



**INSTITUTO FEDERAL DE EDUCAÇÃO,  
CIÊNCIA E TECNOLOGIA DO AMAZONAS  
CAMPUS MANAUS DISTRITO INDUSTRIAL**



**INSTITUTO FEDERAL**  
AMAZONAS  
Campus Manaus-Distrito Industrial

**HAYANNA SOARES PINHEIRO**

**SISTEMA DE NAVEGAÇÃO AUTÔNOMA DE VEÍCULOS NÃO TRIPULADOS EM  
AMBIENTE COM OBSTÁCULOS EM MOVIMENTO**

**MANAUS/AM**

**2011**

**HAYANNA SOARES PINHEIRO**

**SISTEMA DE NAVEGAÇÃO AUTÔNOMA DE VEÍCULOS NÃO TRIPULADOS EM  
AMBIENTE COM OBSTÁCULOS EM MOVIMENTO**

Monografia apresentada à banca examinadora do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – IFAM, como requisito parcial para a obtenção de aprovação no Programa Institucional para Concessão de Bolsas de Incentivo à Iniciação Científica – PIBIC, na área de Mecatrônica Industrial.

**Orientador:** Dr José Pinheiro de Queiroz Neto

**MANAUS/AM**

**2011**

**HAYANNA SOARES PINHEIRO**

**SISTEMA DE NAVEGAÇÃO AUTÔNOMA DE VEÍCULOS NÃO TRIPULADOS EM  
AMBIENTE COM OBSTÁCULOS EM MOVIMENTO**

**Defesa em**

Manaus, \_\_\_\_\_ de \_\_\_\_\_ de 20\_\_\_\_

**BANCA EXAMINADORA**

**Presidente:** Orientador: Prof. Dr. José Pinheiro de Queiroz Neto

\_\_\_\_\_

**1º Examinador:**

\_\_\_\_\_

**2º Examinador:**

\_\_\_\_\_

**MANAUS/AM**

**2011**

## RESUMO

A navegação autônoma requer, entre outras capacidades, aprender estratégias de navegação, adaptar-se a novas situações e construir conhecimento a partir de informações obtidas do ambiente. Com base nos dados coletados, os sistemas autônomos de navegação de robôs móveis são capazes de determinar trajetórias, conferindo ao robô a capacidade de obter informações sobre a sua localização, pontos de referência de alvos e de obstáculos. Desenvolver a navegação em um ambiente onde os obstáculos mudam de localização ao longo da trajetória é um problema ainda em aberto, cujas propostas de solução não são triviais e dependem de uma série de restrições. O presente trabalho consiste em uma pesquisa teórica e experimental que propõe desenvolver um sistema de controle para navegação de um robô em ambiente controlado através de técnicas de visão computacional, em que o método de extração de informações do ambiente físico será unicamente através de imagens. As cenas visualizadas por uma câmera trazem informações que determinam as decisões a serem tomadas para que o robô se desloque de um ponto de partida até um alvo desviando-se de obstáculos que mudam de posição enquanto o robô se desloca. O trabalho utiliza um sistema integrado a um par câmera/robô e utiliza a comunicação *Bluetooth* para o fluxo de dados.

**Palavras - chave:** Navegação autônoma. Robótica. Visão Computacional.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1 - Cronograma de Execução das Atividades Por Meta.....</b>	<b>10</b>
<b>Figura 2 - Arranjo esquemático do cenário .....</b>	<b>20</b>
<b>Figura 3 - Fluxograma do sistema de navegação autônoma proposto.....</b>	<b>21</b>
<b>Figura 4 - Algoritmo de trajetória.....</b>	<b>27</b>
<b>Figura 5 - Ambiente de navegação com câmera ortogonal ao plano da cena.....</b>	<b>29</b>
<b>Figura 6 - Imagem do ambiente de navegação.....</b>	<b>31</b>
<b>Figura 7 - Matriz de posições.....</b>	<b>31</b>
<b>Figura 8 - Instante de navegação do robô.....</b>	<b>32</b>

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>7</b>
<b>1.1 Motivação.....</b>	<b>7</b>
<b>1.2 Problema.....</b>	<b>8</b>
<b>1.3 Objetivos.....</b>	<b>8</b>
1.3.1 Objetivo Geral.....	8
1.3.2 Objetivos específicos.....	8
<b>1.4 Abordagem para solução do problema.....</b>	<b>9</b>
<b>1.5 Cronograma de atividades.....</b>	<b>10</b>
<b>1.6 Organização do Trabalho.....</b>	<b>10</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>11</b>
<b>2.1 Robótica Móvel.....</b>	<b>11</b>
<b>2.2 Navegação Autônoma de Robôs Terrestres.....</b>	<b>12</b>
<b>2.3 Visão Computacional e Processamento Digital de Imagens.....</b>	<b>12</b>
<b>2.4 Fundamentos de Cores e Imagem Digital.....</b>	<b>13</b>
2.4.1 Modelo de cor RGB.....	14
2.4.2 Modelo de cor HSI.....	14
<b>2.5 Tecnologia <i>Bluetooth</i>.....</b>	<b>15</b>
<b>2.6 Trabalhos Correlatos.....</b>	<b>15</b>
<b>3 METODOLOGIA.....</b>	<b>18</b>
<b>3.1 Descrição do <i>kit</i> robótico <i>LEGO MINDSTORMS</i>.....</b>	<b>19</b>
<b>3.2. Descrição do Ambiente.....</b>	<b>20</b>
<b>3.3 Descrição do Sistema de Navegação.....</b>	<b>21</b>
<b>3.4 O Sistema de Visão Computacional.....</b>	<b>23</b>
3.4.1 Aquisição de Imagens.....	23
3.4.2 Conversão RGB para HSI .....	24
3.4.3 Detecção dos Elementos.....	24
3.4.4 Filtro Gaussiano.....	25
3.4.5 Limiarização e Binarização.....	25
3.4.6 Matriz de posição.....	26
<b>3.5 O Algoritmo da Trajetória.....</b>	<b>26</b>
<b>3.6 Execução por Parte do Robô.....</b>	<b>27</b>

<b>4 IMPLEMENTAÇÃO E RESULTADOS PARCIAIS.....</b>	<b>29</b>
<b>5 CONSIDERAÇÕES FINAIS.....</b>	<b>34</b>
<b>REFERÊNCIAS .....</b>	<b>35</b>

# 1 INTRODUÇÃO

## 1.1 Motivação

A automação como tecnologia está preocupada com o uso da mecânica, elétrica, eletrônica e sistemas baseados em controle por computador para substituir os seres humanos por máquinas, não apenas em trabalhos físicos, mas também em processamento inteligente de informação, com o intuito de facilitar e diminuir o tempo empregado nas atividades e preservar a integridade humana (MITTAL & NAGRATH, 2003).

A Navegação Robótica tem sido alvo de muitos estudos, pois auxilia na solução de diversos problemas em diferentes áreas. Ela pode ser utilizada, por exemplo, para sensoriamento, para desarmamento de bombas, condução de veículos robotizados, exploração de ambientes hostis, e até assessoria de cadeiras de rodas (RUSSELL, NORVIG, 2004; HEINEN, 2007). Todos esses sistemas possuem em comum a capacidade de perceber o ambiente no qual estão inseridos e de modo semi ou completamente automático obtêm os comandos necessários à sua locomoção, sem colocar em risco a sua integridade ou a dos demais elementos presentes (HEINEN, 2007).

As soluções para essas diferentes áreas de aplicação são distintas, vários mecanismos para o controle de robôs móveis têm sido utilizados tais como sensores de ultra-som, *lasers* e câmeras.

Nas últimas décadas, visão computacional tem se tornado uma das tecnologias mais baratas e promissoras para que um robô possa perceber o mundo físico, com aplicações como localização, construção automática de mapas, navegação autônoma, rastreamento de trajetória, inspeção, monitoramento ou detecção de situações de risco. Técnicas de navegação visual têm sido aplicadas em todos os tipos de ambientes e com todos os tipos de robôs (BONIN-FONT *et al*, 2007).

Quando se trata de navegação, ao incorporar vários tipos de sensores ao robô, pode-se aumentar seu grau de autonomia e inteligência, em contrapartida, os tipos e quantidades de sensores determinam o volume de dados gerados para o processamento de um mapa do ambiente, o que pode impossibilitar sua aplicação em tempo real e gerar um alto custo, acabando por inviabilizar o projeto.

Neste trabalho foi utilizado como dispositivo de entrada somente uma câmera, o que diminui o fluxo de dados e o custo computacional e atende às necessidades do projeto.



Conforme Josef Pauli (PAULI, 2001 *apud* BONATO, 2004), não existe um sistema de visão computacional genérico capaz de solucionar todos os problemas relativos à visão computacional, apesar de atualmente existirem diversas técnicas difundidas sobre o assunto. Assim, o projetista escolhe uma dentre as já existentes, aquela que mais se adéqua à solução de seu problema, ou então faz umas adaptações de forma a atender às suas necessidades.

