

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

NAYARA OLIVA FERREIRA

**REQUISITOS PARA UM PROCESSO DE ORDENHA MECÂNICA
AUTOMATIZADA**

TRABALHO DE CONCLUSÃO DE CURSO 2

**PATO BRANCO
2016**

NAYARA OLIVA FERREIRA

**REQUISITOS PARA UM PROCESSO DE ORDENHA MECÂNICA
AUTOMATIZADA**

Trabalho de Conclusão de Curso de graduação, apresentado como requisito parcial para obtenção do título de bacharel do Curso Superior de Engenharia de Computação, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco.

Orientador: Prof. Dr. Marco Antonio de Castro Barbosa

**PATO BRANCO
2016**



MINISTÉRIO DA EDUCAÇÃO
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Engenharia de Computação



TERMO DE APROVAÇÃO

Às 15 horas e 30 minutos do dia 28 de junho 2016, na sala V004, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, reuniu-se a banca examinadora composta pelos professores Marco Antonio de Castro Barbosa (orientador), Beatriz Terezinha Borsoi e Luciene de Oliveira Marin para avaliar o trabalho de conclusão de curso com o título **Requisitos para um processo de ordenha mecânica automatizada**, da aluna **Nayara Oliva Ferreira**, matrícula 01065173, do curso de Engenharia de Computação. Após a apresentação a candidata foi arguida pela banca examinadora. Em seguida foi realizada a deliberação pela banca examinadora que considerou o trabalho aprovado.

Marco Antonio de Castro Barbosa
Orientador (UTFPR)

Beatriz Terezinha Borsoi
(UTFPR)

Luciene de Oliveira Marin
(UTFPR)

Beatriz Terezinha Borsoi
Coordenador de TCC

Pablo Gauterio Cavalcanti
Coordenador do Curso de
Engenharia de Computação

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

Ao Grande Arquiteto do Universo, fonte inspiradora.

Dans la vie, rien n'est à craindre, tout est à comprendre.

Marie Curie

RESUMO

FERREIRA, Nayara O. Requisitos para um processo de ordenha mecânica automatizada. 2016. 70 folhas. Monografia (Trabalho de Conclusão de Curso 2) - Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2016.

Este trabalho apresenta o levantamento de requisitos para a construção de um protótipo automatizado para a ordenha mecânica. O levantamento de requisitos tem como finalidade contribuir para o desenvolvimento de um protótipo, com a finalidade de tentar auxiliar o desenvolvimento de tecnologia brasileira para ordenhas robotizadas, preferencialmente com barateamento de custos. O sistema comporta um braço robótico que atua através de entradas obtidas com técnicas de Visão Computacional. O escopo do trabalho envolve o levantamento de requisitos para o desenvolvimento de um sistema de visão para identificação dos tetos e obtenção das suas coordenadas. O levantamento de requisitos do protótipo foi construído em um ambiente controlado e inicialmente calibrado. Foram realizados alguns testes utilizando filtros para a detecção de bordas como etapa pré-processamento da imagem para localização dos tetos. Este trabalho envolve o estudo da cinemática direta para estabelecer a posição atual do braço mecânico, e da cinemática inversa para determinar um possível caminho para o posicionamento do braço mecânico abaixo dos tetos.

Palavras-chave: Automação. Ordenha mecânica. Robótica. Cinemática. Visão computacional.

ABSTRACT

FERREIRA, Nayara O. Requirements for Mechanical Milking Automated Machine Process. 2016. 70 folhas. Monografia (Trabalho de Conclusão de Curso 2) - Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2016.

This work presents the requirements gathering to construct a prototype of a mechanical milking automated. The requirements capture has the purpose to contribute to the development of Brazilian technology for robotic milking, preferably with lower costs. The system contain a robotic arm that works through inputs obtained with Computer Vision techniques. The scope of this work involves the gathering of requirements to development of a vision system for identifying the teats and to obtain their coordinates. The gathering of requirements of the prototype was constructed for a controlled environment and initially calibrated. Some tests were made utilizing filters to edge detection as a pre-processing step of the image to the location of the teats. This work involves the study of the direct kinematics to establish the current position of the mechanical arm, and inverse kinematics to determine a possible path for positioning the mechanical arm below the teats.

Keywords: Automation. Mechanical milking. Robotics. Kinematics. Computer Vision.

LISTA DE FIGURAS

FIGURA 1 - PASSOS FUNDAMENTAIS EM PROCESSAMENTO DE IMAGENS	15
FIGURA 2 - CONCEITOS DE VIZINHANÇA-DE-4 (A), VIZINHANÇA DIAGONAL (B), E VIZINHANÇA-DE-8 (C)	17
FIGURA 3 - HISTOGRAMAS CORRESPONDENTES A QUATRO TIPOS BÁSICOS DE IMAGENS	18
FIGURA 4 - CORES PRIMÁRIAS E SECUNDÁRIAS	20
FIGURA 5 - CUBO DE CORES RGB. OS PONTOS AO LONGO DA DIAGONAL PRINCIPAL TÊM VALORES DE CINZA DESDE PRETO ATÉ BRANCO NO PONTO (1,1,1)	21
FIGURA 6 - ELEMENTOS DE ANÁLISE DE IMAGEM	24
FIGURA 7 - CONFIGURAÇÃO DE QUATRO TIPOS DE BRAÇOS DIFERENTES	26
FIGURA 8 - DIAGRAMA INDICANDO RELAÇÃO ENTRE A CINEMÁTICA DIRETA E A CINEMÁTICA INVERSA	28
FIGURA 9 - CONFIGURAÇÃO GERAL DO ROBÔ <i>SHELLEY</i>	30
FIGURA 10 - VISTA SUPERIOR DA DIMENSÃO HORIZONTAL	30
FIGURA 11 - PROTÓTIPO DE ÚBERE	30
FIGURA 12 – ROBIX	36
FIGURA 13 – CÂMERA DIGITAL	37
FIGURA 14 – KINECT XBOX 360	37
FIGURA 15 - FLUXOGRAMA DO SISTEMA	41
FIGURA 16 – FOTOS DE DIFERENTES ÚBERES	42
FIGURA 17 - CLASSIFICAÇÃO DOS TIPOS DE ÚBERE	43
FIGURA 18 - CLASSIFICAÇÃO DOS TETOS	43
FIGURA 19 – RESULTADOS DOS FILTROS	45
FIGURA 20 - TESTES UTILIZANDO FILTROS COM IMAGEM DO PROTÓTIPO DE UM ÚBERE	46
FIGURA 21 - TESTES UTILIZANDO FILTROS COM IMAGEM DO PROTÓTIPO DE UM ÚBERE	47
FIGURA 22 - VISTA DE CIMA DAS POSIÇÕES DOS TETOS DO PROTÓTIPO DE ÚBERE	49
FIGURA 23 - GEOGEBRA, OBTENDO PONTOS DESEJADOS DENTRO DA ÁREA DE TRABALHO DO ROBIX	49
FIGURA 24 - CÓDIGO DO MATLAB PARA OBTER ÂNGULOS	50
FIGURA 25 - IMPRECISÃO DO ROBIX	51
FIGURA 26 – KIT DE DESENVOLVIMENTO PARA KINECT	52
FIGURA 27 – SEPARAÇÃO DE OBJETO DE INTERESSE E PLANO DE FUNDO	53
FIGURA 28 – IMAGEM COM FUNDO PRETO E JOGADOR EM TONS DE CINZA	54
FIGURA 29 – ESCLARECIMENTO DO MODO DE CAPTURA DE DISTÂNCIA REALIZADO PELO KINECT	55
FIGURA 30 – TESTE DE PROFUNDIDADE COM PAPEL PARANÁ	56
FIGURA 31 – PROTÓTIPO DE ÚBERE	57
FIGURA 32 – IMAGEM ABERTA NO MATLAB	58
FIGURA 33 – COORDENADAS OBTIDAS MAIS PRÓXIMAS AO CHÃO	59
FIGURA 34 – TESTE COM COORDENADAS BASEADAS NO ARQUIVO TEXTO	60
FIGURA 35 – VALOR DAS COORDENADAS PRÓXIMO AO TETO ESQUERDO	60
FIGURA 36 – VALOR DA COORDENADA DO TETO ESPERADA NO ARQUIVO TEXTO	61
FIGURA 37 – COORDENADA OBTIDA NO ARQUIVO TEXTO	61
FIGURA 38 – VALOR DAS COORDENADAS PRÓXIMO AO TETO DIREITO	62
FIGURA 39 – PIXEL ENCONTRADO NO ARQUIVO TEXTO	62
FIGURA 40 – IMAGEM ILUSTRATIVA DO PROTÓTIPO. A – ROBIX, B- PROTÓTIPO DO ÚBERE, C – KINECT XBOX 360, D- ARDUÍNO	64

LISTA DE SIGLAS

bps	bits por segundo
CIE	<i>Commission Internationale d'Eclairage</i>
cm	centímetros
HSI	<i>Hue, Saturation, Intesity</i>
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	<i>Integrated Development Enviroment</i>
mm	milímetros
nm	nanômetros
RGB	<i>Red, Green, Blue</i>

LISTA DE ACRÔNIMOS

MATLAB	MATrix LABoratory
SEAB	Secretaria da Agricultura e do Abastecimento do Paraná

SUMÁRIO

TERMO DE APROVAÇÃO.....	2
ABSTRACT	6
1 INTRODUÇÃO	11
1.1 OBJETIVOS	11
1.1.1 Objetivo Geral	11
1.1.2 Objetivos Específicos	11
1.2 JUSTIFICATIVA	12
1.3 ESTRUTURA DO TRABALHO	13
2 REFERENCIAL TEÓRICO	14
2.1 VISÃO COMPUTACIONAL E PROCESSAMENTO DE IMAGENS.....	14
2.1.2 Processamento de Imagens.....	14
2.1.2.1 Vizinhaça e conectividade	16
2.1.2.2 Histograma	17
2.1.2.3 Limiarização	19
2.1.2.4 Cor.....	19
2.1.2.5 Filtros.....	21
2.1.2.6 Transformada de Hough.....	23
2.1.2.7 Reconhecimento e interpretação.....	23
2.2 CINEMÁTICA DE UM BRAÇO MECÂNICO.....	26
2.2.1 Cinemática direta	28
2.2.2 Cinemática Inversa.....	29
2.2.3 Calibração	32
2.3 AUTOMAÇÃO E ROBÓTICA	33
3 MATERIAS E MÉTODOS.....	36
3.1 MATERIAIS	36
3.2 MÉTODOS	37
4 PROJETO DO SISTEMA E TESTES PRELIMINARES	39
4.1 ESCOPO DO SISTEMA	39
4.2 MODELAGEM DO SISTEMA.....	40
4.3 EXPERIMENTOS PRELIMINARES	41
4.3.1 Testes preliminares com visão computacional e processamento de imagens .	42
4.3.1 Implementação.....	44
4.4 SEÇÃO MÓDULO AUTOMAÇÃO E ROBÓTICA	47
5 INTEGRAÇÃO DOS MÓDULOS DO SISTEMA.....	52
5.1 AQUISIÇÃO DE IMAGENS E SEGMENTAÇÃO DAS IMAGENS	53
5.2 PROCESSAMENTO DE IMAGENS	54
5.3 PROFUNDIDADE.....	55
5.4 IDENTIFICAÇÃO DOS TETOS	56
5.5 PROTÓTIPO DE ORDENHA	63
6 CONCLUSÃO.....	65
REFERÊNCIAS.....	67

1 INTRODUÇÃO

Neste capítulo são apresentadas as principais motivações para a realização deste trabalho, seus objetivos centrais bem como a estruturação do mesmo.

1.1 OBJETIVOS

Os sistemas robóticos estão, direta ou indiretamente, presentes no cotidiano das pessoas. Isto se deve à capacidade que estes sistemas possuem de perceber o ambiente à sua volta e tomar decisões. O trabalho proposto visa o levantamento de requisitos para o desenvolvimento de um sistema de ordenha mecânica automatizada. Foram estudadas técnicas de visão computacional, processamento de imagens e a robótica para verificar quais dispositivos de *hardware* seriam necessários, quais métodos e técnicas de computação gráfica poderiam ser empregadas bem como a melhor forma de integrar os módulos sensores a atuadores.

1.1.1 Objetivo Geral

O objetivo deste trabalho é levantar requisitos para o desenvolvimento de um protótipo automatizado para ordenha mecânica.

1.1.2 Objetivos Específicos

- Realizar o levantamento de requisitos para o desenvolvimento de um sistema de visão computacional para a captura de imagens.
- Realizar o processamento das imagens obtidas.
- Realizar a localização dos objetos no espaço de trabalho.
- Utilizar um braço robótico para alcançar os tetos em suas coordenadas espaciais.

1.2 JUSTIFICATIVA

No Brasil hoje existem poucas fazendas com o sistema de ordenha voluntária. A pioneira da América Latina foi noticiada pelo jornal Globo Rural (G1, 2013) e também no site da empresa que forneceu o produto, DeLaval (DELAVAL, 2014). De acordo com informações disponíveis no site do G1 (2013), a fazenda Santa Cruz em Castro no Paraná automatizou o processo e o ambiente de ordenha. O sistema identifica o animal através de uma coleira e libera a quantidade de ração baseado na produção de leite do animal. Um braço mecânico faz a higienização e em seguida a ordenha. O sistema mantém o ambiente climatizado e o proprietário tem acesso a informações do estábulo pela internet. O investimento foi de 2,6 milhões de reais e a tecnologia é Suíça.

Outra empresa que desenvolve tecnologia de ordenha é a GEA WestfaliaSurge. Esta empresa desenvolve um sistema de ordenha totalmente robotizado com *design* sob medida. O site da GEA WestfaliaSurge (2014) descreve o funcionamento desse sistema, chamado de *Multibox Mlone*. Segundo o site, o sistema utiliza um direcionador de câmera 3D para a localização dos tetos no úbere e das teteiras. Uma vez localizado os objetos, os alinha de forma que ocorra o acoplamento do conjunto de forma rápida e segura.

Um sistema de automatização de ordenha proporciona mais tempo, aos proprietários e aos funcionários de uma fazenda, para se dedicarem a outras atividades como, por exemplo, o plantio. Além disso, um manipulador mecânico tem sempre o mesmo comportamento, o mesmo som e o mesmo cheiro, ao contrário de um ser humano que pode ter grandes variações nesses quesitos, gerando menos *stress* aos animais.

Além disso, segundo o IBGE (INSTITUTO..., 2016) o Paraná é a terceira maior bacia leiteira do Brasil, totalizando 4.532.614 litros de leite adquiridos em 2014. Logo existe uma viabilidade econômica no desenvolvimento desse tipo de tecnologia no Paraná uma vez que Pato Branco adquiriu 428.743 litros em 2014 gerando um valor bruto de produção (VBP) de R\$420.167.748,00 de acordo com o DERAL (DEPARTAMENTO..., 2012). Ainda no sudoeste do Paraná, segundo o DERAL, Francisco Beltrão em 2014 adquiriu 529.200 litros de leite com um VPB de R\$

518.956.000,00; Dois Vizinhos no mesmo ano produziu 206.400 litros de leite e um VBP de R\$ 202.272.000,00.

1.3 ESTRUTURA DO TRABALHO

Os demais capítulos desse trabalho estão organizados da seguinte forma:

- Capítulo 2 – Apresenta a base teórica necessária para o desenvolvimento deste trabalho. São abordados conceitos como Visão Computacional, Processamento de Imagens, bem como Cinemática Inversa e Direta de um braço mecânico.
- Capítulo 3 – Descrição dos materiais e métodos utilizados nesse trabalho.
- Capítulo 4 – Este capítulo faz uma breve descrição do sistema, a modelagem do sistema bem como os procedimentos usados para atingir-se os objetivos do trabalho e também os estudos e experimentos realizados para familiarização com os conceitos e procedimentos relacionados ao projeto.
- Capítulo 5 – Implementação.
- Capítulo 6 – Uma síntese dos resultados alcançados é apresentada bem como as possibilidades de trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 VISÃO COMPUTACIONAL E PROCESSAMENTO DE IMAGENS

2.1.1 VISÃO COMPUTACIONAL

O ser humano é capaz de constatar o mundo a sua volta de maneira relativamente simples, distinguindo facilmente diferentes formas, cores, texturas, sombras, dentre outras coisas, devido a complexa anatomia do olho e seu sistema de visão.

Segundo Szeliski (2011), conseguir que um computador tenha uma visão parecida com a dos humanos é difícil por ser um problema inverso, no qual se busca recuperar incógnitas dadas informações insuficientes para especificar integralmente a solução. Deve-se então recorrer a física e a modelos de probabilidade para escolher a melhor solução. A visão computacional tenta descrever o mundo que enxerga em uma ou mais imagens reconstruindo suas características como cor, formato e iluminação.

2.1.2 Processamento de Imagens

A visão computacional visa à interpretação de imagens. Para cumprir com este objetivo é necessário fazer um processamento dessas imagens para que assim o computador possa fazer uma melhor leitura dos dados. Segundo Orlandini (2012) apud Orth (2001), uma vez que a imagem é digitalizada, esta é armazenada em forma de matriz na memória do computador. Cada elemento dessa matriz numérica representa um *pixel* da imagem. Processando essa imagem é possível então extrair as informações desejadas de forma descritiva que é a referência matemática dos objetos visualizados. Logo, cada aplicação de visão computacional necessita de uma relação de processamentos diferentes a serem feitos para adquirir-se as informações específicas desejadas.

Porém, existem alguns passos que são fundamentais em processamento de imagens, como mostra a Figura 1. Estes serão explicados a seguir conforme citados em Gonzalez e Woods (2000).

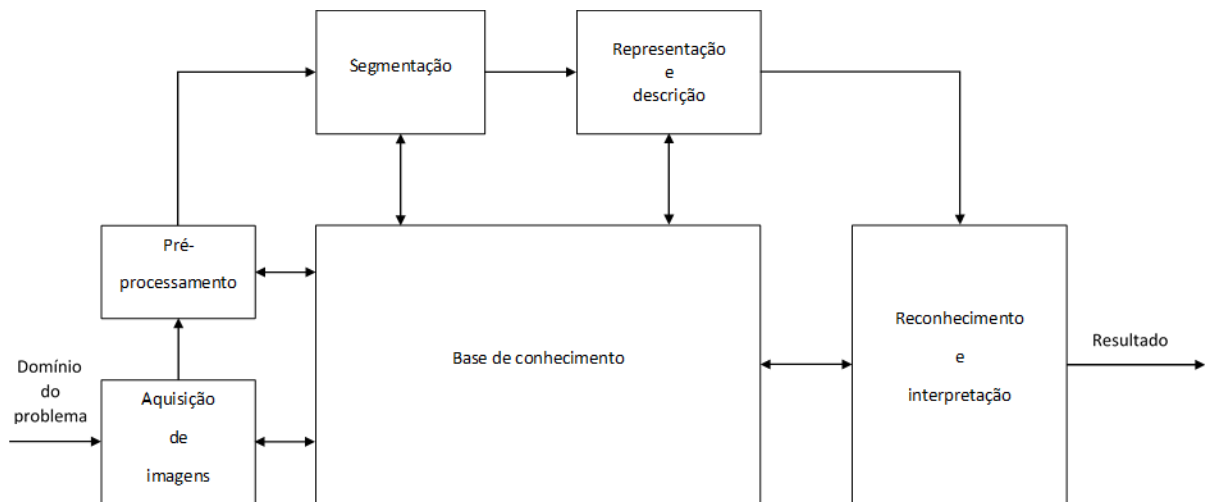


Figura 1 - Passos fundamentais em processamento de imagens

Fonte: Gonzalez e Woods (2000, p. 5).

