

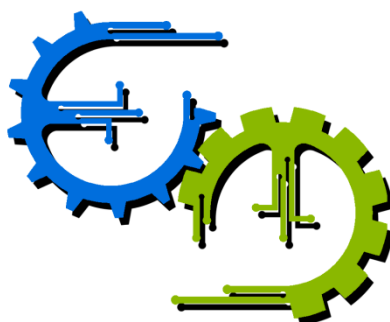


TRABALHO DE GRADUAÇÃO

Reconhecimento do alfabeto de Libras usando sensor Kinect e marcadores visuais

Por,
Giordano Bruno de Melo Gois

Brasília, Dezembro de 2014



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

TRABALHO DE GRADUAÇÃO

Reconhecimento do alfabeto de Libras usando sensor Kinect e marcadores visuais

Por,

Giordano Bruno de Melo Gois

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação

Banca Examinadora

Prof. Dr. Marcus Vinicius Lamar, UnB/CIC

(Orientador)

Prof. Msc. Marcos Fagundes Caetano, UnB/CIC

(Examinador Interno)

Msc. Juarez Paulino da Silva Jr., UnB/CIC

(Examinador Interno)

Brasília, Dezembro de 2014

FICHA CATALOGRÁFICA

GIORDANO, GOIS Reconhecimento do alfabeto de Libras usando sensor Kinect e marcadores visuais, [Distrito Federal] 2014. v, 65p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, Ano). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.	
1.Libras	2.kinect
3.Marcadores visuais	4.Câmera
I. Mecatrônica/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

GOIS, G, (2014). Reconhecimento do alfabeto de Libras usando sensor Kinect e marcadores visuais. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 20, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 65p.

CESSÃO DE DIREITOS

AUTOR: Giordano Bruno de Melo Gois.

TÍTULO DO TRABALHO DE GRADUAÇÃO: Reconhecimento do alfabeto de Libras usando sensor Kinect e marcadores visuais.

GRAU: Engenheiro

ANO: 2014

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Giordano Bruno de Melo Gois
Quadra 2 conjunto C-3 casa 23 – Sobradinho - DF.
73015-303 Brasília – DF – Brasil.

Dedicatórias

Dedico este trabalho a minha mãe, ao meu pai, a minhas irmãs, a minha sobrinha e minha namorada que são as pessoas mais importantes da minha vida.

Giordano Bruno de Melo Gois

Agradecimentos

Agradeço aos meus pais José Maria e Artemes, pelo amor incondicional, pelo apoio durante toda a minha vida e principalmente pelo orgulho demonstrado durante todos esses longos anos de estudo.

Agradeço a minhas irmãs Pamela e Sarah e a minha sobrinha Duda, pelo amor e carinho dedicados toda minha vida, pela compreensão nos dias difíceis.

Agradeço a minha namorada, Kamylla Novais, pelo amor, pelo carinho, dedicação e apoio durante os anos de graduação, por sempre acreditar no meu potencial e por confiar no nosso futuro.

Agradeço ao professor Antônio Jacó, pela dedicação em ensinar robótica durante o ensino médio, o que foi decisivo para a escolha mais certa que fiz na vida, pelo apoio durante os anos de estudo, e pelo incentivo em eu também ministrar aulas de robótica.

Agradeço a Explora Tecnologia, e especialmente a Bruno Rodrigues e Lucas dos Santos, pelo incentivo e pela oportunidade de desenvolver esse trabalho, por todo suporte e paciência que viabilizaram a sua realização.

Agradeço à equipe DROID de competição de robótica inteligente, aos funcionários dos laboratórios de engenharia mecânica e elétrica e aos professores da universidade de Brasília, em especial aos meus orientadores, professor Marcus Vinicius Lamar e professor Jones Yudi, por todas as experiências adquiridas e pelo grande desenvolvimento acadêmico, ao professor Gláucio Júnior por ser despor a ficar duas horas coletando os dados para esse trabalho.

Por último, mas não menos importante, agradeço aos meus colegas de faculdade que contribuíram para minha formação acadêmica e também aos meus amigos que de alguma forma me ajudaram, principalmente ao meu grande amigo Felipe de Paula pela parceria em todos os projetos já realizados e em todos os próximos que estão por vir.

RESUMO

Este trabalho tem como intuito apresentar o desenvolvimento de uma solução para o reconhecimento do alfabeto manual da Língua Brasileira de Sinais (Libras). Devido à grande complexidade e semelhança dos sinais, é alta a exigência por precisão e eficiência no sistema de reconhecimento baseado em processamento de vídeo. Para conseguir o desempenho desejado foi utilizada a combinação do sensor Kinect para análise de profundidade e segmentação das imagens e uma câmera RGB de alta resolução que, juntamente a uma luva com marcadores visuais, possibilitam o rastreamento das posições dos dedos da mão durante a execução dos sinais. Tal configuração permite extrair as características morfológicas que descrevem um sinal com as mãos e o representando em um vetor 12-dimensional baseado nas distâncias e ângulos relativos entre os marcadores. Para o reconhecimento criou-se para cada uma das 26 letras do alfabeto um vetor 12-dimensional representando o seu sinal.

ABSTRACT

This work has as purpose to present the development of a solution to the hand alphabet recognition of the Brazilian Sign Language (Libras). Due to the high complexity and similarity of the signs, it has a high demand for precision and efficiency in the recognition system. To achieve the desired performance, the Kinect sensor, used for analysis of depth and segmentation of images, is combined with a high resolution camera works together a glove with visual markers enable the tracking of finger position during the execution of the signs. Thereby, achieving the extraction of morphological features that describe a hand posture sign and representing in a 12-dimensional vector based on relative distances and angles between the markers. For recognition it is created for each one of the 26 letters of the hand alphabet a 12-dimensional vector representing its signal.

Sumário

1.	INTRODUÇÃO	1
1.1.	Objetivo Geral	2
1.2.	Objetivos Específicos	2
2.	REFERENCIAL TEÓRICO	4
2.1.	Libras	4
2.2.	Reconhecimento de gestos	8
2.3.	Tecnologias disponíveis	8
2.3.1.	Kinect	8
2.3.2.	Câmera infravermelho (OMEX)	9
2.3.3.	Marcadores coloridos	10
2.3.4.	Visão estéreo	11
2.3.5.	Radar laser	12
2.3.6.	Luvas instrumentalizadas	13
2.4.	Trabalhos anteriores	14
3.	METODOLOGIA PROPOSTA	17
3.1.	Sensor Kinect	17
3.2.	Câmera	19
3.3.	Luvas	20
3.4.	Marcadores	21
3.5.	Softwares e bibliotecas utilizados	23
3.5.1.	Visual Studio	23
3.5.2.	OpenCV	23
3.5.3.	Kinect SDK	24
3.6.	Integração câmera e Kinect	24
3.7.	Homografia	25
3.8.	Isolamento da mão	27
3.9.	Remoção do fundo da imagem	28
3.10.	Espaço de cores	30
3.11.	Localização dos marcadores	31
3.12.	Escolha do alfabeto	33
3.13.	Aquisição de dados	34
3.14.	Cálculo das distâncias relativas	36
3.15.	Cálculo do ângulo entre marcadores	41
3.16.	Criação dos padrões	42
3.17.	Estratégia de busca	45
4.	RESULTADOS OBTIDOS	47
4.1.	Primeiro Experimento	47
4.2.	Segundo experimento	50
4.3.	Terceiro experimento	52
4.4.	Quarto experimento	55
4.5.	Comparação entre experimentos	58
4.6.	Análise dos resultados obtidos	59
5.	CONCLUSÃO	61
6.	REFERÊNCIAS BIBLIOGRÁFICAS	63

LISTA DE FIGURAS

Figura 2.1 - Alfabeto em Libras (Língua de Sinais Brasileira) [6].....	5
Figura 2.2 Configurações de mão da LIBRAS[7].	6
Figura 2.3 - Exemplo com os principais parâmetros[7].....	7
Figura 2.4 - Exemplo de imagens obtidas pelo sensor Kinect, retirada de [http://nicolas.burrus.name/index.php/KinectRgbDemoV4]	9
Figura 2.5 - Exemplo de imagem sendo obtida por câmera infravermelho, retirada de [http://www.epnc.co.kr/atl/view.asp?a_id=9458]	10
Figura 2.6 - Exemplo de marcadores coloridos, retirada de [http://www.pranavmistry.com/projects/sixthsense/]	11
Figura 2.7 - Exemplo de visão estéreo, retirada de [http://opencvlib.weebly.com/cvfindstereocorrespondencebm.html]	12
Figura 2.8 – Exemplo de radar laser (a) e imagem capturada (b), retiradas de [http://www.autonomoustuff.com/sick-tim310.html]	13
Figura 2.9 - Exemplo luva com sensores extensômetro e giroscópio, retirada de [http://grathio.com/2010/03/rock_paper_scissors_training_glove/].	14
Figura 3.1 - Sensor kinect e suas funcionalidades, retirada de [http://xboxconsole.blogspot.com.br/2012/03/kinect-for-microsoft-xbox-360.html]	18
Figura 3.2 - Ambiente iluminado pelo projetor IR do Kinect, retirada de [http://gamerant.com/kinect-night-vision-video-dyce-51156/]	19
Figura 3.3 - Modelo da câmera utilizada, retirada de [http://www.microsoft.com/hardware/pt-br/p/lifecam-hd-5000]	20
Figura 3.4 - Luvas de Lycra utilizadas.	21
Figura 3.5 componentes de cores RGB.....	22
Figura 3.6 Luva final com os marcadores pintados.	22
Figura 3.7 - Luva inicial com o dorso pintado.	22
Figura 3.8 Montagem câmera e sensor kinect.	25
Figura 3.9 - Mapeamento entre planos.	25
Figura 3.10 - Imagem dos marcadores (rosa claro) obtida pela câmera.	26
Figura 3.11 - Imagem dos marcadores obtida pelo Kinect	26
Figura 3.12 - Homografia entre imagem colorida e de profundidade com a colorida sobreposta à imagem de profundidade.	27
Figura 3.13 - Demonstração do isolamento da mão esquerda.	28
Figura 3.14 - Imagem de profundidade com resolução dez vezes maior.	30
Figura 3.15 - Imagem de profundidade original.....	30
Figura 3.16 - Etapas de remoção do fundo. (a) Imagem original colorida, (b) imagem de profundidade e (c) resultado da segmentação.	30
Figura 3.18 - Imagem RGB normalizada.	31
Figura 3.17 - Imagem RGB original.....	31
Figura 3.19 - Exemplo de sinais e os centroides de cada marcador sinalizado.	32
Figura 3.20- Letra E: busca dos marcadores no espaço RGB.....	33
Figura 3.21- Letra E: busca dos marcadores no espaço YCrCb.....	33
Figura 3.22 - Letra E: interseção dos pixels detectados em RGB e YCrCb.....	33
Figura 3.23- Letra E: união dos pixels detectados em RGB e YCrCb	33
Figura 3.24- Letra G: imagem RGB	35
Figura 3.25- Letra G: imagem de profundidade	35
Figura 3.26 Letra G: imagem RGB normalizada	36
Figura 3.27- Letra G: imagem RGB normalizada com fundo removido.....	36

Figura 3.28- Letra G, distâncias entre o marcador vermelho e os outros quatro marcadores.	39
Figura 3.29 - Gráfico com as distâncias normalizadas para as 12 letras que tem classificador no grupo 31.	40
Figura 3.30 - Dois gestos com mesma configuração de mão com rotação 180° entre as imagens.	41
Figura 3.31 Letra C: Exemplo das 10 imagens coloridas em diferentes posições	42
Figura 3.32- Letra C: Exemplo das imagens de profundidade em diferentes posições.	42
Figura 4.1- Exemplo do sinal “F”	49
Figura 4.2- Exemplo do sinal “T”	49
Figura 4.4- Letra D: imagem para análise	51
Figura 4.3 - Letra D: imagem de profundidade.....	51
Figura 4.5 - Exemplo sinal “O”	51
Figura 4.6 - Exemplo sinal “C”	51
Figura 4.7 - Exemplo sinal “N”	54
Figura 4.8 - Exemplo sinal “M”	54
Figura 4.9- Exemplo sinal “Z”	57
Figura 4.10- Exemplo sinal “X”	57
Figura 4.11- Exemplo sinal da “E” feita pelo usuário A.....	60
Figura 4.12- Exemplo sinal da “E” feita pelo usuário B.....	60

LISTA DE TABELAS

Tabela 3.1 – Valores das componentes RGB e YCrCb e suas margens.	31
Tabela 3.2 – Dados de cada letra com relação a visibilidade dos marcadores e posição da mão.	34
Tabela 3.3 – Pesos dos marcadores para classificação.....	37
Tabela 3.4 – Distribuição das imagem de acordo com a letra e o classificador calculado.	45
Tabela 4.1 – Distribuição das letras com relação ao modelo considerado correto, sem a utilização do ângulo como parâmetro.	48
Tabela 4.2 – Distribuição das letras com relação ao modelo considerado correto, com a utilização do ângulo como parâmetro.	49
Tabela 4.3 – Distribuição das 260 letras com relação ao modelo do primeiro experimento, utilizando a busca automática dos marcadores.....	50
Tabela 4.4– Validação da classificação das 810 letras que geraram o modelo do terceiro experimento.....	53
Tabela 4.5– Validação da classificação dos 292 Sinais separados para teste.	54
Tabela 4.6– Validação da classificação das 540 letras que geraram o modelo do quarto experimento.	56
Tabela 4.7– Validação da classificação dos 236 Sinais separados para teste.	57
Tabela 4.8– Resultado reconhecimento das 776 imagens do quarto experimento comparadas com os padrões do terceiro experimento.....	58
Tabela 4.9– Resultado reconhecimento das 1102 imagens do terceiro experimento comparadas com os padrões do quarto experimento.....	58

1. INTRODUÇÃO

Este capítulo apresenta considerações gerais preliminares relacionadas à proposta desse projeto e os objetivos pretendidos.

Atualmente, a principal forma de interação do homem com os computadores para a entrada de dados é por meios de **periféricos** como *mouse*, joysticks e teclado. Entretanto, as novas aplicações cada dia mais complexas demandam novas formas de interação. De acordo com Bebis *et al*, 2002 [1], uma solução seria trazer os meios de **comunicação naturais** como fala e **gestos** para a interação homem-maquina. Neste trabalho abordaremos a integração por meio de gestos, com foco no reconhecimento da Língua Brasileira dos Sinais (Libras).

Cada vez mais o mundo vem se preocupando com a acessibilidade de deficientes físicos, bons exemplos disso são as rampas para cadeirantes, o sistema *closed caption* para deficientes auditivos e o código Braille que permite a leitura e escrita para cegos.

Para **auxiliar a comunicação** dos deficientes auditivos, segundo Sousa, 2012 [2], foi desenvolvida a língua gestual, uma língua de movimento e espaço, que utiliza as mãos, face e olhos. A língua gestual é a língua adotada pela comunidade surda e é a principal portadora da cultura das pessoas que a utilizam como forma de comunicação. É bastante comum confundir língua com linguagem. A linguagem gestual é um elemento para-linguístico, sendo utilizada para complementar a comunicação oral, contribuindo para uma maior expressividade da comunicação. Porém, a língua gestual deve ser encarada como uma língua humana, na medida em que obedecem a parâmetros linguísticos uniformes, como a arbitrariedade, a convencionalidade, a recursividade e a criatividade.

Acredita-se que hoje em dia ainda não exista nenhuma **solução comercial** que consiga reconhecer os gestos de uma língua gestual automaticamente e os converter em áudio ou texto, facilitando a comunicação entre surdos e ouvintes. O que existe atualmente é uma solução que consiste em um serviço de interlocução. O usuário faz sinais para um atendente, ou tradutor, que domina a língua de sinais, e esse se torna o canal de comunicação com o ouvinte. Deste modo, para um

deficiente auditivo poder se comunicar ele precisa passar todas as informações para um terceiro. São inúmeros os problemas que esse tipo de solução pode causar. O surdo perde a sua privacidade ao precisar se comunicar com o intérprete, caso ele deseje fazer um contato mais particular. Outro problema é que o usuário depende da disponibilidade dos atendentes, além do custo do serviço e da abrangência, que ainda não é disponível em todo Brasil.

A detecção de sinais gestuais depende de tecnologias específicas. Atualmente temos **duas abordagens gerais** para o reconhecimento dos sinais: abordagem baseada em visão e abordagem baseada em luvas instrumentalizadas (*datagloves*). Essas abordagens possuem suas vantagens e desvantagens. A solução com a luva instrumentalizada necessita que o usuário vista uma luva, reduzindo a praticidade, por que geralmente esse tipo de luva tem componentes eletrônicos acoplados que se comunicam com um computador. Por outro lado, possui uma maior precisão na captação dos gestos. Já a solução baseada em visão permite uma interação mais livre por não precisar vestir nenhum equipamento. Em contra partida, essa abordagem apresenta uma alta complexidade computacional devido ao processamento em tempo real de imagens.

1.1. Objetivo Geral

Desenvolver uma solução para a entrada de dados utilizando comandos gestuais, que seja capaz de capturar e reconhecer sinais feitos em Libras e os transformar em uma saída de texto válida.

Esse projeto visa contribuir para melhoria da comunicação de deficientes auditivos com as pessoas ouvintes, e como uma nova interface de interação direta com computador.

1.2. Objetivos Específicos

Construir um sistema que adquira imagens RGB e imagens de profundidade, através de um sensor Kinect e câmera de alta resolução, processe conjuntamente esses sinais e classifique as posturas manuais detectadas em letras do alfabeto manual da Libras. Como subproduto, um banco de dados com diversas amostras das 26 letras do alfabeto em Libras será criado.

Desenvolver estratégia para tratamento de imagens que melhore a técnica de reconhecimento dos sinais.

Geração de modelos para cada um dos sinais correspondente ao alfabeto de Libras.

A Classificação dos sinais, fazendo o seu reconhecimento com a taxa de acerto superior a 90%.

