

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

**APLICANDO SEGMENTAÇÃO DE IMAGENS PARA
DETERMINAÇÃO DA ROTA DE UM ROBÔ**

AUTOR: THIAGO FIGUEREDO CARDOSO

TRABALHO DE GRADUAÇÃO DO CURSO DE ENGENHARIA DA COMPUTAÇÃO

Orientador: Prof. Dr. Aluizio Fausto Ribeiro Araújo

Recife, 23 de janeiro de 2008.

THIAGO FIGUEREDO CARDOSO

**APLICANDO SEGMENTAÇÃO DE IMAGENS PARA
DETERMINAÇÃO DA ROTA DE UM ROBÔ**

Trabalho de conclusão de curso apresentado
como parte das atividades para obtenção do
título de Bacharel em Engenharia da
Computação do Centro de Informática da
Universidade Federal de Pernambuco

Prof. orientador: Aluizio Fausto Ribeiro Araújo

Recife, 2008

ASSINATURAS

Este trabalho é o resultado dos esforços do estudante Thiago Figueredo Cardoso, sob a orientação do Prof. Aluizio Fausto Ribeiro Araújo, intitulado “Aplicando Segmentação de Imagens para a Determinação da Rota de um Robô” e conduzido no Centro de Informática da Universidade Federal de Pernambuco (CIn-UFPE). As pessoas listadas abaixo reconhecem o conteúdo deste documento e os resultados do Trabalho de Graduação.

Thiago Figueredo Cardoso

Aluizio Fausto Ribeiro Araújo

AGRADECIMENTOS

Agradeço à minha família, meu pai Valter, minha mãe Vanda e meu irmão Lucas, por estarem a postos sempre que precisei e por todo o esforço que me permitiu estar nesta cidade.

À minha namorada Paula pela pessoa maravilhosa que é e de diversas formas ajudou a mim e a este trabalho, pelos inúmeros momentos agradáveis e pela compreensão da minha ausência e cansaço em vários momentos.

Aos meus colegas de apartamento, Fernando, Júlio e Vinícius, minha segunda família, que agüentaram minha bagunça durante este trabalho e que me levaram para tomar umas cervejas quando era necessário me distrair.

Aos amigos do GER que participaram do ENECA. Sem a crença de vocês nessa “loucura” dificilmente este trabalho teria sido levado adiante.

Ao meu orientador Aluizio Araújo pela paciência e auxílio, e por ter aceitado meu trabalho.

Aos meus grandes amigos, Flávia, Alisson e Denise, que mesmo a distância não separou e que sempre me receberam de braços abertos.

Aos (quase-)engenheiros da minha turma com quem compartilhei alegrias e inquietações ao longo do curso.

Isso é oba oba!

Prof. José Dias

RESUMO

Este trabalho trata da aplicação de algoritmos de segmentação de imagens, em particular os Mapas Auto-Organizáveis e o Local Adaptive Receptive Field Self-Organizing Map, para a detecção de objetos. A detecção é baseada na cor dos objetos e é utilizada para guiar um robô móvel, autônomo e inteligente em um ambiente controlado.

Neste documento será apresentada uma breve revisão sobre técnicas de navegação de robôs, de redes neurais não-supervisionadas e de segmentação de imagens coloridas. Em seguida o leitor encontrará os detalhes da construção do robô e os resultados dos testes executados.

Palavras-chave: Robótica, Segmentação de Cores, Detecção de Objetos, Mapas Auto-Organizáveis (SOM), Visão de Máquina.

ABSTRACT

This work is about the application of image segmentation algorithms, in particular the Self-Organizing Maps and the Local Adaptive Receptive Field Self-Organizing Map, to detect objects. The detection is based on the color of the objects and is used to guide a mobile, autonomous and intelligent robot in a controlled environment.

In this document a short review on techniques for robot navigation, unsupervised neural networks and color image segmentation is presented. Following this, the reader will find the details of the construction of the robot and the results of the tests executed.

Keywords: Robotics, Color Segmentation, Object Detection, Self-Organizing Maps (SOM), Machine Vision

LISTA DE ILUSTRAÇÕES

Figura 2-1. Exemplo de ambiente interno complexo	17
Figura 2-2. Vencedores do DARPA Grand Challenge	20
Figura 3-1. Arquitetura do SOM.....	22
Figura 4-1. Representação cilíndrica do espaço HSI	28
Figura 5-1. O Robô Figa	37
Figura 5-2. O objeto a ser buscado	38
Figura 5-3. Visão física. Imagem gentilmente criada e cedida por Vinícius Fabrino	38
Figura 5-4. Visões da estrutura. a) visão traseira b) visão lateral e c) visão isométrica – Modelo criado e gentilmente cedido por Paula Souza Teixeira	39
Figura 5-5. Placa de controle dos motores	41
Figura 5-6. Visão lógica. Imagem gentilmente criada e cedida por Vinícius Fabrino	42
Figura 5-7. Relação entre pulso recebido e angulação de um servomotor.....	42
Figura 5-8. Imagem dividida em regiões	44
Figura 5-9. Interface gráfica do Figa Vision.....	46
Figura 6-1. Imagens utilizadas nos testes. a) House b) Pepper c) Lena d) Baboon	49
Figura 6-2. Resultados da aplicação do SOM (a)-(d), e do LARFSOM (e)-(h) na imagem House.....	50
Figura 6-3. Resultados da aplicação do SOM (a)-(d), e do LARFSOM (e)-(h) na imagem Pepper.....	51
Figura 6-4. Resultados da aplicação do SOM (a)-(d), e do LARFSOM (e)-(h) na imagem Lena	52

Figura 6-5. Resultados da aplicação do SOM (a)-(d), e do LARFSOM (e)-(h) na imagem Baboon	53
Figura 6-6. Imagem utilizada para os testes no ambiente do robô	55
Figura 6-7. a) Imagem 6-6 segmentada pelo SOM b) Imagem 6-6 segmentada pelo LARFSOM.....	56
Figura 6-8. Histogramas do tempo de execução e de classificação do LARFSOM	57

LISTA DE TABELAS

Tabela 6-1. Resultados da aplicação do SOM e do LARFSOM na imagem House	50
Tabela 6-2. Resultados da aplicação do SOM e do LARFSOM na imagem Pepper	51
Tabela 6-3. Resultados da aplicação do SOM e do LARFSOM na imagem Lena	53
Tabela 6-4. Resultados da aplicação do SOM e do LARFSOM na imagem Baboon	54
Tabela 6-5. Estatísticas da execução do LARFSOM na máquina com processador Intel Core 2 Duo	56
Tabela 6-6. Estatísticas da execução do SOM na máquina com processador Intel Core 2 Duo	56
Tabela 6-7. Estatísticas da execução do LARFSOM na máquina com processador Vortex 86	57
Tabela 6-8. Estatísticas da execução do SOM na máquina com processador Vortex 86	57

SUMÁRIO

1	Introdução	13
1.1	Contexto	13
1.2	Objetivos	14
1.3	Organização do Documento	14
2	Navegação de Robôs	16
2.1	Ambientes Internos	16
2.1.1	Navegação Baseada em Mapas	17
2.1.2	Navegação Baseada na Construção de Mapas	18
2.1.3	Navegação sem Mapas	18
2.2	Ambientes Externos	19
3	Redes Neurais Não-supervisionadas	21
3.1	Mapas Auto-Organizáveis	22
3.1.1	Arquitetura	22
3.1.2	Treinamento	22
3.1.3	Propriedades do mapa	23
4	Segmentação de Imagens Coloridas	25
4.1	Representação de Cores	26
4.1.1	RGB Normalizado	26
4.1.2	HSI	27
4.1.3	Cores CIE	28
4.1.4	YIQ e YUV	29
4.2	Segmentação Baseada no Espaço de Características	30
4.3	Segmentação Baseada em Regiões	31
4.4	Segmentação Baseada em Fronteiras	31
4.5	Segmentação Baseada em Características Físicas	32
4.6	Redes Neurais para Segmentação de Imagens	32
4.6.1	SOM	33
4.6.2	LARFSOM	33

5 O Robô Figa	37
5.1 <i>O Ambiente</i>	38
5.2 <i>Visão Física</i>	38
5.2.1 Estrutura e Motores	39
5.2.2 Câmera	40
5.2.3 Processamento Central	40
5.2.4 Controle dos Motores.....	40
5.2.5 Controle Remoto e Monitoramento e Figa Vision.....	41
5.3 <i>Visão Lógica</i>	41
5.3.1 Controle dos Motores.....	42
5.3.2 Detecção de Objetos	43
5.3.3 Navegação	44
5.3.4 Controle Remoto e Monitoramento e Figa Vision.....	45
6 Testes e Resultados	48
6.1 <i>Resultados em Imagens da USC-SIPI Image Database</i>	48
6.1.1 Imagem House.....	49
6.1.2 Imagem Pepper	50
6.1.3 Imagem Lena	52
6.1.4 Imagem Baboon.....	53
6.1.5 Observações.....	54
6.2 <i>Resultados no Ambiente do Robô</i>	55
7 Considerações finais.....	59

1 INTRODUÇÃO

Durante muito tempo a robótica foi aplicada na indústria, como forma de automatizar os processos, porém este cenário está mudando e cada vez mais se vê robôs no dia-a-dia. Apenas para ilustrar pode-se citar o V-R4000, limpador à vácuo da LG, a linha *WowWee Robotics* de brinquedos da *WowWee*¹, ou a *Gibson Robot Guitar*².

À medida que as interações se tornam mais complexas, cresce a necessidade de sistemas sensoriais poderosos. É natural pensar na visão como um desses sistemas, por ser ela um dos sentidos mais poderosos dos animais e igualmente complexo. A visão permite a interação inteligente com o ambiente sem a necessidade de contato direto [HORN 1986].

Baseado nisso muitas pesquisas surgiram na tentativa de acrescentar visão a um sistema robótico como uma forma de auxiliar em seu controle, tais como [MALIS 2002] e [DESOUZA 2002], e hoje diversos trabalhos são impulsionados por iniciativas como a *RoboCup*³.

1.1 Contexto

Durante o ano de 2007 o Grupo de Estudos em Robótica⁴ (GER) da UFPE desenvolveu pesquisas na área aplicadas ao contexto da competição Desafio Inteligente, sediada em Recife

¹ Empresa que desenvolve brinquedos com alta tecnologia. Para mais informações ver <http://www.wowwee.com>

² Guitarra capaz de se auto-afinar. Ver <http://www.gibson.com/robotguitar/>

³ Iniciativa para desenvolver a Inteligência Artificial que pretende construir um time de robôs até 2050 para vencer o atual campeão mundial de futebol. Mais em <http://www.robocup.org>

⁴ Grupo de estudos independente formado por estudantes de cursos de engenharia da UFPE. Mais em <http://groups.google.com/group/ger-ufpe>

no ano em questão, que consistia em criar um robô subaquático que fosse capaz de resgatar objetos de uma dada cor, evitando capturar outros objetos presentes no ambiente.

Ao longo do desenvolvimento, o autor deste trabalho iniciou pesquisas na área de visão para a resolução do problema citado, mas devido aos recursos humanos e tempo disponíveis, um sistema desse tipo não pode ser criado.

Os resultados dos estudos feitos e a possível continuidade do GER em competições como a *RoboCup* foram definitivos para a definição deste trabalho.

1.2 Objetivos

Sendo assim, o objetivo deste trabalho é desenvolver um sistema robótico capaz de detectar objetos simples baseados em suas cores. Para isso serão aplicadas novas técnicas não-supervisionadas de segmentação de imagens coloridas, em particular o *Local Adaptive Receptive Field Self-Organizing Maps* (LARFSOM) [ARAÚJO & COSTA, 2006], no processo de detecção de objetos e será feita uma análise de seu uso e comparação com o Mapa Auto-Organizável (SOM, do Inglês, *Self-Organizing Maps*).

1.3 Organização do Documento

O próximo capítulo fala sobre a taxonomia de sistemas de visão aplicados à navegação de robôs, de forma a ser possível identificar o tipo de robô que deverá ser desenvolvido e quais suas limitações.

O capítulo 3 aborda as características básicas de redes neurais não-supervisionadas e introduz o modelo básico utilizado no trabalho, o Mapa Auto-Organizável.

O capítulo 4 trata de segmentação de imagens coloridas, mostrando os vários tipos de técnicas existentes e avaliando o uso de redes neurais para este objetivo. Ao fim do capítulo dois modelos de interesse são vistos em detalhe.

O capítulo 5 detalha toda a construção do robô Figa, objetivo deste trabalho, mostrando os componentes usados e suas conexões, bem como todos os sistemas que nele executam.

O capítulo 6 apresenta os resultados dos testes executados para os algoritmos estudados em imagens conhecidas e no ambiente para o qual o robô foi construído.

Por fim considerações são feitas no capítulo 7 e propostas para trabalhos futuros são expostas.