O primeiro passo é a aquisição da imagem através de um dispositivo físico que seja capaz de captar uma faixa de energia no campo eletromagnético, como por exemplo, o espectro visível ou raios infravermelhos, produzindo um sinal elétrico proporcional ao nível de energia detectado. Em seguida um digitalizador converte então esse sinal elétrico em informação digital.

O próximo passo é um pré-processamento - como, por exemplo, realce de contrastes, remoção de ruído, isolamento de regiões - da imagem para que a imagem captada possa ser aperfeiçoada para as próximas etapas.

Em seguida é feita uma segmentação, facilitando o processamento para que sejam retiradas informações das regiões que contenham dados relevantes. Uma boa segmentação pode ser complexa, mas pode contribuir consideravelmente para encontrar a informação desejada no processamento de imagens.

Após a segmentação ser realizada, os dados iniciais são representados de maneira adequada para os processamentos computacionais seguintes. Assim, é feita uma descrição da imagem de modo a extrair características que são de interesse para atingir o objetivo final do processamento.

Na última etapa os objetos a serem identificados na imagem são rotulados na fase de reconhecimento segundo a descrição feita pelo seu relator. Outorgar significado a um conjunto de objetos rotulados faz parte da fase de interpretação.

Marques Filho e Vieira Neto (1999) descrevem a estrutura de um sistema de Visão Computacional da mesma maneira que Gonzalez e Woods (2000). Porém estes

ainda destacam alguns elementos adicionais de um sistema de processamento de imagens digitais o armazenamento, o processamento e a comunicação.

O tipo de armazenamento deve ser adequado para a aplicação desejada. As três características principais de armazenamento são: armazenamento por curto tempo para o uso durante o processamento; armazenamento *online* para acesso relativamente rápido; e armazenamento em arquivo para acesso menos frequente.

O processamento das imagens digitais geralmente é expresso em forma algorítmica. Logo, com exceção das etapas de aquisição e o resultado a ser exibido, as funções de processamento de imagens podem ser implementadas via *software*. O uso de um *hardware* especializado em processamento de imagens dá-se por necessário somente em casos em que o processamento do computador não atende os requisitos, como por exemplo a velocidade de transferência de dados (FILHO, NETO, 1999).

A comunicação abrange a troca de informações entre sistemas de processamento de imagens. Muitas vezes a comunicação é local, pois *hardware* e *software* se encontram em um mesmo computador. Há uma dificuldade maior de fazer essa comunicação a grande distância. Uma vez que as imagens possuem grande quantidade de dados, é necessário pensar no meio em que essa comunicação se dará, considerando-se o tempo de transmissão e possíveis perdas de dados (GONZALEZ, WOODS, 2000).

2.1.2.1 Vizinhaça e conectividade

Uma imagem é representada por uma função bidimensional $f(x, y)$ discretizada tanto espacialmente quanto em amplitude. Cada *pixel* será representado por letras minúsculas como p e q . Na Figura 2 pode-se observar os diferentes conceitos de vizinhaça, devidamente detalhados a seguir conforme citado por Marques Filho e Vieira Neto (1999).

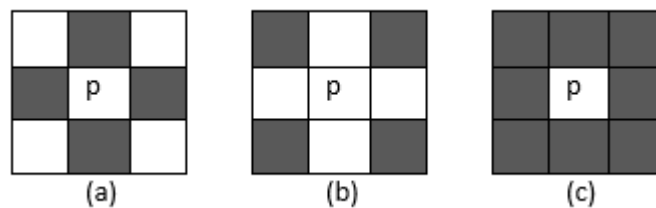


Figura 2 - Conceitos de vizinhança-de-4 (a), vizinhança diagonal (b), e vizinhança-de-8 (c)
 Fonte: Filho e Neto (1999, p. 35).

Na Figura 2 (a) tem-se um *pixel* p de coordenadas (x, y) . Este *pixel* p possui quatro vizinhos horizontais e verticais, cujas coordenadas são: $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$. Estes formam a vizinhança conhecida como $N_4(p)$.

Na Figura 2 (b) os quatros vizinhos diagonais de p tem coordenadas $(x - 1, y - 1)$, $(x - 1, y + 1)$, $(x + 1, y - 1)$, $(x + 1, y + 1)$, sendo denominado de conjunto $N_d(p)$.

Na Figura 2 (c), $N_4(p) \cup N_d(p)$ são chamados de vizinhança-de-8 ou $N_8(p)$.

O conceito de conectividade é de grande relevância para estabelecer bordas de objetos e regiões em uma imagem. Para determinar se um ou mais *pixels* estão conectados é necessário saber se estes são adjacentes; vizinhos-de-4, por exemplo; e se atendem a critérios de similaridade em seus níveis de cinza.

2.1.2.2 Histograma

O histograma apresenta o percentual da ocorrência do nível de cinza baseado nos *pixels* da imagem. “Estes valores geralmente são representados por um gráfico de barras que fornece para cada nível de cinza o número (ou percentual) de *pixels* correspondentes na imagem” (FILHO, NETO, 1999, p. 73).

Logo, um histograma é uma “distribuição de probabilidades e como tal deve obedecer aos axiomas e teoremas da teoria de probabilidade” (FILHO, NETO, 1999, p. 74). Através dos histogramas é possível ter um indício da qualidade da imagem quanto ao nível de contraste e quanto ao seu brilho médio (FILHO, NETO, 1999).

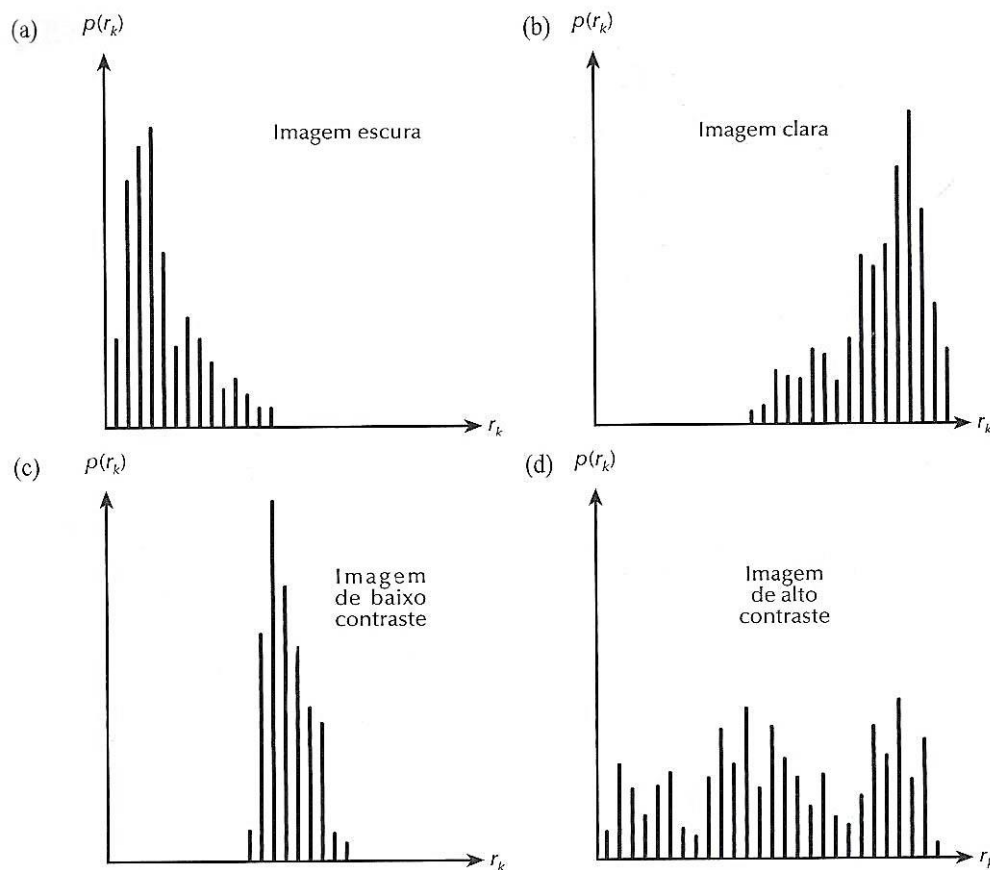


Figura 3 - Histogramas correspondentes a quatro tipos básicos de imagens

Fonte: Gonzalez, Woods (2000, p. 124).

Observando-se a Figura 3 é possível chegar à algumas conclusões. Na Figura 3 (a) os níveis de cinza estão concentrados em direção à extremidade escura do intervalo de níveis de cinza, correspondendo a uma imagem predominantemente escura. Na Figura 3 (b) os pixels estão concentrados em valores próximos ao limite superior da escala de cinza, sendo assim uma imagem predominantemente clara. A Figura 3 (c) tem uma frequência maior de tons de cinza na faixa do meio do histograma, indicando maior incidência de pixels com tons de cinza médios, correspondendo a uma imagem de baixo contraste. Já a Figura 3 (d) é uma imagem com alto contraste, indicando uma distribuição uniforme de pixels que vão desde a faixa inicial do histograma (tons de cinza mais escuros) até a faixa final do histograma (tons de cinza mais claros).

Segundo Orlandini (2012, p. 15), “é possível comparar o grau de semelhança entre duas imagens simplesmente comparando os histogramas destas duas imagens, o que possibilita a aplicação da operação de classificação de objetos”. Através de técnicas específicas os histogramas também podem ser utilizados para mostrar a distribuição espacial dos pixels em uma imagem.

2.1.2.3 Limiarização

Limiarização consiste em separar um objeto de uma imagem do seu plano de fundo. Uma maneira simples de fazer essa separação é a binarização, ou seja, bipartição do histograma, convertendo os pixels cujo tom de cinza é maior ou igual a um determinado valor de limiar (T) em brancos e os demais em pretos (FILHO, NETO, 1999).

Quando existe mais de um objeto na imagem, ou imagens coloridas que possuem intensidades diferentes, é preciso determinar um valor limiar (T) para cada objeto que se deseja separar do fundo. Esse tipo de limiarização multiníveis é mais custosa que a de limiar único, pois existe uma “dificuldade do estabelecimento de múltiplos limiares que isolem efetivamente regiões de interesse, especialmente quando o número de grupos correspondentes do histograma for grande” (GONZALEZ, WOODS, 2000, p. 316).

A iluminação pode interferir na limiarização das imagens, pois altera o histograma e a iluminação não uniforme pode dificultar a segmentação das imagens. Uma das soluções muito utilizadas para compensar a não uniformidade, quando o acesso à fonte de iluminação é disponível, é projetar o padrão de iluminação em uma superfície refletora branca (GONZALEZ, WOODS, 2000).

2.1.2.4 Cor

Fisicamente uma cor é produzida por uma radiação eletromagnética cujo comprimento de onda λ é visível aos humanos no espectro, aproximadamente, entre 400 e 800 nanômetros (nm) (GOMES, VELHO, 2008). Todas as cores vistas pela estrutura do olho humano são combinações das três cores chamadas primárias: vermelho (R, do inglês *red*), verde (G, do inglês *green*), e azul (B, do inglês *blue*). As cores primárias podem ser adicionadas gerando as cores secundárias: magenta (vermelho e azul), ciano (verde e azul), amarelo (vermelho e verde) e branco (vermelho, verde e azul) (GONZALEZ, WOODS, 2000). Essas novas cores geradas,

são as chamadas cores primárias subtrativas, pois quando juntas alguns comprimentos de ondas são absorvidos no espectro de cores, gerando as cores vermelho, verde, azul e preto (SZELISKI, 2011). Todas essas combinações podem ser visualizadas na Figura 4 a seguir.

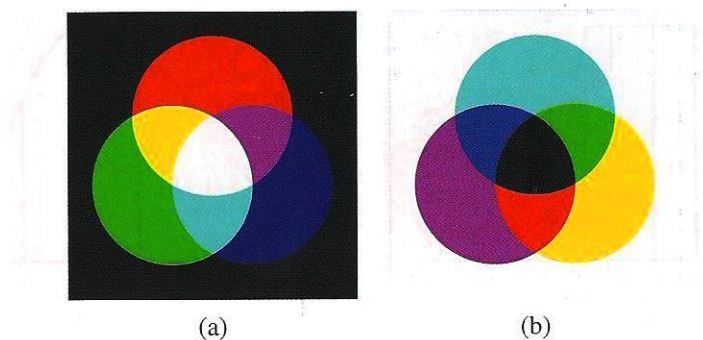


Figura 4 - Cores primárias e secundárias

Fonte: Szelisk (2011, p. 72).

Na Figura 4 (a) são as cores aditivas vermelho, verde, e azul que podem ser misturadas para criar ciano, magenta, amarelo e branco. Na Figura 4 (b) as cores subtrativas ciano, magenta, e amarelo podem ser misturadas para produzir vermelho, verde, azul e preto.

Na análise de imagens automatizada a cor é uma grande aliada que constantemente simplifica a identificação do objeto e a extração de uma cena. (GONZALEZ, WOODS, 2000).

“Young-Helmholtz estabelece que o sistema de processamento de cor do olho humano é um sistema físico de amostragem de dimensão três”, logo, para representar as cores é feita uma redução espectral de cor para um espaço de dimensão infinita, constituída das três cores primárias. (GOMES, VELHO, 2008, p. 127).

Assumindo que todas as cores foram normalizadas, o modelo RGB baseia-se em um sistema de coordenadas cartesianas, em que os valores de R, G e B estão no intervalo de $[0,1]$, como mostra a Figura 5.

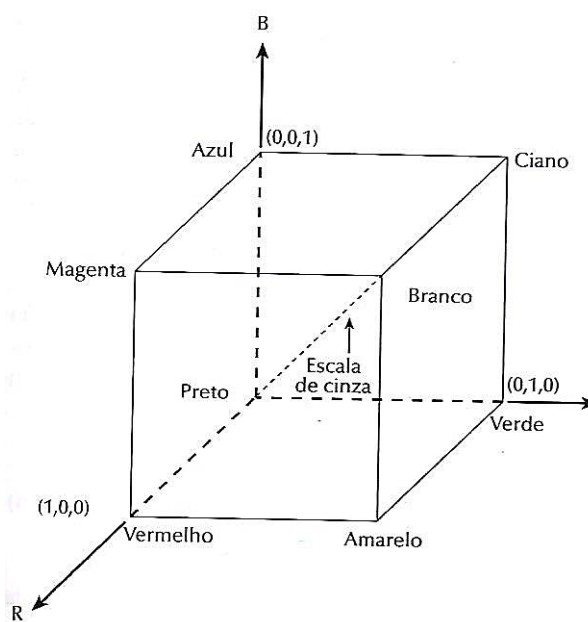


Figura 5 - Cubo de cores RGB. Os pontos ao longo da diagonal principal têm valores de cinza desde preto até branco no ponto (1,1,1)
 Fonte: Gonzalez , Woods (2000, p. 161).

O modelo *Hue Saturation Intensity* (HSI) de cores é formado por três componentes: a matiz, a saturação e o valor. A matiz é a descrição da cor pura ao passo que saturação é o “grau de diluição de uma cor pura por luz branca” (GONZALEZ, WOODS, 2000, p. 162).

O desenvolvimento de algoritmos de processamento de imagens baseados em percepção do sistema visual humano é perfeitamente amparado pelo modelo HSI uma vez que este separa o componente de intensidade *I* da informação de cor da imagem. Exemplos de sistemas que empregam o modelo HSI são, “sistemas de imageamento para determinação automática de amadurecimento de frutas e vegetais, a sistemas para correspondência de amostras de cores, ou inspeção de qualidade de produtos coloridos” (GONZALEZ, WOODS, 2000, p. 162).

2.1.2.5 Filtros

Marques Filho e Vieira Neto (1999) classificam os filtros em duas categorias: técnicas de filtragem espacial e técnicas de filtragem no domínio da frequência. Os filtros que se aplicam no domínio da frequência, também são conhecidos como filtros lineares e não-lineares. Esses filtros atuam especificamente na matriz de *pixels*

empregando operações de convolução. As técnicas de filtragem no domínio da frequência alicerçam-se em transformadas de Fourier da imagem.

A suavização de imagens no domínio espacial através dos filtros lineares é obtida através da convolução de sua entrada com sua função de resposta a impulso unitário. Eles podem ser divididos em passa-baixa, passa-alta e passa-faixa.

Os passa-baixa atenuam ou extinguem os elementos de alta frequência no domínio das transformadas de Fourier – que são as regiões de borda e/ou detalhes finos na imagem – suavizando a imagens causando um leve borramento da mesma.

Já os filtros passa-alta atenuam ou extinguem os elementos de baixa frequência – que são as variações suaves dos níveis de cinza – enfatizando as bordas e as regiões de alto contraste.

Por fim, os filtros passa-faixa filtram regiões específicas de frequências espaciais.

Marques Filho e Vieira Neto (1999) afirmam que os principais métodos de suavização de imagens são o filtro da média e o filtro da mediana. O filtro da média remove os ruídos, porém não consegue conservar as bordas e detalhes finos da imagem. Para então manter essas características o filtro da mediana torna-se mais viável. Este filtro não-linear consiste em percorrer uma imagem substituindo cada pixel pelo elemento mediano da vizinhança. Essa técnica proporciona uma boa performance em imagens com ruído impulsivo, já os ruídos do tipo gaussiano são melhores atenuados pelo filtro da média.

Já um dos filtros para realce de imagens no domínio espacial é o filtro de Sobel. É basicamente um operador que calcula diferenças finitas, dando uma aproximação do gradiente da intensidade dos *pixels* da imagem. Em cada ponto da imagem, o resultado da aplicação do filtro Sobel devolve o gradiente ou a norma deste vetor. Esse filtro realça separadamente as bordas horizontais e verticais mais escuras que o fundo, em uma imagem em escalas-de-cinza, sem realçar os pontos isolados. As cores da imagem são transformadas de RGB para escalas-de-cinza.

2.1.2.6 Transformada de Hough

Jamundá (2000) afirma que essa técnica é utilizada para detectar formas analiticamente representáveis como retas, círculos e elipses em imagens binarizadas. A transformada de Hough define um mapeamento entre o espaço de imagem e o espaço de parâmetros. Cada borda de uma imagem é transformada pelo mapeamento para determinar células no espaço de parâmetros, indicadas pelas primitivas definidas através do ponto analisado. Essas células são incrementadas, e indicarão no final do processo, através da máxima local do acumulador, quais os parâmetros correspondentes a forma especificada.

2.1.2.7 Reconhecimento e interpretação

Um dos processos mais importantes e interessantes do processamento de imagens é o reconhecimento e interpretação destas. Robôs podem participar desse processo, mas para serem capazes de assimilar capacidades similares as humanas eles precisam adquirir diferentes graus de inteligência condizentes com as atividades que irão desenvolver. O conceito de inteligência é altamente discutido na comunidade científica, mas algumas das atividades citadas como inteligentes são a “sensopercepção (como tato, audição e visão), as capacidades intelectuais (como aprendizado de conceitos e juízos, raciocínio dedutivo e memória), a linguagem (como as verbais e gráficas) e atenção (decisão no sentido de concentrar as atividades sobre um determinado estímulo)” (JUNIOR, YONEYAMA, 2010, p. 1).

É possível atualmente dotar uma máquina, em ambientes operacionais limitados, a fim de que esta reconheça padrões e desta forma tenha “(1) a habilidade de extrair informação pertinente a partir de um fundo de detalhes irrelevantes; (2) a capacidade de aprender a partir de exemplos e de generalizar o conhecimento de maneira que ele possa ser aplicado em circunstâncias novas e diferentes; e (3) a habilidade de fazer inferências a partir de informação completa.” (GONZALEZ, WOODS, 2000, p. 407). Gonzalez e Woods (2000), dizem que as análises de imagens baseadas em formulações heurísticas são altamente específicas, logo os sistemas

que possuem análise de imagens não são capazes de fazer algo mais genérico devido às limitações atuais teóricas e de implementação na área de análise de imagens.