2. REFERENCIAL TEÓRICO

Neste capítulo será feita uma breve descrição dos conceitos envolvidos neste trabalho, além de uma análise de trabalhos similares e a apresentação das tecnologias disponíveis para a sua realização.

2.1. Libras

Segundo Oliveira, 2011 [3], a língua de sinais foi trazida ao Brasil em 1856 pelo professor francês Ernest Huet, deficiente auditivo que introduziu o alfabeto gestual francês. Baseado nesse alfabeto começou a surgir a língua brasileira de sinais. A língua de sinais surgiu na Europa no século XVI, onde teve início as primeiras metodologias voltadas ao desenvolvimento e comunicação dos deficientes auditivos. O instituto Nacional de Jovens Surdos, localizado em Paris na França, desenvolveu a metodologia para instruir jovens surdos, através de datilografia, alfabeto manual e gestos desenvolvidos por eles.

Para Pacheco, 2008 [4], a língua de sinais auxilia o deficiente auditivo no seu desenvolvimento e comunicação com outros surdos ou ouvintes. A língua de sinais não se baseia unicamente no alfabeto, são utilizados ainda gestos que podem significar palavras e até mesmo expressões inteiras. Porém essa língua não é universal. Cada país possui a sua própria língua de sinais. Não existe, portanto, uma padronização, podendo uma mesma palavra ser representada por gestos diferentes dependendo do país. No Brasil temos a Língua Brasileira de Sinais, conhecida como Libras, cujo alfabeto manual é mostrado na figura 2.1.



Figura 2.1 - Alfabeto em Libras (Língua de Sinais Brasileira) [6].

Porém, todo o processo, até que a língua de sinais fosse de fato reconhecida, passou por diversas dificuldades. Houve época que qualquer tipo de gesto, feito pelos surdos, era considerado algo anormal e fora dos padrões, chegando ao ponto de ser proibido o seu uso nas escolas. Isso fez com que os próprios deficientes se sentissem desestimulados a se comunicar, prejudicando o avanço e popularização da língua de sinais. Com o passar dos anos, a comunicação foi ganhando real importância e o deficiente auditivo pode se integrar mais e melhor com a sociedade.

Em Abril de 2002, foi sancionada a lei nº 10.436 que regulamenta a Língua Brasileira de Sinais (Libras), com isso os deficientes auditivos vêm ganhando espaço na sociedade. Mas ainda existe muito a ser feito para que a comunicação entre ouvintes e surdos seja plena. Em Dezembro de 2005, pelo decreto nº 5.626 foi regulamenta a lei que tornou Libras uma disciplina obrigatória na formação de professores nos cursos superiores.

A Língua Brasileira de Sinais possui **estrutura gramatical própria**, sendo um sistema linguístico legítimo que propicia ao surdo a integração com a sociedade. A Libras é desenvolvida através de gestos realizados por movimentos das mãos e expressões faciais, sendo extremamente complexa. Nesta língua, utiliza-se do

próprio corpo e seus movimentos de forma muito expressiva, demonstrando atitudes, comportamentos e sentimentos dos mais diversos tipos.

Quando se analisa os níveis fonológicos e morfológicos da Língua Brasileira de Sinais podemos apontar cinco parâmetros que constituem cada sinal. Os sinais são formados a partir da combinação do movimento das mãos com um determinado formato e em um determinado lugar, onde esse lugar pode ser uma parte do corpo ou um espaço em frente ao corpo.



Figura 2.2 Configurações de mão da LIBRAS[7].

Dá-se a denominação de **Configuração de Mão** (CM) para as formas das mãos, que podem ser da datilologia (alfabeto manual) ou outras formas com significados próprios, exemplos de configuração de mão podem ser visto na Figura 2.2. Alguns gestos possuem a mesma configuração de mão, porém diferenciam entre si devido a outros parâmetros, tais como movimentação e localização.

Ponto de Articulação (PA) é o lugar onde incide a mão configurada, podendo estar tocando alguma parte do corpo ou estar em um espaço neutro que vai do meio do corpo até acima da cabeça.

Os sinais podem ter **movimento** (M) ou não. O movimento é a movimentação das mãos ou do corpo durante o sinal, o que pode fazer com que mude completamente seus significados.

Orientação (O) é a direção em que o movimento é feito. Normalmente a sua inversão significa a ideia de oposição, contrário. Caso o gesto não tenha movimento não faz sentido se falar em orientação.

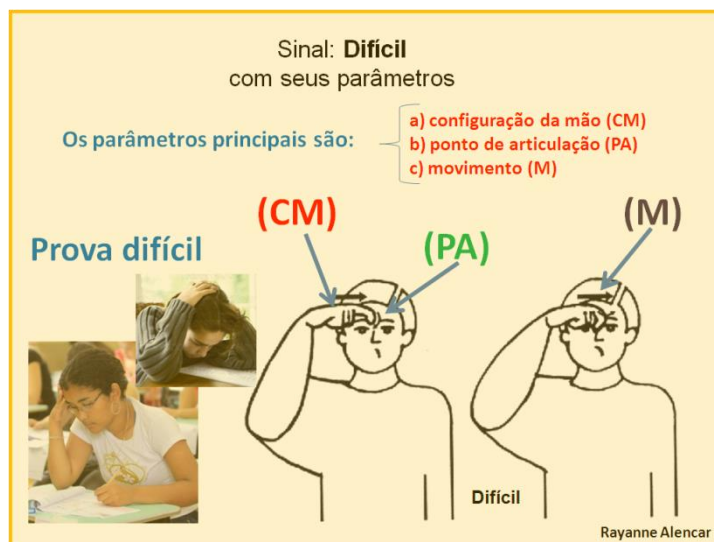


Figura 2.3 - Exemplo com os principais parâmetros[7].

Muitos sinais, além dos quatro parâmetros anteriores, têm em sua configuração um traço diferenciador através da expressão facial e/ou corporal, na Figura 2.3 é possível observar essa expressão. Podem existir gestos feitos apenas por meio desse parâmetro.

Ainda é muito forte a crença por parte da sociedade ouvinte que a língua de sinais não possua gramática, acreditando-se que ela seria composta apenas por mímica e pantomimas. Porém, o objetivo é fazer com que o interlocutor veja o objeto representado. É possível expressar conceitos abstratos na língua de sinais. Assim como os falantes de língua orais, é possível discutir filosofia, política, escrever poemas e peças teatrais, contar e inventar histórias. O fato de Libras ser um língua de modalidade espaço-visual leva a pensar que seja um língua exclusivamente icônica, embora existam muitos sinais icônicos.

O alfabeto manual é utilizado para soletrar manualmente palavras. A datilologia, ou soletração digital, é apenas um recurso utilizado pelos usuários das línguas de sinais. Não é uma língua nem representa a língua de sinais como um

todo, mas sim um código de representação das letras, logo o alfabeto é composto por 26 sinais.

2.2. Reconhecimento de gestos

Atualmente existem várias soluções no mercado que podem ser utilizadas para o reconhecimento das mãos, dentre as quais podemos citar o Kinect, câmera e marcadores coloridos, câmeras infravermelho, visão estéreo, radar laser e luvas com acelerômetros, cada uma dessas soluções será avaliada posteriormente com relação a vantagens, desvantagens e custos.

Apesar de todo esforço dos pesquisadores e dessa variedade de soluções, a interação homem-máquina ainda difere muito da interação homem-homem. A interação natural entre os seres humanos não envolve nenhum dispositivo, pois nós temos a habilidade de interpretar o ambiente com olhos e ouvidos. Idealmente espera-se que os computadores possam imitar essas habilidades com câmeras e microfones. Existe um número grande de pesquisadores trabalhando em soluções de rastreamento baseado em visão, com uma quantidade vasta de técnicas, tais como soluções baseadas em filtro de Kalman, detecção da cor da pele e modelagem 3D da mão.

2.3. Tecnologias disponíveis

Dentre as técnicas pesquisadas foi feita a análise de vantagens, desvantagens e estimativa de custo para cada uma das técnicas encontradas. Será feita uma breve descrição de todas as técnicas e a razão da escolha das que serão adotadas nesse trabalho.

2.3.1. Kinect

Utiliza um **canhão de luz Infra Vermelha (IR)** e uma câmera com filtro para o IR. Analisando a distância entre dois ponto é possível estimar a distância entre o ponto e a câmera. O sensor conta ainda com uma câmera colorida comum, com **resolução**

640x480 pixels e uma matriz de microfones. Este sensor possui um custo aproximado de R\$ 500 [www.walmart.com.br/].

Entre as suas vantagens podemos citar:

- Muito utilizado atualmente em pesquisas e desenvolvimento de produtos;
- Disponibilidade de vasto material para consulta;
- Preço acessível e fácil obtenção;
- Ambiente e ferramentas de trabalho disponibilizados pela própria fabricante Microsoft.

As principais desvantagens são:

- Câmera colorida com baixa resolução, para fins científicos mais específicos;
- Ambiente de desenvolvimento proprietário;

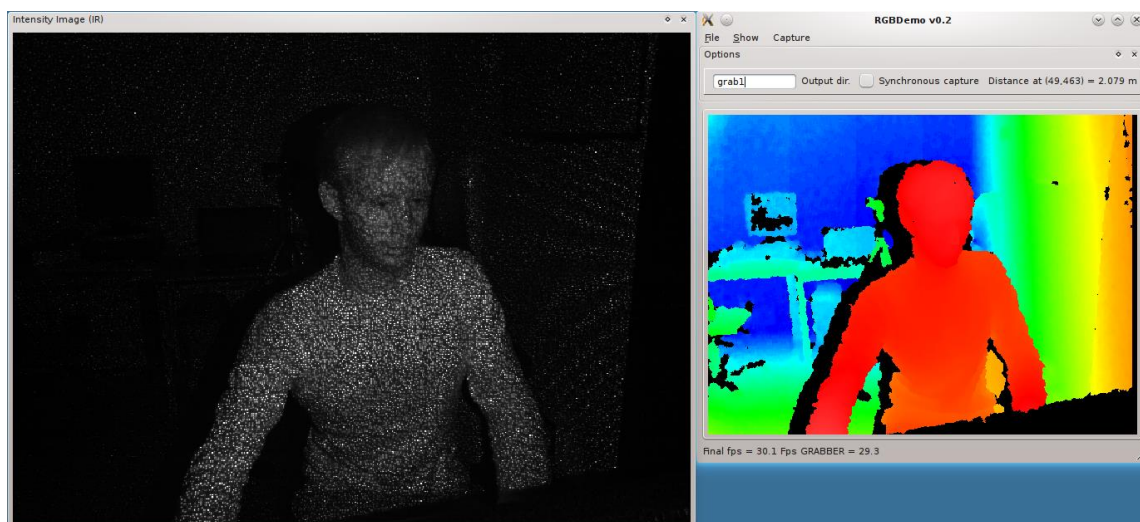


Figura 2.4 - Exemplo de imagens obtidas pelo sensor Kinect, retirada de [http://nicolas.burrus.name/index.php/KinectRgbDemoV4]

Na Figura 2.4 é possível ver um ambiente iluminado pelo projetor IR, e a direita a imagem de profundidade do mesmo ambiente quantizada em cores.

2.3.2. Câmera infravermelho (OMEX)

Utiliza também de um canhão de luz IR e uma câmera com filtro para a luz IR, analisando o tempo entre a emissão e a recepção do sinal entre dois pontos é possível determinar a distância utilizando um software proprietário da empresa

OMEK [omekinteractive.com]. Possui custo aproximado de R\$ 4.500,00 [http://www.digikey.com/].

Tem como vantagens:

- Variedade de *hardware* disponível;
- Utilizado atualmente em várias pesquisas científicas;
- Disponibilidade de vasto material para consulta;
- Ambiente e ferramentas de trabalho disponibilizado pela OMEK.

Suas desvantagens:

- Custo elevado;
- Necessidade de comprar o software da OMEK;
- Ambiente de desenvolvimento proprietário.



Figura 2.5 - Exemplo de imagem sendo obtida por câmera infravermelho, retirada de [http://www.epnc.co.kr/at1/view.asp?a_id=9458]

Na Figura 2.5 é possível observar uma demonstração do funcionamento da câmera fazendo o rastreamento de um alvo utilizando a câmera com infravermelho.

2.3.3. Marcadores coloridos

Utiliza de uma câmera de alta resolução para detectar marcadores de cores específicas, podendo seguir as rotas dos marcadores na imagem e reconhecer

padrões. Custo a partir de R\$ 300,00 [<http://www.fnac.com.br/>] dependendo da resolução e da taxa de captura.

Vantagens:

- Custo baixo;
- Grande variedade de hardware possível.
- Fácil fabricação e manutenção;
- Melhor resolução, que facilita a diferenciação dos gestos;
- Muitos trabalhos publicados na área.

Desvantagens:

- Muito influenciado por variações luminosas do ambiente;
- Requer a utilização de marcadores no corpo;
- Dificuldade para perceber alterações na profundidade.

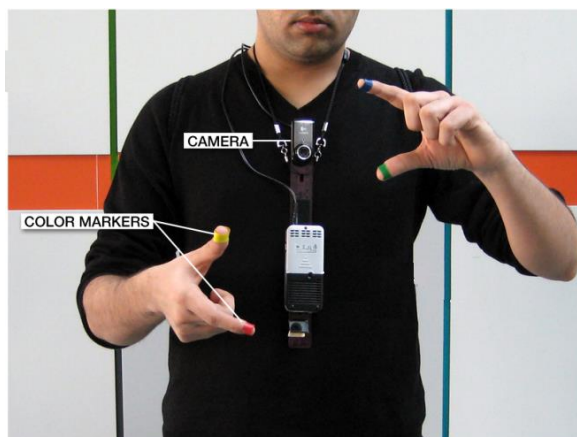


Figura 2.6 - Exemplo de marcadores coloridos, retirada de [<http://www.pranavmistry.com/projects/sixthsense/>]

A Figura 2.6 mostra um exemplo de marcadores coloridos, nesse caso foi utilizado para a localização dos dedos.

2.3.4. Visão estéreo

Utiliza um par de câmeras de boa qualidade, fisicamente posicionadas e separadas lateralmente por alguns centímetros. Através da análise das imagens é possível calcular a diferença na posição de um ponto e por triangulação calcular a

distância conjunto de câmeras ao ponto. Custo aproximado R\$400,00 [<http://www.ptgreystore.com/>].

Vantagens:

- Custo baixo e grande variedade de hardware possível;
- Fácil fabricação e manutenção;
- Melhor resolução, que facilita a diferenciação dos gestos;
- Vários trabalhos publicados na área;
- Boa percepção de profundidade em ambientes controlados.

Desvantagens:

- Muito influenciado por variações luminosas do ambiente;
- Requer ambientes estáticos de fundo para um melhor desempenho;
- Baixa qualidade dos dados de profundidade.

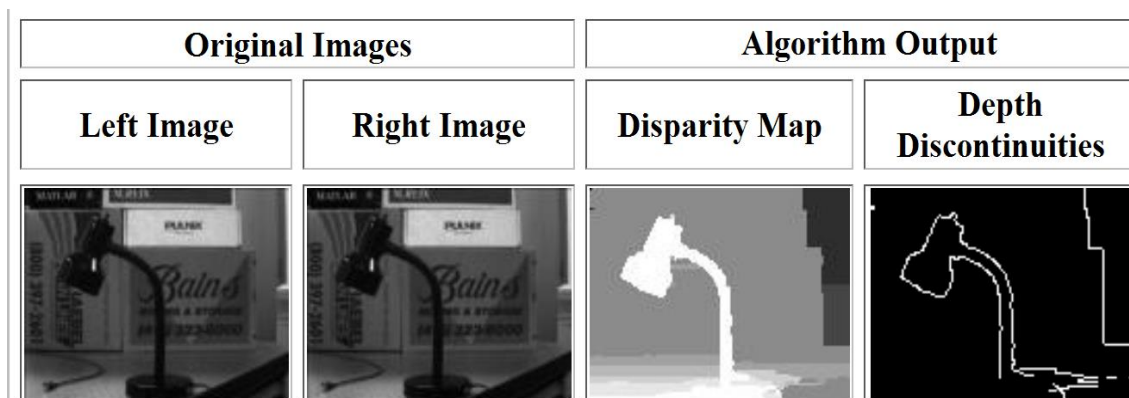


Figura 2.7 - Exemplo de visão estéreo, retirada de
[<http://opencvlib.weebly.com/cvfindstereocorrespondencebm.html>]

Na Figura 2.7 tem-se a detecção de profundidade dos objetos utilizando as imagens de duas câmeras, retirando-se o corte da imagem de profundidade para segmentar o objeto.

2.3.5. Radar laser

Utiliza feixes de luz infravermelha para calcular a distância baseado no tempo de retorno. Necessita fazer uma varredura vertical para poder gerar uma imagem 3D. Usualmente utilizado em digitalizadores 3D. Custo aproximado de R\$ 4.000,00 [<http://www.robotshop.com/en/laser-scanners-rangefinders>].

Vantagens:

- Não é influenciado por condições ambientais;
- Precisão das distâncias medidas.

Desvantagens:

- Custo Elevado e pouca variedade de hardware possível;
- Baixa resolução, que dificulta a diferenciação dos gestos;
- Pouca pesquisa em trabalhos semelhantes.

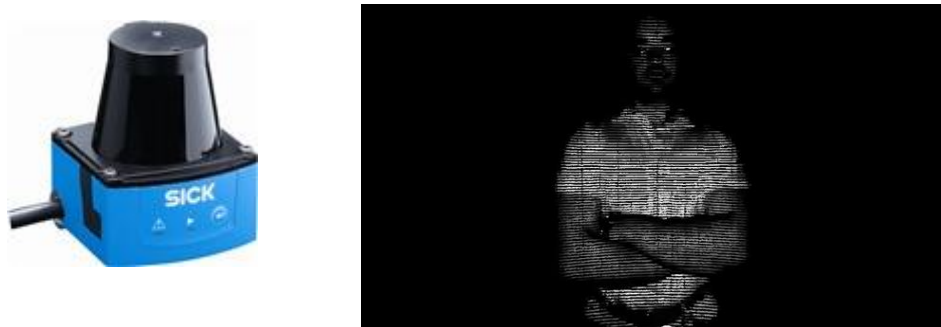


Figura 2.8 – Exemplo de radar laser (a) e imagem capturada (b), retiradas de
[<http://www.autonomoustuff.com/sick-tim310.html>]

A figura 2.8 (b) apresenta uma imagem de uma pessoa sendo capturada pelo radar laser e tendo a imagem formada com base nos dados de captura do laser.

