

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS PATO BRANCO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

EDUARDO DOS SANTOS

**O USO DE VISÃO COMPUTACIONAL PARA O CONTROLE DE UM
MANIPULADOR ROBÓTICO**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2014

EDUARDO DOS SANTOS

**O USO DE VISÃO COMPUTACIONAL PARA O CONTROLE DE UM
MANIPULADOR ROBÓTICO**

Trabalho apresentado à disciplina de Trabalho de Conclusão de Curso do Curso de Engenharia de Computação da UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Engenheiro de Computação.

Orientador: Prof. Dr. Marco Antonio de Castro Barbosa

Coorientadora: Prof.^a Dr.^a Gilda Aparecida de Assis

PATO BRANCO
2014



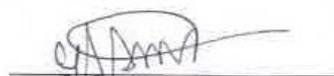
MINISTÉRIO DA EDUCAÇÃO
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Engenharia de Computação



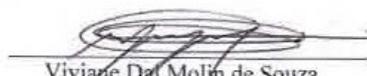
TERMO DE APROVAÇÃO

Às 10 horas do dia 14 de agosto de 2014, na sala V107, UTFPR, Câmpus Pato Branco, reuniu-se a banca examinadora composta pelos professores Marco Antonio de Castro Barbosa (orientador), Gilda Aparecida de Assis (coorientador), Vinicius Pegorini e Viviane Dal Molin de Souza para avaliar o trabalho de conclusão de curso com o título **O uso de visão computacional para o controle de um manipulador robótico**, do aluno **Eduardo dos Santos**, matrícula 1114891, do curso de Engenharia de Computação. Após a apresentação o candidato foi arguido pela banca examinadora. Em seguida foi realizada a deliberação pela banca examinadora que considerou o trabalho aprovado.


Marco Antonio de Castro Barbosa
Orientador (UTFPR)


Gilda Aparecida de Assis
Coorientador (UTFPR)


Vinicius Pegorini
(UTFPR)


Viviane Dal Molin de Souza
(UTFPR)


Beatriz Terezinha Borsoi
Coordenador de TCC


Marco Antonio de Castro Barbosa
Coordenador do Curso de
Engenharia de Computação

RESUMO

SANTOS, Eduardo dos. O uso da visão computacional para o controle de um manipulador robótico. 2014. 60 f. Monografia (Trabalho de Conclusão de Curso) - Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2014.

Manipuladores robóticos surgiram para auxiliar na automatização de processos industriais, e devido a essa característica, eles se tornaram elementos amplamente explorados na robótica, o que também incentiva pesquisas na área. Grande parte dos manipuladores utilizados atualmente estão limitados a movimentos sequenciais cuja malha de controle é diretamente estruturada pelo engenheiro. Qualquer mudança em sua área de trabalho pode resultar em problemas, sendo que um grande desafio nesta área tem sido fazer com que o manipulador possa perceber mudanças no ambiente e tomar decisões sobre qual ação será realizada, tomando como base informações coletadas no ambiente físico. Nesse contexto, foi desenvolvido um jogo computacional de estratégia simples entre o manipulador robótico e um jogador humano, o qual integra visão computacional para identificação de objetos e sua localização no espaço e cinemática inversa para que o manipulador possa executar os movimentos corretamente. A solução de visão computacional adotada utilizou o processo de segmentação baseado em cor para distinguir os objetos de interesse. Para a movimentação do manipulador adotou-se um ponto base fixo no tabuleiro, de forma que todas as posições foram calculadas a partir deste ponto âncora.

Palavras-chave: Manipular mecânico. Segmentação de imagens. Processamento de imagens.

ABSTRACT

SANTOS, Eduardo dos. Integrating Computer Vision and Robotic manipulator. 2014. 60 f. Monografia (Trabalho de Conclusão de Curso) - Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2014.

Robotic manipulators can support the automation of industrial processes, and due to this, they became broadly used in robotics, which also encourages research in the field. Currently, the majority of the robotic manipulators are limited to sequential movements whose control mesh is directly structured by the engineer. Any changes in its workstation can lead to problems, and a major challenge in this area has been making the manipulator capable of perceiving changes in the real environment and make decisions about which action will be held, based on the information collected in the physical environment. In this context, a simple computational strategy game between the robotic manipulator and a human player was developed, which integrates computer vision for the identification of objects and its location in space and inverse kinematics so that the manipulator can carry out the movements correctly. The adopted computer vision solutions were the color-based segmentation to distinguish the objects of interest. For the moving of the manipulator a fixed root point (anchor point) was adopted in the board, so that all the positions were calculated from that anchoring point.

Keywords: Mechanical manipulator. Image segmentation. Image processing.

LISTA DE FIGURAS

Figura 1 - Configuração de manipulador mecânico	11
Figura 2 - Tipos de robôs	13
Figura 3 - Manipuladores com coordenadas: (a) cartesiana; (b) cilíndrica; (c) esférica; (d) revolução.....	15
Figura 4 - 4a. N4(p)- 4-vizinhança e 4b. N8(p) - 8-vizinhança.....	17
Figura 5 - Filtro da média.....	20
Figura 6 - Aplicação do filtro de Sobel.....	21
Figura 7 - Histograma	21
Figura 8 - Transformada de Hough de linhas.....	22
Figura 9 - Robô hidráulico - HYD - 2800.....	23
Figura 10 - Robô elétrico - S400	24
Figura 11 - Robô pneumático - BOADICEA	24
Figura 12 - Etapas de execução do sistema.....	27
Figura 13 - Materiais e equipamentos.....	28
Figura 14 -Arquitetura do sistema	31
Figura 15 - Etapas do módulo de processamento de imagem	31
Figura 16 - Eliminação da marca d'agua.....	32
Figura 17 - Etapas do módulo de visão computacional	32
Figura 18 - Etapas do módulo de tomada de decisão	34
Figura 19 - Etapas do modulo de cinemática inversa	35
Figura 20 - Layout físico da área de trabalho (Vista superior)	37
Figura 21 - Vista superior a uma altura de 60 e 110cm	38
Figura 22 - Software GeoGebra.....	40
Figura 23 - Software para captura das imagens.....	41
Figura 24 - Marca adicionada na imagem.....	41
Figura 25 - Aplicação dos filtros a uma imagem da área de trabalho	42
Figura 26 - Aplicação dos filtros a uma imagem do tabuleiro com algumas peças ...	42
Figura 27 - Numeração das células do tabuleiro	43
Figura 28 - Imagem original e peças vermelhas.....	44
Figura 29 - Imagem original e peças azuis.....	44
Figura 30 - Imagem original e tabuleiro	45
Figura 31 - Imagem original e peças verdes	45

Figura 32 - Imagem original e tabuleiro	46
Figura 33 - Imagem original e tabuleiro	46
Figura 34 - Imagem original e peças vermelhas.....	47
Figura 35 - Imagem original e peças verdes	47
Figura 36 - Saída tabuleiro	48
Figura 37 - Imagem original e processada com pontos de saída.....	48
Figura 38 - Imagem original e processada	49
Figura 39 - Identificação das peças contidas no tabuleiro	49
Figura 40 - Peças externas ao tabuleiro	50
Figura 41 - Estado do tabuleiro e próximas peças a serem utilizadas	51
Figura 42 - Tabuleiro sem peças.....	51
Figura 43 - Execução do experimento 1	52
Figura 44 - Eixos	53
Figura 45 - Diferença no resultado de duas execuções.....	54
Figura 46 - Peças com sombreamento	56
Figura 47 - Posição da garra.....	56

SUMÁRIO

1 INTRODUÇÃO/CONTEXTUALIZAÇÃO	9
1.1 Problema e Proposta	10
1.2 Justificativa	11
1.3 Objetivo Geral.....	11
1.4 Objetivos Específicos	12
2 REVISÃO DA LITERATURA	12
2.1 Estudo sobre as características do robô	13
2.1.1 Definição de Robô	13
2.1.2 Servo-motores.....	14
2.1.3 Coordenadas Generalizadas	14
2.1.4 Graus de liberdade	14
2.1.5 Espaço de trabalho	15
2.1.6 Anatomia dos manipuladores	15
2.1.7 Cinemática.....	16
2.2 Estudo sobre processamento de imagens	16
2.2.1 Imagens digitais.....	16
2.2.2 Processamento de imagens.....	17
2.2.3 Vizinhança	17
2.2.4 Visão computacional	18
2.2.5 Aquisição da imagem	18
2.2.6 Pré-processamento	18
2.2.7 Segmentação	19
2.2.8 Extração de características	19
2.2.9 Reconhecimento e interpretação.....	19
2.2.10 Base de conhecimento.....	19
2.2.11 Filtragem de imagens	19
2.2.12 Suavização de imagens no domínio espacial.....	20
2.2.13 Detecção de bordas.....	20
2.2.14 Histograma	21
2.2.15 Transformada de Hough.....	22
2.3 Estado da arte.....	22

2.3.1 Acionamento Hidráulico	23
2.3.2 Acionamento Elétrico.....	23
2.3.3 Acionamento Pneumático	24
3 MATERIAIS E MÉTODOS	26
3.1 Espaço de estados (Tabuleiro)	27
3.2 Tecnologias	28
3.3 Recursos de Hardware.....	28
3.4 Recursos de Software.....	29
4 DESCRIÇÃO DO SISTEMA	30
4.1 Arquitetura do sistema	30
4.1.1 Módulo de processamento de imagem.....	31
4.1.2 Módulo de visão computacional.....	32
4.1.3 Módulo de tomada de decisão.....	33
4.1.4 Modulo de cinemática inversa	34
5 RESULTADOS	36
5.1 Área de trabalho do manipulador	36
5.2 Cinemática inversa.....	38
5.3 Experimentos Preliminares.....	40
5.4 Testes de Validação	43
5.5 Experimento de movimentação das peças.....	51
5.6 Experimento manipulador x humano	55
6 CONCLUSÕES.....	57
7 REFERÊNCIAS BIBLIOGRÁFICAS	58

1 INTRODUÇÃO/CONTEXTUALIZAÇÃO

A robótica vem cada vez mais sendo utilizada em diversas áreas, dentre elas destacam-se a indústria, a segurança e a saúde. Na saúde, por exemplo, os manipuladores mecânicos têm sido utilizados em cirurgias minimamente invasivas, como descrito no trabalho de (LORENZO; CAMPERCHIOLI; GASPARI, 2007), no qual um manipulador robótico é utilizado para auxiliar cirurgias em procedimentos cirúrgicos complexos.

A indústria é o local em que a robótica traz um grande número de aplicações, pois os robôs podem executar tarefas de forma automática, rápida e eficiente (PAZOS, 2002). Os robôs auxiliam também na segurança, monitorando áreas, identificando objetos, entre outras funções.

Os sistemas robóticos podem interagir com o meio de diferentes maneiras, uma delas é manipulando objetos, a qual é amplamente utilizada em linhas de produção, em locais insalubres, onde existam altas temperaturas, situações de risco a pessoas, em operações que necessitem de movimentos precisos e possuam restrições temporais.

“O uso de robôs dotados de visão artificial em tarefas, tais como, controle de qualidade em linhas de produção, aumenta a cada ano, num cenário de crescente automação industrial. Diversas outras áreas tão distintas como Astronomia, Segurança e Direito – para citar apenas algumas – vêm sendo beneficiadas com os avanços nas áreas de processamento de imagens e visão por computador.” (MARQUES FILHO; VIEIRA NETO, 1999).

Dentre os vários tipos de sistemas robóticos, existe o manipulador mecânico, que possui estas mesmas características, e apresenta como principal objetivo mover objetos (PAZOS, 2002).

Para que se torne possível o deslocamento de objetos, um robô manipulador possui em sua estrutura física um dispositivo conhecido como efetuador, que possui como função operar sobre o objeto a ser manipulado. A maioria dos manipuladores tem uma garra como efetuador e são montados sobre uma base fixa. Os movimentos que o manipulador pode executar são determinados pelos tipos das juntas e o número de graus de liberdade é determinado pelo número de juntas/articulações, sendo que, quanto maior o número de juntas, maior será a dificuldade para controlá-lo (PAZOS, 2002).

Algumas áreas da computação agregaram contribuições à robótica auxiliando no avanço desta área, tornando possível o aumento das aplicações e da confiabilidade. Dentre elas, pode-se citar o processamento de imagens, que consiste na captura das imagens e seu

tratamento com o objetivo de melhorar a informação visual para interpretação humana ou melhorá-la para percepção/interpretação automática através de máquina. Outra área é a visão computacional, que consiste na extração de informações de imagens, como por exemplo, localização de objetos e identificação de alterações no ambiente, fazendo o robô “enxergar” o ambiente de trabalho (MARQUES FILHO; VIEIRA NETO, 1999).

Segundo Gomes e Velho (2008), a computação gráfica é uma forma de *interação homem-máquina*, que proporciona aplicações em diversas áreas de conhecimento, permitindo visualizar objetos que ainda se encontram em fase de projeto, que estão fora de alcance de nossa percepção visual e que são abstratos, ou seja, não existem na realidade física, tornando possível realizar uma análise rápida de dados complexos.

Também contribuindo com a robótica tem-se a inteligência artificial, que permite que os robôs sejam capazes de comportamentos inteligentes usando dados adquiridos a partir de imagens, sons e outras fontes de informações. (SHHEIBIA, 2001). A inteligência artificial pode ser definida como a capacidade de um sistema ser racional, tomando assim decisões corretas com base nos dados que possui (RUSSELL; NORVIG, 2004), ela é utilizada em diversas aplicações, dentre elas os sistemas especialistas, concebidos para atuar como consultores qualificados em uma determinada área do conhecimento (NASCIMENTO JUNIOR; YONEYAMA, 2000).

1.1 Problema e Proposta

Percebendo as limitações do manipulador robótico propõe-se desenvolver um sistema de visão computacional, para que o mesmo possa interagir com o ambiente e se adaptar a mudanças. Por exemplo, considere uma linha de produção na qual peças de diferentes formas e tamanhos trafegam através de uma esteira e podem se apresentar nos mais diversos ângulos, inclusive em posições incorretas ou estarem defeituosas.

Utilizando-se um sistema de visão computacional, seria possível identificar o defeito, podendo assim tomar uma decisão de acordo com a situação/cenário que se apresenta e é nessa problemática que esse trabalho se insere.

O manipulador a ser utilizado neste trabalho é articulado, com acionamentos elétricos e possui uma garra do tipo pinça como poder ser vista na Figura 1. Para incorporar a visão computacional será utilizada uma câmera digital, que possibilitará ao robô ter uma visão do ambiente de atuação e com isso será possível tomar decisões e realizar as ações necessárias.

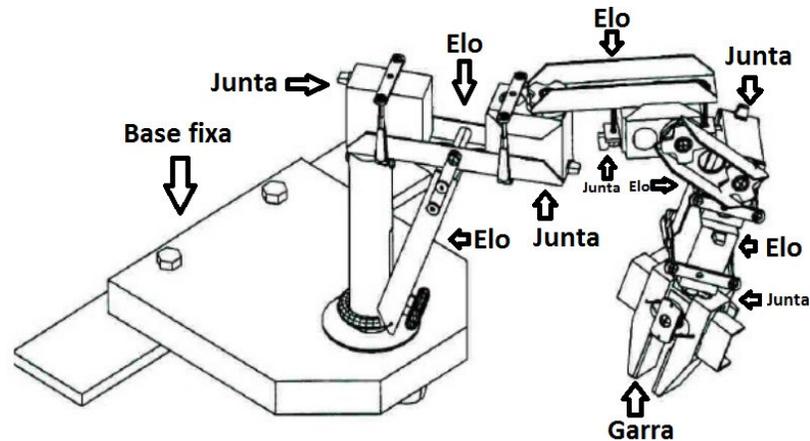


Figura 1 - Configuração de manipulador mecânico

Fonte: The Iris ... (2013).

