

UNIVERSIDADE ESTADUAL PAULISTA - UNESP

Faculdade de Engenharia - Campus Guaratinguetá

WAGNER DE AGUIAR

**TÉCNICAS DE ANÁLISE BIOMÉTRICA POR IMAGENS PARA O AUXÍLIO DE
ESPECIALISTAS NO ACOMPANHAMENTO DE DOR CERVICAL CRÔNICA**

Guaratinguetá

2024

WAGNER DE AGUIAR

**TÉCNICAS DE ANÁLISE BIOMÉTRICA POR IMAGENS PARA O AUXÍLIO DE
ESPECIALISTAS NO ACOMPANHAMENTO DE DOR CERVICAL CRÔNICA**

Tese de Doutorado apresentada à Universidade Estadual Paulista (UNESP), Faculdade de Engenharia do Campus de Guaratinguetá, para a obtenção do título de Doutor em Engenharia Mecânica na área de Projetos Curso de Doutorado

Orientador: Prof^o Dr. José Celso Freire Junior

Coorientador: Prof^o Dr. Thomann Guillaume

Guaratinguetá

2024

A283t	Aguiar, Wagner de Técnicas de análise biométrica por imagens para o auxílio de especialistas no acompanhamento de dor cervical crônica / Wagner de Aguiar - Guaratinguetá, 2024. 163 f : il. Bibliografia: f. 126-133 Tese – Universidade Estadual Paulista, Faculdade de Engenharia e Ciências de Guaratinguetá, 2024. Orientador: Prof. Dr. José Celso Freire Junior Coorientador: Prof. Dr. Thomann Guillaume 1. Sensores. 2. Dor cervical. 3. Biometria.I. Título. CDU 62(043)
-------	--

Luciana Máximo
Bibliotecária CRB-8/3595

IMPACTO POTENCIAL DESTA PESQUISA

Baseado na pesquisa realizada e nos resultados obtidos, pode-se afirmar que os objetivos delineados no início deste trabalho foram alcançados com sucesso. O desenvolvimento da ferramenta de baixo custo para captação e análise de dados durante o teste de realocação cervicocefálica provou-se eficaz como ferramenta que pode ser utilizado na assistência a fisioterapeutas no tratamento de pacientes com disfunção proprioceptiva cervical, responsável por dores cervicais crônicas. O desenvolvimento foi realizado em colaboração com o laboratório G-SCOP na França e a utilização de sensores de movimento permitiram criar uma solução técnica robusta, enquanto a implementação da solução técnica com o MediaPipe da Google para a identificação do posicionamento da face demonstrou ser uma escolha apropriada, devido à sua precisão e rapidez. A pesquisa experimental validou o uso de câmeras de notebooks como uma ferramenta prática e acessível, confirmando que os modelos analisados podem ser utilizados de forma eficaz. Além disso, apesar do potencial das câmeras RealSense, a decisão de priorizar soluções mais econômicas e amplamente disponíveis reforça o objetivo de criar uma ferramenta acessível. Em síntese, a metodologia empregada, aliada às tecnologias selecionadas, possibilitou a criação de um sistema eficiente que atende às necessidades dos profissionais de saúde e facilita o tratamento de indivíduos com disfunções cervicais.

POTENTIAL IMPACT OF THIS RESEARCH

Based on the research carried out and the results obtained, it can be stated that the objectives outlined at the beginning of this work were successfully achieved. The development of the low-cost tool for data capture and analysis during the cervicocephalic relocation test proved to be an effective tool that can be used to assist physiotherapists in the treatment of patients with cervical proprioceptive dysfunction, responsible for chronic neck pain. The development was carried out in collaboration with the G-SCOP laboratory in France and the use of motion sensors allowed the creation of a robust technical solution, while the implementation of the technical solution with Google's MediaPipe for identifying face positioning proved to be an appropriate choice due to its accuracy and speed. The experimental research validated the use of notebook cameras as a practical and accessible tool, confirming that the models analyzed can be used effectively. Furthermore, despite the potential of RealSense cameras, the decision to prioritize more economical and widely available solutions reinforces the objective of creating an accessible tool. In summary, the methodology used, combined with the selected technologies, enabled the creation of an efficient system that meets the needs of health professionals and facilitates the treatment of individuals with cervical dysfunctions.

**ESTA TESE FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE
“DOUTOR EM ENGENHARIA MECÂNICA”**

**PROGRAMA: ENGENHARIA MECÂNICA
CURSO: DOUTORADO**

APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO

Profa. Dra. Ana Paula Rosifini Alves
Coordenadora

BANCA EXAMINADORA:



Documento assinado digitalmente

JOSE CELSO FREIRE JUNIOR

Data: 06/08/2024 15:41:25-0300

Verifique em <https://validar.iti.gov.br>

JOSE CELSO FREIRE JUNIOR
Orientador – UNESP



Documento assinado digitalmente

MARCELO TAVELLA NAVEGA

Data: 06/08/2024 20:46:01-0300

Verifique em <https://validar.iti.gov.br>

MARCELO TAVELLA NAVEGA
UNESP



Documento assinado digitalmente

NISE RIBEIRO MARQUES

Data: 24/09/2024 12:18:43-0300

Verifique em <https://validar.iti.gov.br>

NISE RIBEIRO MARQUES
Universidade do Sagrado Coração



Documento assinado digitalmente

DENISE MARTINELI ROSSI

Data: 26/09/2024 16:50:05-0300

Verifique em <https://validar.iti.gov.br>

DENISE MARTINELI ROSSI
Universidade Federal do Triângulo Mineiro



Documento assinado digitalmente

TARCÍSIO FERNANDES LEAO

Data: 26/09/2024 17:07:37-0300

Verifique em <https://validar.iti.gov.br>

TARCÍSIO FERNANDES LEÃO
Instituto Federal de Ciência e Tecnologia de São Paulo

AGOSTO de 2024

Dedico este trabalho à minha esposa Rosana, companheira de todos os momentos e minha maior incentivadora, minhas filhas Danielle e Débora e aos meus pais José Aguiar (em memória) e Maria Aguiar, meus primeiros orientadores. Sem eles eu nada teria conseguido.

AGRADECIMENTOS

Quero agradecer inicialmente a Deus, que me deu a vida e a quem devo tudo em minha trajetória.

Ao meu orientador, Prof. Dr. José Celso Freire Junior que desde o início acreditou em mim e sempre me incentivou, um amigo que pude contar em todos os momentos, e sem a sua orientação e auxílio, o estudo aqui apresentado seria impossível.

O Prof. Dr. Guillaume Thomann meu coorientador e amigo a quem devo tudo que desenvolvi e me direcionou na minha pesquisa e me acompanhou em meu crescimento a quem serei eternamente grato.

Ao professor e amigo Prof. Dr. Gilberto Cuarelli do IFSP por me auxiliar durante toda a caminhada no desenvolvimento desse trabalho, nas pesquisas e experimentações realizadas.

Ao Prof. Dr. Tarciso Fernandes Leão, mestre e amigo, que com sensibilidade e persistência, me transmitiu seus conhecimentos, ensinando que devo aprender sempre e dividir o que aprendi.

Aos professores e amigos do IFSP, Prof. Dr. Jacyro Gramulia, Prof. Dr. Alexandre Ventieri e Prof. Ms. Alberto Akio Shiga, que me incentivaram e são pessoas em quem me inspiro na arte de ensinar e a toda equipe do departamento de elétrica.

Ao Prof. Dr. Marcelo Tavella Navega por seu incentivo e orientação.

Aos funcionários da Faculdade de Engenharia do campus de Guaratinguetá, pela sua dedicação.

Ao meu sobrinho Victor Aguiar pelo apoio, auxílio e participação nos testes realizados.

Agradeço ao apoio financeiro da CAPES, pelo programa de Doutorado-Sanduíche no exterior.

Por fim, agradeço a minha esposa, minhas filhas, meus pais, irmãos e demais membros da minha família, que me incentivaram e compreenderam os momentos de ausência.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de
Nível Superior- Brasil (CAPES) - Código de Financiamento 88881.622775/2021-01

“Um homem se humilha se castram seu sonho, seu sonho é sua vida e vida é trabalho e sem o seu trabalho o homem não tem honra e sem a sua honra se morre, se mata “
(Gonzaguinha)

RESUMO

Dor cervical mecânica é um termo genérico usado para descrever dor cervical em indivíduos com disfunções cervicais, lesões ou dor no ombro. Esta condição é prevalente em jovens atletas, profissionais que executam movimentos repetitivos e pessoas que sofreram acidentes envolvendo movimentos bruscos da cabeça, resultando em desgaste das articulações. Fisioterapeutas frequentemente acompanham esses pacientes, analisando uma variedade de fatores durante a avaliação. Entre esses fatores se pode citar, alterações nos sistemas visuais, equilíbrio postural, déficits nos movimentos da cabeça, mobilidade reduzida do pescoço e dor muscular na região cervical, entre outros. Estas alterações são frequentemente relacionadas a modificações proprioceptivas que levam ao aumento da ativação e subsequente fadiga dos músculos flexores cervicais. Um método de avaliação utilizado pelos fisioterapeutas na análise de pacientes que apresentem sintomas de dor cervical mecânica, é o teste de relocação cervicocefálica, que envolve movimentar a cabeça para um lado e, com os olhos vendados, recolocá-la na posição inicial com máxima precisão. A diferença entre as duas posições é registrada e posteriormente analisada pelo especialista, em um processo que não é imediato. Com o avanço das tecnologias de medição da atividade humana, como a realidade virtual, várias aplicações estão sendo desenvolvidas para reabilitação e diagnóstico, mas muitas ainda necessitam de validação entre a comunidade especializada. Neste contexto, este trabalho apresenta um sistema desenvolvido como alternativa às técnicas atualmente utilizadas no acompanhamento de pacientes com dor cervical mecânica. Nele, utilizam-se um computador e uma câmera como ferramentas para captura e tratamento de imagens e dados de posicionamento. O sistema foi implementado utilizando a biblioteca Google MediaPipe, com a câmera atuando como sensor de movimento e posicionamento, capturando em tempo real sinais biométricos que permitem ao especialista avaliar clinicamente um paciente de forma rápida, eficiente e acessível. Além disso, o sistema proposto oferece uma ferramenta de baixo custo que facilita a captura de dados dos pacientes, simplificando e agilizando o processo diagnóstico em comparação aos métodos atuais. Essa ferramenta permite afirmar que é possível repetir o processo na avaliação de pacientes com dor cervical mecânica com uma câmera de notebook. Este trabalho foi realizado em colaboração com pesquisadores das áreas de Saúde e Engenharia do Laboratório G-SCOP, na França.

PALAVRAS-CHAVE: Análise de dados; sensores de movimento; dor cervical; realidade virtual.

ABSTRACT

Mechanical neck pain is a generic term used to describe neck pain in individuals with cervical dysfunctions, injuries, or shoulder pain. This condition is prevalent among young athletes, professionals who perform repetitive movements, and people who have experienced accidents involving sudden head movements, resulting in joint wear. Physiotherapists often monitor these patients, analyzing various factors during evaluation. These factors include changes in visual systems, postural balance, deficits in head movements, reduced neck mobility, and muscle pain in the cervical region, among others. These alterations are often related to proprioceptive changes that lead to increased activation and subsequent fatigue of the cervical flexor muscles. One method used by physiotherapists to evaluate patients with symptoms of mechanical neck pain is the cervicocephalic relocation test, which involves moving the head to one side and, with eyes closed, repositioning it to the initial position with maximum precision. The difference between the two positions is recorded and subsequently analyzed by the specialist in a process that is not immediate. With advancements in human activity measurement technologies, such as virtual reality, various applications are being developed for rehabilitation and diagnosis, but many still require validation within the specialized community. In this context, this work presents a system developed as an alternative to the techniques currently used in monitoring patients with mechanical neck pain. A computer and a camera are used as tools for capturing and processing images and positioning data. The system was implemented using the Google MediaPipe library, with the camera acting as a motion and positioning sensor, capturing real-time biometric signals that allow the specialist to clinically evaluate a patient quickly, efficiently, and accessibly. Additionally, the proposed system offers a low-cost tool that facilitates patient data capture, simplifying and speeding up the diagnostic process compared to current methods. This tool allows us to confirm that it is possible to repeat the process in the evaluation of patients with mechanical neck pain with a notebook camera. This work was carried out in collaboration with researchers from the Health and Engineering fields at the G-SCOP Laboratory in France.

KEYWORDS: Data analysis; motion sensors; neck pain; virtual reality.

LISTA DE ILUSTRAÇÕES

Figura 1	Diagrama Global com fases do trabalho	25
Figura 2	Equipamentos utilizado no procedimento de teste na medição da sensibilidade cinestésica cervicocefálica	30
Figura 3	Coleta de dados do TRC no alvo e na análise inicial	31
Figura 4	Visão estereoscópica ou projeção no plano de imagem com duas câmeras de um ponto M no espaço 3D	32
Figura 5	Componentes do Kinect - V1	35
Figura 6	Captura da Imagem em Profundidade	38
Figura 7	O Motor e os ângulos de visão	38
Figura 8	Câmeras de profundidade 3D da Microsoft	40
Figura 9	Câmeras de profundidade 3D da Asus	41
Figura 10	Câmeras de profundidade 3D Intel RealSense	42
Figura 11	Arquitetura do SDK do Kinect	47
Figura 12	A Arquitetura OpenNI	48
Figura 13	Rastreamento Esquelético no Kinect	53
Figura 14	Procedimento de leitura TRC e câmera	56
Figura 15	Folha milimetrada com laser indicando posição	57
Figura 16	Pontos de referência do corpo utilizando o MediaPipe pose	60
Figura 17	<i>468 points face texture mesh</i>	60
Figura 18	Malha de atenção: Visão geral do modelo	61
Figura 19	Detecção de face usando MediaPipe	62
Figura 20	Figura pontos MediaPipe	63
Figura 21	Captando posicionamento com sensor de movimento	65
Figura 22	Captando posicionamento de face com câmera e MediaPipe	66
Figura 23	Captando posicionamento de face com câmera e sensor de movimento utilizando o MediaPipe	66
Figura 24	Foto do capacete com laser fixado na ponta	66
Figura 25	Fluxograma do sistema primeira versão proposta Pose	68
Figura 26	Procedimento de pré-processamento de imagens	68
Figura 27	Primeira versão do sistema desenvolvido	69
Figura 28	Gráfico de linhas de movimento em coordenadas x e y	70
Figura 29	Tela para exportar dados	71
Figura 30	Tela de busca dos arquivos	71
Figura 31	Gráfico Pose X Y do ponto 0	72
Figura 32	Gráfico Alvo	72
Figura 33	Tela inicial do sistema versão Face Mesh	73

Figura 34	Tela de inicialização dos componentes Face Mesh	73
Figura 35	Tela de captação de pontos com o Face Mesh	74
Figura 36	Fluxograma do sistema versão Face Mesh	74
Figura 37	Imagem medição ponto nariz com sistema	79
Figura 38	Procedimento para realização dos cálculos	80
Figura 39	Comparação gráfica das medições de mesmo ponto calculado	83
Figura 40	Imagem do boneco posicionado em frente ao notebook e captado pelo sistema	83
Figura 41	Comparação gráfica das medições de mesmo ponto calculado	86
Figura 42	Inserção da média entre as medidas do ponto	87
Figura 43	Comparação gráfica de medições do mesmo ponto calculado - valores médios	90
Figura 44	Relação entre ombro e nariz com valores de média	91
Figura 45	Comparação gráfica de medições do nariz aos pontos do ombro calculado	93
Figura 46	Ampliação da figura para identificação dos pontos	94
Figura 47	Pontos próximos ao centro do nariz - 94	94
Figura 48	Medição pontos do nariz com Face Mesh	95
Figura 49	Comparação gráfica de medições de pontos diferentes, mas de mesmo quadro e ao mesmo tempo com cálculos	97
Figura 50	Imagem do boneco posicionado frente a câmera	98
Figura 51	Imagens das cores de fundos pretos e brancos	98
Figura 52	Imagens das cores de fundos azul e verde	99
Figura 53	Imagens das cores de fundos rosa e rosa claro	100
Figura 54	Comparação gráfica de medições com fundos pretos e brancos, com boneco e distância fixa	101
Figura 55	Comparação gráfica de medições com fundos azul e verde, com boneco e distância fixa	102
Figura 56	Comparação gráfica de medições com fundos rosa e rosa claro, com boneco e distância fixa	102
Figura 57	Distância de 80 cm	103
Figura 58	Câmera RealSense D435i posicionada no tripe	108
Figura 59	Câmera RealSense D455 posicionada no tripe	109
Figura 60	Comparação gráfica de medições do mesmo ponto calculado	111
Figura 61	Indivíduo posicionado a 80 cm do computador para realização de testes dinâmicos	111
Figura 62	Visualização do indivíduo com o capacete posicionado a 80 cm do compu- tador para realização de testes dinâmicos	112
Figura 63	Tela do sistema de captura dados com o Face Mesh	113
Figura 64	Comparação entre as medições realizadas com Face Mesh e a câmera Notebook utilizando capacete e papel milimetrado, com distância de 5cm	114

Figura 65	Comparação entre as medições realizadas com Face Mesh e a câmera Notebook utilizando capacete e papel milimetrado, com distância de 10cm	116
Figura 66	Tela do sistema de medições com a câmera Intel D435i RealSense utilizando capacete e papel milimetrado	116
Figura 67	Comparação gráfica de medições do deslocamento com câmera D435i RealSense calculado	118
Figura 68	Tela do sistema de medições com a câmera Intel D455 RealSense utilizando capacete e papel milimetrado	118
Figura 69	Comparação gráfica de medições do deslocamento com câmera D455 RealSense calculado	120

LISTA DE TABELAS

Tabela 1 – Visão geral dos sistemas de câmeras de profundidade	44
Tabela 2 – Comparação das Frameworks	49
Tabela 3 – Primeira leitura sem movimento da face	79
Tabela 4 – Segunda Leitura sem movimento de face	80
Tabela 5 – Comparação entre as medições de mesmo ponto calculado leitura 1	81
Tabela 6 – Comparação entre as medições de mesmo ponto calculado leitura 2	82
Tabela 7 – Tabela com cálculos de média e variância	82
Tabela 8 – Leitura 1 realizada com o boneco estático e posicionado a 80 cm do notebook	84
Tabela 9 – Leitura 2 realizada com o boneco estático e posicionado a 80 cm do notebook	84
Tabela 10 – Cálculos de distância, média e desvio padrão Pto. 1 - M1	85
Tabela 11 – Cálculos de distância, média e desvio padrão Pto. 1 - M2	86
Tabela 12 – Tabela com cálculos de média e variância	86
Tabela 13 – Primeira leitura sem movimento da face - com média de leituras	88
Tabela 14 – Segunda leitura sem movimento da face - com média de leituras	88
Tabela 15 – Cálculos de distância, média e desvio da leitura Pto. 1 - M1	89
Tabela 16 – Cálculos de distância, média e desvio da leitura Pto. 1 - M2	89
Tabela 17 – Tabela com cálculos de média e variância	90
Tabela 18 – Comparação entre as medições de mesmo ponto com inserção do valor médio entre o ombro e nariz	91
Tabela 19 – Comparação entre as medições de mesmo ponto com inserção do valor médio entre o ombro e nariz	92
Tabela 20 – Medições do ponto 141 com o Face Mesh	95
Tabela 21 – Medições do ponto 94 com o Face Mesh	96
Tabela 22 – Medições do ponto 370 com o Face Mesh	96
Tabela 23 – Tabela com cálculos de média e variância	97
Tabela 24 – Valores obtidos com o fundo preto e branco	99
Tabela 25 – Valores obtidos com o fundo azul e verde	100
Tabela 26 – Valores obtidos com o fundo rosa e rosa claro	101
Tabela 27 – Tabela com cálculos de média e variância	101
Tabela 28 – Tabela Notebook - DELL Latitude 7400	104
Tabela 29 – Análise ANOVA da tabela Notebook - DELL Latitude 7400	105
Tabela 30 – Tabela Notebook - Aspire 3 AMD Ryzen 5 3500U	105
Tabela 31 – Análise ANOVA da tabela Notebook - Aspire 3 AMD Ryzen 5 3500U	106
Tabela 32 – Tabela Notebook - Lenovo G40 70	106
Tabela 33 – Análise ANOVA da tabela Notebook - Lenovo G40 70	107

Tabela 34 – Medições de mesmo ponto com Face Mesh utilizando câmera intel D435i RealSense	110
Tabela 35 – Medições de mesmo ponto com Face Mesh utilizando câmera intel D455 RealSense	110
Tabela 36 – Medições do ponto 94 com Face Mesh e movimento 5 cm	114
Tabela 37 – Medições do ponto 94 com Face Mesh e movimento 10 cm	115
Tabela 38 – Medições de mesmo ponto com Face Mesh e a câmera Intel D435i RealSense utilizando capacete e papel milimetrado	117
Tabela 39 – Medições de mesmo ponto com Face Mesh e a câmera Intel D455 RealSense utilizando capacete e papel milimetrado	119

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i> (Interface de Programação de Aplicação)
ASC	<i>Advanced Storage Capacity</i> (Capacidade Avançada de Armazenamento)
ANNs	<i>Artificial Neural Networks</i> (Redes Neurais Artificiais)
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CE	<i>Closed Eyes</i> (Olhos fechados)
CNNs	<i>Convolutional Neural Networks</i> (Redes Neurais Convolucionais)
DP	Desvio Padrão
FE	<i>Feature Extration</i> (Extração de Recursos)
FPS	<i>Frames Per Second</i> (Quadros Por segundo)
GPS	<i>Global Position System</i> (Sistema de Posicionamento Global)
GPUs	<i>Graphics Processing Units</i> (Unidade de Processamento Gráfico)
HD	<i>High Definition</i> (Alta Definição)
IA	<i>Artificial Intelligence</i> (Inteligencia Artificial)
IHM	Interface Homem Máquina
MCP	<i>Multi-core processor</i> (Processador de Vários Núcleos)
MS	<i>Mobile Sensing</i> (Sensores Moveis)
MV	<i>Mobile Vision</i> (Visão Móvel)
OC	<i>Object Classification</i> (Classificação de Objetos)
OE	<i>Open Eyes</i> (Olhos Abertos)
OpenNI	<i>Open Natural Interface</i> (Código Aberto Interface)
PDSE	Programa de Doutorado-Sanduíche no Exterior
RGB	<i>Red, Green, Blue</i> (Vermelho, Verde, Azul)
SDK	Kit de desenvolvimento de software

TCC	Trabalho de Conclusão de Curso
TPUs	<i>Tensor Processing Units</i> (Unidade de Processamento Tensor)
UNESP	Universidade Estadual Paulista

SUMÁRIO

1	INTRODUÇÃO	21
1.1	DESENVOLVIMENTO DO TRABALHO DE PESQUISA	21
1.1.1	O Trabalho no Brasil	22
1.1.2	O Trabalho na França	22
1.2	PROBLEMÁTICA E OBJETIVOS	23
1.3	JUSTIFICATIVA	23
1.4	SINOPSE DO TRABALHO DE TESE	24
1.5	ESTRUTURA DO DOCUMENTO	26
2	CONTEXTUALIZAÇÃO DO TRABALHO	27
2.1	SENSIBILIDADE CINESTÉSICA CERVICOCEFÁLICA	27
2.1.1	Síndrome proprioceptiva cervical	28
2.1.2	Teste de realocação cervicocefálica - TRC	29
2.1.3	Técnica Clínica TRC Adaptada	30
2.2	VISUALIZAÇÃO DIGITAL DE OBJETOS	31
2.3	SENSORES DE MOVIMENTO	33
2.3.1	Panorama histórico	33
2.3.2	Sensores Microsoft	34
2.3.2.1	Sensores de Movimento Microsoft: Componentes e Interação	35
2.3.2.2	Sensores de Movimento Microsoft: evolução	39
2.3.3	Sensores Asus	40
2.3.4	Sensores Intel	41
2.3.5	Comparação de sensores de movimento	42
2.3.6	Interagindo com Sensores de Movimento	45
2.3.6.1	Kinect for Windows SDK 2.0 da Microsoft	46
2.3.6.2	OpenNI	47
2.3.6.3	OpenKinect	48
2.3.7	Aplicações com Sensores de Movimento	49
2.3.7.1	Rastreamento de um modelo esquelético	52
2.3.8	Repetibilidade e Reprodutibilidade	53
2.3.9	Ferramentas para Implementação	54
2.3.9.1	Linguagem JavaScript	54
2.3.9.2	Biblioteca TensorFlow	54
2.4	SÍNTESE	55
3	METODOLOGIA EXPERIMENTAL E TECNOLOGIA ENVOLVIDA	56

3.1	PROCEDIMENTO DE LEITURA	56
3.2	A ESTRUTURA MEDIAPIPE	58
3.2.1	Medição com o MediaPipe	63
3.2.2	Aplicações com MediaPipe	64
3.3	ARQUITETURA DO SISTEMA DESENVOLVIDO	64
3.4	DESENVOLVIMENTO DO SISTEMA	67
3.4.1	Primeira versão do sistema desenvolvido	68
3.4.2	Segunda versão do sistema desenvolvido	73
3.5	Síntese	75
4	EXPERIMENTAÇÃO E ANÁLISE DOS RESULTADOS	76
4.1	AVALIAÇÃO E ANÁLISE DO MOVIMENTO DE FACE	76
4.2	TESTE ESTÁTICO - MEDIAPIPE POSE	77
4.2.1	Medições com câmera do Notebook	78
4.2.2	Medições com câmera do Notebook - Boneco	83
4.2.3	Medições com a câmera do Notebook - Valores médios	87
4.2.4	Medições com a câmera do Notebook - Três pontos de leitura	90
4.3	TESTE ESTÁTICO - MEDIAPIPE FACE MESH	93
4.3.1	Medições com a câmera do Notebook	94
4.3.2	Medições com câmera Notebook usando FaceMesh com alteração das cores de fundo	97
4.3.3	Medições com câmera Notebook usando FaceMesh com Boneco em dife- rentes distâncias	102
4.3.4	Medições com a câmera Intel RealSense D435i e D455	108
4.4	TESTES DINÂMICOS	111
4.4.1	Medições com câmera de notebook - Teste 01	112
4.4.2	Medições com câmera de notebook - Teste 02	114
4.4.3	Medições com a câmera Intel RealSense D435i	116
4.4.4	Medições com a câmera Intel RealSense D455	118
4.5	ANÁLISE E COMPARAÇÃO ENTRE OS MÉTODOS E RESULTADOS OBTIDOS NOS EXPERIMENTOS	120
4.5.1	Etapa Estática	120
4.5.2	Etapa Dinâmica	121
5	CONSIDERAÇÕES FINAIS	123
5.1	SUGESTÕES PARA TRABALHOS FUTUROS	125
	REFERÊNCIAS	127
	APÊNDICE A – SOFTWARE JAVASCRIPT	135

APÊNDICE B – MINHAS REFERÊNCIAS BIBLIOGRÁFICAS . . . 164

1 INTRODUÇÃO

Este trabalho de pesquisa aborda a realização de teste de realocação cervicocefálica que é essencial para o diagnóstico de disfunções cervicais. O teste oferece um método de avaliação utilizado medir a precisão proprioceptiva do pescoço (capacidade do sistema nervoso de detectar e controlar a posição e o movimento da cabeça e do pescoço no espaço). Durante o teste, o paciente move a cabeça para um lado e, com os olhos vendados, tenta reposicioná-la na posição inicial com a máxima precisão. A diferença entre as posições é registrada e analisada pelo especialista para avaliar déficits de percepção e controle cervical. Este teste é frequentemente utilizado para diagnosticar e monitorar disfunções cervicais e problemas relacionados à propriocepção.

Conforme se poderá observar ao longo do trabalho, a pesquisa realizada oferece uma alternativa a solução adotada atualmente para a realização de teste de realocação cervicocefálica. Este trabalho aborda processos de "interação incorporada"¹, que são definidos por Hartson como: "[...] a capacidade de envolver o corpo físico em interação com a tecnologia de forma natural, utilizando, por exemplo, gestos"² (Hartson; Pyla, 2012, p. 328, tradução nossa).

No cenário da interação incorporada, diversos tipos de aparatos podem ser utilizados. Entre eles, pode-se citar os seguintes dispositivos (Majgaard, 2014): Robôs e tecnologias robóticas, táteis e tangíveis, "tecnologias vestíveis"³, de grandes objetos e arredores interativos⁴. Este estudo irá se concentrar na última categoria (arredores interativos) e irá envolver um processo de aquisição da descrição geométrica de uma cena.

A pesquisa está inserida em um projeto de cooperação entre a Universidade Estadual Paulista "Júlio de Mesquita Filho- Unesp, e o Grenoble INP da *l'Université Grenoble Alpes* - UGA, na França, onde se localiza o Laboratório G-SCOP (*Laboratoire des Sciences pour la Conception, l'Optimisation et la Production de Grenoble*) que abrigou parte do desenvolvimento do protótipo e os experimentos realizados.

1.1 DESENVOLVIMENTO DO TRABALHO DE PESQUISA

O trabalho de pesquisa desenvolvido nesta tese foi realizado, em maior parte, no Brasil e, por um período de seis meses, no laboratório do G-SCOP na França. As diferentes atividades desenvolvidas são apresentadas a seguir.

¹ *embodied interaction*

² *"embodied interaction refers to the ability to involve one's physical body in interaction with technology in a natural way, such as by gestures"*

³ *Interactive Wearables* - de colocação junto ao corpo e portáteis

⁴ *Interactive Surroundings*

1.1.1 O Trabalho no Brasil

O trabalho teve início em agosto de 2019 com a fase de pesquisa, acompanhamento e validação de disciplinas específicas. Esta etapa teve duração de dois anos e seis meses, finalizando-se no primeiro semestre de 2021.

Em 2020, enquanto as disciplinas eram cursadas, teve início a pesquisa bibliográfica. Esta atividade constou da avaliação de trabalhos existentes e de tecnologias utilizadas na análise dos movimentos do corpo humano, bem como da utilização de sensores de movimento e de outras ferramentas de medição. Paralelamente ao trabalho de pesquisa bibliográfica, a interação com as ferramentas existentes permitiu a compreensão das escolhas iniciais de tecnologia e a familiarização com sua utilização, envolvendo a aquisição de câmeras Intel RealSense, o que tornou possível verificar vantagens e desvantagens em relação a outras existentes no mercado. Dessa forma, pôde-se fazer um trabalho exploratório no desenvolvimento de um sistema de captação da imagem para posterior conversão em dados.

Realizou-se, ainda simultaneamente à pesquisa bibliográfica, o pré-projeto para estágio no exterior que foi submetido à CAPES no Edital PDSE da Unesp de dezembro de 2020, tendo sido aprovado em junho de 2021. Este financiamento permitiu que a etapa francesa do trabalho pudesse ser desenvolvida a partir de fevereiro de 2022.

1.1.2 O Trabalho na França

Conforme mencionado, a parte do trabalho de pesquisa realizada na França teve como local de acolhida o laboratório G-SCOP do Grenoble INP, que é um dos membros da Université Grenoble Alpes (UGA).

Essa etapa do trabalho se iniciou com uma interação com as atividades desenvolvidas por estudantes de engenharia da instituição que empregavam o Kinect, o RealSense e outras ferramentas na análise de movimentos do corpo humano. A compreensão das tecnologias empregadas pela equipe foi fundamental para a continuidade da pesquisa.

Essas atividades eram naturalmente orientadas à escolha do componente necessário para a leitura dos dados e da medição pela câmera. A aquisição de conhecimentos teóricos relacionados à análise do movimento da face e a busca por tecnologia apropriada relacionada à captura de movimento também foram frutos dessas atividades e se mostraram de grande importância para a continuidade do trabalho de pesquisa.

Diversas outras atividades foram desenvolvidas durante a estadia na França. As principais estão listadas a seguir.

- Reuniões com fisioterapeutas para refinar o problema de pesquisa, estabelecer a lista de requisitos e experimentos de trabalho para validação de protótipos;
- Estudos para compreender e caracterizar os sensores específicos utilizados pelo sistema para a detecção da imagem em frente ao computador;

- Desenvolvimento de um protótipo do software e discussão de algumas funções com o fisioterapeuta;
- Participação em eventos científicos no laboratório G-SCOP e em Paris (Congresso francês HANDICAP 2022). Esta atividade contribuiu com o aprimoramento dos conhecimentos associados à preparação de comunicações científicas;
- Colaboração em eventos para desenvolver a parceria internacional e a rede de pesquisa e intercâmbio científico entre Grenoble INP – UGA, G-SCOP e o Instituto Federal de São Paulo (IFPS), que é a instituição de trabalho do doutorando;
- Presença em eventos sociais organizados dentro do laboratório G-SCOP.

1.2 PROBLEMÁTICA E OBJETIVOS

Atualmente, para a realização do teste de realocação cervicocefálica, é necessária uma sala com no mínimo 16 metros quadrados e que ofereça um ambiente controlado com câmeras que possam gravar todo o procedimento ao qual o paciente é submetido.

Para a análise dos dados captados durante o procedimento, é necessário inicialmente, que o especialista reveja a gravação e avalie se o procedimento foi realizado em conformidade com o protocolo estabelecido. Em seguida, ele deve anotar os dados de movimentação e os inserir em uma planilha para a realização dos cálculos de movimentação. Após, deve calcular as diferenças em graus entre as posições iniciais e finais dos movimentos realizado pelo indivíduo.

Conforme se pode depreender, o tempo para a análise dos dados coletados e o custo do ambiente são fatores dificultadores para uma ampla utilização de testes de realocação cervicocefálica, e em diferentes situações, constitui um problema para disseminação da utilização desta técnica.

A problemática que este trabalho aborda é, portanto, associada a questão da dificuldade que pode existir para uma ampla utilização de testes de realocação cervicocefálica. E para tratar este problema, este trabalho de pesquisa estabeleceu como objetivo o desenvolvimento de uma ferramenta de baixo custo que facilite a captação e análise dos dados obtidos durante a realização de um teste de realocação cervicocefálica, para auxiliar fisioterapeutas no tratamento de pacientes com disfunção proprioceptiva cervical que levam ao aparecimento de dores cervicais crônicas.

1.3 JUSTIFICATIVA

Diversos fatores podem ser citados para justificar o desenvolvimento deste projeto de pesquisa. Conforme introduzido, um deles está associado a questões financeiras. A análise e o acompanhamento de pessoas com dor cervical crônica, com a utilização de espaços próprios, com comprimentos superiores a 4 metros, e a necessidade da gravação do processo com equipamentos específicos, que são algumas das exigências para que o fisioterapeuta possa realizar as análises adequadas, envolvem um alto custo. A solução apresentada neste trabalho busca minimizar esses

custos, pois propõe que esses testes sejam realizados em consultórios menores com equipamentos de baixo custo, que, muitas vezes, já se encontram nos consultórios.

Outro fator a considerar está associado ao tempo que o profissional de saúde deve dedicar às fases de avaliação clínica. Com o processo atual, o especialista leva muito tempo para analisar os dados capturados, uma vez que a análise dos dados é feita posteriormente à gravação dos movimentos do paciente. Nesta análise, o especialista deve assistir à gravação de todo o procedimento para ir obtendo os valores de movimentação. Só então pode realizar a avaliação clínica. Neste contexto, o presente trabalho se justifica por auxiliar o profissional de saúde na análise dos movimentos, fornecendo informações gráficas sobre os dados captados pela câmera em tempo real e sem que seja necessário assistir a todo o procedimento novamente.

Pode-se mencionar também o fato de que, para utilização dos procedimentos estabelecidos atualmente, existe a necessidade do deslocamento dos pacientes aos grandes centros para realização de exames em salas dedicadas, situação que ainda pode dificultar as consultas. A possibilidade de utilização do sistema proposto neste projeto de pesquisa em instalações localizadas em consultórios comuns pode contribuir para minimizar este problema.

O fato deste projeto de pesquisa envolver também uma parceria internacional, entre o Departamento de Engenharia Elétrica da FEG / Unesp e o Laboratório G-SCOP, fortalece a atuação internacional da FEG/Unesp e pode também ser mencionada como uma razão para seu desenvolvimento.

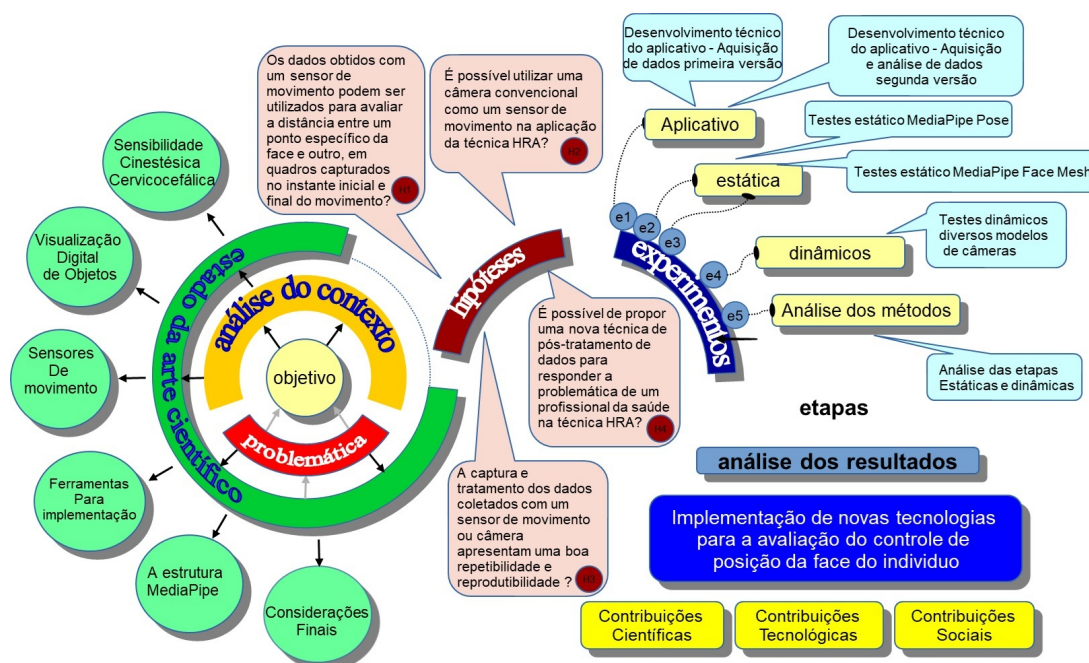
Para além da questão da parceria, o grupo do G-SCOP foi escolhido pois, no laboratório, desenvolvem-se várias atividades de pesquisa que envolvem a utilização eficiente dos sensores de movimento para medir parâmetros cinemáticos associados à análise da postura humana bem como a medição da capacidade motora de indivíduos. Esta escolha envolveu também a possibilidade de envolvimento do Prof. Guillaume Thomann, especialista na área, e que atuou como coorientador da pesquisa.

Além disso, a expertise dos pesquisadores do laboratório na utilização de sensores de movimento, a participação da equipe de engenheiros e estudantes de Grenoble, bem como a colaboração de fisioterapeutas do hospital da cidade, que atuam em vários tópicos relacionados ao trabalho de tese, foram fundamentais para o sucesso deste trabalho, e justificam a parceria internacional.

1.4 SINOPSE DO TRABALHO DE TESE

Com o objetivo de ajudar na compreensão do desenvolvimento desta pesquisa, apresenta-se na Figura 1 um diagrama descrevendo as fases do trabalho. Nessa figura, pode-se observar a problemática envolvida com o trabalho e as soluções implementadas, bem como qual o caminho adotado no desenvolvimento da presente pesquisa.

Figura 1 – Diagrama Global com fases do trabalho



Fonte: Elaborado pelo autor

Pode-se também observar as hipóteses formuladas (H1 a H4) e os experimentos de captação da posição da face (estáticos e dinâmicos) que foram definidos para verificá-las.

- H1 - Os dados obtidos com um sensor de movimento podem ser utilizados para avaliar a distância entre dois pontos específico da face de um indivíduo, em quadros capturados no instante inicial e final do movimento?
- H2 - É possível utilizar uma câmera convencional como um sensor de movimento na aplicação da técnica TRC?
- H3 - A captura e o tratamento dos dados coletados com um sensor de movimento ou câmera apresentam uma boa repetibilidade e reprodutibilidade?
- H4 - É possível propor uma nova técnica de pós-tratamento de dados para responder à problemática de um profissional da saúde na técnica TRC?

Ao longo deste trabalho, detalhar-se-ão as diferentes atividades desenvolvidas na construção de um sistema que permite a leitura das posições iniciais e finais de um ponto na face de um indivíduo de acordo com uma movimentação específica à qual ele será submetido, fornecendo todas essas informações em tempo real ao profissional de saúde. A forma como este trabalho foi desenvolvido está detalhada na próxima seção.

1.5 ESTRUTURA DO DOCUMENTO

O presente trabalho de pesquisa está estruturado de forma que, após a exposição de seus objetivos, de sua justificativa e das hipóteses adotadas, se apresenta, no Capítulo 2, sua contextualização, descrevendo os conceitos necessários à sua compreensão. Nesta parte, serão analisados os sensores de movimento considerando o panorama histórico, as suas características e a visão geral de aplicações que utilizam esses equipamentos. Também serão apresentadas as ferramentas e os métodos utilizados atualmente para análise de movimentos do corpo. Além disso, procura-se identificar quais os softwares são necessários para o desenvolvimento do sistema com os equipamentos disponíveis. Realiza-se também uma pesquisa sobre a viabilidade do emprego de softwares livres ou que não tenham custo de utilização, principalmente com o uso das câmeras de profundidade. Por fim, são apresentadas as aplicações com sensores de movimento e com a tecnologia Google MediaPipe e as considerações finais dessa contextualização.

No Capítulo 3, introduz-se a metodologia utilizada na pesquisa e suas hipóteses, considerando o desenvolvimento do trabalho técnico e do sistema. São introduzidos também os testes e experimentos que serão desenvolvidos para verificar as hipóteses propostas e atingir os objetivos do trabalho de pesquisa.

Em seguida, no Capítulo 4, introduz-se o Teste de Realocação Cervicocefálica - TRC, e sua importância nos procedimentos de medição realizados com o Google MediaPipe. Na sequência, apresentam-se os procedimentos experimentais realizados, sendo individualizados a partir da forma que foram executados, a saber: testes estáticos e dinâmicos. A cada teste, realiza-se uma análise dos valores obtidos e o ambiente em que o experimento foi realizado, a fim de que o teste seguinte seja direcionado com objetivo de resolver os eventuais problemas apresentados no teste anterior. Esses testes envolvem a análise dos resultados adquiridos com o sistema e uma representação gráfica. Por fim, conclusões mais detalhadas sobre cada experimento são feitas e perspectivas de pesquisas são propostas.

O Capítulo 5, finaliza o trabalho com a discussão dos resultados do que foi realizado ao longo deste trabalho e indicam-se sugestões sobre novos desenvolvimentos utilizando esta tecnologia. Alguns apêndices também fazem parte deste trabalho de pesquisa. Enquanto o primeiro contém os elementos de software desenvolvido em Javascript, o segundo apêndice disponibiliza os programas que foram desenvolvidos para implementação do sistema.

2 CONTEXTUALIZAÇÃO DO TRABALHO

Este capítulo introduz as principais ideias e conceitos utilizados para desenvolver a ferramenta proposta para auxiliar fisioterapeutas na realização do teste de realocação cervicocefálica. Inicialmente se aborda os aspectos clínicos associados ao trabalho, em particular a temática da Sensibilidade Cinestésica Cervicocefálica, central na questão abordada na pesquisa, seguida da apresentação da Síndrome Proprioceptiva Cervical e da técnica clínica utilizada na avaliação da condição médica de pacientes acometidos por tal síndrome. Na sequência se introduz os sensores de movimento e técnicas que possibilitam sua aplicação para a captação de imagens e movimentos. O capítulo é finalizado com a apresentação de algumas ferramentas complementares que foram utilizadas para a implementação deste trabalho de pesquisa.

2.1 SENSIBILIDADE CINESTÉSICA CERVICOCEFÁLICA

O pescoço desempenha uma função crucial na vida cotidiana, sustentando a cabeça. A coluna vertebral na região do pescoço, conhecida como coluna cervical, é composta por sete vértebras separadas por discos de material gelatinoso e cartilagem. A medula espinhal passa através da coluna vertebral, com nervos espinhais emergindo entre as vértebras para conectar-se aos nervos do corpo. A raiz nervosa espinhal está próxima à medula espinhal, enquanto músculos e ligamentos do pescoço dão suporte à coluna vertebral.

Nesse sistema complexo, a propriocepção é definida como a interação entre receptores aferentes e eferentes que controlam a posição e o movimento do corpo e suas partes no espaço, sendo essencial para a função articular adequada. A propriocepção pode ser alterada por diferentes mecanismos, como trauma direto aos mecanorreceptores, prejuízo funcional dos músculos (incluindo aumento da fadigabilidade), ou alterações degenerativas, como transformação de fibras, infiltração gordurosa, atrofia ou inibição muscular.

Devido à sua flexibilidade, a coluna cervical é suscetível a desgaste e lesões, sendo a dor resultante desses processos, juntamente com a dor lombar, uma das razões mais comuns pelas quais as pessoas procuram atendimento médico. Problemas de má postura e o processo degenerativo relacionado à idade tornam a função primordial do pescoço mais difícil e as dores subsequentes mais frequentes. Sintomas como vertigem, instabilidade e cefaleia são frequentemente encontrados em pessoas com disfunções cervicais e podem ser provocados por trauma direto, trauma esportivo, doenças articulares degenerativas, problemas no sistema vestibular, lesão cerebral e até ansiedade.

Lesões podem ocorrer devido a traumas, como quedas ou acidentes de carro, mas são mais comumente resultantes de atividades rotineiras, como levantamento de peso, movimentos inesperados ou durante a prática de exercícios físicos. Déficits proprioceptivos podem surgir de danos aos tecidos moles da coluna cervical, sendo considerados fatores importantes para os

sintomas mencionados anteriormente.

2.1.1 Síndrome proprioceptiva cervical

Os sistemas somatossensorial cervical (proprioceptivo), visual e vestibular recebem informações referentes à postura, à posição e aos movimentos da cabeça e dos olhos, respectivamente, sendo que os impulsos aferentes convergem em diversas regiões do córtex (Treleaven, 2008). Em particular, as informações proprioceptivas têm a função de proteção das articulações, através da ativação dos mecanorreceptores (Solomonow *et al.*, 1998).

A integração da região cervical com o sistema visual e vestibular que é a integração dos três sistemas (proprioceptivo cervical, vestibular e ótico) é realizada pela existência de três reflexos: o cérvico-ocular, o vestibulo-ocular e o opto cinético. O cérvico-ocular age em conjunto com o vestibulo-ocular e o opto cinético atua na função de estabilidade da visão durante o movimento da cabeça (Raizah *et al.*, 2023).

Para, por exemplo, permitir que os olhos se movimentem em direção oposta à cabeça, estímulos aferentes proprioceptivos da musculatura cervical e das capsulas articulares alcançam o núcleo vestibular. O reflexo vestibulo-ocular pode ser dividido em componentes rotacional e translacional que estimulam os canais semicirculares e órgãos otolíticos, sendo que, quando a cabeça realiza a rotação, este reflexo permite que os olhos se movimentem em direção oposta.

Já foi demonstrado que o funcionamento adequado do complexo cabeça-pescoço-olhos está relacionado a informações provenientes dos receptores localizados na cápsula das articulações e aos músculos da região cervical (Cordo *et al.*, 1995).

Nos casos de disfunção cervical e comprometimento proprioceptivo, existe aumento da latência do reflexo vestibulo-ocular e do cérvico-ocular, que se manifestam pela alteração no controle fino do movimento dos olhos. A hipótese que sustenta esta afirmação está pautada nas informações aferentes dos músculos cervicais (Montfoort *et al.*, 2006). Considerando estes estudos, é esperado que pessoas com comprometimento cervical traumático ou insidioso apresentem déficits importantes no controle da postura e dos movimentos dos olhos (Rix; J., 2001).

O comprometimento do sistema postural (somatossensorial) pode vir acompanhado de vertigem, otalgia, dor cervical, tensão muscular e, ainda, zumbidos agudos e alterações auditivas, os quais são explicados pelo distúrbio cervical (Wenngren *et al.*, 2002). Pela gravidade que desses problemas, bem como pelo desconforto que eles causam nos indivíduos, eles passaram a ser estudados em profundidade.

Esta situação é destacada por (Rix; J., 2001), que indica que uma maior atenção tem sido dada a essa disfunção conhecida como síndrome proprioceptiva cervical e segundo (Pinsault *et al.*, 2006) uma das formas para detectar essas alterações na propriocepção cervical é pela realização de um teste denominado Teste de Realocação Cervicocefálica (TRC) e será abordado na próxima seção.

2.1.2 Teste de realocação cervicocefálica - TRC

Descrita originalmente por (Revel; Andre-Deshays; Minguet, 1991) como uma medida do senso de posicionamento articular cervical, a técnica *Head Repositioning Accuracy* (HRA) ou Teste de Realocação Cervicocefálica (TRC) descreve um procedimento para verificar a capacidade de um paciente em mover sua cabeça lateralmente e retornar à posição anterior ao movimento (Raizah *et al.*, 2023).

