalho, baseado no que foi proposto em (NEVEROVA et al., 2014), propõe um descritor composto por 7 vetores de característica, dentre elas características espaciais e temporais. Cada um deles serão descritos a seguir em sessões separadas, finalizando com um comentário geral sobre estas características.

4.2.1 Características

4.2.1.1 Posição das juntas

O primeiro conjunto de características se refere à posição das 11 juntas em um dado instante de tempo. Dessa maneira, seja $\tilde{\boldsymbol{p}}_i = [\tilde{x}_i, \tilde{y}_i, \tilde{z}_i]$ a posição da *i-ésima* junta, para $i \in [1, 11]$, que já passou pelo processo de normalização. Para o conjunto de posições das juntas, é aplicado um filtro para que se evite variações bruscas nas posições. Em (NEVEROVA et al., 2014; ZANFIR MARIUS LEORDEANU, 2013), é utilizado um filtro Gaussiano com *kernel* de tamanho 5×1 e $\sigma = 1$. Contudo, uma vez que este trabalho propõe uma abordagem em tempo real, a utilização de um filtro desse tipo é inviável, pois é não causal e precisaria de duas amostras futuras para determinar o valor filtrado no instante atual.

Assim, a filtragem das posições das juntas é feita utilizando um EMA (Média Móvel Exponencial, do inglês *Exponential Moving Average*). Este filtro calcula uma média móvel, dando maior importância aos valores mais recentes, sendo definido por

$$\boldsymbol{y}[n] = (1 - \alpha)\boldsymbol{y}[n-1] \; + \; \alpha\boldsymbol{x}[n], \tag{4.2}$$

em que $\boldsymbol{x}[n]$ corresponde à amostra atual, $\boldsymbol{y}[n-1]$ à saída anterior do filtro e, α é o fator de suavização e, variando no intervalo fechado $[0, 1]$. Para valores próximos de 1, o fator de suavização é menor, enquanto que para valores próximos de 0, maior. Dessa maneira, pode-se definir um vetor de características $\boldsymbol{\rho}$, contendo todas as posições de todas juntas excluindo a junta de referência, uma vez que após o processo de normalização esta é sempre um vetor nulo, totalizando um vetor de dimensão 30 tal que

$$\boldsymbol{\rho} = [\tilde{\boldsymbol{p}}_1, \cdots, \tilde{\boldsymbol{p}}_{11}] \; = \; [\tilde{x}_1, \tilde{y}_1, \tilde{z}_1, \cdots, \tilde{x}_{11}, \tilde{y}_{11}, \tilde{z}_{11}], \tag{4.3}$$

aplica-se então nele o filtro EMA obtendo o vetor $\boldsymbol{\rho_s}$ tal que

$$\boldsymbol{\rho_s}[n] = (1-\alpha)\boldsymbol{\rho_s}[n-1] + \alpha\boldsymbol{\rho}[n]. \tag{4.4}$$

Em continuação, de acordo com a Equação 4.4, para uma operação em tempo real, temos que armazenar o estado anterior do vetor de posições das juntas filtrado, ou seja, $\boldsymbol{\rho_s}[n-1]$. Além de filtrar grandes variações na posição das juntas, o que pode causar falhas na classificação de um dado instante como gesto em execução ou repouso, esta filtragem também funciona como um estimador se alguma das juntas não for detectada.

A utilização da posição das juntas é sempre questionável até que estas passem por algum processo de normalização. Este processo garante invariância em relação ao referencial, uma vez que adota-se uma das juntas para tal. Além disso, neste processo os comprimentos das conexões entre juntas são escalados. Com isso, garante-se que mesmo com diferentes indivíduos executando o mesmo gesto, o comportamento do pose ao longo do tempo será parecido, tornando o detector mais eficiente.

4.2.1.2 Velocidade das juntas

Esta é uma característica temporal, que corresponde à velocidade das juntas em um instante de tempo. Em (NEVEROVA et al., 2014), o vetor de velocidades $\delta\boldsymbol{\rho_s}$ é calculado por

$$\delta\boldsymbol{\rho_s}[n] = \boldsymbol{\rho_s}[n+1] - \boldsymbol{\rho_s}[n-1], \tag{4.5}$$

ou seja, utilizam-se uma amostra anterior e uma futura. Este método é conhecido como diferença central, que apesar de possuir um menor erro de aproximação, necessita de um valor futuro, o que não é possível em sistemas de tempo real. Dessa maneira, a primeira derivada da posição é calculada com diferença para trás, na qual

$$\delta\boldsymbol{\rho_s}[n] = \boldsymbol{\rho_s}[n] - \boldsymbol{\rho_s}[n-1]. \tag{4.6}$$

Dessa forma, teremos um vetor $\delta\boldsymbol{\rho_s}$ composto pela velocidade das coordenadas de cada uma das 10 juntas utilizadas no vetor de posições, compreendendo um vetor de características de dimensão 30.

4.2.1.3 Aceleração das juntas

Assim como a velocidade, o cálculo da aceleração utilizando-se em (NEVEROVA et al., 2014; ZANFIR MARIUS LEORDEANU, 2013) é feito utilizando uma aproximação que necessita de valores futuros, como está apresentado na Equação 4.7 e que corresponde

à calcular a aceleração utilizando diferença central com os valores de velocidade, expressos em termos de posição.

$$\delta^2 \boldsymbol{\rho_s}[n] = \boldsymbol{\rho_s}[n+2] + \boldsymbol{\rho_s}[n-2] - 2\boldsymbol{\rho_s}[n]. \tag{4.7}$$

Apesar desta expressão possuir um pequeno erro de aproximação, assim como no cálculo da velocidade, ela não é viável para uso em tempo real porque utiliza valores futuros. Outra abordagem é utilizar o valor passado da velocidade e o valor atual, semelhante ao que foi feito anteriormente, o que resulta em uma aceleração aproximada por

$$\delta^2 \boldsymbol{\rho_s}[n] = \delta\boldsymbol{\rho_s}[n] - \delta\boldsymbol{\rho_s}[n-1], \tag{4.8}$$

e que pode ser escrita em termos de posição, utilizando a Equação 4.6 tal que

$$\delta^2 \boldsymbol{\rho_s}[n] = \boldsymbol{\rho_s}[n] - \boldsymbol{\rho_s}[n-1] - \boldsymbol{\rho_s}[n-1] + \boldsymbol{\rho_s}[n-2], \tag{4.9}$$

resultando em

$$\delta^2 \boldsymbol{\rho_s}[n] = \boldsymbol{\rho_s}[n] - 2\boldsymbol{\rho_s}[n-1] + \boldsymbol{\rho_s}[n-2]. \tag{4.10}$$

Assim, torna-se necessário armazenar a pose de dois instantes passados. Ao fim, tem-se o vetor de acelerações $\delta^2\boldsymbol{\rho_s}$ também de dimensão igual a 30, semelhante ao de posição e velocidade das juntas.

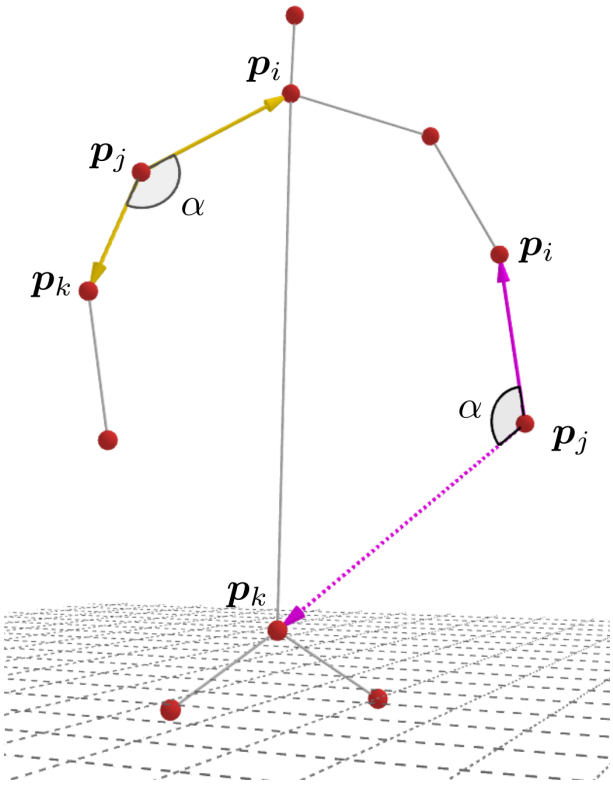
4.2.1.4 Ângulos de inclinação

Estes ângulos são formados por todas as triplas conectadas anatomicamente, além de dois ângulos virtuais formados pelas triplas (*LeftElbow*, *LeftWrist*, *HipCenter*) e (*RightElbow*, *RightWrist*, *HipCenter*). Cada junta da tripla é representada pelos subíndices i, j e k, sendo a junta de índice j aquela na qual o ângulo é medido. O ângulo é calculado por

$$\alpha_{(i,j,k)} = \arccos \frac{(\boldsymbol{p}_k - \boldsymbol{p}_j) \cdot (\boldsymbol{p}_i - \boldsymbol{p}_j)}{\|\boldsymbol{p}_k - \boldsymbol{p}_j\| \cdot \|\boldsymbol{p}_i - \boldsymbol{p}_j\|}. \tag{4.11}$$

Na Figura 27 estão representados dois ângulos α, sendo um deles, destacado por dois vetores na cor amarela, que é formado por juntas que estão anatomicamente conectadas. Já o outro ângulo α, este ilustrado na cor magenta, é formado por juntas que não são anatomicamente conectadas. Isto é mostrado pelo vetor $\boldsymbol{p}_k - \boldsymbol{p}_j$, que está desenhado com seu corpo pontilhado, e conecta a junta *RightWrist* à junta *HipCenter*. Todos os ângulos de inclinação formam o vetor de característica $\boldsymbol{\alpha}$ que possui dimensão igual à 14.

Figura 27 – Representação dos vetores de dois ângulos de inclinação, sendo aquele representado por vetores na cor amarela formado por juntas conectadas anatomicamente, enquanto que o ângulo formado pelos vetores na cor magenta possui uma conexão que não corresponde a uma conexão anatômica, destacada por um vetor pontilhado.



Fonte: Produção do próprio autor.

4.2.1.5 Ângulos de azimute

Para o cálculo dos ângulos de azimute e flexão, que será apresentado na próxima seção, é necessário determinar um sistema de coordenadas tridimensional que será utilizado como referência. Este deve sofrer a menor variação possível ao longo do tempo, e ser calculado de uma maneira que seja consistente independente da posição da pessoa em relação ao sistema de captura, dentre outros fatores, como movimentos bruscos, imprecisão na detecção, etc. Em (NEVEROVA et al., 2014), este referencial é determinado aplicando PCA (do inglês, *Principal Component Analysis*) em um conjunto de 6 juntas do torso, são elas: *HipCenter*, *RightHip*, *LeftHip*, *CenterShoulder*, *RightShoulder* e *LeftShoulder*. A partir das 3 componentes com maior contribuição, determinavam-se 3 eixos ortonormais, que representavam os eixos $\boldsymbol{u}_x$, aproximadamente paralelo com a linha dos ombros, $\boldsymbol{u}_y$, alinhado com a espinha, e $\boldsymbol{u}_z$ perpendicular ao torso.

Contudo, para a finalidade que se deseja, o cálculo do PCA não se faz necessário. Basta executar a etapa inicial do PCA, que consiste em determinar os auto-valores e seus

respectivos auto-vetores da matriz de covariância dos dados de entrada, neste caso, as coordenadas das 6 juntas do torso. O cálculo dos auto-valores e auto-vetores pode ser feito através da decomposição de valores singulares, que normalmente é realizada através de um processo iterativo. Mesmo sendo algo relativamente rápido, por ser um processo iterativo, é possível que ocorra uma parada precoce, resultando em componentes que não representam bem os dados de entrada. Isso pode resultar em eixos não consistentes ao longo do tempo, no sentido de que, por exemplo, o vetor do $eixo - y$ pode não estar devidamente alinhado com a espinha, como se deseja, ocasionando grandes variações dos ângulos calculados quando se toma o mesmo como referência.

Além disso, ao se calcular os três autovetores que correspondem aos três eixos do referencial que se deseja obter, deve-se determinar qual autovetor corresponde a cada um dos eixos coordenados, isto é, qual corresponde a $\boldsymbol{u}_x$, $\boldsymbol{u}_y$ e $\boldsymbol{u}_z$. Neste caso, pode-se verificar quais vetores encontram-se mais alinhados com a linha formada pelas juntas do ombro, e aquele que se alinha melhor com a espinha, obtendo-se então $\boldsymbol{u}_x$ e $\boldsymbol{u}_y$, e consequentemente $\boldsymbol{u}_z$ por exclusão. Entretanto, este artifício pode fazer com que, dependendo dos vetores obtidos, ocorra uma inversão entre $\boldsymbol{u}_x$ e $\boldsymbol{u}_y$. Isto ocorrendo, fará com que os ângulos calculados a partir deste eixos coordenados sofram uma grande variação em relação à amostra anterior, podendo causar classificação incorreta daquele instante.

Portanto, devido aos fatores considerados anteriormente, optou-se por não calcular os vetores do sistema de coordenadas como foi feito em (NEVEROVA et al., 2014). Os vetores que formam o sistema de coordenadas necessário para o cálculo dos ângulos dessa seção e da próxima são calculados apenas por geometria, garantindo que os três vetores unitários são ortonormais, isto é, são unitários e formam um ângulo reto entre si. Na Figura 28 traz uma ilustração com as 11 juntas utilizadas e suas conexões, além dos vetores necessários para o entendimento.

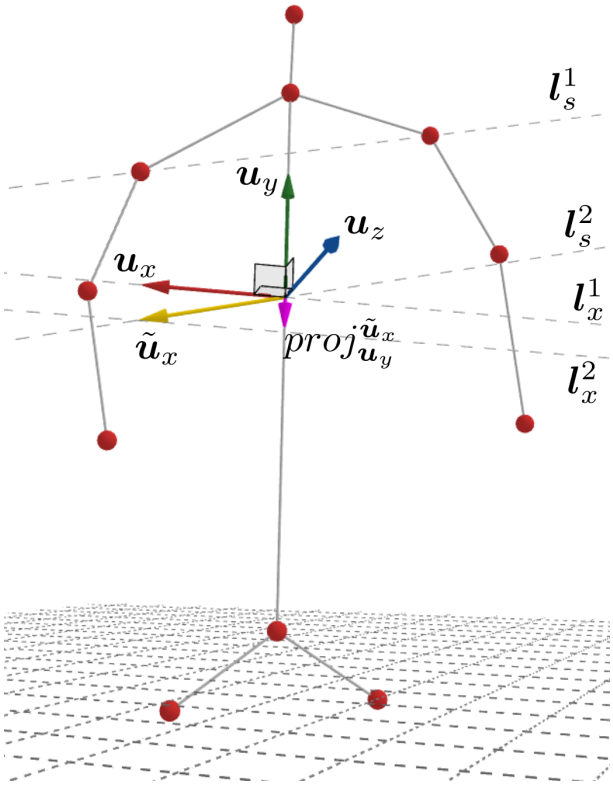
Foi escolhido um ponto qualquer sob a linha da espinha, para desenhar o referencial e demais vetores. Sobre esta linha, no sentido das juntas *HipCenter*→*CenterShoulder*, define-se o vetor unitário $\boldsymbol{u}_y$, desenhado na cor verde na Figura 28. Sobre a linha imaginária $\boldsymbol{l}_s^2$, que por sua vez é paralela à linha $\boldsymbol{l}_s^1$ e passa pelas duas juntas do ombro esquerdo e direito, define-se o vetor unitário $\tilde{\boldsymbol{u}}_x$, no sentido *LeftShoulder*→*RightShoulder*, representado na cor amarela. Este por sua vez pode não ser ortogonal ao vetor $\boldsymbol{u}_y$. Assim, realiza-se uma ortogonalização da base formada pelos vetores $\tilde{\boldsymbol{u}}_x$ e $\boldsymbol{u}_y$, obtendo-se o vetor $\boldsymbol{u}_x$, desenhado na cor vermelha e definido por

$$\boldsymbol{u}_x = \tilde{\boldsymbol{u}}_x - proj_{\boldsymbol{u}_y}^{\tilde{\boldsymbol{u}}_x} = \tilde{\boldsymbol{u}}_x - \boldsymbol{u}_y \cdot (\tilde{\boldsymbol{u}}_x \cdot \boldsymbol{u}_y)\,, \tag{4.12}$$

em que o vetor $proj_{\boldsymbol{u}_y}^{\tilde{\boldsymbol{u}}_x}$, representado na cor rosa, corresponde à projeção do vetor $\tilde{\boldsymbol{u}}_x$ sobre o vetor $\boldsymbol{u}_y$. De posse dos vetores $\boldsymbol{u}_x$ e $\boldsymbol{u}_y$, obtém-se $\boldsymbol{u}_z$ calculando-se o produto

vetorial $\boldsymbol{u}_x \times \boldsymbol{u}_y$. Dessa maneira, garante-se que os três vetores, cada um para um eixo de coordenadas, são ortogonais entre si.

