Pós-Graduação em Ciência da Computação

Ricardo Rossiter Barioni

HuTrain: um Framework para a Rápida Concepção de Datasets de Estimação de Poses Humanas

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2020

Ricardo Rossiter Barioni

HuTrain: um Framework para a Rápida Concepção de Datasets de Estimação de Poses Humanas

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Mídia e Interação
Orientadora: Veronica Teichrieb
Coorientador: Lucas Silva Figueiredo

Recife
2020

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

B253h Barioni, Ricardo Rossiter
HuTrain: um framework para a rápida concepção de datasets de estimação de poses humanas / Ricardo Rossiter Barioni. – 2020.
101 f.: il., fig., tab.

Orientadora: Veronica Teichrieb.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2020.
Inclui referências e apêndices.

1. Mídia e interação. 2. Redes neurais convolucionais. I. Teichrieb, Veronica (orientadora). II. Título.

006.7 CDD (23. ed.) UFPE - CCEN 2020 - 163

Ricardo Rossiter Barioni

HuTrain: um Framework para a Rápida Concepção de Datasets de Estimação de Poses Humanas

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 30/07/2020.

Orientadora: Profa. Dra. Veronica Teichrieb

BANCA EXAMINADORA

Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPE

Prof. Dr. Thales Vieira
Instituto de Computação / UFAL

Prof. Dr. Lucas Silva Figueiredo (Coorientador)
Voxar Labs / Centro de Informática / UFPE

Dedico este trabalho ao meu avô Dácio, que sempre estará presente.

AGRADECIMENTOS

A meus pais, Fátima e Ricardo, que entendem meu rumo e me apoiam sempre com muito amor. Ao meu irmão, Daniel, pelo companheirismo.

A Lucas e VT, por me orientarem e me inspirarem a sempre dar o meu melhor.

À família Voxar, por sempre estar presente quando precisei, principalmente nos momentos mais difíceis.

Aos amigos e amigas, por me darem força nos momentos de cansaço, alegria nos momentos de estresse e leveza nos momentos de quarentena. Em especial, aos Porteiros.

RESUMO

Algoritmos de rastreamento de corpos baseados em imagem são úteis em diversos cenários, tais como animações de avatares e interação por gestos para aplicações em realidade virtual (*VR*, *Virtual Reality*). Nos últimos anos, as melhores soluções presentes no estado da arte referentes ao rastreamento de corpos (de acordo com as melhores bases de dados da área) são fortemente baseadas em algoritmos de redes neurais convolucionais (*CNNs*, *Convolutional Neural Networks*) e utilizam extensas bases de dados para o treinamento e validação de seus modelos propostos. Apesar de essas soluções atingirem altas pontuações de precisão quando avaliadas em algumas dessas bases de dados, existem alguns desafios de rastreamento específicos (por exemplo, cenários *upside-down*) os quais não são bem cobertos e, consequentemente, não são corretamente rastreados. Ao invés de condensar todos os casos em uma única solução de *dataset*, nesta dissertação é proposto o *HuTrain*, um *framework* para a criação de bases de dados (*datasets*) para a tarefa de estimação de poses humanas (*HPE*, *Human Pose Estimation*) de uma forma fácil e rápida. O *HuTrain* é composto por uma série de passos, incluindo a calibração automática de câmeras, estimação de poses humanas refinadas e conversão para formatos de bases de dados conhecidas. Com o *HuTrain*, o usuário é capaz gerar bases de dados referentes a poses humanas, atacando desafios de rastreamento específicos para o contexto de aplicação desejado, sem a necessidade de anotar instâncias de poses humanas manualmente. Neste cenário, os resultados apotam a utilização do *HuTrain* como sendo uma solução rápida, prática e efetiva para a concepção de *datasets* de estimação de poses humanas, sendo capaz de gerar um *dataset* com dezenas de milhares de amostras em poucas horas.

Palavras-chave: Geração de *datasets*. Poses humanas. Captura de movimento. Calibração de câmeras. Redes neurais convolucionais.

ABSTRACT

Image-based body tracking algorithms are useful in several scenarios, such as avatar animations and gesture interaction for Virtual Reality (VR) applications. In the last few years, the best-ranked solutions presented on the state of the art of body tracking (according to the most popular datasets in the field) are intensively based on Convolutional Neural Networks (CNNs) algorithms and use large datasets for the training and validation of their proposed models. Although these solutions achieve high precision scores when evaluated with some of these datasets, there are specific tracking challenges (for example, upside-down scenarios) that are not well-covered and, therefore, not correctly tracked. Instead of lurking an all-in-one dataset solution for all cases, in this dissertation it is proposed HuTrain, a framework for creating Human Pose Estimation (HPE) datasets quickly and easily. HuTrain is composed by a series of steps, including automatic camera calibration, refined human pose estimation, and conversion to known dataset formats. With HuTrain, the user is capable of generating human pose datasets, targeting specific tracking challenges for the desired application context, with no need to annotate human pose instances manually. In this scenario, results show the usage of *HuTrain* as a quick, practical and effective solution for the conception of human pose estimation datasets, being able to generate a dataset with tens of thousands of samples in a few hours.

Keywords: Dataset generation. Human poses. Motion capture. Camera calibration. Convolutional neural networks.

LISTA DE FIGURAS

Figura 1 – Exemplos de contextos específicos de presença de poses humanas. . . . 20
Figura 2 – Representação pictorial do corpo humano em partes, cujas conexões são deformáveis. 23
Figura 3 – Modelo de regressão da localização de *keypoints* baseado em *CNNs* (TOSHEV; SZEGEDY, 2014). Destaca-se a utilização de camadas convolucionais ao longo do modelo. 24
Figura 4 – Arquitetura para a estimação de *heatmaps* de localizacão de *keypoints* (TOMPSON et al., 2015). Nela, destaca-se o *output* como sendo um conjunto de *heatmaps* para a localização de cada um dos *keypoints* da pose humana. 25
Figura 5 – Modelo de *cnn* multi-estágios (WEI et al., 2016). Nele, destaca-se o uso da supervisão intermediária, na qual o *output* é avaliado ao longo da arquitetura. 25
Figura 6 – Arquitetura de *CNN* empregando o uso de módulos *encoder-decoder* chamados *hourglasses* (**b.**) (NEWELL; YANG; DENG, 2016). 26
Figura 7 – Modelo de *CNN* no qual os *Part Confidence Maps* e *Part Affinity Fields*, estimados por uma *CNN*, são utilizados para agregar os *keypoints* em suas respectivas poses (CAO et al., 2017). 27
Figura 8 – Modelo de *CNN* cuja menor complexidade, quando comparada a outros modelos, apresenta resultados competitivos na tarefa de *HPE* (XIAO; WU; WEI, 2018). 28
Figura 9 – Arquitetura de *CNN* cujas *features* de alta resolução são mantidas ao longo da rede, enquanto novas conexões com menores resoluções são aprendidas (SUN et al., 2019). 29
Figura 10 – Modelo de regressão baseado em *CNNs* (LI; CHAN, 2014). Seu módulo de regressão assimila-se ao adotado pelo estimador de pose 2D proposto por Toshev et al. (TOSHEV; SZEGEDY, 2014). 30
Figura 11 – Abordagem adotada por Chen et al. (CHEN; RAMANAN, 2017). Ao estimar a pose 2D, esta é analisada a um biblioteca de poses 3D prégravadas, a fim de descobrir a sua pose 3D correspondente. 31
Figura 12 – Módulo de estimação de pose 3D proposto por Martinez et al. (MARTINEZ et al., 2017). A partir de uma pose 2D de entrada, foi utilizada uma arquitetura MLP para a estimação de pose 3D final. 32
Figura 13 – Arquitetura proposta por Sun et al. (SUN et al., 2018). Seu diferencial é a utilização de mapas de confiança tridimensionais para localizar cada um dos *keypoints*. 32

Figura 14 – Arquitetura desenvolvida por Rhodin et al. (RHODIN; SALZMANN; FUA, 2018). Seu diferencial é o treinamento não-supervisionado de um extrator de *features* tridimensionais latentes, otimizando o processo de treinamento de um modelo de estimação de poses 3D. 33
Figura 15 – *Pipeline* do *framework*. **a.** Vídeos *RGB* de entrada; **b.** Estimação inicial de poses humanas 2D seguida da reidentificação das pessoas em múltiplas vistas; **c.** Estimação dos parâmetros extrínsecos das câmeras baseada em humanos; **d.** Triangulação de poses humanas (setas laranjas) e estimações refinadas de poses humanas 2D (setas rosas); **e.** Conversão para formatos de *datasets* conhecidos. 42
Figura 16 – Exemplos de estimação de poses humanas 2D oriundas do algoritmo *OpenPose* (CAO et al., 2017). 43
Figura 17 – Exemplo de re-identificação de pessoas em múltiplos pontos de vista. . 44
Figura 18 – Modelo de câmera *GoPro Hero 7*. 49
Figura 19 – Tabuleiro de xadrez, utilizado para estimar os parâmetros intrínsecos da câmera. 50
Figura 20 – Sala utilizada para a realização dos testes. As câmeras utilizadas foram posicionadas em uma viga, ao redor da sala (círculos em vermelho) e apontadas para o centro (setas em vermelho). 51
Figura 21 – Campo de visão de cada uma das 8 câmeras utilizadas. 52
Figura 22 – Ferramenta de visualização das capturas de movimentos geradas pelo *HuTrain*. 53
Figura 23 – Resultados qualitativos do *dataset* gerado. 54
Figura 24 – Aspectos abordados pelo *UEQ* e seus respectivos itens. 60
Figura 25 – Itens presentes no *UEQ* utilizados na avaliação. Nota-se a presença de características positivas em ambos os lados (direito e esquerdo). Além disso, itens pertencentes aos mesmos aspectos são distribuídos em uma ordem aleatória. 61
Figura 26 – Média dos resultados do *UEQ* de cada um dos itens respondidos pelos usuários. Estes itens são agrupados nos aspectos de *attractiveness* (roxo), *perspicuity* (verde), *efficiency* (azul escuro) e *dependability* (azul claro). 62
Figura 27 – Média dos resultados do *UEQ* (pontuações em preto) para os aspectos avaliados (*attractiveness*, *perspicuity*, *efficiency* e *dependability*). Cada uma das escalas é representada por uma cor: Excelente (verde escuro), bom (verde), acima da média (verde claro), abaixo da média (laranja) e ruim (vermelho). 63

Figura 28 – Estrutura do *System Usability Scale*. As sentenças são respondidas em uma escala *Likert* de 1 a 5. Além disso, as sentenças ímpares possuem conotações positivas, enquanto que as pares possuem conotações negativas. 65
Figura 29 – Média de pontuações de cada sentença do *SUS*, bem como a média das pontuações individuais. 67
Figura 30 – Resultados da avaliação do *HuTrain* no *System Usability Scale* em relação aos graus de aceitabilidade (BANGOR; KORTUM; MILLER, 2008), mapeamento de pontuação para características (BANGOR; KORTUM; MILLER, 2009) e pontuação no *SUS* (BROOKE, 1996). 68
Figura 31 – Exemplos de diferentes representações de poses humanas. **a.** *MS COCO* (LIN et al., 2014); **b.** *MPII Human Pose* (ANDRILUKA et al., 2014); **c.** *Microsoft Kinect*; **d.** *LSP* (JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011); **e.** *DensePose* (GüLER; NEVEROVA; KOKKINOS, 2018). 84
Figura 32 – Exemplos de tecnologias de captura de movimento utilizando aparatos externos. 85
Figura 33 – Exemplos de sensores de profundidade para captura de movimento (imagens à esquerda), bem como os seus respectivos *outputs* (imagens à direita). De cima para baixo: *Microsoft Kinect*, *OpenNI* e *Orbecc Astra*. 86
Figura 34 – Exemplos de aparatos eletrônicos contendo câmeras *RGB*. 87
Figura 35 – Exemplos de desafios de estimações de poses humanas em imagens *RGB*. 89
Figura 36 – Exemplo de estimação de poses humanas utilizando abordagens *top-down* (**a.**) e *bottom-up* (**b.**). 90
Figura 37 – Exemplo de estimação de pose humana 2D. 91
Figura 38 – Exemplo de estimação de pose humana 3D. 91
Figura 39 – Visualização da operação de convolução, com ênfase na operação de produto interno entre o *input* e o *kernel*. 93
Figura 40 – Visualização da operação de convolução com a presença de múltiplos canais na entrada. A operação de produto interno é realizada entre os canais do *kernel* correspondentes (cores similares). 93
Figura 41 – Visualização da operação de convolução com a presença de múltiplos canais no *input* e mais de um filtro no *kernel*.. Para cada filtro, é gerado como saída um mapa de *features*. 94
Figura 42 – Visualização da operação de convolução em um *input* de um canal, com ênfase nos saltos entre as subregiões do *input*. Nesta convolução, foi aplicado um *stride* igual a 2. 95
Figura 43 – Visualização de uma operação de *Max Pooling* em um *input* contendo um canal, com *stride* igual a 2. 95

Figura 44 – Visualização de uma operação de *Average Pooling* em um *input* contendo um canal, com *stride* igual a 2. 96
Figura 45 – Visualização de uma operação de *ReLU* em um *input* contendo um canal. 96
Figura 46 – Modelo de câmera *pinhole*. 97
Figura 47 – Representação da transformação de coordenadas de mundo para coordenadas de câmera utilizando a matriz de parâmetros extrínsecos $[R|t]$. 99
Figura 48 – Correspondência entre os pontos 2D e o plano epipolar definido utilizando a geometria epipolar. 100

LISTA DE TABELAS

Tabela 1 – Visão geral dos principais *MLDs*. Para cada *dataset* foram coletadas as seguintes informações (caso disponíveis): o número de indivíduos presentes por *sample* (**# Por Imagem**), a dimensionalidade das poses (**Dim.**), o número de *samples* (**# *Samples***), a origem dos conteúdos visuais dos *samples* (**Fonte dos Dados**), o contexto das poses presentes no *dataset* (**Contexto**) e a origem dos *workers* envolvidos no processo de anotação (**Método de *Crowdsourcing***). 35

Tabela 2 – Visão geral dos principais *AGDs* (incluindo o *dataset* concebido ao utilizar o *HuTrain*). Para cada *dataset* foram coletadas as seguintes informações (caso disponíveis): o número de indivíduos presentes por *sample* (**# Por Imagem**), a dimensionalidade das poses (**Dim.**), o número de *samples* (**# *Samples***), o número de câmeras (**# Câmeras**), a presença de marcadores aos corpos dos usuários capturados (***Marker-Based*?**), a origem dos conteúdos visuais dos *samples* (**Fonte dos Dados**) e o contexto das poses presentes no *dataset* (**Contexto**). . . . 38

Tabela 3 – Visão geral dos principais *SGDs*. Para cada *dataset* foram coletadas as seguintes informações (caso disponíveis): o número de indivíduos presentes por *sample* (**# Por Imagem**), o número de *samples* (**# *Samples***), a origem das poses humanas utilizadas (**Fonte das Poses**), o modelo de corpo virtual aplicado às poses humanas (**Modelo de Corpo**), a origem das texturas de corpos e roupas utilizadas (**Texturas**), o método utilizado para a geração de *samples* fotorrealistas (**Fotorrealismo**) e o contexto das poses presentes no *dataset* (**Contexto**). 40

Tabela 4 – Tempo de execução de cada uma das etapas necessárias para a geração de um *dataset* utilizando o *HuTrain*. 55

Tabela 5 – Custo de tempo para a geração dos *datasets* para *HPE* existentes na academia, incluindo o proposto. Os custos de tempo dos *datasets* marcados com asterisco (*) foram obtidos atráves do contato com os autores envolvidos, enquanto que os não marcados foram extraídos diretamente dos seus respectivos *papers*. 56

Tabela 6 – Resultados do erro de reprojeção médio para cada um dos tipos de *keypoints*. São apresentados os resultados utilizando o *HRNet* (SUN et al., 2019) junto ao *YOLOv3* (REDMON; FARHADI, 2018), o *OpenPose* (CAO et al., 2017) e o proposto pelo *HuTrain*. Os erros marcados em negrito (para cada tipo de *keypoint*) representam os melhores resultados. 58

Tabela 7 – Usuários participantes da avaliação, bem como os seus respectivos graus de *expertise*, autoavaliados de 1 a 5, na linguagem de programação *Python* e na plataforma de desenvolvimento *Unity3D*. 58

LISTA DE ABREVIATURAS E SIGLAS

6-DoF	Six Degrees of Freedom
AGD	Automatically Generated Dataset
AGDs	Automatically Generated Datasets
AMT	Amazon Mechanical Turk
BA	Bundle Adjustment
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
DCNNs	Deep Convolutional Neural Networks
fps	Frames per Second
GB	Gigabyte
GPU	Graphic Processing Unit
H-BCPC	Human-Based Camera Poses Calibration
HMD	Head-Mounted Display
HPE	Human Pose Estimation
IMUs	Inertial Measurement Units
MLDs	Manually Labeled Datasets
MLP	Multilayer Perceptron
MMRE	Maximum Mean Reprojection Error
MoCap	Motion Capture
PAFs	Part Affinity Fields
PnP	Perspective-n-Point
RAM	Random Access Memory
re-ID	Re-Identification
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
SfM	Structure from Motion
SGD	Synthetically Generated Dataset
SGDs	Synthetically Generated Datasets
SMPL	Skinned Multi-Person Linear
SUS	System Usability Scale

SVD	Singular Value Decomposition
SVM	Support Vector Machine
UEQ	User Experience Questionnaire
VR	Virtual Reality

SUMÁRIO

1 INTRODUÇÃO 18
1.1 PERGUNTAS 20
1.2 OBJETIVOS 21
1.2.1 Objetivo Principal 21
1.2.2 Objetivos Específicos 21
1.3 CONTRIBUIÇÕES 21
1.4 ORGANIZAÇÃO DO TRABALHO 22

2 REVISÃO DA LITERATURA 23
2.1 MÉTODOS DE ESTIMAÇÃO DE POSES HUMANAS 23
2.1.1 Métodos de Estimação de Poses Humanas 2D 23
2.1.2 Métodos de Estimação de Poses Humanas 3D 29
2.2 *DATASETS* PARA ESTIMAÇÃO DE POSES HUMANAS 33
2.2.1 *Datasets* Manualmente Anotados 34
2.2.2 *Datasets* Automaticamente Gerados 36
2.2.3 *Datasets* Sinteticamente Gerados 39

3 FRAMEWORK DE GERAÇÃO DE DATASETS PARA ESTIMAÇÕES DE POSES HUMANAS 41
3.1 MÓDULO DE ESTIMAÇÃO INICIAL DE POSES HUMANAS 2D 42
3.2 MÓDULO DE CALIBRAÇÃO DOS PARÂMETROS EXTRÍNSECOS DAS CÂMERAS BASEADA EM HUMANOS 45
3.3 MÓDULO DE ESTIMAÇÃO DE POSES HUMANAS 3D 47
3.4 MÓDULO DE CONVERSÃO DE FORMATOS DE *DATASETS* 48

4 EXPERIMENTOS E RESULTADOS 49
4.1 CONFIGURAÇÃO E IMPLEMENTAÇÃO 49
4.2 CONCEPÇÃO DO *DATASET* 50
4.3 VISUALIZAÇÃO DOS RESULTADOS 52
4.4 TEMPO DE EXECUÇÃO 55
4.5 RESULTADOS QUANTITATIVOS 56
4.5.1 Erro de Reprojeção Médio 57
4.6 USABILIDADE 58
4.6.1 *User Experience Questionnaire* 59
4.6.2 *System Usability Scale* 63

5 DISCUSSÃO 69

5.1 ANÁLISE DO TEMPO DE EXECUÇÃO 69
5.2 ANÁLISE DOS RESULTADOS QUANTITATIVOS 70
5.3 ANÁLISE DOS TESTES DE USABILIDADE 72

6 CONCLUSÕES 74

REFERÊNCIAS 75

APÊNDICE A – ESTIMAÇÃO DE POSES HUMANAS 84

APÊNDICE B – REDES NEURAIS CONVOLUCIONAIS 92

APÊNDICE C – MATRIZ DE CÂMERA 97

APÊNDICE D – LISTA DE BIBLIOTECAS DE PYTHON UTILIZADAS 101

1 INTRODUÇÃO

Estimação de poses humanas (*HPE*, *Human Pose Estimation*) é a tarefa de localizar um conjunto de pontos característicos (*keypoints*) de corpos humanos os quais correspondam à configuração espacial de indivíduos presentes em uma imagem. Tal tarefa contempla uma série de desafios, tais como a incerteza do número de pessoas presentes, a oclusão de partes do corpo por outros elementos, pessoas ou até o próprio corpo (esta também chamada de *self-occlusion*), variações de iluminação e também o conjunto de possíveis poses as quais um indivíduo pode assumir. Diversas aplicações demandam esta tarefa vinculada à visão computacional, tais como o reconhecimento de atividades, animação de modelos virtuais e interação natural a partir da execução de gestos no ar.

Trabalhos recentes relacionados a *HPE* mostram melhorias significativas ao utilizarem abordagens baseadas em redes neurais convolucionais profundas (*DCNNs*, *Deep Convolutional Neural Networks*) em imagens *RGB* (CAO et al., 2017; WEI et al., 2016; NEWELL; YANG; DENG, 2016; TOSHEV; SZEGEDY, 2014; MEHTA et al., 2017; SUN et al., 2019; MARTINEZ et al., 2017; GüLER; NEVEROVA; KOKKINOS, 2018; CHU et al., 2017), as quais têm apresentado resultados excepcionais na tarefa de *HPE* (ANDRILUKA et al., 2014; LIN et al., 2014; JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011; SAPP; TASKAR, 2013; TOMPSON et al., 2014; IONESCU et al., 2014) ao substituirem técnicas manualmente modeladas (YANG; RAMANAN, 2013; MORI; MALIK, 2006; REN; BERG; MALIK, 2005; BOURDEV et al., 2010; TAYLOR, 2000). Com o objetivo de gerar estimações de poses confiáveis, modelos de *CNNs* são submetidos a um processo de treinamento, no qual são alimentadas amostras de dados contendo imagens representativas do problema bem como anotações referentes ao problema a ser atacado, com o objetivo de convergir o modelo à sua tarefa. Para tal, é necessária uma base de dados de imagem-anotação a qual contemple os desafios de *HPE* mencionados.

Os *benchmarks* de *HPE* variam em função de quais desafios eles buscam atacar: alguns *datasets* buscam cobrir o máximo de cenários possíveis (LIN et al., 2014; ANDRILUKA et al., 2014; GüLER; NEVEROVA; KOKKINOS, 2018), enquanto que outros focam em tarefas relacionadas a esportes (JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011; WANG; TRAN; LIAO, 2011), atividades ao ar livre (SAPP; TASKAR, 2013; TOMPSON et al., 2014; MEHTA et al., 2017; MEHTA et al., 2018; IQBAL; MILAN; GALL, 2017) ou até mesmo pessoas utilizando um *Head-Mounted Display* (*HMD*) (TOME et al., 2019). Apesar de a maioria dos casos possíveis de poses serem contemplados pelos *datasets* disponíveis na academia, ainda existem cenários os quais os algoritmos de *HPE* falham em cobrir devido à falta de amostras (*samples*) fornecidas durante a etapa de treinamento do modelo (CAO et al., 2017; TOYODA; KONO; REKIMOTO, 2019). Para tal, a solução concentra-se em criar novos *samples* de treinamento. Contudo, o processo de concepção de um *dataset* para

HPE pode trazer alguns obstáculos.

Em relação a *benchmarks* de *HPE* criados a partir da anotação manual feita por pessoas, a quantidade desejada de *samples* do *dataset* define os custos financeiro (SAPP; TASKAR, 2013; TOMPSON et al., 2014) e de tempo (LIN et al., 2014; JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011). Além disso, pessoas podem ter diferentes entendimentos a respeito da localização precisa de partes do corpo humano, os que pode introduzir ambiguidades no processo (LIN et al., 2014). Desta forma, as anotações manuais devem ser cuidadosamente analisadas a fim de que a qualidade desejada seja garantida.

Em relação aos *datasets* de *HPE* automaticamente gerados, estes superaram o trabalhoso e impreciso processo de anotações manuais (SIGAL; BALAN; BLACK, 2010; IONESCU et al., 2014; JOO et al., 2015; RHODIN et al., 2016; TRUMBLE et al., 2017; MARCARD et al., 2018; CHARLES et al., 2016; CHARLES et al., 2014). Alguns destes buscam montar um ambiente de captura controlado a partir de múltiplas câmeras em uma configuração de fora pra dentro (*outside-in*) (IONESCU et al., 2014; RHODIN et al., 2016; SIGAL; BALAN; BLACK, 2010), produzindo estimações de pose com uma maior qualidade, enquanto que outros métodos utilizam técnicas de *HPE* provenientes de terceiros para atacar cenários específicos (CHARLES et al., 2016; CHARLES et al., 2014; PFISTER et al., 2014; CHARLES et al., 2013). Em ambos os casos, a acurácia do algoritmo utilizado determina a qualidade do *dataset* gerado. Apesar de diversos trabalhos buscarem gerar novos *benchmarks* atacando cenários específicos de *HPE*, usuários não-especialistas não conseguem gerar *datasets* automaticamente de acordo com as suas necessidades de aplicação, em função da natureza técnica desafiadora.

Uma outra possível abordagem é a geração sintética de *datasets* para *HPE*. Neste caso, um gerador de *dataset* recebe como entrada um conjunto de capturas de movimento e busca criar novos pares imagem-anotação para *HPE* a partir da geração de imagens fotorrealísticas as quais descrevam as informações de pose recebidas (VAROL et al., 2017; CHEN et al., 2016; LASSNER et al., 2017; FABBRI et al., 2018; XU et al., 2019; TOME et al., 2019). Apesar dessas abordagens apresentarem realismo suficiente, estas não focam na problemática de geração de *samples* para poses não capturadas, e sim na renderização de conteúdos visuais baseados na realidade.

Apesar de existirem *benchmarks* os quais contemplam os mais variados contextos de *HPE*, ainda haverá cenários de pose não vistos, como por exemplo danças culturais regionais, novas formas de interação com objetos ou até mesmo esportes específicos (Figura 1). Uma forma de atender a esses cenários é prover aos usuários um meio de criar o seu próprio *dataset* especializado. Contudo, ao revisar a criação dos *datasets* existentes na academia, evidencia-se que esta tarefa demanda tempo, recursos financeiros e *know-how* técnico.

Figura 1 – Exemplos de contextos específicos de presença de poses humanas.

1.1 PERGUNTAS

Considerando este cenário, o presente trabalho visa auxiliar o processo de criação de *datasets* para estimações de poses humanas. Para tal, foram elicitadas duas perguntas fundamentais neste contexto:

- O quão **fácil** é criar um *dataset* para a tarefa de *HPE*?
- O quão **rápido** é criar um *dataset* para a tarefa de *HPE*?

Nesse contexto, a facilidade de um *dataset* de ser concebido refere-se à acessibilidade de usuários não especialistas poderem realizar tal atividade, enquanto que um *dataset* rápido de ser elaborado não demanda longos investimentos de tempo por parte dos usuários.

1.2 OBJETIVOS

1.2.1 Objetivo Principal

O principal objetivo deste trabalho é o desenvolvimento de um *framework* semi-automático com o objetivo de auxiliar na geração de *datasets* para *HPE* em três dimensões, cobrindo etapas relacionadas à calibração de câmeras no ambiente, estimação de poses e anotação de *samples* a partir de gravações realizadas pelo usuário, contemplando um fácil e ágil processo a partir de câmeras RGB. Por fim, o *framework* proposto lida com o processo de conversão das anotações geradas para formatos dos *datasets* existentes mais populares, a fim de que a utilização do novo *dataset* no treinamento e validação de métodos de *HPE* seja a mais transparente possível para o usuário.

1.2.2 Objetivos Específicos

Em relação aos objetivos específicos deste trabalho, podemos destacar:

- Estudo de técnicas de estimação de poses humanas em duas (2D) e três dimensões (3D) em uma abordagem cronológica, considerando as contribuições-chave de cada método proposto para o incremento dos resultados nos principais *benchmarks*;
- Compreensão das dificuldades na elaboração de um *dataset* e automatização das etapas presentes no processo, a fim de facilitar e agilizar tal tarefa;
- Implementação de um *framework* para a geração de *datasets* para estimação de poses humanas;
- Concepção de um *dataset* para estimação de poses humanas utilizando o *framework* proposto, a fim de avaliar quantitativamente os *samples* gerados, bem como analisar o custo de tempo envolvido em sua geração e a usabilidade do *framework*.

1.3 CONTRIBUIÇÕES

Dentre as contribuições do trabalho, destacam-se:

- A proposta de um *framework* para a realização de captura de movimento utilizando câmeras convencionais;
- Um algoritmo de calibração de câmeras baseado em humanos (*H-BCPC*, *Human-Based Camera Poses Calibration*), o qual estima as poses (parâmetros extrínsecos) das câmeras a partir das localizações das poses humanas estimadas em cada um dos pontos de vista;

- Um estudo aprofundado dos principais *datasets* de estimação de poses humanas existentes na academia, com ênfase no processo de concepção, contextos de poses contemplados e recursos envolvidos em suas concepções.

1.4 ORGANIZAÇÃO DO TRABALHO

O presente trabalho está estruturado da seguinte maneira: nos Apêndices A, B e C, são explanados alguns conceitos-base para o desenvolvimento deste documento, concentrando-se nas áreas de estimação de poses humanas, redes neurais convolucionais e visão computacional. O Capítulo 2 aborda os principais trabalhos do estado da arte relacionados a estimações de poses humanas, bem como os *benchmarks* existentes para esta tarefa. O Capítulo 3 descreve detalhadamente o *framework* desenvolvido, dando ênfase a sua arquitetura linear e modularizada. O Capítulo 4 expõe os experimentos realizados, bem como os resultados obtidos no contexto quantitativo e de performance. No Capítulo 5, analisa-se os resultados obtidos. Por fim, o Capítulo 6 conclui o trabalho, apresentando suas principais contribuições e trabalhos futuros.

2 REVISÃO DA LITERATURA

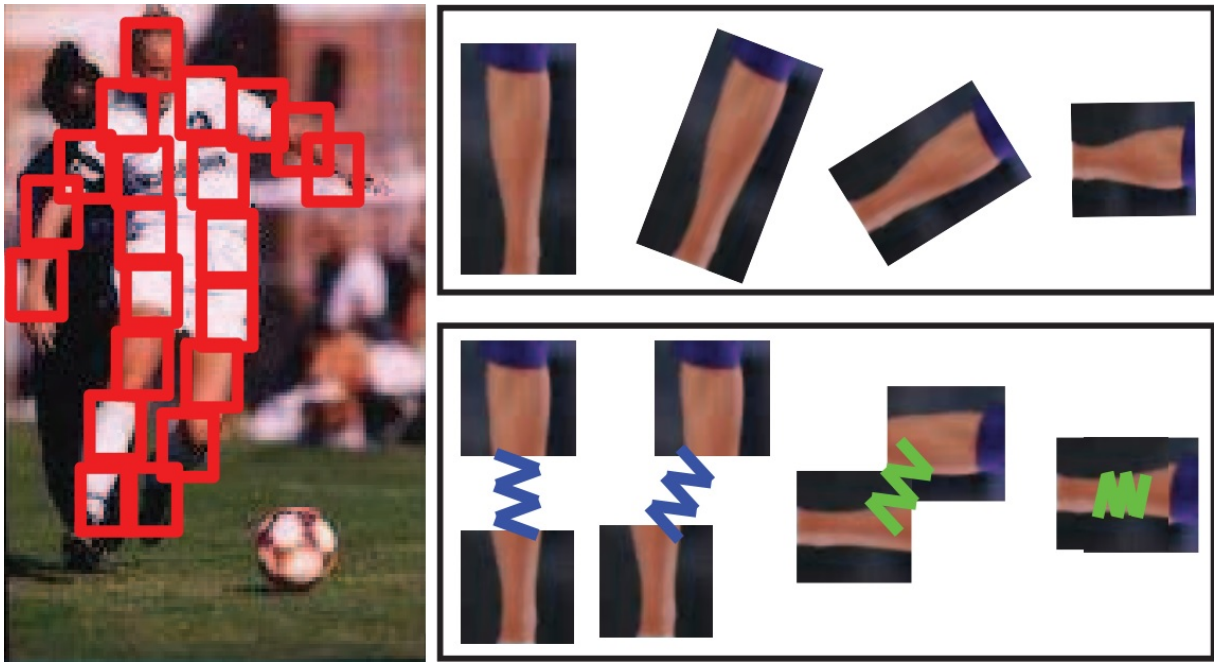
Focamos a revisão de literatura em dois tópicos. A Seção 2.1 discute os avanços referentes aos métodos de estimação de poses humanas, incluindo as abordagens 2D e 3D. A Seção 2.2 traz uma visão geral a respeito dos principais *benchmarks* para estimações de poses humanas existentes na literatura, em uma perspectiva baseada no método de concepção empregado para gerá-los.