O manipulador mecânico possui juntas que se movem de forma semelhante aos movimentos de um braço humano, por este motivo muitas vezes é chamado de braço robótico (SHHEIBIA, 2001).

1.2 Justificativa

Conforme dito anteriormente, os sistemas robóticos possuem diversas aplicações, porém muitos deles estão limitados a movimentos sequenciais predefinidos ou necessitando do controle humano, ou seja, eles não possuem a capacidade de identificar mudanças no ambiente e tomar decisões.

Neste contexto, propõe-se desenvolver um jogo computacional tendo como jogadores o manipulador robótico e um ser humano, explorando assim os recursos do manipulador mecânico através de uma aplicação motivadora e desafiadora para os alunos de Engenharia de Computação do Campus de Pato Branco da UTFPR, promovendo o embate Homem x Máquina.

1.3 Objetivo Geral

Utilizar um sistema de visão computacional para que o sistema robótico possa perceber o ambiente e realizar as ações com base em um algoritmo de decisão.

E desenvolver um algoritmo capaz de realizar a conversão de coordenadas de imagem para coordenadas físicas.

1.4 Objetivos Específicos

O protótipo da aplicação será a interação de um humano com o manipulador robótico através do clássico jogo da velha, no qual o robô será controlado pelas técnicas propostas, com o intuito de vencer o jogo. Para a realização do projeto será necessário agregar conhecimento de diferentes áreas, dentre elas: computação gráfica, visão computacional, inteligência artificial e automação e controle. Segue o detalhamento das atividades desenvolvidas.

- Projetar e construir um sistema de hardware e software para aplicação em um ambiente controlado;
- Pré-processar as imagens de entrada;
- Segmentar, detectar e reconhecer os objetos nas imagens de entrada;
- Aplicar a cinemática inversa para a localização e alocação dos elementos do jogo;
- Realizar testes de validação;
- Documentar o projeto.

2 REVISÃO DA LITERATURA

Neste capítulo são apresentados os principais assuntos estudados para o desenvolvimento do trabalho. Primeiramente são apresentados os principais tipos de robôs e seus principais componentes físicos e anatomia. A seguir são descritos os principais conceitos de processamento de imagens utilizados, incluindo técnicas de segmentação de imagem e filtros para processamento no domínio espacial.

2.1 Estudo sobre as características do robô

A revisão de literatura descrita a seguir, referente às características do robô, foi realizada com base no livro de Pazos (2002). Nesta seção são definidos os principais elementos dos manipuladores mecânicos (servo-motores, coordenadas generalizadas, graus de liberdade, espaço de trabalho e cinemática).

2.1.1 Definição de Robô

Para este termo existem diversas definições, cada uma com um ponto de vista diferente. A definição de robô segundo a Associação das Industrias da Robótica é que um robô industrial é um manipulador reprogramável, multifuncional, projetado para mover materiais, peças, ferramentas ou dispositivos especiais em movimentos variáveis programados para a realização de uma variedade de tarefas. Os robôs podem ser divididos em quatro categorias, como mostrado na Figura 2.

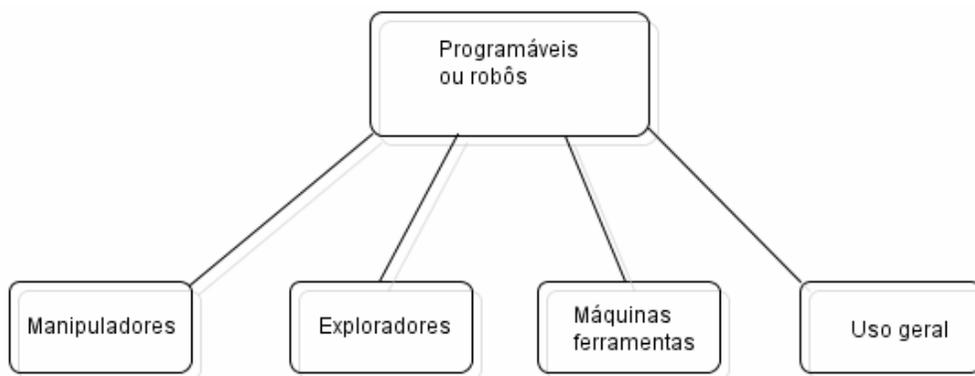


Figura 2 - Tipos de robôs

Fonte: Adaptado de Pazos (2002).

A primeira classe de robôs é a classe dos robôs manipuladores, visam deslocar objetos de um ponto para outro do espaço. Na segunda classe tem-se os exploradores, que possuem como objetivo explorar um determinado ambiente e através de sensores extrair as características físicas dele, um exemplo dessa classe é o robô enviado a Marte para monitoramento da superfície do planeta. A terceira classe é a das máquinas-ferramenta, que tem por objetivo processar uma matéria prima, aumentando o seu valor agregado, os exemplos envolvem robôs de solda, furadeiras de controle numérico, etc. Por fim tem-se a quarta classe, os de uso geral, na qual podem se citar os controladores de temperatura, que mesmo não realizando movimentos também são considerados robôs.

2.1.2 Servo-motores

Os servo-motores são os motores mais utilizados na robótica. Eles são motores de corrente contínua, com um sensor de posição ou de velocidade, permitindo assim que o controlador conheça as grandezas físicas e possa controlá-las. A grande maioria dos servo-motores comerciais exigem como entrada de referência um sinal pulsado, no qual a largura do pulso corresponde a posição desejada, esse sinal é conhecido como sinal modulado por largura de pulso.

2.1.3 Coordenadas Generalizadas

As coordenadas generalizadas são descritas a partir das variáveis características das juntas que representam o movimento relativo de um elo com respeito ao anterior, e através destas coordenadas é possível se estabelecer a posição do efetuador.

2.1.4 Graus de liberdade

O número de graus de liberdade é dado pelo número de juntas que o manipulador possui. Dependendo da aplicação é necessário uma quantidade maior de graus de liberdade, no entanto quanto maior for esse número, mais complexo se torna o seu controle.

2.1.5 Espaço de trabalho

O espaço de trabalho é o espaço dentro do qual o manipulador pode movimentar o efetuador e é definido como o volume total conformado pelo percurso do extremo do último elo, o punho, quando o manipulador efetua todas as trajetórias possíveis, onde a posição do punho é representada por um vetor de três componentes (x, y, z) .

2.1.6 Anatomia dos manipuladores

Existe uma grande diversidade de anatomias de manipuladores, cada forma se destina a uma aplicação específica, e essas configurações físicas são caracterizadas de acordo com as três primeiras coordenadas generalizadas que estão relacionadas com as três primeiras juntas, classificando assim em quatro configurações básicas: coordenadas cartesianas, coordenadas cilíndricas, coordenadas esféricas ou polares e coordenadas de revolução (Figura 3).

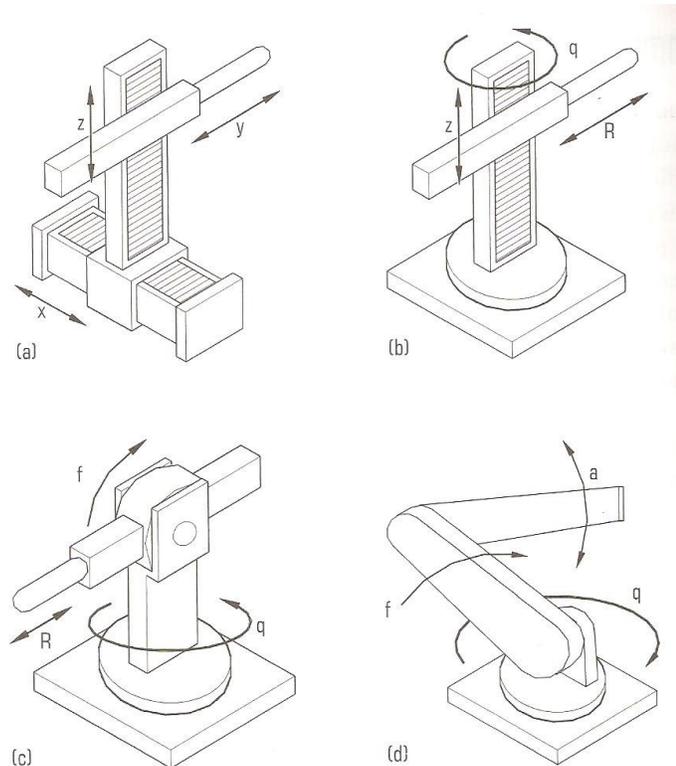


Figura 3 - Manipuladores com coordenadas: (a) cartesiana; (b) cilíndrica; (c) esférica; (d) revolução

Fonte: Pazos (2002).

2.1.7 Cinemática

A maioria das tarefas que utilizam manipuladores consiste em levar o efetuador de uma determinada posição para outra. Para que isso ocorra, é necessário que sejam calculados os ângulos e distâncias que as juntas devem se deslocar para que o efetuador atinja a posição desejada. Tais problemas são estudados pela cinemática inversa do manipulador, que encontra as coordenadas generalizadas a partir da posição desejada para o efetuador. Já a cinemática direta ocupa-se do problema contrário, ela encontra a posição do efetuador a partir das coordenadas generalizadas.

2.2 Estudo sobre processamento de imagens

Para que seja possível o processamento de uma imagem, é necessário que a mesma esteja em formato digital. A revisão de literatura descrita a seguir apresenta os principais conceitos relacionados ao processamento de imagens e foi realizada com base no livro de Marques Filho e Vieira Neto (1999).

2.2.1 Imagens digitais

A imagem digital pode ser considerada como uma matriz regular bidimensional de elementos denominados *pixels*, na maioria dos casos tais *pixels* são considerados quadrados. Um *pixel* pode representar um ou mais valores, que podem ser interpretados por diversos sistemas de cores, ou simplesmente o *pixel* pode ser um índice para uma tabela de cores. Por exemplo, uma imagem obtida a partir de uma câmera digital é uma matriz com dimensões correspondentes a sua resolução, por exemplo 640x480, com três bandas que representam a intensidade dos componentes vermelho, verde e azul, ou seja, três números tipicamente entre 0 e 255 para cada *pixel*.

As imagens podem ser armazenadas em diferentes formatos, dentre eles, TIFF (*Tagged Image File Format*), PNG (*Portable Network Graphics*), JPEG (*Joint Photographic Experts Group*) e outros, cada um possuindo algumas limitações e vantagens. Por exemplo, o

formato JPEG, causa perda de precisão na representação dos pixels, e o formato TIFF permite armazenar imagens com pixels com mais de quatro bandas e/ou valores de ponto flutuante.

2.2.2 Processamento de imagens

O processamento de imagens vem sendo objeto de crescente interesse, por permitir viabilizar um grande número de aplicações em duas categorias: O aprimoramento de informações pictóricas para interpretação humana; e a análise automática por computador de informações extraídas de uma cena. O processamento de imagens digitais é implementado normalmente por softwares sob forma algorítmica, mas em situações nas quais as limitações do computador principal forem relevantes, o processamento de imagens pode ser realizado com o uso de hardware especializado.

2.2.3 Vizinhança

É um importante conceito que define os *pixels* próximos em processamento de imagens. Por exemplo, um *pixel* p de coordenadas (x, y) , possui 4 vizinhos, horizontais e verticais, com coordenadas $(x+1, y)$, $(x-1, y)$, $(x, y+1)$ e $(x, y-1)$, os quais, formam a chamada “4-vizinhança” de p , também denotada por $N_4(p)$. O *pixel* p possui também 4 vizinhos diagonais, com coordenadas $(x-1, y-1)$, $(x-1, y+1)$, $(x+1, y-1)$ e $(x+1, y+1)$, que formam o conjunto $N_d(p)$. A união dos conjuntos $N_4(p)$ e $N_d(p)$ formam a chamada “8-vizinhança”,

definida também por $N_8(p) = N_4(p) \cup N_d(p)$. Os tipos de vizinhança são ilustrados na Figura 4.



Figura 4 - 4a. $N_4(p)$ - 4-vizinhança e 4b. $N_8(p)$ - 8-vizinhança

Fonte: Drakos e Moore (2002).

2.2.4 Visão computacional

A visão computacional é responsável por fazer o robô “enxergar” o ambiente. Para que isso seja possível são necessárias três características: Uma base de dados; velocidade de processamento para análise em tempo real; e a capacidade de trabalhar sob condições variadas. Um sistema de visão computacional tem como principal objetivo encontrar um resultado para um problema. Geralmente ele é dividido em etapas, sendo elas: aquisição, pré-processamento, segmentação, extração de características e reconhecimento e interpretação, cada etapa estando associada a uma base de conhecimento.

2.2.5 Aquisição da imagem

Essa etapa é realizada por uma câmera que transforma a imagem analógica em digital. Os aspectos de projeto envolvidos nesta etapa são: Condições de iluminação, velocidade de aquisição resolução, dentre outros.

A iluminação influencia o processo de limiarização, pois altera o histograma da imagem original, eliminando eventualmente regiões de vale entre dois picos, a qual poderia ser utilizada para definição de um limiar. Uma das técnicas que podem ser utilizadas para compensar a não uniformidade da iluminação é a projeção do padrão de iluminação em uma superfície refletora branca, de forma a normalizar a imagem utilizando o limiar escolhido.

2.2.6 Pré-processamento

Esta etapa consiste em corrigir imperfeições e aprimorar a qualidade, para que as etapas seguintes sejam realizadas com sucesso. Diversas técnicas de processamento de imagens podem ser utilizadas nesta fase do projeto, como, por exemplo, a limiarização e a suavização.

2.2.7 Segmentação

A segmentação é responsável por dividir a imagem em unidades, ou seja, em objetos de interesse que a compõem. Técnicas de processamento de imagens como a detecção de bordas, afinamento, operadores morfológicos e detecção de regiões, podem ser utilizadas nesta etapa. Algoritmos específicos para identificação de formas (linhas, curvas, e outras formas que podem ser parametrizadas) em imagens, cores ou intensidades também são utilizados para conduzir o processo de extração dos objetos de interesse da imagem.

No reconhecimento de padrões de imagens, dependendo do domínio da aplicação, a cor é um bom descritor das propriedades de um objeto, que pode ajudar a simplificar o processo de identificação e segmentação, mas ainda podem ser utilizados outros descritores baseados em formas e texturas.

2.2.8 Extração de características

Nesta etapa características são extraídas das imagens resultantes da etapa anterior através de descritores. A entrada ainda é uma imagem, porém a saída desta etapa é um conjunto de dados referentes à imagem, comumente denominado vetor de características.

2.2.9 Reconhecimento e interpretação

A partir das características obtidas na etapa anterior, traduzidas por seus descritores, são atribuídos rótulos aos objetos identificados e o significado de um conjunto de objetos também é descrito.

2.2.10 Base de conhecimento

Para que as tarefas citadas anteriormente sejam executadas é necessária a existência de um conhecimento sobre o problema a ser resolvido, armazenado em uma base de dados.

2.2.11 Filtragem de imagens

As técnicas de filtragem de imagens são utilizadas com o objetivo de ajustá-las mais

adequadamente para a aplicação desejada. A filtragem pode ser no domínio espacial ou da frequência. No domínio espacial as técnicas de filtragem atuam diretamente sobre a matriz de *pixels*, já no domínio da frequência é utilizado o teorema da convolução e ambos os filtros podem ser matematicamente convertidos de um domínio para outro.

2.2.12 Suavização de imagens no domínio espacial

É baseada no uso de máscaras de convolução adequadas ao objetivo, normalmente são utilizadas para borramento e remoção de ruídos. As técnicas mais conhecidas para suavização são a filtragem pela média e o filtro da mediana.