Figura 28 – Representação dos vetores e linhas necessárias para se obter o referencial tridimensional utilizado para o cálculo dos ângulos de azimute e de flexão.



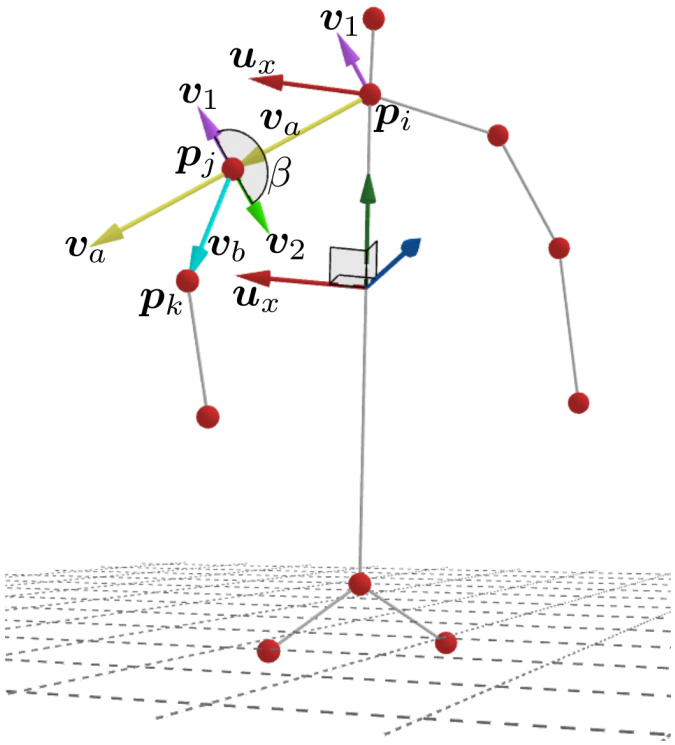
Fonte: Produção do próprio autor.

Com os vetores do referencial calculados, pode-se então determinar os ângulos de azimute. Estes por sua vez são calculados para todos os pares de conexões adjacentes, isto é, as mesmas trincas utilizadas para os ângulos de inclinação com exceção das utilizadas para os ângulos virtuais. Entretanto, se acrescenta uma trinca formada por 2 conexões virtuais, que inclui as juntas (*LeftWrist*, *HipCenter*, *RightWrist*). Cada ponto de junta da tripla também é representado pelos subíndices i, j e k. Na Figura 29, foi escolhida uma trinca para exemplificar o processo de obtenção dos ângulos de azimute. Cada uma das juntas das trincas estão identificadas por $\boldsymbol{p}_i$, $\boldsymbol{p}_j$ e $\boldsymbol{p}_k$. Para o cálculo dos ângulo de azimute, é necessário definir dois vetores: $\boldsymbol{v}_a = \boldsymbol{p}_j - \boldsymbol{p}_i$ e $\boldsymbol{v}_b = \boldsymbol{p}_k - \boldsymbol{p}_j$, representados pelas cores amarelo e ciano, respectivamente. Além disso, o vetor $\boldsymbol{u}_x$ também é utilizado e foi redesenhado com sua base sobre o ponto $\boldsymbol{p}_i$.

O ângulo de azimute corresponde então ao ângulo da projeção do vetor $\boldsymbol{u}_x$ e do vetor $\boldsymbol{v}_b$ sobre o plano perpendicular ao vetor $\boldsymbol{v}_a$, cada uma das projeções é definida respectivamente pelos vetores $\boldsymbol{v}_1$ e $\boldsymbol{v}_2$. Em termos práticos, podemos fazer um paralelo com a representação de uma rotação utilizando a fórmula de Rodrigues, em que se tem um

vetor que indica o eixo de rotação, neste caso representado pelo vetor $\boldsymbol{v}_a$, e com apenas um ângulo, é possível determinar uma rotação no espaço tridimensional. Pode-se portanto, determinar a orientação de $\boldsymbol{v}_b$ em relação ao eixo $\boldsymbol{u}_x$, utilizando o eixo de rotação $\boldsymbol{v}_a$. Assim, projetam-se ambos os vetores sobre o plano perpendicular à $\boldsymbol{v}_a$, obtendo $\boldsymbol{v}_1$ e $\boldsymbol{v}_2$, e calcula-se então o ângulo β entre eles.

Figura 29 – Representação dos vetores necessários para o cálculo do ângulo de azimute para uma das trincas de juntas, além dos vetores que representam o sistema de coordenadas utilizado como referência para o cálculo dos ângulos.



Fonte: Produção do próprio autor.

Para o vetor $\boldsymbol{v}_1$, podemos obtê-lo a partir de

$$\boldsymbol{v}_1 = \boldsymbol{u}_x - \boldsymbol{v}_a \frac{\boldsymbol{u}_x \cdot \boldsymbol{v}_a}{\|\boldsymbol{v}_a\|^2} = \boldsymbol{u}_x - (\boldsymbol{p}_j - \boldsymbol{p}_i) \frac{\boldsymbol{u}_x \cdot (\boldsymbol{p}_j - \boldsymbol{p}_i)}{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|^2}, \tag{4.13}$$

enquanto que o vetor $\boldsymbol{v}_2$, obtém-se da seguinte equação

$$\boldsymbol{v}_2 = \boldsymbol{v}_b - \boldsymbol{v}_a \frac{\boldsymbol{v}_b \cdot \boldsymbol{v}_a}{\|\boldsymbol{v}_a\|^2} = (\boldsymbol{p}_k - \boldsymbol{p}_j) - (\boldsymbol{p}_j - \boldsymbol{p}_i) \frac{(\boldsymbol{p}_k - \boldsymbol{p}_j) \cdot (\boldsymbol{p}_j - \boldsymbol{p}_i)}{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|^2}. \tag{4.14}$$

Por fim, a partir dos dois vetores $\boldsymbol{v}_1$ e $\boldsymbol{v}_2$, obtém-se o ângulo β por

$$\beta = \arccos \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{\|\boldsymbol{v}_1\| \, \|\boldsymbol{v}_2\|}. \tag{4.15}$$

O vetor de características correspondente aos ângulos de azimute possui dimensão igual a 13, e é definido por $\boldsymbol{\beta}$.

4.2.1.6 Ângulos de flexão

Assim como os ângulos de azimute apresentados anteriormente, os ângulos de flexão são calculados utilizando o sistema de coordenadas obtido a partir das juntas do torço. O ângulo de flexão é medido entre o vetor formado pela posição de uma junta em relação à junta de referência, e o vetor $\boldsymbol{u}_z$. Portanto, para cada *i-ésima* junta, com exceção da junta de referência, pois esta resultaria em um vetor nulo, define-se o ângulo de flexão

$$\gamma = \arccos \frac{\boldsymbol{u}_z \cdot \boldsymbol{p}_i}{\|\boldsymbol{p}_i\|}. \tag{4.16}$$

Na Figura 30, estão representados os vetores $\boldsymbol{u}_z$ e $\boldsymbol{p}_i$, posicionados na junta de referência, e o ângulo γ formado entre eles, que corresponde a um dos ângulos que flexão que formam o vetor de características $\boldsymbol{\gamma}$. Este por sua vez possui dimensão igual a 10, uma vez que são calculados ângulos de flexão para todas as 11 juntas utilizadas, exceto a junta de referência.

4.2.1.7 Distâncias em pares

Este vetor de características corresponde à distância entre todos os pares possíveis de juntas utilizadas. Dessa maneira, temos um total de distâncias em pares igual à combinação 2 a 2 de 11 juntas, que resulta em um vetor de características $\boldsymbol{\rho}$ de dimensão igual a $\binom{11}{2} = \frac{11!}{2!9!} = 55$. Para um par de juntas $(\boldsymbol{p}_i,\ \boldsymbol{p}_j)$, a distância é definida por

$$\rho = \|\boldsymbol{p}_j - \boldsymbol{p}_i\|. \tag{4.17}$$

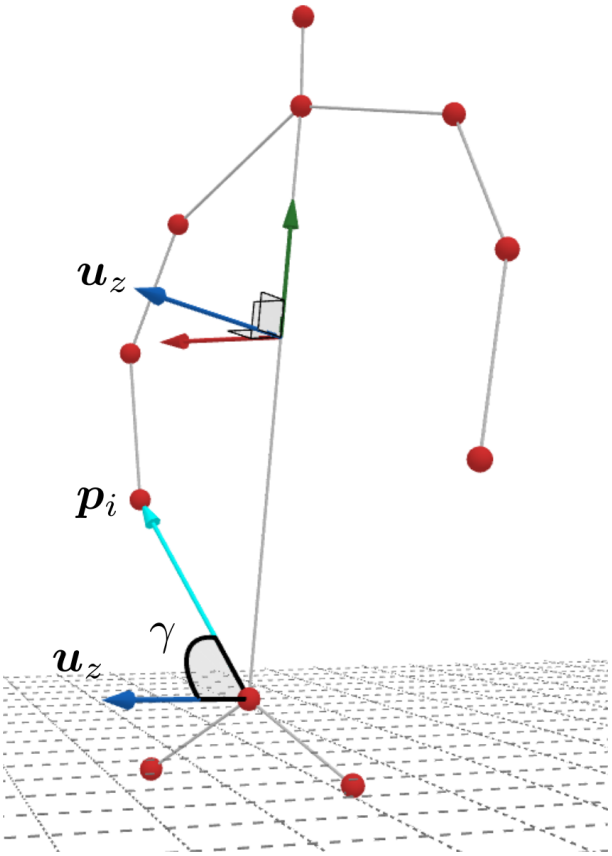
4.2.2 Normalização

Na seção anterior foram expostas 7 características que são extraídas a partir da pose devidamente normalizada. Essas características são de natureza espacial e temporal, e agrupadas compõem um único vetor $\boldsymbol{D}$, definido por

$$\boldsymbol{D} = \left[\boldsymbol{p}, \delta\boldsymbol{p}, \delta^2\boldsymbol{p}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\rho}\right]. \tag{4.18}$$

Como foi mencionado no fim da subseção que descreve cada uma das características que compõem o vetor $\boldsymbol{D}$, os vetores possuem respectivamente dimensões iguais a 30, 30, 30, 14, 13, 10 e 55, o que produz um vetor $\boldsymbol{D}$ de dimensão igual a 182. Este vetor então passa por um processo de *standardization*, que consiste em escalar cada uma das 182 características para que tenha média igual a 0 e desvio padrão igual a 1. A importância

Figura 30 – Representação de um ângulo de flexão γ, cujo vetor associado à junta, $\boldsymbol{p}_i$, está ilustrado na cor ciano.



Fonte: Produção do próprio autor.

deste processo se deve ao fato do vetor de características ser composto por 7 vetores que possuem naturezas diferentes e, portanto, diferentes unidades e escalas. Além disso, este processo é importante para se introduzir tais dados em um algoritmo de aprendizagem de máquinas que utilizam processos de otimização, nos quais certos pesos envolvidos neste processo podem atualizar mais rapidamente que outros, gerando modelos que não representam bem o conjunto de dados apresentado no treinamento.

Dessa maneira, a partir do *dataset* de treino, computa-se para cada uma das características seu valor médio e desvio padrão, obtendo um vetor para cada uma dessas medidas estatísticas, respectivamente $\boldsymbol{\mu}$ e $\boldsymbol{\sigma}$, também de dimensão igual a 182. O vetor $\boldsymbol{D}$ escalado é então obtido por

$$\bar{\boldsymbol{D}} = \frac{\boldsymbol{D} - \boldsymbol{\mu}}{\boldsymbol{\sigma}}. \tag{4.19}$$

De posse do vetor de característica $\bar{\boldsymbol{D}}$ devidamente escalado em relação à média e desvio padrão, o próximo passo é utilizá-lo como a entrada de algum algoritmo de

aprendizagem de máquinas, para que seja possível classificar um instante da execução de um gesto ou ação como repouso ou executando.

4.3 Classificador

A tarefa de localizar um gesto ou ação consiste em determinar quando este inicia e quando termina, seja ela executada com imagens, poses de esqueletos ou ambas informações ao longo de uma sequência. Quando também se deseja realizar a classificação, independente se localização e classificação ocorrem em tarefas separadas ou não, é importante definir o pré-processamento a ser realizado nos dados de entrada, e o que se espera na saída do algoritmo. Neste trabalho, a localização da pose é feita em uma tarefa isolada da classificação. Em uma visão mais alto nível, para uma sequência de poses que o sistema receberá em sua entrada, deseja-se obter em sua saída uma sinalização nos instantes de início e fim do gesto.

Assim sendo, a partir de cada observação da pose de um indivíduo, a informação da localização das juntas pertinentes mencionadas anteriormente é então processada, obtendo-se o vetor de características. Pensando apenas na tarefa de localizar o gesto, podemos primeiro classificar se um determinado instante corresponde a repouso, ou a qualquer instante da execução de um gesto ou ação. Ou seja, de maneira simples significa classificar repouso ou movimento que seja relacionado à execução de gestos ou ações. Dessa maneira, esta etapa consiste em uma classificação binária, que, independente do classificador escolhido, sua saída será a probabilidade de aquele determinado instante pertencer à classe MOVIMENTO ou REPOUSO.

Independente do método a ser escolhido para tal tarefa, deve-se levar em consideração não apenas a acurácia alcançada, mas também o tempo médio de execução para a amostra inserida no modelo, uma vez que se deseja uma execução em tempo real. Em (NEVEROVA et al., 2014), o classificador consistia em apenas uma rede neural MLP (do inglês, *Multilayer perceptron*) com uma camada oculta de 300 neurônios. A mesma arquitetura foi utilizada como ponto de partida neste trabalho, contudo, como foram feitas modificações na construção do vetor de características para atender o requisito de tempo real, alterações foram necessárias para se obter um resultado satisfatório. Contudo, também foram realizados testes com *Random Forest*. Essa escolha foi baseada no fato desses algoritmos possuírem naturezas distintas e, portanto, poderem apresentar diferentes resultados para um mesmo dado de entrada, abrindo a possibilidade de se utilizar uma combinação dos dois para gerar uma saída mais precisa. Na Seção 4.5, serão apresentados os testes realizados para cada um dos modelos escolhidos, descrevendo o procedimento adotado, e apontando os pontos avaliados para a escolha do modelo a ser utilizado.

A saída dos dois modelos possui o mesmo formato, isto é, ambos geram um vetor de

duas posições, cada uma referente a probabilidade de cada uma das classes. Estes valores são então processados para determinar o instante que corresponde o início e o fim de um gesto ou ação. Na próxima seção será explicado como é feito este processo.