## **1.2 Problema**

Desenvolver um sistema de visão computacional para auxiliar na navegação de um robô autônomo em um ambiente onde os obstáculos mudam de posição enquanto o robô realiza seu trajeto, tendo um ponto de partida e um alvo fixo como ponto de chegada. É uma tarefa que impõe dificuldades quanto aos aspectos de monitoramento do ambiente – em mudança contínua com os deslocamentos do robô e dos obstáculos – e quanto à eficiência do sistema de navegação que deve transmitir ao robô informações de sua trajetória a tempo de permitir o desvio dos obstáculos.

No escopo deste problema, este trabalho vai tratar uma versão em escala menor, ou seja, estritamente controlada, onde o robô móvel, o alvo de chegada e os obstáculos são definidos por cores distintas, visíveis por uma câmera posicionada ortogonalmente em relação ao plano da cena, mas que ainda assim apresenta características reais de um ambiente e trata a solução de problemas reais do tema.

A criticidade está na particularidade do sistema em ser ágil na transmissão automática de dados, onde os comandos devem ser instantaneamente enviados para o robô, fazendo com que sua execução aconteça em tempo suficiente para a sua navegação.

## **1.3 Objetivos**

### **1.3.1 Objetivo Geral**

Desenvolvimento de um método computacional, utilizando visão computacional, para navegação de um robô autônomo não tripulado em ambiente com obstáculos móveis, baseando-se em imagens capturadas a intervalos regulares para tomada de decisões inerentes à sua trajetória.

### 1.3.2 Objetivos específicos

- Elaboração do algoritmo de planejamento de trajetória;
- Integração do sistema a um processo automático de transmissão de dados;
- Elaboração de um artigo científico.

## 1.4 Abordagem para solução do problema

O sistema tem o intuito de auxiliar na navegação autônoma de um robô, que deverá seguir uma trajetória entre um ponto de partida e um alvo, buscando o menor caminho e desviando-se de obstáculos com posição dinâmica, o que restringe a sua atuação para agentes móveis que se locomovem com a utilização de rodas, em ambientes planos, e com a possibilidade de obtenção de imagens do espaço a ser navegado contendo informações sobre a estrutura do lugar, como a posição do alvo e de possíveis obstáculos.

O robô escolhido para a implementação do método e realização de testes foi o *Kit robótico da linha Lego Mindstorms* destinado à construção de pequenos robôs, devido às facilidades que a sua mecânica confere, possibilitando privilegiar a parte computacional do projeto, o que não exclui o uso do método em outras plataformas.

Quanto ao uso de dispositivos para percepção do ambiente, adotou-se somente o uso de uma câmera digital como sensor, pois viabiliza a extração de características do ambiente sem a necessidade do contato físico. As imagens capturadas devem ser da vista superior do ambiente. Usando tais informações, o trajeto é calculado por um algoritmo de busca que, a partir de funções heurísticas, tenta escolher o melhor caminho, considerando as particularidades do ambiente.

O método proposto está dividido em duas etapas, sem que as duas sejam independentes entre si: a primeira é a aquisição e processamento de imagens, onde as imagens capturadas passam por técnicas básicas de processamento, como conversão de modelo de cor de RGB para HSI, binarização, e Filtragem Gaussiana para remoção de ruídos, como será detalhado na seção 3; a segunda etapa trata justamente do desenvolvimento do algoritmo de busca integrado com a função de geração e transmissão automática de código via *bluetooth* para o robô, o que pode ser realizado com base nas informações extraídas na primeira etapa.

O sistema de visão proposto tem em vista solucionar um problema de navegação, espera-se que utilizando técnicas de processamento de imagens mais simples e

conseqüentemente algoritmos mais rápidos, o sistema será capaz de apresentar um bom desempenho. Tecnologia desta espécie será útil para auxiliar no sensoriamento de regiões de difícil acesso.

O aplicativo de comunicação remota foi desenvolvido através de uma biblioteca auxiliar e contou com o apoio do projeto de pesquisa “**Estudo da Transferência Automática de Dados para um Robô Móvel na Plataforma Lego Mindstorms**”.

### 1.5 Cronograma de atividades

<b>CRONOGRAMA DE EXECUÇÃO DAS ATIVIDADES POR META</b>													
Meta/Atividades	Período												
	2010					2011							
	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Ju.	
Implementação do sistema de navegação	X	X	X										
Desenvolvimento de um sistema de transmissão de dados				X	X								
Integração entre o sistema de transmissão e o de navegação						X	X						
Experimentos práticos para validação							X	x	x	x	x		
Escrita de artigo científico									x				x
Monografia											x		x
Defesa Final													

X- Atividades executadas.

**Figura 1 - Cronograma de Execução das Atividades por Meta.**

### 1.6 Organização do Trabalho

A Seção 2 apresenta a fundamentação teórica necessária para o desenvolvimento do trabalho. São apresentadas definições sobre robótica móvel, navegação autônoma de robôs com ênfase em robôs terrestres, um breve discurso sobre visão computacional e processamento digital de imagens, conceitos básicos de imagem digital e cor, bem como

noções sobre tecnologia *Bluetooth*. A Seção se encerra com uma abordagem de trabalhos relacionados com a área de navegação autônoma de robôs.

A Seção 3 descreve a metodologia usada para a evolução do trabalho. Detalha as etapas percorridas e as características do modelo de navegação proposto, o sistema de visão, as técnicas de processamento digital utilizadas, e descreve o ambiente utilizado durante a simulação e do *kit* robótico escolhido.

A seção 4 versa sobre os resultados do funcionamento do sistema proposto. Os problemas encontrados e as considerações finais serão descritos na seção 5.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Robótica Móvel

O termo robótica móvel refere-se ao estudo e à utilização de robôs com capacidade de locomoção (RUSSELL; NORVIG, 2004).

Os avanços em robótica beneficiam diretamente diferentes áreas de conhecimento, como a Medicina, onde manipuladores auxiliam cirurgiões e a Engenharia, em fábricas de automóveis, linha de montagem industrial ou numa estação espacial, trazendo para elas a possibilidade de execução de uma tarefa com mais exatidão e segurança.

“Os robôs são agentes físicos que executam tarefas manipulando o mundo físico.” (RUSSELL; NORVIG, 2004, p. 870).

Além de manipuladores robóticos, existem também estudos de agentes que recebem percepções do ambiente e executam ações. Os robôs móveis são aqueles que, após receberem estímulos do meio, podem deslocar-se por um ambiente usando rodas, pernas ou dispositivos semelhantes e podem ainda ser equipados com sensores, que lhes permitem perceber o ambiente em que estão inseridos; o robô não pode ver através de obstáculos e comandos de movimentação estão sujeitos à robustez do seu sistema operacional (RUSSELL; NORVIG, 2004).

É admissível que os sistemas robóticos práticos incorporem conhecimento prévio sobre as características do robô, seu ambiente físico e as tarefas que ele executará, bem como realizem testes auxiliados por simuladores, caso contrário, possíveis erros poderiam trazer danos aos dispositivos robóticos, além do fato de que testes realizados fora de ambientes simulados poderiam ser dispendiosos e demorados (RUSSELL; NORVIG, 2004).

Robôs com rodas são bastante eficientes em superfícies planas e regulares, mas em ambientes com algumas irregularidades eles precisariam de mecanismos de locomoção mais complexos como pernas, que lhes permitiriam o livre deslocamento sobre inclinações, desníveis e degraus (HEINEN, 2007).

É importante ressaltar que, embora solucionar o problema da navegação de robôs em superfícies irregulares seja algo célebre para a comunidade científica, muitas das pesquisas ainda se concentram no deslocamento em superfícies regulares, pois ainda é um problema que não foi completamente solucionado (HEINEN, 2007).

## 2.2 Navegação Autônoma de Robôs Terrestres

Dentro das diversas áreas de pesquisa em robótica autônoma, existem tópicos que são de fundamental importância, a saber: localização instantânea do robô, planejamento de trajetórias e navegação (HEINEN, 2002; RIGHES, 2004).

Navegação Autônoma é a propriedade que um dispositivo (robô) tem de mover-se até o seu objetivo sem intervenção humana (HOFFMANN *et al*, 2004).

A pesquisa em Navegação Autônoma visa desenvolver métodos que permitam a operação não supervisionada de veículos em situações e em lugares onde a presença humana se torna difícil, arriscada e até mesmo impossível, como o fundo do mar ou o espaço e em ambientes que não permitem o controle à distância devido a fatores diversos (OLIVEIRA, 1995).

Em navegação, a localização é um dos problemas mais recorrentes, tem-se que determinar onde estão os objetos, pois o conhecimento de suas posições é o núcleo de qualquer interação física bem-sucedida, por exemplo, os robôs de navegação têm de saber onde se localizam a fim de encontrar seu caminho até as posições-objetivo. Se a posição inicial do objeto a ser localizado (alvo) for conhecida, a localização será um problema de rastreamento, que se caracteriza pela incerteza limitada. (RUSSELL; NORVIG, 2004).

O maior problema em sistemas de navegação é identificar e ou prever os obstáculos que se encontram entre o ponto de partida e o alvo, que no caso simples considera obstáculos com posição fixa, mas na prática os ambientes não estão livres de obstáculos que têm posição dinâmica, o que torna a solução não trivial, uma vez que será necessário perceber o ambiente a cada dado instante de tempo e replanejar a trajetória do robô.