2.3.6. Luvas instrumentalizadas

Utiliza de sensores de movimento como acelerômetros ou sensores de contração nos dedos, para obter dados de movimentação e posturas e assim comparar com padrões conhecidos. Custo de R\$ 250,00.

Vantagens:

- Custo baixo e variedade de hardware disponível;
- Grande quantidade de dados, que facilita a diferenciação de gestos parecidos;
- Sensores amplamente utilizados.

Desvantagens:

- Necessidade de utilizar um par de luvas;
- Necessidade de baterias e/ou cabos para as luvas;
- Difícil fabricação e manutenção.

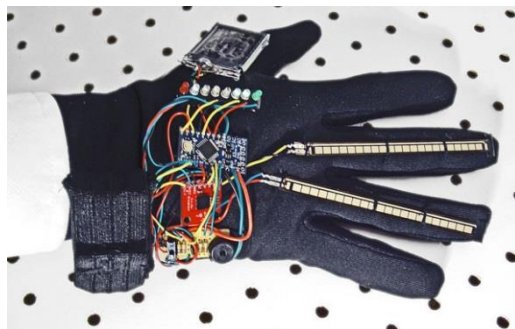


Figura 2.9 - Exemplo luva com sensores extensômetro e giroscópio, retirada de [http://grathio.com/2010/03/rock_paper_scissors_training_glove/].

A Figura 2.9 mostra uma luva equipada com extensômetros e sensores de giro e aceleração, a combinação destes dados fornece a informação da postura e movimentação da mão.

Com essa pesquisa foi possível concluir que nenhuma das soluções é capaz de atender sozinha a todas as necessidades que o reconhecimento de Libras exige. Então se buscou uma solução barata e que possibilitasse uma elevada taxa de reconhecimento e distinção de gestos. Assim, decidiu-se escolher pelo desenvolvimento de uma solução com a união de ideias anteriores. A proposta consiste em utilizar o Kinect juntamente com uma luva com marcadores coloridos que serão capturados por uma câmera com uma resolução superior à câmera do Kinect. Deste modo, os problemas inerentes da baixa resolução e dificuldade de modelagem precisa da postura manual são tratados através da imagem da câmera, e os problemas de detecção, localização e extração da mão são resolvidos pelos dados de profundidade fornecidos pelo Kinect.

2.4. Trabalhos anteriores

As novas tecnologias favoreceram o surgimento de muitos trabalhos inovadores que buscam por meio de visão computacional, uma melhor utilização da combinação das imagens RGB e dos dados de profundidade. Os dados de profundidade contribuem bastante para resolver problemas descritos em trabalhos anteriores, como por exemplo, os casos de segmentação das imagens e o tratamento de oclusão, tarefas que se implementadas apenas em imagens RGB demandam um grande custo computacional e algoritmos complexos.

Lamar et al, 2000 [9], propõem o uso de marcadores coloridos para o reconhecimento de posturas manuais do alfabeto da língua japonesa de sinais. Essa

pesquisa serviu de base para vários trabalhos por trazer uma proposta nova na captura e modelagem das informações da postura da mão. O trabalho utiliza uma câmera colorida e uma luva especialmente projetada para que cada dedo e a palma da mão tenham uma cor específica no espaço de cores RGB. Cada gesto é modelado por um vetor de 22 dimensões, sendo que cada quatro componentes é referente a cada dedo. O trabalho apresenta ainda uma rede neural *T-CombNET* treinada para reconhecer os sinais manuais. Os resultados obtidos para um conjunto de 42 gestos distintos indicam uma taxa de aproximadamente 90% de correto reconhecimento. O sistema proposto mostrou-se robusto o suficiente para reconhecer o mesmo sinal a distâncias diferentes da câmera e com *background* natural da imagem.

O trabalho de Silva *et al.*, 2014 [10], tem como objetivo o reconhecimento do alfabeto da língua americana de sinais, fazendo uso apenas da imagem de profundidade fornecida pelo sensor Kinect, sem a utilização de luvas ou marcadores. A ideia geral é construir uma arquitetura simples de casamento de modelos, onde o reconhecimento é feito comparando-se um determinado caso de teste contra um conjunto de imagens de profundidade em um banco de dados. A imagem de profundidade é preparada para ser realizado o procedimento de *Iterative Closest Point* (ICP). O ICP é um algoritmo de alinhamento dominante na literatura que tem como objeto recuperar uma solução de qualidade para o movimento euclidiano entre duas formas de pontos 3D. O ICP é feito em um subconjunto do banco de dados que é formado pelo valor de métrica média para cada letra, o reconhecimento é alcançado pela letra com o melhor valor médio comparado ao valor da amostra de teste. Com um percentual de 99% de comparações corretas o trabalho também se mostrou uma boa solução, porém com algumas limitações de distância para os sensores e grandes variações na configuração de mão no mesmo gesto.


Outro trabalho inspirador foi realizado por Yuan Yao *et al.*[15], que também faz o uso de luvas coloridas, porém além de pintar os dedos, foi utilizada a técnica de pintar as pontas dos dedos de cor diferente do resto do dedo, além de pintar palma e dorso das mãos de cores diferentes. Utiliza como técnica de treinamento, um banco com um grande conjunto de imagens e a partir dele gera as estimativas das posições das mãos. Para fazer o reconhecimento utilizou a forma da mão, a localização e orientação. Usando uma câmera de profundidade para fazer a

segmentação da mão, reduzindo a área de análise e as perturbações que o *background* poderia gerar. As taxas de acerto foram relativamente baixas durante os experimentos.

Esses três trabalhos mostram boas soluções para o problema do reconhecimento de alfabetos manuais. Neste trabalho, busca-se a utilização das ideias principais de cada uma das pesquisas anteriores, buscando uma solução alternativa para os problemas não resolvidos.


3. METODOLOGIA PROPOSTA

Este capítulo descreve as ferramentas e as técnicas que são utilizadas para atingir os objetivos deste trabalho.

Para a tarefa de reconhecer os gestos utilizando a combinação das técnicas para tratamento de imagens coloridas e de profundidade, previu-se uma alta demanda por processamento devido à grande quantidade de dados para ser tratados simultaneamente. Usou-se então um computador com processador Intel Core i7, com 8GB de memória RAM, HD de 500GB, além de uma placa de vídeo dedicada. O sistema operacional de trabalho foi o Windows 8 devido à disponibilização do *software development kit* (SDK) do Kinect pela Microsoft. 

3.1. Sensor Kinect

O sensor Microsoft Kinect foi anunciado em junho de 2009 e teve seu lançamento comercial realizado em novembro de 2010 para o console Xbox 360. Seu nome origina-se das palavras cinética e conexão em inglês.

Uma das características principais do Kinect é a câmera, que permite ao usuário interagir com o console sem a necessidade de controles, através de uma interface natural utilizando gestos. A técnica de mapeamento 3D do ambiente utilizada no Kinect é conhecida como *time-of-flight* (TOF) que se baseia no princípio do eco. Um pulso ultrassônico ou eletromagnético é emitido e mede-se o tempo demandado pelo pulso atingir um objeto e retornar. Conhecendo a velocidade do som e da luz no ar pode-se calcular a distância entre o objeto e o emissor. 

Especificações:

- Campo de visão (horizontal, vertical e diagonal) 58° H, 45° V, 70° D;
- Resolução imagem VGA de 640 x 480 pixels;
- Resolução espacial x e y de 3 mm e de profundidade de 1 cm;
- Taxa de quadros máxima de 60FPS com resolução de 320x240;
- Taxa de quadros máxima de 30FPS com resolução de 640x480;
- Alcance de operação de 0,8m a 3,5m;

- Interface de dados USB 2.0;
- Recomendado para ambientes fechados sob qualquer tipo de iluminação.

O dispositivo fica sobre uma base com eixo-motorizado, permitindo que o sensor se mova de acordo com a necessidade do usuário, conforme mostrado na figura 2.10. Tendo com principal exigência que o usuário esteja a uma distância entre 1,2 metros e 3 metros para que possa ser obtido o melhor resultado.



Figura 3.1 - Sensor kinect e suas funcionalidades, retirada de
 [http://xboxconsole.blogspot.com.br/2012/03/kinect-for-microsoft-xbox-360.html]

A área útil para o usuário capturada pelo dispositivo é de 6m², em um campo de visão angular de 57° na horizontal e 43° na vertical. O eixo motorizado que movimenta a cabeça do sensor pode inclinar 27° para cima ou para baixo.

O Kinect pode funcionar em qualquer condição de iluminação devido à luz que utiliza para fazer o rastreamento tem como fonte um projetor infravermelho. Essa luz infravermelha além de estar fora do espectro da luz visível ainda carrega informação de polaridade o que evita a interferência de outras fontes de IR. Com isso o sensor de captura CMOS faz a análise do ambiente que foi iluminado pela luz infravermelha semelhante a uma câmera infravermelha convencional.

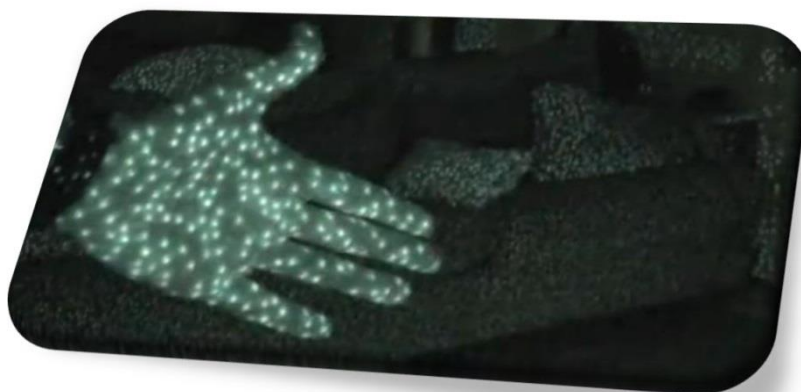


Figura 3.2 - Ambiente iluminado pelo projetor IR do Kinect, retirada de
[<http://gamerant.com/kinect-night-vision-video-dyce-51156/>]

A figura 2.11 apresenta uma imagem, feita por uma câmera IR, do ambiente ao ser iluminado pelo projetor de luz infravermelha. O cálculo da distância entre o objeto e a fonte, baseia-se no raio de retorno, quanto menor é a distância, mais intensidade luminosa tem o ponto.

Então, o Kinect basicamente possui um sensor de profundidade que emite o laser infravermelho e combinado com o sensor monocromático CMOS para infravermelho captura dados de vídeos em 3D.

3.2. Câmera

A câmera escolhida é um modelo comercial da Microsoft, LifeCam HD-5000 (Figura 3.3), por possuir uma alta resolução de imagem e ângulo de abertura bastante abrangente que são as principais exigências para obtenção de boas imagens para análise.



Figura 3.3 - Modelo da câmera utilizada, retirada de [<http://www.microsoft.com/hardware/pt-br/p/lifecam-hd-5000>]

Especificações:

- Sensor CMOS;
- Resolução de vídeo de 1280 X 720 pixels;
- Taxa de frames de 30 frames por segundo;
- Ângulo de abertura de 66°;
- Autofoco e *range* de 15 cm até o infinito;
- Imagem 16:9 *widescreen*;
- Cores em 24-bits.

Com essas especificações a câmera se mostrou adequada para o trabalho de reconhecer gestos estáticos, porém com a taxa de 30 quadros por segundo se tornou inviável o tratamento dos gestos dinâmicos, devido aos movimentos ficarem borrados em grande parte dos quadros durante a movimentação das mãos.

3.3. Luvas

Como mencionado anteriormente, foi escolhida a utilização de luvas para reduzir a influência prejudicial no rastreamento dos marcadores causada pelos diferentes tons de pele, e pela própria diferença entre palma e dorso da mão.

Foram testados vários modelos de luva, com o objetivo de escolher a mais ergonômica e que não atrapalhasse os movimentos, permitindo o abrir e fechar dos dedos e seus cruzamentos. A luva deve ser confortável, não aquecer em demasia, que possua coloração homogênea e a mais distinta possível das cores dos marcadores. Entre os modelos testados estavam luvas de borra de uso geral, normalmente utilizada para limpeza, luva de lã para trabalhos pesados e luva de Lycra utilizada como segunda pele.



Figura 3.4 - Luvas de Lycra utilizadas.

Escolhemos a luva de Lycra, por ser mais confortável, mais leve, restringir menos os movimentos e ter uma coloração clara e distante das cores dos marcadores.

3.4. Marcadores

Escolhemos usar marcadores coloridos nas pontas dos dedos para, inicialmente, facilitar o rastreamento dos dedos na imagem. Posteriormente foi feito um estudo das posturas manuais do alfabeto em Libras buscando os dedos que são visíveis em cada sinal e se os marcadores escolhidos ficariam também visíveis.

Decidiu-se, então, colocar os marcadores em cada falange distal de todos os dedos, contornando-a completamente. Assim, torna-se possível detectar a presença do marcador tanto em gestos com a palma ou com o dorso da mão.

As cores utilizadas foram as mais distantes entre si considerando o espaço de cores RGB. Ou seja, as próprias componentes vermelho, verde e azul e as cores compostas amarelo e magenta, mostradas na Figura 3.5.

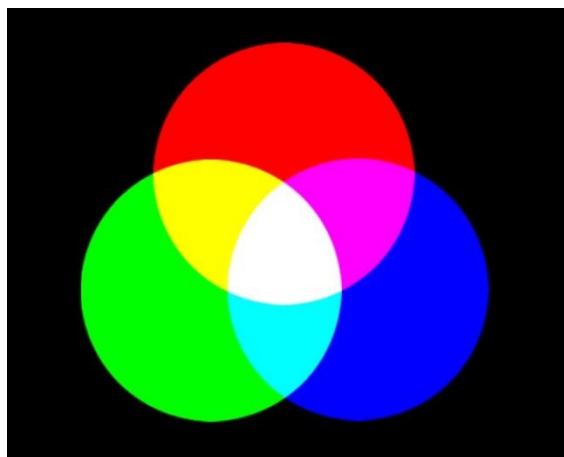


Figura 3.5 componentes de cores RGB.

Os primeiros testes foram feitos com fitas isolantes nas cores pretendidas. Porém o resultado obtido não foi muito bom devido à fita não se moldar muito bem aos dedos. Assim, decidiu-se pintar os marcadores diretamente na luva, o que apresentou um ótimo resultado. Também se acreditava que fosse necessária a diferenciação do dorso com a frente da mão. **Nos primeiros modelos foi pintado de preto o dorso da mão, conforme mostrado na Figura 3.6, porém com os testes foi visto que não existia esta necessidade.** A cor do dorso foi removida, pois foi possível fazer a diferenciação entre frente e dorso da mão com base no ângulo e na ordem que os marcadores aparecem, mostrados na Figura 3.7.



Figura 3.7 - Luva inicial com o dorso pintado.



Figura 3.6 Luva final com os marcadores pintados.

Foram definidas as cores para cada um dos dedos: vermelho polegar, verde indicador, amarelo médio, rosa anelar e azul mínimo, para facilitar a compressão que será apresentada posteriormente.

3.5. Softwares e bibliotecas utilizados

Nesta sessão serão descritos os softwares e bibliotecas utilizadas neste trabalho.

3.5.1. Visual Studio

Microsoft Visual Studio é um pacote de programas da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e J# (J Sharp). Também é um produto para o desenvolvimento na área web, usando a plataforma do ASP.NET. As linguagens com maior frequência nessa plataforma são o VB.NET (Visual Basic.Net) e o C#.

Foi decidido trabalhar com Visual Studio devido a sua maior integração com o SDK do Kinect, por estarmos trabalhando em um ambiente Windows e devido às boas experiências anteriores na integração com a biblioteca OpenCV.

3.5.2. OpenCV

A *OpenCV* é uma biblioteca *open source* de ferramentas e algoritmos para processamento de imagens desenvolvida inicialmente pela *Intel*. Esta biblioteca é composta por algoritmos otimizados, de tal forma que utiliza os recursos do sistema para atingir o menor custo computacional [11]. Essa biblioteca atualmente possui mais de 500 funções de processamento com aplicações dentre as quais se destacam: operações entre imagens, filtros, transformações morfológicas, calibração de câmeras, *tracking*, estimação de pose, reconhecimento e identificação de faces, gestos e objetos [12].

Escrita em C, C++ e Python para os sistemas operacionais Linux, Windows e Mac OS X, foi escolhida a *OpenCV* como ferramenta principal para o desenvolvimento dos algoritmos de visão do sistema. As bibliotecas adotadas foram em C e C++, devido a maior familiaridade com as linguagens.

3.5.3. Kinect SDK

A SDK do Kinect é uma ferramenta para programação para desenvolvedores. Ela possibilita fácil acesso a todos os recursos oferecidos pelo Microsoft Kinect conectados a computadores com sistema operacional Windows.

Inclui drivers, API ricas para os sensores de profundidade e webcam, documentos para instalação, e materiais auxiliares. A SDK permite desenvolvimento de aplicações com linguagens do .net framework (C#, Vb.net, C++)[13].

A SDK possui a possibilidade de captura de dados da câmera de profundidade, também captura a webcam e microfone, nesse trabalho não foi utilizado o microfone. Podemos também fazer a captura de dados do corpo, que consiste em rastrear as principais juntas do corpo de uma pessoa dentro do campo de visão do Kinect, possibilitando de forma fácil a criação de soluções baseadas na localização dos membros superiores do usuário.