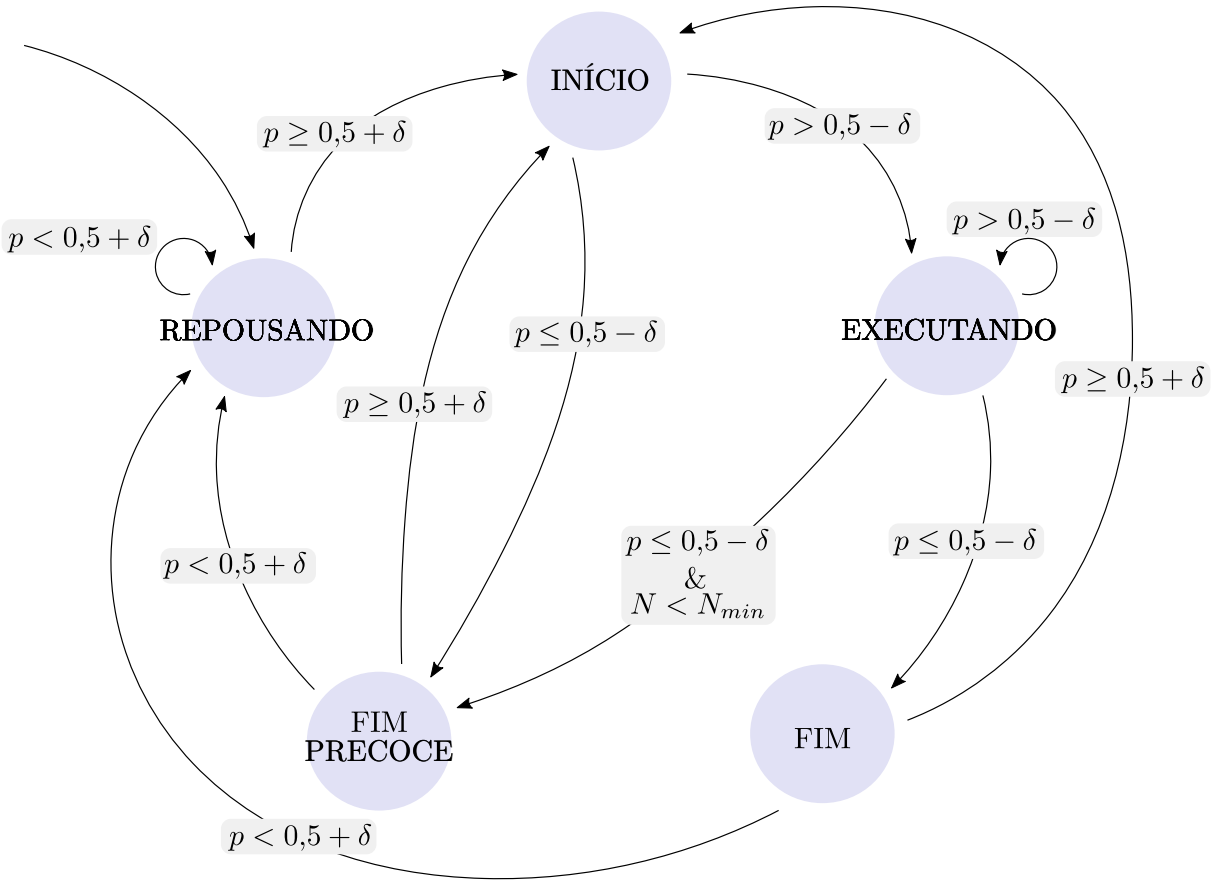
4.4 Processo de localização

Independente de qual modelo tenha sido escolhido, sua saída corresponde à probabilidade de aquele instante pertencer à classe REPOUSO ou MOVIMENTO. Deve-se portanto realizar um pós processamento para determinar o que representa a mudança de estado entre cada uma dessas duas possibilidades. De maneira resumida, a mudança de REPOUSO para MOVIMENTO indica o início de uma execução, e, o contrário, o fim desta. Assim, foram definidos cinco estados possíveis os quais podem ser obtido a partir de uma sequência de probabilidades resultante da classificação do vetor de características em um determinado instante. São eles: REPOUSANDO, INÍCIO, EXECUTANDO, FIM e FIM PRECOCE.

A transição de estados é feita de acordo com o valor da probabilidade da classe MOVIMENTO p. Com intuito de se evitar transições de estado inadequadas, representando um início ou fim não esperado de uma execução, optou-se por utilizar dois limiares de decisão, ambos representados por um deslocamento δ em torno de 0,5. Dessa forma, transições de estados só são feitas quando se ultrapassa o limite superior ou inferior. Isto equivale a criar uma terceira saída possível para o modelo, que representa uma indefinição. Além da probabilidade p, o número de instantes de uma execução são contabilizados em N e, caso a probabilidade atinja um valor $p \leq 0{,}5 - \delta$, e a quantidade de instantes seja tal que $N < N_{min}$, a transição é então feita para o estado FIM PRECOCE, indicando que aquela execução não teve amostras suficientes para ser considerada um gesto ou ação podendo, assim, ser descartada. Na Figura 31 está ilustrada a máquina de estados que descreve este processo.

Na Figura 32, pode-se observar o gráfico com as probabilidades obtidas pelo modelo para um vídeo de exemplo. De acordo com o seu valor, esta probabilidade pode corresponder à MOVIMENTO, REPOUSO ou INDEFINIDO, cada um representado por marcadores diferentes no gráfico. No instante inicial, a probabilidade é tal que $p < 0{,}5 - \delta$ e, portanto, a saída da máquina de estados permanece no estado REPOUSANDO até que se atinge um valor $p \geq 0{,}5 + \delta$, transitando para o estado INÍCIO. O estado permanece então em EXECUTANDO, uma vez que $p > 0{,}5 - \delta$, até que, a probabilidade assuma valores $p < 0{,}5 - \delta$, como o número de instâncias foi maior que o mínimo aceitável N_{min}, ocorre uma transição para o estado FIM. Em seguida, a saída permanece no estado REPOUSO até que p seja maior ou igual que $0{,}5 + \delta$ novamente. A quantidade mínima de amostras N_{min} está diretamente relacionada com a taxa de amostragem que o sistema está operando,

Figura 31 – Representação da máquina de estados utilizada para determinar em qual instante a execução de um gesto ou ação se encontra.



Fonte: Produção do próprio autor.

e seu ajuste é importante pois evita que pequenos movimentos sejam interpretados como uma execução.

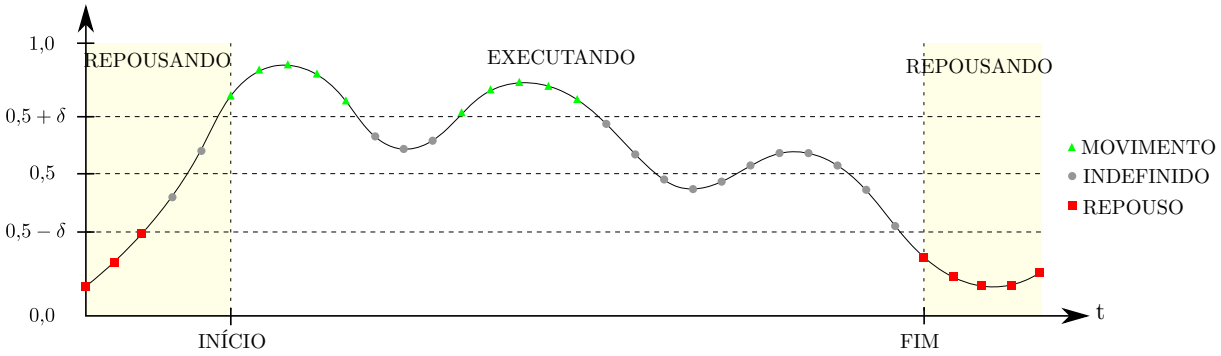
Outra execução está ilustrada na Figura 33. Esta, por sua vez, teve seu fim indicado pelo estado FIM PRECOCE, já que a quantidade de amostras da execução não atingiu o mínimo aceitável N_{min}.

Como já mencionado, o processo descrito nesta sessão pode ser aplicado para qualquer que seja o modelo que classifica cada instante de uma execução. Como parâmetros de ajuste, temos o fator δ, e o número mínimo de amostras N_{min}. Na próxima seção serão apresentados os resultados obtidos com diferentes ajustes destes parâmetros, e como estes influenciam no desempenho do processo de localização de um gesto ou ação.

4.5 Experimentos e Resultados

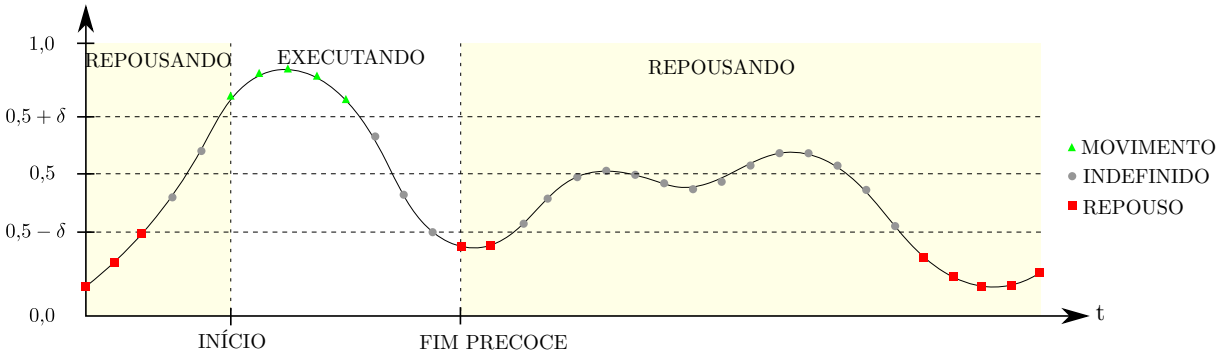
Esta sessão compreende desde os experimentos realizados para a escolha do modelo utilizado para a classificação dos instantes, bem como a determinação dos parâmetros

Figura 32 – Ilustração do processo de localização de um gesto ou ação, mostrando a probabilidade de cada instante para a classe MOVIMENTO. Os limiares de decisão estão indicados por três linhas tracejadas horizontais. Este exemplo teve um fim considerado normal pois apresentou a quantidade de amostras maior que a mínima aceitável para uma execução.



Fonte: Produção do próprio autor.

Figura 33 – Ilustração do processo de localização de um gesto ou ação em que o número de amostras no intervalo de execução foi menor que o mínimo aceitável.



Fonte: Produção do próprio autor.

da etapa de localização do gesto, e finalmente os experimento utilizando os parâmetros e modelo selecionados. Para isso, é importante a escolha de um *dataset* para a validação de cada um dos processos.

4.5.1 *Dataset Montanbano v2*

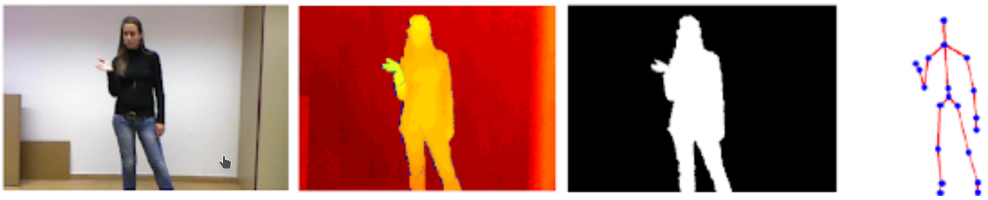
Na área de detecção e classificação de gestos existem diversos tipos de *datasets*. Estes podem variar na maneira a qual os dados são capturados e rotulados, além das fontes de dados disponíveis (ZHANG, 2016). De maneira geral, como a tarefa de localização e classificação de gestos e ações estão diretamente relacionadas, mas podem ser desenvolvidas de forma separada, é comum encontrar bases de dados que contemplem ambas as tarefas. Contudo, especificamente para a tarefa de localização, é necessário que haja na rotulação a indicação do início e do fim de uma execução. Ou ainda, caso não haja, se houver a

rotulação de quais são os instantes em que uma execução está ocorrendo, é possível inferir os momentos de seu início e fim.

Outro ponto importante sobre a escolha da base de dados é que esta contenha momentos em que a pessoa que está executando o gesto ou ação esteja em repouso. Isto porque alguns *datasets* são compostos de clipes em que a execução começa exatamente no primeiro *frame*. Estes são normalmente utilizados apenas para a tarefa de classificação. Dessa maneira, optou-se por uma base de dados que foi utilizada em um desafio conhecido como *ChaLearn Looking at People*[1], realizado no ano de 2014 (ESCALERA et al., 2015). Este possuiu três trilhas diferentes, a primeira voltada para a recuperação de pose humana, a segunda para o reconhecimento de ações e interações, e a terceira voltada para o reconhecimento de gestos. A última das trilhas utilizou a base de dados que foi adotada neste trabalho.

Esta base de dados é composta por aproximadamente 14000 execuções, de um conjunto de 20 gestos italianos emblemáticos. Todas as execuções foram gravadas com o usuário executando uma sequência de gestos distintos de frente para um *Kinect I*. A sequência de execuções envolvia não só o gesto, mas também fala. A captura realizada foi multimodal, isto é, possuía diversas fontes, são elas: imagem colorida, mapa de profundidade, silhueta e pose do corpo humano, mostradas respectivamente na Figura 34, além do áudio. A pose do corpo humano compreende as posições de juntas no espaço. As juntas disponíveis são aquelas apresentadas para o modelo *Kinect I* no Quadro 2.

Figura 34 – Exemplo das fontes de dados disponíveis no *Montalbano*. Da esquerda para a direita, imagem RGB, mapa de profundidade, silhueta e pose humana.



Fonte: Adaptado de (ESCALERA et al., 2015).

O Quadro 7 estão apresentadas as principais características do *dataset* escolhido. Vale ressaltar que esta base de dados possui a rotulação para 20 classes de gestos diferentes, entretanto, esta teve de ser adaptada para o problema aqui abordado. As rotulações entregues indicam o *frame* de início e fim, e a qual classe a execução pertence. A partir desta informação, foram gerados rótulos que indicavam a classe MOVIMENTO para estes intervalos, e REPOUSO para todos os outros instantes. Após a adaptação da rotulação, para as sequências de treino foi obtido um percentual de 42,54% de instantes com o rótulo

[1] http://chalearnlap.cvc.uab.es/

MOVIMENTO, 42,32% para as de teste e 42,18% para as de validação. Isso mostra que as classes estão pouco desbalanceadas, o que é favorável para o treinamento dos modelos.

Quadro 7 – Principais características do *dataset Montalbano v2*.

Sequências de treino	339 (7.754 gestos)
Sequências de teste	287 (3.362 gestos)
Sequências de validação	276 (2.742 gestos)
Duração das sequências	1$\tilde{2}$ minutos
FPS	20
Classes de Gestos	20
Frames rotulados	1.720.800

Fonte: Adaptado de (ESCALERA et al., 2015).

4.5.2 Treinamento e validação dos modelos

Como mencionado na Seção 4.3, dois classificadores diferentes foram testados: MLP e *Random Forest*, a fim de encontrar aquele que apresenta o melhor custo benefício entre tempo de execução e acurácia. O primeiro deles, MLP, foi o mesmo utilizado em (NEVEROVA et al., 2014). Nesta Seção será apresentada a metodologia para o treinamento de cada um dos classificadores, e os resultados obtidos para cada um deles.

4.5.2.1 MLP

Para o classificador do tipo *Multilayer Perceptron*, foram testadas 5 configurações diferentes, apresentadas no Quadro 8. Todos modelos possuem em comum as camadas de entrada, com um total de 182 neurônios, uma vez que esta é a dimensão do vetor de características, e uma camada de saída com 2 neurônios, conectada a uma camada de ativação com função *Softmax*, que representa as classes MOVIMENTO e REPOUSO. Todas as demais camadas densas possuem *bias*, e a função de ativação conectada a sua saída é do tipo *ReLU*. As arquiteturas 3, 4 e 5 possuem regularização do tipo *Dropout* nas camadas ocultas, cuja taxa está apresentada também no Quadro 8.