2 NAVEGAÇÃO DE ROBÔS

A localização e movimentação de robôs em um dado ambiente são feitas com o auxílio de diversos dispositivos e técnicas, tais como sensores de obstáculos e distâncias, odometria e visão computacional. Embora a visão não seja obrigatória, muitos problemas são mais facilmente resolvidos a partir de seu uso para navegação e/ou controle de um robô [HORN, 1986]. Por tal razão, nestes trabalhos são abordadas apenas as técnicas que se baseiam em sistemas de visão.

A visão é uma das ferramentas mais poderosas que os animais possuem. Provê uma enorme quantidade de informações, mas também é extremamente complicada, de tal forma que várias tentativas de imitá-la não foram bem sucedidas [HORN, 1986]. Não existe, até agora, a tentativa de criar um sistema de visão geral, mas sim aplicado a certos problemas ou certos ambientes. Assim a discussão neste trabalho separa as abordagens utilizadas para visão na navegação em função do ambiente no qual o robô deverá atuar. A seguir estão as técnicas utilizadas em ambientes internos e externos.

2.1 Ambientes Internos

Seguindo um fluxo natural, esses foram os primeiros ambientes utilizados nas pesquisas. Grandes avanços foram alcançados nessa frente (ver [DESOUZA & KAK, 2002]) e hoje robôs conseguem se localizar em ambientes internos complexos, como a da figura a seguir.



Figura 2-1. Exemplo de ambiente interno complexo

As técnicas de navegação em ambientes internos são divididas em três: baseada em mapas, baseada na construção de mapas e navegação sem mapas (baseada na detecção de marcadores).

2.1.1 Navegação Baseada em Mapas

Nesta categoria estão os sistemas que dependem de um modelo do ambiente inserido pelo usuário. Estes modelos podem ter diversos graus de complexidade, variando de um modelo CAD (*Computer-Aided Design*) completo a um simples grafo de conexões entre os objetos presentes no ambiente.

O papel de tais mapas é prover ao robô uma forma de calcular sua posição e conseguir determinar um caminho até seu objetivo. Quatro passos são definidos nesses sistemas [BORENSTEIN et al., 1996]:

- ⌘ Adquirir informações dos sensores: principalmente da câmera, para o caso deste trabalho.
- ⌘ Detectar características.
- ⌘ Efetuar busca pelos modelos a partir das características extraídas.
- ⌘ Calcular posição.

A forma como as técnicas executam o terceiro passo e geram os dados para a quarta etapa demanda especificação de sub etapas: cálculo de posição absoluta, cálculo incremental de posição e busca de marcadores.

As técnicas de cálculo absoluto de posição consideram que a posição inicial do robô não é conhecida. Assim é necessário fazer a localização dos marcadores e, por técnicas de triangulação, encontrar a localização do robô. Devido ao erro da detecção e do próprio cálculo é necessário efetuar novas observações quando o erro cresce acima de um limiar.

Quando a posição inicial do robô é conhecida, torna-se possível calcular a posição de forma incremental, acumulando as incertezas das posições em cada instante, até que um certo limite seja excedido, quando o robô deverá recalculer sua posição.

A terceira técnica é utilizada quando tanto a posição inicial do robô e a identidade dos marcadores é conhecida e pode ser rastreada [DESOUZA & KAK, 2002].

2.1.2 Navegação Baseada na Construção de Mapas

Nem sempre é possível construir uma representação do ambiente, principalmente quando se devem informar relações métricas entre os objetos, sobretudo em ambientes dinâmicos. Por conta disso, pesquisas foram feitas acerca da construção (semi-)automática de representações do ambiente. Para mais informações sobre técnicas de construção de mapas sugere-se ao leitor a referência [THRUN, 2003].

Após a construção do mapa, a navegação é feita por técnicas similares às descritas na seção anterior.

2.1.3 Navegação sem Mapas

A navegação sem mapas é feita quando não há nenhuma descrição prévia sobre o ambiente. Também não possui tal descrição a técnica descrita na seção anterior, porém existe o passo de construção de um mapa antes de qualquer locomoção.

Sem a existência de uma descrição do ambiente, três classes de técnicas são definidas em [DESOUZA & KAK, 2002]: navegação utilizando o fluxo óptico, navegação baseada em aparência e navegação utilizando o reconhecimento de objetos.

A primeira utiliza o conceito de determinação do movimento dos objetos presentes na cena para estimar o caminho a ser feito. Fluxo óptico é um campo de vetores que indicam a direção e magnitude da variação das mudanças de intensidade na imagem [TEMIZER, 2001]. Na pesquisa feita em [DESOUZA & KAK, 2002], apresenta-se o exemplo do *robee*, um robô que tenta imitar o reflexo centralizador de uma abelha.

A navegação baseada em aparência visa memorizar o ambiente, guardando localizações para determinação do caminho a ser seguido.

Por último, a navegação utilizando o reconhecimento de objetos, cujo nome é auto-explicativo, utiliza uma abordagem diferente da navegação baseada em aparência. No lugar de memorizar a localização de objetos na cena, o robô recebe objetivos, tais como “vá até a porta”, e então faz a busca pelo objeto porta e assim determina seu caminho.

2.2 Ambientes Externos

Em ambientes externos aparecem dificuldades não existentes em ambientes internos, normalmente mais controlados. Por exemplo, raramente se conhece um mapa completo [DESOUZA & KAK, 2002], além dos problemas causados pela variação de iluminação, que dependem da hora do dia, das condições climáticas, entre outros (uma discussão sobre trabalhos que endereçam esse problema pode ser encontrada em [DESOUZA & KAK, 2002])

Os ambientes externos são de dois tipos: estruturados e não-estruturados. Um dos primeiros trabalhos para ambientes estruturados endereçava o problema de um robô para dirigir um carro em uma estrada. Hoje existem diversas soluções para tal problema, como nos mostram os competidores do *DARPA Grand Challenge*⁵.

⁵ Visite <http://www.darpa.mil/GRANDCHALLENGE> Para mais informações sobre o DARPA Grande Challenge



Figura 2-2. Vencedores do DARPA Grand Challenge

Os ambientes não-estruturados são aqueles que não possuem propriedades regulares que possam ser extraídas. Viagens através de países ou interplanetárias são exemplos de situações em que o ambiente é não-estruturado. No caso de exploração interplanetária, como é o caso de Marte, o robô deve apenas explorar o ambiente sem nenhum objetivo específico. Quando o robô possui um objetivo, tal como ir de um lugar a outro, técnicas de localização são necessárias, tais como observação por câmera externa, triangulação por marcadores distantes e *Global Positioning System* (GPS).

3 REDES NEURAIIS NÃO-SUPERVISIONADAS

Aprendizagem não-supervisionada procura extrair informação de dados não-classificados [DUDA et al., 2000]. Em contraste com a aprendizagem supervisionada ou por reforço, não há um professor ou um crítico para auxiliar no treinamento.

Seu estudo é de grande importância, pois aparenta ser mais comum no cérebro que sistemas de aprendizagem supervisionada, como por exemplo na visão [DAYAN, 1999]. Além disso, de acordo com [DUDA et al., 2000], há ao menos cinco razões básicas para o uso desse tipo de aprendizagem:

- ⌘ A coleta a classificação de um grande conjunto de dados para utilizar no treinamento pode ser extremamente custoso;
- ⌘ Com aprendizagem não-supervisionada é possível criar as classes a serem utilizadas por um método supervisionado;
- ⌘ Se os dados mudarem com o tempo, a performance de um classificador pode ser melhorada utilizando uma técnica não-supervisionada que seja capaz de capturar as mudanças;
- ⌘ Um método não-supervisionado pode ser utilizado para a extração de características dos dados analisados;
- ⌘ Nos estágios iniciais da investigação pode ser importante visualizar as características básicas do dados.

Em redes neurais, dois algoritmos não-supervisionados se destacam: *Adaptive Resonance Theory* e Mapas Auto-Organizáveis. Neste trabalho será dado foco ao segundo algoritmo, descrito nas seguintes seções.

3.1 Mapas Auto-Organizáveis

As idéias de auto-organização foram propostas por von der Malsburg e Willshaw na década de 70 para explicar o funcionamento de certas partes do córtex visual. Mais tarde, na década de 80, Kohonen introduziu seu modelo chamado *Self-Organizing Maps* (SOM) que captura as características essenciais do modelo de von der Malsburg e Willshaw, mas mantendo o problema computacionalmente tratável [HAYKIN, 1998].

O SOM tem como objetivo mapear dados multi-dimensionais em um mapa de, predominantemente, duas dimensões, através de um processo de aprendizagem competitivo e não-supervisionado.

3.1.1 Arquitetura

Os neurônios estão organizados em uma grade, onde todas as entradas se conectam a todos os nós e *feedback* é restrito às conexões laterais imediatas [BEALE & JACKSON, 1990]. Todos os neurônios da grade são nós de saída, como mostrado na figura seguinte.

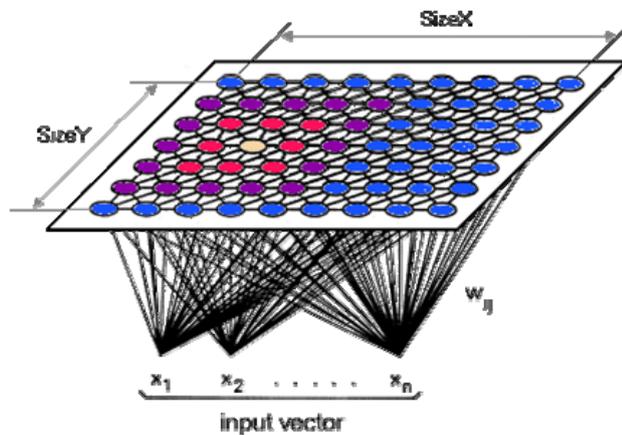


Figura 3-1. Arquitetura do SOM

3.1.2 Treinamento

O algoritmo de aprendizagem é competitivo (*winner-take-all*) e possui os passos a seguir.

Passo 1: Inicializar pesos e parâmetros. Defina o tamanho da grade e a taxa de aprendizagem η .

Passo 2: Para cada padrão da entrada repita os seguintes passos.

Passo 3: Calcule a distância do padrão de entrada a todos os nós e aquele cuja a distância seja a menor é a *Best Matching Unit* (BMU).

Passo 4: Atualize os pesos da BMU e de seus vizinhos

Passo 5: Ajuste o tamanho da vizinhança e a taxa de aprendizagem η .

A inicialização dos pesos influencia na forma como o mapa vai se formar. Para diminuir esta dependência, algumas técnicas são utilizadas: inicializar os pesos com o mesmo valor; iniciar os pesos com valores aleatórios; ou adicionar ruído aos dados de entrada [BEALE & JACKSON, 1990].

O passo 5 possui comportamento distinto de acordo com a fase em que a rede se encontra, a saber: fase de ordenação e fase de convergência. Na primeira, a taxa de aprendizagem deve decrescer gradualmente, mas ficar sempre acima de 0,01, enquanto o tamanho da vizinhança inicia com o mesmo valor do tamanho do mapa e decai vagarosamente. Kohonen sugere que esta fase dure pelo menos 1000 iterações.

Na fase seguinte, o tamanho da vizinhança e a taxa de aprendizagem permanecem pequenos, permitindo um ajuste fino da rede, até que o treinamento termine.

3.1.3 Propriedades do mapa

O SOM cria uma representação do espaço de dados de entrada com as seguintes propriedades [HAYKIN, 1998]:

- ⌘ Boa aproximação dos espaço de dados de entrada;
- ⌘ Possui ordenamento topológico;
- ⌘ Representação de densidade dos dados de entrada;
- ⌘ Efetua a seleção de características que melhor representam os dados de entrada.

Tais propriedades fazem do SOM uma poderosa ferramenta para agrupamento, em especial pela representação de densidade [XU & WUNSCH, 2005]. No próximo capítulo, este algoritmo será revisitado, aplicado ao agrupamento de cores em imagens.

4 SEGMENTAÇÃO DE IMAGENS COLORIDAS

A segmentação de imagens é um dos primeiros passos em um sistema de visão ou de análise de imagens, a ponte entre as operações de baixo nível e as análises de alto nível, tais como reconhecimento de objetos e entendimento da cena. Seu papel é crítico e determina a qualidade das operações posteriores [CHENG et al., 2001].

Define-se segmentação como o processo que tem como objetivo separar a imagem em regiões que satisfaçam um critério de uniformidade [SKARBEEK & KOSCHAN, 1994]. Mais formalmente [LUCCHESI & MITRA, 2001]: Seja I uma imagem e H um dado predicado de homogeneidade; a segmentação de I é a partição P de I em um conjunto de N regiões $R_n, n = 1..N$, tal que:

1. $\bigcup_{n=1}^N R_n = I$, com $R_n \cap R_m \neq \emptyset, n \neq m$
2. $H(R_n)$ é verdadeiro para qualquer n
3. $H(R_n \cup R_m)$ é falso para qualquer R_n adjacente a R_m

Duas propriedades são desejáveis às técnicas de segmentação [DONG & XIE, 2005]: homogeneidade, como explicitado na definição, e compactação espacial, propriedade que se refere a quão compactas são as regiões formadas.