Para realizar esse teste, utiliza-se um capacete com um feixe de luz que se projeta em um alvo a uma distância de 90 cm. Após marcada a posição inicial no alvo, o paciente (com a visão obstruída por óculos de proteção) é instruído a memorizar sua posição e, sem instruções de velocidade, movimentar sua cabeça em rotação máxima à esquerda, retornando em seguida à posição inicial, quando o ponto é novamente marcado. Esse mesmo teste foi realizado por (Raizah *et al.*, 2023), com distância de 3 metros.

Esses pontos que são sucessivamente marcados possibilitam calcular o erro de posicionamento global, sendo esses valores convertidos de centímetros para medidas angulares a partir do centro de rotação. São realizadas dez tentativas com rotações à esquerda e dez à direita, repetindo o mesmo procedimento. Os estudos (Revel; Andre-Deshays; Minguet, 1991), indicaram que indivíduos que apresentam um erro superior a 4,5° evidenciam a possibilidade de problema cervicocefálicos.

Com o tempo e estudos que se seguiram, o teste de reposicionamento TRC foi se consolidando como método efetivo de avaliação de função proprioceptiva. Vários autores realizaram testes de reposicionamento de cabeça e também apresentaram seus resultados, diferenciando o grupo de controle que foi analisado.

Em 1996, por exemplo, (Heikkila; Astrom, 1996) utilizaram o teste para avaliar um grupo de pacientes vítimas de lesão do tipo chicote causado por indivíduos envolvidos em acidentes de carro e teve problemas com o pescoço. O mesmo procedimento realizado por (Revel; Andre-Deshays; Minguet, 1991) foi repetido, e neste grupo de controle, foi encontrado um valor médio de 3,84 cm de erro no posicionamento articular em uma distância de 90 cm (neste estudo o resultado médio foi apresentado em centímetros).

Já (Treleaven, 2008), realizou em 2007 estudos que afirmam que os erros pequenos de 3 ou 4 graus, que correspondem a um valor entre 4 e 5 cm aproximadamente, já indicam um déficit de posicionamento articular cervical.

Outro estudo realizado com o objetivo de determinar a precisão do reposicionamento da cabeça foi realizado por (Rix; J., 2001), no qual o grupo de controle se diferenciava por indivíduos com dor crônica da coluna cervical não traumática, ou seja, sem lesão, e voluntários assintomáticos e intactos, sendo que os erros de posicionamento global mediano para indivíduos com dor cervical crônica foi de 4,2°. Outras análises também reforçam a utilidade da técnica e (Kristjansson; Treleaven, 2009) relatam várias pesquisas que foram realizadas com pacientes que possuíam recorrência ou progressão de sintomas com dor cervical. A proposta deste trabalho

de pesquisa é facilitar a aplicação desta técnica e para tanto se propõe a automatização de uma adaptação da técnica, conforme detalhamento feito na próxima seção.

2.1.3 Técnica Clínica TRC Adaptada

A adaptação da técnica TRC em tela, é a proposta por (Pinsault; Vuillerme; Pavan, 2008). Nela, três dispositivos são utilizados para realizar o procedimento. O primeiro deles é uma folha de papel milimetrado de 40 x 40 cm que é fixada em uma parede. No centro desta folha, desenham-se os eixos horizontal (x) e vertical (y), dividindo-a em 4 quadrantes. O segundo dos dispositivos é um capacete utilizado pelo paciente, sobre o qual é fixado um apontador laser, e o terceiro é uma máscara para ocultar a visão do paciente.

Com a utilização dos dispositivos, solicita-se a um indivíduo que se posicionasse a uma distância de 3 metros do alvo e que realizasse 10 repetições de movimentos de rotação à esquerda. Após cada uma das movimentações, o local onde o feixe ficava posicionado no alvo era marcado com uma caneta e a posição rotulada de acordo com o número da repetição e o sentido de rotação. Esse procedimento deveria ser repetido para a rotação à direita após aproximadamente 2 minutos de descanso (Pinsault *et al.*, 2008). A Figura 2 representado esquematicamente como o procedimento deveria ser realizado.

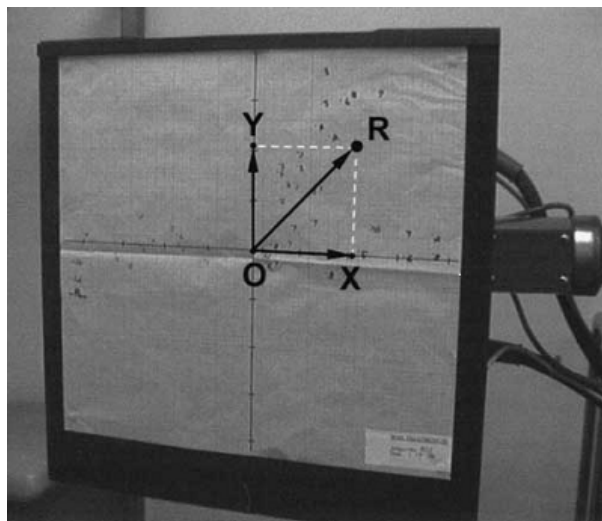
Figura 2 – Equipamentos utilizado no procedimento de teste na medição da sensibilidade cines-tésica cervicocefálica



Fonte:(Rix; J., 2001)

Na Figura 3, apresenta-se esquematicamente o procedimento utilizado para cálculos dos deslocamentos entre movimentos. O ponto 0 indica o centro do alvo (coordenadas 0,0) e a ele deve ser posicionado o feixe de laser no início do procedimento. O ponto R indica a posição em que o feixe de luz parou quando a cabeça foi reposicionada após um movimento. A distância 0 – R convertida em graus representa o erro global de posicionamento (R). A projeção horizontal (0 – X) e a projeção vertical (0 – Y) indicam os componentes horizontal e vertical do erro global segundo procedimento descrito em (Pinsault *et al.*, 2008).

Figura 3 – Coleta de dados do TRC no alvo e na análise inicial



Fonte:(Rix; J., 2001)

Vários trabalhos foram realizados pelo professor (Pinsault; Vuillerme; Pavan, 2008), utilizando esse método, com o objetivo de comparar indivíduos com controle assintomáticos intactos com aqueles pacientes com defeito labiríntico bilateral, pacientes com dor cervical crônica traumática e não traumática, entre outros (Pinsault *et al.*, 2006) (Pinsault *et al.*, 2008).

A proposta deste trabalho é adotar uma câmera para a realização da análise TRC sem a necessidade do indivíduo usar um capacete, apenas sendo necessária a utilização de uma máscara para manter os olhos vendados. Pode-se, assim, realizar o procedimento com equipamentos simples que facilitem a utilização por vários especialistas, permitindo também agilizar os procedimentos de observação dos dados coletados. Dessa forma, o sistema desenvolvido pode ser executado em um computador, fornecendo prontamente a análise completa dos dados armazenados.

Esta proposta vai ao encontro, por exemplo, de estudos realizados por (Brito *et al.*, 2021), que indicam que a análise de imagens tem demonstrado grande potencial na área médica, principalmente no auxílio ao diagnóstico de diversas doenças. A aplicação desses métodos pode melhorar ainda mais a capacidade de analisar dados de imagens multimodais complexos e a eficiência desses diagnósticos.

Para que seja possível compreender como a ferramenta desenvolvida funciona, se apresenta na próxima seção, conceitos associados a visualização digital de objetos.

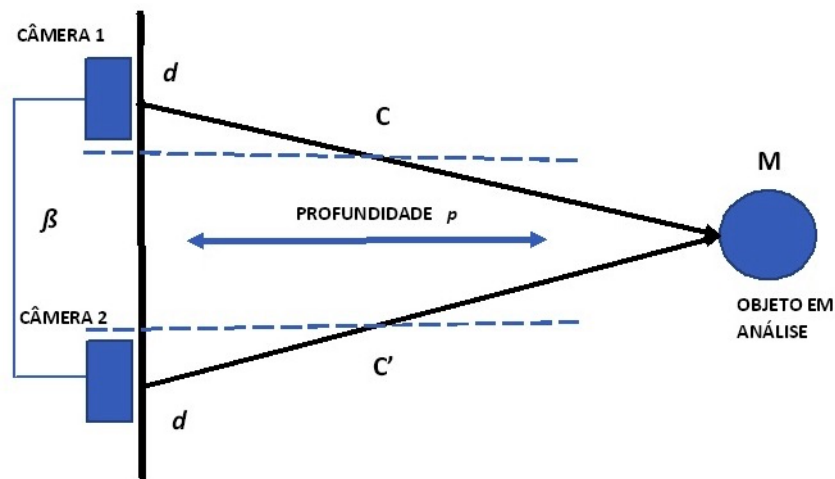
2.2 VISUALIZAÇÃO DIGITAL DE OBJETOS

O processo de visão digital de objetos é realizado por meio de sensores, que podem ser ativos ou passivos. Segundo (Rodrigues; Mencari, 1999), sensores ativos são aqueles que emitem um sinal – onda sonora, raios luminosos, onda eletromagnética – e que captam o mesmo sinal após este ter sido refletido em um objeto existente na cena. Por sua vez, sensores passivos são aqueles que captam sinais luminosos gerados por uma fonte externa ao sistema após reflexão em objeto na

cena.

Nas aplicações com sensores, além do tipo de sensor, outra característica importante da visualização, a percepção de profundidade, deve ser considerada, pois ela assegura a possibilidade de identificar o espaço tridimensional. Sistemas que integram esta característica são denominados sistemas com visão estereoscópica, ou de visão estéreo. Em sistemas de processamento de imagens, esta característica é integrada através da utilização de duas ou mais câmeras deslocadas horizontalmente uma da outra em um processo de triangulação, situação representada na Figura 4. Nesses sistemas, utiliza-se a estereoscopia para determinar as coordenadas 3D de um ponto M, que representa o objeto em análise. As câmeras são usadas para obter visões diferentes a partir dos sinais refletidos pelos objetos na cena, conforme apresentado na Figura 4 (FRANÇA, 2004).

Figura 4 – Visão estereoscópica ou projeção no plano de imagem com duas câmeras de um ponto M no espaço 3D



Fonte: (FRANÇA, 2004)

Por meio dos sinais refletidos, o dispositivo é capaz de calcular a posição 3D, o tamanho ou a velocidade dos objetos (Estereoscopia, 2022; Siscoutto; Soares, 2020). Além disso, ao comparar imagens de um ponto no espaço, capturadas sob diferentes pontos de vista, informações de profundidade podem ser obtidas a partir da distância relativa, ou disparidade, entre dois pontos que o representam (Stivanello *et al.*, 2008).

No processamento da imagem, essas distâncias são armazenadas em um mapa de disparidades que irá codificar a diferença nas coordenadas horizontais dos pontos correspondentes da imagem. As diferenças serão inversamente proporcionais à profundidade da cena no local do pixel correspondente (vision, 2020). Conforme apresentado na equação abaixo pode-se, portanto, afirmar que, considerando o caso de câmeras com planos de imagem coplanares e eixos óticos alinhados a profundidade p de um ponto, estará relacionada à sua disparidade d por (FRANÇA, 2004):

$$p = (\beta \times f) \frac{1}{d}$$

onde f é a distância focal das duas câmeras e β é a distância entre os centros de projeções da câmera 1 e da câmera 2 nos dois planos. Esta configuração torna difícil determinar com precisão a profundidade do ponto M se ele estiver muito distante das câmeras. Porém, ao se limitar as distâncias mínimas e máximas desse ponto através de uma calibração, é possível utilizar a disparidade para determinar a distância do ponto M (FRANÇA, 2004).

A automatização da realização do teste TRC com dispositivos que viabilizam a visualização digital de objetos, neste trabalho é realizada pelos sensores de movimento que são descritos a seguir.

2.3 SENSORES DE MOVIMENTO

Nesta seção, serão abordados diferentes tópicos associados aos Sensores de Movimento que estão no cerne do projeto de pesquisa apresentado neste documento. Inicialmente, apresenta-se um panorama histórico desses sensores, para, na sequência, discorrer-se sobre alguns dos principais sensores que foram desenvolvidos ao longo dos anos, realizando uma comparação entre eles. Em seguida, apresentam-se algumas aplicações que podem ser desenvolvidas com esses sensores. A sessão é finalizada com a apresentação da Estrutura Google MediaPipe, adotada na solução desenvolvida pela pesquisa.

2.3.1 Panorama histórico

No contexto dos dispositivos que envolvem arredores interativos, foco deste estudo, é importante destacar o papel desempenhado pela indústria de jogos, que introduziu em 2006 um dispositivo inovador: o controle remoto do console de jogos Wii produzido pela Nintendo. Este dispositivo permitia ao usuário manipular e interagir com a ação apresentada na tela por meio de um apontador portátil (semelhante a um mouse aéreo) que detectava gestos do jogador em três dimensões. Utilizando-se de uma comunicação Bluetooth e de um sensor infravermelho, esse controle utilizava três acelerômetros e outros sensores para detectar a orientação e a aceleração aproximadas do jogador, ao mesmo tempo em que servia como um dispositivo simulador de movimentos (Controller, 2020).

Outros dispositivos se seguiram, até que, em 1º de junho de 2009, sob o codinome Projeto Natal, a Microsoft deu início ao desenvolvimento que culminou, em 13 de junho de 2010, na criação do dispositivo Kinect, que começou a ser vendido como um acessório do console de jogos XBox 360, produzido pela mesma empresa. Este dispositivo, por suas características, revolucionou o mundo da tecnologia, pois, pela primeira vez, um controlador de jogo com diversas e interessantes novas funcionalidades utilizava uma porta USB aberta como meio de conexão, o que iria permitir uma interação facilitada com computadores pessoais (Melgar; Díez; Jaworski, 2013).

Como a Microsoft não havia lançado nenhum driver que permitisse acessar o dispositivo, uma corrida para acessar (*hackear*) suas funcionalidades foi lançada. A empresa Adafruit Industries

ofereceu uma recompensa de três mil dólares para a pessoa que conseguisse entregar uma biblioteca (*drivers*) de código aberto que pudesse ser utilizada para interagir com o dispositivo. O vencedor foi Héctor Martin, que desenvolveu uma biblioteca para o sistema operacional Linux que permitia que usuários tivessem acesso a dados produzidos pelo Kinect.

A disponibilização de uma biblioteca de código aberto causou novo frenesi na comunidade de tecnologia e levou ao surgimento de inúmeras aplicações para o sensor¹. Na sequência, empresas que trabalharam com a Microsoft para o desenvolvimento do dispositivo, ao perceberem as inúmeras e fantásticas possibilidades oferecidas pelo sensor, decidiram lançar seus próprios drivers. Uma dessas empresas foi a israelense PrimeSense² que era a empresa responsável pelo desenvolvimento do mecanismo de captação de imagens 3D do Kinect. Esta empresa lançou o pacote OpenNI que oferecia uma biblioteca e estrutura de programação para o Kinect. Na sequência, a PrimeSense estabeleceu uma parceria com outra empresa de tecnologia, a Asus³, para a produção do dispositivo Xtion que oferecia funcionalidades similares ao Kinect e que poderia ser utilizado no desenvolvimento de diversas aplicações.

Em 2011, a Microsoft entrou no jogo ao lançar uma ferramenta gratuita para o desenvolvimento de aplicações independentes com o Kinect. No ano seguinte, a empresa foi mais longe, lançando uma versão comercial do Kinect para o sistema operacional Windows, para uso independente do console de jogos. Esta versão era composta pelo próprio dispositivo e por uma ferramenta de desenvolvimento.

O lançamento do sensor Kinect abriu inúmeras e interessantes oportunidades tecnológicas fundamentadas na utilização de dispositivos de aquisição de movimento. Para facilitar a compreensão do funcionamento desses dispositivos, será feito o detalhamento de seu funcionamento, com a apresentação dos componentes de hardware que o compõem. Na sequência, outros que o sucederam serão apresentados.

2.3.2 Sensores Microsoft

A Microsoft produziu o primeiro sensor de movimento que alcançou uma inserção bastante grande no mercado, principalmente devido a seu preço e às funcionalidades que oferecia. O Microsoft Kinect, desenvolvido inicialmente como parte da plataforma Xbox, pode ser descrito como (SDK, 2020):

¹ O site <http://www.kinecthacks.com> desenvolvido na época e mantido até hoje apresenta alguns exemplos

² <https://en.wikipedia.org/wiki/PrimeSense>

³ <https://pt.wikipedia.org/wiki/asus>

[...] dispositivo de entrada de dados com detecção de movimento, lançado pela primeira vez em 2010. A tecnologia inclui um conjunto de hardware originalmente desenvolvido pela empresa PrimeSense, incorporando câmeras RGB, projetores infravermelhos, um conjunto de microfones e detectores que mapeiam a profundidade através de cálculos estruturados de luz ou de uma técnica denominada tempo de voo. Esses dispositivos, juntamente com software e inteligência artificial da Microsoft, permitem que o dispositivo realize reconhecimento de gestos em tempo real, reconhecimento de fala e detecção de esqueletos corporais para até quatro pessoas, entre outros recursos. Isso permite que o Kinect seja usado como um dispositivo de interface de usuário natural, com acesso viva-voz, para interagir com um sistema de computador.

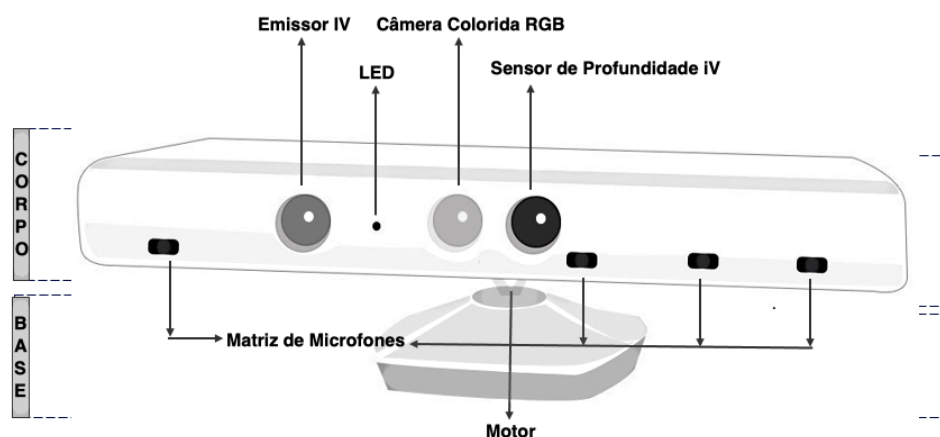
Para facilitar a compreensão do funcionamento de sensores de movimento, nesta seção serão apresentados os componentes que fazem parte de um sensor de movimento padrão. Como o Kinect foi o primeiro dispositivo deste tipo a ter descortinado novas possibilidades para aplicações de interação homem-computador, sua estrutura será utilizada.

Assim, o detalhamento dos componentes irá considerar primordialmente sua primeira versão, conhecida como Kinect for Xbox 360. Entretanto, como a tecnologia utilizada para a medição da profundidade de elementos em uma cena evoluiu, funcionalidades introduzidas na segunda versão do dispositivo, denominada Kinect V2 (c.f. 2.3.2.1), serão também abordadas.

2.3.2.1 Sensores de Movimento Microsoft: Componentes e Interação

A Figura 5 apresenta de forma esquemática os diferentes componentes que compunham o Kinect for Xbox 360. Esses componentes são detalhados na sequência.

Figura 5 – Componentes do Kinect - V1



Fonte: Adaptação de (Pham; Le; Vuillerme, 2016)

Matriz de Microfones

Trata-se de um grupo de quatro microfones auto direcionáveis, capazes de isolar as vozes do usuário do ruído da sala. Como eles estão posicionados em locais diferentes, a fonte de voz pode ser localizada a partir da comparação do atraso percebido no som captado em cada microfone. Esses microfones podem também ser utilizados para identificação do usuário e comandos de voz.

Emissor infravermelho (IV)

O conceito de disparidade foi utilizado através da tecnologia de **luz estruturada**. Nesta tecnologia, o dispositivo emite ativamente luz estruturada infravermelha (um padrão conhecido – geralmente grades ou barras horizontais) com o objetivo de criar uma representação em profundidade 3D do espaço à sua frente. Antes da utilização, a luz estruturada é projetada na cena para calibrar o dispositivo. Com a utilização, o padrão irá se deformar diferentemente ao atingir superfícies, permitindo, desta forma, que se calcule a profundidade e as informações de superfície dos objetos na cena. A tecnologia evoluiu nos modelos seguintes do Kinect, deixando de usar luz estruturada e passando a usar o princípio de **tempo de voo** (*Time-of-Flight - ToF*) que é apresentado a seguir.

A tecnologia de Tempo de Voo

Com o aparecimento da segunda versão do Kinect, a captura de dados de profundidade de elementos em uma cena, evoluiu e passou a utilizar a tecnologia de tempo de voo que trouxe diversas melhorias no processo, entre elas a possibilidade de obtenção de 25 a 30 quadros por segundo, tornando possível, com essa frequência, registrar o movimento do objeto na cena e captar mais informações 3D.

Conforme indicam (Mutto; Zanuttigh; Cortelazzo, 2013), para a medição de superfícies de cenas, vários sistemas de medição de distância são montados no sensor de tempo de voo, permitindo a varredura da cena. (Zennaro *et al.*, 2015) indicam também que esta tecnologia levou à obtenção de uma maior precisão da imagem e ainda permitiu a detecção, por exemplo, da frequência cardíaca do jogador, de sua expressão facial, posição, orientação e de 25 articulações individuais (verificar Figura 13). Assim, tornou-se possível medir com boa precisão distâncias, volumes, reconhecimento de gestos e rastreamento de objetos, o que ajudou a melhorar a experiência de realidade aumentada. Entretanto, conforme indicam (May *et al.*, 2008), os dados obtidos precisam ser avaliados, pois dependem de fatores externos ao sistema, como: iluminação ambiente, distâncias envolvidas, orientações e refletividade entre as superfícies dos objetos e o sensor, além da calibração do dispositivo. A presença de ruídos e erros sistemáticos deve também ser analisada de acordo com a aplicação.

Sensor de profundidade infravermelho (IV)

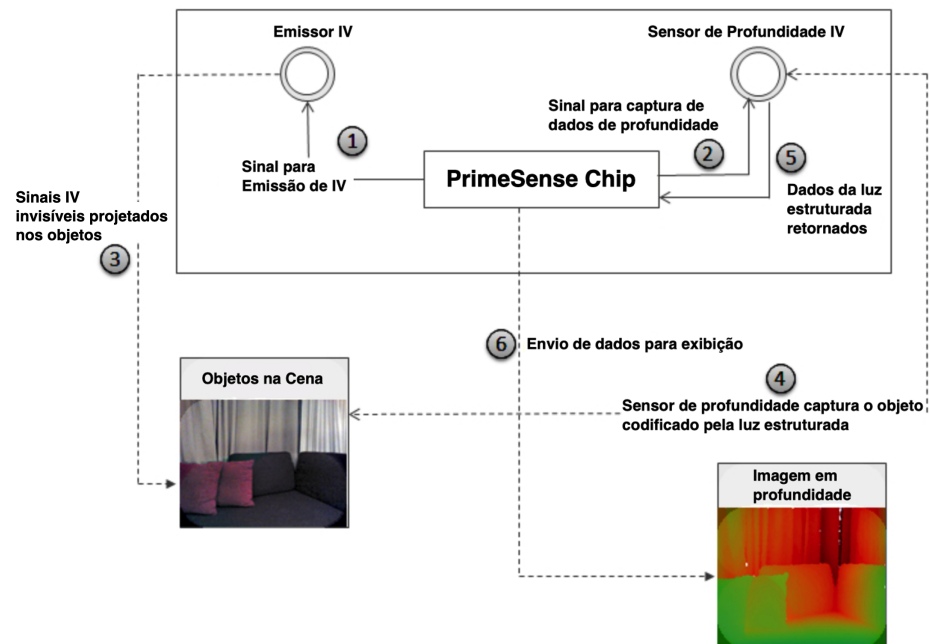
Este dispositivo é um sensor monocromático CMOS (*Complementary Metal Oxide Semiconductor*) que trabalha conjuntamente com o emissor infravermelho para capturar o reflexo do sinal por ele emitido. O CMOS é um dispositivo de carga acoplada, ou seja, um sensor semicondutor para captação de imagem, que captura a luz que chega à câmera e a qualidade e quantidade de informações capturadas estão relacionadas diretamente ao seu tamanho. A imagem captada pela objetiva é projetada, dentro da câmera, sobre um minúsculo painel semicondutor de células fotoelétricas (fotodiodos) onde cada um corresponde a um pixel do dispositivo e o sensor gera valores diferentes de voltagem para cada pixel conforme a intensidade de luz sobre ele. Nesses sensores existem fileiras de células fotoelétricas com a função de converter a luz em carga elétrica e executar uma varredura lendo cada pixel e enviando para o processador, que monta a imagem e a armazena em formato de mídia. Existem duas formas para efetuar a leitura através de seu tempo de exposição, chamadas de *Global Shutter* ou *Rolling Shutter* (obturador global ou obturador de rolamento). Nas câmeras CMOS que utilizam *Global Shutter*, os pixels do sensor são sensíveis à luz na mesma quantidade de tempo, sendo os valores do sinal de integração armazenados num buffer e depois libertados para transferência ao mesmo tempo. Outras câmeras CMOS que utilizam a tecnologia *Rolling Shutter* utilizam-se de uma arquitetura no controle de exposição em que o período de captura se move em uma janela de integração ao longo do sensor. A largura da janela de integração, conjuntamente com a velocidade de varredura, define o tempo da integração, em que cada linha de pixels é lida de forma consecutiva e é transferida da mesma forma. O problema é que sempre que existir um movimento, teremos uma distorção na imagem⁴. Por usar uma fonte de luz infravermelha ativa, o sensor Kinect funciona mesmo em condições de baixa iluminação.

Desta forma, é possível extrair imagens de profundidade em ambiente escuro para os nossos olhos (Kepski; Kwolek, 2013). O coração do dispositivo é um chip desenvolvido e licenciado para a Microsoft pela empresa israelense PrimeSense. A Figura 6 indica o princípio de funcionamento utilizado na captura da imagem em profundidade. Os números indicados nesta figura indicam os passos que são realizados para capturar a imagem em profundidade.

Quando se deseja capturar dados de profundidade, o chip PrimeSense enviará um sinal ao emissor de infravermelho para ativar a luz infravermelha (1). Na sequência, envia outro sinal ao sensor de profundidade de infravermelho para iniciar a captura de dados de profundidade (2). Este começará a enviar a luz infravermelha (invisível aos olhos humanos) para os objetos à frente do dispositivo (3). O sensor de profundidade de infravermelho começará então a ler os dados inferidos do objeto com base na distância dos pontos de reflexão individuais da luz (4) e os encaminhará para o chip PrimeSense (5). O chip irá na sequência analisar os dados capturados para então criar a imagem de profundidade que será utilizada (6) (PACKT, 2022).

⁴ <https://www.infaimon.com/pt-pt/enciclopedia-de-la-visao/rolling-shutter-e-global-shutter/>

Figura 6 – Captura da Imagem em Profundidade

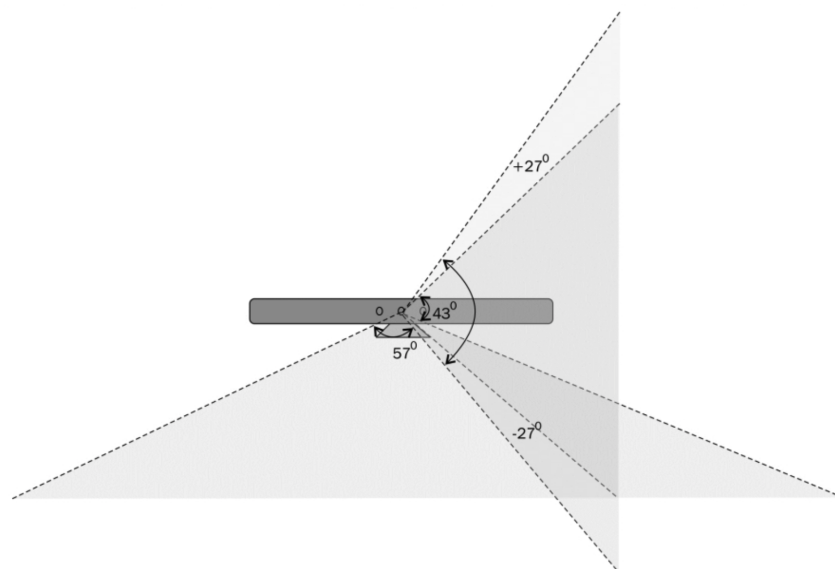


Fonte: adaptação de Packt (PACKT, 2022)

Motor

O motor do dispositivo é utilizado para que se possa acompanhar a movimentação de uma pessoa na cena, ou seja, ele opera para que as câmeras possam obter o melhor ângulo de visão. Conforme se pode observar na Figura 7, o Kinect pode ser inclinado verticalmente para cima ou para baixo em 27 graus. Quando se considera o ângulo de visão horizontal que é de 57 graus, o ângulo de visão vertical se reduz a 43 graus (PACKT, 2022).

Figura 7 – O Motor e os ângulos de visão



Fonte: tradução de Packt (PACKT, 2022)

Câmera RGB

Além de capturar as informações de profundidade da imagem, o Kinect captura também um fluxo de vídeo colorido (uma sucessão de quadros de imagens estáticas) e esta câmera é responsável por esta atividade. A primeira versão do Kinect era capaz de suportar, em uma velocidade de 30 quadros por segundo (*Frames Per Second* - FPS), uma resolução de 640 x 480 pixels e, em até 12 FPS, uma resolução máxima de 1280 x 960 pixels (PACKT, 2022).

A operação conjunta da câmera RGB com os sensores de profundidade assegurava uma importante possibilidade de detectar e gravar a movimentação de objetos em uma cena. Entretanto, o campo de visão do Kinect impunha uma limitação das aplicações a uma distância máxima de 4,5 m. A partir desta distância, os resultados deixavam de ser satisfatórios (Geerse; Coolen; Roerdink, 2015; Monterossas *et al.*, 2018).

Cabo e conector USB

Embora não representado na Figura 5, foi o conector USB que despertou o enorme interesse para o Kinect, pois, conforme mencionado, por se tratar de um padrão aberto, foi possível conectar o sensor a computadores pessoais. Através do cabo e conector, é que o fluxo de vídeo, informações de profundidade e fluxo de áudio são enviados ao computador (ou console de jogo). Como o consumo do Kinect é de 12W enquanto a potência máxima transmitida por um cabo USB normal está limitada a 2,5W, uma fonte de energia extra deve estar conectada ao Kinect para que o mesmo possa operar (PACKT, 2022).

2.3.2.2 Sensores de Movimento Microsoft: evolução

Inicialmente, o sensor de movimento Kinect foi criado com objetivo de ser utilizado em jogos, após uma concorrente que introduziu um sensor que facilitava a interação dos jogadores com os jogos. A empresa, então, lançou um console que, à época, foi considerado inovador, pois era capaz de captar os movimentos do jogador e utilizá-los nos jogos. O dispositivo contava com uma câmera RGB com resolução de 640 x 480 pixels e com um projetor de laser infravermelho, que projeta diversos pontos em uma superfície e criava modelos 3D, os quais eram lidos por um sensor.

A primeira versão do "Kinect for Windows" foi lançada em fevereiro de 2012 a partir de uma modificação da unidade do Xbox 360. Novos recursos não imaginados pela empresa, como a utilização em hospitais, centros de reabilitação, showrooms automobilísticos, supermercados e até em performance artísticas, incentivaram a Microsoft a lançar o Kinect for Windows junto com o SDK para uso comercial.

Na Figura 8, os dispositivos como Microsoft Kinect e suas versões incluindo o Microsoft Azure Connecta®, um dos últimos lançamentos, são alguns modelos de dispositivos sensores de movimento, que alcançaram grande penetração no mercado devido a seus preços e a suas funcionalidades. Eles suportam uma ampla variedade de sistemas operacionais e linguagens

Figura 8 – Câmeras de profundidade 3D da Microsoft



Fonte: (Corporation, 2024) (Corporation, 2014b)

de programação, permitindo uma extração mais detalhada de dados e interação com diferentes modelos de câmeras para análise e interpretação de dados de forma mais fácil. Esses dispositivos são conhecidos como câmeras de profundidade 3D (RGBD).

Lançado em 2019, o Microsoft Azure Connecta® é um novo dispositivo visto como uma nova evolução da tecnologia Kinect, que pretendia se estabelecer como uma evolução do Kinect para nuvem. Ele foi projetado principalmente para atender ao software empresarial Azure de Inteligência Artificial e tinha a proposta de combinar o sensor de profundidade com os serviços de Inteligência Artificial (IA) do Azure. O dispositivo foi projetado sobre a plataforma da nuvem Microsoft Azure com o objetivo de melhorar as percepções e operações e oferecia uma câmera melhor (12 Megapixel), além de apresentar um formato menor.

Desde a criação do Kinect, os dispositivos de manipulação de imagem receberam vários avanços tecnológicos e, atualmente, mesmo aqueles com custos mais baixos, possuem recursos de hardware avançados e apresentam um conjunto de melhorias significativas quando comparados ao sensor original da Microsoft.

Entre elas, pode-se mencionar a utilização nos dispositivos de processadores de vários núcleos (*Multi-core processor - MCP*), a integração com Sensores Móveis ⁵ (*Mobile Sensing – MS*), a presença de grande quantidade de memória interna (*Advanced Storage Capacity -ASC*), o uso de interface de usuário natural, a classificação de objetos (*Object Classification – OC*), a melhoria da interação humano-computador, a extração de recursos (*Feature Extration – FE*), a visão móvel (*Mobile Vision – MV*), o sistema de posicionamento global (*Global Positioning System – GPS*) e a possibilidade de adotar armazenamento externo. Os dispositivos passaram também a contar com mecanismos para assegurar uma conexão mais rápida à Internet (Avola *et al.*, 2019). Isso fez que fossem aumentadas ainda mais as possibilidades de utilização desses dispositivos, através do desenvolvimento de novas aplicações. Para explorar as diferentes possibilidades, será apresentado nesta seção um panorama das características e da utilização dos sensores de movimento.

2.3.3 Sensores Asus

Com o imenso interesse levantado pelo Kinect, com as propostas de bibliotecas independentes para acessar o dispositivo, a empresa PrimeSense decidiu colaborar com a Asus® para produzir

⁵ dispositivos que contam com um conjunto de sensores como câmeras de imagem, receptores com sistema de posicionamento global, acelerômetros e giroscópios, embutidos em um dispositivo

uma versão compatível com PC, especificamente para uso independente. Em março de 2012, como resultado desta colaboração, foi desenvolvido o Asus® Xtion Pro Live Figura 9.

O Asus Xtion Pro Live utilizava sensores infravermelho de tecnologia adaptativa de detecção de profundidade, detecção de imagem em cores e fluxo de áudio na captura de imagens, aumentando a precisão no rastreamento de usuários. O dispositivo apresentava um bom conjunto de ferramentas de desenvolvedor pois permitia ser acessado via OpenNI e podia ser utilizado na detecção de gestos, detecção de corpo inteiro e detecção de imagens em cores RGB, além de oferecer suporte para controle de voz e outros aplicativos de reconhecimento de voz.

Figura 9 – Câmeras de profundidade 3D da Asus



Fonte:(AsusTek, 2019)

Como solução à demanda do mercado, em 2013, a Asus passou a cooperar com a empresa ArtCreation 3D Technology Limited para produção de seus sensores e o Asus Xtion Pro Live foi substituído pelo Asus Pro Xtion LETV Figura 9, que era fabricado pela LETV e desenvolvido em conjunto com a Asus (AsusTek, 2019). Tal dispositivo apresentava um design similar ao Xtion Pro Live, oferecendo possibilidades de plug and play, e continuava compatível com o middleware OpenNI.

O Asus Xtion 2 suporta vários sistemas operacionais e possui um sensor de profundidade com boa resolução, além de uma alta resolução RGB de até 5MP (2592x1944), apresentando baixo consumo e sendo alimentado através do USB 3.0. Todos os sensores permitem desenvolvimentos em C++ e em Java e possuem kits de desenvolvimento de software OpenNI SDK.

2.3.4 Sensores Intel

Outra possibilidade foi oferecida pela Intel, que apresentou a linha Intel RealSense (Intel, 2019), com modelos de sensores com propostas para diferentes aplicações. A proposta da tecnologia Intel RealSense é fornecer tecnologias de profundidade e rastreamento, como as que vêm sendo utilizadas em drones autônomos, robôs, realidade aumentada (AR), realidade virtual (VR) e dispositivos domésticos, entre outros.

O sensor inicialmente oferecido pela Intel foi o RealSense D400 (Intel, 2019), que atualmente possui quatro modelos distintos de câmeras de profundidade para aplicações específicas. Lançadas em janeiro de 2018, a série Intel RealSense D400 apresenta boa resolução de profundidade e pode ser utilizada com boa precisão em ambientes internos ou externos, com qualquer tipo de exposição à luz. Todos os modelos possuem tecnologia de profundidade estéreo infravermelho ativo com uma resolução de saída de até 1920x1080 / 1280x720 e com uma taxa de quadros

de até 90 FPS e uma distância máxima de aproximadamente 10 metros. A precisão varia de acordo com a calibração e com as condições da iluminação. Internamente, o conjunto da placa que compõe as câmeras da linha D400 disponibiliza conectores do tipo USB-C e 3.1 Gen 1, que são denominadas RealSense Vision Processor D4.

Figura 10 – Câmeras de profundidade 3D Intel RealSense



Fonte:(Intel, 2020)

O modelo D415 é ideal para aplicações em que precisão e exatidão sejam determinantes e esse dispositivo trabalha com um campo de visão mais estreito e um obturador de rolamento que opera varrendo o ambiente de um lado para outro no processo de criação de uma imagem. Com essa característica, tal modelo é mais interessante na captura de imagem de pequenos objetos ou quando se necessita de precisão na geração da imagem. O modelo D435i apresentado na Figura 10 oferece uma câmera de profundidade com um campo de visão mais amplo e deve ser utilizado em aplicações que envolvam movimento rápido. Ele integra um obturador global no sensor de profundidade, ou seja, um dispositivo CMOS de maior velocidade, sendo ideal para o desenvolvimento de aplicações focadas no movimento e apresentando-se como uma solução de baixo custo que oferece uma boa profundidade através de uma solução estéreo. Tal dispositivo é indicado para aplicações como robótica ou AR / VR, com a adição de uma unidade de medida inercial (IMU) que é utilizada na detecção de movimentos e rotações em 6 graus de liberdade (6DoF). Essa funcionalidade permite um melhor alinhamento da luz refletida nos objetos ao combinar uma variedade de sensores com giroscópios para detectar rotação e movimento. Com a utilização da IMU em conjunto com o módulo e o processador de visão, é possível entender o próprio movimento da câmera. O uso da IMU facilita o processo de calibração em sistemas de digitalização portátil, aumentando sua aplicação em AR / VR. Lançado no mercado em junho de 2020, o modelo D455 Figura 10 tem como objetivo melhorar a imagem RGB, apresentando sensor com obturador global e câmera integrada a uma IMU. O dispositivo oferece uma melhor percepção de profundidade mesmo com a câmera em movimento, aumentando a chamada consciência ambiental. Todos os sensores da Intel RealSense podem ser programados utilizando o Intel RealSense SDK 2.0 que é oferecido em código aberto e permite a utilização de uma grande variedade de linguagens e plataformas populares de programação (INTEL, 2022).

2.3.5 Comparação de sensores de movimento

A tabela 1 apresenta diferentes características dos dispositivos abordados neste documento. Embora não apresentado no quadro, todos utilizam a tecnologia de infravermelho estéreo para reconhecimento de profundidade, sendo que, nos dispositivos da Intel, é utilizada a tecnologia

estéreo ativa que se diferencia por oferecer uma percepção de profundidade mais precisa, facilitando o reconhecimento quando o objeto ou o dispositivo está em movimento. A resolução das câmeras está entre QVGA (320x240) e Full HD (1920x1080). As três bibliotecas mencionadas neste documento (página 42) podem ser usadas para manipular os dispositivos e a principal característica técnica a diferenciá-los é a distância para utilização (Intervalo no quadro). O melhor desempenho neste fator é obtido com o Intel RealSense D435, que, não coincidentemente, entre os dispositivos analisados, está entre os mais modernos (Intel, 2020).

Tabela 1 – Visão geral dos sistemas de câmeras de profundidade

<u>Câmera</u>	<u>Resolução^a</u>	<u>Intervalo (m)</u>	<u>Programação</u>	<u>Notas / Interface</u>
Microsoft Kinect para Windows V2	512x424 a 30 qps e Full HD a 30qps	0,5 a 4,5	Kinect SDK	Adptador USB 3.0
Microsoft Azure Kinect DK	Full HD	0,25 a 5,46	Azure Kinect DK	Disponível nos EUA, Brasil, China e voltado a aplicações em nuvem USB-C
Asus Xtion Pro Live	QVGA a 60qps e VGA a 30qps	0,8 a 3,5	Kinect SDK, OpenNI	USB 2.0
Asus LETV Pro Xtion	QVGA a 60qps e VGA a 30qps	0,8 a 3,5	Kinect SDK, OpenNI	Produzido pela ArtCreation com base nos sensores Asus Xtion Pro Live USB 2.0
Asus Xtion 2	QVGA a 60qps e VGA a 30qps	0,8 a 3,5	OpenNI 2.2 SDK	Produzido pela ArtCreation com base nos sensores Asus Xtion Pro Live USB 3.0
Intel RealSense D415	VGA a 30qps e Full HD a 30 qps	0,16 a 10	RealSense SDK 2.0 / Nitrack	Campo de Visão mais estreito que o D435 USB-C
Intel RealSense D435	VGA a 30qps e Full HD a 30qps	0,105 a 10	RealSense SDK 2.0 / Nitrack	Intel RealSense Module D430 + RGB USB-C Campo de visão amplo
Intel RealSense D435i	VGA a 30qps e Full HD a 30 qps	0,105 a 10	RealSense SDK 2.0 / Nitrack	Intel RealSense Module D430 + RGB USB-C interpreta o próprio movimento
Intel RealSense D455	VGA a 30qps e Full HD a 30 qps	0,4 a 10	RealSense SDK 2.0 / Nitrack	Intel RealSense Module D455 + RGB USB-C dobro do alcance da D435i

^a QVGA: 320x240, VGA: 640x480, Full HD: 1920x1080

Fonte: Elaborado pelo autor

2.3.6 Interagindo com Sensores de Movimento

Conforme mencionado anteriormente, ao utilizar uma interface USB para conectar o Kinect a seu console de jogo, a Microsoft tornou possível a conexão do dispositivo a computadores pessoais. As características do Kinect e dos dispositivos que seguiram também facilitou enormemente o desenvolvimento de Interfaces Naturais de Usuário (*Natural User Interface – NUI*). Neste tipo de interface, a Interação Humano-Computador (*Human-computer Interaction - HCI*) deixa de ser realizada através de teclado e mouse e passa a ser feita considerando a fisiologia humana (uiux.pt, 2020). As NUI são um conjunto de interfaces que buscam tornar o mais natural a interação humano-computador, por meio da integração de tecnologia artificial baseada nas interações humanas, em que os gestos, movimentos do usuário, reconhecimento de fala, interação *touch-screen* e até mesmo do pensamento para o gerenciamento de dispositivos que podem ser móveis ou um computador e sua tela (Avola *et al.*, 2019) (Kramer, 2012).

A implementação dessas Interfaces Naturais de Usuário implicou no desenvolvimento de bibliotecas que dessem acesso às funcionalidades oferecidas pelo dispositivo. É por meio dessas bibliotecas, que normalmente recebem o nome de Interface de Programação de Aplicativos (*Application Programming Interface - API*), que o acesso é realizado, já que elas oferecem um conjunto de rotinas para comunicação com os mecanismos internos do dispositivo.

Um elemento ainda mais elaborado que pode ser utilizado no desenvolvimento de aplicativos para acesso aos dispositivos, é denominado Kit de Desenvolvimento de Software (*Software Development Kit - SDK*). Sendo normalmente disponibilizado pelas empresas fabricantes de sensores, ele irá incluir além da API necessária ao acesso, um ambiente completo e integrado de desenvolvimento de software, facilitando desta forma a implementação de aplicações. Foi o que fez a Microsoft com o lançamento da versão Kinect for Windows que incluía um SDK. Com o lançamento do sensor Azure, a Microsoft introduziu um novo SDK que adicionou novos suportes e melhorias em relação à versão inicial. Além dessa alternativa, duas outras, a OpenNI e Open Kinect, foram desenvolvidas e se consolidaram como as principais opções.

Complementarmente, deve-se mencionar a alternativa criada pela empresa 3DiVi Inc.⁶ que desenvolveu um site denominado NUITRACK⁷ através do qual ela oferecia um *middleware*⁸ que disponibilizava diversas funcionalidades. Trata-se de uma multilinguagem e uma multiplataforma onde as APIs incluem um conjunto de interfaces para o desenvolvimento de aplicativos, além de suportar diversos dispositivos como os sensores Kinect V1 e V2, os da linha da Asus Xtion Pro, Pro Live e Xtion 2, os sensores Astra S/Pro da Orbbec, além do Intel RealSense 415 e 435 (NUITRACK IN ACADEMIA, 2022).

Entretanto, conforme mencionado, o SDK inicial da Microsoft, a OpenNI e Open Kinect se consolidaram como alternativas mais utilizadas e por isso serão detalhadas nas próximas seções. Essas três alternativas apresentam diferenças na forma de suporte, opções de linguagem

⁶ <https://www.3divi.com/>

⁷ <https://nuitrack.com>

⁸ Software que fornece serviços específicos a outros softwares.

de programação e acessibilidade real ao dispositivo. Desta forma, é impossível afirmar que uma seja melhor que outra. As habilidades de desenvolvimento do usuário, o tipo de aplicação pretendido e o sistema operacional a ser usado, podem favorecer uma ou outra alternativa. Embora todas elas funcionem com Windows, com alguma versão do Linux e com MacOS, elas não são compatíveis e não podem ser usadas simultaneamente. É fundamental também considerar que com a interrupção da fabricação do Kinect em 2017, as iniciativas lançadas tiveram que evoluir para interagir com dispositivos de diversos fabricantes.

2.3.6.1 Kinect for Windows SDK 2.0 da Microsoft

Da mesma forma que para a apresentação dos componentes dos sensores de movimento se introduziu uma figura com os elementos do Kinect, apresenta-se na Figura 11 a arquitetura oferecida pelo Kinect for Windows SDK 2.0 para na sequência detalhar seus componentes.

O SDK oferece uma biblioteca (um conjunto de APIs) e ferramentas para auxiliar no desenvolvimento de aplicações (retângulo superior em rosa na Figura 11) para o Kinect. A biblioteca fornece recursos e serviços comuns a aplicações e encontra-se entre os sistemas operacionais e os aplicativos executados. Através da biblioteca, pode-se ter acesso de forma facilitada aos sensores do Kinect, desde o rastreamento de esqueleto e do movimento humano até o reconhecimento de voz e objetos. A biblioteca funcionará, portanto, com um mediador entre o dispositivo Kinect e a aplicação que será desenvolvida. Assim, por exemplo, quando for necessário acessar um sensor, a aplicação enviará uma chamada de API ao drive para controlar, desta forma, o acesso aos dados do sensor (Corporation, 2014b; Corporation, 2014a).

A partir das funcionalidades oferecidas pelos elementos da parte 1 Hardware da Figura 11, é possível acessar a porta USB e os dispositivos do Kinect (motor, câmeras e array de microfones). Os elementos da parte 2 dão acesso a drivers com o nível mais baixo de abstração e asseguram a interação com o software dos controladores do hardware do dispositivo. Esta camada está dividida em duas partes, Kernel Mode ou User Mode. A diferença básica entre esses dois níveis é que, embora seja mais fácil acessar os dispositivos através da camada User Mode, é mais eficiente por meio do nível Kernel Mode ⁹.

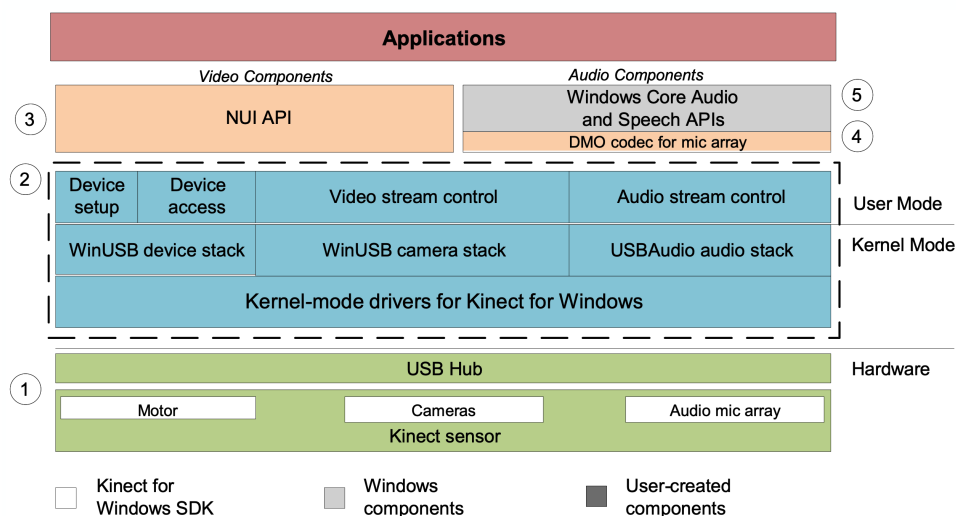
A parte 3 (NUI API) dá acesso à estrutura de vídeo e oferece elementos que podem ser usados para interagir com o Kinect por meio de Interfaces Naturais de Usuários. Já a partir dos elementos das partes 4 e 5, pode-se ter acesso às funcionalidades de áudio oferecidas pelo Kinect, as quais permitem às aplicações desenvolvidas ter acesso a recursos como transmissão de áudio resiliente a falhas em baixa latência e com confiabilidade aprimorada e segurança.

