

UNIVERSIDADE SAGRADO CORAÇÃO

MURILO DE OLIVEIRA E SOUZA

**DETECÇÃO E SEGMENTAÇÃO DE
ELEMENTOS GESTUAIS DE LIBRAS COM O USO
DO MICROSOFT KINECT**

BAURU

2013

MURILO DE OLIVEIRA E SOUZA

**DETECÇÃO E SEGMENTAÇÃO DE
ELEMENTOS GESTUAIS DE LIBRAS COM O USO
DO MICROSOFT KINECT**

Trabalho de conclusão de curso
apresentado ao Centro de Ciências Exatas e
Sociais Aplicadas como parte dos requisitos
para obtenção do título de Bacharel em
Ciência da Computação, sob orientação do
Prof. Me. Patrick Pedreira Silva.

BAURU

2013

Souza, Murilo de Oliveira e
S7294d

Detecção e segmentação de elementos gestuais de Libras com o uso do Microsoft Kinect / Murilo de Oliveira e Souza -- 2013.
113f. : il.

Orientador: Prof. Me. Patrick Pedreira Silva.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.

1. Detecção de sinais. 2. Microsoft Kinect. 3. Libras. 4. TI. I. Silva, Patrick Pedreira. II. Título.

MURILO DE OLIVEIRA E SOUZA

**DETECÇÃO E SEGMENTAÇÃO DE
ELEMENTOS GESTUAIS DE LIBRAS COM O USO
DO MICROSOFT KINECT**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob orientação do Profº. Me. Patrick Pedreira Silva.

Banca examinadora:

Profº. Me. Patrick Pedreira Silva
Universidade Sagrado Coração

Profº. Dr. Elvio Gilberto da Silva
Universidade Sagrado Coração

Prof^a. Dra. Patricia Bellin Ribeiro
Universidade Sagrado Coração

DEDICATÓRIA

Dedico este trabalho em primeiro lugar a Deus, pois Ele é o motivo de estarmos aqui, lutando e vencendo sempre. Sem Ele nada seríamos, e nada teríamos.

Dedico este trabalho aos meus pais Luiz Carlos e Andrea, que me proporcionaram uma base para que eu pudesse realizar muitas conquistas, aos meus amigos e todos que acreditaram em mim, e que um dia conseguiria realizar tal ato e chegar à algum lugar.

AGRADECIMENTOS

Gostaria de agradecer em primeiro lugar a todos aqueles que de alguma forma doaram um pouco de si para que a conclusão deste trabalho se tornasse possível.

Ao meu professor orientador, Prof. Me. Patrick Pedreira Silva pelo auxílio tanto na vida acadêmica como profissional, disponibilidade de tempo e material, e acima de tudo muito empenho e dedicação para que se tornasse possível à realização deste trabalho.

RESUMO

Hoje em dia a tecnologia contribui em diversas áreas graças ao avanço tecnológico. A todo o momento surgem novos softwares, ferramentas e aplicativos criados com a finalidade de ajudar o ser humano. Diante deste quadro, fica claro a importância de desenvolver softwares que auxiliem uma pessoa com deficiência auditiva na comunicação e no acesso aos computadores, contribuindo para a sua inclusão na sociedade. Considerando que a principal forma de comunicação dos deficientes auditivos se dá por meio de sinais, a hipótese é que um software projetado de forma a privilegiar esta maneira de interação, pode melhorar o processo de interação entre estes indivíduos e os computadores. Diante da complexidade de implementação de um sistema completo de reconhecimento de gestos, o objetivo desta pesquisa foi desenvolver uma aplicação que seja responsável pelo módulo de segmentação de um sistema de reconhecimento de gestos para auxílio aos portadores de deficiência auditiva que seja capaz de reconhecer gestos referentes a um conjunto restrito da LIBRAS, especificamente os números de 0 a 4, utilizando a tecnologia Microsoft Kinect. Com o objetivo de avaliar o sistema, foram testados cinco sinais (gestos) em dois ambientes de teste (luz artificial e luz natural), avaliando-se o desempenho do sistema em seis casos de testes para cada sinal. Os resultados mostraram que o sistema é capaz de reconhecer os gestos realizados, entretanto, existe uma influência direta da iluminação do ambiente, no processo de reconhecimento.

Palavras-chave: Detecção de sinais. Microsoft Kinect. LIBRAS. TI

ABSTRACT

Nowadays, technology contributes too many different fields thanks to its improvement. Every moment a new software, tool, or programs especially designed to help the human being. In front of that, the importance of softwares development that help in the communication and computer accessibility of people with hearing impairment becomes clear, for their inclusion in society. Considering the fact that most people with hearing loss communicate through signals, the hypothesis of projecting a software grants a privilege to this interaction by improving the process between computer and people. In the face of the complexity of the implementation of a whole gesture recognition system, the reason for this research was to develop an application which is responsible for the module of segmentation of that type of system, to auxiliate people with hearing impairment by being capable of recognizing different and restricted gestures from LIBRAS (Brazilian Signal Language), especially from numbers 0 to 4, using the Microsoft Kinect technology. In order to evaluate the system, we tested five signs (gestures) in two test environments (artificial light and natural light), evaluating system performance in six test cases for each signal. The results showed that the system is able to recognize gestures made, however, there is a direct influence of the ambient light in the recognition process.

Key words: Signal detection, Microsoft Kinect, LIBRAS, IT.

LISTA DE ILUSTRAÇÕES

Figura 1 – Alfabeto gestual de números usado na LIBRAS.....	28
Figura 2 - Fases do processo unificado	37
Figura 3 - Componentes do sensor Kinect.....	40
Figura 4 - Fórmula de subtração pixel a pixel.	46
Figura 5 - Exemplo de resultado da subtração simples, sendo a) o modelo de fundo, b) a imagem atual e c) o resultado do algoritmo.	47
Figura 6 - Probabilidade de observar um certo valor x de um pixel no tempo t utilizando um conjunto (K) de gaussianas multivariadas.	49
Figura 7 - Fórmula para máscara binarizada.	50
Figura 8 - Fórmula máscara binarizada contendo os pixels classificados como sombra.	51
Figura 9 - Exemplo de correção de White Balance.....	53
Figura 10 - Operação de fechamento.	56
Figura 11 - Gesto que representa o número 0.....	59
Figura 12 - Gesto que representa o número 1.....	60
Figura 13 - Gesto que representa o número 2.....	61
Figura 14 - Gesto que representa o número 3.....	61
Figura 15 - Gesto que representa o número 4.....	62
Figura 16 - Funcionamento do sistema proposto.....	64
Figura 17 - Atribuição do dispositivo e câmera RGB.	67
Figura 18 - Divisão da imagem da câmera em pixels e criação das BitmapImages. 67	
Figura 19 - Controle de ativação do sensor com o uso da classe KinectSensor.....	68
Figura 20 - Algoritmo de captura da imagem para o formato jpg.	69
Figura 21 - Carregar imagens na tela e chamada do método de suavização.	69
Figura 22 - Algoritmo de suavização parte 1.....	71
Figura 23 - Algoritmo de suavização parte 2.....	72
Figura 24 - Algoritmo de suavização parte 3.....	72
Figura 25 - Algoritmo de suavização parte 4.....	73
Figura 26 - Variáveis usadas no algoritmo de subtração de fundo.	74
Figura 27 - Primeira estrutura de repetição para subtração de fundo.....	75
Figura 28 - Segunda estrutura de repetição para subtração de fundo.....	76

Figura 29 - Terceira estrutura de repetição para subtração de fundo.....	77
Figura 30 - Método para salvar imagem final.....	77
Figura 31 - Algoritmo de binarização parte 1.....	78
Figura 32 - Algoritmo de binarização parte 2.....	79
Figura 33 - Linhas imaginárias criadas para realizar a detecção do sinal.	80
Figura 34 - Variáveis utilizadas no algoritmo de identificação.	81
Figura 35 - Verificação da quantidade de dedos na linha horizontal.	82
Figura 36 - Verificação da quantidade de dedos na linha vertical esquerda.	83
Figura 37 - Verificação da quantidade de dedos na linha vertical direita.	84
Figura 38 - Soma total da quantidade de dedos e atribuição à Label.....	84
Figura 39 - Casos de testes utilizados para os sinais referentes aos dígitos 0 e 1.....	86
Figura 40 - Casos de testes utilizados para os sinais referentes aos dígitos de 2 a 4.87	
Figura 41 - Teste com o sinal de número um em um ambiente sob luz artificial.	88
Figura 42 - Teste com o sinal de número dois em um ambiente sob luz solar.....	88
Figura 43 - Modelo 1 (Sinais emitidos em ambiente sob luz artificial).	89
Figura 44 - Modelo 2 (Sinais emitidos em ambiente sob luz solar).	89
Figura 45 - Casos de testes utilizados para os sinais referentes aos dígitos 0 e 1 (apenas sob iluminação artificial).....	90
Figura 46 - Casos de testes utilizados para os sinais referentes aos dígitos de 2 a 4 (apenas sob iluminação artificial).	91
Figura 47 - Análise referente à posição do sinal emitido (apenas casos sob luz artificial).	92
Figura 48 - Análise referente à distância entre os dedos (apenas casos sob luz artificial).	92
Figura A. 1 - Tela inicial do sistema.	97
Figura A. 2 - Botão "iniciar" da tela inicial.	97
Figura A. 3 - Botão "iniciar" da tela inicial.	98
Figura A. 4 - Tela de menu do sistema.....	98
Figura A. 5 - Botão "Capturar" da tela de menu.....	99
Figura A. 6 - Botão "Segmentar" da tela de menu.....	99
Figura A. 7 - Botão "Identificar" da tela de menu.	99
Figura A. 8 - Botão "Sair" da tela de menu.	100

Figura A. 9 - Tela de captura de imagem do sistema.....	101
Figura A. 10 - Botão "Ligar" da tela de captura.	101
Figura A. 11 - Botão "Capturar" da tela de captura.	101
Figura A. 12 - Botão "Desligar" da tela de captura.....	102
Figura A. 13 - Botão "Sair" da tela de captura.	102
Figura A. 14 - Tela de segmentação do sistema.	103
Figura A. 15 - Botão "Abrir" da tela de segmentação.....	103
Figura A. 16 - Botão "Segmentar" da tela de segmentação.....	103
Figura A. 17 - Botão "Sair" da tela de segmentação.....	103
Figura A. 18 - Tela de identificação do sistema.	104
Figura A. 19 - Botão "Abrir" da tela de identificação.....	104
Figura A. 20 - Botão "Identificar" da tela de identificação.	104
Figura A. 21 - Botão "Sair" da tela de identificação.....	104

SUMÁRIO

LISTA DE ILUSTRAÇÕES

1. INTRODUÇÃO.....	14
1.1 OBJETIVOS	17
1.1.1 Objetivo Geral	17
1.1.2 Objetivo Específico.....	17
2 - DEFICIÊNCIA AUDITIVA.....	19
2.1 - SURDEZ E SOCIEDADE.....	21
2.2 - DEFICIÊNCIA AUDITIVA E EDUCAÇÃO	26
2.3 - SURDEZ E TECNOLOGIAS.....	30
3 - ENGENHARIA DE SOFTWARE.....	32
4 - TECNOLOGIAS MICROSOFT.....	39
4.1 - MICROSOFT KINECT ®.....	39
4.2 - LINGUAGEM DE PROGRAMAÇÃO C SHARP (C#).....	42
4.3 MICROSOFT XNA.....	43
4.4 BANCO DE DADOS SQL SERVER	43
5 – SEGMENTAÇÃO.....	45
5.1 – CONSIDERAÇÕES INICIAIS	45
5.2 SUBTRAÇÃO DE FUNDO (BACKGROUND SUBTRACTION).....	45
5.2.1 Subtração Simples.....	46
5.2.2 <i>Running Gaussian Average</i>	47
5.2.3 Temporal Median Filter	48
5.2.4 Mistura de Gaussianas (<i>mixture of gaussians</i>).....	48
5.2.5 Principais Problemas da Detecção de Movimento por Background Subtraction.....	49
5.2.5.1 Alterações de Brilho (highlights).....	50
5.2.5.2 Sombras	51

5.2.5.3 Fantasmas (ghosts)	51
5.2.5.4 Câmeras de Configuração Automática.....	52
5.2.5.5 Regra de Decisão para Atualizar Modelo de Fundo	53
5.2.6 Técnicas de Pós-Processamento	54
5.2.6.1 Remoção de Ruídos.....	54
5.2.6.2 Identificação de Blobs.....	54
5.2.6.3 Area Thresholding.....	55
5.2.6.4 Morphological Closing.....	55
5.2.6.5 Saliency Test.....	56
5.2.6.6 Optical Flow Test	57
6 – METODOLOGIA	58
7 – DESENVOLVIMENTO.....	66
7.1 – ALGORITMOS.....	66
7.1.1 – ALGORITMO DE EXIBIÇÃO DE IMAGEM.....	66
7.1.2 – ALGORITMO DE CAPTURA DE IMAGEM	68
7.1.3 – ALGORITMO PARA CARREGAR IMAGENS NA TELA	69
7.1.4 – ALGORITMO DE SUAUIZAIÇÃO	70
7.1.5 – ALGORITMO DE SUBTRAÇÃO DE FUNDO.....	73
7.1.6 – ALGORITMO DE BINARIZAÇÃO	77
7.1.7 – ALGORITMO DE IDENTIFICAÇÃO.....	79
8 – RESULTADOS.....	85
9 – CONCLUSÕES.....	93
REFERÊNCIAS.....	94
APÊNDICE A: MANUAL DO SISTEMA	97

1. INTRODUÇÃO

A surdez é a perda ou o enfraquecimento do sentido da audição, há diferentes graus de perda de audição, uma pessoa pode ouvir alguns ruídos, outras ter apenas dificuldade em compreender algumas palavras, e também, pode não ouvir nada, esta diferença varia de pessoa para pessoa. A deficiência pode ser ocasionada por diversas maneiras, uma pessoa pode nascer surda, adquirir por contração de doenças que afetam o aparelho auditivo, ou então, sofrer alguma lesão que danifique o ouvido, como um barulho muito forte, ou exposição a som alto por um longo período de tempo. Cerca de 9,7 milhões declaram ter deficiência auditiva (5,1%). A deficiência auditiva severa foi declarada por mais de 2,1 milhões de pessoas. Destas, 344,2 mil são surdas e 1,7 milhão de pessoas têm grande dificuldade de ouvir (G1, 2013).

Segundo Kojima (2008) apud Zoliani (2012), as pessoas com deficiência auditiva foram prejudicadas por não exercitar adequadamente a comunicação, impossibilitando, assim, os surdos de não conseguirem se comunicar. A audição está altamente ligada aos fatores de equilíbrio emocional e também na orientação espacial do homem, já que os órgãos dos sentidos são muito importantes para o desenvolvimento das células e centros nervosos.

Neste contexto, os surdos-mudos são excluídos da sociedade e acabam perdendo a fala por falta de não ter uma educação adequada ou, então, por não terem um contato direto com a fala, mas este não é um quadro geral. “Existem surdos que conseguem aprender a falar e outros que aprendem a Língua dos Sinais. Estes desenvolvem a habilidade espacial no cérebro de forma mais significativa que o outro” (ZOLIANI, 2012, p. 9).

Portanto, é necessária a utilização de meios eficazes para se obter uma comunicação de qualidade, para a pessoa se sentir parte do todo e não um cidadão excluído e incomunicável. O estímulo visual é muito utilizado na linguagem de sinais, pois o receptor da mensagem precisa estar atendo aos movimentos que o emissor faz para que haja uma comunicação eficaz e sem ruídos.

As línguas de sinais distinguem-se das línguas orais porque se utilizam de um meio visual-espacial e oral-auditivo, ou seja,

na elaboração das línguas de sinais é necessário olhar os movimentos que o emissor realiza para entender sua mensagem. Já na língua oral, é imprescindível apenas ouvi-lo, sem a necessidade estar olhando para ele (HONORA; FRIZANCO, 2009, p. 41 apud ZOLIANI, 2012, p. 8).

Sendo assim, pode-se afirmar que os surdos utilizam mais a habilidade visual para trabalhar corretamente a comunicação. A pessoa que possui essa deficiência necessita de meios mais específicos que trabalhem o modo visual, assim, estes poderão ter maior capacidade de compreender o que é necessário.

A comunicação através dos sinais possui regras morfológicas, que trabalham com a estrutura e forma das palavras, a sintática, que estuda a linguagem sob o aspecto formal, a semântica, que estuda a palavra e seus significados, e pragmática, que são os fatores contextuais e linguísticos de uma oração, que auxiliam no processo de aprendizagem da pessoa que possui a deficiência auditiva, fazendo com que esta tenha uma vida normal em sociedade (KOJIMA, 2008 apud ZOLIANI, 2012).

A possibilidade de ter um desenvolvimento mais natural pode favorecer o processo educacional da criança surda. Esta é uma maneira de aproveitar o potencial dos Surdos, pois eles estabelecem de uma forma visual/espacial, relações diretas com a imagem que o pensamento lhes permite (ZOLIANI, 2012, p. 9).

Hoje em dia a tecnologia contribui em diversas áreas graças ao avanço tecnológico. A todo o momento surgem novos softwares, ferramentas e aplicativos criados com a finalidade de ajudar o ser humano. Diante deste quadro, fica claro a importância de desenvolver softwares que auxiliem uma pessoa com deficiência auditiva na comunicação e no acesso aos computadores, contribuindo para a sua inclusão na sociedade.

Portanto, a pessoa surda que tiver a oportunidade de trabalhar com softwares voltados à questão da deficiência auditiva, possivelmente, ampliará suas possibilidades de integração na sociedade ouvinte. Considerando-se que essa integração à sociedade é algo essencial, softwares voltados para este público-alvo se tornam bastante relevantes. Desta forma os softwares podem atuar como recurso auxiliar de comunicação. Neste

contexto, esta pesquisa pretende culminar no desenvolvimento de uma ferramenta que possa representar uma das etapas (módulo de segmentação) de um sistema completo de detecção e interpretação de elementos gestuais que compõe a LIBRAS (Língua Brasileira de Sinais).

Um sistema de reconhecimento de gestos completo para a língua de sinais exige a criação de um software composto por quatro módulos principais:

- Módulo de segmentação (modelagem da cena no campo de visão da câmera, buscando separar as partes do corpo humano dos objetos ou regiões pertencentes ao fundo)
- Módulo de modelagem e identificação (agrupamentos de pixel segmentados a partir de uma característica comum, são agrupados e identificados de acordo com um modelo pré-determinado de partes do corpo humano)
- Módulo de *tracking* (tarefa de rastreamento de movimento de regiões de interesse detectadas através de modelos estatísticos para previsão de próximas posições)
- Módulo de reconhecimento e interpretação (consiste em fornecer um resultado final para o sistema, isto é, o gesto da língua de sinais que foi reconhecido)

Considerando que a principal forma de comunicação dos deficientes auditivos se dá por meio de sinais, a hipótese é que um software projetado de forma a privilegiar esta maneira de interação, pode melhorar o processo de interação entre estes indivíduos e os computadores. Diversas tecnologias podem auxiliar quanto a isso como, por exemplo o Kinect. O dispositivo Microsoft Kinect consiste em um aparelho capaz de reconhecer movimentos. Ele, a partir de suas câmeras com sensores de movimento, é capaz de captar e interpretar movimentos das mãos e rosto das pessoas, podendo transformá-los em texto e até em voz, sendo muito útil em situações que necessitem deste tipo de interação.

Trata-se de um complemento para o console 'X-Box 360®' que, através de câmeras, capta os movimentos do(os) jogador(es) dentro de

um espaço específico, interpreta estes movimentos e transforma-os em comandos que são transmitidos ao console para proporcionar a interação entre o(os) jogador(es) e o jogo em si (ROCHA; DEFAVARI; BRANDÃO, 2012, p.17).

Sendo assim, diante da complexidade de implementação de um sistema completo de reconhecimento de gestos, o objetivo desta investigação é desenvolver uma aplicação que seja responsável pelo módulo de segmentação do sistema, utilizando como base o Microsoft Kinect para captar e segmentar as imagens referentes a um subconjunto de gestos da LIBRAS. Este trabalho está estruturado em 8 capítulos. O capítulo 2 apresenta um breve histórico sobre a deficiência auditiva. No capítulo 3, é apresentada a surdez em meio à sociedade. No capítulo 4, a surdez na educação é apresentada. No capítulo 5, são apresentadas as tecnologias ligadas à deficiência auditiva. No capítulo 6, é tratada a engenharia de software. No capítulo 7, as tecnologias Microsoft são apresentadas. No capítulo 8 é apresentada a metodologia para o desenvolvimento do trabalho.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver uma aplicação que seja responsável pelo módulo de segmentação de um sistema de reconhecimento de gestos para auxílio aos portadores de deficiência auditiva que seja capaz de reconhecer gestos referentes a um conjunto restrito da LIBRAS, especificamente os números de 0 a 4, utilizando a tecnologia Microsoft Kinect.

1.1.2 Objetivo Específico

- Fazer um levantamento de informações sobre a linguagem brasileira de sinais (LIBRAS);

- Conhecer o funcionamento do dispositivo Microsoft Kinect em conjunto com a linguagem de programação C# e a ferramenta Microsoft XNA;
- Estudar algoritmos de segmentação de imagens e selecionar aquele que melhor se adequa ao projeto;
- Desenvolver um subsistema para capturar e segmentar imagens referentes a gestos com o uso do Microsoft Kinect;

2 - DEFICIÊNCIA AUDITIVA

“Existem muitas crianças que podem ter nascido com a deficiência auditiva ou ter perdido a audição após o nascimento. Quando isso acontece, ela deve ter muitas dificuldades para adquirir a linguagem oral”. (COUTO, 1985, p. 11). Isso ocorre, pelo fato da criança nunca ter entrado em contato com a comunicação para conseguir entendê-la. Quando uma pessoa já possui a comunicação e perde a audição, é muito mais fácil mantê-la.

De acordo com Godinho (1982, p. 15), o homem utiliza todos os seus sentidos para se comunicar, pois é através dos estímulos que é possível captar informações, percepções e conceitos que ajudam na construção de um pensamento e na estrutura psicológica e social de uma pessoa. Sendo assim, “a falta ou a danificação de um dos receptores sensoriais altera o funcionamento e a integração dos demais, na razão direta da ausência ou da deficiência da reação perceptual específica aos atributos correspondentes do objeto”. Portanto, a deficiência auditiva causa problemas em diversas áreas que envolvem o ser humano e não apenas na parte afeta. O trabalho realizado com uma pessoa surda vai além dos problemas com a fala e a comunicação, pois este precisa de cuidados com a interação em sociedade e também no âmbito psicológico. “A deficiência auditiva, um tipo de privação sensorial, considerada aqui introdutoriamente sob o tríplice aspecto: físico, psicológico e social, é uma das mais significativas limitações da oportunidade de crescimento do indivíduo”. O ser humano passa a se sentir limitado e diferente das pessoas normais, trazendo sentimentos de vergonha de si próprio de incapacidade, levando o surdo ao isolamento. Por isso, o trabalho de acompanhamento psicológico correto é tão importante quanto o cuidado com a educação e a integração. “Esta privação de dados sensoriais cria uma lacuna nos processos psicológicos de integração da experiência, afetando o equilíbrio e a capacidade normal de desenvolvimento do indivíduo. O deficiente, portanto, em decorrência do instrumental sensório reduzido, fica restringido pela carência de informações necessárias à plenitude das operações da inteligência”.

“Surdez s.f. 1. Privação ou enfraquecimento considerável do sentido da audição; ensurdecência, ensurdecimento. – 2. Estado

de surdo. – 3. Surdez cortical, surdez causada por uma lesão do córtex cerebral. II Surdez labiríntica, surdez devida à doença do labirinto. II Surdez verbal, distúrbio da percepção auditiva que consiste na dificuldade de compreensão de sons linguísticos”. (GRANDE ENCICLOPÉDIA LAROUSSE CULTURAL, 1995, p. 5.544).

Segundo Godinho (1982), um indivíduo pode nascer com deficiência auditiva (congenita) ou adquirir por degeneração, por meio de doenças. A congênita pode ser endógena, herança genética, ou então, exógena, quando a mãe contrai alguma doença durante o período de gestação, que afeta o aparelho auditivo do feto. Problemas de incompatibilidade sanguínea e partos prematuros também podem afetar o sentido de audição. Porém, ainda podem ocorrer problemas de malformação depois que a criança nasce, durante o desenvolvimento, ocasionando a atrofia ou inexistência do aparelho auditivo. Ainda na infância, a criança corre o risco de surdez por contração de doenças como: meningite, sinusite, furúnculos e diversos antibióticos que possam afetar o canal de audição. Já na fase adulta, a perda auditiva está ligada à poluição sonora, gerando traumas por grande exposição sonora.

Funcionalmente, entretanto, os sentidos atuam reciprocamente, mediante um processo de percepção intersensorial conhecido por sinestesia, em razão do qual o conteúdo percebido é o resultado da elaboração mental das sensações elementares provenientes dos diversos sentidos, que transmitidas pela via neural própria difundem-se pela área da associação do córtex cerebral e são interpretadas como experiência sensorial integrada pelo conjunto de experiência sensitiva (SOUSA, 1982, p. 18).

Ainda para o autor acima (1982, p. 19), há uma interferência na percepção quando um dos sentidos é afetado. Quando isso ocorre, um dos sentidos que não foi afetado será recompensado fisiologicamente para suprir essa deficiência. “Por serem vista e ouvido receptores de distância, existe maior reciprocidade funcional entre ambos, operando a audição mais como sentido de fundo, enquanto a visão se ocupa dos primeiros planos”. Portanto, uma pessoa terá melhor percepção visual em decorrência da audição afetada. A surdez pode ainda afetar a percepção alterando a consciência da pessoa, tanto no mundo que está a sua volta como no desenvolvimento da

personalidade, afetando ainda o psicológico e o emocional, levando à dificuldade em aprender e se socializar.

[...] o surdo se coloca em inferioridade relativa aos normais em razão da estruturação diferente de sua percepção e a conseqüente alteração da inteligência, personalidade, nível emocional, habilidade psicomotriz, verbalização e sociabilidade, o que se reflete no comportamento global do indivíduo, acentuando o isolamento e os níveis de dependência (SOUSA, 1982, p. 25).

Por isso, quanto antes uma criança for diagnosticada com a deficiência, melhor e mais rápido será o seu tratamento, podendo trabalhar corretamente a linguagem e o desempenho cognitivo, social e emocional. “O diagnóstico audiológico é de extrema importância uma vez que norteia os procedimentos de habilitação e reabilitação do indivíduo surdo” (SANTOS; LIMA; ROSSI, 2003, p. 37).

2.1 - SURDEZ E SOCIEDADE

Qualquer pessoa pode ter deficiência auditiva, já que esta não escolhe entre classe social, cor, etnia, etc. Ainda há preconceito, principalmente, pela falta de comunicação, já que a surdez está ligada a falta de linguagem, colocando essa pessoa em uma posição diferente da habitual para as outras, dificultando assim o processo de socialização do indivíduo. Portanto, é importante desenvolver meios que busquem a quebra de barreiras para adaptar diferentes modos de se comunicar para trabalhar com deficientes auditivos, pois a comunicação está altamente ligada ao processo de socialização, que traz conhecimentos e valores culturais. “Os problemas da linguagem irão dificultar a comunicação da criança com a família, sua vida em sociedade, sua aprendizagem na escola. Se a criança não for amada, compreendida e ajudada em suas dificuldades, poderá, ainda, ter problemas emocionais” (COUTO, 1985, p. 11).