Seguindo estes conceitos, diversas técnicas foram propostas e podem ser dispostas em três classes de acordo com a utilização dos elementos da imagem para o propósito da segmentação, a saber: baseada no espaço de características, baseada no domínio da imagem e baseada em características físicas.

A seguir neste capítulo encontram-se algumas representações de cores, característica básica de uma imagem colorida, e em seguida entramos em mais detalhes sobre as abordagens citadas no parágrafo anterior.

4.1 Representação de Cores

A visão humana percebe as cores como a representação de três estímulos: R (*Red*), G (*Green*) e B (*Blue*). Essas componentes são extraídas da cena a partir da aplicação de três filtros, descritos pelas seguintes equações:

$$\begin{aligned}R &= \int_{\lambda} E(\lambda) S_R(\lambda) d\lambda \\G &= \int_{\lambda} E(\lambda) S_G(\lambda) d\lambda \\B &= \int_{\lambda} E(\lambda) S_B(\lambda) d\lambda\end{aligned}$$

onde $E(\lambda)$ representa a radiação da luz, $S_X(\lambda)$, $X = \{R, G, B\}$ é o filtro passa-banda na frequência de X, e λ é o comprimento de onda.

A representação (espaço de cores) RGB é adequada para exibição de cores, mas não para análise, devido à alta correlação entre suas componentes [CHENG et al., 2001].

Diversos outros espaços, que podem ser mais adequados à segmentação, são derivados a partir do RGB através de transformações lineares e não-lineares, mas nenhum é melhor em todas as situações. A seguir estão listados alguns espaços de cores.

4.1.1 RGB Normalizado

Para diminuir a dependência da iluminação na segmentação, normaliza-se o espaço RGB. O espaço normalizado é formulado como:

$$\begin{aligned}r &= \frac{R}{(R+G+B)} \\g &= \frac{G}{(R+G+B)} \\b &= \frac{B}{(R+G+B)}\end{aligned}$$

Embora sejam relativamente robusto a variações de iluminação, as cores normalizadas apresentam bastante ruído quando estão em baixas intensidades.

4.1.2 HSI

HSI (*hue-saturation-intensity*) é outro espaço de cores baseado no modelo de percepção humana. *Hue* (coloração) representa a cor básica, *saturation* (saturação) indica o grau de pureza, ou seja, a quantidade de branco que está misturado com a cor, e *intensity* (intensidade) indica a quantidade de luz presente.

Este sistema é derivado do RGB a partir de [SKARBEEK & KOSCHAN, 1994]:

$$I = \frac{(R + G + B)}{3}$$
$$S = 1 - \frac{\min(R, G, B)}{I}$$
$$H = \arctan\left(\frac{\sqrt{3}(G - B)}{(R - G) + (R - B)}\right)$$

Este espaço de cores possui boa capacidade de representar as cores perceptíveis pelos humanos. Por permitir uma separação entre a noção de cor e iluminação, é adequado para o uso em segmentação, inclusive diminuindo a dimensão do problema para uma variável, por exemplo, a segmentação de tons de cinza pode ser feita com a análise da componente I, enquanto a segmentação de cores com a componente H. A desvantagem deste modelo está na instabilidade da componente H para cores próximas do eixo do cilindro (ver Figura 4-1), onde uma pequena variação de R, G ou B gera uma grande mudança em H. Se as cores estiverem muito próximas de branco ou preto H e S não são muito úteis na distinção de cores [CHENG et al., 2001].

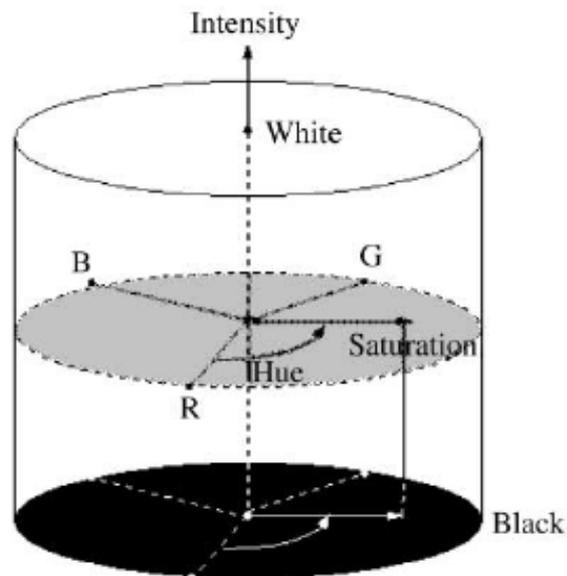


Figura 4-1. Representação cilíndrica do espaço HSI

4.1.3 Cores CIE

O sistema RGB exclui certas cores visíveis, além de depender da resolução dos sensores. Por estes motivos, o *Commission Internationale de l'Éclairage* (CIE) fixou as coordenadas XYZ, que podem ser obtidas a partir de uma transformação linear do espaço RGB, porém a matriz deve ser determinada empiricamente [SKARBEBK & KOSCHAN, 1994]. O sistema NTSC (*National Television System Commission*) utiliza a seguinte matriz, derivada a partir de experimentos:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0,607 & 0,174 & 0,200 \\ 0,299 & 0,587 & 0,114 \\ 0,000 & 0,066 & 1,116 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

A partir das coordenadas XYZ é possível construir os espaços CIE. Dois espaços muito conhecidos são o CIE Lab, onde L corresponde à iluminação, a corresponde à relação entre vermelho/verde e b à relação entre verde/azul, e o CIE Luv, cujas coordenadas possuem o mesmo significado que as do CIE Lab, porém são calculadas de forma diferente.

$$L = 116 \left(\sqrt[3]{\frac{Y}{Y_0}} \right) - 16$$

$$a = 500 \left[\sqrt[3]{\frac{X}{X_0}} - \sqrt[3]{\frac{Y}{Y_0}} \right]$$

$$b = 200 \left[\sqrt[3]{\frac{Y}{Y_0}} - \sqrt[3]{\frac{Z}{Z_0}} \right]$$

$$L = 116 \left(\sqrt[3]{\frac{Y}{Y_0}} \right) - 16$$

$$u = 13W \left(\frac{4X}{X + 15Y + 3Z} \right) - 0,199$$

$$v = 13W \left(\frac{6Y}{X + 15Y + 3Z} \right) - 0,308$$

onde (X_0, Y_0, Z_0) são os valores de XYZ para o branco.

Esses espaços são ditos uniformes, o que permite a utilização da distância euclidiana como boa métrica para definir diferença entre as cores. O uso do CIE Lab provê bons resultados na segmentação de imagens coloridas [SKARBEEK & KOSCHAN, 1994].

4.1.4 YIQ e YUV

Os espaços descritos anteriormente possuíam como característica o uso de uma transformação não-linear para obtenção de seus parâmetros. Para sistemas que possuem pouco poder computacional, transformações lineares são desejáveis.

Estes dois espaços, utilizados para codificação de cor nos sistemas de televisão americano e europeu, respectivamente, são obtidos pelas seguintes transformações:

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,274 & -0,322 \\ 0,211 & -0,253 & -0,312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

onde $0 \leq R \leq 1, 0 \leq G \leq 1, 0 \leq B \leq 1, 0$

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,437 \\ 0,615 & -0,515 & -0,100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

onde $0 \leq R \leq 1,0$; $0 \leq G \leq 1,0$; $0 \leq B \leq 1,0$

As componentes I e Q (U e V) descrevem conjuntamente a saturação e coloração, enquanto a componente Y descreve a iluminação.

4.2 Segmentação Baseada no Espaço de Características

Também conhecida por “baseada em *pixel*”, assume que cada região forma um agrupamento no espaço de características, que pode ser um dos espaços de cores descritos anteriormente ou um espaço induzido por outras características de cores, mas não considera nenhuma relação espacial entre os *pixels*.

Esse tipo de segmentação é mais simples de implementar e a classificação é direta, porém não garante que as regiões sejam contíguas por não considerar informações espaciais.

As técnicas mais comuns de segmentação baseada em *pixel* são limiarização de histograma e agrupamento. A primeira, bastante utilizada em imagens monocromáticas, consiste em encontrar picos e vales no histograma gerado a partir das informações de cor da imagem. Em histogramas de uma dimensão, encontrar picos e vales é uma tarefa facilmente executada, mas em imagens coloridas os histogramas são normalmente irregulares, devido a ruídos principalmente, o que dá origem a falsos picos, e por sua vez ambigüidade na segmentação [LUCCHESI & MITRA, 2001].

Agrupamento tem sido usado como uma importante técnica para reconhecimento de padrões [CHENG et al., 2001]. Consiste em classificar objetos em partições de forma não supervisionada, sem nenhum conhecimento *a priori*. Um dos algoritmos mais conhecidos é o *k-means* (ou *c-means*). Variações de algoritmos de agrupamento utilizando lógica difusa também são muito empregados, como o *fuzzy c-means*, de tal forma que algumas análises dedicam seções separadas para esses algoritmos.

4.3 Segmentação Baseada em Regiões

Esta classe de técnicas considera, além das informações fornecidas pelo espaço de características, as relações espaciais entre os *pixels*. Duas abordagens são utilizadas: *split-and-merge* e crescimento de regiões.

Split-and-merge consiste em iniciar com uma partição não homogênea, usualmente a imagem inteira, e dividi-la até que se obtenham apenas partições homogêneas. Neste ponto, uma heurística de união (*merge*) é utilizada para juntar as partições correlatas. A maior desvantagem desse método é que a imagem resultante tenta imitar a estrutura utilizada para representar a imagem [CHENG et al., 2001]

Crescimento de regiões faz o caminho inverso de *split-and-merge*. Inicia de uma semente pré-selecionada e aplica técnicas para aglomerar pontos vizinhos que satisfaçam um critério de homogeneidade. A vantagem desse tipo de técnica é que as regiões geradas são conexas e compactas, mas é clara a dependência na escolha da semente e da ordem de apresentação dos *pixels* [LUCCHESI & MITRA, 2001].

4.4 Segmentação Baseada em Fronteiras

Em contraste com as outras descritas anteriormente, as técnicas de segmentação de fronteiras (ou detecção de bordas) se utilizam da descontinuidade entre os *pixels* vizinhos para efetuar a segmentação.

Embora a detecção de bordas em imagens monocromáticas tenha sido largamente investigada e técnicas robustas tenham sido propostas, este tipo de técnica de segmentação apresenta limitações. Se uma imagem colorida possui regiões adjacentes com o mesmo brilho mas diferentes níveis de coloração e saturação, tal fronteira não seria identificada em sua versão monocromática. Por tal razão várias técnicas foram propostas para imagens coloridas, entre elas extensões de técnicas utilizadas em imagens monocromáticas, como o cálculo de gradiente em três dimensões.

4.5 Segmentação Baseada em Características Físicas

Todas as técnicas anteriores estão sujeitas a erros causados por reflexos ou sombras dos objetos presentes na cena. Por exemplo, uma esfera pode ser segmentada em várias regiões devido às diferenças de iluminação. Para contornar esse problema, as técnicas baseadas em características físicas adicionam propriedades físicas, tais como modelos de reflexão de certos materiais.

Esta abordagem é eficiente em situações em que os modelos físicos dos materiais constituintes dos objetos da cena sejam bem conhecidos. Porém tal premissa é tão rígida que limita bastante a aplicação dessas técnicas [CHENG et al., 2001].

4.6 Redes Neurais para Segmentação de Imagens

As redes neurais possuem algumas desvantagens em relação a outras técnicas quando utilizadas para a segmentação de imagens, como por exemplo o tempo de treinamento muito longo e a sensibilidade aos parâmetros de inicialização [CHENG et al., 2001]. No entanto o alto grau de paralelismo, a capacidade de mapeamento não-linear, a capacidade de auto-adaptação, a tolerância a erros e a simplicidade de implementação dos algoritmos [DONG & XIE, 2005] fazem das redes neurais uma alternativa bastante interessante.

Assim como para agrupamento [XU & WUNSCH, 2005], o SOM e suas variantes se destacam na resolução do problema da segmentação de cores. Sua capacidade de descrição do conjunto de dados de forma não-supervisionada aliado ao pequeno custo computacional foram decisivos na escolha desse método. Por possuir um treinamento rápido, que pode ser executado *online*, não pertence à classe das redes com a primeira desvantagem listada acima, porém é sensível aos parâmetros de inicialização.