2.1 MÉTODOS DE ESTIMAÇÃO DE POSES HUMANAS

2.1.1 Métodos de Estimação de Poses Humanas 2D

Esta tarefa consiste em estimar as coordenadas Cartesianas (x, y) das partes do corpo das pessoas presentes em uma imagem. Abordagens clássicas buscam solucionar o problema de *HPE* em duas dimensões a partir de estruturas pictoriais, no qual o corpo humano é separado em um conjunto de partes em uma configuração deformável (Figura 2). Esta configuração deformável pode ser comparada a "molas", as quais conectam as diferentes partes do corpo e devem ser arranjadas a ponto de corresponderem com o indivíduo na imagem (YANG; RAMANAN, 2013; BOURDEV et al., 2010; MORI; MALIK, 2006; REN; BERG; MALIK, 2005; PISHCHULIN et al., 2013).

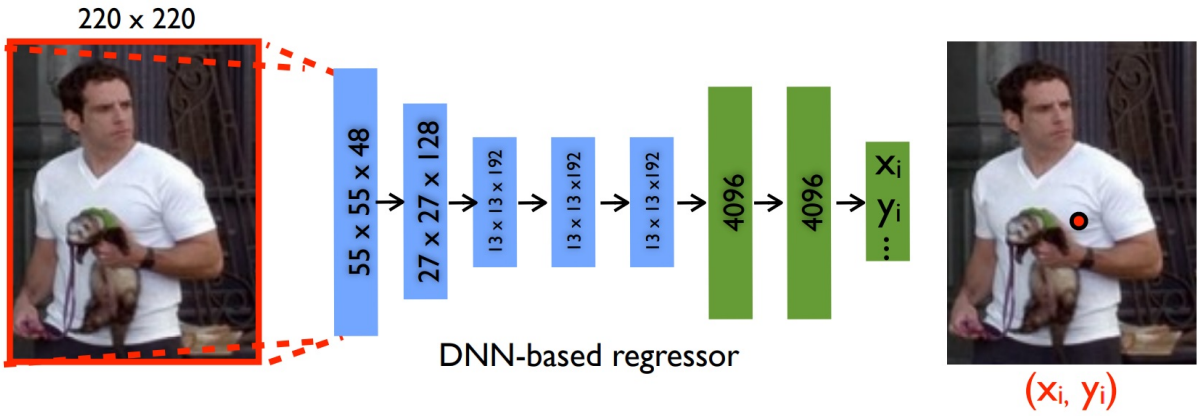
Figura 2 – Representação pictorial do corpo humano em partes, cujas conexões são deformáveis.



Fonte: (YANG; RAMANAN, 2013)

Nos tempos recentes, *DCNNs* trouxeram um incremento significativo na acurácia dos métodos de *HPE* (TOSHEV; SZEGEDY, 2014; TOMPSON et al., 2015; WEI et al., 2016; CARREIRA et al., 2016; NEWELL; YANG; DENG, 2016; XIAO; WU; WEI, 2018; SUN et al., 2019; CAO et al., 2017) (Apêndice B). Pela primeira vez, Toshev et al. aplicou técnicas de *Deep Learning* na tarefa de estimação de poses humanas, sendo assim um marco no direcionamento da pesquisa (TOSHEV; SZEGEDY, 2014). Nele, é proposto um modelo de regressão baseado em redes neurais convolucionais (*CNNs*, *Convolutional Neural Networks*) cuja saída é um conjunto de valores $\{(x_i, y_i) \mid i \in \{1, 2, \ldots, k\}\}$, no qual (x_i, y_i) refere-se à localização do *keypoint* i do corpo e k representa o número de *keypoints* que representam o corpo (Figura 3). Na ocasião, tal abordagem trouxe performances que superaram o estado da arte nos *datasets LSP* (JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011) e *FLIC* (SAPP; TASKAR, 2013).

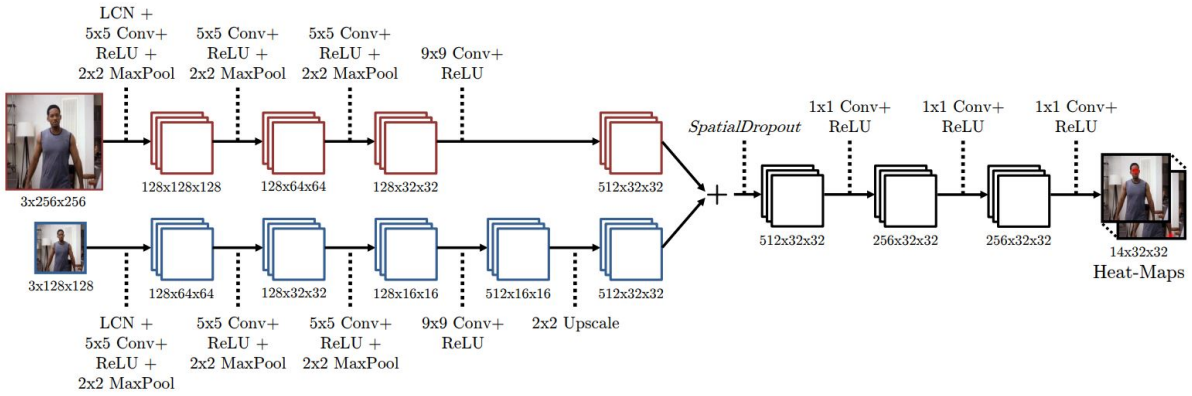
Figura 3 – Modelo de regressão da localização de *keypoints* baseado em *CNNs* (TOSHEV; SZEGEDY, 2014). Destaca-se a utilização de camadas convolucionais ao longo do modelo.



Fonte: (TOSHEV; SZEGEDY, 2014)

Seguindo a proposta de utilização de redes neurais convolucionais, Tompson et al. introduziu uma arquitetura de *CNN* cuja saída são mapas de confiança (*heatmaps*), nos quais a sua predição refere-se à probabilidade de ocorrência de um *keypoint* para cada *pixel* da imagem de entrada (TOMPSON et al., 2015). Neste caso, o conjunto de *heatmaps* é definido por $H \in \mathbb{R}^{k \times w \times h}$, onde w e h são as dimensões da imagem e k o número de *keypoints*. Apesar deste trabalho estimar a pose de apenas uma pessoa por imagem, a introdução de *heatmaps* deu margem à criação de métodos de *HPE* para múltiplas pessoas presentes, uma vez que tornou-se possível localizar mais de uma ocorrência de um determinado *keypoint* na imagem (Figura 4). No momento de sua publicação, seus resultados superaram o estado da arte nos *datasets FLIC* (SAPP; TASKAR, 2013) e *MPII Human Pose* (ANDRILUKA et al., 2014).

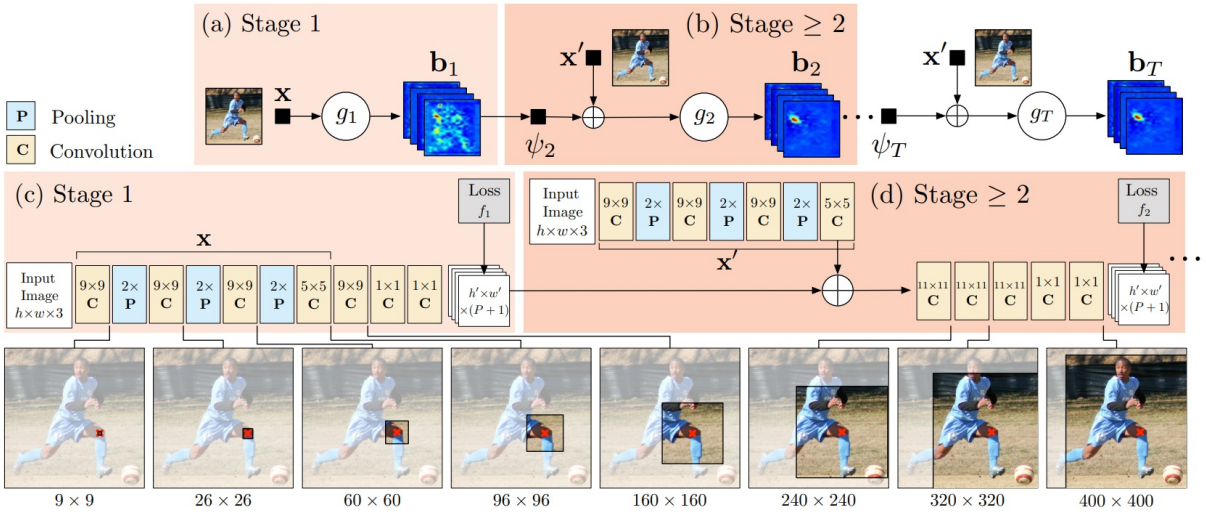
Figura 4 – Arquitetura para a estimação de *heatmaps* de localizacão de *keypoints* (TOMPSON et al., 2015). Nela, destaca-se o *output* como sendo um conjunto de *heatmaps* para a localização de cada um dos *keypoints* da pose humana.



Fonte: (TOMPSON et al., 2015)

Com o intuito de incrementar a localização dos *keypoints*, Wei et al. propôs a utilização de supervisão intermediária das predições dos *heatmaps*; o modelo é dividido em estágios, nos quais os *heatmaps* são utilizados, junto à imagem original, como entrada para o estágio seguinte, refinando-o (WEI et al., 2016). A ideia, originalmente proposta por Ramakrishna et al. (RAMAKRISHNA et al., 2014), visa incorporar relações espaciais entre diferentes *keypoints* detectados (Figura 5). Aliado a técnicas de redes neurais convolucionais, esta técnica superou os resultados existentes no estado da arte referentes aos *datasets LSP* (JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011), *FLIC* (SAPP; TASKAR, 2013) e *MPII Human Pose* (ANDRILUKA et al., 2014).

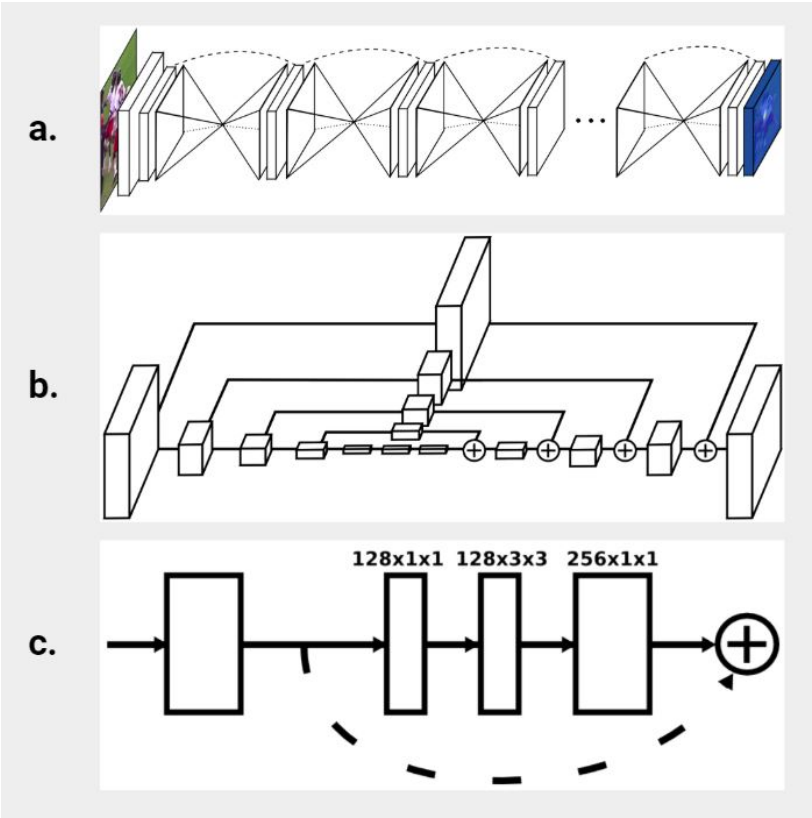
Figura 5 – Modelo de *cnn* multi-estágios (WEI et al., 2016). Nele, destaca-se o uso da supervisão intermediária, na qual o *output* é avaliado ao longo da arquitetura.



Fonte: (WEI et al., 2016)

Seguindo a ideia de realizar as supervisões intermediárias dos *keypoints*, Newell et al. propõe o uso de blocos intermediários chamados *hourglass*, que consiste em combinar camadas de *poolings* e *upsamplings*, a fim de capturar a informação de pose sob diferentes escalas (NEWELL; YANG; DENG, 2016). O termo *hourglass* pode ser traduzido para "ampulheta", devido ao formato de sua arquitetura convolucional, caracterizada por transições entre camadas de baixa e alta resolução (Figura 6). Com isso, seus resultados superaram técnicas do estado da arte nos *datasets FLIC* (SAPP; TASKAR, 2013) e *MPII Human Pose* (ANDRILUKA et al., 2014).

Figura 6 – Arquitetura de *CNN* empregando o uso de módulos *encoder-decoder* chamados *hourglasses* (**b.**) (NEWELL; YANG; DENG, 2016).

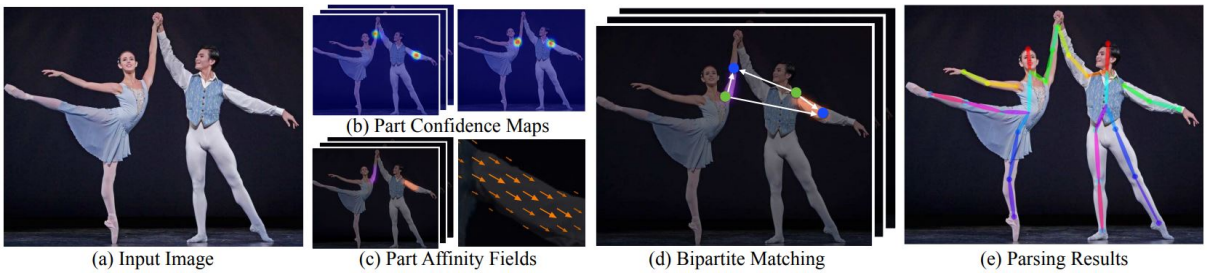


Fonte: (NEWELL; YANG; DENG, 2016)

Apesar do grande avanço dos resultados de estimação de poses humanas 2D, tais métodos partem do cenário cuja entrada seja a imagem de uma pessoa previamente detectada. Com isso, os tempos de execução destes algoritmos são diretamente proporcionais à quantidade de pessoas da imagem original. Considerando esta problemática, o *OpenPose* descreve uma arquitetura convolucional a qual não só estima *heatmaps* de localização dos *keypoints* das pessoas presentes, como também estima *heatmaps* referentes às conexões entre *keypoints*, bem como os seus vetores diretores ao longo das conexões, sendo estas conexões denominadas *Part Affinity Fields* (*PAFs*) (CAO et al., 2017). Tendo estas conexões, também é proposto um pós-processamento capaz de associar diferentes *keypoints*

em poses (Figura 7). Proporcionalmente, a ordem de grandeza do tempo de execução deste pós-processamento é menor que a ordem de grandeza da execução da rede neural convolucional. Com isso, o tempo de processamento do algoritmo para uma imagem é praticamente invariante à quantidade de pessoas presentes na imagem. Além disso, seus resultados quantitativos superaram métodos estado da arte nos *datasets MS COCO* (LIN et al., 2014) e *MPII Human Pose* (ANDRILUKA et al., 2014).

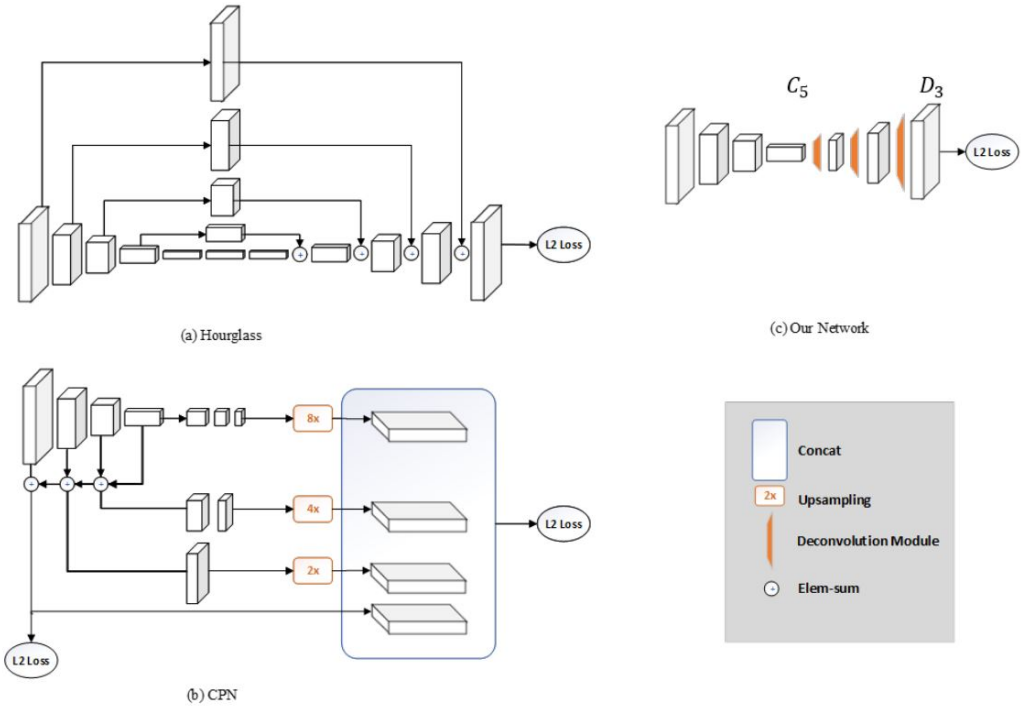
Figura 7 – Modelo de *CNN* no qual os *Part Confidence Maps* e *Part Affinity Fields*, estimados por uma *CNN*, são utilizados para agregar os *keypoints* em suas respectivas poses (CAO et al., 2017).



Fonte: (CAO et al., 2017)

Em contrapartida à complexidade dos trabalhos descritos anteriormente, Xiao et al. propôs um modelo básico de *CNN* com o intuito de entender o quão bom um método simples pode ser (XIAO; WU; WEI, 2018). Para tal, foi adotada a *ResNet* (HE et al., 2016) como *backbone* da rede, seguida de uma sequência de deconvoluções a partir da última camada de convolução da *ResNet* (Figura 8). Na ocasião, atingiu os melhores resultados no *dataset MS COCO* (LIN et al., 2014).

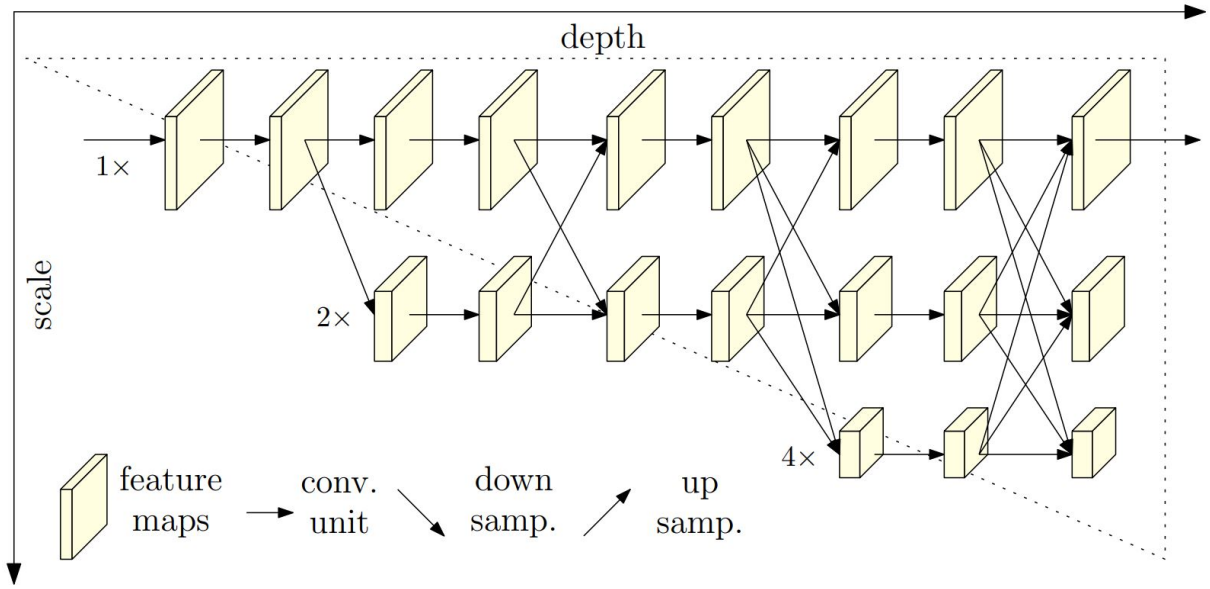
Figura 8 – Modelo de *CNN* cuja menor complexidade, quando comparada a outros modelos, apresenta resultados competitivos na tarefa de *HPE* (XIAO; WU; WEI, 2018).



Fonte: (XIAO; WU; WEI, 2018)

Até o momento, os trabalhos propostos visavam uma representação das *features* com uma etapa intermediária de baixa resolução, fortemente influenciados por Newell et al. (NEWELL; YANG; DENG, 2016). Já Sun et al. propôs um modelo cuja representação de alta resolução é mantida por toda a arquitetura, ainda assim mantendo conexões com *features* de média e baixa resoluções (SUN et al., 2019) (Figura 9). Mesmo não fazendo uso de supervisão intermediária dos *heatmaps*, este trabalho superou os melhores resultados até o seu momento de publicação nos *datasets MS COCO* (LIN et al., 2014) e *MPII Human Pose* (ANDRILUKA et al., 2014).

Figura 9 – Arquitetura de *CNN* cujas *features* de alta resolução são mantidas ao longo da rede, enquanto novas conexões com menores resoluções são aprendidas (SUN et al., 2019).

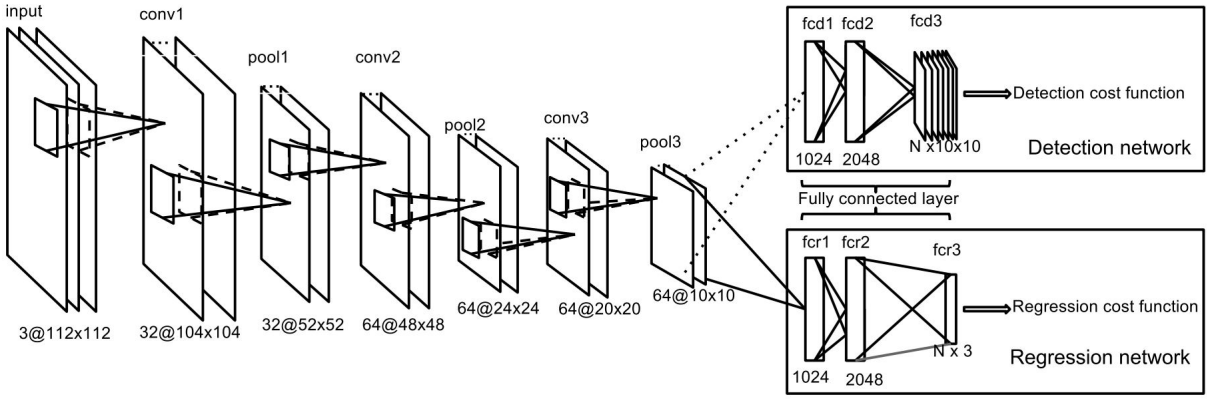


Fonte: (SUN et al., 2019)

2.1.2 Métodos de Estimação de Poses Humanas 3D

Esta tarefa consiste em estimar a localização espacial (x, y, z) das partes do corpo das pessoas em uma imagem. Similar à abordagem adotada por Toshev et al. na tarefa de estimação de poses 2D (TOSHEV; SZEGEDY, 2014), Li et al. propôs um modelo de regressão utilizando redes neurais convolucionais, cuja saída é um conjunto de valores $\{(x_i, y_i, z_i) \mid i \in \{1, 2, \ldots, k\}\}$, no qual (x_i, y_i, z_i) refere-se à localização espacial do *keypoint* i do corpo e k representa o número de *keypoints* que representam o corpo (LI; CHAN, 2014) (Figura 10). Na ocasião, seus resultados superaram os métodos do estado da arte existentes, no *dataset Human3.6M* (IONESCU et al., 2014).

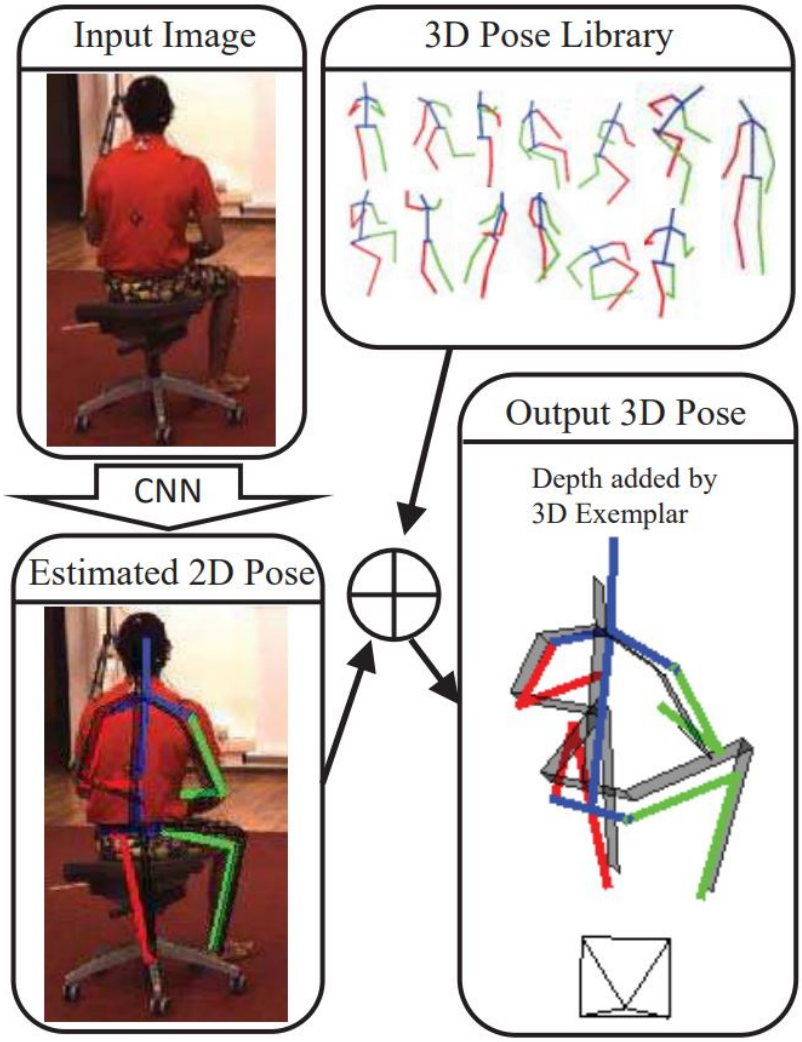
Figura 10 – Modelo de regressão baseado em *CNNs* (LI; CHAN, 2014). Seu módulo de regressão assimila-se ao adotado pelo estimador de pose 2D proposto por Toshev et al. (TOSHEV; SZEGEDY, 2014).



Fonte: (LI; CHAN, 2014)

Ao invés de estimar a pose 3D diretamente, Chen et al. considera os bons resultados de estimadores de poses humanas 2D baseados em *CNNs* para propôr um método que faça a ligação entre poses 2D e 3D, baseado no resultado da projeção da pose 3D a partir de poses de câmeras virtuais (CHEN; RAMANAN, 2017). Com isso, é proposto um algoritmo que primeiramente faz a estimação de poses 2D e, dada uma base de poses 3D pré-gravadas, é escolhida a pose 3D cuja projeção de uma pose de câmera virtual corresponde com mais fidelidade à pose 2D (Figura 11). Utilizando o estimador de pose 2D proposto por Wei et al. (WEI et al., 2016), este método superou resultados do estado da arte no *dataset Human3.6M* (IONESCU et al., 2014).

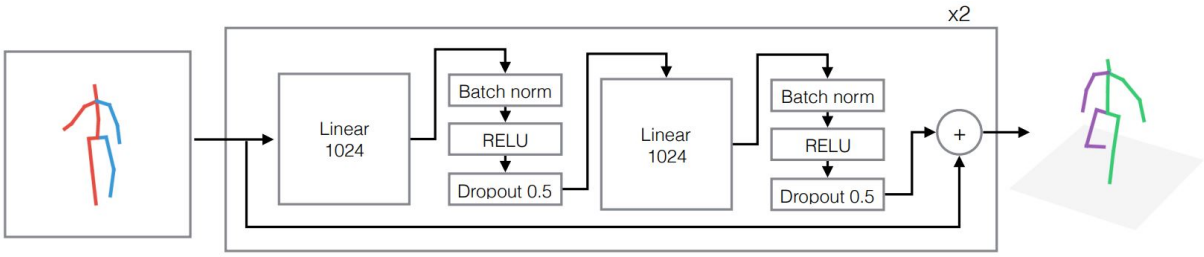
Figura 11 – Abordagem adotada por Chen et al. (CHEN; RAMANAN, 2017). Ao estimar a pose 2D, esta é analisada a um biblioteca de poses 3D pré-gravadas, a fim de descobrir a sua pose 3D correspondente.



Fonte: (CHEN; RAMANAN, 2017)

Apesar dos avanços de qualidade dos métodos de *HPE* em três dimensões, Martinez et al. busca entender se os erros remanescentes dos algoritmos são devidos à qualidade das estimações de pose 2D ou ao algoritmo de mapeamento de poses 2D para 3D (MARTINEZ et al., 2017). Para tal, foi proposta uma arquitetura *MLP* para a estimação de pose 3D a partir de uma pose 2D e, além disso, foram utilizadas poses 2D previamente anotadas (*ground-truth*) como *input*, a fim de abstrair a qualidade das poses 2D, partindo da hipótese de que ela seja de alta qualidade (Figura 12). Com isso, a arquitetura MLP mostrou os melhores resultados no *dataset Human3.6M* (IONESCU et al., 2014). Tal abordagem sugere que grande parte do erro de *HPE* em três dimensões se deve aos resultados intermediários de pose 2D, os quais fazem a análise visual do *input*.

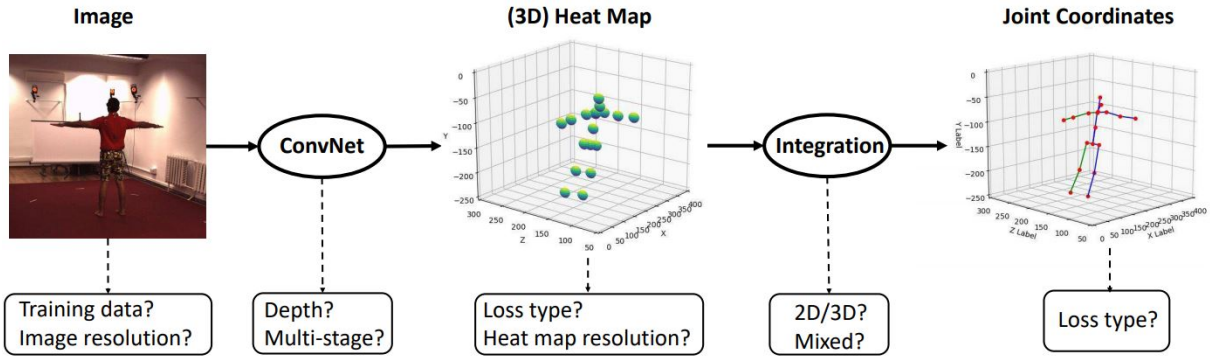
Figura 12 – Módulo de estimação de pose 3D proposto por Martinez et al. (MARTINEZ et al., 2017). A partir de uma pose 2D de entrada, foi utilizada uma arquitetura MLP para a estimação de pose 3D final.