Segundo Couto (1985, p. 12), a surdez possui quatro níveis dependendo da perda auditiva: leve, média, severa e profunda. Na surdez leve, a pessoa tem perda de audição de vinte a quarenta decibéis. Esse tipo mais leve de surdez não impede a criança de

desenvolver a linguagem e ter uma educação normal. Porém, esta poderá apresentar problemas com falta de atenção e fazer confusão em sons parecidos de algumas palavras. Já na surdez média, a perda auditiva é de quarenta a setenta decibéis. A pessoa que apresenta esse grau de surdez, geralmente, tem dificuldade em entender o que os outros falam, piorando o quadro se estiver em ambientes com bastante barulho. Além de apresentar problemas como atraso no desenvolvimento da linguagem, juntamente com problemas linguísticos.

A surdez severa é conceituada pela perda de setenta a noventa decibéis. Nesse caso, a pessoa consegue ouvir apenas alguns barulhos, é preciso que se fale bem alto e ela utiliza mais o visual para conseguir entender algo. Na surdez profunda, a perda é maior do que noventa decibéis. A criança é totalmente privada da audição e isso a impede de ter uma linguagem natural, ela consegue apenas emitir alguns sons, que também não é capaz de ouvir. “Fazendo o uso do aparelho e com sua percepção visual sendo desenvolvida, a criança vai melhorando a compreensão e conseqüentemente a qualidade de sua voz e fala, devendo ainda continuar a ser orientada para que cada vez mais aperfeiçoe sua linguagem”.

Ainda para Couto (1985, p. 24). É muito importante a utilização da prótese auditiva no tratamento do deficiente, pois o aparelho ajuda a pessoa até nos piores casos. “A criança, após ser levada para uma instituição própria para sua deficiência, deverá fazer testes e colocação de próteses auditivas. Entendemos que não existe sentido no tratamento sem a adequada protetização”. Porém, é muito importante ressaltar que não é simplesmente utilizar o aparelho de amplificação e ter uma vida normal. É preciso escolher o aparelho certo para determinado nível de perda auditiva, além de ter um tempo de adaptação com profissionais especializados e fazer o uso deste diariamente, assim, o paciente terá melhores resultados. “Esses aparelhos devem ser utilizados segundo indicação médica e sob orientação de profissionais especializados que procederão a educação ou reeducação auditiva, conforme o caso” (COUTO, 1985, p. 37).

Esquemáticamente, a prótese auditiva é constituída de um microfone, que capta os sons provenientes do meio ambiente (entrada) transformando a energia mecânica em elétrica. A seguir o som é enviado ao amplificador e aumentado pelo

mesmo (saída). Graças a esta amplificação, um sinal acústico menor é sensivelmente elevado. Finalmente esse som amplificado vai para o receptor auricular, o qual, à semelhança do microfone, se constitui num transdutor ou transformador de energia. Todo esse processo leva a uma modificação da tensão alternativa em vibrações sonoras amplificadas (COUTO, 1985, p. 37).

Muitos acreditam que este é o melhor método para os surdos, já que o aparelho funciona como uma tentativa de trazer o som para as pessoas que possuem deficiência auditiva, mas na maioria dos casos, o paciente não consegue se adaptar ao aparelho auditivo. “Tanto no caso de criança como de adultos, o principal objetivo da reeducação auditiva é aprender a decodificar os sons através da prótese, utilizando essa aprendizagem para melhor aprender a compreender o que os outros falam” (COUTO, 1985, p. 56).

Segundo Silva, Kauchakje, Gesueli (2003, p. 17), a audição é a primeira etapa da comunicação, graças a esse sentido, conseguimos nos comunicar e compreender as estruturas da língua para assim desenvolver a comunicação, que auxiliará no desenvolvimento social e emocional. “A língua oral é o principal meio de comunicação entre os seres humanos, e a audição participa efetivamente nos processos de aprendizagem de conceitos básicos, até a leitura e a escrita”.

Para Godinho (1982), o contato com outras pessoas é a primeira barreira de limitações quebrada, por isso é importante acostumar o contato social desde criança, estimular meios diferentes de se estabelecer a comunicação, já que está é a maior dificuldade para os surdos.

Qualquer pessoa que possui deficiência auditiva deve aprender a linguagem de sinais, pois está permite a comunicação e a não exclusão. Por isso, é importante a criança entrar em contato com essa linguagem logo após o descobrimento da surdez, assim não afetará o modo de ser, se expressar e se comunicar. A língua de sinais é “composta de movimentos e formatos específicos de mãos, braços, olhos, face, cabeça e postura corporal, que combinados fornecem as características gramaticais necessárias para a formação de uma língua (fonológicas, sintáticas, semânticas e pragmáticas)” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 46). A comunicação, assim como possuir

meios que permitam isso, é direito de todos. O indivíduo que não pode ter acesso a esses meios pode se sentir discriminado ou fora do contexto social. A comunicação deve expandir as possibilidades, tornando o mundo e a linguagem igual e possível para qualquer um. “Para os grupos minoritários, em particular os surdos, a inclusão diz respeito ao exercício de direitos, tais como o do acesso à cidade, aos equipamentos de educação, ao trabalho, à assistência e a previdência social, à saúde, ao lazer e à cultura” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 67). Sendo assim, pode-se afirmar que é de grande importância desenvolver meios que contribuam para a comunicação de pessoas com deficiência auditiva, deixando-os menos excluídos e mais participantes das atividades sociais normais. Porém, “embora a inscrição em leis e a implementação de políticas seja o resultado almejado pelos movimentos sociais que demandam direitos, isso não é o suficiente, pois a garantia do seu exercício e a sua efetividade reside, principalmente, na mudança de valores e atitudes” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 63).

É necessário mudar o modo como algumas pessoas pensam e agem sobre determinados assuntos. Os surdos conquistaram e ainda estão lutando por um espaço na sociedade, mas, infelizmente, não tem como negar que o preconceito ainda existe. “A manifestação mais evidente da surdez é a falta da fala. Por não ter acesso à linguagem oral, a maior parte dos surdos não consegue adquirir uma língua” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 89). Ou seja, a surdez acaba afetando a fala, já que este não tem contato com o mundo dos ouvintes e não tem a possibilidade de captar o modo de se comunicar, porém a deficiência auditiva não tem relação com dificuldade no modo de aprendizagem do indivíduo. “Os surdos conseguem adquirir uma competência na linguagem, principalmente pela Língua de Sinais, permitindo que eles tenham um desenvolvimento como o de qualquer outra criança, em todos os aspectos: linguísticos, emocionais, sociais e cognitivos” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 96).

Segundo Couto (1985, p. 28), a criança começa entender o significado das palavras pelas entonações que usamos no momento da fala, assim esta vai associando certos sons à determinadas situações, como banho, hora de comer, saber que a mãe está brava ou feliz, etc. “Em decorrência da reação das pessoas que a cercam, a criança começa a compreender que os sonsos tem significado e, mais tarde, que as pessoas se comunicam quando emitem sons”. Já as crianças que possuem surdez severa ou

profunda não conseguem captar esse sinal que é emitido e não desenvolvem o código da língua. “Uma criança que não percebe os sons desde que nasce, vai sofrer uma grande perda, já que o conhecimento do mundo sonoro é que possibilita a conquista natural da comunicação oral” (COUTO, 1985, p. 29).

O contato com a linguagem de sinais, desenvolvida especialmente para as pessoas surdas, é de extrema importância tanto para as pessoas com deficiência auditiva como para os familiares. Assim, a criança aprende desde cedo que essa linguagem é normal, além do apoio que a família mostra, e poderá ter uma educação como qualquer outra pessoa, sem atrapalhar o desenvolvimento cognitivo e emocional. “O uso da Língua de Sinais é de fundamental importância, pois é por meio dela que as crianças buscam ler e interpretar o português escrito” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 158).

A comunicação possibilita as pessoas a ter um convívio social normal, poder se comunicar, se expressar, se conhecer e construir amizades. Por isso é preciso encontrar meios para que a comunicação aconteça de forma natural, assim os surdos poderão ter uma vida normal e não irão desenvolver problemas psicológicos que, geralmente, os afetam. “Crianças surdas em contato inicial com a Língua de Sinais necessitam de referências da linguagem visual com as quais tenham possibilidade de interagir, para conseguirem construir significado” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 161). Isso mostra a importância que o software irá ter para com as pessoas com deficiência auditiva. Levando até elas um meio novo de conseguirem se comunicar e se expressar. A imagem é muito importante no modo de aprendizagem do surdo, assim ele pode desenvolver um conhecimento através do que é apresentado. Isso mostra que “o caminho de aprendizagem necessariamente será visual, daí a importância de os educadores compreenderem mais sobre o poder construtivo da imagem, tanto no sentido de ler imagens quanto no de produzi-las” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 169).

Segundo os autores acima (2003, p. 223), o contato visual é muito importante na comunicação do surdo, já que este utiliza mais o sentido da visão. “Qualquer que seja a modalidade de língua que será usada, a atenção para o rosto do interlocutor é pré-requisito para se começar uma conversa”. Por isso, os surdos observam muito o movimento dos lábios para entender o que as pessoas estão dizendo, a linguagem labial

ajudou consideravelmente a comunicação dos surdos com outras pessoas, mas a linguagem de sinais é a mais utilizada pelos deficientes auditivos.

“Na ausência dos estímulos sonoros, a criança vai aprendendo a transferir sua atenção para os estímulos visuais e reagirá de acordo com a expressão do rosto da mãe e os contatos com o seu corpo” (COUTO, 1985, p. 31).

É necessário criar meios para se estabelecer uma comunicação decente para todos, para que não ocorra exclusão de um cidadão. A comunicação está presente em todos os momentos da vida de uma pessoa, quando algo é afetado e parte do comum se perde, é importante criar outro tipo de técnica para que o sujeito não seja prejudicado pela deficiência. “A inclusão social tem como meta, basicamente, criar uma sociedade capaz de acolher todas as pessoas, independente das diferenças e necessidades individuais e, para isso, preconiza soluções para as diferentes identidades existentes nos ambientes sociais” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 235).

2.2 - DEFICIÊNCIA AUDITIVA E EDUCAÇÃO

Os surdos sofreram ao longo dos anos por não conseguirem se comunicar, muitos foram excluídos como cidadãos e tiveram seus direitos roubados. A comunicação correta possibilitou um modo de aprendizagem e comunicação entre as pessoas com esse tipo de deficiência. O modo de comunicação trouxe, além dos direitos, a esperança de aprendizado e autoestima. Hoje em dia, ainda há preconceito, mas muitas barreiras já estão sendo quebradas e os surdos vêm conquistando seu espaço a cada dia, mostrando ser capazes de aprender uma nova língua, a conseguirem fazer uma faculdade, estabelecer relações afetivas e matrimoniais e trabalhar, como qualquer pessoa. “A força da língua brasileira de sinais, suas ricas características e componentes fazem dela um poderoso instrumento linguístico que permite o indivíduo surdo ser amplamente beneficiado com todo o amplo conhecimento humano, inclusive a aquisição de uma segunda língua, mesmo sendo ela oral ou escrita” (CARVALHO; CASTRO, 2009, p. 15).

Os deficientes auditivos eram impossibilitados de ter uma educação convencional. O sistema de aprendizagem normal era ineficiente para os que não conseguiam ouvir. De acordo com Carvalho e Castro (2009), os surdos sofreram terríveis perseguições na antiguidade, muitos eram oferecidos aos Deuses em rituais de sacrifício. Esse preconceito também estava presente no pensamento da igreja, como também por Aristóteles, que acreditava que a fala era o que tornava o ser humano um indivíduo, ou seja, quem não possuía não era considerado um ser normal, portanto, não teria os mesmos privilégios de ter educação, casamento ou qualquer prática de um cidadão.

Segundo Carvalho e Castro (2009, p. 15), no final do século XV, “Giralamo Cardamo, italiano, tentou utilizar sinais de linguagem escrita para ensinar aos surdos. Pedro Ponce de Leon, espanhol, foi um monge beneditino que utilizava, além de sinais, treinamento da voz e leitura dos lábios”.

Para Carvalho e Castro (2009, p. 16), muitos professores tiveram destaca na história com diferentes métodos de educação para surdos. “Alguns defendiam o uso da língua falada, ou seja, um método oral puro enquanto outros priorizavam o uso de um método combinado, que aproveita a língua de sinais para a introdução do ensino da fala”.

Em 1875, Flausino José da Gama, utilizou seus conhecimentos adquiridos no INES para lançar e divulgar pelo Brasil seu livro *Iconographia dos Signaes dos Surdos-Mudos*, que trazia desenhos usados para a comunicação dos surdos.

Em 1880, foi aprovado no Congresso Mundial de Professores de Surdos, em Milão, que o método de oral seria o único utilizado para ensinar os deficientes auditivos.

No Brasil, o Instituto Nacional de Educação de Surdos, o INES, que foi fundado pelo professor francês Hernest Huet em 1857, ou seja, antes deste Congresso Mundial, foi beneficiado devido a esse professor defender o método combinado, assim como o professor Charles Michel l'Épée defendia. (CARVALHO; CASTRO, 2009, p. 16).

De acordo com Rocha, informações retiradas do site do Instituto Nacional de Educação de Surdos (INES), esta foi a primeira instituição especializada em educar os deficientes auditivos no Brasil, por isso, pessoas do País inteiro e também de outros países vinham para o Rio de Janeiro para ter uma educação de qualidade, se

profissionalizar e se socializar. “A partir desse instituto, deu-se o início à criação da Língua Brasileira de Sinais (LIBRAS), mistura da Língua de Sinais Francesa com os sistemas de comunicação já usados pelos surdos das mais diversas localidades”. (CARVALHO; CASTRO, 2009, p. 16). A Figura 1 mostra os gestos referentes aos números de 0 a 9 da Língua Brasileira de Sinais (LIBRAS).



Figura 1 – Alfabeto gestual de números usado na LIBRAS.
Fonte: Jogo somando em LIBRAS. Acesso em: 19/05/2013.

Ainda para os autores (2009), em 1896, um consagrado professor do INES, A. J. de Moura e Silva, foi até o Instituto Francês de Surdos para estudar o método oral escolhido como o mais eficaz pelo Congresso de Milão e chegou a conclusão que o método não atendia com suas expectativas. Por isso, hoje o método mais utilizado no mundo é o oral. A luta pelos direitos à educação para os deficientes auditivos gerou diversas instituições de aprendizado e leis pelo mundo todo.

Em 1970, a professora Ivete Vasconcelos introduziu a comunicação total no Brasil. A partir de 1980, os resultados dos estudos na área das línguas de sinais e a educação dos surdos,

[...] apontam para a filosofia da educação bilíngue, preconizando o acesso da criança surda, o mais cedo possível, tanto à língua de sinais quanto à língua oral de seu país. O oralismo puro, o bilinguismo e a comunicação total são conceitos que, ainda, perduram em nosso país, dando diversos resultados positivos em vários campos. CARVALHO; CASTRO, 2009, p. 16).

O INES foi uma grande conquista dos surdos no Brasil, mostrando a capacidade de aprendizado e que é possível a comunicação para todos. De acordo com informações do próprio site do INES, hoje, “além de oferecer no seu Colégio de Aplicação, Educação Precoce (de zero a três anos), Ensino Fundamental e Médio, oferece também Ensino Superior através do Curso Bilíngue de Pedagogia, experiência pioneira na América Latina”. O Instituto busca, além de educar e preparar a pessoa portadora de deficiência auditiva, socializar e trazer caminhos para um ensino de qualidade visando a profissionalização. A educação é um direito de todos, assim como a comunicação, por isso o objetivo de toda essa batalha dos surdos é oferecer uma educação de qualidade e também direcionar esses cidadãos para a área profissionalizante, garantindo um futuro sólido.

Seu principal foco, como órgão do Ministério da Educação (MEC), é a produção, o desenvolvimento e a divulgação de conhecimentos científicos na área da surdez em todo o território nacional, bem como subsidiar a Política Nacional da Educação, na perspectiva de promover e assegurar o desenvolvimento global da pessoa surda, sua plena socialização e o respeito às suas diferenças (CARVALHO; CASTRO, 2009, p. 17).

A Língua Brasileira de Sinais (LIBRAS) é a língua dos Surdos, fundamentada pela Lei Federal nº. 10.436, de 24 de abril de 2002 e regulamentada pelo decreto 5.626, de 22 de dezembro de 2005.

Segundo Carvalho e Castro (2009, p. 23), a linguagem brasileira de sinais, assim como a língua falada, sofre variações na mudança de estado, mudando também em cada país. Com LIBRAS é possível uma pessoa sinalizar as regras da língua portuguesa, porém esse método não é eficaz para os surdos, já que estes se utilizam mais da parte visual para captar informações. “Esse aparente pequeno detalhe propicia uma enorme diferença de estrutura de pensamento entre os surdos e os ouvintes e é, exatamente, por isso que se torna difícil, não impossível, para muitos surdos o aprendizado da língua portuguesa”. Por isso é necessário uma estrutura de pensamento e, também, gramatical exclusiva da linguagem brasileira de sinais, feita para pessoas que possuem deficiência auditiva.

Ainda segundo os autores já citados, é possível entender como funciona a parte visual, de como é a comunicação por meio da LIBRAS para os surdos tendo como exemplo os filmes mudos, que utilizavam mímicas, gestos e expressões corporais e faciais para se comunicar e se expressar, qualquer pessoa conseguia entender todo o contexto apresentado sem precisar da fala. “Assim sendo, pode-se afirmar que a LIBRAS é, de fato, um idioma gestual-visual principalmente focado na memória fotográfica, na construção de imagens no cérebro e no relacionamento entre si desses elementos visuais”. (CARVALHO; CASTRO, 2009, p. 23).

2.3 - SURDEZ E TECNOLOGIAS

Segundo Silva, Kauchakje e Gesueli (2003, p. 196), o avanço tecnológico teve grande contribuição para os deficientes auditivos, principalmente na área da comunicação, com aparelhos de telefone, relógios, um dispositivo que possibilita aos surdos assistirem televisão com modo de legenda, etc. “Software de vários tipos existem hoje no mercado e/ou na internet: editores de texto, de apresentações, de jornais eletrônicos e de histórias em quadrinhos”. Esse acesso fácil e rápido nos possibilitou uma gama de modos diferentes de utilizar a comunicação, tanto à escrita quanto à leitura. “De forma análoga, a interface de um sistema computacional é o meio pelo qual pessoas e computadores se comunicam: é o elemento do ambiente que (inter) medeia o diálogo homem-computador e que, portanto, demanda um uso cognitivo da linguagem” (SILVA; KAUCHAKJE; GESUELI, 2003, p. 197). A interface auxiliará no modo de comunicação entre a pessoa e um artefato.

Há dois pontos muito importantes que devem ser seguidos no desenvolvimento desse programa. O primeiro é o desenvolvimento do software e o segundo passo é as particularidades humanas estarem atreladas ao seu uso. “Do ponto de vista do sistema, a questão tem sido tratada com grande ênfase em metodologias e técnicas da Engenharia de software para projeto e especificação da parte do código que implementa a interface” (SILVA; KAUCHAKJE; GESUELI, 2003, p. 197).

Já em relação às características humanas, os estudos têm focalizado os aspectos ergonômicos, culturais e cognitivos envolvidos no uso de elemento do ambiente computacional, como por exemplo, a eficiência do uso do mouse, a disposição das teclas de função no teclado, etc. (SILVA, KAUCHAKJE, GESUELI, 2003, p. 197 - 198).

As pesquisas na área da tecnologia em serviço dos diversos tipos de deficiências estão evoluindo de forma significativa. Esses estudos utilizam as tecnologias de informação e comunicação visando diminuir a exclusão social e melhorar a qualidade de vida dos deficientes em situações cotidianas em casa, no serviço e também auxiliando na aprendizagem. “Trata-se, portanto, de uma área de estudo e atuação de caráter multidisciplinar que demanda, por um lado, o delineamento cuidadoso do perfil da população-alvo e, por outro, soluções técnicas nem sempre simples de serem implementadas” (SILVA; KAUCHAKJE; GESUELI, 2003, p. 198). As questões de usabilidade também são de grande importância, buscando diminuir os esforços na utilização do software.

Estudos que também estão crescendo, na área das tecnologias de informação e comunicação voltada para os surdos, são os *SingWriting*, que usa a linguagem de sinais para ajudar a trabalhar com os softwares. “O *SingWriting*, desenvolvido por Valerie Sutton em 1975, representa graficamente as características visuoespaciais da Língua de Sinais, até então considerada ágrafa”. (SILVA; KAUCHAKJE; GESUELI, 2003, p. 205).

3 - ENGENHARIA DE SOFTWARE

Segundo Pressman (2006, p. 1), Os softwares são construídos por profissionais da área de informática que são responsáveis para manter o produto funcionando por longa data. Os softwares são de grande importância para a sociedade atual, esse universo tecnológico afeta nossas atividades rotineiras, as empresas, etc. Incluindo “programas que executam em computadores de qualquer tamanho e arquitetura, conteúdo que é apresentado ao programa a ser executado e documentos tanto em forma impressa quanto virtual que combinam todas as formas de mídia eletrônica”.

Atualmente, o software de computadores é muito importante no mundo da tecnologia, sua criação foi em 1950 e naquela época ninguém imaginava o avanço que essa nova descoberta traria para diversas áreas de estudos. Além de não conseguirmos imaginar o mundo sem os softwares, já que podemos fazer uma pesquisa pela internet, comprar produtos pelo computador sem ter o trabalho de sair de casa e até manter contato com amigos e conhecer pessoas novas. A internet possibilitou uma gama de variedades e conforto. O avanço tecnológico mudou muito a vida das pessoas, hoje é impossível imaginar como seria a vida sem esses pequenos detalhes. “Ninguém poderia ter previsto que o software estaria embutido em sistemas de toda a espécie: transporte, médico, telecomunicações, militar, industrial, entretenimento, máquinas de escritório – a lista é quase sem fim” (SANO, NETO, 2008, p. 11). Sem contar que o processo de manutenção desses softwares requerem tanto recursos e trabalho quanto para criar um software.

Do ponto de vista do engenheiro de software, o produto do trabalho são os programas, conteúdo (dados), e documentos que compõem um software de computador. Mas, do ponto de vista do usuário, o produto do trabalho é a informação resultante que, de algum modo, torna melhor o mundo do usuário (PRESSMAN, 2006, p. 1).

Hoje, o software adquiriu dois papéis na sociedade, como produto e de veículo que irá entregar este produto. “Como produto ele disponibiliza o potencial de computação presente no hardware do computador ou, mais amplamente, por uma rede de computadores acessível pelo hardware local” (PRESSMAN, 2006, p. 2). Podendo

operar desde em aparelhos de telefone celular até em computadores de tamanhos normais, “produzindo, gerindo, adquirindo, modificando, exibindo ou transmitindo informações que podem ser tão simples como um único bit ou tão complexas quanto uma apresentação multimídia”. Já operando como veículo de entrega do produto, este vai ser o responsável pela informação, podendo organizar as informações, colocá-las em um contexto mais adequado ou então, auxiliar no modo de se obter uma informação de diversos modos diferentes. Assim, “o software age como base para controle do computador (sistemas operacionais), para a comunicação da informação (redes) e para a criação e o controle de outros programas (ferramentas e ambientes de softwares)”.

De acordo com Pressman (2006), desde o lançamento até agora, o software passou por grandes mudanças, sendo melhorado e contribuindo para o avanço na parte interna de computadores, exemplo disso são maior memória e capacidade no armazenamento de dados. Todo esse processo originou computadores mais rápidos e modernos.

Hoje em dia uma enorme indústria de software tornou-se fator dominante nas economias do mundo industrializado. O programador solitário de antigamente foi substituído por uma equipe de especialistas em software, cada um se concentrando numa parte da tecnologia necessária para produzir uma aplicação complexa (PRESSMAN, 2006, p. 3).

Ainda para Pressman (2006), a qualidade do software depende muito do projeto. Um projeto bem elaborado com certeza desenvolverá um software sem problemas. Muitas pessoas acreditam saber o que realmente significa um software, mas na verdade estas têm apenas uma vaga ideia de toda complexidade do produto.

Softwares são (1) instruções (programas de computadores) que quando executadas fornecem as características, função e desempenho desejados; (2) estrutura de dados que permitem aos programas manipular adequadamente a informação; e (3) documentos que descrevem a operação e o uso dos programas (PRESSMAN, 2006, p. 4).

Para o autor acima (2006, p. 4), um software possui todo um processo de engenharia para ser feito, não é como outras coisas que o homem produz, este tem todo um trabalho envolvido no seu desenvolvimento. “O software é um elemento de um sistema lógico e

não de um sistema físico”. Atualmente existem sete tipos de softwares: software de sistemas, software de aplicação, software científico e de engenharia, software embutido, software para linha de produtos, aplicações da web e software para inteligência artificial. A categoria que será utilizada para o desenvolvimento do projeto será o software de aplicação, que irá resolver um problema específico através de programas isolados. “Aplicações nessa área processam dados comerciais ou técnicos de um modo que facilita as operações ou gestão/tomada de decisões técnicas do negócio”. (PRESSMAN, 2006, p. 6). Porém, com o passar do tempo, é necessário inserir esse software em um novo contexto, como uma manutenção para deixá-lo mais moderno, por isso, é importante estar atualizando um software que já é mais antigo e adaptando-o a sociedade moderna e deixando-o com uma qualidade ainda melhor.

O software tornou-se o elemento-chave na evolução de sistemas e produtos baseados em computador, e uma das tecnologias mais importantes em todo o mundo. Ao longo dos últimos 50 anos, o software evoluiu de um ferramental especializado em soluções de problemas e análise de informação para um produto de indústria (PRESSMAN, 2006, p. 13).

Ainda de acordo com o autor já citado (2006, p. 16), é muito importante elaborar um roteiro bem estruturado com passos para seguir, isso fará com que o profissional não se perca durante o processo de desenvolvimento e o resultado, provavelmente, será de alta qualidade. Esse roteiro é conhecido como processo de software, responsável por guiar o profissional e se antecipar das possíveis necessidades do projeto. “Além disso, o pessoal que solicitou o software tem um papel a desempenhar no processo de defini-lo, construí-lo e testá-lo”. Assim, o projeto terá maior estabilidade, controle de qualquer situação e, principalmente, organização. Esse processo dependerá muito do tipo de software que será desenvolvido. “Do ponto de vista de um engenheiro de software, os produtos do trabalho são os programas, documentos e dados produzidos em consequência das atividades e tarefas definidas pelo processo”. O que irá denominar se um projeto está em boas condições de uso é testando e vendo se está funcionando corretamente. “Todavia, a qualidade, pontualidade e viabilidade a longo prazo do produto que você constrói são os melhores indicadores da eficácia do processo usado”.

Para Pressman (2006), a engenharia de software, como qualquer área da engenharia, preza pela organização e a qualidade. Por isso, o desenvolvimento é feito através da

tecnologia de camadas, que deixará o processo mais estruturado e tem como base a qualidade.

Gestão de qualidade total, Seis Sigmas (*Six Sigma*) e filosofias análogas levam à cultura de um processo contínuo de aperfeiçoamento, e é essa cultura que, em última análise, leva ao desenvolvimento de abordagens cada vez mais efetivas para a engenharia de software (PRESSMAN, 2006, p. 17).

Portanto, isso mostra que para obter uma boa qualidade de um software, é preciso que o desenvolvimento siga as seis etapas e levem em consideração que tudo pode passar por modificações ao longo do tempo graças ao avanço tecnológico.