Outra técnica não-supervisionada para segmentação é o *Local Adaptive Receptive Field Self-Organizing Map* (LARFSOM). O LARFSOM combina características do SOM com *Grow When Required* (GWR), permitindo um maior grau de adaptatividade que o SOM e convergência mais rápida [ARAÚJO & COSTA, 2006]. Por tais motivos, decidiu-se analisar esse algoritmo e confrontá-lo com o SOM.

As próximas seções detalham como esses dois algoritmos são utilizados para a segmentação de imagens coloridas.

4.6.1 SOM

O algoritmo básico do SOM encontra-se no capítulo 3. Nesta seção pretende-se falar dos detalhes da aplicação deste método à segmentação de imagens coloridas.

Como esta rede é utilizada como um algoritmo de segmentação baseado em *pixel*, os valores RGB de cada *pixel* constituem o vetor de entrada.

Dois são os parâmetros que devem ser passados à rede: tamanho da grade e taxa de aprendizagem inicial. O tamanho da grade define a quantidade de cores que o mapa pode representar e é também o tamanho inicial da vizinhança. Ambos os parâmetros decaem linearmente⁶ com o número de iterações.

A segmentação é feita do seguinte modo:

Passo 1. Inicialize a rede com o tamanho da grade e a taxa de aprendizado inicial.

Passo 2: Treine a rede segundo o algoritmo descrito no capítulo 2, passando como entrada um *pixel* da imagem escolhido aleatoriamente, até que todos sejam apresentados.

Passo 3: Substitua a cor de cada *pixel* pelos pesos da *Best Matching Unit* (BMU) referente a tal pixel.

4.6.2 LARFSOM

Assim como o SOM, o LARFSOM é um algoritmo de segmentação baseado no espaço de características. A descrição completa do modelo presente em [ARAÚJO & COSTA, 2006] contém um segundo estágio para a segmentação de fronteiras que não será abordado neste texto.

⁶ Embora outras formas de variação desses parâmetros são sugeridas em [HAYKIN, 1998], o decaimento linear deu resultados satisfatórios e evitou o uso de operações custosas como a exponenciação e radiciação.

O LARFSOM contém as características interessantes do SOM e de GWR. Do primeiro a aprendizagem por competição, a capacidade de agrupamento e a distribuição topológica, e do segundo a capacidade de aumentar o mapa apenas quando necessário.

Assim como no SOM, o tamanho do mapa define a quantidade de cores que o algoritmo pode representar. As relações entre o tamanho do mapa e a quantidade de cores relevantes determinam a qualidade da classificação. Se o primeiro for maior que o segundo, classificações falsas podem ocorrer, e se for o contrário regiões distintas podem ser unidas em uma só. Por isso, a capacidade de dinamicamente modificar o tamanho do mapa representa uma propriedade muito útil desta técnica.

O algoritmo do LARFSOM é dividido em 10 passos: a inicialização dos parâmetros, seleção do padrão de entrada, procura pela *Best Matching Unit* (BMU), inserção da conexão entre as BMUs, cálculo do campo receptivo local da BMU, possível inserção de um novo nó ou atualização dos pesos, checagem do critério de parada, construção da paleta de cores e construção da imagem segmentada.

Passo 1: Inicialize os parâmetros: taxa de aprendizagem final (ρ_f), modulador da taxa de aprendizagem (ε), limiar de atividade (a_T), quantidade de vitórias do nó i (d_i), máxima quantidade de vitórias de um nó (d_m), erro mínimo (e_{\min}) e o número inicial de nós conectados ($N = 2$). Os nós iniciais são aleatoriamente escolhidos da imagem.

Passo 2: Selecione aleatoriamente um padrão $\xi = [r \quad g \quad b]^T$ e apresente-o à rede.

Passo 3: Calcule a distância Euclidiana para todos os nós da rede de forma a encontrar a BMU s_1 , que corresponde àquele nó cuja distância é a menor.

$$d(\xi, \mathbf{w}_i) = \|\xi - \mathbf{w}_i\|^2$$

$$\|\xi - \mathbf{w}_i\|^2 = (r - w_{ir})^2 + (g - w_{ig})^2 + (b - w_{ib})^2$$

Passo 4: Insira uma conexão entre as duas BMUs, s_1 e s_2 .

Passo 5: Calcule o campo receptivo local da BMU utilizando a fórmula abaixo. O campo receptivo local indica a similaridade entre as duas BMUs.

$$r_{s_1} = \sqrt{(w_{s_1r} - w_{s_2r})^2 + (w_{s_1g} - w_{s_2g})^2 + (w_{s_1b} - w_{s_2b})^2}$$

Passo 6: Calcule a atividade da BMU s_1 com a seguinte fórmula:

$$a_{s_1} = \frac{\exp(-\|\xi - \mathbf{w}_{s_1}\|)}{r_{s_1}}$$

Passo 7: Insira um novo nó se a atividade for menor que o limiar de ativação, senão atualize os pesos. A inserção de um novo nó é feita em cinco passos:

- ⌘ Adicione um novo nó com o peso igual ao valor RGB normalizado do padrão de entrada atual ξ .
- ⌘ Incremente o contador de nós, $N = N + 1$;
- ⌘ Remova a conexão entre as duas BMUs s_1 e s_2 . Logo adiante as conexões serão refeitas, ou entre as duas BMUs, ou entre elas e o novo nó;
- ⌘ Calcule as distâncias dois a dois entre o novo nó, e as duas BMUs.
- ⌘ Insira uma conexão entre os nós que correspondem às menores distâncias

A atualização de pesos é feita pela seguinte fórmula:

$$\Delta \mathbf{w}_{s_1} = \rho \times (\xi - \mathbf{w}_{s_1})$$

$$\text{onde } \rho = \begin{cases} \varepsilon \times \rho_f^{(d_i/d_m)}, & d_i \leq d_m \\ \varepsilon \times \rho_f, & d_i > d_m \end{cases}$$

Passo 8: Verifique o critério de parada, definido pela fórmula abaixo e volte ao Passo 2 a menos que ele seja satisfeito.

$$e = \frac{1}{N} \sum_{i=0}^{N-1} \|\mathbf{w}_{i\text{current}} - \mathbf{w}_{i\text{former}}\|^2 \leq e_{\min} = 10^{-4}$$

Este é o último passo para o treinamento da rede. Para seu uso, considere os próximos passos:

Passo 9: Aplique a cada nó uma cor da paleta que mais se aproxime de seus pesos.

Passo 10: Atribua a cada *pixel* da imagem a cor mais próxima na paleta.

Em [ARAÚJO & COSTA, 2006] há uma discussão sobre a interpolação de valores RGB para gerar mais cores, porém não é de interesse deste trabalho executar esse passo, pois o objetivo é reduzir bastante o número de cores.

A descrição do algoritmo é extensa e aparentemente deve demorar muito mais que o SOM para terminar seu treinamento, no entanto os resultados em [ARAÚJO & COSTA, 2006] mostram que a convergência desse algoritmo é muito rápida. Mais a frente, no capítulo de resultados uma comparação é feita entre o LARFSOM e SOM implementados para este trabalho.

5 O ROBÔ FIGA

A partir do estudo relatado nas seções anteriores, foi desenvolvido o robô Figa, plataforma sobre a qual um sistema de detecção de objetos baseado em suas cores foi implantado.

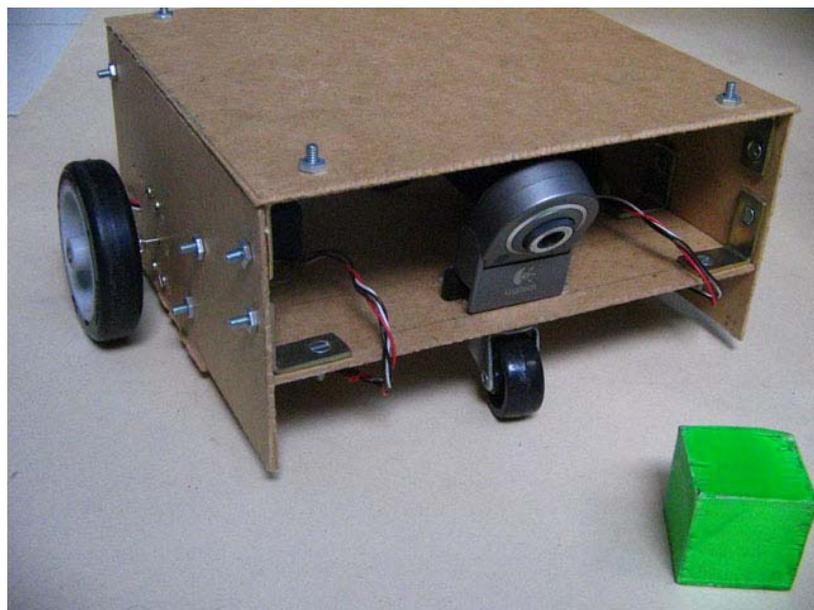


Figura 5-1. O Robô Figa

Figa é um robô móvel, terrestre, criado para ambientes internos, com navegação sem mapas baseada na detecção de objetos. Para que seja possível controlá-lo devidamente a partir da detecção de objetos, impõe-se a restrição de que este processo não deve demorar mais que dois segundos, ou seja, as imagens consigam ser processadas em uma taxa de 0,5 *frames* por segundo (*fps*). Nas seções seguintes detalha-se o ambiente e a implementação do robô.

5.1 O Ambiente

Em função da complexidade da criação de um robô para ambientes externos, principalmente relacionado aos problemas de iluminação, decidiu-se por testar o sistema robótico em um ambiente controlado.

O ambiente é formado por um piso branco, feito de um material não reflexivo, de forma a evitar confusão acerca de reflexos do objeto. Neste local estão distribuídos de forma aleatória dois ou mais cubos verdes de aresta 3,5cm. Os cubos verdes representam os objetos a serem identificados pelo robô.



Figura 5-2. O objeto a ser buscado

5.2 Visão Física

A visão física do sistema apresenta os componentes utilizados em sua construção e suas interdependências.

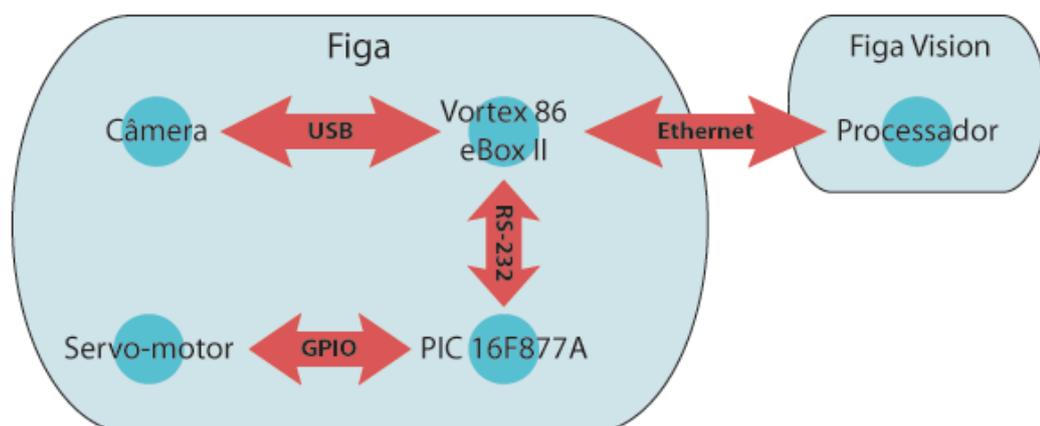


Figura 5-3. Visão física. Imagem gentilmente criada e cedida por Vinícius Fabrino

O bloco central da Figura 5-3, o Vortex 86, é responsável pelo processamento do robô. Ele se conecta através da porta serial ao microcontrolador PIC 16F877A, responsável pelo controle dos motores. A comunicação entre o microcontrolador e os servos-motores é feita por *General Purpose Input/Output* (GPIO). O Vortex 86 se conecta à câmera através da porta USB e através da rede permite a conexão com o sistema Figa Vision. Nas seções seguintes cada bloco desses é visto em detalhe.

5.2.1 Estrutura e Motores

Todos os componentes do Figa estão montados em uma estrutura construída com peças de MDF (*Medium Density Fiberboard*), material derivado da madeira. A estrutura possui formato de paralelepípedo.

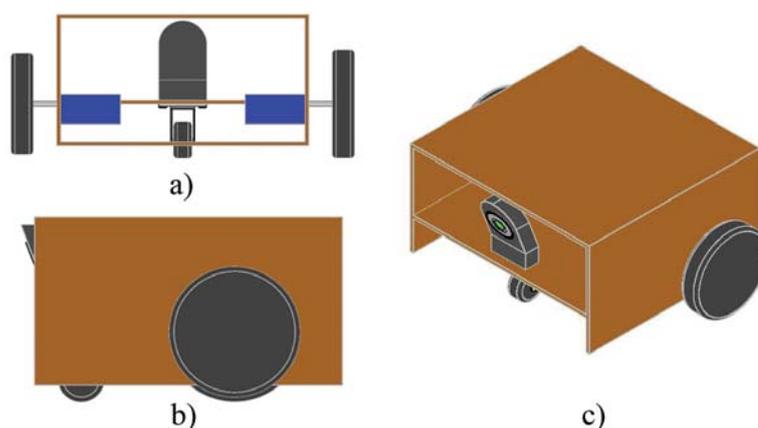


Figura 5-4. Visões da estrutura. a) visão traseira b) visão lateral e c) visão isométrica – Modelo criado e gentilmente cedido por Paula Souza Teixeira

Na frente encontra-se uma roda livre, para auxiliar no equilíbrio do robô. Nas laterais estão fixados dois servo-motores azuis da marca Motor Tech conectados a duas rodas, formando o sistema de tração.