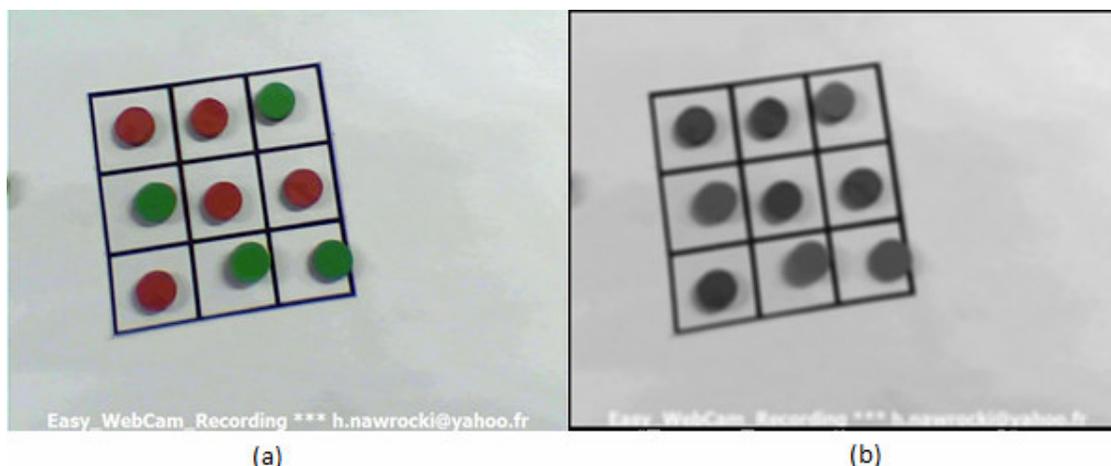


Figura 5 - Filtro da média

Fonte: Autoria própria.

2.2.13 Detecção de bordas

As bordas são as fronteiras entre duas regiões, cujos níveis de cinza/cores predominantes são razoavelmente diferentes. Para detecção e realce de bordas aplicam-se filtros espaciais lineares de um dos dois tipos: Baseados no gradiente da função de luminosidade da imagem e baseados no laplaciano. Ambos são aproximados por máscaras de convolução ou operadores 3x3. Entre eles, podem-se citar como exemplo os operadores de Roberts, Sobel, Prewitt e Frei-Chen.

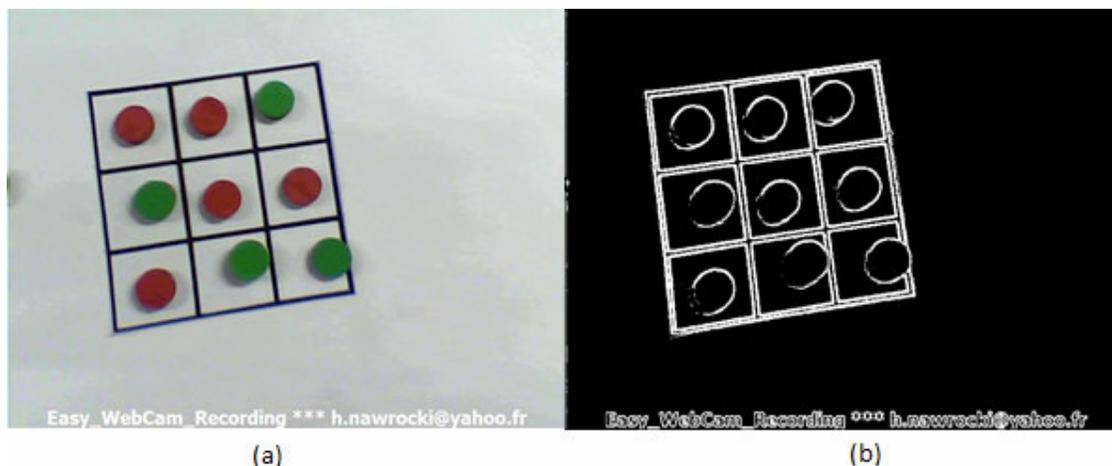


Figura 6 - Aplicação do filtro de Sobel

Fonte: Autoria própria.

2.2.14 Histograma

O histograma indica o percentual de *pixels* de uma imagem que representam um determinado nível de cinza, tais valores são normalmente representados por um gráfico de barras que representa para cada nível de cinza o percentual ou a quantidade de *pixels* correspondentes na imagem. A partir do mesmo é possível obter características da imagem como qualidade, nível de contraste e brilho médio.

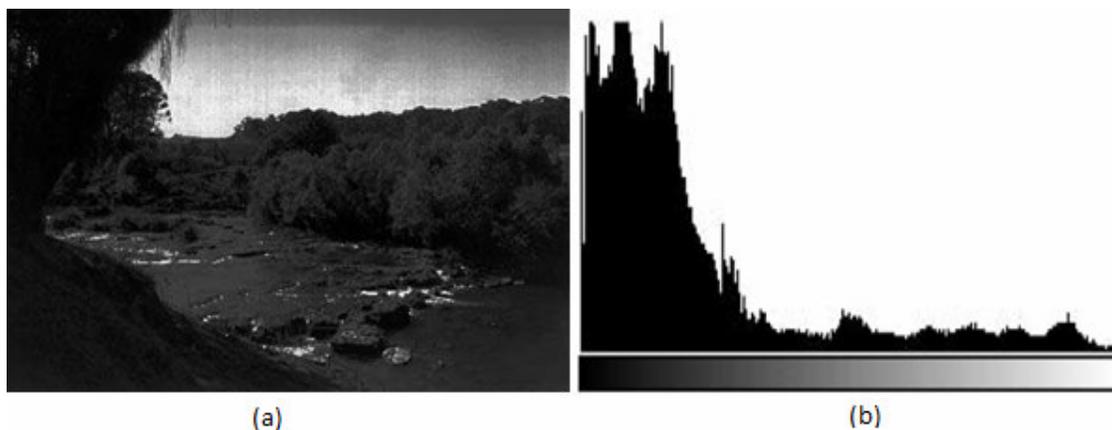


Figura 7 - Histograma

Fonte: Adaptado de Marques Filho e Vieira Neto (1999).

2.2.15 Transformada de Hough

A transformada de Hough é um algoritmo que busca obter a figura geométrica de ajuste mais adequado a partir de um conjunto de pontos (FERRIANI; RIBEIRO, 2005). Foi desenvolvida por Paul Hough em 1962 e patenteada pela IBM. Ela foi elaborada para detectar características de formas que podem ser parametrizadas como, linhas, círculos e elipses. A transformada de Hough pode ser utilizada para qualquer função na forma implícita $g(v, c) = 0$, em que v é um vetor de coordenadas e c é um vetor de coeficientes. A detecção de curvas sem representações analíticas também é possível, a partir de generalizações da transformada de Hough (GONZALEZ; WOODS, 2000).

A Figura 8 ilustra a utilização da transformada de Hough para detecção de linhas. A imagem original (a) foi obtida através da aplicação do filtro de Sobel para detecção de bordas em uma imagem do tabuleiro com todas as células ocupadas. Na Figura 8 (b) pode-se observar que as linhas do tabuleiro, tanto internas quanto externas, foram detectadas pelo algoritmo. Por outro lado, as peças (círculos) não foram classificadas como linhas pela transformada de Hough, como era esperado.

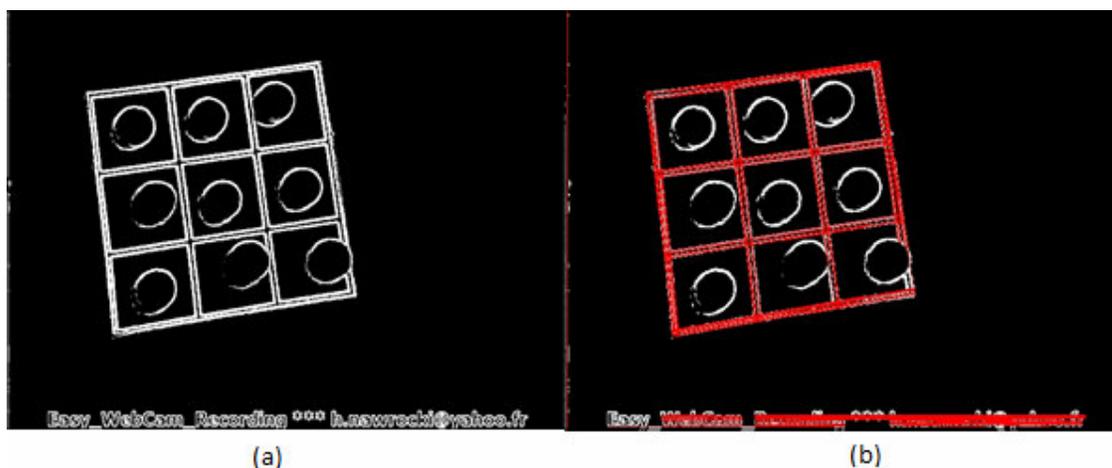


Figura 8 - Transformada de Hough de linhas

Fonte: Autoria própria.

2.3 Estado da arte

A ideia de se construir robôs começou a tomar força no início do século XX, com a necessidade de aumentar a produtividade e melhorar a qualidade dos produtos. Foi nessa época que os robôs industriais encontraram suas primeiras aplicações (SHHEIBIA, 2001).

Suas capacidades de movimentação dependem do sistema utilizado para acioná-los, os tipos de acionamentos podem ser divididos em:

2.3.1 Acionamento Hidráulico

Utilizado em robôs de maior porte, pois propicia maior força e velocidade, porém tendem a vazarem óleo o que é um inconveniente (SHHEIBIA, 2001), um exemplo de robô hidráulico é o HYD – 2800 mostrado na Figura 9.

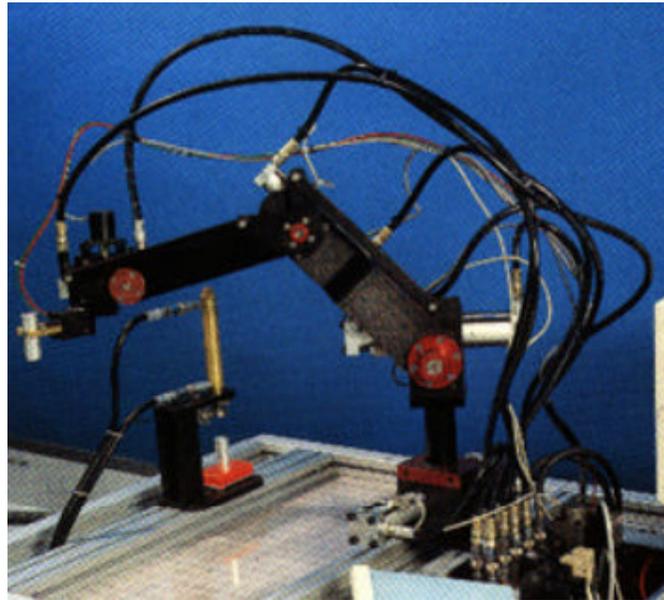


Figura 9 - Robô hidráulico - HYD - 2800

Fonte: Shheibia (2001).

2.3.2 Acionamento Elétrico

Os manipuladores elétricos não possuem velocidade e potência equivalente aos hidráulicos, no entanto, têm mais precisão e são menores em tamanho (SHHEIBIA, 2001). Um exemplo de robô elétrico é o S400 (Figura 10) utilizado por empresas para soldagem a ponto e a arco e outros.



Figura 10 - Robô elétrico - S400

Fonte: Shheibia (2001).

2.3.3 Acionamento Pneumático

É utilizado em robôs de pequeno porte com até quatro graus de liberdade e limitados a operações simples. Um exemplo é o robô BOADICEA apresentado na Figura 11.

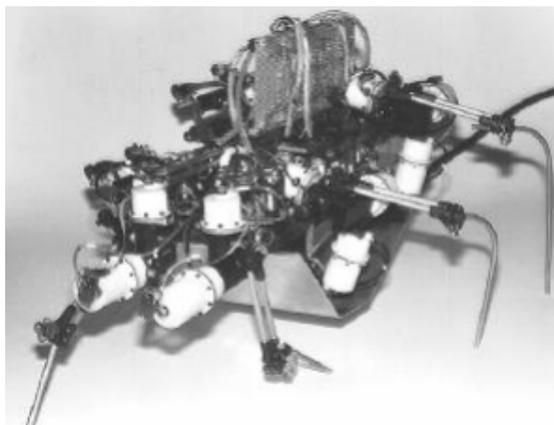


Figura 11 - Robô pneumático - BOADICEA

Fonte: Shheibia (2001).

O conceito do moderno robô industrial foi criado por Joseph Engelberger, que em 1962, juntamente com Devol, desenvolveram o primeiro protótipo de robô, chamado de *Unimate*, cuja primeira instalação registrada aconteceu na *Ford Motor Company* e teve como objetivo o descarregamento de uma máquina de fundição sob pressão (PAZOS, 2002).

Sistemas robóticos podem ser utilizados em diversas tarefas e aplicações. O trabalho de Sampaio Junior et al (2009), mostra a utilização de robô para jogar golfe. Já os trabalhos de (AMATO et al., 2013) e (MALYSZ; SIROUSPOUR, 2013) descrevem robôs manipuladores conduzidos por atuadores pneumáticos musculares e sua eficácia na utilização de manipuladores tele operados.

Atualmente, a grande maioria dos robôs industriais utiliza como controlador um computador pessoal ou algum tipo de controlador digital programável. Por isso, a robótica frequentemente é considerada uma combinação de diversas ciências, como eletromecânica e informática (PAZOS, 2002).

A integração da robótica com a visão computacional e a inteligência computacional, também vem se desenvolvendo com o passar dos anos. Os trabalhos de (FELICIANO; SOUZA; LETA, 2005) e (FERREIRA et al., 2004) mostram a visão computacional sendo utilizada para inspeção e medição de peças e o reconhecimento de gestos para controle de um servomecanismo.

Na robótica, a visão computacional é utilizada em basicamente três tipos de aplicações, sendo elas a Inspeção que é utilizada em processo de controle de qualidade, a Identificação tendo por aplicação a classificação e identificação de peças e a Servo-movimentação visual com finalidade de obter dados para dirigir a trajetória de um robô manipulador ou explorador (PAZOS, 2002).

A partir da união da robótica e a inteligência artificial podem ser vislumbradas novas fronteiras de pesquisa. Com isso é possível o desenvolvimento de agentes robóticos inteligentes, robôs humanóides, emoções e inteligência, robótica evolutiva, semiótica e sistemas inteligentes, cognição artificial, mentes artificiais e consciência. Exemplos de robôs que utilizam estes conceitos são os robôs insetos, em que pesquisadores optaram por alcançar uma inteligência equiparável à dos insetos. Outro exemplo são os robôs humanóides capazes de subir escadas, descritos em (GUDWIN, 2005).

A cinemática inversa de manipuladores é um dos problemas presentes na robótica, e que tem sido solucionada de diferentes formas. No trabalho de (SANTOS; LOPES; GEBARA JUNIOR, 2005), foi utilizada a técnica de algoritmos genéticos com redução progressiva do espaço de busca para resolver este problema.

3 MATERIAIS E MÉTODOS

O projeto foi desenvolvido em quatro etapas, a seguir:

Etapa 1: Revisão de Literatura

O início do projeto se deu através do estudo de conceitos relacionados às áreas envolvidas. Foram estudados conceitos sobre visão computacional, processamento de imagens, Inteligência Artificial e Robótica.

Etapa 2: Sistema de visão

Nesta etapa foi desenvolvido o ambiente de trabalho integrando o manipulador robótico e a câmera digital. A câmera foi fixada em um suporte sobre o manipulador, adquirindo imagens de forma temporal, permitindo que a mesma possa ter a visão de todo o sistema, incluindo o manipulador, peças e tabuleiro. Após a obtenção das imagens, foi executado o processamento das imagens, seguido da segmentação da imagem para identificar os objetos de interesse (tabuleiro, peças). No processamento após a retirada do tabuleiro foi aplicado o filtro de Sobel e realizado um afinamento para uma melhor identificação do mesmo.