A base do processo de fabricação de um software de qualidade é desenvolvida através de camadas, ou seja, quando todas as camadas estiverem prontas, o resultado é um software completo e bem desenvolvido, com risco mínimo de erros ou falhas. “O alicerce da engenharia de software é a camada de processo. O processo de engenharia de software é o adesivo que mantém unidas as camadas de tecnologia e permite o desenvolvimento racional e oportuno de softwares de computador” (PRESSMAN, 2006, p. 17).

Os processos de software formam a base para o controle gerencial de projetos de softwares e estabelecem o contexto no qual os métodos técnicos são aplicados, os produtos de trabalho (modelos, documentos, dados, relatórios, formulários, etc.) são produzidos, os marcos são estabelecidos, a qualidade assegurada e as modificações são adequadamente geridas (PRESSMAN, 2006, p. 17-18).

Ou seja, o controle gerencial do projeto de um software deverá trazer um relatório contendo informações de todos os processos do desenvolvimento do produto. Esse material irá mostrar a segurança e qualidade do produto, deixando o cliente mais seguro para utilizá-lo. Outro ponto importante são as informações técnicas que poderão auxiliar outros profissionais para as modificações e testes deste software.

Segundo Pressman (2006, p. 18), os métodos irão mostrar a técnica de como construir o software, auxiliando o profissional na hora do desenvolvimento do software.

“Eles abrangem um amplo conjunto de tarefas que incluem comunicação, análise de requisitos, modelagem de projeto, construção de programas, testes e manutenção”.

“Um arcabouço de processo estabelece o alicerce para um processo de software completo pela identificação de um pequeno número de atividades de arcabouço aplicáveis a todos os projetos de software, independentemente de seu tamanho ou complexidade” (PRESSMAN, 2006, p. 18).

Ou seja, o arcabouço é um conjunto de ações de engenharia de software que irá facilitar o desenvolvimento do software desde a parte do planejamento do produto até a implantação do mesmo. “Além disso, o arcabouço de processo engloba um conjunto de atividades guarda-chuva que são aplicáveis durante todo o processo de software”. (PRESSMAN, 2006, p. 18).

Ainda segundo o autor já citado (2006, p. 19), no processo de software cada ação de arcabouço é desempenhada por um composto de ações de engenharia de software, que irá produzir um significativo produto resultante deste trabalho de engenharia. Logo abaixo segue um exemplo de um arcabouço que pode ser aplicado em grande parte dos trabalhos de software:

“**Comunicação:** Essa atividade de arcabouço envolve alta comunicação e colaboração com o cliente (e outros interessados) e abrange o levantamento de requisitos e outras atividades relacionadas”. Sendo assim, pode-se notar a importância da comunicação durante o desenvolvimento do projeto. A cliente também possui clareza de tudo o que acontece e colabora com informações e ideias para que o produto final atenda às suas expectativas. Isto mostra a complexidade do trabalho e toda a responsabilidade que o profissional precisa ter na hora de fazer um projeto.

Já a etapa de **Planejamento** irá construir um plano de trabalho de engenharia de software que o profissional deverá seguir durante o desenvolvimento do projeto. “Descreve as tarefas técnicas a ser conduzidas, os riscos prováveis, os recursos que serão necessários, os produtos de trabalho a ser produzidos e um cronograma de trabalho”.

Para a ação de **Modelagem** implica a geração de modelos que irão permitir maior entendimento dos requisitos do software e do projeto em si, tanto para o profissional que está trabalhando no projeto quanto para o cliente.

A **Construção** é uma ação de combinação entre a elaboração de código, seja este manual ou automático, e as verificações necessárias para mostrar os erros destes códigos gerados.

Na **Implantação** do software, o produto será conferido pelo cliente, que será o responsável pela avaliação, levando em consideração se atende todas as necessidades e o cumprimento do que foi pedido inicialmente, e assim, irá enviar um *feedback* para o profissional. A Figura 2 mostra a relação dessas atividades com as fases do processo unificado.

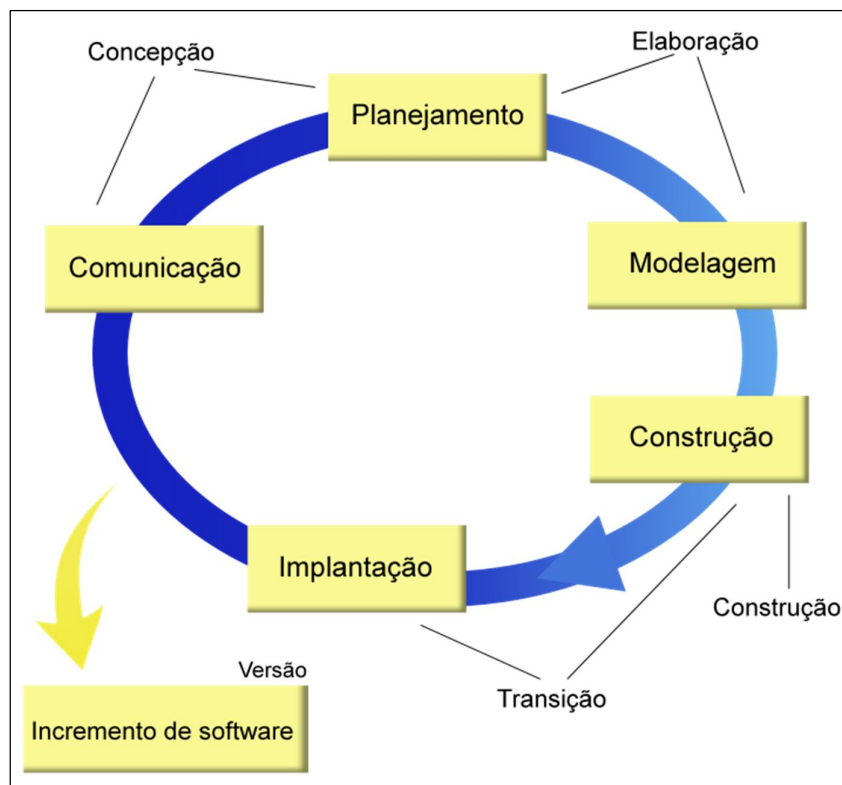


Figura 2 - Fases do processo unificado
 Fonte: Adaptada de PRESSMAN (2006) Acesso em: 22/04/2013.
 URL: <http://www.ebah.com.br/content/ABAAAAsdkAG/monografia>

Essas cinco atividades genéricas de arcabouço podem ser usadas durante o desenvolvimento de pequenos programas, durante a criação de grandes aplicações para a internet e para a

engenharia de grandes e complexos sistemas baseados em computador. Os detalhes do processo de software serão muito diferentes em cada caso, mas as atividades de arcabouço permanecem as mesmas (Pressman, 2006, p. 19).

4 - TECNOLOGIAS MICROSOFT

4.1 - MICROSOFT KINECT ®

Segundo Cruz, Lucio e Velho (2012) apud Correria (2013), A apresentação do software foi em 1 de junho de 2009, ainda com seu nome original “Projeto Natal”. Em novembro de 2010 teve seu lançamento como acessório do Xbox 360, mostrando o resultado do trabalho em conjunto da PrimeSense e a Microsoft.

De acordo com Correria (2013), o Microsoft Kinect ® é um dispositivo que é capaz de captar movimentos e sons através de uma câmera RGB, um acelerômetro, um sensor de profundidade, um motor e quatro microfones. O sensor de profundidade é composto por um emissor e uma câmera infravermelhos que são capazes de captar os movimentos que são emitidos por quem está à frente do dispositivo.

O emissor de infravermelhos cria um padrão estruturado de luz infravermelha e a câmara lê a reflexão desses raios. Por sua vez, a câmara interpreta a deformação da projeção e converte essa informação em valores de profundidade, medindo a distância entre o objeto e o sensor (CORREIA, 2013, p. 21).

De acordo com Correria (2013), a profundidade é medida através de uma escala de tons de cinza, por isso, a qualidade do pixel é melhor depende da boa iluminação dos ambientes onde o dispositivo está instalado. Porém, foi criada a versão *Near Mode* feita para Windows, que permite o melhor reconhecimento e com maior precisão em apenas 40cm.

Segundo o site oficial da Microsoft, “O sistema de microfones é composto por uma série de quatro destes aparelhos. Com os microfones o Kinect é capaz de gravar áudio, determinar a localização da fonte sonora e a direção da onda de áudio”.

Para Correria (2013, p. 23), ao se posicionar em frente ao dispositivo, o mesmo faz um reconhecimento do esqueleto a partir das articulações do corpo humano. “Rastreamento do esqueleto é um processo que representa o corpo humano por um número de

articulações representativas de partes do corpo, como a cabeça, o pescoço, os ombros e os braços. Cada articulação é representada pela sua localização no espaço 3D”.

Segundo Correria (2013, p. 24), as expressões faciais são facilmente detectadas pelo dispositivo, também como as posições da cabeça, auxiliando em muitas outras áreas de estudo. “A detecção da expressão facial e da posição da cabeça têm sido uma área de investigação ativa na área de visão por computador. A sua realização tem impacto em diversas áreas como interação homem-computador e reconhecimento facial”.

O dispositivo também possui alta tecnologia, capaz de capturar imagens em três dimensões, possibilitando uma melhor imersão e maior proximidade do real. “Para o efeito usam-se digitalizadores capazes de captar imagens 3D de alta qualidade usando sistemas de lasers de luz estruturada, produzindo bons resultados, embora a um custo elevado ou sendo necessário muito tempo para produzir uma digitalização”. A Figura 3 mostra a componentes do dispositivo Microsoft Kinect®.

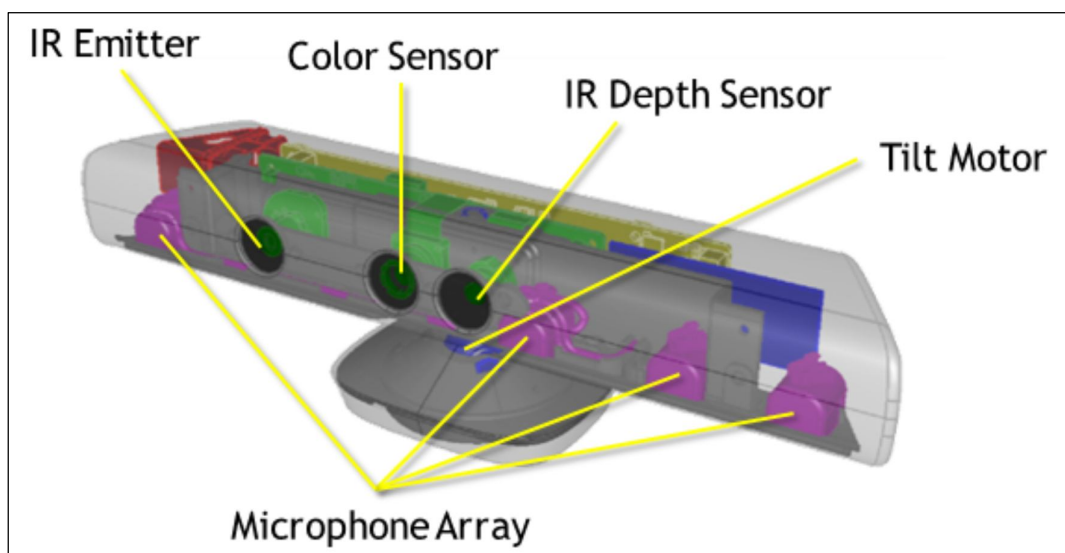


Figura 3 - Componentes do sensor Kinect.
Microsoft. MSDN - Kinect for Windows Sensor, 2012. Último acesso em 22/04/2013. URL:
<http://msdn.microsoft.com/en-us/library/hh855355.aspx>.

Assim que o dispositivo obtém as informações de profundidade, os valores podem não estar precisos. “O erro da medição de profundidade aumenta com o quadrado da distância. Assim, para formular a distância entre o modelo da face e o mapa de

profundidade, cada ponto do mapa tem a sua própria matriz de covariância adequada para modelar a sua incerteza” (CORREIA, 2013, p. 24).

No entanto, as expressões faciais usadas na Língua Gestual Portuguesa tomam variadas formas, algumas das quais testes com o sensor mostraram não serem identificadas nativamente. É exemplo disso a expressão “bochechas cheias”. Para superar este problema presume-se conseguir refinar a malha de detecção da expressão facial, tornando-a mais sensível na zona das bochechas do indivíduo. Com este processo espera-se tornar o sistema mais sensível a expressões mais comedidas, evitando a necessidade de exagerar a expressão (CORREIA, 2013, p. 28).

Com a ajuda dessa tecnologia será possível desenvolver o software que tornará mais acessível os computadores para as pessoas surdas. Segundo Prada (2010, p. 1), a pessoa irá controlar o que acontece nos jogos através da própria voz e dos movimentos. “Além de um imenso passo para a interatividade com o universo virtual, o acessório representa uma verdadeira revolução para os vídeo games”.

Com o Kinect é possível interagir com o computador através da voz e dos movimentos corporais. Ele ainda é capaz de reconhecer a aproximação destas pessoas. Com esse software, pessoas surdas poderão ter acesso a computadores através dos gestos programados para serem reconhecidos pelo Kinect. “Em Junho de 2011 a Microsoft lançou um Software Development Kit (SDK) para usar o sensor Kinect com o sistema operativo Windows 7, sendo que em Fevereiro de 2012 a versão para Windows do Kinect, Kinect for Windows, foi lançada” (CORREIA, 2013, p. 19).

De acordo com Correria (2013), o Kinect teve grande aceitação, tanto por seu preço baixo quanto pela capacidade tecnológica que permite a interação. Sem contar o sucesso entre profissionais das diversas áreas tecnológicas que se sentiram tentados em testar as possibilidades que esse software permite.

4.2 - LINGUAGEM DE PROGRAMAÇÃO C SHARP (C#)

C# (ou C Sharp) é uma linguagem de programação orientada a objetos desenvolvida pela Microsoft como parte da plataforma .NET. Sua sintaxe foi baseada em C++ e também herda algumas características do Delphi e Java (BRITO, 2009, p. 22)

Durante o desenvolvimento da plataforma .NET, as bibliotecas foram escritas originalmente numa linguagem chamada *Simple Managed C* (SMC), que tinha um compilador próprio. Já, em Janeiro de 1999, foi formada uma equipe de desenvolvimento por Anders Hejlsberg, que fora escolhido pela *Microsoft* para desenvolver a linguagem. Inicia-se então à criação da linguagem chamada *Cool*. Mais tarde, em 2000, o projeto .NET era apresentado ao público na *Professional Developers Conference* (PDC), e a linguagem *Cool* fora renomeada e apresentada como C# (MSDN, 2008).

A linguagem, embora tenha sido desenvolvida por vários programadores, é atribuída principalmente a Anders. Ele foi o arquiteto de alguns compiladores da *Borland*, e entre suas criações mais conhecidas estão o *Turbo Pascal* e o *Delphi* (MSDN, 2008).

Baseada nas linguagens C++ e Java, é considerada a linguagem símbolo do .NET, por ter sido criada praticamente do zero para funcionar na nova plataforma, sem preocupações de compatibilidade com código existente. O compilador C# foi o primeiro a ser desenvolvido, e a maior parte das classes da plataforma foi desenvolvida nesta linguagem (MSDN, 2008).

O C# é constituído por características diversas. Por exemplo, a linguagem suporta ponteiros utilizando-se da palavra reservada *unsafe* (código não-seguro), que é obrigatório. Seu uso não é aconselhável, e blocos de códigos que o usam geralmente requisitam permissões mais altas de segurança para poderem ser executados (MSDN, 2008).

4.3 MICROSOFT XNA

Desenvolvida pela Microsoft, a nova plataforma de programação gráfica chamada XNA Game Studio foi apresentada em novembro de 2006. Tem como objetivo interligar as APIs do DirectX e o programador, contendo múltiplas ferramentas, efeitos e funcionalidade previamente instaladas, contribuindo no uso de efeitos e geometria espacial (BIRCK, 2007, p. 3).

Uma das vantagens é a capacidade de carregar modelos, sons e texturas em apenas uma linha, sem precisar se preocupar com o formato dos arquivos, ao contrário de se trabalhar diretamente com o DirectX. Para a criação de um pequeno jogo, vantagens como essas facilitam bastante a vida de quem é iniciante no ramo de desenvolvimento de jogos, pois não estão preocupados em deixar o jogo otimizado e sim funcional (BIRCK, 2007, p. 3).

A linguagem padrão escolhida pela Microsoft foi o C# (C Sharp), mas como o processo utiliza o .NET Framework, qualquer linguagem .NET é capaz de rodar o XNA, bastando apenas alterar a sintaxe (BIRCK, 2007, p. 3).

4.4 BANCO DE DADOS SQL SERVER

Criado pela Microsoft, o MS SQL Server é um SGBD – Sistema Gerenciador de Banco de Dados Relacional (BRITO, 2009, p. 22).

Com a versão o MS SQL Server 2008 é fornecida uma plataforma de dados confiável, produtiva e inteligente que permite executar suas aplicações mais exigentes, reduza o tempo e o custo com o desenvolvimento e o gerenciamento de aplicações e entregue percepção que se traduz em ações estratégicas em toda sua organização. O SQL é um Banco de dados robusto e usado por sistemas corporativos dos mais diversos portes. O SQL permite a encriptação de um banco de dados inteiro, arquivos de dados ou arquivos de log, com necessidade de alterações nas aplicações. Os benefícios dessa encriptação incluem: consultas de dados encriptados, usando consultas em série ou

associadas, proteger os dados de consultas de usuários não autorizados e encriptação de dados sem requerer qualquer mudança nas aplicações existentes (MICROSOFT, 2008).

5 – SEGMENTAÇÃO

5.1 – CONSIDERAÇÕES INICIAIS

A segmentação tem como objetivo subdividir a imagem em regiões para facilitar a interpretação da informação contida na mesma (GONZALEZ, 2007). Esta subdivisão é feita a partir do agrupamento de pixels utilizando características (*features*) comuns a este conjunto que podem ser cor, textura, intensidade ou continuidade. São muitas as técnicas de segmentação existentes, como por exemplo: Segmentação através de algoritmos de clusterização como o K-Means (LLOYD, 1982); Métodos baseados em Histogramas (SHAPIRO et al., 2010); Métodos baseados em compressão de imagens (MOBAHI et al., 2010); *Edge Detection* (CANNY, 1986); *Region Growing* (SHAPIRO et al., 2010); Segmentação através de Grafos (GRADY, 2006); *Thresholding* (GONZALEZ et al., 2007); Segmentação utilizando *Partial Differential Equations* (OSHER et al., 2003); *Watershed transformation* (BEUCHER et al., 1993); Segmentação em múltiplas escalas (LINDBERG, 1993); Segmentação através de modelos pré-estabelecidos (MESTER et al., 1988); Segmentação de regiões através da utilização de redes neurais como *Multi-Layer Perceptron* (MLP) (HAYKIN, 1998), Kohonen map (KOHONEN, 2001), *Pulse-Coupled Neural Networks* (PCNN) (JOHNSON et al., 1999), entre outras.

Considerando as características previamente levantadas com relação aos algoritmos de segmentação bem como a adequação aos objetivos desta pesquisa, optou-se por restringir os estudos aos algoritmos por diferenças (*Background Subtraction*). A seguir será apresentada a conclusão sobre os algoritmos estudados e a solução proposta para este projeto.

5.2 SUBTRAÇÃO DE FUNDO (BACKGROUND SUBTRACTION)

Normalmente utilizado para detecção de movimento através de diferenças, este método consiste na dicotomização da imagem em pixels pertencentes ao fundo e pixels

referentes aos objetos de interesse ou em movimento na cena. Esta tarefa é efetuada através da modelagem de fundo que consiste em uma modelagem estatística realizada por meio de observações de uma sequência de imagens de uma cena para tentar estimar quais regiões da imagem pertencem ao fundo. Para obter os pixels de interesse, que compõe os objetos em movimento, é feita uma simples subtração do modelo de fundo da imagem atual, seguido de um *threshold* (ANJO, 2012).

De acordo com Piccardi (2004) são vários os métodos de *Background Subtraction*, onde as principais variações estão na forma de estimação do modelo de fundo e no método de comparação deste modelo com a imagem atual. Alguns métodos citados por Piccardi, como *Kernel Density Estimation* (KDE), *Sequential KD Approximation*, *Eigen Backgrounds* entre outros, não possibilitam a implementação em tempo real.

Considerando o estudo teórico relatado em (ANJO, 2012), neste trabalho foram focados os métodos passíveis de utilização em tempo real, sendo estes: Subtração simples; *Running Gaussian Average*; *Temporal Median Filter*; e *Mixture of Gaussians*. Estes métodos serão descritos brevemente nas próximas seções. Posteriormente na seção 8.2.5 será apresentado um estudo sobre os principais problemas dos métodos de *Background Subtraction* e, por último, na seção 8.2.6 serão apresentadas técnicas de Pós-Processamento para obter melhores resultados após a aplicação dos algoritmos de *Background Subtraction*.

5.2.1 Subtração Simples

Recurso mais simples que consiste na subtração pixel a pixel ($l_{x,y}$) a imagem anterior ($Frame_{i-1}$) da imagem atual ($Frame_i$). Posteriormente, é aplicado um processo de uma binarização utilizando um limiar previamente estabelecido, apontado na fórmula exibida na Figura 4 e ilustrado na Figura 5 (ANJO, 2012).

$$l_{x,y} = \begin{cases} 1 & \text{se } |Frame_i(x,y) - Frame_{i-1}(x,y)| > \sigma \\ 0 & \text{se } |Frame_i(x,y) - Frame_{i-1}(x,y)| < \sigma \end{cases}$$

Figura 4 - Fórmula de subtração pixel a pixel.
Fonte: (ANJO, 2012).

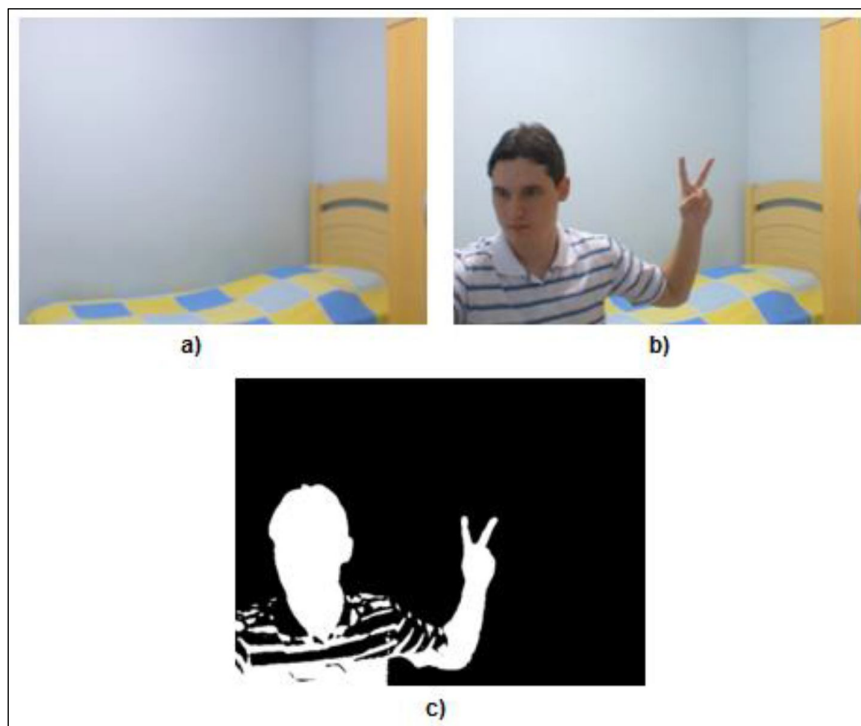


Figura 5 - Exemplo de resultado da subtração simples, sendo a) o modelo de fundo, b) a imagem atual e c) o resultado do algoritmo.
Fonte: (ANJO, 2012).

5.2.2 *Running Gaussian Average*

Consiste em tomar a decisão entre pixels que sofreram alteração em relação ao modelo de fundo e os que são equivalentes ao fundo, não pode simplesmente se basear na imagem anterior para a maioria das aplicações. Neste contexto, Christopher Wren (1997) propôs um método no qual à medida que novas imagens são observadas, o modelo estatístico do fundo é modificado para se adaptar às pequenas alterações detectadas.

Mas esta alteração do modelo de fundo não pode permitir que as modificações sejam absorvidas pelo modelo rapidamente, ou seja, as alterações devem ser mínimas para adaptar o modelo lentamente, ao longo do tempo, a pequenas variações no fundo da cena.

Assim, o modelo de fundo atual é calculado através da soma ponderada entre a imagem recentemente observada e o modelo de fundo anterior.

Com o modelo de fundo, uma subtração simples é efetuada e seguida de uma binarização como no método anterior.

A modificação do modelo de fundo também pode ser efetuada de uma forma seletiva, ou melhor, caso o pixel da imagem atual corresponda a um pixel de uma região em movimento (diferença detectada), ele não será utilizado para atualizar o modelo de fundo.

5.2.3 Temporal Median Filter

Outra maneira de se modelar o fundo do cenário, considerada mais estável (por gerar menos ruído como pixels de regiões em movimento) por alguns pesquisadores, é armazenar um buffer de frames para depois calcular a média e obter um modelo de fundo. A maior desvantagem deste método é o armazenamento em memória deste buffer de imagens, e também a definição do tamanho do mesmo que determinará quanto tempo será necessário para que um objeto estático passe a fazer parte do fundo (ANJO, 2012).

5.2.4 Mistura de Gaussianas (*mixture of gaussians*)

Todos os métodos descritos anteriormente definem para cada pixel um único valor no modelo de fundo. Em alguns casos, pode-se ter mudanças no fundo que não devem ser detectadas como diferenças, mas que acontecem em um intervalo de tempo menor que o da atualização do modelo.

Um exemplo desta situação seria uma árvore que cobre parcialmente uma casa e se movimenta com o vento. Um mesmo pixel pode assumir o valor da casa, de um galho ou de uma folha.

Para solucionar este problema Stauffer e Grimson (1999) modelaram cada pixel da imagem com um conjunto de Gaussianas, que seriam responsáveis pela modelagem de cada uma das situações do fundo que não devem ser detectadas como diferenças.

De acordo com Stauffer e Grimson (1999), a probabilidade de observar um certo valor x de um pixel no tempo t utilizando um conjunto (K) de gaussianas multivariadas é dada pela fórmula presente na Figura 6.

$$P(x_t) = \sum_{i=1}^k w_{i,t} \mathcal{N}(x_t - \mu_{i,t}, \sum_{i=1}^k w_{i,t}) \text{ sendo } \sum_{i=1}^k w_{i,t} = 1$$

Figura 6 - Probabilidade de observar um certo valor x de um pixel no tempo t utilizando um conjunto (K) de gaussianas multivariadas.
Fonte: (ANJO, 2012).

Mas para que esta fórmula possa modelar o fundo, um critério deve ser estabelecido para diferenciar entre as distribuições gaussianas referentes ao fundo e as distribuições gaussianas das diferenças. Para isto, as distribuições são ordenadas segundo relação entre a amplitude e o desvio padrão, e as primeiras são definidas como fundo. Por exemplo, se cada pixel for modelado com três gaussianas, pode-se definir que as duas primeiras colocadas serão definidas como fundo e a restante será correspondente as diferenças.

Normalmente o número de gaussianas utilizadas (K) varia entre três e cinco. Quanto maior o número de gaussianas maior é o tempo de processamento, dificultando a sua utilização em aplicações de tempo real.

A cada novo frame observado, os parâmetros das gaussianas que modelaram cada um dos pixels identificados devem ser atualizados. Inicialmente o sistema está em fase de adaptação, modelando o ambiente. Após alguns frames ele já é capaz de classificar os pixels.