Deve-se, entretanto, realçar que o SDK pode ser usada exclusivamente em ambiente Windows. Além disso, ele está restrito às seguintes versões do Windows: Windows 8 (x64), Windows 8.1 (x64), Windows 8 Embedded Standard (x64) e Windows 8.1 Embedded Standard (x64) ¹⁰.

⁹ Para maiores detalhes, consultar <https://docs.microsoft.com/en-us/windows-hardware/drivers/audio/user-mode-versus-kernel-mode>

¹⁰ [https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782036\(v=ie8.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782036(v=ie8.10))

Figura 11 – Arquitetura do SDK do Kinect



Fonte: (Corporation, 2011)

2.3.6.2 OpenNI

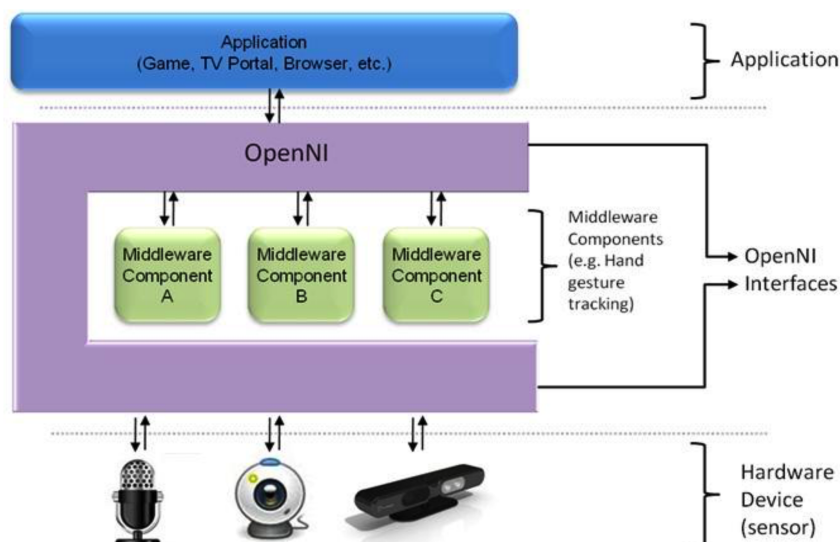
Quando de seu lançamento, conforme Rehem (SLIDEPLAYER INC, 2022), o termo OpenNI designava uma instituição sem fins lucrativos ao mesmo tempo que era uma marca registrada da empresa PrimeSense que havia desenvolvido parte do hardware do Kinect. A OpenNI buscava promover a interoperabilidade e compatibilidade entre aplicações, *middleware* e dispositivos de interação natural, atuando na verificação e certificação da conformidade das soluções de diferentes fabricantes e desenvolvedores aos padrões definidos pela OpenNI.

Para assegurar acesso de aplicações de usuário aos dispositivos internos de sensores de movimento e a seu *middleware*, o OpenNI definiu uma arquitetura (*framework*) padrão de comunicação. Esta arquitetura incluía a API apresentada na Figura 12. Além de oferecer acesso a dados brutos produzidos pelos sensores, ela assegurava a possibilidade de se construir dispositivos que funcionassem em qualquer aplicação compatível com o padrão proposto pelos fabricantes (SLIDEPLAYER INC, 2022).

Conforme se pode observar na Figura 12, a arquitetura OpenNI permitia que um usuário interagisse com dois tipos de componentes, os módulos de sensores e *middlewares*. Entre os primeiros, inserem-se a Câmera RGB, o Sensor de Profundidade IV, o Emissor IV e a Matriz de Microfones (c.f. Figura 5). Permitia também trabalhar com o *middleware* responsável pela: a-) geração do esqueleto da pessoa em cena; b-) geração da posição 3D da mão da pessoa na cena; c-) detecção de gestos pré-definidos realizados pela pessoa na cena; d-) geração de dados específicos da cena, como primeiro plano (*foreground*) e plano de fundo (*background*), coordenadas do chão e identificação dos diferentes atores na cena.

Por permitir trabalhar com dados representativos de cenas, independentemente dos fabricantes de sensores ou de um determinado *middleware*, a arquitetura viabilizou sua utilização com os mais diversos tipos de dispositivos (SLIDEPLAYER INC, 2022).

Figura 12 – A Arquitetura OpenNI



Fonte: OpenNI (OpenNI, 2014)

Uma importante mudança que ocorreu com a arquitetura, foi que, em novembro de 2013, a Apple comprou o PrimeSense e decidiu encerrar a iniciativa OpenNI. Entretanto, o desenvolvimento da arquitetura continuou ativo com o site sendo mantido atualmente pela empresa Occipital ¹¹. Atualmente as versões disponíveis na página criada para fornecer os recursos do OpenNI não recebem mais atualizações, mas a informação disponibilizada no site é que a hospedagem ocorrerá por um tempo indefinido. Entretanto, é possível ter acesso a versões da OpenNI 2.2 para MacOS, Windows e Linux.

2.3.6.3 OpenKinect

Diferentemente da OpenNi, o OpenKinect se caracteriza por ser uma comunidade de pessoas, e não uma biblioteca. Esta comunidade mantém um software denominado `libfreenect`, que teve como origem o driver Kinect inicial, e que foi construído através de um processo de engenharia reversa. Ele é mantido por meio de licenças Apache 2.0 ou, opcionalmente, GPLv2. A `libfreenect` oferece todos os elementos necessários para ativar, inicializar e comunicar dados com o Kinect. Isto inclui drivers e uma API multi-plataforma que funciona em Windows, Linux e OS X (Project, 2014b). Por meio do driver, é possível ter acesso aos seguintes componentes do Kinect: imagens RGB e Imagens de Profundidade, Motores, Acelerômetro, LED e Áudio. A interação pode ser realizada utilizando-se diferentes linguagens de programação, entre elas C, C++, C#, Python, Rubi, Actionscript e Java (via JNA-Java Native Access) (Project, 2014a). Caso se deseje interagir com o Kinect v2 (XBox One), a OpenKinect/libfreenect2 (Project, 2014a) deve ser utilizada.

¹¹ <https://occipital.com>

As principais diferenças entre a OpenKinect e a OpenNI estão relacionadas ao controle do motor que a `libfreenect` possui e a OpenNI não oferece. Entretanto, a OpenNi oferece suporte a NUI enquanto a OpenKinect não. Além disso, a OpenKinect não oferece uma solução direta para captar o esqueleto de uma pessoa na cena, mas o driver oferece acesso a dados brutos para a construção e rastreamento de um esqueleto. Desta forma, uma solução específica deve ser desenvolvida ou adaptada. Essas diferenças tendem a direcionar projetos para uma das duas bibliotecas. Fora essas diferenças, as funcionalidades oferecidas pelas duas são bastante similares.

A Tabela 2 1 apresenta um resumo comparativo entre as características das três alternativas apresentadas acima.

Tabela 2 – Comparação das Frameworks

Características	OpenKinect ^a	OpenNI	Microsoft Kinect SDK
Licença	LGPLv3+	LGPLv3+	Proprietária
Driver	libfreenect	Sensor Kinect Avin2	Microsoft Research Driver
Suporte sensor Primesense	Não	Sim	Não
Motor Vertical	Sim	Não	Sim
Áudio	Sim	Não	Sim
Detecção de Gestos	Exemplos OpenCV	NITE Middleware	Nenhum modulo Incorporado
Detecção de Esqueleto	Não	Sim	Sim
Analizador de Cenas	Não	Sim	Não

^a <https://pt.slideshare.net/marcosprod/apresentacao-kinect>

Fonte: Elaborado pelo autor

2.3.7 Aplicações com Sensores de Movimento

Com o aparecimento do Kinect, possibilidades de utilização de sensores de movimento no desenvolvimento de aplicações que envolviam interfaces naturais foram largamente sugeridas. As interfaces naturais introduziram aplicações em que o foco na utilização dos sensores de movimento é o indivíduo. A facilidade com que o utilizador se abstrai da tecnologia e foca no

objetivo de suas tarefas diárias é a grande razão para o aumento observado, então, de dispositivos utilizando essa interface. A seguir, se apresentam alguns exemplos.

A utilização da linguagem de sinais com o objetivo de comunicação com pessoas surdas ou mudas, cuja interação é realizada através de gestos, expressões faciais e corporais, câmeras podem ser utilizadas para facilitar o reconhecimento de uma informação que a pessoa queira transmitir. Assim, através do uso das aplicações com sensores de movimento, é possível atender às regras de comunicação por sinais, como a ASL (American Sign Language) ou LIBRAS (Linguagem Brasileira de Sinais). Como exemplos, pode-se citar o trabalho de (Alvarenga; Correa; Osório, 2012) onde se menciona a possibilidade de reconhecer gestos e de se comunicar através da utilização da linguagem de sinais ou, ainda, de utilizar tais dispositivos em substituição ao mouse no computador. O trabalho realizado por Zhao e Lun (Lun; Zhao, 2015), que envolve uma análise com a utilização do sensor de movimento do tipo Kinect na área da saúde e no reconhecimento de linguagem de sinais apresenta outra utilização similar.

Como outro registro, se pode mencionar o trabalho de (Lira, 2020) que analisa a utilização de interfaces naturais em celulares por meio dos sensores de profundidade que são acoplados à câmera principal do aparelho. Desta forma, o sensor "ToF" presente nos sensores de movimento pode ser utilizado para implementar novas aplicações, como determinar a distância entre objetos, a altura de uma parede ou a dimensão de um cômodo, além do reconhecimento facial e das usabilidades em realidade virtual e aumentada, representando, desta forma, mais um sensor explorável pelo usuário.

Já (Kang; Seo; Jung, 2011) realizaram estudos de controle em aplicativos 3D, nos quais é possível controlar o uso da informação de distância e da localização de juntas para reconhecimento de objetos. Por sua vez, (FRANÇA, 2004) criaram dois sistemas projetado para a interação gestual do usuário no controle de mapas geográficos 3D em um computador desktop, utilizando as tecnologias do Nintendo Wii e do Microsoft Kinect.

Outro trabalho que pode ser mencionado é o de (Farhadi-Niaki; GhasemAghaei; Arya, 2012) que utilizaram o Kinect na execução de simples e típicas tarefas de um desktop, utilizando-se de gestos e do braço como entrada de dados no lugar do mouse.

Conforme indicado em (uiux.pt, 2020), muito ainda pode ser feito utilizando-se voz, movimentos e pensamentos para acessibilidade a dispositivos. Existem alguns princípios na utilização da NUI que devem ser considerados, desde a utilização dos dedos na área de toque, a fisiologia, os excessos de repetições, as noções da tecnologia e a preocupação com ações involuntárias que podem acionar algum comando.

No contexto deste trabalho de pesquisa, vale também mencionar o trabalho de (Alvarenga; Correa; Osório, 2012) que apontam uma aplicação que envolve a aprendizagem e o reconhecimento de gestos a partir do rastreamento da posição de uma pessoa, da identificação da face e do mapeamento e movimentação de seu esqueleto. Já (Prediger *et al.*, 2018) detalham aplicações focadas na assistência a idosos em suas residências, utilizando um sistema de reconhecimento de gestos que interage de forma inteligente com os ambientes. Por meio desse sistema, é possível

controlar, por exemplo, a iluminação no ambiente do idoso, diminuindo a possibilidade de quedas e acionando o auxílio mais rapidamente, caso esses acidentes ocorram.

Por sua vez, (Hanke *et al.*, 2015) apresentam uma aplicação para reconhecimento de emoções a partir da imagem facial e da voz, sendo esta aplicação parte da arquitetura geral de um sistema que tem como objetivo disponibilizar uma interface de inteligência artificial para auxiliar os idosos, criando um parceiro de apoio virtual, para, por exemplo, desenvolver uma conversação e a partir da análise da voz e da imagem da face, identificar necessidades específicas de auxílio.

(Clark *et al.*, 2019) destacam outro tipo de aplicação voltada à identificação biométrica que pode ser utilizada com diferentes objetivos, entre eles a detecção do batimento cardíaco, a detecção de equilíbrio de um indivíduo, a identificação de acesso e a análise de distúrbios decorrentes de doenças neuro-degenerativas, entre outros. Os autores destacam que a análise de imagens é a principal ferramenta entre as técnicas de identificação biométrica em ambientes abertos e fechados. Destacam, também, que, embora a técnica seja considerada promissora por oferecer uma ferramenta de avaliação da função física, ela ainda desperta desconfiças, pois, para muitos, as câmeras 3D ainda não fornecem dados confiáveis.

Entre as técnicas de identificação biométrica empregadas, pode-se citar a análise de imagens da íris (Iris, 2021), os sistemas de reconhecimento facial que utilizam câmeras de vigilância (Ambeth-Kumar *et al.*, 2018) e as técnicas que, através da análise da variação da cor da face de um indivíduo em um vídeo, tornam possível realizar a medição da frequência cardíaca de forma passiva e remota (Xiaobai *et al.*, 2014)(Poh; McDuff; Picard, 2010).

Um outro tipo de aplicação, de certa forma relacionado ao trabalho apresentado neste estudo, envolve a avaliação do estado de pacientes com doenças neuro-degenerativas.¹² Vários estudos (Prediger *et al.*, 2018; Hanke *et al.*, 2015; Dranca *et al.*, 2018; Galna *et al.*, 2014) mostraram a viabilidade de avaliar, em ambientes controlados e de forma objetiva, a marcha deste tipo de paciente. Os estudos indicam que, utilizando sensores de movimento, é possível quantificar anormalidades associadas a doenças a partir da avaliação de deficiências na marcha e da análise da face.

Na doença de Parkinson, por exemplo, que tem como principais sintomas corporais tremores, rigidez, bradicinesia e dificuldade para caminhar, sensores de movimento foram utilizados para capturar a postura humana, para, a partir da análise dos dados, avaliar a gravidade da doença (Torres *et al.*, 2015). Um outro estudo verificou que, por meio de sensores de movimento, pode-se medir com precisão as características espaciais brutas de movimentos clinicamente relevantes, mas não movimentos menores, nos quais não é possível ter a mesma precisão (Galna *et al.*, 2014). Foram também realizados estudos para classificar os estágios da doença de Parkinson utilizando sensores de movimento a partir da avaliação de deficiências da marcha provocadas pelo uso prolongado de medicamentos como o Levodopa (Dranca *et al.*, 2018).

Ainda na mesma linha, pode-se mencionar o trabalho de (Gholami *et al.*, 2016) que atestam

¹² Neuro-degenerativo é um termo genérico na análise de doenças que afetam os neurônios do cérebro humano, causando problemas de movimento (ataxias) ou de função mental (Research, 2017)

que, na esclerose múltipla (EM), doença degenerativa em que o comprometimento da marcha é uma dificuldade predominante, a utilização de sensores de movimento em um ambiente controlado pode ajudar os médicos no processo de avaliação clínica.

Pode-se destacar também a tecnologia denominada Motion Capture (MoCap) que é utilizada em várias aplicações para medir e registrar a posição e a orientação do corpo humano em movimento, utilizando sensores de movimentos (Regazzoni; Vecchi; Rizzi, 2014). Neste contexto, (Dror *et al.*, 2014) alude sobre a importância da “capacidade de envolver o corpo físico em uma interação com a tecnologia” e criou um sistema não invasivo baseado no sensor de profundidade Kinect da Microsoft utilizando um novo algoritmo de rastreamento.

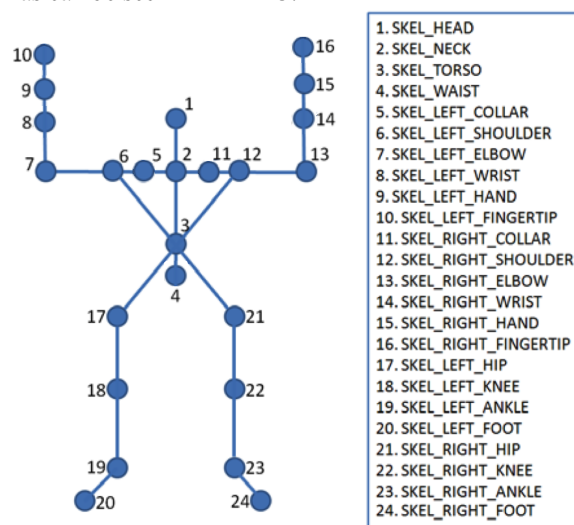
Uma outra aplicação é a proposta neste trabalho a qual propõe uma leitura de "processos de interação" para uso de especialistas no acompanhamento de acuidade proprioceptiva cervical. Esta aplicação envolve o rastreamento de um modelo esquelético que tem seu princípio de funcionamento apresentado na sequência.

2.3.7.1 Rastreamento de um modelo esquelético

Outra das funcionalidades extremamente interessantes introduzidas pelo Kinect, acessíveis através do SDK, foi a possibilidade de rastrear um modelo esquelético de uma pessoa que se movimenta sobre uma cena em observação. Isto é feito processando dados brutos de imagem produzidos pelo Kinect para obter mais informações além da profundidade de cada pixel. Esses dados são processados para estabelecer as posições (X, Y, Z) de várias articulações do esqueleto humano, como ombros, joelhos e cotovelos, além do centro de massa da pessoa. No processamento, as imagens de profundidade são analisadas usando algoritmos que envolvem transformações matriciais, aprendizado de máquina e outros meios (Webb; Ashley, 2012). Uma vez identificadas essas articulações, o software as conecta a um esqueleto humanoide e passa a rastrear sua posição na cena em tempo real (INTEL, 2022).

Para tanto, no Kinect original, era necessário realizar uma calibração inicial com a pessoa se posicionando estática, em frente ao dispositivo, com todo o corpo visível. A pessoa deveria ficar com os braços erguidos e mãos levantadas por cerca de 10 segundos (s). O Kinect processava então a imagem e rastreava a posição das articulações e membros, gerando uma representação similar à apresentada na Figura 13. Nesta figura, pode-se também observar o nome específico que cada uma das articulações recebe para utilização em aplicações computacionais.

Figura 13 – Rastreamento Esquelético no Kinect



Fonte: (Sinthanayothin; Wongwaen; Bholsithi, 2012)

Devido a diferentes questões mercadológicas, a Microsoft decidiu descontinuar a produção do Kinect. Ao mesmo tempo alguns fabricantes de sensores de movimento estavam também reduzindo sua participação no mercado. Decidiu-se então buscar uma alternativa de software que pudesse ser utilizada sem os kits de desenvolvimento apresentados na seção 2.3.6. Após uma busca por possibilidades, se decidiu testar a utilização da estrutura MediaPipe. Conforme se verá na sequência deste trabalho, esta escolha se mostrou apropriada e será apresentada no Capítulo 3.

2.3.8 Repetibilidade e Reprodutibilidade

Com o objetivo de garantir a confiabilidade das medições de forma estatística e dinâmica, utilizamos uma verificação de repetibilidade e reprodutibilidade das leituras.

Por repetibilidade, entende-se à capacidade do sistema de medição de obter resultados consistentes quando a mesma amostra é medida várias vezes sob as mesmas condições, entende-se um conceito de certa forma análogo ao da repetibilidade dos processos estocásticos, que pode ser definido como "a variação das medidas feitas repetidamente com o mesmo instrumento de medida e pelo mesmo operador, usando o mesmo procedimento, no mesmo item"(Campos, 2019). Ou seja, pretende-se identificar se a posição de um ponto da face de um boneco ou de um indivíduo que se mantém imóvel, capturado em quadros sucessivos, apresenta ou não variação.

Já a reprodutibilidade, relaciona-se à capacidade de obter resultados consistentes quando a mesma amostra é medida sob condições diferentes, como por exemplo diferentes operadores, diferentes equipamentos, ou em diferentes locais e momentos. Também em analogia aos processos estocásticos, pode ser definido como "a variação que pode ser esperada quando há mais de um operador (avaliador) fazendo medições"(Campos, 2019). Se o processo pode ser repetido por diferentes equipamentos e operadores.

A verificação de repetibilidade e reprodutibilidade ajuda a identificar e quantificar as fontes de variação no processo de medição, permitindo que você avalie a precisão e a confiabilidade do seu sistema de medição e faça ajustes para melhorar a qualidade e a consistência dos resultados obtidos.

2.3.9 Ferramentas para Implementação

No desenvolvimento do projeto, foram utilizadas as linguagens JavaScript, HTML e CCS. Além desta, foram empregadas também outras ferramentas, como MediaPipe, descrito acima, e TensorFlow. Elas são descritas na sequência

2.3.9.1 Linguagem JavaScript

A linguagem JavaScript foi criada em 1995 por Brendan Eich e é utilizada principalmente em construção de páginas interativas da Internet, sendo uma parte essencial dos aplicativos Web, junto com o HTML e CCS. O HTML é uma linguagem de marcação projetada para representar páginas da Web que permite escrever hipertexto e estruturar semanticamente uma página Web, formatar conteúdo, criar formulários de entrada ou incluir recursos de multimídia, como imagens e vídeos de computador (JAVASCRIPT, 2022). O CCS (*cascading styles sheets*) ou folha de estilo está presente desde a origem do *World Wide Web* (www), permitindo que os documentos sejam formatados e formando uma linguagem de computador que descreve a apresentação de documentos HTML (SHEETS, 2023).

O JavaScript está no centro das linguagens que são utilizadas pelos desenvolvedores Web e é utilizada pela grande maioria dos sites, sendo necessário aos navegadores da web um mecanismo para interpretá-lo, como, por exemplo, o Node.js ou o Deno.

2.3.9.2 Biblioteca TensorFlow

A biblioteca TensorFlow foi desenvolvida por pesquisadores e engenheiros que trabalham na equipe *Google Brain*, dentro da organização *Machine Intelligence Research* da Google e que realizam pesquisas em aprendizado de máquina. Essa ferramenta é utilizada em larga escala no tratamento de imagem, uma vez que utiliza eficientemente centenas de servidores que possuem unidades de processamento gráfico (*Graphics Processing Units – GPUs*) e unidade de processamento de tensor (*Tensor Processing Units – TPUs*) para treinamento e execução de modelos treinados (Abadi *et al.*, 2016). TPUs são circuitos integrados de aplicação específica, desenvolvidos especialmente pelo Google para acelerar cargas de trabalho de aprendizado de máquina, unidades de processamento de algoritmos de inteligência artificial (IA) e Redes Neurais Convolucionais (do inglês, *Convolutional Neural Networks - CNNs*). As GPUs são processadores dedicados ao processamento e à renderização de gráficos em tempo real.

A primeira versão publicada do TensorFlow (v1.0) data de 2018, época em que já possuía portabilidade para as linguagens Python, JavaScript, C++, Java, Go e Swift, sendo que nas versões mais recentes essa portabilidade foi mantida. O uso e a implementação da biblioteca são

incentivados na comunidade de desenvolvedores como forma de manter o suporte para outras linguagens de programação. O *TensorFlow.js models* é um repositório de modelos previamente treinados que foram portados para a linguagem JavaScript. Os modelos TensorFlow.js são executados no navegador e no ambiente Node.js⁷. Alguns dos modelos presentes no repositório são o PoseNet para estimativa de pose, o Coco SSD para o reconhecimento de objetos e o DeepLab V3 para a segmentação semântica, em que, a partir de uma imagem, retorna-se a descrição dos elementos presentes.

O TensorFlow é especializado em Redes Neurais Artificiais (*Artificial Neural Networks - ANNs*). ANN são uma ferramenta de inteligência computacional inspirada no sistema neural humano, que visa replicar o processo de comunicação entre neurônios (Basheer; Hajmeer, 2000), gerando um modelo de estruturas interconectadas capazes de realizar cálculos para processamento de dados e representação de conhecimento. Uma CNN é um algoritmo de aprendizado profundo que pode captar uma entrada, atribuir importância a vários aspectos/objetos do elemento de entrada e ser capaz de diferenciar ou classificar um do outro.¹³

2.4 SÍNTESE

Neste capítulo, foram detalhadas questões clínicas centrais para que se buscasse desenvolver este trabalho de pesquisa. Foi introduzida a importância da análise do comprometimento cervical traumático, e na sequência se apresentou o Teste de Realocação Cervicocefálica. Pôde-se observar que o teste TRC é uma das estratégias mais viáveis, por se tratar de um método simples e com custos razoáveis, pacientes acometidos de problemas cervicais.

Na sequência foram apresentados conceitos associados aos sensores de movimento que seriam utilizados para a captação de imagens e movimentos para a realização do teste TRC. Foi feito um detalhamento de diversos sensores e uma comparação entre as características de alguns deles. As alternativas de software que possibilitam uma utilização diversa dos sensores também foi apresentada. Foi relatado também que com a descontinuação do desenvolvimento de alguns dispositivos pelos fabricantes se buscou uma alternativa que pudesse ser utilizada de forma independente dos diferentes SDKs apresentados. O capítulo terminou com a apresentação de algumas ferramentas adicionais que foram utilizadas no desenvolvimento do trabalho.

Uma análise deste capítulo permite avaliar a quantidade de trabalho que foi necessária para que se chegasse a solução técnica que foi adotada. Mais do que simplesmente mencionar diversos dispositivos, o trabalho envolvido abordou uma análise profunda que passou pela utilização dos dispositivos e testes em diferentes situações para que fosse possível chegar a solução técnica adotada. Esta solução é apresentada no próximo capítulo.

¹³ <https://github.com/tensorflow/tensorflow>

3 METODOLOGIA EXPERIMENTAL E TECNOLOGIA ENVOLVIDA

Neste capítulo se apresentam questões relevantes ao desenvolvimento do sistema proposto para atingir os objetivos definidos na seção 1.2. A forma como os dados são capturados e a viabilidade do sistema proposto e suas dificuldades de implementação são abordadas.

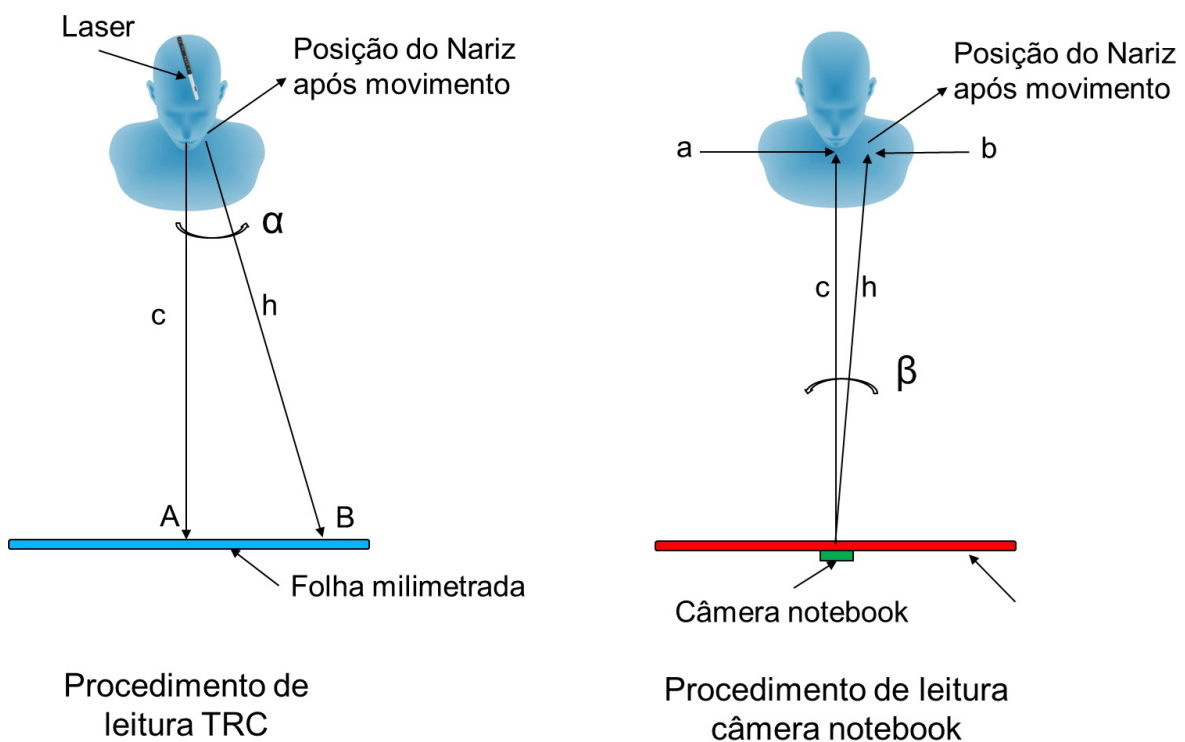
Apresenta-se também a estrutura de software que foi utilizada para a viabilização da ferramenta desenvolvida, passando por uma apresentação genérica das possibilidades oferecidas pelo MediaPipe. Dois dos modos de utilização, MediaPipe Pose e FaceMesh são detalhados, pois foram utilizados nos experimentos e na solução final adotada.

Na sequência se apresenta a arquitetura do sistema que foi desenvolvido para que o teste TRC pudesse ser aplicado bem como as diferentes versões construídas até se obter a versão final da ferramenta. O capítulo termina com uma breve avaliação da utilização da ferramenta.

3.1 PROCEDIMENTO DE LEITURA

A figura Figura 14 apresenta a forma em que as leituras são realizadas com o teste realocação cervicocefálica e como repetimos o processo utilizando o sistema. No teste TRC atual, quando o indivíduo movimenta a cabeça o laser acompanha o movimento e podemos visualizar na folha milimetrada o quanto representa de deslocamento, para o cálculo é utilizado Pitágoras.

Figura 14 – Procedimento de leitura TRC e câmera

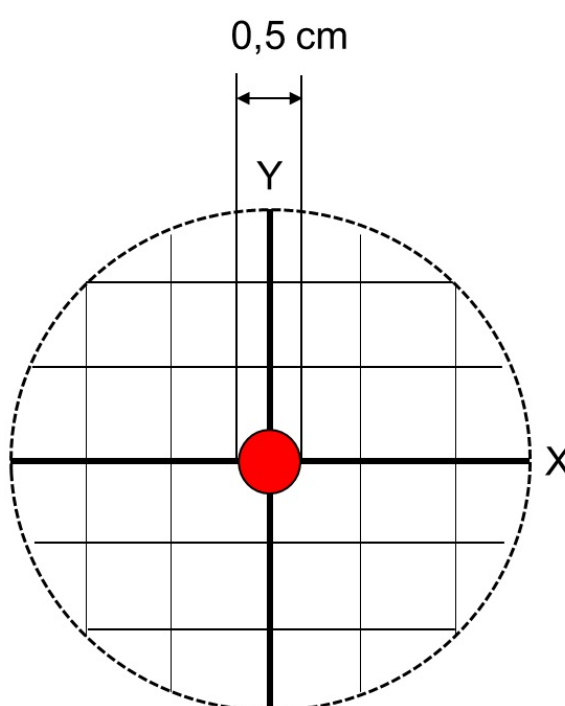


Fonte: Elaborado pelo autor

Dessa forma podemos calcular, que na distância de 80 cm o deslocamento da face sendo de 1 cm. A indicação do laser é de aproximadamente 5 cm, o que significa um deslocamento de aproximadamente $3,6^\circ$. A proposta é repetir essa medição com o MediaPipe e uma câmera.

O processo realizado pelo sistema, utiliza o mesmo princípio só que agora a identificação do movimento é realizado pela câmera do notebook. Utilizando Pitágoras, podemos determinar a distância do movimento da face do indivíduo sendo de 1 cm em frente a câmera e a indicação do sistema deve ser de aproximadamente $3,6^\circ$. Outro ponto a ser considerado é que o laser a uma distância de 80 cm ocupa aproximadamente 0,5 cm da indicação na folha milimetrada, como apresentado na Figura 15.

Figura 15 – Folha milimetrada com laser indicando posição



Fonte: Elaborado pelo autor)

Considerando um erro de indicação para valores maior que 10 % do deslocamentos, por exemplo de 5 cm, um erro de 0,5 cm, é admissível. Foi adotado o nível de significância igual a 0,05 (5%). Testes de hipóteses serão utilizados nas comparações das médias (Pereira, 2015). Dessa forma utilizamos o ANOVA (Análise de variância), para comparar as medias nos testes realizados. Entre as medições de forma estática é considerado que a indicação de 0,5 cm ainda como um valor aceitável de precisão. Realizamos cálculos com os valores de leituras obtidos e determinamos a média, a variância e o desvio padrão. Dessa forma quando o resultado obtido é um p maior que 0,05 podemos afirmar que as variâncias são aceitáveis.

3.2 A ESTRUTURA MEDIAPIPE

A construção de aplicativos ou sistemas com a capacidade de perceber o mundo ao redor é um grande desafio. A visão computacional, que permite essa funcionalidade, é uma das áreas mais inovadoras e complexas da atualidade. Sua utilização efetiva requer uma compreensão profunda da percepção visual humana para, a partir dela, "enxergar" e capturar a realidade externa por meio de diversos sensores. É também essencial desenvolver algoritmos e modelos de aprendizado de máquina, juntamente com demonstrações e protótipos, para que a máquina possa capturar, processar e interpretar os raios de luz do mundo externo (Melo, 2023).

A visão computacional visa extrair informações significativas de vídeos e imagens, transformando-os em dados úteis. Este processo envolve o uso de modelos de previsão e análises exploratórias de imagens. Desde 2012, o Google tem utilizado a Estrutura MediaPipe em diversos produtos, inicialmente para a análise de áudio e vídeo em tempo real no YouTube. Em 2019, o Google lançou o projeto MediaPipe, cujo código foi aberto em junho de 2020, atraindo a atenção de desenvolvedores que utilizam câmeras de profundidade em seus aplicativos.

A MediaPipe é uma estrutura para construir pipelines e realizar inferências em dados arbitrários. Ela permite a percepção por meio de componentes gráficos modulares, inferência de modelos, algoritmos de processamento de mídia e transformação de dados (Castro *et al.*, 2020) (Lugaresi *et al.*, 2019).

Conforme mencionado, as tecnologias de detecção de imagem estéreo utilizam sensores com câmeras para calcular a profundidade e o posicionamento de pessoas ou objetos. Isso permite que os dispositivos visualizem, compreendam, interajam e aprendam com seu ambiente de forma tridimensional (MediaPipe, 2024).

O uso do MediaPipe permite que desenvolvedores se concentrem no algoritmo ou modelo, melhorando iterativamente seus aplicativos (Lugaresi *et al.*, 2019). Construir sistemas ou aplicativos que percebam o ambiente ao redor é um desafio que envolve várias etapas. A estrutura do MediaPipe oferece respostas de qualidade para muitos desses desafios (MediaPipe, 2024). No entanto, é importante ressaltar que a identificação biométrica usando a biblioteca MediaPipe é considerada uma tarefa complexa. A visão computacional é uma solução de alta fidelidade e sua aplicação para detectar movimentos da face deve ser utilizada com cuidado (MediaPipe, 2024).

A tecnologia MediaPipe oferece diversos modelos para aplicações de visão computacional e processamento de mídia. Esses modelos são amplamente utilizados em diversas aplicações, desde realidade aumentada e jogos até diagnósticos médicos e interação homem-máquina. Alguns dos principais modelos disponíveis incluem:

- *Face Detection*: Detecção de rostos em imagens e vídeos.
- *Face Mesh*: Reconstrução de uma malha 3D detalhada do rosto a partir de uma imagem.
- *MediaPipe Iris*: Identificação da íris e medição de distância interpupilar.

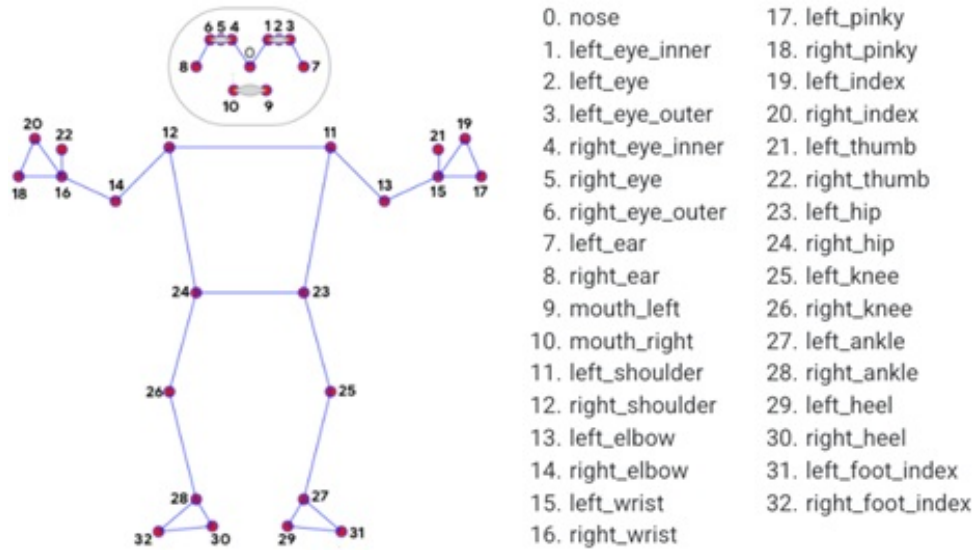
- *Hands Detection*: Detecção e rastreamento de mãos e dedos.
- *Pose Detection*: Detecção e rastreamento de poses corporais, incluindo 33 pontos de referência para articulações e partes do corpo.
- *Hair Segmentation*: Segmentação do cabelo em imagens e vídeos.
- *Object Detection*: Detecção e reconhecimento de objetos em tempo real.
- *Holistic*: Integração de vários modelos (face, mãos, pose) para uma percepção holística do corpo humano.
- *Selfie Segmentation*: Segmentação do corpo humano em selfies, separando o fundo do sujeito.
- *BlazeFace*: Um modelo rápido de detecção de rosto otimizado para dispositivos móveis.
- *BlazePose*: Um modelo de detecção de pose de corpo inteiro otimizado para desempenho em dispositivos móveis.
- *Multi-Hand Tracking*: Rastreamento de várias mãos simultaneamente.

Este trabalho se concentrou em dois dos modelos disponíveis. O primeiro deles, o MediaPipe Pose, ou BlazePose (BP) captura os 33 pontos de referência apresentados na Figura 16). Cada ponto é composto por dados dos eixos x, y e z, sendo que os valores de x e y são normalizados [0,0 ; 1,0] respectivamente para a largura e altura da imagem, com z representando sua profundidade (quanto mais próximo da câmera, menor será o seu valor). Conforme se poderá observar na sequência do trabalho, somente o ponto 0 indicado na (c.f. Figura 16) foi utilizado para a captação da posição e realização da análise.

O BP realiza o rastreamento do corpo em alta fidelidade para inferir os 33 pontos de referência, sendo inspirado no modelo *BlazeFace* (Bazarevsky *et al.*, 2020). Trata-se de um detector de face e corpo leve e de bom desempenho adaptado para inferência utilizando-se GPU móvel. Esse BP alimenta a API de detecção de corpo e face do kit de aprendizado de máquina que é uma solução leve e versátil para que desenvolvedores de aplicativos possam detectar a posição de um corpo visualizado em tempo real a partir de um vídeo contínuo ou de uma imagem estática.

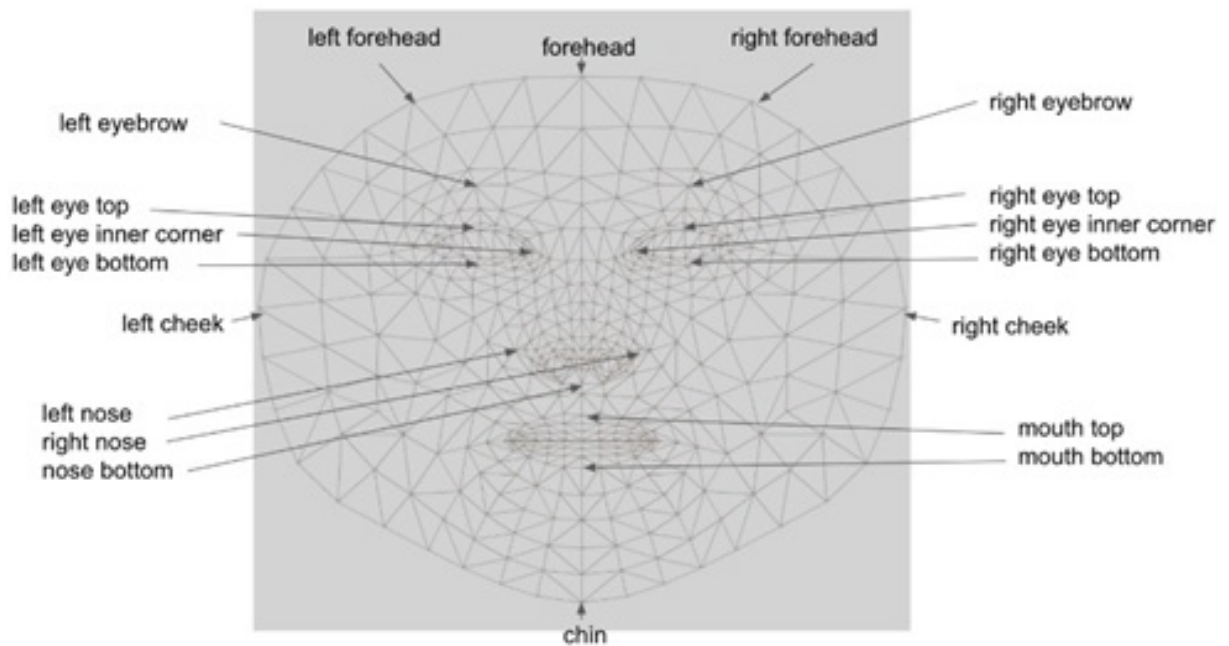
O segundo modelo, o MediaPipe Face Mesh, é uma solução que estima 468 marcos faciais 3D em tempo real, conforme detalhado na Figura 17. Tal modelo utiliza também o aprendizado de máquina para inferir uma superfície facial 3D, com a utilização de uma única câmera sem a necessidade do sensor de profundidade. Par tanto, um módulo do FaceMesh denominado Transformação de face (*module Face Transform*) é utilizado. Este módulo, por sua vez, estabelece um espaço 3D métrico e usa as posições da tela de referência para estimar primitivas de face 3D (MediaPipe, 2023).

Figura 16 – Pontos de referência do corpo utilizando o MediaPipe pose



Fonte:(MediaPipe, 2024)

Figura 17 – 468 points face texture mesh

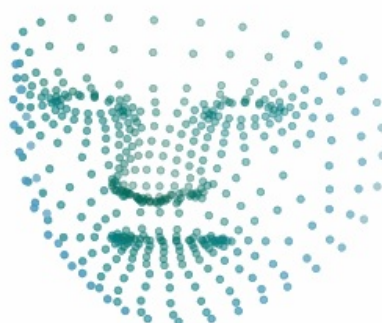


Fonte:(MediaPipe, 2024)

A Figura 18 apresenta de forma esquemática, a máscara que permite visualizar os pontos identificados pelo Face Mesh. O MediaPipe se destaca na análise semântica superior, pois opera em um nível muito mais alto na resolução de tarefas como inferência de modelos, processamento de mídia e transformações de dados permitindo um comportamento mais dinâmico e complexo o que acelera a GPU, no qual uma entrada pode gerar zero, uma ou várias saídas (Bazarevsky *et al.*, 2020).

A estrutura MediaPipe foi também projetada para oferecer soluções de aprendizado de máquina personalizáveis e em multiplataforma de código aberto para mídia ao vivo e streaming,

Figura 18 – Malha de atenção: Visão geral do modelo



Fonte:(MediaPipe, 2024)

abordando os desafios do desenvolvedor com uma estrutura de baixo custo e boa precisão. Uma das principais aplicações é a prototipagem rápida de pipelines de percepção com modelos de inferência e outros componentes reutilizáveis, facilitando a implantação tecnológica de percepção em aplicativos e uma ampla variedade de plataformas de hardwares, o que favorece, ainda, melhorias incrementais nos pipelines de percepção com uma ótima linguagem de configuração e ferramentas de avaliação. São disponibilizados modelos que podem ser implementados em diversas plataformas, tais como Windows, Linux, Web, Android e iOS. Sua integração pode ser feita por meio de código desenvolvido em diversas linguagens como Python, Java, C/C++ e, ainda, em linguagens utilizadas para aplicativos como Android e iOS (MediaPipe, 2024) (Lugaresi *et al.*, 2019).

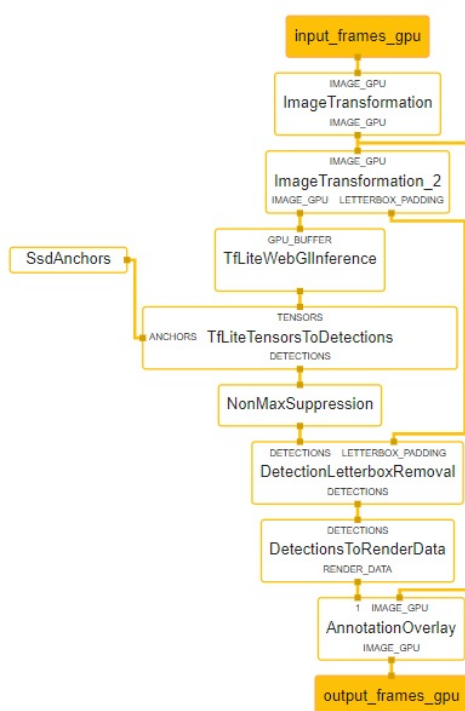
Na academia ou na indústria, a análise de mídia é uma área ativa de pesquisa e um arquivo de mídia ou entrada de câmera, contendo áudio e vídeo, é extraído de forma separada por meio do decodificador de mídia para ser analisado separadamente. Nessa análise, normalmente se utiliza um sistema de aprendizado de máquina em larga escala, como o oferecido pelo "TensorFlow"(Abadi *et al.*, 2016), (Agarwal *et al.*, 2015) (c.f. seção 2.3.9.2) que representa suas redes neurais em forma de gráficos direcionados e com nós simples e determinísticos, em que uma entrada gera uma saída, permitindo uma melhor execução do gráfico de computação.

A Figura 19 apresenta o fluxograma de captação de imagem com as caixas transparentes representando nós de computação (calculadoras) em um gráfico MediaPipe. A execução da calculadora usando um sistema de agendamento abrangente permite que o MediaPipe tenha suporte para várias plataformas e fornece ferramentas para avaliar o desempenho gráfico. Desta forma, é possível ajudar o desenvolvedor rapidamente na implementação de protótipos e na execução dos aplicativos de percepção com eficiência, sendo o ecossistema de calculadoras e gráficos reutilizáveis exemplos de sucesso na utilização do MediaPipe.

O gráfico da Figura 19 consiste em dois subgráficos, um para face detecção e outro para cálculo de pontos de referência. Uma chave otimização que o MediaPipe fornece é que o detector

de face só é executado conforme necessário (raramente), economizando computação.

Figura 19 – Detecção de face usando MediaPipe



Fonte:(MediaPipe, 2024)

Na Figura 19 as caixas sólidas representam as entradas / saídas externas do gráfico. Já as linhas que entram na parte superior e saem da parte inferior dos nós representam os fluxos de entrada e saída, respectivamente. Um pipeline é como um gráfico direcionado de componentes, no qual cada componente é uma calculadora, permitindo desta forma, que um desenvolvedor crie um protótipo de pipeline de forma incremental. Ou seja, uma vez definido o que se deseja captar na imagem o processo de tratamento da imagem se repete.

Os desenvolvedores podem, portanto, definir calculadoras personalizadas e, desta forma, enquanto o gráfico é analisado pelas calculadoras em paralelo, cada calculadora executa no máximo um thread¹ por vez. Assim, com essa restrição, junto com a impossibilidade de mudança dos pacotes de dados, as calculadoras personalizadas podem ser definidas sem conhecimento especializado em programação multithread².

Um pacote consiste em um timestamp³ numérico e um ponteiro compartilhado, os quais podem ser de qualquer tipo C++, sendo que o tipo de carga também é referido como tipo de pacote. Os pacotes são tratados como classes de valor e podem ser copiados, pois cada cópia tem seu próprio carimbo de data e hora. (Bazarevsky *et al.*, 2020), (Lugaresi *et al.*, 2019).