Etapa 3: Tomada de decisão

Quando o sistema identifica alterações no cenário, é feita uma verificação se houve algum vencedor e em caso afirmativo é finalizada a execução, caso contrário o módulo de busca a melhor jogada com o objetivo de vencer o jogo e segue para a etapa seguinte, como ilustra o diagrama da Figura 12.

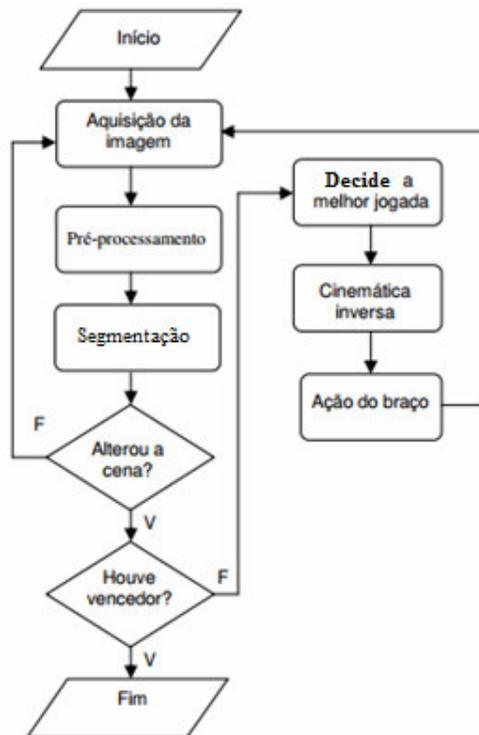


Figura 12 - Etapas de execução do sistema

Etapa 4: Cinemática Inversa

As coordenadas virtuais obtidas na etapa anterior são convertidas para coordenadas reais, para que seja possível identificar os movimentos necessários que o manipulador mova as peças para o local desejado.

3.1 Espaço de estados (Tabuleiro)

O tabuleiro utilizado é quadrado e foi feito em uma folha de papel A4. As células têm o tamanho de duas vezes o diâmetro das peças para evitar que a garra do manipulador invada a célula adjacente nos momentos em que for largar/pegar as peças. Percebeu-se também, através de experimentos, que o tabuleiro deve ser fixado na mesa para evitar que o mesmo se mova nos instantes em que o manipulador executa as jogadas. O tabuleiro é branco com linhas pretas, visando evitar a etapa de binarização no algoritmo e também facilitando a impressão. Ele é composto por nove células que durante o jogo podem estar em três estados diferentes vazias, contendo a peça X ou contendo a peça O.

3.2 Tecnologias

Foram realizados testes com bibliotecas de processamento de imagens na IDE (*Integrated Development Environment*) *CodeBlocks*, no sistema operacional *Windows 7*. Para o processamento de imagens foram selecionadas três bibliotecas para realização de testes, sendo elas, *OpenCV*, *Filters* e *CImg*, todas livres. A primeira testada foi a *OpenCV*, porém não houve sucesso na sua instalação e configuração no ambiente supra-citado. Ocorreram erros referentes a não localização de arquivos necessários para a compilação. A segunda testada foi a *Filters*, na qual os testes de instalação e configuração ocorreram com sucesso. Após testar o funcionamento da mesma, foram levantadas as funções disponíveis e percebeu-se que as funções consideradas em um primeiro momento necessárias estavam disponíveis. Por fim foi testada a biblioteca *CImg*, que também teve sucesso, porém constatou-se que ela disponibiliza um número menor de funções em relação à *Filters* testada anteriormente. Devido a tais motivos a biblioteca de processamento de imagens selecionada para a realização do projeto foi a *Filters*.

3.3 Recursos de Hardware

Os recursos de hardware necessários para a execução do jogo da velha são:

- Manipulador robótico do kit didático Robix, ilustrado na Figura 13a.;
- 01 Câmera digital USB (Figura 13b);
- 01 Suporte para a câmera;
- 01 Tabuleiro, ilustrado na Figura 13c.;
- 09 Peças de madeira do Jogo; e
- Computador.



Figura 13 - Materiais e equipamentos

Fonte: Autoria própria.

3.4 Recursos de Software

- Editor e compilador em linguagem C;
- Software/Biblioteca para captura de imagens em tempo real;
- Biblioteca para processamento de imagens;
- Biblioteca para movimentação do manipulador mecânico no processador Arduino.

4 DESCRIÇÃO DO SISTEMA

O sistema desenvolvido tem como entrada somente as imagens capturadas pela webcam, que foi conectada à porta USB do computador. As imagens são capturadas por um software continuamente e armazenadas em um diretório, em um intervalo de tempo pré-definido. No computador é executado o módulo de visão computacional, implementado na linguagem C, que tem acesso às imagens capturadas, e também realiza os processamentos necessários para a identificação dos objetos e a verificação se houve alterações na cena (ocorreu alguma jogada) na imagem. A partir dessas informações é possível identificar o estado atual do jogo, para então, através do módulo de tomada de decisão, escolher a melhor jogada para aquela situação, e, através da cinemática inversa, calcular os movimentos necessários para que o manipulador possa executar a jogada. Tais valores são enviados através da porta serial para o Arduino que é o responsável por executar os movimentos.

Requisitos do sistema

Para o funcionamento do sistema algumas condições de contorno devem ser satisfeitas. Primeiramente, é utilizado um ambiente controlado, ou seja, deve-se manter a mesma iluminação, manter uma distância fixa entre a câmera e o ambiente de trabalho, utilizar a abertura da câmera fixa, uma vez que as alterações destas condições de trabalho poderiam resultar em falhas no sistema. As peças e o tabuleiro devem ter tamanhos fixos, pois o tamanho do tabuleiro serve de base para que seja possível calcular a posição onde as peças se encontram e a posição para onde devem ser movidas. O tabuleiro e as peças podem ser colocados em qualquer posição e orientação dentro do espaço de trabalho, no entanto, no início do jogo o tabuleiro deve ser fixado em uma posição para evitar que o mesmo se mova durante a partida. Além disso, as peças devem estar a uma distância mínima de 1,5cm uma da outra, tal espaço é necessário para que a garra do manipulador possa pegar e soltar as peças.

4.1 Arquitetura do sistema

O sistema proposto foi dividido em quatro módulos, sendo eles: Módulo de processamento de imagem; módulo de visão computacional; tomada de decisão; e cinemática inversa, conforme mostrado na Figura 14.

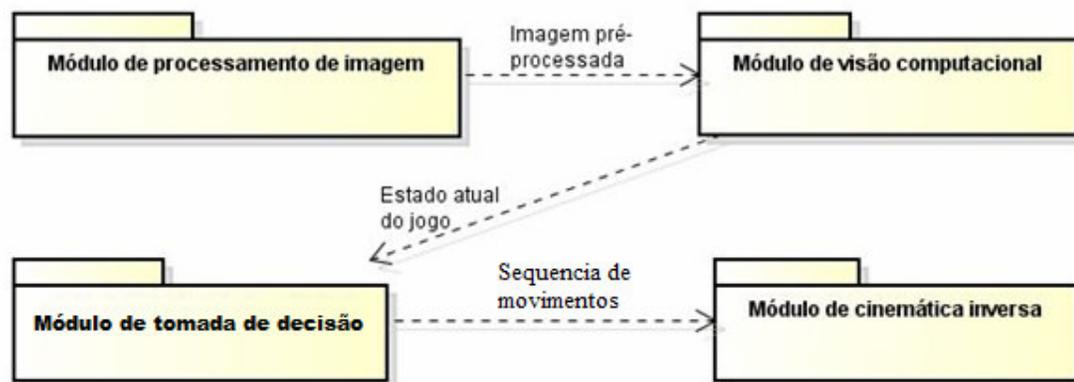


Figura 14 - Arquitetura do sistema

Fonte: Autoria própria.

4.1.1 Módulo de processamento de imagem

Este módulo é responsável pela aquisição e pré-processamento das imagens como mostra a Figura 15. É responsável por obter as imagens e fazer os tratamentos necessários para identificar os objetos de interesse (peças, células do tabuleiro). Após a realização destas etapas a imagem processada está em condições para ser utilizada pelo módulo seguinte, que é o de visão computacional.

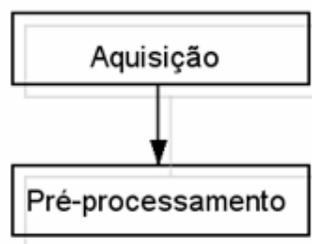


Figura 15 - Etapas do módulo de processamento de imagem

Fonte: Autoria própria.

Na etapa de pré-processamento foi realizado o corte da parte inferior da imagem, para a eliminação da marca d'água inserida pelo software de captura de imagens, como mostra a Figura 16.

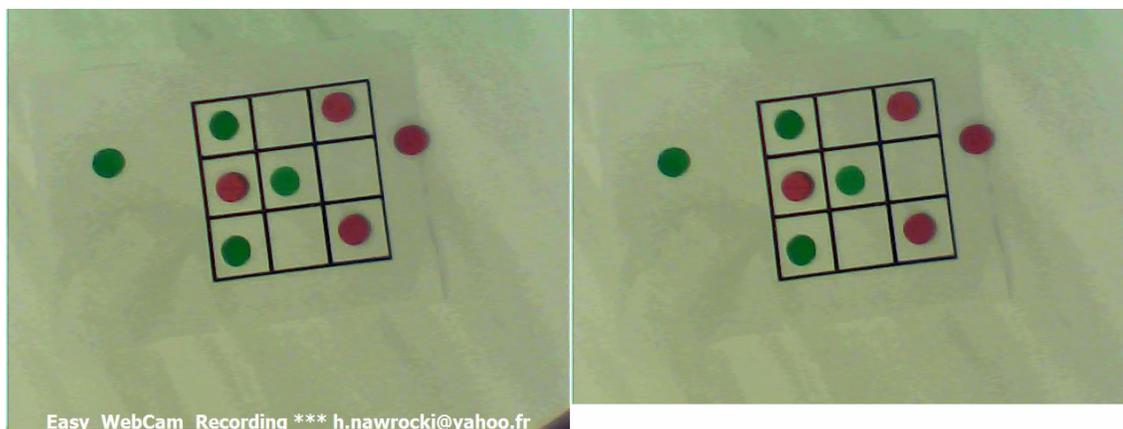


Figura 16 - Eliminação da marca d'agua

Fonte: Autoria própria.

4.1.2 Módulo de visão computacional

Consiste no processo de segmentação, extração de características, reconhecimento e interpretação dos resultados. Ele é responsável por identificar e localizar os objetos e retornar uma lista dos objetos e a sua localização. A sequência de etapas deste modulo é apresentada na Figura 17.



Figura 17 - Etapas do módulo de visão computacional

Fonte: Autoria própria.

O processo de segmentação da imagem foi dividido em três etapas: Segmentação do tabuleiro (Figura 31), segmentação das peças vermelhas (Figura 32) e segmentação das peças verdes (Figura 33). A segmentação utilizada foi baseada em cor, na qual foi feita uma varredura *pixel a pixel*, e a partir de limiares pré-definidos nos três canais de cor (R, G, B) foram separados os objetos da imagem. Após a segmentação encontrou-se a posição do tabuleiro na imagem através da realização de uma nova varredura buscando os quatro pontos extremos (x, y), que foram utilizados para o cálculo das coordenadas das nove células do tabuleiro.

A solução de segmentação baseada em cor foi selecionada por sua simplicidade e desempenho. Outra alternativa poderia ser a utilização da transformada de Hough para detectar linhas e círculos, em uma primeira etapa, sendo seguida por uma análise baseada em cor para diferenciar as peças dos jogadores. Em uma versão futura, pretende-se disponibilizar o jogo através da Internet, de forma que o desempenho se tornou um critério importante para a decisão do método a ser adotado.

4.1.3 Módulo de tomada de decisão

A partir das informações extraídas da imagem é possível identificar o estado atual do jogo, e através do módulo, definir a melhor jogada. O algoritmo de decisão foi desenvolvido pelo autor como mostrado na Figura 18. Os jogos são um tipo bastante especializado em que os agentes se restringem a um pequeno número de ações, cujos resultados são definidos através de regras precisas, podendo ter objetivos que podem ser competitivos ou cooperativos. O estado inicial e o conjunto de movimentos válidos formam a árvore de jogo, a partir da qual o algoritmo de decisão faz uma busca para obter a melhor jogada possível (RUSSELL; NORVIG, 2004).



Figura 18 - Etapas do módulo de tomada de decisão

Fonte: Autoria própria.

4.1.4 Módulo de cinemática inversa

Após a escolha da jogada é preciso executá-la. O módulo de cinemática inversa é responsável por calcular e executar os movimentos necessários para agarrar a peça e leva-la até a posição escolhida no tabuleiro a partir das coordenadas fornecidas pelo módulo anterior, como apresentado na Figura 19.

Tais coordenadas são obtidas realizando cálculos com os valores das coordenadas virtuais do tabuleiro dos pontos de origem e destino e as medidas reais do tabuleiro.

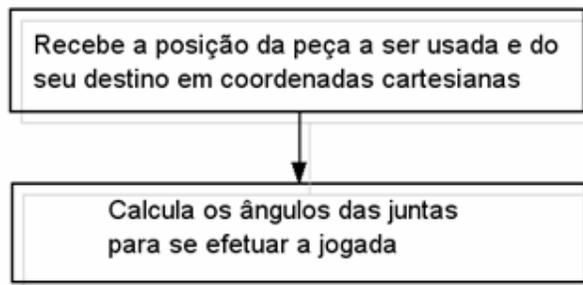


Figura 19 - Etapas do modulo de cinemática inversa

Fonte: Autoria própria.

5 RESULTADOS

Neste capítulo são apresentados o dimensionamento da área de trabalho para o jogo, os cálculos da cinemática inversa e os testes de validação dos módulos de processamento de imagens, decisão e cinemática inversa.

5.1 Área de trabalho do manipulador

Com base nas medidas do manipulador e através de testes para verificar o alcance da câmera foi possível definir o layout físico da área de trabalho, que é mostrado na Figura 20. A área de trabalho que o manipulador pode alcançar e que foi utilizada é um pouco menor que o alcance total do manipulador, ou seja, a área de trabalho foi delimitada neste projeto, porém a área de trabalho foi limitada devido ao alcance da câmera e à necessidade de deixar um espaço livre para que o manipulador pudesse permanecer enquanto aguarda para executar uma jogada sem ficar no campo de visão da câmera.

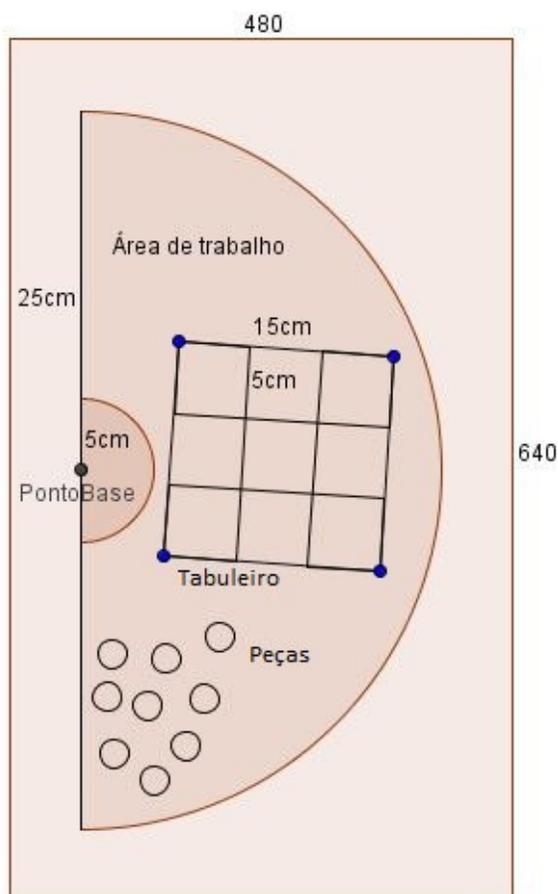


Figura 20 - Layout físico da área de trabalho (Vista superior)

Fonte: Autoria própria.