Os vários processos que compõem um sistema autônomo de análise de imagens podem ser esquematizados em três áreas básicas: (1) processamento de baixo nível, (2) processamento de nível intermediário e (3) processamento de alto nível. A Figura 6 a seguir ilustra os processos.

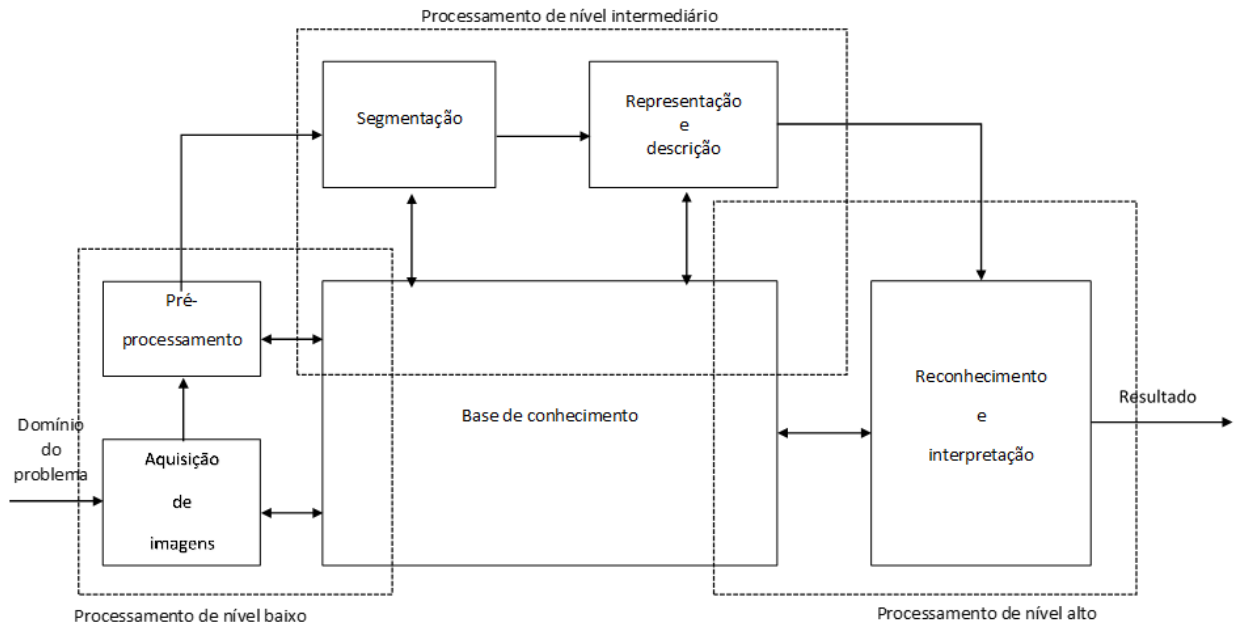


Figura 6 - Elementos de análise de imagem

Fonte: Gonzalez, Woods (2000, p. 408).

No processamento de baixo nível estão os processos que não requerem inteligência. São estes a aquisição de imagens e o pré-processamento de imagens, como por exemplo, redução de ruído ou borramento da imagem.

O processamento de nível intermediário utiliza-se das imagens resultantes do processamento de baixo nível para extrair e caracterizar componentes. Processos como a segmentação e a descrição estão incluídos no processamento de nível intermediário.

Por fim, o processamento de alto nível envolve cognição inteligente para realizar o reconhecimento e interpretação das imagens. À medida que a interpretação de imagens se torna mais profunda, necessita-se de ferramentas que formulem restrições e idealizações com o objetivo de reduzir a complexidade da tarefa, produzindo um sistema altamente especializado.

Para que a análise de imagens ocorra, é necessário então realizar o reconhecimento de padrões em um determinado nível. Gonzalez e Woods (2000, p

409) afirmam que “*padrão* é uma descrição quantitativa ou estrutural de um objeto ou algumas entidades de interesse em uma imagem”. Isto é, padrão é um arranjo de características (descritores). “Uma *classe de padrões* é uma família que compartilha algumas propriedades em comum ” (GONZALEZ, WOODS, 2000, p. 409).

Segundo Gonzalez e Woods (2000, p. 409) vetores, cadeias e árvores são as três principais formas de representação de padrões. Utilizado nas descrições quantitativas, os vetores são os principais arranjos de padrões usados na prática. Os componentes desses padrões então variam conforme a técnica utilizada para descrever o próprio padrão físico do objeto em questão. A escolha de uma medida ou medidas nas quais os componentes do vetor de padrões se baseiam, é imprescindível, pois estas medidas possuem grande influência no desempenho de um sistema de análise de imagens.

Um exemplo dado por Gonzalez e Woods é a separação de três tipos de flores *Iris setosa*, *virginica* e *versicolor* a partir da largura e comprimento de suas pétalas. Logo os vetores têm padrões bidimensionais da forma:

$$F = \begin{bmatrix} l \\ c \end{bmatrix}$$

em que l e c correspondem respectivamente à largura e ao comprimento da pétala. Três classes de padrões são descritas correspondentes as flores.

Quando estruturas estão conectadas de formas relativamente simples, muitas vezes associadas com a fronteira da forma, a representação por cadeia é ideal para a geração de padrões. Descrições por árvores são geralmente mais poderosas e geralmente são utilizadas quando há esquema de organização hierárquica, de acordo com Gonzalez e Woods (2000, p.412).

Os métodos de decisão teórica se baseiam em funções de decisão. Dado um vetor de padrões n -dimensional $x = (x_1, x_2, \dots, x_n)^T$ para M classes de padrões $\omega_1, \omega_2, \dots, \omega_n$ o problema dá-se em encontrar M funções de decisão $d_{1(x)}, d_{2(x)}, \dots, d_{M(x)}$, em que se o padrão x pertence a classe ω_i , então tem-se:

$$d_{i(x)} > d_{j(x)} \quad j = 1, 2, \dots, M; j \neq i$$

“Em outras palavras, um padrão desconhecido x pertence a i -ésima classe de padrões se a substituição de x em todas as funções de decisão fizer com que $d_{i(x)}$ tenha o maior valor numérico. Empates são resolvidos arbitrariamente”. (GONZALEZ, WOODS, 2000, p. 413).

2.2 CINEMÁTICA DE UM BRAÇO MECÂNICO

Um braço mecânico é constituído de elos conectados entre si por meio de juntas. O número de juntas é o que define o grau de liberdade que o robô possui.

Materik (1997), afirma que na extremidade de um braço mecânico encontra-se o efetuador, que pode ser uma ferramenta especial, como uma pistola de pintura ou uma ponta de solda comprida; ou podem ser garras mecânicas.

Dadas essas informações, os braços mecânicos podem ter as mais diversas funções como soldagem, manuseio de materiais, carregamento e descarregamento de máquinas, acabamento de pintura em *spray*, montagem (além de juntar as peças fazer a soldagem), inspeção e manipulação remota.

Materik (1997) descreve dois tipos de juntas: o grupo que desliza, que inclui as juntas cilíndricas, prismáticas ou planas; e o outro grupo que permite apenas a rotação das juntas. As combinações desses diferentes tipos de juntas podem formar quatro tipos de braços robóticos, como podem ser vistos na Figura 7. Esses braços são classificados como cartesianos, cilíndricos, polar e articulado.

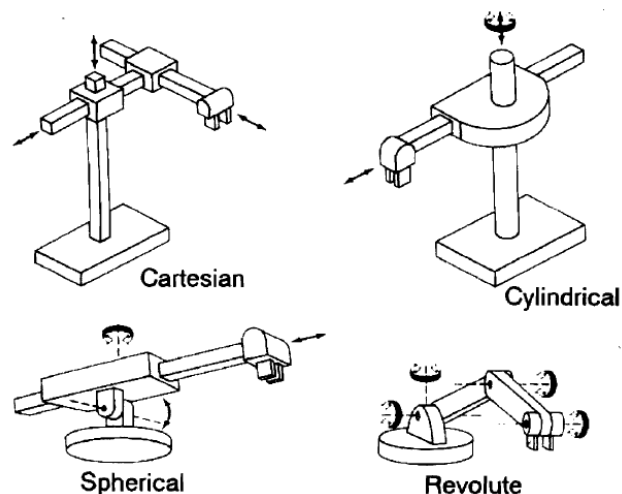


Figura 7 - Configuração de quatro tipos de braços diferentes

Fonte: Materik (1997, p. 30).

Os robôs polares (referido como *spherical* na Figura 7) possuem duas juntas rotacionais e uma prismática. Os robôs cilíndricos (referido como *cylindrical* na Figura 7) possuem uma junta rotacional e dois eixos lineares. Os robôs cartesianos (referido

como *cartesian* na Figura 7) realizam seus movimentos através de três eixos lineares (SHHEIBIA, 2001). Já o articulado (referido como *revolute* na Figura 7), é composto de juntas articuladas que permitem a rotação somente em duas direções, horário e anti-horário em torno de um ponto fixo. (MATERIK, 1997). Os valores de rotação dessas juntas articuladas, em graus, em radianos, são os conjuntos de valores (vetores) que representam a configuração do robô. (SANTOS, LOPES, JUNIOR, 2005). São esses dados que foram acompanhados nos movimentos do braço Robix.

A cinemática visa - considerando a geometria do braço, e um sistema de coordenadas fixas - saber a posição atual e controlar a posição final do braço e não o caminho do movimento. Existem duas maneiras de analisar esse problema. O primeiro é olhar individualmente a posição de cada servo, começando da base até o efetuator, e assim determinar onde o efetuator está. E o segundo é primeiramente determinar a posição desejada do efetuator e então, de trás para frente, ou seja, do efetuator até a base, calcular as configurações dos servos para entregar o efetuator na posição desejada.

O primeiro ponto de vista é o que se denomina de cinemática direta, no qual a questão é determinar a atual posição do efetuator em relação ao ponto de referência do sistema de coordenadas. O segundo ponto de vista citado, é a chamada cinemática inversa. Esta busca, dado um ponto desejado de alcance, saber se este ponto é realmente acessível ao braço levando em consideração sua geometria. Se o ponto desejado for atingível, é interessante saber, quantas configurações diferentes dos servos satisfazem a condição (MATERIK, 1997).

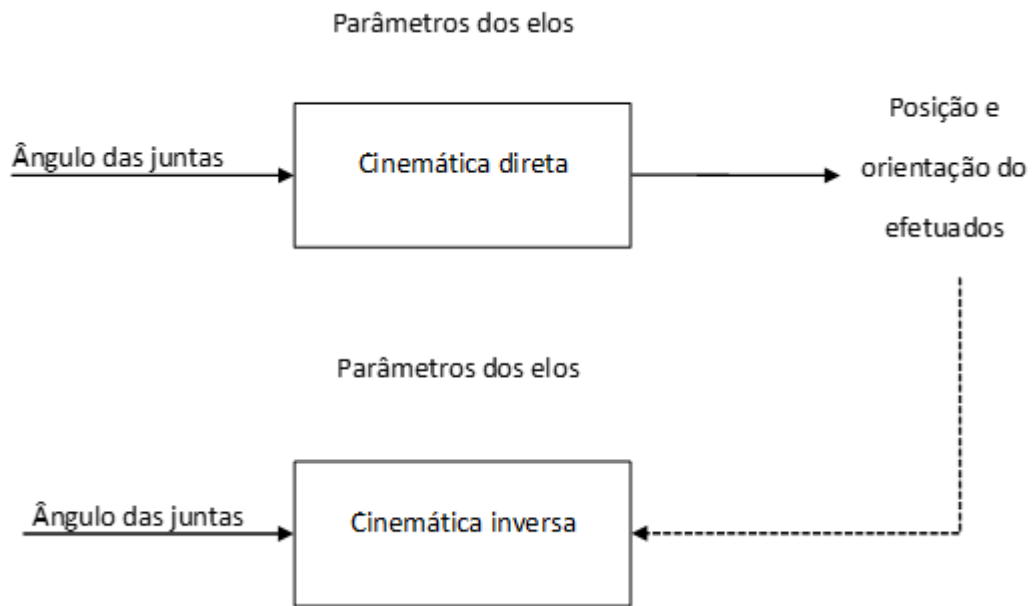


Figura 8 - Diagrama indicando relação entre a cinemática direta e a cinemática inversa
 Fonte: Traduzido de Marterik (1997, p. 32).

2.2.1 Cinemática direta

A álgebra matricial, de acordo com Marterik (1997), é um dos métodos mais populares para representar a geometria espacial dos pares de juntas-elos do braço robótico em relação à referência fixa dos sistemas de coordenadas.

Essa solução é conhecida como método de Denavit e Hantenberg e utiliza uma transformação homogênea de matrizes para descrever o relacionamento espacial entre os dois elos adjacentes.

A matriz de Denavit e Hantenberg, como citada por Tsai (1999), pode ser descrita associando o eixo de junta ao eixo z, e tendo cada matriz representada pelo produto de quatro transformações envolvendo rotações e translações, da seguinte maneira:

$${}^i - 1_{A_i} = T(z, d)T(z, \theta)T(x, a)T(x, \alpha)$$

Onde $T(z, d)$ é a translação d ao longo do eixo z; $T(z, \theta)$ representa uma rotação θ em torno do eixo z; $T(x, a)$ é a translação a ao longo do eixo x; e por fim $T(x, \alpha)$ representa a rotação α em torno do eixo x.

Assim tem-se:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

onde, α , a e d são constantes, e θ é uma variável de junta para uma junta rotativa.

2.2.2 Cinemática Inversa

Este problema é o inverso da cinemática direta. Logo todos os cálculos são feitos baseados no mundo do sistema de coordenadas cartesianas ao invés de dados anexados ao braço. São usados métodos analíticos e numéricos iterativos que fazem a inversão do modelo geométrico.

Uma maneira mais direta de descobrir os ângulos das juntas para posicionar o efetuator é conhecida como método geométrico. Assim o alcance de cada junta-elo também conhecido por *work envelope* (âmbito de movimentos que o braço pode realizar, se movimentando para frente, para trás, para cima e para baixo) é representado por uma forma geométrica. Os pontos de intersecção dessas formas geométricas, geradas pelos alcances de cada junta-elo, são calculados iniciando no efetuator em direção à base. Um ponto de intersecção criado pelo alcance de uma junta-elo com o alcance da próxima junta-elo em sequência é escolhido para representar a junta do braço (Marterik, 1997).

Dos vários pontos de intersecção podem então ser calculadas as configurações rotacionais dos vários servos. Marterik (1997) descreveu o modelo matemático que realizou para o controle do braço Shelley. O braço Shelley foi produzido pela empresa Robix/*Advanced Design* e contém seis servos de articulados, que rotacionam certa de 180 graus.

Marterik (1997) afirma que o movimento tridimensional pode ser fragmentado em movimentos horizontais e verticais uma vez que esses movimentos são parecidos embora realizados em dimensões diferentes. Para isso decidiram que os servos (1) e (2) controlariam o movimento horizontal, os servos (3) e (4) controlariam o movimento

vertical e os servos (5) e (6) controlariam a rotação e o aperto da garra. A Figura 9 mostra a configuração do braço dado pelo autor.

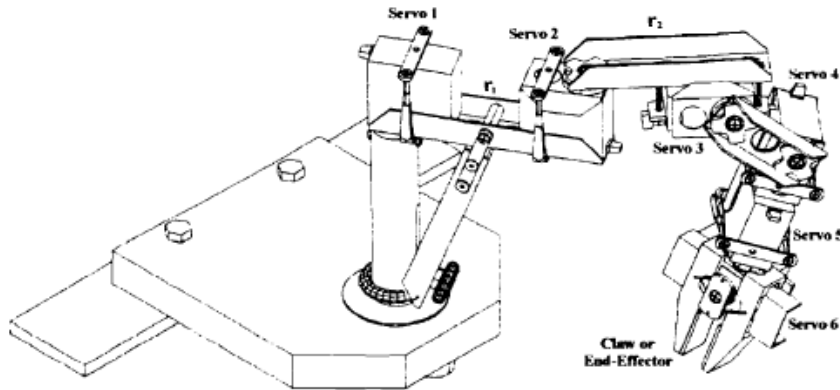


Figura 9 - Configuração geral do robô Shelley

Fonte: Martelik (1997, p. 31).

Dado então que a base do robô está do centro do círculo r_1 , como mostra a Figura 10, na posição $(0,0)$, tem-se a equação do círculo:

$$x^2 + y^2 = r_1^2 \quad (1)$$

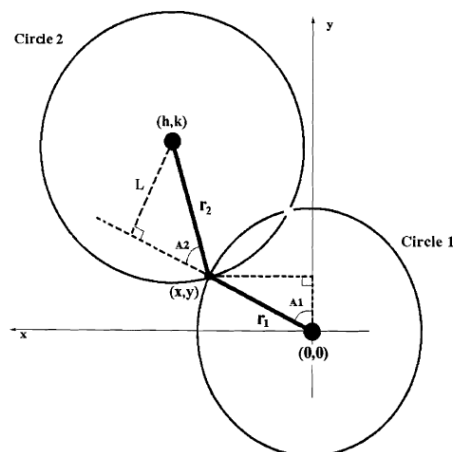


Figura 10 - Vista superior da dimensão horizontal

Fonte: Martelik (1997, p. 33).

Na Figura 10 o ponto $(0,0)$ representa a base, (h,k) o efetuador, (x,y) o ponto de intersecção, r_1 é o comprimento do elo 1 e r_2 o comprimento do elo 2.

Com esses dados obtém-se a Equação 22 do círculo:

$$(x - h)^2 + (y - k)^2 = r_2^2 \quad (2)$$

Ainda na Figura 10 é possível perceber que o servo 1 está associado com o ângulo A1 e o servo 2 associado ao ângulo A2. Logo para descobrir o ponto de intersecção entre os dois círculos é necessário garantir que essas figuras geométricas de alcance se encontrem. Isso ocorre quando as coordenadas do efetuador não estão em seu alcance máximo (r_1+r_2), nem em seu alcance mínimo. O alcance mínimo do efetuador ocorre com o servo 1 em qualquer posição e o servo 2 em sua máxima rotação. As equações envolvidas para encontrar o alcance mínimo do efetuador são:

$$\theta = \pi - \alpha \quad (3)$$

$$B = r_2 \sin \theta \quad (4)$$

$$C = r_2 \cos \theta \quad (5)$$

$$A = r_1 - C \quad (6)$$

$$\text{Distância} = (A^2 + B^2) \quad (7)$$

onde α é a máxima rotação do servo 2.

Assim Martelik (1997) afirma que o efetuador não está nem em ponto máximo nem em ponto mínimo, garantido que existe pelo menos um ponto de intersecção entre os círculos. Caso exista mais de um ponto de intersecção o escolhido será o que exigir menos rotações dos servos.

O próximo passo é então encontrar o ponto (x, y) . Para encontrar o valor de y , subtrai-se a Equação 2 da Equação 1 e obtém-se:

$$y = \left(\frac{h}{k}\right)x + \left(\frac{r_1^2 - r_2^2 + h^2 + k^2}{2k}\right) \quad (8)$$

Então se substitui esse valor de y na Equação 21 e temos:

$$\left(\frac{h^2}{k^2} + 1\right)x^2 - \left[\frac{h(r_1^2 - r_2^2 + h^2 + k^2)}{k^2}\right]x + \left(\frac{r_1^2 - r_2^2 + h^2 + k^2}{2k}\right)^2 - r_1^2 = 0 \quad (9)$$

A
B
C

Resolvendo os coeficientes A, B e C indicados na Equação 9 obtém-se x . A partir desses valores é possível então encontrar os ângulos desejados A1 e A2, através das seguintes equações:

$$A1 = \sin^{-1} x/r_1 \quad (10)$$

$$A2 = \sin^{-1} L/r_2 \quad (11)$$

Depois disso a tarefa realizada então por Martelik (1997) foi transformar essas configurações rotacionais, em radianos, para valores de passos para o Robix.