Todos os treinamentos foram realizados com os hiperparâmetros apresentados no Quadro 9. Para cada um deles, foi acompanhado durante o treinamento o valor da acurácia e da função de custo, tanto para o conjunto de dados de treinamento quanto para o de validação. Os valores para cada um destes parâmetros estão apresentados nos gráficos da Figura 35. Apesar de terem sido realizados os treinamentos com um total de 200 épocas, apenas 150 estão apresentadas nos gráficos, pois para ambas as métricas os valores já haviam estabilizado.

A escolha das arquiteturas apresentadas foi feita de maneira incremental a partir da primeira, que foi baseada na utilizada em (NEVEROVA et al., 2014). Esta, que por sua

Quadro 8 – Diferentes configurações de MLP (*Multilayer perceptron*) testadas. As camadas **D** são do tipo densa com *bias*, contendo sua dimensão, **A** é do tipo ativação, e **Drop** corresponde a uma regularização do tipo *Dropout*, com sua respectiva taxa.

	Configuração #				
	1	2	3	4	5
Camadas	**In**[182]				
	D[300]	**D**[512]	**D**[2560]	**D**[1460]	**D**[1024]
	A[ReLU]	**A**[ReLU]	**Drop**[0.2]	**Drop**[0.35]	**Drop**[0.5]
	—	**D**[128]	**A**[ReLU]	**A**[ReLU]	**A**[ReLU]
	—	**A**[ReLU]	**D**[640]	**D**[366]	**D**[256]
	—	—	**Drop**[0.2]	**Drop**[0.35]	**Drop**[0.5]
	—	—	**A**[ReLU]	**A**[ReLU]	**A**[ReLU]
	Out[2]				
	A[Softmax]				

Fonte: Produção do próprio autor.

Quadro 9 – Hiperparâmetros utilizados nos treinamentos das diferentes arquiteturas de MLP para o classificador do localizador de gestos.

Tamanho do *batch*	32
Número de épocas	200
Learning-rate	5×10^{-2}
Learning-decay	5×10^{-3}
Função de custo	*Binary Crossentropy*
Otimizador	SGD

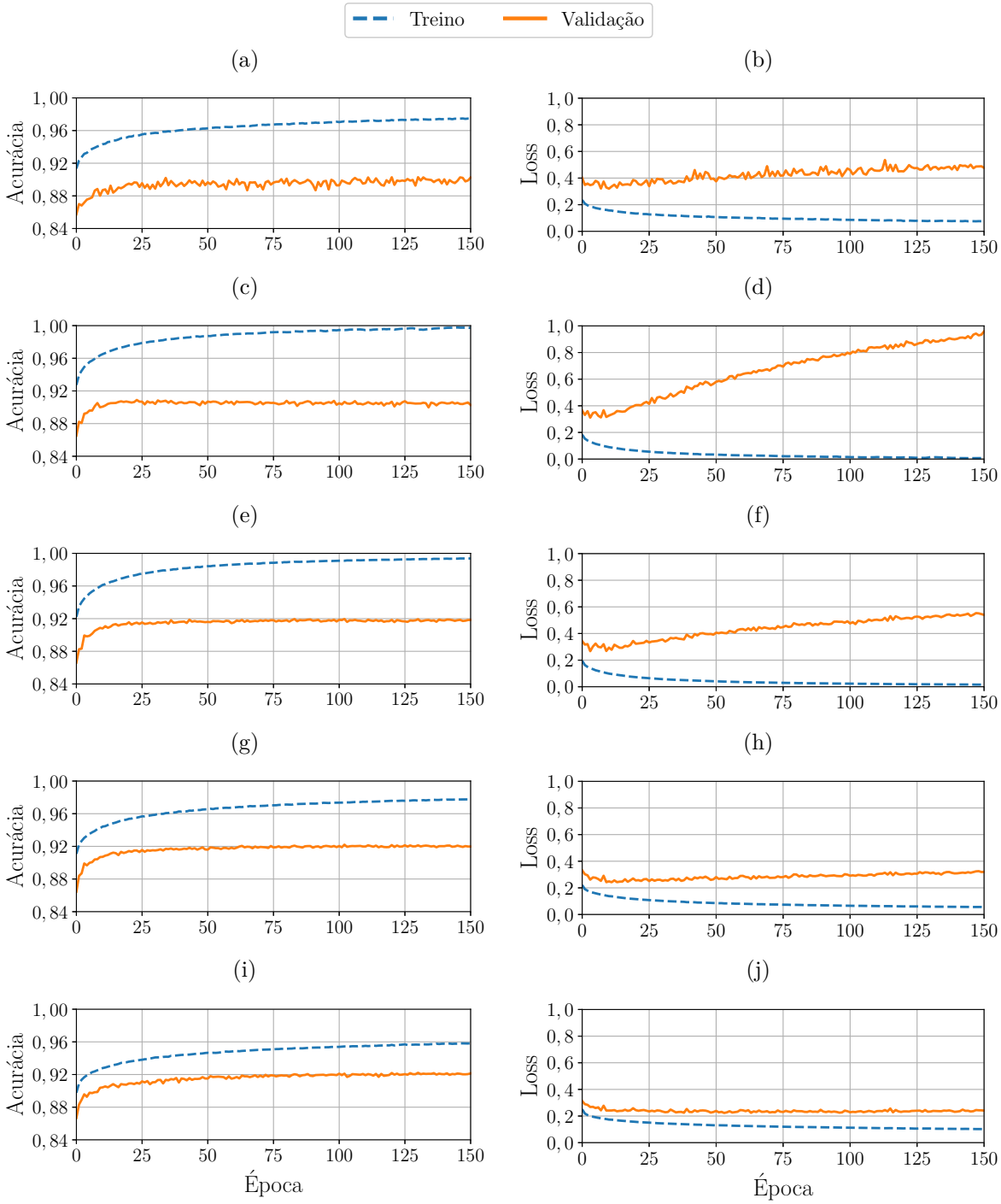
Fonte: Produção do próprio autor.

vez apresenta apenas uma camada oculta com 300 neurônios, não foi capaz de aprender o conjunto de dados de treinamento mesmo após 150 épocas, como pode ser observado no gráfico de acurácia da Figura 35a.

Isso indica que sua arquitetura não possui graus de liberdade suficientes para formar uma região de fronteira que separe todas as amostras apresentadas durante o treino. Assim, o próximo passo foi aumentar a capacidade de aprendizado do modelo, incluindo uma nova camada e aumentando a quantidade de neurônios destas camadas. Este modelo então contém duas camadas ocultas, a primeira com 512 neurônios e a segunda com 128.

Para a segunda configuração, com 150 épocas de treinamento, o modelo atingiu uma acurácia próxima de 100% para o conjunto de dados de treino, como pode ser observado na Figura 35c, o que indica que este é capaz de aprender o conjunto de treinamento. Isto também pode ser visto no gráfico de função de custo da Figura 35d, que se aproxima de zero ao fim das 150 épocas. Contudo, espera-se que um modelo após treinado, ao ser apresentado a um conjunto de dados desconhecidos, apresente um bom desempenho. Como

Figura 35 – Resultados dos treinamentos realizados para diferentes configurações de modelos MLP, para os *datasets* de treino, apresentado em linhas tracejadas de cor azul, e validação em linhas contínuas na cor laranja. Para os gráficos da coluna da esquerda, encontram-se os valores de acurácia, enquanto que na outra coluna os valores de função de custo, ambos apresentados ao longo das épocas de treinamento. Cada par de gráficos de acurácia e função de custo representa uma das cinco configurações testadas.



Fonte: Produção do próprio autor.

pode ser visto na Figura 35c, a acurácia para o conjunto de validação ficou ligeiramente maior que 90%, além disso, o valor da função de custo para este conjunto de dados começou a aumentar com o passar das épocas. Este cenário indica que o modelo apresenta *overfitting*, isto é, ele é capaz de classificar muito bem amostras do conjunto de treinamento, mas não apresenta bom desempenho com dados não vistos.

Uma maneira de contornar este problema é adicionar algum mecanismo de regularização ao modelo, para evitar que ocorra o *overfitting*. Uma técnica muito utilizada é conhecida como *dropout* (SRIVASTAVA et al., 2014). Este método parte do princípio que, se utilizarmos vários modelos treinados com parâmetros diferentes, e combinarmos suas saídas, obtém-se um melhor resultado. Contudo, esta abordagem é custosa pois consiste em um grande número de processos de treinamento para cada diferente configuração. O que a técnica de *dropout* propõe, é desligar aleatoriamente conexões entre camadas, fazendo assim que a cada interação, uma diferente configuração do mesmo modelo seja utilizada. A quantidade de conexões que é desligada de forma aleatória é controlada por uma taxa percentual, que indica a quantidade de conexões que permanecerá ligada.

Assim, para a terceira configuração adicionou-se o *dropout* na saída de cada camada oculta. Para esta configuração foi utilizada uma taxa igual a 0,2, isto é, 20% dos neurônios da camada são ativados. Uma recomendação comum é que, ao adicionar o *dropout*, aumente-se a quantidade de neurônios para que, ao ocorrer o desligamento aleatório, a quantidade de neurônios ligados seja a mesma de antes. Dessa forma, as duas camadas ocultas desta configuração apresentam $512/0{,}2 = 2560$ e $128/0{,}2 = 640$ neurônios. No gráfico de acurácia para esta configuração, apresentado na Figura 35e, pode-se verificar duas coisas: o valor da acurácia para o conjunto de treino continua a se aproximar de 100%, com uma pequena redução, e houve um aumento na acurácia com o conjunto de validação. Já no gráfico de função de custo, Figura 35f, nota-se que, para o conjunto de validação, o valor apresentou uma redução considerável ao se comparar com a segunda configuração.

Contudo, mesmo adicionando o *dropout*, o modelo ainda apresenta comportamento que caracteriza como *overfitting*. Foram então testadas mais duas configurações, aumentando gradualmente o valor da taxa do *dropout*, primeiramente para 0,35 e em seguida para 0,5, seguindo o mesmo critério para determinar a quantidade de neurônios nas camadas ocultas. Nos gráficos de acurácia das Figuras 35e, 35g e 35i, pode-se verificar o efeito do aumento desta taxa: a acurácia para o conjunto de treinamento diminui e, há um incremento, mesmo que pequeno, para a acurácia com o conjunto de validação. Não só isso, mas pode-se verificar também a evolução da função de custo nas Figuras 35f, 35h e 35j. Esta por sua vez, para a configuração com taxa de *dropout* igual a 0,5 estabiliza em torno de um valor com o passar das épocas de treinamento, o que indica que o *overfitting* foi controlado.

A quinta e última configuração foi aquela que, além de apresentar um valor de

função de custo com comportamento que caracteriza menos *overfitting*, foi a que atingiu o melhor valor de acurácia para o conjunto de validação, com o valor de 92,22%, na época de número 130. Apesar da quarta configuração ter apresentado acurácia igual a 92,21% na época 102, é possível notar uma sutil presença de *overfitting* pelo gráfico da função de custo apresentado na Figura 35h.

Em comparação com o trabalho de referência utilizado (NEVEROVA et al., 2014), que apresentou uma acurácia de 98%, o método aqui apresentado obteve desempenho equiparável, uma vez que alterações tiveram que ser feitas na etapa de preparação dos dados para tornar o método viável de se executar em tempo real. Entretanto, esta acurácia representa apenas a classificação de cada instante em MOVIMENTO e REPOUSO. Para a tarefa de localização de gestos e ações, outra métrica é avaliada e será apresentada posteriormente. Na próxima seção serão apresentados os procedimentos para treinamento e seus respectivos resultados, utilizando classificador do tipo *Random Forest*.

4.5.2.2 *Random Forest*

Esse algoritmo consiste na combinação de inúmeras árvores de decisão, treinadas individualmente, com parcelas diferentes do conjunto total dos dados de treinamento. A decisão final é tomada baseada na votação de todas as árvores que compõem o conjunto. Essa estratégia se baseia no princípio de que um conjunto de modelos operando juntos, com dados pouco relacionados, se sai melhor que um único modelo baseado em todos os dados. Contudo, para se obter um bom resultado utilizando *Random Forests* é preciso garantir que os modelos sejam o mais diferentes possíveis. Para isto, este algoritmo se utiliza de alguns artifícios. Um deles é chamado de *bagging*, em que cada árvore seleciona aleatoriamente e com reposição, uma porção dos dados do conjunto de treinamento. Além disso, o processo de escolha da *feature* que será utilizada é diferente das árvores de decisão tradicionais: ao invés de utilizar todas as *features*, apenas um subconjunto destas é escolhido aleatoriamente para ser utilizado na divisão do nó de cada uma das árvores que compõe o conjunto.

Este tipo de modelo possui uma grande quantidade de parâmetros para serem ajustados, como por exemplo o número de estimadores, que corresponde ao número de árvores, a profundidade máxima dessas árvores, o critério para decidir qual característica será utilizada para criar uma derivação na árvore, dentre inúmeros outros. O ajuste destes parâmetros geralmente é um processo complicado, entretanto, como o processo de treinamento geralmente é mais rápido ao se comparar com o treinamento de uma rede neural MLP, é comum utilizar técnicas para se buscar a melhor combinação destes parâmetros. A técnica mais conhecida é a *grid-search*, e foi a utilizada neste trabalho. Esta técnica consiste em uma busca feita a partir da combinação de diversos parâmetros. No Quadro 10 estão apresentados os parâmetros utilizados e seus respectivos valores. Demais parâmetros foram mantidos em seus valores padrão.

Quadro 10 – Parâmetros e seus respectivos valores utilizados no *grid-search* para o algoritmo *Random Forest*. O nome de cada parâmetros, bem como os valores utilizados são iguais à interface oferecida pela biblioteca *Scikit Learn*.

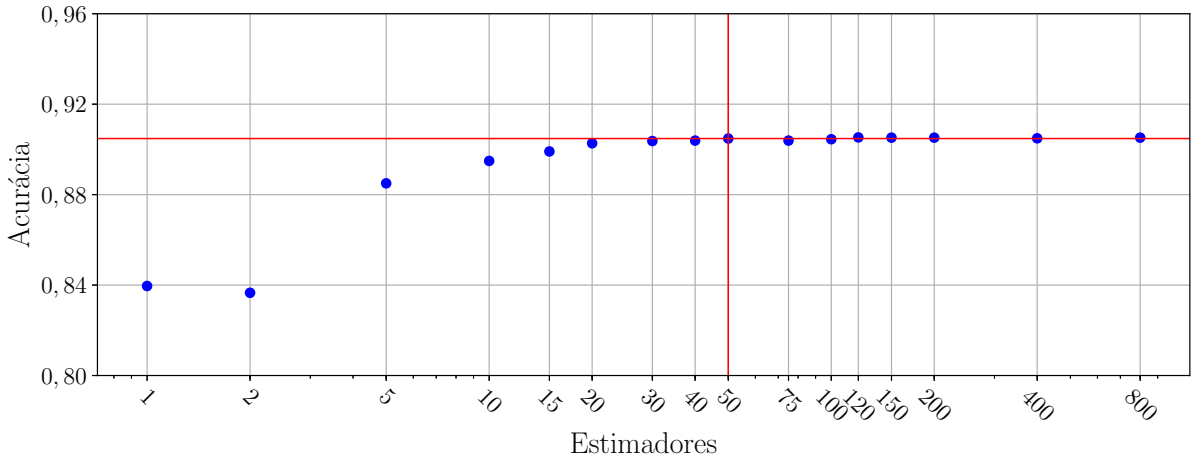
Parâmetro	**Valores Utilizados**	**Observações**
n_estimators	50	-
criterion	"gini"	-
max_depth	[10, 50, 100, 200]	-
max_features	["sqrt", "log2", "None"]	Define a quantidade máxima de características utilizadas para escolher aquela que será utilizada para dividir um nó da árvore. Para "sqrt" e "log2", representam os resultados destas funções aplicadas ao número de características, enquanto que "None" corresponde a utilizar todas elas.
bootstrap	[true, false]	Quando habilitado (*true*), apenas uma parcela dos dados de treinamento é utilizado para construir uma árvore.
class_weight	"balanced"	Pondera os pesos para compensar conjuntos de dados com classes desbalanceadas.