Com base na área de trabalho e no tamanho das peças, que são circulares e tem raio de 1 cm, foi possível, então, definir o tamanho do tabuleiro. O tamanho escolhido foi de 15x15cm, não ultrapassando o espaço disponível na área de trabalho, porém grande o suficiente para que seja possível efetuar as jogadas sem que uma peça invada o espaço destinado a outra.

Para definir a posição da *webcam* foram realizados testes a partir dos quais observou-se que, para que fosse possível utilizar a área de trabalho, conforme mostrado na Figura 20, é necessário que a *webcam* esteja a uma altura de aproximadamente 1,10m. A Figura 21 mostra uma foto capturada a uma altura de 60cm com a *webcam* fixada na parte superior de uma bancada e outra com uma altura de 1,10m.



Figura 21 - Vista superior a uma altura de 60 e 110cm

Fonte: Autoria própria.

5.2 Cinemática inversa

A partir das informações extraídas das imagens e da decisão da jogada a ser executada, foi definida a localização física da peça a ser utilizada pelo manipulador para realizar a jogada e a posição de destino da peça para que seja finalizada a jogada. Tais posições são representadas por coordenadas cartesianas (x , y). Após a obtenção dessas coordenadas as mesmas são convertidas para coordenadas polares calculando-se o raio (r) através da fórmula:

$$r = \sqrt{x^2 + y^2}$$

E o ângulo (θ) pela fórmula:

$$\theta = \text{arctg}\left(\frac{y}{x}\right)$$

As coordenadas obtidas são utilizadas para o cálculo dos movimentos necessários para se executar a jogada.

Foram utilizados os seguintes caracteres para representar os ângulos e medidas do braço:

θ_1 : Ângulo da primeira junta;

θ_2 : Ângulo da segunda junta;

θ_3 : Ângulo da terceira junta;

A: Comprimento do primeiro elo;

B: Comprimento do segundo elo;

C: Comprimento do terceiro elo;

Para o braço alcançar a posição informada, primeiramente foi efetuado o cálculo do ângulo da segunda junta. Para isso primeiro foi necessário calcular o valor de b que representa a distância do efetuador em relação à segunda junta através da fórmula:

$$b = (B + C \cdot \cos(\theta_2))$$

Após obter este valor definiu-se o valor do ângulo da segunda junta, que foi feito através da lei dos cossenos, isolando o ângulo na fórmula, obtém-se:

$$\theta_2 = \arccos\left(\frac{A^2 + b^2 - r^2}{2 \cdot A \cdot b}\right)$$

Após, os valores obtidos foram substituídos na equação anterior para se encontrar o valor do ângulo θ_2 .

O valor do ângulo da primeira junta foi definido ou pela soma ou diferença, dependendo do padrão adotado (horário ou anti-horário) entre o valor do ângulo informado θ (recebido pela porta serial do manipulador mecânico) e o valor de θ_i , que foi obtido através da equação:

$$\theta_i = \arccos\left(\frac{A^2 + r^2 - b^2}{2 \cdot A \cdot r}\right)$$

Os movimentos necessários para descer o efetuador até a peça para agarrá-la ou soltá-la, foram predefinidos através de testes e sempre executados da mesma forma, uma vez que todas as peças possuem as mesmas dimensões e estão dispostas sob uma base plana (área de trabalho), ou seja, estão a uma mesma altura da garra do efetuador.

Para realização de testes com os resultados do módulo de cinemática inversa, foi utilizado o software matemático GeoGebra (Figura 22), a partir do qual foi possível validar os resultados.

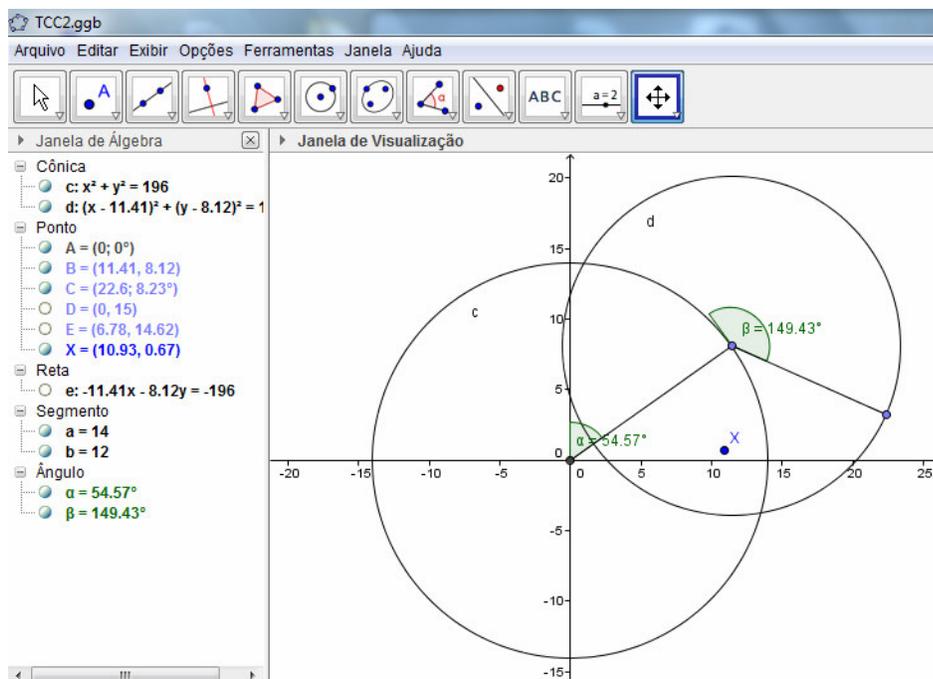


Figura 22 - Software GeoGebra

5.3 Experimentos Preliminares

Foram realizados experimentos em laboratório para validação do sistema desenvolvido. Para a comunicação entre a *webcam* e o computador foram realizados testes com alguns softwares com o intuito de obter o mais adequado para adquirir imagens continuamente em um intervalo de tempo predefinido de amostragem armazenando-as em uma pasta/diretório a ser utilizada pelo módulo seguinte. Para desempenhar esta função o software *Easy WebCam Recording* foi selecionado. Trata-se de um software livre e que atende os requisitos necessários. A

Figura 23 mostra a sua interface.

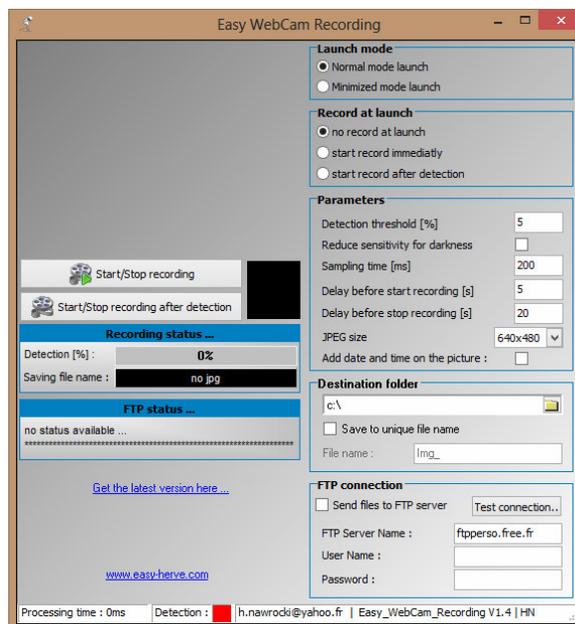


Figura 23 - Software para captura das imagens

Um problema encontrado no uso desse software foi o fato de adicionar uma marca na parte inferior da imagem, como pode ser visto na Figura 24. No entanto, esta marca foi removida previamente (pré-processamento) durante a identificação dos objetos, conforme mostrado na página 32.

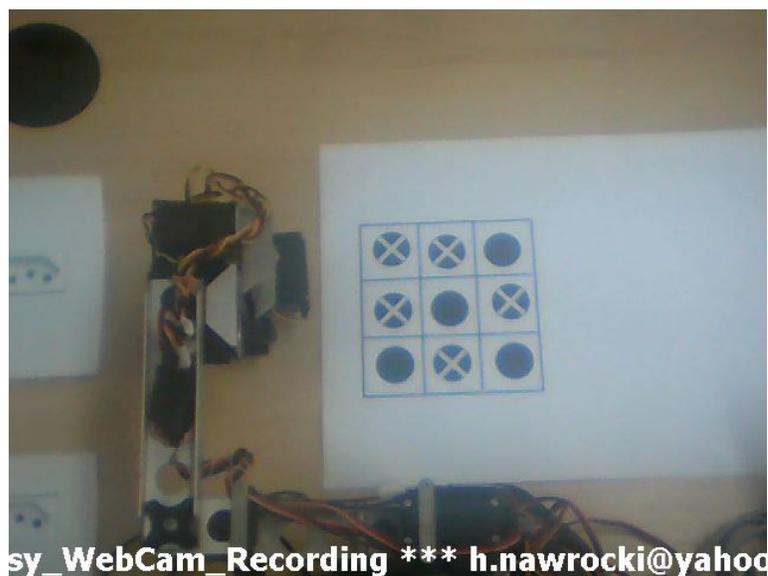


Figura 24 - Marca adicionada na imagem

Após a realização de alguns testes com a câmera adotada percebeu-se um problema, notou-se que ela sempre ajustava o histograma da imagem, após alguns segundos de captura a

imagem dependendo da iluminação do ambiente era escurecida ou clareada, e ainda que este recurso não poderia ser desabilitado, motivo pelo qual levou à aquisição de outra câmera.

Foram feitos testes com as imagens obtidas utilizando a biblioteca Filters. As Figuras 25 e 26 mostram a utilização de algumas funções presentes nesta biblioteca, onde as figuras marcadas com a letra (a) representam a imagem original, com a letra (b) mostram a utilização da binarização, com a letra (c) a utilização do filtro de Sobel e a letra (d) mostra a utilização de uma função encontrada na biblioteca Filters, chamada BlobExplorer que segmenta as regiões da imagem.

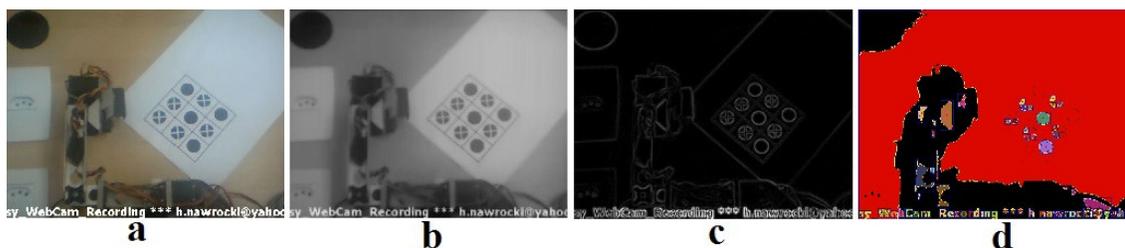


Figura 25 - Aplicação dos filtros a uma imagem da área de trabalho

Fonte: Autoria própria.

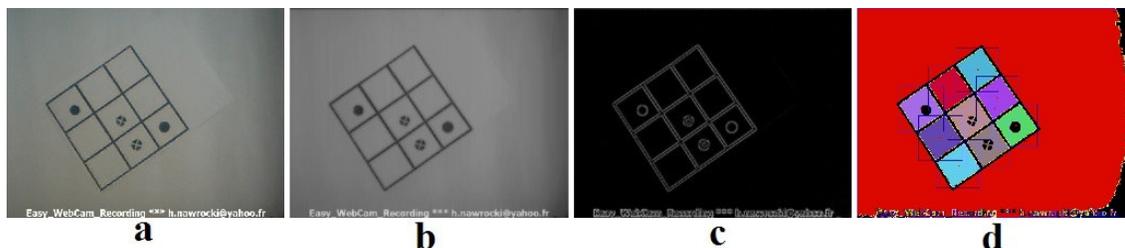


Figura 26 - Aplicação dos filtros a uma imagem do tabuleiro com algumas peças

Fonte: Autoria própria.

Também foi implementado o algoritmo de tomada de decisão, foram definidas as regras de jogo, as condições de vitória e derrota e iniciado o algoritmo na linguagem C, bem como o interfaceamento do algoritmo com o manipulador mecânico. Esta interface foi feita da seguinte forma: o algoritmo de tomada de decisão retornou a célula (numeradas de 1 a 9 conforme mostra a Figura 27) para o módulo de cinemática inversa, o qual fez a conversão deste dado em coordenadas polares. A partir destas coordenadas calcula-se os movimentos necessários para se executar a jogada e através da porta serial é enviada a sequência de movimentos para que seja executada.

1	2	3
4	5	6
7	8	9

Figura 27 - Numeração das células do tabuleiro

Fonte: Autoria própria.

Foram ainda realizados testes com o manipulador executando movimentos sequenciais predefinidos, agarrando uma peça em um lugar de origem e levando e soltando em outro.

5.4 Testes de Validação

Os primeiros testes feitos para a identificação do tabuleiro e peças foram realizados com a utilização dos filtros de Sobel e BlobExplorer, porém, devido ao fato do autor não ser capaz de identificar na literatura uma documentação detalhada sobre o funcionamento do BlobExplorer, decidiu-se então optar por outro método para a identificação dos objetos do jogo.

A solução encontrada foi através da utilização de objetos de cores diferentes, juntamente com um algoritmo de segmentação por cor, mantendo a cor do tabuleiro em preto e utilizando inicialmente peças nas cores vermelho e azul (Figura 28).

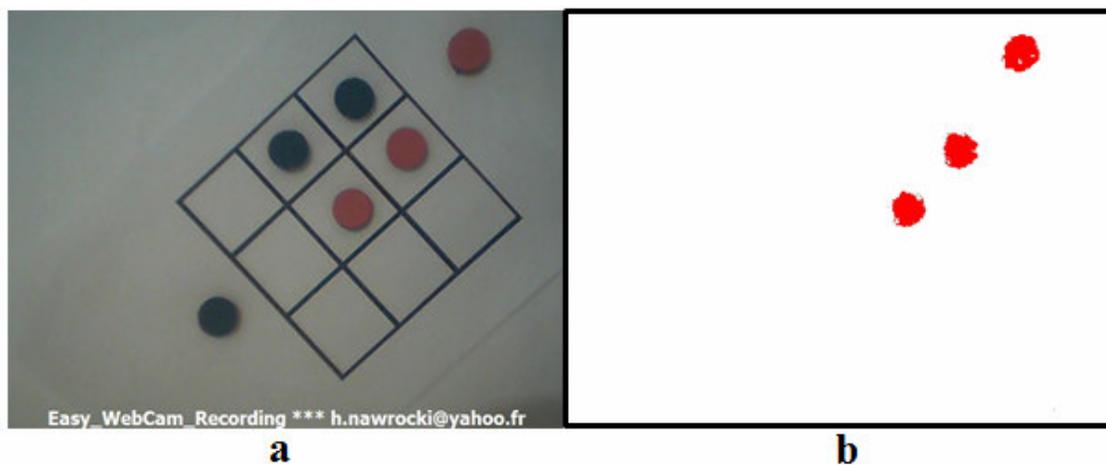


Figura 28 - Imagem original e peças vermelhas

Fonte: Autoria própria.

Porém nos testes realizados ainda não era possível separar as peças azuis do tabuleiro como é possível visualizar nas Figuras 29 e 30.