Fonte: (MARTINEZ et al., 2017)

Com o objetivo de utilizar as características da imagem, Sun et al. (SUN et al., 2018) optaram por realizar a estimação de poses 3D diretamente da imagem ao propôr uma rede neural convolucional cuja saída é um conjunto de *heatmaps* tridimensionais $H \in \mathbb{R}^{k \times w \times h \times d}$, onde w, h e d são as dimensões dos *heatmaps* e k o número de *keypoints*, no qual cada posição dos *heatmaps* corresponde à probablidade de conter *keypoints* (Figura 13), sendo assim uma abordagem análoga ao estimador de poses humanas 2D proposto por Tompson et al. (TOMPSON et al., 2015). Seus resultados mostraram um avanço de qualidade no *dataset Human3.6M* (IONESCU et al., 2014).

Figura 13 – Arquitetura proposta por Sun et al. (SUN et al., 2018). Seu diferencial é a utilização de mapas de confiança tridimensionais para localizar cada um dos *keypoints*.

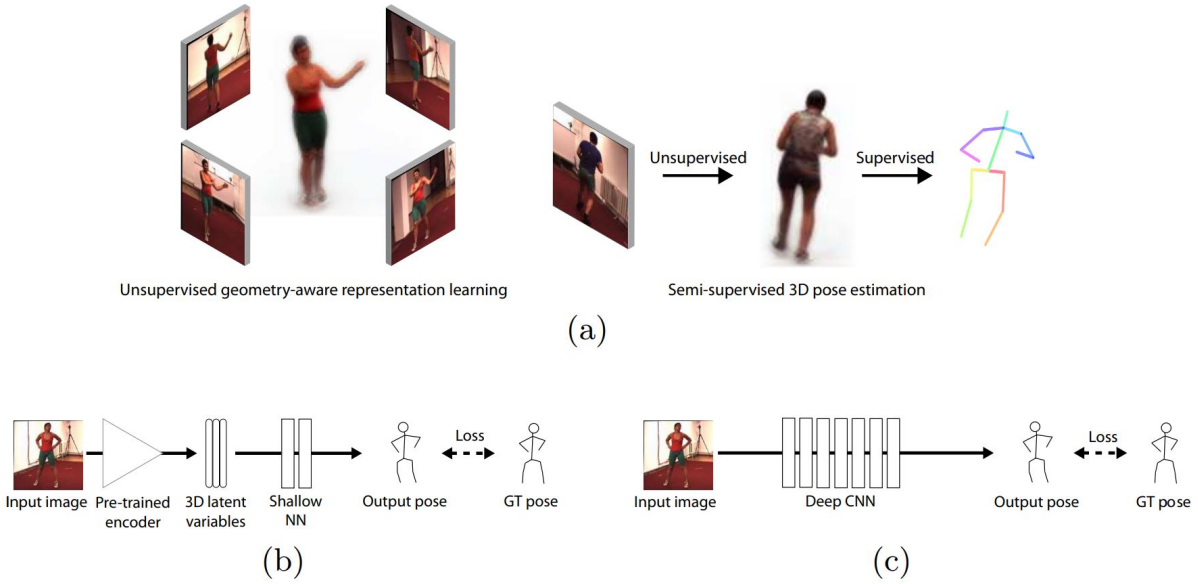


Fonte: (SUN et al., 2018)

Diferentemente dos trabalhos anteriores, Rhodin et al. traz a problemática de treinar modelos de *CNN* para estimação de poses, o que demanda uma quantidade elevada de amostras de dados de pose (RHODIN; SALZMANN; FUA, 2018). Tendo isso em conta, é proposto um modelo não-supervisionado com o objetivo de extrair representações latentes de geometria do corpo humano a partir de múltiplas câmeras. Com isso, é possível treinar de forma supervisionada, com uma quantidade reduzida de amostras, uma rede neural

rasa capaz de receber como *input* as representações latentes do corpo e estimar as poses humanas 3D (Figura 14). Este trabalho, no momento de publicação, superou os resultados do estado da arte no *dataset Human3.6M* (IONESCU et al., 2014).

Figura 14 – Arquitetura desenvolvida por Rhodin et al. (RHODIN; SALZMANN; FUA, 2018). Seu diferencial é o treinamento não-supervisionado de um extrator de *features* tridimensionais latentes, otimizando o processo de treinamento de um modelo de estimação de poses 3D.



Fonte: (RHODIN; SALZMANN; FUA, 2018)

2.2 *DATASETS* PARA ESTIMAÇÃO DE POSES HUMANAS

Nos anos recentes, o sucesso dos métodos de *Human Pose Estimation* baseado em redes neurais convolucionais profundas tem como um de seus principais fatores o uso de *benchmarks* para estimação de poses humanas contendo vastas quantidades de *samples*, os quais buscam cobrir variados desafios de poses. A prática de esportes (JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011; WANG; TRAN; LIAO, 2011), ambientes congestionados e cheios (LIN et al., 2014; GüLER; NEVEROVA; KOKKINOS, 2018; FABBRI et al., 2018), ações cotidianas (ANDRILUKA et al., 2014; LIN et al., 2014; IONESCU et al., 2014; TOME et al., 2019; SIGAL; BALAN; BLACK, 2010; GüLER; NEVEROVA; KOKKINOS, 2018; JHUANG et al., 2013), interações entre pessoas (JHUANG et al., 2013; LIN et al., 2014; SAPP; TASKAR, 2013; TOMPSON et al., 2014; JOO et al., 2015; IQBAL; MILAN; GALL, 2017) ou com objetos (ANDRILUKA et al., 2014; LIN et al., 2014; TOME et al., 2019; IQBAL; MILAN; GALL, 2017) são alguns dos contextos de *HPE* contemplados pelos *datasets* existentes na literatura. Neste trabalho, é apresentada uma classificação destes *datasets* baseada no método de anotação empregado em suas concepções, a fim de compreender os desafios referentes às suas criações.

2.2.1 *Datasets* Manualmente Anotados

Os *datasets* manualmente anotados (*MLDs*, *Manually Labeled Datasets*) são as bases de dados de poses humanas cuja anotação dos *samples* é realizada por seres humanos diretamente no conteúdo visual (*workers*). Em geral, são utilizados serviços de hospedagem de vídeos e imagens para a obtenção de conteúdos visuais que contemplem contextos de *HPE*, tais como o *Flickr* (LIN et al., 2014; JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011; BOURDEV et al., 2010) ou o *YouTube* (ANDRILUKA et al., 2014). Uma pose humana é representada por um conjunto de localidades características do corpo (*keypoints*) e, para cada imagem, o *worker* deve indicar a localização de cada *keypoint* do corpo. Em geral, os *datasets* manualmente anotados consistem de *samples* em duas dimensões, representando a localização dos *keypoints* em *pixels* (ANDRILUKA et al., 2014; LIN et al., 2014; JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011; SAPP; TASKAR, 2013; TOMPSON et al., 2014; GONG et al., 2017; IQBAL; MILAN; GALL, 2017; WANG; TRAN; LIAO, 2011; TRAN; FORSYTH, 2010; JHUANG et al., 2013), com exceção de dois *datasets* com *samples* em 3D: o *dataset DensePose-COCO* (GüLER; NEVEROVA; KOKKINOS, 2018), o qual realiza a correspondência entre anotações manuais em coordenadas UV e o modelo de corpo virtual *Skinned Multi-Person Linear* (*SMPL*) (LOPER et al., 2015), e o *dataset H3D* (BOURDEV; MALIK, 2009), o qual, além de prover a localização 3D dos *keypoints*, também provê uma *interface* para a anotação manual da ordem de distância de cada um dos *keypoints* para a câmera (TAYLOR, 2000).

Em geral, existem duas possibilidades de escolha da origem dos *workers*: indivíduos internos à equipe do projeto de concepção do *dataset* (comumente chamados de *in-house workers*), ou então provedores terceirizados de serviços de *crowdsourcing*, por exemplo o *Amazon Mechanical Turk* (*AMT*), *Clickworker* ou *Lionbridge*. Apesar dos *workers* realizarem a anotação manual ao analisar os *samples*, ainda existe a possibilidade de ocorrência de anotações incorretas. Isto pode acontecer por diversos motivos, tais como a localização grosseira dos *keypoints*, a distinção entre diferentes *workers* a respeito do entendimento da localização precisa de um determinado tipo de *keypoint* e o decremento da qualidade do trabalho dos *workers* ao longo do processo, uma vez que esta tarefa apresenta-se extremamente repetitiva e consequentemente cansativa e propensa a erros humanos (ANDRILUKA et al., 2014; LIN et al., 2014). Existem alguns procedimentos para garantir uma maior qualidade nas anotações, tais como a aplicação de um teste de qualificação de *workers* (LIN et al., 2014), a anotação de um mesmo *sample* por mais de um *worker* (LIN; O'BRIEN, 2019; SAPP; TASKAR, 2013) e avaliações quantitativas de subconjuntos dos *samples* anotados. A Tabela 1 apresenta um panorama geral dos principais *MLDs* da literatura. Pode-se observar que a maioria dos *MLDs* possuem bancos externos de imagens e vídeos como fonte dos conteúdos visuais. Além disso, a maioria dos *MLDs* buscam contemplar o contexto geral de poses humanas, bem como contextos de esportes, os quais também trazem variadas possibilidades de poses.

Tabela 1 – Visão geral dos principais *MLDs*. Para cada *dataset* foram coletadas as seguintes informações (caso disponíveis): o número de indivíduos presentes por *sample* (**# Por Imagem**), a dimensionalidade das poses (**Dim.**), o número de *samples* (**# *Samples***), a origem dos conteúdos visuais dos *samples* (**Fonte dos Dados**), o contexto das poses presentes no *dataset* (**Contexto**) e a origem dos *workers* envolvidos no processo de anotação (**Método de *Crowdsourcing***).

Dataset	**# Por Imagem**	**Dim.**	**# *Samples***	**Fonte dos Dados**	**Contexto**	**Método de *Crowdsourcing***
Iterative Image Parsing (RAMANAN, 2007)	1	2D	305	-	Esportes	*In-house*
H3D (BOURDEV; MALIK, 2009)	1	3D	2,000	*Flickr*	Diverso	*AMT*
UIUC People (TRAN; FORSYTH, 2010)	1	2D	593	-	Esportes	*In-house*
LSP (JOHNSON; EVERINGHAM, 2010)	1	2D	2,000	*Flickr*	Esportes	*AMT*
LSP Extended (JOHNSON; EVERINGHAM, 2011)	1	2D	10,000	*Flickr*	Esportes	*AMT*
Sport Image (WANG; TRAN; LIAO, 2011)	1	2D	1,299	-	Esportes	*In-house*
MPII Cooking (ROHRBACH et al., 2012)	1	2D	1,277	Vídeos (autorais)	Cozinha	*In-house*
FLIC (SAPP; TASKAR, 2013)	n	2D	± 20,000	Filmes (*Hollywood*)	Filmes	*AMT*
J-HMDB (JHUANG et al., 2013)	1	2D	31,838	Vídeos (*Internet*)	Diverso	*AMT*
MPII Human Pose (ANDRILUKA et al., 2014)	n	2D	+ 40,000	Vídeos (*YouTube*)	Diverso	*AMT* e *in-house*
MS COCO (LIN et al., 2014)	n	2D	± 250,000	*Flickr*	Diverso	*AMT*
FLIC-plus (TOMPSON et al., 2014)	n	2D	17,380	Filmes (*Hollywood*)	Filmes	*AMT*
LIP (GONG et al., 2017)	1	2D	7,559	(LIN et al., 2014)	Diverso	-
PoseTrack (IQBAL; MILAN; GALL, 2017)	n	2D	16,000	Vídeos (*Internet*)	Diverso	*In-house*
DensePose-COCO (GüLER; NEVEROVA; KOKKINOS, 2018)	n	3D	50,000	(LIN et al., 2014)	Diverso	-

2.2.2 *Datasets* Automaticamente Gerados

Similar aos *MLDs*, os *datasets* automaticamente gerados (*AGDs*, *Automatically Generated Datasets*) provêm anotações para a tarefa de *HPE* a partir de conteúdos visuais, porém com a diferença de que não precisam de esforços humanos diretos para a geração dos *samples*. Apesar de alguns *AGDs* objetivarem cobrir os desafios de *HPE* em geral, estes não contemplam inteiramente o espaço de poses humanas possíveis.

Comumente, *AGDs* são concebidos a partir de técnicas e ferramentas de captura de movimento (*MoCap*, *Motion Capture*), as quais buscam realizar a estimação de poses humanas a partir da organização de câmeras e sensores em um ambiente controlado (MEHTA et al., 2018; MEHTA et al., 2017; IONESCU et al., 2014; RHODIN et al., 2016; JOO et al., 2015; SIGAL; BALAN; BLACK, 2010; TRUMBLE et al., 2017; MARCARD; PONS-MOLL; ROSENHAHN, 2016). Sigal et al. introduziram um dos primeiros *datasets* para *HPE* utilizando marcadores reflexivos anexados ao corpo e um conjunto de câmeras, contemplando assim uma variedade de ações humanas, tais como andar, trotar, lutar boxe, arremessar e capturar bolas (SIGAL; BALAN; BLACK, 2010). Ionescu et al. aumentou a quantidade de poses capturadas para aproximadamente 3,6 milhões de *samples*, incluindo atividades cotidianas tais como alimentar-se, aguardar, tirar fotografias e cumprimentar (IONESCU et al., 2014). Trumble et al. utilizou unidades de medição inercial (*IMUs*, *Inertial Measurement Units*) ao longo do corpo a fim de estimar poses humanas em três dimensões de forma precisa (TRUMBLE et al., 2017). Apesar de existerem ferramentas de *MoCap* as quais usufruem ao máximo de marcadores externos para localizar as partes do corpo (MARCARD et al., 2018; TRUMBLE et al., 2017; MARCARD; PONS-MOLL; ROSENHAHN, 2016; IONESCU et al., 2014; SIGAL; BALAN; BLACK, 2010), o progresso dos algoritmos de *HPE* baseados em imagens RGB tornou possível o advento de tecnologias de *MoCap* sem o uso de marcadores (*markerless*) (MEHTA et al., 2017; RHODIN et al., 2016; JOO et al., 2015; SIGAL; BALAN; BLACK, 2010), tornando mais naturais os movimentos capturados. (MEHTA et al., 2017) criou um *dataset* capturando uma variedade de poses humanas utilizando o *The Captury*[1], uma ferramenta *markerless* para *MoCap*. Joo et al. produziu um *benchmark* contemplando interações entre pessoas ao utilizar um massivo sistema de múltiplos pontos de vista (*multi-view*) com 480 câmeras (JOO et al., 2015). Rhodin et al. explorou a criação de um *dataset* para a captura de movimentos de forma egocêntrica, de forma que a câmera possui a sua posição e orientação relativas ao movimento do usuário rastreado (RHODIN et al., 2016). É importante mencionar que, para conceber um sistema de *MoCap*, é necessário o conhecimento prévio de fundamentos de visão computacional e rastreamento, especificamente no âmbito de rastreamento de poses humanas e calibração de câmeras (Apêndice C).

Uma outra abordagem para a geração automática de *datasets* para *HPE* é a obtenção de vídeos e imagens a partir de provedores externos (similar aos *MLDs*), os quais tam-

[1] https://thecaptury.com/

bém contemplam desafios de *HPE* e geram *samples* a partir da combinação de algoritmos para *HPE* (CHARLES et al., 2016; CHARLES et al., 2014; PFISTER et al., 2014; CHARLES et al., 2013; SHEN et al., 2014; SCHULDT; LAPTEV; CAPUTO, 2004). Schuldt et al. criou uma base de dados para o reconhecimento de ações humanas a partir de imagens ao utilizar uma máquina de vetores de suporte (*SVM*, *Support Vector Machine*) (SCHULDT; LAPTEV; CAPUTO, 2004). Charles et al. utilizou um algoritmo de correspondência entre estimações de poses humanas realizadas entre *frames* temporalmente distantes, provendo assim melhorias substanciais aos *samples* finais (CHARLES et al., 2016). A Tabela 2 apresenta um panorama geral dos principais *AGDs* da literatura. Observa-se que a maioria dos *AGDs* provê anotações de poses humanas em três dimensões, devido à possibilidade de estimá-las em uma configuração de maior controle. Além disso, observa-se que a fonte dos conteúdos visuais é majoritariamente autoral, partindo do objetivo de gerar novos *datasets* focados em contextos de poses humanas específicas.

Tabela 2 – Visão geral dos principais *AGDs* (incluindo o *dataset* concebido ao utilizar o *HuTrain*). Para cada *dataset* foram coletadas as seguintes informações (caso disponíveis): o número de indivíduos presentes por *sample* (**# Por Imagem**), a dimensionalidade das poses (**Dim.**), o número de *samples* (**# *Samples***), o número de câmeras (**# Câmeras**), a presença de marcadores aos corpos dos usuários capturados (***Marker-Based?***), a origem dos conteúdos visuais dos *samples* (**Fonte dos Dados**) e o contexto das poses presentes no *dataset* (**Contexto**).

Dataset	**# Por Imagem**	**Dim.**	**# *Samples***	**# Câmeras**	***Marker-based?***	**Fonte dos Dados**	**Contexto**
KTH (SCHULDT; LAPTEV; CAPUTO, 2004)	1	2D	2,391	1	-	Vídeos (autorais)	Ações
HumanEva (SIGAL; BALAN; BLACK, 2010)	1	3D	± 40,000	7	✓	*MoCap*	Ações
Campus (BERCLAZ et al., 2011)	n	3D	2,000	3	-	Vídeos (autorais)	Caminhada
Short BBC Pose (CHARLES et al., 2013)	1	2D	± 1,000	1	-	Vídeos (transmissão de TV)	Linguagem de sinais
FYDP / UYDP (SHEN et al., 2014)	1	2D	± 2,000	1	-	Vídeos (*YouTube*)	Dança
BBC Pose (CHARLES et al., 2014)	1	2D	± 2,000	1	-	Vídeos (transmissão de TV)	Linguagem de sinais
Extended BBC Pose (PFISTER et al., 2014)	1	2D	± 7,000,000	1	-	Vídeos (transmissão de TV)	Linguagem de sinais
Shelf (BELAGIANNIS et al., 2014)	n	3D	-	5	-	Vídeos (autorais)	Interações sociais
Human3.6M (IONESCU et al., 2014)	1	3D	3,600,000	4	✓	*MoCap*	Ações
CMU MoCap (JOO et al., 2015)	n	3D	1,500,000	480	-	*MoCap*	Interações sociais
TNT15 (MARCARD; PONS-MOLL; ROSENHAHN, 2016)	1	3D	± 13,000	8	✓	*MoCap*	Ações
MPI-MARCOnI (ELHAYEK et al., 2016)	n	3D	6,190	2 to 4	-	Vídeos (autorais)	Diverso
EgoCap (RHODIN et al., 2016)	1	3D	100,000	9	-	*MoCap*	Interações com objetos
YouTube Pose (CHARLES et al., 2016)	1	2D	± 5,000	1	-	Vídeos (*YouTube*)	Diverso
MPI-INF-3DHP (MEHTA et al., 2017)	1	3D	1,300,000	5	-	*MoCap*	Diverso
TotalCapture (TRUMBLE et al., 2017)	1	3D	± 1,892,176	8	✓	*MoCap*	Ações
MuCo-3DHP / MuPots-3D (MEHTA et al., 2018)	n	3D	+ 8,000	5	-	(MEHTA et al., 2017)	Diverso
3DPW (MARCARD et al., 2018)	n	3D	51,000	1	✓	Vídeos (autorais)	Diverso
Proposto	**1**	**3D**	**32,488**	**8**	**-**	**Vídeos (autorais)**	**Ações**

2.2.3 *Datasets* Sinteticamente Gerados

Criar *datasets* apresenta-se como uma tarefa árdua. Contudo, para diversas tarefas no contexto de visão computacional, o treinamento de redes neurais convolucionais profundas a partir de dados sintéticos tem mostrado resultados significantemente positivos, com a vantagem de um menor esforço para a produção dos *samples* (CUNHA, 2019; TEKIN; BOGO; POLLEFEYS, 2019; MCCORMAC et al., 2016; GE et al., 2019). Os *datasets* sinteticamente gerados (*SGDs*, *Synthetically Generated Datasets*) são os *benchmarks* os quais buscam cobrir desafios de *HPE* a partir de *samples* virtuais que contenham fotorrealismo suficiente para serem utilizados da mesma maneira que *samples* reais são utilizados. Uma de suas principais vantagens é a capacidade de controle do espaço virtual, a fim de variar ao máximo o conteúdo visual dos *samples* produzidos. É importante destacar que *HPE* enquadra-se como uma tarefa de detecção não-rígida, ou seja, as distâncias entre os diferentes *keypoints* do corpo podem variar livremente. Com isso, torna-se desafiador descrever o espaço de poses humanas sinteticamente, diferentemente de tarefas envolvendo instâncias rígidas, por exemplo a tarefa de deteção de objetos em seis graus de liberdade (*6-DoF*, *Six Degrees of Freedom*) (CUNHA, 2019). Portanto, os *SGDs* para a tarefa de *HPE* adotam a utilização de poses pré-gravadas a partir de fontes externas (usualmente *MLDs* e *AGDs*) e focam em variar hiperparâmetros, tais como os formatos dos corpos, cenários, texturas e iluminação.

Muitos dos *SGDs* adotaram o *dataset CMU MoCap* (JOO et al., 2015) como sendo a fonte de poses pré-gravadas (XU et al., 2019; CHEN et al., 2016; VAROL et al., 2017), uma vez que inclui uma diversidade de ações. Além disso, é frequentemente empregado o modelo virtual de corpo *SMPL*, a fim de que representações volumétricas de pessoas possam ser posicionadas em um ambiente virtual (XU et al., 2019; VAROL et al., 2017; LASSNER et al., 2017). A fim de prover fotorrealismo aos *samples*, alguns *SGDs* utilizam imagens reais a fim de texturizar virtualmente as roupas vestidas pelas pessoas presentes (CHEN et al., 2016; PISHCHULIN et al., 2012). Para simular fundos fotorrealísticos, alguns *benchmarks* utilizam cenários oriundos de imagens reais (CHEN et al., 2016; PISHCHULIN et al., 2012; XU et al., 2019; TOME et al., 2019) ou utilizam conteúdos virtuais (FABBRI et al., 2018).

Pishchulin et al. (PISHCHULIN et al., 2012) focou em modificar o já existente *dataset LSP* (JOHNSON; EVERINGHAM, 2010) ao readaptar as poses provenientes de (HASLER et al., 2009). Xu et al. propôs um *dataset* para *HPE* a partir de câmeras egocêntricas no qual um chapéu, vestido pelo usuário rastreado, posiciona virtualmente a câmera (XU et al., 2019). Fabbri et al. optou por conceber um *dataset* totalmente virtual ao utilizar um jogo de *video game* fotorrealista contendo diferentes poses humanas em ambientes urbanos congestionados (FABBRI et al., 2018). A Tabela 3 apresenta um panorama geral dos principais *SGDs* da literatura.

Tabela 3 – Visão geral dos principais *SGDs*. Para cada *dataset* foram coletadas as seguintes informações (caso disponíveis): o número de indivíduos presentes por *sample* (**# Por Imagem**), o número de *samples* (**# *Samples***), a origem das poses humanas utilizadas (**Fonte das Poses**), o modelo de corpo virtual aplicado às poses humanas (**Modelo de Corpo**), a origem das texturas de corpos e roupas utilizadas (**Texturas**), o método utilizado para a geração de *samples* fotorrealistas (**Fotorrealismo**) e o contexto das poses presentes no *dataset* (**Contexto**).

Dataset	**# Por Imagem**	**# *Samples***	**Fonte das Poses**	**Modelo de Corpo**	**Texturas**	**Fotorrealismo**	**Contexto**
Multi-scale LSP (PISHCHULIN et al., 2012)	n	2,551	(HASLER et al., 2009)	(HASLER et al., 2009)	(JOHNSON; EVERINGHAM, 2010)	(JOHNSON; EVERINGHAM, 2010)	Esportes
Human3D+ (CHEN et al., 2016)	1	5,099,405	(JOO et al., 2015) e rede bayesiana	(ANGUELOV et al., 2005)	Imagens reais	Imagens reais	Ações
SURREAL (VAROL et al., 2017)	1	6,500,000	(JOO et al., 2015) (JOHNSON; EVERINGHAM, 2010)	(LOPER et al., 2015) (ROBINETTE et al., 2002)	(LOPER et al., 2015)	(GREEN, 2003; YU et al., 2015)	Ações
UP-3D (LASSNER et al., 2017)	1	27,652	(JOHNSON; EVERINGHAM, 2011) (ANDRILUKA et al., 2014)	(LOPER et al., 2015)	-	-	Esportes, diverso
JTA (FABBRI et al., 2018)	n	± 10,000,000	(NORTH et al., 2013)	(NORTH et al., 2013)	(NORTH et al., 2013)	(NORTH et al., 2013)	Multidão
Mo^{2}Cap^2 (XU et al., 2019)	1	530,000	(JOO et al., 2015)	(LOPER et al., 2015)	(VAROL et al., 2017)	(GREEN, 2003)	Ações
xR-EgoPose (TOME et al., 2019)	1	383,000	-	(INC., 2008)	Imagens (autoral)	(XU et al., 2019; GROUP, 1997)	Ações

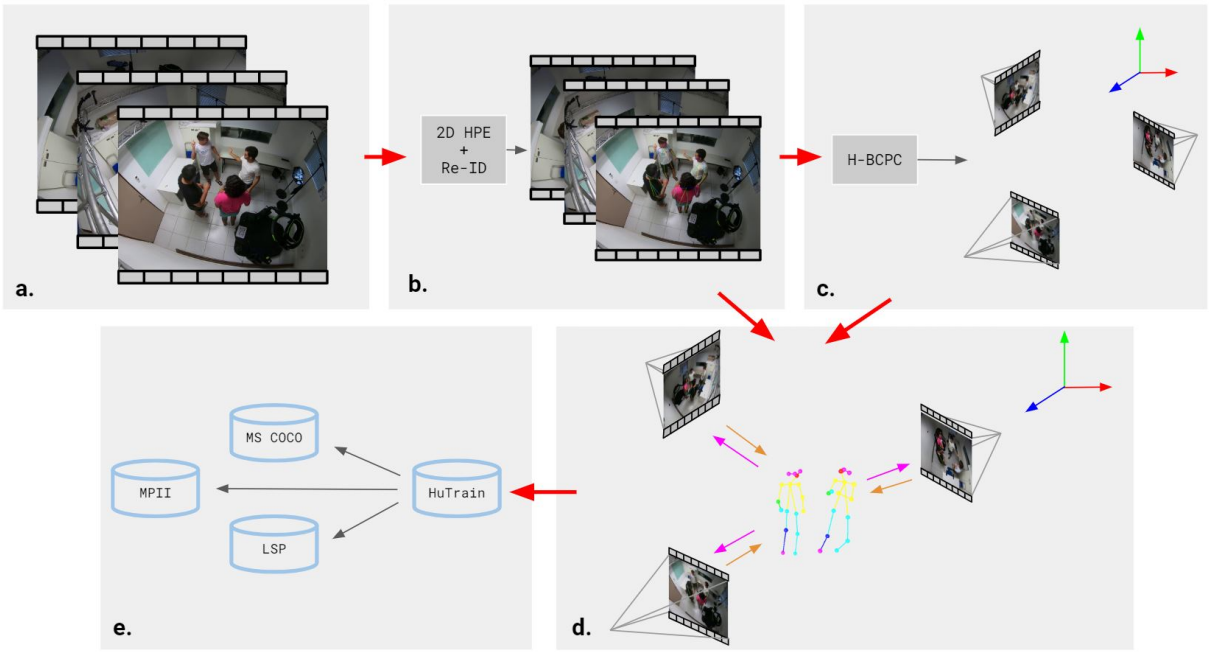
3 FRAMEWORK DE GERAÇÃO DE DATASETS PARA ESTIMAÇÕES DE POSES HUMANAS

Ao analisar os *datasets* para estimações de poses humanas existentes na literatura, observa-se que as suas concepções envolvem custos financeiros, custos de tempo e know-how técnico. Esta dissertação visa trazer alternativas a partir da proposição de um *pipeline*:

- Que faça uso de recursos materiais de forma acessível;
- Que demande uma quantidade de tempo reduzida;
- Que possa ser usufruída por usuários não-especialistas.

Neste trabalho, foi desenvolvido o *HuTrain* como um *framework* para facilitar e agilizar a geração de *datasets* para *HPE* a partir de vídeos. A Figura 15 resume o *pipeline* do *framework*. É recebido, como *input*, um conjunto de vídeos *RGB* sincronizados $V = \{V_1, V_2, \ldots, V_C\}$ de dimensões $w_c \times h_c$ (em *pixels*), contendo F *frames* para cada câmera estacionária $c \in \{1 \ldots C\}$, portanto $V_c \in \mathbb{R}^{F \times w_c \times h_c \times 3}$ (Figura 15a). Em resumo, o *framework* primeiramente realiza estimações de poses humanas 2D em todos os *frames* pertencentes a $\mathbf{V}$ (Figura 15b). Após isso, é calculada as poses (parâmetros extrínsecos) das câmeras utilizadas (Figura 15c), seguido de um algoritmo de *HPE* em três dimensões considerando múltiplas vistas (*views*) (Figura 15d). Por fim, as poses geradas são convertidas para formatos de *datasets* de *HPE* existentes na academia (Figura 15e). Vale ressaltar que a qualidade das poses humanas finais estimadas depende da qualidade das estimações de poses humanas 2D iniciais.

Figura 15 – *Pipeline* do *framework*. **a.** Vídeos *RGB* de entrada; **b.** Estimação inicial de poses humanas 2D seguida da reidentificação das pessoas em múltiplas vistas; **c.** Estimação dos parâmetros extrínsecos das câmeras baseada em humanos; **d.** Triangulação de poses humanas (setas laranjas) e estimações refinadas de poses humanas 2D (setas rosas); **e.** Conversão para formatos de *datasets* conhecidos.



3.1 MÓDULO DE ESTIMAÇÃO INICIAL DE POSES HUMANAS 2D

Foi adotado o *OpenPose* (CAO et al., 2017) para a tarefa de estimação de poses 2D de múltiplas pessoas (*multi-person*) a partir de uma única imagem *RGB*. Vale ressaltar que, devido à natureza modularizada do *framework* proposto, outros algoritmos de estimação de poses humanas 2D podem ser utilizados. Sua arquitetura convolucional procura estimar mapas de confiança de localização de *keypoints* (*Part Confidence Maps*) além de mapas de confiança de localização de conexões entre *keypoints* (*Part Affinity Fields* ou *PAFs*), os quais podem ser combinados por um algoritmo guloso a fim de inferir as localizações dos *keypoints* 2D de todos os indivíduos da imagem (Figura 7). Este algoritmo é aplicado para todos os *frames* de V como sendo as estimações de poses humanas iniciais.