## 2.3 Visão Computacional e Processamento Digital de Imagens

A Visão Computacional é a ciência e a tecnologia que permite que os sistemas computacionais sejam capazes de interpretar o mundo físico através de imagens (MITTAL; NAGRATH, 2003). É a área da ciência que se dedica a desenvolver teorias e métodos voltados à extração automática de informações úteis contidas em imagens (TRUCCO, MURINO, 2000).

Uma das aplicações mais significativas envolvendo um sistema de visão computacional é a análise ou interpretação de imagens, cujo objetivo elementar é obter uma descrição com informações suficientes para distinguir entre diferentes objetos de interesse, de forma confiável e requerendo o mínimo de intervenção humana (PEDRINI; SCHWARTZ, 2008).

A visão é uma característica auxiliar na navegação autônoma de robôs, possibilita a percepção do ambiente através do uso de imagens adquiridas a partir de uma ou mais câmeras. As imagens permitem que se utilizem técnicas de determinação de posicionamento relativo e absoluto de estimativa de deslocamento, assim como detecção de obstáculos. Um sistema robótico móvel pode ter diferentes níveis de autonomia e inteligência, o que será definido pelo tipo de sistema de controle integrado ao sistema (JUNG *et al*, 2005).

Sistemas que usam visão para navegação estão classificados naqueles que precisam de conhecimento prévio do ambiente a ser navegado e aqueles que percebem o ambiente enquanto navegam por ele (BONIN-FONT *et al*, 2007).

“O processamento digital de imagens consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio de computador” (PEDRINI, SCHWARTZ, 2008, p. 1). A aplicação de tal técnica permite identificar informações das imagens e melhorar a qualidade visual de certos aspectos, envolve operações como redução de ruídos e aumento de contrastes, entre outras.

## 2.4 Fundamentos de Cores e Imagem Digital

O uso de cores em processamento de imagens é motivado, entre outras razões, pelo fato de que a cor é um descritor que freqüentemente simplifica a identificação do objeto e a extração de uma cena (GONZALEZ, WOODS, 2000).

“A cor é uma manifestação perceptual da luz que, por sua vez, é um sinal eletromagnético. Um raio luminoso é constituído de partículas chamadas *fótons*, cuja freqüência e velocidade determinam o comprimento de onda dos *fótons*” (PEDRINI; SCHWARTZ, 2008, p. 467). Os diferentes comprimentos de onda estão associados a cores distintas, do ponto de vista perceptual. Existe um intervalo de comprimentos de onda que é capaz de sensibilizar o olho humano, fazendo com que o cérebro interprete diversas cores e tonalidades que vão do violeta ao vermelho (PEDRINI; SCHWARTZ, 2008).

Uma imagem digital pode ser considerada como sendo uma matriz cujos índices de linhas e de colunas identificam um ponto na imagem, e o correspondente valor numérico do elemento da matriz representa a cor do *pixel* – menor elemento da imagem digital (GONZALEZ, WOODS, 2000; PEDRINI, SCHWARTZ, 2008).

Uma imagem colorida é uma imagem multibanda onde a cor de cada *pixel* é representada de acordo com um modelo de cor. Esses modelos de cores têm o propósito de facilitar a especificação das cores em alguma forma padrão para enfatizar certas características, mas serão ressaltados apenas os modelos de cores RGB e HSI que são os utilizados no trabalho.

#### 2.4.1 Modelo de cor RGB

“Um modelo de cor é uma especificação de um sistema de coordenadas tridimensionais e um subespaço dentro deste sistema onde cada cor é representada por um único ponto” (GONZALEZ, WOODS, 2000, p160).

No modelo RGB, (R, do inglês *red*; G, do inglês *green*; B, do inglês *blue*) as imagens consistem em três planos de imagem independentes, um para cada cor primária: vermelho, verde e azul; esses três planos combinam-se para produzir uma imagem de cores compostas (GONZALEZ, WOODS, 2000; RIGHES, 2004; PEDRINI, SCHWARTZ, 2008).

A maioria das câmeras coloridas usadas para aquisição de imagens digitais utiliza o formato RGB de cores, por isso a ênfase dada a esse modelo especificamente.

#### 2.4.2 Modelo de cor HSI

Do inglês *Hue* (matiz), *Saturation* (saturação), *Intensity* (intensidade). Matiz é um atributo que descreve a cor propriamente dita, como percebida por um observador; quando chamamos um objeto de verde, amarelo, ou azul estamos especificando seu matiz. Saturação refere – se à pureza relativa da cor, ao grau de diluição de uma cor pura com o branco; as cores do espectro puro são completamente saturadas, cores como lilás (violeta e branco) e azul claro (azul e branco) são menos saturadas, assim de um modo simples, pode-se dizer que a saturação indica quão forte é a cor. De maneira sucinta, a componente intensidade informa se a cor está mais próxima do branco ou preto, denota a quantidade de luz (GONZALEZ, WOODS, 2000; RIGHES, 2004; PEDRINI, SCHWARTZ, 2008).



## 2.5 Tecnologia Bluetooth

*Bluetooth* é um padrão global de comunicação sem fio e de baixo consumo de energia que provê uma maneira de conectar e trocar informações entre dispositivos eletrônicos através de uma frequência de rádio de curto alcance, permitindo que um dispositivo detecte o outro independente de suas posições, desde que estejam dentro do limite de proximidade. (BLUETOOTH, 2011).

A tecnologia é útil quando é necessária transferência de informações entre dois ou mais dispositivos que estão perto um do outro ou em outras situações onde não é necessária alta taxa de transferência.

Um computador pessoal deve ter um adaptador *Bluetooth* instalado para poder se comunicar com outro dispositivo *Bluetooth*. Enquanto alguns computadores *desktop* já contêm um adaptador instalado internamente, a maioria requer um *dongle Bluetooth* (conhecido como "chaveiro *bluetooth*"). Ele permite a comunicação com vários dispositivos ao mesmo tempo apenas com um único adaptador (BLUETOOTH, 2011).

## 2.6 Trabalhos Correlatos

Várias propostas são encontradas na literatura para solução do problema de navegação robótica, nesta seção são catalogados alguns sistemas que consistem em maneiras diferentes de resolver o problema de navegação e outros que deram suporte na elaboração do trabalho.

Em seus trabalhos, (DeSouza; Kak, 2002) e (BONIN-FONT *et al*, 2007) fizeram um levantamento (*survey*) a respeito de navegação visual para robôs móveis, onde eles listam uma série de trabalhos, a partir da década de 90, que constituem as maiores contribuições de soluções para localização baseada em visão aplicada e desenvolvidas para veículos autônomos terrestres, aéreos e subaquáticos. Os temas navegação ao ar livre (*outdoor navigation*) e navegação em ambientes fechados (*indoor navigation*) e suas subdivisões também estão inclusos.

Em (PINAGÉ, 2009) desenvolveu-se um sistema que permite, com apenas uma imagem do ambiente, a detecção de uma trajetória em que o robô atinge um determinado alvo sem a interferência de obstáculos. Adotou-se o uso de uma câmera posicionada acima do ambiente e técnicas para extração das posições do robô, obstáculos e do alvo. O autor valida o

método utilizado afirmando sucesso na busca por trajetória, contudo o sistema desenvolvido é eficaz para ambientes com obstáculos fixos, mas, não possui solução para ambientes com obstáculos móveis.

O presente trabalho teve como referencial o sistema apresentado em (PINAGÉ, 2009) com o intuito de dar continuidade ao mesmo no sentido de lhe conferir um grau de robustez maior permitindo a navegação em ambientes com obstáculos de posição dinâmica, diferindo do mesmo por possuir a particularidade de replanejar a trajetória para o robô a cada passo, para que haja a identificação de obstáculos não encontrados desde a verificação anterior. Antes a trajetória era planejada uma única vez, e se surgisse um obstáculo não esperado no trajeto encontrado o robô se chocaria.

Em (LIBRANTZ, ARAÚJO, KOYOMA, 2006) e em (ARAÚJO, LIBRANTZ, FLÓRIO FILHO, 2006) o método de navegação descrito é baseado em visão computacional e foi usado como suporte no desenvolvimento deste projeto. Uma imagem colorida do ambiente é capturada e armazenada, esta imagem é então convertida para níveis de cinza, binarizada e transformada em uma matriz para determinar a localização do robô, alvo e obstáculos. Finalmente, um algoritmo de roteamento faz uso da matriz para calcular o trajeto a ser percorrido até o alvo. Apesar de obter resultados significativos, o mesmo não é aplicável em ambientes com obstáculos móveis.

Em (RIGHES, OSÓRIO, 2005) a solução implantada para navegação se deu através da comparação entre duas imagens (imagem prévia do caminho x imagem atual capturada pelo robô) estabelecendo uma correlação entre elas. Esta comparação busca identificar referências nas imagens, estabelecendo o quanto o robô está deslocado em relação à trajetória previamente definida. A solução apresentou-se funcional e foi possível validar o uso da técnica proposta através de experimentos práticos. Entretanto, devido ao fato das imagens utilizadas estarem no formato RGB, foi constatado que será necessário melhorar a robustez do sistema quanto a variações de luminosidade da cena e quanto à velocidade de processamento (o custo computacional é mais alto devido ao processamento das três componentes de cor da imagem).