5.2.5 Principais Problemas da Detecção de Movimento por Background Subtraction

Apenas modelar o fundo e aplicar subtração para detectar as diferenças não é o bastante para garantir um bom resultado na segmentação. Essas diferenças detectadas podem ser *highlights*, *ghosts* ou sombras, que não deveriam fazer parte do resultado

final. Estes problemas, em companhia com possíveis soluções, serão apresentados nesta seção.

Outro assunto importante que será abordado são regras de decisão para atualizar o modelo de fundo e câmeras com configurações automáticas que definem como o sistema irá se comportar nas atualizações do modelo que envolvem pixels detectados como diferença.

5.2.5.1 Alterações de Brilho (highlights)

Este tipo de variação acontece com mudanças repentinas na iluminação do cenário que está sendo observada. Esta variação não é facilmente detectada e modelada no espaço de cor RGB (*Red, Green e Blue*), por exemplo.

Por isso, Duque et al. (2005) propuseram uma solução utilizando o espaço de cor HSV (*Hue, Saturation e Value*) para modelar as variações de cromacidade dos objetos que seriam classificadas como *Highlights*.

Este método tem como entrada a imagem atual e a imagem do modelo de fundo no tempo, e retorna como resultado uma máscara binarizada obtida através da seguinte fórmula apresentada na Figura 7:

$$HM_n(x, y) = \begin{cases} 1, & \text{se } \frac{1}{\beta} \leq \frac{I_n^V(x, y)}{B_n^V(x, y)} \leq \frac{1}{a} \wedge \\ & |I_n^S(x, y) - B_n^S(x, y)| < t_s \wedge \\ & |I_n^H(x, y) - B_n^H(x, y)| \leq t_h \\ 0, & \text{caso contrário} \end{cases}$$

Figura 7 - Fórmula para máscara binarizada.
Fonte: (ANJO, 2012).

A regra depende da fonte de iluminação e das propriedades de reflectância e irradiação dos objetos na cena. Fontes de iluminação intensas e objetos com altas taxas de irradiação e reflectância implicam em baixos valores de a .

Reflectância é a proporção entre o fluxo de raios luminosos que ocorre sobre o objeto e o fluxo de raios que é refletido. Já a irradiação corresponde à emissão de ondas

do espectro visível ou não por objetos, podendo ser fontes luminosas ou fontes de calor, por exemplo, porém este tipo de característica só irá influenciar caso a câmera sendo utilizada seja sensível a este espectro de ondas.

O parâmetro β é utilizado para evitar a identificação de ruídos como *highlights*. Já os limiares são a máxima variação permitida para os componentes Saturação e Hue do espaço de cor.

5.2.5.2 Sombras

O aparecimento de sombras dos objetos, que são projetadas pela fonte de luz na cena sendo observada, pode alterar a forma do objeto, resultando em uma falha na segmentação. Para conter este problema Duque et al. (2005) também propuseram uma solução similar a detecção de *Highlights* com a seguinte fórmula apresentada na Figura 8:

$$SM_n(x, y) = \begin{cases} 1, & \text{se } \alpha \leq \frac{I_n^V(x, y)}{B_n^V(x, y)} \leq \beta \wedge \\ & |I_n^S(x, y) - B_n^S(x, y)| < t_s \wedge \\ & |I_n^H(x, y) - B_n^H(x, y)| \leq t_h \\ 0, & \text{caso contrário} \end{cases}$$

Figura 8 - Fórmula máscara binarizada contendo os pixels classificados como sombra.
Fonte: (ANJO, 2012).

Esta fórmula retorna uma máscara binarizada contendo os pixels classificados como sombra.

5.2.5.3 Fantasmas (ghosts)

Os *ghosts* são regiões (*blobs*) erradamente classificadas como diferenças que foram geradas pela movimentação de um objeto que fazia parte do modelo de fundo. Este tipo de erro de classificação impede o acerto do modelo de fundo naquela região, em caso de atualização seletiva do modelo, gerando um *deadlock* (CUCCHIARA et al., 2003).

Rita (CUCCHIARA et al.,2003) propôs uma solução de escopo global que identifica *blobs* detectados como diferenças e classifica-os como objetos em movimento, sombra de objetos em movimento, *ghosts* e sombra de *ghosts*. Para diferenciar um blob de um objeto em movimento e um *ghost*, é utilizado o método proposto por (BAINBRIDGE-SMITH, 1997) para o cálculo aproximado do Fluxo Ótico (*optical flow*) através de equações diferenciais. Este fluxo ótico quantifica a tendência de movimento de um *blob* ao longo do tempo. Então *ghosts* tendem a ficar imóveis diferenciando, assim, *ghosts* de objetos em movimento.

5.2.5.4 Câmeras de Configuração Automática

Os algoritmos mencionados até agora para *Background Subtraction* se baseiam nos valores dos componentes do espaço de cor da imagem para tomar uma decisão, e é a câmera que discretiza esses valores contidos nos pixels da imagem (ANJO, 2012).

As webcams que vêm embutidas em notebooks, ou mesmo as externas via conexão USB, são normalmente soluções de baixo custo que tentam se adaptar às alterações do ambiente (iluminação e cor) para discretizar imagens digitais. Esta adaptação é danosa para os algoritmos mencionados anteriormente, pois geram alterações repentinas que podem ser classificadas como diferenças (ANJO, 2012).

As principais configurações que se habitam automaticamente nas câmeras são *White Balance*, *Exposure Time* e *Automatic Gain Control* que serão definidas aqui segundo Mchugh (ANJO, 2012).

White Balance é o procedimento de remoção de predominância de cor irreal em uma imagem digitalizada, para que objetos da cor branca sejam digitalizados na cor branca. O algoritmo de *White Balance* da câmera deve levar em consideração a temperatura da fonte de luz que ilumina a cena para adquirir um bom resultado. Um exemplo de correção é apresentado na Figura 9 (MCHUGH, 2013).



Figura 9 - Exemplo de correção de White Balance.
Fonte: (ANJO, 2012).

Exposure Time, também conhecido como *Shutter Speed*, define o intervalo de tempo que o sensor da câmera ficará exposto à luz para digitalizar uma imagem. Isto permite que a câmera se habitue a diferentes condições de iluminação, buscando manter um bom contraste na imagem (MCHUGH, 2013).

Automatic Gain Control é utilizado para favorecer a visualização da imagem quando o *Exposure Time* não conseguiu uma imagem nítida. É um algoritmo de multiplicação dos valores da imagem, na esforço de aumentar o contraste (MCHUGH, 2013).

5.2.5.5 Regra de Decisão para Atualizar Modelo de Fundo

Após adquirir como resultado as diferenças entre o frame atual e o modelo de fundo, todos os algoritmos de *Background Subtraction* mencionados anteriormente precisam atualizar de alguma forma o modelo.

Na atualização existem duas alternativas:

- **Atualização direta:** Atualizar o modelo de fundo sem utilizar a informação contida no resultado do *Background Subtraction*. Desta forma, todo o frame corrente é utilizado na atualização.

- **Atualização condicional:** Atualizar o modelo de fundo utilizando informações contidas no resultado do *Background Subtraction*. Os blobs desta imagem podem designar objetos em movimento que não devem intervir no modelo de fundo.

De acordo com Parks et al. (2008) a atualização condicional aumenta as chances de detecção de diferenças com precisão devido ao fato de objetos em movimento não poluírem o modelo de fundo. Ela pode também solucionar problemas como *Ghosts*, validando primeiro os *blobs* antes da atualização. Porém um *blob* mal classificado pode se tornar um problema, pois continuará sendo identificado como diferença e nunca será incorporado ao modelo de fundo.

Assim, no caso de aplicação de uma metodologia de atualização condicional, deve-se ter a cautela de garantir a classificação correta dos *blobs*, ou adicionar um algoritmo de verificação para eliminar *blobs* que permaneçam no resultado da segmentação por um período de tempo, evitando dessa forma, *deadlocks*.

5.2.6 Técnicas de Pós-Processamento

Parks et al. (2008) constatou em seu estudo dos principais algoritmos de *Background Subtraction*, algumas técnicas de pós-processamento que costumam ser desfrutadas para tentar garantir a morfologia dos objetos segmentados e eliminar ruídos indesejáveis. Algumas destas técnicas serão relatadas a seguir.

5.2.6.1 Remoção de Ruídos

Ruídos são normalmente gerados pela câmera e devem ser eliminados o quanto antes para não intervir em etapas decorrentes. Os mesmos podem ser removidos através da convolução de uma máscara de média na imagem, seguida de um *threshold* simples.

5.2.6.2 Identificação de Blobs

Como o resultado dos algoritmos de *Background Subtraction* normalmente é uma máscara binária, pode-se adaptar algoritmos de identificação de *blobs*. Estes algoritmos têm por finalidade a identificação de regiões conexas na imagem, rotulando as mesmas com um identificador único.

Este processo permite calcular características de cada um desses *blobs* e desfrutar essas informações para pretender modelar e classificar o que está sendo observado pela câmera.

Chang et al. (2004) apresentou um método, que trabalha em tempo linear, para rotular *blobs* em uma imagem binária. Percorrendo somente uma vez a imagem, o algoritmo é capaz de rotular os *blobs* utilizando seus contornos internos e externos. Este algoritmo pode ser ajustado para também rotular os “buracos” contidos nos *blobs* através dos contornos internos.

5.2.6.3 Area Thresholding

Area Thresholding baseia-se na remoção de *blobs* que possuam área inferior a β . Pode ser conceituado também um algoritmo de remoção de ruído, mas o tempo de execução será maior. Isto ocorre pelo fato da necessidade de identificação (rotulação) de *blobs* seguida de *Area Thresholding* que são processos que têm tempo de processamento variável dependendo da quantidade de *blobs* na imagem.

De acordo com (PARKS et al., 2008) para garantir um bom desempenho β deve ser bem abaixo da área do menor objeto de interesse, por exemplo, 25% da mesma.

5.2.6.4 Morphological Closing

Antes de definir a operação *Morphological Closing*, é necessária a exibição de três conceitos comuns em processamento de imagens, sendo eles: *Structuring Element*; o operador Dilatação (\oplus); o operador Erosão (\ominus) (ANJO, 2012).

O *structuring element* é semelhante a uma máscara de convolução, que é utilizado como elemento principal das operações morfológicas em processamento de imagens, e é convolucionado ou deslocado pela imagem para efetivar a operação desejada (ANJO, 2012).

O operador de Dilatação (\oplus) baseia-se basicamente na convolução de um *structuring element* em uma imagem binária marcando todos os pontos por onde passa

desde que ao menos um ponto da máscara seja adicionado a um pixel previamente marcado na imagem (ANJO, 2012).

Já o operador de Erosão (θ) baseia-se basicamente também na convolução de um *structuring element* em uma imagem binária, mas só irá marcar os pontos em que a máscara esteja drasticamente contida em uma região de pixels previamente marcada (ANJO, 2012).

Por fim, a operação *Morphological Closing*, ou operação de fechamento, é indicada pelo operador \cdot , onde A é a imagem binária e B é o chamado *structuring element* ou kernel que estabelecerá a forma de interação do algoritmo com a imagem. Esta operação é desenvolvida como indicado na Figura 10 a seguir:

$$A \cdot B = (A \oplus B) \theta B$$

Figura 10 - Operação de fechamento.
Fonte: (ANJO, 2012).

Onde \oplus é o operador de dilatação e θ é o operador de erosão. Assim, a operação de fechamento é a erosão da dilatação da imagem A por um kernel B (ANJO, 2012).

Parks et al. (2008) relataram em seu estudo comparativo, que todos os algoritmos de *Background Subtraction* melhoraram seus desempenhos quando a operação de *Morphological closing* é utilizada (ANJO, 2012).

5.2.6.5 Saliency Test

De acordo com (PARKS et al., 2008) o *Saliency test* ou *Teste de “saliência”* é desfrutado para remover *blobs* em que dos seus pixels não são salientes o suficiente. Um pixel é saliente se ele continuar sendo classificado como desigual após o limiar para *threshold* ser multiplicado por um fator a . Geralmente o fator a é igual a 2.

Assim, este teste busca garantir que o *blob* seja suficientemente desigual do modelo de fundo, tentando evadir falsos positivos na classificação de *blobs* de objetos que estão em movimento.

5.2.6.6 Optical Flow Test

Este teste é desfrutado para solucionar o problema de *ghosts* mencionado anteriormente. Baseia-se no cálculo da média da tendência de movimento dos pixels do *blob*. Caso a média se aproxime de zero este *blob* é um *ghost* e deve ser apagado.

6 – METODOLOGIA

Considerando que o objetivo principal deste trabalho era desenvolver uma aplicação responsável pelo módulo de segmentação do sistema que lide com subconjunto da Língua Brasileira de Sinais (LIBRAS), utilizando o Kinect e, tendo em vista a noção da complexidade do problema, pretendeu-se desenvolver o projeto de uma forma incremental. Desta forma dividiu-se o projeto em diferentes fases, definidas como:

1. Fase de investigação teórica
2. Estudo do Microsoft Kinect e das tecnologias vinculadas ao dispositivo;
3. Implementação do algoritmo de segmentação das imagens capturadas;
4. Implementação do software (módulo de segmentação);

Na primeira fase foi feito um estudo teórico sobre técnicas de desenvolvimento de software, que auxiliaram na criação do software e também foi feito um estudo sobre a Linguagem de Sinais (LIBRAS), com o objetivo de conhecer os gestos específicos referentes aos dígitos 0 (figura 11), 1 (Figura 12), 2 (Figura 13), 3 (Figura 14) e o 4 (Figura 15).

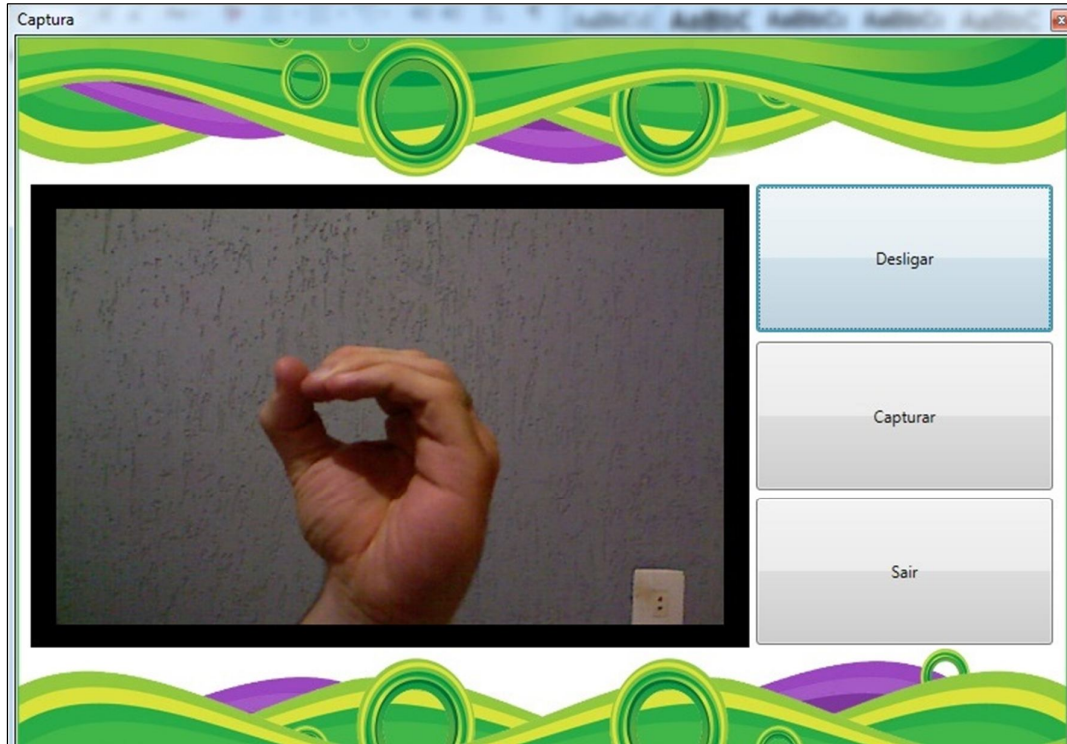


Figura 11 - Gesto que representa o número 0.
Fonte: Elaborada pelo autor.

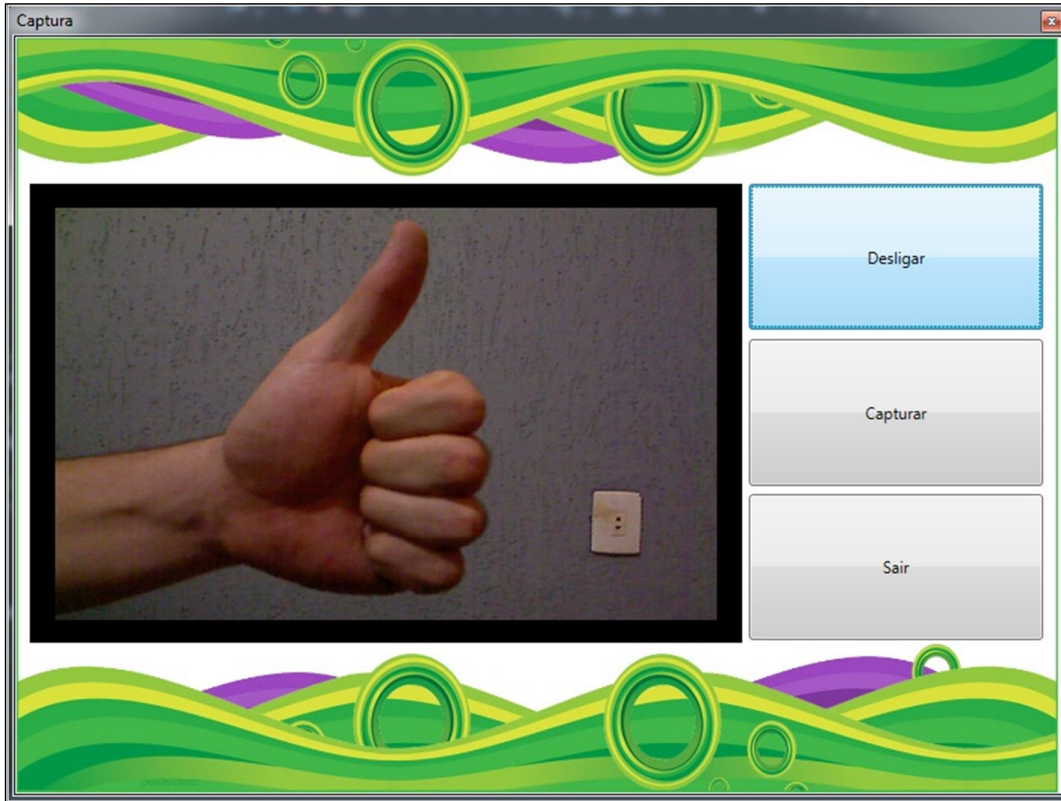


Figura 12 - Gesto que representa o número 1.
Fonte: Elaborada pelo autor.

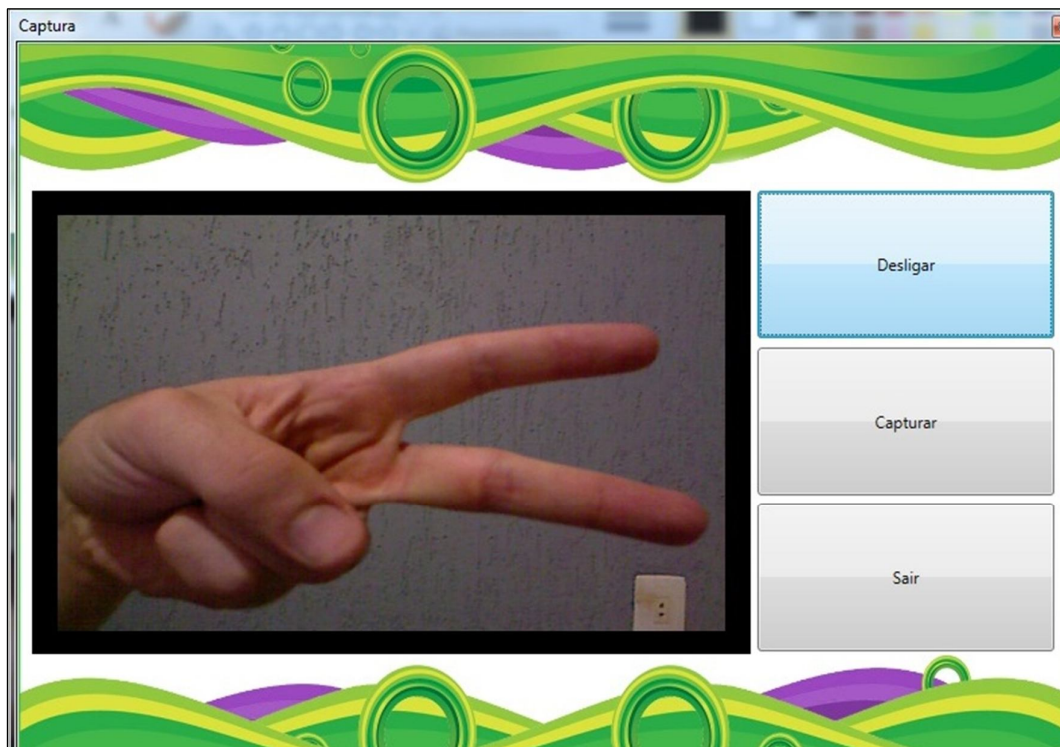


Figura 13 - Gesto que representa o número 2.
Fonte: Elaborada pelo autor.

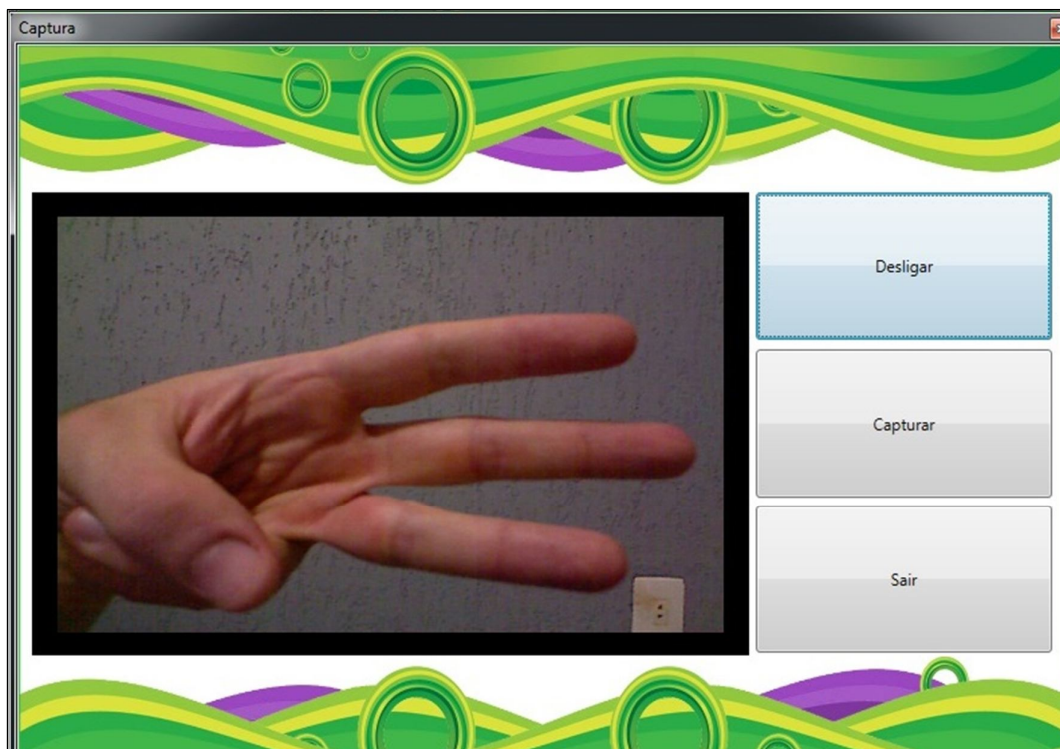


Figura 14 - Gesto que representa o número 3.
Fonte: Elaborada pelo autor.

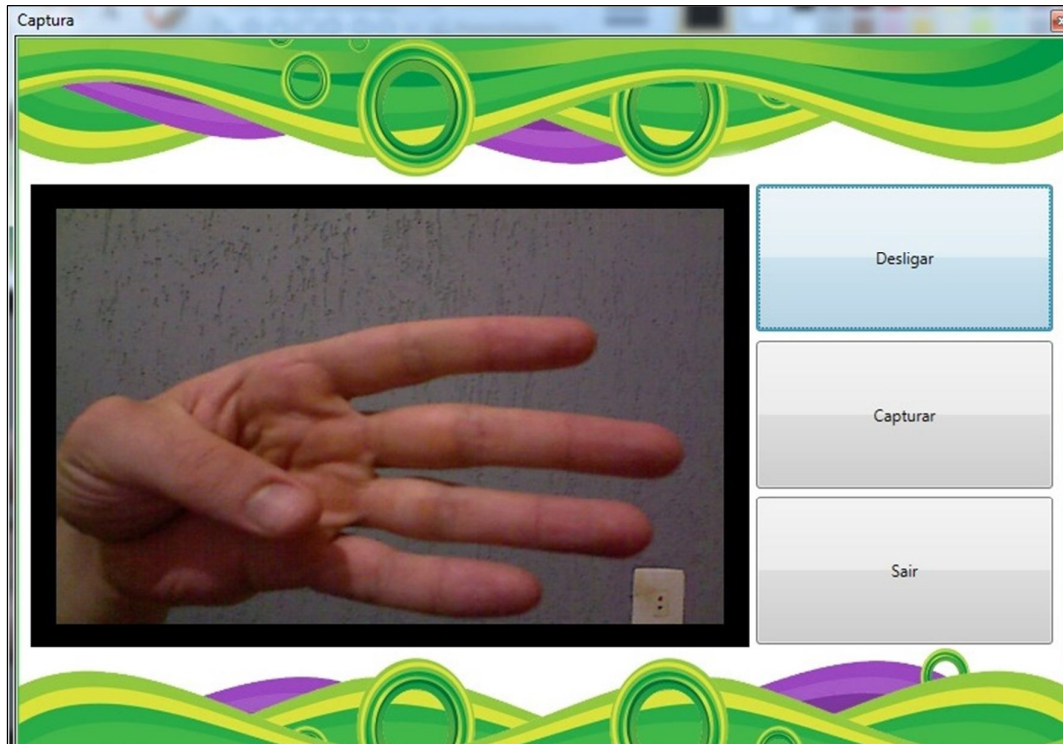


Figura 15 - Gestos que representa o número 4.
Fonte: Elaborada pelo autor.

A detecção dos pequenos e precisos movimentos dos dedos não é uma capacidade nativa do sensor Kinect. Para superar esta limitação foi necessário fazer um levantamento sobre qual a melhor maneira de estimar as articulações dos dedos das mãos, a fim de melhorar a interpretação dos dados. Ainda nesta etapa de investigação teórica foi realizado um estudo detalhado sobre os principais algoritmos de segmentação por diferença (*background subtraction*). Tal escolha foi baseada em um estudo preliminar de trabalhos correlatos a esta investigação, cujo resultado apontou para maior adequação de tais algoritmos aos objetivos deste trabalho. Especificamente foram estudadas as potencialidades e possíveis formas de implementação dos seguintes algoritmos: subtração simples, *running gaussian average*, *temporal median filter* e mistura de gaussianas. Os principais problemas da detecção de movimento por segmentação por diferenças, bem como seus possíveis impactos no desenvolvimento do software proposto foram também analisados. De modo particular, foram analisados os seguintes problemas: alterações de brilho, sombras e fantasmas. Também foram analisados os impactos das configurações automáticas das câmeras do Kinect durante a

aquisição de imagens bem como a implementação de técnicas de pós-processamento (remoção de ruídos, identificação de blobs, *area thresholding*, *morphological closing*, *saliency test* e *optical flow test*) que possam contribuir para a melhor segmentação das imagens.