Os servo-motores precisaram ser adaptados para atingir o giro contínuo. A adaptação foi feita segundo o tutorial [PATSKO, 2006]. Estes servo-motores possuem o giro mais forte para uma direção e na montagem os motores ficam invertidos, ou seja, um possuía o giro mais forte para frente enquanto o outro para trás. Por isso foi necessário também inverter a polaridade do motor de corrente contínua (ou motor DC) contido em um dos servo-motores, de forma a proporcionar um movimento mais preciso.

5.2.2 Câmera

A câmera utilizada é uma Quickcam for Notebooks Deluxe do fabricante Logitech. Possui um sensor CMOS, com resolução máxima de 1.3MP e foco manual.

A câmera está localizada em cima da roda livre, na região frontal do robô.

5.2.3 Processamento Central

O processamento do robô é feito em um processador Vortex 86 de 200MHz em um sistema *All-in-one Single Board Computer*, chamado eBox II, fornecido pela ICOP Technology. Esse sistema possui as interfaces básicas de um *desktop*, tais como porta serial, porta paralela, porta PS/2, VGA, USB e *ethernet*, e o processador é capaz de executar instruções de um processador x86 genérico.

Nesta plataforma está rodando um sistema operacional Linux mínimo, baseado na distribuição Gentoo, chamado TinyGentoo. O TinyGentoo é compilado com a biblioteca básica de C uCLibc, versão micro da GNU Libc.

5.2.4 Controle dos Motores

Devido às restrições temporais do acionamento dos motores, detalhados adiante, foi necessário utilizar um microcontrolador. Por questões de familiaridade do autor e disponibilidade, escolheu-se o microcontrolador PIC 16F877A, do fabricante Microchip

A placa, na qual o PIC está instalado, é formada por um sistema de regulação de tensão, uma fonte de tempo, um sistema de compatibilização de tensão para RS-232 e o microcontrolador.

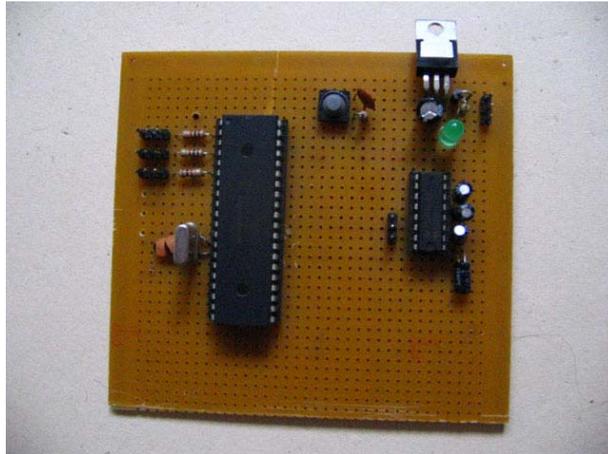


Figura 5-5. Placa de controle dos motores

O sistema de regulação de tensão garante que a tensão fique estável em 5V, caso sua entrada seja maior que esse valor. Tal nível de tensão é o nominal para o funcionamento de todos os outros sistemas.

A fonte de tempo tem como principal componente um cristal de 20MHz montado conforme o manual do PIC [MICROCHIP, 1997]. A essa frequência o PIC executa uma instrução a cada 200ns.

O sistema de compatibilização de tensão para RS-232 é utilizado para a comunicação do PIC com a porta serial do computador onde é executada a inteligência do robô. Sua montagem segue o tutorial em [MESSIAS].

5.2.5 Controle Remoto e Monitoramento e Figa Vision

O robô possui um módulo que permite controle remoto e monitoramento do processamento visual. As operações são disponibilizadas através de uma rede *ethernet*.

Em um processador separado, encontra-se o software que se conecta ao sistema controle remoto e monitoramento do robô, o Figa Vision.

5.3 Visão Lógica

Na visão lógica, os componentes estão separados em funcionalidades, como se pode ver na Figura 5-6.

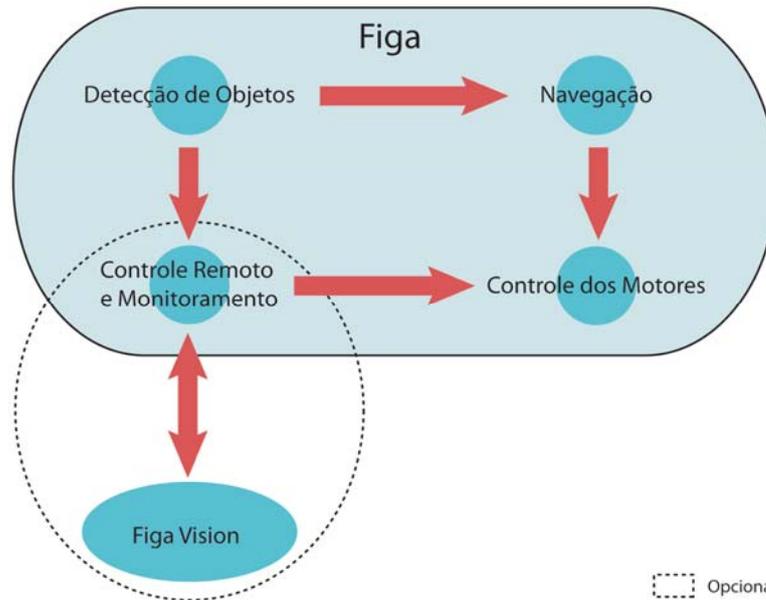


Figura 5-6. Visão lógica. Imagem gentilmente criada e cedida por Vinícius Fabrino

São quatro os módulos funcionais básicos do sistema: módulo de detecção de objetos, módulo de navegação, módulo de controle dos motores e módulo de controle remoto e monitoramento. Para ter acesso ao último é necessário o uso de um outro sistema chamado Figa Vision, que será detalhado adiante. O módulo de controle remoto e monitoramento e o Figa Vision estão marcados como opcionais pois não são parte essencial do sistema.

5.3.1 Controle dos Motores

Os motores são acionados por modulação em largura de pulso (PWM – *Pulse Width Modulation*). Tal modulação permite a transmissão de informação a partir da modificação da largura de um pulso.

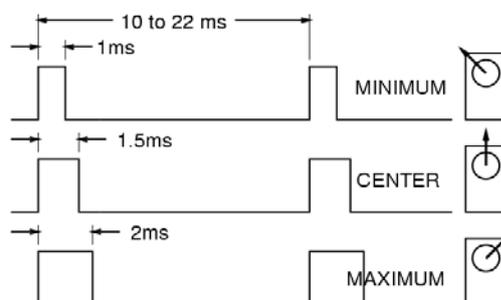


Figura 5-7. Relação entre pulso recebido e angulação de um servomotor

Os servomotores recebem o pulso PWM e convertem em um ângulo em relação à posição inicial. No entanto os servomotores deste projeto são modificados para alcançar o giro contínuo. Assim, se o pulso for abaixo de 1ms o motor irá girar no sentido anti-horário; caso contrário, no sentido horário. O sistema de controle dos motores é feito em dois módulos: comunicação com a porta serial e geração do PWM.

A comunicação com a porta serial é feita utilizando a *Universal Asynchronous Receiver Transmitter* (UART) presente no PIC. O dado recebido corresponde a uma direção (direita, esquerda, frente, trás) ou ao comando pare. O PIC responde com 'o' (OK), se o dado estiver correto, ou 'n' (Not OK), caso contrário. O valor recebido pela porta serial é armazenado para que o sistema de geração do PWM utilize-o para enviar os sinais corretos a cada motor.

O PIC possui um módulo de geração de PWM, porém permite apenas uma saída e como é necessário controlar dois motores, a geração do sinal PWM é feita via software. De forma a garantir o período do pulso, a geração do PWM é feita utilizando a interrupção de tempo.

```
IF tempo = 0.5 ms THEN
    Nível lógico '0' para as portas que desejam
    duty cycle de 0.5ms
ELSIF tempo = 1.5 ms THEN
    Nível lógico '0' para as portas que desejam duty cycle
    de 0.5ms
ELSIF tempo = 20 ms THEN
    Nível lógico '1' em todas as portas
    Reinicie a contagem de tempo
```

Como percebe-se pelo pseudo-código acima, esse módulo gera apenas sinais PWM com *duty cycle* de 0,5ms ou 1,5ms, mas esses são os valores necessários para esta aplicação.

5.3.2 Detecção de Objetos

A detecção de objetos é feita em três passos: aquisição da imagem, segmentação em cores e busca por região vencedora. A imagem capturada da câmera possui tamanho 320x240 pixels, codificada no sistema RGB com 8 bits para cada componente. Um ajuste de brilho é feito automaticamente pela câmera, dispensando esse pré-processamento.

Em seguida, a imagem é utilizada para segmentação de regiões. Para isso foram implementados dois algoritmos: o SOM e o LARFSOM. Ambos possuem dois estágios: treinamento da rede e classificação das instâncias. O estágio de treinamento é feito utilizando pixels seleccionados aleatoriamente. Com a rede treinada, cada *pixel* da imagem é submetido à rede e sua cor é substituída pelos pesos do nó vencedor.

A busca por região vencedora se inicia com a binarização da imagem, transformando o que é verde em preto e o resto em branco. A cor verde é definida pela seguinte fórmula:

$$V(R, G, B) = \begin{cases} 1 & R > L \therefore R^2 > G^2 + B^2 \\ 0 & \text{caso contrário} \end{cases}$$

onde L é um limiar empiricamente escolhido.

Em seguida a imagem é dividida em 9 retângulos. Para cada um é calculada a porcentagem de *pixels* pretos. Aquele retângulo com a maior porcentagem torna-se a região vencedora. Se a maior porcentagem estiver abaixo de 25%, nenhum retângulo é dito vencedor.

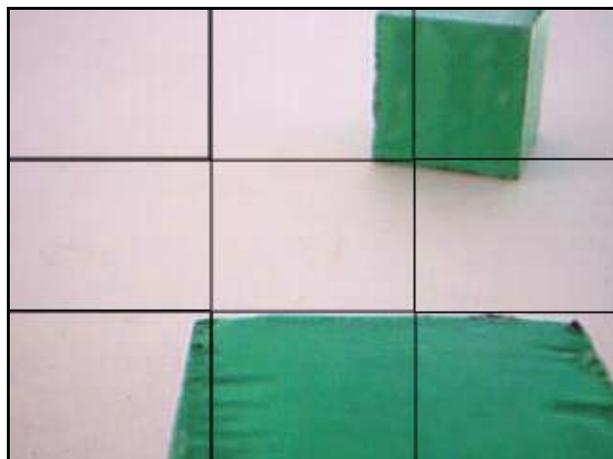


Figura 5-8. Imagem dividida em regiões

5.3.3 Navegação

A estratégia de navegação não utiliza mapas, mas a detecção de objetos. A saída do processo de detecção é o retângulo da imagem com maior chance de conter um objeto verde, ou nenhum retângulo.

O robô segue a seguinte estratégia:

```
IF não existe retângulo THEN
  IF estava buscando em frente THEN
    Vire a esquerda por alguns segundos
  ELSE
    Siga em frente por alguns segundos
ELSE
  IF retângulo = 1 ou retângulo = 4 THEN
    vire à esquerda;
  ELSIF retângulo = 3 ou retângulo = 6 THEN
    vire à direita;
  ELSIF retângulo = 2 ou retângulo = 5 THEN
    vá em frente;
ELSE
  pare; (estágio final, encontrou o objeto)
```

Se não existir objetos na cena, o robô inicia um processo de exploração do ambiente até que um objeto seja enquadrado. Quando isso acontecer, o robô centraliza o objeto e segue em sua direção, parando quando estiver próximo.

5.3.4 Controle Remoto e Monitoramento e Figa Vision

Para ajudar nos estágios iniciais de desenvolvimento e como forma de visualização e apresentação dos resultados, foi criada uma interface de controle remoto e monitoramento do robô.

Essa interface provê através de uma rede *ethernet* os seguintes serviços:

- ⌘ Comandos de movimentação (frente, trás, esquerda, direita e pare);
- ⌘ Comandos de ajuste da imagem (brilho, contraste, coloração e *whiteness*);
- ⌘ Visualização da imagem capturada e do resultado do processamento.