Em seu projeto de módulo de aquisição e pré-processamento de imagem colorida baseado em computação reconfigurável e aplicado a robôs móveis (BONATO, 2004) utilizou uma imagem RGB, mas encontrou dificuldades, pois esse modelo é muito sensível a alteração da iluminação, por isso utilizou-se da conversão RGB para o modelo HSI, como realizado neste trabalho.

Em (NETTO, WOLF, 2007) há uma abordagem sobre navegação em ambientes externos utilizando processamento de imagens. Imagens foram adquiridas por uma câmera monocular e posteriormente processadas por técnicas de segmentação baseadas em estatística, o segundo passo foi a implementação, em C, de um programa que utiliza a imagem segmentada para definir a trajetória do robô que é definida tentando manter o robô no centro da via, através da contagem do número de pixels. Os deslocamentos do robô foram compatíveis com o que se esperava baseado nas informações visuais.

Em (MIRANDA NETO, 2007) o sistema de navegação desenvolvido considera apenas informações oriundas de um sistema de visão computacional monocular, isto é, um sensor câmera adquire imagens do ambiente e estas alimentam o sistema computacional. Embora seja bem econômico, possui por desvantagem a não obtenção de informações de profundidade da imagem, limitando percepção do sistema.

Em (ALVES, 2006) implantou-se um sistema de visão computacional para navegação de robôs moveis baseado em rastreamento de trajetória através de marcas artificiais adicionadas ao teto do ambiente em forma de linhas e curvas, para tanto se admitiu um sistema de rastreamento que trabalha concorrentemente com o sistema de controle do robô. Utilizou-se de um robô móvel equipado com uma câmera com foco para o teto, dentre as técnicas utilizadas destaca-se a de odometria, extração de bordas e redução de ruídos na imagem. Uma restrição do sistema apresentada pelo autor corresponde a um importante problema em rastreamento de trajetórias para navegação de robôs moveis, é a determinação do ponto de partida da trajetória, essa restrição consiste em um empecilho para total automatização da tarefa de navegação do robô.

Em (OLIVEIRA, 1995), o problema de navegação autônoma foi resolvido com uma proposta baseada em paradigmas de computação flexível: redes neurais artificiais, sistemas nebulosos e algoritmos genéticos, o que conferiu ao sistema propriedades como localidade de operação e referencia local e aquisição dinâmica de conhecimento. Os resultados dos experimentos realizados indicaram que o navegador proposto não foi capaz de aprender simultaneamente a evitar obstáculos e encontrar alvos, embora a evolução separada desses dois comportamentos tenha sido bem sucedida, indicando que o algoritmo de aprendizagem pode ser aplicado com êxito a esta espécie de problema.

### 3 METODOLOGIA

O presente trabalho dá continuidade ao de Pinheiro *et al* (2010), e portanto a metodologia segue a partir de um estágio já avançado em relação ao trabalho anterior, no sentido de lhe conferir um maior grau de robustez. Apesar do núcleo principal se encontrar em funcionamento, o mesmo ainda dependia de ações manuais para aquisição das imagens, e a comunicação entre o sistema de controle e o robô ainda era muito limitada, restringindo a velocidade para captura de informações do ambiente, processamento e resposta do robô.

Diante do problema de navegação autônoma, aspectos específicos devem ser contemplados no sistema de controle que rege o veículo robótico para que o deslocamento do robô se dê com eficácia e sem colocar em risco a sua integridade e a dos demais elementos presentes. Para este trabalho, foram escolhidas determinadas etapas, discriminadas a seguir, compondo as especificações do projeto a serem cumpridas para a realização de testes e para que o software possa ser validado.

No caso geral de navegação (aérea, subaquática ou terrestre) o robô tem que saber sua localização em relação ao ambiente para saber quais serão as providências que terá de tomar para desempenhar suas habilidades de maneira adequada, além de obter essa e as demais informações necessárias como a existência de obstáculos e a localização de um alvo de maneira autônoma (RUSSEL; NORVIG 2004; HEINEN, 2007).

Para sistemas de controle que utilizam visão computacional, a forma de percepção do ambiente é através de imagens ou vídeos. No caso da utilização de imagens (maneira adotada no projeto), a leitura e interpretação das mesmas devem ser de modo automático e com um tempo de resposta que permita uma navegação segura, sem dar margem a possíveis choques com obstáculos não esperados.

Os aspectos dinâmicos do problema são essenciais, rastrear uma trajetória livre de obstáculos e recalculá-la diante de uma adversidade é uma característica fundamental para corresponder às expectativas de navegação autônoma, essa é a etapa principal que deve ser contemplada de forma confiável.

O *software* de navegação não está acoplado ao robô. O sistema de controle está em um computador pessoal e a presença do robô se faz necessária apenas na fase de execução. Como o dispositivo de controle e o de execução não estão conectados, existe a necessidade de integrar as duas partes, e para que haja maior autonomia, opta-se pela comunicação remota,

que é possível devido ao robô utilizado para implementação possuir um adaptador *bluetooth* instalado internamente.

Para validação do método, uma última fase é essencial para se tirar conclusões a respeito da sua eficiência, que é a fase de testes experimentais. O sistema será submetido a diversas situações de variação da localização dos obstáculos e será analisado seu desempenho: ele deverá prover ao robô a trajetória correta do ponto de partida até o alvo, levando em consideração o posicionamento dos obstáculos e o mapa gerado através das imagens adquiridas do ambiente; deverá ainda permitir que o robô execute os comandos gerados com a devida eficácia e em um tempo considerado aceitável, aquele que permite que o robô se locomova sem se chocar com nenhum obstáculo.

Neste capítulo são descritos os passos realizados para concepção do sistema. Na seção 3.3 é exposto o sistema de navegação, que relata quais dados de entrada devem ser fornecidos e quais as etapas a serem seguidas para se obter o resultado esperado - as coordenadas a serem seguidas pelo robô para navegar sem se chocar com obstáculos em movimento.

Na seção 3.4 é descrito o sistema de visão explanando a quais técnicas de processamento digital as imagens de entrada devem ser submetidas para que os elementos do ambiente físico sejam identificados.

O algoritmo da trajetória é apresentado na seção 3.5, ele rastreia o menor caminho livre de obstáculos até o alvo e a seção 3.6 explica como se dá a transmissão de comandos para o robô.

Utiliza-se as linguagens C e C++ e ainda uma biblioteca de processamento de imagens chamada *FreeImage* com licença GLP (FREEIMAGE, 2011).

Para facilitar os testes em laboratório, simulou-se um ambiente com iluminação bem definida, fundo fixo (preto) e posição dos objetos dinâmica. O método computacional define-se com as etapas de aquisição de imagens do ambiente em questão, processamento dessas imagens para retirar informações úteis à descrição do lugar, e a busca de um percurso mais curto até o alvo. Antes de descrever os passos da metodologia, faz-se necessário descrever as particularidades do *Kit* robótico e do ambiente, respectivamente nas seções 3.1 e 3.2.

### **3.1 Descrição do *kit* robótico *LEGO MINDSTORMS***

Adotou-se o *Kit* robótico da linha *LEGO MINDSTORMS* destinado à construção de pequenos robôs. Estes *kits* são compostos por pequenos blocos de plástico (peças tradicionais

*LEGO*) e outras peças complementares como servomotores, sensores de toque, som, luz e ultrassom, bem como uma unidade de controle programável - *NXT brick* (MINDSTORMS, 2009).

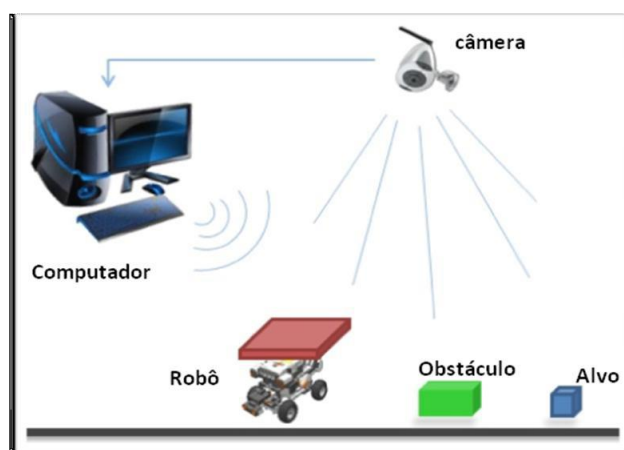
O *NXT brick* permite a comunicação via *Bluetooth*, e a linguagem de programação do seu sistema operacional é o *NXC (Not eXactly C)* que é uma linguagem simples da plataforma *Lego* e o *software* para programação utilizado foi o *BricX Command Center*. (HANSEN, 2007; BENEDETTELLI, 2007).

Ele foi montado conforme o modelo veículo com três rodas, com dois servomotores, mas nenhum sensor do *kit* foi utilizado.

### 3.2 Descrição do Ambiente

O ambiente de navegação é controlado, montado de forma a simular um ambiente real com obstáculos móveis. A frequência com que os obstáculos se movem não é constante e sem um tempo definido para acontecer.