2.2.3 Calibração

Para que o braço mecânico possa se deslocar até um objeto de interesse é necessário retirar de pontos específicos das imagens as coordenadas para o braço mecânico. A fim de que estas coordenadas sejam obtidas, é essencial determinar a posição e a orientação da câmera (seu sistema de coordenadas) em relação ao sistema de coordenadas do robô. Este procedimento é conhecido como calibração de câmera (CAMBRON, NORTHRUP, 2006).

Uma câmera pode ser fragmentada em dois tipos de parâmetros, os internos e os externos. Os parâmetros extrínsecos (externos) descrevem a posição e orientação da câmera no espaço. Já os parâmetros intrínsecos (internos) descrevem os aspectos do mapeamento, como por exemplo, o comprimento focal ou a posição do ponto principal (THORMÄLLEN, BROSZIO, MIKULASTIK, 2006).

Özuysal (2004) realiza um estudo em calibrações manual e automática para fazer medições tridimensionais em ambiente desconhecido. A calibração manual calcula os parâmetros da calibração com a ajuda de um objeto de calibração planar. Já o método de auto calibração exige pontos correspondentes entre as imagens.

Guerra (2004) realizou um trabalho calibrando um par de câmeras em movimentos desconhecidos. O trabalho descreve como foi feito o casamento de pontos nas imagens e os cálculos da calibração. Esse método é implementado em sistemas cuja calibração precisa ser feita remotamente.

Heyden e Åström (1999) desenvolveram uma reconstrução Euclidiana utilizando o mínimo de parâmetros intrínsecos, também conhecido como calibragem flexível.

De acordo com Segundo (2007), os pesquisadores ampliaram as técnicas de auto calibração - além da restrição dos parâmetros intrínsecos - utilizando limitações na movimentação das câmeras e restrições de cena.

Loaiza, Raposo e Gattass (2010) definem, analisando diferentes métodos em várias literaturas, que o processo de calibração é dividido em três etapas: aquisição de dados, calibração inicial e otimização da calibração.

A aquisição de dados abrange algoritmos de processamento de imagens e reconhecimento de imagens que propicia rápida captura e reconhecimento dos pontos

de referência, conhecidos como marcadores. Muitos dos padrões desses algoritmos, usados para calibração de câmeras individuais, foram reutilizados para calibração de múltiplas câmeras. São exemplos desses padrões: padrão planar, padrão unidimensional e padrão adimensional. Esses padrões propiciam a captura de pontos de referência com base na detecção e reconhecimento de características implícitas do seu formato (LOAIZA; RAPOSO; GATTASS, 2010).

A próxima etapa, a calibração inicial, é a realização dos cálculos dos parâmetros intrínsecos, extrínsecos e da distorção das múltiplas lentes que integram o sistema (Loaiza, Raposo e Gattass, 2010).

Um tipo de calibração, pouco utilizado por ser trabalhoso, é a calibração fotogramétrica. Nesta calibração os parâmetros são calculados usando informações dadas por um padrão geométrico conhecido no espaço tridimensional. Dois métodos desse tipo de calibração são propostos, um por Tsai e outro por Zhang (mais flexível) o que utiliza um padrão planar. Com as informações obtidas pelos pontos referenciais colocados sobre o plano planar, é possível determinar medidas dos objetos do mundo real que contém informações tridimensionais com medidas atuais capturadas no plano bidimensional (Loaiza, Raposo e Gattass, 2010)

Como na calibração inicial, a auto calibração calcula os parâmetros intrínsecos e extrínsecos através dos “pontos referenciais devidamente correlacionados e identificados entre as diferentes câmeras que visualizam o mesmo cenário.” (Loaiza, Raposo e Gattass, 2010, p. 3).

A última fase, a otimização da calibração, faz uma coletânea de métodos usados para aperfeiçoar o valor inicial dos parâmetros de calibração. A solução é auferida com base na minimização algébrica, que aproxima uma solução algébrica linear para um problema que não é linear.

2.3 AUTOMAÇÃO E ROBÓTICA

Robôs são máquinas que copiam o homem, não em sua forma física, e sim em sua capacidade de raciocínio e aprendizagem, de locomoção, de manipulação entre outras atividades. (GALHANO, 1992).

Robôs podem ser dotados de visão e Inteligência Artificial (IA), apresentando, então, a capacidade de responder às mudanças no ambiente e executar a tarefa

desejada. Para isso são necessários a captura e o processamento de imagens orientado especificamente para um tipo de problema. Existem diferentes técnicas de processamento de imagens e os resultados das mesmas devem ser interpretados pelo computador e pelo usuário. (MENEZES et al.,2012). A partir desses resultados a visão computacional extrai os dados que necessita e os utiliza de maneira mais conveniente.

Grande parte dos benefícios que são proporcionados pela robótica e automação se deram graças ao uso de manipuladores mecânicos. ” Um manipulador robótico é composto por diversos elos mecânicos interligados por eixos (rotacionais ou lineares) e são acionados por motores apropriados” (GALHANO, 2012, p.15). Este manipulador efetuará uma tarefa sobre um objeto desejado em certo tempo e com certa precisão. Para se obter um movimento preciso e ágil é necessário que o manipulador tenha flexibilidade e percepção do ambiente condizentes com a precisão e agilidade requeridas. (SHHEIBIA, 2001).

O robô manipulador destaca-se por ser semelhante a um braço humano realizando movimentos semelhantes e também por agregar várias áreas do conhecimento da ciência e da engenharia. (ROSSATO, 2009).

O número de manipuladores robóticos utilizados na indústria vem crescendo gradativamente. Como estes precisam ter precisão para realizar suas tarefas necessita-se de um estudo aprofundado sobre os diversos fatores envolvidos na automação, como, por exemplo, a cinemática (SILVA et al, 2006). Os manipuladores robóticos têm como tarefa, segundo a Associação de Indústrias da Robótica (RIA), mover peças, materiais ou ferramentas. Assim precisa-se programa-los de maneira a levar essas peças, materiais ou ferramentas de maneira desejada, para isto ocupa-se a cinemática que relaciona posições das juntas e a posição do efetuator, este último opera sobre o objeto a ser manipulado (PAZOS,2002). Silva et al (2006) apresenta a modelagem cinemática de um robô, porém não utiliza conceitos de cinemática direta.

Outra ciência que auxilia no processo de automatização nas indústrias é a visão computacional. Esta desenvolve métodos de capturar imagens, analisa-las, retirar informações dessas imagens e assim permitir que um computador “enxergue” o mundo a sua volta através de câmeras de vídeo, sensores, *scanners* entre outros. (MILANO, HONORATO,2009).

Ferreira et al. (2004) desenvolveu um sistema de controle de um servomecanismo, baseado em visão computacional. Fisher (2004) afirma que visão

computacional e inteligência artificial estão estritamente relacionadas em se tratando do controle de um manipulador mecânico.

Klomp et al. (1990) desenvolveram um sistema robótico para ordenhar as vacas sem interferência de seres humanos. O foco do trabalho foi o braço mecânico desenvolvido especialmente para o projeto. O trabalho teve como principal foco a estratégia de encaixe da teteira no teto da vaca. A posição dos tetos da vaca, no trabalho, era obtida por um sistema ultrassônico.

Segundo Becker (2013) “uma definição apropriada para a técnica de visão artificial ou visão computacional, é que pode ser classificada como a ação de um determinado atuador em função de imagens coletadas em tempo real e devidamente processadas.”

3 MATERIAS E MÉTODOS

Este capítulo descreve os materiais utilizados no desenvolvimento deste trabalho bem como os métodos adotados para a sua realização.

3.1 MATERIAIS

Para os experimentos preliminares foram adotados os materiais elencados na sequência:

- Manipulador robótico do kit didático Robix, ilustrado na Figura 12;
- Controlador do kit Arduino;
- Protótipo de um úbere de uma vaca, ilustrado na Figura 11;



Figura 11 - Protótipo de úbere

Fonte: do autor

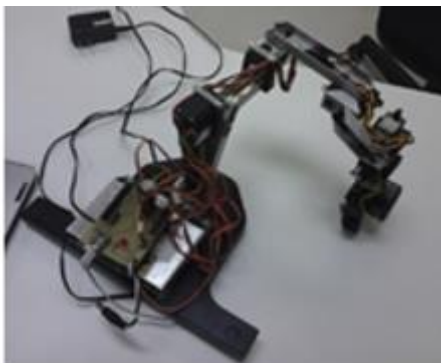


Figura 12 – Robix

Fonte do autor.

- Câmera digital, ilustrado na Figura 13;
- A IDE CodeBlocks;

- A IDE MPIDE (MultiPlataform Arduino compatible IDE);
- Visual Studio 2015



Figura 13 – Câmera Digital

- Um módulo de sensor de movimentos Kinect Xbox 360 (conforme Figura 14).



Figura 14 – Kinect Xbox 360

3.2 MÉTODOS

Os métodos adotados na realização deste trabalho foram divididos em etapas as quais são enumeradas no texto que segue.

- Etapa 1 - Revisão de Literatura: O início dos trabalhos se deu pela imersão na literatura dos temas relacionados ao trabalho. Foram consultadas as referências básicas da área, bem como, foi verificado o estado-da-arte dos temas em causa. Nesta etapa foram vistos os temas: Computação Gráfica e suas vertentes Processamento de Imagens e Visão Computacional, robótica e cinemática.

- Etapa 2 – Coletas de Dados e Requisitos em Campo: Nesta etapa foram registadas imagens de diferentes úberes com uma câmara em ambiente real de ordenha. Junto dessas imagens foram também manualmente registradas a distância entre a câmara e os tetos mais próximos desta. Em outras propriedades foram averiguadas as melhorias necessárias nos sistemas de ordenha implantados. Foram também analisados no ambiente de ordenha a rotação entre os animais, o sistema que registra a produção de leite e outros dados de cada animal, o sistema de higienização, método de identificação de mastite, etc. Com essas informações foi possível perceber quais são os requisitos, e problemas (como por exemplo a iluminação do ambiente) a serem resolvidos para o desenvolvimento do protótipo e um futuro produto.
- Etapa 3 – Entrevista com Usuários: Estas entrevistas ajudaram a recolher dados importantes sobre o funcionamento das ordenhas mecânicas. Foi possível observar os cuidados para com o animal. Estes cuidados têm relevância pois o bem-estar do animal ajuda no aumento de produção de leite. Tendo em mente esse dado percebeu-se a importância da precisão e silêncio do manipulador mecânico. Os usuários demonstraram interesse e entusiasmo na descrição do possível produto oriundo deste projeto.
- Etapa 4 – Sistema de Visão: Nesta etapa foram realizados alguns experimentos para a tomada de decisão sobre o equipamento ideal a ser utilizado para a captura de imagens. Foram utilizados a câmara digital ilustrada na Figura 12, bem como, o sensor de movimentos Kinect, ilustrado na Figura 13. Nesta etapa a câmara digital foi integrada ao ambiente de análise. Primeiramente, a câmara foi fixada em um suporte para capturar imagens. O sistema com apenas uma câmara não foi suficiente para o rastreamento dos tetos. A partir desta premissa foi utilizado o Kinect para o sistema de visão a fim de permitir o rastreamento e reconhecimento dos objetos (tetos do úbere). O Kinect foi então utilizado em um ambiente de trabalho controlado, ou seja, utilizando o protótipo de um úbere. As imagens capturadas foram tratadas e processadas extraindo as informações necessárias.
- Etapa 5 – Cinemática: Nessa etapa foi desenvolvido o módulo que trata da cinemática do manipulador robótico. O módulo de cinemática direta calcula a posição atual do braço mecânico. A cinemática inversa converte as coordenadas virtuais obtidas na Etapa 4 para coordenadas reais, permitindo

assim que o manipulador mecânico possa posicionar a teteira no teto da vaca. Nesta Etapa foi utilizado o manipulador Robix.

- Etapa 6 – Validação e Testes: Nesta etapa foram levantados os requisitos necessários para o desenvolvimento de um sistema de ordenha automatizada. Foram ainda realizadas nessa etapa, baterias de testes para avaliar as possíveis soluções e selecionar a mais adequada ao problema. Nesta etapa foi feita a correção de eventuais problemas encontrados bem como a identificação de futuras melhorias para o sistema de ordenha.

4 PROJETO DO SISTEMA E TESTES PRELIMINARES

Este capítulo descreve o escopo do sistema, a modelagem do sistema e os experimentos preliminares.

4.1 ESCOPO DO SISTEMA

O trabalho objetiva o levantamento de requisitos básicos para o desenvolvimento de um protótipo para um sistema de ordenha mecânica automatizada. Considerando-se que um galpão automatizado para ordenha é rico em tecnologias e detalhes, é fundamental o levantamento de requisitos, necessários para a completa automatização de um sistema de ordenha mecanizado. Devido ao grandioso escopo desta tarefa, optou-se pela utilização de um protótipo de ordenha mecanizada para fins de estudo de caso e análise e validação de tecnologias, métodos e necessidades para a completa automação do processo de ordenha e, desta forma, conseguir dimensionar como se chegar a um sistema em escala de maior dimensão.

O protótipo permite identificar quais técnicas de visão computacional e processamento de imagens são ideais para a utilização no sistema. Para tal tarefa foram consideradas fotos em ambiente real e em um ambiente de simulação controlado. Com o protótipo é possível obter os cálculos cinemáticos necessários para a movimentação do manipulador robótico para o posicionamento da teteira abaixo dos tetos.

Portanto, para cumprir com estes objetivos o protótipo realiza tarefas básicas e específicas para que a ordenha seja efetuada, sendo elas o reconhecimento dos tetos do úbere do gado leiteiro e então o deslocamento do braço robótico até eles.

A alimentação do sistema é realizada com imagens dos úberes, no sistema de cores RGB, capturadas por um sistema de visão. Nos experimentos realizados neste trabalho, o braço mecânico Robix se movimentará até atingir um protótipo de úbere de uma vaca. Para cumprir com os objetivos propostos neste trabalho não foram levados em consideração a possível movimentação do gado leiteiro (fato que gera um problema computacional complexo que é a reconfiguração dinâmica de sistemas) e o reconhecimento de uma teteira (insuflador) e seu encaixe no teto.

Fotos reais dos úberes dos animais foram utilizadas na tentativa de que a identificação dos tetos fossem o mais próximo de uma situação de atuação em ambiente real.

Neste trabalho o braço mecânico Robix movimenta-se até abaixo de um protótipo de úbere de gado bovino para fins de validação.

4.2 MODELAGEM DO SISTEMA

O fluxograma ilustrado na Figura 15 mostra quais são os passos realizados para auferir os objetivos do trabalho.

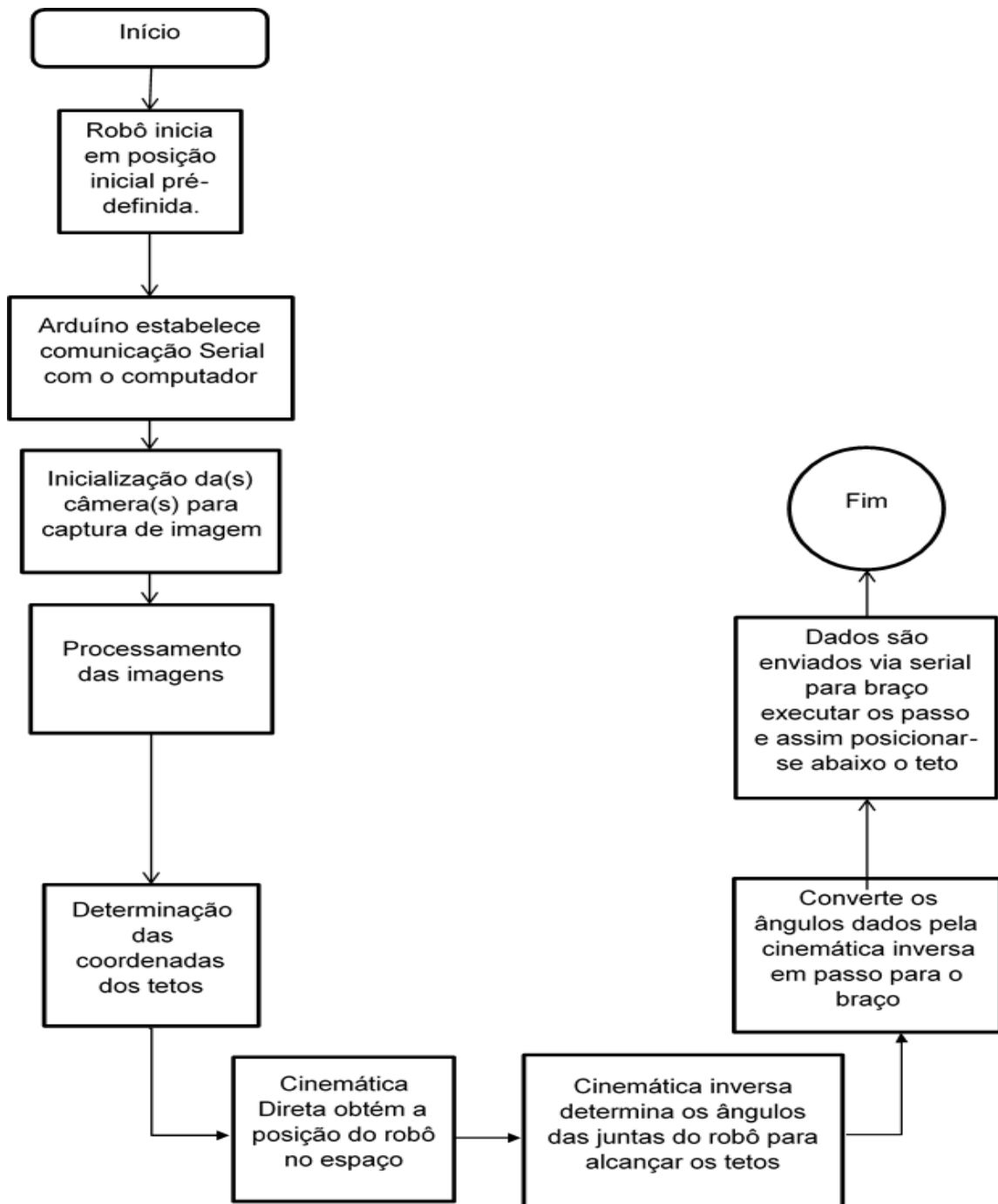


Figura 15 - Fluxograma do sistema
Fonte do autor.

4.3 EXPERIMENTOS PRELIMINARES

Esta seção descreve os experimentos realizados para a familiarização dos elementos envolvidos no trabalho tais como, a captura e o processamento de imagens, comunicação entre Arduíno e Robix, e envio de coordenadas para a atuação do manipulador robótico.

4.3.1 Testes preliminares com visão computacional e processamento de imagens

Em pesquisa de campo em uma propriedade rural da região de Pato Branco foram coletadas 52 imagens de mamas. Os animais foram fotografados por diferentes ângulos e foram feitas medições para auferir as distâncias do objeto alvo (os tetos neste caso) até a câmera. A Figura 16 apresenta algumas imagens capturadas.