A segunda etapa consistiu no estudo pontual sobre tecnologia usada pelo Kinect e seu funcionamento. Nesta etapa foram realizadas atividades que incluíram pesquisas sobre o funcionamento do driver do dispositivo com vistas a utilizá-lo no desenvolvimento da aplicação foco deste projeto. Esta etapa teve como resultado a indicação sobre a forma mais adequada de integrar os dados provenientes do Kinect à aplicação de forma que foi possível de manipular esses dados para a geração de imagens segmentadas (etapa 3). Cabe destacar que a segmentação bem sucedida consiste em obter imagens que possam ser utilizadas para o reconhecimento do gesto, de forma que o software de reconhecimento de gestos possa comparar o movimento emitido ao dispositivo. A segmentação de imagens referentes aos gestos, ainda que de um conjunto limitado aos dígitos, pôde-se revelar uma tarefa complicada, seja pela necessidade de recursos com elevada capacidade de processamento ou pela falta de qualidade na detecção dos elementos gestuais utilizando o Kinect. Neste último caso, se o problema ocorresse, foi feito um levantamento de adaptações de forma a diminuir a complexidade do sistema, reduzindo a necessidade de recursos, com o desenvolvimento de algoritmos que fossem demasiadamente complexos.

Para a etapa de desenvolvimento foram utilizados como recursos um equipamento notebook do fabricante DELL com processador Intel Core i5 2.40GHz, 4gb de memória RAM DDR3, Disco rígido de 500Gb, com *Windows Seven Ultimate* 64 bits como Sistema Operacional, e de um dispositivo Microsoft Kinect. Os drivers de comunicação do dispositivo Kinect com o computador podem ser obtidos gratuitamente no site oficial do SDK.

O Software foi desenvolvido utilizando a Linguagem de Programação C SHARP (C#) e a plataforma de programação gráfica XNA Game Studio. O objetivo desse software é reconhecer gestos emitidos ao dispositivo Kinect, através de elementos gestuais. Esse Software recebeu os dados do dispositivo e interpretou os movimentos, convertendo-os em números. As imagens capturadas pelo dispositivo Kinect que

necessitaram de algum tipo de armazenamento foram manipuladas pelo Sistema gerenciador de Banco de Dados MS SQL SERVER.

O módulo implementado compreende as etapas de aquisição de dados, preparação de dados e interpretação de dados. A figura seguinte mostra estas etapas, destacando o fluxo de funcionamento do sistema.

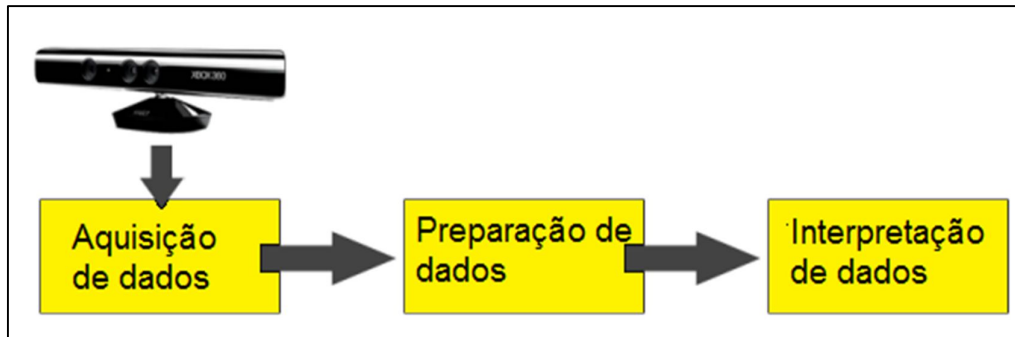


Figura 16 - Funcionamento do sistema proposto.
Fonte: Elaborada pelo autor.

A aquisição refere-se à programação utilizando o SDK do Kinect para capturar os dados dos sensores e armazená-los em estruturas para serem encaminhados à etapa de preparação. Na fase de aquisição dos dados é feita a captura das imagens pelos sensores do Kinect, considerando a câmera de vídeo, o sensor de profundidade e a informação do corpo humano. Os parâmetros de aquisição de imagem (resolução da profundidade e da câmera de vídeo, limitadas à 320X240 e 640X480 respectivamente). A forma de captura dos frames pelos sensores também pode ser configurada. O modelo de captura escolhido foi o *Event Model*, que consiste em transmitir quando um botão foi pressionado, executando um evento previamente configurado, para que assim o usuário do software tenha livre controle de quando a imagem será capturada. Na etapa de preparação de dados, as imagens fornecidas pelo Kinect foram formatadas com algoritmos de processamento de imagens.

Primeiramente, as imagens passaram pelo algoritmo de suavização (*Smoothing*), feito isso, foi passado o algoritmo de subtração de fundo (*Background Subtraction*). Para resolver o problema com sombras (*Shadows*), as imagens passaram por um algoritmo de tratamento de sombras (*Shadows*) e, por último, o algoritmo de binarização foi passado para deixar na cor branca os pixels onde existe movimento e na

cor preta onde não existe. Mesmo com o algoritmo de tratamento de sombras (*Shadows*), permanece uma linha na cor preta entre a sombra e o objeto e para resolver esse problema foi feito um controle do tamanho de pixels desta linha, assim, pôde-se resolver o problema.

Para o reconhecimento dos dedos foram traçadas três linhas, sendo uma horizontal e duas verticais, podendo reconhecer os dedos tanto da mão esquerda, quanto da direita. Na posição das linhas traçadas, um contador foi incrementado a cada vez em que passava por um pixel branco, podendo, assim, identificar a presença dos dedos.

A preparação de dados compreende, sobretudo o refinamento das imagens obtidas da etapa anterior. Após capturar os quadros de cor e de profundidade na etapa de aquisição, verificou-se os corpos humanos reconhecidos pelo hardware do dispositivo. Para isso, percorreu-se todos os quadros de profundidade e realizou-se a verificação de cada pixel (os pixels indicados por 1 mostram aqueles pontos que pertencem a um determinado corpo). Nesta etapa, o sistema foi capaz de processar os pixels referentes ao elementos gestuais (precisamente a mão e os dedos), gerando as imagens segmentadas. Por fim, na última etapa, as imagens estavam preparadas para ser interpretadas, de modo a indicar de forma clara o dígito correspondente ao gesto capturado.

7 – DESENVOLVIMENTO

Esta seção descreve os processos utilizados no desenvolvimento do sistema implementado neste trabalho.

7.1 – ALGORITMOS

7.1.1 – ALGORITMO DE EXIBIÇÃO DE IMAGEM

Para exibir a imagem do dispositivo Kinect na tela, foi utilizada a classe *KinectSensor*, que é uma das classes mais importantes no processo de programação utilizando Kinect. É na classe *KinectSensor* onde se encontram as informações sobre o dispositivo, quantidade de dispositivos conectados na máquina, status, configurações, etc.

O primeiro passo foi atribuir a classe à variável chamada “sensor”, para que fosse possível ter acesso às informações do dispositivo. Como foi utilizado apenas um dispositivo Kinect para o desenvolvimento deste software, atribuiu-se o primeiro dispositivo armazenado no vetor chamado “*KinectSensor.KinectSensors[]*”. Um vetor armazena uma série de elementos e o gerenciamento funciona com a ajuda de um índice, utilizado para percorrer todas as posições do vetor. De modo específico o índice utilizado foi o zero, que indica a primeira posição do vetor (Figura 17).

O segundo passo foi habilitar a câmera RGB, para isso foi utilizada a classe *ColorImageFormat*. Nesta classe definiu-se a resolução da câmera utilizada em 640x480 (Figura 17).

Feito isso, foi associado o dispositivo aos frames de imagem, isso é feito para indicar que o dispositivo está pronto para ligar a câmera RGB (Figura 17).

```
//Pega o sensor conectado
sensor = KinectSensor.KinectSensors[0];

//Habilita a camera RGB
sensor.ColorStream.Enable(ColorImageFormat.RgbResolution640x480Fps30);

//Associo o evento ao AllFrameReady
sensor.AllFramesReady += new EventHandler<AllFramesReadyEventArgs>(sensor_AllFramesReady);
```

Figura 17 - Atribuição do dispositivo e câmera RGB.
Fonte: Elaborada pelo autor.

Para que fosse possível exibir a imagem do dispositivo na tela, foi necessário criar uma variável do tipo *BitmapImage*, que representa uma imagem à partir de um conjunto de pixels. Procurando dividir a imagem em pixels, foram utilizados os frames obtidos através do dispositivo para dividi-los em vários pixels e armazená-los em um vetor do tipo byte. Com o auxílio do método *PixelDataLength* foi possível ter acesso às informações relacionadas aos pixels (Figura 18).

Buscando reaproveitar este código foi criada outra variável do tipo *BitmapImage* com o nome de “imagemSalva”. Essa variável foi utilizada no momento da gravação da imagem para o formato .JPG que será apresentado no próximo tópico (Figura 18).

Feito isso, foi atribuído o valor da variável do tipo *BitmapImage* ao elemento *Image* da tela, a atribuição foi feita para que o elemento receba os valores que estão contidos na variável (Figura 18).

```
//Se não vier nada então sai do metodo
if (cframe == null)
{
    return;
}
//Array de bytes com o tamanho do frame
byte[] cbytes = new byte[cframe.PixelDataLength];

//Copia os bytes para o array de bytes
cframe.CopyPixelDataTo(cbytes);

//Numero de bytes por linha / width * (RGB Vazio)
int stride = cframe.Width * 4;

//Cria um bitmap baseado nos bytes, passando o tamanho da imagem (640x480), o dpi (no nosso caso é um padrão 96x96 mesmo)
//passando o formato do pixel (32 bits), a paleta (nulo), os bytes (cbytes) e o stride
imgKinect.Source = BitmapImage.Create(640, 480, 96, 96, PixelFormats.Bgr32, null, cbytes, stride);
var imagemSalva = BitmapImage.Create(640, 480, 96, 96, PixelFormats.Bgr32, null, cbytes, stride);
imagemJpg = imagemSalva;
```

Figura 18 - Divisão da imagem da câmera em pixels e criação das BitmapImages.
Fonte: Elaborada pelo autor.

Com a classe *Kinect Sensor*, foi possível controlar as ações de ligar/desligar do sensor Kinect (Figura 19).

```
private void btn_ligar_Click_1(object sender, RoutedEventArgs e)
{
    //Inicia o sensor
    if (sensor.IsRunning)
    {
        sensor.Stop();
        imgKinect.Source = null;
        btn_ligar.Content = "Ligar";
    }
    else
    {
        sensor.Start();
        btn_ligar.Content = "Desligar";
    }
}
```

Figura 19 - Controle de ativação do sensor com o uso da classe KinectSensor.
Fonte: Elaborada pelo autor.

7.1.2 – ALGORITMO DE CAPTURA DE IMAGEM

Para que fosse possível salvar a imagem do dispositivo Kinect em uma imagem no formato jpg, uma das classes utilizadas foi a *SaveFileDialog*. Com a ajuda desta classe, tem-se acesso ao *browser* do *Windows*, podendo, assim, procurar o diretório, escolher o nome que a imagem foi salva e o filtro com que o browser seria inicializado (Figura 20).

Feito isso, foi criado um codificador com a ajuda da classe *JpegBitmapEncoder*, usado para formatar a imagem para o tipo jpg. Com a imagem já no formato jpg, foram utilizados os recursos da classe *BitmapFrame* para criar a imagem no formato jpg. Também foi utilizada a classe *FileStream*, passando as informações obtidas através da classe *SaveFileDialog* e o método *FileMode.Create*, indicando a utilização do recurso para criar a imagem (Figura 20).

Para finalizar, foi utilizado o método *encoder.Save* passando os dados da classe *FileStream* para que fosse possível salvar a imagem com o nome e diretório escolhidos (Figura 20).

```

SaveFileDialog saveFileDialog = new SaveFileDialog();
saveFileDialog.Filter = "Arquivos de Imagem (*.jpg)|*.jpg|Show All Files (*.*)|*.*";
saveFileDialog.Title = "Save As";
if (saveFileDialog.ShowDialog(this) == true)
{
    JpegBitmapEncoder encoder = new JpegBitmapEncoder();

    encoder.Frames.Add(BitmapFrame.Create((imagemJpg)));
    using (var filestream = new FileStream(saveFileDialog.FileName, FileMode.Create))
        encoder.Save(filestream);
}

```

Figura 20 - Algoritmo de captura da imagem para o formato jpg.
Fonte: Elaborada pelo autor.

7.1.3 – ALGORITMO PARA CARREGAR IMAGENS NA TELA

Antes de realizar a segmentação, foi necessário carregar as duas imagens na tela, a imagem do fundo e a imagem do sinal. Para carregar as imagens na tela foi utilizado os recursos da classe *OpenFileDialog*, que tem o funcionamento semelhante ao da classe *SaveFileDialog*. Esta classe oferece acesso ao *browser* do *Windows*, podendo assim, procurar o diretório onde as imagens foram salvas, podendo filtrar a busca por arquivos no formato jpg (Figura 21).

Posteriormente foi atribuída a imagem que foi carregada à partir da classe *OpenFileDialog* à variável do elemento *Image* da tela, fazendo com que o elemento da tela contenha a imagem carregada. Logo após a imagem ser carregada na tela, aplica-se o algoritmo de suavização da imagem (que será apresentado no próximo tópico), para se obter uma melhor segmentação (Figura 21).

```

OpenFileDialog openFileDialog = new OpenFileDialog();
openFileDialog.Filter = "Arquivos de Imagem (*.jpg)|*.jpg|Show All Files (*.*)|*.*";
openFileDialog.Title = "Browse Image Files";
openFileDialog.DefaultExt = ".jpg";
if (openFileDialog.ShowDialog(this) == true)
{
    imgJPG_1.Source = new BitmapImage(new Uri(openFileDialog.FileName));
    imagem_1 = new Bitmap(openFileDialog.FileName);
    double larg_1 = 0;
    Aplicar_Smooth(larg_1, imagem_1);
}

```

Figura 21 - Carregar imagens na tela e chamada do método de suavização.
Fonte: Elaborada pelo autor.

7.1.4 – ALGORITMO DE SUAVIZAÇÃO

Causado por erros na transmissão de dados, os ruídos são pixels corrompidos ou modificados que provocam uma diferença brusca de tons entre este pixel e seus vizinhos.

Buscando a remoção dos ruídos encontrados nas imagens de fundo e do sinal, foi aplicado um algoritmo de suavização.

O primeiro passo foi atribuir o valor da imagem que foi carregada na tela à uma nova variável do tipo *Bitmap*, isso foi feito para evitar alterações na imagem original, podendo, assim, manter a imagem original armazenada (Figura 22).

O segundo passo baseou-se em criar duas matrizes, uma do tipo *double* que foi atribuída com o valor passado por parâmetro no chamado do método e outra do tipo *Color*, uma matriz tem o funcionamento semelhante ao vetor, porém, a matriz funciona com o uso de dois índices (Figura 22).

O terceiro passo consistiu em implementar duas estruturas de repetição do tipo *For*. Tal estrutura consiste em executar um algoritmo passando um índice com valor inicial, condição de parada e o modo de incremento do índice, fazendo com que o algoritmo fosse executado até que a condição de parada fosse verdadeira. Uma das estruturas teve como condição de parada a altura da imagem e a outra a largura, deste modo, era percorrida toda a imagem (Figura 22).

A cada passagem da estrutura de repetição, o algoritmo primeiramente alimenta as posições da matriz com os pixels de cor azul contidos na imagem, isso foi feito com a utilizando o método *GetPixel*, já que ele permite ter acesso ao valor do pixel. O próximo passo do algoritmo foi atribuir o valor da posição [1,1] da matriz à variável “A”, que foi usada no decorrer do algoritmo (Figura 22).

No caso da variável “R”, foi atribuída uma conversão para número inteiro da soma entre os pixels de cor vermelha da imagem multiplicados pelo valor da matriz com o valor passado via parâmetro, e o resultado foi multiplicado pelo fator e offset (ambos tinham valor 1) (Figura 22).

```

Bitmap newImg = (Bitmap)b.Clone();
System.Drawing.Color[,] pixelColor = new System.Drawing.Color[3, 3];
int A, R, G, B;

for (int y = 0; y < b.Height - 2; y++)
{
    for (int x = 0; x < b.Width - 2; x++)
    {
        pixelColor[0, 0] = b.GetPixel(x, y);
        pixelColor[0, 1] = b.GetPixel(x, y + 1);
        pixelColor[0, 2] = b.GetPixel(x, y + 2);
        pixelColor[1, 0] = b.GetPixel(x + 1, y);
        pixelColor[1, 1] = b.GetPixel(x + 1, y + 1);
        pixelColor[1, 2] = b.GetPixel(x + 1, y + 2);
        pixelColor[2, 0] = b.GetPixel(x + 2, y);
        pixelColor[2, 1] = b.GetPixel(x + 2, y + 1);
        pixelColor[2, 2] = b.GetPixel(x + 2, y + 2);

        A = pixelColor[1, 1].A;

        R = (int)((((pixelColor[0, 0].R * m.Matrix[0, 0]) +
                    (pixelColor[1, 0].R * m.Matrix[1, 0]) +
                    (pixelColor[2, 0].R * m.Matrix[2, 0]) +
                    (pixelColor[0, 1].R * m.Matrix[0, 1]) +
                    (pixelColor[1, 1].R * m.Matrix[1, 1]) +
                    (pixelColor[2, 1].R * m.Matrix[2, 1]) +
                    (pixelColor[0, 2].R * m.Matrix[0, 2]) +
                    (pixelColor[1, 2].R * m.Matrix[1, 2]) +
                    (pixelColor[2, 2].R * m.Matrix[2, 2]))
                / m.Factor) + m.Offset);
    }
}

```

Figura 22 - Algoritmo de suavização parte 1.
 Fonte: Elaborada pelo autor.

Posteriormente foi feita uma verificação utilizando o algoritmo de condição *If*. Essa condição foi feita para verificar se o valor da variável “R”. Caso este valor fosse maior do que 255, seria atribuído o valor 255 e se fosse menor do que zero, seria atribuído o valor 0. Isso é feito pois os valores de cor possuem o valor mínimo de 0 e máximo de 255 (Figura 23).

O processo realizado para a variável “R” foi repetido também para as variáveis “G” (Figura 23) e “B” (Figura 24).

```

if (R < 0)
{
    R = 0;
}
else if (R > 255)
{
    R = 255;
}

G = (int)((((pixelColor[0, 0].G * m.Matrix[0, 0]) +
            (pixelColor[1, 0].G * m.Matrix[1, 0]) +
            (pixelColor[2, 0].G * m.Matrix[2, 0]) +
            (pixelColor[0, 1].G * m.Matrix[0, 1]) +
            (pixelColor[1, 1].G * m.Matrix[1, 1]) +
            (pixelColor[2, 1].G * m.Matrix[2, 1]) +
            (pixelColor[0, 2].G * m.Matrix[0, 2]) +
            (pixelColor[1, 2].G * m.Matrix[1, 2]) +
            (pixelColor[2, 2].G * m.Matrix[2, 2]))
          / m.Factor) + m.Offset);

if (G < 0)
{
    G = 0;
}
else if (G > 255)
{
    G = 255;
}

```

Figura 23 - Algoritmo de suavização parte 2.
Fonte: Elaborada pelo autor.

```

B = (int)((((pixelColor[0, 0].B * m.Matrix[0, 0]) +
            (pixelColor[1, 0].B * m.Matrix[1, 0]) +
            (pixelColor[2, 0].B * m.Matrix[2, 0]) +
            (pixelColor[0, 1].B * m.Matrix[0, 1]) +
            (pixelColor[1, 1].B * m.Matrix[1, 1]) +
            (pixelColor[2, 1].B * m.Matrix[2, 1]) +
            (pixelColor[0, 2].B * m.Matrix[0, 2]) +
            (pixelColor[1, 2].B * m.Matrix[1, 2]) +
            (pixelColor[2, 2].B * m.Matrix[2, 2]))
          / m.Factor) + m.Offset);

if (B < 0)
{
    B = 0;
}
else if (B > 255)
{
    B = 255;
}

```

Figura 24 - Algoritmo de suavização parte 3.
Fonte: Elaborada pelo autor.

Para finalizar a estrutura de repetição, um novo pixel é gerado à partir dos resultados obtidos durante o algoritmo, isso foi feito com a ajuda do método *SetPixel*. Por meio dele é possível determinar o valor de um pixel à partir dos valores de *alpha*, *red*, *green* e *blue* que são as respectivas variáveis “A”, “R”, “G” e “B” (Figura 25).

Todo esse processamento é chamado à partir do método “Aplicar_Smooth” que, ao final, gera uma nova imagem já suavizada à partir dos pixels gerados pelo algoritmo (Figura 25).

```

        b.SetPixel(x + 1, y + 1, System.Drawing.Color.FromArgb(A, R, G, B));
    }
}
return b;
}

public void Aplicar_Smooth(double weight, Bitmap imagem_smooth) // APLICA A SUAVIZAÇÃO NA IMAGEM
{
    ConvolutionMatrix matrix = new ConvolutionMatrix(3);
    matrix.SetAll(1);
    matrix.Matrix[1, 1] = weight;
    matrix.Factor = weight + 8;
    imagem_smooth = Convolution3x3(imagem_smooth, matrix);
}

```

Figura 25 - Algoritmo de suavização parte 4.
Fonte: Elaborada pelo autor.

7.1.5 – ALGORITMO DE SUBTRAÇÃO DE FUNDO

Esta corresponde à parte mais importante do software, pois, sua execução torna possível obter a imagem somente do sinal emitido ao dispositivo, sem o fundo.

Primeiramente foram criadas cinco variáveis do tipo *Bitmap*. A variável “imagem_11” é uma cópia da primeira imagem exibida na tela, criada para que fosse utilizada durante o algoritmo de subtração de fundo. O mesmo aconteceu para a variável “imagem_22” mas, nesse caso, para a segunda imagem exibida na tela (Figura 26).

A variável “bmp” foi criada para receber a imagem resultado da subtração da primeira imagem da segunda. Já para a variável “bmp_2” acontece o inverso, ela recebe o resultado da subtração da segunda imagem da primeira. Para obter a imagem proveniente da soma entre as duas variáveis obtidas através da subtração, foi criada a variável “bmp_final” (Figura 26).

Para receber o valor das cores dos pixels da imagem, foram criadas três variáveis, “r” para receber o valor da cor vermelha, “g” para receber o valor da cor verde e “b” recebendo o valor da cor azul (Figura 26).

```

Bitmap imagem_11 = new Bitmap(imagem_1, largura, altura);
Bitmap imagem_22 = new Bitmap(imagem_2, largura, altura);
Bitmap bmp       = new Bitmap(imagem_11.Width, imagem_22.Height);
Bitmap bmp_2     = new Bitmap(imagem_11.Width, imagem_22.Height);
Bitmap bmp_FINAL = new Bitmap(imagem_11.Width, imagem_22.Height);
int r, g, b;

```

Figura 26 - Variáveis usadas no algoritmo de subtração de fundo.
Fonte: Elaborada pelo autor.

Feito isso, o passo seguinte foi realizar a primeira subtração entre imagens, a subtração foi feita dentro de duas estruturas de repetição do tipo *For* com o intuito de eliminar os pixels do fundo da imagem e manter apenas os pixels onde houve movimento. As duas estruturas começaram com índice no valor zero, porém, uma com condição de parada contendo a largura da imagem e outra contendo a altura da mesma (Figura 27).

A cada passagem pela estrutura de repetição, é feita a subtração entre o valor do pixel vermelho da primeira imagem e do valor do pixel vermelho da segunda imagem, o resultado dessa subtração é atribuído a variável “r”, convertendo o valor para inteiro. Logo após, é feito um algoritmo de condição do tipo *If* para verificar se o valor do pixel é menor do que zero, caso isso ocorra, é atribuído o valor zero, pois, este é o valor mínimo do pixel (Figura 27).

O procedimento realizado para o pixel da cor vermelha, é também realizado para o pixel da cor verde, atribuindo o valor da subtração para a variável “g” e para o pixel da cor azul, atribuindo o valor da subtração para a variável “b” (Figura 27).

Para finalizar a primeira estrutura de repetição, foi criado um pixel com as cores obtidas através das subtrações feitas anteriormente, a criação do pixel foi feita com a ajuda do método *SetPixel*, passando os valores das variáveis “r”, “g” e “b” (Figura 27).

```
for (int x = 0; x < imagem_11.Width; x++)
{
    for (int y = 0; y < imagem_11.Height; y++)
    {
        r = (int)(imagem_11.GetPixel(x, y).R - imagem_22.GetPixel(x, y).R);

        if (r < 0)
        {
            r = 0;
        }

        g = (int) (imagem_11.GetPixel(x,y).G - imagem_22.GetPixel(x,y).G);

        if (g < 0)
        {
            g = 0;
        }

        b = (int) (imagem_11.GetPixel(x,y).B - imagem_22.GetPixel(x,y).B);

        if (b < 0)
        {
            b = 0;
        }

        bmp.SetPixel(x, y, System.Drawing.Color.FromArgb(r, g, b));
    }
}
```

Figura 27 - Primeira estrutura de repetição para subtração de fundo.
Fonte: Elaborada pelo autor.

O procedimento executado na primeira estrutura de repetição é executado novamente, porém, desta vez é feita a subtração da segunda imagem pela primeira (Figura 28).

```

for (int x = 0; x < imagem_11.Width; x++)
{
    for (int y = 0; y < imagem_11.Height; y++)
    {
        r = (int)(imagem_22.GetPixel(x, y).R - imagem_11.GetPixel(x, y).R);

        if (r < 0)
        {
            r = 0;
        }

        g = (int)(imagem_22.GetPixel(x, y).G - imagem_11.GetPixel(x, y).G);

        if (g < 0)
        {
            g = 0;
        }

        b = (int)(imagem_22.GetPixel(x, y).B - imagem_11.GetPixel(x, y).B);

        if (b < 0)
        {
            b = 0;
        }

        bmp_2.SetPixel(x, y, System.Drawing.Color.FromArgb(r, g, b));
    }
}

```

Figura 28 - Segunda estrutura de repetição para subtração de fundo.
 Fonte: Elaborada pelo autor.

O próximo passo realizado pelo algoritmo de subtração de fundo corresponde a somar os pixels das imagens obtidas através das duas estruturas de repetição para subtração de imagem (Figura 29).

O procedimento realizado é similar ao das estruturas de repetição anteriores, porém, ao invés de subtrair os pixels, é feita a soma dos pixels das duas imagens. No algoritmo de condição, é verificado se o valor do pixel é maior do que 255, caso isso seja constatado, é atribuído o valor 255 (o valor máximo do pixel) (Figura 29).

```

for (int x = 0; x < bmp.Width; x++)
{
    for (int y = 0; y < bmp.Height; y++)
    {
        r = (int)(bmp_2.GetPixel(x, y).R + bmp.GetPixel(x, y).R);

        if (r > 0)
        {
            r = 255;
        }

        g = (int)(bmp_2.GetPixel(x, y).G + bmp.GetPixel(x, y).G);

        if (g > 0)
        {
            g = 255;
        }

        b = (int)(bmp_2.GetPixel(x, y).B + bmp.GetPixel(x, y).B);

        if (b > 0)
        {
            b = 255;
        }

        bmp_FINAL.SetPixel(x, y, System.Drawing.Color.FromArgb(r, g, b));
    }
}

```

Figura 29 - Terceira estrutura de repetição para subtração de fundo.
Fonte: Elaborada pelo autor.