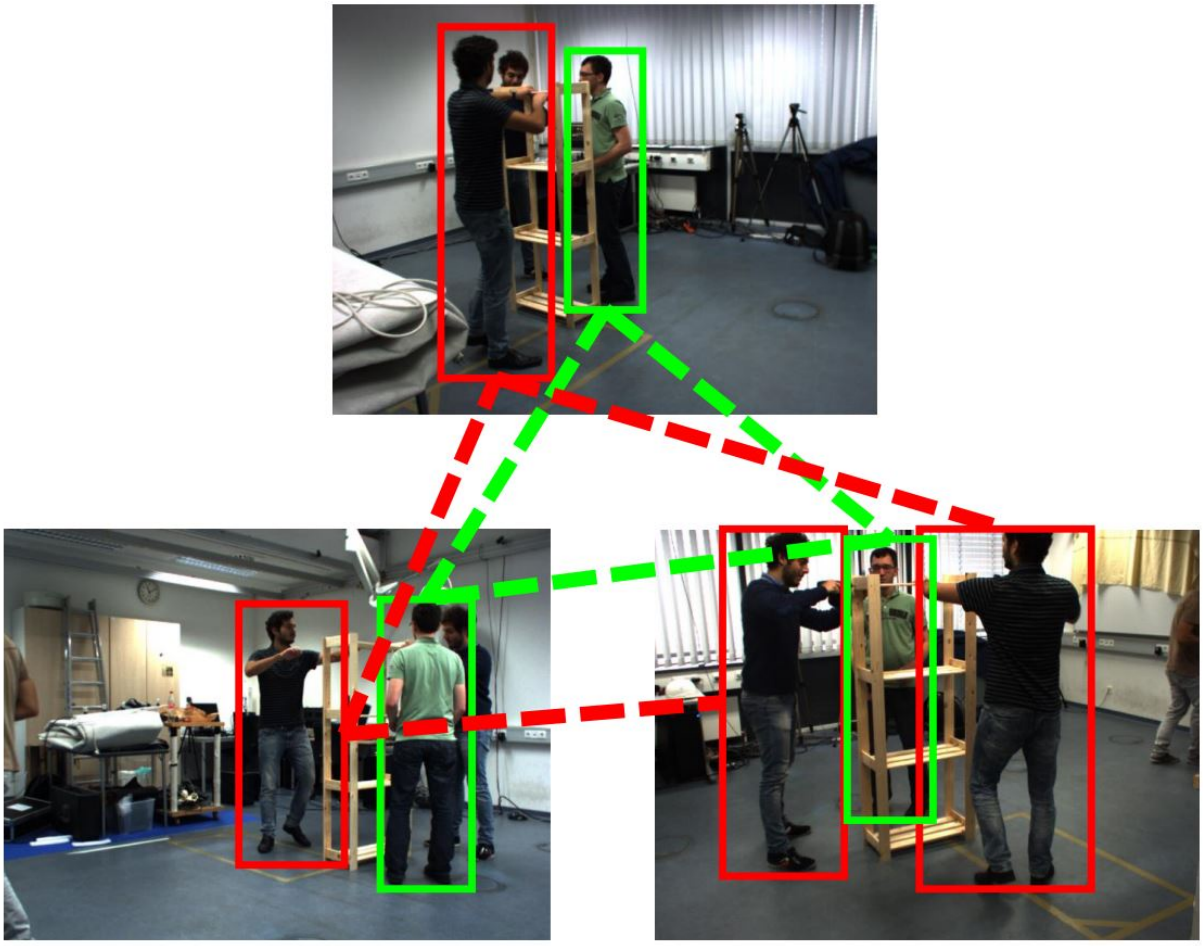
Contudo, os vídeos de entrada podem conter um número indefinido de pessoas. Portanto, a estimação de poses humanas 2D para cada câmera individualmente não é o suficiente, uma vez que ainda existe a necessidade de corresponder as instâncias da mesma pessoa em diferentes câmeras. Para tal, foi utilizado o algoritmo de re-identificação (*re-ID*, *Re-Identification*) de indivíduos baseado em aparência entre diferentes câmeras (*cross-view*), proposto pelo trabalho do *MVPose* (DONG et al., 2019) para cada *frame* $f \in \{1 \dots F\}$. Considerando isso, os *inputs* do algoritmo *re-ID* são um conjunto de imagens $V^f = \{V_1^f, V_2^f, \dots, V_C^f\}$, além de um conjunto de poses humanas 2D detectadas

Figura 16 – Exemplos de estimação de poses humanas 2D oriundas do algoritmo *OpenPose* (CAO et al., 2017).

(junto com os seus respectivos valores de confiança $\{q \in \mathbb{R} \mid 0 \leq q \leq 1\}$) definido por $O^f = \{O_1^f, O_2^f, \ldots, O_C^f\}$, no qual $O_c^f \in \mathbb{R}^{P_c^f \times K \times 3}$, P_c^f representa o número de poses 2D detectadas (o qual pode variar de acordo com o ponto de vista) e K representa o número de tipos de *keypoints* (ombro esquerdo, cotovelo direito, etc.). A abordagem de *re-ID* é realizada da seguinte maneira:

- Para cada pessoa $p \in \{1 \ldots P_c^f\}$ presente na imagem V_c^f, são gerados os *bounding-boxes* $BB_{c,p}^f$ a partir das localizações dos *keypoints* em $O_{c,p}^f$;
- Para cada uma das imagens das pessoas detectadas $BB_{c,p}^f$, é utilizado o extrator de características (*features*) baseado em imagem proposto pelo *CamStyle* (ZHONG et al., 2018), o qual busca gerar vetores de *features* que sejam invariantes à pose da câmera. Dessa maneira, estima-se a afinidade de aparência entre todas as imagens dos diferentes indivíduos a partir do cálculo de distância euclidiana entre os descritores e mapeia-se a valores variando entre 0 e 1 (DONG et al., 2019);
- É aplicado o algoritmo de correspondência (*matching*) cíclica entre descritores de aparência invariantes à pose de câmera, proposto por Dong et al. (DONG et al., 2019), cujo *output* é uma matriz binária de correspondências entre os vetores de características de todas as imagens de todos os indivíduos $M \in \{0,1\}^{S \times \sum_{c=1}^{C} P_c^f}$,

Figura 17 – Exemplo de re-identificação de pessoas em múltiplos pontos de vista.



Fonte: (DONG et al., 2019).

na qual S representa o real número de pessoas no ambiente. Vale ressaltar que S é implicitamente inferido pelo algoritmo;

- A partir de M, é gerada uma lista de correspondências da re-identificação de indivíduos nos múltiplos pontos de vista $R = \{R_1, R_2, \ldots, R_L\}$, na qual $R_l = \langle f, c_i, c_j, p, g \rangle$ possui o seguinte significado: no *frame* f, a pessoa p nas câmeras c_i e c_j possui uma confiança $\{g \in \mathbb{R} \mid 0 \leq g \leq 1\}$ de representar a mesma pessoa real. O valor de confiança g deriva da estimação de afinidade de aparência, mencionado anteriormente. A informação de re-identificação R é utilizada posteriormente na etapa de calibração de câmeras.

- A partir de O e M, é produzido um conjunto de ocorrências de *keypoints* em todos os vídeos $O' = \{O'_1, O'_2, \ldots, O'_{B'}\}$, no qual $O'_{b'} = \langle f, c, p, k, x, y, q \rangle$ possui o seguinte significado: no *frame* f do vídeo V_c, o indivíduo p possui um *keypoint* do tipo k, localizado no *pixel* $\langle x, y \rangle$, com um valor de confiança de q.

3.2 MÓDULO DE CALIBRAÇÃO DOS PARÂMETROS EXTRÍNSECOS DAS CÂMERAS BASEADA EM HUMANOS

A fim de cruzar informações de *HPE* 2D a partir de instâncias de câmeras independentes e gerar poses humanas 3D baseadas em múltiplas vistas, é necessario que as poses das câmeras sejam representadas no mesmo espaço de coordenadas. Técnicas de *Structure from Motion* (*SfM*) buscam recuperar a informação de estrutura tridimensional e a movimentação da câmera a partir de imagens 2D (ULLMAN, 1979; KOENDERINK; DOORN, 1991; SCHONBERGER; FRAHM, 2016). Estes métodos utilizam detectores de *features* 2D (LOWE, 1999; BAY; TUYTELAARS; GOOL, 2006; RUBLEE et al., 2011; ALCANTARILLA; BARTOLI; DAVISON, 2012; ALCANTARILLA; SOLUTIONS, 2011) a fim de encontrar correspondências 2D representativas entre as imagens e reconstruir estas correspondências tridimensionalmente, bem como as poses das câmeras em cada imagem. Porém, os algoritmos de *matching* de *features* baseados puramente no conteúdo da imagem não funcionam como desejado em cenários nos quais a distância entre as imagens é suficientemente elevada.

É proposto um algoritmo de calibração de poses de câmeras baseado em humanos (*H-BCPC*, *Human-Based Camera Poses Calibration*), visando o desafio de encontrar a relação entre os parâmetros extrínsecos dos pontos de vista utilizados, em um cenário de captura de movimento com um número de câmeras reduzido. Sua estratégia principal empregada é a utilização das estimações de poses humanas 2D provenientes da etapa anterior (O') como sendo as *features* 2D extraídas. Apesar de O', diferentemente de extratores de *features* convencionais, não prover nenhuma informação de descrição, o *matching* das *features* é resolvido implicitamente: dois *keypoints* de pontos de vista diferentes, do mesmo tipo (por exemplo, ombro esquerdo) e pertencentes à mesma pessoa em um mesmo *timestamp* geram o que é considerado um *match*. O algoritmo de *H-BCPC* funciona da seguinte maneira:

- Primeiramente, todas as ocorrências de *keypoints* em O' são agrupadas considerando três parâmetros: o *frame* (f), o índice da pessoa (p) e o tipo do *keypoint* (k). Com isso, é obtido um novo conjunto de *keypoints* $O'' = \{O''_1, O''_2, \ldots, O''_{B''}\}$, no qual $O''_{b''} = \langle f, p, k, oq \rangle$ e $oq \in \mathbb{R}^{C\times 3}$ representam todas as ocorrências de um determinado *keypoint* em todas as câmeras. Além de ter a localização e valor de confiança dos *keypoints* para cada câmera, os valores de *re-ID* dos indivíduos sob diferentes câmeras R são agrupados considerando dois parâmetros: o *frame* (f) e o índice da pessoa (p). Com isso, é obtido um novo conjunto de valores de *re-ID* $R' = \{R'_1, R'_2, \ldots, R'_{L'}\}$, no qual $R'_{l'} = \langle f, p, rg \rangle$ e $rg \in \mathbb{R}^{\binom{C}{2}}$ descrevem os valores de *re-ID* de todas as combinações de pares de imagens da pessoa p no *frame* f;

- São ordenados todos os $O''_{b''} \in O''$ de acordo com os valores de confiança dos *keypoints* $oq^q \in \mathbb{R}^C$ e seus valores de *re-ID* referentes às pessoas a quem pertencem, e a ordenação é feita em função do valor mínimo de confiança, tanto a partir dos *keypoints* quanto dos valores de *re-ID*, definido por θ (Equação 3.1). Tal métrica

valoriza a presença de *keypoints* com altos valores de confiança, bem como uma maior probabilidade de correspondência entre suas ocorrências nos diferentes pontos de vista. Com isso, são escolhidos os 100 melhores conjuntos de *keypoints* 2D $H^{2D} \in \mathbb{R}^{C\times 100\times 3}$, em função de θ;

$$\theta = min(min(oq^q), min(rg)) \tag{3.1}$$

- É estimado um conjunto inicial de *keypoints* em espaço de coordenadas 3D, o qual será utilizado para calcular os parâmetros extrínsecos das câmeras. Para cada conjunto de *keypoints* proveniente de todas as combinações de pares de câmeras $\binom{H^{2D}}{2} = \{H^{2D}_{c_{ij}} \in Comb(H^{2D}) \;\middle|\; |H^{2D}_{c_{ij}}| = 2\}$ onde $H^{2D}_{c_{ij}} \in \mathbb{R}^{2\times 100\times 3}$, é gerado um conjunto de *keypoints* 3D $H^{3D} \in \mathbb{R}^{100\times 3}$ (sem a presença de um valor de confiança) da seguinte maneira:

  - Considerando $H^{2D}_{c_{ij}} = \{H^{2D}_{c_i}, H^{2D}_{c_j}\}$, é utilizado o algoritmo de 8 pontos (*8-point*) (LONGUET-HIGGINS, 1981) a fim de estimar a matriz essencial $E_{c_{ij}} \in \mathbb{R}^{3\times 3}$, a qual correlaciona *keypoints* em imagens estéreo (Equação 3.2):

$$\begin{aligned} &(y')^T E y = 0, \\ &\text{s.t. } y' = H^{2D}_{c_j},\ y = H^{2D}_{c_i} \end{aligned} \tag{3.2}$$

  - Agora, é calculada a pose da câmera C_j em relação à câmera C_i. Dessa maneira, é assumida a posição e orientação canônica para C_i. Para tal, é realizada a decomposição em valores singulares (*SVD*, *Singular Value Decomposition*) na matriz essencial $E_{c_{ij}}$ (GOLUB; REINSCH, 1971), a qual estima 4 hipóteses de pose para $C_j : \{\langle R_1, t\rangle, \langle R_1, -t\rangle, \langle R_2, t\rangle, \langle R_2, -t\rangle\}$. Cada hipótese é submetida a um teste de visibilidade (*cheirality check*) (NISTÉR, 2004), o qual verifica qual das hipóteses, quando utilizadas para triangular os *keypoints* 2D em $H^{2D}_{c_i}$ e $H^{2D}_{c_j}$, resulta em *keypoints* 3D com os valores de profundidade mais positivos (que, na prática, são *keypoints* visualizados por C_j);
  - Com os valores de parâmetros extrínsecos $C^{Ext}_i = \langle R_{c_i}, t_{c_i}\rangle$ e $C^{Ext}_j = \langle R_{c_j}, t_{c_j}\rangle$ (matriz de rotação e vetor de translação, respectivamente), é possível realizar a triangulação dos *keypoints* 2D observados em $H^{2D}_{c_{ij}}$ para gerar um conjunto de *keypoints* 3D $H^{3D}_{c_{ij}}$;
  - Agora que temos um conjunto de *keypoints* 3D para cada combinação de pares de câmeras $\binom{H^{3D}}{2} = \{T(H^{2D}_{c_i}, H^{2D}_{c_j}) \;\middle|\; \langle H^{2D}_{c_i}, H^{2D}_{c_j}\rangle \in \binom{H^{2D}}{2}\}$, é possível realizar correspondências 3D-2D utilizando $\binom{H^{2D}}{2}$ a fim de computar os parâmetros extrínsecos de todas as câmeras. Para tal, é adotado o algoritmo de otimização de *Levenberg-Marquardt* (LEVENBERG, 1944; MARQUARDT, 1963) a fim

de solucionar o problema de *PnP* (*Perspective-n-Point*), o qual busca calcular as poses das câmeras $\binom{C^{3D}}{2}$ as quais minimizem o erro de reprojeção entre os *keypoints* 3D, quando projetados, e os *keypoints* 2D de origem (Equação 3.3);

$$\begin{aligned}\binom{C^{3D}}{2} = \{\{PnP(H_j^{3D}, H_{j,c_i}^{2D}) \;\big|\; H_{j,c_i}^{2D} \in H_j^{2D}\} \\ \big|\; H_j^{3D} \in \binom{H^{3D}}{2} \;\big|\; H_j^{2D} \in \binom{H^{2D}}{2}\}\end{aligned} \tag{3.3}$$

– A esta altura, tem-se o conjunto de *keypoints* 3D $\binom{H^{3D}}{2}$ para todas as combinações de pares de câmeras $\binom{C^{3D}}{2}$. Dentre todas as combinações, é selecionado o conjunto de parâmetros extrínsecos de câmera $C_j^{3D} \in \binom{C^{3D}}{2}$ cuja correspondência com os *keypoints* 2D de origem é a mais confiável. Para tal, é calculado o máximo dos erros de reprojeções médios (*MMRE*, *Maximum Mean Reprojection Error*), o qual representa a média das distâncias euclidianas entre H_j^{3D} quando projetadas para cada uma das câmeras $C_{j,i}^{3D} \in C_j^{3D}$ e todos os conjuntos de *keypoints* 2D $H_{j,c_i}^{2D} \in H_j^{2D}$ e, como referência, é escolhida a correspondência 3D-2D cujo erro de reprojeção médio é o máximo (Equação 3.4). Por fim, são escolhidos os parâmetros de câmera $C^{PnP} \in \binom{C^{3D}}{2}$ e *keypoints* 3D $H^{3D,PnP} \in \binom{H^{3D}}{2}$ cujo *MMRE* é mínimo.

$$\begin{aligned}&MMRE = max(\{MRE(H_j^{3D}, C_{j,i}^{3D}, H_{j,c_i}^{2D}) \\ &\big|\; C_{j,i}^{3D} \in C_j^{3D} \;\big|\; H_{j,c_i}^{2D} \in H_j^{2D}\}), \\ &s.t.\ MRE = mean(Proj(H_j^{3D}, C_{j,i}^{3D}) - H_{j,c_i}^{2D}) \\ &\text{s.t. } H_j^{3D} \in \binom{H^{3D}}{2},\ C_j^{3D} \in \binom{C^{3D}}{2}, H_j^{2D} \in \binom{H^{2D}}{2}\end{aligned} \tag{3.4}$$

- Apesar de C^{PnP} e $H^{3D,PnP}$ representarem a cena, o processo de calibração das câmeras possui uma última etapa de refinamento do resultado em C^{PnP}, consistindo de um algoritmo de *Bundle Adjustment* (*BA*), o qual busca refinar os parâmetros de câmera e *keypoints* ao minimizar o erro de reprojeção das múltiplas câmeras em C^{BA} (TRIGGS et al., 1999; SIMÕES et al., 2012).

3.3 MÓDULO DE ESTIMAÇÃO DE POSES HUMANAS 3D

Dado que temos os *keypoints* 2D e os parâmetros extrínsecos de câmera C^{BA}, são estimadas as poses dos *keypoints* dos corpos humanos em 3D. A abordagem utilizada leva em consideração os elevados valores de confiança de localização de *keypoints* 2D nos diferentes pontos de vista, uma vez que as estimações de alto grau de confiança podem variar

em relação às câmeras, em função dos movimentos realizados pelos indivíduos. Similar à etapa de *H-BCPC*, é utilizado o conjunto de *keypoints* 2D O'', o qual agrupa os *keypoints* por *frame* (f), índice da pessoa (p) e tipo do *keypoint* (k).

Para cada grupo de *keypoints* 2D $O''_{b''} = \langle f, p, k, oq \rangle$, é utilizada a abordagem de triangulação algébrica para calcular a posição 3D do *keypoint* $T''_{b''}$, como proposto por Iskakov et al. (ISKAKOV et al., 2019). Com isso, busca-se solucionar o seguinte sistema (Equação 3.5):

$$(oq^q \circ A_k) {T''_{b''}}^{homo} = 0 \tag{3.5}$$

no qual $oq^q \in \mathbb{R}^C$ representa os valores de confiança dos *keypoints* 2D para cada câmera, $oq^p \in \mathbb{R}^{C*2}$ representa as localizações do *keypoint* 2D em cada câmera e $A_k \in \mathbb{R}^{2*C*4}$ agrega informações da matriz de projeção das câmeras, bem como das posições dos *keypoints* oq^p (HARTLEY; ZISSERMAN, 2003). Para solucionar o sistema, é utilizado o *Singular Value Decomposition* (*SVD*) (GOLUB; REINSCH, 1971), o qual estima o vetor de coordenadas homogêneas 3D ${T''_{b''}}^{homo}$. Então, ${T''_{b''}}^{homo}$ é dividido por sua quarta coordenada a fim de se obter a posição 3D final $T''_{b''}$.

É produzido um conjunto de *keypoints* 3D $T' = \{T'_1, T'_2, \ldots, T'_{G'}\}$ a partir de T'', no qual $T'_{g'} = \langle f, p, k, x, y, z, u \rangle$ possui o seguinte significado: no *frame* f, a pessoa de índice p possui um *keypoint* do tipo k localizado em $\langle x, y, z \rangle$ com um valor de confiança de u, definido pelo erro de reprojeção médio de $T'_{g'}$ em relação aos seus *keypoints* 2D de origem.

Por fim, são estimadas as novas localizações dos *keypoints* em duas dimensões a partir da reprojeção de T' em cada uma das câmeras $C_i \in C$. Este novo conjunto de *keypoints* 2D $D' = \{D'_1, D'_2, \ldots, D'_{B'}\}$, no qual $D'_b = \langle f, c, p, k, x, y, q \rangle$, possuindo o mesmo formato de $O'_{b'}$, combina a informação de múltiplos pontos de vista e pode ser utilizado para gerar *datasets* de estimações de poses humanas.

3.4 MÓDULO DE CONVERSÃO DE FORMATOS DE *DATASETS*

A última etapa do *framework* encapsula o cenário de criação de *datasets* para *HPE* a partir de conteúdos visuais. Apesar dos *benchmarks* existentes seguirem as suas próprias regras de formatações, foi escolhido criar algoritmos de conversão de *datasets* de *HPE* para os formatos dos três dos *benchmarks* mais utilizados: o *MS COCO* (LIN et al., 2014), o *MPII Human Pose* (ANDRILUKA et al., 2014) e o *LSP* (JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011). Dessa maneira, o *dataset* gerado pode ser utilizado em *pipelines* de treinamento de validação de modelos de *Deep Learning* para *HPE* de forma transparente.

4 EXPERIMENTOS E RESULTADOS

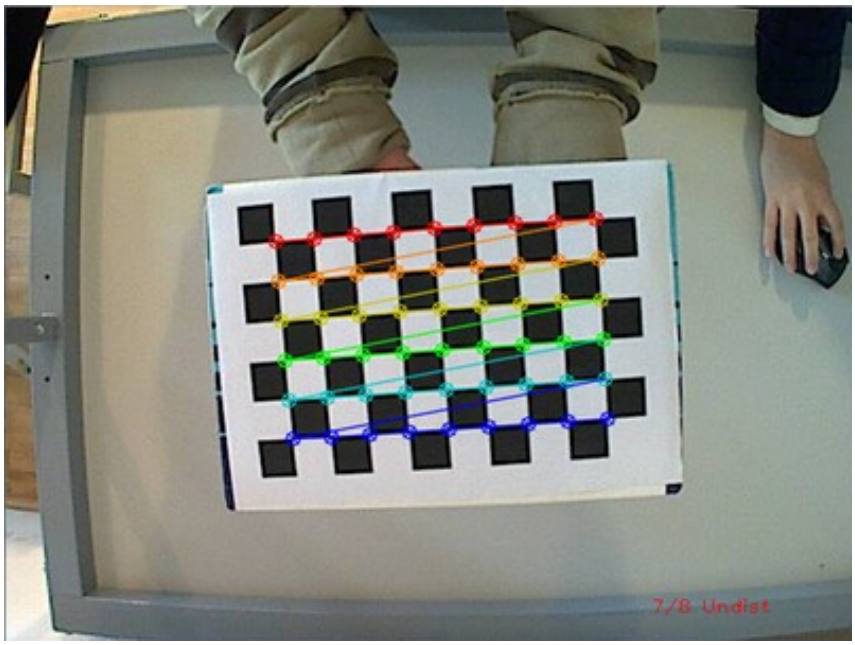
Nesta seção, serão detalhados os passos realizados para a avaliação do *HuTrain* no seu propósito de concepção de *datasets*. Serão apresentados resultados os quais ajudem a evidenciar a facilidade e agilidade em criar *datasets* ao utilizar o *framework*. Também serão descritos os recursos de *software* e físicos utilizados. A fim de entender a qualidade dos resultados, também são realizadas comparações em relação a um método de estimações de poses humanas presente no estado da arte.

4.1 CONFIGURAÇÃO E IMPLEMENTAÇÃO

Para a execução dos experimentos do *HuTrain*, foram utilizadas câmeras do modelo *GoPro Hero 7* (Figura 18). Estes experimentos foram realizados em uma sala de dimensões 3.72m $\times$ 3.27m. Além disso, foi utilizado um tabuleiro de xadrez impresso em uma folha de papel a fim de capturar imagens com a câmera e realizar a estimação dos seus parâmetros intrínsecos.

Figura 18 – Modelo de câmera *GoPro Hero 7*.

Figura 19 – Tabuleiro de xadrez, utilizado para estimar os parâmetros intrínsecos da câmera.



Fonte: https://docs.opencv.org/3.4/d4/d94/tutorial_camera_calibration.html

A implementação do *framework* foi desenvolvida na linguagem de programação *Python* (ROSSUM, 1995), na versão 3.7. Dentre as bibliotecas utilizadas, destaca-se o uso do *PyTorch* (PASZKE et al., 2017), um *framework* para técnicas de aprendizagem de máquina, empregado no módulo de estimações de poses 2D iniciais (CAO et al., 2017) e no algoritmo de re-identificação de indivíduos (DONG et al., 2019). Em relação aos algoritmos envolvendo a calibração das câmeras e triangulação das poses humanas, foi utilizado o *OpenCV* (BRADSKI; KAEHLER, 2000), uma biblioteca a qual contempla algoritmos relacionados a processamento de imagens e álgebra linear. Da etapa de estimações de poses 2D iniciais ao algoritmo de conversão de formatos de *datasets*, contou-se com o auxílio do *NumPy* (OLIPHANT, 2006), um pacote em *Python* utilizado na manipulação de matrizes multidimensionais de forma otimizada. Todas as bibliotecas de *Python* utilizadas, bem como as suas respectivas versões, são descritas no Apêndice D.

Para o desenvolvimento da implementação e testes do *framework*, foi utilizado um *laptop* contendo o sistema operacional *Windows 10* (64-*bit*), processador *Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz*, memória *RAM* de 16GB, além de uma placa gráfica (*GPU*) de modelo *GeForce GTX 1060* da marca *Nvidia*, esta contendo 6GB de memória *RAM*.

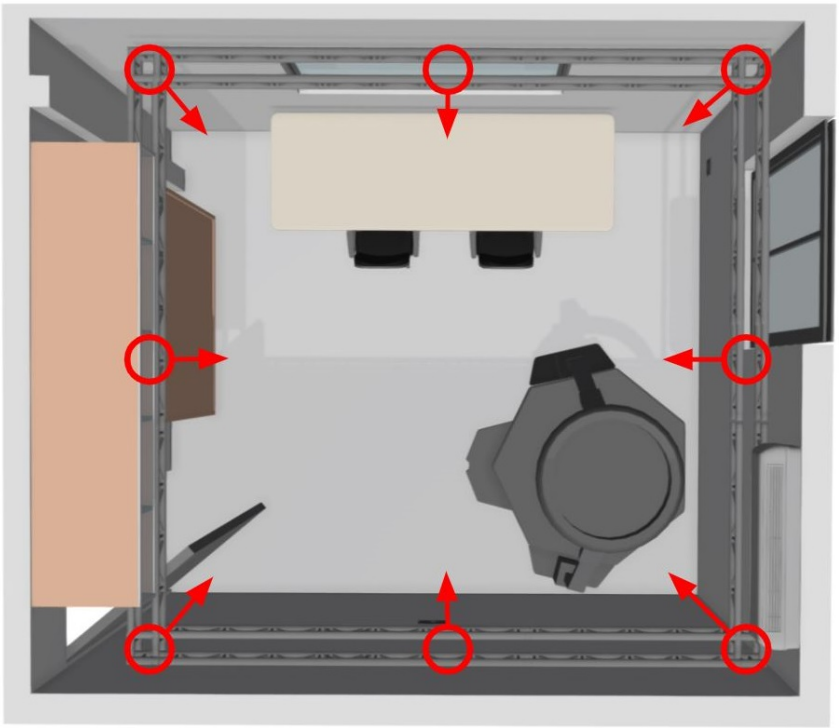
4.2 CONCEPÇÃO DO *DATASET*

Para a avaliação do processo de geração do *dataset* utilizando o *HuTrain*, foram realizadas gravações de um indivíduo realizando movimentos aleatórios com os membros superiores e inferiores, além de rotações e círculos ao redor da sala. As câmeras foram posicionadas

em vigas instaladas nas regiões laterais superiores da sala, apontando para o centro e de cima para baixo (*top-down*), como podem ser vistas na Figuras 20 e 21.

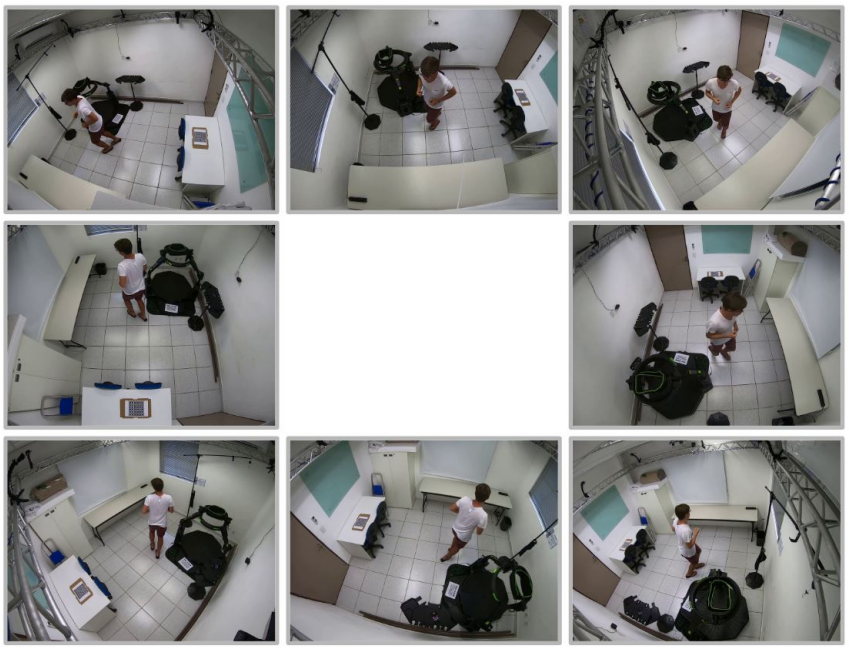
Para tal, foram utilizados 8 pontos de vista, gerando assim 8 vídeos. Cada um dos vídeos gravados possui resolução de 1920 × 1440. Após isso, foi utilizado o *software DaVinci Resolve*[1] para realizar a sincronização dos vídeos, a fim de que todos possuam a mesma taxa de quadros por segundo, a mesma quantidade de quadros e que o quadro inicial de cada vídeo represente o mesmo *timestamp* do mundo real. Por fim, cada um dos 8 vídeos possui 4,061 *frames* a 60 *fps*, totalizando 32,488 *frames*.

Figura 20 – Sala utilizada para a realização dos testes. As câmeras utilizadas foram posicionadas em uma viga, ao redor da sala (círculos em vermelho) e apontadas para o centro (setas em vermelho).



[1] https://www.blackmagicdesign.com/br/products/davinciresolve/

Figura 21 – Campo de visão de cada uma das 8 câmeras utilizadas.



4.3 VISUALIZAÇÃO DOS RESULTADOS

Também foi desenvolvido um visualizador no qual o usuário possa analisar as capturas de movimento geradas a partir da utilização do *HuTrain* (Figura 22). Esta ferramenta foi desenvolvida utilizando a plataforma de desenvolvimento *Unity3D*[2], na versão 2019.3.1f1 (64-*bit*). Nela, é possível visualizar o conjunto de poses geradas para cada *frame* do *dataset*, a partir da navegação pelo espaço tridimensional fornecido pelo *Unity3D*.

[2] https://unity.com/pt

Figura 22 – Ferramenta de visualização das capturas de movimentos geradas pelo *HuTrain*.