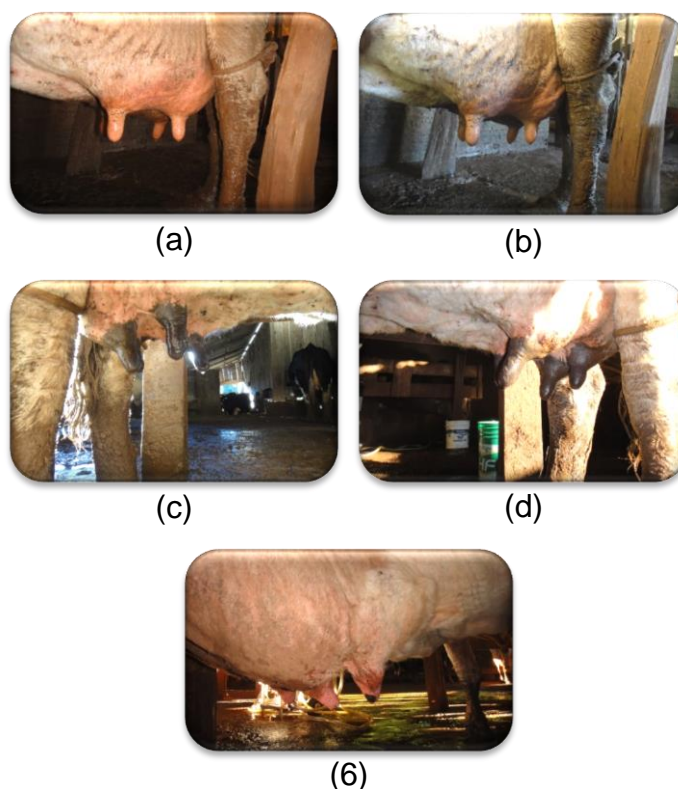


Figura 16 – Fotos de diferentes úberes
Fonte do autor.

Como pode-se observar nas figuras há uma diversidade visível nos úberes do gado leiteiro. Pode-se verificar diferenças de tamanho, cor, forma, etc. A iluminação do ambiente é outra característica preponderante na qualidade da imagem adquirida.

Os diferentes tipos de úberes são classificados conforme a Figura 17 (ARANTES, 2014):

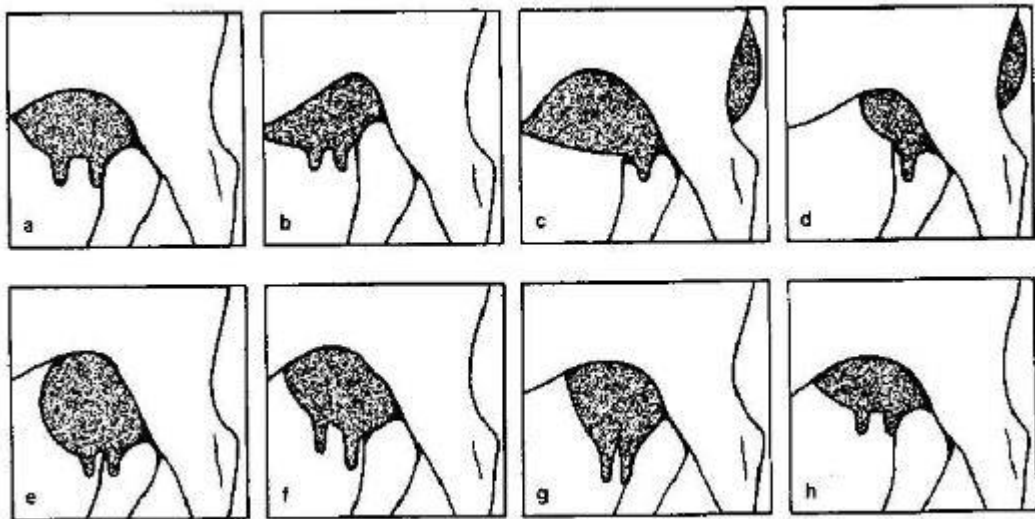


Figura 17 - Classificação dos tipos de úbere

Fonte: Arantes (2014), p. 14

São eles:

- a) Típico para ordenha
- b) Úbere abdominal – inserção estende-se na parede abdominal ventral
- c) Abdominocoxal – volumoso, com base larga (inserção alongada e anterior e caudal)
- d) Úbere coxal – (se localiza entre os membros posteriores com grande inserção caudal)
- e) Úbere esférico (mama pêndula)
- f) Úbere em escada
- g) Úbere triangular
- h) Úbere juvenil

Existem ainda os diferentes tipos de tetos mostrados também por Arantes (2014 que cita Rubin 2011) como mostra a figura abaixo.

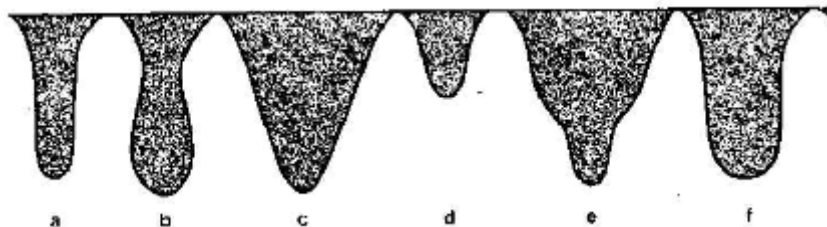


Figura 18 - Classificação dos tetos

Fonte: Arantes (2014, p.14)

São eles:

- a) Teto cilíndrico

- b) Teto volumoso e dilatado na extremidade distal
- c) Teto cônico
- d) Teto pequeno
- e) Teto com dilatação na cisterna do teto

A captura destas imagens foi o primeiro passo para os experimentos preliminares.

4.3.1 Implementação

Os estudos realizados preliminarmente envolveram a programação na linguagem C utilizando-se a IDE CodeBlocks para o processamento das imagens e a programação para o *chipKIT Uno32* plataforma Arduino utilizando a IDE MPIDE.

O primeiro passo foi aplicação de filtros passa-alta nas imagens obtidas.

As imagens apresentam regiões e objetos que não são de interesse, ao utilizar filtro de Sobel para detecção de bordas observaram-se muitos “ruídos” na imagem. Para evitá-los, utilizou-se inicialmente o filtro passa-baixa da Mediana (Figura 19 (b)) para suavização da imagem, antes da utilização do filtro passa-alta (Sobel Figura 19(c)). Na sequência, foi aplicada a transformação de Hough.

Para codificar esses filtros na linguagem C, utilizou-se a biblioteca FilterDII.

A Figura 19, apresenta os resultados dos filtros obtidos através da codificação anteriormente descrita.

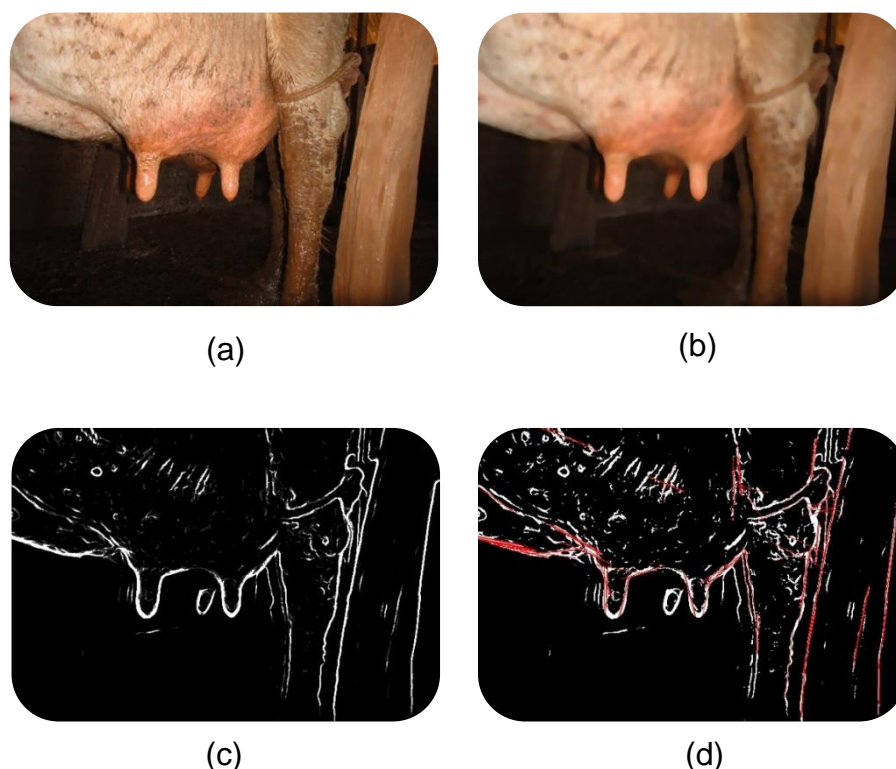


Figura 19 – Resultados dos Filtros

Fonte do autor.

Na Figura 19 (a) tem-se a imagem original. Primeiramente foi aplicado um filtro da Mediana com precisão 4, cujo resultado é o apresentado na Figura 19 (b). Em seguida, com esse resultado, aplicou-se o filtro de Sobel (Figura 19 (c)), para realçar as bordas. A partir dessa imagem, aplicou-se a transformada de Hough na imagem resultante. O resultado obtido pode ser visto na Figura 19 (d).

A utilização destes filtros mostrou-se inadequada pelo fato destes não indicarem com a precisão necessária os tetos. As causas para isto são várias e pode-se destacar: as variações de luminosidade do ambiente, o plano de fundo da imagem que pode causar ruídos atrapalhando a identificação dos tetos, os diferentes tipos de úberes e os diferentes tipos de tetos.

Após os testes com as imagens reais dos úberes, foram realizados os testes com o protótipo do úbere. A Figura 20 e 21 mostram os resultados.

A primeira abordagem de solução, com plano de fundo branco, foi a aplicação de um filtro de bordas. Inicialmente foi utilizado o filtro de mediana como um filtro suavizador para a redução de ruídos, seguido por filtro de Sobel realçando bordas, e, por fim, filtro de Hough, na tentativa de identificar elipses que possam representar os tetos do úbere.

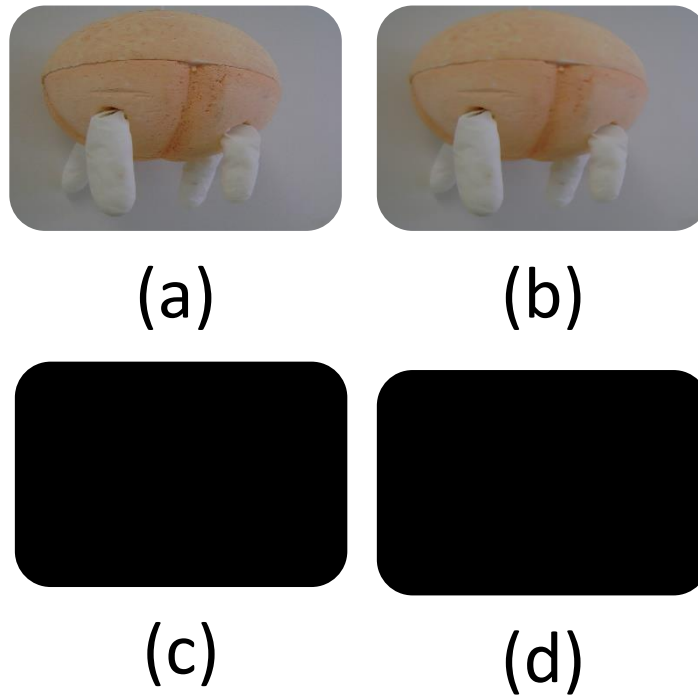


Figura 20 - Testes utilizando filtros com imagem do protótipo de um úbere
Fonte do autor.

Na Figura 20 o item (a) é a imagem original, item (b) a imagem com o filtro da mediana, item (c) com o filtro de Sobel e imagem (d) com o filtro de transformada de Hough. Nota-se que os testes não identificaram as bordas pelo filtro de Sobel nem linhas pelo teste de Hough.

Com os resultados desse teste, identificou-se a necessidade de utilizar um plano de fundo para a captura das imagens, de forma a evitar o ruído produzido pela interferência de outros elementos visuais presentes no ambiente.

Novos testes foram realizados – em outras imagens - modificando o plano de fundo a fim de aumentar a diferença entre as cores do plano de fundo e o protótipo de úbere para a identificação dos tetos.

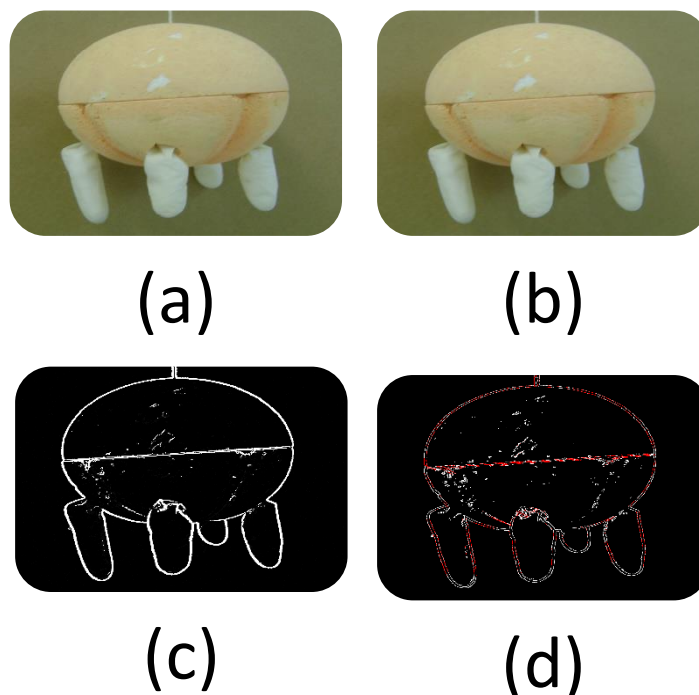


Figura 21 - Testes utilizando filtros com imagem do protótipo de um úbere
 Fonte do autor.

A Figura 21 (a) apresenta a imagem original. Na Figura (b) aplicou-se o filtro da mediana, na Figura (c) o filtro de Sobel e na Figura (d) o teste de Hough. Esses testes foram realizados em um ambiente controlado (com iluminação fixa, sem movimentação do úbere e um plano de fundo monocromático) e chegou-se à conclusão de que a utilização da biblioteca *filters* não é suficiente para a identificação dos tetos.

Além de a utilização dos filtros não ser suficiente para a identificação dos tetos ainda existia a dificuldade de encontrar a distância deste teto em relação a câmera. Uma possível solução seria a utilização de duas câmeras. Mas optou-se pela utilização do Kinect, pois este apresenta além de uma câmera RGB um sensor de profundidade. Detalhes serão apresentados no capítulo 5.

4.4 SEÇÃO MÓDULO AUTOMAÇÃO E ROBÓTICA

Segundo Silveira (2012, p. 2), a estrutura básica de programação do Arduino é constituída por blocos de funções que carregam outros blocos de funções escritas em linguagem C/C++. O primeiro bloco de funções é iniciado com a função `setup()` que é

chamada quando o programa inicia e é executada somente uma vez. Nessa função os pinos podem ser configurados e porta serial pode ser inicializada, por exemplo. O segundo bloco de funções é o `loop()`, onde seu conteúdo é repetidamente executado.

Para comandar o Robix é primeiramente feita a declaração dos servos, que são seis, e a definição dos pinos do Arduino onde estão conectados os motores. No bloco de funções `setup()` esses motores e servos são associados uns aos outros da seguinte maneira, por exemplo, `servo1.attach(M1)`, onde M1 seria o Motor 1 conectado ao pino 11. Depois disso a porta serial é iniciada com uma taxa de transmissão de 9600 bps (bits por segundo) usando o comando `Serial.begin(9600)`.

Em seguida, ainda dentro do bloco `setup()`, o Robix recebe comandos para iniciar em uma posição pré-definida. O comando informa o ângulo que cada servo deve atingir, por exemplo, `servo1.write(30)`.

No bloco de `loop()`, o Arduino, em modo de espera, aguarda comandos vindos pela serial e armazena uma série desses comandos em um vetor. Então, para cada posição desse vetor é armazenada uma letra entre *a* e *f* e mais um valor. As letras entre *a* e *f* dirão qual motor será movido e o valor dirá quantos graus o mesmo deve rotacionar.

Assim, quando o objetivo do código em C for posicionar o Robix em certo ponto, os ângulos de movimentação para cada servo devem ser enviados via serial para o Arduino. Ou seja, se o objetivo é que o servo 1 se mova 50 graus, o comando enviado via serial deverá ser "a50".

Um código em C foi desenvolvido para enviar dados via serial ao programa do Arduino, movendo o Robix conforme os ângulos informados. Os ângulos são definidos com o auxílio da função de cinemática que faz as conversões de coordenadas virtuais (obtidas através das imagens) para as coordenadas reais (posicionamento do manipulador robótico nos úberes).

Testes foram realizados a partir de um ponto desejado (como por exemplo algum dos pontos vermelhos indicados na Figura 22) a ser alcançado no plano 2D. O ponto desejado foi obtido através do programa GeoGebra (Figura 23) para se ter garantias que os pontos estão dentro da área de trabalho, ou seja dentro do espaço que o Robix pode alcançar. Os ângulos foram calculados pelo *Matlab* para alcançar o ponto desejado e em seguida informados no programa para mover o Robix. A seguir comparou-se o ponto desejado com o ponto alcançado.

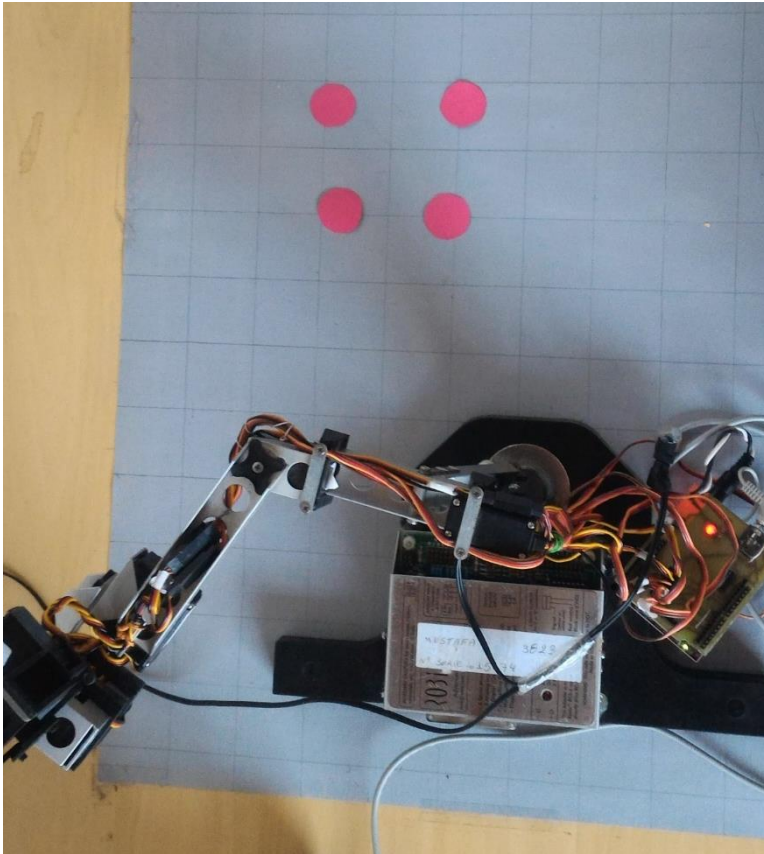


Figura 22 - Vista de cima das posições dos tetos do protótipo de úbere

Fonte do autor.