3.6. Integração câmera e Kinect

Ao utilizar o Kinect, que fornece uma imagem com os dados de profundidade, e a câmera, que fornece uma imagem colorida para detecção dos marcadores; foi preciso fazer uma correspondência entre as duas imagens. O objetivo é que o uso conjunto das duas imagens referentes ao mesmo gesto possa retornar as informações necessárias.

Inicialmente se posicionou a câmera de alta resolução sobre o Kinect, alinhada com a câmera RGB do Kinect, supondo que já seria suficiente para a existência de uma boa correlação entre a imagem colorida da câmera e a de profundidade do Kinect. Porém o que se observou é que além do desalinhamento vertical devido à câmera estar ligeiramente acima do Kinect, as imagens tinham os ângulos de abertura diferentes o que dificultava muito o casamento correto das duas imagens. Então ficou evidente a necessidade de se fazer uma transformação que levasse a uma boa correlação entre as duas imagens.



Figura 3.8 Montagem câmera e sensor kinect.

3.7. Homografia

Homografia é uma transformação projetiva planar que mapeia pontos de um plano para outro plano. Este processo é ilustrado na Figura 3.9, em que o ponto x no plano π é mapeado para seu o ponto correspondente x' no plano π' . Este mapeamento linear de pontos pode ser escrito em coordenadas homogêneas como $x'_i = Hx_i$ em que H é a matriz de Homografia que define o mapeamento de um conjunto de pontos correspondentes $x_i \leftrightarrow x'_i$ entre dois planos.

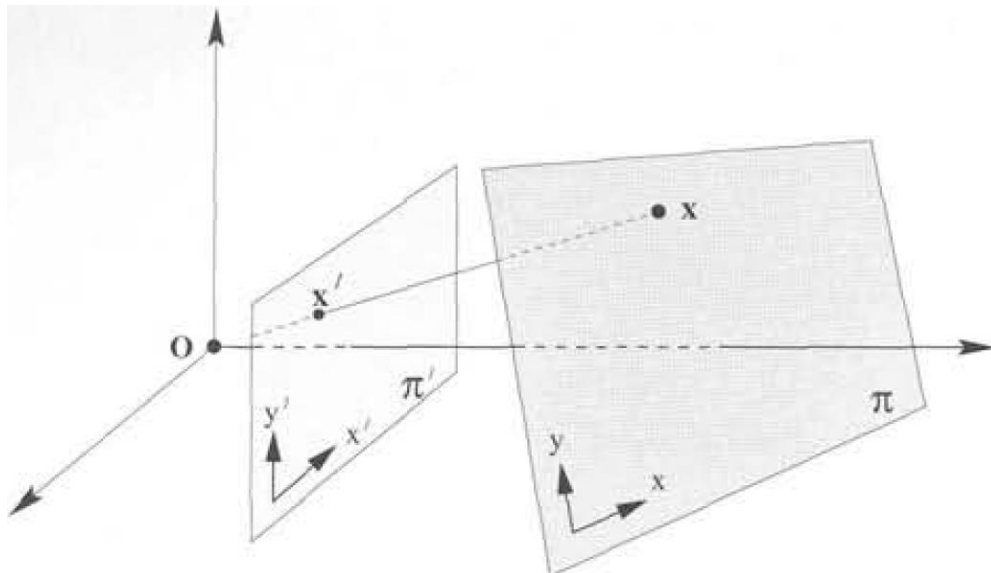


Figura 3.9 - Mapeamento entre planos.

O cálculo da matriz de homografia foi feito através da função *findHomography* da biblioteca OpenCV. Esta função estima a matriz de homografia através de um método não descrito na documentação, recebe como entrada o conjunto de pontos

da imagem base no plano π e os pontos equivalentes na segunda imagem no plano π' , calculando e retornando a matriz de homografia H . Quanto maior o número de pontos fornecidos e a distância entre os marcadores melhor será a aproximação.

Inicialmente, tentou-se fazer a homografia entre a câmera colorida do Kinect e a câmera HD adicional. A estimação da matriz de homografia H foi feita com sucesso. Porém ao se aplicar esta matriz H sobre a imagem da câmera HD, e verificar a sua sobreposição com o esqueleto obtido da câmera de profundidade, ficou visível que o casamento entre as imagens coloridas do Kinect e a de profundidade não pode ser feita diretamente por meio de homografia. Foi necessário então fazer a transformação da imagem da câmera HD para a imagem de profundidade.

Como a imagem de profundidade depende da distância entre os objetos e o sensor de captura, e a câmera HD não disponibiliza qualquer a informação de distância, marcadores para os dois modelos de sensores foram fixados no ambiente. Foram adicionados marcadores coloridos espaciais, compostos de quadrados de papel com dimensão 4x4 cm suficiente para que sejam detectados pelo sensor de profundidade e também na imagem colorida, conforme mostrado nas Figuras 3.3 e 3.4.



Figura 3.10 - Imagem dos marcadores (rosa claro) obtida pela câmera.



Figura 3.11 - Imagem dos marcadores obtida pelo Kinect

A partir dessas duas imagens foram retirados os quatro pontos da imagem base e os quatro pontos na imagem de destino, a marcação dos pontos foi feita manualmente. Na imagem colorida os pontos de base são os marcadores rosa. Os quatro pontos na imagem de destino são os pontos cinza claros na imagem de

profundidade. Posteriormente aplicou-se a matriz de transformação H na imagem colorida tendo como resultado a imagem da figura 3.5;

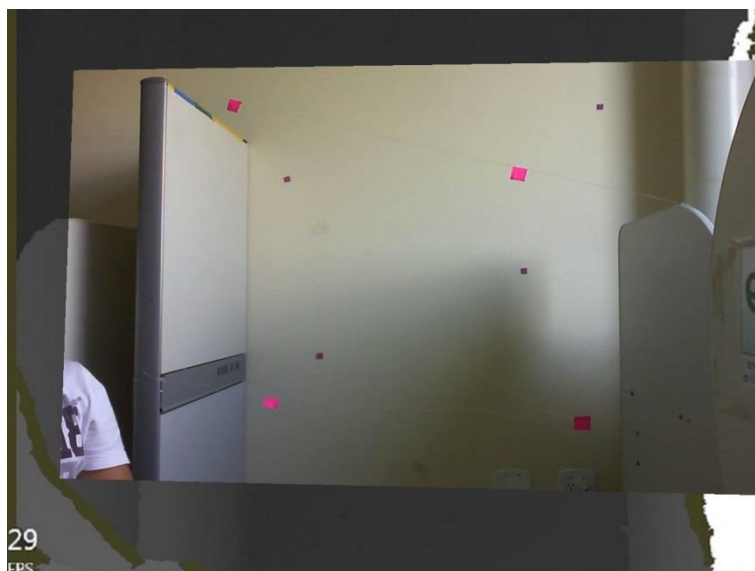


Figura 3.12 - Homografia entre imagem colorida e de profundidade com a colorida sobreposta à imagem de profundidade.

Essa transformação homográfica também é usada para transformar a imagem de profundidade para o plano da imagem colorida. A imagem da câmera HD possui maior resolução e por isso não sofre perda de qualidade significativa ao ser ampliada ou reduzida para se adequar à imagem de profundidade.

Obteve-se assim uma correspondência adequada entre os pixels da imagem de profundidade e da imagem colorida, facilitando a etapa de localização dos marcadores e remoção do fundo, como será visto posteriormente.

3.8. Isolamento da mão

Para a tarefa de isolar a mão foi utilizada a função `Nui_DrawSkeleton` da API do Kinect. Essa função tem como objetivo estimar as juntas do esqueleto do usuário, ou seja, identifica a localização da cabeça, do pescoço, dos ombros, cotovelos, e mãos na imagem de profundidade. Adaptou-se o código desta função de modo a enviar a localização da mão esquerda (ou direita) do usuário para o programa de análise das imagens, conforme visto na figura 3.6.

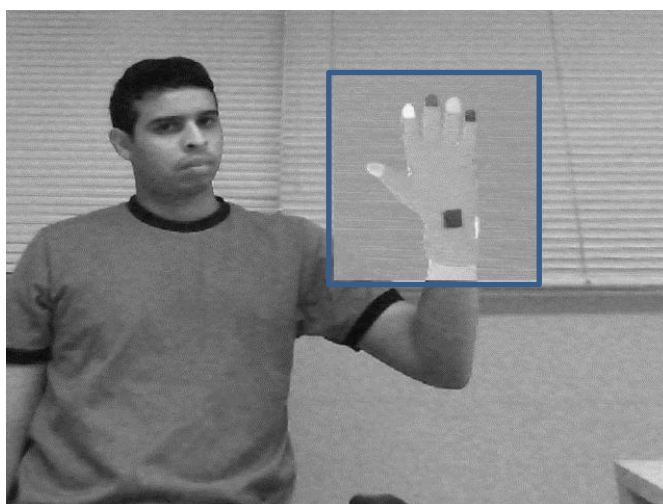


Figura 3.13 - Demonstração do isolamento da mão esquerda.

Aplicou-se a matriz de homografia, já calculada, para estimar a posição do centro da mão na imagem colorida HD a partir da sua posição na imagem profundidade.

Inicialmente planejou-se utilizar tamanhos variáveis na área entorno da mão que seria de tamanho variável de acordo com o gesto, isso para tentar reduzir a influência que o fundo da imagem pode ter na busca dos marcadores. Com a estratégia de remoção de fundo que será vista a seguir, foi definido que a área de análise seria uma imagem de 240x240 pixels, capaz de cobrir todos os gestos para uma distância entre um e dois metros.

3.9. Remoção do fundo da imagem


Os testes mostraram que mesmo reduzindo a área de análise, ainda sim a busca pelos marcadores era facilmente perturbada por objetos ou mesmo o fundo da imagem. Então foi preciso desenvolver uma técnica para remoção do fundo da imagem, isolando apenas a mão.

Por meio da imagem de profundidade e sobre ela aplicada a homografia com a imagem colorida, tem-se um bom alinhamento e podemos usar a imagem de profundidade como máscara para a imagem colorida.

A imagem de profundidade possui os dados em centímetros, ou seja, cada 1 cm de distância para o sensor corresponde a 1 nível a mais no pixel, então a imagem de

profundidade que é escala cinza variando de 0 a 255 o nível do pixel, cobre uma distância de 2,5 metros. Porém essa resolução não é a ideal para extrair com maior riqueza os detalhes das mãos e dos marcadores.

Foi adicionado ao código da função *CSkeletalViewerApp::Nui_ShortToQuad_Depth()* do SDK do kinect uma análise adicional, onde reescalou-se os pixels, deixando de ser o nível 255 próximo à câmera, passando a ser o nível 255 o pixel mais próximo capturado. Os pontos a mais de 25 cm desse pixel mais próximo recebem o nível 0. Partiu-se do princípio que, quando fazemos gestos, as mãos são as partes mais avançadas do corpo com relação ao sensor. O sensor tem a resolução em milímetros, porém utiliza centímetros para facilitar a visualização em uma única imagem. Mas para esse trabalho, é interessante ter a maior resolução na área das mãos. Como um gesto dificilmente pode ocupar um espaço maior que 25 cm, com uma única imagem é possível capturar a mão com o grau de riqueza 10 vezes maior do que a imagem original disponibilizada pelo Kinect.

Deste modo obtém-se uma imagem em escala cinza de 0 a 255. Como já se tem a área do recorte da mão, o mesmo corte também é feito nessa imagem. Porém o sensor de profundidade não tem uma resolução muito boa na extremidade dos dedos, perdendo com facilidade essa região em alguns frames. Para isso decidiu-se capturar um conjunto de 6 frames e usar a média entre eles. Com isso mesmo perdendo a região dos dedos em algum desses frames, a informação ainda é preservada. 

Utilizando essa imagem como máscara para imagem colorida, a região não detectada como mão é removida da imagem. Temos assim uma remoção de fundo que auxilia na redução de interferências por objetos no fundo da imagem.



Figura 3.15 - Imagem de profundidade original

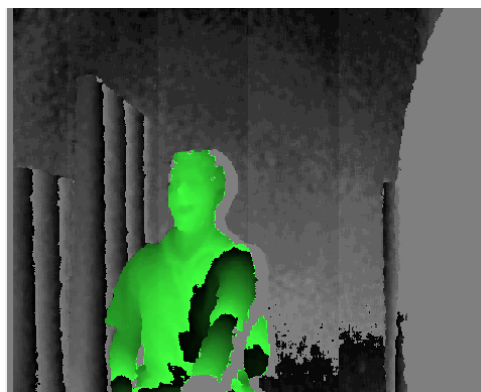
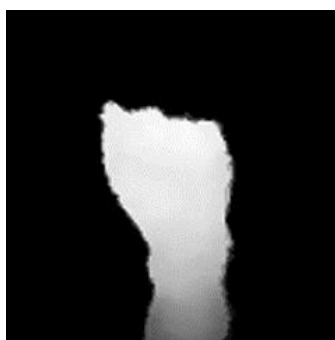


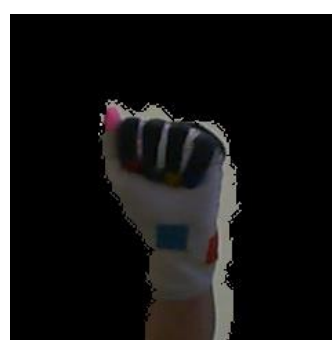
Figura 3.14 - Imagem de profundidade com resolução dez vezes maior.



(a)



(b)



(c)

Figura 3.16 - Etapas de remoção do fundo. (a) Imagem original colorida, (b) imagem de profundidade e (c) resultado da segmentação.

3.10. Espaço de cores

Para fazer as buscas dos marcadores precisamos escolher um espaço de cores. Como as cores selecionadas foram as três principais do espaço RGB, vermelho, verde, azul, ainda o amarelo e o magenta, então se decidiu usar para análise o espaço RGB. Após alguns testes foi detectado que alguns marcadores, devido ao gesto, podiam sofrer alguma variação nas suas componentes RGB. Então decidiu-se também fazer análise no espaço YCrCb (Y luminância, Cr cromaticidade do vermelho e Cb cromaticidade do azul) por ser menos sensível à variações de iluminação.

Mesmo utilizando dois espaço de cores, ainda sim devido a variação de iluminação ambiente, os valores das cores sofria alteração, para isso foi feita a normalização dos 2 espaços. Cada pixel da imagem sofreu alteração de suas 3 componentes RGB e YCrCb pelas equações a seguir.

$$SomaRGB = \sqrt[2]{R^2 + G^2 + B^2} \quad (1)$$

$$R' = 255 \frac{R}{SomaRGB} \quad (3)$$

$$G' = 255 \frac{G}{SomaRGB} \quad (5)$$

$$B' = 255 \frac{B}{SomaRGB} \quad (7)$$

$$SomaYCrCb = \sqrt[2]{Y^2 + Cr^2 + Cb^2} \quad (2)$$

$$Y' = 255 \frac{Y}{SomaYCrCb} \quad (4)$$

$$Cr' = 255 \frac{Cr}{SomaYCrCb} \quad (6)$$

$$Cb' = 255 \frac{Cb}{SomaYCrCb} \quad (8)$$

Com isso obteve-se uma maior fidelidade nas cores dos marcadores, sendo possível criar o padrão para cada marcador de acordo com os dois espaços de cores, semelhante ao utilizado por **François Malric [14]**.



Figura 3.18 - Imagem RGB original.



Figura 3.17 - Imagem RGB normalizada.

3.11. Localização dos marcadores

Com as cores definidas devido às normalizações, necessita-se localizá-las na imagem. Para isso definiu-se um padrão para cada marcador, que consiste nos três valores no espaço RGB e os três do espaço YCrCb. Definiu-se ainda as margens aceitáveis para cada um dos 5 marcadores. A Tabela 3.1 apresenta os valores de cada componente e as margens aceitáveis para cada componente de cor.

	R	Margem R	G	Margem G	B	Margem B	Y	Margem Y	Cr	Margem Cr	Cb	Margem Cb
Vermelho	237	10	71	18	66	20	85	15	214	12	107	12
Verde	101	20	187	12	143	17	161	7	93	15	122	12
Amarelo	206	14	146	12	36	20	110	18	180	14	76	16
Rosa	209	20	98	16	108	20	112	15	185	20	122	12
azul	89	20	151	14	186	20	157	7	91	17	152	16

Tabela 3.1 – Valores das componentes RGB e YCrCb e suas margens.

Então iniciou a análise nas imagens coloridas recortadas da mão com a remoção do fundo, buscando os pixels com as três componentes de cada espaço de cores com o erro dentro da margem estipulada. Este procedimento foi aplicado para os cinco marcadores, caso se encontre as três componentes do mesmo espaço de cores dentro da margem, inicia-se a busca por esta cor nos 8 pixels que fazem fronteira com o pixel detectado. Deste modo, uma vez validado o pixel como reconhecido, salva-se sua posição x e y da imagem e conta-se o número de pixels de cada cor encontrado. Após a análise de toda a imagem calculam-se os centroides de cada um dos marcadores visíveis na imagem (Figura 3.19).