Os serviços estão disponíveis através de um sistema do tipo cliente/servidor, em que o robô faz o papel de servidor e o sistema Figa Vision faz o papel de cliente. O servidor permite a conexão de um usuário por vez através das portas 1024, utilizada para requisição de operações e suas respostas, e 1025, para o envio de imagens.

As mensagens trocadas pelos sistemas são as seguintes:

- ⌘ GET CAPTURE: enviada do cliente ao servidor requisitando o início do envio das imagens;
- ⌘ GET WINDOW INFO: enviada do cliente ao servidor requisitando as informações da janela da imagem, isto é, a largura e altura;
- ⌘ GET IMAGE INFO: enviada do cliente ao servidor requisitando os parâmetros utilizados na captura da imagem, ou seja, brilho, contraste, coloração e *whiteness*;
- ⌘ SET IMAGE INFO: enviada do cliente ao servidor requisitando a mudança dos parâmetros da captura para os valores passados;
- ⌘ MOVE: enviada do cliente ao servidor requisitando a movimentação do robô na direção especificada;
- ⌘ QUIT: pode ser enviada pelas duas partes para indicar o término da conexão;
- ⌘ INVALID: mensagem enviada pelas duas partes para indicar recebimento de mensagem inválida.

O Figa Vision é uma interface gráfica para tais comandos, mostrada na figura a seguir.

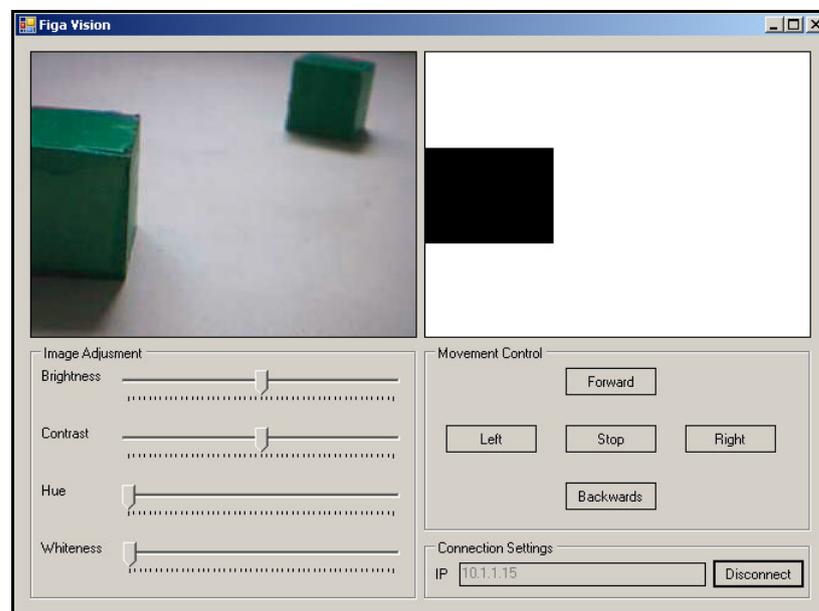


Figura 5-9. Interface gráfica do Figa Vision

Os controles da esquerda permitem o ajuste da câmera para a captura da imagem. Os da direita permitem controlar a movimentação do robô. Embora este seja autônomo, durante a

fase de testes foi preciso controlá-lo remotamente. Na parte inferior encontra-se o controle de conexão com o Figa.

A imagem da esquerda é a capturada pela câmera e a da direita é o resultado do processamento. No exemplo acima, o robô reconhece o retângulo pintado de preto como o predominante.

6 TESTES E RESULTADOS

Este capítulo apresenta os resultados obtidos com a aplicação dos algoritmos de segmentação. A análise desses resultados será feita levando em consideração o tempo de execução e a qualidade da segmentação.

Dois tipos de testes foram executados. Primeiro aplicaram-se os algoritmos às imagens coletadas da USC-SIPI Image Database em um modo *stand-alone*, ou seja, separado da aplicação do robô. Aproveita-se para fazer uma comparação com os resultados obtidos pelos autores do LARFSOM. Em seguida serão analisados os resultados da aplicação dos algoritmos na segmentação de imagens capturadas no ambiente de execução do robô.

A execução dos testes foi feita em duas máquinas: a eBox II, descrita no capítulo anterior, e em um Intel Core 2 Duo E6300 1.87GHz com 2GB de memória RAM.

6.1 Resultados em Imagens da USC-SIPI Image Database

Quatro imagens, comumente utilizadas em análises de algoritmos de processamento de imagens, foram selecionadas: House (a), Pepper (b), Lena (c) e Baboon (d), mostradas na figura abaixo.

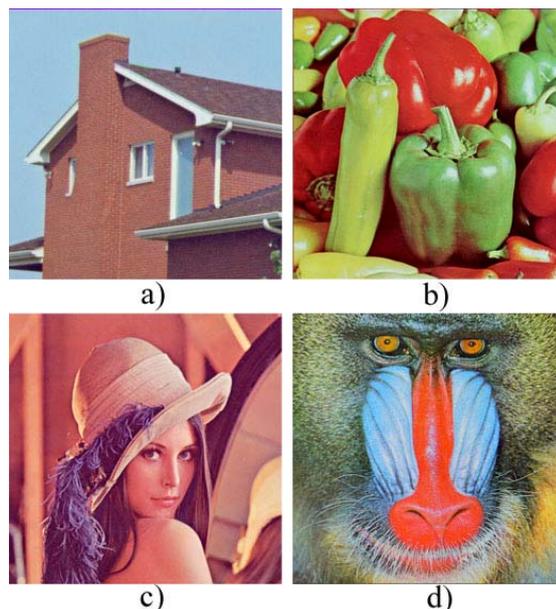


Figura 6-1. Imagens utilizadas nos testes. a) House b) Pepper c) Lena d) Baboon

Os parâmetros escolhidos para o LARFSOM foram: $\rho = 0,05$, $\varepsilon = 0,3$, $d_m = 100$ e quatro valores para a_t , 4,65, 3,65, 2,65 e 1,65. Os limiares de ativação 2,65, 1,65 foram sugeridos em [ARAÚJO & COSTA, 2006] e os outros foram escolhidos nos experimentos deste trabalho por motivos justificados adiante. Para o SOM foram utilizados três tamanhos de mapa, 2x2, 3x3, 4x4 e 5x5, e a taxa de aprendizagem inicial de 0,7, valor escolhido empiricamente.

Nesta seção será dado um enfoque à qualidade da segmentação, porém sem desconsiderar o tempo de execução. Os testes foram executados apenas na máquina com Intel Core 2 Duo.

6.1.1 Imagem House

É a imagem mais simples de segmentar. Possui 33.925 cores e poucas são dominantes [ARAÚJO & COSTA, 2006]. Seu tamanho é 256x256 *pixels*. A Figura 6-2 mostra os resultados do SOM à esquerda e do LARFSOM à direita. Os tamanhos da grade do SOM são 2x2 para (a), 3x3 para (b), 4x4 para (c) e 5x5 para (d). Os limiares de ativação (a_t) são 1,65 para (e), 2,65 para (f), 3,65 para (g) e 4,65 para (h).

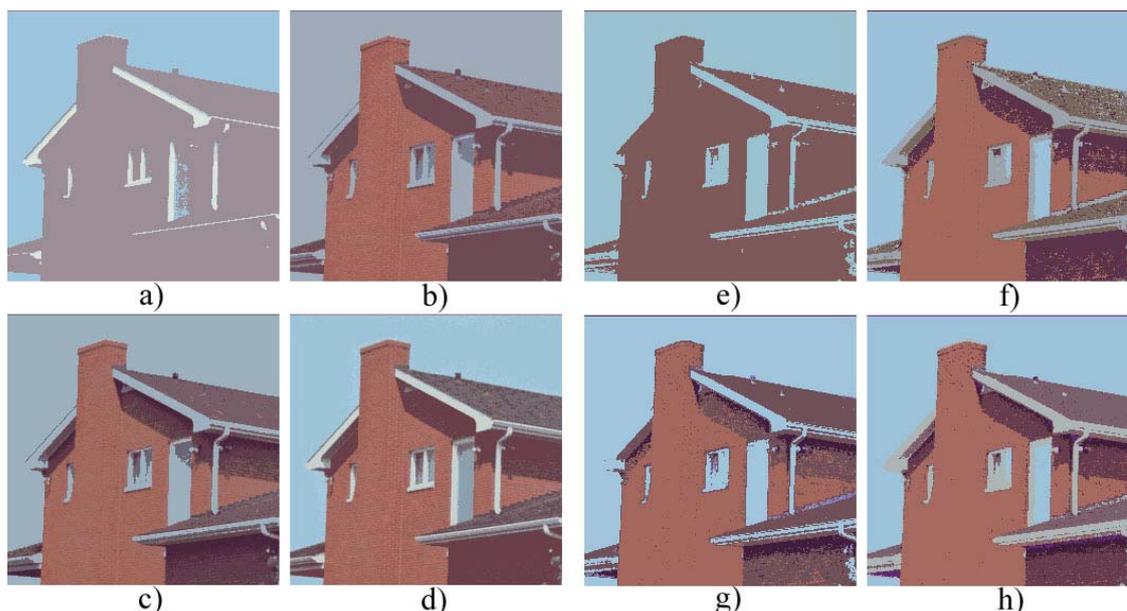


Figura 6-2. Resultados da aplicação do SOM (a)-(d), e do LARFSOM (e)-(h) na imagem House.

Ambos conseguem bons resultados para a segmentação. O LARFSOM consegue com apenas duas cores, em (a), consegue diferenciar não só o fundo da casa, mas também as janelas. O SOM por outro lado consegue uma representação muito boa da imagem com 25 cores (d).

Image m	Nº de nós	Iterações	Tempo de Treinamento (s)	Tempo de Classificação (s)
a)	4	65536	0,060	~0,000
b)	9	65536	0,060	0,020
c)	16	65536	0,070	0,020
d)	25	65536	0,090	0,030
e)	2	18	~0,000	~0,000
f)	5	97	~0,000	0,010
g)	5	57	~0,000	0,010
h)	7	155	~0,000	0,020

Tabela 6-1. Resultados da aplicação do SOM e do LARFSOM na imagem House

6.1.2 Imagem Pepper

A imagem pepper possui tamanho 512x512 pixels e 183.525 cores. Como percebe-se as bordas dos objetos são definidas por grandes contrastes de cores. A Figura 6-3 mostra os resultados do SOM à esquerda e do LARFSOM à direita. Os tamanhos da grade do SOM são

2x2 para (a), 3x3 para (b), 4x4 para (c) e 5x5 para (d). Os limiares de ativação (a_T) são 1,65 para (e), 2,65 para (f), 3,65 para (g) e 4,65 para (h).

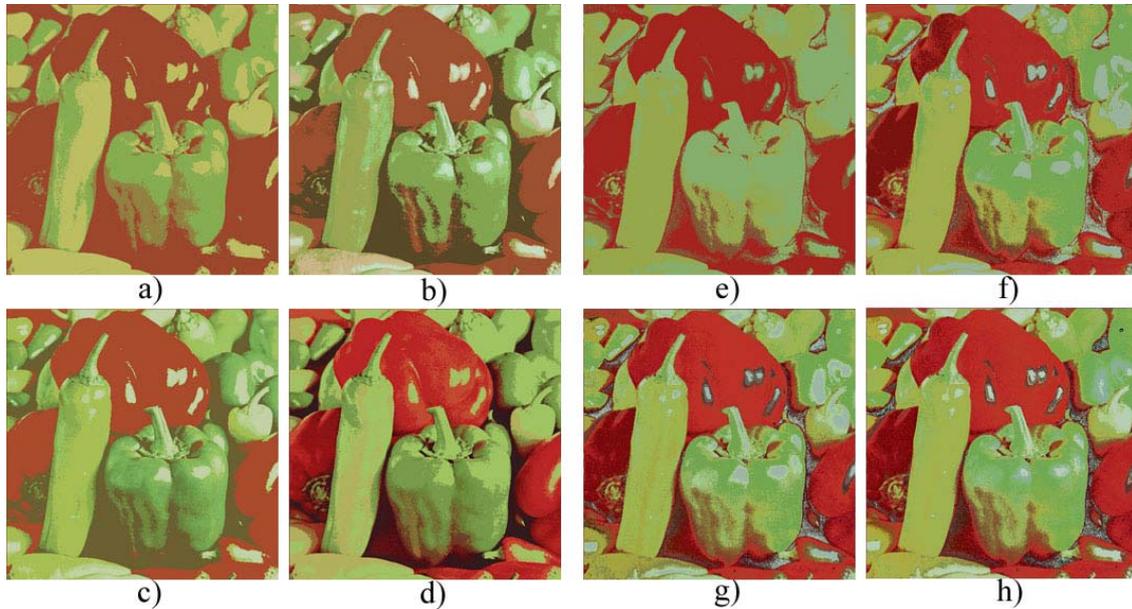


Figura 6-3. Resultados da aplicação do SOM (a)-(d), e do LARFSOM (e)-(h) na imagem Pepper.