O cenário consiste em um ambiente plano onde estão presentes o robô, os obstáculos dispostos aleatoriamente e um alvo também disposto aleatoriamente. As dimensões do cenário devem ser proporcionais ao número de pixels da imagem a ser capturada. Uma câmera deve ser posicionada perpendicularmente ao ambiente, permitindo uma vista superior de todos os elementos constituintes do cenário, conforme mostrado na Figura 2.



**Figura 2-** Arranjo esquemático do cenário apresentando os elementos que devem estar presentes. A câmera superior ao ambiente captura e transmite as imagens ao computador que possui o sistema que então envia os comandos para o Robô via *Bluetooth*.

Para cada elemento do ambiente define-se uma cor, essas cores específicas foram escolhidas por serem as cores primárias do modelo RGB: vermelho para o robô, verde para obstáculos e azul para o alvo e um fundo preto. Cada elemento é coberto com um papel da cor previamente selecionada.

### 3.3 Descrição do Sistema de Navegação

Para se locomover em ambiente com obstáculos móveis, o robô utilizará o sistema de controle e navegação proposto em (PINHEIRO *et al*, 2010) conforme Figura 3, que é definido desde a montagem do cenário de navegação até o envio do comando para a execução da trajetória pelo robô.

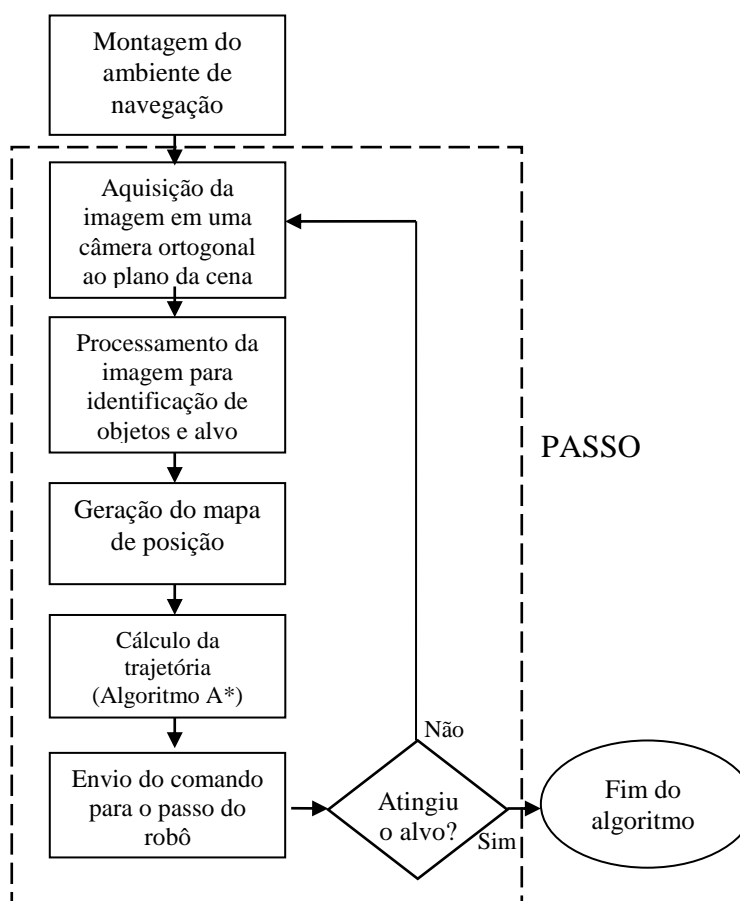


Figura 3 - Fluxograma do sistema de navegação autônoma proposto.

Adaptado de (PINHEIRO *et al*, 2010)

Sabendo que os obstáculos não são fixos, o processo de mudança de posição pode ser visualizado como uma série de instantâneos, cada um dos quais descreve o estado do trajeto em um instante específico. Cada instante deve ser verificado para ver como a mudança deve interferir na ação que está sendo executada, no caso, a navegação.

O método proposto consiste em obter dados em instantes definidos (passo) e gerar uma trajetória parcial para o robô em direção ao alvo, que será obtida novamente a cada passo do robô até atingir o objetivo.

Primeiramente, é feita a montagem do cenário onde se pretende a navegação autônoma (ambiente simulado), o qual é um ambiente plano de fundo fixo preto que deve conter elementos que representam os obstáculos, o alvo e o robô cobertos com cores distintas e dispostos aleatoriamente.

Uma câmera deve ser posicionada perpendicularmente ao plano da cena, permitindo uma vista superior de todos os elementos constituintes do cenário. É estabelecido um tempo de realimentação para o sistema e é feita a captura de imagens pela câmera.

Para alimentar o Sistema de Navegação, a cada passo uma nova imagem é capturada e é realizado um processamento digital para, através de uma matriz de posições, identificar os obstáculos - que mudam de posição enquanto o robô navega - o alvo a ser alcançado e o próprio robô, enquanto ele realiza o trajeto.

A etapa de processamento de imagens gera como saída uma imagem binarizada dividida em células (grade), onde cada célula tem dimensão de 100x100 pixels. A grade é transformada em um mapa de posição em forma de matriz, cada célula da grade na imagem equivale a um valor na matriz, a qual tem o mesmo número de linhas e colunas da grade, e cada célula é proporcional ao tamanho do robô.

O valor que cada posição na matriz assume é determinado pelo objeto que ocupa aquela posição. Fixou-se 9 (nove) para obstáculos, 3 (três) para o robô, 2 (dois) para o alvo e 1 (um) para células livres. A célula é considerada ocupada caso tenha pelo menos parte de um elemento nela.

O algoritmo de busca utilizado é o A\* (A-estrela), onde as heurísticas fornecem informações sobre a qualidade de cada estado, de forma que a procura é realizada primeiro nas rotas mais promissoras (HEINEN, 2002). A matriz gerada no processamento anterior é passada para este algoritmo A\* que, com base nas informações contidas nela, calcula o trajeto mais curto que o robô fará para chegar ao alvo. O algoritmo de busca fornece em coordenadas



cartesianas os movimentos necessários que o robô deve seguir para manter a trajetória até o alvo.

Tendo a expectativa de onde está o alvo, e considerando que o mesmo é fixo, o sistema irá controlar diretamente o robô via *Bluetooth* com o exato comando a ser executado, ligando seus servomotores para conduzir o robô à direita, à esquerda ou para frente, para isso, alterando a angulação da trajetória de 90° em 90° (trajetória definida através das coordenadas cartesianas fornecidas pelo A\*).

A cada passo do sistema de navegação uma nova trajetória é obtida e um novo comando é executado pelo robô, que se move de uma célula à próxima, considerando as novas posições dos obstáculos e a trajetória gerada. Se as coordenadas do passo a ser realizado coincidirem com as coordenadas em que está o alvo, entende-se que foi encontrada a trajetória até o alvo satisfazendo o objetivo e determinando o fim do algoritmo.

Essas particularidades certificam que o robô navegue pelo ambiente passo a passo sem se chocar com os obstáculos que mudam de posição a instantes indefinidos.

### **3.4 O Sistema de Visão Computacional**

O sistema de visão assume as etapas seguintes: Aquisição de imagens em formato RGB do ambiente onde o robô precisa navegar a instantes definidos e constantes, que passam por algumas técnicas de processamento de imagens para redução de ruídos e para auxiliar na segmentação e detecção do robô, alvo e obstáculos; a partir da análise de cada imagem, é gerada uma matriz correspondente. A análise dessa matriz capacita o algoritmo de busca a gerar uma trajetória.

Cada etapa será aprofundada a seguir.

#### **3.4.1 Aquisição de Imagens**

Uma câmera comum deve capturar imagens enquanto o robô navega, a partir das imagens, o sistema computacional é capaz de perceber o mundo físico, interpretá-lo e tomar decisões através das informações obtidas.

Como os obstáculos podem variar suas posições, o sistema deve ser realimentado para atualizar as novas posições dos elementos variantes, bem como a nova posição do robô em relação a eles.

Uma a uma, são capturadas imagens coloridas com a vista superior do ambiente, com o robô já presente. A imagem deve estar no modelo RGB de cores e ter dimensões de 640 x 480.

A partir da imagem capturada, o sistema de visão deve identificar, através de uma matriz de posições, a localização do robô, obstáculos e local de destino, realimentando com uma nova imagem sempre que o robô der mais um passo.

### 3.4.2 Conversão RGB para HSI

Para facilitar na localização de cada elemento, pode-se separar as bandas de cores de modo a destacar só o elemento que se quer enfatizar, a banda verde para obstáculos, a banda vermelha para o robô, e a banda azul para o alvo, como detalhado na seção 3.4.3, mas há um inconveniente no modelo RGB de cores que, apesar de haver facilidade na separação de cores, há muita sensibilidade quanto à iluminação, o que pode induzir à detecção de características errôneas da imagem. Para contornar esse inconveniente, o modelo HSI pode ser usado. Neste modelo, os componentes H, S e I do *pixel* são valores independentes o que torna possível realizar operações individuais (GONZALEZ, WOODS, 2000; BONATO, 2004).

Como a imagem de entrada está no formato RGB será feita uma conversão pixel a pixel para o formato HSI, conforme equações dadas em (PINAGÉ, 2009).