Por fim, a imagem final é salva no formato jpg de acordo com os parâmetros passados na classe *SaveFileDialog*. O método responsável por salvar a imagem é o *Bitmap.Save* (Figura 30).

```

bmp_FINAL.Save(saveFileDialog.FileName, ImageFormat.Jpeg);

```

Figura 30 - Método para salvar imagem final.
Fonte: Elaborada pelo autor.

7.1.6 – ALGORITMO DE BINARIZAÇÃO

Após ser realizado o algoritmo de subtração de fundo, foi realizado o algoritmo de binarização, que consiste em alterar a cor para branco dos pixels onde houve movimento e para preto onde não houve movimento (Figura 31).

O processo foi semelhante ao utilizado para o algoritmo de subtração de fundo, assim, foram usadas duas estruturas de repetição, uma percorrendo a imagem na horizontal e a outra percorrendo-a na vertical, porém, onde era feita a subtração de pixels, foram feitas três estruturas de condição verificando se o valor do pixel era maior do que 35, isso indicava um pixel mais claro, que por sua vez indica que houve movimento. O procedimento foi realizado para os pixels na cor vermelha, armazenado na variável “r”, para o pixel na cor verde, armazenado na variável “g” e para o pixel na cor azul, armazenado na variável “b” (Figura 31).

Caso o valor do pixel fosse maior do que 35, era atribuída a cor branca para o pixel, utilizando o método *bitmap.SetPixel*, utilizando a classe *Color* (Figura 31).

```
for (int i = 0; i < bmp_2.Width; i++)
{
    for (int j = 0; j < bmp_2.Height; j++)
    {
        r = (int)(bmp_2.GetPixel(i, j).R);

        if (r > 35)
        {
            bmp_2.SetPixel(i, j, System.Drawing.Color.White);
        }

        g = (int)(bmp_2.GetPixel(i, j).G);

        if (g > 35)
        {
            bmp_2.SetPixel(i, j, System.Drawing.Color.White);
        }

        b = (int)(bmp_2.GetPixel(i, j).B);

        if (b > 35)
        {
            bmp_2.SetPixel(i, j, System.Drawing.Color.White);
        }
    }
}
```

Figura 31 - Algoritmo de binarização parte 1.
Fonte: Elaborada pelo autor.

Após este procedimento, a parte da imagem em que houve movimentação estará na cor branca. Tal procedimento feito é executado novamente, dessa vez, verificando se o valor dos pixels eram diferentes de 255, o que indica cor branca. Caso o valor do pixel fosse diferente de 255, era atribuída a cor preta para o pixel (Figura 32).

```
for (int i = 0; i < bmp.Width; i++)
{
    for (int j = 0; j < bmp.Height; j++)
    {
        r = (int)(bmp_2.GetPixel(i, j).R);
        g = (int)(bmp_2.GetPixel(i, j).G);
        b = (int)(bmp_2.GetPixel(i, j).B);

        if (r != 255 | g != 255 | b != 255)
        {
            bmp_2.SetPixel(i, j, System.Drawing.Color.Black);
        }
    }
}
```

Figura 32 - Algoritmo de binarização parte 2.
Fonte: Elaborada pelo autor.

7.1.7 – ALGORITMO DE IDENTIFICAÇÃO

Após aplicar os algoritmos apresentados, pôde-se obter a imagem em seu estado final, ou seja, pronta para aplicação do algoritmo de identificação. O algoritmo de identificação consiste em identificar qual o número correspondente ao sinal emitido.

Para realizar a identificação, foram feitas três linhas imaginárias na imagem, sendo uma horizontal na parte superior da imagem e duas verticais, uma no começo e outra no final da imagem (Figura 33).

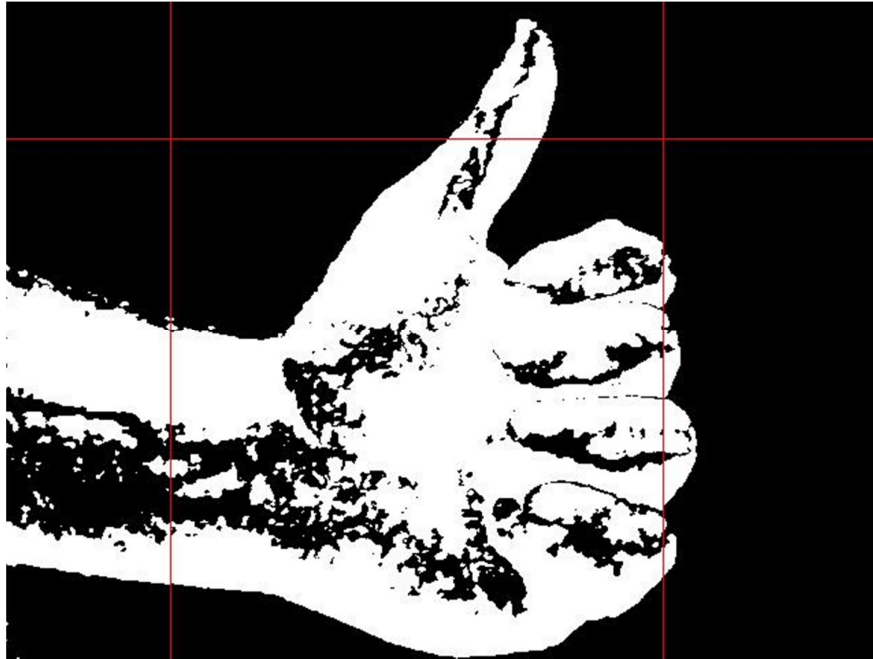


Figura 33 - Linhas imaginárias criadas para realizar a detecção do sinal.
Fonte: Elaborada pelo autor.

No espaço onde foram feitas as linhas, verificou-se a quantidade de pixels de cor branca existentes e a quantidade de pixels de cor preta entre os pixels de cor branca, podendo, deste modo, identificar a quantidade de dedos presentes na imagem e assim, identificar o número referente ao sinal.

Para o desenvolvimento do algoritmo de identificação, foram usadas nove novas variáveis: “imagem_final” (do tipo *Bitmap*, foi criada para armazenar a imagem carregada na tela), “contador_pixel_branco_HORIZONTAL” (criada para armazenar a quantidade de pixels de cor branca encontrados no espaço ocupado pela linha horizontal), “contador_dedos_HORIZONTAL” (criada para armazenar a quantidade de dedos existentes no espaço ocupado pela linha horizontal), “contador_pixel_branco_VERTICAL_ESQUERDA” (criada para receber a quantidade de pixels de cor branca encontrados no espaço ocupado pela linha vertical no início da imagem), “contador_dedos_VERTICAL_ESQUERDA” (criada para receber a quantidade de dedos existentes no espaço ocupado pela linha vertical no início da imagem), “contador_pixel_branco_VERTICAL_DIREITA” (criada para receber a quantidade de pixels de cor branca encontrados no espaço ocupado pela linha vertical no final da imagem), “contador_dedos_VERTICAL_DIREITA” (criada para receber a quantidade de dedos existentes no espaço ocupado pela linha vertical no final da

imagem), “contador_dedos_TOTAL” (criada para receber a soma de todas as variáveis responsáveis pela quantidade de dedos existentes na imagem), “intervalo” (criada para receber a quantidade de pixels de cor preta existentes entre os pixels de cor branca). Além disso, também são utilizadas as variáveis já citadas anteriormente “r”, “g” e “b” (Figura 34).

```
public Bitmap imagem_final;
public int contador_pixel_branco_HORIZONTAL = 0;
public int contador_dedos_HORIZONTAL = 0;
public int contador_pixel_branco_VERTICAL_ESQUERDA = 0;
public int contador_dedos_VERTICAL_ESQUERDA = 0;
public int contador_pixel_branco_VERTICAL_DIREITA = 0;
public int contador_dedos_VERTICAL_DIREITA = 0;
public int contador_dedos_TOTAL = 0;
public int intervalo = 0;
public int r;
public int g;
public int b;
```

Figura 34 - Variáveis utilizadas no algoritmo de identificação.
Fonte: Elaborada pelo autor.

Na primeira parte do algoritmo, foi feita uma estrutura de repetição que percorre a imagem na horizontal, traçando uma linha reta imaginária até o final da imagem (Figura 35). Depois, atribuiu-se os valores das quantidades de cores vermelha, verde e azul para as respectivas variáveis e, então, verificou-se se os valores das variáveis “r” ou “g” ou “b” eram iguais a zero, caso fossem, o pixel analisado seria da cor preta, caso contrário, seria da cor branca (Figura 35).

No caso do pixel de cor preta, era verificado se o contador de pixels de cor branca tinha valor maior do que 40. Tal procedimento indica se já houve a passagem de pixels de cor branca o suficiente para ser analisado como sendo um dedo, uma vez que 40 é, em média, a quantidade de pixels branco de um dedo. Cabe destacar que esse valor foi empiricamente definido, considerando os dados coletados nesta investigação (Figura 35).

Caso o contador de pixels de cor branca tenha valor superior a 40, é adicionado 1 ao valor do intervalo e também verificado se o intervalo é maior do que 5, o que indica que já não faz parte do sinal, sendo assim, é adicionado 1 ao contador de dedos da linha e atribuído o valor 0 para o intervalo e ao contador de pixels de cor branca, podendo, assim, ser realizada outra verificação (Figura 35).

Caso as cores das variáveis “r” ou “g” ou “b” sejam diferentes de zero, indicando que o pixel é de cor branca, é adicionado 1 ao contador de pixels de cor branca e atribuído o valor 0 para o intervalo, indicando que o pixel pertence ao sinal (Figura 35).

```
//Verifica a quantidade de dedos na horizontal
for (int i = 0; i < imagem_final.Width; i++)
{
    r = (int)(imagem_final.GetPixel(i, 100).R);
    g = (int)(imagem_final.GetPixel(i, 100).G);
    b = (int)(imagem_final.GetPixel(i, 100).B);

    if (r == 0 | g == 0 | b == 0)
    {
        if (contador_pixel_branco_HORIZONTAL > 40)
        {
            intervalo = intervalo + 1;

            if (intervalo > 5)
            {
                contador_dedos_HORIZONTAL = contador_dedos_HORIZONTAL + 1;
                contador_pixel_branco_HORIZONTAL = 0;
                intervalo = 0;
            }
        }
    }
    else
    {
        contador_pixel_branco_HORIZONTAL = contador_pixel_branco_HORIZONTAL + 1;
        intervalo = 0;
    }
}
}
```

Figura 35 - Verificação da quantidade de dedos na linha horizontal.
Fonte: Elaborada pelo autor.

O procedimento é feito também para as linhas vertical esquerda (Figura 36) e direita (Figura 37), porém, nas linhas verticais é feita uma nova verificação para checar se o contador de pixels de cor branca é maior do que 80, caso seja, seria indicado o braço contido na imagem.

```
for (int i = 0; i < imagem_final.Height; i++)
{
    r = (int)(imagem_final.GetPixel(120, i).R);
    g = (int)(imagem_final.GetPixel(120, i).G);
    b = (int)(imagem_final.GetPixel(120, i).B);

    if (contador_pixel_branco_VERTICAL_ESQUERDA > 80)
    {
        contador_dedos_VERTICAL_ESQUERDA = 0;
        break;
    }

    if (r == 0 | g == 0 | b == 0)
    {
        if (contador_pixel_branco_VERTICAL_ESQUERDA > 40)
        {
            intervalo = intervalo + 1;

            if (intervalo > 5)
            {
                contador_dedos_VERTICAL_ESQUERDA = contador_dedos_VERTICAL_ESQUERDA + 1;
                contador_pixel_branco_VERTICAL_ESQUERDA = 0;
                intervalo = 0;
            }
        }
    }
    else
    {
        contador_pixel_branco_VERTICAL_ESQUERDA = contador_pixel_branco_VERTICAL_ESQUERDA + 1;
        intervalo = 0;
    }
}
```

Figura 36 - Verificação da quantidade de dedos na linha vertical esquerda.
Fonte: Elaborada pelo autor.

```

for (int i = 0; i < imagem_final.Height; i++)
{
    r = (int)(imagem_final.GetPixel(480, i).R);
    g = (int)(imagem_final.GetPixel(480, i).G);
    b = (int)(imagem_final.GetPixel(480, i).B);

    if (contador_pixel_branco_VERTICAL_DIREITA > 80)
    {
        contador_dedos_VERTICAL_DIREITA = 0;
        break;
    }

    if (r == 0 | g == 0 | b == 0)
    {
        if (contador_pixel_branco_VERTICAL_DIREITA > 40)
        {
            intervalo = intervalo + 1;

            if (intervalo > 5)
            {
                contador_dedos_VERTICAL_DIREITA = contador_dedos_VERTICAL_DIREITA + 1;
                contador_pixel_branco_VERTICAL_DIREITA = 0;
                intervalo = 0;
            }
        }
    }
    else
    {
        contador_pixel_branco_VERTICAL_DIREITA = contador_pixel_branco_VERTICAL_DIREITA + 1;
        intervalo = 0;
    }
}
}

```

Figura 37 - Verificação da quantidade de dedos na linha vertical direita.
 Fonte: Elaborada pelo autor.

O próximo passo consiste em atribuir à variável “contador_dedos_TOTAL”, a soma de todas as variáveis responsáveis por armazenar a quantidade de dedos e converter o conteúdo da variável ao elemento *Label* da tela com a ajuda do método *ToString* (Figura 38).

```

contador_dedos_TOTAL = contador_dedos_HORIZONTAL + contador_dedos_VERTICAL_ESQUERDA + contador_dedos_VERTICAL_DIREITA;
lbl_resultado.Content = contador_dedos_TOTAL.ToString();

```

Figura 38 - Soma total da quantidade de dedos e atribuição à *Label*.
 Fonte: Elaborada pelo autor.

8 – RESULTADOS

Com o objetivo de avaliar o sistema, foram testados cinco sinais (gestos) em dois ambientes de teste (luz artificial e luz natural), avaliando-se o desempenho do sistema em seis casos de testes para cada sinal.

Buscando alcançar os objetivos deste trabalho, realizou-se uma sessão de testes em dois ambientes, com diferentes tipos de iluminação. O primeiro consiste de um ambiente sob luz artificial enquanto o segundo consiste de um ambiente sob luz solar.

Como recurso, foi utilizado o mesmo equipamento utilizado na etapa de desenvolvimento do sistema de acordo com a metodologia descrita na Seção 6.

Para os sinais referentes aos dígitos de 2 a 4, foram feitos casos de testes utilizando três diferentes distâncias entre os dedos (pequena, média e grande) e para os sinais referentes aos dígitos 0 e 1, foram utilizadas três posições onde o sinal foi emitido (esquerda, centro e direita). O objetivo aqui foi, portanto, testar até que ponto o local de emissão do sinal e a iluminação poderiam influenciar na detecção dos gestos.

Os casos escolhidos para os sinais referentes aos dígitos 0 e 1 podem ser vistos na Figura 39 e para os referentes aos dígitos de 2 a 4 podem ser vistos na Figura 40.

DIGITO REFERENTE AO SINAL	POSICÃO DO SINAL EMITIDO AO DISPOSITIVO KINECT	AMBIENTE DE TESTE
0	Esquerda	Luz Solar
0	Esquerda	Luz Artificial
0	Centro	Luz Solar
0	Centro	Luz Artificial
0	Direita	Luz Solar
0	Direita	Luz Artificial
1	Esquerda	Luz Solar
1	Esquerda	Luz Artificial
1	Centro	Luz Solar
1	Centro	Luz Artificial
1	Direita	Luz Solar
1	Direita	Luz Artificial

Figura 39 - Casos de testes utilizados para os sinais referentes aos dígitos 0 e 1.
 Fonte: Elaborado pelo autor.

DIGITO REFERENTE AO SINAL	DISTANCIA ENTRE OS DEDOS	AMBIENTE DE TESTE
2	Pequena	Luz Solar
2	Pequena	Luz Artificial
2	Média	Luz Solar
2	Média	Luz Artificial
2	Grande	Luz Solar
2	Grande	Luz Artificial
3	Pequena	Luz Solar
3	Pequena	Luz Artificial
3	Média	Luz Solar
3	Média	Luz Artificial
3	Grande	Luz Solar
3	Grande	Luz Artificial
4	Pequena	Luz Solar
4	Pequena	Luz Artificial
4	Média	Luz Solar
4	Média	Luz Artificial
4	Grande	Luz Solar
4	Grande	Luz Artificial

Figura 40 - Casos de testes utilizados para os sinais referentes aos dígitos de 2 a 4.
 Fonte: Elaborado pelo autor.

Para cada caso de teste foi feita a captura do sinal emitido e aplicado os algoritmos de suavização, subtração de fundo, segmentação, binarização e identificação (conforme descrito na seção 7). Dois dos testes realizados podem ser visualizados nas figuras 41 e 42.

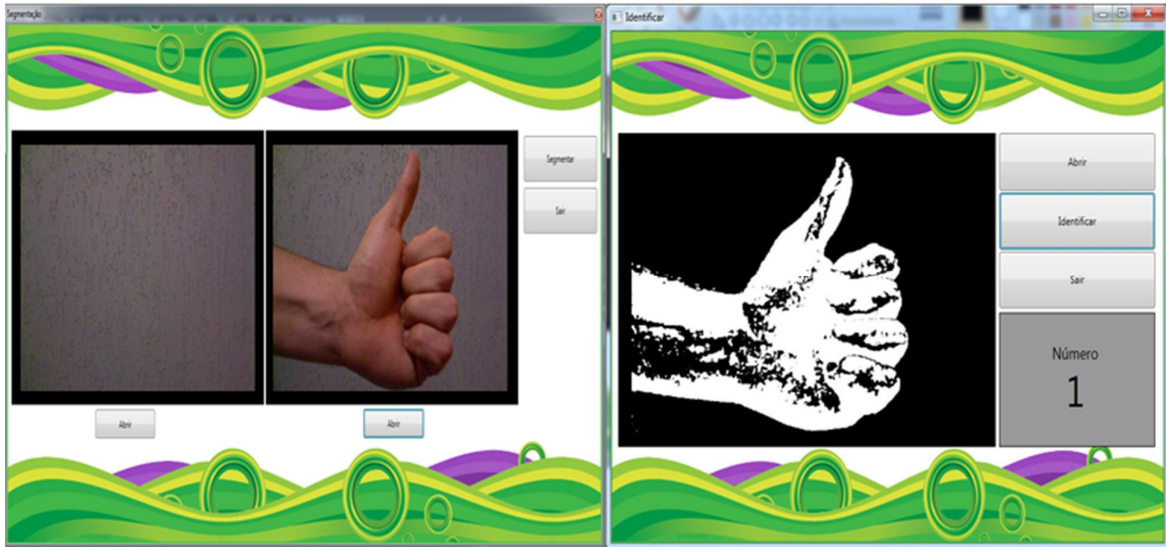


Figura 41 - Teste com o sinal de número um em um ambiente sob luz artificial.
 Fonte: Elaborada pelo autor.

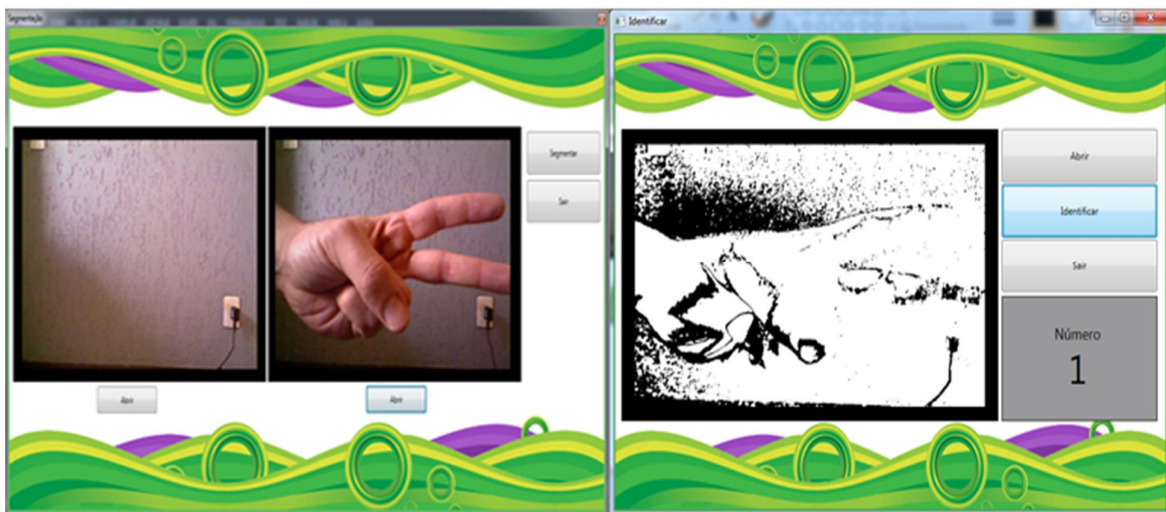


Figura 42 - Teste com o sinal de número dois em um ambiente sob luz solar.
 Fonte: Elaborada pelo autor.

A seguir são apresentados os gráficos de colunas com os números de erros e acertos de cada sinal para cada ambiente de teste. Os gráfico podem ser visualizados nas figuras 43 e 44.

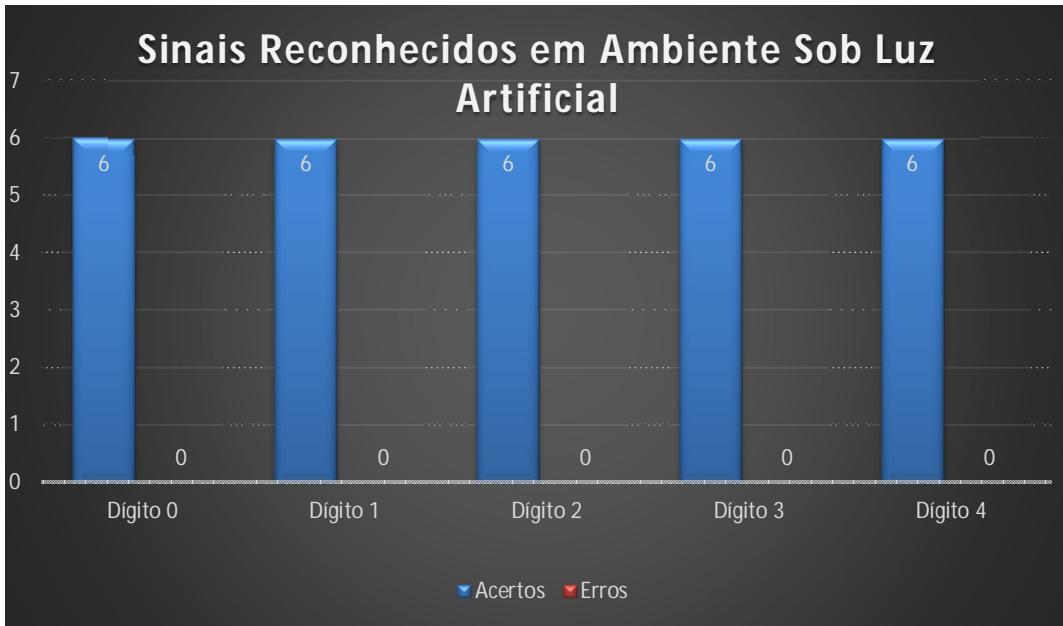


Figura 43 - Modelo 1 (Sinais emitidos em ambiente sob luz artificial).
Fonte: Elaborada pelo autor.

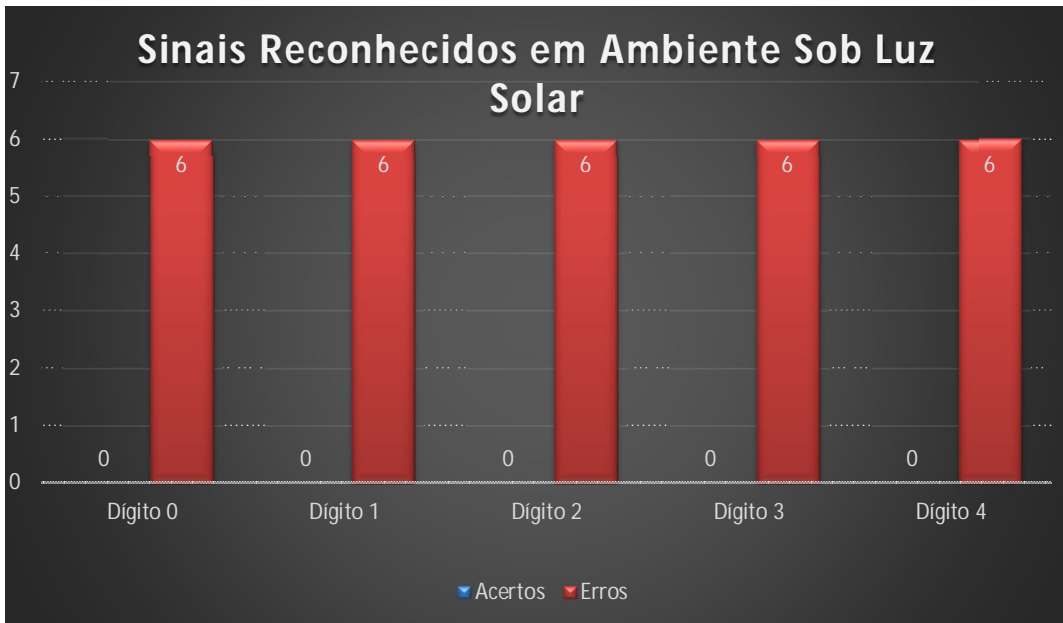


Figura 44 - Modelo 2 (Sinais emitidos em ambiente sob luz solar).
Fonte: Elaborada pelo autor.

Conforme dados da figura 43 e 44, existe uma grande influência da iluminação nos resultados obtidos mediante o processamento dos sinais. Por ser um sensor voltado para uso em ambientes internos (indoor), o Kinect sofre uma forte interferência quando exposto diretamente a luz solar. Isso ocorre porque a luz solar é uma enorme fonte de raios infravermelhos, o que afeta diretamente o funcionamento do hardware. Deste

modo, o valor de 100% de erros causados na etapa de medição (sob a luz solar), registrados no gráfico da figura 44, são relacionados à condição de luz e não à geometria do objeto (dedos). A luz solar pode interferir ou até impedir a medição infravermelha, gerando ruídos ou falha completa na medição, interferindo no processamento dos pixels da imagens capturadas.

A grande diferença entre os resultados obtidos nos dois ambientes se deve a limitação da subtração de fundo através da subtração de duas imagens, pois, uma subtração de duas imagens necessita que ambas possuam o mesmo fundo e isso não é possível em um ambiente sob luz solar devido a variação de luminosidade.

O fato de que a luminosidade nas imagens obtidas em um ambiente sob luz solar pode variar entre uma imagem e outra, provoca diferença nos valores dos pixels entre a primeira e segunda imagem. Isso não ocorre nas imagens obtidas em um ambiente sob luz artificial devido ao fato de que não ocorre variação de luminosidade.

Por conta dessa influência da luz, para verificar uma possível influência dos demais fatores (tipo e posição do sinal emitido e distância entre os dedos), foram considerados apenas os casos de teste (figura 45 e figura 46) sob a iluminação artificial.