Fonte: Adaptado de
<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.

Para o parâmetro referente ao número de estimadores, antes de se escolher os parâmetros para realizar o *grid-search*, foram executados treinamentos com diversos valores deste parâmetros, mantendo todos os outros em seus valores padrão entregues pela biblioteca utilizada. Ao final de cada treinamento, foi calculado o valor da acurácia para o *dataset* de validação. Esse procedimento foi realizado para determinar qual seria a quantidade de estimadores, uma vez que se sabe que até um certo ponto, aumentar o número de estimadores melhora o desempenho do modelo, entretanto, também faz o seu tempo de inferência ficar maior. No gráfico apresentado na Figura 36, estão os valores de acurácia para cada treino realizado com valores de estimadores diferentes. Pode-se verificar que a partir de 20 estimadores, todos valores de acurácia estão próximos de 90%, e não há incremento expressivo com o aumento de estimadores. Escolheu-se então o valor de 50 estimadores pois foi a melhor acurácia, (90,48%), após ultrapassar o valor de 90%.

O processo de busca utilizado necessita de uma métrica para avaliar qual a melhor configuração. Para isso, foi utilizada *cross-validation*, dividindo os dados de treinamento em 3 partes. Considerando todas as combinações de parâmetros apresentados no Quadro 10, levando em conta o *cross-validation*, foram executados um total de $1 \times 1 \times 4 \times 3 \times 2 \times 1 \times 3 = 72$ treinamentos. A escolha do melhor modelo foi aquele que apresentou a melhor acurácia média entre as três divisões, para uma determinada combinação de parâmetros. O melhor modelo obtido foi o com os parâmetros apresentados no Quadro 11.

Figura 36 – Valores de acurácia obtidos com o conjunto de dados de validação, ao variar o número de estimadores do *Random Forest*, mantendo os outros parâmetros fixos.



Fonte: Produção do próprio autor.

Quadro 11 – Parâmetros do melhor modelo obtido após o *grid-search* realizado para o algoritmo *Random Forest*.

Parâmetro	**Valor**
n_estimators	50
criterion	“gini”
max_depth	100
max_features	“log2”
bootstrap	true
class_weight	“balanced”

Fonte: Produção do próprio autor.

Após treinado, este modelo apresentou uma acurácia no *dataset* de validação de 90,65%, e de 91,26% para o conjunto de teste. O resultado foi abaixo do obtido com MLP, de 92,22%, mas foi bem próximo. Contudo, como anteriormente mencionado, ter uma boa acurácia de classificação em REPOUSO e MOVIMENTO é apenas um bom caminho para obter uma boa localização de gestos e ações. Além disso, o tempo de computação deve ser avaliado para determinar se é viável sua utilização em aplicações que requisitam execução em tempo real. Tais métricas serão avaliadas adiante.

4.5.3 Metodologia experimental e métricas utilizadas

Com os modelos MLP e *Random Forest* treinados e validados, a próxima etapa consiste em utilizar suas saídas como entrada do processo de localização explicado na Seção 4.4. Mesmo que ambos modelos tenham apresentado valores de acurácias bem

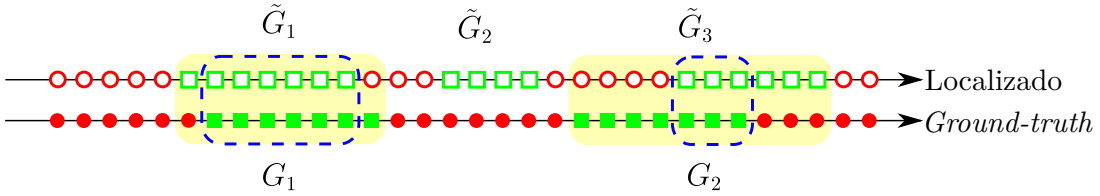
próximos para o *dataset* de teste, é necessário validar seu desempenho na tarefa completa de localização de um gesto ou ação. Além disso, deve-se avaliar seu tempo de computação para determinar a viabilidade de executá-lo em aplicações de tempo real e com requisitos de baixo tempo de resposta.

Dessa maneira, para avaliar a metodologia aqui proposta, será levado em consideração o tempo para avaliação de um instante, enquanto a acurácia da localização do gesto será avaliada utilizando-se a mesma metodologia do desafio no qual o *dataset Montalbano V2* foi proposto, apenas com uma pequena alteração. Este desafio, como já mencionado, consistia em realizar a tarefa de localizar e classificar os gestos das sequências disponíveis. Portanto, era levado em consideração não só os instantes de início e fim do gesto, mas também em qual classe este foi considerado, e então calculava-se um índice. No final, o valor médio de todos os índices era calculado e utilizado para determinar o melhor método apresentado para solucionar o desafio.

Este índice utilizado é o índice de *Jaccard*. Este consiste em calcular a razão entre a interseção e a união, do intervalo de execução de um gesto localizado e do *ground-truth* fornecido. Na Figura 37 está representada uma linha do tempo com o *ground-truth*, na qual os marcadores representados por círculos vermelhos representam instantes de repouso, e aqueles representados por quadrados verdes, momentos de execução do gesto. Nesta figura, também está ilustrada a sequência hipotética gerada pelo localizador de gestos. Como pode-se verificar, três gestos foram localizados para esta sequência. Estão também destacadas as regiões de união e interseção com o *ground-truth*. Para o gesto localizado $\tilde{G}_1$ por exemplo, o índice de *Jaccard* é calculado por

$$J_1 = \frac{\tilde{G}_1 \cap G_1}{\tilde{G}_1 \cup G_1} = \frac{6}{8} = 0{,}75. \tag{4.20}$$

Figura 37 – Exemplo que ilustra o cálculo de índice de *Jaccard*. Os instantes rotulados por círculos vermelhos correspondem a instantes de repouso, enquanto aqueles com quadrados verdes, momentos de execução. Rótulos preenchidos internamente correspondem ao *ground-truth*. As áreas de interseção e união para cada gesto localizado estão identificadas.



Fonte: Produção do próprio autor.

Para o gesto localizado $\tilde{G}_2$, o índice de *Jaccard* é $J_2 = 0$. Importante dizer que este também é contabilizado ao calcular a média geral. Já o gesto $\tilde{G}_3$ possui índice diferente de

zero, uma vez que possui interseção com a anotação G_2, e seu índice é igual a

$$J_3 = \frac{\tilde{G}_3 \cap G_2}{\tilde{G}_3 \cup G_2} = \frac{3}{10} = 0{,}3. \tag{4.21}$$

Assim, o índice de *Jaccard* médio para esta sequência é igual a $\frac{0{,}75+0+0{,}3}{3} = 0{,}35$.

Além disso, esta métrica será avaliada para diferentes configurações do localizador, isto é, com diferentes valores de δ e N_{min}, para ambos os melhores modelos obtidos de MLP e *Random Forest*. Uma vez que estes parâmetros são apenas limiares que não influenciam no tempo de processamento, este só será avaliado para a melhor configuração encontrada. No Quadro 12 estão os valores utilizados para os dois parâmetros do localizador. Com tais valores, foram geradas todas combinações possíveis, e estas foram utilizadas nos testes. Além disso, a escolha dos valores de N_{min} foi baseada na taxa de amostragem da base de dados, que é igual a 20 *fps*, como apresentado no Quadro 7. O índice de suavização adotado para os experimentos foi de $\alpha = 0{,}8$.

Quadro 12 – Parâmetros do localizador de gestos utilizados nos testes realizados com os modelos MLP e *Random Forest*, para os conjuntos de dados de treino, validação e teste.

$\boldsymbol{\delta}$	[0,0, 0,05, 0,10, 0,15]
$\boldsymbol{N_{min}}$	[20, 25, 30, 35, 40]

Fonte: Produção do próprio autor.

4.5.4 Resultados

Nesta seção serão apresentados os resultados do método de localização de gestos aqui proposto, avaliando sua eficiência através do índice de *Jaccard*, apresentado na Seção 4.5.3, além do tempo de computação para ambos os modelos, tanto MLP quando *Random Forest*. Primeiramente, foi avaliado o modelo MLP para os parâmetros apresentados no Quadro 12. Na Tabela 3 estão apresentados os valores do índice de *Jaccard* obtidos para o conjuto de dados de teste.

Como pode ser observado, nas colunas da Tabela 3, estão colocados os resultados para valores fixos de δ, variando apenas os parâmetros N_{min}. Dessa maneira, é possível observar a melhora no desempenho ao se aumentar o tamanho da janela mínima de tempo, na qual considera-se possível que um gesto tenha sido realizado. Este valor está associado à taxa de amostragem. Para este *dataset*, esta taxa é de 20 *fps*, o que para os valores de N_{min} testados representa gestos com tempos de execuções entre 1 e 2 segundos. Este resultado mostra que utilizar esta política para excluir detecções com durações pequenas ajuda a reduzir a quantidade de falsas detecções.

Tabela 3 – Valores médios de índice de *Jaccard* obtidos com o modelo MLP, para o conjuntos de dados de teste, utilizando os valores de N_{min} e δ apresentados no Quadro 12.

N_{min} \ δ	0,0	**0,05**	0,10	0,15
20	0,8457	0,8545	0,8564	0,8561
25	0,8565	0,8625	0,8619	0,8597
30	0,8734	**0,8769**	0,8760	0,8715
35	0,8683	0,8703	0,8652	0,8575
40	0,8505	0,8495	0,8408	0,8305

Fonte: Produção do próprio autor.

De outra maneira, para uma mesma linha da Tabela 3, o valor de N_{min} é constante, sendo então possível verificar a influência do valor do fator de tolerância δ. Para este parâmetro, nota-se que na maioria dos cenários, o desempenho para $\delta = 0{,}0$, que é equivalente à não haver uma zona de estado indefinido, é pior do que para o próximo valor testado, $\delta = 0{,}05$. Isto mostra que a utilização desta faixa de tolerância melhora o desempenho do localizador de gestos, uma vez que esta impede que flutuações do valor para dentro da faixa de tolerância não faz com que se indique um falso fim ou início do gesto, evitando assim localizações errôneas.

Dentre todos os testes executados com o modelo MLP, o tempo de execução para cada instante avaliado foi de aproximadamente 1,5 ms. Este tempo foi obtido usando-se um computador com processador Intel Core i5-7200U @ 2,50GHz. Vale ressaltar que este tempo inclui desde a preparação dos dados de entrada, a inferência do modelo, e a classificação final do estado atual da execução do gesto. Tal resultado mostra que este método seria capaz de ser utilizado em aplicações de tempo real com altas taxas de amostragem e, consequentemente, baixo tempo de resposta. O melhor resultado obtido está destacado em negrito na Tabela 3, com um índice de *Jaccard* médio percentual de 0,8769%, com $\delta = 0{,}05$ e $N_{min} = 30$, o que foi muito próximo ao índice obitido na configuração com mesmo valor de N_{min}, mas com $\delta = 0{,}10$.

Também foram realizados os mesmos testes utilizando-se o melhor modelo *Random Forest*. Na Tabela 4 estão apresentados os índices de *Jaccard*. Com este modelo, obteve-se um índice igual a 0,8682, para $\delta = 0{,}15$ e $N_{min} = 30$. Mesmo que esse resultado tenha sido bem próximo do obtido com um modelo MLP, o tempo médio de execução para cada instante, nas mesmas condições, isto é, mesmo recurso computacional, foi de aproximadamente 107 ms. Com base neste resultado, utilizar um modelo baseado em MLP corresponde a uma melhor opção, levando em conta a precisão da localização do gesto obtida, e também o tempo de resposta necessário, o que possibilita a utilização em aplicações de tempo real.

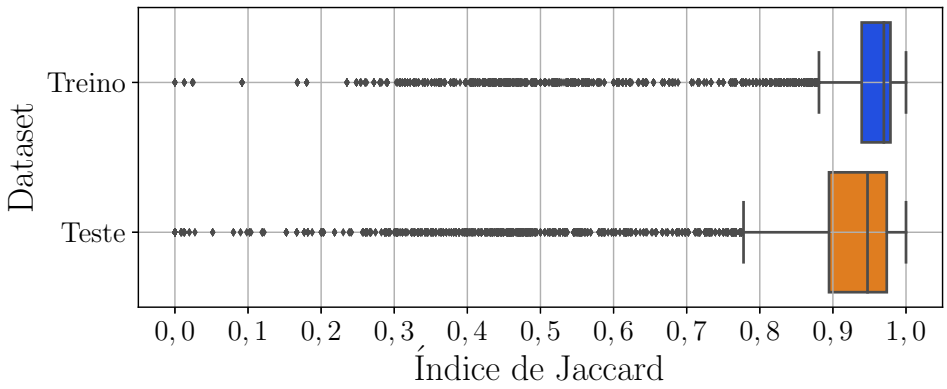
Tabela 4 – Valores médios de índice de *Jaccard* obtidos com o modelo *Random Forest*, para o conjuntos de dados de teste, utilizando os valores de N_{min} e δ apresentados no Quadro 12.

N_{min} \ δ	0,0	0,05	0,10	**0,15**
20	0,8327	0,8378	0,8522	0,8576
25	0,8491	0,8574	0,8656	0,8675
30	0,8604	0,8640	0,8675	**0,8682**
35	0,8547	0,8550	0,8561	0,8500
40	0,8359	0,8349	0,8243	0,8128

Fonte: Produção do próprio autor.

Aplicando-se a melhor configuração obtida com o modelo MLP, foi construído um *Boxplot* a partir das sequências dos conjuntos de treino e teste, com todos os índices de *Jaccard* obtidos. Este está apresentado na Figura 38, na qual é possível ter uma visão geral de como o localizador de gestos proposto se comporta para todas as sequências do conjunto avaliado. Para o conjunto de dados de treino, o valor mediano ficou em 0,9697, e 50% dos índices ficaram entre 0,9394 e 0,9787, enquanto que para o conjunto de testes, a mediana foi de 0,9474, e 50% dos valores obtidos estão entre 0,8947 e 0,9737. Isso mostra que, do ponto de vista da eficiência do localizador baseada no índice de *Jaccard*, este apresenta pouca variação.

Figura 38 – *Boxplots* de todos os índices de *Jaccard* obtidos com a melhor configuração do localizador de gestos, para os conjuntos de treino e teste.