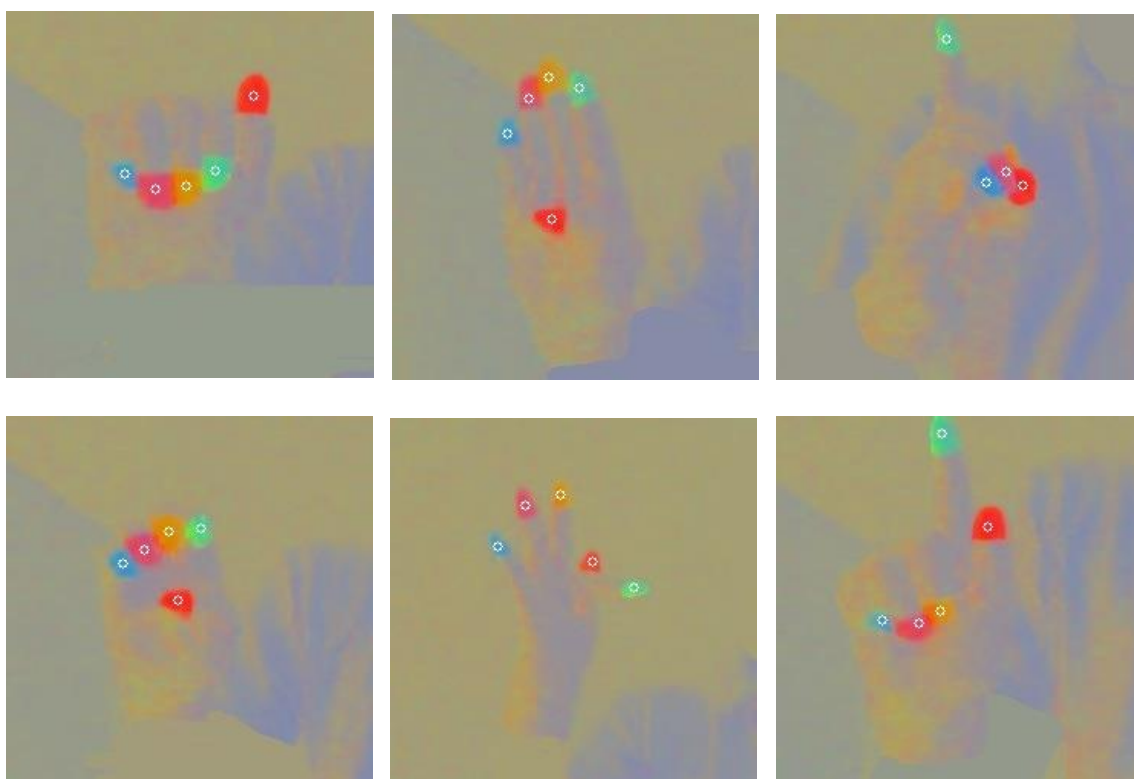


Figura 3.19 - Exemplo de sinais e os centroides de cada marcador sinalizado.

Para melhorar a precisão, foram utilizados os dois espaços de cores, RGB e YCrCb. Como a normalização de cada cor foi feita separadamente e a margem de erro também difere, mesmo um espaço sendo uma transformação do outro, conseguimos detectar a mesma cor em pontos diferentes dos marcadores, conforme apresentado na Figura 3.20.



Figura 3.20- Letra E: busca dos marcadores no espaço RGB

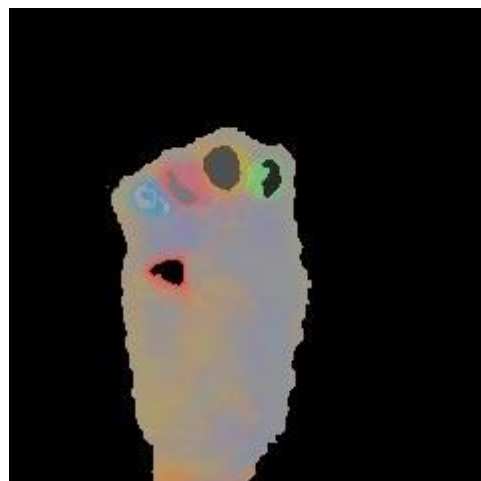


Figura 3.21- Letra E: busca dos marcadores no espaço YCrCb

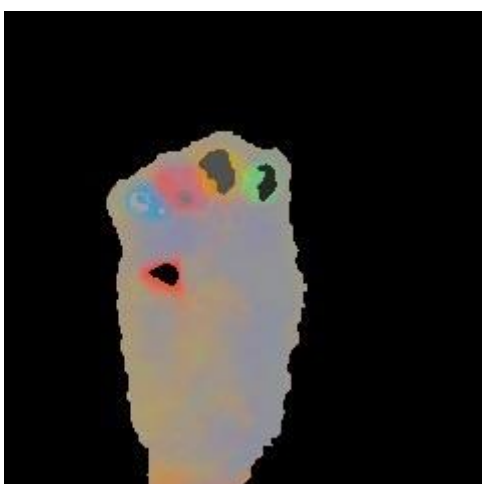


Figura 3.22 - Letra E: interseção dos pixels detectados em RGB e YCrCb

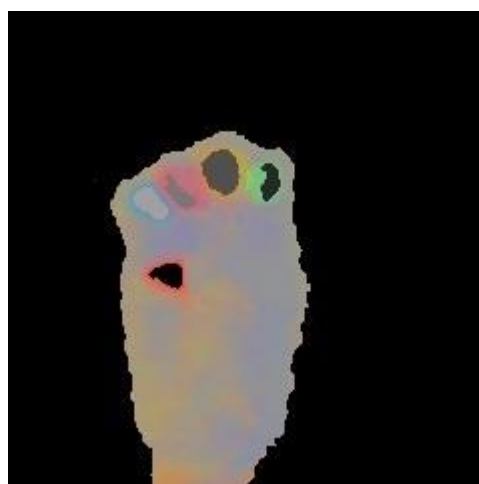


Figura 3.23- Letra E: união dos pixels detectados em RGB e YCrCb

É possível observar a diferença entre as imagens da interseção dos marcadores no espaço RGB e YCrCb nas figura 3.22. A imagem da união dos espaços é apresentada na Figura 3.23. É possível constatar que na união dos espaços o número de pixels é maior para cada marcador. Os pixels reconhecidos são regiões em escala de cinza presentes no interior de cada um dos marcadores. Este procedimento auxilia a encontrar o centroide correto e reduz o risco de perder algum marcador devido à sua oclusão parcial.



3.12. Escolha do alfabeto

Para fazer os testes de validação da técnica de reconhecimento proposta é preciso definir o espaço amostral desejado. Como o objetivo é reconhecer os gestos

em Libras, foi escolhido o alfabeto manual, pois a partir dele é possível soletrar qualquer palavra, além de ser formado por 26 diferentes sinais manuais, compondo um espaço suficientemente grande para os testes.

O alfabeto em Libras possui alguns sinais com movimento, é o exemplo das letras “H”, “J”, “K”, “X” e “Z”. Para essas letras foi definida a configuração final da mão obtida ao final do gesto como a postura a ser reconhecida.

De posse da luva e de tutorial de como fazer cada uma das letras em Libras iniciamos os estudos tentando prever como seria o comportamento de cada marcador para cada uma das letras. **Classificamos a possibilidade de cada marcador aparecer em todos os gestos, podendo variar de sempre visível, pouco visível e ocluso.** Também foi **avaliada a posição da mão que poderia ser classificada em frente, verso, misto e lado.** Com esses dados foi gerada a tabela 3.2.

		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	z	y	z
Rosa	visível	ok	ok	ok	ok	ok	ok	ok		ok		ok	ok			ok	ok	ok	ok	ok	ok	ok	ok		ok	ok	ok
	pouco																										
	ocluso								ok		ok			ok	ok									ok			
Vermelho	visível	ok	ok		ok	ok	ok	ok	ok			ok	ok	ok	ok		ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
	pouco									ok						ok											
	ocluso			ok							ok																
Verde	visível	ok	ok			ok	ok	ok	ok			ok	ok	ok	ok		ok		ok		ok	ok	ok	ok		ok	
	pouco			ok	ok											ok				ok					ok		ok
	ocluso									ok	ok							ok									
Amarelo	visível	ok	ok	ok		ok	ok	ok		ok			ok	ok					ok		ok			ok		ok	
	pouco				ok											ok				ok		ok	ok		ok		ok
	ocluso								ok		ok	ok			ok		ok	ok									
Azul	visível	ok	ok	ok	ok	ok	ok	ok		ok	ok		ok			ok				ok	ok					ok	
	pouco																		ok			ok	ok				ok
	ocluso								ok			ok		ok	ok		ok	ok						ok	ok		
palma	frente		ok			ok	ok	ok														ok					
	verso										ok	ok					ok	ok						ok			
	misto	ok								ok			ok	ok	ok				ok	ok		ok	ok		ok	ok	ok
	lado			ok	ok				ok							ok											

Tabela 3.2 – Dados de cada letra com relação a visibilidade dos marcadores e posição da mão.

Os dados da Tabela 3.2 possibilitaram realizar uma breve previsão de quais marcadores aparecem com mais frequência, como a posição da mão pode variar e se haveria muitos problemas de oclusão.

3.13. Aquisição de dados

Com a escolha do alfabeto e sua análise prévia, **criou-se um banco de imagens para realização de outras análises e testes de validação da técnica proposta.** Para o

processo de captura dos gestos, foram feitas inicialmente **10 capturas de cada letra do alfabeto**. Cada imagem foi feita a aquisição da imagem RGB original e a imagem de profundidade, ambas já com o recorte para mostrar apenas a mão. O alfabeto Libras possui 26 letras, então foram necessárias 260 capturas para a geração do banco de imagens.

A partir das imagens RGB transformou-se cada imagem para o espaço de cores YCrCb, utilizando a Biblioteca OpenCV e a função *cvtColor(InputArray src, OutputArray dst, int code)*. A partir das imagens em RGB e YCrCb realizou-se a normalização através das equações 1 a 8.

As imagens de profundidade foram preparadas para serem utilizadas como máscara de corte para a remoção de fundo das imagens coloridas normalizadas. Para isso basta sobrepor a imagem de profundidade sobre as imagens coloridas, analisando onde os pixels da imagem de profundidade for preto e removendo esses pixels da imagem colorida.

Deste modo, obtiveram-se as 260 imagens para análise (figura 3.27), que são as imagens coloridas (figura 3.24), que são normalizadas (figura 3.26) e com o fundo removido pela imagem de profundidade (figura 3.25).



Figura 3.24- Letra G: imagem RGB



Figura 3.25- Letra G: imagem de profundidade



Figura 3.26 Letra G: imagem RGB normalizada

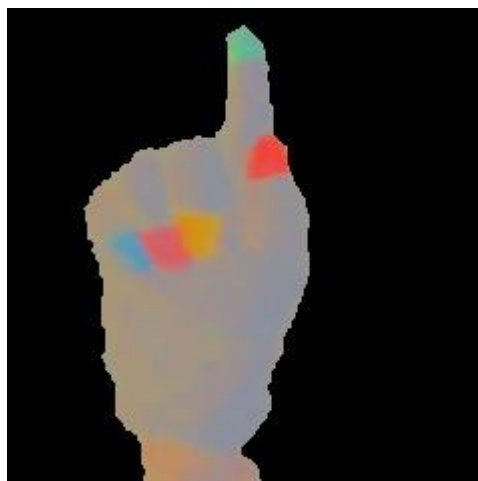


Figura 3.27- Letra G: imagem RGB normalizada com fundo removido

Todas as análises seguintes basearam-se nesse banco de 260 imagens. Isso foi feito para que fosse possível garantir que as diferenças nos resultados obtidos nas etapas seguintes desse trabalho não fossem devido à alteração na execução do gesto ou na variação do ambiente de teste.

3.14. Cálculo das distâncias relativas

Uma vez adquiridas as imagens para análise partiu-se para a **definição da técnica a ser utilizada** para o reconhecimento das letras do alfabeto Libras. Então foi realizado um **estudo sobre os aspectos que diferenciam os gestos.**

O estudo de Libras mostrou a existência de cinco parâmetros que compõem cada gesto:

- As expressões faciais/corporais, que não contribuem para a identificação do alfabeto manual.
- O movimento e a orientação das mãos, que também não contribuem, pois neste trabalho focamos apenas posturas manuais estáticas.
- O ponto de articulação, isto é, onde o gesto é feito. Idealmente as letras deveriam ser todas gesticuladas a frente do corpo, não contribuindo também para a sua identificação.
- A configuração de mão, que realmente varia de um sinal para outro.

Então, do ponto de vista de reconhecimento do alfabeto manual em Libras é a configuração da mão. Dependendo da configuração de mão de um gesto, podemos

ter ou não a presença de cada um dos marcadores na imagem de análise. O número de marcadores e quais marcadores aparecem mostram-se informações importantes, mas não suficiente para diferenciar as 26 letras. A ordem com que os marcadores aparecem na imagem é útil para diferenciar os gestos e a distância entre os marcadores também é uma medida que varia muito entre os gestos, sendo, portanto uma escolha adequada.

Assim definiu-se avaliar em um gesto quais marcadores estão visíveis e a distância entre eles. Para descobrir se um determinado marcador aparece ou não na imagem avalia-se a quantidade de pixels encontrada com a cor do marcador, caso não se obtenha um número mínimo o marcador é considerado como ocluso.

Foram colocados cinco marcadores na mão, que podem aparecer ou não o que gera um total de 32 possibilidades de combinações. Usou-se essas 32 combinações para realizar uma pré-classificação dos sinais.

Foi dado um peso para cada marcador, escolheu-se dar os pesos binários, pois assim garantimos que a soma dos pesos dos marcadores de cada gesto iria gerar um número único, e que nele estaria contida a informação dos marcadores visíveis. A distribuição do peso é mostrada na tabela 3.3.

Vermelho	16
Verde	8
Amarelo	4
Rosa	2
Azul	1

Tabela 3.3 – Pesos dos marcadores para classificação.

Por exemplo, para a letra “B” é possível ver os 5 marcadores então temos, 16 do vermelho. 8 do verde, 4 referente ao amarelo, 2 do rosa e 1 do azul, somando temos 31, então consideramos o classificador do B como sendo do grupo 31. Para a letra “N” temos apenas 2 marcadores, então somamos 8 do verde e 4 do amarelo, o classificador do N pertence ao grupo 12. Porém, algumas letras devido aos marcadores nem sempre estarem visíveis, podem aparecer em mais de um grupo de classificadores, como exemplo temos a letra “D” que pode ter os 5 marcadores visíveis e classificador pertencente ao grupo 31, ou o marcador rosa estar ocluso, o que resulta no grupo de classificação 29. Assim, a letra pertencente a dois grupos de classificadores torna-se necessário criar dois padrões diferentes para a letra “D”.

Com a pré-classificação, o tamanho do espaço de busca para o reconhecimento é reduzido, tendo em vista que sinais com mesmo número de marcadores visíveis estão agrupados, ou seja, quando é mostrando o sinal da letra “N” esse sinal será comparado apenas com os sinais que pertencerem ao grupo 12, então apenas os sinais que possuam os marcadores verde e amarelo visíveis serão considerados para um possível reconhecimento.

Essa pré-classificação é útil, mas não é suficiente para fazer o reconhecimento dos sinais. Propôs-se a utilização das distâncias entre os marcadores para fazer a distinção dos sinais dentro de um mesmo grupo. Na imagem calcula-se a posição X e Y de cada centroide dos marcadores visíveis. Porém essa posição pode variar conforme o gesto e a visibilidade do marcador. A utilização da posição absoluta do marcador na imagem não é recomendável, uma vez que gera a necessidade de que os sinais sejam feitos sempre exatamente na mesma posição. Outro problema em utilizar a posição absoluta é que qualquer rotação ou inclinação da mão altera muito a posição dos marcadores, necessitando um modelo mais flexível para cada gesto de forma a comportar essas variações. Esta flexibilização, na prática, mostrou-se inviável devido à grande semelhança de alguns sinais, tais como “M” e “N”, aumentando consideravelmente a taxa de erro no reconhecimento.

A solução aqui proposta consiste em utilizar as distâncias relativas entre os marcadores (Figura 3.28). A distância relativa é calculada através do traçado de uma linha reta entre o centroide de um marcador para os centroides de cada um dos outros marcadores. O conjunto dos tamanhos dessas retas define um vetor de até dez dimensões. Nos casos onde um dos marcadores não está visível assumimos o valor zero para essa distância, facilitando a etapa de comparação posterior. Com esta proposta, minimiza-se o problema da localização absoluta da mão na imagem. Outro problema que também é resolvido é o caso de rotação da mão no plano XY, dado que calculamos a distância entre dois marcadores, mesmo com o giro da mão, a distância entre eles não deve se alterar. Porém surgem problemas adicionais, caso o gesto seja feito mais próximo ou mais distante dos sensores o valor das componentes desse vetor aumentam ou diminuem, o que também exigiria que o gesto fosse feito na mesma posição com relação ao eixo Z dos sensores.

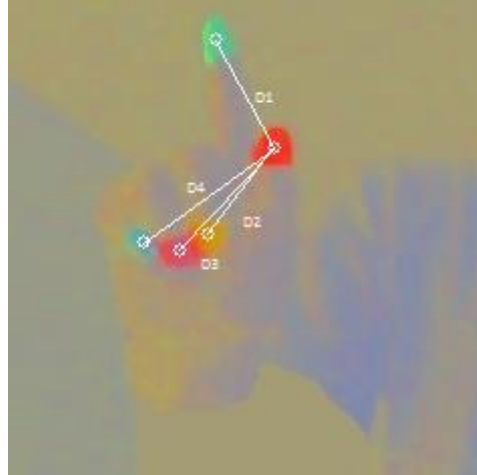


Figura 3.28- Letra G, distâncias entre o marcador vermelho e os outros quatro marcadores.

A solução encontrada consiste em **normalizar este vetor de distâncias**. Com isso conseguimos minimizar a influência da distância do usuário para os sensores (Kinect e câmera), sabendo que todas as componentes irão aumentar ou diminuir na mesma proporção, preservando a contribuição individual de cada uma das distâncias relativas entre os marcadores.

Dado o conjunto de 5 vetores bidimensionais \vec{P}_n , correspondentes as localizações dos centroides das regiões correspondentes a cada um dos cinco marcadores. Calcula-se o conjunto de 10 distâncias Euclidianas D_k segundo

$$D_1 = \|\vec{P}_1 - \vec{P}_2\| \quad (9)$$

$$D_2 = \|\vec{P}_1 - \vec{P}_3\| \quad (10)$$

$$D_3 = \|\vec{P}_1 - \vec{P}_4\| \quad (11)$$

$$D_4 = \|\vec{P}_1 - \vec{P}_5\| \quad (12)$$

$$D_5 = \|\vec{P}_2 - \vec{P}_3\| \quad (13)$$

$$D_6 = \|\vec{P}_2 - \vec{P}_4\| \quad (14)$$

$$D_7 = \|\vec{P}_2 - \vec{P}_5\| \quad (15)$$

$$D_8 = \|\vec{P}_3 - \vec{P}_4\| \quad (16)$$

$$D_9 = \|\vec{P}_3 - \vec{P}_5\| \quad (17)$$

$$D_{10} = \|\vec{P}_4 - \vec{P}_5\| \quad (18)$$

onde o índice n dos vetores \vec{P}_n , correspondem ao polegar 1, indicador 2, médio 3, anelar 4 e dedo mínimo 5.