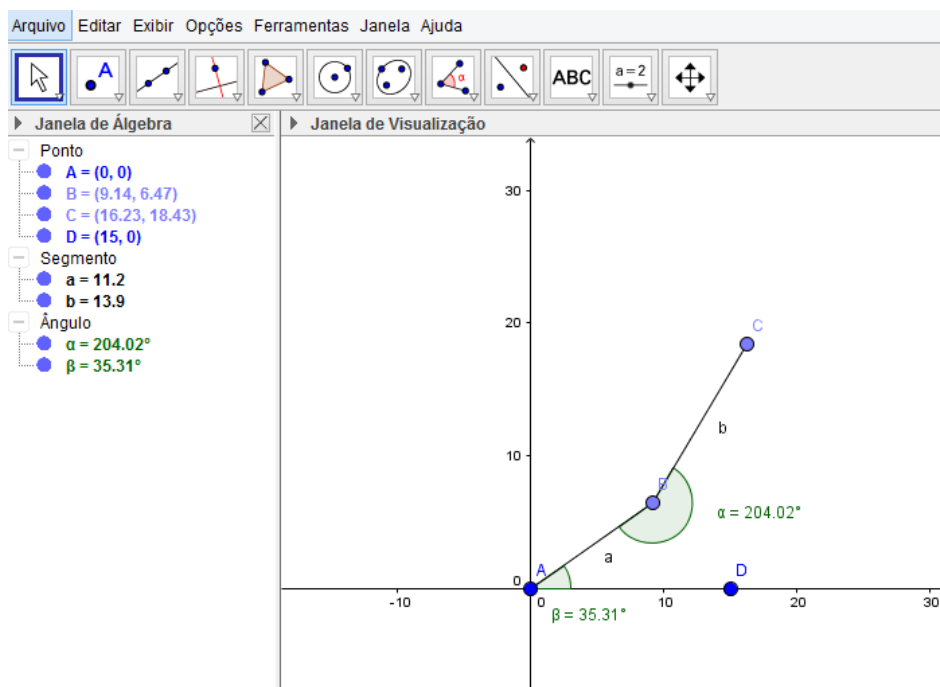


Figura 23 - GeoGebra, obtendo pontos desejados dentro da área de trabalho do Robix

Fonte do autor.

```

1 - x = 6.65;
2 - y = 24.2;
3 - l1 = 11.2;
4 - l2 = 13.9;
5 - D = ((x*x) + (y*y) - (l1*l1) - (l2*l2))/(2*l1*l2)
6 - theta2 = atand(-((sqrt(1-D*D)))/D)
7 - theta1 = atand(y/x) - atand(l2*sind(theta2)/(l1+l2*cosd(theta2)))

```

Figura 24 - Código do MatLab para obter ângulos

Fonte do autor.

Na Figura 24 tem-se o código utilizado para obtenção dos ângulos que as juntas precisam mover para alcançar os pontos desejados. Na primeira e segunda linha do código são inseridos o ponto desejado obtido pelo Geogebra. Na linha 6 muda-se o sinal do cálculo de theta 2 conforme o desejo de obter-se o cotovelo para cima ou o cotovelo para baixo. A tabela abaixo mostra os resultados do experimento.

Ponto desejado (x,y)	Ponto alcançado com cotovelo para cima	Ponto alcançado com o cotovelo para baixo
(16,16)	(18.5,11.7)	(10.4,11)
(24.4,0.01)	(21.4,1)	(21.4,-1.4)
(6.65,24)	(11.1,10.7)	(3.5,19.5)
(14.7,3.96)	(10.4,6)	(16.4,-2.1)
(22.59,4.02)	(19,-2.5)	(19.1,-0.8)
(24.27,-3.72)	(20.5,-7.4)	(-20.3,-9.3)
(21.26,-9)	(16.5,10.3)	(15,-15.4)
(19.51,-15.2)	(1.4,-16.9)	(11.5,-20)
(6,-22.27)	(2.3,-16.4)	(0.9,-16.4)
(17.67,-10)	(12.7,-9.4)	(10.1,-15)

Através desse experimento, nota-se que o Robix é impreciso para a construção do protótipo, tendo como base que o encaixe da teteira no teto deve ser preciso. A Figura 25 ilustra essa imprecisão.

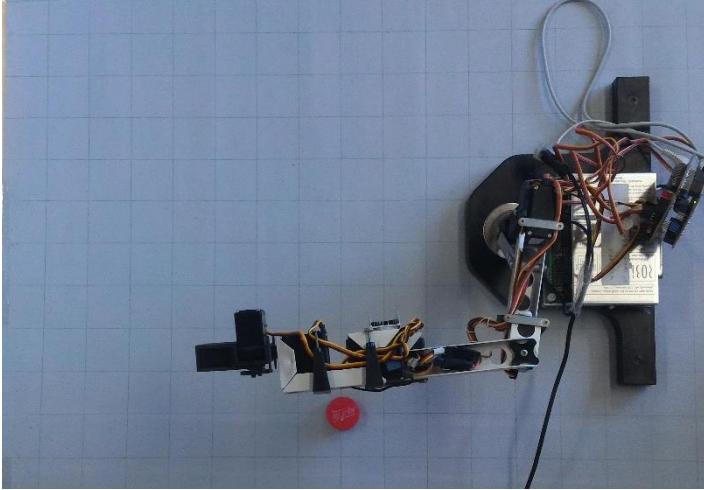


Figura 25 - Imprecisão do Robix
Fonte do autor.

5 INTEGRAÇÃO DOS MÓDULOS DO SISTEMA

Este capítulo detalha sobre aquisição de imagens, segmentação de imagens, processamento de imagens, profundidade e identificação dos tetos e a integração dos módulos deste trabalho.

Após os resultados insatisfatórios obtidos nos testes preliminares realizados com a câmera digital, para aquisição das imagens em tempo real optou-se pelo sensor de movimentos Kinect para Xbox 360.

O Kinect apresenta um kit de ferramentas com a documentação do SDK do Kinect, exemplos de códigos em várias linguagens para, como por exemplo: a captura de áudio; remoção de plano de fundo; básico de profundidade D2D com o Kinect no modo perto e modo longe; rastreamento de face; utilização de infravermelho; etc. Esse *toolkit*, ilustrado na Figura 26, contém além de outras ferramentas códigos de exemplos.

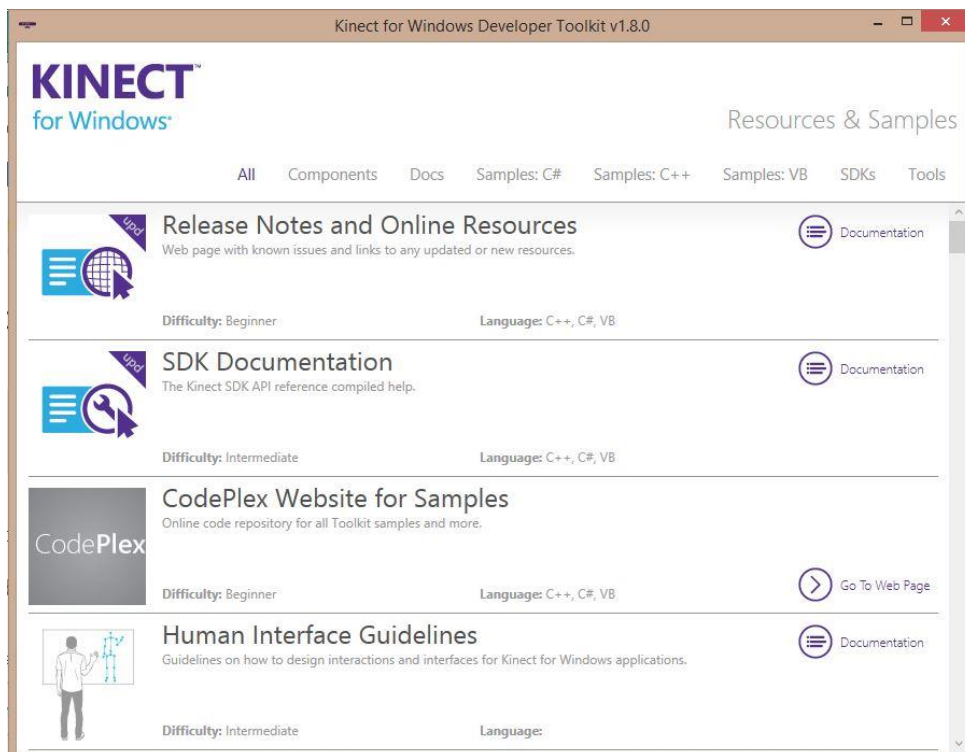


Figura 26 – Kit de desenvolvimento para Kinect

5.1 AQUISIÇÃO DE IMAGENS E SEGMENTAÇÃO DAS IMAGENS

Foram realizados experimentos com o Kinect para se destacar um objeto de interesse de uma imagem de fundo. Quando o objeto (mais próximo a câmera) se move em frente a câmera, seu corpo é combinado com a imagem de fundo, como mostra a Figura 27. Isso é possível devido aos dados de profundidade que são capturados pelo Kinect, e usados para escolher a cor do *pixel*. Se está mais à frente a cor do *pixel* na imagem reproduzida será a do objeto, caso contrário será a da imagem de fundo.

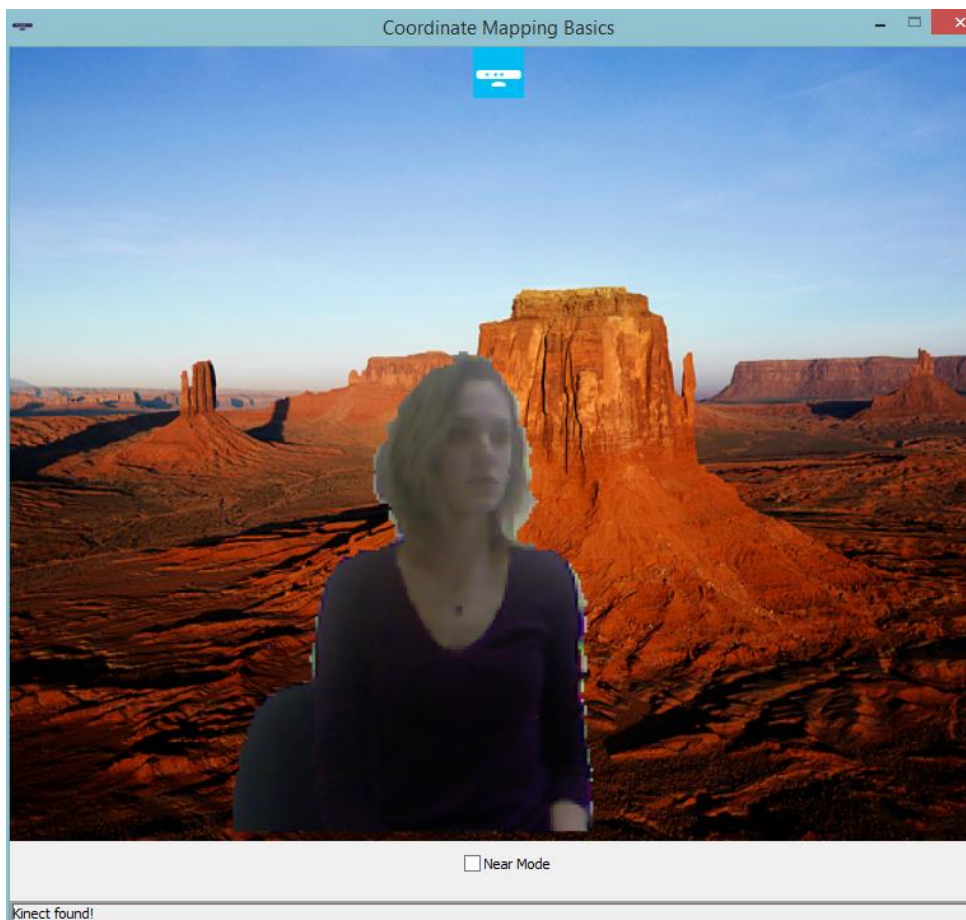


Figura 27 – Separação de objeto de interesse e plano de fundo
Fonte do autor.

Tendo como base esse exemplo, modificou-se o código para obter o resultado desejado. A primeira modificação feita no código foi a mudança da imagem de fundo para uma imagem de fundo preto. Segundo passo foi modificar a cor do jogador para tons de cinza. Este processo é de fundamental importância para se destacar no sistema final a região de interesse (os úberes) do fundo das imagens.

5.2 PROCESSAMENTO DE IMAGENS

Para transformar a imagem em tons de cinza utilizam-se dos valores da escala de cores RGB (*Red-Green-Blue*) de cada *pixel*. Com esses valores é possível realizar um cálculo que transforma cada *pixel* no tom adequado de cinza. Cada *pixel* da imagem então é percorrido, sendo avaliada sua profundidade para defini-lo como plano de fundo ou objeto de interesse (neste caso um jogador). O resultado é o mostrado na Figura 28.

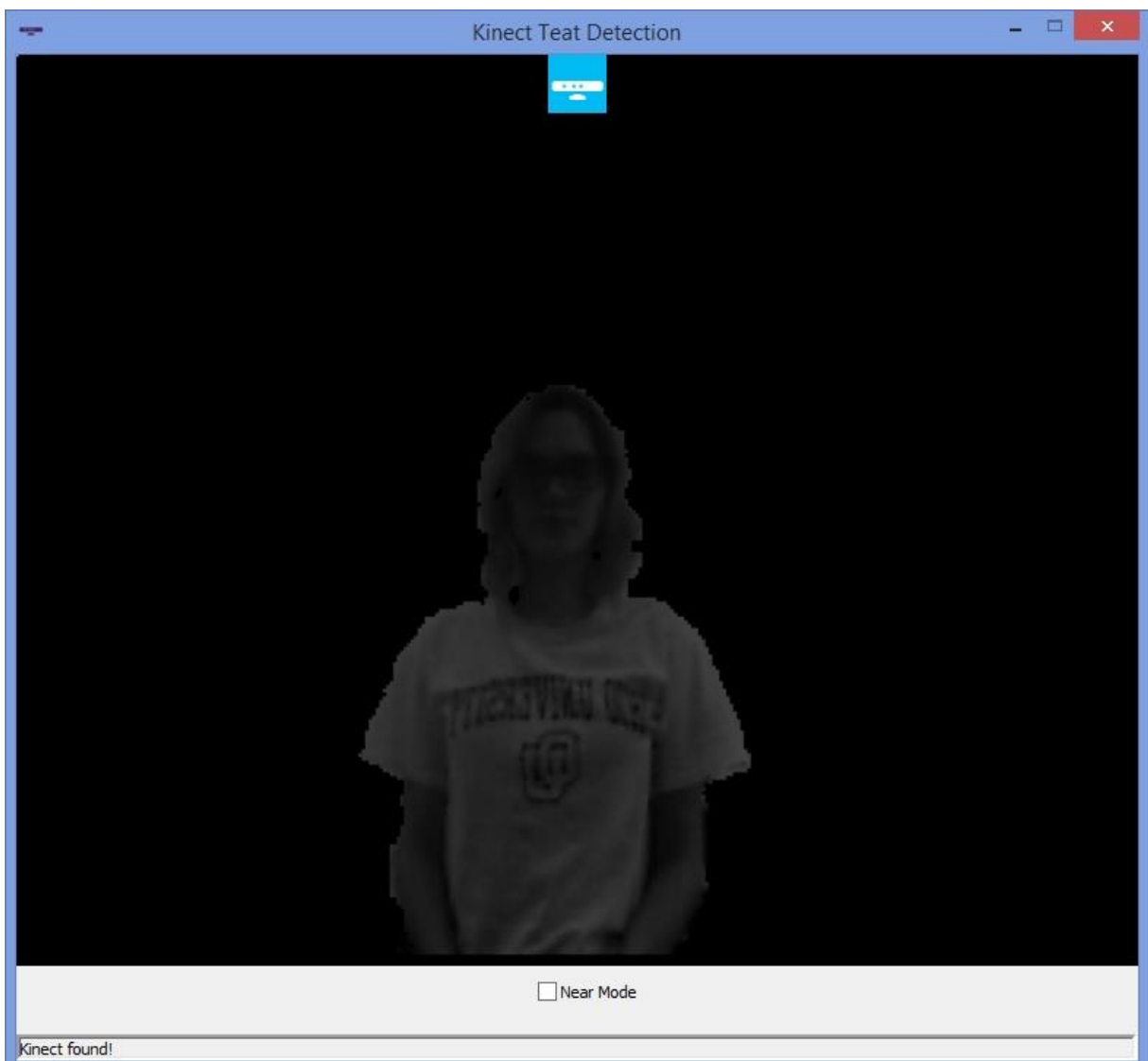


Figura 28 – Imagem com fundo preto e jogador em tons de cinza

Fonte do autor.

Na Figura 28 é possível notar que o jogador não contém muitos ruídos, portanto não foi necessário aplicar um filtro da mediana para suavizar a imagem.

5.3 PROFUNDIDADE

O próximo passo foi então, uma vez que já foi segmentado no vídeo (30 *frames* por segundo) o que é jogador e o que é fundo, obter a distância que esse jogador se encontra. A variável que armazena as distâncias em milímetros é chamada de *depth* (profundidade).

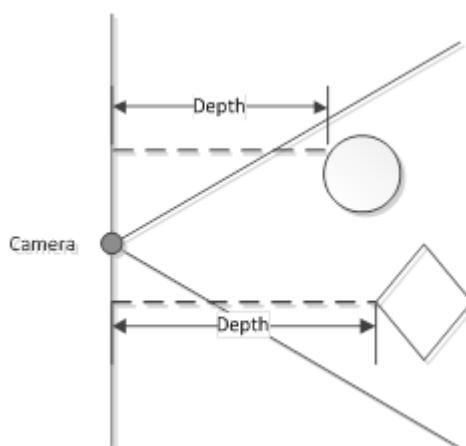


Figura 29 – Esclarecimento do modo de captura de distância realizado pelo Kinect

FONTE: <http://i.msdn.microsoft.com/dynimg/IC568991.png>

O Kinect captura as distâncias em relação ao plano em que se encontra como mostra a Figura 29.

O teste realizado para comparar se os dados do Kinect eram valores reais foi o uso de uma folha de papel Paraná em frente ao Kinect e fazer a medição através do Kinect com uma trena e comparar com a média dos valores obtidos no arquivo texto.

Através desse teste percebeu-se que o Kinect não tem uma precisão adequada, uma hipótese levantada é que as diferenças de problemas de precisão são causadas pela falta de calibração. A calibração recomendada pela Microsoft pode ser realizada com o uso de um controle do aparelho do Kinect Xbox 360, um cartão especial de calibração e um aparelho televisor.

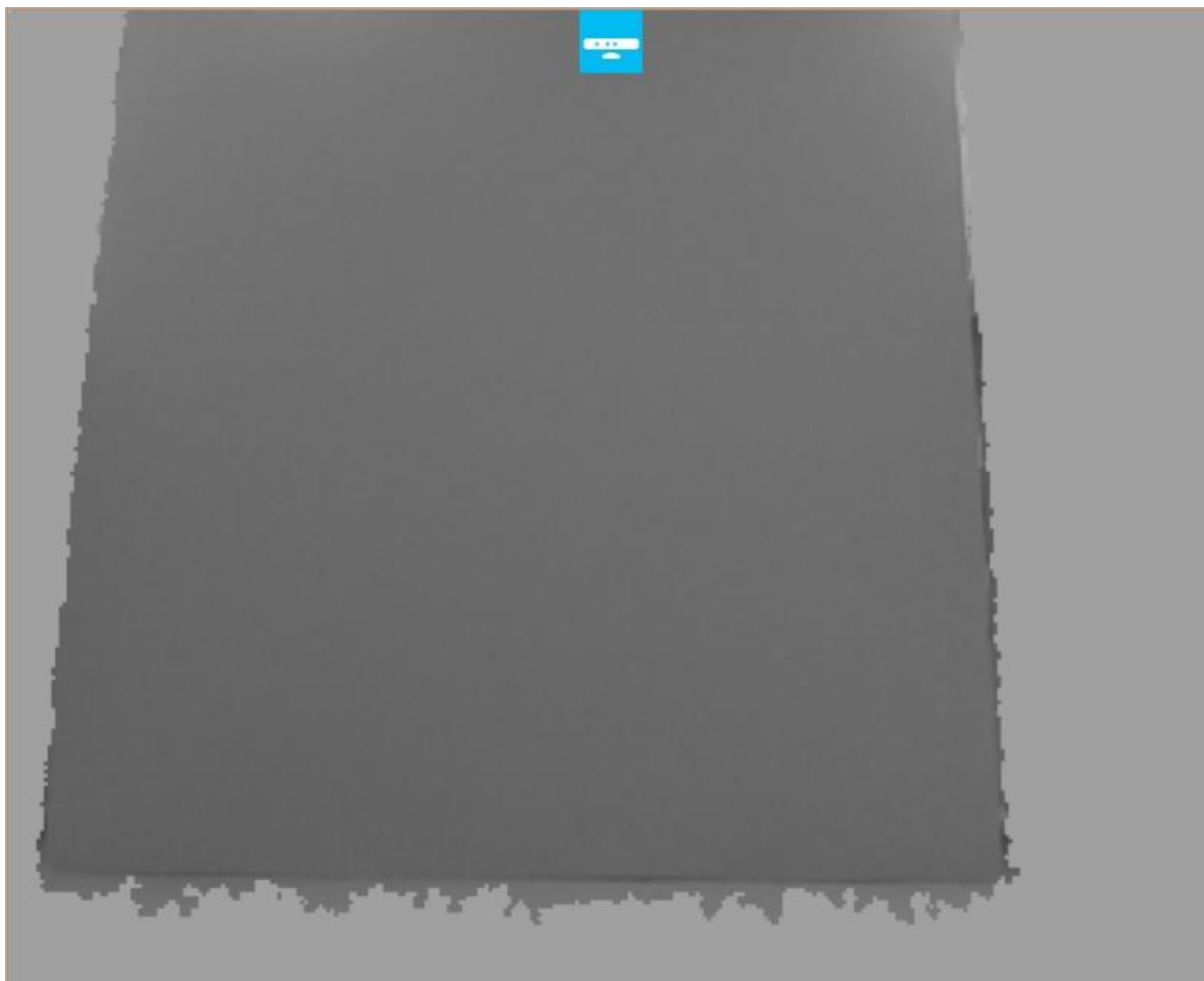


Figura 30 – Teste de profundidade com papel Paraná

Outra observação adquirida através desse teste é um *delay* para localizar o player, além de que ele não localiza primeiramente o papelão. Só foi possível obter essas imagens quando primeiramente o Kinect detectava a pessoa (apesar do rastreamento de pessoa estar desativado no código) que segurava o papelão para em seguida incluir o papelão na imagem (Figura 30).