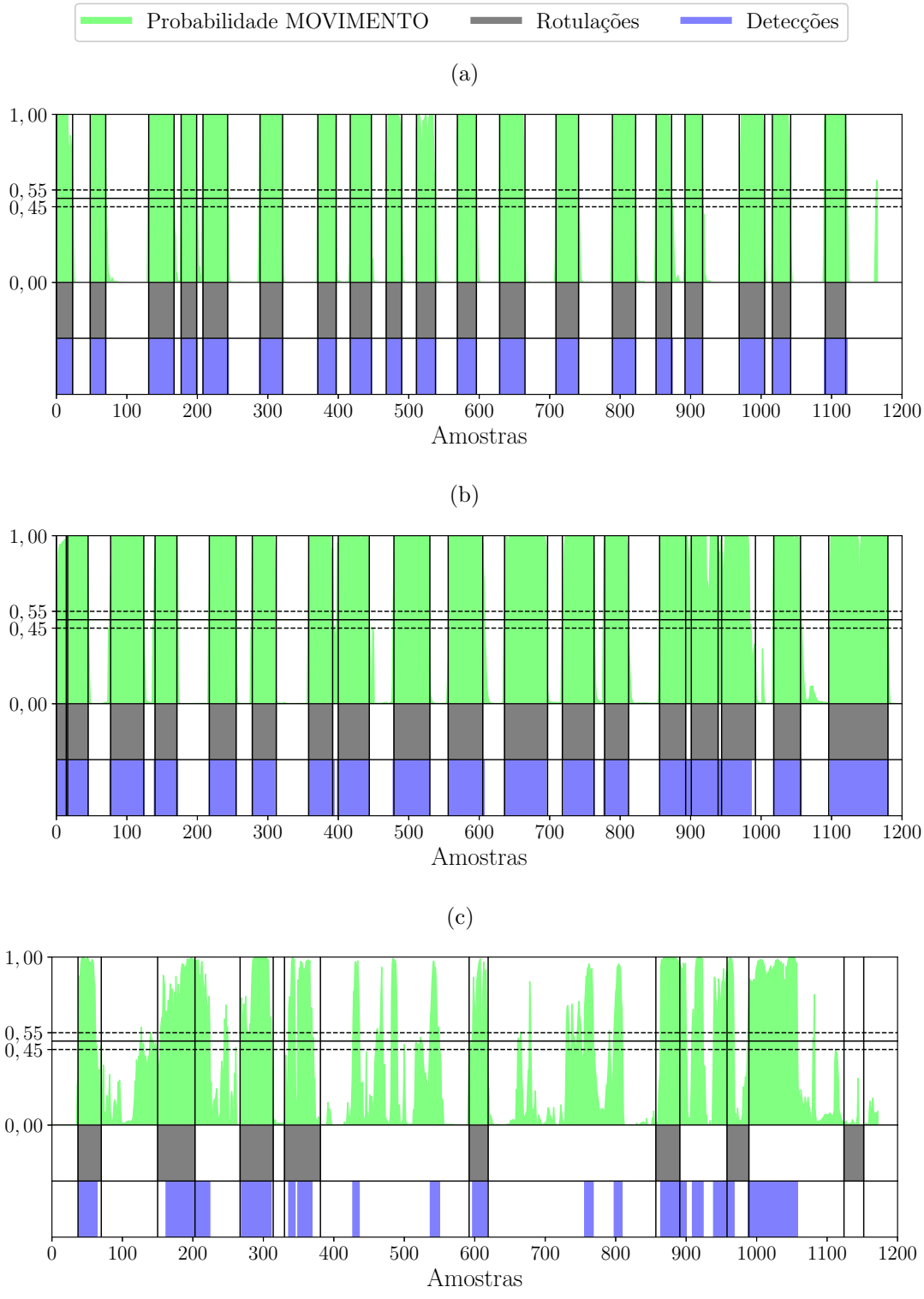


Fonte: Produção do próprio autor.

Uma maneira mais visual de se verificar a eficiência do localizador, é vendo para uma dada sequência, o valor da probabilidade de saída do modelo, e como todo o sistema identificou o início e o fim de uma execução. Na Figura 39 estão representadas três sequências: aquelas que apresentaram respectivamente o melhor, o mediano, e o pior índice de *Jaccard* médio. Nos gráficos que ilustram as três sequências estão indicados os limites

superior e inferior da zona correspondente a um estado indefinido, bem como as rotulações de cada execução presente na sequência, e o que o sistema identificou como o início e fim de uma execução.

Figura 39 – Ilustração com três sequências do conjunto de dados de teste, sendo (a) aquela que apresentou o melhor índice de *Jaccard* médio, (b) o índice mediano, e (c) o pior.



Para a melhor execução, ilustrada na Figura 39a, nota-se que todas execuções foram identificadas corretamente, e, ao final da sequência, entre as amostras de número 1100 e 1200, a probabilidade para a classe MOVIMENTO ultrapassou o limiar superior, o que indicaria um início de gesto, porém, como ela voltou para baixo do limiar inferior antes de completar o número N_{min} de amostras, foi então descartada. A execução mediana, apresentada na Figura 39b, também conseguiu localizar praticamente todas execuções. Um ponto a se observar nesta sequência, é entre as amostras 800 e 1000. De acordo com a rotulação representada na cor cinza, houve três execuções. Contudo, como não houve diminuição no valor da probabilidade entregue pelo modelo, o sistema não foi capaz de informar o fim da execução corretamente. Nesta situação, um motivo pode estar relacionado com esta falha: execuções muito próximas uma das outras com pouco tempo de repouso. A pior execução, Figura 39c, conseguiu localizar bem apenas a primeira execução da sequência, e aquelas próximas das amostras de número 300 e 600. Nesta sequência houve também várias falsas detecções.

O método aqui proposto teve parte dele baseado em (NEVEROVA et al., 2014), que na época do desafio proposto junto com o *dataset*, obteve o melhor resultado com um índice de *Jaccard* de 0,8500. Contudo, ele fazia, não só a localização, mas a classificação do gesto. Dessa maneira, o índice era avaliado também baseado na classe do gesto, isto é, se um gesto foi localizado, mas classificado errado, o índice é considerado igual a zero. Além disso, eram utilizado não apenas os dados de localização das juntas de esqueleto, mas também as imagens RGB, mapa de profundidade, e regiões de interesse em torno das mãos, uma vez que alguns gestos possuem movimentos manuais marcantes. (NEVEROVA et al., 2014) reportou também, que foi atingida uma taxa de 24 fps, mas para uma configuração simplificada, que utilizava apenas uma escala e que, apresentou desempenho inferior que a melhor configuração alcançada.

Ao comparar (NEVEROVA et al., 2014) com o trabalho aqui proposto, tendo em vista um sistema como um todo, (NEVEROVA et al., 2014) consegue trabalhar com uma taxa que atenderia aplicações de tempo real, contudo, mesmo sendo um modelo capaz de também classificar, necessita de GPU para sua execução. Já o sistema aqui proposto, é capaz de localizar a execução de gestos com um tempo de aproximadamente 1,5 ms sem a necessidade de GPU, mas em contrapartida, necessita de uma etapa posterior de classificação. De todo modo, com o vídeo já segmentado, a etapa de classificação se torna mais simples.

Assim como (NEVEROVA et al., 2014), (WU et al., 2016) também realiza localização e classificação dos gestos no mesmo *dataset*, utilizando as mesmas informações como entrada para seu modelo: pose humana, imagens RGB e mapa de profundidade, aplicando redes neurais convolucionais 3D. (WU et al., 2016) obteve um índice de *Jaccard* de 0,809, e se utilizar apenas os dados de pose, obtém um índice de 0,779. Além disso, é capaz de

ser executado à 25 *fps* em GPU, com um limitante em sua taxa devido a uma etapa de pré-processamento. A comparação deste trabalho é semelhante a feita com (NEVEROVA et al., 2014): mesmo com tempo de execução sendo capaz de executar em tempo real, necessita de GPU e, precisa de todas informações vindas de um sensor como o *Kinect* para atingir este resultado.

Em (PIGOU et al., 2018), foram testadas várias arquitetura envolvendo redes neurais convolucionais e recorrentes, variando-se os dados de entrada disponíveis no *dataset Montalbano v2*. Por exemplo, foram realizados testes com e sem a informação de profundidade, e adicionou-se informação de fluxo óptico. Seu melhor resultado obteve um índice de *Jaccard* de 0,906. Sobre o tempo de inferência, nada é mencionado neste trabalho.

De maneira diferente aos trabalhos mencionados, (JOSHI et al., 2017) também utiliza o *dataset Montalbano*, mas propõe duas arquiteturas para localizar a classificar os gestos: a primeira delas, utiliza um modelo do tipo *Random Forest* para simultaneamente localizar e classificar os gestos, enquanto que a segunda arquitetura, possui uma etapa inicial com um classificador binário utilizando *Random Forest*, para primeiro localizar o gesto, e em seguida classificar. Ambas arquiteturas utilizam as imagens RBGs e as poses fornecidas pelo *dataset*. Para a primeira arquitetura, (JOSHI et al., 2017) obteve um índice de *Jaccard* de 0,68, enquanto que na segunda arquitetura foi de 0,72. Como a segunda arquitetura proposta consistia em uma etapa separada para a classificação, (JOSHI et al., 2017) atingiu um *score* de 88,91% na classificação dos gestos, considerando-os já segmentados. Nada foi mencionado sobre o tempo de execução das arquiteturas propostas.

Mesmo que o método aqui proposto realize apenas a localização de gestos, este apresentou uma boa precisão comparado com os trabalhos que realizam localização e classificação. Dessa maneira, a localização aqui gerada pode então ser entregue à uma etapa posterior, parar por exemplo segmentar vídeos para serem classificados. O método também se mostrou capaz de entregar a localização com baixa latência, impactando pouco no tempo de resposta total do sistema, em um cenário que haja uma etapa posterior de classificação. Esta etapa pode utilizar não só os dados de esqueleto, mas também imagens, se o sistema as possuir, podendo torná-la mais precisa.

No próximo capítulo serão abordados aspectos gerais do sistema aqui proposto, e como ele pode ser integrado em uma aplicação com requisito de tempo real. São discutidos os pontos de melhoria e também características arquiteturais que podem ser adotadas para se atingir um sistema capaz de localizar e classificar gestos de múltiplas pessoas em tempo real.

5 Conclusões, Visão Geral do Sistema e Trabalhos Futuros

O trabalho aqui apresentado teve por finalidade mostrar a viabilidade de um sistema que fosse capaz de, a partir de imagens de um sistema multicâmeras, produzir a informação referente à localização da execução de gestos ou ações em tempo real. Para isso, no Capítulo 3, foi apresentada uma metodologia para estimar a localização de juntas tridimensionais utilizando apenas imagens de múltiplas câmeras. O sistema proposto possui uma etapa inicial que utiliza o detector bidimensional de esqueletos proposto em (WEI et al., 2016). O método foi avaliado no *dataset CMU Panoptic*, calculando-se o erro de reconstrução para diferentes configurações do detector de esqueletos, além do tempo gasto nas etapas do método, sempre levando em consideração o *hardware* utilizado.

Os erros de localização das juntas de esqueletos ficaram entre 40 e 80 mm para a configuração C_1 do Quadro 5. Para os tempos de execução, a etapa de detecção foi avaliada para diferentes configurações do detector e quatro modelos de GPU, apresentando tempo mínimo de 30,16 ms e máximo de 68,68 ms. O tempo gasto no processo de busca de correspondências e reconstrução tridimensional foi também avaliado, e seu comportamento apresentado no gráfico da Figura 20, mostrando que, no pior caso avaliado, que é aquele que havia o maior número de detecções em todos as câmeras do conjunto, gastou-se menos de 16 ms.

Em seguida, no Capítulo 4, foi apresentada uma metodologia para, a partir de uma sequência de poses humanas, determinar o início e final da execução de um gesto ou ação. O método foi avaliado no *dataset Montalbano v2*, e adotando-se o índice de *Jaccard* para quantificar sua eficiência. O resultado apresentou um índice igual a 0,8769 no conjunto de dados de teste. Foram testados dois algoritmos de *Machine Learning* diferentes, levando em consideração, além da acurácia, o tempo de execução. Para o melhor modelo obtido, obteve-se um tempo médio de 1,5 ms para cada instante avaliado.

Neste capítulo, será mostrada uma visão geral do sistema composto pelos dois métodos desenvolvidos nos Capítulos 3 e 4, e como uma aplicação pode utilizar esses módulos. Além disso, serão discutidas pontos de melhoria no sistema, apresentando as limitações e uma possível arquitetura para contorná-las. Na Figura 40 está apresentada a visão geral do sistema com os módulo desenvolvidos. Toda a comunicação entre os componentes é feita utilizando um barramento de mensagens. Módulos com reticências ao lado de seus ícones podem ser replicados sob demanda. O barramento de mensagens foi representado por barras horizontais e, mesmo que hajam várias barras, é um barramento único. De cima para baixo, temos os seguintes módulos:

1. Câmeras e um eventual módulo para converter seus *frames* em um formato padrão que os outros módulos irão consumir;

2. Detecção de esqueletos nas imagens. Estes possuem um requisito de GPU em sua representação, mas esta pode não ser uma dependência.

3. Estimativa tridimensional das juntas de esqueletos, referente ao Capítulo 3.

4. Localização dos gestos ou ações, referente ao Capítulo 4.

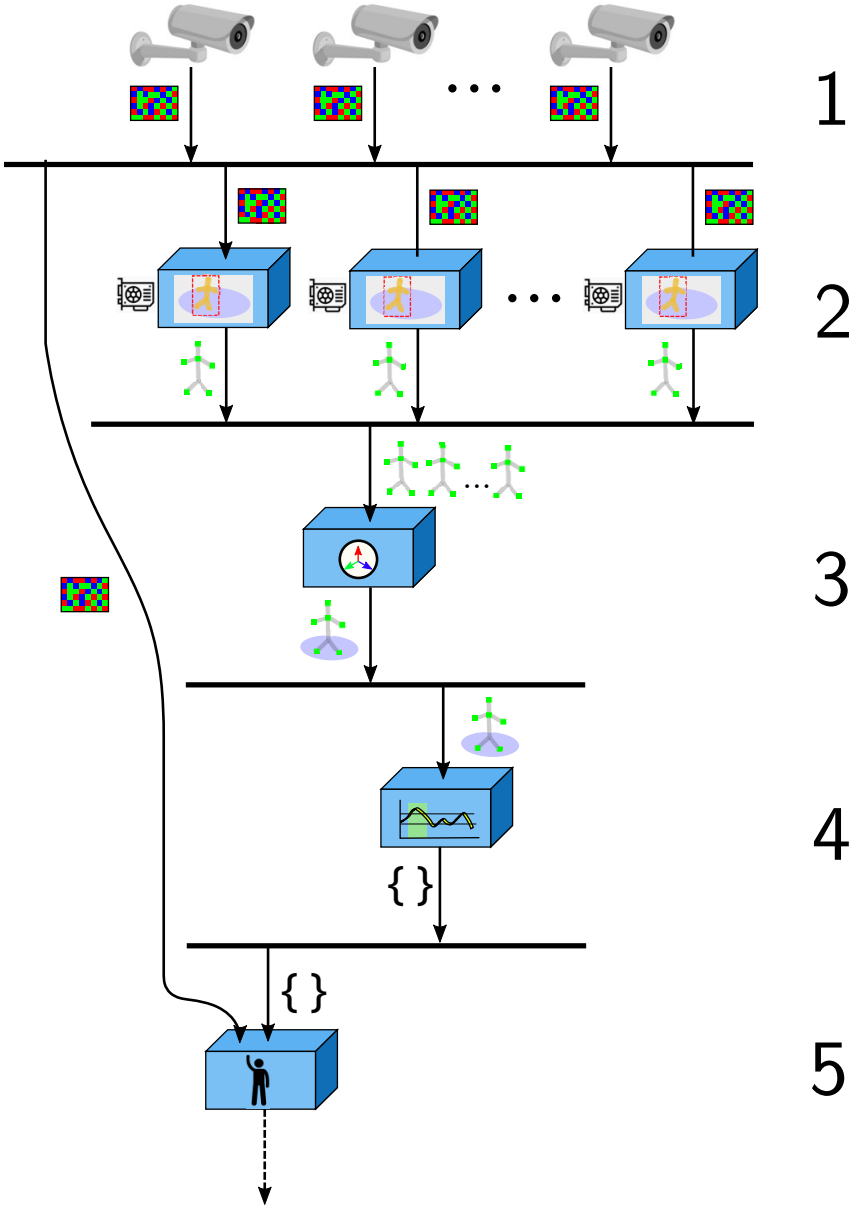
5. Aplicação que consome as mensagens referentes à localização dos gestos, além das imagens da câmeras, e as classifica.

Como já foi comentado, os módulos de detecção de esqueletos (2) podem ser escalados de acordo com a configuração do sistema, isto é, número de câmeras, taxa de amostragem destas e parâmetros do detector. Como a comunicação do sistema é feita por um barramento de mensagens, ocorre um balanceamento de carga natural nas instâncias deste módulo. Este por sua vez, produz mensagens que contém anotações na imagem, que correspondem a uma estrutura de dados com uma lista de posições na imagem, e um identificador correspondente à qual junta aquela posição representa. Esta mensagem pode conter múltiplas anotações, referentes à vários indivíduos presentes na imagem.