DIGITO REFERENTE AO SINAL	POSIÇÃO DO SINAL EMITIDO AO DISPOSITIVO KINECT	AMBIENTE DE TESTE
0	Centro	Luz Artificial
1	Centro	Luz Artificial
0	Direita	Luz Artificial
1	Direita	Luz Artificial
0	Esquerda	Luz Artificial
1	Esquerda	Luz Artificial

Figura 45 - Casos de testes utilizados para os sinais referentes aos dígitos 0 e 1 (apenas sob iluminação artificial)

Fonte: Elaborado pelo autor.

DIGITO REFERENTE AO SINAL	DISTANCIA ENTRE OS DEDOS	AMBIENTE DE TESTE
2	Pequena	Luz Artificial
3	Pequena	Luz Artificial
4	Pequena	Luz Artificial
2	Média	Luz Artificial
3	Média	Luz Artificial
4	Média	Luz Artificial
2	Grande	Luz Artificial
3	Grande	Luz Artificial
4	Grande	Luz Artificial

Figura 46 - Casos de testes utilizados para os sinais referentes aos dígitos de 2 a 4 (apenas sob iluminação artificial).
Fonte: Elaborado pelo autor.

Conforme pode ser visto na figura 47, todos os sinais foram reconhecidos, independente da sua posição (esquerda, centro, direita) com relação ao dispositivo Kinect.

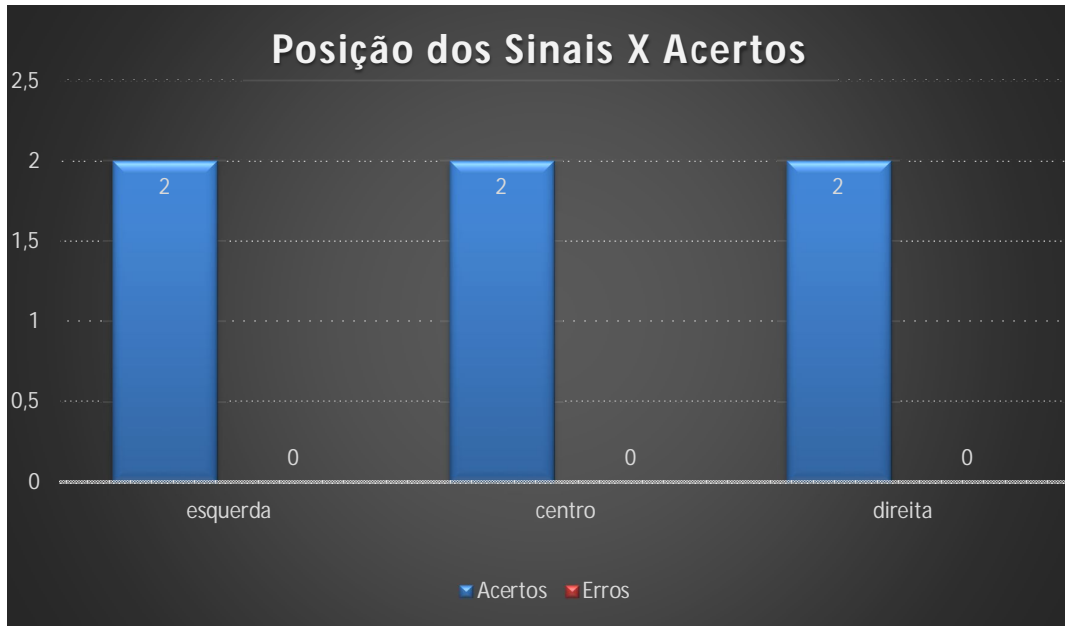


Figura 47 - Análise referente à posição do sinal emitido (apenas casos sob luz artificial).
Fonte: Elaborada pelo autor.

Do mesmo modo, conforme figura 48, ao analisar os resultados apenas sob o aspecto da distância entre os dedos, nota-se que esse fator não apresentou influência no processo de reconhecimento dos gestos.

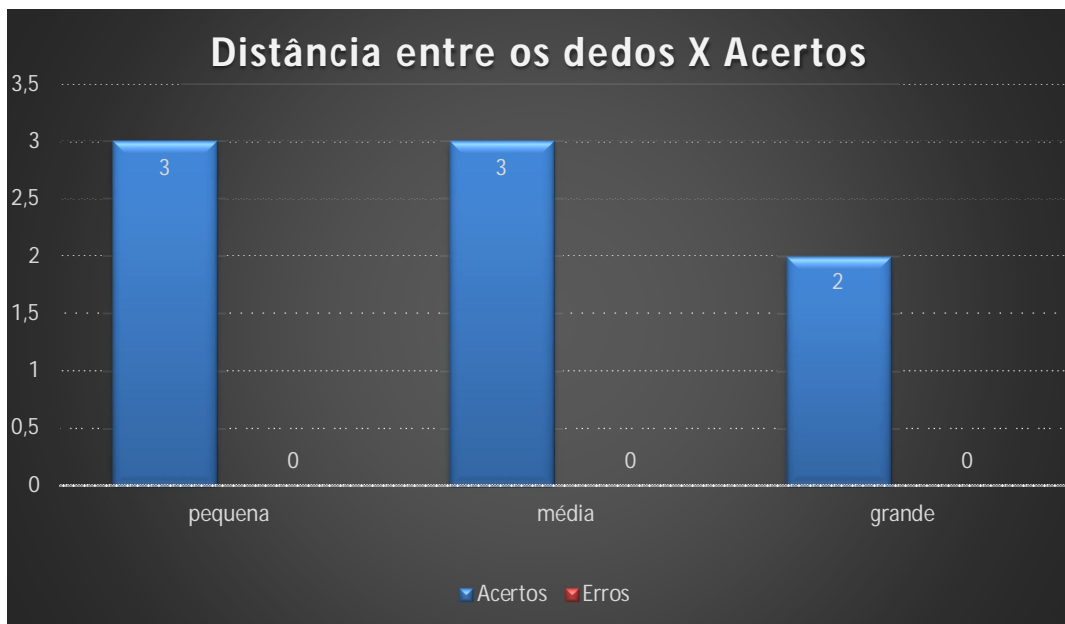


Figura 48 - Análise referente à distância entre os dedos (apenas casos sob luz artificial).
Fonte: Elaborada pelo autor.

9 – CONCLUSÕES

Este trabalho apresentou a implementação de um software que identifica o número referente ao sinal do alfabeto de números da Linguagem Brasileira de Letras (LIBRAS), utilizando o dispositivo Kinect para captura de imagens, de modo que pudessem ser segmentadas, validando o objetivo principal deste trabalho.

Ao longo do trabalho, verificou-se que o único critério para o reconhecimento de cada sinal foi encontrar a quantidade de dedos que estavam levantados na mesma posição de linhas imaginárias traçadas pelo algoritmo de identificação. Verificou-se que o algoritmo de subtração de fundo através da subtração de imagens necessita de imagens com o fundo idêntico, com isso, tornando o software ineficiente em ambientes sob luz solar, onde há muita variação de luminosidade, causando diferença entre os valores dos pixels das imagens.

Como trabalhos futuros, sugere-se analisar um novo algoritmo de subtração de fundo, que possa ser eficaz tanto para ambientes sob luz artificial quanto para ambientes sob luz solar.

Para avaliar ainda mais o sistema, sugere-se testar o sistema com uma base de dados maior, ou seja, com um número maior de mãos.

Pode-se também buscar um novo método de identificação, podendo assim, identificar os demais números que compõe o alfabeto de números de LIBRAS e também identificar o alfabeto completo de letras de LIBRAS.

Poderão ser analisados também, métodos para o reconhecimento do sinal emitido ao dispositivo Kinect em tempo real.

REFERÊNCIAS

- ANJO, M. S. Reconhecimento de Gestos Estáticos de Mãos Utilizando a Rede Neural Neocognitron e a Multi-Layer Perceptron. Projeto de iniciação científica, Orientador: Prof. Dr. Ednaldo Brigante Pizzolato, Co-Orientador: José Hiroki Saito. Fundação de Amparo a Pesquisa do Estado de São Paulo, FAPESP, 2009.
- BAINBRIDGE-SMITH, A.; LANE, R. G. Determining Optical Flow Using a Differential Method. *Image and Vision Computing*, vol. 15, pp. 11-22, 1997.
- BEUCHER, S.; MEYER, F. The morphological approach to segmentation: the watershed transformation. In *Mathematical Morphology in Image Processing*, (Ed. E.R. Dougherty), p. 433-481, 1993.
- BIRCK, Fernando. Fergonez Blog, 2007. Guia Prático para Iniciantes – Microsoft® XNA. Disponível em: < <http://www.fergonez.net> >. Acesso em: 17 abr. 2013.
- BRITO, D. R. Sistema de Monitoração e Gerenciamento do Consumo Elétrico Residencial Utilizando a Malha Elétrica para Comunicação, Orientador Prof. Amarildo Geraldo Reichel. Universidade Positivo, UP, 2009.
- CANNY, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8 , Issue 6, Pages 679 – 698, 1986.
- CASTRO, Alberto Rainha de; CARVALHO, Ilza Silva de. **Comunicação por língua brasileira de sinais**. 3. ed. Brasília: Senac, 2009, 272 p.
- CHANG, F.; CHEN, C.; LU, C. A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique. *ACM Journal Computer Vision and Image Understanding*, Vol. 93, Issue 2, Pages 206 - 220, 2004.
- CORREIA, M. M. Reconhecimento de Elementos Gestuais com Kinect, Orientador: Eurico Manuel Elias de Moraes Carrapatoso, Co-Orientador: António Abel Vieira de Castro. Faculdade de Engenharia da Universidade do Porto, FEUP, 2013.
- COUTO, Alpia Ferreira, COSTA, Arnaldo Marcio et al. Como compreender o deficiente auditivo. Rio de Janeiro: Expansão Editorial, 1985, 85 p.
- CUCCHIARA, R.; GRANA, C.; PICCARDI, M.; PRATI, A. Detecting Moving Objects, Ghosts, and Shadows in Video Streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No.10, 2003.
- DUQUE, D.; SANTOS, H.; CORTEZ, P. Moving Object Detection Unaffected by Cast Shadows, Highlights and Ghosts. *IEEE International Conference on Image Processing (ICIP)*, 2005.

EEBLMLIBRAS . **Vejo Vozes Tenho o Mundo em Minhas Mãos**, 2011. Disponível em: <<http://eeblmlibras.blogspot.com.br/2011/05/jogo-somando-em-libras.html>>. Acesso em: 28 mar. 2013.

G1. **23,9% dos brasileiros declaram ter alguma deficiência, diz IBGE. Portal G1 Brasil**, disponível em: <<http://g1.globo.com/brasil/noticia/2012/04/239-dos-brasileiros-declaram-ter-alguma-deficiencia-diz-ibge.html>>. Acesso em: 17 de maio de 2013.

GRANDE ENCICLOPÉDIA LAROUSSE CULTURAL, Livro 22, 1995, página 5.544.

GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing, 3ª Edição, Agosto 2007.

GRADY, L. Random Walks for Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1768-1783, Vol. 28, No. 11, 2006.

HAYKIN, S. Neural Networks: A Comprehensive Foundation. Prentice Hall, 2º ed., 1998.

JOHNSON, J. L.; PADGETT, M. L. PCNN models and applications. IEEE Transactions on Neural Networks, v. 10-3 p. 480-498, 1999.

KOHONEN, T. Self-Organizing Maps. Third extended edition, Springer, 2001.

KOJIMA, C. K. , Libras Lingua Brasileira De Sinais A Imagem Do Pensamento, Vol. 3, 2012, 113 p.

LINDEBERG, T. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. International Journal of Computer Vision, v. 11(3), p. 283-318, 1993.

LLOYD, S. P. Least square quantization in PCM. IEEE Transactions on Information Theory, Vol. 28, Pages 129-137, 1982.

MCHUGH, Sean. Understanding White-Balance. Disponível em: <<http://www.cambridgeincolour.com/tutorials/white-balance.htm>>. Acessado em: 17/05/2013.

MESTER, R.; FRANKE, U. Statistical Model Based Image Segmentation Using Region Growing, Countour Relaxation and Classification. Cambridge Symposium on Visual Communications and Image Processing, p. 616-624, November, 1988.

MICROSOFT. **SQL Server 2008**, 2008. Disponível em: <<http://www.microsoft.com/brasil/servidores/sql/prodinfo/default.aspx>>. Acesso em: 28 mar. 2013.

MOBAHI, H.; RAO, S.; YANG, A.; SASTRY, S.; MA, Y. Segmentation of Natural Images by Texture and Boundary Compression. Artificial Intelligence and Information Systems (AIIS) Seminar, 2010.

MSDN. **Microsoft: Kinect for Windows Sensor**, c2013. Disponível em: <<http://msdn.microsoft.com/en-us/library/hh855355.aspx>>. Acesso em: 17 abr. 2013.

- MSDN. **Visual C# Developer Center**, 2008. Disponível em: <[http://msdn.microsoft.com/pt-pt/vcsharp/default\(en-us\).aspx](http://msdn.microsoft.com/pt-pt/vcsharp/default(en-us).aspx)>. Acesso em: 10 abr. 2013.
- OSHER, S.; PARAGIOS, N. *Geometric Level Set Methods in Imaging Vision and Graphics*. Springer Verlag, 2003.
- PARKS, D. H.; FELLS, S. S. Evaluation of Background Subtraction Algorithms with Post-processing. *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, 2008.
- PICCARDI, M. Background subtraction techniques: a review. *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, p. 3099–3104, 2004.
- PRESSMAN, Roger S. **Engenharia de software**. 6. ed. São Paulo: McGraw-Hill, 2006, 720 p.
- ROCHA, P. R.; DEFAVARI, A. H.; BRANDÃO, P. S. Estudo da viabilidade da utilização do Kinect como ferramenta no atendimento fisioterapêutico de pacientes neurológicos, SBGames, 2012, Disponível em: <<http://sbgames.org/sbgames2012/proceedings/papers/gamesforchange/g4c-04.pdf>>. Acesso em: 28 mar. 2013.
- SANO, NETO. *Software livre e negócios com software: um estudo na perspectiva da propriedade intelectual*. 2008. p. 44. Trabalho de conclusão de curso – Universidade Federal Fluminense, Niterói.
- SANTOS, M. F. C. dos; LIMA, M. C. M. P.; ROSSI, T. R. de F. Surdez: diagnóstico audiológico. In: SILVA, I. R.; KAUCHAKJE, S.; GESUELI, Z. M. (Org.) **Cidadania, surdez e linguagem: desafios e realidades**. São Paulo: Plexus, 2003. p. 17-40.
- SILVA, Ivani Rodrigues; KAUCHAKJE, Samira; GESUELI, Zilda Maria. **Cidadania, surdez e linguagem: desafios e realidade**. São Paulo: Plexus Editora, 2003, 248 p.
- SHAPIRO, L. G.; STOCKMAN, G. C. *Computer Vision*. Prentice-Hal, New Jersey, pp 279-325, 2010.
- SOUSA, Eloysia Godinho de. **Surdez e significado social**. São Paulo: Cortez, 1982, 96 p.
- STAUFFER, C.; GRIMSON, W. Adaptive background mixture models for real time tracking. *IEEE Computer Vision and Pattern Recognition*, 1999.
- WREN, C.; AZARHAYEJANI, A.; DARRELL, T.; PENTLAND, A. P. Pfinder: real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, Pages 78-85, 1997.

APÊNDICE A: MANUAL DO SISTEMA

O presente tutorial tem como objetivo ensinar o usuário as funcionalidade e detalhes do sistema.

Ao executar o sistema, a primeira tela a ser exibida é a tela inicial (Figura A.1), com a tela inicial já carregada, basta dar dois cliques com o botão esquerdo do mouse no botão “Iniciar” para começar a utilizar o sistema (Figura A.2).

Caso queira fechar o sistema, basta dar dois cliques com o botão esquerdo do mouse no botão “Fechar” (Figura A.3).



Figura A. 1 - Tela inicial do sistema.
Fonte: Elaborada pelo autor.

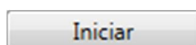


Figura A. 2 - Botão "iniciar" da tela inicial.
Fonte: Elaborada pelo autor.



Figura A. 3 - Botão "iniciar" da tela inicial.
Fonte: Elaborada pelo autor.

Pressionando o botão “Iniciar”, será exibida a tela do menu do sistema (Figura A.4). No menu é possível escolher a ação que será executada pelo sistema, para efetuar a escolha, basta clicar duas vezes com o botão esquerdo do mouse no botão referente à ação desejada. As ações disponíveis no menu principal são acionadas pelos botões “Capturar” (Figura A.5), “Segmentar” (Figura A.6), “Identificar” (Figura A.7) e “Sair” (Figura A.8).

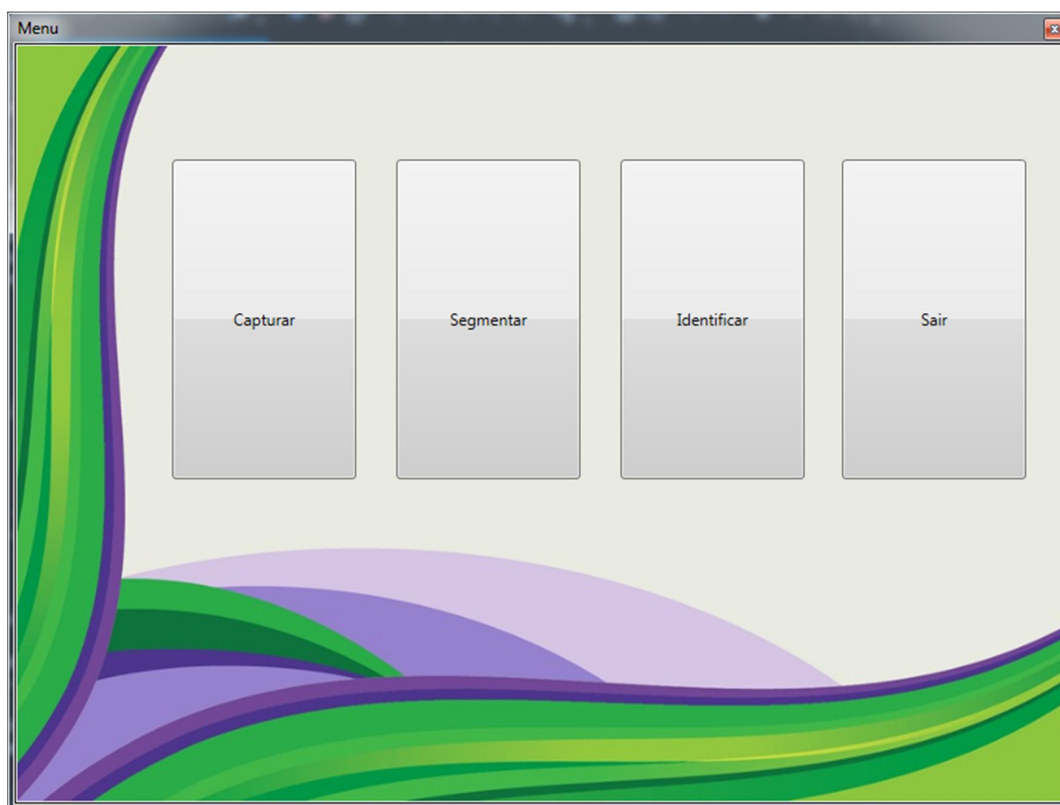


Figura A. 4 - Tela de menu do sistema.
Fonte: Elaborada pelo autor.

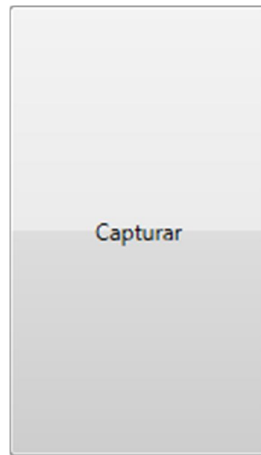


Figura A. 5 - Botão "Capturar" da tela de menu.
Fonte: Elaborada pelo autor.

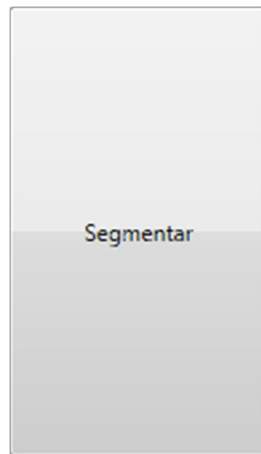


Figura A. 6 - Botão "Segmentar" da tela de menu.
Fonte: Elaborada pelo autor.

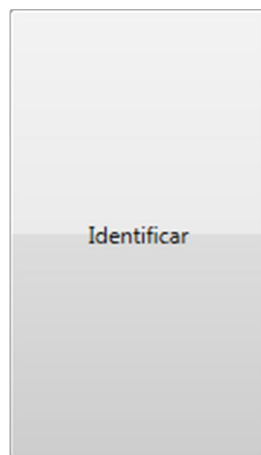


Figura A. 7 - Botão "Identificar" da tela de menu.
Fonte: Elaborada pelo autor.

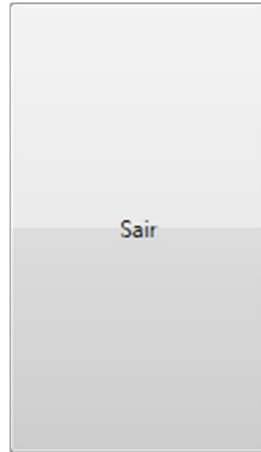


Figura A. 8 - Botão "Sair" da tela de menu.
Fonte: Elaborada pelo autor.

Pressionando o botão “Capturar” da tela de menu, será acionada a tela de captura de imagem (Figura A.9), nela é possível exibir a imagem do dispositivo Kinect na tela e capturar imagens. Para ligar o dispositivo, basta pressionar o botão “Ligar” (Figura A.10) e para capturar a imagem pressione o botão “Capturar” (Figura A.11).

Com o dispositivo ligado, o botão “Ligar” passa a se chamar “Desligar” (Figura A.12), com ele, é possível desligar o dispositivo e interromper a imagem do dispositivo que está sendo exibida na tela.

Caso deseje sair da tela de captura e retornar a tela de menu, basta pressionar o botão “Sair” (Figura A.13).

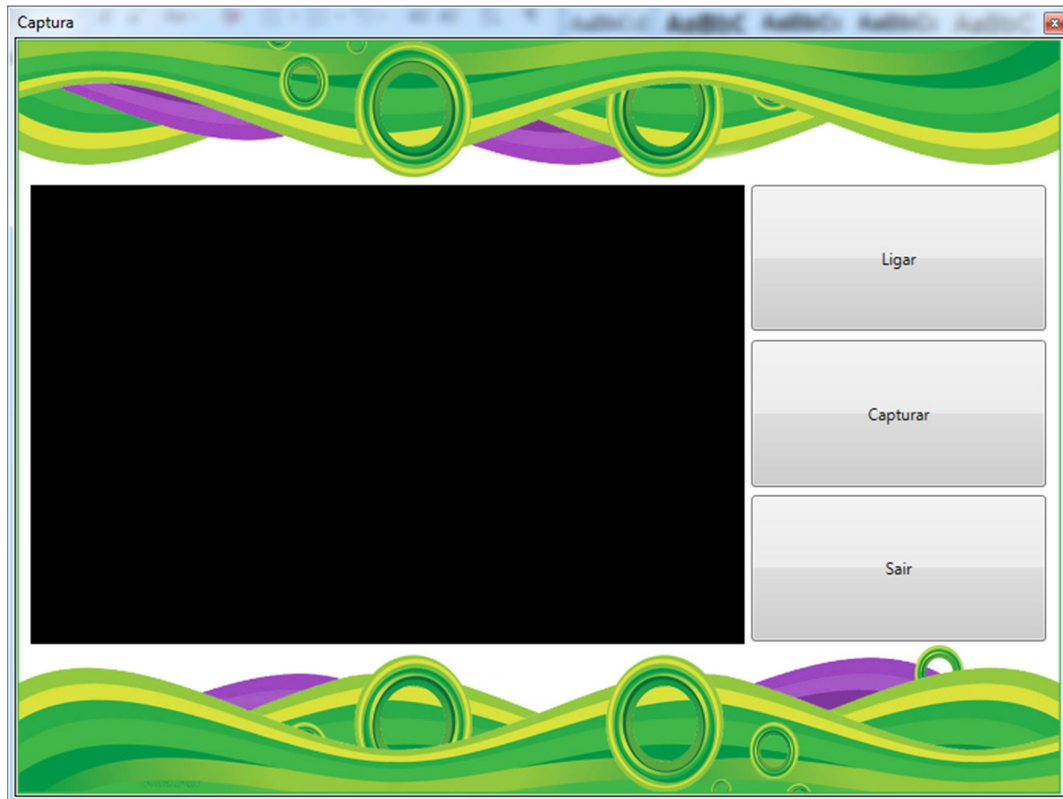


Figura A. 9 - Tela de captura de imagem do sistema.
Fonte: Elaborada pelo autor.

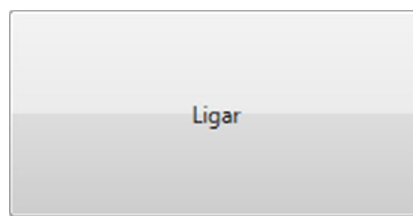


Figura A. 10 - Botão "Ligar" da tela de captura.
Fonte: Elaborada pelo autor.

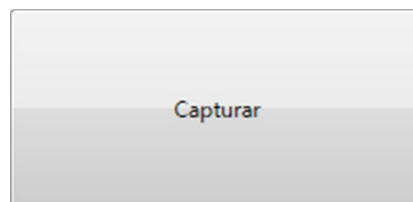


Figura A. 11 - Botão "Capturar" da tela de captura.
Fonte: Elaborada pelo autor.



Figura A. 12 - Botão "Desligar" da tela de captura.
Fonte: Elaborada pelo autor.

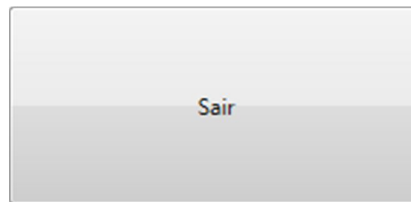


Figura A. 13 - Botão "Sair" da tela de captura.
Fonte: Elaborada pelo autor.

Pressionando o botão “Segmentar” da tela de menu, a tela de segmentação será acionada (Figura A.14). Com a tela de segmentação é possível carregar as imagens na tela, e aplicar a subtração de fundo e a segmentação nas imagens.

Para carregar as imagens, basta pressionar o botão “Abrir” (Figura A.15) da tela de captura, o procedimento é o mesmo para carregar as duas imagens. Feito isso, basta pressionar o botão “Segmentar” (Figura A.16) para aplicar a subtração de fundo, segmentação nas imagens e salvar a imagem final.

Caso deseje retornar à tela de menu, basta pressionar o botão “Sair” da tela de segmentação (Figura A.17).