¹ um pequeno programa que trabalha como um subsistema, sendo uma forma de um processo se auto-dividir em duas ou mais tarefas

² é a capacidade que o sistema operacional possui de executar vários threads simultaneamente sem que uma interfira na outra

³ é uma sequência de caracteres ou informações codificadas que identificam quando um determinado evento ocorreu, geralmente fornecendo data e hora do dia

Construir uma pipeline de percepção como um gráfico de calculadoras reutilizáveis é uma das características oferecidas pelo MediaPipe, pois a pipeline é flexível, permitindo inserir ou substituir qualquer nó do grafo e customizá-lo.

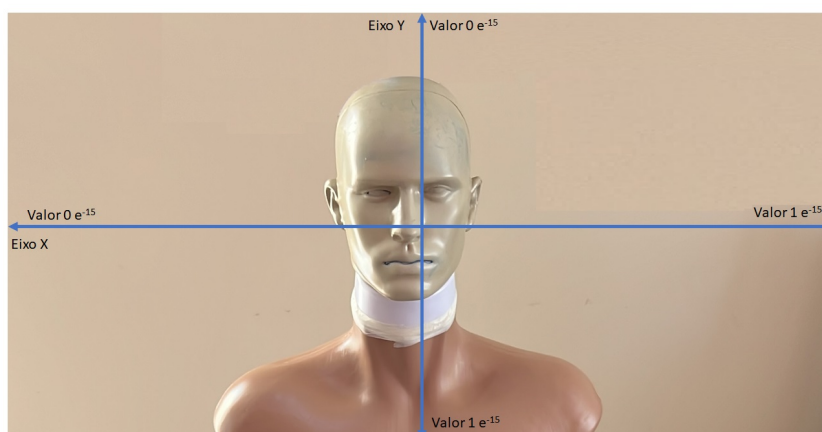
3.2.1 Medição com o MediaPipe

Com o MediaPipe é possível capturar pontos em uma imagem obtida por meio de câmeras para então verificar o posicionamento dos pontos captados, sendo que essa análise pode ser realizada de forma *online* ou *offline*.

Na visão computacional, a estimativa de pose humana (pontos de referencia do corpo) é uma tarefa importante e que tem aumentado suas aplicações como a interação humano-computador, os jogos, a vida assistida, os efeitos especiais, entre outros. As redes neurais na era do aprendizado profundo (*deep learning*) facilitaram essa progressão. Estimar a localização das articulações de um ou mais corpos humanos em espaço 2D ou 3D em uma única imagem é um dos objetivos buscados, pois as articulações conectadas ajudam a descrever a pose do indivíduo (Campos, 2019).

Com a proposta de facilitar o entendimento da forma como os pacotes de dados podem ser utilizados, apresenta-se na Figura 20 um único quadro com um boneco e se indica a forma como os dados de pontos da tela são armazenados e enviados ao programa. Pode-se observar que o MediaPipe divide a tela no eixo vertical (X) e horizontal (Y). A indicação de posicionamento dos pontos é realizada com 15 casas após a virgula. Dessa forma, os pontos variam, tanto no eixo vertical quanto no eixo horizontal, de 0 e-15 até o valor de 1 e-15.

Figura 20 – Figura pontos MediaPipe



Fonte: Elaborado pelo autor

É importante salientar que na ferramenta utilizada não se utiliza o eixo (Z), para indicação de distância, pois, nos testes realizados, se observou que o valor obtido sofre variação para cada modelo de câmera, o que obrigaria a realizar uma calibragem quando da mudança de câmera. Como os valores de (X) e (Y) já atendem à proposta, pois necessitamos somente de uma única posição na tela e se definiu que todos os experimentos fossem realizados a uma distância de 80

cm dessa forma o eixo (Z) é fixo. Dessa forma, o valor obtido para o deslocamento pode ser capturado em tempo real. Foram realizados testes para comprovar essa informação.

Nos experimentos, se optou por utilizar seis casas após a vírgula, conforme a regra de arredondamento 5891 da ABNT (TÉCNICAS, 1977), com a proposta de ter uma melhor visualização dos pontos capturados. Como se realizou uma comparação entre as medições nos experimentos, é importante observar as diferenças entre os valores lidos, sendo que, com um número composto por várias casas, esse processo é dificultado. Para a indicação do ponto de intersecção entre o eixo X e o eixo Y, tem-se o valor de 0,5 e-15 nos dois eixos.

3.2.2 Aplicações com MediaPipe

No âmbito do trabalho de pesquisa apresentado neste documento, as aplicações que têm maior importância são aquelas desenvolvidas com a Estrutura MediaPipe que foi apresentada na seção 3.2. Uma aplicação MediaPipe pode ser facilmente implementada com poucas linhas de código e algumas demonstrações desenvolvidas em Python ou JavaScript já estão prontas no site da google. Com a linguagem Python são apresentados exemplos de detecção da face, de malha facial, das mãos, holística, de objetos, de postura e de segmentação de selfie, que podem ser utilizadas pelo usuário na construção de seus próprios aplicativos (PYTHON, 2022). Em JavaScript, as mesmas opções são disponibilizadas acompanhadas de um maior pacote de demonstrações de seu funcionamento para auxiliar o usuário da linguagem (MediaPipe, 2024) (JAVASCRIPT, 2022).

Existem várias publicações, como vídeos, eventos e comunidades que apresentam utilizações do MediaPipe, o qual atualmente se encontra na versão alfa v0.7. Entre as publicações, temos: SignAll SDK, interface de linguagem de sinais usando MediaPipe (MediaPipe, 2020); MediaPipe Holistic, previsão simultânea de rosto, mãos e pose no dispositivo (HOLISTIC, 2022); MediaPipe Iris, Rastreamento de pose corporal em tempo real no dispositivo (IRIS, 2022), além de muitas outras publicadas no Google Research.

3.3 ARQUITETURA DO SISTEMA DESENVOLVIDO

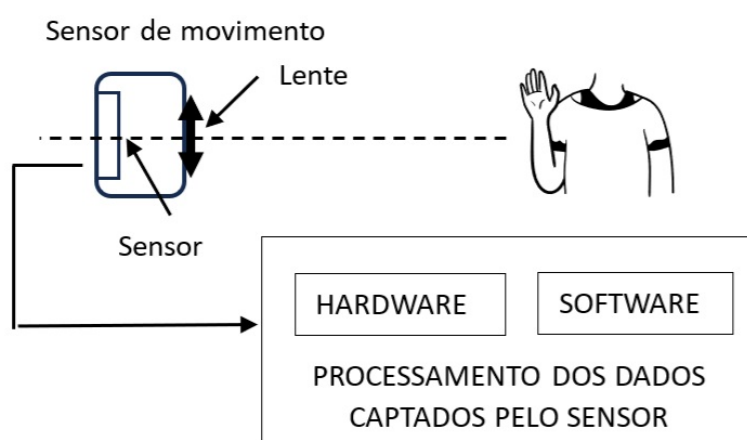
As ferramentas e equipamentos necessários para tratamento de imagens estão sempre relacionados a qualidade do hardware e software específicos para realizar essas tarefas de processamento. Um melhor hardware permite a análise e manipulação de imagens que exigem a utilização de computadores com alta capacidade de processamento e sensores de movimento ou câmeras digitais de alta qualidade. Neste documento no capítulo 2.3, se explicou a importância dos sensores de movimento que já possuem muitas vezes um hardware com melhor qualidade para facilitar essa implementação.

Com relação a software, muitos dos fabricante de sensores de movimento fornecem um software como kit de desenvolvimento (Software Development Kit - SDK) com uma biblioteca

que oferece diversos serviços e recursos que permitem sua utilização em diversos sistemas operacionais.

A partir da utilização dessas ferramentas se construiu uma proposta inicial do sistema que é apresentada na Figura 21. Conforme se pode observar, trata-se de uma estrutura para leitura de dados de imagens, onde se tem o indivíduo posicionado a frente do sensor de movimento, que recebe essa imagem e a converte em dados que possam ser tratados através de um software e hardware específico para atender a etapa de processamento dos dados captados pelo sensor. Este processo de captação e tratamento de dados apresenta como questões a serem consideradas, o custo do hardware e software para captação e tratamento dos dados.

Figura 21 – Captando posicionamento com sensor de movimento



Fonte: Elaborado pelo autor

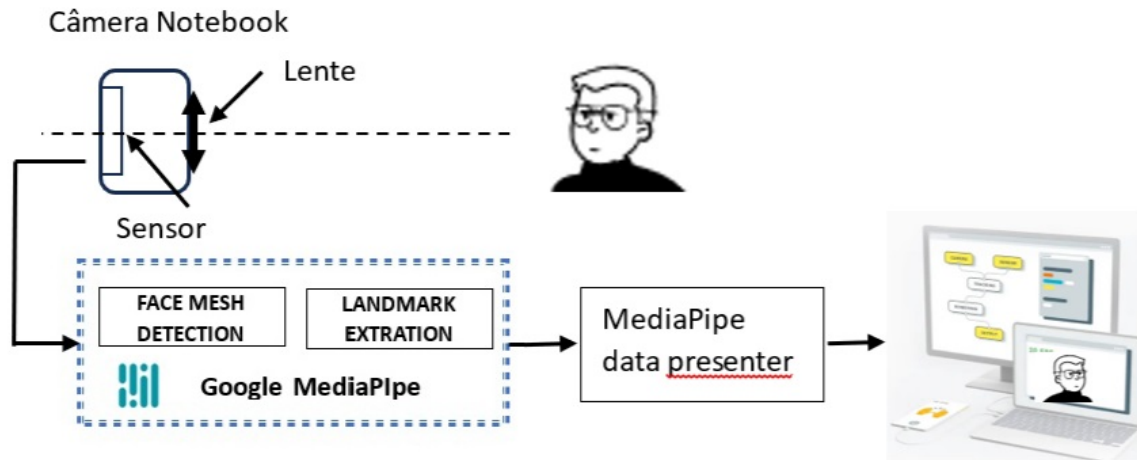
A partir desta arquitetura genérica inicial, se passou a uma mais específica com a utilização do MediaPipe. Esta solução é apresentada na Figura 22 e nela se pode observar a arquitetura de aplicações pré-construída, definida como "MediaPipe data presenter".

Esta arquitetura inclui as soluções prontas mencionadas anteriormente para uso como detecção de rosto, pose humana, entre outras. São aplicações de rastreamento do MediaPipe em realidade aumentada (RA), onde os desenvolvedores podem criar suas próprias calculadoras e pipelines personalizados para atender necessidades específicas.

Com a proposta de identificação de pontos da face, se desenvolveu uma evolução da arquitetura apresentadas na Figura 22 com a utilização do pipeline e da calculadora disponibilizada pelo MediaPipe. Conforme se pode observar na Figura 23, a diferença desta proposta com a anterior é que nesta última se tem componentes de software desenvolvidos pelo autor.

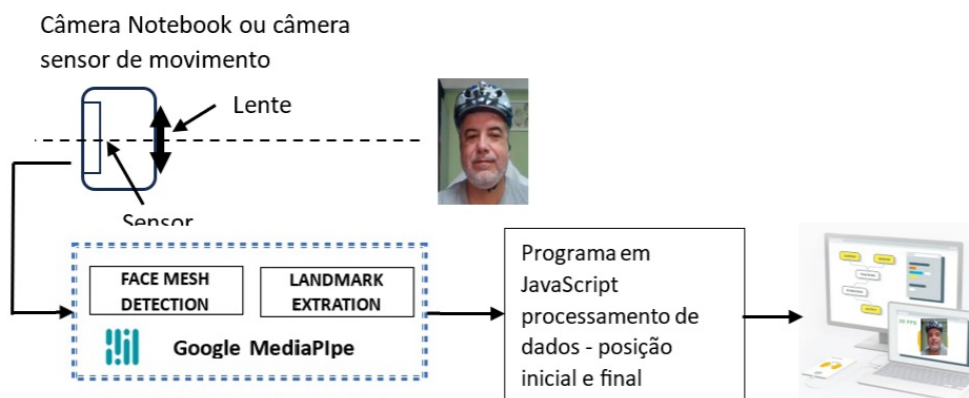
A partir desta proposta, se desenvolveu o sistema utilizando o MediaPipe pose, com a utilização do ponto zero indicando a posição do nariz conforme apresentado na Figura 16. Esta primeira tentativa buscava repetir o procedimento da técnica clínica TRC (c.f. seção 2.1.3). Para tanto, foi feita a montagem de um laser em um capacete de ciclismo para repetir o procedimento adotado por (Pinsault *et al.*, 2006), conforme representado na Figura 24, sendo o feixe apontado a um alvo em uma folha de papel milimetrado.

Figura 22 – Captando posicionamento de face com câmera e MediaPipe



Fonte: Elaborado pelo autor) e (MediaPipe, 2024)

Figura 23 – Captando posicionamento de face com câmera e sensor de movimento utilizando o MediaPipe



Fonte:Elaborado pelo autor) e (MediaPipe, 2024)

Figura 24 – Foto do capacete com laser fixado na ponta



Fonte: Elaborado pelo autor

Na análise de sensibilidade de reposicionamento da cabeça (técnica TRC), utilizou-se um alvo quadrado (40 x 30 cm) posicionado a 90 centímetros de distância da pessoa que utilizava o capacete e uma máscara de dormir para ocluir a visão. Repetiu-se o procedimento com a proposta de se ter uma referência ao posicionamento da face em relação a um alvo, sendo este dispositivo utilizado em conjunto com o sistema desenvolvido (Figura 23).

O processo para verificar o posicionamento da cabeça em relação ao laser fixado no capacete foi facilmente repetido e não houve dificuldade em filmar ou fotografar o procedimento. Esse processo foi repetido em alguns momentos para comparar com os dados do sistema. Como foi possível replicar estes testes, seguiu-se com o desenvolvimento do sistema conforme descrito na próxima seção.

3.4 DESENVOLVIMENTO DO SISTEMA

Esta seção irá apresentar as diferentes versões que foram desenvolvidas a partir da arquitetura inicial Figura 23 para que se pudesse obter uma ferramenta adequada aos objetivos propostos neste trabalho de pesquisa.

Para atender a técnica clínica TRC, desenvolveu-se um fluxograma para elaboração do programa, conforme apresentado na Figura 25.

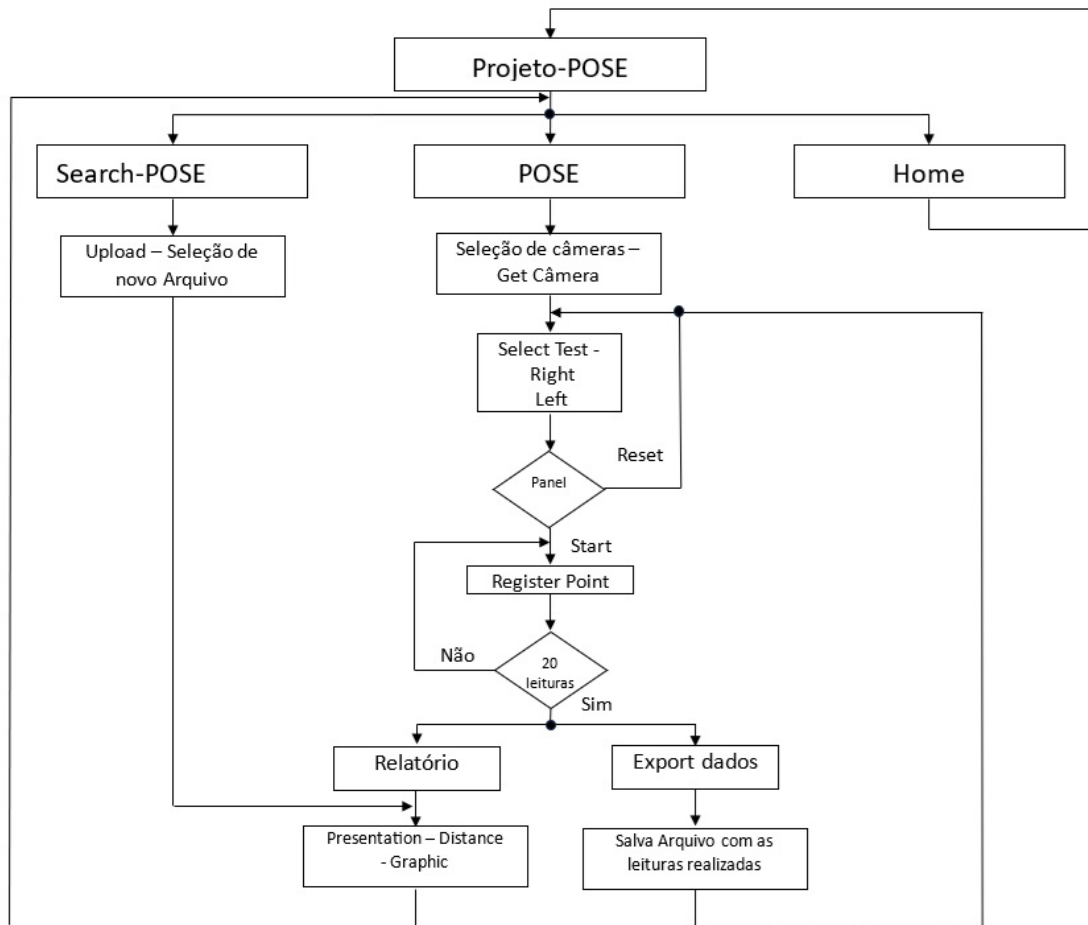
A primeira etapa desenvolvida foi realizar um sistemas para captura de dados das câmeras onde fosse possível selecionar qual câmera seria utilizada, pois havia a necessidade de realizar testes com mais de um modelo de câmera. Em seguida se definiu em qual formato os dados deveriam ser armazenados, com o objetivo de facilitar o seu tratamento.

Iniciou-se o desenvolvimento em linguagem Python, e após vários testes, percebeu-se que havia dificuldade em apresentar uma tela mais intuitiva ao usuário. Dessa forma decidiu-se por alterar para a linguagem JavaScript, que permitia apresentar uma tela com uma interface mais compreensível e uma grande facilidade de interação com o usuário. A próxima etapa foi a de definição de que informações seria apresentadas e solicitadas para a realização do teste. A Figura 26 ilustra o procedimento proposto para o pré-processamento de imagens.

Conforme se pode observar na Figura 26, as imagens são recebidas inicialmente pela câmera (*INPUT FRAMES*). Em seguida, é feita a identificação da face e a extração dos dados de posicionamento utilizando o (*Google MediaPipe*). Na sequência, os dados recebidos pelo MediaPipe são armazenados pelo programa e podem ser exportados para um arquivo no formato *.CSV* (*LANDMARK EXPORT IN CSV*) e então, apresentados graficamente (*GRAFIC DATA*), ficando disponíveis para análise. É possível também ou gerar um relatório com informações sobre os dados obtidos.

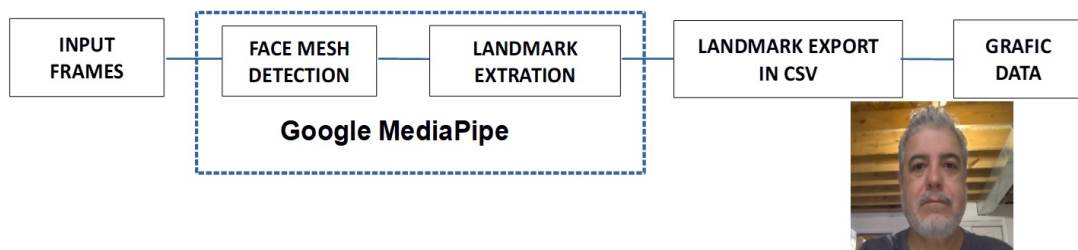
Uma câmera deve ser utilizada para a entrada de dados. Nos experimentos realizados e que serão relatados, utilizou-se inicialmente a câmera de um notebook, com resolução HD de 1280x720 e possibilidade de captação de 30 fps. Foram também utilizadas outras câmeras da Intel: as RealSense, modelos D435i, que possui um sensor RGB e oferece resolução de 1280x720

Figura 25 – Fluxograma do sistema primeira versão proposta Pose



Fonte: Elaborado pelo autor

Figura 26 – Procedimento de pré-processamento de imagens



Fonte: Elaborado pelo autor

com 30 fps, e D455, com sensor RGB e CMOS com resolução de 1280x720 com 90 fps. É importante ressaltar que o MediaPipe trabalha com uma resolução máxima de 640x420 a 30fps.

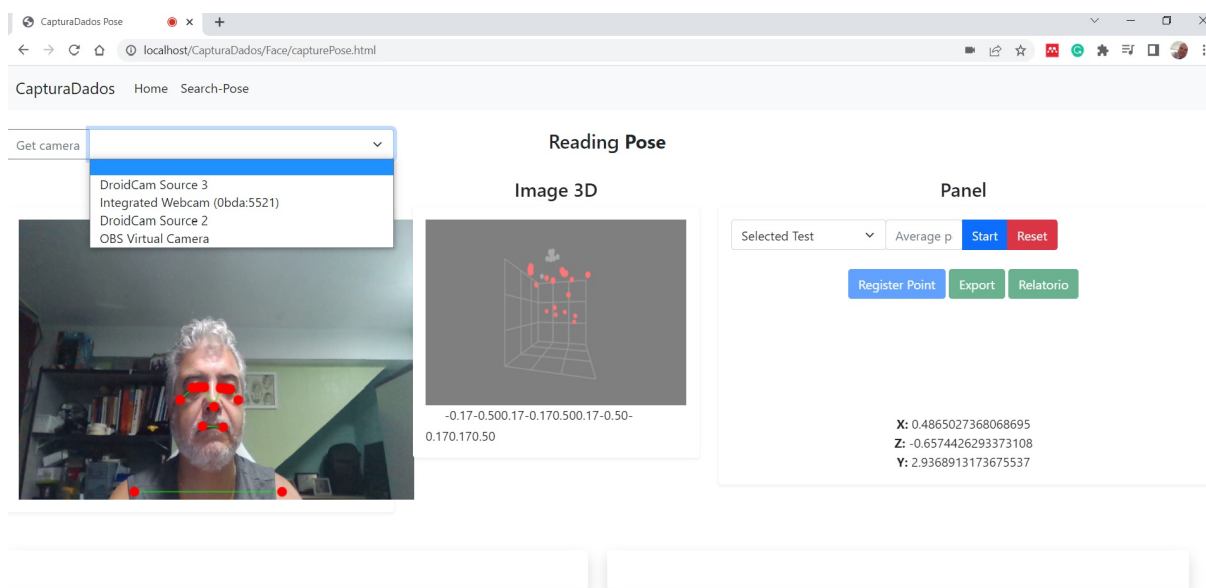
3.4.1 Primeira versão do sistema desenvolvido

A Figura 27 apresenta a primeira versão do sistema que foi desenvolvido, tendo por base o fluxograma proposto. Esta versão denominada (*Projeto POSE*), teve seu desenvolvimento

baseado no fluxograma apresentado na Figura 25 e no procedimento de pré processamento apresentado na Figura 26.

Na Figura 27 pode-se observar logo abaixo do menu, o **ComboBox** para selecionar o tipo de câmera que será utilizada no experimento através do botão (*Get camera*). Uma vez selecionada, a câmera o sistema identifica os pontos captados pelo MediaPipe. Abaixo do **ComboBox** de seleção da câmera, na área (*Image*), os pontos de referência captados e posicionados são apresentados. Ao lado, a área (*Reading Pose*), apresenta a parte (*Image 3D*) que mostra uma representação 3D dos pontos captados pela câmera, sendo essa imagem fornecida pelo MediaPipe. À direita dessa área, encontra-se a área (*Painel*). Nela, é possível selecionar o tipo de movimento que o indivíduo em avaliação deverá realizar a partir do botão (*Select test*), sendo as possibilidades, movimento à direita (*Face Right*) e à esquerda (*Face Left*). Na próxima box, é possível selecionar a quantidade de captação de quadros que pode ser realizada em um segundo (*Average Point*) e esse valor pode variar de 1 a 30 leituras. Após a realização das medidas, os valores apresentados representam uma média das leituras realizadas.

Figura 27 – Primeira versão do sistema desenvolvido



Fonte: Elaborado pelo autor

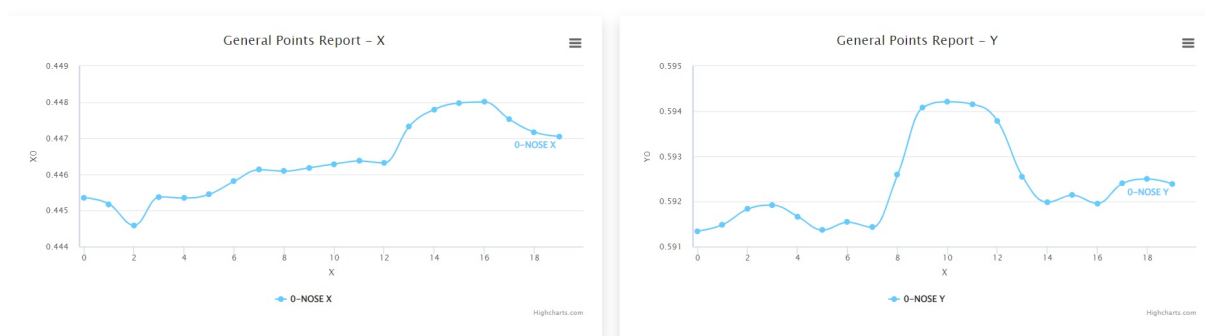
O botão (*Start*) inicia o processo de captura dos dados e o botão (*Reset*) pode ser utilizado em qualquer momento para reiniciar o processo. A partir do botão (*Register Point*), é possível salvar a posição instantânea que está sendo apresentada, sendo que obrigatoriamente são capturadas 20 posições da face, sendo a primeira na posição inicial do movimento da face e a segunda na final após o retorno da face.

Por sua vez, o botão (*Relatório*) produz um relatório com as informações sobre os dados capturados. Uma análise do processo de captura dos dados deve ser realizada pelo especialista, que deve validar o procedimento, definindo que as medições foram realizadas corretamente. Na parte inferior da área (*Panel*), o sistema apresenta a leitura instantânea da posição espacial do

nariz do indivíduo. Em uma outra etapa de desenvolvimento do sistema, foram inseridos sons que indicam a captação nos instantes inicial e final do movimento.

Ainda na Figura 27, é possível observar o botão (*Export*) da área *Panel* que permite gerar uma representação gráfica de todos os pontos salvos. Esta representação é apresentada na Figura 28. Pode-se verificar que são gerados gráficos apenas para as coordenadas x e y do detector. Os pontos x e y representados no eixo das ordenadas dos gráficos são referentes ao (*Ponto 0*) definido pelo MediaPipe. O eixo das abscissas indica o número da leitura.

Figura 28 – Gráfico de linhas de movimento em coordenadas x e y



Fonte: Elaborado pelo autor

Os dados estão disponíveis no programa e podem ser armazenados em uma planilha .csv. Para o armazenamento, é necessário definir o nome do arquivo. Entretanto, alguns parâmetros já estão pré-definidos, como por exemplo, se a medição foi realizada com rotação à esquerda ou à direita. Informações sobre a indivíduo em teste, como nome, idade e sexo são solicitadas, e esses dados, bem como o código do sentido da leitura realizada, irão compor o nome do arquivo, ou seja, os nomes dos arquivos serão da forma `name-age-sex-codxx.csv`.

A Figura 29 apresenta a interface utilizada para a entrada desses dados, e nela pode-se observar os ComboBox Name, Age e Sex através dos quais se pode fornecer os dados. Um exemplo de nome de arquivo pode ser, por exemplo:

`wagner-movimento-esquerda-80cm-53-M-FACERIGHT`

Esta estrutura tem como objetivo facilitar a identificação do paciente, podendo constar, além do movimento, a data em que a captura dos dados foi realizada.

A Figura 30 apresenta uma tela que pode ser utilizada na busca de arquivos previamente salvos. O botão (*Escolher arquivo*) permite o acesso à pasta do computador onde os arquivos .csv foram salvos. Após a indicação do arquivo, por meio do botão (*Upload*) os dados são carregados no sistema.

Figura 29 – Tela para exportar dados

Exportar Dados

Name:

Age

Sex

Export Close

Fonte: Elaborado pelo autor

Figura 30 – Tela de busca dos arquivos

Graphics Generation

Upload Escolher arquivo Nenhum arquivo escolhido

☒ Display in only one graph

Start

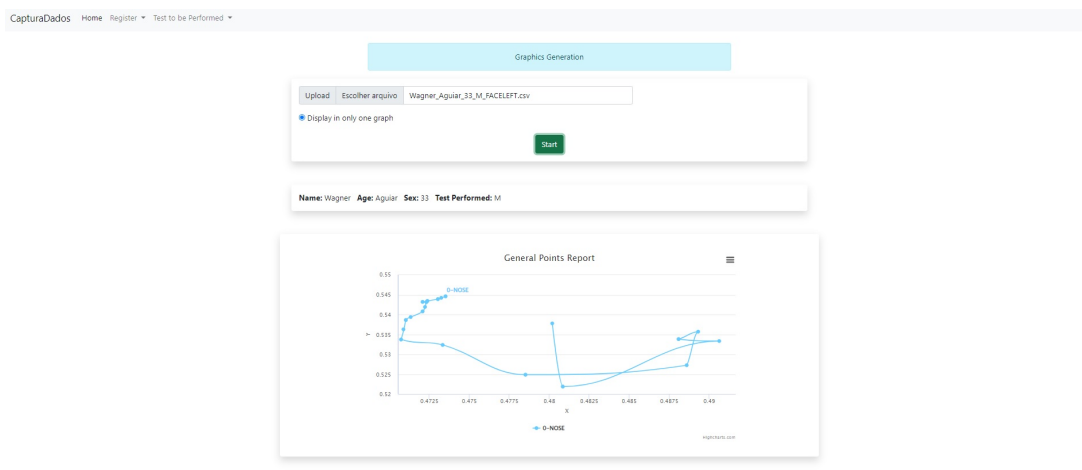
Fonte: Elaborado pelo autor)

Na sequência, a partir do botão (*Start*) é possível gerar automaticamente o gráfico de Pose com informações dos dados armazenados do eixo X e Y, conforme se pode observar na Figura 31. O gráfico gerado faz uma relação entre o eixo x e o eixo y das medidas. Nele, é possível visualizar o movimento do ponto zero após (n) movimentos de rotação (na figura, são utilizados 20 pontos).

O arquivo (.csv) gerado pode então ser utilizado para a realização de cálculos para determinar o valor absoluto do erro de cada medição e para a avaliação da precisão e da consistência do reposicionamento da articulação cervical. A Figura 32 apresenta um gráfico no formato de alvo que indica por cores as diferenças entre movimentações.

Essa opção gráfica é utilizada como uma sugestão de identificar em um gráfico em forma de alvo o ponto em que leitura é apresentada no eixo x e y, a cada ponto do gráfico temos uma forma geométrica e uma cor distinta, produzindo uma apresentação mais intuitiva dos pontos lidos. O procedimento de cálculo utilizado será detalhado no primeiro experimento realizado no

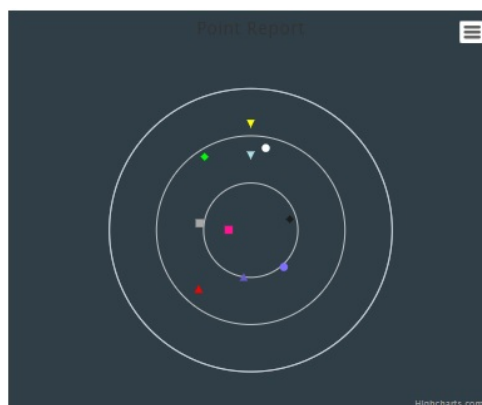
Figura 31 – Gráfico Pose X Y do ponto 0



Fonte: Elaborado pelo autor

Capítulo 4, seção 4.2.1.

Figura 32 – Gráfico Alvo



Fonte: Elaborado pelo autor

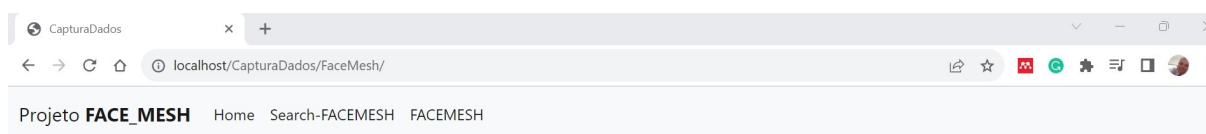
É importante ressaltar que esta foi a primeira versão do sistema, a qual posteriormente sofreu alterações para ser possível atender os objetivos propostos, e permitir uma análise mais detalhada sobre a repetibilidade e reprodutibilidade do software. A versão foi modificada conforme os experimentos estavam sendo realizados e ao termino dos testes todas as modificações foram agregadas ao sistema final.

A distância referenciada nos testes realizados por (Pinsault *et al.*, 2006) no Capítulo 3 é de 3 metros. Porém, ao utilizar-se a câmera junto ao Google MediaPipe no procedimento, houve dificuldade para visualizar e identificar os pontos. Dessa forma, decidiu-se por posicionar a câmera no sistema a 80 cm do indivíduo. Foram realizados experimentos para demonstrar que essa distância atende à proposta do sistema.

3.4.2 Segunda versão do sistema desenvolvido

Na segunda versão do sistema, foi mantida toda interface do desenvolvida em Java Script e optou-se por alterar o método de captação da imagem, passando-se a utilizar o Face Mesh do MediaPipe. A partir dos experimentos apresentados na seção 4.3 se poderá observar que esta técnica apresenta uma resposta mais precisa com relação à repetibilidade do posicionamento da Face e à reprodutibilidade dos dados em diferentes dispositivos. A Figura 33 apresenta parte da interface desenvolvida sendo que ela foi denominada (*Projeto FACEMESH*).

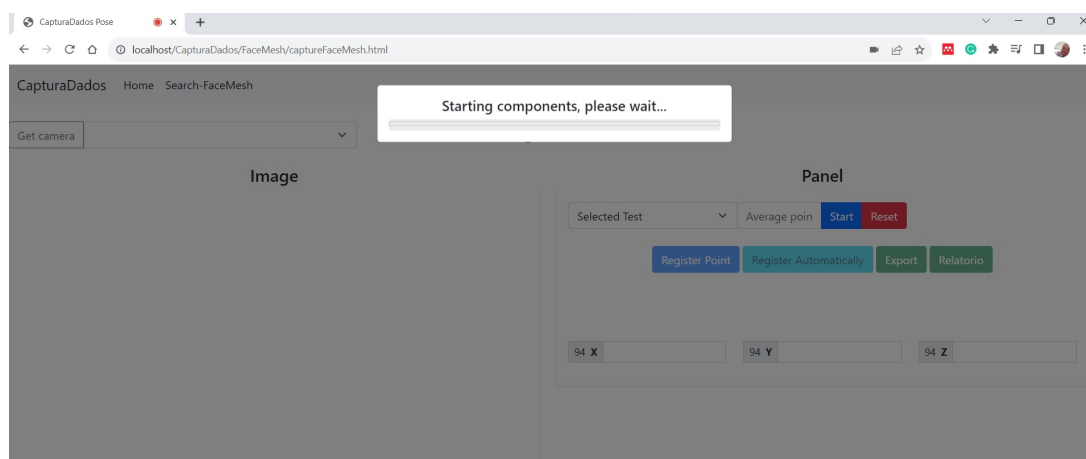
Figura 33 – Tela inicial do sistema versão Face Mesh



Fonte: Elaborado pelo autor

Conforme indicado na Figura 34, quando é acessada a tela de leitura de dados (*FACEMESH*), apresenta-se a informação para aguardar a inicialização dos componentes, como a existência da câmera e o acesso à internet (*Starting components, please wait*). Com os componentes inicializados, as etapas de configurações para as leitura e tratamento dos dados similares às da primeira versão são realizadas.

Figura 34 – Tela de inicialização dos componentes Face Mesh

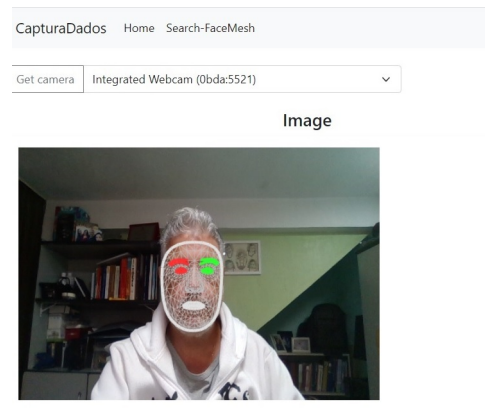


Fonte: Elaborado pelo autor

A tela para captação dos dados é apresentada na Figura 35 e pode-se visualizar que o rosto é identificado com os 468 marcos faciais em tempo real.

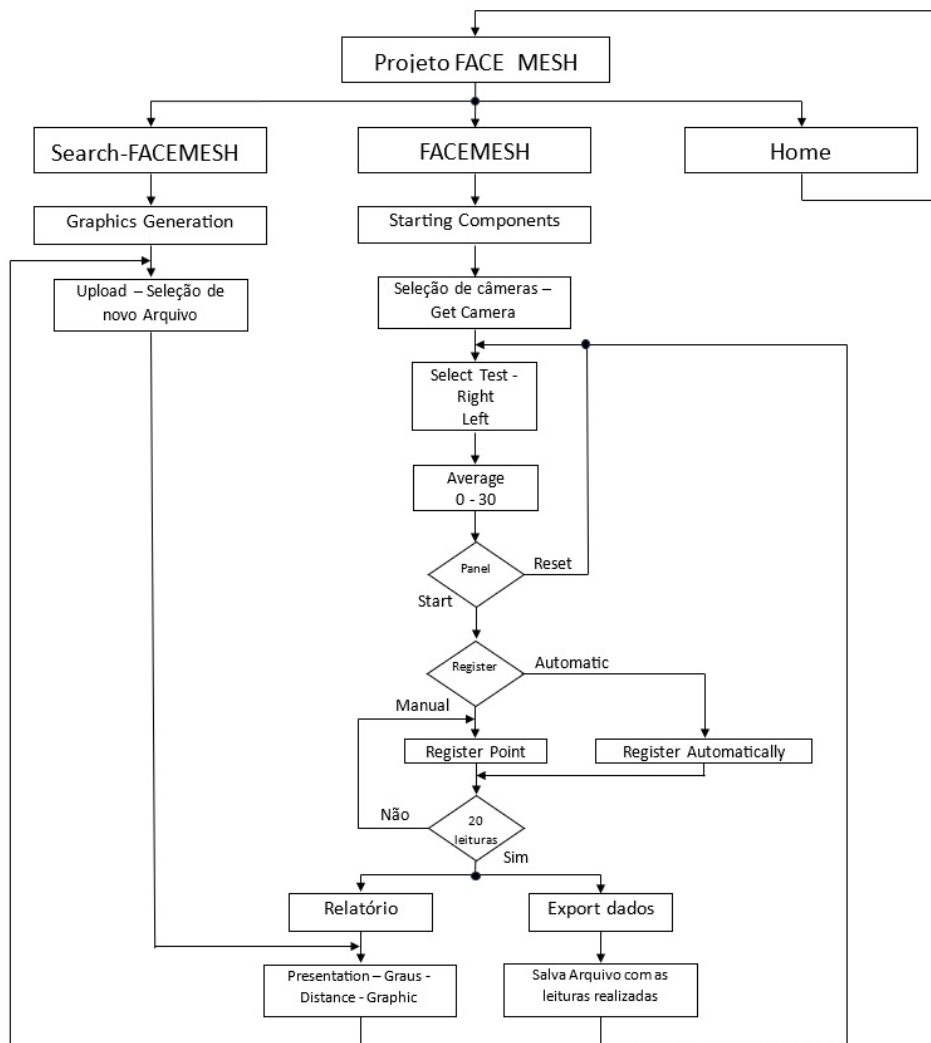
O fluxograma para utilização e programação do sistema nesta nova versão é apresentado na Figura 36. Conforme se pode observar, a estrutura segue as etapas de utilização do software, em que inicialmente são apresentadas as opções de (*Search-FACEMESH - FACEMESH - Home*). Em qualquer etapa do fluxograma, é possível retornar à etapa inicial acionando o (*Home*).

Figura 35 – Tela de captação de pontos com o Face Mesh



Fonte: Elaborado pelo autor

Figura 36 – Fluxograma do sistema versão Face Mesh



Fonte: Elaborado pelo autor

3.5 SÍNTESE

Neste capítulo se detalhou como foi realizado o desenvolvimento do sistema e também se definiu a forma de captação, armazenamento e tratamento dos dados de imagem, bem como a sua apresentação com o objetivo de auxiliar o especialista no teste TRC.

Inicialmente a proposta de desenvolver um sistema utilizando-se sensores de movimento foi apresentada e analisada. Conforme se aprofundou nas análise e tratamento de imagens com a utilização do *TensorFlow*, descobriu-se uma outra forma de captação desses dados, que utiliza o mesmo sistema de aprendizado de máquina. Optou-se pela utilização da estrutura MediaPipe da Google, que emprega a visão computacional para perceber o ambiente e que pode atuar com algoritmos em aprendizado de máquina na captação de dados de imagem. Mostrou-se que a forma como esses dados são captados e tratados internamente no MediaPipe facilita a sua utilização. Da apresentação se pôde perceber que a alternativa se mostra como uma forma de captação dos dados e detecção de imagem estéreo com baixo custo e que pode ser utilizada com várias alternativas de linguagem. Inicialmente se decidiu pela utilização do MediaPipe Pose e posteriormente se alterou para o MediaPipe Face Mesh. O Capítulo também apresentou a forma como o MediaPipe captura os pontos em uma imagem utilizando-se da câmera e em seguida se apresentou algumas aplicações que utilizam o MediaPipe.

O Capítulo também apresentou a arquitetura proposta para o sistema com o capacete que é uma ferramenta utilizada no teste de TRC . A importância da captação dos dados de imagem como uma ferramenta de auxílio aos especialistas na área da saúde, para os quais se faz necessária a avaliação dos grupos de pessoas com a dor cervical traumática também foi destacada.

Para o desenvolvimento do sistema se apresentou o primeiro fluxograma proposto e se mencionou que em sua implementação foi utilizada a linguagem de programação Python e o modelo MediaPipe Pose da Google. Trabalhou-se também com possibilidade de alterar a câmera utilizada. Como apresentado houve a necessidade de se alterar a programação para a linguagem JavaScript para melhorar a interface. Atendendo todas as etapas do fluxograma se desenvolveu a primeira versão do sistema. Com esse sistema se iniciaram os testes para verificar se o aplicativo atendia aos objetivos da proposta.

Em função dos resultados encontrados durante a realização dos testes houve a necessidade de alterar o modelo utilizado para o MediaPipe Face Mesh. As alterações no programa foram realizadas conforme os dados foram sendo obtidos a cada um dos experimentos realizados, sempre com o objetivo de implementar melhorias na forma das leituras dos dados e no seu tratamento. Dessa forma se desenvolveu um fluxograma mais completo com outras formas de tratamento dos dados e apresentações para utilização nos testes. No próximo Capítulo, relata-se a sequência em que os experimentos foram realizados e os objetivos buscados a cada etapa.

4 EXPERIMENTAÇÃO E ANÁLISE DOS RESULTADOS

Este capítulo apresenta inicialmente os experimentos que foram realizados com o objetivo de verificar a repetibilidade e a reprodutibilidade dos dados obtidos, ou seja, entre as leituras de posicionamento de face e dos dispositivos, com esse objetivo definimos a distância de 80cm da câmera para as leituras. É importante salientar que um dos objetivos, também foi verificar se as medidas de posicionamento da face se repetem, mesmo que a posição da câmera sofra uma mudança entre 75 e 85 cm de distância em cada teste. Outro objetivo foi verificar se essas leituras se repetem em modelos de câmeras diferentes.

4.1 AVALIAÇÃO E ANÁLISE DO MOVIMENTO DE FACE

Após a definição da forma como os dados deveriam ser captados pelo sistema, iniciaram-se os testes utilizando um indivíduo de forma estática em frente a câmera e posteriormente foi utilizado um boneco estático (c.f. seção 3.2.1), sendo realizadas medições para verificar a repetibilidade das medidas e a reprodutibilidade, sendo testes realizados com a utilização de outros modelos de câmeras com o objetivo de verificar a sua reprodutibilidade.

Os testes iniciais realizados tiveram como objetivo determinar se, com o desenvolvimento, o sistema apresentava todas as informações sobre a captação e o tratamento dos dados recebidos pelas câmeras. Nos testes realizados com diferentes modelos de câmeras, foi possível observar que alguns parâmetros utilizados no sistema deveriam ser modificados.

Conforme mencionado, o primeiro teste utilizou um indivíduo parado com distância fixa da câmera e sem movimentos com a versão MediaPipe Pose. Realizaram-se varias medições e a análise dos dados obtidos, sendo efetuados os cálculos de média com os valores obtidos. Dessa forma foi possível determinar se a proposta e o procedimento de captação dos dados atendia aos objetivos do trabalho.

Estes testes tiveram como objetivo determinar se, com o desenvolvimento, o sistema apresentava todas as informações necessárias sobre a captação e o tratamento dos dados recebidos pelas câmeras. No capítulo 4, apresentam-se os testes que foram realizados com o sistema e com os diversos modelos de câmeras, tendo sido possível observar que alguns parâmetros utilizados no sistema deveriam ser modificados.

Durante a realização dos primeiros testes com o MediaPipe Pose, foi observado que os objetivos não poderiam ser atendidos e houve a necessidade de alterar o sistema, pois utilizando o MediaPipe Pose não se conseguia uma boa repetibilidade. Essas mudanças e alterações do sistema e dos tipos de testes são explicados após a apresentação do experimento.

Os mesmos testes foram repetidos com o MediaPipe FaceMesh e os valores obtidos foram mais precisos e apresentavam melhores respostas aos objetivos buscados no trabalho.

Nos experimentos foram utilizados os dois modos distintos de captação de dados do MediaPipe introduzidos na seção 3.2. O sistema utilizando o MediaPipe Pose, é detalhado na seção 4.2, enquanto a versão usando o MediaPipe Face Mesh, é detalhada na seção 4.3. A seção 4.4 apresenta os testes realizados de forma dinâmica, com o sistema indicando os valores obtidos mais próximos de uma condição real. Ao término de cada experimento, realiza-se uma análise dos dados obtidos para que fosse possível direcionar os próximos experimentos.

4.2 TESTE ESTÁTICO - MEDIAPIPE POSE

Os primeiros testes apresentados nesta seção tiveram como objetivo verificar o comportamento do sistema em aplicações onde não houvesse movimentação da face, ou seja, o rosto não se movimenta e a câmera é colocada em diferentes posições, mantendo a distância, com objetivo de verificar se os valores obtidos são próximos. Nesta configuração se denominou os testes realizados como *Teste Estático*.

Verificou-se se, mesmo com a câmera em posições diferentes, a indicação de movimento da face obteve valor próximo de zero de distância,. Nestas etapas, repetiu-se os procedimentos alterando os modelos de câmeras. As câmeras utilizadas nesta etapa estática possuem as seguintes características básicas:

- Câmera do notebook - Dimensions, 1280x720 - Aspect Ratio 16:9 e Frame Rate 30 fps
- Câmera Intel RealSense D435i - 1280x720 - Frame Rate 30 fps
- Câmera Intel RealSense D455 - 1280x720 - Frame Rate 30 fps

Optou-se por essas câmeras pela facilidade de acesso e dispositivos recentes no mercado, o que permitiu verificar diferenças nas informações sobre os dados dos quadros que foram armazenados em mesmo ambiente de leitura. O primeiro teste realizado foi o de repetibilidade das medições. Nos três primeiros experimentos, descritos nas seções 4.2.1, 4.2.3 e 4.2.4, utilizou-se o MediaPipe Pose e isso permitiu verificar que tal método não atendia ao que era necessário, como repetibilidade das medições. A partir da seção 4.3, se apresenta o procedimento alterado para a utilização do MediaPipe Face Mesh.

Foi possível verificar que os dados obtidos em alguns modelos de câmeras apresentavam respostas com valores incoerentes e muito diferentes. Após uma análise dos dados, verificou-se que isso acontece principalmente com as câmeras da RealSense, pois esses equipamentos possuem especificações diferentes das câmeras de notebooks. Na seção 4.3, apresentam-se as medições realizadas com esses modelos de câmeras e realiza-se uma análise dos dados obtidos. A análise indicou que não seria viável nesta aplicação, utilizar estes modelos de sensores de movimento. Outro teste realizado foi com câmeras de celulares, em que se verificou a influência das lentes utilizadas nos valores obtidos. Esses testes também levaram ao abandono da utilização

desse tipo de sensor. Futuramente, se pretende realizar outros estudo sobre a possibilidade do desenvolvimento de um aplicativo para as câmeras mencionadas e para celular.

De acordo com o avanço dos experimentos, verificou-se que as câmeras utilizadas nos notebooks não sofreram grandes alterações nos últimos anos. Mesmo com a evolução dos equipamentos, esses dispositivos apresentam características similares. Dessa forma, definiu-se a câmera do notebook como a principal ferramenta para os testes. Esta utilização é detalhada na próxima seção.