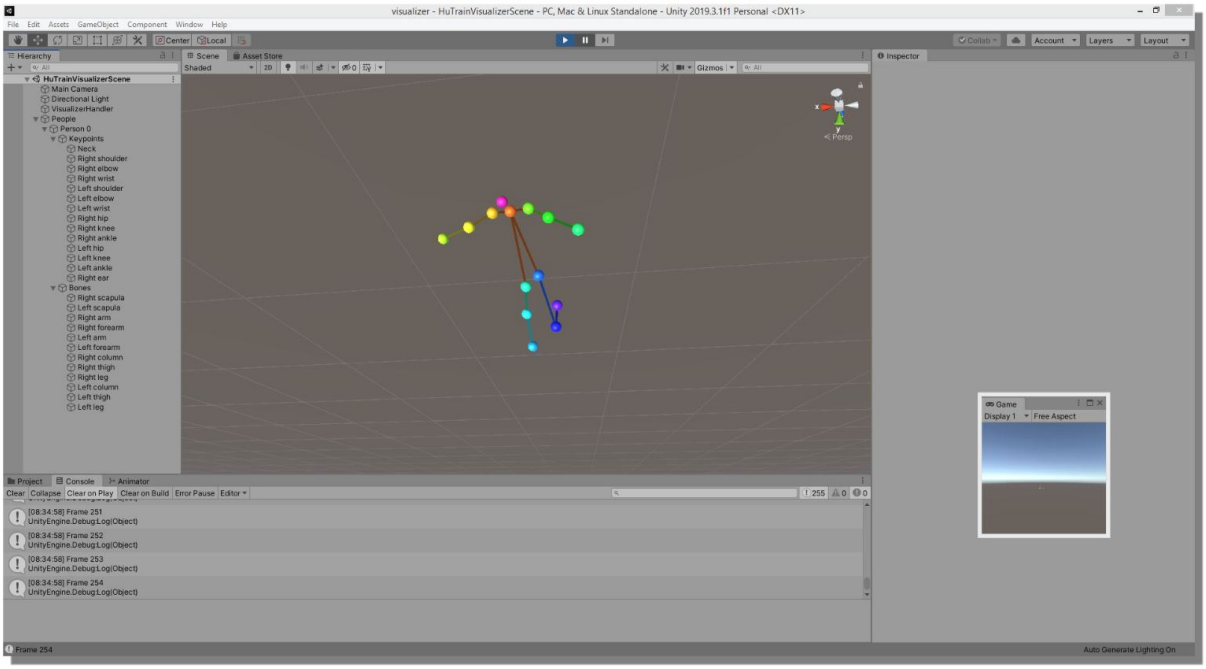
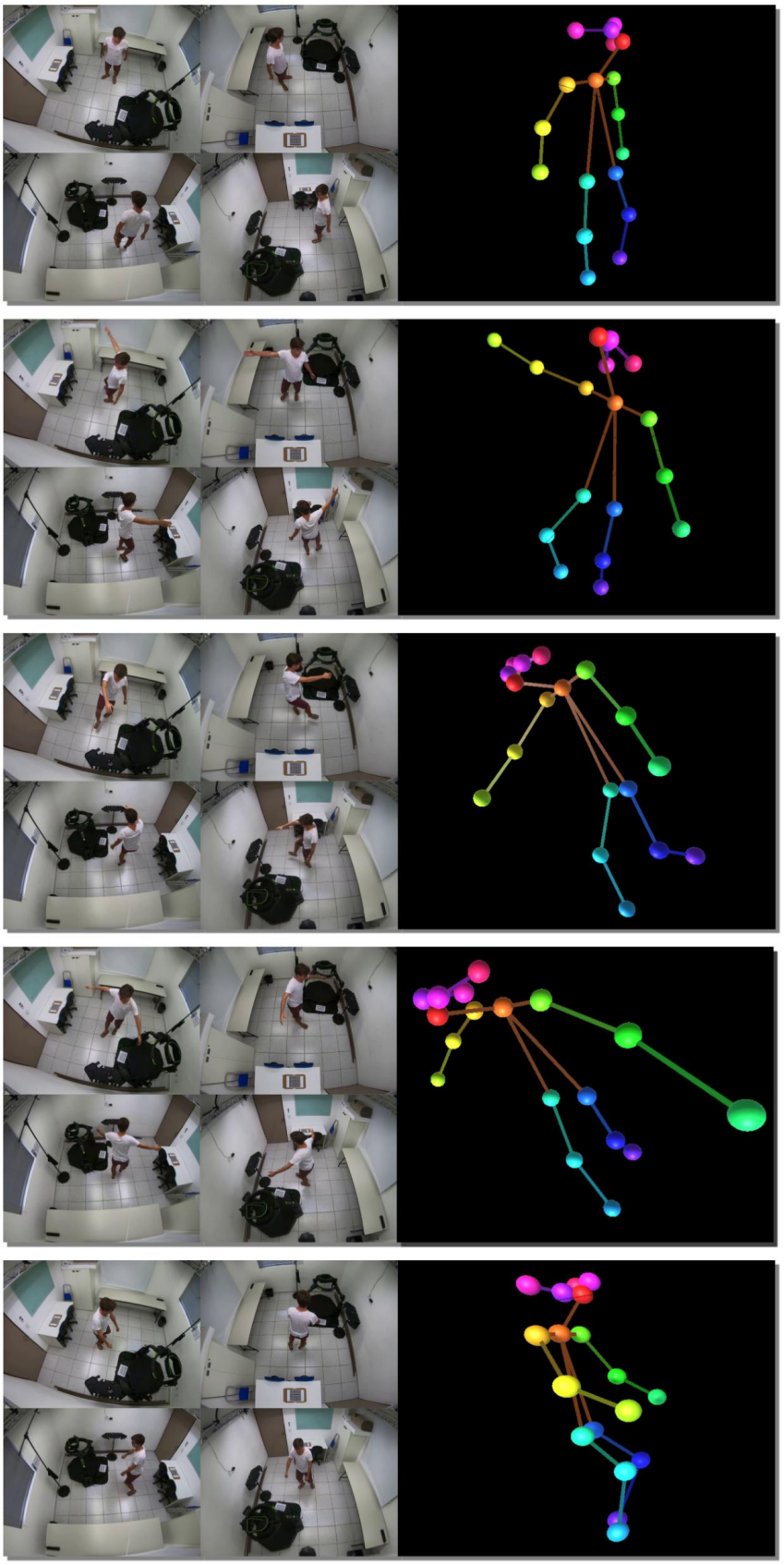


Figura 23 – Resultados qualitativos do *dataset* gerado.



4.4 TEMPO DE EXECUÇÃO

Para avaliarmos a velocidade de geração de um *dataset* utilizando o *HuTrain*, foram calculados os tempos necessários para realizar cada uma das etapas (Tabela 4). Estas etapas estão divididas em duas categorias:

- **Etapas pré-*HuTrain***: São as etapas necessárias anteriores à execução do *pipeline* descrito pelo *HuTrain*. Nela, são realizados o cálculo dos parâmetros intrínsecos das câmeras utilizadas, o posicionamento (*setup*) das câmeras no ambiente a ser gravado e a gravação dos movimentos;
- **Etapas do *HuTrain***: São as etapas de geração do *dataset* a partir dos vídeos e parâmetros intrínsecos provenientes das câmeras. Estas compreendem as abordagens descritas na Seção 3.

Tabela 4 – Tempo de execução de cada uma das etapas necessárias para a geração de um *dataset* utilizando o *HuTrain*.

Etapa	Tempo Percorrido
Etapas pré-*HuTrain*	
Calibração dos parâmetros intrínsecos	1min
Setup das câmeras	5min
Gravação dos movimentos	5min
Etapas do *HuTrain*	
Estimações de poses humanas 2D iniciais	1h 30min
Calibração dos parâmetros extrínsecos	2min
Estimações de poses humanas 3D	1min
Conversão de formatos de *datasets*	10s
Total (pré-*HuTrain*)	11min
Total (*HuTrain*)	1h 33min 10s
Total	**1h 44min 10s**

Além disso, foi realizada uma comparação com os tempos de geração dos *datasets* de *HPE* existentes na academia. Uma vez que poucos *datasets* dispõem de dados referentes ao tempo gasto para serem concebidos, foram contatados os autores dos trabalhos referentes aos *datasets* propostos cujo tempo de concepção não foi informado. Além disso, esta comparação foi separada em duas categorias:

- ***MLDs* e *AGDs***: Nas quais o tempo de concepção concentra-se na **anotação** dos *samples*;

- ***SGDs***: Na qual o tempo de concepção concentra-se na **renderização** dos *samples.*

Tabela 5 – Custo de tempo para a geração dos *datasets* para *HPE* existentes na academia, incluindo o proposto. Os custos de tempo dos *datasets* marcados com asterisco (*) foram obtidos atráves do contato com os autores envolvidos, enquanto que os não marcados foram extraídos diretamente dos seus respectivos *papers.*

Dataset	# Amostras	Custo de Tempo Total	Custo de Tempo por Amostra
MLDs e AGDs			
MPII Human Pose* (ANDRILUKA et al., 2014)	+ 40,000	± 1000 - 1333h	1.5 - 2min
MS COCO (LIN et al., 2014)	± 250,000	+ 70,000h	± 16min 48s
LSP* (JOHNSON; EVERINGHAM, 2010)	2,000	20h	36s
LSP Extended (JOHNSON; EVERINGHAM, 2011)	10,000	25h	9s
H3D (BOURDEV; MALIK, 2009)	2,000	± 166h	5min
MPII Cooking* (ROHRBACH et al., 2012)	1,277	± 21h	± 1min
HuTrain	**32,488**	**1h 44min 10s**	**0.19s**
SGDs			
xR-EgoPose* (TOME et al., 2019)	383,000	± 25500h	4min
SURREAL* (VAROL et al., 2017)	6,500,000	± 3600 - 9000h	2 - 5s
JTA* (FABBRI et al., 2018)	± 10,000,000	± 3700h	1.33s
Human3D+* (CHEN et al., 2016)	**5,099,405**	**48h**	**± 0.03s**

4.5 RESULTADOS QUANTITATIVOS

Para entender a qualidade do *dataset* gerado, foram realizadas anotações manuais de imagens pertencentes ao *dataset* gerado. Uma vez que o *dataset* anota informações tridi-

mensionais, existe a dificuldade de se gerar anotações manuais das amostras diretamente em três dimensões (BOURDEV; MALIK, 2009). Para tal, são escolhidos conjuntos de imagens provenientes de diferentes câmeras, porém pertencentes a *timestamps* iguais.

Foram escolhidos 13 *timestamps* aleatórios, em 12 dos quais foram extraídas as imagens das 8 câmeras, enquanto que, em um *timestamp*, foram extraídas imagens de 4 câmeras, totalizando 100 imagens. Com isso, foi realizada a anotação manual da localização dos *keypoints*, em *pixels*, em cada uma das imagens. Após isso, calcula-se o erro associado às poses estimadas pelo *HuTrain* em relação às poses anotadas manualmente, a partir da métrica descrita na Subseção 4.5.1.

Os resultados obtidos foram comparados a outras duas abordagens de estimações de poses humanas do estado da arte:

- O algoritmo de estimações de poses humanas *bottom-up OpenPose* (CAO et al., 2017), este treinado no *dataset MS COCO* (LIN et al., 2014);
- O algoritmo de estimações de poses humanas *top-down HRNet* (SUN et al., 2019), este também treinado no *dataset MS COCO* (LIN et al., 2014), realizando o uso prévio do algoritmo de detecção de pessoas (*bounding-box*) *YOLOv3* (REDMON; FARHADI, 2018), este treinado no *dataset PASCAL VOC* (EVERINGHAM et al., 2010)).

4.5.1 Erro de Reprojeção Médio

Foi adotado o erro de reprojeção, definido pela distância euclidiana entre a anotação manual do *keypoint* 2D em uma imagem e a projeção do *keypoint* 3D (estimada pelo *HuTrain*) no ponto de vista referente à mesma imagem. Esta métrica foi escolhida pois, além de anotações manuais em duas dimensões apresentarem maior precisão quando comparadas às tridimensionais (BOURDEV et al., 2010), a avaliação do erro de reprojeção de um *keypoint* 3D em múltiplas vistas é uma avaliação implícita da qualidade de localização espacial (LEPETIT; FUA, 2005). Com isso, calcula-se a média das distâncias euclidianas para cada um dos tipos de *keypoints* (por exemplo, ombros, joelhos, nariz, pescoço, etc.) a partir da seguinte fórmula:

$$E = \frac{1}{O_k} \sum_{i}^{O_k} \sqrt{(H_i^x - M_i^x) + (H_i^y - M_i^y)}^2 \tag{4.1}$$

na qual k é o tipo do *keypoint*, O_k é a quantidade total de ocorrências do *keypoint* de tipo k nas 100 imagens, H_i é a projeção do *keypoint* 3D na imagem correspondente ao *keypoint* 2D anotado manualmente M_i.

Tabela 6 – Resultados do erro de reprojeção médio para cada um dos tipos de *keypoints*. São apresentados os resultados utilizando o *HRNet* (SUN et al., 2019) junto ao *YOLOv3* (REDMON; FARHADI, 2018), o *OpenPose* (CAO et al., 2017) e o proposto pelo *HuTrain*. Os erros marcados em negrito (para cada tipo de *keypoint*) representam os melhores resultados.

Keypoint	***HRNet* (SUN et al., 2019) + *YOLOv3* (REDMON; FARHADI, 2018)**	***OpenPose* (CAO et al., 2017)**	***HuTrain***
Nariz	72.0	309.0	**10.2**
Pescoço	51.1	**26.5**	27.2
Ombros	54.6	14.7	**13.7**
Cotovelos	76.0	27.1	**16.8**
Pulsos	111.8	96.7	**48.0**
Pélvis	63.1	**24.5**	26.1
Joelhos	80.1	33.2	**18.6**
Tornozelos	63.7	25.7	**14.0**
Olhos	70.1	442.8	**12.1**
Orelhas	52.9	300.2	**11.9**

4.6 USABILIDADE

Para entender o grau de praticidade do uso do *HuTrain*, foi realizada uma avaliação com 3 usuários que possuem experiência em computação, porém nunca participaram da concepção de *datasets* para estimações de poses humanas (Tabela 7). Estes foram instruídos a conceber um *dataset* para estimação de poses humanas a partir da utilização do *framework*.

Tabela 7 – Usuários participantes da avaliação, bem como os seus respectivos graus de *expertise*, autoavaliados de 1 a 5, na linguagem de programação *Python* e na plataforma de desenvolvimento *Unity3D*.

	Grau de experiência em *Python*	**Grau de experiência em *Unity3D***
Usuário 1	5	1
Usuário 2	2	4
Usuário 3	4	4

A fim de focar a avaliação do uso do *HuTrain*, são disponibilizadas aos usuários gravações de vídeos (previamente sincronizadas) provenientes das diferentes câmeras utilizadas,

bem como as imagens dos tabuleiros de xadrez para a realização da calibração dos parâmetros intrínsecos das câmeras. Com isso, abstraem-se as etapas pré-*HuTrain* (descritas na Tabela 4).

A partir disso, os usuários são instruídos a executar o passo a passo da concepção do *dataset*, descrito no *README* do *framework*. Para realizar a escrita do *README*, foi utilizado o *WRITEME*[3] (LEITE, 2019), uma ferramenta para o auxílio à escrita de *READMEs* a partir da recomendação de estruturas de escrita baseada nos repositórios mais populares existentes no *GitHub*. Ao término da geração do *dataset*, os usuários também são instruídos a visualizar os resultados concebidos a partir da utilização da ferramenta em *Unity3D*.

4.6.1 *User Experience Questionnaire*

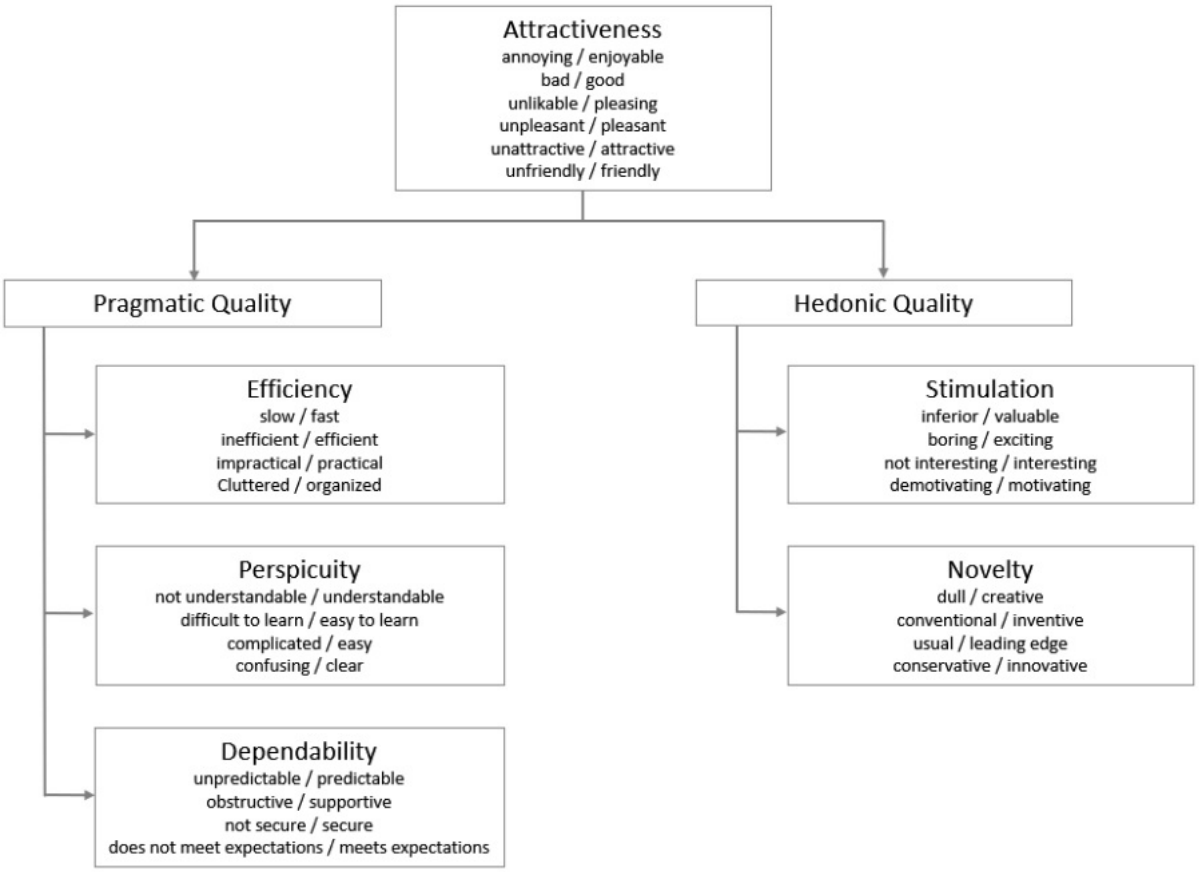
Os usuários envolvidos foram submetidos a um questionário, cujo objetivo é avaliar aspectos referentes à usabilidade do *framework*. Para tal, foi utilizado o *User Experience Questionnaire* (*UEQ*), um questionário focado em avaliar a experiência de uso de um sistema (LAUGWITZ; HELD; SCHREPP, 2008). O *UEQ*, originalmente, busca avaliar 6 aspectos gerais:

- ***Attractiveness*** **(Atratividade)**: Avalia o quanto os usuários gostaram do *framework*, de forma geral;
- ***Perspicuity*** **(Perspicuidade)**: Avalia o quão fácil é utilizar e aprender a usar o sistema;
- ***Efficiency*** **(Eficiência)**: Avalia o quanto de esforço é necessário para o usuário atingir as suas finalidades;
- ***Dependability*** **(Confiabilidade)**: Avalia o quanto o usuário se sente em controle de utilizar o sistema;
- ***Stimulation*** **(Estimulação)**: Avalia a empolgação e motivação do usuário ao utilizar o sistema;
- ***Novelty*** **(Inovação)**: Analisa o quanto os usuários acharam o sistema inovador e criativo.

Uma vez que o *framework* não apresenta interface gráfica, optou-se por não avaliar os aspectos de *stimulation* e *novelty*, os quais não avaliam aspectos diretamente relacionados à finalidade do sistema. Enquanto que o aspecto de *attractiveness* avalia o sistema de forma geral, *perspicuity*, *efficiency* e *dependability* são considerados aspectos pragmáticos de usabilidade (Figura 24).

[3] https://hpbl.github.io/WRITEME/

Figura 24 – Aspectos abordados pelo *UEQ* e seus respectivos itens.



Fonte: (LAUGWITZ; HELD; SCHREPP, 2008)

Cada um dos itens presentes nos aspectos citados é respondido pelos usuários em uma escala *Likert* de 1 a 7, na qual 1 aproxima-se mais da palavra descrita à esquerda e 7 aproxima-se mais da sua palavra oposta, descrita à direita (Figura 25). A fim de evitar enviesamento nas respostas providas pelos usuários, o *UEQ* propõe que as características opostas (negativas e positivas) sejam posicionadas em lados aleatórios.

Figura 25 – Itens presentes no *UEQ* utilizados na avaliação. Nota-se a presença de características positivas em ambos os lados (direito e esquerdo). Além disso, itens pertencentes aos mesmos aspectos são distribuídos em uma ordem aleatória.

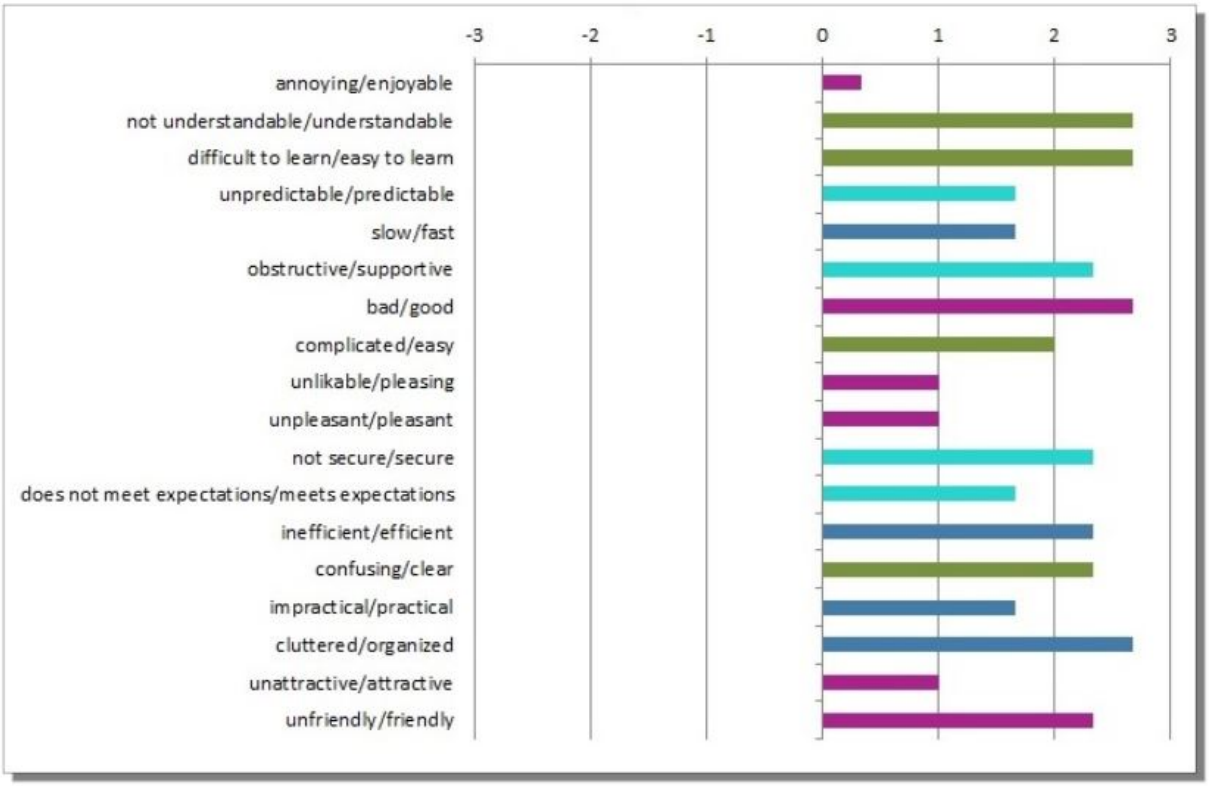
Fonte: (LAUGWITZ; HELD; SCHREPP, 2008)

Uma vez que os usuários respondem cada um dos itens na escala *Likert* de 1 a 7, estes valores são convertidos para pontuações definidas em um intervalo de -3 a 3. Esta conversão funciona da seguinte maneira:

- Caso o item possua a característica **positiva** no lado direito e seu oposto esteja no lado esquerdo, subtrai-se 4 ao valor obtido na escala *Likert*;
- Caso o item possua a característica **negativa** no lado direito e seu oposto esteja no lado esquerdo, primeiramente subtrai-se 4 e depois multiplica-se por -1 o valor obtido na escala *Likert*.

A Figura 26 apresenta a média dos resultados obtidos para cada um dos itens contemplados pelos aspectos de *attractiveness*, *perspicuity*, *efficiency* e *dependability*.

Figura 26 – Média dos resultados do *UEQ* de cada um dos itens respondidos pelos usuários. Estes itens são agrupados nos aspectos de *attractiveness* (roxo), *perspicuity* (verde), *efficiency* (azul escuro) e *dependability* (azul claro).

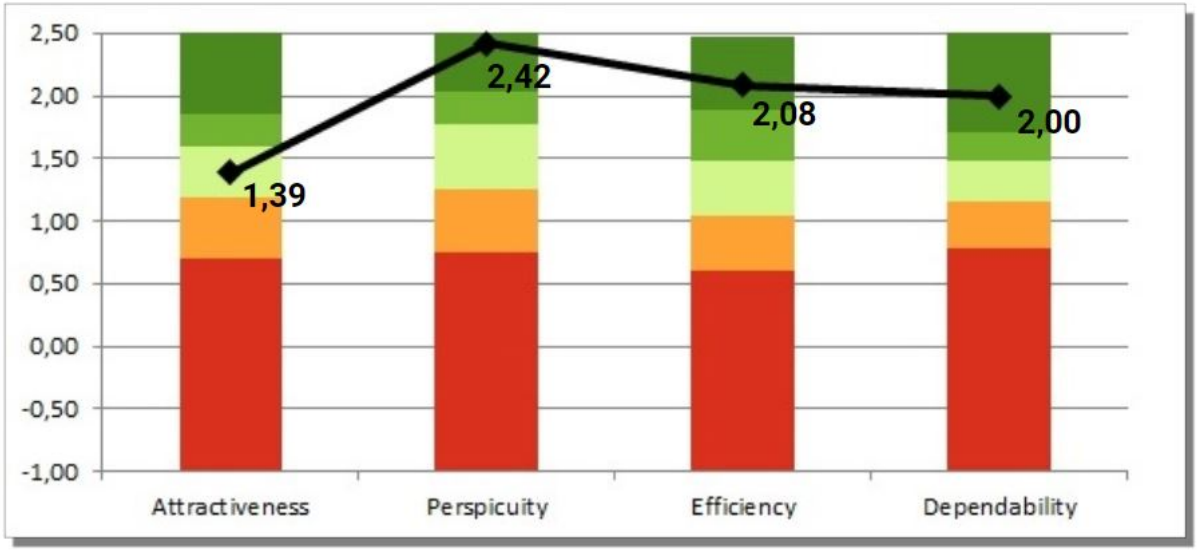


O *UEQ* também provê a avaliação de 452 sistemas (com um total de 20190 participantes) a partir do questionário proposto. Com isso, estes sistemas são categorizados em 5 escalas:

- ***Excellent* (Excelente)**: Representa o intervalo de 10% dos melhores resultados;
- ***Good* (Bom)**: Representa o intervalo entre 10% dos melhores resultados e 75% dos piores resultados;
- ***Above average* (Acima da média)**: Representa o intervalo entre 25% dos melhores resultados e 50% dos piores resultados;
- ***Below average* (Abaixo da média)**: Representa o intervalo entre 50% dos melhores resultados e 25% dos piores resultados;
- ***Bad* (Ruim)**: Representa o intervalo de 25% dos piores resultados.

Portanto, cada um dos aspectos do *UEQ* possuirá limiares de pontuações distintos, para cada uma das escalas. A Figura 27 apresenta estas variações de escalas para cada um dos aspectos, bem como os resultados obtidos através da avaliação do *HuTrain* pelos usuários.

Figura 27 – Média dos resultados do *UEQ* (pontuações em preto) para os aspectos avaliados (*attractiveness*, *perspicuity*, *efficiency* e *dependability*). Cada uma das escalas é representada por uma cor: Excelente (verde escuro), bom (verde), acima da média (verde claro), abaixo da média (laranja) e ruim (vermelho).



4.6.2 *System Usability Scale*

Além do *UEQ*, os usuários também foram instruídos a responderem o *System Usability Scale* (*SUS*), um questionário simples e de baixo custo, cujo objetivo é de analisar o nível de usabilidade de um sistema (BROOKE, 1996). O *SUS* auxilia na avaliação dos seguintes critérios:

- ***Effectiveness*** **(Efetividade)**: Avalia se os usuários conseguem atingir os seus objetivos ao utilizar o sistema;
- ***Efficiency*** **(Eficiência)**: Avalia o quanto de esforço e recursos investidos pelos usuários foram necessários ao utilizar o sistema;
- ***Satisfaction*** **(Satisfação)**: Avalia se as expectativas dos usuários foram atingidas ao utilizar o sistema.

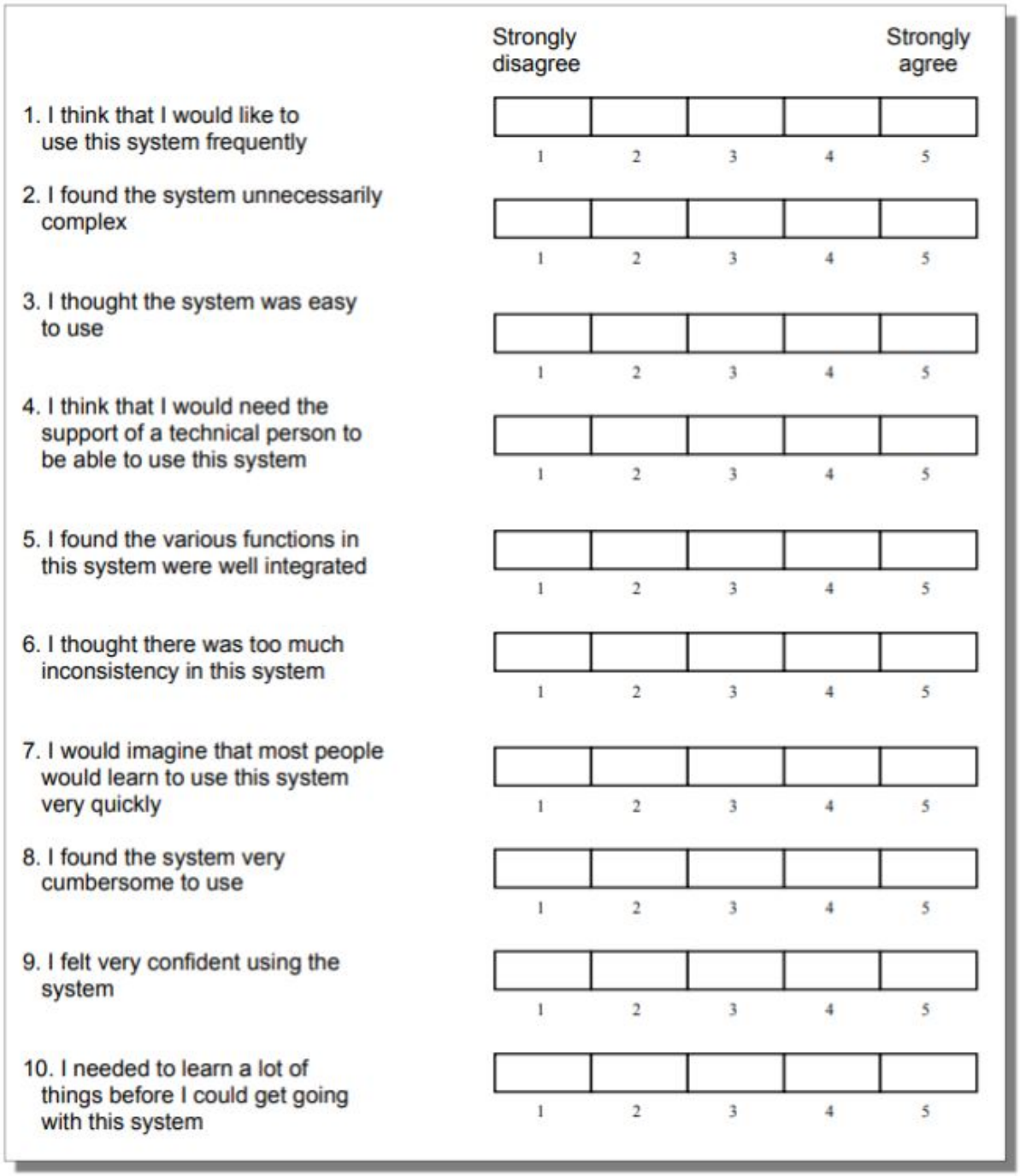
Este teste consiste de 10 sentenças, as quais os usuários devem responder informando o quanto concordam com cada uma delas. Similar ao *UEQ*, as respostas são realizadas em uma escala *Likert* de 1 a 5, onde 1 representa discordância total com a sentença e 5 representa concordância total. São as sentenças do *SUS*:

1. Eu acho que eu gostaria de usar esse sistema frequentemente;
2. Eu achei o sistema desnecessariamente complexo;

3. Eu achei o sistema fácil de usar;
4. Eu acho que eu precisaria de suporte técnico para ser capaz de usar esse sistema;
5. Eu achei que as várias funcionalidades do sistema estavam bem integradas;
6. Eu achei que havia muita inconsistência no sistema;
7. Eu imagino que a maioria das pessoas aprenderiam a usar esse sistema rapidamente;
8. Eu achei o sistema muito incômodo de se usar;
9. Eu me senti muito confiante usando o sistema;
10. Eu precisei aprender várias coisas antes que eu pudesse usar o sistema adequadamente.

Vale notar a presença de sentenças com conotações positivas e negativas, estas dispostas de forma alternada (similarmente ao *UEQ*), a fim de evitar respostas enviesadas em uma mesma alternativa (Figura 28).

Figura 28 – Estrutura do *System Usability Scale*. As sentenças são respondidas em uma escala *Likert* de 1 a 5. Além disso, as sentenças ímpares possuem conotações positivas, enquanto que as pares possuem conotações negativas.



Fonte: (BROOKE, 1996)

Após os usuários responderem o questionário, os resultados são calculados da seguinte maneira:

- Para cada sentença ímpar, subtrai-se 1 da pontuação respondida pelo usuário;

- Para cada sentença par, subtrai-se a pontuação respondida de 5;
- Soma-se todos os valores obtidos das duas etapas anteriores e multiplica-se o resultado por 2.5.

Com isso, a pontuação do *SUS* é situada em um intervalo de 0 (pontuação mínima) a 100 (pontuação máxima). Também é possível calcular a pontuação de cada sentença individualmente ao multiplicar o resultado por 25. Segundo Brooke et al., sistemas com pontuações abaixo de 68 pontos apontam para possíveis problemas de usabilidade.