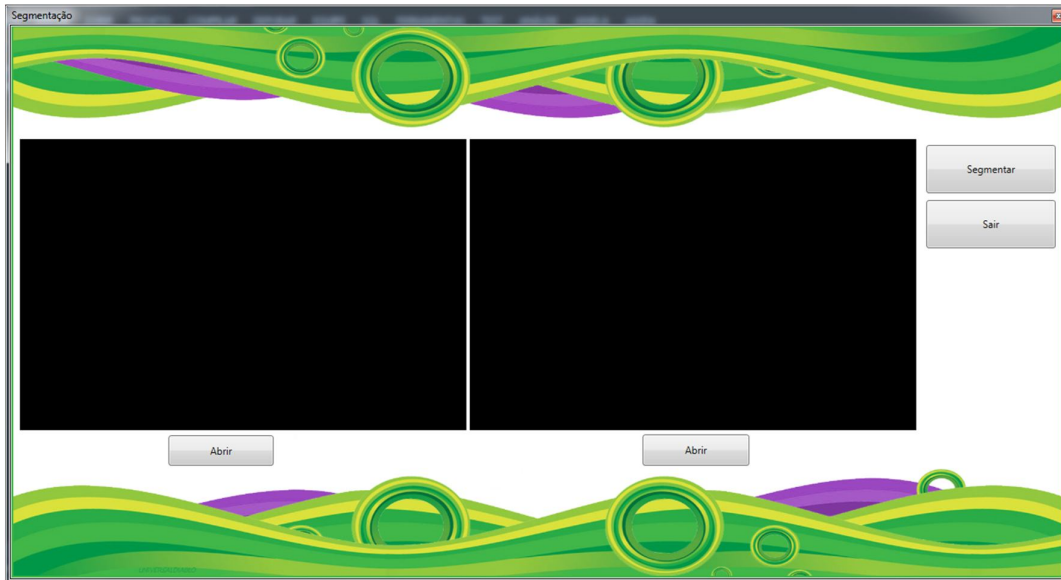


Figura A. 14 - Tela de segmentação do sistema.
Fonte: Elaborada pelo autor.



Figura A. 15 - Botão "Abrir" da tela de segmentação.
Fonte: Elaborada pelo autor.

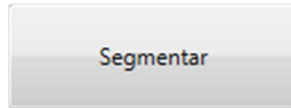


Figura A. 16 - Botão "Segmentar" da tela de segmentação.
Fonte: Elaborada pelo autor.

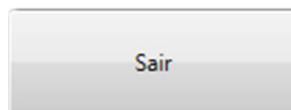


Figura A. 17 - Botão "Sair" da tela de segmentação.
Fonte: Elaborada pelo autor.

Pressionando o botão “Identificar” da tela de menu, a tela de identificação será acionada (Figura A.18). Na tela de identificação é possível carregar a imagem final, identificar qual o número que representa o sinal emitido e retornar a tela de menu.

Para carregar a imagem final, basta pressionar o botão “Abrir” (Figura A.19) da tela de identificação, feito isso, basta pressionar o botão “Identificar” (Figura A.20) da tela de identificação para exibir na tela o número que representa o sinal emitido.

Caso deseje voltar à tela de menu, deve ser pressionado o botão “Sair” da tela de identificação (Figura A.21).

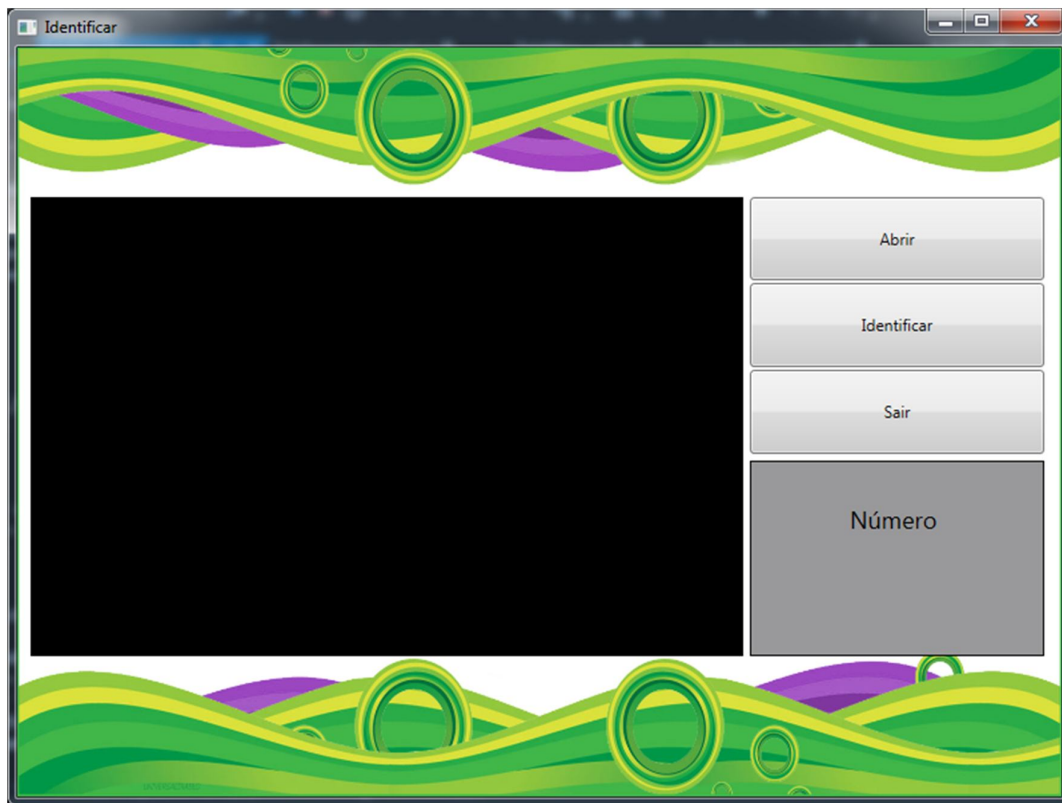


Figura A. 18 - Tela de identificação do sistema.
Fonte: Elaborada pelo autor.



Figura A. 19 - Botão "Abrir" da tela de identificação.
Fonte: Elaborada pelo autor.

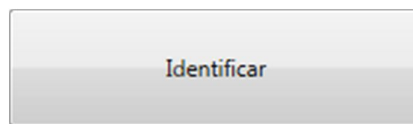


Figura A. 20 - Botão "Identificar" da tela de identificação.
Fonte: Elaborada pelo autor.

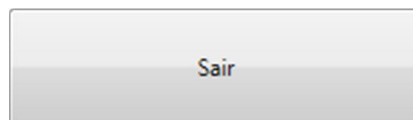


Figura A. 21 - Botão "Sair" da tela de identificação.
Fonte: Elaborada pelo autor.

Caso deseje sair da tela de menu e retornar a tela inicial do sistema, basta pressionar o botão “Sair” da tela de menu do sistema.

DETECÇÃO E SEGMENTAÇÃO DE ELEMENTOS GESTUAIS DE LIBRAS COM O USO DO MICROSOFT KINECT

Murilo de O. Souza, Patrick P. Silva, Elvio G. Silva, Patricia B. Ribeiro

Instituto de Informática – Universidade Do Sagrado Coração (USC) Bauru – SP

Centro de Ciências Exatas e Sociais Aplicadas– Universidade do Sagrado Coração,
USC.

moslite@hotmail.com, patrickpsilva@gmail.com, egsilva@usc.br,
patriciabellin@yahoo.com.br

Abstract. *This article presents will shows how to develop an application which is responsible for the module of segmentation of that type of system, to auxiliate people with hearing impairment by being capable of recognizing different and restricted gestures from LIBRAS (Brazilian Signal Language), especially from numbers 0 to 4, using the Microsoft Kinect technology. In order to evaluate the system, we tested five signs (gestures) in two test environments (artificial light and natural light), evaluating system performance in six test cases for each signal. The results showed that the system is able to recognize gestures made, however, there is a direct influence of the ambient light in the recognition process.*

Resumo. *Este artigo mostra como desenvolver uma aplicação que seja responsável pelo módulo de segmentação de um sistema de reconhecimento de gestos para auxílio aos portadores de deficiência auditiva que seja capaz de reconhecer gestos referentes a um conjunto restrito da LIBRAS, especificamente os números de 0 a 4, utilizando a tecnologia Microsoft Kinect.*

Com o objetivo de avaliar o sistema, foram testados cinco sinais (gestos) em dois ambientes de teste (luz artificial e luz natural), avaliando-se o desempenho do sistema em seis casos de testes para cada sinal. Os resultados mostraram que o sistema é capaz de reconhecer os gestos realizados, entretanto, existe uma influência direta da iluminação do ambiente, no processo de reconhecimento.

1. INTRODUÇÃO

A surdez é a perda ou o enfraquecimento do sentido da audição, há diferentes graus de perda de audição, uma pessoa pode ouvir alguns ruídos, outras ter apenas dificuldade em compreender algumas palavras, e também, pode não ouvir nada, esta diferença varia de pessoa para pessoa. A deficiência pode ser ocasionada por diversas maneiras, uma pessoa pode nascer surda, adquirir por contração de doenças que afetam o aparelho auditivo, ou então, sofrer alguma lesão que danifique o ouvido, como um barulho muito forte, ou exposição a som alto por um longo período de tempo. Cerca de 9,7 milhões declaram ter deficiência auditiva (5,1%). A deficiência auditiva severa foi declarada por mais de 2,1 milhões de pessoas. Destas, 344,2 mil são surdas e 1,7 milhão de pessoas têm grande dificuldade de ouvir (G1, 2013).

Considerando que a principal forma de comunicação dos deficientes auditivos se dá por meio de sinais, a hipótese é que um software projetado de forma a privilegiar esta maneira de interação, pode melhorar o processo de interação entre estes indivíduos e os computadores. Diversas tecnologias podem auxiliar quanto a isso como, por exemplo o Kinect. O dispositivo Microsoft Kinect consiste em um aparelho capaz de reconhecer movimentos. Ele, a partir de suas câmeras com sensores de movimento, é capaz de captar e interpretar movimentos das mãos e rosto das pessoas, podendo transformá-los em texto e até em voz, sendo muito útil em situações que necessitem deste tipo de interação.

Sendo assim, diante da complexidade de implementação de um sistema completo de reconhecimento de gestos, o objetivo desta investigação é desenvolver uma aplicação que seja responsável pelo módulo de segmentação do sistema, utilizando como base o Microsoft Kinect para captar e segmentar as imagens referentes a um subconjunto de gestos da LIBRAS.

2. DEFICIÊNCIA AUDITIVA

“Existem muitas crianças que podem ter nascido com a deficiência auditiva ou ter perdido a audição após o nascimento. Quando isso acontece, ela deve ter muitas dificuldades para adquirir a linguagem oral”. (COUTO, 1985, p. 11). Isso ocorre, pelo fato da criança nunca ter entrado em contato com a comunicação para conseguir

entende-la. Quando uma pessoa já possui a comunicação e perde a audição, é muito mais fácil mantê-la.

Segundo Godinho (1982), um indivíduo por nas com deficiência auditiva (congenita) ou adquirir por degeneração, por meio de doenças. A congênita pode ser endógena, herança genética, ou então, exógena, quando a mãe contrai alguma doença durante o período de gestação, que afeta o aparelho auditivo do feto. Problemas de incompatibilidade sanguínea e partos prematuros também podem afetar o sentido de audição. Porém, ainda podem ocorrer problemas de malformação depois que a criança nasce, durante o desenvolvimento, ocasionando a atrofia ou inexistência do aparelho auditivo. Ainda na infância, a criança corre o risco de surdez por contração de doenças como: meningite, sinusite, furúnculos e diversos antibióticos que possam afetar o canal de audição. Já na fase adulta, a perda auditiva esta liga a poluição sonora, gerando traumas por grande exposição sonora.

Funcionalmente, entretanto, os sentidos atuam reciprocamente, mediante um processo de percepção Inter sensorial conhecido por sinestesia, em razão do qual o conteúdo percebido é o resultado da elaboração mental das sensações elementares provenientes dos diversos sentidos, que transmitidas pela via neural própria difundem-se pela área da associação do córtex cerebral e são interpretadas como experiência sensorial integrada pelo conjunto de experiência sensitiva. (SOUSA, 1982, p. 18).

Por isso, quanto antes uma criança for diagnosticada com a deficiência, melhor e mais rápido será o seu tratamento, podendo trabalhar corretamente a linguagem e o desempenho cognitivo, social e emocional. “O diagnóstico audiológico é de extrema importância uma vez que norteia os procedimentos de habilitação e reabilitação do indivíduo surdo”. (SANTOS; LIMA; ROSSI, 2003, p. 37).

2.1 SURDEZ E A SOCIEDADE

Qualquer pessoa pode ter deficiência auditiva, já que esta não escolhe entre classe social, cor, etnia, etc. Ainda há preconceito, principalmente, pela falta de comunicação, já que a surdez está ligada a falta de linguagem, colocando essa pessoa em uma posição diferente da habitual para as outras, dificultando assim o processo de socialização do indivíduo. Portanto, é importante desenvolver meios que busquem a quebra de barreiras para adaptar diferentes modos de se comunicar para trabalhar com deficientes auditivos, pois a comunicação está altamente ligada ao processo de socialização, que traz conhecimentos e valores culturais. “Os problemas da linguagem irão dificultar a comunicação da criança com a família, sua vida em sociedade, sua aprendizagem na escola. Se a criança não for amada, compreendida e ajudada em suas dificuldades, poderá, ainda, ter problemas emocionais”. (COUTO, 1985, p. 11).

“Na ausência dos estímulos sonoros, a criança vai aprendendo a transferir sua atenção para os estímulos visuais e reagirá de acordo com a expressão do rosto da mãe e os contatos com o seu corpo” (COUTO, 1985, p. 31).

É necessário criar meios para se estabelecer uma comunicação decente para todos, para que não ocorra exclusão de um cidadão. A comunicação está presente em todos os momentos da vida de uma pessoa, quando algo é afetado e parte do comum se perde, é importante criar outro tipo de técnica para que o sujeito não seja prejudicado pela deficiência. “A inclusão social tem como meta, basicamente, criar uma sociedade capaz de acolher todas as pessoas, independente das diferenças e necessidades individuais e, para isso, preconiza soluções para as diferentes identidades existentes nos ambientes sociais” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 235).

2.2 DEFICIÊNCIA AUDITIVA E EDUCAÇÃO

Os surdos sofreram ao longo dos anos por não conseguirem se comunicar, muitos foram excluídos como cidadãos e tiveram seus direitos roubados. A comunicação correta possibilitou um modo de aprendizagem e comunicação entre as pessoas com esse tipo de deficiência. O modo de comunicação trouxe, além dos direitos, a esperança de aprendizado e autoestima. Hoje em dia, ainda há preconceito, mas muitas barreiras já estão sendo quebradas e os surdos vêm conquistando seu espaço a cada dia, mostrando ser capazes de aprender uma nova língua, a conseguirem fazer uma faculdade, estabelecer relações afetivas e matrimoniais e trabalhar, como qualquer pessoa. “A força da língua brasileira de sinais, suas ricas características e componentes fazem dela um poderoso instrumento linguístico que permite o indivíduo surdo ser amplamente beneficiado com todo o amplo conhecimento humano, inclusive a aquisição de uma segunda língua, mesmo sendo ela oral ou escrita” (CARVALHO; CASTRO, 2009, p. 15).

De acordo com Rocha, informações retiradas do site do Instituto Nacional de Educação de Surdos (INES), esta foi a primeira instituição especializada em educar os deficientes auditivos no Brasil, por isso, pessoas do País inteiro e também de outros países vinham para o Rio de Janeiro para ter uma educação de qualidade, se profissionalizar e se socializar. “A partir desse instituto, deu-se o início à criação da Língua Brasileira de Sinais (LIBRAS), mistura da Língua de Sinais Francesa com os sistemas de comunicação já usados pelos surdos das mais diversas localidades”. (CARVALHO; CASTRO, 2009, p. 16).

O INES foi uma grande conquista dos surdos no Brasil, mostrando a capacidade de aprendizado e que é possível a comunicação para todos. De acordo com informações do próprio site do INES, hoje, “além de oferecer no seu Colégio de Aplicação, Educação Precoce (de zero a três anos), Ensino Fundamental e Médio, oferece também Ensino Superior através do Curso Bilíngue de Pedagogia, experiência pioneira na América Latina”. O Instituto busca, além de educar e preparar a pessoa portadora de deficiência auditiva, socializar e trazer caminhos para um ensino de qualidade visando a profissionalização. A educação é um direito de todos, assim como a comunicação, por isso o objetivo de toda essa batalha dos surdos é oferecer uma educação de qualidade e também direcionar esses cidadãos para a área profissionalizante, garantindo um futuro sólido.

Seu principal foco, como órgão do Ministério da Educação (MEC), é a produção, o desenvolvimento e a divulgação de conhecimentos científicos na área da surdez em todo o território nacional, bem como subsidiar a Política Nacional da Educação, na perspectiva de promover e assegurar o desenvolvimento global da pessoa surda, sua plena socialização e o respeito às suas diferenças. (CARVALHO; CASTRO, 2009, p. 17).

A Língua Brasileira de Sinais (LIBRAS) é a língua dos Surdos, fundamentada pela Lei Federal nº. 10.436, de 24 de abril de 2002 e regulamentada pelo decreto 5.626, de 22 de dezembro de 2005.

2.3 SURDEZ E TECNOLOGIAS

Segundo Silva, Kauchakje e Gesueli (2003, p. 196), o avanço tecnológico teve grande contribuição para os deficientes auditivos, principalmente na área da comunicação, com aparelhos de telefone, relógios, um dispositivo que possibilita aos surdos assistirem televisão com modo de legenda, etc. “Software de vários tipos existem hoje no mercado e/ou na internet: editores de texto, de apresentações, de jornais eletrônicos e de histórias em quadrinhos”. Esse acesso fácil e rápido nos possibilitou uma gama de modos diferentes de utilizar a comunicação, tanto à escrita quanto à leitura. “De forma análoga, a interface de um sistema computacional é o meio pelo qual pessoas e computadores se comunicam: é o elemento do ambiente que (inter) medeia o diálogo homem-computador e que, portanto, demanda um uso cognitivo da linguagem” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 197). A interface auxiliará no modo de comunicação entre a pessoa e um artefato.

Há dois pontos muito importantes que devem ser seguidos no desenvolvimento desse programa. O primeiro é o desenvolvimento do software e o segundo passo é as particularidades humanas estarem atreladas ao seu uso. “Do ponto de vista do sistema, a questão tem sido tratada com grande ênfase em metodologias e técnicas da Engenharia de software para projeto e especificação da parte do código que implementa a interface” (SILVA, KAUCHAKJE, GESUELI, 2003, p. 197).

Já em relação às características humanas, os estudos têm focalizado os aspectos ergonômicos, culturais e cognitivos envolvidos no uso de elemento do ambiente computacional, como por exemplo, a eficiência do uso do mouse, a disposição das teclas de função no teclado, etc. (SILVA, KAUCHAKJE, GESUELI, 2003, p. 197 - 198).

Estudos que também estão crescendo, na área das tecnologias de informação e comunicação voltada para os surdos, são os *SingWriting*, que usa a linguagem de sinais

para ajudar a trabalhar com os softwares. “O *SingWriting*, desenvolvido por Valerie Sutton em 1975, representa graficamente as características visuoespaciais da Língua de Sinais, até então considerada ágrafa”. (SILVA, KAUCHAKJE, GESUELI, 2003, p. 205).

3. SEGMENTAÇÃO

3.1 CONSIDERAÇÕES INICIAIS

A segmentação tem como objetivo subdividir a imagem em regiões para facilitar a interpretação da informação contida na mesma. Esta subdivisão é feita a partir do agrupamento de pixels utilizando características (*features*) comuns a este conjunto que podem ser cor, textura, intensidade ou continuidade.

Considerando as características previamente levantadas com relação aos algoritmos de segmentação bem como a adequação aos objetivos desta pesquisa, optou-se por restringir os estudos aos algoritmos por diferenças (*Background Subtraction*). A seguir será apresentada a conclusão sobre os algoritmos estudados e a solução proposta para este projeto.

3.2 SUBTRAÇÃO DE FUNDO (BACKGROUND SUBTRACTION)

Normalmente utilizado para detecção de movimento através de diferenças, este método consiste na dicotomização da imagem em pixels pertencentes ao fundo e pixels referentes aos objetos de interesse ou em movimento na cena. Esta tarefa é efetuada através da modelagem de fundo que consiste em uma modelagem estatística realizada por meio de observações de uma sequência de imagens de uma cena para tentar estimar quais regiões da imagem pertencem ao fundo. Para obter os pixels de interesse, que compõe os objetos em movimento, é feita uma simples subtração do modelo de fundo da imagem atual, seguido de um *threshold* (ANJO, 2012).

De acordo com Piccardi (2004) são vários os métodos de *Background Subtraction*, onde as principais variações estão na forma de estimação do modelo de fundo e no método de comparação deste modelo com a imagem atual. Alguns métodos citados por Piccardi, como *Kernel Density Estimation* (KDE), *Sequential KD Approximation*, *Eigen Backgrounds* entre outros, não possibilitam a implementação em tempo real.

4. METODOLOGIA

Para a etapa de desenvolvimento foram utilizados como recursos um equipamento notebook do fabricante DELL com processador Intel Core i5 2.40GHz, 4gb de memória RAM DDR3, Disco rígido de 500Gb, com *Windows Seven Ultimate* 64 bits como

Sistema Operacional, e de um dispositivo Microsoft Kinect. Os drivers de comunicação do dispositivo Kinect com o computador podem ser obtidos gratuitamente no site oficial do SDK. O módulo implementado compreende as etapas de aquisição de dados, preparação de dados e interpretação de dados.

A aquisição refere-se à programação utilizando o SDK do Kinect para capturar os dados dos sensores e armazená-los em estruturas para serem encaminhados à etapa de preparação. Na fase de aquisição dos dados é feita a captura das imagens pelos sensores do Kinect, considerando a câmera de vídeo, o sensor de profundidade e a informação do corpo humano. Os parâmetros de aquisição de imagem (resolução da profundidade e da câmera de vídeo, limitadas à 320X240 e 640X480 respectivamente). A forma de captura dos frames pelos sensores também pode ser configurada. O modelo de captura escolhido foi o *Event Model*, que consiste em transmitir quando um botão foi pressionado, executando um evento previamente configurado, para que assim o usuário do software tenha livre controle de quando a imagem será capturada. Na etapa de preparação de dados, as imagens fornecidas pelo Kinect foram formatadas com algoritmos de processamento de imagens.

Primeiramente, as imagens passaram pelo algoritmo de suavização (*Smoothing*), feito isso, foi passado o algoritmo de subtração de fundo (*Background Subtraction*). Para resolver o problema com sombras (*Shadows*), as imagens passaram por um algoritmo de tratamento de sombras (*Shadows*) e, por último, o algoritmo de binarização foi passado para deixar na cor branca os pixels onde existe movimento e na cor preta onde não existe. Mesmo com o algoritmo de tratamento de sombras (*Shadows*), permanece uma linha na cor preta entre a sombra e o objeto e para resolver esse problema foi feito um controle do tamanho de pixels desta linha, assim, pôde-se resolver o problema.

Para o reconhecimento dos dedos foram traçadas três linhas, sendo uma horizontal e duas verticais, podendo reconhecer os dedos tanto da mão esquerda, quanto da direita. Na posição das linhas traçadas, um contador foi incrementado a cada vez em que passava por um pixel branco, podendo, assim, identificar a presença dos dedos.

A preparação de dados compreende, sobretudo o refinamento das imagens obtidas da etapa anterior. Após capturar os quadros de cor e de profundidade na etapa de aquisição, verificou-se os corpos humanos reconhecidos pelo hardware do dispositivo. Para isso, percorreu-se todos os quadros de profundidade e realizou-se a verificação de cada pixel (os pixels indicados por 1 mostram aqueles pontos que pertencem a um determinado corpo). Nesta etapa, o sistema foi capaz de processar os pixels referentes ao elementos gestuais (precisamente a mão e os dedos), gerando as imagens segmentadas. Por fim, na última etapa, as imagens estavam preparadas para ser interpretadas, de modo a indicar de forma clara o dígito correspondente ao gesto capturado.

5. RESULTADOS

Com o objetivo de avaliar o sistema, foram testados cinco sinais (gestos) em dois ambientes de teste (luz artificial e luz natural), avaliando-se o desempenho do sistema em seis casos de testes para cada sinal.

Buscando alcançar os objetivos deste trabalho, realizou-se uma sessão de testes em dois ambientes, com diferentes tipos de iluminação. O primeiro consiste de um ambiente sob luz artificial enquanto o segundo consiste de um ambiente sob luz solar.

Para os sinais referentes aos dígitos de 2 a 4, foram feitos casos de testes utilizando três diferentes distâncias entre os dedos (pequena, média e grande) e para os sinais referentes aos dígitos 0 e 1, foram utilizadas três posições onde o sinal foi emitido (esquerda, centro e direita). O objetivo aqui foi, portanto, testar até que ponto o local de emissão do sinal e a iluminação poderiam influenciar na detecção dos gestos.

Existe uma grande influência da iluminação nos resultados obtidos mediante o processamento dos sinais. Por ser um sensor voltado para uso em ambientes internos (indoor), o Kinect sofre uma forte interferência quando exposto diretamente a luz solar. Isso ocorre porque a luz solar é uma enorme fonte de raios infravermelhos, o que afeta diretamente o funcionamento do hardware. Deste modo, o valor de 100% de erros causados na etapa de medição (sob a luz solar), são relacionados à condição de luz e não à geometria do objeto (dedos). A luz solar pode interferir ou até impedir a medição infravermelha, gerando ruídos ou falha completa na medição, interferindo no processamento dos pixels da imagens capturadas.

A grande diferença entre os resultados obtidos nos dois ambientes se deve a limitação da subtração de fundo através da subtração de duas imagens, pois, uma subtração de duas imagens necessita que ambas possuam o mesmo fundo e isso não é possível em um ambiente sob luz solar devido a variação de luminosidade.

O fato de que a luminosidade nas imagens obtidas em um ambiente sob luz solar pode variar entre uma imagem e outra, provoca diferença nos valores dos pixels entre a primeira e segunda imagem. Isso não ocorre nas imagens obtidas em um ambiente sob luz artificial devido ao fato de que não ocorre variação de luminosidade.

Por conta dessa influência da luz, para verificar uma possível influência dos demais fatores (tipo e posição do sinal emitido e distância entre os dedos), foram considerados apenas os casos de teste sob a iluminação artificial.

6. REFERÊNCIAS

- ANJO, M. S. Reconhecimento de Gestos Estáticos de Mãos Utilizando a Rede Neural Neocognitron e a Multi-Layer Perceptron. Projeto de iniciação científica, Orientador: Prof. Dr. Ednaldo Brigante Pizzolato, Co-Orientador: José Hiroki Saito. Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, 2009.
- COUTO, Alpia Ferreira, COSTA, Arnaldo Marcio et al. Como compreender o deficiente auditivo. Rio de Janeiro: Expansão Editorial, 1985, 85 p.
- G1. **23,9% dos brasileiros declaram ter alguma deficiência, diz IBGE. Portal G1 Brasil**, disponível em: <<http://g1.globo.com/brasil/noticia/2012/04/239-dos-brasileiros-declaram-ter-alguma-deficiencia-diz-ibge.html>>. Acesso em: 17 de maio de 2013.
- SANTOS, M. F. C. dos; LIMA, M. C. M. P.; ROSSI, T. R. de F. Surdez: diagnóstico audiológico. In: SILVA, I. R.; KAUCHAKJE, S.; GESUELI, Z. M. (Org.) **Cidadania, surdez e linguagem: desafios e realidades**. São Paulo: Plexus, 2003. p. 17-40.

SILVA, Ivani Rodrigues; KAUCHAKJE, Samira; GESUELI, Zilda Maria. **Cidadania, surdez e linguagem:** desafios e realidade. São Paulo: Plexus Editora, 2003, 248 p.

SOUSA, Eloysia Godinho de. **Surdez e significado social.** São Paulo: Cortez, 1982, 96 p.