### 3.4.3 Detecção dos Elementos

Para detectar cada elemento do ambiente, foi feita a separação da cor atribuída a cada um deles.

O processo é feito após a conversão, quando é realizada uma varredura na banda H - que indica uma cor exata - cujos valores são dados em graus: o vermelho é indicado por 0 grau, o verde por 120 graus, e o azul por 240 graus. A varredura terá a finalidade de detectar onde na imagem existem *pixels* de cada uma das três cores para então separar em três imagens diferentes, uma para cada cor identificada.

Cada imagem contém os *pixels* da cor a ser enfatizada representados na cor branca, artifício utilizado para facilitar a interpretação por parte do sistema computacional, e os pixels das demais cores aparecem na cor preta.

Para garantir ainda mais precisão na segmentação dos elementos, também é feita uma varredura nas bandas S e I.

#### 3.4.4 Filtro Gaussiano

Devido a fatores diversos como a iluminação do ambiente, as imagens depois de digitalizadas podem apresentar falhas, essas falhas são chamadas de ruídos na imagem. A redução de ruídos é necessária em imagens para assegurar que as informações contidas sejam verdadeiras, para tornar as informações mais apuradas (BONATO *et al*, 2003).

O filtro gaussiano é aplicado nas três imagens obtidas na separação das bandas, processo necessário para eliminar possíveis características indesejáveis geradas durante o processamento, as quais podem induzir a erros de análise. Este filtro suaviza os valores associados a cada pixel. Esta ação minimiza ou até elimina informações indesejáveis geradas por fatores externos (BONATO *et al*, 2003).

#### 3.4.5 Limiarização e Binarização

Mesmo com o uso do filtro gaussiano as imagens ainda apresentam ruídos, apesar de estes terem sido suavizados. Com a limiarização e binarização há a eliminação total dos ruídos remanescentes.

Existem técnicas de segmentação de imagens, cujo objetivo é destacar os objetos de interesse do resto da imagem, uma das mais comuns é a limiarização (*thresholding*) que consiste na determinação da cor do pixel do fundo da imagem (escolha de um valor ótimo entre os níveis de cinza, de um limiar) (GONZALEZ, WOODS, 2000).

Para binarizar as imagens eliminando os ruídos remanescentes e fazendo o realce das áreas de interesse, utiliza-se um limiar para determinação da cor do *pixel*, se o *pixel* de entrada for maior ou igual ao limiar previamente determinado, o *pixel* de saída recebe o valor 255 (cor branca), se o pixel de entrada for menor que o limiar, o *pixel* de saída assume o valor 0 (cor preta).

Assim, as três imagens (uma contendo somente os elementos que eram da cor verde, outra somente com elementos da cor vermelha e ainda outra somente com elementos da cor azul) terão seus objetos em destaque representados pela cor branca e os demais estarão na cor preta.

### 3.4.6 Matriz de posição

Depois de binarizadas, as imagens são divididas em células (grade), onde cada célula corresponde exatamente a um passo a ser percorrido pelo robô.

A imagem dividida em células é transformada em uma matriz, que possui o mesmo número de linhas e colunas da grade e cada célula na imagem equivale a um valor na matriz.

O valor que cada posição na matriz assume é determinado pelo objeto que ocupa aquela posição. Fixou-se 9 (nove) para obstáculos, 3 (três) para o robô, 2 (dois) para o alvo e 1 (um) para células livres.

Para compor a matriz, as informações contidas nas três imagens são utilizadas, resultando em uma única matriz de posições contendo as posições do robô, alvo e obstáculos.

## 3.5 O Algoritmo da Trajetória

O algoritmo de busca utilizado é o  $A^*$  (A-estrela), onde as heurísticas fornecem informações sobre a qualidade de cada estado, de forma que a procura é realizada primeiro nas rotas mais promissoras (RUSSELL, NORVIG, 2004).

Para rastreamento de uma trajetória do local onde o robô se encontra até um destino especificado, o espaço de estados é representado como uma matriz e cada célula dessa matriz é um estado. As possibilidades possíveis são as oito células adjacentes ao robô (HEINEN, 2002; RUSSELL, NORVIG, 2004).

A etapa de processamento de imagens gera como saída três imagens binarizadas (as três componentes R-G-B) divididas em células (grade), e a grade é transformada em um mapa de posição em forma de matriz, na qual estão reunidas as informações das três componentes. A matriz gerada é passada para este algoritmo que calcula o trajeto mais curto que o robô fará para chegar ao alvo.

A função heurística usada foi:  $F=G+H$ . Onde  $H$  é o custo estimado do caminho que ainda falta percorrer;  $G$  é custo do caminho já percorrido;  $F$  a estimativa da distância total (RUSSELL, NORVIG, 2004). Vide Figura 4.

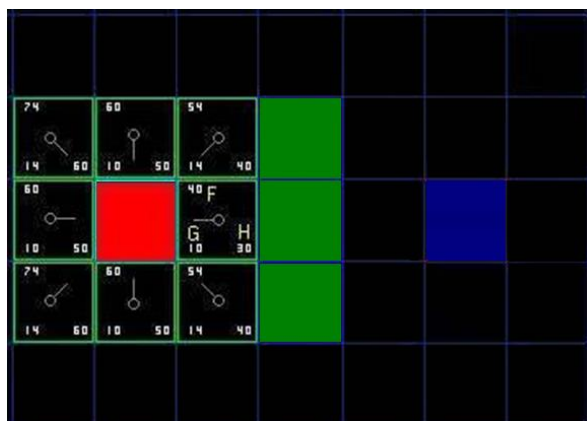


Figura4 - Algoritmo de trajetória. Figura adaptada de (A STAR, 2009).

Os movimentos a serem executados dependem do robô utilizado, neste caso são: esquerda, direita e frente. O algoritmo A\* atua do seguinte modo (A ESTRELA).

1. São guardadas as coordenadas da célula em que se encontra o valor 3 (robô), e este é o ponto de partida.
2. As células adjacentes ao ponto de partida formam uma lista aberta, elas são verificadas e, se contendo o valor 1, permitem a passagem do robô. A célula escolhida é a que possui o menor valor de F. Essa nova célula escolhida torna-se um novo ponto de partida.
3. A célula atual sai da lista aberta e passa para uma lista fechada, para não ser verificada novamente.

O algoritmo de busca fornece como saída passo a passo da trajetória que o robô deve seguir em coordenadas cartesianas.

### 3.6 Execução por Parte do Robô

As coordenadas cartesianas são analisadas pelo programa e os comandos dobrar à direita, dobrar à esquerda ou ir para frente (trajetória definida através das coordenadas) são enviadas para o robô. Existem duas maneiras de transmitir o pseudocódigo (chamado NXC-Not eXactly C - linguagem entendida pelo processador do robô *Lego Mindstorms NXT*) do compilador para o robô, via USB ou *Bluetooth*.

Realizar a transmissão via cabo USB é inviável, pois não permite mobilidade ao robô, visto que o mesmo ficaria conectado ao computador através do cabo, limitando seu perímetro de alcance e a sua livre navegação.

A escolha pela utilização do *Bluetooth* permite que assim que o sistema tomar a decisão do movimento a ser executado, o robô realiza instantaneamente o comando. O tempo de transmissão de dados é da ordem de 30 ms (TOLEDO, 2006).

Neste caso, é importante que o mesmo reproduza as ações necessárias ao percurso da trajetória. A cada leitura do sistema de navegação, um novo comando é gerado e executado pelo robô, considerando as novas posições dos obstáculos e a trajetória obtida.

## 4 IMPLEMENTAÇÃO E RESULTADOS

A metodologia proposta foi implementada utilizando um conjunto de robótica para área educacional, o *kit Lego Mindstorms NXT* e uma câmera digital colorida modelo *LifeCam VX7000*.

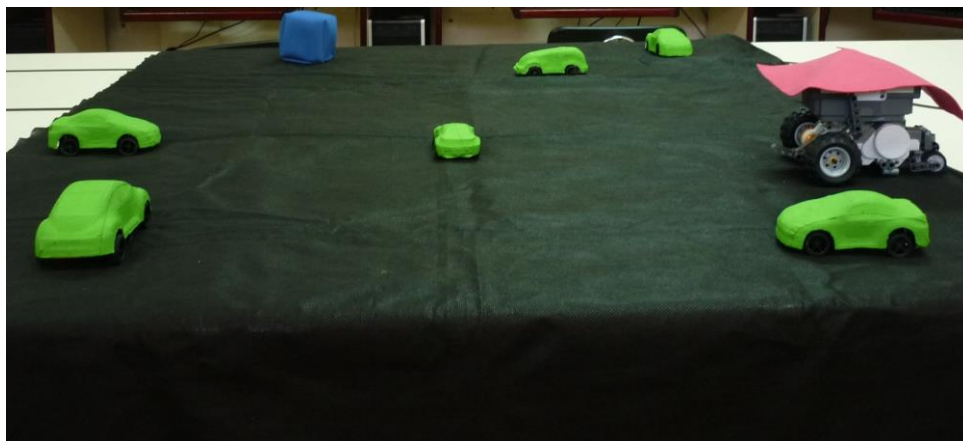
O sistema de navegação proposto utilizou para implementação as linguagens C e C++ e a biblioteca *Free Image* com licença GLP (FREEIMAGE, 2011).