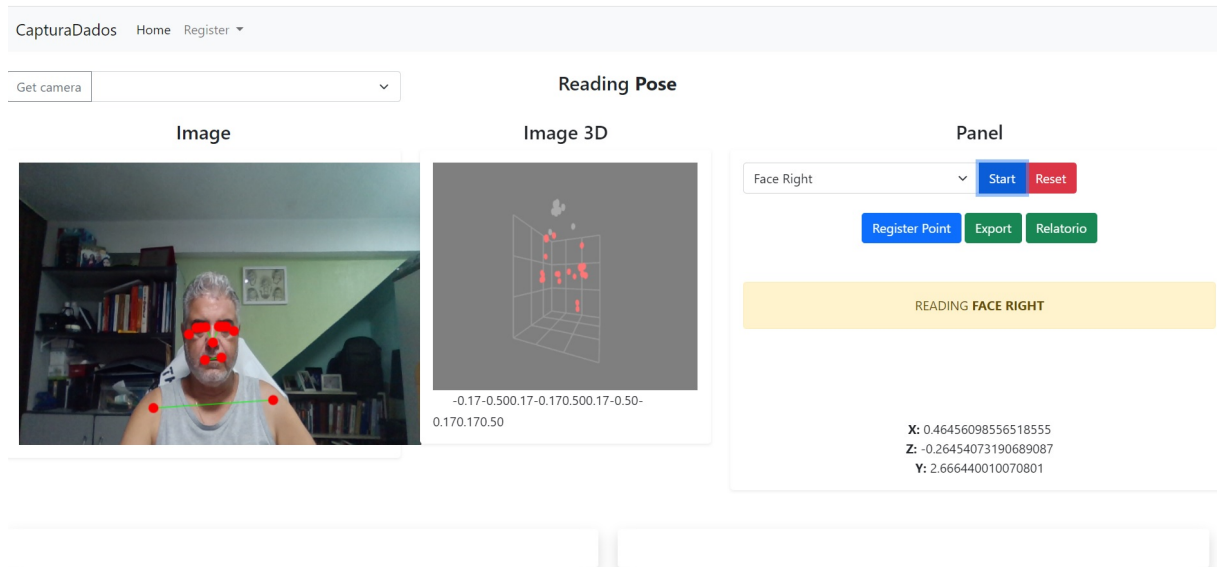
4.2.1 Medições com câmera do Notebook

Para este teste, utilizou-se o MediaPipe Pose. Foi feita a captação unicamente do ponto "0" do nariz da face do indivíduo. Para a realização do teste utilizou-se um notebook cuja câmera captava imagens com as seguintes características: Dimensions, 1280x720, Aspect Ratio 16:9 e Frame Rate 30 fps. O indivíduo foi posicionado a 80 cm da câmera do notebook com o melhor alinhamento possível entre o nariz e a câmera. Foi solicitado ao indivíduo que não movimentasse a cabeça, e então realizou-se a captação por 20 vezes de um ponto do nariz, com intervalo de tempo de aproximadamente 1 segundo.

A Figura 37 apresenta uma imagem da tela do computador com o sistema, em que se apresenta uma imagem da face . Nesta figura, pode-se observar à esquerda na imagem obtida pela câmera os pontos de referência da parte do corpo captada indicadas em cor vermelha. À direita desta imagem, é possível observar uma representação 3D do corpo humano construída pelo aplicativo a partir da imagem captada pela câmera. Nela, também se podem observar os pontos de referência do corpo humano. Ainda à direita desta imagem 3D, é possível observar uma área usada no aplicativo para interação, onde se indica o tipo de teste.

Neste primeiro teste, foi selecionado o movimento à direita "Somente para o funcionamento do sistema" e aciona-se o botão de `start` para começar a captar os pontos. Para que cada ponto seja armazenado, aciona-se o botão de `Register Point` e captam-se os pontos X, Y e Z do quadro. Ao final de 20 leituras, é possível gerar um `Relatório` que apresenta em forma de gráfico de linha a relação entre os pontos X e Y, conforme se pode observar no gráfico já apresentado na Figura 28.

Figura 37 – Imagem medição ponto nariz com sistema



Fonte: Elaborado pelo autor

A Tabela 3 apresenta os dados relativos à primeira medição realizada. Para cada ponto armazenado, há uma leitura de quadro e o registro de 20 leituras. Na Tabela 4, repete-se o procedimento de medições com as mesmas condições da primeira leitura. Realizou-se um pequeno reposicionamento da face no eixo X e Y posicionado frente à câmera, sendo mantida a distância.

Tabela 3 – Primeira leitura sem movimento da face

Leitura Pto 1 - M1	
X	Y
0,517091	0,569278
0,516946	0,566577
05173090	0,566337
0,516713	0,572946
0,516705	0,573207
0,516650	0,572882
0,516024	0,573159
0,519333	0,574570
0,519398	0,574579
0,520124	0,574559
0,521063	0,573802
0,521035	0,573470
0,520664	0,573168
0,520948	0,572233
0,520192	0,573588
0,518970	0,573053
0,519410	0,572716
0,519947	0,571696
0,519175	0,571626
0,518769	0,571248

Fonte: Elaborado pelo autor

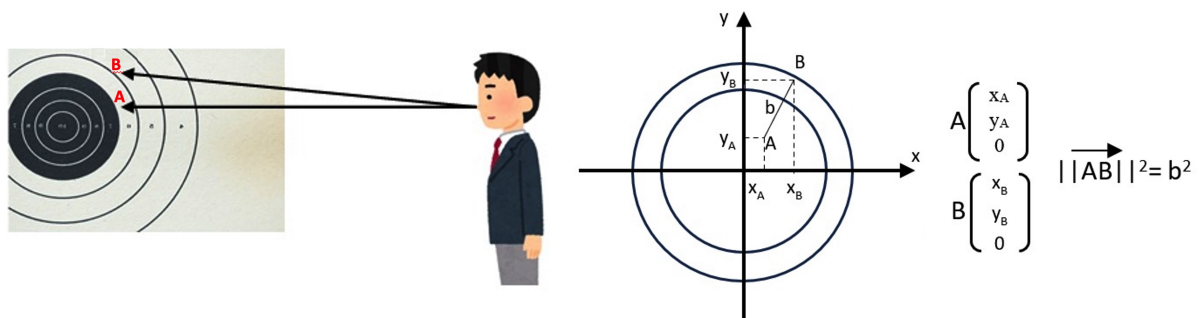
Tabela 4 – Segunda Leitura sem movimento de face

Leitura Pto 2 - M1	
X	Y
0,517845	0,520386
0,518850	0,520513
0,518687	0,521021
0,518315	0,520962
0,517836	0,522610
0,517875	0,523121
0,516741	0,522703
0,517109	0,521769
0,516953	0,522855
0,516973	0,524566
0,514624	0,522688
0,515874	0,521763
0,514658	0,522007
0,515156	0,523374
0,516164	0,522562
0,516739	0,522856
0,516015	0,523126
0,516042	0,522191
0,517179	0,520677
0,516754	0,520313

Fonte: Elaborado pelo autor

Observou-se que, mesmo mantidas as condições nas medições, os valores obtidos foram diferentes. Isso indicou que os pontos foram captados em posições diferentes. Para calcular cada uma das distâncias entre as medições realizadas no primeiro e o segundo ponto captado, X e Y leitura 1 e X e Y leitura 2 se utilizou o Teorema de Pitágoras. A Figura 38 apresenta de forma esquemática o posicionamento e os dados usados nos cálculos.

Figura 38 – Procedimento para realização dos cálculos



Fonte: Elaborado pelo autor

A equação 1 apresentada a seguir detalha, a partir da representação a direita da Figura 38 , como os cálculos foram feitos.

$$h^2 = (x_B - x_A)^2 + (y_B - y_A)^2 \quad (1)$$

onde b = distância entre os pontos (cm)

x_A = ponto inicial

x_B = ponto final

y_A = ponto inicial

y_B = ponto final

A partir dos dados apresentados nas Tabelas 3 e 4, calculou-se a distância entre os valores obtidos nas leituras, e os pontos utilizados no cálculo foram o da primeira e segunda linhas sucessivamente até que todas as leituras tenham sido realizadas. Os valores calculados são apresentados na tabela onde as medidas do ponto 1 apresentam os valores obtidos de distância e em seguida multiplicado por 1000 para indicar as distâncias de movimentação em centímetros. Os valores médios e de desvio padrão das leituras realizadas são apresentados em cálculos nas Tabelas 5 e 6.

Tabela 5 – Comparação entre as medições de mesmo ponto calculado leitura 1

Leituras	Medidas Pto 1 - M1		Medição pto 1	Mult.1000
	X	Y		
1	0,517091	0,569278	0,002705	2,704653
	0,516946	0,566577		
2	0,517309	0,566337	0,006637	6,636580
	0,516713	0,572946		
3	0,516705	0,573207	0,000330	0,330362
	0,516650	0,572882		
4	0,516024	0,573159	0,003597	3,596866
	0,519333	0,574570		
5	0,519398	0,574579	0,000727	0,726686
	0,520124	0,574559		
6	0,521063	0,573802	0,000333	0,333312
	0,521035	0,573470		
7	0,520664	0,573168	0,000978	0,977537
	0,520948	0,572233		
8	0,520192	0,573588	0,001334	1,333703
	0,518970	0,573053		
9	0,519410	0,572716	0,001153	1,152814
	0,519947	0,571696		
10	0,519175	0,571626	0,000554	0,554370
	0,518769	0,571248		
Valor da média			1,065176	
Desvio padrão			1,890897	

Fonte: Elaborado pelo autor

Tabela 6 – Comparação entre as medições de mesmo ponto calculado leitura 2

Leituras	Medidas Pto 1 - M2		Medição pto 1	Mult.1000
	X	Y		
1	0,517845	0,520386	0,001013	1,012842
	0,518850	0,520513		
2	0,518687	0,521021	0,000377	0,377133
	0,518315	0,520962		
3	0,517836	0,522610	0,000512	0,511931
	0,517875	0,523121		
4	0,516741	0,522703	0,001005	1,004739
	0,517109	0,521769		
5	0,516953	0,522855	0,001711	1,710654
	0,516973	0,524566		
6	0,514624	0,522688	0,001555	1,554927
	0,515874	0,521763		
7	0,514658	0,522007	0,001455	1,455185
	0,515156	0,523374		
8	0,516164	0,522562	0,000645	0,645180
	0,516739	0,522856		
9	0,516015	0,523126	0,000936	0,935502
	0,516042	0,522191		
10	0,517179	0,520677	0,000559	0,559167
	0,516754	0,520313		
Valor da média			0,970120	
Desvio padrão			0,443340	

Fonte: Elaborado pelo autor

Pode-se observar que os valores não se repetem e não apresentam uma boa repetibilidade. Na tabela 7 ANOVA apresentamos os valores da média e variância. Como a proposta é obter dados de média inferiores a 0,5 cm e como o valor da média é muito superior ao do objetivo, descartamos esse método para leitura.

Tabela 7 – Tabela com cálculos de média e variância

Anova: fator único				
RESUMO				
Grupo	Contagem	Soma	Média	Variância
Leitura Pto. 1 M1	10	9,76726	0,976726	0,218389
Leitura Pto. 1 M2	10	18,346882	1,834688	3,972769

Fonte: Elaborado pelo autor

A Figura 39 apresenta de forma gráfica a distância entre as leituras feitas em dois momentos distintos, entre as Tabelas 5 e 6. A partir desse gráfico e em todos os outros do mesmo tipo serão apresentados, se irá utilizar linhas na cor azul para indicar o valor desejado e linhas na cor laranja para indicar o resultado obtido.

Dessa forma considerando os dados obtidos, optou-se por realizar um novo teste e para esse novo teste utilizamos o boneco estático com o objetivo de verificar o comportamento do sistema, utilizando uma face imóvel.

Figura 39 – Comparação gráfica das medições de mesmo ponto calculado

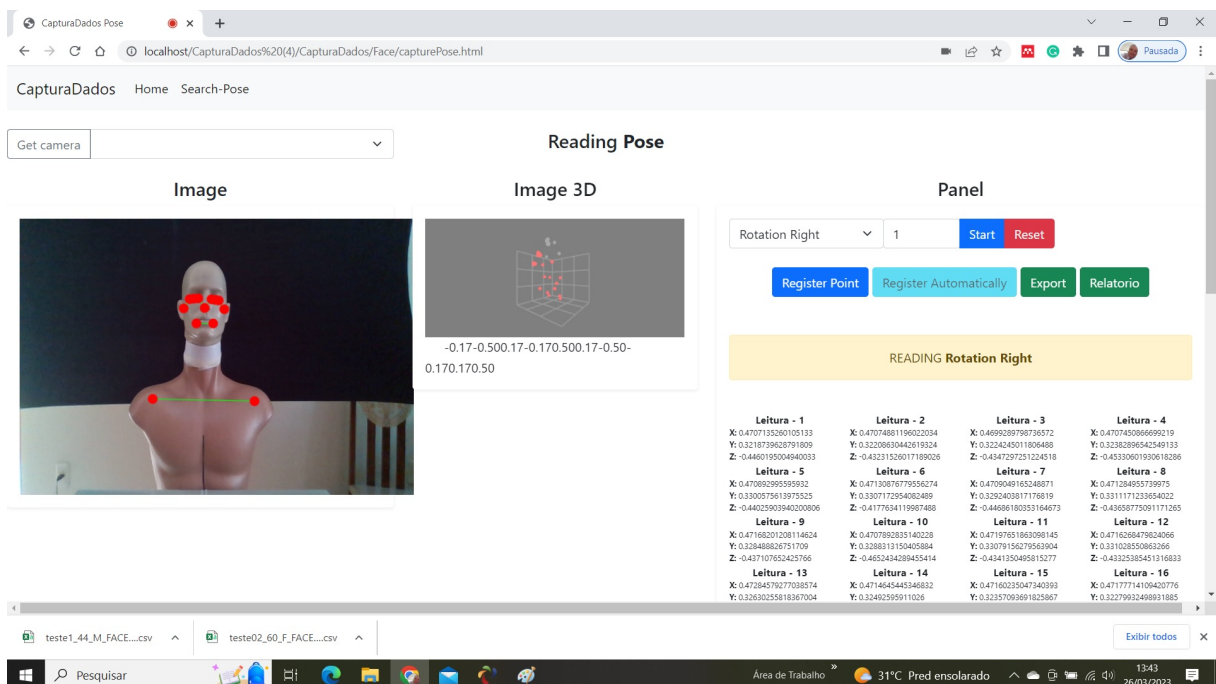


Fonte: Elaborado pelo autor

4.2.2 Medições com câmera do Notebook - Boneco

Para assegurar que as discrepâncias encontradas não fossem devidas a movimentações involuntárias da face do indivíduo em análise, se decidiu pela realização de um novo teste substituindo-se a pessoa por um boneco, assegurando que não ocorreria nenhum movimento. Neste teste, repetiu-se o procedimento utilizado no experimento anterior com o boneco também posicionado em frente à câmera do notebook a uma distância de 80cm e o procedimento foi repetido com as mesmas condições de ambiente. O boneco foi fixado em um tripé para controle da altura e da distância até a câmera do notebook conforme se pode observar na imagem apresentada a esquerda da Figura 40.

Figura 40 – Imagem do boneco posicionado em frente ao notebook e captado pelo sistema



Fonte: Elaborado pelo autor

Da mesma forma que o experimento anterior, realizamos duas leituras na mesma posição e

repetiram-se as 20 leituras, sendo os valores obtidos apresentados nas Tabelas 8 e 9.

Tabela 8 – Leitura 1 realizada com o boneco estático e posicionado a 80 cm do notebook

Leitura Pto 1 - M1 - Boneco	
X	Y
0,470714	0,321874
0,470749	0,322086
0,469929	0,322425
0,470745	0,323829
0,470893	0,330058
0,471309	0,330717
0,470905	0,329240
0,471285	0,331117
0,471682	0,328489
0,470789	0,328831
0,471977	0,330792
0,471627	0,331029
0,472846	0,326303
0,471465	0,324926
0,471602	0,323571
0,471777	0,322799
0,471897	0,321382
0,471135	0,321602
0,471938	0,321329
0,471027	0,322017

Fonte: Elaborado pelo autor)

Tabela 9 – Leitura 2 realizada com o boneco estático e posicionado a 80 cm do notebook

Leitura Pto 1 - M2 - Boneco	
X	Y
0,474402	0,327752
0,472247	0,327345
0,471802	0,329078
0,472100	0,328443
0,470328	0,326889
0,471550	0,327944
0,472931	0,328493
0,471311	0,328732
0,470810	0,326772
0,473491	0,325004
0,474541	0,325081
0,473400	0,324538
0,473272	0,325073
0,470695	0,325538
0,470324	0,328284
0,470027	0,327615
0,470368	0,326745
0,469785	0,327843
0,469860	0,326663
0,470215	0,328999

Fonte: Elaborado pelo autor

Conforme se pode observar a partir dos dados apresentados nas Tabelas 10 e 11, as medições

foram realizadas em posições diferentes e com esses dados foi realizada uma análise das diferenças entre os valores obtidos a cada medição.

Tabela 10 – Cálculos de distância, média e desvio padrão Pto. 1 - M1

Leituras	Medidas Pto 1 - M1		Medição pto 1	Mult.1000
	X	Y		
1	0,470714	0,321874	0,000215	0,215253
	0,470749	0,322086		
2	0,469929	0,322425	0,001624	1,624361
	0,470745	0,323829		
3	0,470893	0,330058	0,000780	0,779818
	0,471309	0,330717		
4	0,470905	0,329240	0,001915	1,914834
	0,471285	0,331117		
5	0,471682	0,328489	0,000956	0,956171
	0,470789	0,328831		
6	0,471977	0,330792	0,000422	0,422413
	0,471627	0,331029		
7	0,472846	0,326303	0,001950	1,950095
	0,471465	0,324926		
8	0,471602	0,323571	0,000791	0,791162
	0,471777	0,322799		
9	0,471897	0,321382	0,000793	0,792715
	0,471135	0,321602		
10	0,471938	0,321329	0,001142	1,141733
	0,471027	0,322017		
Valor da média			0,874443	
Desvio padrão			0,565800	

Fonte: Elaborado pelo autor

Verificou-se novamente que os mesmo não atendiam ao objetivo buscado pois os valores médios e de desvio padrão obtidos continuam diferentes, mesmo não havendo movimento da face do boneco. Pode-se observar que os valores não se repetem e não apresentam uma boa repetibilidade. Na tabela 12 ANOVA apresentamos os valores da média e variância.

Tabela 11 – Cálculos de distância, média e desvio padrão Pto. 1 - M2

Leitura	Medidas Pto 1 - M2		Medição pto 1	Mult.1000
	X	Y		
1	0,474402	0,327752	0,002193	2,193426
	0,472247	0,327345		
2	0,471802	0,329078	0,000701	0,701093
	0,472100	0,328443		
3	0,470328	0,326889	0,001615	1,614902
	0,471550	0,327944		
4	0,472931	0,328493	0,001638	1,637656
	0,471311	0,328732		
5	0,470810	0,326772	0,003212	3,211885
	0,473491	0,325004		
6	0,474541	0,325081	0,001264	1,263549
	0,473400	0,324538		
7	0,473272	0,325073	0,002618	2,618320
	0,470695	0,325538		
8	0,470324	0,328284	0,000732	0,731646
	0,470027	0,327615		
9	0,470368	0,326745	0,001243	1,243346
	0,469785	0,327843		
10	0,469860	0,326663	0,002363	2,363175
	0,470215	0,328999		
Valor da média			1,626279	
Desvio padrão			0,782594	

Fonte: Elaborado pelo autor

Tabela 12 – Tabela com cálculos de média e variância

Anova: fator único				
RESUMO				
Grupo	Contagem	Soma	Média	Variância
Leitura Pto. 1 M1	10	10,588555	1,058855	0,355699
Leitura Pto. 1 M2	10	17,578997	1,757899	0,680503

Fonte: Elaborado pelo autor

Como a proposta é obter dados de média inferiores a 0,5 cm não foi atendida, descartamos mais esse método para leitura. Esta situação é reforçada pela imagem apresentada na Figura 41.

Figura 41 – Comparação gráfica das medições de mesmo ponto calculado



Fonte: Elaborado pelo autor

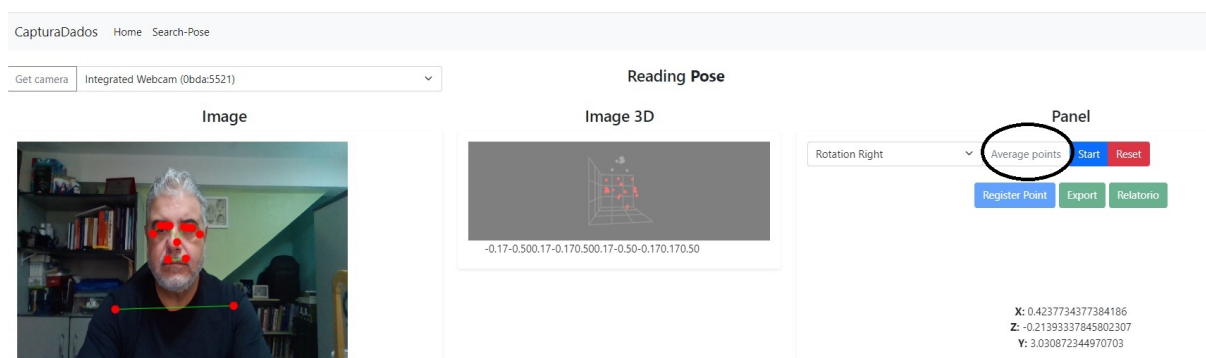
Em função deste resultado, decidiu-se pela realização de um novo teste, no qual se optou por calcular uma média das leituras de cada ponto captado. Estas mudanças no sistema para a realização do experimento são apresentadas e explicada na próxima seção.

4.2.3 Medições com a câmera do Notebook - Valores médios

Neste teste, realizou-se uma modificação no sistema. Como as captações de quadros podem ser realizadas em até 30 leituras por segundo (valor máximo de leituras que podem ser realizadas do ponto neste intervalo de tempo), optou-se por realizar novamente, mais de uma leitura do mesmo ponto e utilizar a média dessas leituras como sendo o valor do ponto captado.

O sistema foi alterado de forma a ser possível determinar a quantidade de vezes que o ponto deveria ser lido. Para que fosse possível alterar a quantidade de leituras realizadas, optou-se por inserir uma nova "caixa" denominada "Average points". A partir dela é possível inserir o número de leituras a serem realizadas (mínimo 1 e máximo 30), sendo o valor da média futuramente utilizado para cálculos. Esta nova tela do sistema é apresentada na Figura 42.

Figura 42 – Inserção da média entre as medidas do ponto



Fonte: Elaborado pelo autor

Como realizado no experimento anterior, novamente o processo de captação do ponto da face foi repetido. O sistema efetua a leitura de um único ponto do quadro selecionado, mas com o valor de "Average points" tendo sido definido em 20 leituras de quadros por segundo.

Dessa forma, optou-se por ler mais de um quadro e armazenar a média desses pontos. Assim, buscava-se apresentar um valor mais preciso dos pontos lidos, a partir da apresentação somente do valor médio. O restante do sistema não sofreu alterações e o procedimento utilizado nos testes anteriores foi repetido.

Novamente 20 pontos foram lidos. As Tabelas 13 e 14 indicam esses valores lidos, enquanto as Tabelas 15 e 16 indicam as distâncias calculadas entre os pontos X e Y das leituras.

Conforme se pode observar pelos dados apresentados nas Tabelas 15 e 16, os valores continuam diferentes. Verificou-se novamente que os mesmos não atendiam ao objetivo buscado pois os valores médios e de desvio padrão obtidos continuam diferentes do objetivo, mesmo não havendo movimento da face. Pode-se observar que os valores não apresentam uma boa repetibilidade. Na tabela 17 ANOVA apresentamos os valores da média e variância.

Tabela 13 – Primeira leitura sem movimento da face - com média de leituras

Leitura Pto 1 - M1	
X	Y
0,489704	0,468772
0,484662	0,472996
0,484673	0,473949
0,484806	0,473589
0,488232	0,474588
0,487679	0,477787
0,491226	0,472692
0,491879	0,471605
0,492230	0,471628
0,491369	0,472171
0,488316	0,476832
0,488116	0,478127
0,488694	0,476455
0,488261	0,475701
0,484206	0,475826
0,484592	0,476424
0,483332	0,474048
0,481648	0,474737
0,482163	0,472987
0,481270	0,480727

Fonte: Elaborado pelo autor

Tabela 14 – Segunda leitura sem movimento da face - com média de leituras

Leitura Pto 1 - M2	
X	Y
0,473530	0,504362
0,472467	0,505984
0,472763	0,506387
0,473131	0,507854
0,470161	0,504023
0,469915	0,504863
0,470708	0,507204
0,470640	0,503911
0,470292	0,503012
0,470913	0,506920
0,471241	0,504738
0,471312	0,506640
0,474333	0,508335
0,473901	0,509298
0,473111	0,510578
0,472562	0,510639
0,473627	0,511432
0,473336	0,513998
0,473742	0,513303
0,473832	0,513754

Fonte: Elaborado pelo autor

Como a proposta é obter dados de média inferiores a 0,5 cm não foi atendida, descartamos mais esse método para leitura. Esta situação é reforçada pela imagem apresentada na Figura 43.

Tabela 15 – Cálculos de distância, média e desvio da leitura Pto. 1 - M1

Leituras	Medidas Pto 1 - M1		Medição pto 1	Mult.1000
	X	Y		
1	0,489704	0,468772	0,006578	6,577931
	0,484662	0,472996		
2	0,484673	0,473949	0,000385	0,384550
	0,484806	0,473589		
3	0,488232	0,474588	0,003246	3,246171
	0,487679	0,477787		
4	0,491226	0,472692	0,001268	1,267531
	0,491879	0,471605		
5	0,492230	0,471628	0,001018	1,017769
	0,491369	0,472171		
6	0,488316	0,476832	0,001311	1,310750
	0,488116	0,478127		
7	0,488694	0,476455	0,000870	0,869581
	0,488261	0,475701		
8	0,484206	0,475826	0,000712	0,711778
	0,484592	0,476424		
9	0,483332	0,474048	0,001820	1,820108
	0,481648	0,474737		
10	0,482163	0,472987	0,007792	7,791969
	0,481270	0,480727		
Valor da média			2,499814	
Desvio padrão			2,472143	

Fonte: Elaborado pelo autor

Tabela 16 – Cálculos de distância, média e desvio da leitura Pto. 1 - M2

Leituras	Medidas Pto 1 - M2		Medição pto 1	Mult.1000
	X	Y		
1	0,473530	0,504362	0,001940	1,939507
	0,472467	0,505984		
2	0,472763	0,506387	0,001512	1,512134
	0,473131	0,507854		
3	0,470161	0,504023	0,000876	0,875949
	0,469915	0,504863		
4	0,470708	0,507204	0,003293	3,292781
	0,470640	0,503911		
5	0,470292	0,503012	0,003957	3,956718
	0,470913	0,506920		
6	0,471241	0,504738	0,001903	1,902751
	0,471312	0,506640		
7	0,474333	0,508335	0,001055	1,054842
	0,473901	0,509298		
8	0,473111	0,510578	0,000552	0,552325
	0,472562	0,510639		
9	0,473627	0,511432	0,002582	2,582303
	0,473336	0,513998		
10	0,473742	0,513303	0,000460	0,459702
	0,473832	0,513754		
Valor da média			1,812901	
Desvio padrão			1,111992	

Fonte: Elaborado pelo autor

Tabela 17 – Tabela com cálculos de média e variância

Anova: fator único				
RESUMO				
Grupo	Contagem	Soma	Média	Variância
Leitura Pto.1 M1	10	24,998138	2,499813	6,790546
Leitura Pto.1 M2	10	18,129010	1,812901	1,373919

Fonte: Elaborado pelo autor

Figura 43 – Comparação gráfica de medições do mesmo ponto calculado - valores médios



Fonte: Elaborado pelo autor

Este experimento demonstrou que, mesmo utilizando a média das leituras realizadas, tal procedimento não atende aos objetivos da pesquisa, os valores da media e do desvio padrão nos dois momentos de leitura apresentaram valores muito diferentes dos desejados e novamente foram realizadas alteração no sistema. Na próxima mudança, optamos por realizar a leitura de três pontos, sendo que a próxima seção apresenta e explica esta nova alteração.

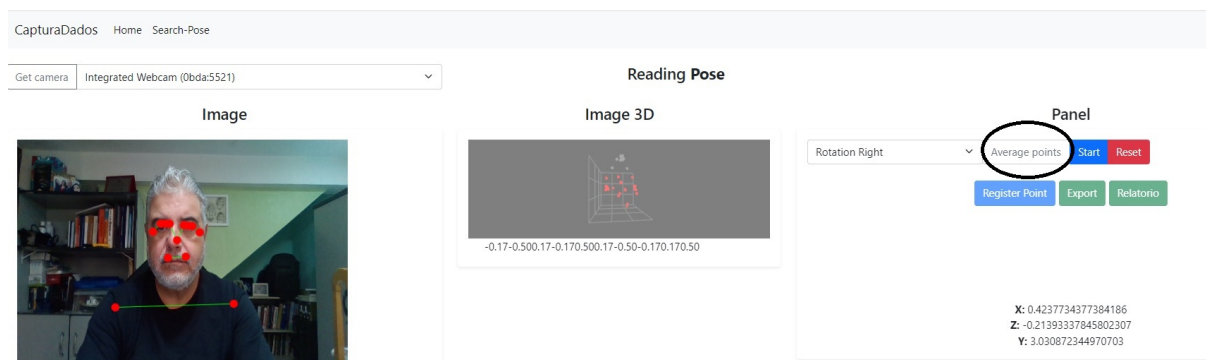
4.2.4 Medições com a câmera do Notebook - Três pontos de leitura

Este teste envolveu a inserção de outros dois pontos no processo de leitura. Esses pontos correspondem aos ombros dos lados direito e esquerdo, conforme apresentado na Figura 44. A ideia foi verificar se o MediaPipe Pose poderia dar uma resposta satisfatória em uma situação onde se adicionaria ao ponto do nariz os pontos dos ombro em relação ao nariz e verificar o ângulo formado entre eles para determinar a posição do nariz. Nesta nova atualização, a opção *Average points* foi mantida e o valor 20 foi novamente adotado. Na apresentação dos valores, os três pontos captados são listados e, no relatório, as informações sobre os pontos armazenados no quadro são também apresentadas.

Na Tabela 18, apresentam-se os valores de posicionamento dos pontos que foram armazenados, o nariz e os dois pontos do ombro, sendo o lado direito indicado pelo (ponto 12) e o esquerdo pelo (ponto 11). Os dados são apresentados em uma única Tabela.

Neste procedimento, como três pontos estão sendo utilizados, optou-se por empregar um procedimento de cálculo que permitisse determinar o ângulo formado entre o nariz, (ponto 0) e os dois pontos do ombro, lado esquerdo e direito. Dessa forma, se poderia determinar se houve

Figura 44 – Relação entre ombro e nariz com valores de média



Fonte: Elaborado pelo autor

Tabela 18 – Comparação entre as medições de mesmo ponto com inserção do valor médio entre o ombro e nariz

Ponto do nariz		Ponto 11 esquerda		Ponto 12 direita	
X	Y	X	Y	X	Y
0,568880	0,429545	0,782398	0,763238	0,388912	0,775194
0,573280	0,430950	0,783094	0,763989	0,388382	0,776367
0,574322	0,430170	0,783417	0,763997	0,388187	0,776768
0,573507	0,430501	0,783173	0,767093	0,387391	0,779703
0,574062	0,428579	0,784476	0,771713	0,386931	0,781035
0,575881	0,426478	0,787140	0,775718	0,385316	0,781663
0,578123	0,425678	0,788894	0,777692	0,386501	0,780737
0,578266	0,424606	0,788663	0,780200	0,384099	0,780213
0,579197	0,427558	0,787628	0,778772	0,384232	0,784194
0,573808	0,422634	0,785998	0,777105	0,382581	0,774484
0,576484	0,426250	0,786704	0,775710	0,381848	0,776622
0,575805	0,429259	0,786733	0,773796	0,385509	0,771836
0,575365	0,427567	0,787223	0,774946	0,385029	0,774368
0,575576	0,423899	0,787098	0,772873	0,385390	0,771561
0,573215	0,420237	0,788693	0,771276	0,387316	0,771363
0,574304	0,430885	0,788158	0,774273	0,388994	0,774625
0,576185	0,424995	0,789132	0,775541	0,386147	0,778312
0,578217	0,425551	0,788491	0,778994	0,384713	0,782974
0,578446	0,425924	0,788216	0,777819	0,386884	0,779251
0,580527	0,426740	0,786509	0,774628	0,388436	0,773092

Fonte: Elaborado pelo autor

movimento entre o nariz, mais próximo do ombro esquerdo ou do direito, e então verificar a posição do nariz.

Utilizou-se o procedimento de cálculo do ângulo formado por dois vetores, sendo o primeiro entre o nariz e o ponto 11, em seguida entre o nariz e o ponto 12. O procedimento de cálculo adotado para determinar o ângulo é apresentado a seguir:

$$\cos \phi = \frac{\langle v, u \rangle}{|v| * |u|} \quad (2)$$

Os cálculos a serem realizados são:

$$\cos \phi = \frac{(x1 * x2) + (y1 * y2)}{\sqrt{(x1^2 + y1^2)} * \sqrt{(x2^2 + y2^2)}} \quad (3)$$

onde $\cos \phi$ = ângulo (graus)

x1 = ponto inicial

x2 = ponto final

y1 = ponto inicial

y2 = ponto final

A Tabela 19 apresenta os dados calculados para os ângulos formados entre o nariz e os dois pontos dos ombros. Pode-se verificar que os valores são muito próximos.

De forma similar ao que foi feito anteriormente, apresentam-se graficamente os valores obtidos na Figura 45. Nesta figura, é possível verificar que o ângulo formado entre o nariz e os pontos 11 e 12 indicam um resultado da leitura, com valores diferentes e uma menor precisão dos dados.

Tabela 19 – Comparação entre as medições de mesmo ponto com inserção do valor médio entre o ombro e nariz

Medidas do ângulo formado entre o nariz ponto 0 e os pontos 11 e 12		
Ângulo nariz e pto 11	Pontos	Ângulo nariz e pto 12
ARCOS= Raiz (eq. 3) pto 11		ARCOS= Raiz (eq. 3) pto 12
0,479554	POSIÇÃO 1	0,480310
0,480121	POSIÇÃO 2	0,479942
0,480201	POSIÇÃO 3	0,479549
0,479697	POSIÇÃO 4	0,479700
0,479000	POSIÇÃO 5	0,479046
0,478528	POSIÇÃO 6	0,478151
0,478487	POSIÇÃO 7	0,477589
0,478075	POSIÇÃO 8	0,477280
0,478616	POSIÇÃO 9	0,477828
0,477772	POSIÇÃO 10	0,477637
0,478590	POSIÇÃO 11	0,478130
0,478990	POSIÇÃO 12	0,479197
0,478650	POSIÇÃO 13	0,478742
0,478691	POSIÇÃO 14	0,477791
0,478335	POSIÇÃO 15	0,477184
0,478851	POSIÇÃO 16	0,479800
0,478483	POSIÇÃO 17	0,477789
0,478308	POSIÇÃO 18	0,477469
0,478529	POSIÇÃO 19	0,477649
0,479296	POSIÇÃO 20	0,477731

Fonte: Elaborado pelo autor

Com os valores obtidos de ângulos, verificamos que não seria possível afirmar que o indivíduo não se movimentou. Dessa forma a partir do resultado obtido com este teste, com captação de dados com o MediaPipe Pose, verificou-se a necessidade de que o corpo seja visualizado por inteiro e não só a face e ombros, para que se tenha uma maior precisão entre os pontos

Figura 45 – Comparação gráfica de medições do nariz aos pontos do ombro calculado



Fonte: Elaborado pelo autor

capturados. Após análise do procedimento utilizado com o MediaPipe Pose, se pode ponderar que com a utilização dos pontos 23 e 24, que correspondem ao quadril (conforme indicado na Figura 16), se poderia fazer com que o aprendizado de máquina oferecesse uma melhor precisão. Mas esses testes não foram realizados pois seria necessário trabalhar com uma pessoa ereta. Para melhorar a captação sem alterar o procedimento adotado, optou-se por utilizar outro método de leitura da face e se passou a utilizar o Face Mesh do MediaPipe. A próxima seção apresenta os procedimentos realizados.

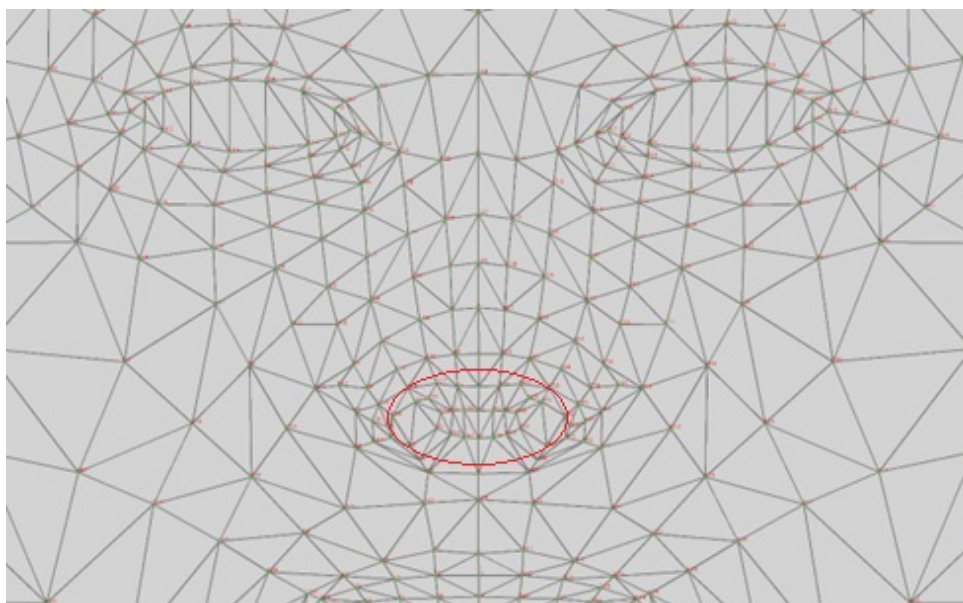
4.3 TESTE ESTÁTICO - MEDIAPIPE FACE MESH

Uma nova versão do sistema foi desenvolvida para a leitura de dados, com a utilização do MediaPipe Face Mesh, conforme o fluxograma apresentado na Figura 36. Nesta etapa, repetiram-se os procedimentos utilizados anteriormente com as mesmas propostas de verificar a repetibilidade e a reprodutibilidade dos experimentos. Nesta primeira abordagem, os pontos a serem analisados estão localizados mais próximos do ponto central do nariz.

Outro fator importante é que, ao utilizar o MediaPipe Face Mesh somente a face é priorizada, aumentando a precisão dos dados lidos. A Figura 46 apresenta os pontos da face que são analisados. A área apresentada em destaque na elipse vermelha nesta Figura indica os pontos que serão efetivamente utilizados.

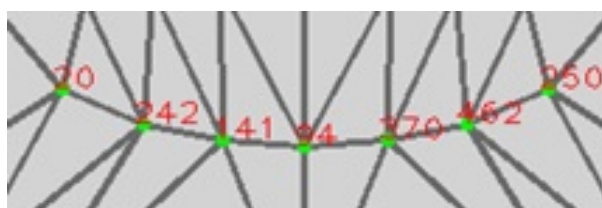
A ampliação da área destacada em vermelho na Figura 46 é apresentada na Figura 47. Nesta, pode-se identificar os pontos 20, 242, 141, 94, 370, 462 e 250, como sendo os pontos mais próximos do ponto central do nariz. Na nova versão do sistema, inicialmente se utilizou os pontos centrais (141, 94 e 370) para os cálculos. Outras análises e testes devem ser realizados para verificar a repetibilidade e precisão desses pontos, não descartando a possibilidade de se utilizarem outros.

Figura 46 – Ampliação da figura para identificação dos pontos



Fonte: (MediaPipe, 2024)

Figura 47 – Pontos próximos ao centro do nariz - 94



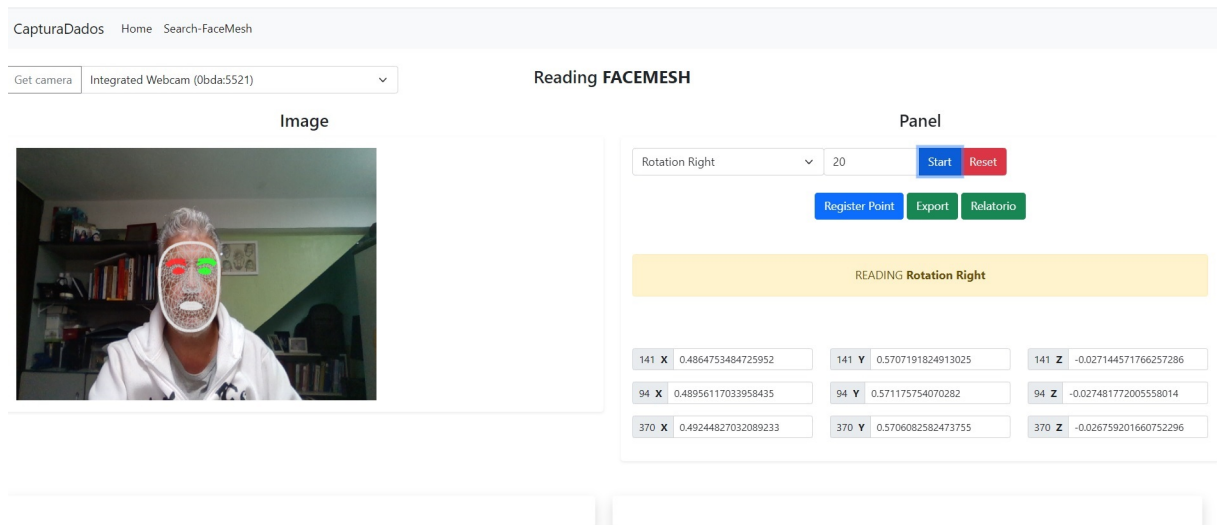
Fonte: (MediaPipe, 2024)

4.3.1 Medições com a câmera do Notebook

Para a execução deste teste, um indivíduo foi posicionado a 80 cm da câmera do notebook, e a estrutura do sistema foi mantida. Utilizou-se o ponto 94 (centro do nariz) como ponto central para a leitura dos dados, e o valor do `Average points` foi mantido em 20. Com o objetivo de analisar o procedimento de leitura do Face Mesh e na possibilidade de verificar se ocorreu algum erro no procedimento de leitura, outros dois pontos, o 141 e 370, foram utilizados, por esses pontos estarem posicionados mais próximos do centro do nariz (ponto 94). A Figura 48 apresenta a interface da nova versão do sistema. Dessa forma, informações detalhadas sobre os valores obtidos foram armazenados e o mesmo processo adotado nos experimentos anteriores foi utilizado. Solicitou-se que o indivíduo buscasse não movimentar a face e que tentasse permanecer estático com uma distância controlada.

Da mesma forma que nos experimentos anteriores, apresentam-se os valores obtidos para os pontos nas Tabelas 20, 21 e 22. Nestas tabelas, apresenta-se os cálculos de distâncias entre os pontos X e Y; foram calculados os valores médios e de desvio padrão. Para o cálculo da movimentação, utilizou-se o mesmo procedimento adotado na fórmula ?? apresentada na seção 4.2.1. Foi mantida a apresentação dos valores com seis casas após a vírgula nas tabelas, conforme

Figura 48 – Medição pontos do nariz com Face Mesh



Fonte: Elaborado pelo autor

apresentado no capítulo 3.2.1.

Tabela 20 – Medições do ponto 141 com o Face Mesh

Leituras	Medidas Pto 141 - M1		Medição pto 141	Mult.1000
	X	Y		
1	0,485427	0,469241	0,000553	0,553096
	0,485496	0,469790		
2	0,485488	0,470034	0,000507	0,506878
	0,485767	0,470458		
3	0,485731	0,470346	0,000259	0,259190
	0,485737	0,470087		
4	0,485975	0,470772	0,000208	0,207716
	0,486112	0,470615		
5	0,486136	0,470641	0,001094	1,094256
	0,487164	0,471017		
6	0,487283	0,470863	0,000152	0,151683
	0,487280	0,471015		
7	0,487617	0,470915	0,000433	0,433022
	0,488046	0,470856		
8	0,488170	0,470811	0,000176	0,175898
	0,488343	0,470780		
9	0,488389	0,470725	0,000117	0,117066
	0,488487	0,470661		
10	0,488487	0,470309	0,000451	0,451043
	0,488850	0,470042		
Valor da média			0,394985	
Desvio padrão			0,277314	

Fonte: Elaborado pelo autor

Tabela 21 – Medições do ponto 94 com o Face Mesh

Leituras	Medidas Pto 94 - M1		Medição pto 94	Mult.1000
	X	Y		
1	0,487224	0,469427	0,000574	0,574141
	0,487302	0,469995		
2	0,487312	0,470269	0,000477	0,477270
	0,487606	0,470645		
3	0,487567	0,470582	0,000261	0,261003
	0,487549	0,470322		
4	0,487802	0,470974	0,000199	0,198763
	0,487942	0,470833		
5	0,487997	0,470863	0,001100	1,099581
	0,489030	0,471242		
6	0,489087	0,471089	0,000159	0,158767
	0,489124	0,471243		
7	0,489440	0,471161	0,000475	0,474743
	0,489909	0,471088		
8	0,490029	0,471024	0,000163	0,163252
	0,490186	0,470980		
9	0,490236	0,470940	0,000120	0,119804
	0,490339	0,470879		
10	0,490319	0,470527	0,000482	0,482035
	0,490701	0,470234		
Valor da média			0,400936	
Desvio padrão			0,281173	

Fonte: Elaborado pelo autor

Tabela 22 – Medições do ponto 370 com o Face Mesh

Leituras	Medidas Pto 370 - M1		Medição pto 370	Mult.1000
	X	Y		
1	0,489083	0,469216	0,000563	0,562782
	0,489168	0,469772		
2	0,489170	0,470040	0,000494	0,493595
	0,489468	0,470434		
3	0,489426	0,470339	0,000280	0,279985
	0,489396	0,470060		
4	0,489677	0,470730	0,000215	0,214621
	0,489819	0,470569		
5	0,489857	0,470623	0,001116	1,116381
	0,490907	0,471003		
6	0,490965	0,470883	0,000162	0,162440
	0,491042	0,471026		
7	0,491326	0,470927	0,000463	0,463049
	0,491785	0,470872		
8	0,491875	0,470823	0,000184	0,183716
	0,492054	0,470782		
9	0,492120	0,470741	0,000112	0,112385
	0,492205	0,470667		
10	0,492164	0,470316	0,000457	0,456853
	0,492554	0,470078		
Valor da média			0,394985	
Desvio padrão			0,277314	

Fonte: Elaborado pelo autor

Os valores apresentados nas Tabelas 20, 21 e 22 onde temos medias inferiores a 0,5 cm o que nos permite afirmar que as leituras realizadas são coerentes e esses valores atendem aos objetivos buscados pela pesquisa. Pode-se observar que os valores apresentam uma boa repetibilidade e na tabela 23 temos os valores de media e variância para as três tabelas. Dessa forma efetuamos o calculo de teste t e obtivemos um valor $p > 0,05$ o que indica as medias nas leituras não apresentam valores significativos. Este fato é confirmado pela representação gráfica dos pontos que pode ser observada na Figura 49.

Tabela 23 – Tabela com cálculos de média e variância

Anova: fator único				
RESUMO				
Grupo	Contagem	Soma	Média	Variância
Medida pto. 141	10	3,949849	0,394984	0,085448
Medida pto. 94	10	4,009359	0,400935	0,087842
Medida pto. 370	10	4,045807	0,404580	0,087795

Fonte: Elaborado pelo autor

Figura 49 – Comparação gráfica de medições de pontos diferentes, mas de mesmo quadro e ao mesmo tempo com cálculos



Fonte: Elaborado pelo autor

Para avançar ainda mais na busca de precisão da leitura, optou-se por realizar os experimentos com novas abordagens, sendo que elas são apresentadas nas próximas seções. Conforme se irá detalhar, ao serem realizados os experimentos, observou-se que, em algumas medições, parece existir a possibilidade de alteração dos dados obtidos de acordo com a cor do fundo da tela de captação. Assim, na próxima seção, realizam-se as medições utilizando cores de fundos diferentes.

4.3.2 Medições com câmera Notebook usando FaceMesh com alteração das cores de fundo

Realizaram-se medições com cores de fundo diferentes para verificar se existem alterações nos dados obtidos. Para esses experimentos, optou-se por utilizar o boneco apresentado na Figura 50 em posição fixa, e utilizando a mesma distância de 80 cm e com *Average points* de valor 10, sendo alterados os fundos, conforme indicado na Figura 50.