Novamente os dois conseguem bons resultados para a segmentação. O SOM com um mapa 5x5 (d) separa melhor as regiões como os brilhos. No resultado com 25 nós do LARFSOM (g) aparece uma borda escura não existente na imagem original ao redor do brilho do pimentão vermelho central. Por outro lado o LARFSOM consegue detectar a tonalidade de vermelho dominante em todos os experimentos e a utiliza para a representação dessa classe.

Image m	Nº de nós	Iterações	Tempo de Treinamento (s)	Tempo de Classificação (s)
a)	4	262144	0,390	0,030
b)	9	262144	0,410	0,040
c)	16	262144	0,430	0,060
d)	25	262144	0,470	0,100
e)	4	121	~0,000	0,060
f)	4	37	~0,000	0,050
g)	25	4228	0,010	0,140
h)	48	11532	0,020	0,210

Tabela 6-2. Resultados da aplicação do SOM e do LARFSOM na imagem Pepper

6.1.3 Imagem Lena

A imagem Lena possui mais cores que os casos anteriores e várias cores similares, tornando a segmentação mais difícil. Seu tamanho é 512x512 *pixels*. A Figura 6-4 mostra os resultados do SOM à esquerda e do LARFSOM à direita. Os tamanhos da grade do SOM são 2x2 para (a), 3x3 para (b), 4x4 para (c) e 5x5 para (d). Os limiares de ativação (a_T) são 1,65 para (e), 2,65 para (f), 3,65 para (g) e 4,65 para (h).

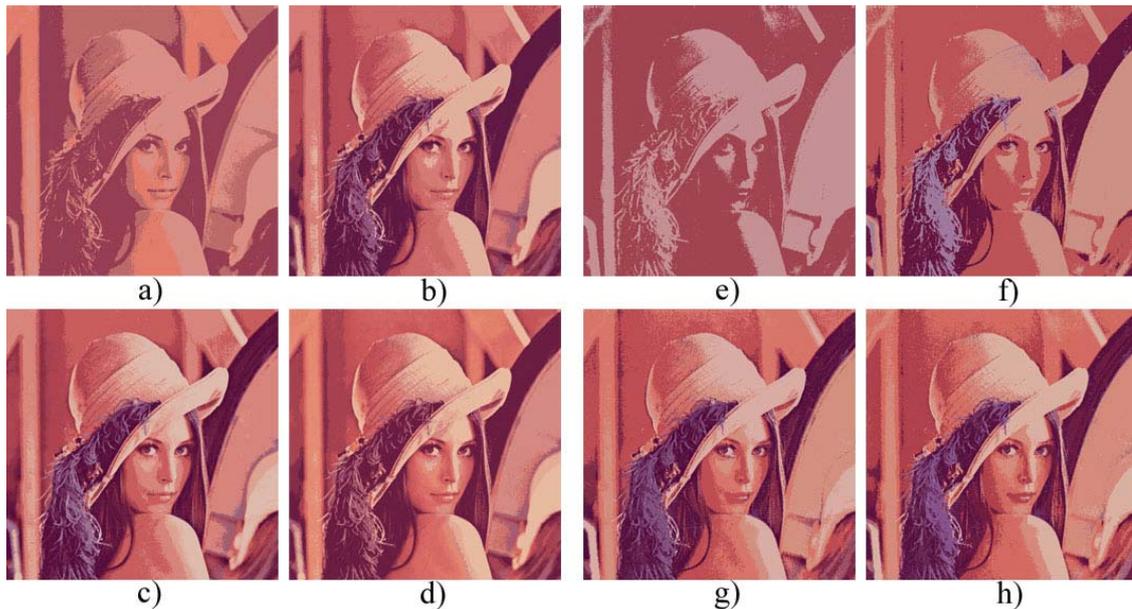


Figura 6-4. Resultados da aplicação do SOM (a)-(d), e do LARFSOM (e)-(h) na imagem Lena

Comparando os resultados com 4 cores do LARFSOM, imagens (f) e (g), com o resultado com o mesmo número de cores do SOM, imagem (a), percebe-se que o primeiro foi capaz de selecionar melhor as cores.

Aqui percebe-se que a escolha dos parâmetros é fundamental para o resultado e deve ser orientada pela aplicação. Enquanto com duas cores é possível separar a casa do fundo e das janelas, na Figura 6-2 (e), acima em (e) algumas regiões foram perdidas, unindo parte do rosto e costas com o cabelo abaixo do chapéu. No entanto nessa mesma imagem, percebemos uma boa separação das regiões iluminadas e das não iluminadas, independente de sua cor básica.

Image m	Nº de nós	Iterações	Tempo de Treinamento (s)	Tempo de Classificação (s)
a)	4	262144	0.360	0.030

b)	9	262144	0.380	0.050
c)	16	262144	0.410	0.070
d)	25	262144	0.440	0.110
e)	2	47	~0.000	0.040
f)	4	62	~0.000	0.550
g)	4	62	~0.000	0.060
h)	9	126	~0.000	0.070

Tabela 6-3. Resultados da aplicação do SOM e do LARFSOM na imagem Lena

6.1.4 Imagem Baboon

Esta é a imagem mais complexa devido ao grande número de cores (230.427) e às abruptas mudanças de cor [ARAÚJO & COSTA, 2006]. Seu tamanho é 512x512 *pixels*. A Figura 6-5 mostra os resultados do SOM à esquerda e do LARFSOM à direita. Os tamanhos da grade do SOM são 2x2 para (a), 3x3 para (b), 4x4 para (c) e 5x5 para (d). Os limiares de ativação (a_T) são 1.65 para (e), 2.65 para (f), 3.65 para (g) e 4.65 para (h).

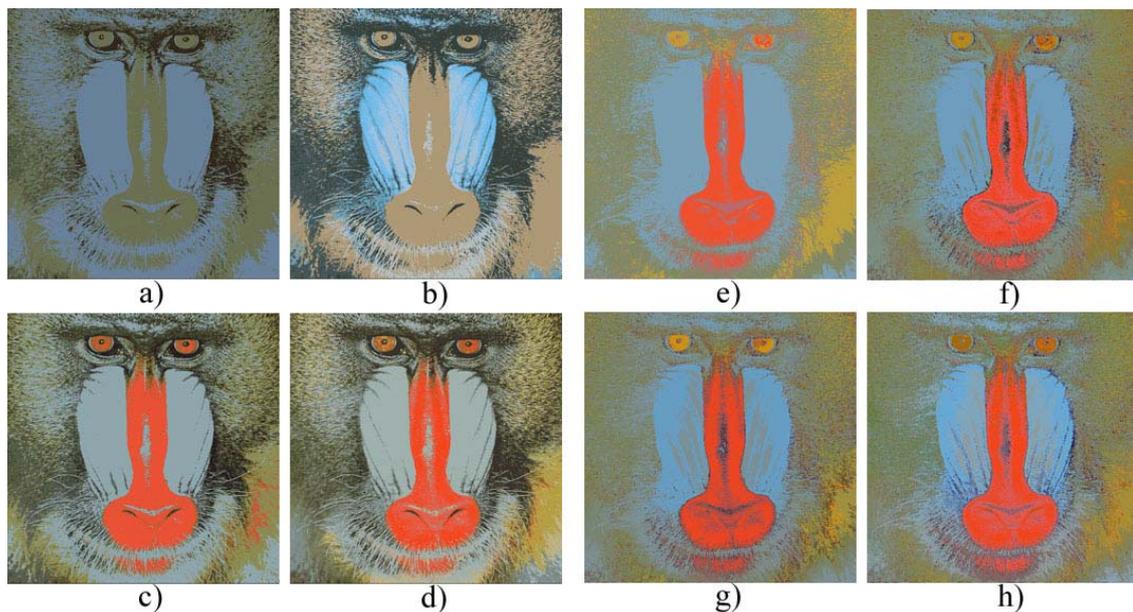


Figura 6-5. Resultados da aplicação do SOM (a)-(d), e do LARFSOM (e)-(h) na imagem Baboon

Contrariando as expectativas, o LARFSOM não gerou bons resultados com essa imagem. Até mesmo com um maior número de cores, como na imagem (h), as cores mais relevantes não se sobressaíram.

Da mesma forma como diferentes limiares de ativação foram necessários para esta implementação, pois os resultados encontrados em [ARAÚJO & COSTA, 2006] não puderam

ser reproduzidos com os parâmetros sugeridos, novas combinações de parâmetros precisam ser exploradas para esta imagem. Sugere-se a modificação dos parâmetros de aprendizagem, ρ e ε , uma vez que o número de nós não foi o que determinou a qualidade do resultado, como pode ser visto na tabela 6-4.

Image m	Nº de nós	Iterações	Tempo de Treinamento (s)	Tempo de Classificação (s)
a)	4	262144	0.350	0.020
b)	9	262144	0.430	0.040
c)	16	262144	0.460	0.060
d)	25	262144	0.680	0.110
e)	5	150	~0.000	0.570
f)	12	1163	~0.000	0.100
g)	13	514	~0.000	0.090
h)	34	3687	~0.000	0.360

Tabela 6-4. Resultados da aplicação do SOM e do LARFSOM na imagem Baboon

6.1.5 Observações

Nos resultados mostrados acima percebe-se que a qualidade da segmentação é aparentemente boa para os dois algoritmos. Vê-se também a capacidade do LARFSOM de encontrar as cores relevantes, mesmo com um número reduzido de nós, como observado em [ARAÚJO & COSTA, 2006]. A grande diferença está na rápida convergência do LARFSOM, o que capacita este algoritmo a ser utilizado em aplicações onde tempo é crítico.

Mas deve-se observar as diferenças entre os resultados obtidos neste trabalho e os resultados do autor do LARFSOM. Primeiramente, em [ARAÚJO & COSTA, 2006] os testes foram executados com os limiares de ativação 1.0, 1.65 e 2.65. Devido a resultados fracos obtidos com o limiar 1.0, tal valor para o parâmetro foi desconsiderado e novos valores, como mostrados nos experimentos acima foram utilizados.

As outras diferenças estão nas situações excepcionais que foram percebidas: a primeira em que a convergência é muito rápida, em cerca de duas iterações, resultando em uma segmentação pobre, e a segunda em que o algoritmo fica preso em um mínimo local. A primeira acontece freqüentemente ao se utilizar o limiar de ativação de 1.0, motivo pelo qual

ele foi evitado. Para contornar a segunda situação foi incluída outra condição de parada que é o número de iterações ser igual à quantidade de *pixels* na imagem.

Embora o SOM, da forma como foi implementado neste trabalho, possua uma grande desvantagem em relação ao tempo de computação, problemas como os citados no parágrafo anterior não aconteceram.

Os resultados obtidos nesses experimentos guiaram os experimentos no ambiente do robô, tratados em mais detalhes na próxima seção.

6.2 Resultados no Ambiente do Robô

Nesta seção faremos uma análise mais profunda em relação ao tempo de execução dos algoritmos e a influência de cada um no processo de encontrar a região da imagem com predominância da cor verde, assim como no método descrito no capítulo anterior.

No ambiente do robô foi analisada a aplicação dos algoritmos em uma situação em que dois cubos verdes aparecem, situação ilustrada na Figura 6-6. Os parâmetros utilizados no LARFSOM são os mesmos dos testes anteriores, mas com o limiar de ativação $a_T = 2,65$. Para o SOM utilizou-se uma grade de 3x3.

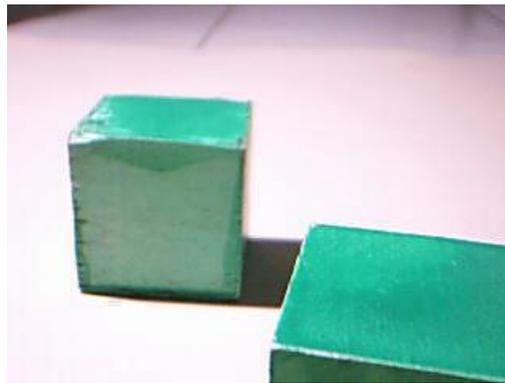


Figura 6-6. Imagem utilizada para os testes no ambiente do robô

A aplicação dos algoritmos nesse ambiente resulta nas seguintes imagens.