Os testes foram desenvolvidos em um laboratório de pesquisa na área de robótica móvel no Campus Manaus Distrito Industrial do Instituto Federal do Amazonas, com um computador na plataforma PC 64 bits, sistema operacional Windows Seven, Pentium Dual-core 1.7Ghz, 2 GB de memória, e uma câmera digital colorida modelo *LifeCam VX7000*. O laboratório contém, ainda, uma mesa ao centro de tamanho 150x400mm onde foi montado o ambiente de navegação do robô, com o posicionamento inicial do alvo e obstáculos. Conforme Figura 5.



**Figura 5 - Ambiente de navegação com câmera ortogonal ao plano da cena.**

A realização dos testes para verificação do sistema de visão e da trajetória do robô consistiu primeiramente na organização do ambiente, distribuindo os obstáculos, o alvo e robô, todos cobertos com cores distintas: Robô na cor vermelha, alvo na cor azul e obstáculos na cor verde. Para contornar a inviabilidade de manipular os obstáculos a intervalos mínimos, o experimento realizado consistiu na montagem do ambiente conforme a Figura 6, e depois os obstáculos foram manipulados para ocuparem posições diferentes e uma imagem foi capturada para registrar cada mudança.



**Figura 6-Montagem esquemática do ambiente controlado.**

O sistema obteve o comando para o passo do robô e este atendeu instantaneamente ao comando. Foi colocada uma pausa no sistema para a realização da mudança das posições dos obstáculos que é desfeita pelo operador. O tempo de processamento do código é de 92 milissegundos, somando-se o tempo de execução por parte do robô, a navegação foi possível de acordo com as restrições impostas.

O sistema foi submetido a diversas situações de variação da localização dos obstáculos e foi analisado seu desempenho obtendo os seguintes resultados: No caso em que a passagem até o alvo esteja completamente obstruída pelos obstáculos, o robô fica imóvel; Em um ambiente com obstáculos móveis, mas sem um alvo, o sistema responde com uma mensagem de erro (“Alvo não encontrado”) e não gera trajetória; No caso em que os obstáculos obstruem todas as possibilidades de movimento ao redor do robô, este ficará impossibilitado de realizar uma ação.

Lembrando que essas situações são instantâneas, caso os obstáculos mudem de posição e a situação mude, o sistema fornecerá a resposta esperada, pois o mesmo encontra-se atualizando as informações do ambiente em tempo real.

Foi proposto um algoritmo de navegação autônoma, no qual estão integradas várias funções específicas:

- Leitura automática de imagens;
- Processamento digital de imagens uma a uma;
- Identificação dos elementos de cada imagem;
- Composição de uma matriz de posições, que armazena números que identificam os elementos presentes em cada imagem;



- Rastreamento de trajetória através de um algoritmo específico de inteligência artificial, o A-estrela ( $A^*$ );
- Transmissão via *Bluetooth* de instruções a serem executadas pelo robô a cada passo e em tempo real.

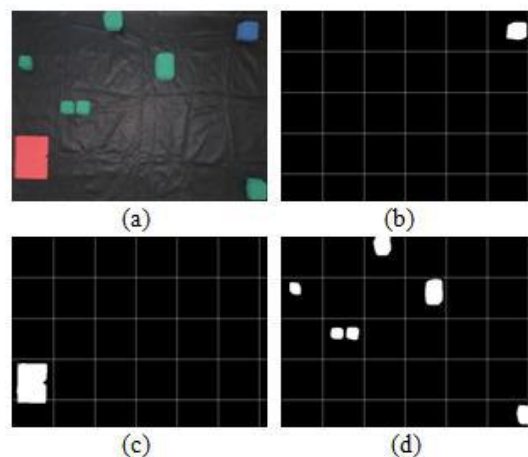
Antes da inicialização do algoritmo de navegação, imagens da cena onde o robô móvel precisa se deslocar são capturadas (com o robô já inserido no ambiente), através de uma câmera digital, em formato RGB de cores e dimensões de 640 x 480.

A primeira função que o algoritmo realiza é a leitura automática da primeira imagem capturada, que é então processada digitalmente.

As técnicas de processamento digital utilizadas foram selecionadas de maneira a permitir extrair e identificar informações das imagens, facilitando a interpretação automática por meio do sistema de navegação.

Como resultado do processamento, obtém-se a matriz de posições, que é analisada pelo  $A^*$  que gera o passo a ser executado. O sistema funciona como um *joystick*, tão logo o comando é ordenado, tão logo o robô o executa.

O sistema realiza a cada passo a aquisição da imagem com a vista superior instantânea do ambiente. Em seguida, a imagem de entrada (Figura 7a) é processada e depois de gerada a grade, o sistema gera três imagens de saída, cada uma identificando as posições relativas dos objetos: alvo, robô e obstáculos (ver Figura 7b, c e d respectivamente).



**Figura 7 - (a) Vista superior da imagem do ambiente. (b) Imagem de saída contendo a posição do alvo. (c) Imagem de saída contendo a posição relativa do robô. (d) Imagem de saída contendo as posições relativas dos obstáculos.**

Cada célula da grade da imagem representa uma célula da matriz  $M(5,7)$ , onde cada elemento da matriz assume o valor correspondente ao elemento que representa. Conforme representação:

- 1- Células livres;
- 2- Alvo;
- 3- Robô;
- 9- Obstáculos.

9	9	1	1	1	3	3
9	9	1	9	9	3	3
1	1	1	9	9	1	1
2	9	9	1	1	1	1
2	9	9	9	9	1	1

**Figura 8. Matriz de posições**

Essa matriz de posições garante as informações necessárias para que o algoritmo de busca  $A^*$  rastreie o próximo passo até o alvo, a fim de ser codificado e enviado ao robô via *Bluetooth*.

Cada célula tem seu estado registrado como livre ou com obstáculo. São analisadas as células adjacentes ao robô e em seguida o sistema determina a próxima célula a ser ocupada. Cabe registrar que todo o processamento digital, geração da matriz de posições e cálculo do passo da trajetória é realizado ininterruptamente e a resposta é executada pelo robô até que obtenha o próximo passo.

Neste experimento foram realizados testes onde os obstáculos móveis foram carrinhos de pressão, manipulados por auxiliares que os pressionavam para gerarem o movimento e os soltavam livremente no ambiente. O caso mais interessante seria se houvessem mais robôs móveis para servir de obstáculos em movimento, mas restrições orçamentárias à pesquisa só permitiram a aquisição de um robô móvel.

Os testes foram realizados com sucesso, com o robô alcançado o alvo e desviando-se dos obstáculos em movimento, conforme previsto no sistema proposto.

Alguns testes realizados deixaram claras as limitações do algoritmo de navegação. No caso em que a passagem até o alvo esteja completamente obstruída pelos obstáculos, o robô ficaria imóvel; em um ambiente com obstáculos móveis, mas sem um alvo, o sistema responde com uma mensagem de erro (“Alvo não encontrado”) e não gera trajetória; no caso em que os obstáculos obstruem todas as oito possibilidades de movimento ao redor do robô, este ficará impossibilitado de realizar uma ação. Lembrando que essas situações são

instantâneas, caso os obstáculos mudem de posição e a situação mude, o sistema fornecerá a resposta esperada.

Especificamente, foram feitas algumas alterações no algoritmo de navegação para que, entre a captura de imagens e a execução por parte do robô, houvesse melhor tempo de resposta.

Para leitura das imagens capturadas a partir de uma câmera, o algoritmo utiliza duas funções da biblioteca *Free Image*, a primeira, *FreeImage\_GetFIFFromFilename*, pega um nome ou uma extensão de arquivo e retorna o *plugin* que pode ler / escrever arquivos com essa extensão sob a forma de um identificador, e a segunda, *FreeImage\_Load*, carrega a imagem (FREEIMAGE, 2011). Para que a imagem possa ser analisada, é preciso antes inserir seu nome como parâmetro da função, esse procedimento era realizado manualmente imagem por imagem, o que atrasava o processamento e diminuía a velocidade de resposta.

Para garantir que não fosse necessária nenhuma intervenção humana, foi acrescentado um novo método no algoritmo para adquirir o nome automaticamente de maneira bem simples, inicialmente, deve-se definir o número máximo de leituras, o sistema obtém o nome da primeira imagem e todas as funções são aplicadas até que o sistema detecte satisfatoriedade na última função, que é então, quando a leitura da próxima imagem é realizada, e assim por diante, até a última imagem.

Cabe registrar que todo o processamento digital, geração da matriz de posições e cálculo de trajetória é realizado ininterruptamente e a resposta é executada pelo robô até que este atinja o alvo.

A conexão que possibilita a ação instantânea – e não um código a ser enviado ao Robô, como era feito anteriormente – é o aplicativo desenvolvido pelo projeto “Estudo da Transferência Automática de Dados para um Robô Móvel na Plataforma Lego *Mindstorms*” através de uma biblioteca auxiliar.