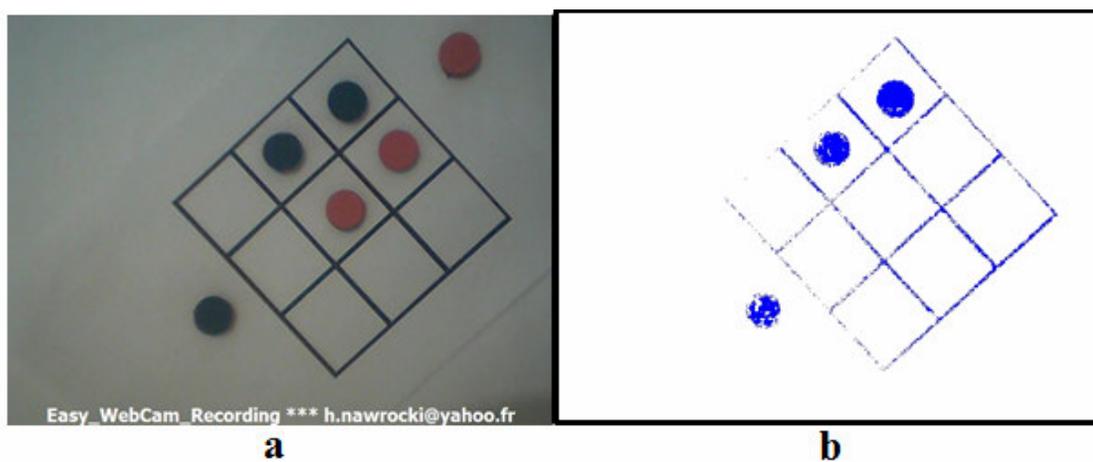


Figura 29 - Imagem original e peças azuis

Fonte: Autoria própria.

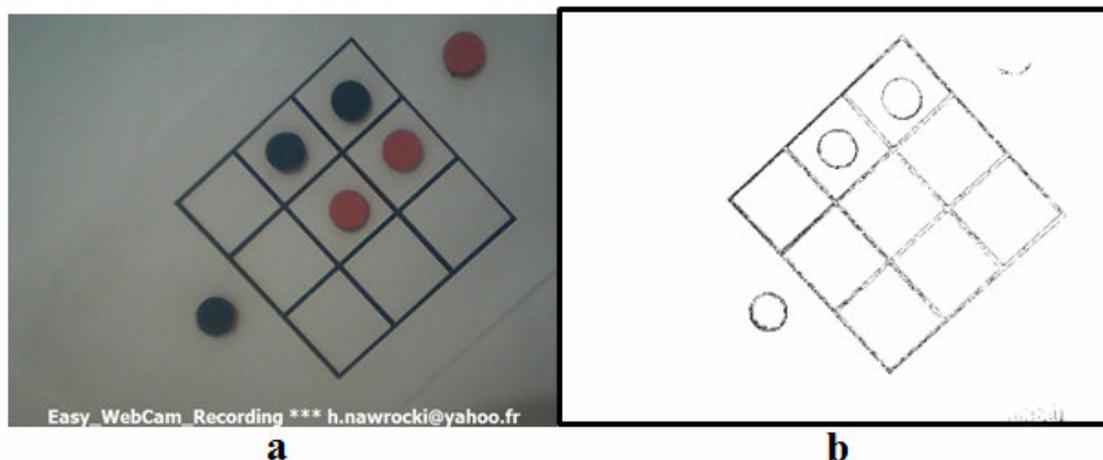


Figura 30 - Imagem original e tabuleiro

Fonte: Autoria própria.

Como visto nas figuras anteriores, mesmo fazendo alterações nos limiares para separar as peças azuis do tabuleiro o resultado ainda não foi adequado para segmentar os objetos de interesse.

Uma hipótese levantada é de que a cor azul (R 36, G 50, B 59) utilizada poderia ser próxima do preto, dificultando o processo. Então escolheu-se a cor verde para realizar testes, primeiramente utilizou-se um verde claro (R 132, G 125, B 96) para evitar problemas de ser similar com o tabuleiro, o que poderia dificultar a segmentação.

Com a utilização da cor verde o problema de separar as demais peças do tabuleiro foi solucionado, porém, poucos *pixels* verdes foram segmentados no tabuleiro como ilustra as Figuras 31 e 32.

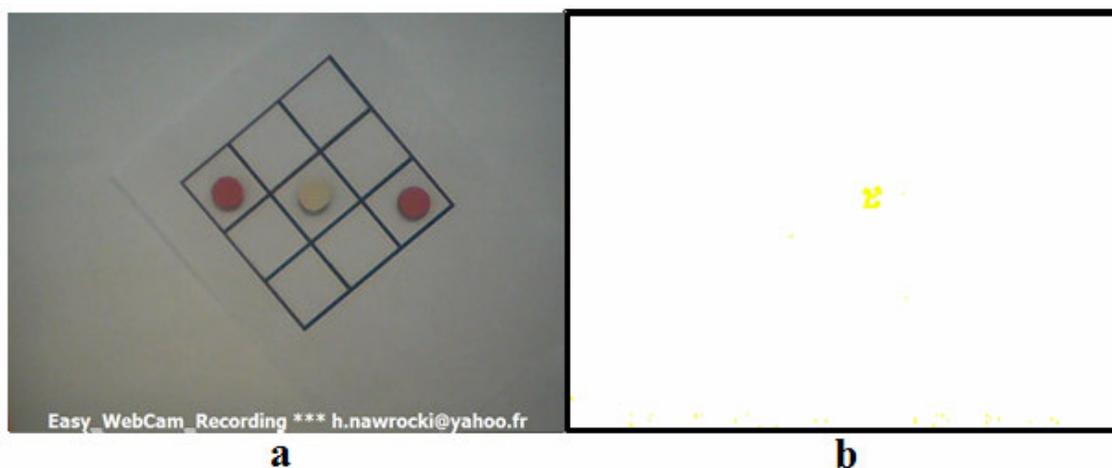


Figura 31 - Imagem original e peças verdes

Fonte: Autoria própria.

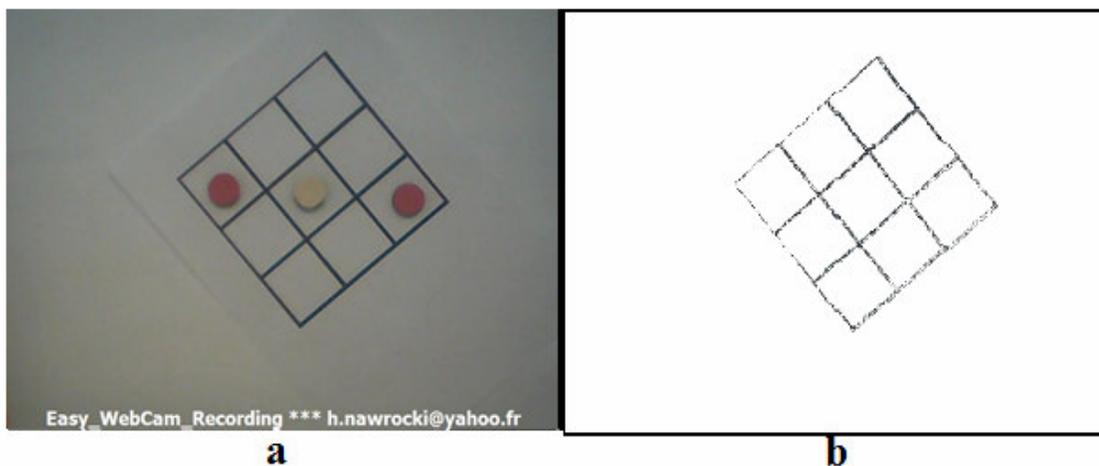


Figura 32 - Imagem original e tabuleiro

Fonte: Autoria própria.

Por fim decidiu-se alterar as peças verdes (R 132, G 125, B 96) para outra tonalidade de verde (R 62, G 114, B 31) obtendo um melhor resultado no processo de segmentação como pode ser visto nas Figuras 33, 34 e 35.

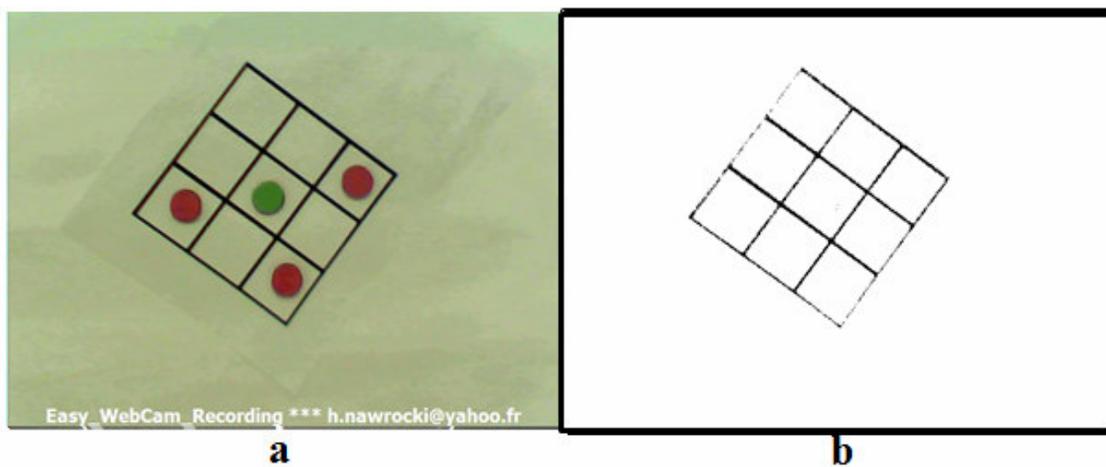


Figura 33 - Imagem original e tabuleiro

Fonte: Autoria própria.

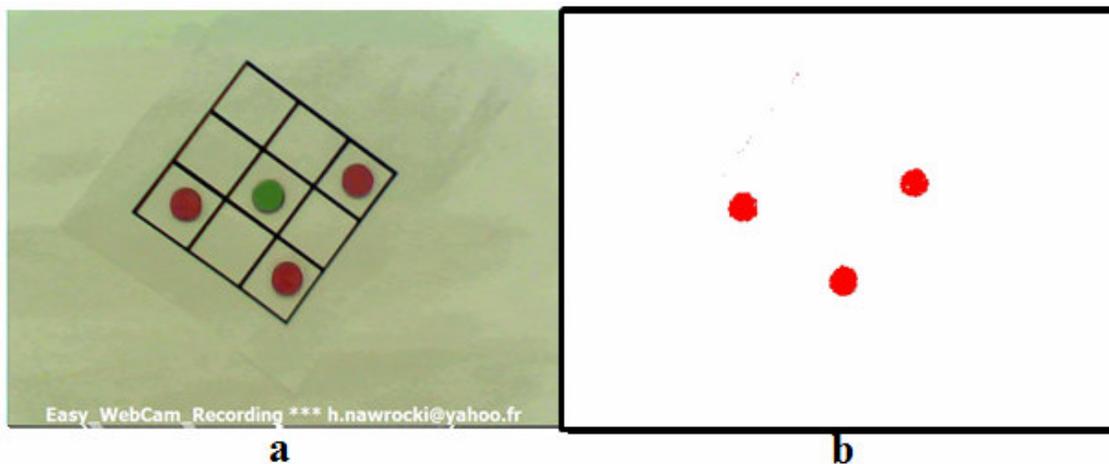


Figura 34 - Imagem original e peças vermelhas

Fonte: Autoria própria.

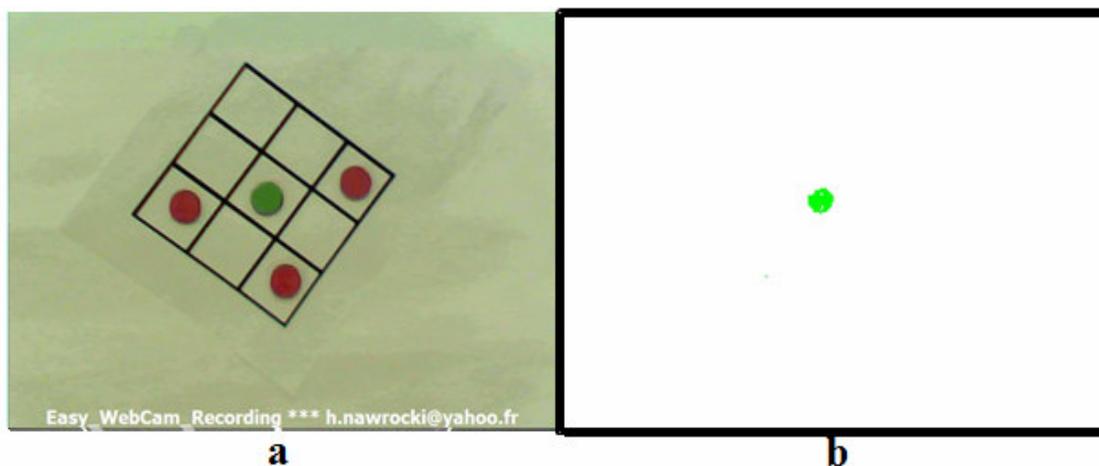


Figura 35 - Imagem original e peças verdes

Fonte: Autoria própria.

Após a segmentação das peças e o tabuleiro o próximo passo então foi a localização do tabuleiro e os limites das células, que foram obtidos a partir dos *pixels* classificados como cor preta. Nas Figuras 37 e 38 é possível ver os pontos obtidos, a partir da imagem capturada foram primeiramente extraídos os extremos do tabuleiro, para então calcular os demais pontos. Para obter os extremos do tabuleiro foi realizada uma varredura vertical iniciando da margem superior esquerda até a margem inferior direita na imagem do tabuleiro (Figura 36) obtida na etapa anterior.

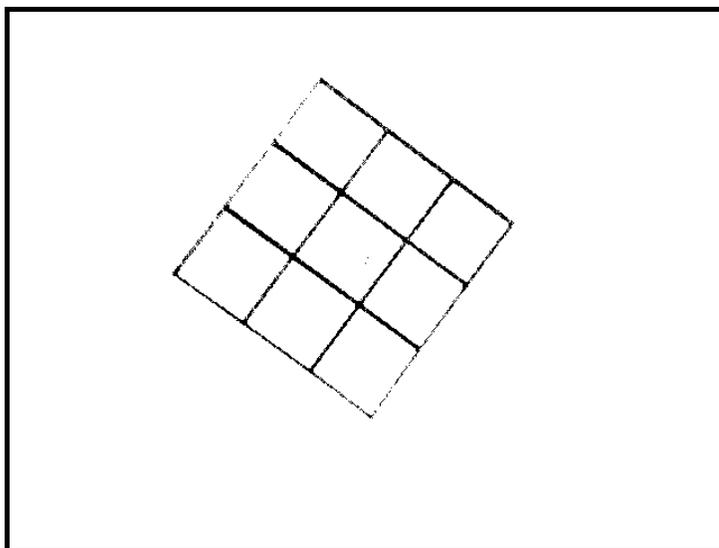


Figura 36 - Saída tabuleiro

Fonte: Autoria própria.

Realizando a varredura foram encontrados os 4 extremos do tabuleiro, e através da equação de reta determinou-se os vértices de cada célula como mostra a Figura 37.

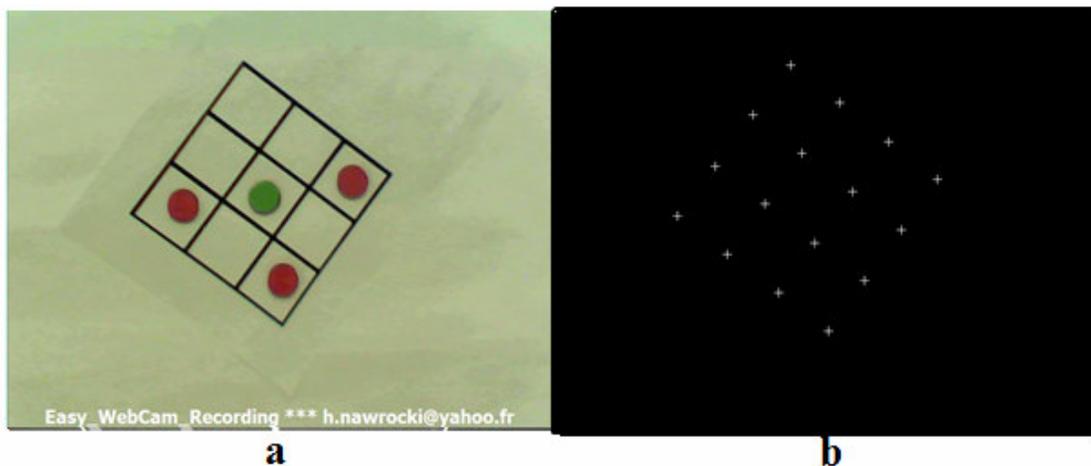


Figura 37 - Imagem original e processada com pontos de saída

Fonte: Autoria própria.