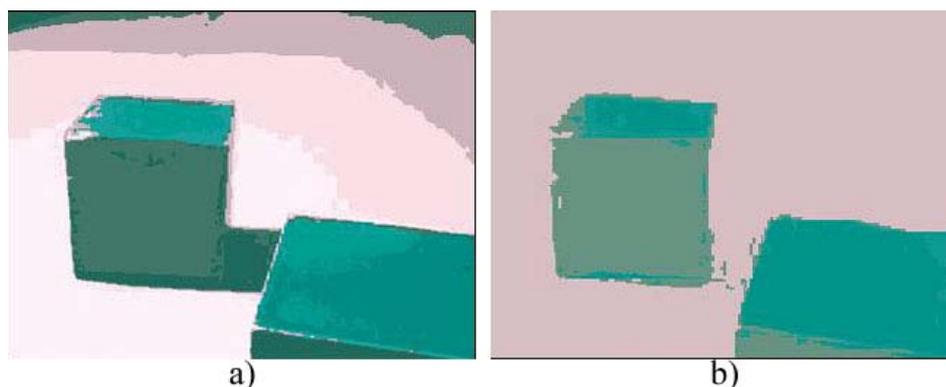


Figura 6-7. a) Imagem 6-6 segmentada pelo SOM b) Imagem 6-6 segmentada pelo LARFSOM

Como é possível perceber, o LARFSOM diminuiu drasticamente o número de cores e conseguiu separar o fundo dos objetos, bem como as sombras. O SOM classificou a sombra de um objeto como sendo verde e também a região mais escura do fundo.

Para tal situação foi capturado um conjunto massivo de dados que está sumarizado nas tabelas a seguir. Essas tabelas mostram algumas medidas estatísticas extraídas das seguintes características: número de iterações que o algoritmo teve que rodar, número de cores final, tempo de treinamento da rede e tempo de classificação dos *pixels*.

Característica	Média	Desvio-padrão	Valor máximo	Valor mínimo
Iterações	263,5977	2660,6561	76800	2
Nº de cores	4,0181	1,4025	19	2
Tempo de treinamento	0,000471	0,004175	0,120002	~0,000000
Tempo de classificação	0,014377	0,005116	0,060001	0,010000

Tabela 6-5. Estatísticas da execução do LARFSOM na máquina com processador Intel Core 2 Duo

Característica	Média	Desvio-padrão	Valor máximo	Valor mínimo
Tempo de treinamento	0,098402	0,016905	0,490007	0,070001
Tempo de classificação	0,023215	0,008321	0,340005	0,020000

Tabela 6-6. Estatísticas da execução do SOM na máquina com processador Intel Core 2 Duo

A quantidade de iterações no LARFSOM varia bastante, por isso foi criado também um histograma para esses dados. O tempo de treinamento, como já visto, é muito menor para o LARFSOM, cerca de 200 vezes mais rápido. O tempo de classificação é muito próximo, porém o LARFSOM ainda leva vantagem devido ao reduzido número de cores. Como visto

na tabela, a média de cores geradas pelo LARFSOM é de aproximadamente 4, contra as 9 geradas no SOM.

Característica	Média	Desvio-padrão	Valor máximo	Valor mínimo
Iterações	419,873	4147,460	76800	2
Nº de cores	4,235	1,328	14	2
Tempo de treinamento	0,038803	0,363354	7,028968	0,000126
Tempo de classificação	0,540474	0,075959	1,032928	0,404299

Tabela 6-7. Estatísticas da execução do LARFSOM na máquina com processador Vortex 86

Característica	Média	Desvio-padrão	Valor máximo	Valor mínimo
Tempo de treinamento	4,477624	0,621809	6,786104	3,201113
Tempo de classificação	0,847259	0,024992	0,914179	0,811829

Tabela 6-8. Estatísticas da execução do SOM na máquina com processador Vortex 86

Na eBox a proporção entre o tempo de classificação do LARFSOM e SOM diminui para um pouco mais de 100 vezes, porém continua a ser uma diferença significativa. No entanto, deve-se dar importância nesse caso aos valores absolutos. Os tempos médios de treinamento do SOM nessa arquitetura são proibitivos, enquanto os do LARFSOM ainda são aceitáveis para a média, mas igualmente proibitivos para o pior caso. Por isso foi necessário analisar esse valores de outra forma, como a seguir.

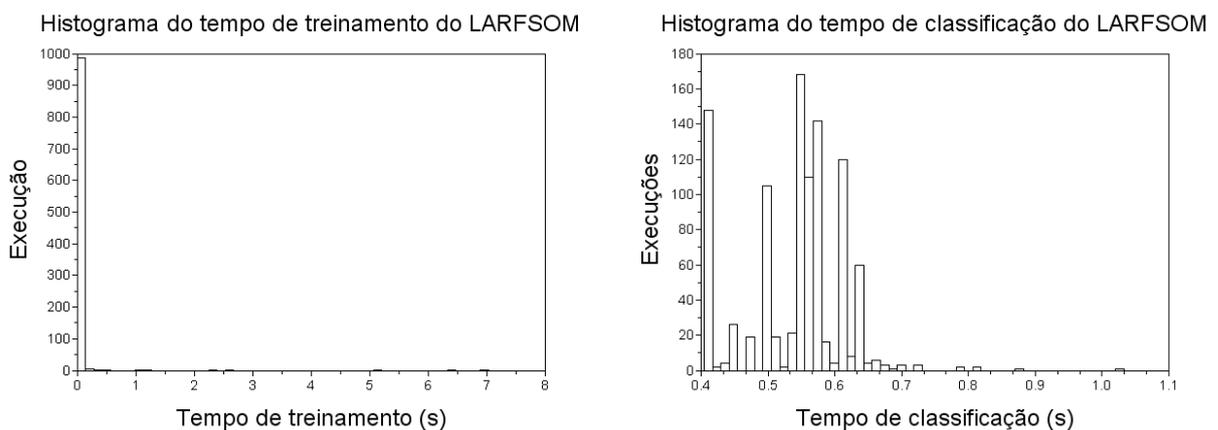


Figura 6-8. Histogramas do tempo de execução e de classificação do LARFSOM

Embora os pior caso do treinamento do LARFSOM seja proibitivo, em 1000 execuções mais de 950 ficaram em menos de 0,25 ms. O tempo de classificação sofre uma variação maior, mas a parte mais significativa está abaixo de aproximadamente 0.66 ms. Com tais

limitantes superiores consegue-se um processamento levemente maior que 1 *fps*. Dessa forma considera-se que o LARFSOM é uma boa escolha, mesmo para arquiteturas mais limitadas, como é o caso da eBox II.

7 CONSIDERAÇÕES FINAIS

Os conceitos básicos para o desenvolvimento deste trabalho foram vistos, tais como as técnicas de navegação baseadas em visão, as redes neurais não-supervisionadas, o problema da segmentação em detalhes e soluções baseadas em redes neurais.

Todo esse conhecimento descrito, aliado a conhecimento técnico adquirido ao longo do curso e em atividades extra-curriculares, tornaram possível a implementação de um robô capaz de detectar objetos de uma dada cor e se movimentar até as proximidades desse.

As técnicas utilizadas foram descritas em detalhe e uma avaliação de sua performance foi feita, tanto em termos de qualidade dos resultados quanto de velocidade de processamento, permitindo concluir que em uma arquitetura de processamento mais poderosa, ambos os algoritmos conseguem processar a uma taxa maior ou igual a 0,5 *fps*, mas em uma arquitetura mais simples, como a da eBox, o Mapa Auto-Organizável é inadequado, mas o LARFSOM se mostra como um forte competidor, pois alcança os objetivos na maior parte do tempo, como foi mostrado nos resultados.

Os resultados encontrados permitem a continuação deste trabalho, seja por melhoramentos ou pela extensão do escopo do projeto. Assim algumas propostas para trabalhos futuros devem ser citados:

⌘ Quanto à parte física do projeto, sugere-se melhorar a estrutura mecânica, sobretudo os motores, que possuem um eixo frágil, permitindo o estabelecimento de uma arquitetura robusta para estudos em inteligência artificial aplicada à robótica móvel;

⌘ Quanto à estrutura computacional, indica-se uma análise mais profunda das necessidades computacionais para algoritmos de visão computacional desta natureza, de forma a determinar ou não o quão adequada é essa arquitetura utilizada;

⌘ Quanto aos algoritmos, deve-se reavaliar o critério de parada para o SOM, de forma a diminuir o tempo de treinamento desta rede, e verificar as causas da convergência muito rápida do LARFSOM, em certa de duas iterações, e os casos em que o algoritmo fica preso em um possível mínimo local;

⌘ E por fim, ao pensar em aumentar o escopo do projeto, deve-se considerar a inserção de técnicas de detecção de formas permitindo uma maior diferenciação dos objetos além da cor.

REFERÊNCIAS BIBLIOGRÁFICAS

- [HORN, 1986] HORN, Berthold Klaus Paul. **Robot Vision**. New York: MIT Press, 1986.
- [MALIS, 2002] MALIS, Ezio. **Survey of Vision Based Robot Control**. *European Naval Ship Design, Captain Computer IV Forum*. Brest, France, April, 2002.
- [DESOUZA et al, 2002] DESOUZA, Guilherme N.; KAK, Avinash C. **Vision for Mobile Robot Navigation: A Survey**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 24, No 2, February, 2002.
- [ARAÚJO & COSTA, 2006] ARAÚJO, Aluizio R. F.; COSTA, Diogo C. **Local Adaptive Receptive Field Self-Organizing Map for Image Segmentation**. *Proceedings of the Workshop on Self-Organizing Networks (WSOM 2007)*, Bielefeld, Germany, September 03-06.
- [COSTA, 2007] COSTA, Diogo C. **Mapa Auto-Organizável com Campo Receptivo Adaptativo Local para Segmentação de Imagens**. 2007. 109 f. Dissertação (Mestrado em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco. Recife.
- [BORENSTEIN et al., 1996] BORENSTEIN, J.; EVERETT, H. R.; FENG, L. **Navigating Mobile Robots: Systems and Techniques**. Wellesley, MA.: A K Peters, 1996.
- [THRUN, 2003] THRUN, Sebastian. **Robotic Mapping: A Survey**. In: *Exploring Artificial Intelligence in the New Millennium*. San Francisco, CA.: Morgan Kauffman, 2003. cap. 1, p. 1-36.
- [TEMIZER, 2001] TEMIZER, Selim. **Optical Flow Based Robot Navigation**. Online em: http://people.csail.mit.edu/lpk/mars/temizer_2001/Optical_Flow/ . Acesso em: 23 de janeiro de 2008.
- [DUDA et al., 2000] DUDA, Richard O.; HART, Peter E.; STORK, David G. **Pattern Classification**. 2.ed. New York: Wiley, 2000. p.517-600.
- [DAYAN, 1999] DAYAN, Peter. **Unsupervised Learning**. In: WILSON, Robert A.; KEIL, Frank C. *The MIT Encyclopedia of the Cognitive Sciences*. New York: MIT Press, 2001.
- [HAYKIN, 1998] HAYKIN, Simon. **Neural Networks: A Comprehensive Foundation**. 2.ed. New York: Prentice-Hall, 1998. p. 465-505.

[BEALE & JACKSON, 1990] BEALE, R.; JACKSON, T. **Neural Computing: An Introduction**. Philadelphia: Institute of Physics Publishing, 1990.

[XU & WUNSCH, 2005] XU, Rui; WUNSCH, Donald. **Survey of Clustering Algorithms**. *IEEE Transactions on Neural Networks*, Vol. 16, No. 3, May, 2005.

[CHENG et al., 2001] CHENG, H. D.; JIANG, X. H.; SUN, Y.; WANG, J. **Color image segmentation: advances and prospects**. *Pattern Recognition*, Vol. 34, No. 12, 2001. pp.2259-2281

[SKARBEEK & KOSCHAN, 1994] SKARBEEK, Wladyslaw; KOSCHAN, Andreas. **Colour Image Segmentation – A Survey**. Technical Report. Technical University Berlin, 1994.

[LUCCHESI & MITRA, 2001] LUCCHESI, L.; MITRA, S. K. **Color Image Segmentation: A State-of-the-Art Survey**. *Proceedings of The Indian National Science Academy*, 2001.

[DONG & XIE, 2005] DONG, Gui; XIE, Ming. **Color Clustering and Learning for Image Segmentation Based on Neural Networks**. *IEEE Transactions on Neural Networks*, Vol. 16, No. 4, July, 2005.

[PATSKO, 2006] PATSKO, Luís F. **Tutorial de Adaptação de Servo-Motores**. Online em: <http://www.maxwellbohr.com.br/downloads/Tutorial%20Mecanica%20-%20Adaptacao%20de%20servo-motores.pdf>. Acesso em: 23 de janeiro de 2008.

[MICROCHIP, 1997] MICROCHIP. **PICmicro™ Mid-Range MCU Family Reference Manual**. Online em: <http://ww1.microchip.com/downloads/en/devicedoc/33023a.pdf>. Acesso em: 23 de janeiro de 2008.

[MESSIAS] MESSIAS, Antônio R. **Conectando 8 Teclados na Porta Serial Através de um Microcontrolador PIC 16F877 para Controle de Acessos**. Online em: <http://rogercom.com/PortaSerial/ControleAcesso/Controle.htm>. Acesso em: 23 de janeiro de 2008.