Deste modo, define-se o vetor de distâncias \vec{D} , no espaço 10-D, para uma determinada configuração de mão por

$$\vec{D} = (D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9, D_{10}) \quad (19)$$

A fim de obter a invariância quanto à posição relativa do usuário aos sensores, normaliza-se o vetor distâncias através de

$$\vec{Z} = \text{round} \left(1000 \left(\frac{\vec{D}}{\|\vec{D}\|} \right) \right), \quad (20)$$

onde \vec{Z} é o vetor 10-D de distâncias normalizado e escalonado por um fator de 1000 a fim de obter um **vetor de números naturais** através do arredondamento efetuado pela função round.

Para testar se a solução proposta é válida, realizou-se um teste com uma amostra de cada imagem correspondente às letras que possuem os cinco marcadores sempre visíveis (grupo 31), isto é, as letras A, B, E, F, G, I, K, L, R, T, U, V e Y. O gráfico mostrado na Figura 3.29 apresenta, para cada letra, as magnitudes das 10 componentes do vetor de distâncias normalizado.

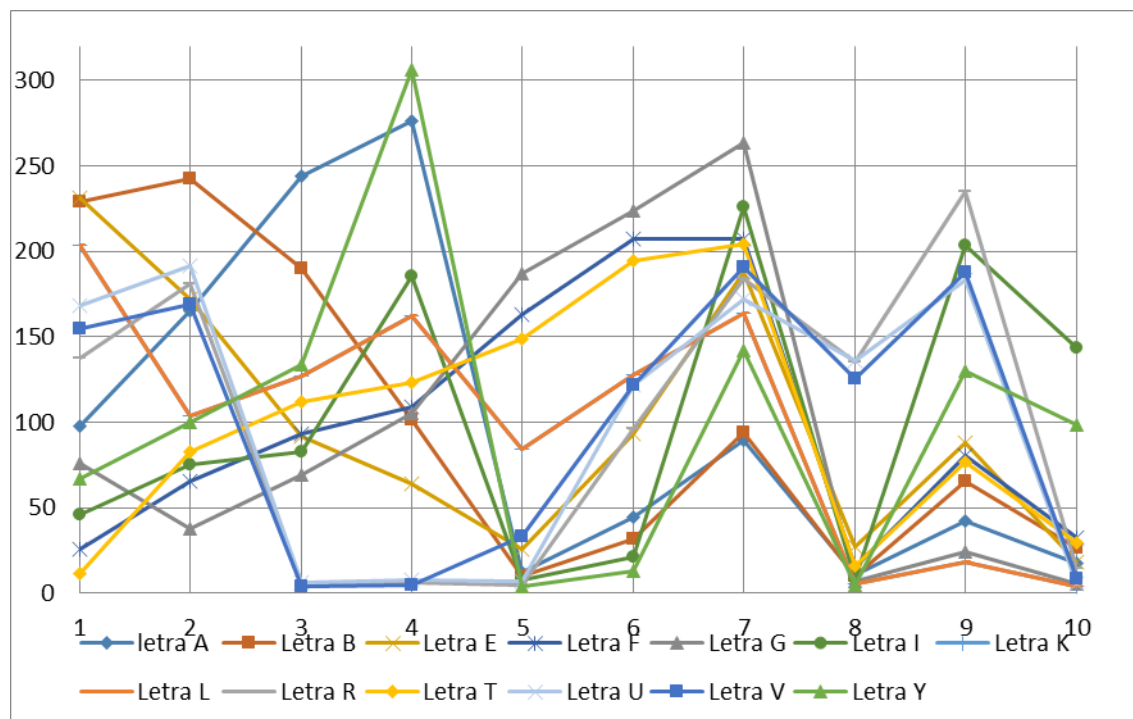


Figura 3.29 - Gráfico com as distâncias normalizadas para as 12 letras que tem classificador no grupo 31.

Analisando o gráfico da Figura 3.29 observa-se que não existem duas letras com valores iguais para todas as dez componentes do vetor de distâncias. É interessante notar que as curvas correspondentes às letras U e V possuem grande similaridade, indicando que as posturas manuais destas duas letras são semelhantes, conforme mostrado na Figura 2.1.

3.15. Cálculo do ângulo entre marcadores

O vetor de distâncias normalizadas resolveu os problemas iniciais percebidos, porém pela sua definição perde-se a informação de direção das retas que ligam um marcador ao outro. Dois gestos que alternem apenas a ordem dos dedos possuirão o mesmo vetor de distâncias, bem como gestos que se diferenciam apenas por rotação, conforme mostrado na Figura 3.30.

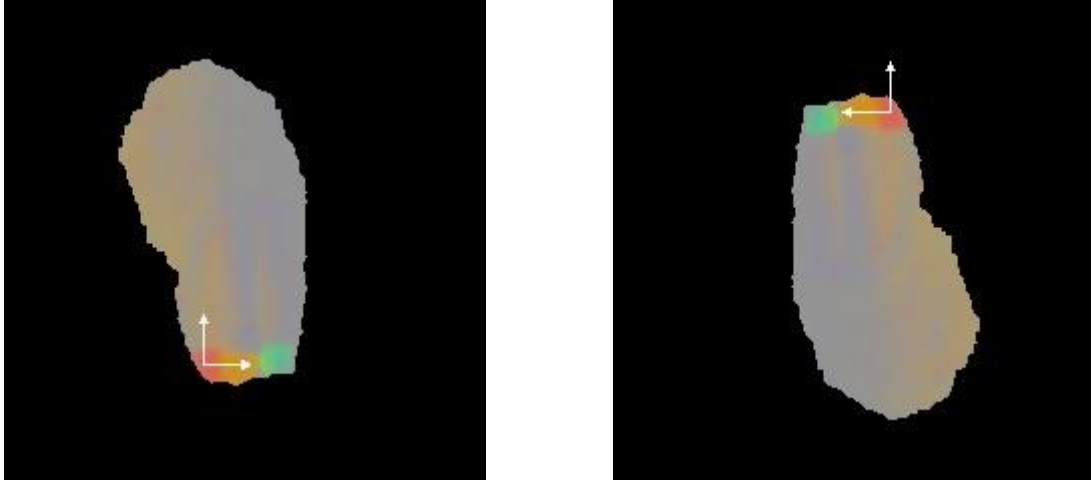


Figura 3.30 - Dois gestos com mesma configuração de mão com rotação 180° entre as imagens.

Para resolver esse problema decidiu-se **adicionar duas novas componentes ao vetor de distância**, que correspondem às coordenadas do vetor normalizado obtido pela diferença entre os dois primeiros marcadores localizados, na sequência polegar, indicador, médio, anelar e mínimo, de acordo com

$$\overrightarrow{Pd} = \min_i \{\vec{P}_i\} - \min_{i \neq j} \{\vec{P}_j\}, \quad (21)$$

onde \overrightarrow{Pd} corresponde ao vetor diferença e

$$\vec{d} = \text{round} \left(100 \cdot \frac{\overrightarrow{Pd}}{\|\overrightarrow{Pd}\|} \right), \quad (22)$$

o vetor diferença normalizado e escalado por 100. Podemos então definir o vetor de características \vec{A} como

$$\vec{A} = (d_x, d_y, Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8, Z_9, Z_{10}) \quad (23)$$

onde d_x, d_y correspondem às coordenadas X e Y do vetor \vec{d} , e Z_k cada uma das componentes do vetor \vec{Z} .

O vetor \vec{A} contém, além das informações de distâncias normalizadas entre os marcadores visíveis, a informação do vetor unitário escalonado da primeira distância que pode ser diretamente relacionada ao **ângulo de rotação da mão.**

Desta forma pretendeu-se reduzir o número de sinais que possam estar muito próximos no modelo.

3.16. Criação dos padrões

Após a aquisição dos dados e os testes de validade da solução de reconhecimento, foram criados os padrões para cada gesto. Para isso foi utilizado os dados de 10 imagens de profundidade e coloridas para cada uma das 26 letras do alfabeto.

Para se obter uma maior flexibilidade do modelo de cada letra, cada uma das 10 imagens contém o gesto feito em uma posição diferente (Figuras 3.31 e 3.32), ou seja, variamos a distância para os sensores, a altura da mão, a proximidade com o corpo, a rotação da mão, essa abordagem também foi utilizada por *Yuan Yao* [15] e *Pedro* [16]. Com isso, se aumentou a complexidade do problema.

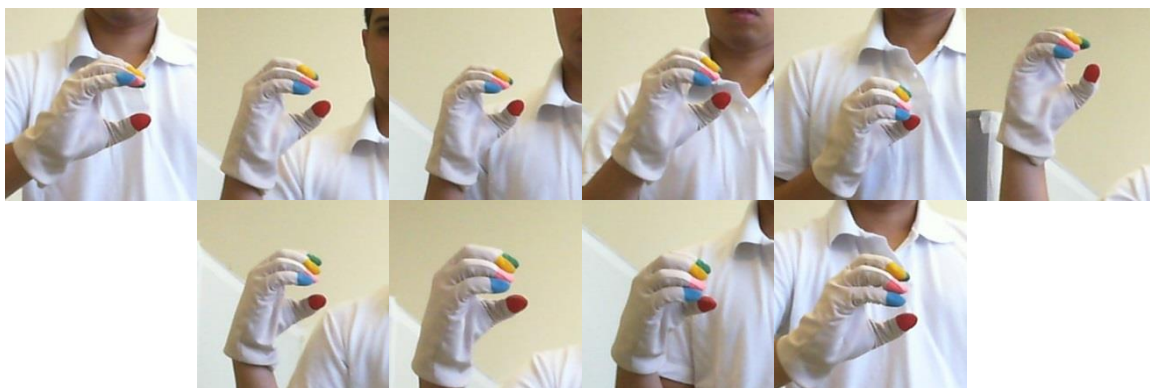


Figura 3.31 Letra C: Exemplo das 10 imagens coloridas em diferentes posições.

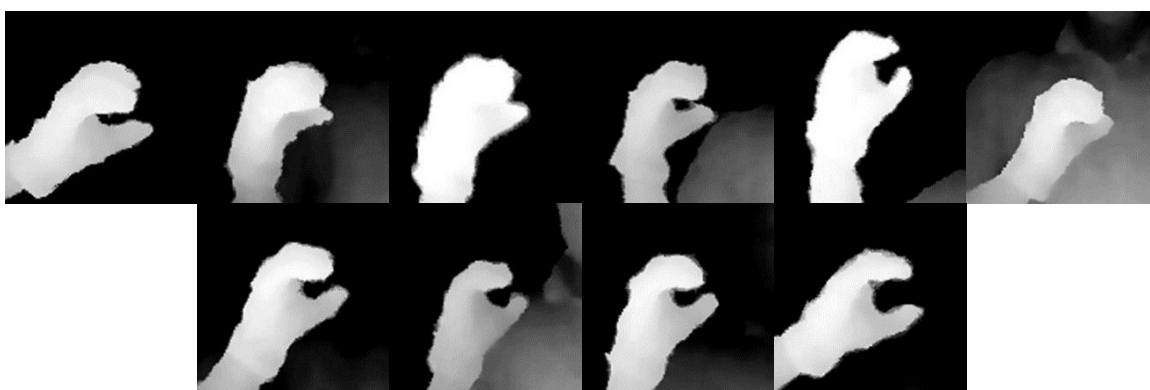


Figura 3.32- Letra C: Exemplo das imagens de profundidade em diferentes posições.

Para obter um modelo mais similar ao obtido pela visão humana, adotou-se, inicialmente, a estratégia de marcar manualmente os centroides dos marcadores visíveis.



Para cada letra se carregava os dados das imagens, os centroides de cada marcador eram clicados e a sua posição salva. Aproveitou-se ainda a informação de cor do pixel clicado a fim de melhorar o algoritmo de busca dos marcadores. Este procedimento foi feito para as 10 imagens de cada uma das 26 letras. Caso o marcador não estivesse visível, a posição desse marcador foi zerada de forma a auxiliar no cálculo das distâncias relativas.

Após o procedimento de marcação manual dos centroides dos marcadores, foi realizado o cálculo das distâncias relativas entre os marcadores conforme apresentado na seção 3.14. Deste modo, criou-se um vetor de distâncias para cada uma das 10 imagens de cada letra.

De posse dos dez vetores é feita a análise de classificação de cada um dos sinais nas imagens, o número de marcadores encontrados e quais são visíveis é anotado conforme explicado anteriormente. Caso o mesmo gesto possua três ou mais imagens com o mesmo classificador então é considerado como um gesto válido com esse classificador. Então é possível que um mesmo gesto possa ter mais de uma classificação.

Para cada gesto considerado válido, ou seja, que possui ao menos três imagens com o mesmo classificador, foi calculado a média e o desvio padrão de cada um dos 12 elementos que compõem os vetores de distância.



Para armazenar as informações de cada gesto foi criada uma *struct* com os dados de ID, número do classificador dos marcadores, ângulo entre os dois primeiros marcadores, desvio do ângulo, vetor de médias, vetor de desvio padrão, e o símbolo correspondente.

```

struct letra{
    string id;
    int marcadores;
    int tamBanco;
    int angulo[2];
    int desvAng[2];
    int media[10];
    int desvPad[10];
    char simbolo;
};

```

O símbolo é a letra a ser representada pelo sinal. O número do classificador de marcadores é o calculado baseando na detecção dos marcadores como demonstrado anteriormente. O ID é criado com símbolo da letra e adicionando o classificador, exemplo símbolo “A” e classificador 31 o ID será A31. **Tamanho do banco é o numero de imagens que possuem o mesmo classificador, essa informação será útil no caso de duplo reconhecimento.** O ângulo é referente às medias das componentes d_x e d_y calculadas a partir dos dois primeiros marcadores encontrados. O desvio do ângulo é o desvio padrão de d_x e d_y . Vetor de média é o conjunto das médias dos 10 valores de distâncias ponderadas. O vetor de desvio padrão é o conjunto dos desvios dos 10 valores ponderados.



Foi feita a análise dos marcadores das 260 imagens, com isso se gerou uma tabela para cada letra e o número dos classificadores. Assim, foi possível criar 35 diferentes padrões de gestos válidos para as 26 letras, ou seja, têm-se letras que podem ser feitas de formas diferentes, isso devido à detecção ou não de alguns dos marcadores.

Classificador	31	30	29	28	27	25	24	23	21	19	17	14	12	Total
A	10	-	-	-	-	-	-	-	-	-	-	-	-	10
B	10	-	-	-	-	-	-	-	-	-	-	-	-	10
C	5	-	-	-	-	-	-	5	-	-	-	-	-	10
D	5	-	5	-	-	-	-	-	-	-	-	-	-	10
E	10	-	-	-	-	-	-	-	-	-	-	-	-	10
F	10	-	-	-	-	-	-	-	-	-	-	-	-	10
G	10	-	-	-	-	-	-	-	-	-	-	-	-	10
H	-	-	-	4	-	-	-	-	-	-	-	-	6	10
I	10	-	-	-	-	-	-	-	-	-	-	-	-	10
J	-	-	-	-	-	-	-	-	1	-	9	-	-	10
K	10	-	-	-	-	-	-	-	-	-	-	-	-	10
L	10	-	-	-	-	-	-	-	-	-	-	-	-	10
M	-	4	-	-	-	-	-	-	-	-	-	6	-	10
N	-	-	-	7	-	-	-	-	-	-	-	-	3	10
O	1	-	-	-	1	1	-	1	-	3	3	-	-	10
P	1	-	-	9	-	-	-	-	-	-	-	-	-	10
Q	-	-	-	-	-	-	10	-	-	-	-	-	-	10
R	10	-	-	-	-	-	-	-	-	-	-	-	-	10
S	-	-	-	-	5	1	-	-	-	4	-	-	-	10
T	10	-	-	-	-	-	-	-	-	-	-	-	-	10
U	7	-	3	-	-	-	-	-	-	-	-	-	-	10
V	10	-	-	-	-	-	-	-	-	-	-	-	-	10
W	10	-	-	-	-	-	-	-	-	-	-	-	-	10
X	6	3	-	-	-	-	1	-	-	-	-	-	-	10
Y	10	-	-	-	-	-	-	-	-	-	-	-	-	10
Z	10	-	-	-	-	-	-	-	-	-	-	-	-	10
Total	165	7	8	20	6	2	11	6	1	7	12	6	9	260

Tabela 3.4 – Distribuição das imagem de acordo com a letra e o classificador calculado.

Na Tabela 3.4 é apresentado o número de imagens da mesma letra com o mesmo classificador. As letras que possuem algum classificador com menos de três imagens não teve um padrão criado. Desta tabela também é possível ver os 35 padrões criados: A31, B31, C31, C23, D31, D29, E31, F31, G31, H28, H12, I31, J17, K31, L31, M30, M14, N28, N12, O19, O17, P28, Q24, R31, S27, S19, T31, U31, U29, V31, W31, X31, X30, Y31, Z31.

3.17. Estratégia de busca

Com os modelos das 26 letras já criados pode-se partir para o reconhecimento dos sinais feitos. Para isso cada gesto capturado passa por um processo de preparação para poder ser comparado com os modelos dos sinais.

Cada imagem capturada passa por todas as etapas descritas anteriormente. Primeiro se faz a homografia entre a imagem colorida e a de profundidade, após é feita a localização da mão a partir da imagem de profundidade e o recorte da área a ser analisada. Então se cria as imagens coloridas nos espaços RGB e YCrCb normalizadas.