Após calculados os vértices das células, foram calculados os centros de cada célula pela média das coordenadas dos pontos extremos de cada célula, a saída é mostrada na Figura 38. Esses pontos serão os locais de destino para onde o manipulador irá mover as peças e ainda sabendo do tamanho real das células foi possível obter o valor em pixels do raio aproximado das peças na imagem.

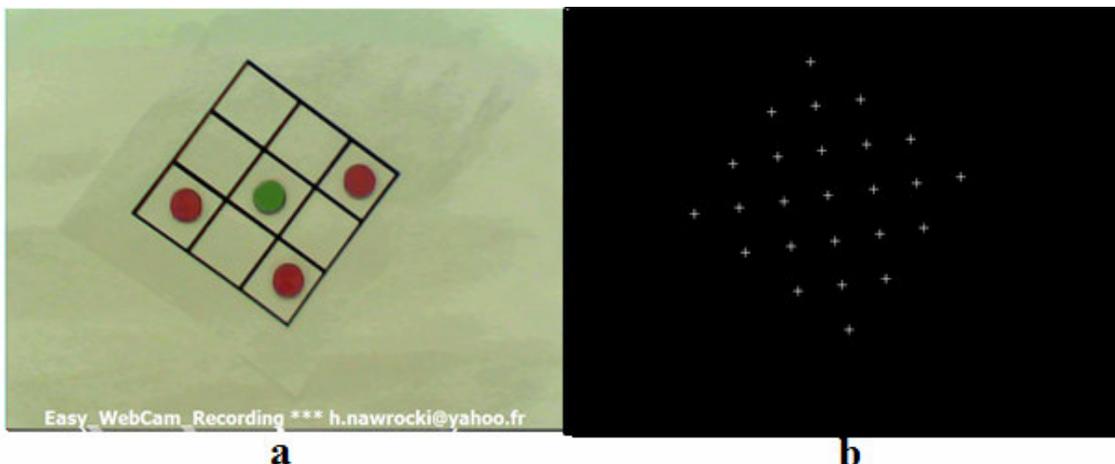


Figura 38 - Imagem original e processada

Fonte: Autoria própria.

Após a identificação do tabuleiro o próximo passo foi a identificação das peças contidas no tabuleiro, para isso foi feita uma varredura na imagem iniciando na coluna de coordenada x mínima do tabuleiro e finalizando na coluna de coordenada x máxima. Nessa varredura, buscou-se uma continuidade significativa de *pixels* de uma mesma cor para garantir a existência e identificação da peça. A Figura 39 mostra a saída do algoritmo após a identificação das peças contidas no tabuleiro.

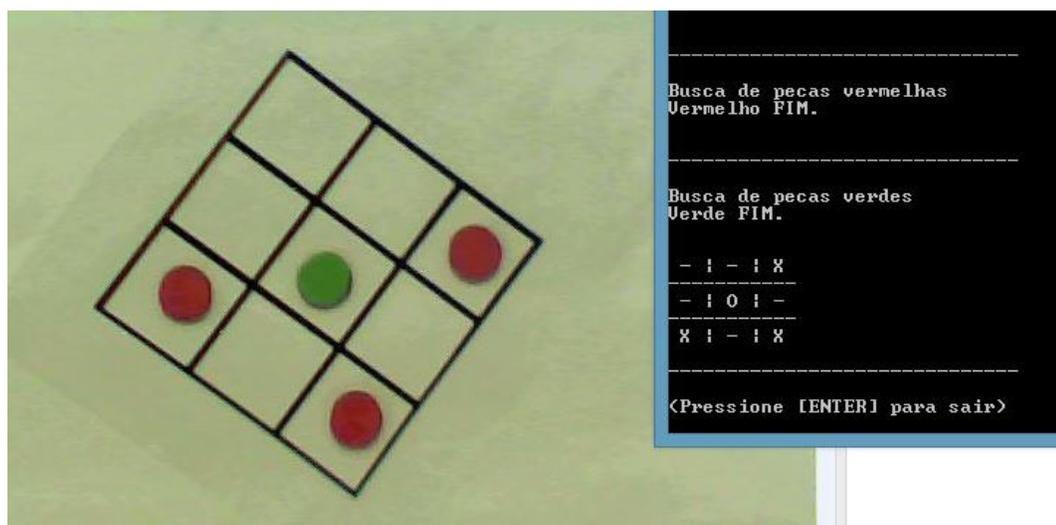


Figura 39 - Identificação das peças contidas no tabuleiro

Fonte: Autoria própria.

A partir da identificação do tabuleiro e do estado do jogo ainda era necessário localizar as peças externas ao tabuleiro que serão utilizadas nas próximas jogadas e definir um

ponto de referência (ponto base) para a conversão de coordenadas da imagem para coordenadas físicas.

Para as peças externas ao tabuleiro optou-se por utilizar sempre a primeira peça disponível para realizar a jogada, ou seja, a cada execução do algoritmo ele pesquisa por uma peça verde ou vermelha que esteja fora do tabuleiro e assim que a localiza, calcula o seu ponto central para que seja definido com ponto de origem onde o manipulador irá buscar a peça para realizar a jogada. Para a localização das peças externas optou por buscar também uma continuidade significativa de *pixels* de uma mesma cor. Após a localização da peça externa na área de trabalho, são selecionados os *pixels* das extremidades da peça, para a partir deles calcular o centro da peça em questão, o resultado pode ser visto na Figura 40.

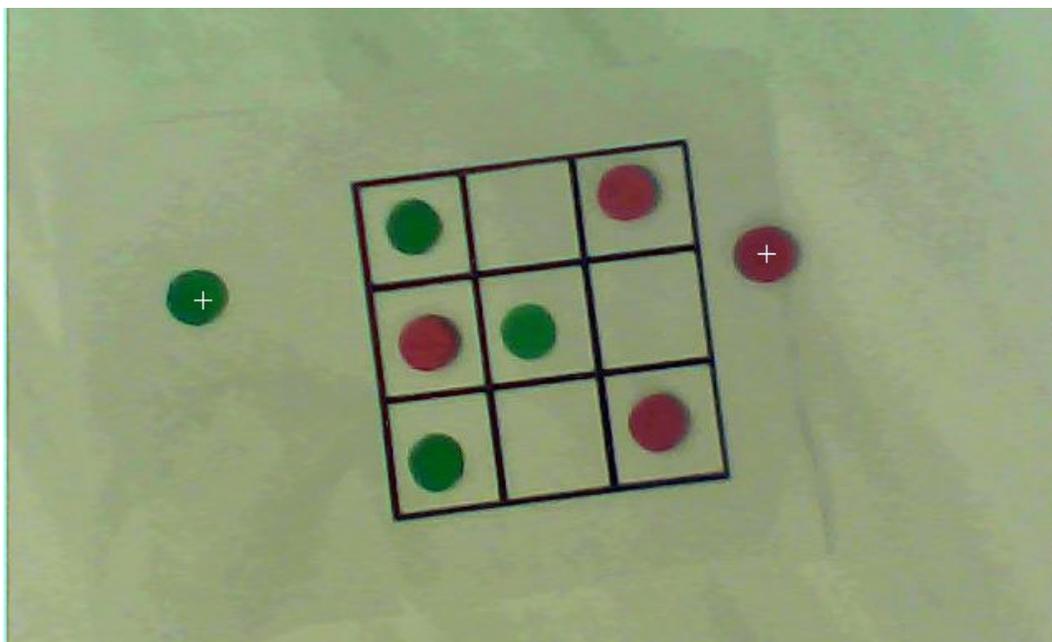


Figura 40 - Peças externas ao tabuleiro

Fonte: Autoria própria.

De posse da localização das peças a serem utilizadas (x_{origem} , y_{origem}) e o seu destino ($x_{destino}$, $y_{destino}$) como ilustra a Figura 41, é feita a conversão das coordenadas de origem e destino da imagem para coordenadas físicas. Que são encontradas através de um cálculo utilizando a medida original do tabuleiro e sua medida na imagem é possível encontrar um fator de proporcionalidade dividindo a medida da imagem pela física conhecida, e como as grandezas são diretamente proporcionais basta multiplicar este coeficiente pela medida encontrada na imagem para encontrar a medida física.

```

X | - | X
---|---|---
- | 0 | -
---|---|---
0 | X | 0

Posicao das pecas a serem usadas:
Vermelha (X) = x=460 y=148
Verde (O) = x=120 y=176
-----

```

Figura 41 - Estado do tabuleiro e próximas peças a serem utilizadas

Fonte: Autoria própria.

5.5 Experimento de movimentação das peças

Foram realizados experimentos com as peças dispostas na área de trabalho em posições pré-determinadas e com uma sequência de movimentos predefinidos com o objetivo de mensurar o erro de posicionamento do manipulador.

Primeiramente foram obtidas as coordenadas físicas bidimensionais dos centros das células do tabuleiro, tendo como ponto base o centro do manipulador. (Figura 42).

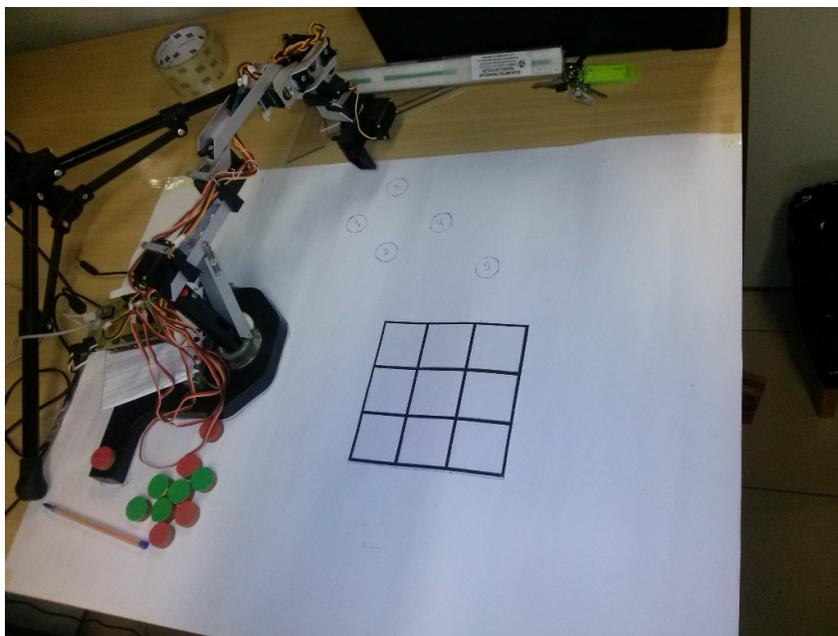


Figura 42 - Tabuleiro sem peças

Fonte: Autoria própria.

O tabuleiro foi fixado à área de trabalho durante o experimento. Para levar cada peça da origem (externa ao tabuleiro) para o destino (célula do tabuleiro) foram calculados os

ângulos da primeira e segunda junta utilizando as equações baseadas em cinemática inversa, apresentadas nos materiais e métodos 5.2.

A Figura 43 mostra uma série de imagens de uma execução do experimento.

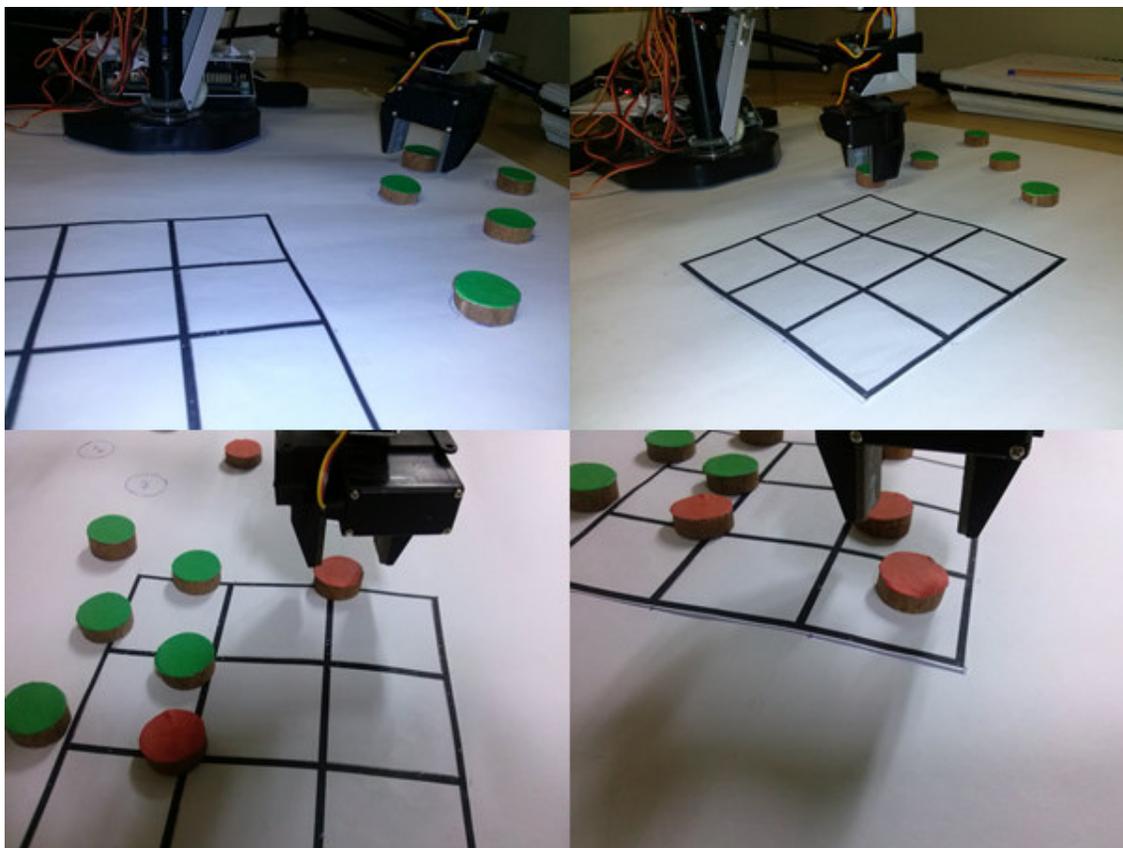


Figura 43 - Execução do experimento 1

Fonte: Autoria própria.

A Tabela 1 apresenta os resultados de execução para cada peça. Nessa tabela as coordenadas esperadas correspondem aos centros das células a partir do ponto base, medidas em centímetros. Da mesma, as coordenadas foram obtidas também foram utilizando a mesma unidade e sistema de referência (Figura 44). A coluna erro foi calculada com a distância relativa entre os dois pontos (esperado e obtido).

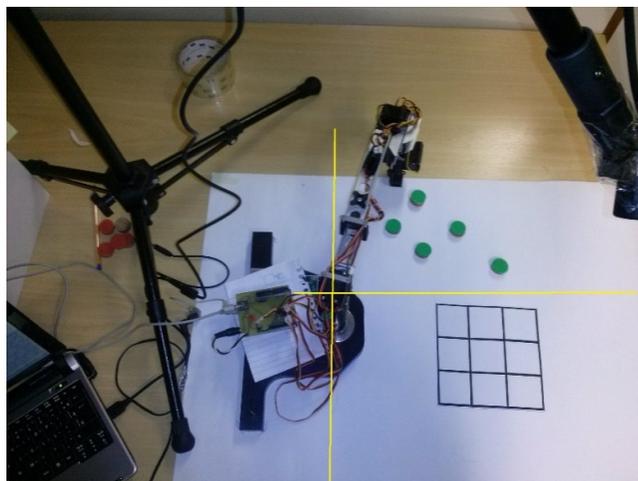


Figura 44 - Eixos

Fonte: Autoria própria.