O módulo de número (3), está conectado ao barramento de mensagens consumindo as anotações de esqueletos. Ele as acumula na mesma taxa de amostragem que as câmeras estão operando, e ao fim de um período de amostragem, estima a localização tridimensional das juntas. Também são geradas anotações, mas estas possuem três dimensões para cada junta. Múltiplos anotações podem ser geradas. As mensagens são enviadas ao barramento, e são consumidas pelo módulo (4). Neste módulo, algumas limitações começam a aparecer. Como cada mensagem que chega pode chegar com vários esqueletos que foram reconstruídos, é necessário diferenciar qual deles corresponde ao instante anterior, uma vez que este módulo utiliza informação temporal para realizar a tarefa de localizar a execução de gestos. Dessa maneira, o modelo atual só é capaz de realizar a tarefa por completo para um único indivíduo, mesmo que o módulo de estimativa de localização das juntas seja capaz de lidar com múltiplos indivíduos.

Assim, considerando que há apenas um indivíduo na cena, o módulo (4) poderá tratar as anotações recebidas como sendo do mesmo indivíduo. Este módulo produz mensagens com a informação referente aos estados na Figura 31. O módulo (5), corresponde a uma aplicação hipotética que, consumiria em conjunto as imagens das câmeras, e as mensagens produzidas pelo módulo (4), acumularia as imagens de um intervalo de execução, isto é, a partir do momento que recebesse uma mensagem com o estado INÍCIO, e ao receber uma mensagem com FIM, utilizaria esta sequência de imagens para classificar o gesto

Figura 40 – Visão geral do sistema com os componentes desenvolvidos nesta dissertação, além de um exemplo de aplicação integrada à arquitetura.

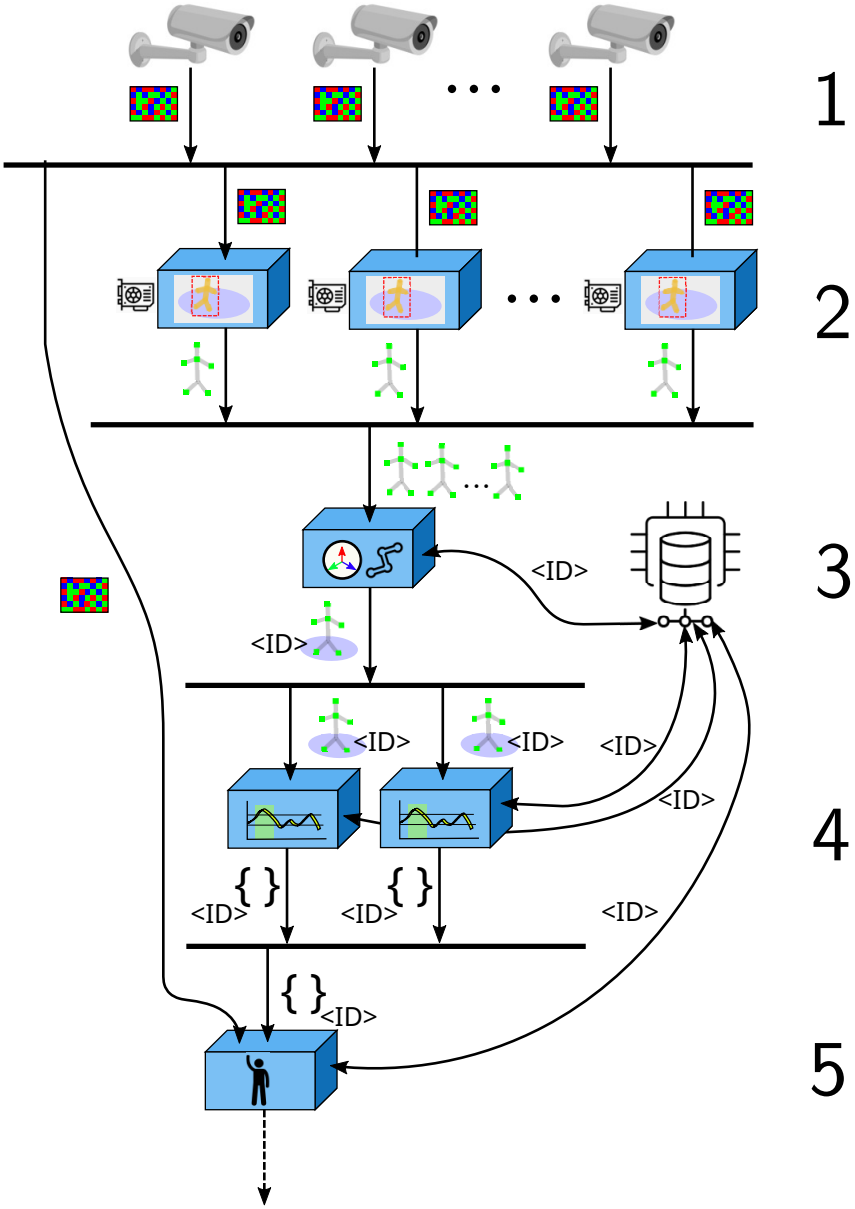


Fonte: Produção do próprio autor.

realizado. Ela poderia também consumir as localizações tridimensionais dos esqueletos, ou ambas as informações.

Além das limitações já citadas, o modelo atual do módulo (3) e (4) não são possíveis de haver réplicas, uma vez que estes possuem estado referente à execução de um único indivíduo. Na Figura 41, está apresentada uma possível arquitetura que resolveria algumas das limitações. Nesta Figura, foi adicionado um elemento de persistência em memória desacoplado dos demais módulos. Este deve possuir baixa latência para operações de escrita/leitura, a fim de não impactar o desempenho do sistema.

Figura 41 – Visão geral do sistema com os componentes desenvolvidos nesta dissertação, incluindo um elemento de persistência para resolver limitações do sistema apresentado na Figura 40.



Fonte: Produção do próprio autor.

Além da adição do elemento de armazenamento, algumas alterações teriam que ser feitas nos módulos (3), (4) e (5). O módulo (3) já é capaz de lhe dar com uma cena com múltiplos indivíduos, entretanto, estes não são correlacionados temporalmente. Isto é, dado uma situação hipotética em que há dois indivíduos na cena, em um instante $t = t_0$, estes seriam identificados e, no instante seguinte $t = t_1$, caso os dois sejam identificados, eles devem ser associados às detecções realizadas no instante t_0. Isso se faz necessário pois as etapas seguintes referentes aos módulos (4) e (5), necessitam do estado anterior referente ao mesmo indivíduo.

Para resolver este problema, duas abordagens podem ser consideradas e, adotadas concomitantemente. A primeira é adicionar um sistema de rastreamento ao módulo (3). O estado deste seria armazenado no elemento de persistências. Dessa maneira, o módulo (3) passaria a não ter mais estado, tornando o sistema mais robusto e menos suscetível a uma falha. Este estado seria associado à um identificado único, na Figura 41 indicado por $< ID >$. Dessa forma, cada pose tridimensional gerada pelo módulo (3) teria um identificador associado à ela. A segunda abordagem, seria adicionar algum sistema de biometria, podendo por exemplo utilizar informação visual, para aprimorar o rastreamento. Uma abordagem não exclui a outra e, podem trabalhar juntas para melhorar o desempenho do sistema.

O módulo seguinte, de número (4), seria capaz agora de ser replicado, sendo capaz de suportar um volume maior de detecções sendo geradas, no caso de um sistema maior com múltiplas câmeras e vários indivíduos na cena. Este, ao receber uma nova mensagem com poses humanas, consultaria no elemento de persistência a informação referente ao estado anterior associada ao identificador daquela pose. O resultado gerado, que corresponde ao estado em que a execução se encontra, também seria associado ao identificador único, e propagado para o próximo módulo, responsável por classificar o gesto ou ação localizado. O quinto módulo por sua vez, também guardaria no elemento de persistência o estado recebido do módulo (4), para que também seja um robusto à falha, e consiga se recuperar sem perder informações. Dessa maneira, o sistema seria capaz de lidar com execuções de múltiplas pessoas.

O sistema aqui proposto, uma vez que realiza sua comunicação através de um barramento de mensagens, pode ser integrado com outras aplicações que processem tais mensagens para relacionar a ocorrência de eventos, pensando em um cenário que haja outras fontes geradoras de mensagens, relacionadas a outros comportamentos ocorridos no ambiente. Além disso, a localização tridimensional pode ser utilizada para outros propósitos, como análise de marcha ou interação com sistemas de realidade aumentada. Dessa maneira, conclui-se que é possível conceber um sistema capaz de, a partir de somente imagens de múltiplas câmeras, obter a localização tridimensional das juntas de esqueletos de múltiplos indivíduos, e utilizá-la para localizar a execução de gestos ou ações, em um contexto no qual deseja-se operar em tempo real, e com baixo tempo de resposta.

Referências

AKHTER, I.; BLACK, M. J. Pose-Conditioned Joint Angle Limits for 3D Human Pose Reconstruction. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015)*. [S.l.: s.n.], 2015. Citado na página 35.

ASADI-AGHBOLAGHI, M. et al. A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences. In: *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. [S.l.]: IEEE, 2017. p. 476–483. ISBN 978-1-5090-4023-0. Citado na página 27.

ASADI-AGHBOLAGHI, M. et al. A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences. In: *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*. [S.l.: s.n.], 2017. p. 476–483. Citado na página 38.

BACCOUCHE, M. et al. Sequential deep learning for human action recognition. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.: s.n.], 2011. v. 7065 LNCS, p. 29–39. ISBN 9783642254451. ISSN 03029743. Citado 3 vezes nas páginas 9, 41 e 42.

BALAZIA, M.; SOJKA, P. You are how you walk: Uncooperative MoCap gait identification for video surveillance with incomplete and noisy data. In: *IEEE International Joint Conference on Biometrics, IJCB 2017*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. v. 2018-Janua, p. 208–215. ISBN 9781538611241. Citado na página 25.

BELAGIANNIS, V. et al. 3D Pictorial Structures for Multiple Human Pose Estimation. In: *IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. Citado 2 vezes nas páginas 34 e 37.

BELAGIANNIS, V. et al. *Multiple Human Pose Estimation with Temporally Consistent 3D Pictorial Structures*. [S.l.], 2014. Citado na página 67.

BREAZEAL, C.; ARYANANDA, L. Recognition of Affective Communicative Intent in Robot-Directed Speech. *Autonomous Robots*, Kluwer Academic Publishers, v. 12, n. 1, p. 83–104, 2002. ISSN 09295593. Citado na página 20.

BREGLER, C. Motion capture technology for entertainment [In the spotlight]. *IEEE Signal Processing Magazine*, Institute of Electrical and Electronics Engineers Inc., v. 24, n. 6, 2007. ISSN 10535888. Citado na página 25.

BYUN, J. et al. An intelligent self-adjusting sensor for smart home services based on ZigBee communications. *IEEE Transactions on Consumer Electronics*, v. 58, n. 3, p. 794–802, 8 2012. ISSN 0098-3063. Citado na página 19.

CAO, Z. et al. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. 11 2016. Citado 2 vezes nas páginas 22 e 32.

CARRARO, M. et al. Real-time marker-less multi-person 3D pose estimation in RGB-Depth camera networks. In: *International Conference on Intelligent Autonomous Systems*. [S.l.: s.n.], 2018. Citado 2 vezes nas páginas 36 e 73.

CASILLAS-PEREZ, D. et al. Full Body Gesture Recognition for Human-Machine Interaction in Intelligent Spaces. In: . [S.l.]: Springer, Cham, 2016. p. 664–676. Citado na página 21.

CHENARLOGH, V. A.; RAZZAZI, F. Multi-stream 3D CNN structure for human action recognition trained by limited data. *IET Computer Vision*, Institution of Engineering and Technology, v. 13, n. 3, p. 338–344, 4 2019. ISSN 17519640. Citado na página 40.

CHOUTAS, V. et al. *PoTion: Pose MoTion Representation for Action Recognition*. [S.l.], 2018. Citado 2 vezes nas páginas 9 e 46.

CLOETE, T.; SCHEFFER, C. Repeatability of an off-the-shelf, full body inertial motion capture system during clinical gait analysis. In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10*. [S.l.: s.n.], 2010. p. 5125–5128. ISBN 9781424441235. ISSN 1557-170X. Citado na página 25.

COLYER, S. L. et al. A Review of the Evolution of Vision-Based Motion Analysis and the Integration of Advanced Computer Vision Methods Towards Developing a Markerless System. *Sports Medicine - Open*, Springer Science and Business Media LLC, v. 4, n. 1, 12 2018. ISSN 2199-1170. Citado 2 vezes nas páginas 25 e 26.

COOK, D. J. et al. CASAS: A Smart Home in a Box. *Computer*, v. 46, n. 7, p. 62–69, 7 2013. ISSN 0018-9162. Citado na página 19.

COWIE, R. et al. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, v. 18, n. 1, p. 32–80, 2001. ISSN 10535888. Citado na página 20.

DANG, Q. et al. Deep learning based 2D human pose estimation: A survey. *Tsinghua Science and Technology*, Tsinghua University Press, v. 24, n. 6, p. 663–676, 12 2019. ISSN 18787606. Citado 2 vezes nas páginas 31 e 32.

DONG, J. et al. Fast and Robust Multi-Person 3D Pose Estimation from Multiple Views. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019. Citado 4 vezes nas páginas 37, 68, 69 e 73.

ELHAYEK, A. et al. Fully automatic multi-person human motion capture for VR applications. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.]: Springer Verlag, 2018. v. 11162 LNCS, p. 28–47. ISBN 9783030017897. ISSN 16113349. Citado na página 25.

EREN, G. et al. *Multi-view pose estimation with mixtures of parts and adaptive viewpoint selection*. [S.l.]: Institution of Engineering and Technology, 2017. v. 12. 403–411 p. ISSN 1751-9640. Citado na página 36.

ERSHADI-NASAB, S. et al. Multiple human 3D pose estimation from multiview images. *Multimedia Tools and Applications*, Springer US, v. 77, n. 12, p. 15573–15601, 6 2018. ISSN 1380-7501. Citado na página 37.

ESCALERA, S.; ATHITSOS, V.; GUYON, I. Challenges in multimodal gesture recognition. *Journal of Machine Learning Research*, v. 17, n. 72, p. 1–54, 2016. Citado 2 vezes nas páginas 47 e 76.

ESCALERA, S.; ATHITSOS, V.; GUYON, I. Challenges in Multi-modal Gesture Recognition. In: . [S.l.]: Springer, Cham, 2017. p. 1–60. Citado na página 21.

ESCALERA, S. et al. ChaLearn Looking at People Challenge 2014: Dataset and Results. In: . [S.l.]: Springer, Cham, 2015. p. 459–473. Citado 3 vezes nas páginas 48, 95 e 96.

FEICHTENHOFER, C.; PINZ, A.; ZISSERMAN, A. Convolutional Two-Stream Network Fusion for Video Action Recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. Citado na página 40.

FIORINI, P.; ALI, K.; SERAJI, H. Health care robotics: a progress report. In: *Proceedings of International Conference on Robotics and Automation*. [S.l.]: IEEE. v. 2, p. 1271–1276. ISBN 0-7803-3612-7. Citado na página 19.

GOLDIN-MEADOW, S. et al. The natural order of events: how speakers of different languages represent events nonverbally. *Proceedings of the National Academy of Sciences of the United States of America*, National Academy of Sciences, v. 105, n. 27, p. 9163–8, 7 2008. ISSN 1091-6490. Citado na página 20.

GONG, W. et al. Human Pose Estimation from Monocular Images: A Comprehensive Survey. *Sensors*, 2016. Citado na página 28.

GRANATA, C. et al. Voice and graphical -based interfaces for interaction with a robot dedicated to elderly and people with cognitive disorders. In: *19th International Symposium in Robot and Human Interactive Communication*. [S.l.]: IEEE, 2010. p. 785–790. ISBN 978-1-4244-7991-7. Citado na página 19.

GREER, A. D.; NEWHOOK, P. M.; SUTHERLAND, G. R. Human–Machine Interface for Robotic Surgery and Stereotaxy. *IEEE/ASME Transactions on Mechatronics*, v. 13, n. 3, p. 355–361, 6 2008. ISSN 1083-4435. Citado na página 20.