Nessa etapa do processo temos apenas as imagens para análise onde é feita a localização dos marcadores. Com os marcadores localizados é feito o cálculo do classificador. O número do classificador é importante, pois baseado nele será feita a busca no banco de gestos.

Todos os modelos são carregados em um vetor de *struct*, ou seja, têm-se os 35 possíveis sinais, ordenados pelo número do classificador. Para um novo sinal feito ser reconhecido, ele é comparado apenas com os que possuem o mesmo classificador.

Dentro do grupo de sinais possíveis por terem o mesmo classificador o próximo passo é comparar as componentes referentes ao ângulo dado por d_x e d_y , caso esses dois valores estejam fora do limite do valor médio mais três vezes o desvio padrão esse modelo é descartado, caso contrário continua-se a análise para as outras 10 distâncias. Se alguma dessas distâncias não estiver dentro dos limites, o modelo também é descartado. No caso de todas as comparações o sinal feito ficar dentro dos limites do modelo, esse modelo é marcado como provável sinal feito. Para afirmar como reconhecido é preciso fazer a análise de todos os modelos com mesmo classificador, se por ventura dois ou mais modelos se apresentarem como corretos será preciso utilizar uma técnica para desempatar esses modelos.

A diferenciação de sinais com múltiplo reconhecimento foi feito analisando o menor erro entre o sinal analisado e os potenciais modelos reconhecidos, assim o modelo com menor erro é indicado como a letra reconhecida.

4. RESULTADOS OBTIDOS

Neste capítulo serão descritos os procedimentos de teste e validação para o sistema, bem como os resultados obtidos.

Foram feitos 4 experimentos para testar a solução de reconhecimento. Foram utilizados dois usuários diferentes nos experimentos. Chamados de usuário A e usuário B para facilitar a identificação.

4.1. Primeiro Experimento

Para garantir que os modelos criados para as 26 letras são válidos, decidiu-se inicialmente fazer o teste de validação, aplicando-se ao sistema as mesmas entradas que geraram o modelo. Essa técnica foi adotada para verificar se os modelos conseguem cobrir todas as entradas e se existe ainda o problema de duplo reconhecimento para um mesmo gesto.

As cinco posições clicadas de cada marcador em todas as imagens foram colocadas como entrada da função de reconhecimento. Então esses dados passaram por todas as etapas de preparação até obter o vetor com os dados de ângulo, distâncias relativas e classificadores. Tendo os dados de classificador como referência, foi realizada a comparação de cada modelo com o classificador, gerando uma resposta sobre quais modelos satisfizeram as condições de ângulo e de distâncias relativas.

Têm-se 260 imagens de entrada, 26 para cada letra feita pelo usuário A. Com esses dados gerou-se uma tabela de reconhecimento. Cada letra é representada por uma linha e cada um dos modelos é representado em uma coluna. Essa tabela é útil para se avaliar quantos sinais da mesma letra foram reconhecidos corretamente, quantos tiveram mais de uma correspondência e quantos não foram reconhecidos.

Inicialmente, foi feito o teste sem considerar a informação de ângulo, o que gerou vários casos de duplo reconhecimento e até mesmo de falso reconhecimento conforme mostrado na Tabela 4.1 a seguir.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	1	1	1	1	0	1	0	1	0	1	0	1	0	C	1	1	1	1	E	1	1	1	0	1	1
1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	1	1	E	1	1	C	0	0	1
1	0	1	1	C	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1
1	0	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	0	1	0	0	1	0	1	1
1	1	E	1	1	1	1	0	1	0	1	1	1	0	0	1	1	1	1	1	0	1	1	0	1	X
1	0	1	1	1	0	1	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	Z	1	X
1	0	1	1	1	1	1	0	1	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	0	1	0	1	0	1	1	1	M	0	1	1	1	1	1	1	1	1	0	1	X
1	0	1	1	1	0	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	0	0	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1
10	2	9	10	9	3	10	4	10	1	10	9	9	0	3	9	10	10	9	8	8	6	8	3	9	7
0	8	0	0	0	7	0	0	0	9	0	1	1	9	6	1	0	0	1	0	2	4	1	6	1	0
0	0	1	0	1	0	0	6	0	0	0	0	0	1	1	0	0	0	0	2	0	0	1	1	0	3
100%	20%	90%	100%	90%	30%	100%	40%	100%	10%	100%	90%	90%	0%	30%	90%	100%	100%	90%	80%	80%	60%	80%	30%	90%	70%
Acertos	186		72%																						
Não Reconhecidos	57		22%																						
Erros	17		7%																						

Tabela 4.1 – Distribuição das letras com relação ao modelo considerado correto, sem a utilização do ângulo como parâmetro.

Na Tabela 4.1 os elementos em verde são os corretamente reconhecidos, os amarelos os que tiveram duplo reconhecimento ou não foram reconhecidos, e os elementos na cor rosa são os reconhecidos erradamente. Esta tabela apresenta também o percentual de acerto de cada letra, o resultado do teste com quantidade e percentual de acertos e erros.

Sem a análise de ângulo obteve-se 186 posturas corretamente reconhecidas (72%), 57 com duplo reconhecimento dos gestos (22%) e 17 reconhecidos erroneamente (7%).

A Tabela 4.2 apresenta os resultados obtidos considerando no vetor, as informações de ângulo.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	0	0	1
1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0
1	0	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0
1	0	1	1	1	1	1	1	1	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0
1	1	1	1	1	0	1	1	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1
10	8	10	9	9	5	10	10	9	9	9	9	9	10	2	9	10	10	9	8	8	6	9	6	9	7
0	2	0	1	1	5	0	0	1	1	1	1	1	0	8	1	0	0	1	2	2	4	1	4	1	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100%	80%	100%	90%	90%	50%	100%	100%	90%	90%	90%	90%	90%	100%	20%	90%	100%	100%	90%	80%	80%	60%	90%	60%	90%	70%
Acertos	219		84%																						
Não Reconhecidos	41		16%																						
Erros	0		0%																						

Tabela 4.2 – Distribuição das letras com relação ao modelo considerado correto, com a utilização do ângulo como parâmetro.

Com a análise de ângulo obteve-se 219 posturas corretamente reconhecidas (84%), 40 com duplo reconhecimento dos gestos (16%) e nenhum reconhecido erroneamente (0%).

Após o teste é possível afirmar que os modelos para cada letra foram bem gerados, com uma taxa de reconhecimento de aproximadamente 84% e com erro nulo. Este resultado pode ser considerado aceitável, exceto em alguns casos onde houve dúvidas entre os modelos das Letras “F” e “T”, que já era esperado pela semelhança na configuração das mãos, o mesmo ocorre para as duplas “U” “V” e “X” “Z”.

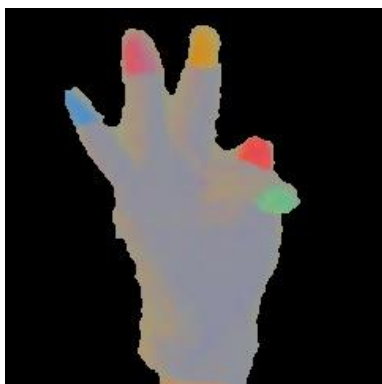


Figura 4.1- Exemplo do sinal “F”

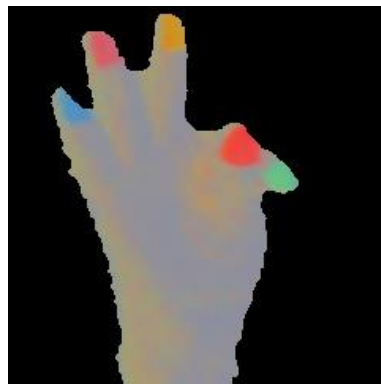


Figura 4.2- Exemplo do sinal “T”

Como se pode observar na Figura 4.1, os sinais das letras “F” e “T” são visualmente muito semelhantes, tornando a distinção entre elas uma tarefa árdua.

4.2. Segundo experimento

Após o modelo ser validado, foi preciso testar a solução completa, ou seja, a entrada do sistema apenas com a imagem colorida normalizada e de profundidade, com a homografia já feita. Toda parte de localização dos marcadores e cálculo dos centroides foi automática. Após o cálculo dos centroides foi feito o cálculo dos classificadores, ângulo e valores das distâncias ponderadas.

Com os dados dos classificadores como referência foi feita a comparação com cada um dos modelos e foi gerada uma resposta de quais modelos satisfizeram as condições de ângulo e de distâncias relativas.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	0	0	0	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	0	1	0	1	1	1	0
0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	0	0	0
1	0	1	0	1	0	1	1	1	1	1	1	n	0	1	1	1	0	1	1	0	0	1	0	1	0
0	1	0	0	1	f	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1	1
1	1	0	0	1	f	0	1	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0
0	0	0	1	1	0	0	1	0	1	1	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0
1	0	0	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1
0	1	1	0	1	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	1	1	0	0	0
1	1	0	0	0	0	0	1	0	1	1	1	0	1	1	0	1	0	1	0	0	0	0	1	0	1
1	1	0	0	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1
6	5	2	2	8	3	6	10	8	7	8	6	6	7	6	5	8	5	9	7	2	2	5	6	5	4
4	5	8	8	2	5	4	0	2	3	2	4	3	3	4	5	2	5	1	3	8	8	5	4	5	6
0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
60%	50%	20%	20%	80%	30%	60%	100%	80%	70%	80%	60%	60%	70%	60%	50%	80%	50%	90%	70%	20%	20%	50%	60%	50%	40%
Acertos				148	57%																				
Não Reconhecidos				109	42%																				
Erros				3	1%																				

Tabela 4.3 – Distribuição das 260 letras com relação ao modelo do primeiro experimento, utilizando a busca automática dos marcadores.

Com a análise das imagens reais obteve-se 148 corretos reconhecimento (57%), 109 com duplo reconhecimento ou não reconhecimento (42%) e 3 reconhecido erroneamente (2%).

O número de 109 sinais não reconhecidos é elevado, **investigando as causas temos que 40 deles foram devido a problemas na captura da imagem de profundidade,** o que causou o erro no corte e por fim na não localização dos marcadores, conforme apresentado na Figura 4.2

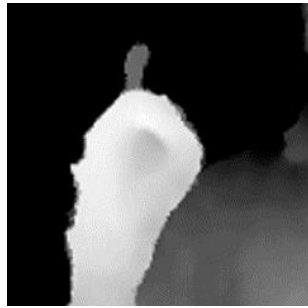


Figura 4.4 - Letra D: imagem de profundidade

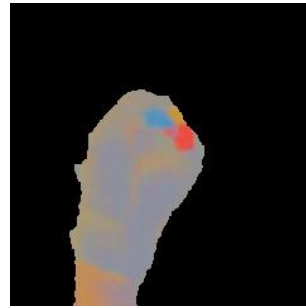


Figura 4.3- Letra D: imagem para análise

Na figura 4.3 é possível ver um exemplo de caso onde perdemos a localização de um marcador devido a perda da informação de profundidade em alguns dos frames.

Outra causa de erros são as oclusões parciais, onde o método de busca dos marcadores não foi capaz de localizar corretamente todos os marcadores visíveis.



Figura 4.6 - Exemplo sinal "C"



Figura 4.5 - Exemplo sinal "O"

Conforme apresentado nas Figuras 4.6 e 4.7, as letras que mostraram mais dificuldades de ser localizadas automaticamente foram "O" e "C" devido a suas configurações de mão gerarem oclusões de marcadores. A dificuldade de diferenciar as letras "F" e "T" também continuou.

Os resultados obtidos comprovam que a abordagem adotada é viável. Os casos de duplo reconhecimento já eram esperados devido à semelhança dos sinais, e os casos de sinais não reconhecidos também eram esperados devido ter se forçado bastante as variações na configuração de mão para o mesmo sinal.

4.3. Terceiro experimento

Para revalidar a ideia que é possível gerar um modelo para cada letra do alfabeto Libras, e corrigir as prováveis falhas do primeiro ensaio, foi solicitada a ajuda de um professor especialista em Libras para fazer os sinais corretamente. Além disso, foram feitas 50 imagens de cada uma das 26 letras, e mais um sinal para representar a separação entre as palavras, totalizando 27 sinais. Isso foi feito acreditando que o modelo do primeiro ensaio falhou devido ao pequeno número de amostras, que resultou em um modelo muito restrito.

Após a coleta das 50 imagens de cada letra foi feita uma pequena triagem nas imagens, uma vez que foi percebido que podem acontecer falhas no momento da captura da imagem. As imagens que tiveram falhas no recorte da mão foram retiradas. Das 1350 imagens capturadas 18% tinham erros na captura e foram descartadas. Restando 1102 imagens. Esta abordagem foi utilizada considerando-se que um trabalho futuro irá aprimorar a técnica de recorte da mão usando imagens de profundidade.

As imagens restantes foram separadas em dois grupos. O primeiro grupo ficou com 30 imagens de cada sinal e é utilizado para gerar o padrão para os 27 sinais. O segundo grupo com 292 imagem é o grupo de teste, essas imagens não fazem parte do modelo e serão usadas para verificar o grau de generalização e robustez do método proposto.

O procedimento adotado é similar ao utilizado no experimento um, utilizando as 30 imagens de cada sinal para gerar os modelos. Neste caso obtivemos 56 formas possíveis para fazer os 27 sinais, um aumento significativo de modelos quando comparado aos 35 gerados no primeiro ensaio. Isso é devido ao uso de um maior número de amostras, o que abrange uma maior variação na quantidade de marcadores visíveis para a mesma letra. Os resultados obtidos neste ensaio são apresentados na Tabela 4.4.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	0	1	1	1	C	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	V	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	C	1	1	1	1	1	1	1	1	1			
1	1	1	1	L	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	V	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	S	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	Y	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	0	S	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	S	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	U	1	L	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1			
0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	X	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1			
28	29	30	27	30	27	29	25	28	30	30	30	30	19	29	25	30	29	30	30	30	30	30	30	28	30	30			
2	0	0	1	0	2	1	1	2	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	1	0	0			
0	1	0	2	0	1	0	4	0	0	0	0	0	10	1	3	0	1	0	0	0	0	0	0	1	0	0			
93%	97%	100%	90%	100%	90%	97%	83%	93%	100%	100%	100%	100%	63%	97%	83%	100%	97%	100%	100%	100%	100%	100%	100%	93%	100%	100%			
Acertos				773	95%																								
Não Reconhecidos				13	2%																								
Erros				24	3%																								

Tabela 4.4– Validação da classificação das 810 letras que geraram o modelo do terceiro experimento.

Com a análise das 810 que geraram os modelos obteve-se 773 imagens com o reconhecimento correto (95%), 13 não reconhecidos (2%) e 24 reconhecido erroneamente (3%). Esse resultado mostra que o modelo realmente representa bem os elementos que compõem cada um dos sinais.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_
1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1			1	1	1	1	1	1	1		1	1	1	1	1	V	1	1	1	1	1	1	1	1	1	1
1			1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1			1	1	1	1	1	S	1		1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1
1			1	1	1	1	1	1			1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1
1			1	1	1	1	1	1			1		1	M	1	1	1	1	1	1	1	1	1	1	1	1
1				1	1	1	1	1			1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1
1				1	1			1			1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1
1				1	1			1			1	1	1	1	1	1	1	1	1		1	1	1	1	1	1
1				1							1	1	1	1	0	1	1	1	1		1	1	1	1	1	
1				1							1	1	1	M	1		1	1	1		1	1	1	1	1	
1				1							1	1	1	1	1		1	1	1		1	1		1	1	
1				1							1		1		1		1	1	1		1	U		1	1	
				1								1		1		1	1	1		1	1			1		
				1										1		1	1	1		1	1			1		
				1										1		1	1	1		1				1		
				1										0			1	1						1		
				1										1			1	1								
				1													1	1								
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							
				1															1							

Tabela 4.5– Validação da classificação dos 292 Sinais separados para teste.

Com a análise das 292 imagens separadas para o teste, obteve-se 281 imagens corretamente reconhecidas (96%), 2 não reconhecidas (1%) e 9 reconhecidas erroneamente (3%).

Esse resultado mostra que o modelo realmente representa os 1102 sinais do ensaio, e confirma o resultado que teoricamente deveria ter sido obtido também no primeiro ensaio.

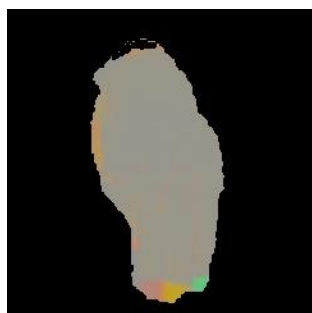


Figura 4.8 - Exemplo sinal “M”

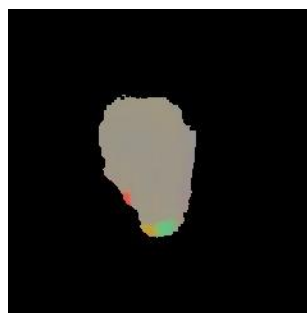


Figura 4.7 - Exemplo sinal “N”

As Figuras 4.8 e 4.9 mostram que as letras “M” e “N” acabaram tendo o modelo muito semelhante, a provável causa é o marcador rosa da letra “M” não ter

aparecido em algumas das imagens que geraram o modelo, gerando um “M” com apenas dois marcadores, muito semelhante ao “N”.

4.4. Quarto experimento

Com os bons resultados obtidos no terceiro experimento, restou a dúvida se a razão do primeiro experimento ter falhado foi devido aos gestos terem sido feitos de forma errônea ou devido ao pequeno número de amostras utilizadas para a formação do modelo de cada letra. Assim, repetiu-se o primeiro experimento, com o mesmo usuário, porém utilizando 30 imagens para a modelagem de cada letra.

Neste terceiro ensaio foram adquiridas 810 imagens e aplicado um filtro para retirar as imagens com que apresentaram defeitos na captura e processamento da imagem de profundidade, restando 776 imagens. Este conjunto de imagens foi dividido também em dois grupos, um formado por 20 imagens de cada sinal, totalizando 540 imagens. O segundo grupo, composto pelas 236 imagens restantes, foi utilizado como teste.