Bangor et al. realizou um estudo para compreender os resultados numéricos obtidos através do *SUS* (BANGOR; KORTUM; MILLER, 2008). Para tal, foi proposto um mapeamento destes valores para entender a aceitabilidade do sistema. Esta classificação propõe três escalas:

- ***Acceptable* (Aceitável)**: Representa os sistemas com pontuações no *SUS* de 71.1 a 100;
- ***Marginal***: Representa os sistemas com pontuações no *SUS* de 51.7 a 71;
- ***Not acceptable* (Não aceitável)**: Representa os sistemas com pontuações no *SUS* de 0 a 51.6.

Bangor et al. (BANGOR; KORTUM; MILLER, 2009) propôs a extensão da classificação do *SUS* realizada por Bangor et al. (BANGOR; KORTUM; MILLER, 2008), a fim de encontrar o mapeamento entre as pontuações obtidas através do *SUS* e adjetivos que as caracterizassem. Este estudo envolveu um total de 964 participantes, e classificou as pontuações a partir dos seguintes adjetivos:

- ***Best imaginable* (Melhor imaginável)**: Representa os sistemas com pontuações no *SUS* de 84.1 a 100;
- ***Excellent* (Excelente)**: Representa os sistemas com pontuações no *SUS* de 80.8 a 84;
- ***Good* (Bom)**: Representa os sistemas com pontuações no *SUS* de 71.1 a 80.7;
- ***OK***: Representa os sistemas com pontuações no *SUS* de 51.7 a 71 (equivalente à escala *marginal* proposta por Bangor et al. (BANGOR; KORTUM; MILLER, 2008));
- ***Poor* (Ruim)**: Representa os sistemas com pontuações no *SUS* de 25.1 a 51.6;
- ***Worst imaginable* (Pior imaginável)**: Representa os sistemas com pontuações no *SUS* de 0 a 25.

A Figura 29 apresenta a média das pontuações de cada sentença respondida pelos usuários, bem como a média dos resultados do questionário em geral. Já a Figura 30 posiciona o *HuTrain* em relação às métricas do *SUS*. Com isso, o resultado final da avaliação do *HuTrain* no *System Usability Scale* classifica-o como um sistema **aceitável** e **bom**.

Figura 29 – Média de pontuações de cada sentença do *SUS*, bem como a média das pontuações individuais.

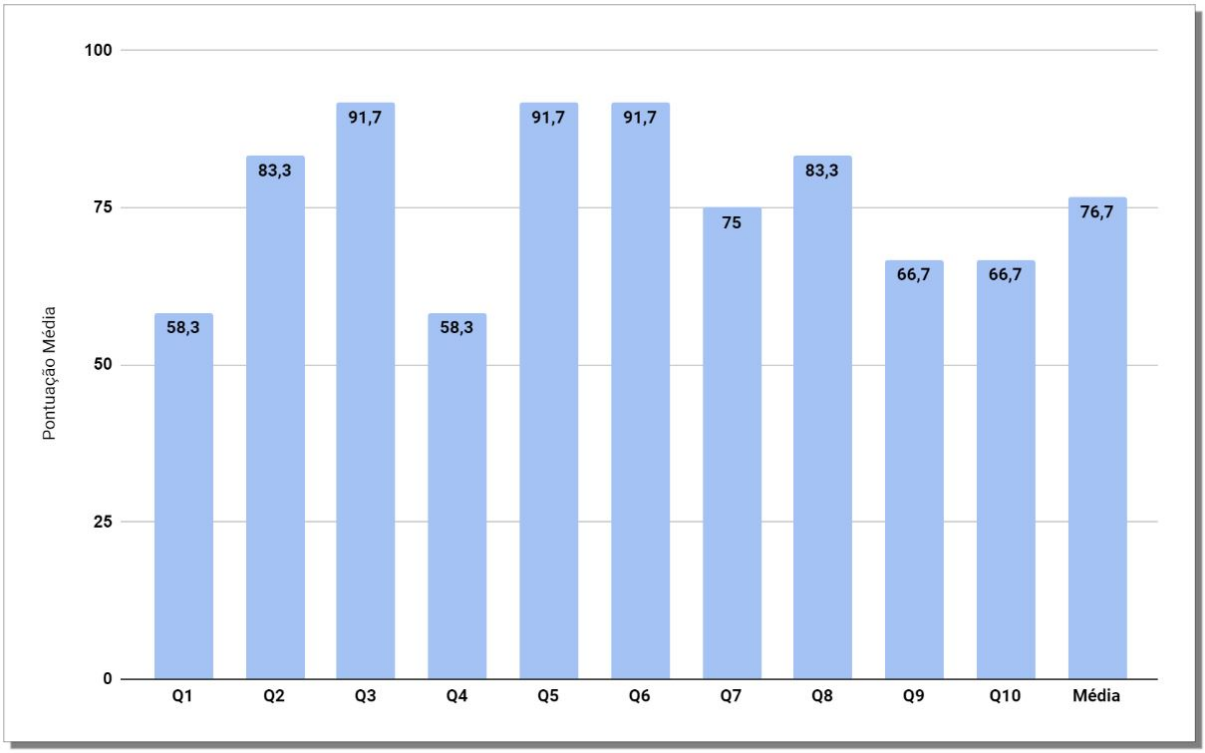
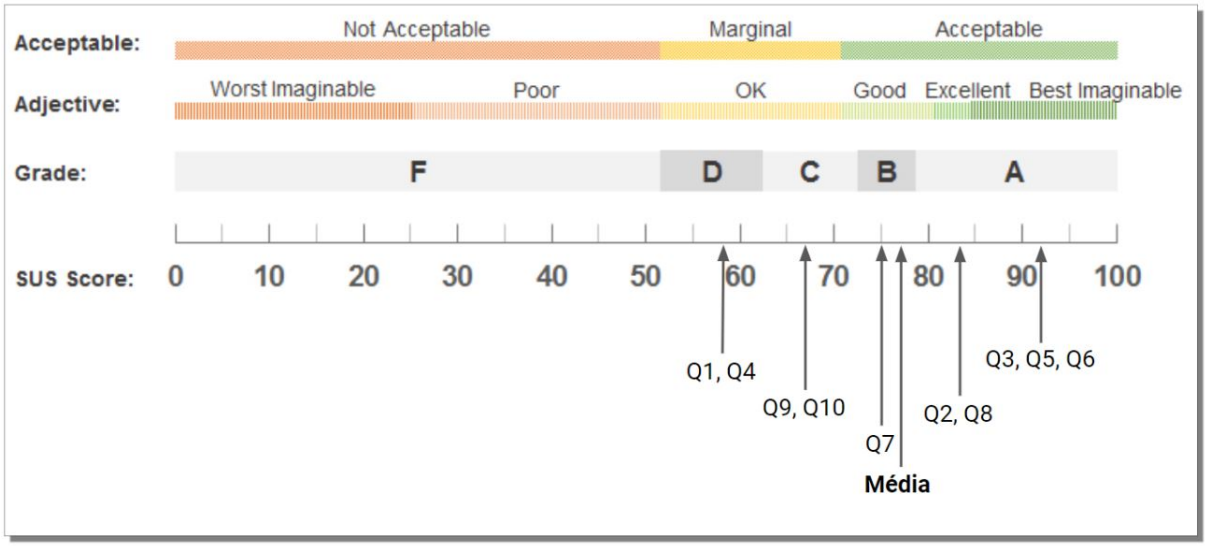


Figura 30 – Resultados da avaliação do *HuTrain* no *System Usability Scale* em relação aos graus de aceitabilidade (BANGOR; KORTUM; MILLER, 2008), mapeamento de pontuação para características (BANGOR; KORTUM; MILLER, 2009) e pontuação no *SUS* (BROOKE, 1996).



5 DISCUSSÃO

5.1 ANÁLISE DO TEMPO DE EXECUÇÃO

Primeiramente, analisou-se o tempo de execução de cada etapa presente no processo de geração do *dataset* utilizando o *HuTrain*. Considere a Tabela 4. Iniciando-se pela etapa de calibração dos parâmetros intrínsecos das câmeras, observa-se um tempo relativamente rápido, uma vez que o uso da implementação em *Python* requer a execução de apenas um comando o qual realiza toda esta etapa de forma automática. Em relação às etapas executadas antes da utilização do *HuTrain*, o *setup* das câmeras e a gravação dos movimentos representam a maioria (aproximadamente 91%) do total de tempo de geração do *input* do *HuTrain*. Todavia, estas etapas são intrínsecas ao processo de captura de movimento e independem do *pipeline* utilizado.

Em relação às etapas contempladas pelo *HuTrain*, as estimações de poses humanas 2D iniciais mostra-se como a de maior consumo de tempo, correspondendo a 86.4% do tempo total de geração do *dataset*. Isso ocorre devido à implementação da etapa pós-*CNN* das estimações de poses 2D em *Python*[1]; enquanto que esta implementação atinge cerca de 6*fps* nas configurações de máquina testada (Seção 4.1), a implementação original[2], esta compilada na linguagem de programação *C++* e realizando uso do *framework* de aprendizagem de máquina *Caffe* (JIA et al., 2014), consegue resultados a 20*fps*.

Também destaca-se o tempo de execução da etapa de calibração dos parâmetros extrínsecos das câmeras; enquanto que o processo de calibração de câmeras no ambiente apresenta-se na academia como uma tarefa que demanda conhecimentos prévios de visão computacional, o algoritmo proposto o realiza de forma automática e com tempo ágil (aproximadamente 1.9%) em relação à totalidade do *pipeline*.

O algoritmo de estimação de poses humanas 3D empregado apresenta um tempo de execução bastante satisfatório; a obtenção de resultados de poses 3D a aproximadamente 67*fps* representa cerca de 0,9% do tempo de geração total do *dataset*. Pela sua natureza puramente algébrica, este algoritmo mostra-se superior em performance às técnicas centradas em minimização ou na seleção da melhor triangulação entre os conjuntos de pares de *keypoints* 2D (DONG et al., 2019).

Já a etapa de conversão das capturas de movimento geradas para formatos de *datasets* já existentes na academia apresenta-se como a de menor tempo de execução (cerca de 0,1% do tempo total), uma vez que se trata apenas da reorganização dos dados, em função do formato que se almeja alcançar; não há modificação nas informações de poses capturadas.

No geral, as etapas contempladas pelo *HuTrain*, uma vez que foram implementadas utilizando a linguagem de programação *Python*, apontam para potenciais otimizações;

[1] https://github.com/Hzzone/pytorch-openpose
[2] https://github.com/CMU-Perceptual-Computing-Lab/openpose

apesar de *Python* permitir uma maior facilidade de implementação, manipulação dos dados e prover uma gama de bibliotecas externas para uso, linguagens de programação como *C* ou *C++* apresentam tempo de execução significativamente menor[3], uma vez que estas realizam a compilação do código para um nível menor de abstração a ser executado por uma máquina.

Apesar disso, o *HuTrain* mostrou-se como uma das abordagens mais rápidas para a concepção de *datasets*. Considere a Tabela 5. Dentre os *MLDs* e *AGDs*, o tempo necessário para se produzir um *sample* com o *HuTrain* (cerca de 0.19s) é aproximadamente 47 vezes menor que o tempo gasto para produzir um *sample* do *dataset LSP Extended* (cerca de 9s, sendo assim o segundo mais rápido dentre os *MLDs* e *AGDs*) (JOHNSON; EVERINGHAM, 2011). Dentre os *MLDs* e *AGDs* cujo tempo de concepção foi disponibilizado ou obtido atráves do contato com os autores, apenas o gerado a partir do *HuTrain* encaixa-se como sendo um *AGD*. Com isso, evidencia-se a diferença de tempo entre realizar as anotações manualmente e algoritmicamente. Além disso, os *AGDs* existentes na academia (cujos tempos de concepção não foram disponibilizados) requerem que os usuários envolvidos na concepção dos *datasets* possuam conhecimentos técnicos na área de estimação de poses humanas e calibração de câmeras.

Já em relação aos *SGDs*, nota-se uma discrepância nos tempos necessários para a geração de um *sample*. Esta discrepância é consequência da complexidade dos métodos empregados para realizar a renderização dos *samples*. Para exemplificar, Tome et al. (TOME et al., 2019) buscou como prioridade maximizar o fotorrealismo dos *samples*, e para isso utilizou o V-Ray[4], um software de renderização o qual apresenta técnicas avançadas de iluminação global e de mapeamento, sendo bastante utilizado no contexto cinematográfico e desenvolvimento de jogos. Em paralelo ao tempo médio de geração de 4 minutos por *sample* em Tome et al., Chen et al. (CHEN et al., 2016) opta por gerar amostras visualmente menos realistas, porém de forma mais rápida; ao concatenar o modelo de corpo virtual texturizado a uma imagem capturada previamente (esta sendo o *background*), Chen et al. foi capaz de realizar sínteses a uma taxa de aproximadamente 30 *samples* por segundo. Apesar de o *HuTrain* mostrar-se como uma proposta mais rápida, quando comparado à maioria dos *SGDs*, a comparação não é direta, devido às suas diferentes propostas: enquanto que o *HuTrain* (junto aos *MLDs* e *AGDs*) focam na geração das anotações das poses do *dataset*, os *SGDs* consistem em sintetizar representações visuais a partir de anotações de poses obtidas previamente.

5.2 ANÁLISE DOS RESULTADOS QUANTITATIVOS

Primeiramente, justifica-se a escolha de avaliação da qualidade das poses humanas 3D obtidas pelo *HuTrain* a partir do erro de reprojeção em relação às poses humanas quando

[3] https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/cpp.html
[4] https://www.chaosgroup.com/pt/vray/sketchup

projetadas em relação aos pontos de vista das câmeras. A principal motivação refere-se à precisão das anotações manuais: o processo de anotação de *keypoints* em 3D demanda maior capacidade por parte do anotador, quando comparado com a anotação de *keypoints* em 2D (JOHNSON; EVERINGHAM, 2011; BOURDEV et al., 2010). Além disso, a seleção do conjunto de imagens para a avaliação quantitativa deu-se considerando que os grupos de imagens pertencem aos mesmos *frames*, porém advindo de câmeras diferentes. Com isso, a avaliação leva em consideração a qualidade da reprojeção das poses 3D em todos os pontos de vistas das câmeras, sendo assim uma avaliação implícita da qualidade das poses humanas em três dimensões (LEPETIT; FUA, 2005; BRACHMANN et al., 2016).

Considerando a Tabela 6, primeiramente compara-se os resultados da utilização do *HRNet + YOLOv3* (SUN et al., 2019; REDMON; FARHADI, 2018) com o *OpenPose* (CAO et al., 2017). Observa-se que as poses geradas pelo *OpenPose* são melhores que as poses geradas pelo *HRNet + YOLOv3* em praticamente todos os tipos de *keypoints* do corpo, com exceção dos *keypoints* pertencentes à face (nariz, olhos e orelhas). Isso se dá pelo fato de que estes *keypoints* ocupam espaços menores na imagem, portanto geram *heatmaps* de localização das partes do corpo com menor confiança. Enquanto que o *OpenPose* estima as poses recebendo a imagem completa como *input*, o *HRNet* utiliza como entrada o resultado da detecção (*bounding-box*) realizada pelo *YOLOv3*. Com isso, O *HRNet* lida com conteúdos visuais que desconsideram informações não referentes a humanos.

Ainda considerando a Tabela 6, pode-se observar que os resultados de poses do *HuTrain* são superiores aos obtidos pelo *OpenPose* e *HRNet + YOLOv3* em aproximadamente todos os tipos de *keypoints*, com exceção do pescoço e pélvis (estes sendo superados pelo *OpenPose*). No caso destes dois *keypoints*, os resultados podem ser justificados por dois motivos. O primeiro refere-se à qualidade das anotações manuais realizadas para estas partes, uma vez que são regiões do corpo as quais ocupam espaço significativo de corpo, além de que o ponto (*pixel*) representativo da localização destas juntas pode ser incerto (por exemplo, centro ou extremidades). O segundo motivo refere-se à facilidade do *OpenPose* em localizar estes *keypoints*, uma vez que estes *keypoints* não sofrem perdas de qualidade em função da orientação do indivíduo em relação à câmera (isto é, de frente, de lado ou de costas). Além disso, tais *keypoints* são de menor mobilidade, quando comparados a outros *keypoints*, tais como pulsos, cotovelos, joelhos e tornozelos. Não obstante, os resultados obtidos pelo *OpenPose* nestes dois tipos de *keypoints* não são consideravelmente superiores aos do *HuTrain*.

Vale ressaltar os resultados oriundos do *HuTrain* os quais são consideravelmente superiores aos das duas abordagens de *HPE* citadas, no tocante aos *keypoints* da cabeça (isto é, nariz, olhos e orelhas). Neste caso, o *HuTrain* evidencia que, a partir do uso de múltiplos pontos de vista, é possível extrair as informações de localização de *keypoints* mais confiáveis, mesmo em cenários de *self-occlusion* por alguns pontos de vista. Os elevados erros obtidos pelo *OpenPose* em relação à localização destes *keypoints* são ocasionados

pelo insucesso nos casos onde o indivíduo está de costas para um subconjunto dos pontos de vista, algo que neste cenário sempre ocorrerá, uma vez que existem pares de pontos de vista com orientações opostas.

Apesar de as comparações entre o *HuTrain* e as abordagens de *HPE* apresentarem uma desigualdade na quantidade de pontos de vista utilizados, mostra-se que o conjunto de poses humanas gerado pelo *HuTrain* pode enriquecer o processo de treinamento destes algoritmos, uma vez que o *HuTrain* é capaz de prover resultados superiores a partir da gravação de movimentos utilizando um conjunto de câmeras. Especula-se que, apesar de o *HuTrain* gerar *datasets* de acordo com as necessidades de poses do usuário, os conteúdos visuais gerados não necessariamente possuem natureza diversificada, uma vez que, assim como em outros *AGDs* presentes na academia, as gravações podem ser realizadas em um ambiente com conteúdo de *background* fixo, bem como a quantidade de atores envolvidos é limitada (IONESCU et al., 2014; JOO et al., 2015; RHODIN et al., 2016; BERCLAZ et al., 2011; SIGAL; BALAN; BLACK, 2010). Com isso, enxerga-se um *pipeline* no qual, ao alinhar o *HuTrain* a uma abordagem de geração de *SGD*, seja possível gerar, a partir de conjuntos de gravações de um ambiente, *datasets* para *HPE* os quais sejam diversificados nos contextos de poses humanas (contemplados pelo *HuTrain*) e no contexto visual (contemplados por uma abordagem de concepção de *SGDs*).

5.3 ANÁLISE DOS TESTES DE USABILIDADE

Considerando a avaliação realizada com os usuários a respeito da usabilidade do *HuTrain* (Seção 4.1), são analisados os resultados obtidos atráves da aplicação dos questionários de usabilidade *UEQ* e *SUS*, descritos nas Subseções 4.6.1 e 4.6.2. Vale ressaltar que as pontuações referentes ao *UEQ* são dispostas em um intervalo de -3 a 3, enquanto que as pontuações do *SUS* variam em um intervalo de 0 a 100.

Em relação à facilidade de utilização do *HuTrain* pelos usuários, foca-se em observar os resultados do *UEQ* referentes ao aspecto de perspicuidade (*perspicuity*). Neste caso, o *HuTrain* atingiu uma pontuação média de 2.42, sendo assim classificado pelo *UEQ* como um sistema excelente nesse aspecto. Já em relação ao *SUS*, a facilidade de uso do *framework* pode ser analisada a partir dos resultados das questões 2 e 3, nos quais o *HuTrain* atingiu as pontuações de 83.3 e 91.7, respectivamente. Estes resultados são justificados pelo *HuTrain* apresentar-se como um *framework* sucinto; uma vez realizadas as gravações, o *pipeline* é capaz de realizar todo o processo de geração do *dataset* de forma totalmente automática a partir de *scripts*, dispensando assim a necessidade de intervenção do usuário. Além disso, este resultado reflete o nível de clareza em relação às instruções dispostas pelo *README* do *framework*, o qual descreve os passos a serem seguidos de forma linear. Por outro lado, os resultados da questões 4 e 10 do *SUS* sutilmente denotam desafios referentes à independência dos usuários em relação à utilização do *framework*.

Também foi analisado o grau de esforço investido pelos usuários. Para tal, analisa-se os resultados obtidos a partir do aspecto de eficiência (*efficiency*) proposto pelo *UEQ*. Neste aspecto, a pontuação de 2.08 também classifica o *HuTrain* como sendo um sistema excelente. As altas pontuações referentes aos itens de organização e eficiência são refletidas na correspondência entre o *pipeline* proposto e a sua implementação; cada etapa intermediária do método é descrita na instruções de execução, bem como os resultados gerados. Em relação ao *SUS*, estas características são refletidas nos resultados das questões 5 a 8, possuindo as pontuações de 91.7, 91.7, 75 e 83.3, respectivamente. Como consequência, todos os usuários envolvidos sugeriram a existência de um comando único, o qual agregasse a execução de todas as funcionalidades linearmente em uma única interação.

Vale ressaltar que os usuários envolvidos na avaliação possuem experiência em computação, porém nunca estiveram envolvidos em um processo de concepção de *datasets* para estimação de poses humanas ou utilização de ferramentas de *MoCap*. Com isso, buscou-se avaliar a confiança dos usuários em relação à utilização do *HuTrain*. Considerando o *UEQ*, analisa-se o aspecto de confiabilidade (*dependability*). Neste, observa-se uma pontuação de 2.00, caracterizando-o como um sistema excelente nesse aspecto. Vale ressaltar as elevadas pontuações nos itens relacionados a auxílio (*obstructive/supportive*) e segurança (*not secure/secure*); isso é justificado pelo fato de que, ao passo que o usuário realiza as etapas do *HuTrain*, as consequências são descritas de forma roteirizada pelo README. Já em relação ao *SUS*, a característica de confiança é refletida na questão 9, possuindo uma pontuação de 66.6, o que representa um nível de confiança regular por parte dos usuários.

Além dos resultados pragmáticos de usabilidade, também foram avaliados aspectos relacionados à satisfação dos usuários ao utilizarem o *framework*. Para tal, analisou-se os resultados referentes ao aspecto de atratividade (*attractiveness*) do *UEQ*. Neste caso, a pontuação média atingida foi de 1.39, o que caracteriza o *HuTrain* como um sistema acima da média nesse aspecto. Esta pontuação de nível mediano é refletida pelo fato de que o *HuTrain* não possui uma interface gráfica a qual guie o usuário no processo de geração de um *dataset* de forma interativa. Com isso, as pontuações de itens referentes a agradabilidade (*annoying/enjoyable*, *unlikable/pleasing* e *unpleasant/pleasant*) e atratividade (*unattractive/attractive*) posicionam-se entre 0 e 1. Já em relação à avaliação do *SUS*, pode-se observar o resultado da questão 1, o qual atingiu uma pontuação média de 58.3, sendo assim um resultado abaixo da média descrita pelo *SUS* (considerando apenas esta questão). Após a avaliação, os usuários sugeriram a implementação de uma interface gráfica simples mínima, cujo propósito seja a substituição da execução de comandos diretamente por botões, os quais realizem a mesma tarefa. Além disso, também foram sugeridas informações de progresso para a execução de algumas das etapas do *HuTrain*.

6 CONCLUSÕES

Nesta dissertação, foi estudada a problemática referente à concepção de *datasets* focados na tarefa de estimação de poses humanas a partir de imagens RGB. A partir da revisão da literatura, compreende-se os avanços dos algoritmos de estimação de poses humanas através da utilização de técnicas de redes neurais convolucionais, bem como dos métodos empregados para a geração de *datasets* com o intuito de alimentar o *pipeline* de treinamento dos modelos de *CNNs* propostos pela academia.

A partir da compreensão dos desafios referentes à geração de *datasets* para estimação de poses humanas e da necessidade de contemplar contextos de poses específicos aos cenários de uso, este trabalho propôs o *HuTrain*: um framework semi-automático com o objetivo de conceber, de forma prática e rápida, *datasets* para a tarefa de estimação de poses humanas. Para tal, o *HuTrain* descreve um *pipeline* que busca automatizar o máximo de etapas possíveis referentes à geração de um *dataset* para *HPE*. Com isso, o *HuTrain* cobre as etapas de estimação de poses humanas 2D para cada uma das câmeras (pontos de vista) utilizadas, a calibração das câmeras (parâmetros extrínsecos) no ambiente, a estimação de poses humanas 3D a partir das informações de poses bidimensionais sob diferentes pontos de vista e a conversão das capturas de movimento geradas para formatos de *datasets* amplamente utilizados por técnicas de estimação de poses humanas presentes no estado da arte.

Com isso, o *HuTrain* foi avaliado considerando três aspectos: o tempo levado para realizar a geração do *dataset*, a qualidade das poses geradas e a facilidade de uso do *framework*. Em relação ao tempo de execução, o *HuTrain* apresenta-se como o método mais rápido para a geração de anotações de poses humanas. Já em relação às anotações do *dataset* gerado, estas demonstram qualidade superior às anotações geradas por estimadores de poses humanas do estado da arte e podem ser utilizadas no incremento do treinamento dos modelos para *HPE*. Por fim, a avaliação de usabilidade realizada com potenciais usuários demonstra uma facilidade no uso do *framework*, porém os resultados indicam possíveis melhorias em relação à interface do *framework*.

Dentre os trabalhos futuros, busca-se utilizar o *dataset* gerado diretamente em um *pipeline* de treinamento de modelos de estimações de poses humanas em contextos de poses específicos. Já em relação à implementação do método, sua modularização torna possível testes, por diferentes abordagens, para cada uma das etapas do *pipeline* proposto, a fim de incrementar o resultado final das poses geradas. Por fim, busca-se realizar o teste de novas interfaces, a fim de tornar o processo de geração de *datasets* potencialmente mais amigável para usuários que não possuam experiência em computação.

REFERÊNCIAS

ALCANTARILLA, P. F.; BARTOLI, A.; DAVISON, A. J. Kaze features. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2012. p. 214–227.

ALCANTARILLA, P. F.; SOLUTIONS, T. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, TrueVision Solutions, v. 34, n. 7, p. 1281–1298, 2011.

ANDRILUKA, M.; PISHCHULIN, L.; GEHLER, P.; SCHIELE, B. 2d human pose estimation: New benchmark and state of the art analysis. In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 3686–3693.

ANGUELOV, D.; SRINIVASAN, P.; KOLLER, D.; THRUN, S.; RODGERS, J.; DAVIS, J. Scape: shape completion and animation of people. In: *ACM SIGGRAPH 2005 Papers*. [S.l.: s.n.], 2005. p. 408–416.

BANGOR, A.; KORTUM, P.; MILLER, J. The system usability scale (sus): An empirical evaluation. *International Journal of Human-Computer Interaction*, v. 24, n. 6, p. 574–594, 2008.

BANGOR, A.; KORTUM, P.; MILLER, J. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, Citeseer, v. 4, n. 3, p. 114–123, 2009.

BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: SPRINGER. *European conference on computer vision*. [S.l.], 2006. p. 404–417.

BELAGIANNIS, V.; AMIN, S.; ANDRILUKA, M.; SCHIELE, B.; NAVAB, N.; ILIC, S. 3d pictorial structures for multiple human pose estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 1669–1676.

BERCLAZ, J.; FLEURET, F.; TURETKEN, E.; FUA, P. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 33, n. 9, p. 1806–1819, 2011.

BOURDEV, L.; MAJI, S.; BROX, T.; MALIK, J. Detecting people using mutually consistent poselet activations. In: SPRINGER. *European conference on computer vision*. [S.l.], 2010. p. 168–181.

BOURDEV, L.; MALIK, J. Poselets: Body part detectors trained using 3d human pose annotations. In: IEEE. *2009 IEEE 12th International Conference on Computer Vision*. [S.l.], 2009. p. 1365–1372.

BRACHMANN, E.; MICHEL, F.; KRULL, A.; YANG, M. Y.; GUMHOLD, S. et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 3364–3372.

BRADSKI, G.; KAEHLER, A. Opencv. *Dr. Dobb's journal of software tools*, v. 3, 2000.

BROOKE, J. Sus: a “quick and dirty’usability. *Usability evaluation in industry*, CRC press, p. 189, 1996.

CAO, Z.; SIMON, T.; WEI, S.-E.; SHEIKH, Y. Realtime multi-person 2d pose estimation using part affinity fields. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 7291–7299.

CARREIRA, J.; AGRAWAL, P.; FRAGKIADAKI, K.; MALIK, J. Human pose estimation with iterative error feedback. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 4733–4742.

CHARLES, J.; PFISTER, T.; EVERINGHAM, M.; ZISSERMAN, A. Automatic and efficient human pose estimation for sign language videos. *International Journal of Computer Vision*, Springer, v. 110, n. 1, p. 70–90, 2014.

CHARLES, J.; PFISTER, T.; MAGEE, D.; HOGG, D.; ZISSERMAN, A. Domain adaptation for upper body pose tracking in signed tv broadcasts. British Machine Vision Association, 2013.

CHARLES, J.; PFISTER, T.; MAGEE, D.; HOGG, D.; ZISSERMAN, A. Personalizing human video pose estimation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 3063–3072.

CHEN, C.-H.; RAMANAN, D. 3d human pose estimation = 2d pose estimation + matching. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 7035–7043.

CHEN, W.; WANG, H.; LI, Y.; SU, H.; WANG, Z.; TU, C.; LISCHINSKI, D.; COHEN-OR, D.; CHEN, B. Synthesizing training images for boosting human 3d pose estimation. In: IEEE. *2016 Fourth International Conference on 3D Vision (3DV)*. [S.l.], 2016. p. 479–488.

CHU, X.; YANG, W.; OUYANG, W.; MA, C.; YUILLE, A. L.; WANG, X. Multi-context attention for human pose estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 1831–1840.

CUNHA, K. B. d. *Detecção de objetos em 6-DoF em tempo real utilizando técnicas de aprendizagem profunda*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2019.

DONG, J.; JIANG, W.; HUANG, Q.; BAO, H.; ZHOU, X. Fast and robust multi-person 3d pose estimation from multiple views. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 7792–7801.

ELHAYEK, A.; AGUIAR, E. de; JAIN, A.; THOMPSON, J.; PISHCHULIN, L.; ANDRILUKA, M.; BREGLER, C.; SCHIELE, B.; THEOBALT, C. Marconi—convnet-based marker-less motion capture in outdoor and indoor scenes. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 39, n. 3, p. 501–514, 2016.

EVERINGHAM, M.; GOOL, L. V.; WILLIAMS, C. K.; WINN, J.; ZISSERMAN, A. The pascal visual object classes (voc) challenge. *International journal of computer vision*, Springer, v. 88, n. 2, p. 303–338, 2010.

FABBRI, M.; LANZI, F.; CALDERARA, S.; PALAZZI, A.; VEZZANI, R.; CUCCHIARA, R. Learning to detect and track visible and occluded body joints in a virtual world. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 430–446.

GE, L.; REN, Z.; LI, Y.; XUE, Z.; WANG, Y.; CAI, J.; YUAN, J. 3d hand shape and pose estimation from a single rgb image. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 10833–10842.

GOLUB, G. H.; REINSCH, C. Singular value decomposition and least squares solutions. In: *Linear Algebra*. [S.l.]: Springer, 1971. p. 134–151.

GONG, K.; LIANG, X.; ZHANG, D.; SHEN, X.; LIN, L. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 932–940.

GREEN, R. Spherical harmonic lighting: The gritty details. In: *Archives of the Game Developers Conference*. [S.l.: s.n.], 2003. v. 56, p. 4.

GROUP, C. *V-Ray*. 1997. Disponível em: <https://www.chaosgroup.com/pt>.

GüLER, R. A.; NEVEROVA, N.; KOKKINOS, I. Densepose: Dense human pose estimation in the wild. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 7297–7306.

HARTLEY, R.; ZISSERMAN, A. *Multiple view geometry in computer vision*. [S.l.]: Cambridge university press, 2003.

HASLER, N.; STOLL, C.; SUNKEL, M.; ROSENHAHN, B.; SEIDEL, H.-P. A statistical model of human pose and body shape. In: WILEY ONLINE LIBRARY. *Computer graphics forum*. [S.l.], 2009. v. 28, n. 2, p. 337–346.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.

INC., A. *Mixamo*. 2008. Disponível em: <https://www.mixamo.com/>.

IONESCU, C.; PAPAVA, D.; OLARU, V.; SMINCHISESCU, C. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 36, n. 7, p. 1325–1339, 2014.