HASHIMOTO, H. Intelligent space: Interaction and intelligence. *Artificial Life and Robotics*, Springer-Verlag, v. 7, n. 3, p. 79–85, 9 2003. ISSN 1433-5298. Citado na página 19.

HELAL, S. et al. The Gator Tech Smart House: a programmable pervasive space. *Computer*, v. 38, n. 3, p. 50–60, 3 2005. ISSN 0018-9162. Citado na página 19.

HERATH, S.; HARANDI, M.; PORIKLI, F. Going deeper into action recognition: A survey. *Image and Vision Computing*, Elsevier, v. 60, p. 4–21, 4 2017. ISSN 0262-8856. Citado na página 39.

HOLTE, M. B. et al. Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments. *IEEE Journal on Selected Topics in Signal Processing*, v. 6, n. 5, p. 538–552, 2012. ISSN 19324553. Citado 2 vezes nas páginas 33 e 34.

HUSSEIN, M. E. et al. *Human Action Recognition Using a Temporal Hierarchy of Covariance Descriptors on 3D Joint Locations*. [S.l.], 2013. Citado na página 44.

JI, S. et al. *3D Convolutional Neural Networks for Human Action Recognition*. [S.l.], 2010. Citado na página 40.

JOHANSSON, G. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, v. 14, n. 2, p. 201–211, 6 1973. ISSN 00315117. Citado na página 43.

JOO, H. et al. Panoptic Studio: A Massively Multiview System for Social Motion Capture *. *The IEEE International Conference on Computer Vision (ICCV)*, 2015. Citado 2 vezes nas páginas 61 e 62.

JOSHI, A. et al. Comparing random forest approaches to segmenting and classifying gestures. *Image and Vision Computing*, Elsevier, v. 58, p. 86–95, 2 2017. ISSN 0262-8856. Citado 2 vezes nas páginas 50 e 109.

KADKHODAMOHAMMADI, A.; PADOY, N. A generalizable approach for multi-view 3D human pose regression. In: *3D HUMANS 2018: 1st International Workshop on HUman pose, Motion, Activities aNd Shape in 3D @ CVPR 2018*. [S.l.: s.n.], 2018. Citado 4 vezes nas páginas 36, 67, 72 e 73.

KE, Q. et al. SkeletonNet: Mining Deep Part Features for 3-D Action Recognition. *IEEE Signal Processing Letters*, v. 24, n. 6, p. 731–735, 6 2017. ISSN 1070-9908. Citado 2 vezes nas páginas 44 e 45.

KEROLA, T.; INOUE, N.; SHINODA, K. Spectral Graph Skeletons for 3D Action Recognition. In: . [S.l.]: Springer, Cham, 2015. p. 417–432. Citado na página 44.

KIM, J.; CHUNG, K.; KANG, M. Continuous Gesture Recognition using HLAC and Low-Dimensional Space. *Wireless Personal Communications*, Springer US, v. 86, n. 1, p. 255–270, 1 2016. ISSN 0929-6212. Citado na página 21.

KIM, Y. Dance motion capture and composition using multiple RGB and depth sensors. *International Journal of Distributed Sensor Networks*, SAGE PublicationsSage UK: London, England, v. 13, n. 2, p. 155014771769608, 2 2017. ISSN 1550-1477. Citado 2 vezes nas páginas 36 e 73.

KONEČN, J.; HAGARA, M.; SEMINÁR, K. M. One-Shot-Learning Gesture Recognition using HOG-HOF Features. *Journal of Machine Learning Research*, v. 15, p. 2513–2532, 2014. Citado na página 21.

KÜHNEL, C. et al. Im home: Defining and evaluating a gesture set for smart-home control. *International Journal of Human Computer Studies*, Academic Press, v. 69, n. 11, p. 693–704, 10 2011. ISSN 10715819. Citado na página 21.

LAPTEV, I.; LINDEBERG, T. *Velocity adaptation of space-time interest points **. [S.l.]. Citado 2 vezes nas páginas 40 e 41.

LARABA, S. et al. 3D skeleton-based action recognition by representing motion capture sequences as 2D-RGB images. *Computer Animation and Virtual Worlds*, Wiley-Blackwell, v. 28, n. 3-4, p. e1782, 5 2017. ISSN 15464261. Citado na página 27.

LEE, J.-E. et al. Human and Robot Localization Using Histogram of Oriented Gradients (HOG) Feature for an Active Information Display in Intelligent Space. *Advanced Science Letters*, v. 9, n. 1, p. 99–106, 4 2012. ISSN 19366612. Citado na página 19.

LEE, J.-H. Human Centered Ubiquitous Display in Intelligent Space. In: *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*. [S.l.]: IEEE, 2007. p. 22–27. ISBN 1-4244-0783-4. Citado na página 19.

LEE, J.-H.; APPENZELLER, G.; HASHIMOTO, H. An agent for intelligent spaces: functions and roles of mobile robots in sensored, networked and thinking spaces. In: *Proceedings of Conference on Intelligent Transportation Systems*. [S.l.]: IEEE, 1998. p. 983–988. ISBN 0-7803-4269-0. Citado na página 19.

LI, D. et al. Pose Guided Deep Model for Pedestrian Attribute Recognition in Surveillance Scenarios. In: *Proceedings - IEEE International Conference on Multimedia and Expo*. [S.l.]: IEEE Computer Society, 2018. v. 2018-July. ISBN 9781538617373. ISSN 1945788X. Citado na página 25.

LI, R. et al. Exploring 3D Human Action Recognition: from Offline to Online. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 3, p. 633, 2 2018. ISSN 1424-8220. Citado na página 22.

LI, S.; CHAN, A. B. 3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network. In: *Asian Conference on Computer Vision (ACCV)*. Singapore: [s.n.], 2014. Citado na página 34.

LIN, M. et al. Recurrent 3D pose sequence machines. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. v. 2017-Janua, p. 5543–5552. ISBN 9781538604571. Citado na página 35.

LIU, J. et al. Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 40, n. 12, p. 3007–3021, 12 2018. ISSN 0162-8828. Citado na página 47.

LIU, X. et al. Hidden states exploration for 3D skeleton-based gesture recognition. In: *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. p. 1846–1855. ISBN 9781728119755. Citado na página 50.

LIU, Z.; ZHANG, C.; TIAN, Y. 3D-based Deep Convolutional Neural Network for action recognition with depth sequences. *Image and Vision Computing*, Elsevier Ltd, v. 55, p. 93–100, 11 2016. ISSN 02628856. Citado na página 40.

LORA, M. et al. A geometric approach to multiple viewpoint human body pose estimation. *2015 European Conference on Mobile Robots (ECMR)*, p. 1–6, 2015. Citado 3 vezes nas páginas 31, 36 e 73.

LUN, R.; ZHAO, W. A Survey of Applications and Human Motion Recognition with Microsoft Kinect. *International Journal of Pattern Recognition and Artificial Intelligence*, World Scientific Publishing Company, v. 29, n. 05, p. 1555008, 8 2015. Citado 2 vezes nas páginas 25 e 27.

MAZO, M. et al. Wheelchair for physically disabled people with voice, ultrasonic and infrared sensor control. *Autonomous Robots*, Kluwer Academic Publishers, v. 2, n. 3, p. 203–224, 1995. ISSN 0929-5593. Citado 2 vezes nas páginas 19 e 20.

MCNEILL, D. *Language and gesture*. [S.l.]: Cambridge University Press, 2000. 409 p. ISBN 9780511620850. Citado na página 20.

MEHTA, D. et al. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera ACM Reference format: VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Trans. Graph. Article*, v. 36, n. 14, 2017. Citado 2 vezes nas páginas 35 e 74.

MONNIER, C.; GERMAN, S.; OST, A. A multi-scale boosted detector for efficient and robust gesture recognition. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.]: Springer Verlag, 2015. v. 8925, p. 491–502. ISBN 9783319161778. ISSN 16113349. Citado na página 50.

MURPHY, R. Human–Robot Interaction in Rescue Robotics. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, v. 34, n. 2, p. 138–153, 5 2004. ISSN 1094-6977. Citado na página 20.

NEVEROVA, N. et al. ModDrop: Adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 38, n. 8, p. 1692–1706, 2016. ISSN 01628828. Citado 3 vezes nas páginas 48, 49 e 50.

NEVEROVA, N. et al. Multi-scale deep learning for gesture detection and localization. In: . [S.l.: s.n.], 2014. Citado 17 vezes nas páginas 9, 22, 48, 49, 50, 76, 77, 81, 82, 83, 85, 86, 91, 96, 100, 108 e 109.

OFLI, F. et al. Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, Academic Press, v. 25, n. 1, p. 24–38, 1 2014. Citado 3 vezes nas páginas 9, 44 e 45.

PATRONA, F. et al. Motion analysis: Action detection, recognition and evaluation based on motion capture data. *Pattern Recognition*, Elsevier Ltd, v. 76, p. 612–622, 2018. ISSN 00313203. Citado 4 vezes nas páginas 21, 22, 27 e 48.

PIGOU, L. et al. Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video. *International Journal of Computer Vision*, Springer New York LLC, v. 126, n. 2-4, p. 430–439, 4 2018. ISSN 15731405. Citado na página 109.

PRESTI, L. L.; CASCIA, M. L. 3D skeleton-based human action classification: A survey. *Pattern Recognition*, Elsevier Science Inc., v. 53, n. C, p. 130–147, 5 2016. ISSN 00313203. Citado na página 44.

PRESTI, L. L.; CASCIA, M. L. 3D skeleton-based human action classification: A survey. *Pattern Recognition*, Elsevier Ltd, v. 53, p. 130–147, 5 2016. ISSN 00313203. Citado 2 vezes nas páginas 44 e 45.

QAMAR, A. M. et al. A Multi-Sensory Gesture-Based Occupational Therapy Environment for Controlling Home Appliances. In: *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR '15*. New York, New York, USA: ACM Press, 2015. p. 671–674. ISBN 9781450332743. Citado na página 21.

Qian Wan et al. Gesture recognition for smart home applications using portable radar sensors. In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. [S.l.]: IEEE, 2014. p. 6414–6417. ISBN 978-1-4244-7929-0. Citado na página 21.

RAHMANI, H. et al. *Real Time Action Recognition Using Histograms of Depth Gradients and Random Decision Forests*. [S.l.], 2014. Citado na página 22.

ROCHA, A. P. et al. Kinect v2 based system for Parkinson's disease assessment. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. [S.l.]: IEEE, 2015. p. 1279–1282. ISBN 978-1-4244-9271-8. Citado na página 29.

SAAD, K. K. et al. An Intelligent Robotic Aid System for Human Services. In: *Conference on Intelligent Robots in Factory, Field, Space, and Service*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 1994. Citado na página 19.

SHARAF, A. et al. *Real-time Multi-scale Action Detection From 3D Skeleton Data*. [S.l.]. Citado na página 47.

SHARAF, A. et al. Real-Time Multi-scale Action Detection from 3D Skeleton Data. In: *2015 IEEE Winter Conference on Applications of Computer Vision*. [S.l.]: IEEE, 2015. p. 998–1005. ISBN 978-1-4799-6683-7. Citado na página 48.

SHOTTON, J. et al. Real-time human pose recognition in parts from single depth images. In: *Cvpr 2011*. [S.l.]: IEEE, 2011. p. 1297–1304. ISBN 978-1-4577-0394-2. ISSN 1063-6919. Citado 2 vezes nas páginas 21 e 27.

SIMONYAN, K.; ZISSERMAN, A. *Two-Stream Convolutional Networks for Action Recognition in Videos*. [S.l.], 2014. Citado 2 vezes nas páginas 40 e 41.

SRIDHAR, S. et al. Real-time joint tracking of a hand manipulating an object from RGB-D input. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.]: Springer Verlag, 2016. v. 9906 LNCS, p. 294–310. ISBN 9783319464749. ISSN 16113349. Citado na página 25.

SRIVASTAVA, N. et al. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. [S.l.], 2014. v. 15, 1929–1958 p. Citado na página 99.

TAYLOR, C. J. Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image. *Computer Vision and Image Understanding*, Academic Press, v. 80, n. 3, p. 349–363, 12 2000. ISSN 1077-3142. Citado na página 33.

THOMAS, A. M. et al. Smart care spaces: needs for intelligent at-home care. *International Journal of Space-Based and Situated Computing*, v. 3, n. 1, p. 35, 2013. ISSN 2044-4893. Citado na página 19.

VEMULAPALLI, R.; ARRATE, F.; CHELLAPPA, R. *Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group*. [S.l.], 2014. Citado na página 44.

WANG, P. et al. RGB-D-based human motion recognition with deep learning: A survey. *Computer Vision and Image Understanding*, v. 171, p. 118–139, 6 2018. ISSN 10773142. Citado 2 vezes nas páginas 27 e 38.

WEI, P. et al. Concurrent action detection with structural prediction. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2013. p. 3136–3143. ISBN 9781479928392. Citado na página 44.

WEI, S.-E. et al. Convolutional Pose Machines. 1 2016. Citado 3 vezes nas páginas 32, 51 e 111.

WU, D. et al. Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 38, n. 8, p. 1583–1597, 8 2016. ISSN 0162-8828. Citado na página 108.

XIA, L.; CHEN, C.-C.; AGGARWAL, J. K. View invariant human action recognition using histograms of 3D joints. In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.]: IEEE, 2012. p. 20–27. ISBN 978-1-4673-1612-5. Citado na página 44.

XU, D. et al. Online Dynamic Gesture Recognition for Human Robot Interaction. *Journal of Intelligent & Robotic Systems*, Springer Netherlands, v. 77, n. 3-4, p. 583–596, 3 2015. ISSN 0921-0296. Citado na página 21.

YANG, Z. et al. Action Recognition with Spatio-Temporal Visual Attention on Skeleton Image Sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, p. 1–1, 2018. ISSN 1051-8215. Citado 3 vezes nas páginas 9, 46 e 47.

YAO, A. et al. Does Human Action Recognition Benefit from Pose Estimation? 2011. Citado na página 44.

ZANFIR MARIUS LEORDEANU, C. S. M. The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection. In: *Proceedings of ICCV 2013*. [S.l.: s.n.], 2013. Citado 6 vezes nas páginas 45, 47, 48, 77, 82 e 83.

ZHANG, C. *Human Activity Analysis using Multi-modalities and Deep Learning*. 115 p. Tese (Doutorado) — The City College of New York, 2016. Citado na página 94.

ZHANG, H. B. et al. A comprehensive survey of vision-based human action recognition methods. *Sensors (Switzerland)*, v. 19, n. 5, p. 1005, 2 2019. ISSN 14248220. Citado 3 vezes nas páginas 21, 38 e 47.

ZHAO, Y. et al. *Temporal Action Detection with Structured Segment Networks*. [S.l.]. Citado na página 22.

ZHU, G. et al. Multimodal Gesture Recognition Using 3-D Convolution and Convolutional LSTM. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 5, p. 4517–4524, 2017. ISSN 21693536. Citado 3 vezes nas páginas 9, 42 e 43.

ZHU, G. et al. Multimodal Gesture Recognition Using 3-D Convolution and Convolutional LSTM. *IEEE Access*, v. 5, p. 4517–4524, 2017. ISSN 2169-3536. Citado na página 21.

ZHU, G. et al. Continuous Gesture Segmentation and Recognition Using 3DCNN and Convolutional LSTM. *IEEE Transactions on Multimedia*, v. 21, n. 4, p. 1011–1021, 4 2019. ISSN 1520-9210. Citado na página 50.

ZHU, Y.; CHEN, W.; GUO, G. Fusing Spatiotemporal Features and Joints for 3D Action Recognition. Citado na página 25.

ZIEGLER, J.; NICKEL, K.; STIEFELHAGEN, R. *Tracking of the Articulated Upper Body on Multi-View Stereo Image Sequences*. [S.l.], 2006. Citado na página 34.