## 5 CONSIDERAÇÕES FINAIS

O uso da robótica móvel tem aplicações diversas na indústria, nos setores de segurança, de monitoramento ambiental, entre outros. Permitir a navegação autônoma de robôs móveis em ambientes com obstáculos em movimento se aproxima desta realidade de utilização destes sistemas.

No decorrer do projeto, foi realizado um estudo do estado da arte em navegação robótica. Conceitos de processamento digital de imagens, navegação autônoma e visão computacional foram aprofundados, bem como o aprendizado das particularidades do robô *Lego Mindstorms*, adotado para experimentos, e da linguagem adotada por ele: NXC.

Sem o interesse de ser uma solução definitiva, este trabalho apresentou uma proposta de um sistema para navegação autônoma para robôs móveis, considerando obstáculos móveis, cujos resultados obtidos indicam que o sistema é funcional dentro das restrições a que se propôs, e que em geral são função do próprio ambiente e das limitações de equipamentos de aquisição e processamento.

A dificuldade em desenvolver experimentos mais controlados, que necessitariam de diversos robôs móveis para representar os obstáculos, não permitiu experimentos mais consistentes o que, contudo, foi apenas uma dificuldade a mais no trabalho, não invalidando o experimento realizado como prova de funcionalidade.

Como contribuições para o incremento da ciência em busca de uma solução mais completa para o problema tratado neste trabalho pode ser destacado a proposta de passo incremental a cada possibilidade de mudança do ambiente, com a transmissão em tempo real das informações do ambiente para o robô e dos comandos de navegação.

Como trabalhos futuros, pretende-se em curto prazo utilizar métodos mais robustos de identificação dos elementos da cena, o alvo e os obstáculos móveis, para melhorar a localização destes na cena e adaptar o sistema para funcionar com um alvo móvel.

Numa visão mais ampla, o método pode ser estendido e adaptado para a navegação de diversos robôs móveis no mesmo ambiente, onde cada um seria ao mesmo tempo o agente e um obstáculo para o outro robô, permitindo inclusive uma integração em um sistema de navegação robótica cooperativa.

## REFERÊNCIAS

- ALVES, D. S. **Estimação de Curvas Polinomiais em seqüências de imagens para Navegação de Robôs Móveis**. 2006. 92f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Tecnologia, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília.
- A STAR. **A \* Pathfinding**. Disponível em: <[www.policyalmanac.org/games/aStarTutorial\\_port.htm](http://www.policyalmanac.org/games/aStarTutorial_port.htm)>. Acesso em: 17 ago. 2009.
- A ESTRELA **.Algoritmo A\***. Disponível em: <[http://www.dainf.ct.utfpr.edu.br/~fabro/IA\\_I/busca/Algoritmo\\_A\\_Estrela.pdf](http://www.dainf.ct.utfpr.edu.br/~fabro/IA_I/busca/Algoritmo_A_Estrela.pdf)>. Acesso em: 17 ago. 2009.
- ARAÚJO, S. A. de; LIBRANTZ, A. F. H. ; FLÓRIO FILHO, O. **Navegação autônoma de robôs: uma implementação utilizando o kit *Lego Mindstorms***. In CONGRESSO SUL CATARINENSE DE COMPUTAÇÃO, 2, 2006. Criciúma: Sulcomp, 2006. Disponível em: <<http://www.dcc.unesc.net/sulcomp/artigos/sessaoOral/22003.pdf>> Acesso em: 27 jul. 2009
- BENEDETTELLI, D. **Programming LEGO NXT Robots using NXC**. Version 2.2, June 7, 2007.
- BLUETOOTH SIG. **Bluetooth Basics**. Disponível em: <<http://www.bluetooth.com/Pages/Basics.aspx>>. Acesso em 12 de jan. 2011.
- BONATO, V.; MOLZ, R. F.; MORAES, F. G.; FURTADO, J. C.; FERÃO, M. F. **Proposta de um sistema para processamento de impressões digitais implementado em hardware**. REIC. Revista Eletrônica de Iniciação Científica, v. 3, p. 1-12, 2003.
- BONATO, V. **Projeto de um módulo de aquisição e pré – processamento de imagem colorida baseado em computação reconfigurável e aplicado a robôs móveis**. 2004. 82f. Dissertação (Mestrado em ciências da computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2004.
- BONIN-FONT, F.; ORTIZ, A.; OLIVER, G. **Visual Navigation for Mobile Robots: a Survey**. University of the Balearic Islands: Palma de Mallorca , 2007.

DESOUZA, G.N.; KAK A.C. **Vision for Mobile Robot Navigation: A Survey**. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 2, pp. 237–267, 2002.

FREEIMAGE. **FreeImage: a free, open source graphics library**. Documentation Library version 3.15.0. Disponível em <<http://freeimage.sourceforge.net/>> Acesso em: 28 jan. 2011.

GONZALEZ, R. C.; WOODS, R. E. **Processamento Digital de Imagens**. São Paulo: Blucher, 2000.

HANSEN, J. **Not eXactly C (NXC), Programmer's Guide**, Version 1.0.1 b33. October 10, 2007.

HEINEN, F. J. **Sistema de Controle Híbrido para Robôs Móveis Autônomos**. 2002. 130f. Dissertação (Mestrado em Computação Aplicada)- Centro de Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos, São Leopoldo.

HEINEN, M. R. **Controle Inteligente do Caminhar de Robôs Móveis Simulados**. 2007. 121f. Dissertação (Mestrado em Computação Aplicada) – Ciências exatas e tecnológicas, Programa Interdisciplinar de Computação Aplicada, Universidade do Vale do Rio dos Sinos, São Leopoldo.

HOFFMANN, L. T. et al. **Controle inteligente de um robô móvel utilizando imagens**. In: Encontro de Robótica inteligente - Programa de Pós - graduação em Computação Aplicada, INPE, Salvador, 2004.

JUNG, C. R, et al. **Computação Embarcada: Projeto e Implementação de Veículos Autônomos Inteligentes**. In: Congresso da Sociedade Brasileira de Computação, 25, 2005, São Leopoldo. 1406p. p. 1358-1397.

LIBRANTZ, A. F. H.; ARAÚJO, S. A. de; KOYOMA, C. **Sistema de Navegação Autônoma de Robôs: uma proposta de uso como instrumento pedagógico multidisciplinar no curso de Ciência da Computação**. Exacta, São Paulo, v. 4, n. especial, p. 123-125, 2006.

NETTO, C. N. da S.; WOLF D. F. **Utilização de processamento de imagens para navegação autônoma em ambientes externos**. Escola de Engenharia de São Carlos. Instituto de Ciências Matemáticas e da Computação, USP, SP, 2007.

MINDSTORMS. **Lego Mindstorms**. Disponível em:

<[http://mindstorms.lego.com/eng/Paris\\_Destination/Default.aspx](http://mindstorms.lego.com/eng/Paris_Destination/Default.aspx)> Acesso em: 18 jun. 2009.

MIRANDA NETO, A de. **Navegação de Robôs Autônomos Baseada em Monovisão**. 2007. 110f. Dissertação (Mestrado em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Departamento de Mecânica Computacional, Universidade Estadual de Campinas, Campinas.

MITTAL, R.K.; NAGRATH, I.J. **Robotics and Control**. Tata McGraw-Hill, New Delhi, 2003.

OLIVEIRA, M. A. de. **Aplicação de Métodos de Computação Flexível em Navegação Autônoma de Veículos**. 1995. 135f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica, Departamento de Engenharia da Computação e Automação Industrial, Universidade Estadual de Campinas, Campinas.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de Imagens Digitais: Princípios, Algoritmos e aplicações**. São Paulo: Thomson Learning, 2008.

PINAGÉ, F. A. **Navegação de Robôs Autônomos Utilizando Imagens**. 2009. Monografia (Graduação em Mecatrônica Industrial) – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, Manaus.

PINAGÉ, F. A. et al. **Um Sistema Para Navegação Autônoma De Robôs Terrestres**. In: CONNEPI, IV, Belém, 2009.

PINHEIRO, Hayanna S.; PINAGÉ, Felipe A.; QUEIROZ-NETO, José P. **Sistema de Navegação Autônoma de Veículos Não Tripulados em Ambiente com Obstáculos em Movimento**. In Workshop of Undergraduate Works – SIBGRAPI, XXIII, Gramado, 2010.

RIGHES, E. M. **Processamento de Imagens para Navegação de Robôs Autônomos**. 2004. 60f. Monografia (Bacharelado em Ciência da Computação) – Universidade do Vale do Rio dos Sinos, São Leopoldo.

RIGHES, E. M.; OSÓRIO, F. S. **Correlação de Imagens Coloridas visando auxiliar na Navegação e Controle de Robôs Autônomos**. In: Encontro Nacional de Inteligência Artificial, 5, São Leopoldo, 2005.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. 2 ed. Rio de Janeiro: Elsevier, 2004.

TOLEDO, S. **Analysis of the NXT Bluetooth-Communication Protocol**. 2006. Disponível em: < <http://www.tau.ac.il/~stoledo/lego/btperformance.html> > Acesso em: 06 jan. 2010.

TRUCCO, A. e MURINO, V. **Guest editors' introduction. Special Issue on Underwater Computer Vision and Pattern Recognition**, 79(1), 2000.