Replicou-se o teste realizado com o FaceMesh para diferentes cores de fundo. Foram utilizadas as cores Preto, Branco, Verde, Azul, Rosa e Rosa claro e optou-se por apresentar as imagens em tabelas a cada duas cores. Realizamos os calculos ANOVA para o valores das tabelas

Figura 50 – Imagem do boneco posicionado frente a câmera

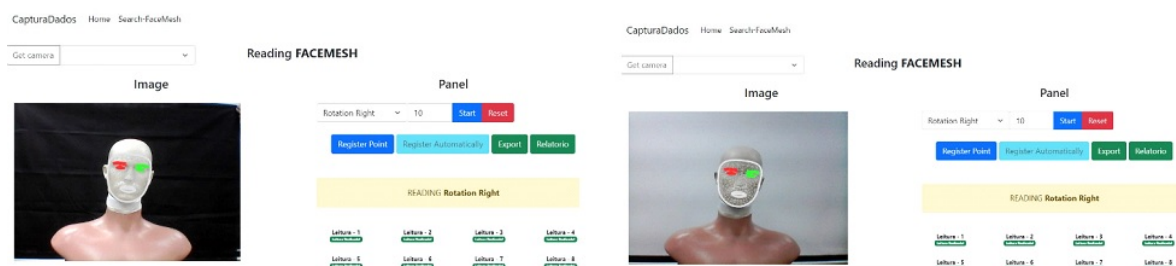


Fonte: Elaborado pelo autor

e para os próximos experimentos com as diferentes cores de fundo, apresenta-se as tabelas e figuras com os dados obtidos em cada um dos experimentos, sendo apresentados os valores de média e desvio padrão. Dessa forma, pode-se verificar as diferenças dos valores obtidos e relacioná-los às cores utilizadas.

Na Tabela 24, apresentam-se os cálculos de média e desvio padrão das medidas obtidas com as cores de fundo preto e branco apresentadas na Figura 51. Esses valores permitem verificar também a repetibilidade das medições conforme a alterações das cores dos fundos.

Figura 51 – Imagens das cores de fundos pretos e brancos



Fonte: Elaborado pelo autor

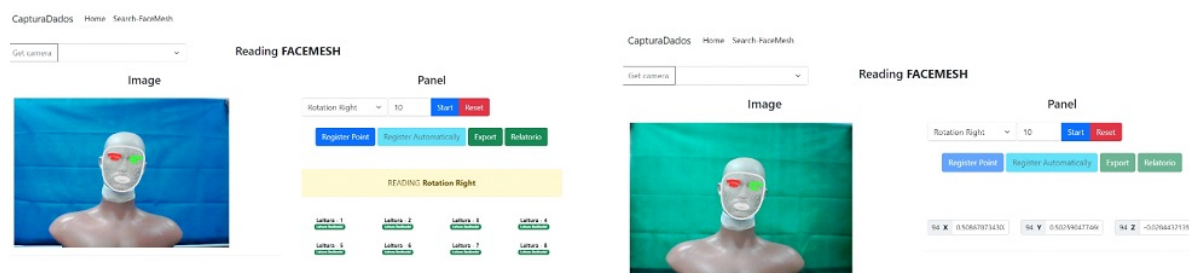
Tabela 24 – Valores obtidos com o fundo preto e branco

Leitura	Valores obtidos com fundo preto			Valores obtidos com fundo branco		
	X	Y	Mult.1000	X	Y	Mult.1000
1	0,506004	0,528741	0,814386	0,508685	0,530024	0,612408
	0,506060	0,527929		0,508075	0,529972	
2	0,506306	0,529118	0,564788	0,508481	0,529787	0,413806
	0,506415	0,528564		0,508883	0,529685	
3	0,506724	0,529084	0,747499	0,508467	0,529170	0,511770
	0,505977	0,529088		0,507977	0,529316	
4	0,505905	0,529118	0,460264	0,507474	0,525079	1,054810
	0,505903	0,529578		0,506728	0,525826	
5	0,505635	0,529083	0,230778	0,507271	0,525769	0,114139
	0,505511	0,529277		0,507161	0,525736	
6	0,506413	0,529500	0,907688	0,507395	0,525776	0,302572
	0,505748	0,528882		0,507609	0,525990	
7	0,505978	0,528994	0,702311	0,507541	0,526222	0,582856
	0,505738	0,529654		0,507633	0,525647	
8	0,505640	0,529172	0,257386	0,506856	0,526103	0,871354
	0,505827	0,529349		0,507727	0,526129	
9	0,505691	0,528558	0,497802	0,507204	0,525686	0,346509
	0,505819	0,529039		0,506860	0,525650	
10	0,506212	0,528923	0,540731	0,507436	0,526004	0,467197
	0,505781	0,529250		0,507173	0,525618	
	Valor da Média		0,572363	Valor da Média		0,527742
	Desvio Padrão		0,212991	Desvio Padrão		0,260306

Fonte: Elaborado pelo autor

As próximas cores de fundo analisadas foram verde e azul e o experimento é apresentado na Figura 52. Novamente não houve movimentação do boneco e foram captados 20 pontos. A Tabela 25 apresenta os valores obtidos e os cálculos realizados.

Figura 52 – Imagens das cores de fundos azul e verde



Fonte: Elaborado pelo autor

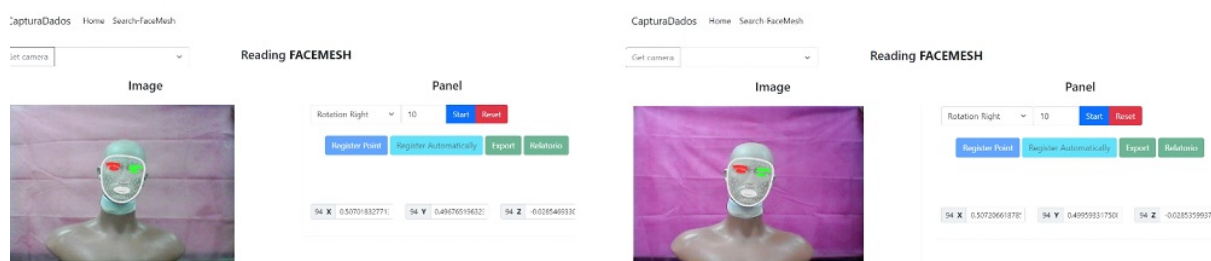
Na sequência se utilizou cores rosa e rosa claro, conforme apresentado na Figura 53, que mostra as telas de captura de dados. Repetindo o processo sem a movimentação do boneco e captando os 20 pontos, a Tabela 26 exibe os valores obtidos após cálculos.

Tabela 25 – Valores obtidos com o fundo azul e verde

Leitura	Valores obtidos com fundo azul				Valores obtidos com fundo verde		
	X	Y	Mult.1000		X	Y	Mult.1000
1	0,501178	0,529660	0,133744		0,506428	0,527413	0,934779
	0,501285	0,529579			0,506739	0,526531	
2	0,500849	0,529821	0,127135		0,507229	0,527041	0,411199
	0,500969	0,529777			0,506825	0,526969	
3	0,500319	0,530362	0,703126		0,506723	0,527660	1,125649
	0,501021	0,530321			0,506660	0,526536	
4	0,501722	0,529181	1,061168		0,507145	0,526882	0,279835
	0,500886	0,529834			0,506917	0,526720	
5	0,500738	0,529656	0,530957		0,506748	0,526663	0,163543
	0,501016	0,530108			0,506912	0,526664	
6	0,500880	0,529242	0,536559		0,506887	0,526448	0,233975
	0,500939	0,529775			0,507007	0,526648	
7	0,500830	0,530144	0,725244		0,506996	0,527118	0,316882
	0,500403	0,529558			0,506860	0,526832	
8	0,500593	0,529775	0,197983		0,507029	0,526987	0,306331
	0,500563	0,529580			0,506724	0,527024	
9	0,500655	0,530093	0,496949		0,506635	0,527408	0,717788
	0,500491	0,529624			0,507124	0,526883	
10	0,500456	0,529662	0,220640		0,506533	0,526631	0,553390
	0,500291	0,529808			0,507071	0,526762	
	Valor da Média		0,473351		Valor da Média		0,504337
	Desvio Padrão		0,290609		Desvio Padrão		0,306909

Fonte: Elaborado pelo autor

Figura 53 – Imagens das cores de fundos rosa e rosa claro



Fonte: Elaborado pelo autor

Os valores apresentados nas Tabelas 24, 25 e 26 onde temos médias próximas de 0,5 cm, excetuando a cor rosa claro, o que nos permite afirmar que as leituras realizadas são coerentes e esses valores atendem aos objetivos buscados pela pesquisa. Pode-se observar que os valores apresentam uma boa repetibilidade e na tabela 27 temos os valores de media e variância para as três tabelas. Dessa forma efetuamos o calculo de teste t e obtivemos um valor $p > 0,05$ o que indica as médias nas leituras não apresentam valores significativos. Apresentamos os gráficos Figura 54, Figura 55 e Figura 56 que nos permite visualizar essa diferença.

Analisando novamente os valores obtidos com os cálculos apresentados nas tabelas, é possível afirmar que as diferentes cores de fundo exercem uma influência mínima nos valores obtidos. Como não foi possível determinar esse valor exato com os experimentos realizados, pode-se

Tabela 26 – Valores obtidos com o fundo rosa e rosa claro

Leitura	Valores obtidos com fundo rosa			Valores obtidos com fundo rosa claro		
	X	Y	Mult.1000	X	Y	Mult.1000
1	0,508780	0,526459	0,436028	0,510061	0,525505	0,926131
	0,508588	0,526068		0,510985	0,525572	
2	0,509080	0,525982	0,728879	0,511227	0,524681	1,465158
	0,508386	0,526206		0,510551	0,525981	
3	0,507863	0,525845	0,280648	0,510855	0,525856	1,208005
	0,507583	0,525864		0,512053	0,526014	
4	0,508699	0,525410	0,607798	0,510795	0,525130	0,841998
	0,508746	0,526016		0,510331	0,525833	
5	0,508869	0,526231	0,553923	0,510813	0,525410	0,640583
	0,508352	0,526034		0,510414	0,525910	
6	0,508472	0,526082	0,811475	0,510474	0,525494	0,782257
	0,507839	0,525574		0,511068	0,526002	
7	0,508366	0,525817	0,395303	0,510582	0,525567	0,997799
	0,508569	0,525477		0,511418	0,525023	
8	0,508257	0,526080	0,294646	0,510639	0,525423	0,294513
	0,508213	0,525789		0,510904	0,525296	
9	0,507712	0,525574	1,009881	0,511275	0,525617	0,313194
	0,508645	0,525961		0,511257	0,525305	
10	0,508285	0,525766	0,362449	0,510201	0,525898	0,670527
	0,507991	0,525979		0,510479	0,525287	
	Valor da Média		0,548103	Valor da Média		0,814016
	Desvio Padrão		0,229135	Desvio Padrão		0,346226

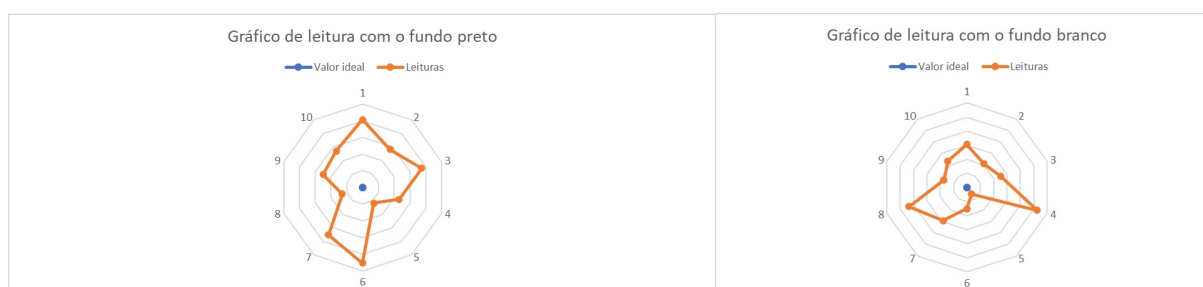
Fonte: Elaborado pelo autor

Tabela 27 – Tabela com cálculos de média e variância

Anova: fator único				
RESUMO				
Grupo	Contagem	Soma	Média	Variância
Preto	10,000000	5,723633	0,572363	0,050406
Branco	10,000000	5,277420	0,527742	0,075288
Azul	10,000000	4,733506	0,473351	0,093837
Verde	10,000000	5,043372	0,504337	0,104659
Rosa	10,000000	5,481030	0,548103	0,058337
Rosa claro	10,000000	8,140164	0,814016	0,133192

Fonte: Elaborado pelo autor

Figura 54 – Comparação gráfica de medições com fundos pretos e brancos, com boneco e distância fixa



Fonte: Elaborado pelo autor

Figura 55 – Comparação gráfica de medições com fundos azul e verde, com boneco e distância fixa

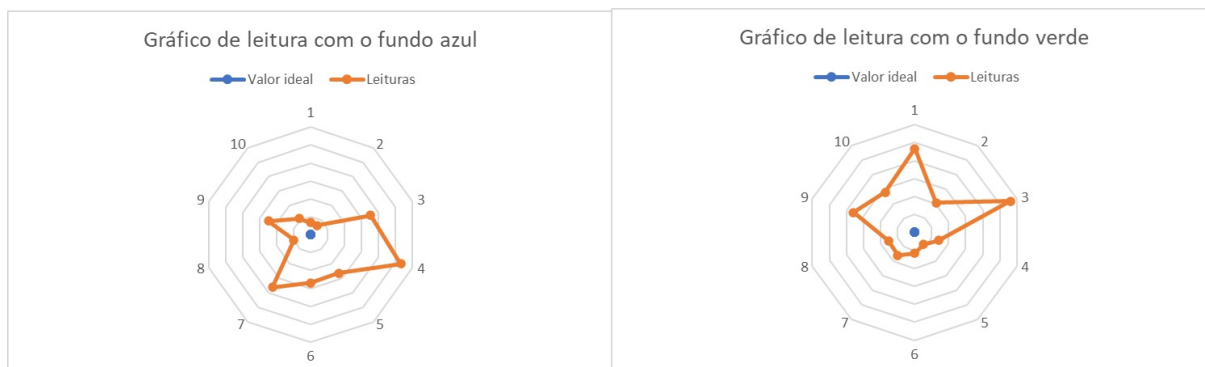


Figura 56 – Comparação gráfica de medições com fundos rosa e rosa claro, com boneco e distância fixa



afirmar que as diferentes cores não alteram a utilização do sistema. Como as cores preto e rosa apresentam um menor desvio padrão nos valores obtidos se decidiu pela utilização de uma cor de fundo escura ou de cores neutras e de preferência cores que reflitam menos luz e com maior grau de absorção.

A próxima seção apresenta outro tipo de teste que foi feito a partir de medições para determinar qual a influência da distância do boneco em relação a câmera e em diferentes equipamentos.

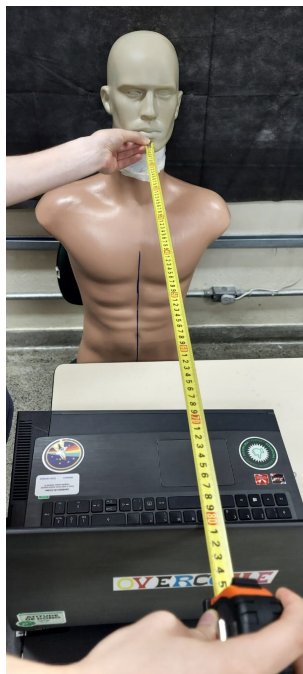
4.3.3 Medições com câmera Notebook usando FaceMesh com Boneco em diferentes distâncias

A proposta nestes experimentos é determinar se pequenas variações de distância entre o nariz do boneco e a câmera e diferentes modelos de notebooks, produzem alterações significativas nos valores das medições.

Neste experimento foi utilizado um fundo preto se alterando os modelos de notebook e as distâncias utilizadas nas medições, sendo elas 75, 80 e 85 cm. Apresenta-se na Figura 57 a forma

como os dados foram obtidos e como a medida de 80 cm de distância da câmera até o nariz do boneco foi realizada.

Figura 57 – Distância de 80 cm



Fonte: Elaborado pelo autor

O mesmo procedimento foi utilizado para os notebooks diferentes, na mesma posição e com medições de distância sendo realizadas de forma similar para a obtenção dos dados. Realizamos o teste ANOVA nos valores de distância do mesmo notebook para comparar a média e assim identificar se essas médias diferem significativamente entre elas.

A Tabela 28 apresenta os valores obtidos com o Notebook 01, um modelo DELL Latitude 7400 e nela se pode observar que a média e o desvio padrão entre as três medições em distância diferentes apresentam valores muito próximos.

Tabela 28 – Tabela Notebook - DELL Latitude 7400

Leitura	Distância 75 cm		Distância 80 cm		Distância 85 cm	
	Distância	Mult.1000	Distância	Mult.1000	Distância	Mult.1000
1	0,000212	0,211659	0,000281	0,281322	0,000462	0,462409
2	0,000189	0,189292	0,000566	0,565888	0,000381	0,381277
3	0,000365	0,364587	0,000052	0,051624	0,000221	0,220804
4	0,000146	0,145646	0,000227	0,226893	0,000085	0,085203
5	0,000122	0,122273	0,071710	0,071710	0,000336	0,336236
6	0,000242	0,242233	0,485647	0,485647	0,000240	0,239980
7	0,000030	0,030432	0,145628	0,145628	0,000322	0,322018
8	0,000226	0,226037	0,236005	0,236005	0,000493	0,493016
9	0,000236	0,235796	0,178092	0,178092	0,000072	0,072041
10	0,000150	0,149512	0,101006	0,101006	0,000276	0,276186
	Valor da Média	0,191746		0,234381		0,288917
	Desvio Padrão	0,084083		0,162477		0,074827

Fonte: Elaborado pelo autor

Na tabela 29 ANOVA realizamos uma análise de variância para comparar as médias de três grupos de valores com o objetivo de verificar se existem diferenças significativas entre eles. O valor p ajuda a determinar se a estatística observada é significativa. Um valor p acima de $> 0,05$ indica que as diferenças nas médias do grupo não são estatisticamente significativas.

Tabela 29 – Análise ANOVA da tabela Notebook - DELL Latitude 7400

Anova: fator único				
RESUMO				
Grupo	Contagem	Soma	Média	Variância
75 Cm	10	1,917467726	0,191746773	0,007855514
80 cm	10	2,343813725	0,234381373	0,029332242
85 cm	10	2,889170313	0,288917031	0,019927995
ANOVA				
Fonte da variação	SQ	gl	MQ	valor-P
Entre grupos	0,047446355	2	0,023723177	0,303655224
Dentro dos grupos	0,514041755	27	0,019038584	
Total	0,561488109	29		

Fonte: Elaborado pelo autor

Repetiu-se o experimento para o Notebook 02, modelo Aspire 3 AMD Ryzen 5 3500U e a Tabela 30 apresenta os valores obtidos. Novamente observa-se que a média e o desvio padrão entre as medições apresentam valores muito próximos.

Tabela 30 – Tabela Notebook - Aspire 3 AMD Ryzen 5 3500U

Leitura	Distância 75 cm		Distância 80 cm		Distância 85 cm	
	Distância	Mult.1000	Distância	Mult.1000	Distância	Mult.1000
1	0,000159	0,159251	0,000379	0,379235	0,000135	0,135074
2	0,000097	0,096758	0,000134	0,134380	0,000167	0,167027
3	0,000205	0,205074	0,000151	0,151105	0,000251	0,250912
4	0,000522	0,521593	0,000250	0,250141	0,000145	0,144848
5	0,000291	0,290936	0,000139	0,139377	0,000051	0,050695
6	0,000293	0,293322	0,000135	0,134534	0,000441	0,440778
7	0,000176	0,175705	0,000164	0,163550	0,000427	0,426814
8	0,000137	0,136579	0,000356	0,356403	0,000182	0,182044
9	0,000185	0,184578	0,000277	0,276888	0,000287	0,286555
10	0,000236	0,236043	0,000405	0,404681	0,000099	0,099247
	Valor da Média	0,229984		0,239029		0,218399
	Desvio Padrão	0,113975		0,103760		0,125390

Fonte: Elaborado pelo autor

Na tabela 31 ANOVA realizamos novamente a análise de variância para comparar as médias de três grupos de valores, utilizando um outro equipamento e procurando manter o mesmo ambiente de teste e observamos que não existem diferenças significativas entre os valores obtidos. O valor p maior que $> 0,05$ nos permite afirmar que as diferenças nas médias não são estatisticamente significativas.

Tabela 31 – Análise ANOVA da tabela Notebook - Aspire 3 AMD Rysen 5 3500U

Anova: fator único				
RESUMO				
Grupo	Contagem	Soma	Média	Variância
75 Cm	10	2,299840143	0,229984014	0,014433867
80 cm	10	2,390294172	0,239029417	0,011962307
85 cm	10	2,183994717	0,218399472	0,017469634
ANOVA				
Fonte da variação	SQ	gl	MQ	valor-P
Entre grupos	0,002138719	2	0,001069359	0,929659839
Dentro dos grupos	0,394792277	27	0,014621936	
Total	0,396930995	29		

Fonte: Elaborado pelo autor

Nas mesmas condições, passou-se aos testes com o Notebook 03, modelo Lenovo G40 70. Os valores obtidos e apresentados na Tabela 32 uma vez mais mostram valores muito próximos para a média e o desvio padrão entre as medições.

Tabela 32 – Tabela Notebook - Lenovo G40 70

Leitura	Distância 75 cm		Distância 80 cm		Distância 85 cm	
	Distância	Mult.1000	Distância	Mult.1000	Distância	Mult.1000
1	0,000095	0,095172	0,000059	0,058797	0,000413	0,412635
2	0,000387	0,387410	0,000260	0,260451	0,000296	0,296182
3	0,000522	0,522244	0,000208	0,208120	0,000351	0,351134
4	0,000535	0,534649	0,000615	0,615063	0,000267	0,266992
5	0,000422	0,422391	0,000288	0,288446	0,000414	0,413820
6	0,000183	0,183339	0,000611	0,611038	0,000242	0,242378
7	0,000451	0,450944	0,000337	0,336794	0,000664	0,664024
8	0,000121	0,120869	0,000292	0,291640	0,000185	0,185248
9	0,000308	0,308333	0,000270	0,269879	0,000472	0,471854
10	0,000251	0,251035	0,000106	0,105746	0,000314	0,314440
	Valor da Média	0,327639		0,304597		0,361871
	Desvio Padrão	0,152426		0,174408		0,130317

Fonte: Elaborado pelo autor

A análise dos dados apresentados nas três tabelas permite afirmar que é possível utilizar notebooks diferentes e obter valores apropriados aos objetivo buscado, com erros menores que 0,35 cm de média. Dessa forma podemos concluir que se tem uma boa repetibilidade e reprodutibilidade dos dados lidos utilizando diferentes dispositivos. Na próxima seção se

Tabela 33 – Analise ANOVA da tabela Notebook - Lenovo G40 70

Anova: fator único				
RESUMO				
Grupo	Contagem	Soma	Média	Variância
75 cm	10	3,276386629	0,327638663	0,025815207
80 cm	10	3,045973352	0,304597335	0,03379792
85 cm	10	3,61870641	0,361870641	0,018869377
ANOVA				
Fonte da variação	SQ	gl	MQ	valor-P
Entre grupos	0,016609876	2	0,008304938	0,7306784
Dentro dos grupos	0,706342536	27	0,026160835	
Total	0,722952411	29		

Fonte: Elaborado pelo autor

apresentam experimentos que foram realizados para verificar se uma mudança no modelo de câmeras, altera a os resultados obtidos. Passamos a realizar o teste com câmeras da RealSense.

4.3.4 Medições com a câmera Intel RealSense D435i e D455

Para este experimento, utilizou-se o MediaPipe Face Mesh com o mesmo sistema que foi utilizado com as medições realizadas anteriormente. Entretanto foi capturado e analisado somente o ponto 94. A proposta desse teste foi verificar a reprodutibilidade dos dados com outros modelos de câmeras. Foram utilizados os seguintes sensores de movimento:

- Câmera Intel RealSense D435i - 1280x720 - Frame Rate 30 fps
- Câmera Intel RealSense D455 - 1280x720 - Frame Rate 30 fps

Optou-se por esses sensores de movimento, pela facilidade de acesso e por serem dispositivos recentes no mercado que possuem lentes com ângulos diferentes para a captação de imagem. Isso permitiu verificar se num mesmo ambiente, com diferentes câmeras, os dados dos quadros que foram armazenados são similares. Da mesma forma que os experimentos anteriores, não houve movimento da face.

A Figura 58 e Figura 59 apresentam a montagem similar feita para as duas câmeras, sendo elas montadas em um tripé para evitar movimentação durante o processo e posicionadas a 80 cm de distância do nariz. No teste se substituiu um modelo de câmera pelo outro para realizar os procedimentos de mesma forma que os anteriores para evitar alterações nos resultados.

Figura 58 – Câmera RealSense D435i posicionada no tripe



Fonte: Elaborado pelo autor

De forma similar ao apresentado seção 4.2.3, e com a mesma estrutura para as duas câmeras, foram realizadas 20 leituras de captação de imagem, sem a movimentação da face e com o mesmo valor médio de 20 leituras em cada quadro.

A Tabela 34 apresenta os valores da média e do desvio padrão obtidos com a câmera D435i posicionada a 80 cm do indivíduo. Esses dados são provenientes de cálculos que foram realizados

Figura 59 – Câmera RealSense D455 posicionada no tripe



Fonte: Elaborado pelo autor

após se solicitar ao indivíduo que não movimentasse a face entre as medições. Em seguida, o processo foi repetido com uma nova câmera, a RealSense, a D455, tendo sido mantida a mesma distância. A Tabela 35 apresenta os valores lidos e os cálculos realizados nas mesmas condições, ou seja, sem a movimentação da face. Ao analisar-se a Tabela 34 e a Tabela 35 observa-se que os valores de distância obtidos entre os diferentes modelos de câmeras, estão bem próximos entre eles.

No gráfico apresentado na Figura 60, pode-se observar que os valores de distância, de média e desvio padrão entre as leituras indicam que as câmeras RealSense D435i e D455 apresentam boas características de repetibilidade. É importante realizar mais experimentos para afirmar que elas também apresentam uma boa reprodutibilidade dos valores obtidos. Com uma primeira análise, pode-se afirmar que com a "face estática", as leituras realizadas indicam que as câmeras da RealSense podem atender às necessidades do sistema proposto. Entretanto, testes suplementares devem ser realizados com o reposicionamento da face entre as leituras para confirmar que esses sensores atendem às necessidades apresentadas por este trabalho de pesquisa.

Na sequência deste documento, a partir da seção 4.4, apresenta-se os testes dinâmicos realizados com a utilização do capacete para observar a partir de movimentos controlados, que deslocamento a movimentação da cabeça de um indivíduo sob análise produziria com ele posicionado a uma distância de 80cm do sensor de movimento ou da câmera. Para todos os experimentos a seguir, utilizou-se o procedimento de cálculo indicado na fórmula ?? apresentado na seção 4.2.1.

Tabela 34 – Medições de mesmo ponto com Face Mesh utilizando câmera intel D435i RealSense

Leituras	Medidas D435i		Medição pto 1	Mult.1000
	X	Y		
1	0,526143	0,510589	0,000729	0,729281
	0,526399	0,511272		
2	0,526317	0,511444	0,000277	0,276935
	0,526404	0,511707		
3	0,525942	0,511857	0,000454	0,453815
	0,526016	0,512305		
4	0,526351	0,512235	0,000316	0,316392
	0,526376	0,512550		
5	0,526425	0,512971	0,000237	0,236812
	0,526188	0,512973		
6	0,525937	0,513556	0,000284	0,284473
	0,525957	0,513840		
7	0,526025	0,513766	0,000556	0,555799
	0,525510	0,513974		
8	0,526105	0,514457	0,000627	0,627348
	0,525989	0,515074		
9	0,525512	0,515391	0,001064	1,063676
	0,526161	0,516234		
10	0,525256	0,515696	0,000453	0,452653
	0,524911	0,515989		
Valor da média			0,499718	
Desvio padrão			0,243556	

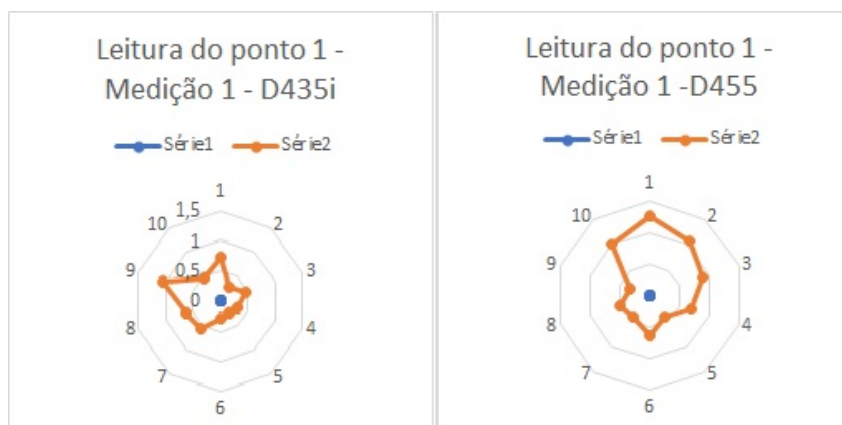
Fonte: Elaborado pelo autor

Tabela 35 – Medições de mesmo ponto com Face Mesh utilizando câmera intel D455 RealSense

Leituras	Medidas D455		Medição pto 1	Mult.1000
	X	Y		
1	0,492918	0,480672	0,000502	0,501783
	0,492533	0,480993		
2	0,492116	0,481104	0,000436	0,435575
	0,491967	0,481513		
3	0,491972	0,481455	0,000364	0,364352
	0,492251	0,481220		
4	0,492170	0,481511	0,000285	0,285001
	0,491982	0,481297		
5	0,491886	0,481249	0,000172	0,172215
	0,491791	0,481105		
6	0,491678	0,480950	0,000249	0,249181
	0,491639	0,481197		
7	0,491762	0,480927	0,000168	0,168300
	0,491813	0,480767		
8	0,492043	0,480793	0,000194	0,194338
	0,492202	0,480681		
9	0,491913	0,480656	0,000125	0,125485
	0,491796	0,480610		
10	0,492187	0,480810	0,000407	0,407296
	0,492470	0,480517		
Valor da média			0,290352	
Desvio padrão			0,123256	

Fonte: Elaborado pelo autor

Figura 60 – Comparação gráfica de medições do mesmo ponto calculado

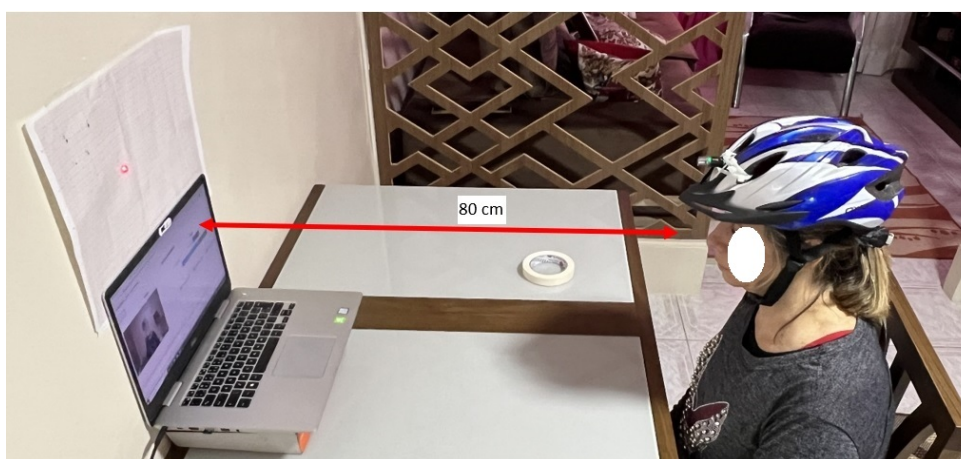


Fonte: Elaborado pelo autor

4.4 TESTES DINÂMICOS

Até essa etapa os testes foram realizados de forma estática, ou seja, sem movimentação da face. A partir desta seção apresentam-se os testes dinâmicos, que contemplam experimentos com movimentações da face sob distâncias de deslocamento de face controladas. Para a realização dos testes, o indivíduo sob análise deveria usar um capacete com um laser acoplado e estar posicionado a uma distância de 80 cm de uma folha de papel milimetrado fixado em uma parede. Ele deveria também posicionar sua face para a câmera do notebook conforme apresentado na Figura 61 e na Figura 62.

Figura 61 – Indivíduo posicionado a 80 cm do computador para realização de testes dinâmicos



Fonte: Elaborado pelo autor

Baseando na figura 14, realizamos um cálculo do ângulo no teste TRC, onde o valor da movimentação é de 5 cm e a distância é de 80 cm, dessa forma, obtemos o ângulo de 3.576 graus. Baseando-se nesses cálculos, determinamos o fator de multiplicação para repetir esses valores a ser multiplicado pelas distâncias obtidas pela câmera, e o valor do fator é 936 e não 1000 que vinha sendo utilizado. Para cálculos sem movimentação esse valor nos permitiu verificar se os

Figura 62 – Visualização do indivíduo com o capacete posicionado a 80 cm do computador para realização de testes dinâmicos



Fonte: Elaborado pelo autor

dados obtidos estavam coerentes com a proposta em relação a repetibilidade. Dessa forma a partir dos próximos testes e na primeira tabela usaremos o valor do fator como multiplicador.

Para permitir uma análise dos valores obtidos com o sistema e a movimentação da face, se iniciou os testes com uma distância controlada e fixa entre o ponto inicial e final. Sem exigir que o indivíduo respeitasse alguma restrição de tempo, foi solicitado que ele realizasse um movimento da face com deslocamento de 5cm e de 10cm em relação ao ponto inicial, para que a indicação do laser fosse anotada na folha de papel milimetrado e capturada pelo sistema. Esse processo foi repetido 20 vezes até que todos os valores fossem armazenados, tanto com os modelos de câmeras RealSense como com o notebook. Verificou-se que os valores captados pelo sistema eram compatível com o deslocamento anotado no papel milimetrado.

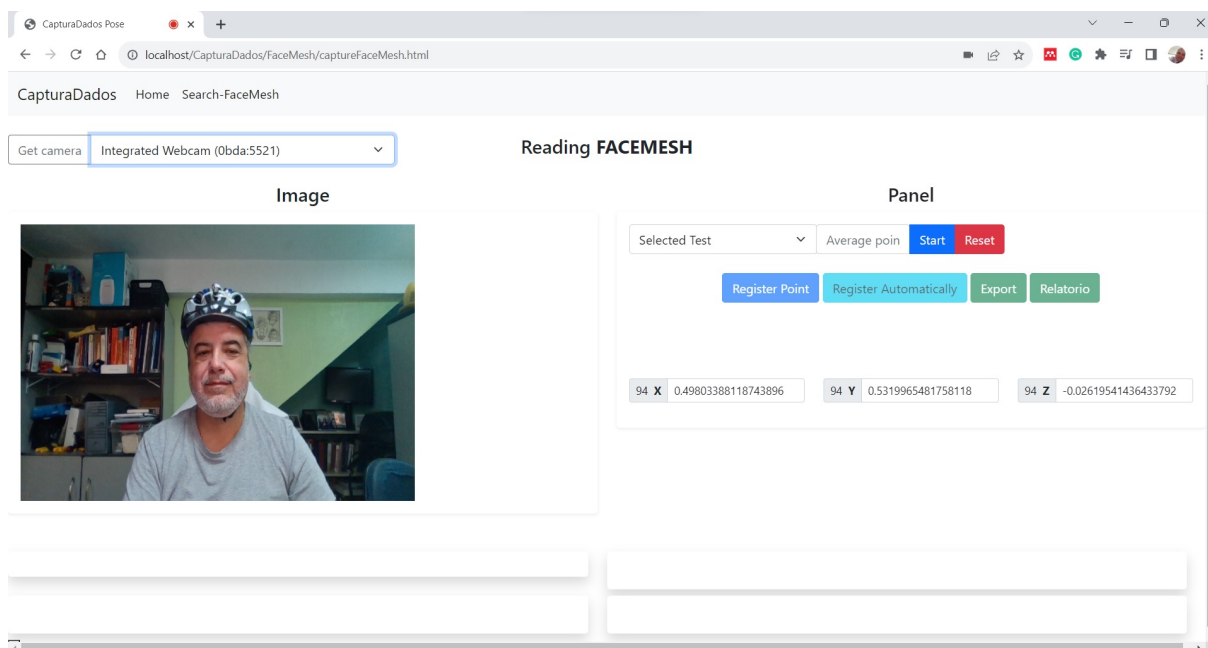
Um detalhe que se deve observar, é que nas novas interfaces da ferramenta, nos testes dinâmicos optou-se por não apresentar a malha facial do MediaPipe, que foi facilmente oculta no programa.

4.4.1 Medições com câmera de notebook - Teste 01

Para o teste apresentado nesta seção, utilizou-se a câmera do notebook modelo Dell Inspiron 7580, com o indivíduo utilizando o capacete e o laser apontado para um alvo (folha milimetrada). Com o laser apontado para o primeiro ponto, captou-se com o sistema o posicionamento da face conforme apresentado na Figura 63. Na sequência se solicitou ao indivíduo que deslocasse a cabeça e o laser do capacete para um segundo ponto posicionado a 5cm do primeiro, para com o sistema fazer uma nova captura de dados. Na sequência ele deveria reposicionar sua cabeça na posição original. Este procedimento foi repetido 20 vezes.

Utilizou-se o mesmo procedimento de captação de posição da face que foi apresentado na

Figura 63 – Tela do sistema de captura dados com o Face Mesh



Fonte: Elaborado pelo autor

Figura 48. Conforme explicado, nesse experimento, somente o ponto 94 foi utilizado, pois, em experimentos anteriores, verificou-se que não existe a necessidade de comparação com outros pontos próximos do nariz.

Na Tabela 36, apresentam-se os cálculos de distância realizados. Esses valores foram comparados ao valor de controle de 5 cm do movimento que foi realizado com o laser e o capacete. A partir do resultado de distância em centímetros obtido, foi possível comparar as duas leituras: a realizada com o capacete e o papel milimetrado e com o sistema. A partir dos dados apresentados nesta tabela, foi inserida a indicação de deslocamento em graus, que vai constar nos próximos experimentos.

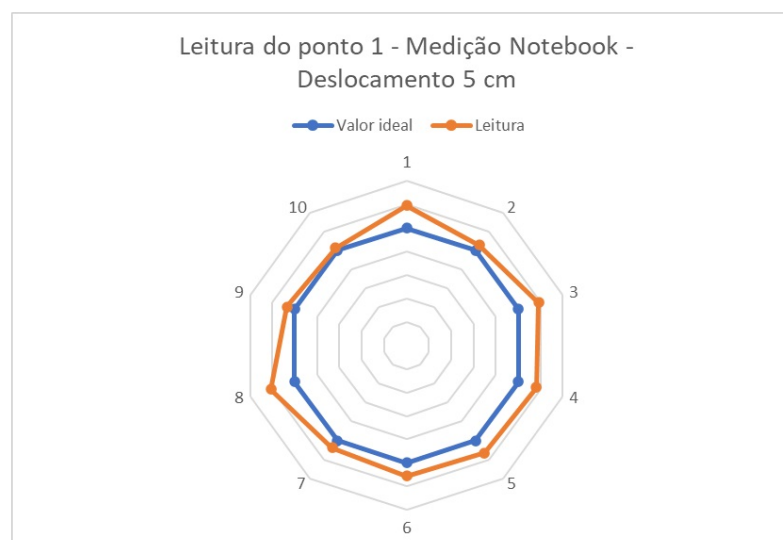
De forma similar ao que foi descrito anteriormente, na Figura 64 apresenta-se graficamente os valores obtidos com a leitura do sistema. Neste gráfico a linha azul indica os valores de movimentação esperados, com o deslocamento controlado em 5 cm e a linha laranja os dados obtidos pelo sistema. Nota-se que ainda não foi obtido um valor preciso, onde as linhas deveriam se sobrepor, embora o cálculo do desvio padrão apresente um valor aceitável. Outros testes devem ser realizados, principalmente com modelos diferentes de câmera. Tais testes foram realizados e são apresentados nas próximas seções.

Tabela 36 – Medições do ponto 94 com Face Mesh e movimento 5 cm

Leituras	Medição pto 1		Distância	Mult.936	Graus
	X	Y			
1	0,500314	0,508291	0,005968	5,586133	3,987828
	0,494409	0,507426			
2	0,500155	0,511279	0,005639	5,277657	3,768922
	0,494518	0,511145			
3	0,500939	0,515606	0,006319	5,914542	4,220623
	0,494621	0,515680			
4	0,500489	0,520082	0,006206	5,808519	4,145498
	0,494286	0,520274			
5	0,500321	0,524843	0,006035	5,648915	4,032353
	0,494287	0,524972			
6	0,499984	0,528570	0,005948	5,567464	3,974586
	0,494045	0,528911			
7	0,500760	0,531504	0,005756	5,387187	3,846675
	0,495012	0,531809			
8	0,501295	0,536272	0,006069	5,680841	4,054991
	0,495235	0,535929			
9	0,501223	0,540703	0,005721	5,354496	3,823471
	0,495513	0,541059			
10	0,501138	0,545568	0,005477	5,126635	3,661669
	0,495686	0,546097			
Valor da média			5,535239		
Desvio padrão			0,233280		

Fonte: Elaborado pelo autor

Figura 64 – Comparação entre as medições realizadas com Face Mesh e a câmera Notebook utilizando capacete e papel milimetrado, com distância de 5cm



Fonte: Elaborado pelo autor

4.4.2 Medições com câmera de notebook - Teste 02

Nesta seção realizou-se novo teste com outro modelo de notebook, o Aspire 3 AMD Ryzen 5 3500U, com os valores de deslocamento alterado para 10cm. Repetindo o procedimento anterior, utilizou-se o capacete com laser e a folha de papel milimetrado e realizaram-se novas medições,

agora com deslocamento de 10 cm do laser na folha de papel milimetrado.

A Tabela 37 apresenta os valores de distância, média, desvio padrão e o ângulo em graus obtidos, os quais atendem à proposta do trabalho.

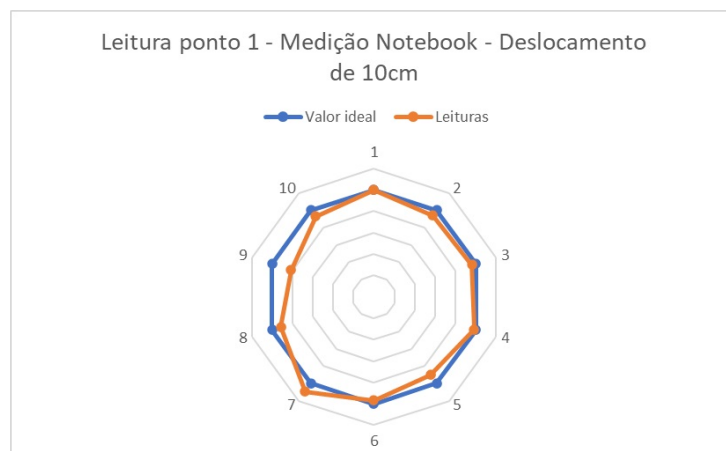
Tabela 37 – Medições do ponto 94 com Face Mesh e movimento 10 cm

Leituras	Medição pto 1		Distância	Mult.1000	Graus
	X	Y			
1	0,484127	0,621291	0,010647	9,965156	7,064262
	0,473581	0,622751			
2	0,483596	0,622044	0,010010	9,369552	6,649871
	0,473763	0,623920			
3	0,484079	0,622711	0,010290	9,631341	6,832191
	0,473909	0,624280			
4	0,483955	0,623615	0,010541	9,866312	6,995594
	0,473716	0,626122			
5	0,484136	0,625735	0,009637	9,020073	6,406053
	0,474557	0,626786			
6	0,484902	0,625467	0,010337	9,675769	6,863104
	0,474677	0,626989			
7	0,485611	0,626299	0,011697	10,948700	7,745224
	0,473956	0,627301			
8	0,485408	0,627149	0,009779	9,152800	6,498708
	0,475697	0,628293			
9	0,485476	0,627931	0,008693	8,136462	5,787517
	0,476812	0,628634			
10	0,486851	0,627868	0,009898	9,264878	6,576895
	0,477103	0,629589			
Valor da média			9,503104		
Desvio padrão			0,691357		

Fonte: Elaborado pelo autor

Na Figura 65, observa-se que os valores obtidos ainda não são precisos, mas atendem aos objetivos buscados. Novamente a cor azul indica a condição ideal para o deslocamento de 10 cm e a laranja a resposta do sistema. Outros testes foram realizados com novos modelos de câmeras e distâncias.

Figura 65 – Comparação entre as medições realizadas com Face Mesh e a câmera Notebook utilizando capacete e papel milimetrado, com distância de 10cm

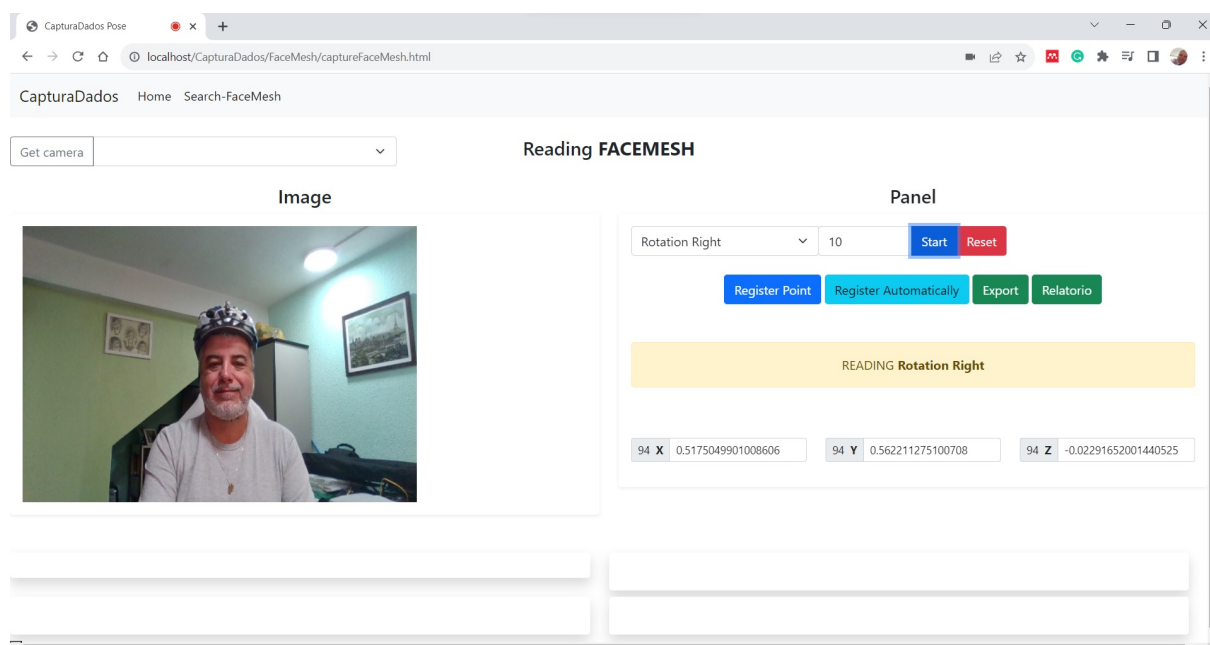


Fonte: Elaborado pelo autor

4.4.3 Medições com a câmera Intel RealSense D435i

Utilizando o mesmo procedimento anterior, foram realizadas medições com a câmera Intel RealSense D435i no ponto 94, sendo realizadas 20 leituras por quadro e apresentados o valor médio e o desvio padrão. A câmera foi montada em um tripé e estava posicionada a 80 cm de distância do nariz do indivíduo, com as medições sendo realizadas com a câmera Intel RealSense e com o capacete com laser acoplado conforme indicado na Figura 66. A folha de papel milimetrado estava marcada com pontos de distâncias controladas. O objetivo foi verificar os dados que seriam obtidos com outros modelos de câmera e com a movimentação controlada.

Figura 66 – Tela do sistema de medições com a câmera Intel D435i RealSense utilizando capacete e papel milimetrado



Fonte: Elaborado pelo autor

O procedimento descrito acima foi repetido 20 vezes para que as distâncias fossem capturadas para serem comparadas. A Tabela 38 apresenta os valores obtidos. Ao analisar esta tabela, observa-se que os valores de distância obtidos, próximos de 8 cm, estão bem diferentes dos de movimentação do laser, pois o ideal seria obter valores próximos de 5cm.