IQBAL, U.; MILAN, A.; GALL, J. Posetrack: Joint multi-person pose estimation and tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 2011–2020.

ISKAKOV, K.; BURKOV, E.; LEMPITSKY, V.; MALKOV, Y. Learnable triangulation of human pose. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 7718–7727.

JHUANG, H.; GALL, J.; ZUFFI, S.; SCHMID, C.; BLACK, M. J. Towards understanding action recognition. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2013. p. 3192–3199.

JI, Y.; CHANG, C. C. Vision-based sensing for assessing and monitoring civil infrastructures. In: *Sensor Technologies for Civil Infrastructures*. [S.l.]: Elsevier, 2014. p. 383–409.

JIA, Y.; SHELHAMER, E.; DONAHUE, J.; KARAYEV, S.; LONG, J.; GIRSHICK, R.; GUADARRAMA, S.; DARRELL, T. Caffe: Convolutional architecture for fast feature embedding. In: *Proceedings of the 22nd ACM international conference on Multimedia*. [S.l.: s.n.], 2014. p. 675–678.

JOHNSON, S.; EVERINGHAM, M. Clustered pose and nonlinear appearance models for human pose estimation. In: *BMVC*. [S.l.: s.n.], 2010. v. 2, n. 4, p. 5.

JOHNSON, S.; EVERINGHAM, M. Learning effective human pose estimation from inaccurate annotation. In: IEEE. *CVPR 2011*. [S.l.], 2011. p. 1465–1472.

JOO, H.; LIU, H.; TAN, L.; GUI, L.; NABBE, B.; MATTHEWS, I.; KANADE, T.; NOBUHARA, S.; SHEIKH, Y. Panoptic studio: A massively multiview system for social motion capture. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2015. p. 3334–3342.

KOENDERINK, J. J.; DOORN, A. J. V. Affine structure from motion. *JOSA A*, Optical Society of America, v. 8, n. 2, p. 377–385, 1991.

LASSNER, C.; ROMERO, J.; KIEFEL, M.; BOGO, F.; BLACK, M. J.; GEHLER, P. V. Unite the people: Closing the loop between 3d and 2d human representations. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 6050–6059.

LAUGWITZ, B.; HELD, T.; SCHREPP, M. Construction and evaluation of a user experience questionnaire. In: SPRINGER. *Symposium of the Austrian HCI and usability engineering group*. [S.l.], 2008. p. 63–76.

LEITE, H. P. B.

WRITEME: uma Ferramenta de Auxílio à Escrita de READMEs Baseada em Dados, 2019.

LEPETIT, V.; FUA, P. *Monocular model-based 3D tracking of rigid objects*. [S.l.]: Now Publishers Inc, 2005.

LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, v. 2, n. 2, p. 164–168, 1944.

LI, S.; CHAN, A. B. 3d human pose estimation from monocular images with deep convolutional neural network. In: SPRINGER. *Asian Conference on Computer Vision*. [S.l.], 2014. p. 332–347.

LIN, J.; O'BRIEN, J. *Temporal IK: Data-Driven Pose Estimation for Virtual Reality*. [S.l.], 2019.

LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; BOURDEV, L.; GIRSHIK, R.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft coco: Common objects in context. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 740–755.

LONGUET-HIGGINS, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature*, Springer, v. 293, n. 5828, p. 133–135, 1981.

LOPER, M.; MAHMOOD, N.; ROMERO, J.; PONS-MOLL, G.; BLACK, M. J. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, ACM New York, NY, USA, v. 34, n. 6, p. 1–16, 2015.

LOWE, D. G. Object recognition from local scale-invariant features. In: IEEE. *Proceedings of the seventh IEEE international conference on computer vision*. [S.l.], 1999. v. 2, p. 1150–1157.

MARCARD, T. V.; PONS-MOLL, G.; ROSENHAHN, B. Human pose estimation from video and imus. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 38, n. 8, p. 1533–1547, 2016.

MARCARD, T. von; HENSCHEL, R.; BLACK, M. J.; ROSENHAHN, B.; PONS-MOLL, G. Recovering accurate 3d human pose in the wild using imus and a moving camera. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 601–617.

MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, SIAM, v. 11, n. 2, p. 431–441, 1963.

MARTINEZ, J.; HOSSAIN, R.; ROMERO, J.; LITTLE, J. J. A simple yet effective baseline for 3d human pose estimation. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2017. p. 2640–2649.

MCCORMAC, J.; HANDA, A.; LEUTENEGGER, S.; DAVISON, A. J. Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. *arXiv preprint arXiv:1612.05079*, 2016.

MEHTA, D.; RHODIN, H.; CASAS, D.; FUA, P.; SOTNYCHENKO, O.; XU, W.; THEOBALT, C. Monocular 3d human pose estimation in the wild using improved cnn supervision. In: IEEE. *2017 International Conference on 3D Vision (3DV)*. [S.l.], 2017. p. 506–516.

MEHTA, D.; SOTNYCHENKO, O.; MUELLER, F.; XU, W.; SRIDHAR, S.; PONS-MOLL, G.; THEOBALT, C. Single-shot multi-person 3d pose estimation from monocular rgb. In: IEEE. *2018 International Conference on 3D Vision (3DV)*. [S.l.], 2018. p. 120–130.

MEHTA, D.; SRIDHAR, S.; SOTNYCHENKO, O.; RHODIN, H.; SHAFIEI, M.; SEIDEL, H.-P.; XU, W.; CASAS, D.; THEOBALT, C. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, ACM, v. 36, n. 4, p. 44, 2017.

MORI, G.; MALIK, J. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 28, n. 7, p. 1052–1062, 2006.

NEWELL, A.; YANG, K.; DENG, J. Stacked hourglass networks for human pose estimation. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2016. p. 483–499.

NISTÉR, D. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 26, n. 6, p. 756–770, 2004.

NORTH, R. et al. Grand theft auto v. *New York: Rockstar Games*, 2013.

OLIPHANT, T. E. *A guide to NumPy*. [S.l.]: Trelgol Publishing USA, 2006. v. 1.

PASZKE, A.; GROSS, S.; CHINTALA, S.; CHANAN, G. Pytorch. *Computer software. Vers. 0.3*, v. 1, 2017.

PFISTER, T.; SIMONYAN, K.; CHARLES, J.; ZISSERMAN, A. Deep convolutional neural networks for efficient pose estimation in gesture videos. In: SPRINGER. *Asian Conference on Computer Vision*. [S.l.], 2014. p. 538–552.

PISHCHULIN, L.; ANDRILUKA, M.; GEHLER, P.; SCHIELE, B. Poselet conditioned pictorial structures. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2013. p. 588–595.

PISHCHULIN, L.; JAIN, A.; ANDRILUKA, M.; THORMÄHLEN, T.; SCHIELE, B. Articulated people detection and pose estimation: Reshaping the future. In: IEEE. *2012 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.], 2012. p. 3178–3185.

RAMAKRISHNA, V.; MUNOZ, D.; HEBERT, M.; BAGNELL, J. A.; SHEIKH, Y. Pose machines: Articulated pose estimation via inference machines. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2014. p. 33–47.

RAMANAN, D. Learning to parse images of articulated bodies. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2007. p. 1129–1136.

REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

REN, X.; BERG, A. C.; MALIK, J. Recovering human body configurations using pairwise constraints between parts. In: IEEE. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. [S.l.], 2005. v. 1, p. 824–831.

RHODIN, H.; RICHARDT, C.; CASAS, D.; INSAFUTDINOV, E.; SHAFIEI, M.; SEIDEL, H.-P.; SCHIELE, B.; THEOBALT, C. Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)*, ACM, v. 35, n. 6, p. 162, 2016.

RHODIN, H.; SALZMANN, M.; FUA, P. Unsupervised geometry-aware representation for 3d human pose estimation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 750–767.

ROBINETTE, K. M.; BLACKWELL, S.; DAANEN, H.; BOEHMER, M.; FLEMING, S. *Civilian american and european surface anthropometry resource (caesar), final report. volume 1. summary*. [S.l.], 2002.

ROHRBACH, M.; AMIN, S.; ANDRILUKA, M.; SCHIELE, B. A database for fine grained activity detection of cooking activities. In: IEEE. *2012 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.], 2012. p. 1194–1201.

ROSSUM, G. Python reference manual. CWI (Centre for Mathematics and Computer Science), 1995.

RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. Orb: An efficient alternative to sift or surf. In: IEEE. *2011 International conference on computer vision*. [S.l.], 2011. p. 2564–2571.

SAPP, B.; TASKAR, B. Modec: Multimodal decomposable models for human pose estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2013. p. 3674–3681.

SCHONBERGER, J. L.; FRAHM, J.-M. Structure-from-motion revisited. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 4104–4113.

SCHULDT, C.; LAPTEV, I.; CAPUTO, B. Recognizing human actions: a local svm approach. In: IEEE. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* [S.l.], 2004. v. 3, p. 32–36.

SHEN, H.; YU, S.-I.; YANG, Y.; MENG, D.; HAUPTMANN, A. Unsupervised video adaptation for parsing human motion. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2014. p. 347–360.

SIGAL, L.; BALAN, A. O.; BLACK, M. J. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, Springer, v. 87, n. 1-2, p. 4, 2010.

SIMÕES, F.; ALMEIDA, M.; PINHEIRO, M.; ANJOS, R. D.; SANTOS, A. D.; ROBERTO, R.; TEICHRIEB, V.; SUETSUGO, C.; PELINSON, A. Challenges in 3d reconstruction from images for difficult large-scale objects: A study on the modeling of electrical substations. In: IEEE. *2012 14th Symposium on Virtual and Augmented Reality*. [S.l.], 2012. p. 74–83.

SUN, K.; XIAO, B.; LIU, D.; WANG, J. Deep high-resolution representation learning for human pose estimation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 5693–5703.

SUN, X.; XIAO, B.; WEI, F.; LIANG, S.; WEI, Y. Integral human pose regression. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 529–545.

TAYLOR, C. J. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, Elsevier, v. 80, n. 3, p. 349–363, 2000.

TEKIN, B.; BOGO, F.; POLLEFEYS, M. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 4511–4520.

TOME, D.; PELUSE, P.; AGAPITO, L.; BADINO, H. xr-egopose: Egocentric 3d human pose from an hmd camera. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 7728–7738.

TOMPSON, J.; GOROSHIN, R.; JAIN, A.; LECUN, Y.; BREGLER, C. Efficient object localization using convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 648–656.

TOMPSON, J. J.; JAIN, A.; LECUN, Y.; BREGLER, C. Joint training of a convolutional network and a graphical model for human pose estimation. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2014. p. 1799–1807.

TOSHEV, A.; SZEGEDY, C. Deeppose: Human pose estimation via deep neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2014. p. 1653–1660.

TOYODA, K.; KONO, M.; REKIMOTO, J. Post-data augmentation to improve deep pose estimation of extreme and wild motions. In: IEEE. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. [S.l.], 2019. p. 1570–1574.

TRAN, D.; FORSYTH, D. Improved human parsing with a full relational model. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2010. p. 227–240.

TRIGGS, B.; MCLAUCHLAN, P. F.; HARTLEY, R. I.; FITZGIBBON, A. W. Bundle adjustment—a modern synthesis. In: SPRINGER. *International workshop on vision algorithms*. [S.l.], 1999. p. 298–372.

TRUMBLE, M.; GILBERT, A.; MALLESON, C.; HILTON, A.; COLLOMOSSE, J. Total capture: 3d human pose estimation fusing video and inertial sensors. In: *BMVC*. [S.l.: s.n.], 2017. v. 2, p. 3.

ULLMAN, S. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, The Royal Society London, v. 203, n. 1153, p. 405–426, 1979.

VAROL, G.; ROMERO, J.; MARTIN, X.; MAHMOOD, N.; BLACK, M. J.; LAPTEV, I.; SCHMID, C. Learning from synthetic humans. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 109–117.

WANG, Y.; TRAN, D.; LIAO, Z. Learning hierarchical poselets for human parsing. In: IEEE. *CVPR 2011*. [S.l.], 2011. p. 1705–1712.

WEI, S.-E.; RAMAKRISHNA, V.; KANADE, T.; SHEIKH, Y. Convolutional pose machines. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 4724–4732.

XIAO, B.; WU, H.; WEI, Y. Simple baselines for human pose estimation and tracking. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 466–481.

XU, W.; CHATTERJEE, A.; ZOLLHOFER, M.; RHODIN, H.; FUA, P.; SEIDEL, H.-P.; THEOBALT, C. Mo²cap²: Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE transactions on visualization and computer graphics*, IEEE, 2019.

YANG, Y.; RAMANAN, D. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 35, n. 12, p. 2878–2890, 2013.

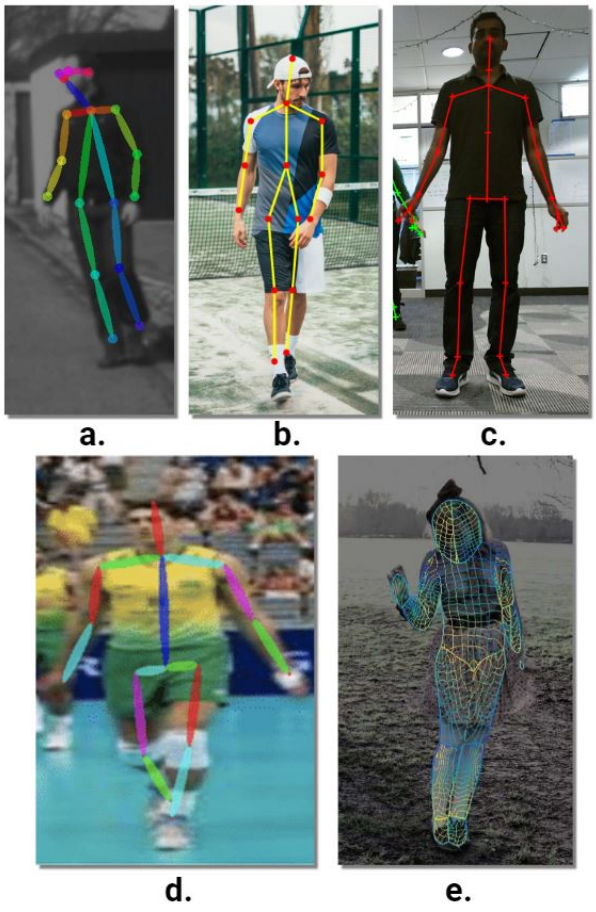
YU, F.; SEFF, A.; ZHANG, Y.; SONG, S.; FUNKHOUSER, T.; XIAO, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

ZHONG, Z.; ZHENG, L.; ZHENG, Z.; LI, S.; YANG, Y. Camera style adaptation for person re-identification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 5157–5166.

APÊNDICE A – ESTIMAÇÃO DE POSES HUMANAS

A tarefa de estimação de poses humanas (*HPE*, *Human Pose Estimation*) é uma tarefa amplamente estudada na área de Visão Computacional, cujo objetivo é localizar a posição dos pontos característicos de indivíduos presentes em um ambiente. Estes pontos característicos (os quais também chamamos de *keypoints* ou *joints*), quando relacionados, definem o que chamamos de **pose humana**. Trabalhos presentes na literatura mostram uma tendência na representação das poses humanas, a partir da seleção de *keypoints* tais como articulações, regiões centrais do corpo e da cabeça, com o intuito de se obterem representações sucintas do corpo a ponto de serem visualmente condizentes com a pose realizada pelo indivíduo e analisáveis de acordo com o contexto de uso (aplicação) das poses (Figura 31).

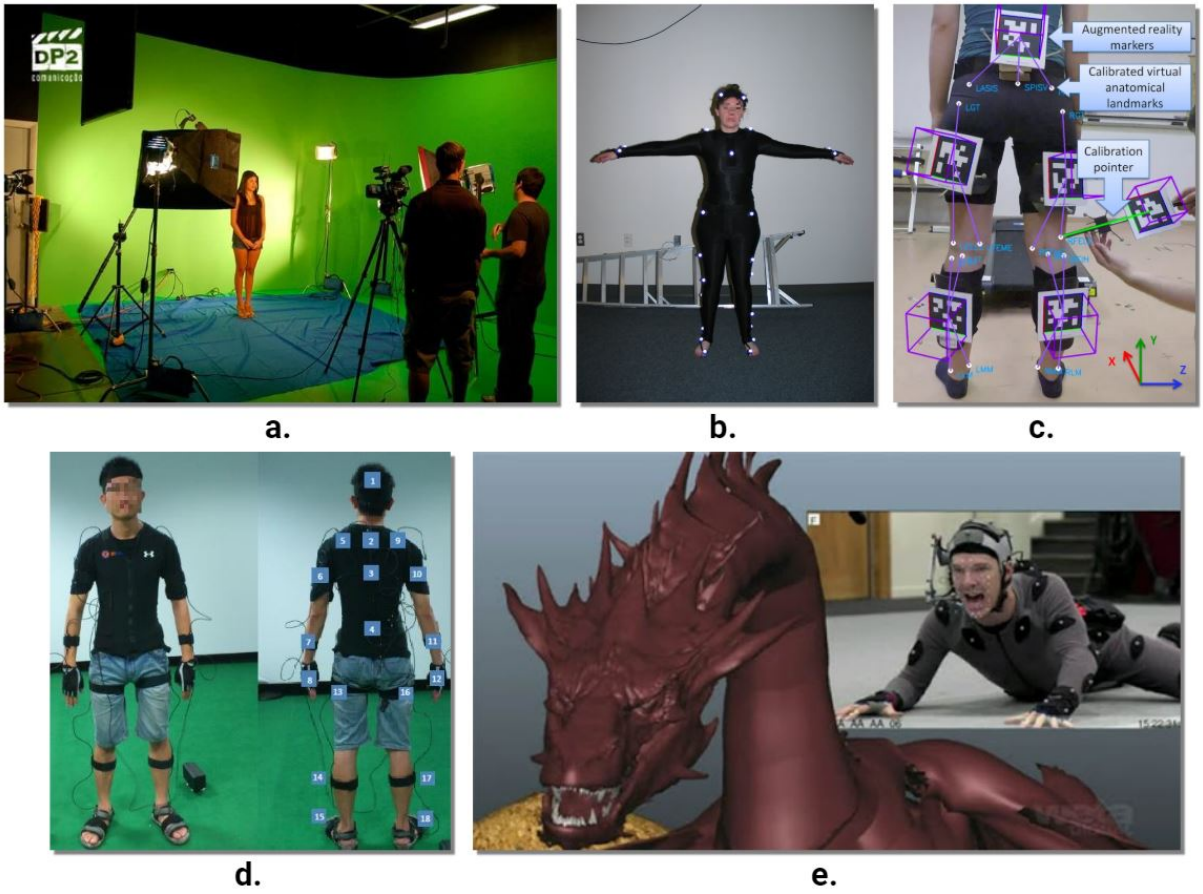
Figura 31 – Exemplos de diferentes representações de poses humanas. **a.** *MS COCO* (LIN et al., 2014); **b.** *MPII Human Pose* (ANDRILUKA et al., 2014); **c.** *Microsoft Kinect*; **d.** *LSP* (JOHNSON; EVERINGHAM, 2010; JOHNSON; EVERINGHAM, 2011); **e.** *DensePose* (GüLER; NEVEROVA; KOKKINOS, 2018).



Para realizar esta tarefa, pode-se utilizar acessórios externos aos corpos detectados,

os quais fornecem informações adicionais a fim de auxiliar no processo de estimação de poses. Estes acessórios podem trazer incrementos no contexto visual, como por exemplo a troca do fundo por uma cor sólida característica (*chroma key*, Figura 32a) ou a utilização de marcadores anexos ao corpo, os quais podem trazer dicas visuais por reflexões características (Figura 32b) ou pela presença de marcadores fiduciais ao corpo (Figura 32c). Também podem ser utilizados sensores inerciais, os quais são espalhados pelo corpo a fim de se obter diretamente a posição e fluxo de movimento do indivíduo (Figura 32d). Este último apresenta-se bastante comum no contexto de captura de movimento para a animação de avatares em filmes e jogos em alta fidelidade a partir de tecnologias vestíveis (*wearables*), as quais contemplam sensores para esta finalidade (Figura 32e).

Figura 32 – Exemplos de tecnologias de captura de movimento utilizando aparatos externos.



Por outro lado, a utilização direta de acessórios ao corpo pode trazer alterações na captura de movimento desejada, uma vez que o indivíduo rastreado pode ser enviesado a realizar movimentos de forma não natural. Com isso, há a opção de se realizar a estimação de poses humanas a partir somente de sensores visuais (câmeras), os quais isentam qualquer tipo de sensor ligado diretamente ao corpo, resultando em capturas de movimento mais naturais. Com isso, aumenta-se o conjunto de possíveis aplicações desses algoritmos,

principalmente no tocante à ubiquidade desses serviços. Nesta categoria, destacam-se duas categorias de sensores:

- **Sensores de profundidade**: São os sensores capazes de estimar a profundidade (geralmente o eixo z) de todas as posições (*pixels*) da imagem. Dessa maneira, quando um indivíduo apresenta-se em frente ao sensor, sua "silhueta" de profundidade é destacada. Soluções de mercado existentes, tais como o *Microsoft Kinect*, o *OpenNI* e o *Orbecc Astra*, buscam realizar a segmentação semântica dos indivíduos presentes a qual, aliada à informação de profundidade, torna possível a estimação de poses humanas de forma tridimensional no ambiente. Apesar de serem robustas em contextos *indoor*, estes possuem custo mais elevado e são menos acessíveis no mercado;

Figura 33 – Exemplos de sensores de profundidade para captura de movimento (imagens à esquerda), bem como os seus respectivos *outputs* (imagens à direita). De cima para baixo: *Microsoft Kinect*, *OpenNI* e *Orbecc Astra*.

- **Câmeras *RGB***: Estas são as câmeras convencionais, definidas por um sistema de três cores: vermelho, verde e azul (*Red*, *Green* e *Blue*, respectivamente) que, quando renderizados separadamente, são capazes de descrever o espectro visível de cores. Uma vez que as abordagens de *HPE* a partir de imagens *RGB* não utilizam informações de profundidade, seu uso pode trazer desafios referentes à tridimensionalidade das poses humanas. Por outro lado, estes algoritmos podem ser utilizados em contextos envolvendo *smartphones*, *notebooks*, câmeras de segurança e computadores, o que traz uma maior acesso ao uso desses algoritmos, além de se mostrarem uma solução para a realização de rastreamento em ambientes *outdoor*.

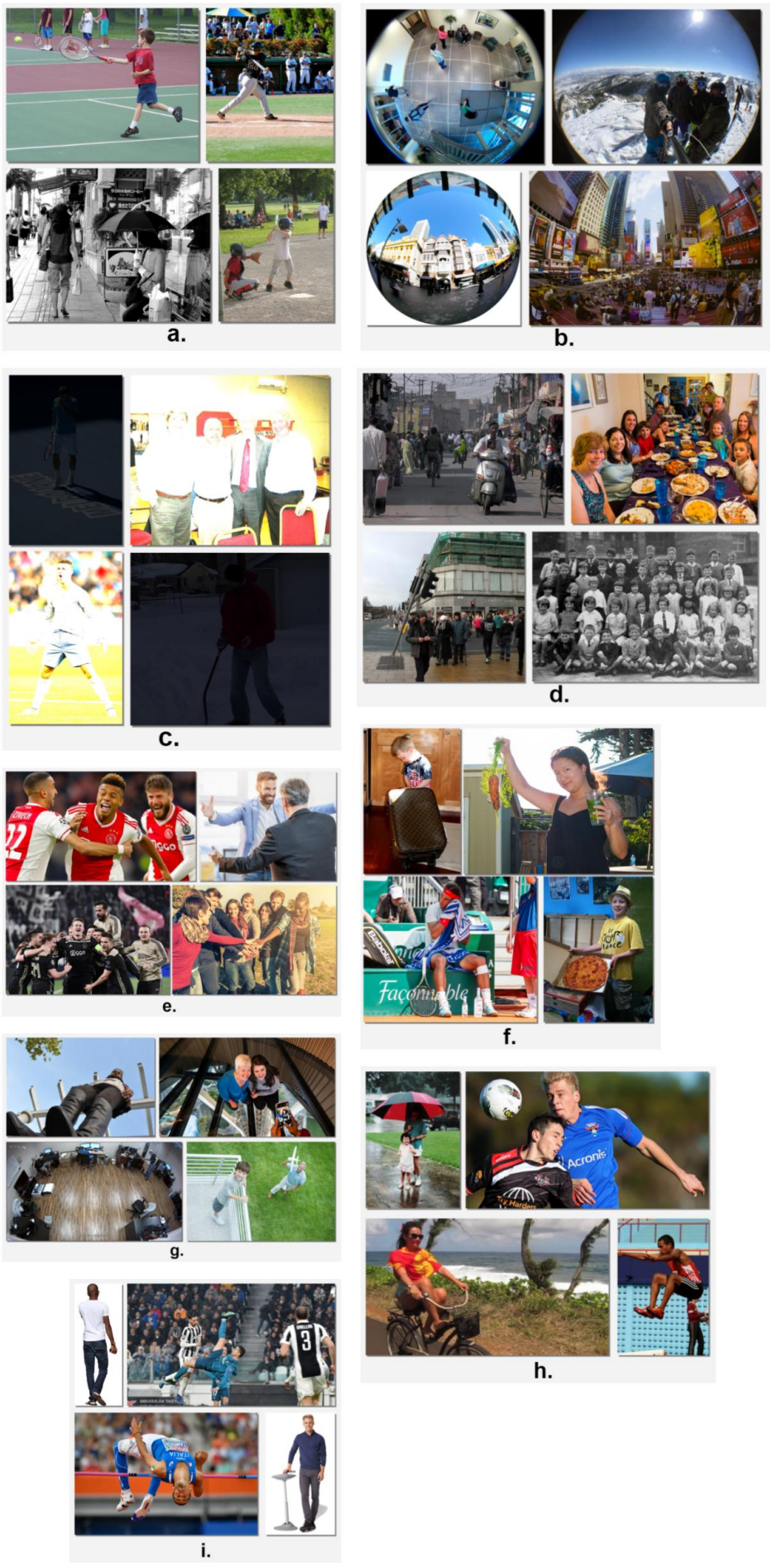
Figura 34 – Exemplos de aparatos eletrônicos contendo câmeras *RGB*.

Considerando o amplo conjunto de aplicações de estimadores de poses humanas, aumenta-se o espaço latente de conteúdos visuais os quais podem ser deparados por um estimador de pose. Com isso, surge uma série de desafios com os quais os métodos devem estar aptos a lidar, tais como:

- A variedade de perspectiva (distância) entre as pessoas (Figura 35a);
- As distorções visuais existentes em relação à câmera utilizada (Figura 35b);
- A variedade de iluminação das imagens (Figura 35c);
- A variedade da quantidade de pessoas presentes (Figura 35d);
- O conjunto de possíveis interações com outros indivíduos (Figura 35e);

- O conjunto de possíveis interações com objetos (Figura 35f);
- Os possíveis pontos de vista da imagem capturada em relação ao ambiente (Figura 35g);
- Oclusões de regiões do corpo em relação a outras entidades no ambiente, incluindo os casos de oclusões ocasionadas pelo próprio corpo (*self-occlusion*) (Figura 35h);
- O conjunto de possíveis poses que podem ser realizadas por indivíduos (Figura 35i).

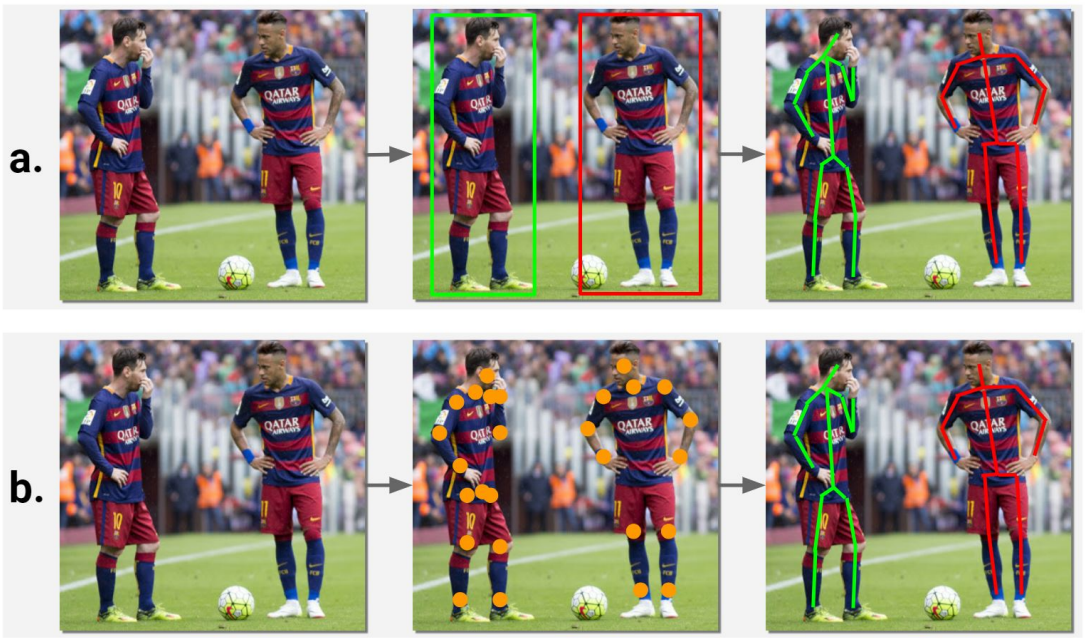
Figura 35 – Exemplos de desafios de estimações de poses humanas em imagens *RGB*.



Dentre os desafios de *HPE* listados anteriormente, vale destacar o fato de poderem haver múltiplos indivíduos no ambiente; nesse cenário, é necessário adotar-se uma estratégia para realizar a agregação dos *keypoints* localizados em poses respectivas às pessoas presentes na imagem. Para tal, existem duas possíveis estratégias:

- **Abordagens Top-down**: Nesta abordagem, utiliza-se primeiramente um algoritmo para a detecção de indivíduos na imagem. Considerando n indivíduos na imagem, o algoritmo de detecção, quando aplicado na imagem, deverá gerar n regiões da imagem (*bounding-boxes*), onde cada região delimita o espaço visual que cada indivíduo ocupa na imagem. Após isso, assume-se que cada *bounding-box* contém apenas uma pessoa, e, portanto, o algoritmo de estimação de poses humanas é aplicado em cada um dos *bounding-boxes*, nos quais os *keypoints* localizados pertencem ao indivíduo detectado na imagem (Figura 36a);
- **Abordagens Bottom-up**: Nesta abordagem, utiliza-se um algoritmo de localização de *keypoints* do corpo humano em toda a imagem. Considerando múltiplas pessoas na imagem, a presença de múltiplos *keypoints* do mesmo tipo (por exemplo, ombro esquerdo) também ocorrerá. Após esta etapa, é empregada uma estratégia de pós-processamento a fim de agregar corretamente os conjuntos de *keypoints* em suas poses reais respectivas (Figura 36b).

Figura 36 – Exemplo de estimação de poses humanas utilizando abordagens *top-down* (**a.**) e *bottom-up* (**b.**).

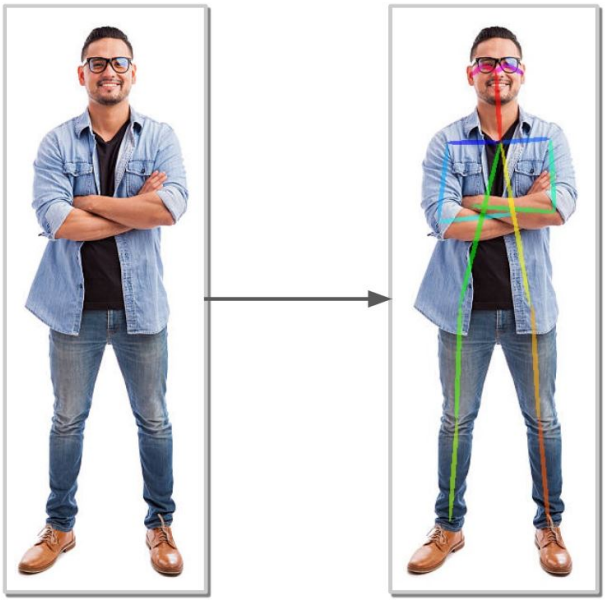


Fonte: autoral.