5.4 IDENTIFICAÇÃO DOS TETOS

Existem dificuldades para realizar testes que envolvam animais em tempo real, pois o Kinect solicita como entrada o vídeo capturado em tempo real, uma vez que ele também utiliza os dados capturados pelo sensor de distância. Logo não se pode utilizar para os testes vídeos gravados ou imagens dos animais. Levando isso em

consideração foi desenvolvido um protótipo de um úbere para realizar os testes em laboratório, conforme Figura 31.



Figura 31 – Protótipo de úbere

Fonte do autor.

Com esse protótipo foi desenvolvido então uma metodologia simples para a identificação dos tetos. Essa consiste em percorrer todos os valores de y para cada coluna x da matriz e encontrar o maior valor de y , ou seja, o valor mais perto do chão de y . Lembrando que o fundo já foi removido, e o jogador está em tons de cinza, a ideia do algoritmo é que o que estiver mais próximo ao chão será candidato a teto.

O algoritmo de identificação dos tetos não utilizou forma nem cor pois os tetos podem apresentar formatos diferentes e também cores diferentes.

Assim como já havia sido feita a segmentação de fundo e o jogador em tons de cinza, para desenvolver o algoritmo basta percorrer o frame para cada coluna da largura da imagem iniciando em zero (esquerda para direita), e em seguida percorrer todos os pixels da altura da imagem que estão naquela coluna iniciando do maior valor de y (altura máxima da imagem). Inicia-se do maior valor de y pois por padrão o sistema de coordenadas de tela tem sua origem no canto superior esquerdo da tela.

Como os tons de cinza foram obtidos através de uma média aritmética, saber se o pixel do jogador pode ser facilmente diferenciado dos pixels do plano de fundo

através de uma comparação entre os tons de vermelho, verde, azul e alfa. Se os quatro valores forem idênticos significa que estes pixels pertencem ao jogador.

Em seguida, faz-se uma comparação se o valor y do pixel em questão é maior que a altura da imagem menos um limiar em pixels. Esta comparação é feita pois quanto mais próximo ao chão maior é o valor de y , e o limiar serve para eliminar possíveis pixels que estejam muito próximos ao chão como as pernas do animal.

Todas essas coordenadas dos pixels mais próximos ao chão foram armazenadas em um arquivo texto. Testes foram feitos para comparar se os pixels obtidos eram compatíveis com o que era apresentado em cada figura.

Tendo a imagem corretamente salva com bitmap, necessita-se então, finalmente comparar-se as coordenadas salvas nos arquivos como mais próximas ao chão com as coordenadas que estão no arquivo bitmap. Para visualizar a coordenada de um *pixel* na imagem usou-se o MATLAB R2012b.

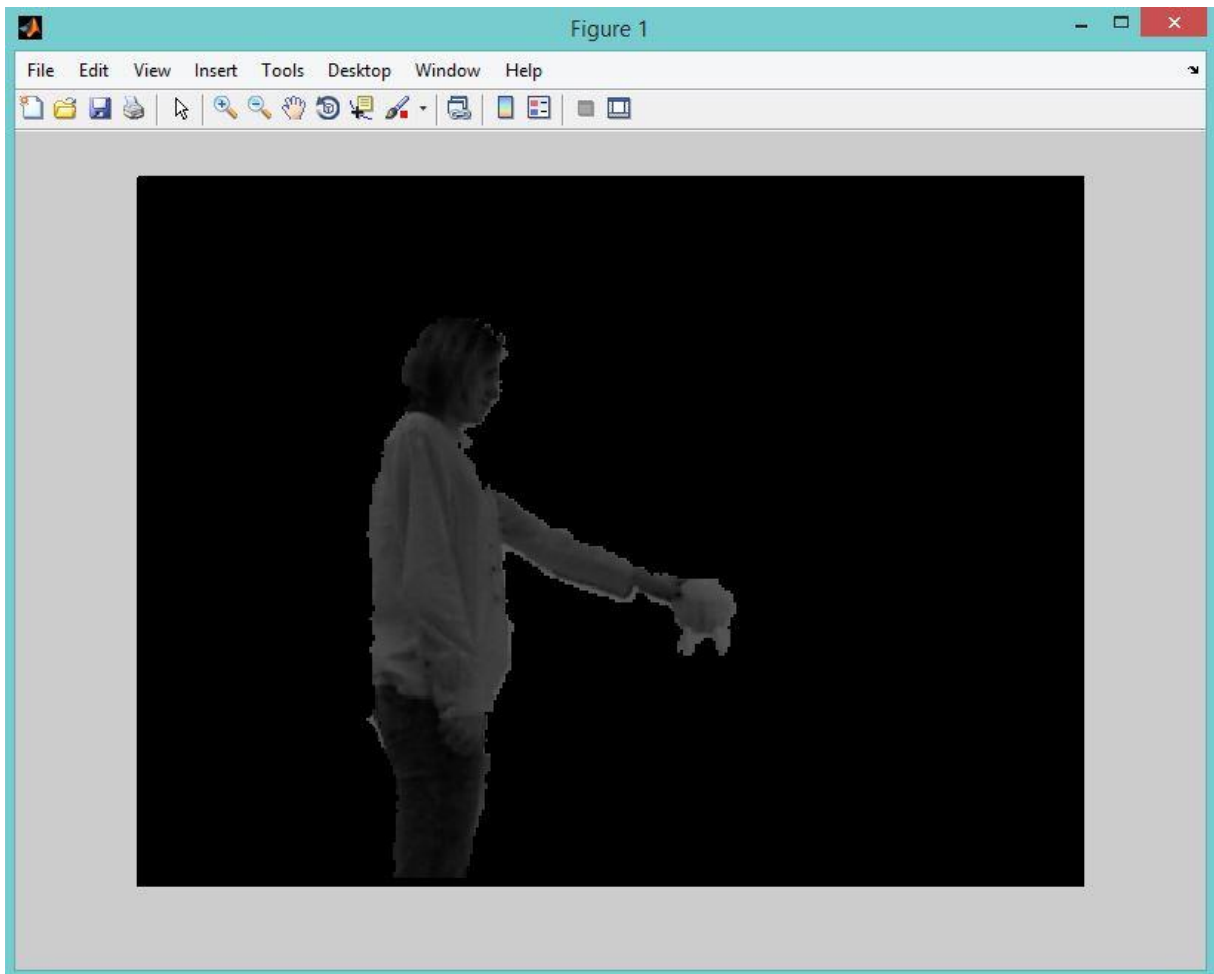
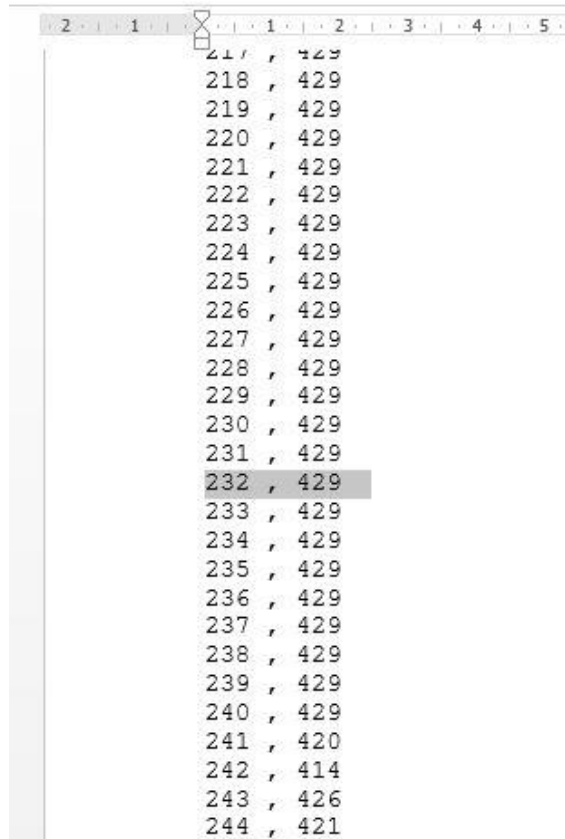


Figura 32 – Imagem aberta no MATLAB

Alguns testes foram realizados. A primeira coordenada escolhida no arquivo texto foi (232,429) pois haviam várias coordenadas cujo y era 429 como mostra a Figura 33.



217	, 429
218	, 429
219	, 429
220	, 429
221	, 429
222	, 429
223	, 429
224	, 429
225	, 429
226	, 429
227	, 429
228	, 429
229	, 429
230	, 429
231	, 429
232	, 429
233	, 429
234	, 429
235	, 429
236	, 429
237	, 429
238	, 429
239	, 429
240	, 429
241	, 420
242	, 414
243	, 426
244	, 421

Figura 33 – Coordenadas obtidas mais próximas ao chão

Essa coordenada encontra-se na região das pernas na Figura 34, ou seja, por ser uma curvatura cada os valores de x vão crescendo enquanto o valor de y permanece o mesmo. Logo esse seria um possível candidato a teto. Existem duas maneiras para melhorar essa solução. A primeira é aumentar o limiar y que elimina os pixels muito próximos ao chão. A segunda é usar um método que compare todos os valores de x quando o valor de y permanece o mesmo ou muito próximo. Se esses valores de x estão muito próximos só considerar como candidato a teto o próximo valor de y quando houver uma variação grande, ou considerável variação em x.

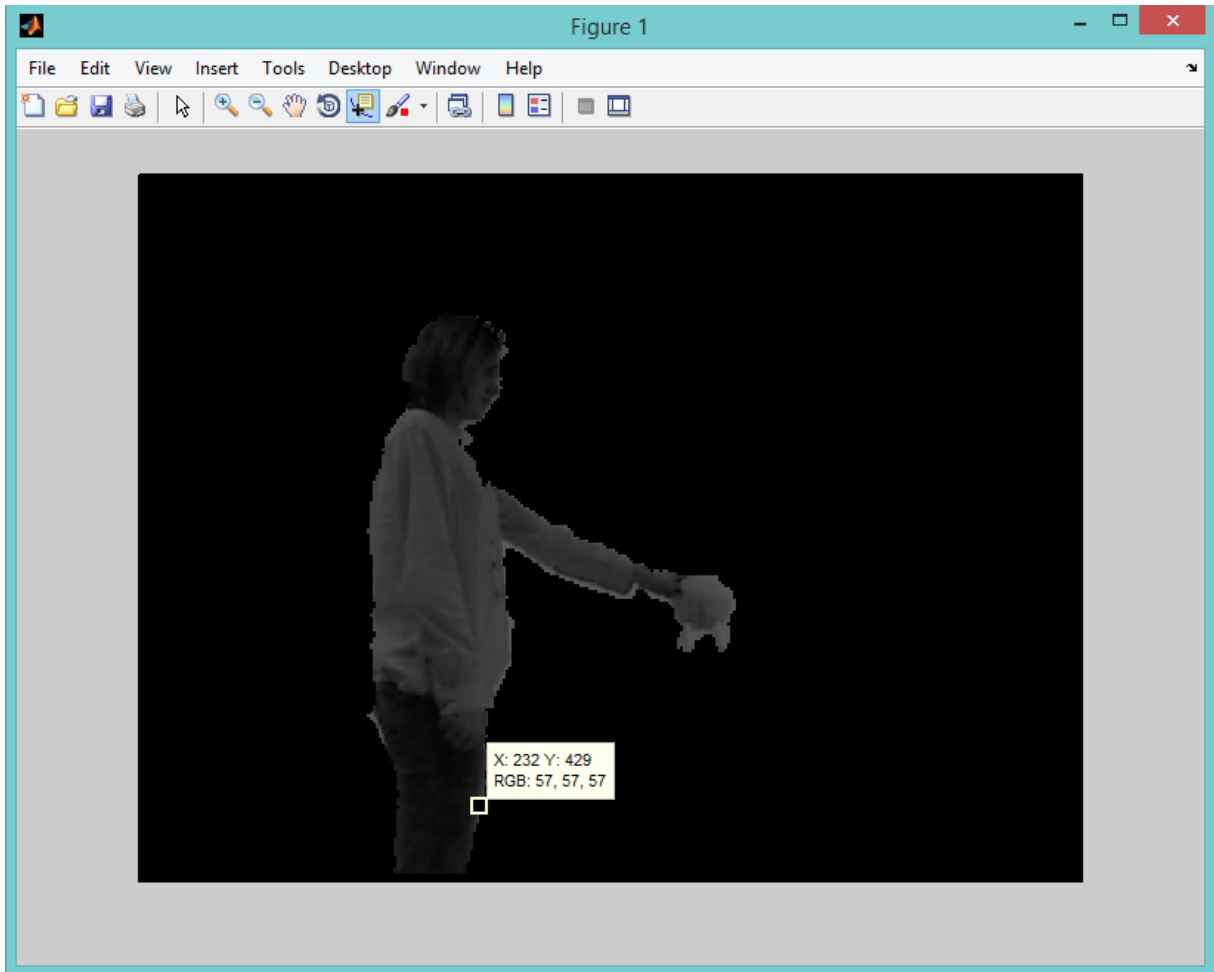


Figura 34 – Teste com coordenadas baseadas no arquivo texto

Um segundo teste foi então realizando dando um zoom somente na parte do protótipo do úbere conferindo suas coordenadas e comparando se realmente aquele pixel está dentro do arquivo texto anteriormente salvo. Dos dois tetos mais aparentes na imagem o teto mais à esquerda os dados obtidos no arquivo texto (Figura 35) não coincidem com os dados (Figura 36 e Figura 37). O teto mais à direita obteve resultados bem próximos comparando o arquivo texto com os dados do MATLAB. (Figura 38 e Figura 39).

8	367	, 342
9	368	, 341
10	369	, 341
11	370	, 342
	371	, 342
	372	, 338
	373	, 341
	374	, 341

Figura 35 – Valor das coordenadas próximo ao teto esquerdo

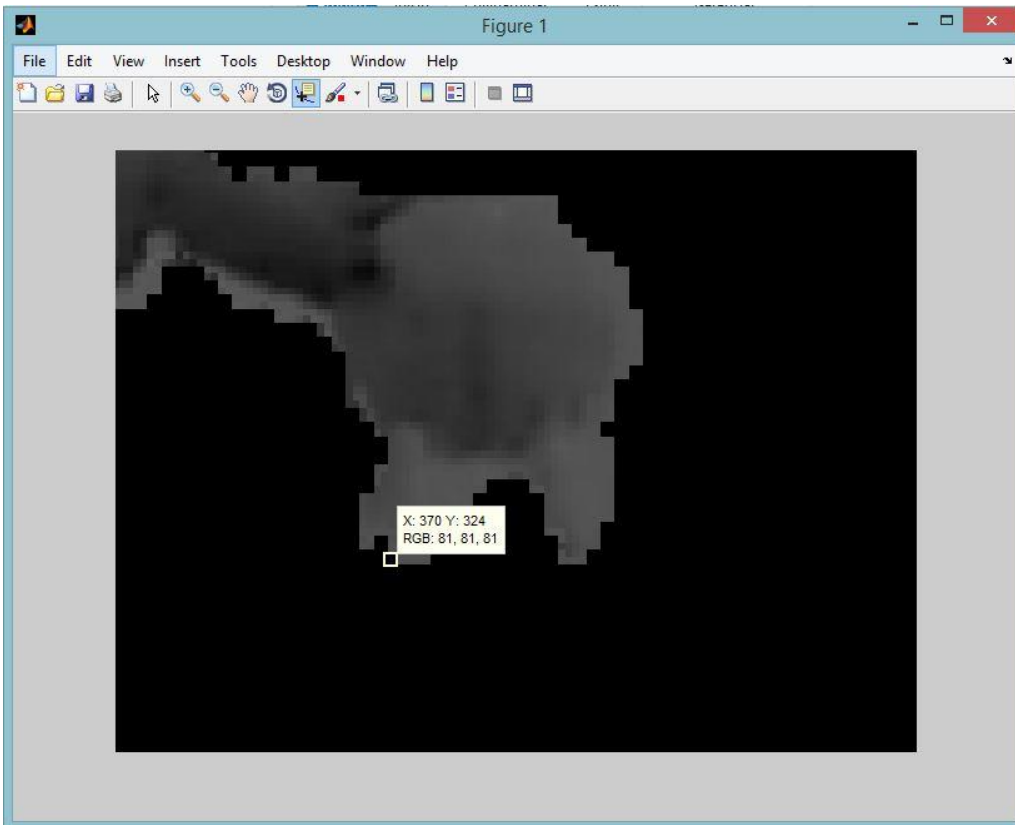


Figura 36 – Valor da coordenada do teto esperada no arquivo texto

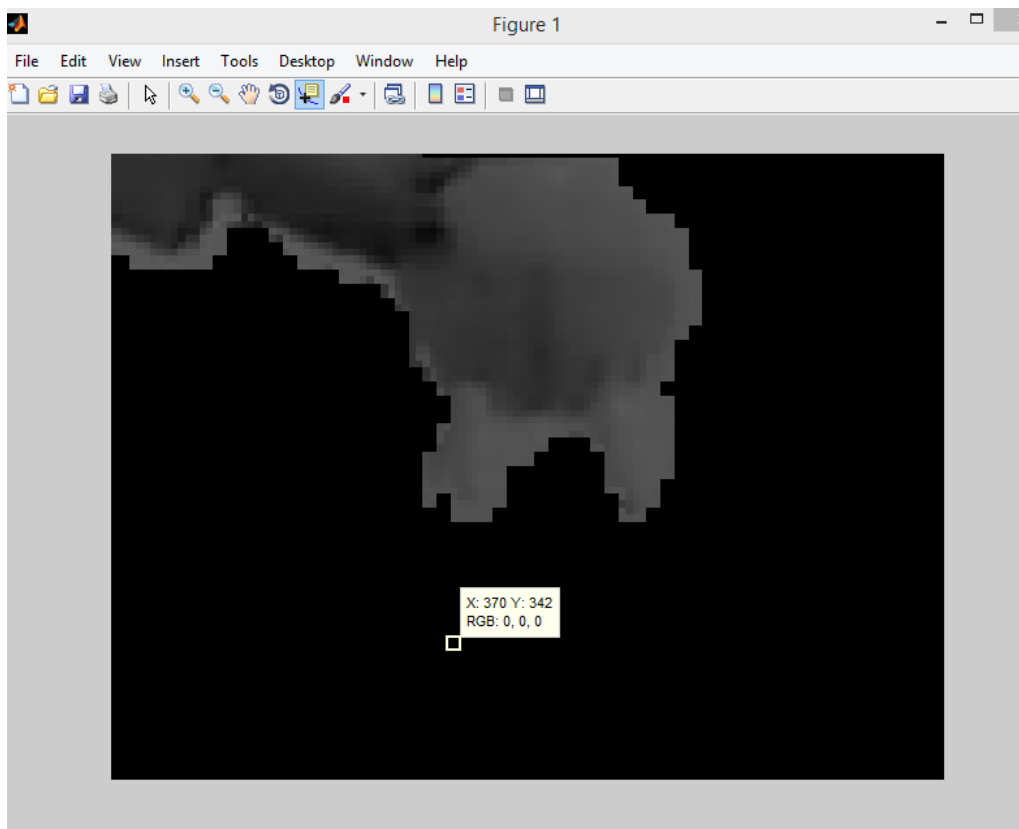


Figura 37 – Coordenada obtida no arquivo texto.