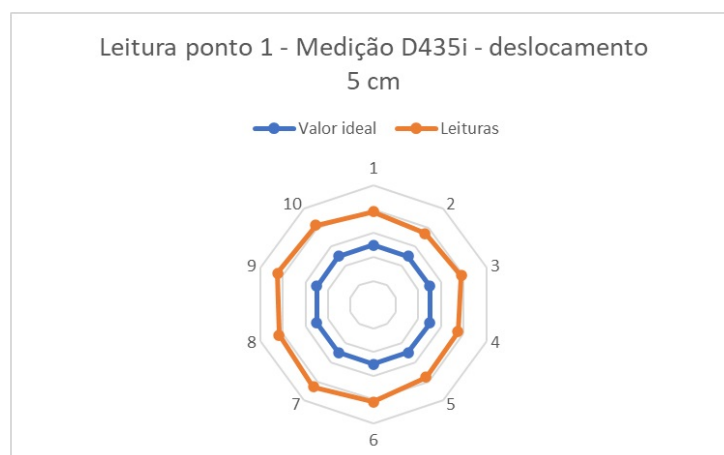
Tabela 38 – Medições de mesmo ponto com Face Mesh e a câmera Intel D435i RealSense utilizando capacete e papel milimetrado

Leituras	Medição pto 1		Distância	Mult.936	Graus
	X	Y			
1	0,688791	0,417439	0,008342	7,808415	5,557166
	0,680765	0,415166			
2	0,687474	0,418553	0,007901	7,394978	5,266336
	0,679592	0,419097			
3	0,688348	0,422470	0,008345	7,810565	5,558677
	0,680021	0,421932			
4	0,690233	0,427030	0,007985	7,473830	5,321847
	0,682334	0,425861			
5	0,689763	0,430327	0,008101	7,582749	5,398492
	0,681736	0,429230			
6	0,689633	0,432694	0,008744	8,184841	5,821457
	0,680888	0,432625			
7	0,688964	0,435944	0,009151	8,565042	6,087888
	0,679875	0,434881			
8	0,688125	0,438484	0,008902	8,332456	5,924962
	0,679272	0,437550			
9	0,687711	0,442127	0,009043	8,464522	6,017499
	0,678903	0,444174			
10	0,687106	0,448606	0,008803	8,239586	4,549567
	0,680303	0,443019			
Valor da média			7,985699		
Desvio padrão			0,403075		

Fonte: Elaborado pelo autor

Quando se calcula o desvio padrão e a repetibilidade dos valores, verifica-se que eles são aceitáveis. Na Figura 67, observam-se os valores desejados e os obtidos para comparação. No próximo experimento, uma análise dos valores obtidos pela RealSense 455 vai permitir verificar se esses valores estão relacionados a esses modelos de câmeras.

Figura 67 – Comparação gráfica de medições do deslocamento com câmera D435i RealSense calculado

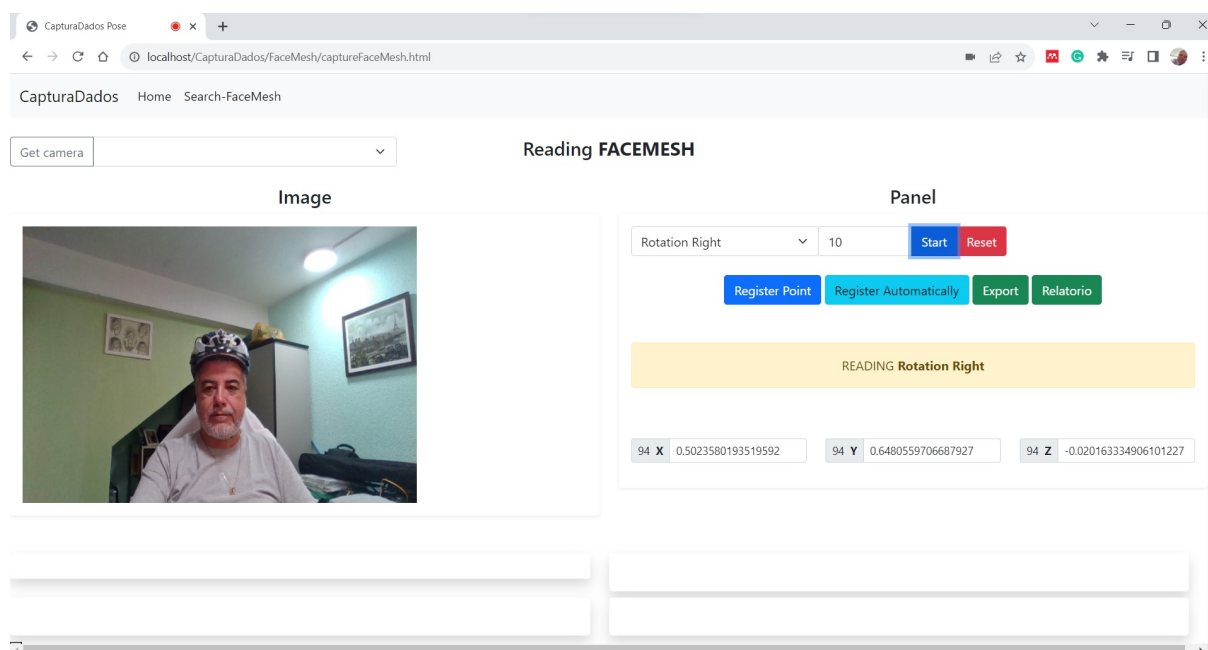


Fonte: Elaborado pelo autor

4.4.4 Medições com a câmera Intel RealSense D455

Novamente, foram realizadas medições do ponto 94, agora com a câmera Intel RealSense D455 e com o capacete com laser acoplado. Foram realizadas 20 leituras por quadro, sendo apresentado somente o valor médio. O objetivo foi o mesmo do experimento anterior: verificar o comportamento da câmera com a movimentação controlada. Na montagem, a câmera estava posicionada em um tripé a 80 cm de distância do nariz. A Figura 68 apresenta uma imagem da ferramenta sendo usada para o teste.

Figura 68 – Tela do sistema de medições com a câmera Intel D455 RealSense utilizando capacete e papel milimetrado



Fonte: Elaborado pelo autor

O mesmo procedimento de deslocamentos de 5 cm, com o capacete com laser direcionado a posições específicas na folha de papel milimetrado e com pontos e distâncias controladas foi utilizado. Os dados obtidos com as medições são apresentados na Tabela 39. Sua análise permite observar que os valores de distância obtidos estão bem próximos dos números observados com as medições feitas com capacete e o laser.

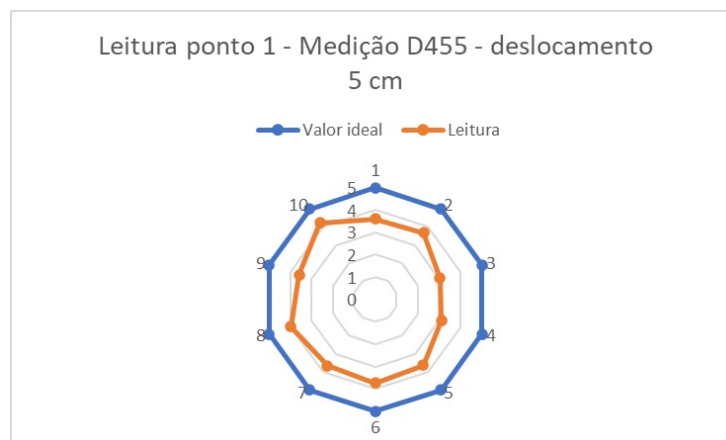
Tabela 39 – Medições de mesmo ponto com Face Mesh e a câmera Intel D455 RealSense utilizando capacete e papel milimetrado

Leituras	Medição pto 1		Distância	Mult.1000	Graus
	X	Y			
1	0,592112	0,434424	0,003849	3,602747	2,062438
	0,588273	0,434148			
2	0,591881	0,435631	0,003923	3,671707	2,101845
	0,588036	0,436408			
3	0,591698	0,438028	0,003246	3,037881	1,739508
	0,588506	0,438613			
4	0,591687	0,440288	0,003330	3,116679	1,784570
	0,588412	0,440885			
5	0,592543	0,441940	0,003894	3,644880	2,086515
	0,588728	0,442721			
6	0,592222	0,444061	0,004002	3,745722	2,144136
	0,588245	0,444507			
7	0,592441	0,445545	0,003905	3,655310	2,092475
	0,588549	0,445867			
8	0,592896	0,446827	0,004248	3,976484	2,275959
	0,588695	0,447461			
9	0,595123	0,448437	0,003823	3,578495	2,582581
	0,591305	0,448627			
10	0,594129	0,449926	0,004528	4,237867	2,425217
	0,589660	0,450658			
Valor da média			3,626777		
Desvio padrão			0,3351103		

Fonte: Elaborado pelo autor

No gráfico Figura 69, pode-se observar que os valores de distância entre as leituras apresentam um valor de desvio padrão aceitável. O problema é a indicação de distância entre os modelos de sensores de movimento. Para se buscar a melhor solução técnica, na seção 4.5.1 são apresentadas análises de comparações entre os sensores de movimento. A partir dos dados obtidos com os diferentes testes realizados e apresentados neste capítulo, apresentamos uma teoria sobre os valores obtidos.

Figura 69 – Comparação gráfica de medições do deslocamento com câmera D455 RealSense calculado



Fonte: Elaborado pelo autor

4.5 ANÁLISE E COMPARAÇÃO ENTRE OS MÉTODOS E RESULTADOS OBTIDOS NOS EXPERIMENTOS

Nesta seção se realiza uma comparação entre os métodos e resultados obtidos após a realização de vários experimentos. Conforme se mencionou, os testes foram divididos em estáticos e dinâmicos com o objetivo de definir mudanças que deveriam ser realizadas no sistema para atender a proposta.

As informações obtidas com os testes permitiram definir como deve ser o software e que características o sistema deve oferecer para facilitar a interação com o usuário e agilizar o trabalho do especialista na identificação do quanto o indivíduo tem seu senso de posicionamento articular cervical alterado em determinados grupos. A seguir, análises específicas são apresentadas considerando os diferentes tipos de testes que foram realizados de forma estática e dinâmica.

4.5.1 Etapa Estática

Na etapa estática definiu-se a forma como os dados lidos pelo sistema deveriam ser tratados e como os experimentos auxiliaram na evolução do sistema.

A análise dos dados obtidos sem a movimentação da face permitiu determinar a repetibilidade e a reprodutibilidade dos valores de leituras com o sistema utilizando o MediaPipe Pose. Com os testes realizados nesta etapa, foi possível se definir a forma de cálculo que deveria ser utilizada pelo sistema. Foi possível também confirmar que a distância ideal da câmera até o nariz do indivíduo deveria ser de 80 cm. Definiu-se também a quantidade de leituras que deveriam ser realizadas a cada quadro para aumentar a precisão dos dados lidos. Além disso, foi definido o formato dos valores que deveriam se apresentados em todas as tabelas.

A partir da realização dos testes se observou a necessidade de obter os valores de distância em centímetros, média e desvio padrão, para facilitar uma comparação entre os diferentes modos

de medição. O valor do fator utilizado para determinar o deslocamento em centímetros foi 1000. Esse fator foi utilizado em todos os experimentos realizado em que a indicação de distância foi utilizada.

Os testes iniciais com a realização das leituras de somente um quadro por segundo, não apresentaram uma repetibilidade aceitável dos valores obtidos. Realizou-se então alterações no software para que fosse efetuada mais de uma leitura por quadro, sendo possível serem lidos mais quadros por segundo. Mesmo assim, não foram obtidos valores aceitáveis de medição. Para as leituras realizadas durante esse processo, a fim de facilitar a análise dos dados e repetição de testes, o valor de média adotado foi de 20 leituras.

Ainda buscando um procedimento que oferecesse melhores resultados com o MediaPipe Pose, utilizou-se outro método no qual se utilizava a posição do nariz e a posição dos ombros direito e esquerdo para efetuar os cálculos da distância entre os ombros e o nariz para determinar sua posição. Esses testes também não apresentaram resultados adequados pois não houve uma boa repetibilidade dos valores obtidos.

Dessa forma, optou-se por alterar o modo de leitura com o MediaPipe, deixando-se de usar o MediaPipe Pose e passando-se a utilizar o MediaPipe Face Mesh. Nesta nova etapa, inicialmente foram realizadas alterações no software para que fosse possível fazer essas leituras e, já nas primeiras medições, foram obtidos valores mais coerentes e que atendiam melhor ao objetivo buscado, que é uma melhor repetibilidade e reprodutibilidade das medições. A proposta, a partir desse ponto, foi determinar se com câmeras de maior qualidade seria obtida uma melhor precisão, o que, logo nas primeiras medições, foi descartado.

Na continuação dos testes, como se observou pequenas mudanças nos valores obtidos mesmo com repetição de processo, optou-se por realizar experimentos que permitissem alterar as cores de fundo da tela de captação de dados. Nesses testes foi utilizado um boneco para evitar movimentações involuntárias. Observou-se que com fundos de cores diferentes se obtinha diferenças nos valores lidos, mas que essas não alteravam significativamente a repetibilidade e reprodutibilidade dos dados lidos. Entretanto, como salientado no experimento, é importante procurar manter uma cor fixa e, se possível, mais escura do fundo da tela para uma melhor precisão das leituras.

4.5.2 Etapa Dinâmica

Nesta etapa, foi possível determinar que o sistema permite realizar medições com a face estática antes e depois das movimentações da face. Realizaram-se testes controlados de distâncias com movimento da cabeça para esquerda e direita e as leituras tiveram uma boa precisão.

A distância de 80 cm entre a câmera e o nariz do indivíduo foi mantida e, com a utilização de um capacete com laser acoplado, deu-se início às medições com distâncias de movimentação controladas, conforme se apresentou na seção 4.4.

Foram realizados experimentos com movimentações de 5 cm e de 10 cm com diferentes equipamentos e foram apresentados os valores obtidos para a média, e desvio padrão, e a

partir destes testes se passou a calcular quantos graus de deslocamento ocorreriam entre as movimentações.

Pôde-se observar que em todos os experimentos, os valores de leitura atendiam aos objetivo buscado, mesmo que as câmeras do modelo RealSense apresentassem valores um pouco diferentes dos ideais. Realizaram-se cálculos de posicionamento da face, sendo indicados em centímetros para facilitar uma comparação com o deslocamento do laser no uso do capacete. O fator utilizado para indicação em centímetros foi mantido em 1000. Desta forma, visualizou-se uma diferença entre os valores obtidos, que poderia ser corrigida.

Adotar um fator fixo para as medições permitiu identificar as diferenças entre os modelos de câmeras e observar que existe influência das lentes nas medições. Esta observação permite afirmar que é necessário um estudo mais detalhado dos modelos de lentes utilizadas nas câmeras. Essa diferença que poderia ser observada com as lentes foi que descartou a possibilidade de utilizar câmeras de celular para captar os pontos.

As seções 4.4.3 e 4.4.4, apresentaram as medições realizadas com os sensores de movimento da RealSense.

Nelas se identificou a necessidade de utilizar um fator diferente para as câmeras do modelo RealSense pois essas câmeras possuem características específicas de lente que obrigam a realização de um estudo suplementar sobre sua aplicação. A partir desses estudo, se descartou neste projeto de pesquisa a utilização destes modelos de sensores de movimento. De certa forma esta análise levou ao abandono da ideia original que passava pela utilização de câmeras como sensores de movimento.

Os experimentos e os estudos realizados possibilitaram afirmar que as câmeras utilizadas nos notebooks não sofreram grandes alterações nos últimos anos. Mesmo com a evolução dos equipamentos, esses dispositivos apresentam características similares. Dessa forma, definiu-se a câmera do notebook como a principal ferramenta nos testes.

5 CONSIDERAÇÕES FINAIS

O objetivo buscado com a realização deste trabalho foi definido no início deste documento como sendo "[...] o desenvolvimento de uma ferramenta de baixo custo que facilite a captação e análise dos dados obtidos durante a realização de um teste de realocação cervicocefálica, para auxiliar fisioterapeutas no tratamento de pacientes com disfunção proprioceptiva cervical que levam ao aparecimento de dores cervicais crônicas".

A atividades de pesquisa desenvolvidas e apresentadas neste documento mostram como se buscou atingir este objetivo. O texto indica como se procurou criar uma ferramenta para facilitar o trabalho do fisioterapeuta com pessoas que sentem dor cervical crônica devido a problemas com disfunção proprioceptiva cervical, facilitar a análise dos dados e o local onde a captação dos dados é realizada.

Ao longo do documento explicou-se os procedimentos realizados no desenvolvimento do trabalho proposto e onde foi realizada a pesquisa. Um desenvolvimento em conjunto com o laboratório G-SCOP, permitiu uma interação com estudantes que utilizavam sensores de movimento em suas pesquisas. Em reuniões com fisioterapeutas que possuem parceria com o G-SCOP, foi possível definir a problemática e os objetivos.

Na contextualização do trabalho, determinou-se qual a ferramenta de visualização digital deveria ser utilizada, tendo sido essa uma das primeiras etapas implementadas. Desta forma, realizou-se uma pesquisa entre os diversos modelos de sensores de movimento na busca da melhor solução técnica para se determinar qual ferramenta deveria ser utilizada para leitura desses dados. Essa pesquisa envolveu as aplicações de sensores de movimentos na utilização de interatividade com indivíduos, ou seja em aplicações de realidade virtual e realidade aumentada. As pesquisas realizadas permitiram também que se verificasse que é possível utilizar esses sensores em aplicações que interagem com o corpo humano.

Após uma pesquisa sobre formas de tratamento da imagem na identificação do posicionamento da face, encontrou-se o MediaPipe, ferramenta disponibilizada em 2019 de forma gratuita pela Google. No site da empresa, apresentam-se várias sugestões de aplicações desta ferramenta, e percebeu-se que ela atendia perfeitamente às necessidades da pesquisa por apresentar tempo reduzido de análise dos dados e uma forma simples de identificação do posicionamento da face.

A metodologia experimental e a tecnologia envolvida permitiu definir a arquitetura do sistema e como deveriam ser organizado os experimentos de forma a captar os dados de posicionamento da face; permitiu também definir como devem ser apresentados pelo sistema.

Os testes e a análise dos resultados realizados na etapa de experimentos estáticos e dinâmicos, permitiu concluir que é possível utilizar câmeras como uma ferramenta para determinar o posicionamento da face do indivíduo. Ao analisarem-se os dados obtidos nos experimentos dinâmicos, pode-se afirmar que os modelos analisados podem ser utilizados. Além disso, como um dos objetivos do trabalho era facilitar o uso do software, sugere-se que sua utilização seja

realizadas com câmeras de notebooks que se mostraram apropriadas.

As câmeras da RealSense também atendem ao objetivo do trabalho, mas seu uso foi descartado, pois apresentavam custo elevado e, em agosto de 2021, encontravam-se em processo de finalização da produção pelo fabricante. A partir de 2024 o fabricante resolveu retornar ao mercado, e atualmente as câmeras Realsense disponíveis são as D455, D457 e D435f. A diferença entre esses modelos estão associadas a classes de proteção, redução de brilho e profundidade melhorada para utilização em ambientes internos e externos. Aparentemente elas podem atender às necessidades da aplicação.

Baseado na pesquisa realizada e no breve relato apresentado acima e nos resultados obtidos, pode-se afirmar que os objetivos delineados no início deste trabalho foram alcançados com sucesso. O desenvolvimento da ferramenta de baixo custo para captação e análise de dados durante o teste de realocação cervicocefálica provou-se eficaz como ferramenta que pode ser utilizado na assistência a fisioterapeutas no tratamento de pacientes com disfunção proprioceptiva cervical, responsável por dores cervicais crônicas. A colaboração com o laboratório G-SCOP e a utilização de sensores de movimento permitiram criar uma solução técnica robusta, enquanto a implementação da solução técnica com o MediaPipe da Google para a identificação do posicionamento da face demonstrou ser uma escolha apropriada, devido à sua precisão e rapidez. A pesquisa experimental validou o uso de câmeras de notebooks como uma ferramenta prática e acessível, confirmando que os modelos analisados podem ser utilizados de forma eficaz. Além disso, apesar do potencial das câmeras RealSense, a decisão de priorizar soluções mais econômicas e amplamente disponíveis reforça o objetivo de criar uma ferramenta acessível. Em síntese, a metodologia empregada, aliada às tecnologias selecionadas, possibilitou a criação de um sistema eficiente que atende às necessidades dos profissionais de saúde e facilita o tratamento de disfunções cervicais.

Para finalizar esta análise, é fundamental afirmar que este trabalho de tese permitiu validar as hipóteses propostas no início do documento. A seguir, cada uma delas é retomada e para cada uma delas se apresenta como os estudos realizados asseguram que elas foram respondidas de forma afirmativa.

- *H1 - Os dados obtidos com um sensor de movimento podem ser utilizados para avaliar a distância entre dois pontos específico da face de um indivíduo, em quadros capturados no instante inicial e final do movimento?*

Conforme apresentado no Capítulo 2, seção 2.3.7 se realizou uma análise das aplicações dos diferentes sensores de movimento. A seção 2.4, apresentou uma síntese do Capítulo que permitiu afirmar que os dispositivos permitem identificar pontos em um quadro e atendem a proposta do trabalho.

- *H2 - É possível utilizar uma câmera convencional como um sensor de movimento na aplicação da técnica TRC?*

A partir da apresentação realizada no Capítulo 3, seção 3.2, onde foi introduzida a ferramenta MediaPipe e sua aplicação na identificação de um ponto da face, pode-se afirmar que é possível utilizar uma câmera convencional como um sensor de movimento na aplicação da técnica TRC com o sistema desenvolvido.

- *H3 - A captura e o tratamento dos dados coletados com um sensor de movimento ou câmera apresentam uma boa repetibilidade e reprodutibilidade?*

Conforme apresentado no Capítulo 4, foi necessário realizar vários experimentos (apresentados nas seções 4.2 e 4.3) com a utilização de câmeras e sensores de movimento para confirmar a hipótese de se obter uma boa repetibilidade dos dados. Os testes realizados na seção 4.4 indicaram ser possível também obter uma boa reprodutibilidade dos valores capturados.

- *H4 - É possível propor uma nova técnica de pós-tratamento de dados para responder à problemática de um profissional da saúde na técnica TRC?*

Conforme apresentado no Capítulo 4, seção 4.5, onde foi realizado uma análise dos resultados obtidos, pode-se afirmar que o sistema pode ser utilizado no procedimento de análise da técnica TRC pelo profissional da saúde

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Ao final do trabalho se observa que inúmeras outras possibilidades podem ser exploradas a partir da conclusão do que, pode-se entender, trata-se de uma etapa inicial de uma pesquisa que pode dar inúmeros outros frutos.

Talvez o primeiro passo para iniciar uma segunda fase de pesquisa passe pela definição de um protocolo que permita a avaliação da ferramenta desenvolvida com pacientes reais, já que todos os experimentos realizados foram feitos em laboratório. A realização de testes clínicos é de suma importância no processo de validação de uma ferramenta desenvolvida para avaliar pacientes com dor cervical crônica. Estes testes não só fornecem dados empíricos essenciais para comprovar a eficácia e a segurança da ferramenta, como também garantem que a mesma atenda aos padrões clínicos e regulamentares exigidos.

Primeiramente, testes clínicos permitem a avaliação objetiva da precisão e confiabilidade da ferramenta em condições reais de uso. Pacientes com dor cervical crônica apresentam uma diversidade de sintomas e variações individuais que são difíceis de reproduzir em ambientes controlados de laboratório. Portanto, a aplicação da ferramenta em um cenário clínico oferece uma compreensão mais abrangente de seu desempenho e limitações.

Além disso, os testes clínicos são fundamentais para identificar possíveis efeitos adversos ou imprevistos. Mesmo uma ferramenta que funcione perfeitamente em um ambiente experimental pode apresentar desafios quando aplicada a uma população mais ampla. A identificação precoce

de tais questões permite ajustes e melhorias no design da ferramenta, aumentando sua segurança e eficácia.

A participação de profissionais de saúde nos testes clínicos é igualmente crucial. Fisioterapeutas e outros especialistas podem fornecer insights valiosos sobre a usabilidade da ferramenta, bem como sugestões para otimização de suas funcionalidades. A colaboração com esses profissionais também facilita a integração da ferramenta na prática clínica diária, aumentando sua aceitação e uso generalizado.

Finalmente, a validação clínica através de testes rigorosos contribui significativamente para a credibilidade científica da ferramenta. Publicações de resultados positivos em revistas científicas renomadas e apresentações em conferências especializadas ajudam a disseminar o conhecimento sobre a nova tecnologia, promovendo sua adoção em uma escala mais ampla.

Em resumo, a continuidade deste trabalho com a realização de testes clínicos, poderia se constituir em um passo indispensável para um desenvolvimento e implementação de uma ferramenta comercial para avaliar pacientes com dor cervical crônica. Estes testes garantiriam que a ferramenta não apenas funcione como esperado, mas também que atenda aos mais altos padrões de segurança, eficácia e usabilidade, beneficiando tanto os pacientes quanto os profissionais de saúde.

É importante também explorar outra via para a continuidade da pesquisa com a utilização do Google MediaPipe. Uma sugestão de alteração e melhoria passa por sua utilização para capturar pontos quando indivíduos realizam movimentos de extensão e flexão da cabeça. Neste caso o sistema poderia ser utilizado como uma ferramenta na análise da medição de ângulos articulares nas articulações humanas. Pesquisas deste topo poderiam facilitar a criação de novos sistemas que auxiliem nessas análises pelo fisioterapeuta.

Outro ponto importante a ser destacado é sobre a utilização dessa ferramenta no auxílio de pessoas que necessitam realizar fisioterapia após cirurgias de mãos e braços. Alterando-se o sistema para determinar os ângulos articulares, pode-se permitir a realização de acompanhamento de especialistas em telemedicina, sendo possível utilizar tais recursos na identificação da presença ou não de disfunções e quantificação das limitações de ângulos articulares. Uma proposta a ser também implementada é a utilização dos valores de distância indicados pelo MediaPipe para determinar a posição do indivíduo.

A pesquisa não se aprofundou na utilização de câmeras de celulares e nem na criação de um aplicativo que pudesse corrigir os erros de leituras com diferentes lentes. Novos desenvolvimentos e pesquisas poderiam ser realizadas para dar respostas a essas questões.

Uma última sugestão de desenvolvimento futuro toma por base os testes dinâmicos já apresentados que permitiram propor resultados promissores aos fisioterapeutas. Com novos testes, é possível, por exemplo, que o sistema seja disponibilizado inclusive para alunos, deixando-lhes oportunidade de utilizar a ferramenta e as tecnologias envolvidas e os parâmetros de análise de pessoas com dor cervical crônica.

REFERÊNCIAS

- ABADI, M. *et al.* TensorFlow: A system for Large-Scale machine learning. In: SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION. **12th USENIX Symposium on operating systems design and implementation**. Savannah, GA: USENIX Association, 2016. p. 265–283.
- AGARWAL, A. *et al.* **TensorFlow**: Large-scale machine learning on heterogeneous systems. Tensorflow, 2015. Disponível em: <https://www.tensorflow.org/>. Acesso em: 10 fev. 2023.
- ALVARENGA, M. L. T.; CORREA, D. O.; OSÓRIO, F. S. Redes neurais artificiais aplicadas no reconhecimento de gestos usando o kinect. In: COMPUTER ON THE BEACH, Florianópolis. **Anais [...]**. Itajai: UNIVALI, 2012.
- AMBETH-KUMAR, V. D. *et al.* Facial recognition system for suspect identification using a surveillance camera. Software and sware Pattern Recognition and Image Analysis, New York, v. 28, p. 410 – 420, 07 2018.
- ASUSTEK, C. I. Use **Xtion PRO developer solution to make motion-sensing applications and games**. [S.l.] Asus, 2019. Disponível em: https://www.asus.com/3D-Sensor/Xtion_PRO/. Acesso em: 12 set. 2022.
- AVOLA, D. *et al.* Mobile applications for automatic object recognition. In: KHOUSROW POUR, M. **Advanced methodologies and technologies in network architecture, Mobile Computing, and Data Analytics**. Chicago, 2019. p. 1008–1020. ISBN 9781522575993.
- BASHEER, I. A.; HAJMEER, M. Artificial neural networks: fundamentals, computing, design, and application. **Journal of microbiological methods**, 2000. Disponível em: [https://doi.org/10.1016/S0167-7012\(00\)00201-3](https://doi.org/10.1016/S0167-7012(00)00201-3). Acesso em: 10 mar. 2021.
- BAZAREVSKY, V. *et al.* Blazepose: On-device real-time body pose tracking. **Computer vision and pattern recognition**, 2020. Disponível em: <https://arxiv.org/abs/2006.10204>. Acesso em: 10 ago. 2021.
- BRITO, E. N. D. *et al.* Artificial intelligence in the diagnosis of neurodegenerative diseases: a systematic literature review. **Research, Society and Development**, v. 10, n. 11, p. e482101120004, Sep. 2021. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/20004>. Acesso em: 10 mar. 2023.
- CAMPOS, M. S. **Avaliação de Sistemas de Medição**: Estudos de repetibilidade e reprodutibilidade. [Ouro Preto]: UFOP, 2019. 18 slides. SlideShare. Disponível em: http://professor.ufop.br/sites/default/files/magno/files/capitulo_5_-_rr.pdf. Acesso em: 2023-09-01.
- CASTRO, M. C. L. *et al.* Interaction by hand-tracking in virtual reality. **Sociedade brasileira de computação** — BCOPENLIB Anais do simposio de realidade virtual e aumentada (SVR), Porto Alegre, RS, p. 44–48, 11 nov. 2020.
- CLARK, R. A. *et al.* Three-dimensional cameras and skeleton pose tracking for physical function assessment: A review of uses, validity, current developments and kinect alternatives. **National library of medicine** — Gait & Posture, Bethesda, p. 193 – 200, 22 nov. 2019.

CONTROLLER, M. **Motion controller**. São Francisco, CA: Wikimedia, 2020. Disponível em: https://en.wikipedia.org/wiki/Motion_controller. Acesso em: 10 mar. 2020.

CORDO, P. *et al.* Proprioceptive consequences of tendon vibration during movement. **JPN Journal of neurophysiology**, Bethesda, 1995. Disponível em: <https://doi.org/10.1152/jn.1995.74.4.1675>. Acesso em: 20 mar. 2021.

CORPORATION, M. **Programming with the Kinect for Windows SDK**. EUA: Microsoft Research, 2011. Disponível em: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/en-us-events-fs2011-jancke_kinect_programming.pdf. Acesso em: 10 abr. 2020.

CORPORATION, M. **About Kinect for Windows SDK NUI API**. EUA: Microsoft Ignite, 2014. Disponível em: [https://docs.microsoft.com/pt-br/previous-versions/windows/kinect/dn782041\(v=ieb.10\)](https://docs.microsoft.com/pt-br/previous-versions/windows/kinect/dn782041(v=ieb.10)). Acesso em: 25 maio 2021.

CORPORATION, M. **Kinect for Windows SDK**. EUA: Microsoft Research, 2014. Disponível em: <https://www.microsoft.com/en-us/download/details.aspx?id=44561>. Acesso em: 12 set. 2020.

CORPORATION, M. **Azure Kinect DK Documentation**. EUA: Microsoft Ignite, 2024. Disponível em: <https://docs.microsoft.com/pt-br/azure/kinect-dk/>. Acesso em: 01 nov. 2020.

DRANCA, L. *et al.* Using kinect to classify parkinson's disease stages related to severity of gait impairment. **BMC part of springer nature** — Bioinformatics, EUA, n. 19, 2018. Acesso em: 10 mar. 2021.

DROR, B. *et al.* Automatic assessment of parkinson's disease from natural hands movements using 3d depth senso. **IEEE xplore** — IEEE 28th Convention of Electrical Electronics Engineers in Israel (IEEEI), Eilat, Israel, p. 1–5, 2014. Acesso em: 10 mar. 2022.

ESTEREOSCOPIA. **Estereoscopia, dois pontos de visão**. São Francisco, CA: Wikimedia, 2022. Disponível em: <https://pt.wikipedia.org/wiki/Estereoscopia>. Acesso em: 15 abr. 2022.

FARHADI-NIAKI, F.; GHASEMAGHAEI, R.; ARYA, A. Empirical study of a vision-based depth-sensitive human-computer interaction system. In: ACM DIGITAL LIBRARY. **APCHI '12: Proceedings of the 10th asia pacific conference on Computer human interaction**. Matsue-city Shimane Japan: ACM - Association for Computing Machinery, 2012. p. 101–108. Acesso em: 01 set. 2023.

FRANÇA, J. A. **Desenvolvimento de Algoritmos de Visão Estereoscópica para Aplicações em Robótica Móvel**. 2004. 103 p. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis, S.C, 2004.

GALNA, B. *et al.* Accuracy of the microsoft kinect sensor for measuring movement in people with parkinson's disease. **Gait & Posture**, ScienceDirect, Netherlands, v. 39, n. 4, p. 1062 – 1068, 2014. ISSN 0966-6362. Acesso em: 10 mar. 2021.

GEERSE, D. J.; COOLEN, B. H.; ROERDINK, M. Kinematic validation of a multi-kinect v2 instrumented 10-meter walkway for quantitative gait assessments. **Plos one**, São Francisco, California, v. 10, n. 10, p. 1 – 15, 10 2015. Disponível em: <https://doi.org/10.1371/journal.pone.0139913>. Acesso em: 20 mar. 2022.

GHOLAMI, F. *et al.* A microsoft kinect-based point-of-care gait assessment framework for multiple sclerosis patients. **IEEE Journal of biomedical and health informatics**, New York, v. 21, p. 1–1, 01 2016. Acesso em: 10 mar. 2021.

HANKE, S. *et al.* **The Technical Specification and Architecture of a Virtual Support Partner**. Athens, 2015. (CEUR-WS, v. 1528). Disponível em: <http://ceur-ws.org/Vol-1528/paper4.pdf>. Acesso em: 15 maio 2021.

HARTSON, H. R.; PYLA, P. S. Process and guidelines for ensuring a quality user experience. In: SCIENCE DIRECT. **The UX Book**. 12. ed. Waltham, MA: Imprint Morgan Kaufmann, 2012. p. 299 – 332. Acesso em: 20 set. 2020.

HEIKKILA, H.; ASTROM, P. Cervicocephalic kinesthetic sensibility in patients with whiplash injury. **National library of medicine — Scandinavian journal of rehabilitation medicine**, Bethesda, p. 133–138, set. 1996. Acesso em: 20 mar. 2021.

HOLISTIC, G. M. **MediaPipe Holistic**. Mountain View: Google Inc., 2022. Disponível em: <https://ai.googleblog.com/2020/12/mediapipe-holistic-simultaneous-face.html>. Acesso em: 23 jul. 2022.

INTEL, C. **Tecnologia Intel RealSense**. Santa Clara: [s.n.], 2019. Disponível em: <https://www.intel.com.br/content/www/br/pt/architecture-and-technology/realsense-overview.html>. Acesso em: 15 set. 2021.

INTEL, C. **Intel Realsense Product Family D400 Serie**. Santa Clara: [s.n.], 2020. Disponível em: <https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf>. Acesso em: 10 jul. 2021.

INTEL CORPORATION. **Skeletal Tracking Overview**. Santa Clara: INTEL, 2022. Disponível em: <https://www.intelrealsense.com/skeletal-tracking/>. Acesso em: 10 out. 2022.

IRIS. **Iris recognition**. São Francisco, CA: Wikimedia, 2021. Disponível em: https://en.wikipedia.org/wiki/Iris_recognition. Acesso em: 15 abr. 2021.

IRIS, G. M. **MediaPipe Iris**. Mountain View: Google Inc., 2022. Disponível em: <https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/iris.md>. Acesso em: 23 jul. 2022.

JAVASCRIPT. **Semana do livro e das bibliotecas 2022**. São Francisco, CA: wikipedia, 2022. Disponível em: [https://pt.wikipedia.org/wiki/Java_\(linguagem_de_programacao\)](https://pt.wikipedia.org/wiki/Java_(linguagem_de_programacao)). Acesso em: 10 abr. 2023.

KANG, J. W.; SEO, D. S.; JUNG, D. S. A study on the control method of 3-dimensional space applications using kinect system. **IJCSNS International Journal of Computer Science and Network Security**, United Kingdom, v. 11, p. 55 – 59, set. 2011.

KEPSKI, M.; KWOLEK, B. Human fall detection using kinect sensor. In: PROCEEDINGS OF THE 8TH INTERNATIONAL CONFERENCE ON COMPUTER RECOGNITION SYSTEMS CORES. **Springer international publishing switzerland**. Switzerland: Springer Nature, 2013. v. 226, p. 743–752.

KRAMER, K. L. Approaches to a sustainable user experience. In: **SCIENCE DIRECT JOURNAL AND BOOKS. User experience in the age of sustainability**. 12. ed. Bethesda: Imprint Morgan Kaufmann, 2012. p. 31–68. Disponível em: <https://www.sciencedirect.com/topics/computer-science/natural-user-interface>. Acesso em: 15 mar. 2021.

KRISTJANSSON, E.; TRELEAVEN, J. Sensorimotor function and dizziness in neck pain: Implications for assessment and management. In: **MOVEMENT SCIENCE MEDIA JOSPT. Journal of orthopaedic and sports physical therapy**. 5. ed. New York: Imprint Morgan Kaufmann, 2009. v. 39, p. 364–377. Disponível em: <https://www.jospt.org/doi/10.2519/jospt.2009.2834>. Acesso em: 20 mar. 2021.

LIRA, V. **O que é o sensor de profundidade das cameras de celular**. São Paulo: Promobit S. T. D., 2020. Disponível em: <https://www.promobit.com.br/blog/o-que-e-o-sensor-de-profundidade-das-cameras-de-celular/>. Acesso em: 30 dez. 2021.

LUGARESI, C. *et al.* MediaPipe: A framework for building perception pipelines. **Cornell University, Google Research, Computer Science**, New York, jun. 2019. Disponível em: [arXiv:1906.08172v1](https://arxiv.org/abs/1906.08172v1). Acesso em: 10 fev. 2022.

LUN, R.; ZHAO, W. A survey of applications and human motion recognition with microsoft kinect. **International journal of pattern recognition and artificial intelligence**, New York, v. 29, n. 05, p. 1555008, 2015. Disponível em: <https://doi.org/10.1142/S0218001415550083>.

MAJGAARD, G. Teaching design of emerging embodied technologies. In: **MOVEMENT SCIENCE MEDIA JOSPT. Human-Computer interfaces and interactivity: emergent research and applications**. 5. ed. United States: IGI Global, 2014. p. 188–206. Disponível em: <https://www.igi-global.com/book/human-computer-interfaces-interactivity/102234#indices>. Acesso em: 20 mar. 2021.

MAY, S. *et al.* 3d pose estimation and mapping with time-of-flight cameras. ResearchGate GmbH, Germany, jan. 2008. Disponível em: https://www.researchgate.net/publication/228662715_3D_pose_estimation_and_mapping_with_time-of-flight_cameras.

MEDIAPIPE, G. **Apresentação do MediaPipe Solutions para aprendizado de máquina no dispositivo**. Mountain View: Google Inc., 2024. Disponível em: <https://developers.googleblog.com/pt-br/introducing-mediapipe-solutions-for-on-device-machine-learning/>. Acesso em: 23 jan. 2024.

MEDIAPIPE, G. A. **Guia de soluções MediaPipe**. Mountain View: Google Inc., 2020. Disponível em: <https://ai.google.dev/edge/mediapipe/solutions/guide?hl=pt-br>. Acesso em: 23 jul. 2023.

MEDIAPIPE, G. A. **Guia de detecção de pontos de referência de poses**. Mountain View: Google Inc., 2023. Disponível em: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker?hl=pt-br. Acesso em: 23 jul. 2023.

MELGAR, E. R.; DíEZ, C. C.; JAWORSKI, P. **Arduino and Kinect Projects: Design, build, blow their minds**. Carlstadt, N.J.: Technology in action, 2013. ISBN 9781430241676. Acesso em: 27 abr. 2021.

MELO, C. O que é visão computacional? **Sigmoidal Inc.**, Brasil, 07 2023. Disponível em: <https://sigmoidal.ai/o-que-e-visao-computacional/>. Acesso em: 27 jul. 2023.

MONTEROSSAS, P. R. D. *et al.* A brief review on the validity and reliability of microsoft kinect sensors for functional. In: FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE - STEFAN CEL MARE UNIVERSITY OF SUCEAVA. **AECE Advances in electrical and computer engineering**. Romania, 2018. v. 18, n. 1, p. 131 – 136. Acesso em: 20 mar. 2021.

MONTFOORT, I. *et al.* Interaction between ocular stabilization reflexes in patients with whiplash injury. **PubMed.gov - national library of medicine**, Bethesda, 2006. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/16799028/>. Acesso em: 20 mar. 2021.

MUTTO, C.; ZANUTTIGH, P.; CORTELAZZO, G. M. Image and speech processing, signal, image processing and computer vision. In: SPRINGER NATURE. **Time of flight cameras and microsoft Kinect**. New York: <https://link.springer.com/book/10.1007/978-1-4614-3807-6>, 2013. p. 17 – 47. ISBN 978-1-4614-3807-6. Acesso em: 20 mar. 2021.

NUITRACK 3DIVI, INC. **Documentation misc, nuitrack-sdk**. São Francisco: GitHub, Inc., 2022. Disponível em: www.github.com/3DiVi/nuitrack-sdk. Acesso em: 10 out. 2022.

OPENNI. **OpenNI User Guide**. EUA: Github, 2014. Disponível em: <https://github.com/OpenNI/OpenNI/tree/master/Documentation>. Acesso em: 10 abr. 2021.

PACKT PUBLISHING LIMITED. **Components of Kinect for Windows**. Birmingham, UK: Packt, 2022. Disponível em: https://subscription.packtpub.com/book/game_development/9781849692380/1/ch01lv1sec08/components-of-kinect-for-windows. Acesso em: 10 mar. 2022.

PEREIRA, J. C. R. Book. **Bioestatística em outras palavras**. São Paulo: Edusp: Fapesp, 2015. ISBN 10: 8531412269. Disponível em: <https://www.edusp.com.br/livros/bioestatistica-em-outras-palavras/>. Acesso em: 23 nov. 2021.

PHAM, H. H.; LE, T.; VUILLERME, N. Real-time obstacle detection system in indoor environment for the visually impaired using microsoft kinect sensor. **Journal of sensors** — Sensors, instrumentation and measurement, Switzerland, v. 2016, n. 1, p. 1 – 13, 2016. Disponível em: <http://dx.doi.org/10.1155/2016/3754918>.

PINSAULT, N. *et al.* Cervicocephalic relocation test: a study of performance stability. *ann readapt med phys. Archives of physical medicine and rehabilitation*, Bethesda, EUA, v. 49, n. 9, p. 647–651, 12 2006.

PINSAULT, N. *et al.* Test-retest reliability of cervicocephalic relocation test to the neutral head position. *Archives of physical medicine and rehabilitation*, Bethesda, EUA, v. 24, n. 5, p. 380–381, 10 2008.

PINSAULT, N.; VUILLERME, N.; PAVAN, P. Cervicocephalic relocation test to the neutral head position: Assessment in bilateral labyrinthine-defective and chronic, non traumatic neck pain patients. *Archives of physical medicine and rehabilitation*, Bethesda, EUA, v. 89, n. 12, p. 2375–2378, 12 2008.

POH, M.-Z.; MCDUFF, D.; PICARD, R. Advancements in noncontact, multiparameter physiological measurements using a webcam. **IEEE transactions on bio-medical engineering**, New York, v. 58, p. 7 – 11, 10 2010. Disponível em: <https://doi.org/10.1109/TBME.2010.2086456>.

PREDIGER, M. *et al.* Using kinect to classify parkinson's disease stages related to severity of gait impairment. **Springer nature** — Distributed, Ambient, and Pervasive Interactions, Heraklion, Greece, 2018. Acesso em: 10 mar. 2021.

PROJECT, O. **OpenKinect / libfreenect**. São Francisco, CA: Wiki, 2014. Disponível em: <https://github.com/OpenKinect/libfreenect/>. Acesso em: 16 ago. 2021.

PROJECT, O. **OpenKinect project**. São Francisco, CA: Wiki, 2014. Disponível em: https://openkinect.org/wiki/Main_Page. Acesso em: 16 ago. 2021.

PYTHON, G. M. **MediaPipe Python**. Mountain View: Google Inc., 2022. Disponível em: <https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/holistic.md>. Acesso em: 23 jul. 2022.

RAIZAH, A. *et al.* A Cross-Sectional Study on Mediating Effect of Chronic Pain on the Relationship between Cervical Proprioception and Functional Balance in Elderly Individuals with Chronic Neck Pain: Mediation analysis study. **Journal of clinical medicine**, Basel, Switzerland, p. 265 – 283, 12 2023. Disponível em: <https://doi.org/10.3390/jcm12093140>. Acesso em: 25 mar. 2021.

REGAZZONI, D.; VECCHI, G. de; RIZZI, C. Rgb cams vs rgb-d sensors: Low cost motion capture technologies performances and limitations,. *Journal of manufacturing systems*, Waltham, MA, v. 33, p. 719 – 728, jan. 2014. Disponível em: [doi:10.1016/j.jmsy.2014.07.011](https://doi.org/10.1016/j.jmsy.2014.07.011). Acesso em: 23 jul. 2021.

REHEM A. **Desenvolvimento de Ambientes Virtuais Interativos usando Java e Kinect**. São Paulo: SlidePlayer, 2022. Disponível em: <https://slideplayer.com.br/slide/11825630/>. Acesso em: 10 abr. 2022.

RESEARCH, J. **O que é uma doença neurodegenerativa**. Portugal: JPND Research, 2017. Disponível em: <https://www.neurodegenerationresearch.eu/pt-pt/o-que-e-uma-doenca-neurodegenerativa/>. Acesso em: 12 abr. 2020.

REVEL, M.; ANDRE-DESHAYS, C.; MINGUET, M. Cervicocephalic kinesthetic sensibility in patients with cervical pain. **Archives of physical medicine and rehabilitation**, Bethesda, EUA, v. 72, n. 5, p. 288–291, 12 1991. Acesso em: 20 set. 2020.

RIX, G. D.; J., B. Cervicocephalic kinesthetic sensibility in patients with chronic, nontraumatic cervical spine pain. **Archives of physical medicine and rehabilitation**, Bethesda, EUA, v. 82, n. 7, p. 911–919, 07 2001. Acesso em: 20 set. 2020.

RODRIGUES, C. J. M.; MENCARI, M. A. **Sistema de Visao Estereo para Formas Polimetricas**. 1999. — UNB - Universidade de Brasília, Brasília, BR, 1999. Disponível em: <http://www.ene.unb.br/adolfo/Monographs/Graduation/TG99%20Cristiano%20J.%20Miosso%20e%20Mirele%20A.%20Mencari.pdf>. Acesso em: 20 set. 2020.

SDK. **Kit de desenvolvimento de software**. São Francisco, CA: Wikimedia, 2020. Disponível em: https://pt.wikipedia.org/wiki/Kit_de_desenvolvimento_de_software. Acesso em: 15 abr. 2020.

SHEETS, C. S. **Linguagem de programação CCS**. São Francisco, CA: wikipedia, 2023. Disponível em: https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade. Acesso em: 10 abr. 2022.

SINTHANAYOTHIN, C.; WONGWAEN, N.; BHOLSITHI, W. Skeleton tracking using kinect sensor & displaying in 3d virtual scene. *International journal of advancements in computing technology - IJACT*, Germany, v. 4, n. 11, p. 213 – 223, 2012. Disponível em: https://www.researchgate.net/publication/272853569_Skeleton_Tracking_using_Kinect_Sensor_Displaying_in_3D_Virtual_Scene. Acesso em: 10 mar. 2021.

SISCOUTTO, R. A.; SOARES, L. P. Livros e relatórios. **Introdução a Realidade Virtual e Aumentada**. Porto Alegre, R.S.: SBC - Sociedade Brasileira de Computação, 2020. Disponível em: <https://books-sol.sbc.org.br/index.php/sbc/catalog/book/66>. Acesso em: 23 abr. 2022.

SOLOMONOW, M. *et al.* The ligamento-muscular stabilizing system of the spine. **Archives of physical medicine and rehabilitation**, Bethesda, EUA, v. 23, n. 23, p. 2552–2562, 12 1998. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/9854754/>. Acesso em: 10 mar. 2020.

STIVANELLO, M. *et al.* Desenvolvimento de uma biblioteca para sistemas de visao estereoscopica para robotica movel. In: JAFFRAY, D. A. (Ed.). **INDUSCON - VIII Conferencia internacional de aplicacoes industriais**. Germany: [s.n.], 2008. Disponível em: https://www.researchgate.net/publication/266485538_DESENVOLVIMENTO_DE_UMA_BIBLIOTECA_PARA_SISTEMAS_DE_VISAO_ESTEREOSCOPICA_PARA_ROBOTICA_MOVEL. Acesso em: 23 abr. 2022.

TÉCNICAS, A. B. D. N. **Regra de arredondamento**. São Paulo: [s.n.], 1977. Disponível em: <https://www.normas.com.br/visualizar/abnt-nbr-nm/5276/nbr5891-regras-de-arredondamento-na-numeracao-decimal>. Acesso em: 10 fev. 2023.

TORRES, R. *et al.* Diagnosis of the corporal movement in parkinson's disease using kinect sensors. In: JAFFRAY, D. A. (Ed.). **World Congress on medical physics and biomedical engineering, June 7-12, 2015**. Toronto: Springer International Publishing, 2015. p. 1445–1448. ISBN 978-3-319-19387-8. Acesso em: 10 mar. 2021.