Célula	Posições esperadas		Ângulos calculados		Posições obtidas		Erros em cm		
	X	Y	Junta A	Junta B	X	Y	Erro X	Erro Y	Erro
1	-1,6	16,7	29	191	-5,9	12,8	4,3	3,9	5,81
2	3,6	16,5	47	190	-0,4	14	4	2,5	4,72
3	8	16	65	184	4,8	13	3,2	3	4,39
4	-1,5	21,3	47	160	-3,6	17,3	2,1	4	4,52
5	3,5	21	61	161	1,5	17,8	2	3,2	3,77
6	7,7	21	76	152	5,5	18,5	2,2	2,5	3,33
7	-1,3	25,5	75	112	-4,2	23,5	2,9	2	3,52
8	3,5	25,5	89	107	3,5	25	0	0,5	0,50
9	7	25	102	97	7,6	26,5	-0,6	-1,5	1,62
						Média	2,23	2,23	3,57

Tabela 1

Com base nos dados da Tabela 1 observou-se que o erro cresce à medida que o destino se aproxima do ponto base do manipulador. O erro médio de 3,57 cm.

Das nove células, apenas duas das peças foram posicionadas dentro dos limites das células correspondentes no tabuleiro.

A Figura 45 mostra os resultados finais de duas execuções do experimento 1. Pode-se observar que apesar das coordenadas de origem e destino serem as mesmas e os ângulos calculados da mesma forma, houve uma diferença nas posições alcançadas nas duas execuções do experimento. Uma hipótese é imprecisão dos motores e outra ao soltar a peça,

em que a mesma pode se deslocar da posição de descarte, uma vez que ela é largada a uma determinada altura do tabuleiro.

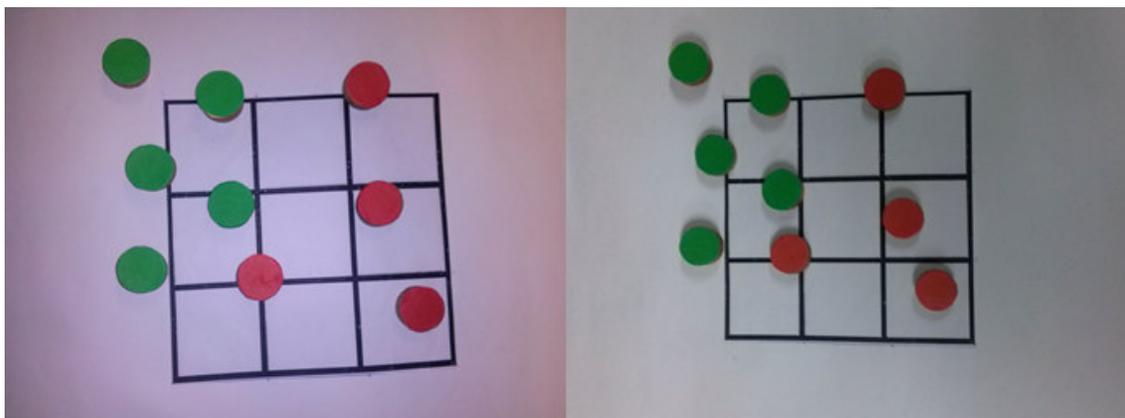


Figura 45 - Diferença no resultado de duas execuções

Fonte: Autoria própria.

Dado a magnitude dos erros de posicionamento obtidos utilizando os ângulos calculados pelo sistema, optou-se pela definição dos ângulos de forma empírica, por tentativa e erro. Assim, foram utilizados os ângulos apresentados na Tabela 2.

Célula	Posições esperadas		Ângulos empíricos		Posições obtidas		Erros em cm		
	X	Y	Junta A	Junta B	X	Y	Erro X	Erro Y	Erro
1	-1,6	16,7	50	170	-1,5	16,3	-0,1	0,4	0,41
2	3,6	16,5	66	165	3,7	17	-0,1	-0,5	0,51
3	8	16	84	155	8,8	16,7	-0,8	-0,7	1,06
4	-1,5	21,3	67	140	-0,7	21,2	-0,8	0,1	0,81
5	3,5	21	80	135	4,4	21,5	-0,9	-0,5	1,03
6	7,7	21	93	125	9	21,3	-1,3	-0,3	1,33
7	-1,3	25,5	92	90	-0,9	26,2	-0,4	-0,7	0,81
8	3,5	25,5	100	90	4,5	25,4	-1	0,1	1,00
9	7	25	106	90	9,1	25,9	-2,1	-0,9	2,28
						Média	-0,83	-0,33	1,03

Tabela 2

O experimento realizado com ângulos empíricos teve um melhor resultado obtendo um erro médio de 1,03 cm.

Foram comparados os ângulos calculados com os empíricos conforme a Tabela 3.

Célula	Ângulos calculados		Ângulos empíricos		Diferença A	Diferença B
	Junta A	Junta B	Junta A	Junta B		
1	29	191	50	170	21	21
2	47	190	66	165	19	25
3	65	184	84	155	19	29
4	47	160	67	140	20	20
5	61	161	80	135	19	26
6	76	152	93	125	17	27
7	75	112	92	90	17	22
8	89	107	100	90	11	17
9	102	97	106	90	4	7
				Média	16,33	21,56

Tabela 3

Conforme a Tabela 3 as diferenças entre os ângulos calculados e os empíricos não são sempre as mesmas para ambos os motores, portanto, não é possível ajustar o resultado fazendo apenas um deslocamento.

5.6 Experimento manipulador x humano

No segundo experimento realizado, as peças do manipulador foram novamente colocadas em posições pré-determinadas. Entretanto, após cada jogada do humano, o manipulador recebe a célula de destino, fornecida pelo módulo lógico do jogo.

Neste experimento não foram utilizados os cálculos do experimento 1 pois verificou-se que os erros de posicionamento eram consideráveis. Portanto, utilizaram-se os ângulos empíricos apresentados na tabela 2.

Ocorreram também problemas com sombras ao testar o sistema em uma das salas da UTFPR, pois na mesma havia diversas fontes de luz gerando sombras das peças. Nos testes preliminares realizados em outro ambiente havia somente uma fonte de luz acima do sistema o que evitava a ocorrência de sombras, como mostra a Figura 46. Devido a tal motivo optou-se por dispor as peças do jogador humano e do manipulador fora do campo de visão do manipulador.

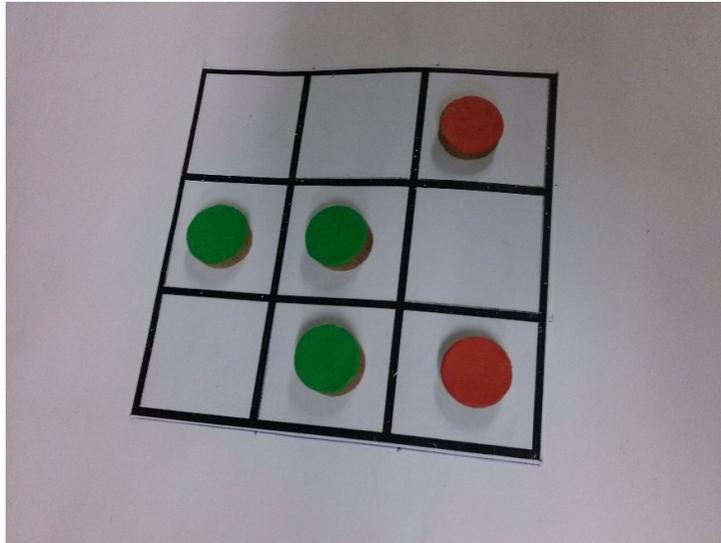


Figura 46 - Peças com sombreamento

Fonte: Autoria própria.

Para agarrar as peças, a garra do manipulador fica paralela ao segundo elo do manipulador para evitar que a mesma colida com a peça (Figura 47a), e ao soltar a peça, a garra do manipulador é rotacionada em 90 graus para evitar que ao retornar para a posição inicial venha a colidir com a peça (Figura 47b).

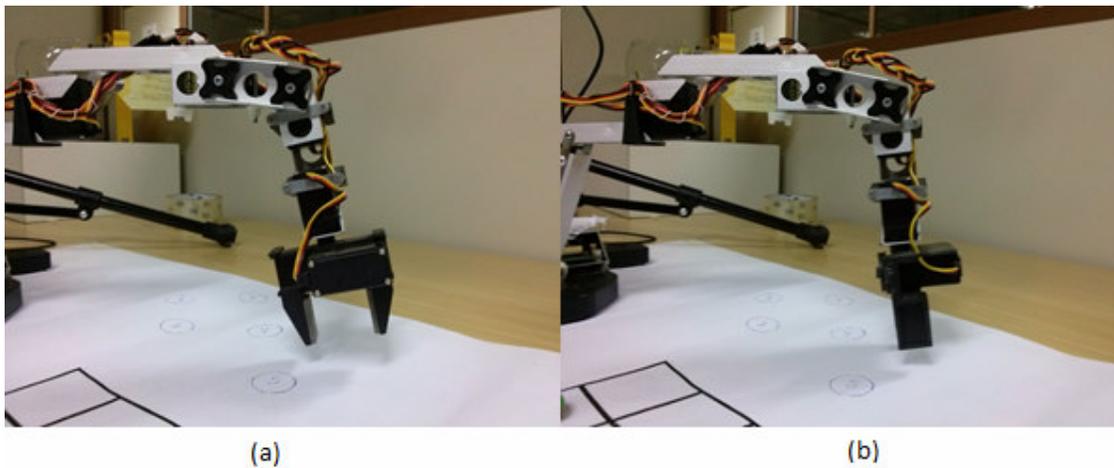


Figura 47 - Posição da garra

Fonte: Autoria própria.

6 CONCLUSÕES

Neste trabalho foi apresentado um jogo computacional entre um manipulador mecânico e um oponente humano, e também alguns resultados obtidos a partir de testes. Foi realizado um levantamento bibliográfico para estudo de conceitos relacionados às áreas envolvidas e entendimento dos processos a serem realizados. Percebeu-se, no decorrer do projeto, que o maior desafio é a interpretação das imagens, pois as mesmas podem sofrer alteração devido a diversos fatores, dentre eles a iluminação, sombras e a qualidade da câmera utilizada.

Através dos experimentos realizados foram tomadas decisões sobre a biblioteca de processamento de imagem, o software para a captura das imagens, o espaço utilizado como área de trabalho do manipulador e também que o processamento da cinemática inversa seria dividido entre o computador e o manipulador, reduzindo o fluxo de dados através da porta serial. Porém, após alguns testes verificou-se que todo o processamento necessário deveria ser executado no computador, pois os cálculos executados no processador do Arduino reduzem a precisão dos resultados devido a utilização de uma menor quantidade de casas decimais.

Os resultados mostraram que a identificação do tabuleiro e peças ocorreu com sucesso e a comunicação do programa em C no computador com o Arduino através da serial também funcionou adequadamente. Entretanto, os posicionamentos das peças a partir dos resultados do módulo de cinemática inversa não obtiveram sucesso. Apesar dos ângulos calculados estarem corretos, o manipulador não se movimentou de acordo com os ângulos fornecidos.

O objetivo de desenvolver um jogo computacional, tendo como jogadores o manipulador robótico e um ser humano, explorando os recursos do manipulador mecânico foi atingido. A importância deste trabalho localiza-se autonomia de decisão do manipulador robótico no cenário que foi apresentado, ou seja, relacionado ao espaço e à localização de objetos.

Para além dos objetivos alcançados propõe-se, ainda, como trabalhos futuros modificar o módulo de visão computacional para, a partir da imagem processada pelo filtro de bordas (Sobel, por exemplo), detectar as peças e tabuleiro como círculos e linhas, respectivamente. Como o diâmetro das peças é conhecido e pode ser calibrado, ou seja, pode-se obter o diâmetro equivalente em pixels, essa proposta futura poderia utilizar a transformada de Hough para determinar os círculos existentes na imagem.

REFERÊNCIAS BIBLIOGRÁFICAS

AMATO, Francesco; COLACINO, Domenico; COSENTINO, Carlo; MEROLA, Alessio, **"Robust and optimal tracking control for manipulator arm driven by pneumatic muscle actuators,"** Mechatronics (ICM), 2013 IEEE International Conference on, vol., no. pp.827,834, Feb. 27 2013-March 1 2013.

DRAKOS, Nikos; MOORE, Ross. Processamento de imagens, 2002. Disponível em: <<http://www.vision.ime.usp.br/~thsant/pool/image-processing-iso/>>. Acesso em: 27 Agosto 2013.

FELICIANO, F.F.; SOUZA, I. L.; LETA, F. R. Visão Computacional Aplicada à Metrologia Dimensional Automatizada: Considerações sobre sua Exatidão. **ENGEVISTA**, v.7, n.2, p.38-50, Dezembro 2005.

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.

FERRIANI, Vinícius Meirelles Pereira; RIBEIRO, C. H. C.. **Deteção de Guias utilizando Variações da Transformada de Hough**. XI ENCITA - Encontro de Iniciação Científica do ITA, 2005, São José dos Campos.

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

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais**. São Paulo: Editora Edgard Blücher Ltda., 2000.

GUDWIN, Ricardo Ribeiro. **Novas fronteiras na inteligência artificial e na robótica**. 4º Congresso Temático de Dinâmica, Controle e Aplicações, Junho de 2005.

LORENZO, Nicola Di; CAMPERCHIOLI, Ida; GASPARI, Achille L, **Radius surgical system and conventional laparoscopic instruments in abdominal surgery: Application, learning curve and ergonomics**, *Surgical Oncology*, Volume 16, Supplement, December 2007, Pages 69-72, ISSN 0960-7404.

MALYSZ, P.; SIROUSPOUR, S., "**Task Performance Evaluation of Asymmetric Semi-Autonomous Teleoperation of Mobile Twin-Arm Robotic Manipulators**," *Haptics*, IEEE Transactions on, vol.PP, no.99, pp.1,1, 0, 2013.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. ISBN 8574520.

NASCIMENTO JUNIOR, Cairo Lúcio; YONEYAMA, Takashi. **Inteligência artificial em controle e automação**. São Paulo: Editora Edgard Blücher Ltda., 2000.

PAZOS, Fernando. **Automação de Sistemas & Robótica**. Rio de Janeiro: Axcel Books do Brasil Editora, 2002.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial**. São Paulo: Elsevier Editora Ltda., 2004.

SAMPAIO JUNIOR, Sérgio L.; NASCIMENTO, José Fábio de Lima; SANTOS, Francisco das C. M. dos. **Modelagem geométrica e cinemática para o planejamento de trajetória de um robô jogador de golfe**. IV Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica, 2009.

SANTOS, Alfranci Freitas; LOPES, Heitor Silvério; GEBARA JUNIOR, Munif. **Cinemática inversa de trajetórias de manipuladores robóticos redundantes utilizando algoritmos genéticos com redução progressiva do espaço de busca**. VII SBAI/ II IEEE LARS. São Luís, setembro de 2005.

SHHEIBIA, Tarig Ali Abdurrahman El. **Controle de um Braço Robótico Utilizando Uma Abordagem de Agente Inteligente**. Universidade Federal da Paraíba, Centro de Ciências e

Tecnologia, Coordenação de Pós-Graduação em Informática, Campina Grande, PB, Julho de 2001. Dissertação de Mestrado.

THE IRIS ROBOT SERIES; IN THE CLASSROOM. Disponível em: <http://www.mind.ilstu.edu/dev/robotlab/physical_robot_teacher_page.php>. Acesso em 25 maio 2013.