Em relação aos métodos de estimação de poses humanas existentes e a dimensionalidade das poses, estes dividem-se em duas categorias:

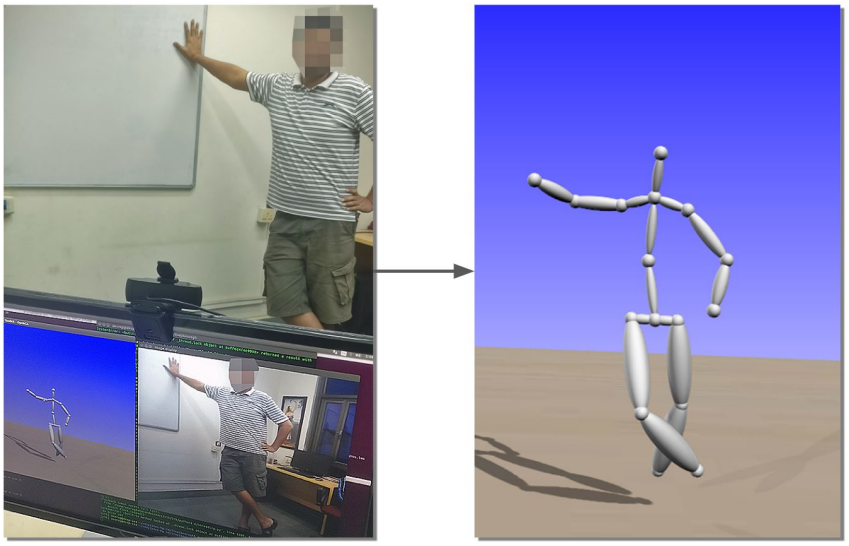
- **Estimações de Poses Humanas 2D**: Buscam localizar a posição bidimensional (x, y), em *pixels*, de cada um dos *keypoints* da pose, os quais representa a projeção da pose real na imagem de entrada (Figura 37);

Figura 37 – Exemplo de estimação de pose humana 2D.



- **Estimações de Poses Humanas 3D**: Buscam localizar a posição tridimensional (x, y, z), em coordenadas de mundo, de cada um dos *keypoints* da pose, os quais representam a configuração espacial da pose real a partir da imagem de entrada (Figura 38).

Figura 38 – Exemplo de estimação de pose humana 3D.



APÊNDICE B – REDES NEURAIS CONVOLUCIONAIS

No contexto de redes neurais artificiais, a classe das redes neurais convolucionais (*CNNs*, *Convolutional Neural Networks*) se tornou uma das mais importantes abordagens no contexto de visão computacional, trazendo resultados significativamente positivos para tarefas relacionadas à classificação de imagens, detecção de objetos, reconhecimento de faces, reconstrução de objetos, dentre outros. As *CNNs* são uma variação do modelo de *perceptron* multicamadas (*MLP*, *Multilayer Perceptron*) com o objetivo de diminuir a quantidade de processamento envolvido a partir da extração de características locais.

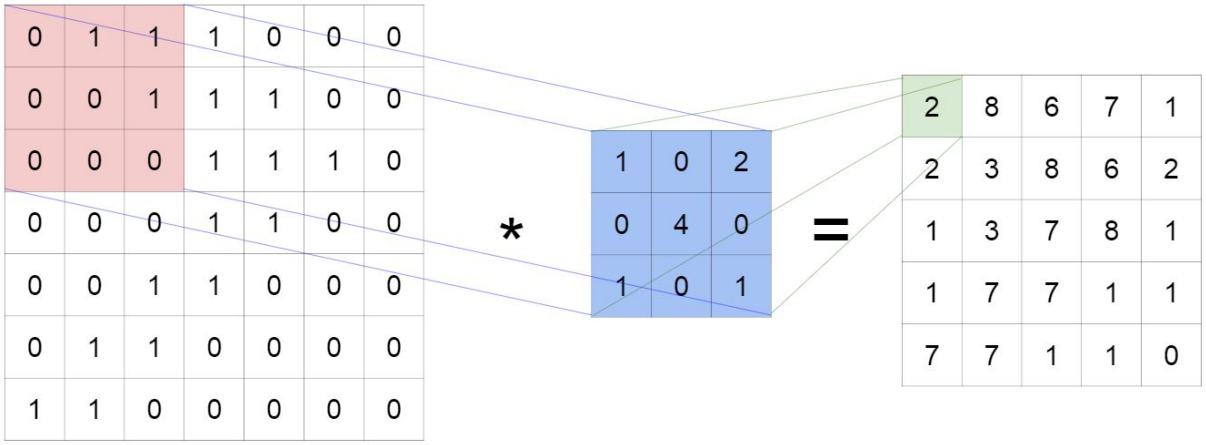
Seu nome deriva do termo "convolução", uma operação matemática na qual, dadas duas funções f e g, calcula-se o produto interno entre elas à medida em que g é deslocada ao longo de f. Para funções f e g discretas, a fórmula de convolução $f * g$ pode ser definida por:

$$(f * g)(x) = \sum_{j=0}^{x} f(j) \ . \ g(x - j) \tag{B.1}$$

No contexto de redes neurais convolucionais, a convolução é uma operação linear a qual envolve a multiplicação entre um conjunto de pesos g e a entrada f. Considerando o uso de *CNNs* para o processamento de imagens *RGB*, g é uma matriz de pesos (também chamada de *kernel*), onde $g \in \mathbb{R}^{g_w \times g_h \times c}$ e f representa uma matriz tridimensional equivalente a uma lista de mapas de *features*, onde $f \in \mathbb{R}^{w \times h \times c}$. Tradicionalmente, as operações de convolução são arquitetadas sequencialmente em uma *CNN*, e cada instância é chamada de camada convolucional (*convolutional layer*).

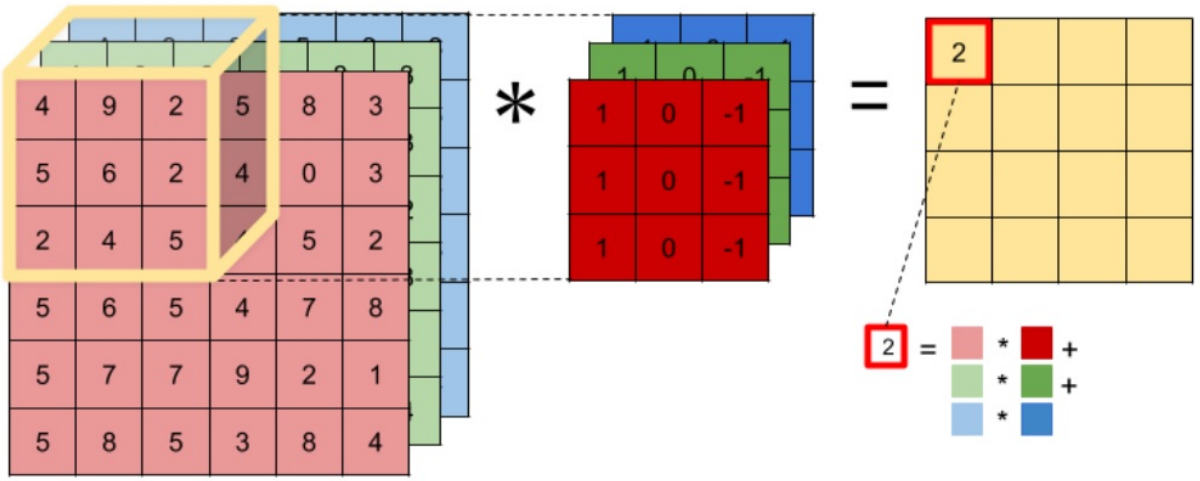
Considere a Figura 39 abaixo. Nela, temos um exemplo de convolução sendo realizada em uma matriz de entrada f (à esquerda) contendo apenas um canal. Para realizar a convolução, o *kernel* g (meio) é deslizado ao longo de f e é calculado o produto interno entre g (em azul) e a atual subregião sobreposta em f (em vermelho). Com isso, o resultado do produto interno (em verde) para cada subregião sobreposta é salvo em uma nova matriz de saída (à direita).

Figura 39 – Visualização da operação de convolução, com ênfase na operação de produto interno entre o *input* e o *kernel*.



Porém, operações de convolução em *CNNs* são mais comuns em entradas com mais de um canal presente. Considere a Figura 40 abaixo. Neste caso, a entrada possui dimensões (n_w, n_h, n_c), no qual n_c representa o número de canais e (n_w, n_h) representa as dimensões (altura e largura) dos canais. Portanto, o *kernel* terá dimensões (k_w, k_h, n_c), correspondendo a quantidade de canais do *kernel* com a quantidade de canais da entrada. Neste caso, a convolução se dá pela soma das operações de convoluções de um canal de entrada (produto interno) com o seu canal do *kernel* correspondente.

Figura 40 – Visualização da operação de convolução com a presença de múltiplos canais na entrada. A operação de produto interno é realizada entre os canais do *kernel* correspondentes (cores similares).

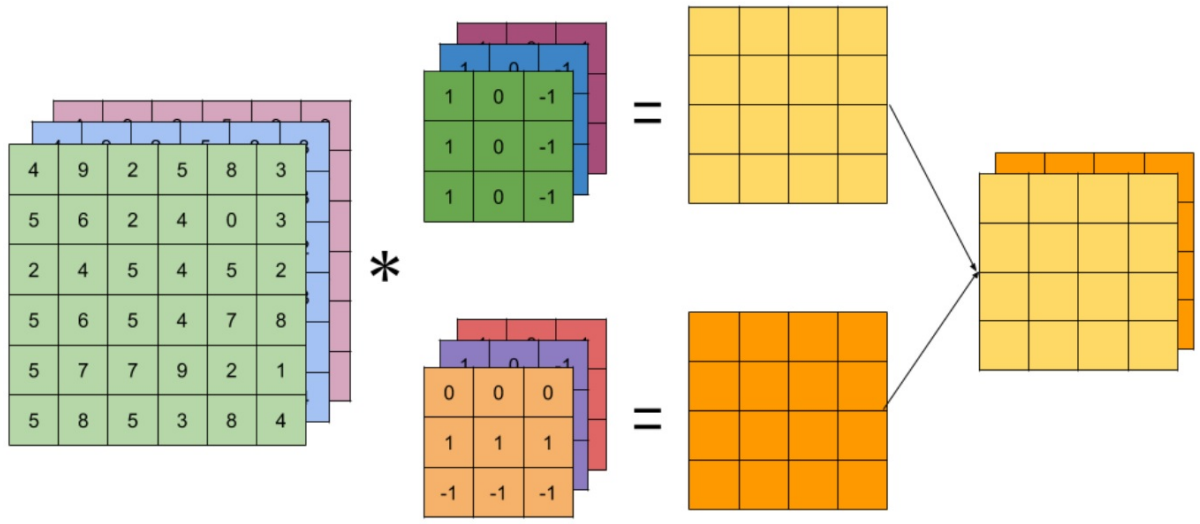


Fonte: https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/

Além disso, uma camada de convolução pode possuir múltiplos filtros, os quais equivalem às instâncias do *kernel*. Isso é comumente realizado a fim de detectar um conjunto

maior *features* a partir da entrada. Com isso, cada filtro do *kernel* gerará o seu mapa de *features* correspondente, como pode ser visualizado na Figura 41.

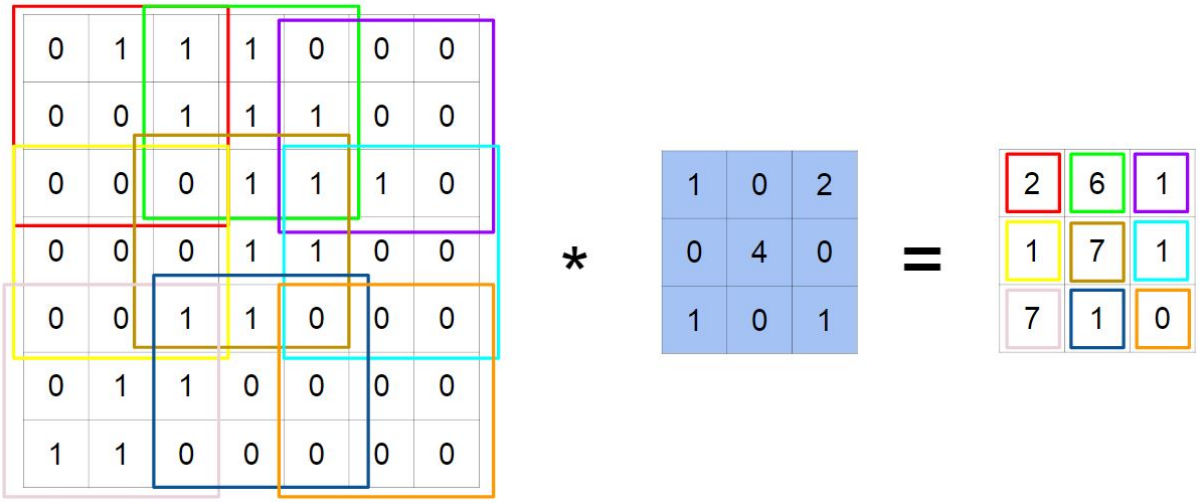
Figura 41 – Visualização da operação de convolução com a presença de múltiplos canais no *input* e mais de um filtro no *kernel*.. Para cada filtro, é gerado como saída um mapa de *features*.



Fonte: https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/

A fim de reduzir a quantidade de cálculos do modelo, também é possível realizar convoluções nas quais o *kernel*, ao deslizar sobre a entrada, realiza saltos, também conhecidos como *strides*. Com isso, apenas algumas das subregiões da entrada são computadas em relação ao *kernel*, ganhando assim desempenho na execução da rede (Figura 42).

Figura 42 – Visualização da operação de convolução em um *input* de um canal, com ênfase nos saltos entre as subregiões do *input*. Nesta convolução, foi aplicado um *stride* igual a 2.



Em *CNNs*, também são comuns as operações de *pooling*, as quais buscam reduzir o tamanho das representações das *features* ao realizar cálculos aritméticos entre as subregiões da entrada. Existem dois tipos principais de *pooling*:

- **Max Pooling:** Realiza o cálculo do valor máximo entre as subregiões da entrada (Figura 43);
- **Average Pooling:** Realiza o cálculo da média entre as subregiões da entrada (Figura 44);

Figura 43 – Visualização de uma operação de *Max Pooling* em um *input* contendo um canal, com *stride* igual a 2.

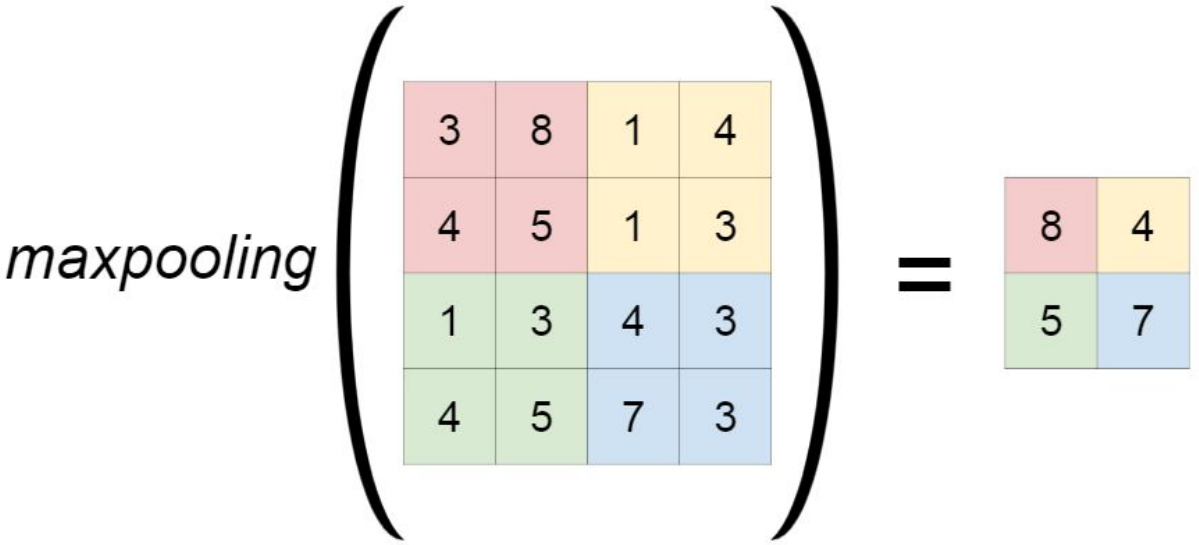
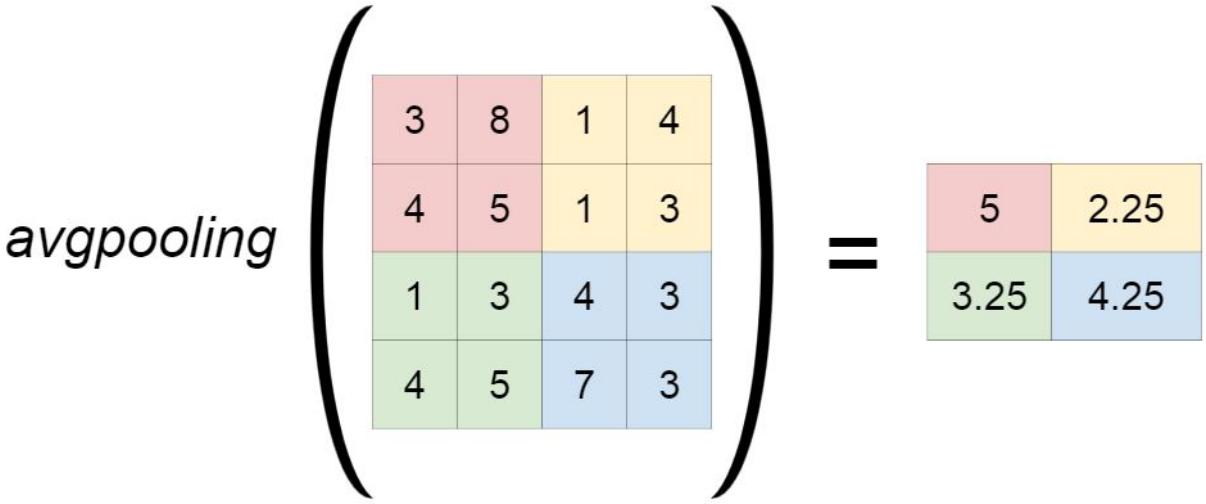
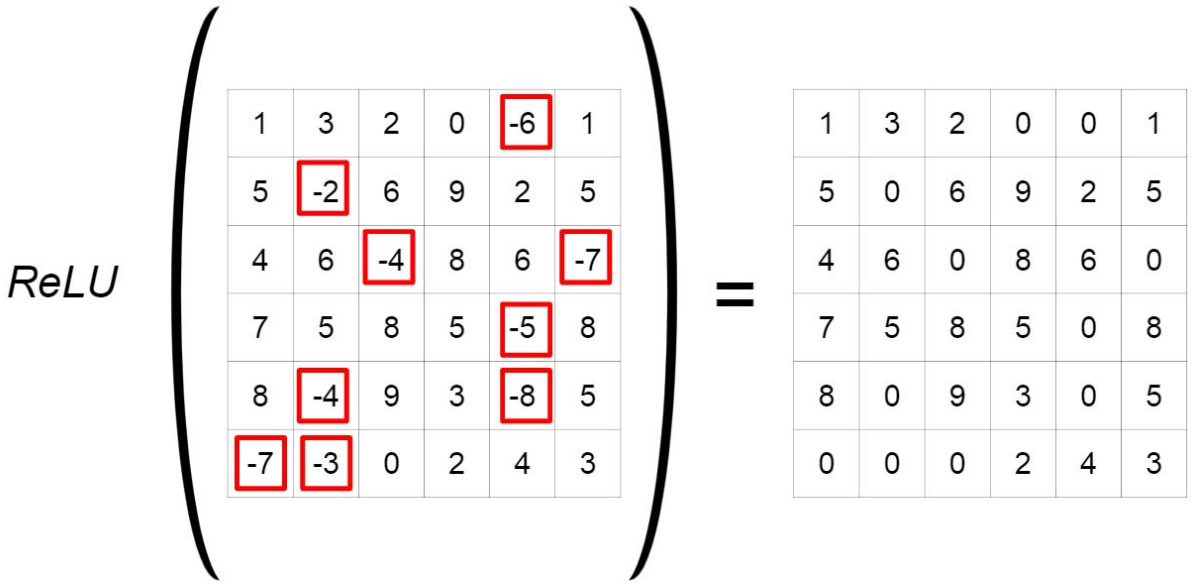


Figura 44 – Visualização de uma operação de *Average Pooling* em um *input* contendo um canal, com *stride* igual a 2.



Uma outro cálculo comumente utilizado em redes neurais convolucionais é a função de ativação chamada *ReLU* (abreviação de *Rectified Linear Unit*). Esta operação é aplicada no canal de entrada e substitui todas as ocorrências de *pixels* com valores negativos por zero (Figura 45). Uma vez que operações de convolução são lineares (envolvem somas e multiplicações entre valores das matrizes de entrada e *kernel*), o objetivo da *ReLU* é introduzir não-linearidade ao modelo, que pode assim aprender mapeamentos mais complexos entre a entrada e a saída esperada.

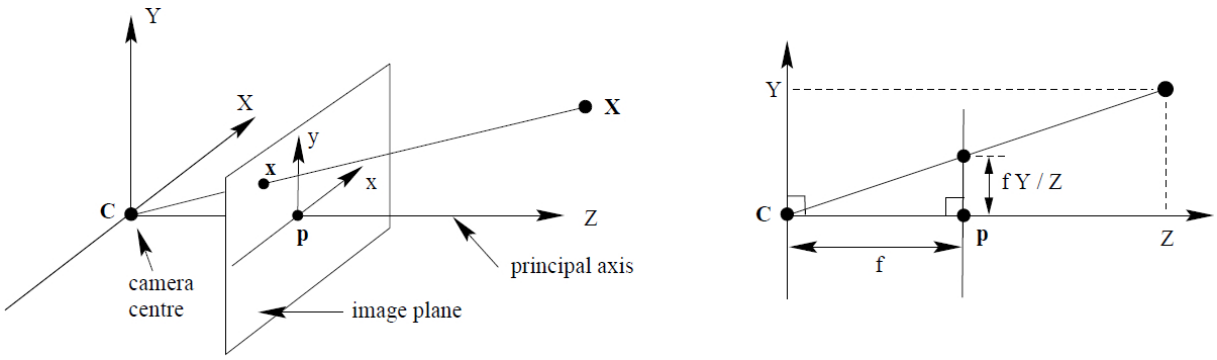
Figura 45 – Visualização de uma operação de *ReLU* em um *input* contendo um canal.



APÊNDICE C – MATRIZ DE CÂMERA

Para cruzar informações provenientes de diferentes câmeras, é preciso utilizar um modelo de câmera virtual a fim de que, além de se realizar a descrição dos conteúdos visuais representados por imagens em *pixels*, associe-se dados provenientes de diferentes câmeras em um único sistema de coordenadas virtuais. Para tal tarefa, um dos sistemas mais utilizados refere-se ao modelo de câmera estenopeica. Também conhecida como câmera *pinhole*, esta busca descrever a projeção por perspectiva, ou seja, o mapeamento geométrico entre o mundo 3D para o plano de imagem 2D. Nesta seção, serão descritos alguns dos principais conceitos referentes ao modelo de câmera *pinhole*, bem como abordagens para o cruzamento de informações entre diferentes câmeras.

Figura 46 – Modelo de câmera *pinhole*.



Fonte: (HARTLEY; ZISSERMAN, 2003).

C.0.1 Matriz de Parâmetros Intrínsecos

A matriz de parâmetros intrínsecos, comumente representada por K, refere-se aos parâmetros de câmera os quais descrevem o mapeamento entre os pontos 3D em coordenadas de câmera e suas respectivas coordenadas 2D, em *pixels*, na imagem. A matriz intrínseca é variante ao tipo real de câmera utilizado, e descreve as características ópticas e geométricas das imagens obtidas por ela. A matriz de parâmetros intrínsecos K possui o seguinte formato:

$$K = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.1}$$

no qual:

- $\boldsymbol{f_x}$ e $\boldsymbol{f_y}$: representam as distâncias focais (*focal length*) em relação aos eixos $\boldsymbol{x}$ e $\boldsymbol{y}$, respectivamente;
- $\boldsymbol{s}$: define a inclinação (*axis skew*) entre os eixos $\boldsymbol{x}$ e $\boldsymbol{y}$, e geralmente possui valor zerado;
- $\boldsymbol{x_0}$ e $\boldsymbol{y_0}$: representam o ponto principal, ou seja, o ponto definido pela intersecção entre o plano da imagem e a reta, perpendicular ao plano da imagem, que passa pelo centro da câmera.

C.0.2 Matriz de Parâmetros Extrínsecos

A matriz de parâmetros extrínsecos, definida por $[R|t]$, representa os parâmetros de câmera os quais descrevem o mapeamento entre os pontos 3D em coordenadas de mundo e seus respectivos pontos 3D em coordenadas de câmera. $[R|t]$ representa a posição e orientação da câmera no espaço de coordenadas de mundo, e é definido por:

$$[R|t] = \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array}\right] \tag{C.2}$$

no qual:

- $\boldsymbol{R}$: representa a matriz de rotação 3D da câmera, cuja definição deriva da representação em ângulos de *Euler*, tal que:

$$\boldsymbol{R} = \boldsymbol{R_x}(\boldsymbol{\alpha})\boldsymbol{R_y}(\boldsymbol{\beta})\boldsymbol{R_z}(\boldsymbol{\gamma}) \tag{C.3}$$

onde:

$$\boldsymbol{R_x}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cos\boldsymbol{\theta} & -\sin\boldsymbol{\theta} \\ \mathbf{0} & \sin\boldsymbol{\theta} & \cos\boldsymbol{\theta} \end{bmatrix} \tag{C.4}$$

$$\boldsymbol{R_y}(\boldsymbol{\theta}) = \begin{bmatrix} \cos\boldsymbol{\theta} & \mathbf{0} & \sin\boldsymbol{\theta} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ -\sin\boldsymbol{\theta} & \mathbf{0} & \cos\boldsymbol{\theta} \end{bmatrix} \tag{C.5}$$

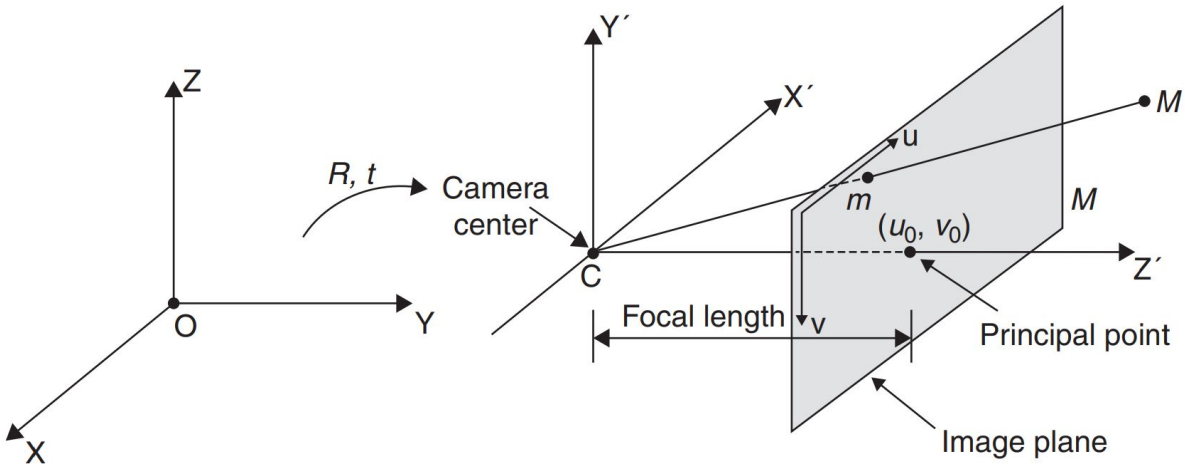
$$\boldsymbol{R_z}(\boldsymbol{\theta}) = \begin{bmatrix} \cos\boldsymbol{\theta} & -\sin\boldsymbol{\theta} & \mathbf{0} \\ \sin\boldsymbol{\theta} & \cos\boldsymbol{\theta} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{C.6}$$

onde $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ e $\boldsymbol{\gamma}$, definidos respectivamente pelos termos *yaw*, *pitch* e *roll*, representam as angulações em relação ao sistema de coordenadas;

- $\boldsymbol{t}$: representa o vetor de translação 3D, e pode ser interpretado como sendo a posição da origem do mundo em coordenadas de câmera. Junto a $\boldsymbol{R}$, pode ser utilizada para calcular a posição da câmera $\boldsymbol{C}$ em coordenadas de mundo, pela seguinte fórmula (LEPETIT; FUA, 2005):

$$\boldsymbol{C} = -\boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{t} \tag{C.7}$$

Figura 47 – Representação da transformação de coordenadas de mundo para coordenadas de câmera utilizando a matriz de parâmetros extrínsecos $[R|t]$.



Fonte: (JI; CHANG, 2014).

C.0.3 Matriz de Projeção

A matriz de projeção, comumente representada por P, é a matriz que descreve diretamente o cálculo das coordenadas 2D, em *pixels*, a partir da posição 3D dos pontos em coordenadas de mundo. Portanto, P deriva diretamente dos valores dos parâmetros intrínsecos e extrínsecos, pela seguinte fórmula:

$$P = K * [R|t] \tag{C.8}$$

C.0.4 Matriz Essencial

A matriz essencial é um tipo de matriz utilizada para calcular a relação entre conjuntos de pontos localizados por diferentes câmeras. Considere y_a e y_b dois conjuntos de pontos 2D

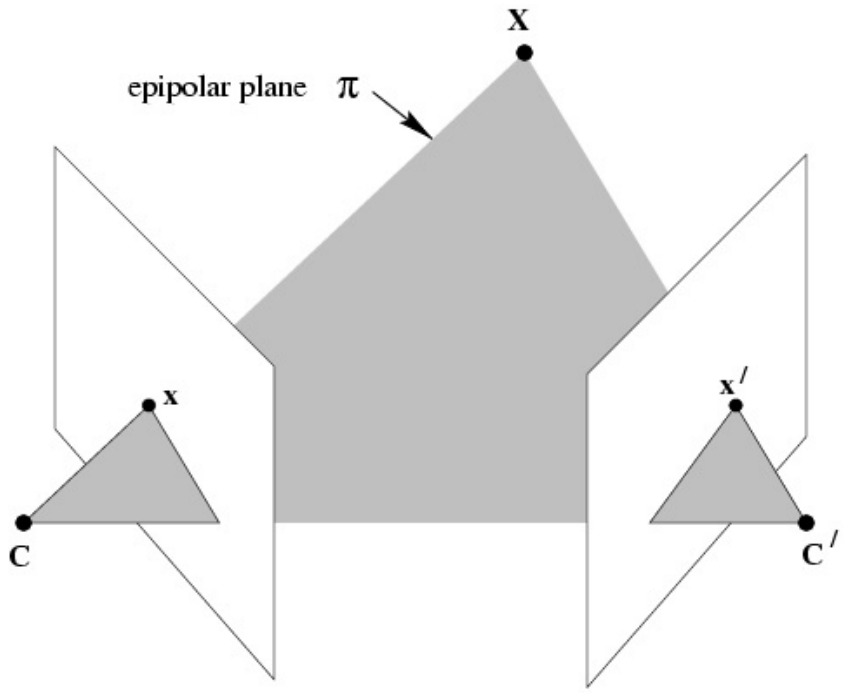
em, *pixels*, localizados pelas câmeras a e b, onde y_a e y_b representam pontos equivalentes no mundo real. A matriz essencial E pode ser definida como:

$$y_b K_b^{-1} E K_a^{-1} y_a = 0 \tag{C.9}$$

onde K_a e K_b são as matrizes de parâmetros intrínsecos das câmeras a e b.

A matriz essencial é uma representação da geometria epipolar, a qual busca calcular a relação entre pontos 2D equivalentes localizados em diferentes imagens, e representa um mapeamento projetivo de pontos de uma imagem para as linhas determinadas pela intersecção do plano da imagem com o plano epipolar (este definido pelos centros das câmeras e o ponto 3D triangulado), como pode ser visto na Figura 48.

Figura 48 – Correspondência entre os pontos 2D e o plano epipolar definido utilizando a geometria epipolar.



Fonte: (HARTLEY; ZISSERMAN, 2003).

APÊNDICE D – LISTA DE BIBLIOTECAS DE PYTHON UTILIZADAS

- **Matplotlib** (versão **3.2.1**);
- **NumPy** (versão **1.18.2**);
- **OpenCV** (versão **4.1.2.30**);
- **Pandas** (versão **1.0.3**);
- **PyTorch** (versão **1.4.0**);
- **SciPy** (versão **1.4.1**);
- **tqdm** (versão **4.43.0**);