Ao primeiro grupo de imagens foi aplicado o mesmo processo de geração dos modelos dos outros experimentos, resultando em 57 padrões para os 27 sinais. A seguir se confrontou as imagens do primeiro, e também do segundo grupo com esses padrões.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_
1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Z	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	0	1	1	U	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	F	1	1	1	1	Z	1	X
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	Z	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	0	1	1	1	Z	1	0
1	1	1	1	1	0	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	0	1	X	1
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1	1	1	M	1	1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	M	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	1	1	1	1	1	M	K	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	D	1	M	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	L	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	L	1	1	1	1	1	M	1	1	1	1	1	F	1	1	0	1	1	1	1
1	1	1	1	1	0	1	U	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	X	1
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	Z	1	X
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	U	1	1	1	1	P	1	1	X
1	1	1	1	1	1	1	1	0	1	1	1	1	M	1	1	1	1	U	1	1	V	1	1	Z	1	0
19	20	20	20	20	17	16	19	19	19	20	17	20	4	18	20	20	18	17	17	19	19	18	13	20	15	16
1	0	0	0	0	3	2	0	1	1	0	1	0	1	1	0	0	0	3	1	1	0	0	1	0	0	4
0	0	0	0	0	0	2	1	0	0	2	0	15	1	0	0	2	0	2	0	1	2	6	0	5	0	0
95%	100%	100%	100%	100%	85%	80%	95%	95%	95%	100%	85%	100%	20%	90%	100%	100%	90%	85%	85%	95%	95%	90%	65%	100%	75%	80%
Acertos	480				89%																					
Não Reconhecidos	21				4%																					
Erros	39				7%																					

Tabela 4.6– Validação da classificação das 540 letras que geraram o modelo do quarto experimento.

A Tabela 4.6 apresenta os resultados obtidos com a classificação do primeiro grupo, isto é, com a análise das 540 imagens que geraram os modelos. Foram obtidas 480 imagens corretamente classificadas (89%), 21 imagens não foram reconhecidas (4%) e 39 reconhecidas erroneamente (7%). Esse resultado mostra que, com o uso do modelo proposto, pode-se representar adequadamente os elementos que compõem cada um dos sinais.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	Z	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	Z	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	Z	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	M	1	1	1	1	1	1	1	1	1	1	1	1	
	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	V	1	1	1	1	1	
	1	1	1	1			1	0	1	1	1	1	M	1	1		1	1	1	V	1	1	1	1	1	
		1	1	1			1	1	1	1	1	1	M	1	1		1	1	1		1	1	Z	1		
		1		1			1	1	1	1	1	1	M	1	1		1	1	1			1	Z	1		
		1		1			1			1		1	M		1			1	1			1	Z			
				1								1	M		1											
												1	M													
												1	M													
4	7	10	8	11	5	6	10	8	9	10	9	13	1	9	11	6	9	10	10	5	8	10	4	9	7	3
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	2	0	0	6	0	2	0
80%	100%	100%	100%	100%	100%	100%	100%	89%	100%	100%	100%	100%	8%	100%	100%	100%	100%	100%	100%	71%	100%	100%	40%	100%	78%	100%
Acertos	212		90%																							
Não Reconhecidos	1		0%																							
Erros	23		10%																							

Tabela 4.7– Validação da classificação dos 236 Sinais separados para teste.

A Tabela 4.7 apresenta os resultados obtidos com o conjunto de teste, isto é, as 236 imagens não utilizadas para a construção do modelo. Foram obtidas 212 imagens corretamente reconhecidas (90%), 1 imagem não foi reconhecida (<1%) e 23 imagens reconhecidas erroneamente (10%).

Esse resultado mostra que o modelo realmente representa as 776 imagens, e confirma os resultados obtidos no segundo e terceiro experimentos.

Este quarto experimento apresentou os mesmo problemas de reconhecimento das letras, “M” e “N”, como também “X” e “Z”.

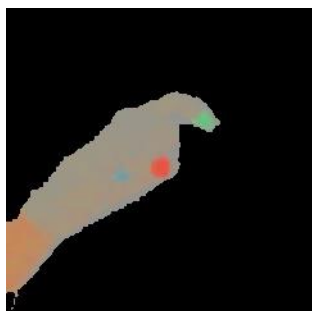


Figura 4.10- Exemplo sinal “X”

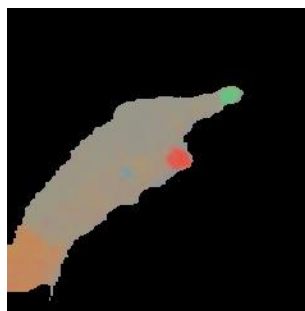


Figura 4.9- Exemplo sinal “Z”

As Figuras 4.9 e 4.10 mostram a semelhança entre as posturas manuais dos sinais “X” e “Z”. Em Libras, os sinais “X” e “Z” possuem movimentos bem diferentes, facilitando a identificação de cada um. É interessante observar que neste experimento não houve confusões entre os sinais “F” e “T”, mostrando que o modelo

proposto pode ser usado com sucesso para distinguir posturas muito similares, desde que não ocorram erros graves de pré-processamento nas etapas de extração do fundo, identificação e cálculo dos centroides.

4.5. Comparação entre experimentos

O terceiro e quarto experimentos obtiveram um ótimo resultado. Mostrando que o modelo proposto é adequado para gerar os padrões e representar as imagens usadas. No entanto, é interessante investigar se os padrões gerados em cada experimento são compatíveis entre si.

Inicialmente, aplicou-se os padrões gerados no terceiro experimento como modelo para analisar todas as imagens usadas no quarto experimento.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_
73%	100%	18%	65%	19%	3%	95%	88%	0%	0%	0%	66%	100%	8%	19%	0%	100%	0%	83%	0%	27%	41%	95%	5%	73%	52%	61%

Acertos	338	44%
Não Reconhecidos	152	20%
Erros	286	37%

Tabela 4.8– Resultado reconhecimento das 776 imagens do quarto experimento comparadas com os padrões do terceiro experimento.

A Tabela 4.8 mostra que a taxa de reconhecimento dos sinais foi bem irregular. Muitos sinais com acertos acima de 70% e outros sinais com taxa de reconhecimento abaixo de 20%.

A seguir, foi feito um experimento inverso, isto é, aplicou-se os padrões gerados no quarto experimento para analisar todas as imagens usadas no terceiro experimento.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_
15%	82%	75%	55%	0%	33%	75%	21%	3%	50%	0%	60%	82%	60%	39%	0%	50%	0%	37%	0%	7%	33%	93%	36%	26%	14%	85%
Acertos				376	34%																					
Não Reconhecidos				410	37%																					
Erros				316	29%																					

Tabela 4.9– Resultado reconhecimento das 1102 imagens do terceiro experimento comparadas com os padrões do quarto experimento.

O resultado apresentado na Tabela 4.9 indica que a taxa de reconhecimento dos sinais também foi irregular, alguns sinais com acertos acima de 70% e outros sinais com taxas de reconhecimento abaixo de 20%.

4.6. Análise dos resultados obtidos

Foram feitos quatro experimentos com bancos de imagens diferentes. O resultado obtido no primeiro experimento difere consideravelmente dos outros dois ensaios. No primeiro experimento usaram-se imagens bastante diversas quanto ao ambiente de aquisição, ângulo da câmera, posturas serem feitas por um usuário não fluente em Libras, além de ser usado um pequeno número de amostras sem qualquer pré-processamento quanto à qualidade da imagem RGB e de profundidade obtidas. Assim, a taxa de reconhecimento deste experimento ficou muito abaixo do esperado.

O terceiro e o quarto experimentos tiveram uma ótima taxa de reconhecimento, acima de 89%, muito semelhante ao que era esperado, quando comparado a outros trabalhos semelhantes publicados. As taxas de acerto dos dois modelos foram similares para os dois grupos de imagens, o grupo que gerou o modelo e o grupo de imagens separado usado para teste.

As falhas de reconhecimento dos dois experimentos também foram semelhantes. Para ambos os experimentos a causa de alguns dos sinais não serem reconhecidos foi devido ao modelo proposto não gerar padrões para sinais com menos de três ocorrências. Por exemplo, se o sinal correspondente à letra “B” é analisada como tendo 4 marcadores, e os padrões obtidos para esta letra “B” não possuir nenhum modelo com 4 marcadores, esse sinal não é reconhecido.

Outra falha que foi indicada nos dois experimentos é a questão da letra “M” perdendo o marcador do dedo indicador, o que faz com que o padrão se assemelhe muito com da letra “N”, causando assim cerca de 50% dos erros de reconhecimento. Outro grande causador de erro foi a grande similaridade entre as posturas manuais estáticas das letras “X” e “Z”.

No confronto entre os experimentos o resultado foi muito abaixo do esperado, em cada experimento obteve-se uma taxa de reconhecimento da ordem de 90%

para quase todas as letras. Esperava-se que ao se confrontar as imagens de experimentos diferentes o resultado fosse próximo a 90% também, porém foi observada uma queda significativa da taxa de reconhecimento para aproximadamente 45% para os dois experimentos. No caso do terceiro para o quarto experimentos, não foram efetuadas grandes mudanças no ambiente, nem na quantidade e qualidade dos dados que geraram os modelos.

A baixa taxa de reconhecimento no confronto pode ser creditada à mudança do usuário. O usuário B, do terceiro experimento é professor de Libras e faz uso da língua de sinais diariamente para se comunicar. Para ele fazer os sinais manuais é natural e a repetibilidade não é um problema. O usuário A, dos outros experimentos não possui muita experiência com Libras, e apenas tentou copiar os sinais manuais ensinados pelo professor. Como a solução proposta neste trabalho se baseia nos gestos feitos por uma única pessoa, ao tentar copiar, o gesto não sai exatamente como feito pelo usuário que gerou o padrão, causando os erros de reconhecimento.

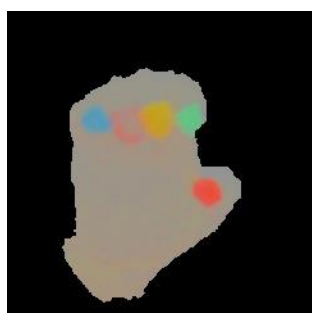


Figura 4.12- Exemplo sinal da “E” feita pelo usuário B

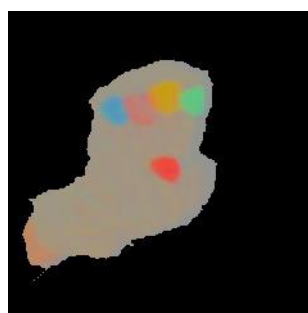


Figura 4.11- Exemplo sinal da “E” feita pelo usuário A

Pelas figuras 4.11 e 4.12 é possível ver uma pequena diferença na posição relativa do marcador vermelho (polegar), o que resulta em um ângulo diferente ao marcador verde (indicador). Assim, nenhum dos modelos gerados por um experimento foi capaz de reconhecer a letra “E” do outro experimento.

5. CONCLUSÃO

Neste capítulo serão expostas as conclusões do trabalho e comentários finais, além de propor formas de continuação da linha de pesquisa relacionada ao projeto.

Este trabalho apresentou uma proposta para geração de padrões baseadas no modelamento geométrico das posturas manuais, para o reconhecimento do alfabeto manual da Libras. Os padrões são aplicados a classificadores baseados em distâncias, e seus resultados para quatro experimentos diferentes são analisados.

Primeiramente, neste trabalho são discutidos alguns aspectos sobre a língua de sinais e a sua carência por tecnologias voltadas para a autonomia e acessibilidade dos surdos. Para esse trabalho também foi feita uma pesquisa das tecnologias disponíveis de interação com o computador que pudessem auxiliar a atingir os objetivos deste projeto, onde se definiu a utilização do sensor Kinect, técnicas de visão computacional e processamento de imagens como formas de aquisição e pré-processamento dos dados.

A escolha se mostrou satisfatória, foi definido o alfabeto em Libras como universo a ser reconhecido automaticamente. Criaram-se então modelos para cada um dos sinais. Os modelos se mostraram eficientes na capacidade de representar diretamente cada um dos sinais com taxa de acerto acima de 90%.

Os testes de reconhecimento automático, apesar de ter uma taxa de acerto semelhante à encontrada no teste com os modelos tiveram, muitas falhas no tratamento inicial das imagens, falhas que podem ser corrigidas com técnicas mais eficazes para o rastreamento dos marcadores e na técnica de reconhecimento mais robustas.

Considerando que o objetivo principal desse trabalho foi verificar a capacidade de se reconhecer sinais utilizando visão computacional usando características geométricas dos sinais e não na comparação de modelos de imagem, a solução se mostrou eficiente na sua capacidade de reconhecer posturas manuais mesmo com variações das distâncias para os sensores, rotações e oclusão de marcadores.

Na tarefa de reconhecer automaticamente os sinais da Libras, este trabalho é apenas um pequeno passo, mas que tem o seu valor, pois abordou o problema e

trouxe uma proposta de solução, mesmo que restrita apenas para gestos estáticos do alfabeto manual. A complexidade da língua de sinais apresentou-se muito maior do que inicialmente se esperava.

Por fim, acredita-se que esse trabalho de graduação tenha atingindo grande parte dos objetivos determinados e que possa servir de base para muitas pesquisas futuras na associação de tecnologias e acessibilidade.

Algumas sugestões de trabalhos futuros são listadas abaixo.

- Desenvolver um algoritmo eficiente para a solução do problema do duplo reconhecimento de uma letra.
- Ampliação do universo de sinais reconhecidos, utilizando todas as configurações de mão listadas em Libras.
- Realizar os testes e a criação dos modelos com um banco de dados maior para cada gesto, usando gestos de vários usuários a fim de tentar minimizar a dependência do sistema ao usuário.
- Desenvolver uma interface para reconhecimento dos sinais em tempo real.
- Utilizar técnicas iterativas e de otimização para aperfeiçoar a criação e atualização dos modelos dos sinais.
- Desenvolver técnicas mais eficientes para a localização dos marcadores, usando, por exemplo, uma calibração inicial.
- Utilizar uma taxa de captura de imagens mais alta que 30 quadros por segundo, a fim de aumentar a qualidade das imagens das posturas manuais de gestos dinâmicos ("X", "Z", "K" e "J") e fazer o reconhecimento destes gestos.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] **G. Bebis, F. Harris, A. Erol, e B. Yi.** Development of a nationally competitive program in computer vision technologies for effective human computer interaction in virtual environments. Space Grant, EPSCoR Annual Meeting, 2002.
- [2] **SOUSA, Ana Paula de Almeida,** Interpretação da língua gestual portuguesa. Universidade de Lisboa, Tese de mestrado, 2012.
- [3] **OLIVEIRA, P. M. T. D. Sobre Surdos.** Site de Jonas Pacheco, 2011. Disponível em: <<http://www.surdo.org.br/informação.php?lg=pt&info=Historiadossurdos>>.
- [4] **PACHECO, J., ESTRUC, E. e ESTRUC, R.** Curso Básico de Libras. Disponível em: <www.surdo.org.br>, v. 8, 2008.
- [5] **H. Pistori e J. J. Neto.** An experiment on handshape sign recognition using adaptive technology: Preliminary results. In Lecture Notes in Artificial Intelligence 2004.
- [6] **ARAÚJO, A. P.** InfoEscola. Linguagem de Sinais Brasileiras (Libras), 2007. Disponível em: <<http://www.infoescola.com/portugues/lingua-brasileira-de-sinais-libras/>>.
- [7] **Félix, Rayanne.** Os cinco Parâmetros: Libras, 2010. Disponível em: <<http://librasitz.blogspot.com.br/2010/07/os-cinco-parametros.html>>.
- [8] **GUGELMIN, F.** TECMUNDO, 2011. Disponível em: <<http://www.tecmundo.com.br/10421-asus-anuncia-o-wavi-xtion-sistema-de-captura-de-movimentos-para-pc.htm>>
- [9] **LAMAR, M. V.,** Hand Gesture Recognition using T-CombNET: A New Neural Network Model. IEICE Transactions on Information and Systems, Japão. 2000.
- [10] **DA SILVA, JUAREZ PAULINO.** A study of the ICP algorithm for recognition of the hand alphabet. In: 2013 Latin American Computing Conference (CLEI), 2013.
- [11] **BRADSKI, D. G. R.; KAEHLER, A.** Learning opencv, 1st edition. First. [S.l.]: O'Reilly Media, Inc.
- [12] **Marengoni, M; Stringhini, D.** Tutorial: Introdução à Visão Computacional usando OpenCV, 2010.
- [13] **Ana Paula Bento Teixeira.** Kinlib: protótipo em ensino de libras utilizando o kinect, 2011.
- [14] **François Malric.** Artificial Neural Networks for Real-Time Optical Hand Posture Recognition Using a Color-Coded Glove, 2008.
- [15] **Yuan Yao.** Real-time hand pose estimation from RGB-D sensor, EUA. 2012.

[16] **Trindade, Pedro.** Hand gesture recognition using color and depth images enhanced with hand angular pose data. 2012.

ANEXOS

I. DESCRIÇÃO DO CONTEÚDO DO CD

Os arquivos, diretórios e subdiretórios do CD são divididos da seguinte forma:

- Resumo.pdf
- Trabalho de Graduação – Versao Final.pdf
- Apresentação
- Codigos
 - Captura
 - Reconhecimento
- Banco de dados

II. INSTRUÇÕES PARA O PROGRAMA

Todos os programas foram testados na seguinte plataforma:

- Sistema operacional Windows 7.
- OpenCV versão 2.1.
- SDK kinect versão 1.5.

Todos os programas desenvolvidos ao longo do projeto estão na pasta “Códigos”, estão divididos em captura e reconhecimento, para a captura é necessário o uso do sensor Kinect e da câmera. Para fazer o reconhecimento não é necessário os sensores, porém é preciso a utilização do banco de dados contidos na pasta “ banco de dados”.