Comparando as Figuras 35, 36 e 37 nota-se que o valor de y esperado na coluna 370 da imagem seria próximo a 324, porém o valor de y capturado no arquivo texto foi 342. Um valor distante do esperado e com valores de RGB iguais a zero. A hipótese é que esse pixel tenha o valor de alfa também igual a zero.

```
392 , 344  
393 , 344  
394 , 344  
395 , 321  
396 , 343  
397 , 326  
398 , 326  
399 , 318  
400 , 321  
401 , 313
```

Figura 38 – Valor das coordenadas próximo ao teto direito

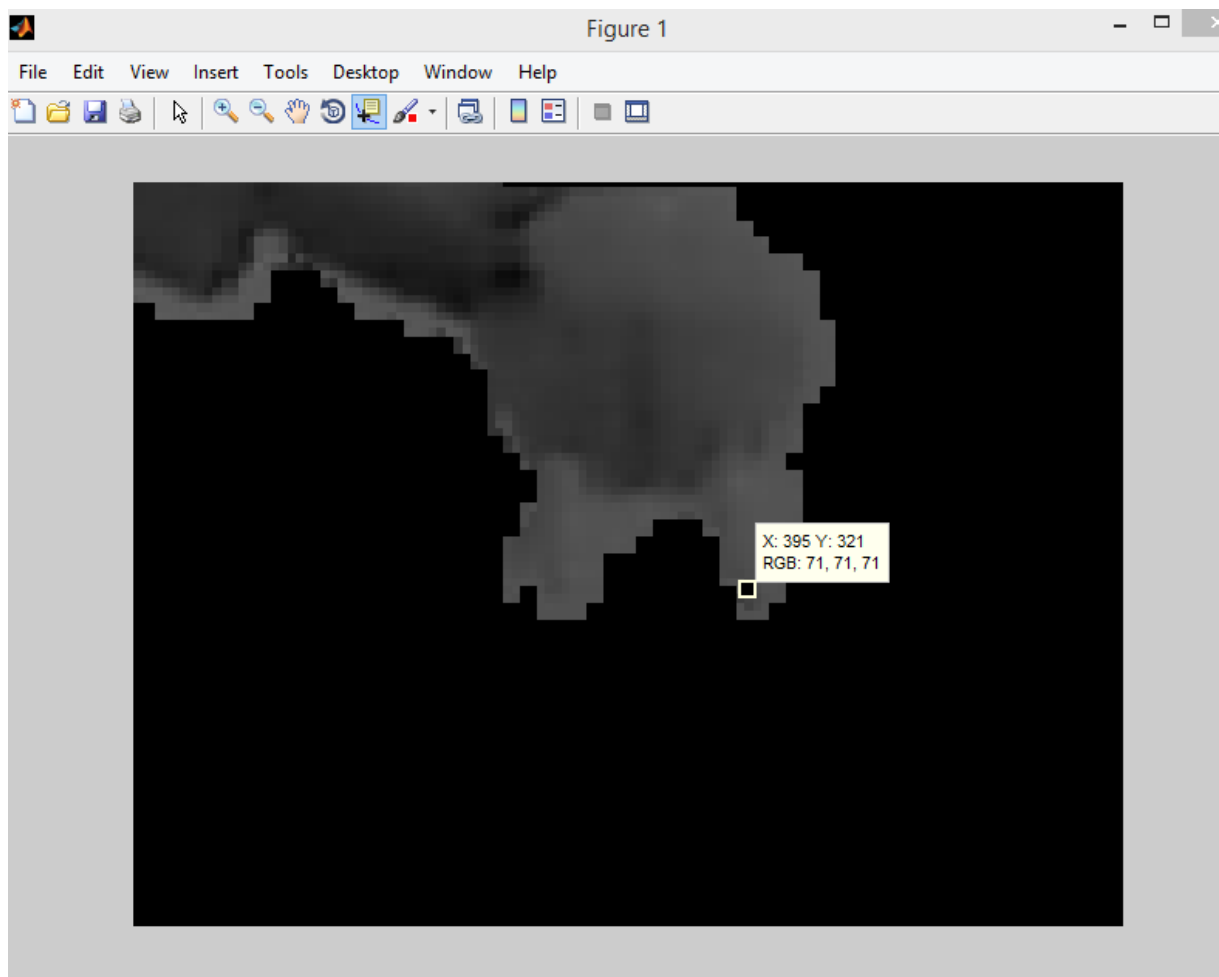


Figura 39 – Pixel encontrado no arquivo texto

Comparando a Figura 38 e a Figura 39 nota-se que o valor da coordenada encontrado no arquivo texto e no MATLAB são próximos e condizentes com um candidato a teto.

5.5 PROTÓTIPO DE ORDENHA

Através desses testes foram levantados os requisitos para a possível construção de um protótipo para a automação de uma ordenha para gado leiteiro. Para a construção de um protótipo faz-se necessário a utilização de um manipulador robótico milimetricamente preciso e ágil pois, por mais que o ambiente seja controlado o encaixe de uma teteira precisa ser milimetricamente preciso devido ao tamanho dos tetos. Este encaixe é baseado em coordenadas 3D que são uma resposta do sistema de visão. Para que o manipulador alcance as coordenadas faz-se necessário a calibração entre a câmera e o manipulador e o cálculo da cinemática inversa já realizada em testes no trabalho.

O processamento de imagens para a obtenção destas coordenadas envolve a utilização de imagens capturadas em tempo real e sensores, como mostraram os testes. Porém o processamento de imagens necessita de melhoramentos uma vez que estes identificam falsos tetos.

A solução também necessita de melhoramentos para a identificação dos tetos que ficam escondidos. Para isso possíveis soluções seriam a utilização de mais de uma câmera, ou acoplar a câmera na ponta do manipulador para a identificação dos tetos. Porém ambas soluções têm empecilhos.

A primeira solução necessita da descoberta das coordenadas z de cada teto (sua distância até o chão). Uma possível solução é a utilização de uma régua colocada no plano de fundo em relação ao animal e o Kinect. Desta maneira é possível saber quantos *pixels* equivalem a um centímetro pois a régua servirá como plano de projeção. Sabendo-se as coordenadas, (x,y,z) de um ponto do plano de projeção, através de um cálculo de semelhanças de triângulo encontra-se o ponto z do teto desejado.

O problema desta solução e da solução de acoplar a câmera na extremidade do manipulador é a sujeira que pode ser levada pelos animais tanto para a régua tanto para a câmera, logo será necessário a criação de um mecanismo de limpeza.

A Figura 40 apresenta a configuração do ambiente candidato à protótipo de automação de ordenha com os hardwares para a manipulação da ordenha (Figura 40 A - Robix) bem como o seu controlador (Figura 40 D - kit Arduino), o dispositivo para aquisição de imagens (Figura 40 C- Kinect) e o protótipo de úbere (Figura 40 B – protótipo de úbere) posicionado de forma similar ao que seria encontrado em um ambiente real de ordenha.

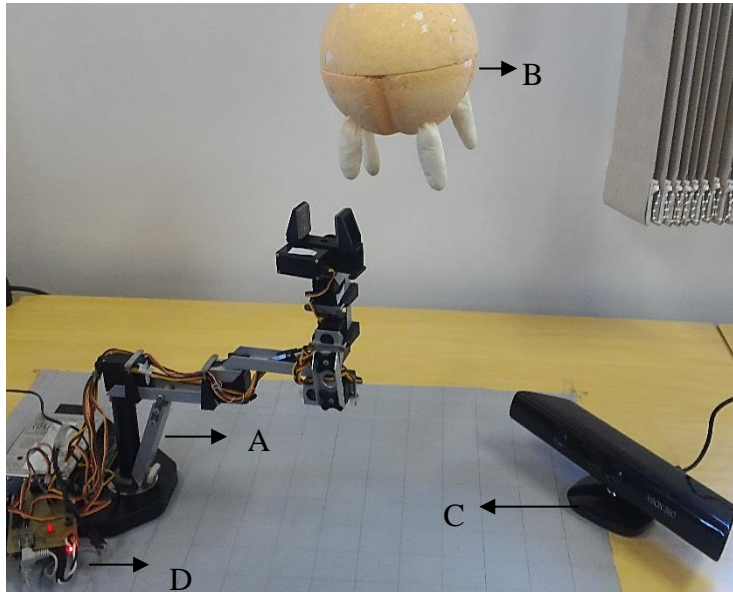


Figura 40 – Imagem ilustrativa do protótipo. A – Robix, B- protótipo do úbere, C –Kinect Xbox 360, D- Arduino.

Fonte do autor.

6 CONCLUSÃO

O trabalho proposto teve como objetivo utilizar visão computacional, processamento de imagens e controle dos movimentos de um braço robótico (cinemática direta e cinemática inversa), para levantar requisitos para o desenvolvimento de um protótipo de ordenha mecânica automatizada. Teve-se por objetivo levantar requisitos para apenas realizar a identificação dos tetos no úbere do animal, obter as coordenadas 2D dos mesmos e identificar a precisão necessária do manipulador robótica para o encaixe das teteiras no teto.

Os testes foram realizados em ambiente controlado, utilizando um protótipo de modelo sintético de úbere nos experimentos de posicionamento do manipulador mecânico. A proposta inicial para identificação dos tetos na imagem era a utilização de filtros para identificar os potenciais tetos na imagem. Entretanto, foi necessário realizar testes em tempo real, optando-se então pela utilização do Kinect. Através do Kinect foi possível localizar em um plano 2D os possíveis candidatos a tetos.

Embora identificados os possíveis tetos, os testes realizados com o Kinect apresentam falhas, como a apresentação de falsos tetos. Além disto faz-se necessário para o desenvolvimento do protótipo encontrar coordenadas 3D dos tetos e a utilização de mais de uma câmera para encontrar os tetos que ficam escondidos utilizando-se apenas uma câmera.

Quanto à obtenção das coordenadas espaciais (3D) dos tetos, os estudos iniciais mostraram que ela pode ser feita de diferentes maneiras, uma vez que pode se utilizar sensores de distância, ou ainda há a possibilidade do uso de duas ou mais câmeras para localizar as coordenadas espaciais dos tetos.

A partir da realização deste trabalho outros trabalhos propostos podem ser realizados de modo a aperfeiçoar a automatização de uma ordenha mecânica. Problemas como a identificação da teteira (insuflador), encaixe da teteira no teto, utilização de outra metodologia para a identificação dos tetos. Uma possível solução para a identificação de tetos é a utilização de um algoritmo para a extração de características de objetos e a geração de classificadores em cascata capaz de reconhecer padrões de objetos em uma imagem.

O levantamento de requisitos não leva em consideração mudanças no ambiente uma vez que, as vacas são capazes de realizar movimentos. Nem se leva

em consideração a alteração da luminosidade no ambiente e sujeira na lente da câmera. Além disso, as vacas podem ter tamanhos de úberes diferentes, e suas mamas mudam de formato durante o dia e durante a sua vida (KLOMP et al., 1990). Assim o trabalho levantou requisitos de possíveis maneiras de identificar tetos dentro de um ambiente controlado, sem permitir diferentes modelos de úberes e tetos. O trabalho levantou requisitos da precisão necessária da movimentação de um manipulador para a produção de um protótipo, assim como a utilização da cinemática inversa.

REFERÊNCIAS

BECKER, Djeily T. **Visão Computacional Aplicada em um Braço robótico Antropomórfico Didático**. Faculdade Assis Gurgagz, Cascavel, 2013.

CAMBRON, Mark E.; NORTHRUP, Steven G. **Calibration of a Pole-Mounted Camera Using a Neural Network**. In : 38th Southeastern Symposium on System Theory, Tennessee Technological University, p. 265-269, 2006.

DELAVAL. **DeLaval inaugura 1º VMS da América Latina**. Disponível em <<http://www.delaval.com.br/Sobre-a-DeLaval/DeLaval-Newsroom/?nid=107230>>. Acessado em 14 de Abril de 2014.

DEPARTAMENTO DE ECONOMIA RURAL. **Produção de leite estadual - 2012**. Fonte SEAB (Secretaria da Agricultura e do Abastecimento do Paraná).

ECKERT, John P. The Integration of Man and Machine. **IEEE Journals & Magazines**, p. 612 , 1962.

FERREIRA, Ricardo et al. Comando de um servomecanismo usando visão computacional. **Revista Robótica** ISSN: 0874-9019, Publindústria, nº 54, p. 16-19, Março de 2004.

FISHER, Robert B. Is Computer Vision still AI? **AI Magazine** , p.21-7, Summer, 1994.

FILHO, Ogê M.; NETO, Hugo V. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.

GALHANO, Alexandra M. S. F. **Uma Abordagem Estatística à Modelização de Manipuladores Robóticos**. Tese de Doutorado. Faculdade de Engenharia da Universidade do Porto, Departamento de Engenharia Eletrotécnica e de Computadores, 1992.

GEA WestfaliaSurge. **Sistema Multibox**. Disponível em <http://www.westfalia.com/br%5C/pt/bu/milking_cooling/automatic_milking/multibox_systems_mione/default.aspx>. Acessado em 03 de Agosto de 2014.

G1. **Produtor de leite utiliza ordenhadeira comandada por robô no Paraná**. Disponível em <<http://g1.globo.com/economia/agronegocios/noticia/2013/09/produtor-de-leite-utiliza-ordenhadeira-comandada-por-robo-no-parana.html>>. Acessado em 14 de Abril de 2014.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de Imagens Digitais**. Editora Edgard Blucher, 2000.

GOMES, Jonas; VELHO, Luiz. **Fundamentos da computação gráfica**. Rio de Janeiro: IMPA, 2008.

GUERRA, Rodrigo S. **Calibração automática de sistemas de visão estéreo a partir de movimentos desconhecidos**. Tese de Mestrado Universidade Federal do Rio Grande Do Sul, 2004, Tese de mestrado.

HAYKIN, Simon. **Redes neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2007. 900 p. ISBN 8573077182.

HEYDEN, Anders; ASTROM, Kalle. Flexible Calibration: Minimal Cases for Auto-Calibration. In: Seventh IEEE International Conference on, **Proceedings...** Greece, Kerkyra, 1999, p. 350-355.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Indicadores do IBGE –Produção de origem animal, leite – 2014**. Disponível em <<http://www.sidra.ibge.gov.br/bda/tabela/protabl.asp?c=74&z=t&o=24&i=P>> Acessado em 25 de Abril de 2016.

JAMUNDÁ, Teobaldo. **Reconhecimento de Formas: A Transformada de Hough**. Seminário Visão Computacional – CPGCC, Universidade Federal de Santa Catarina, 2000. Disponível em: < <http://www.inf.ufsc.br/~visao/2000/Hough/> >. Acessado em 05 de Agosto de 2014.

JUNIOR, Cairo L. N.; YONEYAMA, Takashi. **Inteligência Artificial em Controle e Automação**. Editora Edgard Blücher, 2010.

KLOMP, Cornelis et al. Development of an Autonomous Cow-Milking Robot Control System. **IEEE Journals & Magazines**, p. 11-19, 1990.

LOAIZA, Manuel E.; RAPOSO, Alberto B.; GATATASS, Marcelo. **Calibração de múltiplas câmeras baseado em um padrão invariante**. In : XII Symposium on Virtual and Augmented Reality Natal, RN, Maio de 2010, p 212-221.

MATERICK, Craig A. **Mapping Robotic Movement to a Three-Dimensional Coordinate System**. Honors Projects, Illinois Wesleyan University , 1997.

MENEZES, Paulo R. et al. **Introdução ao Processamento de Imagens de Sensoriamento Remoto**. CNPQ, Universidade de Brasília , 2012.

MILANO, Danilo de; HONORATO, Luciano Barrozo. **Visão Computacional**. Monografia. Universidade Estadual de Campinas, 2009.

ORLANDINI, Guilherme. **Desenvolvimento de Aplicativos Baseados em Técnicas de Visão Computacional para Robô Móvel Autônomo**. Tese de Mestrado. Universidade Metodista de Piracicaba, Piracicaba, SP, 2012.

ÖZUYSAL, Mustafauysal .**Manual and Auto Calibration of Stereo Camera Systems**. Tese de Mestrado , Middle East Technical University, 2004.

PAZOS, Fernando. **Automação de sistemas & Robótica**. Editora Axcel Books do Brasil , 2002.

PINHEIRO, Hayanne S.;NASCIMENTO José. F. L.; QUEIROZ-NETO José P. **Simulador de Cinemática Direta de um Robô Didático (Robix) em Ambiente Labview**. In: V CONNEPI , 2010.

ROSSATO, Daniel B.; **Desenvolvimento de um Sistema Aberto para Ensino de Robôs Manipuladores**. Tese de Mestrado. Universidade de São Paulo, Departamento de Engenharia de Telecomunicações e Controle, Escola Politécnica, São Paulo, 2009.

SEGUNDO, José S. A. de S. **Um Método para Determinação da Profundidade Combinando Visão Estéreo e Autocalibração para Aplicação em Robótica Móvel**. Tese de Mestrado. Universidade Federal do Rio Grande do Norte, Natal , RN, Abril de 2007.

SHHEIBIA, Tarig Ali Abdurrahman El. **Controle de um Braço Robótico Utilizando Uma Abordagem de Agente Inteligente**. Tese de Mestrado. Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, Coordenação de Pós-Graduação em Informática, Campina Grande, PB, Julho de 2001.

SILVA, João C. S et al. **Modelagem Cinemática de um Robô Manipulador**. In: XXXIV COBENGE, Universidade de Passo Fundo, 2006.

SILVEIRA, João A. da; **Arduíno – Cartilha para programação em C**. Disponível em: <<http://www.revistadoarduino.com.br>>. Acesso em 04 de Agosto de 2014.

SIMON, Steffen et al. Object classification with simple visual attention and hierarchical neural network for subsymbolic-symbolic coupling. In : Computational Intelligence in Robotics and Automation, **Proceedings IEEE International Symposium on** . p. 244-249 , 1999.

SZELISKI, Richard. **Computer vision: algorithms and applications**. New York: Springer, 2011.

THORMÄLLEN, Thorsten; BROSZIO, Hellward; MIKULASTIK, Patrick. **Robust Linear Auto-Calibration of a Moving Camera from Image Sequences**. University of Hannover , ACCV, 2006.

TSAI, Lung-wen. **Robot analysis: The mechanics of serial and parallel manipulators**. Maryland: Wiley-interscience, 1999.