TRELEAVEN, J. Sensorimotor disturbances in neck disorders affecting postural stability, head and eye movement control. **Archives of physical medicine and rehabilitation**, Bethesda, EUA, v. 13, n. 1, p. 2–11, 02 2008. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/17702636/>. Acesso em: 10 mar. 2021.

UIUX.PT. **About NUI - Natural User Interface**. São Paulo: Uinux-pt, 2020. Disponível em: <https://www.uiux.pt/2016/11/07/nui-natural-user-interface/>. Acesso em: 10 abr. 2020.

VISION, C. stereo. **Computer stereo vision**. São Francisco, CA: Wikimedia, 2020. Disponível em: https://en.wikipedia.org/wiki/Computer_stereo_visionr. Acesso em: 10 mar. 2021.

WEBB, J.; ASHLEY, J. **Beginning Kinect Programming with the Microsoft Kinect SDK**. Springer Nature. 1. ed. Toronto: Apress Berkeley, CA, 2012. ISBN 978-1-4302-4104-1. Disponível em: <https://doi.org/10.1007/978-1-4302-4105-8>. Acesso em: 10 abr. 2022.

WENNGREN, B. I. *et al.* Eye motility and auditory brainstem response dysfunction after whiplash injury. **Archives of physical medicine and rehabilitation**, Bethesda, EUA, v. 122, n. 3, p. 276–283, 12 2002. Acesso em: 10 mar. 2020.

XIAOBAI, L. *et al.* Remote heart rate measurement from face videos under realistic situations. **The IEEE Conference on computer vision and pattern recognition (CVPR)**, EUA, p. 4264 – 4271, 2014. Disponível em: <https://doi.org/10.1109/TBME.2010.2086456>. Acesso em: 10 mar. 2020.

ZENNARO, S. *et al.* Performance evaluation of the 1st and 2nd generation kinect for multimedia applications. **2015 IEEE International conference on multimedia and expo (ICME)**, EUA, p. 1 – 6, 2015. Disponível em: <https://ieeexplore.ieee.org/document/7177380>. Acesso em: 10 mar. 2020.

APÊNDICE A – SOFTWARE JAVASCRIPT

O sistema de interface foi desenvolvido em JavaScript. Mesmo não sendo o foco do trabalho, é importante deixar claro como foi produzido. O programa foi desenvolvido utilizando a linguagem JavaScript, CCS e HTML, para que fosse possível a utilização em um navegador. Pode ser utilizado pelo Google Chrome, Internet Explore, Microsoft Edge, Opera, Mozilla Firefox, Safari (Apple), Internet Explorer e Microsoft Edge. Os testes foram realizados com todos, mas é importante salientar que o aplicativo MediaPipe pertence a Google e sugerimos a utilização do Google Chrome como o principal navegador. O programa desenvolvido é apresentado a seguir.

Programação em JavaScript

```
<html> \\
<head> \\
<title>CapturaDados Pose</title> \\
<meta charset="UTF-8"> \\
<meta name="viewport" \\
content="width=device-width,
initial-scale=1.0">\\
<script type="text/javascript" src="imports/jquery-3.6.0.js">
</script>\\
<link href="imports/bootstrap/bootstrap.min.css" rel="stylesheet" >\\
<script src="imports/bootstrap/bootstrap.min.js"></script>\\
<script src="imports/funcoes.js" type="text/javascript"></script>\\
<script src="https://cdn.jsdelivr.net/npm/@mediapipe/camera
_utils/camera_utils.js"\\ crossorigin="anonymous"></script>\\
<script src="https://cdn.jsdelivr.net/npm/@mediapipe/control
_utils/control_utils.js"\\ crossorigin="anonymous"></script>\\
<script src="https://cdn.jsdelivr.net/npm/@mediapipe/drawing_utils
/drawing
_utils.js"\\ crossorigin="anonymous"></script>\\
<script src="https://cdn.jsdelivr.net/npm/@mediapipe/face
_utils/face_utils.js"\\ crossorigin="anonymous"></script>\\
<script src="https://code.highcharts.com/highcharts.js"></script>\\
<script src="https://code.highcharts.com/highcharts-more.js">
</script>\\
<script src="https://code.highcharts.com/modules/series-label.js">
</script>\\
<script src="https://code.highcharts.com/modules/exporting.js">
</script>\\
<script src="https://code.highcharts.com/modules/export-data.js">
</script>\\
<script src="https://code.highcharts.com/modules/accessibility.js">
```

```

</script>\\
</head>\\
<nav class="navbar navbar-expand-lg navbar-light bg-light">\\
<div class="container-fluid">\\
<a class="navbar-brand" href="#">CapturaDados </a>\\
<button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-controls=
"navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">\\
<span class="navbar-toggler-icon"></span>\\
</button>\\
<div class="collapse navbar-collapse" id="navbarSupportedContent">\\
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<li class="nav-item">\\
<a class="nav-link active" aria-current="page" href="index.html">
Home</a>\\
</li>\\
<li class="nav-item">\\
<a class="nav-link active" aria-current="page"
href="searchfacemesh.html">Search-FaceMesh</a>\\
</li>\\
</ul>\\
</div>\\
</div>\\
</nav>\\
<br>\\
<!-- Modal Para Export -->\\
<div class="modal fade" id="modalExport" tabindex="-1"
\\ aria-labelledby=
"exampleModalLabel" aria-hidden="true">\\
<div class="modal-dialog">\\
<div class="modal-content">\\
<div class="modal-header">\\
<h5 class="modal-title" id="exampleModalLabel">Exportar Dados</h5>\\
<button type="button" class="btn-close" data-bs-dismiss="modal"
\\ aria-label="Close"></button>\\
</div>\\
<div class="modal-body">\\
<div class="input-group">\\
<div class="row">\\
<div class="col-sm-12 col-md-12 col-lg-12">\\
<label for="name-text" class="col-form-label">Name:</label>\\
<input id="name" type="text" class="form-control">\\
</div>\\
<div class="col-sm-12 col-md-4 col-lg-4">\\
<label for="old-text" class="col-form-label">Age</label>\\
<input id="age" type="number" class="form-control">\\

```

```

</div>\\
<div class="col-sm-12 col-md-4 col-lg-4">
<label for="sex-text" class="col-form-label">Sex</label>\\
<select id="sex" class="form-select "
aria-label=".form-select-sm example">\\
<option selected disabled value="">Sex</option>\\
<option value="M">Male</option>\\
<option value="F">Famele</option>\\
</select>\\
</div>\\
</div>\\
</div>\\
</div>\\
<div class="modal-footer">\\
<button type="button" class="btn btn-success exportcsv">
Export</button>\\
<button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Close</button>\\
</div>\\
</div>\\
</div>\\
</div>\\
<div class="row">\\
<div class="col-sm-12 col-md-4 col-lg-4">\\
<div class="input-group mb-3">\\
<button type="button" id="button" class="btn btn-outline-secondary"
>Get camera</button>\\
<select class="form-select" id="select"><option></option></select>\\
</div>\\
</div>\\
<div class="col-sm-12 col-md-4 col-lg-4">\\
<center><h4>Reading\\ <b>FACEMESH</b></h4></center><br>\\
</div>\\
</div>\\
<div class="row">\\
<div class="col-sm-12 col-md-6 col-lg-6">\\
<center><h4>Image</h4></center>\\
<div class="shadow-sm p-3 mb-5 bg-body rounded">\\
<video id="video" autoplay playsinline class="input\_video"
style="width:
100\%; height:\\ 500px" hidden></video>\\
<canvas class="output\_canvas" width="500px" height="350px">
</canvas>\\
</div>\\
</div>\\
<div class="col-sm-12 col-md-6 col-lg-6">\\
<center>\\

```

```

<h4>Panel </h4>\\
</center>\\
<div class="shadow-sm p-3 mb-5 bg-body rounded">\\
<div class="row justify-content-center">\\
<center>\\
<div class="input-group">\\
<div class="col-sm-12 col-md-4 col-lg-4">\\
<select class="form-select" id="start\_test" aria-label=
"Default select example">\\
<option value="" selected>Selected Test</option>\\
<option value="1">Rotation Right</option>\\
<option value="2">Rotation Left</option>\\
</select>\\
</div>\\
<div class="col-sm-12 col-md-2 col-lg-2">\\
<input type="text" placeholder="Average points" id="mediapoints"
class="form-control">\\
</div>\\
<button type="button" id="seleccionartest" class="btn btn-primary
seleccionartest">Start</button>\\
<button type="button" class="btn btn-danger btnresetarponto">
Reset</button>\\
</div>\\
</center>\\
</div>\\
<br>\\
<center>\\
<button type="button" class="btn btn-primary btnregistrarpon
to">Register Point</button>\\
<button type="button" class="btn btn-info btnregistrarpon
toautomatico">Register Automatically</button>\\
<button type="button" class="btn btn-success btnabrirmodalexport"
>Export</button>\\
<button type="button" id="gerarrelatorio" class="btn btn-success"
>Relatorio</button>\\
</center>\\
<br><br>\\
<center>\\
<div id="posicaodeleitura"></div>\\
<input id="inputposicaoleitura" hidden>\\
</center>\\
<br>\\
<div id="contadorderegistrosLoader"\\ class="row"></div>\\
<div id="contadorderegistros" class="row"></div>\\
<br>\\
<div style="display: none" class="row mt-2">\\
<div class="col-sm-12 col-md-4 col-lg-4">\\

```

```

<div class="input-group input-group-sm mb-3">\
<span class="input-group-text" id="basic-addon1">14l \&nbsp;<b>X</b>
</span>\
<input type="text" id="coletadados14lx" class="form-control">\
</div>\
</div>\
<div class="col-sm-12 col-md-4 col-lg-4">\
<div class="input-group input-group-sm mb-3">\
<span class="input-group-text" id="basic-addon1">14l \&nbsp;<b>Y</b>
</span>\
<input type="text" id="coletadados14ly" class="form-control">\
</div>\
</div>\
<div class="col-sm-12 col-md-4 col-lg-4">\ <div class="input-group
input-group-sm mb-3">\
<span class="input-group-text" id="basic-addon1">14l \&nbsp;<b>Z</b>
</span>\
<input type="text" id="coletadados14lz" class="form-control">\
</div>\
</div>\
</div>\
<div class="row">\
<div class="col-sm-12 col-md-4 col-lg-4">\
<div class="input-group input-group-sm mb-3">\
<span class="input-group-text" id="basic-addon1">94 \&nbsp;<b>X</b>
</span>\
<input type="text" id="coletadados94x" class="form-control">\
</div>\
</div>\
<div class="col-sm-12 col-md-4 col-lg-4">\
<div class="input-group input-group-sm mb-3">\
<span class="input-group-text" id="basic-addon1">94 \&nbsp;<b>Y</b>
</span>\
<input type="text" id="coletadados94y" class="form-control">\
</div>\
</div>\
<div class="col-sm-12 col-md-4 col-lg-4">\
<div class="input-group input-group-sm mb-3">\
<span class="input-group-text" id="basic-addon1">94 \&nbsp;<b>Z</b>
</span>\
<input type="text" id="coletadados94z" class="form-control">\
</div>\
</div>\
</div>\
<div style="display: none" class="row">\
<div class="col-sm-12 col-md-4 col-lg-4">\
<div class="input-group input-group-sm mb-3">\

```

```

<span class="input-group-text" id="basic-addon1">370 \&nbsp;<b>X</b>
</span>\\
<input type="text" id="coletadados370x" class="form-control">\\
</div>\\
</div>\\
<div class="col-sm-12 col-md-4 col-lg-4">\\
<div class="input-group input-group-sm mb-3">\\
<span class="input-group-text" id="basic-addon1">370 \&nbsp;<b>Y</b>
</span>\\
<input type="text" id="coletadados370y" class="form-control">\\
</div>\\
</div>
<div class="col-sm-12 col-md-4 col-lg-4">\\
<div class="input-group input-group-sm mb-3">\\
<span class="input-group-text" id="basic-addon1">370 \&nbsp;<b>Z</b>
</span>\\
<input type="text" id="coletadados370z" class="form-control">\\
</div>\\
</div>\\
</div>
</div>
<div class="row">\\
<div style="display: none" class="col-sm-12 col-md-6 col-lg-6">\\
<div id="relatorio\_point\_geral" class="shadow p-3 mb-5 bg-body
rounded">\\
<figure class="highcharts-figure">\\
<div id="container141x"></div>\\
</figure>\\
</div>\\
</div>\\
<div style="display: none" class="col-sm-12 col-md-6 col-lg-6">\\
<div id="relatorio\_point\_geral" class="shadow p-3 mb-5 bg-body
rounded">\\
<figure class="highcharts-figure">\\
<div id="container141y"></div>\\
</figure>\\
</div>\\
</div>\\
<div class="col-sm-12 col-md-6 col-lg-6">\\
<div id="relatorio\_point\_geral" class="shadow p-3 mb-5 bg-body
rounded">\\
<figure class="highcharts-figure">\\
<div id="container94x"></div>\\
</figure>\\
</div>\\
</div>\\

```

```

<div class="col-sm-12 col-md-6 col-lg-6">\
<div id="relatorio\_point\_geral" class="shadow p-3 mb-5 bg-body
rounded">\
<figure class="highcharts-figure">\
<div id="container94y"></div>\
</figure>\
</div>\
</div>\
<div style="display: none" class="col-sm-12 col-md-6 col-lg-6">\
<div id="relatorio\_point\_geral" class="shadow p-3 mb-5 bg-body
rounded">\
<figure class="highcharts-figure">\
<div id="container370x"></div>\
</figure>\
</div>\
</div>\
<div style="display: none" class="col-sm-12 col-md-6 col-lg-6">\
<div id="relatorio\_point\_geral" class="shadow p-3 mb-5 bg-body
rounded">\
<figure class="highcharts-figure">\
<div id="container370y"></div>\
</figure>\
</div>\
</div>\
</div>\
<br>\
<div class="row">\
<div class="col-sm-12 col-md-6 col-lg-6">\
<div class="container">\
<div class="shadow p-3 mb-5 bg-body rounded">\
\
</div>\
</div>\
</div>\
</div>\
</div>\
<video style="display: none" class="input\_video"></video>\
<button class="btnregistrarpontoAUX"></button>\
<script type="module">\
const videoElement = document.getElementsByClassName
( 'input\_video ' )[0];\
const canvasElement = document.getElementsByClassName
( 'output\_canvas ' )[0];\
const canvasCtx = canvasElement.getContext( '2d ' );\
var coletadeDados = []; // Array de dados\
var coletadeDadoszero = []; // Array de dados\
var coletadeDadosoito = []; // Array de dados\

```

```

var coletadeDadossete = []; // Array de dados \\
var dadosrelatorio141X = []; \\
var dadosrelatorio141Y = []; \\
var dadosrelatorio94X = []; \\
var dadosrelatorio94Y = []; \\
var dadosrelatorio370X = []; \\
var dadosrelatorio370Y = []; \\
var indexponto = 0; \\ \\
//SCRIPTS DE SELECIONAR CAMERA \\
const video = document.getElementById('video '); \\
const button = document.getElementById('button '); \\
const select = document.getElementById('select '); \\
var selectCamera = null; \\
let currentStream; \\
function stopMediaTracks(stream) {
    stream.getTracks().forEach(track => {
        track.stop();
    }); \\
}
function gotDevices(mediaDevices) {
    select.innerHTML = ''; \\
    select.appendChild(document.createElement('option ')); \\
    let count = 1; \\
    mediaDevices.forEach(mediaDevice => \\ {
        if (mediaDevice.kind === 'videoinput') {
            const option = document.createElement('option '); \\
            option.value = mediaDevice.deviceId;
            const label = mediaDevice.label || 'Camera \${count++}'; \\
            const textNode = document.createTextNode(label); \\
            option.appendChild(textNode); \\
            select.appendChild(option);
        }
    }); \\
} \\
function DeviceSelect(selectCamera) { \\
    if (typeof currentStream !== 'undefined') \\ {
        stopMediaTracks(currentStream);
    } \\
    const videoConstraints = {}; \\
    if (selectCamera === '') {
        videoConstraints.facingMode = 'environment'; \\
    } else {
        videoConstraints.deviceId = {exact: selectCamera}; \\
    }
    const constraints = {
        video: videoConstraints,
        audio: false
    }
}

```

```

    });\\
    navigator.mediaDevices
    .getUserMedia(constraints)
    .then(stream => {
        currentStream = stream;\\
        video.srcObject = stream;\\
        return navigator.mediaDevices.enumerateDevices();\\
    })\\
    .then(gotDevices)\\
    .catch(error => {
        console.error(error);\\
    });\\
}
navigator.mediaDevices.enumerateDevices().then(gotDevices);\\
\\$("#btnregistrarponto").prop("disabled", true);\\
\\$("#btnregistrarpontoautomatico").prop("disabled", true);\\
\\$("#btnabrirmodalexport").prop("disabled", true);\\
\\$("#gerarrelatorio").prop("disabled", true);\\
\\$(document).on("click", "\\#selecionartest", function () {\\
    var starttest = \\$("#start\\_test option:selected").val();\\
    console.log(starttest);\\
    if (starttest !== "")\\ {
        if (starttest === '1') \\{
            \\$("#posicaodeleitura").html('<div class="alert
            alert-warning" role="alert"><span>READING <b>' +
            \\$("#start\\_test option:selected").text()
            + '</b></span></div>');\\
        } else if (starttest === '2')\\ {
            \\$("#posicaodeleitura").html('<div class="alert
            alert-warning" role="alert"><span>READING <b>' +
            \\$("#start\\_test option:selected").text() +
            '</b></span></div>');\\
        }
        \\$("#btnregistrarponto").prop("disabled", false);\\
        \\$("#btnregistrarpontoautomatico").
        prop("disabled", false);\\
        \\$("#btnabrirmodalexport").prop("disabled", false);\\
        \\$("#gerarrelatorio").prop("disabled", false);\\
    } else {\\
        \\$("#btnregistrarponto").prop("disabled", true);\\
        \\$("#btnregistrarpontoautomatico").prop("disabled", true);\\
        \\$("#btnabrirmodalexport").
        prop("disabled", true);\\
        \\$("#gerarrelatorio").prop("disabled", true);\\
        alert("Select test to start");\\
    }
});\\

```

```

\$(document).on("change", "\#select", function () {\
    selectCamera = \$("#select option:selected").val();
});\\\
//FINAL SCRIPT DO FINAL CAMERA\\
\$(document).on("click", ".btnabrirmodalexport", function () {\
    if (indexponto < 20) {\
        const music = new Audio('imports/sons/sommerro.mp3');\
        music.play();\
        alert("Unable to export, finish readings. (20 Readings)");\
        return false;
    }\\\
    //Limpar campos Modal\\
    \$("#name").val(' ');
    \$("#age").val(' ');
    \$("#sex").val(' ');
    //Abre Modal\\
    \$('#modalExport').modal('show');
});\\
\$(document).on("click", ".exportcsv", function () {\
    if (coletadeDados.length <= 0) {\
        alert('Start Collectiong');\
    } else {\
        var name = \$("#name").val();\
        var age = \$("#age").val();\
        var sex = \$("#sex").val();\
        if (age <= 0) {\
            alert('Age cannot be less than 0');\
            return false;
        }\\
        if (name !== "" && age !== "" && sex !== "") {\
            var starttest = \$("#start\_test option:selected").val();\
            if (starttest === '1') {\
                starttest = 'FACERIGHT';\
            } else if (starttest === '2') {\
                starttest = 'FACELEFT';\
            }
            var concatName = name + "\_" + age + "\_" +
            sex + "\_" + starttest;\
            var csv = '141X;141Y;141Z;94X;94Y;94Z;370X\\;370Y;370Z
            ;$\\backslash$n';\
            var cont = 0;\
            coletadeDados.forEach(function (row) {\
                csv += row[0].pontumx;\
                csv += ';' + row[0].pontumy;\
                csv += ';' + row[0].pontumz;\
                //csv += ';' + row[0].pointsutilizadosumX;\
                //csv += ';' + row[0].pointsutilizadosumY;\
            });
        }
    }
}

```

```

// csv += ';' + row[0].pointsutilizadosumZ;\\
csv += ';' + row[0].pontdoisx;\\
csv += ';' + row[0].pontdoisy;\\
csv += ';' + row[0].pontdoisz;\\
// csv += ';' + row[0].pointsutilizadosdoisX;\\
// csv += ';' + row[0].pointsutilizadosdoisY;\\
// csv += ';' + row[0].pointsutilizadosdoisZ;\\
csv += ';' + row[0].ponttresx;\\
csv += ';' + row[0].ponttresy;\\
csv += ';' + row[0].ponttresz + ";\$\\backslash$n";
// csv += ';' + row[0].pointsutilizadostresX;\\
// csv += ';' + row[0].pointsutilizadostresY;\\
// csv += ';' + row[0].pointsutilizadostresZ +
";\$\\backslash$n";

});\\

var hiddenElement = document.createElement('a');\\
hiddenElement.href = 'data:text/csv;charset=utf-8,
' + encodeURIComponent(csv);\\
hiddenElement.target = '_blank';\\
hiddenElement.download = concatName + '.csv';\\
hiddenElement.click();\\

} else {\\
    alert('Enter data');
}

}
}); \\
var coletadados141x = [];\\
var coletadados141y = [];\\
var coletadados141z = [];\\
var coletadados94x = [];\\
var coletadados94y = [];\\
var coletadados94z = [];\\
var coletadados370x = [];\\
var coletadados370y = [];\\
var coletadados370z = [];\\
\\$(document).on("click", ".btnregistrarpontoAUX", function () {\\
    coletadados141x.push(\\$("#coletadados141x").val());\\
    coletadados141y.push(\\$("#coletadados141y").val());\\
    coletadados141z.push(\\$("#coletadados141z").val());\\
    coletadados94x.push(\\$("#coletadados94x").val());\\
    coletadados94y.push(\\$("#coletadados94y").val());\\
    coletadados94z.push(\\$("#coletadados94z").val());\\
    coletadados370x.push(\\$("#coletadados370x").val());\\
    coletadados370y.push(\\$("#coletadados370y").val());\\
    coletadados370z.push(\\$("#coletadados370z").val());
});\\
\\$("#inputposicaooleitura").val('0');\\

```

```

var zerox = 0;\\
var zerox = 0;\\
var zeroz = 0;\\
var html = ' ';\\
\\$(document).on("click", ".btnregistrarponto", function () {\\
    coletadados141x = [];\\
    coletadados141y = [];\\
    coletadados141z = [];\\
    coletadados94x = [];\\
    coletadados94y = [];\\
    coletadados94z = [];\\
    coletadados370x = [];\\
    coletadados370y = [];\\
    coletadados370z = [];\\
    var typereading = "";\\
    if (indexponto >= 20) {\\
        alert('Maximum readings performed');\\
        return false;
    }\\
    \\$(".btnregistrarponto").prop("disabled", true);\\
    var cont = 200;\\
    for (let i = 0; i < \\$("#mediapoints").val(); i++) {
        setTimeout(function () {\\
            \\$(".btnregistrarpontoAUX").click();\\
        }, cont);\\
        cont = cont + 100;
    }\\
    setTimeout(function () {\\
        \\$(".btnregistrarponto").prop("disabled", false);\\
        //Ponto 141 \\
        var valorcoletadados141x = 0;\\
        coletadados141x.forEach(function (item, indice, array) {\\
            valorcoletadados141x = parseFloat(valorcoletadados141x)
            + parseFloat(item);
        });\\
        var valorcoletadados141y = 0;\\
        coletadados141y.forEach(function (item, indice, array) {\\
            valorcoletadados141y = parseFloat(valorcoletadados141y)
            + parseFloat(item);
        });\\
        var valorcoletadados141z = 0;\\
        coletadados141z.forEach(function (item, indice,
        array) {\\
            valorcoletadados141z = parseFloat
            (valorcoletadados141z)
            + parseFloat(item);
        });\\
    }

```

```

////////////////////////////////////
Ponto 94
////////////////////////////////////\
var valorcoletadados94x = 0;\
coletadados94x.forEach(function (item, indice,
array) \{
    valorcoletadados94x =
    parseFloat
    (valorcoletadados94x) + parseFloat(item);
});\
var valorcoletadados94y = 0;\
coletadados94y.forEach(function (item, indice,
array) \{
    valorcoletadados94y = parseFloat
    (valorcoletadados94y) + parseFloat(item);
});\
var valorcoletadados94z = 0;\
coletadados94z.forEach(function (item, indice,
array)\{
    valorcoletadados94z = parseFloat
    (valorcoletadados94z) + parseFloat(item);
});\
////////////////////////////////////
Ponto 370
//////////////////////////////////// \
var valorcoletadados370x = 0;\
coletadados370x.forEach(function (item, indice,
array)\{
    valorcoletadados370x = parseFloat
    (valorcoletadados370x) + parseFloat(item);
});\
var valorcoletadados370y = 0;\
coletadados370y.forEach(function (item, indice,
array) \{
    valorcoletadados370y = parseFloat
    (valorcoletadados370y) + parseFloat(item);
});\
var valorcoletadados370z = 0;\
coletadados370z.forEach(function (item, indice,
array)\{
    valorcoletadados370z = parseFloat
    (valorcoletadados370z) + parseFloat(item);
});\
//Coletando Dados\
var dados = [
{
pontumx: (valorcoletadados141x /\ \$("#mediapoints")

```

```

        .val()),
        pontumy: (valorcoletadados141y /\ \$("#mediapoints")
        .val()),
        pontumz: (valorcoletadados141z /\ \$("#mediapoints")
        .val()),
        pointsutilizadosumX: coletadados141x.join(", "),\\
        pointsutilizadosumY: coletadados141y.join(", "),\\
        pointsutilizadosumZ: coletadados141z.join(", "),\\
        pontdoisx: (valorcoletadados94x /\ \$("#mediapoints")
        .val()),\\
        pontdoisy: (valorcoletadados94y /\ \$("#mediapoints")
        .val()),\\
        pontdoisz: (valorcoletadados94z /\ \$("#mediapoints")
        .val()),\\
        pointsutilizadosdoisX: coletadados94x.join(", "),\\
        pointsutilizadosdoisY: coletadados94y.join(", "),\\
        pointsutilizadosdoisZ: coletadados94z.join(", "),\\
        ponttresx: (valorcoletadados370x /\ \$("#mediapoints")
        .val()),\\
        ponttresy: (valorcoletadados370y /\ \$("#mediapoints")
        .val()),\\
        ponttresz: (valorcoletadados370z /\ \$("#mediapoints")
        .val()),\\
        pointsutilizadostresX: coletadados370x.join(", "),\\
        pointsutilizadostresY: coletadados370y.join(", "),\\
        pointsutilizadostresZ: coletadados370z.join(", "),
    }

    ];\\
    coletadeDados[indexponto] = dados;\\
    // console.log(coletadeDados[indexponto]);\\
    html = '<div class="col-sm-12 col-md-3 col-lg-3">';\\
    html += '<center style="font-size: 13px"><b>Leitura - ' + (indexponto + 1) + '</b></center>';\\
    html += '<center><label style="font-size: 10px"><span class="badge bg-success">Leitura Realizada!</span></label></center><br>';\\
    html += '</div>';\\
    \$("#contadorderegistros").append(html);\\
    indexponto++;\\
    if (indexponto % 2 === 0) {
        const music = new Audio('imports/sons/beep.wav');\\
        music.play();\\
    } else {\\
        const music = new
        Audio('imports/sons/beepretorno.wav');\\
        music.play();
    }\\
}

```

```

}, cont);
});\\
\\$(document).on("click", "\\ ".btnregistrarponautoautomatico",
function () {\\
    var contlopping = 1000;\\
    var contatorletirauvisul = 1;\\
    for (let i = 1; i <= 20; i++) {\\
        \\$("##contadorderegistrosLoader").html('<center><p>
Processing Read... <a id="contadorleituravisul"></a><div
class="spinner-grow text-danger" role="status"><span
class="visually-hidden">Loading ... </span></div>
</p></center>');\\
        setTimeout(function () {\\
            \\$(".btnregistrarponauto").
            prop("disabled", true);\\
            \\$(".btnregistrarponautoautomatico").
            prop("disabled", true);\\
            coletadados141x = [];\\
            coletadados141y = [];\\
            coletadados141z = [];\\
            coletadados94x = [];\\
            coletadados94y = [];\\
            coletadados94z = [];\\
            coletadados370x = [];\\
            coletadados370y = [];\\
            coletadados370z = [];\\
            var cont = 200;\\
            for (let i = 0; i < \\$("##mediapoints").val(); i++) {
                setTimeout(function () {\\
                    \\$(".btnregistrarponautoAUX").click();\\
                }, cont);\\
                cont = cont + 100;\\
            }
            setTimeout(function () {\\
                \\$(".btnregistrarponauto").
                prop("disabled", false);\\
                //////////////////////////////////
                Ponto 141
                ////////////////////////////////// \\
                var valorcoletadados141x = 0;\\
                coletadados141x.forEach(function
                (item, indice, array) {\\
                    valorcoletadados141x =
                    parseFloat(valorcoletadados141x)
                    + parseFloat(item);
                });\\
                var valorcoletadados141y = 0;\\

```

```

coletadados141y.forEach(function
(item, indice, array) {\
    valorcoletadados141y =
    parseFloat(valorcoletadados141y)
    + parseFloat(item);
});\
var valorcoletadados141z = 0;\
coletadados141z.forEach(function
(item, indice, array) {\
    valorcoletadados141z =
    parseFloat(valorcoletadados141z)
    + parseFloat(item);
});\
////////////////////////////////
Ponto 94
//////////////////////////////// \
var valorcoletadados94x = 0;\
coletadados94x.forEach(function
(item, indice, array) {\
    valorcoletadados94x =
    parseFloat(valorcoletadados94x)
    + parseFloat(item);
});\
var valorcoletadados94y = 0;\
coletadados94y.forEach(function
(item, indice, array) {\
    valorcoletadados94y =
    parseFloat(valorcoletadados94y)
    + parseFloat(item);
});\
var valorcoletadados94z = 0;\
coletadados94z.forEach(function
(item, indice, array) {\
    valorcoletadados94z =
    parseFloat(valorcoletadados94z)
    + parseFloat(item);
});\
////////////////////////////////
Ponto 370
//////////////////////////////// \
var valorcoletadados370x = 0;\
coletadados370x.forEach(function
(item, indice, array) {\
    valorcoletadados370x =
    parseFloat(valorcoletadados370x)
    + parseFloat(item);
});\

```

```

var valorcoletadados370y = 0;\n
coletadados370y.forEach(function
(item, indice, array)\n {
    valorcoletadados370y =
    parseFloat(valorcoletadados370y)
    + parseFloat(item);
});\n
var valorcoletadados370z = 0;\n
coletadados370z.forEach(function
(item, indice, array) \n{
    valorcoletadados370z =
    parseFloat(valorcoletadados370z)
    + parseFloat(item);
});\n
//////// Coletando Dados //////////\n
var dados = [
{
    pontumx: (valorcoletadados141x / \$("#mediapoints").val()),\n
    pontumy: (valorcoletadados141y / \$("#mediapoints").val()),\n
    pontumz: (valorcoletadados141z / \$("#mediapoints").val()),\n
    pointsutilizadosumX: coletadados141x.join(", "),\n
    pointsutilizadosumY: coletadados141y.join(", "),\n
    pointsutilizadosumZ: coletadados141z.join(", "),\n
    pontdoisx: (valorcoletadados94x / \$("#mediapoints").val()),\n
    pontdoisy: (valorcoletadados94y / \$("#mediapoints").val()),\n
    pontdoisz: (valorcoletadados94z / \$("#mediapoints").val()),\n
    pointsutilizadosdoisX: coletadados94x.join(", "),\n
    pointsutilizadosdoisY: coletadados94y.join(", "),\n
    pointsutilizadosdoisZ: coletadados94z.join(", "),\n
    ponttresx: (valorcoletadados370x / \$("#mediapoints").val()),\n
    ponttresy: (valorcoletadados370y / \$("#mediapoints").val()),\n
    ponttresz: (valorcoletadados370z / \$("#mediapoints").val()),\n
    pointsutilizadostresX: coletadados370x.join(", "),\n
    pointsutilizadostresY: coletadados370y.join(", "),\n
    pointsutilizadostresZ: coletadados370z.join(", "),
    }
];\n
coletadeDados[indexponto] = dados;\n
//////// console.log(coletadeDados[indexponto]);////////\n
html = '<div class="col-sm-12 col-md-3 col-lg-3">';\n
html += '<center style="font-size: 13px"><b>Leitura - '
+ (indexponto + 1) + '</b></center>';\n
html += '<center><label style="font-size: 10px"><span
class="badge bg-success">Leitura
Realizada!</span></label></center><br>';\n
html += '</div>';\n
\$("#contadorderegistros").append(html);\n

```

```

indexponto++;
if (indexponto % 2 === 0) {\
const music = new Audio('imports/sons/beep.wav');
music.play();\
    } else {\
const music = new Audio('imports/sons/beepretorno.wav');\
music.play();
    }\
    \$("#contadorleituravisul").text(contatorletirauvisul++
+ '/20');\
    }, cont);\
    }, contlopping);\
    contlopping = contlopping + 1000;
}\
setTimeout(function () {\
    \$("#contadorderegistrosLoader").html("");\
}, (contlopping + 500));\
});\
\$(document).on("click", ".btnresetarponto", function () {
    coletadeDados = [];\
    indexponto = 0;\
    \$("#contadorderegistros").html('');\
    \$(".btnregistrar ponto").prop("disabled", true);\
    \$(".btnabrirmodalexport").prop("disabled", true);\
    \$("#gerarrelatorio").prop("disabled", true);\
    \$("#start\_test").val('');\
});\
function onResults(results) {
    canvasCtx.save();
    canvasCtx.clearRect(0, 0,
    canvasElement.width, canvasElement.height);\
    canvasCtx.drawImage(
    results.image, 0, 0, canvasElement.width, canvasElement.height);\
    if (results.multiFaceLandmarks) {\
        for (const landmarks of results.multiFaceLandmarks) {
            drawConnectors(canvasCtx, landmarks,
            FACEMESH\_TESSELATION,
            {color: '\#C0C0C070', lineWidth: 1});\
            drawConnectors(canvasCtx, landmarks,
            FACEMESH\_RIGHT\_EYE, {color: '\#FF3030'});\
            drawConnectors(canvasCtx, landmarks,
            FACEMESH\_RIGHT\_EYE, {color: '\#FF3030'});\
            drawConnectors(canvasCtx, landmarks,
            FACEMESH\_RIGHT\_IRIS, {color: '\#FF3030'});\
            drawConnectors(canvasCtx, landmarks,
            FACEMESH\_LEFT\_EYE, {color: '\#30FF30'});\
            drawConnectors(canvasCtx, landmarks,

```

```

    FACEMESH\_LEFT\_EYEBROW, { color: '\#30FF30' });\\
    drawConnectors(canvasCtx, landmarks,
    FACEMESH\_LEFT\_IRIS, { color: '\#30FF30' });\\
    drawConnectors(canvasCtx, landmarks,
    FACEMESH\_FACE\_OVAL, { color: '\#E0E0E0' });\\
    drawConnectors(canvasCtx, landmarks,
    FACEMESH\_LIPS, { color: '\#E0E0E0' });\\
    \$.each(landmarks, function (index, value) {\
      if (index === 141) {\
        \$("#coletadados141x").val(value.x);\\
        \$("#coletadados141y").val(value.y);\\
        \$("#coletadados141z").val(value.z);\\
      }
      if (index === 94) {\
        \$("#coletadados94x").val(value.x);\\
        \$("#coletadados94y").val(value.y);\\
        \$("#coletadados94z").val(value.z);\\
      }
      if (index === 370) {\
        \$("#coletadados370x").val(value.x);\\
        \$("#coletadados370y").val(value.y);\\
        \$("#coletadados370z").val(value.z);
      }
    });
  }
}\\
canvasCtx.restore();
}\\
const faceMesh = new FaceMesh({ locateFile: (file) => {\
  return
  'https://cdn.jsdelivr.net/npm/@mediapipe/face/_mesh/\${file}';
}});\\
faceMesh.setOptions({
  maxNumFaces: 1,
  refineLandmarks: true,
  minDetectionConfidence: 0.5,
  minTrackingConfidence: 0.5
});\\
faceMesh.onResults(onResults);\\
const camera = new Camera(videoElement, {
  onFrame: async () => {
    await faceMesh.send({ image: videoElement });
  },\\
  width: 1280,\\
  height: 720
});\\
camera.start();\\

```

```

button.addEventListener('click', event=> {
    DeviceSelect(selectCamera);\
});
\$(document).on("click", "#gerarrelatorio", function () {\
    var dados141x = [];\
    var dados141y = [];\
    var dados94x = [];\
    var dados94y = [];\
    var dados370x = [];\
    var dados370y = [];\
    var cont1410x = 0;\
    var cont1410y = 0;\
    var cont94x = 0;\
    var cont94y = 0;\
    var cont370x = 0;\
    var cont370y = 0;\
    coletadeDados.forEach(function (row) {\
        dados141x[0] = parseFloat(row[0].pontumx);\
        dados141y[0] = parseFloat(row[0].pontumy);\
        dadosrelatorio141X[cont1410x++] = dados141x;\
        dadosrelatorio141Y[cont1410y++] = dados141y;\
        //////////////////////////////////\
        dados94x[0] = parseFloat(row[0].ponttresx);\
        dados94y[0] = parseFloat(row[0].ponttresy);\
        dadosrelatorio94X[cont94x++] = dados94x;\
        dadosrelatorio94Y[cont94y++] = dados94y;\
        //////////////////////////////////\
        dados370x[0] = parseFloat(row[0].pontdoisx);\
        dados370y[0] = parseFloat(row[0].pontdoisy);\
        dadosrelatorio370X[cont370x++] = dados370x;\
        dadosrelatorio370Y[cont370y++] = dados370y;\
        dados141x = [];\
        dados141y = [];\
        dados94x = [];\
        dados94y = [];\
        dados370x = [];\
        dados370y = [];\
    });\
    //relatoriogeral141x();\
    //relatoriogeral141y();\
    relatoriogeral94x();\
    relatoriogeral94y();\
    //relatoriogeral370x();\
    //relatoriogeral370y();
});\
function relatoriogeral141x() {\
    Highcharts.chart('container141x', {

```

```

chart: {
    type: 'spline'
},\n
title: {
    text: 'General Points Report - 141 X'
},
subtitle: {
    text: ''
},\n
xAxis: {
    reversed: false ,
    title: {
        enabled: true ,
        text: 'X'
    },\n
    labels: {
        format: '{ value }'
    },
    maxPadding: 0.05,
    showLastLabel: true
},\n
yAxis: {
    title: {
        text: 'X0'
    },\n
    labels: {
        format: '{ value }'
    },
    lineWidth: 2
},\n
tooltip: {
    headerFormat: '<b>{ series.name }</b><br/>',
    pointFormat: 'X: { point.y}: Y:{ point.x}'
},\n
plotOptions: {
    series: {
        marker: {
            enabled: true
        }
    }
},\n
colors: ['\#6CF', '\#39F', '\#06C', '\#036', '\#000'],\n
// Define the data points. All series have a dummy year of 1970/71
in order to be compared on the same x axis.
Note : that in JavaScript, months start at 0 for January ,
1 for February etc.\n\n
series: [{

```

```

        name: "141-NOSE X",
        data: dadosrelatorio141X
    }],\\
    responsive: {\\
        rules: [{\\
            condition: {\\
                maxWidth: 500
            },\\
            chartOptions: {\\
                plotOptions: {
                    series: {\\
                        marker: {\\
                            radius: 2.5
                        }
                    }
                }
            }
        }]
    }
});

}\\
function relatoriogeral141y() {
    Highcharts.chart('container141y', {\\
        chart: {
            type: 'spline'
        },\\
        title: {
            text: 'General Points Report - 141 Y'
        },\\
        subtitle: {
            text: ''
        },\\
        xAxis: {
            reversed: false,
            title: {
                enabled: true,
                text: 'X'
            },\\
            labels: {
                format: '{value}'
            },\\
            maxPadding: 0.05,
            showLastLabel: true
        },\\
        yAxis: {
            title: {
                text: 'Y0'
            }
        }
    });
}

```

```

        },\n
        labels: {\n
            format: '{value}\'\n
        },\n
        lineWidth: 2\n
    },\n
    tooltip: {\n
        headerFormat: '<b>{series.name}</b><br/>'\n
        pointFormat: 'X: {point.x}: Y:{point.y}\'\n
    },\n
    plotOptions: {\n
        series: {\n
            marker: {\n
                enabled: true\n
            }\n
        }\n
    },\n
    colors: ['#6CF', '#39F', '#06C', '#036', '#000'],\n
    // Define the data points. All series have a dummy year of 1970/71\n
    in order to be compared on the same x axis.\n
    Note that in JavaScript, months start at 0 for January,\n
    1 for February etc.\\\n
    series: [{\n
        name: "141-NOSE Y",\n
        data: dadosrelatorio141Y\n
    }],\n
    responsive: {\n
        rules: [{\n
            condition: {\n
                maxWidth: 500\n
            },\n
            chartOptions: {\n
                plotOptions: {\n
                    series: {\n
                        marker: {\n
                            radius: 2.5\n
                        }\n
                    }\n
                }\n
            }\n
        }]\n
    }\n
});\n
}\n
function relatoriogeral94x() {\n
    Highcharts.chart('container94x', {\n
        chart: {\n

```

```

        type: 'spline'
    },\n
    title: {
        text: 'General Points Report - 94 X'
    },\n
    subtitle: {
        text: ''
    },\n
    xAxis: {
        reversed: false,\n
        title: {
            enabled: true,\n
            text: 'X'
        },\n
        labels: {
            format: '{value}'
        },\n
        maxPadding: 0.05,\n
        showLastLabel: true
    },\n
    yAxis: {
        title: {
            text: 'X0'
        },\n
        labels: {
            format: '{value}'
        },\n
        lineWidth: 2
    },\n
    tooltip: {
        headerFormat: '<b>{series.name}</b><br/>',\n
        pointFormat: 'X: {point.y}: Y:{point.x}'
    },\n
    plotOptions: {
        series: {
            marker: {
                enabled: true
            }
        }
    },\n
    colors: ['#6CF', '#39F', '#06C', '#036', '#000'],\n
    // Define the data points. All series have a dummy year of 1970/71
    in order to be compared on the same x axis.
    Note that in JavaScript, months start at 0 for January,
    1 for February etc.\n\n
    series: [{
        name: "94-NOSE X",

```

```

        data: dadosrelatorio94X
    }],\\
    responsive: {
        rules: [{
            condition: {
                maxWidth: 500
            },\\
            chartOptions: {
                plotOptions: {
                    series: {
                        marker: {
                            radius: 2.5
                        }
                    }
                }
            }
        }]
    }
});
}\\
function relatoriogeral94y () {\\
    Highcharts.chart('container94y', {
        chart: {
            type: 'spline'
        },\\
        title: {
            text: 'General Points Report - 94 Y'
        },\\
        subtitle: {
            text: ''
        },\\
        xAxis: {
            reversed: false,
            title: {
                enabled: true,
                text: 'X'
            },\\
            labels: {
                format: '{value}'
            },\\
            maxPadding: 0.05,
            showLastLabel: true
        },\\
        yAxis: {
            title: {
                text: 'Y0'
            },

```

```

        labels: {
            format: '{ value }'
        },
        lineWidth: 2
    },\n
    tooltip: {
        headerFormat: '<b>{ series.name }</b><br/>',
        pointFormat: 'X: { point.x } Y:{ point.y }'
    },\n
    plotOptions: {
        series: {
            marker: {
                enabled: true
            }
        }
    },\n
    colors: [ '\#6CF', '\#39F', '\#06C', '\#036', '\#000' ],\n
    // Define the data points. All series have a dummy year of 1970/71
    // in order to be compared on the same x axis.
    Note that in JavaScript, months start at 0 for January,
    1 for February etc.\n\n
    series: [{
        name: "94-NOSE Y",
        data: dadosrelatorio94Y
    }],\n
    responsive: {\n
        rules: [{\n
            condition: {
                maxWidth: 500
            },\n
            chartOptions: {
                plotOptions: {
                    series: {
                        marker: {
                            radius: 2.5
                        }
                    }
                }
            }
        }
    ]
}
});\n
}\n
function relatoriogeral370x() {\n
    Highcharts.chart('container370x', {\n
        chart: {
            type: 'spline'

```

```

    },\\
    title: {
        text: 'General Points Report - 370 X'
    },\\
    subtitle: {
        text: ''
    },\\
    xAxis: {
        reversed: false ,
        title: {
            enabled: true ,
            text: 'X'
        },\\
        labels: {
            format: '{ value }'
        },\\
        maxPadding: 0.05 ,
        showLastLabel: true
    },\\
    yAxis: {
        title: {
            text: 'X0'
        },\\
        labels: {
            format: '{ value }'
        },\\
        lineWidth: 2
    },\\
    tooltip: {
        headerFormat: '<b>{ series.name }</b><br/>',
        pointFormat: 'X: { point.y}: Y:{ point.x}'
    },\\
    plotOptions: {
        series: {
            marker: {
                enabled: true
            }
        }
    },\\
    colors: ['\#6CF', '\#39F', '\#06C', '\#036', '\#000'],\\
    // Define the data points. All series have a dummy year of 1970/71
    in order to be compared on the same x axis.
    Note that in JavaScript, months start at 0 for January,
    1 for February etc.\\
    series: [{
        name: "370-NOSE X",
        data: dadosrelatorio370X
    }

```

```

    }],\\
    responsive: {
        rules: [{
            condition: {
                maxWidth: 500
            },\\
            chartOptions: {
                plotOptions: {
                    series: {
                        marker: {
                            radius: 2.5
                        }
                    }
                }
            }
        }]
    }
});
}\\
function relatoriogeral370y() {\\
    Highcharts.chart('container370y', {
        chart: {
            type: 'spline'
        },\\
        title: {
            text: 'General Points Report - 370 Y'
        },\\
        subtitle: {
            text: ''
        },\\
        xAxis: {
            reversed: false,
            title: {
                enabled: true,
                text: 'X'
            },\\
            labels: {
                format: '{value}'
            },\\
            maxPadding: 0.05,
            showLastLabel: true
        },\\
        yAxis: {
            title: {
                text: 'Y0'
            },\\
            labels: {

```

```

                                format: '{value}'
                                },\\
                                lineWidth: 2
                                },\\
                                tooltip: {
                                    headerFormat: '<b>{series.name}</b><br/>',
                                    pointFormat: 'X: {point.x}: Y:{point.y}'
                                },\\
                                plotOptions: {
                                    series: {
                                        marker: {
                                            enabled: true
                                        }
                                    }
                                },\\
                                colors: ['#6CF', '#39F', '#06C', '#036', '#000'],
                                // Define the data points. All series have a dummy year of 1970/71
                                // in order to be compared on the same x axis.
                                Note that in JavaScript, months start at 0 for January,
                                1 for February etc.\\ \\ \\
                                series: [{
                                    name: "370-NOSE Y",
                                    data: dadosrelatorio370Y
                                }],\\
                                responsive: {
                                    rules: [{
                                        condition: {
                                            maxWidth: 500
                                        },\\
                                        chartOptions: {
                                            plotOptions: {
                                                series: {
                                                    marker: {
                                                        radius: 2.5
                                                    }
                                                }
                                            }
                                        }
                                    }
                                ]
                            }
                        });
                    }\\
                </script>
            </html>
        }
    }

```

APÊNDICE B – MINHAS REFERÊNCIAS BIBLIOGRÁFICAS

Publicações científicas – Conferências Internacionais

AGUIAR W.; THOMANN G.; FREIRE JUNIOR J. C.; CUARELLI G. "Evaluation d'un outil d'analyse biométrique de la main, pour l'assistance au diagnostic des maladies neurodégénératives" Journées G-SCOP 2022, Autran, 03 junho, 2022.

AGUIAR W.; THOMANN G.; FREIRE JUNIOR J. C.; CUARELLI G. "Evaluation d'un outil d'analyse biométrique de la main, pour l'assistance au diagnostic des maladies neurodégénératives", Poster, Handicap 2022, Paris, 08-10 jun., 2022.

Disponível em: <https://ifrath.fr/handicap2022/programme-du-jeudi-9-juin-2022/>

AGUIAR, WAGNER DE; CUARELLI, GILBERTO; GUILLAUME, THOMANN ; FREIRE JUNIOR, JOSÉ CELSO. Développement d'un système de mesure de proprioception cervicale, Conception dédiée aux professionnels de santé. Handicap 2024, Des Solution Personnalisées pour des Besoins Spécifiques, Poster, Paris, France, 5-7 Juin, 2024

Conférence Handicap 2024 – 13ème édition de la conférence sur les recherches pluridisciplinaires pour l'autonomie des personnes en situation de handicap

Disponível em: <https://ifrath.fr/handicap2024/>

Submissão de artigo em revista internacional

AGUIAR W.; FREIRE JUNIOR J. C.; GUILLAUME T.; CUARELLI G. ; "Biometric image analysis techniques for monitoring chronic neck pain (Aceito para publicação), Applied Sciences (Switzerland).